Lecture Notes

Institute of Bioinformatics, Johannes Kepler University Linz

Basic Methods of Data Analysis

Winter Semester 2014/2015

by Sepp Hochreiter

Institute of Bioinformatics
Johannes Kepler University Linz
A-4040 Linz, Austria

Tel. +43 732 2468 8880
Fax +43 732 2468 9308
http://www.bioinf.jku.at

Literature

- R. Peck, C. Olsen and J. L. Devore; Introduction to Statistics and Data Analysis, 3rd edition, ISBN: 9780495118732, Brooks/Cole, Belmont, USA, 2009.
- B. Shahbaba; Biostatistics with R: An Introduction to Statistics Through Biological Data; Springer, series UseR!, ISBN 9781461413011, New York, 2012.
- C. T. Ekstrøm and H. Sørensen; Introduction to Statistical Data Analysis for the Life Sciences; CRC Press, Taylor & Francis Group, ISBN: 9781439825556, Boca Raton, USA, 2011.
- A. Dobson; An Introduction to Generalized Linear Models, 2nd edition, ISBN: 1-58488-165-8, Series: Texts in Statistical Science, Chapman & Hall / CRC, Boca Raton, London, New York, Washington D.C., 2002.
- A. C. Rencher and G. B. Schaalje; Linear Models in Statistics, 2nd edition, Wiley, Hoboken, New Jersey, USA, 2008.
- L. Kaufman and P. J. Rousseeuw; Finding Groups in Data. An Introduction to Cluster Analysis, Wiley, 1990.

Contents

1 Introduction **1**

1.1 Machine Learning Introduction 1

1.2 Course Specific Introduction 2

1.3 Examples in R 3

1.4 Data-Driven or Inductive Approach 5

2 Representing Observations **7**

2.1 Feature Extraction, Selection, and Construction 7

2.2 Example 1: Anderson's Iris Data Set 12

2.3 Example 2: Multiple Tissues Microarray Data Set 14

2.4 Example 3: Breast Cancer Microarray Data Set 14

2.5 Example 4: Diffuse Large-B-Cell Lymphoma 14

2.6 Example 5: US Arrests 15

2.7 Example 6: EU Stock Markets 15

2.8 Example 7: Lung Related Deaths 15

2.9 Example 8: Sunspots 16

2.10 Example 9: Revenue Time Series 16

2.11 Example 10: Case-Control Study of Infertility 16

3 Summarizing Univariate and Bivariate Data **17**

3.1 Summarizing Univariate Data 17

3.1.1 Measuring the Center 17

3.1.2 Measuring the Variability 22

3.1.3 Summary Statistics 23

3.1.4 Boxplots 25

3.1.5 Histograms 30

3.1.6 Density Plots 31

3.1.7 Violin Plots 34

3.2 Summarizing Bivariate Data 36

3.2.1 Scatter Plot 36

3.2.2 Correlation 43

3.2.3 Test for Correlation 44

3.2.4 Linear Regression 45

4 Summarizing Multivariate Data 51
4.1 Matrix of Scatter Plots . . . 54
4.2 Principal Component Analysis . . . 55
4.2.1 The Method . . . 55
4.2.2 Variance Maximization . . . 58
4.2.3 Uniqueness . . . 58
4.2.4 Properties of PCA . . . 60
4.2.5 Examples . . . 63
4.2.5.1 Iris Data Set . . . 63
4.2.5.2 Multiple Tissue Data Set . . . 64
4.3 Clustering . . . 76
4.3.1 k-Means Clustering . . . 76
4.3.1.1 The Method . . . 76
4.3.1.2 Examples . . . 78
4.3.2 Hierarchical Clustering . . . 95
4.3.2.1 The Method . . . 95
4.3.2.2 Examples . . . 96

5 Linear Models 107
5.1 Linear Regression . . . 107
5.1.1 The Linear Model . . . 107
5.1.2 Interpretations and Assumptions . . . 108
5.1.2.1 Interpretations . . . 108
5.1.2.2 Assumptions . . . 111
5.1.3 Least Squares Parameter Estimation . . . 112
5.1.4 Evaluation and Interpretation of the Estimation . . . 113
5.1.4.1 Residuals and Error Variance . . . 113
5.1.4.2 Coefficient of determination . . . 114
5.1.4.3 Outliers and Influential Observations . . . 114
5.1.4.3.1 Outliers. . . . 114
5.1.4.3.2 Influential Observations. . . . 115
5.1.5 Confidence Intervals for Parameters and Prediction . . . 117
5.1.5.1 Normally Distributed Error Terms . . . 117
5.1.5.2 Error Term Distribution Unknown . . . 118
5.1.6 Tests of Hypotheses . . . 119
5.1.6.1 Test for a Set of Variables Equal to Zero . . . 119
5.1.6.2 Test for a Single Variable Equal to Zero . . . 120
5.1.7 Examples . . . 121
5.1.7.1 Hematology Data . . . 121
5.1.7.1.1 Computing Estimates, Confidence Intervals, Tests. . . 123
5.1.7.1.2 Using Predefined R Functions. . . . 126
5.1.7.2 Carbohydrate Diet Data . . . 128
5.2 Analysis of Variance . . . 131
5.2.1 One Factor . . . 131
5.2.2 Two Factors . . . 133
5.2.3 Examples . . . 138

5.2.3.1 Dried Plant Weights 138
5.2.3.2 Extended Dried Plants 142
5.2.3.3 Two-Factor ANOVA Toy Example 144
5.3 Analysis of Covariance 147
5.3.1 The Model 147
5.3.2 Examples 148
5.3.2.1 Achievement Scores 148
5.3.2.2 Birthweights of Girls and Boys 151
5.4 Mixed Effects Models 156
5.4.1 Approximative Estimator 156
5.4.1.1 Estimator for Beta 157
5.4.1.2 Estimator for u 158
5.4.2 Full Estimator 159
5.5 Generalized Linear Models 159
5.5.1 Logistic Regression 163
5.5.1.1 The Model 163
5.5.1.2 Maximizing the Likelihood 164
5.5.2 Multinomial Logistic Regression: Softmax 166
5.5.3 Poisson Regression 167
5.5.4 Examples 168
5.5.4.1 Birthweight Data: Normal 168
5.5.4.2 Beetle Mortality: Logistic Regression 172
5.5.4.3 Embryogenic Anthers: Logistic Regression 175
5.5.4.4 Toy Example 1: Poisson Regression 180
5.5.4.5 Toy Example 2: Poisson Regression 183
5.5.4.6 Detergent Brand: Poisson Regression 185
5.5.4.7 Tumor Data: Poisson Regression 188
5.5.4.8 Ulcers and Aspirin Use: Logistic Regression 191
5.6 Regularization 196
5.6.1 Partial Least Squares Regression 196
5.6.2 Ridge Regression 197
5.6.3 LASSO 200
5.6.4 Elastic Net 201
5.6.5 Examples 202
5.6.5.1 Example: Ridge Regression, LASSO, Elastic Net 202
5.6.5.2 Example: Diabetes using Least Angle Regression 208
5.6.5.3 Example: Relevant Variable but No Correlation to Response . . 211
5.6.5.4 Example: Irrelevant Variable but High Correlation to Response 212
5.6.5.5 Gas Vapor: Ridge Regression and LASSO 214
5.6.5.6 Chemical Reaction: Ridge Regression and LASSO 219
5.6.5.7 Land Rent: Ridge Regression and LASSO 224

A Mean and Median of Symmetric Distributions 229
A.1 Mean, Median, and Symmetry Point are Equal 229
A.2 Definitions of Mean, Median, and Variance 230
A.3 Mean Squared Error for Estimating Mean and Median 231

A.3.1 Gaussian: Variance of the Mean and the Median 231
A.3.2 Laplace: Variance of the Mean and the Median 232
A.4 Variance of the Variance 232
A.4.1 Biased Variance 232
A.4.2 Unbiased Variance 232
A.4.3 Gaussian Distribution 232
A.4.4 Laplace Distribution 233

B R Code for Plots in Manuscript 235
B.1 Histograms 235
B.2 Densities 236

List of Figures

2.1 Relevant vs. irrelevant feature . . . 8
2.2 The design cycle for machine learning . . . 9
2.3 Combination of features help . . . 10
2.4 Features with different variance . . . 10
2.5 Iris Flowers . . . 12
2.6 Flowerparts petal and septal . . . 13

3.1 Comparison of mean, median, and mode . . . 20
3.2 Quartiles . . . 23
3.3 Boxplot for iris sepal length . . . 26
3.4 Boxplots of sepal length per iris species . . . 27
3.5 Boxplot iris petal length . . . 28
3.6 Boxplots of petal length per iris species . . . 29
3.7 Histograms of sepal and petal lengths . . . 30
3.8 Kernel density estimator . . . 31
3.9 Densities of sepal lengths per species . . . 33
3.10 Densities of petal lengths per species . . . 33
3.11 Violin plots of sepal lengths . . . 34
3.12 Violin plots of petal lengths . . . 35
3.13 Scatter plot . . . 37
3.14 Scatter plots of linearly dependent features . . . 38
3.15 Scatter plot of noisy linearly dependent features . . . 39
3.16 Scatter plots of linearly dependent and independent features . . . 40
3.17 Scatter plot of non-linearly dependent features . . . 41
3.18 Matrix of scatter plots . . . 42
3.19 Linear regression of bivariate data . . . 45
3.20 Outlier in regression . . . 48
3.21 Anscombe's 4 regression data sets . . . 50

4.1 The descriptive framework . . . 52
4.2 The generative framework . . . 53
4.3 Matrix of scatter plots for the iris data . . . 54
4.4 Principal component analysis for a two-dimensional data set . . . 55
4.5 Principal component analysis: projection . . . 56
4.6 PCA applied to the iris data . . . 65
4.7 PCA for multiple tissues . . . 69
4.8 PCA for multiple tissues with 101 genes . . . 70

4.9 PCA for multiple tissues with 13 genes 71
4.10 PCA for multiple tissues with 5 genes 72
4.11 PCA for multiple tissues with 10 genes 73
4.12 PCA for multiple tissues: feature selection by clustering 74
4.13 PCA for multiple tissues: feature selection by corr. clust. 75
4.14 k-means clustering: five cluster, optimal 80
4.15 k-means clustering: five cluster, local minimum (1) 81
4.16 k-means clustering: five cluster, local minimum (2) 82
4.17 k-means clustering: five cluster, local minimum (3) 83
4.18 k-means clustering: five cluster, optimal 84
4.19 k-means clustering: five cluster, local minimum (1) 85
4.20 k-means clustering: five cluster, local minimum (2) 86
4.21 k-means clustering: five cluster, local minimum (3) 87
4.22 PCA applied to the Iris data set 88
4.23 k-means clustering of the Iris data set (1) 89
4.24 k-means clustering of the Iris data set (2) 90
4.25 PCA applied to the multiple tissues data set 91
4.26 k-means clustering of the multiple tissue data set (1) 92
4.27 k-means clustering of the multiple tissue data set (2) 93
4.28 k-means clustering of the multiple tissue data set (3) 94
4.29 Dendrogram after hierarchical clustering of animal species 95
4.30 Hierarchical clustering US Arrests: Ward 97
4.31 Hierarchical clustering US Arrests: single linkage 98
4.32 Hierarchical clustering US Arrests: complete linkage 99
4.33 Hierarchical clustering US Arrests: average linkage 100
4.34 Hierarchical clustering US Arrests: McQuitty 101
4.35 Hierarchical clustering US Arrests: median 102
4.36 Hierarchical clustering US Arrests: centroid 103
4.37 Hierarchical clustering: five cluster 104
4.38 Hierarchical clustering: iris data 105
4.39 Hierarchical clustering: multiple tissues data 106

5.1 Influential Observations 116
5.2 Carbohydrate vs. Protein 130
5.3 Dobson's dried plant data 139
5.4 Results of ANOVA for dried plant data 141
5.5 Dobson's extended dried plant data 143
5.6 Two-factor ANOVA toy data 145
5.7 Scatter plot achievement scores data 149
5.8 Scatter plot birthweight data 152
5.9 The sigmoid function 163
5.10 Dobson's beetle data 173
5.11 Fitting of Dobson's beetle data 176
5.12 Dobson's embryogenic anther data 178
5.13 Best model for Dobson's embryogenic anther data 181
5.14 Toy Poisson Regression data 182

5.15 Poisson regression for randomized controlled trial 184
5.16 Dobson's malignant melanoma data 190
5.17 Dobson's gastric and duodenal ulcers and aspirin use 192
5.18 LASSO vs. ridge regression 200
5.19 Example correlated explanatory variables 203
5.20 Example of ridge regression 204
5.21 Example of LASSO 206
5.22 Example of elastic net 207
5.23 Example of least angle regression 209
5.24 Example: relevant Variable with no target correlation 213
5.25 Example: irrelevant Variable has high target correlation 215

List of Tables

2.1 Feature-target correlations 11
2.2 Part of the iris data set 13

3.1 Overview of mean, median, and mode 19

5.1 ANOVA table 120
5.2 Rencher’s hematology data 122
5.3 Dobson’s carbohydrate diet data 129
5.4 Data of dried plant weights 138
5.5 Extended dried plant weights 142
5.6 Two-factor ANOVA toy data 145
5.7 Achievement scores data 150
5.8 Birthweight data 151
5.9 Common GLM models 162
5.10 Count data distribution 167
5.11 Dobson’s beetle data 172
5.12 Dobson’s embryogenic anther data 177
5.13 Poisson regression for randomized controlled trial 183
5.14 Ries & Smith (1963) data 185
5.15 Dobson’s malignant melanoma data 189
5.16 Dobson’s gastric and duodenal ulcers and aspirin use 192
5.17 Example: relevant Variable with no target correlation 211
5.18 Example: irrelevant Variable has high target correlation 212
5.19 Rencher’s gas vapor data 216
5.20 Rencher’s chemical reaction data 220
5.21 Rencher’s land rent data 225

List of Algorithms

4.1 k-means 77
4.2 Fuzzy k-means 78
5.1 Partial least squares regression 198

Chapter 1

Introduction

1.1 Machine Learning Introduction

This course is part of the curriculum of the master in computer science (in particular the majors “Computational Engineering” and “Intelligent Information Systems”) and part of the master in bioinformatics at the Johannes Kepler University Linz.

Machine learning is currently a major research topic at companies like Google, Microsoft, Amazon, Facebook, AltaVista, Zalando, and many more. Applications are found in computer vision (image recognition), speech recognition, recommender systems, analysis of Big Data, information retrieval. Companies which have their domain in the world wide web like companies offering search engines, social networks, videos, information, or connecting people with specific interest use machine learning techniques to analyze their data. Machine learning methods are used to annotate web pages, images, videos, and sound recordings in web data. They can find specific objects in images and detect a particular music style if only given the raw data. Therefore Google, Microsoft, Facebook are highly interested in machine learning methods. Machine learning methods attracted the interest of companies offering products via the web. These methods are able to identify groups of similar users, to predict future behavior of customers, and can give recommendation of products in which customers will be interested based previous costumer data.

Machine learning has major applications in biology and medicine. Modern measurement techniques in both biology and medicine create a huge demand for new machine learning approaches. One such technique is the measurement of mRNA concentrations with microarrays and sequencing techniques. The measurement data are first preprocessed, then genes of interest are identified, and finally predictions made. Further machine learning methods are used to detect alternative splicing, nucleosome positions, gene regulation, etc. Alongside neural networks the most prominent machine learning techniques relate to support vector machines, kernel approaches, projection method and probabilistic models like latent variable models. These methods provide noise reduction, feature selection, structure extraction, classification / regression, and assist modeling. In the biomedical context, machine learning algorithms categorize the disease subtype or predict treatment outcomes based on DNA characteristics, gene expression profiles. Machine learning approaches classify novel protein sequences into structural or functional classes. For analyzing data of association studies, machine learning methods extract new dependencies between DNA markers (SNP - single nucleotide polymorphisms, SNV - single nucleotide variants, CNV - copy number variations) and diseases (Alzheimer, Parkinson, cancer, multiples sclerosis, schizophrenia or alcohol dependence).

The machine learning course series comprises:

- “Basic Methods of Data Analysis”: this course gives a smooth introduction to machine learning with examples in R ; it covers summary statistics (mean, variance), data summary plots (boxplot, violin plot, scatter plot), basic methods to analyze multivariate data (principal component analysis and clustering), and linear models (least squares regression, ANOVA, ANCOVA, logistic regresssion, Poisson regression, Ridge regression, LASSO, elastic net).

- “Machine Learning: Supervised Methods”: classification and regression techniques, time series prediction, kernel methods, support vector machines, neural networks, deep learning, deep neural and belief networks, ARMA and ARIMA models, recurrent neural networks, LSTM

- “Machine Learning: Unsupervised Methods”: maximum likelihood estimation, maximum a posterior estimation, maximum entropy, expectation maximization, principal component analysis, statistical independence, independent component analysis, independent component analysis, multidimensional scaling (Kruskal’s or Sammon’s map), locally linear embedding, Isomap, hierarchical clustering, mixture models, k-means, similarity based clustering (affinity propagation), biclustering, factor analysis, sparse codes, population codes, kernel PCA, hidden Markov models (factorial HMMs and input-output HMMs), Markov networks and random fields, restricted Boltzmann machines, auto-associators, unsupervised deep neural networks

- “Theoretical Concepts of Machine Learning”: estimation theory (unbiased and efficient estimator, Cramer-Rao lower bound, Fisher information matrix), consistent estimator, complexity of model classes (VC-dimension, growth, annealed entropy), bounds on the generalization error, Vapnik and worst case bounds on the generalization error, optimization (gradient based methods and convex optimization), Bayes theory (posterior estimation, error bounds, hyperparameter optimization, evidence framework), theory on linear functions (statistical tests, intervals, ANOVA, generalized linear functions, mixed models)

In this course the most prominent machine learning techniques are introduced and their mathematical basis and derivatives are explained. If the student understands these techniques, then the student can select the methods which best fit to the problem at hand, the student is able to optimize the parameter settings for the methods, the student can adapt and improve the machine learning methods, and the student can develop new machine learning methods.

Most importantly, students should learn how to chose appropriate methods from a given pool of approaches for solving a specific problem. To this end, they must understand and evaluate the different approaches, know their advantages and disadvantages as well as where to obtain and how to use them. In a step further, the students should be able to adapt standard algorithms for their own purposes or to modify those algorithms for particular applications with certain prior knowledge or problem-specific constraints.

1.2 Course Specific Introduction

Data analysis and visualization are essential to most fields in science and engineering. The goal of this course is to provide students with a basic tool chest of methods for pre-processing, analyzing, and visualizing scientific data.

This course introduces basic concepts of data analysis by examples. The theoretical background of the introduced methods is not covered in this course but in other courses.

1.3 Examples in R

The examples are presented in R . In order to present plots and graphics it is necessary to use a software environment, where the methods are ready to use and the results can be visualized. For the course it is not necessary to install R on your computer. However, to try out the methods, play with different parameter settings, and to get a feeling for the methods, we recommend to install R on your computer.

R is free and an implementation of the **S** language which has been used by statisticians and data analysts for two decades. R is probably the most widely used software tool for bioinformatics and became popular due to its data handling (e.g. importing microarray data), statistical algorithms, machine learning / data modeling implementations and integrated data visualization. One of the largest sources of R tools for bioinformatics is the Bioconductor Project (www.bioconductor.org) which will be utilized in this course. These days R is increasingly popular in machine learning even outside bioinformatics e.g. for modeling the financial market or for forecasting.

R has the advantages:

- it is free and open source with a large community,
- it is flexible and extensible,
- it has implementations of major machine learning and statistical methods,
- it has graphics for data visualization,
- it has convenient data handling tools, and
- it has matrix and vector calculation tools.

To install R , you should download it from http://cran.r-project.org/ for Windows, Linux, or MacOS X. First, the R base has to be installed and then the packages. The R base can be found at the main web page but also under “R Sources” (the R source code to compile) and under “R Binaries” (pre-compiled binary files). The packages are located in “Packages” and can be installed subsequently.

For Linux, you have to decide whether to install binaries or the source. To install the binaries, one has to know the Linux installation (Ubuntu, RedHat, Debian, Suse). The source code is `R-2.10.1.tar.gz` and can be compiled using “make” and “Makefiles”. After installation, you can run R in the command line by typing `R`.

For Windows, we recommend to install the binaries. The binary for the R base is `R-*.exe` which is an executable and installs itself. After installation, you can run the RGui by a desktop click or run R in the command line by typing `R`.

R is installed with the following manuals:

- "An Introduction to R "

- "The R Language Definitions"
- "R Installation and Administration"
- "Writing R Extensions"
- "R Data Import and Export"

The packages are in the following repositories:

1. CRAN
2. CRAN (extras)
3. BioC software
4. BioC annotation
5. BioC experiment
6. BioC extra
7. R-Forge

“CRAN” and “R-Forge” supply R packages and “BioC” supplies the Bioconductor packages (http://www.bioconductor.org).

For installing the packages, first the repositories, where the packages are found must be chosen. This can be done by the R function `setRepositories()` which is called by `setRepositories()` at the command line. Then a Gui should open, where you can select the repositories.

To install the packages, use the R function `install.packages()` by typing `install.packages()` into the command line. Then a Gui should open which asks for the mirror, where you can choose the Austrian mirror http://cran.at.r-project.org/.

Alternatively, you can also install the Bioconductor packages by command lines in R :

```
R> source("http://www.bioconductor.org/biocLite.R")
R> biocLite("PACKAGE-NAME")
```

To install the packages under a Windows system, you can go to the menu “packages” and first choose a download site by the button “Set CRAN mirror” or by the command `chooseCRANmirror()`. For example, the Austrian mirror http://cran.at.r-project.org/ may be chosen.

Then set the repositories by the button “Select repositories” in the menu “packages” or by the command `setRepositories()`.

The packages can be installed by the according button “Install packages” in the menu “packages” or by the command `install.packages()`.

Alternatively, you can also install the Bioconductor packages by command lines in R :

```
R> source("http://www.bioconductor.org/biocLite.R")
R> biocLite("PACKAGE-NAME")
```

1.4 Data-Driven or Inductive Approach

The conventional approach to solve problems with the help of computers is to write programs which solve the problem. With this approach the programmer must understand the problem, find a solution appropriate for the computer, and implement this solution on the computer. We call this approach *deductive* because the human deduces the solution from the problem formulation. However in biology, chemistry, biophysics, medicine, and other life science fields a huge amount of data is produced which is hard to understand and to interpret by humans. A solution to a problem may be found by statistical methods or by a machine that learns. Statistics tries to explain variability in the data in order to draw conclusions and to make decisions. Machine learning tries to find patterns and structures in the data. Statistical and machine learning methods automatically find descriptions of the data, regularities in the data, characteristics of the data, and dependencies between parts of the data. The knowledge about the extracted characteristics, regularities, and structures can be used to solve the problem at hand. We call this approach *inductive*. Statistics and machine learning is about inductively solving problems by using computers.

In this course we present tools and basic techniques for analyzing data with statistical and machine learning methods. We demonstrate these tools with examples and explain how the results should be interpreted. Furthermore, we discuss strengths and weaknesses of the approaches for different kinds of data. In particular we discuss which tool should be used for which kind of tasks and for which kind of data.

Chapter 2

Representing Observations

Observations and measurements of the real world objects are represented as data on a computer. Subsequently, these data are analyzed to explain variation and to find structures in the data. These analyzes allow to perform forecasting and classification, where the outcome of measurements of the objects and future events are predicted. On the other hand, data analyzes allows to characterize and categorize the objects and find relation between them or to other objects: they reveal unknown states of the objects, relations between the objects, processes involving the objects and other objects, etc.

2.1 Feature Extraction, Selection, and Construction

Features or characteristics of objects must be extracted from the original data that are obtained from measurements or recordings of the objects. The process of generating features from the raw data is called *feature extraction*. In many applications features are directly measured, like length, weight, etc. However, for other applications feature extraction is necessary like the extraction of features from an image, like the length or width of an object in an image.

For some tasks a huge number of features is available if they are automatically measured or extracted. Examples in bioinformatics are data produced by the RNA microarray technique which contains simultaneous measurements of the expression level of 20,000 genes Hochreiter et al. [2006]. The number of features are even larger for genotyping measurements by cDNA arrays Clevert et al. [2011] or by next generation sequencing Klambauer et al. [2012, 2013]. Further examples are bio-techniques, like peptide arrays, protein arrays, mass spectrometry, etc., which produce high-dimensional data Mahrenholz et al. [2011], Schwarzbauer et al. [2012]. The techniques simultaneously measure many features as they are not designed for a specific task. Thus, for a specific task many measurements may be irrelevant. For example, only a few genes may be important for a specific biological question, e.g. detecting cancer related genes or predicting the outcome of a cancer therapy Talloen et al. [2007], Kasim et al. [2010], Talloen et al. [2010]. To chose relevant features to solve the task at hand is called *feature selection* Hochreiter and Obermayer [2004, 2005]. An feature selection example is given in Fig. 2.1, where one variable is related to the classification task and the other is not. Feature selection is a very important step in bioinformatics both to construct appropriate models but also, sometimes more importantly, to gain insight into biological processes.

The first step of data analysis is to select the relevant features or chose a model which automatically identifies the relevant features. Fig. 2.2 shows the design cycle for generating a model

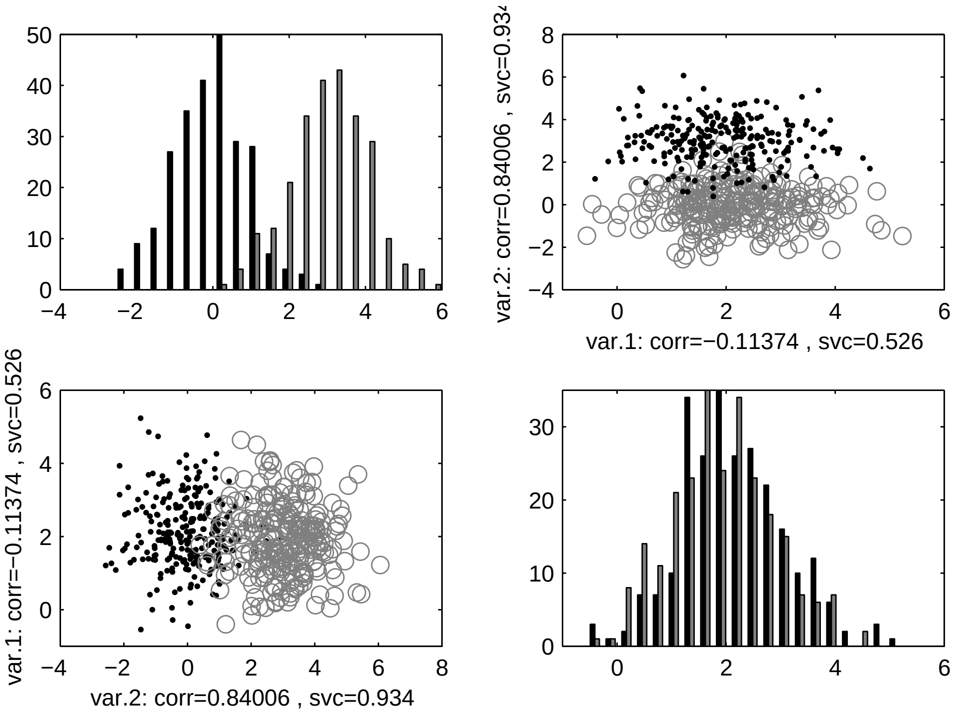

Figure 2.1: Relevant vs. irrelevant feature. A simple classification problem with two features. Feature 1 (var. 1) is noise and feature 2 (var. 2) is correlated to the classes. Between the upper right figure and lower left figure only the axis are exchanged. The upper left figure gives the class histogram along feature 2, whereas the lower right figure gives the histogram along feature 1. The correlation to the class (corr) and the performance of the single variable classifier (svc) is given. Copyright © 2006 Springer-Verlag Berlin Heidelberg.

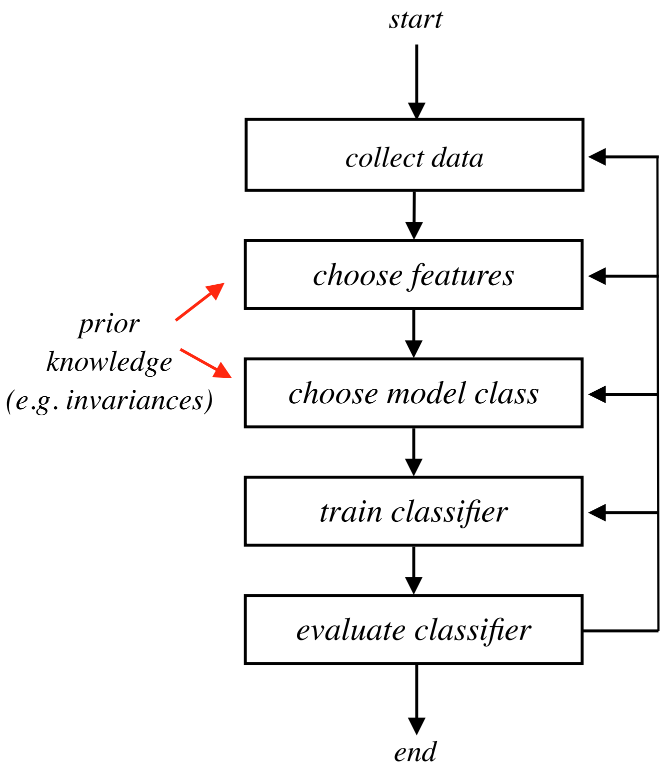

Figure 2.2: The design cycle for machine learning in order to solve a certain task. Copyright © 2001 John Wiley & Sons, Inc.

with statistical or machine learning methods. After collecting the data and extracting the features, the relevant features are selected.

The problem of selecting the right features / variables can be difficult. Fig. 2.3 shows an example where single features cannot improve the classification performance but a combination of features improves it. Fig. 2.4 shows an example where the feature's mean values and variances in the left sub-figure are equal to the same values in the right sub-figure. However, the direction of the variances of the features differs in the sub-figures which leads to different performances in classification.

Features that are not correlated with the target can be important for constructing a good model and, therefore, should be selected. On the other hand, features with large correlation to the target may be not important for constructing a good model and should not be selected. For example, given the values of the left hand side in Tab. 2.1, the target t is computed from two features f_1 and f_2 as $t = f_1 + f_2$. All values have mean zero and the correlation coefficient between t and f_1 is zero. In this case f_1 should be selected because it has negative correlation with f_2. The top ranked feature may not be correlated to the target, e.g. if it contains target-independent information which can be removed from other features. The right hand side of Tab. 2.1 depicts another situation, where $t = f_2 + f_3$. f_1, the feature which has highest correlation coefficient with the target (0.9 compared to 0.71 of the other features) should not be selected because it is correlated to all other

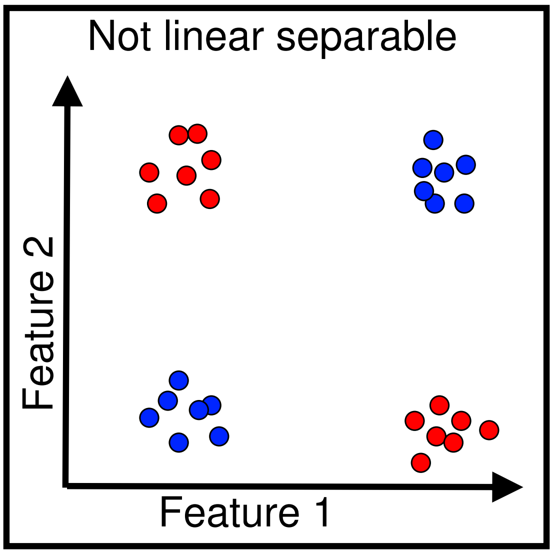

Figure 2.3: Only a combination of features helps. An XOR problem with two features, where each single feature is neither correlated to the outcome nor helpful for classification. Only both features together improve the classification performance.

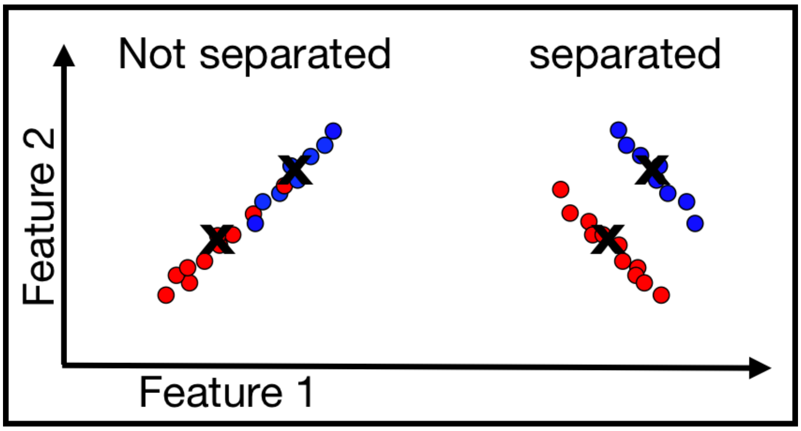

Figure 2.4: Features with different variance. The left and right sub-figure both show two classes. The feature's mean values and variances are equal in both sub-figures. However, the direction of the variances of the features differs in the sub-figures leading to different performances in classification.

f_1	f_2	t	f_1	f_2	f_3	t
-2	3	1	0	-1	0	-1
2	-3	-1	1	1	0	1
-2	1	-1	-1	0	-1	-1
2	-1	1	1	0	1	1

Table 2.1: Examples of features-target correlations. Left hand side: the target t is $t = f_1 + f_2$, however f_1 is not correlated with t. Right hand side: $t = f_2 + f_3$, however f_1 has highest correlation coefficient with the target.

features.

In some tasks it is helpful to combine some features to a new feature, that is to construct features. To create new features from the existing features is called *feature construction*. For example, for gene expression data combining values to a meta-gene value gives more robust features because the noise is "averaged out". In this example, genes that are always activated simultaneously because they belong to the same pathway may be combined to a more robust feature which corresponds to a pathway activation. A popular way of feature construction is principal component analysis (PCA) or independent component analysis (ICA). These linear feature construction or projection methods will be introduced later. The components of PCA or ICA are new constructed features, which, however, are often difficult to interpret. PCA and ICA can be generalized to non-linear feature construction methods Hochreiter and Schmidhuber [1997a,c, 1998, 1999a,d,b,c].

Another way to construct features is to use kernel methods. The original vectors of features are mapped into another space where implicitly new features are created. Besides this very generic kernel-based way, often non-linear features are constructed using prior knowledge of the problem to solve. For example, in bioinformatics a sequence of nucleotides or amino acids may be characterized by their similarity to other sequences. To compute these similarities alignment methods that include biological knowledge are used. These similarity scores are non-linear functions of the original sequence elements Hochreiter et al. [2007].

For some tasks objects are not measured individually but together with another object. We encountered such an example by alignment algorithms in bioinformatics, where a sequence is characterized by its similarity to other sequences. Another example are web pages which are described by their link structure: each link is described by two web pages. If an article is described by its citations of other articles, then we have again pairs of objects. Very prominent in these days are social networks were a user is described by her interactions with other users. These data are dyadic data where objects are characterized by their relation to other objects and for which specialized machine learning methods were developed Hochreiter and Obermayer [2002, 2003, 2006].

In the next sub-sections some data sets are presented in order to give an intuition of the kind of features that typically appear in real world data.

Figure 2.5: Examples of Iris flowers.

2.2 Example 1: Anderson's Iris Data Set

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by Sir Ronald Fisher (1936) as an example of discriminant analysis Fisher [1936]. Iris is a genus of 260–300 species of flowering plants with showy flowers (see Fig. 2.5). The name stems from the Greek word for a rainbow, as the flower colors have a broad variety. The three species of the data set are Iris setosa (Beachhead Iris), Iris versicolor (Larger Blue Flag, Harlequin Blueflag), and Iris virginica (Virginia Iris). This data set is sometimes called Anderson's Iris data set because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species Anderson [1935]. Two of the three species were collected in the Gaspe Peninsula "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus" Anderson [1935].

Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters(see Fig. 2.6). For each of the three species 50 flowers were measured (see part of the data in Tab. 2.2). Based on these four features, Fisher developed a linear discriminant model to distinguish the species from each other.

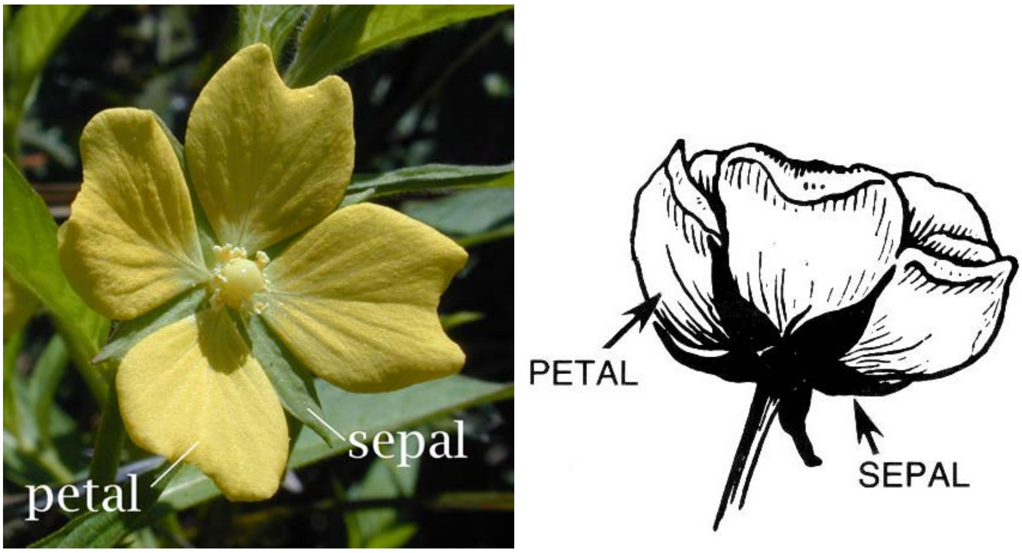

Figure 2.6: Flowerparts petal and septal are depicted and marked.

Table 2.2: Part of the iris data set with features sepal length, sepal width, petal length, and petal width.

No.	Sepal		Petal		Species
	Length	Width	Length	Width	
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
54	5.5	2.3	4.0	1.3	versicolor
55	6.5	2.8	4.6	1.5	versicolor
101	6.3	3.3	6.0	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica
103	7.1	3.0	5.9	2.1	virginica
104	6.3	2.9	5.6	1.8	virginica
105	6.5	3.0	5.8	2.2	virginica

2.3 Example 2: Multiple Tissues Microarray Data Set

This data set consists of microarray data from the Broad Institute "Cancer Program Data Sets" which was produced by Su et al. [2002]. The data contains gene expression profiles from human and mouse samples across a diverse set of tissues, organs, and cell lines. The goal was to have a reference for the normal mammalian transcriptome. The microarray platforms were Affymetrix human (U95A) or mouse (U74A) high-density oligonucleotide arrays. The authors profiled the gene expression level from 102 human and mouse samples and selected 5,565 genes. Gene selection is an important first step when analyzing microarray data Hochreiter and Obermayer [2004, 2005], Talloen et al. [2007], Kasim et al. [2010], Talloen et al. [2010].

The samples predominantly come from a normal physiological state in the human and the mouse. The data set represents a preliminary, but substantial, description of the normal mammalian transcriptome. Mining these data may reveal insights into molecular and physiological gene function, mechanisms of transcriptional regulation, disease etiology, and comparative genomics. Hoshida et al. [2007] used this data set to identify subgroups in the samples by using additional data of the same kind. The four distinct tissue types are:

- breast (Br),
- prostate (Pr),
- lung (Lu),
- and colon (Co).

These tissue types are indicated in the data set.

2.4 Example 3: Breast Cancer Microarray Data Set

Also this data set consists of microarray data from the Broad Institute "Cancer Program Data Sets" which was produced by van't Veer et al. [2002]. Goal of van't Veer et al. [2002] was to find a gene signature to predict the outcome of a cancer therapy, that is, to predict whether metastasis will occur. A signature of 70 genes has been discovered. We removed array S54, because we identified it as an outlier. Thereafter, the data set contains 97 samples for which 1213 gene expression values are provided — these genes were selected by the authors.

Hoshida et al. [2007] found 3 subclasses and verified that 50/61 cases from class 1 and 2 were estrogen receptor (ER) positive and only 3/36 from class 3. The subclasses were reconfirmed by an independent second breast cancer data set. The three subclasses are indicated in the data set.

2.5 Example 4: Diffuse Large-B-Cell Lymphoma

Another microarray data set from the Broad Institute "Cancer Program Data Sets" which was produced by Rosenwald et al. [2002]. The gene expression profile of diffuse large-B-cell lymphoma (DLBCL) was measured. Goal was to predict the survival after chemotherapy. The data set consists of 180 DLBCL samples with 661 preselected genes.

Hoshida et al. [2007] divided the data set into three subclasses:

- “OxPhos” (oxidative phosphorylation),
- “BCR” (B-cell response), and
- “HR” (host response).

These subclasses were confirmed on independent DLBCL data. We mark these subclasses in the data set.

2.6 Example 5: US Arrests

This data set contains statistics, in arrests per 100,000 residents, for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas.

A data consists of 50 observations with 4 features / variables:

- Murder: Murder arrests (per 100,000),
- Assault: Assault arrests (per 100,000),
- UrbanPop: Percent urban population,
- Rape: Rape arrests (per 100,000).

2.7 Example 6: EU Stock Markets

Time series are another kind of data sets. Specific to time series is that the observations are dependent. Therefore it is possible to use a model which directly accesses past data or which memorizes past data in order to predict future observations Hochreiter and Schmidhuber [1997b,d], Hochreiter [1997].

This data set contains the daily closing prices of major European stock indices: Germany DAX (Ibis), Switzerland SMI, France CAC, and UK FTSE. The data are sampled in business time, i.e., weekends and holidays are omitted. The data were kindly provided by Erste Bank AG, Vienna, Austria. A multivariate time series with 1860 observations and 4 variables that correspond to the 4 stock indices is provided.

2.8 Example 7: Lung Related Deaths

Three time series giving the monthly deaths from lung related diseases bronchitis, emphysema and asthma in the UK during 1974–1979. The counts are for both sexes, only males, and only females.

2.9 Example 8: Sunspots

Monthly mean relative sunspot numbers from 1749 to 1983. Collected at Swiss Federal Observatory, Zürich until 1960, then Tokyo Astronomical Observatory. During each month the number of sunspots are counted.

2.10 Example 9: Revenue Time Series

Freeny's data on quarterly revenue and explanatory variables. The time series has 39 observations on quarterly revenue (lagged 1Q) from 1962, 2Q, to 1971, 4Q. Furthermore, explanatory variables are supplied which include price index, income level, and market potential.

2.11 Example 10: Case-Control Study of Infertility

The data reports infertility after spontaneous and induced abortion. The data is from a matched case-control study.

1. education:
   - 0 = 0-5 years
   - 1 = 6-11 years
   - 2 = 12+ years
2. age: age in years of case
3. parity count
4. number of prior induced abortions
   - 0 = 0
   - 1 = 1
   - 2 = 2 or more
5. case status:
   - 1 = case
   - 0 = control
6. number of prior spontaneous abortions:
   - 0 = 0
   - 1 = 1
   - 2 = 2 or more
7. stratum

Chapter 3

Summarizing Univariate and Bivariate Data

In this chapter we focus on the two most simple cases of data: univariate data and bivariate data. Univariate data is only a set of numbers, that is, a set of scalars. Each number corresponds to an observation. The observations for bivariate data are described by a pair of numbers, that is, each observations is described by two values.

3.1 Summarizing Univariate Data

We present tools to summarize univariate data, that is, a set of numbers or a set of scalars. Such data are produced by single measurements of for example the weight, the height, the amplitude, the temperature, etc. Instead of reporting all data points, in most cases it is more appropriate to summarize the data and report numerical values like where are the data located (the center), what is the variability of the data etc.

3.1.1 Measuring the Center

A univariate data set $\boldsymbol{x}$ consists of a set of scalar numbers $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$. For a discrete probability distribution of a random variable X, the *mean* or *expected value* is equal to the sum over every possible value weighted by the probability of that value; that is, it is computed by taking the product of each possible value x of X and its probability $\Pr(x)$, and then adding all these products together:

$$\mu \; = \; \sum_{x \in X} x \, \Pr(x) \; . \tag{3.1}$$

For continuous probability distributions we have

$$\mu \; = \; \int_X x \, \Pr(x) \; dx \; , \tag{3.2}$$

where $\Pr(x)$ is a density.

The *sample mean*, *empirical mean*, or *arithmetic mean* of samples $(x_1, x_2, \ldots, x_n)$ is

$$\bar{x} \; = \; \frac{1}{n} \sum_{i=1}^{n} x_i \; . \tag{3.3}$$

The sample mean approximates the mean / expected value. Other means are the geometric mean (n-th root of the product of the numbers) and harmonic mean. The arithmetic mean is larger or equal to the geometric mean which is in turn larger or equal to the harmonic mean.

The *median* separates the higher half of a data from the lower half or, in the continuous case, is the value, where the probability mass is 0.5. For a finite set of samples, the *sample median* can be found by sorting the samples and then selecting the middle sample for an odd number of samples, or calculating the mean of the two middle samples for an even number of samples. The median is used in robust statistics as it is not affected by outliers. If less than 50% of the samples are outliers, then the median still works. With the Lebesgue-Stieltjes integral the median m is the value for which

$$\Pr(X \leq m) \ \geq \ \frac{1}{2} \ \text{ and } \ \Pr(X \geq m) \ \geq \ \frac{1}{2} \tag{3.4}$$

or

$$\int_{(-\infty,m]} \mathrm{d}F(x) \ \geq \ \frac{1}{2} \ \text{ and } \ \int_{[m,\infty)} \mathrm{d}F(x) \ \geq \ \frac{1}{2} \, . \tag{3.5}$$

For continuous probability density functions this is

$$\Pr(X \leq m) \ = \ \Pr(X \geq m) \ = \ \int_{-\infty}^{m} f(x) \, \mathrm{d}x \ = \ \frac{1}{2} \, . \tag{3.6}$$

For unimodal distributions, the median m and the mean $\bar{x}$ lie within $(3/5)^{1/2}$ standard deviations (see later) of each other:

$$\frac{|m \ - \ \bar{x}|}{\sigma} \ \leq \ (3/5)^{1/2} \ \approx \ 0.7746 \, . \tag{3.7}$$

For distributions with finite variance we have

$$\begin{aligned} |\mu \ - \ m| \ = \ |\mathrm{E}(X \ - \ m)| \ &\leq \ \mathrm{E}\left(|X \ - \ m|\right) \\ &\leq \ \mathrm{E}\left(|X \ - \ \mu|\right) \\ &\leq \sqrt{\mathrm{E}((X \ - \ \mu)^2)} \ = \ \sigma \, . \end{aligned} \tag{3.8}$$

The first and third inequality stem from Jensen's inequality applied to the absolute-value and the square function, each of which is convex. The second inequality comes from

$$m \ = \ \arg\min_{a} \mathrm{E}(|X \ - \ a|) \, . \tag{3.9}$$

This means that the ratio between median-mean distance and variance is smaller than 1 in contrast to the lower upper bound of 0.7746 for unimodal distributions for the sample mean.

The *mode* is the sample that appears most often in the data. The mode of a discrete probability distribution is the value x at which its probability mass function is maximal. This is the most typical sample and the sample that is most likely to be randomly chosen. The mode of a continuous probability distribution is the sample x at which its probability density function is maximal, that is, the global density peak.

$$\text{mode} \ = \ \arg\max_{x} \Pr(x) \quad \text{or} \quad \arg\max_{x} f(x) \, . \tag{3.10}$$

Table 3.1: Overview of mean, median, and mode.

Type	Description	Example	Result
Arithmetic mean	Sum of values of a data set divided by number of values: $\bar{x}= \frac{1}{n}\sum_{i=1}^{n} x_i$	(1+2+2+3+4+7+9) / 7	4
Median	Middle value separating the greater and lesser halves of a data set	1, 2, 2, 3, 4, 7, 9	3
Mode	Most frequent value in a data set	1, 2, 2, 3, 4, 7, 9	2

The median and the mode lie within $3^{1/2}$ standard deviations of each other:

$$\frac{|m \ - \ \mathrm{mode}|}{\sigma} \ \leq \ 3^{1/2} \ \approx \ 1.732 \ . \tag{3.11}$$

Tab. 3.1 gives an overview of the mean, median, and mode with examples. Fig. 3.1 shows a comparison of mean, median, and mode of two log-normal distributions with different skewness.

We have

- the mean minimizes the average squared deviation (L^2-norm),
- the median minimizes average absolute deviation (L^1-norm),
- the mid-range (0.5 times the range — see later) minimizes the maximum absolute deviation (L^∞-norm).

The median minimizes the average absolute deviation as we show in the following. The average absolute deviation is

$$\mathrm{E}(|x-a|) \ = \ \int_{-\infty}^{\infty} |x-a| \ p(x) \ dx \ = \ \int_{-\infty}^{a} (a-x) \ p(x) \ dx \ + \ \int_{a}^{\infty} (x-a) \ p(x) \ dx \ . \tag{3.12}$$

Setting the derivative with respect to a to zero gives

$$\frac{\partial \mathrm{E}(|x-a|)}{\partial a} \ = \ \int_{-\infty}^{a} p(x) \ dx \ - \ \int_{a}^{\infty} p(x) \ dx \ = \ 0 \ , \tag{3.13}$$

which leads to

$$\begin{aligned} 1 \ - \ 2 \int_{a}^{\infty} p(x) \ dx \ &= \ 0 \\ \int_{a}^{\infty} p(x) \ dx \ &= \ \frac{1}{2} \ . \end{aligned} \tag{3.14}$$

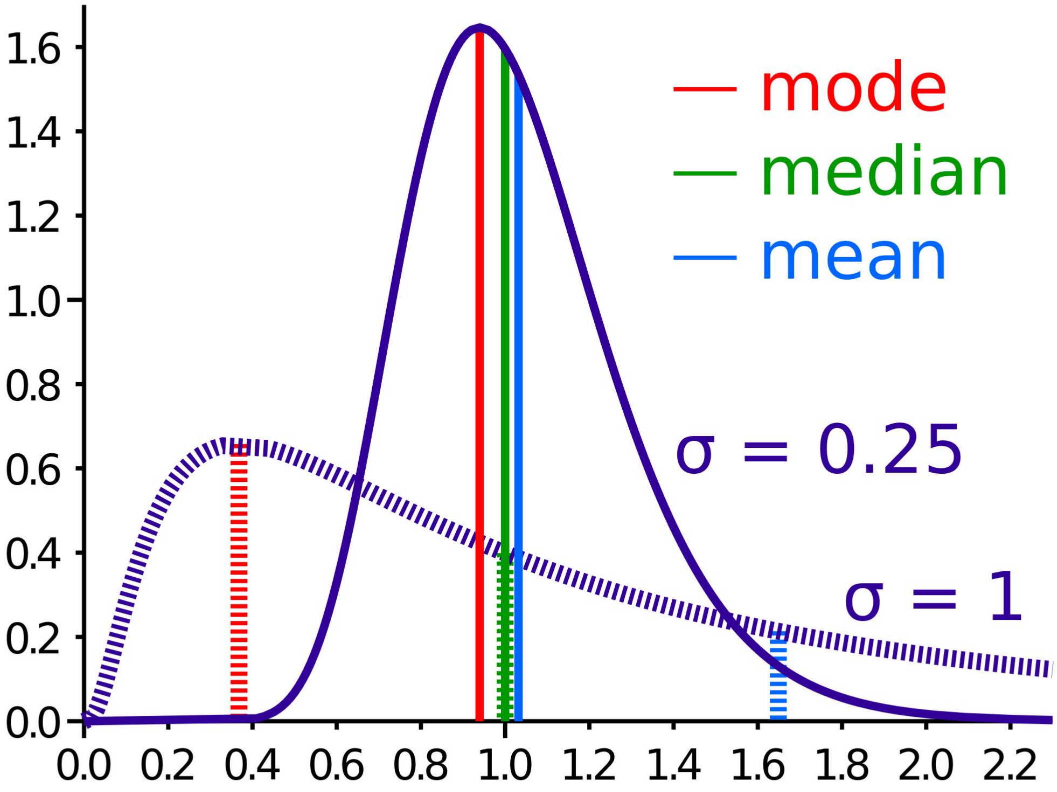

Figure 3.1: Comparison of mean, median, and mode of two log-normal distributions with different skewness.

This optimal value is the median $a = m$. The second derivative is larger than zero:

$$\frac{\partial^2 \mathrm{E}(|x - a|)}{\partial a^2} \; = \; p(a) \; - \; (-p(a)) \; = \; 2\, p(a) \; \geq \; 0 \; . \tag{3.15}$$

Therefore the median minimizes the average absolute deviation.

In Appendix A we show that for symmetric distributions the mean is equal to the median, which is equal to the symmetry point.

Furthermore, we show in Appendix A that both the mean and the median of samples from a Gaussian distribution should be estimated by the empirical mean. However for a Laplace distribution, both the mean and the median should be estimated by the empirical median.

3.1.2 Measuring the Variability

We now describe the spread of the data around the center. The *range* of a data set $\boldsymbol{x}$ is defined as largest observation minus smallest observation:

$$\text{range} \ = \ \max \boldsymbol{x} \ - \ \min \boldsymbol{x} \ . \tag{3.16}$$

The range take only the largest and the smallest observation into account, while all other observations are ignored.

The n *deviations from the sample mean* are the differences $(x_1 - \bar{x}), (x_2 - \bar{x}), \ldots, (x_n - \bar{x})$. The average deviation is zero because

$$\sum_{i=1}^{n} (x_i - \bar{x}) \ = \ \sum_{i=1}^{n} x_i \ - \ n\, \bar{x} \ = \ n\, \bar{x} \ - \ n\, \bar{x} \ = \ 0 \ . \tag{3.17}$$

The unbiased *sample variance* is denoted by s^2 and is defined as

$$s^2 \ = \ \frac{1}{n-1} \ \sum_{i=1}^{n} (x_i - \bar{x})^2 \ . \tag{3.18}$$

The data $\boldsymbol{x}$ contain $(n - 1)$ pieces of information ($(n - 1)$ degrees of freedom or df) on the deviations. One degree of freedom was used up by the empirical mean $\bar{x}$. If we know $\bar{x}$ and the $(n - 1)$ deviations $(x_i - \bar{x})$, $1 \leq i \leq (n - 1)$, then we can calculate $x_1, \ldots, x_{n-1}$ and $x_n = n\bar{x} - \sum_{i=1}^{n-1} x_i$, therefore also the deviation $(x_n - \bar{x})$. The biased *sample variance* is

$$\frac{1}{n} \ \sum_{i=1}^{n} (x_i - \bar{x})^2 \ . \tag{3.19}$$

The unbiased *sample standard deviation* (sd) is $s = \sqrt{s^2}$. The variance s^2 and the standard deviation s indicate the variability of the data. The larger the variance or the sd, the larger the data variability. The standard deviation is an estimate of how much a sample will deviate from the data center. sd is the size of a typical deviation from the mean.

The *population variance* σ^2 is

$$\sigma^2 \ = \ \sum_{x \in X} (x \ - \ \mu)^2 \ \Pr(x) \tag{3.20}$$

and continuous probability distributions

$$\sigma^2 \ = \ \int_X (x \ - \ \mu)^2 \ \Pr(x) \ dx \ . \tag{3.21}$$

The *population standard deviation* is σ.

In Appendix A we show that for a Gaussian distribution the biased variance (factor $1/n$) has a lower mean squared error than the unbiased variance (factor $1/(n - 1)$). The same holds for the Laplace distribution. This means that the biased sample variance is on average closer to the

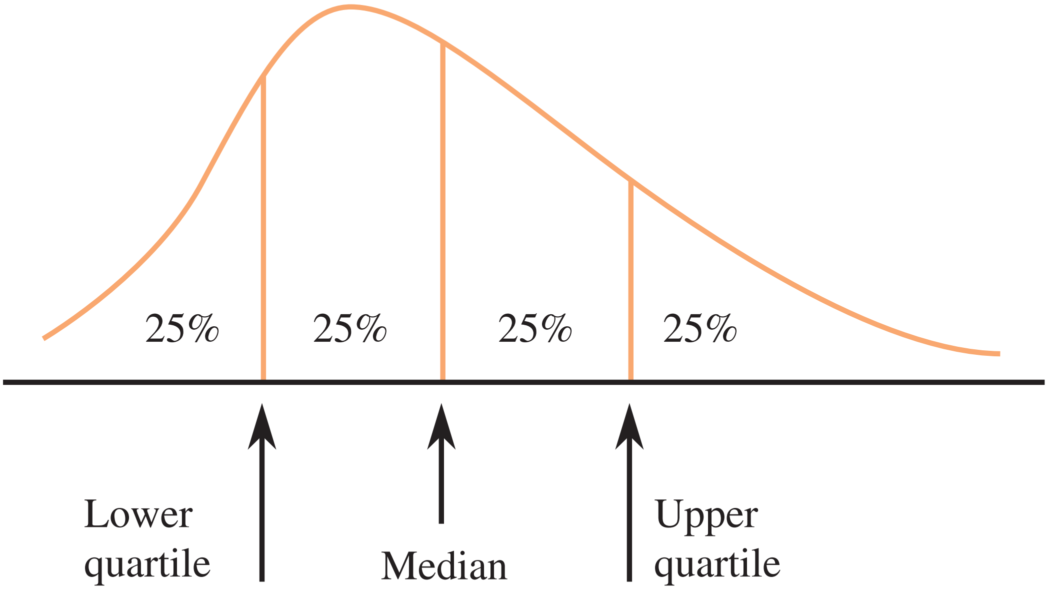

Figure 3.2: The quartiles of a distribution are depicted. Figure from R. Peck and Devore [2009].

population variance than the unbiased sample variance even if the biased sample variance makes a systematic error by underestimating the population variance.

The median was introduced as a more robust estimate of the data center than the mean, that is, it is not as much affected by outliers as the mean. We now introduce a robust estimate of the variability of the data.

A robust measure of variability is the interquartile range. It is based on quantities called *quartiles*. The *lower quartile* separates the bottom 25% of the data set from the upper 75% (the median of the lower half), and the *upper quartile* separates the top 25% from the bottom 75% (the median of the upper half). The middle quartile is the robust measure of the data center, the median. Fig. 3.2 shows the quartiles of a distribution. The *interquartile range* is the upper quartile minus the lower quartile. That is the range which contains 50% of the data around the center.

3.1.3 Summary Statistics

We revisit the iris data set from Section 2.2 and consider the feature “sepal length”. We compute center and variability of “sepal length” in the software environment R :

```
x <- iris[,"Sepal.Length"]
mean(x)
 [1] 5.843333
median(x)
 [1] 5.8
var(x)
 [1] 0.6856935
sd(x)
```

```
 [1] 0.8280661
sqrt(var(x))
 [1] 0.8280661
quantile(x)
  0%  25%  50%  75% 100%
 4.3  5.1  5.8  6.4  7.9
summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.300   5.100   5.800   5.843   6.400   7.900
```

We now look at the iris species separately:

```
iS <- iris$Species == "setosa"
iV <- iris$Species == "versicolor"
iG <- iris$Species == "virginica"
xS <- x[iS]
xV <- x[iV]
xG <- x[iG]
summary(xS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.300   4.800   5.000   5.006   5.200   5.800
summary(xV)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.900   5.600   5.900   5.936   6.300   7.000
summary(xG)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.900   6.225   6.500   6.588   6.900   7.900
```

We see that the centers of versicolor are larger than the centers of setosa, and that the centers of virginica are larger than the centers of versicolor. The same figure can be seen for the upper quartile and for the maximum.

Now we do the same for petal length.

```
x1 <- iris[,"Petal.Length"]
mean(x1)
 [1] 3.758
median(x1)
 [1] 4.35
var(x1)
 [1] 3.116278
sd(x1)
 [1] 1.765298
quantile(x1)
   0%  25%  50%  75% 100%
 1.00 1.60 4.35 5.10 6.90
summary(x1)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.600   4.350   3.758   5.100   6.900
x1S <- x1[iS]
x1V <- x1[iV]
x1G <- x1[iG]
summary(x1S)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.400   1.500   1.462   1.575   1.900
summary(x1V)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.00    4.00    4.35    4.26    4.60    5.10
summary(x1G)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.500   5.100   5.550   5.552   5.875   6.900
```

We see a similar figure as with the sepal length. Interestingly, the maximum of setosa is below the minimum of virginica and versicolor. This means that we can identify the species setosa in the set of three species if we only measure the petal length.

The *z-score* or the *standardized data* is

$$z = \frac{x - \bar{x}}{s} . \tag{3.22}$$

The z score measures for each data point x how many standard deviations it is away from the mean. z is positive if x is larger than the mean and negative otherwise.

The quartiles can be generalized to the r-th percentile. The r-th percentile of a data set is the value for which r percent of the observations are smaller or equal to this value.

Reliability or variance of the descriptive values. The numerical measures do not include a reliability value or a variance estimation. In particular for few observations the numerical measures may be misleading because their variance is large. If these values would be assessed a second time then the outcome may be quite different. For example, the mean booting time of my new notebook is 10 minutes. I tried it only 3 times (it is new). The first boot took 30 minutes as all hardware and software programs were initialized and confirmed. The other two boots took only few seconds. This shows that outliers may have large effect on the numerical measures if only few examples are available. Therefore, the number of observations and the variation of the single measurements should be considered when reporting summary statistics.

3.1.4 Boxplots

The summary statistics can be visualized by boxplots. Boxplots are box-and-whisker plots of the given observations and visualize different values simultaneously. The default statistics, which are shown for the boxplot in R , are:

- the median as horizontal bar
- the box ranging from the lower to the upper quartile

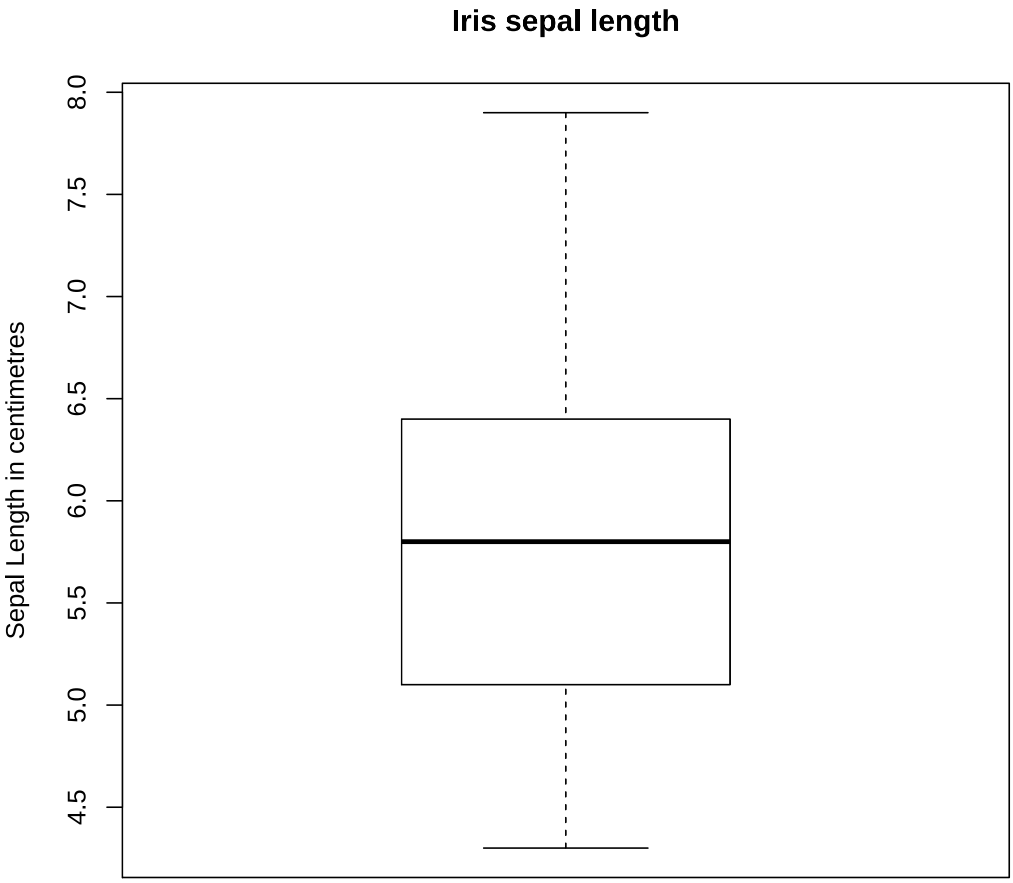

Figure 3.3: Boxplot of the feature sepal length of the iris data set.

- whiskers range from the maximal to the minimal observations without the outliers
- outliers as points; outliers are observations that have larger deviation than `fact` times the interquartile range from the upper or lower quartile. In R default is `fact=1.5`.

Fig. 3.3 shows a boxplot of the sepal length of the iris data set. This boxplot was created by the R command:

```
boxplot(x,main="Iris sepal length",ylab="Sepal Length in centimetres")
```

Fig. 3.4 shows boxplots of the sepal length of the iris data set per species. These boxplots were created by the R command:

```
boxplot(x ~ unclass(iris$Species),main="Iris sepal length",
```

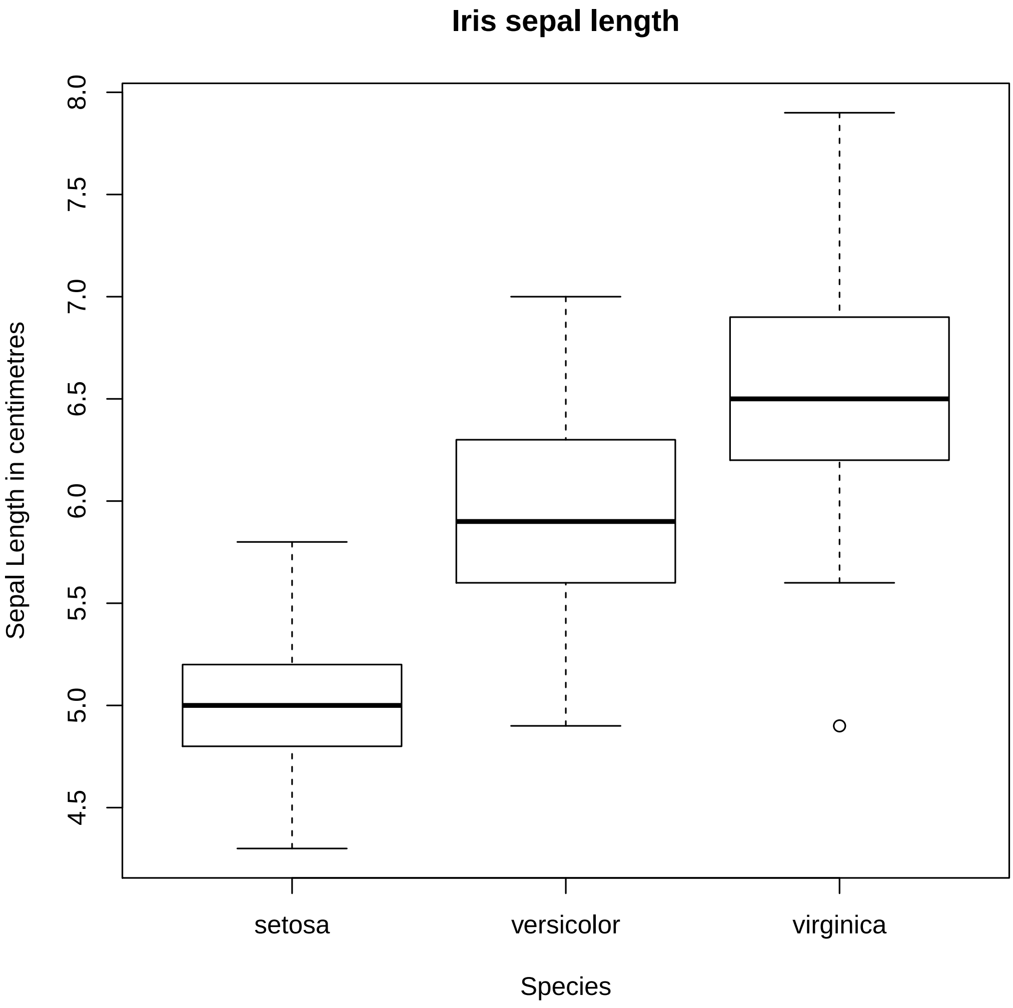

Figure 3.4: Boxplots of sepal length per species of the iris data set.

```
+ names=c("setosa","versicolor","virginica"),
+ xlab="Species",ylab="Sepal Length in centimetres")
```

The boxplots show that setosa can be distinguished from the other two species by the sepal length in most cases. The sepal length of virginica is on average and in most cases larger than the sepal length of versicolor.

We move on to the the next feature, the petal length. Fig. 3.5 shows the boxplot of the petal length of the iris data set. This boxplot was created by the R command:

```
boxplot(x1,main="Iris petal length",ylab="Petal Length in centimetres")
```

Fig. 3.6 shows boxplots of the petal length of the iris data set per species. These boxplots were created by the R command:

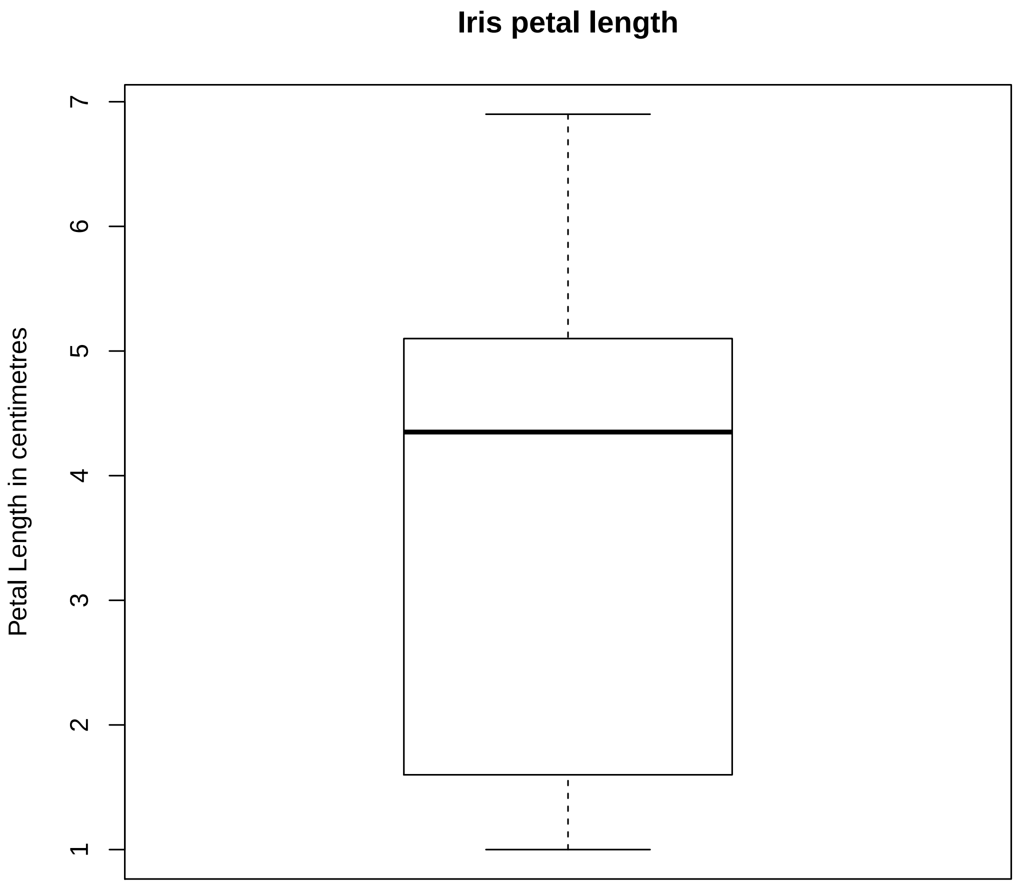

Figure 3.5: Boxplot of the feature petal length of the iris data set.

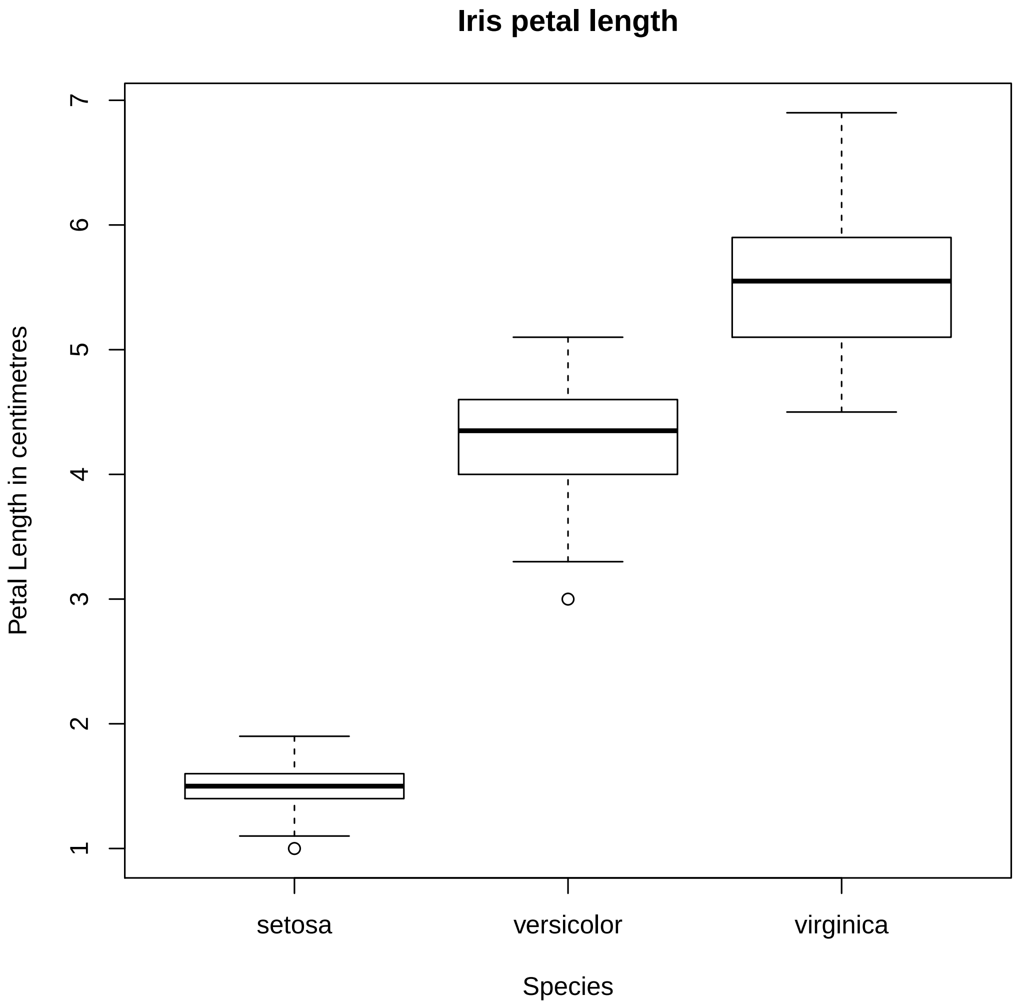

Figure 3.6: Boxplots of petal length per species of the iris data set.

```
boxplot(x1 ~ unclass(iris$Species),main="Iris petal length",
+ names=c("setosa","versicolor","virginica"),xlab="Species",
+ ylab="Petal Length in centimetres")
```

The boxplots show that setosa can be distinguished from the other two species by the petal length in all cases. Setosa has clearly shorter petal lengths than the other two species. The petal length of virginica allows a better discrimination to versicolor than the sepal length. In summary, petal lengths can much better discriminate the species than sepal lengths.

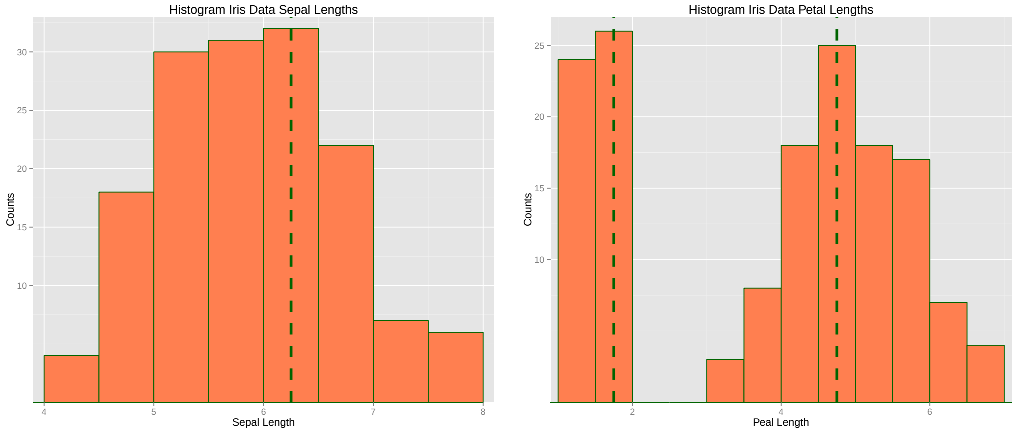

Figure 3.7: Histograms of sepal and petal lengths. Green vertical lines mark peaks in the histograms.

3.1.5 Histograms

A histogram is a graphical representation of the distribution of data. Tabulated frequencies are shown as adjacent rectangles which erect over discrete intervals (bins). The area of the rectangle is equal to the frequency of the observations in the interval. For equidistant bins the area of the rectangles is proportional to the height of the rectangles, therefore, to the frequency of the observations in the according bin.

Histograms help to assess the extent of spread or variation of the observations, the general shape allows to determine location of peaks, further low density regions or outliers may be found. Thus, histograms may give an informative overview of the observations. Fig. 3.7 shows the histograms of sepal and petal lengths. For petal length a gap is visible between short and long petals. We already know that the species setosa has shorter petals then the other two species, therefore, we known the reason for this gap. In R the command `hist()` provides simple histogram plots. We plotted the histograms with the R package `ggplot2`. Appendix B lists the code which produced the histograms in Fig. 3.7.

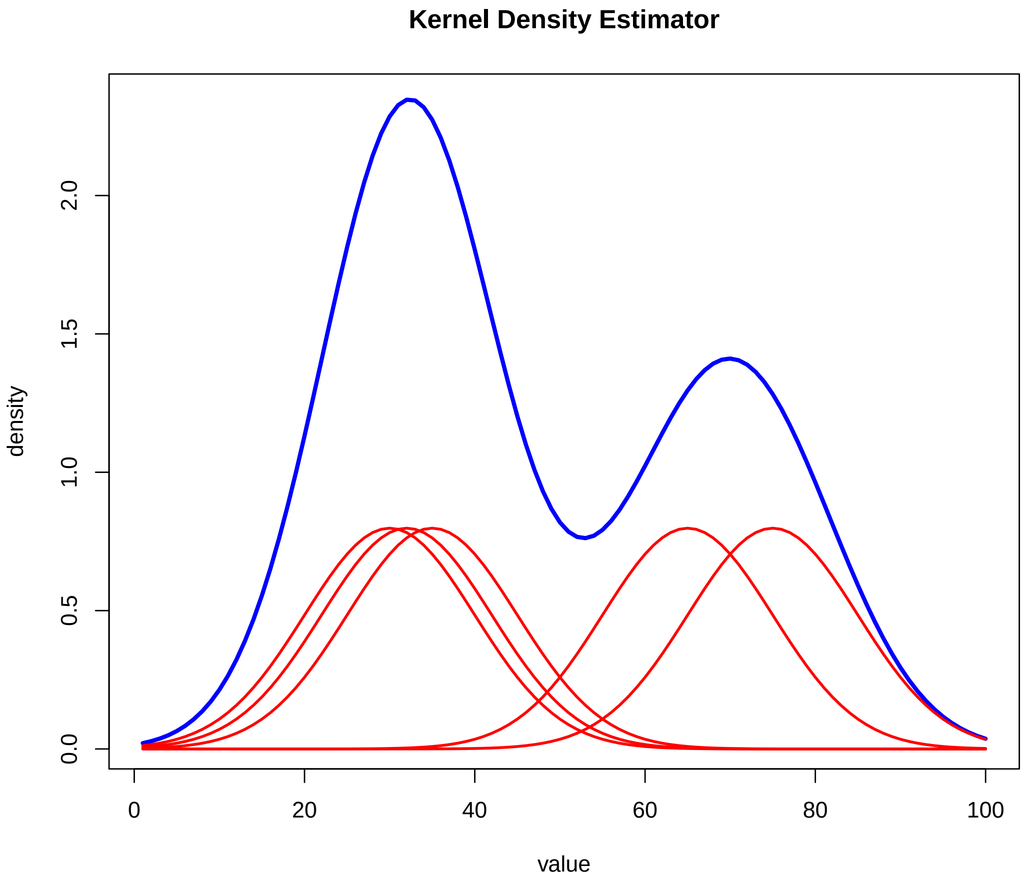

Figure 3.8: Kernel density estimator. The blue density is approximated by the average of the red kernel densities.

3.1.6 Density Plots

Smoothing the surface of a histogram leads to a probability density function. In general, probability density functions are obtained by kernel density estimation (KDE) which is a non-parametric (except for the bandwidth) method also called Parzen-Rosenblatt window method.

A *kernel density estimator* $\hat{f}_h$ has the following form:

$$\hat{f}_h(x) \ = \ \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i) \ = \ \frac{1}{nh} \ \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) , \tag{3.23}$$

where $K(.)$ is the kernel (symmetric, positive function that integrates to one) and $h > 0$ is the *bandwidth*. Fig. 3.8 depicts a kernel density estimator where the blue density is approximated by the average of the red kernel densities. This means that the red kernels are scaled by $\frac{1}{n}$, where n is the number of data points (R code in appendix). The locations x_i of the kernels are 30, 32, 35, 65, and 75, while the bandwidth is $h = 10$.

The most tricky part of KDE is the bandwidth selection. If the bandwidth is too small, then the density has many peaks and is wiggly. If the bandwidth is too large, then the peaks vanish and details are no longer visible. A practical estimation of the bandwidth for Gaussian kernels is the rule-of-thumb (Silverman's rule):

$$h \;=\; \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}} \;\approx\; 1.06\,\hat{\sigma}\,n^{-1/5}\;, \tag{3.24}$$

where $\hat{\sigma}$ is the standard deviation of the observations. The closer the true density is to a Gaussian, the better the bandwidth estimation.

We demonstrate the density estimation on the iris data set from Section 2.2. Fig. 3.9 shows the densities of sepal lengths per species of the iris data set. The estimate uses Gaussian kernels with rule-of-thumb bandwidth adjustment. The densities show that the species differ in their sepal length as the peaks of the densities and their location show. Setosa has the least overlap with the other species. However, versicolor and virginica have a considerable overlap of density mass even if their peaks are clearly separated.

Fig. 3.10 shows the densities of petal lengths per species of the iris data set. Setosa has no overlap with the other species and the density is very narrow (small variance). Versicolor and virginica have less overlap than with sepal length and can be separated quite well.

Appendix B lists the code which produced the densities in Figs. 3.9 and 3.10.

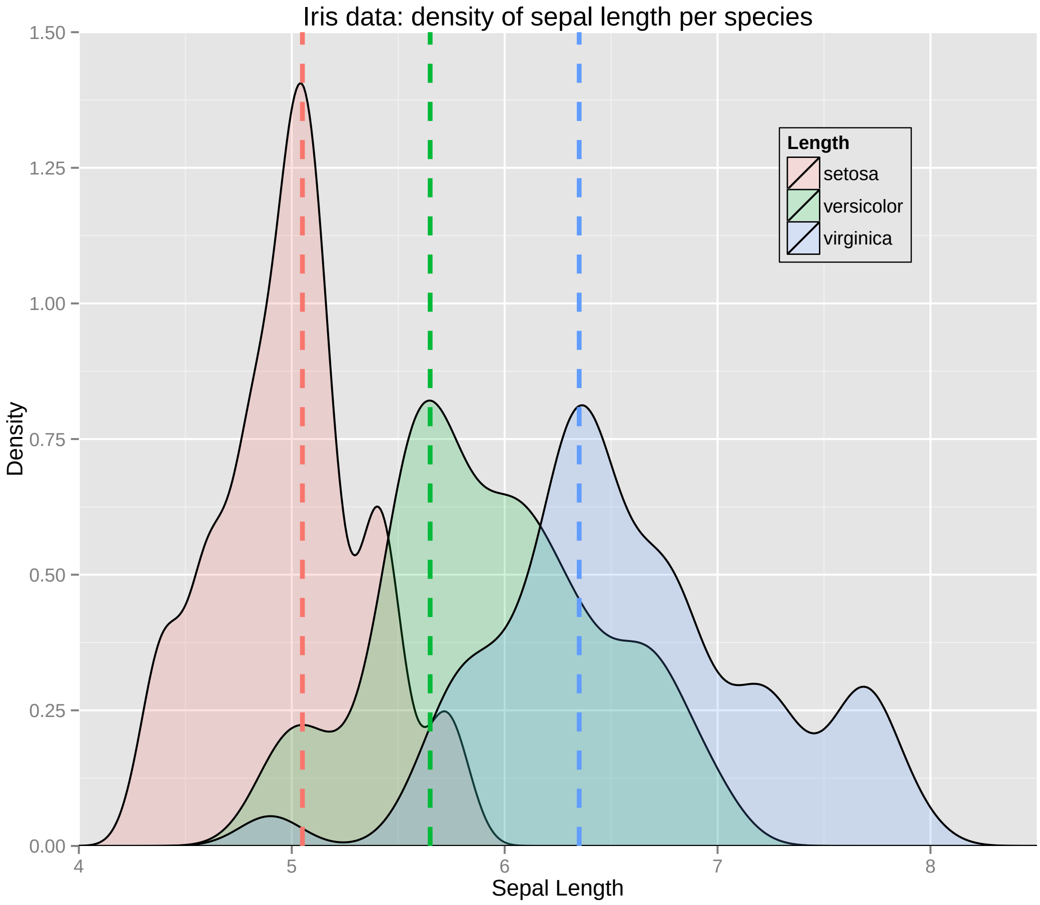

Figure 3.9: Densities of sepal lengths per species of the iris data set. Dashed lines mark peaks of the densities.

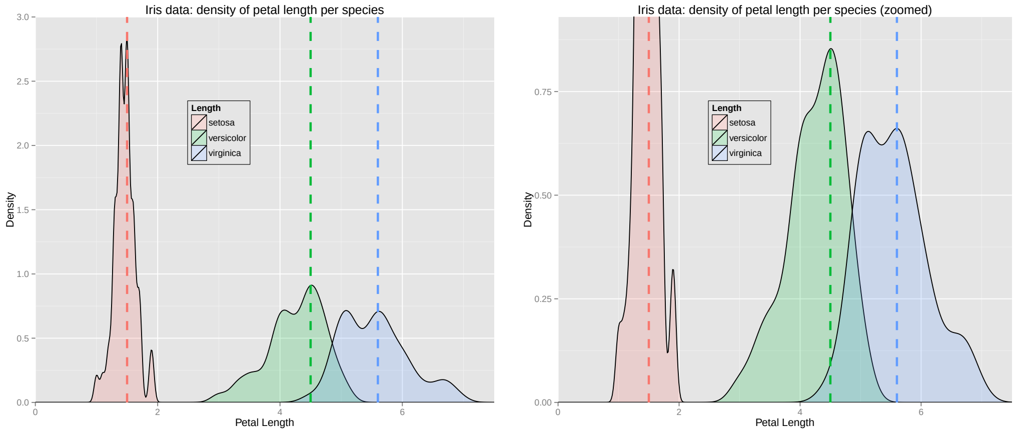

Figure 3.10: LEFT: Densities of petal lengths per species of the iris data set. RIGHT: The same densities as on the left hand side but zoomed in. Dashed lines mark peaks of the densities.

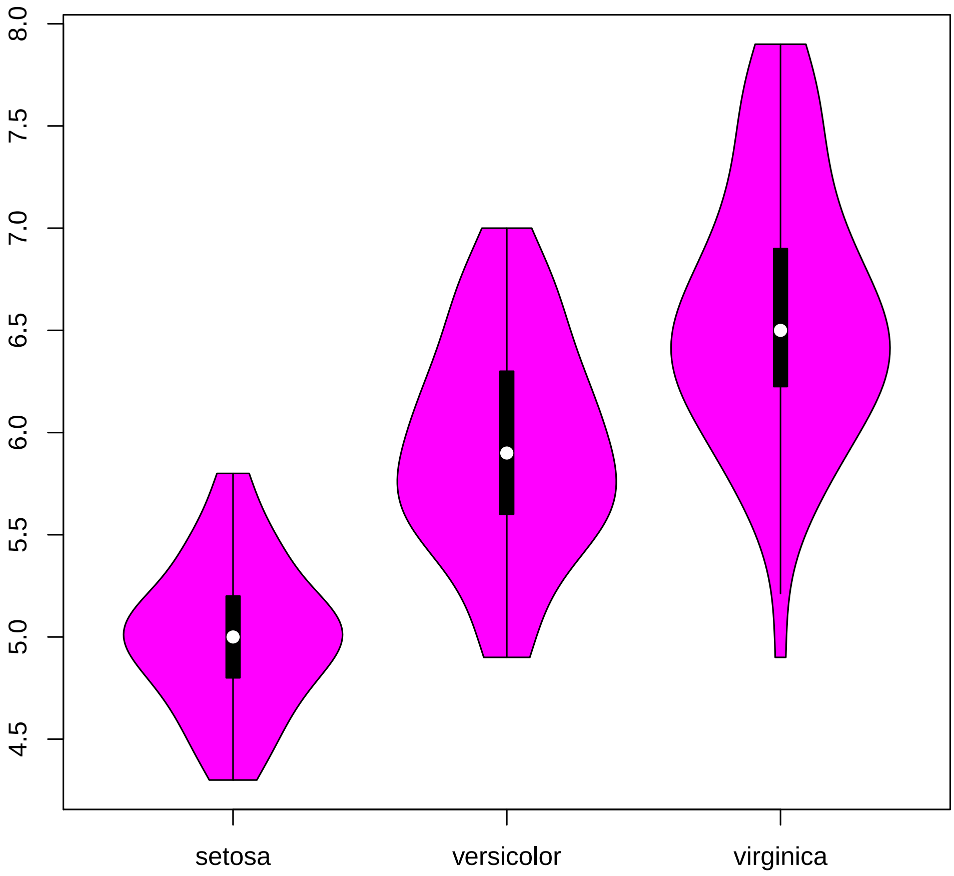

Figure 3.11: Violin plots of sepal lengths per species of the iris data set.

3.1.7 Violin Plots

A violin plot is a combination of a boxplot and a density estimation. The violin plot enhances a box plot by a rotated kernel density plot at each side. Fig. 3.11 shows violin plots of the sepal lengths of the iris data set and Fig. 3.12 shows violin plot of the petal lengths. The R code for producing Fig. 3.11 is

```
library(vioplot)
vioplot(x ~ unclass(iris$Species),main="Iris sepal length",
+ names=c("setosa","versicolor","virginica"),
+ xlab="Species",ylab="Sepal Length in centimetres")
```

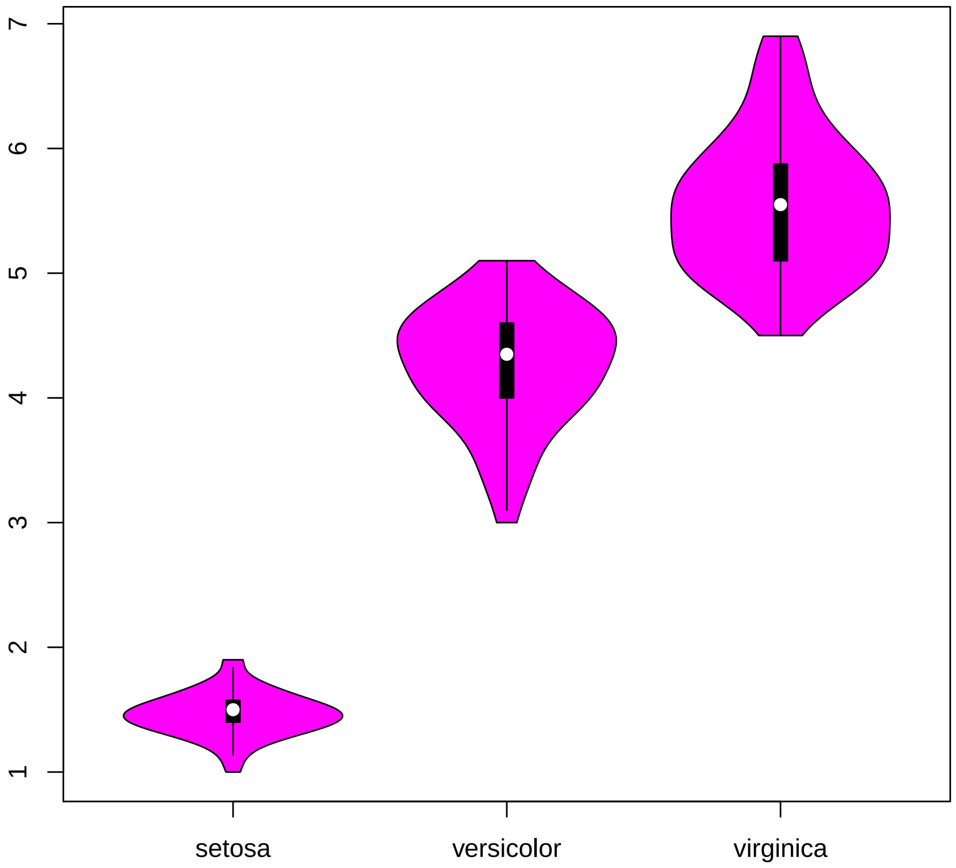

Figure 3.12: Violin plots of petal lengths per species of the iris data set.

3.2 Summarizing Bivariate Data

We next consider how to describe *bivariate data*, that is, data which has two scalar variables. Bivariate data is a set of pairs of data points in most cases called x and y. The data is given as $\{(y_1, x_1), (y_2, x_2), \ldots, (y_n, x_n)\}$. For some application x and y can be distinguished: y is the *response* or *dependent variable* and x is the *explanatory variable*, *independent variable*, *regressor*, or *feature*. In this situations the assumption is that the response y is influenced or even caused by x. Often causality is introduced because the assumption is that y is (partly) caused by x. Without prior knowledge or certain assumptions, statistical or machine learning methods cannot determine causality but only dependencies, relations, or structures in the data.

3.2.1 Scatter Plot

A scatter plot shows each observation as a point, where the x-coordinate is the value of the first variable and the y-coordinate is the value of the second variable.

The R command

```
plot(anscombe[,1:2],main = "Anscombe Data",pch = 21,bg = c("red"),
+ cex=2,xlab="feature 1",ylab="feature 2")
```

produced the scatter plot in Fig. 3.13. Since the values of feature 1 and feature 2 are identical, the points are on the 45° diagonal. In cases that feature 1 and feature 2 are linearly dependent, the points are on an line (see Fig. 3.14). Fig. 3.15 shows a scatter plot for linearly dependent features but with noise. The points deviate from the line due to the noise. Fig. 3.16 shows a scatter plot for linearly dependent features (upper right matrix) and independent features (lower left matrix). Fig. 3.17 shows a scatter plot for non-linearly dependent features. The points lie on a one-dimensional curve. Fig. 3.18 shows a matrix of scatter plots for the Anscombe data produced by the R command:

```
pairs(anscombe[,c(1,2,5,6,7,8)], main = "Anscombe Data",
+ pch = 21, bg = c("red"))
```

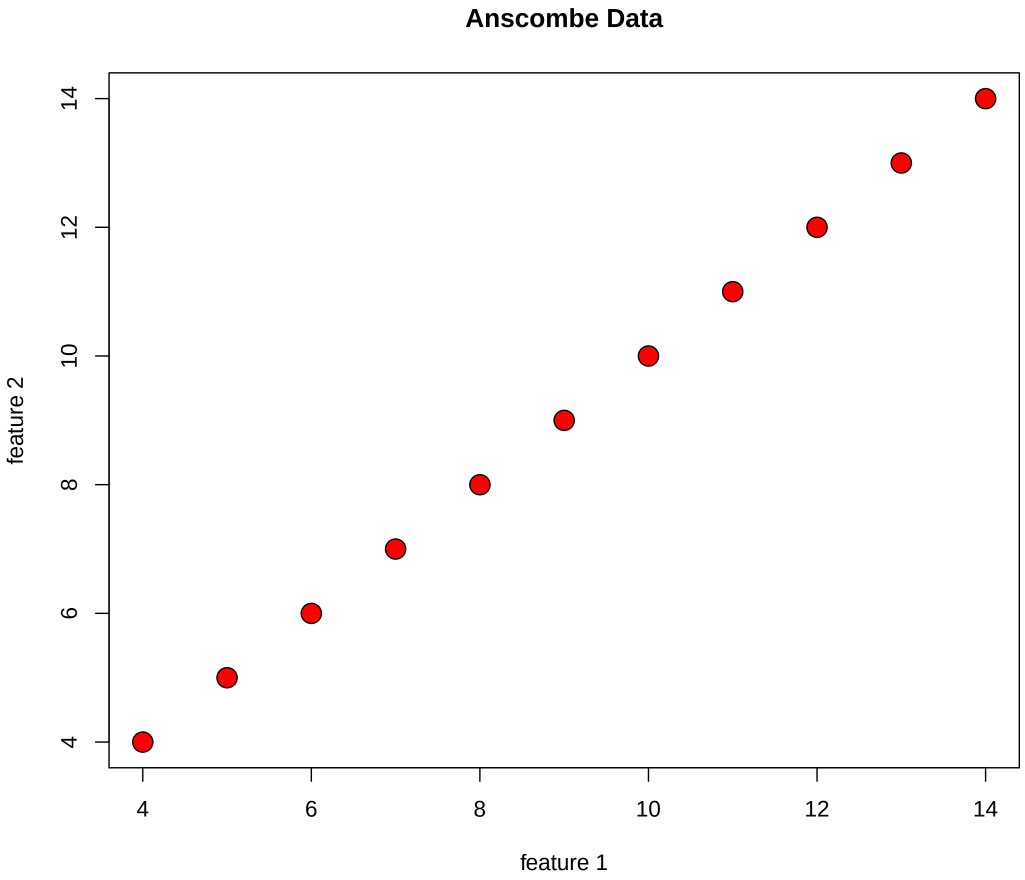

Figure 3.13: Scatter plot of data with two features.

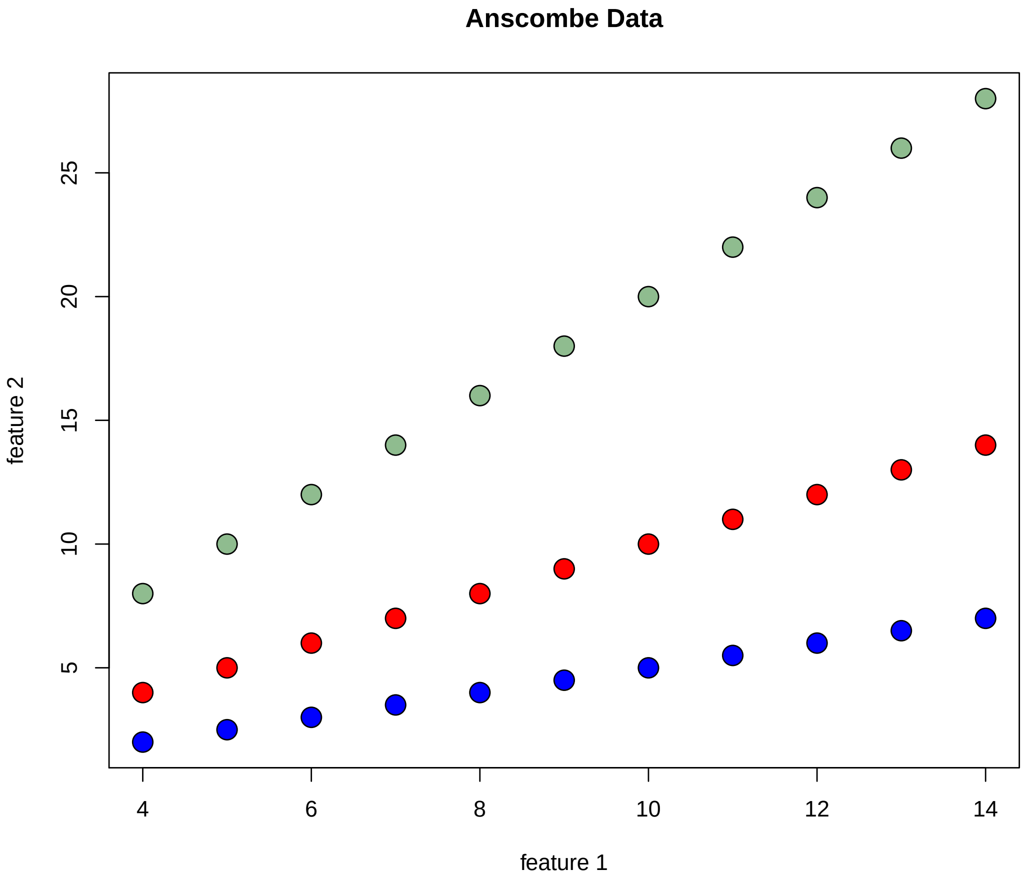

Figure 3.14: Scatter plots of linearly dependent features.

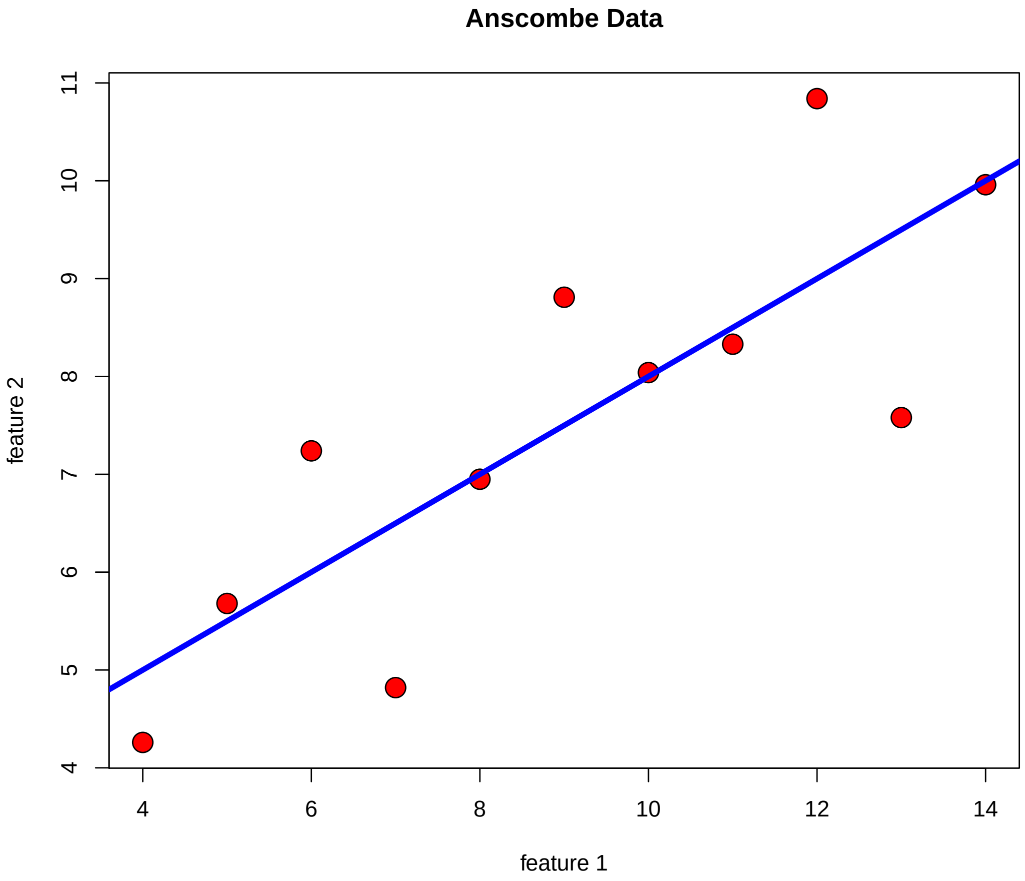

Figure 3.15: Scatter plot of noisy linearly dependent features. The blue line is the extracted linear dependency from which points deviate because of the noise.

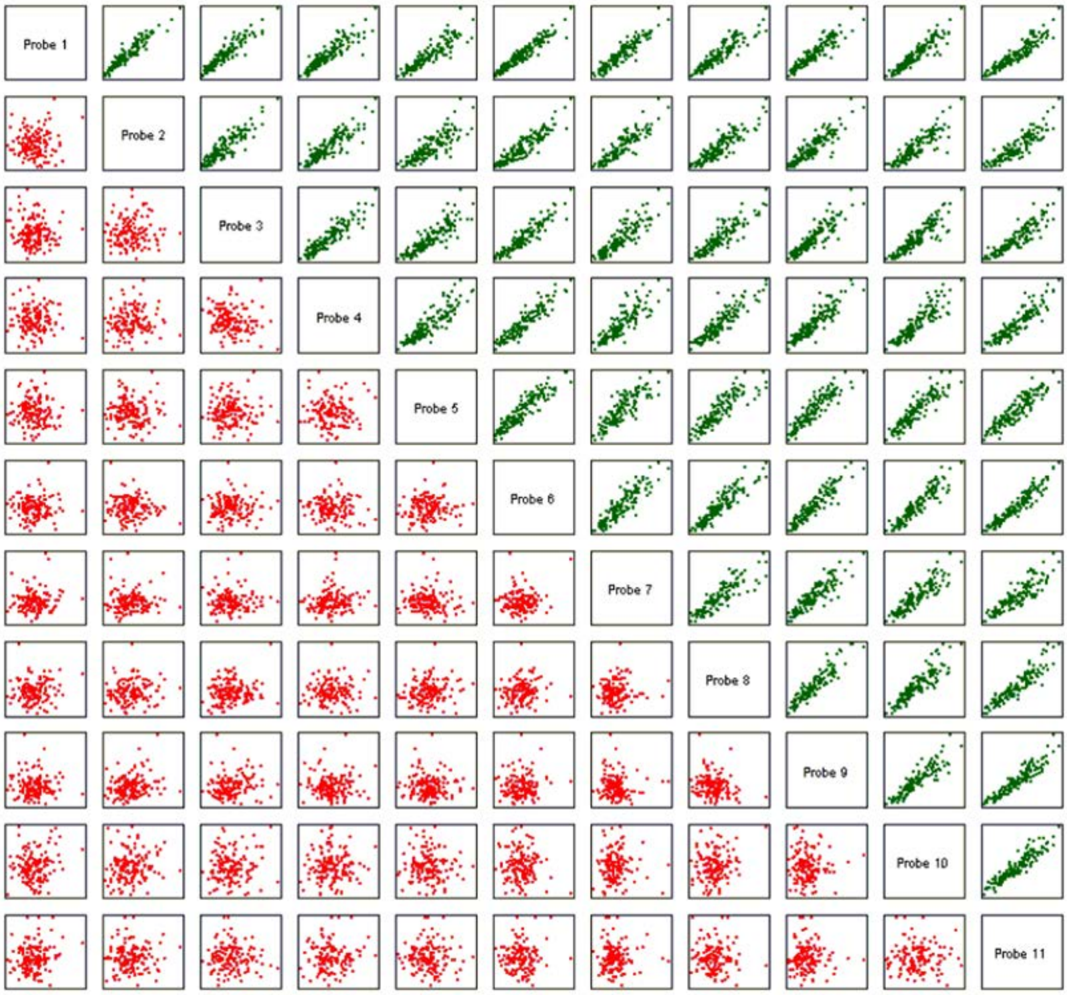

Figure 3.16: Scatter plots of linearly dependent and independent features. Upper right matrix shows scatter plots of linearly dependent features while the lower left matrix shows scatter plots of independent features.

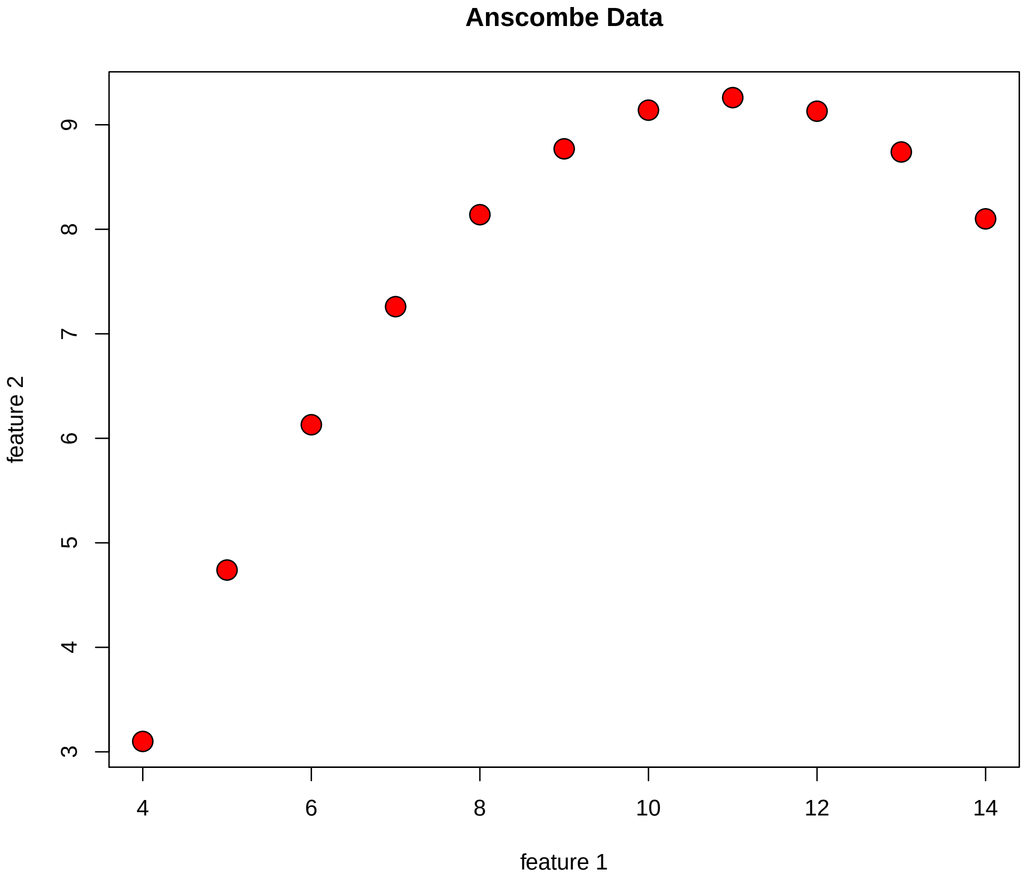

Figure 3.17: Scatter plot of non-linearly dependent features. The points lay on a one-dimensional curve.

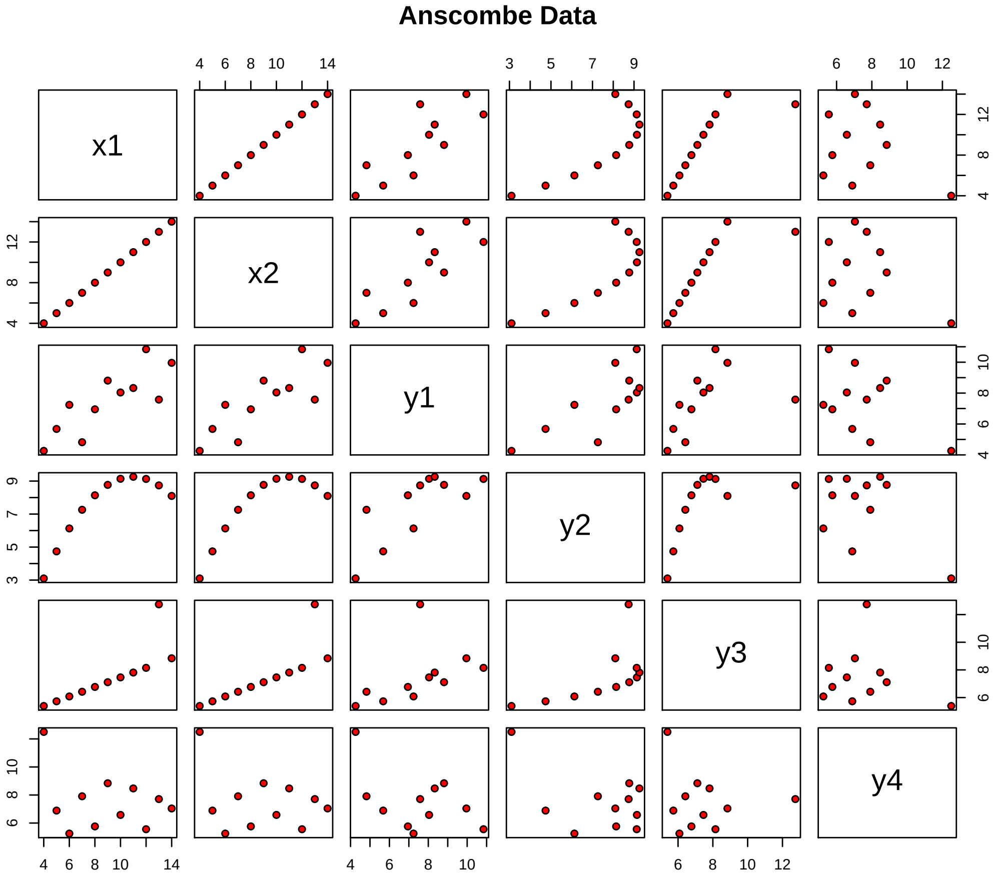

Figure 3.18: Matrix of scatter plots for the Anscombe data.

3.2.2 Correlation

We saw that two variables are linearly dependent if their points are on a line (see Fig. 3.13). Variables are to some degree linearly dependent if their points lie around a line (see Fig. 3.15). The more the points are on the line, the higher is the linear dependence.

For the bivariate data $(y_1, x_1), (y_2, x_2), \ldots, (y_n, x_n)$ *Pearson's sample correlation coefficient* r is a measure of the linear correlation (dependence) between the two variables x and y:

$$r \ = \ \frac{\sum_{i=1}^{n}(x_i \ - \ \bar{x})(y_i \ - \ \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i \ - \ \bar{x})^2} \, \sqrt{\sum_{i=1}^{n}(y_i \ - \ \bar{y})^2}} \tag{3.25}$$

or equivalently with z-scores

$$r \ = \ \frac{1}{n-1} \sum_{i=1}^{n} (z_x)_i (z_y)_i \ . \tag{3.26}$$

Pearson's population correlation coefficient is denoted by ρ. For a perfect linear dependency $x_i \ = \ a y_i$ the correlation coefficient is $r \ = \ 1$ or $r \ = \ -1$. Since $\bar{x} \ = \ a\bar{y}$, the numerator has the factor a while the denominator has the factor $|a|$, therefore only the sign of a is kept as a factor.

The correlation coefficient of the variables in Fig. 3.15 is $r = 0.82$ obtained by the R code:

```
cor(anscombe[,c(1,5)])
          x1        y1
 x1 1.0000000 0.8164205
 y1 0.8164205 1.0000000
```

Using z-scores that is

```
1/(length(anscombe[, 1])-1)*
+ crossprod(scale(anscombe[,1]),scale(anscombe[, 5]))
          [,1]
 [1,] 0.8164205
```

In the Anscombe data set, the correlations between x1 and y1, x1 and y2, as well as x1 and y3 are almost the same. However the correlations between y1, y2, and y3 are considerably lower:

```
cor(anscombe[,c(1,5,6,7)])
          x1        y1        y2        y3
 x1 1.0000000 0.8164205 0.8162365 0.8162867
 y1 0.8164205 1.0000000 0.7500054 0.4687167
 y2 0.8162365 0.7500054 1.0000000 0.5879193
 y3 0.8162867 0.4687167 0.5879193 1.0000000
```

The correlation measures the extent of association, but association does not imply causality. Two correlated variables may be related to a third variable. John Paulos gave the following example in ABCNews.com:

> "Consumption of hot chocolate is correlated with low crime rate, but both are responses to cold weather."

3.2.3 Test for Correlation

For a bivariate normal population, the test of independence is a test for a correlation coefficient of $\rho = 0$. The test is a t-test with the null hypothesis H_0 that $\rho = 0$. The test is only valid if both variables are drawn from a normal distribution.

The test statistic is

$$t = \frac{r}{\sqrt{\frac{1-r^2}{n-2}}} . \tag{3.27}$$

The degree of freedom is $\mathrm{df} = n - 2$.

The Student t-distribution with df degrees of freedom has the density

$$f(x) = \frac{\Gamma((\mathrm{df}+1)/2)}{\sqrt{\mathrm{df}\pi}\Gamma(\mathrm{df}/2)} \left(1 + \frac{x^2}{\mathrm{df}}\right)^{-(\mathrm{df}+1)/2} . \tag{3.28}$$

In R the p-value can be computed by the command:

```
1-pt(t,df=n-2)
```

`pt()` is the probability function of the Student t-distribution.

The correlation between x1 and y1 of the Anscombe data set is $r = 0.8164205$ which gives a p-value of:

```
r=0.8164205
t=r/(sqrt((1-r^2)/9))
t
[1] 4.241455
1-pt(t,9)
[1] 0.001084815
```

That means the p-value is smaller than 0.0011.

For the correlation between y1 and y3 of the Anscombe data set is $r = 0.4687167$ which gives for the p-value:

```
r=0.4687167
t=r/(sqrt((1-r^2)/9))
t
[1] 1.591841
1-pt(t,9)
[1] 0.07294216
```

That means the correlation is not significant for a level of 0.05.

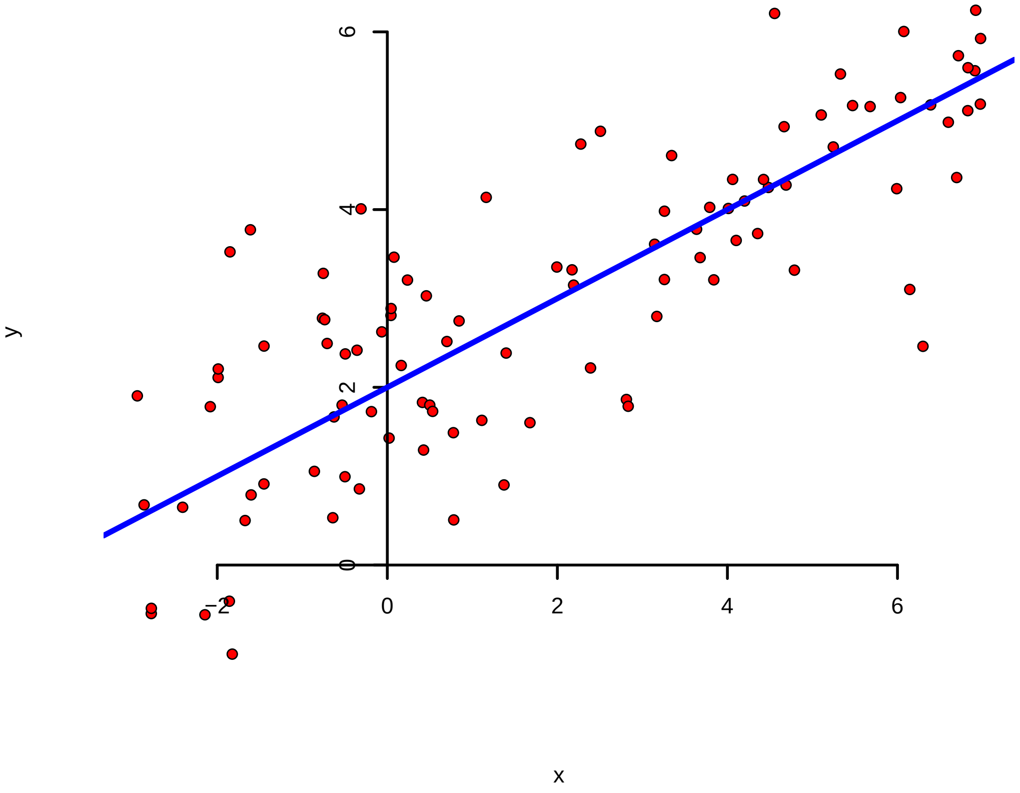

Figure 3.19: Linear regression of bivariate data. The blue line is the fitted regression line with intercept $a = 2$ and slope $b = 0.5$.

3.2.4 Linear Regression

The goal is to fit a line to bivariate data as shown in Fig. 3.15 in order to extract information about the relation of the two variables from the fitted line. In regression one variable is used to predict or to estimate the second variable.

The functional relationship between the variables x and y for linear regression is

$$y \ = \ a \ + \ b \, x \ , \tag{3.29}$$

where a is the *intercept* and b is the *slope*. Fig. 3.19 shows a regression curve with $a = 2$ and $b = 0.5$. The value a is obtained by $x = 0$, therefore the line intersects the y-axis at a value of $y = 2 = a$. $b = 0.5$ means that an increase of x of one unit leads to an increase of y of half a unit.

We have to define a *goodness of fit* criterion or *objective* which is a quality criterion for fitting. These criteria is important to find the line which fits the data best. The most widely used objective is the *sum of the squared deviations* between the y values and the regression line. The *least squares* *objective* is

$$\sum_{i=1}^{n} \Big(y_i \ - \ (\tilde{a} \ + \ \tilde{b} \, x_i) \Big)^2 \tag{3.30}$$

for some candidate intercept $\tilde{a}$ and candidate slope $\tilde{b}$.

The line which minimizes this objective (called "least squares") is the *least squares line*. The values $\hat{a}$ and $\hat{b}$ which minimize the objective, the least squares criterion, are

$$\begin{aligned} \hat{b} \ &= \ \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \ = \ \frac{\sum_{i=1}^{n} x_i \, y_i \ - \ \frac{1}{n} \sum_{i=1}^{n} x_i \sum_{j=1}^{n} y_j}{\sum_{i=1}^{n}(x_i^2) - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2} \\ &= \ \frac{\overline{xy} \ - \ \bar{x} \, \bar{y}}{\overline{x^2} \ - \ \bar{x}^2} \ = \ \frac{\mathrm{Cov}(x,y)}{\mathrm{Var}(x)} \ = \ r_{xy} \, \frac{s_y}{s_x} \end{aligned} \tag{3.31}$$

and

$$\hat{a} \ = \ \bar{y} \ - \ \hat{b} \, \bar{x} \ . \tag{3.32}$$

Here r_{xy} is the correlation coefficient between x and y, s_x is the standard deviation of x, and s_y the standard deviation of y.

From

$$y \ = \ a \ + \ b \, x \tag{3.33}$$

follows that

$$x \ = \ \frac{1}{b} \, (y \ - \ a) \ = \ -\frac{a}{b} \ + \ \frac{1}{b} \, y \ . \tag{3.34}$$

This relation does not hold for the estimated variables.

If the variables x and y are interchanged, then, in general, the estimated functional dependency changes. Least squares on y as desired function output is different from least squares on x as desired function output. The objective is different. If we denote the estimated slope with y as output by b_y and the estimated slope with x as output by b_x, then we have: $\hat{b}_y \ = \ r_{xy} \ s_y / s_x$ and $\hat{b}_x \ = \ r_{xy} \ s_x / s_y$. We see $\hat{b}_y \ \neq \ 1/\hat{b}_x$, because $r_{xy} \ \neq \ 1/r_{xy}$, except for $r_{xy} \ = \ 1$. For example, a horizontal line with constant y-values around which observations are located, cannot be inverted to a function of x with y as variable.

The estimated regression line is

$$y \ = \ \hat{a} \ + \ \hat{b} \, x \ , \tag{3.35}$$

which can be reformulated as

$$\frac{y \ - \ \bar{y}}{s_y} \ = \ r_{xy} \, \frac{x \ - \ \bar{x}}{s_x} \tag{3.36}$$

or with z-scores

$$z_y \ = \ r_{xy} \, z_x \ . \tag{3.37}$$

The correlation coefficient r_{xy} is the slope of the regression line of the standardized data points. Scaling r_{xy} gives the least squares slope $\hat{b}$. For the standardized data, the intercept vanishes because all data are centralized around the origin.

If the error terms

$$\hat{\varepsilon}_i \;=\; y_i \;-\; \hat{a} \;-\; \hat{b}\, x_i \tag{3.38}$$

are normally distributed, then $\hat{b}$ is normally distributed with mean b and variance $\sigma^2 / \sum (x_i - \bar{x})^2$, where σ^2 is the variance of the error terms.

Furthermore, the sum of squared error terms is distributed proportionally to χ^2 with $(n-2)$ degrees of freedom, and independently from $\hat{b}$. Thus the t-statistic

$$t \;=\; \frac{\hat{b} \;-\; b}{s_{\hat{b}}} \;\sim\; t_{n-2} \;, \tag{3.39}$$

has a Student's t-distribution with $(n-2)$ degrees of freedom. We used

$$s_{\hat{b}} \;=\; \sqrt{\frac{\frac{1}{n-2} \sum_{i=1}^{n} \hat{\varepsilon}_i^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}} \;. \tag{3.40}$$

This t-statistic allows constructing confidence intervals for both a and b. Further, we can construct a confidence interval for r_{xy}, which allows to express how confident we are that the two variables are linearly related.

The R^2 statistic is the *fraction of variance explained by the model* also called the *coefficient of determination*:

$$R^2 \;=\; 1 \;-\; \frac{\sum_{i=1}^{n} \hat{\varepsilon}_i^2}{\sum_{i=1}^{n} (y_i \;-\; \bar{y})^2} \;. \tag{3.41}$$

The linear least squares regression in R gives for the data in Fig. 3.19:

```
res <- lm(y ~ x)
summary(res)

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-2.55541 -0.64589  0.05834  0.66114  2.42824

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.09223    0.12103   17.29   <2e-16 ***
x            0.46427    0.03417   13.59   <2e-16 ***
---
```

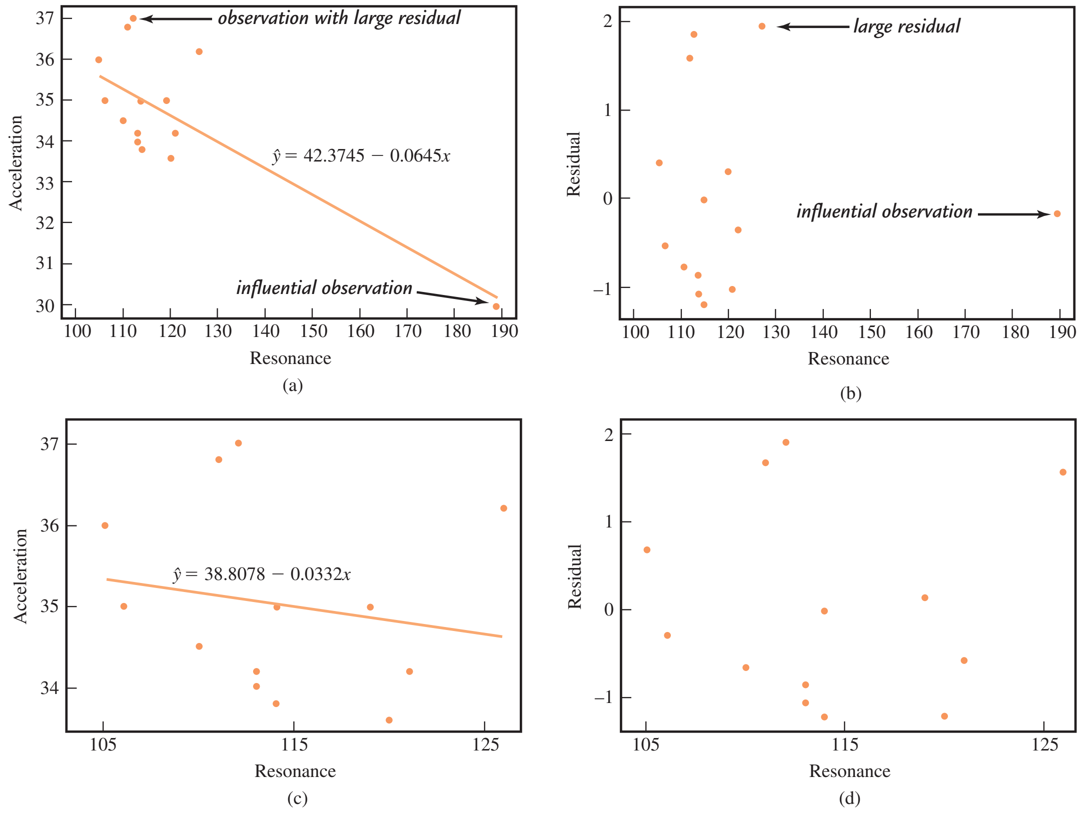

Figure 3.20: An outlier observation is far to the right but has large influence on the regression line. (a) scatter plot for the full sample; (b) residual plot for the full sample; (c) scatter plot when the outlier observation is deleted; (d) residual plot when the outlier observation is deleted. Figure from R. Peck and Devore [2009].

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

Residual standard error: 1.014 on 98 degrees of freedom
Multiple R-squared:  0.6532,    Adjusted R-squared:  0.6496
F-statistic: 184.6 on 1 and 98 DF,  p-value: < 2.2e-16
```

The confidence intervals are supplied as standard error and the t-statistic is supplied together with a p-value. Furthermore, the R^2 statistic is given.

An outlier can have large influence on the fitting of a regression line. Fig. 3.20 shows a fitting with an outlier in Fig. 3.20(a) and without the outlier in Fig. 3.20(c). The residuals are shown in Fig. 3.20(b) with the outlier and in Fig. 3.20(d) without the outlier. The single outlier observation has a large effect on the slope which is estimated by a least squares fit. Interestingly, the outlier does not possess the largest residual. Therefore, identification of outliers by residuals after a least squares fit is not always possible.

Next we show Anscombe's 4 regression data sets. These data sets have the same statistical properties (mean, variance, correlation, regression line, etc.), yet are quite different. Fitting the 4 data sets by a regression line lead to almost the same line:

```
$lm1
             Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 3.0000909  1.1247468 2.667348 0.025734051
x1          0.5000909  0.1179055 4.241455 0.002169629

$lm2
            Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 3.000909  1.1253024 2.666758 0.025758941
x2          0.500000  0.1179637 4.238590 0.002178816

$lm3
             Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 3.0024545  1.1244812 2.670080 0.025619109
x3          0.4997273  0.1178777 4.239372 0.002176305

$lm4
             Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 3.0017273  1.1239211 2.670763 0.025590425
x4          0.4999091  0.1178189 4.243028 0.002164602
```

Fig. 3.21 shows the regression line and the data to which it is fitted. The data sets are quite different, yet lead to the same regression line. Consequently, the statistical properties and the regression line do not fully characterize the data set.

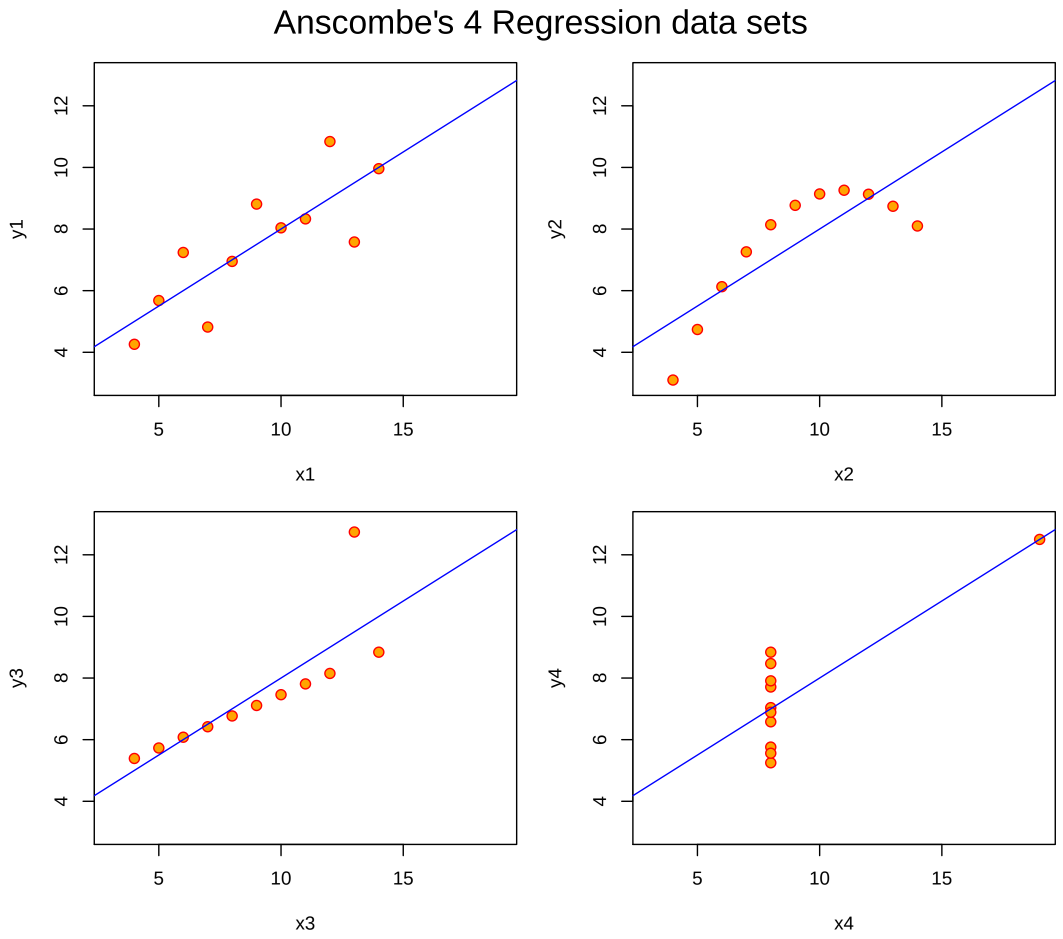

Figure 3.21: Anscombe's 4 regression data sets which show that the same line is fitted but the observations are quite different.

Chapter 4

Summarizing Multivariate Data

We now move on to data with more than two variables. This is the most common case in data analysis as in general more than two features can be extracted from the raw data. Further, feature construction allows to create new features. In some cases the number of features is very large like for gene expression profiles, where for each gene we have measurement — humans have 20,000 genes.

Multivariate data can be summarized by *descriptive* and *generative* methods. In the descriptive framework (see Fig. 4.1) the model maps or transforms observations to another representation which has desired characteristics in terms of components, factors, or codes. Typical cases are projection methods (principal component analysis, independent component analysis, projection pursuit). Projection methods can be described as matrix factorizations, where the matrix of observations is represented by the factor matrix multiplied by the inverse mapping matrix. Goal of the descriptive framework is to represent the observations in an appropriate way, e.g. by a desired density of the components, by a low dimensional representation, by high variance of the components (large information), by non-Gaussian components, or statistically independent components. For most descriptive methods the dimensionality of the original data is higher than for the representations, that is, the number of factors / components. Descriptive methods are used to represent the input compactly and non-redundantly for data storage or transmission, for data visualization, or for feature selection. In machine learning, besides for data visualization, descriptive methods are used as preprocessing methods for subsequent data analysis. Goal in these cases is dimensionality reduction and feature selection in order to obtain simpler models with higher generalization capability. Further simpler models and in particular models with few features are easier to interpret. In biology simpler models give more insights into the biological processes and allow to create new hypotheses.

In the generative framework (see Fig. 4.2) the goal is to model or to simulate the real world by generating model samples with the same underlying distribution as the real world observations. The selected models describe or represent the data generation process. The data generation process has random components which drive the process and are included into the model. Important for the generative approach is to include into the model as much prior knowledge about the real world domain as possible. Prior knowledge restricts the degree of freedom in the model class and, thereby, achieves higher modeling quality. Advantages of generative models:

- determining model parameters which are important for experts like calcium concentration in a cell, distribution of ion channels on the cell membrane, rate of a metabolic process, pH value, etc.,

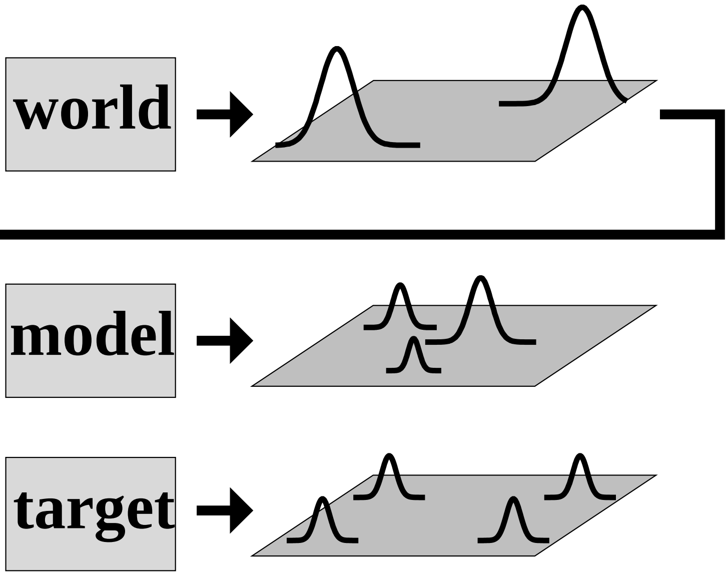

Figure 4.1: The descriptive framework is depicted. Observations are mapped through a model to factors/components/codes which have desired properties like a target density, low dimensionality, high variance, large information, non-Gaussianity, or independent components.

- generating new simulated observations,
- simulating the data generation process in unknowns regimes (new parameter setting),
- assessing the noise and the signal in the data which may supply noise and quality measures for the measurements like the signal-to-noise ratio,
- supplying distributions and error bars for latent variables (factors and components) and observations, e.g. the 95% confidence interval of factors,
- detection of outliers as very unlikely observations,
- detection and correction of noise in the observations.

If a descriptive model has a unique inverse, then the generative framework can be applied in the descriptive context. Within the generative framework a model is selected which models the observations. Subsequently, the inverse model of the model selected by the generative approach gives descriptive model. Density estimation, projection pursuit and vector quantization can be treated is such a way Hinton and Sejnowski [1999], Attias [1999]. Example of descriptive methods that do not have inverse models are *principal curves* Mulier and Cherkassky [1995], Ritter et al. [1992], which are a nonlinear generalization of principal components Hastie and Stuetzle [1989]. The first and most basic methods, that we will present are descriptive models like principal component analysis or multidimensional scaling. However generative models will also be considered like density estimation, factor analysis, independent component analysis, generative topographic mapping, etc.

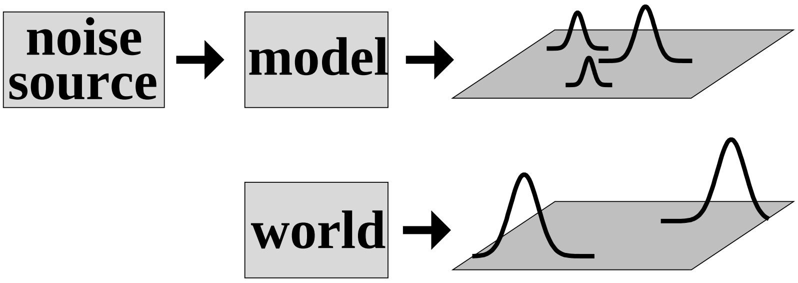

Figure 4.2: The generative framework is depicted. A noise source “drives” the model and produces model samples which should match the distribution of the observations.

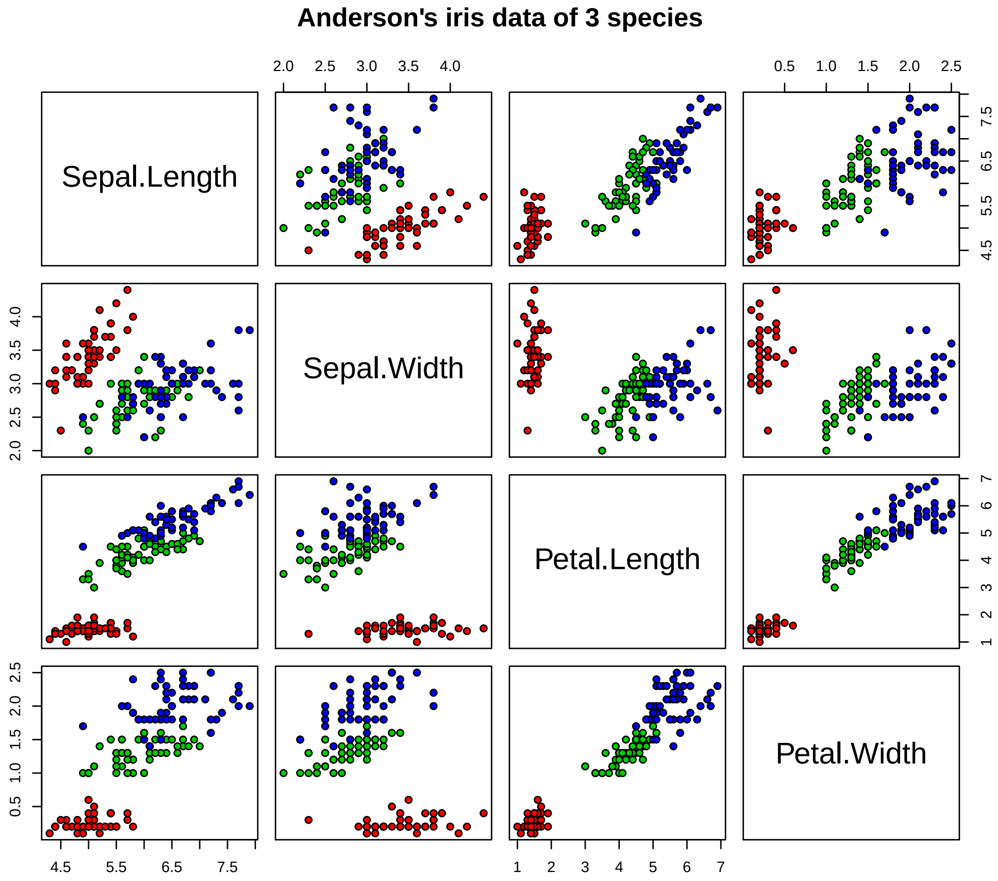

Figure 4.3: Matrix of scatter plots for the Anderson's iris data.

4.1 Matrix of Scatter Plots

A first approach to summarize multivariate data is to apply bivariate data summarization to all pairs of variables. However this is only feasible for few variables / features because the number of combinations increases quadratically with the number of features. For the iris data set from Section 2.2 we produce a matrix of scatter plots by the R command:

```
pairs(iris[1:4], main = "Anderson's iris data of 3 species",
+ pch = 21, bg = c("red", "green3", "blue")[unclass(iris$Species)])
```

The result is shown in Fig. 4.3. For the 4 features of this data set the pairings of variables can be comprehended by a human.

4.2 Principal Component Analysis

Principal Component Analysis (PCA) Jöreskog [1967], Everitt [1984], Neal and Dayan [1997] also known as *Karhunen-Loéve transform* (KTL) or as *Hotelling transform* makes a transformation of the coordinate system so that the data has largest variance along the first coordinate, the second largest data variance is along the second coordinate, etc. The coordinates, that are vectors, are called *principal components*. Fig. 4.4 shows the principal components of a two-dimensional data set and Fig. 4.5 shows how the projection onto the first principal component is extracted from data points.

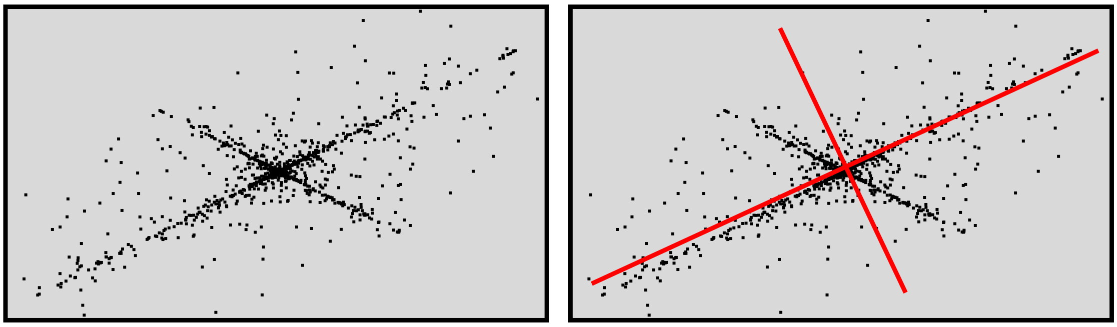

Figure 4.4: Principal component analysis for a two-dimensional data set. Left: the original data set. Right: The first principal component is shown as the line from lower left to upper right. The second principal component is orthogonal to the first component.

PCA is a very useful tool to summarize multivariate data because the first principal components capture most of the variation in the data. Therefore, they capture the most prominent structures in the data. Plotting observations by their projection onto the first two principal components often gives a first insight into the nature of the data.

Instead of a single scalar x, an observation is now represented as a vector $\boldsymbol{x}$ of m features: $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$. The data consisting of n observations $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ can be summarized in a data matrix $\boldsymbol{X}$ of size $n \times m$ which means n rows and m columns. The rows of $\boldsymbol{X}$ contain the n observations (each row contains one observation), while the columns contain the m features. We assume that the columns of $\boldsymbol{X}$, that are the features, have zero sample mean. Otherwise, the feature mean must be subtracted from each feature of each observation.

4.2.1 The Method

The $m \times m$ *sample covariance matrix* $\boldsymbol{C}$ of the features across the observations is defines as

$$C_{st} \; = \; \frac{1}{n} \sum_{i=1}^{n} x_{is} \, x_{it} \; , \tag{4.1}$$

where $x_{is} = (\boldsymbol{x}_i)_s$ and $x_{it} = (\boldsymbol{x}_i)_t$. For an unbiased estimate of the covariance matrix a factor $\frac{1}{n-1}$ instead of $\frac{1}{n}$ should be used. The covariance matrix $\boldsymbol{C}$ can be expressed as

$$\boldsymbol{C} \; = \; \frac{1}{n} \, \boldsymbol{X}^T \boldsymbol{X} \; = \; \frac{1}{n} \, \boldsymbol{U} \boldsymbol{D}_m \, \boldsymbol{U}^T \; , \tag{4.2}$$

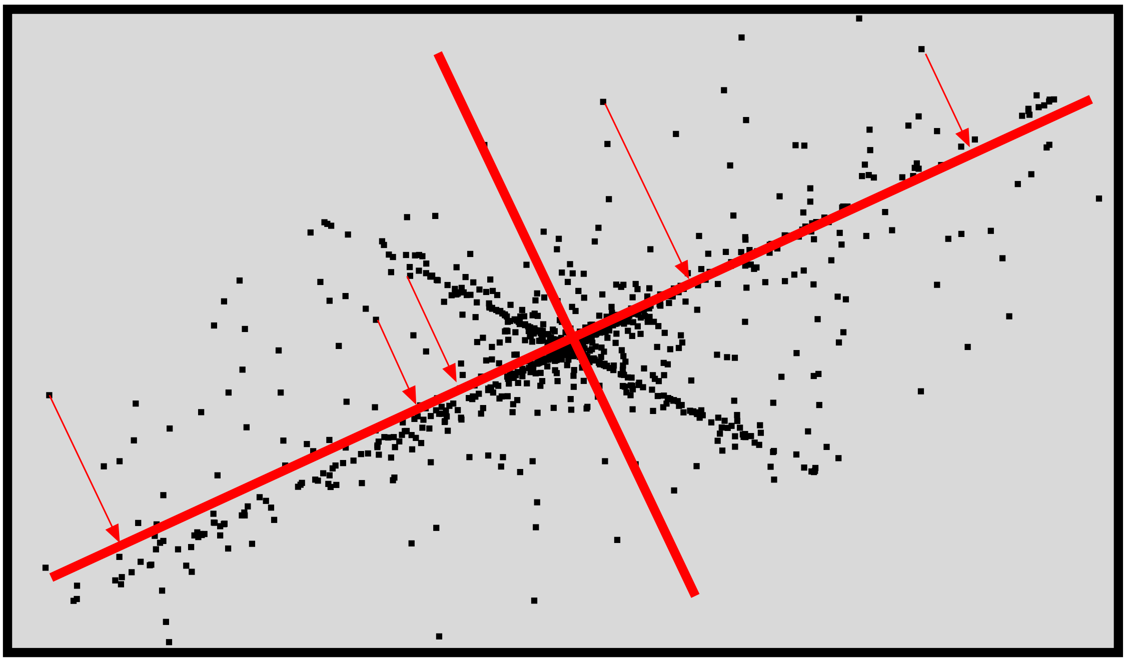

Figure 4.5: Principal component analysis for a two-dimensional data set. The projection onto the first principal component is extracted for data points. The points are projected onto the first component and then the distance to the origin is measured.

where $\boldsymbol{U}$ is an orthogonal $m \times m$ matrix and $\boldsymbol{D}_m$ is an $m \times m$ diagonal matrix. This decomposition of $\boldsymbol{C}$ into $\boldsymbol{U}$ and $\boldsymbol{D}_m$ is the *eigendecomposition* or *spectral decomposition* of $\boldsymbol{C}$. This decomposition exists because $\boldsymbol{C}$ is a symmetric positive definite matrix. The diagonal entries of $\boldsymbol{D}_m$ are called *eigenvalues* and the column vectors $\boldsymbol{u}_i = [\boldsymbol{U}]_i$ are called *eigenvectors*. We assume that the eigenvalues of $\boldsymbol{D}_m$ are sorted decreasingly, so that the first value is the largest eigenvalue. $\boldsymbol{C}$ as a symmetric real matrix is always diagonalizable and, since it is positive definite, its eigenvalues are larger than or equal to zero. In the context of PCA, the eigenvectors $\boldsymbol{u}_i$ are called the *principal components*, where the first principal component corresponds to the largest eigenvalue.

We assume that $n \geq m$ and that at least m linear independent observations exist, in order to ensure that $\boldsymbol{C}$ has full rang. To ensure $n \geq m$, typically feature selection is performed prior to a PCA analysis. Unsupervised feature selection may be based on variability, signal strength measured by the range, correlation between features (only one feature is kept if two features are highly correlated), non-Gaussianity, etc.

The *singular value decomposition* (SVD) of an $n \times m$ matrix $\boldsymbol{X}$ is

$$\boldsymbol{X} = \boldsymbol{V}\, \boldsymbol{D}\, \boldsymbol{U}^T , \tag{4.3}$$

where $\boldsymbol{U}$ is an orthogonal $m \times m$ matrix, $\boldsymbol{V}$ an orthogonal $n \times n$ matrix, and $\boldsymbol{D}$ is a diagonal (diagonal for the first m rows) $n \times m$ matrix with positive entries, the *singular values*. The diagonal values of $\boldsymbol{D}$ are sorted decreasingly, so that the first value is the largest value (the largest singular value), the second value is the second largest value, etc. Computing $\boldsymbol{X}^T\boldsymbol{X}$, we see that $\boldsymbol{D}_m = \boldsymbol{D}^T\, \boldsymbol{D}$ (the eigenvalues are the singular values squared) and $\boldsymbol{U}$ is the same orthogonal matrix as in the eigendecomposition. SVD is often used to perform PCA.

For performing PCA, it is sufficient to know $\boldsymbol{U}$, because the projection of feature vector $\boldsymbol{x}$ onto the principal directions is given by $\boldsymbol{U}^T\boldsymbol{x}$. Therefore, the data $\boldsymbol{X}$ is projected onto $\boldsymbol{U}$, which gives $\boldsymbol{Y}$:

$$\boldsymbol{Y} \;=\; \boldsymbol{X}\,\boldsymbol{U} \;=\; \boldsymbol{V}\,\boldsymbol{D}\,. \tag{4.4}$$

We see that the SVD automatically provides the PCA projections via $\boldsymbol{V}\,\boldsymbol{D}$. For single observations $\boldsymbol{x}$ that is

$$\boldsymbol{y} \;=\; \boldsymbol{U}^T\boldsymbol{x}\,. \tag{4.5}$$

In principle, PCA is a matrix decomposition problem:

$$\boldsymbol{X} \;=\; \boldsymbol{Y}\boldsymbol{U}^T\,, \tag{4.6}$$

where $\boldsymbol{U}$ is orthogonal, $\boldsymbol{Y}^T\,\boldsymbol{Y} \;=\; \boldsymbol{D}_m$ (the $\boldsymbol{y}$ are orthogonal, that is they are decorrelated), and the eigenvalues of $\boldsymbol{D}_m$ are sorted decreasingly. For single observations that is

$$\boldsymbol{x} \;=\; \boldsymbol{U}\,\boldsymbol{y}\,. \tag{4.7}$$

The SVD allows an outer product representation of the matrix $\boldsymbol{X}$:

$$\begin{aligned}\boldsymbol{X} \;=\; \sum_{i=1}^{m} D_{ii}\;\boldsymbol{v}_i\boldsymbol{u}_i^T \;=\; \\ \sum_{i=1}^{m} \boldsymbol{y}_i\boldsymbol{u}_i^T\,,\end{aligned} \tag{4.8}$$

where u_i is the i-th orthogonal column vector of $\boldsymbol{U}$, $\boldsymbol{v}_i$ is the i-th orthogonal column vector of $\boldsymbol{V}$, and $\boldsymbol{y}_i \;=\; D_{ii}\,\boldsymbol{v}_i$.

Iterative methods for PCA are sometimes to prefer if the dimension m is large or if on-line methods should be implemented. Most famous is Oja's rule Oja [1982]. If the current projection is

$$t \;=\; \boldsymbol{u}^T\boldsymbol{x} \tag{4.9}$$

then Oja's rule is

$$\boldsymbol{u}^{\mathrm{new}} \;=\; \boldsymbol{u} \;+\; \eta\;\left(t\,\boldsymbol{x} \;-\; t^2\,\boldsymbol{u}\right)\,, \tag{4.10}$$

where η is the learning rate.

The eigenvectors of $\boldsymbol{C}$ are the fixed points of Oja's rule and only the eigenvector with largest eigenvalue is a stable fixed point.

$$\begin{aligned}\mathrm{E}_{\boldsymbol{x}}(\boldsymbol{u}^{\mathrm{new}}) \;&=\; \boldsymbol{u} \;+\; \eta\,\mathrm{E}_{\boldsymbol{x}}\left(\boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{u}) \;-\; (\boldsymbol{u}^T\boldsymbol{x})(\boldsymbol{x}^T\boldsymbol{u})\,\boldsymbol{u}\right) \\ &=\; \boldsymbol{u} \;+\; \eta\;\left(\mathrm{E}_{\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{u} \;-\; \left(\boldsymbol{u}^T E_{\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{u}\right)\;\boldsymbol{u}\right) \\ &=\; \boldsymbol{u} \;+\; \eta\;\left(\boldsymbol{C}\boldsymbol{u} \;-\; (\boldsymbol{u}^T\boldsymbol{C}\boldsymbol{u})\,\boldsymbol{u}\right)\,.\end{aligned} \tag{4.11}$$

If $\boldsymbol{u}$ is and eigenvector of $\boldsymbol{C}$ with eigenvalue λ then

$$\mathrm{E}_{\boldsymbol{x}}(\boldsymbol{u}^{\mathrm{new}}) \;=\; \boldsymbol{u} \;+\; \eta\;(\lambda\boldsymbol{u} \;-\; \lambda\,\boldsymbol{u}) \;=\; \boldsymbol{u}\,. \tag{4.12}$$

Therefore each eigenvector of $\boldsymbol{C}$ is a fixed point of Oja's rule.

4.2.2 Variance Maximization

The first principal component $\boldsymbol{u}_1$ is the direction of the maximum possible data variance:

$$\boldsymbol{u}_1 \; = \; \underset{\|\boldsymbol{u}\|=1}{\arg\max} \sum_{i=1}^{n} \left(\boldsymbol{u}^T \, \boldsymbol{x}_i\right)^2 \; . \tag{4.13}$$

This can easily be seen because

$$\begin{aligned} \sum_{i=1}^{n} \left(\boldsymbol{u}^T \boldsymbol{x}_i\right)^2 \; &= \; \sum_{i=1}^{n} \left(\boldsymbol{u}^T \boldsymbol{x}_i\right) \left(\boldsymbol{x}_i^T \boldsymbol{u}\right) \; = \\ & \boldsymbol{u}^T \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T \boldsymbol{u} \; = \; n \, \boldsymbol{u}^T \boldsymbol{C} \boldsymbol{u} \; . \end{aligned} \tag{4.14}$$

With $\boldsymbol{C} = \sum_{i=1}^{m} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T$, $\boldsymbol{u} = \sum_{i=1}^{m} a_i \boldsymbol{u}_i$, and $\sum_{i=1}^{m} a_i^2 = 1$ we have

$$\boldsymbol{u}^T \boldsymbol{C} \boldsymbol{u} \; = \; \sum_{i=1}^{m} \lambda_i a_i^2 \tag{4.15}$$

and $\sum_{i=1}^{m} a_i^2 = 1$. The value $\sum_{i=1}^{m} \lambda_i a_i^2$ is maximal for $a_1 = 1$ and all other $a_i = 0$, because all $\lambda_i > 0$ and λ_1 is the largest eigenvalue.

Furthermore, principal components correspond to the direction of the maximum possible variance orthogonal to all previous components. If we remove the subspace of all previous components $1, \ldots, k$:

$$\boldsymbol{x}_i^k \; = \; \boldsymbol{x}_i \; - \; \sum_{t=1}^{k-1} \left(\boldsymbol{u}_t^T \, \boldsymbol{x}_i\right) \boldsymbol{u}_t \tag{4.16}$$

then the k-th principal component is the direction of the maximum data variance:

$$\boldsymbol{u}_k \; = \; \underset{\|\boldsymbol{u}\|=1}{\arg\max} \sum_{i=1}^{n} \left(\boldsymbol{u}^T \, \boldsymbol{x}_i^k\right)^2 \; . \tag{4.17}$$

This can inductively been proved analog to the first principal component. Since the components are sorted, the first l components span the l-dimensional subspace with maximal data variance.

4.2.3 Uniqueness

Is PCA unique or not, that is, is there only one PCA solution. Multiple solutions may fulfill the PCA criteria. We consider the decomposition

$$\boldsymbol{X} \; = \; \boldsymbol{Y} \boldsymbol{U}^T \, , \tag{4.18}$$

where $\boldsymbol{U}$ is orthogonal, $\boldsymbol{Y}^T \, \boldsymbol{Y} \; = \; \boldsymbol{D}_m$ with $\boldsymbol{D}_m$ as m-dimensional diagonal matrix, and the eigenvalues of $\boldsymbol{D}_m$ are sorted decreasingly.

PCA is unique up to signs, if the eigenvalues of the covariance matrix are different from each other.

Begin proof
To prove this statement, assume another representation

$$\boldsymbol{X} \;=\; \boldsymbol{Y}'\boldsymbol{U}'^{T} \,, \tag{4.19}$$

where $\boldsymbol{U}'$ is orthogonal, $(\boldsymbol{Y}')^T\boldsymbol{Y}' \;=\; \boldsymbol{D}'_m$ with $\boldsymbol{D}'_m$ as m-dimensional diagonal matrix, and the eigenvalues of $\boldsymbol{D}'_m$ are sorted decreasingly.

If eigenvalues of $\boldsymbol{D}_m$ are different from each other, then at most one eigenvalue can be zero. If one eigenvalue of $\boldsymbol{D}_m$ is zero, the observations do not have any variance in the direction of the according eigenvector. This direction is unique, becomes principal component $\boldsymbol{u}_m$, and can be removed from the data. Subsequent, we can perform PCA on the remaining $(m-1)$-dimensional space, where all eigenvalues are larger than zero.

We assume that all eigenvalues of $\boldsymbol{D}_m$ (therefore also of $\boldsymbol{Y}$) are larger than zero. Therefore a matrix $\boldsymbol{A} = \boldsymbol{Y}^{-1}\boldsymbol{Y}'$ exists with

$$\boldsymbol{Y}' \;=\; \boldsymbol{Y}\,\boldsymbol{A}\,. \tag{4.20}$$

We obtain

$$\boldsymbol{X} \;=\; \boldsymbol{Y}\,\boldsymbol{U}^T \;=\; \boldsymbol{Y}'\,\boldsymbol{U}'^{T} \;=\; \boldsymbol{Y}\,\boldsymbol{A}\,\boldsymbol{U}'^{T} \tag{4.21}$$

after multiplying with $\boldsymbol{Y}^{-1}$ from the left we get

$$\boldsymbol{U}^T \;=\; \boldsymbol{A}\,\boldsymbol{U}'^{T}\,. \tag{4.22}$$

Since $\boldsymbol{U}$ and $\boldsymbol{U}'$ are orthogonal, we obtain

$$\boldsymbol{I} \;=\; \boldsymbol{U}^T\boldsymbol{U} \;=\; \boldsymbol{A}\boldsymbol{U}'\boldsymbol{U}'^{T}\boldsymbol{A}^T \;=\; \boldsymbol{A}\boldsymbol{A}^T\,. \tag{4.23}$$

Therefore A is an orthogonal matrix and

$$\boldsymbol{U}' \;=\; \boldsymbol{U}\,\boldsymbol{A}\,. \tag{4.24}$$

We obtain

$$\boldsymbol{D}'_m \;=\; (\boldsymbol{Y}')^T\,\boldsymbol{Y}' \;\;=\; \boldsymbol{A}^T\,\boldsymbol{Y}^T\,\boldsymbol{Y}\,\boldsymbol{A} \;=\; \boldsymbol{A}^T\,\boldsymbol{D}_m\boldsymbol{A}\,. \tag{4.25}$$

Thus, the eigenvalues of $\boldsymbol{D}'_m$ match the diagonal elements of $\boldsymbol{D}_m$. Further the i-th eigenvalue λ_i of the covariance matrix $\boldsymbol{C}$ is $\lambda_i = D_{ii}$. According to our assumption, the eigenvalues are sorted in both $\boldsymbol{D}'_m$ and $\boldsymbol{D}_m$. The sorting is unique, because we assumed mutually different eigenvalues. Therefore we have

$$\boldsymbol{D}'_m \;=\; \boldsymbol{D}_m\,. \tag{4.26}$$

It follows that

$$\boldsymbol{A}\,\boldsymbol{D}_m \;=\; \boldsymbol{D}_m\,\boldsymbol{A} \tag{4.27}$$

and

$$[\boldsymbol{A}\,\boldsymbol{D}_m]_{ij} \;=\; a_{ij}\,D_{jj} \;=\; a_{ij}\,d_{ii} \;=\; [\boldsymbol{D}_m\,\boldsymbol{A}]_{ij} \tag{4.28}$$

which gives

$$a_{ij}\,(D_{jj} \;-\; D_{ii}) \;=\; 0\;. \tag{4.29}$$

For $i \neq j$ our assumption is that $D_{jj} \;\neq\; D_{ii}$ ($\lambda_j = D_{jj}$), therefore we deduce $a_{ij} \;=\; 0$. Hence, $\boldsymbol{A}$ is diagonal and orthogonal. Consequently, $\boldsymbol{A}$ is diagonal and contains only ones and minus ones on its diagonal. Thus, PCA is unique up to signs, if the eigenvalues are mutually different. **End proof**

4.2.4 Properties of PCA

- The projection of the data on the first principal component (PC) has maximal variance of all possible one-dimensional projections. That means $\boldsymbol{u}_1$ maximizes

  $$\boldsymbol{u}^T\,\boldsymbol{C}\,\boldsymbol{u} \;\text{ s.t. }\; \|\boldsymbol{u}\| = 1\;. \tag{4.30}$$

  The first l PCs maximize

  $$\sum_{i=1}^{l} \boldsymbol{u}_i^T\,\boldsymbol{C}\,\boldsymbol{u}_i \;\text{ s.t. }\; \boldsymbol{u}_i^T\,\boldsymbol{u}_j = \delta_{ij}\;. \tag{4.31}$$

- The projections onto PCs have zero means:

  $$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{u}_k^T\,\boldsymbol{x}_i \;=\; \boldsymbol{u}_k^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\right) \;=\; \boldsymbol{u}_k^T\boldsymbol{0} \;=\; 0\;. \tag{4.32}$$

- The projections onto PCs are mutually uncorrelated (second moment), that is, they are orthogonal to each other. We already expressed this by $\boldsymbol{Y}^T\boldsymbol{Y} \;=\; \boldsymbol{I}$ but it can also be seen at

  $$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{u}_t^T\,\boldsymbol{x}_i)\,(\boldsymbol{u}_s^T\,\boldsymbol{x}_i) \;&=\; \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{u}_t^T\,\boldsymbol{x}_i)\,(\boldsymbol{x}_i^T\,\boldsymbol{u}_s) \\ &=\; \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{u}_t^T\,(\boldsymbol{x}_i\,\boldsymbol{x}_i^T)\,\boldsymbol{u}_s \\ &=\; \boldsymbol{u}_t^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\,\boldsymbol{x}_i^T\right)\boldsymbol{u}_s \\ &=\; \boldsymbol{u}_t^T\,\boldsymbol{C}\,\boldsymbol{u}_s \;=\; \lambda_s\,\boldsymbol{u}_t^T\,\boldsymbol{u}_s \;=\; 0\;. \end{aligned} \tag{4.33}$$

  For the last equation we used $\boldsymbol{C} \;=\; \sum_{j=1}^{m}\lambda_j\boldsymbol{u}_j\boldsymbol{u}_j^T$. Therefore correlation coefficients between projections of the observations onto PCs are zero.

- The sample variance of the k-th projection is equal to the k-th eigenvalue of the sample covariance matrix $\boldsymbol{C}$

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{u}_k^T\boldsymbol{x}_i\right)^2 &= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{u}_k^T\left(\boldsymbol{x}_i\boldsymbol{x}_i^T\right)\boldsymbol{u}_k \\ &= \boldsymbol{u}_k^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\,\boldsymbol{x}_i^T\right)\boldsymbol{u}_k \\ &= \boldsymbol{u}_k^T\,\boldsymbol{C}\,\boldsymbol{u}_k \;=\; \lambda_k\,\boldsymbol{u}_k^T\,\boldsymbol{u}_k \;=\; \lambda_k\;.\end{aligned} \tag{4.34}$$

  where λ_k is the k-th eigenvalue of the covariance matrix $\boldsymbol{C}$.

- PCs are ranked decreasingly according to their eigenvalues which are the variances in the PC directions.

- The first l PCs minimize the mean-squared error.

  The representation of $\boldsymbol{x}$ by $\hat{\boldsymbol{x}}$ with the first l PCs is

$$\hat{\boldsymbol{x}} \;=\; \sum_{k=1}^{l}\boldsymbol{u}_k\,\boldsymbol{u}_k^T\boldsymbol{x}\;, \tag{4.35}$$

  where

$$\boldsymbol{C} \;=\; \sum_{k=1}^{m}\lambda_k\,\boldsymbol{u}_k\,\boldsymbol{u}_k^T\;. \tag{4.36}$$

For the approximation of $\boldsymbol{x}$ by $\hat{\boldsymbol{x}}$, the mean-squared error is

$$\begin{aligned}
\mathrm{E}\left(\|\boldsymbol{x} \ - \ \hat{\boldsymbol{x}}\|^2\right) \ &= \ \mathrm{E}\left(\boldsymbol{x}^T\boldsymbol{x} \ - \ 2\,\boldsymbol{x}^T \ \hat{\boldsymbol{x}} \ + \ \hat{\boldsymbol{x}}^T\hat{\boldsymbol{x}}\right) \qquad (4.37)\\
&= \ \mathrm{E}\left(\mathrm{Tr}\left(\boldsymbol{x}\boldsymbol{x}^T\right) \ - \ 2\ \mathrm{Tr}\left(\sum_{k=1}^{l} \boldsymbol{u}_k \ \boldsymbol{u}_k^T \boldsymbol{x}\boldsymbol{x}^T\right) \ + \ \mathrm{Tr}\left(\sum_{k=1}^{l} \boldsymbol{u}_k \ \boldsymbol{u}_k^T \boldsymbol{x}\boldsymbol{x}^T\right)\right)\\
&= \ \mathrm{Tr}\left(\mathrm{E}\left(\boldsymbol{x}\boldsymbol{x}^T\right) \ - \ 2\ \sum_{k=1}^{l} \boldsymbol{u}_k \ \boldsymbol{u}_k^T \mathrm{E}\left(\boldsymbol{x}\boldsymbol{x}^T\right) \ + \ \sum_{k=1}^{l} \boldsymbol{u}_k \ \boldsymbol{u}_k^T \mathrm{E}\left(\boldsymbol{x}\boldsymbol{x}^T\right)\right)\\
&= \ \mathrm{Tr}\left(\boldsymbol{C} \ - \ \sum_{k=1}^{l} \boldsymbol{u}_k \ \boldsymbol{u}_k^T \boldsymbol{C}\right)\\
&= \ \mathrm{Tr}\left(\boldsymbol{C} \ - \ \sum_{k=1}^{l} \boldsymbol{u}_k \ \boldsymbol{u}_k^T \sum_{k=1}^{m} \lambda_k \boldsymbol{u}_k \ \boldsymbol{u}_k^T\right)\\
&= \ \mathrm{Tr}\left(\sum_{k=1}^{m} \lambda_k \ \boldsymbol{u}_k \ \boldsymbol{u}_k^T \ \ - \ \sum_{k=1}^{l} \lambda_k \boldsymbol{u}_k \ \boldsymbol{u}_k^T\right)\\
&= \ \mathrm{Tr}\left(\sum_{k=l+1}^{m} \ \lambda_k \ \boldsymbol{u}_k \ \boldsymbol{u}_k^T\right)\\
&= \ \sum_{k=l+1}^{m} \ \lambda_k \ \mathrm{Tr}\left(\boldsymbol{u}_k \ \boldsymbol{u}_k^T\right)\\
&= \ \sum_{k=l+1}^{m} \ \lambda_k \ \mathrm{Tr}\left(\boldsymbol{u}_k^T \ \boldsymbol{u}_k\right)\\
&= \ \sum_{k=l+1}^{m} \ \lambda_k \ .
\end{aligned}$$

where λ_k is the square root of the k-th eigenvalue of $\boldsymbol{C}$ or the k-th singular value of $\boldsymbol{X}$.

Each representation of the data by projections to other l vectors $(\boldsymbol{u}')_k$ will have a larger mean squared error. Using the transformations of the last equation, we obtain for the mean squared error

$$\mathrm{Tr}\left(\boldsymbol{C} \ - \ \sum_{k=1}^{l} (\boldsymbol{u}')_k (\boldsymbol{u}')_k^T \boldsymbol{C}\right) \ . \qquad (4.38)$$

If $(\boldsymbol{u}')_k = \sum_{i=1}^{m} b_{ki} \boldsymbol{u}_i$ with $\sum_{i=1}^{m} b_{ki}^2 = 1$.

The mean squared error for projection onto the l vectors $(\boldsymbol{u}')_k$ is

$$\begin{aligned} \mathrm{Tr}\left(\boldsymbol{C} - \sum_{k=1}^{l} (\boldsymbol{u}')_k (\boldsymbol{u}')_k^T \boldsymbol{C} \right) & \qquad (4.39) \\ = \sum_{i=1}^{m} \lambda_i \ - \ \sum_{k=1}^{l} \sum_{i=1}^{m} b_{ki}^2 \lambda_i & \\ = \sum_{i=1}^{m} \lambda_i \left(1 \ - \ \sum_{k=1}^{l} b_{ki}^2 \right) \ . & \end{aligned}$$

The Hessian matrix of this objective with respect to the parameters b_{ki} has negative eigenvalues, therefore this is a strict concave function. The maximum principle states that the minimum of this objective is found on the boundary. That means $b_{ki} = 0$ or $b_{ki} = 1$. Therefore the $(\boldsymbol{u}')_k$ are a permutation of $\boldsymbol{u}_k$. $\sum_{k=l+1}^{m} \lambda'_k \geq \sum_{k=l+1}^{m} \lambda_k$ where equality is only achieved if the λ'_k is a permutation of λ_k for $l + 1 \leq k \leq m$. Therefore the first l vectors $(\boldsymbol{u}')_k$ are a permutation of the first l $\boldsymbol{u}_k$. If we assume that the eigenvectors are sorted according to the eigenvalues, then $(\boldsymbol{u}')_k = B u_k$ for $1 \leq k \leq l$. Thus, a projection onto other l vectors than the principal components leads to a larger mean squared error than those of PCA.

4.2.5 Examples

4.2.5.1 Iris Data Set

We revisit the iris data set from Section 2.2 and perform PCA on this data. The R command `princomp()` supplies the PCA solution:

```
xp <- princomp(iris[,1:4],scores=TRUE)
summary(xp)
Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4
Standard deviation     2.0494032 0.49097143 0.27872586 0.153870700
Proportion of Variance 0.9246187 0.05306648 0.01710261 0.005212184
Cumulative Proportion  0.9246187 0.97768521 0.99478782 1.000000000
```

We see that the first principal component explains 92% of the variance in the data. This means that the features are correlated and the variance driving this correlation is captured by principal component 1. Probably PC1 expresses the size of the blossom which is reflected in all four features.

Fig. 4.6 shows scatter plots for pairs of principal components, more precisely, scatter plots of the projection of the observations to pairs of PCs. The R command was

```
irisPC <- xp$scores
## irisPC <- sweep(as.matrix(iris[,1:4]),2,xp$center)%*%xp$loadings
lP <- length(colnames(irisPC))
colnames(irisPC) <- paste(colnames(irisPC),rep("(",lP),
+ as.character(round(100*xp$sdev^2/sum(xp$sdev^2))),rep("%)",lP),sep="")
op <- par(mfrow = c(3, 2), mar = 0.1+c(4,4,1,1), oma =  c(0, 0, 2, 0))
plot(irisPC[,c(1,2)], main = "PC1 and PC2",pch = 21,
+ bg = c("red", "green3","blue")[unclass(iris$Species)])
plot(irisPC[,c(1,3)], main = "PC1 and PC3",pch = 21,
+ bg = c("red", "green3","blue")[unclass(iris$Species)])
plot(irisPC[,c(1,4)], main = "PC1 and PC4",pch = 21,
+ bg = c("red", "green3","blue")[unclass(iris$Species)])
plot(irisPC[,c(2,3)], main = "PC2 and PC3",pch = 21,
+ bg = c("red", "green3","blue")[unclass(iris$Species)])
plot(irisPC[,c(2,4)], main = "PC2 and PC4",pch = 21,
+ bg = c("red", "green3","blue")[unclass(iris$Species)])
plot(irisPC[,c(3,4)], main = "PC3 and PC4",pch = 21,
+ bg = c("red", "green3","blue")[unclass(iris$Species)])
par(op)
```

Only PC1 helps to separate the species.

4.2.5.2 Multiple Tissue Data Set

We apply PCA to the multiple tissue microarray data set which is described in Section 2.3. Gene expression values for different tissue types for human and mouse are measured. The data set contains 102 samples for each of which expression values of 5,565 genes are available. Four distinct tissue types are indicated in the data: breast (Br), prostate (Pr), lung (Lu), and colon (Co). We want to see if PCA allows to identify these tissue types.

The projections to the principal components are obtained via a singular value decomposition:

```
sv <- svd(t(XMulti))
PCS <- sv$u%*%diag(sv$d)
```

The variable PCS contains the projections to the principal components.

Fig. 4.7 shows scatter plots for pairs of principal components, i.e. the projections of the observations to pairs of PCs. PC1 separates the prostate samples (green) from the rest. PC2 separates the colon samples (orange) but also breast samples (red). PC3 separates some lung samples (blue).

Next we perform variance filtering before PCA. For microarray data, variance filtering is justified because genes that are differentially expressed across the samples have higher variance. For such genes the noise variance and the variance due to the signal add up. Therefore, genes with largest variance are assumed to contain a signal and to have higher signal-to-noise ratio. The following filtered data sets are considered:

```
vv <- diag(var(t(XMulti)))
length(which(vv>2))
 101
length(which(vv>4))
 13
```

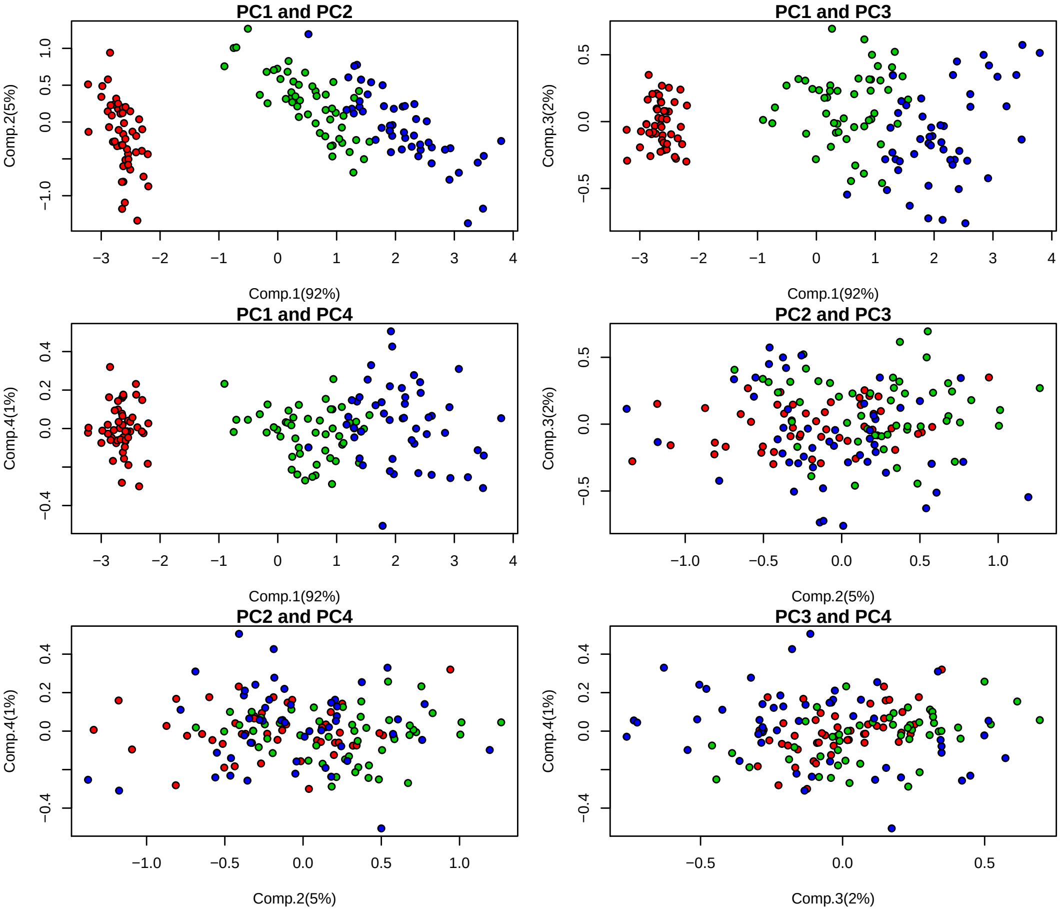

Figure 4.6: PCA applied to Anderson's iris data. The matrix shows scatter plots for pairs of principal components.

```
length(which(vv>5))
 5
XMultiF1 <- t(XMulti[which(vv>2),])   # 101
XMultiF2 <- t(XMulti[which(vv>4),])   # 13
XMultiF3 <- t(XMulti[which(vv>5),])   # 5
```

For the 101 genes with the highest variance, Fig. 4.8 shows scatter plots for pairs of principal components, i.e. the projections of the observations to pairs of PCs. Principal component 1 separates the prostate samples (green) from the rest. PC2 separates the colon samples (orange) from the rest. PC3 separates the breast samples (red) from the rest and at the same time lung samples (blue) from the rest. PCA on the filtered data set separates the tissues better than PCA on the whole data set.

PCA on the multiple tissue data with 13 genes that have largest variance is shown in Fig. 4.9. PC1 separates the prostate samples (green) from the rest. PC2 separates the colon samples (orange) from the rest. PC3 separates the breast samples (red) from the rest at one side but at the other side time lung samples (blue).

PCA on the multiple tissue data with 5 genes that have largest variance is shown in Fig. 4.10. Still PC1 separates the prostate samples (green) from the rest. However other tissues are difficult to separate. Four out of the 5 genes are highly correlated and give the same signal. Probably this signal is indicative for the prostate tissue.

```
cor(XMultiF3)
              ACPP        KLK2         KRT5        MSMB       TRGC2
ACPP   1.000000000  0.97567890 -0.004106762  0.90707887 0.947433227
KLK2   0.975678903  1.00000000 -0.029900946  0.89265825 0.951841913
KRT5  -0.004106762 -0.02990095  1.000000000 -0.05565599 0.008877815
MSMB   0.907078869  0.89265825 -0.055655985  1.00000000 0.870922667
TRGC2  0.947433227  0.95184191  0.008877815  0.87092267 1.000000000
```

In the GeneCards database `http://www.genecards.org` we find:

> ACPP “is synthesized under androgen regulation and is secreted by the epithelial cells of the prostate gland.”

further we find

> KLK2 “is primarily expressed in prostatic tissue and is responsible for cleaving pro-prostate-specific antigen into its enzymatically active form.”

and

> MSMB “is synthesized by the epithelial cells of the prostate gland and secreted into the seminal plasma.”

We now select genes which are not so closely correlated to each other. Toward this end we first cluster (see later in the course) the genes and then select one prototype from each cluster:

```
hc1 <-hclust(dist(t(XMultiF1)))
ct <- cutree(hc1,h=25)
table(ct)
ct
 1  2  3  4  5  6  7  8  9 10
21 14 12 21  6  4  2  9  3  9

l1 <- length(table(ct))
sel <- c()
for(i in 1:l1) {
+ sel <- c(sel,which(ct==i)[1])
+ }
XMultiF4 <- XMultiF1[,sel]
```

These 10 genes are not as closely related as the genes which are selected based on variance alone:

```
cor(XMultiF4)
               ABP1       ACPP      AKR1C1     ALDH1A3      ANXA8          APOD
ABP1     1.00000000 -0.1947766 -0.04224634 -0.21577195 -0.2618053 -0.3791812658
ACPP    -0.19477662  1.0000000 -0.22929893  0.88190657 -0.2978638  0.4964638048
AKR1C1  -0.04224634 -0.2292989  1.00000000 -0.07536066  0.4697886 -0.1793466620
ALDH1A3 -0.21577195  0.8819066 -0.07536066  1.00000000 -0.1727669  0.4113925823
ANXA8   -0.26180526 -0.2978638  0.46978864 -0.17276688  1.0000000 -0.1863923785
APOD    -0.37918127  0.4964638 -0.17934666  0.41139258 -0.1863924  1.0000000000
BST2    -0.02752210 -0.1858633  0.03341592 -0.18706898  0.1672327  0.0001475666
CA12    -0.03390577 -0.5266892  0.20825388 -0.55430511  0.1535930 -0.0861446268
CLDN3    0.33206818  0.3547601 -0.52997065  0.24516720 -0.6819272  0.2272871855
IGHA1   -0.14341643 -0.2835074  0.45479347 -0.08918854  0.2726503 -0.1157383141
                 BST2        CA12      CLDN3       IGHA1
ABP1    -0.0275221025 -0.03390577  0.3320682 -0.14341643
ACPP    -0.1858633000 -0.52668918  0.3547601 -0.28350737
AKR1C1   0.0334159199  0.20825388 -0.5299707  0.45479347
ALDH1A3 -0.1870689799 -0.55430511  0.2451672 -0.08918854
ANXA8    0.1672327418  0.15359297 -0.6819272  0.27265032
APOD     0.0001475666 -0.08614463  0.2272872 -0.11573831
BST2     1.0000000000  0.08971880 -0.1918497  0.16460367
CA12     0.0897187966  1.00000000 -0.3170681  0.17639489
CLDN3   -0.1918497331 -0.31706813  1.0000000 -0.39690211
IGHA1    0.1646036701  0.17639489 -0.3969021  1.00000000
```

Fig. 4.11 shows the PCA result. The tissues are not as well separated as with maximizing the variance of the genes because some highly variable genes are missed. Tissues can be separated but not with the same quality as with more genes.

Feature selection based on hierarchical clustering and variance maximization within one cluster:

```
hc1 <-hclust(dist(XMulti))
ct <- cutree(hc1,h=16)
table(ct)
ct
  1   2    3   4   5   6   7   8  9 10 11 12 13 14 15 16 17 18
682 126 1631 742 347 797 196 104 44 35  5  8 12  5 12 14  5 71
 19  20   21  22  23  24  25  26 27 28 29 30 31 32 33 34 35 36
 22   8   16  32  48  72   2  93 22 22 56  9 54  7  4  2 16 26
 37  38   39  40  41  42  43  44 45 46 47 48 49 50 51 52 53 54
  3   8   42   1   9   1   7  14  1  2  8  3  2 20  3  2  9  7
 55  56   57  58  59  60  61  62 63 64 65 66 67 68 69 70 71 72
  3   2    1   5   2   2   1   1  1  3  9  3  3  3  3  1  2  3
 73  74   75  76  77  78  79  80 81 82 83 84 85 86 87 88 89 90
  1   1    1   1   2   2   1   3  1  2  1  1  2  1  2  2  1  1
 91  92
  1   1
l1 <- length(table(ct))
sel <- c()
for(i in 1:l1) {
clas <- which(ct==i)
M <- which.max(diag(var(t(XMulti[clas,]))))
sel <- c(sel,clas[M])
}
XMultiF5 <- t(XMulti[sel,])
```

For each cluster the gene with maximal variance is selected. Fig. 4.12 shows the PCA result for feature selection based on hierarchical clustering, which gave 92 genes. Results are very similar to variance based feature selection. However one improvement is visible. PC3 separates breast samples (red) from lung samples (blue) which was not achieved by the other projections.

Feature selection based on hierarchical clustering but now the distance between genes is based on their correlation:

```
# Genes 2964 and 4663 are constant !!
# First remove these genes
XMultiC <-  XMulti[-c(2964,4663),]
D <- 1 - abs(cor(t(XMultiC)))
D <- as.dist(D)
hc1 <-hclust(D)
ct <- cutree(hc1,h=0.999)
l1 <- length(table(ct))
sel <- c()
for(i in 1:l1) {
clas <- which(ct==i)
M <- which.max(diag(var(t(XMultiC[clas,]))))
sel <- c(sel,clas[M])
}
```

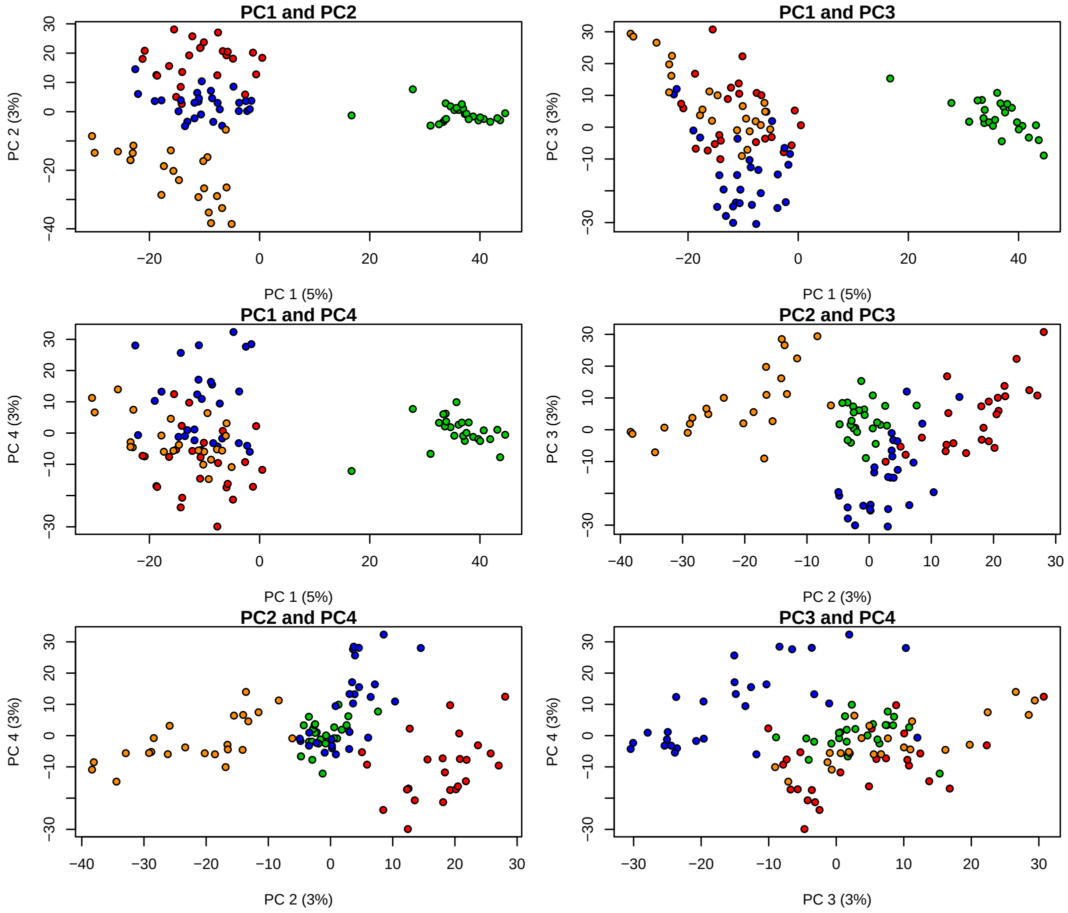

Figure 4.7: PCA applied to multiple tissue data. PC1 separates the prostate samples (green) from the rest. PC2 separates the colon samples (orange) from the rest. PC3 separates some lung samples (blue).

```
XMultiF7 <- t(XMultiC[sel,])
```

Fig. 4.13 shows the PCA result for feature selection based on hierarchical clustering based on the correlation coefficient matrix. For each of the 95 clusters the gene with maximal variance was selected. Again, the results are very similar to variance based feature selection. PC3 separates breast samples (red) from lung samples (blue) almost as good as in previous example.

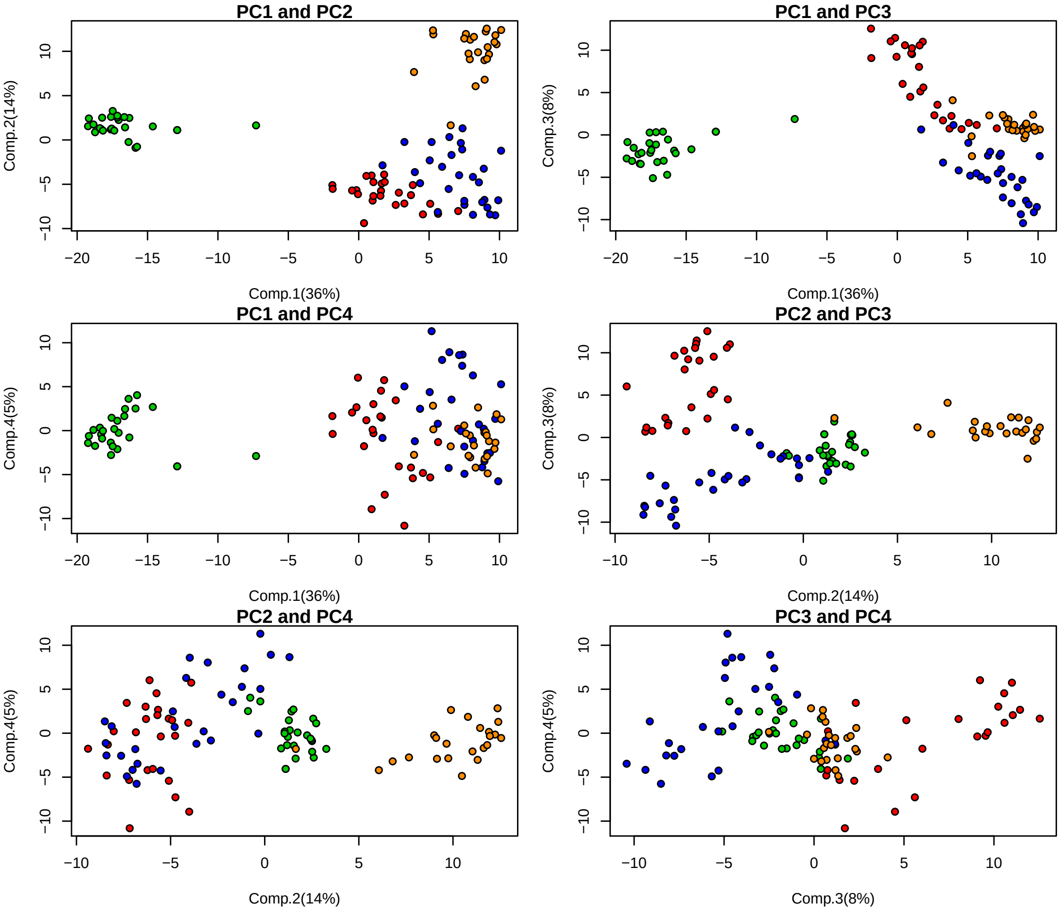

Figure 4.8: PCA applied to multiple tissue data with 101 most variable genes. PC1 separates the prostate samples (green) from the rest. PC2 separates the colon samples (orange) from the rest. To the left, PC3 separates the breast samples (red) from the rest but, to the right, it also separates lung samples (blue).

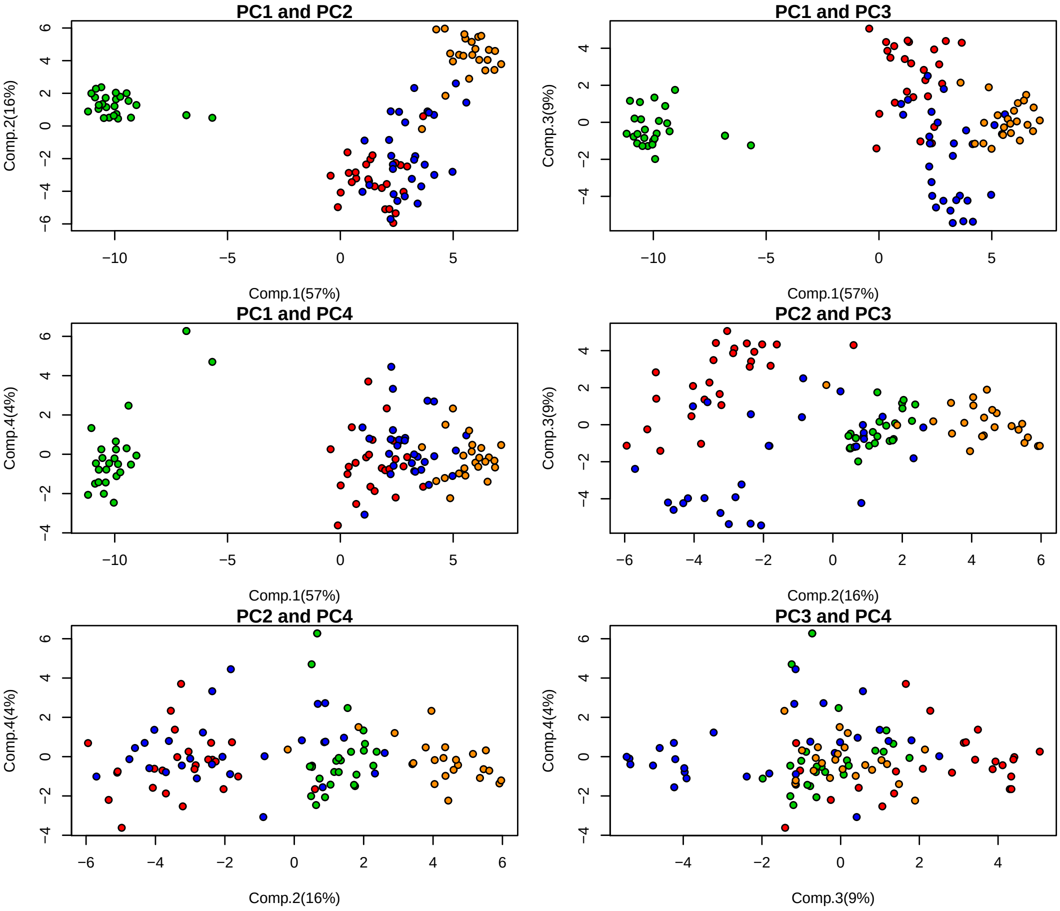

Figure 4.9: PCA applied to multiple tissue data with 13 most variable genes. Again PC1 separates the prostate samples (green) from the rest. However, the separation of colon samples (orange) by PC2 is worse than with 101 genes. Also the separation of the breast samples (red) and lung samples (blue) by PC3 is worse that with 101 genes.

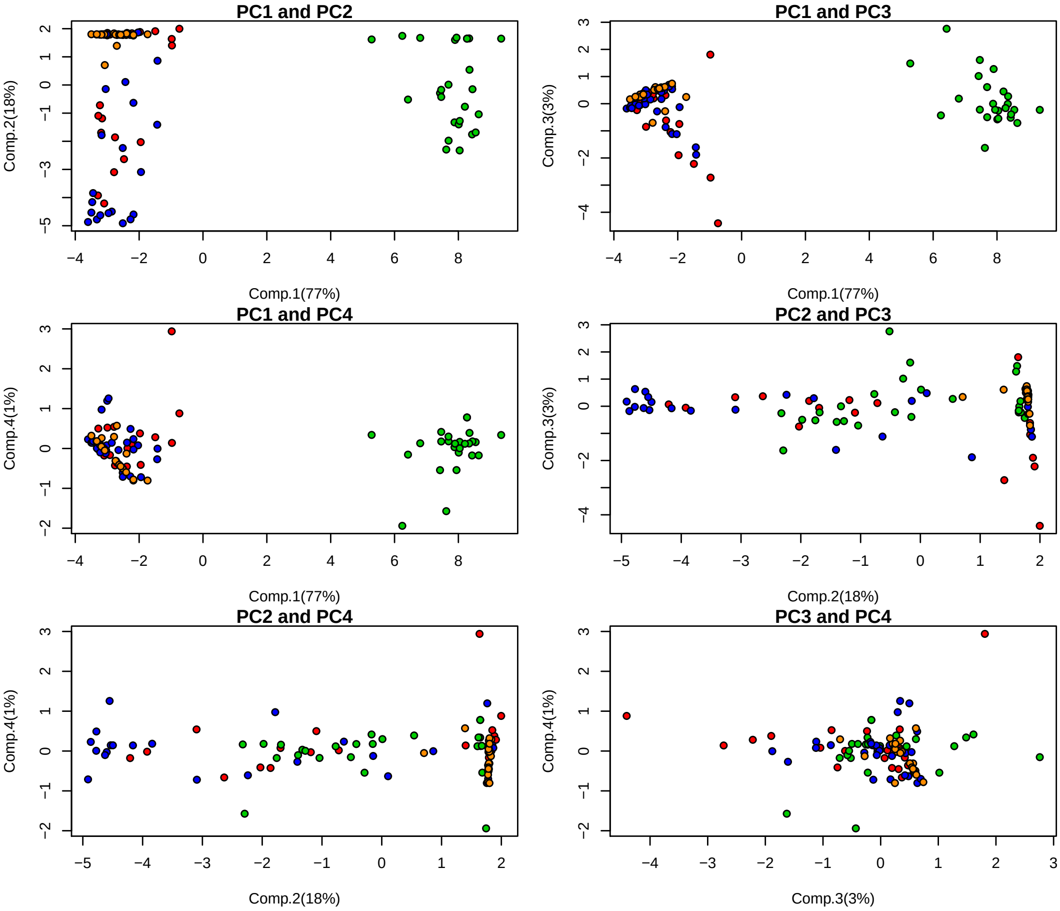

Figure 4.10: PCA applied to multiple tissue data with 5 most variable genes. Still PC1 separates the prostate samples (green) from the rest. However other tissues were not separated.

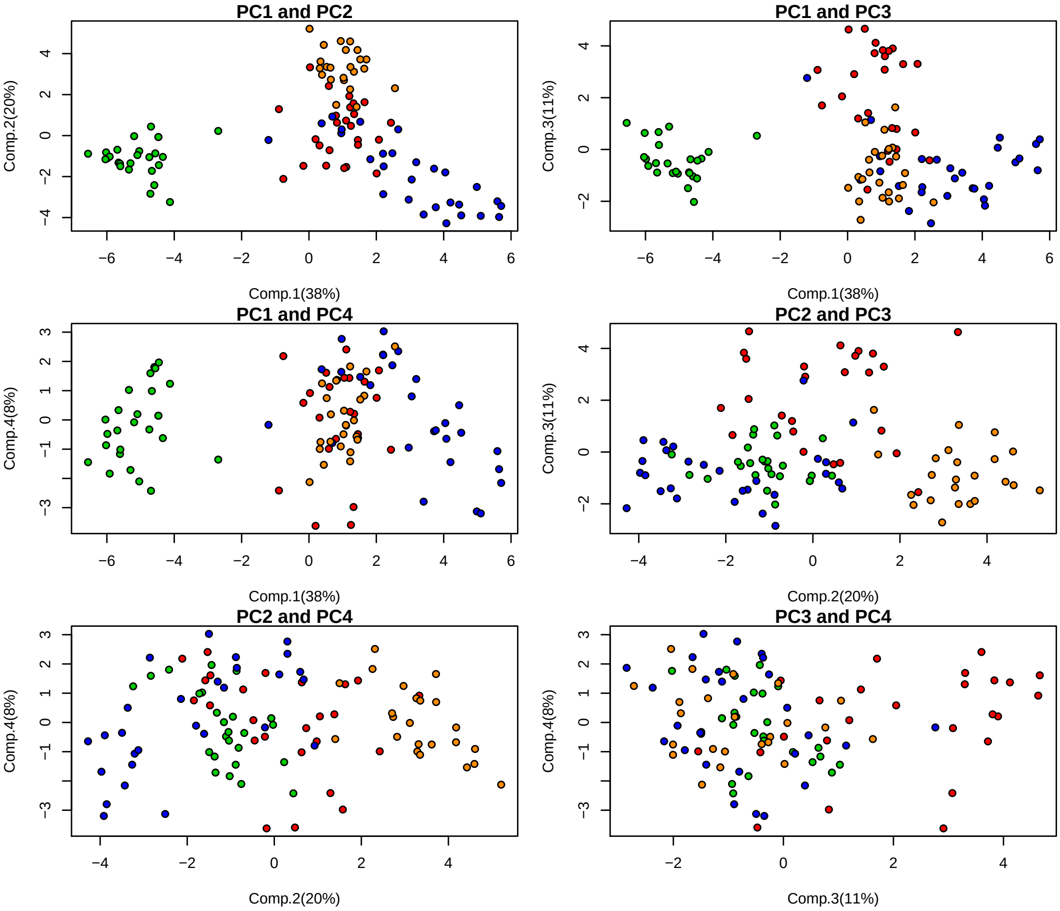

Figure 4.11: PCA applied to multiple tissue data with 10 genes which are not too closely correlated.

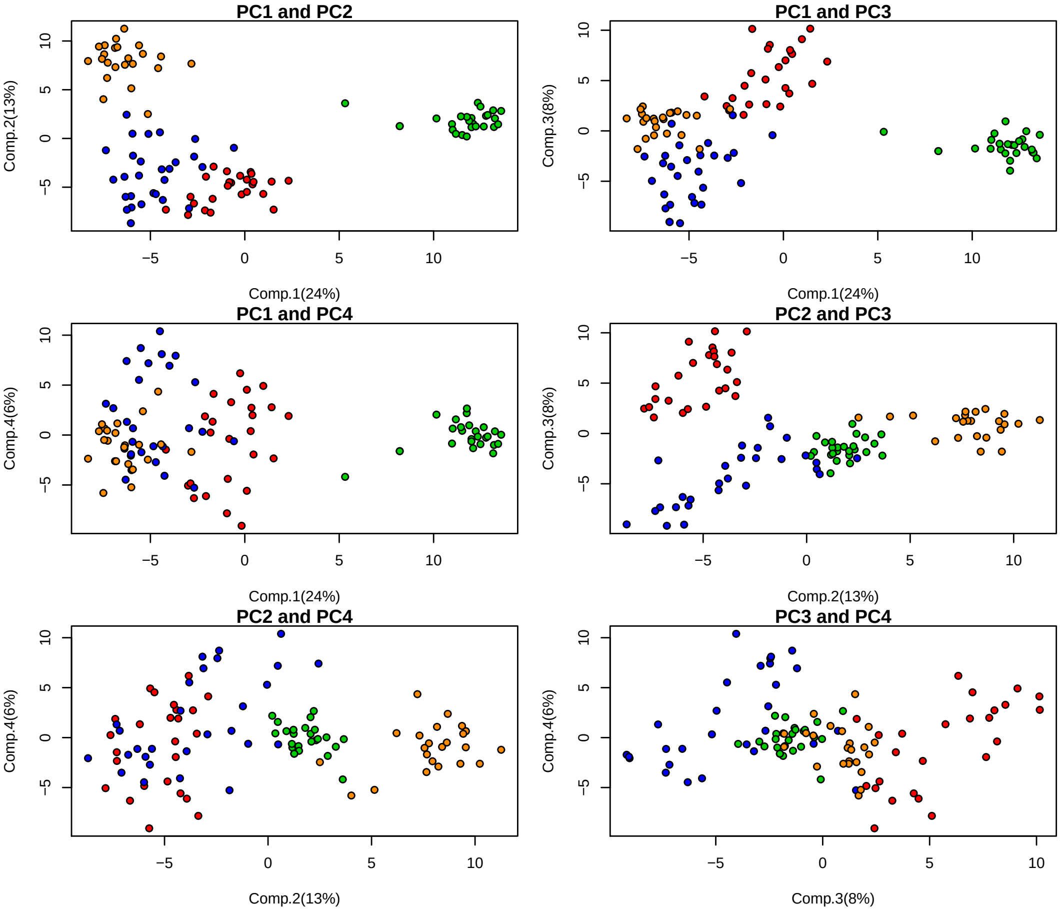

Figure 4.12: PCA applied to multiple tissue data with 92 genes selected by hierarchical clustering. PC3 separates breast samples (red) from lung samples (blue) which was not achieved by the other projections.

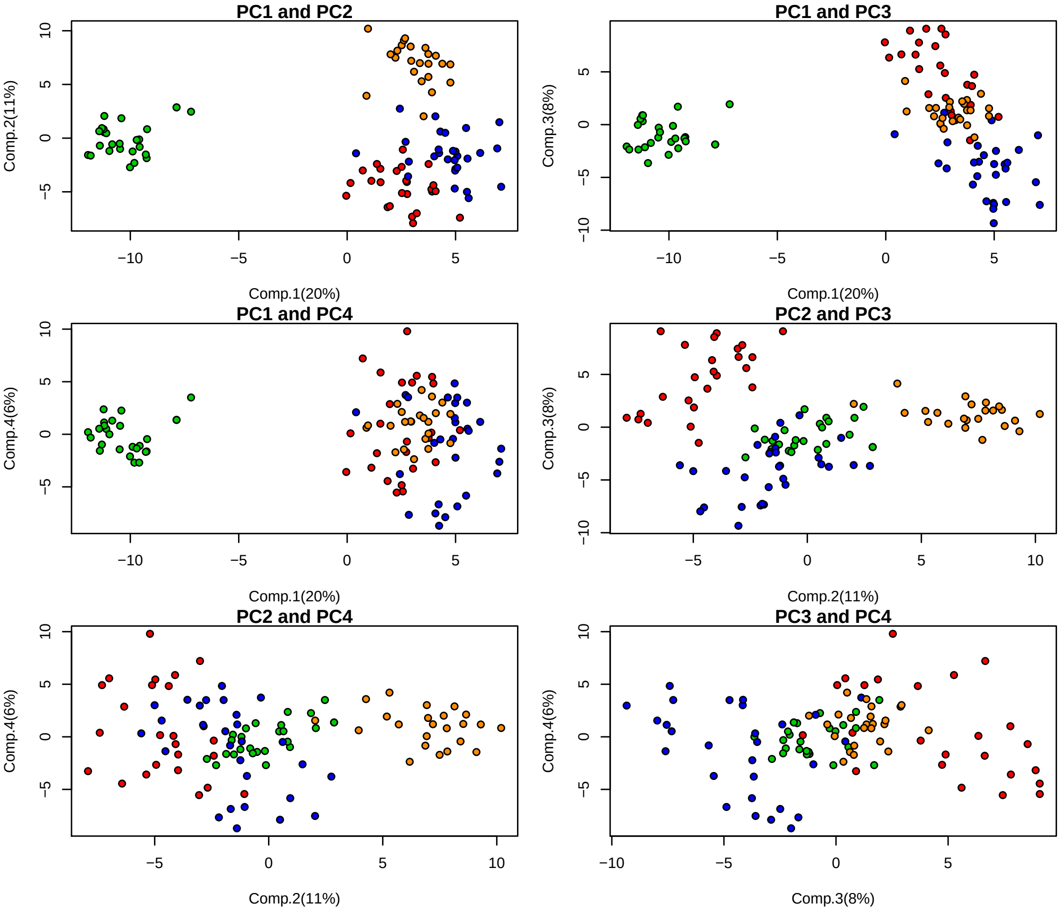

Figure 4.13: PCA applied to multiple tissue data with 95 genes selected by hierarchical clustering on the correlation coefficient matrix. PC3 separates breast samples (red) from lung samples (blue) almost as good as in previous example.

4.3 Clustering

One of the best known and most popular unsupervised learning techniques is clustering. “Clusters” in the data are regions where observations group together or, in other words, regions of high data density. Often these clusters are observations which stem from one “prototype” via noise perturbations. The prototype may represent a certain situation in the real world which is repeated but has slightly different environments or is measured with noise, so that, the feature values for this situation differ for each occurrence.

Clustering extracts structures in the data and can identify new data classes which were unknown so far. An important application of clustering is data visualization, where in some cases both down-projection and clustering are combined, e.g. as for self-organizing maps which were previously considered. If observations are represented by their prototypes then clustering is a data compression method called “vector quantization”.

4.3.1 k-Means Clustering

4.3.1.1 The Method

Probably the best known clustering algorithm is *k-means clustering* Forgy [1965], Hartigan [1972, 1975], Hartigan and Wong [1979]. k-means assumes k clusters but we denote the number of clusters by l to keep the notation that we used for other methods.

The only parameters are the cluster centers in contrast to mixture clustering which has as parameters cluster centers, cluster covariance, cluster weight, i.e. number of points in the cluster.

The cluster membership is determined very simple: x_i belongs to the cluster j with the closest center $\boldsymbol{\mu}_j$, i.e. the smallest Euclidean distance $\|\boldsymbol{x}_i \, - \, \boldsymbol{\mu}_j\|$ to $\boldsymbol{x}_i$.

$$c_{\boldsymbol{x}_i} \; = \; \arg\min_k \|\boldsymbol{x}_i \, - \, \boldsymbol{\mu}_k\| \; . \tag{4.40}$$

Only the centers are updated:

$$\boldsymbol{\mu}_j^{\mathrm{new}} \; = \; \frac{1}{n_j} \sum_{i=1, \; j=c_{\boldsymbol{x}_i}}^{n} \boldsymbol{x}_i \tag{4.41}$$

$$n_j \; = \; \sum_{i=1, \; j=c_{\boldsymbol{x}_i}}^{n} 1 \, , \tag{4.42}$$

where n_j is the number of data points assigned to cluster j. Therefore $\boldsymbol{\mu}_j^{\mathrm{new}}$ is the mean of the data points assigned to cluster j. The *k-means clustering* algorithm is given in Alg. 4.1.

The k-means clustering:

- fast,
- robust to outliers (covariance),
- simple (can be an advantage or a disadvantage),

Algorithm 4.1 k-means

```
  Given: data $\{\boldsymbol{x}\} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$, number of clusters $l$

BEGIN initialization
  initialize the cluster centers $\boldsymbol{\mu}_j$, $1 \leq j \leq l$
END initialization

BEGIN Iteration

  Stop=false
  while Stop=false do
    for $(i = 1 \; ; \; i \; \geq \; n \; ; \; i + +)$ do
      assign $\boldsymbol{x}_i$ to the nearest $\boldsymbol{\mu}_j$
    end for
    for $(j = 1 \; ; \; j \; \geq \; l \; ; \; j + +)$ do

      $\boldsymbol{\mu}_j^{\text{new}} = \frac{1}{n_j} \sum_{i=1,\; j=c_{\boldsymbol{x}_i}}^{n} \boldsymbol{x}_i$

    end for
    if stop criterion fulfilled then
      Stop=true
    end if
  end while
END Iteration
```

- prone to the initialization.

For example, consider an initialization which places one center near several outliers which are separated from the rest of the data points. The other data points have other cluster centers closer to them. Then this outlier cluster will remain in each iteration at the outliers even if other cluster are not modeled. This behavior can be serious in high dimensions.

We define the softmax membership with parameter b as

$$w_j(\boldsymbol{x}_i) \;=\; \frac{\|\boldsymbol{x}_i \;-\; \boldsymbol{\mu}_j\|^{-2/(b-1)}}{\sum_{k=1}^{l} \|\boldsymbol{x}_i \;-\; \boldsymbol{\mu}_k\|^{-2/(b-1)}} \;. \tag{4.43}$$

and obtain as update rule

$$\boldsymbol{\mu}_j^{\text{new}} \;=\; \frac{\sum_{i=1}^{n} w_j(\boldsymbol{x}_i)\; \boldsymbol{x}_i}{\sum_{i=1}^{n} w_j(\boldsymbol{x}_i)} \;. \tag{4.44}$$

This algorithm is called *fuzzy k-means clustering* and described in Alg. 4.2.

Algorithm 4.2 Fuzzy k-means

```
Given: data $\{\boldsymbol{x}\} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$, number of clusters $l$, parameter $b$

BEGIN initialization
  initialize the cluster centers $\boldsymbol{\mu}_j$, $1 \leq j \leq l$, and $w_j(\boldsymbol{x}_i)$ so that $\sum_{j=1}^{l} w_j(\boldsymbol{x}_i) = 1$, $w_j(\boldsymbol{x}_i) \geq 0$.
END initialization

BEGIN Iteration

  Stop=false
  while Stop=false do

      $\boldsymbol{\mu}_j^{\text{new}} = \frac{\sum_{i=1}^{n} w_j(\boldsymbol{x}_i)\, \boldsymbol{x}_i}{\sum_{i=1}^{n} w_j(\boldsymbol{x}_i)}$

      $w_j(\boldsymbol{x}_i) = \frac{\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^{-2/(b-1)}}{\sum_{k=1}^{l} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^{-2/(b-1)}}$

    if stop criterion fulfilled then
      Stop=true
    end if
  end while
END Iteration
```

4.3.1.2 Examples

We demonstrate k-means on an artificial data set in two dimensions with five clusters. The five cluster data set was generated as follows:

```
x <- rbind(
      matrix(rnorm(100, sd = 0.2), ncol = 2),
      matrix(rnorm(100, mean = 1, sd = 0.2), ncol = 2),
      matrix(rnorm(100, mean = -1, sd = 0.2), ncol = 2),
      matrix(c(rnorm(100, mean = 1, sd = 0.2),
          rnorm(100, mean = -1, sd = 0.2)), ncol = 2),
      matrix(c(rnorm(100, mean = -1, sd = 0.2),
          rnorm(100, mean = 1, sd = 0.2)), ncol = 2))
colnames(x) <- c("x", "y")
k=5
```

Fig. 4.14 shows the result of k-means with $k = 5$ where an optimal solution is found. Filled circles mark the cluster centers. Local minima are shown in Fig. 4.15 and Fig. 4.16. In both cases one cluster explains two true clusters while one true cluster is divided into two model clusters. Fig. 4.17 shows a local minimum, where three model clusters share one true cluster.

We apply k-means with $k = 8$ to the five cluster data set. In this case the number of model clusters does not match the number of true clusters. Therefore the solution will always be worse than the optimal solution with the correct number of clusters. Fig. 4.18 shows a solution where three true clusters are shared by pairs of model clusters. Fig. 4.19 shows a solution where one true cluster is shared by 3 model clusters and another by 2 model clusters. This solution is very typical and another example is presented in Fig. 4.20. Fig. 4.21 shows a solution where a true cluster is shared by four model clusters. The remaining true clusters are correctly explained by one model cluster.

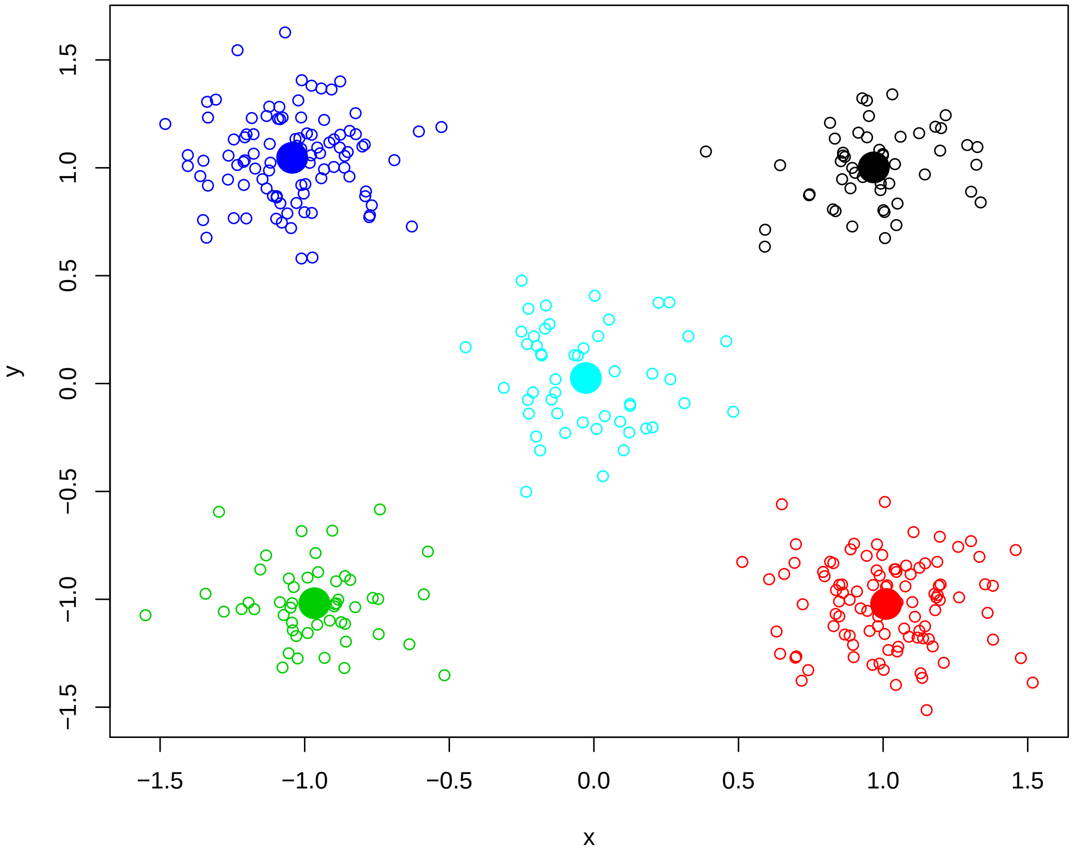

Figure 4.14: k-means clustering of the five cluster data set with $k = 5$ where filled circles mark the cluster centers. An optimal solution is found.

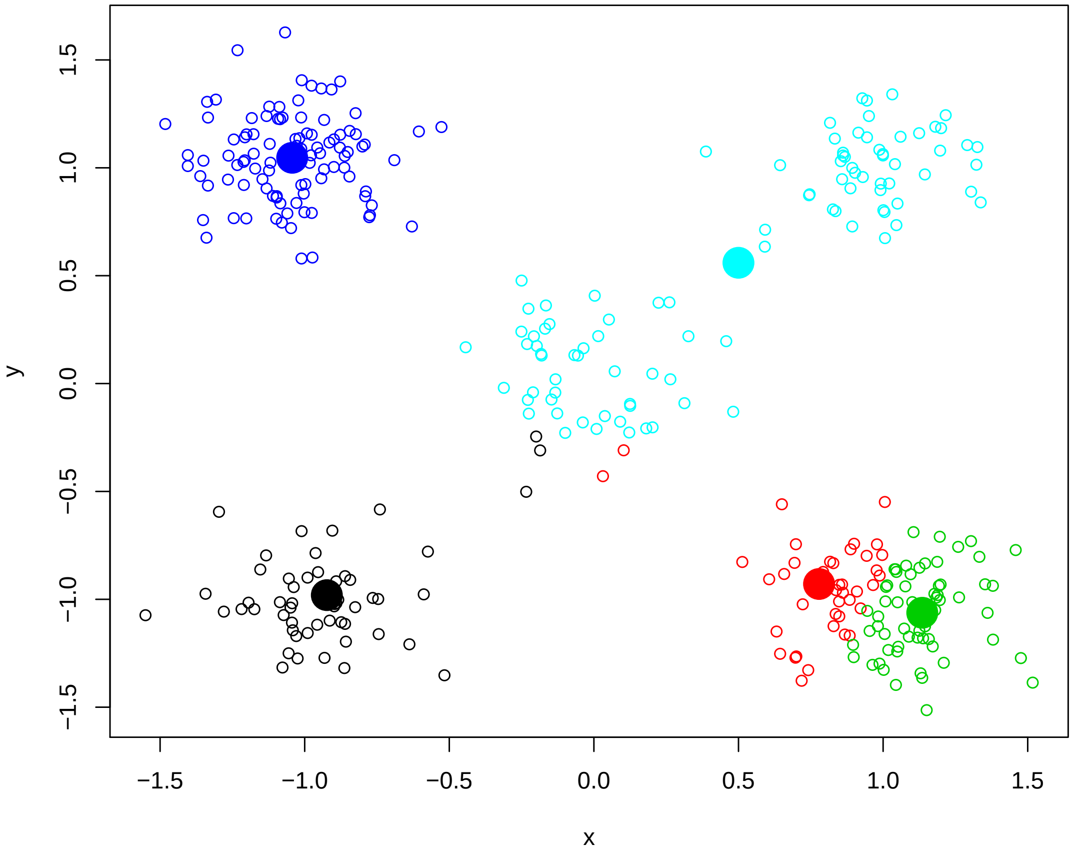

Figure 4.15: k-means clustering of the five cluster data set with $k = 5$ where filled circles mark the cluster centers. A local minimum is found.

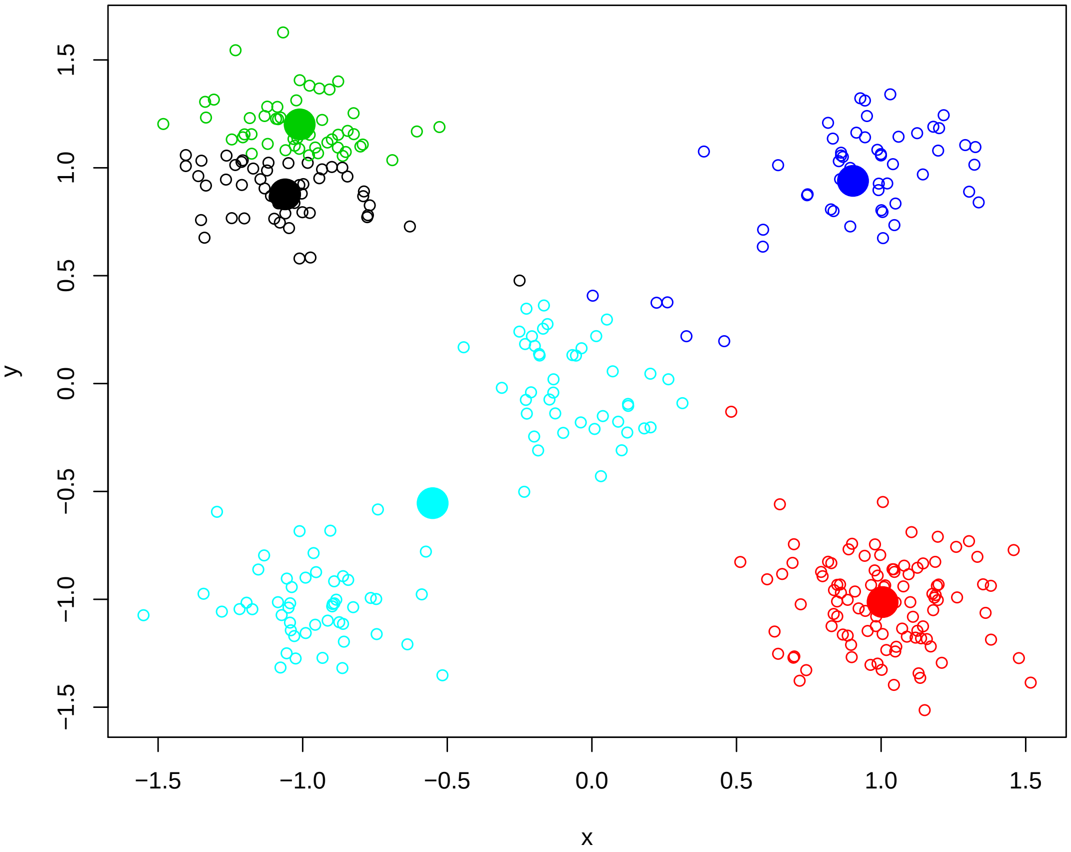

Figure 4.16: k-means clustering of the five cluster data set with $k = 5$ where filled circles mark the cluster centers. A local minimum is found.

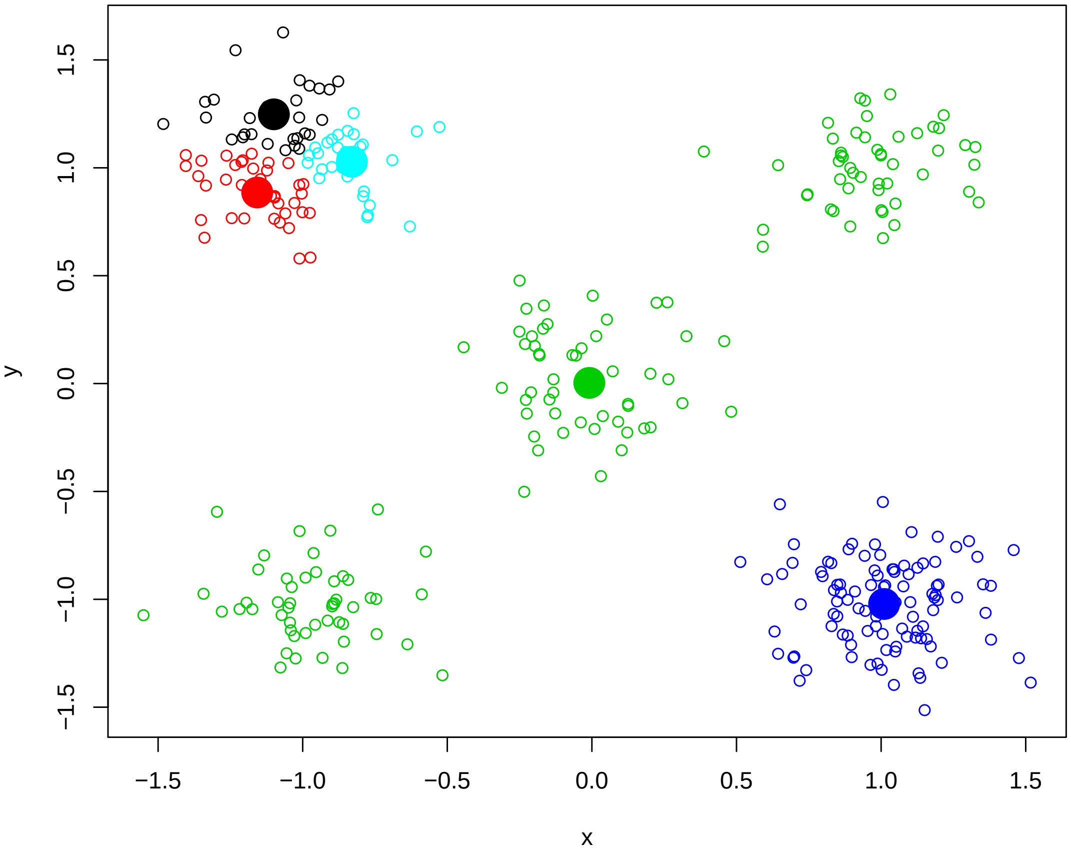

Figure 4.17: k-means clustering of the five cluster data set with $k = 5$ where filled circles mark the cluster centers. A local minimum is found where three model cluster share one true cluster.

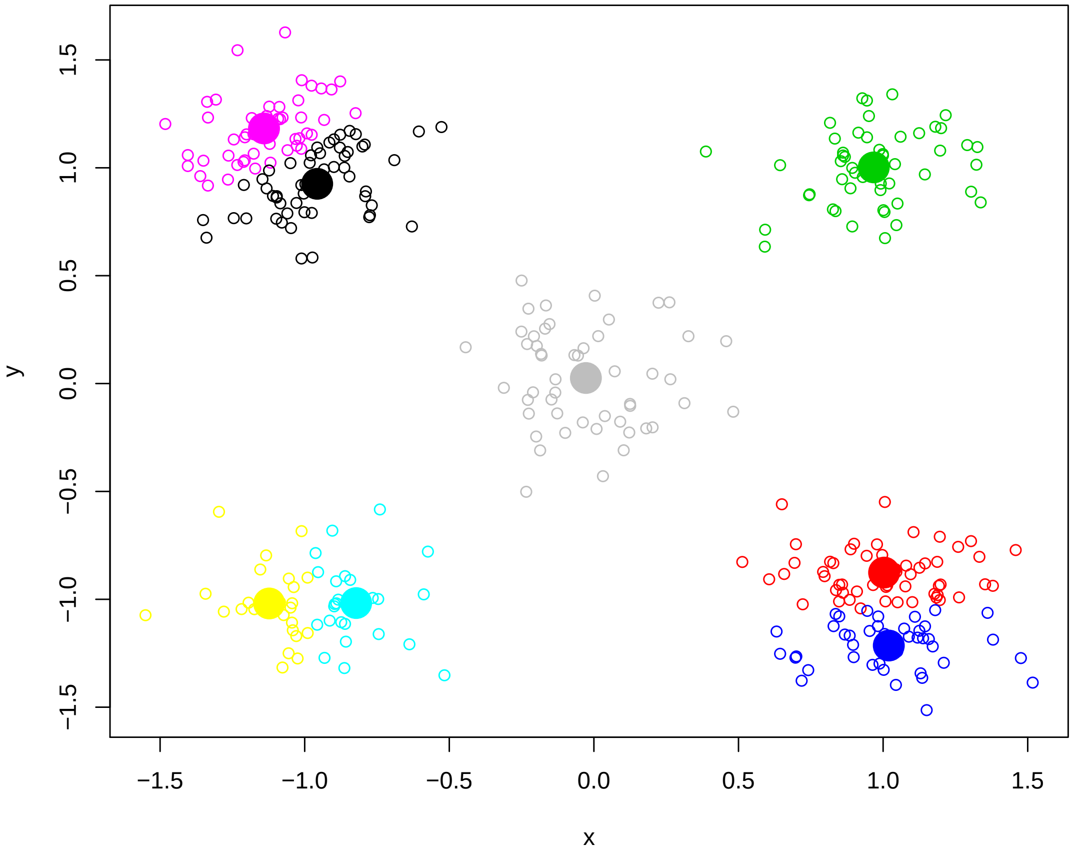

Figure 4.18: k-means clustering of the five cluster data set with $k = 8$ where filled circles mark the cluster centers. True clusters are shared by pairs of model clusters.

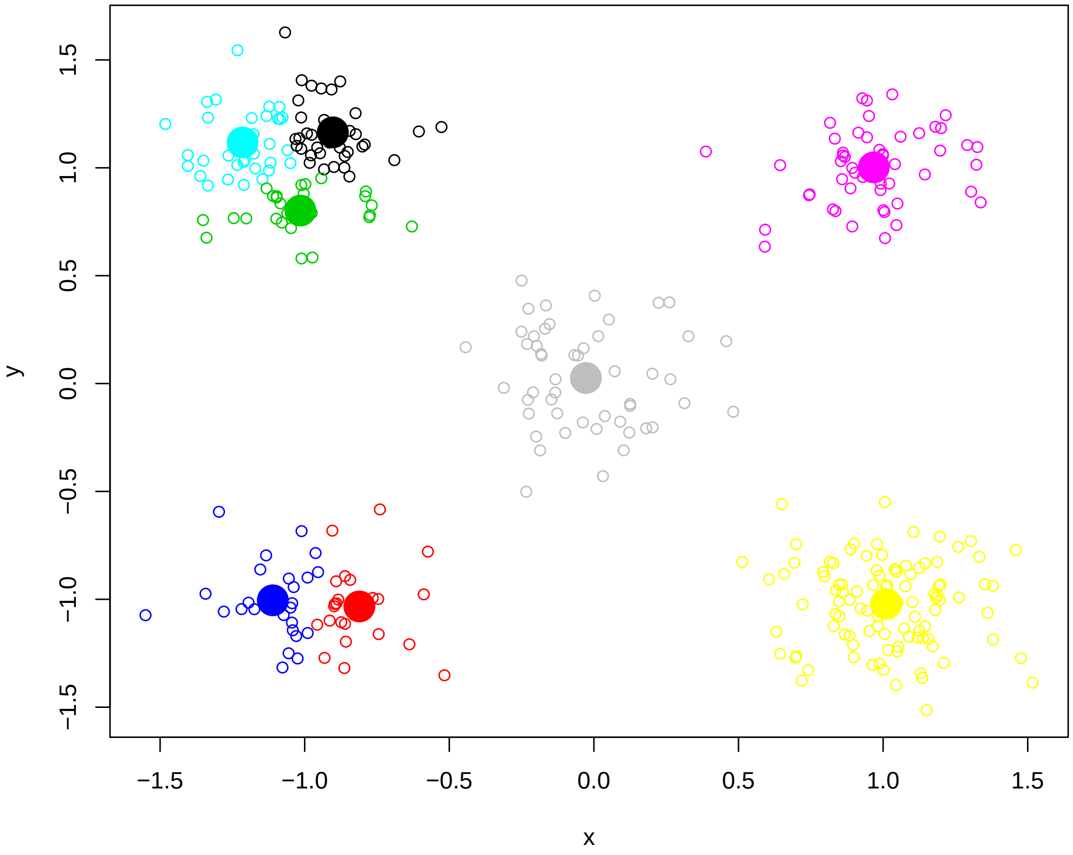

Figure 4.19: k-means clustering of the five cluster data set with $k = 8$. Typical case where one true cluster is shared by 3 model clusters and another by 2 model clusters.

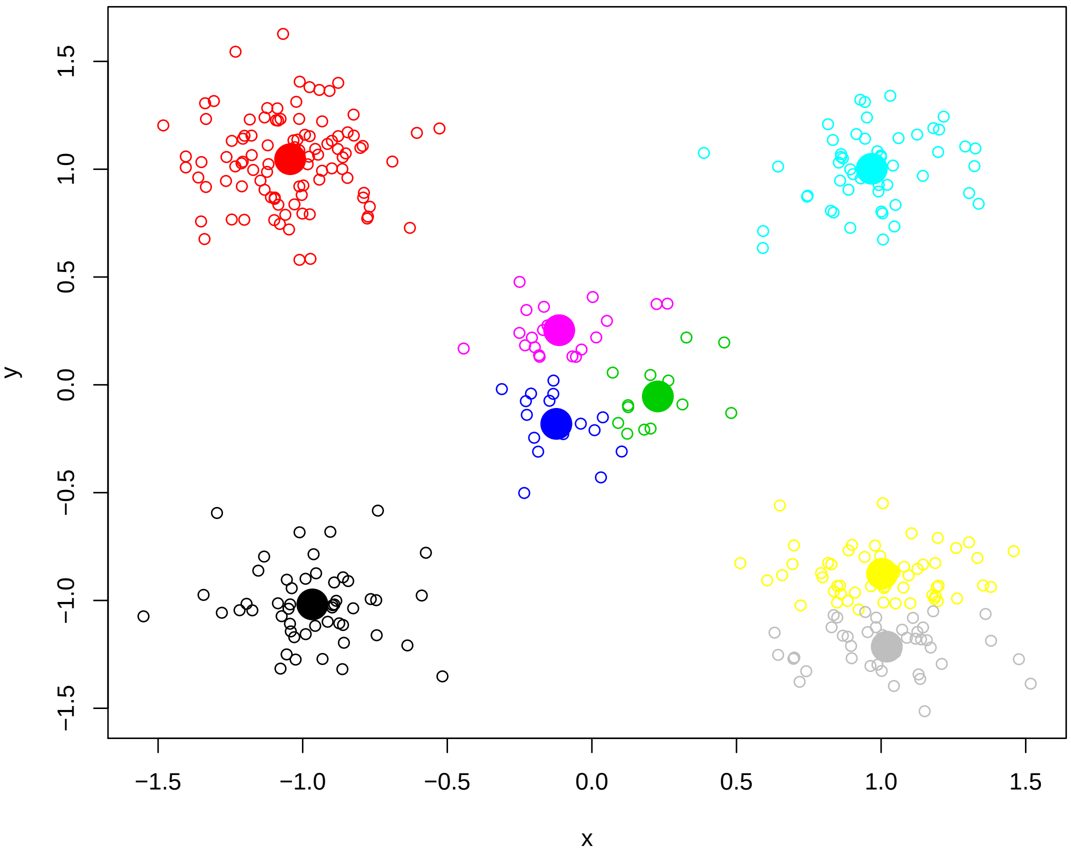

Figure 4.20: k-means clustering of the five cluster data set with $k = 8$. Another example of the situation like Fig. 4.19.

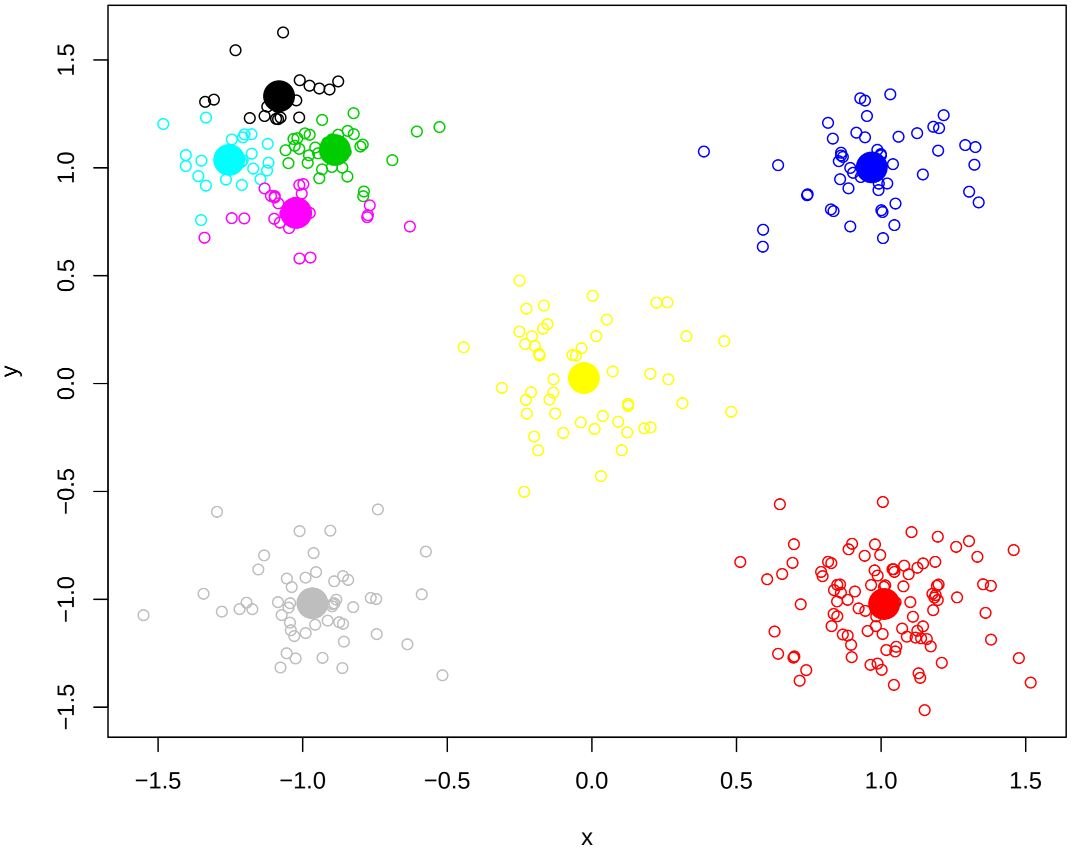

Figure 4.21: k-means clustering of the five cluster data set with $k = 8$. A true cluster is shared by four model clusters.

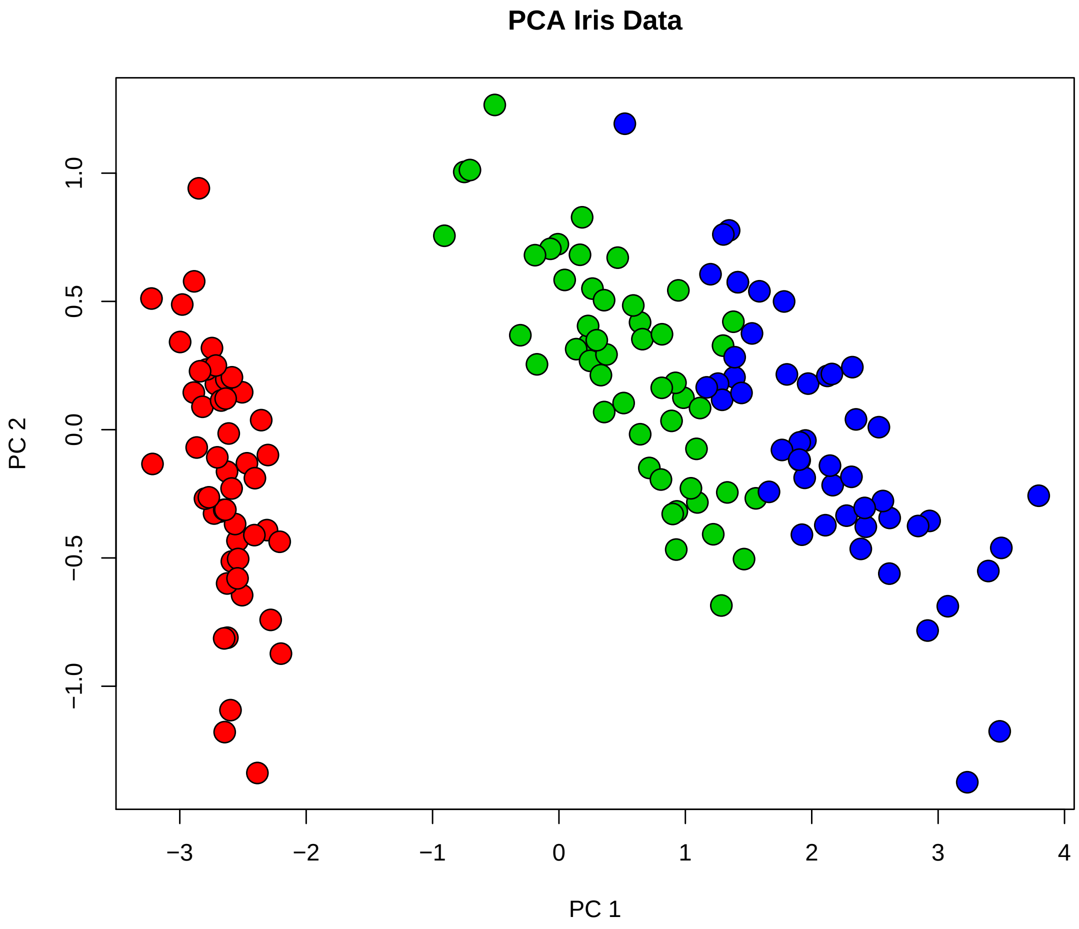

Figure 4.22: Down-projection of the iris data set to two dimensions by PCA. The true classes are marked by colors.

We apply k-means to the Iris data set. To remind the reader, the down-projection onto two dimensions by PCA is again given in Fig. 4.22, where the true classes are marked by colors. Fig. 4.23 shows a typical solution of k-means applied to the Iris data set. The solution is quite good, only at the border assignments are made wrong. Fig. 4.24 gives another typical solution of k-means for the Iris data set. This solution is not good as two components share a cluster. Important question is whether the quality of these solutions can be distinguished if the true classes are not known.

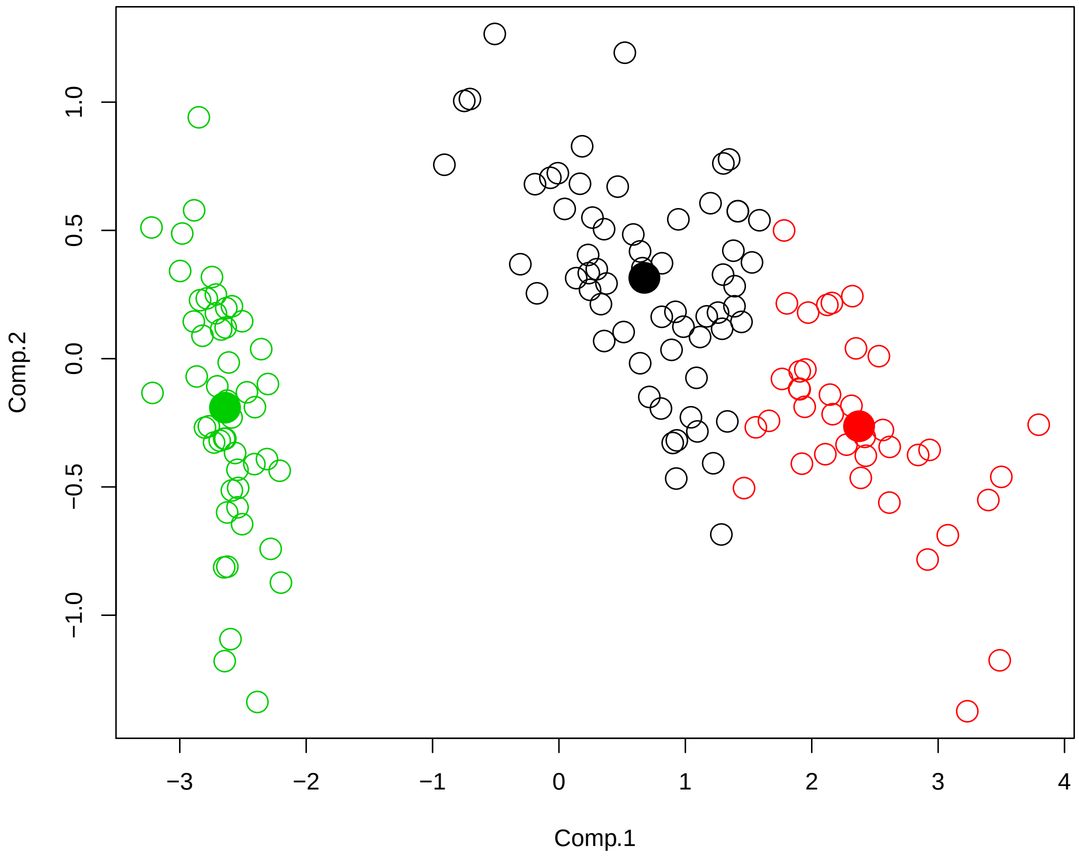

Figure 4.23: k-means clustering of the Iris data set. The **first** typical solution where filled circles are cluster centers. The solution is quite good, only at the border assignments are made wrong.

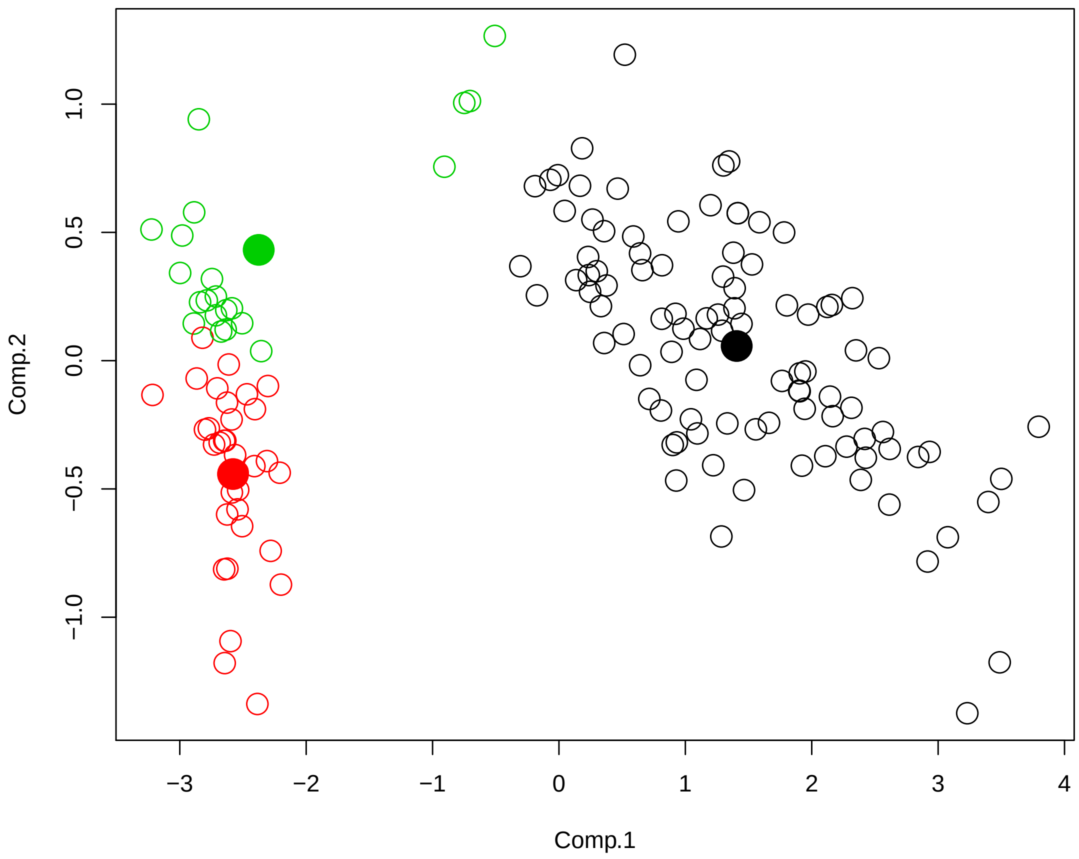

Figure 4.24: k-means clustering of the Iris data set. The **second** typical solution where filled circles are cluster centers. This solution is not good as two components share a cluster.

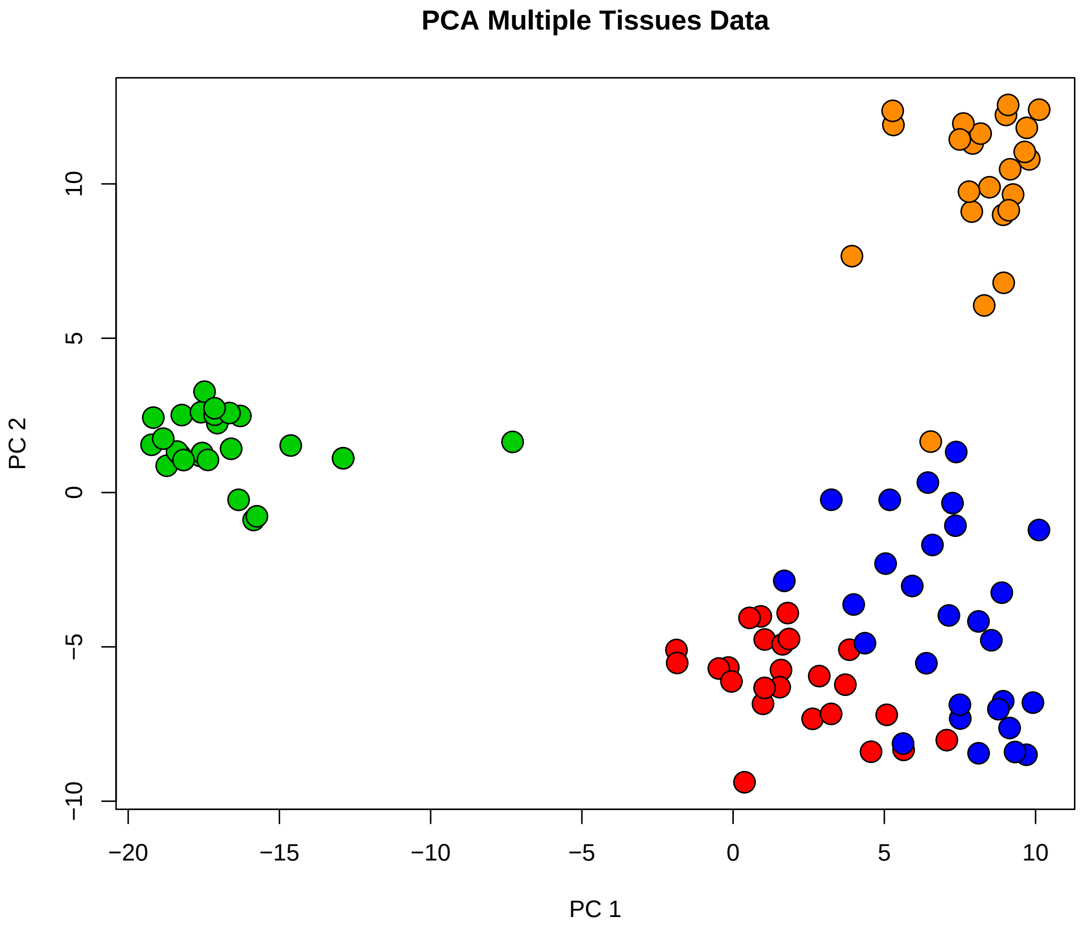

Figure 4.25: Down-projection of the multiple tissue data set to two dimensions by PCA. The true classes are marked by colors.

We apply k-means to the multiple tissues data set. To remind the reader, the down-projection onto two dimensions by PCA is again given in Fig. 4.25, where the true classes are marked by colors. For the down-projection the 101 features with the largest variance are used. k-means is applied to the full data set. Fig. 4.26 shows the typical solution of k-means applied to the multiple tissues data set. This solution appears in almost all cases. The classes are almost perfectly identified. Fig. 4.27 gives another solution of k-means for the multiple tissues data set. This solution is not as good as the solution in Fig. 4.26. Another suboptimal solution is shown in Fig. 4.28.

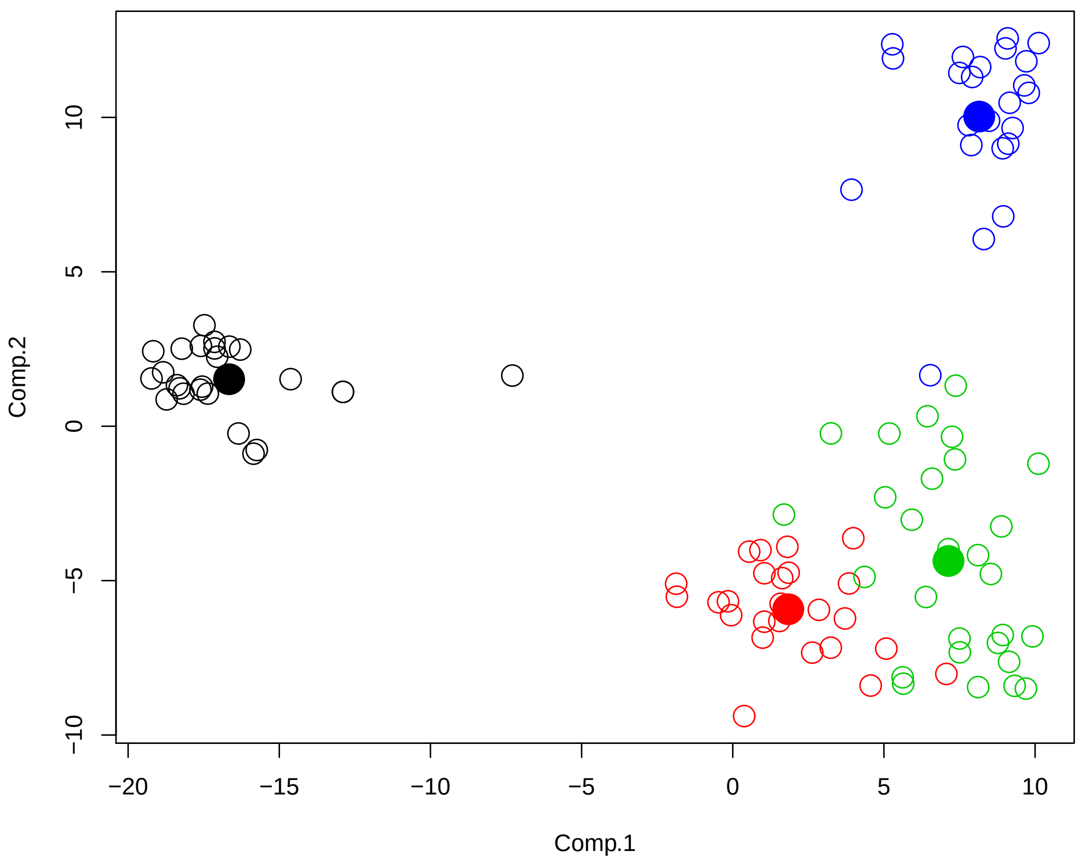

Figure 4.26: k-means clustering of the multiple tissue data set with $k = 4$. Filled circles are cluster centers. This is the solution found in almost all initializations. The classes are almost perfectly identified.

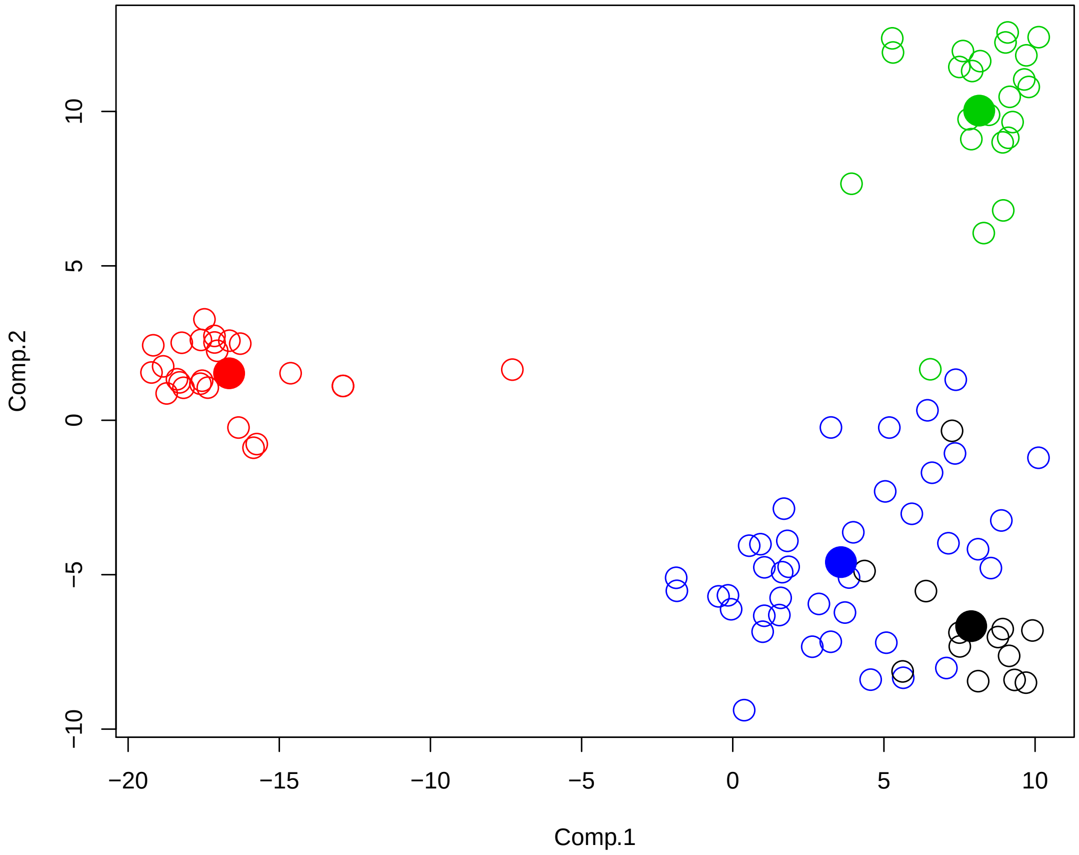

Figure 4.27: k-means clustering of the Iris data set with $k = 4$. This solution is not as good as the typical solution from Fig. 4.26.

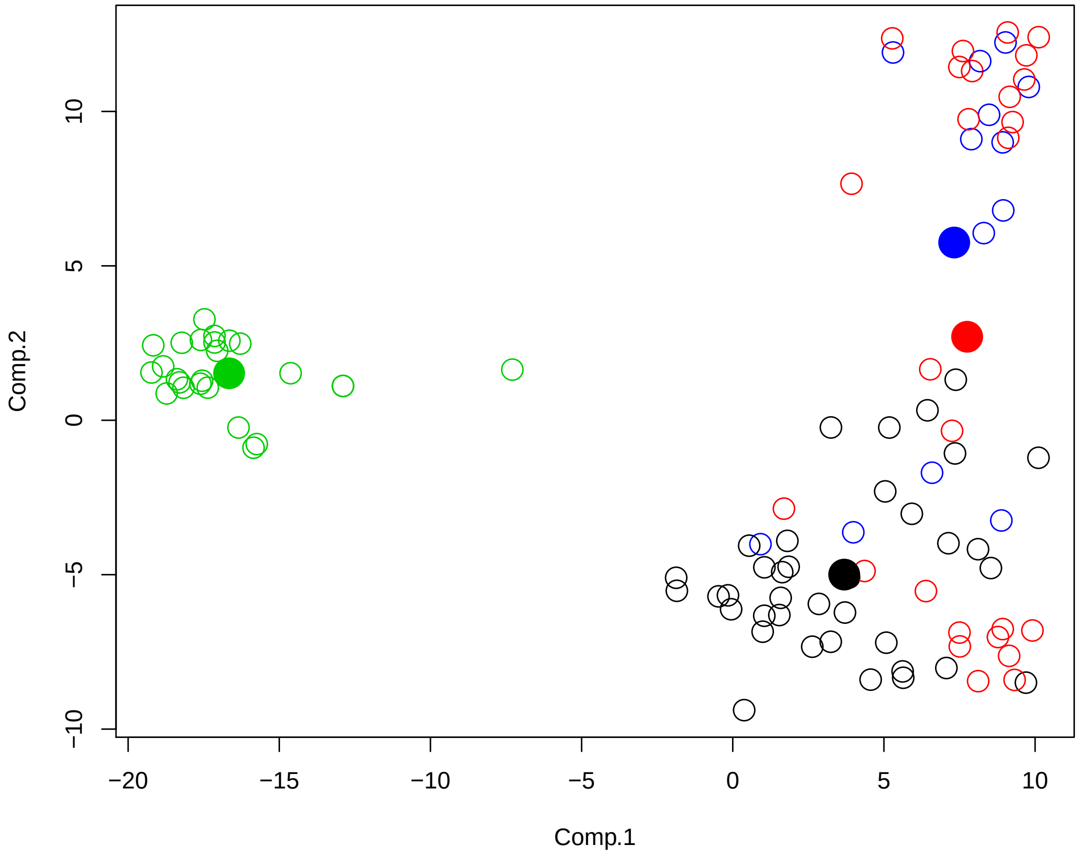

Figure 4.28: k-means clustering of the Iris data set with $k = 4$. Again, this solution is not as good as the typical solution from Fig. 4.26.

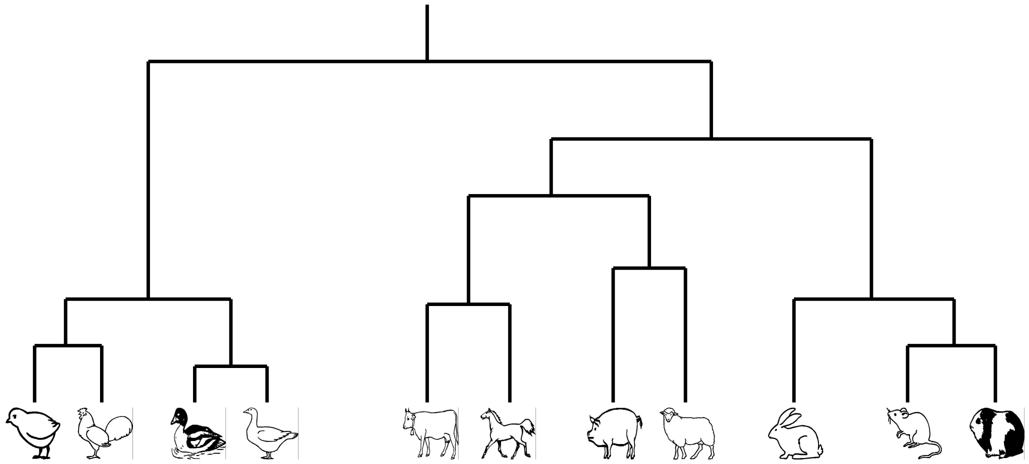

Figure 4.29: Example of hierarchical clustering of animal species where the result is given as a dendrogram (corresponding tree).

4.3.2 Hierarchical Clustering

4.3.2.1 The Method

So far we did not consider distances and structures between the clusters. Distances between clusters help to evaluate the clustering result and single clusters. In particular it would help to decide whether clusters should be merged or not. *Hierarchical clustering* supplies distances between clusters which are captured in a dendrogram. Fig. 4.29 depicts a dendrogram as the result of hierarchical clustering. Hierarchical clustering can be performed

- *agglomerative*, that is, *bottom up*, where the clustering starts with all clusters having a single observations and then clusters are merged until only one cluster remains
- *divisive*, that is, *top down*, where the clustering starts with one cluster and clusters are split until only clusters with a single observation remain.

In Bioinformatics the method "Unweighted Pair Group Method using arithmetic Averages" (UPGMA) applies hierarchical clustering in order to construct a phylogenetic tree. In machine learning the UPGMA method is called *agglomerative hierarchical clustering*, where the closest clusters are merged to give a new cluster. Agglomerative hierarchical clustering is initialized by clusters that consist of a single observation. Then clusters are iteratively merged until only one cluster remains.

Agglomerative hierarchical clustering can be used with different distance measures between clusters A and B:

$$\begin{array}{rclr} d_{\mathrm{min}}(A, B) & = & \min_{\boldsymbol{a} \in A, \boldsymbol{b} \in B} \|\boldsymbol{a} \ - \ \boldsymbol{b}\| & \text{(single linkage)} \\ d_{\mathrm{max}}(A, B) & = & \max_{\boldsymbol{a} \in A, \boldsymbol{b} \in B} \|\boldsymbol{a} \ - \ \boldsymbol{b}\| & \text{(complete linkage)} \\ d_{\mathrm{avg}}(A, B) & = & \frac{1}{n_A \ n_B} \sum_{\boldsymbol{a} \in A} \sum_{\boldsymbol{b} \in B} \|\boldsymbol{a} \ - \ \boldsymbol{b}\| & \text{(average linkage)} \\ d_{\mathrm{mean}}(A, B) & = & \|\bar{\boldsymbol{a}} \ - \ \bar{\boldsymbol{b}}\| & \text{(average linkage)} \end{array} , \tag{4.45}$$

where n_A (n_B) is the number of elements in A (B) and $\bar{\boldsymbol{a}}$ ($\bar{\boldsymbol{b}}$) is the mean of cluster A (B). For the element distance $\|.\|$ any distance measure is possible like the Euclidean distance, the Manhattan distance, or the Mahalanobis distance.

For clusters with a single element these distance measures are equivalent, however for clusters with more than one element there is a difference.

- complete linkage $d_{\max}$ avoids that clusters are elongated in some direction, that is, the smallest distance between points may remains small. This means that the cluster may not be well separated.

- single linkage $d_{\min}$ ensures that each pair of elements, where one is from one cluster and the other is from another cluster, has a minimal distance. The result of single linkage guarantees that after a cut of the hierarchical clustering tree, the distance between clusters has a minimal value. In machine learning single linkage clustering is relevant for *leave-one-cluster-out* cross-validation. Leave-one-cluster-out cross-validation assumes that a whole new group of objects is unknown and left out. Therefore in the training set there is no object that is similar to a test set object. Leave-one-cluster-out cross-validation is known from protein structure prediction.

- average linkage d_{avg} is the "Unweighted Pair Group Method using arithmetic Averages" (UPGMA) method.

Instead of starting with clusters containing a single object *bottom up* clustering can start *top down*, that is, starting with a single cluster containing all objects. Such *divisive* or top down clustering methods are based on graph theoretic considerations. First the minimal spanning tree is built. Then the largest edge is removed which gives two clusters. Now the second largest edge can be removed and so on. It might be more appropriate to compute the average edge length within a cluster and find the edge which is considerably larger than other edges in the cluster. This means long edges are selected locally as an edge that does not fit to the cluster structure and not globally. At node level, the edge of each node can be determined which is considerably larger than other edges of this node. The inconsistent (considerably larger) edges can be removed stepwise and new clusters are produced.

4.3.2.2 Examples

We perform hierarchical clustering on the US Arrest data set from Subsection 2.6. We test the distance measures "ward", "single", "complete", "average", "mcquitty", "median", and "centroid". For hierarchical clustering we use the R function `hclust` and plot the results by the following commands (shown for "ward"):

```
hc <- hclust(dist(USArrests), method="ward")
opar <- par(cex=0.7,font=2)
plot(hc, hang = -1,main=paste("Hierarchical Clustering US Arrests: ",
+ "ward",sep=""))
par(opar)
```

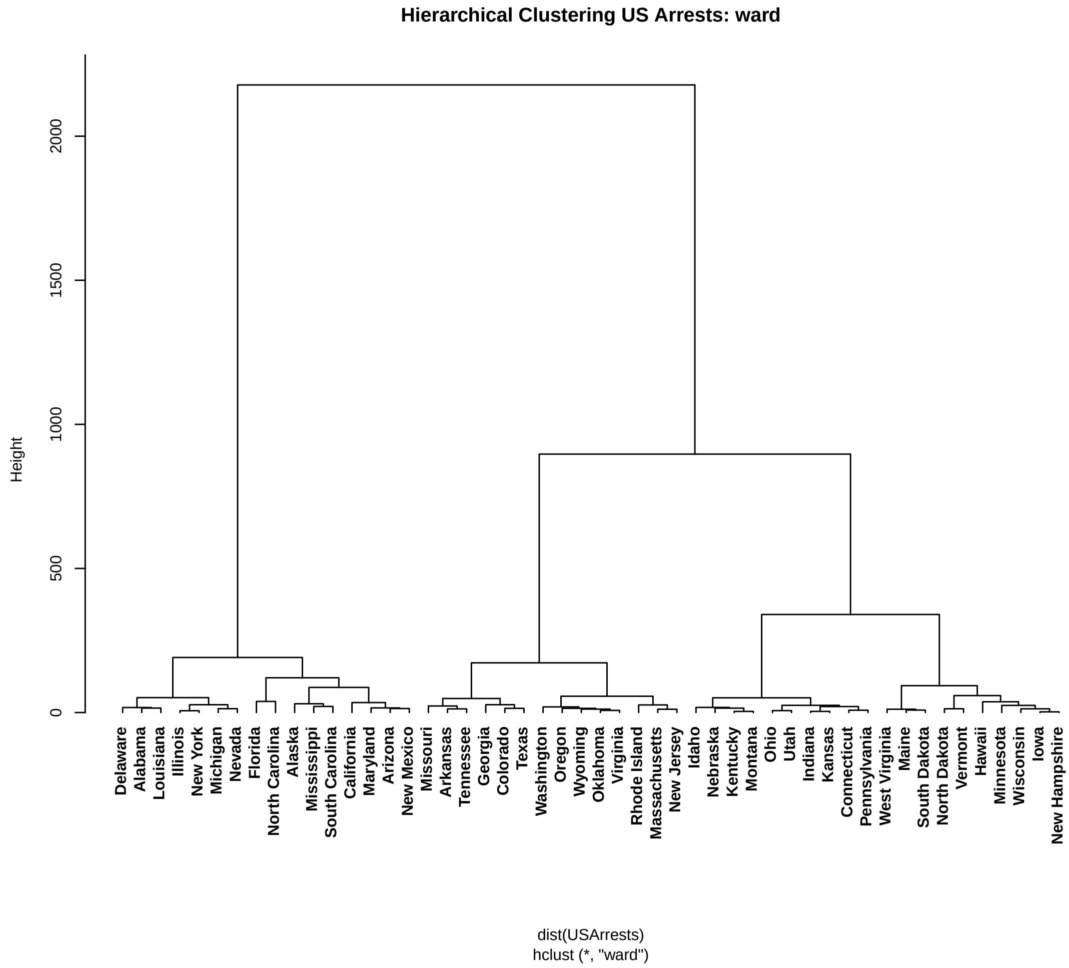

Figure 4.30: Hierarchical clustering of the US Arrest data set using Ward's minimal variance which gives compact, spherical clusters.

Fig. 4.30 shows the results agglomerative hierarchical clustering using Ward's minimal variance as distance which gives compact, spherical clusters. Fig. 4.31 shows the results for single linkage which gives similar clusters with a minimal distance. Fig. 4.32 shows the results for complete linkage (minimal spanning tree) which is a "friends of friends" clustering. Fig. 4.33 shows the results for average linkage, which corresponds to UPGMA in bioinformatics (distance between averages of cluster elements). Fig. 4.34 shows the results for the McQuitty distance. Fig. 4.35 shows the results for median distance which is not a monotone distance measure. Fig. 4.36 shows the results for centroid distance which is also not a monotone distance measure.

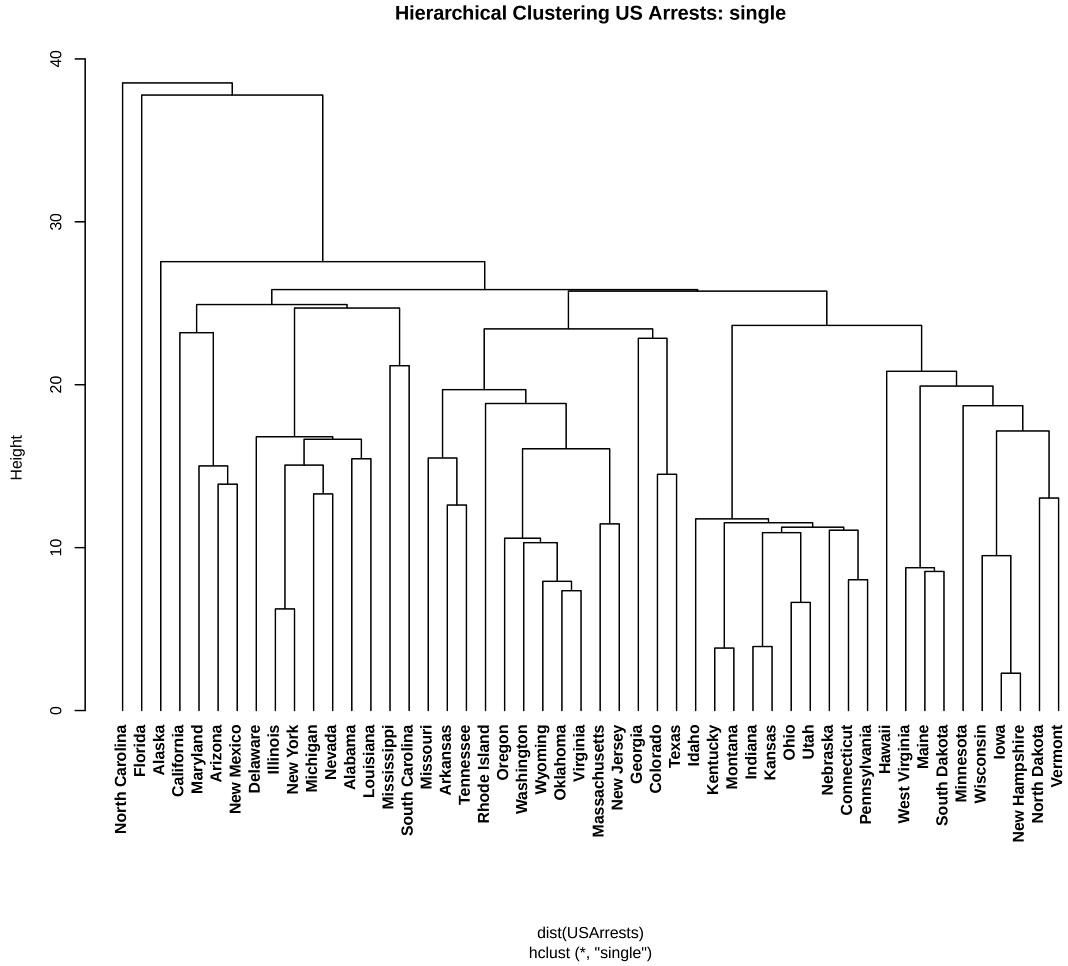

Figure 4.31: Hierarchical clustering of the US Arrest data set using single linkage which gives similar clusters with a minimal distance.

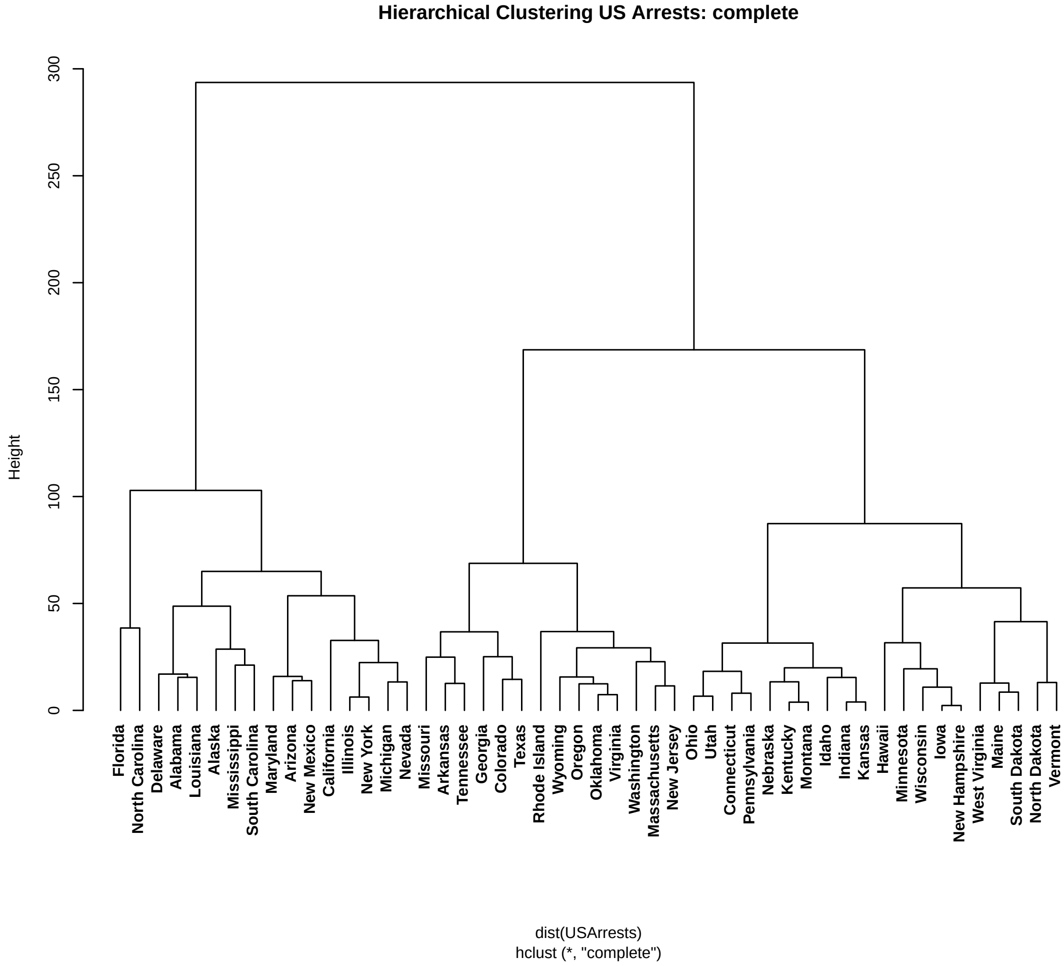

Figure 4.32: Hierarchical clustering of the US Arrest data set using complete linkage (minimal spanning tree) which is a “friends of friends” clustering.

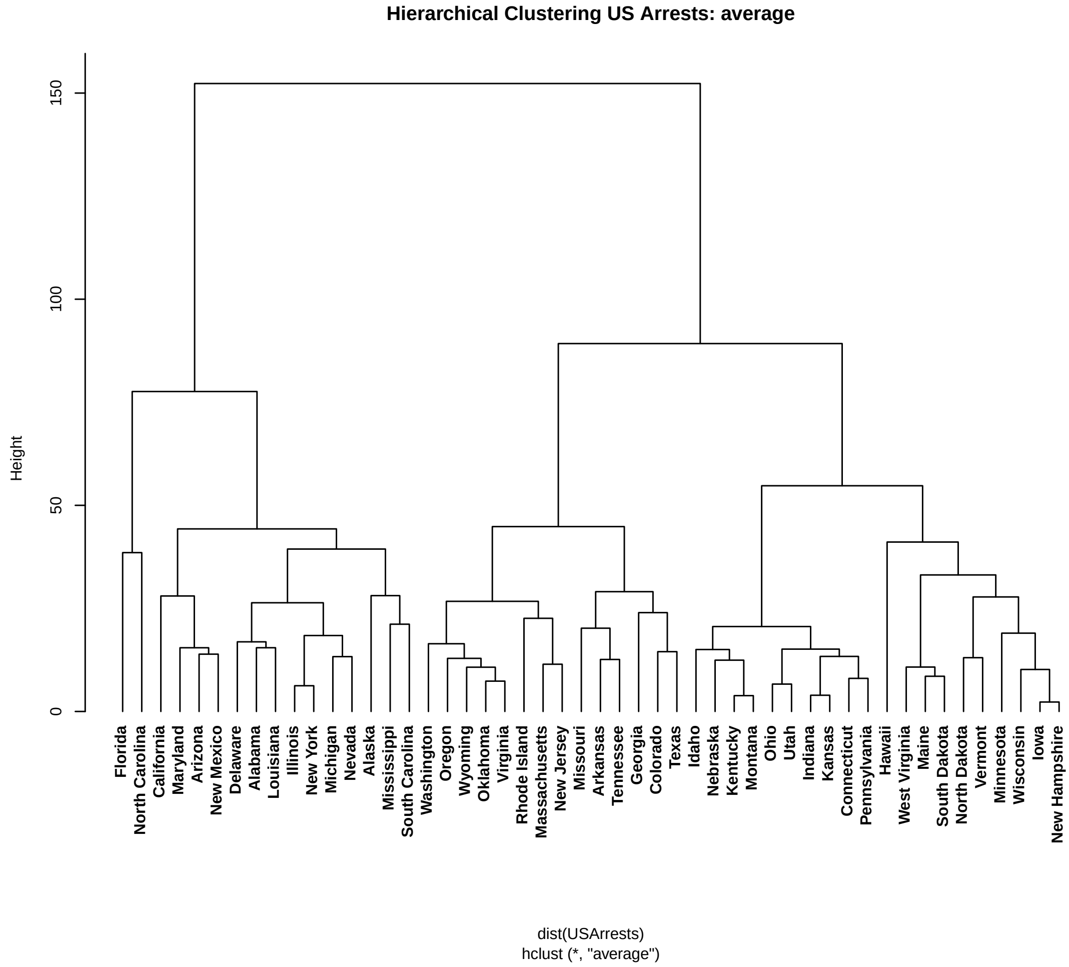

Figure 4.33: Hierarchical clustering of the US Arrest data set using average linkage.

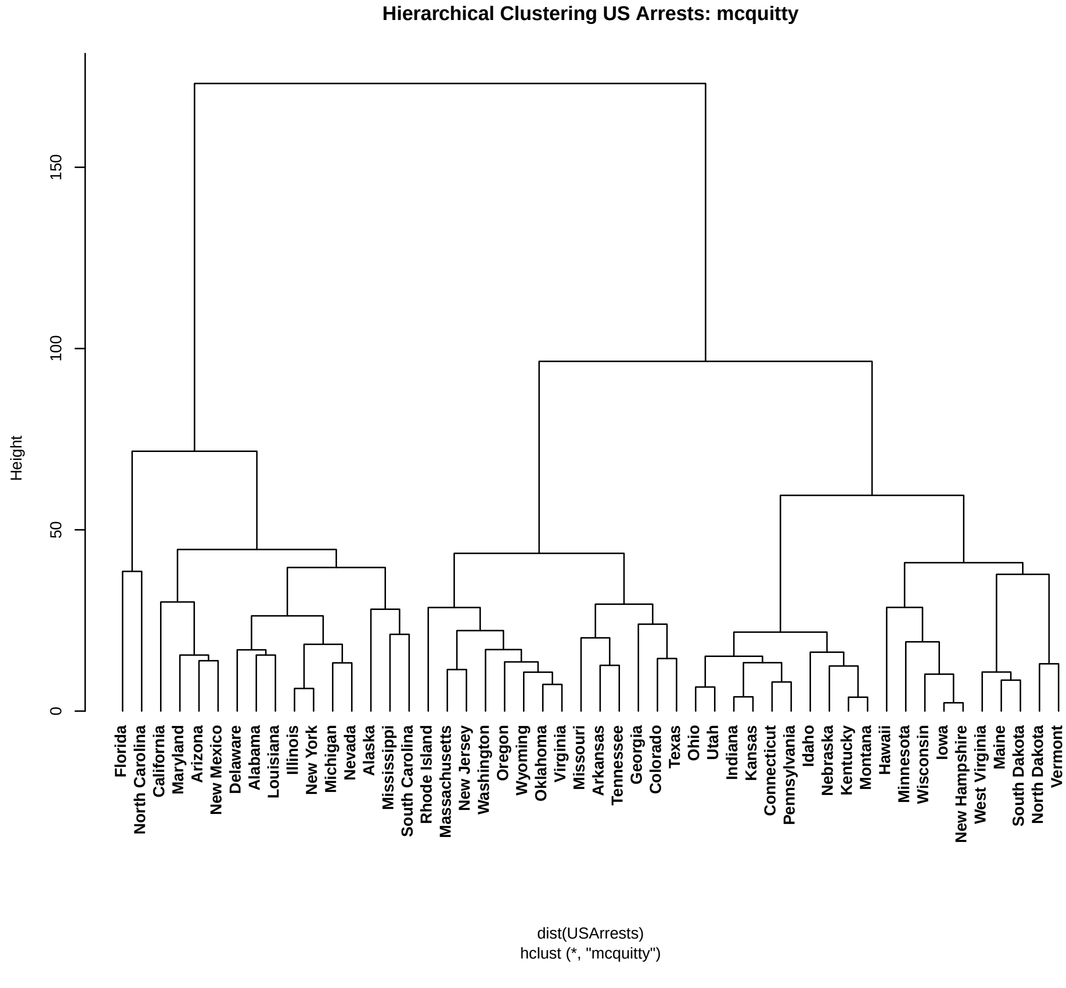

Figure 4.34: Hierarchical clustering of the US Arrest data set using McQuitty's distance.

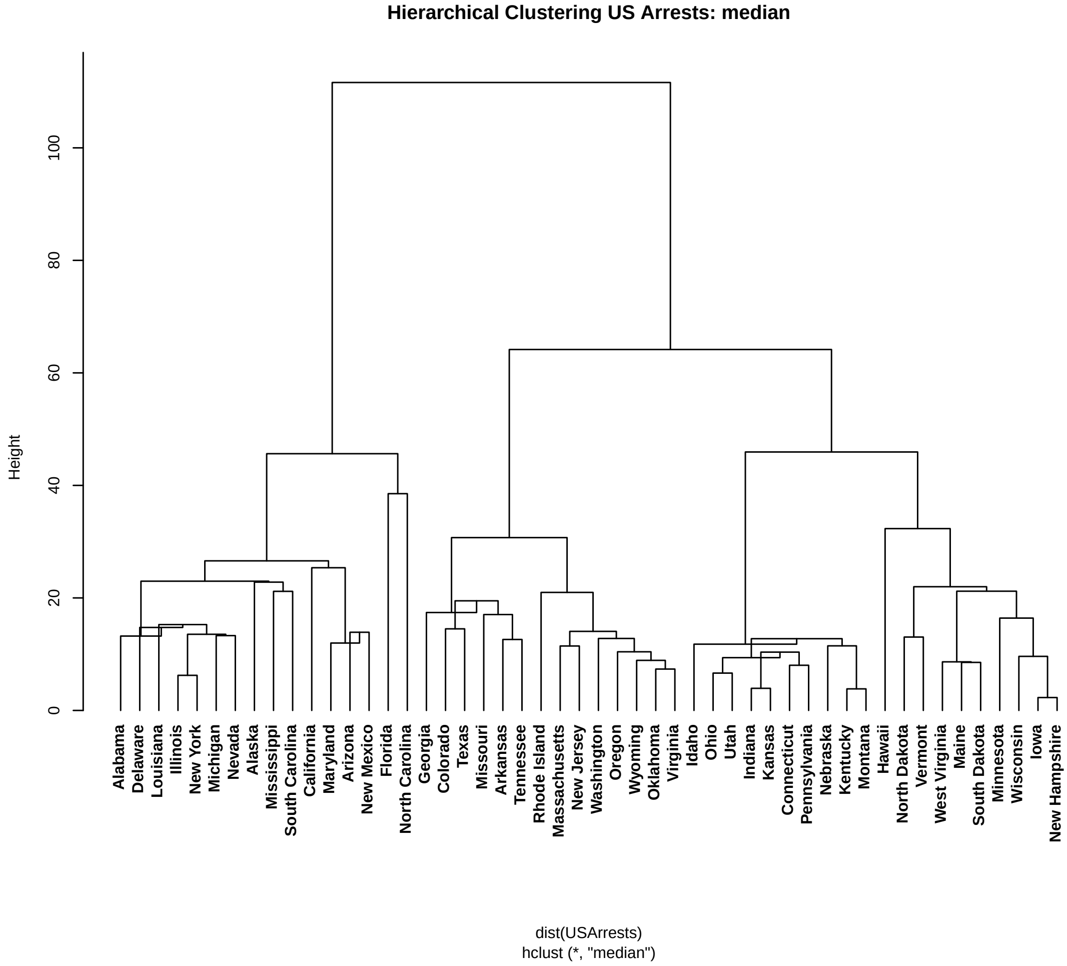

Figure 4.35: Hierarchical clustering of the US Arrest data set using median distance (not monotone).

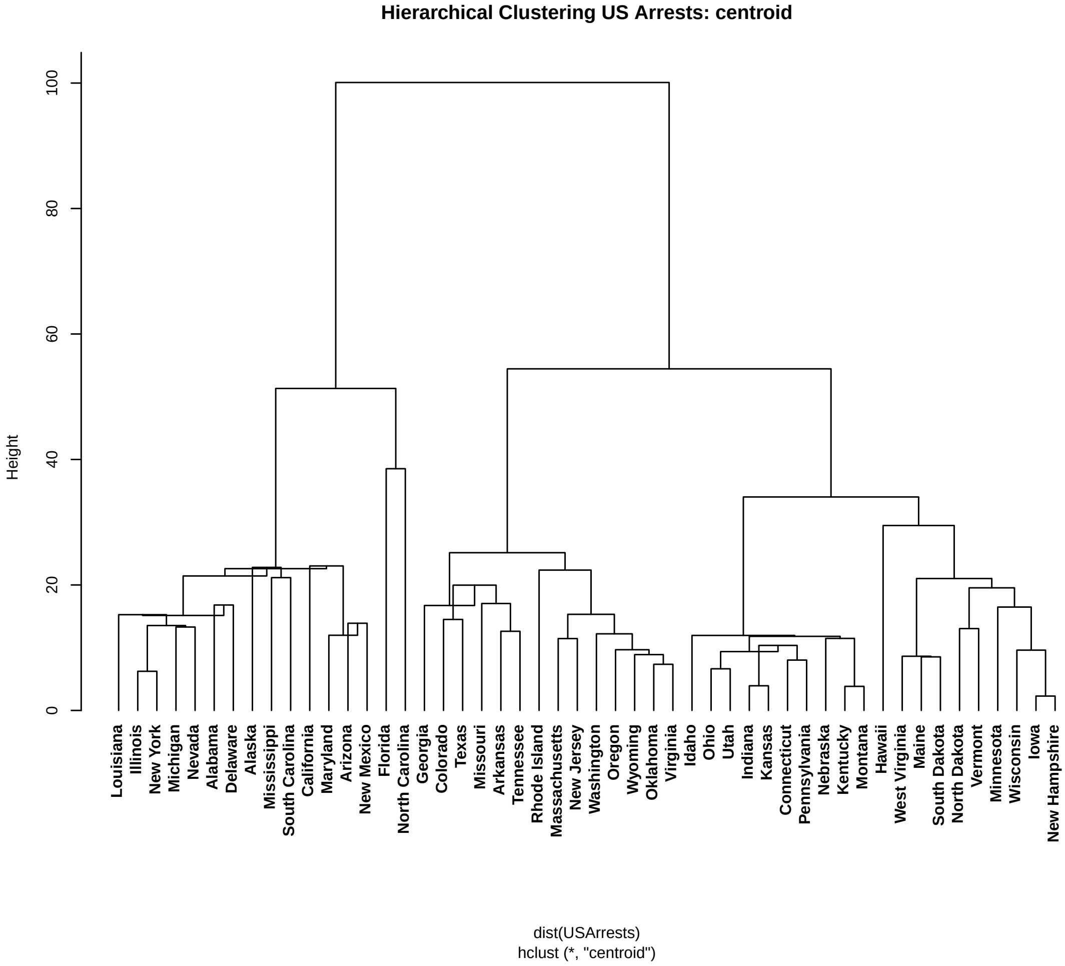

Figure 4.36: Hierarchical clustering of the US Arrest data set using centroid distance (not monotone).

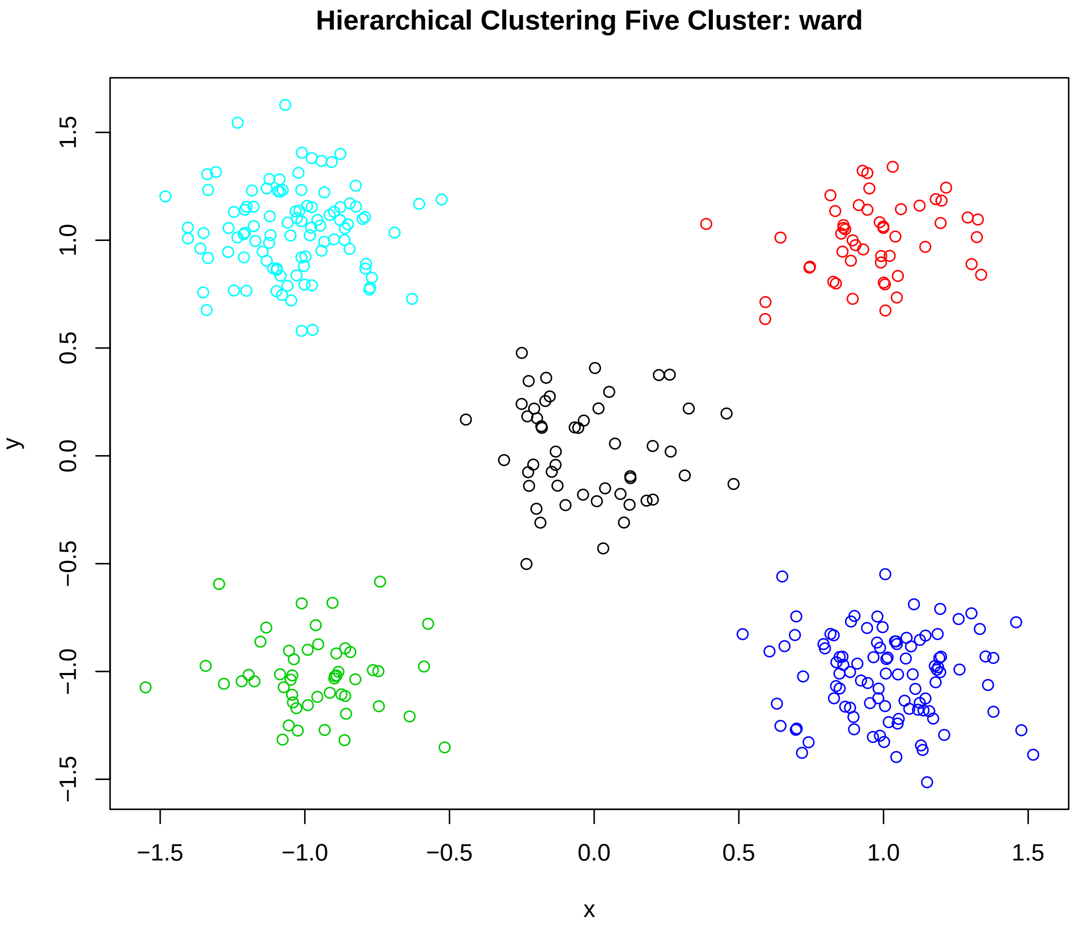

Figure 4.37: Hierarchical clustering of the five cluster data set. With all distance measures the optimal solution is found.

Next we apply hierarchical clustering to the five cluster data set. Fig. 4.37 shows the result for Ward's distance which is perfect. The results do not change if other distance measures than Ward's are used. To determine the clusters, the dendrogram has to be cut by the R function `cutree()`. We used following code:

```
hc <- hclust(dist(x), method="ward")
cl <- cutree(hc,k=5)
plot(x, col = cl,main=paste("Hierarchical Clustering Five Cluster: ","ward",sep=""))
```

We apply hierarchical clustering to the iris data set. Fig. 4.38 shows the results for the distance measures Ward, average linkage, complete linkage, and single linkage for 3 and 5 components. Ward with 3 components performs well and average linkage with 3 components is worse. However, hierarchical clustering has problems to separate the close iris species. For 5 components either equal large cluster or small clusters are separated. Ward divides true clusters in equal large

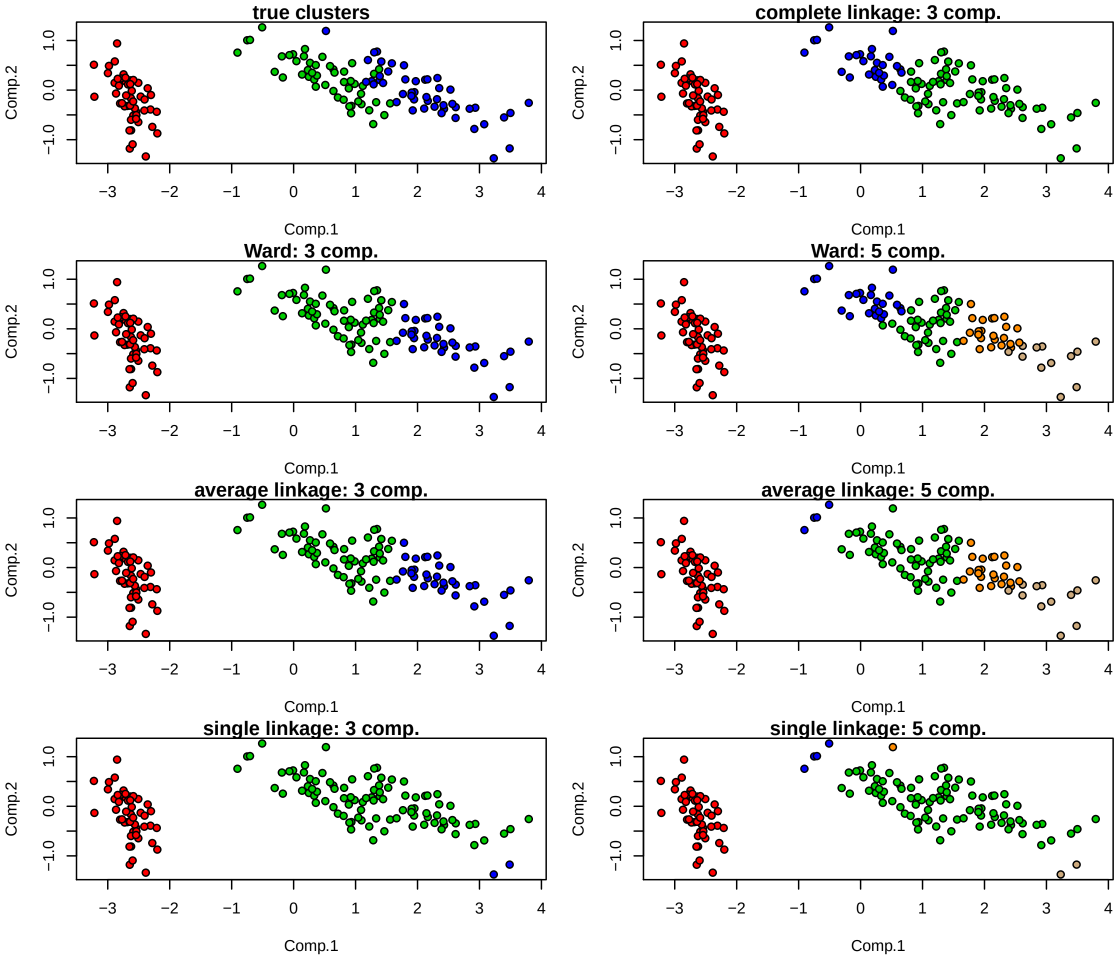

Figure 4.38: Hierarchical clustering of the iris data set. Ward with 3 components performs well and average linkage with 3 components is worse. For 5 components either equal large cluster or small clusters are separated.

clusters while other methods separate small clusters. Single linkage separates out clusters with large distances to other clusters which do not reflect the true clusters.

Next we apply hierarchical clustering to the multiple tissue data set. Fig. 4.39 shows the results for the distance measures Ward, average linkage, complete linkage, and single linkage for 4 and 6 components. Ward with 4 components performs well. Again the correct number of clusters is essential to obtain good clustering results.

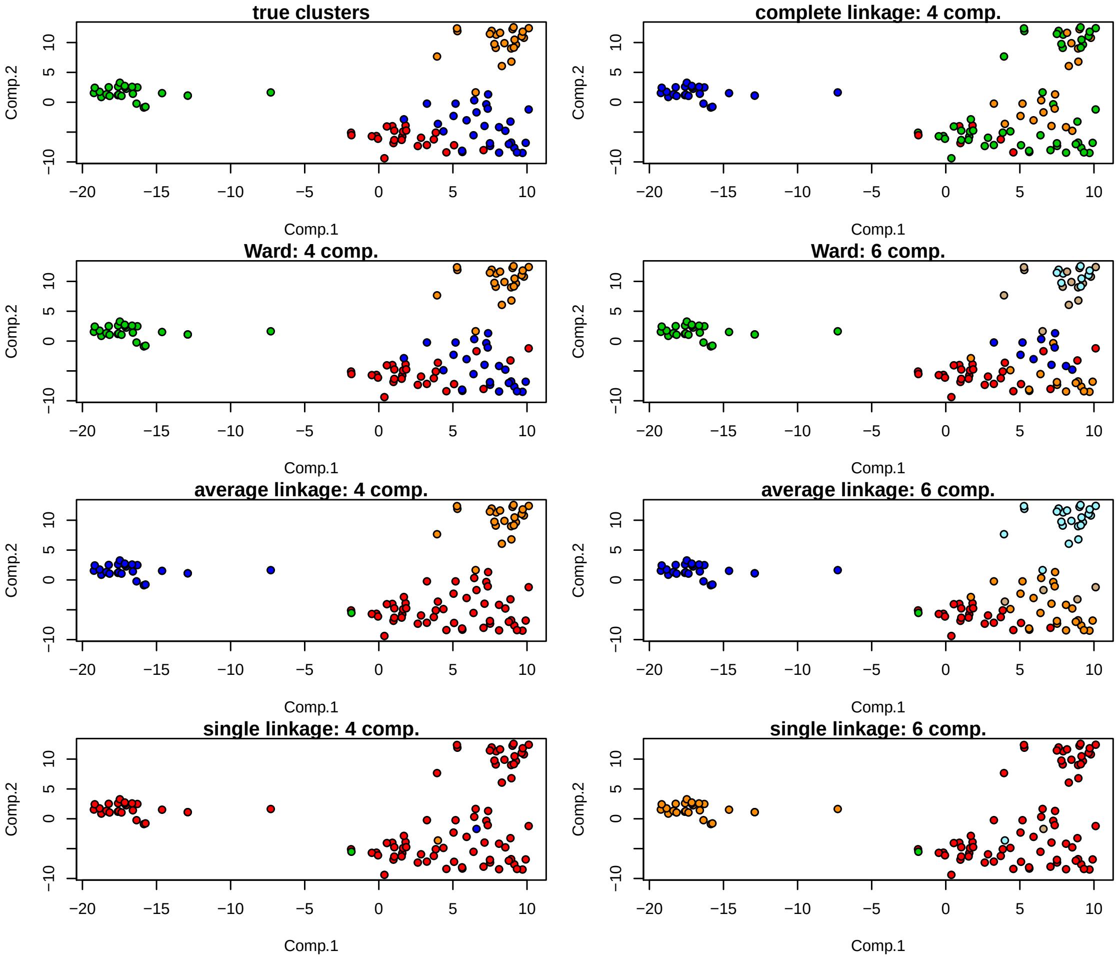

Figure 4.39: Hierarchical clustering of the multiple tissues data set. Ward with 4 components performs well. If correct number of cluster is known, the performance is better than with wrong number of clusters.

Chapter 5

Linear Models

In Subsection 3.2.4 we have considered linear regression for bivariate variables. We expressed one variable y as a linear function of the other variable x:

$$y \; = \; a \; + \; b \, x \; . \tag{5.1}$$

If fitting a linear function (a line), that is, to find optimal parameters a and b, the objective was the *sum of the squared deviations* between the y values and the regression line. The line that optimized this criterion is the *least squares line*. We now generalize this approach to the multivariate case. We already noticed in the simple bivariate case that interchanging the role of x and y may result in a different functional dependency between x and y.

The scalar variable y is called the *dependent variable*. We now generalize x to a vector of features $\boldsymbol{x}$ with components x_j which are called *explanatory variables*, *independent variables*, *regressors*, or *features*.

The estimation of y from a vector of explanatory variables $\boldsymbol{x}$ is called *multiple linear regression*. If y is generalized to a vector $\boldsymbol{y}$, then this is called *multivariate linear regression*. We focus on multiple linear regression, that is, the case where multiple features are summarized in the vector $\boldsymbol{x}$.

5.1 Linear Regression

5.1.1 The Linear Model

We assume to have m features $x_1, \ldots, x_m$ which are summarized by the vector $\boldsymbol{x} = (x_1, \ldots, x_m)$. The general form of a linear model is

$$y \; = \; \beta_0 \; + \; \sum_{j=1}^{m} x_j \, \beta_j \; + \; \epsilon \; . \tag{5.2}$$

This model has $(m + 1)$ *parameters* $\beta_0, \beta_1, \ldots, \beta_m$ which are unknown and have to be estimated. ϵ is an additive *noise* or *error* term which accounts for the difference between the predicted value and the observed outcome y.

To simplify the notation, we extend the vector of features by a one: $\boldsymbol{x} = (1, x_1, \ldots, x_m)$. Consequently, we use the parameter vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_m)$ to denote the linear model in vector notation by:

$$y \; = \; \boldsymbol{x}^T \boldsymbol{\beta} \; + \; \epsilon \; . \tag{5.3}$$

If the constant 1 is counted as independent variable, then $(m+1)$ is both the number of parameters and the number of the independent variables. In some textbooks this might be confusing because m and $(m+1)$ may appear in the formulas.

We assume to have n observations $\{(y_i, \boldsymbol{x}_i) \mid 1 \leq i \leq n\}$. The y_i are summarized by a vector $\boldsymbol{y}$, the $\boldsymbol{x}_i$ in a matrix $\boldsymbol{X} \in \mathbb{R}^{n \times (m+1)}$ ($\boldsymbol{x}_i$ is the i-th row of $\boldsymbol{X}$), and the ϵ_i in a vector $\boldsymbol{\epsilon} \in \mathbb{R}^n$. For n observations we obtain the matrix equation:

$$\boldsymbol{y} \; = \; \boldsymbol{X} \, \boldsymbol{\beta} \; + \; \boldsymbol{\epsilon} \; . \tag{5.4}$$

5.1.2 Interpretations and Assumptions

The linear model can be applied in different frameworks, where the independent variables have different interpretations and assumptions. The parameter estimation depends only on the noise assumption. The task which must be solved or the study design, from which the data comes, determines interpretations, assumptions, and design of the dependent variables.

5.1.2.1 Interpretations

One of the main differences is whether the explanatory / independent variables are random variables sampled together with the dependent variable or constants which are fixed according to the task to solve.

Our model for bivariate data from Subsection 3.2.4 is a model with one independent variable:

$$y_i \; = \; \beta_0 \; + \; \beta_1 x_i \; + \; \epsilon_i \; , \tag{5.5}$$

where β_0 is the y-intercept and β_1 the slope.

An example with 7 observations in matrix notation is

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} \; = \; \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \\ 1 & x_5 \\ 1 & x_6 \\ 1 & x_7 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \; + \; \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \\ \epsilon_7 \end{pmatrix} \; . \tag{5.6}$$

An example for a model with two regressors is

$$y_i \; = \; \beta_0 \; + \; \beta_1 x_{i1} \; + \; \beta_2 x_{i2} \; + \; \epsilon_i \; . \tag{5.7}$$

For 7 observations this model leads to following matrix equation:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ 1 & x_{31} & x_{32} \\ 1 & x_{41} & x_{42} \\ 1 & x_{51} & x_{52} \\ 1 & x_{61} & x_{62} \\ 1 & x_{71} & x_{72} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \\ \epsilon_7 \end{pmatrix} . \tag{5.8}$$

We show an example for a *cell means model* or a *one-way ANOVA* model. We assume that from the study design we know 3 groups and want to find the mean for each group. The model is

$$y_{gi} \; = \; \beta_g \; + \; \epsilon_{gi} \; , \tag{5.9}$$

where β_g is the mean of group g. For example we have three groups and 3 examples for the first group, and two examples for the second and third group. In matrix notation this example with 7 observations and three groups is

$$\begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{31} \\ y_{32} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{31} \\ \epsilon_{32} \end{pmatrix} . \tag{5.10}$$

We present another example of an ANOVA model which is again a one-way ANOVA model. We are interested in the offset from a reference group. This model is typically for a study design with one *control group* or *reference group* and multiple *treatment groups*. The offset of group g from group 1 is denoted by β_g, thus $\beta_1 = 0$.

$$y_{gi} \; = \; \beta_0 \; + \; \beta_g \; + \; \epsilon_{gi} \; . \tag{5.11}$$

For three groups and 7 observations (3 in group $g = 1$, 2 in group $g = 2$, and 2 in group $g = 3$), the matrix equation is

$$\begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{31} \\ y_{32} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_2 \\ \beta_3 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{31} \\ \epsilon_{32} \end{pmatrix} . \tag{5.12}$$

The mean of the reference group is β_0 and β_g is the difference to the reference group. In this design we know that $\beta_1 = 0$, therefore we did not include it.

A more complicated model is the *two-way ANOVA* model which has two known groupings or two known *factors*. Each observation belongs to a group of the first grouping and at the same time to a group of the second grouping, that is, each observation is characterized by two factors.

The model is

$$y_{ghi} \;=\; \beta_0 \;+\; \beta_g \;+\; \alpha_h \;+\; (\beta\alpha)_{gh} \;+\; \epsilon_{ghi} \;, \tag{5.13}$$

where g denotes the first factor (grouping) and h the second factor (grouping), and i indicates the replicate for this combination of factors. The term $(\beta\alpha)_{gh}$ accounts for *interaction effects* between the factors, while β_g and α_h are the *main effects* of the factors.

This model has too many parameters to possess a unique solution for the parameters if each combination of groups is observed exactly once. One observation per combination of groups is the minimal data set. Consequently, noise free observations can be modeled by more than one set of parameters. Even for a large number of noise free observations the situation does not change: there is more that one set of parameters which gives the optimal solution. The solution to this over-parametrization is to include additional constraints which use up some degrees of freedom. These constraints are that either

- the main and interaction effect parameters sum to zero for each index (*sum-to-zero constraint*) or
- all parameters that contain the index 1 are zero (*corner point parametrization*).

With the corner point parametrization we have

$$\begin{aligned} \alpha_1 \;&=\; 0 && (5.14)\\ \beta_1 \;&=\; 0 && (5.15)\\ (\beta\alpha)_{1h} \;&=\; 0 && (5.16)\\ (\beta\alpha)_{g1} \;&=\; 0 \,. && (5.17) \end{aligned}$$

We present an example, where the first factor has 3 levels $1 \leq g \leq 3$, the second factor has 2 levels $1 \leq h \leq 2$, and for each combination of factors there are two replicates $1 \leq i \leq 2$. In matrix notation we have

$$\begin{pmatrix} y_{111}\\ y_{112}\\ y_{211}\\ y_{212}\\ y_{311}\\ y_{312}\\ y_{121}\\ y_{122}\\ y_{221}\\ y_{222}\\ y_{321}\\ y_{322} \end{pmatrix} \;=\; \begin{pmatrix} 1&0&0&0&0&0\\ 1&0&0&0&0&0\\ 1&0&0&1&0&0\\ 1&0&0&1&0&0\\ 1&1&0&0&0&0\\ 1&1&0&0&0&0\\ 1&1&0&1&1&0\\ 1&1&0&1&1&0\\ 1&0&1&0&0&0\\ 1&0&1&0&0&0\\ 1&0&1&1&0&1\\ 1&0&1&1&0&1 \end{pmatrix} \begin{pmatrix} \beta_0\\ \beta_2\\ \beta_3\\ \alpha_2\\ (\beta\alpha)_{22}\\ (\beta\alpha)_{32} \end{pmatrix} \;+\; \begin{pmatrix} \epsilon_{111}\\ \epsilon_{112}\\ \epsilon_{211}\\ \epsilon_{212}\\ \epsilon_{311}\\ \epsilon_{312}\\ \epsilon_{121}\\ \epsilon_{122}\\ \epsilon_{221}\\ \epsilon_{222}\\ \epsilon_{321}\\ \epsilon_{322} \end{pmatrix} \;. \tag{5.18}$$

5.1.2.2 Assumptions

The standard linear regression model has the following assumptions:

- **Strict exogeneity.** The errors have zero mean conditioned on the regressors:

  $$\mathrm{E}(\boldsymbol{\epsilon} \mid \boldsymbol{X}) \ = \ \mathbf{0} \ . \tag{5.19}$$

  Therefore the errors have zero mean $\mathrm{E}(\boldsymbol{\epsilon}) = \mathbf{0}$ and they are independent of the regressors $\mathrm{E}(\boldsymbol{X}^T \boldsymbol{\epsilon}) = \mathbf{0}$.

- **Linear independence.** The regressors must be linearly independent almost surely.

  $$\Pr(\mathrm{rank}(\boldsymbol{X}) = m + 1) \ = \ 1 \ . \tag{5.20}$$

  If $\boldsymbol{X}$ does not have full rank, then estimation is only possible in the subspace spanned by the $\boldsymbol{x}_i$. To obtain theoretical properties of the estimator, the second moments should be finite to ensure $\mathrm{E}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X})$ to be finite and positive definite.

- **Spherical errors.**

  $$\mathrm{Var}(\boldsymbol{\epsilon} \mid \boldsymbol{X}) \ = \ \sigma^2 \, \boldsymbol{I}_n \ . \tag{5.21}$$

  Therefore the error has the same variance in each observation $\mathrm{E}(\epsilon_i^2 \mid \boldsymbol{X}) = \sigma^2$ (*homoscedasticity*). If this is violated, then a weighted least squared estimate should be used. Further the errors of different observations are not correlated $\mathrm{E}(\epsilon_i \epsilon_k \mid \boldsymbol{X}) = 0$ for $i \neq k$ (no autocorrelation).

Normality of the Errors. For further theoretical properties often the errors are assumed to be normally distributed given the regressors:

$$\boldsymbol{\epsilon} \mid \boldsymbol{X} \ \sim \ \mathcal{N}\big(\mathbf{0} \, , \, \sigma^2 \, \boldsymbol{I}_n\big) \tag{5.22}$$

In this case the estimator is the maximum likelihood estimator, which is asymptotically efficient, that is, it is asymptotically the best possible estimator. Further it is possible to test hypotheses based on the normality assumption because the distribution of the estimator is known.

In many applications the samples $\{(y_i, \boldsymbol{x}_i)\}$ are assumed to be *independent and identically distributed* (iid). The samples are independent,

$$\Pr\left((y_i, \boldsymbol{x}_i) \mid (y_1, \boldsymbol{x}_1), \ldots, (y_{i-1}, \boldsymbol{x}_{i-1}), (y_{i+1}, \boldsymbol{x}_{i+1}), \ldots, (y_n, \boldsymbol{x}_n)\right) \ = \ \Pr\left((y_i, \boldsymbol{x}_i)\right) \ , \tag{5.23}$$

and are identically distributed,

$$\Pr\left((y_i, \boldsymbol{x}_i)\right) \ = \ \Pr\left((y_k, \boldsymbol{x}_k)\right) \ . \tag{5.24}$$

For iid samples the assumptions simplify to

- **Exogeneity.** Each error has zero mean conditioned on the regressor:

  $$\mathrm{E}(\epsilon_i \mid \boldsymbol{x}_i) \ = \ 0 \ . \tag{5.25}$$

- **Linear independence.** The covariance matrix

$$\mathrm{Var}(\boldsymbol{x}) \;=\; \mathrm{E}(\boldsymbol{x}\boldsymbol{x}^T) \;=\; \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}(\boldsymbol{x}_i\boldsymbol{x}_i^T) \;=\; \mathrm{E}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X})\,. \tag{5.26}$$

  must have full rank.

- **Homoscedasticity.**

$$\mathrm{Var}(\epsilon_i \mid \boldsymbol{x}_i) \;=\; \sigma^2\,. \tag{5.27}$$

For *time series models* the iid assumption does not hold. In this case the assumptions are

- the stochastic process $\{(y_i, \boldsymbol{x}_i)\}$ is stationary (probability distribution is the same when shifted in time) and ergodic (time average is the population average);
- the regressors are predetermined: $\mathrm{E}(\boldsymbol{x}_i\epsilon_i) = 0$ for all $i = 1, \ldots, n$;
- the $(m+1) \times (m+1)$ matrix $\mathrm{E}(\boldsymbol{x}_i\boldsymbol{x}_i^T)$ is of full rank;
- the sequence $\{\boldsymbol{x}_i\epsilon_i\}$ is a martingale difference sequence (zero mean given the past) with existing second moments $\mathrm{E}(\epsilon_i^2\boldsymbol{x}_i\boldsymbol{x}_i^T)$.

Linear models for time series are called *autoregressive models*.

5.1.3 Least Squares Parameter Estimation

The *residual* for the i-th observation is

$$r_i \;=\; y_i \;-\; \boldsymbol{x}_i^T\,\tilde{\boldsymbol{\beta}}\,, \tag{5.28}$$

where $\tilde{\boldsymbol{\beta}}$ is a candidate for the parameter vector $\boldsymbol{\beta}$. The residual r_i measures how well y_i is predicted by the linear model with parameters $\tilde{\boldsymbol{\beta}}$.

To assess how well all observations are fitted simultaneously by a linear model, the squared residuals of all observation are summed up to S, which is called the *sum of squared residuals* (SSR), the *error sum of squares* (ESS), or *residual sum of squares* (RSS):

$$S(\tilde{\boldsymbol{\beta}}) \;=\; \sum_{i=1}^{n} r_i^2 \;=\; \sum_{i=1}^{n}\left(y_i \;-\; \boldsymbol{x}_i^T\,\tilde{\boldsymbol{\beta}}\right)^2 \;=\; \left(\boldsymbol{y} \;-\; \boldsymbol{X}\tilde{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{y} \;-\; \boldsymbol{X}\,\tilde{\boldsymbol{\beta}}\right)\,. \tag{5.29}$$

The *least squares estimator* $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$ minimizes $S(\tilde{\boldsymbol{\beta}})$:

$$\hat{\boldsymbol{\beta}} \;= \arg\min_{\tilde{\boldsymbol{\beta}}} S(\tilde{\boldsymbol{\beta}}) \;=\; \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}\,. \tag{5.30}$$

The solution is obtained by setting the derivative of $S(\tilde{\boldsymbol{\beta}})$ with respect to the parameter vector $\tilde{\boldsymbol{\beta}}$ to zero:

$$\frac{\partial S(\tilde{\boldsymbol{\beta}})}{\partial\tilde{\boldsymbol{\beta}}} \;=\; 2\,\boldsymbol{X}^T\left(\boldsymbol{y} \;-\; \boldsymbol{X}\,\tilde{\boldsymbol{\beta}}\right) \;=\; \boldsymbol{0}\,. \tag{5.31}$$

The matrix $\boldsymbol{X}^+ = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$ is called the *pseudo inverse* of the matrix $\boldsymbol{X}$ because $\boldsymbol{X}^+\boldsymbol{X} = \boldsymbol{I}_m$.

The least squares estimator is the minimal variance linear unbiased estimator (MVLUE), that is, it is the best linear unbiased estimator. Under the normality assumption for the errors, the least squares estimator is the maximum likelihood estimator (MLE).

Concerning notation and the parameter vector, we have the true parameter vector $\boldsymbol{\beta}$, a candidate parameter vector or a variable $\tilde{\boldsymbol{\beta}}$, and an estimator $\hat{\boldsymbol{\beta}}$, which is in our case the least squares estimator.

5.1.4 Evaluation and Interpretation of the Estimation

5.1.4.1 Residuals and Error Variance

The estimated values for y are

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}\,\boldsymbol{X}^+\boldsymbol{y} = \boldsymbol{X}\,\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{P}\boldsymbol{y}\,, \tag{5.32}$$

where

$$\boldsymbol{P} = \boldsymbol{X}\,\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T \tag{5.33}$$

is a projection matrix, the *hat matrix* as it puts a hat on $\boldsymbol{y}$. We have $\boldsymbol{P}\boldsymbol{X} = \boldsymbol{X}$ and $\boldsymbol{P}^2 = \boldsymbol{P}$.

The minimal residuals or the least squares residuals are

$$\hat{\boldsymbol{\epsilon}} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}} = (\boldsymbol{I}_n - \boldsymbol{P})\,\boldsymbol{y} = (\boldsymbol{I}_n - \boldsymbol{P})\,\boldsymbol{\epsilon}\,. \tag{5.34}$$

Both $\boldsymbol{P}$ and $(\boldsymbol{I}_n - \boldsymbol{P})$ are symmetric and idempotent ($\boldsymbol{P} = \boldsymbol{P}^2$).

$S(\hat{\boldsymbol{\beta}})$ is the sum of squared residuals for the least squares estimator $\hat{\boldsymbol{\beta}}$, which can be used to estimate σ^2.

$$\begin{aligned} S(\hat{\boldsymbol{\beta}}) &= \left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right) \\ &= \boldsymbol{y}^T\boldsymbol{y} - 2\,\hat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{y} + \hat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}} \\ &= \boldsymbol{y}^T\boldsymbol{y} - \hat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{y} = \hat{\boldsymbol{\epsilon}}^T\boldsymbol{y}\,, \end{aligned} \tag{5.35}$$

where we used $\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}^T\boldsymbol{y}$.

The least squares estimate for σ^2 is

$$s^2 = \frac{1}{n-m-1}S(\hat{\boldsymbol{\beta}}) \tag{5.36}$$

and the maximum likelihood estimate for σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n}S(\hat{\boldsymbol{\beta}})\,. \tag{5.37}$$

The estimate s^2 is an unbiased estimator for σ^2 while the ML estimate $\hat{\sigma}^2$ is biased. Both are asymptotically optimal, that is, unbiased and efficient. The estimator with minimal mean squared error is

$$\tilde{\sigma}^2 \;=\; \frac{1}{n-m+1} S(\hat{\boldsymbol{\beta}}) \; . \tag{5.38}$$

The covariance of the vector of residuals is

$$\begin{aligned} \mathrm{E}(\hat{\boldsymbol{\epsilon}}\hat{\boldsymbol{\epsilon}}^T) \;&=\; (\boldsymbol{I}_n \;-\; \boldsymbol{P}) \; \mathrm{E}(\boldsymbol{\epsilon}\,\boldsymbol{\epsilon}^T) \; (\boldsymbol{I}_n \;-\; \boldsymbol{P}) \\ &=\; \sigma^2 \, (\boldsymbol{I}_n \;-\; \boldsymbol{P})^2 \;=\; \sigma^2 \, (\boldsymbol{I}_n \;-\; \boldsymbol{P}) \; , \end{aligned} \tag{5.39}$$

where we used Eq. (5.34). Further we assumed that the residuals have the covariance structure $\sigma^2 \boldsymbol{I}_n$ as the assumptions state.

5.1.4.2 Coefficient of determination

The *coefficient of determination* R^2 is the ratio of the variance "explained" by the model to the "total" variance of the dependent variable y:

$$\begin{aligned} R^2 \;&=\; \frac{\sum_{i=1}^{n} \left(\hat{y}_i \;-\; \overline{\hat{y}}\right)^2}{\sum_{i=1}^{n} \left(y_i \;-\; \overline{y}\right)^2} \\ &=\; \frac{\boldsymbol{y}^T \boldsymbol{P}^T \boldsymbol{L} \, \boldsymbol{P} \, \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{L} \, \boldsymbol{y}} \;=\; 1 \;-\; \frac{\boldsymbol{y}^T (\boldsymbol{I} \;-\; \boldsymbol{P}) \, \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{L} \, \boldsymbol{y}} \;=\; 1 \;-\; \frac{\mathrm{SSR}}{\mathrm{TSS}} \; , \end{aligned} \tag{5.40}$$

where $\boldsymbol{L} = \boldsymbol{I}_n - (1/n) \boldsymbol{1} \, \boldsymbol{1}^T$, with $\boldsymbol{1}$ as the n-dimensional vector of ones. $\boldsymbol{L}$ is the centering matrix which subtracts the mean from each variable. "TSS" is the total sum of squares for the dependent variable and "SSR" the sum of squared residuals denoted by S. To account for a constant offset, that is, the regression intercept, the data matrix $\boldsymbol{X}$ should contain a column vector of ones. In that case R^2 is between 0 and 1, the closer R^2 is to 1, the better the fit.

5.1.4.3 Outliers and Influential Observations

An *outlier* is an observation which is worse fitted by the model than other observations, that is, it has large error compared to other errors. An *influential observation* is an observation which has large effect on the model fitting or has large effect on the inferences based on the model. Outliers can be influential observations but need not be. Analogously, influential observations can be outliers but need not be.

5.1.4.3.1 Outliers. We define the *standardized residuals* or *studentized residuals* ρ_i as

$$\rho_i \;=\; \frac{\hat{\epsilon}_i}{\hat{\sigma} \, \sqrt{1 \;-\; P_{ii}}} \; . \tag{5.41}$$

P_{ii} are the diagonal elements of the hat matrix $\boldsymbol{P}$ defined in Eq. (5.33) and $\hat{\sigma}^2$ is an estimate of σ^2. The standardized residuals can be used to check the fitted model and whether the model assumptions are met or not. Such assumptions are linearity, normality, and independence. In particular, an outlier may be detected via the standardized residuals ρ_i because they have the same variance.

Another way is to do leave-one-out regression, where observation $(y_i, \boldsymbol{x}_i)$ is removed from the data set and a least squares estimate performed on the remaining $(n - 1)$ observations. The least squares estimator $\hat{\boldsymbol{\beta}}_{(i)}$ on the data set where $(y_i, \boldsymbol{x}_i)$ is left out is:

$$\hat{\boldsymbol{\beta}}_{(i)} \;=\; \hat{\boldsymbol{\beta}} \;-\; \frac{\hat{\epsilon}_i}{1 \;-\; P_{ii}} \, (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x}_i \; . \tag{5.42}$$

Therefore the residual of the left-out observation is

$$\hat{\epsilon}_{(i)} \;=\; \frac{\hat{\epsilon}_i}{1 \;-\; P_{ii}} \; . \tag{5.43}$$

Plotting the leave-one-out residuals against the standard residuals may reveal outliers. However outliers can already be detected by $(1 - P_{ii})$: the closer P_{ii} to one, the more likely is the i-th observation an outlier.

5.1.4.3.2 Influential Observations. An influential observation $(y_i, \boldsymbol{x}_i)$ has large effect on the estimates $\hat{\boldsymbol{\beta}}$ or $\boldsymbol{X}\hat{\boldsymbol{\beta}}$. This means that the estimates are considerably different if observation $(y_i, \boldsymbol{x}_i)$ is removed. Fig. 5.1 shows a simple linear regression with three marked outliers. Observations 1 and 3 deviate in the x-direction. Observations 2 and 3 appear as outliers in the y-direction. Observation 1 is located close to the regression line which would be obtained without it. Thus, it is not influential. However, observation 3 has a large effect on the regression line compared to regression if it is removed. Thus, observation 3 is influential. Observation 2 is influential to some degree but much less than observation 3.

With the hat matrix $\boldsymbol{P}$ we can express $\hat{\boldsymbol{y}}$ as $\hat{\boldsymbol{y}} = \boldsymbol{P}\boldsymbol{y}$, therefore

$$\hat{y}_i \;=\; \sum_{j=1}^{n} P_{ij} y_j \;=\; P_{ii} y_i \;+\; \sum_{j, j \neq i} P_{ij} y_j \; . \tag{5.44}$$

If P_{ii} is large, then P_{ij} for $j \neq i$ is small because $\boldsymbol{P}$ is idempotent. Therefore P_{ii} is called the *leverage* of y_i, that is, how much y_i contributes to its estimate.

The influence of the i-th observation can be measured by *Cook's distance*

$$D_i \;=\; \frac{\rho_i^2}{m+1} \, \frac{P_{ii}}{1 \;-\; P_{ii}} \; . \tag{5.45}$$

If D_i is large, the observation $(y_i, \boldsymbol{x}_i)$ has considerable influence on the estimates.

This distance can be written as

$$\begin{aligned} D_i \;&=\; \frac{(\hat{\boldsymbol{\beta}}_{(i)} \;-\; \hat{\boldsymbol{\beta}})^T \boldsymbol{X}^T \boldsymbol{X} (\hat{\boldsymbol{\beta}}_{(i)} \;-\; \hat{\boldsymbol{\beta}})}{(m+1)\, s^2} \\ &=\; \frac{(\boldsymbol{X}\, \hat{\boldsymbol{\beta}}_{(i)} \;-\; \boldsymbol{X}\, \hat{\boldsymbol{\beta}})^T (\boldsymbol{X}\, \hat{\boldsymbol{\beta}}_{(i)} \;-\; \boldsymbol{X}\, \hat{\boldsymbol{\beta}})}{(m+1)\, s^2} \\ &=\; \frac{(\hat{\boldsymbol{y}}_{(i)} \;-\; \hat{\boldsymbol{y}})^T (\hat{\boldsymbol{y}}_{(i)} \;-\; \hat{\boldsymbol{y}})}{(m+1)\, s^2} \; . \end{aligned} \tag{5.46}$$

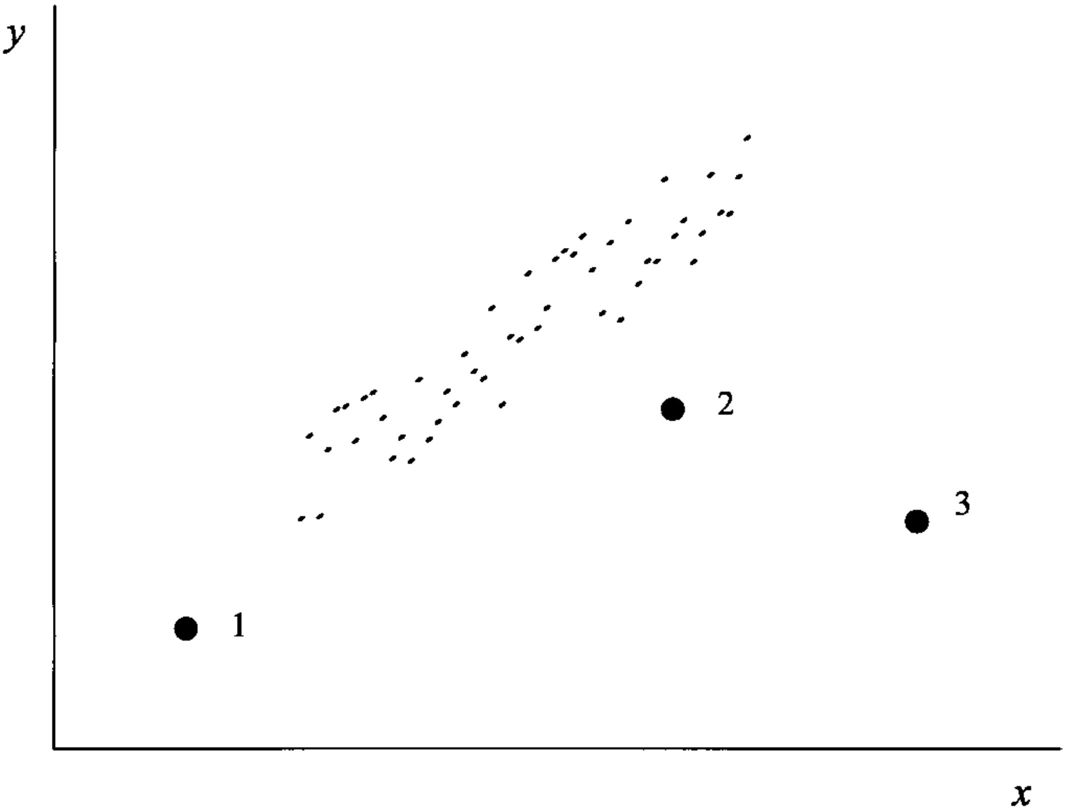

Figure 5.1: Simple linear regression with three marked outliers. Observations 1 and 3 deviate in the x-direction. Observations 2 and 3 appear as outliers in the y-direction. Observation 1 is not influential but observation 3 is. Figure from Rencher and Schaalje [2008].

Thus, D_i is proportional to the Euclidean distance between the estimate $\hat{\boldsymbol{y}}$ using all data and the estimate $\hat{\boldsymbol{y}}_{(i)}$ where observation $(y_i, \boldsymbol{x}_i)$ is removed.

More complicated but giving the same result is first to perform a leave-one-out estimate, where each observation $(y_i, \boldsymbol{x}_i)$ is left out, and, subsequently, compare the estimated values to the estimated values with all data.

5.1.5 Confidence Intervals for Parameters and Prediction

5.1.5.1 Normally Distributed Error Terms

If the error terms are normally distributed then the least squares estimator is a maximum likelihood estimator which is asymptotically normally distributed:

$$\hat{\boldsymbol{\beta}} \overset{d}{\rightarrow} \mathcal{N}\left(\boldsymbol{\beta},\, \sigma^2 \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1}\right) , \tag{5.47}$$

where $\overset{d}{\rightarrow}$ means convergence in distribution. This means that the distribution of $\hat{\boldsymbol{\beta}}$ is increasingly (with number of samples) better modeled by the normal distribution. This maximum likelihood estimator is efficient and unbiased, that is, it reaches the Cramer-Rao lower bound, and therefore is optimal for unbiased estimators.

This asymptotic distribution gives an approximated two-sided confidence interval for the j-th component of the vector $\hat{\boldsymbol{\beta}}$:

$$\beta_j \in \left[\hat{\beta}_j \pm t_{\alpha/2,n-m-1}\, s \sqrt{\left[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\right]_{jj}} \right] \tag{5.48}$$

where $t_{\alpha/2,n-m-1}$ is the upper $\alpha/2$ percentage point of the central t-distribution and α is the desired significance level of the test (probability of rejecting H_0 when it is true). This means we are $100(1-\alpha)\%$ confident that the interval contains the true β_j. It is important to known that the confidence intervals do not hold simultaneously for all β_j.

The confidence interval for the noise free prediction is

$$\boldsymbol{x}^T\boldsymbol{\beta} \in \left[\boldsymbol{x}^T\hat{\boldsymbol{\beta}} \pm t_{\alpha/2,n-m-1}\, s \sqrt{\boldsymbol{x}^T \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1} \boldsymbol{x}} \right] . \tag{5.49}$$

Again this holds only for a single prediction but not for multiple simultaneous predictions.

If noise ϵ is added then we have a confidence interval for the prediction:

$$y \in \left[\hat{y} \pm t_{\alpha/2,n-m-1}\, s \sqrt{1 + \boldsymbol{x}^T \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1} \boldsymbol{x}} \right] , \tag{5.50}$$

where $\hat{y} = \boldsymbol{x}^T\hat{\boldsymbol{\beta}}$. Of course, this holds only for a single prediction but not for multiple simultaneous predictions.

The estimator s^2 is distributed according to a chi-squared distribution:

$$s^2 \sim \frac{\sigma^2}{n-m-1}\, \chi^2_{n-m-1} . \tag{5.51}$$

The variance is $2\sigma^4/(n-m-1)$ and does not attain the Cramer-Rao lower bound $2\sigma^4/n$. There is no unbiased estimator with lower variance, this means that the estimator is the minimal variance unbiased estimator (MVUE). The estimator $\tilde{\sigma}^2$ from above has the minimal mean squared error. An advantage of s^2 is that it is independent of $\hat{\boldsymbol{\beta}}$ which helps for tests based on these estimators.

A confidence interval for σ^2 is given by

$$\frac{(n-m-1)\, s^2}{\chi^2_{\alpha/2,n-m-1}} \leq \; \sigma^2 \; \leq \frac{(n-m-1)\, s^2}{\chi^2_{1-\alpha/2,n-m-1}} \tag{5.52}$$

at $100(1-\alpha)\%$ confidence.

5.1.5.2 Error Term Distribution Unknown

For unknown error distributions we still known that the least squares estimator for $\boldsymbol{\beta}$ is consistent, that is, $\hat{\boldsymbol{\beta}}$ converges in probability to the true value $\boldsymbol{\beta}$. The following results are obtained by the law of large number and the central limit theorem. The estimator $\hat{\boldsymbol{\beta}}$ is asymptotically normally distributed:

$$\sqrt{n}\,(\hat{\boldsymbol{\beta}} \,-\, \boldsymbol{\beta}) \xrightarrow{d} \mathcal{N}\big(\mathbf{0},\, \sigma^2\, (\boldsymbol{X}^T\boldsymbol{X})^{-1}\big)\;, \tag{5.53}$$

which gives

$$\hat{\boldsymbol{\beta}} \;\sim_a\; \mathcal{N}\big(\boldsymbol{\beta},\, \frac{\sigma^2}{n}\, (\boldsymbol{X}^T\boldsymbol{X})^{-1}\big)\;, \tag{5.54}$$

where $\sim_a$ means asymptotically distributed. This asymptotic distribution gives an approximated two-sided confidence interval for the j-th component of the vector $\hat{\boldsymbol{\beta}}$:

$$\beta_j \;\in\; \left[\, \hat{\beta}_j \;\pm\; q^{\mathcal{N}(0,1)}_{1-\alpha/2}\, \sqrt{\tfrac{1}{n}\hat{\sigma}^2\big[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\big]_{jj}}\, \right] \tag{5.55}$$

at a $(1-\alpha)$ confidence level. Here q is the quantile function of the standard normal distribution.

If the fourth moment of the error ϵ exists, the least squares estimator for σ^2 is consistent and asymptotically normal, too. The asymptotic normal distribution is

$$\sqrt{n}(\hat{\sigma}^2 \,-\, \sigma^2) \xrightarrow{d} \mathcal{N}\big(0,\, \mathrm{E}(\epsilon^4) \,-\, \sigma^4\big)\;, \tag{5.56}$$

which gives

$$\hat{\sigma}^2 \;\sim_a\; \mathcal{N}\big(\sigma^2\;,\; (\mathrm{E}(\epsilon^4) \,-\, \sigma^4)\,/\,n\big)\;. \tag{5.57}$$

Also the predicted response $\hat{y}$ is a random variable given $\boldsymbol{x}$, the distribution of which is determined by that of $\hat{\boldsymbol{\beta}}$:

$$\sqrt{n}\,(\hat{y} \,-\, y) \xrightarrow{d} \mathcal{N}\big(0,\, \sigma^2\, \boldsymbol{x}^T\, (\boldsymbol{X}^T\boldsymbol{X})^{-1}\, \boldsymbol{x}\big)\;, \tag{5.58}$$

which gives

$$\hat{y} \;\sim_a\; \mathcal{N}\big(y,\, \frac{\sigma^2}{n}\, \boldsymbol{x}^T\, (\boldsymbol{X}^T\boldsymbol{X})^{-1}\, \boldsymbol{x}\big)\;. \tag{5.59}$$

This distribution gives a confidence interval for mean response y, that is, an error bar on the prediction:

$$y \in \left[\boldsymbol{x}^T \hat{\boldsymbol{\beta}} \pm q_{1-\alpha/2}^{\mathcal{N}(0,1)} \sqrt{\tfrac{1}{n} \hat{\sigma}^2 \boldsymbol{x}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x}} \right] \tag{5.60}$$

at a $(1 - \alpha)$ confidence level.

5.1.6 Tests of Hypotheses

We want to test whether some independent variables (regressors) are relevant for the regression. These tests assume the null hypothesis that models without some variables have the same fitting quality as models with these variables. If the null hypothesis is rejected, then the variables are relevant for fitting.

5.1.6.1 Test for a Set of Variables Equal to Zero

We remove h variables from the original model (or data) and fit a reduced model. The error is assumed to be normally distributed. We divide the data in $m - h + 1$ variables (including the constant variable) and h variables which will be removed:

$$\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2) \tag{5.61}$$

with $\boldsymbol{X}_1 \in \mathbb{R}^{n \times (m-h+1)}$ and $\boldsymbol{X}_2 \in \mathbb{R}^{n \times h}$. Also the parameters are accordingly partitioned:

$$\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2) \tag{5.62}$$

with $\boldsymbol{\beta}_1 \in \mathbb{R}^{m-h+1}$ and $\boldsymbol{\beta}_2 \in \mathbb{R}^h$. We want to test the null hypothesis H_0

$$\boldsymbol{\beta}_2 = \boldsymbol{0} \,. \tag{5.63}$$

We denote the least squares estimator for the reduced model that uses only $\boldsymbol{X}_1$ by $\hat{\boldsymbol{\beta}}_r \in \mathbb{R}^{m-h+1}$. In contrast, $\hat{\boldsymbol{\beta}}_1 \in \mathbb{R}^{m-h+1}$ are the first $(m - h + 1)$ components of the least squares estimator $\hat{\boldsymbol{\beta}}$ of the full model. We define an F statistic as follows:

$$F = \frac{\boldsymbol{y}^T (\boldsymbol{P} - \boldsymbol{P}_1) \boldsymbol{y} / h}{\boldsymbol{y}^T (\boldsymbol{I} - \boldsymbol{P}) \boldsymbol{y} / (n - m - 1)} = \frac{\left(\hat{\boldsymbol{\beta}}^T \boldsymbol{X}^T \boldsymbol{y} - \hat{\boldsymbol{\beta}}_r^T \boldsymbol{X}_1^T \boldsymbol{y} \right) / h}{\left(\boldsymbol{y}^T \boldsymbol{y} - \hat{\boldsymbol{\beta}}^T \boldsymbol{X}^T \boldsymbol{y} \right) / (n - m - 1)} \,, \tag{5.64}$$

where $\hat{\boldsymbol{\beta}}$ is the least squares estimator of the full model and $\hat{\boldsymbol{\beta}}_r$ the least squares estimator of the reduced model. The distribution of the F statistic is the following:

(i) If H_0: $\boldsymbol{\beta}_2 = \boldsymbol{0}$ is **false**, then F is distributed according to $F(h, n - m - 1, \lambda)$, where

$$\lambda = \boldsymbol{\beta}_2^T \left(\boldsymbol{X}_2^T \boldsymbol{X}_2 - \boldsymbol{X}_2^T \boldsymbol{X}_1 \left(\boldsymbol{X}_1^T \boldsymbol{X}_1 \right)^{-1} \boldsymbol{X}_1^T \boldsymbol{X}_2 \right) \boldsymbol{\beta}_2 / (2\sigma^2) \,. \tag{5.65}$$

(ii) If H_0: $\boldsymbol{\beta}_2 = \boldsymbol{0}$ is **true**, then $\lambda = 0$ and F is distributed according to $F(h, n - m - 1)$.

Source of Variation	Degrees of freedom	Sum of squares	Mean square
reduced $\boldsymbol{\beta}_r$	$\mathrm{df} = m - h + 1$	$S \;=\; \hat{\boldsymbol{\beta}}_r^T \boldsymbol{X}_1^T \boldsymbol{y}$	$S \;/\; \mathrm{df}$
improved $\boldsymbol{\beta}$	$\mathrm{df} = h$	$S \;=\; \hat{\boldsymbol{\beta}}^T \boldsymbol{X}^T \boldsymbol{y} \;-\; \hat{\boldsymbol{\beta}}_r^T \boldsymbol{X}_1^T \boldsymbol{y}$	$S \;/\; \mathrm{df}$
residual	$\mathrm{df} = n - m - 1$	$S \;=\; \boldsymbol{y}^T \boldsymbol{y} \;-\; \hat{\boldsymbol{\beta}}^T \boldsymbol{X}^T \boldsymbol{y}$	$S \;/\; \mathrm{df}$
total center	$\mathrm{df} = n - 1$	$S \;=\; \boldsymbol{y}^T \boldsymbol{y} \;-\; n\, \bar{y}$	$S \;/\; \mathrm{df}$
total	$\mathrm{df} = n$	$S \;=\; \boldsymbol{y}^T \boldsymbol{y}$	$S \;/\; \mathrm{df}$

Table 5.1: ANOVA table for F test of H_0: $\boldsymbol{\beta}_2 = \mathbf{0}$.

H_0 is rejected if $F \geq F_{\alpha,h,n-m-1}$, where $F_{\alpha,h,n-m-1}$ is the upper α percentage of the central F distribution. That is, H_0 is rejected if the p-value is smaller than α.

The statistic F can also be expressed by R^2:

$$F \;=\; \frac{\left(R^2 \;-\; R_1^2\right) \;/\; h}{\left(1 \;-\; R^2\right) \;/\; (n - m - 1)} \;, \tag{5.66}$$

where R^2 is the coefficient of determination for the full model and R_1^2 is the coefficient of determination for the reduced model using only $\boldsymbol{X}_1$. It can be shown that this test is equivalently to a likelihood ratio test.

These hypotheses tests are often summarized by the Analysis-of-Variance (ANOVA) table as shown in Tab. 5.1.

5.1.6.2 Test for a Single Variable Equal to Zero

To test the null hypothesis H_0: $\beta_j = 0$, the F statistic

$$F \;=\; \frac{\hat{\beta}_j^2}{s^2 \, \left[(\boldsymbol{X}^T \boldsymbol{X})^{-1}\right]_{jj}} \tag{5.67}$$

can be used. If H_0: $\beta_j = 0$ is **true**, then F is distributed according to $F(1, n - m - 1)$. We reject H_0: $\beta_j = 0$ if $F \geq F_{\alpha,1,(n-m-1)}$ or, equivalently, if the p-value is smaller than α.

Alternatively, the t-statistic

$$t_j \;=\; \frac{\hat{\beta}_j}{s \, \sqrt{\left[(\boldsymbol{X}^T \boldsymbol{X})^{-1}\right]_{jj}}} \tag{5.68}$$

can be used. We reject H_0: $\beta_j = 0$ if $|t_j| \geq t_{\alpha/2,(n-m-1)}$ or, equivalently, if the p-value is smaller than α.

If several β_j are tested for being zero, then we have to correct for multiple testing. The false discovery rate (FDR) can be controlled by the Benjamini-Hochberg procedure Benjamini and Hochberg [1995], Benjamini and Yekutieli [2001]. Alternatively, the familywise α level can be adjusted by the Bonferroni approach Bonferroni [1936].

5.1.7 Examples

5.1.7.1 Hematology Data

This data set is from Rencher and Schaalje [2008] page 252, Ex. 10.3, Table 10.1 and stems from Royston (1983). The following six hematology variables were measured on 51 workers:

1. y: lymphocyte count,
2. x_1: hemoglobin concentration,
3. x_2: packed-cell volume,
4. x_3: white blood cell count ($\times.01$),
5. x_4: neutrophil count,
6. x_5: serum lead concentration.

The data are given in Tab. 5.2.

```
hemData <- matrix(c(
+ 14,13.4,39,41,25,17,  15,14.6,46,50,30,20,  19,13.5,42,45,21,18,
+ 23,15.0,46,46,16,18,  17,14.6,44,51,31,19,  20,14.0,44,49,24,19,
+ 21,16.4,49,43,17,18,  16,14.8,44,44,26,29,  27,15.2,46,41,13,27,
+ 34,15.5,48,84,42,36,  26,15.2,47,56,27,22,  28,16.9,50,51,17,23,
+ 24,14.8,44,47,20,23,  26,16.2,45,56,25,19,  23,14.7,43,40,13,17,
+  9,14.7,42,34,22,13,  18,16.5,45,54,32,17,  28,15.4,45,69,36,24,
+ 17,15.1,45,46,29,17,  14,14.2,46,42,25,28,   8,15.9,46,52,34,16,
+ 25,16.0,47,47,14,18,  37,17.4,50,86,39,17,  20,14.3,43,55,31,19,
+ 15,14.8,44,42,24,29,   9,14.9,43,43,32,17,  16,15.5,45,52,30,20,
+ 18,14.5,43,39,18,25,  17,14.4,45,60,37,23,  23,14.6,44,47,21,27,
+ 43,15.3,45,79,23,23,  17,14.9,45,34,15,24,  23,15.8,47,60,32,21,
+ 31,14.4,44,77,39,23,  11,14.7,46,37,23,23,  25,14.8,43,52,19,22,
+ 30,15.4,45,60,25,18,  32,16.2,50,81,38,18,  17,15.0,45,49,26,24,
+ 22,15.1,47,60,33,16,  20,16.0,46,46,22,22,  20,15.3,48,55,23,23,
+ 20,14.5,41,62,36,21,  26,14.2,41,49,20,20,  40,15.0,45,72,25,25,
+ 22,14.2,46,58,31,22,  61,14.9,45,84,17,17,  12,16.2,48,31,15,18,
+ 20,14.5,45,40,18,20,  35,16.4,49,69,22,24,  38,14.7,44,78,34,16),
+ ncol=6,byrow=TRUE)
y <- hemData[,1]
x <- hemData[,2:6]
```

If we look at the correlation matrix

```
cor(hemData)
           [,1]        [,2]      [,3]       [,4]       [,5]        [,6]
[1,] 1.00000000  0.23330745 0.2516182 0.79073232 0.02264257  0.08290783
```

#	y	x_1	x_2	x_3	x_4	x_5	#	y	x_1	x_2	x_3	x_4	x_5
1	14	13.4	39	41	25	17	27	16	15.5	45	52	30	20
2	15	14.6	46	50	30	20	28	18	14.5	43	39	18	25
3	19	13.5	42	45	21	18	29	17	14.4	45	60	37	23
4	23	15.0	46	46	16	18	30	23	14.6	44	47	21	27
5	17	14.6	44	51	31	19	31	43	15.3	45	79	23	23
6	20	14.0	44	49	24	19	32	17	14.9	45	34	15	24
7	21	16.4	49	43	17	18	33	23	15.8	47	60	32	21
8	16	14.8	44	44	26	29	34	31	14.4	44	77	39	23
9	27	15.2	46	41	13	27	35	11	14.7	46	37	23	23
10	34	15.5	48	84	42	36	36	25	14.8	43	52	19	22
11	26	15.2	47	56	27	22	37	30	15.4	45	60	25	18
12	28	16.9	50	51	17	23	38	32	16.2	50	81	38	18
13	24	14.8	44	47	20	23	39	17	15.0	45	49	26	24
14	26	16.2	45	56	25	19	40	22	15.1	47	60	33	16
15	23	14.7	43	40	13	17	41	20	16.0	46	46	22	22
16	9	14.7	42	34	22	13	42	20	15.3	48	55	23	23
17	18	16.5	45	54	32	17	43	20	14.5	41	62	36	21
18	28	15.4	45	69	36	24	44	26	14.2	41	49	20	20
19	17	15.1	45	46	29	17	45	40	15.0	45	72	25	25
20	14	14.2	46	42	25	28	46	22	14.2	46	58	31	22
21	8	15.9	46	52	34	16	47	61	14.9	45	84	17	17
22	25	16.0	47	47	14	18	48	12	16.2	48	31	15	18
23	37	17.4	50	86	39	17	49	20	14.5	45	40	18	20
24	20	14.3	43	55	31	19	50	35	16.4	49	69	22	24
25	15	14.8	44	42	24	29	51	38	14.7	44	78	34	16
26	9	14.9	43	43	32	17							

Table 5.2: Rencher's hematology data Rencher and Schaalje [2008] page 252, Ex. 10.3, Table 10.1 — originally from Royston (1983). The variables are y: lymphocyte count, x_1: hemoglobin concentration, x_2: packed-cell volume, x_3: white blood cell count ($\times .01$), x_4: neutrophil count, and x_5: serum lead concentration.

```
[2,] 0.23330745  1.00000000 0.7737330 0.27650957 0.05537581 -0.08376682
[3,] 0.25161817  0.77373300 1.0000000 0.30847841 0.07642710  0.12970593
[4,] 0.79073232  0.27650957 0.3084784 1.00000000 0.60420947  0.07147757
[5,] 0.02264257  0.05537581 0.0764271 0.60420947 1.00000000  0.03169314
[6,] 0.08290783 -0.08376682 0.1297059 0.07147757 0.03169314  1.00000000
```

we see that the largest correlation between the response y and an explanatory variable is 0.79 between y and x_3.

5.1.7.1.1 Computing Estimates, Confidence Intervals, Tests. The mean $\bar{y}$ of the response variable y is

```
(by <- mean(y))
[1] 22.98039
```

The means of the explanatory variables x_1 to x_5 are:

```
(bx <- as.vector(colMeans(x)))
[1] 15.10784 45.19608 53.82353 25.62745 21.07843
```

We assume centered data and estimate the coefficients without β_0 which is estimated separately. The covariance matrix $\mathrm{Cov}(\boldsymbol{X})$ of the explanatory variables is

```
(Sxx <- var(x))
           [,1]      [,2]       [,3]       [,4]       [,5]
[1,]  0.6907373  1.494431   3.255412  0.3509804 -0.2966275
[2,]  1.4944314  5.400784  10.155294  1.3545098  1.2843137
[3,]  3.2554118 10.155294 200.668235 65.2729412  4.3141176
[4,]  0.3509804  1.354510  65.272941 58.1584314  1.0298039
[5,] -0.2966275  1.284314   4.314118  1.0298039 18.1537255
```

The covariance $\mathrm{Cov}(\boldsymbol{y}, \boldsymbol{X})$ between the response and the explanatory variables is

```
(syx <- as.vector(var(y,x)))
[1]   1.878157   5.663922 108.496471   1.672549   3.421569
```

We now compute

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{y} \, . \tag{5.69}$$

for the centered data, i.e. we assume that $\boldsymbol{X}$ is centered. In this case $\left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} = 1/n \, (\mathrm{Cov}(\boldsymbol{X}))^{-1}$ is the inverse of the covariance matrix divided by the number of samples n. $\boldsymbol{X}^T \boldsymbol{y} = n\mathrm{Cov}(\boldsymbol{y}, \boldsymbol{X})$ is the covariance between the response and the explanatory variables multiplied by the number of samples n. Since the number of samples n cancel, we have

$$\left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{y} = (\mathrm{Cov}(\boldsymbol{X}))^{-1} \mathrm{Cov}(\boldsymbol{y}, \boldsymbol{X}) \, . \tag{5.70}$$

Therefore the least squares estimate can be computed as:

```
(bbeta <- solve(Sxx)%*%syx)
            [,1]
[1,] -0.21318219
[2,] -0.28884109
[3,]  0.85984756
[4,] -0.92921309
[5,]  0.05380269
```

We assumed centered data. Now we estimate β_0 using the mean of the response $\bar{y}$ and the mean of the explanatory variables:

```
(bbeta0 <- by-t(syx)%*%solve(Sxx)%*%bx)
         [,1]
[1,] 15.65486
```

In our derivation of the least squares estimator, we used the formula

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} \, , \tag{5.71}$$

where the first column of $\boldsymbol{X}$ contains 1's to account for the intercept. Therefore the least squares estimate is:

```
x1 <- cbind(rep(1,51),x)
(b1 <- solve(crossprod(x1))%*%t(x1)%*%y)
            [,1]
[1,] 15.65485611
[2,] -0.21318219
[3,] -0.28884109
[4,]  0.85984756
[5,] -0.92921309
[6,]  0.05380269
```

This is the same result as previously, where we first estimated the parameter for the centered data and then adjusted β_0.

$$S(\hat{\boldsymbol{\beta}}) = \boldsymbol{y}^T\boldsymbol{y} - \hat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{y} \, . \tag{5.72}$$

The estimate for the error variance s^2 is

$$s^2 = \frac{1}{n-m-1}S(\hat{\boldsymbol{\beta}}) \tag{5.73}$$

where $n = 51$ and $m = 5$ in our example. We compute s^2 and the standard error s:

```
(s2 <- (crossprod(y)-t(b1)%*%t(x1)%*%y)/(51-6))
       [,1]
[1,] 4.3729
sqrt(s2)
         [,1]
[1,] 2.091148
```

The coefficient of determination R^2 is

$$R^2 \;=\; \frac{\sum_{i=1}^{n} \left(\hat{y}_i \;-\; \overline{y}\right)^2}{\sum_{i=1}^{n} \left(y_i \;-\; \overline{y}\right)^2} \;, \tag{5.74}$$

which we compute by

```
fitted <- x1%*%b1
(R2 <- var(fitted)/var(y))
          [,1]
[1,] 0.9580513
```

R^2 is the variance of the estimated response divided by the variance of the response.

The approximate two-sided confidence intervals for components of the vector $\hat{\boldsymbol{\beta}}$ are:

$$\beta_j \;\in\; \left[\; \hat{\beta}_j \;\pm\; t_{\alpha/2,n-m-1} \; s \; \sqrt{\left[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\right]_{jj}} \;\right] \tag{5.75}$$

where $t_{\alpha/2,n-m-1}$ is the upper $\alpha/2$ percentage point of the central t-distribution and α is the desired significance level of the test. In R we compute these confidence intervals as:

```
bup <- b1 - qt(0.025,45)*s*sqrt(diag(solve(crossprod(x1))))
blow <- b1 + qt(0.025,45)*s*sqrt(diag(solve(crossprod(x1))))
cbind(blow,bup)
            [,1]       [,2]
[1,]  3.03587336 28.2738389
[2,] -1.40187932  0.9755149
[3,] -0.71833021  0.1406480
[4,]  0.80366905  0.9160261
[5,] -1.02844916 -0.8299770
[6,] -0.09389755  0.2015029
```

Only for the intercept (component 1), x_3 (component 4), and x_4 (component 5), the confidence intervals do not include zero.

For testing whether the components of the estimated parameter vector are significantly different from zero, we compute the t-statistics:

$$t_j \;=\; \frac{\hat{\beta}_j}{s\,\sqrt{\left[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\right]_{jj}}} \tag{5.76}$$

In R we compute the t-statistics as

```
(t <- b1/(s*sqrt(diag(solve(crossprod(x1))))))
            [,1]
[1,]   2.4986561
[2,]  -0.3612114
[3,]  -1.3545299
[4,]  30.8271243
[5,] -18.8593854
[6,]   0.7336764
```

These t-statistics together with $n - m - 1 = 51 - 5 - 1 = 45$ degrees of freedom allow to compute the p-values:

```
2*pt(-abs(t),45)
                [,1]
[1,] 1.618559e-02
[2,] 7.196318e-01
[3,] 1.823298e-01
[4,] 6.694743e-32
[5,] 5.395732e-23
[6,] 4.669514e-01
```

Only the intercept, x_3, and x_4 are significant, where the latter two are highly significant.

5.1.7.1.2 Using Predefined R Functions. We now do the same estimate using the R function `lm()` which is a software for fitting linear models.

```
l1 <- lm(y ~ x)
l1

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)           x1           x2           x3           x4           x5
    15.6549      -0.2132      -0.2888       0.8598      -0.9292       0.0538

anova(l1)
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
x          5 4494.2  898.84  205.55 < 2.2e-16 ***
Residuals 45  196.8    4.37
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


summary(l1)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-5.6860 -0.9580  0.3767  1.0973  4.1742
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 15.65486    6.26531   2.499   0.0162 *
x1          -0.21318    0.59019  -0.361   0.7196
x2          -0.28884    0.21324  -1.355   0.1823
x3           0.85985    0.02789  30.827   <2e-16 ***
x4          -0.92921    0.04927 -18.859   <2e-16 ***
x5           0.05380    0.07333   0.734   0.4670
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.091 on 45 degrees of freedom
Multiple R-squared:  0.9581,    Adjusted R-squared:  0.9534
F-statistic: 205.5 on 5 and 45 DF,  p-value: < 2.2e-16
```

Only x_3 and x_4 are significant. The t-statistics and the p-values agree exactly with the values that we have computed. The “Residual standard error: 2.091” agrees to our estimate as does the R^2 value of 0.9581.

The confidence intervals can be assessed by the R function `confint()`:

```
confint(l1)
                  2.5 %     97.5 %
(Intercept)  3.03587336 28.2738389
x1          -1.40187932  0.9755149
x2          -0.71833021  0.1406480
x3           0.80366905  0.9160261
x4          -1.02844916 -0.8299770
x5          -0.09389755  0.2015029
```

We again see that these values agree with the values, that we have computed.

We assess the AIC (Akaike information criterion) which is used to compare models. The model with the largest values is most suited for the data.

```
extractAIC(l1)
[1]  6.00000 80.86343

drop1(l1, test = "F")
Single term deletions

Model:
y ~ x
       Df Sum of Sq    RSS     AIC F value    Pr(>F)
<none>               196.8  80.863
x       5    4494.2 4691.0 232.600  205.55 < 2.2e-16 ***
---
```

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

 drop1(l1, test = "Chisq")
Single term deletions

Model:
y ~ x
       Df Sum of Sq    RSS     AIC  Pr(>Chi)
<none>               196.8  80.863
x       5    4494.2 4691.0 232.600 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

5.1.7.2 Carbohydrate Diet Data

This example is from Dobson [2002], page 96, data of Table 6.3. The data are shown in Tab. 5.3 and contain for twenty male insulin-dependent diabetics: responses, age, weight, and percentages of total calories obtained from complex carbohydrates. The individuals had been on a high-carbohydrate diet for six months. Compliance with the regime was thought to be related to age (in years), body weight (relative to “ideal” weight for height) and other components of the diet, such as the percentage of calories as protein. These other variables are treated as explanatory variables.

We fitted a normal linear model by least squares via following R code:

```
calorie <- data.frame(
+     carb = c(33,40,37,27,30,43,34,48,30,38,
+      50,51,30,36,41,42,46,24,35,37),
+     age = c(33,47,49,35,46,52,62,23,32,42,
+      31,61,63,40,50,64,56,61,48,28),
+     wgt = c(100, 92,135,144,140,101, 95,101, 98,105,
+      108, 85,130,127,109,107,117,100,118,102),
+     prot = c(14,15,18,12,15,15,14,17,15,14,
+      17,19,19,20,15,16,18,13,18,14))


summary(lmcal <- lm(carb~age+wgt+prot, data= calorie))

Call:
lm(formula = carb ~ age + wgt + prot, data = calorie)

Residuals:
     Min       1Q   Median       3Q      Max
-10.3424  -4.8203   0.9897   3.8553   7.9087

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 36.96006   13.07128   2.828  0.01213 *
```

Carbohydrate	Age	Weight	Protein
y	x_1	x_2	x_3
33	33	100	14
40	47	92	15
37	49	135	18
27	35	144	12
30	46	140	15
43	52	101	15
34	62	95	14
48	23	101	17
30	32	98	15
38	42	105	14
50	31	108	17
51	61	85	19
30	63	130	19
36	40	127	20
41	50	109	15
42	64	107	16
46	56	117	18
24	61	100	13
35	48	118	18
37	28	102	14

Table 5.3: Dobson's carbohydrate diet data Dobson [2002], page 96, data of Table 6.3. Carbohydrate, age, relative weight, and protein for twenty male insulin-dependent diabetics.

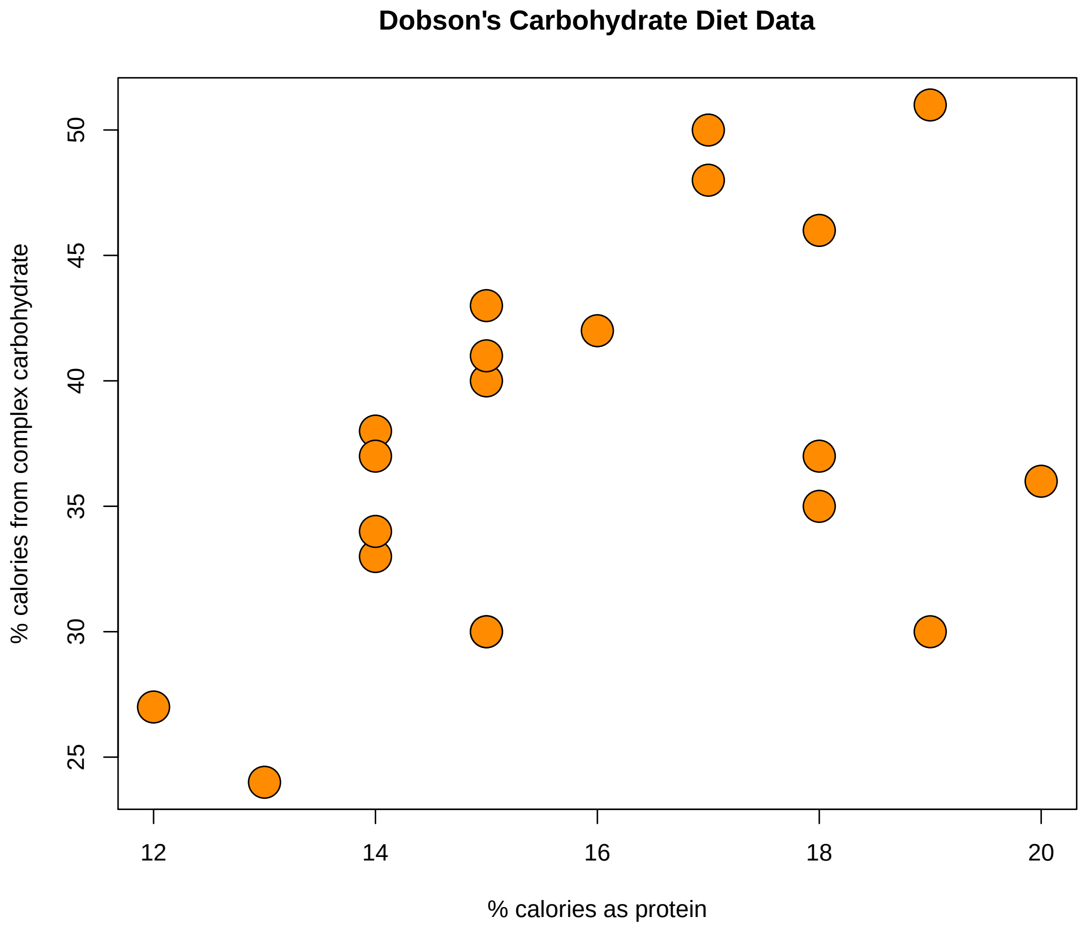

Figure 5.2: Dobson's carbohydrate diet data Dobson [2002] with percentages of total calories obtained from complex carbohydrates plotted against percentage of calories as protein.

```
age         -0.11368    0.10933  -1.040  0.31389
wgt         -0.22802    0.08329  -2.738  0.01460 *
prot         1.95771    0.63489   3.084  0.00712 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.956 on 16 degrees of freedom
Multiple R-squared:  0.4805,    Adjusted R-squared:  0.3831
F-statistic: 4.934 on 3 and 16 DF,  p-value: 0.01297
```

The feature “Protein” seems to be the feature that is most related to carbohydrates. We verify this by a scatter plot. Fig. 5.2 shows percentages of total calories obtained from complex carbohydrates plotted against percentage of calories as protein. A linear dependence is visible which supports the finding that protein is significantly related to carbohydrate.

5.2 Analysis of Variance

Analysis-of-variance (ANOVA) models apply linear models to compare means of responses to different treatments. The treatments are the levels of one *factor*. Thus, they compare means of different groups which are known a priori. Typically, the results of fitting linear models are analyzed by the variance explained as shown previously. $\boldsymbol{x}$ is neither measured nor a sample but constructed and contains dummy variables, therefore, the matrix $\boldsymbol{X}$ is called *design matrix*. Typically, ANOVA models use more parameters than can be estimated, therefore $\boldsymbol{X}$ may not have full rank. We first consider the case where observations are divided into different groups corresponding to a factor. Then we consider the case where observations can be divided by two ways into different groups, that is, two factors. In this case, besides the treatment, a second factor influences the outcome of a study.

5.2.1 One Factor

The response variable, that is, the dependent variable, has now two indices: the first index gives the group to which the observation belongs and the second index gives the replicate number for this group. The standard case is a treatment-control study, where one group are controls and the other group are the treatments. It is possible to analyze different treatments if they are mutually exclusive.

The response variable is y_{gi} with $y_{11}, y_{12}, \ldots, y_{1n_1}, y_{21}, y_{22}, \ldots, y_{2n_2}, y_{31}, \ldots, y_{Gn_G}$, where the j-th group has n_j replicates and G denotes the number of groups. The model is

$$y_{gi} \ = \ \beta_0 \ + \ \beta_g \ + \ \epsilon_{gi} \ . \tag{5.77}$$

The value β_0 is a constant offset or the mean of group 1 if we force $\beta_1 = 0$. The value β_g is the mean difference to the offset (or group 1). As previously ϵ_{gi} is an additive error term with previously introduced assumptions.

For each group the model uses different parameters, therefore the model equation depends on the group to which the observation belongs. The model equations are written down as a matrix equation. For example, in a case-control study with 3 controls and 3 cases, we write:

$$\begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{23} \end{pmatrix} \ = \ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} \ + \ \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{23} \end{pmatrix} \ . \tag{5.78}$$

In matrix notation we have the linear model:

$$\boldsymbol{y} \ = \ \boldsymbol{X} \, \boldsymbol{\beta} \ + \ \boldsymbol{\epsilon} \ , \tag{5.79}$$

where $\boldsymbol{X}$ is designed depending on the groups to which the observations $\boldsymbol{y}$ belong.

However $\boldsymbol{X}$ has lower rank than parameters. In the example above, the 6×3 matrix has rank 2 because rows are identical (or first column is sum of other two). The least squares estimator cannot be computed because $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ does not exist. The model is not identifiable, that is, for every data set there exists more than one solution. For example, we can subtract δ from β_0 and, at the same time, add δ to β_1 and β_2. The solution will not change, only the parameters.

There are different ways to ensure that $\boldsymbol{X}$ has full rank and the least squares estimate can be applied:

(i) *re-parametrization* using fewer parameters, e.g., corner point parametrization,

(ii) *side conditions* as constraints on the parameters, e.g., sum-to-zero constraints,

(iii) *linear projections* $\boldsymbol{a}^T\boldsymbol{\beta}$ of parameter vector $\boldsymbol{\beta}$ which is estimable.

ad (i) re-parametrization:
We assume that β_0 is the mean response of the controls and β_g is the offset of group g to the controls. Therefore we set $\beta_1 = 0$ because controls have zero offset to themselves. We obtain:

$$\begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{23} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{23} \end{pmatrix} . \tag{5.80}$$

Setting $\beta_1 = 0$ is called *corner point parametrization* which removes β_1 from the equations. In general corner point parametrization removes all variables that contain the index one. This means that variables that contain the index one are considered as reference groups or as reference group combinations.

In general, the re-parametrization is

$$\boldsymbol{\gamma} = \boldsymbol{U}\,\boldsymbol{\beta} \tag{5.81}$$

which gives with

$$\boldsymbol{X} = \boldsymbol{Z}\,\boldsymbol{U} \tag{5.82}$$

$$\boldsymbol{y} = \boldsymbol{Z}\,\boldsymbol{\gamma} + \boldsymbol{\epsilon} . \tag{5.83}$$

The matrix $\boldsymbol{Z}$ has full rank and $\boldsymbol{U}$ blows $\boldsymbol{Z}$ up to $\boldsymbol{X}$, therefore, $\boldsymbol{Z}$ and $\boldsymbol{X}$ have the same rank.

ad (ii) side conditions:
We can assume that $\beta_1 + \beta_2 = 0$. If group 1 and group 2 have the same number of replicates, then β_0 is the mean over all groups. From the condition $\beta_1 + \beta_2 = 0$ we immediately obtain $\beta_2 = -\beta_1$. This gives the matrix equation

$$\begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{23} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{23} \end{pmatrix} . \tag{5.84}$$

Variable β_2 is removed from these equations.

The constraint $\sum_{g=1}^{G} \beta_g = 0$ is the *sum-to-zero constraint.* This ensures that β_0 is the overall mean, because $\frac{1}{G} \sum_{g=0}^{G} \beta_g = \frac{1}{G}\beta_0$. The β_g estimate the deviation of the mean of group g from the overall mean. In general sum-to-zero constraints set sums over an index to zero and, thereby, define the constant offset as the overall mean.

ad (iii) linear projection:
$\boldsymbol{a} = (0, 1, -1))$ gives $\beta_1' = \boldsymbol{a}^T \boldsymbol{\beta} = \beta_1 - \beta_2$, which is estimable. This approach is of interest, if specific questions have to be answered. In our example, the difference of the means of group 1 and group 2 may be relevant but not the means themselves. We obtain the matrix equation:

$$\begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{23} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \beta_1' + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{23} \end{pmatrix} . \tag{5.85}$$

The models can be used to test hypotheses. A common null hypothesis is H_0: $\beta_1 = \beta_2 = \ldots = \beta_G$, where the null hypothesis states that the means of all groups are equal. This can be expressed by the new variables $\beta_1^* = \beta_1 - \beta_2, \beta_2^* = \beta_1 - \beta_3, \ldots, \beta_{G-1}^* \beta_1 - \beta_G$, which are tested for $\beta_1^* = \beta_2^* = \ldots = \beta_{G-1}^* = 0$. Or we introduce the constraint $\sum_{g=1}^{G} \beta_g = 0$ while keeping β_0. We then can test for $\beta_1 = \beta_2 = \ldots = \beta_G = 0$, that is, deviation from the overall mean β_0. Tests for these hypotheses have been presented earlier. The reduced model has only the overall mean β_0 as parameter.

5.2.2 Two Factors

We now consider the case where two factors influence the response. Consequently, the response variable has now three indices: the first index gives the group for the first factor, the second index the group for the second factor, and the third index gives the replicate number for this combination of groups.

The response variable is y_{ghi} with the model

$$y_{ghi} = \beta_0 + \beta_g + \alpha_h + (\alpha\beta)_{gh} + \epsilon_{ghi} . \tag{5.86}$$

The value β_0 is a constant offset. The values β_g are the mean difference for the first factor and α_h the mean differences for the second factor. The new term $(\alpha\beta)_{gh}$ models the *interaction effects* between the two factors. As always, ϵ_{ghi} is the additive error with previously introduced assumptions.

The following hypotheses are often tested and correspond to different reduced models:

(i) the *additive model* with the hypothesis H_0: $(\alpha\beta)_{gh} = 0$ for all g and all h:

$$y_{ghi} = \beta_0 + \beta_g + \alpha_h + \epsilon_{ghi} . \tag{5.87}$$

This model should be compared to the full model.

(ii) factor corresponding to α has no effect:

$$y_{ghi} \;=\; \beta_0 \;+\; \beta_g \;+\; \epsilon_{ghi} \; . \tag{5.88}$$

This model should be compared to the additive model in (i).

(iii) factor corresponding to β has no effect:

$$y_{ghi} \;=\; \beta_0 \;+\; \alpha_h \;+\; \epsilon_{ghi} \; . \tag{5.89}$$

As for the model in (ii), also this model should be compared to the additive model in (i).

These models should be either tested with *sum-zero constraints*

(i) $\sum_{g=1}^{G} \beta_g = 0$,

(ii) $\sum_{h=1}^{H} \alpha_h = 0$,

(iii) $\forall_g : \; \sum_{h=1}^{H} (\alpha\beta)_{gh} = 0$,

(iv) $\forall_h : \; \sum_{g=1}^{G} (\alpha\beta)_{gh} = 0$,

or with *corner point constraints*

(i) $\beta_1 = 0$,

(ii) $\alpha_1 = 0$,

(iii) $\forall_g : \; (\alpha\beta)_{g1} = 0$,

(iv) $\forall_h : \; (\alpha\beta)_{1h} = 0$.

We have one offset parameter β_0, G factor parameters β_g, H factor parameters α_h, and GH interaction parameters $(\alpha\beta)_{gh}$, which sums up to $GH + G + H + 1 = (G+1)(H+1)$ parameters. The minimal data set has only GH observations, one observation for each combination of factors. For both sets of constraints the β_g equations use up one degree of freedom, the α_h use also up one degree of freedom, the $(\alpha\beta)_{gh}$ equations for all g use up G degrees of freedom, and the $(\alpha\beta)_{gh}$ equations for all h use up H degrees of freedom. We have to add one degree of freedom because for corner point constraints $(\alpha\beta)_{11}$ is counted twice and for sum-zero constraints the last equation follows from the other equations. We have $1 + 1 + G + H - 1 = G + H + 1$ degrees of freedom used up. Therefore we have $(G+1)(H+1) - (G+H+1) = GH$ free parameters. For sum-zero constraints we show that the last equation follows from the others. From $\forall_g : \; \sum_{h=1}^{H} (\alpha\beta)_{gh} = 0$ follows that $\sum_{g=1}^{G} \sum_{h=1}^{H} (\alpha\beta)_{gh} = 0$. We have $\sum_{h=1}^{H} (\sum_{g=1}^{G} (\alpha\beta)_{gh}) = 0$ and $\sum_{g=1}^{G} (\alpha\beta)_{gh} = 0$ for $h < H$ since the last equation is not used. Thus, $\sum_{g=1}^{G} (\alpha\beta)_{gH} = 0$, which is the last equation. We showed that the last equation can be deduced from the others. Therefore for both constraint sets we have GH free parameters, as desired.

The design matrix $\boldsymbol{X}$ should have at least rank GH to distinguish all interaction effects $(\alpha\beta)_{gh}$. Thus, the least squares estimator can be computed and the according tests performed.

To simplify notations, means are denoted by

(i) mean of group combination gh:

$$\bar{y}_{gh} \;=\; \frac{1}{n_{gh}} \sum_{i=1}^{n_{gh}} y_{ghi} \;, \tag{5.90}$$

where n_{gh} are the number of replicates of group combination gh.

(ii) mean of group g:

$$\bar{y}_{g.} \;=\; \frac{1}{\sum_{h=1}^{H} n_{gh}} \sum_{h=1}^{H} \sum_{i=1}^{n_{gh}} y_{ghi} \;, \tag{5.91}$$

(iii) mean of group h:

$$\bar{y}_{.h} \;=\; \frac{1}{\sum_{g=1}^{G} n_{gh}} \sum_{g=1}^{G} \sum_{i=1}^{n_{gh}} y_{ghi} \;, \tag{5.92}$$

(iv) overall mean:

$$\bar{y}_{..} \;=\; \frac{1}{\sum_{g,h=1,1}^{G,H} n_{gh}} \sum_{g=1}^{G} \sum_{h=1}^{H} \sum_{i=1}^{n_{gh}} y_{ghi} \;. \tag{5.93}$$

If we use the full design matrix $\boldsymbol{X}$ then the *normal equations* are

$$\boldsymbol{X}^T \boldsymbol{X} \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\alpha} \\ (\boldsymbol{\alpha\beta}) \end{pmatrix} \;=\; \boldsymbol{X}^T \, \boldsymbol{y} \;, \tag{5.94}$$

where β_0 is the first component of $\boldsymbol{\beta}$. The matrix $\boldsymbol{X}^T\boldsymbol{X}$ is not invertible. However for the optimal solution $(\hat{\boldsymbol{\beta}}^T, \hat{\boldsymbol{\alpha}}^T, (\hat{\boldsymbol{\alpha}}\hat{\boldsymbol{\beta}})^T)^T$ with sum-zero constraints or with corner point constraints the normal equations must hold.

The normal equations can be written as:

$$\begin{aligned}
&\left(\sum_{g,h=1,1}^{G,H} n_{gh}\right) \hat{\beta}_0 \;+\; \sum_{g=1}^{G} \left(\sum_{h=1}^{H} n_{gh}\right) \hat{\beta}_g \;+\; \sum_{h=1}^{H} \left(\sum_{g=1}^{G} n_{gh}\right) \hat{\alpha}_h \;+\; \sum_{g=1}^{G} \sum_{h=1}^{H} n_{gh} (\hat{\alpha}\hat{\beta})_{gh} \\
&= \sum_{g,h=1,1}^{G,H} n_{gh} \, \bar{y}_{..} \\
&\left(\sum_{h=1}^{H} n_{gh}\right) \hat{\beta}_0 \;+\; \left(\sum_{h=1}^{H} n_{gh}\right) \hat{\beta}_g \;+\; \sum_{h=1}^{H} n_{gh} \, \hat{\alpha}_h \;+\; \sum_{h=1}^{H} n_{gh} (\hat{\alpha}\hat{\beta})_{gh} \;=\; \sum_{h=1}^{H} n_{gh} \, \bar{y}_{g.}, \; 1 \leq g \leq G \\
&\left(\sum_{g=1}^{G} n_{gh}\right) \hat{\beta}_0 \;+\; \sum_{g=1}^{G} n_{gh} \, \hat{\beta}_g \;+\; \left(\sum_{g=1}^{G} n_{gh}\right) \hat{\alpha}_h \;+\; \sum_{g=1}^{G} n_{gh} (\hat{\alpha}\hat{\beta})_{gh} \;=\; \sum_{g=1}^{G} n_{gh} \, \bar{y}_{.h}, \; 1 \leq h \leq H \\
&n_{gh} \, \hat{\beta}_0 \;+\; n_{gh} \, \hat{\beta}_g \;+\; n_{gh} \, \hat{\alpha}_h \;+\; n_{gh} (\hat{\alpha}\hat{\beta})_{gh} \;=\; n_{gh} \bar{y}_{gh}, \;\; 1 \leq g \leq G, \;\; 1 \leq h \leq H \;\; .
\end{aligned} \tag{5.95}$$

These are $1 + G + H + GH = (G + 1)(H + 1)$ equations but in the worst case we have only GH observations. The constraints use up $G + H + 1$ degrees of freedom, e.g. via the zero sum conditions

$$\sum_{g=1}^{G} \hat{\beta}_g \; = \; 0 \, , \tag{5.96}$$

$$\sum_{h=1}^{H} \hat{\alpha}_h \; = \; 0 \, , \tag{5.97}$$

$$\sum_{g=1}^{G} (\hat{\alpha}\hat{\beta})_{gh} \; = \; 0 \, , \tag{5.98}$$

$$\sum_{h=1}^{H} (\hat{\alpha}\hat{\beta})_{gh} \; = \; 0 \, . \tag{5.99}$$

We then have at least GH observations and GH free parameters and the normal equations can be solved.

For the *balanced case* the number of replicates is the same for each combination of conditions. That means

$$n_{gh} \; = \; \tilde{n} \, . \tag{5.100}$$

In this case the means simplify to:

(i) mean of group combination gh:

$$\bar{y}_{gh} \; = \; \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} y_{ghi} \tag{5.101}$$

(ii) mean of group g:

$$\bar{y}_{g.} \; = \; \frac{1}{H \, \tilde{n}} \sum_{h=1}^{H} \sum_{i=1}^{\tilde{n}} y_{ghi} \, , \tag{5.102}$$

(iii) mean of group h:

$$\bar{y}_{.h} \; = \; \frac{1}{G \, \tilde{n}} \sum_{g=1}^{G} \sum_{i=1}^{\tilde{n}} y_{ghi} \, , \tag{5.103}$$

(iv) overall mean:

$$\bar{y}_{..} \; = \; \frac{1}{G \, H \, \tilde{n}} \sum_{g=1}^{G} \sum_{h=1}^{H} \sum_{i=1}^{\tilde{n}} y_{ghi} \, . \tag{5.104}$$

The normal equations become:

$$\begin{aligned}
&G\,H\,\tilde{n}\,\hat{\beta}_0 \;+\; H\,\tilde{n}\,\sum_{g=1}^{G}\,\hat{\beta}_g \;+\; G\,\tilde{n}\,\sum_{h=1}^{H}\,\hat{\alpha}_h \;+\; \tilde{n}\,\sum_{g=1}^{G}\sum_{h=1}^{H}(\hat{\alpha}\hat{\beta})_{gh} \;=\; G\,H\,\tilde{n}\,\bar{y}_{..}\\
&H\,\tilde{n}\,\hat{\beta}_0 \;+\; H\,\tilde{n}\,\hat{\beta}_g \;+\; \tilde{n}\,\sum_{h=1}^{H}\hat{\alpha}_h \;+\; \tilde{n}\,\sum_{h=1}^{H}(\hat{\alpha}\hat{\beta})_{gh} \;=\; H\,\tilde{n}\,\bar{y}_{g.},\;\; 1 \leq g \leq G\\
&G\,\tilde{n}\,\hat{\beta}_0 \;+\; \tilde{n}\,\sum_{g=1}^{G}\hat{\beta}_g \;+\; G\,\tilde{n}\,\hat{\alpha}_h \;+\; \tilde{n}\sum_{g=1}^{G}(\hat{\alpha}\hat{\beta})_{gh} \;=\; G\,\tilde{n}\,\bar{y}_{.h},\;\; 1 \leq h \leq H\\
&\tilde{n}\,\hat{\beta}_0 \;+\; \tilde{n}\,\hat{\beta}_g \;+\; \tilde{n}\,\hat{\alpha}_h \;+\; \tilde{n}\,(\hat{\alpha}\hat{\beta})_{gh} \;=\; \tilde{n}\,\bar{y}_{gh},\;\; 1 \leq g \leq G,\;\; 1 \leq h \leq H\;\;. \qquad (5.105)
\end{aligned}$$

Using the zero sum conditions

$$\sum_{g=1}^{G}\hat{\beta}_g \;=\; 0 \qquad (5.106)$$

$$\sum_{h=1}^{H}\hat{\alpha}_h \;=\; 0 \qquad (5.107)$$

$$\sum_{g=1}^{G}(\hat{\alpha}\hat{\beta})_{gh} \;=\; 0 \qquad (5.108)$$

$$\sum_{h=1}^{H}(\hat{\alpha}\hat{\beta})_{gh} \;=\; 0 \qquad (5.109)$$

the normal equations further simplify to

$$\begin{aligned}
&G\,H\,\tilde{n}\,\hat{\beta}_0 \;=\; G\,H\,\tilde{n}\,\bar{y}_{..}\\
&H\,\tilde{n}\,\hat{\beta}_0 \;+\; H\,\tilde{n}\,\hat{\beta}_g \;=\; H\,\tilde{n}\,\bar{y}_{g.},\;\; 1 \leq g \leq G\\
&G\,\tilde{n}\,\hat{\beta}_0 \;+\; G\,\tilde{n}\,\hat{\alpha}_h \;=\; G\,\tilde{n}\,\bar{y}_{.h},\;\; 1 \leq h \leq H\\
&\tilde{n}\,\hat{\beta}_0 \;+\; \tilde{n}\,\hat{\beta}_g \;+\; \tilde{n}\,\hat{\alpha}_h \;+\; \tilde{n}\,(\hat{\alpha}\hat{\beta})_{gh} \;=\; \tilde{n}\,\bar{y}_{gh},\;\; 1 \leq g \leq G,\;\; 1 \leq h \leq H\;\;, \qquad (5.110)
\end{aligned}$$

which gives

$$\begin{aligned}
\hat{\beta}_0 \;&=\; \bar{y}_{..}\\
\hat{\beta}_g \;&=\; \bar{y}_{g.} \;-\; \hat{\beta}_0 \;=\; \bar{y}_{g.} \;-\; \bar{y}_{..},\;\; 1 \leq g \leq G\\
\hat{\alpha}_h \;&=\; \bar{y}_{.h} \;-\; \hat{\beta}_0 \;=\; \bar{y}_{.h} \;-\; \bar{y}_{..},\;\; 1 \leq h \leq H\\
(\hat{\alpha}\hat{\beta})_{gh} \;&=\; \bar{y}_{gh} \;-\; \hat{\beta}_0 \;-\; \hat{\beta}_g \;-\; \hat{\alpha}_h\\
&=\; \bar{y}_{gh} \;-\; \bar{y}_{g.} \;-\; \bar{y}_{.h} \;+\; \bar{y}_{..}\;. \qquad (5.111)
\end{aligned}$$

These are the estimators for the means which one would use intuitively. Actually these are unbiased estimators for the according means.

Treatment group		Control group	
4.81	5.36	4.17	4.66
4.17	3.48	3.05	5.58
4.41	4.69	5.18	3.66
3.59	4.44	4.01	4.50
5.87	4.89	6.11	3.90
3.83	4.71	4.10	4.61
6.03	5.48	5.17	5.62
4.98	4.32	3.57	4.53
4.90	5.15	5.33	6.05
5.75	6.34	5.59	5.14

Table 5.4: Weights of dried plants which were grown under two conditions. The data are from Dobson [2002], page 46, data of Table 2.7.

5.2.3 Examples

5.2.3.1 Dried Plant Weights

The first example is from Dobson [2002], page 46, data from Table 2.7. Genetically similar seeds are randomly assigned to be raised in either a nutritionally enriched environment (treatment group) or standard conditions (control group) using a completely randomized experimental design. After a predetermined time, all plants are harvested, dried and weighed. The results, expressed in grams, for 20 plants in each group are shown in Tab. 5.4 and in Fig. 5.3. The goal is to test whether there is a difference in yield between the treatment and the control group.

We perform the analysis in R , therefore first the data set is defined:

```
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.98,4.90,5.75,
+ 5.36,3.48,4.69,4.44,4.89,4.71,5.48,4.32,5.15,6.34)
ctl <- c(4.17,3.05,5.18,4.01,6.11,4.10,5.17,3.57,5.33,5.59,
+ 4.66,5.58,3.66,4.50,3.90,4.61,5.62,4.53,6.05,5.14)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
```

To obtain an overview of the data, we do a simple summary:

```
summary(ctl)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.050   4.077   4.635   4.726   5.392   6.110
summary(trt)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.480   4.388   4.850   4.860   5.390   6.340
```

We see that the treatment has larger median and larger mean. Is this significant? When looking at the data in Fig. 5.3 there could be some doubts.

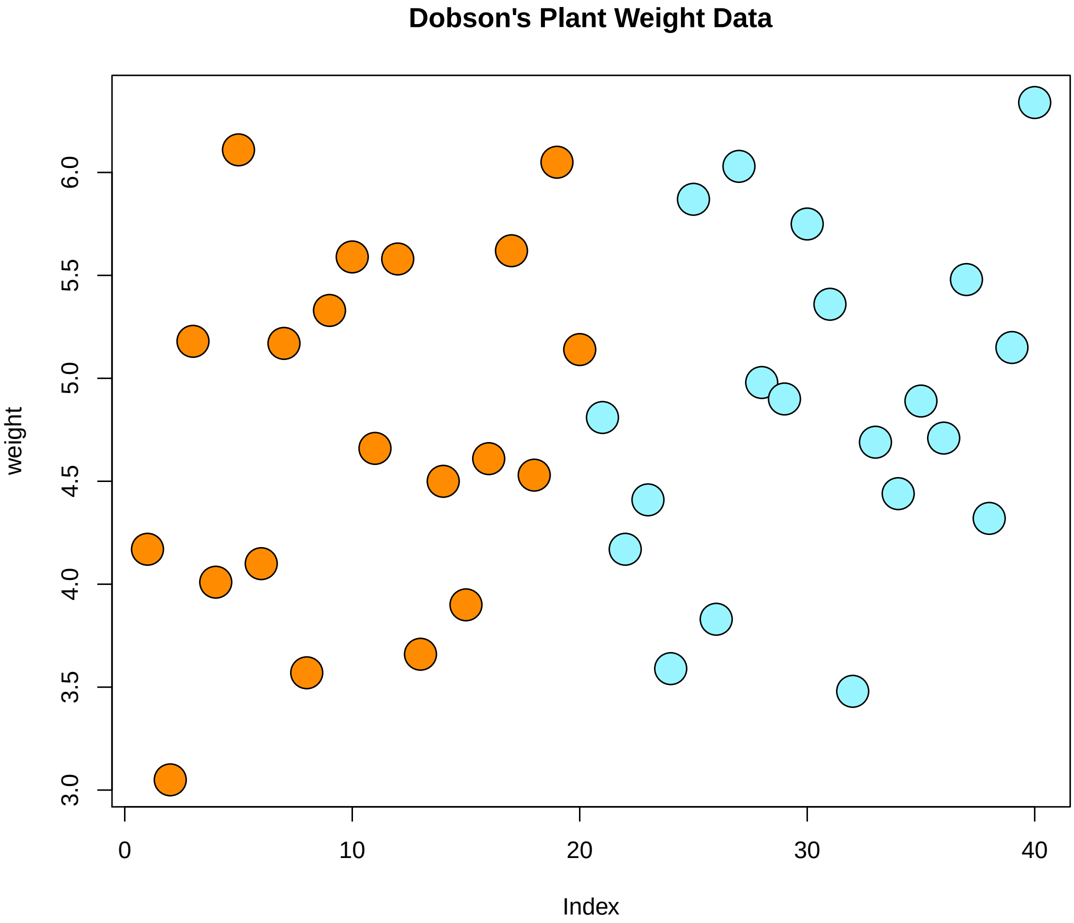

Figure 5.3: Dobson’s dried plant data: orange indicates the control and blue the treatment group.

To answer the question whether the difference in means is significant or not, we fit a linear model and print the ANOVA table:

```
lm.D9 <- lm(weight ~ group)

anova(lm.D9)
Analysis of Variance Table

Response: weight
          Df  Sum Sq Mean Sq F value Pr(>F)
group      1  0.1782 0.17822  0.2599 0.6131
Residuals 38 26.0535 0.68562
```

The difference in means between treatment and control is not significant, i.e. the treatment did not show more or less average yield. We plot the results of the linear model by

```
opar <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))
plot(lm.D9, las = 1)      # Residuals, Fitted, ...
par(opar)
```

and shown them in Fig. 5.4.

Next we fit a model without an intercept

```
lm.D90 <- lm(weight ~ group - 1) # omitting intercept
summary(lm.D90)

Call:
lm(formula = weight ~ group - 1)

Residuals:
     Min       1Q   Median       3Q      Max
-1.67650 -0.57400 -0.05825  0.60763  1.48000

Coefficients:
         Estimate Std. Error t value Pr(>|t|)
groupCtl   4.7265     0.1852   25.53   <2e-16 ***
groupTrt   4.8600     0.1852   26.25   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.828 on 38 degrees of freedom
Multiple R-squared:  0.9724,    Adjusted R-squared:  0.971
F-statistic: 670.3 on 2 and 38 DF,  p-value: < 2.2e-16
```

The intercept is replaced by the groups because always one of them is present. Therefore both groups are significantly different from zero (sure: dried plants have a weight), however there is no difference between the groups.

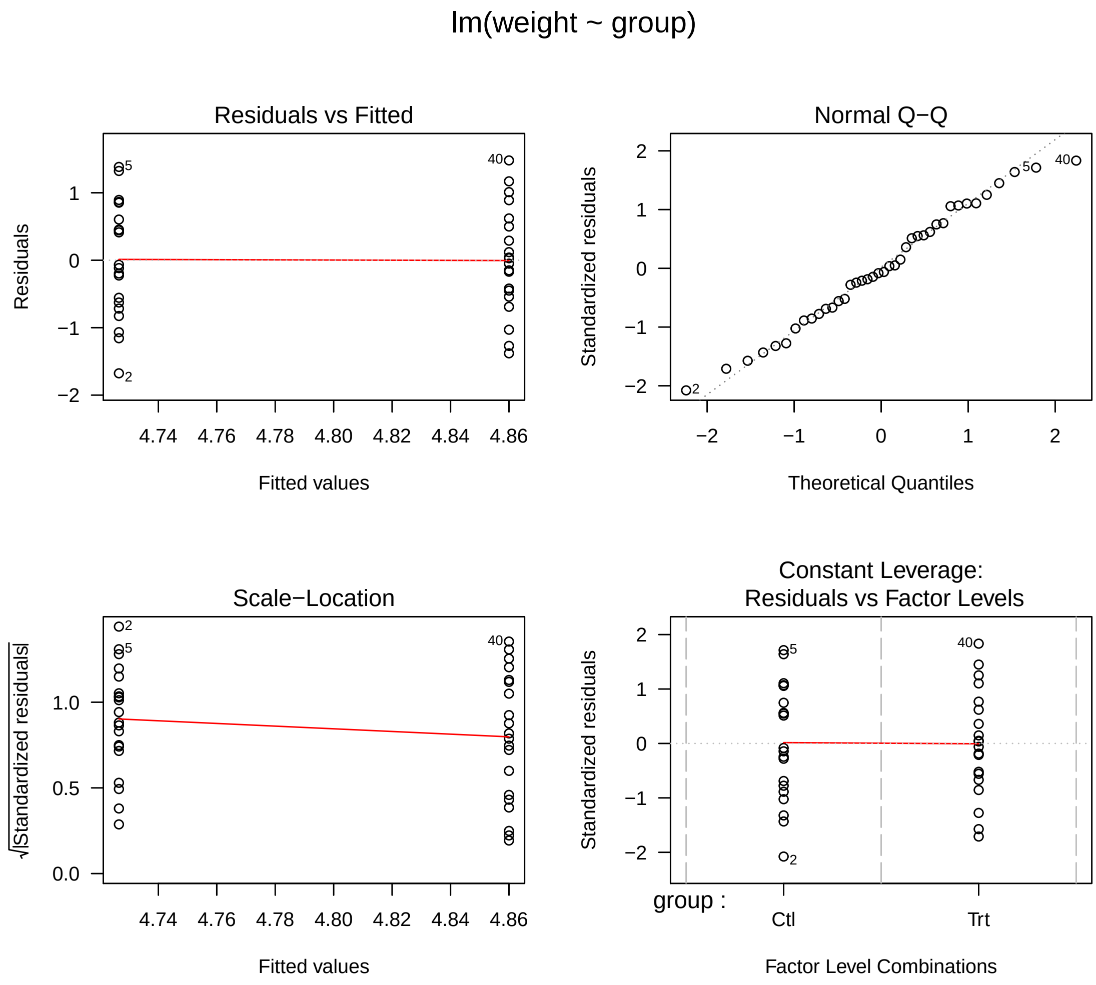

Figure 5.4: Results of ANOVA for dried plant data.

	Control	Treatment A	Treatment B
	4.17	4.81	6.31
	5.58	4.17	5.12
	5.18	4.41	5.54
	6.11	3.59	5.50
	4.50	5.87	5.37
	4.61	3.83	5.29
	5.17	6.03	4.92
	4.53	4.89	6.15
	5.33	4.32	5.80
	5.14	4.69	5.26
$\sum_i y_i$	50.32	46.61	55.26
$\sum_i y_i^2$	256.27	222.92	307.13

Table 5.5: Dried weight of plants grown under three conditions from Dobson [2002], page 101, data of Table 6.6.

5.2.3.2 Extended Dried Plants

The second example extends the first example and is from Dobson [2002], page 101, data of Table 6.6. The results of plant weights in grams for three groups (control, treatment A, treatment B) are shown in Tab. 5.5 and in Fig. 5.5. Plants from treatment B group (green) seem to be larger than the others. We will check whether this impression also holds after fitting a linear model and analyzing the results.

The ANOVA models fitted in R are:

```
ctl <-c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trtA <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
trtB <- c(6.31,5.12,5.54,5.50,5.37,5.29,4.92,6.15,5.80,5.26)
group <- gl(3, length(ctl), labels=c("Ctl","A","B"))
weight <- c(ctl,trtA,trtB)
anova(lmwg <- lm(weight~group))

Analysis of Variance Table

Response: weight
          Df  Sum Sq Mean Sq F value  Pr(>F)
group      2  3.7663  1.8832  4.8461 0.01591 *
Residuals 27 10.4921  0.3886
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


summary(lmwg)
```

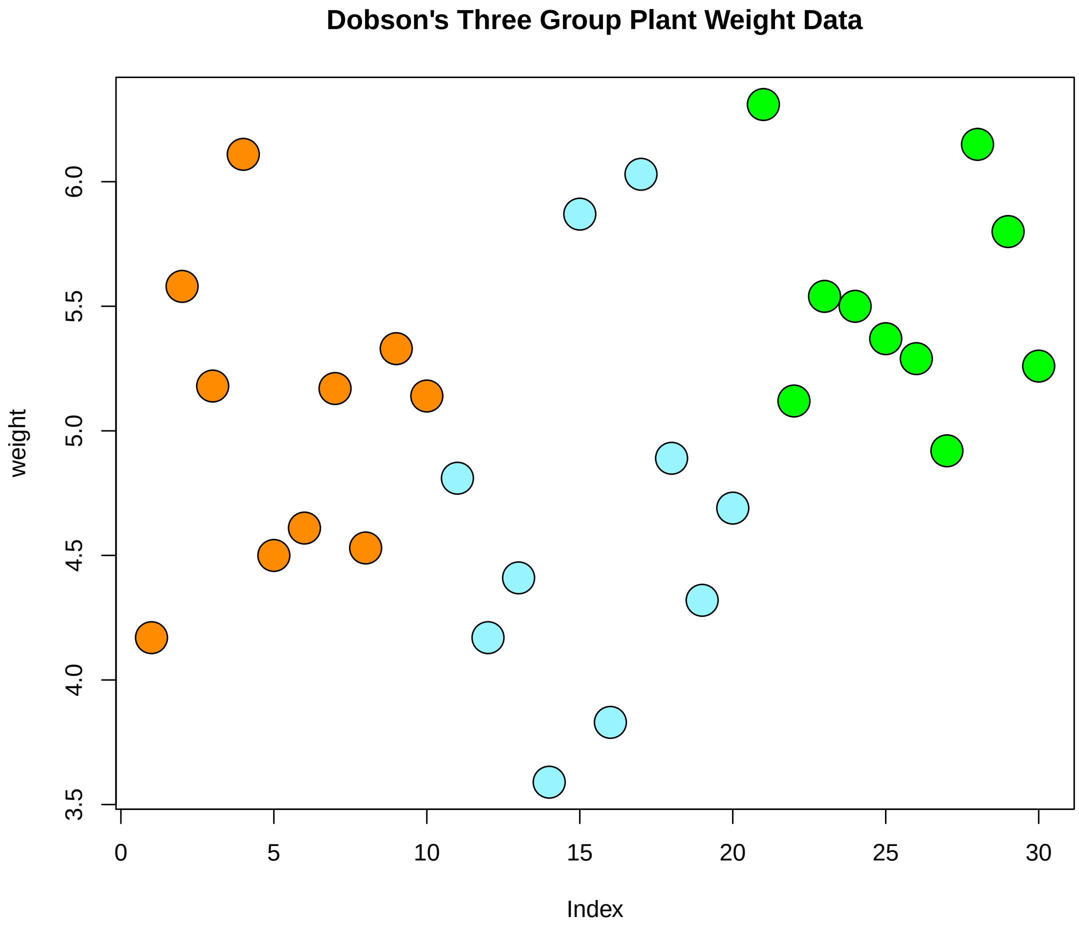

Figure 5.5: Dobson's dried plant data for three groups: orange indicates the control, blue the treatment A, and green treatment B group. Treatment B group seem to be larger than the others.

```
Call:
lm(formula = weight ~ group)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0710 -0.4180 -0.0060  0.2627  1.3690

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.0320     0.1971  25.527   <2e-16 ***
groupA       -0.3710     0.2788  -1.331   0.1944
groupB        0.4940     0.2788   1.772   0.0877 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6234 on 27 degrees of freedom
Multiple R-squared:  0.2641,    Adjusted R-squared:  0.2096
F-statistic: 4.846 on 2 and 27 DF,  p-value: 0.01591

coef(lmwg)
(Intercept)      groupA      groupB
      5.032      -0.371       0.494

coef(summary(lmwg))#- incl.  std.err,  t- and P- values.
            Estimate Std. Error   t value     Pr(>|t|)
(Intercept)    5.032  0.1971284 25.526514 1.936575e-20
groupA        -0.371  0.2787816 -1.330791 1.943879e-01
groupB         0.494  0.2787816  1.771996 8.768168e-02
```

Group B can be distinguished best from other groups. Its coefficient has a p-value of 0.09 which is almost significant. The F-statistic and its p-value of 0.016 shows that the groups together are significant. The estimated parameters show that group B is larger (0.494) and group A smaller (-0.371) than the control group.

5.2.3.3 Two-Factor ANOVA Toy Example

This example for a two-way ANOVA problem is from Dobson [2002], page 106, data of Table 6.9. The fictitious data is shown in Tab. 5.6, where factor A has 3 levels and factor B has 2 levels. This gives $2 \times 3 = 6$ subgroups which form all combinations of A and B levels. Each subgroup has 2 replicates. The data is shown in Fig. 5.6.

Questions for this data set can be:

- are there interaction effects?,
- are there different responses for different levels of factor A?,

	Levels of factor B		
Levels of factor A	B_1	B_2	Total
A_1	6.8, 6.6	5.3, 6.1	24.8
A_2	7.5, 7.4	7.2, 6.5	28.6
A_3	7.8, 9.1	8.8, 9.1	34.8
Total	45.2	43.0	88.2

Table 5.6: Fictitious data for two-factor ANOVA with equal numbers of observations in each subgroup from Dobson [2002].

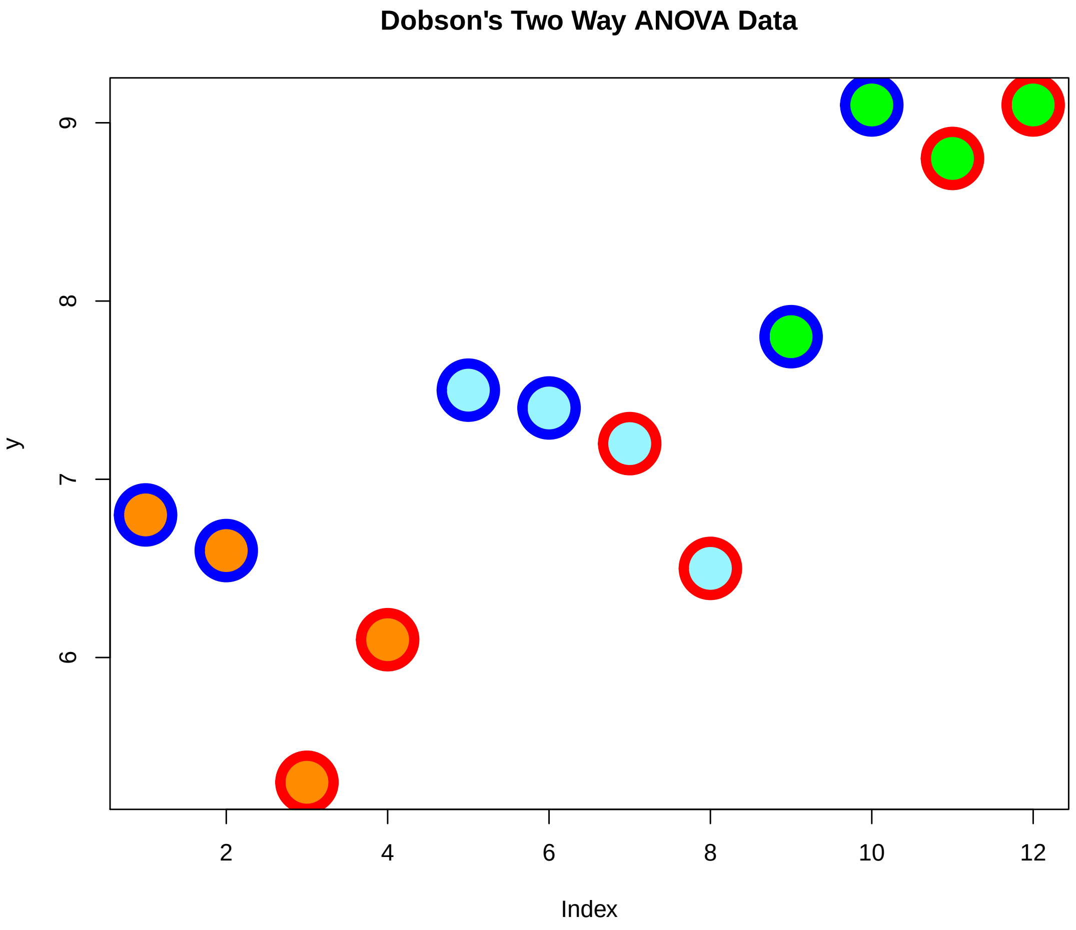

Figure 5.6: Fictitious data for two-factor ANOVA with equal numbers of observations in each subgroup from Dobson [2002]. Levels of factor A are indicated by the interior color of the circles while levels of factor B are indicated by the border color of the circles.

- are there different responses for different levels of factor B?

Each question corresponds to a hypothesis.

We analyze this data in R by an ANOVA table:

```
y <- c(6.8,6.6,5.3,6.1,7.5,7.4,7.2,6.5,7.8,9.1,8.8,9.1)
a <- gl(3,4)
b <- gl(2,2, length(a))
anova(z <- lm(y~a*b))

Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value   Pr(>F)
a          2 12.7400  6.3700 25.8243 0.001127 **
b          1  0.4033  0.4033  1.6351 0.248225
a:b        2  1.2067  0.6033  2.4459 0.167164
Residuals  6  1.4800  0.2467
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The is no evidence against the hypothesis that the levels of factor B do not influence the response. Similar there is no evidence against the hypothesis that the interaction effect does not influence the response. Therefore we conclude that the response is mainly affected by differences in the levels of factor A.

5.3 Analysis of Covariance

5.3.1 The Model

We now consider models that combine covariates (variables or regressors measured together with y) and designed or dummy variables as in the ANOVA models. These models are called analysis of covariance (ANCOVA) models. Thus, we know treatment groups but have also additional measurements. The additional measurements, the covariates, reduce the error variance because some variance is explained by them. Therefore, the unexplained variance is reduced before comparing the means of groups which is supposed to increase the performance of the ANOVA models.

The model is

$$\boldsymbol{y} \;=\; \boldsymbol{X}\boldsymbol{\beta} \;+\; \boldsymbol{Z}\boldsymbol{u} \;+\; \boldsymbol{\epsilon} \,, \tag{5.112}$$

where $\boldsymbol{X}\boldsymbol{b}$ is the same as in the ANOVA model but now the covariate values $\boldsymbol{Z}$ together with their coefficients $\boldsymbol{u}$ are added. The designed $\boldsymbol{X}$ contains zeros and ones while $\boldsymbol{Z}$ contains measured values.

For example, a one-way balanced model with only one covariate is

$$y_{gi} \;=\; \beta_0 \;+\; \beta_g \;+\; u\, z_{gi} \;+\; \epsilon_{gi} \;, 1 \leq g \leq G, \; 1 \leq i \leq \tilde{n} \,, \tag{5.113}$$

where β_g is the treatment effect, z_{gi} is the covariate that was observed together with sample y_{gi}, and u is the coefficient or slope for z_{gi}. With q covariates the model is

$$y_{gi} \;=\; \beta_0 \;+\; \beta_g \;+\; \sum_{r}^{q} u_r\, z_{gir} \;+\; \epsilon_{gi} \;, 1 \leq g \leq G, \; 1 \leq i \leq \tilde{n} \tag{5.114}$$

which is in matrix notation

$$\boldsymbol{Z}\boldsymbol{u} \;=\; \begin{pmatrix} z_{111} & z_{112} & \ldots & z_{11q} \\ z_{121} & z_{122} & \ldots & z_{12q} \\ \vdots & \vdots & & \vdots \\ z_{G\tilde{n}1} & z_{G\tilde{n}2} & \ldots & z_{G\tilde{n}q} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_q \end{pmatrix} \,. \tag{5.115}$$

The matrices $\boldsymbol{X}$ and $\boldsymbol{Z}$ can be combined:

$$\boldsymbol{y} \;=\; (\boldsymbol{X}, \boldsymbol{Z}) \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{u} \end{pmatrix} \;+\; \boldsymbol{\epsilon} \,. \tag{5.116}$$

The normal equations are

$$\begin{pmatrix} \boldsymbol{X}^T \\ \boldsymbol{Z}^T \end{pmatrix} (\boldsymbol{X}, \boldsymbol{Z}) \begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{u}} \end{pmatrix} \;=\; \begin{pmatrix} \boldsymbol{X}^T\boldsymbol{X} & \boldsymbol{X}^T\boldsymbol{Z} \\ \boldsymbol{Z}^T\boldsymbol{X} & \boldsymbol{Z}^T\boldsymbol{Z} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{u}} \end{pmatrix} \;=\; \begin{pmatrix} \boldsymbol{X}^T\boldsymbol{y} \\ \boldsymbol{Z}^T\boldsymbol{y} \end{pmatrix} \,. \tag{5.117}$$

We obtain two equations:

$$\boldsymbol{X}^T\boldsymbol{X}\,\hat{\boldsymbol{\beta}} \;+\; \boldsymbol{X}^T\boldsymbol{Z}\,\hat{\boldsymbol{u}} \;=\; \boldsymbol{X}^T\boldsymbol{y} \tag{5.118}$$

$$\boldsymbol{Z}^T\boldsymbol{X}\,\hat{\boldsymbol{\beta}} \;+\; \boldsymbol{Z}^T\boldsymbol{Z}\,\hat{\boldsymbol{u}} \;=\; \boldsymbol{Z}^T\boldsymbol{y} \,. \tag{5.119}$$

Solving the first equation for $\hat{\boldsymbol{\beta}}$ gives

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\boldsymbol{X}^T\boldsymbol{X})^{+}\boldsymbol{X}^T\boldsymbol{y} - (\boldsymbol{X}^T\boldsymbol{X})^{+}\boldsymbol{X}^T\boldsymbol{Z}\,\hat{\boldsymbol{u}} \\ &= \hat{\boldsymbol{\beta}}_0 - (\boldsymbol{X}^T\boldsymbol{X})^{+}\boldsymbol{X}^T\boldsymbol{Z}\,\hat{\boldsymbol{u}}\,, \end{aligned} \tag{5.120}$$

where $(\boldsymbol{X}^T\boldsymbol{X})^{+}$ denotes the pseudo inverse of $(\boldsymbol{X}^T\boldsymbol{X})$ and $\hat{\boldsymbol{\beta}}_0 = (\boldsymbol{X}^T\boldsymbol{X})^{+}\boldsymbol{X}^T\boldsymbol{y}$ is the solution to the normal equations of the model without covariates.

We now substitute this equation for $\hat{\boldsymbol{\beta}}$ into the second equation in order to solve for $\hat{\boldsymbol{u}}$:

$$\boldsymbol{Z}^T\boldsymbol{X}\,\left((\boldsymbol{X}^T\boldsymbol{X})^{+}\boldsymbol{X}^T\boldsymbol{y} - (\boldsymbol{X}^T\boldsymbol{X})^{+}\boldsymbol{X}^T\boldsymbol{Z}\,\hat{\boldsymbol{u}}\right) + \boldsymbol{Z}^T\boldsymbol{Z}\,\hat{\boldsymbol{u}} = \boldsymbol{Z}^T\boldsymbol{y}\,. \tag{5.121}$$

We define

$$\boldsymbol{P} = \boldsymbol{X}\,(\boldsymbol{X}^T\boldsymbol{X})^{+}\boldsymbol{X}^T \tag{5.122}$$

and obtain for $\hat{\boldsymbol{u}}$:

$$\hat{\boldsymbol{u}} = \left(\boldsymbol{Z}^T(\boldsymbol{I} - \boldsymbol{P})\,\boldsymbol{Z}\right)^{-1}\boldsymbol{Z}^T(\boldsymbol{I} - \boldsymbol{P})\,\boldsymbol{y}\,. \tag{5.123}$$

We immediately obtain a solution for $\hat{\boldsymbol{\beta}}$:

$$\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_0 - (\boldsymbol{X}^T\boldsymbol{X})^{+}\boldsymbol{X}^T\boldsymbol{Z}\,\hat{\boldsymbol{u}}\,. \tag{5.124}$$

Different hypotheses can be tested like H_0: $\beta_1 = \beta_2 = \ldots = \beta_G$ (equality of treatment effects), H_0: $\boldsymbol{u} = \boldsymbol{0}$ (slope equal to zero), or H_0: $u_1 = u_2 = \ldots = u_q$ (equal slopes, homogeneity of slopes) Rencher and Schaalje [2008]. Also two-way models with covariates can be constructed Rencher and Schaalje [2008].

5.3.2 Examples

5.3.2.1 Achievement Scores

The data are from Dobson [2002], page 111, data of Table 6.12. The data are listed in Tab. 5.7 which is originally from Winer (1971), page 776. The responses are achievement scores measured at three levels of a factor representing three different training methods. The covariates are aptitude scores measured before training commenced. We want to compare the training methods, taking into account differences in initial aptitude between the three groups of subjects. The data is plotted in Fig. 5.7, where the data points are jittered to avoid data points covering others.

The figure shows that the achievement scores y increase linearly with aptitude x. Further the achievement scores y are generally higher for training methods B and C if compared to A. We want to test the hypothesis that there are no differences in mean achievement scores among the three training methods, after adjustment for initial aptitude.

```
y <- c(6,4,5,3,4,3,6, 8,9,7,9,8,5,7, 6,7,7,7,8,5,7)
x <- c(3,1,3,1,2,1,4, 4,5,5,4,3,1,2, 3,2,2,3,4,1,4)
m <- gl(3,7)
```

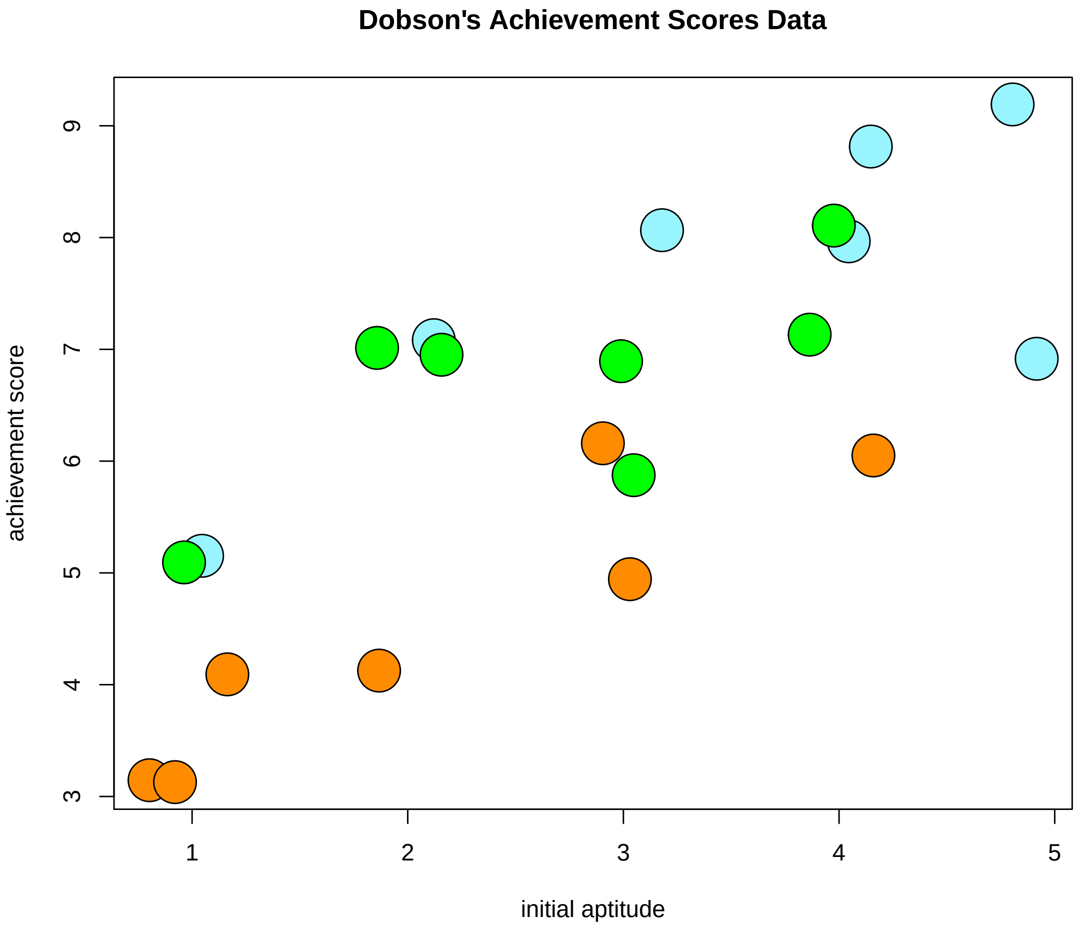

Figure 5.7: Scatter plot of Dobson's achievement scores data. Observations are jittered to avoid data points covering others.

	Training method					
	A		B		C	
	y	x	y	x	y	x
	6	3	8	4	6	3
	4	1	9	5	7	2
	5	3	7	5	7	2
	3	1	9	4	7	3
	4	2	8	3	8	4
	3	1	5	1	5	1
	6	4	7	2	7	4
$\sum x$ / $\sum y$	31	15	53	24	47	19
$\sum x^2$ / $\sum y^2$	147	41	413	96	321	59
$\sum xy$	75		191		132	

Table 5.7: The responses are achievement scores measured at three levels of a factor representing three different training methods. The data is from Dobson [2002] and originally from Winer (1971), p. 776.

```
anova(z <- lm(y~x+m))

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
x          1 36.575  36.575  60.355 5.428e-07 ***
m          2 16.932   8.466  13.970 0.0002579 ***
Residuals 17 10.302   0.606
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Of course, the initial aptitude x is significant for the achievement scores y. More importantly, the training methods, which are given by m, show significant differences concerning the achievement scores. We obtain the same result by looking at the ANOVA table of different models:

```
z0 <- lm(y~x)
anova(z,z0)
Analysis of Variance Table

Model 1: y ~ x + m
Model 2: y ~ x
  Res.Df    RSS Df Sum of Sq     F    Pr(>F)
1     17 10.302
2     19 27.234 -2   -16.932 13.97 0.0002579 ***
```

	Boys		Girls	
	Age	Birthweight	Age	Birthweight
	40	2968	40	3317
	38	2795	36	2729
	40	3163	40	2935
	35	2925	38	2754
	36	2625	42	3210
	37	2847	39	2817
	41	3292	40	3126
	40	3473	37	2539
	37	2628	36	2412
	38	3176	38	2991
	40	3421	39	2875
	38	2975	40	3231
Means	38.33	3024.00	38.75	2911.33

Table 5.8: Birthweight and gestational age for boys and girls from Dobson [2002].

Again we see that the training methods show significant differences after adjusting for the initial aptitude.

5.3.2.2 Birthweights of Girls and Boys

The data set is from Dobson [2002], page 30, data of Table 2.3. Birthweights (in grams) and estimated gestational ages (in weeks) of 12 male and female babies are sampled. Tab. 5.8 shows the data. The mean ages are almost the same for both sexes but the mean birthweight for boys is higher than the mean birthweight for girls. The data are shown in a scatter plot in Fig. 5.8. There is a linear trend of birth weight increasing with gestational age and the girls tend to weigh less than the boys of the same gestational age. The question of interest is whether the rate of increase of birthweight with gestational age is the same for boys and girls.

We analyze the data in R . First we create the data:

```
age <- c(40, 38, 40, 35, 36, 37, 41, 40, 37, 38, 40, 38,
 40, 36, 40, 38, 42, 39, 40, 37, 36, 38, 39, 40)
birthw <- c(2968, 2795, 3163, 2925, 2625, 2847, 3292, 3473, 2628, 3176,
    3421, 2975, 3317, 2729, 2935, 2754, 3210, 2817, 3126, 2539,
    2412, 2991, 2875, 3231)
sex <- gl(2,12, labels=c("Male","Female"))
```

The scatter plot is produced by

```
plot(age, birthw, pch=21,bg=c("cadetblue1","darkorange")[as.numeric(sex)],
+ main="Dobson’s Birth Weight Data",cex=2)
lines(lowess(age[sex==’Male’], birthw[sex==’Male’]),
```

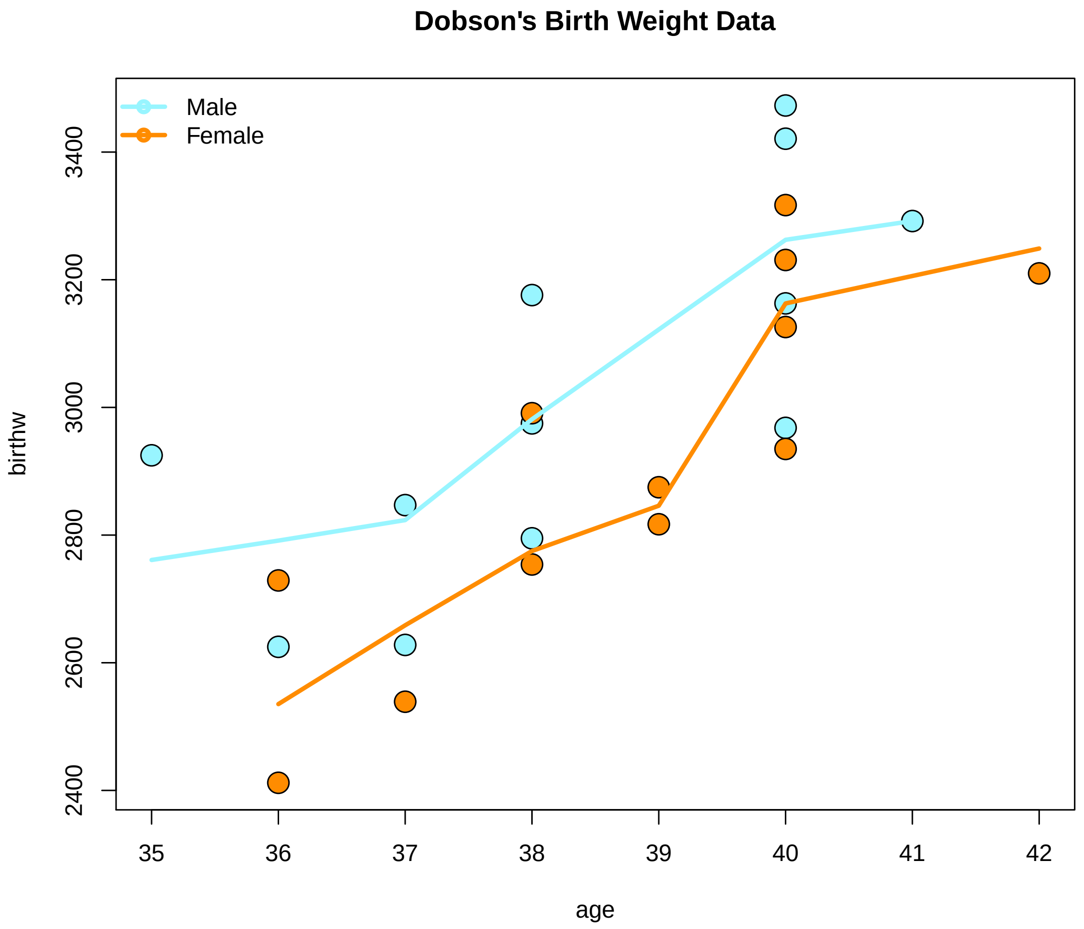

Figure 5.8: Scatter plot of Dobson’s birthweight data. Regression lines are shown.

```
+ col="cadetblue1",lwd=3)
lines(lowess(age[sex=='Female'], birthw[sex=='Female']),
+ col="darkorange",lwd=3)
legend("topleft", levels(sex), col=c("cadetblue1","darkorange"),
+ pch=21, lty=1, bty="n",lwd=3)
```

For analysis we fit a linear model where the groups are male and female and the covariate is the age:

```
summary(l1 <- lm(birthw ~ sex + age), correlation=TRUE)

Call:
lm(formula = birthw ~ sex + age)

Residuals:
    Min      1Q  Median      3Q     Max
-257.49 -125.28  -58.44  169.00  303.98

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1610.28     786.08  -2.049   0.0532 .
sexFemale    -163.04      72.81  -2.239   0.0361 *
age           120.89      20.46   5.908 7.28e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 177.1 on 21 degrees of freedom
Multiple R-squared:   0.64,     Adjusted R-squared:  0.6057
F-statistic: 18.67 on 2 and 21 DF,  p-value: 2.194e-05

Correlation of Coefficients:
          (Intercept) sexFemale
sexFemale  0.07
age       -1.00       -0.12
```

Of course, the birthweight depends on the age, which is highly significant. However also the sex is significant at a level of 0.05. Females weigh less than males as the coefficient for females is -163.04.

The intercept was not important, we fit the model without an intercept:

```
summary(l0 <- lm(birthw ~ sex + age -1), correlation=TRUE)

Call:
lm(formula = birthw ~ sex + age - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-257.49 -125.28  -58.44  169.00  303.98

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
sexMale   -1610.28     786.08  -2.049   0.0532 .
sexFemale -1773.32     794.59  -2.232   0.0367 *
age         120.89      20.46   5.908 7.28e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 177.1 on 21 degrees of freedom
Multiple R-squared:  0.9969,    Adjusted R-squared:  0.9965
F-statistic:  2258 on 3 and 21 DF,  p-value: < 2.2e-16

Correlation of Coefficients:
          sexMale sexFemale
sexFemale  1.00
age       -1.00   -1.00
```

The intercept is now attributed to the males. This is in agreement to the result in previous setting, where the males were the reference group. Either the reference group effect or the constant offset (the intercept) is set to zero.

We compare the models by an ANOVA table:

```
anova(l1,l0)
Analysis of Variance Table

Model 1: birthw ~ sex + age
Model 2: birthw ~ sex + age - 1
  Res.Df    RSS Df  Sum of Sq F Pr(>F)
1     21 658771
2     21 658771  0 1.5134e-09
```

The intercept is not required.

Next we fit a more complex model which contains the interaction of factor sex with variable age:

```
summary(li <- lm(birthw ~ sex + sex:age -1), correlation=TRUE)

Call:
lm(formula = birthw ~ sex + sex:age - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-246.69 -138.11  -39.13  176.57  274.28
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
sexMale       -1268.67    1114.64  -1.138 0.268492
sexFemale     -2141.67    1163.60  -1.841 0.080574 .
sexMale:age     111.98      29.05   3.855 0.000986 ***
sexFemale:age   130.40      30.00   4.347 0.000313 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 180.6 on 20 degrees of freedom
Multiple R-squared:  0.9969,    Adjusted R-squared:  0.9963
F-statistic:  1629 on 4 and 20 DF,  p-value: < 2.2e-16

Correlation of Coefficients:
              sexMale sexFemale sexMale:age
sexFemale      0.00
sexMale:age   -1.00    0.00
sexFemale:age  0.00   -1.00      0.00
```

The interaction terms (interaction of factor sex with variable age) explain significant variance in the data. However the interaction factors are driven by age. Thus, age is now less significant as it is divided into two interaction factors.

The ANOVA table shows

```
anova(li,l0)
Analysis of Variance Table

Model 1: birthw ~ sex + sex:age - 1
Model 2: birthw ~ sex + age - 1
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     20 652425
2     21 658771 -1   -6346.2 0.1945 0.6639
```

The difference between the models is not significant. Only age is separated into the combined factors containing the sex.

5.4 Mixed Effects Models

So far, we considered only the noise $\boldsymbol{\epsilon}$ as random variable given $\boldsymbol{x}$. Thus, only $\boldsymbol{\epsilon}$ could explain the variance of $p(y \mid \boldsymbol{x})$. We now assume there is a second source of variation which is represented by a hidden or latent variable $\boldsymbol{u}$. If the variance of $\boldsymbol{u}$ is not known, then the parameter estimation becomes more complicated. The error variance has to be distinguished from the variance through $\boldsymbol{u}$. So far the parameters could be estimated without knowing the error variance. We assumed that the errors have the same spherical variance. Therefore this variance would factor out in the objective and the normal equations would not change. For mixed effect models that is no longer the case.

For each observation y there is a corresponding latent variable $\boldsymbol{u}$:

$$y \;=\; \boldsymbol{x}^T \boldsymbol{\beta} \;+\; \boldsymbol{z}^T \boldsymbol{u} \;+\; \epsilon \,. \tag{5.125}$$

$\boldsymbol{z}$ is a vector indicating the presence of the latent variable, which can be sampled with y or be designed via dummy variables.

We assume that

$$\begin{aligned} \mathrm{E}(\boldsymbol{u}) \;&=\; \mathbf{0} \,, & (5.126)\\ \mathrm{E}(\epsilon) \;&=\; \mathbf{0} \,, & (5.127)\\ \mathrm{Var}(\boldsymbol{u}) \;&=\; \boldsymbol{G} \,, & (5.128)\\ \mathrm{Var}(\boldsymbol{\epsilon}) \;&=\; \boldsymbol{R} \,, & (5.129)\\ \mathrm{Cov}(\boldsymbol{\epsilon}, \boldsymbol{u}) \;&=\; \mathbf{0} \,. & (5.130) \end{aligned}$$

The model in matrix notation is

$$\boldsymbol{y} \;=\; \boldsymbol{X}\boldsymbol{\beta} \;+\; \boldsymbol{Z}\boldsymbol{u} \;+\; \boldsymbol{\epsilon} \,. \tag{5.131}$$

The design matrix is $\boldsymbol{X}$ with $\boldsymbol{\beta}$ as the coefficient vector of fixed effects. $\boldsymbol{u}$ is the vector of random effects with $\boldsymbol{Z}$ as fixed predictor matrix. $\boldsymbol{Z}$ is often used to specify group memberships or certain measurement conditions.

We have

$$\begin{aligned} \mathrm{E}(\boldsymbol{y}) \;&=\; \boldsymbol{X}\boldsymbol{\beta} \,, & (5.132)\\ \mathrm{Var}(\boldsymbol{y}) \;&=\; \boldsymbol{Z}^T \boldsymbol{G} \boldsymbol{Z} \;+\; \boldsymbol{R} \,. & (5.133) \end{aligned}$$

These properties of $\boldsymbol{y}$ follow immediately from the assumptions.

5.4.1 Approximative Estimator

We want to find an estimator for both $\boldsymbol{\beta}$ and $\boldsymbol{u}$. The estimator for $\boldsymbol{u}$ is the posterior, that is, the distribution of $\boldsymbol{u}$ after having seen the observation, while the prior is the distribution of $\boldsymbol{u}$ without an observation.

5.4.1.1 Estimator for Beta

We assume that both $\boldsymbol{G} = \sigma_u^2 \boldsymbol{I}$ and $\boldsymbol{R} = \sigma^2 \boldsymbol{I}$ are normally distributed. Then, we approximate $\boldsymbol{G}$ by $\hat{\sigma}_u^2 \boldsymbol{I}$ and $\boldsymbol{R}$ by $\hat{\sigma}^2 \boldsymbol{I}$. We have to find an estimator $\hat{\sigma}_u^2$ for σ_u^2 and an estimator $\hat{\sigma}^2$ for σ^2. One approach for this estimate is the restricted (or residual) maximum likelihood (REML) estimator.

We define

$$\boldsymbol{K} \;=\; \boldsymbol{C}(\boldsymbol{I} \;-\; \boldsymbol{P}) \;=\; \boldsymbol{C}\left(\boldsymbol{I} \;- \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{+}\boldsymbol{X}^T\right) \;, \tag{5.134}$$

where $\boldsymbol{C}$ is a full-rank transformation of the rows of $(\boldsymbol{I} \;-\; \boldsymbol{P})$. We immediately see that

$$\boldsymbol{K}\,\boldsymbol{X} \;=\; \boldsymbol{0} \,. \tag{5.135}$$

We define

$$\boldsymbol{\Sigma} \;=\; \sigma_u^2\,\boldsymbol{Z}\boldsymbol{Z}^T \;+\; \sigma^2 \boldsymbol{I}_n \,. \tag{5.136}$$

We know the distribution of $\boldsymbol{K}\,\boldsymbol{y}$:

$$\boldsymbol{K}\,\boldsymbol{y} \;\sim\; \mathcal{N}\left(\boldsymbol{0}\,,\; \boldsymbol{K}\,\boldsymbol{\Sigma}\,\boldsymbol{K}^T\right) \;. \tag{5.137}$$

Using this distribution, estimators for σ^2 and σ_u^2 can be obtained by solving the equations:

$$\mathrm{Tr}\left(\boldsymbol{K}^T\left(\boldsymbol{K}\,\boldsymbol{\Sigma}\,\boldsymbol{K}^T\right)^{-1}\boldsymbol{K}\right) \;=\; \boldsymbol{y}^T\boldsymbol{K}^T\left(\boldsymbol{K}\,\boldsymbol{\Sigma}\,\boldsymbol{K}^T\right)^{-1}\boldsymbol{K}\,\boldsymbol{K}^T\left(\boldsymbol{K}\,\boldsymbol{\Sigma}\,\boldsymbol{K}^T\right)^{-1}\boldsymbol{K}\,\boldsymbol{y} \tag{5.138}$$

$$\mathrm{Tr}\left(\boldsymbol{K}^T\left(\boldsymbol{K}\,\boldsymbol{\Sigma}\,\boldsymbol{K}^T\right)^{-1}\boldsymbol{K}\boldsymbol{Z}\boldsymbol{Z}^T\right) \;=\; \boldsymbol{y}^T\boldsymbol{K}^T\left(\boldsymbol{K}\,\boldsymbol{\Sigma}\,\boldsymbol{K}^T\right)^{-1}\boldsymbol{K}\;\boldsymbol{Z}\boldsymbol{Z}^T\boldsymbol{K}^T\left(\boldsymbol{K}\,\boldsymbol{\Sigma}\,\boldsymbol{K}^T\right)^{-1}\boldsymbol{K}\,\boldsymbol{y}\,. \tag{5.139}$$

These equations are obtained by setting the derivatives of the likelihood of $\boldsymbol{K}\boldsymbol{y}$ with respect to σ^2 and to σ_u^2 to zero.

The solution of these equations are the estimators $\hat{\sigma}^2$ and $\hat{\sigma}_u^2$ for σ^2 and σ_u^2, respectively. Using these estimators, we define

$$\hat{\boldsymbol{\Sigma}} \;=\; \hat{\sigma}_u^2\,\boldsymbol{Z}\boldsymbol{Z}^T \;+\; \hat{\sigma}^2 \boldsymbol{I}_n \,. \tag{5.140}$$

to obtain an estimator for $\boldsymbol{\beta}$ as

$$\hat{\boldsymbol{\beta}} \;=\; \left(\boldsymbol{X}^T\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\right)^{+}\boldsymbol{X}^T\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y}\,. \tag{5.141}$$

This is the estimated generalized least squares (EGLS) estimator. The EGLS estimator is only asymptotically the minimum variance unbiased estimator (MVUE).

Similarly, an approximated estimate for the covariance of $\boldsymbol{\beta}$ is

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) \;=\; \left(\boldsymbol{X}^T\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\right)^{+}\boldsymbol{X}^T\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\;\left(\boldsymbol{X}^T\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\right)^{+} \;. \tag{5.142}$$

For full rank $\boldsymbol{X}$ that is

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) \;=\; \left(\boldsymbol{X}^T\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\right)^{-1} \;. \tag{5.143}$$

Large-sample estimator. Using this approximation for the variance, we can approximate $100(1-\alpha)\%$ confidence intervals by

$$\boldsymbol{a}^T\boldsymbol{\beta} \in \boldsymbol{a}^T\hat{\boldsymbol{\beta}} \pm z_{\alpha/2}\sqrt{\boldsymbol{a}^T\left(\boldsymbol{X}^T\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\right)^{+}\boldsymbol{a}}\,. \tag{5.144}$$

Using the $\boldsymbol{a} = \boldsymbol{e}_j$ gives a confidence interval for β_j. However this confidence interval is not valid for a small number of samples. For a small number of samples we use a different approach.

Small-sample estimator. For

$$t = \frac{\boldsymbol{a}^T\hat{\boldsymbol{\beta}}}{\sqrt{\boldsymbol{a}^T\left(\boldsymbol{X}^T\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\right)^{+}\boldsymbol{a}}} \tag{5.145}$$

often a t-distribution with unknown degrees of freedom is assumed. The task is to estimate the degrees of freedom to compute confidence intervals or p-values. Different approaches exist to estimate the degrees of freedom Rencher and Schaalje [2008].

5.4.1.2 Estimator for u

If $\boldsymbol{u}$ is normally distributed, then we know the posterior $p(\boldsymbol{u} \mid \boldsymbol{y})$.

We use the following connection between two normally distributed variables:

$$\begin{aligned}
&\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_{uu})\ ,\ \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_{yy})\ , \\
&\boldsymbol{\Sigma}_{uv} = \mathrm{Cov}(\boldsymbol{y}, \boldsymbol{u}) \ \text{ and } \ \boldsymbol{\Sigma}_{vu} = \mathrm{Cov}(\boldsymbol{u}, \boldsymbol{y}) : \\
&\boldsymbol{u} \mid \boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{\mu}_u + \boldsymbol{\Sigma}_{vu}\boldsymbol{\Sigma}_{yy}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y)\ ,\ \boldsymbol{\Sigma}_{uu} - \boldsymbol{\Sigma}_{vu}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{uv}\right)
\end{aligned} \tag{5.146}$$

The covariance between $\boldsymbol{u}$ and $\boldsymbol{y}$ is

$$\mathrm{Cov}(\boldsymbol{u}, \boldsymbol{y}) = \mathrm{Cov}(\boldsymbol{u}\ ,\ \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\epsilon}) = \boldsymbol{G}\,\boldsymbol{Z}^T\,. \tag{5.147}$$

and we have

$$\mathrm{E}(\boldsymbol{u}) = \boldsymbol{0}\,, \tag{5.148}$$

$$\mathrm{Var}(\boldsymbol{u}) = \boldsymbol{G}\,, \tag{5.149}$$

$$\mathrm{E}(\boldsymbol{y}) = \boldsymbol{X}\boldsymbol{\beta}\,, \tag{5.150}$$

$$\mathrm{Var}(\boldsymbol{y}) = \boldsymbol{Z}^T\boldsymbol{G}\boldsymbol{Z} + \boldsymbol{R}\,. \tag{5.151}$$

Therefore we obtain

$$\boldsymbol{u} \mid \boldsymbol{y} \sim \mathcal{N}\Big(\boldsymbol{G}\,\boldsymbol{Z}^T\left(\boldsymbol{Z}^T\boldsymbol{G}\boldsymbol{Z} + \boldsymbol{R}\right)^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\ ,\ \boldsymbol{G} - \boldsymbol{G}\,\boldsymbol{Z}^T\left(\boldsymbol{Z}^T\boldsymbol{G}\boldsymbol{Z} + \boldsymbol{R}\right)^{-1}\boldsymbol{Z}\,\boldsymbol{G}^T\Big)\,. \tag{5.152}$$

This posterior can be computed if $\boldsymbol{G}$, $\boldsymbol{R}$, and $\boldsymbol{\beta}$ are known. We can use above approximation for these values

$$\boldsymbol{G} = \hat{\sigma}_u^2\boldsymbol{I}, \tag{5.153}$$

$$\boldsymbol{R} = \hat{\sigma}^2\boldsymbol{I}, \tag{5.154}$$

$$\boldsymbol{Z}^T\boldsymbol{G}\boldsymbol{Z} + \boldsymbol{R} = \hat{\boldsymbol{\Sigma}} \tag{5.155}$$

to estimate the posterior.

5.4.2 Full Estimator

Here we consider the full estimator and not only an approximation. Henderson's "mixed model equations" (MME) are:

$$\begin{pmatrix} \boldsymbol{X}^T \boldsymbol{R}^{-1} \boldsymbol{X} & \boldsymbol{X}^T \boldsymbol{R}^{-1} \boldsymbol{Z} \\ \boldsymbol{Z}^T \boldsymbol{R}^{-1} \boldsymbol{X} & \boldsymbol{Z}^T \boldsymbol{R}^{-1} \boldsymbol{Z} \, + \, \boldsymbol{G}^{-1} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{u}} \end{pmatrix} \; = \; \begin{pmatrix} \boldsymbol{X}^T \boldsymbol{R}^{-1} \boldsymbol{y} \\ \boldsymbol{Z}^T \boldsymbol{R}^{-1} \boldsymbol{y} \end{pmatrix} . \tag{5.156}$$

The solutions to these equations are best linear unbiased estimates (BLUE).

Mixed effect models can also be fitted by the EM algorithm. Variance components are considered as unobserved variables which are estimated in the E-step. The M-step maximizes the other parameters. For example, the R function `lme()` ("linear mixed effect") of the R package `nlme` ("non-linear mixed effect") implements such an EM algorithm.

For $\boldsymbol{R} = \sigma^2 \boldsymbol{I}$ we obtain for the MME

$$\begin{pmatrix} \boldsymbol{X}^T \boldsymbol{X} & \boldsymbol{X}^T \boldsymbol{Z} \\ \boldsymbol{Z}^T \boldsymbol{X} & \boldsymbol{Z}^T \boldsymbol{Z} \, + \, \sigma^{-2} \, \boldsymbol{G}^{-1} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{u}} \end{pmatrix} \; = \; \begin{pmatrix} \boldsymbol{X}^T \boldsymbol{y} \\ \boldsymbol{Z}^T \boldsymbol{y} \end{pmatrix} . \tag{5.157}$$

5.5 Generalized Linear Models

So far, we assumed spherical and often normal errors. However other distributions may be possible — even discrete or count distributions. For example with counts the error corresponds to the deviation from the mean count. The error-free model is obtained by the expectation of the observation y_i: $\mathrm{E}(y_i) = \mu_i = \boldsymbol{x}_i^T \boldsymbol{\beta}$. We now generalize this relation by introducing a link function g. The link function g relates the mean $\mathrm{E}(y_i) = \mu_i$ to the linear component $g(\mu_i) = \boldsymbol{x}_i^T \boldsymbol{\beta}$.

Generalized linear models require

(i) a *random component* or an error distribution which specifies the probability distribution of the response y,

(i) a *systematic component* which is a linear function of the explanatory variables / regressors,

(i) a *link function* which determines the functional relation between the expectation of the random variable and the systematic component, i.e. the linear function.

For the *exponential dispersion model* with the natural parameter θ_i and dispersion parameter ϕ, the density is

$$f(y_i \mid \theta_i, \phi) \; = \; \exp\left(\frac{y_i \theta_i \, - \, b(\theta_i)}{a(\phi)} \; + \; c(y_i, \phi) \right) , \tag{5.158}$$

where

$$b(\theta_i) \; = \; a(\phi) \, \ln \int \exp\left(\frac{y_i \theta_i}{a(\phi)} \; + \; c(y_i, \phi) \right) \, dy_i \, . \tag{5.159}$$

The function b is a normalizing constant (in y_i) that ensures f to be a distribution:

$$\int f(y_i \mid \theta_i, \phi)\, dy_i \;=\; \frac{\int \exp\left(\frac{y_i\theta_i}{a(\phi)} \;+\; c(y_i,\phi)\right)\; dy_i}{\int \exp\left(\frac{y_i\theta_i}{a(\phi)} \;+\; c(y_i,\phi)\right)\; dy_i} \;=\; 1\;. \tag{5.160}$$

Using this density we can derive Rao and Toutenburg [1999]:

$$\begin{aligned} \mathrm{E}(y_i) \;&=\; \mu_i \;=\; b'(\theta_i) && (5.161)\\ \mathrm{Var}(y_i) \;&=\; b''(\theta_i)\, a(\phi)\;. && (5.162)\end{aligned}$$

These equations can be derived as follows. For the mean we have

$$\begin{aligned} \frac{\partial b(\theta_i)}{\partial \theta_i} \;&=\; a(\phi)\; \frac{\int (y_i/a(\phi)) \exp\left(\frac{y_i\theta_i}{a(\phi)} \;+\; c(y_i,\phi)\right)\; dy_i}{\int \exp\left(\frac{y_i\theta_i}{a(\phi)} \;+\; c(y_i,\phi)\right)\; dy_i} && (5.163)\\ &=\; a(\phi)\; \frac{\int (y_i/a(\phi)) \exp\left(\frac{y_i\theta_i}{a(\phi)} \;+\; c(y_i,\phi)\right)\; dy_i}{exp(b(\theta_i)/a(\phi))}\\ &=\; \int y_i \exp\left(\frac{y_i\theta_i \;-\; b(\theta_i)}{a(\phi)} \;+\; c(y_i,\phi)\right)\; dy_i\\ &=\; \mu_i\;. \end{aligned}$$

and for the variance we obtain

$$\begin{aligned} \frac{\partial^2 b(\theta_i)}{\partial \theta_i^2} \;&=\; -\,\frac{1}{a(\phi)}\; \frac{\left(\int y_i\; \exp\left(\frac{y_i\theta_i}{a(\phi)} \;+\; c(y_i,\phi)\right)\; dy_i\right)^2}{\left(\int \exp\left(\frac{y_i\theta_i}{a(\phi)} \;+\; c(y_i,\phi)\right)\; dy_i\right)^2} \;+ && (5.164)\\ &\frac{1}{a(\phi)}\; \frac{\int y_i^2\; \exp\left(\frac{y_i\theta_i}{a(\phi)} \;+\; c(y_i,\phi)\right)\; dy_i}{\int \exp\left(\frac{y_i\theta_i}{a(\phi)} \;+\; c(y_i,\phi)\right)\; dy_i}\\ &=\; \frac{1}{a(\phi)}\left(-\,\mu_i^2 \;+\; \mathrm{E}(y_i^2)\right) \;=\; \frac{1}{a(\phi)}\mathrm{Var}(y_i)\;. \end{aligned}$$

The log-likelihood is

$$\ln \mathcal{L} \;=\; \sum_{i=1}^{n} \ln \mathcal{L}_i \;=\; \sum_{i=1}^{n} \left(\frac{y_i\theta_i \;-\; b(\theta_i)}{a(\phi)} \;+\; c(y_i,\phi)\right)\;, \tag{5.165}$$

where $\mathcal{L}_i = f(y_i \mid \theta_i, \phi)$ is the conditional likelihood of y_i given $\boldsymbol{x}_i$.

The derivative of the log-likelihood with respect to the coefficient β_j is

$$\frac{\partial \ln \mathcal{L}_i}{\partial \beta_j} \;=\; \frac{\partial \ln \mathcal{L}_i}{\partial \theta_i}\; \frac{\partial \theta_i}{\partial \mu_i}\; \frac{\partial \mu_i}{\partial g(\mu_i)}\; \frac{\partial g(\mu_i)}{\partial \beta_j}\;. \tag{5.166}$$

We only applied the chain rule a couple of times.

The derivatives which appear in the chain rule can be computed separately. We compute these derivatives, where we use $\mu_i = b'(\theta_i)$:

$$\frac{\partial \ln \mathcal{L}_i}{\partial \theta_i} = \frac{y_i - b'(\theta_i)}{a(\phi)} = \frac{y_i - \mu_i}{a(\phi)} \tag{5.167}$$

$$\frac{\partial \theta_i}{\partial \mu_i} = \left(\frac{\partial \mu_i}{\partial \theta_i} \right)^{-1} = \left(b''(\theta_i) \right)^{-1} = \frac{a(\phi)}{\mathrm{Var}(y_i)} \tag{5.168}$$

$$\frac{\partial \mu_i}{\partial g(\mu_i)} = \left(\frac{\partial g(\mu_i)}{\partial \mu_i} \right)^{-1} \tag{5.169}$$

$$\frac{\partial g(\mu_i)}{\partial \beta_j} = x_{ij} \; . \tag{5.170}$$

For finding the maximum, the derivative of the log-likelihood is set to zero

$$\frac{\partial \ln \mathcal{L}}{\partial \beta_j} = \sum_{i=1}^{n} \frac{(y_i - \mu_i) \, x_{ij}}{\mathrm{Var}(y_i)} \left(\frac{\partial g(\mu_i)}{\partial \mu_i} \right)^{-1} = 0 \; . \tag{5.171}$$

The maximum likelihood solution is obtained by solving this equation for the parameters $\boldsymbol{\beta}$.

Since $g(\mu_i) = \boldsymbol{x}_i^T \boldsymbol{\beta}$, $\mu_i = b'(\theta_i)$, and $\mathrm{Var}(y_i) = b''(\theta_i) \, a(\phi)$, this equation is nonlinear in $\boldsymbol{\beta}$ depending on the functions g and b. Therefore numerical methods are used to solve this equation. The probability function is determined by the functions a and b while the link function is given by g. A popular method to solve this equation is the iteratively re-weighted least squares algorithm. Using

$$w_i = \frac{\left(\frac{\partial \mu_i}{\partial g(\mu_i)} \right)^2}{\mathrm{Var}(y_i)} \tag{5.172}$$

and the diagonal matrix $\boldsymbol{W} = \mathrm{diag}(w_i)$ the iterative algorithm is

$$\left(\boldsymbol{X}^T \boldsymbol{W}^{(k)} \boldsymbol{X} \right) \boldsymbol{\beta}^{(k+1)} = \left(\boldsymbol{X}^T \boldsymbol{W}^{(k)} \boldsymbol{X} \right) \boldsymbol{\beta}^{(k)} + \frac{\partial \ln \mathcal{L}}{\partial \boldsymbol{\beta}^{(k)}} \; . \tag{5.173}$$

Here $\left(\boldsymbol{X}^T \boldsymbol{W}^{(k)} \boldsymbol{X} \right)$ approximates the Fisher information matrix $\mathcal{F}$:

$$\mathcal{F} \approx \boldsymbol{X}^T \boldsymbol{W}^{(k)} \boldsymbol{X} \; . \tag{5.174}$$

If $\boldsymbol{X}$ has full rank then the update rule becomes

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \left(\boldsymbol{X}^T \boldsymbol{W}^{(k)} \boldsymbol{X} \right)^{-1} \frac{\partial \ln \mathcal{L}}{\partial \boldsymbol{\beta}^{(k)}} \; . \tag{5.175}$$

If different models are fitted, then the maximum likelihood solutions of different models can be compared by a likelihood ratio test. The likelihood ratio test is of interest when using reduced models to test which variables are relevant. Also the interaction of variables might be tested.

Tab. 5.9 shows commonly used generalized linear models described by their distribution and link function. The last three models are known as *logistic regression* and *multinomial logistic regression* for more than two classes.

distribution	link function	link name	support	application
normal	$\boldsymbol{X}\boldsymbol{\beta} = g(\mu) = \mu$	identity	real, $(-\infty, +\infty)$	linear response
exponential	$\boldsymbol{X}\boldsymbol{\beta} = g(\mu) = -\mu^{-1}$	inverse	real, $(0, +\infty)$	exponential response
Gamma	$\boldsymbol{X}\boldsymbol{\beta} = g(\mu) = -\mu^{-1}$	inverse	real, $(0, +\infty)$	exponential response
inv. Gaussian	$\boldsymbol{X}\boldsymbol{\beta} = g(\mu) = -\mu^{-2}$	inv. squared	real, $(0, +\infty)$	
Poisson	$\boldsymbol{X}\boldsymbol{\beta} = g(\mu) = \ln(\mu)$	log	integer, $[0, +\infty)$	count data
Bernoulli	$\boldsymbol{X}\boldsymbol{\beta} = g(\mu) = \ln\left(\frac{\mu}{1-\mu}\right)$	logit	integer, $[0, 1]$	two classes, occurrence
binomial	$\boldsymbol{X}\boldsymbol{\beta} = g(\mu) = \ln\left(\frac{\mu}{1-\mu}\right)$	logit	integer, $[0, n]$	two classes, count
categorical	$\boldsymbol{X}\boldsymbol{\beta} = g(\mu) = \ln\left(\frac{\mu}{1-\mu}\right)$	logit	integer, $[0, K]$	K classes, occurrence
multinomial	$\boldsymbol{X}\boldsymbol{\beta} = g(\mu) = \ln\left(\frac{\mu}{1-\mu}\right)$	logit	integer, $[0, n]^K$	K classes, count

Table 5.9: Commonly used generalized linear models described by their distribution and link function. The probability distribution and the link function are given. Further the support of the distribution, the short-cut name for the link, and the typical application.

Commonly used link functions are: “logit”, “probit”, “cauchit”, “cloglog”, “identity”, “log”, “sqrt”, “inverse squared”, and “inverse”.

The “cloglog” is the “complementary log log function” given as

$$g(x) \;=\; \log\left(- \;\log(x)\right) \; . \tag{5.176}$$

The “cloglog” model is similar to the logit models around 0.5 but differs near 0 or 1.

The R function `glm()` and `glm.fit()` can be used for fitting generalized linear models. For `glm()` the following models are predefined:

```
binomial(link = "logit")
gaussian(link = "identity")
Gamma(link = "inverse")
inverse.gaussian(link = "1/mu^2")
poisson(link = "log")
quasi(link = "identity", variance = "constant")
quasibinomial(link = "logit")
quasipoisson(link = "log")
```

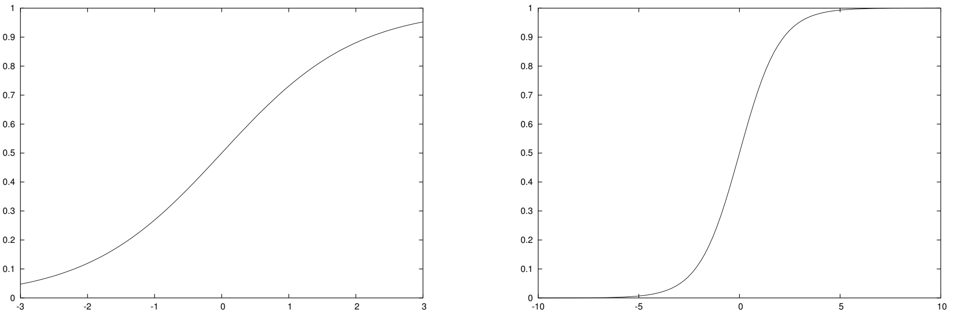

Figure 5.9: The sigmoid function $\frac{1}{1+\exp(-x)}$.

5.5.1 Logistic Regression

5.5.1.1 The Model

The inverse of the logit function

$$g(x) \;=\; \ln\left(\frac{x}{1\;-\;x}\right) \tag{5.177}$$

is the sigmoid function:

$$f(x) \;=\; \frac{1}{1\;+\;e^{-\;x}}\;, \tag{5.178}$$

which is depicted in Fig. 5.9.

Since

$$1\;-\;\frac{1}{1\;+\;e^{-\;x}} \;=\; \frac{e^{-\;x}}{1\;+\;e^{-\;x}}\;, \tag{5.179}$$

we obtain the probabilities

$$p(y\;=\;1\mid \boldsymbol{x};\boldsymbol{\beta}) \;=\; \frac{1}{1\;+\;e^{-\;\boldsymbol{x}^T\boldsymbol{\beta}}} \tag{5.180}$$

and

$$p(y\;=\;0\mid \boldsymbol{x};\boldsymbol{\beta}) \;=\; \frac{e^{-\;\boldsymbol{x}^T\boldsymbol{\beta}}}{1\;+\;e^{-\;\boldsymbol{x}^T\boldsymbol{\beta}}}\;. \tag{5.181}$$

The logit as link function gives

$$\boldsymbol{x}^T\boldsymbol{\beta} \;=\; \ln\left(\frac{p(y\;=\;1\mid \boldsymbol{x})}{1\;-\;p(y\;=\;1\mid \boldsymbol{x})}\right)\;. \tag{5.182}$$

5.5.1.2 Maximizing the Likelihood

We aim at maximizing the likelihood by gradient ascent. Therefore we have to compute the gradient, that is, the first order derivatives of the likelihood $\mathcal{L}$ with respect to the parameters β_j.

The log-likelihood for iid sampled data is

$$\begin{aligned} \ln \mathcal{L}(\{(y_i, \boldsymbol{x}_i)\}; \boldsymbol{\beta}) \ &= \ \sum_{i=1}^{n} \ln p(y_i, \boldsymbol{x}_i; \boldsymbol{\beta}) \ = \\ \sum_{i=1}^{n} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ &+ \ \sum_{i=1}^{n} \ln p(\boldsymbol{x}_i) \ . \end{aligned} \tag{5.183}$$

Only the first sum depends on the parameters, therefore maximum likelihood maximizes the sum of the conditional probabilities

$$\sum_{i=1}^{n} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ , \tag{5.184}$$

This term is often called the conditional likelihood.

Next we will consider the derivative of the log-likelihood. First we will need some algebraic properties:

$$\begin{aligned} &\frac{\partial}{\partial \beta_j} \ln p(y = 1 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ = \ \frac{\partial}{\partial \beta_j} \ln \frac{1}{1 \ + \ e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}} \ = \\ &\left(1 \ + \ e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}\right) \left(- \ \frac{e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}}{\left(1 \ + \ e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}\right)^2} \right) \frac{\partial \ \boldsymbol{x}_i^T \boldsymbol{\beta}}{\partial \beta_j} \ = \\ &- \ \frac{e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 \ + \ e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}} \ \frac{\partial \ \boldsymbol{x}_i^T \boldsymbol{\beta}}{\partial \beta_j} \ = \ - \ p(y \ = \ 0 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ x_{ij} \end{aligned} \tag{5.185}$$

and

$$\begin{aligned} &\frac{\partial}{\partial \beta_j} \ln p(y = 0 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ = \ \frac{\partial}{\partial \beta_j} \ln \frac{e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 \ + \ e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}} \ = \\ &\frac{1 \ + \ e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}}{e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}} \left(\frac{e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 \ + \ e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}} \ - \ \frac{e^{- \ 2 \ \boldsymbol{x}_i^T \boldsymbol{\beta}}}{\left(1 \ + \ e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}\right)^2} \right) \frac{\partial \ \boldsymbol{x}_i^T \boldsymbol{\beta}}{\partial \beta_j} \ = \\ &\frac{1}{1 \ + \ e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}} \ x_{ij} \ = \ p(y \ = \ 1 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ x_{ij} \ . \end{aligned} \tag{5.186}$$

We can rewrite the likelihood as

$$\begin{aligned} &\sum_{i=1}^{n} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ = \\ &\sum_{i=1}^{n} y_i \ln p(y = 1 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ + \ \sum_{i=1}^{n} (1 \ - \ y_i) \ \ln p(y = 0 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \end{aligned} \tag{5.187}$$

which gives for the derivative

$$\begin{aligned}
&\frac{\partial}{\partial \beta_j} \sum_{i=1}^{n} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ = \\
&\sum_{i=1}^{n} y_i \frac{\partial}{\partial \beta_j} \ln p(y = 1 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) + \\
&\sum_{i=1}^{n} (1 \ - \ y_i) \ \frac{\partial}{\partial \beta_j} \ln p(y = 0 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ = \\
&\sum_{i=1}^{n} - \ y_i \ p(y \ = \ 0 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) x_{ij} \ + \\
&\sum_{i=1}^{n} (1 \ - \ y_i) \ p(y \ = \ 1 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) x_{ij} \ = \\
&\sum_{i=1}^{n} \left(- \ y_i \ (1 \ - \ p(y \ = \ 1 \mid \boldsymbol{x}_i; \boldsymbol{\beta})) \right. \\
&\left. (1 \ - \ y_i) \ p(y \ = \ 1 \mid \boldsymbol{x}_i; \boldsymbol{\beta})\right) \ x_{ij} \ = \\
&\sum_{i=1}^{n} \left(p(y \ = \ 1 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ - \ y_i\right) \ x_{ij} \ ,
\end{aligned} \tag{5.188}$$

where

$$p(y \ = \ 1 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ = \ \frac{1}{1 \ + \ e^{- \ \boldsymbol{x}_i^T \boldsymbol{\beta}}} \tag{5.189}$$

For computing the maximum, the derivatives have to be zero

$$\forall_j : \ \sum_{i=1}^{n} \left(p(y \ = \ 1 \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ - \ y_i\right) x_{ij} \ = \ 0 \ . \tag{5.190}$$

A gradient ascent based method may be used to find the solutions to this equation.

Alternative formulation with $y \in +1, -1$

We now give an alternative formulation of logistic regression with $y \in +1, -1$. We remember

$$p(y \ = \ 1 \mid \boldsymbol{x}; \boldsymbol{\beta}) \ = \ \frac{1}{1 \ + \ e^{- \ \boldsymbol{x}^T \boldsymbol{\beta}}} \tag{5.191}$$

and

$$p(y \ = \ -1 \mid \boldsymbol{x}; \boldsymbol{\beta}) \ = \ \frac{e^{- \ \boldsymbol{x}^T \boldsymbol{\beta}}}{1 \ + \ e^{- \ \boldsymbol{x}^T \boldsymbol{\beta}}} \ = \ \frac{1}{1 \ + \ e^{\ \boldsymbol{x}^T \boldsymbol{\beta}}} \ . \tag{5.192}$$

Therefore we have

$$- \ln p(y \ = \ y_i \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \ = \ \ln \left(1 \ + \ e^{- \ y_i \ \boldsymbol{x}_i^T \boldsymbol{\beta}}\right) \tag{5.193}$$

and the objective which is minimized to find the maximum likelihood solution is

$$\mathcal{L} \;=\; - \sum_{i=1}^{n} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \;=\; \sum_{i=1}^{n} \ln \left(1 \;+\; e^{-\, y_i \, \boldsymbol{x}_i^T \boldsymbol{\beta}}\right) \tag{5.194}$$

The derivatives of the objective with respect to the parameters are

$$\begin{aligned} \frac{\partial L}{\partial \beta_j} \;&=\; - \sum_{i=1}^{n} y_i \, \frac{\partial \, \boldsymbol{x}_i^T \boldsymbol{\beta}}{\partial \beta_j} \, \frac{e^{-\, y_i \, \boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 \;+\; e^{-\, y_i \, \boldsymbol{x}_i^T \boldsymbol{\beta}}} \;= \qquad (5.195) \\ & - \sum_{i=1}^{n} y_i \, x_{ij} \; \left(1 \;-\; p(y_i \mid \boldsymbol{x}; \boldsymbol{\beta})\right) \; . \end{aligned}$$

The last equation is similar to Eq. (5.188). In matrix notation we have for the gradient:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} \;=\; - \sum_{i=1}^{n} y_i \left(1 \;-\; p(y_i \mid \boldsymbol{x}; \boldsymbol{\beta})\right) \, \boldsymbol{x}_i \; . \tag{5.196}$$

The objective, the log likelihood, of logistic regression is strictly convex. Therefore efficient gradient-based techniques to find the maximum likelihood solution can be used.

5.5.2 Multinomial Logistic Regression: Softmax

For multi-class problems logistic regression can be generalized to Softmax. We assume K classes with $y \in \{1, \ldots, K\}$ and the probability of $\boldsymbol{x}$ belonging to class k is

$$p(y \;=\; k \mid \boldsymbol{x}; \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K) \;=\; \frac{e^{\boldsymbol{x}^T \boldsymbol{\beta}_k}}{\sum_{j=1}^{K} e^{\boldsymbol{x}^T \boldsymbol{\beta}_j}} \tag{5.197}$$

which gives a multinomial distribution across the classes.

The objective, which is minimized in order to maximize the likelihood, is

$$L \;=\; - \sum_{i=1}^{n} \ln p(y = y_i \mid \boldsymbol{x}_i; \boldsymbol{\beta}) \;=\; \sum_{i=1}^{n} \ln \left(\sum_{j=1}^{K} e^{\boldsymbol{x}^T \boldsymbol{\beta}_j} \right) \;-\; \boldsymbol{x}^T \boldsymbol{\beta}_{y_i} \; . \tag{5.198}$$

In the following we set

$$p(y \;=\; k \mid \boldsymbol{x}; \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K) \;=\; p(k \mid \boldsymbol{x}; \boldsymbol{W}) \; , \tag{5.199}$$

where $\boldsymbol{W} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)$ is the matrix of parameters.

The derivatives are

$$\frac{\partial \mathcal{L}}{\partial \beta_{kt}} \;=\; \sum_{i=1}^{n} \frac{\partial \boldsymbol{x}_i^T \boldsymbol{\beta}_k}{\partial \beta_{kt}} p(k \mid \boldsymbol{x}_i; \boldsymbol{W}) \;-\; \delta_{y_i = k} \sum_{i=1}^{n} \frac{\partial \boldsymbol{x}_i^T \boldsymbol{\beta}_k}{\partial \beta_{kt}} \tag{5.200}$$

$$=\; \sum_{i=1}^{n} x_{it} p(k \mid \boldsymbol{x}_i; \boldsymbol{W}) \;-\; \delta_{y_i = k} \sum_{i=1}^{n} x_{it} \; . \tag{5.201}$$

The objective of Softmax is strictly convex.

distribution	parameters	pmf $\Pr(X = k)$	μ	Var	$r = \mu/\mathrm{Var}$	r
binomial	$n \in \mathbb{N}, p$	$\binom{n}{k} p^k (1-p)^{n-k}$	np	$np(1-p)$	$1/(1-p)$	> 1
Poisson	$0 < \lambda$	$\frac{\lambda^k e^{-\lambda}}{k!}$	λ	λ	1	$= 1$
negative binomial	$0 < r, p$	$\binom{k+r-1}{k}(1-p)^r p^k$	$\frac{pr}{1-p}$	$\frac{pr}{(1-p)^2}$	$(1-p)$	< 1

Table 5.10: Commonly used distributions to model count data. The parameter $p \in [0, 1]$ is the probability of a success. The probability mass function ("pmf"), the mean μ, the variance Var, and the ratio $r = \frac{\mu}{\mathrm{Var}}$ of mean to variance are given. The last column indicates whether r is larger or smaller than 1.

5.5.3 Poisson Regression

To model count data, three distributions are popular: the binomial (variance smaller than the mean), Poisson (variance equal to the mean), negative binomial (variance larger than the mean). Tab. 5.10 shows these distributions.

In many cases the observations can be described by a rate θ and the number of trials n: $\lambda = \theta n$. An observation is the number of successes or failures out of n trials or exposures. Depending on the kind of applications and the problem which should be modeled, either the rate θ changes or the number of exposures changes. For example, n may be the number of kilometers which an individual drives with a car, while θ is the probability of having an accident. In this case, different individuals drove a different number of kilometers, that is, the exposure changes. For another task, all persons drive on a test track 100 km, however, different persons consumed a different amount of alcohol. Therefore, θ, the probability of having an accident, is different for each individual. Consequently, either θ or n can be modeled by a linear regression.

Poisson regression models the case were the rate changes and can be estimated by a linear model using the explanatory variables. We have

$$\mathrm{E}(y_i) \; = \; \lambda_i \; = \; n_i \, \theta_i \; = \; n_i \, e^{\boldsymbol{x}_i^T \boldsymbol{\beta}} \tag{5.202}$$

$$\log \lambda_i \; = \; \log n_i \; + \; \boldsymbol{x}_i^T \boldsymbol{\beta} \, . \tag{5.203}$$

The term $\log n_i$ is an additional offset.

Hypotheses tests can be based on the Wald statistics or on a likelihood ratio statistic. Reduced models allow to test the relevance of different variables for explaining the response variable. Also the combination and interactions of variables can be tested.

The standard error is

$$\mathrm{SE}(\hat{\boldsymbol{\beta}}) \; = \; \sqrt{\frac{1}{\mathcal{F}}} \, , \tag{5.204}$$

where $\mathcal{F}$ is the Fisher information matrix. Confidence intervals can be estimated using

$$\frac{\hat{\beta}_j \; - \; \beta_j}{\mathrm{SE}(\hat{\beta}_j)} \; \sim \; \mathcal{N}(0, 1) \, . \tag{5.205}$$

The estimated values are

$$e_i \;=\; n_i \, e^{\boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}} \tag{5.206}$$

giving the estimated standard deviation $\sqrt{e_i}$. The variance is equal to the mean for a Poisson. The Pearson residuals are

$$r_i \;=\; \frac{o_i \;-\; e_i}{\sqrt{e_i}} \;, \tag{5.207}$$

where o_i are the observed counts. These residuals can be standardized by

$$r_{pi} \;=\; \frac{o_i \;-\; e_i}{\sqrt{e_i}\,\sqrt{1 - P_{ii}}} \;, \tag{5.208}$$

where P_{ii} is the leverage which is the i-th element of the main diagonal of the hat matrix $\boldsymbol{P}$.

The goodness of fit, that is, the error or the objective is chi-squared distributed because

$$\sum_i r_i^2 \;=\; \sum_i \frac{(o_i \;-\; e_i)^2}{e_i} \;, \tag{5.209}$$

which is the definition of a chi-squared statistic.

The Poisson regression is an example of *log-linear models*:

$$\log \mathrm{E}(y_i) = \; c \;+\; \boldsymbol{x}_i^T \boldsymbol{\beta} \;. \tag{5.210}$$

This includes models like

$$\log \mathrm{E}(y_{jk}) = \; \log n \;+\; \log \theta_{j.} \;+\; \log \theta_{.k} \tag{5.211}$$

or

$$\log \mathrm{E}(y_{jk}) = \; \log n \;+\; \log \theta_{jk.} \tag{5.212}$$

which is similar to

$$\log \mathrm{E}(y_{jk}) = \; \mu \;+\; \alpha_j \;+\; \beta_k \;+\; (\alpha\beta)_{jk} \;. \tag{5.213}$$

These models show that ANOVA like approaches are possible in the context of generalized linear models.

5.5.4 Examples

5.5.4.1 Birthweight Data: Normal

We revisit Dobson's birthweight data set from Section 5.3.2.2.

The data is:

```
age <- c(40, 38, 40, 35, 36, 37, 41, 40, 37, 38, 40, 38,
 40, 36, 40, 38, 42, 39, 40, 37, 36, 38, 39, 40)
birthw <- c(2968, 2795, 3163, 2925, 2625, 2847, 3292, 3473, 2628, 3176,
    3421, 2975, 3317, 2729, 2935, 2754, 3210, 2817, 3126, 2539,
    2412, 2991, 2875, 3231)
sex <- gl(2,12, labels=c("Male","Female"))
```

The first model `l0` in Section 5.3.2.2 was a linear model estimated by least squares. This model is a generalized linear model with Gaussian error, therefore `l0` can also be produced by

```
summary(zi <- glm(birthw ~ sex + age, family=gaussian()))

Call:
glm(formula = birthw ~ sex + age, family = gaussian())

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-257.49  -125.28   -58.44   169.00   303.98

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1610.28     786.08  -2.049   0.0532 .
sexFemale    -163.04      72.81  -2.239   0.0361 *
age           120.89      20.46   5.908 7.28e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 31370.04)

    Null deviance: 1829873  on 23  degrees of freedom
Residual deviance:  658771  on 21  degrees of freedom
AIC: 321.39

Number of Fisher Scoring iterations: 2
```

The model without intercept is:

```
summary(z0 <- glm(birthw ~ sex + age - 1, family=gaussian()))

Call:
glm(formula = birthw ~ sex + age - 1, family = gaussian())

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-257.49  -125.28   -58.44   169.00   303.98

Coefficients:
```

```
          Estimate Std. Error t value Pr(>|t|)
sexMale   -1610.28     786.08  -2.049   0.0532 .
sexFemale -1773.32     794.59  -2.232   0.0367 *
age         120.89      20.46   5.908 7.28e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 31370.04)

    Null deviance: 213198964  on 24  degrees of freedom
Residual deviance:    658771  on 21  degrees of freedom
AIC: 321.39

Number of Fisher Scoring iterations: 2
```

We compare these models by an ANOVA table:

```
anova(zi, z0)
Analysis of Deviance Table

Model 1: birthw ~ sex + age
Model 2: birthw ~ sex + age - 1
  Resid. Df Resid. Dev Df    Deviance
1        21     658771
2        21     658771  0 -1.1642e-10
```

The scatter plot in Fig. 5.8 shows that the observation (35,2925) of a male baby looks like an outlier. If we check the residuals, we see

```
z0$residuals
          1           2           3           4           5           6
-257.490545 -188.701891  -62.490545  303.981090 -116.913237  -15.807564
          7           8           9          10          11          12
 -54.384872  247.509455 -234.807564  192.298109  195.509455   -8.701891
         13          14          15          16          17          18
 254.548758  150.126066 -127.451242  -66.662588  -94.239896 -124.556915
         19          20          21          22          23          24
  63.548758 -160.768261 -166.873934  170.337412  -66.556915  168.548758
```

Indeed the fourth observation has the largest residual. We now investigate a subset of the data by removing the observation no. 4. We can use previous models which are updated using the command `update()`:

```
summary(z.o4 <- update(z0, subset = -4))

Call:
glm(formula = birthw ~ sex + age - 1, family = gaussian(), subset = -4)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-253.86  -129.46   -53.46   165.04   251.14

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
sexMale   -2318.03     801.57  -2.892  0.00902 **
sexFemale -2455.44     803.79  -3.055  0.00625 **
age         138.50      20.71   6.688 1.65e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 26925.39)

    Null deviance: 204643339  on 23  degrees of freedom
Residual deviance:    538508  on 20  degrees of freedom
AIC: 304.68

Number of Fisher Scoring iterations: 2
```

Now all regressors are more significant.

Next we add an interaction term:

```
summary(zz <- update(z0, birthw ~ sex+age-1 + sex:age))

Call:
glm(formula = birthw ~ sex + age + sex:age - 1, family = gaussian())

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-246.69  -138.11   -39.13   176.57   274.28

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
sexMale       -1268.67    1114.64  -1.138 0.268492
sexFemale     -2141.67    1163.60  -1.841 0.080574 .
age             111.98      29.05   3.855 0.000986 ***
sexFemale:age    18.42      41.76   0.441 0.663893
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 32621.23)

    Null deviance: 213198964  on 24  degrees of freedom
Residual deviance:    652425  on 20  degrees of freedom
```

Dose ($\log_{10}\mathrm{CS}_2\mathrm{mgl}^{-1}$)	Number of beetles	Number killed
1.6907	59	6
1.7242	60	13
1.7552	62	18
1.7842	56	28
1.8113	63	52
1.8369	59	53
1.8610	62	61
1.8839	60	60

Table 5.11:

```
AIC: 323.16

Number of Fisher Scoring iterations: 2
```

These results are already known from Section 5.3.2.2: the interaction does not help. The ANOVA table tells the same story:

```
anova(z0,zz)
Analysis of Deviance Table

Model 1: birthw ~ sex + age - 1
Model 2: birthw ~ sex + age + sex:age - 1
  Resid. Df Resid. Dev Df Deviance
1       21     658771
2       20     652425  1   6346.2
```

5.5.4.2 Beetle Mortality: Logistic Regression

An example for logistic regression is found in Dobson [2002], page 124, data of Table 7.2. The numbers of dead beetles are counted after five hours exposure to gaseous carbon disulfide at various concentrations. The data stems from Bliss (1935). The data are shown in Tab. 5.11 and as a scatter plot in Fig. 5.10. The dose is actually the logarithm of the quantity of carbon disulfide. For the scatter plot the response was the percentage of dead beetles from all beetles.

The data are binomial because from all beetles a certain number is dead. We produce count data as pairs of (dead,alive):

```
dose <- c(1.6907, 1.7242, 1.7552, 1.7842, 1.8113, 1.8369, 1.861, 1.8839)
x <- c( 6, 13, 18, 28, 52, 53, 61, 60)
n <- c(59, 60, 62, 56, 63, 59, 62, 60)
dead <- cbind(x, n-x)
```

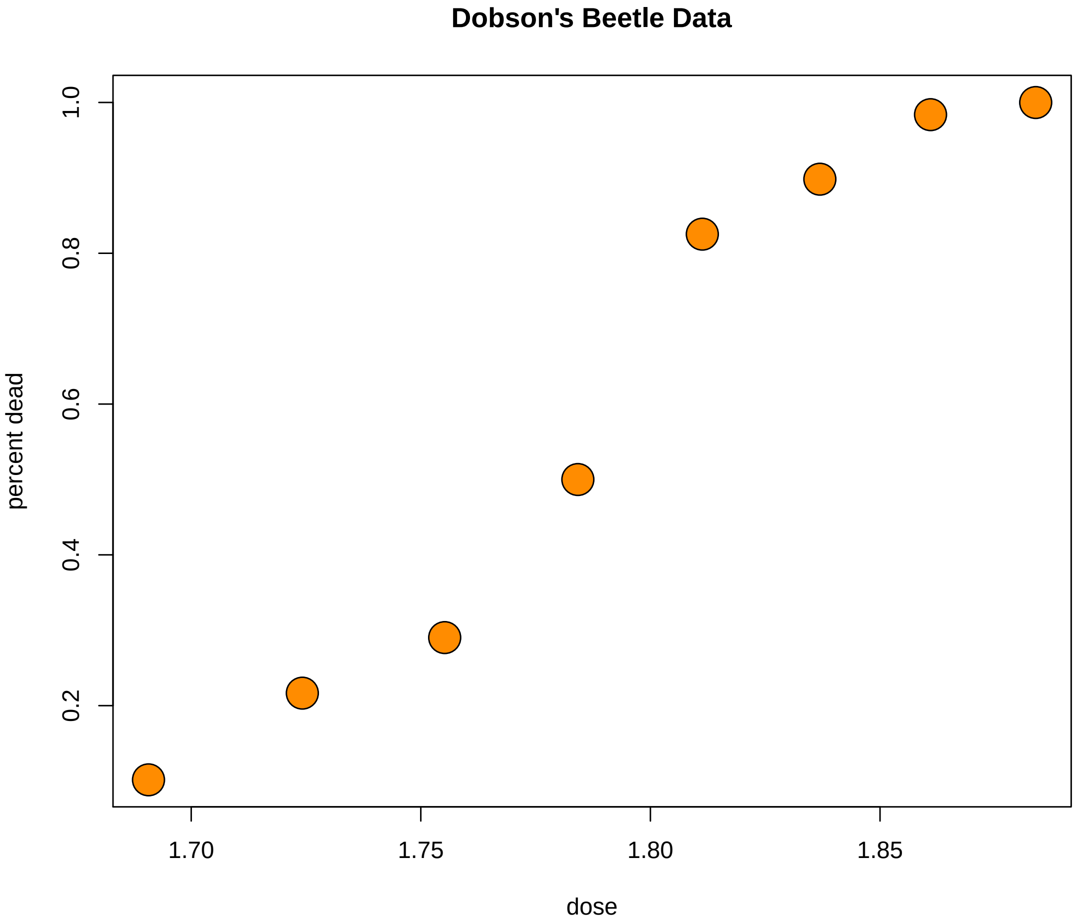

Figure 5.10: Scatter plot of Dobson's beetle data for logistic regression.

We start with logistic regression, that is the distribution is binomial and the link function is logit:

```
summary(zlog <- glm(dead ~ dose, family=binomial(link=logit)))

Call:
glm(formula = dead ~ dose, family = binomial(link = logit))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5941  -0.3944   0.8329   1.2592   1.5940

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -60.717      5.181  -11.72   <2e-16 ***
dose          34.270      2.912   11.77   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 284.202  on 7  degrees of freedom
Residual deviance:  11.232  on 6  degrees of freedom
AIC: 41.43

Number of Fisher Scoring iterations: 4
```

Both intercept and dose are significant. The mean is not around zero, therefore the intercept has to move it. The significance of the dose shows that the number of dead beetles indeed depends on the dose of carbon disulfide.

The next link function, that we try, is the probit.

```
summary(zprob <- glm(dead ~ dose, family=binomial(link=probit)))

Call:
glm(formula = dead ~ dose, family = binomial(link = probit))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5714  -0.4703   0.7501   1.0632   1.3449

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -34.935      2.648  -13.19   <2e-16 ***
dose          19.728      1.487   13.27   <2e-16 ***
---
```

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

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 284.20  on 7  degrees of freedom
Residual deviance:  10.12  on 6  degrees of freedom
AIC: 40.318

Number of Fisher Scoring iterations: 4
```

The result is very similar to the logit link function.

We now test the cloglog link function:

```
summary(zclog <- glm(dead ~ dose, family=binomial(link=cloglog)))

Call:
glm(formula = dead ~ dose, family = binomial(link = cloglog))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.80329  -0.55135   0.03089   0.38315   1.28883

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -39.572      3.240  -12.21   <2e-16 ***
dose          22.041      1.799   12.25   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 284.2024  on 7  degrees of freedom
Residual deviance:   3.4464  on 6  degrees of freedom
AIC: 33.644

Number of Fisher Scoring iterations: 4
```

For this cloglog link function the residual deviance is 3.4464 while it was 11.232 and 10.12 for the logit and probit link function, respectively. Also the AIC (Akaike information criterion) of the last model is lower. This hints at the fact that the last model fits the data better. The fitting of the different link functions is shown in Fig. 5.11, where it is clear that the cloglog link function fits the data best.

5.5.4.3 Embryogenic Anthers: Logistic Regression

Another example for logistic regression is found in Dobson [2002], page 128, data of Table 7.5. The data are taken from Sangwan-Norrell (1977) and are shown in Tab. 5.12. The authors counted

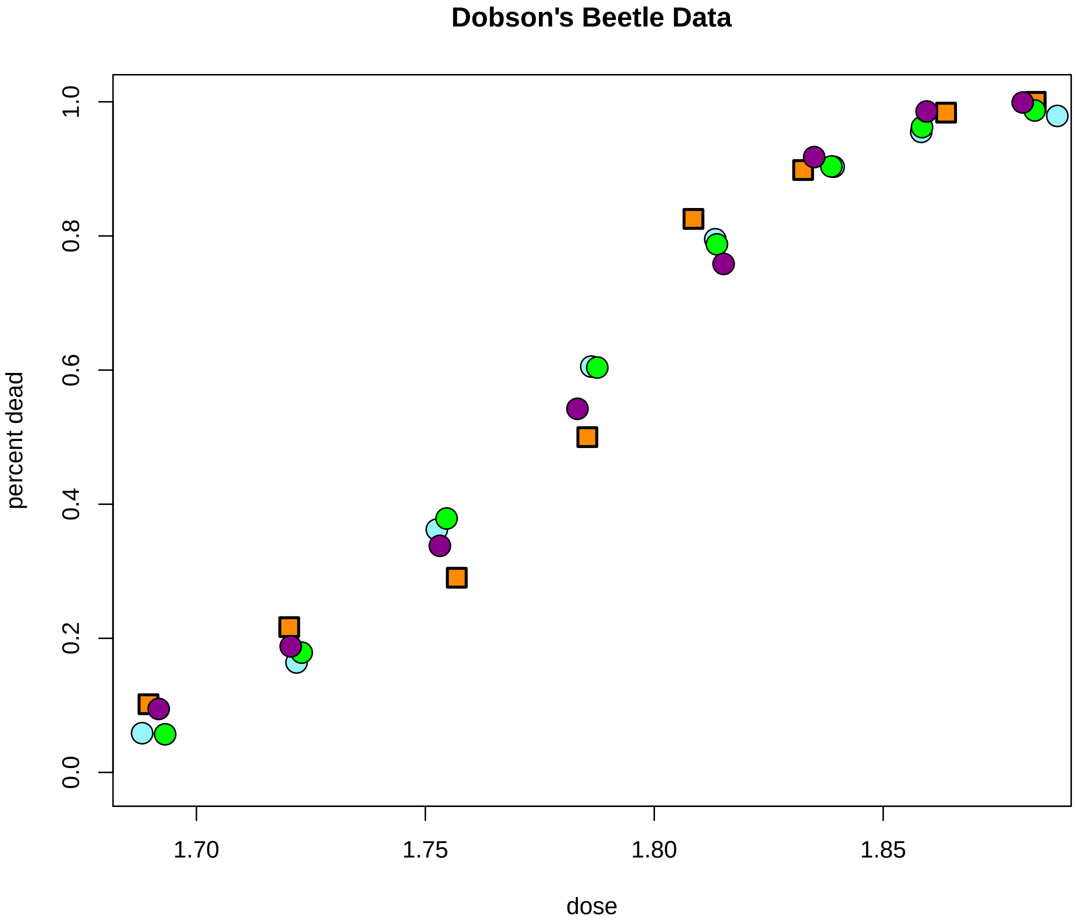

Figure 5.11: Fitting of Dobson's beetle data with different link functions. Orange rectangles are the original data, blue circles are the fitted points with logistic link function, green circles are the fitted points with the probit link function, and the magenta circles are the fitted points with the cloglog link function. The x-axis values are jittered. The cloglog link function fits the points best.

		Centrifuging force (g)		
Storage condition		40	150	350
Control	y	55	52	57
	n	102	99	108
Treatment	y	55	50	50
	n	76	81	90

Table 5.12: Dobson's embryogenic anther data taken from Sangwan-Norrell (1977).

the embryogenic anthers of the plant species *Datura innoxia* Mill. obtained from a particular number of anthers prepared. The embryogenic anthers were obtained under different conditions. The first factor has two levels which relate to the storage type, which is either a control storage or a storage at 3 °C for 48 hours. The second factor has three levels corresponding to the centrifuging forces. The data is shown in Fig. 5.12. The task is to compare the treatment and the control storage type after adjusting for the centrifuging force.

The anther data are generated as follows:

```
x1 <- c(55,52,57,55,50,50)
n <- c(102,  99,   108, 76,   81,   90)
p <- c(0.539,0.525,0.528,0.724,0.617,0.555)
x <- round(n*p)
y <- cbind(x,n-x)
f <- rep(c(40,150,350),2)
(g <- gl(2,3))
```

f gives the centrifuging force and g the storage type.

We first fit a full model:

```
summary(glm(y ~ g*f, family=binomial(link="logit")))

Call:
glm(formula = y ~ g * f, family = binomial(link = "logit"))

Deviance Residuals:
       1          2          3          4          5          6
 0.08269   -0.12998    0.04414    0.42320   -0.60082    0.19522

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.1456719  0.1975451   0.737   0.4609
g2           0.7963143  0.3125046   2.548   0.0108 *
f           -0.0001227  0.0008782  -0.140   0.8889
g2:f        -0.0020493  0.0013483  -1.520   0.1285
```

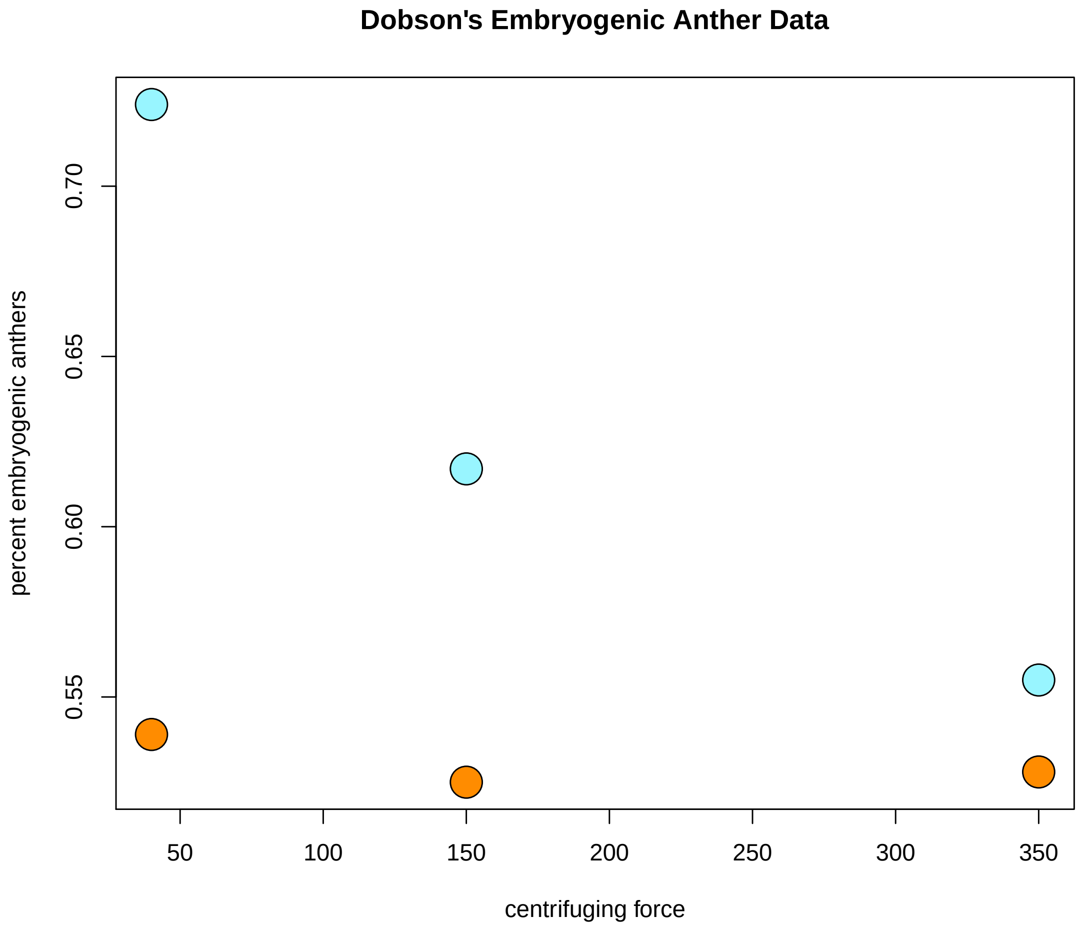

Figure 5.12: Dobson’s embryogenic anther data taken from Sangwan-Norrell (1977). The color mark the groups, which are the storage types.

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

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 10.45197  on 5  degrees of freedom
Residual deviance:  0.60387  on 2  degrees of freedom
AIC: 38.172

Number of Fisher Scoring iterations: 3
```

Next we do not consider the interaction effect between centrifuging force and storage type:

```
summary(glm(y ~ g + f, family=binomial(link="logit")))

Call:
glm(formula = y ~ g + f, family = binomial(link = "logit"))

Deviance Residuals:
      1        2        3        4        5        6
-0.5507  -0.2781   0.7973   1.1558  -0.3688  -0.6584

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.306643   0.167629   1.829   0.0674 .
g2           0.405554   0.174560   2.323   0.0202 *
f           -0.000997   0.000665  -1.499   0.1338
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 10.4520  on 5  degrees of freedom
Residual deviance:  2.9218  on 3  degrees of freedom
AIC: 38.49

Number of Fisher Scoring iterations: 3
```

The centrifuging force seems not to be relevant for explaining the yield in embryogenic anthers. Therefore we only consider the group effects, that is the different storage conditions:

```
summary(glm.p84 <- glm(y~g,  family=binomial(link="logit")))

Call:
glm(formula = y ~ g, family = binomial(link = "logit"))

Deviance Residuals:
       1          2          3          4          5          6 
 0.17150  -0.10947  -0.06177   1.77208  -0.19040  -1.39686 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)  
(Intercept)   0.1231     0.1140   1.080   0.2801  
g2            0.3985     0.1741   2.289   0.0221 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 10.452  on 5  degrees of freedom
Residual deviance:  5.173  on 4  degrees of freedom
AIC: 38.741

Number of Fisher Scoring iterations: 3
```

This best model with respect to the AIC, which only considers the groups, is analyzed in Fig. 5.13.

5.5.4.4 Toy Example 1: Poisson Regression

For Poisson regression we present a toy example from Dobson [2002], page 71, data of Table 4.3. The data are shown in Fig. 5.14. There is a clear relation between x and the count data y as counts for $x = 1.0$ are larger than counts for $x = 0.0$ which in turn are larger than counts for $x = -1.0$.

The R code for the data and performing Poisson regression is:

```
x <- c(-1,-1,0,0,0,0,1,1,1)
y <- c(2,3,6,7,8,9,10,12,15)
summary(glm(y~x, family=poisson(link="identity")))

Call:
glm(formula = y ~ x, family = poisson(link = "identity"))

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-0.7019  -0.3377  -0.1105   0.2958   0.7184  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)   7.4516     0.8841   8.428  < 2e-16 ***
x             4.9353     1.0892   4.531 5.86e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 
```

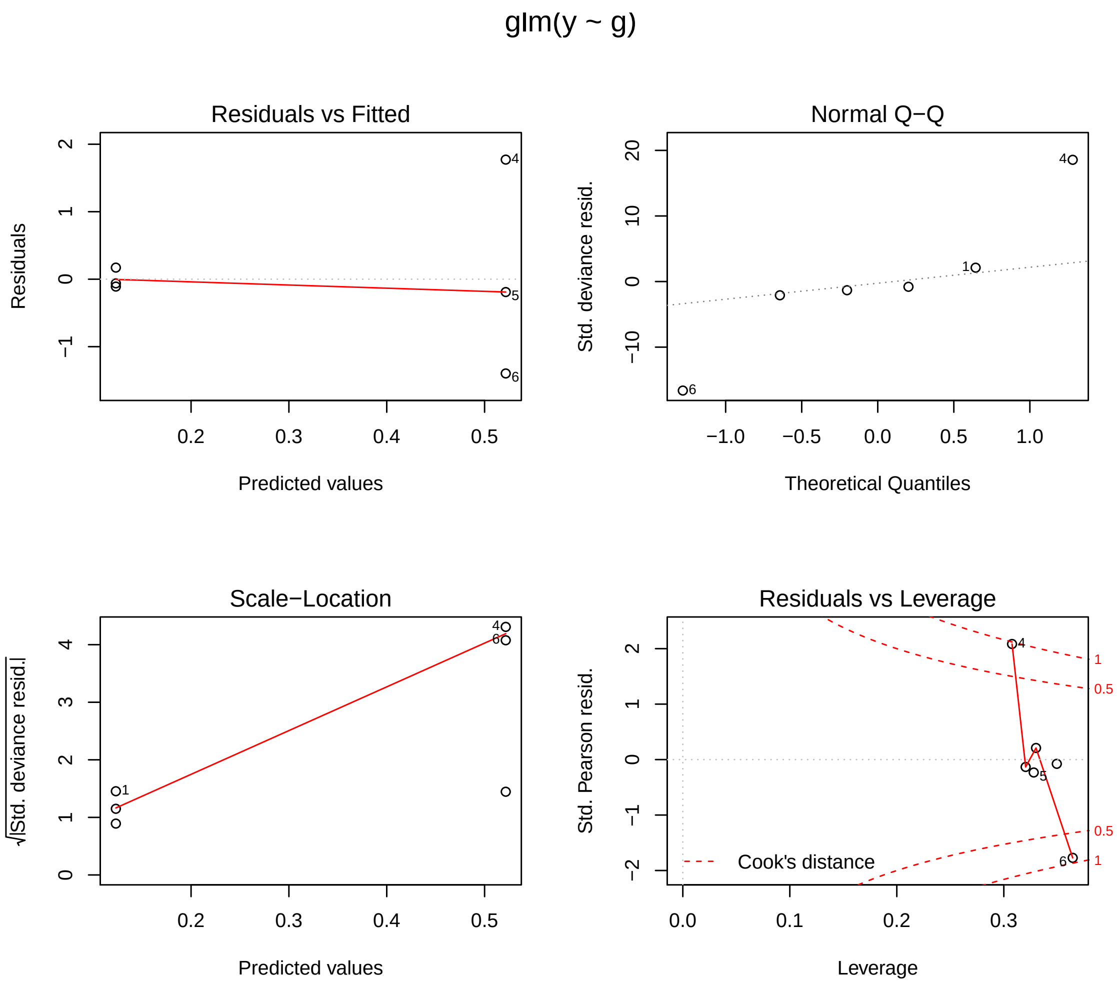

Figure 5.13: The best best model for Dobson's embryogenic anther data with respect to the AIC considers only the groups. The groups are the storage type.

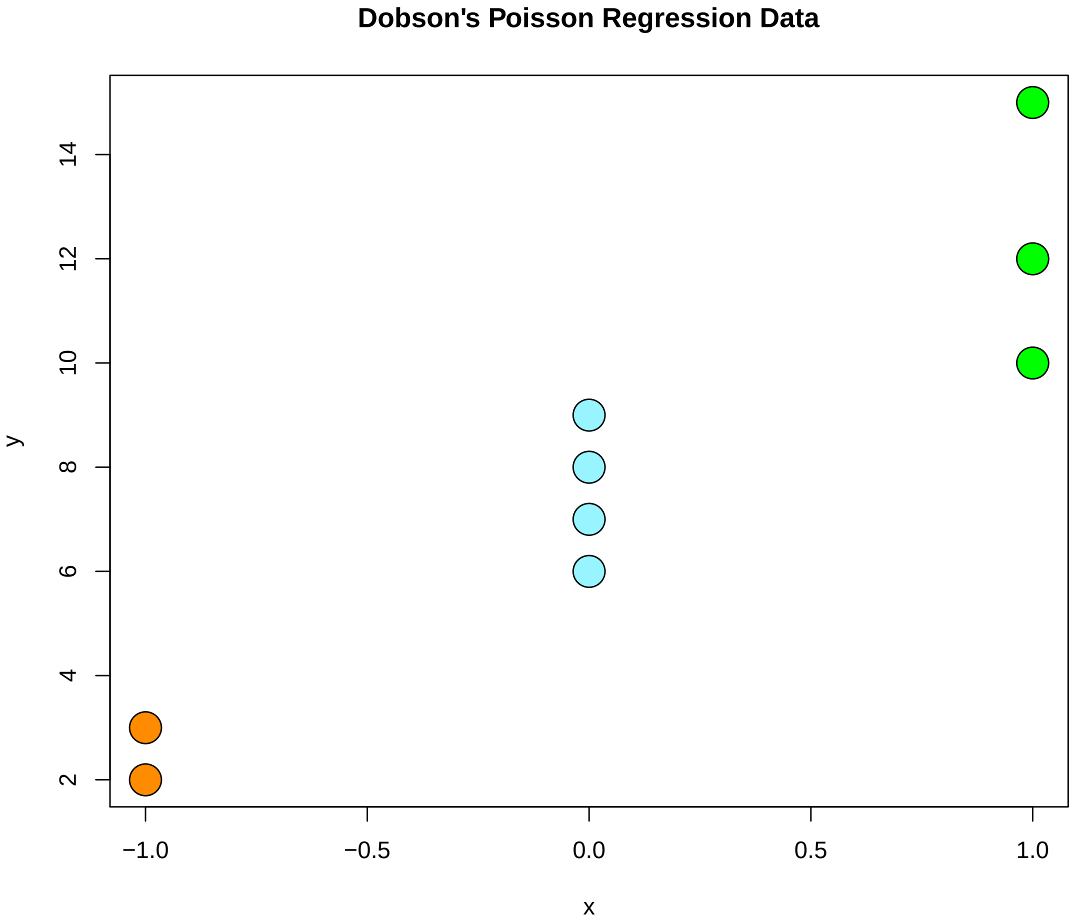

Figure 5.14: Scatter plot of Dobson’s toy data for Poisson regression.

	Outcome			
Treatment	O_1	O_2	O_3	Total
T_1	18	17	15	50
T_2	20	10	20	50
T_3	25	13	12	50
Total	63	40	47	

Table 5.13: Toy data from Dobson [1990] for randomized controlled trial analyzed by Poisson regression.

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 18.4206  on 8  degrees of freedom
Residual deviance:  1.8947  on 7  degrees of freedom
AIC: 40.008

Number of Fisher Scoring iterations: 3
```

Both the intercept and the coefficient are significant. The intercept must move x into the range of the count data.

5.5.4.5 Toy Example 2: Poisson Regression

This is another example for Poisson regression from Dobson [1990] (the first edition), page 93. This example is a randomized controlled trial with two factors. Both factors, outcome and treatment, have three levels. The data is listed in Tab. 5.13. Each treatment group contains 50 samples. Fig. 5.13 shows the data. Outcomes are indicated by the border color of the circles (O_1=blue,O_2=red,O_3=magenta). Treatments are indicated by the interior color of the circles (T_1=orange,T_2=blue,T_3=green). The counts for outcome O_1 are larger than the other two.

We analyze the data by Poisson regression:

```
counts <- c(18,17,15, 20,10,20, 25,13,12)
outcome   <- gl(3, 1, length(counts))
treatment <- gl(3, 3)
summary(z <- glm(counts ~ outcome + treatment, family=poisson()))

Call:
glm(formula = counts ~ outcome + treatment, family = poisson())

Deviance Residuals:
       1         2         3         4         5         6         7         8
-0.67125   0.96272  -0.16965  -0.21999  -0.95552   1.04939   0.84715  -0.09167
       9
```

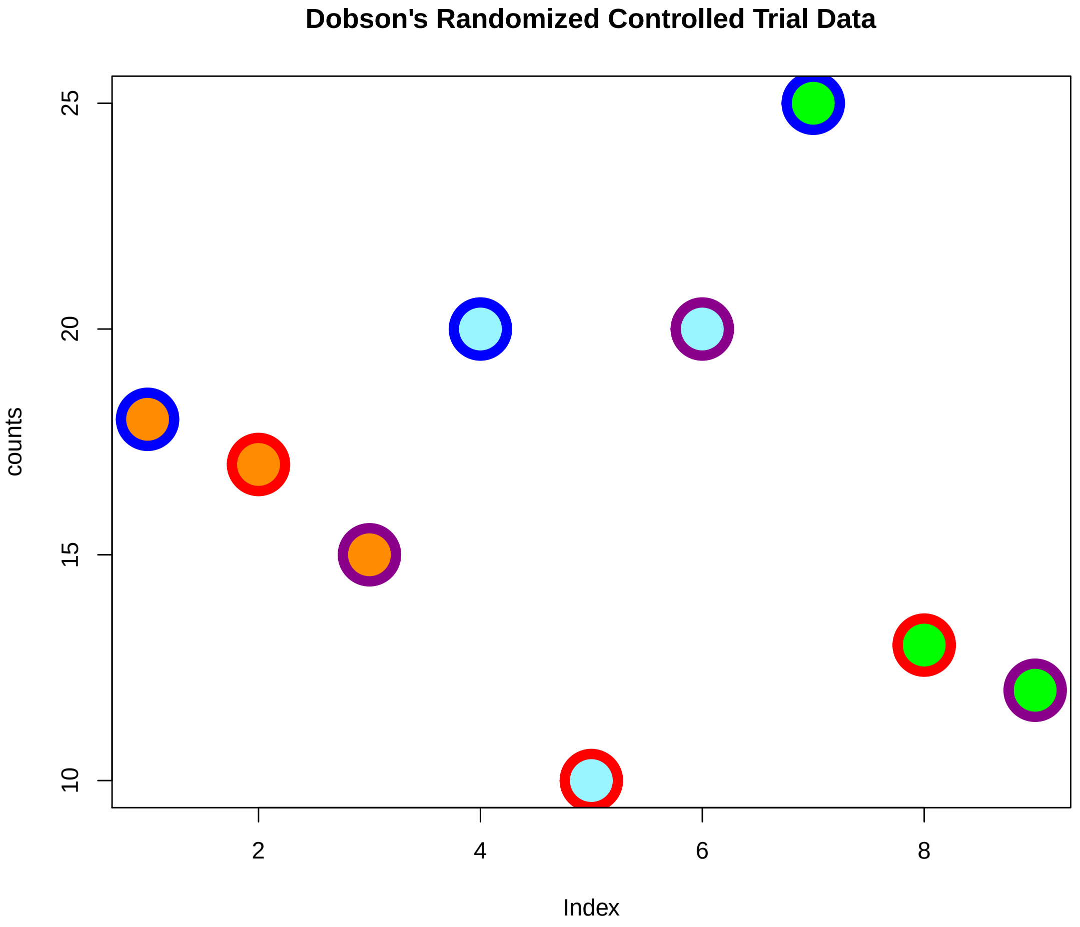

Figure 5.15: Toy data from Dobson [1990] for randomized controlled trial analyzed by Poisson regression. Outcomes are indicated by the border color of the circles (O_1=blue,O_2=red,O_3=magenta). Treatments are indicated by the interior color of the circles (T_1=orange,T_2=blue,T_3=green).

user of M?	No				Yes			
temperature	Low		High		Low		High	
preference	X	M	X	M	X	M	X	M
water softness								
hard	68	42	42	30	37	52	24	43
medium	66	50	33	23	47	55	23	47
soft	63	53	29	27	57	49	19	29

Table 5.14: Data set on detergent brand preference from Ries & Smith (1963) and analyzed by Cox & Snell (1989).

```
-0.96656

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.045e+00  1.709e-01  17.815   <2e-16 ***
outcome2    -4.543e-01  2.022e-01  -2.247   0.0246 *
outcome3    -2.930e-01  1.927e-01  -1.520   0.1285
treatment2   8.717e-16  2.000e-01   0.000   1.0000
treatment3   4.557e-16  2.000e-01   0.000   1.0000
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 10.5814  on 8  degrees of freedom
Residual deviance:  5.1291  on 4  degrees of freedom
AIC: 56.761

Number of Fisher Scoring iterations: 4
```

Of course the intercept is significant as the data is not centered around zero. Outcome 1 and treatment 1 are the reference. Treatment does not have influence on the counts because they are all the same. Outcome O_2 is significant for a level of 0.01. That can be seen in Fig. 5.13 because the reference outcome O_1 indicated by blue border circles is larger than outcome O_2 indicated by red border circles.

5.5.4.6 Detergent Brand: Poisson Regression

These data were reported by Ries & Smith (1963), analyzed by Cox & Snell (1989) and described in Modern Applied Statistics with S+. The user preference for brand M or X is counted. At analyzing these data, different factors are considered. Explanatory variables (regressors, features) are “user of M”, “temperature”, and “water”. The data are presented in Tab. 5.14.

We construct the data set for this example in R as a data frame:

```
Fr <- c(68,42,42,30, 37,52,24,43,
+       66,50,33,23, 47,55,23,47,
+       63,53,29,27, 57,49,19,29)
Temp <- gl(2, 2, 24, labels = c("Low", "High"))
Soft <- gl(3, 8, 24, labels = c("Hard","Medium","Soft"))
M.user <- gl(2, 4, 24, labels = c("N", "Y"))
Brand <- gl(2, 1, 24, labels = c("X", "M"))
detg <- data.frame(Fr,Temp, Soft,M.user, Brand)
```

The results of Poisson regression are:

```
detg.m0 <- glm(Fr ~ M.user*Temp*Soft + Brand, family = poisson, data = detg)
summary(detg.m0)

Call:
glm(formula = Fr ~ M.user * Temp * Soft + Brand, family = poisson,
    data = detg)

     Min        1Q    Median        3Q       Max
-2.20876  -0.99190  -0.00126   0.93542   1.97601

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   4.01524    0.10034  40.018  < 2e-16 ***
M.userY      -0.21184    0.14257  -1.486  0.13731
TempHigh     -0.42381    0.15159  -2.796  0.00518 **
SoftMedium    0.05311    0.13308   0.399  0.68984
SoftSoft      0.05311    0.13308   0.399  0.68984
BrandM       -0.01587    0.06300  -0.252  0.80106
M.userY:TempHigh             0.13987    0.22168   0.631  0.52806
M.userY:SoftMedium           0.08323    0.19685   0.423  0.67245
M.userY:SoftSoft             0.12169    0.19591   0.621  0.53449
TempHigh:SoftMedium         -0.30442    0.22239  -1.369  0.17104
TempHigh:SoftSoft           -0.30442    0.22239  -1.369  0.17104
M.userY:TempHigh:SoftMedium  0.21189    0.31577   0.671  0.50220
M.userY:TempHigh:SoftSoft   -0.20387    0.32540  -0.627  0.53098
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 118.627  on 23  degrees of freedom
Residual deviance:  32.826  on 11  degrees of freedom
AIC: 191.24
```

```
Number of Fisher Scoring iterations: 4
```

Besides the intercept only temperature is significant but not the water characteristic nor the previous use of the brand.

We now try another model:

```
detg.mod <- glm(terms(Fr ~ M.user*Temp*Soft + Brand*M.user*Temp,
+  keep.order = TRUE), family = poisson, data = detg)
summary(detg.mod, correlation = TRUE, symbolic.cor = TRUE)

Call:
glm(formula = terms(Fr ~ M.user * Temp * Soft + Brand * M.user *
    Temp, keep.order = TRUE), family = poisson, data = detg)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.91365  -0.35585   0.00253   0.33027   0.92146

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   4.14887    0.10603  39.128  < 2e-16 ***
M.userY      -0.40521    0.16188  -2.503  0.01231 *
TempHigh     -0.44275    0.17121  -2.586  0.00971 **
M.userY:TempHigh           -0.12692    0.26257  -0.483  0.62883
SoftMedium    0.05311    0.13308   0.399  0.68984
SoftSoft      0.05311    0.13308   0.399  0.68984
M.userY:SoftMedium          0.08323    0.19685   0.423  0.67245
M.userY:SoftSoft            0.12169    0.19591   0.621  0.53449
TempHigh:SoftMedium        -0.30442    0.22239  -1.369  0.17104
TempHigh:SoftSoft          -0.30442    0.22239  -1.369  0.17104
M.userY:TempHigh:SoftMedium  0.21189    0.31577   0.671  0.50220
M.userY:TempHigh:SoftSoft  -0.20387    0.32540  -0.627  0.53098
BrandM       -0.30647    0.10942  -2.801  0.00510 **
M.userY:BrandM              0.40757    0.15961   2.554  0.01066 *
TempHigh:BrandM             0.04411    0.18463   0.239  0.81119
M.userY:TempHigh:BrandM     0.44427    0.26673   1.666  0.09579 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 118.627  on 23  degrees of freedom
Residual deviance:   5.656  on  8  degrees of freedom
AIC: 170.07

Number of Fisher Scoring iterations: 4
```

```
Correlation of Coefficients:

(Intercept)                 1
M.userY                     , 1
TempHigh                    , . 1
M.userY:TempHigh            . , , 1
SoftMedium                  , . .   1
SoftSoft                    , . .   . 1
M.userY:SoftMedium          . ,   . , . 1
M.userY:SoftSoft            . ,   . . , . 1
TempHigh:SoftMedium         .   , . . . .   1
TempHigh:SoftSoft           .   , . . .   . . 1
M.userY:TempHigh:SoftMedium   . . . .   , . , . 1
M.userY:TempHigh:SoftSoft     . . .   . . , . , . 1
BrandM                                     .        1
M.userY:BrandM                               .      , 1
TempHigh:BrandM                                . .  . . 1
M.userY:TempHigh:BrandM                        . .  . . , 1
attr(,"legend")
[1] 0 ‘ ’ 0.3 ‘.’ 0.6 ‘,’ 0.8 ‘+’ 0.9 ‘*’ 0.95 ‘B’ 1
```

Besides the temperature also the brand becomes significant and also, to a lesser degree, the previous use of brand M and the combined previous use of brand M plus brand M.

Finally we compare the two models by an ANOVA table:

```
anova(detg.m0, detg.mod)

Analysis of Deviance Table

Model 1: Fr ~ M.user * Temp * Soft + Brand
Model 2: Fr ~ M.user * Temp * Soft + Brand * M.user * Temp
  Resid. Df Resid. Dev Df Deviance
1        11     32.826
2         8      5.656  3    27.17
```

5.5.4.7 Tumor Data: Poisson Regression

In Dobson [2002], page 162, in Table 9.4, data from Roberts et al. (1981) are presented. The data are from a cross-sectional study of patients with a form of skin cancer called malignant melanoma. For a sample of $n = 400$ patients, the site of the tumor and its histological type were determined. The counts of patients with each combination of tumor type and body site, are given in Tab. 5.15. The patients are categorized by the type of tumor they have, which corresponds to the first factor with four levels: freckle, superficial, nodular, indeterminate. The patients are also categorized by the body site where the tumor was found, which corresponds to the second factor with three levels: head, trunk, extremities. The association between tumor type and site should be investigated.

	Site			
Tumor type	Head & neck	Trunk	Extrem -ities	Total
Hutchinson's melanotic freckle	22	2	10	34
Superficial spreading melanoma	16	54	115	185
Nodular	19	33	73	125
Indeterminate	11	17	28	56
Total	68	106	226	400

Table 5.15: Dobson's malignant melanoma data: frequencies for tumor type and site (Roberts et al., 1981).

Fig. 5.16 shows the data, where the four tumor types are indicated by the interior color of the circles (orange=freckle, blue=superficial, green=nodular, indeterminate=wood). The three locations at the body are indicated by the border color of the circles (head=blue, trunk=red, extremities=magenta).

In R we represent the data by a data frame:

```
counts <- c(22,2,10,16,54,115,19,33,73,11,17,28)
type <- gl(4,3,12,labels=c("freckle","superficial","nodular","indeterminate"))
site <- gl(3,1,12,labels=c("head/neck","trunk","extremities"))
data.frame(counts,type,site)
   counts          type        site
1      22       freckle   head/neck
2       2       freckle       trunk
3      10       freckle extremities
4      16   superficial   head/neck
5      54   superficial       trunk
6     115   superficial extremities
7      19       nodular   head/neck
8      33       nodular       trunk
9      73       nodular extremities
10     11 indeterminate   head/neck
11     17 indeterminate       trunk
12     28 indeterminate extremities
```

We analyze these data by a Poisson regression:

```
summary(z <- glm(counts ~ type + site, family=poisson()))

Call:
glm(formula = counts ~ type + site, family = poisson())

Deviance Residuals:
```

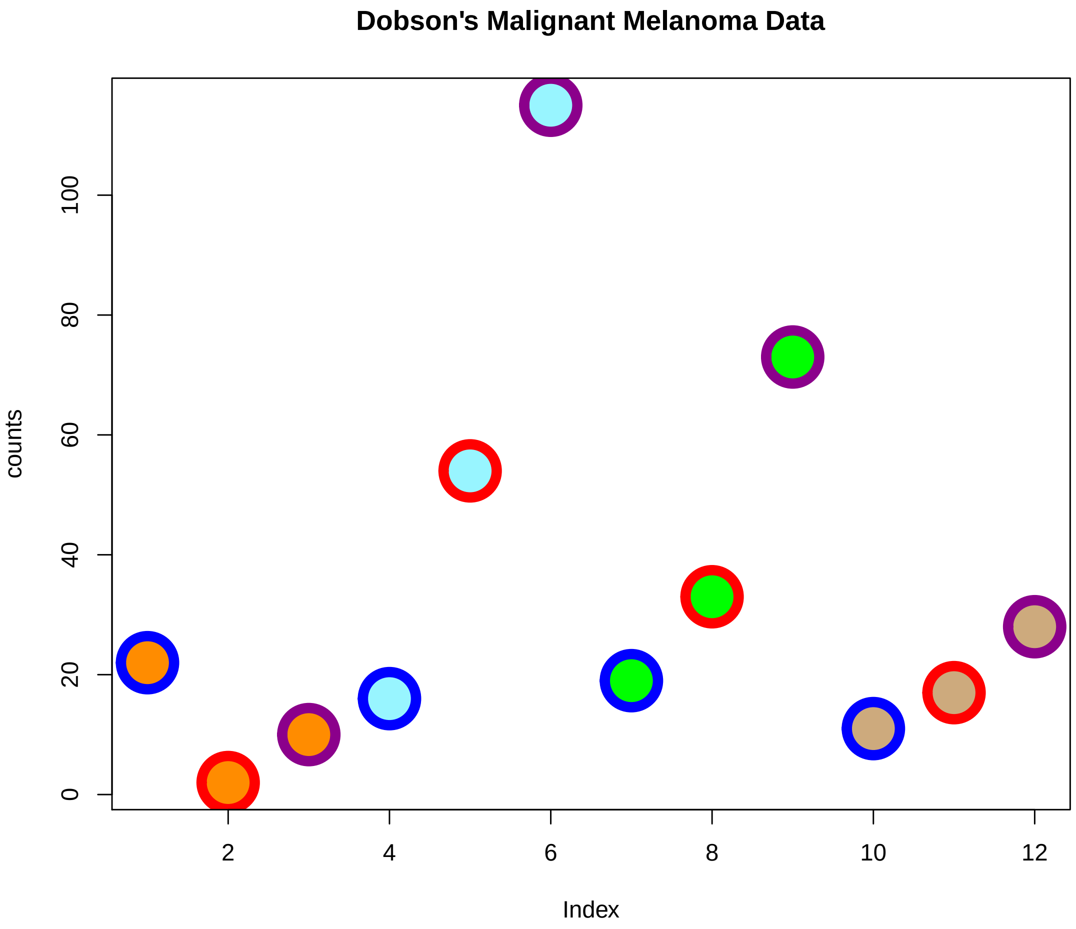

Figure 5.16: Dobson's malignant melanoma data where tumor types are counted. The four tumor types are indicated by the interior color of the circles (orange=freckle, blue=superficial, green=nodular, indeterminate=wood). The three locations at the body are indicated by the border color of the circles (head=blue, trunk=red, extremities=magenta).

```
    Min       1Q   Median       3Q      Max
-3.0453  -1.0741   0.1297   0.5857   5.1354

Coefficients:
   Estimate Std. Error z value Pr(>|z|)
(Intercept)          1.7544     0.2040   8.600  < 2e-16 ***
typesuperficial      1.6940     0.1866   9.079  < 2e-16 ***
typenodular          1.3020     0.1934   6.731 1.68e-11 ***
typeindeterminate   0.4990     0.2174   2.295  0.02173 *
sitetrunk            0.4439     0.1554   2.857  0.00427 **
siteextremities      1.2010     0.1383   8.683  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 295.203  on 11  degrees of freedom
Residual deviance:  51.795  on  6  degrees of freedom
AIC: 122.91

Number of Fisher Scoring iterations: 5
```

This means that type superficial and nodular are highly significant if compared to the counts of type freckle while indeterminate is less significant. This result can be confirmed in Fig. 5.16, where the blue and green interior (blue=superficial, green=nodular) circles have clearly higher counts if compared to freckle. The counts of indeterminate are not so clearly larger. The site extremities is also highly significant. In Fig. 5.16 data points corresponding to counts for extremities have magenta borders. The two largest counts belong to extremities of which one has tumor type superficial and one type nodular. To a lesser degree the site trunk is significant. Also this is confirmed in Fig. 5.16, where the third and fourth largest counts with a red border belong to the site trunk.

5.5.4.8 Ulcers and Aspirin Use: Logistic Regression

This example is a case-control study of gastric and duodenal ulcers and aspirin use from Dobson [2002], page 165/166, with data in Table 9.7. In this retrospective case-control study ulcer patients were compared to controls which are matched with respect to age, sex and socio-economic status. The data is from Duggan et al. (1986). The individuals are categorized:

(1) ulcer cases or controls,

(2) site of the ulcer: gastric or duodenal,

(3) aspirin use or not.

The data is shown in Tab. 5.16 and in Fig. 5.16.

Questions which are of interest for this data set are:

	Aspirin use		
	Non-user	User	Total
Gastric ulcer			
Control	62	6	68
Cases	39	25	64
Duodenal ulcer			
Control	53	8	61
Cases	49	8	57
Total	203	47	250

Table 5.16: Dobson's gastric and duodenal ulcers and aspirin use from Duggan et al. (1986).

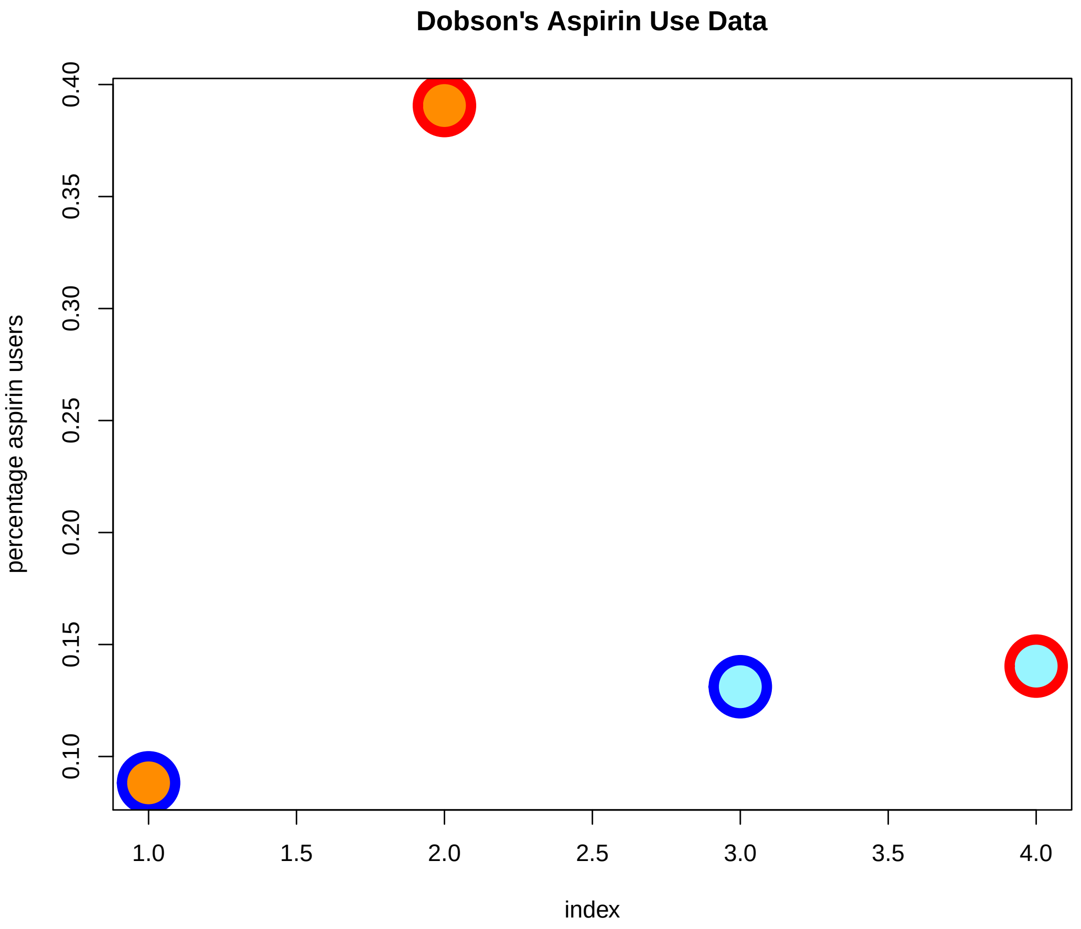

Figure 5.17: Dobson's gastric and duodenal ulcers and aspirin use. The border color indicates ulcer patients, the cases (red), and controls (blue). The interior color indicates the type of ulcer for the cases: gastric (orange) or duodenal (blue).

1. Is gastric ulcer associated with aspirin use?
2. Is duodenal ulcer associated with aspirin use?
3. Is any association with aspirin use the same for both ulcer sites?

We create the data in R which are pairs of user/non-user counts for the different groups and types:

```
counts <- c(62,6,39,25,53,8,49,8)
group <- gl(2, 1, 4, labels=c("controls","cases"))
type <- gl(2,2,4,labels=c("gastric","duodenal"))
aspirin <- gl(2,1,8,labels=c("non-user","user"))
a <- which(aspirin=="user")
n <- which(aspirin=="non-user")
y <- cbind(counts[n],counts[a])
```

We first look at a model without interaction effects:

```
summary(z <- glm(y ~ group + type, family=binomial(link="logit")),correlation = TRUE)

Call:
glm(formula = y ~ group + type, family = binomial(link = "logit"))

Deviance Residuals:
      1        2        3        4  
 1.2891  -0.9061  -1.5396   1.1959  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)    1.8219     0.3080   5.916  3.3e-09 ***
groupcases    -1.1429     0.3521  -3.246  0.00117 ** 
typeduodenal   0.7000     0.3460   2.023  0.04306 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 21.789  on 3  degrees of freedom
Residual deviance:  6.283  on 1  degrees of freedom
AIC: 28.003

Number of Fisher Scoring iterations: 4

Correlation of Coefficients:
             (Intercept) groupcases
groupcases   -0.73                 
typeduodenal -0.38       -0.05     
```

As the count data are not centered, the intercept is significant. Most significant is the group cases for aspirin use. The rate is the percentage of the first count of all counts, that is the rate of aspirin non-users. The coefficient of group cases is -1.14 which means the rate of non-users is smaller than the rate for controls. This means that for cases the percentage of aspirin use is larger than for controls. Less significant and almost not significant is the type of ulcer where gastric is more related to aspirin users.

Next, we investigate the linear model with interaction effects.

```
summary(z1 <- glm(y ~ group*type, family=binomial(link="logit")))

Call:
glm(formula = y ~ group * type, family = binomial(link = "logit"))

Deviance Residuals:
[1]  0  0  0  0

Coefficients:
         Estimate Std. Error z value Pr(>|z|)
(Intercept)               2.3354     0.4275   5.462  4.7e-08 ***
groupcases               -1.8907     0.4984  -3.793 0.000149 ***
typeduodenal             -0.4445     0.5715  -0.778 0.436711
groupcases:typeduodenal   1.8122     0.7333   2.471 0.013460 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2.1789e+01  on 3  degrees of freedom
Residual deviance: 2.3981e-14  on 0  degrees of freedom
AIC: 23.72

Number of Fisher Scoring iterations: 3
```

Again cases are significantly associated with aspirin use. Further cases with gastric are more related to aspirin use.

We compare these two models by an ANOVA table:

```
anova(z, z1, test = "Chisq")
Analysis of Deviance Table

Model 1: y ~ group + type
Model 2: y ~ group * type
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         1      6.283
2         0      0.000  1    6.283  0.01219 *
---
```

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

The deviance shows that the interaction model is significantly better at fitting the data. However, the AIC tells that this may only be due to overfitting to the data.

5.6 Regularization

In machine learning and statistics it is important to avoid that the model is too much fitted to the data. In this case only data specific features are modeled but not the structure in the data. This is called overfitting. Overfitting reduces generalization capabilities because other, new data will not have the specific features of the current data but only the general structures. To avoid overfitting, simple models should be selected Hochreiter and Schmidhuber [1995, 1994, 1997a], Hochreiter and Obermayer [2006], Hochreiter et al. [2007], Knebel et al. [2008]. Simple models are models from low-complex model classes and as such cannot capture specific data characteristics but only general structures in the data. To prefer simple models during model selection is called *regularization*. In the following we present some regularization methods for linear models.

5.6.1 Partial Least Squares Regression

The first kind of regularization is based on models which are based on a $l < m$ variables. This means that regularization is achieved by fitting a model in a lower dimensional space. The idea of *partial least squares* (PLS) is to factorize both the response matrix $\boldsymbol{Y}$ and the regression matrix $\boldsymbol{X}$:

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{T}\,\boldsymbol{P}^T \;+ \boldsymbol{E} && (5.214)\\ \boldsymbol{Y} &= \boldsymbol{U}\,\boldsymbol{Q}^T \;+\; \boldsymbol{F}\;, && (5.215) \end{aligned}$$

where the covariance between $\boldsymbol{T}$ and $\boldsymbol{U}$ is maximized. $\boldsymbol{X}$ is an $n \times m$ matrix of predictors. $\boldsymbol{Y}$ is an $n \times p$ matrix of responses. $\boldsymbol{T}$ and $\boldsymbol{U}$ are $n \times l$ matrices that are, respectively, projections of $\boldsymbol{X}$ and projections of $\boldsymbol{Y}$. $\boldsymbol{P}$ and $\boldsymbol{Q}$ are, respectively, $m \times l$ and $p \times l$ orthogonal matrices. $\boldsymbol{E}$ and $\boldsymbol{F}$ are additive noise terms which are assumed to be independently normally distributed.

Iterative partial least squares finds projection vectors $\boldsymbol{w}$ for $\boldsymbol{X}$ and $\boldsymbol{v}$ for $\boldsymbol{Y}$ which have maximal covariance:

$$\max_{\|\boldsymbol{w}\|=\|\boldsymbol{v}\|=1} \operatorname{Cov}(\boldsymbol{X}\boldsymbol{w}, \boldsymbol{Y}\boldsymbol{v})\;. \tag{5.216}$$

Iterative partial least squares is closely related to *canonical correlation analysis* (CCA) which finds projection vectors $\boldsymbol{w}$ for $\boldsymbol{X}$ and $\boldsymbol{v}$ for $\boldsymbol{Y}$ which have maximal correlation coefficient:

$$\max_{\|\boldsymbol{w}\|=\|\boldsymbol{v}\|=1} \operatorname{corr}(\boldsymbol{X}\boldsymbol{w}, \boldsymbol{Y}\boldsymbol{v})\;. \tag{5.217}$$

PLS takes the variance into account while CCA only looks at the correlation.

For *partial least squares regression* (PLSR) the score matrix $\boldsymbol{T}$ is orthogonal:

$$\boldsymbol{T}^T\boldsymbol{T} \;=\; \boldsymbol{I}\;. \tag{5.218}$$

PLSR defines a *linear inner relation*, which is basically a regression:

$$\boldsymbol{U} \;=\; \boldsymbol{T}\,\boldsymbol{D} \;+\; \boldsymbol{H}\;, \tag{5.219}$$

where $\boldsymbol{D}$ is a diagonal matrix. Via this regression the covariance between $\boldsymbol{T}$ and $\boldsymbol{U}$ is maximized. This regression gives

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{T}\,\boldsymbol{D}\,\boldsymbol{Q}^T \;+\; \boldsymbol{H}\,\boldsymbol{Q}^T \;+\; \boldsymbol{F} && (5.220)\\ &= \boldsymbol{T}\,\boldsymbol{C}^T \;+\; \boldsymbol{F}' \,, && (5.221) \end{aligned}$$

where $\boldsymbol{C}^T = \boldsymbol{D}\boldsymbol{Q}^T$ are the regression coefficients and $\boldsymbol{F}' = \boldsymbol{H}\boldsymbol{Q}^T + \boldsymbol{F}$ is the noise. We obtained a least squares estimate with projections $\boldsymbol{T}$ from orthogonal matrices.

For the noise free case, we have the decompositions

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{T}\,\boldsymbol{P}^T \,, && (5.222)\\ \boldsymbol{T} &= \boldsymbol{X}\,\boldsymbol{W} \,, && (5.223)\\ \hat{\boldsymbol{Y}} &= \boldsymbol{T}\,\boldsymbol{D}\,\boldsymbol{Q}^T \,, && (5.224)\\ \boldsymbol{U} &= \hat{\boldsymbol{Y}}\,\boldsymbol{Q} \,. && (5.225) \end{aligned}$$

The matrix $\hat{\boldsymbol{Y}}$ approximates $\boldsymbol{Y}$, the columns of $\boldsymbol{T}$ are the “latent vectors”, $\boldsymbol{D}$ are the “regression weights” (see Eq. (5.219)) and $\boldsymbol{Q}$ is the “weight matrix” of the dependent variables $\boldsymbol{Y}$.

$\boldsymbol{W}$ is pseudo inverse of $\boldsymbol{P}^T$ which leads to the following equations:

$$\begin{aligned} \boldsymbol{T}^T\boldsymbol{T} &= \boldsymbol{I} \,, && (5.226)\\ \boldsymbol{Q}^T\boldsymbol{Q} &= \boldsymbol{I} \,, && (5.227)\\ \boldsymbol{W} &= (\boldsymbol{P}^T)^+ \,, && (5.228)\\ \boldsymbol{U} &= \boldsymbol{T}\,\boldsymbol{D} \,, && (5.229)\\ \boldsymbol{D} &= \boldsymbol{T}^T\,\boldsymbol{U} \,. && (5.230) \end{aligned}$$

Using these equations the partial least squares regression algorithm Alg. 5.1 can be derived.

Partial least squares regression can be based on the singular value decomposition of $\boldsymbol{X}^T\boldsymbol{Y}$. If noise terms are ignored then we have

$$\boldsymbol{X}^T\boldsymbol{Y} \;=\; \boldsymbol{P}\,\boldsymbol{T}^T\boldsymbol{U}\,\boldsymbol{Q}^T \;=\; \boldsymbol{P}\,\boldsymbol{D}\,\boldsymbol{Q}^T \,, \qquad (5.231)$$

where the second equality follows from Eq. (5.219). The largest singular value gives the first $\boldsymbol{w}$ and the first $\boldsymbol{q}$. The first $\boldsymbol{t}$ is the first eigenvector of $\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{Y}\boldsymbol{Y}^T$ and the first $\boldsymbol{u}$ is the first eigenvector of $\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{X}\boldsymbol{X}^T$.

If $\boldsymbol{T}$ are the projections onto the first l principal components of $\boldsymbol{X}$ then this is called *principal components regression*.

5.6.2 Ridge Regression

Ridge regression is also known as *Tikhonov regularization* for ill-posed problems. The objective of the least squares estimate is the sum of squares

$$\|\boldsymbol{X}\boldsymbol{\beta} \;-\; \boldsymbol{y}\|^2 \,. \qquad (5.232)$$

Algorithm 5.1 Partial least squares regression

```
  Given: matrix $\boldsymbol{X}$, matrix $\boldsymbol{Y}$
initialization
  initialize $\boldsymbol{u}$ by random values
  $\boldsymbol{A}$ is set to the column centered and column normalized $\boldsymbol{X}$
  $\boldsymbol{B}$ is set to the column centered and column normalized $\boldsymbol{Y}$
main loop
  while $\boldsymbol{A}$ is not the null matrix do
    while not converged do
      $\boldsymbol{w} = \boldsymbol{A}^T \boldsymbol{u}$ (estimate $\boldsymbol{X}$ weights)
      $\boldsymbol{t} = \boldsymbol{A}\boldsymbol{w}$ (estimate $\boldsymbol{X}$ factor scores)
      $\boldsymbol{t} = \boldsymbol{t}/\|\boldsymbol{t}\|$ (normalize factor scores)
      $\boldsymbol{q} = \boldsymbol{B}^T \boldsymbol{t}$ (estimate $\boldsymbol{Y}$ weights)
      $\boldsymbol{q} = \boldsymbol{q}/\|\boldsymbol{q}\|$ (normalize weights)
      $\boldsymbol{u} = \boldsymbol{B}\boldsymbol{q}$ (estimate $\boldsymbol{Y}$ factor scores)
      use $\boldsymbol{w}$ to test if loop has converged
    end while
    $d = \boldsymbol{t}^T \boldsymbol{u}$
    $\boldsymbol{p} = \boldsymbol{A}^T \boldsymbol{t}$
    $\boldsymbol{A} = \boldsymbol{A} - \boldsymbol{t}\boldsymbol{p}^T$ (partial out the effect of $\boldsymbol{t}$ from $\boldsymbol{X} \sim \boldsymbol{A}$)
    $\boldsymbol{B} = \boldsymbol{B} - d\boldsymbol{t}\boldsymbol{q}^T$ (partial out the effect of $\boldsymbol{t}$ from $\boldsymbol{Y} \sim \boldsymbol{B}$)
    store all computed values $\boldsymbol{t}$, $\boldsymbol{u}$, $\boldsymbol{w}$, $\boldsymbol{q}$, $\boldsymbol{p}$ in the corresponding matrices
    store $d$ as diagonal element of $\boldsymbol{D}$
  end while
result
  training: $\hat{\boldsymbol{Y}} = \boldsymbol{T}\boldsymbol{D}\boldsymbol{Q}^T$
  prediction: $\boldsymbol{\tau} = \boldsymbol{x}^T \boldsymbol{W}$ ($\boldsymbol{x}$ is normalized like $\boldsymbol{A}$); $\hat{\boldsymbol{y}} = \boldsymbol{\tau}\boldsymbol{D}\boldsymbol{Q}^T$
```

If the number of regressors is large, then overfitting is a problem. Overfitting refers to the fact that specific observations are fitted even if they are noisy or outliers. In this case the estimated parameters are adjusted to specific characteristics of the observed data which reduces the generalization to new unknown data. To avoid overfitting simple models should be selected even if they do not fit the observed data as well as the model with minimal squared error. Regularization fits the data while preferring simple models, that is, there is a trade-off between simple models and small squared error. This trade-off is controlled by a hyperparameter.

Regularization can be performed by an additional objective on the parameters, like a squared term in the parameters:

$$\|\boldsymbol{X}\boldsymbol{\beta} \ - \ \boldsymbol{y}\|^2 \ + \ \|\boldsymbol{\Gamma}\,\boldsymbol{\beta}\|^2 \,. \tag{5.233}$$

The estimator for ridge regression $\hat{\boldsymbol{\beta}}$, which minimizes this objective, is

$$\hat{\boldsymbol{\beta}} \ = \ \left(\boldsymbol{X}^T\boldsymbol{X} \ + \ \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\right)^{-1} \boldsymbol{X}^T\boldsymbol{y} \,. \tag{5.234}$$

Often $\boldsymbol{\Gamma} = \sqrt{\gamma}\boldsymbol{I}$ is used, where γ is a hyperparameter of the method which has to be adjusted. γ controls the trade-off between simple models and low squared error. For $\boldsymbol{\Gamma} = \sqrt{\gamma}\boldsymbol{I}$ we have the estimator

$$\hat{\boldsymbol{\beta}} \ = \ \left(\boldsymbol{X}^T\boldsymbol{X} \ + \ \gamma\boldsymbol{I}\right)^{-1} \boldsymbol{X}^T\boldsymbol{y} \,. \tag{5.235}$$

The variance of the ridge regression estimator is:

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) \ = \ \sigma^2 \ \left(\boldsymbol{X}^T\boldsymbol{X} \ + \ \gamma\boldsymbol{I}\right)^{-1} \boldsymbol{X}^T\boldsymbol{X} \left(\boldsymbol{X}^T\boldsymbol{X} \ + \ \gamma\boldsymbol{I}\right)^{-1} \ . \tag{5.236}$$

The bias of ridge regression estimator is:

$$\mathrm{bias}(\hat{\boldsymbol{\beta}}) \ = \ -\,\gamma \ \left(\boldsymbol{X}^T\boldsymbol{X} \ + \ \gamma\boldsymbol{I}\right)^{-1} \boldsymbol{\beta} \,. \tag{5.237}$$

It has been shown, that there is always a γ for which the parameter mean squared error of ridge regression is smaller than this error of least squares. However, this γ is not known. Ridge regression is consistent if $\gamma/n \overset{n}{\to} 0$ Knight and Fu [2000].

Ridge regression is an L^2-norm regularizer, that is the squares of the parameters (or products of them) are weighted and summed up and thereby penalized. Therefore small absolute parameter values around zero are preferred by ridge regression. However, in general the ridge regression estimator has its parameters not exactly at zero. The regularizing term hardly changes if the values are already small because the derivatives are proportional to the values. If very small parameter values still improve the squared error, they will be kept. Setting these small parameters to zero would increase the error more than it would decrease the regularization term. On the other hand, larger values are very strongly penalized.

Ridge regression gives a solution even if the parameters are under-determined for few data points because $\left(\boldsymbol{X}^T\boldsymbol{X} \ + \ \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\right)^{-1}$ always exists. This means that ridge regression has a unique solution.

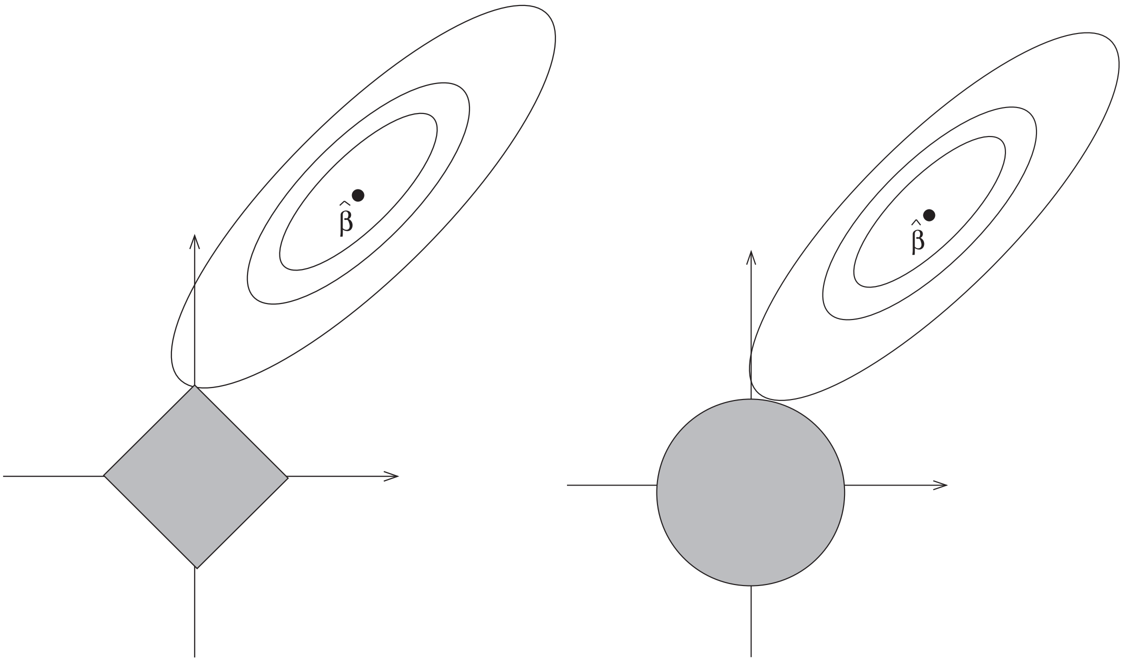

Figure 5.18: Optimization LASSO (left) vs. ridge regression (right). The error objective, the ellipse, touches in most cases a corner of the L^1-norm where at least one component is zero. In contrast the L^2-norm does not possess corners as all points with the same regularization value are on a hyperball.

5.6.3 LASSO

Least absolute shrinkage and selection operator (LASSO) Tibshirani [1996] performs a L^1-norm regularization. The objective is

$$\|\boldsymbol{X}\boldsymbol{\beta} \; - \; \boldsymbol{y}\|^2 \; + \; \gamma \; \|\boldsymbol{\beta}\|_1 \; . \tag{5.238}$$

In contrast to ridge regression, the LASSO estimate has many zero components (see Fig. 5.18). The decrease of the regularization term if the absolute values of parameters are made smaller, does not depend on the current values of the parameters. Thus, small parameter values are pushed toward zero. Therefore LASSO is often used for feature selection because features, of which the corresponding parameters are zero, can be removed from the model without changing regression result.

The minimization of the LASSO objective is a quadratic optimization problem. It can be solved by techniques of constrained quadratic optimization. An alternative method for finding a solution is the *forward stepwise regression algorithm*:

1. Start with all coefficients β_j equal to zero.

2. Find the predictor x_j which is most correlated with y, and add it to the model. Take residuals $r = y - \hat{y}$.

3. Continue, at each stage adding to the model the predictor which is most correlated with r.

4. Until: all predictors are in the model

A even better approach to finding the LASSO estimator is the *least angle regression procedure*. In contrast to forward stepwise regression, a predictor is not fully added to the model. The coefficient of that predictor is increased only until that predictor is no longer the one which is most correlated with the residual r. Then some other competing predictor is pushed by increasing its parameter.

1. Start with all coefficients β_j equal to zero.

2. Find the predictor x_j most correlated with y. Increase the coefficient β_j in the direction of the sign of its correlation with y. Take residuals $r = y - \hat{y}$ and compute correlations. Stop when some other predictor x_k has the same correlation with r than x_j.

3. Increase (β_j, β_k) in their joint least squares direction, until some other predictor x_m has the same correlation with the residual r.

4. Until: all predictors are in the model

This procedure gives the entire path of LASSO solutions if one modification is made. This modification is: if a non-zero coefficient is set to zero, remove it from the active set of predictors and recompute the joint direction.

Lasso is consistent if $\gamma/n \xrightarrow{n} 0$ Knight and Fu [2000].

LASSO is implemented in the R package `lars` which can be used to fit least angle regression, LASSO, and infinitesimal forward stagewise regression models.

5.6.4 Elastic Net

The L^1-norm has also disadvantages. For many features m and few samples n, only the first n features are selected. For correlated variables LASSO only selects one variable and does not use the others. *Elastic net* is a compromise between ridge regression and LASSO. It has both an L^1-norm as well as an L^2-norm regularizer. The objective is

$$\|\boldsymbol{X}\boldsymbol{\beta} \ - \ \boldsymbol{y}\|^2 \ + \ \gamma \ \|\boldsymbol{\beta}\|_1 \ + \ \delta \ \|\boldsymbol{\beta}\|_2^2 \, . \tag{5.239}$$

The elastic net estimator minimizes this objective.

The problem is that now two hyperparameters are introduced. If $\gamma = 0$ then the elastic net is ridge regression. If $\delta = 0$ then the elastic net is LASSO.

The elastic net is consistent if $\gamma/n \xrightarrow{n} 0$ Knight and Fu [2000].

Elastic net is implemented in the R package `glmnet`. This package allows to fit a generalized linear model via penalized maximum likelihood. The regularization path is computed for the LASSO or elastic net penalty at a grid of values for the regularization parameter γ.

5.6.5 Examples

5.6.5.1 Example: Ridge Regression, LASSO, Elastic Net

We generate data with highly correlated explanatory variables:

```
x1 <- rnorm(20)
x2 <- rnorm(20,mean=x1,sd=.01)
y <- rnorm(20,mean=3+x1+x2)

cor(cbind(x1,x2,y))
          x1        x2         y
x1 1.0000000 0.9999319 0.8927331
x2 0.9999319 1.0000000 0.8919416
y  0.8927331 0.8919416 1.0000000
```

The data is shown as pairs of scatter plots in Fig. 5.19.

First we fit a standard linear model:

```
l1 <- lm(y~x1+x2)$coef

summary(l1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-12.710  -4.842   3.027   1.723   8.941  14.850

l1
(Intercept)          x1          x2
   3.026583   14.854954  -12.711132
```

Next we fit the model with ridge regression:

```
library(MASS)
l2 <- lm.ridge(y~x1+x2,lambda=1)
l2
               x1       x2
2.985240 1.051382 1.011735
```

The ridge regression is much closer to the true parameter values. Fig. 5.20 shows the results. The response data are the wooden-colored squares. Standard least squares gives the green circles while ridge regression gives the orange circles. The noise free data is indicated by crosses. Ridge regression is less prone to overfitting and closer to the crosses and, therefore, it generalizes better.

We are interested in the LASSO solution:

```
library(lars)
l3 <- lars(cbind(x1,x2),y)
```

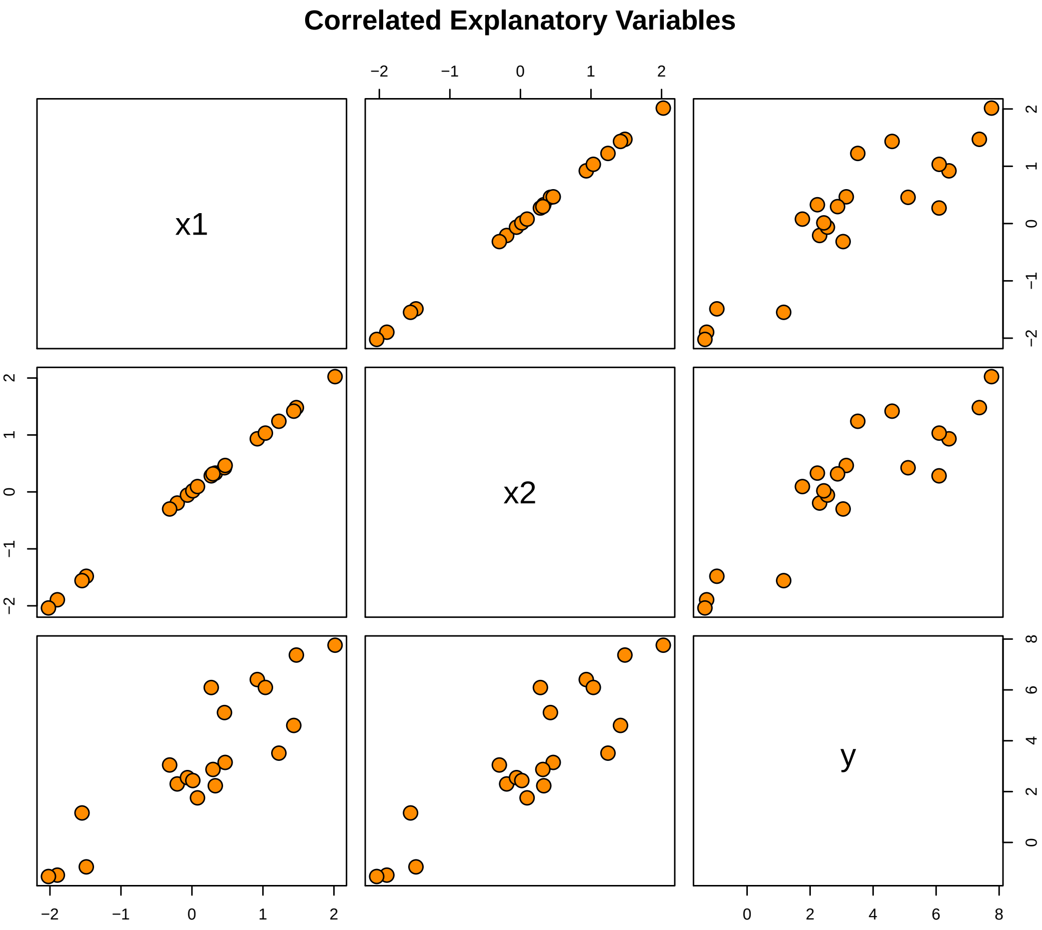

Figure 5.19: An Example for highly correlated explanatory variables.

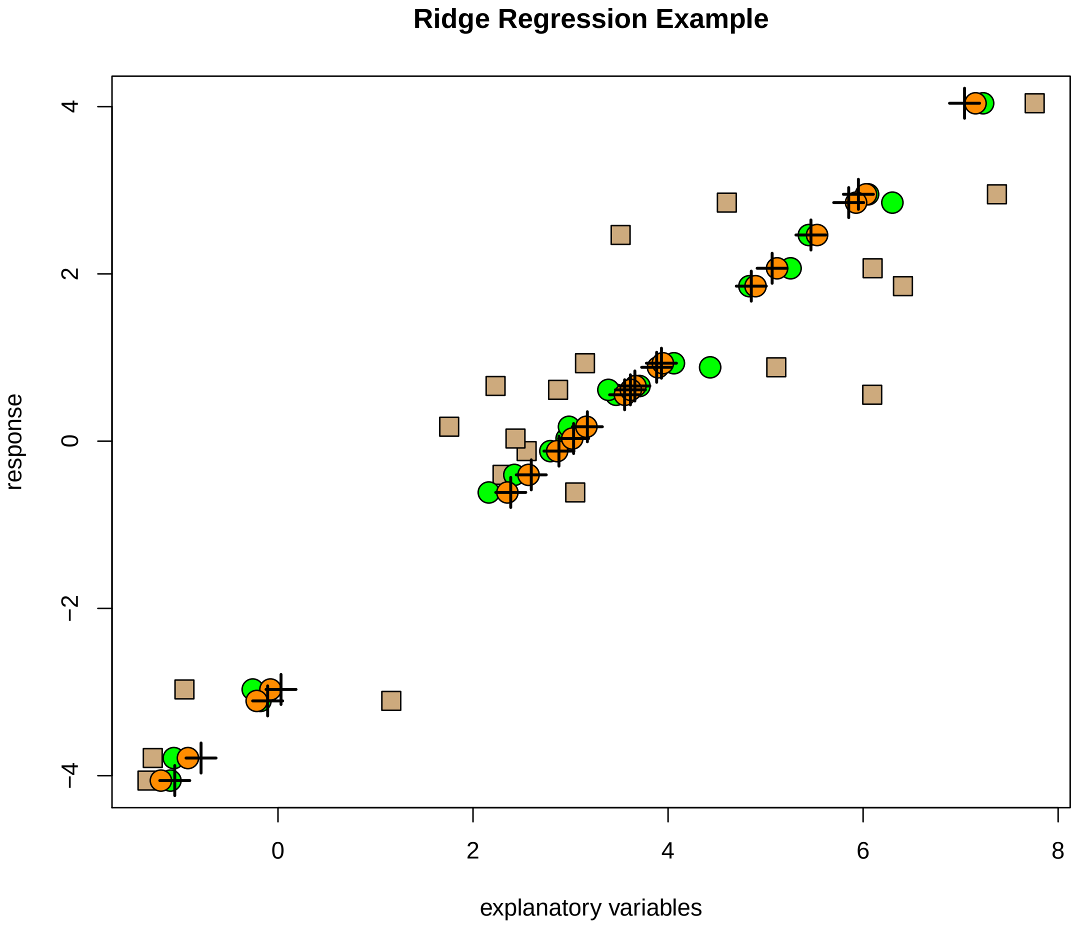

Figure 5.20: Example of ridge regression. The response data are the wooden-colored squares. Standard least squares gives the green circles while ridge regression gives the orange circles. The noise free data is indicated by crosses. Ridge regression is less prone to overfitting and closer to the crosses and, therefore, it generalizes better.

```
l3

Call:
lars(x = cbind(x1, x2), y = y)
R-squared: 0.801
Sequence of LASSO moves:
     x1 x2
Var   1  2
Step  1  2

summary(l3)
LARS/LASSO
Call: lars(x = cbind(x1, x2), y = y)
  Df     Rss      Cp
0  1 138.062 67.3827
1  2  28.030  1.3351
2  3  27.489  3.0000

l3$beta
         x1        x2
0  0.000000   0.00000
1  2.116893   0.00000
2 14.854954 -12.71113
attr(,"scaled:scale")
[1] 4.953151 4.963644

predict(l3,rbind(c(0.0,0.0)))$fit
[1] 3.244128 2.982374 3.026583
```

The last call `predict(l3,rbind(c(0.0,0.0)))$fit` supplies the intercepts for the LASSO solutions. Since in step 2 the residual does not change much compared to step 3 which all variables, we select step 2 solution $y = 2.982374 + 2.116893 x_1$. The solution is shown in Fig. 5.21. LASSO is almost as good as ridge regression since the orange circles are covered by the blue circles obtained from LASSO. However, LASSO used only one explanatory variable.

A call to elastic net, where the L^1 and the L^2 norms are equally weighted ($\alpha = 0.5$), is:

```
library(glmnet)
l4 <- glmnet(cbind(x1,x2),y,alpha=0.5)

summary(l4$lambda)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.002078 0.014340 0.098850 0.628400 0.681200 4.691000
```

λ is the factor which weighs the penalty term that includes both L^1 and the L^2 norm.

We choose a small penalty term:

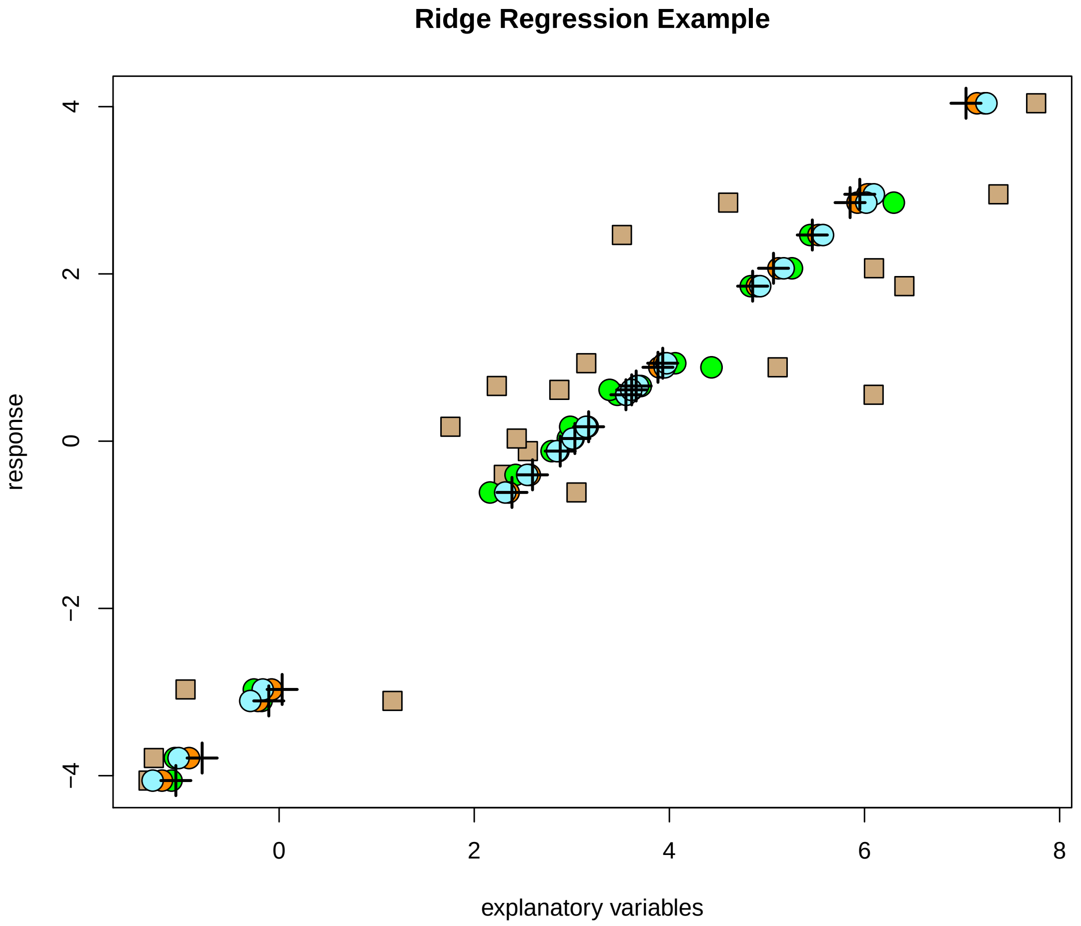

Figure 5.21: Example of LASSO. The same figure as Fig. 5.20 except that now LASSO with only one variable is shown (blue circles). This solution is almost as good as the ridge regression because the orange circles are covered by the blue circles. However, LASSO used only one explanatory variable.

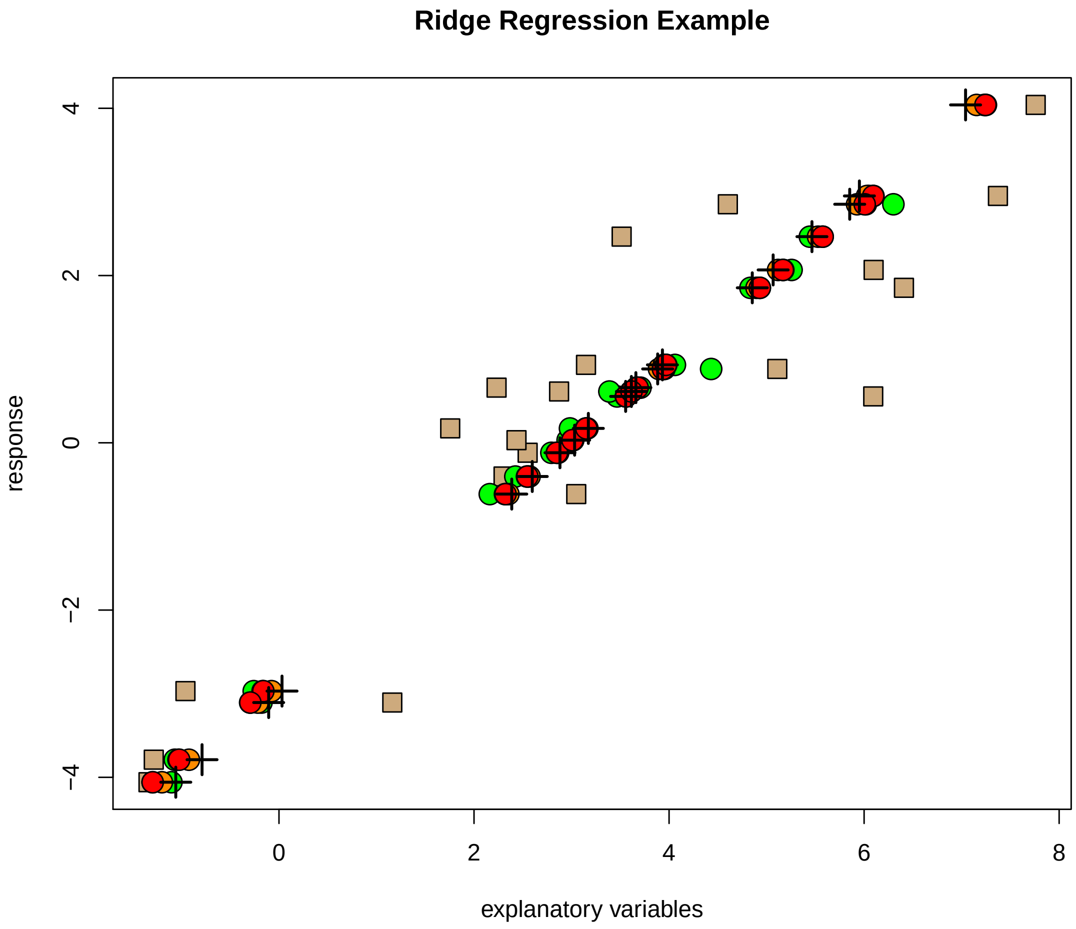

Figure 5.22: Example of elastic net. The same figure as Fig. 5.21 except that now elastic net with $\alpha = 0.5$ is shown (red circles). This solution does not differ much from the LASSO solution because the red circles overlay the blue circles.

```
coef(l4,s=0.004)
3 x 1 sparse Matrix of class "dgCMatrix"
            1
(Intercept) 2.981441
x1          1.738632
x2          0.374484
```

The elastic net solution is shown in Fig. 5.22. This solution does not differ much from the LASSO solution because the red circles overlay the blue circles.

5.6.5.2 Example: Diabetes using Least Angle Regression

The data contain blood and other measurements in diabetics and are taken from Efron, Hastie, Johnstone and Tibshirani (2003) “Least Angle Regression”, *Annals of Statistics*. The diabetes data frame has 442 rows and 3 columns:

1. x: a matrix with 10 columns with explanatory variables “age”, “sex2”, “bmi”, “map”, “tc2”, “ldl”, “hdl”, “tch”, “ltg”, “glu”. That is age, sex, body mass index (bmi), and blood measurements like cholesterol levels (ldl and hdl) etc.
2. y: a numeric vector,
3. x_2: a matrix with 64 columns which contains all explanatory variables, their squared values, and measurements of interaction effects.

The x matrix has been standardized to have unit L^2 norm in each column and zero mean. The matrix x_2 consists of x plus certain interactions.

```
library(lars)
data(diabetes)
x <- diabetes$x
y <- diabetes$y
x2 <- diabetes$x2
op <- par(mfrow=c(2,2))
object1 <- lars(x,y,type="lasso")
plot(object1)
object2 <- lars(x,y,type="lar")
plot(object2)
object3 <- lars(x,y,type="forward.stagewise") # Can use abbreviations
plot(object3)
object4 <- lars(x,y,type="stepwise")
plot(object4)
par(op)
```

Fig. 5.23 shows coefficients at different steps for the diabetes data set fitted by LASSO, least angle regression, forward stagewise, and forward stepwise.

In the following, the different solution paths for the different methods are listed (LASSO, least angle regression, forward stagewise, and forward stepwise):

```
     age  sex  bmi  map   tc  ldl  hdl  tch  ltg   glu
[1,]   0    0    0    0    0    0    0    0    0     0
[2,]   0    0   60    0    0    0    0    0    0     0
[3,]   0    0  362    0    0    0    0    0  302     0
[4,]   0    0  435   79    0    0    0    0  375     0
[5,]   0    0  506  191    0    0 -114    0  440     0
[6,]   0  -75  511  234    0    0 -170    0  451     0
```

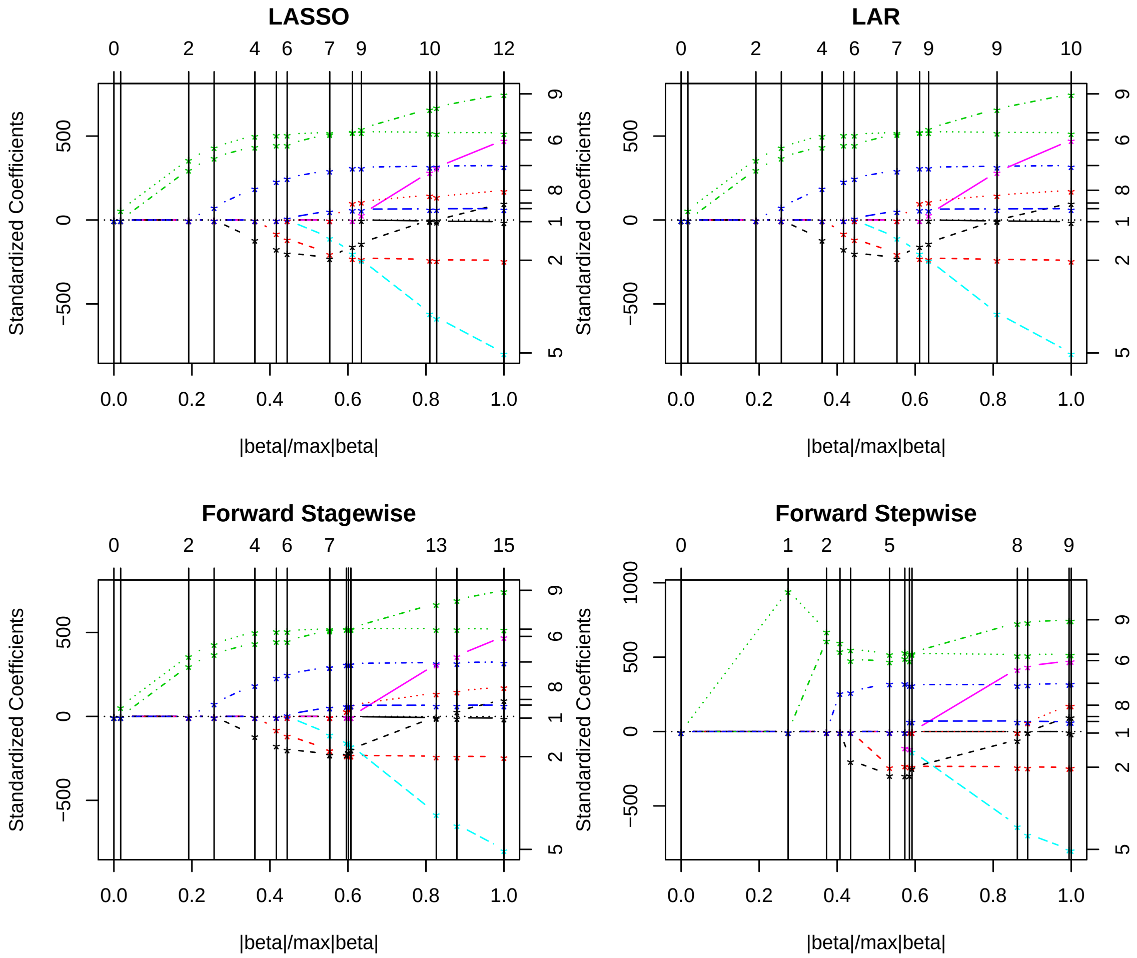

Figure 5.23: Example of least angle regression. The diabetes data set was fitted by LASSO, least angle regression, forward stagewise, and forward stepwise. The figure shows the coefficients that obtain certain values at certain steps.

```
 [7,]    0 -112  512  253    0    0 -196    0  452    12
 [8,]    0 -198  522  297 -104    0 -224    0  515    55
 [9,]    0 -226  527  314 -195    0 -152  106  530    64
[10,]    0 -227  526  315 -237   34 -135  111  545    65
[11,]   -6 -234  523  320 -554  287    0  149  663    66
[12,]   -7 -237  521  322 -580  314    0  140  675    67
[13,]  -10 -240  520  324 -792  477  101  177  751    68

       age  sex  bmi  map   tc  ldl  hdl  tch  ltg   glu
 [1,]    0    0    0    0    0    0    0    0    0     0
 [2,]    0    0   60    0    0    0    0    0    0     0
 [3,]    0    0  362    0    0    0    0    0  302     0
 [4,]    0    0  435   79    0    0    0    0  375     0
 [5,]    0    0  506  191    0    0 -114    0  440     0
 [6,]    0  -75  511  234    0    0 -170    0  451     0
 [7,]    0 -112  512  253    0    0 -196    0  452    12
 [8,]    0 -198  522  297 -104    0 -224    0  515    55
 [9,]    0 -226  527  314 -195    0 -152  106  530    64
[10,]    0 -227  526  315 -237   34 -135  111  545    65
[11,]  -10 -240  520  324 -792  477  101  177  751    68

       age  sex  bmi  map   tc  ldl  hdl  tch  ltg   glu
 [1,]    0    0    0    0    0    0    0    0    0     0
 [2,]    0    0   60    0    0    0    0    0    0     0
 [3,]    0    0  362    0    0    0    0    0  302     0
 [4,]    0    0  435   79    0    0    0    0  375     0
 [5,]    0    0  506  191    0    0 -114    0  440     0
 [6,]    0  -75  511  234    0    0 -170    0  451     0
 [7,]    0 -112  512  253    0    0 -196    0  452    12
 [8,]    0 -198  522  297 -104    0 -224    0  515    55
 [9,]    0 -198  522  297 -104    0 -224    0  515    55
[10,]    0 -230  522  313 -148    0 -224   35  524    65
[11,]    0 -231  522  315 -159    0 -211   50  526    66
[12,]    0 -231  522  315 -159    0 -211   50  526    66
[13,]   -1 -232  523  316 -172    0 -195   68  528    66
[14,]   -1 -232  523  316 -172    0 -195   68  528    66
[15,]   -8 -238  523  322 -644  362   31  151  697    67
[16,]  -10 -240  520  324 -792  477  101  177  751    68

       age  sex  bmi  map   tc  ldl  hdl  tch  ltg   glu
 [1,]    0    0    0    0    0    0    0    0    0     0
 [2,]    0    0  949    0    0    0    0    0    0     0
 [3,]    0    0  675    0    0    0    0    0  615     0
 [4,]    0    0  603  262    0    0    0    0  544     0
 [5,]    0    0  555  270    0    0 -194    0  485     0
 [6,]    0 -236  524  326    0    0 -289    0  474     0
```

x_1	-2	2	-2	2
x_2	3	-3	1	-1
t	1	-1	-1	1

Table 5.17: A toy example where variable x_1 is relevant because $t = x_1 + x_2$ but has no target correlation.

```
 [7,]    0 -227  538  328    0 -103 -291    0  498     0
 [8,]    0 -233  527  315    0 -111 -289    0  479    70
 [9,]    0 -236  518  316 -632  423  -55    0  732    71
[10,]    0 -241  520  322 -791  474  100  177  750    66
[11,]  -10 -240  520  324 -792  477  101  177  751    68
```

The final solution is the same. The variables that were selected first and second agree between the different methods. The first variable that has been selected is body mass index followed by "lgt" and then "map" and thereafter "hdl".

The features that are selected for the combined variables in x_2 are:

```
objectN <- lars(x2,y,type="lar")
name <- colnames(x2)
name[which(abs(objectN$beta[2,])>0)]
[1] "bmi"
name[which(abs(objectN$beta[3,])>0)]
[1] "bmi" "ltg"
name[which(abs(objectN$beta[4,])>0)]
[1] "bmi" "map" "ltg"
name[which(abs(objectN$beta[5,])>0)]
[1] "bmi" "map" "hdl" "ltg"
name[which(abs(objectN$beta[6,])>0)]
[1] "bmi"     "map"     "hdl"     "ltg"     "bmi:map"
name[which(abs(objectN$beta[7,])>0)]
[1] "bmi"     "map"     "hdl"     "ltg"     "age:sex" "bmi:map"
```

The most important variables are the variables which were identified previously.

5.6.5.3 Example: Relevant Variable but No Correlation to Response

We demonstrate on a toy example that relevant variables may be not correlated to the response / target variable. The toy example is shown in Tab. 5.17.

We now perform least squares regression, ridge regression, and LASSO:

```
x1 <- c(-2,2,-2,2)
x2 <- c(3,-3,1,-1)
x <- cbind(x1,x2)
```

x_1	0	1	-1	1
x_2	-1	1	0	0
x_3	0	0	-1	1
t	-1	1	-1	1

Table 5.18: A toy example where variable x_1 is irrelevant because $t = x_2 + x_3$ but has high target correlation.

```
t <- c(1,-1,-1,1)
cor(cbind(t,x1,x2))
          t        x1         x2
t  1.0000000  0.0000000  0.4472136
x1 0.0000000  1.0000000 -0.8944272
x2 0.4472136 -0.8944272  1.0000000

#t <- x1+x2

lm(t~x1+x2)$coef
  (Intercept)           x1           x2
-8.326673e-17  1.000000e+00  1.000000e+00

lm.ridge(t~x1+x2,lambda=1)
(Intercept)      x1        x2
0.0000000 0.2622951 0.3278689

e1 <- lars(x,t)
e2 <- lars(x,t,type="lar")
e3 <- lars(x,t,type="for") # Can use abbreviations
op <- par(mfrow=c(2,2))
plot(e1)
plot(e2)
plot(e3)
par(op)
```

Fig. 5.24 shows the solution paths for different LASSO fitting methods. The variable x_1 is always selected in the second step even if it is not correlated to the response variable.

5.6.5.4 Example: Irrelevant Variable but High Correlation to Response

We demonstrate on a toy example that irrelevant variables may be correlated to the response / target variable. The toy example is shown in Tab. 5.18.

Again we fit the data by least squares regression, ridge regression, and LASSO:

```
x1 <- c(0,1,-1,1)
```

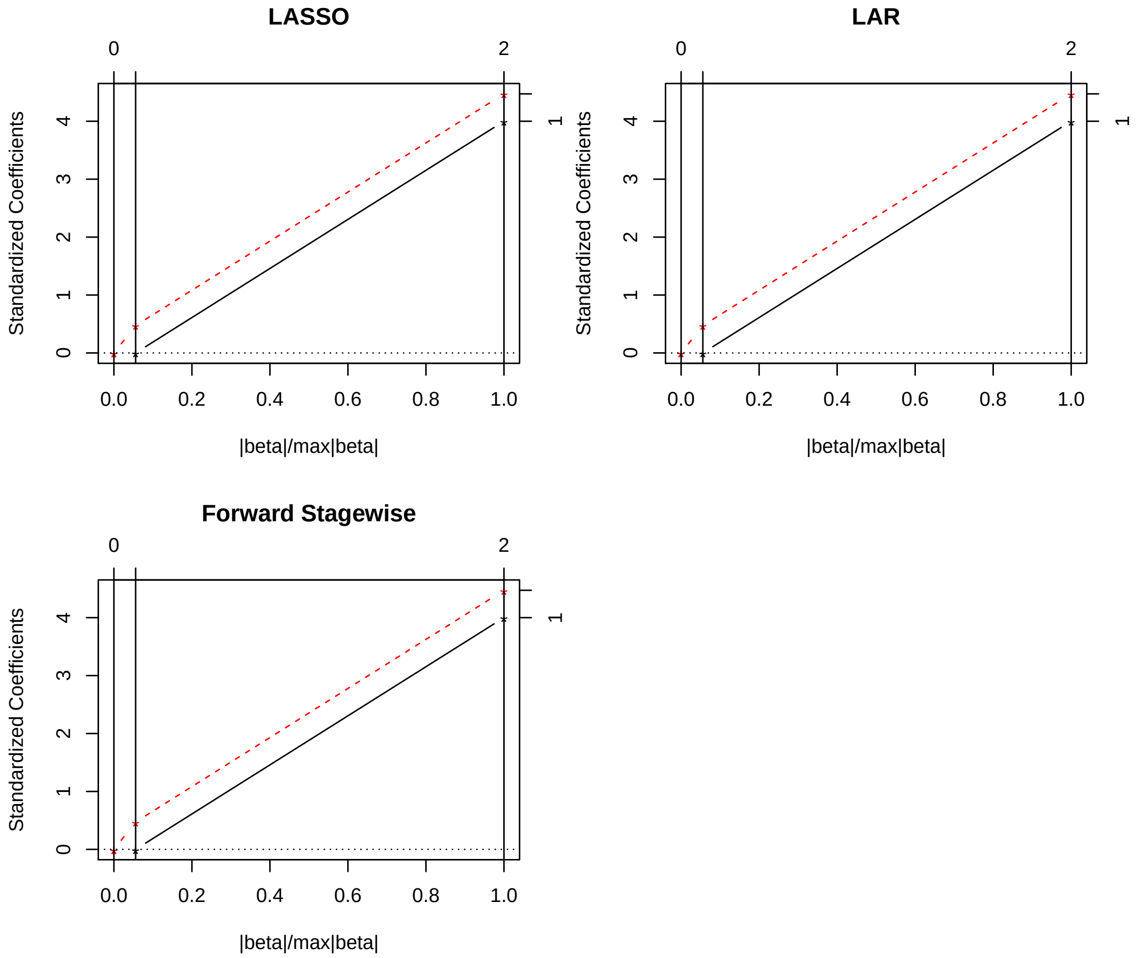

Figure 5.24: A toy example where variable x_1 is relevant because $t = x_1 + x_2$ but has not target correlation. The solution paths for different LASSO fitting methods. The variable x_1 is selected in the second step.

```
x2 <- c(-1,1,0,0)
x3 <- c(0,0,-1,1)
x <- cbind(x1,x2,x3)
t <- c(-1,1,-1,1)
cor(cbind(t,x1,x2,x3))
          t        x1        x2        x3
t  1.0000000 0.9045340 0.7071068 0.7071068
x1 0.9045340 1.0000000 0.4264014 0.8528029
x2 0.7071068 0.4264014 1.0000000 0.0000000
x3 0.7071068 0.8528029 0.0000000 1.0000000

#t1 <- x2+x3

lm(t~x1+x2+x3)$coef
  (Intercept)            x1            x2            x3
-1.171607e-16  4.686428e-16  1.000000e+00  1.000000e+00

lm.ridge(t~x1+x2+x3,lambda=1)
(Intercept)         x1         x2         x3
-0.1043478  0.4173913  0.6330435  0.4660870

e1 <- lars(x,t)
e2 <- lars(x,t,type="lar")
e3 <- lars(x,t,type="for") # Can use abbreviations
plot(e1)
plot(e2)
plot(e3)
par(op)
```

Least squares finds the correct solution while ridge regression uses the highly correlated variable to reduce the overall squared sum of coefficients (to obtain small regularization terms). Fig. 5.25 shows the solution paths for different LASSO fitting methods. The variable x_1 is selected first but in the last step correctly removed.

5.6.5.5 Gas Vapor: Ridge Regression and LASSO

This data set is from Rencher and Schaalje [2008] page 182, Ex. 7.53, Table 7.3, and originally from Weisberg (1985), page 138. When gasoline is pumped into the tank of a car, vapors are vented into the atmosphere. An experiment was conducted to determine whether the response y, the amount of vapor, can be predicted using the following four variables based on initial conditions of the tank and the dispensed gasoline:

1. $x_1 =$ tank temperature (°F),

2. $x_2 =$ gasoline temperature (°F),

3. $x_3 =$ vapor pressure in tank (psi),

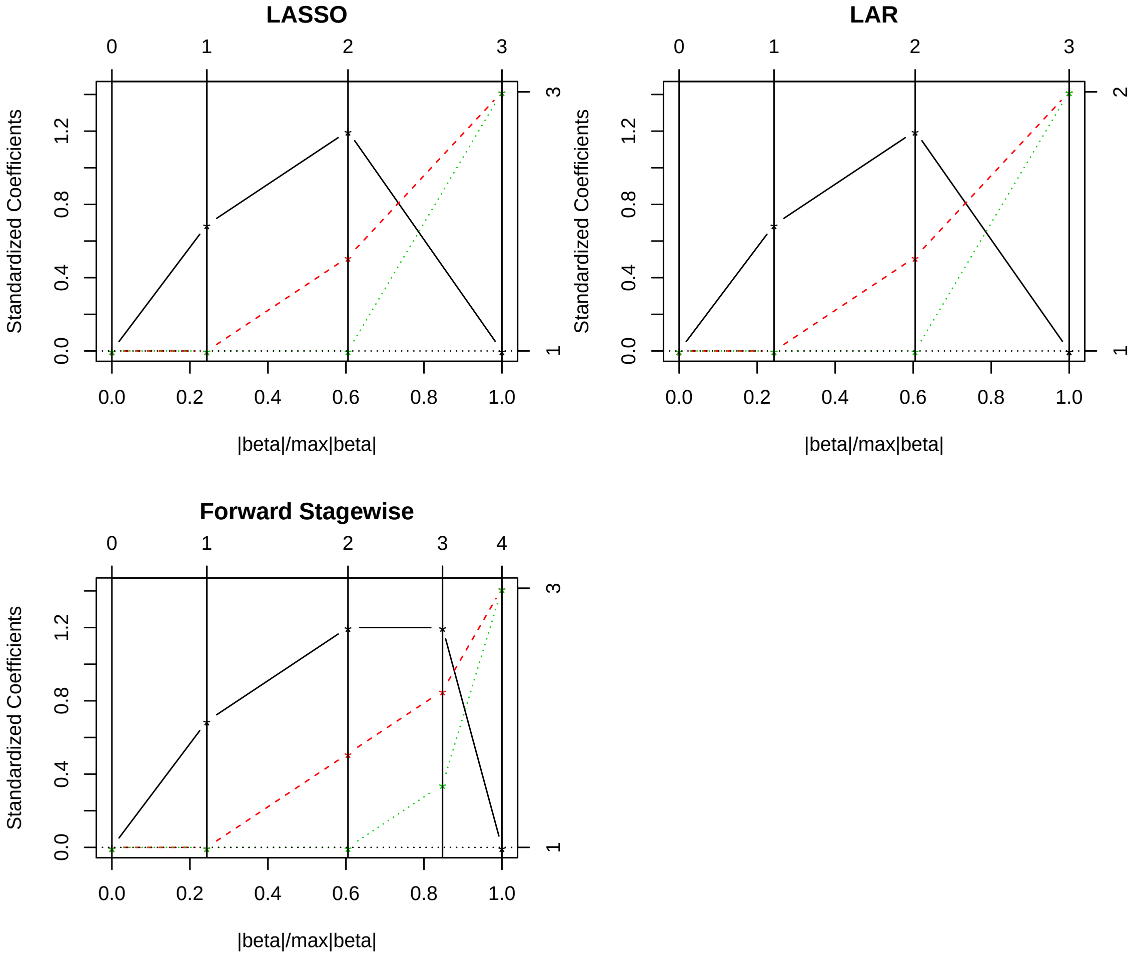

Figure 5.25: A toy example where variable x_1 is irrelevant because $t = x_2 + x_3$ but has high target correlation. The solution paths for different LASSO fitting methods are shown. The variable x_1 is selected first but in the last step correctly removed.

y	x_1	x_2	x_3	x_4	y	x_1	x_2	x_3	x_4
29	33	53	3.32	3.42	40	90	64	7.32	6.70
24	31	36	3.10	3.26	46	90	60	7.32	7.20
26	33	51	3.18	3.18	55	92	92	7.45	7.45
22	37	51	3.39	3.08	52	91	92	7.27	7.26
27	36	54	3.20	3.41	29	61	62	3.91	4.08
21	35	35	3.03	3.03	22	59	42	3.75	3.45
33	59	56	4.78	4.57	31	88	65	6.48	5.80
34	60	60	4.72	4.72	45	91	89	6.70	6.60
32	59	60	4.60	4.41	37	63	62	4.30	4.30
34	60	60	4.53	4.53	37	60	61	4.02	4.10
20	34	35	2.90	2.95	33	60	62	4.02	3.89
36	60	59	4.40	4.36	27	59	62	3.98	4.02
34	60	62	4.31	4.42	34	59	62	4.39	4.53
23	60	36	4.27	3.94	19	37	35	2.75	2.64
24	62	38	4.41	3.49	16	35	35	2.59	2.59
32	62	61	4.39	4.39	22	37	37	2.73	2.59

Table 5.19: Rencher's gas vapor data from Rencher and Schaalje [2008] and originally from Weisberg (1985).

4. $x_4 =$ vapor pressure of gasoline (psi).

The data are given in Tab. 5.19.

We analyze these data in R . First we define the data set:

```
m <- matrix(c(
29,33,53,3.32,3.42,
24,31,36,3.10,3.26,
26,33,51,3.18,3.18,
22,37,51,3.39,3.08,
27,36,54,3.20,3.41,
21,35,35,3.03,3.03,
33,59,56,4.78,4.57,
34,60,60,4.72,4.72,
32,59,60,4.60,4.41,
34,60,60,4.53,4.53,
20,34,35,2.90,2.95,
36,60,59,4.40,4.36,
34,60,62,4.31,4.42,
23,60,36,4.27,3.94,
24,62,38,4.41,3.49,
32,62,61,4.39,4.39,
40,90,64,7.32,6.70,
46,90,60,7.32,7.20,
55,92,92,7.45,7.45,
52,91,92,7.27,7.26,
29,61,62,3.91,4.08,
22,59,42,3.75,3.45,
31,88,65,6.48,5.80,
45,91,89,6.70,6.60,
37,63,62,4.30,4.30,
37,60,61,4.02,4.10,
33,60,62,4.02,3.89,
27,59,62,3.98,4.02,
34,59,62,4.39,4.53,
19,37,35,2.75,2.64,
16,35,35,2.59,2.59,
22,37,37,2.73,2.59),ncol=5,byrow=TRUE)
y <- m[,1]
x <- m[,2:5]
```

Correlation of the variables often give a first impression which variables might be helpful for prediction:

```
cor(m)
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.8260665 0.9093507 0.8698845 0.9213333
[2,] 0.8260665 1.0000000 0.7742909 0.9554116 0.9337690
[3,] 0.9093507 0.7742909 1.0000000 0.7815286 0.8374639
[4,] 0.8698845 0.9554116 0.7815286 1.0000000 0.9850748
[5,] 0.9213333 0.9337690 0.8374639 0.9850748 1.0000000
```

The response y is highly correlated with all explanatory variables which in turn are correlated among themselves. y is most correlated with x_4 followed by x_2. x_4 is very highly correlated with x_3 and least with x_2.

We start with standard least squares regression:

```
lm(y ~ x)

Call:
l1 <- lm(formula = y ~ x)
l1
Coefficients:
(Intercept)           x1           x2           x3           x4
    1.01502     -0.02861      0.21582     -4.32005      8.97489

anova(l1)
Analysis of Variance Table

Response: y
```

```
          Df  Sum Sq Mean Sq F value    Pr(>F)
x          4 2520.27  630.07   84.54 7.249e-15 ***
Residuals 27  201.23    7.45
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The variables x_3 and x_4 seem to be relevant. We know that they are highly correlated and lead to overfitting effects.

The relevance of the variables is checked by ridge regression which deals with these highly correlated variables:

```
l2 <- lm.ridge(y ~ x,lambda=1)
l2
                     x1          x2          x3          x4
 0.72339986 -0.04937793  0.27780519  0.35225191  3.74029965
```

Here variable x_4 sticks out.

Next we analyze the data set by LASSO:

```
la <- lars(x,y,type="lar")
la$beta[2,]
[1] 0.0000000 0.0000000 0.0000000 0.4963341
la$beta[3,]
[1] 0.0000000 0.2695754 0.0000000 3.5437050
la$beta[4,]
[1] -0.06804859  0.27044138  0.00000000  4.48953562
```

Here it becomes clear that x_4 is the most important variable and next the less correlated variable x_2 is selected.

We perform feature selection and use only the variables x_2 and x_4:

```
l3 <- lm(formula = y ~ x[,c(2,4)])
l3

Call:
lm(formula = y ~ x[, c(2, 4)])

Coefficients:
  (Intercept)  x[, c(2, 4)]1  x[, c(2, 4)]2
       0.1918         0.2747         3.6020

anova(l3)
Analysis of Variance Table

Response: y
```

```
             Df  Sum Sq Mean Sq F value    Pr(>F)
x[, c(2, 4)]  2 2483.11 1241.56  151.04 4.633e-16 ***
Residuals    29  238.39    8.22
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We now compare the full model with the model where only two features are selected:

```
anova(l1,l3)
Analysis of Variance Table

Model 1: y ~ x
Model 2: y ~ x[, c(2, 4)]
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     27 201.23
2     29 238.39 -2   -37.159 2.4929 0.1015
```

The model with only two features does not perform significantly worse.

We want to check which model is better suited by Akaike's information criterion (AIC):

```
extractAIC(l1)
[1]  5.00000 68.83842
extractAIC(l3)
[1]  3.00000 70.26103
```

The model with only two variables should be chosen.

5.6.5.6 Chemical Reaction: Ridge Regression and LASSO

This data set is from Rencher and Schaalje [2008] page 182, Ex. 7.54, Table 7.4 and originally from Box and Youle (1955) and was also used in Andrews and Herzberg (1985), page 188. The yield in a chemical reaction should be maximized, therefore the values of the following variables were used to control the experiment:

1. $x_1 =$ temperature (°C),
2. $x_2 =$ concentration of a reagent (%),
3. $x_3 =$ time of reaction (hours).

The response variables were:

1. $y_1 =$ percent of unchanged starting material,
2. $y_2 =$ percent converted to the desired material.

y_1	y_2	x_1	x_2	x_3
41.5	45.9	162	23	3
33.8	53.3	162	23	8
27.7	57.5	162	30	5
21.7	58.8	162	30	8
19.9	60.6	172	25	5
15.0	58.0	172	25	8
12.2	58.6	172	30	5
4.3	52.4	172	30	8
19.3	56.9	167	27.5	6.5
6.4	55.4	177	27.5	6.5
37.6	46.9	157	27.5	6.5
18.0	57.3	167	32.5	6.5
26.3	55.0	167	22.5	6.5
9.9	58.9	167	27.5	9.5
25.0	50.3	167	27.5	3.5
14.1	61.1	177	20	6.5
15.2	62.9	177	20	6.5
15.9	60.0	160	34	7.5
19.6	60.6	160	34	7.5

Table 5.20: Rencher's chemical reaction data from Rencher and Schaalje [2008].

The data are given in Tab. 5.20.

First we define the data and check the correlation among the variables:

```
m <- matrix(c(
+ 41.5,45.9,162,23,3,
+ 33.8,53.3,162,23,8,
+ 27.7,57.5,162,30,5,
+ 21.7,58.8,162,30,8,
+ 19.9,60.6,172,25,5,
+ 15.0,58.0,172,25,8,
+ 12.2,58.6,172,30,5,
+ 4.3,52.4,172,30,8,
+ 19.3,56.9,167,27.5,6.5,
+ 6.4,55.4,177,27.5,6.5,
+ 37.6,46.9,157,27.5,6.5,
+ 18.0,57.3,167,32.5,6.5,
+ 26.3,55.0,167,22.5,6.5,
+ 9.9,58.9,167,27.5,9.5,
+ 25.0,50.3,167,27.5,3.5,
+ 14.1,61.1,177,20,6.5,
+ 15.2,62.9,177,20,6.5,
+ 15.9,60.0,160,34,7.5,
+ 19.6,60.6,160,34,7.5),ncol=5,byrow=TRUE)
y1 <- m[,1]
y2 <- m[,2]
x <- m[,3:5]
cor(m)
           [,1]        [,2]        [,3]        [,4]        [,5]
[1,]  1.0000000 -0.60782343 -0.67693865 -0.22472586 -0.45253956
[2,] -0.6078234  1.00000000  0.40395099  0.07998377  0.39273121
[3,] -0.6769387  0.40395099  1.00000000 -0.46200145 -0.02188275
[4,] -0.2247259  0.07998377 -0.46200145  1.00000000  0.17665667
[5,] -0.4525396  0.39273121 -0.02188275  0.17665667  1.00000000
```

The first response variable has negative correlation to the first regressor and less negative correlation to the third regressor. The second response variable is negatively correlated to the first response variable which was to be expected. The second response variable is equally correlated to the first and third regressor.

We start with a least square estimator:

```
l1 <- lm(y1 ~ x)
l1

Call:
lm(formula = y1 ~ x)

Coefficients:
(Intercept)           x1           x2           x3
    332.111       -1.546       -1.425       -2.237

anova(l1)
Analysis of Variance Table

Response: y1
          Df  Sum Sq Mean Sq F value    Pr(>F)
x          3 1707.16  569.05  106.47 2.459e-10 ***
Residuals 15   80.17    5.34
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

All variables are relevant for prediction. x_3 is the most relevant variable.

We perform regularization using ridge regression:

```
l2 <- lm.ridge(y1 ~ x,lambda=1)
l2
    x1         x2         x3
307.512361  -1.424838  -1.279060  -2.179261
```

The figure did not change compared to standard least squares estimation. This is a hint that indeed all variables are required.

Next we perform LASSO:

```
la <- lars(x,y1,type="lar")
la$beta[2,]
[1] -0.3518723  0.0000000  0.0000000
la$beta[3,]
[1] -0.5182233  0.0000000 -0.6334936
```

The first and last variable seem to be the most relevant ones.

We fit a least squares model with the two most important variables:

```
l3 <- lm(formula = y1 ~ x[,c(1,3)])
l3

Call:
lm(formula = y1 ~ x[, c(1, 3)])

Coefficients:
  (Intercept)  x[, c(1, 3)]1  x[, c(1, 3)]2
      222.957         -1.101         -2.853


anova(l3)
Analysis of Variance Table

Response: y1
             Df  Sum Sq Mean Sq F value    Pr(>F)
x[, c(1, 3)]  2 1209.61  604.81   16.75 0.0001192 ***
Residuals    16  577.72   36.11
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

An ANOVA table shows that all variables are required to predict the response:

```
anova(l1,l3)
Analysis of Variance Table

Model 1: y1 ~ x
Model 2: y1 ~ x[, c(1, 3)]
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     15  80.17
2     16 577.72 -1   -497.55 93.088 7.988e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We move on the second response variable y_2, that is, the converted material to the desired product. We start again with least squares:

```
l12 <- lm(y2 ~ x)
l12

Call:
lm(formula = y2 ~ x)

Coefficients:
(Intercept)           x1           x2           x3
   -26.0353       0.4046       0.2930       1.0338

anova(l12)
Analysis of Variance Table

Response: y2
          Df Sum Sq Mean Sq F value  Pr(>F)
x          3 151.00  50.334  3.0266 0.06235 .
Residuals 15 249.46  16.631
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Again x_3 is the most relevant variable but now even more dominant.

Next we perform ridge regression:

```
l22 <- lm.ridge(y2 ~ x,lambda=1)
l22
      x1          x2          x3
-19.9403245   0.3747668   0.2617700   0.9933463
```

The figure remains the same for ridge regression.

We perform fitting with LASSO:

```
la2 <- lars(x,y2,type="lar")
la2$beta[2,]
[1] 0.008327752 0.000000000 0.000000000
la2$beta[3,]
[1] 0.1931751 0.0000000 0.7039310
```

Interestingly, x_1 is selected before x_3. Looking at the correlation matrix, we see that indeed x_1 is more correlated to y_2 than x_3 (0.40 vs. 0.39).

If we select the two variables which would be first selected by LASSO, then we have for the least squares fit:

```
l32 <- lm(formula = y2 ~ x[,c(1,3)])
l32

Call:
lm(formula = y2 ~ x[, c(1, 3)])

Coefficients:
  (Intercept)  x[, c(1, 3)]1  x[, c(1, 3)]2
      -3.5856         0.3131         1.1605

anova(l32)
Analysis of Variance Table

Response: y2
             Df Sum Sq Mean Sq F value  Pr(>F)
x[, c(1, 3)]  2 129.96  64.978  3.8433 0.04334 *
Residuals    16 270.51  16.907
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Comparing the full model with the model, where only two features are selected by an ANOVA table gives:

```
anova(l12,l32)
Analysis of Variance Table

Model 1: y2 ~ x
Model 2: y2 ~ x[, c(1, 3)]
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     15 249.46
2     16 270.51 -1   -21.047 1.2655 0.2783
```

Therefore, the model with only two features is not significantly worse than the full model.

5.6.5.7 Land Rent: Ridge Regression and LASSO

This data set is from Rencher and Schaalje [2008] page 184, Ex. 7.55, Table 7.5 and originally from Weisberg (1985) page 162. For 34 counties in Minnesota the following variables were recorded in 1977:

1. y: average rent paid per acre of land with alfalfa,
2. x_1: average rent paid per acre for all land,
3. x_2: average number of dairy cows per square mile,
4. x_3: proportion of farmland in pasture.

y	x_1	x_2	x_3	y	x_1	x_2	x_3
18.38	15.50	17.25	.24	8.50	9.00	8.89	.08
20.00	22.29	18.51	.20	36.50	20.64	23.81	.24
11.50	12.36	11.13	.12	60.00	81.40	4.54	.05
25.00	31.84	5.54	.12	16.25	18.92	29.62	.72
52.50	83.90	5.44	.04	50.00	50.32	21.36	.19
82.50	72.25	20.37	.05	11.50	21.33	1.53	.10
25.00	27.14	31.20	.27	35.00	46.85	5.42	.08
30.67	40.41	4.29	.10	75.00	65.94	22.10	.09
12.00	12.42	8.69	.41	31.56	38.68	14.55	.17
61.25	69.42	6.63	.04	48.50	51.19	7.59	.13
60.00	48.46	27.40	.12	77.50	59.42	49.86	.13
57.50	69.00	31.23	.08	21.67	24.64	11.46	.21
31.00	26.09	28.50	.21	19.75	26.94	2.48	.10
60.00	62.83	29.98	.17	56.00	46.20	31.62	.26
72.50	77.06	13.59	.05	25.00	26.86	53.73	.43
60.33	58.83	45.46	.16	40.00	20.00	40.18	.56
49.75	59.48	35.90	.32	56.67	62.52	15.89	.05

Table 5.21: Rencher's land rent data from Rencher and Schaalje [2008].

The data is shown in Tab. 5.21. A relevant question is: can the rent for alfalfa land be predicted from the other three variables?

We first code the data in R variables and check the correlation:

```
m <- matrix(c(
+ 18.38,15.50,17.25,.24,
+ 20.00,22.29,18.51,.20,
+ 11.50,12.36,11.13,.12,
+ 25.00,31.84,5.54,.12,
+ 52.50,83.90,5.44,.04,
+ 82.50,72.25,20.37,.05,
+ 25.00,27.14,31.20,.27,
+ 30.67,40.41,4.29,.10,
+ 12.00,12.42,8.69,.41,
+ 61.25,69.42,6.63,.04,
+ 60.00,48.46,27.40,.12,
+ 57.50,69.00,31.23,.08,
+ 31.00,26.09,28.50,.21,
+ 60.00,62.83,29.98,.17,
+ 72.50,77.06,13.59,.05,
+ 60.33,58.83,45.46,.16,
+ 49.75,59.48,35.90,.32,
+ 8.50,9.00,8.89,.08,
+ 36.50,20.64,23.81,.24,
+ 60.00,81.40,4.54,.05,
+ 16.25,18.92,29.62,.72,
+ 50.00,50.32,21.36,.19,
+ 11.50,21.33,1.53,.10,
+ 35.00,46.85,5.42,.08,
+ 75.00,65.94,22.10,.09,
+ 31.56,38.68,14.55,.17,
+ 48.50,51.19,7.59,.13,
+ 77.50,59.42,49.86,.13,
+ 21.67,24.64,11.46,.21,
+ 19.75,26.94,2.48,.10,
+ 56.00,46.20,31.62,.26,
+ 25.00,26.86,53.73,.43,
+ 40.00,20.00,40.18,.56,
+ 56.67,62.52,15.89,.05),ncol=4,byrow=TRUE)
y <- m[,1]
x <- m[,2:4]
cor(m)
           [,1]       [,2]      [,3]       [,4]
[1,]  1.0000000  0.8868392 0.2967901 -0.3838808
[2,]  0.8868392  1.0000000 0.0296753 -0.5212982
[3,]  0.2967901  0.0296753 1.0000000  0.4876448
[4,] -0.3838808 -0.5212982 0.4876448  1.0000000

sd(m[,1])
[1] 21.53698
sd(m[,2])
[1] 22.45614
sd(m[,3])
[1] 14.21056
sd(m[,4])
[1] 0.1532131
```

We also computed the standard deviations of the variables because x_3 has smaller values than the other variables. x_3 is about a factor of 100 smaller than the other variables.

We start with a least squares regression:

```
l1 <- lm(y ~ x)
l1

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)           x1           x2           x3
     0.6628       0.7803       0.5031     -17.1002
```

```
anova(l1)
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value    Pr(>F)
x          3 13266.9  4422.3  65.037 3.112e-13 ***
Residuals 30  2039.9    68.0
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

x_3 has the largest coefficient but it has to be divided by a factor of 100 to be in the range of the other variables. Thus, x_3 has actually the smallest influence on the response variable after fitting by least squares.

Ridge regression confirms the observations we had for the least squares estimator:

```
l2 <- lm.ridge(y ~ x,lambda=1)
l2
                     x1          x2          x3
  2.1360609   0.7542789   0.4955992 -18.2104311
```

Since ridge regression penalizes the coefficients for the standardized variables, the absolute coefficient for x_3 even increases. The other two coefficients decrease as they are pushed toward zero by ridge regression.

LASSO confirms our findings:

```
la <- lars(x,y,type="lar")
la$beta[2,]
[1] 0.5832042 0.0000000 0.0000000
la$beta[3,]
[1] 0.7872064 0.3223731 0.0000000
```

The first two explanatory variables are the most relevant. From the correlations we see that the first explanatory variable has largest correlation with the response and is therefore selected first. Interestingly, x_3 has the second largest correlation to the response variable but is not selected. The reason for this is that x_3 has also large correlation to x_1 and does not bring in much new information. In contrast to x_3, x_2 has low correlation to x_1 and brings in new information.

We again fit a least squares model, but now with only the first two explanatory variables:

```
l3 <- lm(formula = y ~ x[,c(1,2)])
l3

Call:
lm(formula = y ~ x[, c(1, 2)])
```

```
Coefficients:
  (Intercept)  x[, c(1, 2)]1  x[, c(1, 2)]2
      -3.3151         0.8428         0.4103

anova(l3)
Analysis of Variance Table

Response: y
             Df  Sum Sq Mean Sq F value    Pr(>F)
x[, c(1, 2)]  2 13159.3  6579.6  94.981 6.015e-14 ***
Residuals    31  2147.5    69.3
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Comparing the full model with the model that has only the first two variables shows that the error difference is not significant:

```
anova(l1,l3)
Analysis of Variance Table

Model 1: y ~ x
Model 2: y ~ x[, c(1, 2)]
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     30 2039.9
2     31 2147.5 -1   -107.58 1.5821 0.2182
```

Therefore the reduce model may be chosen for analysis.

Appendix A

Mean and Median of Symmetric Distributions

A.1 Mean, Median, and Symmetry Point are Equal

The mean μ of a distribution is

$$\mu \;=\; \int_{-\infty}^{\infty} p(x)\; x \; dx \; . \tag{A.1}$$

m is the median of a distribution if

$$\int_{-\infty}^{m} p(x)\; dx \;=\; \int_{m}^{\infty} p(x)\; dx \; . \tag{A.2}$$

We assume a symmetric distribution around s, the symmetry point:

$$p(x+s) \;=\; p(s-x) \; . \tag{A.3}$$

We obtain

$$\begin{aligned} \int_{-\infty}^{s} p(x)\; (x \;-\; s)\; dx \;&=\; \int_{-\infty}^{0} p(a+s)\; a\; da \\ &=\; \int_{\infty}^{0} p(-a+s)\; a\; da \\ &=\; - \int_{0}^{\infty} p(a+s)\; a\; da \end{aligned} \tag{A.4}$$

and

$$\int_{s}^{\infty} p(x)\; (x \;-\; s)\; dx \;=\; \int_{0}^{\infty} p(a+s)\; a\; da \; , \tag{A.5}$$

where we used $a = x - s$.

Thus,

$$\int_{-\infty}^{\infty} p(x)\; (x \;-\; s)\; dx \;=\; 0 \tag{A.6}$$

and, therefore,

$$\int_{-\infty}^{\infty} p(x)\, x\, dx \;=\; s\;. \tag{A.7}$$

The mean is $\mu = s$.

For the median we obtain

$$\begin{aligned}\int_{-\infty}^{s} p(x)\, dx \;&=\; \int_{-\infty}^{0} p(a\,+\,s)\, da \\ &=\; \int_{-\infty}^{0} p(s\,-\,a)\, da \\ &=\; -\int_{\infty}^{0} p(s\,+\,a)\, da \\ &=\; \int_{0}^{\infty} p(s\,+\,a)\, da \\ &=\; \int_{s}^{\infty} p(x)\, dx\;.\end{aligned} \tag{A.8}$$

Thus, the median is $m = s$, too.

A.2 Definitions of Mean, Median, and Variance

We consider the empirical mean, median, and variance of n samples $\boldsymbol{x} = (x_1, \ldots, x_n)$ drawn from $p(x)$. The first, second and fourth central moments are

$$\mu \;=\; \mathrm{E}(x) \;=\; \int_{-\infty}^{\infty} p(x)\, x\, dx \tag{A.9}$$

$$\mu_2 \;=\; \mathrm{E}((x-\mu)^2) \;=\; \int_{-\infty}^{\infty} p(x)\, (x-\mu)^2\, dx \tag{A.10}$$

$$\mu_4 \;=\; \mathrm{E}((x-\mu)^4) \;=\; \int_{-\infty}^{\infty} p(x)\, (x-\mu)^4\, dx\;. \tag{A.11}$$

The variance is sometimes denoted by

$$\sigma^2 \;=\; \mu_2\;. \tag{A.12}$$

The empirical mean is

$$\bar{x} \;=\; \frac{1}{n}\sum_{i=1}^{n} x_i\;, \tag{A.13}$$

the empirical median is

$$m \;=\; \left\{ \begin{array}{cl} (\mathrm{sort}(\boldsymbol{x}))_{(n+1)/2} & \text{for } \;n\; \text{ odd} \\ \frac{1}{2}\,\Big((\mathrm{sort}(\boldsymbol{x}))_{n/2} \,+\, (\mathrm{sort}(\boldsymbol{x}))_{n/2+1}\Big) & \text{for } \;n\; \text{ even} \end{array}\right.\;, \tag{A.14}$$

the biased empirical variance is

$$\mathrm{var}_b(\boldsymbol{x}) \;=\; \frac{1}{n}\sum_{i=1}^{n}(x_i \;-\; \bar{x})^2 \;, \tag{A.15}$$

the unbiased empirical variance is

$$\mathrm{var}_u(\boldsymbol{x}) \;=\; \frac{1}{n-1}\sum_{i=1}^{n}(x_i \;-\; \bar{x})^2 \;. \tag{A.16}$$

the bias of $\mathrm{var}_b(\boldsymbol{x})$ is

$$\begin{aligned} \mathrm{E}\left(\mathrm{var}_b(\boldsymbol{x})\right) \;-\; \sigma^2 \;&=\; \mathrm{E}\left(\frac{n \;-\; 1}{n}\mathrm{var}_u(\boldsymbol{x})\right) \;-\; \sigma^2 \\ &=\; \frac{n \;-\; 1}{n}\,\sigma^2 \;-\; \sigma^2 \\ &=\; -\,\frac{1}{n}\,\sigma^2 \;. \end{aligned} \tag{A.17}$$

A.3 Mean Squared Error for Estimating Mean and Median

The variance of the mean is

$$\mathrm{var}(\bar{x}) \;=\; \frac{\sigma^2}{n} \;. \tag{A.18}$$

The variance of the median is

$$\mathrm{var}(m) \;=\; \frac{1}{4\,p(m)\,n} \;. \tag{A.19}$$

A.3.1 Gaussian: Variance of the Mean and the Median

The variance of the mean for a Gaussian distribution is

$$\mathrm{var}(\bar{x}) \;=\; \frac{\sigma^2}{n} \;. \tag{A.20}$$

The variance of the median for a Gaussian distribution is

$$\mathrm{var}(m) \;=\; \frac{\pi}{2}\,\frac{\sigma^2}{n} \;. \tag{A.21}$$

Consequently, both the mean and the median of a Gaussian should be estimated by the empirical mean.

A.3.2 Laplace: Variance of the Mean and the Median

We assume a Laplacian with density

$$p(x) \;=\; \frac{1}{2\,b}\,\exp\left(-\,1/b\,|x\;-\;\mu|\right) \tag{A.22}$$

with both mean and median equal to μ and variance $\sigma^2 = 2b^2$.

The variance of the mean for a Laplace distribution is

$$\mathrm{var}(\bar{x}) \;=\; \frac{\sigma^2}{n}\;. \tag{A.23}$$

The variance of the median for a Laplace distribution is

$$\mathrm{var}(m) \;=\; \frac{\sigma^2}{2\,n}\;. \tag{A.24}$$

Consequently, both the mean and the median of a Laplacian should be estimated by the empirical median.

A.4 Variance of the Variance

A.4.1 Biased Variance

The variance of the biased variance is

$$\begin{aligned}\mathrm{var}(\mathrm{var}_b(\boldsymbol{x})) \;&=\; \frac{(n\;-\;1)^2}{n^3}\left(\mu_4\;-\;\frac{n\;-\;3}{n\;-\;1}\mu_2^2\right) \\ &=\; \frac{(n\;-\;1)^2}{n^3}\left(\mu_4\;-\;\mu_2^2\;+\;\frac{2}{n\;-\;1}\mu_2^2\right)\;.\end{aligned} \tag{A.25}$$

A.4.2 Unbiased Variance

The variance of the unbiased variance is

$$\mathrm{var}(\mathrm{var}_u(\boldsymbol{x})) \;=\; \frac{1}{n}\left(\mu_4\;-\;\frac{n\;-\;3}{n\;-\;1}\mu_2^2\right) \;=\; \frac{1}{n}\left(\mu_4\;-\;\mu_2^2\;+\;\frac{2}{n\;-\;1}\mu_2^2\right)\;. \tag{A.26}$$

The biased variance has a lower variance, which is lower by a factor of $\frac{(n-1)^2}{n^2} \;=\; 1 - \frac{2}{n} + \frac{1}{n^2}$.

A.4.3 Gaussian Distribution

For the biased case we have

$$\mathrm{var}(\mathrm{var}_b(\boldsymbol{x})) \;=\; \frac{2\,(n\;-\;1)\,\sigma^4}{n^2} \tag{A.27}$$

and for the unbiased case

$$\mathrm{var}(\mathrm{var}_u(\boldsymbol{x})) \;=\; \frac{2\,\sigma^4}{n\,-\,1}\,. \tag{A.28}$$

We now compute the mean squared error. We first consider the biased case:

$$\begin{aligned}\mathrm{mse}_b \;&=\; \mathrm{var}(\mathrm{var}_b(\boldsymbol{x}))\;+\;\mathrm{bias}^2 \;=\; \frac{2\,(n\,-\,1)\,\sigma^4}{n^2}\;+\;\frac{\sigma^4}{n^2} \\ &=\; \frac{(2\,n\,-\,1)\,\sigma^4}{n^2} \;=\; \left(1\,-\,\frac{3\,n\,-\,1}{2\,n^2}\right)\,\frac{2\,\sigma^4}{n\,-\,1}\,.\end{aligned} \tag{A.29}$$

The unbiased case is

$$\mathrm{mse}_u \;=\; \mathrm{var}(\mathrm{var}_u(\boldsymbol{x})) \;=\; \frac{2\,\sigma^4}{n\,-\,1}\,. \tag{A.30}$$

Consequently,

$$\mathrm{mse}_b \;<\; \mathrm{mse}_u\;, \tag{A.31}$$

which means for a Gaussian the biased variance has a lower mean squared error.

A.4.4 Laplace Distribution

For the biased case we have

$$\mathrm{var}(\mathrm{var}_b(\boldsymbol{x})) \;=\; \frac{(n\,-\,1)\,(5\,n\,-\,3)\,\sigma^4}{n^3} \tag{A.32}$$

and for the unbiased case

$$\mathrm{var}(\mathrm{var}_u(\boldsymbol{x})) \;=\; \frac{(5\,n\,-\,3)\,\sigma^4}{n\,(n\,-\,1)}\,. \tag{A.33}$$

We now compute the mean squared error. We first consider the biased case:

$$\begin{aligned}\mathrm{mse}_b \;&=\; \mathrm{var}(\mathrm{var}_b(\boldsymbol{x}))\;+\;\mathrm{bias}^2 \\ &=\; \frac{(n\,-\,1)\,(5\,n\,-\,3)\,\sigma^4}{n^3}\;+\;\frac{\sigma^4}{n^2} \;=\; \frac{(5\,n^3\,-\,12\,n^2\,+\,10\,n\,-3)\,\sigma^4}{n^3\,(n\,-\,1)}\,.\end{aligned} \tag{A.34}$$

The unbiased case is

$$\mathrm{mse}_u \;=\; \mathrm{var}(\mathrm{var}_u(\boldsymbol{x})) \;=\; \frac{(5\,n\,-\,3)\,\sigma^4}{n\,(n\,-\,1)} \;=\; \frac{(5\,n^3\,-\,3\,n^2)\,\sigma^4}{n^3\,(n\,-\,1)}\,. \tag{A.35}$$

Consequently,

$$\mathrm{mse}_b \;<\; \mathrm{mse}_u\;, \tag{A.36}$$

which means for a Laplacian the biased variance has a lower mean squared error, too.

Appendix B

R Code for Plots in Manuscript

The following libraries are required:

```
library(ggplot2)
library(doBy)
```

B.1 Histograms

Histogram 1 in Fig. 3.7:

```
df <- data.frame(counts=x)

ggplot(df, aes(x=counts)) +
  geom_histogram(aes(fill = ..count.. < 10), +
    colour="darkgreen",fill="coral",binwidth=0.5) +
    scale_y_continuous(breaks=c(10,15,20,25,30)) +
    scale_x_continuous(limits=c(0,8)) +
    geom_vline(xintercept=c(6.25),  linetype="dashed", size=1.5,colour="darkgreen") +
    labs(title = "Histogram Iris Data Sepal Lengths") +
    theme(legend.position="none")+
    coord_cartesian(ylim = c(0,33),xlim=c(3.9,8.1)) +
    ylab("Counts") +
    xlab("Sepal Length")
```

Histogram 2 in Fig. 3.7:

```
df <- data.frame(counts=x1)

ggplot(df, aes(x=counts)) +
  geom_histogram(aes(fill = ..count.. < 10), +
    colour="darkgreen",fill="coral",binwidth=0.5) +
    scale_y_continuous(breaks=c(10,15,20,25)) +
    scale_x_continuous(limits=c(0,8)) +
    geom_vline(xintercept=c(1.75,4.75),  linetype="dashed", size=1.5,colour="darkgreen") +
    labs(title = "Histogram Iris Data Petal Lengths") +
    theme(legend.position="none")+
    coord_cartesian(ylim = c(0,27),xlim=c(0.9,7.1)) +
    ylab("Counts") +
    xlab("Peal Length")
```

B.2 Densities

R code for Fig. 3.8:

```
z <- 1:100/100

G <- function(x,m,s) {
return(1/(sqrt(2*pi)*s)*exp(-((x - m)^2/(2*s^2))))
}


M <- function(x,m,s) {

res <- rep(0,length(x))
for (i in 1:length(m)) {
    res <- res + G(x,m[i],s[i])
}
res<- res/length(m)
return(res)
}


means <- c(0.3,0.32,0.35,0.65,0.75)
sds <- c(0.1,0.1,0.1,0.1,0.1)
plot(M(z,means,sds),type="l",col="blue",lwd=3,
+ main="Kernel Density Estimator",xlab="value",ylab="density")
for (i in 1:length(means)) {
points(G(z,means[i],sds[i])/length(means),col="red",type="l",lwd=2)
}
```

R code for Fig. 3.9:

```
l1 <- length(xS)
l2 <- length(xV)
l3 <- length(xG)

df <- data.frame(Length = factor( c(rep("setosa",l1),
+ rep("versicolor",l2),rep("virginica",l3))),proportion = c(xS,xV,xG))
```

```
cdf <- summaryBy(data=df, proportion ~ Length, FUN=mean, na.rm=TRUE)

ggplot(df, aes(x=proportion, fill=Length)) + geom_density(alpha=.2,adjust=0.7) +
    geom_vline(data=cdf, aes(xintercept=c(5.05,5.65,6.35),  colour=Length),
    linetype="dashed", size=1.2) +
    labs(title = "Iris data: density of sepal length per species") +
    scale_y_continuous(breaks=c(0,0.25,0.5,0.75,1.0,1.25,1.5)) +
    scale_x_continuous(limits=c(4,8.5)) +
    theme(legend.position=c(.8, .8))+
    theme(legend.background = element_rect(fill="gray90", size=0.3, +
    linetype="solid",colour="black")) +
    coord_cartesian(ylim = c(0,1.5),xlim=c(4,8.5)) +
    ylab("Density") +
    xlab("Sepal Length")
```

R code for Fig. 3.10 part 1:

```
l1 <- length(x1S)
l2 <- length(x1V)
l3 <- length(x1G)

df <- data.frame(Length = factor( c(rep("setosa",l1),
+ rep("versicolor",l2),rep("virginica",l3))),proportion = c(x1S,x1V,x1G))

cdf <- summaryBy(data=df, proportion ~ Length, FUN=mean, na.rm=TRUE)

ggplot(df, aes(x=proportion, fill=Length)) + geom_density(alpha=.2,adjust=0.7) +
    geom_vline(data=cdf, aes(xintercept=c(1.5,4.5,5.6),  colour=Length),
               linetype="dashed", size=1.2) +
   labs(title = "Iris data: density of petal length per species") +
   scale_y_continuous(breaks=c(0,0.5,1.0,1.5,2.0,2.5,3.0)) +
    scale_x_continuous(limits=c(0,7.5)) +
    theme(legend.position=c(.4, .7))+
    theme(legend.background = element_rect(fill="gray90",
+   size=0.3,linetype="solid",colour="black")) +
    coord_cartesian(ylim = c(0,3),xlim=c(0,7.5)) +
    ylab("Density") +
    xlab("Petal Length")
```

R code for Fig. 3.10 part 2:

```
l1 <- length(x1S)
l2 <- length(x1V)
l3 <- length(x1G)

df <- data.frame(Length = factor( c(rep("setosa",l1),
+ rep("versicolor",l2),rep("virginica",l3))), proportion = c(x1S,x1V,x1G))
```

```
cdf <- summaryBy(data=df, proportion ~ Length, FUN=mean, na.rm=TRUE)

ggplot(df, aes(x=proportion, fill=Length)) + geom_density(alpha=.2,adjust=0.9) +
    geom_vline(data=cdf, aes(xintercept=c(1.5,4.5,5.6),  colour=Length),
               linetype="dashed", size=1.2) +
   labs(title = "Iris data: density of petal length per species (zoomed)") +
   scale_y_continuous(breaks=c(0,0.25,0.5,0.75,1.0)) +
    scale_x_continuous(limits=c(0,7.5)) +
    theme(legend.position=c(.4, .7))+
    theme(legend.background = element_rect(fill="gray90",
+   size=0.3,linetype="solid",colour="black")) +
    coord_cartesian(ylim = c(0,0.93),xlim=c(0,7.5)) +
    ylab("Density") +
    xlab("Petal Length")
```

Bibliography

E. Anderson. The irises of the Gaspe Peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.

H. Attias. Independent Factor Analysis. *Neural Computation*, 11(4):803–851, 1999.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. B*, 57(1):289–300, 1995.

Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188, 2001.

C. E. Bonferroni. Il calcolo delle assicurazioni su gruppi di teste. Studii in Onore del Profesor S. O. Carboni, 1936. Roma.

D.-A. Clevert, A. Mitterecker, A. Mayr, G. Klambauer, M. Tuefferd, A. DeBondt, W. Talloen, H. Göhlmann, and S. Hochreiter. cn.FARMS: a latent variable model to detect copy number variations in microarray data with a low false discovery rate. *Nucleic Acids Res.*, 39(12):e79, 2011.

A. J. Dobson. *An Introduction to Generalized Linear Models*. Chapman & Hall, Boca Raton, London, New York, Washington D.C., 1 edition, 1990. ISBN 0412311100 / 9780412311109.

A. J. Dobson. *An Introduction to Generalized Linear Models*. Texts in Statistical Science. Chapman & Hall / CRC, London, 2 edition, 2002. ISBN 1-58488-165-8.

B. S. Everitt. *An introduction to latent variable models*. Chapman and Hall, London, 1984.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 (Part II):179–188, 1936.

E. W. Forgy. Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics*, 21:768–769, 1965.

J. A. Hartigan. Direct clustering of a data matrix. *J. Am. Stat. Assoc.*, 67(337):123–129, 1972.

J. A. Hartigan. *Clustering Algorithms*. Wiley, New York, 1975.

J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84: 502–516, 1989.

G. Hinton and T. J. Sejnowski. Introduction. In G. Hinton and T. J. Sejnowski, editors, *Unsupervised Learning: Foundations of Neural Computation*, pages VII–XVI. MIT Press, Cambridge, MA, London, England, 1999.

S. Hochreiter. Recurrent neural net learning and vanishing gradient. In C. Freksa, editor, *Proceedings in Artificial Intelligence — Fuzzy-Neuro-Systeme 97*, pages 130–137. INFIX, Sankt Augustin, Germany, 1997.

S. Hochreiter, D.-A. Clevert, and K. Obermayer. A new summarization method for Affymetrix probe level data. *Bioinformatics*, 22(8):943–949, 2006.

S. Hochreiter, M. Heusel, and K. Obermayer. Fast model-based protein homology detection without alignment. *Bioinformatics*, 23(14):1728–1736, 2007.

S. Hochreiter and K. Obermayer. Classification of pairwise proximity data with support vectors. In Y. LeCun and Y. Bengio, editors, *The Learning Workshop*. Computational & Biological Learning Society, Snowbird, Utah, 2002.

S. Hochreiter and K. Obermayer. Feature selection and classification on matrix data: From large margins to small covering numbers. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 889–896. MIT Press, Cambridge, MA, 2003.

S. Hochreiter and K. Obermayer. Gene selection for microarray data. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology*, pages 319–355. MIT Press, 2004.

S. Hochreiter and K. Obermayer. Nonlinear feature selection with the potential support vector machine. In I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors, *Feature extraction, Foundations and Applications*. Springer, 2005.

S. Hochreiter and K. Obermayer. Support vector machines for dyadic data. *Neural Computation*, 18(6):1472–1510, 2006.

S. Hochreiter and J. Schmidhuber. Flat minimum search finds simple nets. Technical Report FKI-200-94, Fakultät für Informatik, Technische Universität München, 1994.

S. Hochreiter and J. Schmidhuber. Simplifying nets by discovering flat minima. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 529–536. MIT Press, Cambridge MA, 1995.

S. Hochreiter and J. Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997a.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997b.

S. Hochreiter and J. Schmidhuber. Low-complexity coding and decoding. In K. M. Wong, I. King, and D. Yeung, editors, *Theoretical Aspects of Neural Computation (TANC 97), Hong Kong*, pages 297–306. Springer, 1997c.

S. Hochreiter and J. Schmidhuber. LSTM can solve hard long time lag problems. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 473–479. MIT Press, Cambridge MA, 1997d.

S. Hochreiter and J. Schmidhuber. Lococode versus PCA and ICA. In L. Niklasson, M. Boden, and T. Ziemke, editors, *Proceedings of the International Conference on Artificial Neural Networks, Skövde, Sweden*, pages 669–674. Springer, 1998.

S. Hochreiter and J. Schmidhuber. Feature extraction through LOCOCODE. *Neural Computation*, 11(3):679–714, 1999a.

S. Hochreiter and J. Schmidhuber. LOCOCODE performs nonlinear ICA without knowing the number of sources. In J.-F. Cardoso, C. Jutten, and P. Loubaton, editors, *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation, Aussois, France*, pages 149–154, 1999b.

S. Hochreiter and J. Schmidhuber. Nonlinear ICA through low-complexity autoencoders. In *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems (ISCAS'99)*, volume 5, pages 53–56. IEEE, 1999c.

S. Hochreiter and J. Schmidhuber. Source separation as a by-product of regularization. In M. S. Kearns, S. A. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 459–465. MIT Press, Cambridge, MA, 1999d.

Y. Hoshida, J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Subclass mapping: Identifying common subtypes in independent disease data sets. *PLoS ONE*, 2(11):e1195, 2007.

K. G. Jöreskog. Some contributions to maximum likelihood factor analysis. *Psychometrika*, 32: 443–482, 1967.

A. Kasim, D. Lin, S. VanSanden, D.-A. Clevert, L. Bijnens, H. Göhlmann, D. Amaratunga, S. Hochreiter, Z. Shkedy, and W. Talloen. Informative or noninformative calls for gene expression: A latent variable approach. *Statistical Applications in Genetics and Molecular Biolog*, 9 (1):1544–6115, 2010.

G. Klambauer, K. Schwarzbauer, A. Mayr, D.-A. Clevert, A. Mitterecker, U. Bodenhofer, and S. Hochreiter. cn.MOPS: mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate. *Nucleic Acids Res.*, 40(9):e69, 2012.

G. Klambauer, T. Unterthiner, and S. Hochreiter. DEXUS: Identifying differential expression in RNA-Seq studies with unknown conditions. *Nucleic Acids Res.*, 2013.

T. Knebel, S. Hochreiter, and K. Obermayer. An SMO algorithm for the potential support vector machine. *Neural Computation*, 20:271–287, 2008.

K. Knight and W. Fu. Asymptotics for lasso-type estimators. *The Annals of Statistics*, 28(5): 1356–1378, 2000.

C. C. Mahrenholz, I. Abfalter, U. Bodenhofer, R. Volkmer, and S. Hochreiter. Complex networks govern coiled coil oligomerization — predicting and profiling by means of a machine learning approach. *Mol. Cell. Proteomics*, 10:M110.004994, 2011.

F. Mulier and V. Cherkassky. Self-organization as an iterative kernel smoothing process. *Neural Computation*, 7(6):1165–1177, 1995.

R. M. Neal and P. Dayan. Factor analysis using delta-rule wake-sleep learning. *Neural Computation*, 9(8):1781–1803, 1997.

E. Oja. A simplified neuron model as principal component analyser. *Journal of Mathematical Biology*, 15:267–273, 1982.

C. Olsen R. Peck and J. L. Devore. *Introduction to Statistics and Data Analysis*. Brooks/Cole, Belmont, USA, 3rd edition, 2009. ISBN 9780495118732.

C. R. Rao and H. Toutenburg. *Linear Models — Least Squares and Alternatives*. Springer Series in Statistics. Springer, New York, Berlin, Heidelberg, London, Tokyo, 2 edition, 1999. ISBN 0-387-98848-3.

A. C. Rencher and G. B. Schaalje. *Linear Models in Statistics*. Wiley, Hoboken, New Jersey, 2 edition, 2008.

H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps. An Introduction*. Addison-Wesley, Reading, MA, 1992.

A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, R. I. Fisher, R. D. Gascoyne, H. K. Muller-Hermelink, E. B. Smeland, J. M. Giltnane, E. M. Hurt, H. Zhao, L. Averett, L. Yang, W. H. Wilson, E. S. Jaffe, R. Simon, R. D. Klausner, J. Powell, P. L. Duffey, D. L. Longo, T. C. Greiner, D. D. Weisenburger, W. G. Sanger, B. J. Dave, J. C. Lynch, J. Vose, J. O. Armitage, E. Montserrat, A. L'opez-Guillermo, T. M. Grogan, T. P. Miller, M. LeBlanc, G. Ott, S. Kvaloy, J. Delabie, H. Holte, P. Krajci, T. Stokke, and L. M. Staudt. The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *New Engl. J. Med.*, 346:1937–1947, 2002.

K. Schwarzbauer, U. Bodenhofer, and S. Hochreiter. Genome-wide chromatin remodeling identified at GC-rich long nucleosome-free regions. *PLoS ONE*, 7(11):e47924, 2012.

A. I. Su, M. P. Cooke, K. A.Ching, Y. Hakak, J. R. Walker, T. Wiltshire, A. P. Orth, R. G. Vega, L. M. Sapinoso, A. Moqrich, A. Patapoutian, G. M. Hampton, P. G. Schultz, and J. B. Hogenesch. Large-scale analysis of the human and mouse transcriptomes. *P. Natl. Acad. Sci. USA*, 99(7):4465–4470, 2002.

W. Talloen, D.-A. Clevert, S. Hochreiter, D. Amaratunga, L. Bijnens, S. Kass, and H. Göhlmann. I/NI-calls for the exclusion of non-informative genes: a highly effective filtering tool for microarray data. *Bioinformatics*, 23(21):2897–2902, 2007.

W. Talloen, S. Hochreiter, L. Bijnens, A. Kasim, Z. Shkedy, and D. Amaratunga. Filtering data from high-throughput experiments based on measurement reliability. *Proc. Natl. Acad. Sci. USA*, 107(46):173–174, 2010.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58(1): 267–288, 1996.

L. J. van't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.