Forest biometrics with examples in R

Lauri Mehtätalo
University of Eastern Finland
School of Computing

January 4, 2013
# Contents

## 1 Random variables

1.1 Introduction to random variables ................................. 1  
1.2 Distribution of a random variable ............................... 2  
  1.2.1 Univariate distribution ................................. 2  
  1.2.2 Making transformations to random variables ................. 14  
  1.2.3 Quantile function .................................. 17  
  1.2.4 Multivariate distribution ............................... 19  
1.3 Expectation, variance and covariance ......................... 26  
  1.3.1 Expectation .................................. 26  
  1.3.2 Variance and standard deviation ....................... 29  
  1.3.3 Covariance and correlation ......................... 30  
  1.3.4 Algebraic rules for variance, and covariance ............ 32  
  1.3.5 Covariance and correlation matrices .................. 32  
  1.3.6 Sample estimators ................................ 34  
1.4 Common distribution functions .............................. 36  
  1.4.1 Discrete distributions ............................... 36  
  1.4.2 Continuous distributions ............................ 38  
  1.4.3 Distributions of important transformations of a standard Normal variate ................................. 45  
  1.4.4 Specialities related to modelling tree diameter ........ 46  
1.5 Fitting distribution functions to data ...................... 47  
  1.5.1 The method of moments ............................... 47  
  1.5.2 Maximum likelihood ................................. 52  
  1.5.3 Other methods .................................. 62  
1.6 Linear prediction .................................. 64  
1.7 Exercises .................................. 66
2 Linear model

2.1 Single-predictor regression ........................................... 74
2.1.1 Model formulation ................................................. 74
2.1.2 Estimation with least squares .................................... 76
2.1.3 Modeling nonlinear relationships ............................... 78

2.2 Multiple regression ...................................................... 84
2.2.1 Model formulation .................................................. 84
2.2.2 Estimation with least squares .................................... 87
2.2.3 The design matrix .................................................. 90
2.2.4 Least squares for LM with a general residual variance structure 94

2.3 Inference and tests of the regression relationship ........ 101
2.3.1 Sums of squares ................................................... 101
2.3.2 The assumption normality of residuals ......................... 102
2.3.3 F-test for the significance of regression ....................... 103
2.3.4 The t-test for coefficients ....................................... 104
2.3.5 Testing several coefficients at the same time ............... 106

2.4 Checking the validity of assumptions ............................ 108
2.4.1 Model shape ....................................................... 108
2.4.2 Independence of observations .................................. 109
2.4.3 Constant residual variance ..................................... 111
2.4.4 Normality ........................................................ 112

2.5 Analysis of variance .................................................. 112

2.6 Prediction ............................................................... 115

2.7 Estimation using maximum likelihood ......................... 117
2.7.1 ML for the single predictor regression ......................... 118
2.7.2 ML for the linear model with uncorrelated errors and constant variances ..................................................... 119
2.7.3 ML for LM with a general residual variance structure ..... 120
2.7.4 Restricted maximum likelihood ................................. 120

2.8 About modeling strategies ......................................... 121
2.8.1 Selection of predictors ........................................... 121
2.8.2 The purpose of modeling ....................................... 121

2.9 Exercises ............................................................... 122

3 Linear mixed models .................................................... 125

3.1 Single level of grouping ............................................. 126
3.1.1 The variance component model .................................. 126
3.1.2 Mixed-effects model with a random constant ............... 129
## CONTENTS

3.1.3 Mixed-effects model with a random constant an slope . . . . . 132  
3.1.4 Checking of the assumptions of the mixed model . . . . . . . 135  
3.1.5 Tests on the model . . . . . . . . . . . . . . . . . . . . . . 138  
3.2 Multiple levels of grouping . . . . . . . . . . . . . . . . . . . . . 140  
3.2.1 A nested multilevel model . . . . . . . . . . . . . . . . . . 140  
3.2.2 Crossed grouping structures . . . . . . . . . . . . . . . . . . 151  
3.3 Matrix formulation . . . . . . . . . . . . . . . . . . . . . . . . . . 161  
3.3.1 Single level of grouping . . . . . . . . . . . . . . . . . . . . 161  
3.3.2 Multiple levels of grouping . . . . . . . . . . . . . . . . . . 165  
3.3.3 Crossed grouping structures . . . . . . . . . . . . . . . . . . 166  
3.3.4 The general matrix formulation . . . . . . . . . . . . . . . . 166  
3.4 Estimation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 167  
3.4.1 Analysis of variance . . . . . . . . . . . . . . . . . . . . . . 167  
3.4.2 Maximum Likelihood . . . . . . . . . . . . . . . . . . . . . . 167  
3.4.3 Restricted maximum likelihood . . . . . . . . . . . . . . . . 168  
3.5 Prediction of random effects . . . . . . . . . . . . . . . . . . . . . 169  
3.6 Extending the linear mixed model . . . . . . . . . . . . . . . . . . . 172  
3.7 An analysis of H-D curve . . . . . . . . . . . . . . . . . . . . . . . 172  
3.7.1 Selection of model form . . . . . . . . . . . . . . . . . . . . 172  
3.7.2 Fitting the linear mixed-effects model . . . . . . . . . . . . . . 176  
3.7.3 Model application . . . . . . . . . . . . . . . . . . . . . . . . 195  
3.8 Exercises . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 202

4 Nonlinear models . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205  
4.1 Nonlinear fixed-effects model . . . . . . . . . . . . . . . . . . . . 205  
4.1.1 Introduction and motivating example . . . . . . . . . . . . . . 205  
4.1.2 The nonlinear regression model . . . . . . . . . . . . . . . . 210  
4.1.3 Estimation using nonlinear least squares . . . . . . . . . . . . 210  
4.1.4 Initial estimates . . . . . . . . . . . . . . . . . . . . . . . . . 216  
4.1.5 Different parameterizations in nls models . . . . . . . . . . . 218  
4.2 Nonlinear mixed-effects models . . . . . . . . . . . . . . . . . . . 220  
4.2.1 Single level of grouping . . . . . . . . . . . . . . . . . . . . 220  
4.2.2 Multiple levels of grouping . . . . . . . . . . . . . . . . . . 231  
4.3 Exercises . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 233

5 GLM’s and GLMM’s . . . . . . . . . . . . . . . . . . . . . . . . . . . . 235  
5.1 Formulation of the LM as a GLM . . . . . . . . . . . . . . . . . . . 236  
5.2 GLM for exponential family . . . . . . . . . . . . . . . . . . . . . 237
### CONTENTS

5.2.1 Model formulation ........................................ 237  
5.2.2 Logistic regression ........................................ 238  
5.2.3 Poisson regression ......................................... 239  
5.2.4 Estimation .................................................. 240  
5.2.5 Inference and tests ........................................ 244  
5.3 Weibull regression ........................................... 246  
5.4 Generalized linear mixed models ............................ 250  
5.5 Exercises ....................................................... 255  
6 Model systems .................................................. 257  
6.1 Types of model systems ....................................... 257  
6.2 Two stage least squares ...................................... 258  
6.2.1 Illustration through height and volume models .......... 258  
6.2.2 General formulation ....................................... 259  
6.3 Seemingly unrelated regression .............................. 259  
6.3.1 Model formulation ......................................... 259  
6.3.2 Estimation .................................................. 261  
6.3.3 Prediction .................................................. 261  
6.3.4 Why simultaneous estimation ............................. 261  
6.4 Three stage least squares .................................... 262  
6.5 Simultaneous mixed models ................................ 263  
6.6 Exercises ....................................................... 271  
7 Modeling tree size distributions ............................. 273  
7.1 Background .................................................... 273  
7.1.1 Overview of approaches ................................... 273  
7.1.2 Tree size and size distribution .......................... 275  
7.2 Plotting an empirical diameter distribution ............... 275  
7.2.1 Histogram .................................................. 275  
7.2.2 Smoothing .................................................. 276  
7.3 Computations with diameter distributions ................. 279  
7.3.1 Transformation of tree size ............................... 280  
7.3.2 Basal-area weighted distribution ........................ 286  
7.3.3 Weighted distributions .................................... 296  
7.3.4 Scaling the weighted distributions ........................ 303  
7.3.5 Utilizing the arithmetic relationships of stand variables . 312  
7.4 Modeling diameter distributions for prediction ............ 324  
7.4.1 Parameter prediction method .............................. 324
CONTENTS

7.4.2 Parameter recovery approaches .................................. 351
7.4.3 Calibrating the predictions ...................................... 355
7.5 Exercises ................................................................. 356

8 Additional topics ......................................................... 361
  8.1 Regression splines .................................................. 361
  8.2 Quantile regression .................................................. 367
  8.3 Numerical computations .......................................... 373
    8.3.1 Numerical integration ....................................... 373
    8.3.2 Solving equations ............................................. 375
  8.4 exercises .............................................................. 376

A Matrix algebra ......................................................... 379

B A short story on R .................................................... 389
Chapter 1

Random variables

1.1 Introduction to random variables

Random variables are real-valued variables with value specified by a random process. The value the random variable gets is called a realization of the random variable. Thus, there is a distinction between the random variable and the realized value, the former being usually denoted by a capital letter and the latter by a lowercase letter.

Example 1.1. Let us think of trees of a large forest stand $s$.

a The diameter of a tree at the breast height in stand $s$ is a random variable, which can be denoted by $X$. If we go to the stand $s$ and observe that the diameter of a randomly selected tree is $x$, we have obtained observation $X = x$. In this way of thinking, the diameter of a given tree is a fixed number, and the randomness arises through random selection of the tree from the fixed, finite population of trees in the stand.

b Another way of thinking is to think of the diameter of a specific tree, say tree $i$. We may also think that the diameter of tree $i$, $X_i$, is a random variable. In this way of thinking, we assume that there is an underlying model or stochastic process that generates the diameter for tree $i$, and the realized diameter of that tree, $x_i$, is just an outcome of that process. Then we may further assume that the process is the same for all trees of the stand or plot. However, because of some unexplained, random factors, the same process led to different values for different trees.

c Still another way of thinking arises if we assume that the diameter of a tree, or more specifically, the cross section at the breast height, is a fixed set or area. Then we may denote the measurement of the diameter of that set for tree $i$ by random variable $X_i$. An observation of this random variable, $x_i$, is obtained by callipering
the tree at the breast height. The randomness of $X_i$ may be related, for example, due to that the area is not completely circular, so the measurements taken in different directions vary. Furthermore, there may be some other factors, such as the determination of breast height and accuracy of the measurement equipment that contribute to the randomness of $X_i$.

There are two types of random variables: discrete and continuous. For a discrete random variable, all possible values can be enumerated. However, the number of possible values can be infinite. A continuous random variable can get any values between given minimum and maximum values. The minimum and maximum can also be infinite.

Example 1.2. In the large forest stand, the tree diameter is a continuous random variable, getting any values between $[0, \infty)$. This interpretation uses the b way of thinking in example 1.1.

Example 1.3. Let us define random variable $Y$ as tree species. This random variable is discrete, as it can get only values $y \in \{1, 2, 3, 4\}$, where the numbers correspond to “Pine”, “Spruce”, “Birch” and “Other”. Another example of a discrete random variable is $Z$, the number of trees on a sample plot, which can get only integer values. In this case, all possible values can be enumerated, even though the number of possible values is infinite.

1.2 Distribution of a random variable

1.2.1 Univariate distribution

The probability of a random variable for getting a specific value is described by the distribution of a random variable. The distribution of a random variable, also called the cumulative distribution function (c.d.f) expresses the probability that the random variable $X$ gets a value that is smaller than or equal to $x$:

$$F(x) = P(X \leq x) \text{ for all } x$$

Example 1.4. Maybe the most important and well-known probability distribution function is the standard normal distribution. Unfortunately, the c.d.f. of this distribution cannot be written in a closed form. However, numerical evaluation is possible, e.g. using R function `pnorm`. Figure 1.1 shows the cdf of the well-known standard normal distribution. The plot was produced using

```r
> x <- seq(-4, 4, 0.01)
> y <- pnorm(x)
> plot(x, y, type="l", xlab="x", ylab="Cumulative density")
```
1.2. DISTRIBUTION OF A RANDOM VARIABLE

The normal distribution is widely used because the central limit theorem states that the distribution of sample mean approaches the normal distribution as the sample size increases. In addition, the distribution is analytically tractable in many computations (even though the c.d.f. is not!)

Apparently, the distribution function of a discrete variable has jumps at the possible values of the random variable, and it is constant elsewhere. In contrast, the distribution function of a continuous random variable is continuous.

Example 1.5. In the large forest stand, we may be interested in the distribution of species, and diameter of individual trees. In addition, we are interested about the number of trees on a sample plot of size 0.01 ha. The left panel of figure 1.2 shows examples on the distribution functions for tree species, tree diameter, and the number of trees in a large forest stand.

Example 1.6. A commonly used distribution function with tree diameter data is the Weibull($\alpha, \beta$) distribution. The two-parameter Weibull distribution function is (also a three parameter version is used by foresters)

$$F(x|\alpha, \beta) = 1 - \exp \left\{ -\left( \frac{x}{\beta} \right)^\alpha \right\},$$

where $\alpha > 0$ and $\beta > 0$ are the shape and scale parameters, respectively. The middle graph of Figure 1.2 shows the c.d.f. Weibull(5,15) distribution.

The R-code for Figure 1.2
Figure 1.2: The cumulative distribution function of tree species \((Y)\), tree diameter \((X)\), and the number of trees on a sample plot \((Z)\) (left), and the corresponding probability mass functions for \(Y\) and \(Z\) and probability density function for \(X\).
1.2. DISTRIBUTION OF A RANDOM VARIABLE

For a function $F(x)$ to be a cdf, the following three conditions need to hold (Casella and Berger 2002).

1. \[ \lim_{x \to -\infty} F(x) = 0 \] and \[ \lim_{x \to \infty} F(x) = 1. \]

2. $F(x)$ is a nondecreasing function of $x$.

3. $F(x)$ is right continuous, i.e., for any $x_0$, \[ \lim_{x \to x_0^+} F(x) = F(x_0). \]

Vice versa, any function fulfilling the above conditions is a proper distribution function. Thus, the function may have jumps, as the distribution functions of $Y$ and $Z$ in figure 1.2 have. The function may also be a mixture of continuous pieces and jumps. An unusual and well-known example of a continuous cdf is the normal cdf, which is the
CHAPTER 1. RANDOM VARIABLES

Figure 1.3: Illustration of the percentile-based distribution of Example 1.7.

For a continuous random variable, the cdf does not need to be differentiable, and it may be defined by parts.

Example 1.7. The percentile-based diameter distribution is defined by the diameters that correspond certain, pre-specified values of the cumulative distribution function. These values are called percentiles. For example, we may specify that the 0th, 25th 50th 75th and 100th diameter percentiles are 5, 10, 13, 17, and 24 cm. This is equivalent to the statement that the 0th, 0.25th, 0.50th, 0.75th and 1st quantiles of the distribution are 5, 10, 13, 17, and 24 cm. A common notation for this is $\xi_0 = 5, \xi_{0.25} = 10, \ldots$ Furthermore assume that the intermediate values are obtained through interpolating between these percentiles.

These assumptions imply that it is assumed that the cdf satisfies $F(5) = 0, F(10) = 0.25, F(13) = 0.5, F(17) = 0.75, F(24) = 1$, and is linear between these. Thus, the cdf is defined by parts:

$$F(x) = \begin{cases} 
0 & x < 5 \\
-0.25 + 0.050x & 5 \leq x < 10 \\
-0.58 + 0.083x & 10 \leq x < 13 \\
-0.31 + 0.063x & 13 \leq x < 17 \\
0.14 + 0.036x & 17 \leq x < 24 \\
1 & x \geq 24 
\end{cases}$$

The left plot of Figure 1.3 shows that the function is increasing, right continuous, and has a limit of 1 as $x$ goes to infinity. Thus, the function is a proper distribution function.

**The R-code for Figure 1.3**

```r
> windows(width=6,height=2.5)
> par(mfcol=c(1,2),
+   mai=c(0.6,0.5,0.1,0.1),
+   mgp=c(2,0.7,0),
+   cex=0.8)
> d<-c(5,10,13,17,24)
```
1.2. DISTRIBUTION OF A RANDOM VARIABLE

> p<-c(0,0.25,0.5,0.75,1)
> a<-(p[-1]-p[-5])/(d[-1]-d[-5])
> b<-p[-5]-a*d[-5]
> plot(c(0,d,30),
+ c(0,p,1),
+ type="l",
+ ylab="Cumulative density",
+ xlab="Diameter, cm")
> plot(c(0,rep(d,each=2),30),
+ c(0,0,rep(a,each=2),0,0),
+ type="l",
+ ylab="Density",
+ xlab="Diameter, cm")

The probability for a random variable to get a value between specified limits $[a, b]$ can be computed by subtraction

$$P(a \leq x \leq b) = F(b) - F(a). \tag{1.1}$$

**Example 1.8.** Let random variable $X$ have the normal distribution with mean 8 and standard deviation of 2. The probability $P(X \leq 10)$ is computed as

> pnorm(10,8,2)
[1] 0.8413447

and the probability $P(9 < X \leq 10)$ as

> pnorm(10,8,2)-pnorm(9,8,2)
[1] 0.1498823

**Example 1.9.** In the previous example, the probability of having the tree diameter between 20 and 10 cm is

$$P(10 \leq x \leq 20) = F(20) - F(10)
= 0.14 + 0.036 \times 20 - (-0.58 + 0.083 \times 10)
= 0.61$$

With the interpretations a and b of example 1.1, this would mean that the probability of having the diameter of a randomly chosen tree between 10 and 20 cm is 0.61. Another interpretation is, that 61% of the trees have the diameter between the given values.

The example was computed using

> F1020
[1] 0.6071429

The cumulative distribution function of $X$ expresses the probability that the random variable gets the value of $x$ or less. However, we are often interested in the probability that $X$ would get exactly a certain value $x$. For discrete random variables, this probability is given by the probability mass function. It is defined only for a discrete random variable as

$$f(x) = P(X = x) \text{ for all } x$$
Thus, the pmf gives the point probabilities for the possible values of the discrete random variable. The relationship between pmf and cdf is simply $P(X \leq x) = \sum_{k<x} f(k) = F(x)$.

For continuous distributions, a similar relationship holds, except for replacement of sum operator by integral:

$$P(X \leq x) = F(x) = \int_{-\infty}^{x} f(t)\,dt.$$  \hspace{1cm} (1.2)

Stating it in different way, the probability density function (pdf) of continuous random variable $x$ is defined as the first derivative of $F(x)$

$$f(x) = \frac{d}{dx} F(x).$$

An important distinction appears between pdf and pmf. The value of pmf for $x$ gives the probability for random variable $X$ to get value $x$. With continuous distribution, probability of any distinct value is always 0. For example, the probability of getting a tree with diameter of exactly 20 cm is always zero. However we have a positive probability of getting a tree with diameter between 19.95 and 20.05 cm, which would be classified to 20 cm using one millimeter classification in callipering. To compute such probability, we would need to integrate the density over the one-millimeter diameter class. This leads to the use of equation (1.1).

**Example 1.10.** The number of trees on a sample plot may be distributed according to $\text{Poisson}(\lambda)$ distribution with pmf

$$P(X = x|\lambda) = \frac{e^{-\lambda} \lambda^x}{x!}$$

The lower graph of figure 1.2 shows example of the $\text{Poisson}(7)$ distribution. Examples on computations using that distribution are $P(X = 3)$

\begin{verbatim}
> dpois(3,7) [1] 0.05212925
\end{verbatim}

and $P(X = 3|X = 4)$

\begin{verbatim}
> dpois(3,7)+dpois(4,7) [1] 0.1433554
> ppois(4,7)-ppois(2,7) [1] 0.1433554
\end{verbatim}

**Example 1.11.** The pdf and pmf of example 1.5 are shown in the right panel of figure 1.2.

**Example 1.12.** The density of percentile-based diameter distribution of example 1.7


1.2. DISTRIBUTION OF A RANDOM VARIABLE

is obtained by differentiating the cdf of example 1.7, which gives

\[
f(x) = \begin{cases} 
0 & x < 5 \\
0.050 & 5 \leq x < 10 \\
0.083 & 10 \leq x < 13 \\
0.063 & 13 \leq x < 17 \\
0.036 & 17 \leq x < 24 \\
0 & x \geq 24 .
\end{cases}
\]

Graph of the obtained density is shown in the right plot of Figure 1.3.

Example 1.13. The Weibull density is obtained by differentiating the c.d.f. of Example 1.6 with respect to \( x \)

\[
f(x|\alpha, \beta) = F'(x|\alpha, \beta)
\]

\[
= - \exp \left\{ - \left( \frac{x}{\beta} \right) ^\alpha \right\} \left\{ -\alpha \left( \frac{x}{\beta} \right) ^{\alpha - 1} \frac{1}{\beta} \right\}
\]

\[
= \frac{\alpha}{\beta^\alpha} x^{\alpha - 1} e^{-(x/\beta)^{\alpha}},
\]

The distribution and density functions of the Weibull-distribution are shown in Figure 1.4. Figure 1.4 demonstrates the Weibull distribution with parameters \( \alpha = 2.98 \) and \( \beta = 15.26 \), i.e. the distribution Weibull(2.98,15.26). With these values of the parameters, The weibull distribution is very similar to the percentile- based distribution of Example 1.7.

The R-code for Example 1.13

```r
plot(c(0,d,30),c(0,p,1),type="l",ylab="Cumulative density",xlab="Diameter, cm")
x<-seq(0,30,0.1)
lines(x,pweibull(x,2.98,15.26))
points(dlim,Fvalp,cex=0.5) # add the points into the plot
points(dlim,Fvalw,cex=0.5) # add the points into the plot
plot(c(0,rep(d,each=2),30),c(0,0,rep(a,each=2),0,0),type="l",ylab="Density",xlab="Diameter, cm")
lines(x,dweibull(x,2.98,15.26))
```

Example 1.14. The diameter distribution can also be specified through a discrete distribution. For example, we may classify the diameters into 2 cm classes. Then each diameter class has a positive probability. This way we get a table, which is called by foresters as the stand table, and by statisticians as the multinomial distribution. The table includes class mean diameters, the relative frequencies (or probabilities) of the class, and the total number of stems in each of the classes.

The code below computes the stand table using the percentile-based distribution defined in example 1.7 and Weibull-distribution defined in example 1.13. We note that the diameter class frequencies from the Weibull distribution do not sum to one, but to 0.999. This is because the Weibull distribution has infinite maximum value for \( x \), and the value of the cdf at the used maximum diameter (30 cm) is 0.999.
> # The stand table example
> # # The multinomial distribution
> # Linear interpolation.
> # Assuming given values of X and Y, the function
> # returns the interpolated values at points x and the first derivative.
> # In the case of percentile-based distribution, these are the distribution function and density.
> interpolate<-function(x,X,Y) {
+ n<-length(X) # the number of points, the number of intervals is n-1
+ iv<-1:n 
+ # define the interval at which each of x values is located
+ i<-as.numeric(sapply(x,function(x) max(c(1,iv[x>=X][1]))))
+ i[X[n]]<-(n-1) # extrapolation at the upper end
+ i[X[1]]<-1 # extrapolation at the lower end
+ # compute the n-1 long vector of slope coefficients
+ slope<-(Y[iv+1]-Y[iv])/((X[iv+1]-X[iv]))
+ # compute the value for each value in x by picking the correct constant and slope
+ y<-Y[i]+slope[i]*(x-X[i])
+ list(y=y,dy=slope[i])
+ }
>
> N<-800 # stand density is 800 trees per ha
> dlim<-seq(0,30,by=2) # the diameter class limits
> Fvalp<-interpolate(dlim,c(0,d,100),c(0,p,1))$y # cdf at class limits for pp
> FvalW<-pweibull(dlim,2.98,15.26) # ... and Weibull
> stable<-cbind(DBH=dlim[-1]-1,pp=Fvalp[-1]-Fvalp[-length(Fvalp)])
> stable<-cbind(stable,Np=N*stable[,2])
> stable<-cbind(stable,NW=N*stable[,4])
> stable<-rbind(stable,c(NA,sum(stable[,2]),sum(stable[,3]), sum(stable[,4]),sum(stable[,5])))
> print(stable)

<table>
<thead>
<tr>
<th>DBH</th>
<th>pp</th>
<th>Np</th>
<th>PW</th>
<th>NW</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.00000000</td>
<td>0.00000000</td>
<td>0.002341897</td>
<td>1.873518</td>
</tr>
<tr>
<td>[2,]</td>
<td>3.00000000</td>
<td>0.005986963</td>
<td>0.015986963</td>
<td>12.789571</td>
</tr>
<tr>
<td>[3,]</td>
<td>5.005000000</td>
<td>0.041722075</td>
<td>0.041722075</td>
<td>33.776660</td>
</tr>
<tr>
<td>[4,]</td>
<td>7.001000000</td>
<td>0.075751508</td>
<td>0.075751508</td>
<td>60.61206</td>
</tr>
<tr>
<td>[5,]</td>
<td>9.010000000</td>
<td>0.112779333</td>
<td>0.112779333</td>
<td>89.022366</td>
</tr>
<tr>
<td>[6,]</td>
<td>11.016666666</td>
<td>0.139444713</td>
<td>0.139444713</td>
<td>111.555770</td>
</tr>
<tr>
<td>[7,]</td>
<td>13.014583333</td>
<td>0.152867212</td>
<td>0.152867212</td>
<td>121.669377</td>
</tr>
<tr>
<td>[8,]</td>
<td>15.012500000</td>
<td>0.145243705</td>
<td>0.145243705</td>
<td>116.194964</td>
</tr>
<tr>
<td>[9,]</td>
<td>17.009821429</td>
<td>0.121340201</td>
<td>0.121340201</td>
<td>97.072161</td>
</tr>
<tr>
<td>[10,]</td>
<td>19.007142857</td>
<td>0.088257786</td>
<td>0.088257786</td>
<td>70.614288</td>
</tr>
<tr>
<td>[11,]</td>
<td>21.007142857</td>
<td>0.055484120</td>
<td>0.055484120</td>
<td>44.387296</td>
</tr>
<tr>
<td>[12,]</td>
<td>23.007142857</td>
<td>0.029897146</td>
<td>0.029897146</td>
<td>23.917717</td>
</tr>
<tr>
<td>[13,]</td>
<td>25.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>10.941445</td>
</tr>
<tr>
<td>[14,]</td>
<td>27.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>4.206499</td>
</tr>
<tr>
<td>[15,]</td>
<td>29.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>1.344797</td>
</tr>
<tr>
<td>[16,]</td>
<td>31.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.999444692</td>
</tr>
</tbody>
</table>
The following conditions for density (or pmf) \( f(x) \) can be deduced from the conditions of a cdf.

1. \( f(x) \geq 0 \) for all \( x \)
2. \( \sum_x f(x) = 1 \) (pmf) or \( \int_{-\infty}^{\infty} f(x) \, dx = 1 \) (pdf)

**Example 1.15. The distribution of laser hits on a tree crown.** The radius of a tree crown between the height of maximal crown radius and total height \( h \) can be expressed by an ellipsoid centered at \((x_0h, y_0h)\), \( x_0 < 1, y_0 < 0 \), and having half-axes \( ah \) and \( bh \).

Having an airborne laser scanning (ALS) in mind, we can assume that the observable crown radius remains constant after that, even though the actual radius decreases. The function based on these assumptions is defined by parts as

\[
Y(z) = \begin{cases} 
  h \left( y_0 + b \right) & z \leq x_0h \\
  h \left( y_0 + b \sqrt{1 - \left( \frac{z - x_0h}{a^2} \right)^2} \right) & x_0h < z \leq h \\
  0 & z > h
\end{cases}
\]  

(1.6)

where \( a = \sqrt{\frac{b^2(1-x_0)^2}{b^2-y_0^2}} \) is defined so that \( Y(h, h) \) passes through point \((h, 0)\), the tree top. The assumed function is illustrated in figure 1.5 using parameter values \( h = 20, x_0 = 0.2, y_0 = -0.1 \), and \( b = 0.25 \).

Airborne laser scanner could be used to produce observations on the height of canopy surface at given points. Let random variable \( Z \) describe the height of tree crown at random location within a single tree crown. The distribution of \( Z \) would be the proportion of crown area below \( z \) of the total area:

\[
P(Z < z) = \frac{A_{\text{max}} - A(z)}{A_{\text{max}}} = 1 - \frac{A(z)}{A_{\text{max}}}.
\]

With the ellipsoid tree crown, the maximum area is obtained at height \( x_0 \), \( A_{\text{max}} = \pi h^2(y_0 + b)^2 \). The crown area at any height between \( x_0 \) and \( h \) is \( A(z) = \pi Y(z)^2 \). The distribution of \( Z \) becomes

\[
F_Z(z) = 1 - \frac{y_0^2 + 2y_0b\sqrt{1 - \left( \frac{z - x_0h}{a^2} \right)^2} + b^2 \left( 1 - \left( \frac{z - x_0}{a^2} \right)^2 \right)}{(y_0 + b)^2}
\]

The density is obtained by differentiating with respect to \( z \)

\[
f_Z(z) = \frac{1}{(y_0 + b)^2} \left[ \frac{2b^2}{a^2h} \left( \frac{z}{h} - x_0 \right) + y_0b \left( 1 - \left( \frac{z - x_0h}{a^2} \right) \right)^{-1/2} \frac{2}{a^2h} \left( \frac{z}{h} - x_0 \right) \right]
\]

(1.7)
An interesting result is obtained when $y_0 = 0$, i.e., when the center of ellipsoid is at the x-axis:

$$f_Z(z) = \frac{2}{a^2} \left( \frac{z}{h} - x_0 \right),$$

which is the density of the triangular distribution illustrated in the lower right plot of Figure 1.5.

The R-code for Example 1.15

```r
> # Ellipsoidal tree crown
> radius.ellipse<-function(x,b,x0=0,y0=0,h) {
+ x<-x
+ a<-sqrt(b^2 * (1-x0)^2/(b^2-y0^2))
+ r<-rep(h *y0+b, length(x))
+ r[x=x0+h&x<=h]<-h *y0+b*sqrt(1-(x[x=x0+h&x<=h]/h-x0)^2/a^2))
+ r[x=h]<-0
+ r
+ }
> 
> # cdf of laser observations
> cdf.laser<-function(x,b,x0,y0,h) {
+ a<-sqrt(b^2 * (1-x0)^2/(b^2-y0^2))
+ value<-rep(0,length(x))
+ value[x>=x0 *h&x<h]<-1-
+ y0^2+2*y0*b*sqrt(1-(x[x>=x0*h&x<h]/h-x0)^2/a^2)+
+ b^2*(1-(x[x=x0+h&x<h]/h-x0)^2/a^2))/(y0+b)^2
+ value[x=h]<-1
+ value
+ }
> 
> # corresponding density
> pdf.laser<-function(x,b,x0,y0,h) {
+ a<-sqrt(b^2 * (1-x0)^2/(b^2-y0^2))
+ value<-rep(0,length(x))
+ value[x=x0+h&x<h]<-1/(y0+b)^2*
+ y0*b*(1-(x[x=x0+h&x<h]/h-x0)^2/a^2)^(-1/2)*
+ (-2/(a^2*h)*(x[x=x0+h&x<h]/h-x0))+
+ b^2*(-2/(a^2*h)+x[x=x0+h&x<h]/h-x0))
+ value
+ }
> x0<-0.2
> y0<-0.1
> b<-0.25
> h<-20
> x<-seq(0,22,0.01)
> r<-radius.ellipse(x,b,x0,y0,h)
> windows(width=6, height=5)
> par(mfcol=c(2,2), mai=c(0.6,0.5,0.1,0.1), mgp=c(2,0.7,0), cex=0.8)
> plot(x, r, type="l",
+ xlab="Height, m", ylab="Crown radius, m", ylim=h*c(-0.1,0.15))
> points(h*x0,h*y0)
> text(h*x0, h*y0, expression((hx[0] *paste(",")*hy[0])))
> lines(h*rep(x0,2), h*c(y0,y0+b), lty="dashed")
> points(h*rep(x0,2), h*c(y0,y0+b), pch="."*
> text(h*x0, h+y0+b/2, "hb", pos=4)
> lines(c(0,h), c(0,0), lty="dotted")
> points(c(0,h), c(0,0), pch="dotted")
> text(h/2, 0, "h", pos=3)
> 
> plot(x, cdf.laser(x,b,x0,y0,h), type="l",
+ xlab="Height, m", ylab="Cumulative density")
> plot(x, pdf.laser(x,b,x0,y0,h), type="l",
+ xlab="Height, m", ylab="Density")
> plot(x, pdf.laser(x,b,x0,0,h), type="l",
+ xlab="Height, m", ylab="Density")
Figure 1.5: Illustration for example 1.15 using parameter values $h = 20$, $x_0 = 0.2$, $y_0 = -0.1$, and $b = 0.25$. The upper left figure shows the crown radius as a function of height. The cdf of laser observations is shown in the lower left figure, and the corresponding density in the upper right figure. The lower right figure illustrates the triangular density, obtained using $y_0 = 0$. 
1.2.2 Making transformations to random variables

In many cases, we do transformations to random variables. For example, tree volumes may be transformed to log scale before fitting a volume model. It is important to understand the effect of transformation into the distribution of a random variable. The usually used transformations are monotonic one-to-one transformations, which mean that a single value of the original variable \( X \) always corresponds to only one value of \( Y \). This means that the transformation function is always monotonic, either increasing or decreasing. Let \( X \) have cdf \( F_X(x) \) and let us define random variable \( Y \) as \( g(X) \). The cdf of \( Y \) is

\[
F_Y(y) = \begin{cases} 
F_X(g^{-1}(y)) & \text{if } g(x) \text{ is increasing} \\
1 - F_X(g^{-1}(y)) & \text{if } g(x) \text{ is decreasing}
\end{cases} \tag{1.8}
\]

**Example 1.16.** Let \( X \) be tree diameter, having cdf according to the Weibull distribution function

\[
F(x|\alpha, \beta) = 1 - \exp \left\{ - \left( \frac{x}{\beta} \right)^\alpha \right\},
\]

with values \( \alpha = 5 \) and \( \beta = 15 \) for the shape and scale parameters, respectively. Assume that tree height \( Y \) depends on tree diameter according to the power function

\[
Y = g(X) = aX^b,
\]

where the parameters are \( a = 8 \) and \( b = 0.3 \). With these values for parameters, the assumed H-D curve is an increasing function of tree diameter. For the distribution of tree heights, we need the inverse of the H-D curve,

\[
g^{-1}(y) = \left( \frac{y}{a} \right)^{1/b}.
\]

The distribution of tree heights results from writing the inverse transformation into the c.d.f. of diameter. We get

\[
F(y|\alpha, \beta, a, b) = 1 - \exp \left\{ - \left( \frac{y}{\beta} \right)^\alpha \right\} \tag{1.9}
\]

\[
= 1 - \exp \left\{ - \left( \frac{y}{a\beta} \right)^{a/b} \right\}
\]

\[
= F(y/a\beta, a\beta^b), \tag{1.10}
\]

which shows that, assuming a Weibull distribution for tree height and a power equation as the H-D curve, the tree height is also distributed according to Weibull distribution with shape and scale parameters \( \alpha/b \) and \( a\beta^b \). Note that this is not a general results, but it holds only for these specific functions. In general, there is no rule that the functional form of the distributions of tree height and diameter would be the same.
Another way of deriving the distribution of a transformation random variables with a continuous density can be stated through pdf. If $X$ has pdf $f_X(x)$ and $g(X)$ is a monotone transformation, with $g^{-1}(y)$ having a continuous derivative, then the pdf of $y$ in the support of $Y$ is

$$f_Y(y) = f_X(g^{-1}(y)) \left| \frac{d}{dy} g^{-1}(y) \right|,$$

and 0 elsewhere. This expression is easily obtained from the definition of $F_Y(y)$ by differentiating it with respect to $y$.

**Example 1.17.** In example 1.16, the first derivative of the inverse H-D curve is

$$\frac{d}{dy} g^{-1}(y) = \frac{1}{ab} \left( \frac{y}{a} \right)^{1/b-1}.$$

The density of height can be obtained either through the Weibull distribution using our result in this special case, or using the general formula for the density of a transformation.

Figure 1.6 illustrates the applied transformations and distributions of diameter and height. The following code shows how the distribution of tree heights can be derived using the result of the special case of example 1.16 and the general expression.
Chapter 1. Random Variables

Figure 1.6: Illustration of the example 1.16.
1.2. DISTRIBUTION OF A RANDOM VARIABLE

Example 1.18. Forestry student has taken observations on the thickness of Norway spruce needles. The observations are skewed to the right, and are naturally all positive. Based on these observations, she assumes that the needle thickness is distributed according to the lognormal distribution, which has the pdf

\[
f_X(x) = \frac{1}{\sqrt{2\pi \sigma^2}} e^{-\frac{(\log(x)-\mu)^2}{2\sigma^2}}.
\]

For modeling purposes, she makes the logarithm transformation \( Y = g(X) = \ln(X) \) to the observations. The inverse function of the applied transformation \( g^{-1}(y) = e^y \) is continuous and increasing, and has derivative \( \frac{d}{dy}g^{-1}(y) = e^y \). The distribution of logarithmic observations is

\[
f_Y(y) = \frac{1}{\sqrt{2\pi \sigma^2}} e^{\frac{-(x\mu)^2}{2\sigma^2}} e^y = \frac{1}{\sqrt{2\pi \sigma^2}} e^{\frac{-(x\mu)^2}{2\sigma^2}},
\]

which is the density of normal distribution with mean \( \mu \) and variance \( \sigma^2 \).

1.2.3 Quantile function

In addition to cdf and pdf (or pmf), one function describing the distribution is the quantile function, which is defined as the inverse of the cdf

\[Q(p) = F^{-1}(p)\]

where \( F(x) \) is the distribution function of random variable \( X \). This function gives the \( p \)th quantile of distribution \( F \), that is \( Q(p) = \xi_p \). This function is useful, for example in random number generation, or in computing percentiles or the median of a distribution. It may not be possible to solve the quantile function analytically, but a numerical solution is always possible by using e.g., a simple up-and-down algorithm.

Assume that we want to generate observations of random variable \( X \), which has continuous cdf \( F(x) \). The random variable \( Y = F(x) \) (so called probability integral transformation) is uniformly distributed between 0 and 1. If we have observations of random variable \( U \), which has uniform distribution between 0 and 1, then a random sample from the desired distribution is obtained through transformation \( q_X(U) \). Thus, a random sample from any (univariate) distribution function can be easily generated if we are able to generate a sample from an uniform distribution and we know the quantile function.
Example 1.19. The quantile function of the percentile-based diameter distribution is defined by parts as

\[ Q(u) = \begin{cases} 
5 + 20u & 5 \leq u < 10 \\
7 + 12u & 10 \leq u < 13 \\
5 + 16u & 13 \leq u < 17 \\
-4 + 28u & 17 \leq u < 24 
\end{cases} \]

The graph of this function is shown in the top left corner of Figure 1.7. Sets of 1000, 10000 and 100000 uniformly distributed random numbers were generated by using function `runif()`. Using the above-defined quantile function, they were transformed to samples from percentile-based diameter distribution. Figure 1.7 shows the histograms of the obtained samples from the percentile-based distribution.

**The R-code for Example 1.19**

```
> windows(width=6,height=6)
> par(mfcol=c(2,2),mai=c(0.6,0.5,0.1,0.1),mgp=c(2,0.7,0),cex=0.8)
> d<-c(5,10,13,17,24)
> p<-c(0,0.25,0.5,0.75,1)
> plot(c(0,p,1),c(0,d,30),type="l",ylab="Quantile function",xlab="Probability")
```
1.2. DISTRIBUTION OF A RANDOM VARIABLE

> interpolate<-function(x,X,Y) {
+ n<-length(X) # the number of points, the number of intervals is n-1
+ iv<-1:n
+ # define the interval at which each of x values is located
+ i<-sapply(x,function(x) max(c(1,iv[x>=X])))
+ i[x>=X[n]]<-n-1 # extrapolation at the upper end
+ i[x<X[1]]<-1 # extrapolation at the lower end
+ # compute the n-1 long vector of slope coefficients
+ slope<-(Y[iv+1]-Y[iv])/(X[iv+1]-X[iv])
+ slope<-slope[-length(slope)]
+ # compute the value for each value in x by picking the correct constant and slope
+ y<-Y[i]+slope[i]*(x-X[i])
+ list(y=y,dy=slope[i],const=Y[-length(slope)]-slope*X[-length(slope)],slope=slope)
+ }

> interpolate(runif(1,0,1),p,d)$const
[1] 5 7 5 -4
> interpolate(runif(1,0,1),p,d)$slope
[1] 20 12 16 28

> yvec<-interpolate(runif(1000,0,1),p,d)$y
> hist(yvec,breaks=5:24,freq=FALSE,xlab="Diameter, cm",ylab="Density",main="N=1000")
> yvec<-interpolate(runif(10000,0,1),p,d)$y
> hist(yvec,breaks=5:24,freq=FALSE,xlab="Diameter, cm",ylab="Density",main="N=10000")
> yvec<-interpolate(runif(100000,0,1),p,d)$y
> hist(yvec,breaks=5:24,freq=FALSE,xlab="Diameter, cm",ylab="Density",main="N=100000")

1.2.4 Multivariate distribution

We may also have two or more random variables that are somehow related. For example, we may have observed both diameter and height for a tree, thus resulting our observations to be random vectors of length two. The multiple variables may also have been produced by the same process. For example, $X_1$ and $X_2$ may be random variables “diameter of tree 1” and “diameter of tree 2” from a given stand. The $n$ multivariate random variables are often combined into a $n$-dimensional random vector. For example, a random vector of two random variables may be defined as $X = (X_1, X_2)'$. The correlation and covariance are common characteristics that are related to random vectors rather than on scalar random variables.

Example 1.20. In example 1.3 the observations on tree species and diameter provide observations of a bivariate random vector.

The distribution of a random vector is expressed with a multivariate distribution. The multivariate distribution is the joint distribution of the component variables of the random vector. It completely defines the probabilities of all possible value combinations of the component variables of a random vector. For a bivariate case, let us denote these components by $X_1$ and $X_2$, the random vector being then $X = (X_1, X_2)'$. In this case, the joint distribution gives the probabilities for all possible realizations of pair $X_1$ and $X_2$. The joint distribution can be used to construct conditional and marginal probability distributions of any single component variable of the random vector joint distribution.
The conditional distribution expresses probability of one component variable for given values of the other variables. For example, the distribution of of $X_1|X_2 = x_2$ defines the distribution of $X_1$ given that $X_2$ has got the value of $x_2$. Thus, the conditional distribution is a univariate distribution in this case.

For a random vectors of length 3 or more, a univariate conditional distribution is obtained by conditioning on all other component variables. For example, for random vector $X = (X_1, X_2, \ldots, X_n)$, we may be interested in conditional distribution of $X_1|X_2 = x_2, \ldots, X_n = x_n$. The conditional distribution may also be multivariate. For example, the distribution of $(X_1, X_2)|X_3 = x_3$ is a bivariate distribution.

The marginal distribution of a random variable is the distribution in the whole population, assuming that no information on other random variables is available or used. For example, if we have a bivariate random vector $(X_1, X_2)'$, then the marginal distribution of $X_1$ is the sum (for discrete random variables) or integral (for continuous ones) of $X_1|X_2 = x_2$ over all possible values of $x_2$. In a table of a joint distribution, the marginal distribution is in the lower or right marginals of the table, which gives the name “marginal”. For a discrete random variable, the probability for any distinct value of marginal distribution is the sum of conditional probabilities. For a continuous variable, the sums are replaced with integration.

The marginal distribution can be derived from the joint distribution. However, the joint distribution cannot be derived from marginal distribution without making additional assumptions on the relationship between the component variables $X_1$ and $X_2$.

**Example 1.21.** Assume that trees have been observed for species and health. The species is coded in the classes of example 1.3, and health in classes 1, 2 and 3, indicating dead, weakened, and healthy, respectively. The joint pmf of the two discrete variables can be expressed in tabular form

<table>
<thead>
<tr>
<th>$X$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>$\Sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.01</td>
<td>0.21</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>0.05</td>
<td>0.02</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.3</td>
<td>0.3</td>
<td>0</td>
<td>0.02</td>
<td>0.62</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>0.5</td>
<td>0.3</td>
<td>0.15</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

The conditional distribution of $X|Y = 1$, i.e., the distribution of Scots pine trees into three classes, can obtained by dividing the values of the first column by the marginal probability of Scots pine. The conditional probabilities are $0.1/0.5 = 0.2$, $0.1/0.5 = 0.2$ and $0.3/0.5 = 0.6$ for dead, weakened, and healthy, respectively.

The conditional distribution $Y|X = 1$ is the distribution of dead trees by species. The conditional probabilities, obtained by dividing the values of the first row with the
1.2. DISTRIBUTION OF A RANDOM VARIABLE

row sum of 0.21, are 0.1/0.21 = 0.476, 0/0.21 = 0.000, 0.1/0.21 = 0.476, and 0.01/0.21 = 0.048 for tree species 1, 2, 3 and 4, respectively. Division by 0.21 makes conditional probabilities sum up to 1.

The marginal distributions of tree species are given in last column and row of the table.

The code below demonstrates the use of R with computations of a matrix

```
> jointd<-cbind(c(0.1,0.1,0.3),c(0,0,0.3),c(0.1,0.05,0),c(0.01,0.02,0.02))
> jointd
[1,] 0.1 0.0 0.10 0.01
[2,] 0.1 0.0 0.05 0.02
[3,] 0.3 0.3 0.00 0.02
> sum(jointd)
[1] 1
> condr1<-jointd[1,]/sum(jointd[1,])
> condr1
[1] 0.47619048 0.00000000 0.47619048 0.04761905
# The marginal distribution of tree species
> margr<-jointd[1,]+jointd[2,]+jointd[3,]
> margr
[1] 0.50 0.30 0.15 0.05
# a more general alternative for the same purpose
> apply(jointd,2,sum)
[1] 0.50 0.30 0.15 0.05
# the marginal distribution of health
> margc<-apply(jointd,1,sum)
> margc
[1] 0.21 0.17 0.62
```

Let \( X \) be a bivariate random vector

\[
X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix},
\]

where the component variables are continuous. The joint probability density function expresses the probability for the random vector \( X \) to fall within area \( A, A \in \mathbb{R}^2 \):

\[
P((X_1, X_2)' \in A) = \int_A f(x_1, x_2)dx_1, dx_2.
\]

For example, the probability that \( X_1 \) gets a value between \( a \) and \( b \), and \( X_2 \) gets a value between \( c \) and \( d \) is

\[
P(a < X_1 \leq b, c < X_2 \leq d) = \int_a^b \int_c^d f(x_1, x_2)dx_1dx_2
\]

. The marginal pdf is obtained by integrating out the other component variables. For a bivariate case, this implies that

\[
f_1(x_1) = \int_{\infty}^{\infty} f(x_1, y_1)dx.
\]

Two random variables are independent if all joint probabilities result from multiplication of the marginal probabilities

\[
p(x, y) = p(x)p(y) \quad \text{for discrete } X \tag{1.12}
\]

\[
f_{XY}(x, y) = f_X(x)f_Y(y) \quad \text{for continuous } X. \tag{1.13}
\]
Example 1.22. The products of the marginal probabilities of the previous example are

\[
\begin{align*}
\text{# product of marginal probabilities for 1st row} \\
> \text{margc[1]}*\text{margr[1]} \\
& 0.1050 \ 0.0630 \ 0.0315 \ 0.0105 \\
\text{# 2nd row} \\
> \text{margc[2]}*\text{margr[1]} \\
& 0.0850 \ 0.0510 \ 0.0255 \ 0.0085 \\
\text{# 3rd row} \\
> \text{margc[3]}*\text{margr[1]} \\
& 0.310 \ 0.186 \ 0.093 \ 0.031
\end{align*}
\]

These are all different from the probabilities of the joint distribution. Therefore, the health and tree species are not independent in this example.

An easier way to compute the products of marginal probabilities is to use the outer product of the vectors:

\[
> \text{margc} \% \text{outer} \% \text{margr}
\]

\[
\begin{array}{cccc}
\[1,\] & \[2,\] & \[3,\] & \[4,\] \\
\[1,\] & 0.105 & 0.063 & 0.0315 & 0.0105 \\
\[2,\] & 0.085 & 0.051 & 0.0255 & 0.0085 \\
\[3,\] & 0.310 & 0.186 & 0.093 & 0.031
\end{array}
\]

Example 1.23. Let the joint distribution of tree diameter \(X\) and height \(Y\) be expressed by a bivariate density \(f(x, y)\). The marginal density of diameter is

\[
f_X(x) = \int_{-\infty}^{\infty} f(x, y) \, dy.
\]

The marginal density of height is

\[
f_Y(y) = \int_{-\infty}^{\infty} f(x, y) \, dx.
\]

Example 1.24. A very commonly used multivariate distribution is the bivariate normal distribution. It is a generalization of the bell-shaped normal distribution to two-dimensional random vectors, say \(X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}\). It is specified by the mean vector \(\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\), and the variance-covariance matrix \(\text{var}(X) = \Sigma = \begin{pmatrix} \text{var}(X_1) & \text{cov}(X_1, X_2) \\ \text{cov}(X_2, X_1) & \text{var}(X_2) \end{pmatrix}\).

The upper left graph of Figure 1.8 demonstrates the density of this distribution with parameters \(\mu_1 = 20, \mu_2 = 15, \text{var}(X_1) = 5^2, \text{var}(X_2) = 5^2\), and \(\text{cov}(X_1, X_2) = 0\). The distribution is a circular bell-shaped function. The top right graph shows another bivariate normal distribution with the same means as distribution 1 but different variances: \(\text{var}(X_1) = 7^2, \text{var}(X_2) = 3^2\). Now the distribution is a “horizontally stretched” bell. The second row shows the distribution with the same means and variances as distribution 2, but the distribution 3 has a positive covariance of \(\text{cov}(X_1, X_2) = 3^2\) which implies correlation coefficient of 0.76. The distribution 4 has a similar, negative correlation. We see that correlation causes “stretching” the bell to other than horizontal or vertical direction. However, an important property of the multinormal distribution is that the correlation is always linear, which practically means that the distribution is

1.2. DISTRIBUTION OF A RANDOM VARIABLE

Figure 1.8: Figure for example 1.24 UPDATE AXIS LABELS.
always circular or a “stretched” circle or ellipsoid, but cannot be a “bended”, stretched circle.

The third row demonstrates the shape of the conditional distribution of $X_2$ in the case of distribution 4. The left plot shows the profile of the distribution of $X_2$ for three different values of $X_1$ ($X_1=10, 20$ and $30$; see the vertical lines in the plot of joint distribution). The right plot shows the profile of the distribution of $X_1$ for three values of $X_2$ ($X_2=10,15$ and $20$). We notice that all the graphs have the same bell-shaped profile than the univariate normal distribution. In addition, they are of the same width. This demonstrates the second important feature of multivariate normal distribution: all conditional distributions are univariate normal with equal variances. Note that conditional p.d.f’s are obtained from the profiles by scaling them so that the area under the curve would be one.

The lowest plots demonstrate the profile of marginal distribution. They are obtained by summing the joint density at regular intervals over the values of $X_1$ (left) to demonstrate the profile of the marginal distribution of $X_1$. The right plot shows a similar analysis to demonstrate the marginal distribution of $X_2$. Also the marginal distributions have a bell-shaped profile, which demonstrates a third important property of the bivariate normal distribution: the marginal distributions are univariate normal. Also in this case, scaling the marginal profiles to have area of 1 would give the marginal densities.

The R-code for Example 1.24

```r
library(mnormt)

x0<-seq(0,40,0.4)
y0<-seq(0,30,0.4)
# define a 0.4 by 0.4 grid for x=0,...,40 and y=0,...,30
x<-rep(x0,each=length(y0))
y<-rep(y0,length(x0))
xymat<-cbind(x,y)

# Define the parameters of teh distributions
mu<-c(20,15)
Sigma1<-matrix(c(5^2,0,0,5^2),ncol=2)
Sigma2<-matrix(c(7^2,0,0,3^2),ncol=2)
Sigma3<-matrix(c(7^2,4^2,4^2,3^2),ncol=2)
Sigma4<-matrix(c(7^2,-4^2,-4^2,3^2),ncol=2)

# compute bivariate normal densities
f1<-dmnorm(xymat, mean = mu , varcov=Sigma1)
f2<-dmnorm(xymat, mean = mu , varcov=Sigma2)
f3<-dmnorm(xymat, mean = mu , varcov=Sigma3)
f4<-dmnorm(xymat, mean = mu , varcov=Sigma4)

windows(8,14)
par(mfrow=c(4,2),mai=c(0.4,0.4,0.2,0.05),cex=0.7,mgp=c(1.5,0.8,0))

# Plot the grid grayescaled colors defined by the bivariarte normal densities
plot(xymat,col=gray((max(f1)-f1)/max(f1)),pch=15,cex=0.7,main="Joint distribution 1")
plot(xymat,col=gray((max(f2)-f2)/max(f2)),pch=15,cex=0.7,main="Joint distribution 2")
plot(xymat,col=gray((max(f3)-f3)/max(f3)),pch=15,cex=0.7,main="Joint distribution 3")
plot(xymat,col=gray((max(f4)-f4)/max(f4)),pch=15,cex=0.7,main="Joint distribution 4")
```
1.2. DISTRIBUTION OF A RANDOM VARIABLE

The joint distribution generalizes also for longer random vectors. Let \( X \) be a random vector

\[
X = \begin{pmatrix}
X_1 \\
X_2 \\
\vdots \\
X_n
\end{pmatrix}
\]

The marginal density of the \( k \) the component variable is

\[
f(x_k) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x_1, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots, x_n) \, dx_1 \cdots dx_{k-1} \cdots dx_{k+1} \cdots dx_1. \tag{1.14}
\]

This is a function of only \( x_k \), i.e., it is a density of one-dimensional random variable. For such a distribution, all results of section 1.2.1 apply.

From the conditions set for a univariate pdf, it can be generalized that any function for which \( f(x, y) \geq 0 \) for all \((x, y) \in \mathbb{R}^2\), and which satisfies

\[
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \, dx \, dy = 1
\]

is a bivariate joint density function for some random vector \((X, Y)\).
The definition of joint distribution function is analogous to the univariate case

\[ F(X) = P(X \leq x) = P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_n \leq x_n) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} \cdots \int_{-\infty}^{x_n} f(u_1, u_2, \ldots, u_n) \, du_n \cdots du_2 du_1 = \int_{-\infty}^{x} f(u) \, du. \]

The marginal pdf of the \( k \)th component is obtained using (1.14) in (1.2). Thus, we just integrate the marginal density from \(-\infty\) to \( x_k \). But this integral is same as the conditional distribution function at \( x = (\infty, \ldots, \infty, x_k, \infty, \ldots, \infty) \). Thus, the marginal cdf of the \( k \)th component variable is

\[ F_{X_k}(x_k) = F_{\infty, \ldots, \infty, x_k, \infty, \ldots, \infty}. \]

### 1.3 Expectation, variance and covariance

#### 1.3.1 Expectation

In many cases, the distribution of a random variable is unknown, and we don’t want to make strong assumptions on it. However, we may want to make inference about the behavior of the random variable. In such case, we can summarize the most interesting and important characteristics to some summary figures, that describe the most important properties of the random variable we are interested in. However, these summary characteristics do not specify the whole distribution of the random variable.

The most commonly used characteris for describing the properties of a random variable is the expected value. The expected value is merely the mean of the random variable, which describes the average of the distribution, or the center of gravity. The expected value is used in the hope that it would as well as possible summarize the typical or expected value of an observation drawn from the underlying distribution.

The expected value of random variable \( X \) is the weighted mean of the random variable, the weight being defined by the pmf or pdf:

\[ E(X) = \begin{cases} \sum x_j p_j & \text{for discrete } X \\ \int_{-\infty}^{\infty} x f_X(x) \, dx & \text{for continuous } X \end{cases} \]

(1.15)

It is also possible that the expected value is infinity, if the tails of the distribution are thick enough (e.g. the t- distribution).
1.3. EXPECTATION, VARIANCE AND COVARIANCE

Example 1.25. The mean diameter in the stand of example 1.12 is

\[
E(x) = \int_{-\infty}^{\infty} x f_X(x) \, dx
\]

\[
= \int_{5}^{10} 0.05 x \, dx + \int_{17}^{24} 0.036 x \, dx
\]

\[
= \frac{1}{2} \left[ 0.05(10^2 - 5^2) + \ldots + 0.036(24^2 - 17^2) \right]
\]

= 13.625

It was computed using

```r
> # The mean diameter
> mu<-sum(a/2*(d[-1]^2-d[-5]^2))
> mu
```

[1] 13.625

Example 1.26. The expected value of the normal distribution is the parameter \( \mu \). We could compute it analytically from the equation

\[
\int_{-\infty}^{\infty} x \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \, dx
\]

to see that the result of the integral is \( \mu \).

Sometimes a quick solution is the use of numerical methods. In R, function `integrate` evaluates the integral of a specified function over a given interval. The code below evaluates numerically the expected value of the \( N(10, 2) \) distribution.

```r
> fun<-function(x) {
+ x*dnorm(x,10,2)
+ }
> integrate(fun,-Inf,Inf)
```

10 with absolute error < 0.0011

Example 1.27. The code below evaluates numerically the expected value of the Weibull distribution of example 1.13

```r
> funw<-function(x) {
+ x*dweibull(x,2.98,15.26)
+ }
> integrate(funw,0,Inf)
```

13.62288 with absolute error < 0.00085

As the distribution is rather similar to the percentile-based distribution of example 1.25 (see Figure 1.4), the means are also very similar.

The expected value for function (transformation) \( g(X) \) is defined analogously

\[
E[g(X)] = \begin{cases} 
\sum_{j} p_j g(x_j) & \text{for discrete } X \\
\int_{-\infty}^{\infty} g(x) f_X(x) \, dx & \text{for continuous } X
\end{cases}
\] (1.16)
Example 1.28. In the large forest stand of Example 1.3, the random variable $X$, tree species, gets values 1, 2, 3 and 4 with probabilities 0.5, 0.3, 0.15, 0.05, respectively. The mean height by species is expressed as

$$g(x) = \begin{cases} 18 & x = 1 \\ 15 & x = 2 \\ 12 & x = 3 \\ 22 & x = 4 \end{cases}$$

The mean height is

$$EH = \sum_j p_j g(X_j) = 0.5 \times 18 + 0.3 \times 15 + 0.15 \times 12 + 0.05 \times 22 = 16.4$$

Let $c$ be a constant and $X$ and $Y$ random variables. The expected value has the following properties

$$E(c) = c$$  \hspace{1cm} (1.17)

$$E(cX) = cE(X)$$  \hspace{1cm} (1.18)

$$E(X + Y) = E(X) + E(Y)$$  \hspace{1cm} (1.19)

These properties hold also when $X$ and $Y$ are correlated.

The expected value of a bivariate random variable is the vector of the expectations of the component variables, where expectations are defined through the marginal distribution. Thus, computing the expectation of a joint density requires only knowledge of the marginal distributions. However, if a transformation depends on several components, the joint density of the related components is needed. For example, the expectation of a bivariate transformation $g(X,Y)$ which depends on a bivariate random vector $(X,Y)$ is analoguously to (1.16)

$$E[g(X,Y)] = \begin{cases} \sum_i \sum_j p_{ij} g(x_i,y_j) f_{X,Y}(x_i,y_j) & \text{for discrete } (X,Y) \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x,y) f(x,y) dy dx & \text{for continuous } (X,Y) \end{cases}$$  \hspace{1cm} (1.20)

Example 1.29. Consider again example 1.23. The expected value of diameter is just the expected value over the marginal distribution of diameters

$$E(X) = \int_{-\infty}^{\infty} x f_X(x) dx.$$  

Correspondingly, the mean height is the expected value over the marginal distribution of heights

$$E(Y) = \int_{-\infty}^{\infty} y f_Y(y) dy.$$
1.3. EXPECTATION, VARIANCE AND COVARIANCE

As an example of the expected value of a bivariate transformation, assume that the volume as a function of diameter and height is given by \( v(x, y) \). The mean volume is the expected value of \( v(x, y) \)

\[
E(v(X, Y)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(x, y) f(x, y) \, dx \, dy.
\]

1.3.2 Variance and standard deviation

In addition to the mean of a random variable, we may be interested in how much the values of a random variable vary around the mean. This variation is described by the variance, which is defined as the expected value of the squared difference between the random variable and its mean:

\[
\text{var} X = E (X - EX)^2.
\]

For the sake of simplicity, denote constant \( EX \) by \( \mu \). A computationally better form is obtained through simple algebra as

\[
\begin{align*}
\text{var} X & = E (X - \mu)^2 \\
& = E (X^2 - 2\mu X + \mu^2) \\
& = E X^2 - 2\mu EX + \mu^2 \\
& = E X^2 - \mu^2 \\
& = \mu^2 - \mu^2 = 0
\end{align*}
\]

(1.21)

Variance is not easy to interpret. An easier-to-interpret measure for the variation around mean is the standard deviation, which is the square root of variance

\[
\text{sd}(X) = \sqrt{\text{var} X}.
\]

(1.22)

It has the same unit as \( X \). In practice, variance is usually used in computations, and the results are transformed to standard deviations for reporting and interpretation.

**Example 1.30.** For the variance of tree diameter in Example 1.12, we need to compute \( \mu = E(X) \) and \( E(X^2) \). From example 1.25, \( \mu = 13.625 \). Equation (1.16) gives

\[
\begin{align*}
E(X^2) & = \int_{-\infty}^{\infty} x^2 f_X(x) \, dx \\
& = \int_{5}^{10} 0.05x^2 \, dx + \ldots + \int_{17}^{24} 0.036x^2 \, dx \\
& = \int_{5}^{10} \frac{0.05}{3} x^3 + \ldots + \int_{17}^{24} \frac{0.036}{3} x^3 \\
& = \frac{1}{3} \left[ 0.05(10^3 - 5^3) + \ldots + 0.036(24^3 - 17^3) \right] \\
& = 210.5
\end{align*}
\]
The variance is
\[
\text{var} X = \mathbb{E} (X^2) - \mu^2 \\
= 210.5 - 13.625^2 \\
= 24.86,
\]
which gives standard error \(sd(X) = \sqrt{24.86} = 4.99\).

These results were computed using
\[
\begin{align*}
> & \# \text{ The second moment} \\
> & \text{mu2<-sum(a/3*(d[-1]^3-d[-5]^3))} \\
> & \text{mu2} \\
> & [1] 210.5 \\
> \end{align*}
\]
\[
\begin{align*}
> & \# \text{ Variance} \\
> & \text{sigma2<-mu2-mu^2} \\
> & \text{sigma2} \\
> & [1] 24.85938
\end{align*}
\]

These results mean that taking randomly trees from a stand with the assumed percentile-based distribution, the mean diameter would in long run be 13.6 cm. Loosely speaking the sampled tree diameters would differ from 13.6 cm, on average, by 5 cm in long run. However, the latter interpretation does not hold exactly. To be specific, the second power of the deviation from 13.6cm would be, on average, \(5^2 \text{cm}^2\) in long run.

**Example 1.31.** To compute the standard deviation of the \(\text{Weibull}(2.98, 15.26)\) distribution, we first compute \(\mathbb{E}(X^2)\) numerically. Thereafter, we use the previously computed \(\mathbb{E}X\) in (1.21) to compute the variance.

\[
\begin{align*}
> & \text{funw2<-function(x) } \\
> & + \ x^2*\text{dweibull(x,2.98,15.26)} \\
> & + \ } \\
> & \text{integrate(funw2,0,Inf)$value->EX2} \\
> & \text{EX2} \\
> & [1] 210.3929 \\
> & \text{integrate(funw,0,Inf)$value->EX} \\
> & \text{EX} \\
> & [1] 13.62288 \\
> & \text{EX2-EX^2} \\
> & [1] 24.81001 \\
> & \text{sqrt(EX2-EX^2)} \\
> & [1] 4.980965
\end{align*}
\]

We see that the variance is very similar to that of the percentile-based distribution, as expected due to the similarity of these two distributions.

**1.3.3 Covariance and correlation**

The average linear joint variation of random variables \(X\) and \(Y\) is called covariance. Consider the difference of \(X\) from the expected value \(\mathbb{E}(X)\), and correspondingly, the difference of \(Y\) from its expected value \(\mathbb{E}(Y)\). The covariance is the expected value of the product of these differences:

\[
cov(X, Y) = \mathbb{E} \left[ (X - \mathbb{E}(X)) (Y - \mathbb{E}(Y)) \right]
\] (1.23)
Computing the product by components and simplifying gives an alternative formulation of
\[ \text{cov}(X, Y) = E(XY) - E(X)E(Y) \] (1.24)

Covariance is a special case of variance. It can be seen by computing the covariance of \( X \) with itself,
\[
\text{cov}(X, X) = E(XX) - E(X)E(X) = E(X^2) - [E(X)]^2
\]
\[ = \text{var}(X). \] (1.25)

**Example 1.32.** To compute the covariance between tree diameter and height in example 1.23, we need to compute the expected value of the product of diameter and height. It is obtained using equation (1.20) as (in this case, \( g(X, Y) = XY \))
\[
E(XY) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xyf(xy)dydx
\]
The expected values of \( X \) and \( Y \) were given in example 1.29. The covariance can be computed as \( \text{cov}(X, Y) = E(XY) - E(X)E(Y) \).

As was variance, also covariance is hard to interpret, because it depends on the variation of the original variables. However, the covariance can be scaled to a much more easy-to-interpret figure, called correlation. The correlation is defined as
\[
\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\text{sd}(X)\text{sd}(Y)}. \]
(1.26)
The correlation varies between -1 and 1. The higher the absolute value of correlation is, the stronger is the linear relationship between \( X \) and \( Y \). If the absolute value of correlation is 1, the relationship between \( X \) and \( Y \) can be exactly expressed by \( Y = aX + b \), where \( a \) and \( b \) are constants. If correlation is negative, \( a \) is negative, and if correlation is positive, \( a \) is positive. If correlation (and covariance) is 0, the random variables are uncorrelated. Correlation is related to independence so that independent random variables are also uncorrelated. However, uncorrelated variables are not necessarily independent (Why?).

**Example 1.33.** Correlations for example 1.23

The properties related to the mean of a random variable are sometimes called the *first order properties*. The properties related to variance and covariance are called the *second order properties*. These are related to the moments of the distribution: expected value is the first moment and variance is the second moment of the distribution. The moments for a distribution function can be computed using the moment generating
function. The third and fourth moment, which are seldom used are the skewness and kurtosis. The analysis of a statistical model is mostly based on the first and second order properties of the random variable. More detailed assumptions on the distribution are usually needed only for testing purposes, where an assumption on a distribution is always needed.

1.3.4 Algebraic rules for variance, and covariance

The following rules are useful in calculating of variance

\[
\text{var}(aX) = a^2 \text{var}(X) \quad (1.27)
\]
\[
\text{var}(X + Y) = \text{var}(X) + \text{var}(Y) + 2\text{cov}(X, Y) \quad (1.28)
\]
\[
\text{var}(X - Y) = \text{var}(X) + \text{var}(Y) - 2\text{cov}(X, Y) \quad (1.29)
\]

Combining rules (1.28) and (1.29) with rule (1.27) gives

\[
\text{var}(aX + bY) = a^2 \text{var}(X) + b^2 \text{var}(Y) + 2ab\text{cov}(X, Y) \quad (1.30)
\]
\[
\text{var}(aX - bY) = a^2 \text{var}(X) + b^2 \text{var}(Y) - 2ab\text{cov}(X, Y) \quad (1.31)
\]

The following rules are useful in computations of covariance

\[
\text{cov}(X, Y) = \text{cov}(Y, X) \quad (1.32)
\]
\[
\text{cov}(X, Y + Z) = \text{cov}(X, Y) + \text{cov}(X, Z) \quad (1.33)
\]
\[
\text{cov}(aX, bY) = ab\text{cov}(X, Y) \quad (1.34)
\]

Combining rules (1.33) and (1.34) gives

\[
\text{cov}(aX+bY, cZ+dW) = ac\text{cov}(X, Z)+ad\text{cov}(X, W)+bc\text{cov}(Y, Z)+bd\text{cov}(Y, W),
\quad (1.35)
\]

where \(X, Y, Z,\) and \(W\) are random variables and \(a, b, c,\) and \(d\) are constants. It can be shown that all rules (1.27) – (1.34) are special cases of rule (1.35).

Adding a constant to a random variable has no effect on variance and covariance

\[
\text{var}(X + a) = \text{var}(X) \quad (1.36)
\]
\[
\text{cov}(X + a, Y + b) = \text{cov}(X, Y) \quad (1.37)
\]

1.3.5 Covariance and correlation matrices

The second order properties of a random vector can be described with a matrix called variance-covariance matrix, also called variance matrix or covariance matrix. The
matrix summarizes all variances and covariances into a single matrix. Let $X = (X_1, X_2, \ldots, X_k)$ be a random vector. The covariance matrix of $X$ is defined as

$$\text{cov}(X) = \text{E} \{ [X - \text{E}(X)][X - \text{E}(X)]' \}$$

$$= \begin{pmatrix}
\text{var}(X_1) & \text{cov}(X_1, X_2) & \cdots & \text{cov}(X_1, X_k) \\
\text{cov}(X_2, X_1) & \text{var}(X_2) & \cdots & \text{cov}(X_2, X_k) \\
\vdots & \vdots & \ddots & \vdots \\
\text{cov}(X_k, X_1) & \text{cov}(X_k, X_2) & \cdots & \text{var}(X_k)
\end{pmatrix}$$

Applying (1.32) shows that the variance-covariance matrix is symmetric. It also positive semidefinite (proof is omitted).

The covariance matrix of random vectors $X_k$ and $Y_p$ is a $k \times p$ matrix, defined as

$$\text{cov}(X, Y') = \text{E} \{ [X - \text{E}(X)][Y - \text{E}(Y)]' \}$$

$$= \begin{pmatrix}
\text{cov}(X_1, Y_1) & \text{cov}(X_1, Y_2) & \cdots & \text{cov}(X_1, Y_p) \\
\text{cov}(X_2, Y_1) & \text{cov}(X_2, Y_2) & \cdots & \text{cov}(X_2, Y_p) \\
\vdots & \vdots & \ddots & \vdots \\
\text{cov}(X_k, Y_1) & \text{cov}(X_k, Y_2) & \cdots & \text{cov}(X_k, Y_p)
\end{pmatrix}.$$ 

Remembering from (1.25) that variance is a special case of covariance, it can be seen that the variance-covariance matrix of $X$ is just a special case of the covariance matrix: $\text{var}(X) = \text{cov}(X, X)$.

In the univariate case, correlation was obtained from covariance by dividing the covariance by the product of standard deviations. Correspondingly, the correlation matrix is obtained by multiplying the covariance matrix from by a diagonal matrix including the inverted standard deviations

$$\text{cor}(X, Y) = \text{diag}[\text{var}(X)]^{-1/2} \text{cov}(X, Y) \text{diag}[\text{var}(Y)]^{-1/2},$$

where

$$\text{diag}[\text{var}(X)]^{-1/2} = \begin{pmatrix}
\frac{1}{\text{sd}(X_1)} & 0 & \cdots & 0 \\
0 & \frac{1}{\text{sd}(X_2)} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \frac{1}{\text{sd}(X_k)}
\end{pmatrix},$$

and correspondingly for $y$.

The algebraic rules of the previous section generalize also for matrices. Equations (1.18), (1.27), and (1.34) generalize to

$$\text{E}(a'X) = a'\text{E}(X)$$

$$\text{var}(a'X) = a'\text{var}(X)a$$

$$\text{cov}(a'X, b'Y) = a'\text{cov}(X, Y')b$$
1.3.6 Sample estimators

The previous sections have shown several important rules for computations on random variables. These results were based on the distribution and resulting expected values, variances and covariances, which are assumed to be known. However, the true distribution of a random variable is seldom, if ever, known in real-life situations. Therefore, also expectations, variances and covariances are usually unknown.

In most cases, the researcher has collected observations on the realizations of the random variables. Using these realizations, we can estimate interesting statistics such as the sample mean, sample variance and sample covariance which are all estimators of the true underlying statistics.

Let us define random variables $X_1, X_2, \ldots, X_n$ to describe a certain characteristic of $n$ different sampling units. The sample mean of is

$$\bar{X} = \frac{X_1 + X_2 + \ldots + X_n}{n} = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

Note that sample mean is not a fixed number but a random variable: different samples from the same population lead to different sample mean. Therefore, the randomness of $\bar{X}$ is due to the randomness of $X_i$. From mathematical point of view, the sample mean $\bar{X}$ is the sum of random variables, $X_1, \ldots, X_n$, which is multiplied by constant $1/n$. Therefore, the expected value and variance of $\bar{X}$ can be derived from rules (1.18), (1.19), (1.27), and (1.28) if the expectation and variance of $X_1, \ldots, X_n$ are known. Furthermore, the central limit theorem states that under mild conditions, the distribution of sample mean approaches the normal distribution as sample size increases, irrespective of the form of the original distribution.

The sample variance is defined as

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2.$$

The sample standard deviation is defined analogously to (1.22) as $S = \sqrt{S^2}$.

**Example 1.34.** Let $X_1, X_2, \ldots, X_n$ be describe the diameters of trees 1, 2, $\ldots$, $n$ from a forest stand, all of which have the same expected value $\mu$ and variance $\sigma^2$. We assume that the diameters are independent. The mean diameter of $n$ trees, $\bar{X}$, is a random
variable that has expectation

\[ E(\bar{X}) = E\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) \]
\[ = \frac{1}{n} E\left(\sum_{i=1}^{n} X_i\right) \]
\[ = \frac{1}{n} \sum_{i=1}^{n} E(X_i) \]
\[ = \frac{1}{n} n E(X_i) \]
\[ = \mu. \]

The second line resulted from application of rule (1.18) and the third row from application of rule (1.19).

The variance of mean diameter \( \bar{X} \) would be

\[ \text{var}(\bar{X}) = \text{var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) \]
\[ = \left(\frac{1}{n}\right)^2 \text{var}\left(\sum_{i=1}^{n} X_i\right), \]

which was obtained through application of rule (1.27). Independence of \( X_i \) implies that \( \text{cov}(X_i, X_j) = 0 \) for all \( i \neq j \). Using rule (1.28) we get

\[ \text{var}(\bar{X}) = \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} \text{var}(X_i) \]
\[ = \frac{1}{n^2} n \sigma^2 \]
\[ = \frac{1}{n} \sigma^2 \]

The standard deviation of mean diameter of the sample would be \( \text{sd}(\bar{X}) = \sigma/\sqrt{n} \).

This equation is the widely applicable equation for the standard error of the sample mean.

The above definitions used uppercase letters to show that these quantities are random variables, not realizations. The realizations of sample statistics are obtained by replacing random variable \( X \) with its realization \( x \). Those realized values of \( \bar{X}, S^2 \), and \( S \) would be denoted by \( \bar{x}, s^2 \), and \( s \), correspondingly.

**Example 1.35.** Assume that we have independently taken observations of diameter for 4 trees in stand, \( x_1 = 15, x_2 = 4, x_3 = 12 \), and \( x_4 = 11 \)cm. The sample mean is

\[ \bar{x} = \frac{1}{4}(15 + 4 + 12 + 11) = 10.5 \text{cm}. \]
The sample variance is

\[ s^2 = \frac{1}{4} \left[ (15 - 10.5)^2 + (4 - 10.5)^2 + (12 - 10.5)^2 + (11 - 10.5)^2 \right] = 21.67\text{cm}^2. \]

The sample standard deviation is \( s = \sqrt{21.67} = 4.65\text{cm}. \)

The sample mean can also be seen as the least squares estimator of the expected value as follows. Consider the sum of squares \( \sum_i = 1^n(x_i - a)^2 \) as a function of \( a \). The value of \( a \) that minimizes is the sample mean is \( \bar{x} \), the sample mean. Because the sample mean minimizes the sum of squares the true expected value necessarily has a higher sum of squares than \( \sum_i = 1^n(x_i - a)^2 \). Therefore, estimating sample variance by \( \frac{1}{n} \sum_i = 1^n(x_i - a)^2 \) provides underestimate of the true variance. An unbiased estimator results from replacing \( \frac{1}{n} \) by \( \frac{1}{n-1} \). Therefore the reason for having \( n-1 \) in the denominator is due to one degree of freedom that was used for estimating \( \bar{x} \). Let random variables \( Y_1, Y_2, \ldots, Y_n \) describe some other property of the same sampling units than \( X_1, X_2, \ldots, X_n \). The sample covariance is defined as

\[ C = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}). \]

The sample correlation can be defined using sample covariance and sample standard deviation in equation (1.26). The expression reduces to

\[ R = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2 \sum_{i=1}^n (Y_i - \bar{Y})^2}}. \]

As with other sample statistics, \( C \) and \( R \) are also random variables. The realizations \( c \) and \( r \) are obtained by utilizing the observed sample values in the equations.

### 1.4 Common distribution functions

This section shortly presents distribution functions that are important in linear modeling applications. The distributions are classified in three classes: Discrete distributions, continuous distributions, and distributions of important functions of Standard normal variable. The last two classes differ in that the third class is derived for inference on normally distributed populations.

#### 1.4.1 Discrete distributions

**Bernoulli and Binomial distribution**

A bernoulli trial has two possible discrete outcomes: success and failure. A random variable \( X \) has \( \text{Bernoulli}(p) \) distribution if it takes the value of 1 ("success") with probability \( p \), and value 0 ("failure") with probability \( 1 - p \). The expected value and variance
of \( X \) are obtained using equations (1.15) and (1.21) as
\[
E(X) = 1p + 0(1 - p) = p \\
E(X^2) = 1^2p + 0^2(1 - p) = p \\
\text{var}(X) = E(X^2) - [E(X)]^2 = p - p^2 = p(1 - p)
\]

Assume that we make \( n \) independent bernoulli trials, and let \( Y \) be the number of successes. The variable \( Y \) is said to have a \textit{Binomial}(\( n, p \)) distribution with parameters \( n \) and \( p \), which has pmf
\[
P(Y = y|n, p) = \binom{n}{y} p^y (1 - p)^{n-y}
\]
In forestry, binomial distribution may be used, for example, to present plant health or survival, tree species, etc. The expectation and variance of binomial distribution can be deduced using rules (1.19) and (1.28) as
\[
E(Y) = np \\
\text{var}(Y) = np(1 - p)
\]

A generalization of binomial distribution is the \textit{Multinomial}(\( n, p_1, \ldots, p_k \)) distribution, where the number of classes is \( k \) instead of 2, the number of classes in the binomial distribution. Let random variable \( Z \) be “the number of bernoulli trials needed to get \( r \) successes”. Then \( Z \) would have the \textit{Negative binomial}(\( r, p \)) distribution. Such a distribution could be used, for example, in assessing how many trees should be bored in a field experiment to get at least \( r \) trees with a butt rot infection.

In R, functions \texttt{dbinom}, \texttt{pbinom}, \texttt{qbinom} can be used for computing pmf, cdf and inverse cdf of the binomial distribution. Function \texttt{rbinom} can be used for generating a sample from a specified binomial distribution. The corresponding functions for multinomial and negative binomial are (guess what they do) \texttt{dmultinom}, \texttt{rmultinom}, \texttt{rnegbin}.

\textbf{Poisson distribution}

Poisson distribution is a discrete distribution, that can be used for modeling of counts. It takes only nonnegative integer values. It can be used, for example, in modeling the number of individuals in a line, if new individuals enter and leave the line at random. If trees in a forest stand are located completely randomly, the number of trees within a fixed area is distributed according to the \textit{Poisson}(\( \lambda \)) distribution. Random variable \( X \) has the Poisson distribution if the pmf is
\[
P(X = x|\lambda) = \frac{e^{-\lambda} \lambda^x}{x!}
\]
where $\lambda$ is the parameter, which is often called intensity. In the random forest example, it is the stand density (trees per area unit). Both mean and variance of Poisson random variable can be shown to be $\lambda$. The lower plots in figure 1.2 show an example of the Poisson distribution.

R has functions `dpois`, `ppois` and `rpois` for computing pmf, cdf and inverse cdf of the Poisson distribution. There is also function `rpois` for generating Poisson distributed random numbers.

### 1.4.2 Continuous distributions

#### Continuous uniform distribution

The $\text{Uniform}(a, b)$ distribution is defined by spreading the probability mass uniformly over interval $[a, b]$. The cdf of uniform distribution is

$$F(x|a,b) = \begin{cases} 
0 & x < a \\
\frac{a}{b-a} + \frac{1}{b-a}x & a \leq x < b \\
1 & x \geq b 
\end{cases},$$

and the density is

$$f(x|a,b) = \begin{cases} 
\frac{1}{b-a} & a \leq x < b \\
0 & \text{otherwise}
\end{cases}.$$  

The expectation and variance are $E(X) = \frac{b-a}{2}$ and $\text{var}(X) = \frac{(b-a)^2}{12}$.

R has functions `dunif`, `punif`, `qunif` and `runif` for density, distribution function, quantile function and random number generation with uniform distribution. Uniform distribution is seldom a good assumption for a distribution of a random variable. However, uniform random numbers are very often needed in simulations. They are also needed in generating random numbers from any specific distribution using the probability integral transformation approach.

**Example 1.36.** One wants to simulate 10 tosses of a coin. One alternative to proceed is to simulate 10 random numbers from the $\text{Uniform}(0, 1)$ distribution. Realizations smaller than 0.5 are interpreted as heads (H) and realizations equal to or above 0.5 as tails (T). The random numbers obtained using `runif(10)` are 1 1 0 1 1 0 0 1 1 0, results simulated tosses H H T H T T H H T.

Another example of using the uniform distribution to generate random numbers was provided in example 1.19.

#### Normal distribution

The normal (Gaussian) distribution is probably the most well-known statistical distribution. Casella and Berger (2002) mention three main for the special role of Normal
1.4. COMMON DISTRIBUTION FUNCTIONS

A distribution among the body of statistics. First, Normal distribution is very tractable analytically. Second, the shape of Normal distribution is the symmetric Bell-shape, which makes it an appealing alternative for many population models. Third, the central limit theorem shows that the normal distribution is a good approximation for several distributions under mild conditions and with large sample sizes.

The density of normal distribution $\mathcal{N}(\mu, \sigma^2)$ is

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

where $E(X) = \mu$ and $\text{var}(X) = \sigma^2$. The cdf of normal distribution cannot be expressed in a closed form. The standard Normal distribution is the normal distribution with mean of 0 and variance of 1, $\mathcal{N}(0,1)$.

It is interesting and important to note that the sum of independent normal random variables $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ is also normal with $X_1 + X_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$. In several statistical tests, including the tests related to regression analysis and the linear model, the random variables to be analyzed are assumed to follow the normal or multinormal distribution. However, it should be pointed out that normal distribution needs not to be assumed in the estimation of parameters using the methods based on least squares. The central limit theorem states that the distribution of the means from samples of same size approaches the normal distribution as sample size gets large, regardless of the distribution of the random variable in the population.

**Example 1.37.** Assume that the lifetime of a butterfly is highly skewed to the right, having distribution shown in the upper left plot of Figure 1.9. 1000 samples of size 5, 10 and 20 were taken from the population. The other plots of 1.9 show the distributions of sample means with the three applied sample sizes. The sample mean approaches the normal distribution quite fast, being fairly close to normal even with sample size of 10. The example was done using the following code:

```r
windows(width=6,height=5)
par(mfcol=c(2,2),mai=c(0.6,0.5,0.1,0.1),mgp=c(2,0.7,0),cex=0.8)
pop<-rexp(10000,1/2) # generate the population from exponential distribution
mu<-mean(pop)
sd<-sd(pop)
means5<means10<means20<-c() # initialize the means- vectors
for (i in 1:1000) { # repeat 1000 times
  s5<-sample(pop,5) # take samples from populations
  s10<-sample(pop,10)
  s20<-sample(pop,20)
  means5<-c(means5,mean(s5)) # add sample mean to the end of mean vector
  means10<-c(means10,mean(s10))
  means20<-c(means20,mean(s20))
}
x<-seq(0,5,0.1)
# produce teh graphs
hist(pop,main="Population")
lines(x,dnorm(x,mu,sd/sqrt(5)))
hist(means5,freq=FALSE,main="Mean of 5 observations")
lines(x,dnorm(x,mean(means5),sd/sqrt(5)))
hist(means10,freq=FALSE,main="Mean of 10 observations")
lines(x,dnorm(x,mean(means10),sd/sqrt(10)))
hist(means20,freq=FALSE,main="Mean of 20 observations")
lines(x,dnorm(x,mean(means20),sd/sqrt(20)))
```
Figure 1.9: Illustration of Example 1.37.
1.4. COMMON DISTRIBUTION FUNCTIONS

Multivariate normal distribution

Random vector $Y_{n \times 1} = (Y_1, Y_2, \ldots, Y_n)'$ follows the $n$-dimensional multivariate normal distribution with expectation $\mu_{n \times 1}$ and variance-covariance matrix $\Sigma_{n \times n}$, $N_n(\mu, \Sigma)$, if $Y$ has (joint) density

$$f(y|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2} (y-\mu)' \Sigma (y-\mu)}.$$

An alternative definition for normal distribution is obtained by defining that $Y \sim N_n(\mu, \Sigma)$ if $t'Y \sim N(t'\mu, t'\Sigma t)$ with any $t \in \mathbb{R}$. If $\Sigma$ is a diagonal matrix, then the result that the sum of independent normal variates are also normally distributed can be proved through appropriate selection of $t$. The marginal distributions of a multinormal distribution are univariate normal distributions. Furthermore, the correlation between component variables is always linear. This results in that under multivariate normality, the Best Linear Prediction is always also the Best Predictor. The properties of the multinormal distribution were demonstrated in example 1.24.

Lognormal distribution

If $\ln X$ is normally distributed with mean $\mu$ and variance $\sigma^2$, then $X$ is said to be lognormally distributed. The pdf of lognormal distribution is obtained straightforwardly using (1.11) with normal distribution

$$f_X(x) = \frac{1}{\sqrt{2\pi \sigma^2}} \frac{e^{-(\ln(x)-\mu)^2/(2\sigma^2)}}{x} \quad 0 \leq x < \infty$$

The expectation and variance are $E_X = e^{\mu+\sigma^2/2}$ and $\text{var} X = e^{2(\mu+\sigma^2)}$.

In many forestry applications, the response to be modeled does not express a normal distribution. A very common strategy for those cases is to take a logarithmic transformation of the response, and assume the residuals to be normally distributed. This implies that the original response is assumed to be lognormally distributed. The predictions from such a model are unbiased in the logarithmic scale. However, in the back-transformed scale, the predictions are downward biased, due to that for lognormal $X$, $E X = e^{\mu+\sigma^2/2}$, not $E X = e^\mu$. If the normality of residuals holds, a suitable strategy for correction of bias is to add half of the residual variance to the logarithmic predictions before applying the exponential transformation to get unbiased predictions in the original scale. However, this may not be a good strategy if normality of residuals does not hold. An alternative has been presented in (Lappi et al. 2006, p. 103 – 105).

Example 1.18 demonstrated the relationship of normal and lognormal distribution. For making computations with lognormal distribution in R, the standard R includes functions dlnorm, plnorm, qlnorm and rlnorm.
Weibull distribution

The (two-parameter) Weibull($\alpha, \beta$) distribution has distribution function

$$F(x|\alpha, \beta) = 1 - \exp \left\{ - \left( \frac{x}{\beta} \right)^\alpha \right\},$$

and density

$$f(x|\alpha, \beta) = \frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-\left( x/\beta \right)^\alpha}, \quad (1.41)$$

where $\alpha$ and $\beta$ are the shape and scale parameters, respectively. The expectation and variance are $E_X = \beta \Gamma(1 + \frac{1}{\alpha})$ and $\text{var} \, X = \beta^2 \left[ \Gamma(1 + \frac{2}{\alpha}) - \Gamma^2(1 + \frac{1}{\alpha}) \right]$.

The Weibull has been used for modeling failure time data. A special case of Weibull distribution is obtained with $\alpha = 1$, called the Exponential($\beta$) distribution. The exponential distribution is used for modeling lifetimes. The population distribution of Example 1.37 was generated using the exponential distribution with $\beta = 2$. In forestry, the Weibull distribution has been used for modeling distributions of tree DBH since the paper of Bailey and Dell (1973). Also a three-parameter version with an additional parameter for location has been used. The reason for the popularity of Weibull distribution its flexibility with only two parameters. In addition, it has a closed form solution for the cdf, which eases the computation of diameter class frequencies. In particular, there are no theoretical reasons why Weibull should be favored over other alternatives.

R has function `dweibull`, `pweibull`, `qweibull` and `rweibull` for computations with Weibull distribution.

Other distributions used in forestry

In addition to the Weibull distribution, many other distributions have been used for modeling tree size within a plot or stand. These distributions include include the beta, Jonson’s SB and the logit-logistic distribution.

The beta($\alpha, \beta$) distribution has density

$$f(x|\alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1 - x)^{\beta-1},$$

where $\alpha, \beta > 0$ are the two shape parameters. This two-parameter form of the distribution is restricted to the range $0 \leq x \leq 1$. The expectation and variance of this constrained form are $E_X = \frac{\alpha}{\alpha + \beta}$ and $\text{var} \, X = \frac{\alpha \beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)}$. The beta function can be defined through gamma function $B(\alpha, \beta) = \frac{\Gamma(\alpha) \Gamma(\beta)}{\Gamma(\alpha + \beta)}$. R has functions `pbeta`, `dbeta`, `qbeta`, and `rbeta` for the distribution function, density, quantile function, and random number generation using the two-parameter version of the beta distribution.
If the beta distribution is used for modelling tree diameters, a four-parameter form is used by using the minimum and maximum diameters $d_{\text{min}}$ and $d_{\text{max}}$:

$$f(x|\alpha, \beta) = \frac{1}{(d_{\text{max}} - d_{\text{min}})B(\alpha, \beta)} \left( \frac{x - d_{\text{min}}}{d_{\text{max}} - d_{\text{min}}} \right)^{\alpha-1} \left( 1 - \frac{x - d_{\text{min}}}{d_{\text{max}} - d_{\text{min}}} \right)^{\beta-1}.$$ 

In such applications, also the minimum and maximum diameters are treated as parameters rather than interpreting them as fixed constants, resulting in that the distribution has a total of four parameters.

The density of Jonsons $SB(\delta, \gamma, \lambda, \xi)$ distribution using notations of Siipilehto (1999) is

$$f(x|\delta, \gamma, \lambda, \xi) = \frac{\delta}{\sqrt{2\pi}\lambda} (\xi + \lambda - x)(x - \xi) e^{-\frac{1}{2}\left[\gamma + \delta\ln\left(\frac{x - \xi}{\xi + \lambda - x}\right)\right]^2},$$

where $\xi$ is the location parameter, $\lambda$ the scale parameter, and $\delta$ and $\gamma$ are the shape parameters affecting in kurtosis and asymmetry of the distribution, respectively.

The logit-logistic distribution (Tadikamalla et al. 1982, Wang and Rennolls 2005) can represent larger variation of skewness-kurtosis combinations than several other distributions including Weibull, beta and Jonhsons SB distribution. In contrast to beta and SB, it also has a closed-form expression for cdf. The p.d.f. and density of the logit-logistic($\psi, \lambda, \phi, \sigma$) distribution between the minimum and maximum diameters $\psi$ and $\lambda$ are

$$F(x|\psi, \lambda, \phi, \sigma) = \frac{1}{1 + \exp\left(\frac{\phi}{\sigma}\left(\frac{x - \psi}{\lambda - \psi}\right)\right)^\frac{1}{2}}$$

$$f(x|\psi, \lambda, \phi, \sigma) = \frac{\lambda - \psi}{\sigma(x - \psi)(\lambda - x)} \exp\left(\frac{-\psi}{\sigma}\left(\frac{x - \psi}{\lambda - \psi}\right)\right)^\frac{1}{2} + \exp\left(\frac{\phi}{\sigma}\left(\frac{x - \psi}{\lambda - \psi}\right)\right)^\frac{1}{2} + 2.$$

The SB distribution is formulated by assuming that logit-transformation of $x$ that is scaled to range $[0, 1]$, $z = \logit\left(\frac{x - \xi}{\lambda - x}\right) = \logit(y) = \ln\left(\frac{y}{1-y}\right)$ follows the normal distribution with $\mu = -\frac{\gamma}{\delta}$ and $\sigma^2 = \frac{1}{\delta^2}$. Thus, it could be called logit-normal distribution, by an analogy to the lognormal distribution. Correspondingly, the logit-logistic distribution is obtained by a similar approach, assuming that the logit follows the logistic distribution instead of the normal distribution.

**Truncated distributions**

A left-truncated version of any distribution function $F(x)$ with truncation point $t$ is

$$F_t(x) = \frac{1}{1 - F(t)}(F(x) - F(t)). \quad (1.42)$$

Using Weibull distribution in equation (1.42) gives the c.d.f of truncated Weibull distribution as

$$F_t(x|\alpha, \beta) = 1 - \exp\left\{\left(\frac{t}{\beta}\right)^\alpha - \left(\frac{x}{\beta}\right)^\alpha\right\}.$$
The truncated distribution may be useful in situations where only diameters above a fixed limit of have been measured. The truncated Weibull function was found to be the best alternative from among several candidates in the recent study of Pahláš et al. (2007). Especially, the truncated distribution is favoured over the well-known three-parameter version of the Weibull-distribution, because a fixed measurement limit for diameter leads to censored data. Such data are realistically modelled with a truncated distribution, having a jump in the density at \( x = t \).

**Exponential family**

One important group of distributions is the exponential family of distributions. It includes those distributions whose density or pmf can be expressed as

\[
 f(x|\theta) = h(x)c(\theta) \exp \left( \sum_{i=1}^{k} w_i(\theta)t_i(x) \right),
\]

where \( h(x) \geq 0, c(\theta) \geq 0, t_1, \ldots, t_k \) are real-valued functions of \( x \) that do not depend on \( \theta \), and \( w_1, \ldots, w_k \) are real-valued functions of \( \theta \) that do not depend on \( x \). The exponential family is important in that the expectations, variances and covariances of the random variables, as well as the maximum likelihood estimators for the parameter vector \( \theta \) can be expressed in a form that eases the computations. Furthermore, the ML estimators have certain desired properties. For further details, see Casella and Berger (2002). For these reasons, applications of generalized linear models and generalized linear mixed models, which are strongly based on ML theory, have been developed for the distributions belonging to the exponential family. Of the distributions presented in this paper, the binomial, multinomial, negative binomial, Poisson, normal, lognormal, and beta (with fixed \( a \) and \( b \)) distributions belong to the exponential family.

**Example 1.38.** The lognormal distribution can be expressed as

\[
 f_X(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-(\ln(x) - \mu)^2/(2\sigma^2)}
\]

By defining \( h(x) = 1/x, c(\theta) = 1/\sqrt{2\pi}\sigma^2, w_1(\theta) = -1/(2\sigma^2), w_2(\theta) = \mu/\sigma^2, w_3(\theta) = -\mu^2/(2\sigma^2), t_1(x) = \ln(x)^2, t_2(x) = \ln(x), \) and \( t_3(x) = 1 \) we see that the lognormal density is of form 1.43.

R library `MASS` has function `mvrnorm` for generating multivariate normal random numbers.
1.4. COMMON DISTRIBUTION FUNCTIONS

1.4.3 Distributions of important transformations of a standard Normal variate

**χ² distribution**

If \( Z \) follows the standard normal distribution \( N(0, 1) \), then the sum of squares \( Y = \sum_{i=1}^{p} Z_i^2 \) follows the \( \chi^2(p) \) distribution with \( p \) degrees of freedom. It has density

\[
f(x|p) = \frac{1}{\Gamma(p/2)2^{p/2}}x^{p/2-1}e^{-x/2} \quad 0 \leq x < \infty, \; p = 1, 2, \ldots,
\]

where \( \Gamma(\alpha) = \int_{0}^{\infty} t^{\alpha-1}e^{-t} dt \) is the gamma function that needs to be evaluated numerically. The expected value and variance of \( \chi^2(p) \) distribution are \( p \) and \( 2p \), respectively. The expectation results directly from that

\[
E(Z) = E\left( \sum_{i=1}^{p} Z_i^2 \right) = \sum_{i=1}^{p} E(Z_i^2) = p(1+0) = p
\]

Chi square distribution is used in construction of tests for the residual sum of squares for a regression model with normally distributed residuals. R has functions `dchisq`, `pchisq`, `rchisq` and `qchisq` for different computations with the \( \chi^2 \)-distribution. There is also function `gamma` for evaluating the gamma function.

**Students t-distribution**

If \( Z \sim N(0, 1) \) and \( u \sim \chi^2(n) \), then the ratio \( v = \frac{z}{\sqrt{u/n}} \) follows the Students t-distribution, \( t(n) \), with \( n \) 4 degrees of freedom. The density of students t-distribution is

\[
f(x|n) = \frac{\Gamma((n+1)/2)}{\Gamma(n/2)} \frac{1}{\sqrt{n\pi}} \frac{1}{(1 + x^2/n)^{(n+1)/2}} \quad -\infty < x < \infty, \; n = 1, 2, \ldots
\]

The mean and variance are \( E(X) = 0 \) for \( (n > 1) \) and \( \text{var}(X) = \frac{n}{n-2} \) for \( (n > 2) \).

The t-distribution was originally developed for testing if the mean of a sample of size \( n \) from a normally distributed population significantly differs from a fixed mean. The Students t-distribution is utilized instead of the normal because the variance is estimated from a sample, thus being a random variable with \( \chi^2 \) distribution. The density is bell-shaped, and symmetric as normal distribution is too, but it is wider than the standard normal distribution. However, as \( n \) gets large, the t-distribution approaches the
CHAPTER 1. RANDOM VARIABLES

The value of 0.975th quantile

Degrees of freedom (n)

Figure 1.10: The 0.975th quantile is needed for constructing two-sided 95% confidence intervals for means. The horizontal line at 1.96 level shows the 0.975th quantile of normal distribution, and the decreasing line that of t-distribution as the degrees of freedom change from 2 to 100.

standard normal distribution. Thus, normal distribution leads to very similar inference as t distribution with large sample sizes tests on a large sample can be based on normal distribution (Figure 1.10).

R has functions \texttt{dt, pt, qt} and \texttt{rt} for computations with Student’s t-distribution.

\textbf{F distribution}

Let \( X \) and \( Y \) be independent random variables that follow the \( \chi^2 \) distribution: \( X \sim \chi^2(m) \) and \( Y \sim \chi^2(n) \). As an example, \( X \) and \( Y \) could be two different sums of squares. The distribution of ratio \( F = \frac{X/m}{Y/n} \) follows the \( F(m, n) \) distribution with \( m \) and \( n \) degrees of freedom. The density is

\[
f(z|m,n) = \frac{\Gamma\left(\frac{m+n}{2}\right)}{\Gamma\left(\frac{m}{2}\right)\Gamma\left(\frac{n}{2}\right)} \left(\frac{m}{n}\right)^{m/2} \frac{x^{\frac{m-2}{2}}}{(1 + \frac{m}{n}x)^{\frac{m+n}{2}}} \quad 0 \leq z < \infty, \quad m, n = 1, 2, \ldots.
\]

The mean and variance are \( E(Z) = \frac{n}{n+2} \) for \( n > 2 \) and \( \text{var}(Z) = \frac{2n}{n+2} \frac{m+n-2}{m(n-2)} \) for \( n > 4 \).

The F-distribution is used in testing two regression models against each other. R has function \texttt{pf, df, qf} and \texttt{rf} for computations with F distribution.

\textbf{1.4.4 Specialities related to modelling tree diameter}

\textbf{The mixture distributions}

The mixture distribution is obtained by assuming separate unimodal distributions for each age class, tree species or other subpopulation of the trees in the stand. The density is the weighted sum of the component densities, where the weights show the proportion of the component subpopulations. See e.g. Liu et al. (2002) and Mehtätalo (2006).
1.5. FITTING DISTRIBUTION FUNCTIONS TO DATA

Segmented distribution

In a segmented distribution, more flexibility is obtained to the diameter distribution by taking parts of different densities at different diameter ranges. The segmented distribution is constructed from pieces of the selected distribution, and the number of pieces and locations of cutting points are fixed before the fitting procedure. For example, the distribution may be assumed to be of the Weibull form, as in Cao and Burkhart (1984).

The percentile-based distribution


1.5 Fitting distribution functions to data

We have already shown how to estimate the first and second-order properties of a statistical distribution using observed data. However, sometimes one might want to estimate the whole distribution. This leads to the question how to estimate the parameters of an assumed distribution given an observed dataset.

To do so, one first needs to find an appropriate distribution function for the dataset in hand. In many cases, there are some theoretical results supporting the choice (e.g., if the data represent sample means or number of trees on a sample plot with random pattern of tree locations). In other situations, no theoretical results exist on what distribution is theoretically justified for the data at hand. For example, there are no theoretical results on the diameter distribution of trees in an even-aged stand. In this case, the choice of the distribution may be based e.g. on the flexibility and range of the distribution. For example, one may want to use a function that does not allow negative diameters, probably has a finite upper bound, and are able to express the myriad of shapes that exist in forest stands.

1.5.1 The method of moments

The method of moments is the maybe the oldest and most simple method for fitting an assumed distribution function to observed data. It has been used in late 1800 by Carl Pearson (Casella and Berger 2002). The earliest application to tree diameter data dates back to 1914 (Cajanus 1914), see Figure 1.11. Burk and Newberry (1984) presented the method of moments for the three-parameter Weibull distribution.

Let $X_1, X_2, \ldots, X_n$ be a sample from a population with density $f(x|\theta_1, \theta_2, \ldots, \theta_p)$. For example, it may be a sample of tree diameters, and the assumed distribution may be a Weibull density with $\theta_1 = \alpha$ and $\theta_2 = \beta$. The method of moments seeks for
CHAPTER 1. RANDOM VARIABLES

Figure 1.11: A table for computing the first four moments of diameter data and, based on these moments, estimating the parameters of the Gram-Charlier approximation of tree diameter distribution (Ilvesalo 1914). The method was originally developed by Cajanus (1914), who visited professor C.W.L. Charlier in Sweden to get supervision in statistics and also to use his mechanical calculator for computations with large datasets.
parameter estimates by setting the first \( p \) moments of the assumed distribution to be equal to the corresponding sample moments. This leads to a system of \( p \) equations, that are solved for \( \theta_1, \ldots, \theta_p \). With the two-parameter Weibull-distribution or Normal distribution, we need only two first moments, i.e., the mean and variance, so the number of equations would be only 2. With four-parameter distributions, such as Johnsons SB, logit-logistic or the Gram-Charlier approximation, the number of parameters and equations would be four.

To define the method formally, we define the sample moments and corresponding population moments as (Casella and Berger 2002)

\[
\begin{align*}
m_1 &= \frac{1}{n} \sum_{i=1}^{n} X_i \quad \mu'_1 = E(X) \\
m_2 &= \frac{1}{n} \sum_{i=1}^{n} X_i^2 \quad \mu'_2 = E(X^2) \\
&\vdots \\
m_p &= \frac{1}{n} \sum_{i=1}^{n} X_i^p \quad \mu'_p = E(X^p)
\end{align*}
\]

The population moments are the expected values of transformations \( X, X^2, \ldots, X^p \), which were previously defined for a continuos distribution as \( E(X^n) = \int_{-\infty}^{\infty} x^n f(x|\theta_1, \ldots, \theta_p)dx \).

For a corresponding expression for a discrete distribution, see Equation (1.16). The expectation is a function of parameters \( \theta_1, \ldots, \theta_p \). The moment estimators of distribution parameters are obtained by solving the following system of equations for \( \theta_1, \ldots, \theta_p \):

\[
\begin{align*}
m_1 &= \mu'_1(\theta_1, \ldots, \theta_p) \\
m_2 &= \mu'_2(\theta_1, \ldots, \theta_p) \\
&\vdots \\
m_p &= \mu'_p(\theta_1, \ldots, \theta_p)
\end{align*}
\]

**Example 1.39.** Assume that we have observations \( x_1, \ldots, x_5 = 15.1, 17.1, 10.0, 13.5, 16.6, 18.1, 11.0, 13.4, 15.2, 17.1, 15.9 \) of tree diameters. We get \( m_1 = \bar{X} = 14.54 \) and \( m_2 = \bar{X}^2 = 216.64 \). Assuming that the distribution is \( N(\mu, \sigma^2) \), the first equation becomes \( \mu = \bar{X} = 14.54 \). For the second equation, recall that \( var(X) = E(X^2) - E(X)^2 \), which gives \( E(X^2) = \sigma^2 + \mu^2 \). We get \( \sigma^2 + \mu^2 = 216.64 \), which gives \( \sigma^2 = 216.64 - 14.54^2 = \frac{1}{n} \sum (X_i - \bar{X})^2 \).

Thus, the moment-estimates for \( \mu \) is just the sample mean, and that for \( \sigma^2 \) is the sample variance in the form that has \( n \) in the denominator instead of \( n - 1 \).
CHAPTER 1. RANDOM VARIABLES

Figure 1.12: Figure for examples 1.39 and 1.40. The histogram shows the sample distribution, the solid line the fitted normal distribution, and the dashed line the fitted Weibull distribution. The fits are based on the method of moments.

Figure 1.12 shows the histogram of observations and the estimated normal density.

The R-code for Example 1.39

```r
> x<-c(15.1, 17.1, 10.0, 13.5, 16.6, 18.1, 11.0, 13.4, 15.2, 17.1,
+ 15.9, 17.7, 14.4, 13.8, 14.8, 11.9, 13.0, 14.7, 16.5, 11.0)
> m1<-mean(x)
> m2<-mean(x^2)
> m1
[1] 14.54
> m2
[1] 216.637
> mu<m1
> sigma<sqrt(m2-m1^2)
> hist(x,freq=FALSE,xlim=c(0,40),ylim=c(0,0.08))
> x0<-seq(0,40,0.1)
> lines(x0,dnorm(x0,mu,sigma))
```

Example 1.40. Assume that the data of the previous example (1.39) follows the Weibull distribution.

The moments for Weibull distribution are defined as $E(X)^n = \beta^n \Gamma \left( 1 + \frac{n}{\alpha} \right)$. Thus, we get the system of equations

$$\beta \Gamma \left( 1 + \frac{1}{\alpha} \right) = 14.54$$
$$\beta^2 \Gamma \left( 1 + \frac{2}{\alpha} \right) = 216.64$$

The solution for this system was obtained numerically as follows. First, we defined a function for the sum of squared differences between the theoretical and sample moments,

$$m(\alpha, \beta) = (m_1 - \mu'_1(\alpha, \beta))^2 + (m_2 - \mu'_2(\alpha, \beta))^2.$$
1.5. FITTING DISTRIBUTION FUNCTIONS TO DATA

If both theoretical moments equal to the corresponding sample moments, then this function should get the value of 0, otherwise the value should be above 0. For minimizing $m$ with respect to the parameters, we used function `nlm` which minimizes a one-dimensional function for given parameters. The obtained solution was $\hat{\alpha} = 10.01$ and $\hat{\beta} = 15.36$. The value of function $m$ at the solution was 0.0054, which is close to 0.

The dashed line in figure 1.12 shows the graph of the estimated density. It seems to fit the data quite well, being slightly skewed to the right.

The R-code for Example 1.40

```r
> # Moment-based recovery of Weibull-parameters
> # A function that computes the moments of weibull-function
> wmom<-function(n,alpha,beta) {
+ beta^n*gamma(1+n/alpha)
+ }
>
> # sum of squared differences (m1-mu1)^2+(m2-mu2)^2;
> # gets value of 0 if the system of equations has a solution.
> fn<-function(theta,m1,m2) {
+ alpha<-theta[1]
+ beta<-theta[2]
+ (wmom(1,alpha,beta)-m1)^2+(wmom(2,alpha,beta)-m2)^2
+ }
>
> solution<-nlm(fn,p=c(10,15),m1=m1,m2=m2)
> solution
$minimum
[1] 0.005399023
$estimate
[1] 10.01556 15.35970
$gradient
[1] 0.0023474506 -0.0001002351
$code
[1] 2
$iterations
[1] 4

> alphamom<-solution$estimate[1]
> betamom<-solution$estimate[2]
>
> lines(x0,dweibull(x0,alphamom,betamom),lty="dashed")
```

Example 1.41. We may not have expressions for the moments available, and for some reasons (e.g., lack of skills or time) we may not be able to obtain expressions for the moments. However, we may use numerical integration to compute the moments.

Assume that we aim at moment-based estimation of Weibull parameters but we do not know the expressions for the moments. That is why we just utilize the definition of expected value and integrate it numerically. That is, we replace function

$$E(X^n) = \beta^n \Gamma \left(1 + \frac{n}{\alpha}\right)$$
with the equivalent, more general equation
\[ E(X^n) = \int_{-\infty}^{\infty} x^n f(x|\alpha, \beta) \, dx. \]

The R-code for this purpose is shown below

```r
> # A function that numerically computes the moments of Weibull-function
> wmom2<-function(n,alpha,beta) {
+   f1<-function(x) x^n *dweibull(x,alpha,beta)
+   integrate(f1,0,Inf)$value
+ }
> > # sum of squared differences (m1-mu1)^2+(m2-mu2)^2;
> # gets value of 0 if the system of equations has a solution.
> fn<-function(theta,m1,m2) {
+   alpha<-theta[1]
+   beta<-theta[2]
+   (wmom2(1,alpha,beta)-m1)^2+(wmom2(2,alpha,beta)-m2)^2
+ }
> > solution<-nlm(fn,p=c(10,10),m1=m1,m2=m2)
> solution
$minimum
[1] 0.005708579
$estimate
[1] 10.12320 15.35496
$gradient
[1] 2.348666e-03 -7.952121e-05
$code
[1] 2
$iterations
[1] 7
```

The solution is similar to that of the previous example. It is many times easiest way to obtain solutions by just applying the numerical algorithms that are readily available. However, the numerical solution may take few microseconds more time. In addition, there may be a higher risk of unstability and wrong solutions. Thus, the more numerical methods are used, the more careful the researcher should be. However, note that function $\Gamma$ is also evaluated numerically using function gamma, but one could expect that this implicit R-function has a more stable and faster algorithm than the general function integrate.

### 1.5.2 Maximum likelihood

In developing the distribution theory, the probability of getting a value of random variable between specified values is expressed with a distribution function, that is specified through parameters. Those parameters are thought of as fixed constants that specify the distribution of the random variable in the population we are interested in. In reality, we observe realizations of random variable, and the interesting question is often “What values do the parameters of the underlying distribution have”. This leads to change in the role of parameters and variables.
1.5. FITTING DISTRIBUTION FUNCTIONS TO DATA

Treating the distribution function as a function of its parameters for the sample we have in hands leads to the definition of likelihood. Assume that observations \( x = (x_1, x_2, \ldots, x_n) \) have been observed, and the joint distribution function of those observations is \( f(x|\theta) \), where vector \( \theta \) includes the parameters that specify the joint distribution function. If the observations are independent, the joint pdf is just the product of univariate densities: \( f(x|\theta) = \prod_{i=1}^{n} f(x_i|\theta) \). The likelihood function is the joint pdf for the fixed \( x \) as a function of \( \theta = (\theta_1, \ldots, \theta_k)' \)

\[
L(\theta|x) = f(x|\theta) \tag{1.44}
\]

If the components of \( x \) are independent, the likelihood becomes

\[
L(\theta|x) = \prod_{i=1}^{n} f(x_i|\theta) = \prod_{i=1}^{n} f(x_i|\theta_1, \theta_2, \ldots, \theta_k)
\]

The value of likelihood cannot be interpreted as a probability, even though it is somehow related to it. However, it can be used to compare two parameter values against each other. If we observe that \( L(\theta_1|x) > L(\theta_2|x) \), we could say that \( \theta_2 \) is more likely the value of the parameter of the underlying population than \( \theta_2 \). This deduction leads to the principle of maximum likelihood in estimating the parameters of an underlying distribution of an assumed functional form.

As the likelihood itself has no clear meaning except for being a tool for comparing two estimates, it does not have any effect if we make an increasing transformation to the likelihood. In many situations, the computations get simpler if we use logarithmic likelihood instead of the likelihood. The log-likelihood is defined as

\[
l(\theta|x) = \ln(L(\theta|x)) = \ln(f(x|\theta))
\]

Especially, with independent observations, minimizing the product of densities is equivalent to minimizing the sum of logarithmic densities

\[
l(\theta|x) = \ln \left( \prod_{i=1}^{n} f(x_i|\theta) \right) = \sum_{i=1}^{n} \ln(f(x_i|\theta_1, \theta_2, \ldots, \theta_k))
\]

The maximum likelihood estimator \( \hat{\theta} = \hat{\theta}(x) \) is the value of \( \theta \) that maximizes the likelihood 1.44. If the likelihood function is differentiable with respect to the components of \( \theta \), then the \( k \) likelihood equations are formulated as

\[
\frac{\partial}{\partial \theta_k} l(\theta|x) = 0 \quad k = 1, \ldots, k
\]

which yields systems of \( k \) equations. The solutions for that system of equations gives candidates of likelihood estimates. Those candidates may be global minima or maxima,
local minima or maxima, or inflection points. Other candidates are at the boundary of the parameter space. Of these candidates, the maximum likelihood estimate (ML estimate) is the one that gives the global maximum value of likelihood within the parameter space. The ML estimator is the expression that yields the ML estimate when evaluated using the data, \(x\).

One important feature of the ML estimator is the invariance property. Let \(\hat{\theta}\) be the MLE of \(\theta\), and \(\tau(\theta)\) any function. The invariance means that the MLE of \(\tau(\theta)\) is \(\tau(\hat{\theta})\).

There are no guarantee that the ML estimates would be unbiased or best in any sense. However, with certain distributions, there are results that show they are unbiased minimum variance estimators. Especially, these hold with distributions of the exponential family. In addition, if the first and second derivatives of the log-likelihood exist, and the Fisher information matrix is not zero, the ML estimators are asymptotically normal, minimum-variance unbiased estimators. The term asymptotically means that these results are true with large samples, whereas they may be badly violated with small samples. The asymptotic variance of ML estimate is that defined by the Crao-Ramer lower bound:

\[
\text{var}(\hat{\theta}) = \left\{ - \mathbb{E} \left[ \frac{\partial^2 l(\theta|x)}{\partial \theta^2} \right] \right\}^{-1}
\]

where \(\mathbb{E} \left[ \frac{\partial^2 l(\theta|x)}{\partial \theta^2} \right]\) is the Fisher information matrix. In applications, the variance is estimated by replacing \(\theta\) with ML estimates \(\hat{\theta}\).

**Example 1.42.** Assume that \(x = x_1, x_2, \ldots, x_n\) are observations from normal distribution with variance 1 and mean \(\mu\). The log likelihood function with respect to unknown \(\mu\) is

\[
l(\mu|x) = \sum_{i=1}^{n} \left[ \ln \left( \frac{1}{\sqrt{2\pi}} \right) \cdot \frac{1}{2} (x_i - \mu)^2 \right]
\]

\[
= n \ln \left( \frac{1}{\sqrt{2\pi}} \right) - \frac{1}{2} \sum_{i=1}^{n} (x_i - \mu)^2
\]

Differentiating the log likelihood with respect to \(\mu\) and setting equal to 0 gives the
likelihood equation

\[-\frac{1}{2} \sum_{i=1}^{n} 2(-1)(x_i - \mu) = 0\]

\[\sum_{i=1}^{n} x_i - n\mu = 0\]

\[n\mu = \sum_{i=1}^{n} x_i\]

\[\mu = \frac{1}{n} \sum_{i=1}^{n} x_i = \bar{x}\]

Thus, the sample mean is a candidate for MLE, and the only solution to the ML equation. To verify that it is maximum, not a minimum, we need to study the second derivate of the log likelihood, \(\frac{\partial}{\partial \mu} l(\mu, x) = \frac{\partial}{\partial \mu} \sum_{i=1}^{n} x_i - n\mu = -n\). The second derivative is a negative constant, so the first derivative is decreasing for \(-\infty < \mu < \infty\). Thus, the first derivative has to be positive for \(\mu < \bar{x}\) and negative for \(\mu > \bar{x}\), which implies that \(\mu = \bar{x}\) is a maximum. Furthermore, since the first derivative is continuous and the solution is unique, it is also a global maximum. Thus, we deduce that being within the domain of the parameter value, the global maximum \(\bar{x}\) is the ML-estimate for \(\mu\). Had we a restriction for \(\mu\) to be positive, then the ML estimator would be \(\mu = \bar{x}\) if \(\bar{x} > 0\) and \(\mu = 0\) if \(\bar{x} \leq 0\).

As shown above, the second derivative called Fishers information is \(\frac{\partial^{2} l(\mu|x)}{\partial \mu^{2}} = -n\). Writing this to the Rao-Cramer lower bound gives

\[\text{var}(\hat{\mu}) = \frac{1}{-E(-n)} = 1/n.\]

Thus, the standard error of estimate is \(1/\sqrt{n}\). It is a special case of the well-known standard error of mean \(\sigma^{2}/\sqrt{n}\). Even though we will not show it here, we conclude by saying that the standard error of ML-estimate for a population with other variance than 1 would be just \(\sigma/\sqrt{n}\).

**Example 1.43.** A lazy forester does not much care on exact results, but uses numerical easy-to-use algorithms always for maximum likelihood estimation instead. Even though this approach should not be suggested for a true scientist, at least for such a simple distribution as normal, we do that for demonstration purposes.

Thus, we have samples from normal distribution below, with known variance of 1 and unknown mean.

```r
> library(stats4)
> y10<-c(4.99, 4.42, 5.95, 4.49, 5.75, 3.92, 7.74, 5.77, 4.75, 5.61)
> y20<-c(5.80, 6.00, 5.35, 3.98, 4.99, 5.34, 5.14, 5.36, 3.19, 4.62,
+ 4.57, 3.23, 5.57, 4.18, 4.66, 8.76, 5.83, 4.74, 6.91, 4.44)
```
CHAPTER 1. RANDOM VARIABLES

-160 -140 -120 -100 -80 -60 -40 -20

mu

log likelihood

log likelihood

0 2 4 6 8 10

mu

Figure 1.13: The normal likelihood based on the data of 10 observations (upper) and that based on 20 observations (lower). The vertical lines show the ML estimates.

Next, we define the log likelihoods as the sum over logarithmic densities. The likelihood is a function of $\mu$. The mle package needs the negative log likelihood, so that is why we return value $-l$.

```r
> # define a function giving the negative log likelihood for y10
> nll10<-function(mu) {
+ l<-sum(log(dnorm(y10,mu)))
+ -l
+ }
> # The same function for the larger sample
> nll20<-function(mu) {
+ l<-sum(log(dnorm(y20,mu)))
+ -l
+ }

For demonstration purposes, we plot the likelihoods for both datasets. These plots are shown in Figure 1.13. We see that the likelihood based on larger dataset is narrower and has a sharper peak than the likelihood based on the smaller dataset.

```r
> mu<-seq(0,10,0.1)
> plot(mu,-sapply(mu,function(x) nll10(x)),type="l",xlab="mu",ylab="log likelihood")
> lines(mu,-sapply(mu,function(x) nll20(x)),type="l")

The ML estimates are obtained numerically using function mle.

```r
> sol10<-mle(minuslogl=nll10,start=list(mu=0))
> sol20<-mle(minuslogl=nll20,start=list(mu=0))
>
> summary(sol10)
Maximum likelihood estimation

Call:
1.5. FITTING DISTRIBUTION FUNCTIONS TO DATA

\begin{verbatim}
mle(minuslogl = nll10, start = list(mu = 0))

Coefficients:
             Estimate Std. Error
mu         5.339     0.3162278

-2 log L: 28.99266

> summary(sol10)
Maximum likelihood estimation

Call:
mle(minuslogl = nll120, start = list(mu = 0))

Coefficients:
             Estimate Std. Error
mu         5.033     0.2236068

-2 log L: 55.47056

> lines(rep(coef(sol10),2),c(-500,0))
> lines(rep(coef(sol20),2),c(-500,0))

We see that the estimates are 5.339 for the smaller data and 5.033 for larger data. The standard errors are 0.316 and 0.223, respectively. The vertical lines added to the plot of likelihoods show that they are at the maximums of the respective likelihoods. The higher sample size leads to more accurate estimates, as indicated by the standard errors. This can also be seen from the plot of likelihoods: standard errors are inversely related to the second derivative of the likelihood, and the second derivative is the higher the more peaked the likelihood is.

Finally, we use the exact results from the previous example to compute the MLE:s. The estimate is just the sample mean, and standard error is \(1/n\).

> mean(y10)
[1] 5.339
> sqrt(1/10)
[1] 0.3162278

> mean(y20)
[1] 5.033
> sqrt(1/20)
[1] 0.2236068

The numerically obtained estimates and standard errors are the same as the exact values up to the default reporting accuracy of R.

Casella and Berger (2002) presents a number of other examples on finding ML estimators for binomial and normal distribution under different assumptions. Below are two examples on finding ML-estimators in R.

Example 1.44. Assume that tree diameters follow \(\text{Weibull}(\alpha, \beta)\) distribution, and an independent sample of diameters \(x = (15.1, 17.1, 10.0, 13.5, 16.6, 18.1, 11.0, 13.4, 15.2, 17.1, 15.9, 17.7, 14.4, 13.8, 14.8, 11.9)\) has been taken from a stand. Find numerically ML-estimates for parameters \(\alpha\) and \(\beta\).

We use R-function \texttt{mle} in package \texttt{stats4}. For this purpose, we need to define a function that evaluates the negative log likelihood.

\end{verbatim}
library(stats4)

# the measured diameters in a vector
DBH<-c(15.1, 17.1, 10.0, 13.5, 16.6, 18.1, 11.0, 13.4, 15.2, 17.1, 15.9, 17.7, 14.4, 13.8, 14.8, 11.9, 13.0, 14.7, 16.5, 11.0)

# negative of Weibull log likelihood
nll<-function(shape,scale) {
  cat(shape,scale," 
  value<-sum(log(dweibull(DBH,shape,scale)))
  cat(value,"\n")
  value
}

# save ML-estimate into object 'solution'
solution<-mle(minuslogl=nll,start=list(shape=10,scale=10))

10 10 1806.286
10.001 10 1807.236
9.999 10 1805.337
10 10.001 1804.436
10 9.999 1808.139
-939.2394 1861.296 NaN
-179.8479 380.2591 NaN
-27.96957 84.05182 NaN
2.406085 24.81036 67.82072
2.407085 24.81036 67.82053
2.405085 24.81036 67.82059
2.406085 24.81136 67.8221
2.406085 24.80936 67.81933
...

Warning messages:
1: NaNs produced in: dweibull(x, shape, scale, log)
2: NaNs produced in: dweibull(x, shape, scale, log)
3: NaNs produced in: dweibull(x, shape, scale, log)
>
> # Take the summary and asymptotic variance-covariance matrix
> summary(solution)
> summary(solution)

Maximum likelihood estimation

Call: mle(minuslogl = nll, start = list(shape = 10, scale = 10))

Coefficients:

  Estimate Std. Error
shape 7.603536  1.3646050
scale 15.50216  0.4802482

-2 log L: 88.8954
>
> cov2cor(vcov(solution))

  shape    scale
shape 1.0000000 0.3144117
scale 0.3144117 1.0000000

# save the estimates for later use
> alphamle<-coef(solution)[1]
> betatamle<-coef(solution)[2]

The above output shows that the ML estimates for shape and scale are \( \hat{\alpha} = 7.60 \) and \( \hat{\beta} = 15.50 \), with estimation errors of 1.36 and 0.48, respectively. The correlation of estimation errors is 0.314.

The estimation gave three warnings. To see the causes for these warnings, the function giving the negative log likelihood was set to print the evaluated values of \( \alpha \).
1.5. FITTING DISTRIBUTION FUNCTIONS TO DATA

and $\beta$, as well as the value of the negative log likelihood onto the screen using function `cat()`. We see that for three cases, the Negative likelihood got the value NaN (Not a number). This resulted from that the algorithm used in finding estimators tried to evaluate the likelihood with a negative value of shape parameter. However, the shape can only be positive, and function `dweibull()` returns value NaN for negative values of shape. However, the algorithm did not fail to converge for this problem.

A link function can be used to ensure that only values within the parameter range would be evaluated. Defining $\alpha = \exp(\theta_1)$ and $\beta = \exp(\theta_2)$, ensures that $\alpha$ and $\beta$ are always positive, just because the exponential function is defined for any real-valued $x$, and it gives only values above zero. Thus, we are applying link function $\tau(\theta) = \ln(\theta)$ The invariance property of the MLE implies that such approach should lead to same point estimates than the previous approach.

```r
nll2<-function(theta1,theta2) {
  shape<-exp(theta1)
  scale<-exp(theta2)
  value<--sum(log(dweibull(DBH,shape,scale)))
  cat(shape,scale,value,"
  value
}
```

# save ML-estimate into object 'solution'
solution2<-mle(minuslogl=nll,start=list(theta1=log(10),theta2=log(10)))

```
10 10 1806.286
10.01001 10 1815.809
9.990005 10 1796.824
10 10.01001 1787.867
10 9.990005 1824.893
0 Inf NaN
0 Inf NaN
1.256009e-164 Inf NaN
```

There were 14 warnings (use warnings() to see them)

```
exp(coef(solution2))
theta1 theta2
7.603944 15.502181
```

We see that the evaluation still results warnings, because the algorithm evaluates the likelihood with ultimately high or small positive values. The resulting point estimates are equal to the one of the previous approach up to three significant digits. The small differences in the estimates arise from the numerical accuracy of the algorithm.

**Example 1.45.** In Example 1.15, we derived the distribution of canopy height observations, assuming that the crown shape seen from the side is an ellipsoid centered at $(x_0, y_0)$. Assume that we have observed the canopy height at random points within the crown with sampling density 4 observations per m$^2$ from within a tree crown using laser scanner. We know that the tree height is necessarily larger than equal to the maximum observed height, where the equality corresponds to the improbable situation that one of the observations has hit the tree top. Assuming that the crown shape can be well described with the ellipsoid, we want to estimate the crown shape, and especially, the tree height. We proceed by fitting the density of canopy height observations, by maximizing a profile likelihood, where one of the parameters is profiled out of the
CHAPTER 1. RANDOM VARIABLES

likelihood using the information on the sampling density. We first reparameterize the
density (1.7) using the maximum crown radius, i.e., the relative radius (as a fragment
of total height) at relative height \( y_0 \) by defining \( \text{maxr} = y_0 + b \). Then, based on the
assumption about circular shape of the cross-sectional crown, the maximum crown ra-
dius can be solved from equation \( n/4 = \pi \times (h \times \text{maxr})^2 \) as \( \text{maxr} = \frac{1}{2 \pi} \sqrt{\frac{n}{\pi}} \), where
\( n \) is the number of laser observations we have on the tree crown. Thus we can eli-
minate \( b \) from our density, resulting in that the likelihood has only three parameters to be
estimated, namely \( h, x_0 \), and \( y_0 \). The likelihood is defined in the R-code below.

```r
# Read the data
> thistree<-read.table("c:/laurim/biometria/lasertree.txt",header=TRUE)
# save laser heights to x
> x<-thistree$z
# initial guesses for the three parameters
> hinit<-max(x+0.01)
> x0init<-min(x-0.01)/hinit
# the profile likelihood
> minuslogl<-function(h=hinit,x0=x0init,y0=-0.015) {
  + maxr<-sqrt(length(x)/(4*pi))/h
  + -sum(log(pdf.laser(x,maxr-y0,x0,y0,h)))
+ }
```

The assumed model was fitted to data. Note that we have constraints for the pa-
rameters: \( h, x_0 \) should be less than the lowest laser observation, \( h \) should be greater
than the highest laser observation, and \( y_0 \) should be below zero. These constraints
could be implemented using log links, as we did in an earlier example on Weibull
distribution. However, this time we use constrained optimization in finding the param-
eter estimation, by using algorithm L-BFGS-B in estimation. The simple bounds for
the parameters are given using parameters \( \text{lower} \) and \( \text{upper} \). We need to use slightly
lower maximum for \( x_0 \) than \( \text{min} \times x \) and a slightly higher minimum for \( h \) than \( \text{max} \times x \),
because equality would lead to infinite profile likelihood. The summary of fit is given
below.

```r
> fit<-mle(minuslogl,method="L-BFGS-B",lower=c(hinit,-Inf,-Inf),upper=c(Inf,x0init,0))
> summary(fit)
```

The estimated tree height was 15.16 meters, being 0.17 meters higher than the
maximum of laser observations, 14.99 meters. The standard error of estimate was 0.15
meters. The upper plot of figure 1.14 shows a histogram of the observations and the
fitted distribution. The lower plot shows the corresponding crown shape. The utilized
functions \( \text{pdf.laser} \) and \( \text{radius.ellipse} \) were defined in example 1.15.
Figure 1.14: Illustration of example 1.45
1.5.3 Other methods

The principle of Maximum likelihood is the best justified approach for finding point estimates for unknown parameters. Especially, it is the only method that also provides tools to assess the accuracy of the obtained estimates.

The method of percentiles is analogous to the method of moments, but it utilizes sample percentiles, which are set equal to the theoretical quantiles of the assumed distribution function. For example, Bailey and Dell (1973) used, in addition to the ML, the percentile-based method because of its simplicity. They also presented closed-form solutions for the percentile equations.

Example 1.46. According to Bailey and Dell (1973), Dubey (1967) showed that the 17th and 97th percentiles are asymptotically best for estimating the shape parameter with no prior knowledge on the scale parameter. On the other hand, the 40th and 82th percentiles should be best for the scale. In this example, we use these two percentiles to estimate both the shape and scale parameters.

In our data of 20 tree diameters, these quantiles get the values of 14.16 cm and 16.89 cm. The estimation is carried out numerically by using function nlm. A very similar approach was taken previously in example 1.40. The estimates are different from both the MLE estimates of example 1.44 and from the moment-based estimates of example 1.40.

The R-code for Example 1.46

```r
> probs<-c(0.40,0.82)
> quants<quantile(DBH,probs)
> quants
40% 82%
14.16 16.89
> # sum of squared differences (q40-q40(theta))^2+(q82-q82(theta))^2;
> # gets value of 0 if the system of equations has a solution.
> fn<-function(theta,probs,quants) {
+ alpha<-theta[1]
+ beta<-theta[2]
+ sum((quants-qweibull(probs,alpha,beta))^2)
+ }
>
> solution<-nlm(fn,p=c(10,15),probs=probs,quants=quants)
Warning messages:
1: In qweibull(p, shape, scale, lower.tail, log.p) : NaNs produced
2: In nlm(fn, p = c(10, 15), probs = probs, quants = quants) :
   NA/Inf replaced by maximum positive value
> solution
$minimum
[1] 1.245229e-10
$estimate
[1] 6.86907 15.61467
$gradient
[1] -2.730390e-09 -2.619079e-08
$code
[1] 1
$iterations
```
1.5. FITTING DISTRIBUTION FUNCTIONS TO DATA

Histogram of \( x \)

![Histogram of x](image)

Figure 1.15: Histogram of tree diameters and the graph of estimated Weibull density obtained using MLE (solid) the method of moments (dashed), the method of percentiles (dotdash), and the method of cdf regression.

\[ \text{\texttt{hist(x,freq=FALSE,xlim=c(5,25),ylim=c(0,0.25),breaks=seq(0,40,2))}} \]
\[ \text{\texttt{x0<seq(0,40,0.1)}} \]
\[ \text{\texttt{lines(x0,dweibull(x0,alphamle,betamle),lty="solid")}} \]
\[ \text{\texttt{lines(x0,dweibull(x0,alphamom,betamom),lty="dashed")}} \]
\[ \text{\texttt{lines(x0,dweibull(x0,alphapp,betapp),lty="dotdash")}} \]
\[ \text{\texttt{lines(x0,dweibull(x0,alphareg,betareg),lty="dotted",lwd=2)}} \]

The R-code for Figure 1.15

In addition to moments and percentiles, also other quantities can be used for finding equations for a system of \( k \) equations. For example, the values of basal area or the number of stems could be set equal to those derived from the assumed distribution. These approaches are discussed in more detail in Section ???. Methods based on percentiles or other stand characteristics are commonly called parameter recovery methods in the literature of diameter distributions. In some approaches, the number of
equations is higher than \( k \), and a subset of the equations is used for each parameter. In such case, the solution fulfills all the equations only in a rare special case.

Some studies have used the method of cdf regression to fit an assumed distribution function to tree diameter data. In that approach, the diameter observations are ordered, and the \( i \)th smallest observation of the sample of size \( n \) is interpreted as \( i/(n+1) \)th quantile of the distribution. The estimated values are obtained by fitting the cdf of the assumed distribution function to the data of diameters and the corresponding values of the empirical cdf. I cannot see any reason to favor this method over the theoretically better justified methods, such as the method of maximum likelihood. Some authors misleadingly call this method as the method of maximum likelihood, based on the utilized method of Maximum Likelihood (or Restricted Maximum Likelihood) in the fitting of the nonlinear regression model.

**Example 1.47.** The following code performs the cdf-regression for the dataset of previous examples. Again, we get estimates that differ from the estimates of the other applied methods. Figure 1.16 demonstrates the approach. The estimated Weibull densities of all the applied methods have been shown in Figure 1.15. The R-code for Example 1.47

```r
dord<-DBH[order(DBH)]
quants<-(1:length(dord))/(length(dord)+1)
solution<-nls(quants˜pweibull(dord,shape,scale),start=list(shape=10,scale=15))
solution
Nonlinear regression model
model: quants ~ pweibull(dord, shape, scale)
data: parent.frame()
shape scale
6.356 15.609
residual sum-of-squares: 0.01258
Number of iterations to convergence: 5
Achieved convergence tolerance: 7.974e-07
> alphareg<-coef(solution)[1]
betareg<-coef(solution)[2]
plot(dord,quants)
lines(x0,pweibull(x0,alphareg,betareg))
```

The bayesian approach is still one more method of finding estimators. In the bayesian approach, the parameters are treated as random variables. The prior belief of the value of a parameter are parameterized using a prior distribution, which is combined with the distribution of the data. This yields a posterior distribution of the parameter of interest. A point estimate of the parameter could be obtained as the mean of the posterior distribution. The posterior distribution can also be used to make more detailed inference on the parameter of interest, such as computing different confidence intervals.

### 1.6 Linear prediction

The general case from (Lappi et al. 2006).
Assume that a random vector $h$ of length $k$ can be divided into two parts

$$h = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}$$

where $h_1$ and $h_2$ are random vectors, or scalars in the special case that the length of random vector is 1. It is assumed that $E(h_1) = \mu_1$, $E(h_2) = \mu_2$, $\text{var}(h_1) = V_1$, $\text{var}(h_2) = V_2$, and $\text{cov}(h_1, h_2) = V_{12}$. Using the notation of McCulloch and Searle (2001, p. 247), this can be written as

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \sim \left[ \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} V_1 & V_{12} \\ V_{12}' & V_2 \end{pmatrix} \right]$$

Assume that we have observed the random vector $h_2$ and want to predict vector $h_1$. The Best Linear Predictor (BLP) of $h_1$ is

$$\text{BLP}(h_1) = \hat{h}_1 = \mu_1 + V_{12}V_2^{-1}(h_2 - \mu_2)$$ (1.45)

with a prediction variance of

$$\text{var}(\hat{h}_1 - h_1) = V_1 - V_{12}V_2^{-1}V_{12}'$$ (1.46)

(McCulloch and Searle 2001, p. 250). This result means that if the expectations and variance-covariance matrices of two random vectors are known and either one of them is observed, the other one can be predicted using (1.45). Furthermore, the variance of the prediction error can be calculated using Equation (1.46).
If \( h \) follows the multinormal distribution, i.e.,
\[
\begin{pmatrix}
    h_1 \\
    h_2
\end{pmatrix}
\sim
N_k
\left[
\begin{pmatrix}
    \mu_1 \\
    \mu_2
\end{pmatrix},
\begin{pmatrix}
    V_{11} & V_{12} \\
    V_{12} & V_{22}
\end{pmatrix}
\right],
\]
then BLP is the Best Predictor. This results from that with multivariate normal distribution, all covariances are linear, implying that no nonlinear predictor can be better than the best linear one.

In practice, matrices \( V_1 \), \( V_2 \), and \( V_{12} \) as well as vectors \( \mu_1 \) and \( \mu_2 \) are unknown, and will be replaced with their estimates. The resulting predictor is the Estimated Best Linear Predictor (EBLP).

**Example 1.48.** Assume that a model system is available for prediction of tree volume and tree height using tree diameter only. Assuming that the form of the model system is correct and the parameters do not include errors, such a model system would give the expected values for height and volume for trees with known diameter. Furthermore, the residuals of the model are assumed to have constant variances so that the residual variance of height would be \( V_2 = 0.5^2 \) and that of volume would be \( V_1 = 2^2 \), and the covariance between them is \( V_{12} = 0.5 \).

Assume that a sample tree has been measured for diameter and height, the observed height being \( h_2 = 14 \) m. The volume model gives the expected volume as \( \mu'_1 = 70 \) liters, and the height model the expected height as \( \mu'_2 = 13 \) meters. In this case, both vectors are of length 1. However, we write them using boldface font in order to avoid introduction of new notations, and to show that they can just be treated as vectors of length one. The BLP is
\[
\hat{h}_1 = \mu + V_{12} V_2^{-1} (h_2 - \mu_2)
\]
\[
= 70 + 0.5 \times 0.25^{-1} (14 - 13)
\]
\[
= 70 + 2 \times 1 = 72 \text{dm}^3
\]

Further examples on linear prediction will be given when dealing the linear mixed models and models systems. With mixed-effects models, the BLP is used to predict stand effects using observations of the response. For example, we can predict stand effects of a height-diameter curve using measured heights and diameters from the stand of interest, resulting into a local models for that particular stand. With models systems, the BLP could be utilized to carry information from one model to another, as we did in the above example.

### 1.7 Exercises

1. Tree height (H) from a forest stand were measured in one meter classes and tree diameter (D) in 2 cm classes. The following table shows the observed joint dis-
1.7. EXERCISES

(a) Compute the marginal distributions of tree diameter and height.

(b) What is the probability that the diameter of a tree falls between 2 and 10 cm?

(c) What is the probability that the height of a tree falls between 4 and 10 meters?

(d) What is the probability that the diameter of a tree falls between 2 and 10 cm and height falls between 4 and 10 meters?

(e) Compute the conditional distribution of height when diameter is 6-8 cm.

(f) Compute the conditional distribution of diameter when tree height is 3-4 meters.

(g) Compute the conditional distribution of height when tree diameter is 4-10 cm.

2. It can be shown that if trees in a stand have completely random locations, then the number of trees within a plot of area |A| is distributed according to Poisson(λ|A|) distribution. It is a discrete distribution and has

3. Simulate square 20 by 20 meter sample plot from a forest stand, where the stand density is 1000 trees per ha, trees are located randomly, and tree diameter follows Weibull(5, 15) distribution as follows

(a) Generate the realized number of sample trees for the plot from Poisson(λA) distribution, where λ is stand density and A is the plot area.

(b) Generate x and y coordinates for tree locations from the Uniform distribution.

(c) Generate tree diameters from the Weibull distribution.

(d) Plot the trees using plot(x, y, cex), where x and y are coordinates and cex is the size of symbol which is proportional to tree diameter.
(e) Fit Weibull distribution to the simulated data using the method of maximum likelihood.

4. Simulate samples of sizes 100, 1000, 10000 and 100000 from the logit-logistic distribution using parameter values $\lambda = 20, \sigma = 0.622, \psi = 0, \phi = -1$.

5. Plot the distribution of tree heights when the diameter distribution is the logit-logistic distribution with $\lambda = 20, \sigma = 0.622, \psi = 0, \phi = -1$ and the H-D curve is $h = 8d^{0.3}$.

6. In a forest stand, tree joint distribution of tree diameter and height is characterized with the Multinormal distribution with $E\begin{bmatrix} D \\ H \end{bmatrix} = \begin{bmatrix} 20 \\ 18 \end{bmatrix}$ and $\text{cov} \begin{bmatrix} D \\ H \end{bmatrix} = \begin{bmatrix} 9 & 3 \\ 3 & 4 \end{bmatrix}$.

(a) Predict tree heights for a tree with known diameter of 10 cm using BLP.

(b) List the properties of your estimate assuming that the multivariate normality holds.

(c) Plot the implicitly assumed H-D curve. Hint, predict heights for several diameters with regular intervals and plot the results.

(d) Write down the mathematical expression of the assumed H-D relationship.

7. Using equation on equation (1.23), show that covariance is a special case of variance.

8. Derive the expected value and variance of the uniform distribution.

9. Let $Y$ have the Binomial($n, p$) distribution. Using rules (1.19) and (1.28), show that $E(Y) = np$ and $\text{var}(Y) = np(1-p)$.

10. Let $X$ have Poisson($\lambda$) distribution. Using (1.15) and (1.21), show that $E(X) = \text{var}(X) = \lambda$.

11. Show that if $X$ follows the two-parameter version of beta distribution, then $Y = a + (b - a)X$ follows the four-parametric version of the beta distribution. Based on this relationship, derive the expected value and variance of $Y$.

12. The density of random variable $Y_{r,n}$, that is, the $r$th smallest tree in a sample of size $n$, from a population with underlying cdf and pdf of $F_Y(y)$ and $f_Y(y)$, respectively, is given by

$$f_{r,n}(y) = \frac{n!}{(r-1)!(n-r)!} f_Y(y) \left[ F_Y(y) \right]^{r-1} [1 - F_Y(y)]^{n-r}.$$
1.7. EXERCISES

Assume that trees have been sampled from a stand with the percentile-based
diameter distribution of example 1.7?

(a) What is the density of $Y_{r:n}$?

(b) Compute the expected value of the minimum diameter of a sample of 12
trees, $Y_{1:12}$.

(c) Compute the variance and standard deviation of $Y_{1:12}$.

(d) Interpret the computed values for expected value and standard deviation.

13. The joint density of $Y_{r1:n}$ and $Y_{r2:n}$ ($r1 < r2$), i.e., the $r_1$th smallest and $r_2$th
smallest trees in a sample of size $n$ from a population with underlying cdf and
pdf of $F_Y(y)$ and $f_Y(y)$, respectively, is given by

$$f(y_1, y_2) = \frac{n!}{(n-r_2)!} \frac{r_2!}{(r_2-r_1)!} \frac{r_1!}{(r_1-1)!} \frac{f_Y(y_1)}{n-r_2} [F_Y(y_1)]^{r_1-1} [F_Y(y_2) - F_Y(y_1)]^{r_2-r_1-1} [1 - F_Y(y)]^{n-r_2}.$$

Assume that the underlying population distribution is the percentile-based distri-
bution of example 1.7.

(a) Derive the joint density of $Y_{r1:n}$ and $Y_{r2:n}$.

(b) Derive the conditional density of $Y_{1:12}$ given that $Y_{2:12} = 10$.

(c) Compute the expected value $E(Y_{1:12} | Y_{2:12})$.

(d) Compute the expected values $E(Y_{1:12})$, $E(Y_{2:12})$, $E(Y_{1:12}^2)$, and $E(Y_{2:12}^2)$
using the results from the earlier exercise on a univariate order statistic.

(e) Using the previous results, compute $cov(Y_{1:12}, Y_{2:12})$ and $corr(Y_{1:12}, Y_{2:12})$.

14. Plot the density of the two-parameter beta distribution (use function $dbeta()$)
using (at least) the following values for parameters $\alpha$ and $\beta$. How do the values
affect to the shape of the distribution? Note that the $dbeta$ is defined only for
$0 \leq x \leq 1$

(a) $\alpha = 0.5$ and $\beta = 0.5$

(b) $\alpha = 1$ and $\beta = 1$

(c) $\alpha = 5$ and $\beta = 5$

(d) $\alpha = 10$ and $\beta = 10$

(e) $\alpha = 2$ and $\beta = 10$

(f) $\alpha = 10$ and $\beta = 2$
(g) \( \alpha = 1 \) and \( \beta = 10 \)

(h) \( \alpha = 0.5 \) and \( \beta = 10 \)

15. Define a function that specifies the distribution function and density for the four-parameter beta density.

(a) Define R-function for the cdf using the R-function `pbeta()` and equation (1.8). Plot the distribution function using some realistic values of the parameters, assuming that the random variable is tree diameter.

(b) Define R-function for the pdf using the R-function `dbeta()` and equation (1.11). Plot the density using some realistic values of the parameters, assuming that the random variable is tree diameter.

(c) Check that the area under the density is one as follows. Compute the value of the density using regularly placed diameters between \( d_{\text{min}} \) and \( d_{\text{max}} \). Multiply the mean of these values by \( d_{\text{max}} - d_{\text{min}} \). This value should be close to 1. (Why?)

16. Fit the beta distribution to the tree diameter data of Example 1.44 using the method of maximum likelihood.

(a) Define a likelihood that uses the four parameters and perform the fit. Why did the problem occur?

(b) Define the likelihood in terms of logarithmic \( \alpha, \beta \), minimum diameter and range. Perform the fit again. What was the reason for the problem now?

(c) Assume that the minimum and maximum diameters are \( d_{\text{min}} = 0.95 \text{min}(DBH) \) and \( d_{\text{max}} = 1.05 \text{max}(DBH) \), and define the likelihood in terms of the two shape parameters. Plot the histogram of the data and the fitted density.

(d) Try ML-fit of all parameters using \texttt{method = L-BFGS-B} in \texttt{mle} and give bounds for the parameter estimates.

17. Fit the beta distribution to the tree diameter data of Example 1.44 using the following methods.

(a) Method of moments assuming that \( d_{\text{min}} = 0.95 \text{min}(DBH) \) and \( d_{\text{max}} = 1.05 \text{max}(DBH) \).

(b) Method of percentiles (select the percentiles you want to use).

(c) Method of cdf regression.
18. Consider random variables $X_1$, $X_2$, and $X_3$, which have the following first and second order properties: $\mathbb{E}(X_1) = 5$, $\mathbb{E}(X_2) = 10$, $\mathbb{E}(X_3) = 11$, $\text{var}(X_1) = 1$, $\text{var}(X_2) = 2$, $\text{var}(X_3) = 3$, $\text{cov}(X_1, X_2) = 0.5$, $\text{cov}(X_1, X_3) = -1$, and $\text{cov}(X_2, X_3) = 0$. Let vectors $x$, $y$, and $z$ be $x = (X_1, X_2, X_3)'$, $y = (X_1, X_2)'$, and $z = (X_1, X_3)'$.

(a) Construct $\mathbb{E}(x)$ and $\text{var}(x)$.

(b) Compute $\text{cor}(x)$ without using function $\text{cov2cor}$.

(c) Compute the inverse of $\text{var}(x)$.

(d) Construct $\text{cov}(y, z)$.

(e) Construct $\text{cor}(y, z)$. Note that $\text{cov2cor}$ can be applied only for variance matrices.

(f) Compute $\text{var}(\mathbb{E}(x)'x)$. 
Chapter 2

Linear model

Linear regression, or the linear model, is maybe the most widely used statistical tool. It is used to analyze the dependence of variable $y$, e.g., tree volume, on other variables $x$, e.g., tree diameter and height. The terminology of regression modeling depends much on the field of study, and also on the purpose of the modeling. The different names used for variable $y$ include terms dependent variable, response variable, the regressand, the measured variable, the responding variable, the explained variable, and the outcome variable. For variable $x$, terms independent variable, predictor variable, regressor, controlled variable, manipulated variable, and explanatory variable are used. In the basic regression analysis, it is often assumed that $x$-variables are set by the researcher, and the effect of changes in $x$ on the value of $y$ is analyzed. However, in many cases, the values of $x$ cannot be controlled. This does not cause problems into the analysis, but the inference on the model is valid for the dataset used in analysis.

In the case of single predictor regression, we will have one response and one predictor. For example, we may predict tree height on tree diameter. In multiple regression, the number of predictors may be higher than 1. For example, Standing tree volume may be predicted on tree diameter, height and upper diameter. In most cases, the number of responses is one. However, we may also have several responses, for example, we may wish to simultaneously predict tree volume and height. In this situation, a system of models is fitted. Those situations are handled in chapter 6.

The regression may be either nonlinear or linear. In this context, the linearity means that variable $y$, or any transformation of it, is linear in predictors $x$ or any transformations of it. Thus, the linear regression applies in all instances where the relationship between $x_1, \ldots, x_n$ and $y$ can be written as

$$f_y(y) = \beta_0 + \beta_1 f_1(x_1) + \beta_2 f_2(x_1) + \ldots + \beta_n f_n(x_n),$$

where $\beta_1, \ldots, \beta_n$ are parameters to be estimated from the data.
In nonlinear regression, the relationship may be expressed with any function

\[ y = f(x_1, \ldots, x_n, \beta_1, \ldots, \beta_m). \]

For example, we could assume relationship

\[ y = \beta_0 x^{\beta_1}, \]

between \( x \) and \( y \), which is no more linear. However, the relationship could be linearized by taking logarithms from both sides to get

\[ \ln y = \ln \beta_0 + \beta_1 \ln(x). \] (2.1)

The above example shows a very common way to linearize nonlinear relationships, namely taking logarithms. However, in many cases the relationship cannot be linearized. The widely used and flexible Chapman-Richards function (Richards 1959) is an example of such a function

\[ y = \beta_1 \left(1 - e^{-\beta_2 x}\right)^{\beta_3}. \]

### 2.1 Single-predictor regression

#### 2.1.1 Model formulation

In single predictor regression, we have one response and one predictor. Assume that variables \( y_i \) and \( x_i \) are measured for individuals \( i, i = 1, \ldots, n \). The simple linear regression model for the \( i \)th individual can be written as

\[ y_i = b_0 + b_1 x_i + e_i. \] (2.2)

In equation (2.2), we assume that variable \( y_i \) comprises of two parts: of a systematic part that depends on the values of \( x_i \) through the assumed relationship, and of a random part caller residual or random error. Parameters \( b_0 \) and \( b_1 \) are parameters of the assumed regression model, which specify the systematic dependence on \( x_i \), and \( e_i \) is the residual of individual \( i \). Part \( b_0 + b_1 x_i \) expresses the average or mean relationship between variables \( x \) and \( y \). It can also be interpreted as the expected value of \( y \) for an individual with given value of \( x \). The residual \( e_i \) in turn expresses how much the value of \( y \) of the particular tree \( i \) differs from that expected value or mean in the horizontal direction. It is usually assumed that \( E(e_i = 0), \text{var}(e_i) = \sigma^2 \), and \( \text{cov}(e_i, e_j) = 0 \) when \( i \neq j \).

**Example 2.1.** 20 Trees were measured for diameter and height in a stand. The observations have been printed in table 2.1 plotted in figure 2.1. It seems a linear model of form

\[ h_i = b_0 + b_1 d_i + e_i \]
2.1. SINGLE-PREDICTOR REGRESSION

Table 2.1: Twenty measured diameter-height pairs from a Scots pine - Norway spruce mixture.

<table>
<thead>
<tr>
<th>d</th>
<th>h</th>
<th>pl</th>
<th>d</th>
<th>h</th>
<th>pl</th>
<th>d</th>
<th>h</th>
<th>pl</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>22.8</td>
<td>1</td>
<td>31.9</td>
<td>21.3</td>
<td>1</td>
<td>26.9</td>
<td>21.3</td>
<td>2</td>
</tr>
<tr>
<td>29.9</td>
<td>21.6</td>
<td>2</td>
<td>28.7</td>
<td>21.3</td>
<td>1</td>
<td>25.7</td>
<td>20.7</td>
<td>2</td>
</tr>
<tr>
<td>30</td>
<td>20.5</td>
<td>2</td>
<td>27.9</td>
<td>21.3</td>
<td>1</td>
<td>26.3</td>
<td>19.3</td>
<td>1</td>
</tr>
<tr>
<td>29.6</td>
<td>22.5</td>
<td>1</td>
<td>27.6</td>
<td>20.8</td>
<td>2</td>
<td>26.2</td>
<td>20.2</td>
<td>2</td>
</tr>
<tr>
<td>29.4</td>
<td>20.4</td>
<td>1</td>
<td>27.3</td>
<td>24</td>
<td>2</td>
<td>25.4</td>
<td>20.5</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 2.1: Twenty measured diameter-height pairs from a Scots pine - Norway spruce mixture.
could be a good starting point for modeling the height-diameter relationship in the example stand.

2.1.2 Estimation with least squares

In the method of least squares, values for parameter vector $b$ are estimated by searching the value that minimize the sum of squared residuals. With single predictor regression (2.2), the sum of squared residuals can be written as

$$
\sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2
$$

The values of $b_0$ and $b_1$ that minimize the squared residuals is found at a point where the first derivative is zero, given that that point is the global minimum. Differentiating the sum of squared residuals with respect to $b_0$ and $b_1$ (we believe that they are global minimums), and setting the derivates equal to 0 gives equations

$$
-2 \sum_{i=1}^{n} y_i + 2nb_0 - 2b_1 \sum_{i=1}^{n} x_i = 0
$$

$$
\sum_{i=1}^{n} 2y_i x_i - 2b_0 \sum_{i=1}^{n} x_i - 2b_1 \sum_{i=1}^{n} x_i^2 = 0
$$

Solving the upper equation for $b_0$ yields $\hat{b}_0 = \bar{y} - b_1 \bar{x}$. Writing the solution into the latter equation and solving for $b_1$ yields $\hat{b}_1 = \frac{\frac{1}{n} \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i y_i}{\frac{1}{n} \left( \sum_{i=1}^{n} x_i \right)^2 - \sum_{i=1}^{n} x_i^2}$. We use hats on $b_0$ and $b_1$ tho emphasize that the resulting numerical values are not true values of $b_0$ and $b_1$ in the population. Instead, they are just estimates based on our sample. The equations used for computing these estimates are estimators.

The residual variance is estimated by

$$
\hat{\sigma}^2 = \frac{RSS}{n-2},
$$

where $RSS = \sum_i (y_i - \hat{b}_0 - \hat{b}_1 x_i)^2$ is the residual sum of squares.

**Example 2.2.** In example 2.1, the required sums are $\sum_i d_i = 547.1$, $\sum_i h_i = 416$, $\sum_i d_i h_i = 11403.52$, and $\sum_i d_i^2 = 15035.59$. The OLS estimates for regression coefficients become

$$
\hat{b}_1 = \frac{\frac{1}{n} \sum_{i=1}^{n} d_i \sum_{i=1}^{n} h_i - \sum_{i=1}^{n} d_i h_i}{\frac{1}{n} \left( \sum_{i=1}^{n} d_i \right)^2 - \sum_{i=1}^{n} d_i^2}
= \frac{1/20 \times 547.1 \times 416 - 11403.52}{1/20 \times 547.1^2 - 15035.59} = 0.3421870
$$

and

$$
\hat{b}_0 = \bar{h} - b_1 \bar{d}
= 1/20 \times 416 - 0.3421870 \times 1/20 \times 547.1 = 11.43947
$$
2.1. SINGLE-PREDICTOR REGRESSION

The estimate of residual variance is

$$\hat{\sigma}^2 = \frac{\sum_i (h_i - \hat{b}_0 - \hat{b}_1 d_i)^2}{n - 2}$$

The above computations were carried out with the following code:

```r
> sumd<-sum(onestand$d)
> sumh<-sum(onestand$h)
> sumdh<-sum(onestand$d*onestand$h)
> sumd2<-sum(onestand$d^2)
> n<-dim(onestand)[1]
> b1<-(1/n*sumd*sumh-sumdh)/(1/n*sumd^2-sumd2)
> b1
[1] 0.3421870
> b0<-sumh/n-b1*sumd/n
> b0
[1] 11.43947
> RSS<-sum((onestand$h-b0-b1*onestand$d)^2)
> sigma2<-RSS/(n-2)
> sigma2
[1] 1.215681
> sqrt(sigma2)
[1] 1.102579
```

The same results are obtained using the R-function `lm`

```r
fm1<-lm(h~d,data=onestand)
fm1
```

A little bit more information on the model is obtained using function `summary()`

```r
> summary(fm1)
Call:
  lm(formula = h ~ d, data = onestand)
Residuals:
    Min      1Q  Median      3Q     Max
-1.42337 -1.02454 -0.01555  0.51366  3.21882
Coefficients:  
    Estimate Std. Error t value Pr(>|t|)
(Intercept)   11.4395    3.6219   3.158 0.00544 **
      d        0.3422    0.1321   2.590 0.01847 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 1.103 on 18 degrees of freedom
Multiple R-squared: 0.2716,    Adjusted R-squared: 0.2311
F-statistic: 6.71 on 1 and 18 DF,  p-value: 0.01847
```

At this stage, we just note that our manually computed figures agree with those given by function `lm`. The fitted line can be added to the plot using

```r
> abline(b0,b1) # try also abline(fm1)
```

Furthermore, we illustrate the residuals using vertical dashed lines

```r
> fn1<-function(x) lines(rep(onestand$d[x],2),c(onestand$h[x],predict(fm1)[x]),lty="dashed")
> sapply(1:dim(onestand)[1],fn1)
```
2.1.3 Modeling nonlinear relationships

If the relationship between the response and predictors is not linear, then something else needs to be done than just fitting. The alternatives are

1. to make transformations into the predictors,
2. to make transformations into the response, and
3. to switch to nonlinear regression.

The first two options are most commonly considered, as they retain the model as linear. Then the well-developed theory of linear models remains to be applicable in model fitting and the following inference. However, also the nonlinear regression is becoming more and more applicable as the theory has developed and currently applied software include good capabilities for model-fitting. However, during this course, we consider only the linear modeling approaches.

When selecting whether to make transformations into predictors or response, it is important to recognize that making transformations to response means that also predictions is carried out in the same scale. This results that the residuals in the original scale do not have mean of 0, and the model is biased in the original scale (because $g(E(X)) \neq E(g(X))$, see example 1.18). On the other hand, making transformations to the predictors retains unbiasedness for the response. Thus, if unbiasedness of the predictions is important, then this issue should be taken into account. Alternatives are
Figure 2.3: The plot of tree height against diameter in the data of example 2.3.

then (i) transforming the predictor(s), (ii) transforming the response and applying a bias correction and (iii) switching to the nonlinear regression. These alternatives are considered in the following example.

**Example 2.3.** A total of 200 Scots pine trees were measured for diameter and height from 44 plots in North Carelia. The data were read to R using:

```r
> hddata<-read.table("d:/laurim/biometria/hddata.txt",header=TRUE)
```

To see if the relationship between diameter and height is linear, we start by plotting the data. We also fit a simple model and add it onto the plot (the narrow straight line in the upper graph) to see if the assumption on linear relationship would be realistic.

```r
> windows(width=2.5,height=6)
> par(mfcol=c(3,1),mai=c(0.6,0.5,0.1,0.1),mgp=c(2,0.7,0),cex=0.8)
> plot(hddata$d,hddata$h)
> # Fit the model
> fm1<-lm(h~d,data=hddata)
> summary(fm1)
```

```plaintext
Call:
  lm(formula = h ~ d, data = hddata)

Residuals:
    Min     1Q  Median     3Q    Max
  -5.740  -1.856  -0.199  1.944  5.847

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)   4.87241    0.38501   12.65   <2e-16 ***
 d            0.55180    0.02176   25.36   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.533 on 198 degrees of freedom
Multiple R-Squared: 0.7646,  Adjusted R-squared: 0.7634
F-statistic: 643 on 1 and 198 DF,  p-value: < 2.2e-16
```

```r
> # Compute the fitted values, tru also fitted(fm1) or predict(fm1)
```

Figure 2.3: The plot of tree height against diameter in the data of example 2.3.
The plot shows slight curvature. However, we see that the mean of residuals is close to 0, which indicates that the fitted values are unbiased over the whole data, i.e., they are marginally unbiased. However, the observed curvature indicates that they are not conditionally unbiased, e.g., for small diameters (say, d < 8 cm), the model seems to overestimate height.

Plotting the logarithmic heights against logarithmic diameters shows that model \(2.1\), which is based on the power function, could fit the data better (see the middle plot of 2.3). The model is fitted in R using code

```r
> # The middle plot
> plot(log(hddata$d),log(hddata$h))
> # fit the model
> fm2<-lm(log(h) ~ log(d),data=hddata)
> summary(fm2)
```

```
Call: lm(formula = log(h) ~ log(d), data = hddata)

Residuals:
     Min       1Q   Median       3Q      Max
-0.654895 -0.112837  0.007841  0.155181  0.404031

Coefficients:                     Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.83338     0.06424   12.97   <2e-16 ***
log(d)             0.64895     0.02409   26.93   <2e-16 ***
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1922 on 198 degrees of freedom
Multiple R-Squared: 0.7856, Adjusted R-squared: 0.7845
F-statistic: 725.4 on 1 and 198 DF, p-value: < 2.2e-16
```

However, the fit is unbiased only for logarithmic heights. For the heights in meters, the estimated bias is 0.22, as seen from the computations below. This means that the model underestimates tree heights in the modeling data, on average, by 0.22 meters.

```r
> # compute fitted values
> fit2<-coef(fm2)[1]+coef(fm2)[2]*log(hddata$d)
> # mean of residuals
> mean(log(hddata$h)-fit2)
[1] -4.607309e-16
> # mean of residuals in meters
> mean(hddata$h-exp(fit2))
[1] 0.2168802
```

The obtained result arises from the difference in the means of lognormal and normal distributions. It was discussed in Section 1.4.2 when the lognormal distribution was introduced. That discussion also gives one possible solution to the problem. It was shown that the expected value of the lognormal distribution is \(e^{\mu+\sigma^2/2}\). Based on this result, a correction for bias could be performed by adding half of the residual variance into the logarithmic predictions before exponential transformation. However, this implies that we rely on the assumption of normality on residuals, which would not be otherwise needed. However, the same bias correction could be derived also based on Taylor series (Lappi 1991). The predictions corrected for bias are computed below.
2.1. SINGLE-PREDICTOR REGRESSION

We notice that the mean of residuals was clearly reduced in absolute value. However, it is still higher than the mean of residuals when the response was height instead of logarithmic height.

Another option for linearizing the relationship is to look for such transformation into diameter that makes the relationship on height linear. After some trials, we find that the relationship between $\ln(d + 10)$ and $h$ is quite close to linear (The lowest plot in Figure 2.3). Thus, a starting point for our modeling purposes could be the following simple linear model

$$h_i = b_0 + b_1 \ln(d_i + 10) + e_i$$

where $h_i$, $d_i$, and $e_i$ are the height, diameter and residual of tree $i$, $i = 1, \ldots, 200$. The residual is assumed to have expectation of zero, i.e., the average difference between observed height and the fitted model is assumed to be zero. Furthermore, it is assumed that the variance of true height around the regression line is constant, i.e., it does not vary according to the predicted height. The code below makes the plot, fits the model and computes the mean of residuals.

```r
> plot(log(hddata$d+10), hddata$h)
> fm3<-lm(h~log(d+10), data=hddata)
> summary(fm3)
Call:
lm(formula = h ~ log(d + 10), data = hddata)
Residuals:
     Min       1Q   Median       3Q      Max
-6.07213 -1.45458  0.02429  1.70491  5.28456
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -33.7831   1.8300  -18.46   <2e-16 ***
log(d + 10)  14.7956   0.5698   25.97   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 2.487 on 198 degrees of freedom
Multiple R-Squared: 0.773,  Adjusted R-squared: 0.7718
F-statistic: 674.2 on 1 and 198 DF,  p-value: < 2.2e-16
> abline(fm3)
> fit3<-coef(fm3)[1]+coef(fm3)[2]*log(hddata$d+10)
> mean(hddata$h-fit3)
```

We notice that the mean of residuals in the original scale are now close to 0. However, this approach could be criticized for that there is actually one additional parameter that was estimated to be 10 by a trial-and-error algorithm.

Even though nonlinear regression is out of scope of this chapter, we finally show an example how the power function could be fitted into the data using nonlinear least squares. We first define a R-function for power equation and then fit the model. In NLS, we also need to give initial guesses for the parameter estimates. Good guesses are those obtained from the linearized version of the model (2.1). Note that coefficient $\beta_0$ is obtained from the constant of model fm2 by using the exponential transformation.
Figure 2.4: Illustration of the effect of transformation in the predictor.
2.1. SINGLE-PREDICTOR REGRESSION

> # define the nonlinear function
> powerf<-function(x,a,b) {
+   a*x^b
+ }
> # use nls for nonlinear least squares
> fm4<-nls(h~powerf(d,a,b),
+ data=hddata,
+ start=list(a=exp(coef(fm2))[1],b=coef(fm2)[2]))
> summary(fm4)

Formula: h ~ powerf(d, a, b)

Parameters:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| a 2.41964| 0.18571    | 13.03   | <2e-16 ***|
| b 0.63680| 0.02571    | 24.77   | <2e-16 ***|

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.462 on 198 degrees of freedom

> # compute fitted values (try also fitted(fm4))
> fit4<-powerf(hddata$d,coef(fm4)[1],coef(fm4)[2])
> # mean of residuals
> mean(hddata$h-fit4)

[1] -0.001575283

The the marginal unbiasedness is guaranteed because the squared residuals are computed in the original scale. Thus, the mean of residuals is quite close to 0.

The graphs of models fm2, fm3 and fm4 were added to the upper graph of figure 2.3 using thick dashed gray, dotted black, and solid black lines, respectively.

> lines(dapu,powerf(dapu,coef(fm4)[1],coef(fm4)[2]),lwd=2)
> lines(dapu,powerf(dapu,exp(coef(fm2)[1]),coef(fm2)[2]),lwd=2,col="gray",lty="dashed")
> lines(dapu,coef(fm3)[1]+coef(fm3)[2]*log(dapu+10),lwd=2,lty="dotted")

Example 2.4. Exploring the systematic parts of the model (i.e., ignoring the residual), the previous example used the following expressions to compute the fitted values to the $H - D$ data

1. $h = b_0 + b_1 d$
2. $h = \exp(b_0 + b_1 \ln(d)) = \exp(b_0)d^{b_1}$
3. $h = \exp(b_0 + b_1 \ln(d) + \sigma^2/2) = \exp(b_0 + \sigma^2/2)d^{b_1}$
4. $h = b_0 + b_1 \ln(d+10)$
5. $h = b_1 d^{b_2}$

Models 1-3 and 5 are special cases of model $H = a + bD^c$

with the following parameter values

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.87</td>
<td>0.55</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2.30</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>2.53</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>2.42</td>
</tr>
</tbody>
</table>
The numerical values above show that even though models 2, 3 and 5 assume exactly the same mathematical relationship for the average dependence of $H$ on $D$, the numerical values are different. With models 2 and 3, the difference arises from the bias correction, whereas model 5 differs in that the residuals were minimized in different scale: model 2 minimized them for logarithmic height, whereas model 5 did that for metric height.

2.2 Multiple regression

2.2.1 Model formulation

The situation does not change very much from that of the previous subsection if we have more than one predictor. The linear model is

$$ y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + b_p x_{pi} + e_i. \quad (2.4) $$

Now coefficients $b_0, b_1, \ldots, b_p$ specify the expected dependence of $y_i$ on the predictors $x_{1i}, \ldots, x_{pi}$, or the systematic part of the model. Again, the residual $e_i$ in expresses how much the value of $y$ of the particular tree $i$ differs from that expected value or mean in the horizontal direction, the assumptions are the same as with single-predictor regression, namely that $E(e_i) = 0$, $\text{var}(e_i) = \sigma^2$, and $\text{cov}(e_i, e_j) = 0$ when $i \neq j$. Note that no specific distribution is assumed for $e_i$ at this stage. Assumptions $\text{var}(e_i) = \sigma^2$ and $\text{cov}(e_i, e_j) = 0$ can be relaxed so, that a specific structure may be assumed for variances and covariances. These issues are discussed later in chapters 3 and 2.1.2

**Example 2.5.** As an alternative to model of example 2.3, one could assume that tree height depends not only on tree diameter, but on some properties of the stand, such as the basal area, basal area median diameter, and geographical northern co-ordinate of the plot location. The assumed model would be

$$ h_i = b_0 + b_1 \ln(d_i + 10) + b_2 G_i + b_3 DGM_i + b_4 YK_i + e_i $$

where $G_i$, $DGM_i$, and $YK_i$ are the basal area ($m^2/ha$), basal area median diameter (cm), and the geographical north co-ordinate of $i$, $i = 1, \ldots, 200$. The model was fitted using the following code:

```r
> fm2 <- lm(h ~ d + G + Dg + Y, data = hddata)
> summary(fm2)
```

**Call:**

`lm(formula = h ~ d + G + Dg + Y, data = hddata)`

**Residuals:**

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>-5.6890</td>
<td>-1.4557</td>
<td>0.0954</td>
<td>1.5984</td>
<td>4.4351</td>
</tr>
</tbody>
</table>
2.2. MULTIPLE REGRESSION

Figure 2.5: The plot of tree height against diameter in the data of example 2.5.

Coefficients:

|            | Estimate | Std. Error | t value | Pr(>|t|) |
|------------|----------|------------|---------|----------|
| (Intercept)| 0.177824 | 0.829785   | 0.214   | 0.8305   |
| d          | 0.449815 | 0.022739   | 19.781  | <2e-16   ***|
| G          | 0.080571 | 0.041143   | 1.958   | 0.0516   . |
| Dg         | 0.097022 | 0.054801   | 1.770   | 0.0782   . |
| Y          | 0.009065 | 0.003426   | 2.646   | 0.0088   **|

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 1

Residual standard error: 2.182 on 195 degrees of freedom
Multiple R-Squared: 0.8279,  Adjusted R-squared: 0.8244
F-statistic: 234.5 on 4 and 195 DF,  p-value: < 2.2e-16

A practical difference to the single-predictor regression is immediately noticed, namely that the predictions are represented by points that do not form a line in the $h - d$-plane. Plot 2.5 shows the observed heights with open circles, and the fitted values of a multiple regression model with black triangles. The plot was produced using

\[
\text{plot(hddata$d,hddata$h,xlab="Diameter, cm",ylab="Height, m",ylim=c(3,30))}
\]

\[
\text{points(hddata$d,predict(fm2),pch=17)}
\]

\[
\text{fn1<-function(x) lines(rep(hddata$d[x],2),c(hddata$h[x],predict(fm2)[x]),lty="dashed")}
\]

\[
\text{sapply(1:dim(hddata)[1],fn1)}
\]

According to the type of predictors, the linear model can be called regression model, analysis of variance model, and analysis of covariance model. In the regression analysis model, predictors are usually continuous random variables, such as tree diameter, or altitude above sea level. In variance analysis (ANOVA) model, the predictors are so called dummy variables or indicator variables, which only can get values 0 and 1, indicating either presence or absence of a certain property or feature. They may indicate, for example, the origin of a seedling (e.g., 1 means planted and 0 natural), or if
a certain treatment was applied or not. For example, value 1 may indicate the new silvicultural treatment and value 0 the old treatment. Furthermore, indicator variables can be used to parameterize categorical variables, such as site fertility class, to a regression model. Classifications with more than two classes require several dummy variables. The covariance analysis model is a combination of these two models, including both continuous variables and indicator variables as predictors.

It is often convenient to write the linear model (2.4) in a matrix form. In the matrix form, all observations of $y$ are written into a column vector $y_{n×1}$, the predictors into matrix $X_{n×(p+1)}$, the coefficients into column vector $b_{p+1×1}$, and the residuals into a column vector $e_{n×1}$. The model becomes

$$
\begin{pmatrix}
y_1 \\
y_2 \\
\vdots \\
y_n
\end{pmatrix} = \begin{pmatrix}
1 & x_{11} & x_{21} & \cdots & x_{p1} \\
1 & x_{12} & x_{22} & \cdots & x_{p2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_{1n} & x_{2n} & \cdots & x_{pn}
\end{pmatrix} \begin{pmatrix}
b_0 \\
b_1 \\
\vdots \\
b_p
\end{pmatrix} + \begin{pmatrix}
e_1 \\
e_2 \\
\vdots \\
e_n
\end{pmatrix}
$$

or simply

$$y = Xb + e \quad (2.5)$$

Matrix $X$ is called the design matrix. Note that the first column of the design matrix has ones. This is to include the constant to the model, i.e, the constant of (2.4) can be written as $b_i \times 1$. The assumptions about the residual vector $e$ can be stated as

$$E(e) = 0$$
$$\text{var}(e) = \sigma^2 I$$

Furthermore, because the fixed part does not include randomness (design matrix $X$ and coefficients $b$ are thought as fixed), these assumptions imply that

$$E(y) = Xb$$
$$\text{var}(y) = \text{var}(e) = \sigma^2 I$$

Example 2.6. On one sample plot, 20 trees were measured for diameter and height. The data are shown in table 2.1. The simple linear model of Example 2.3 can be presented in a matrix form as

$$h = Xb + e,$$
2.2. MULTIPLE REGRESSION

\[ \begin{pmatrix} 22.8 \\ 21.6 \\ 20.5 \\ 22.5 \\ 20.4 \\ 19.7 \\ 21.3 \\ 21.3 \\ 20.8 \\ 24.0 \\ 21.3 \\ 20.7 \\ 19.3 \\ 20.2 \\ 20.5 \\ 19.3 \\ 20.3 \\ 21.0 \\ 19.2 \\ 19.3 \end{pmatrix} \]

\[ \begin{pmatrix} 1 & 31 \\ 1 & 29.9 \\ 1 & 30 \\ 1 & 29.6 \\ 1 & 29.4 \\ 1 & 28.3 \\ 1 & 28.7 \\ 1 & 27.9 \\ 1 & 27.6 \\ 1 & 27.3 \\ 1 & 26.9 \\ 1 & 25.7 \\ 1 & 26.3 \\ 1 & 26.2 \\ 1 & 25.4 \\ 1 & 26.2 \\ 1 & 25.3 \\ 1 & 25.5 \\ 1 & 25.6 \\ 1 & 24.3 \end{pmatrix} \]

\[ \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \\ e_{10} \\ e_{11} \\ e_{12} \\ e_{13} \\ e_{14} \\ e_{15} \\ e_{16} \\ e_{17} \\ e_{18} \\ e_{19} \\ e_{20} \end{pmatrix} \]

where

\[ h = \begin{pmatrix} 22.8 \\ 21.6 \\ 20.5 \\ 22.5 \\ 20.4 \\ 19.7 \\ 21.3 \\ 21.3 \\ 20.8 \\ 24.0 \\ 21.3 \\ 20.7 \\ 19.3 \\ 20.2 \\ 20.5 \\ 19.3 \\ 20.3 \\ 21.0 \\ 19.2 \\ 19.3 \end{pmatrix} \]

\[ X = \begin{pmatrix} 1 & 31 \\ 1 & 29.9 \\ 1 & 30 \\ 1 & 29.6 \\ 1 & 29.4 \\ 1 & 28.3 \\ 1 & 28.7 \\ 1 & 27.9 \\ 1 & 27.6 \\ 1 & 27.3 \\ 1 & 26.9 \\ 1 & 25.7 \\ 1 & 26.3 \\ 1 & 26.2 \\ 1 & 25.4 \\ 1 & 26.2 \\ 1 & 25.3 \\ 1 & 25.5 \\ 1 & 25.6 \\ 1 & 24.3 \end{pmatrix} \]

\[ b = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \]

\[ e = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \\ e_{10} \\ e_{11} \\ e_{12} \\ e_{13} \\ e_{14} \\ e_{15} \\ e_{16} \\ e_{17} \\ e_{18} \\ e_{19} \\ e_{20} \end{pmatrix} \]

2.2.2 Estimation with least squares

Solution of the model (2.4) becomes easily tedious to compute, because the number of equations needed equals to the number of parameters in \( b \). That is why linear models and the least squares solutions are usually presented in the matrix form. The sum of squared residuals is expressed in the matrix form as \( (y - Xb)'(y - Xb) \). The OLS estimator of \( b = (b_0, b_1)' \) is in the matrix form

\[ \hat{b} = (X'X)^{-1}X'y \]

It can be shown that the least squares estimator is unbiased and has variance-covariance matrix \( \sigma^2 (X'X)^{-1} \):

\[
E(\hat{b}) = E[(X'X)^{-1}X'y] \\
= (X'X)^{-1}X'Ey \\
= (X'X)^{-1}X'Xb \\
= b
\]
\[
\text{cov}(\hat{b}) = \text{cov}
\left[
(X'X)^{-1}X'y
\right]
\]
\[
= (X'X)^{-1}X'\text{cov}(y)X(X'X)^{-1}
\]
\[
= \sigma^2 I
\left[
(X'X)^{-1}X'X(X'X)^{-1}
\right]
\]
\[
= \sigma^2 (X'X)^{-1}
\]

For estimation of residual variance, let us define matrix \( H \) as
\[
H = XX'X^{-1}X'
\]

An unbiased estimator for the residual variance is
\[
\hat{\sigma}^2 = \frac{y'(I-H)y}{n-p}, \tag{2.6}
\]
where \( p \) is the number of fixed parameters in the model, i.e., the length of vector \( b \).

Note that we did not need any assumption on the distribution of residuals for these proofs. It can be shown that the LS estimator is the Best Linear Unbiased Estimator of the parameters of model (2.5). This means that from among estimators that are unbiased and linear with respect to observations, (i.e., are of form \( ay \)), the OLS-estimator has the smallest variance. However, this holds only when the assumptions (constant variance and no correlation) on residuals hold.

**Example 2.7.** Using \( X \) and \( h \) as defined in example 2.6, the OLS estimate for \( b \) is

\[
\hat{b} = (X'X)^{-1}X'h
\]
\[
= \begin{pmatrix}
11.4394735 \\
0.3421870 
\end{pmatrix}
\]

The residual variance is
\[
\hat{\sigma}^2 = \frac{y'(I-H)y}{n-p-1} = 1.215681
\]

**The R-code for Example 2.7**

```R
> onestand<read.table("d:/laurim/biometria/onestand.txt",header=TRUE)
> X<-cbind(1,onestand$d)
> h<onestand$h
> b<-solve(t(X)%*%X)%*%t(X)%*%h
> H<-X%*%solve(t(X)%*%X)%*%t(X)
> I<-diag(rep(1,20))
> sigma2<-t(h)%*%(I-H)%*%h/18
> b
[,1]
[1,] 11.4394735
[2,] 0.3421870
> sigma2
[,1]
[1,] 1.215681
```
The estimates are, again, exactly the same as those obtained in example 2.2

**Example 2.8.** Let us fit the model of example 2.3 to the data of 200 height-diameter pairs from different stands. To study the trends in residuals, we plot the mean of variable \( y \) as a function of \( x \) in a given number of classes. In addition, we add two vertical lines, centered at the class mean. The first one is proportional to the class-specific standard deviation (multiplied by \( 2 \times 1.96 \) to yield a 95% confidence interval based on assumption of normality) at. The second one is proportional to the class-specific standard errors of means is a similar way. Adding these lines to a scatterplot of residuals, gives a good plot for analyzing the trends in residuals. The lines based on class-specific standard deviations can be used to analyse if the residuals have constant variance or not. The lines based on the standard error of means can be used to assess if the model fits well enough over the whole range of predictions.

Function `mywhiskers` can be used to make such a plot. Using `se=TRUE` produces the first version and using `se=FALSE` produces the second version.

The following code was used to fit the model and plot the residuals:

```r
> fm3 <- lm(h ~ log(d+10), data=hddata)
> summary(fm3)
Call:
  lm(formula = h ~ log(d + 10), data = hddata)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -33.781     1.960  -17.23  < 2e-16
log(d + 10)  14.801     0.960   15.26  < 2e-16

Residual standard error: 6.173 on 197 degrees of freedom
Multiple R-squared:  0.79,   Adjusted R-squared:  0.788
F-statistic:  231 on 1 and 197 DF,  p-value: < 2.2e-16
```

```r
> with(fm3, mywhiskers(add=TRUE))
```

```r
> plot(predict(fm3), resid(fm3), xlab="Predicted value", ylab="Residual", ylim=c(-8,8))
> abline(0,0)
```
All 95% confidence intervals for mean (the thick vertical lines) overlap the x-axis (Figure 2.6). This means that the model of form \( h = b_0 + b_1 \ln(d + 10) \) fits rather well to the data. However, the class-specific standard deviations (the thin vertical lines) indicate increasing variance as a function of prediction.

### 2.2.3 The design matrix

The design matrix specifies the structure of the model. It is usually a big matrix: the number of rows equals to the number of observations, and it has at least as many columns than there are predictors in your model. Some important points about the structure and construction of the design matrix are given here. Statistical packages such as R, SPSS or SAS automatically construct the design matrix, but it is still important for a modeler to understand what is done in the computer memory.

Continuous predictors are included in the original form, as we did in example 2.6. Including factors is a little different. Factors having only two levels are included as one dummy variable, which gets values 0 or 1. The coefficient of that variable expresses the difference of the level coded as 1 to the level coded as 0. Factors having \( p > 2 \)
levels are coded as \( p - 1 \) dummy variables, each of them indicating if a specific level was present (dummy=1) or not (dummy=0). One level is selected as the default or reference level, which the other levels are compared to. From mathematical point of view, the selection of default level is unimportant. However, the selection of the default level affects the interpretation of the coefficients: the coefficients of other levels specify the difference to the default level. Also the t-tests on the individual coefficients test the hypothesis “do this level significantly differ from the default level’’. That is why the level can, and should be selected so that the interpretation makes sense.

**Example 2.9.** In example 2.6, half of the trees were Scots pines, the rest 10 sample trees being Norway spruces. In addition to tree diameter, we could include a dummy variable into our model, indicating whether the tree is spruce or pine. There is no good reason to select which of the tree species would be coded as 0 (the default level). We just select Scots pine as the default.

The model would be

\[
h_i = b_0 + b_1 d_i + b_2 SPRUCE_i + e_i \tag{2.7}
\]

where \( SPRUCE_i \) is an indicator variable getting value 1 when tree \( i \) is spruce, and 0 elsewhere. The design matrix and parameter vector \( b \) become

\[
X = \begin{pmatrix}
1 & 31 & 0 \\
1 & 29.9 & 1 \\
1 & 30 & 1 \\
1 & 29.6 & 0 \\
1 & 29.4 & 0 \\
1 & 28.3 & 0 \\
1 & 28.7 & 0 \\
1 & 27.9 & 0 \\
1 & 27.6 & 1 \\
1 & 27.3 & 1 \\
1 & 26.9 & 1 \\
1 & 25.7 & 1 \\
1 & 26.3 & 0 \\
1 & 26.2 & 1 \\
1 & 25.4 & 1 \\
1 & 26.2 & 0 \\
1 & 25.3 & 0 \\
1 & 25.5 & 1 \\
1 & 25.6 & 0 \\
1 & 24.3 & 1 \\
\end{pmatrix}
\]

The parameter estimates are obtained as

\[
> X<-cbind(1,onestand$d,onestand$pl-1) \\
> h<-onestand$h \\
> b<-solve(t(X)%*%X)%*%t(X)%*%h
\]
CHAPTER 2. LINEAR MODEL

> H <- X %*% solve(t(X) %*% X) %*% t(X)
> I <- diag(rep(1, 20))
> sigma2 <- t(h) %*% (I - H) %*% h / 17
> b

1] 9.6564516
[2] 0.3935878
[3] 0.7539084
> sigma2

1] 1.063

or using

> onestand$pl <- as.factor(onestand$pl)
> fm2 <- lm(h ~ d + as.factor(pl), data = onestand)
> summary(fm2)

Call:
  lm(formula = h ~ d + as.factor(pl), data = onestand)

Residuals:
     Min      1Q  Median      3Q     Max
-1.7180 -0.6700 -0.1904  0.5805  2.8447

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)       9.6565 3.6817 2.623  0.01782 *
d               0.3936 0.1317 2.988  0.00827 **
as.factor(pl)2  0.7539 0.4918 1.533  0.14366
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 1.063 on 17 degrees of freedom
Multiple R-squared: 0.36, Adjusted R-squared: 0.2847
F-statistic: 4.782 on 2 and 17 DF, p-value: 0.02251

Note that we used function as.factor to coerce the tree species to a factor. The output shows that tree species 1 was set as the default level, as we did also with our design matrix.

> plot(onestand$d, onestand$h, pch = as.numeric(onestand$pl), xlab = "Diameter, cm", ylab = "Height, m")
> abline(coef(fm2)[1], coef(fm2)[2])
> abline(sum(coef(fm2)[c(1, 3)]), coef(fm2)[2])

The above design matrix specified a model where the dependence of height on diameter is similar for both tree species, except for a horizontal shift in the level of the model. Thus, the assumed model for Scots pine is

\[ h_i = b_0 + b_1 d_i + e_i \]

For Norway spruce, the assumed model is

\[ h_i = b_0 + b_2 + b_1 d_i + e_i \]

Those are illustrated in plot 2.7.

One could also be interested in fitting a models where also the slope varies between the tree species. Such a model is obtained by including an interaction term into the model, as demonstrated in the following example.

**Example 2.10.** Assume that we want to have also different slopes for Scots pine and Norway spruce by including an interaction term into our model. Such a model is defined as

\[ h_i = b_0 + b_1 d_i + b_2 SPRUCE_i + b_3 \ast SPRUCE_i d_i + e_i \]
2.2. MULTIPLE REGRESSION

Figure 2.7: Observations of the sample stand (circles for pine and triangles for spruce), and the fitted line of model (2.7) for spruce (upper) and pine (lower).

The design matrix corresponding to this model is defined as

\[
X = \begin{pmatrix}
1 & 31 & 0 & 0 \\
1 & 29.9 & 1 & 29.9 \\
1 & 30 & 1 & 30 \\
1 & 29.6 & 0 & 0 \\
1 & 29.4 & 0 & 0 \\
1 & 28.3 & 0 & 0 \\
1 & 28.7 & 0 & 0 \\
1 & 27.9 & 0 & 0 \\
1 & 27.6 & 1 & 27.6 \\
1 & 27.3 & 1 & 27.3 \\
1 & 26.9 & 1 & 26.9 \\
1 & 25.7 & 1 & 25.7 \\
1 & 26.3 & 0 & 0 \\
1 & 26.2 & 0 & 0 \\
1 & 25.4 & 1 & 25.4 \\
1 & 26.2 & 0 & 0 \\
1 & 25.3 & 0 & 0 \\
1 & 25.5 & 1 & 25.5 \\
1 & 25.6 & 0 & 0 \\
1 & 24.3 & 1 & 24.3
\end{pmatrix}
\]

\[
b = \begin{pmatrix}
b_0 \\
b_1 \\
b_2 \\
b_3
\end{pmatrix}
\]

The assumed model for Scots pine is now \( h_i = b_0 + b_1 d_i + e_i \), and that for Norway spruce is \( h_i = b_0 + b_2 + (b_1 + b_3) d_i + e_i \). Thus, we actually assume separate models for the tree species. An equivalent model would be obtained by splitting the data according to tree species, and fitting separate models for both sub-dataset.
Another important question about the design matrix is if the constant is included or not. Usually the constant is included, i.e., the first column of the design matrix consists of ones. However, sometimes the phenomenon to be modeled itself is such that constant does not have a practical meaning, or the model without constant would be more realistic. In such cases, the constant can be left out of the model.

**Example 2.11.** As diameters have been measured at breast height, it could be realistic to assume that height for diameter \( d = 0 \) is 1.3. Thus, a realistic assumption would be to assume the constant to be 1.3. Such an assumption can be implemented by subtracting the value 1.3 meters from the measured heights, and assuming that the constant \( b_0 \) is zero in the resulting model

\[
h_i - 1.3 = b_0 + b_1 d_i + e_i.
\]

Vectors and matrices \( h, X, \) and \( b \) become

\[
h = \begin{pmatrix} 21.5 \\ 20.3 \\ 19.2 \\ 21.2 \\ 19.1 \\ 18.4 \\ 20 \\ 20 \\ 19.5 \\ 22.7 \\ 20 \\ 19.4 \\ 18 \\ 18.9 \\ 19.2 \\ 18 \\ 19 \\ 19.7 \\ 17.9 \\ 18 \end{pmatrix} \quad X = \begin{pmatrix} 31 \\ 29.9 \\ 30 \\ 29.6 \\ 29.4 \\ 28.3 \\ 28.7 \\ 27.9 \\ 27.6 \\ 27.3 \\ 26.9 \\ 25.7 \\ 26.3 \\ 26.2 \\ 25.4 \\ 26.2 \\ 25.3 \\ 25.5 \\ 25.6 \\ 24.3 \end{pmatrix} \quad b = \begin{pmatrix} b_1 \end{pmatrix}
\]

### 2.2.4 Least squares for LM with a general residual variance structure

If the assumptions on the variance and correlation of error terms do not hold, OLS estimation is still unbiased. However, the OLS estimator is no more the minimum-variance-estimator. With such data, Generalized least squares (GLS) estimation can be used. In GLS, the variance among residuals may vary, and residuals may also be correlated. These assumptions are parameterized into the variance-covariance matrix
of residuals, which is then used to obtain GLS estimates for the parameters of the linear model. A WLS estimator is also sometimes used. It is a special case of GLS, where residuals are assumed to be uncorrelated, but the variance is assumed to vary among observations. The different methods are consistent so, that GLS with zero correlations leads to WLS. Furthermore, WLS with constant variance yields OLS. We will go directly to GLS principle, as WLS is just a special case of the more general GLS approach.

For presentation of the GLS estimation, we assume that the response $y$ depends on the design matrix $X$ in the similar fashion than before.

$$y = Xb + e$$  \hspace{1cm} (2.8)

The model differs from model (2.5) in the assumptions about the residual vector. We assume that

$$E(e) = 0$$
$$\text{var}(e) = \sigma^2 V,$$

where $V$ is any symmetric positive definite matrix.

We are still assuming that design matrix $X$ and coefficients $b$ are fixed, and these assumptions imply that the following results still hold

$$E(y) = Xb$$
$$\text{var}(y) = \text{var}(e) = \sigma^2 I$$

An interesting special case of GLS results if we know only means of $y$ in $p$ classes of variable $x$, and classes have unequal number of observations. Assumption of constant variance for raw unobserved data implies that classes have unequal variances, variance of class $i$ being $\sigma^2/n_i$. The variance-covariance matrix of the classified data is then $\sigma^2 W$, where

$$W = \begin{pmatrix}
1/n_1 & 0 & \ldots & 0 \\
0 & 1/n_2 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & 1/n_p
\end{pmatrix}$$

The OLS estimator generalizes to the case. The Estimator that properly accounts for the relaxed assumptions on the residual is the estimator that minimizes the sum of squares $(y - Xb)'V^{-1}(y - Xb)$. The Generalized Least Squares (GLS) estimator is

$$\hat{b} = (X'V^{-1}X)^{-1}X'V^{-1}y$$  \hspace{1cm} (2.9)
For model (2.8), this is the BLUE, i.e., it has the smallest variance from among all possible unbiased estimators (Gauss-Markov theorem).

It can be shown that the GLS estimator is unbiased and has variance-covariance matrix $\sigma^2 X'V^{-1}X^{-1}$:

$$
E(\hat{b}) = E[(X'V^{-1}X)^{-1}X'V^{-1}y] = (X'V^{-1}X)^{-1}X'V^{-1}Exy = (X'V^{-1}X)^{-1}X'V^{-1}Xb = b
$$

$$
cov(\hat{b}) = cov[(X'V^{-1}X)^{-1}X'V^{-1}y] = (X'V^{-1}X)^{-1}X'V^{-1}covy[(X'V^{-1}X)^{-1}X'V^{-1}] = \sigma^2 [(X'V^{-1}X)^{-1}X'V^{-1}X'(X'V^{-1}X)^{-1}] = \sigma^2 (X'V^{-1}X)^{-1}
$$

For estimation of residual variance, let us define matrix $G$ as

$$
G = X(X'V^{-1}X)^{-1}X'V^{-1}
$$

An unbiased estimator for the residual variance is

$$
\widehat{\sigma^2} = \frac{y'(I - G)V^{-1}(I - G)y}{n - p}, \quad (2.10)
$$

where $p$ is the number of fixed parameters in the model, i.e., the length of vector $b$.

In R, function `lm` can be used in fitting OLS and WLS models, i.e., it can be used for modeling when all off-diagonal elements of matrix $V$ are zeros. For fitting GLS models with any positive definite $V$, one alternative is to use function `gls` in package `nlme`.

**Example 2.12.** In example 2.8, a good form was found for the fixed part of the model, but the residuals showed an increasing variance as a function of prediction. In order to make a better assumption on the residual variance, we model the squared residuals of the OLS model on prediction.

Instead of assuming constant variance, a widely applicable assumption on the residual is provided by the function

$$
var(e_i) = \sigma^2 |d_i|^{25}
$$
2.2. MULTIPLE REGRESSION

Depending of the value of delta, we may have decreasing variance as a function of \( d \) (\( \delta < 0 \)), constant variance (\( \delta = 0 \)) and increasing variance (\( \delta > 0 \)).

In function \texttt{gls} of package \texttt{nlme}, we have several options for the form of the variance function. A model assuming the above-specified variance can be fitted using

```r
> library(nlme)
> fm7<-gls(h~log(d+10),data=hddata,weights=varPower(0.5,~d))
> fm7
Generalized least squares fit by REML
Model: h ~ log(d + 10)
Data: hddata
Log-restricted-likelihood: -452.6542

Coefficients:
(Intercept) log(d + 10)
-32.08281 14.25532

Variance function:
Structure: Power of variance covariate
Formula: ~d
Parameter estimates:
power
0.5590031
Degrees of freedom: 200 total; 198 residual
Residual standard error: 0.5427085
```

The lower plot of figure2.9 shows the standardized residuals from this model.

**Example 2.13.** This example refits the model using another variance function than the \texttt{varPower} of the previous example. We use \texttt{weights=varFixed( x)}, where \( x \) is a variable of our data. The variance is assumed to depend on the given covariate according to \( \text{var}(e) = \sigma^2 x \). To specify a model where variance depends on the predicted value, we first save the predicted value into our data, and then use it as a covariate in the variance function.

```r
> library(nlme)
> hddata$fitted.fm3<-predict(fm3)
> fm5<-gls(h~log(d+10),data=hddata,weights=varFixed(~fitted.fm3))
> fm5
Generalized least squares fit by REML
Model: h ~ log(d + 10)
Data: hddata
Log-restricted-likelihood: -454.3684

Coefficients:
(Intercept) log(d + 10)
-32.58795 14.42168

Variance function:
Structure: fixed weights
Formula: ~fitted.fm3
Degrees of freedom: 200 total; 198 residual
Residual standard error: 0.660629
```

```r
> windows(width=3,height=2.5)
> par(mai=c(0.6,0.5,0.1,0.1),mgp=c(2,0.7,0),cex=0.8)
> plot(predict(fm5), resid(fm5,type="pearson"),
+ xlab="Predicted value",
+ ylab="Standardized residual")
> points(predict(fm4), resid(fm4,type="pearson"),pch=3)
> mywhiskers(predict(fm5), resid(fm5,type="pearson"),se=FALSE,add=TRUE)
> mywhiskers(predict(fm5), resid(fm5,type="pearson"),add=TRUE,lwd=3)
```
Figure 2.8: Plot for example 2.12
2.2. MULTIPLE REGRESSION

Plot 2.9 shows that, the residuals differ slightly, and estimates for residual standard error are very different. The slight differences result from the use of REML for estimating the residual variance. The difference in standard error is an artifact, resulting from that we assumed the variance to be proportional to predicted value from fm4, whereas in the previous example it was assumed to be proportional to the predicted variance. The following code illustrates a manually computed model that corresponds to the REML-model. The squared ratio of standard errors \((0.6606/0.9775)^2\) is equal to the coefficient of the variance function, 0.4568.

```r
> fm6<-update(fm3, weight=1/predict(fm3))
> summary(fm6)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -32.5880   1.5801  -20.62  <2e-16 ***
log(d + 10)  14.4217   0.5115   28.19  <2e-16 ***

Residual standard error: 0.6606 on 198 degrees of freedom

It is impossible to say which of the two alternative variance functions was better.

Slight differences are seen in the estimates of the fixed parameters.

Example 2.14. It is also possible to use the argument `weights` of function `lm` to fit models with heterogeneous residuals, as demonstrated in the following example. The following code plots the squared residuals and fits a linear model to them.

```R
> plot(predict(fm3), resid(fm3)^2)
> lm.var<-lm(I(resid(fm3)^2) ~ predict(fm3) - 1)
> summary(lm.var)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -32.5880   1.5801  -20.62  <2e-16 ***
log(d + 10)  14.4217   0.5115   28.19  <2e-16 ***

Residual standard error: 0.6606 on 198 degrees of freedom

Call:
  lm(formula = I(resid(fm3)^2) ~ predict(fm3) - 1)

Among the standardized residuals of model fm4 and the circles those of fm5.

Figure 2.9: Plot for example 2.13.
Coefficients:
predict(fm3)
   0.4568 > abline(lm.var)

The model of squared residuals has a positive coefficient, implying that the residual variance is increasing as a function of predicted value. A model assuming a heterogeneous residual variance according to the variance model is fitted by using the
weight=1/predict(lm.var) in function lm.

> fm4<-update(fm3,weight=1/predict(lm.var)) > fm4
Call: lm(formula = h ~ log(d + 10), data = hddata, weights = 1/predict(lm.var))
Coefficients:
(Intercept) log(d + 10)
 -32.59 14.42

> summary(fm4)$sigma
[1] 0.9774896

The coefficients are slightly different from those obtained using OLS. With weighted fit, the standardized residuals should be homogeneous, indicating that the assumed variance function properly takes into account the heteroscedastic variance. Those residuals are obtained by using type="pearson" in function resid. The plot of standardized residuals shows that the residuals are not that much heteroscedastic any more.

> plot(predict(fm4), resid(fm4,type="pearson"), xlab="Predicted value", ylab="Standardized residual") > mywhiskers(predict(fm4),resid(fm4,type="pearson"),se=FALSE,add=TRUE) > mywhiskers(predict(fm4),resid(fm4,type="pearson"),add=TRUE,lwd=3) > abline(0,0)

Even though estimates seldom need to be computed manually using the matrix equations, it is useful to be able to do it in order to better understand the general applicability of the GLS-estimator 2.9. Furthermore, it may be sometimes useful to be able to compute the estimates. For example, it is not always very clearly reported if the weights needed in a function fitting a wls model should be given as variances or standard deviations, and if they need to be inverted or not. To test which alternative should be used, one option is to fit a small model manually, and to test which definitions leads to the estimates obtained using the function used for model fitting.

**Example 2.15.** To show that the estimates are based on the estimator 2.9, we fit the model of example 2.12 manually.

> V<-diag(predict(lm.var)) > X<-cbind(1,log(hddata$d+10)) > y<-hddata$h > est<-solve(t(X)%*%solve(V)%*%X)%*%t(X)%*%solve(V)%*%y > est
[,1]
[1,] -32.58795
[2,]  14.42168 > G<-X%*%solve(t(X)%*%solve(V)%*%X)%*%t(X)%*%solve(V) > I<-diag(rep(1,length(y))) > sigma2<1/(dim(X)[1]-dim(X)[2])*t(y)%*%(I-G)%*%solve(V)%*%(I-G)%*%y
We see that the estimates are exactly equal to those obtained using `lm`.

It is easy to see, how all the results on OLS can be easily derived by replacing $V$ with $I$. Furthermore, we still rely only on assumptions on expectation, variance and covariance of residuals, and no specific assumption on the distribution was made.

### 2.3 Inference and tests of the regression relationship

#### 2.3.1 Sums of squares

The overall variation of response around its mean is defined as

$$SS_{tot} = \sum (y_i - \bar{y})^2.$$  

It can be divided into two components: the unexplained variation (residual sum of squares)

$$RSS = \sum (y_i - \hat{y}_i)^2$$  

and explained variation

$$SS_{reg} = SS_{tot} - RSS.$$  

A widely used figure for evaluation of the goodness of fit of the regression relationship is the degree of determination

$$R^2 = \frac{1 - RSS}{SS_{tot}} = \frac{SS_{reg}}{SS_{tot}},$$

which tells which portion of the total variation in $y$ was explained by the estimated regression relationship. However, even though $R^2$ is a useful figure for comparison, the model evaluations should never be based on it alone. It does not tell if the assumptions of the linear model were met or not. In addition, one should note that $R^2$ is different in different scale. With the single-predictor regression, the coefficient of determination is the squared correlation coefficient between $x$ and $y$.

The adjusted $R^2$ takes into account the degrees of freedom that are used for model fitting. it is defined as

$$R^2_{adj} = 1 - (1 - R^2) \frac{n - 1}{n - p - 1}.$$  

**Example 2.16.** The following code manually computes $R^2$ for the model of example 2.1.
The same value is printed by `summary`:

```r
> summary(fm1)
...
Multiple R-squared: 0.2716, Adjusted R-squared: 0.2311
...
```

**Example 2.17.** Two models, where one has $\log(V)$ as the response and the other has $V$. The value of $R^2$ is completely different.

### 2.3.2 The assumption normality of residuals

Until now, we only have assumed that $\epsilon$ is independent and has constant variance or a variance specified by a variance function. However, as noted earlier in chapter 1, tests need an assumption on the distribution of a random variable. Therefore, in order to be able to test the significance of predictors, we additionally assume that $\epsilon_i$ are generated by the normal distribution. This assumption is made without further theoretical results suggesting it. Therefore, an essential part of evaluation of the quality of the test result is a check on the normality of the residuals.

We assume that

$$
\epsilon_i \sim N(0, \sigma^2).
$$

This implies that

$$
\epsilon \sim N_n(0, \sigma^2 I)
$$

and

$$
y \sim N_n(Xb, \sigma^2 I)
$$

It results from the normality of the residuals (or of the response) that also the parameter estimates have the normal distribution (assuming that the residual standard error is known), the residual and regression sum of squares have the $\chi^2$ distribution, and the ratio of regression and residual sum of squares has the $F$ distribution

$$
\frac{(n - p - 1)(y - X\hat{b})'(y - X\hat{b})}{\sigma^2} \sim \chi^2(n - p - 1)
$$

$$
\frac{SS_{reg}/(p)}{RSS/(n - p - 1)} \sim F(p, n - p - 1)
$$

Replacing the true residual standard error in the above equation with the corresponding estimate (which has the $\chi^2$ distribution) leads to the $t$ distribution of regression coefficients.
2.3. INFERENCE AND TESTS OF THE REGRESSION RELATIONSHIP

2.3.3 F-test for the significance of regression

A very basic question about the regression relationship is, whether the explanatory variables significantly explain the variation of the response. This question can be answered by testing if the explained variation is big when compared to the unexplained variation. The Null and alternative hypotheses are

\[ H_0 : b_0 = b_1 = \ldots = b_p = 0 \]
\[ H_1 : \text{some } b_i \neq 0 \]

The test statistic is

\[ F_{obs} = \frac{SS_{reg}/p}{RSS/(n - p - 1)} \]

It results from the normality of residuals that the two sums of squares in the test statistic are distributed according to \( \chi^2 \) distribution with \( p \) and \( n - p - 1 \) degrees of freedom for numerator and denominator, respectively. Thus, we remember from section 1.4.3 that the test statistic is distributed according to \( F(p, n - p - 1) \) degrees of freedom. The p-value is

\[ p = P(F > F_{obs}) \]

i.e., the value of cdf of F-distribution with \( p \) and \( n - p - 1 \) degrees of freedom at \( F_{obs} \).

**Example 2.18.** To do tests, the first step is to check if the assumptions on the model were met. To do this, we plot the Normal quantile-quantile plots, which plot the realized residual quantiles against the theoretical quantiles based on Normal distribution. If normality is met, these observations should be close to a line having constant of 0 and coefficient of 1.

```r
qqnorm(resid(fm1),main="Q-Q-plot for fm1")
abline(0,1)
qqnorm(resid(fm2),main="Q-Q-plot for fm2")
abline(0,1)
```

The plots are shown in figure 2.10. The largest observed residual is overly large to be from a normal distribution, especially for model fm1. However, we regard this plot good enough, and trust on normality in our tests.

**Example 2.19.** For model 2.1, the F test statistic is computed as follows

```r
> Fobs<-SSreg/1/(RSS/(20-1-1))
> Fobs
[1] 6.710426
> 1-pf(Fobs,1,18)
[1] 0.01846628
```

The results show that there is a statistically significant (p=0.018) dependence between tree height and diameter in the example stand.
2.3.4 The t-test for coefficients

The confidence intervals and tests for parameter vector $b$ can be derived using result

$$\frac{p'\hat{b} - p'b}{\hat{\sigma}/\sqrt{p'(X'X)^{-1}p}} \sim t(n - p)$$

where $p$ is any vector of length same length as $b$. It is used to pick one or several coefficients from the vector of coefficients for testing. The most commonly used application is to test one coefficient at a time. In such case, the distribution of only one element of $b$, say $b_i$, is obtained by having all elements of $p$ as zeros, except for the $i$th element, which is given the value of 1. In this case, the above general result simplifies to

$$\frac{\hat{b}_i - b_i}{s.e.(\hat{b}_i)} \sim t(n - p)$$

(2.11)

where $s.e.(\hat{b}_i)$ is the standard error of the estimate, which is obtained by taking the square root of the $i$th diagonal element of matrix $(X'X)^{-1}$, and multiplying it by $\hat{\sigma}$.

The most common situation is to test whether a coefficient significantly differs from zero. The null and alternative hypotheses are

$$H_0 : p'b = 0$$

$$H_1 : p'b \neq 0$$
And the test statistic simplifies to

$$t_{obs} = \frac{p'\hat{b}}{\hat{\sigma}\sqrt{p'(X'X)^{-1}p}},$$

which simplifies to the form of (2.11) in the case where only one coefficient is tested.

Under the null hypothesis, $t \sim t(n - p)$. The null hypothesis is rejected if $|t| > t_{\alpha/2, n-p}$. Note that we use absolute value of the t-statistic to ensure that we are computing the probability of the tail of the distribution also in the case of negative coefficient. Instead of performing the test, we could compute and report the p-value itself. The p-value of a two-sided hypothesis is computed as $p = P(|t| > |t_{obs}|) = 2P(t > |t_{obs}|)$, i.e., twice the value of the cumulative distribution function of t-distribution with $n - p$ degrees of freedom. The t-tests and p-values reported by statistical model-fitting procedures are applications of this test.

**Example 2.20.** A researcher may be interested whether the level of the H-D curve could be assumed to be the same for both tree species. A test on such a null hypothesis is seen directly from the t-test of the tree species dummy variable. In this case, null hypothesis on different constant for different tree species would be rejected (p-value = 0.14).

```r
> summary(fm2)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.6565    3.6817   2.623  0.01782 *
d           0.3936    0.1317   2.988  0.00827 **
as.factor(pl)2 0.7539    0.4918   1.533  0.14366
```

**Example 2.21.** We want to test whether the constant of the model of example 2.1 is 1.3 meters. For this purpose, we could define

$$p = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1.3 \\ 0 \end{bmatrix}$$

and perform the test using the general results as follows.

```r
> p<-c(1,0) # t-value for constant
> b<-c(1.3,0)
> bhat<-coef(fm1) # estimate of b
> bhat

            (Intercept)     d
11.4394735  0.3421870

> sigma<-summary(fm1)$sigma # residual s.e.
> sigma

[1] 1.102579

> X<-cbind(1,onestand$d)
> se<-sigma*sqrt(p%*%solve(t(X)%*%X)%*%p)
> se

[,1]
[1,] 3.621877

> t<-p%*%bhat-p%*%b/se
> t

[,1]
[1,] 2.799500

> 2*(1-pt(t,20-2))

[,1]
[1,] 0.01184922
```
A simpler approach would be to use

```r
> t <- (coef(fm1)[1] - 1.3) / (sqrt(solve(t(X) %*% X)[1,1]) * sigma)
```

for computing the t-value. The p-value is 0.01, which shows that the null hypothesis about constant of 1.3 meters is rejected. Our model predicts height of 11 meters for trees of diameter 0. Maybe a better model specification is needed.

The same result is obtained by fitting two models: a full model and a restricted model using `offset` in our model. As the restriction is made for the constant, we first need to add a variable including ones into our dataset. Then we fit a model without constant using `-1` in `h ~ d-1`, but specify a predefined constant of 1.3 by using argument `offset=I(1.3*ones)`. The two models can then be obtained by using `anova()`, which performs a comparison of two models that is based on the F-test. This approach for testing two models is explained in the next subsection.

```r
> onestand$ones <- 1
> fm2 <- lm(h ~ d-1, data=onestand, offset=I(1.3*ones))
> anova(fm1, fm2)
```

Analysis of Variance Table

<table>
<thead>
<tr>
<th>Model 1: h ~ d</th>
<th>Model 2: h ~ d - 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Res.Df</td>
<td>18</td>
</tr>
<tr>
<td>RSS</td>
<td>21.8823</td>
</tr>
<tr>
<td>Df</td>
<td>-1</td>
</tr>
<tr>
<td>Sum of Sq</td>
<td>-9.5276</td>
</tr>
<tr>
<td>F</td>
<td>7.8372</td>
</tr>
<tr>
<td>Pr(&gt;F)</td>
<td>0.01185 *</td>
</tr>
</tbody>
</table>

---

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

The same result can be seen from the following table by comparing the standard error of the constant to the difference between assumed and estimated value of the constant.

```r
> summary(fm1)
```

Call: lm(formula = h ~ d, data = onestand)

... Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.4395 3.6219 3.158 0.00544 **
d 0.3422 0.1321 2.590 0.01847 *

... 2.3.5 Testing several coefficients at the same time

The tests produced by statistical packages usually perform two kinds of standard tests: the F-test for significance of the regression relationship, and a t-test for testing if individual coefficients significantly differ from 0. As demonstrated in the previous section, the t-test can also be used for testing differences from any other constants than 0, or several individual coefficients at the same time. However, a perhaps more easily applicable approach is to use F-test. In that test, two models: a full model and a restricted model, are tested against each other. The test reports whether the difference between the models could result on random variation.

Assume that we have regression relationship \( y = Xb + e \), and we want to test a reduced model with different constraints for the model. Those constraints may be whatever, including dropping some predictors, giving a constant value for a coefficient,
or setting some coefficients to be equal. Whatever constraints are made, they result to a **constrained model**, which has to be some kind of special case of the full, more general model.

A test for testing if removing the constraints significantly improved the fit is formulated as follows. Define $RSS_{NH}$ as the sum of squares from the reduced (constrained) model, having $q$ predictors, and $RSS_{AH}$ as the sum of squares from the full (unconstrained) model having $p$ predictors. These sums of squares are independent, $\chi^2$-distributed random variables. Furthermore, also the difference between two $\chi^2$-distributed random variables is distributed according to $\chi^2$ distribution. Thus, the test statistic

$$F_{obs} = \frac{(RSS_{NH} - RSS_{AH})/(p - q)}{RSS_{AH}/(n - p - 1)}$$

is distributed according to $F$-distribution with $p - q$ and $n - p - 1$ degrees of freedom, if the null hypothesis is true. The p-value is obtained from F distribution in a similar way as earlier.

It is worth noting, that even if the likelihood principle will be introduced only later in this chapter, it would lead to the same test using the principle of likelihood ratio.

In R, these comparisons can be carried out easily by using the `anova` function, as we already demonstrated in the previous section, and will further demonstrate in the following examples. The constrained model can be fitted by dropping predictors. A constant coefficient is assumed for a specific predictor can be given by using the `offset` argument in `lm(formula,data,offset)`.

**Example 2.22.** Assume we want to test if the model with separate constants and slopes for the two species was significantly different from the model with common slope and constant. The full model is the one with species-specific coefficients and the constrained model is the one with common coefficients. The models are fitted and test statistics computed as follows. Note that all main effects and interactions are included by having model equation $h \sim d*p.l$.

```r
> full<-lm(h~d*p.l,data=onestand)
> constr<-lm(h~d,data=onestand)
> RSSnh<-sum(resid(constr)^2)
> RSSah<-sum(resid(full)^2)
> Fobs<-(RSSnh-RSSah)/(4-2)/(RSSah/(20-3-1))
> Fobs
[1] 1.988027
> 1-pf(Fobs,2,16)
[1] 0.1693878
```

The full model is not significantly better than the restricted model. Using function `anova` results into an identical result:

```r
> anova(constr,full)
Analysis of Variance Table

Model 1: h ~ d
Model 2: h ~ d * p.l
  Res.Df RSS Df Sum of Sq F Pr(>F)
1     18 21.8823
2     16 17.5268 2  4.3555 1.988 0.1694
```

The full model is not significantly better than the restricted model. Using function `anova` results into an identical result:
The t-test and F test lead to similar inference if the test is such that it can be performed with both methods. This is demonstrated by the following example.

**Example 2.23.** In example 2.20, model fm2 included separate coefficients for both tree species. A constrained version is the model fm1, which includes no species-specific parameters. It was shown that the p-value for the species dummy was 0.14366, which indicates that the level does not vary between tree species. The same hypothesis can be tested by making F-test between models fm2 and fm1. We get

```
> anova(fm1,fm2)
Analysis of Variance Table

Model 1: h ~ d
Model 2: h ~ d + as.factor(pl)

  Res.Df   RSS Df Sum of Sq F Pr(>F)
1  18 21.8823
2  17 19.2244 1  2.6578 2.3503 0.1437
```

The p-value and resulting inference are exactly the same as we got from the t-test.

**2.4 Checking the validity of assumptions**

The basic assumptions of a linear model are that the model is correct, observations are independent and have equal variance. In addition, for testing hypotheses on the model estimated by least squares, an assumption about normality of residuals is needed. For ML-estimation, the normality needs to be assumed already at the model fitting stage. In this section, strategies for situations where those assumptions are not met are discussed. R has method `plot()` for producing some standard plots (such as residual and qq-plots) from a fitted model.

**2.4.1 Model shape**

The assumption that the model is correct means that the assumed function for the modeled relationship is able to describe the true relationship between the variables. In practice, we seldom have any well-justified theory for the assumed relationship, and the model form is many times a result of a trial-and-error procedure. The simplest way to account for the observed non-linearities in the relationships is to make transformations to the predictors. The relationship could also be linearized by making transformations to the response, but this should be avoided if possible.

With single-predictor regression, an initial guess for the model shape can be found by plotting the response against predictor. If the relationship is concave, then a concave transformation to the predictor would linearize the relationship. With convex relationship, a convex transformation can be used. Commonly used convex transformations are $e^x$ and $x^2$. Commonly used concave transformations are $\ln(x)$, $\sqrt{x}$ and $(-1/x)$. With more complex relationships, the same predictor can be included twice, with different transformations. If the relationship is good, a plotting model residuals against
the predictors should not show any trend. To recognize the trends, one can use lowess smoothers. One alternative is to use the approach that was presented in example 2.8, i.e., plotting the residuals against the predictor, and computing class means of the residuals, and plotting their confidence intervals. If roughly 95% of the confidence intervals overlap x-axis and no clear trend can be found in class means, I would believe that the model form is good. Function mywhiskers() of example 2.8 with option se=TRUE can be used to perform such analysis.

The above rules can also be used with multiple regression. In multiple regression, the response should be first plotted against all potential predictors to get a view on the shape of the relationship. The fitted model should be further analyzed by plotting the residuals against all the predictors, and checking if there are nonlinear trends in the residuals. Lowess smoothers or function mywhiskers can be used to detect also those trends.

The nonlinearity can also be tested through a formal test. In such a test, two alternative regressions are fitted: The null model with the predictor $x$ included as a continuous predictor and the alternative model in which all $k$ distinct values of $x$ are included as an indicator (dummy variable). Testing the performance of alternative model against the null model using a F-test gives a test for nonlinearity.

$$H_0 : \text{Model is linear}$$
$$H_1 : \text{Model is nonlinear}$$

$$F_{obs} = \frac{(RSS_0 - RSS_1)/(k - 2)}{RSS_1/(n - k)}$$

where $RSS_0$ is the residual sum of squares from the null model and $RSS_1$ that of the alternative model. The p-value is obtained from F-distribution with $k - 2$ and $n - k$ degrees of freedom.

It is quite common that none of the commonly used transformations linearizes the relationship. In such case, one could try Box-Cox transformations (Box and Cox 1962), which is a generalized transformation giving the most commonly used transformations as special cases. Another approach is spline regression. Harrell (2001, p. 20-34) presents an easy-to-use restricted cubic spine approach, which is able to produce various shapes of relationship. A third approach is the nonlinear regression, where no restrictions are made for the function of the relationship.

**Example 2.24.** Harrells spline regression

### 2.4.2 Independence of observations

The independence of observations should be thought about when collecting the data. Maybe the most important sources causing dependence among observations are spatial
and temporal autocorrelation.

In forestry, the spatial autocorrelation often arises from that trees that are close to each other tend to be more similar to each other than distant trees. If the distances between trees are known, the spatial autocorrelation can be modeled through a correlogram, variogram and covariogram, which are related to each others in a similar way than variance, covariance and correlation are related to each other. The covariogram is constructed by fitting an OLS regression to the data, and then plotting a the cross-products of all possible pairs of residuals against the geographical distance between these two observations. The covariogram is then obtained by fitting an appropriate (decreasing) function to these data. If such a trend is found, then spatial autocorrelation exists among the observations. The covariogram can be used to estimate the covariance between any two observations with known distance, which are then used in GLS approach to account for the autocorrelation. The estimated covariances (up to a scaling constant) can be written to matrix $V$, and the model can be then fitted using generalized least squares. This approach, which is clearly one special case of the linear model, is known as kriging in spatial statistics (e.g., Cressie 1993).

The temporal autocorrelation can be analyzed and taken into account in analogous way than the spatial autocorrelation. We just have the geographical distance replaced with the time difference between the observations. Such approach leads to analysis of lagged residuals, which are used to fit an appropriate model for the temporal autocorrelation. After the autocorrelation model has been estimated, GLS approach is used to fit a linear model that properly accounts for the temporal autocorrelation.

Another cause of interdependencies among observations is hierarchy of the data. For example, branches may be sampled from different trees for determination of biomass, the branches of one tree being somehow similar to each other. As an another example, trees may be measured from different stands, trees of one stand being similar and different from trees of other stands. Furthermore, repeated measurements may be taken from several plots, observations of one and the same plot at different points in time being similar and different to observations of other plots. All these examples lead to a situation where the data consists of several groups, and observations within the same group are correlated. This kind of structure can be taken into account through variance component modeling, which is dealt in more detail in the next chapter. In the variance component approach, a constant correlation is assumed among observations from different groups.
2.4. CHECKING THE VALIDITY OF ASSUMPTIONS

2.4.3 Constant residual variance

In ordinary least squares, the variance of residuals is assumed to be constant. The validity of this assumption can be taken into account by plotting the residuals of a fitted OLS model against the fitted values. The obtained plot should not show any trend in the variance. However, it is not always easy to see if there is trend in the variance or not, especially if the observations are not distributed evenly along the x-axis, having the fitted values. To help with this problem, one approach is to compute standard deviations of the residuals in different classes of fitted value, and compare them to each other. If the standard deviations are equal and do not show any trend with respect to the fitted values, the assumption about constant variance seems to be met. An example of such analysis was done in example 2.8 using the function `mywhiskers` with option `se=FALSE`.

A formal test can also be conducted for testing the heteroscedasticity (see Weisberg2005).

The violation of the constant variance can be corrected either by making a transformation to the response or through the use of a variance function. A very usually applied transformation is the logarithmic transformation, which is justified by the log-normal distribution for variables getting only positive values. A straightforward way for using a variance function is to fit OLS model, then model the variance using the squared residuals, and refit the model with GLS using the estimated variance function. An example of such procedure was shown in example 2.12. To check if the utilized variance function properly accounts for the heterogeneous variance, a plot of standardized residuals should express a constant variance against the response.

A problem with the transformation approach is that it also has an effect on the relationship between predictors and response. Thus, a transformation should be found that both homogenizes the variance and results in linear relationship between response and predictor. Furthermore, making transformation causes that the model gives unbiased predictions for the transformed variables (e.g., for the logarithmic height), implying that the prediction for the back transformed variable (e.g., for the total height) is biased. For these reasons, I would recommend using variance functions rather than making transformations into the response to homogenize the residual variance. If the transformation approach is used and normality of residuals of the fitted model are assumed, the bias can be corrected by adding half of the residual error to the prediction before backward transformation. This bias correction is based on the expected value of lognormal random variable $EX = e^{\mu + \sigma^2/2}$.
2.4.4 Normality

As noted several times before, the normality of residuals is needed only for testing purposes. Thus, if the previously stated assumptions are met, normality is not needed e.g. for the OLS or GLS estimates $\hat{b}$ to be best linear unbiased estimators for $b$. However, all t- and F-test on the regression relationship require normality. Thus, the normality becomes important for example, in studies where the linear model is used for testing the impacts of predictors to the response.

The very general statistical results (central limit theorem and law of large numbers) can be used as a justification to assume that distribution of any random variable is normal, unless it is clearly violated. In regression analysis, the normality can be checked using quantile-to-quantile plots (qq-plots), which plot the ordered residuals against corresponding quantiles of standard normal distribution. If the qq-plot are close to a straight line, the residuals follow the normal distribution.

2.5 Analysis of variance

Example 2.25. Data Immer of package MASS includes observations of barley yield of 4 varieties on six farms in years 1931 and 1932. The variables in the data are: Farm, Origin, Yield in 1931 and Yield in 1932, respectively. We want to study whether the varieties differ in terms of yield. However, we also allow possible effect for the farm.

```
> library(MASS) # The library of book "Modern applied statistics with S and S-Plus"
> data(immer)
> summary(immer)

Loc Var Y1 Y2
C :5 M:6 Min. : 69.10 Min. : 49.90
D :5 P:6 1st Qu.: 87.42 1st Qu.: 76.85
GR:5 S:6 Median :102.95 Median : 92.95
M :5 T:6 Mean :109.05 Mean : 93.13
UF:5 V:6 3rd Qu.:124.60 3rd Qu.:107.35
W :5 Max. :191.50 Max. :147.70

> head(immer)

Loc Var Y1 Y2
1 UF M 81.0 80.7
2 UF S 105.4 82.3
3 UF V 119.7 80.4
4 UF T 109.7 87.2
5 UF P 98.3 84.2
6 W M 146.6 100.4
```

If the class of a variable is factor, R automatically treats the different values as separate levels. If needed, you could define a variable as factor by using `immer$Var <- as.factor(immer$Var)`. However, here the Farm and Variety have already been defined as factor.

```
> class(immer$Var)
[1] "factor"
> class(immer$Loc)
[1] "factor"
```

We compute the mean yield into a new variable called Y. Figure 2.11 shows the yield as a function of Variety and Farm.
2.5. ANALYSIS OF VARIANCE

![Boxplot of Mean Barley yield on Variety and Farm (Loc) in the dataset immer of package MASS.](image)

Figure 2.11: A boxplot of the Mean Barley yield on Variety and Farm (Loc) in the dataset *immer* of package MASS.
We fit linear models having both Farm and Variety as predictors, as well as models having only Farm and only Variety as a predictor. The graphs of residuals in Figure 2.12 do not indicate any problems in the model assumptions. Therefore, we proceed to testing the effects of Variety and Farm.

The significance of the dropped predictor can be tested using `anova`.

```r
> immer$Y <- (immer$Y1 + immer$Y2) / 2
> mod1 <- lm(Y ~ Var + Loc, data = immer)
> mod2 <- lm(Y ~ Loc, data = immer)
> mod3 <- lm(Y ~ Var, data = immer)
> anova(mod1, mod2)

Analysis of Variance Table
Model 1: Y ~ Var + Loc
Model 2: Y ~ Loc
Res.Df RSS Df Sum of Sq F Pr(>F)
1 20 2216.5
2 24 4871.5 -4 -2655 5.9891 0.002453 **
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> anova(mod1, mod3)

Analysis of Variance Table
```
2.6. PREDICTION

The results showed significant effects for both Variety and Location. To see separate effects of each variety and location, we print the summary of the model. Now the constant is the yield of variety M at location C. The other coefficients show the difference to this combination. The t-tests of coefficients therefore test whether the difference to the default variety/location is different from zero. To specify more meaningful tests, one should specify the meaning of intercept in different way. This could be done using function `contrasts`.

```r
> summary(mod1)
Call:
  lm(formula = Y ~ Var + Loc, data = immer)
Residuals:
   Min     1Q Median     3Q    Max
  -22.070  -5.894  -1.068   7.146  16.450
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 102.2020    6.0781 16.815 2.88e-13 ***
VarP         8.1500    6.0781  1.341 0.194983
VarS        -3.2580    6.0781  -0.536 0.597810
VarT        23.8080    6.0781  3.917 0.000854 ***
VarV         4.7920    6.0781  0.788 0.439728
LocD       -26.0600   6.6578  -3.914 0.000860 ***
LocGR      -28.3400   6.6578  -4.256 0.000386 ***
LocM        -3.5900   6.6578  -0.539 0.595705
LocUF      -16.0100   6.6578  -2.405 0.025996 *
LocW       27.1400    6.6578   4.076 0.000589 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 10.53 on 20 degrees of freedom
Multiple R-squared: 0.8568,  Adjusted R-squared: 0.7924
F-statistic: 13.3 on 9 and 20 DF,  p-value: 1.216e-06
```

Finally, the same analysis could be done using function `aov`.

```r
> summary(aov(Y~Var+Loc,data=immer))
Df  Sum Sq Mean Sq F value Pr(>F)
Var   4 2655  663.7  5.989 0.00245 **
Loc  5 10610 2122.1 19.148 5.21e-07 ***
Residuals 20 2217 110.8
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

2.6 Prediction

With the linear model, prediction for a new individual with predictors $X^*$ is obtained as $\hat{y} = \hat{b}X^*$.

The prediction includes errors caused by
CHAPTER 2. LINEAR MODEL

- model uncertainty,
- estimation errors of the parameters, \( \text{var}(\hat{b}) \), and
- residual variance.

The variance-covariance matrix of prediction, that accounts for the residual and parameter uncertainty, is given as

\[
\text{var}(\hat{y}) = \sigma^2 + \sigma^2 X_0' X^{-1} X_0' = \sigma^2 + X_0 \text{var}(\hat{b}) X_0
\]

The confidence interval for the prediction can be formulated based on the prediction variance and t-distribution.

**Example 2.26.** We want to make predictions from model \( fm1 \) for diameters \( 25, \ldots, 30 \).

The code below defines the matrix \( X_0 \) for these predictions.

```r
> Xstar<-cbind(1,25:30)
> attributes(fm1) # look for attributes that include the desired matrices and vectors
$names
[1] "coefficients" "residuals" "effects" "rank"
[5] "fitted.values" "assign" "qr" "df.residual"
[9] "xlevels" "call" "terms" "model"
$class
[1] "lm"
> attributes(summary(fm1)) # cov unscaled includes the matrix solve(X%*%X)
$names
[1] "call" "terms" "residuals" "coefficients"
[5] "aliased" "sigma" "df" "coefficients"
[9] "adj.r.squared" "fstatistic" "cov.unscaled"
$class
[1] "summary.lm"
> sigma<-summary(fm1)$sigma
> varb<-sigma^2*summary(fm1)$cov.unscaled
>
> varh<-diag(rep(sigma^2,dim(Xstar)[1]))+Xstar%*%varb%*%t(Xstar)
>[1,] 1.373239251 0.116465081 0.07537208 0.03427907 -0.006813936 -0.047906941
[2,] 0.116465081 1.308502499 0.06917758 0.04553384 0.021890096 -0.001753651
[3,] 0.075372075 0.069177588 1.27866427 0.05678861 0.050594127 0.044399640
[4,] 0.034279070 0.045533842 0.05678861 1.28372455 0.079298158 0.136706221
[5,] -0.006813936 0.021890096 0.05059413 0.07929818 1.323683354 0.136706221
[6,] -0.047906941 -0.001753651 0.04439964 0.07929818 0.136706221 1.398540676
> cov2cor(varh)
```

The variance-covariance matrix is not a diagonal, but the predictions are slightly correlated. The correlations arise from the estimation errors of the parameters: all predictions are based on the same estimates of \( \hat{b} \), and thus are correlated due to this. However, the correlations are not very strong, as indicated by the correlation matrix shown in the lower matrix. Thus, we ignore these correlations and plot the confidence intervals as if the predictions were uncorrelated. The confidence intervals are shown in Figure 2.13. The narrow intervals plotted using dashed line do not take into account the estimation errors of parameters. The wider intervals shown by the solid lines are the prediction intervals that also take into account the estimation errors.
2.7 Estimation using maximum likelihood

The previous section presented how the parameters of the linear model can be estimated using methods based on least squares. In this section, we make an assumption about the distribution of the random variable \( y \), and estimate the parameters using the method of maximum likelihood. Section 1.5.2 presented the general idea of maximum likelihood.

In the ML estimation of the linear model, the response \( y \) is assumed to be normally distributed. Furthermore, parameter \( \mu \) is not assumed to be constant, but it is written as a linear function of predictors as \( \mu = \mu(Xb) \). The likelihood (or log-likelihood) will then be a function of \( b \) and \( \sigma^2 \), and the ML estimate of them is the combination that maximizes the (log-)likelihood.

Letting also \( \exp(\sigma^2) \) to be a function of predictors (say, \( \exp(\sigma^2) = f(X, c) \)), we can fit a model with heteroscedastic residuals. The exponential transformation is used to constraint the \( \sigma^2 \) to be positive. This is based on the invariance property of ML-estimator, and was used also in example 1.44. Both the parameters \( b \) and \( c \) can be estimated by maximizing the (log-)likelihood on \( b \) and \( c \). Furthermore, we may define
any positive definite structure for $V$, parameterize it and estimate the parameters by maximizing the likelihood on all the specified parameters.

We could also assume some other distribution for $y$ than the normal. For example, $y_i$ may be a binary random variables having the Bernoulli$(p)$ distribution. Such a variable would get only values 0 and 1, the value 0 indicating absence and value 1 presence of the characteristic being modeled. Furthermore, if $y$ are counts, a realistic assumption could be the Poisson$(\lambda)$ distribution. In those cases, the parameters of the assumed distribution are just written as a linear combination of predictors, (e.g., $\logit^{-1}(p) = f(Xb)$ in the case of bernoulli distribution and $\exp(\lambda) = \lambda Xb$ in the case of Poisson distribution. The functions $\logit^{-1}$ and $\exp$ are link functions which are used to ensure that the parameter is within the parameter space, i.e., $0 \leq p \leq 1$ and $\lambda \geq 0$. The parameters are then estimated by maximizing the log-likelihood on $b$. Non-normal models where a function of parameters is assumed to be linear in $b$ are called generalized linear models, and are dealt more in detail in chapter 5.

### 2.7.1 ML for the single predictor regression

Let us state model (2.2) in a bit different form:

\[
E(y_i) = \mu(x_i) = b_0 + b_1 x_i \\
y_i \sim \text{indep.} \mathcal{N}[\mu(x_i), \sigma^2] \quad i = 1, 2, \ldots, n
\]

where $\mu(x_i)$ indicates that the expectation of $y$ is a function of $x$. The four important assumptions incorporated in the models are (i) The $y_i$ follow a normal distribution, (ii) are independent, (iii) have equal variance, and (iv) the mean of $y$ is linear in $x$. Compared to the least squares assumptions, the only difference is the assumption on normality of $y$.

The likelihood based on normal distribution is

\[
L = \prod_i f_N(\mu(x_i), \sigma) \\
= \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2} (y_i - \mu(x_i))^2\right] \\
= \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2} (y_i - b_0 - b_1 x_i)^2\right]
\]
2.7. ESTIMATION USING MAXIMUM LIKELIHOOD

Taking logarithms yields the log likelihood as

\[ l = \ln L = \sum_i \left[ -\ln (2\pi\sigma^2)^{1/2} - \frac{1}{2\sigma^2} (y_i - b_0 - b_1 x_i)^2 \right] \]

\[ = -\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_i (y_i - b_0 - b_1 x_i)^2. \]

Differentiating the log likelihood with respect to the three parameters \((b_0, b_1, \sigma^2)\) gives the following equations

\[ \frac{\partial l}{\partial b_0} = \frac{1}{\sigma^2} \sum_i (y_i - b_0 - b_1 x_i) \]

\[ \frac{\partial l}{\partial b_1} = \frac{1}{\sigma^2} \sum_i (x_i y_i - b_0 x_i - b_1 x_i^2) \]

\[ \frac{\partial l}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_i (y_i - b_0 - b_1 x_i)^2 \]

Setting these equations to 0 and solving for the three parameters gives the ML estimators. Start with the lowest equation by solving it for \(\sigma^2\). Setting it equal to 0 gives

\[ \hat{\sigma}^2 = \frac{1}{n} \sum_i \left( y_i - \hat{b}_0 - \hat{b}_1 x_i \right)^2, \]

which shows the ML estimator for the residual variance, given the estimates for \(b_0\) and \(b_1\). It also shows that \(\hat{\sigma}^2\) is always positive and greater than zero, unless predictions and observations are exactly the same. Thus, the equations setting the two upper equations to zero can be multiplied with \(\sigma^2\) which yields exactly same system of equations that was solved to get the OLS estimates. Thus, the ML-estimators

\[ \hat{b}_0 = \bar{y} - b_1 \bar{x} \]

\[ \hat{b}_1 = \frac{\frac{1}{n} \sum_i x_i \sum_i y_i - \sum_i x_i y_i}{\frac{1}{n} (\sum_i x_i)^2 - \sum_i x_i^2} \]

are exactly same as the OLS-estimators. The estimator for the residual variance, \(\hat{\sigma}^2 = \frac{RSS}{n} \) differs from (2.3) in that nominator \(n - 2\) is replaced by \(n\). Thus, the ML estimate of residual variance is biased downwards.

2.7.2 ML for the linear model with uncorrelated errors and constant variances

As with LS methods, the results on simple linear regression generalize to the multivariate case. For estimation of the linear model with the method of maximum likelihood,
the model with uncorrelated errors and constant variance is defined as

$$\mathbf{y} \sim N[\mathbf{Xb}, \sigma^2 \mathbf{I}]$$

The likelihood is based on the multivariate normal distribution:

$$L = L(b, \sigma^2|\mathbf{y}) = \exp \left[ -\frac{1}{2} \mathbf{(y - Xb)}'(\mathbf{y - Xb}) / (a \pi \sigma^2)^{1/2n} \right]$$

The log likelihood is

$$l = -\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma^2 - \frac{1}{2} \mathbf{(y - Xb)}'(\mathbf{y - Xb}) / \sigma^2.$$ 

The partial derivatives with respect to \(b\) and \(\sigma^2\) are

$$\frac{\partial l}{\partial b} = \frac{\mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{Xb}}{\sigma^2} \quad \frac{\partial l}{\partial \sigma^2} = \frac{(\mathbf{y - Xb})'(\mathbf{y - Xb})}{2\sigma^2} - \frac{n}{a \sigma^2}$$

Setting the upper equation to zero gives the ML estimator for \(b\) as \(\hat{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}\), which is identical to the OLS estimator. From equating the lower derivative to zero and solving for \(\sigma^2\) yields the ML estimator for residual variance, which is \(\hat{\sigma}^2 = \frac{1}{n}(\mathbf{y - Xb})'(\mathbf{y - Xb})\). This is identical to the estimator (2.6), except for the important replacement of \(n - p\) by \(n\) in the denominator.

### 2.7.3 ML for LM with a general residual variance structure

The more general model, corresponding to the GLS model, is

$$\mathbf{y} \sim N[\mathbf{Xb}, \mathbf{V}]$$

The log likelihood is

$$l = -\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln |\mathbf{V}| - \frac{1}{2} \mathbf{(y - Xb)}'\mathbf{V}^{-1}(\mathbf{y - Xb}),$$

The derivation for the ML estimators can be found from McCulloch and Searle (2001, p. 178-181). As one could guess, the results are identical to the results on GLS, except for that the denominator of (2.10) is replaced with \(n\).

As a summary, the method of maximum likelihood is equivalent to OLS and GLS approaches in estimating the parameters of a linear model, except that it results in downward biased estimates for the residual variance. The bias is higher with smaller datasets, and vanishes with very large datasets.

### 2.7.4 Restricted maximum likelihood

It has been seen that the LS and ML principles lead to identical results for estimation of \(b\), except for that LS principle is based on more relaxed assumptions about the distribution of residuals (or distribution of \(\mathbf{y}\)). However, with the variance of the residual
error, these two methods differ in the denominator of the estimator for residual variance. The estimator obtained from OLS is unbiased, whereas the ML estimator is an underestimate. The Restricted or Residual ML (REML) approach is such aversion of the ML that does not have this problem, and that is why it has become one of the most widely applied estimation methods for linear mixed models and variance components models. However, it can (and should) also be applied with other models, especially if the interest lies on tests on the structure of matrix $V$. The REML method has been described in more detail in chapter 3.

2.8 About modeling strategies

2.8.1 Selection of predictors

There are several ways for prediction of predictors to a regression model. The researcher should always use his/her own expertise in selecting the predictors. If a theory exists on the relationship between the variables, the theory should be used in selecting the predictors and also the shape of the model. Automated procedures, such as stepwise backward or forward algorithms should be avoided, and reasons for having the selected predictors in the model should be searched for. Plotting residuals, predictors and responses in all possible ways always helps, for example, in detecting peculiar observations that may cause unexpected predictors to be significant.

2.8.2 The purpose of modeling

It is important for a researcher to recognize the aim of the modeling. With different research problems, different assumptions of the model become fundamental. As an example let us consider two distinct uses of regression model: explaining and prediction.

Explaining is used, for example, when the effect of a new silvicultural method on the growth of a sapling is evaluated. With such a model, the interesting research questions are (i) does the method have an impact on the growth and (b) how big is the impact. In this situation, the regression model is used in a similar fashion as the analysis of variance or analysis of covariance. When regression model is used for explanation or for testing interesting effects, study of the assumptions behind the test of regression coefficients is very crucial. It is, for example, quite common to make transformations to the response in order to make the residuals normal and homogeneous. If the assumptions are not met, also the test may be biased. Also high correlation among the predictors may cause problems. Of two highly correlated predictors, it may be impossible to say which one has an effect on the response or which not. It is also a quite common situation that the highly correlated predictors are insignificant predictors.
when used alone, but including them both may give very high significance to them.

The way of selecting the predictors is also important with models used for explanation and testing purposes, as well as documenting all utilized predictors; also those that were found insignificant. For example, suppose we have response variable \( y \) and data where we have 100 potential \( x \)-variables, of which none is a significant predictor in reality. However, by testing the significance of all these predictors, one at a time, at the risk level of 0.05, the expected number of predictors with \( p \)-value of 0.05 or less is 5 just due to the definition of \( p \)-value. Reporting only those 5 variables without mentioning the 95 insignificant predictors would lead to very different inference than reporting of all the tests.

In prediction, the aim is to find a function that as accurately as possible predicts the expected value of the response with given values of the predictors. With this model, different assumptions become crucial. With prediction, the model form needs to be chosen with care. Especially, the behavior of the selected function needs to be analyzed so that it would not lead to unrealistic results when applied beyond the range of the modeling data. Such a risk is especially high if the model includes polynomial terms. Furthermore, with predictive models, making transformations to the response causes biased predictions in the original scale, correction of which requires assumptions on the distribution. The bias correction is also quite vulnerable to a violation in the assumption.

On the other hand, violation of the assumptions about the normality and homogeneity of residuals is not that big a problem with predictive models: the estimates are still unbiased, even though not the best possible. The violations in these assumptions may just lead to problems in testing, leading to including predictors that are insignificant in reality. Including insignificant predictors is possible especially if the predictors are selected from among a large set of potential predictors through trial-and-error procedure, or using a automatic rule, such as backward or forward procedure using large datasets. However, these predictors usually have very small coefficient, meaning that they do not have a big effect on the predictions. However, including extra predictors may not lead to bad predictions, even though it may cause extra work in applications, where the values of all predictors need to be known for the prediction to be possible.

### 2.9 Exercises

1. Data `cherry.txt` give the volume (cubic feet), height (feet) and diameter (inches) (at 54 inches above ground) for a sample of 31 black cherry trees in the Allegheny National Forest, Pennsylvania. The data were collected in order to find an estimate for the volume of a tree (and therefore the timber yield), given
its height and diameter (Hand et al. 1994). Start your model development from relationship $V_i = b_0 D_i^{b_1} H_i^{b_2}$.

(a) Linearize the model using an appropriate transformation. Specify the relationship between original parameters and the parameters of the linearized model.

(b) Fit the model using `lm`.

(c) Define the model matrix $X$ and response vector $y$ for this model and estimate the parameters using matrix operations. Confirm that you got the same values as returned by function `lm`.

(d) Compute the hat matrix $H$, and estimate the residual variance $\hat{\sigma}^2$. Confirm that you got the same values as returned by function `lm`.

(e) Compute the standard errors of parameter estimates by first computing matrix $\hat{\sigma}^2(X'X)^{-1}$, and then taking square root of its diagonal elements. Compute absolute t-statistics and p-values. Compare to `coef(summary(model))`.

(f) Compute the model residuals as $e = y - Xb$. Compute $R^2$ for the model, $F$-statistic and corresponding p-value.

(g) Based on that volume is determined by three dimensions, and diameter and height by one dimension, specify a justified null hypothesis about the values of $b_1$ and $b_2$. Using argument `offset` in function `lm`, fit a null model. Using function `anova` test whether the full model is significantly better than the null model.

2. Data `plot48.txt` includes data from one measured plot. The variables in the data are: puunro=tree id, pl=tree species (1=pine), xk=x-coordinate of the tree location in the stand, yk=y-coordinate of tree location in the stand, d=tree diameter (cm), h=tree height (m), t=tree age, ig1=basal area growth during the coming 5 year period, id1=diameter growth in the coming 5 year period, ig2=past basal area growth, id2=past diameter growth. The aim is to model the coming basal area growth (ig1) of a tree using information on past growth, and current age, diameter and height of the tree.

(a) Graphically explore the relationships of potential explanatory variables and ig1. What would you do with the one outlier of the data.

(b) Start from model $ig1_i = b_0 + b_1*ig2_i + e_i$). Check the validity of your assumptions, and find a model that best fulfills the assumptions.

(c) Include other predictors and test whether they improved the model or not. Report your final model.
(d) Extra. Using different models for spatial autocorrelation structures, test whether the model could be improved by taking into account the spatial autocorrelation.

3. Solve LS estimators \( b_0 = \bar{y} - b_1 \bar{x} \) and \( b_1 = \frac{\frac{1}{n} \sum_i x_i \sum_i y_i - \sum_i x_i y_i}{\frac{1}{n} \sum_i x_i^2 - \sum_i x_i} \) from equations

\[
-2 \sum_{i=1}^{n} y_i + 2nb_0 + 2b_1 \sum_{i=1}^{n} x_i = 0 \\
\sum_{i=1}^{n} 2y_i x_i - 2b_0 \sum_{i=1}^{n} x_i - 2b_1 \sum_{i=1}^{n} x_i^2 = 0
\]

Furthermore, show that the solution for \( b_1 \) is equivalent to the more commonly used form \( b_1 = \frac{S_{xy}}{S_{xx}} \) where \( S_{xy} = \sum_i (x_i - \bar{x})(y_i - \bar{y}) \) and \( S_{xx} = \sum_i (x_i - \bar{x})^2 \)
Chapter 3

Linear mixed models

The mixed effect model is a model for populations consisting of groups and individuals within groups. For example, the group may be a sample plot, and the individual may be a tree within the plot. As another example, individuals may be stands within a municipality, municipalities representing the groups. Such populations, called hierarchical populations, could be modeled either by using mixed models or by using fixed effect models.

In the fixed effect model, we could fit a model including a binary indicator variable describing if the individual belongs to a certain group or not. The coefficients would then be estimated for each group in the data. Essentially, we would assume that all the groups are represented in the modeling data, or we are interested only in those particular groups that are included. However, we would not be able to make inference on groups that are not included in our data, or on the variation between groups.

If the groups represented in our data represent only a sample from a population of groups, and if we are interested in the variation among groups, then the mixed model approach would be preferred. Such situation appears, for example, when the data consists of trees within sample plots. All possible plots would not be included in the data, and it is thus realistic to assume that the data only represents a sample from a large, probably infinite population of sample plots. In a mixed model, the total residual variation of the observations is divided to within-plot and between-plot variation. However, after estimating these variance components, we can use the observations of our data to predict also effects for individual groups. These effects compare to the group-specific coefficients of the fixed effect-model.

In contrast to the fixed-effect model, with random-effects models we can also predict effects for groups that were not included in the original modeling data, if we have some observations on individuals within that group. This makes the random effects models highly applicable in many situations in forestry. For example, an estimated
height-diameter model can be calibrated for a new stand using observed height(s) of sample tree(s) from that stand. With fixed-effect model, we would need several sample trees to fit a model for a stand, whereas a mixed-effect model can be calibrated even with only one sample tree.

### 3.1 Single level of grouping

#### 3.1.1 The variance component model

A variance component model for variable \( y \) of individual \( i \) in group \( k \) is

\[
y_{ki} = \mu + a_k + e_{ki} \tag{3.1}
\]

where \( \mu \) is a fixed population mean and part \( a_k + e_{ki} \) includes the random parameters (Pinheiro and Bates 2000). It divides the residual error of linear model into two mutually independent parts, to a random group effect for group \( k \) and into a random residual for individual \( i \) of group \( k \). It is assumed that the group-effects and residual are independent, normally distributed random variables with

\[
a_k \sim NID(0, \sigma^2_a)
\]

and

\[
e_{ki} \sim NID(0, \sigma^2)
\]

The parameters of this mixed-effects models are \( \mu, \sigma^2_a \), and \( \sigma^2 \)

**Example 3.1.** Data `spati` includes tree-specific measurements from 66 Scots pine stands in North Carelia, collected by Timo Pukkala and Jari Miina for growth modeling purposes. The variables in the dataset are `plot=plot id, X,Y=geographical location of the sample plot, N=number of stems on the plot, G=basal area on the plot, V=plot volume, Dg, Basal area median diameter, Hg=height of basal area median tree, Tg=age of basala area median tree, Hdom=dominant height, maos=proportion of pines, kuos=proportion of spruces, kanro=plot id, puunro=tree id, pl=tree species, xk,yk=location of the tre within the plot, d=tree diameter, h=tree height, ttree age, dk=stump diameter, X2b=?, id1=diameter growth for the coming 5 years, id2=past diameter growth.

We first fit a simple fixed-effect model for the coming diameter growth. The growth is modeled through sample mean, i.e., the model includes the constant term only

\[
idl_1 = b_0 + e_.
\]

Even though this model might not make sense in practice, it helps us to understand the difference between fixed-effects and mixed-effects models.
3.1. SINGLE LEVEL OF GROUPING

> spati<-read.table("d:/laurim/biometria/spati.txt")
> lm1<-lm(id1~1, data=spati)
> summary(lm1)

Call: lm(formula = id1 ~ 1, data = spati)

Residuals:
    Min     1Q Median     3Q    Max
-15.401 -6.651 -2.731  3.849  57.849

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.2307    0.0980    145.1 <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.929 on 10254 degrees of freedom

The estimate for the constant is 14.23, i.e., the mean of id1 over the whole data:

> mean(spati$id1)
[1] 14.23069

The residual standard error, 9.929, is just the sample standard deviation of id1.

> sd(spati$id1)
[1] 9.929181

Thus, it describes how much the individual tree growth varies around the average growth over the whole data.

However, one could expect that trees of same plot have similar growth pattern, whereas trees from different plots may have different pattern. This variation may be due to site factors (e.g., site fertility) and properties of the growing stock (e.g., stand age), but we ignore these factors and just explain the variation by stand-specific dummy variables. This leads to the following fixed-effects model

\[ id1_{ki} = b_k + \epsilon_{ki} \]

for the growth of tree \(i\) in stand \(k\). The model can be fitted using

lm2<-lm(id1~as.factor(kanro), data=spati)
> summary(lm2)

Call: lm(formula = id1 ~ as.factor(kanro), data = spati)

Residuals:
    Min     1Q Median     3Q    Max
-31.7418 -3.2708 -0.1733  2.7553 36.4060

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.0826     0.5004   14.152  < 2e-16 ***
as.factor(kanro)2  1.2353     0.7252    1.703   0.08852  .
as.factor(kanro)3  0.1169     0.7007    0.167   0.86752 1
as.factor(kanro)4 -0.2866     0.6597   -0.434   0.66400 3
...<part of the output was removed>

Residual standard error: 5.505 on 10189 degrees of freedom
Multiple R-Squared: 0.6946, Adjusted R-squared: 0.6926
F-statistic: 356.5 on 65 and 10189 DF, p-value: < 2.2e-16
Instead of the previous model where the fitted value was the sample mean over the whole data, we now have one specific parameter for each stand. As discussed in earlier chapters, using factors as explanatory variables leads to dropping of one class away from the model. Then the constant of the model is the mean value for that class. The other coefficients of other levels show difference between the level in question and the dropped level. In our case, R selected to drop the level for plot 1. Thus, the constant of our model is the sample mean for stand 1. Adding the coefficient of stand 2 to the constant gives just the sample mean of stand 2.

```r
> mean(spati$id1[spati$kanro==1])
[1] 7.082562
> coef(lm2)[1]
(Intercept)  
7.082562
> mean(spati$id1[spati$kanro==2])
[1] 8.317909
> coef(lm2)[1]+coef(lm2)[2]
(Intercept)  
8.317909
```

Thus, we were able to take the hierarchy of the data into account by including the classes as predictors into our model. However, in many cases we might be interested in analyzing the variation between stands and within stands, not only on the stands that happen to be included in the data. In that case, we would fit a variance component model

\[ id_{ki} = \mu + a_k + e_{ki}, \]

where \( \mu \) is the grand mean over all stands, \( a_k \) is a random stand effect, and \( e_{ki} \) is the random residual. The mixed-effects model is fitted using

```r
> library(nlme)  # this needs to be run only once at each R session
> lm1<-lm(id1˜1,random=˜1|kanro,data=spati)
> summary(lm1)
Linear mixed-effects model fit by REML
Data: spati
AIC  BIC logLik
64471.48 64493.19 -32232.74
Random effects:
Formula: ˜1 | kanro
(Intercept) Residual
StdDev: 8.429983 5.504926
Fixed effects: id1 ˜ 1
Value Std.Error  DF  t-value p-value
(Intercept) 13.79040 1.039348 10189 13.26833 0
Standardized Within-Group Residuals:
          Min Q1 Med Q3 Max
Standardized Residuals: -5.74989246 -0.59583705 -0.03261568 0.49812371 6.61831353
Number of Observations: 10255
Number of Groups: 66
```

The output shows that the mean diameter growth is 13.79040. The estimates of variance components show that even though both models take into account the distribution of observations into sample plots, the variance component model assumes that the data is a random sample of the population of stands. Thus, the random effects \( a_k \) for the stands we have in our data are just a sample from a distribution of random effects. The data we have can be used for making inference on this distribution. There are only three parameters in the model: \( \mu \), \( \sigma_a^2 \) and \( \sigma_e^2 \). In lm2, the stand effects are
3.1. SINGLE LEVEL OF GROUPING

fixed parameters, not realizations of a random variable, so we had a total of 66 parameters + the parameter $\sigma^2$. On the other hand, compared to $\text{lml}$, the mixed effect model divides the residual of model $\text{lml}$ into two distinct parts: to variation between stands, and to variation within stands. Thus, estimated residual standard error of $\text{lml}$ (9.929) is roughly equal to $\sqrt{\frac{\sigma^2_a}{k} + \sigma^2}$ in the mixed effects model (The difference is due the different assumptions in estimating.):

$$\sqrt{\text{sum}(8.429983^2, 5.504926^2)}$$

$$[1] 10.06821$$

Even though the stand effects are not estimated as the parameters of the variance component model, we may still be interested in them as well. For that purpose, the random effects can be predicted using BLP (see Section 3.5). The predictions can be obtained in R as follows

$$\text{ranef(lmm1)}$$

(Intercept)

1 -6.6842854
2 -5.4513622
3 -6.5687288
4 -6.9762890
5 -0.3633957
6 -2.6665063

which show the predicted difference between the grand population mean and stand mean. To predict the mean growth for the stands, the grand mean needs to be added into the random effects.

$$\text{fixef(lmm1)} + \text{ranef(lmm1)[1:5,]}$$


$$\text{coef(lmm1)[1:5,]}$$


These values are very similar to the sample means for stands 1-5. However, they are slightly shrunken towards the grand population mean $\mu = 13.79$.

3.1.2 Mixed-effects model with a random constant

The linear mixed model is a generalization of the variance component model. It includes both fixed and random parameters. With only one level of grouping and $k$ fixed predictors, it is defined as

$$y_{ki} = b_0 + b_1x_{1ki} + \ldots + b_px_{pki} + a_k + e_{ki}.$$  \hspace{1cm} (3.2)

Part $b_0 + b_1x_1 + \ldots + b_px_p$ is the fixed part, which has exactly the same meaning and interpretation as the fixed part of the linear model of the previous section had. That is,

$$a_k \sim NID(0, \sigma^2_a)$$

and

$$e_{ki} \sim NID(0, \sigma^2).$$

Part $a_k + e_{ki}$ is called the random part, which has exactly the same assumption and interpretation as the random part of a variance component model had. Compared to the
CHAPTER 3. LINEAR MIXED MODELS

variance component model (3.1), mixed model (3.2) differs only in that the fixed population mean has been replaced with a linear function of fixed predictors and parameters. Compared to the fixed-effects model, the residual of model (2.4), is partitioned into two parts in model (3.2).

The mixed model can be seen as such a special case of the multiple regression model, where observations of the same group are correlated. The covariance between observations \( i \) and \( i' \) of group \( k \) is

\[
\text{cov}(a_k + e_{ki}, a_k + e_{ki'}) = \text{cov}(a_k, a_k) + \text{cov}(a_k, e_{ki}) + \text{cov}(e_{ki}, e_{ki'})
\]

By reorganizing terms, the mixed-effect model 3.2 can be written as

\[
y_{ki} = (b_0 + a_k) + b_1 x_{1ki} + \ldots + b_p x_{pki} + e_{ki}.
\]

which shows that we are actually assuming that the constant of the model varies between groups, whereas the other coefficients are fixed.

**Example 3.2.** Consider the dependence of future diameter growth on the past diameter growth in data `spati`. Let us first plot the data, and add lines that show the plot-specific dependence of future growth on past growth (Figure 3.1, upper graph).

```r
> plot(spati$id2, spati$id1, col="red")
> plots<-unique(spati$kanro)
> for (i in 1:length(plots)) {
+ thisplot<-spati[spati$kanro==plots[i],]
+ model<-lm(id1~id2, data=thisplot)
+ xapu<-seq(min(thisplot$id2), max(thisplot$id2), length=10)
+ lines(xapu, coef(model)[1]+coef(model)[2]*xapu)
+ }
```

We see that the lines are at different levels, which means that for stands with similar past growth, the level of future growth varies. Thus, we could assume a mixed model with random constant

\[
id_{1ki} = b_0 + b_1 id_{2ki} + a_k + e_{ki}.
\]

The model is fitted and summary of it printed using

```r
> lmm2<-lme(id1~id2, random=~1|kanro, data=spati)
> summary(lmm2)
```

Linear mixed-effects model fit by REML

Data: spati

BIC logLik AIC

27053.3 27079.19 -13522.65

Random effects:

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name</th>
<th>Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>kanro</td>
<td>(Intercept)</td>
<td>3.787522</td>
</tr>
<tr>
<td>Residual</td>
<td></td>
<td>3.977081</td>
</tr>
</tbody>
</table>

Std Dev: 3.787522 3.977081
3.1. SINGLE LEVEL OF GROUPING

Figure 3.1: Plot of the future diameter growth on past growth.
**CHAPTER 3. LINEAR MIXED MODELS**

Fixed effects: \( id1 \sim id2 \)

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Std.Error</th>
<th>DF</th>
<th>t-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>2.7419192</td>
<td>0.5197030</td>
<td>4725</td>
<td>5.27594</td>
<td>0</td>
</tr>
<tr>
<td>id2</td>
<td>0.6986602</td>
<td>0.0089685</td>
<td>4725</td>
<td>77.90168</td>
<td>0</td>
</tr>
</tbody>
</table>

Correlation:

<table>
<thead>
<tr>
<th></th>
<th>(Intr)</th>
<th>id2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intr)</td>
<td>-0.292</td>
<td></td>
</tr>
<tr>
<td>id2</td>
<td></td>
<td>-0.292</td>
</tr>
</tbody>
</table>

Standardized Within-Group Residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Q1</th>
<th>Med</th>
<th>Q3</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-6.15423509</td>
<td>-0.49396582</td>
<td>-0.07426128</td>
<td>0.44280439</td>
<td>9.52845924</td>
</tr>
</tbody>
</table>

Number of Observations: 4785

Number of Groups: 59

Our estimates show that for an average sample plot, the future growth of tree \( i \) on plot \( k \) depends on the past growth according to

\[
\text{id1}_{ki} = 2.742 + 0.699\text{id2}_{ki}
\]

However, the level of the line varies so that the constant of the model varies around the value of 2.742 with standard deviation of 3.788. Comparing this figure to the residual standard error of the model (3.977) shows that about half of the residual variation seems to be within plots and half between plots.

The plot-specific fits can also be shown, after predicting the random effects of the model for the plots of the modeling data. To see the plot-specific fits, we make a plot that shows the fitted lines by plots (Figure 3.1, lower graph)

```r
> kert<-coef(lmm2)
> plots<-rownames(kert)
> plot(spati$id2,spati$id1,col="red",
+     xlab="Past diameter growth",ylab="Coming diameter growth")
> for (i in 1:length(plots)) {
+    thisplot<-spati[spati$kanro==plots[i],]
+    xapu<-seq(min(thisplot$id2),max(thisplot$id2),length=10)
+    lines(xapu,kert[i,1]+kert[i,2]*xapu)
+ }
```

### 3.1.3 Mixed-effects model with a random constant an slope

Also other parameters than the constant can be assumed to be random. This yields a special case of the mixed-effects model that has sometimes been called the random coefficient model. With the single predictor regression, the model becomes

\[
y_{ki} = (b_0 + a_k) + (b_1 + c_k)x_{ki} + e_{ki},
\]

where \( a_k \) and \( c_k \) are random, group effects which have bivariate normal distribution with mean 0 and variance-covariance matrix

\[
D = \text{var} \begin{pmatrix} a_k \\ c_k \end{pmatrix} = \begin{pmatrix} \text{var}(a_k) & \text{cov}(a_k, c_k) \\ \text{cov}(a_k, c_k) & \text{var}(c_k) \end{pmatrix}.
\]

The other assumptions are as they were before. It is also possible to make restrictions to the above structure, for example, by assuming that the covariance between random effects is 0.
3.1. SINGLE LEVEL OF GROUPING

Presentation 3.3 is a good way to show that we are actually assuming random constants and coefficients in a mixed modeling. However, it is often better to organize terms into fixed and random parts. Furthermore, we can add also other predictors to the model to get a general mixed-effects model for a population with one level of grouping

\[ y_{ki} = b_0 + b_1 x_{1ki} + \ldots + b_p x_{pki} + a_{0k} + a_{1k} x_{1ki} + \ldots + a_{qk} x_{qki} + e_{ki}, \] (3.5)

where \( b_0 + b_1 x_{1ki} + \ldots + b_p x_{pki} \) is the fixed part and \( a_{0k} + a_{1k} x_{1ki} + \ldots + a_{qk} x_{qki} + e_{ki} \) is the random part. The assumptions about them are as stated earlier.

**Example 3.3.** Assuming that also the slope of the model varies between stands leads to model

\[ id_{1ki} = b_0 + b_1 id_{2ki} + a_k + c_k id_{2ki} + e_{ki}. \]

This model was fitted and printed using

```r
> lmm3 <- lme(id1 ~ id2, random = ~id2 | kanro, data = spati)
> summary(lmm3)
```

Linear mixed-effects model fit by REML

Data: spati

AIC  BIC logLik
26651.57 26690.41 -13319.79

Random effects:
Formula: ~id2 | kanro
Structure: General positive-definite, Log-Cholesky parametrization

<table>
<thead>
<tr>
<th>StdDev</th>
<th>Corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>5.5286236 (Intr)</td>
</tr>
<tr>
<td>id2</td>
<td>0.1746247 -0.58</td>
</tr>
<tr>
<td>Residual</td>
<td>3.7617195</td>
</tr>
</tbody>
</table>

Fixed effects: id1 ~ id2

<table>
<thead>
<tr>
<th>Value</th>
<th>Std.Error</th>
<th>DF</th>
<th>t-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>1.8887165 0.7418434 4725 2.545978 0.0109</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id2</td>
<td>0.8052221 0.0258520 4725 31.147363 0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Correlation:

<table>
<thead>
<tr>
<th>(Intr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>id2 -0.594</td>
</tr>
</tbody>
</table>

Standardized Within-Group Residuals:

<table>
<thead>
<tr>
<th>Min</th>
<th>Q1</th>
<th>Med</th>
<th>Q3</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5.54687251</td>
<td>-0.46097804</td>
<td>-0.04964521</td>
<td>0.42959160</td>
<td>9.92662442</td>
</tr>
</tbody>
</table>

Number of Observations: 4785

Number of Groups: 59

The residual standard error decreased to 3.76 from the previous value of 3.97. The fixed coefficients changed also quite a lot (Figure 3.2). Figure 3.2 was plotted using

```r
> kert <- coef(lmm3)
> plots <- rownames(kert)
> plot(spati$id2, spati$id1, col = "red",
+     xlab = "Past diameter growth", ylab = "Coming diameter growth")
>
> for (i in 1:length(plots)) {
+     thisplot <- spati[spati$kanro == plots[i],]
+     xapu <- seq(min(thisplot$id2), max(thisplot$id2), length = 10)
+     lines(xapu, kert[i,1] + kert[i,2] * xapu)
+ }
> abline(fixef(lmm3), lwd=2)
```
Figure 3.2: Plot of the future diameter growth on past growth and the prediction of model lmm3.
3.1. SINGLE LEVEL OF GROUPING

3.1.4 Checking of the assumptions of the mixed model

The model form

With the mixed model, checking that the assumed model form fits to the data is as important as with the simpler models, too. However, checking the assumptions may not be as simple as it was with fixed-effects models. The plot of residuals on all the predictors is a good starting point to see if the model has a good shape. However, there may also other ways of checking the fit. an example of such a case is presented in the following example. For correcting the model shape, the same rules apply as in the case of linear models (see section 2.4)

Example 3.4. In the previous example, we modeled future diameter growth using the past diameter growth. To evaluate the selected model shape, we plot the residuals ($e_{ki}$) of the $1mm3$ against fitted values (those including the plot effects). In addition, we use mywhiskers to help in detecting trends. The plot is shown on the upper left corner of Figure 3.3. Even though a very slight indication of a trend can be detected, we do not change our model shape.

Other assumptions to be checked are, if the assumption of random parameters are
realistic. For example, in a variance component model, only the level of the assumed model is assumed to vary among groups, whereas the model shape is assumed to be the same. Plots should be used to see if this is a realistic assumption.

**Example 3.5.** In the previous example, we first plotted the plot-specific models and then the models lmm2 and lmm3, of which the latter one allows plot-specific slope coefficients. A test between these models will be shown later.

**Assumptions on the residual variance**

A plot of standardized (conditional) residuals on the predicted value should express a constant variance with no trends. A variance function should be used to homogenize the residuals, or alternatively, a transformation could be made to the response. The normality of residuals should be checked, for example, by using q-q plots, as the ML and REML methods are based on normality. Slight discrepancy from normality is usually allowed, as no very good methods for normalizing the data have been presented. Transformations to the response could be used to get the data better met the normality. However, this results into the bias problem, which was discussed in section 2.4.3.

**Example 3.6.** The residuals of model lmm3 have been plotted on the fitted values in the lower left plot of figure 3.3. A clear increasing trend was observed in the variance. A new model with the power-type variance function

$$\text{var}(\varepsilon_{ki}) = \sigma^2 |id2_{ki}|^{2\rho}$$

was fitted using the following code. Furthermore, the standardized residuals were plotted on fitted values, and a whiskers plot was added on the residuals (upper right plot of Figure 3.3). The applied variance function seems to fit very well.

```r
> lmm4<-lme(id1~id2, + random=~id2|kanro, + data=spati, + weights=varPower(0.5,~id2))
> plot(fitted(lmm4),resid(lmm4,type="pearson"),col="red")
> mywhiskers(spati$id2,resid(lmm4,type="pearson"),add=TRUE,se=FALSE)

Furthermore, we make a normal q-q plot to see whether the residuals are normal.

```r
> qqnorm(resid(lmm4,type="pearson"))
> qqline(resid(lmm4,type="pearson"))
```

The qq-plot indicates too heavy tails for the normality to hold. However, we do not have tools to fix this problem, so we just remember this when we report this model.

In a repeated measurements data, lagged residuals could be used to analyze the possible temporal autocorrelation. In spatial data, covariagrams could be used to analyze the possible spatial autocorrelation.
Assumptions on the random effects

Also the random effects are assumed to have constant variance and normal distribution. These assumptions should be checked by plotting the BLUPs of random effects against the predicted value. The normality of random effects should be checked by using q-q-plots. If several random effects are included at a certain level, the linearity of the correlation can be checked by plotting the predicted random effects against each other. The nonlinearity is usually linked with discrepancy from (multi)normality; this results from that with multinormal distribution, all marginal and joint distributions are normal and all correlations are linear.

Example 3.7. The normality of random effects and linearity of the correlation was tested using the following R-code. The resulting plots show that the normality is not met, and the correlation seems to be nonlinear. All these observations decrease the quality of our analysis, even though we do not have simple tricks to fix these problems. They indicate that the tests we will do are not as reliable as we would like them to be.

```r
qqnorm(ranef(lmm4)[,1])
qqline(ranef(lmm4)[,1])
qqnorm(ranef(lmm4)[,2])
qqline(ranef(lmm4)[,2])
plot(ranef(lmm4)[,1], ranef(lmm4)[,2])
```
3.1.5 Tests on the model

Tests on the fixed part can be based on the same principles as in the case of linear model. However, one should keep in mind that LR test for two models with different fixed effects should be carried out with models estimated by ML, not with models based on REML. Assumptions on the random part can be tested with either of the models.

**Example 3.8.** The previously performed analysis showed a need for variance function. Thus, we first estimate a model without random effects, using a variance function only, and a model with random constant.

```r
> gls1 <- gls(id1~id2, + data=spati, + weights=varPower(0.5, "id2"))
> lmm1 <- lme(id1~id2, + data=spati, + random="1|kanro, + weights=varPower(0.5, "id2")
> anova(gls1,lmm1)

Model df AIC BIC logLik Test L.Ratio p-value
---
gls1 1 4 27950.12 27976.01 -13971.06
lmm1 2 5 25284.03 25316.40 -12637.02 1 vs 2 2668.090 <.0001

We see that it would have been very unlikely to get a data like the one we got, if the null hypothesis on the common constant for all stands would be true. Thus, we reject the null hypothesis and proceed with lmm1.

The next step is to test whether the observed variation in the slopes among stands could be a result of random variation. For that purpose, we fit a model with random slope and constant and test it against model lmm1.

```r
e > lmm2 <- lme(id1~id2, + data=spati, + random="1{id2}|kanro, + weights=varPower(0.5, "id2")
> anova(lmm1,lmm2)

Model df AIC BIC logLik Test L.Ratio p-value
---
lmm1 1 5 25284.03 25316.40 -12637.02
lmm2 2 7 24571.53 24616.84 -12278.77 1 vs 2 716.4991 <.0001

The resulting p-value is very low, indicating that we reject the null hypothesis on the common slope for all the plots.

Finally, we test whether the need for variance function by fitting a model without variance function and comparing it with the one with variance function.

```r
> lmm3 <- lme(id1~id2, + data=spati, + random="1{id2}|kanro
> anova(lmm2,lmm3)

Model df AIC BIC logLik Test L.Ratio p-value
---
lmm2 1 7 24571.53 24616.84 -12278.77
lmm3 2 6 26651.57 26690.41 -13319.79 1 vs 2 2082.038 <.0001

The variance function clearly reduced the log likelihood, and we keep the model lmm2.

Finally, we perform some tests to see if additional predictors could significantly improve our model. For testing models with different fixed effects, the models need to be fitted using Maximum likelihood instead of the restricted maximum likelihood. We re-estimate model lmm2 using ML into object lmm4. Furthermore, we fit a model with ln(id2) as an additional predictor.
3.1. SINGLE LEVEL OF GROUPING

```r
> lmm4<-lme(id1˜id2,
+ data=spati,
+ random=˜1+id2|kanro,
+ weights=varPower(0.5,˜id2),
+ method="ML")
> lmm5<-lme(id1˜id2+log(id2),
+ data=spati,
+ random=˜1+id2|kanro,
+ weights=varPower(0.5,˜id2),
+ method="ML")
> anova(lmm4,lmm5)
Model df AIC BIC logLik Test L.Ratio p-value
lmm4 1 7 24566.86 24612.18 -12276.43
lmm5 2 8 24568.66 24620.45 -12276.33 1 vs 2 0.2037587 0.6517
```

An alternative and better strategy for testing fixed-effects is the use of Wald test. That is because the above mentioned LR-tests may be anti-conservative (i.e., they would reject the null hypothesis too easily). However, in this case we did not reject the null-hypotheses so we can trust our results. The wald tests can be carried out in R by calling anova with only one model as the argument.

Comparison of the model shows no statistical significant improvement. So we keep model lmm4. In addition, we test whether including tree age as an additional predictor would help.

```r
> lmm6<-lme(id1˜id2+t,
+ data=spati,
+ random=˜1+id2|kanro,
+ weights=varPower(0.5,˜id2),
+ method="ML")
> anova(lmm4,lmm6)
Model df AIC BIC logLik Test L.Ratio p-value
lmm4 1 7 24566.86 24612.18 -12276.43
lmm6 2 8 24566.18 24617.97 -12275.09 1 vs 2 2.682168 0.1015
```

Again, we do not find any improvement. Thus, we report model lmm2 as our final model.

```r
> summary(lmm2)
Linear mixed-effects model fit by REML
Data: spati
AIC BIC logLik
24571.53 24616.84 -12278.77
Random effects:
Formula: ˜1 + id2 | kanro
Structure: General positive-definite, Log-Cholesky parametrization
StdDev Corr
(Intercept) 5.4077955 (Intr)
id2 0.1906105 -0.565
Residual 0.4970949
Variance function:
Structure: Power of variance covariate
Formula: ˜id2
Parameter estimates:
  power 0.6926241
Fixed effects: id1 ˜ id2
  Value Std.Error DF t-value p-value
(Intercept) 1.6358858 0.7196234 4725 2.273253 0.0231
id2 0.8256847 0.0264244 4725 31.247050 0.0000
Correlation:
  (Intr)
id2 -0.576
```

Standardized Within-Group Residuals:
3.2 Multiple levels of grouping

There may be also other random effects, due to a more complicated structure of the data. For example, individual \( i = 1, \ldots, n_k \) may have been observed at different points in time \( t = 1, \ldots, T \). A candidate model for such data would be

\[
y_{it} = \mu + a_i + b_{it} + e_{it}
\]

where \( a_i \sim NID(0, \sigma_a^2) \), \( b_{it} \sim NID(0, \sigma_b^2) \), and \( e_{it} \sim NID(0, \sigma^2) \). In this model, we assume that the observations of different individuals have been taken in the same point in time, thus assuming each observation to include the same effect for time \( t \). In this model, the group and time effects are crossed.

If the measurements are taken at different points in time, it would be better to assume them to be nested, i.e., measurements taken for individual \( i \) at different points are somehow similar, but the \( t \)th measurement occasion of individual \( i \) is not similar to the \( t \)th measurement occasion for other individuals. The nested model would be

\[
y_{kit} = \mu + a_k + b_{ki} + e_{kiti}
\]

where \( a_k \sim NID(0, \sigma_a^2) \), \( b_{ki} \sim NID(0, \sigma_b^2) \), and \( e_{kiti} \sim NID(0, \sigma^2) \). Now points in time are nested within individuals, which are further nested within groups. This kind of structure arises, for example, in an analysis of permanent plot data, where the plots have been established at different years and remeasured with fixed (e.g., 5 year) intervals. It would also be possible to define a model that is a combination of these two models, i.e., has both crossed and nested time effects. In most cases, the hierarchical populations lead to mixed models having one or more nested effects.

3.2.1 A nested multilevel model

In the nested model, we have lower-level groups (\( i \)) within upper level groups (\( k \)). For example, if branch-level biomass data have been collected from several (randomly sampled) branches of a tree, and from several (randomly sampled) trees per plot, the level \( k \) would indicate plot level, level \( ki \) would indicate the tree level for plot \( k \), and finally the residual and possibly some predictors would be specified for the branch level. Another example would be a dataset having several trees per sample plot and several sample plots within a stand.
A nested multilevel model with random constant is defined as

\[ y_{kit} = \beta_0 + \beta_1 x_{kit} + \ldots + \beta_p x_{pkit} + a_k + b_{ki} + e_{kit}, \]

where \( a_k \sim \text{NID}(0, \sigma_a^2) \), \( b_{ki} \sim \text{NID}(0, \sigma_b^2) \), and \( e_{kit} \sim \text{NID}(0, \sigma_e^2) \). The random effects are independent, i.e. \( \text{cov}(a_k, a_{k'}) = 0 \), \( \text{cov}(b_{ki}, b_{ki'}) = 0 \). In this model, we assume that the relationship between the response and predictors is similar among groups up to a group-specific constant.

A straightforward result from having a nested grouping structure is that we can have predictions at different levels:

\[
\begin{align*}
\tilde{y} &= \tilde{\beta}_0 + \tilde{\beta}_1 x_{kit} + \ldots + \tilde{\beta}_p x_{pkit} \\
\tilde{y}_k &= \tilde{\beta}_0 + \tilde{\beta}_1 x_{kit} + \ldots + \tilde{\beta}_p x_{pkit} + \tilde{a}_k \\
\tilde{y}_{ki} &= \tilde{\beta}_0 + \tilde{\beta}_1 x_{kit} + \ldots + \tilde{\beta}_p x_{pkit} + \tilde{a}_k + \tilde{b}_{ki}
\end{align*}
\]

Correspondingly, the models have residuals at different levels:

\[
\begin{align*}
\tilde{e} &= y_{kit} - \tilde{y} \\
\tilde{e}_k &= y_{kit} - \tilde{y}_k \\
\tilde{e}_{ki} &= y_{kit} - \tilde{y}_{ki}
\end{align*}
\]

However, often the term residual is used for the highest-level residual \( \tilde{e}_{ki} \), which is also the default in R function \texttt{resid}. Other residuals are obtained by using argument \texttt{level}, which is given values form 0 (population level) to the total number of levels in the model.

The random effects for the multilevel models are predicted using the BLUP’s as they are done in the case of single-level models, too.

It is possible to have also other random effects than just the constant. For example, we may have random constant and random slope of a linear model for two levels of grouping. The model would then be

\[ y_{kit} = \beta_0_{kit} + \beta_1 x_{kit} + e_{kit} \]

This model could be fitted separately for each group, leading to estimation of fixed parameters \( \beta_0_{kit} \) and \( \beta_1_{kit} \) separately for each highest level group.

Assuming that the groups of the data are a random sample of a population of groups and that we are more interested on the distribution of the group-specific random effects, we could assume a mixed-effects model having random effects of the constant and slope at both levels of grouping:

\[ y_{kit} = \alpha + a_k + a_{ki} + (\beta + b_k + b_{ki}) x_{kit} + e_{kit}, \]
which can be reorganized to

\[ y_{kit} = \alpha + \beta x_{kit} + a_k + b_k x_{kit} + a_{ki} + b_{ki} x_{kit} + e_{kit}, \]

where the first two terms are the fixed part, and the rest of the equation is the random part. To be more specific, \( \alpha \) and \( \beta \) are the population-level means of the constant and slope; \( a_k \) and \( b_k \) the random effects for the first level of grouping, with

\[ D_k = \begin{pmatrix} a_k \\ b_k \end{pmatrix} \sim \begin{pmatrix} \text{var}(a_k) & \text{cov}(a_k, b_k) \\ \text{cov}(a_k, b_k) & \text{var}(b_k) \end{pmatrix}; \]

and \( a_{ki} \) and \( b_{ki} \) are the random effects for the second level of grouping with

\[ D_{ki} = \begin{pmatrix} a_{ki} \\ b_{ki} \end{pmatrix} \sim \begin{pmatrix} \text{var}(a_{ki}) & \text{cov}(a_{ki}, b_{ki}) \\ \text{cov}(a_{ki}, b_{ki}) & \text{var}(b_{ki}) \end{pmatrix} \]

with any positive definite \( D_k \) and \( D_{ki} \).

The random effects at different levels, as well as the random errors are multinormally distributed and uncorrelated across the levels.

**Example 3.9.** Figure 3.5 shows repeated measurements of tree increment core basal areas from felled trees over a 17 year follow-up period as a function of tree age. The age trend in the data seems quite negligible, but the thinning intensity shows clear effect on the basal are growth. We also see that clearly the level of growth remains quite much as the same for a given tree, but it varies between trees. In addition, there might be a common plot effect for all trees of a given plot.

Therefore, the full model for the data would be

\[ RBA_{kit} = \beta_0 + \beta_1 A G E_{kit} + \beta_2 \text{THIN}_{1k} + \beta_3 \text{THIN}_{2k} + \beta_4 \text{THIN}_{3k} + b_0_k + b_1_k A G E_{kit} + b_0_{ki} + b_1_{ki} A G E_{kit} + e_{kit}, \]

where \( \beta_0, \ldots, \beta_4 \) are fixed parameters; \( A G E_{kit} \) is the age of tree \( i \) of plot \( k \) at the year \( t \); \( b_0_k \) and \( b_1_k \) are the random effects for plot \( k \) and \( b_0_{ki} \) and \( b_1_{ki} \) are the random effects for tree \( i \) of plot \( k \).

We start by fitting the model by first including only random constants.

```r
thinning82 <- read.table("d:/laurim/biometria/datasets/thinning82.txt")
afterthin <- thinning82[thinning82$Year>1990,]
library(lmfor)
pdf("d:/laurim/biometria/lecturenotes/figpat1.pdf")
linesplot(afterthin$CA, afterthin$RBA, group=100*afterthin$Plot+afterthin$Tree, lty=afterthin$SDClass,cex=0, col.lin=afterthin$Plot,
```
Figure 3.5: The basal area of annual rings as a function of tree age. The lines connect observations of a tree, and the line type indicates the intensity of a thinning that happened 4 years before the first observation (solid=Control, dashed=light, dotted=Moderate and dotdash=Heavy). The same color is used for trees of the same plot.
The model is fitted using

```r
mod1 <- lme(RBA ~ CA + SDClass,
random = ~1|Plot/Tree,
data = afterthin)
```

The model summary is shown below

```r
> summary(mod1)
Linear mixed-effects model fit by REML
Data: afterthin
AIC BIC logLik
15873.36 15914.81 -7928.679
Random effects:
Formula: ˜1 | Plot
(Intercept) StdDev: 0.03573927
Formula: ˜1 | Tree %in% Plot
(Intercept) Residual
StdDev: 158.7277 88.01713
Fixed effects: RBA ~ CA + SDClass
Value Std.Error DF t-value p-value
(Intercept) 165.46715 37.01533 1230 4.470233 0.0000
CA 3.07727 0.55972 1230 5.497857 0.0000
SDClass2 27.02180 44.91552 6 0.601614 0.5694
SDClass3 143.61295 48.39648 6 2.967426 0.0250
SDClass4 219.04863 49.16893 6 4.455022 0.0043
Correlation:
 (Intr) CA SDCls2 SDCls3
CA -0.528
SDClass2 -0.591 -0.006
SDClass3 -0.548 -0.007 0.455
SDClass4 -0.548 0.009 0.448 0.415
Standardized Within-Group Residuals:
 Min Q1 Median Q3 Max
-3.52787460 -0.51008912 -0.06629868 0.47785446 5.67638955
Number of Observations: 1319
Number of Groups:
 Plot Tree %in% Plot
 10 88
```

The model summary shows that the plot-level standard deviation in the level of growth is 0.0357 mm$^2$, whereas the between-tree variation is much higher, 158.7 mm$^2$ and the residual is 88.02 mm$^2$. This indicates that maybe the plot-level is not needed in the model. Figure 3.6 shows the model residuals on the predicted value. Because they show increasing variance with respect to the predicted value, we update the model to allow increasing variance of the power form $\text{var}(e_{kti}) = \sigma^2 y_{kti}^{2+\delta}$. The new model is fitted and the residual plot plotted.

```r
mod2 <- lme(RBA ~ CA + SDClass,
random = ~1|Plot/Tree,
data = afterthin,
weights = varPower())
```

```r
pdf("d:/laurim/biometria/lecturenotes/figpat3.pdf")
plot(predict(mod2, type="p"), resid(mod2, type="p"))
mywhiskers(predict(mod2, type="p"), resid(mod2, type="p"), add=TRUE, se=FALSE)
dev.off()
```
Figure 3.6: The residual plot of mod1.
The standardized residuals of the updated model do not show heteroscedasticity, and we continue the model development from the updated model (Figure 3.7).

The summary of the first model showed quite small variation between plots when compared to the variation between trees. Therefore, we drop the level “plot” from the model and perform a LR-test to test whether the new restricted model is sufficient.

> mod3<-lme(RBA~CA+SDClass, 
+       random=~1|Tree, 
+       data=afterthin, 
+       weights=varPower())
>
> anova(mod3,mod2)

<table>
<thead>
<tr>
<th>Model</th>
<th>df</th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>Test</th>
<th>L.Ratio</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mod3</td>
<td>1</td>
<td>16234.45</td>
<td>16275.90</td>
<td>-8109.227</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mod2</td>
<td>2</td>
<td>15407.76</td>
<td>15454.39</td>
<td>-7694.880</td>
<td>1 vs 2</td>
<td>828.6942</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

The result is that even though the variation between plots was low, the model including both levels is significantly better fitting than the model with tree level only.
3.2. MULTIPLE LEVELS OF GROUPING

The next question is, whether also the coefficient of age should be random. We fit a model with random coefficients for age at both levels of grouping. After that, we will perform a likelihood ratio test to test whether the new, more general model fits the data significantly better than the current model.

```r
> mod4 <- lme(RBA ~ CA + SDClass,
+ random =˜1+CA|Plot/Tree,
+ data = afterthin,
+ weights = varPower())
>
> anova(mod2, mod4)
```

<table>
<thead>
<tr>
<th>Model</th>
<th>df</th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>Test</th>
<th>L.Ratio</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mod2</td>
<td>1</td>
<td>15407.76</td>
<td>15454.39</td>
<td>-7694.880</td>
<td>1 vs 2</td>
<td>329.7556</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>mod4</td>
<td>2</td>
<td>15086.00</td>
<td>15153.35</td>
<td>-7530.002</td>
<td>1 vs 2</td>
<td>329.7556</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

The test result indicates that the coefficient of age should also be random at both levels. However, it is still possible that only tree level for AGE is enough. To test this, we fit model with random constant for Plot level and random constant and slope for tree level.

```r
> mod5 <- lme(RBA ~ CA + SDClass,
+ random = list(˜1|Plot, ˜1+CA|Tree),
+ data = afterthin,
+ weights = varPower())
>
> anova(mod5, mod4)
```

<table>
<thead>
<tr>
<th>Model</th>
<th>df</th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>Test</th>
<th>L.Ratio</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mod5</td>
<td>1</td>
<td>15082.00</td>
<td>15138.99</td>
<td>-7530.002</td>
<td>1 vs 2</td>
<td>1.429653e-05</td>
<td>1</td>
</tr>
<tr>
<td>mod4</td>
<td>2</td>
<td>15086.00</td>
<td>15153.35</td>
<td>-7530.002</td>
<td>1 vs 2</td>
<td>1.429653e-05</td>
<td>1</td>
</tr>
</tbody>
</table>

The test indicates that the new model is sufficient but significantly better than mod3. Finally, we repeat the test on the need of the plot level in the model using the last model.

```r
> mod6 <- lme(RBA ~ CA + SDClass,
+ random = list(˜1+CA|Tree),
+ data = afterthin,
+ weights = varPower())
>
> anova(mod6, mod5)
```

<table>
<thead>
<tr>
<th>Model</th>
<th>df</th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>Test</th>
<th>L.Ratio</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mod5</td>
<td>1</td>
<td>15082.00</td>
<td>15138.99</td>
<td>-7530.002</td>
<td>1 vs 2</td>
<td>1041.525 &lt;.0001</td>
<td></td>
</tr>
<tr>
<td>mod6</td>
<td>2</td>
<td>16121.53</td>
<td>16173.34</td>
<td>-8050.764</td>
<td>1 vs 2</td>
<td>1041.525 &lt;.0001</td>
<td></td>
</tr>
</tbody>
</table>

The result still supports the inclusion of the plot level in the model. Thus, the final model for the data is the following

Linear mixed-effects model fit by REML
Data: afterthin
AIC 15082 15138.99  -7530.002

Random effects:
  Formula: ˜1 | Plot
  (Intercept) StdDev: 0.09313557

  Formula: ˜1 + CA | Tree %in% Plot
CHAPTER 3. LINEAR MIXED MODELS

Structure: General positive-definite, Log-Cholesky parametrization

StdDev Corr
(Intercept) 311.7853646 (Intr)
CA 9.8802767 -0.879
Residual 0.5079901

Variance function:
Structure: Power of variance covariate
Formula: ~ fitted(.)
Parameter estimates:
power
0.8347299

Fixed effects: RBA ~ CA + SDClass

| Value Std.Error DF t-value p-value |
|---|---|---|---|---|
| (Intercept) | 142.55562 | 43.99178 | 1230 | 3.240506 | 0.0012 |
| CA | 3.10147 | 1.14055 | 1230 | 2.719267 | 0.0066 |
| SDClass2 | 39.14948 | 42.37174 | 6 | 0.923953 | 0.3912 |
| SDClass3 | 173.63293 | 46.05687 | 6 | 3.769232 | 0.0093 |
| SDClass4 | 238.5913 | 46.7037 | 6 | 5.099577 | 0.0022 |

Correlation:
(Intr) CA SDClass2 SDClass3
CA -0.739
SDClass2 -0.469 -0.003
SDClass3 -0.421 -0.017 0.450
SDClass4 -0.414 -0.017 0.443 0.408

Standardized Within-Group Residuals:

<table>
<thead>
<tr>
<th>Min</th>
<th>Q1</th>
<th>Med</th>
<th>Q3</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.5010518</td>
<td>-0.6066507</td>
<td>-0.0400077</td>
<td>0.5833341</td>
<td>3.7400193</td>
</tr>
</tbody>
</table>

Number of Observations: 1319
Number of Groups:
Plot Tree % in Plot
10 88

The variation between plots is very low compared to that of residual and tree effects. The fixed effects show slight increase in the ring basal area as a function of tree age. The differences between stand density classes are high and logical. The power-parameter of the variance function is positive, indicating that the residual variance increases as a function of predicted value.

Figure 3.8 shows the predictions of the model at the different levels. The tree level predictions are shown by the straight moderately thick lines. The thick lines show both population and plot level predictions: differences between these predictions are so small that no difference can be noticed visually.

Finally, figure 3.9 shows the residual plots, qq-plots of residuals and random effects, and the correlation plot of tree-level random effects. These plots do not show any severe violation of the model assumptions and we regard the fit satisfactory.
Figure 3.8: The tree, plot and population level predictions from model mod5. The population and plot-level predictions are almost the same and cannot be visually separated.
Figure 3.9: Some diagnostic graphs of mod5.
3.2. MULTIPLE LEVELS OF GROUPING

```r
add=TRUE)
linesplot(afterthin$CA,
predict(mod5, level=1),
group=100*afterthin$Plot+afterthin$Tree,
cex=0,
col.lin=afterthin$Plot,
lwd=3,
add=TRUE)
linesplot(afterthin$CA,
predict(mod5, level=0),
group=100*afterthin$Plot+afterthin$Tree,
cex=0,
lwd=4,
add=TRUE)
dev.off()
```

```r
dev.off()
```

```r
pdf("d:/laurim/biometria/lecturenotes/figpat5.pdf",width=7,height=9)
par(mfrow=c(3,2))
qqnorm(resid(mod5,type="p"),main="residual")
qqline(resid(mod5,type="p"))
qqnorm(ranef(mod5,level=1)[,1],main="Plot effects")
qqline(ranef(mod5,level=1)[,1])
qqnorm(ranef(mod5,level=2)[,1],main="Tree level constant")
qqline(ranef(mod5,level=2)[,1])
qqnorm(ranef(mod5,level=2)[,2],main="Tree level slope")
qqline(ranef(mod5,level=2)[,2])
plot(ranef(mod5,level=2)[,1],ranef(mod5,level=2)[,2])
abline(lm(ranef(mod5,level=2)[,2]~ranef(mod5,level=2)[,1]))
dev.off()
```

### 3.2.2 Crossed grouping structures

In the model with crossed random effects, multiple levels of grouping are not groups within groups. Instead, we could group the data in several different ways so that the membership of a given group in one grouping do not restrict the possibilities for memberships according to the other grouping variable.

A typical example of a crossed grouping structure arises from that the repeated measurements of a hierarchical dataset may have a common year effect. For example, increment of different trees at a given calendar year may be similar because of the climate of that year.

Assuming two crossed levels of grouping, and a random constant for each group, the model might be

\[
y_{kt} = \beta_0 + \beta_1 x_{1kt} + \ldots, + \beta_p x_{pkt} + a_k + b_t + e_{kt},
\]

where \(a_k \sim NID(0, \sigma^2_a)\), \(b_t \sim NID(0, \sigma^2_b)\), and \(e_{kt} \sim NID(0, \sigma^2)\). The difference to the nested model is very small in the equation: only the index of the random effect of the second level has changed from \(b_{kt}\) to \(b_t\). In addition, the index \(i\) is missing because the combination of \(k\) and \(t\) specifies each observations in an unique way; that is, we only have one observation of each combination of \(k\) and \(t\). However, this difference makes a significant difference to the model. It indicates that the different value of \(k\) for
Figure 3.10: The observed Ring Basal Area after four thinning treatments (RBA) on calendar year.

A given observation does not indicate that the observations would belong to a different group with respect to $t$.

However, we could also have several observations for each group, in which case the additional index $i$ would be needed.

Handling of crossed random effect structures in the nlme package of R is not very convenient. However, package lme4 (by the same authors as nlme) is able to handle such structures. The package has not yet been officially published even though a beta version of it has been available at CRAN for some years. Also a manual for it and lecture notes by the authors can be found at http://lme4.r-forge.r-project.org/.

**Example 3.10.** Figure 3.10 shows the dataset of Figure 3.5. However, it has now been plotted on the calendar year instead of tree age. The graph indicates that the relative level of growth has been similar for different trees at a given calendar year. For
3.2. MULTIPLE LEVELS OF GROUPING

example, we can see that most trees have had quite low growth in year 1993, and high
growth in years 1994 and 2000. Still we have clear tree effects, which suggest a model
with crossed random effects.

We use package lme4 to fit an updated model with crossed random effects. First
we need to define all grouping varaibles as factors. lme4 does not make difference
between nested and crossed random effects. Therefore, nesting needs to be taken into
account in the factors. Therefore, to have trees nested within plots, the interaction with
plot an tree need to be specified as a new grouping variable (Tree2 below).

```r
afterthin$SDClass<-as.factor(afterthin$SDClass)
afterthin$Tree<-as.factor(afterthin$Tree)
afterthin$Plot<-as.factor(afterthin$Plot)
afterthin$Tree2<-with(afterthin, factor(Plot:Tree))
```

The model was fitted using lmer and the summary was printed. The sumamry
shows that the variation between plots is 0, that between trees is 162.7 mm$^2$, between
years 30.9 mm$^2$ and the residual is 83.4 mm$^2$. In the earlier example using nlme, these
standard deviations were 0.037 for the plot level, 158.7 for the tree level, and 88.0 for
residual. For example, the decrease ion teh residual standard deviation is explained by
the year effect as we can see by computing $\sqrt{83.4^2 + 29.9^2} = 88.6$, which is roughly
equal to the residual standard error of our nlme model.

```r
> mod1c<-lmer(RBA~SDClass+CA+(1|Plot)+(1|Tree2)+(1|Year),data=afterthin)
> summary(mod1c)
```

```
Linear mixed model fit by REML
Formula: RBA ~ SDClass + CA + (1 | Plot) + (1 | Tree2) + (1 | Year)
Data: afterthin
AIC  BIC logLik deviance REMLdev
15774 15821 -7878 15795 15756
Random effects:
Groups   Name        Variance Std.Dev.
Tree2    (Intercept) 24900.42  157.799
Year     (Intercept)   893.59   29.893
Plot     (Intercept)     0.00     0.000
Residual             6956.26   83.404
Number of obs: 1319, groups: Tree2, 88; Year, 15; Plot, 10
Fixed effects:
(Intercept)  118.496   70.774   1.674
SDClass2    26.328    44.621   0.590
SDClass3   142.749    48.083   2.969
SDClass4   220.029    48.855   4.504
CA           4.424    1.807   2.448

Correlation of Fixed Effects:
 (Intr) SDClass2 SDClass3 SDClass4
SDClass2 -0.290
SDClass3 -0.265  0.455
SDClass4 -0.308  0.447  0.415
CA       -0.891 -0.021 -0.024  0.029
```

Clearly, the plot level is not needed in the model. This can be confirmed by fitting
a model without that level of grouping and comparing the two models using anova.
```
> mod2c<-lmer(RBA~SDClass+CA+(1|Tree2)+(1|Year),data=afterthin)
> anova(mod1c,mod2c)
Data: afterthin
Models:
  mod2c: RBA ~ SDClass + CA + (1 | Tree2) + (1 | Year)
  mod1c: RBA ~ SDClass + CA + (1 | Plot) + (1 | Tree2) + (1 | Year)
   Df  AIC   BIC logLik Chisq Chi Df Pr(>Chisq)
mod2c  8 15811 15853 -7897.6
mod1c  9 15813 15860 -7897.6  0  1  1

We can also test whether the year level is significant. The test shows that it is.
```
```
> anova(mod3c,mod2c)
Data: afterthin
Models:
  mod3c: RBA ~ SDClass + CA + (1 | Tree2)
  mod2c: RBA ~ SDClass + CA + (1 | Tree2) + (1 | Year)
   Df  AIC   BIC logLik Chisq Chi Df Pr(>Chisq)
mod3c  7 15908 15944 -7946.8
mod2c  8 15811 15853 -7897.6  98.414 1 < 2.2e-16 ***

The previous model included a random effect also for the age. Including it and
performing a test on it shows that it is needed also now that the year level is included
in the model.
```
```
> mod4c<-lmer(RBA~SDClass+CA+(1+CA|Tree2)+(1|Year),
+ data=afterthin)
> anova(mod2c,mod4c)
Data: afterthin
Models:
  mod2c: RBA ~ SDClass + CA + (1 | Tree2) + (1 | Year)
  mod4c: RBA ~ SDClass + CA + (1 + CA | Tree2) + (1 | Year)
   Df  AIC   BIC logLik Chisq Chi Df Pr(>Chisq)
mod2c  8 15811 15853 -7897.6
mod4c 10 15496 15548 -7738.2  318.81 2 < 2.2e-16 ***

> mod4c
Linear mixed model fit by REML
Formula: RBA ~ SDClass + CA + (1 + CA | Tree2) + (1 | Year)
Data: afterthin
AIC BIC logLik deviance REMLdev
15457 15509 -7719 15476 15437
Random effects:
Groups   Name       Variance Std.Dev. Corr
Tree2    (Intercept) 121024.74  347.886
          CA        114.31  10.692 -0.899
Year     (Intercept) 928.45   30.471
          (Intercept) 928.45  30.471
Residual            4632.07  68.059
Number of obs: 1319, groups: Tree2, 88; Year, 15
```
```
Fixed effects:
             Estimate Std. Error t value
(Intercept)  96.679     78.022  1.239
SDClass2    37.102     43.591  0.851
SDClass3   -176.277    46.981 -3.752
SDClass4   -236.796    47.635 -4.971
CA           4.517      2.140  2.111
```

From the previous analysis using nlme, we already know that the model residuals
are heteroscedastic. Unfortunately, variance functions have not (yet) been implemented
in nlme4 package. However, the functions allows specifying a weighting variable to
the model. Thus, we do one-step IGLS fitting as follows:

1. Fit the model without weighting.
3.2. **MULTIPLE LEVELS OF GROUPING**

![Graph showing plots of standardized residuals](image)

**Figure 3.11:** The plots of standardized residuals without using the weighting (left) and with using it (right).

2. Save the predictions \( \tilde{y} \) from the model into the dataset.

3. Re-estimate the model by using weights \( 1/\tilde{y} \).

We could further continue the iteration by saving the predictions from the 3rd step and re-estimating the model by using these as the weight.

The following code implements the weighted estimation.

```r
> afterthin$F<-fitted(mod4c)
> mod5c<-lmer(RBA~SDClass+CA+(1|Tree2)+(1|Year),
  + data=afterthin,
  + weights=1/afterthin$F)

par(mfcol=c(1,2))
plot(fitted(mod4c),resid(mod4c,type="p"),cex=0.1)
mywhiskers(fitted(mod4c),resid(mod4c,type="p"),add=TRUE,se=FALSE)
plot(fitted(mod5c),resid(mod5c,type="p"),cex=0.1)
mywhiskers(fitted(mod5c),resid(mod5c,type="p"),add=TRUE,se=FALSE)
```

**Figure 3.11** However, we see that the estimated variance components are very high, most probably they should be re-scaled by something that is computed from the used weights. In addition, the \( t \) values of the fixed effects seem to depend on the scale of the weights. Therefore, we wait for an update to lme4, and before getting it we use model mod4c as our final model.

```r
> summary(mod5c)
Linear mixed model fit by REML
Formula: RBA ~ SDClass + CA + (1 | Tree2) + (1 | Year)
Data: afterthin
AIC  BIC logLik deviance REMLdev
7913 7954  -3948 7935 7897
Random effects:
Groups     Name     Variance Std.Dev. Std. Dev.
Tree2     (Intercept) 5790187.9 2406.281
Year      (Intercept) 74214.4  272.423
Residual            4102.9   64.054
Number of obs: 1319, groups: Tree2, 88; Year, 15

Fixed effects:
             Estimate Std. Error t value
(Intercept) 229.14     767.90   0.298
```

The predictions from a crossed-effects model is not as easy to compute as are the predictions from a nested models. Especially, we can make predictions at different levels, but the levels cannot be ordered as higher or upper levels. Therefore, lmer does not have a predict function available. Instead, the predictions need to be computed manually at the desired levels using the predicted random effects. The random effects and fixed coefficients can be extracted from a fitted model using functions fixef and ranef.

However, these functions have a conflict with the same methods of package nlme. Therefore, to use them you should not have package nlme loaded in your workspace when you use the functions ranef and fixef with a model fitted using lmer. If you have, these methods just give an error.

Function ‘fitted’ provides predictions (or fitted values) for the modelling data. Those values include the contribution of fixed effects and all random effects.

**Example 3.11.** The model mod2c of the previous example includes predicted random effects at the tree and year levels

```r
> ranef(mod2c)

$Tree2
   (Intercept) CA
1:79 -160.926304  0.22042915
1:111 -721.135270  26.23913067
1:181  90.603299  3.36681665
 .
.
.
10:142  41.003853 -6.81271623
10:196 170.818423 -2.73683651
10:237 109.873075 -2.12049948

$Year
   (Intercept)
1991  20.423269
1992 -17.583520
1993 -26.401921
1994  39.689407
1995  5.386164
1996 16.285901
1997 -9.780424
1998 -11.292646
1999 -18.297703
2000  51.016680
2001  35.207387
2002 12.405756
2003 -49.867211
2004 -20.455931
2005  26.935207
```

We notice, for example, that the have grown 51 mm higher in year 2000 than in an average. On the other hand, years 2003 has had very slow growth. These growth trends could most probably be explained by the weather of the year in question.
3.2 MULTIPLE LEVELS OF GROUPING

These random effects allow prediction at the following four levels:

- **Population level (fixed part only)**
- **Tree level (fixed part plus tree effect)**
- **Year level (Fixed part plus year effect)**
- **Year and tree level (Fixed part + Tree and year effects)**

The following code computes all those predictions. Figure 3.12 shows the predictions at different levels, plotted over the original data.

```r
# Predictions at population level
# formulate the model matrix
X<-cbind(1,afterthin\$SDClass==2,afterthin\$SDClass==3,afterthin\$SDClass==4,afterthin\$CA)
# Compute the fixed part predictions
predPop<-as.numeric(X%*%fixef(mod4c))
# Extract the random effects to object re
re<-ranef(mod4c)

# Predictions at the tree level
predTree<-predPop
trees<-rownames(re\$Tree2)
# Go through all the trees
for (i in 1:length(trees)) {
  # Form the model matrix of the random part
  # We need this because we have both random constant and random slope
  thisZ<-X[afterthin\$Tree2==trees[i],c(1,5)]
  # compute and add the Tree-related random part
  predTree[afterthin\$Tree2==trees[i]]<-predTree[afterthin\$Tree2==trees[i]]+thisZ%*%t(re\$Tree[i,])
}

# Predictions at the Year level
predYear<-predPop
years<-rownames(re\$Year)
# Go through all the trees
for (i in 1:length(years)) {
  # add the year-related random effect
  predYear[afterthin\$Year==years[i]]<-predYear[afterthin\$Year==years[i]]+re\$Year[i,]
}

# We make a trick to compute the tree-year level predictions
# An alternative would be fitted(mod4c)
# or two similar loops as above, wirst one adding tree effects and the
# second one the year effects
predAll<-predYear+predTree-predPop

pdf("d:/laurim/biometria/lecturenotes/figpat8.pdf")
par(mfcol=c(2,2))
linesplot(afterthin\$Year, afterthin\$RBA, group=afterthin\$Tree2, lty=as.numeric(afterthin\$SDClass),cex=0, col.lin=afterthin\$Plot, xlab="Tree age", ylab="Ring basal area, mm^2", main="Population level")
linesplot(afterthin\$Year, predPop, group=afterthin\$Tree2, lty=as.numeric(afterthin\$SDClass),cex=0, col.lin=afterthin\$Plot, lwd=1,add=TRUE)
```
Example 3.12. This example shows an analysis where linear mixed-effects models were used to extract thinning effects. The data are the same as in the previous examples, but it includes also observation before the thinning, which occurred after the growth period of year 1986. The idea is to first take a data with control plots for all years and thinned plots only for the years before thinning. Then a mixed-effects model with crossed year and tree level random effects is fitted to that data. Predictions from that model, including both tree and year level random effects, are then computed also for the thinned plots after thinning. The effect of thinning on the growth is then estimated as the difference between the observed basal area growth and the predicted growth.

The model is fitted for logarithmic basal area growths, and the age trend is modeled using a restricted cubic spline with three knots (Harrell 2001).

Read the data, compute basal area growth and the predictors.

```
patti<-read.table("d:/laurim/patti/pattidata.txt",header=TRUE,sep="\t")
patti$Tree<-100*patti$Tree+patti$Plot
```
Figure 3.12: The predicted values of ring basal area at different levels of hierarchy for model mod4c. The observations are also shown on the background.
# Use only data after calendar year 1982
patti<-patti[patti$Year>1982,]
# Compute Ring basal area
patti$RBA<-pi*((patti$Diameter/2)^2-(patti$PrevDiam/2)^2)
# Transformation for natural spline regression
# Eq 2.25 in Harrells book, p. 20
# t is a vector of knots, j the component to be calculated
natural.spline.comp<-function(X,t,j) {
  k<-length(t)
  pmax(0,X-t[j])^3-pmax(0,X-t[k-1])^3*(t[k]-t[j])/(t[k]-t[k-1])+pmax(0,X-t[k])^3*(t[k-1]-t[j])/(t[k]-t[k-1])
}
# Use 3 knots to model the age trend using a regression spline
# Compute the required spline component
t<--quantile(patti$CA,probs=c(0.1,0.5,0.9))
patti$e1<-natural.spline.comp(X=patti$CA,t=t,j=1)

Next, we formulate the model fitting dataset including only observed growths from years before thinning and from the control plot.

# Remove the observations of thinned plots after thinning
patti2<-patti[patti$SDClass==1|patti$Year<1987,]

Fit the model into dataset patti2. Details on model fitting are not shown, just the final model.
mallie5<-lmer(log(RBA)~CA+e1+(1|Tree)+(1|Year),data=patti2)

Compute the logarithmic predictions for dataset patti, which includes also observations of thinned years. Back-transform the predictions to the original scale using a bias correction proposed by (Lappi et al. 2006). Finally, subtract the predictions from the observations to see the thinning effects and plot them. These effects will be modeled using nonlinear mixed-effects model in the next section.

# Compute predictions for tree and year level
trees<--as.numeric(as.character(rownames(ranef(mallie5)$Tree[1])))
for (i in 1:length(trees)) {
patti$treeb5[patti$Tree==trees[i]]<-ranef(mallie5)$Tree[i,]
}
years<--as.numeric(as.character(rownames(ranef(mallie5)$Year[1])))
for (i in 1:length(years)) {
patti$yearb5[patti$Year==years[i]]<-ranef(mallie5)$Year[i,]
}
patti$predb5<-cbind(1,patti$CA,patti$e1)%*%fixef(mallie5)+patti$yearb5+unlist(patti$treeb5)

# Back-transformation with the two-point bias correction
patti$trend<-(exp(patti$predb5+attributes(mallie5)$deviance[6])+
  exp(patti$predb5-attributes(mallie5)$deviance[6]))/2
# Compute the thinning effects to the dataset.
patti$ThEf<-patti$RBA-patti$trend

Next, we formulate the model fitting dataset including only observed growths from years before thinning and from the control plot.

# Remove the observations of thinned plots after thinning
patti2<-patti[patti$SDClass==1|patti$Year<1987,]

Fit the model into dataset patti2. Details on model fitting are not shown, just the final model.
mallie5<-lmer(log(RBA)~CA+e1+(1|Tree)+(1|Year),data=patti2)

Compute the logarithmic predictions for dataset patti, which includes also observations of thinned years. Back-transform the predictions to the original scale using a bias correction proposed by (Lappi et al. 2006). Finally, subtract the predictions from the observations to see the thinning effects and plot them. These effects will be modeled using nonlinear mixed-effects model in the next section.

# Compute predictions for tree and year level
trees<--as.numeric(as.character(rownames(ranef(mallie5)$Tree[1])))
for (i in 1:length(trees)) {
patti$treeb5[patti$Tree==trees[i]]<-ranef(mallie5)$Tree[i,]
}
years<--as.numeric(as.character(rownames(ranef(mallie5)$Year[1])))
for (i in 1:length(years)) {
patti$yearb5[patti$Year==years[i]]<-ranef(mallie5)$Year[i,]
}
patti$predb5<-cbind(1,patti$CA,patti$e1)%*%fixef(mallie5)+patti$yearb5+unlist(patti$treeb5)

# Back-transformation with the two-point bias correction
patti$trend<-(exp(patti$predb5+attributes(mallie5)$deviance[6])+
  exp(patti$predb5-attributes(mallie5)$deviance[6]))/2
# Compute the thinning effects to the dataset.
patti$ThEf<-patti$RBA-patti$trend

Next, we formulate the model fitting dataset including only observed growths from years before thinning and from the control plot.

# Remove the observations of thinned plots after thinning
patti2<-patti[patti$SDClass==1|patti$Year<1987,]

Fit the model into dataset patti2. Details on model fitting are not shown, just the final model.
mallie5<-lmer(log(RBA)~CA+e1+(1|Tree)+(1|Year),data=patti2)

Compute the logarithmic predictions for dataset patti, which includes also observations of thinned years. Back-transform the predictions to the original scale using a bias correction proposed by (Lappi et al. 2006). Finally, subtract the predictions from the observations to see the thinning effects and plot them. These effects will be modeled using nonlinear mixed-effects model in the next section.
3.3 Matrix formulation

3.3.1 Single level of grouping

Mixed-effects model (3.5) can be stated in a matrix form as follows

\[ y = Xb + Zc + e. \]  (3.9)

where \( Xb \) is the fixed part and \( Zc + e \) the random part. The fixed part is similar to that of the fixed-effects model.

For the random part, \( Z \) is the design matrix of the random part and \( c \) is the vector of random parameters. The length of \( c \) is \( k \times q \), where \( k \) is the number of groups, and \( q \) the number of group-specific random parameters. Correspondingly, matrix \( Z \) is an \( n \times x \times p \) matrix. Thus, the number of columns in \( Z \) and the length of \( c \) depends on the number of groups in the data. Matrix \( Z \) is usually organized so that all parameters of the first group are included first, then those of the second group. This organization leads to a block-diagonal structure of \( \text{var}(e) \) and \( Z \), which eases the computations. The structure is probably best described by the following example.

**Example 3.13.** Assume model \( y_{ki} = \alpha + \beta x_{ki} + a_k + b_k x_{ki} + e_{ki}, \) where \( i = 1, \ldots, n_k \) and \( k = 1, \ldots, K \). The model can be written as

\[ y = Xb + Zc + e \]
Figure 3.13: The upper graph shows the dataset of ring basal areas for the whole period. Thin dashed lines are used for observations of thinned plots after the thinning, solid, modelrately thick lines show the observations from thinned plots before thinning and control plots for the whole period. The thick lines show the average trend, based on a lowess smoother, separately for each thinning treatment. The colors indicate the thinning reatment as follows: Black=control, red=light, green=moderate and blue=heavy thinning. The dashed lines in the lower graph shows the extracted thinning effects and the thick solid lines the corresponding lowess smoothers by treatments.
by defining

\[
\begin{bmatrix}
  y_{11} \\
y_{12} \\
  \vdots \\
y_{1n_1} \\
y_{21} \\
y_{22} \\
  \vdots \\
y_{2n_2} \\
y_{K1} \\
y_{K2} \\
  \vdots \\
y_{Kn_K}
\end{bmatrix},
\begin{bmatrix}
  1 & x_{11} \\
  1 & x_{12} \\
  \vdots \\
  1 & x_{1n_1} \\
  1 & x_{21} \\
  1 & x_{22} \\
  \vdots \\
  1 & x_{2n_2} \\
  1 & x_{K1} \\
  1 & x_{K2} \\
  \vdots \\
  1 & x_{Kn_K}
\end{bmatrix},
\begin{bmatrix}
  e_{11} \\
e_{12} \\
  \vdots \\
e_{1n_1} \\
e_{21} \\
e_{22} \\
  \vdots \\
e_{2n_2} \\
e_{K1} \\
e_{K2} \\
  \vdots \\
e_{Kn_K}
\end{bmatrix},
\begin{bmatrix}
  \alpha \\
  \beta
\end{bmatrix},
\begin{bmatrix}
  e_1 \\
  e_2 \\
  \vdots \\
e_n \\
e_2 \\
  \vdots \\
e_K \\
  \vdots \\
e_K
\end{bmatrix},
\begin{bmatrix}
  1 & x_{11} & 0 & 0 & \cdots & 0 & 0 \\
  1 & x_{12} & 0 & 0 & \cdots & 0 & 0 \\
  \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
  1 & x_{1n_1} & 0 & 0 & \cdots & 0 & 0 \\
  0 & 0 & 1 & x_{21} & \cdots & 0 & 0 \\
  0 & 0 & 1 & x_{22} & \cdots & 0 & 0 \\
  \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
  0 & 0 & 1 & x_{2n_2} & \cdots & 0 & 0 \\
  \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
  0 & 0 & 0 & 0 & \cdots & 1 & x_{K1} \\
  0 & 0 & 0 & 0 & \cdots & 1 & x_{K2} \\
  \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
  0 & 0 & 0 & 0 & \cdots & 1 & x_{Kn_K}
\end{bmatrix},
\begin{bmatrix}
  a_1 \\
  b_1 \\
  a_2 \\
  b_2 \\
  \vdots \\
  a_K \\
  b_K
\end{bmatrix}.
\]

The above example showed the matrices for the whole data. Matrices \(Z\) and \(D\) consist of blocks, each of which presents a single group. The model for each group can also be written in matrix a form, as shown in the following example. The model for a single group is needed, for example, in the prediction of random effects for a new group.

**Example 3.14.** In example 3.13, the model for a single group is written in matrix form as

\[
y_k = X_k b + Z_k c_k + e_k
\]

where

\[
y_k = \begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{kn_k} \end{bmatrix},
X_k = \begin{bmatrix} 1 & x_{k1} \\ 1 & x_{k2} \\ \vdots \\ 1 & x_{kn_k} \end{bmatrix},
b = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.
\]
Note that because all parameters were random in our model, matrices $X_k$ and $Z_k$ are the same. However, often only some of the parameters are random. A very common case is that only constant is assumed to be random, and other parameters are fixed. In such case, the model of example 3.13 would be written as

$$y_{ki} = \alpha + \beta x_{ki} + a_k + e_{ki}.$$  

and $Z_k$ and $c_k$ in the model for group $k$ would be just

$$Z_k = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, c_k = \begin{bmatrix} a_k \end{bmatrix}$$

Thus, to write a mixed model in a matrix form, the essential task is to be able to write the model for a single group. The matrices and vectors for whole data are then obtained as follows

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_K \end{bmatrix}, Z = \begin{bmatrix} Z_1 & 0 & \cdots & 0 \\ 0 & Z_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Z_K \end{bmatrix}, c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \end{bmatrix}, e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_K \end{bmatrix}$$

(3.10)

The vector of fixed parameters, $b$, is the same for whole data and a single group.

In the model for a single group,

$$y_k = X_k b + Z_k c_k + e_k,$$  

(3.11)

e_k and $e_k$ are random vectors. Of the vector of random effects, it is assumed that $e_k \sim N_q(0, D_k)$, where $D$ is a positive definite matrix. With model (3.3) such an assumption was specified in (3.4). Of $e_k$, the very general assumption is that $e_k \sim N_{n_k}(0, R_k)$, where $R_k$ parameterizes the assumed trends in the conditional residual variance and dependencies in the conditional residuals within a group. The usual (starting) hypothesis is that $R_k = \sigma^2 I$, which implies an assumption on uncorrelated conditional residuals with a constant variance. Relaxing this assumption is possible, for example assumptions on heteroscedastic variance and different autocorrelation structures are possible.

The assumptions about the residual error and random effects lead to that

$$y_k \sim N_{n_k}(X_k b, Z_k D Z_k' + R)$$
and

\[ \text{cov}(c_k, y'_k) = DZ'_k \]

These variance-covariance matrices can be easily derived by writing the assumed expressions for \( u_k \) into the equations of variance and covariance, and then applying the rules of chapter 1.

### 3.3.2 Multiple levels of grouping

Assume that we have two nested levels of grouping, the first (outer) level of grouping being indexed by \( k = 1, \ldots, K \), and the innermost grouping being indexed by \( l = 1, \ldots, n_k \). The linear mixed-effects model for group \( l \) within group \( k \) is written as

\[ y_{kl} = X_{kl}b + Z_{k,l}c'_k + Z_{kl}c_{kl} + e_{kl} \]

where \( c'_k \) includes the first (outer) level random effects for group \( k \), and \( c_{kl} \) the second (innermost) level random effects for level \( l \) within \( k \). Matrices \( Z_{k,l} \) and \( Z_{kl} \) are the corresponding design matrices.

The model for group \( l \) within group \( k \) can be written as ...

The model for the whole group \( k \) is

\[ y_k = X_kb + Z_k c_k + e_k \]

where

\[ y_k = \begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{kn_k} \end{bmatrix}, \quad X_k = \begin{bmatrix} X_{k1} \\ X_{k2} \\ \vdots \\ X_{kn_k} \end{bmatrix}, \quad c_k = \begin{bmatrix} c'_k \\ c_{k1} \\ c_{k2} \\ \vdots \\ c_{kn_k} \end{bmatrix} \]

and the vector of fixed parameters is the same as in the previous model, \( b \). The model for the whole data can be written as described in equation (3.10)

**Example 3.15.** Model

\[ h_{kti} = \alpha + \beta d_{kti} + a_k + b_k d_{kti} + a_{kt} + b_{kt} d_{kti} + e_{kti}, \]

for time point \( t \) within stand \( k \) can be written as

\[ h_{kt} = X_{kt}b + Z_{kt}c'_k + Z_{kt}c_{kt} + e_{kt} \]
by defining

\[
 h_{kt} = \begin{bmatrix} h_{kt1} \\ h_{kt2} \\ \vdots \\ h_{ktn} \end{bmatrix}, \quad X_{kt} = Z_{k,t} = Z_{kt} = \begin{bmatrix} 1 & d_{kt1} \\ 1 & d_{kt2} \\ \vdots & \vdots \\ 1 & d_{ktn} \end{bmatrix},
\]

\[
 c_{k}' = \begin{bmatrix} a_k \\ b_k \end{bmatrix}, \quad c_{kt} = \begin{bmatrix} a_{kt} \\ b_{kt} \end{bmatrix}, \quad e_{kt} = \begin{bmatrix} e_{kt1} \\ e_{kt2} \\ \vdots \\ e_{ktn} \end{bmatrix}
\]

The model for time \( t \) for plot \( k \) becomes...

The model for plot \( k \) becomes

\[
 h_k = X_k b + Z_k c_k + e_k
\]

where the design matrix for random part and the vector of random effects are

\[
 Z_k = \begin{bmatrix}
 1 & d_{k11} & 1 & d_{k11} & 0 & 0 & \cdots & 0 & 0 \\
 1 & d_{k12} & 1 & d_{k12} & 0 & 0 & \cdots & 0 & 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
 1 & d_{k1n_1} & 1 & d_{k1n_1} & 0 & 0 & \cdots & 0 & 0 \\
 1 & d_{k21} & 0 & 0 & 1 & d_{k21} & \cdots & 0 & 0 \\
 1 & d_{k22} & 0 & 0 & 1 & d_{k22} & \cdots & 0 & 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
 1 & d_{k2n_2} & 0 & 0 & 1 & d_{k2n_2} & \cdots & 0 & 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
 1 & d_{kn_1} & 0 & 0 & 0 & 0 & \cdots & 1 & d_{kn_1} \\
 1 & d_{kn_2} & 0 & 0 & 0 & 0 & \cdots & 1 & d_{kn_2} \\
 1 & d_{kn_n} & 0 & 0 & 0 & 0 & \cdots & 1 & d_{kn_n} \\
\end{bmatrix}, \quad b_k = \begin{bmatrix} a_k \\ b_k \\ a_{k1} \\ b_{k1} \\ a_{k2} \\ b_{k2} \\ \vdots \\ a_{kn} \\ b_{kn} \end{bmatrix}
\]

### 3.3.3 Crossed grouping structures

With the crossed grouping structure, all groupings have a similar structure as shown in example 3.14. However, pooling the group-specific matrices into the matrix of whole data is not as simple. especially, it does not anymore lead to diagonal or partially diagonal matrices.

### 3.3.4 The general matrix formulation

In all the above cases, the mixed-effects model for the whole data can be written in form

\[
 y = Xb + Zc + e'
\]  \hspace{1cm} (3.12)
where $X b$ is the fixed part and $Z c + e'$ is the random part. By defining that $e = Z c + e'$, we see that the mixed-effects model is a special case of the linear model (2.5). In the mixed-effects model, the residual of the linear model is partitioned into several terms. Some of these terms are common for some observations. For example, the group effects are common to all observations of a specific group. These shared parts of the residual cause correlation among the residuals of the linear model, $e$. This correlation is parameterized into the matrix $V$ through the variance-covariance matrix of the random effects, $D$. The variance-covariance matrix of the residual of the linear model can be computed as

$$
V = \text{var}(e) = \text{var}(Z c + e') = Z D Z' + R
$$

Thus, model (3.12) is such a special case of model (2.5), where the variance covariance-matrix of residuals is specified as

$$
\text{var}(e) = V = Z D Z' + R.
$$

### 3.4 Estimation

#### 3.4.1 Analysis of variance

The earliest estimation methods of a mixed-effects model are based on an ANOVA-approach. In the anova approach, the group-effects are first included into the model as dummy variables, and then the estimates for the variance components were based on the residual and regression sums of squares, which are divided with the number of observations and corrected for the degrees of freedom used. The ANOVA approach leads to unique, unbiased estimators for datasets with equal number of observations in each group and no missing data. However, problems arise with unbalanced data and missing observations, and no agreement exists in which estimators should be used in such cases. Thus, currently the most often applied and most widely accepted methods are those based on maximum likelihood and restricted (or residual) maximum likelihood.

#### 3.4.2 Maximum Likelihood

The method of maximum likelihood is a straightforward application of what was presented in section 2.7. One should be aware that it requires the assumption on normality already in the estimation stage, as does the REML method, too. In the most common case, an unrestricted positive definite structure is given to the variance-covariance
matrix for random effects at each level of grouping, and matrix $R$ is defined as the
diagonal matrix $\sigma^2 I$. The ML yields estimates for the variance-covariance parameters
for the random parameters at each level of grouping, for the residual error variance,
and for vector $b$. Furthermore, the dependence of residuals or heteroscedastic vari-
ance can be modeled by giving a more sophisticated structure for $R$. An important
result for the ML-method is, that whatever structure $V$ has, the ML-estimates of $b$ are
the GLS estimates, where the required variance-covariance matrix is replaced with the
ML-estimate. A widely known problem with the ML-estimates is that it is downward
biased. The extent of bias can be computed only for some special cases.

3.4.3 Restricted maximum likelihood

In restricted maximum likelihood, a specific linear combination of the original data is
modeled using maximum likelihood. The assumed model is specified as

$$K'y \sim N(0, K'VV'K),$$

where $K$ is a $n \times n - p$ matrix having full column rank. Essentially, the new vector
of responses, $K'y$, has only $n - p$ elements instead of the $n$ elements of original
$y$. However, it does include all the information included in the original residuals.
Furthermore, the fixed part is just 0, not including $b$ at all. Because the number of
observations in $K'y$ has been deduced to $n - p$, using ML to fit the new model into the
transformed data leads to such estimates of the random part that take into account the
degrees of freedom that are used for estimating the fixed part. The REML estimators
can thus be derived from ML-estimators by making replacements

$$y = K'yX = K'XV = K'VK.$$

It is important to notice that REML can be used only for estimating the parameters
specifying matrix $V$, because matrix $K$ has been selected in such a way that the effect
of the fixed part has been removed from the data. After estimation of the parameters of
the fixed part using REML, the estimation of the fixed parameters, $b$, is carried out by
applying GLS using the estimated matrix $\hat{V}$.

There are also an important distinction when different models are tested against each
other, which results from that the fixed part is filtered out from the REML-likelihood.
Namely, the LR-tests based on REML likelihood should be used only for models that
differ from each other in their random part. When models that differ in teh fixed part
are being tested, the models should be fitted using the standard ML. The final model
can then be estimated using REML, after appropriate structure has been found for the
fixed part.
3.5. PREDICTION OF RANDOM EFFECTS

The REML approach has become popular in mixed-effect-modeling for the following reasons (McCulloch and Searle 2001):

1. REML is sensible for balanced data, where REML solutions are the GLS and ANOVA estimators, which are minimal variance unbiased estimators under normality. However it is not known whether these properties apply to unbalanced data.

2. REML takes into account the degrees of freedom used in estimating the fixed effects $b$. This is important when the length of $b$ is large in relation to the number of observations. Presumably this feature occurs for unbalanced data, too.

3. REML estimates of the parameters of $V$ are invariant to the estimates of $b$.

4. REML estimates may not be as sensitive to outliers as ML estimates.

3.5 Prediction of random effects

Model estimation yields the estimates for the fixed parameters, those for the variances and covariances at different levels of grouping, and for the variance of the residual. However, estimation of the LMM does not yield estimates for the group effects, $c_k$ for groups $k = 1, \ldots, K$. These effects can, however, be predicted using the Best Linear Predictor, which was presented for a general case in section 1.6 and is taken back below. If 

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \sim \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} V_1 \\ V_{12} \\ V_2 \end{pmatrix},$$

then

$$BLP(h_1) = \hat{h}_1 = \mu_1 + V_{12}V_2^{-1}(h_2 - \mu_2)$$

with a prediction variance of

$$\text{var}(\hat{h}_1 - h_1) = V_1 - V_{12}V_2^{-1}V_{12}'.$$

Consider the model (3.11) for a single group $k$. We want to predict the random effects $c_k$ using the observations $y_k$. Thus, $h_1 = c_k$ and $h_2 = y_k$. The required variances and covariances were given before

$$V_1 = D$$

$$V_2 = Z_kDZ_k' + R$$

$$V_{12} = DZ_k'$$
The BLUP of random effects, and their prediction error variance become
\[
\hat{c}_k = DZ_k^\prime (Z_kDZ_k^\prime + R)^{-1} (y_k - X_kb)
\]
\[
\text{var}(\hat{c}_k - c_k) = D - DZ_k^\prime (Z_kDZ_k^\prime + R)^{-1} Z_kD
\]
Henderson mixed model equations give equivalent equations
\[
\hat{c}_k = (Z_kR^{-1}Z_k + D^{-1})^{-1} Z^\prime R^{-1} (y_k - X_kb)
\]
\[
\text{var}(\hat{c}_k - c_k) = (Z_kR^{-1}Z_k + D^{-1})^{-1}
\]
(Searle et al. 1992, Lappi 1991, see). The lower form requires inversion of a matrix of dimension \(p \times p\) in addition to the inversion of \(R\), which is of dimension \(n_k \times n_k\), but is usually diagonal. The upper requires the inversion of a non-diagonal matrix of dimension \(n \times n\). Thus, the lower form is computationally better if \(R\) is diagonal and the number of observations is high compared to the number of random parameters.

**Example 3.16.** In the previous example, we fitted a model for coming diameter growth on past growth. The model was
\[
id_{1k} = 1.6358858 + 0.8256847id_{2k} + a_k + c_k id_{2k} + e_{ki}.
\]
with \((\text{var})(a_k, c_k)\) as shown in the code below, and
\[
\text{var}(e_{ki}) = 0.4970949^2 |id_{2ki}|^2 + 0.6926241
\]
Assume that the coming (15.30,44.54) and past (17.78,40.18) diameter growth have been measured on two trees only from a stand, and we want to get a local model for that stand. The BLUP for stand effects has been computed in the code below. The localized model is shown in figure 3.14.

```r
> id1<-c(15.30,44.54)
> id2<-c(17.78,40.18)
> D<-getVarCov(lmm2)
> D
Random effects variance covariance matrix
     (Intercept) id2
(Intercept)   29.24400 -0.582380
id2          -0.58238  0.036332
Standard Deviations: 5.4078 0.19061
>
> Z<-cbind(1,id2)
> Z
          id2
[1,]  1 17.78
[2,]  1 40.18
> rho<-attributes(lmm2$apVar)$Pars[4]
> rho
varStruct.power
0.6926241
> sigma<-lmm2$sigma
> sigma
[1] 0.4970949
> R<-diag(sigma^2*id2^2*rho)
> R
     [,1]     [,2]
[1,] 13.31509 0.00000
```
3.5. PREDICTION OF RANDOM EFFECTS

Figure 3.14: The prediction of fixed part (thick line) and the localized curve (thin line) based on the two observations (circles).

```r
> b <- fixef(lmm2)
> b
  (Intercept) id2
1   1.6358858 0.8256847
> X <- Z
> X
id2
[1,] 1 17.78
[2,] 1 40.18
> ranef <- Z %*% t(Z) %*% solve(Z %*% t(Z) + R) %*% (id1 - X %*% b)
> ranef
[,1]
(Intercept) id2
1 -1.5320781 0.1248518
> fixrand <- b + ranef
> fixrand
[,1]
(Intercept) id2
1 0.1038078 0.9505366
plot(id2, id1, xlim=c(0, 80), ylim=c(0, 70))
abline(b, lwd=2)
abline(fixrand)
```
3.6 Extending the linear mixed model

The assumption of constant variance and independence of residuals can be relaxed also with the linear mixed-effects-model. In R, the same functions can be used for specifying the structure for the residual term than we used with the linear model and function glm.

3.7 An analysis of H-D curve

3.7.1 Selection of model form

Data on 1678 height-diameter observations for Scots Pine trees from 56 plots in North Carelia, Finland is analyzed. The data is cross-sectional, so the only cause for hierarchy is the organization of data into plots and trees within plots. The analysis has the following steps

1. Selecting appropriate model form from among four alternatives: Power equation, Meyer equation, Korf Curve, and Näslunds equation.
2. Fitting a first mixed-effects model, and checking the assumptions on model form and residual errors, and development of the model to better meet the assumptions.
3. Including additional stand-specific covariates to capture part of the between-stand variation.
4. Demonstrate the use of these two models for prediction when one sample tree has been measured for diameter and height.
5. Demonstrate the use of these two models for prediction when three sample tree has been measured for diameter and height.

The R-code of in file examples_ch3.R includes the analysis.

First, define the possible functions for the relationship, and find initial estimates for the parameter values.

```r
> spati<-read.table("d:\laurim\biometria\spati.txt",header=TRUE)
> # Include only pine sample trees
> spati<-spati[spati$pl==1,]
> spati$h2<-spati$h-1.3
> 
> # Select appropriate model form from among alternatives:
> # power H=aD^b
> HDpower<-function(d,a,b) {
+ a*d^b
+ }
> # Meyer: H=a(1-exp{-bD})
> HDmeyer<-function(d,a,b) {
+ a*(1-exp(-b*d))
+ }
```
The following code fits each of the four alternative models for each plot. It also plots the observations of each plot, and adds the fitted curves to the plots.

```r
> # A vector including plot numbers
> plots<-unique(spati$plot)
> # do these computations for each plot
> # and plot the models
> korf<-forfb<rep(NA,length(plots))
> # open a new window at every fourth iteration
> if (i%%4==0) {
+ windows()
+ par(mfcol=c(2,2))
+ # this plot will include observations only from i:th plot
+ thisplot<-spati[spati$plot==plots[i],]
+ # fit each of the models for each plot using nls
+ fmpower<-nls(h2~HDpower(d,a,b),
+ data=thisplot,
+ start=list(a=1.9,b=0.7))
+ ffmeyer<-nls(h2~HDmeyer(d,a,b),
+ data=thisplot,
+ start=list(a=30,b=0.04))
+ # common start values for each plot did not converge for each plot.
+ # That is why startng values are estimated separately for each plot
+ # by fitting a linearized form of the model.
+ # This could be automatized by defining a self-start function.
```
CHAPTER 3. LINEAR MIXED MODELS

```r
corfstart <- lm(log(h2) ~ 1/d, data = thisplot)
fmkorf <- nls(h2 ~ HDkorf(d, a, b), data = thisplot,
               start = list(a = exp(coef(korfstart)[1]), b = -coef(korfstart)[2]))
fnaslund <- nls(h2 ~ HDnaslund(d, a, b), data = thisplot,
                start = list(a = 1.2, b = 0.18))

# Make a new plot that shows the h-d data of this particular plot
# and add the fitted lines from each of the four fits into the plot.
plot(thisplot$d, thisplot$h2)
lines(d, HDpower(d, coef(fmpower)[1], coef(fmpower)[2]), lwd = 2)
lines(d, HDmeyer(d, coef(fmmeyer)[1], coef(fmmeyer)[2]), col = "red", lwd = 2)
lines(d, HDkorf(d, coef(fmkorf)[1], coef(fmkorf)[2]), col = "green", lwd = 2)
lines(d, HDnaslund(d, coef(fmnaslund)[1], coef(fmnaslund)[2]), col = "blue", lwd = 2)

# Save the residuals from each of these fits
# Note: the columns are automatically created in the first iteration.
spati$respow[spati$plot == plots[i]] <- resid(fmpower)
spati$resmey[spati$plot == plots[i]] <- resid(fmmeyer)
spati$reskorf[spati$plot == plots[i]] <- resid(fmkorf)
spati$resnas[spati$plot == plots[i]] <- resid(fmnaslund)

The following code computes means, standard deviations and standard errors of residuals for each model. Näsland’s model seems to be the best, followed by Korf, Meyer and Power functions.

```r
> rbind(sd = sd(spati[, c("respow", "resmey", "reskorf", "resnas")]),
      mean = mean(spati[, c("respow", "resmey", "reskorf", "resnas")]),
      se = sd(spati[, c("respow", "resmey", "reskorf", "resnas")]) / sqrt(dim(spati)[1]))
   respow resmey reskorf resnas
sd 1.09239050 0.97568785 0.97223176 0.97005669
mean -0.01628642 -0.01793159 0.00914928 -0.00378441
se 0.02666748 0.02381853 0.02373416 0.02368110

Next, plot the residuals against diameter and standardized diameter to see the fit of each of these models.

```r
# plot the residuals in the standard way
# against tree diameter
windows()
par(mfcol = c(2, 2))
plot(spati$d, spati$respow)
abline(0, 0)
plot(spati$d, spati$resmey)
abline(0, 0)
plot(spati$d, spati$reskorf)
abline(0, 0)
plot(spati$d, spati$resnas)
abline(0, 0)

# and against standardized d
windows()
par(mfcol = c(2, 2))
plot(spati$dstd, spati$respow)
abline(0, 0)
plot(spati$dstd, spati$resmey)
abline(0, 0)
plot(spati$dstd, spati$reskorf)
abline(0, 0)
plot(spati$dstd, spati$resnas)
abline(0, 0)

# Residuals against standardized diameter
```
3.7. AN ANALYSIS OF H-D CURVE

Figure 3.16: Data and fitted models for four example plots.
The plots are shown in figures 3.17 and 3.18. Residuals against diameter do not show any trend, whereas plotting against standardized diameter shows that Power and Meyer equations are not flexible enough for our purposes. They lead underestimation of height for medium sized trees of a stand, and overestimation of height for largest and smallest trees. No difference can be seen between Korf and Näslunds curves. Plots 3.19 and 3.20 show the corresponding plots from mywhiskers. I selected Korf curve, but Näslunds curve could have been as good or even a better alternative.

3.7.2 Fitting the linear mixed-effects model

The previously shown computations were carried out in the original scale using the nonlinear regression approach. However, many modeling tasks become much easier in the linear scale, even though it leads to biased predictions in nonlinear scale. However, we select a linearized model and then do the normality-based bias correction. The linearized Korf model with random effects for both $a$ and $b_2$ is obtained using logarithmic transformation as

$$\ln(h_{ki} - 1.3) = \alpha + a_k - (\beta + b_k) \frac{1}{d_i}$$

The constant of the linearized model, $\alpha + a_k$, is obtained by making the logarithmic transformation to the original scale parameter. The following code fits the linearized korf curve to each plot, and plots the obtained parameters. The plots are shown in figure 3.21. The distribution of the stand-specific parameters seems not to be normal, and the correlation shows a nonlinear trend. However, we do not have good tools to fix...
Figure 3.17: Residuals against diameter.
Figure 3.18: Residuals against standardized diameter.
3.7. AN ANALYSIS OF H-D CURVE

Figure 3.19: Residuals against diameter.
Figure 3.20: Residuals against standardized diameter.
these departures, and we will proceed with this model, even though we are aware that
the assumptions are not very well met.

```r
korfpar<-data.frame(a=rep(NA,length(plots)),b=rep(NA,length(plots)))
for (i in 1:length(plots)) {
  thisplot<-spati[spati$plot==plots[i],]
  lmkorf<-lm(log(h2)~I(1/d),data=thisplot)
  korfpar[i,]<-c(1,-1)*coef(lmkorf)
}
head(korfpar)
```

```
 a  b
1 3.107455 8.633308
2 2.981028 6.266065
3 2.753103 4.404059
4 2.964051 6.243494
5 3.051922 5.531834
6 3.226972 7.581148
```

```r
# Plot the plot-specific estimates
windows(4,7)
par(mfcol=c(3,2))
hist(korfpar$a)
qqnorm(korfpar$a,main="a")
hist(korfpar$b)
qqnorm(korfpar$b,main="b")
plot(korfpar)
```

Next, we will fit the mixed-effects model into the data.

```r
# This model can be fitted using function lme in package nlme (OLD)
library(nlme)
lmm1<-lme(log(h2)~I(1/d),random=˜1+I(1/d)|plot,data=spati)
lmm1
```

```
Linear mixed-effects model fit by REML
Data: spati
Log-restricted-likelihood: 639.0039
Fixed: log(h2) ~ I(1/d)
(Intercept) I(1/d)
2.807888 -6.279249
Random effects:
Formula: ˜1 + I(1/d) | plot
Structure: General positive-definite, Log-Cholesky parametrization
StdDev Corr
(Intercept) 0.4922689 (Intr)
I(1/d) 2.272998 -0.875
Residual 0.1472408
Number of Observations: 1678
Number of Groups: 56
```

```r
# Explore object lmm1
attributes(lmm1)
```

Compared to the residual variation, the between-stand variation is high, and the
need for random effects is obvious. We could also make a formal test by fitting a
restricted model and test if the full model is significantly better than the restricted
model.

Next, study if the assumptions of the model are met. We assumed
Figure 3.21: Plots of stand-specific parameter estimates.
3.7. AN ANALYSIS OF H-D CURVE

1. The dependence of log(H−1.3) on D can be described using korf curve

2. Residuals have constant variance

3. Residuals are normally distributed

4. Random effects are multinormally distributed

The validity of these assumptions are studied in the following code. The resulting plots are shown in figure 3.22. The plots show

```r
# Assumption 1: model form
windows(4,8)
par(mfcol=c(4,2))
plot(spati$d,resid(lmm1),col="red")
mywhiskers(spati$d,resid(lmm1),add=TRUE,lwd=2)
mywhiskers(spati$d,resid(lmm1),se=FALSE,add=TRUE)
abline(0,0)
# Model seems to fit fairly well.

# Residual variance has a decreasing trend:
plot(fitted(lmm1),resid(lmm1))
mywhiskers(fitted(lmm1),resid(lmm1),add=TRUE,se=FALSE)

# Normality of residuals
#qqnorm(resid(lmm2,type="pearson"))
#abline(0,1)
qqnorm(resid(lmm1,type="pearson"))
abline(0,1)
# The residuals have heavy tails!

# Normality of random effects
qqnorm(unlist(random.effects(lmm1)[1]))
qqnorm(unlist(random.effects(lmm1)[2]))
plot(unlist(random.effects(lmm1)[1]),unlist(random.effects(lmm1)[2]))
```

The following code makes a plot of the fixed part and plot-specific models. The plot is shown in Figure 3.23.

```r
> windows()
> plot(spati$d,log(spati$h2),col="red")
> for (i in 1:56) {
+ thisplot<-spati[spati$plot==plots[i],]
+ d<-seq(min(thisplot$d),max(thisplot$d),length=20)
+ lines(d,coef(lmm1)[i,1]+coef(lmm1)[i,2]/d)
+ }
> d<-seq(min(spati$d),max(spati$d),length=20)
> lines(d,fixef(lmm1)[1]+fixef(lmm1)[2]/d,lwd=3,col="blue")
```

We fit a new model with heteroscedastic variances. The plot looks better (Figure 3.24), and the model fits significantly better.

```r
> lmm1b<-lme(log(h2)~1/d,
+ random="1+1/d"|plot,
+ data=spati,
+ weights=varPower(-0.5,"d")
>
>
> # Residual variance looks better
> plot(fitted(lmm1b),resid(lmm1b,type="pearson"))
> mywhiskers(fitted(lmm1b),resid(lmm1b,type="pearson"),add=TRUE,se=FALSE)
>
> # The model also fits better
> anova(lmm1,lmm1b)
```

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
<th>df</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>lmm1b</td>
<td>1</td>
<td>6</td>
<td>-1266.008</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>7</td>
<td>1619.514</td>
</tr>
</tbody>
</table>

Model df AIC BIC logLik Test L.Ratio p-value
lmm1 1 6 -1266.008 -1233.463 639.0039
lmm1b 2 7 -1619.514 -1581.545 816.7572 1 vs 2 355.5065 <.0001
Figure 3.22: Diagnostic plots on lmm1.
Figure 3.23: Data (red), marginal (blue) and conditional (black) curves.

Figure 3.24: Residuals of model fmm1b.
CHAPTER 3. LINEAR MIXED MODELS

Assume we want to model the dependence in stand specific values of \(a\) and \(b\) by using some predictors. First take a look on how these parameters depend on stand characteristics. These plots are made using the following commands. The plots are shown in figures 3.25 and 3.26. We see that the random parameters clearly depend on stand characteristics. The dependence seems to be strongest for mean diameter \(D_g\). For this covariate, the relationship between \(a\) is nonlinear, but the relationship with \(b\) looks fairly linear. A logarithmic transformation in \(D_g\) seems to linearize the relationship fairly well (Figure 3.27)

```
plotdata<-cbind(unique(spati[,c("plot","X","Y","N","G","Dg","Tg")]),korfpar)
windows(4,7)
par(mfcol=c(3,2))
plot(plotdata$X,plotdata$a)
plot(plotdata$Y,plotdata$a)
plot(plotdata$N,plotdata$a)
plot(plotdata$G,plotdata$a)
plot(plotdata$Dg,plotdata$a)
plot(plotdata$Tg,plotdata$a)
windows(4,7)
par(mfcol=c(3,2))
plot(plotdata$X,plotdata$b)
plot(plotdata$Y,plotdata$b)
plot(plotdata$N,plotdata$b)
plot(plotdata$G,plotdata$b)
plot(plotdata$Dg,plotdata$b)
plot(plotdata$Tg,plotdata$b)
```

Thus, our new model would be

\[
\ln(h_{ki}) = \alpha_0 + \alpha_1 \ln(D_g) + \alpha_k + (\beta_0 + \beta D_g + b_k) \cdot (1/d_{ki}) + e_{ki}
\]

This model is fitted using the following code. Note that we refit also the restricted model using ML, because we are going to make a LR-test on the fixed effects. The new relaxed model is significantly better than the old one.

```
> lmm4<-lme(log(h)^I(1/d),
+ random="1+I(1/d)|plot,
+ data=spati,
+ weights=varPower(-0.5,`d`),
+ method="ML"
>
> > lmm4b<-lme(log(h)^I(1/d)+log(Dg)+I(1/d):Dg,
+ random="1+I(1/d)|plot,
+ data=spati,
+ weights=varPower(-0.5,`d`),
+ method="ML"
>
> > # Make the LR test
>anova(lmm4,lmm4b)
```

The next question is, whether including additional predictors would improve the model. Let us add stand age as an additional predictor and do the test. The additional predictor provides no improvement to the model. The explanation can be seen by plotting the stand effects of model lmm4b against stand-specific predictors (Figures

```
Figure 3.25: Parameter a against stand variables.
Figure 3.26: Parameter $b$ against stand variables.
3.7. AN ANALYSIS OF H-D CURVE

There seems not to be any unexplained trend in them, and the trends that were seen in plots 3.25 and 3.26 were caused by correlations among stand variables.

```r
> lmm4c<-lme(log(h2)~I(1/d)+log(Dg)+Tg+I(1/d):Dg, + random=~1+I(1/d)|plot, + data=spati, + weights=varPower(~0.5,"d"), + method="ML")
> anova(lmm4b,lmm4c)
```

Thus, we refit the final model using REML and save the two final models into objects fm1 and fm2. Furthermore, we plot the fixed parts of the model (Figure 3.31), as well as the modeled dependence of $a$ and $b$ on $Dg$ (Figure 3.30).

```r
# Refit lmm4b with REML
lmm4b<-lme(log(h2)~I(1/d)+log(Dg)+I(1/d):Dg, + random=~1+I(1/d)|plot, + data=spati, + weights=varPower(-0.5,"d"), + method="REML")

# Final models are lmm1b and lmm4b
fm1<-lmm1b
fm2<-lmm4b

# Plot the plot-specific estimates of $a$, and their estimates
windows(3,5)
par(mfcol=c(2,1))
plot(plotdata$Dg,plotdata$a)
```

Figure 3.27: Dependence of $\log(b)$ on $Dg$. 

3.28 and 3.29). There seems not to be any unexplained trend in them, and the trends that were seen in plots 3.25 and 3.26 were caused by correlations among stand variables.
Figure 3.28: Stand effects a of lmm4b against stand variables.
3.7. AN ANALYSIS OF H-D CURVE

Figure 3.29: Stand effects b of lmm4b against stand variables.
Figure 3.30: Modeled dependence of a and b on Dg.
3.7. AN ANALYSIS OF H-D CURVE

The same diagnostic plots were plotted for the model including Dg as a predictor (Figure 3.32) than for the model without Dg (Figure 3.22). The assumptions about normality seem to be better met when the nonlinear dependence of a on Dg has been modeled.
Figure 3.32: Diagnostic plots from model fm2.
3.7. AN ANALYSIS OF H-D CURVE

As a final plot of the model, Figure 3.33 shows conditional predictions (i.e., those with stand effects predicted for the modeling data) for both models. The conditional models are very close to each other. Further analysis could be done on which of the models is better.

Figure 3.33 was generated using code

```r
windows()
plot(spati$d, log(spati$h2), col="red")
for (i in 1:56) {
  thisplot<-spati[spati$plot==plots[i],]
  d<-seq(min(thisplot$d), max(thisplot$d), length=20)
  lines(d, cbind(1, 1/d, log(thisplot$d[1]), thisplot$d[1]/d) %*% t(coef(fm2)[i,]))
}
for (i in 1:56) {
  thisplot<-spati[spati$plot==plots[i],]
  d<-seq(min(thisplot$d), max(thisplot$d), length=20)
  lines(d, coef(fm1)[i,1]+coef(fm1)[i,2]/d, col="green", lty="dashed")
}
```

3.7.3 Model application

A very useful application of the linear mixed-effects model is using it for prediction. In such case, we can use either marginal prediction, i.e., the fixed part only. However, if we have measurements of tree height and diameter available from even one sample tree, we can make conditional, stand specific prediction by predicting the stand effects of our model for the stand in hand. In the following code, we make predictions

- for fm1 using one measured sample tree
• for fm2 using one measured sample tree
• for fm1 using three measured sample trees
• for fm1 using three measured sample trees

The predictions of \( \ln(h - 1.3) \) are a straightforward application of the best linear predictor. Also the conditional predictions plotted in Figure 3.33 utilize the same predictors, but they are carried out in R.

In mixed model, we estimated the fixed parameters and variance components. We are often interested also in the random effects. They are predicted using the BLP or BLUP. For the plots of the data, R has calculated them. They can be studied using random.effects() or ranef(). However, a useful situation is that we want to predict the random effects for a plot that was not included in the data. For example, we may have one tree height measured for one stand, and we want to utilize that information in predicting the HD-curve for that stand. First, we assume that from a stand with \( D_g = 10 \), one tree with \( d = 12 \) and \( h = 10 \) has been observed. We first specify the vectors and matrices for BLP, and then compute the BLP.

```r
> hobs<-10
> dobs<-12
> # To predict random effects of model 1, we use BLP with
> # h1=(a_k,b_k)
> # h2= observed ln(height-1.3)=ln(8.7)= 2.163323
> h2<-log(hobs-1.3)
> # the expectations are
> # mu1=c(0,0)
> mu1<-c(0,0)
> # mu2= the fixed part of the model for d=12
> mu2<-c((1,/dobs) %*% fixed.effects(fm1)
> # V1 = the variance-covariance matrix of random effects
> V1<-D<-getVarCov(fm1)
> # V2 = the variance-covariance matrix of heights, i.e., ZD’+R,
> # where D = V1
> # rho< attributes(fm1$apVar)$Pars[4]
> sigma<-fm1$sigma
> R<-sigma^2*dobs^(2*rho)
> V2<-V1*V1%*%t(V2)+R
> V12<-V1*V1%*%t(V2)
> h2
> [1] 2.163323
> mu1
> [1] 0 0
> mu2
> [,1]
> [1,] 2.292262
> V1
> Random effects variance covariance matrix
> (Intercept) I(1/d)
> (Intercept) 0.23954 -0.98661
> I(1/d) -0.98661 5.49210
> Standard Deviations: 0.48942 2.3435
> V2
> [,1]
> [1,] 0.126992
> V12
> [,1]
> (Intercept) 0.1573194
> I(1/d) -0.5289329
3.7. AN ANALYSIS OF H-D CURVE

Figure 3.34: Local and population-level curve for the example stand in logarithmic scale.

```R
> # The blup of random effects is
> h1<-bhat<V12%*%solve(V2)%*%(h2-mu2)
> h1

     [,1]
(Intercept) -0.1598475
1(1/d)   0.5374327

> fixef(fm1)+h1

     [,1]
(Intercept) 2.670914
1(1/d)   -5.924571
```

Thus, the intercept \(a\) for this particular stand is \(a + 0.157 = 2.97\) and the shape parameter is \(\beta + 0.529 = 5.925\). The localized, (or calibrated) conditional curve for this plot is shown in Figure 3.34.

```R
# The local curve almost passes through the observed height. Why?
> cbind(1,1/12)%*%(fixed.effects(fm)+bhat)

     [,1]
[1,] 2.1772

> h2

[1] 2.163323
```
Next, we want to make predictions in the original scale. A bias correction based on normality of logarithmic residuals is applied by adding half of the prediction error variance into the predictions before applying the exponential transformation. The following code makes the computations and produces the plot shown in figure 3.35.

```r
# Transformation to the original scale.
# Half of the prediction variance needs to be added before back-transformation.
# 1. The prediction based on fixed and random parts
# The prediction errors of random parameters are
# Transformation to the original scale.
# Half of the prediction variance needs to be added before back-transformation.
# 1. The prediction based on fixed and random parts
# The prediction errors of random parameters are
> varb<-V1-V12%*%solve(V2)%*%t(V12)
> varb
Random effects variance covariance matrix
 (Intercept) I(1/d)
 (Intercept) 0.044505 -0.33088
 I(1/d) -0.330880 3.28740
 Standard Deviations: 0.21096 1.8131
> V1
Random effects variance covariance matrix
 (Intercept) I(1/d)
 (Intercept) 0.23954 -0.98661
 I(1/d) -0.98661 5.49210
 Standard Deviations: 0.48942 2.3435
> Z0<-cbind(1,1/d)
> R0<-sigma^2*d^(2*rho)
> var.lnh2<-Z0%*%varb%*%t(Z0)+diag(R0)
> hpred.fr<-1.3+exp(pred.fr+0.5*diag(var.lnh2))
>
# 2. The prediction based on fixed part only
> var.lnh2<-Z0%*%D%*%t(Z0)+diag(R0)
> hpred.f<-1.3+exp(pred.f+0.5*diag(var.lnh2))
```

Figure 3.35: Local (solid) and population-level (dashed) curve for the example stand in original scale.
3.7. AN ANALYSIS OF H-D CURVE

The same computations were made using the model with Dg as a predictor. The resulting figure in original scale is shown in Figure 3.36. Even though there is a huge difference in the fixed parts of the model, the one sample tree moves the local curves fairly close to each other.

```r
# Let us make the corresponding predictions using fm2
Dg<-10
fixed.effects(fm2)

(Intercept) I(1/d) log(Dg) I(1/d):Dg
-0.1187020 -1.4058238 1.0464768 -0.2791521
mu2<-c(1,1/dobs,log(Dg),Dg/dobs)%*%fixed.effects(fm2)

# V1 = the variance-covariance matrix of random effects
V1<-VarCov(fm2)

# V2 = the variance-covariance matrix of heights, i.e., ZDZ'+R,
# where D = V1
Z<-cbind(1,1/dobs)
rho<-attributes(fm2$apVar)$Pars[4]
sigma<-fm2$sigma
R<-sigma^2*Z%*%Z'+R
V2<-Z%*%V1%*%t(Z)+R
V12<-V1%*%t(Z)

# The blup of random effects is
bhat<-V12%*%solve(V2)%*%(h2-mu2)

# plot the localized or calibrated H-D curve.
# Save the predictions to vectors
# pred.f - logarithmic prediction based on the fixed part only
# pred.fr - logarithmic prediction based on fixed and random parts
d<-seq(1,50)
plot(d,pred.f2<-cbind(1,1/d,log(Dg),Dg/d)%*%fixed.effects(fm2),type="l",col="green",lwd=2)
points(dobs,h2)
lines(d,pred.fr2<-cbind(1,1/d,log(Dg),Dg/d)%*%(fixed.effects(fm2)+c(bhat,0,0)),type="l",col="green",lwd=2)

# The local curve does not pass that close to the observed height. Why?
cbind(1,1/12,log(Dg),Dg/12)%*%(fixed.effects(fm2)+c(bhat,0,0))

[,1]
[1,] 2.093866
h2
[1] 2.163323

# Transformation to the original scale.
# Half of the prediction variance needs to be added before back-transformation.
# 1. The prediction based on fixed and random parts
# The prediction errors of random parameters are
varb<-V1-V12%*%solve(V2)%*%t(V12)

Random effects variance-covariance matrix
(Intercept) I(1/d)
(Intercept) 0.020079 -0.12404
I(1/d) -0.124040 1.44110
Standard Deviations: 0.1417 1.2005
V1

Random effects variance-covariance matrix
(Intercept) I(1/d)
(Intercept) 0.020079 -0.12404
I(1/d) -0.124040 1.44110
Standard Deviations: 0.1417 1.2005
V1

# 2. The prediction based on fixed part only
var.lnh2<-Z0%*%varb%*%t(Z0)+diag(R0)
hpred.fr2<-1.3+exp(pred.fr2+0.5*diag(var.lnh2))

> var.lnh2

[,1]
[1,] 2.093866
h2
[1] 2.163323

> # Transformation to the original scale.
> h2
> [1] 2.163323
> h2
> [1] 2.163323
> h2
> [1] 2.163323
> 
> # Transformation to the original scale.
> # Half of the prediction variance needs to be added before back-transformation.
> # 1. The prediction based on fixed and random parts
> # The prediction errors of random parameters are
> varb<-V1-V12%*%solve(V2)%*%t(V12)
> ```
CHAPTER 3. LINEAR MIXED MODELS

Figure 3.36: Local (solid) and population-level (dashed) curve for the example stand in original scale for fm1 (black) and fm2 (red).

```r
> plot(d,hpred.f,type="l",lwd=2,lty="dashed")
> lines(d,hpred.f2,lwd=2,lty="dashed",col="red")
> lines(d,hpred.fr,lwd=2,col="red")
> lines(d,hpred.fr2,lwd=2,col="red")
> points(dob,hobs)
```

Finally, we repeat the prediction using three sample trees. The commands are shown below. The plot of local curves from both models is shown in figure 3.37

```r
> # EXAMPLE 2 Three trees
> # Assume that from a stand with Dg=10, three trees with d=c(5,9,12), and h=10 have been observed.
> hobs<-c(6,8,10)
> dob<-c(5,9,12)
> # To predict random effects of model 1, we use BLUP with
> # h1=c(a_k,b_k)
> # h2=the observed ln(height-1.3)=ln(8.7)= 2.163323
> h2<-log(hobs-1.3)
> # the expectations are
> # mu1=c(0,0)
> # mu2=c(0,0)
> # The blup of random effects is
> bhat<-V12%*%solve(V2)%*%(h2-mu2)
```
3.7. AN ANALYSIS OF H-D CURVE

Figure 3.37: Local (solid) and population-level (dashed) curve for the example stand in original scale for fm1 (black) and fm2 (red).

```r
> # plot the localized or calibrated H-D curve.
> # Save the predictions to vectors
> # pred.f - logarithmic prediction based on the fixed part only
> # pred.fr - logarithmic prediction based on fixed and random parts
> d<-seq(1,50)
> plot(d,pred.f<-cbind(1,1/d)%*%fixed.effects(fm1),type="l",col="green",lwd=2)
> points(dobs,h2)
> lines(d,pred.fr<-cbind(1,1/d)%*%(fixed.effects(fm1)+bhat),type="l",col="green",lwd=2)
>
> # Transformation to the original scale.
> # Half of the prediction variance needs to be added before back-transformation.
> # 1. The prediction based on fixed and random parts
> varb<-V1-V12%*%solve(V2)%*%t(V12)
> varb
Random effects variance covariance matrix
     (Intercept) I(1/d)  
(Intercept) 0.033286 -0.22755
I(1/d) -0.227550 1.88180
Standard Deviations: 0.18244 1.3718
> V1
Random effects variance covariance matrix
     (Intercept) I(1/d)  
(Intercept) 0.23954 -0.98661
I(1/d) -0.98661 5.49210
Standard Deviations: 0.48942 2.3435
> Z0<-cbind(1,1/d)
> R0<-sigma^2*d^(2*rho)
> var.lnh2<-Z0%*%varb%*%t(Z0)+diag(R0)
> hpred.fr<-1.3+exp(pred.fr+0.5*diag(var.lnh2))
>
> # 2. The prediction based on fixed part only
> var.lnh2<-Z0%*%varb%*%t(Z0)+diag(R0)
> hpred.f<-1.3+exp(pred.f+0.5*diag(var.lnh2))
```

CHAPTER 3. LINEAR MIXED MODELS

> plot(d, hpred.f, type="l", lwd=2, lty="dashed")
> lines(d, hpred.fr, lwd=2)
> points(dobs, hobs)

# Let us make the corresponding predictions using fm2
> Dg <- 10
> mu2 <- cbind(1, 1/dobs, log(Dg), Dg/dobs) %*% fixed.effects(fm2)

# V1 = the variance-covariance matrix of random effects
# V2 = the variance-covariance matrix of heights, i.e., ZZ′ + R
# where D = V1
> Z <- cbind(1, 1/dobs)
> rho <- attributes(fm2$apVar)$Pars[4]
> sigma <- fm2$sigma
> K <- diag(sigma^2 * dobs^2 * (2 * rho))
> V2 <- Z %*% V1 %*% t(Z) + K
> V12 <- V1 %*% t(Z)

# The blup of random effects is
> bhat <- V12 %*% solve(V2) %*% (h2 - mu2)

# plot the localized or calibrated H-D curve.
# Save the predictions to vectors
# pred.f - logarithmic prediction based on the fixed part only
# pred.fr - logarithmic prediction based on fixed and random parts
> d <- seq(1, 50)
> plot(d, pred.f2 <- cbind(1, 1/d, log(Dg), Dg/d) %*% fixed.effects(fm2), type="l", col="green", lwd=2)
> points(dobs, h2)
> lines(d, pred.fr2 <- cbind(1, 1/d, log(Dg), Dg/d) %*% (fixed.effects(fm2) + c(bhat, 0, 0)), type="l", col="green", lwd=2)

# Transformation to the original scale.
# Half of the prediction variance needs to be added before back-transformation.
# 1. The prediction based on fixed and random parts
> var.lnh2 <- Z0 %*% varb %*% t(Z0) + diag(R0)
> hpred.fr2 <- 1.3 + exp(pred.fr2 + 0.5 * diag(var.lnh2))

# 2. The prediction based on fixed part only
> var.lnh2 <- Z0 %*% D %*% t(Z0) + diag(R0)
> hpred.f2 <- 1.3 + exp(pred.f2 + 0.5 * diag(var.lnh2))

> plot(d, hpred.f, type="l", lwd=2, lty="dashed")
> lines(d, hpred.fr, lwd=2)
> lines(d, hpred.f2, lwd=2, lty="dashed", col="red")
> points(dobs, hobs)

3.8 Exercises

1. Show that in the linear mixed-effects model $y_k = X_k b + Z_k c_k + e_k$, $\text{var}(y_k) = Z_k D Z_k^\prime + R$ and $\text{cov}(c_k, y_k) = D Z_k^\prime$.

2. Perform similar analysis as we did in the last section of this chapter using spruce
dataset from file spruce.txt. The data structure is similar to the pine data used in the analysis.

(a) Analyse possible model shapes.

(b) Develop and fit appropriate linear mixed-effects model. Report different stages of the modeling work and justify the choices you have made.

(c) File sprucestand.txt includes 60 measured height diameter pairs from a stand that was not included in the modeling data. Select randomly (i) one sample tree and (ii) five sample trees from the example stand, and compute the stand effects of your model. Plot all the data of the example stand. Add to the plot (i) the prediction of the fixed part (ii) conditional (localized, calibrated) prediction based on the one sample tree, (iii) the conditional (localized, calibrated) prediction based on the five sample trees, and (iv) mark the sample trees utilized in prediction.

3. File hdmod.R includes functions that can be used for easy application of the longitudinal height models of (Mehtätalo 2004b) and (Mehtätalo 2005a).

(a) Run the code using the measurements given in the example at the end of the file.

(b) Use the model for prediction for a pine stand stand with DGM=30 cm, and two sample trees measured 10 years ago. At that time, DGM was 22 cm, and the measured diameters were 15 and 22 cm, and the corresponding heights were 14 and 19 meters.

(c) Plot the fixed part predictions, and localized H-D curves 10 years before and now.

4. Using plots 1-16 of data pinevol, analyze the dependence of tree volume on diameter. Use the fitted model for prediction for plot 51. Some help for this data can be found in Lappi et al. (2006).

5. Data patti.txt includes observations on increment cores using densiometer. The data includes the following variables: Plot: plot id, Tree: tree id, SDClass: thinning treatment (1=control ... 3=heavy), Diam1986: DBH before thinning; Year: the year of the increment; CA: Age of the tree; RW ring width (mm); RD: ring density; RBA: Ring basal area. Fit a model for the development of ring basal area or ring width for the years before thinning.

6. Extract the thinning effects of ring basal area in the dataset patti.txt.
Chapter 4

Nonlinear models

4.1 Nonlinear fixed-effects model

4.1.1 Introduction and motivating example

See background on nonlinear models from (Fox and Weisberg 2010) Nonlinear Regression and Nonlinear Least Squares in R An Appendix to An R Companion to Applied Regression, second edition John Fox Sanford Weisberg

The regression may be either nonlinear or linear. In this context, the linearity means that variable $y$, or any transformation of it, is linear in predictors $x$ or any transformations of it.

In linear regression, we wrote the relationship between $x_1, \ldots, x_n$ and $y$ as

$$f_y(y) = \beta_0 + \beta_1 f_1(x_1) + \beta_2 f_2(x_1) + \ldots + \beta_p f_p(x_p),$$

where $\beta_0, \ldots, \beta_p$ are parameters to be estimated from the data.

In nonlinear regression, the relationship may be expressed with any function

$$y = f(x_1, \ldots, x_q, \beta_1, \ldots, \beta_p).$$

For example, we could assume the power-type relationship

$$y = \beta_0 x^{\beta_1}, \quad (4.1)$$

between $x$ and $y$. This relationship is no more linear. However, the relationship of equation (4.1) can be linearized by taking logarithms from both sides to get

$$\ln y = \ln \beta_0 + \beta_1 \ln(x). \quad (4.2)$$

The above example shows a very common way to linearize nonlinear relationships, namely taking logarithms.
In many cases the relationship cannot be easily linearized. The widely used and flexible Champman-Richards function (Richards 1959) is an example of such a function that cannot linearized easily

\[ y = \beta_1 \left(1 - e^{-\beta_2 x}\right)^{\beta_3}. \]

**Example 4.1.** A total of 200 Scots pine trees were measured for diameter and height from 44 plots in North Carelia. The data were read to R using

```r
> hddata<-read.table("d:/laurim/biometria/hddata.txt",header=TRUE)
```

We start by plotting the data. We also fit a simple model and add it onto the plot (the narrow straight line in the upper graph of figure 4.1.1) to see if the assumption on linear relationship would be realistic.

```r
> windows(width=2.5,height=6)
> par(mfcol=c(3,1),mai=c(0.6,0.5,0.1,0.1),mgp=c(2,0.7,0),cex=0.8)
> plot(hddata$d,hddata$h)
> # Fit the model
> fm1<-lm(h~d,data=hddata)
> summary(fm1)

Call:
  lm(formula = h ~ d, data = hddata)

Residuals:
     Min      1Q  Median      3Q     Max
-5.7400 -1.8560 -0.1990  1.9440  5.8470

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.872411   0.384981  12.651  < 2e-16 ***
d            0.551805   0.021764   25.364  < 2e-16 ***

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.533 on 198 degrees of freedom
4.1. NONLINEAR FIXED-EFFECTS MODEL

Figure 4.2: Illustration of the effect of transformation in the predictor.
CHAPTER 4. NONLINEAR MODELS

> fit1<-coef(fm1)[1]+coef(fm1)[2]*hddata$d
> # add the regression line into the plot
> abline(fm1)
> # compute the mean of residuals
> mean(hddata$h-fit1)
[1] -6.393291e-16

The plot shows slight curvature of the data. However, we see that the mean of residuals is close to 0, which indicates that the fitted values are unbiased over the whole data, i.e., they are marginally unbiased. However, the observed curvature indicates that they are not conditionally unbiased, e.g., for small diameters (say, d<8 cm), the model overestimates tree height.

Plotting the logarithmic heights against logarithmic diameters shows that model 4.2, could fit the data better (see the middle plot of 4.1.1). The model is fitted in R using code

> plot(log(hddata$d),log(hddata$h))
> fm2<-lm(log(h)~log(d),data=hddata)
> summary(fm2)

Call:
  lm(formula = log(h) ~ log(d), data = hddata)

Residuals:
   Min     1Q Median     3Q    Max
-0.6549 -0.1128  0.0078  0.1552  0.4040

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.833380   0.064242  12.973  < 2e-16 ***
log(d)        0.648950   0.024084  26.928  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1922 on 198 degrees of freedom
Multiple R-Squared: 0.7856,   Adjusted R-squared: 0.7845
F-statistic: 725.4 on 1 and 198 DF,  p-value: < 2.2e-16
> abline(fm2)

However, the fit is unbiased only for logarithmic heights. For the heights in meters, the estimated bias is 0.22 meters, as shown below. This means that the model underestimates tree heights in the modeling data, on average, by 0.22 meters.

> fit2<-coef(fm2)[1]+coef(fm2)[2]*log(hddata$d)
> # mean of residuals in meters
> mean(hddata$h-exp(fit2))
[1] 0.2168802

The observed bias result arises from the difference in the means of lognormal and normal distributions. The expected value of the lognormal distribution is \(e^{\mu+\sigma^2/2}\). Based on this result, a correction for bias could be performed by adding half of the residual variance into the logarithmic predictions before exponential transformation. However, this bias correction relies on the assumption of normality on residuals, which would not be otherwise needed. However, the same bias correction could be derived also based on Taylor series (Lappi 1991). The predictions corrected for bias are computed below.
We notice that the mean of residuals was clearly reduced in absolute value. However, it is still higher than the mean of residuals when the response was height instead of logarithmic height. Also other bias corrections have been proposed. For example, one can compute a correction factor based on the observed bias, and use it to adjust the predictions. Another approach is a two-point distribution approach, where predictions for lower and upper 67% confidence intervals are first computed in the logarithmic scale, the predictions are transformed to original scale, and finally the mean of these two predictions are taken as the final prediction (see, Lappi et al. (2006)).

An alternative to linearizing the model is to fit it in the original power form using nonlinear least squares. To do so, we first define a R-function for power equation and then fit the model. An essential difference between the linear and nonlinear regression is the need of initial guesses for the parameter estimates. In this example, the guesses can be based on the fit of the linearized version of the model (4.2). Coefficient $\beta_0$ is obtained from the constant of model $fm2$ by using the exponential transformation, because the the estimated constant of the linear model was estimate for $\ln \beta_0$, and the initial gues for $\beta_1$ is directly the estimated slope of the logarithmic model.

The graphs of models $fm2$, $fm3$ and $fm4$ were added to the upper graph of figure 4.1.1 using thick dashed gray, dotted black, and solid black lines, respectively.
4.1.2 The nonlinear regression model

In the nonlinear regression, we assume the model

\[ y_i = f(x_{1i}, \ldots, x_{qi}, \beta_1, \ldots, \beta_p) + e_i, \]

or

\[ y = E(y|x) + \epsilon = f(x; \beta) + \epsilon \]

where we have \( q \) predictors and \( p \) parameters. Term \( e_i \) is the residual and it has the same interpretation as in the context of linear models. Usually we assume that the residual has mean 0 and constant variance, but also nonhomogeneous variance can be assumed by using variance functions in similar way as with linear models. To make inference on the model or to use maximum likelihood in estimation, we may make an additional assumption that the residual is normally distributed.

Form a technical point of view, the basic differences in the nonlinear and linear regression are

- The regression function can be any function of the predictors and parameters
- The number of parameters is not necessarily the \( n + 1 \)
- Initial guesses for the parameters need to be given for estimation. However, model-specific algorithms to find the initial guesses can be implemented.
- There is no need for transformations of the response, and the predictions from the model are unbiased in the original scale.

From a more philosophical point of view, the differences between nonlinear and linear regression are (Pinheiro and Bates 2000).

- The parameters may have a good interpretation
- The model is more parsimonious than a linear model would be
- The model have better validity beyond the observed range of the data

These points are valid especially if one compares a polynomial model to a nonlinear model.

4.1.3 Estimation using nonlinear least squares

The nls estimates \( \beta \) by minimizing the residual sum of squares,

\[ S(\beta) = \sum [y - f(\beta; x)]^2 \]
with respect to $\beta$. The resulting least squares solution is noted as $\hat{\beta}$. In the linear least-squares problem, we could find a closed form solution for $\hat{\beta}$. With nonlinear regression, this is not the case. That is why an iterative procedure is performed to find the estimates:

1. The user gives an initial guess for parameter $\beta$, denote it by $b$. If the guess is good enough, the algorithm will find a solution of $\beta$ that minimizes $S$.

2. At each iteration, the guess for $\hat{\beta}$ is updated. If the updated estimate provides decrease to $S(\beta)$, then the current guess is accepted and iteration os continued.

3. The iteration is continued until (i) the algorithm cannot found such a guess of $\beta$ that would provide decrease to $S(\beta)$ (i.e., estimation converged) or (ii) the pre-specified number of iterations is completed or the improvement was below a pre-specified limit for significant improvement (i.e., estimation failed)

By default, R function `nls` uses the Gauss-Newton iteration using numerically evaluated gradients. However, the user may supply the gradients as well. The user should also be aware on the possibility to have such function $S$ that has several minima, and the iteration may find only the local minimum but not the global one.

In estimation, the nonlinear function is approximated by a piece of linear function. After this approximation, the estimators are very similar to those of linear models. Also the tests are based on thios linear approximation. Therefore, the test results are only approximate. The approximate is teh more accurate the better the linear approximations are. Thus, the more heavily nonlinear the model is with respect to the parameters, the more caution should be used in interpreting the test results.

More information on the estimation of nonlinear model can be found in Seber and Wild (1989).

**Example 4.2.** Modeling of thinning effects using logistic curve Dataset thinning.txt includes estimates of the effect of thinning on the basal area growth ($mm^2/yr$) of a Scots Pine tree in Eastern Finland. The tree is a part of a larger dataset including several sample plots of different thinning treatments. The data are based on measured increment core widths from felled trees of the thinning experiment; the measurements were taken using an X-ray microdensiometer. The observed ring were first transformed to ring basal areas. A linear mixed-effects model fitted to the data of the data of 88 trees (observations from thinned stands after thinning year were excluded) was used to predict the age-trend without thinnings, annual calendar year effects, and tree specific random effects. These effects have been subtracted from the observed time series of basal area growths. The resulting time series of thinning effect as a function of calendar year is shown in Figure 4.3.
Figure 4.3: The observed thinning effect (dashed) after a heavy thinning in winter 1986-1987 (vertical line) and the fitted logistic curve (solid) for one Scots Pine Tree in North Carelia.
4.1. NONLINEAR FIXED-EFFECTS MODEL

Figure 4.3 shows a clear thinning effect as a function of time since thinning. However, the tree needs time to recover from the competition before thinning, and therefore the full thinning effect is reached only after some years after thinning. To model this recovery, we model the thinning effect at year $t$ with the following logistic function

$$ThEf_t = \frac{\phi_1}{1 + \exp\left\{\frac{(\phi_2 - T_t)}{\phi_3}\right\} + e_t},$$

where $T_t$ is the calendar year at time $t$. The parameters have the following interpretations (Pinheiro and Bates 2000, p.519)

- $\phi_1$: The upper asymptote, i.e. the full thinning effect after the recovery period,
- $\phi_2$: The calendar year when half of the thinning effect is reached, and
- $\phi_3$: The rate parameter (the time needed to increase the thinning effect from 50% to 73%)

This function has been implemented in R library nlme as a self-starting nonlinear function with name SSlogis. Therefore, the fitting of this model can be done without finding initial parameter estimates.

```r
> plot(thinning$Year, thinning$ThEff, type="l", xlab="Year", ylab="Thinning effect in Ring Basal area, mm^2")
> m <- nls(ThEff ~ SSlogis(Year, phi1, phi2, phi3), data=thinning)
> summary(m)
Formula: ThEff ~ SSlogis(Year, phi1, phi2, phi3)
Parameters:

  Estimate Std. Error t value Pr(>|t|)
phi1 341.4070   15.5300  21.984  1.76e-15 ***
phi2 1988.5600   0.4840 4108.758 < 2e-16 ***
phi3  1.3455    0.4207   3.198  0.00451 **

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53.81 on 20 degrees of freedom
Number of iterations to convergence: 0
Achieved convergence tolerance: 3.323e-06
> lines(thinning$Year, fitted(m))
```

The parameter estimates of the model show that the maximum thinning effect is 341 mm$^2$/yr, and half of the thinning effect is reached between years 1988-1989. The rate parameter is 1.34 years. The smooth curve in figure 4.3 shows a good fit to the data. However, we notice that the thinning effect seems to have started already at year 1985; two years before the thinning. This phenomena is an artifact resulting from that the random variation in the growth of this tree has a slightly increasing trend just before the thinning.

By restricting the ratio of the scale and rate parameters we can force the thinning effect to start at the thinning year. Using the time since thinning as the predictor, restriction $\phi_3 = \phi_2/4$ ensures that the value of the logistic curve at $t = 0$ is always
0.180; i.e. only 2% of the full thinning effect is accumulated at the year of thinning. We recognize that in reality this value should be 0, but also notice that 2% is not too high, and point out that requesting this property strictly would force us to switch from the logistic curve to something else because of the asymptotic behaviour of the logistic function in the tails.

An additional problem is the interpretation of $\phi_2$ as the half of the recovery time, a more natural parameterization being the full recovery time (or more specifically, the time needed to reach 98% of the full thinning effect). To implement this, we reparameterize the original parameter $\phi_2$ as $\phi_{2,new} = 2\phi_2$ to get the following interpretations:

- $\phi_1$: The upper asymptote, i.e. the full thinning effect after the recovery period,
- $\phi_{2,new}$: The length of the recovery period, years

The newly parameterized logistic function is defined below by name mylogis. Note that we have left the original expression untouched but just written the reparameterizations in the first lines of the function. The way of writing functions is a matter of taste; I prefer writing the new transformations as new lines in the function to make it easier to see the changes and to further edit them.

```r
mylogis<-function(t,phi1,phi2n) {
  phi2<-phi2n/2 # phi2 is the full reaction time
  phi3<-phi2/4 # We assume that the rate is 1/8th of the
  # reaction time, which ensures that the
  # reaction starts approximately when t=0
  phi1/(1+exp((phi2-t)/phi3))
}
```

The next question is how to get the starting values now that we cannot anymore use the self-starting function. With this kind of simple functions, a practical and simple way is to plot the function with different parameter values and select the one that goes somewhat nicely throughout the observations. We tried three different recovery period lengths, and used the value 320 for the thinning effect size. The graph in figure 4.4 suggests that the reaction time of four years could be a good starting estimate (even though any other tried values would lead to convergence as well).

```r
> plot(thinning$Year-1986,thinning$ThEff,type="l",
+     xlab="Time since thinning, yr",
+     ylab="Thinning effect in Ring Basal area, mm^2")
> tapu<-seq(0,20,0.5)
> lines(tapu,mylogis(tapu,320,2),lty="dashed")
> lines(tapu,mylogis(tapu,320,4),lty="solid")
> lines(tapu,mylogis(tapu,320,6),lty="dotted")
> thinning$Ythin<-thinning$Year-1986
> m2<nls(ThEff~mylogis(Ythin,phi1,phi2n),
+     data=thinning,
+     start=list(phi1=320,phi2n=4))
> summary(m2)
```

Formula: ThEff ~ mylogis(Ythin, phi1, phi2n)

Parameters:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| phi1     | 330.0811   | 14.4298 | 22.875   |
| phi2n    | 4.6415     | 0.6421  | 7.228    |

...
Figure 4.4: The original data and the graph of function mylogis with different parameter values.
The estimates from the newly parameterized model suggest the full thinning effect size of 330 mm$^2$/year, which is slightly lower than the estimate from the previous model, and a recovery time of 4.64 years. These values sound logical; especially the recovery time being slightly longer than the time required to renew all the needles (3-4 years for Scots Pine).

To evaluate the goodness of fit, we plot the residuals of the model on predicted value (Figure 4.5). The graph indicates increasing variance as a function of predictor. To formally test this, we re-estimate our model using function `gnls` of package `nlme`, which allows different models for heteroscedastic residual variance. We try the power-type variance function and make the approximate likelihood ratio test to see whether the model allowing heteroscedastic variances fits the data significantly better. Fitting the model without variance function using `gnls` fails in NLS step. This problem could be fixed by decreasing the `minFactor` through the `control` argument of `nls` (see `?gnlsControl`).

```r
> m1<-gnls(ThEff~mylogis(Ythin,phi1,phi2n),
+ data=thinning,
+ start=list(phi1=320,phi2n=4))
Error in gnls(ThEff ~ mylogis(Ythin, phi1, phi2n), data = thinning, start = list(phil = 320, :
  Step halving factor reduced below minimum in NLS step
>
> m1<-gnls(ThEff~mylogis(Ythin,phi1,phi2n),
+ data=thinning,
+ start=list(phi1=320,phi2n=4),
+ control=list(minScale=0.0001))
>
> m2<-gnls(ThEff~mylogis(Ythin,phi1,phi2n),
+ data=thinning,
+ start=list(phi1=320,phi2n=4),
+ weights=varPower())
> anova(m1,m2)
Model df AIC BIC logLik Test L.Ratio p-value
m1 1 3 254.7567 258.1632 -124.3784
m2 2 4 254.7109 259.2529 -123.3555 1 vs 2 2.045762 0.1526
```

### 4.1.4 Initial estimates

Initial estimates can be found at least using the following procedures
Figure 4.5: The residuals of thinning effect model m2 on the prediction.
• Making a rough guess, testing your success.

• Plotting the function to be estimated onto the data and exploring which parameter values make the function to follow roughly the data. Usually it is enough to “find the data”, i.e., to adjust the parameter values so that the function changes its values in somewhat realistic manner within the range of predictors in the data (See example 4.2)

• Performing a grid search, i.e., defining a region of realistic values and trying systematically several possible combinations from different parts of the region.

• Linearizing the model using appropriate transformations, fitting the linearized model into the data, and finally recovering the estimates of the original parameters that correspond to the estimates of the linearized model (See example 4.1)

4.1.5 Different parameterizations in nls models

In nonlinear regression it is important to recognize that the same mathematical function can be parameterized in several different ways. For example, ?, p. 12 present 7 different parameterizations of the same mathematical function, arising from different disciplines and scientific traditions. However, the parameterizing of a nonlinear function is not only a matter of tradition, but something that one should keep in mind in fitting the model.

Example 4.3. In example 4.2, the following function was used to model the thinning effects:

\[
f(x) = \frac{\phi_1}{1 + \exp\left[8 (\phi_2/2 - x)/\phi_2\right]},
\]  

(4.3)

Figure 4.3 shows the value of this function using different values of parameters. Naturally, a negative value of asymptote \( \phi_1 \) yields a decreasing function. The negative value of \( \phi_2 \) causes a big change to the model shape.

Another parameterization of the function is provided by specifying \( \phi_1 = \exp(\theta_1) \) and \( \phi_2 = \exp(\theta_2) \), i.e., by making exponential transformations to parameters. This yields

\[
f(x) = \frac{\exp(\theta_1)}{1 + \exp\left[8 (\exp(\theta_1)/2 - x)/\exp(\theta_2)\right]},
\]  

(4.4)

The new parameterization provides the same function than the earlier function. However, as \( \exp(\theta) > 0 \) for all values of \( \theta \), the new parameterization restricts the potential shapes of the function to those where both \( \phi_1 > 0 \) and \( \phi_2 > 0 \). This parameterization may be useful e.g. in modelling the thinning effects.
4.1. NONLINEAR FIXED-EFFECTS MODEL

Figure 4.6: The value of function (4.3) for different combinations of negative and positive values of parameters $\phi_1$ and $\phi_2$.

The two possible parameterizations were fitted to the thinning effect data of example 4.2 using nls. The resulting residual errors are exactly the same and parameters also match (i.e., exponential transformations of the latter model parameters equal to the estimates of the first model). However, quite a big difference occurs in the test statistics of the coefficients (t-values). In the first model, the values are 22.875 and 7.228 for $\phi_1$ and $\phi_2$, respectively, whereas in the new parameterization they are 132.7 and 11.1 for the corresponding parameters $\theta_1$ and $\theta_2$. Even though this difference does not lead to any change in the final decision about whether to include a parameter in a model or not in this case, it is easy to believe that in some cases different parameterizations may lead to different conclusions on the significance of some model parameters.

```r
> mylogis2<-function(t,theta1,theta2) {
+  phi1<-exp(theta1)
+  phi2n<-exp(theta2)
+  phi2<-phi2n/2  # phi2 is the full reaction time
+  phi3<-phi2/4  # We assume that the rate is 1/8th of the
+  # reaction time, which ensures that the
+  # reaction starts approximately when t=0
+  phi1/(1+exp((phi2-t)/phi3))
+}
>
> m2<-nls(ThEff˜mylogis(Ythin,phi1,phi2n),
+  data=thinning,
+  start=list(phi1=320,phi2n=4))
>
> m2b<-nls(ThEff˜mylogis2(Ythin,theta1,theta2),
+  data=thinning,
+  start=list(theta1=log(320),theta2=log(4)))
>
> summary(m2)
```

Formula: ThEff ~ mylogis(Ythin, phi1, phi2n)
### 4.2 Nonlinear mixed-effects models

Nonlinear mixed-effects models generalize the linear mixed effects models in the similar way as do nonlinear models generalize the linear models.

#### 4.2.1 Single level of grouping

Denote the response variable $y$ of individual $i$ of group $k$ by $y_{ki}$ and the corresponding predictor variable by $x_{ki}$. For example, $y_{ki}$ and $x_{ki}$ may be the height of tree $i$ on plot $k$, or diameter increment of tree $k$ at year $i$. The nonlinear mixed-effects model for group $k$ can be expressed as

$$y_{ki} = f(x_{ki}; \beta + b_k) + e_{ki} \quad i = 1, \ldots, M; j = 1, \ldots, n_i,$$

where vector $\beta$ is the vector of fixed effects and $b_k$ is the vector of random effects with $b_k \sim N(0, \Psi)$. Term $e_{ki}$ is the unexplained residual. An usual assumption is that the residuals independent and have equal variance, i.e. $\text{var}(e_{ki}) = \sigma^2_k$. For fitting using the method of maximum likelihood, and for tests on the regression coefficients, an additional assumption on normality is needed. In addition, a multivariate normal distribution is usually assumed for the random effects as $b_k \sim N(0, \Psi)$.

If we have more than just one covariate, we will have

$$y_{ki} = f(x_{ki}; \beta + b_k) + e_{ki}$$
This formulation assumes that we have a random effect corresponding to each of the fixed effects. However, this is not necessarily the case. A more general formulation (but probably a bit more time-consuming to understand) is the following

\[ y_{ki} = f(x_{ki}; \phi_{ki}) + e_{ki}, \]

where

\[ \phi_{ki} = A_{ki} \beta + B_{ki} b_i \quad b_i \sim N(0, \Psi) \]

where \( A_{ki} \) and \( B_{ki} \) are simple incidence matrices that select from the vectors of fixed and random effects of group \( k \) those elements that are related to the parameter of question. They have as many rows as the model has parameters. The number of columns in \( A_{ki} \) equals to the number of fixed coefficients and the number of columns in \( B_{ki} \) equals to the number random effects.

The model for group \( k \) can be written in the matrix form

\[ y_k = f_k(x; \phi_k) + e_k, \]

where

\[
\begin{align*}
y_k &= \begin{bmatrix} y_{k1} \\ \vdots \\ y_{kn_k} \end{bmatrix}, \\
\phi_k &= \begin{bmatrix} \phi_{k1} \\ \vdots \\ \phi_{kn_k} \end{bmatrix}, \\
e_k &= \begin{bmatrix} e_{k1} \\ \vdots \\ e_{kn_k} \end{bmatrix}, \\
f_k(x; \phi_k) &= \begin{bmatrix} f(x_{k1}; \phi_{k1}) \\ \vdots \\ f(x_{kn_k}; \phi_{kn_k}) \end{bmatrix}, \\
v_k &= \begin{bmatrix} v_{k1} \\ \vdots \\ v_{kn_k} \end{bmatrix}, \\
A_k &= \begin{bmatrix} A_{k1} \\ \vdots \\ A_{kn_k} \end{bmatrix}, \\
B_k &= \begin{bmatrix} B_{k1} \\ \vdots \\ B_{kn_k} \end{bmatrix}.
\end{align*}
\]

The tests on nested models can be based on the likelihood ratio tests, and those tests can be performed in R by using \texttt{anova(model1, model2)}. These tests are asymptotic and approximate, but are generally used in textbooks. Non-nested models can be compared against each other by using different information criteria (AIC, BIC). The significance of fixed coefficients can also be compared using likelihood ratio tests with models fitted using maximum likelihood. However, those tests may be badly anti-conservative for fixed-effects. Therefore, Pinheiro and Bates (2000) suggest Wald test for the fixed effects.

Predictions from nonlinear mixed-effects models can be done at different levels of hierarchy. With linear mixed-effects model, the random effects enter into the model in a linear form. Therefore, with LMM’s in balanced datasets (i.e., in experimental datasets where all groups have the same number of observations and the same values of predictors), the fixed part gives the expected value of the response for the given values of the predictor; i.e. \( E(y|X) = E(Xb + Ze + e) = Xb + ZE(e) + E(e) = Xb \).
the nonlinear mixed-effects models, this does not hold even with balanced datasets because the model is nonlinear with respect to the random effects. However even with linear mixed-effects models, the lack of balance may violates the unbiasedness because, the distribution of random effects may not be the same for different values of the predictor. For example, for plots with low diameters, the random effects for a H-D model may be usually positive, whereas for plots with high diameters they may be commonly negative. Therefore, the fixed part of a mixed-effects model, either linear or nonlinear, should be interpreted as a group-specific model for an average group in the data; not as the expected value of the response conditional on the fixed predictor. As a result, using the fixed part of the mixed-effects model as the prediction usually performs worse than using a prediction from a marginal model; i.e. from a model that ignores the hierarchical structure of the data.

**Example 4.4.** Dataset “thinningeffects” includes estimated effects of thinning on the basal area growth of Scots Pine trees in Eastern Finland. The effects of tree age, climatic year effects, and tree-specific random effects have been removed as described in Example 3.12.

The variable ThEf includes the thinning effects. In addition, the tree age (CA) and thinning treatment (SDClass: 1=light, 2=moderate, 3=heavy) are known for each plot. The data are grouped according to plots and trees within plot, but we use only the tree level in this example. In addition, the data includes information on the calendar year, but that information is not used in this example.

Our starting model for the thinning effect of year \( t \) for tree \( i \) is of the logistic form:

\[
ThEf_{it} = \frac{\exp(\theta_1 + a_i)}{1 + \exp\left[8 \left(1/2 - x/\exp(\theta_2 + b_i)\right)\right]} + e_{it}, \tag{4.6}
\]

The model is fitted and the summary printed using the following code. The initial estimates are guessed from figure 4.7, which indicates that the thinning effect size is around 200 mm² and the reaction time is roughly 5 years. We experience a convergence problem even with this simplest model: The call ends to an error saying that the maximum number of iterations is reached without convergence. Such problems are common with nonlinear mixed-effects models, but they can be sometimes fixed by adjusting the parameters of the model fitting algorithm. This happened also in this case (see `nlmeControl`)

```r
library(nlme)
library(lmfor)

## Add variable for years since thinning
patti$ythin <- patti$Year - 1986

## Patti3 includes observations after thinning
patti3 <- patti[patti$Year > 1986,]
```

Example 4.4. Dataset “thinningeffects” includes estimated effects of thinning on the basal area growth of Scots Pine trees in Eastern Finland. The effects of tree age, climatic year effects, and tree-specific random effects have been removed as described in Example 3.12.

The variable ThEf includes the thinning effects. In addition, the tree age (CA) and thinning treatment (SDClass: 1=light, 2=moderate, 3=heavy) are known for each plot. The data are grouped according to plots and trees within plot, but we use only the tree level in this example. In addition, the data includes information on the calendar year, but that information is not used in this example.

Our starting model for the thinning effect of year \( t \) for tree \( i \) is of the logistic form:

\[
ThEf_{it} = \frac{\exp(\theta_1 + a_i)}{1 + \exp\left[8 \left(1/2 - x/\exp(\theta_2 + b_i)\right)\right]} + e_{it}, \tag{4.6}
\]

The model is fitted and the summary printed using the following code. The initial estimates are guessed from figure 4.7, which indicates that the thinning effect size is around 200 mm² and the reaction time is roughly 5 years. We experience a convergence problem even with this simplest model: The call ends to an error saying that the maximum number of iterations is reached without convergence. Such problems are common with nonlinear mixed-effects models, but they can be sometimes fixed by adjusting the parameters of the model fitting algorithm. This happened also in this case (see `nlmeControl`)

```r
library(nlme)
library(lmfor)

# Add variable for years since thinning
patti$ythin <- patti$Year - 1986

# Patti3 includes observations after thinning
patti3 <- patti[patti$Year > 1986,]
```
Figure 4.7: The upper graph shows the dataset of ring basal areas for the whole period. Thin dashed lines are used for observations of thinned plots after the thinning, solid, modelrately thick lines show the observations from thinned plots before thinning and control plots for the whole period. The thick lines show the average trend, based on a lowess smoother, separately for each thinning treatment. The colors indicate the thinning treatment as follows: Black=control, red=light, green=moderate and blue=heavy thinning. The dashed lines in the lower graph shows the extracted thinning effects and the thick solid lines the corresponding lowess smoothers by treatments.
\# Patti4 includes observations of thinned plots after thinning.  
patti4<-patti3[patti3$SDClass>1,]  
\# We define patti4 as a grouped data object to allow  
\# the use of grouped data plotting functions  
patti4<-groupedData(ThEf~ythin|Tree,data=patti4)  
> ThinNlme0<-nlme(ThEf~mylogis2(ythin,theta1,theta2),  
+ fixed=list(theta1~1,theta2~1),  
+ groups=˜Tree,  
+ data=patti4,  
+ start= c(log(200),log(5)),  
+ control=list(maxIter=500,nsMaxIter=500,tolerance=1e-6,pnlsMaxIter=20,niterEM=50)  
+ )  
> summary(ThinNlme0)  
Nonlinear mixed-effects model fit by maximum likelihood  
Model: ThEf ~ mylogis2(ythin, theta1, theta2)  
Data: patti4  
AIC BIC logLik  
14853.83 14884.55 -7420.913  
Random effects:  
Formula: list(theta1 ~ 1, theta2 ~ 1)  
Level: Tree  
Structure: General positive-definite, Log-Cholesky parametrization  
StdDev Corr  
theta1 0.7907342 theta1  
theta2 0.5956760 0.45  
Residual 87.0659957  
Fixed effects: list(theta1 ~ 1, theta2 ~ 1)  
Value Std.Error DF t-value p-value  
theta1 4.876837 0.1075885 1175 45.32863 0  
theta2 1.643883 0.1004794 1175 16.36039 0  
Correlation:  
theta1 theta2 0.383  
Standardized Within-Group Residuals:  
Min Q1 Med Q3 Max  
-2.9567213 -0.6441442 -0.1594479 0.3881066 4.6699126  
Number of Observations: 1238  
Number of Groups: 62  
> exp(fixef(ThinNlme0))  
theta1 theta2  
131.214932 5.175224  

The results show that the thinning effect size for an average tree is \( \exp(4.87) = 131 \text{ mm}^2/\text{yr} \) and the reaction time is \( \exp(1.64) = 5.17 \) years. The residual standard error is 87 \( \text{mm}^2 \). The logarithmic thinning effect size of a tree varies around the mean over trees, 4.87, with the standard deviation of 0.79. The logarithmic reaction time in turn varies around the mean of 1.64 with the standard deviation of 0.60. The correlation between the random effects is moderate, 0.45. In general, the variation between trees is quite high compared to the mean values over the trees. This result was expected, because figure 4.7 showed high variation between trees in the reaction to thinning.

Next, we might want to explain some variation in the reaction times and thinning effect sizes by the thinning intensity. To start with such an analysis, we extract the random effects of the model and explore their dependence on the thinning treatment (Figure 4.8). We notice that both random effects have some correlation with the thinning, but especially the logarithmic thinning effect size correlates strongly with the treatment.
Figure 4.8: Random effects of the logarithmic thinning effect size (upper) and logarithmic reaction time (lower) on the thinning treatments (1 = light, ..., 3 = heavy).
Based on this observation, we extend our model to include the thinning treatments as the fixed predictors for both parameters $\theta_1$ and $\theta_2$

$$ ThEf_{it} = \frac{\theta_1.T1 + \theta_1.T2 + \theta_1.T3 + a_i + \exp\left(\frac{\theta_2.T1 + \theta_2.T2 + \theta_2.T3 + b_i}{1 - x/\exp(\theta_2.T1 + \theta_2.T2 + \theta_2.T3 + b_i)}\right)}{1 + \exp\left(\frac{\theta_2.T1 + \theta_2.T2 + \theta_2.T3 + b_i}{1 - x/\exp(\theta_2.T1 + \theta_2.T2 + \theta_2.T3 + b_i)}\right)} + e_{it}, \quad (4.7) $$

The Wald test of fixed coefficients shows that both parameters depend significantly on the treatment. The coefficients show that the thinning effect size is the higher the heavier the thinning. In addition, the reaction time is longer for heavy thinnings.

```r
> ThinNlme2<-nlme(ThEf~mylogis2(ythin,theta1,theta2),
+ fixed=list(theta1˜as.factor(SDClass),theta2˜as.factor(SDClass)),
+ random=list(theta1˜1,theta2˜1),
+ groups=˜Tree,
+ data=patti4,
+ start= c(init[1,],init[2,]),
+ control=list(maxIter=500,msMaxIter=500,tolerance=1e-6,pnlsMaxIter=20,niterEM=50),
+ )
> anova(ThinNlme2)
    numDF denDF F-value p-value
theta1.(Intercept) 1 1171 3385.370 <.0001
theta1.as.factor(SDClass) 2 1171 6.009 0.0025
theta2.(Intercept) 1 1171 104.413 <.0001
theta2.as.factor(SDClass) 2 1171 5.761 0.0032
> summary(ThinNlme2)
Nonlinear mixed-effects model fit by maximum likelihood
Model: ThEf ~ mylogis2(ythin, theta1, theta2)
Data: patti4

AIC  BIC  logLik
14802.72 14853.94 -7391.361

Random effects:
Formula: list(theta1 ~ 1, theta2 ~ 1)
Level: Tree
Structure: General positive-definite, Log-Cholesky parametrization

StdDev Corr
theta1.(Intercept) 0.7534821 t1.(I)
theta2.(Intercept) 0.8343218 0.806
Residual 85.3056976

Fixed effects: list(theta1 ~ as.factor(SDClass), theta2 ~ as.factor(SDClass))

Value Std.Error DF t-value p-value
theta1.(Intercept) 4.078573 0.1815826 1171 22.461262 0.0000
theta1.as.factor(SDClass)3 1.011646 0.2543856 1171 3.976820 0.0001
theta1.as.factor(SDClass)4 1.480885 0.256413 <.0001
theta2.(Intercept) 0.626110 0.2534281 1171 2.470565 0.0136
theta2.as.factor(SDClass)3 0.787603 0.3344415 1171 2.354979 0.0187
theta2.as.factor(SDClass)4 1.106197 0.3293077 1171 3.359160 0.0008
```

Figure 4.9 shows the predictions from the model. We notice that there are various age trends among the trees of the data. Therefore, we would like to allow the growth rate to decrease after the reaction time. One opportunity to model this is to allow the years since thinning to affect the thinning effect size. Thus, we would like to extend the model to

$$ ThEf_{it} = \frac{\theta_1.T1 + \theta_1.T2 + \theta_1.T3 + YTHIN_i t + a_i + c_i YTHIN_i t}{1 + \exp\left(\frac{\theta_2.T1 + \theta_2.T2 + \theta_2.T3 + b_i}{1 - x/\exp(\theta_2.T1 + \theta_2.T2 + \theta_2.T3 + b_i)}\right)} + e_{it}. \quad (4.8) $$

Because, this model fitting does not converge, we tried a model where the reaction time does not depend on treatment. That model was fitted using
Figure 4.9: The observed and predicted thinning effects of model ThinNlme2.
> ThinNlme3<-nlme(ThinNlme3<-nlme(ThEf~mylogis2(ythin,theta1,theta2), + fixed=list(theta1~as.factor(SDClass)+ythin,theta2~1), + random=list(theta1~1,theta2~1), + groups=Tree, + data=patti4, + start= c(init[1,,0,mean(init[2,]), + control=list(maxIter=500,maMaxIter=500,tolerance=1e-6,pnlsMaxIter=20,niterEM=50), + ) > anova(ThinNlme2,ThinNlme3)

<table>
<thead>
<tr>
<th>Model</th>
<th>df</th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>Test</th>
<th>L.Ratio</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThinNlme2</td>
<td>10</td>
<td>14802.72</td>
<td>14853.94</td>
<td>-7391.361</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ThinNlme3</td>
<td>9</td>
<td>14759.26</td>
<td>14805.35</td>
<td>-7370.629</td>
<td>1 vs 2</td>
<td>41.46557</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

> anova(ThinNlme3)

<table>
<thead>
<tr>
<th>numDF</th>
<th>denDF</th>
<th>F-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>theta1.(Intercept)</td>
<td>1</td>
<td>1172</td>
<td>995.6028</td>
</tr>
<tr>
<td>theta1.as.factor(SDClass)</td>
<td>2</td>
<td>1172</td>
<td>12.2070</td>
</tr>
<tr>
<td>theta1.ythin</td>
<td>1</td>
<td>1172</td>
<td>25.6141</td>
</tr>
<tr>
<td>theta2</td>
<td>1</td>
<td>1172</td>
<td>297.7307</td>
</tr>
</tbody>
</table>

We see that the new model is better than ThinNlme2 was. This can be seen by comparing the information criteria, which are lower for ThinNlme3 than for ThinNlme2. The p-value given by the anova should not be used because the two models we compare are not nested. In addition, LR-test is not recommended to compare models with different fixed effects even if they are nested (see Pinheiro and Bates (2000, p87, 322, 367)). With nlme, the anova-function cannot see whether the models are nested or not, and therefore it is the responsibility of the user to know when the two models can be compared using anova and when they cannot.

The Wald test for fixed coefficients shows that they are all significant also in the new model. Next, we test whether the dependence of thinning effect size on ythin should be included also in the random part.

> ThinNlme4<-nlme(ThinNlme4<-nlme(ThEf~mylogis2(ythin,theta1,theta2), + fixed=list(theta1~as.factor(SDClass)+ythin,theta2~1), + random=list(theta1~1+ythin,theta2~1), + groups=Tree, + data=patti4, + start= c(init[1,,0,mean(init[2,]), + control=list(maxIter=500,maMaxIter=500,tolerance=1e-6,pnlsMaxIter=20,niterEM=50), + ) > anova(ThinNlme3,ThinNlme4)

<table>
<thead>
<tr>
<th>Model</th>
<th>df</th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>Test</th>
<th>L.Ratio</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThinNlme3</td>
<td>9</td>
<td>14759.26</td>
<td>14805.35</td>
<td>-7370.629</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ThinNlme4</td>
<td>12</td>
<td>14536.15</td>
<td>14597.60</td>
<td>-7256.075</td>
<td>1 vs 2</td>
<td>229.1071</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

pdf("d:/laurim/biometria/lecturenotes/figpat11.pdf",width=4,height=7)
plot(augPred(ThinNlme3),cex=0.1)
dev.off()

dpdf("d:/laurim/biometria/lecturenotes/figpat12.pdf",width=4,height=7)
plot(augPred(ThinNlme4),cex=0.1)
dev.off()

dpdf("d:/laurim/biometria/lecturenotes/figpat13.pdf",width=4,height=7)
plot(ThinNlme4)
dev.off()

The LR-test shows a significant improvement to the model by including the variable ythin to explain the variation in the thinning effect size. The predictions from models 3 and 4 are shown in Figures 4.10 and 4.11. The residual plot of model ThinNlme4 is shown in Figure 4.12.
4.2. NONLINEAR MIXED-EFFECTS MODELS

Figure 4.10: The observed and predicted thinning effects of model ThinNlme3.
Figure 4.11: The observed and predicted thinning effects of model ThinNlme4.
4.2. NONLINEAR MIXED-EFFECTS MODELS

Finally, figure 4.13 demonstrates the predictions from the model at different levels of hierarchy. The fixed part provides predictions for the average tree of the thinning treatment, whereas the tree-level predictions express the prediction for each tree.

\[
\text{pdf("d:/laurin/biometria/lecturenotes/figpat15.pdf")}
\]
\[
linesplot(patti4$Year, patti4$ThEf, patti4$TreeID, col.lin=patti4$SDClass, cex=0,ylab="Thinning Effect", xlab="Year", lty="dashed")
linesplot(patti4$Year, predict(ThinNlme4), patti4$TreeID, col.lin=patti4$SDClass, cex=0,lwd=2,add=TRUE,lty="dashed")
linesplot(patti4$Year, predict(ThinNlme4,level=0), patti4$TreeID, col.lin=patti4$SDClass, cex=0,lwd=4,add=TRUE)
\]
\[
\text{dev.off()}
\]

4.2.2 Multiple levels of grouping

Nonlinear mixed-effects model can be generalized to multiple nested and crossed levels of grouping. Currently, R provides easy-to-use tools for only nested classifications in package nlme.

**Example 4.5.** A multilevel version of the model ThinNlme3 can be fitted using

\[
\text{ThinNlme8<-nlme(ThEf~mylogis2(ythin,theta1,theta2), fixed=list(theta1~as.factor(SDClass)+ythin,theta2~1), random=theta1+theta2~1|Plot/Tree,}
\]
Figure 4.13: The predictions of model ThinNlme4 at tree level (thick dashed) and the predictions based on the fixed part only (thick solid) for the three thinning treatments.

data=patti4,
# start= c(init[1,],init[2,]),
start= fixef(ThinNlme8),
control=list(maxIter=500,msMaxIter=500,tolerance=1e-6,pnlsMaxIter=20,niterEM=50),
# corr=corAR1(0.467,form=~1|Tree)
# weights=varPower()

> summary(ThinNlme8b)
Nonlinear mixed-effects model fit by maximum likelihood
  Model: ThEf ~ mylogis2(ythin, theta1, theta2)
  Data: patti4

AIC      BIC   logLik
14655.98 14717.43 -7315.99

Random effects:
  Formula: list(theta1 ~ 1, theta2 ~ 1)
  Level: Plot
  Structure: General positive-definite, Log-Cholesky parametrization

<table>
<thead>
<tr>
<th>StdDev</th>
<th>Corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>theta1.(Intercept)</td>
<td>0.003355539</td>
</tr>
<tr>
<td>theta2</td>
<td>0.000495369</td>
</tr>
</tbody>
</table>

Formula: list(theta1 ~ 1, theta2 ~ 1)
  Level: Tree %in% Plot
  Structure: General positive-definite, Log-Cholesky parametrization

<table>
<thead>
<tr>
<th>StdDev</th>
<th>Corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>theta1.(Intercept)</td>
<td>1.2651329</td>
</tr>
<tr>
<td>theta2</td>
<td>0.9239019</td>
</tr>
</tbody>
</table>

Residual 75.8023749

Fixed effects: list(theta1 ~ as.factor(SDClass) + ythin, theta2 ~ 1)

<table>
<thead>
<tr>
<th>Value</th>
<th>Std.Error</th>
<th>DF</th>
<th>t-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>theta1.(Intercept)</td>
<td>5.008045</td>
<td>0.2610991</td>
<td>1172</td>
<td>19.180632</td>
</tr>
<tr>
<td>theta1.as.factor(SDClass)3</td>
<td>1.214011</td>
<td>0.3262976</td>
<td>1172</td>
<td>3.726693</td>
</tr>
<tr>
<td>theta1.as.factor(SDClass)4</td>
<td>1.454453</td>
<td>0.3286423</td>
<td>1172</td>
<td>4.425643</td>
</tr>
<tr>
<td>theta1.ythin</td>
<td>-0.063160</td>
<td>0.0041759</td>
<td>1172</td>
<td>-15.125042</td>
</tr>
<tr>
<td>theta2</td>
<td>2.246348</td>
<td>0.1322136</td>
<td>1172</td>
<td>16.990292</td>
</tr>
</tbody>
</table>

Correlation:

<table>
<thead>
<tr>
<th>t1.(I)</th>
<th>t1.(SDC)3</th>
<th>t1.(SDC)4</th>
<th>tth1.y</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1.(I)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t1.(SDC)3</td>
<td>-0.631</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t1.(SDC)4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tth1.y</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.3 Exercises

1. File Spat63.txt includes plot 63 of the spati dataset. Model the H-D relationship using the best-fitting one from among the following three functions:

   \[ H(D) = BH + a \exp(-bD^{-1}) \]

   \[ H(D) = BH + \frac{aD}{(1 + D)^5} \]

   \[ H(D) = BH + \frac{D^2}{(a + bD)^\frac{3}{2}}. \]

2. Model the dataset Spat63.txt using the following three-parameter logistic function

   \[ H(D) = BH + \frac{a}{1 + b \exp(-cD)}. \]

3. File crowns.txt includes information on individual Spruce and pine tree crowns. It has the following information: Treed, tree species (1 = Scots Pine, 2=Spruce), height=Tree height (meters), maxarea=area of the crown projection at the ground (m²), p=relative height (between 0=ground level and 1=tree top), and r= crown radius (meters). Select one tree of the dataset. Model the crown shape as an ellipse that is centered at \((x_0, y_0)\) \((x_0 < 1, y_0 \leq 0)\):

   \[ Y(z_{ij}, h) = \begin{cases} 
   h_i (y_0 + b) & \frac{z_{ij}}{h_i} \leq x_0 \\
   h_i \left( y_0 + b \sqrt{1 - \frac{(z_{ij} - x_0)^2}{a^2}} \right) & x_0 \leq \frac{z_{ij}}{h_i} \leq 1 \\
   0 & \frac{z_{ij}}{h_i} > 1 
   \end{cases} \] 

   \[ (4.9) \]
where \( a = \sqrt{\frac{b^2(1-x_0)^2}{b^2-x_0^2}} \) is defined so that function \( Y(z_{ij}, h_i) \) passes through point \((h_i, 0)\).

4. Explore whether a model based on the Lame curve better describes the crown shape than the ellipsoidal model. The Lame curve is defined as

\[
\left( \frac{z - q h}{h - q h} \right)^t + \left( \frac{r}{R} \right)^t = 1
\]

(4.10)

where \( z \) is canopy height, \( h \) the height of the tree, \( q = \begin{bmatrix} 0 & 1 \end{bmatrix} \) the relative crown height, \( h - q h \) the distance between tree top and the maximum crown radius, \( r \) the crown radius, \( R \) maximum crown radius and \( t \) the shape parameter of Lamé curve.

5. Further develop the model of example 4.4.

(a) Explore whether the variation in the thinning effect size or reaction time could be explained by tree relative status of the tree in the plot (dominant or dominated) by exploring the dependence of the random effects on the diameter or tree relative diameter of the tree before thinning. Update the model correspondingly and report the model.

(b) Explore whether the assumption of equal variance was fulfilled. If needed, refit the model using an appropriate variance function.

(c) Try to model the residual autocorrelation using AR(1) or other possible processes.

6. Model the crown shape of dataset crowns.txt using a nonlinear mixed-effects model

7. Model crowns.txt using two alternative formulations: \( r^2 = f(H, R, h)^2 + e \) and \( r = f(H, R, h) + e \). Compare the resulting parameter estimates and predictions. Why do they differ? Which form would you suggest as the model for estimating crown area-related attributes such as area of canopy intersections at a given height in ALS applications.

8. package lmfor includes functions fithd, plot.hdmod and ImputeHeights. These functions automatically fit a nonlinear mixed-effects model into a provided data of H-D curves, and ImputeHeights uses the fitted model to make plot-specific predictions for imputation of tally tree heights. Using those functions, fit a best-fitting function to your own dataset and impute the missing heights if you have such ones.
Chapter 5

Generalized linear models (GLM’s) and Generalized linear mixed models (GLMM’s)

The linear mixed-effects model assume normality of the response. With the linear (fixed-effects) model, normality is needed for the test. However, the response may also have some other distribution. For example, it may be binary: dead or alive, success or failure. Or it may be count: e.g., number of dead trees within a plot. It can also be a percentage or proportion: e.g., canopy cover. Furthermore, the diameter of a tree within a stand or plot may be assumed to have a Weibull distribution. Even in those cases, the linear model may lead to fairly good model. However, a better justified model is obtained through generalized linear model (GLM). Note the distinction between terms GLS and GLM. They have nothing in common, except for the two first letters.

The generalized linear models are closely related to fitting assumed distributions to data using the method of maximum likelihood. Generalized linear model actually means just fitting the assumed distribution to the data using the maximum likelihood. The only difference to the examples of section 1.5.2 is that in GLM, the parameter(s) of the assumed distribution, or some functions of them, are written as a linear function of predictors, instead of assuming a common value for the whole data. To ensure that the predicted parameter is always within the support (e.g. probability is between 0 and 1, or the expected number of stems is positive), a link function is used to link a linear function of predictors to the parameter to be modeled. This link has exactly the same idea as we had in example 1.44, where the parameters of a Weibull-distribution were restricted to be above 0 by using a log link.

Section 2.7 presented the maximum likelihood estimation for the linear model. GLM generalizes that estimation method to the situation where the original data have
some other distribution than the normal.

5.1 Formulation of the LM as a GLM

With the linear model, the usual way of thinking is that there is a regression line which is usually of interest to the modeler. In addition, there is a normally distributed error term, which is added to the regression line to get the realized observations. This way of thinking is used in development of the least squares estimation methods, thus we call it the LS-formulation of the linear model. It can be written as

\[ y = Xb + e \]

where \( e \sim N_n(0, V) \).

Another way of thinking is the formulation we used in section 2.7 to derive the ML-estimators for the model parameters. This formulation, called ML-formulation, is specified as

\[ y \sim N_n(Xb, V) \]

Thus, we do not explicitly write the error term, but the assumed linear relationship between response and the predictors has been written to the mean of the assumed normal distribution.

These two formulations, LS and ML-formulation, are equivalent, but arise from two different ways of thinking. The lower alternative represents such a way of thinking that is used in generalized linear models. The most essential difference is that the generalized linear model does not have an explicitly written error term. Furthermore, the LS-way of thinking is not necessarily possible for GLM with other distributions than the normal. Thus, the GLM’s are based on thinking the model through the the ML-formulation.

The ML estimation of the linear model, which was presented in section 2.7, is the generalized linear model formulation for normally distributed response. The generalized linear model can be developed also for other types of response in a similar manner. We just write the parameters (or functions of them) as a linear function of predictors, and fit the assumed distribution into the data using the method of maximum likelihood.

With distributions belonging to the exponential family, certain good properties hold for the estimators, and statistical packages generally do not provide tools for estimation of GLM for other distributions than those belonging the exponential family.
5.2 GLM for exponential family

5.2.1 Model formulation

PIVIT! Onko b(j):ss alla miinus edessä vai ei? The generalized linear model for a random variable that follows a distribution of exponential family is (McCulloch and Searle 2001)

\[ y_i \sim \text{indep.} f_{Y_i|\gamma_i}(y_i) \]

\[ f_{Y_i|\gamma_i} = \exp \left( \frac{y_i \gamma_i - b(\gamma_i)}{\tau^2} - c(y_i, \tau) \right) \]  
(5.1)

The first expression states the family of the distribution. It is assumed that the data, \( y_i, i = 1, \ldots, n \), constitute an independent, random sample from a distribution with density \( f_{Y_i} \). The second expression states a general formulation to that density, i.e., the so called canonical formulation of the exponential family, where \( \gamma_i \) and \( \tau \) are the parameters of the distribution. Especially, \( \tau \) is related to the variance of the distribution and \( \gamma_i \) to the mean. Functions \( b \) and \( c \) are assumed to be known. Parameter \( \gamma \) is called the canonical parameter and is related to the mean through

\[ \mu_i = E(Y_i) = b'(\gamma_i) \]

and the variance is

\[ \text{Var}(Y_i) = \tau^2 b''(\gamma_i) \]
\[ = \tau^2 b''(b^{-1}(\mu_i)) \]
\[ = \tau^2 v(\mu_i) \]

In addition to the above assumptions, we assume that a function of the mean is linear in predictors,

\[ g(\mu_i) = x' b \]

which is equivalent to

\[ g(-b'(\gamma_i)) = x' b. \]

Function \( g \) is the applied link function, which is usually selected so that it retains the estimated parameters within its support.

For example, if the expected value of the random variable, \( \mu = -b'(\gamma_i) \) has to be between 0 and 1, the link \( g(\mu) \) should be a monotone function between \( \mu \in [0, 1] \) that gets all possible values between \( ] - \infty, \infty[ \). Such a function is obtained as the inverse cdf (i.e., quantile function) of any continuous distribution. Another typical example restricts the parameter \( \mu = -b'(\gamma_i) \) to be positive. An appropriate link for this case is a monotone function that gets values \( ] - \infty, \infty[ \) for \( \mu \in [0, \infty] \). The idea of the link is,
Table 5.1: Canonical link functions for the most common distributions of the exponential family (adopted from Wikipedia 28.4.2008)

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Name</th>
<th>Link Function</th>
<th>Mean Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Identity</td>
<td>(X\beta = \mu)</td>
<td>(\mu = X\beta)</td>
</tr>
<tr>
<td>Exponential</td>
<td>Inverse</td>
<td>(X\beta = \mu^{-1})</td>
<td>(\mu = (X\beta)^{-1})</td>
</tr>
<tr>
<td>Gamma</td>
<td>Inverse</td>
<td>(X\beta = \mu^{-1})</td>
<td>(\mu = (X\beta)^{-1})</td>
</tr>
<tr>
<td>Inverse Gaussian</td>
<td>Inverse squared</td>
<td>(X\beta = \mu^{-2})</td>
<td>(\mu = (X\beta)^{-1/2})</td>
</tr>
<tr>
<td>Poisson</td>
<td>Log</td>
<td>(X\beta = \ln(\mu))</td>
<td>(\mu = \exp(X\beta))</td>
</tr>
<tr>
<td>Binomial</td>
<td>Logit</td>
<td>(X\beta = \ln\left(\frac{\mu}{1-\mu}\right))</td>
<td>(\mu = \frac{\exp(X\beta)}{1+\exp(X\beta)})</td>
</tr>
<tr>
<td>Multinomial</td>
<td>Logit</td>
<td>(X\beta = \ln\left(\frac{\mu}{1-\mu}\right))</td>
<td>(\mu = \frac{\exp(X\beta)}{1+\exp(X\beta)})</td>
</tr>
</tbody>
</table>

that whatever value the linear predictor \(x'b\) gets within \([-\infty, \infty]\), applying the inverse link \(\mu = g^{-1}(x'b)\) ensures that the estimate of \(\mu\) is always within its range.

In addition, for each density function \(f\), the so called canonical link ensures that \(X'y\) is a sufficient statistic for \(\gamma_i\). The canonical links for some distributions are given in table 5.1.

5.2.2 Logistic regression

In addition to the Linear model under normality, formulation and estimation of which was explained in detail in section 2.7, another common application of the generalized linear model is the logistic regression. It is the best-justified method for data of binary observations, such as success or failure, or presence or absence of a certain feature. Let us define the logit-transformation as

\[
g(\mu) = \text{logit}(\mu) = \log\left(\frac{\mu}{1-\mu}\right)
\]

Where \(\mu\) is the expected value of the binomial variable, and is always between 0 and 1. The inverse logit is obtained by solving the above equation for \(\mu\). We get

\[
\mu = g^{-1}(y) = \frac{1}{1+e^{-y}}.
\]

These functions are illustrated in figure 5.1

In the Logistic regression model, we assume

\[
y_i \sim \text{indep. Bernoulli}(\mu(x_i)) \quad g(\mu) = \text{logit}(\mu) \quad g(\mu) = x_i'b
\]

The estimation and inference follows the procedure outlined for the general model.
5.2. GLM FOR EXPONENTIAL FAMILY

5.2.3 Poisson regression

The Poisson regression is an appropriate model for count data. In the Poisson regression model, we assume

\[ y_i \sim \text{indep. Poisson}(\mu(x_i)) \]

\[ g(\mu_i) = \ln(\mu_i) = x_i' \beta \]

The log link is used to ensure that the expected density, \( \mu_i \), is greater than zero. The estimation and inference follows the procedure outlined for the general model, and details are omitted.

As we remember from Chapter 1, the mean and variance of Poisson distribution are the same. However, a very common situation is that this relationship does not hold. In this case, an additional parameter is introduced that specifies the relation of the variance and expectation. It should be noted that this means that the Poisson distribution is not any more assumed. However, this model is commonly referred as extra-poisson model. In reality, no specific distribution is assumed, but the distribution is approximated by a discrete distribution that allows different mean and variance. Such distribution is, for example, the negative binomial distribution.

**Example 5.1.** In the previous examples, we fitted Poisson GLM to the count data of spruce saplings. The assumption on the distribution was not evaluated, even though we carried out tests etc for the data. In Poisson situation, a crude test on the assumption can be carried out by comparing variances to the mean in discrete classes of the predictor. Thus, we classified the data from 123 plots to 5 classes of hdecid, with each class having equal number of observations. The class means and variances were computed, and are added to the previously shown plot of data in Figure 5.2.
CHAPTER 5. GLM’S AND GLMM’S

We see that the variances are higher than mean for all classes except for the last one, for which mean is slightly higher than the variance. However, the differences are fairly small for three classes. For the rest two classes, variances are about two times as high as the means. This implies that the assumption on Poisson distribution did not hold very well, but the number of seedlings varies more than one could expect. This is related to the spatial pattern of tree locations. We remember that the number of trees within a plot is Poisson distributed if tree locations are independent. Thus, this result shows that this independence is not met. The higher variance than the mean indicates clustering of spruce saplings. The lower-than-the-mean variance would mean regular spatial pattern. Thus, we conclude that our data indicates clustering, and some other assumption than Poisson could be better. This is a very common situation with real-life datasets.

A better assumption would be an "overdispersed Poisson distribution. It is important to recognize that this is a misleading term, as we are actually not any more assuming a Poisson distribution. Instead, we assume an unspecified distribution having two parameters: one for mean, and another for the ratio of variance and mean. As no specific assumption on distribution is not made, the likelihood cannot be written, and MLE:s cannot be computed. However, we can write a quasi-likelihood that is something similar to the likelihood, but uses only information on mean and variance, not on the distribution itself. This would lead to Quasi-likelihood estimation. More information on these approaches can be found e.g., in McCulloch and Searle (2001).

5.2.4 Estimation

The estimation of GLM is based on the maximum likelihood principle. The log likelihood of a random variable that is distributed according to the exponential family is obtained from (5.1) as

$$l = \frac{1}{\tau^2} \sum_{i=1}^{n} [y_i \gamma_i - b(\gamma_i)] - \sum_{i=1}^{n} c(y_i, \tau)$$

It can be shown (see McCulloch and Searle (2001) for details) that

$$\frac{\partial l}{\partial b} = \frac{1}{\tau^2} \sum (y_i - \mu_i) w_i g(\mu_i) x'_i$$

where $w_i = [v(\mu_i) g^2(\mu_i)]^{-1}$, and $v$ is the variance function, which was defined earlier.
By defining $W$ and $\Delta$ as

$$W_{n \times n} = \begin{bmatrix} w_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_n \end{bmatrix} \quad \Delta_{n \times n} = \begin{bmatrix} g(\mu_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & g(\mu_n) \end{bmatrix}$$

we can write the likelihood equation in the matrix form as

$$\frac{\partial l}{\partial b} = \frac{1}{\tau^2} X' W \Delta (y - \mu).$$

Equating this to 0 gives the ML equations as

$$X' W \Delta y = X' W \Delta \mu.$$
CHAPTER 5. GLM’S AND GLMM’S

... the data. Thus, this method is somehow looking for a counterpart of the LS approach in the case of GLM’s. For some distributions, such as Poisson the ML leads to equivalent results to those of MQL with corresponding assumptions on mean-variance relationship. The MQL also has some advantages in the case of overdispersion (McCullough and Searle 2001).

Example 5.2. The number of spruce saplings has been recorded from 123 sample plots. Each of the plots has been taken from different regeneration area. For count data, the Poisson regression would be appropriate methodology. especially, we remember from chapter 1 that if trees are randomly located in the forest, then the number of trees on the plot has the $\text{Poisson}(\lambda | A |)$ distribution. We want to explain the number of spruce saplings per plot using mean height of deciduous trees. We first plot the data to see that the number of spruces decreases as the height of deciduous trees increases (Figure 5.3).

```r
plants2<-read.table("c:/laurim/biometria/plants2.txt",header=TRUE)
plants2<-plants2[!is.na(plants2$hdecid),] \# remove observations with unknown hdecid
plants2$spruces<-plants2$planted+plants2$spruces
plot(plants2$hdecid,plants2$spruces)

The model is fitted using

```R
> glm1 <- glm(spruces ~ hdecid, family=poisson(), data=plants2)
> summary(glm1)
```

Figure 5.3: Plot of the count data and the fitted regression line.
5.2. GLM FOR EXPONENTIAL FAMILY

> summary(glm1)
> summary(glm1)

Call:
  glm(formula = spruces ~ hdecid, family = poisson(), data = plants2)

Deviance Residuals:
  Min       1Q     Median       3Q      Max
  -2.6740  -0.8421  -0.1711   0.5929   4.7531

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.360962   0.135115   10.073  <2e-16 ***
 hdecid      -0.001739   0.001149   -1.513   0.130
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

(Dispersion parameter for poisson family taken to be 1)
  
  Null deviance: 189.85 on 113 degrees of freedom
  Residual deviance: 187.46 on 112 degrees of freedom
  AIC: 503.69

Number of Fisher Scoring iterations: 5

We add the fitted line into the plot of our data

> x<-seq(0,300,1)
> lines(x,exp(cbind(1,x)%*%coef(glm1)))

Example 5.3. To show that GLM is just an application of fitting a distribution function to data using the method of maximum likelihood, we fit the model of previous example using mle package. For this purpose, we first define a function that computes the negative log likelihood as a function of our coefficients. Then we use function mle to find such values for the coefficients that maximize the log likelihood.

> library(stats4)
> nll<-function(b0,b1) {
+  cat(b0," ",b1," 
+  value<-sum(log(dpois(plants2$spruces,exp(b0+b1 *plants2$hdecid))),na.rm=TRUE)
+  cat(value,"\n")
+  value
+ }
>
> solution<-mle(minuslogl=nll,start=list(b0=0.5,b1=0))

0.5 0 317.5801
0.501 0 317.4012
0.499 0 317.7593
0.5 0.001 300.3687
0.5 -0.001 337.6111
179.5457 18621.23 Inf
...
1.380250 -0.002904336 252.3297
1.381250 -0.001904336 249.8583
1.381250 -0.003904336 259.0869
>
> # Estimates, their standard errors, and AIC
> coef(solution)
 b0  b1
 1.381249811 -0.001904336
> sqrt(diag(vcov(solution)))
 b0  b1
0.134270822 0.001140901
> AIC(solution)
[1] 503.7166
> logLik(solution)
'log Lik.' -249.8583 (df=2)
>
> # The same figures from glm-fit
> summary(glm1)$coefficients

Estimate Std. Error z value Pr(>|z|)
Our own code results in slightly different estimates. To check if the differences result from different likelihoods, we compute the likelihood at the solution of glm using our own function nll.

The resulting value is exactly same as the negative log-likelihood of the GLM model. Thus, the differences result from that glm uses a better algorithm for maximization than our general-purpose function mle, which just calls R function optim.

### 5.2.5 Inference and tests

We remember from section 1.5.2, that the asymptotic variance-covariance matrix of a ML-estimate can be obtained as

\[ \text{var}(\hat{\theta}) \approx [I(\theta)]^{-1} \]

where

\[ I(\theta) = -E \left[ \frac{\partial^2 l}{\partial \theta \partial \theta'} \right] \]

In this case, it can be shown that applying the above equations would give the asymptotic variance as (McCulloch and Searle 2001)

\[ \text{var}(\hat{b}) = \tau^2 (X'WX^{-1}) \]

Thus, the asymptotic distribution of the ML-estimate of \( b, \hat{b} \) is Normally distributed with mean \( b \) and variance-covariance matrix \( \tau^2 (X'WX^{-1}) \). However, note that these results are asymptotic, and are not necessarily valid for small samples. However, exact results do not exist for generalized linear models.

To test if the predictors of the model significantly explain the variance in the data, a Likelihood Ratio (LR) test can be formulated. More specifically, assume a null hypothesis

\[ H_0 : \theta = \theta_0. \]

where \( \theta_0 \) is a specified value we assume for \( \theta \) under the null hypothesis. If the null hypothesis is true, the likelihood ratio test statistic

\[ -2 \ln \Lambda = -2 \left[ l(\theta_0) - l(\hat{\theta}) \right] \] (5.2)

follows the \( \chi^2(q) \) distribution with where \( q \) is the number of restrictions.
The parameter vector can also be partitioned by making specified assumptions only for some of the parameters. This can be formulated as a LR test as follows.

1. Fit the full model and save the value of the maximum log likelihood.

2. Fit a restricted model, where the parameters of interest are restricted according to the null hypothesis, and other parameters are estimated using the method of maximum likelihood.

3. Compute the LRT test statistic (5.2).

4. Compare the test statistic to $\chi^2(q)$ distribution, where $q$ is the number of restrictions in the restricted model.

Another test, called Wald test, can be used for the same purpose. According to McCulloch and Searle (2001), it is computationally simpler, but may lead to worse approximations of the p-value with moderate or small sample sizes and extreme deviations. To test two models against each other, a Likelihood Ratio test can be formulated.

**Example 5.4.** We want to test if dropping the only predictor from our model would cause no remarkable reduction of fit. The restricted model is fitted using

```r
> glm2 <- glm(spruces ~ 1, family=poisson(), data=plants2)
> summary(glm2)
```

The test statistic is computed as

```r
> chiobs <- -2*(logLik(glm2)-logLik(glm1))
> chiobs
[1] 2.392032
```

The same statistics is obtained also using `anova`

```r
> anova(glm2,glm1)
Analysis of Deviance Table

Model 1: spruces ~ 1
Model 2: spruces ~ hdecid
Resid. Df Resid. Dev Df Deviance
1 113 189.853
2 112 187.461 1 2.392
```
However, anova does not perform the test for glm, so we need to do it by ourselves. Comparison of the test statistic to $\chi^2$ distribution with one degree of freedom gives

\[
> 1 - \text{pchisq(chiobs, 1)}
\]

[1] 0.121955

The p-value of 0.12 indicates that the trend seen in the number of spruce saplings could be a result of chance, as obtaining such a trend or stronger would have a probability of 0.12 if null hypothesis is true. The p-value of the coefficient of glm1 would have led to the same inference, even though the p-values differ slightly.

### 5.3 Weibull regression

The Weibull distribution is not a member of the exponential family. However, as an approach that is very similar to the GLM was presented for modeling diameter distributions with Weibull distribution by Cao (2004), I will present such GLM-type model here.

Assume we have observed tree diameters from several stands, and we want to model the stand-specific diameter distribution using the Weibull distribution. Furthermore, we want to relate the Weibull parameters to some stand-specific variables, such as stand age, or mean diameter. The ultimate aim is to fit a model that could be used for predicting the diameter distribution for a stand where only the utilized stand-specific predictors are known. This approach is commonly referred as the parameter prediction method (PPM) in the forestry literature.

A commonly used approach for such a modeling task is to first fit a Weibull distribution to the measured diameters of each stand using the method of maximum likelihood. After this first step, a dataset is obtained where each stand is one observation, and Weibull parameters and values of the potential stand-specific predictors are known for each stand. In the second step, the ML-estimates are modeled using the stand-specific predictors for each stand to estimate general relationships between the Weibull parameters and stand characteristics.

The GLM-type approach of Cao (2004) does these two steps at once, which is theoretically better justified and may also result to better fit. The assumed model for
the diameter of tree \( i, i = 1, \ldots, n \) is stated as follows

\[
y_i \sim \text{indep.} \text{Weibull}(\alpha_i, \beta_i)
\]

\[
g_\alpha(\alpha_i) = \ln(\alpha_i) = x'_\alpha b_\alpha
\]

\[
g_\beta(\beta_i) = \ln(\beta_i) = x'_\beta b_\beta
\]

Thus, we assume that tree diameter follows the \textit{Weibull}(\( \alpha, \beta \)) distribution. Furthermore, we use the log link for both parameters, as they are restricted to be greater than zero. The logarithmic parameters are assumed to be linear functions of the predictors, as specified in the last two equations. The predictors and coefficients may be different for the shape (\( \alpha \)) and scale (\( \beta \)) parameters, that is why indices \( \alpha \) and \( \beta \) are in vectors \( x \) and \( b \).

This specification is based on a similar way of thinking as is the GLM. However, there are some differences. First, Weibull distribution is not a member of the exponential family, so the results based on this assumption do not hold. Second, we are not modeling the mean of the distribution, but the parameters directly. Third, two parameters are written as a function of predictors, instead of having only the mean to be modeled. However, if one manages to fit the model, the asymptotic properties of ML-estimates (consistency, unbiasedness, efficiency and normality) hold. These properties are, however, only large sample properties, and may be badly violated for small samples.

**Example 5.5.** Dataset spati includes 10255 measured tree diameters from 66 plots in North Carelia. To relate the diameter distribution parameters on stand characteristics, we first use the well-known PPM approach (Parameter Prediction Method). In that method, the two-parameter Weibull function is first fitted int the data of each plot. Then the obtained estimates are saved into a plot-specific data, and the estimates are related to stand characteristics using a linear model. We omit all details on model diagnostics and assumptions made, as the aim is just to demonstrate the glm approach.

The following code was used to fit Weibull for each plot using ML, and to save the estimated into data frame plots, which has 66 rows, one for each plot. Then we plot the selected stand variables and parameter estimates to study the relationships between stand variables and distribution parameters (Figure 5.4).

```R
> library(stats4)
>
> # 2 parameter Weibull -logL
```
Figure 5.4: Plot of stand-specific Weibull-parameters on stand characteristics.
5.3. WEIBULL REGRESSION

> nLLweibull<-function(x, shape=5, scale=20) {
+ -sum(dweibull(x, shape=shape, scale=scale, log=TRUE))
+ }

> # Fits two-parameter weibull distribution to tree diameter data using MLE starting from values 5 and 20
> # for shape and scale, respectively
> fitw2<-function(d) {
+ est<-mle(function(shape=5, scale=20) nLLweibull(d, shape, scale))
+ if (class(est)="try-error") list(par=rep(NA,2), neg2LL=NA, conv=NA)
+ else list(par=coef(est), neg2LL=2*attributes(est)$min, conv=attributes(est)$details$convergence)
+ }

> # Fit weibull for each plot
> for (i in 1:dim(plots)[1]) {
+ d<-spati$d[spati$plot==plots$plot[i]]
+ weibullfit<-fitw2(d)
+ plots[i,c("shape","scale")]<-weibullfit$par
+ }

> There were 50 or more warnings (use warnings() to see the first 50)
>
> plot(plots[,-1]) # do not include the first column

The plot shows no correlation between shape and stand variables. The scale parameter is quite highly correlated with mean diameter and mean height. Based on these observations, we use a constant value for shape, and relate logarithmic scale on $D_g$ and $H_g$. The logarithmic scale is used for compatibility with the used Weibull regression with log link. The relationships are clearly not linear, but we assume linearity for simplicity. The parameter estimates for the PPM models are used below.

> lmsc<-lm(log(scale) ˜ Dg+ Hg, data=plots)
> lmsh<-lm(log(shape) ˜ 1, data=plots)
> coef(summary(lmsh))

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|---------|
| (Intercept) | 1.067284 | 0.04159837 | 25.65688 | 1.013757e-35 |

> coef(summary(lmsc))

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|---------|
| (Intercept) | 1.64465659 | 0.055449860 | 29.660248 | 1.002560e-38 |
| Dg | 0.02537290 | 0.008300818 | 3.056674 | 3.281326e-03 |
| Hg | 0.03956507 | 0.010854219 | 3.645133 | 5.428510e-04 |

Next, we formulate the Weibull GLM, where the ML fit and model fitting are carried out in one and the same stage. The coefficients from the PPM model are used as initial estimates of the parameters.

> coef(lmsh)->csh
> coef(lmsc)->csc
> names(csc)<-names(csh)<-c() # Having names in the vector, mle tries to use them as parameter names and gets lost.

> # Refit using glm-type approach
>
> nLL<-function(b0sh=csh, b0sc=csc[1], b1sc=csc[2], b2sc=csc[3]) {
+ shape<-exp(b0sh)
+ scale<-exp(b0sc+b1sc*spati$Dg+b2sc*spati$Hg)
+ -sum(dweibull(spati$d, shape=shape, scale=scale, log=TRUE))
+ }

> weibull.glm<-mle(minuslogl=nLL)

Warning messages:
1: In dweibull(x, shape, scale, log) : NaNs produced
2: In dweibull(x, shape, scale, log) : NaNs produced
3: In dweibull(x, shape, scale, log) : NaNs produced
4: In dweibull(x, shape, scale, log) : NaNs produced
5: In dweibull(x, shape, scale, log) : NaNs produced
6: In dweibull(x, shape, scale, log) : NaNs produced
7: In dweibull(x, shape, scale, log) : NaNs produced
8: In dweibull(x, shape, scale, log) : NaNs produced
9: In dweibull(x, shape, scale, log) : NaNs produced

> summary(weibull.glm)
Maximum likelihood estimation

Call:
mle(minuslogl = nLL)

Coefficients:
Estimate Std. Error
b0sh 0.94660490 0.007817644
b0sc 1.60010673 0.011130143
b1sc 0.04103064 0.001823252
b2sc 0.02342298 0.002213285

-2 log L: 60985.8

The estimates from GLM approach are quite different from those of PPM, but are of the same order of magnitude. Cao (2004) reported that the estimates from GLM approach lead to highly more accurate prediction of diameter distribution than the traditional PPM estimates. The improvement is due to that the GLM does the whole procedure in a single stage, whereas in PPM is carried in two stages, and parameter estimates from the first stage include estimation errors.

5.4 Generalized linear mixed models

As one might guess, the generalized linear mixed model is a generalized linear model with random effects. Such a model is appropriate for modeling hierarchical datasets, as was the linear mixed-effects model, too. The random effects are incorporated into the linear part of the model. Let $c$ include the random effects of the model. The generalized linear mixed model is specified as,

$$
y_i | c \sim \text{indep. } f_{Y_i \mid u}(y_i | c)
$$

$$
f_{Y_i | c} = \exp \left( \frac{y_i \gamma_i - b(\gamma_i)}{\tau^2} - c(y_i, \tau) \right)
$$

$$
g(\mu_i) = x'_i b + z'c
$$

The model specification is very similar to the GLM, but now we have random effects $c$ added to the linear predictor. The random effects are assumed to have a multinormal distribution, and effects of different groups in the data.

It is important to note that the generalized linear model assumes distribution of the exponential family for the conditional observations. Thus, for example, we might assume that tree species within a given stand is bernoulli distributed, or the number of trees of a given species from plots of one stand follows a Poisson distribution. However, the marginal distribution of the data may not be of that family. In linear mixed-effects model, where normality was assumed for the response, the general results on the normal distribution state that all conditional and marginal distributions of a multinormal distribution is normal. Thus, for LMM, marginal distributions are also normal. However, these result do not hold generally, and marginal distribution usually cannot be
5.4. GENERALIZED LINEAR MIXED MODELS

specified. However, the marginal expectation, variance, and covariance can be computed. The general formulas for marginal expectation, variance, and covariance are as follows.

\[
E(y_i) = E \left[ g^{-1}(x'_i b + z'_i c) \right] \\
\text{var}(y_i) = \text{var} \left[ g^{-1}(x'_i b + z'_i c) \right] + E(\tau^2 \nu (g^{-1}[x'_i b + z'_i c])) \\
\text{cov}(y_i, y_j) = \text{cov} \left[ g^{-1}(x'_i b + z'_i c), g^{-1}(x'_j b + z'_j c) \right]
\]

As one can see, the values of these depend on the utilized link function, and on the assumed distribution of the data. Thus, no easily applicable formulas can be presented for these moments of the marginal distribution. In general, random effects cause dependence among the observations so, that observations that share a random effect are correlated. However, computing the correlation is not that simple as it was with LMM.

The estimation of the parameters of a GLMM is based on the principle of maximum likelihood. Each possible value of the random effect vector \(c\) would give different value for the likelihood. Thus, we can think the value of the likelihood as a transformation of the random variable, \(c\). The expected value of the likelihood is then the likelihood of the data. The likelihood becomes

\[
L = \int_c \prod_i f_{Y_i|e}(y_i|c) f_C(c) dc
\]

where \(f_{Y_i|e}(y_i|c)\) is the conditional density of the data (e.g., Poisson density), and \(f_C(c)\) is the \(q\)-dimensional joint distribution of random effects \(c_{q \times 1}\).

**Example 5.6.** Assume that \(n_i\) sample plots of equal area have been measured from stand \(i, i = 1, \ldots, m\). Let \(y_{ij}\) denote the number of Scots pines observed on the \(j\) th plot of stand \(i\). A model for the number of Scots pines within a plot is specifies as

\[
y_{ij} | c \sim \text{indep.Poisson}(\mu_{ij}) \\
\ln \mu_{ij} = x'_i b + c_i \\
c_i \sim \text{i.i.d.} \mathcal{N}(0, \sigma^2)
\]

This model assumes only a random constant for each stand, so that the logarithmic density (and variance, as they are equal with Poisson distribution), behaves in a similar way with respect to the predictors, but the level may change.
The likelihood can be written as
\[
l = \ln \left( \prod_{i=1}^{m} \int_{-\infty}^{\infty} \mu_{ij}^{y_{ij}} e^{-\mu_{ij} y_{ij}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} c_{ij}^2} dc_{ij} \right)
\]
\[
= y'Xb - \sum_{i,j} \ln y_{ij}! + \sum_{i} \ln \int_{-\infty}^{\infty} \exp \left[ y_{i} u_{i} - \sum_{j} e^{x'_{ij} b_{ij}} u_{i} \right] \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} c_{ij}^2} dc_{ij}
\]

This integral cannot be expressed in closed form. Thus, we cannot express the likelihood equations in closed form, and all the computations need to be carried out numerically. However, numerical methods are so well developed nowadays that modern computers are able to find estimators for GLMMs.

The inference and tests on GLMMs are all based on the asymptotic properties of the ML-estimator. Thus, the inference and diagnostics on GLMs applies also for GLMM:s.

The prediction of random effects with GLMMs is not as simple as for linear models.

**Example 5.7.** A more extensive data of spruce saplings can be found in file `plants.txt`. The data includes measurements from 1926 plots from 123 regeneration areas within one county. The data is read using the following code.

```r
> plants<-read.table("c:/laurim/biometria/plants.txt",header=TRUE)
> plants<-plants[!is.na(plants$hdecid),]
> plants$spruces<-plants$planted+plants$spruces
> hist(plants$spruces)
```

A good starting point for a model for spruce sapling counts is a Poisson mixed model with log link and random constant. We fit the model using the default PQL method for approximating the likelihood.

```r
> glmm1.PQL<-lmer(spruces ˜ (1|stand)+hdecid, family=poisson(), data=plants)
```

```
Generalized linear mixed model fit using PQL
Formula: spruces ˜ (1 | stand) + hdecid
Data: plants
Family: poisson(log link)
AIC BIC logLik deviance
2373 2390 -1184 2367
Random effects:
Groups   Name     Variance Std.Dev.
stand (Intercept) 0.065014 0.25498
Number of obs: 1926, groups: stand, 123

Estimated scale (compared to 1) 1.058734

Fixed effects:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.171354  0.046262  25.320  <2e-16 ***

hdecid     -0.000123  0.000346   -0.355     0.723

---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Correlation of Fixed Effects:
 (Intr) hdecid
hdecid    -0.820
```

Hdecid seems to be insignificant predictor, and could probably be dropped. Thus, we fit a restricted model and make a likelihood ratio test.
5.4. GENERALIZED LINEAR MIXED MODELS

Figure 5.5: Histogram of the number of spruce saplings in dataset plants.

```r
> glmm2.PQL <- lmer(spruces ~ (1|stand), family=poisson(), data=plants, method="PQL")
> anova(glmm2.PQL, glmm1.PQL)
Data: plants
Models:
glmm2.PQL: spruces ~ (1 | stand)
glmm1.PQL: spruces ~ (1 | stand) + hdecid
Df  AIC   BIC logLik Chisq Chi Df Pr(>Chisq)
glmm2.PQL 2 2371.5 2382.7  -1183.8
glmm1.PQL 3 2373.4 2390.1  -1183.7  0.1622 1 0.6871

> summary(glmm2.PQL)
Generalized linear mixed model fit using PQL
Formula: spruces ~ (1 | stand)
Data: plants
Family: poisson(log link)
AIC  BIC logLik deviance
2372 2383  -1184  2368
Random effects:
 Groups Name Variance Std.Dev.
 stand (Intercept) 0.064826 0.25461
number of obs: 1926, groups: stand, 123
Estimated scale (compare to 1 ) 1.058644

Fixed effects:
 Estimate Std. Error z value Pr(> |z|)
(Intercept) 1.15792 0.02644 43.8 <2e-16 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

The p-value from anova (0.6871) leads to rejection of glmm1, and we conclude that there is no evidence on that the number of spruce samplings would depend on the height of deciduous trees in this data. The next question could be whether the random effects are significant. The summary above showed, that the estimated between-stand
variation in the logarithmic number of saplings is 0.25, which shows to be quite high. A test on this needs a restricted model, i.e., a model without random effects. Such a model is fitted using glm, and the likelihood ratio test is then carried out.

```r
> glm1<-glm(spruces ~ 1, family=poisson(), data=plants)
> chiobs<-2*(logLik(glm1)-logLik(glmm2.PQL))
> chiobs
[1] 5738.658
> 1-pchisq(chiobs,1)
[1] 0
```

The value of test statistic is extremely high, indicating significant variation in stand density among stands. Thus, we accept the model with random effect for stand and no fixed effects as the final model.

The estimation methods for GLMM:s is under development. In GLMM:s, the problem is that the likelihood needs to be approximated numerically. Different approximation methods work in different situations, and that is why it is important to test different methods and even different implementations of them to see whether the estimates are the true maximums the likelihood. In R, there are three alternatives for GLMM:s: PQL (Penalized Quasi Likelihood), Laplace, and AGQ (Adaptive Gaussian Quadrature). However, AGQ is available only in lmer2, which is a development version of lmer. The authors of lmer write of the estimation methods as follows. "The PQL method is fastest but least accurate. The Laplace method is intermediate in speed and accuracy. The AGQ method is the most accurate but can be considerably slower than the others. However, it appears that AGQ does not work for all situations.

**Example 5.8.** The following code fits the model of previous example using Laplace method.

```r
> glmm2.Laplace<-lmer(spruces ~ (1|stand), family=poisson(), data=plants, method="Laplace")
> glmm2.Laplace
Generalized linear mixed model fit using Laplace
Formula: spruces ~ (1 | stand)
Data: plants
Family: poisson(log link)
AIC BIC logLik deviance
2371 2382 -1183 2367
Random effects:
Groups Name Variance Std.Dev.
stand (Intercept) 0.0758 0.27532
number of obs: 1926, groups: stand, 123
Estimated scale (compare to 1 ) 1.055265
Fixed effects:
(Intercept) 1.1482 0.0281 40.86 <2e-16 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

There are slight differences in the estimates of coefficients. According to the suggestion of the authors of lmer package, I would select the model estimated using Laplace instead of the one estimated using PQL.

The AQL method does not work for this data.
5.5 Exercises

1. Model the dependence of the number of planted spruces on the covariates of dataset plants2 using a GLM.

2. Tomicus insects eat the shoots of Scots pine trees, and the eaten shoots can be found from the forest floor. The insects seem to favor places that are close to small openings within a forested area. To study the amount of eaten shoots, a total of 14 one-meter wide and 50-meter long (12 transects) or 90-meter long (2 transects) transects, starting from an opening, were established to three different regions in Eastern Finland. The number of eaten shoots in the forest floor was counted for each 1m by 1m square within the transect, leading to 50 observations for the shorter transects and 90 for the longer transects. The data, provided by Atte Komonen (Komonen Kouki 2008; Do restoration fellings... For. Ecol and Manag. Online first), are given in file tomicus.txt. The data includes variables: area - code for study area 1, 2 or 3; transect - the transect id; distance - distance from the edge of opening, meters; ntom - observed number of eaten pine shoots in the 1 by 1 meter square. The aim is to model the expected value of eaten shoots on the distance from the edge of the opening.

   (a) Plot the number of eaten shoots against distance for each plot.

   (b) For each plot, divide the data into 5 groups according to the distance, and compute mean and variance of ntom for each group. Add points for the means and variances into each plot. Do you see signs of over- or underdispersion?

   (c) Fit a Poisson GLM separately into the data of each transect using function glm. Add a line that shows the fitted curve.

3. Using data tomicus,

   (a) Fit a Poisson GLMM into the whole data, assuming that both constant and coefficient are random. The command for that purpose is model1:<br>lmer(ntom (distance—transect),family=poisson(),data=tomicus)

   (b) Fit a restricted model with random effect only for constant. Use: model2:<br>lmer(ntom (1—transect)+distance, family=poisson(),data=tomicus)
(c) Using anova(model1, model2), test whether having both parameters as random would lead to better model than having only a random constant.

4. Model the dependence of the number of planted spruces on the covariates of dataset plants using a GLMM.

5. Do some modelling using GLM or GLMM in your own dataset.
Chapter 6

Model systems

6.1 Types of model systems

Models systems arise from a need to simultaneously model several responses. Typically, methods developed for simultaneous model estimation are needed when a model system, comprised of several individual models, is estimated. In fitting the model system, it may not be enough to model the behavior of individual models of the system. Instead, a model system may be needed that realistically describes the system as a whole. For this reason, the interrelationships between models need to be taken into account.

Model systems have been used a lot of in econometrics for modeling the markets. For example, the so called Kleins small macroeconomic model includes equations for consumption, demand and investment (Greene 1997). The individual equations of this system are directly related, for example the demand is a function of consumption and investment. In forestry, model systems are needed, for example, in developing a forest simulator. Such a simulator may include models for stand structure, tree growth, mortality, and silvicultural operations. The stand structure has affects stand growth, which in turn determines the timing of the silvicultural practice, e.g., thinning. Furthermore, the thinning has an effect on the stand structure etc. Thus, different model components should formulate a model system that describes the development of a forest stands in a realistic way.

The model systems may be directly related, seemingly unrelated, or unrelated. In directly related models, response of one model may be the predictor in another model. In seemingly unrelated model, direct relationships do not exist between the individual models. However, the models may be related in that the residuals are correlated. In unrelated models, no direct relationship exists between the models, and the residuals are uncorrelated.
6.2 Estimation of directly related models: 2SLS

6.2.1 Illustration through height and volume models

In directly related models, a predictor of one component model is the response of another component. For example, we may have a model system for height \( h_i \) and volume \( v_i \) for tree \( i \):

\[
\ln h_i = a_h + b_h \ln d_i + e_{hi} \quad (6.1)
\]

\[
\ln v_i = a_v + b_v \ln d_i + c_v \ln h_i + e_{vi} \quad (6.2)
\]

where \( d \) is tree diameter, \( a_h, b_h, a_v, b_v, \) and \( c_v \) are parameters to be estimated, and \( e_{hi} \) and \( e_{vi} \) are the residuals of the models, which are assumed to be uncorrelated and have expectation 0.

The two models of the above system are directly related, because one of the predictors of the volume model, the tree height, is the response of the height model. Assume that we have a model fitting data including observed diameters, heights, and volumes for trees from one stand. We could fit the models separately, i.e., first fit the model for tree height and then for tree volume. This would be a rather good approach in many situations.

Problems arise if we use the model for prediction for trees with unknown height. In such case we would first predict tree height using model (6.1), and further write the predicted height into the lower model to predict tree volume. However the predicted height is a different random variable than the true tree height. Thus, we are replacing a required predictor with something else, which is an unbiased estimator of the random variable we would need.

An estimation method that accounts for this problem is obtained by replacing the true height in the volume model with the predicted height also in the estimation stage. Then using the predicted height utilized in the application stage would be exactly the same random variable which was used in the model fitting stage. Thus, the estimation would be carried out in two stages:

1. Fit the height model (6.1) to the data, and save the fitted values of the model

2. Fit the volume model (6.2) to the data, but replace the height in the RHS by the predicted height from step 1.

This two-stage approach yields a model system that could be used in the application described below. Note that the volume model fitted in the second stage usually fits much worse to the data than a model with the true height as a predictor would fit. This is just a result of the fact that we are able to utilize only that part of the variation in tree
height that could be explained with variation in tree diameter. Thus, the fitted volume model is actually based on tree diameter only, and it does not account for the variation in tree height. However, this is what we actually need, if we do not have information on tree height available in applications.

### 6.2.2 General formulation

In general case, the variables appearing in a model system can be classified into two classes: endogenous and exogenous. The exogenous variables are called instrumental variables, and they are the variables that bring the information into the system. In the previous example, the only instrumental variable was the tree diameter. The instrumental variables are those variables we will actually have known when the model system is applied in practice.

The direct relationships among the components of a model system can be taken into account through a two-stage least squares approach. In that approach, models for all the predictors are first fitted using the instruments as predictors, and the fitted values of these models are saved. In the second stage, the actual values of the predictors are replaced with the predictions from the first stage, to estimate the coefficients of the model system.

Let $X$ be the model matrix of the model component to be estimated, and let $Z$ be the matrix of instruments. The two-stage least squares approach can be written as:

1. $\hat{X} = Z(Z'Z)^{-1}Z'X$
2. $\hat{B}_{IV} = (\hat{X}'\hat{X})^{-1}\hat{X}'Y$

Note that in the case of unrelated models, this approach would lead to OLS as a special case. This is because $Z$ and $X$ would be identical, giving $\hat{X} = X$.

### 6.3 Estimation of unrelated models with correlated residuals: SUR

#### 6.3.1 Model formulation

The seemingly unrelated models do not have such a direct relationship as the directly related models do. However, they are related to each other because the residuals are correlated. Let us consider an example where tree height and volume are both modeled on tree diameter only

\[
\ln h_i = a_h + b_h \ln d_i + e_{hi} \quad (6.3)
\]
\[
\ln v_i = a_v + b_v \ln d_i + e_{vi} \quad (6.4)
\]
where the notations are the same as we used in models (6.1) and (6.2).

If the models are estimated separately, we would assume \( e_{hi} \) are independent, identically distributed random variables with constant variance, and the same assumptions would also be done for \( e_{vi} \). However, the residuals for different models may be correlated for same individuals. For example, it might be realistic to assume that if height is overestimated for a given tree \( i \), then also the volume might be overestimated. The separate estimation does not provide information on this kind of correlation. In a SUR model, we allow correlation among responses by specifying \( \text{cov}(e_{hi}, e_{vi}) \).

The SUR model can also be expressed in a matrix form. Assume that we have \( m \) individual models,

\[
\begin{align*}
y_1 &= X_1 b_1 + e_1 \\
y_2 &= X_2 b_1 + e_2 \\
&\vdots \\
y_m &= X_m b_1 + e_m
\end{align*}
\]

By defining

\[
\begin{align*}
y &= \begin{bmatrix} y_1 \\
y_2 \\
&\vdots \\
y_m \end{bmatrix} \\
X &= \begin{bmatrix} X_1 & 0 & \cdots & 0 \\
&0 & X_2 & \cdots & 0 \\
&\vdots & \vdots & \ddots & \vdots \\
&0 & 0 & \cdots & X_m \end{bmatrix} \\
e &= \begin{bmatrix} e_1 \\
e_2 \\
&\vdots \\
e_m \end{bmatrix} \\
b &= \begin{bmatrix} e_1 \\
e_2 \\
&\vdots \\
e_m \end{bmatrix}
\end{align*}
\]

we can write the model system in the form of linear model (2.5) as

\[
y = X b + e
\]

Thus, this model is just a special case of the (very!) general matrix formulation of the linear model. In the above formulation, \( X \) is a block-diagonal matrix. Matrix \( D = \text{var}(e) \) has quite a peculiar structure

\[
D = \begin{bmatrix} 
\sigma_1^2 I & \varphi_{12} I & \cdots & \varphi_{1 m} I \\
& \sigma_2^2 I & \cdots & \varphi_{2 m} I \\
& \vdots & \ddots & \vdots \\
& \varphi_{1 m} I & \varphi_{2 m} I & \cdots & \sigma_m^2 I 
\end{bmatrix}
\]

If the observations are ordered so that observations of different response of one observation are one after another, then the structure of \( X \) would change to something peculiar, but matrix \( D \) would then be block-diagonal.
6.3. SEEMINGLY UNRELATED REGRESSION

6.3.2 Estimation

As the above specification showed that the SUR model is just a special case of the LM, the estimation methods do not change from those of the Linear model. Thus, the estimation involves first estimating the parameters specifying matrix $D$, and then estimating the model coefficients using GLS.

Matrix $D$ is specified with response-specific variances of residual error, and the between-response covariances. For estimating these parameters, two alternative methods are available. The first one, which we call Zellner's method, is very easy to implement even manually. In that method, the individual models are first fitted to the data separately using OLS. The variance-covariance matrix of residuals is then estimated using the residuals from these fits. Another approach is to assume multinormality of residuals and estimate the parameters of matrix $D$ by using the method of maximum likelihood. In both methods, the estimated variances and covariances are written to matrix $D$, and the parameter vector $b$ is fitted using GLS.

6.3.3 Prediction

Prediction from a SUR model is very similar to prediction from any other linear model. We just write the values of the predictors to the model and make prediction. The prediction intervals are calculated in very similar manner than with Linear models.

However, an interesting application arises if the value of one or more responses of the simultaneous model system has been observed, and other responses are being predicted for that individual. In that case, the estimated between-model correlation can be used to carry information from one model to another. The prediction is a straightforward application of The linear predictor of section 1.6. For example, assume that the response of model 1 has been observed, and others are to be predicted. We just write the observed value of $y_1$ into vector $h_2$, and the other variables for the same sampling unit into vector $h_1$. The predictions is then direct application of the ideas presented in Section 1.6.

6.3.4 Why simultaneous estimation

The across-model correlation may improve the analysis because

1. The fixed parameters of the model may be better estimated.
2. Prediction of the models will be more efficient, if the response of one of the component models have been observed.
3. A more realistic simulations could be obtained from the models system.
The first item of the list is maybe the most well-known result of simultaneous estimation. However, it seldom causes any big improvement to the efficiency. SUR leads to more efficient estimation, if correlations among the residuals exist, and if the predictors of the component models are not the same. However, if a predictor is dropped from one component model because of its statistical insignificance, and the predictors are otherwise the same, the gain of SUR is negligible. I would see simultaneous estimation as an alternative to using all the predictors in all the component models. This being the case, it seldom causes any remarkable gain in the efficiency of predictors. Thus, I would regard the two later points as more important from a practical point of view.

However, even though the information on between-model correlation would not improve the model at all, utilizing a known correlation would lead to much better and more realistic results in prediction or simulation. However, in these cases it is not necessary to fit the model simultaneously, but it may be enough just to estimate the between model covariances using the residuals of individual models. Furthermore, such an approach could lead to easy implementation of more sophisticated models for the interdependencies than just a constant covariance (Lappi 2006).

The prediction from SUR model may be highly more efficient than from separately estimated models if the residuals of the models are correlated, and if observed response of one component model can be used in prediction. For example, assume that simultaneously fitted models (6.3) and (6.4) are available, and they are used for prediction for a tree with known height and diameter. In this case, we can compute the realized residual of the height model for that tree. Furthermore, the between-model correlation among residuals can be utilized to predict the residual of the volume model for that tree, which would lead to significantly better prediction of total volume than using tree diameter alone.

Simulation from a model system is a demanding task, where omitting important interrelationships between models can lead to drastic results. Thus, knowledge of the interrelationships between-models is extremely important in simulation. The simulation very often faces to a situation where an assumed linear, constant correlation is not sufficiently realistic and detailed assumption, and more is needed.

### 6.4 Estimation of directly related models with correlated residuals: 3SLS

The three-stage least squares is an approach combining 2SLS and SUR approaches. In 3SLS approach, The initial models are first estimated using the instrumental variable method, i.e., using 2SLS. Residuals of the estimated 2SLS models are used to estimate
matrix $D$ for the whole models system. This matrix includes also non-zero cross-model covariances. Finally, the final model is estimated by GLS, using matrix $\hat{D}$ in the estimation.

### 6.5 Simultaneous mixed models

Simultaneous mixed models are obtained, if linear mixed models are fitted for several responses at the same time. Assume that we have $m$ individual mixed models,

$$
\begin{align*}
y_1 &= X_1b_1 + Z_1c_1 + e_1 \\
y_2 &= X_2b_1 + Z_2c_2 + e_2 \\
\vdots &= \vdots \\
y_m &= X_mb_1 + Z_mc_m + e_m
\end{align*}
$$

where $\text{cov}(e_i, e_j)$ are nonzero. This means that the residuals of the individual observations are correlated. If only this kind of correlation is assumed, then an easy way to implement the model is to write all the responses in the same column of the data, and add an additional categorical variable that specifies which response is used at that row. In model fitting, the categorical variable indicating the response would be used to specify the innermost level of grouping for the data.

In addition, we may assume also the random effects to be correlated, i.e., we would allow $\text{cov}(e_i, e_j)$ be nonzero. Unfortunately, such models are not that easy to estimate with R. One could surely rely on estimating the models separately, and then computing the cross-model covariances using the residuals and predicted random effects. However, SAS has quite good capabilities for fitting such models.

An application of a simultaneous mixed model was presented by Lappi (1991), and this model was further demonstrated in Lappi et al. (2006). The following pages present an example from Lappi et al. (2006).
Example 6.4. The multivariate case

Lappi (1991) constructed the following multivariate model for the logarithmic height and logarithmic volume of tree $i$ in stand $k$ from stem analysis data (Laasasenaho 1982):

$$\ln H_i = 3.410 - 18.58 \frac{1}{D_{ki}} + a_{0ki} - a_{1ki} \frac{1}{D_{ki}} + e_{ki} \quad \text{and}$$

$$\ln V_i = 2.704 - 48.93 \frac{1}{D_{ki}} + 1.387 \ln D_{ki} + c_{0ki} - c_{1ki} \frac{1}{D_{ki}} + u_{ki},$$

where $D_{ki}$ is DBH+7 cm, parameters $a_{0ki}$, $a_{1ki}$, $c_{0ki}$ and $c_{1ki}$ are stand-specific random parameters and $e_{ki}$ and $u_{ki}$ are residuals with estimated variances of $\text{var}(e_{ki})=0.01113$, $\text{var}(u_{ki})=0.01540$ and covariance $\text{cov}(e_{ki}, u_{ki})=0.01040$. Let us write the random parameters as vectors $\mathbf{a}_k = (a_{0ki}, a_{1ki})'$, $\mathbf{c}_k = (c_{0ki}, c_{1ki})'$ and
define \( b_k = (a_k', c_k')' \). The estimated dispersion matrix of \( b_k \) is (Lappi 1991)

\[
\text{var}(b_k) = D = \begin{bmatrix}
0.04739 & -0.3887 \\
-0.3887 & 20.64 \\
0.05082 & -0.6036 \\
-0.4772 & 24.88 \\
\end{bmatrix}
\]

The last two parts define a partition of matrix \( D \) that is needed in the following calculations. The measured height of a sample tree will be used below to predict the random parameters of the volume function. Assume that two sample trees of diameters 20 and 30 cm and heights 20 and 26 m have been measured. The measured heights follow the model

\[
y_k = \mu_k + Z a_k + e_k,
\]

where vector \( y_k \) includes the measured logarithmic heights, \( y_k = \begin{bmatrix} \ln 20 \\ \ln 26 \end{bmatrix} = \begin{bmatrix} 3.00 \\ 3.26 \end{bmatrix} \), and \( \mu \) their expectations, which are obtained using the first two terms of the height model as \( \mu = \begin{bmatrix} 2.72 \\ 2.91 \end{bmatrix} \). Matrix \( Z \) is the design matrix of the random part, i.e.,

\[
Z = \begin{bmatrix}
1 & \frac{1}{20+7} \\
1 & \frac{1}{30+7} \\
\end{bmatrix},
\]

and \( a_k \) and \( e_k \) are unknown vectors of random parameters and random residuals with variances \( \text{var}(a_k) = D_1 \) and \( \text{var}(e_k) = R = 0.01113 \cdot I \). Using the height and volume models, equation (6.1) can be written as

\[
\begin{bmatrix}
b_k \\
y_k \\
\end{bmatrix} \sim \begin{bmatrix} 0 \\ \mu_k \\ \mu_k \\
\end{bmatrix} \cdot \begin{bmatrix} D & CZ' \\ ZC' & ZDZ' + R \\ \end{bmatrix}
\]

and the BLUP of \( b_k \) is (Equation 6.2)

\[
\hat{b}_k = CZ' (ZDZ' + R)^{-1} (y_k - \mu) = \begin{bmatrix} 0.244 \\ 0.985 \\ 0.230 \\ 1.131 \end{bmatrix}
\]

i.e. the predicted random parameters are \( a_{0k} = 0.244 \), \( a_{1k} = 0.985 \), \( c_{0k} = 0.230 \) and \( c_{1k} = 1.131 \). The predicted logarithmic heights and volumes are obtained by writing these estimates into the height and volume models.
In order to arrive at unbiased predictions of volumes and heights, half of the prediction variance was added to the predicted logarithmic heights and volumes before applying the exponential transformation. The prediction variance of random parameters was first calculated to be

$$\text{var}(\hat{b}_i - b_i) = D - ZC'(ZD, Z' + R)^{-1} CZ' = \begin{pmatrix} 0.0212 & -0.536 & 0.0266 & -0.648 \\ -0.536 & 17.6 & -0.716 & 21.3 \\ 0.0266 & -0.716 & 0.0372 & -0.918 \\ -0.648 & 21.3 & -0.918 & 26.8 \end{pmatrix}$$

Ignoring the estimation errors in the fixed parameters, the prediction variances of the predicted logarithmic heights were then obtained from the diagonal of

$$\text{var}(\hat{y}_i - y_i) = Z^* \text{var}(\hat{a}_i - a_i) Z^{**} + 0.01113I,$$

where $y_i^*$ denotes the heights of the tally trees, $Z^*$ the design matrix of tally trees and $\text{var}(\hat{a}_i - a_i)$ includes the first two rows and columns of $\text{var}(\hat{b}_i - b_i)$ (see the definition of $b_i$). The height and volume models corrected for population level and local bias are shown in Figure 6.4.

![Figure 6.4. Predicted height and volume models when random parameters are 0 (dashed lines) and are predicted using the two observed heights shown in the plot on the left.](image-url)
6.5. SIMULTANEOUS MIXED MODELS

Seemingly unrelated system of mixed-effects models

In a seemingly unrelated mixed-effects model system, we have a total of $K$ linear mixed-effects models. Consider one sample plot with $n$ observations. For the plot in question, the $k$th model is of form

$$ y_k = X_k b_k + Z_k c_k + e_k $$

where $y_k$ is the vector of $k$th response variable $y_k = [y_{k1}, \ldots, y_{kn}]$, $X_k$ is the related model matrix of the fixed part, $b_k$ is the vector of fixed coefficients, $Z_k$ is the related model matrix of the random part, $c_k$ includes the corresponding random effects, and $e_k$ the residuals. The expected value of $c_k$ and $e_k$ are zero, and the variance-covariance matrices are

$$ \text{var}(c_k) = \begin{bmatrix} \text{var}(c_{k1}) & \text{cov}(c_{k2}, c_{k1}) \\ \text{cov}(c_{k1}, c_{k2}) & \text{var}(c_{k2}) \end{bmatrix} $$

(we assume 2 random parameters for simplicity), and $\text{var}(e_k) = a^2 I_{n \times n}$.

The individual models of the plot in question can be aggregated into a single model of the form

$$ y = Xb + Zc + e, $$

by defining

$$ y = [y_1, \ldots, y_K]^T, $$

$$ b = [b_1, \ldots, b_K]^T, $$

$$ c = [c_1, \ldots, c_K]^T, $$

$$ e = [e_1, \ldots, e_K]^T. $$

$$ X = \begin{bmatrix} X_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & X_K \end{bmatrix}, \text{ and} $$

$$ Z = \begin{bmatrix} Z_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Z_K \end{bmatrix}. $$

The expected values of $c$ and $e$ are $0$. The covariance matrix includes blocks of model-specific random effect variance-covariance-matrices on the diagonal and cross-model covariance matrices in the off-diagonal:

$$ \text{var}(c) = D = \begin{bmatrix} \text{var}(c_1) & \text{cov}(c_1, c_2) & \cdots & \text{cov}(c_1, c_K) \\ \text{cov}(c_2, c_1) & \text{var}(c_2) & \cdots & \text{cov}(c_2, c_K) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(c_K, c_1) & \text{cov}(c_K, c_2) & \cdots & \text{var}(c_K) \end{bmatrix}. $$

The variance-covariance matrix of residual is
Because the highest level residuals are assumed to be independent within a response variable, the covariance matrix blocks in \( \text{var}(\mathbf{e}) \) can be written as
\[
\text{cov}(\mathbf{e}_1, \mathbf{e}_1) \quad \text{var}(\mathbf{e}_1) \\
\vdots \\
\text{cov}(\mathbf{e}_K, \mathbf{e}_1) \\
\text{cov}(\mathbf{e}_K, \mathbf{e}_K) \quad \text{var}(\mathbf{e}_K)
\]
and we get \( \text{var}(\mathbf{e}) = R_{KK} \otimes I_{n_K} \), where \( R \) includes the residual variances and cross-model residual covariances.

Individual model fitting provides estimates of the diagonal blocks of \( R \) and \( D \), but estimation of the cross-model covariances requires seemingly unrelated regression fitting. However, this is not an easy task for a mixed-effects model, especially when the number of equations is high (see e.g. Fieuws and Verbecke 2006, Mehtätalo et al 2008). Therefore, we selected an easier procedure, where the individual mixed-effects models were estimated separately and the cross-model covariances were estimated afterwards using the predicted random effects and residuals. Even though this procedure results in slightly decreased efficiency of fixed parameter estimates and may produce estimates of correlations and variances that are shrunk towards zero, we used this approach for simplicity. This also demonstrates that application of the BLP for cross-model calibration does not necessarily require complicated estimation methods that are not necessarily available in statistical softwares or do not necessarily converge. A similar approach was also used by Lappi (2006), who used this simpler approach to more detailed modelling of the correlations and variances for improved cross-model calibration in the case of taper curves.

**Best linear predictor in the general case**

Assume that we have a vector of random variables, \( \mathbf{h} \), which can be divided into two parts
\[
\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{bmatrix}
\]
where both \( \mathbf{h}_1 \) and \( \mathbf{h}_2 \) are random vectors, but can also be scalars in the special case when the length is 0. Assume that the following first and second order properties of these vectors are known: \( E(\mathbf{h}_1) = \mu_1 \), \( E(\mathbf{h}_2) = \mu_2 \), \( \text{var}(\mathbf{h}_1) = \mathbf{V}_1 \), \( \text{var}(\mathbf{h}_2) = \mathbf{V}_2 \), and \( \text{cov}(\mathbf{h}_1, \mathbf{h}_2') = \mathbf{V}_{12} \). Using the notation of McCulloch and Searle (2001, p. 247), this can be written as
\[
\begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} + \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_{12} \end{bmatrix} \begin{bmatrix} \mathbf{V}_2 \end{bmatrix}.
\]
In many cases, we have observed a value of some random vector, and the aim is to predict the most likely value of the other vector. If \( \mathbf{h}_2 \) is observed, the Best Linear Predictor of \( \mathbf{h}_1 \) is
\[
\text{BLP}(\mathbf{h}_1 | \mathbf{h}_2) = \mu_1 + \mathbf{V}_{12} \mathbf{V}_2^{-1} (\mathbf{h}_2 - \mu_2) \tag{1}
\]
with the prediction variance of
\[
\text{var}(\text{BLP}(\mathbf{h}_1 | \mathbf{h}_2)) = \mathbf{V}_1 - \mathbf{V}_{12} \mathbf{V}_2^{-1} \mathbf{V}_{12}^\top.
\]
6.5. SIMULTANEOUS MIXED MODELS

\[ \text{var}(\text{BLP}(h_1) - h_1) = V_1 - V_{12}V_2^{-1}V_{12} \]  \hspace{1cm} (2)

(McCulloch and Searle 2001, p. 250).

If \( h \) is multivariate normal, i.e.,

\[ \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \sim N \left( \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} V_{11} & V_{12} \\ V_{12}' & V_2 \end{bmatrix} \right) \]

BLP will also be the Best Predictor, i.e., the conditional expectation \( E(h_1|h_2) \). If the matrices \( V_1, V_2 \) and \( V_{12} \) and the vector \( \mu_2 \) are replaced in the calculations by their estimates, the resulting predictor is the Estimated Best Linear Unbiased Predictor (EBLUP).

**Prediction of random effects for a seemingly unrelated mixed-effects model**

Assume that the value of the response variable \( y_k \) for some response’s \( k \) (but not necessarily for all of them) has been observed for the \( n \) trees of the plot. We want to predict all the stand effects of the model, that is both the stand effects of the responses that have been observed for the sample trees and the stand effects of the unobserved responses.

Let us start by defining vector \( y_0 \), which is formed in the similar manner as we formed vector \( y \) before, but we use only those vectors \( y_k \) that have been observed. Correspondingly, we define matrices \( D_0 \) and \( R_0 \) by removing from \( D \) and \( R \) those rows and columns that correspond to the unobserved responses. In addition, we define matrix \( C \) by removing from matrix \( D \) all the columns that correspond to the unobserved responses but keeping all the rows; \( Z_0 \) and \( X_0 \) by removing from \( Z \) and \( X \) those rows and columns that correspond to the unobserved response, and \( b_0 \) by removing from \( b \) the coefficients realted to the 2\(^{nd}\), 4\(^{th}\) and 5\(^{th}\) responses. If all the responses have been observed, then nothing needs to be removed from \( D, R, Z, X \), and \( b \) to formulate matrices \( D_0 \) and \( R_0, Z_0, X_0 \) and \( b_0 \).

As an example, assume that we have a system of five responses and each response has two random effects and three fixed parameters. Assume that the first and third response have been observed from 8 trees and consequently the second, fourth and fifth responses have not. The vector \( y_0 \) will be of length 16, including first the 8 observations of response 1 and then the 8 observations of response 3. Matrix \( D_0 \) will be of size 4x4, including the 1\(^{st}\), 2\(^{nd}\), 5\(^{th}\) and 6\(^{th}\) rows and columns of the 10x10 matrix \( D \); Matrix \( R_0 \) will be of size 2x2, including rows and columns 1 and 3 of matrix \( R \); and matrix \( C \) will include all 10 rows of \( D \) but only the 1\(^{st}\), 2\(^{nd}\), 5\(^{th}\) and 6\(^{th}\) columns. Matrix \( Z_0 \) will be a block diagonal matrix of size 16x4, including the 8x2model matrix of model 1 as the first diagonal block and the 8x2 model matrix of response 3 as the second block on the diagonal. Matrix \( X_0 \) will be a block diagonal matrix of size 16x6, including the 8x3 model matrix of model 1 as the first diagonal block and the 8x3 model matrix of response 3 as the second block on the diagonal, and \( b_0 \) will be of length 6, including the three fixed parameters of model 1, followed by the three fixed
parameters of model 3.

The BLP can be applied to predict the vector of random effects by defining $\mathbf{h}_1 = \mathbf{e}$ (i.e., the vector of all random effects for the plot of question, both those of the observed responses and of the unobserved responses) and $\mathbf{h}_2 = \mathbf{y}_0$ (i.e., the vector of observed responses for the n sampled trees). The expected values of these vectors are $\mu_1 = 0$ and $\mu_2 = X_0 \mathbf{b}_0$. The required variance-covariance matrices are $V_1 = D$, $V_2 = Z_0 D_0 Z_0' + R_0 \otimes I_{m \times n}$, and $V_{12} = Z_0 C'$. After defining these vectors, the random effects can be predicted by using these matrices in equation (1). The result will be a vector including the predicted plot effects for all responses for the plot in question. The estimation error of the random effects can be estimated using equation (2). Replacing the resulting predictions of random effects into the model provides the localized, stand level models for all responses to be used for prediction in the stand in question.
6.6  Exercices

1. Show that 2SLS will lead to OLS if the models are unrelated. Explain how this would happen in practice.

2. Fit a system of models into your own dataset. Implement the cross-model calibration.
Chapter 7

Modeling tree size distributions

7.1 Background

7.1.1 Overview of approaches

The diameter distribution is the basis of the stand description. Many approaches have been used to construct the diameter distribution of a stand. Mehtätalo (2004c) divided them three main approaches:

i those based on a sample of diameters

ii those based on prediction or recovery of the parameters of an assumed theoretical distribution model and

iii those based on known diameter distributions of similar stands (imputation methods)

**Approach (i).** The most natural way to obtain the diameter distribution is to measure a diameter sample from the stand. If the sample is large enough, it can be used as such in the simulation (e.g. Pienaar and Harrison 1988, Nepal and Somers 1992, Tang et al. 1997). If the sample is small, it can be smoothed (e.g. Droessler and Burk 1989, Uuttera and Maltamo 1995) or a theoretical distribution function can be fitted to it (e.g. Bailey and Dell 1973, Zarnoch and Dell 1985, Van Deusen 1986, Shiver 1988, Lindsay et al. 1996, Zhou and McTague 1996, Scolforo et al. 2003, Zhang et al. 2003). The theoretical distribution function can be fitted using the method of maximum likelihood, the method of moments, methods based on linear regression or by utilizing properties of certain percentiles or stand variables.

**Approach (ii).** The measurement of a diameter sample is too time-consuming in many inventories. In these cases, the diameter distribution can be predicted with some easily measurable stand variables. Traditionally, these methods are divided into
parameter prediction methods (PPM) and parameter recovery methods (PRM) (Hyink and Moser 1983). In the parameter prediction method, the parameters of the assumed distribution function are predicted with some measured stand variables using estimated regression models (e.g. Schreuder et al. 1979, Little 1982, Rennols et al. 1985, Kilikki and Päivinen 1986, Kilikki et al. 1989, Maltamo 1997, Siipilehto 1999, Temesgen 2003, Robinson 2004). In the PRM the parameters are recovered from some stand variables using known relations between the stand variables and distribution parameters (e.g. Ek et al. 1975, Burk and Newberry 1984, Magnussen 1986, McTague and Bailey 1987, Kuru et al. 1992). The stand variables used in PRM may be, for example, percentiles or moments of the diameter distribution. The recovery is, however, possible only for as many parameters as there are measured stand variables that are linked with the diameter distribution. If the number of parameters is greater partial recovery can be used, i.e. as many variables as possible are recovered and the other parameters are predicted (e.g. Kilikki and Päivinen 1986, Kangas and Maltamo 2000b, Mehtätalo 2004b, 2005b, Mehtätalo and Kangas 2005).

**Approach (iii).** The third approach is to use known diameter distributions of similar stands as the predicted diameter distribution of the stand (e.g. Haara et al. 1997, Maltamo and Kangas 1998, Packalén and Maltamo 2008). These methods have also been called imputation methods (Ek et al. 1997, Temesgen 2003, Temesgen et al. 2003). The similar stands are selected from a neighborhood that is defined with a distance function. The imputation methods used in predicting the diameter distribution are the k-nearest-neighbor method (Altman 1992) and the most similar neighbor method (Moeur and Stage 1995). The approach of Nanos and Montero (2002), where an interpolated surface is used to carry information from geographical neighbors to the target stands, also belongs to this class of approaches.

The above approaches can also be combined. For example the neighbors of the third approach may be smoothed diameter distributions instead of true distributions, thus combining approaches (i) and (iii) (Maltamo and Kangas 1998). Furthermore, sample information can be used to improve a predicted distribution; examples of this are the Bayesian approach of Green and Clutter (2000) where the prior information of neighboring stands (iii) is combined with sample information (i) and the approach of Mehtätalo (2005b), Mehtätalo and Kangas (2005) and Mehtätalo et al. (2010) where sample information (i) is used to improve predicted diameter distribution (ii). The approach of Maltamo et al. (2003) combining empirical distributions of large trees identified from a digital video imagery with predicted distribution of small trees is a combination of (i) and (ii).
7.2. PLOTTING AN EMPIRICAL DIAMETER DISTRIBUTION

7.1.2 Tree size and size distribution

In this section, we assume that $X$ is a random variable characterizing the size of a tree in a forest stand. The most common variable used for tree size is tree diameter. However, there are also other alternatives, such as crown diameter, crown area, basal area, and tree volume, to give some examples.

The within stand (or within plot) variability in tree size is accounted for through tree size distribution. If tree size specifies the tree diameter, then this distribution is called the diameter distribution. However, it might be the height distribution, the distribution of volume, or that of crown area as well.

Tree size distribution of a stand is defined as the probability that a randomly selected tree from the target stand is smaller than a fixed size $x$,

$$F(x) = P(X \leq x)$$

The corresponding density is

$$f(x) = F'(x)$$

An alternative interpretation for diameter distribution is the proportion of trees with the size smaller than $x$. The possible reasons for the randomness of $X$ may be that (i) the tree was selected at random, (ii) we assume a superpopulation model that generated the trees of the stand, or (iii) that there is measurement errors in the tree size (see also section 1.1).

The assumption about a superpopulation model is adopted here. Under this assumption, it is natural to assume that the size distribution of the area (e.g., the stand or plot) is continuous. Then all the trees of the area are identically distributed realizations of this superpopulation model. It may be further assumed that the realizations are independent, i.e., the size of a tree does not depend on the sizes of the other trees (e.g., the neighbours). However, it may be quite unrealistic assumption, since the tree size and other characteristics usually have spatial autocorrelation within a forest stand.

7.2 Plotting an empirical diameter distribution

7.2.1 Histogram

The most usual and natural way to plot an empirical diameter distribution is the histogram. For a histogram, the data (e.g. tree diameters) are first classified in a sufficient number of (diameter) classes. Then a barplot is produced, where the height of each bar shows the number of observations in the class.
An alternative scaling for the bars is to show the relative proportion of the class, which is further divided by the class width. Then the total area of the bars is one. This produces a plot that corresponds to the density of a p.d.f., and it is a natural way of plotting when a graph of a density function is added to the same plot. Figure 7.1 shows an example of such an histogram. It was produced by using R- function `hist` with option `freq=FALSE`.

### 7.2.2 Smoothing

Smoothing can be used to produce nice smooth plots of the diameter distribution (usually the density) without assuming any specified functional form for the distribution. Maybe the most common method for smoothing is the Kernel method (e.g. Härdle 1990).

The idea of Kernel smoothing is to plot each observed diameter on x-axis, to replace these points with a density function called kernel (e.g., Gaussian density), to sum up these densities, and finally, to rescale to unity by dividing the sum by the sample size. This results in such an estimate of the density that is smoother than the histogram of the sample but much more flexible than a fitted parametric density function (e.g the Weibull function) would be. Figure 7.2 demonstrates the principle.

**The R-code for Figure 7.2**

```r
> pdf("figdd11.pdf",width=7,height=6)
> par(mfcol=c(1,1),mai=c(1,1,0.1,0.1))
> d<-c(24.9,32.0,33.3,33.6,41.2)
> plot(d,rep(0,length(d)),xlim=c(15,50),ylim=c(0,0.12),ylab="Density")
> x<-seq(20,45,0.01)
> lines(x,dgkernel(x,d,bw=2),lwd=2)
> sapply(d,function(mu) lines(x,dnorm(x,mean=mu,sd=2)/length(d)))
> dev.off()
```
7.2. PLOTTING AN EMPIRICAL DIAMETER DISTRIBUTION

Figure 7.2: The idea of kernel smoothing. The open circles on the x-axis show the 5 observed values of $D$. The dots have been replaced with the Gaussian density which has been multiplied by $1/5$, so that the summed area under the 5 densities is one. The kernel density at $D = d$ is the sum of the scaled Gaussian densities at $D = d$.

The cumulative distribution function (c.d.f.) corresponding to the kernel density is obtained by replacing each observation with a c.d.f., summing up these, and rescaling to have limit one in the right tail. The kernel densities have been applied for diameter distributions e.g. by Droessler and Burk (1989), Uuttera and Maltamo (1995) and Mehtätalo (2004a).

The density and distribution function of a kernel-smoothed empirical distribution based on sample of size $n$ can be expressed as

$$f_h(d) = \frac{1}{n} \sum_{i=1}^{n} k_h(d - d_{ki})$$

$$F_h(d) = \frac{1}{n} \sum_{i=1}^{n} K_h(d - d_{ki}),$$

where $k_h$ and $K_h$ are the density and distribution function of the utilized kernel with smoothing parameter $h$, and $d_{ki}, i = 1, \ldots, n$ are the sampled tree diameters from plot $k$. The utilized kernel function is usually a symmetric distribution function with mean zero. The width is determined by the smoothing parameter, which is set by the user.

Mehtätalo et al. (2010) utilized Gaussian kernel, where function $k_h$ and $K_h$ are the density and distribution functions of a Normal distribution with mean 0 and standard deviation $h$, to smooth diameter distributions from Spanish NFI data. In the selection of the bandwidth, they utilized Silvermans rule of thumb (Silverman 1986, p. 48)

$$h_s = 0.9 \min(\hat{\sigma}, \frac{\hat{R}}{1.34}) n^{1/5},$$

where $\hat{\sigma}$ is the estimated standard deviation and $\hat{R}$ the
CHAPTER 7. MODELING TREE SIZE DISTRIBUTIONS

interquartile range of the sample. In addition, they evaluated values \(0.5h_s\) and \(2h_s\) for the bandwidth.

**Example 7.1.** Ten tree diameters were measured on a sample plot. To make a nice-graph about these data, a Kernel method was used using Gaussian kernel. Three alternative bandwidths were used: The one based on Silverman (1986, p. 48), and the ones obtained by multiplying the rule by 2 and 0.5. The numerical values of these bandwidths are 2.33, 4.66, and 1.17. Figure 7.3 shows the original data and the smoothed distributions. The two narrowest bandwidths lead to trimodal density, whereas the widest bandwidth gives a unimodal density.

**The R-code for Example 7.1**

```R
# Kernel smoothing
# Same as density()
# Lauri Mehttalo 17 OCT 2007
# dgkernel<-function(x,d,bw=bw.nrd0(d)) {
# + n<-length(d)
# + sapply(x,function(a) 1/(n*bw)*sum(dnorm((a-d)/bw)))
# + }
# }
# # Cumulative distribution function based on a gaussian kernel and bandwidth b
# Lauri Mehttalo 17 OCT 2007
# pgkernel<-function(x,d,bw=bw.nrd0(d)) {
# + n<-length(d)
# + sapply(x,function(a) 1/n*sum(pnorm((a-d)/bw)))
# + }
# # pdf("figdd10.pdf",width=7,height=3)
# par(mfcol=c(1,2),mai=c(0.7,0.7,0.1,0.1),cex=0.7)
# d<-c(39.5,32.0,33.5,32.0,38.8,24.9,25.9,33.3,33.6,41.2)
# hist(d,freq=FALSE,ylim=c(0,0.15),main="",
# xlab="Diameter, cm",ylab="Density")
# points(d,rep(0,length(d)))
# x<-seq(20,45,0.01)
# lines(x,dgkernel(x,d,bw=bw.nrd0(d)))
# lines(x,dgkernel(x,d,bw=0.5*bw.nrd0(d)))
# lines(x,dgkernel(x,d,bw=2*bw.nrd0(d)))
# > plot(stepfun(d[order(d)],seq(0.5,10.5,1)/11),
# + ylim=c(0,1),main=""
```
Computations with diameter distributions

Transformations and weighted distributions are used to compute different stand characteristics from a diameter distribution. These relationships can be utilized, for example, in formulating prediction (PPM) and recovery (PRM) models for diameter distributions. As efficient procedures for evaluating integrals and solving numerical equations are nowadays freely available, closed form solutions for the desired functions are not necessarily needed. If we are able to write the formulas as integrals and systems of equations, numerical methods can be used to solve the equations and evaluate the integrals numerically.

The next two subsections deal with transformations and weighted distributions in more detail. These two issues seem sometimes to be mixed. For example, some authors may speak about basal area distributions even though they mean the diameter distribution weighted by the basal area. The distinction can be demonstrated as follows. Let us first assume that we have a histogram or stand table, where trees of a given stand are classified in diameter classes of width 2 cm, and the proportion of trees in a given class of the total number of trees is calculated. Such a distribution is shown in figure 7.1. In some other applications, we may be willing to classify the trees according to some other size-related characteristics, for example, we could classify them into height classes of width one meter. Then we would obtain the height distribution. This distribution has tree height on the x-axis instead of tree diameter. However, still the bars of the histogram specify the proportion of trees in a given height class. Switching from a diameter distribution to height distribution is carried out by transformation.

In contrast, weighting affects the y-axis of the histogram. For example instead of specifying which proportion of trees is included in a given diameter or height class, we may be willing to specify which proportion of the total basal area or total volume is included in a given diameter or height class. A very common example is the basal area weighted diameter distribution, which specifies the the proportion of the total basal area in diameter classes. However, we could have e.g., the tree height or volume as...
the weight as well. As an example and exercise, one could think of the meaning and potential use of the following distributions:

- diameter distribution weighted by basal area
- basal area distribution weighted by diameter
- diameter distribution weighted by diameter
- diameter distribution weighted by tree height
- height distribution weighted by tree diameter
- distribution of crown diameter weighted by tree height
- distribution of tree height weighted by the crown area
- distribution of crown area weighted by volume
- etc.

Maybe most of these are not useful in practice, but some might be.

### 7.3.1 Transformation of tree size

The relationships between different tree variables can be accounted for through transformations. Let \( Y \) be another variable for tree size, which is obtained from \( X \) through a monotonic transformation \( g(X) \). The distribution of \( Y \) is

\[
F_Y(y) = F_X(g^{-1}(y)) \quad \text{if } g \text{ is increasing}
\]

\[
F_Y(y) = 1 - F_X(g^{-1}(y)) \quad \text{if } g \text{ is decreasing}
\]

**Example 7.2.** Let \( D \) be tree diameter and \( H \) tree height. Assume that the diameter follows the two-parameter Weibull distribution, which has c.d.f.

\[
F_D(d) = 1 - \exp \left[ - \left( \frac{d}{\beta} \right)^\alpha \right].
\]

and height-diameter curve is

\[
h(d) = 1.3 + a \exp \left( \frac{b}{d} \right), \tag{7.1}
\]

where parameters have known values \( \alpha = 4, \beta = 15, a = 25 \) and \( b = -5 \).

Solving (7.1) for \( d \) gives

\[
g^{-1}(h) = \frac{b}{\ln \left( \frac{h-a}{a} \right)}.
\]
Figure 7.4: The diameter distribution and density (left) and the height distribution and density (right) in example 7.2. The vertical line shows the limit of dominant trees. The lowest plots demonstrate the expected value of tree height (Example 7.3) and the density of dominant tree heights (Example 7.4).
The distribution of tree height becomes

\[ F_H(h) = 1 - \exp \left(-\frac{b}{\beta \ln \left(\frac{h-1.3}{a}\right)}\right)^\alpha, \]

which is no more a Weibull distribution (c.f. Example 1.16)

The density is obtained by differentiating \( F_H(h) \) with respect to \( h \),

\[ f_H(h) = \frac{\alpha}{\beta} \left(\frac{b}{\beta \ln \left(\frac{h-1.3}{a}\right)}\right)^{\alpha-1} \exp \left(-\frac{b}{\beta \ln \left(\frac{h-1.3}{a}\right)}\right) \frac{-b}{(h-1.3) \left[\ln \left(\frac{h-1.3}{a}\right)\right]^2} \]

This distribution and the density are shown in the four top graphs of Figure 7.4.

**Example 7.3.** The mean height is the expected value of the height distribution \( \bar{H} = \int_0^{H_{max}} u f_H(u) du \). Numerical evaluation gave the result \( \bar{H} = 18.12103 \) m. The bottom left graph of figure 7.4 demonstrates the computation of the expected value in this case. The expected value is the area of the grayshaded object.

Another way to compute mean and dominant height is to use the expectation of a transformation. Writing the assumed density and H-D curve into equation (1.16) gives

\[ \bar{H} = \int_0^{\infty} f_N(u) h(u) du = 18.12104 \]

**Example 7.4.** The height distribution of dominant trees is obtained by truncating the height distribution and rescaling it to unity (see the bottom right plot of Figure 7.4).

Dominant height is the expected value of the distribution of dominant trees. If we know that the stand density is \( N=500 \) stems/ha, then the limit of dominant trees is \( H_{lim} = F_H^{-1}((N-100)/N) \approx 19.90 \) m; that is, all trees that have the height above 19.9 meters are dominant trees. The dominant height is the mean height of the dominant trees,

\[ H_{dom} = E(H|H > H_{lim}) \] \hspace{1cm} (7.2)

\[ = \frac{N}{100} \int_{H_{lim}}^{\infty} u f_H(u) du = 20.44752. \] \hspace{1cm} (7.3)

Another option would be to use equation (1.16). For this purpose, we need to compute the diameter limit of dominant trees, \( D_{lim} = F_D^{-1}((N-100)/N) \approx 16.90 \) cm. The expected value is with dominant height yields

\[ H_{dom} = E(h(D)|D > D_{lim}) \] \hspace{1cm} (7.4)

\[ = \frac{N}{100} \int_{D_{lim}}^{\infty} f_D(u) h(u) du = 20.44752. \] \hspace{1cm} (7.5)

The numerical values from both integrals are the same at least up to the 5th decimal.

The R-code for Examples 7.2 7.3, and 7.4
7.3. COMPUTATIONS WITH DIAMETER DISTRIBUTIONS

> # The parameters of H-D curve
> a<-25
> b<-5
> # The parameters of the weibull distribution
> alpha<-4
> beta<-15
> # Number of stems and basal area
> N<-500
> # x- and y vectors for plotting purposes
> x<-seq(0,40,0.1)
> y<-seq(1.3,26.2,0.1)
> # HD-curve
> korfh<function(x,a=25,b=-5,c=1) {
+ 1.3+a*exp(b*x[-c])
+ }
> # Inverse of Korf curve
> korfh.inv<function(y,a=25,b=-5,c=1) {
+ (b/log((y-1.3)/a))^(1/c)
+ }
> # Plot the H-d-curve
> plot(x,korfh(x),type="l")
> # The density of the height distribution, when diameter follows a 2-parameter Weibull
> # and Height comes from a Korf curve
> dhdist<function(x,alpha,beta,a,b) {
+ dweibull(korfh.inv(x,a,b),alpha,beta) *(-b)/(x-1.3)/(log((x-1.3)/a))ˆ2
+ }
> pdf("figdd1.pdf",width=7,height=9)
> par(mfcol=c(3,2),mai=c(0.7,0.7,0.5,0.1))
> plot(x,pweibull(x,alpha,beta),type="l",
+ ylab="P(D<d)",xlab="Tree diameter d",
+ main=expression(F[D](d)))
> plot(x,dweibull(x,alpha,beta),type="l",
+ ylab="density",xlab="Tree diameter d",
+ main=expression(f[D](d)))
> plot(y,y*dhdist(y,alpha,beta,a,b),type="l",
+ ylab="h *density",xlab="Tree height h",
+ main=expression(df[D](d)))
> polygon(y,y*dhdist(y,alpha,beta,a,b),col=gray(0.7))
> text(18,1,"E(H)")
> plot(y,pweibull(korfh.inv(y),alpha,beta),type="l",
+ ylab="P(H<h)",xlab="Tree height h",
+ main=expression(F[H](h)))
> lines(rep(domlim,2),c(0,1))
> plot(y,dhdist(y,alpha,beta,a,b),type="l",
+ ylab="density",xlab="Tree height h",
+ main=expression(f[H](h)))
> lines(rep(domlim,2),c(0,1))
> fh.dom<-N/100*dhdist(y,alpha,beta,a,b)
> fh.dom[y<domlim]<-0
> plot(y,fh.dom,type="l",
+ ylab="density",xlab="Dominant tree height h",
+ main=expression(f[HDOM](h)))
> dev.off()
> # The expected value of the height distribution
> Hmean1<integrate(function(y) y*dhdist(y,alpha,beta,a,b),1.3,26.3)
> Hmean1
18.12103 with absolute error < 1.9e-05
> # another way to compute mean height
> Hmean2<integrate(function(x) korfh(x)*dweibull(x,alpha,beta),0,Inf)
> Hmean2
18.12104 with absolute error < 9.4e-06
The above example showed how the $H - D$ transformation can be used to derive the
distribution of height when a given diameter distribution is assumed. This approach can
be easily generalized to any other allometric relationships among trees. Thus, transforma-
tion can be used, for example to derive the distribution of basal area, volume, crown
area, crown height, root biomass, using the distribution of any size-related characteristics.

However, one should bear in mind that no randomness is assumed to the transfor-
mation. Thus it is assumed, for example, that tree height depends on tree diameter
deterministically according to the Korf’s curve with the given parameter values.

Example 7.5. The distribution of volume. We regard tree volume as a transformation,
which is obtained from tree diameter using the volume function of Laasasenaho (1982),

$$v(d) = e^{-5.394+3.481\ln(2+1.25d)-0.0399d}$$

The distribution of volume is expressed as

$$F_V(v) = F(g^{-1}(d)).$$
Unfortunately, transformation $g^{-1}(d)$ cannot be computed in a closed form. However, an easy way to compute the inverse transformation is to use a simple up-and-down algorithm. (The Newton-Raphson algorithm would be another even better option.) The up and down algorithm, can be used to solve equations of form $f(x) = 0$ numerically. To compute the value of the inverse volume function for volume $w$, we solve $v(d) = w$ for $d$ to get the diameter that corresponds to volume $w$. This diameter is then written into the Weibull cdf to get the cdf of volume. Figure 7.5 shows the graphs of the inverse transformation and the distribution of volume. Making a graph of the density would have required the use of numerical differentiation (one could use R-function numericDeriv for this), but it is left as an exercise for those interested in doing it.

The code below was used for making Figure 7.5.

```r
# Predicts tree volumes (liters) using the functions of Laasasenaho.
# pl=species (1= pine)
# malli mean model. Value of 1 uses only d; value of 2 uses d and h.
# predict.volume<-function(pl,d,h=0,malli=1) {
# pl2<pl
# pl2[pl2>4|pl2<1]<-5
# if (malli==1) {
# pars<-matrix(c(-5.39417,3.48060,0.039884,-5.39934,3.46468,0.0273199,
# -5.41948,3.57630,0.0395855,-5.41948,3.57630,0.0395855,
# rep(NA,3)),ncol=3,byrow=TRUE)
# exp(pars[pl2,1]+pars[pl2,2]*log(2+1.25*d)-pars[pl2,3]*d)
# } else if (malli==2) {
# pars<-matrix(c(0.036089,2.01395,0.99676,2.07025,-1.07209,
# 0.022927,1.91505,0.99146,2.82541,-1.53547,
# 0.011197,2.10253,0.98605,3.98519,2.65900,
# 0.011197,2.10253,0.98605,3.98519,-2.65900,
# rep(NA,5)),ncol=5,byrow=TRUE)
# pars[pl2,1]+pars[pl2,2]*log(2+1.25*d)*pars[pl2,3]+h*pars[pl2,4]*(h-1.3)*pars[pl2,5]
# }
# }
# }
# # An up-and-down algorithm
# # Solves fn(x)=0 for x between l (lower bound) and u (upper bound).
# updown<-function(l,u,fn,crit=6) {
# fnuc<-fn(u)
# fnl<-fn(l)
# if (fnl*fnuc<0) return(NA)
# value<-fn((u+l)/2)
# while (round(value,crit)!=0&(|u-l|>10^-crit)) {
# if (fnl*value>0) {
# u<-(u+l)/2
# fnuc<-value
# } else {
# l<-(u+l)/2
# fnl<-value
# }
# value<-fn((u+l)/2)
# }
# nollakohta<-{-1|u-l}/2
# cat(l,fnl,u,fnuc,"\n*
# )
# }
# pinevol.inv<-function(v) {
# voldif<-function(d) predict.volume(pl=1,d=d)-v
# updown(0,80,voldif)
# }
# ginv<-function(v) sapply(v,pinevol.inv)
```
Example 7.6. Mean volume in Example 7.5 The expected value of volume, that is the mean tree volume, can be computed as

$$\bar{V} = \int_0^{\infty} v(u) f_D(u) \, du$$

Numerical evaluation of the integral gave the result $\bar{V} = 85.4$ dm$^3$.

Example 7.7. Mean volume by applying a H-D model and a volume function. The above example did not utilize the information on the H-D curve. To utilize also that information, we use the volume function based on diameter and height (Laasasenaho 1982),

$$v(d, h) = 0.0361 d^{2.014}0.997^d h^{2.070}(h - 1.3)^{-1.072},$$

but replace the height $h$ with the known height model. The mean tree height is thus obtained by evaluating the integral

$$\bar{V} = \int_0^{\infty} v(u, h(u)) f_D(u) \, du.$$ 

Numerical evaluation gave the result $\bar{V} = 145.9$ dm$^3$, which is much higher value than the earlier obtained value 85.4 dm$^3$. The reason for a higher estimate of volume is that the known H-D curve gives much higher heights for a given diameter than the mean heights that are implicitly assumed in the simple volume function based on the diameter only.

The R-code for Example 7.6

```r
> volume2<-function(d) {
+ v<-predict.volume(pl=1, d=d, h=korfhd(d), malli=2)
+ v[d<=1.3]<-0
+ v
+ }
> integrate(function(d) volume1(d)*dweibull(d,alpha,beta),0,Inf)$value
[1] 85.43804
> integrate(function(d) volume2(d)*dweibull(d,alpha,beta),0,Inf)$value
[1] 145.8982

7.3.2 Basal-area weighted distribution

The previous subsection demonstrated the use of transformations to change the variable that is on the x-axis of a distribution graph. In this subsection, we switch to weighted distributions, that are used to change the variable of the y-axis.
With tree size size distributions, the (diameter) class frequencies are usually proportional to the number of stems. Loosely speaking, this means that the diameter distribution specifies how many trees of a given diameter we have in the stand (or, more specifically, how many trees we have within specified diameter limits). On the other hand, the distribution of height specifies the number of trees of a given height; or the distribution of basal area specifies the number of trees of a given basal area.

It is often more convenient to have the frequencies proportional to some other characteristics than the number of stems. With diameter distributions, the most common characteristics, other than the number of stems, is the basal area. The basal-area weighted diameter distribution (I’m sorry about this jargon-like term) measures the amount of trees in a given diameter class with basal area instead of the number of stems. The reasons for basal area weighting are that (i) it is a consistent model for angle-count samples and (ii) it gives more weight to the largest, most valuable trees.

If we assume that the density of the (unweighted) diameter distribution is \( f_N^D(d) \), then the basal-area weighted diameter distribution is

\[
\begin{align*}
f_G^D(d) &= \frac{\frac{\pi}{40000} d^2 f_N^D(d)}{\int_0^\infty \frac{\pi}{40000} u^2 f_N^D(u) \, du} \\
&= \frac{\frac{\pi}{40000} d^2 f_N^D(d)}{\int_0^\infty u^2 f_N^D(u) \, du} \\
&= \frac{d^2 f_N^D(d)}{E(d^2)} \tag{7.7}
\end{align*}
\]

of which either the second or third version are useful in computations. Note the additional superscripts \( N \) and \( G \) in the distribution and density functions to express whether the y-axis represents the proportion of the number of stems or the proportion of the total basal area.

For example, assuming the Weibull diameter distribution, which has the density

\[
f_N^D(d|\alpha, \beta) = \frac{\alpha}{\beta^\alpha} d^{\alpha-1} e^{-(d/\beta)^\alpha},
\]

gove and Patil (1998) showed that the basal-area weighted diameter distribution would be of the form

\[
f_G^D(d|\alpha, \beta) = \frac{d^2 \left( \frac{\alpha}{\beta^\alpha} \right) d^{\alpha-1} e^{-(d/\beta)^\alpha}}{\beta^2 \Gamma(2/\alpha + 1)}
\]

\[
= \frac{1}{\Gamma(k)} y^{k-1} e^{-y},
\]

where \( y = (d/\beta)^\alpha \) and \( k = 2/\alpha + 1 \). We notice that this is the pdf of the standard Gamma distribution. Essentially, assuming the diameter distribution to be of the form of the two-parameter Weibull implies that the basal-area weighted distribution of transformation \( y = (d/\beta)^\alpha \) is of the form of standard Gamma distribution.
A generalized framework for distributions of form

\[ f_\alpha^*(x) = \frac{x^\alpha f(x)}{E(x^\alpha)} \] (7.10)

is provided by the size-biased distribution theory (Gove and Patil 1998, Gove 2003a). The basal-ara weighted diameter distribution, which is obtained by specifying \( \alpha = 2 \), are the most common example of this. This is called the size-biased distribution of order 2.

Maybe the most important application of the size-biased distributions arises when the angle-count sampling (relascope sampling, horizontal point sampling) is used and a specific distribution is assumed for unweighted distribution (Van Deusen 1986, Gove and Patil 1998, Gove 2000, 2002, 2003b,a). In such situation, the sampling distribution is the basal-area weighted version of the assumed distribution. Consequently, the weighted version of the distribution is to be fitted to the data instead of the original assumed distribution.

Weighting by the basal area means that the bigger trees have a larger sampling probability. Thus, the mean diameter of the sampled trees is higher than the mean diameter in the population. In addition, the diameter distribution is also of a different shape.

**Example 7.8. Distribution obtained by angle-count sampling.** Consider a simulated, one-hectar forest stand with 1000 trees and diameters from \( Weibull(3, 20) \) distribution. For simplicity, the tree locations are assumed to be random and tree size is assumed to be uncorrelated within the stand. Furthermore, assume that a sample is taken using a relascope with BAF=1.
7.3. COMPUTATIONS WITH DIAMETER DISTRIBUTIONS

The following R-code presents simulation for such a stand.

```r
> # Set the random number seed (just to get the same sample every time I run the code)
> set.seed(1234567)
> # fitting basal area weighted distributions to data
> # Simulate a one hectar forest stand with 1000 trees
> # Diameters from Weibull(3,20)
> d<-rweibull(1000,3,20)
> # x- and y-coordinates from uniform distribution
> x<-runif(1000,-50,50)
> y<-runif(1000,-50,50)
> # Simulate an angle-count sample with BAF 1
> dist<sqrt(x^2+y^2)
> dsample<d[2*dist<d]
```

A histogram of the diameters of all the trees is shown in gray in Figure 7.6, and
the histogram of the sample in black. The mean diameter in the stand is 17.7 cm,
and the mean of the sample is 22.41 cm. The lines show the underlying Weibull-
distribution and the basal-area weighted sampling distribution. Both distributions are
quite symmetric, but the sampling distribution is more peaked than the distribution of
tree diameters.

The figure was produced using the following R-code:

```r
histw<-function(d,w=rep(1,length(d)),breaks=seq(min(d),max(d),length=10),
               add=FALSE,border="Black",col=NA,density=NA,angle=45,lwd=1,
               lty="solid",xlab="d",ylab="Density",
               freq=FALSE,ylim=c(0,0),nr=c(1,1),vali=0.0) {
  cuts<-breaks
  ncuts<-length(cuts)
  f<-sapply(seq(1,ncuts-1),function(x) sum(w[d>=cuts[x]&d<cuts[x+1]]))
  width<-cuts[2]-cuts[1]
  w2<-(1-vali)*width
  if (!freq) f<-f/sum(f)/width
  if (!add) plot(cuts[-2],f,type="n",xlab=xlab,ylab=ylab,ylim=c(0,max(c(f,ylim))))
  sapply(seq(1,ncuts-1),function(x)
    polygon(rep(0.5*(width-w2)+c(cuts[x]+w2*(nr[1]-1)/nr[2],cuts[x]+w2*nr[1]/nr[2]),each=2),
    c(0,f[x],f[x],0),
    col=col,border=border,density=density,angle=angle,
    lwd=lwd,lty=lty))
  abline(0,0)
}

> pdf("figdd2.pdf",width=7,height=6)
> par(mfcol=c(1,1),mai=c(0.7,0.7,0.5,0.1))
> x<-seq(0,40,0.1)
> histw(d,breaks=seq(0,40,4),nr=c(2,2),vali=0.1,col=gray(0.5),ylim=c(0,0.1))
> lines(x,dweibull(x,3,20),lwd=2,col=gray(0.5))
> histw(dsample,breaks=seq(0,40,4),nr=c(1,2),vali=0.1,add=TRUE,col=gray(0))
> lines(x,x^2*dweibull(x,3,20)/(20^2*gamma(2/3+1)),lwd=2)
> mean(d)
[1] 17.67690
> mean(dsample)
[1] 22.41906
> dev.off()
```

Example 7.9. Fitting Weibull distribution to angle count sampling data. Assume
that tree diameter follows the Weibull- distribution, and the diameter sample of example 7.8 is available from the stand. The diameters have been tabulated below.

\begin{verbatim}
18.8 23.7 23.4 20.1 26.4
26.2 25.0 30.4 22.6 25.1
23.9 10.8 21.7 31.0 22.3
24.9 22.4 15.0 15.1 22.2
29.8 17.3 12.6 27.3
\end{verbatim}

The Weibull- parameters are estimated by fitting the assumed sampling distribution

\[ f_B^G(d|\alpha, \beta) = \frac{d^2\left(\frac{\alpha}{\beta}\right) d^{\alpha-1}e^{-\left(d/\beta\right)^\alpha}}{\beta^2 \Gamma\left(\frac{2}{\alpha} + 1\right)} \]

to the observed data by using the method of maximum likelihood. The log-likelihood is

\[ \ell(\alpha, \beta|d_1, \ldots, d_n) = \sum_{i=1}^n \ln(f_B^G(d_i|\alpha, \beta)), \]

where \( n \) is the number of measured tree diameters on the relascope sample plot. The fit was performed using function \texttt{mle} of package \texttt{stats4}.

The following estimates were obtained \( \hat{\alpha}_{ML} = 3.997 \) and \( \hat{\beta}_{ML} = 21.67 \), with variance-covariance matrix \( \text{var}(\hat{\alpha}_{ML}, \hat{\beta}_{ML}) = \begin{pmatrix} 0.6542 & 0.9034 \\ 0.9034 & 2.064 \end{pmatrix} \).

The following R- code defines the sampling density, negative log likelihood, and performs the ML- fitting using function \texttt{mle} of package \texttt{stats4}.

```r
> dsampling<-function(x,shape,scale) {
+   x^2*dweibull(x,shape,scale)/(scale^2*gamma(2/shape+1))
+ }
```
7.3. COMPUTATIONS WITH DIAMETER DISTRIBUTIONS

> ll2 <- function(shape, scale) {
+ -sum(log(dsampling(dsample, shape, scale)))
+ }
>
> library(stats4)
> est.ml <- mle(ll2, start = list(shape = 3, scale = 20))
>
> est.ml

Call: mle(minuslogl = ll2, start = list(shape = 3, scale = 20))

Coefficients:
shape scale
3.996933 21.666591

> vcov(est.ml)

shape scale
shape 0.6542356 0.9034208
scale 0.9034208 2.0635600

Example 7.10. Another alternative for fitting would be the method of moments. In this application, the two first moments of the relascope sample are computed and set equal to the theoretical moments of the sample distribution. More specifically, the following pair of equations is solved for \( \alpha \) and \( \beta \)

\[
\int_0^\infty x f_D(x|\alpha, \beta)dx - \frac{1}{n} \sum_{i=1}^n d_i = 0
\]

\[
\int_0^\infty x^2 f_D(x|\alpha, \beta)dx - \frac{1}{n} \sum_{i=1}^n d_i^2 = 0
\]

In Example 1.40, we used function \texttt{nls} for solving the equation. That approach led to numerical difficulties in this example. That is why we use the two-dimensional version of the well-known Newton’s method. The methods uses, in adition to the functions themselves, their partial derivatives to search for the solution iteratively. Fairly good initial estimates are needed for a successful solution. Details on the method can be found in most books of numerical methods. The sample moments were 22.41 cm and 529.94 cm\(^2\). Solving the equation provided the estimates \( \hat{\alpha}_{MOM} = 3.739 \) and \( \hat{\beta}_{MOM} = 21.36 \). Figure 7.7 shows the sample and the estimated distributions based on the methods of moments and ML.

The following code performs the method of moments.

> m1 <- mean(dsample)
> m2 <- mean(dsample^2)
> m1
[1] 22.41906
> m2
[1] 529.9396
> 
> # Newton raphson algorithm for n multidimensional functions using numeric differentiation
# solves equations fn_i=0, where fn_i is the ith function of object fnlist,
# a list of scalar valued functions.
# The initial values of the parameters are provided in vector ‘init’.
# The required partial derivatives are obtained numerically using ‘numericDeriv’
> NRnum2 <- function(init, fnlist, crit=6) {


CHAPTER 7. MODELING TREE SIZE DISTRIBUTIONS

```
+ par<-init
+ # sapply(par,function(x) cat(x," "))
+ # cat("\n")
+ value<-sapply(fnlist,function(f) f(par))
+ j<-1
+ while (round(sum(abs(value)),crit)!=0&j<100) {
+ # cat(j,"\n")
+ grad<-t(sapply(fnlist,function(f) attributes(numericDeriv(quote(f(par)),c("par")))\$gradient))
+ deltax<-solve(grad,-value)
+ par<-par+deltax
+ # sapply(par,function(x) cat(x," "))
+ # cat("\n")
+ value<-sapply(fnlist,function(f) f(par))
+ j<-j+1
+ }
+ if (j<10000) {
+ list(par=value)
+ } else {
+ list(par=NA,value=NA)
+ }
+ }
>
> # Define the individual functions and put them into a list
> fn1<-function(parms) wmom2(1,parms[1],parms[2])-m1
> fn2<-function(parms) wmom2(2,parms[1],parms[2])-m2
>
> fnlist<-list(fn1,fn2)
>
> # Solve the parameters
> est.mom<-NRnum2(c(3,20),fnlist)
> est.mom
$par
>
$value
[1] 6.456307e-09 3.789124e-07

The R-code for Figure 7.7

```

Especially in Finnish studies, assumptions are made of the basal-area weighted distributions instead of making them of the unweighted distribution (e.g., Kilkki and Päivinen 1986, Kilkki et al. 1989, Maltamo 1997, Maltamo and Kangas 1998, Kangas and Maltamo 2000a, b, Kangas et al. 2007, Siipilehto 1999, Siipilehto et al. 2007, Mehtätalo 2004a, 2005b). For example, It has been assumed that the basal-area weighted diameter distribution follows The Weibull, beta, Johnsons SB, or the percentile-based distribution. This assumption is useful when the modelling data are collected using angle count sampling, and when the scaling of the distribution to stand or plot level
is performed using the basal area, not the number of stems. However, no other justifications beyond these technical restrictions are presented. Furthermore, they can be overcome fairly easily, as will be demonstrated in section 7.3.5. On the other hand, no justifications for assuming the unweighted density to be of a specific form have ben presented either. Thus, it is justified to present a way to fit an assumed basal-area weighted diameter distribution both to angle count sampling data and to data collected without weighting, e.g., by using sample plots of a fixed-area.

**Example 7.11. Fitting basal area weighted Weibull function to angle count data**

We now assume that the basal-area weighted diameter distribution is of the Weibull form, and the data are collected using angle-count sampling. We fit the Weibull distribution to the angle count sample of Example 7.9. The log likelihood is similar to that of example 7.9:

$$\ell(\alpha, \beta) = \sum \ln(f_G^D(d|\alpha, \beta)),$$

(7.11)

where $f_G^D$ is the basal-area weighted density which is assumed to be of the Weibull form in this example. This is the difference to the previous example 7.9, where a weighted distribution was assumed.

The ML-fit gives the estimates $\hat{\alpha} = 3.60$ and $\hat{\beta} = 23.27$. The dotted gray line of figure 7.7 shows the fitted basal-area weighted Weibull distribution, and the black dotted line the corresponding unweighted density. We cannot compare these to the true values; the true values cannot be specified because a basal-area weighted Weibull distribution was not assumed when the data were generated.

**The R-code for Example 7.11**

```r
> # Fit by assuming the basal-area weighted distribution to be of the Weibull-form
> ll2<-function(shape,scale) {
+   -sum(log(dweibull(dsample,shape,scale)))
+ }
> library(stats4)
> est.ml2<mle(ll2,start=list(shape=3,scale=20))
> est.ml2
Call:
mle(minuslogl = ll2, start = list(shape = 3, scale = 20))

Coefficients:
         shape    scale
3.604142  23.268315
> vcov(est.ml2)
     shape     scale
shape 0.1422802 0.1235781
scale 0.1235781 0.9756642
```

If the data are collected by using fixed area-plots and, for some reasons, we may wish to make assumptions on the basal-area weighted distribution, then we need to do the weighting in the opposite direction. The unweighted density is obtained from the
basal area weighted diameter distribution, \( f_D^G \), through

\[
f_N^D(x) = \frac{d^{-2} f_D^G(d)}{\int_0^\infty u^{-2} f_D^G(u) du}.
\]

The equation is very similar to the equation for the opposite situation, except for that the exponent 2 has now been replaced with exponent -2.

With the two-parameter Weibull-distribution, this distribution becomes (c.f Equation (7.9))

\[
f_N^D(d|\alpha, \beta) = \frac{d^{-2} \left( \frac{\alpha}{\beta^\alpha} \right) d^{\alpha-1} e^{-\left(\frac{d}{\beta}\right)^\alpha}}{\beta^{-2} \Gamma(1 - 2/\alpha)}
\]

The gamma function is defined only with a positive value of its argument. Thus, in contrast to the Weibull density (1.41), the generalized gamma distribution of form (7.13) is defined only when \( \alpha > 2 \) and \( \beta > 0 \).

**Example 7.12.** Fitting basal-area weighted Weibull density to fixed area plot data.

Using the data generated for Example 7.8, we now demonstrate the ML-fit by assuming the basal-area weighted distribution to be Weibull, and the data collected from fixed area plots. We ignore the fact that this assumption is wrong with the simulated dataset of this example. In practice, assuming a basal-area weighted distribution to be of the
Weibull form may be a realistic option, even though no studies have compared these two assumed distributions to each other (at least I am not aware of them).

Assuming the basal area weighted distribution to be of the Weibull form is equivalent to assuming the unweighted distribution to be of the form (7.13). To fit the assumed distribution, we specify the log likelihood as

\[ \ell(\alpha, \beta) = \sum \ln(f_{ND}(d|\alpha, \beta)), \]

where \( f_{ND} \) is as specified in equation (7.13).

The ML estimates for the parameters are \( \hat{\alpha} = 4.58 \) and \( \hat{\beta} = 23.81 \). We cannot compare these to the true values; the true values cannot be specified because the data were generated by assuming a different distribution. However, these values should compare to the estimates obtained in example 7.11 by fitting to an angle-count sample. The density of the fitted curve is shown by the solid black line in figure 7.8.

Example 7.13. Fitting unweighted Weibull density to fixed area plot data For comparison, a fit was also performed by assuming the unweighted distribution to be of the Weibull form. In that case, the Weibull density was used as function \( f_{ND} \) in equation (7.14). The obtained ML estimates for the parameters were \( \hat{\alpha} = 3.11 \) and \( \hat{\beta} = 20.10 \). They are close to the true values 3 and 20 which were used in simulation. The dashed line of Figure 7.8 shows the fitted curve. We observe that assuming the weighted distribution to be of the Weibull form leads to more peaked distribution than assuming the unweighted distribution to be of the Weibull form.

The R-code for Example 7.12

```R
> # size-biased weibull density
> dsbweibull<-function(x,shape,scale,power=0) {
+ x^power *dweibull(x,shape,scale)/(scale^power*gamma(power/shape+1))
+ }
> >
> dsamplef<-d[dist<10]
> >
> # Fit by assuming the Basal-area weighted distribution to be Weibull
> l12<-function(shape, scale) {
+ - sum(log(dsbweibull(dsample,shape,scale,-2)))
+ }
> >
> est.ml1<-mle(l12, start = list(shape=3, scale=20))
> est.ml1

Call:
  mle(minuslogl = l12, start = list(shape = 3, scale = 20))

Coefficients:
  shape  scale
  4.584056 23.809557
>
> # Fit by assuming the unweighted distribution to be Weibull
> l13<-function(shape, scale) {
+ - sum(log(dweibull(dsamplef,shape,scale))
+ }
> >
> est.ml2<-mle(l13, start = list(shape=3, scale=20))

```
> est.ml2

Call:
  mle(minuslogl = ll3, start = list(shape = 3, scale = 20))

Coefficients:
  shape  scale
3.106175 20.090270

> pdf("figdd6.pdf", width=7, height=6)
x<-seq(0,40,0.1)
histw(d, breaks=seq(0,40,4), nr=c(2,2), vali=0.1, col=gray(0.5), ylim=c(0,0.1))
lines(x, dsbweibull(x, 3, 20, 0), lwd=2, col=gray(0.5))

histw(dsamplef, breaks=seq(0,40,4), nr=c(1,2), vali=0.1, add=TRUE, col=gray(0))
aml1<-coef(est.ml1)[1]
bml1<-coef(est.ml1)[2]
lines(x, dsbweibull(x, aml1, bml1, -2), col="black", lwd=2)
aml2<-coef(est.ml2)[1]
bml2<-coef(est.ml2)[2]
lines(x, dsbweibull(x, aml2, bml2), col="black", lwd=2, lty="dashed")

dev.off()

7.3.3 Weighted distributions

A still more general framework than the size-biased distributions is that of weighted
distributions. In general the density of a weighted diameter distribution is

\[
 f_w^w(d) = \frac{w(d) f_D(d)}{\int_0^\infty w(u) f_D(u) \, du} = \frac{w(d) f_D(d)}{E(w(d))}
\]

The nominator scales the density to unity. It is the integral of the numerator over the
range of \( d \), i.e., the mean, or expected value of weight \( w(d) \). With basal-area weighting
it is the mean basal area and, for example with volume weighting it is the mean volume.
We immediately notice that the size-biased distribution is such a special case of the
weighted distributions, where the weight is of the form \( d^\alpha \).

Weighted distributions can be used for fitting assumed distributions to a sample ob-
tained through weighted sampling. In addition to the situation of basal area weighting,
which was considered in the previous subsection, such situations may be faced also in
some other applications. A quite common situation is to have the sampling probability
proportional to the tree size according to a known function. For example, the sample
plot radius may vary according to tree diameter. In addition, the data may be censored.
For example, in an aerial inventory, the small trees may not have been observed at all,
or may have a smaller probability of being observed than the big trees. Furthermore,
we may have a fixed minimum diameter for trees to be included in the sample. In this
situation, the observation probability for small trees is 0, which leads to a truncated
sampling distribution of tree diameters.
If a fixed diameter limit \( t \) is used for measurements, then the weight can be specified as

\[
w(d) = \begin{cases} 
0 & d \leq t \\ 
1 & d > t 
\end{cases}
\]

The weighted density, i.e., the sampling density, becomes

\[
g(d|\theta) = \begin{cases} 
0 & d \leq t \\ 
\frac{1}{I_{t=1}} \frac{1}{f(d|\theta)} \int_{t}^{\infty} f(u|\theta) \, du & d > t 
\end{cases}
\]

which is the truncated density. Thus, sampling trees by using a fixed minimum diameter results in sampling distribution that is a left-truncated version of the true population distribution. Correspondingly, a natural approach for fitting is to fit a truncated distribution to such data.

**Example 7.14. Fitting Weibull distribution to censored data** Assume the one-hectare stand that was generated in Example 7.8. However, assume that only trees with \( DBH > 10 \text{cm} \) have been measured. That is why a truncated Weibull distribution is fitted, where the truncation point is \( t = 10 \). For comparison, also a Weibull distribution without truncation is fitted.

The method of maximum likelihood is used in fitting. The code below carries out the fit. First, R- functions for the truncated pdf, density, quantile function and random number generation are defined (The last two are not needed in this example, but are included to be consistent with the R- style of distribution functions). Then a function
for the likelihood is defined. In addition, a function for the gradient of the likelihood is provided; using it will make the estimation more stable and quicker.

The parameter estimates and their approximate 95% confidence intervals are for the fit that does not account for the censoring $\hat{\alpha}_{\text{wrong}} = 3.69$ (c.i. 3.52 – 3.88) and $\hat{\beta}_{\text{wrong}} = 21.01$. (20.62 – 21.41). For the fit of truncated distribution, the corresponding figures are $\hat{\alpha}_{\text{correct}} = 2.97$ (2.83 – 3.12) and $\hat{\beta}_{\text{correct}} = 19.63$ (19.26 – 20.00). We notice that the confidence intervals from the ’wrong’ fit do not include the true values of parameters (3 and 20) whereas the confidence intervals from the ’correct’ fit do.

Figure 7.9 shows the fitted sampling distributions and the corresponding distributions of the underlying population.

The R-code for Example 7.14:

```r
> dsample2<-d[d>10]
> # cdf of left-truncated Weibull distribution
> pwt<-function(x,shape,scale,limit=0) {
+ F<-pweibull(x,shape,scale) - pweibull(limit,shape,scale)/(1-pweibull(limit,shape,scale))
+ F[x<limit]<-0
+ F
+ }
> # density of left-truncated Weibull distribution
> dwt<-function(x,shape,scale,limit=0) {
+ f<-dweibull(x,shape,scale)/(1-pweibull(limit,shape,scale))
+ f[x<limit]<-0
+ f
+ }
> # quantile function of left-truncated Weibull distribution
> qw<-function(p,shape,scale,limit=0) {
+ p2<-pweibull(limit,shape,scale) + p*(1-pweibull(limit,shape,scale))
+ x<-qweibull(p2,shape,scale)
+ x[x<limit]<-limit
+ x
+ }
> # random number generation from left-truncated Weibull distribution
> rwt<-function(n,shape,scale,limit=0) {
+ qwtr(runif(n),shape,scale,limit)
+ }
> library(stats4)
> library(nlme)
> # parameters in log scale
> nLL<-function(x,lshape,lscale,limit=0) {
+ shape<-exp(lshape)
+ scale<-exp(lscale)
+ # cat(lshape,lscale,"n")
+ -sum(log(dwtr(x,shape=shape,scale=scale,limit=limit)))
+ }
> grnLL<-function(x,lshape,lscale,limit=0) {
+ c<-sum(1+log(x)-lscale)+exp(lshape)+(limit/exp(lscale))
+ -sum(-1-exp(lshape)-1-exp(lshape)*limit/exp(lscale))
+ + x/exp(lscale))
+ x*exp(-lscale))
+ }
> fitwtr<-function(d,limit=5) {
```
In forest inventories, it is quite common to vary the sample plot radius according to tree size (e.g. Breidenbach et al. 2008). For example, the Spanish NFI uses concentric (i.e. circular) plots, in which the plot radius varies between 5 and 25 meters according
Figure 7.10: Figure for Example 7.15. The histograms show the underlying diameter distribution (gray) and the obtained PPS sample (black). The thin solid lines show the true (gray) diameter and sampling (black) distributions and the thick lines show the corresponding functions using the ML-estimates of the parameters.

To the tree diameter. Trees with diameter below $t = 7.5$ cm are not measured at all. Thus, the observations constitute a weighted sample from a truncated distribution with weights

$$w(d) = \begin{cases} 
\frac{5^2}{25^2} & 7.5 \leq d \leq 12.5 \\
\frac{10^2}{25^2} & 12.5 \leq d \leq 22.5 \\
\frac{15^2}{25^2} & 22.5 \leq d \leq 42.5 \\
1 & d \geq 42.5 
\end{cases}$$

assuming that the underlying spatial distributions of trees is homogeneous.

Assuming that $F(d|\theta)$ is the underlying diameter distribution of a stand and $f(d|\theta)$ is the corresponding density, the observations have the weighted density $g(d) = \frac{\int_I w(u)f(u|\theta)du}{\int_I w(u)f(u|\theta)du}$, where $I$ is the truncation point of 7.5 cm. Vector $\theta$ includes the parameter of the underlying diameter distribution of the stand. With variable-radius circular plots, the weighted density can be defined by parts as

$$g(d|\theta) = \begin{cases} 
\frac{5^2}{25^2} f(d|\theta) & 7.5 \leq d \leq 12.5 \\
\frac{10^2}{25^2} f(d|\theta) & 12.5 \leq d \leq 22.5 \\
\frac{15^2}{25^2} f(d|\theta) & 22.5 \leq d \leq 42.5 \\
\frac{7}{7} f(d|\theta) & d \geq 42.5 
\end{cases}$$

where $I = \int_{-\infty}^{\infty} w(u)f(u|\theta)du = 1 - F(42.5|\theta) + \frac{15^2}{25^2}[F(42.5|\theta) - F(22.5|\theta)] + \frac{10^2}{25^2}[F(22.5|\theta) - F(12.5|\theta)] + \frac{5^2}{25^2}[F(12.5|\theta) - F(7.5|\theta)]$.

Example 7.15. Fitting Weibull function to variable radius plot data. Assume that a sample plot of the Spanish NFI was established in the simulated forest stand of Example 7.8. The black histogram of Figure 7.15 shows the diameter distribution of the
35 sampled trees. The shape of the histogram is very different from the shape of the underlying diameter distribution, which is shown by the gray histogram.

ML-fit was performed by maximizing the likelihood \( \ell(\alpha, \beta) = \sum \ln g(d|\alpha, \beta) \), where function \( g \) is as specified in Equation (7.15). The obtained estimates for parameters \( \alpha \) and \( \beta \) were 3.93 and 21.11, respectively. The approximate 95% confidence intervals of parameters, (2.77 – 5.58) and (18.99 – 23.47), include the true values 3 and 20. The black thick line of figure 7.10 shows the fitted sampling distribution, and the thick gray line shows the corresponding distribution of the stand. The thin lines show the distributions by applying the true parameter values of 3 and 20.

**The R-code for Example 7.15**

```r
> dsample2 <- c(d[d<25|d>=42.5],
+ d[d<15&d>=22.5],
+ d[d<10&d>=12.5],
+ d[d<5&d>=7.5])
> dsample2
[33] 17.34299 12.62201 10.82117
>
> # Compute the ML-estimates for Weibull-parameters
> # cdf of left-truncated Weibull distribution
> # limit is the truncation diameter. defaults to 0, i.e. no truncation.
> pwtr<-function(x,shape,scale,limit=0) {
+ F<-(pweibull(x,shape,scale)-pweibull(limit,shape,scale))/(1-pweibull(limit,shape,scale))
+ F[x<limit]<-0
+ F
+ }
>
> # density of left-truncated Weibull distribution using varying-radius circular plots
> dwtrPPS<-function(x,shape,scale,limit=0) {
+ f<-dweibull(x,shape,scale)/(1-pweibull(limit,shape,scale))
+ f[x<limit]<-0
+ w<-rep(1,length(x))
+ w[x<=42.5]<-(15/25)^2
+ w[x<=22.5]<-(10/25)^2
+ w[x<=12.5]<-(5/25)^2
+ lwf<-(1-pwtr(42.5,shape,scale,limit=limit)) +
+ (15/25)^2*(pwtr(42.5,shape,scale,limit=limit)-
+ pwtr(22.5,shape,scale,limit=limit)) +
+ (10/25)^2*(pwtr(22.5,shape,scale,limit=limit)-
+ pwtr(12.5,shape,scale,limit=limit)) +
+ (5/25)^2*(pwtr(12.5,shape,scale,limit=limit)-
+ pwtr(7.5,shape,scale,limit=limit))
+ w=f/lwf
+ }
>
> # Negative log-likelihood, parameters in log scale
> nLLwtrPPS<-function(lshape,lscale) {
+ shape<-exp(lshape)
+ scale<-exp(lscale)
+ nL<--sum(log(dwtrPPS(dsample,shape=shape,scale=scale,limit=7.5)))
+ nL<-
+ nLL
+ }
>
> # ML-estimation
> est<-mle(nLLwtrPPS,start=list(lshape=log(4),lscale=log(20)))
> a<-exp(coef(est)[1])
> b<-exp(coef(est)[2])
> sd<--sqrt(diag(vcov(est)))
> 

```
Figure 7.11: The plot on the left shows the probability of a tree being observed, \( w(z) \). The plot on the right shows an example of an estimated distribution using simulated data. The histogram with thick lines shows the observed sample, and the histogram with thin lines shows all trees of the plot. The thin line shows the true underlying Weibull distribution and the thick line the estimated distribution.

Example 7.16. Overlapping crowns in a Poisson stand (Mehtätalo 2006) In an aerial forest inventory, crown radius \( Z \) describes tree size. It is assumed to follow a two-parameter Weibull distribution. We assume that a tree remains unobserved if the tip is within a crown of a larger tree. Otherwise, the tree is observed and crown area is determined correctly. It can be shown that if tree locations are random and tree size is not correlated within the stand, the probability for a tree being observed depends on...
crown radius according to
\[ w(z) = \pi \int_{0}^{\infty} t^2 f_Z(t|\alpha, \beta) dt \]
where \( \pi \) is the expected canopy closure, replaced with its observed value in applications.

The distribution of observed crown areas is
\[ f(w|z|\alpha, \beta) = \frac{w(z|\alpha, \beta)f_Z(z|\alpha, \beta)}{\int_{0}^{\infty} w(u|\alpha, \beta)f_Z(u|\alpha, \beta) du} \]

Parameters \( \alpha \) and \( \beta \) can be estimated by fitting the above distribution to the observed sample of crown radii. Stand density can then be estimated as \( \lambda = -\frac{\ln(\pi)}{\pi E(Z)} \).

Example 7.17. The volume-weighted distribution.

### 7.3.4 Scaling the weighted distributions

The basal-area weighted distribution is a consistent modeling approach when the utilized diameter data are collected with angle count sampling. This is because it can be used for specifying the proportion of basal area within a specified diameter class. With the unweighted diameter distribution, the proportion of the total number of stems can be specified easily.

For example, assuming class limits \( l \) and \( u \), the proportion of the total number of stems within that class is obtained as \( F_{N_D}^N(u) - F_{N_D}^N(l) \), where \( F_{N_D}^N(d) = \int_{-\infty}^{d} f_{N_D}^N(u) du \) is the cdf of the (unweighted) diameter distribution. The total number of stems within this diameter class is obtained by multiplying the proportion with the total number of stems, denoted by \( N \) to get
\[ N_{[l, u]} = N(F_{N_D}^N(u) - F_{N_D}^N(l)) \]  
(7.16)

In general, unweighted density, where the frequency expresses the proportion of a diameter class of the total number of stems is naturally scaled to the per hectar basis by multiplying by the total number of trees per ha.

Example 7.18. Scaling an unweighted distribution by \( N \). Assume that the unweighted diameter distribution follows the Weibull-distribution. In example 7.13 we got the estimates with parameters \( \hat{\alpha} = 3.11 \) and \( \hat{\beta} = 20.10 \). The circular plot with the radius of 10 meters included 27 trees, giving \( \hat{N} = 859 \) trees per ha. The estimated number of stems within diameter class \([16, 20]\), obtained using equation 7.16, is 204 trees per ha. If we assume that all trees of that class have are of the diameter 18 cm, the approximate basal area is 5.20 m\(^2\)/ha.
CHAPTER 7. MODELING TREE SIZE DISTRIBUTIONS

The proportion of the total basal area within a specified diameter class is obtained by replacing the unweighted distribution with the basal-area weighted distribution to get \( G_D(u) - G_D(l) \), where \( G_D(d) = \int_{-\infty}^{d} f_G(u) du \). The total basal area within this diameter class is obtained by multiplying the proportion with the total basal area \( G \).

We get

\[
G_{[l,u]} = G(G_D(u) - G_D(l)).
\] (7.17)

**Example 7.19. Scaling a basal-area weighted distribution by \( G \).** Assume that the basal-area weighted diameter distribution follows the Weibull-distribution. In example 7.12 we got the estimates \( \hat{\alpha} = 4.58 \) and \( \hat{\beta} = 23.81 \). The angle count sample with BAF 1 included 24 trees, thus giving \( \hat{G} = 24 \text{ m}^2 \text{ per ha} \). The estimated basal area within diameter class \([16, 20]\), obtained using equation 7.17, is \( 5.11 \text{ m}^2 \text{ per ha} \). If we assume that all trees of that class have are of the diameter 18 cm, the approximate number of stems is 201 trees per ha.

If angle count sampling is used in forest inventory, then the total growing stock is measured with the basal area. In this situation, it is natural to model the basal area weighted distribution instead of the unweighted distribution, because the scaling of the basal area weighted distribution is done with the basal area. In addition, another reason for favouring the basal-area weighted distribution is that fitting an assumed basal area weighted distribution to angle count sampling data is straightforward, because the weighting needs not to be taken into account in the fit (see Example 7.11). Similar justification can be given to the use of unweighted distributions when fixed-area sample plots are used. However, these justifications are technical, and might be also practically restrictive before the availability of efficient computers with easily applicable
numerical algorithms. However, nowadays these justifications may not be valid any more.

To define rules to scale the distribution with any measure of the growing stock, let us start with the definition of the expected value $E(g(X)) = \int_{-\infty}^{\infty} g(x)f(x)dx$. For example, the expected value of basal area is

$$\bar{G} = E(\frac{\pi}{4}d^2) = \int_{-\infty}^{\infty} \frac{\pi}{4}x^2f_ND(x)dx,$$

which is the mean basal area of the trees in the stand. Note that this expectation is in the denominator of the basal-area weighted diameter distribution (7.6); however, constant $\pi/4$ has been cancelled because it would occur both in the nominator and numerator of the weighted distribution.

However, another way to write the mean basal area is $\bar{G} = G/N$, which gives an equation to specify the relationship of $G$ and $N$ for a given diameter distribution as

$$\frac{G}{N} = \int_{-\infty}^{\infty} \frac{\pi}{4}d^2f_ND(x)dx.$$ (7.18)

This equation is useful in scaling the distribution. If we want to scale an unweighted diameter distribution by a known basal area, we can first use equation (7.18) to compute the number of stems in that stand as $N = \frac{G}{\int_{-\infty}^{\infty} \frac{\pi}{4}x^2f_ND(x)dx} = \frac{G}{E(\frac{\pi}{4}d^2)}$, and then to scale the distribution with that value.

The following result gives a rule for a general case. Let $t(x)$ define an increasing transformation of size, such as the basal area, volume, crown area etc. The following relationship holds

$$Tf_TX(x) = Nf_NX(t(x)),$$ (7.19)

where $T$ is the total of $t$. Dividing both sides by $N$ and rearranging terms gives

$$\frac{T}{N} = \frac{f_NX(t(x))}{f_TX(x)}$$

$$= \frac{f_NX(t(x)) \int_0^\infty f_NX(t(x))dx}{f_TX(x)t(x)}$$

$$= \int_0^\infty f_NX(t(x))dx$$

$$= E(t(X))$$

Equation (7.18) is maybe the most common application of this rule.

The total of $t$ within specified size limits $x_1$ and $x_2$ is, according to equation (7.19),

$$T_{[x_1,x_2]} = T \int_{x_1}^{x_2} f_TX(u)du = N \int_{x_1}^{x_2} f_NX(u)t(u)du .$$
Example 7.20. Assume the same sample that was used in example 7.18, but scale the distribution by the basal area. The basal area was first computed by summing the sample tree basal areas and dividing by the sample plot area. The resulting estimate was $\hat{G} = 24.84 \text{ m}^2/\text{ha}$. To compute the mean basal area, termed as \( G_{\text{per}N} \) in the code below, a function $\pi(d/400)^2 f(d|\hat{\alpha}, \hat{\beta})$ was integrated over its support, i.e., from 0 to $\infty$. The obtained average basal area was 0.285 \text{ m}^2. Dividing the total basal area by the mean basal area gave and estimate for the number of stems as $\hat{N} = 871$. This is the number of stems that is compatible with the estimated basal area and the diameter distribution. However, it differs from the number of stems in the previous example 7.18 (859 trees per ha), because the sample distribution is not exactly Weibull. The estimated number of stems in the diameter class $[16,20]$ is 207 trees per ha. Assuming all trees to be of diameter 18 cm gives the approximate class basal area of 5.27 trees per ha.

An accurate value for the class basal area was computed, too, by integrating the above defined function $\pi(d/400)^2 f(d|\hat{\alpha}, \hat{\beta})$ from 16 to 20 and multiplying by $\hat{N}$. The obtained value, 5.29 \text{ m}^2 per ha was slightly higher than the approximated value. If we divided the whole range of diameters into diameter classes and computed the accurate class-specific basal areas using integration, they would sum up to the estimated basal area of 24.83, which was used in the scaling of the distribution.

Finally, a note about the integral that was used to compute the average basal area is given. With the Weibull function, the integral can be expressed using the gamma function (Equation 7.6). The last rows of the code below show this alternative way of computing. The numerical values do not differ essentially. However, the general integral may be more applicable because it is not specific to the Weibull distribution only.

```r
> G <- sum(pi * (dsamplef/200)^2) / (pi * 0.1^2)
> G
[1] 24.83815
> gdfd <- function(d, shape, scale)
+ pi * (d/200)^2 * dweibull(d, shape, scale)
+ }
> GperN <- integrate(gdfd, 0, Inf, shape = aml2, scale = bml2)
> GperN
0.02850494 with absolute error < 1.9e-07
> N <- G/GperN$value
> N
[1] 871.3631
> Nclass <- N * (pweibull(20, aml2, bml2) - pweibull(16, aml2, bml2))
> Nclass
[1] 207.1451
> # The approximate class basal area
> Nclass * pi * (18/200)^2
[1] 5.271201
> # The exact class basal area (if the diameter distribution is correct)
> N * integrate(gdfd, 16, 20, shape = aml2, scale = bml2)$value
[1] 5.288232
> # Note: with the Weibull function, we could use also
> bml2^2 * gamma(2/aml2+1) * pi / 400
scale
0.02850494
7.3. COMPUTATIONS WITH DIAMETER DISTRIBUTIONS

In aerial forest inventory, for example, we may face with a situation where the scaling should be done with total volume instead of the number of stems or basal area. The same principle holds also for this situation. We just compute the average tree volume, and then compute the corresponding number of stems. However, for computing the average tree volume, a function to predict individual tree volume on diameter is needed. Alternatively, we may need stand-specific H-D curve and a function for tree volume on diameter and height.

**Example 7.21. Scaling unweighted diameter distribution by total volume.** A forest stand has been predicted to have the following H-D curve (see example 7.2)

\[ h = 1.3 + 25 \exp\left(-\frac{5}{d}\right) \]

and the parameters of the Weibull-distribution are predicted to be \( \alpha = 4 \) and \( \beta = 15 \). The total volume has been predicted to be 200 m\(^3\) per ha. All these predictions may be based, for example, on regression models that utilize remotely sensed data, such as laser data and aerial photographs, and are fitted to a sample of measured ground sample plots. The estimated volume may be more accurate than the estimated basal area or the number of stems would be. That is why it would be desirable to scale the predicted diameter distribution by total volume instead of a basal area or the number of stems.

Assume that the volume as a function of diameter and height is known, and is given by the volume function of Laasasenaho (1982) for Scots pine,

\[
v(d, h) = 0.0361d^{2.014}h^{0.997}\left(h - 1.3\right)^{-1.072}.
\]

Example 7.6 computed the mean tree volume as \( \bar{V} = 145.9 \) dm\(^3\). Thus, the total number of stems is \( V/\bar{V} = 1000 \times 200/145.89 \approx 1371 \) trees per ha. The predicted diameter distribution was scaled with this value. The total basal area was computed, too, as the integral \( G = N \int_0^\infty \pi u^2/40000 * f^N_D(u)du \). The obtained total basal area was 21.46 m\(^2\) per ha.

The number of stems, basal area and the total volume were computed for the diameter class \( d \in [16, 20] \). The obtained values were 317.5 trees per ha, 7.78 m\(^2\) per ha, and 75.6 m\(^3\) per ha. These values comprise 26, 36, and 38 % of the total \( N, G \) and \( V \), respectively.

**The R-code for Example 7.21**

```r
> VperN <- integrate(function(d) volume2(d) * dweibull(d, 4, 15), 0, Inf)
> VperN
145.8982 with absolute error < 0.0088
>
> V <- 200
>
> N <- 1000 * V/VperN$value
> N
```
A slightly more complicated situation arises when a weighted distribution is scaled using a different characteristic than the one used in weighting. For example, a basal area weighted distribution may need to be scaled with the total volume. Next, we derive equations for this particular situation: generalization to other situations can be done in a similar manner.

Recall that the unweighted distribution under an assumption of a basal-area weighted distribution is

\[ f_N(x) = \int_0^\infty u^{-2} f_G(x,u) \, du \]  \hspace{1cm} (7.20)

The mean tree basal area becomes now

\[ \frac{G}{N} = E(G) \]  \hspace{1cm} (7.21)

\[ \frac{G}{N} = \frac{\pi}{40000} \int_0^\infty x^2 f_D(x) \, dx \]

\[ = \frac{\pi}{40000} \int_0^\infty x^2 \left( \frac{1}{\int_0^\infty u^{-2} f_D(u) \, du} \right) \int_0^\infty f_G(x,u) \, dx \]

\[ = \frac{\pi}{40000} \int_0^\infty \frac{1}{u^{-2} f_D(u)} \int_0^\infty f_G(x,u) \, dx \]

Solving for \( N \) gives

\[ N = \frac{G}{40000} \frac{\pi}{\int_0^\infty u^{-2} f_D(u) \, du} \]  \hspace{1cm} (7.23)
The mean tree volume is, correspondingly to the mean tree basal area,

$$\frac{V}{N} = E(V)$$

$$= \bar{V}$$

$$= \int_0^\infty v(x)f_D^N(x)dx$$

$$= \int_0^\infty v(x)\frac{x^{-2}f_G^D(x)}{\int_0^\infty u^{-2}f_D^G(u)du}dx$$

$$= \frac{1}{\int_0^\infty u^{-2}f_D^G(u)du} \int_0^\infty v(x)x^{-2}f_G^D(x)dx,$$

which gives another expression for $N$ as

$$N = \frac{V \int_0^\infty u^{-2}f_D^G(u)du}{\int_0^\infty v(x)x^{-2}f_G^D(x)dx}.$$  \hfill (7.26)

Equating the two alternative expressions for $N$ (Equations 7.23 and 7.26) gives

$$\frac{V \int_0^\infty u^{-2}f_D^G(u)du}{\int_0^\infty v(x)x^{-2}f_G^D(x)dx} = G \frac{40000}{\pi} \int_0^\infty u^{-2}f_D^G(u)du.$$  

Canceling and rearranging terms we finally get the ratio of volume and basal area, that is, the mean tree form height, as

$$\frac{V}{G} = \frac{40000}{\pi} \int_0^\infty v(x)x^{-2}f_G^D(x)dx$$ \hfill (7.27)

To get the basal area that corresponds to the known volume $V$, the known volume is divided by the form height that is based on the known diameter distribution. The total number of stems can be obtained using either of the two above expressions. Class-specific values can be obtained by adjusting the limits of integration correspondingly.

**Example 7.22.** Scaling basal-area weighted diameter distribution by total volume.

For the forest stand of example 7.21, the basal-area weighted diameter distribution has been predicted to be of the Weibull form with parameters $\alpha = 5.2$ and $\beta = 17.3$. The volume and H-D curve are the same as in example 7.21. We need to compute the predicted basal area and number of stems by scaling the basal-area weighted diameter distribution by the predicted volume, and compute the number of stems, basal area, and volume of diameter class $[16,20].$

Using the equation (7.27), the mean form height ($V/G$) was computed numerically. The resulting value was 9365.42 $\text{dm}^3/\text{m}^2$. Dividing the total volume by the form height and multiplying by 1000 (to scale the total volume to $\text{dm}^3$ units) gave the total basal area of 21.35 $\text{m}^2$ per ha. Furthermore, the total number of stems was computed as $N = G \int_0^\infty \frac{1}{\pi x^2/40000} f_D^G(u)du \approx 1321$ trees per ha.
Figure 7.12: The unweighted, basal-area weighted, and volume-weighted densities of the diameter distribution when the unweighted distribution is $Weibull(4, 15)$ (black) and when the basal-area-weighted distribution is $Weibull(5.2, 17.3)$ (gray). The vertical lines show the diameter class $d \in [16, 20]$.
The diameter-class specific number of stems, basal area and volume were 341 trees per ha, 8.42 m$^2$ per ha, and 81.90 m$^3$ per ha. These values correspond to 26, 39 and 41% of the total number of stems, basal area, and volume, respectively. They are slightly higher than the proportions obtained in the previous example 7.21.

Figure 7.12 demonstrates the reasons for these differences by plotting the unweighted, basal-area weighted and volume-weighted versions of the both assumed distributions. The area of the diameter class [16, 20] is larger when the basal area weighted distribution is Weibull(5.2,17.3) than when the unweighted distribution is Weibull(4.15).

The R-code for Example 7.22

```r
> VperG<-integrate(function(d) 40000/pi*volume2(d)/d^2*dweibull(d,5.2,17.3),0,Inf) # dm^3/m^2
> VperG
9365.42 with absolute error < 0.18
> G=V/VperG$value*1000
> G
[1] 21.35516
> # The total number of stems
> N=0-G*integrate(function(d) 1/(pi*d^2/40000)*dweibull(d,5.2,17.3),0,Inf)$value
> N
[1] 1321.789
> V=G*integrate(function(d) volume2(d)/1000+1/(pi*d^2/40000)*dweibull(d,5.2,17.3),0,Inf)$value
> V
[1] 200
> Nclass=0-G*integrate(function(d) 1/(pi*d^2/40000)*dweibull(d,5.2,17.3),16,20)$value
> Nclass
[1] 340.8617
> Gclass=G*(pweibull(20,5.2,17.3)-pweibull(16,5.2,17.3))
> Gclass
[1] 8.421018
> Vclass=G*integrate(function(d) volume2(d)/1000+1/(pi*d^2/40000)*dweibull(d,5.2,17.3),16,20)$value
> Vclass
[1] 81.90166
> Nclass/N
[1] 0.2578790
> Gclass/G
[1] 0.3943318
> Vclass/V
[1] 0.4095083

The R-code for Figure 7.12

```
CHAPTER 7. MODELING TREE SIZE DISTRIBUTIONS

7.3.5 Utilizing the arithmetic relationships of stand variables

This subsection includes additional examples about utilizing the diameter distribution to compute stand variables, or on recovering the diameter distribution (and H-D curve in Example 7.27) that simultaneously fulfill given stand variables.

Function definitions for Examples 7.23 and 7.24

```r
# HD-curve
korfhd<-function(x,a=25,b=-5,c=1) {
  1.3+a*exp(b*x^(-c))
}

# Inverse of Korf curve
korfhd.inv<-function(y,a=25,b=-5,c=1) {
  (b/log((y-1.3)/a)))^(1/c)
}

# Unweighted distribution when basal-area weighted distribution is Weibull.
# Unscaled, i.e., integrates up to the total number of stems
fNw<-function(x,alpha,beta) {
  10000*dweibull(x,alpha,beta)/(pi*x^2/4)
}

# the corresponding distribution function, obtained using numerical integration
FNw<-function(x,alpha,beta) {
  if (x<=0) {
    value<-0
  } else {
    value<-integrate(f=function(u) fNw(u,alpha,beta),lower=0,upper=x)$value
  }
  value
}

# The volume model of Laasasenaho for spruce
vLaas<-function(x,h) {
  0.022927*x^1.915*0.99146^x *h^2.825*(h-1.3)^(-1.535)
}

# The volume weighted density when basal area weighted density is Weibull
# Integrates up to the total volume
fVw<-function(x,alpha,beta,a,b) {
  fNw(x,alpha,beta)*vLaas(x,korfhd(x,a,b))
}

# The corresponding distribution function
FVw<-function(x,alpha,beta,a,b) {
  if (x<=0) {
    value<-0
  } else {
    value<-integrate(f=function(u) fVw(u,alpha,beta,a,b),lower=1,upper=x)$value
  }
  value
}

# The height-weighted density, sums up to the total height
fHw<-function(x,alpha,beta,a,b) {
  fNw(x,alpha,beta)*korfhd(x,a,b)
}

# The corresponding distribution function with numerical integration.
FHw<-function(x,alpha,beta,a,b) {
  if (x<=0) {
    value<-0
  } else {
    value<-integrate(f=function(u) fHw(u,alpha,beta,a,b),lower=0,upper=x)$value
  }
  value
}

# An up-and-down algorithm
Figure 7.13: Figure for example 7.23. Diameter distributions weighted by basal area, number of stems and total volume. The dashed lines demonstrate the effect of harvesting 50% of the stems from below on total volume and basal area.

```r
# Solves fn(x)=0 for x between l (lower bound) and u (upper bound).
updown<-function(l,u,fn,crit=6) {
  fnu<-fn(u)
  fnl<-fn(l)
  if (fnl*fnu>0) return(NA)
  value<-fn((u+l)/2)
  while (round(value,crit)!=0) {
    if (fnu*value>0) {
      u<-(u+l)/2
      fnu<-value
    } else {
      l<-(u+l)/2
      fnl<-value
      value<-fn((u+l)/2)
    }
    nollakohta<-(l+u)/2
    nollakohta
  }
}

# Newton raphson algorithm for n multidimensional functions using numeric differentiation
NRNum2<-function(init,fnlist,crit=6) {
  par<-init
  sapply(par,function(x) cat(x," "))
  cat("n")
  value<-sapply(fnlist,function(f) f(par))
  j<-1
  while (round(sum(abs(value)),crit)!=0&j<10000) {
    grad<-t(sapply(fnlist,function(f) attributes(numericDeriv(quote(f(par)),c("par")))$gradient))
    deltax<-solve(grad,-value)
    par<-par+deltax
    sapply(par,function(x) cat(x," "))
    cat("n")
    value<-sapply(fnlist,function(f) f(par))
    j<-j+1
    if (j<10000) {
      list(par=par,value=value)
    } else {
      list(par=NA,value=NA)
    }
  }
}
```
Example 7.23 (The effect of a harvest on total volume). Consider a forest stand where \( F^G_D \) is Weibull with \( \alpha = 4 \) and \( \beta = 15 \) and \( G = 20 \text{m}^2/\text{ha} \), heights come from the Korf curve
\[
h(d) = 1.3 + a \exp \left( \frac{b}{d} \right),
\]
with \( a = 25 \) and \( b = -5 \), and the volume function is that of Laasasenaho (1982),
\[
v(d, h) = 0.022927d^{1.915}0.99146h^{2.825}(h - 1.3)^{-1.535}.
\]
Half of stems will be harvested from below, and the aim is to compute the remaining volume and basal area.

The unweighted diameter distribution fulfills
\[
N f^N_D (d) = 40000G/\pi d^{-2} f^G_D (d)
\]
Solving for \( N \) gives the total number of stems as \( N = \frac{40000G}{\pi} \int_0^\infty d^{-2} f^G_D (u) du = 2006.007 \).

To compute the cutting diameter limit, we need to compute the median diameter of the unweighted distribution \( f^N_D (d) \). It is obtained by solving \( F^N_D (d) = 0.50 \) for \( d \).

We see that the diameter limit for harvest is \( d = 10.36 \). The distribution of volume is
\[
V_f^V_D (d) = 40000G/\pi f^G_D (d)d^{-2}v(d, h(d)).
\]

The remaining volume and basal area are
\[
V_{\text{remaining}} = \int_{10.36}^\infty 40000G/\pi f^G_D (u) u^{-2} v(u, h(u)) du = 161.9796 \text{m}^3/\text{ha}
\]
and
\[
G_{\text{remaining}} = G \int_{10.36}^\infty f^G_D (u) du = 15.93095 \text{m}^2/\text{ha}.
\]

The R-code for Example 7.23

```r
> # The parameters of H-D curve
> a<-25
> b<-5
> # The parameters of the weibull distribution
> alpha<-4
> beta<-15
> # basal area
> G<-20
> # The total number of stems for G=20.
> H<-G*FNw(100,alpha,beta)
> N
> [1] 2006.007
> # The diameter limit for thinning
> Hraja<updown(0,100,function(x) G*FNw(x,alpha,beta)-0.50*N)
> Hraja
> [1] 10.35909
> # Total volume
> Vtot<20*FVw(100,alpha,beta,a,b)/1000
> Vtot
> [1] 195.8865
```
7.3. **Computations with Diameter Distributions**

> # The volume of unharvested trees
> Vjaa<-20*(FVw(100,alpha,beta,a,b)/1000-FVw(Hraja,alpha,beta,a,b)/1000)
> Vjaa
> [1] 161.9796

> # The basal area of unharvested trees
> Gjaa<-20*(pweibull(100,alpha,beta)-pweibull(Hraja,alpha,beta))
> Gjaa
> [1] 15.93095

Example 7.24: (Recovery of basal-area weighted Weibull parameters for known $G$, $N$, and $DGM$). Assuming the basal-area weighted diameter distribution to be of the Weibull form, solve the Weibull parameters that fulfill the estimated values of $G$, $N$, and $DGM$ based on the angle count sample of Example 7.9.

Based on the angle count sample of example 7.9, the estimated values were $G = 24$, $DGM = 22.99$ and $N = 765$.

Assuming a basal-area weighted distribution $f_D^G(x)$, the unweighted density is $f_D(x) = x^{-2} f_D^G(x|\alpha,\beta)$. The mean tree basal area is the expectation $\bar{g} = \frac{G}{N} = \int_0^\infty g(x) f_D^G(x|\alpha,\beta) du$. For a given measured basal area $G_{\text{meas}}$, the computational number of stems $N_{\text{comp}}$ is
obtained by dividing the measured basal area by the mean tree basal area as

\[ N_{\text{comp}} = \frac{G_{\text{meas}}}{\int_{0}^{\infty} g(x) f_{D}^{N}(x|\alpha, \beta) \, dx}. \]

It is a function of parameters \( \alpha \) and \( \beta \). The solved parameter values should fulfill

\[ N_{\text{comp}} = N_{\text{meas}} \]

where \( N_{\text{meas}} \) is the measured number of stems. Thus, we get a recovery equation

\[ \frac{G_{\text{meas}}}{\int_{0}^{\infty} g(x) f(x, \alpha, \beta)} - N_{\text{meas}} = 0. \]

The computational basal-area median diameter is the 0.5th quantile of the basal-area weighted diameter distribution. It should equal to the measured DGM, which gives another recovery equation.

\[ F_{D}^{G_{-1}}(0.5) - DGM = 0, \]

where \( F_{D}^{G_{-1}} \) is the quantile function of the Weibull cdf.

The resulting system of two equations can be solved numerically, e.g., by using the two-dimensional Newton-Raphson algorithm, that was used before for moment-based recovery of Weibull parameters (see Example 7.10). The two recovery equations were written into a list of equations, which is given as a parameter to the \texttt{NRnum2} -function.
7.3. COMPUTATIONS WITH DIAMETER DISTRIBUTIONS

The resulting parameter estimates were $\hat{\alpha} = 4.73$ and $\hat{\beta} = 24.83$. Figure 7.14 shows the recovered distribution and the histogram of the angle-count sample.

The R-code for Example 7.24

```r
# Unweighted distribution when basal-area weighted distribution is Weibull. # Scaled, i.e., integrates to one
dNW<-function(x,alpha,beta) {
  fn<-function(x) dweibull(x,shape=alpha,scale=beta)/(x^2)
  denom<-integrate(fn,lower=0,upper=Inf)$value
  fn(x)/denom
}

# the corresponding distribution function, obtained using numerical integration
PNW<-function(x,alpha,beta) {
  if (x<=0) {
    value<-0
  } else {
    value<-integrate(f=function(u) dNW(u,alpha,beta),lower=0,upper=x)$value
  }
  value
}

G<-length(dsample)
G
[1] 24
GDM<-mean(dsample[order(dsample)][12:13])
GDM
[1] 22.98554
N<-sum(1/(pi*(dsample/2)^2))*10000
N
[1] 764.6316

f1<-function(theta) {
  GperN<-integrate(function(x) pi *x^2/40000 *dNW(x,theta[1],theta[2]),0,Inf)$value
  G/GperN-N
}

f2<-function(theta) qweibull(0.5,theta[1],theta[2])-DGM

fn<-list(f1,f2)
print(theta<-NRnum2(c(4,DGM),fn))
4 22.98554
4.311644 25.02708
4.661807 24.85211
4.730045 24.83701
4.732122 24.83658
4.732124 24.83658

$par
[1] 4.732124 24.836578

$value
[1] 1.311946e-10 -1.989520e-13

alpha<-theta$par[1]
beta<-theta$par[2]

pdf("figdd8.pdf",width=7,height=6)
hist(dsample,freq=FALSE)
x<-seq(0,40,0.1)
lines(x,dweibull(x,alpha,beta))
dev.off()

Example 7.25 (PRM using volume, mean height and basal area). Assume that $F^G_D$ is a two-parameter Weibull distribution. Recover such values for $\alpha$ and $\beta$ that yield $V=200$ m$^3$/ha, $G=20$m$^2$/ha and $\bar{H}=16$ m. The volume function and H-D curve are as specified in Example 7.23.
The parameter values should simultaneously fulfill the following two equations:

\[
\frac{40000G}{\pi} \int_0^\infty f_G(u|\alpha, \beta)u^{-2}v(u, h(u))du - V = 0
\]

\[
\frac{40000G}{\pi} \int_0^\infty f_G(u|\alpha, \beta)u^{-2}h(u)du - \bar{H} = 0
\]

The system was solved using the same numerical algorithm as before. The obtained solution was \(\alpha = 3.80\) and \(\beta = 17.07\).

**The R-code for Example 7.25**

```r
> V<-200
> G<-20
> H<-16
> a<-25
> b<-5
> f1<-function(theta) G*FVw(1000,theta[1],theta[2],a,b)/1000-V
> f2<-function(theta) FHw(1000,theta[1],theta[2],a,b)/FNw(1000,theta[1],theta[2])-H
> fn<-list(f1,f2)
> NRnum2(c(4,15),fn)
```

```
4 15
3.8417 16.64206
3.802815 17.05015
3.800977 17.07243
3.800971 17.07249

$par
[1] 3.800971 17.072494

$value
[1] -8.323582e-10 -1.052527e-10
```

**Example 7.26 (Example 7.24 continued).** Take a closer look to the situation of Example 7.24. We assume that \(G, N, DGM\) and mean height are known. When \(f_G(d)\) is Weibull density, the system of equations on Example 7.24 simplifies to (Mehtätilo and Nyblom, unpublished)

\[
\beta = \frac{DGM}{\ln 2^{1/\alpha}}
\]

\[
\frac{\pi}{4\Gamma(1-2/\alpha)(\ln 2)^{2/\alpha}} = \frac{G}{N \cdot DGM^2},
\]

where \(\Gamma()\) is the gamma function. These equations lead to a simpler numerical solution, where the shape parameter can be first solved using the lower equation, and the scale parameter can then be solved from the lower equation. The numerical values of the estimates are the same as in the earlier example.

**The R-code for Example 7.26**

```r
> # Solve unidimensional equation using Newton-Rapson algorithm
> # init=Initial guess,
> # fn=function,
> # gr=gradient
> # crit= convergence criteria
```
Example 7.27 (Simultaneous recovery of the H-D curve and diameter distribution (Mehtätalo et al. 2007)). Assume that a forest inventory has been carried out by using ALS, and the following estimates for the number of stems, total volume, basal-area median diameter and height of basal-area median tree have been obtained for a Scots pine sample plot. In addition, a ground based survey has been conducted, including callipering all trees of the stand and measuring one tree for height. The ALS- and ground-based estimates for the characteristics were tabulated below.

<table>
<thead>
<tr>
<th>Variable</th>
<th>ALS</th>
<th>Ground</th>
</tr>
</thead>
<tbody>
<tr>
<td>N, trees per ha</td>
<td>473.41</td>
<td>747</td>
</tr>
<tr>
<td>DGM, cm</td>
<td>28.69</td>
<td>25.60</td>
</tr>
<tr>
<td>HGM, m</td>
<td>20.60</td>
<td>20.70</td>
</tr>
<tr>
<td>V, m² per ha</td>
<td>248.60</td>
<td>288.02</td>
</tr>
</tbody>
</table>

Let us assume that diameter follows the Weibull-distribution. Furthermore, assume that the stand-specific H-D curve is of the form

\[ h(d|A) = \max(1.4, \exp(A - By(d))) , \]
Function $y(d)$ is the following transformation of tree diameter

$$y(d) = \frac{(d + 7)^{\alpha} - (DGM + 17)^{\alpha}}{17^{\alpha} - 37^{\alpha}},$$

where $c = 0.98 + 0.058DGM$. The shape parameter of the H-D curve was predicted on $DGM$ using $B = 0.62 - 0.027DGM + 0.00094 + DGM^2$ (Mehtätalo 2005b). The volume for a given diameter and height was assumed to be given by the model of Laasasenaho (1982) (see Example 7.21).

The recovery is based on three equations that set the measured values of $V$, $DGM$, and $HGM$ equal to the corresponding values based on the diameter distribution and H-D curve.

$$\begin{cases} f_V(\alpha, \beta, A|N_{meas}) - V_{meas} = 0 \\ f_{DGM}(\alpha, \beta, A) - DGM_{meas} = 0 \\ f_{HGM}(\alpha, \beta, A) - HGM_{meas} = 0 \end{cases} \quad (7.28)$$

The essential task is to define the functions $f_V$, $f_{DGM}$ and $f_{HGM}$. The function for the total volume is

$$f_V(\alpha, \beta, A|N_{meas}) = N_{meas} \int_0^\infty f_D^N(u|\alpha, \beta) v(u, h(d|A)) du,$$

and the function for median tree height is

$$f_{HGM}(\alpha, \beta, A) = h(DGM_{meas}|A).$$

The function for basal area median diameter is

$$f_{DGM} = F_D^{-1}(0.5|\alpha, \beta).$$
where
\[ F_D^G = \frac{\int_0^\infty u^2 f_D^N(u|\alpha, \beta) du}{\int_0^{\infty} u^2 f_D^N(u|\alpha, \beta) du} \]
is the cdf of the basal-area weighted diameter distribution and \( f_N^D(u|\alpha, \beta) \) is the assumed Weibull density of the unweighted distribution.

The specified system of equations can be solved numerically. A simple but computationally inefficient way is to minimize function
\[ (f_V(\alpha, \beta, A|N_{meas}) - V_{meas})^2 + (f_DGM(\alpha, \beta, A) - DGM_{meas})^2 + (f_HGM(\alpha, \beta, A) - HGM_{meas})^2 \]
with respect to parameters \( \alpha, \beta \) and \( A \). The obtained values provide the solution if the value of the function is 0 at the solution. The function will find estimates even with quite poor starting values; we used \( A = 3 \), \( \alpha = 5 \), and \( \beta = DGM \). Function recov2 is an implementation of this approach.

Another alternative would be to use the three-dimensional Newton-Raphson algorithm, which is implemented in function recov below. This is a quick algorithm, but needs rather good initial estimates. Finally, function recov3 is an implementation where the initial estimates are first searched for using recov2 and the final estimates are then obtained using recov.

Using the first initial estimates, function recov gives an error message. However, function recov2 is able to do the estimation. Finally, also recov will converge if the estimates from recov2 are given as the initial estimates. The final estimates are \( \hat{\alpha} = 4.872 \), \( \hat{\beta} = 28.16 \), and \( \hat{A} = 3.073 \). They differ slightly from the initial estimates obtained using recov2. The estimates based on ground measurements are \( \tilde{\alpha} = 3.519 \), \( \tilde{\beta} = 24.024 \), and \( \tilde{A} = 3.091 \).

Figure 7.15 shows the observed diameter distribution of the plot, as well as the tree heights, predicted using Näslunds height model and the one observed height from the plot. Because ALS overestimated \( DGM \) the recovered distribution is located to the right from the true distribution. The H-D curve underestimates the true height because \( HGM \) was underestimated. The shape of the distribution is quite ok, because the ratio of \( N \) and \( V \) was quite well estimated, even though both these characteristics were underestimates. The distribution based on ground-based values fits well to the observed data, but the shape of the H-D curve differs much from the “true” heights, which are based on the Näslunds curve, which was used in generalizing the sample tree height to tally trees.

The R-code for Example 7.27

```
> N<-473.41
> DGM<-28.69
> HGM<-20.60
> V<-248.60
> A<-3
> Nt<-747
```
CHAPTER 7. MODELING TREE SIZE DISTRIBUTIONS

```r
DGMt<-25.60
HGMt<-20.70
Vt<-288.02

recov(N,DGM,HGM,V)
Error in drop(.Call("La_dgesv", a, as.matrix(b), tol, PACKAGE = "base")): 
  Lapack routine dgesv: system is exactly singular
recov2(N,DGM,HGM,V)
attr(,"value")
[1] 0.0001629323

est<-recov3(N,DGM,HGM,V)
est
attr(,"value")
[1] 4.017425e-07

estt<-recov3(Nt,DGMt,HGMt,Vt)
> estt
attr(,"value")
[1] 1.563342e-06

25.9,29.6,30.6,31.1,35.6)
hoot<-c(8.00,14.30,16.08,16.36,16.52,16.57,16.62,16.67,16.73,16.83,16.93,16.93,17.29,17.34, +
18.01,20.70,19.27,20.67,21.02,21.18,22.55)

pdf("figdd9.pdf",width=7,height=6)
par(mfcol=c(1,1),mai=c(1,1,0.5,0.1))
hist(puut,freq=FALSE,xlim=c(0,41),ylim=c(0,0.22),xlab="diameter, cm")
points(puut,hoot/100,col="red")
x<-seq(0,40,0.1)
lines(x,dweibull(x,est[1],est[2]),col="blue",lwd=2)
lines(x,korfhd(x,est[3],DGM)/100,lwd=2,col="blue")
lines(x,dweibull(x,estt[1],estt[2]),col="blue",lwd=2,lty="dashed")
lines(x,korfhd(x,estt[3],DGM.tod)/100,lwd=2,col="blue",lty="dashed")

sapply(c(0,0.05,0.10,0.15,0.2),function(x) lines(c(40,41),rep(x,2)))
text(rep(40,5),c(0,0.05,0.10,0.15,0.2),seq(0,20,5),pos=4)
mtext("Height, m",side=4)
dev.off()
```

**Function definitions for example 7.27**

```r
# Parameter-recovery for Weibull distribution and asymptote of the height curve 
# assuming DGM, HGM, N and V to be known.
updown<-function(l,u,fn,crit=6) {
  fnu<-fn(u)
  fnl<-fn(l)
  if (fnl*fnu>0) return(NA)
  value<-fn((u+l)/2)
  while (round(value,crit)!=0&(u-l)>10ˆ(-crit)) {
    if (fnu*value>0) {
      u<-(u+l)/2
      fnu<-value
    } else {
      l<-(u+l)/2
      fnl<-value
    }
    value<-fn((u+l)/2)
  }
  nollakohta<-(l+u)/2
  nollakohta
}

# Newton raphson algorithm for n multidimensional functions using numeric differentiation
NRnum2<-function(init,fnlist,crit=6) {
  par<-init
  ```
7.3. COMPUTATIONS WITH DIAMETER DISTRIBUTIONS

```r
# sapply(par,function(x) cat(x," "))
value<-sapply(fnlist,function(f) f(par))
j<-1
while (round(sum(abs(value)),crit)!=0&j<100) {
  cat(j,"
  grad<-t(sapply(fnlist,function(f) attributes(numericDeriv(quote(f(par)),c("par")))$gradient))
  deltax<-solve(grad,-value)
  par<-par+deltax
  value<-sapply(fnlist,function(f) f(par))
  j<-j+1
  if (j<10000) {
    list(par=par,value=value)
  } else {
    list(par=NA,value=NA)
  }
}

# Reparameterize tree diameter for the height model of Mehtatalo (2005)
dexf<-function(x,DGM) {
  lda<-7
  c<-0.98227850+0.05753439*DGM
  ((x+lda)^(-c)-(DGM+10+lda)^(-c))/((10+lda)^(-c)-(30+lda)^(-c))
}

# Predict height for trees with given diameters x.
# Uses a given scale parameter A for the H-D curve.
# The shape parameter (B) is predicted using the model of Mehtatalo (2005)
korfhd<-function(x,A,DGM) {
  B<-0.6156-0.02707*DGM+0.000935*DGM^2
  pmax(1.4,exp(A-B*dexf(x,DGM)))
}

# predict volume for a given diameter and scale parameter of the height curve
volf<-function(x,A,DGM) {
  h<-korfhd(x,A,DGM)
  pars<-c(0.036089,2.01395,0.99676,2.07025,-1.07209)
}

# basal-area weighted distribution when unweighted is weibull.
# Provides a scaled distribution that sums up to total basal area
fG<-function(x,shape,scale,N) {
  N*pi/4000*dweibull(x,shape,scale)*x^2
}

# CDF corresponding to fG.
# scaled so that FG(Inf)=G
FG<-function(x,shape,scale,N) {
  sapply(x,function(y) integrate(function(u) fG(u,shape,scale,N),0,y)$value)
}

# The median of fG,
# i.e solution to FG=0.5
DGMf<-function(shape,scale,N) {
  G<-FG(qweibull(1-1e-10,shape,scale),shape,scale,N)
  updown(l=qweibull(0.01,shape,scale),
        u=qweibull(0.99,shape,scale),
        fn=function(u) FG(u,shape,scale,N)-0.5*G,
        crit=10)
}

# The height of a DGM-tree, meters
HGMf<-function(shape,scale,N,DGM) {
  korfhd(DGMf(shape,scale,N),A,DGM)
}

# The distribution weighted by volume
fv<-function(x,shape,scale,A,DGM,N) {
  H/1000*dweibull(x,shape,scale)*volf(x,A,DGM)
}
```
# The cdf corresponding to fV

\[
F_V\left( x, \text{shape}, \text{scale}, A, \text{DGM}, N \right) = \text{integrate} \left( f_V(u, \text{shape}, \text{scale}, A, \text{DGM}, N), y \right)
\]

# recovery using the NR-algorithm

\[
\text{recov}\left( N, \text{DGM}, HGM, V, \text{start} = c(5, 15, 3) \right) = \exp(\text{NRnum2}(\text{theta}, \text{fn}) \text{par})
\]

# Recovery using R optim, Nelder-Mead, slow but stable, does not necessarily yield the solution

\[
\text{recov2}\left( N, \text{DGM}, HGM, V, \text{start} = c(5, 15, 3) \right) = \text{exp(optim(\text{log(start)}, \text{fn2}, \text{par})$value)}
\]

# Final estimates using NR, after first computing start values using R-implementation of Nelder-Mead.

\[
\text{recov3}\left( N, \text{DGM}, HGM, V, \text{start} = c(5, \text{DGM}, 3) \right) = \text{rep(NA, 3)}
\]

7.4 Modeling diameter distributions for prediction

7.4.1 Parameter prediction method

A very common situation is that no diameter sample has been measured in the field, but some stand characteristics, such as stand age, site fertility, basal area and mean diameter have been assessed instead. Some of these are directly related to diameter distribution (e.g., basal area and mean diameter), whereas others are related only indirectly (e.g., stand age and site fertility). In this case, parameter prediction method (PPM) and parameter recovery method (PRM) are the alternatives for diameter distribution prediction.

In PPM, an estimated regression relationship between the measured stand characteristics and parameters of an assumed diameter distribution function is utilized in prediction. This regression is estimated from data including measured diameter distributions from a representative sample of stands. According to Cao (2004), the earliest reference to this method is Clutter and Bennett (1965). However, already Cajanus (1914) modeled the parameters of his Gram-Carlier series approximation (that is, the
7.4. MODELING DIAMETER DISTRIBUTIONS FOR PREDICTION

first four moments) on stand characteristics. However, his aim was not to predict the distribution but to classify the stands according to the estimated regression into fertility classes.

The parameter recovery method utilizes the direct relationships between stand characteristics and diameter distribution to solve the parameters from a system of equations (Hyink and Moser 1983). For the system of equations to be feasible, the number of directly related characteristics needs to be \( k + 1 \), where \( k \) is the number of parameters to be estimated. Examples of such PRM method were presented in examples 7.24, 7.25, and 7.27.

However, a more common approach for the parameter recovery is to first predict some directly related characteristics, such as moments or percentiles, using a PPM-type approach for obtaining a sufficient number of directly related characteristics. In this situation, the PRM includes just recovery of the parameters that fulfill the predicted percentiles or moments. Examples of such recovery situations have already been dealt with in examples 1.40 and 7.10.

An assumption about a theoretical distribution is needed in PPM and PRM. Of the several utilized distribution functions, the Weibull function is probably the most commonly used one. It is also the only widely used two-parameter alternative, even though it also has been used in the 3-parameter form for the prediction of diameter distributions. Other convenient and widely used alternatives are beta, Johnson’s SB and the logit-logistic distribution (Wang and Rennolls 2005), all having four parameters. The percentile-based diameter distribution has been found a very flexible model for the PPM method (Borders et al. 1987, Maltamo et al. 2000, Kangas and Maltamo 2000b, Kangas et al. 2007). The common practice in forest inventory is to measure only trees larger than a fixed minimum diameter. With such data, a truncated version of an assumed distribution should be used instead (Palahí et al. 2007).

In the PPM method, a dataset is needed for estimating the PPM models. The dataset should include samples of callipered trees from several stands. From each stand, the sample should be large enough for estimating the parameters of the assumed diameter distribution. However, the number of trees per stand varies a lot among the published studies. For example, Palahí et al. (2006) used all plots that had more than 5 pine trees, Kangas and Maltamo (2000b) used stands with minimum of 15 trees per species, and Magnussen (1986) had more than 100 trees per stand. Another important property of the dataset is the variation in the stand characteristics. The data should cover the natural range of the characteristics that are used as predictors in the model, and the number of stands should be large enough for estimating the regression.

The estimation of the PPM models includes three steps:
CHAPTER 7. MODELING TREE SIZE DISTRIBUTIONS

i Select the model for diameter distribution

ii Fit an assumed functional to the empirical diameter distributions for each stand of
the modeling data and save the stand-specific estimates.

iii Estimate a regression to explain the variation of the fitted parameter values by stand
characteristics, such as stand age, basal area, mean diameter, or site characteristics.

The estimated regression can then be used to predict the parameters of the diameter
distribution for a stand where only the predictors of the model have been measured.

Step i: selecting the diameter distribution model

Different models have been used, including the normal (2 parameters) lognormal (2),
Weibull (2 or 3), beta (4), Johnson’s SB (4). The most recent proposal as the distribu-
tion is the logit-logistic distribution (4 parms) (Wang and Rennolls 2005), which can
accommodate a wider range of sceness-curtois combinations than any of the previ-
ously used models (see Figure 7.16).

Kangas and Maltamo (2000b), Kangas et al. (2007) used the percentile-based method
which has been proposed by Borders et al. (1987). Even though the percentile-based
function has been termed as “distribution-free”, it can be naturally treated as an ap-
plication of the PPM method. In that method, the fitting of distribution by ML is just
replaced by computing the sample percentiles for each stand. In addition, if the inter-
polation between percentiles is carried out by the linear interpolation, the distribution
can be seen as a finite mixture of uniform distributions or, alternatively, as a segmented
uniform distribution (See example 1.7). The number of parameters is the number of
segments + 1.

The most studies have assumed the specified model for the unweighted distribution.
However, if the data collection is based on angle count sampling it may be more natural
to assume a specified model for the basal-area weighted distribution (see Example 7.11).
Since (Kilkki and Päivinen 1986), this approach has a long tradition in Finland where
also the NFI has been for long based on angle count sampling.

For comparison of alternative models, several test statistics or performance criteria
have been used. If the fit has been performed using the method of maximum like-
lihood, a natural selection as the criteria is a statistic that is based on the computed
likelihood. The simplest alternative is just the value of the log likelihood at the solu-
tion, \( \ell(\hat{\theta}) \) (or the negative log likelihood) which is the higher (lower) the better the fit.
However a problem with this criteria is that it does not take into account the number of
parameters. The value of log-likelihood necessarily increases whenever a new pa-
rameter is included in the model. That is why, for example, it would always favour
Figure 7.16: The marks of the upper graph show the skew ($\beta_1 = \mu_3^2 / \mu_2^3$) and kurtosis ($\beta_2 = \mu_4 / \mu_2^2$ where $\mu_k$ is the $k$th central moment) combinations of in the data of Chinese Fir plantations (Wang and Rennolls 2005). The lines demonstrate the range of possible combinations for different diameter distribution models. The lower graph shows skew on kurtosis for distributions that are equally able to represent skew to the left and right. Adopted from Wang and Rennolls (2005).
the three-parameter Weibull model over the two-parameter model. For this reason, the
likelihood itself should be used for comparison only between models with an equal
number of parameters.

The Akaike and Bayesian Information criteria (AIC and BIC) are alternatives that
are based on the log likelihood plus a penalty, which is a function of the number of
parameters. The Akaike Information Criterion is defined as

$$AIC = 2k - 2\ell(\hat{\theta})$$

and the Bayesian Information Criterion is

$$BIC = k \ln(n) - 2\ell(\hat{\theta}),$$

where \(k\) is the number of parameters and \(n\) is the number of observations.

Reynolds et al. (1988) suggested the so-called error index for comparing two di-
ameter distributions. It just computes the sum of absolute differences of diameter class
frequencies:

$$e_i^k = \sum \| f - \hat{f} \|.$$ The value depends on the utilized diameter class width and sample size. The above
version of the index uses the densities that sum up the one. Another option would
be to scale the index with the number of stems, so that the index would sum up the
differences in diameter class-specific values of the number of stems. In addition, the
number of stems could be replaced with basal area or volume to give more weight to
the right end of the distribution.

The Kolmogorov-Smirnov test has also been widely used to test whether a sample
follows a specified continuous distribution function. For the test, the empirical distri-
bution function \(F_n\) for \(n\) i.i.d. observations \(X_i\) is defined as

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{X_i \leq x}$$

where \(I_{X_i \leq x}\) is the indicator function, equal to 1 if \(X_i \leq x\) and equal to 0 other-
wise. The null hypothesis of the test is, that the data were sampled from a continuous
distribution with cdf \(F(x)\). The test statistic is

$$D_n = \sup_x |F_n(x) - F(x)|$$

where \(\sup S\) is the supremum (maximum) of set \(S\). This test statistic follows the Kol-
mogorov distribution (see details and references from http://en.wikipedia.org/wiki/Kolmogorov-
Smirnov_test). In R, function \text{ks.test} of package \text{stats4} can be used to perform
the test.

A problem with all these criteria is that they are data-specific. Thus, they can be
used for comparison only when different distributions are fitted to the same dataset.
Example 7.28. Two Scots pine sample plots of were measured from Spain. Figure 7.17 shows the empirical diameter distributions of the plots. A total of 136 trees were calipered on plot 1 and 128 trees on plot 2. The aim is to select the best-fitting theoretical model for the diameter distribution with these two plots. The alternatives considered are the two- and three-parameter Weibull-distributions, Jonson’s SB distribution and the logit-logistic distribution. The comparisons are based on log-likelihood, AIC, error index with no weighting and basal-area weighting, and Kolmogorov-Smirnov test statistic.

The four alternative distributions were fitted to the dataset using the method of maximum likelihood. The lines in figure 7.17 show the graphs of the fitted functions, and tables 7.1 and 7.2 the values of the different goodness-of-fit criteria. In stand 1, the Weibull-distributions fit better than the four-parameter alternatives. The error indices and Kolmogorov-Smirnov test-statistic favour 2-parameter Weibull, whereas AIC favours the 3-parameter Weibull. The log-likelihood should be used only for comparing models with an equal number of parameters.

In stand 2, all criteria support the selection of logit-logistic distribution, and rank the 2-parameter Weibull distribution as the worst. The Kolmogorov-Smirnov -tests rejects the null hypothesis about the underlying distribution only in the case of 2-parameter Weibull distribution of stand 2.

The R-code for Example 7.28

```r
> pdf("figdd12.pdf",width=7,height=3)
> par(mfcol=c(1,2),mai=c(0.7,0.7,0.1,0.1),cex=0.7)
> result<-c()
> for (i in 1:2) {
```
CHAPTER 7. MODELING TREE SIZE DISTRIBUTIONS

Table 7.1: The fit statistics for plot 1 of example 7.28. **Boldface** indicates the best fit and *italics* the worst fit.

<table>
<thead>
<tr>
<th>Dist</th>
<th>logLik</th>
<th>AIC</th>
<th>ei0</th>
<th>ei2</th>
<th>K-S statistic</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weib2</td>
<td>-414.1</td>
<td>832.3</td>
<td>0.136</td>
<td>0.232</td>
<td>0.033</td>
<td>0.998</td>
</tr>
<tr>
<td>Weib3</td>
<td>-411.7</td>
<td>829.3</td>
<td>0.176</td>
<td>0.286</td>
<td>0.042</td>
<td>0.971</td>
</tr>
<tr>
<td>SB</td>
<td>-412.7</td>
<td>833.5</td>
<td>0.219</td>
<td>0.331</td>
<td>0.049</td>
<td>0.895</td>
</tr>
<tr>
<td>logit-logistic</td>
<td>-412.5</td>
<td>833.1</td>
<td>0.208</td>
<td>0.335</td>
<td>0.054</td>
<td>0.828</td>
</tr>
</tbody>
</table>

Table 7.2: The fit statistics for plot 2 of example 7.28. **Boldface** indicates the best fit and *italics* the worst fit.

<table>
<thead>
<tr>
<th>Dist</th>
<th>logLik</th>
<th>AIC</th>
<th>ei0</th>
<th>ei2</th>
<th>K-S statistic</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weib2</td>
<td>-413.6</td>
<td>831.2</td>
<td>0.369</td>
<td>0.388</td>
<td>0.127</td>
<td>0.032</td>
</tr>
<tr>
<td>Weib3</td>
<td>-404.9</td>
<td>815.8</td>
<td>0.248</td>
<td>0.339</td>
<td>0.087</td>
<td>0.293</td>
</tr>
<tr>
<td>SB</td>
<td>-404.1</td>
<td>816.2</td>
<td>0.224</td>
<td>0.305</td>
<td>0.078</td>
<td>0.416</td>
</tr>
<tr>
<td>logit-logistic</td>
<td>-403.1</td>
<td>814.3</td>
<td>0.220</td>
<td>0.278</td>
<td>0.075</td>
<td>0.465</td>
</tr>
</tbody>
</table>

```R
+ d<-dlist[[1]]
+ estw2<-fitw2(d)
+ estw3<-fitw3(d)
+ estsb<-fitsb(d)
+ estll<-fitll(d)
+ hist(d,freq=FALSE,ylim=c(0,0.08))
+ x<-seq(0,40,0.1)
+ lines(x,dweibull(x,coef(estw2)[1],coef(estw2)[2]), lty="dashed")
+ lines(x,dweib3(x,coef(estw3)[1],coef(estw3)[2],coef(estw3)[3]))
+ lines(x,dsb(x,coef(estsb)[1],coef(estsb)[2],coef(estsb)[3],coef(estsb)[4]), lwd=2, lty="dashed")
+ lines(x,dll(x,coef(estll)[1],coef(estll)[2],coef(estll)[3],coef(estll)[4]), lwd=2)
+ ksw2<-ks.test(d, "pweibull", coef(estw2)[1],coef(estw2)[2])
+ ksw3<-ks.test(d, "pweib3", coef(estw3)[1],coef(estw3)[2],coef(estw3)[3])
+ kssb<-ks.test(d, "psb", coef(estsb)[1],coef(estsb)[2],coef(estsb)[3],coef(estsb)[4])
+ kssl<-ks.test(d, "pll", coef(estll)[1],coef(estll)[2],coef(estll)[3],coef(estll)[4])
+ result<-rbind(result,
+ data.frame(plot=rep(i,4),
+ dist=c("Weib2","Weib3","SB","logit-logistic"),
+ logLik=c(logLik(estw2),logLik(estw3), logLik(estsb), logLik(estll)),
+ AIC=c(AIC(estw2),AIC(estw3), AIC(estsb), AIC(estll)),
+ ei0=c(errind(d,cdf=function(x) pweibull(x,coef(estw2)[1],coef(estw2)[2]),
+ power=0,dlim=c(-20,80),cwidth=2),
+ errind(d,cdf=function(x) pweib3(x,coef(estw3)[1],coef(estw3)[2],coef(estw3)[3]),
+ power=2,dlim=c(-20,80),cwidth=2),
+ errind(d,cdf=function(x) psb(x,coef(estsb)[1],coef(estsb)[2],coef(estsb)[3],coef(estsb)[4]),
+ power=0,dlim=c(-20,80),cwidth=2),
+ errind(d,cdf=function(x) pll(x,coef(estll)[1],coef(estll)[2],coef(estll)[3],coef(estll)[4]),
+ power=0,dlim=c(-20,80),cwidth=2)),
+ a1=c(errind(d,cdf=function(x) pweibull(x,coef(estw2)[1],coef(estw2)[2]),
+ power=0,dlim=c(-20,80),cwidth=2),
+ errind(d,cdf=function(x) pweib3(x,coef(estw3)[1],coef(estw3)[2],coef(estw3)[3]),
+ power=2,dlim=c(-20,80),cwidth=2),
+ errind(d,cdf=function(x) psb(x,coef(estsb)[1],coef(estsb)[2],coef(estsb)[3],coef(estsb)[4]),
+ power=2,dlim=c(-20,80),cwidth=2),
+ errind(d,cdf=function(x) pll(x,coef(estll)[1],coef(estll)[2],coef(estll)[3],coef(estll)[4]),
+ power=2,dlim=c(-20,80),cwidth=2)),
+ statKS=c(Ksw2$statistic,Ksw3$statistic,Kssb$statistic,Kssl$statistic),
+ pKS=c(Ksw2$p,Ksw3$p,Kssb$p,Kssl$p))
+ )
```
7.4. MODELING DIAMETER DISTRIBUTIONS FOR PREDICTION

There were 26 warnings (use warnings() to see them)
> dev.off()
windows
> result
plot dist logLik AIC e10 e12 statKS pKS
1 2 Weib2 -414.1385 832.2771 0.1363902 0.2315458 0.089825047
2 2 Weib3 -411.6572 829.3144 0.1759859 0.2861932 0.08713106
3 1 SB -412.7491 833.4982 0.2185992 0.3312558 0.08942912
4 1 logit-logistic -412.5310 833.0619 0.2082713 0.3275452 0.08942912
5 2 Weib2 -413.6102 831.2204 0.3691010 0.3883123 0.03194008
6 2 Weib3 -404.9069 815.8139 0.2476581 0.3394893 0.08653259 0.29321277
7 2 SB -404.1032 816.2064 0.2236968 0.3047821 0.08048993 0.41634886
8 2 logit-logistic -403.1430 818.2859 0.2200697 0.2775154 0.07516906 0.46465042

Functions for example 7.28

# Reynolds error index
# d includes true diameters, cdf is a function returning the unweighted cdf
# power gives the weights (2 gives BA weight, 0 the unweighted),
# dim the range of diameters.
# cwidth the class width (make sure dim is a multiple of cwidth).
errind<-function(d,cdf=function(x) pweibull(x,5,20),power=0,dim=c(-20,80),cwidth=2) {
  limits<-seq(dim[1],dim[2],cwidth)
  n<-length(limits)-1
  means<-(limits[-1]+limits[-(n+1)])/2
  dp<-d^power
  sumdp<-sum(dp)
  ftrue<-sapply(1:n,function(i) sum(dp[d>=limits[i]&d<limits[i+1]]))/sumdp
  Fpred<-cdf(limits)
  fpred<-Fpred[-1]-Fpred[-n]
  fpred<-fpred*means^power/sum(fpred*means^power)
  sum(abs(fpred-ftrue))
}

# 2 parameter Weibull -logL
nLLweibull<-function(x, shape=5, scale=20) {
  -sum(dweibull(x,shape=shape,scale=scale,log=TRUE))
}

# Fits two-parameter weibull distribution to tree diameter data using MLE starting from values 5 and 20
# for shape and scale, respectively
fitweib2<-function(d) {
  est<-mle(function(shape=5, scale=20) nLLweibull(d,shape, scale))
  if (class(est) == "try-error") list(par=rep(NA,2),neg2LL=NA,conv=NA)
  else est
}

# 3-parametric Weibull density
dweib3<-function(x, shape=5, scale=20, loc=0) {
  dweibull(x-loc, shape, scale)
}

# 3-parametric Weibull density
pweib3<-function(x, shape=5, scale=20, loc=0) {
  pweibull(x-loc, shape, scale)
}

# 3 parameter weibull -logL
nLLweib3<-function(x, shape=5, scale=20, loc=0) {
  -sum(dweibull(x-loc,shape=shape,scale=scale,log=TRUE))
}

# Fits two-parameter weibull distribution to tree diameter data using MLE
# starting from values 5 and 20
# for shape and scale, respectively
fitweib3<-function(d) {
  est<-mle(function(shape=5, scale=20, loc=0) nLLweib3(d,shape, scale, loc))
  if (class(est) == "try-error") list(par=rep(NA,3),neg2LL=NA,conv=NA)
  else est
}
# Density of the logit-logistic distribution of Rennolls and Wang.
# ksi and lambda >0, lambda > ksi
# ksi=minimum, lambda=maximum, sigma controls curtosis and myy skewness.
dll<-function(x, lambda, sigma, ksi, myy) {
  sigma <- exp(sigma)/(1+exp(sigma))
  f <- (lambda-ksi)/sigma * 1/((x-ksi)/(lambda-x)) * 1/(exp(-myy/sigma)*((x-ksi)/(lambda-x))^(-1/sigma)+exp(myy/sigma)*((x-ksi)/(lambda-x))^(-1/sigma)+2)
  if (x <= ksi | x >= lambda) f <- 0
  f
}

# negative log likelihood
nLLll<-function(x, lambda, sigma, ksi, myy) {
  value <- sum(log(dll(x, lambda, sigma, ksi, myy)))
  if (value == Inf) value <- 10^16
  value
}

# derivatives of neg ll
# theta=c(lambda, sigma, ksi, myy)
gradnLLll<-function(x, theta) {
  l <- theta[1]
  s <- theta[2]
  k <- theta[3]
  m <- theta[4]
  awful <- exp(-m/s)*((x-k)/(l-x))^((1-s)/s)+exp(m/s)*((l-x)/(x-k))^((1-s)/s)+2
  c(
    # with respect to lambda
    sum(-1/(l-k)+1/(l-x)+
        1/awful*
        (-exp(-m/s)*(1/s)*((x-k)/(l-x))^(1/s)+exp(m/s)*(1/s)*((l-x)/(x-k))^(1/s))),
    # with respect to sigma
    sum(1/s+
        1/awful*
        (-exp(-m)*((x-k)/(l-x))^((1/s))*(-m+log((x-k)/(l-x)))*s^(-2)+
        exp(m)*((l-x)/(x-k))^((1/s))*(m+log((l-x)/(x-k)))*s^(-2))),
    # with respect to ksi
    sum(1/(l-k)+1/(x-k)+
        1/awful*
        (-exp(-m)*((x-k)/(l-x))^((1/s)+(-m+log((x-k)/(l-x)))*s^(-2))+
        exp(m)*((x-k)/(l-x))^((1/s)))*s^(-2))),
    # with respect to myy
    sum(1/awful*
        (((x-k)/(l-x))^((1/s)+exp(-m/s)*(-1/s)+
          (x-k)/(l-x))^((-1/s)*exp(m/s)*(-1/s)))))
  )
}

# cdf of logit-logistic distribution
pll<-function(x, lambda, sigma, ksi, myy) {
  sigma <- exp(sigma)/(1+exp(sigma))
  F <- 1/(1+exp(myy/sigma)*((x-ksi)/(lambda-x))^(-1/sigma))
  F[x <= ksi] <- 0
  F[x >= lambda] <- 1
  F
}

# Another function for the same purpose
# sigma is bounded using logit transformation
fitll<-function(d, start=NA) {
  ll<-function(lambda, sigma, ksi, myy) nLLll(d, lambda, sigma, ksi, myy)
  if (is.na(start)) start <- list(lambda=max(d)+5, sigma=0, ksi=max(0, min(d)-2), myy=0)
  est <- try(mle(ll, method="L-BFGS-B", lower=c(max(d)+0.1,-Inf, 0, -5), upper=c(45, 3.5, max(0, min(d)-0.1), 5), control=list(maxit=200)))
  est
}
7.4. MODELING DIAMETER DISTRIBUTIONS FOR PREDICTION

```r
if (class(est) == "try-error") list(par = rep(NA, 4), neg2LL = NA, conv = NA) else est
#
# Rennolls&Wang parameterization of Johnsons SB distribution
#
# lambdamaximum, not range
#
density
dsb<-function(x,ksi,lambda,delta,gamma) {
value<-rep(NA,length(x))
value[x>ksi&x<lambda]<-1/(delta*sqrt(2*pi))*(lambda-ksi)/
((lambda-x[x>ksi&x<lambda])* 
(x[x>ksi&x<lambda]-ksi)*
exp(-1/2*((log((x[x>ksi&x<lambda]-ksi)/
(lambda-x[x>ksi&x<lambda]))-gamma)/delta)ˆ2)
value[x<=ksi|x>=lambda]<-0
value
}
#
# c.d.f
p sb<-function(x,ksi,lambda,delta,gamma) {
  y<-rep(NA,length(x))
y[x<=ksi]<-0
y[x>lambda]<-1
y[x>ksi&x<lambda]<-sapply(x[x>ksi&x<lambda],
  function(y) integrate(function(x)
  dsb(x,ksi,lambda,delta,gamma),ksi,y)$value)
y
}
#
# negative log-likelihood
nLLsb<-function(x,ksi,lambda,delta,gamma) {
  value<-sum(log(dsb(x,ksi,lambda,delta,gamma)))
  if (value==Inf) value<-10ˆ16
  cat(ksi,lambda,delta,gamma,value,"\n")
  value
}
#
# gradients of the negative log-likelihood
gradnLLsb<-function(x,theta) {
k<-theta[1]
l<-theta[2]
d<-theta[3]
g<-theta[4]
big<-(log((x-k)/(l-x))-g)/d
  c(
    # with respect to ksi
    sum(1/(l-k)-1/(x-k)-big/d/(x-k)),
    # with respect to lambda
    sum(-1/(l-k)+1/(l-x)-big/(d*(l-x))/remove$d)
  , # with respect to lambda
  sum(-big/d)
    
}
#
# a function for fitting Johnsons SB distribution to tree diameter data using ML.
fitsb<-function(d) {
d<-d
ll<-function(ksi,lambda,delta,gamma) nLLsb(d,ksi,lambda,delta,gamma)
start<-list(ksi=max(0,min(d)-2),lambda=max(d)+5,delta=1,gamma=0)
grad<-function(theta) gradnLLsb(d,theta)
est<-try(mle(ll,gr=grad,
  start=start,
  method="L-BFGS-B",
  lower=c(0,max(0,min(d)-2),min(d)+5,0.01,0.1,-10),
  upper=c(max(0,min(d)-0.1),Inf,Inf,10),
  control=list(maxit=500)))
  if (class(est) == "try-error") list(par = rep(NA, 4), neg2LL = NA, conv = NA) else est
}
```
CHAPTER 7. MODELING TREE SIZE DISTRIBUTIONS

Step ii: fitting the model for plot data

The basic alternatives for fitting the distribution were presented in section 1.5, the most common ones being maybe the methods of Maximum likelihood and moments. The ML- estimation for the distributions used in forest sciences has been problematic earlier, which may be one reason for the use of the simpler methods. However, this is not the case any more, and that is why the method of ML may currently be the most commonly used. It is suggested because of its good statistical properties, such as the asymptotic efficiency (Casella and Berger 2002, p. 472).

In addition to the fitting method, it is important to recognize and take into account the sampling method in the fitting. If an equal-probability sampling design is used (i.e., fixed-area plots), then the sampling distribution is just the assumed distribution function. However, if the sampling uses Probability Proportional to Size designs (e.g., angle-count sampling, variable-radius plots, or a fixed minimum diameter), then the effect of these sampling designs on the sampling distribution need to be taken into account. This is done by writing the sampling distribution of tree diameters under the assumed distribution models. Then the sampling distribution is fitted to the data instead of the assumed distribution itself. Details and examples on these fitting methods were given in sections 7.3.2 and 7.3.3.

Assuming the density of the sampling distribution of plot $k$ to be of form $g(d_{ki}|\alpha, \lambda)$ (e.g., the Weibull density; selection was done in step i) the log-likelihood of stand $k$ is

$$
\ell_k(\alpha_k, \lambda_k) = \sum_{i=1}^{n_k} \ln g(d_{ki}|\alpha_k, \lambda_k),
$$

(7.29)

In step ii, such values for $\alpha_k$ and $\beta_k$ are searched for each stand $k$ that maximize the likelihood.

Step iii: Modeling the parameters on stand variables

In step (iii), the most common and widely used method for regression is to fit a linear model, where the parameters are as the response variable and some stand variables are as predictors. The models are usually fitted individually, even though the model is actually a model system with a seemingly unrelated model structure (Robinson 2004). Another approach for modeling could be the nearest neighbour approaches (k-NN, MSN or k-MSN) (Haara et al. 1997, Maltamo and Kangas 1998, Packalén and Maltamo 2008). In the NN or MSN approaches, the distribution of the most similar sample plot of a reference data is used as the predicted distribution.

Formally, using the estimates from step $ii$ for the distribution parameters (e.g., $\hat{\alpha}_k$...
7.4. MODELING DIAMETER DISTRIBUTIONS FOR PREDICTION

and \( \lambda_k \), the models of the following form are estimated

\[
\hat{\alpha}_k = f(T_k, G_k, DGM_k, \ldots | \beta_{\alpha,0}, \beta_{\alpha,1}, \beta_{\alpha,2}, \ldots) + e_k \tag{7.30}
\]

\[
\hat{\lambda}_k = f(T_k, G_k, DGM_k, \ldots | \beta_{\lambda,0}, \beta_{\lambda,1}, \beta_{\lambda,2}, \ldots) + e_k. \tag{7.31}
\]

These models, where \( T_k, G_k, DGM_k, \ldots \) are just examples of potential predictors, are estimated from a dataset where each sample plot (or stand) provides a single observation.

The R-code for Data reading routines for example 7.29.

```r
> # Read the datasets.
> # The original data files include on plot per file so that
> # The first two rows include plot-specific characteristics
> # and the next rows include tree-specific measurements.
> # This script merges the first rows of each file into plotdata (called plotdata)
> # and the tree-specific measurements of the files into treedata (called treedata).
> # The first rows of file m01
> # 400.00 300.00 1008.33 20.11 154.17 19.58 15.66 69.11 18.75 100.00 .00 1.00
> # 124.00 -1.00 38.00 298.00 30.40 20.19 80.35 .00 29.34 12.59 -1.00
> # 6.00 -1.00 102.00 138.00 29.30 19.87 78.49 .00 27.53 11.96 -1.00
> # 163.00 -1.00 123.00 43.00 28.60 19.65 75.09 .00 24.20 11.32 12.78
> # 58.00 1.00 117.00 163.00 28.30 19.30 84.00 35.60 24.00 3.28 3.88
> # 49.00 1.00 248.00 190.00 26.10 19.70 67.00 32.70 24.00 9.72 13.84
> # 53.00 -1.00 188.00 250.00 25.30 18.54 80.35 .00 28.23 6.12 6.48
> # 19.00 1.00 270.00 199.00 26.60 18.20 66.00 34.80 30.00 7.96 12.56
>
> plotdata<-cbind(read.table("c:/laurim/aineistoja/spati/m01.dat",nrows=1),
+ read.table("c:/laurim/aineistoja/spati/m01.dat",skip=1,nrows=1))

# Remove other tree species and useless columns
```

```r
> names(plotdata)<-c("X","Y","N","G","V","DGM","Hg","Hdom","pineprop","spruceprop","plot_id")
```

```r
> plotdata<-plotdata[-c(8,12,15,21,27,28,44),] # Remove plots that are known to have a bimodal distribution
```

```r
> names(treedata)<-c("plot_id","tree_id","species","xc","yc","d","h","dstump","2b","id1","id2")
```

```r
> treedata$species<-abs(treedata$species) # Negative tree species means tally tree, positive a sample tree
```

```r
> treedata<-treedata[puudata$pl==1,,]
```

```r
> head(treedata)
```

```
plot_id tree_id species xc yc d h
1 1 124 1 38 298 30.4 20.19
2 1 6 1 102 138 29.3 19.87
3 1 163 1 123 43 28.6 19.65
```
```
Figure 7.18: An example graph produced by the fitting routine of example 7.29. The plot shows the empirical and fitted diameter distributions of plots 25-30 of the modeling dataset, as well as the KS-test statistics about the goodness of fit.

Example 7.29 (PPM for North Carelian Scots Pine data). In this example, the aim is to fit PPM models for the parameters of the two-parameter Weibull distribution. The dataset includes diameter measurements from 59 sample plots in North Carelia, Finland. The dataset is included in two data frames the first one is a plot-specific data of stand characters and the second one includes tree-specific data (see the code above for the rather complicated reading routine of these data from separate plot-specific files).

The first step of the analysis would be selection of the distribution family, using, e.g., the criteria of example 7.28. However, we skip this stage and use the two-parameter Weibull distribution.

The next step is to fit the Weibull distribution to each plot. We use the maximum likelihood see 1.5.2. The following R-code performs the ML-fit for each plot of plotdata in a loop. In the same loop, the treedata is used also for computing stand characteristics $DGM$, $N$ and $G$, as well as the first two moments of the diameter distribution.
Furthermore, the code plots a histogram of tree diameters for each plot and adds the fitted Weibull distribution on the plot (See figure 7.18 for an example).

```r
# 2 parameter Weibull -logL
nLLweibull <- function(x, shape=5, scale=20) {
  -sum(dweibull(x,shape=shape,scale=scale,log=TRUE))
}

# Fits the two-parameter weibull distribution to tree diameter data using MLE starting from values 5 and 20
fitw2 <- function(d) {
  est <- mle(function(shape=5, scale=20) nLLweibull(d,shape, scale))
  if (class(est) == "try-error") list(par=rep(NA,2),neg2LL=NA, conv=NA)
  else est
}

# Compute DGM, G, N, V, moments, and ML-estimates of the Weibull distribution
DGM <- G <- N <- V <- shape <- scale <- pvec <- ex <- ex2 <- c()

treedat <- c()
windows()
par(mfcol=c(2,3))
for (i in 1:dim(plotdata)[1]) {
  trees <- treedata[treedata$plot_id==plotdata$plot_id[i],]
  temp <- dmean(puut$d, ala=(plotdata$X * plotdata$Y/100)[i])
  DGM <- c(DGM, temp$DGM)
  G <- c(G, temp$G)
  N <- c(N, temp$N)
  V <- c(V, sum(ennusta.tilavuus(trees$pl, trees$d, m=1))/(plotdata$X * plotdata$Y/1000)[i])
  d <- trees$d
  est <- fitw2(d)
  shape <- c(shape, attributes(est)$details$par[1])
  scale <- c(scale, attributes(est)$details$par[2])
  pvec <- c(pvec, p ks.test(d, "pweibull", coef(est)[1], coef(est)[2])$p
  hist(d, main=paste("Plot ",i," p=",round(p,3)), freq=FALSE)
  lines(x, dweibull(x, shape[i], scale[i]))
  if (!i%%6) { # open a new window at every 6th round
    windows()
    par(mfcol=c(2,3))
  }
  ex <- c(ex, mean(d))
  ex2 <- c(ex2, mean(d^2))
  treedata$DGM[treedata$plot_id==plotdata$plot_id[i]] <- temp$DGM
  treedata$G[treedata$plot_id==plotdata$plot_id[i]] <- temp$G
  treedata$N[treedata$plot_id==plotdata$plot_id[i]] <- temp$N
  treedata$V[treedata$plot_id==plotdata$plot_id[i]] <- temp$V
  treedata$Hg[treedata$plot_id==plotdata$plot_id[i]] <- plotdata$Hg[i]
  treedata$Tg[treedata$plot_id==plotdata$plot_id[i]] <- plotdata$Tg[i]
  treedata$Hdom[treedata$plot_id==plotdata$plot_id[i]] <- plotdata$Hdom[i]
}

There were 50 or more warnings (use warnings() to see the first 50)

treedata$SN <- N
treedata$DG <- G
treedata$V <- V
treedata$ex <- ex
treedata$ex2 <- ex2

treedata <- cbind(treedata, shape, pvec)
head(treedata)
X Y N G V DGM Hg Tg Hdom pineprop spruceprop
1 400 300 1333.333 4.168069 120.2899 7.6 15.66 69.11 18.75 100 0
2 500 300 1066.667 3.334455 139.6050 7.6 15.47 62.49 16.79 100 0
```
The next step of the analysis is to model the Weibull-parameters on stand characteristics. The stand characteristics used in models should be such that they can be measured on the field quickly with sufficient accuracy, or are known in advance (e.g., the age of a plantation). In this example, the potential predictors are

- DGM, the basal area median diameter
- G, the basal area
- N, the number of stems
- Tg, the age of basal area median tree
- Hgm, the height of the basal area median tree

Of these characteristics, the number of stems could be criticized because its ocular assessment includes lot of errors (Kangas et al. 2004). However, using DGM, G
and \( N \) together in prediction improves the prediction of the shape of the diameter distribution a lot, because these three characteristics together include such information on the shape of the distribution that cannot be obtained by two of them alone (Siipilehto 1999). We started the modelling from the scale parameter. Figure 7.19 shows the estimated scale parameter on different stand characteristics. \( DGM \) seems to have the strongest relationship with scale, and this relationship seems to be linear. However, the variation in scale seems to increase with increasing \( DGM \), so a variance function that allows heteroscedasticity could be applicable. In R, function \( \texttt{gls} \) can be used for fitting a linear model with heteroscedastic residual. A widely applicable model for non-constant residual variance is the power function

\[
\text{var}(\epsilon_k) = \sigma^2 |\tilde{y}_k|^{2\delta}
\]

which allows both increasing and decreasing variances as a function of prediction \( \tilde{y}_k \).

Models with and without variance function were fitted and compared graphically and by using the likelihood ratio test. The test suggested the model with variance function.

Figure 7.20 shows some diagnostic plots of that model. The thin lines demonstrate the variation of standardized residuals in ten classes of the \( x \)-axis, each class including approximately equal number of observations. The thick lines demonstrate the confidence interval for the class mean (i.e., clas mean \( \pm 1.96 \times \text{s.e.} \) of the mean). They can be used to detect trends in residuals as a function of potential predictors. The top left

Figure 7.20: The standardized residuals of the PPM model for scale on the predicted value and the other potential predictors.
graph does not indicate heteroscedasticity of the residuals. However, one could speculate of a slight underestimation of the model for highest \( DGM \). The other plots of the figure do not show any clear trend in the residuals with the predictor. Thus, we accept the model

\[
\hat{\beta}_k = a + b DGM_k + \epsilon_k
\]

where the parameter estimates are \( \hat{a} = -0.377 \) and \( \hat{b} = 0.915 \). For the variance function \( \hat{\delta} = 1.31 \) and \( \hat{\sigma} = 0.0407 \). The code below was used to perform the analysis.

```r
> library(nlme)
> scm1<-gls(scale~DGM,data=plotdata)
> scm2<-gls(scale~DGM,data=plotdata,weights=varPower(0.5,~fitted(.)))
> anova(scm1,scm2)

             Model df      AIC      BIC    logLik   Test  L.Ratio p-value
scm1         1   3 229.4084 235.5375 -111.7042
scm2         2   4 210.0966 218.2688 -101.0483 1 vs 2  21.31175  <.0001
```

```r
mywhiskers<-function(x,y, nclass=10, limits=quantile(x,seq(0,1,0.1)),
add=FALSE, se=TRUE, main="", xlab="x", ylab="y",
ylim=NA, lwd=1, highlight="red" ) {
  away<-is.na(x+y)
x<-x[!away]
y<-y[!away]
  if (is.na(limits[1]))
    limits<-seq(min(x),max(x)+1e-10,length=nclass+1)
  else
    nclass=length(limits)-1
  means<-sapply(1:nclass,function(i) mean(y[x>=limits[i]&x<limits[i+1]])
    if (se) {
      ses<-sapply(1:nclass,function(i) sd(y[x>=limits[i]&x<limits[i+1]])/
        sqrt(sum(x>=limits[i]&x<limits[i+1])))
    } else {
      ses<-sapply(1:nclass,function(i) sd(y[x>=limits[i]&x<limits[i+1]])
    }
    lb<-means-1.96*ses
    ub<-means+1.96*ses
    xclass<-1/2*(limits[-1]+limits[-nclass-1])
  if (add) {
    points(xclass,means)
  } else {
    if(is.na(ylim[1])) ylim<-c(min(lb),max(ub))
    plot(xclass,means,ylim=ylim,main=main,xlab=xlab,ylab=ylab,xlim=range(x))
  }
  color<-rep("black",nclass)
  color[ub*lb>0]<-highlight
  sapply(1:nclass,function(i) lines(xclass[c(i,i)],c(lb[i],ub[i]),lwd=lwd,col=color[i]))
}
```

```r
> windows()
> par(mfcol=c(2,3))
> plot(predict(scm2),resid(scm2,type="p"),ylim=c(-4,3))
> mywhiskers(predict(scm2),resid(scm2,type="p"),so=TRUE, lwd=2,add=TRUE)
> mywhiskers(predict(scm2),resid(scm2,type="p"),so=FALSE,add=TRUE)
> abline(0,0)
> plot(plotdata$G,resid(scm2,type="p"))
> mywhiskers(plotdata$G,resid(scm2,type="p"),so=TRUE, lwd=2,add=TRUE)
> abline(h=0)
> plot(plotdata$N,resid(scm2,type="p"))
> mywhiskers(plotdata$N,resid(scm2,type="p"),so=TRUE, lwd=2,add=TRUE)
> abline(h=0)
> plot(plotdata$Hg,resid(scm2,type="p"))
```
7.4. MODELING DIAMETER DISTRIBUTIONS FOR PREDICTION

Figure 7.21: The ML-estimates of the shape parameter on the potential predictors.

The next step is to find a model for the shape parameter. Figure 7.21 shows the estimates on the predictors, as well as on the shape index transformation $S_k = \frac{G_k}{DGM_{k\theta N_k}}$.
Figure 7.22: The standardized residuals of model shm4 against prediction and potential additional predictors. The thick lines show the predicted density and the thin lines the ML- fit (same as in figure 7.18)

(Siipilehto 1999). The shape index seems to be the best primary predictor. However, the trend is not linear, and the plot indicates increasing residual variance with respect to $S_h$. Furthermore, also $DGM$ was evaluated as an additional predictor, but it did not improve the model significantly. The final model was the following nonlinear model

$$\hat{\alpha}_k = \frac{1}{c + d S_h} + c_k,$$

with parameter estimates $\hat{c} = 0.972$ and $\hat{d} = -11069$. The parameter estimates for the variance function of the power form were $\hat{\delta} = 2.202$ and $\hat{\sigma} = 0.0278$. Figure 7.22 shows that the standardized residuals of the model have quite constant variance, and there are no clearly detectable trends in the residuals with respect to the shape index or any potential additional predictor. The R-code below was used to model shape.

```r
> shm1<-gls(I(1/shape) ~ DGM+shind, data=plotdata)
> shm2<-nls(shape~1/(a+b*DGM+c*shind),
+ start=list(a=coef(shm1)[1],b=coef(shm1)[2],c=coef(shm1)[3]),
+ data=plotdata)
> shm3<-gls(shape~1/(a+b*DGM+c*shind),
+ start=c(a=coef(shm1)[1],b=coef(shm1)[2],c=coef(shm1)[3]),
+ params=list(a+b+c~1),
+ data=plotdata,
+ weights=varPower(0.5, ~fitted(.)))
> shm4<-gls(shape~1/(a+c*shind),
+ start=c(a=coef(shm1)[1],c=coef(shm1)[3]),
+ weights=varPower(0.5, ~fitted(.)))
```
7.4. MODELING DIAMETER DISTRIBUTIONS FOR PREDICTION

+ params=list(a+c~1),
+ data=plotdata,
+ weights=varPower(0.5, ~fitted(.)))
>
> shm5<–gnls(shape ~ 1/(a + c * shind),
+ start=c(a=coef(shm1)[1], c=coef(shm1)[3]),
+ params=list(a+c~1),
+ data=plotdata)
>
> anova(shm3,shm4)

<table>
<thead>
<tr>
<th>Model</th>
<th>df</th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>Test</th>
<th>L.Ratio</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>shm3</td>
<td>1</td>
<td>45.86822</td>
<td>56.25591</td>
<td>-17.93411</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>shm4</td>
<td>2</td>
<td>44.21623</td>
<td>52.52638</td>
<td>-18.10811</td>
<td>1 vs 2</td>
<td>0.3480067</td>
<td>0.5552</td>
</tr>
</tbody>
</table>

> anova(shm4,shm5)

<table>
<thead>
<tr>
<th>Model</th>
<th>df</th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>Test</th>
<th>L.Ratio</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>shm4</td>
<td>1</td>
<td>44.21623</td>
<td>52.52638</td>
<td>-18.10811</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>shm5</td>
<td>2</td>
<td>87.89948</td>
<td>94.13210</td>
<td>-40.94974</td>
<td>1 vs 2</td>
<td>45.68325</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

> summary(shm4)

Generalized nonlinear least squares fit
Model: shape ~ 1/(a + c * shind)
Data: plotdata

Variance function:
Structure: Power of variance covariate
Formula: ~fitted(.)

Parameter estimates:

<table>
<thead>
<tr>
<th>Value</th>
<th>Std.Error</th>
<th>t-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.972</td>
<td>31.67223</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>-11068.957</td>
<td>-20.60434</td>
<td>0</td>
</tr>
</tbody>
</table>

Correlation:

<table>
<thead>
<tr>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.989</td>
<td></td>
</tr>
</tbody>
</table>

Standardized residuals:

<table>
<thead>
<tr>
<th>Min</th>
<th>Q1</th>
<th>Med</th>
<th>Q3</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.09051899</td>
<td>-0.72980214</td>
<td>0.73946232</td>
<td>0.72302894</td>
<td>2.17588723</td>
</tr>
</tbody>
</table>

Residual standard error: 0.02775859 Degrees of freedom: 59 total; 57 residual

windows()

par(mfcol=c(2,3))

> plot(predict(shm4), resid(shm4, type="p"), ylim=c(-3,3))
> mywhiskers(predict(shm4), resid(shm4, type="p"), se=TRUE, lwd=2, add=TRUE)
> mywhiskers(predict(shm4), resid(shm4, type="p"), se=FALSE, add=TRUE)
> abline(0,0)

> plot(plotdata$G, resid(shm4, type="p"))
> mywhiskers(plotdata$G, resid(shm4, type="p"), se=TRUE, lwd=2, add=TRUE)
> abline(h=0)

> plot(plotdata$N, resid(shm4, type="p"))
> mywhiskers(plotdata$N, resid(shm4, type="p"), se=TRUE, lwd=2, add=TRUE)
> abline(h=0)

> plot(plotdata$Hg, resid(shm4, type="p"))
> mywhiskers(plotdata$Hg, resid(shm4, type="p"), se=TRUE, lwd=2, add=TRUE)
> abline(h=0)

> plot(plotdata$Tg, resid(shm4, type="p"))
> mywhiskers(plotdata$Tg, resid(shm4, type="p"), se=TRUE, lwd=2, add=TRUE)
> abline(h=0)

> plot(plotdata$DGM, resid(shm4, type="p"))
> mywhiskers(plotdata$DGM, resid(shm4, type="p"), se=TRUE, lwd=2, add=TRUE)
> abline(h=0)
Figure 7.23: The predicted distributions and evaluation statistics for plots 25-31 of the modeling data. The figures at the top show the volumes based on the predicted diameter distribution (Vp), on the measured diameters (Vt), and on the ML-fit of the Weibull distribution (VML). The second row shows the p-values of the K-S test for the predicted (pp) and fitted (pt) Weibull distribution (the higher, the better), and the third row the corresponding values of the error index (the lower, the better).

Finally, the predictions from the model were plotted and the goodness of fit was evaluated in the modeling data using the error index, Kolmogorov Smirnov test, and the RMSE and bias of predicted volume. These values are, however, hard to interpret unless we have two methods that are compared to each other. Furthermore, a proper evaluation would require an independent test data, or alternatively, leave-one-out cross-validation. With the KS tests, the null hypothesis that the observed distribution was generated by the PPM prediction was rejected in more than half of the cases. Thus, maybe the models do not perform very well. Maybe the Weibull model was not flexible enough for these data.

Figure 7.23 shows the predicted distributions of plots 25-31, as well as the values of the test statistics for these plots. The evaluation was carried out using the code below.

```r
> errind<-function(d,cdf=function(x) pweibull(x,5,20),power=0,dlim=c(-20,80),cwidth=2) {
+ limits<-seq(dlim[1],dlim[2],cwidth)
+ n<-length(limits)-1
+ means<-(limits[-1]+limits[-(n+1)])/2
+ dp<-d^power
+ sumdp<-sum(dp)
+ ftrue<-sapply(1:n,function(i) sum(dp[d>=limits[i]&d<limits[i+1]])/sumdp
+ Fpred<-cdf(limits)
```
Combining steps ii and iii: Cao’s approach

Cao (2004) proposed an approach to combine steps i and ii in estimation. The idea was
to write the PPM models of step ii into the (plot-specific) likelihood functions of step i, sum up the plot specific likelihoods, and minimize it with respect to the parameters of the PPM model. The function to be minimized was

\[ \ell(\beta_{\alpha,0}, \beta_{\alpha,2}, \ldots, \beta_{\lambda,0}, \beta_{\lambda,1}, \ldots) = \]

\[ \sum_{k=1}^{N} \frac{1}{n_k} \sum_{i=1}^{n_k} \ln g \left[ d_{ki} \mid \alpha|T_k \ldots, \lambda|T_k \ldots \right] = \]

\[ \sum_{k=1}^{N} \frac{1}{n_k} \sum_{i=1}^{n_k} \ln [d_{ki} | f_{\alpha}(T_k, G_k, DGM_k, \ldots | \beta_{\alpha,0}, \beta_{\alpha,1}, \ldots), f_{\lambda}(T_k, G_k, DGM_k, \ldots | \beta_{\lambda,0}, \beta_{\lambda,1}, \ldots)], \]

where \( N \) is the total number of stands in the data. The likelihood is no more a function of the parameters \( \alpha \) and \( \lambda \) but the coefficients of the PPM models, \( \beta_{\alpha,0}, \beta_{\alpha,2}, \ldots, \beta_{\lambda,0}, \beta_{\lambda,1}, \ldots \).

In this method, the plot-specific values of the parameters need not not to be estimated explicitly. Instead, the assumed regression relationships between them and the distribution parameters are written into the likelihood directly. The gain from this approach is to avoid a model chain. In the original PPM, the plot-specific diameter distributions include errors, as do the regression coefficients of step iii, too. In Cao’s approach, the effect of these errors is minimized by doing the estimation in one step. The results of Cao (2004) showed clearly better performance of this method, when compared to the traditional PPM and PRM approaches.

However, still better method was a method termed as cdf-regression. This method also combined the steps ii and ii of the PPM method. However, the estimation was not based on the likelihood but on a nonlinear regression where the squared difference between plot-specific diameter quantiles and the cdf of the assumed distribution was minimized with respect to the PPM model parameters.

The former approach, however, is better justified from the statistical point of view. If the weighting of likelihoods by \( \frac{1}{n} \) is removed, the likelihood is the likelihood of a generalized linear model with Weibull distribution and link functions \( f_{\alpha} \) and \( f_{\beta} \).

**Example 7.30** (Cao’s approach with North-Carelian Scots Pine dataset). Let us refit the PPM models of example 7.29 using the Cao’s approach by fitting the models of shape and scale of the previous example into the likelihood and fit it to the original diameter data. The likelihood is then maximized with respect to the coefficients of the PPM models. However, we do not weight by the number of stems. Then the likelihood is just a likelihood of a Weibull GLM. The previous example provides good initial estimates.

The likelihood function is defined as

```r
# 2 parameter Weibull -logL
nLWeibull<-function(x, asc, bsc, ash, bsh) {
  scale<-asc+bsc*treedata$DGM
  shape<-1/(ash+bsh*treedata$shind)
  -sum(scale^x*exp(-shape*x))
}
```
### 7.4. Modeling Diameter Distributions for Prediction

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Caos approach</th>
<th>Estimate</th>
<th>s.e.</th>
<th>Original PPM</th>
<th>Estimate</th>
<th>s.e.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>-0.6258</td>
<td>0.0951</td>
<td>-0.3771</td>
<td>0.3022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>0.9698</td>
<td>0.0080</td>
<td>0.9146</td>
<td>0.0255</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>0.9890</td>
<td>0.0187</td>
<td>0.9720</td>
<td>0.0307</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>-10989</td>
<td>294</td>
<td>-11069</td>
<td>537</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.3: The parameter estimates by using the PPM method (See Example 7.29 and Cao’s method.

```r
-\sum(dweibull(x, shape=shape, scale=scale, log=TRUE))
```

The fitting is performed using function `mle` as

```r
> treeedata$shind <- treeedata$G/(treedata$DGM^2*treedata$N)
> est <- mle(function(asc=-0.357, bsc=0.920, ash=0.966, bsh=-10990)
+ nLweibull(treedata$d, asc, bsc, ash, bsh)
+ )
> summary(est)
```

The obtained estimates differ quite a lot from the estimates based on the PPM method (Table 7.3). The mean error index and the number stands with rejected KS-test were both lower with Cao’s one-stage PPM method than with the usual two-stage PPM approach. However, the RMSE and absolute bias (true - predicted) of volume were higher. Figure 7.24 shows the predictions using the two methods. It seems that Cao’s method produces flatter predictions, which generally lead to lower error indices. However, maybe these flat predictions have a bit too heavy right tails on average. Because the trees at the right tail have the largest volumes, these heavy tails might provide an explanation for the overestimation of the total volume.

**Percentile-based PPM**

Figure 7.24: The predicted distributions and evaluation statistics for plots 25-31 of the modeling data. The thick dashed line shows the prediction based on Cao’s method.

<table>
<thead>
<tr>
<th></th>
<th>ML-fit</th>
<th>PPM</th>
<th>Cao</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean EI</td>
<td>0.282</td>
<td>0.338</td>
<td>0.332</td>
</tr>
<tr>
<td># of bad fits</td>
<td>4</td>
<td>33</td>
<td>27</td>
</tr>
<tr>
<td>RMSE(V)</td>
<td>1.197</td>
<td>27.987</td>
<td>34.081</td>
</tr>
<tr>
<td>Bias(V)</td>
<td>0.382</td>
<td>-0.893</td>
<td>-13.087</td>
</tr>
</tbody>
</table>

Table 7.4: The evaluation statistics for ML-fit, PPM and Cao’s method.
In that approach, the step 2 of the PPM method includes computing a fixed number of percentiles from each sample plot of the modelling data. In the applications of this method, the number of percentiles has varied between 7 (Mehtätalo et al. 2008) and 12 (Kangas and Maltamo 2000b). For example, (Kangas and Maltamo 2000b) used the 0th, 10th 20th,...,90th, 95th and 100th percentiles.

In the step 3, regression models for these percentiles are fitted. The number of regression models is equal to the number of utilized percentiles. A problem with a high number of percentiles is that the modelling approach does not necessarily guarantee that the predicted percentiles would be in increasing order. By using the SUR method in model fitting, the probability of this situation can be decreased. Furthermore, the use of different constraints in the model fitting has proven to be useful (Kangas et al. 2007).

An essential question is this method is, how to interpolate between consecutive percentiles to construct a continuous predicted distribution function. Borders et al. (1987), Mehtätalo (2005b), and Mehtätalo et al. (2006) used linear interpolation, whereas Kangas and Maltamo (2000b) and Mehtätalo (2004a) used a rational spline that guarantees monotonicity with a high probability. Kangas et al. (2007) and Mehtätalo et al. (2008) used linear interpolation between intermediate percentiles and a second or third order polynomial in the tails. An example of the prediction with a percentile-based model is shown in figure 7.25.

**Mathematical formulation of the PPM method**

Tree diameter in stand \( k \), \( D_k \), is assumed to be a random variable. The randomness in \( D_k \) might be explained by that the diameter \( D_k \) is the diameter of a tree that was randomly selected from among the (finite) population of trees in the stand, \( d_{ki}, i = 1, \ldots, n_k \). In this case, the diameter of a given tree is assumed fixed, and the population would just be the \( n_k \) trees that are in the stand \( k \). In this case, the underlying diameter
distribution is the discrete distribution that would be obtained by measuring all the trees of the stand for diameter.

However, we adopt another way of thinking by assuming that the diameter $D_{ki}$ for a given tree in the stand is also random, having a specified continuous distribution. Thus, we assume that the diameter of this particular tree $i$ was generated by an underlying, stand-specific model or stochastic process. All trees of the stand are assumed to have been generated by the same model, i.e., the diameters are identically distributed (i.d.). Thus, even though we measured all the trees of the stand, we would only observe a finite number of realizations from this continuous model. Because trees are i.d. for the same stand, we can drop the index $i$ from $D_{ki}$ and use $D_k$ instead to simplify notations.

In the Parameter Prediction Method (PPM), the parameter specifying the diameter distribution (e.g., $\theta_k = (\alpha_k, \beta_k)'$) is assumed to be a random vector that takes a unique value for each stand. Furthermore, it is assumed that the value for stand $k$ depends on stand characteristics $x_k$ according to the multivariate model

$$\theta_k = E(\theta_k|\theta_k) + e_k,$$

where $x_k$ includes the values of predictors for stand $k$ (e.g., basal area, mean diameter, and stand age) and $e_k$ includes the stand level residuals for the parameters of stand $k$.

In this model, it is assumed that there is an underlying relationship that expresses the dependence of the expected value of the parameters $\alpha$ and $\beta$ on some stand characteristics $x_k$. The residuals of the PPM model, $e_k$, express how much the realized values of the stand-specific parameters $\theta_k$ deviate from this conditional expectation. They can also be interpreted as stand effects of a random coefficient model. The variance-covariance matrix of stand effects, $\text{var}(e_k)$, specifies how much the stand effects vary around the conditional expectations. Notation $E(\theta_k|\theta_k)$ emphasizes that we assume the fixed part of the model be true, i.e., that it satisfactorily expresses the conditional expectation of the parameters for the given value of vector $x_k$.

The prediction for a new stand $k$ with known $x_k$ is obtained using the estimated expression for the conditional expectation

$$\hat{\theta}_{k, PPM} = E(\theta_k|\theta_k).$$

Note that using transformations with diameter distribution parameters does not introduce any bias problem in the prediction. That is, there is no reason to require, for example $\alpha$ to be unbiased instead of $\ln \alpha$. 

CHAPTER 7. MODELING TREE SIZE DISTRIBUTIONS
7.4. MODELING DIAMETER DISTRIBUTIONS FOR PREDICTION

7.4.2 Parameter recovery approaches

Moment-based recovery

In the moment-based recovery, the parameter estimates are based on the moments of the distribution. Based on the known moments (e.g., mean and variance of diameter), the parameters of the assumed distribution are solved that correspond to these moments. This method corresponds to the well-known method of moments in fitting of an assumed distribution to diameter data (See examples 1.40 and 7.10).

An important question in the method is, how the moments used in the recovery are obtained. They can be based on, for example, on prediction models. Then the method would be very similar to the PPM method. However, instead of modelling the parameters of the distribution directly, the models are fitted for the moments, usually for the mean and variance or standard deviation. A commonly presented justification for this approach over PPM is that the relationships between moments and stand characteristics are better known and easier to justify and interpret.

With diameter distribution models that have parameters for the minimum and maximum diameters (e.g., 3 parameter Weibull, Johnson’s SB, Beta), the recovery of the minimum and maximum is usually not based on the moments. Instead, rough rules for the minima and maxima are utilized, based on a known sample minimum or maximum. Thus, the moment-based recovery usually utilizes only two first moments of the distribution.

Example 7.31 (Moment-based recovery of Weibull parameters). In the dataset of the previous example 7.29, a moment-based recovery of the parameters of the two-parameter Weibull distribution was also conducted. The first task is to model the mean diameter on the stand characteristics. Figure 7.26 shows that the best potential predictor is the DGM, and a variance function might be useful. The code below performs the fit. Figure 7.27 shows that the additional predictors may not improve the model.

```r
> exm1<-gls(ex ~ DGM, data=plotdata)
> exm2<-gls(ex ~ DGM, data=plotdata, weights=varPower(0.5, "fitted(.))")
> anova(exm1,exm2)

Model df AIC  BIC logLik Test L.Ratio p-value
exm1 1 3 232.6417 238.7709 -113.3209
exm2 2 4 210.3830 218.5552 -101.1915 1 vs 2 24.25874 <.0001
> exm2

Generalized least squares fit by REML
Model: ex ~ DGM
Data: plotdata
Log-restricted-likelihood: -101.1915

Coefficients:
(Intercept)        DGM
-0.4299234 0.8261626

Variance function:
Structure: Power of variance covariate
Formula: ~ fitted(.)
```
Figure 7.26: The first moment (mean diameter) on the potential predictors of the PRM model.

Figure 7.27: Some diagnostic plots of the standardized residuals of model exm2.
Figure 7.28: Example fits using the moment-based recovery on six plots. Fits from the previous methods are shown, too.

Parameter estimates:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>power</td>
<td>1.358173</td>
</tr>
</tbody>
</table>

Degrees of freedom: 59 total; 57 residual
Residual standard error: 0.04118751

The second moment is

\[ E(D^2) = \frac{40000}{\pi} \frac{G}{N} \]

because the mean basal area of a tree is \( \frac{G}{N} \), and the mean of \( D^2 \) is obtained by dividing the mean basal area by the constant \( \frac{\pi}{40000} \); this is the widely used quadratic mean diameter. Thus, if \( G \) and \( N \) are used as predictors, they uniquely define the variance of tree diameter, and the second moment is known without prediction errors. However, these characteristics naturally include sampling and measurement errors.

The recovery in the modelling data was based on the mean diameter that was predicted on \( DGM \), and the second moment based on the ratio of basal area and the number of stems. The recovery applies the routines that were introduced earlier in example 1.40. The code below performed the fit. Figure 7.28 shows examples of the fits.

The recovery was not possible for all stands. In addition, the recovered distribution differed quite a lot from the true distribution on some plots. Maybe the models for mean diameter was not good enough, or the two-parameter Weibull was too restrictive assumed distribution function.

Moment-based recovery approaches have been presented, for example, by Ek et al.
CHAPTER 7. MODELING TREE SIZE DISTRIBUTIONS

(1975), Burk and Newberry (1984) and Magnussen (1986)

# A function that computes the moments of weibull-function
wmom<-function(n,alpha,beta) {  
  beta^n*gamma(1+n/alpha)
}

# Newton raphson algorithm for n multidimensional functions using numeric differentiation
# solves equations fn_i=0, where fn_i is the ith function of object fnlist, a list of scalar valued functions.
# The initial values of the parameters are provided in vector 'init'.
NRnum2<-function(init,fnlist,crit=6) {
  par<-init
  # sapply(par,function(x) cat(x," "))
  # cat("
  value<-sapply(fnlist,function(f) f(par))
  j<-1
  while (round(sum(abs(value)),crit)!=0&j<100) {
    # cat(j,"
    grad<-t(sapply(fnlist,function(f) attributes(numericDeriv(quote(f(par)),c("par")))$gradient))
    deltax<-solve(grad,-value)
    par<-par+deltax
    # sapply(par,function(x) cat(x," "))
    value<-sapply(fnlist,function(f) f(par))
    # cat(value," 
    j<-j+1
  }
  if (j<10000) {
    list(par=par,value=value)
  } else {
    list(par=NA,value=NA)
  }
}

# sum of squared differences (m1-mu1)^2+(m2-mu2)^2;
# gets value of 0 if the system of equations has a solution.
fn<-function(theta,m1,m2) {
  alpha<-theta[1]
  beta<-theta[2]
  (wmom(1,alpha,beta)-m1)^2+(wmom(2,alpha,beta)-m2)^2
}

m1pred<-predict(exm2)
m2pred<-plotdata$G/plotdata$N*40000/pi
momtest<-c()
windows()
par(mfcol=c(2,3))
for (i in 1:dim(plotdata)[1]) {
  puut<-treedata[treedata$plot_id==plotdata$plot_id[i],]
  d<-puut$d
  # Define the individual functions and put them into a list
  fn1<-function(parms) wmom(1,parms[1],parms[2])-m1pred[i]
  fn2<-function(parms) wmom(2,parms[1],parms[2])-m2pred[i]
  fnlist<-list(fn1,fn2)
  solution<-nlm(fn,p=c(predict(shm4)[i],predict(scm2)[i]),m1=m1pred[i],m2=m2pred[i])
  amom<-solution$estimate[1]
bmom<-solution$estimate[2]
mom.test<-solution$minimum
  hist(d,main=paste("Plot ",i," ",solution$minimum<1e-3),freq=FALSE)
x<-seq(0,50,0.1)
  lines(x,dweibull(x,plotdata$shape[i],plotdata$scale[i]),lwd=2)
  lines(x,dweibull(x,plotdata$shpred[i],plotdata$scpred[i]),lwd=2,lty="dashed")
  lines(x,dweibull(x,plotdata$shpred2[i],plotdata$scpred2[i]),lwd=2,col="green")
}
}
**Percentile-based recovery**

The percentile-based recovery method would be very similar to the moment-based recovery. However, it is based on predicted percentiles of the diameter distribution. Then the parameters of an assumed distribution are recovered that fulfill the predicted percentiles. In this approach, different percentiles may be used for fitting different parameters. Thus the estimated distribution is not necessarily exactly compatible with the utilized percentiles. An example of the procedure following the prediction of percentiles was presented in example 1.46. Examples of percentile-based recoveries can be found, for example, in McTague and Bailey (1987) and Kuru et al. (1992).

**Recovery based on measured stand characteristics**

The recovery could also be based on the stand characteristics themselves. Then no prediction models would be necessarily needed. For example, in a situation with known $DGM$, $G$, and $N$, the recovery demonstrated in example 7.24.

Notice also that the moment-based recovery would lead to a recovery of this kind, if arithmetic mean diameter, quadratic mean diameter, and basal area or the number of stems are known. Alternatively, the last two characteristics could be replaced with $N$ and $G$.

**7.4.3 Calibrating the predictions**

**Calibrating with measured stand variables**

Kangas and Maltamo (2000a) proposed a calibration estimator for improving a prediction based on the PPM method. In this approach the starting point is a stand table and a set of known stand variables, such as basal area, number of stems or volume for the whole range of diameters or for a smaller range. The stand table can be based, for example, on an utilized PPM model. The aim is to adjust the diameter class frequencies as little as possible to make the prediction compatible with the stand variables that are used in calibration. The adjustment may change the shape of the distribution quite a lot. However, the predictions from the model become more accurate than without
calibration. Nepal and Somers (1992) and Cao and Baldwin (1999) have also proposed approaches for the problem of incompatibility.

The approach of Mehtätalo (2004a) developed the method of Kangas and Maltamo (2000a) to the percentile-based diameter distributions. In that approach, the prediction is not transformed to a stand table, but the predicted percentiles themselves are calibrated. The gain from this approach was that the prediction error variances of the percentile models could be used in calibration, to allow bigger changes in percentiles with higher inaccuracy of prediction and vice versa.

Calibrating with sample diameter distributions

The predicted diameter distribution could also be calibrated with the diameter sample information itself. For example, Mehtätalo (2005b) proposed a method to calibrate predicted percentiles by using the quantiles of the diameter sample. For example, one could (ocularly) select the 3rd smallest tree of an angle-count sample and measure its diameter. This diameter can be interpreted as a measured percentile of the (basal-area weighted) percentile-based diameter distribution. Then the methodological question is, how to use the information of the measured percentile to improve the predictions of the predefined percentiles of the PPM-approach. The proposed solution utilized the linear prediction theory and distributions of order statistics. Furthermore, Mehtätalo et al. (2006) analyzed which quantiles would be the most useful candidates to be measured on the field. The result depended on the criteria. If the aim was to improve the goodness of fit, then smallest trees of the plot would improve the fit most, but if the aim is to improve the prediction of sawtimber volume, then the largest trees would be the best option.

7.5 Exercises

1. Plot the kernel-smoothed density and cdf of the diameter data of Example 1.44 using Gaussian kernel and different values for the smoothing parameter.

2. Tree diameter in a stand follows Beta distribution with parameters $\alpha = 8$, $\beta = 3$, $d_{\text{min}} = 10$ and $d_{\text{max}} = 35$.

   (a) Plot the distribution function of tree diameter.

   (b) Plot the density of tree diameter.

   (c) Compute the expected value and standard deviation of tree diameter and add a vertical line to the plots of density and distribution function at the mean diameter. Interpret the standard deviation.
3. Tree height in a stand follows \( \text{Beta}(20,3,10,30) \) distribution. Tree diameter \( d \) depends on tree height \( h \) according to the inverse Korf function \( d(h) = \frac{-10}{\ln(h - 1.337)} \).

(a) Plot the diameter-height curve.

(b) Plot the inverse function of the D-H curve.

(c) Plot the distribution function of tree diameter \( (F_D(d)) \) and the corresponding density \( (f_D(d) = F_D'(d)) \).

(d) Numerically compute the mean and standard deviation of tree diameters as follows. Define first R-functions for \( d(h)f_H(h) \) and \( d(h)^2f_H(h) \). Then use \( \text{integrate}(\text{fn,lower,upper}) \) for numerically evaluate \( E(D) = \int_0^\infty d(u)f_H(u)du \) and \( E(D^2) = \int_0^\infty d(u)^2f_H(u)du \). In \( \text{integrate}(\text{fn,lower,upper}) \), \( \text{fn} \) is the function to be integrated and \( \text{lower} \) and \( \text{upper} \) are the bounds of integration. Finally, use rule 1.21 to compute the variance.

4. Tree height in a stand follows \( \text{Beta}(20,3,10,30) \) distribution. Stand density is 500 trees per ha. Tree diameter \( d \) depends on tree height \( h \) according to the inverse Korf function \( d(h) = \frac{-10}{\ln(h - 1.337)} \).

(a) Dominant trees include 100 tallest trees of a stand. Plot the distribution function and density of the height of dominant trees.

(b) Compute the dominant height, i.e., the mean of dominant trees as the expected value of the height distribution of dominant trees.

(c) Plot the distribution function and density of the diameter of dominant trees.

5. Assume that tree diameter follows the \( \text{Beta} \) distribution with parameters \( \alpha = 8 \), \( \beta = 3 \), \( d_{\text{min}} = 10 \) and \( d_{\text{max}} = 35 \) and volume as a function of tree diameter is as given in example 2.5.

(a) Plot the c.d.f of volume.

(b) Plot the density of the volume-weighted diameter distribution.

(c) Plot the c.d.f of the volume-weighted diameter distribution.

6. Assuming the H-D curve of example 7.2 and the simulated sample plot of example 7.8, estimate the parameters of the Weibull distribution by assuming that the volume-weighted diameter distribution is of the Weibull form.

7. Text file simulplot.txt includes the simulated sample plot of example 2.8. Take an angle count sample (use the BAF you want) from the trees.
(a) Fit the beta (or whichever distribution you prefer) distribution to the angle count sampling data by assuming that the basal-area weighted diameter distribution is of the beta form.

(b) Fit the beta (or whichever distribution you prefer) distribution to the angle count sampling data by assuming that the unweighted diameter distribution is of the beta form.

Use e.g. the same expressions as in the earlier exercise for the minimum and maximum diameters, and estimate only the two shape parameters using the method of ML. Plot the histogram of the original data and sample and the corresponding fitted distributions using the estimates.

8. Using the data of simulplot.txt take a circular sample plot of the stand

(a) Fit the beta (or whichever distribution you prefer) distribution to the sample by assuming that the unweighted diameter distribution is of the beta form.

(b) Fit the beta (or whichever distribution you prefer) distribution to the sample by assuming that the unweighted diameter distribution is of the beta form.

9. Using the data of simulplot.txt take the Spanish NFI plot at the middle of the plot.

(a) Fit the beta (or whichever distribution you prefer) distribution to the sample by assuming that the unweighted diameter distribution is of the beta form.

(b) **EXTRA** Fit the beta (or whichever distribution you prefer) distribution to the sample by assuming that the unweighted diameter distribution is of the beta form.

10. Tree diameter in a stand follows \( \text{Beta} \) distribution with parameters \( \alpha = 8, \beta = 3 \), \( d_{\text{min}} = 10 \) and \( d_{\text{max}} = 32 \). Stand density is 550 trees per ha.

(a) Compute the number of trees with diameter above 28 cm.

(b) Compute the number of trees between 23 and 25 cm.

(c) Compute the number of trees below 22 cm.

11. Tree diameter in a stand follows \( \text{Beta} \) distribution with parameters \( \alpha = 8, \beta = 3 \), \( d_{\text{min}} = 10 \) and \( d_{\text{max}} = 32 \). The basal area is 22 m\(^2\) per ha.

(a) Compute the number of trees with diameter above 28 cm.

(b) Compute the number of trees between 23 and 25 cm.

(c) Compute the number of trees below 22 cm.
12. Tree diameter in a stand follows Beta distribution with parameters $\alpha = 8$, $\beta = 3$, $d_{min} = 10$ and $d_{max} = 32$. Tree height depends on tree diameter according to the Korf's curve of example 7.21, and the volume on diameter and height according to the function given in example 7.21. You may alternatively use your own functions as well, e.g., the volume functions that are commonly used in your home country. The total volume is 240 m$^3$ per ha.

   (a) Compute the number of trees with diameter above 28 cm.
   (b) Compute the number of trees between 23 and 25 cm.
   (c) Compute the number of trees below 22 cm.

13. Perform the parameter recovery of example 7.24 by assuming that the unweighted diameter distribution is of the Weibull form.

14. Perform the parameter recovery of example 7.24 by assuming that the basal-area weighted diameter distribution is of the Beta form and the minimum and maximum diameters are known.

15. Perform the parameter recovery of example 7.25 by assuming that the unweighted diameter distribution is of the Weibull form.

16. Perform the parameter recovery of example 7.25 by assuming that the basal-area weighted diameter distribution is of the Beta form and the minimum and maximum diameters are known.

17. Perform the parameter recovery of example 7.27 by using the volume and H-D models from your home country.

18. Perform the parameter recovery of example 7.27 by using the basal area instead of the number of stems.

19. Perform the PPM- method in the North-Carelian Scots pine data, or into your own dataset.

   (a) Analyze the goodness of ML-fit for 2- and 3 parameter Weibull, Beta, Jonson’s SB and logit-logistic distribution.
   (b) Estimate the PPM models for the parameters of the best-fitting distribution
   (c) Evaluate the models in the modelling data.
   (d) Refit the models using Cao’s GLM approach and compare to the earlier models.
20. By using the best fitting distribution function of the previous example, perform the moment-based parameter recovery to the data. If the model is one of the functions using minimum and maximum diameters, fit models for them as well. Compare the fit to that of the previous example.


22. Perform the parameter recovery of example 7.24 by assuming that the weighted diameter distribution is of the logit-logistic form, and the minimum and maximum diameters are 4 and 39 cm, respectively.
Chapter 8

Additional topics

8.1 Regression splines

Sometimes the dependence of the response on the predictors is not of the primary interest in a modeling situation. However, the effect needs to be removed to better recognize the effects of primary interest. In other situations, no good function is known for the relationship and the model may not be willing to spend too much time in finding a good model for the relationship. A polynomial model might be a possibility in these situations, but it is well known that their behavior may be very wild when extrapolated outside the range of data.

An alternative for these situations is the spline regression. Spline is a continuous, smooth function consisting of pieces of 2nd or 3rd order polynomials that are joined at knots. The splines are called correspondingly quadratic and cubic splines. (Harrell 2001) presented a nice approach to use so called restricted cubic splines as a fixed part of a model. A nice property of the applied spline is that the model remains as linear, and the spline just provides a way to compute additional transformation of the predictor variable. The flexibility of the function is controlled by the number of knots, which can be placed e.g. at certain quantiles of the predictor variable.

The following pages show copies of the Harrell’s book (Harrell 2001) on the regression splines.
CHAPTER 8. ADDITIONAL TOPICS

20 Chapter 2. General Aspects of Fitting Regression Models

Such a smooth cubic spline function with three knots \((a, b, c)\) is given by

\[
f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3
+ \beta_4(X - a)^3 + \beta_5(X - b)^3 + \beta_6(X - c)^3,
\]

with the following constructed variables:

\[
\begin{align*}
X_1 &= X \quad X_2 = X^2 \\
X_3 &= X^3 \quad X_4 = (X - a)^3 \\
X_5 &= (X - b)^3 \quad X_6 = (X - c)^3.
\end{align*}
\]

(2.23)

If the cubic spline function has \(k\) knots, the function will require estimating \(k + 3\) regression coefficients besides the intercept. See Section 2.4.5 for information on choosing the number and location of knots.

There are more numerically stable ways to form a design matrix for cubic spline functions that are based on B-splines instead of the truncated power basis used here. However, B-splines are more complex and do not allow for extrapolation beyond the outer knots, and the truncated power basis seldom presents estimation problems (see Section 4.6) when modern methods such as the QR decomposition are used for matrix inversion.

2.4.4 Restricted Cubic Splines

Stone and Kooper have found that cubic spline functions do have a drawback in that they can be poorly behaved in the tails, that is before the first knot and after the last knot. They cite advantages of constraining the function to be linear in the tails. Their restricted cubic spline function (also called natural splines) has the additional advantage that only \(k - 1\) parameters must be estimated (besides the intercept) as opposed to \(k + 3\) parameters with the unrestricted cubic spline. The restricted spline function with \(k\) knots \(t_1, \ldots, t_k\) is given by

\[
f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_{k-1} X_{k-1},
\]

(2.24)

where \(X_1 = X\) and for \(j = 1, \ldots, k - 2,\)

\[
X_{j+1} = \begin{cases} 
(X - t_j)^3 & -(X - t_{j-1})^3 (t_k - t_j)/(t_k - t_{k-1}) \\
+ (X - t_k)^3 (t_k - t_{j-1})/(t_k - t_{k-1}).
\end{cases}
\]

(2.25)

It can be shown that \(X_j\) is linear in \(X\) for \(X \geq t_j\). Figure 2.2 displays the spline component variables \(X_j\) for \(j = 2, 3, 4\) and \(k = 5\) and one set of knots. The left graph magnifies the lower portion of the curves. Figure 2.3 displays some typical
shapes of restricted cubic spline functions with \( k = 3, 4, 5, \) and 6. These functions were generated using random \( \beta \).

Once \( \beta_0, \ldots, \beta_{k-1} \) are estimated, the restricted cubic spline can be restated in the form

\[
 f(X) = \beta_0 + \beta_1 X + \beta_2(X - t_1)^2 + \beta_3(X - t_2)^2 + \ldots + \beta_{k+1}(X - t_k)^2 
\]

by computing

\[
 \beta_k = \frac{[\beta_2(t_2 - t_k) + \beta_3(t_2 - t_k) + \ldots + \beta_{k-1}(t_{k-2} - t_k)]/(t_k - t_{k-1})}{(t_k - t_{k-1})} \quad (2.27) \\
 \beta_{k+1} = \frac{[\beta_2(t_2 - t_{k+1}) + \beta_3(t_2 - t_{k+1}) + \ldots + \beta_{k-1}(t_{k-2} - t_{k+1})]/(t_{k+1} - t_k)}{(t_{k+1} - t_k).} 
\]

A test of linearity in \( X \) can be obtained by testing

\[
 H_0 : \beta_2 = \beta_3 = \ldots = \beta_{k-1} = 0. \quad (2.28) 
\]

The truncated power basis for restricted cubic splines does allow for rational (i.e., linear) extrapolation beyond the outer knots. However, when the outer knots are in the tails of the data, extrapolation can still be dangerous.
FIGURE 2.3: Some typical restricted cubic spline functions for $k = 3, 4, 5, 6$. The $y$-axis is $\delta^2$. Arrows indicate knots.
8.1. REGRESSION SPLINES

2.4 Relaxing Linearity Assumption for Continuous Predictors

<table>
<thead>
<tr>
<th>TABLE 2.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

When nonlinear terms in Equation 2.25 are normalized, for example, by dividing them by the square of the difference in the outer knots to make all terms have units of X, the ordinary truncated power basis has no numerical difficulties when modern matrix algebra software is used.

2.4.5 Choosing Number and Position of Knots

We have assumed that the locations of the knots are specified in advance; that is, the knot locations are not treated as free parameters to be estimated. If knots were free parameters, the fitted function would have more flexibility but at the cost of instability of estimates, statistical inference problems, and inability to use standard regression modeling software for estimating regression parameters.

How then does the analyst preassign knot locations? If the regression relationship were described by prior experience, prespecification of knot locations would be easy. For example, if a function were known to change curvature at X = α, a knot could be placed at α. However, in most situations there is no way to prespecify knots. Fortunately, Stone[80] has found that the location of knots in a restricted cubic spline model is not very crucial in most situations; the fit depends much more on the choice of k, the number of knots. Placing knots at fixed quantiles (percentiles) of a predictor's marginal distribution is a good approach in most datasets. This ensures that enough points are available in each interval, and also guards against letting outliers overly influence knot placement. Recommended equally spaced quantiles are shown in Table 2.3.

The principal reason for using less extreme default quantiles for k = 3 and more extreme ones for k = 7 is that one usually uses k = 3 for small sample sizes and k = 7 for large samples. When the sample size is less than 100, the outer quantiles should be replaced by the fifth smallest and fifth largest datapoints, respectively.80 What about the choice of k? The flexibility of possible fits must be tempered by the sample size available to estimate the unknown parameters. Stone[80] has found that more than 5 knots are seldom required in a restricted cubic spline model. The principal decision then is between k = 3, 4, or 5. For many datasets, k = 4 offers an adequate fit of the model and is a good compromise between flexibility and loss of precision caused by overfitting a small sample. When the sample size is large...
(e.g., n > 100 with a continuous uncensored response variable), k = 5 is a good choice. Small samples (< 30, say) may require the use of k = 3. Akaike's information criterion (AIC, Section 9.8.1) can be used for a data-based choice of k. The value of k maximizing the model likelihood ratio \( \chi^2 \) is the best "for the money" using AIC.

The analyst may wish to devote more knots to variables that are thought to be more important, and risk lack of fit for less important variables. In this way the total number of estimated parameters can be controlled (Section 4.1).

2.4.6 Nonparametric Regression

One of the most important results of an analysis is the estimation of the tendency (trend) of how X relates to Y. This trend is useful in its own right and it may be sufficient for obtaining predicted values in some situations, but trend estimates can also be used to guide formal regression modeling (by suggesting predictor variable transformations) and to check model assumptions.

Nonparametric smoothers are excellent tools for determining the shape of the relationship between a predictor and the response. The standard nonparametric smoothers work when one is interested in assessing one continuous predictor at a time and when the property of the response that should be linearly related to the predictor is a standard measure of central tendency. For example, when \( C(Y) \) is \( E(Y) \) or \( P(Y = 1) \), standard smoothers are useful, but when \( C(Y) \) is a measure of variability or a rate (instantaneous risk), or when Y is only incompletely measured for some subjects (e.g., Y is censored for some subjects), simple smoothers will not work.

The oldest and simplest nonparametric smoother is the moving average. Suppose that the data consist of the points X = 1, 2, 3, 5, and 8, with the corresponding Y values 2.1, 3.8, 5.7, 11.1, and 17.2. To smooth the relationship we could estimate \( E(Y|X = x) \) by \((2.1+3.8+5.7)/3\) and \( E(Y|X = 2+3+6)/3 \) by \((3.8+5.7+11.1)/3\). Note that overlap is fine; that is one point may be contained in two sets that are averaged. You can immediately see that the simple moving average has a problem in estimating \( E(Y) \) at the outer values of X. The estimates are quite sensitive to the choice of the number of points (or interval width) to use in "bining" the data.

A moving least squares linear regression smoother is far superior to a moving flat line smoother (moving average). Cleveland's moving linear regression smoother loess has become the most popular smoother. To obtain the smoothed value of Y at \( X = x \), we take all the data having X values within a suitable interval about x. Then a linear regression is fitted to all of these points, and the predicted value from this regression at \( X = x \) is taken as the estimate of \( E(Y|X = x) \). Actually, loess uses weighted least squares estimates, which is why it is called a locally weighted least squares method. The weights are chosen so that points near \( X = x \) are given
8.2 QUANTILE REGRESSION

8.2 Quantile regression

See background and theory on Quantile regression e.g. from Koenker and Hallock (2001) or in Koenker (2005). The followings subsections are taken from Mehtätalo et al. (2008)

Definition of sample quantile

Suppose that we have $n_{ij}$ measurements of tree diameter on the $i$th plot on the $j$th measurement occasion, namely $y_{ij1}, \ldots, y_{ijn_{ij}}, \ i = 1, \ldots, n, \ j = 1, \ldots, n_i$. Assuming no spatial autocorrelation among trees within a stand, these constitute an i.i.d. random sample from the underlying diameter distribution of the population (stand). From among several possible definitions of sample percentile (Hyndman and Fan 1996), we selected the one consistent with the quantile regression approach. The $100\tau$th sample percentile (i.e., the $\tau$th quantile) for plot $i$ at time $j$ is the value of $\xi_{\tau ij}$ that minimizes the distance

$$\sum_{k=1}^{n_{ij}} \rho_{\tau}(y_{ijk} - \xi_{\tau ij}) ,$$

(8.1)

where

$$\rho_{\tau}(u) = \begin{cases} (\tau - 1)u & u < 0 \\ \tau u & u \geq 0 \end{cases} .$$

The solution can be found by linear programming (Koenker 2005, Koenker and Hallock 2001). This definition implies that the $r$th smallest tree of the plot is interpreted as $100p = 100(r - 1)/(n_{ij} - 1)$th percentile, and the tree with $p$ closest to $\tau$ is selected as $\xi_{\tau ij}$.

Quantile regression

In quantile regression, quantiles of the response are modeled instead of expectations. In contrast to the two stage procedure of the previous subsection, the quantile regression provides a one-stage procedure, where the model is fitted directly to the data of tree diameters.

The least squares estimates for the parameters of (2.7) are found by minimizing the squared distance $\sum_{i=1}^{n} \sum_{j=1}^{n_i} (\xi_{\tau ij} - \mu(x_{ij}, \theta_\tau))^2$, where $\mu(x_{ij}, \theta_\tau)$ is the linear predictor of the assumed mean function (for simplicity of notations, all parameters are stacked into $\theta_\tau$ and predictors into $x_{ij}$). Replacing the sample quantile $\xi_{\tau ij}$ in (8.1) with a linear predictor $\mu(x_{ij}, \theta_\tau)$, gives a corresponding distance measure in the context of quantile regression

$$d_\tau = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \sum_{k=1}^{n_{ij}} \rho_{\tau}(y_{ijk} - \mu(x_{ijk}, \theta_\tau)) .$$

(8.2)
The parameters of the quantile regression model are found by minimizing distance (8.2) with respect to parameter $\theta_\tau$, for example by a linear programming approach (Koenker 2005).

A marginal QR model that corresponds to the linear fixed-effects model, \((\ref{eq:linear-model})\), is obtained by defining

$$
\mu(x_{ij}, \theta_\tau) = \alpha_\tau + \beta_\tau T_{ijk} + \gamma_\tau' x_{ij}, \quad (8.3)
$$

where $\theta_\tau' = [\alpha_\tau, \beta_\tau, \gamma_\tau']$. The conditional QR model that corresponds to fixed-effects model \((2.7)\) is

$$
\mu(x_{ij}, \theta_\tau) = \alpha_{\tau_i} + \beta_{\tau_i} T_{ijk} + \gamma_\tau' x_{ij}. \quad (8.4)
$$

where $\theta_\tau' = [\alpha_{\tau_1}, \ldots, \alpha_{\tau_n}, \beta_{\tau_1}, \ldots, \beta_{\tau_n}, \gamma_\tau']$.

The latter of these models is similar to the quantile regression model with fixed effects, proposed by Koenker Koenker (2004). Because of a small number of observations per subject, Koenker specified only the constant to vary among subjects, and this constant was common for all quantiles, implying that the level of the distribution varied among subjects but the shape was the same. In our case, we specified separate constant and slope parameters for each plot and quantile. Koenker (Koenker 2004) also presented a penalized quantile regression version of his model, where the individual subject effects are shrunk toward a common mean to control the variability introduced by the large number of fixed coefficients. This approach, which may be viewed as a step toward mixed-effect quantile regression modeling even though the parameters remain fixed, would likely have been the best available QR approach for our study. However, as Koenker specified it for random constant only, and it is not yet available in statistical software, we did not investigate that promising approach in this study.

When the diameter distributions do not exhibit uniform variance over the predictors, one may improve the fit using a weighted quantile regression. For weighted estimation, distance (8.2) is replaced with weighted distance

$$
d_\tau = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \sum_{k=1}^{n_{ijk}} w_{\tau_{ijk}} \rho_\tau (y_{ijk} - \mu(x_{ij}, \theta_\tau)). \quad (8.5)
$$

The weights $w_{\tau_{ijk}}$ should be proportional to the local density of tree diameter evaluated at the quantile of interest (Koenker 2005), which is intuitively appealing because sample quantiles are more variable when the local density is low. The greater the local density, the greater should be the weight given to that observation.

In the conditional model, the conditional density of tree diameter can be directly estimated, for example, by applying a kernel density estimator for the data of each measurement occasion within plot. For the marginal model, the density used as weight
should be conditional only on the fixed predictors but not on the plot and measurement occasion. Thus, we need a method that uses observations from several plots to evaluate the local density on fixed predictors. Such a method is obtained by first estimating the conditional \( \tau - h \) and \( \tau + h \) quantiles using the unweighted quantile regression (Koenker 2005). Then the density at \( \tau \)-th quantile is approximated as the average density within window \( (\tau - h, \tau + h) \) as

\[
 w_{\tau ij k} = \frac{2h}{\mu(x_{ij k}, \hat{\theta}_{\tau + h}) - \mu(x_{ij k}, \hat{\theta}_{\tau - h})},
\]

where \( h \) is a positive constant. Koenker (2005) mentions some equations proposed for estimation of the window width. In general, the window should be narrower at the tails than for intermediate quantiles, and the width should decrease with increasing sample size.

**Example 8.1.** We have a dataset including repeated measurements of even-aged, plantation-grown Loblolly pine tree diameter. The aim is to model the diameter distribution through the percentile-based method. In that method, a fixed number of percentiles is predicted for a plot. A continuous diameter distribution is obtained through interpolating between the predicted percentiles. Mehtälo et al. (2008) applied the quantile regression approach for modeling the diameter percentiles. A summary of the modeling data are shown below.

```r
> summary(dat2)
```

<table>
<thead>
<tr>
<th>id</th>
<th>dbh</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1112</td>
<td>460</td>
<td>Min.</td>
</tr>
<tr>
<td>1404</td>
<td>457</td>
<td>1st Qu.</td>
</tr>
<tr>
<td>2004</td>
<td>450</td>
<td>Median:</td>
</tr>
<tr>
<td>2106</td>
<td>449</td>
<td>Mean  :</td>
</tr>
<tr>
<td>3104</td>
<td>434</td>
<td>3rd Qu.:</td>
</tr>
<tr>
<td>3108</td>
<td>428</td>
<td>Max.  :</td>
</tr>
<tr>
<td>(Other):27221</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The first question is to find a sufficiently flexible function to express the dependence of diameter percentiles on stand age. We used the regression spline for that purpose (Harrell 2001). First, we need to decide the placing of knots. Using some trial and error procedure, we decided to use either 3 or 4 knots. The first thing is to compute the corresponding transformations of tree age into the dataset.
Then, we fit quantile regression model using function rq of package quantreg. We model the 7th, 21th, ..., 79th and 93th percentiles with spline regression using 4 knots. Finally, we plot the predicted quantile curves.

```r
rq1 <- rq(dbh ~ age + age_3 + age_4, tau = c(0.07, 0.21, 0.36, 0.50, 0.64, 0.79, 0.93), data = dat2)
rq1

Coefficients:

<table>
<thead>
<tr>
<th>tau</th>
<th>(Intercept)</th>
<th>age</th>
<th>age_3</th>
<th>age_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.07</td>
<td>-0.79952273</td>
<td>0.79844034</td>
<td>-0.01573469</td>
<td>0.02507528</td>
</tr>
<tr>
<td>0.21</td>
<td>-1.81499029</td>
<td>1.10477670</td>
<td>-0.02219417</td>
<td>0.03405455</td>
</tr>
<tr>
<td>0.36</td>
<td>-1.89818030</td>
<td>1.25327251</td>
<td>-0.02547720</td>
<td>0.03940450</td>
</tr>
<tr>
<td>0.50</td>
<td>-1.32627263</td>
<td>1.30337044</td>
<td>-0.02341775</td>
<td>0.03448462</td>
</tr>
<tr>
<td>0.64</td>
<td>-1.97736526</td>
<td>1.47050227</td>
<td>-0.02847747</td>
<td>0.03438256</td>
</tr>
<tr>
<td>0.79</td>
<td>0.07995227</td>
<td>2.41342759</td>
<td>1.60327950</td>
<td>0.07995227</td>
</tr>
<tr>
<td>0.93</td>
<td>0.21</td>
<td>1.77336974</td>
<td>0.07995227</td>
<td></td>
</tr>
</tbody>
</table>
```

The constant and coefficient of age in the model vary among plots.

```r
library(quantreg)
rq1 <- rq(dbh ~ age + age_3 + age_4, tau = c(0.07, 0.21, 0.36, 0.50, 0.64, 0.79, 0.93), data = dat2)

# Print some of teh coefficients
coef(rq1)[c(1:10, 95:107, 195:200), c(1, 2, 4, 5, 7)]
```

Degrees of freedom: 29899 total; 29895 residual
8.2. QUANTILE REGRESSION

Figure 8.1: Tree diameter data plotted on plot age, and the predicted diameter quantile curves.

The columns in the above matrix of coefficients correspond to each quantile. The number of parameters in each of the models is high because we had the constant and coefficient of age specified separately for each plot.

Next, we compute the mean over the plot-specific constants and slopes and use those as the estimate for the diameter percentiles of an average plot. Figure 8.2 shows the original data, predicted quantile curves for an average plot, and predicted quantile curves for plot numbers 1 and 5.

```r
# Extract the plot-specific constants and coefficients of age
coeff2<-(coefficient(rq)[1:100,])
coeff2c(mean(coefficient(rq)[1:100,]))

# Compute the population-level constant and slope as the mean over the plot-specific values
pop.coef<-(cbind(pop(coef2),mean(slope2)))
```

```r
# Compute the population-level constant and slope as the mean over the plot-specific values
pop.coef<-(cbind(mean(coefficient(rq)[1:100,]),
apply(slope2,2,mean),
coefficient(rq)[101],coefficient(rq)[102])
```
Figure 8.2: Tree diameter data plotted on plot age, and the predicted diameter quantile curves for an average plot (red solid lines) and for two arbitrarily selected plots (blue and green dashed line sets).

```r
# compute the constant and slope for plot 1 and plot 5
plot1.coef <- cbind(const2[1,], slope2[1,],
                    coef(rq1)[101], coef(rq1)[102])
plot5.coef <- cbind(const2[5,],
                    slope2[5,],
                    coef(rq1)[101], coef(rq1)[102])

pdf("d:\laurin\biometria\lecturenotes\figquant2.pdf")
par(mai=c(0.5,0.5,0.5,0.5), mar=c(1.5,0.5,0.5))
x<-seq(8,40,0.1)
plot(dat2$age, dat2$dbh, cex=0.1)
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[1,],
      col="red", lwd=2)
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[2,],
      col="red", lwd=2)
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[3,],
      col="red", lwd=2)
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[4,],
      col="red", lwd=2)
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[5,],
      col="red", lwd=2)
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[6,],
      col="red", lwd=2)
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[7,],
      col="red", lwd=2)
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[1,],
      col="blue", lwd=2, lty="dashed")
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[2,],
      col="blue", lwd=2, lty="dashed")
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[3,],
      col="blue", lwd=2, lty="dashed")
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[4,],
      col="blue", lwd=2, lty="dashed")
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[5,],
      col="blue", lwd=2, lty="dashed")
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[6,],
      col="blue", lwd=2, lty="dashed")
lines(x, cbind(1,x,natural.spline.comp(x,t4,1),natural.spline.comp(x,t4,2))%*%plot1.coef[7,],
      col="blue", lwd=2, lty="dashed")
```

8.3. NUMERICAL COMPUTATIONS

R is not only an environment for conventional statistical analyses but it has lot of possibilities for use in different kind of analyses.

8.3.1 Numerical integration

In many applications, one needs to integrate over a specific function. However, analytical solutions to integrals are often hard to compute. Therefore, numerical integration can be used instead. A good example of such computations is computing volume of a stem using a taper curve.

Example 8.2. Integrating taper curves to calculate the stemwood volume.

```r
library(MASS)
kozakII<-function(D,H,h,a0,a1,a2,b1,b2,b3,b4,b5) {
sq <- sqrt(h/H)
sh<-sqrt(1.3/H)
dh<-D/H
colabs-
```

```r
> head(dat)
    Tree Dbh SmoothDbh Hht h d q
177 17 49.5 47.57505 32.5 0.325 54.4 0.01
178 17 49.5 47.57505 32.5 1.625 45.9 0.05
179 17 49.5 47.57505 32.5 3.250 40.2 0.10
180 17 49.5 47.57505 32.5 6.500 37.0 0.20
181 17 49.5 47.57505 32.5 9.750 33.2 0.30
182 17 49.5 47.57505 32.5 13.000 31.2 0.40

fm1.nlme<-nlme(I(dˆ2)˜(kozakII(Dbh,Hht,h,a0=1,a1,a2,b1,b2,b3,b4=0,b5))ˆ2, data=dat, fixed = a1+a2+b1+b2+b3+b4+b5 ˜ 1, random = a2 + b2 + b3 ˜ 1|Tree, start=c(a1=0.967171771,a2=0.082204187, b1=0.519253787,b2=1.816359015,b3=-0.009285914), weights=varPower(form = ~Dbh, 0.5))
```
# returns tree volume in cubic meters
volumef0<-function(D,H,a0=1,a1=fixef(fm1.nlme)[1], a2=fixef(fm1.nlme)[2],b1=fixef(fm1.nlme)[3], b2=fixef(fm1.nlme)[4],b3=fixef(fm1.nlme)[5], b4=0,b5=fixef(fm1.nlme)[6]) {
  fn<-function(h) pi*((kozakII(D,H,h,a0,a1,a2,b1,b2,b3,b4))/2)^2
  integrate(fn,0.1,H)$value/10000
}

> summary(fm1.nlme)
Nonlinear mixed-effects model fit by maximum likelihood
Model: I(d^2) ~ (kozakII(Dbh, Hht, h, a0 = 1, a1, a2, b1, b2, b3, b4 = 0, b5)^2
Data: dat
AIC    BIC    logLik
10121.65 10191.69  -5046.823
Random effects:
  Formula: list(a2 ~ 1, b2 ~ 1, b3 ~ 1)
  Level: Tree
  Structure: General positive-definite, Log-Cholesky parametrization
    StdDev    Corr
     a2 0.01332445 a2 b2
     b2 0.22479680 -0.851
     b3 0.16948806  0.184  0.360
     Residual 0.21367461
Variance function:
  Structure: Power of variance covariate
  Formula: ~Dbh
  Parameter estimates:
     power
     1.508338
Fixed effects: a1 + a2 + b1 + b2 + b3 + b5 ~ 1
  Value   Std.Error   DF  t-value  p-value
   a1  0.9770566 0.01172492 995 83.33164 0.0000
   a2  0.0237965 0.01342539 995  1.77250 0.0766
   b1  0.4189223 0.01778306 995 23.55737 0.0000
   b2  0.7629973 0.04921825 995 15.50233 0.0000
   b3  0.5067801 0.04623502 995 10.96096 0.0000
   b5 -0.0018069 0.00061716 995 -2.92772 0.0035
Correlation:
   a1    a2    b1    b2    b3
a2 -0.994  

b1 -0.062  0.011  

b2  0.014 -0.015 -0.725  

b3  0.001 -0.001  0.765 -0.479 -0.862  

b5  0.765 -0.479 -0.862  

Standardized Within-Group Residuals:
     Q1   Min   Q1        Med        Q3       Max
  -3.47221615 -0.53096475 -0.01215858  0.44067199  3.46177788
Number of Observations: 1100
Number of Groups: 100

> # Plot the taper curve for a 20-20-tree
> pdf("d:/laurim/biometria/lecturenotes/figquant3.pdf")
> h<-seq(0,20,0.1)
> plot(h,kozakII(20,20,h,a0=1,a1=fixef(fm1.nlme)[1], a2=fixef(fm1.nlme)[2],b1=fixef(fm1.nlme)[3], b2=fixef(fm1.nlme)[4],b3=fixef(fm1.nlme)[5], b4=0,b5=fixef(fm1.nlme)[6]),type="l",
  xlab="h, m", ylab="d, cm")
> dev.off()
> # Volume for a tree with DBH of 20 and H of 20
> volumef0(20,20)
[1] 0.2462374
> # Volume for a tree with DBH of 25 and H of 25
> volumef0(25,25)
[1] 0.4603786

Note that the above-defined function volumef0 could be used also as a model func-
tion in nls fitting. That is, one could fit a compatible system of volume model and taper curve without integrating the taper curve analytically.

Integrals over multi-dimensional support (e.g., area integrals) can be computed in R using function adapt in library fCopulae.

8.3.2 Solving equations

Another need for numerical computations arises from solving equations. A specific need for solving equations arises from inverting complicated functions, as shown in the following example.

Example 8.3. To compute the merchantable volume, we need to know the merchantable height. Assume that the merchantable diameter is 10 cm. We want to compute the merchantable height for that three. It is obtained as a solution to the equation

\[ f(x, D, H|\theta) = 10 \]

where \( f(x, D, H|\theta) \) is the taper curve.

Using the taper curve of the previous example, the solution is not easy analytically. To compute it numerically, we can use a simple section-halfing algorithm, implemented in lmfor function updown. Another alternative is the Newton-Raphson method. Package lmfor includes function NR for that purpose, but to use it we should write a function to compute the gradients. However, lmfor also has function NRnum, which computes the gradients numerically. NRnum can also be used for a system of equations (the algorithm known as Gauss-Newton).
# A function that computes the difference between the taper function and value 10 cm.
> fn<-function(x) kozakII(20,20,x,a0=1,a1=fixef(fm1.nlme)[1], + a2=fixef(fm1.nlme)[2],b1=fixef(fm1.nlme)[3], + b2=fixef(fm1.nlme)[4],b3=fixef(fm1.nlme)[5], + b4=0,b5=fixef(fm1.nlme)[6])-10
> 
> pdf("d:/laurim/biometria/lecturenotes/figquant4.pdf")
> h<-seq(0,20,0.1)
> plot(h,fn(h),type="l")
> abline(h=0)
> dev.off()

> # Compute the merchantable height
> # see ?updown
> # see ?NRnum
> updown(0,20,fn)
[1] 12.00688
>
> NRnum(init=10,fnlist=list(fn))
$par
  a1
  12.00688

$value
  a1
  -4.973799e-14

We see that the merchantable height is 12.00688 meters for this tree.
The merchantable volume can be further computed by integration of the taper curve up to the merchantable height. The merchantable volume is 0.22 m³.

> Hm<-NRnum(init=10,fnlist=list(fn))$par
> Hm
  a1
  12.00688

$volumef<-function(D,H,a0=1,a1=fixef(fm1.nlme)[1], + a2=fixef(fm1.nlme)[2],b1=fixef(fm1.nlme)[3], + b2=fixef(fm1.nlme)[4],b3=fixef(fm1.nlme)[5], + b4=0,b5=fixef(fm1.nlme)[6],Hm=H) { fn<-function(h) pi *((kozakII(D,H,h,a0,a1,a2,b1,b2,b3,b4,b5))/2)^2 + integrate(fn,0.1,Hm)$value/10000 + }
>
> volumef(20,20,Hm=Hm)
[1] 0.2247438

8.4 exercises

1. Data patti.txt includes observations on increment cores using densiometer. The dat includes the following variables: Plot: plot id, Tree: tree id, SDClass: thinning treatment (1=control ... 3=heavy), Diam1986: DBH before thinning; Year: the year of the increment; CA: Age of the tree; RW: ring width (mm); RD: ring denity; RBA: Ring basal area. Fit a mixed-effects model for the development of ring basal area for the years before thinning using the fixed part based on the spline regression.

2. Model 90th, 50th and 10th quantiles of diameter growth after thinning on tree age using the observation of patti.txt after calendar year 1990. Try fitting both
8.4. EXERCISES

by using tree dummies and not using them.

3. Fit a model for volume using the function \( volume f0 \) as the volume function. Use your own volume data.

4. Using the taper curve of the example, compute the merchantable height and volume for a tree with diameter 30 cm and height of 20 m using 15 cm as the merchantable diameter.

5. Write a R- function for the merchantable height as a function of merchantable diameter, when tree taper follows the taper curve of the example. Make a plot of the function for a tree with D=20 and H=20.

6. Write a R-function that returns the merchantable volume as a function of merchantable diameter, tree diameter and height.
Appendix A

Matrix algebra

These pages have been taken from lecture notes on Forest biometrics by professor Annika Kangas.
Matrix calculus

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{(n \times p)} = \begin{bmatrix} X_{11} &amp; X_{12} &amp; \cdots &amp; X_{1p} \ X_{21} &amp; X_{22} &amp; \cdots &amp; X_{2p} \ \vdots &amp; \vdots &amp; \ddots &amp; \vdots \ X_{n1} &amp; \cdots &amp; \cdots &amp; X_{np} \end{bmatrix}$</td>
<td>$x_{(p \times 1)} = \begin{bmatrix} x_1 \ x_2 \ \vdots \ x_p \end{bmatrix}$</td>
</tr>
</tbody>
</table>

Element $x_{ij}$ means the $j$th element of row $i$ in matrix $X$.

A matrix with only one column is called vector. $x_i$ is the $j$th element of vector $x$.

A single number is called **scalar**.

<table>
<thead>
<tr>
<th>Transpose of a vector</th>
<th>Transpose of a matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x' = x^T$</td>
<td>$X' = X^T$</td>
</tr>
</tbody>
</table>

Rows and columns interchange.

R: $t(x)$

R: $t(X)$

$X_{(n \times p)}' = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_p \end{bmatrix}$

$X^T_{(p \times n)} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{p1} & \cdots & \cdots & X_{pn} \end{bmatrix}$
Example

\[ X_{(4 \times 3)} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} \]

\[ x_{(4 \times 1)} = \begin{bmatrix} 1 \\ 4 \\ 7 \\ 10 \end{bmatrix} \]

\[ X <- \text{matrix(seq(1,12),ncol=3,byrow=TRUE)} \]

\[ x <- X[,1] \]

\[ X <- \text{rbind(1:3,4:6,7:9,10:12)} \]

Element \( x_{23} = 6 \)
Element \( x_{32} = 8 \)
Element \( x_{3} = 7 \)

\[ Y_{(3 \times 4)} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} \]

\[ Y <- t(X) \]

Element \( x_{23} = 8 \)
Element \( x_{32} = 6 \)

**Square matrix**
- has equal number of rows and columns

\[ X_{(n \times n)} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & \cdots & x_{nn} \end{bmatrix} \]

**Symmetric (square) matrix**
- \( x_{ij} = x_{ji} \)
- For example,

\[ X_{(4 \times 4)} = \begin{bmatrix} 1 & 2 & 3 & 5 \\ 2 & 1 & 4 & 6 \\ 3 & 4 & 2 & 7 \\ 5 & 6 & 7 & 3 \end{bmatrix} \]
### Diagonal

The diagonal of $X$ includes elements $x_{ii}$

$x \leftarrow \text{diag}(X)$

<table>
<thead>
<tr>
<th>Diagonal matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only diagonal has non-zero values</td>
</tr>
</tbody>
</table>

$$X_{(diag)} = \begin{bmatrix} x_{11} & 0 & 0 & \ldots & 0 \\ 0 & x_{22} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & x_{nn} \end{bmatrix}$$

$x \leftarrow \text{diag}(x)$

### Identity matrix

A diagonal matrix where all diagonal elements are ones.

$$I_n = I_{(diag)} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

$I \leftarrow \text{diag}(\text{rep}(1,n))$  

### Block diagonal matrix

A block diagonal matrix has $m$ ($n_m$ by $n_m$) square matrices on the diagonal.

$$X_{(block)} = \begin{bmatrix} x_{11} & x_{21} & \ldots & 0 & 0 \\ x_{12} & x_{22} & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & x_{m-1,n-1} & x_{m,n} \end{bmatrix}$$

$$I \leftarrow \text{diag}(\text{rep}(1,n))$$

Example

---

Annika Kangas 7.11.2006
Product of a matrix and a scalar

\[ aX_{(n \times p)} = \begin{bmatrix}
    ax_{11} & ax_{12} & \cdots & ax_{1p} \\
    ax_{21} & ax_{22} & \cdots & ax_{2p} \\
    \vdots & \vdots & \ddots & \vdots \\
    ax_{n1} & ax_{n2} & \cdots & ax_{np}
\end{bmatrix} = X_{(2 \times 2)} \begin{bmatrix}
    0 & 0 \\
    0 & 0
\end{bmatrix} X_{(2 \times 2)} \]

Example

\[ 3X_{(3 \times 4)} = 3 \begin{bmatrix}
    1 & 4 & 7 & 10 \\
    2 & 5 & 8 & 11 \\
    3 & 6 & 9 & 12
\end{bmatrix} = \begin{bmatrix}
    3 & 12 & 21 & 30 \\
    6 & 15 & 24 & 33 \\
    9 & 18 & 27 & 36
\end{bmatrix} \]

Sum of matrices

The dimensions of the matrices need to be equal. For each element of the sum matrix

\[ z_{ij} = x_{ij} + y_{ij} \]
\[ X_{(n \times p)} + Y_{(n \times p)} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} + \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{np} \end{bmatrix} = \begin{bmatrix} x_{11} + y_{11} & x_{12} + y_{12} & \cdots & x_{1p} + y_{1p} \\ x_{21} + y_{21} & x_{22} + y_{22} & \cdots & x_{2p} + y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} + y_{n1} & x_{n2} + y_{n2} & \cdots & x_{np} + y_{np} \end{bmatrix} \]

Example

\[ X_{(3 \times 4)} + Y_{(3 \times 4)} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} = \begin{bmatrix} 2 & 6 & 10 & 14 \\ 7 & 11 & 15 & 19 \\ 12 & 16 & 20 & 24 \end{bmatrix} \]
Inner product of vectors

\[
Z_{1:p} = x_{1:p} ' y_{p:x} = \begin{bmatrix} x_1 & x_2 & x_3 & \ldots & x_p \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix} = \sum_{j=1}^{p} x_j y_j
\]

Example

\[
z_{1:4} = x_{1:4} ' y_{4:1} = [1 \ 2 \ 3 \ 4] = 5 + 2 \cdot 6 + 3 \cdot 7 + 4 \cdot 8
\]

\[= 5 + 12 + 21 + 32 = 70 \]

\[z <- x %*% y\]

Outer product of vectors

\[
Z_{p:x} = y_{p:1} x_{1:p} ' = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & \ldots \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} y_1 x_1 & y_1 x_2 & \ldots & y_1 x_p \\ y_2 x_1 & y_2 x_2 & \ldots & y_2 x_p \\ \vdots & \vdots & \vdots & \vdots \\ y_p x_1 & y_p x_2 & \ldots & y_p x_p \end{bmatrix}
\]
Example

\[
\begin{bmatrix}
5 \\
6 \\
7 \\
8
\end{bmatrix}
\begin{bmatrix}
1 & 2 & 3 & 4
\end{bmatrix}
= \begin{bmatrix}
5 & 5 & 5 & 4 \\
6 & 6 & 6 & 4 \\
7 & 7 & 7 & 4 \\
8 & 8 & 8 & 4
\end{bmatrix}
= \begin{bmatrix}
5 & 10 & 15 & 20 \\
6 & 12 & 18 & 24 \\
7 & 14 & 21 & 28 \\
8 & 16 & 24 & 32
\end{bmatrix}
\]

\[Z_{4\times 4} = y_{4\times 1}x_{4\times 4}^T\]

Product of matrices

\[
Z_{(m\times q)} = X_{(m\times p)}Y_{(p\times q)} = \begin{bmatrix}
x_{11} & x_{12} & \cdots & x_{1p} \\
x_{21} & x_{22} & \cdots & x_{2p} \\
\vdots & \vdots & \ddots & \vdots \\
x_{m1} & x_{m2} & \cdots & x_{mp}
\end{bmatrix}
\begin{bmatrix}
y_{11} & y_{12} & \cdots & y_{1q} \\
y_{21} & y_{22} & \cdots & y_{2q} \\
\vdots & \vdots & \ddots & \vdots \\
y_{p1} & y_{p2} & \cdots & y_{pq}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\sum_{j=1}^p x_{1j}y_{j1} & \sum_{j=1}^p x_{1j}y_{j2} & \cdots & \sum_{j=1}^p x_{1j}y_{jq} \\
\sum_{j=1}^p x_{2j}y_{j1} & \sum_{j=1}^p x_{2j}y_{j2} & \cdots & \sum_{j=1}^p x_{2j}y_{jq} \\
\vdots & \vdots & \ddots & \vdots \\
\sum_{j=1}^p x_{mj}y_{j1} & \sum_{j=1}^p x_{mj}y_{j2} & \cdots & \sum_{j=1}^p x_{mj}y_{jq}
\end{bmatrix}
\]
Example

\[ Z_{(2 \times 3)} = X_{(2 \times 3)} Y_{(3 \times 2)} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 7 & 10 \\ 8 & 11 \\ 9 & 12 \end{bmatrix} = \begin{bmatrix} 1 \cdot 7 + 2 \cdot 8 + 3 \cdot 9 & 1 \cdot 10 + 2 \cdot 11 + 3 \cdot 12 \\ 4 \cdot 7 + 5 \cdot 8 + 6 \cdot 9 & 4 \cdot 10 + 5 \cdot 11 + 6 \cdot 12 \end{bmatrix} \]

\[ = \begin{bmatrix} 7 + 16 + 27 & 10 + 22 + 36 \\ 28 + 40 + 54 & 40 + 55 + 72 \end{bmatrix} = \begin{bmatrix} 50 & 68 \\ 122 & 167 \end{bmatrix} \]

\[ Z = X \times Y \]

Multiplying a matrix by a diagonal matrix

\[ Z_{(m \times n)} = X_{(m \times o)} Y_{(o \times n)} = \begin{bmatrix} x_{11} & 0 & 0 & \cdots & 0 \\ 0 & x_{22} & 0 & \cdots & 0 \\ 0 & 0 & x_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & x_{nn} \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & \cdots & \cdots & y_{nn} \end{bmatrix} \]

\[ = \begin{bmatrix} x_{11} y_{11} & x_{11} y_{12} & \cdots & x_{11} y_{1m} \\ x_{22} y_{21} & x_{22} y_{22} & \cdots & x_{22} y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{nn} y_{n1} & \cdots & \cdots & x_{nn} y_{nn} \end{bmatrix} \]
Matrix inverse

Inverse matrix $X^{-1}$ is a square matrix which fulfills $XX^{-1} = I$

Inverse of a diagonal matrix is obtained by inverting the diagonal elements.

$$X^{-1}_{(n\times n)} = \begin{bmatrix} x_{11} & 0 & 0 & \ldots & 0 \\ 0 & x_{22} & 0 & \ldots & 0 \\ 0 & 0 & x_{33} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & \ldots & x_{nn} \end{bmatrix}^{-1} = \begin{bmatrix} x_{11}^{-1} & 0 & \ldots & 0 \\ 0 & x_{22}^{-1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & x_{nn}^{-1} \end{bmatrix}$$

Inverting a matrix is hard work even for small matrices.

For a 2 by 2 matrix, $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ the inverse is

$$A^{-1} = \begin{bmatrix} \frac{d}{ad - cb} & -\frac{c}{ad - cb} \\ \frac{-b}{ad - cb} & \frac{a}{ad - cb} \end{bmatrix}$$

Xinv<-solve(X)

Rules for matrix calculations

- $A(BC) = (AB)C$
- $(A+B)C = AC + BC$
- $AB \neq BA$
- $(AB)' = B'A'$
- $IA = AI = A$
- $(AB)^{-1} = B^{-1}A^{-1}$
Appendix B

A short story on R

R is a programming environment that includes lot of ready functions for statistical computing. Most users just use one or some of the most common functions, such as `lm` or `plot`.

The following page includes a R-reference card by Jonathan Baron.
Bibliography


BIBLIOGRAPHY


Mehtätalo, L., C. Comas, T. Pukkala, and M. Palahí. 2010. Improving predicted diameter distribution by using a small sample of diameters. Submitted MS.


