Data analysis

Lecture notes

Univ.-Prof. Dipl.-Ing. Dr.techn.
Peter Filzmoser

Institute for Stochastics and Business Mathematics

Technical University of Vienna

Vienna, March 2021

Preface

A first step in data analysis should be an exploratory approach, in which one tries to identify structures in the data in a more informal way. This access is primarily supported by appropriate graphic processing of the data. Graphic methods have a long tradition in descriptive statistics. Histograms, for example, are among the oldest methods used in applied statistics. Which graphic methods are best suited for a particular analysis also depends on the data to be analyzed. The size of the sample and the dimensions of the data play a decisive role.

The dimension of the data often causes problems. It is true that univariate methods can be applied to each individual variable and certain partial information can be obtained, but this loses connections that can only be obtained through a multivariate approach. While graphic methods for displaying one- and two-dimensional data are practically standard methods, graphic methods for multivariate data are methodologically much more complex. A deeper understanding of such methods is essential in order to understand and interpret the results.

Statistical decisions are based on observations or are made on the basis of special conditions (randomness, independence, normal distribution, etc.). It is precisely these prerequisites under which, on the one hand, the classical methods work and, on the other hand, the mathematically simple handling is enabled. Slight deviations usually lead to false conclusions. In practice, however, these ideal conditions are rarely met.

Another goal of data analysis is to find procedures that are resistant to such deviations. Small deviations from the model should therefore have only minor effects. There are now several methods of doing this. Some of these are covered in this lecture so that an overview of the most common methods is given.

One of the pioneers of exploratory data analysis was John W. Tukey, who knew how to easily develop efficient representations and methods for analyzing statistical data. Although the computer was only used more intensively for data analysis after these developments, many of these methods are still very popular, e.g., box plots. The appropriate processing of data is very important: "*It is important to understand what you can do before you learn to measure how well you seem to have done it.*" (Tukey, 1977)

The vast amounts of data that emerge today have had a profound impact on the world of statistics. For many problems there is no closed mathematical solution, which is why numerical algorithms for finding solutions are becoming increasingly important. In general, the use of efficient algorithms has grown in importance, and computer science has therefore achieved the same importance for modern statistics as mathematics. Today, many so-called statisticians do the same work as computer scientists: they develop program systems for analyzing large amounts of data. There is an increasing need in business for such systems, and especially for people who can use them and understand the "*art of data analysis*". In an August 6, 2009 article in the New York Times, Google's chief economist Hal Varian said, "*I keep saying that the sexy job in the next 10 years will be statisticians. And I'm not kidding.*" Today

we know that the data scientist profession is in enormous demand and that many companies have difficulties in finding suitable people.

This lecture also focuses on the practicability of the methods learned, and practicability naturally means the use of the computer. The statistical software used here is R, see http://www.R-project.org, because the vast majority of newly developed statistical methods are implemented in R today, and because R goes far beyond the limits of the statistical world has also become of central importance. Most of the graphics in the script were created with R. In the legend to the figures, the corresponding R commands are in blocked font.

Literature for the lecture

J.M. Chambers, W.S. Cleveland, B. Kleiner, and P.A. Tukey (1983). *Graphical Methods forData Analysis*, Chapman and Hall, New York.

W.S. Cleveland (1987). *The Collected Works of John W. Tukey*, Volume V, Graphics 1965-1985, Chapman and Hall, New York.

W.S. Cleveland (1993). *Visualizing Data*, Hobart Press, Summit, New Jersey.

S.H.C. du Toit, A.G.W. Steyn, and R.H. Stumpf (1986). *Graphical Exploratory Data Ana-lysis*, Springer, New York.

M. Friendly (2000). *Visualizing Categorical Data*, SAS Press, Cary, NC.

J.R. Gessler (1993). *Statistische Graphik*, Birkhäuser, Basel.

D.C. Hoaglin, F.M. Mosteller, and J.W. Tukey (1983). *Understanding Robust and Explora-tory Data Analysis*, Wiley, New York.

R. Maronna, D. Martin, and V. Yohai (2006). *Robust Statistics. Theory and Methods*, JohnWiley & Sons Canada Ltd., Toronto, ON.

C. Reimann, P. Filzmoser, R.G. Garrett, and R. Dutter (2008). *Statistical Data AnalysisExplained. Applied Environmental Statistics with R*, John Wiley & Sons, Chichester.

J.W. Tukey (1977). *Exploratory Data Analysis*, Addison-Wesley, Reading, Massachussets.

K. Varmuza and P. Filzmoser (2009). *Introduction to Multivariate Statistical Analysis inChemometrics*, CRC Press, Boca Raton, FL.

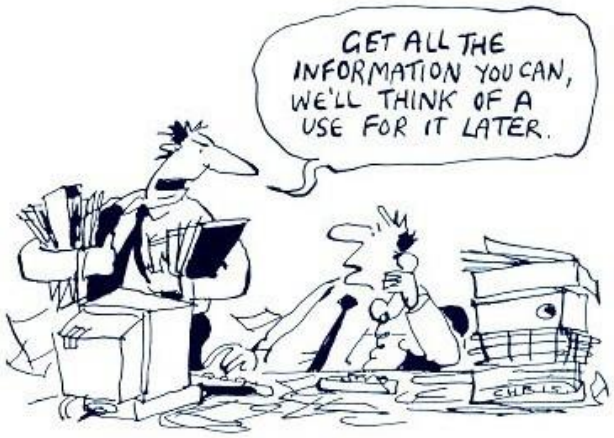
E.J. Wegman E.J. and D.J. DePriest (1986). *Statistical Image Processing and Graphics*,Marcel Dekker, New York.

Chapter 1

Sample Design: From Problem to Statistical Solution

1.1 Introduction

"Gather all the information you can get - we'll think about what to do with it later."



This motto would not be useful for meaningful statistical statements because:

- Collecting information can be costly and time consuming. Collecting a lot of information can become even more costly and time-consuming.
- Already when collecting the information, one should have the problem and the possible statistical methodology in the back of the head (result-oriented collecting).
- The same applies here: quality over quantity
- That is exactly the aim of statistics, that one can make more general statements even with little information (prognosis).
- Data must be comparable in order to be able to make a common analysis with it. In the course of time, the framework conditions could change, with which the information can no longer be compared.
- Statistics has nothing to do with detective work, where one looks for any clues, but rather one would like to come to generally valid conclusions on the basis of representative observations.

1.2 Basic terms

Population: The population is the total of all statistical units with matching identification criteria. Examples of statistical units are people, households, or events. Examples of populations are all persons who had their main residence in Austria on a reference date.

Sample: A sample is a subset of a population in a statistical investigation. The sample should be compiled in such a way as to examine certain characteristics of the total population. If one is interested in the median household income in Austria, the sample will consist of a subset of all possible Austrian households.

With the help of a sample, one would like to infer the population. This procedure is used because the population cannot usually be observed, or because it is too expensive or too time-consuming to "question" or measure all elements of the population.

Sample design: The sample design describes the selection process, how (according to which scheme) the sample is taken from the population. The overriding principle is that the sample must be representative in order to be able to draw valid conclusions about the population.

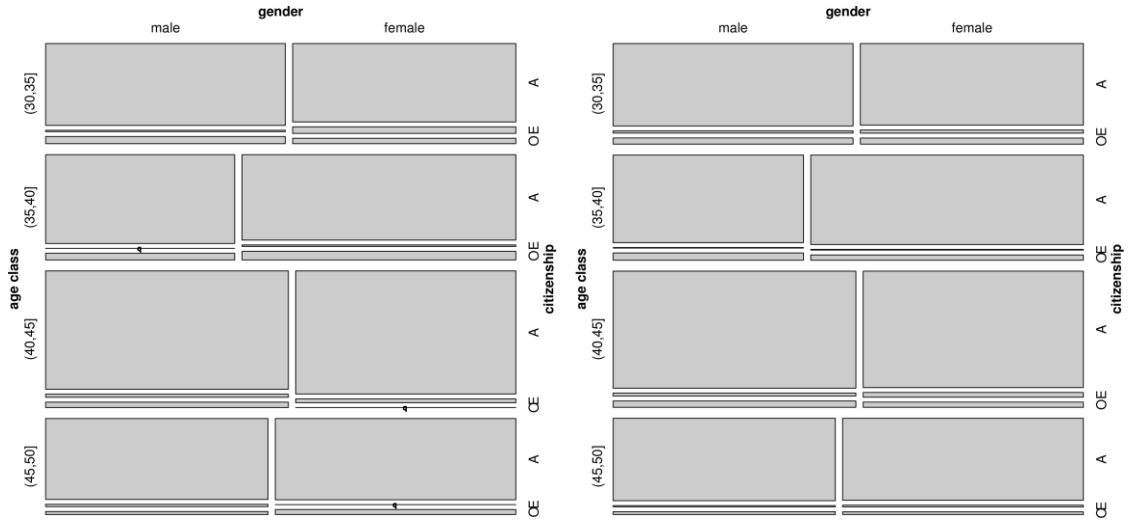
Representative sample: A representative sample should reflect the complexity and composition of the population. If one is interested in an estimate of the median household income, the sample must reflect the structure of the population based on certain characteristics (age, gender, citizenship, income component, etc.). Figure 1.2 shows this schematically with mosaic plots (the number of observations in each category is proportional to the area shown): The structure of the population (left) and that of the sample (right) agree very well.

Random or quota sample: These are two strategies (sample designs) to obtain a sample. In the case of the random sample, the observations are selected purely at random; A certain scheme is used for the quota sample in order to have certain target groups represented in the sample. The selection according to Figure 1.2 can only have been a quota sample, because with a purely random selection the various categories will not correspond so precisely to the ratio of the population. Within a quota sample, however, the observations (corresponding to the quotas) are again selected at random.

1.3 Planning of statistical data collection

First, the problem and objective must be defined. To answer the research questions, the following must be decided:

- how the *population* is defined,



- which *statistical units* should be used for measurement,
- which *variables* are collected,
- how the information is *measured* (number, index),
- about the *type* and *scope* (mostly synonymous with costs) of the survey.

Example: An online trading company wants to improve the advertising effectiveness of its products. The population consists of all computers (IP addresses) from which your products have ever been accessed. The statistical units that are used for measurement could be all accesses from IP addresses that took place in a certain time interval on selected products. The variables to be measured for these products could be the number of sales per time interval and the number of accesses to the products per time interval. If personal data on the IP addresses are known, further variables can be gender, age, residential district, etc. As values for the variables, one receives directly numerical data or categories, numbers and key figures for gender, age and residential district, which can be processed directly. The type of survey is internet-based (and not surveys), and the scope is determined by the time interval chosen and the products selected.

1.4 Statistical data collection

There are three types:

Primary statistical survey: We collect the data ourselves. This can be done through surveys,

> through our own observation and measurement, or through our own recording. All points mentioned earlier must be observed here, from correct planning to the selection of a representative sample. In the case of surveys in particular, important things must be observed, such as ensuring anonymity (otherwise distortion possible) or avoiding interference by the interviewer.

Secondary statistical survey: We use existing databases, e.g. databases from Statistics Austria

or EuroStat, databases from banks, insurance companies, companies, etc. Disadvantages of such databases are that they do not have to be in the exact context of the research question, that they can be out of date, or that they have poor data quality.

Tertiary statistical survey: We use existing aggregated data. For example, company sales are

usually not available as individual data, but only in an aggregated (summarized) form, whereby the aggregation can be done e.g. according to spatial aspects or according to industries.

1.5 Statistical data processing

After the data are available, there is usually still a long way to go before the data analysis, because the data must first be prepared for a meaningful analysis. Possible problems include:

Coding: There is only coded data, but no numerical values that can be compared. A conversion into

numerical values is not always possible or useful; you would then have to work with "*factor variables*".

Data cleansing: The data can contain nonsensical values and must therefore be checked for

plausibility and corrected if necessary. Data can contain outliers that can either be corrected or the influence of which is reduced with suitable robust statistical methods. Missing values are problematic for many statistical calculations. If it is not possible to complete these values (also using statistical methods), appropriate methods must be used that can deal with missing. Transformations of data (variables) might be necessary before statistical analysis.

1.6 Statistical data analysis

It makes sense to do this with suitable statistical software. If you want a very extensive software that is available free of charge, you will choose R (*http://cran.r-project.org*). Other commercial products are e.g. SAS, Statistica, SPSS, Stata, Eviews, Minitab.

The statistical analysis is not done just with the installation of suitable software. Even the ability to "*press the appropriate buttons*" does not necessarily mean that the analysis is meaningful and expedient. A deeper understanding of the methods to be used is essential!

1.7 Interpretation of the results

Interpretation also requires a deeper understanding of the statistical method that led to this result. In particular, the validity of the results must be coordinated with the requirements and limitations of the statistical methods. Rigorous interpretations should always be made by experts who are also familiar with the subject matter of the problem. The interpretation should of course be made with regard to the original question. Today, statistics often fall into disrepute because results are specifically steered in a direction that best corresponds to the desired interpretation. This approach is unscientific and devoid of any objectivity. A (self-) critical attitude cannot harm the interpretation.

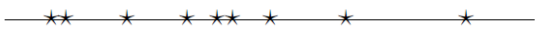
Chapter 2

Representation of one-dimensional data

We assume here that n data values $x_1, x_2, \ldots, x_n$ are given.

2.1 One-dimensional scatter plots

In the case of one-dimensional scatter diagrams, the data is simply plotted on a straight line with a selected symbol according to its value.



Of course, the straight line does not have to be in a horizontal direction, but can in principle run in any direction. In order to be able to recognize the size of the data values, a scale should also be attached.

Difficulties in the representation arise when the same observation values (multiple points) occur or when observations are very close to one another, but also when extreme outliers are present.

2.1.1 Multiple points

They can be indicated by the following variations of the scattergram:

Variations in the symbol type, symbol size, symbol color. These variations are not recommended as they sometimes give the wrong impression of the data set.

Horizontal or vertical offsetting of the symbols. Data values that are close together can cause problems, since symbols then usually overlap.

Marking with vertical lines instead of marker symbols. The line length is chosen proportionally to the number of identical data values. If data points are too close together, some lines can merge into an irregular area.

Random choice of vertical position (jittering): instead of on a straight line, position $x_1, \ldots, x_n$, points (x_i, y_i) are drawn, where $(y_1, \ldots, y_n)$ are realizations of a uniformly distributed random variable. The plot should be shown as a narrow rectangle (y-direction short) because the essential information lies in the x-direction.

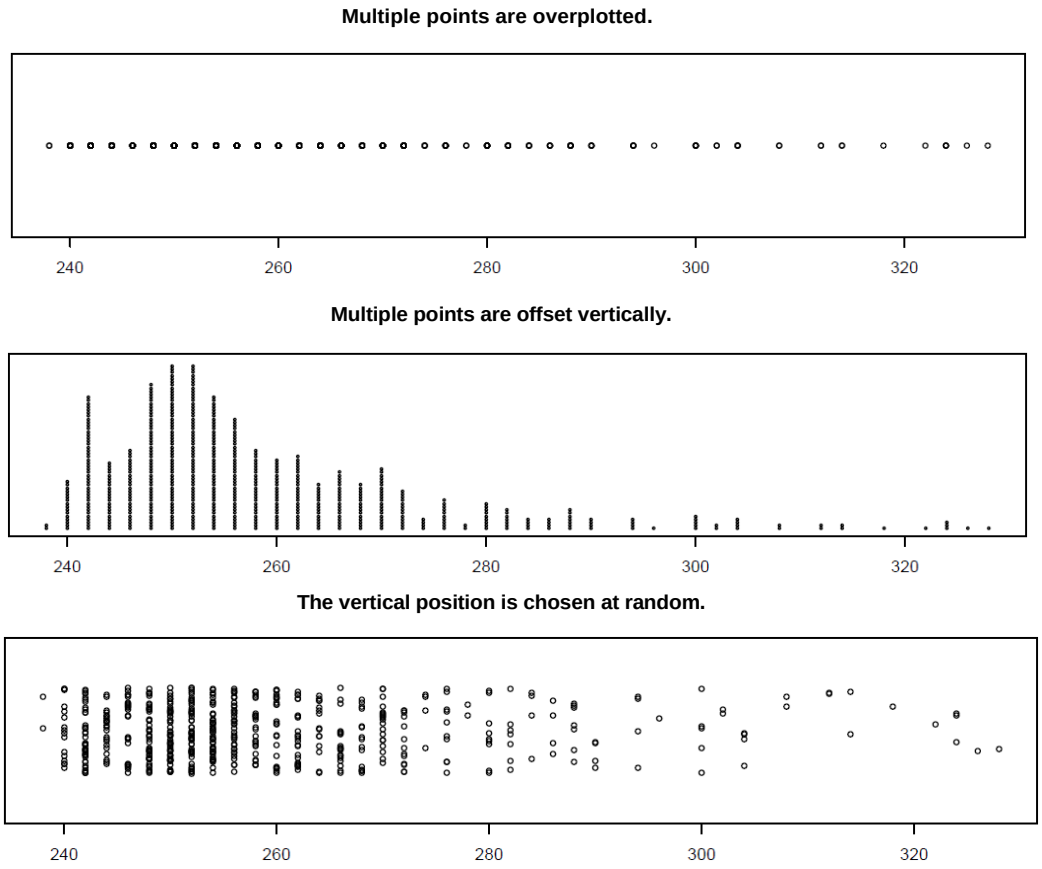


Figure 2.1 Average monthly ozone concentration in December 2000 on a grid of 24 × 24 measuring points in Central America. (strip chart)

The graphics in Figure 2.1 can be generated as follows:

```
data(ozone,package="plyr")      # load the entire data
oz72 <- ozone[,,72]             # select only December 2000
stripchart(oz72, method="overplot")
stripchart(oz72, method="stack")
stripchart(oz72, method="jitter")
```

2.1.2 Outliers

Extreme outliers cause the rest of the observations to be very close together on a scatter plot.

Omitting outliers reveals more details about the body of the data. In this case, however, the presence of the omitted outliers should always be pointed out.

2.2 Histogram

$x_1, \ldots, x_n$	...	Sample
n	...	Sample size
k	...	Number of histogram bars
$t_1, \ldots, t_k$	...	Interval limits
n_i	...	Number of data in the interval $[t_i, t_{i+1})$

The histogram function is defined as:

$$H(x) := \sum_{i=1}^{k-1} \frac{1}{t_{i+1} - t_i} \frac{n_i}{n} I_{[t_i, t_{i+1})}(x)$$

with the indicator function:

$$I_{[t_i, t_{i+1})}(x) := \begin{cases} 1 & \text{for } x \in [t_i, t_{i+1}) \\ 0 & \text{otherwise} \end{cases}$$

The histogram function is defined here with relative frequencies, but can also be displayed with absolute frequencies. The division by the interval width makes sense above all with non-equidistant interval boundaries, because then the individual bars of the histogram are displayed correctly in flat dimensions (see Figure 2.2).

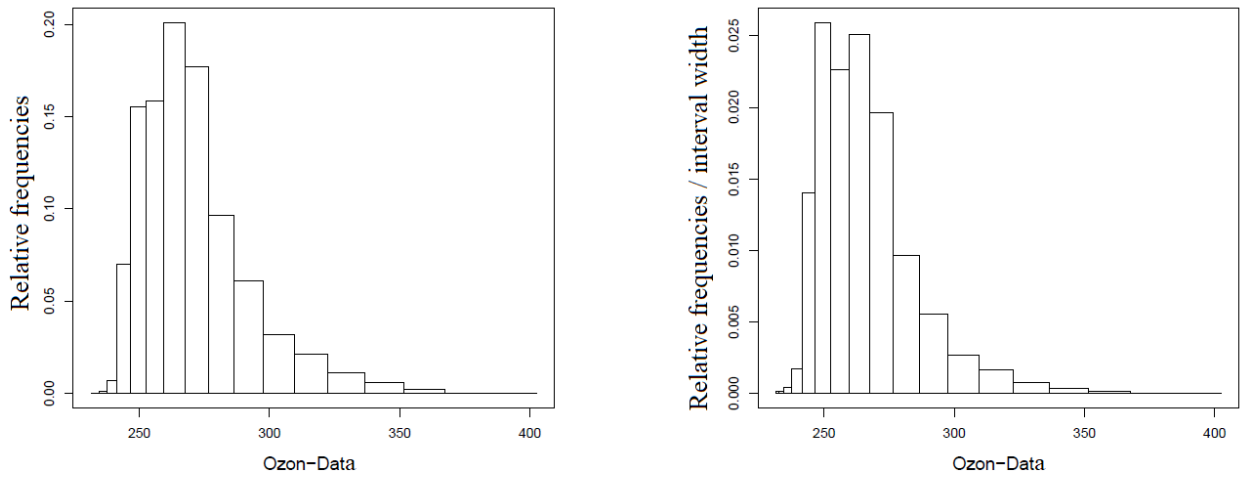


Figure 2.2 - Representation of the histogram function based on the ozone data. LEFT: Only the relative frequencies are plotted. RIGHT: The relative frequencies are divided by the interval width. (hist)

2.2.1 Choice of interval length

The appearance of a histogram depends on the length of the interval, but also on the start (end) point at which the interval division begins (ends). In the following we assume that the interval boundaries t_i are equidistant. This results in an interval length of $h_n = t_{i+1} - t_i$, for all $i = 1, \ldots, k - 1$.

Interval length according to Sturges: The optimal number k of histogram bars is chosen as:

$$k = \lceil \log_2(n) + 1 \rceil$$

where $\lceil \cdot \rceil$ means rounding up to the nearest whole number. Thus, the optimal interval length would be $h_n = (t_k - t_1)/k$. This choice is intended for data that come from a normally distributed population. For a more detailed discussion, see https://robjhyndman.com/papers/sturges.pdf.

Interval length according to Scott: Under the conditions

f Density function of the data
f continuous
f', f'' continuous and limited

is the optimal interval length

$$h_n = t_{i+1} - t_i = \left(\frac{6}{\int_{-\infty}^{\infty} f'(x)^2 dx}\right)^{\frac{1}{3}} n^{-\frac{1}{3}}$$

It is optimal in terms of that MSE (Mean Squared Error)

$$MSE(x) = E(H(x) - f(x))^2 \quad \text{for solid x}$$

The IMSE (Integrated Mean Squared Error) is a measure of the deviations over the entire range:

$$IMSE = \int MSE(x)dx$$

Since f is unknown, the above formula cannot be evaluated. Under normal distribution, the following applies:

$$h_n = \frac{3.5s}{\sqrt[3]{n}} \quad \text{s empirical standard deviation}$$

Interval length according to Freedman and Diaconis: Under similar conditions as with Scott, the optimal interval length is

$$h_n = \frac{2 \cdot IQR}{\sqrt[3]{n}}$$

where IQR = $q_{0.75} - q_{0.25}$, i.e. the difference between the quantiles 0.75 and 0.25. IQR is a robust way of estimating the standard deviation. This rule should therefore be less affected by outliers.

Figure 2.3 shows a comparison of histograms calculated according to Scott and Freedman-Diaconis. Scott's rule is sensitive to outliers.

Figure 2.4 shows a numerical and graphical comparison of the interval lengths between Scott and Freedman-Diaconis under the assumption that the data are normally distributed. A comparison of the number of classes (under normal distribution) can be found in Table 2.1. E (R_n) is the expected value for the range R_n, defined as the difference between maximum and minimum.

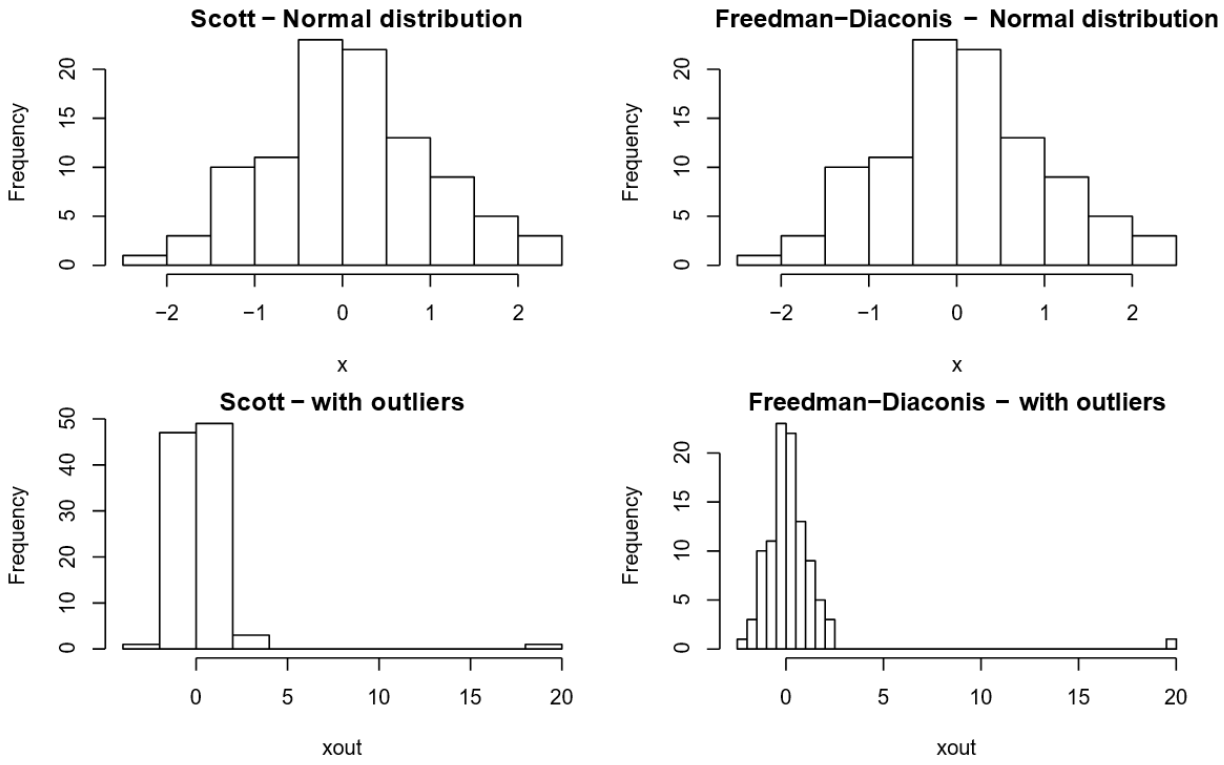


Figure 2.3 Comparison of histograms according to Scott and Freedman-Diaconis with normally distributed data with and without outliers

n	Scott	Freedman-Diaconis
10	1.620	1.252
20	1.286	0.994
30	1.123	0.868
40	1.020	0.789
50	0.947	0.743
75	0.828	0.640
100	0.752	0.582
150	0.657	0.508
200	0.597	0.461
300	0.521	0.403

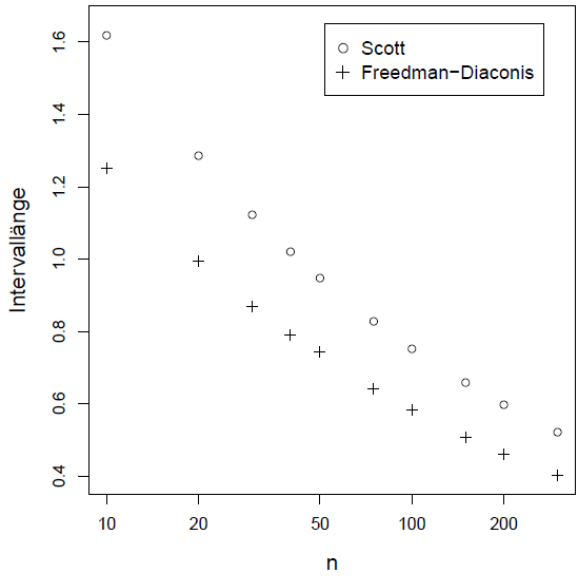


Figure 2.4 Comparison of interval lengths (in units of the estimated scatter s) of optimal histograms under the assumption that the data are normally distributed

n	$\mathrm{E}(R_n)$	$\frac{\mathrm{E}(R_n)}{h_n}$ Scott	$\frac{\mathrm{E}(R_n)}{h_n}$ Freedman-Diaconis
10	3.078	1.90	2.46
20	3.735	2.91	3.76
30	4.086	3.64	4.71
40	4.322	4.23	5.48
50	4.498	4.95	6.14
75	4.806	5.81	7.51
100	5.015	6.67	8.63
150	5.298	8.87	10.43
200	5.492	9.20	11.90
300	5.756	11.04	14.28

Table 2.1 Comparison of the number of classes of optimal histograms under the assumption that the data are normally distributed (EW_n measured in units of the estimated spread s)

2.3 Density estimation

We now try to estimate the density function of the underlying random variable directly. Using a local approach, a density estimate creates a smoother look. The density of the data at point x is calculated from the data presented in a window $[x - \frac{h}{2}, x + \frac{h}{2}]$ (local density!).

h ... Interval length (= window width)

$W(t)$ ... Weight function $\int_{-\infty}^{\infty} W(t) = 1$

$$\hat{f}(x) := \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}W(\frac{x-x_i}{h})$$

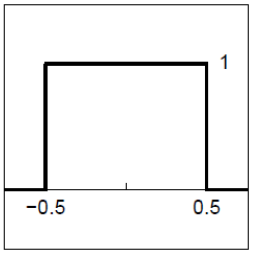
It follows with the substitution $t = \frac{x-x_i}{h}$ $(dt = \frac{1}{h}dx)$:

$$\int_{-\infty}^{\infty}\hat{f}(x)dx := \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty}\frac{1}{h}W(\frac{x-x_i}{h})dx = \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty}\frac{1}{h}hW(t)dt = 1$$

a) Rectangular weight function (boxcar function):

$$W(t) = \begin{cases} 1 & |\,t\,| \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

d.h. $W(\frac{x-x_i}{h}) = 1$ for $|x - x_i| \leq \frac{h}{2}$



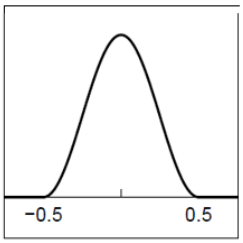
The result is a step function. Usually, however, $\hat{f}(x)$ is determined on equidistant x-values and these values are linearly connected. Then, however, no longer applies $\int \hat{f}(x)dx = 1$.

b) Cosine weight function (cosine function):

$$W(t) = \begin{cases} 1 + \cos 2\pi t & |\,t\,| < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

$\int_{-\frac{1}{2}}^{\frac{1}{2}}(1 + \cos 2\pi t)dt = 1 + \frac{1}{2\pi}\sin 2\pi t\,|_{-\frac{1}{2}}^{\frac{1}{2}} =$

$1 + 0 - 0 = 1$



With this weight function the result is $\hat{f}(x)$ as a continuous function.

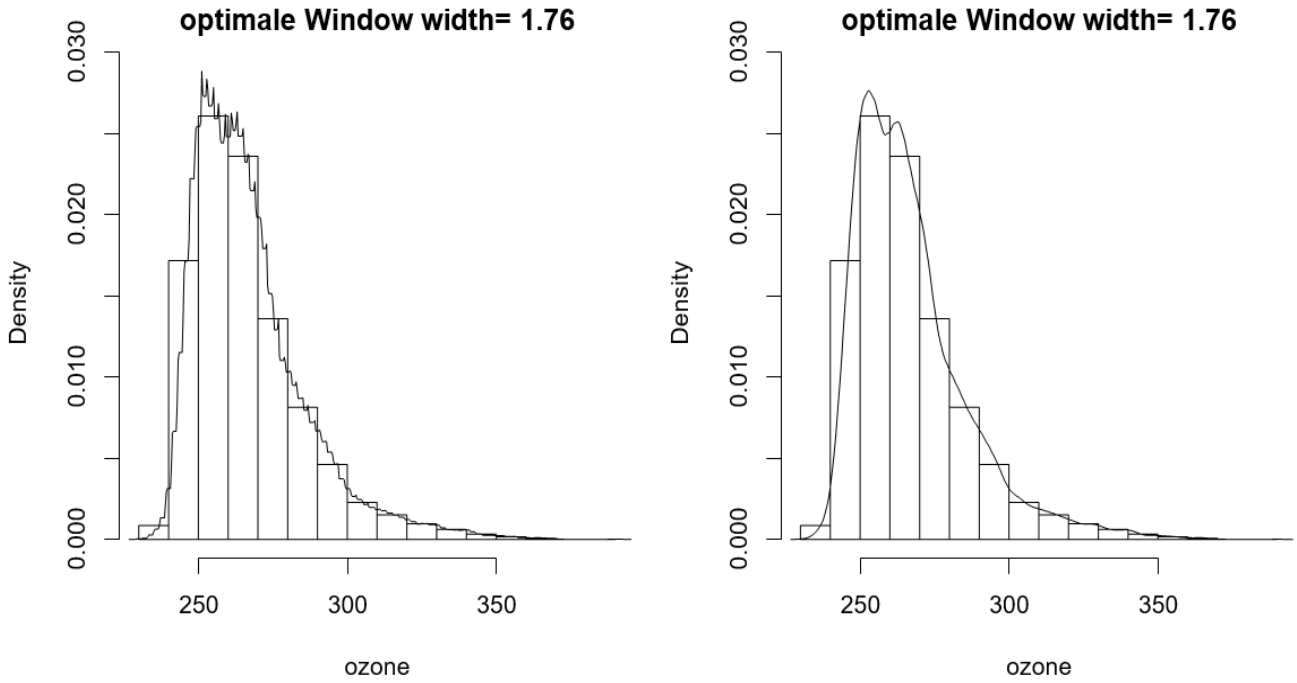


Figure 2.5 Density estimation for the ozone data from Fig. 2.1 with the square weight function (left) and the cosine weight function (right). (density)

2.4 Selection of a discrete probability model

It makes sense to make assumptions about the distribution of the data for the following reasons:

- Compact description of the data as a sample of a theoretical distribution.
- Assumptions about the distribution lead to "better" statistical methods.

Discrete distributions that can be easily identified graphically:

- Binomial distribution:
  $p_x = P(X = x) = \binom{n}{x}\theta^x(1-\theta)^{n-x} \qquad x = 0, 1, \ldots, n \qquad 0 < \theta < 1$
- Negative binomial distribution:
  $p_x = \binom{x+m-1}{m-1}\theta^m(1-\theta)^x \qquad x = 0, 1, \ldots \qquad 0 < \theta < 1, \quad m > 1$
- Poisson distribution:
  $p_x = \frac{\lambda^x e^{-\lambda}}{x!} \qquad x = 0, 1, \ldots \qquad \lambda > 0$
- Logarithmic distribution
  $p_x = -\frac{\theta^x}{x\ln(1-\theta)} \qquad x = 1, 2, \ldots \qquad 0 < \theta < 1$

The following requirements apply to the next two procedures:

$x_1, \ldots, x_n$	Sample
n_x	Number of x_j, for which $x_j = x$
$\hat{p}_x$	$\frac{n_x}{n}$

2.4.1 Procedure of Ord

The following is suitable for a graphical representation for a large number of samples:

$$\hat{U}_x := \frac{x\hat{p}_x}{\hat{p}_{x-1}}$$

Draw all points $(x, \hat{U}_x)$ for which $\hat{p}_{x-1} > 5\%$. If $\hat{U}_x \approx a + bx$ (i.e. linear), then, depending on the course of the straight line, a decision as shown in Figure 2.6 must be made.

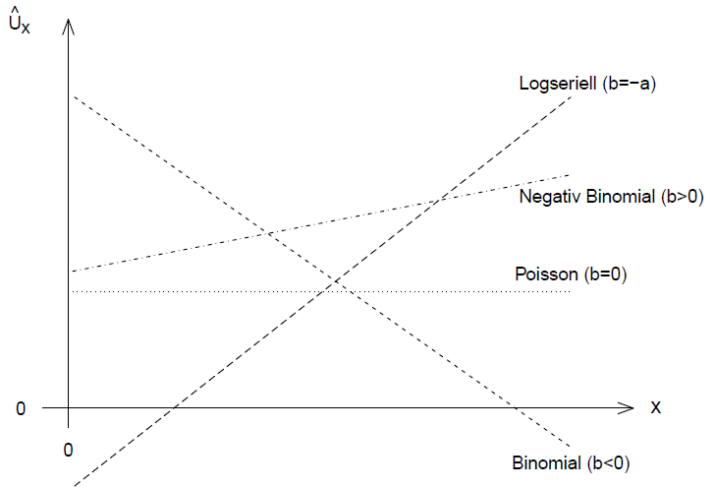


Figure 2.6 Selection of a discrete probability model according to the method of Ord

2.4.2 Procedure by Hoaglin 1960

For small samples. Decision between Poisson distribution and logarithmic distribution. Draw all points $(x, \hat{Y}_x)$ and $(x, \hat{V}_x)$ with:

$$\hat{Y}_x := \ln(x!\,\hat{p}_x)$$

and

$$\hat{V}_x := \ln(x\hat{p}_x)$$

$(x, \hat{Y}_x)$ linear -> Poisson distribution

$(x, \hat{V}_x)$ linear -> Logarithmic distribution

2.5 Empirical distribution function and probability network

2.5.1 Empirical distribution function

There are samples $x_1, \ldots, x_n$ given. We no longer want to characterize the *density function* f as before, but the *distribution function* F. For continuous quantities the (theoretical) distribution function is given by:

$$F(x) = \int_{-\infty}^{x} f(t)\mathrm{d}t$$

for all x from the domain of definition. The empirical distribution function F_n represents an estimate of the theoretical distribution function F, and it is defined by:

$$F_n(x) := \frac{1}{n}\sum_{i=1}^{n} I_{[x_i,\infty)}(x)$$

When comparing empirical and theoretical distribution functions, difficulties can arise if the theoretical distribution function has a domain of definition that contains $-\propto$ or $\propto$ (e.g. normal distribution). The following definition is therefore better:

$$F_n(x) := \begin{cases} \frac{i-0.5}{n} & x = x_i \\ \text{Fixed at the edges and between the } x_i \text{ consistent with the definition of a distribution function, e.g. linearly interpolated,. . .} \end{cases}$$

Note: Usually only the values $F_n(x_i)$ are drawn.

Figure 2.7 shows the empirical distribution functions of randomly generated standard normally distributed values. On the left 30 values were generated, on the right 1000. The dotted line is the theoretical distribution function of the standard normal distribution, and you can see that with few values the differences to the theoretical distribution function can be large, while with a higher number of samples the empirical distribution is hardly is distinguishable any more from the theoretical distribution function. One can also show that for $n \to \propto$ **the empirical to the theoretical distribution function converges.**

Figure 2.8 shows data from geochemistry. On the Kola peninsula in the border area of Norway, Finland and Russia, around 600 samples were taken at various depths of the soil. The earth material was then examined in the laboratory for the concentration of various chemical elements. The graph

on the left shows the values of Scandium (Sc) in the C horizon using an empirical distribution function. Obviously, the values in the laboratory were rounded for higher concentrations, which can be seen from the larger jumps. The right graph shows the values of nickel (Ni) in the O horizon on a log scale. High values of Ni in the O horizon indicate severe soil pollution. The distribution does not appear to be symmetrical.

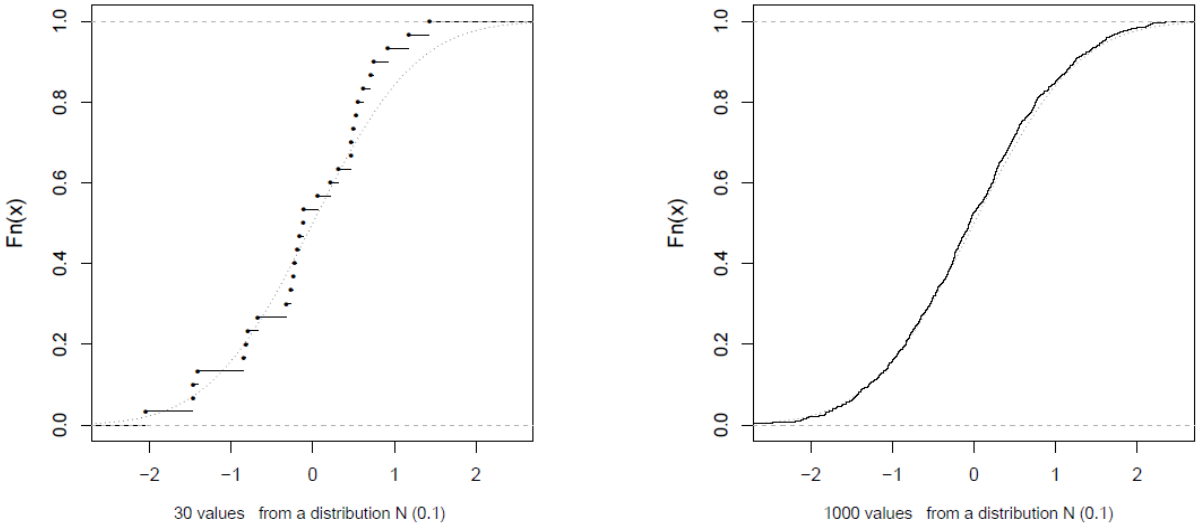


Figure 2.7 Comparison of the empirical distribution function with the theoretical distribution function (dotted). (ecdf)

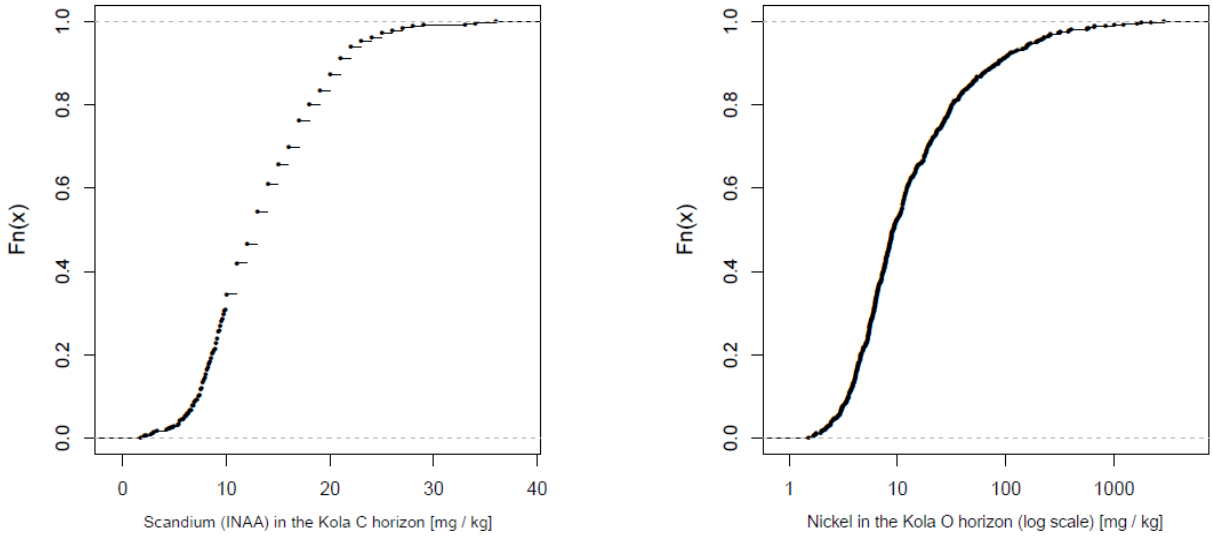


Figure 2.8: Empirical distribution functions. LEFT: Sc in the C horizon, RIGHT: Ni (log) in the O horizon

2.5.2 Probability Nets

When plotting the empirical distribution function, it is often difficult to understand whether the data come from a special distribution (e.g. normal distribution). However, one could distort the vertical axis for this purpose by changing the scaling. In the probability grid, the vertical axis between 0 and 1 is not divided into equal intervals, but the distances are plotted proportionally to ϕ^{-1}, where ϕ^{-1} is the inverse of the distribution function of the standard normal distribution. When using this scaling, $F_n(x)$ is not graphically represented over x, but $\phi^{-1}(F_n(x))$ over x. If the data are now approximately normally distributed, then $F_n(x) \sim F(x)$ (i.e. $N(\mu, \sigma^2)$) will be

$$\Phi^{-1}(F_n(x)) \sim \Phi^{-1}(F(x)) = \Phi^{-1}\left(\Phi\left(\frac{x-\mu}{\sigma}\right)\right) = \frac{x-\mu}{\sigma},$$

so that the points come to lie roughly on a straight line. One then reads an estimate for μ at a probability of 50% and an estimate for $\mu + \sigma$ at 84%. In the case of normal distribution, the deviations from the straight line should be "purely random" and not systematic, but they depend heavily on the number of observations. The more observations there are, the fewer deviations should occur.

For the Kola data from Figure 2.8, the probability nets are shown in Figure 2.9. On the left one sees Sc in the C horizon. The deviations from the straight line seem to be considerable. The rounded values are again noteworthy, which are easier to see here than in Figure 2.8, because each individual value is entered. Due to the scaling of the horizontal axis, it is also possible to draw conclusions about the original data values.

In the right graph of Figure 2.9 Ni is shown in the O horizon (log scale). Thus, one does not check for normal distribution but for logarithmic normal distribution. Here, too, strong deviations from the straight line can be seen. You can also see a "kink" in the function (around the value 10) and can therefore conclude that this is a combination of two distributions. Since nickel is typical for impurities in the OH horizon, one part could describe the uncontaminated areas and the second part the polluted areas.

Note: Since the vertical axis actually corresponds to the quantiles of the standard normal distribution, it was also possible to scale with the quantiles instead of scaling with the probabilities. The structure of the plot would not change, only the scaling of the vertical axis. This is introduced as the Q-Q plot in the following section.

2.6 Quantile-Quantile Plots

With quantile-quantile plots, two distributions can be compared directly with one another. Most of the time, one of these distributions is a hypothetical distribution and the other is the distribution of existing data. For example, the normal distribution can be assumed as the hypothetical

distribution, and the data distribution is thus compared with the normal distribution. The basis of comparison are the quantiles of the distributions.

Two distribution functions F_x and F_y are shown in Figure 2.10. The quantiles $q_x(p)$ and $q_y(p)$ can now be determined for a concrete probability p.

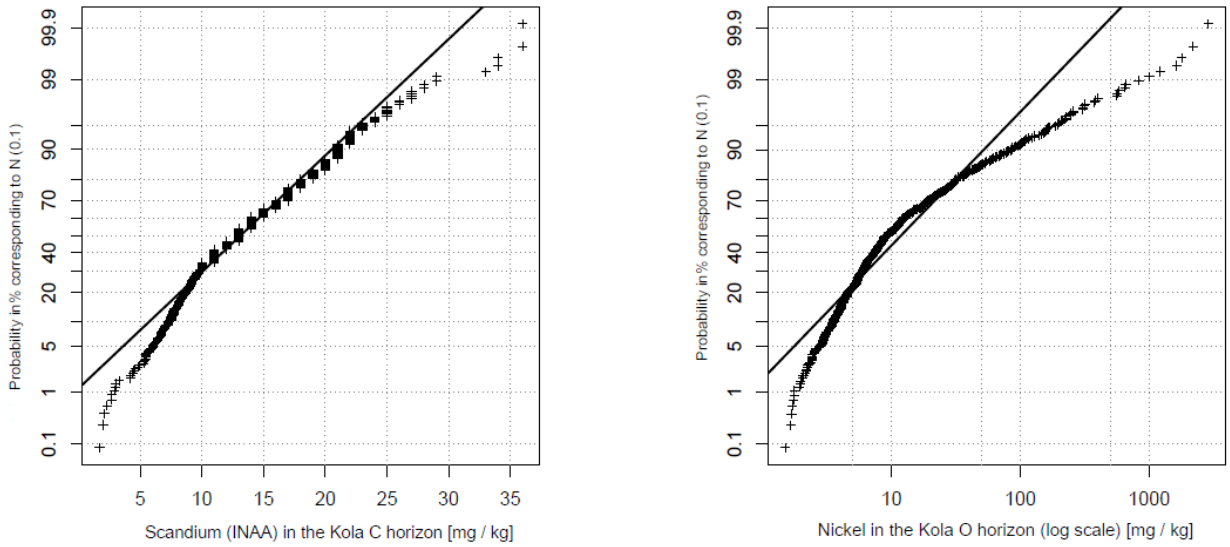


Figure 2.9: Probability nets for the Kola data. LEFT: Sc in the C horizon, RIGHT: Ni (log) in the O horizon

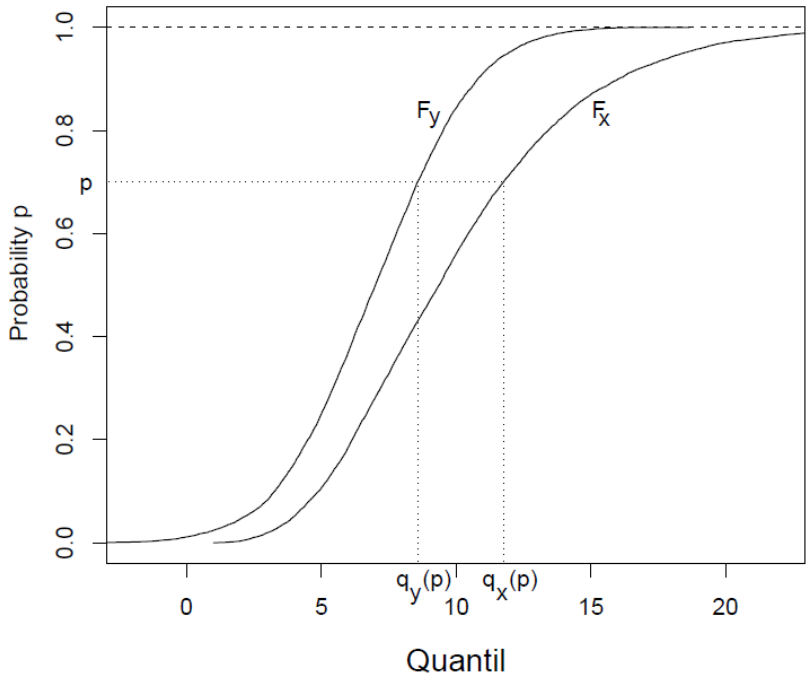


Figure 2.10: Representation of two distribution functions.

Let F_x or F_y be empirical or theoretical distribution functions. Then the quantiles are determined by the inverse functions:

$$F_x(t) = P(X \leq t), \quad q_x(p) = F_x^{-1}(p)$$
$$F_y(t) = P(Y \leq t), \quad q_y(p) = F_y^{-1}(p)$$

Quantile-Quantile Plot (Q-Q Plot): Draw the points $(q_x(p_i), q_y(p_i))$ for selected p_i, e.g. $p_i \coloneqq \frac{i-\alpha}{n-2\alpha+1}$ for $0 \leq \alpha < 1$.

If the two distributions are the same, the quantiles must also be the same and the points must lie on a straight line, so:

$$F_x = F_y \Leftrightarrow \left(q_x(p_i), q_y(p_i)\right) \textit{ lie on the straight line } y = x.$$

In general, however, you will not want to check for exact equality of the distributions, but rather - similar to the probability network - whether the two distributions come from the same distribution family. E.g. one would like to check whether the present data are normally distributed with parameters μ and σ. However, since the parameters are unknown, the comparison can be carried out with a standardized form of the same distribution family, in this case with the standard normal distribution. The question now is whether and how this comparison can be carried out in the Q-Q plot.

Let us assume that the random variable Y results from a transformation from the random variable X, i.e.

$$Y = \mu + \sigma X.$$

Then the following applies:

$$F_y(t) = P(Y \leq t) = P(\mu + \sigma X \leq t) = P\left(X \leq \frac{t-\mu}{\sigma}\right) = F_x\left(\frac{t-\mu}{\sigma}\right)$$

$$q_y(p) = F_y^{-1}(p) \quad | \quad F_y(p)$$
$$F_y(q_y(p)) = F_y(F_y^{-1}(p)) = p$$

With $t = q_y(p)$ we have:

$$F_y(q_y(p)) = F_x\left(\frac{q_y(p)-\mu}{\sigma}\right) = p$$
$$q_x(p) = F_x^{-1}(p) = F_x^{-1}\left(F_x\left(\frac{q_y(p)-\mu}{\sigma}\right)\right) = \frac{q_y(p)-\mu}{\sigma}$$
$$\Rightarrow \qquad q_y(p) = \mu + \sigma q_x(p)$$

This means that for distribution families (e.g. constant uniform distribution, exponential distribution, normal distribution, ...), in which the distributions of the family differ only in terms

of position μ and scatter σ, the points of the QQ plot lie between 2 distributions of the family in a straight line. The intercept term gives an estimate for μ, the slope an estimate for σ.

This procedure is illustrated in Figure 2.11 for annual snowfall data from Buffalo from 1910-1973. In the left graph, the parameters μ and σ of the normal distribution are determined. The right graph shows the histogram with the estimated density function and shows the theoretical normal distribution with the determined parameters. Apparently, the estimation of the parameters led to a very good result.

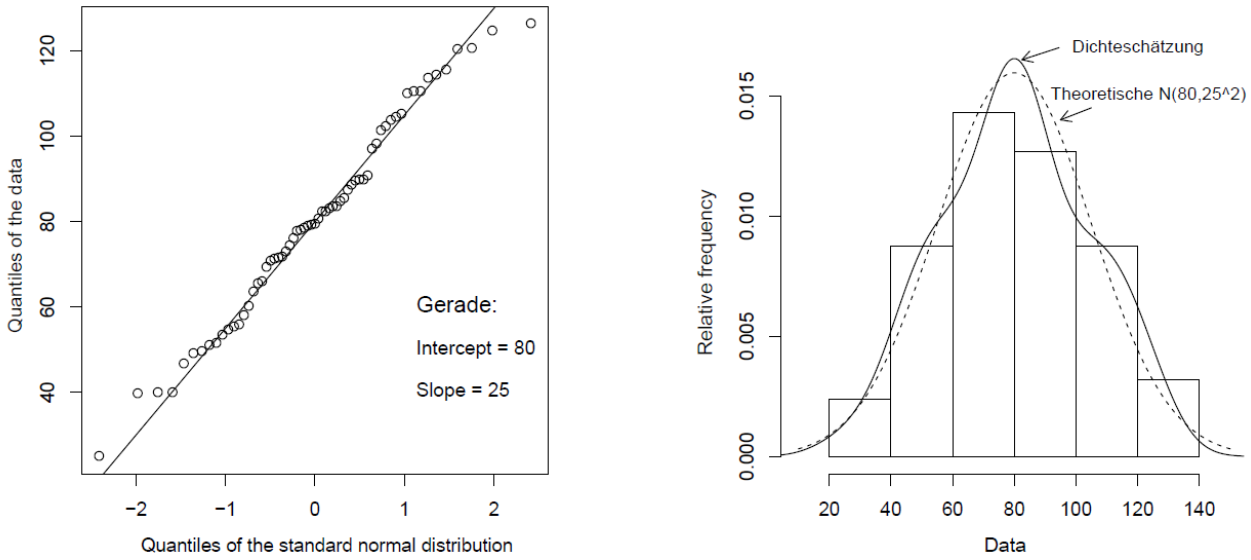
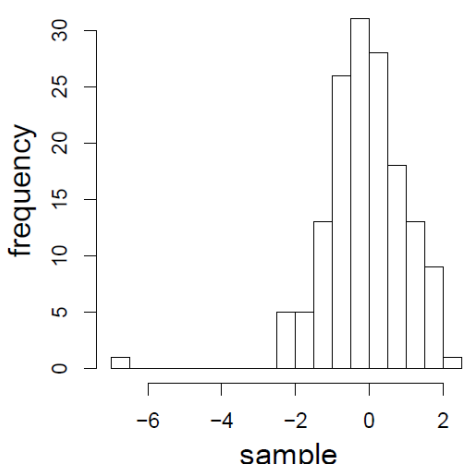


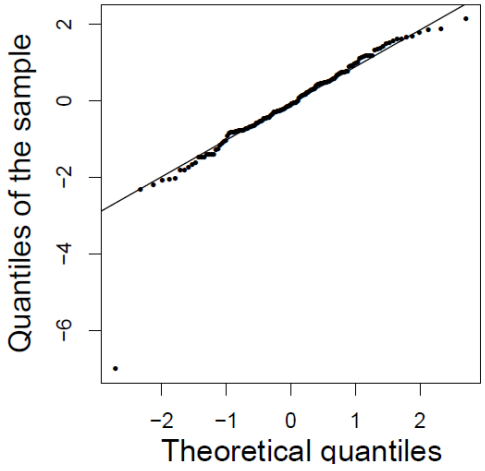
Figure 2.11: LEFT: QQ plot of the snowfall data from Buffalo (1910-1973) with the estimated parameters of the normal distribution; RIGHT: Histogram with estimated density function of the original data, as well as theoretical normal distribution with the determined parameters.

2.6.1 Deviations from the straight line

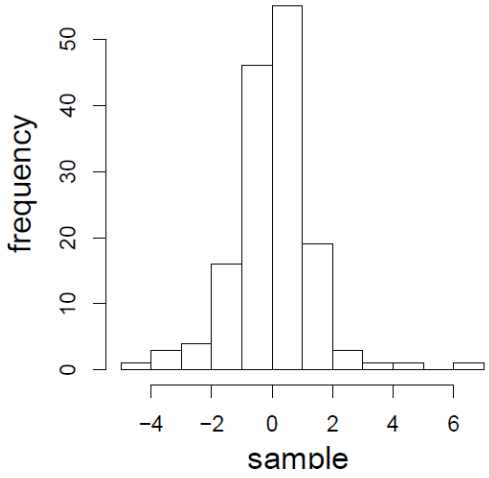
If the Q-Q plot shows deviations from the straight line, there are more or less large differences between the two distribution functions. The differences can be of a random nature, e.g. due to random samples with a very small sample size. In this case one cannot yet conclude that X and Y actually come from different distributions. However, there may be systematic differences that can be attributed to the following causes:

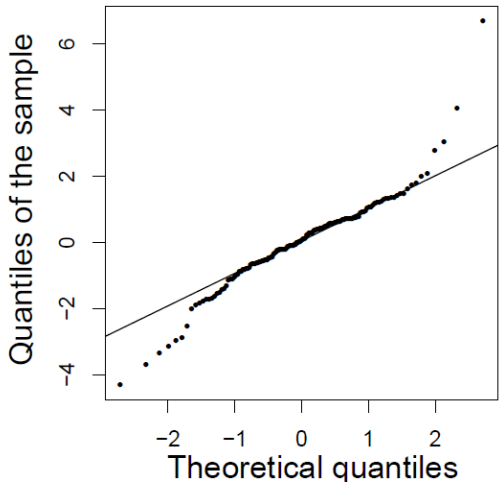
1. Outliers: lead to individual isolated points in the lower or upper area



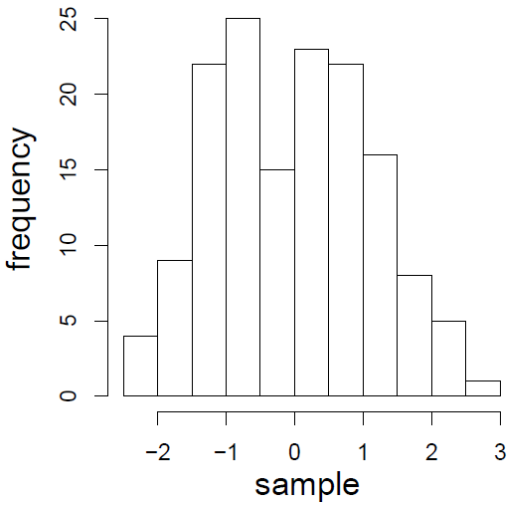


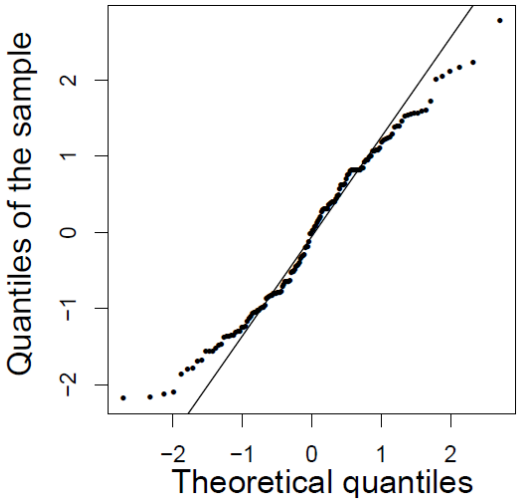
2. Curvature at the ends: Y (empirical VF) has more mass in the tails of the distribution than X (theoretical VF).



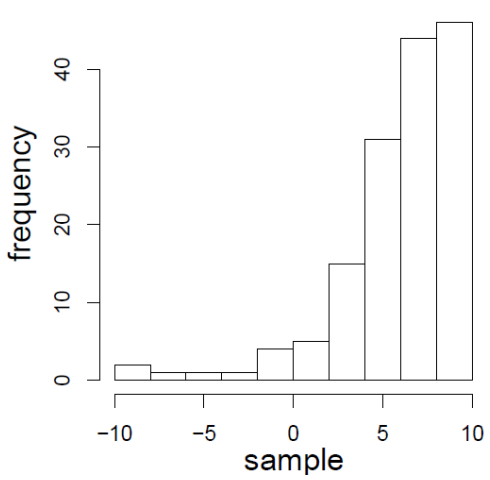


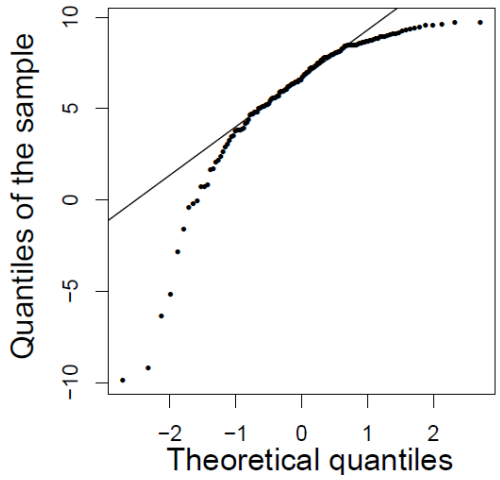
3. Curvature at the ends: Y (empirical VF) has less mass in the tails of the distribution than X (theoretical VF).



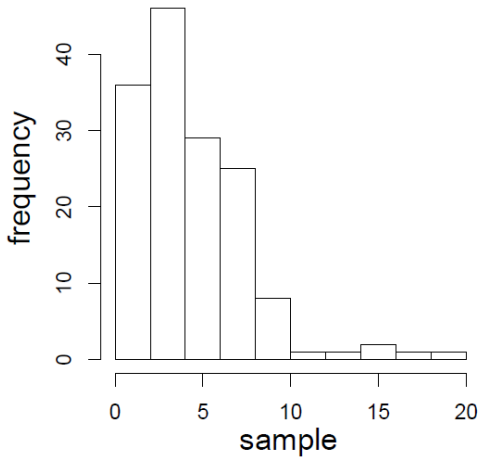


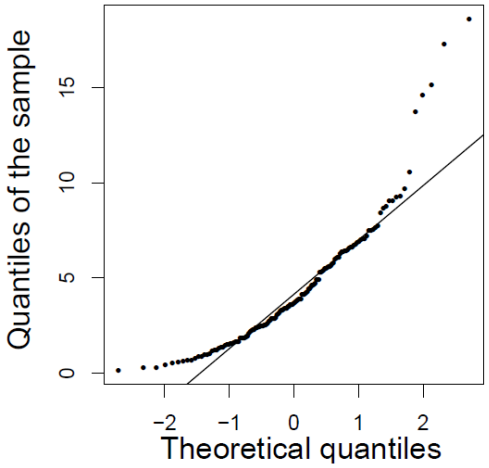
4. Convex or concave shape: If X is symmetrically distributed: Y is skewed to the left if the lower quantiles are further away from the median than the upper quantiles



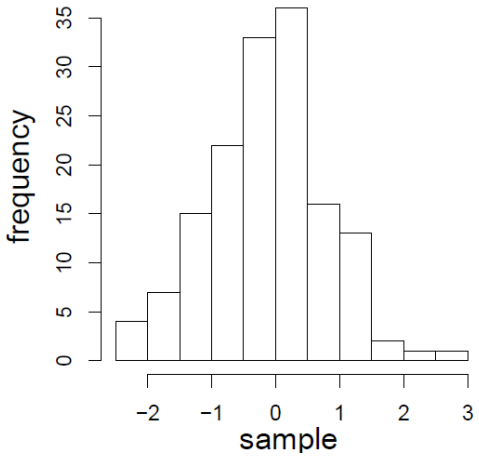


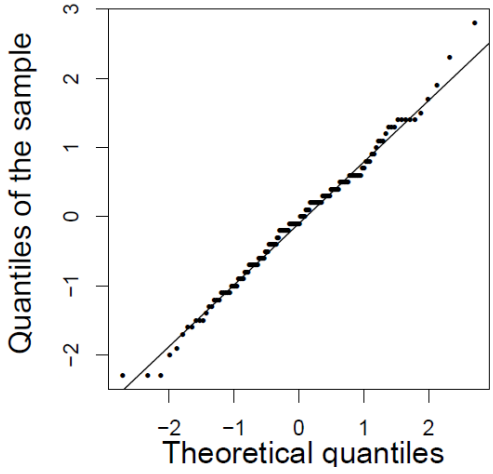
5. Convex or concave shape: If X is symmetrically distributed: Y is skewed to the right if the upper quantiles are further away from the median than the lower quantiles



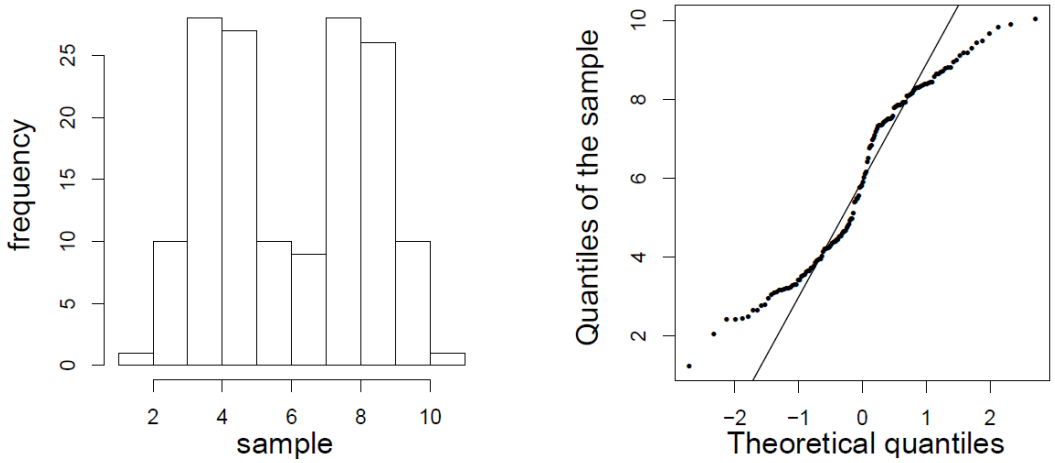


6. Horizontal segments: rounded values or discrete distribution





7. Plateaus: Composition of 2 or more distributions, or clusters in the data



Annotation:

1. Q-Q plots react sensitively to (also random) deviations near the margins in distributions whose definition range goes down to $-\propto$ or $\propto$ (e.g. normal distribution).
2. To see a nice linear shape, one needs a larger number of data values.

2.6.2 Scatter in Q-Q plots

Due to the asymptotic behavior of order statistics, the deviation of points in Q-Q plots is distributed according to $N(Q(p), \frac{p(1-p)}{nf(Q(p))^2})$, where $Q(p)$ is the quantile of the probability p, and f denotes the density of the distribution. We denote by z_i the empirical quantiles of the Q-Q plot. An estimate s_{z_i} for the standard error of the empirical quantiles of a Q-Q plot is

$$s_{z_i} := \frac{\hat{\delta}}{g(q_i)} \sqrt{\frac{p_i(1-p_i)}{n}}$$

with:

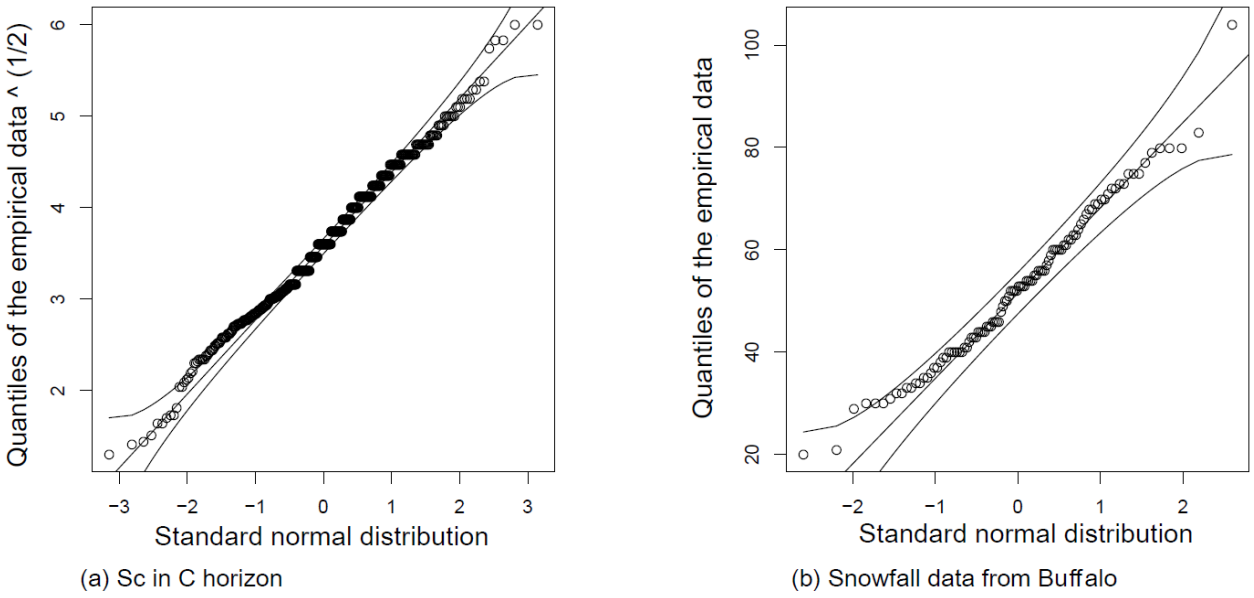
$\hat{\delta}$	Estimated value for the slope in the linear Q-Q plot, e.g. $\frac{Q_{0.75}-Q_{0.25}}{q_{0.75}-q_{0.25}}$ with the theoretical quantiles $q_{0.75}$ and $q_{0.25}$ and the empirical quantiles $Q_{0.75}$ and $Q_{0.25}$
$g(x)$	Density of the standardized distribution (e.g. standard normal distribution).
q_i	theoretical quantiles of the Q-Q plot

Figure 2.12 shows Q-Q plots with additional scatter information. The standard error was determined for each theoretical quantile q_i (horizontal) and plotted twice up and down from the point of intersection with the straight line (concentration interval). The points were then connected

with lines to make the visual impression easier. Points that are thus outside these lines would have a significant deviation from the theoretical distribution.

One notices that the standard errors become smaller as the sample size *n* increases, and thus the two lines are closer together.

From Figure 2.12 (a) it can be seen that these (transformed) data correspond relatively well to the normal distribution. However, the precipitation data in Figure 2.12 (b) show significant deviations in the lower range.



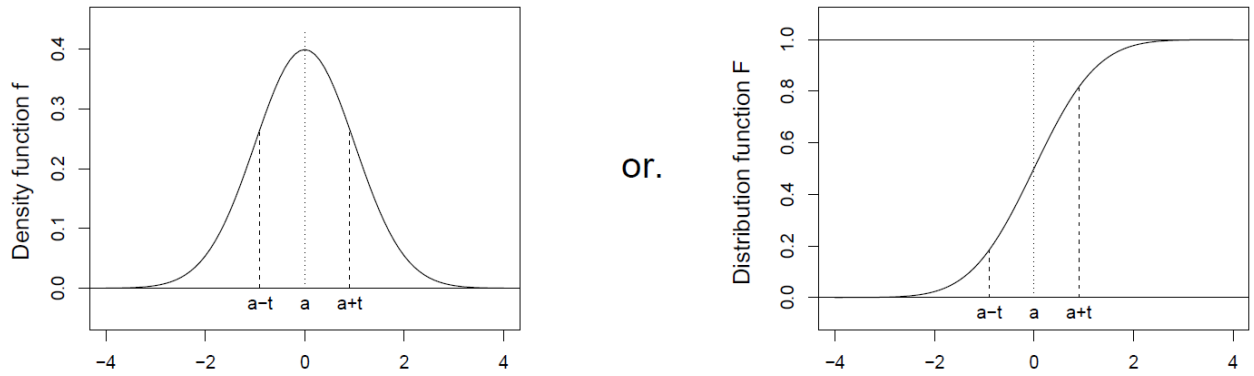
(a) Sc in C horizon (b) Snowfall data from Buffalo

*Figure 2.12: QQ Plots with additional scatter information, generated e.g. with the function **qq.plot** from the **library (car)***

2.6.3 Check for symmetry of a distribution using Q-Q plots

For one around x = a symmetrical density function f (x) applies:

$$f(a-t) = f(a+t) \qquad F(a-t) = 1 - F(a+t)$$



With $p := F(a-t)$ we have $F(a+t) = 1-p$. Let us now consider the quantiles (for $p \leq 0.5$):

$$F^{-1}(p) = F^{-1}(F(a-t)) = a-t \qquad \text{or.} \qquad F^{-1}(1-p) = F^{-1}(F(a+t)) = a+t$$

Hence the quantiles $F^{-1}(p)$ and $F^{-1}(1-p)$ (for $p \leq 0.5$) are symmetric about a. This idea can now be used to test for symmetry.

We consider the ordered sample values $x_{(1)}, \ldots, x_{(n)}$ and the median x_M of the sample. These values are understood as quantiles of the distribution and can be used in the Q-Q plot.

- Symmetry around the median:

  We form the pairs of points

  $$(x_M - x_{(1)}, x_{(n)} - x_M), (x_M - x_{(2)}, x_{(n-1)} - x_M), \ldots,$$

  In case of symmetry, the points had to lie on a 45-degree straight line.

- Symmetry of all quantile pairs:

  We form the pairs of points

  $$(x_{(n)} - x_{(1)}, x_{(n)} + x_{(1)}), (x_{(n-1)} - x_{(2)}, x_{(n-1)} + x_{(2)}), \ldots.$$

  In the case of symmetry, the points had to lie on a horizontal straight line (about 2 times the median).

Example: Figure 2.13 shows both variants of the Q-Q plots for the Buffalo snowfall data.

In the left graph you can see that the points in the center (bottom left) lie quite well around the straight line, which suggests symmetry around the median. The further you go from the center, the more asymmetrical the distribution becomes. Since the values are above the straight line, the upper quantiles have greater distances from the median than the lower quantiles, which suggests a right-skewed distribution. The minimum is apparently much further from the median than the maximum.

Similar statements are obtained from the graph on the right in Figure 2.13. The straight line drawn corresponds to twice the median of the values. It can be seen that the "inner" points still have a similar horizontal position, although they run slightly above the median. After that, an upward trend can be seen, which means that the differences become larger and there is thus a right-skewed distribution. The minimum is again recognizable as a special case.

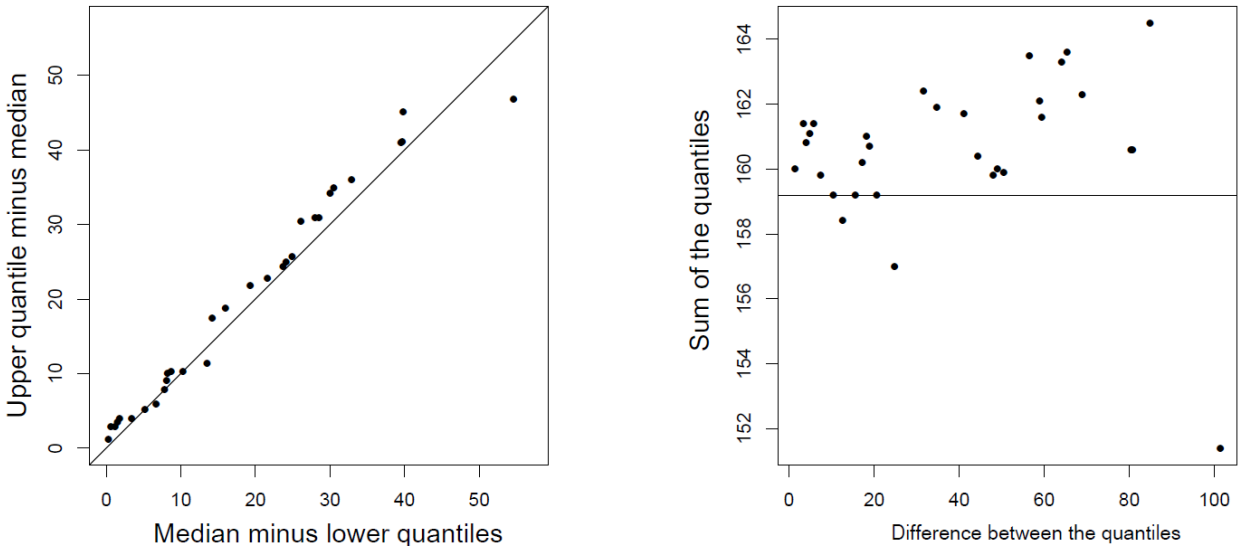


Figure 2.13: Q-Q plots of Buffalo snowfall data. LEFT: Checking for symmetry with normalization around the median; RIGHT: Checking for symmetry without normalization.

2.7 Boxplots

Purpose: To present important characteristics of distributions of one-dimensional random variables.

Comparison of several samples, of data groups, etc.

Sample: $x_1, \dots, x_n$

$x_M := median(x_1, \dots, x_n)$

$Q_{0.25}$ and $Q_{0.75}$ are the quantiles 0.25 and 0.75, respectively

$IQR = \mathrm{Q}_{0.75} - Q_{0.25}$ is the interquartile range

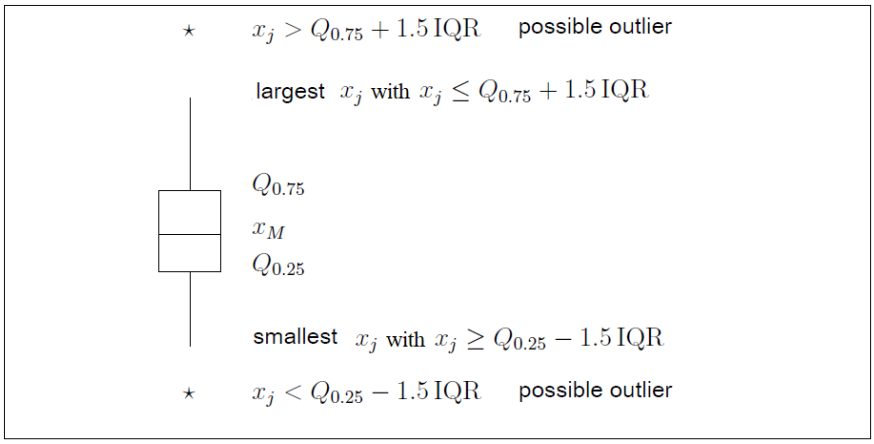


*Figure 2.14: Definition of a box plot (**boxplot**)*

Box plots are a good supplement to the graphs presented earlier, as Figure 2.15 shows for the Buffalo snowfall data (left) and for the logarithmized arsenic data of the Kola O-horizon (right).

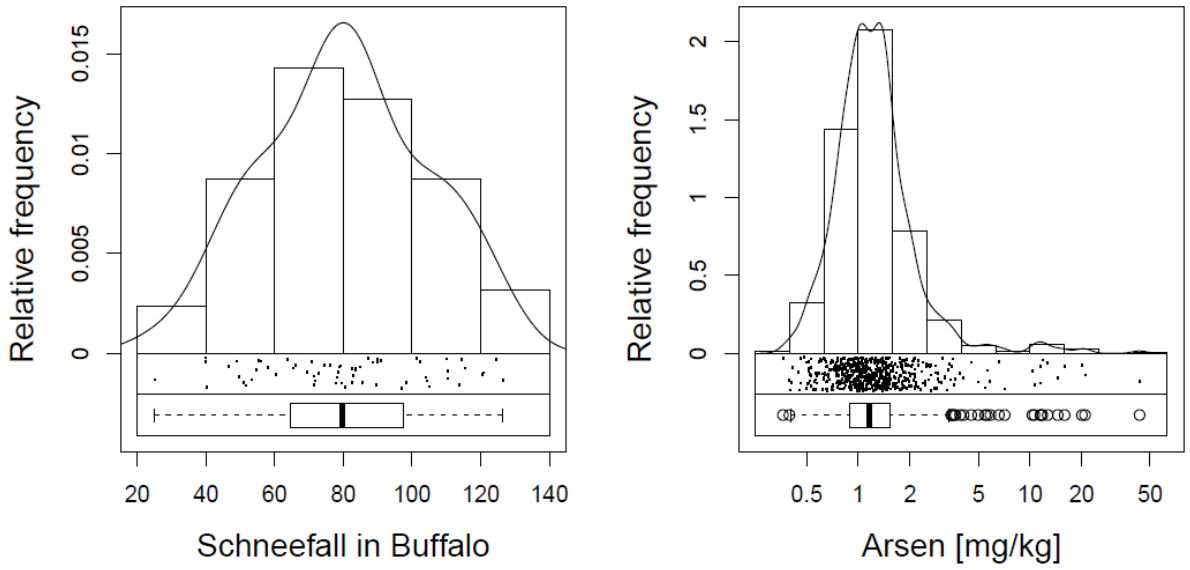


*Figure 2.15: Combination of histogram, density estimate, one-dimensional scatter plot and box plot for the snowfall data from Buffalo (left) and for arsenic from the Kola O-horizon (right). (**edaplot** or **edaplotlog** from **library(StatDA)**)*

Example 1: Box plots are particularly valuable for visually comparing two or more data series. Figure 2.16 shows a comparison of accidents on Swedish motorways in 1961. In that year speed restrictions were introduced. A box plot with data before the introduction of the restriction and a box plot with the data afterwards make it easy to see that the median remains the same, but the spread increases significantly.

Accidents with Restriction		Accidents without Restriction	
9	19	9	20
11	19	9	21
12	21	11	22
12	21	11	24
13	22	13	24
14	22	15	26
15	23	15	28
15	27	17	29
16	29	18	31
18	41	28	32
19	42	19	40

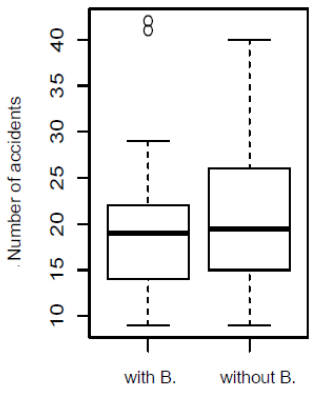


Figure 2.16: Number of accidents on Swedish motorways in 1961 on 22 consecutive days with or without speed limit.

Box plots are very informative due to their construction:

- Robustness through the use of median and quartiles.
- To judge skewness according to the position of the median to the quartiles and according to the occurrence of outliers (i.e. if they occur only on one side or in clusters on one side).
- Assessment of the curvature and the proportion of outliers (see Table 2.2).

distribution	Median	Upper Quartile	Outlier Boundaries	Percent outside	value of 1.96 σ_t	% outside $\mu \pm 1.96\ \sigma$
symmetric distributions						
U (-1.1)	0	0.500	± 2,000	0.00	1,132	0.00
N (0.1)	0	0.674	± 2,698	0.70	1,960	5.00
t_{20}	0	0.687	± 2,748	1.24	2,066	5.20
t_{10}	0	0.700	± 2,800	1.88	2.191	5.32
t_5	0	0.727	± 2.908	3.35	2,530	5.25
t_1	0	1,000	± 4.000	15.59	-	-
skewed distributions						
χ^2_1	0.45	0.102 1,323	- 1,730 3,155	7.58	- 1,772 3,772	5.22
χ^2_5	4.35	2,675 6,626	- 3,252 12,552	2.80	- 1,198 11.198	4.78
χ^2_{20}	19.34	15,452 23,828	2,888 36,392	1.39	7,604 32,396	4.53

+ 95% of the data with normal distribution

Table 2.2: Behavior of different distributions when shown in box plots

Modifications of box plots:

1. Width proportional to $\sqrt{n}$
2. Notches as confidence intervals.
   Notches: $median \pm 1.57 \frac{IQR}{\sqrt{n}} (\alpha = 0.05$ for tests).
   Whether the box plots can be used for tests also depends on whether the variances of the plots differ to a greater or lesser extent.

Example 1 continued: Figure 2.17 shows the comparison with notched box plots for the accident data. Since the notches overlap, a significant difference in the medians cannot be assumed. However, this statement should be made with caution because the variances are very different.

Example 3: The data **Carseats** can be found in the R package **ISLR**. Here, the sales figures (*Sales*) of child car seats were recorded from 400 stores, as well as the quality of the accommodation of the seats on the shelves (*ShelveLoc*) with expression Bad/Median/Good, and No/Yes Information *Urban* (city/country), and *US* for the location of the business. These categorical variables can be used to group the sales data and to compare the individual data groups with box plots. This was first done in Figure 2.18. You can use the formula notation in R here:

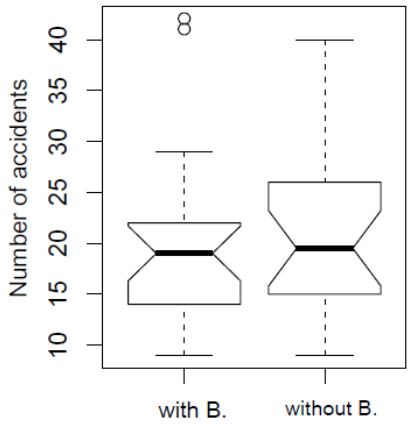


Figure 2.17: Box plots with notches of the accident data from Figure 2.16.

```
library(ISLR)
data(Carseats)
attach(Carseats)
boxplot(Sales~Urban,notch=TRUE)
boxplot(Sales~US,notch=TRUE)
boxplot(Sales~ShelveLoc,notch=TRUE)
```

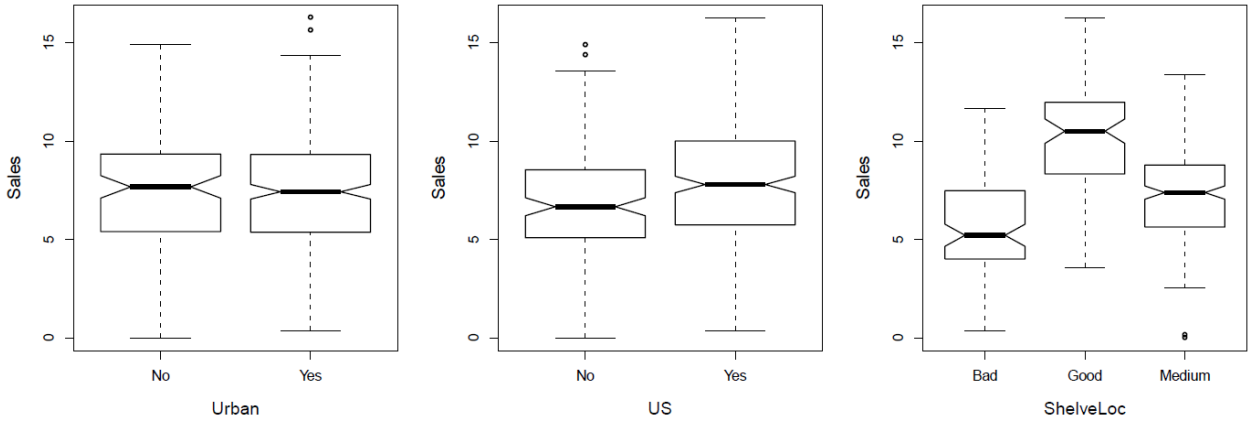


Figure 2.18: Comparison of the sales data for child car seats according to different categories of additional variables.

Figure 2.18 shows that the mean (median) sales for shops in the city and in the country are not significantly different (left), but that one can see significant differences with regard to location US and positioning on the shelves.

You can also subdivide by several categorical variables at the same time, as the following code and Figure 2.19 show:

```
boxplot(Sales~US:ShelveLoc,notch=TRUE)
boxplot(Sales~US:Urban:ShelveLoc,notch=TRUE)
```

In Figure 2.19 (above) it can be seen that the best (average) sales are achieved for stores located in the USA, and where the child seats are clearly visible in the store. The graphic below goes one step further: in addition to the previous insight, sales in urban areas are better than in businesses in the country.

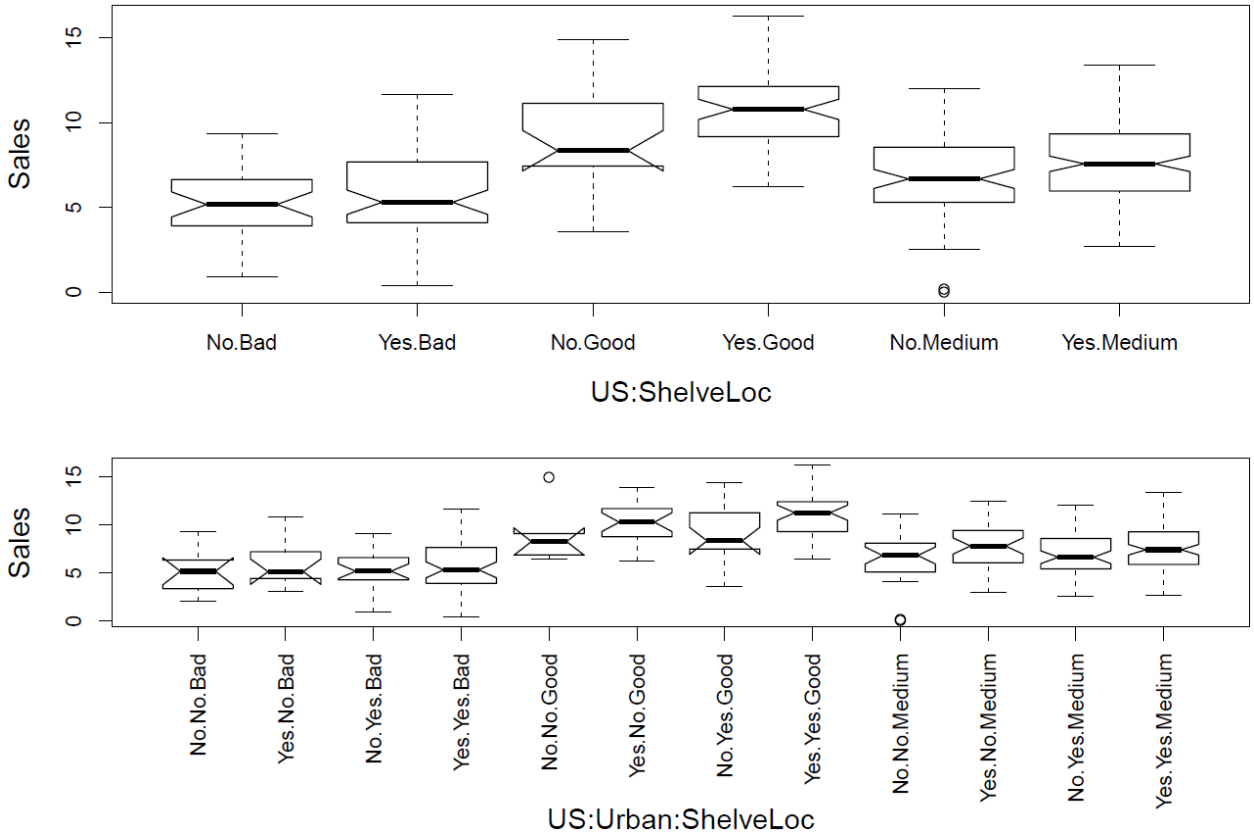


Figure 2.19: Comparison of the sales data for child car seats according to several categories of additional variables.

Chapter 3

Robust univariate estimators

In the previous chapters it became clear that the location and the spread (or variance) play a very central role. For example, for data, $x_1, \ldots, x_n$, which come from a normal distribution $N(\mu, \sigma^2)$, is very important for how location μ and scatter σ are estimated. The estimation of these parameters depends strongly on the data quality. If the quality is good (no outliers, no coarse rounding effects, etc.), the arithmetic mean is recommended.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

as an estimate for μ, and the empirical standard deviation

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

to use as an estimator for σ. However, if the data quality is poor because there are various outliers that deviate greatly from the main part of the data, these estimators will deliver very biased results. In this case it is better to use robust estimators that focus on the main (homogeneous) part of the data.

Univariate here means nothing other than one-dimensional, that is, measurements of one variable are available. In contrast to this, we will get to know multivariate estimators later, when several variables are observed at the same time.

3.1 Robust estimate of location and scatter

Let $x_1, \ldots, x_n$ be the sample and $x_{(1)}, \ldots, x_{(n)}$ be the sample sorted from smallest to largest value.

A well-known robust location estimator is the α-trimmed mean, which for $0 \leq \alpha < 0.5$ and $g = \lfloor n\alpha \rfloor$ is defined as:

$$m(\alpha) = \frac{1}{n-2g}(x_{(g+1)} + \ldots + x_{(n-g)})$$

Here $\lfloor k \rfloor$ means that the largest integer $\leq k$ is taken (round down).

Another possibility for robust location estimation is the median:

$$median(x_1, \ldots, x_n) = \begin{cases} x_{(\frac{n+1}{2})} & \text{n odd} \\ \frac{x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}}{2} & \text{n straight} \end{cases}$$

About the quartiles

$$Q_{0.25} := median\left(x_{(1)}, \ldots, x_{\left(\left[\frac{n+1}{2}\right]\right)}\right) \ldots \textit{ Lower quartile estimator } (\textit{quantile } 0.25)$$

$$Q_{0.75} := median\left(x_{\left(\left[\frac{n}{2}\right]+1\right)}, \ldots, x_{(n)}\right) \ldots \textit{ Upper quartile estimator } (\textit{quantile } 0.75)$$

the interquartile range is obtained as a robust measure of scatter:

$$\text{IQR} = Q_{0.75} - Q_{0.25} \quad \ldots \quad \textbf{I}\text{nterquartile } \textbf{R}\text{ange}$$

Analogous to the trimmed mean, one can also define a trimmed deviation: α-trimmed standard deviation, for $0 \leq \alpha < 0.5$, $g := \lfloor n\alpha \rfloor$

$$S(\alpha) = \sqrt{\frac{1}{n-2g-1} \sum_{i=g+1}^{n-g} (x_{(i)} - m(\alpha))^2}$$

Another robust measure of scatter is the MAD (medmed):

$$\text{MAD} = \underset{1 \leq i \leq n}{median} \left(\left| x_i - \underset{1 \leq j \leq n}{median} \, x_j \right| \right) \quad \ldots \quad \textbf{M}\text{edian } \textbf{A}\text{bsolute } \textbf{D}\text{eviation}$$

The MAD is very robust, but it has poor statistical efficiency. A very robust and efficient scatter estimator is the Qn:

$$Qn = \{|x_i - x_j|; i < j\}_{(k)}$$

where (k) represents the kth value of the ascending order, with $k = \binom{h}{2} \approx \binom{n}{2}/4$ and $h = \left\lfloor \frac{n}{2} \right\rfloor + 1$.

Neither IQR, nor $S(\alpha)$, MAD or Qn are suitable for estimating the parameter σ of a normal distribution, because these are *not consistent estimators* (the "estimator function" does not converge to the parameter σ). If you want to achieve consistency, you have to correct with factors. Consistent estimates for the standard deviation σ are:

1. $s_{\text{IQR}} = \frac{\text{IQR}}{1.35}$
2. $s(\alpha) = \frac{S(\alpha)}{c_\alpha}$
   The constant c_α depends on α. For $\alpha = 0.1$ is $c_{0.10} = 0.66$.
3. $s_{\text{MAD}} = \frac{\text{MAD}}{0.675} = 1.483 \cdot \text{MAD}$
4. $s_{Qn} = 2.219 \cdot Qn$

Note: The square of the stated estimation functions is used as the estimation function for the variance.

Example:

Sample :	2.1	3.7	2.6	5.8	1.6	1.1	32.7	4.7	3.1	4.8
ordered:	1.1	1.6	2.1	2.6	3.1	3.7	4.7	4.8	5.8	32.7

For $\alpha = 0.1$: $g = \lfloor 10 \cdot 0.1 \rfloor = 1 \Rightarrow m(0.1) = \frac{1}{10-2}(x_{(2)} + \ldots + x_{(9)}) = \frac{1}{8}28.4 = 3.55$

$median = \frac{(3.1+3.7)}{2} = 3.4$

$Q_{0.25} = 2.1$

$Q_{0.75} = 4.8$

$s = \sqrt{\frac{1}{10-1}\sum_{i=1}^{10}(x_i - \bar{x})^2} = 9.4$

$s_{\text{IQR}} = \frac{4.8-2.1}{1.35} = 2.0$

$s(0.1) = \frac{1}{0.66}\sqrt{\frac{1}{8-1}\sum_{i=2}^{9}(x_i - m(\alpha))^2} = 2.2$

MAD:

$x_i - \underset{1 \le j \le 10}{median}\ x_j$:	-2.3	-1.8	-1.3	-0.8	-0.3	0.3	1.3	1.4	2.4	29.3
$\lvert x_i - \underset{1 \le j \le 10}{median}\ x_j \rvert_{(i)}$:	0.3	0.3	0.8	1.3	1.3	1.4	1.8	2.3	2.4	29.3

$\text{MAD} = \frac{1.3+1.4}{2} = 1.35$

$s_{\text{MAD}} = \frac{\text{MAD}}{0.675} = 2.0$

Qn:

absolute differences $|2.1 - 3.7|, |2.1 - 2.6|, \ldots, |2.1 - 4.8|, |3.7 - 2.6|, \ldots, |3.1 - 4.8|$ sort, and take the k-largest value with $k = \binom{6}{2} = 15$

The 15th largest value is $Qn = 1.5$. Thus $s_{Qn} = 3.33$

The R function **Qn** in the **robustbase** package returns the value 2.397, because there a correction is made for small sample sizes.

3.2 One-dimensional outlier detection

Box plots are immediately suitable for identifying outliers in a series of measurements. We can, however, derive further rules for the identification of univariate outliers, namely those based on robust location and scatter estimation. Corresponding rules for the robust case can be derived based on the normal distribution, where the inner 95% of the values lie in the range mean $\pm$ 2 times the standard deviation. Figure 3.1 shows the comparison of such rules for simulated normally distributed (left) and log-normally distributed (right) data. It should be noted that the proportion of identified "outliers", which here more likely represent the extremes of the distributions, depends on the scope of the data.

What is interesting, however, is how the outlier rules work when there are actually outliers in the data. For Figure 3.2, simulated data with a correspondingly large number of observations of normal distribution (A) or log-normal distribution (B) were used, but in which a varying proportion comes from a different distribution. This proportion can be viewed as the actual outlier proportion. The outlier rules can now be used to compare how many of the simulated outliers are actually recognized, or how high the proportion of outliers may be until the rule "collapses".

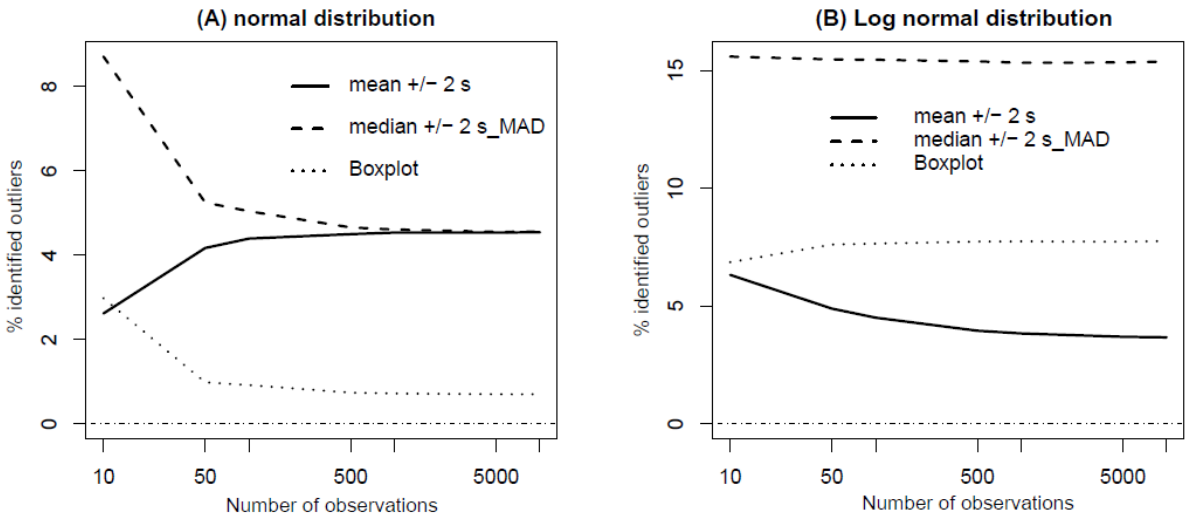


Figure 3.1: Proportion of identified "outliers" for normally distributed (A) and lognormally distributed (B) data.

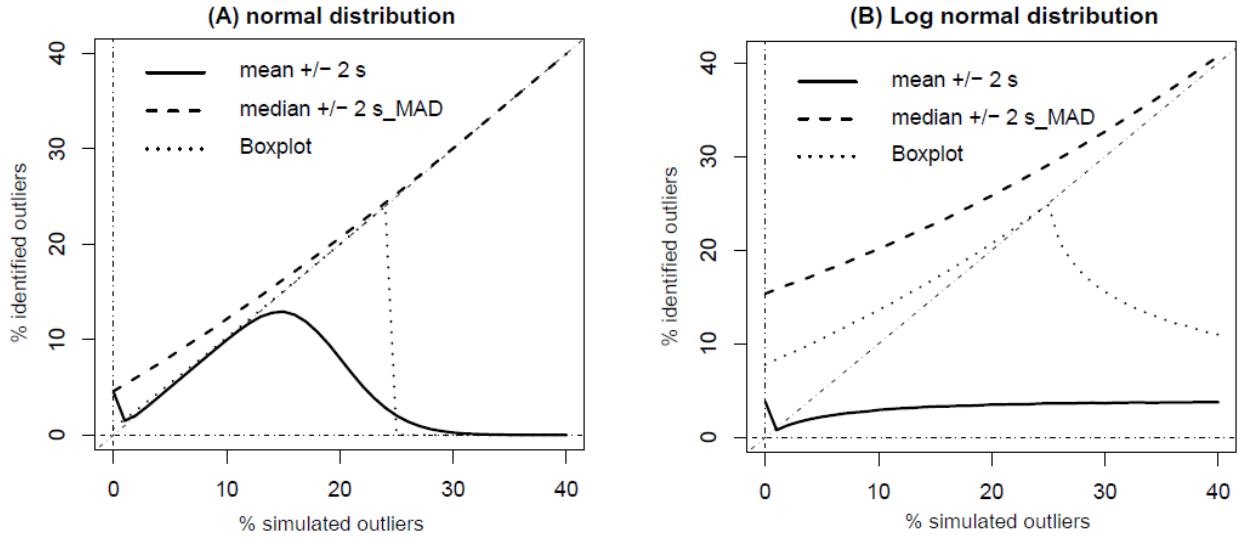


Figure 3.2: Proportion of identified "outliers" in normally distributed (A) and log-normal distributed (B) data with real simulated outliers.

Chapter 4

Representation of two-dimensional data

4.1 Scatter diagram (Scatterplots)

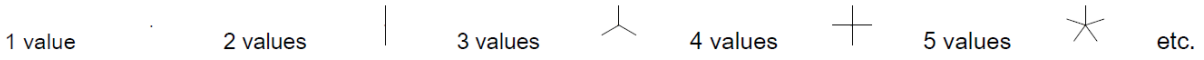
There are observations of the two variables X and Y. Both X and Y can be random variables. The simplest form of a scatter diagram is to draw the pairs of points (x_i, y_i) for i = 1, …, n. However, multiple points are not marked in this way. They can be made visible in the following ways:

- due to random blocking of the data points (jittering)

  $$\tilde{x}_i := x_i + \theta_x u_i$$
  $$\tilde{y}_i := y_i + \theta_y v_i$$

  with u_i, v_i independently continuously in (-1,1) uniformly distributed random numbers and θ_x, θ_y fixed (e.g. $\theta_x = 0.02(x_{max} - x_{min})$ and $\theta_y = 0.02(y_{max} - y_{min})$).
- by Sunflowers

  Encryption of multiple points using the following symbols:

- by division into cells and sunflowers

  The data area is divided into cells and the cell frequencies are displayed according to the Sunflowers transformation rules.

Examples: Table 4.1 gives age data for managers who are employed by *Bell Laboratories*. The data is two-dimensional as the age is in 1982 and the years elapsed since graduation.

The original data is shown in Figure 4.1 (left). However, there are multiple points that cannot be seen in the plot. A division into cells is therefore made. The right graphic shows the individual cell frequencies with the help of sunflowers.

4.2 Stripe box plots

Aside from the problem with multiple points, it is often difficult to read trends from a 2-dimensional point cloud. As a remedy, you can subdivide the data (e.g. in a horizontal direction)

and display it using box plots. This is also a sensible approach if there are an extremely large number of measurements, so that the scatter plot would only give a "large black spot".

Table 4.1: Data from managers at Bell Labs: Age in 1982 (A) and number of years (Y) elapsed since graduation.

A	J	A	J	A	J	A	J	A	J	A	J	A	J	A	J
35	12	42	16	44	18	47	21	50	28	54	28	57	32	59	31
36	10	42	19	44	19	47	21	51	22	54	29	57	37	59	32
36	12	43	15	44	19	47	21	51	27	54	29	58	23	59	33
36	14	43	17	44	20	47	21	52	19	54	30	58	27	59	34
37	10	43	17	45	13	47	23	52	25	55	25	58	28	59	35
37	12	43	17	45	15	47	25	52	25	55	27	58	31	60	27
38	10	43	17	45	20	47	26	52	26	55	29	58	32	60	28
38	14	43	20	45	20	48	18	53	22	55	29	58	33	60	33
39	10	43	21	45	21	48	21	53	23	55	30	58	33	61	34
39	14	44	9	45	21	48	23	53	24	55	31	58	33	61	40
39	15	44	12	46	18	48	26	53	27	55	31	58	34	62	43
40	12	44	14	46	18	49	20	53	30	55	33	58	34	62	35
40	14	44	16	46	19	49	22	53	31	55	33	59	25	63	30
41	10	44	16	46	20	50	17	53	32	56	27	59	28	63	41
41	17	44	17	46	21	50	21	54	21	56	28	59	29	64	40
42	8	44	17	47	18	50	23	54	27	56	28	59	30	64	41
42	12	44	18	47	21	50	24	54	28	56	30	59	30	66	43

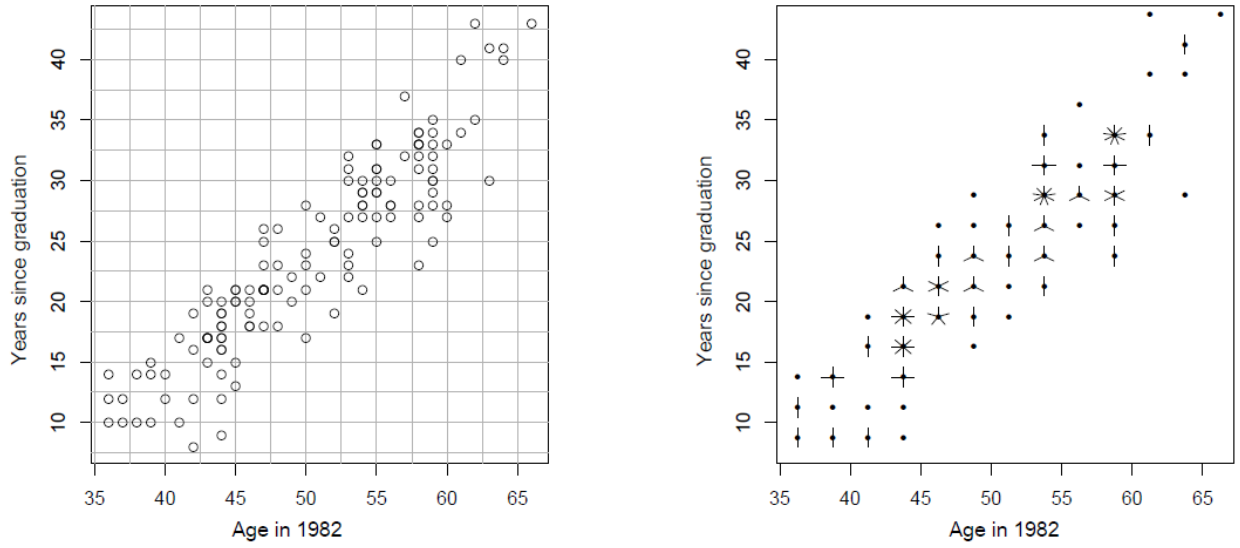


*Figure 4.1: Scatter plot of the manager data with cell division (left) and the resulting sunflowers (right). (**sunflowerplot** from package **graphics**)*

Let us consider the data from Table 4.2, in which hamsters reached age and the percentage of hibernation in their lives.

Table 4.2: Hibernation data from 144 hamsters: age of hamsters at the time of death (A) and percentage of hibernation in their life.

%	A	%	A	%	A	%	A	%	A	%	A
0	116	4	959	12	1124	15	1107	19	1008	23	1025
0	612	4	810	12	876	15	843	19	1174	23	760
0	711	4	678	12	843	15	760	19	1438	24	1587
0	744	6	397	12	826	15	711	20	1256	24	1504
0	760	6	496	12	793	15	512	20	1174	24	1289
0	579	6	727	12	545	16	1388	20	1140	24	1041
0	562	6	1008	12	364	16	826	20	1074	25	909
0	545	7	1058	12	264	16	810	20	810	25	1190
0	496	8	876	13	1289	16	777	20	711	25	1207
0	364	8	975	13	1140	16	331	21	1223	25	1256
0	314	9	975	13	678	17	760	21	1140	25	1587
1	826	9	810	13	446	17	893	21	992	26	760
1	975	9	711	14	1289	17	909	21	942	26	1091
1	893	9	678	14	1273	17	1289	21	860	27	1372
1	826	9	446	14	1157	18	1289	21	843	28	264
1	727	10	579	14	1132	18	1207	22	645	28	1107
1	678	10	694	14	1124	18	1124	22	727	28	1124
1	579	10	810	14	1107	18	1058	22	1107	29	760
1	430	11	1107	14	893	18	1041	23	1421	29	1107
2	826	11	826	14	884	18	1008	23	1306	29	1273
2	860	11	760	14	876	18	909	23	1273	29	1620
2	1074	11	744	14	860	18	860	23	1256	30	760
3	760	11	727	14	760	18	562	23	1174	32	1355
3	975	12	1207	15	1223	19	545	23	1074	33	1074

The observations are shown in the left graph of Figure 4.2. In the plot on the right, the data were divided into vertical strips (you could also choose horizontal strips), whereby each strip should contain approximately the same number of data points.

The observations of each strip should now be shown with box plots. In the left graph of Figure 4.3, the medians in the y-direction are now calculated for the observations of a strip and shown with lines. The length of these lines in the x direction corresponds to the length of the areas separated by the strips, and the position of the lines in the y direction corresponds to the medians. You can also see dotted vertical lines in the plot: The horizontal position of these lines corresponds to the median (in x-direction) of the individual areas (the length in y-direction has no meaning here).

In the right graph of Figure 4.3, the box plots are now shown for the individual areas separated by the stripes. The box plots are centered in the horizontal direction on the dotted lines shown in the graphic on the left.

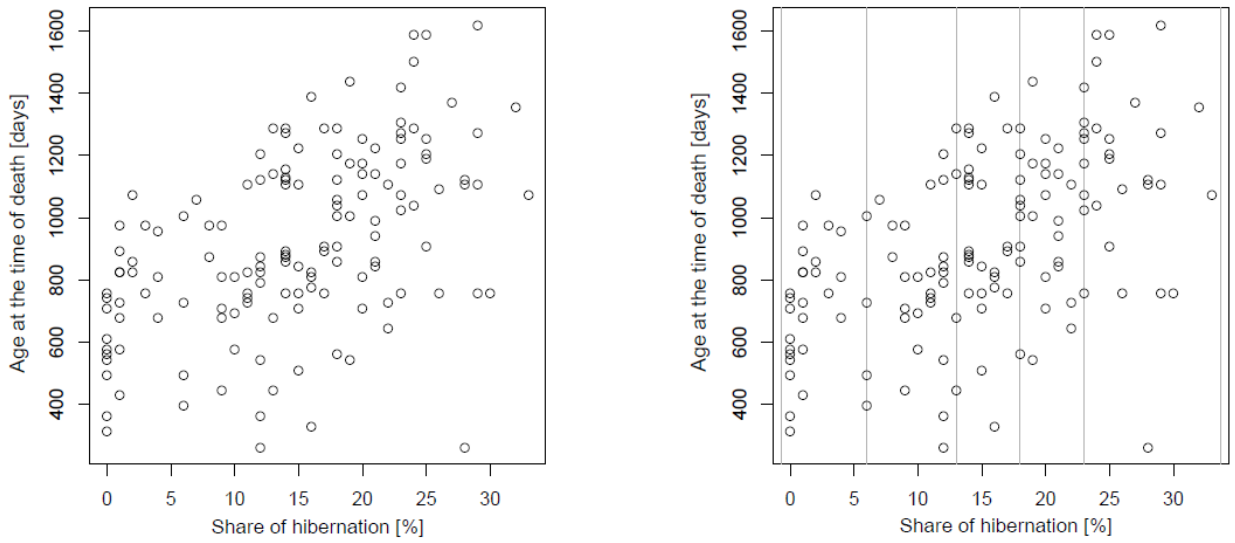


Figure 4.2: Division of the hamster data from Table 4.2 into strips. The areas should contain approximately the same number of data points.

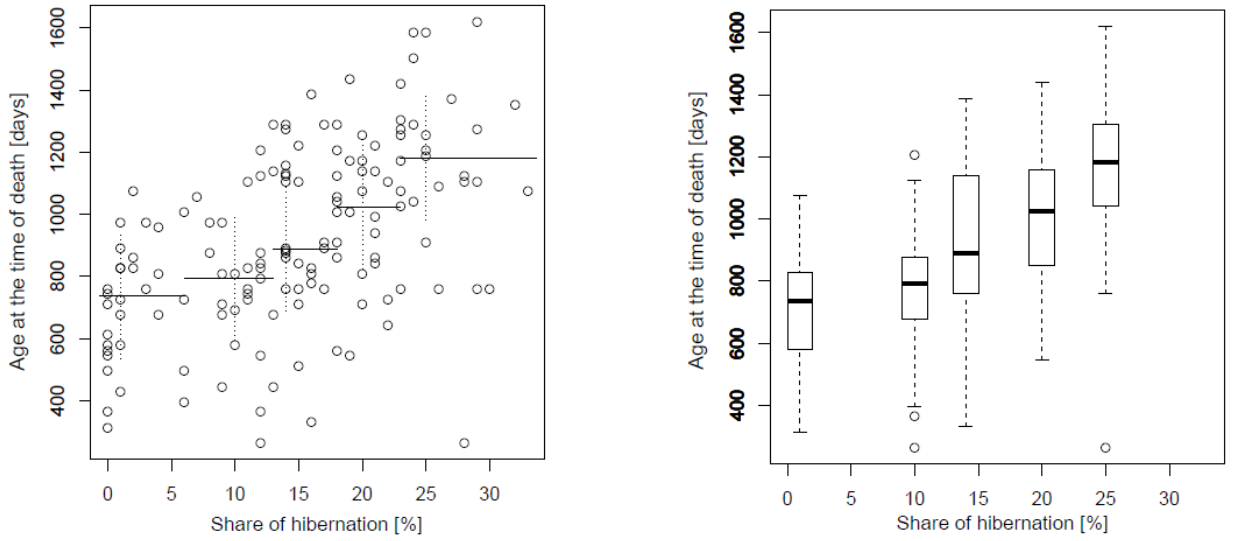


Figure 4.3: Scatter plot of the hamster data from Table 4.2 with strip medians (left) and presentation of the data as strip box plots (right).

The stripe box plots in Figure 4.3 show trends well. You can clearly see that the age of the hamsters increases as the proportion of hibernation increases. Note that the graph on the left in Figure 4.3 shows this trend well, and that the data can also be displayed.

For a more objective representation, the number of strips could be varied.

4.3 Density estimation in two dimensions

Similar to the one-dimensional case, a density function can also be estimated in the two-dimensional case. The basic principle is again to select a sub-area (square, circular) in which a given weight function is evaluated. The weight function can take the form of a *box car* function:

$$W(u,v) = \begin{cases} \frac{1}{\pi} & u^2 + v^2 \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

It applies again to $\int W(u,v)dvdx = 1$. The *local density* then has the form:

$$\hat{f}(x,y) = \frac{1}{h^2 n} \sum_{i=1}^{n} W(\frac{x - x_i}{h}, \frac{y - y_i}{h}) \quad \text{with } h = \text{Window width or window height}$$

Smoother densities can with the cosine weight function

$$W(u,v) = \begin{cases} \frac{1+\cos(\pi\sqrt{u^2+v^2})}{\pi} & u^2 + v^2 \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

be achieved.

Representation of $\hat{f}(x,y)$ by

1. Grayscale or color gradations
2. Isolines
3. Mountains
4. . . .

As an example, consider the Old Faithful Geyser data, which describes hot spring activity in Yellowstone National Park from Nov. 1-15. August 1985. There are 299 observations for the eruption time (in minutes) and the waiting time until the next eruption (the data are available in R in the *library(MASS)* under *data(geyser)*.) Figure 4.4 shows the original data. For the following representations, the bivariate density function of the normal distribution was chosen as the weight function. The window width and height were selected using a special procedure that is not mentioned here. The associated R function is called **kde2d** from the **MASS** package.

Figure 4.5 shows the representation of the two-dimensional density function using gray levels (left) and isolines (right). In Figure 4.6 the density function is represented spatially by mountains.

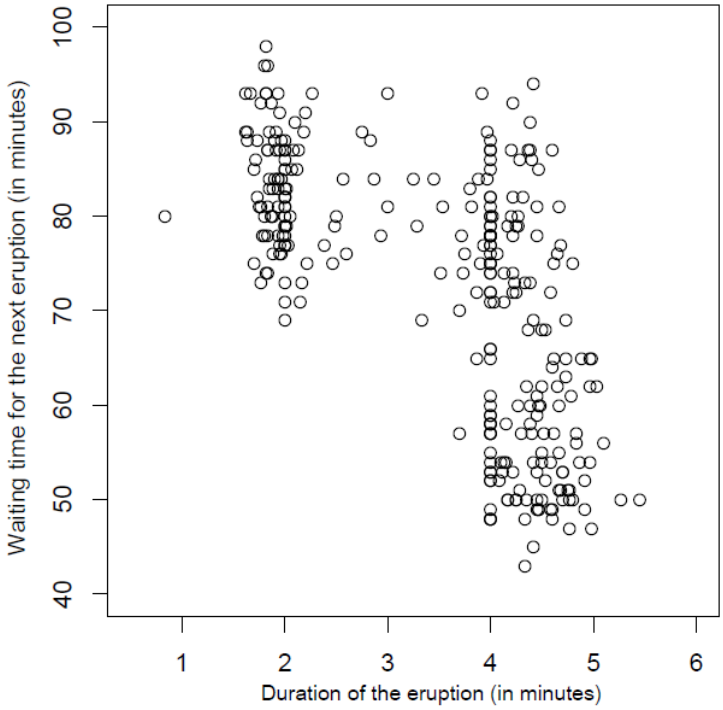


Figure 4.4: Old Faithful Geyser Data

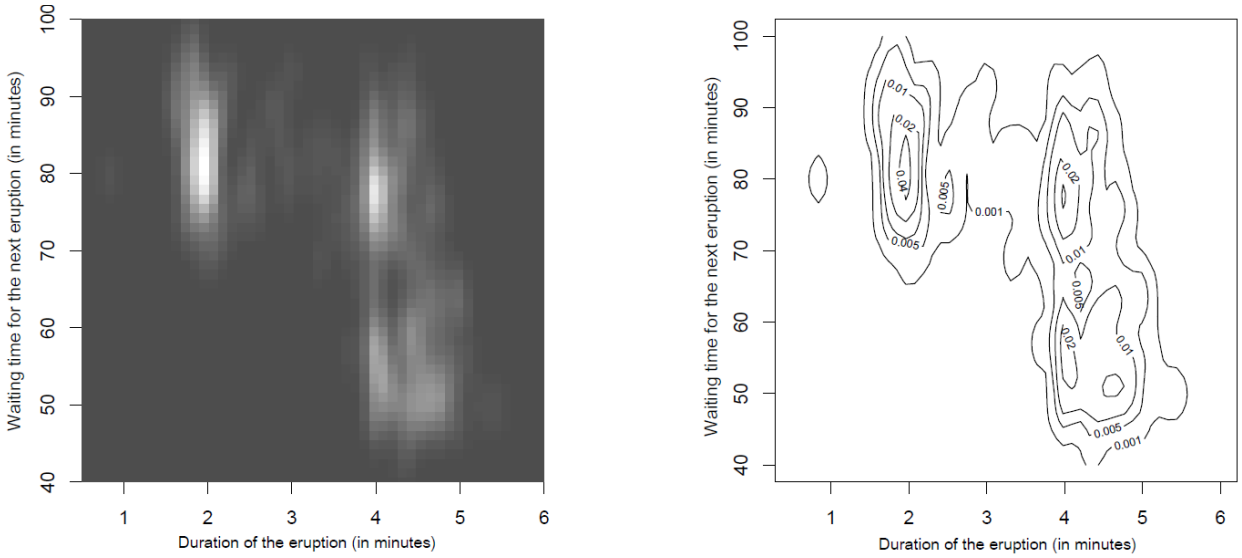


*Figure 4.5: Representation of the two-dimensional density function using gray levels (left) and isolines (right). (**image** or **contour**)*

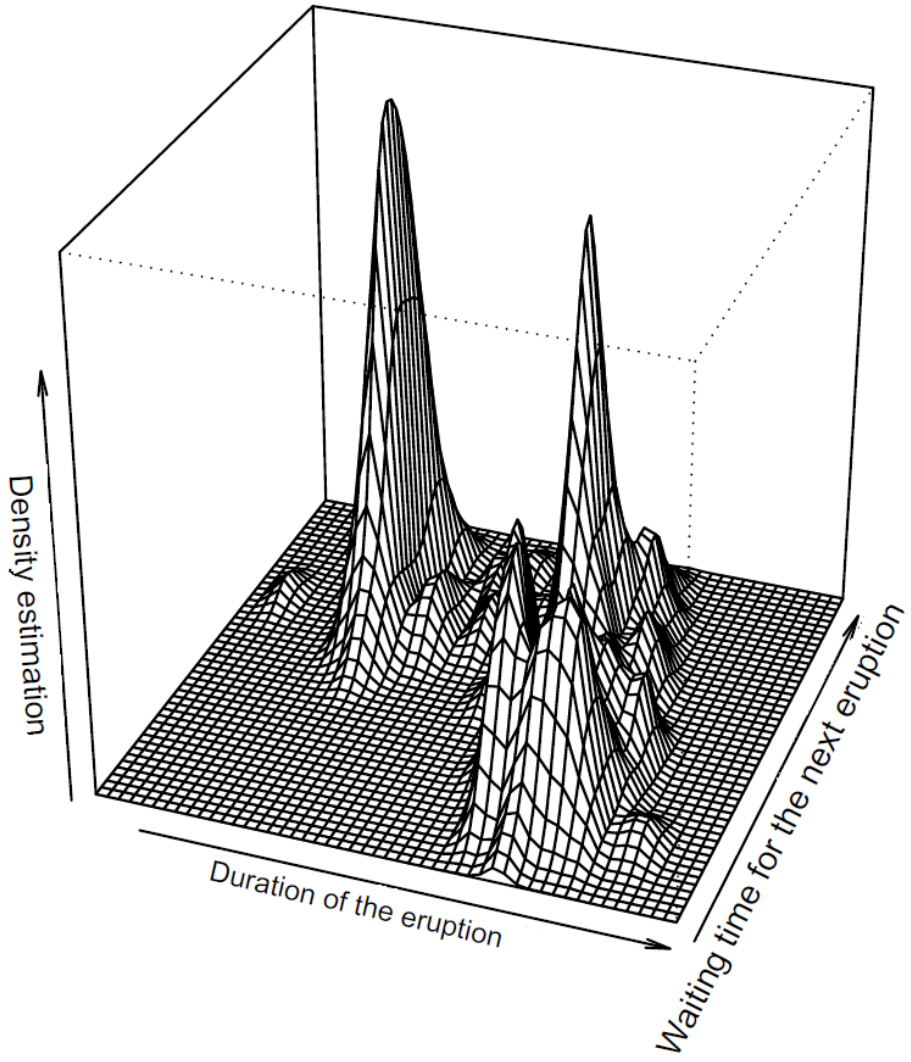


*Figure 4.6: Spatial representation of the 2-dimensional density function. (**persp**)*

Chapter 5

Robust estimate of linear trends

This problem is known in the literature as linear regression. Based on an input variable x (independent variable), an output variable y (dependent variable) should be predicted with the linear function

$$y = \alpha + \beta x + \epsilon \,,$$

where ϵ represents a random error. The abscissa distance α and the slope β are the parameters of a straight line.

There is now a sample of size n, i.e.$(x_1, y_1), \dots, (x_n, y_n)$, the parameters of the straight lines are to be estimated, i.e. coefficients α and β, so that the resulting straight line goes through the pairs of points "as best as possible":

$$y_i \approx \hat{\alpha} + \hat{\beta} x_i \qquad i = 1, \dots, n$$

The estimated y-values are also obtained with the estimated parameters

$$\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i \qquad i = 1, \dots, n,$$

which of course now lie exactly on a straight line. The errors that arise in the prognosis are called *residuals*

$$r_i = y_i - \hat{y}_i = y_i - \hat{\alpha} - \hat{\beta} x_i \qquad i = 1, \dots, n.$$

To estimate the parameters α and β, the least squares (LS) criterion is usually used:

$$(\hat{\alpha}_{LS}, \hat{\beta}_{LS}) = \operatorname{argmin}_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2 = \operatorname{argmin}_{\alpha,\beta} \sum_{i=1}^{n} r_i^2 \,.$$

The solution for the regression parameters is:

$$\hat{\beta}_{LS} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \quad \text{and} \quad \hat{\alpha}_{LS} = \bar{y} - \hat{\beta} \bar{x}$$

Although the LS estimators have good statistical properties under certain conditions, there is the serious disadvantage that they are sensitive to outliers. Outliers in the y-direction, but especially outliers in the x-direction, can have a massive influence on the parameter estimation. The reason is the criterion. Deviating values have a large residual square, and the minimum of the above criterion becomes smaller if the residuals are distributed "more evenly" (see Figure 5.1).

The following chapters therefore discuss regression methods that are more robust against outliers.

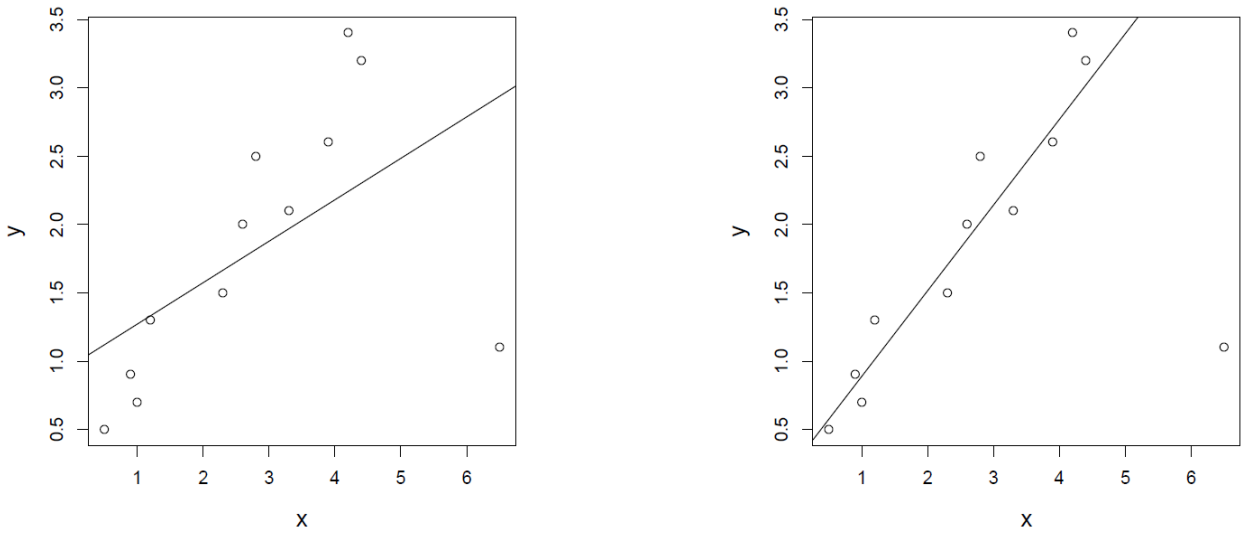


Figure 5.1: The usual least squares regression line (left) is sensitive to outliers. The hypothetical straight line in the plot on the right is robust. ***(lm)***

5.1 Robust straight line according to Tukey

It is an iterative method that was developed by Tukey (1970) and that is defined as follows:

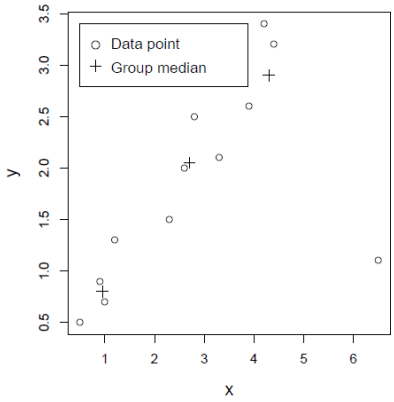
1. Sort the data pairs according to the x-values.
2. Division of the data pairs into 3 groups.
   Group L (left) pairs (x_i, y_i) with the n_L smallest x-values
   Group M (middle) pairs $(x_i, y_i)$with the n_M mean x-values
   Group R (right) pairs (x_i, y_i) with the n_R largest x-values
   $n_L + n_M + n_R = n$.

   Directive:

	$n = 3k$	$n = 3k+1$	$n = 3k+2$
n_L	k	k	$k+1$
n_M	k	$k+1$	k
n_R	k	k	$k+1$

   Pairs with the same x-values should be assigned to the same group.

3. Calculation of the medians for x-values and y-values in the individual groups:
   $(x_L, y_L), (x_M, y_M), (x_R, y_R)$ with
   $x_L = \underset{(x_i,y_i)\in L}{\text{median}}\, x_i\,,\ y_L = \underset{(x_i,y_i)\in L}{\text{median}}\, y_i$, etc.



4. $\hat{\beta}_0 := \frac{y_R - y_L}{x_R - x_L}$ first estimate for β

$$y = \alpha + \beta(x - x_M) \quad \Rightarrow \quad \alpha = y - \beta(x - x_M)$$

$$\begin{aligned} \hat{\alpha}_0^{(*)} &= \frac{1}{3}[(y_L - \hat{\beta}_0(x_L - x_M)) + y_M + (y_R - \hat{\beta}_0(x_R - x_M))] \\ &= \left(\frac{1}{3}(y_L + y_M + y_R) - \hat{\beta}_0\frac{1}{3}(x_L + x_M + x_R)\right) + \hat{\beta}_0 x_M := \hat{\alpha}_0 + \hat{\beta}_0 x_M \end{aligned}$$

5. Residuals

$$r_i^{(0)} := y_i - (\hat{\alpha}_0^{(*)} + \hat{\beta}_0(x_i - x_M)) \qquad \text{for } i = 1, \ldots, n$$

6. Steps 3-5 with the data pairs $(x_i, r_i^{(0)})$ $i = 1, \ldots, n$ results in the straight line $r_i^{(0)} = \hat{\alpha}_1 + \hat{\beta}_1(x_i - x_M)$ and the residuals $r_i^{(1)} := r_i^{(0)} - (\hat{\alpha}_1 + \hat{\beta}_1(x_i - x_M))$
7. Iteration: all further steps as in step 6, i.e. $r_i^{(j)} \rightarrow \hat{\alpha}_{j+1},\ \hat{\beta}_{j+1} \rightarrow r_i^{(j+1)}$ until a suitable termination criterion is met.
8. $\hat{\beta} = \hat{\beta}_0 + \hat{\beta}_1 + \hat{\beta}_2 + \cdots$
   $\hat{\alpha} = \hat{\alpha}_0 + \hat{\alpha}_1 + \hat{\alpha}_2 + \cdots$

Note: weaknesses in the procedure

- possibly very slow convergence with "poor" configuration of the data.
- sometimes no convergence → but can be repaired by modifying the algorithm slightly.

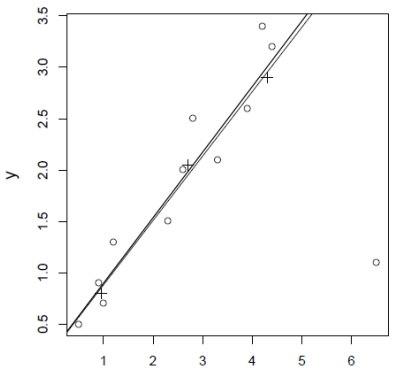


*Figure 5.2: Tukey's robust straight line after 3 iteration steps. **(line)***

5.2 Robust straight line by Theil

Theil (1950) takes into account the medians of pairwise slopes in the calculation of the straight line. The prerequisite for the procedure is that all x_i are different.

$$\hat{\beta}_{ij} := \frac{y_j - y_i}{x_j - x_i} \qquad 1 \le i < j \le n \qquad \text{i.e } \binom{n}{2} = \frac{n(n-1)}{2} \text{ increases.}$$

$$\hat{\beta}_T := \underset{1 \le i < j \le n}{median} \ (\hat{\beta}_{ij})$$

The "degree of robustness" of different methods can be compared. In general, one can even specify the robustness for each estimator (not only for the parameters in regression), i.e. the sensitivity to a proportion of outliers. This measure is called the **breaking point** for an estimator.

n … Sample size

k … maximum number of sample elements that can be replaced by any values without the estimated value becoming unlimited.

The breakpoint of an estimator is defined as the minimum proportion $\frac{k}{n}$ of the data that can be replaced by any value, so that the estimator delivers nonsensical results.

Breakpoints for different procedures:

Sample means	0.00
upper quartile	0.25
lower quartile	0.25
Median	0.50
IQR	0.25
robust line according to TUKEY	$\frac{1}{6}$
robust line according to THEIL	0.29
robust line according to SIEGEL	0.50

If the outlier proportion is smaller than this value, the estimator will change, but meaningful results can still be expected. In special cases, however, a much larger proportion of outliers can occur without having a massive impact on the estimated value, e.g. with IQR, even an outlier proportion of up to 50% can remain with limited influence if the "bad" data values are evenly divided between both ends of the ordered sample. The definition of the breakpoint means, however, a contamination of the data that is designed in such a way that it can cause as much "damage" as possible (e.g. only on one side of the value range).

Breaking point for the robust straight line after Theil:

$$\binom{k}{2} + k(n-k) = \frac{\binom{n}{2}}{2}$$

$\binom{k}{2}$ … Breaking point for the robust straight line after Theil.
$k(n-k)$ … Number of increases with only 1 end point unusable.
$\binom{n}{2}$ … Number of straight lines, breaking point for the Median = 0.5

$$\begin{aligned}
\frac{k(k-1)}{2} + k(n-k) &= \frac{n(n-1)}{4} \\
k^2 - k + 2nk - 2k^2 &= \frac{n(n-1)}{2} \\
-k^2 + 2nk - k &= \frac{n^2(1-\frac{1}{n})}{2} \\
\left(\frac{k}{n}\right)^2 - 2\frac{k}{n} + \frac{k}{n^2} &= -\frac{1}{2} + \frac{1}{2n} \\
\left(\frac{k}{n}\right)^2 - 2\frac{k}{n} + \frac{1}{2} &\approx 0 \qquad \text{for big } n \\
\frac{k}{n} &= 1 \pm \sqrt{1 - \frac{1}{2}} \\
\frac{k}{n} &= 1 - 0.71 \quad = \quad 0.29
\end{aligned}$$

5.3 Robust straight line according to Siegel (repeated median line)

Siegel's estimator (1982) is also based on medians of pairwise slopes, but these are calculated repeatedly:

$$\hat{\beta}_{RM} := \underset{1 \le i \le n}{median} \left(\underset{\substack{1 \le j \le n \\ j \ne i}}{median} (\hat{\beta}_{ij}) \right)$$

$$\hat{\alpha}_{RM} := \underset{1 \le i \le n}{median} (y_i - \hat{\beta}_{RM} x_i)$$

i.e. when calculating the median for $\hat{\beta}_{RM}$, the slope between (x_i, y_i) and (x_j, y_j) appears twice.

Break point: 0.5

Proof: for n = 2k

If $k-1$ elements are outliers, then $\underset{j \ne i}{\text{median}}(\hat{\beta}_{ij})$ is for $k-1$ indices $i_1, \ldots, i_{k-1}$ outliers.

If (x_i, y_i) is a "good" data point, then $\underset{j \ne i}{\text{median}}(\hat{\beta}_{ij})$ is also "good", since only $k-1$ $\left(< \frac{n}{2}\right) \hat{\beta}_{ij}$ are outliers.

$\Rightarrow in \underset{1 \le i \le n}{\text{median}}(\underset{\substack{1 \le j \le n \\ j \ne i}}{\text{median}}(\hat{\beta}_{ij}))$ only $k-1$ $\left(< \frac{n}{2}\right)$ values are outliers.

$$\lim_{n \to \infty} \frac{k-1}{n} = \frac{k-1}{2k} = \frac{1}{2}$$

An analogous proof holds for $n = 2k + 1$.

Note: Greater robustness (= resistance) is bought at the cost of greater computing effort.

5.4 Least Median of Squares (LMS) Regression

While the sum of the squared residuals is minimized in LS regression, the LMS criterion (Rousseeuw, 1984) is defined as:

$$(\hat{\alpha}_{LMS}, \hat{\beta}_{LMS}) = \text{argmin}_{\alpha,\beta} \, \text{median}_i r_i^2 \ .$$

So, you replace the sum with the median, which results in a strong robustness (breaking point 50%).

The solution for α_{LMS} and β_{LMS} is found with a "subsampling" algorithm, see below.

5.5 Least Trimmed Squares (LTS) Regression

LTS regression (Rousseeuw, 1984) tries to minimize only the sum of the smallest squared residuals. Let $r^2_{(1)} \leq r^2_{(2)} \leq \cdots \leq r^2_{(n)}$ be the squared residuals in order of magnitude. Then the LTS criterion is:

$$(\hat{\alpha}_{LTS}, \hat{\beta}_{LTS}) = \text{argmin}_{\alpha,\beta} \sum_{i=1}^{h} r^2_{(i)}$$

with $\frac{n}{2} < h < n$. Depending on the choice of h, you get a method with a breakpoint between 0 and 50%. The robustness suffers with a lower breaking point, but the efficiency (precision of the estimator) is increased.

Note: For a simple regression problem (x versus y), the advantage of robust regression often does not seem to be immediately apparent, because outliers in two-dimensional space can still be recognized graphically without further ado. However, all of the methods mentioned above also work in the case of several x variables (multiple regression), and then the graphic outlier detection is generally no longer possible.

Algorithm: An algorithm based on "subsampling" was developed for both LMS and LTS regression. In the following we assume that there are $p \geq 1$ x-variables that can be used to explain y:

1. Randomly choose $p + 1$ observations. This precisely determines the regression line (regression hyperplane).
2. Calculate all residuals and from them the LMS or LTS criterion.
3. Iterate 1. and 2. "very often", and choose the final (approximate) solution that gives the smallest value of the objective function.

The computational effort increases especially with a larger p.

Fast-LTS algorithm: A faster algorithm was developed for LTS regression:

1. Randomly choose h observations.
2. Estimate the regression parameters with the h observations using LS regression (least squares - not robust!).
3. Calculate the residuals of all observations and order their absolute values after the magnitude.
4. Take those h observations that come from the smallest (absolute) residuals from point 3.
5. Iterate 2.-4. until convergence. This is assured because the sum of the squared residuals chosen never becomes larger.
6. Perform steps 1.-5. several times through. The approximate solution is the one that gives the smallest value of the objective function.

R-code: In the *library(rrcov)* the Fast-LTS algorithm is implemented in the function *ltsReg*. The *library(robustbase)* contains the function *lmrob*, which is copied from the input/output structure

of the function *lm* for classic LS estimation. *lmrob* performs robust MM-regression, a method that achieves the best possible robustness with optimal precision.

5.6 Multiple x variables

In the more general case, one would like to explain an output variable y not only by an input variable x, but there are several input variables $x_1, x_2, \ldots, x_p$. These p inputs could be p different variables that are measured in a production process in order to ultimately predict the product quality y. If one takes again a linear prediction function of the inputs, then the regression model has the following form:

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$$

Here α is again the abscissa distance, the βs are the slope parameters for the individual inputs, and ϵ is the random error.

For each of these quantities there are then n observations, and one thus has the sample $(x_{i1}, x_{i2}, \ldots, x_{ip}, y_i)$ for $i = 1, \ldots, n$. With the estimated parameters $\hat{\alpha}, \hat{\beta}_1, \ldots, \hat{\beta}_p$ you get the predicted values again.

$$\hat{y}_i = \hat{\alpha} + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}$$

and the residuals

$$r_i = y_i - \hat{y}_i \text{ for } i = 1, \ldots, n.$$

As in the simple linear regression case (one x variable), different criteria can also be used here in order to arrive at estimates for the parameters. The LS criterion would be obvious, but again with the disadvantage that it is sensitive to outliers. Note here that outliers in the x-direction (so-called leverage points) can now be outliers in p dimensions, i.e. in the variables $x_1, \ldots, x_p$, and are therefore difficult or impossible to find by graphic means.

Not all of the methods dealt with in the previous chapters can be used here as robust variants, but LMS or LTS regression, as well as the mentioned MM-regression, are ideal. The criteria are exactly the same, only the algorithms for estimating the parameters become a little more complex.

Example: We consider the *rice* data from the *rrcov* package with subjective evaluations of 105 different *rice* varieties. The y variable represents the overall evaluation, the x variables are aroma (flavor), appearance, taste, stickiness and toughness. A model might now be obtained in order to be able to predict the quality on the basis of the x-variables.

For now, we only consider simple linear regression, with the x-variable aroma. LS regression and robust MM regression are compared as methods. The resulting regression lines are shown in Figure 5.3 on the left. Apparently, the estimated coefficients of the two variants are very similar to each other. There seem to be no major outliers that would influence LS regression more strongly.

```
library(robustbase)
library(rrcov)
data(rice)
attach(rice)

plot(Favor,Overall_evaluation)
r1 <- lm(Overall_evaluation~Favor,data=rice)
abline(r1,col="red")
r2 <- lmrob(Overall_evaluation~Favor,data=rice)
abline(r2,col="blue")
legend("topleft",legend=c("LS-Regression","MM-Regression"),
       lty=c(1,1),col=c("red","blue"))
```

Figure 5.3 on the right shows the result of the classical and robust regression for two explanatory variables. Here, too, there is not much difference to be seen, as can be seen from the estimated regression coefficients:

```
mod <- lm(Overall_evaluation~Favor+Appearance,data=rice)
coef(mod)
(Intercept)  Favor        Appearance
-0.1385352 0.5041728 0.7237637

mod2 <- lmrob(Overall_evaluation~Favor+Appearance, data=rice)
coef(mod2)
(Intercept)  Favor        Appearance
-0.1582099 0.4895092 0.7668813
```

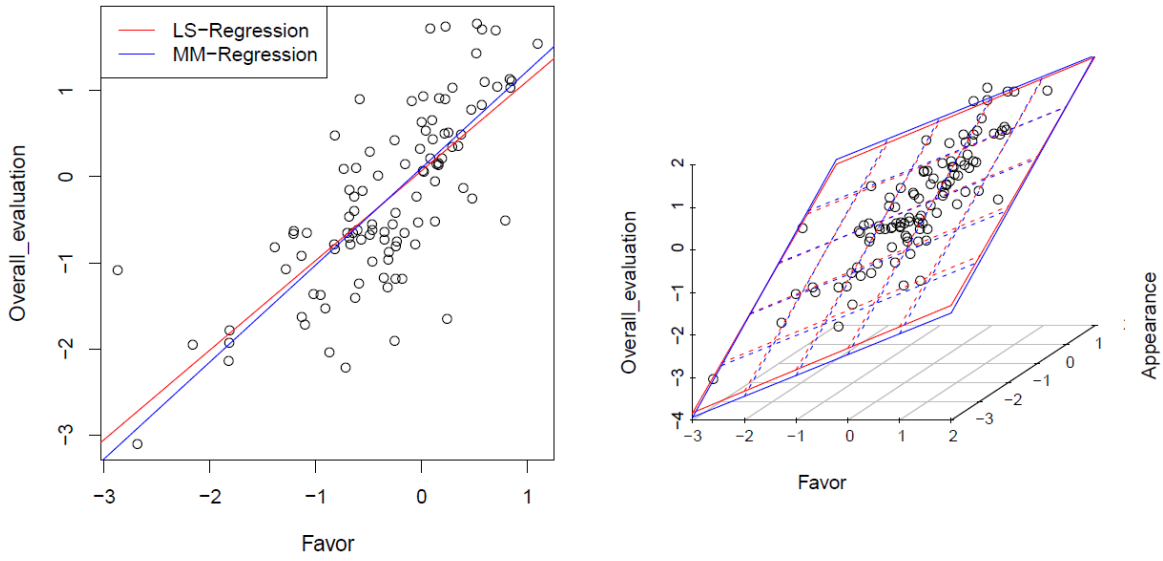


Figure 5.3: Linear regression (LS and MM) for the rice data, with one (left) or two (right) explanatory variables.

Finally, all 5 explanatory variables should be used for the forecast. A visualization of the problem is no longer possible. For this we take a closer look at the so-called inferential statistics:

```
re1 <- lm(Overall_evaluation~.,data=rice)
summary(re1)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.13026    0.03507  -3.715 0.000337 ***
Flavor       0.19359    0.05398   3.586 0.000523 ***
Appearance   0.10829    0.05993   1.807 0.073805 .
Taste        0.53905    0.08163   6.604 2.02e-09 ***
Stickiness   0.40599    0.07146   5.682 1.34e-07 ***
Toughness    0.03513    0.05733   0.613 0.541460
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2745 on 99 degrees of freedom
Multiple R-squared:  0.9306,Adjusted R-squared:  0.927
F-statistic: 265.3 on 5 and 99 DF,  p-value: < 2.2e-16
```

In the case of the LS regression model, the variables *Flavor*, *Taste* and *Stickiness* contribute significantly to the explanation. The estimated regression coefficients can be read under *Estimate*. *Appearance* is on the verge of significance. The model explains the output variable very well, 93% (*multiple* R-*squared*).

The following results are obtained for MM-regression:

```
re2 <- lmrob(Overall_evaluation~.,data=rice)
summary(re2)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.09841    0.03729  -2.639  0.00966 **
Flavor       0.21569    0.07926   2.721  0.00769 **
Appearance   0.02917    0.07986   0.365  0.71572
Taste        0.60889    0.09639   6.317 7.64e-09 ***
Stickiness   0.36465    0.07954   4.584 1.33e-05 ***
Toughness    0.01428    0.04812   0.297  0.76726
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Robust residual standard error: 0.2131
Multiple R-squared:  0.954,Adjusted R-squared:  0.9517
Convergence in 18 IRWLS iterations

Robustness weights:
 observation 75 is an outlier with |weight| = 0 ( < 0.00095);
 6 weights are ~= 1. The remaining 98 ones are summarized as
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0154  0.8960  0.9506  0.8880  0.9795  0.9989
```

The coefficient of *Appearance* is very different from the LS result, and also far from significance. Observation 75 is an outlier. Figure 5.4 shows diagnostic plots for MM-regression that are obtained with *plot(re2)*. The graph on the left shows the estimated values $\hat{y}$ versus y. In fact, there are a few deviating observations that the model does not predict as well. In general, however, you get very good estimates. The right graph shows robust Mahalanobis-Distances (comes later in the script)

against the standardized residuals, i.e. residuals divided by the standard deviation of the residuals, which should be in the area of the dashed lines. In fact, there are a few outliers in the y-direction.

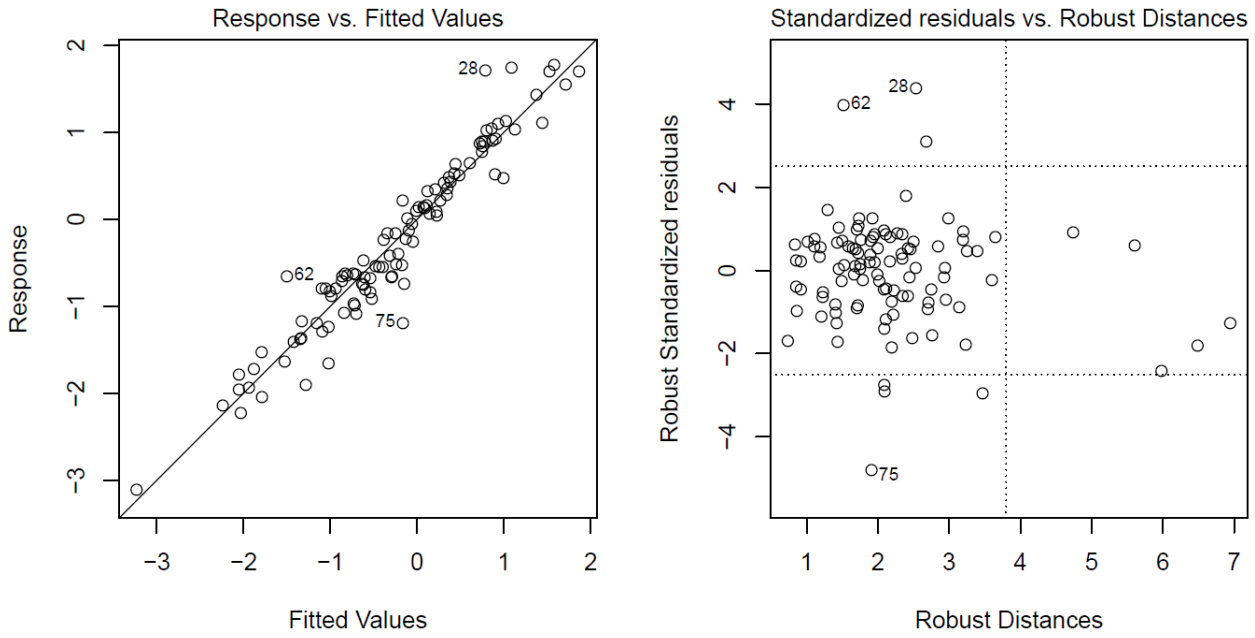


Figure 5.4: Diagnostic plots for robust MM-regression for the rice data.

Chapter 6

Smoothing and estimating non-linear trends

We assume that pairs of observations $(x_1, y_1), \dots, (x_n, y_n)$ exist. Here the x-values could also be points in time, and the y-values could be the observed values of a variable over time.

The following two sections deal with smoothing a two-dimensional signal, and thus estimating non-linear two-dimensional trends. The sections differ in that equidistant y-values are initially assumed (e.g. regular points in time), and then non-equidistant y-values are also possible (e.g. measurements on a second continuous characteristic).

6.1 Nonlinear smoother for equidistant (time-) points

We assume that there is a time series in the form of measurement points $x_t,\ t = \cdots,\ -2,\ -1,\ 0, 1, 2, \dots$ with a constant difference between the measurement times. Our goal is to use simple exploratory methods to smooth the time series, which is also to a certain degree robust against disturbances in the signal (peaks, jumps). The smoothing is intended to identify patterns in the time series.

Time series	=	Smoothing curve	+	Residuals
x_t	=	Gx_t	+	r_t
x_t	=	z_t	+	r_t

G … Smoothing algorithm

$r_t := x_t - Gx_t$

Linear filters:

$$z_t = \sum_{i=-l_1}^{l_2} \alpha_i x_{t+i} \ \ weighted\ sum\ of\ the\ x_i\ in\ a\ neighborhood\ of\ t$$

with $0 \le l_1,\ 0 \le l_2,\ \alpha_{-l_1} \ne 0,\ \alpha_{+l_2} \ne 0$.

Note: Linear filters are very sensitive to outliers.

Robust procedures:

- Usually formed with the help of the median.
- Rather difficult to analyze mathematically.
- Area of application: EDA, robust adjustment of seasonal fluctuations, signal and image processing.

Range of a smoothing algorithm: (largest index - smallest index + 1) with regard to the x_t, from which z_t is calculated.

Median smoothing algorithms with odd span 2s + 1:

$$z_t = Gx_t := median(x_{t-s}, x_{t-s+1}, \dots, x_t, x_{t+1}, \dots, x_{t+s})$$

E.g. $s = 1$: $z_t = median(x_{t-1}, x_t, x_{t+1})$

$\Longrightarrow$ get time series $\dots, z_{t-1}, z_t, z_{t+1}, \dots$

Median smoothing algorithms with an even span 2s:

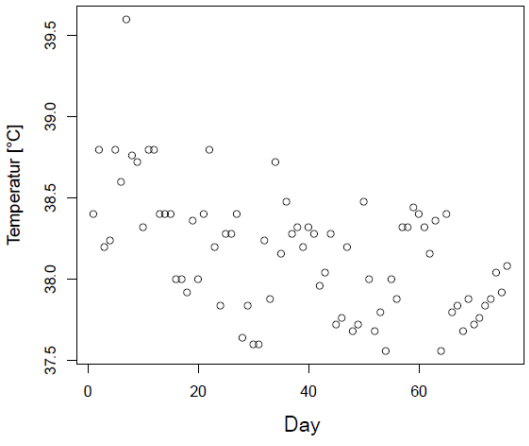
$$z_{t+\frac{1}{2}} = Gx_t := median(x_{t-s+1}, \dots, x_t, x_{t+1}, \dots, x_{t+s})$$

E.g. $s = 1$: $z_{t+\frac{1}{2}} = median(x_t, x_{t+1})$

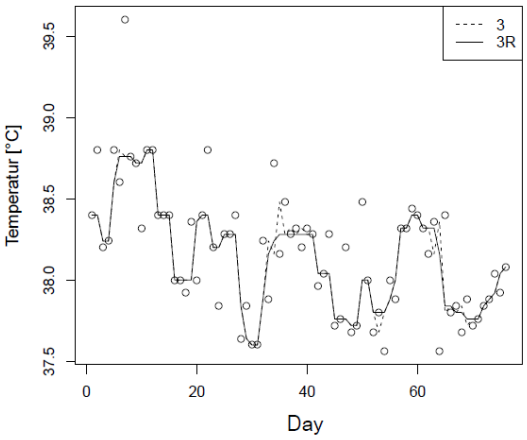
$\Longrightarrow$ get time series $\dots, z_{t-1-\frac{1}{2}}, z_{t-\frac{1}{2}}, z_{t+\frac{1}{2}}, z_{t+1+\frac{1}{2}} \dots$

An integer index t is achieved by a second *even median smoothing algorithm*. Mostly one takes $s = 1$. $\Longrightarrow z_t = G_2 G_1 x_t$

Figure 6.1 (a) shows a time series of body temperatures of a cow, which was measured daily on 75 consecutive days at 6:30 a.m. Figure 6.1 (b) shows the results of various smoothing algorithms.



(a) Original data: Morning body temperatures measured in a cow on 75 consecutive days.



(b) "3" smoothing: one-time application of the median algorithm, and "3R" smoothing: repeated application of the median algorithm with a range of 3 (until convergence).

Figure 6.1: Effects of different smoothers on the time series of the cow data.

6.2 Robust filtering with repeated median

Smoothing and filtering are equivalent tasks. With smoothing one is more interested in the signal, with filters also in the smoothed signal, but also in the residuals, especially residuals that are outliers. Such "significant" deviations can provide important information about the underlying data-generating process. However, a prerequisite for reliable outlier identification is that the smoothing is robust, i.e. that it is influenced as little as possible by the outliers themselves.

Some such filter algorithms are implemented in the R package **robfilter**, which also provide outlier diagnostics. The repeated median (see Section 5.3) is used there because it is very robust and quickly calculable, and is therefore also suitable for online filtering (these algorithms were developed in connection with online monitoring in intensive care medicine).

As in the previous section, we assume a time series with measured values x_t that was observed at the (equidistant) times $t = 1, \ldots, T$. The filter should work locally again, i.e. use information that lies within a certain time window, with the time window being pushed forward continuously. To do this, we assume the range as 2s + 1, with s > 0. So, if we want a smoothing for x_t (filter at time t), then the values $\{x_{t-s}, x_{t-s+1}, \ldots, x_t, x_{t+1}, \ldots, x_{t+s}\}$ are taken into account.

We denote the underlying signal as μ_t, for $t = 1, \ldots, T$, which, however, cannot be observed and should be estimated. The time series then comes about by $x_t = \mu_t + \epsilon_t$, where ϵ_t represents an error that is characterized by a mixture of normal distribution and a "heavy-tailed" distribution (generates outliers). We assume that μ_t can be approximated locally (within a time window with a span of $2s + 1$) by a linear function:

$$\mu_{t+i} \approx \mu_t + \beta_t * i \text{ for } i = -s, -s+1, \ldots, s.$$

The parameters of this linear function are the "level" μ_t and the slope β_t, which have to be estimated. In principle, this can be carried out with any (robust) regression method, whereby the authors propagate the repeated median:

$$\hat{\beta}_t = \underset{-s \le i \le s}{median} \left(\underset{\substack{-s \le j \le s \\ j \ne i}}{median} \left(\frac{x_{t+i} - x_{t+j}}{i-j} \right) \right)$$

$$\hat{\mu}_t = \underset{-s \le i \le s}{median} \left(x_{t+i} - \hat{\beta}_t \cdot i \right)$$

For outlier diagnostics, an estimate of the standard deviation of the residuals σ_t at each point in time t is required. This can be obtained with the above principle of the local linear approximation. The residuals are obtained locally at time t

$$r_t(t+i) = x_{t+i} - \hat{\mu}_t - \hat{\beta}_t * i \text{ for } i = -s, -s+1, \ldots, s.$$

One can now use any (robust) scatter estimator to estimate σ_t, such as the Qn, see Section 3.1:

$$\hat{\sigma}_t = 2.219 * \{|r_t(t+i) - r_t(t+j)|; i < j\}_{(k)} \text{ with } k = \binom{h}{2} \text{ and } h = \left\lfloor \frac{(2s+1)}{2} \right\rfloor + 1.$$

With this scatter estimate one can now do the usual robust outlier diagnosis: an outlier is diagnosed if $|\hat{r}_t| > 2 * \hat{\sigma}_t$, where $\hat{r}_t = x_t - \hat{\mu}_t$ is the estimated residue at time t.

Example: We consider the *cow* data from the last section. The following R code shows an application of the algorithm described above, where the range 2s + 1 = 7 is chosen (must be an odd number). The result is shown in Figure 6.2 on the left. The right graph shows the result for a range of 2s + 1 = 19. So, you can see that the result depends strongly on this parameter. An algorithm (*scarm.filter()*) is also implemented in the *robfilter* package, which automatically selects the range on the basis of statistical tests. However, "enough" data is required for this.

```
library(robfilter)
res <- robust.filter(kuh,width=7) # trend by RM, scale by Qn
plot(res)
res
 # $level
 # [1] 38.4 38.48667 38.57333 38.66 38.47733     ...
 # $slope
 # [1] 0.086667 0.086667 0.086667 0.086667 0.122667     ...
 # $sigma
 # [1] 0.830133 0.830133 0.830133 0.830133 0.833152     ...

# indicate outliers:
ind <- which(res$ol!=0)
points(ind,kuh[ind])
```

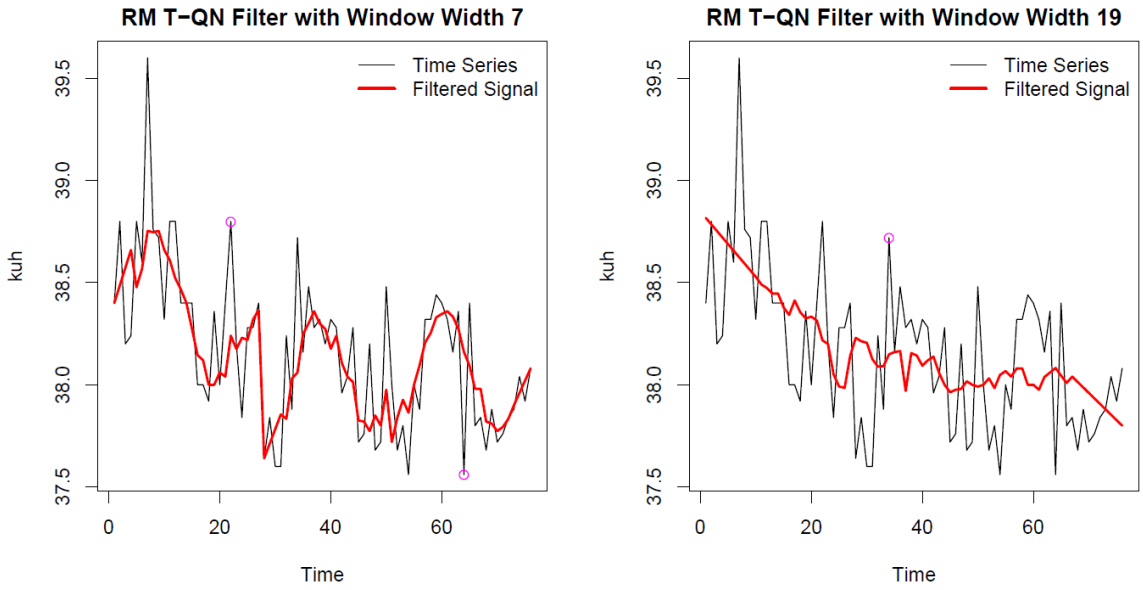


Figure 6.2: Application of a robust filter to the cow data, with window widths 7 (left) and 19 (right). Detected outliers are marked.

6.3 LOWESS

LOWESS stands for *LOcally WEighted regression Scatter plot Smoothing* and is a smoothing method that can be used on scatter plots and that allows trends in the data to be recognized. Conceptually, it works like a non-linear regression method: x-data are given and the corresponding y-data are smoothed. LOWESS could also be used in the context of time series.

About the n 2-dimensional data (x_i, y_i), $i = 1, \ldots, n$, nothing is assumed. LOWESS is described by the following algorithm:

1. f . . . Proportion of points that should be used in smoothing.
   $$q := \lfloor nf + 0.5 \rfloor$$
2. Choose the q points closest to (x_i, y_i) (x_k, y_k) ((x_i, y_i) is contained in these q points), for i = 1,. . . , n.

   $d_{ik} := | x_i - x_k |$ Distance between (x_i, y_i) and (x_k, y_k) (only the x-direction is relevant!).
   $d_i := | x_i - x_{i_{max}} |$ with i_{max} = index of the point furthest from (x_i, y_i) of the selected q points.
   Tricubic weight function:
   $$T(t) := \begin{cases} (1 - |t|^3)^3 & |t| < 1 \\ 0 & \text{sonst} \end{cases}$$
3. Weight of (x_k, y_k) with respect to (x_i, y_i): $t_i(x_k)$
   $$t_i(x_k) := \begin{cases} T\left(\frac{|x_i - x_k|}{d_i}\right) = T\left(\frac{d_{ik}}{d_i}\right) & d_i \neq 0 \\ 1 & d_i = 0 \end{cases}$$

   Note: $d_i = 0$ means that all q points have the same x-coordinate.
4. Weighted regression for all i: i.e.
   $$\min \sum_{k=1}^{n} t_i(x_k)(y_k - a - bx_k)^2 \rightarrow \hat{a}^{(i)}, \hat{b}^{(i)} \quad \rightarrow \quad \hat{y}_i := \hat{a}^{(i)} + \hat{b}^{(i)} x_i \qquad i = 1, \ldots, n$$

   $d_i = 0 \rightarrow \hat{b}^{(i)}$ not estimable $\rightarrow \hat{y}_i := \hat{a}^{(i)}$ ($\hat{y}_i$ is set to a constant, e.g. to the median of the q considered y_k
5. Compute residuals $r_i := y_i - \hat{y}_i$, for $i = 1, \ldots, n$.
6. Calculate new robust weights $w(x_k)$ with the weight function $B(t)$ (Biweight)
   $$B(t) := \begin{cases} (1 - t^2)^2 & |t| < 1 \\ 0 & \text{sonst} \end{cases}$$
   $$m := \underset{1 \leq k \leq n}{median} \; | r_k |$$

   (with normal distribution, m is an estimate for $\approx \frac{2}{3}\sigma \Rightarrow 3m \approx 2\sigma$)
   $$w(x_k) := B\left(\frac{r_k}{3m}\right)$$
   For each point, weights are assigned that depend on the size of the residuals. Points with very large residuals (outside the 2σ range) are completely weighted down.
7. Weighted regression as in point 4, but with weights $w(x_k)t_i(x_k)$. Repeat steps 4 - 7 several times.
8. Steps 1 - 7 for each data point (or for each different x-value) provide a sequence of points that are linearly interpolated.

Annotation:

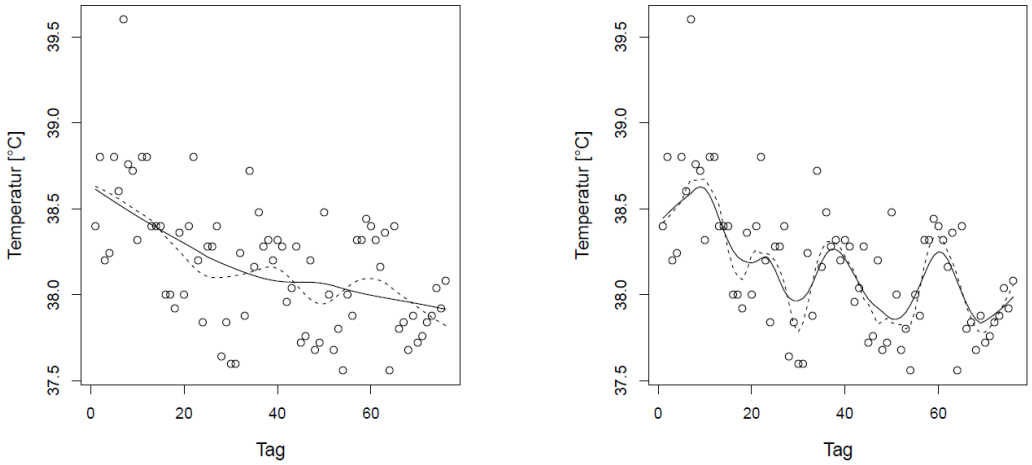
1. f between 1/3 and 2/3 gives good results.

2. In the case of the weighted regression in point 4, there is also the version that non-linear regression, but rather higher order regression is used. Linear regression would consider the model $y = a + bx +$ error, quadratic regression would consider $y = a + bx + cx^2 +$ error. This extension can be found in *loess* R function.
3. In order to see changes in the scatter depending on the x_i, the LOWESS method can be applied to the absolute residuals. This process is also called spread smoothing.

$$(x_i, y_i) \rightarrow \begin{cases} \hat{y}_i & \dots \text{ smoothed values} \\ r_i & := y_i - \hat{y}_i \end{cases}$$

LOWESS Smoothing the scatter plot $(x_i, \mid r_i \mid)$.

In Figure 6.3 the LOWESS algorithm is applied to the cow data from Figure 6.1 (a). The effects of different choices of the parameter *f* (proportion of the points used for smoothing) are shown.



(a) Smoothing with f = 2/3 (solid line) and f = 1/3 (dashed line).

(b) Smoothing with f = 1/5 (solid line) and f = 1/8 (dashed line).

Figure 6.3: LOWESS Smoothing: Effects of the choice of parameter f on the smoothed curve.

Figure 6.4 shows the LOWESS smoothing applied to the hamster data from Table 4.2. Spread smoothing is carried out in the right graph.

6.3.1 Upper and Lower Smoothing

In addition to the smoothed values, Upper and Lower Smoothing also offers scatter information. Proceed as follows:

1. LOWESS smoothing of the scatter diagram (x_i , y_i). The smoothed values are $\hat{y}_i$.
2. Division of the residuals $r_i := y_i - \hat{y}_i$ into positive and negative residuals:

r_i^+ positive residuals
x_i^+ abscissas to the r_i^+
$\hat{y}_i^+$ smoothed values for the r_i^+

r_i^- negative residuals
x_i^- abscissas to the r_i^-
$\hat{y}_i^-$ smoothed values for the r_i^-

3. Smoothing of (x_i^+, r_i^+) and (x_i^-, r_i^-)

$$(x_i^+, r_i^+) \to \hat{r}_i^+;\ (x_i^-, r_i^-) \to \hat{r}_i^-$$

Draw the curves $(x_i, \hat{y}_i)$, $(x_i^+, \hat{y}_i^+ + \hat{r}_i^+)$ and $(x_i^-, \hat{y}_i^- + \hat{r}_i^-)$.

Figure 6.5 shows an example of this procedure with the hamster data from Table 4.2.

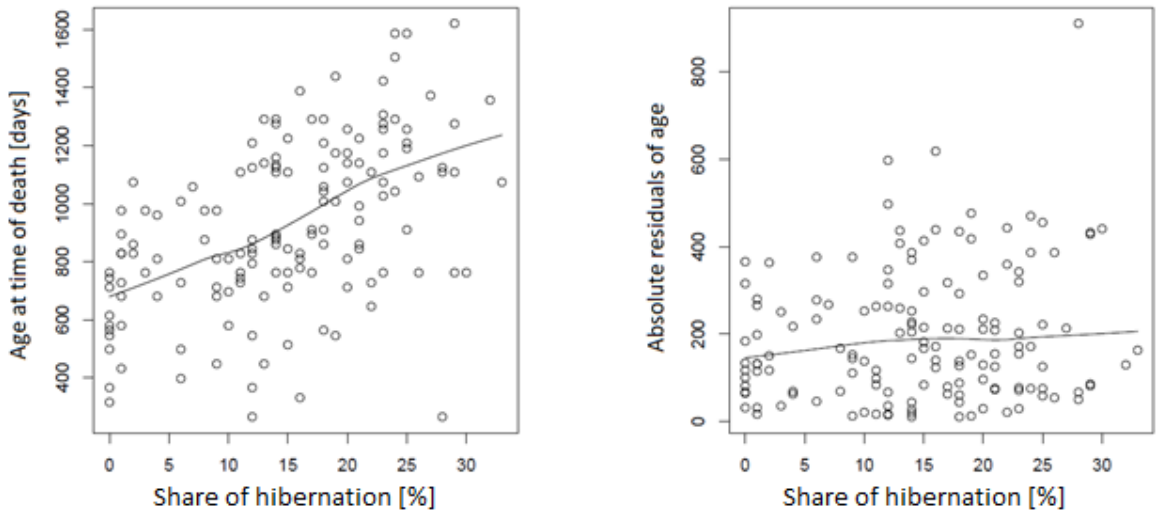


Figure 6.4: LOWESS smoothing of the hamster data from Table 4.2 (left) and spread smoothing of the same data (right), which shows a slight increase in the scatter of the residuals with increasing x-values

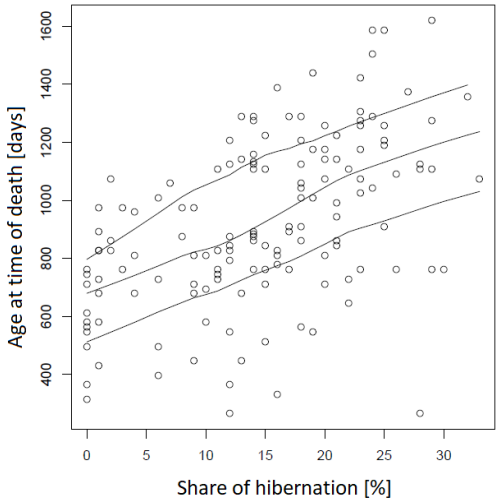


Figure 6.5: Upper and Lower Smoothing of the Hamster Data.

6.3.2 Pairs of Middle Smoothing

After it does not matter whether LOWESS on data (x_i , y_i) or on (y_i , x_i), for i = 1 , . . . , n is used, both variants could be tried. In Figure 6.6 this is done for 4-dimensional data, always comparing pairs. This representation is introduced in the following chapter as a scatter plot matrix or Draftman's display. The curves (x_i, $\hat{y}_i$) and ($\hat{x}_i$,

y_i) are obtained for each pair. The data are measurements of the air quality in New York and can be found as *environmental* in the R-package *library(lattice).*

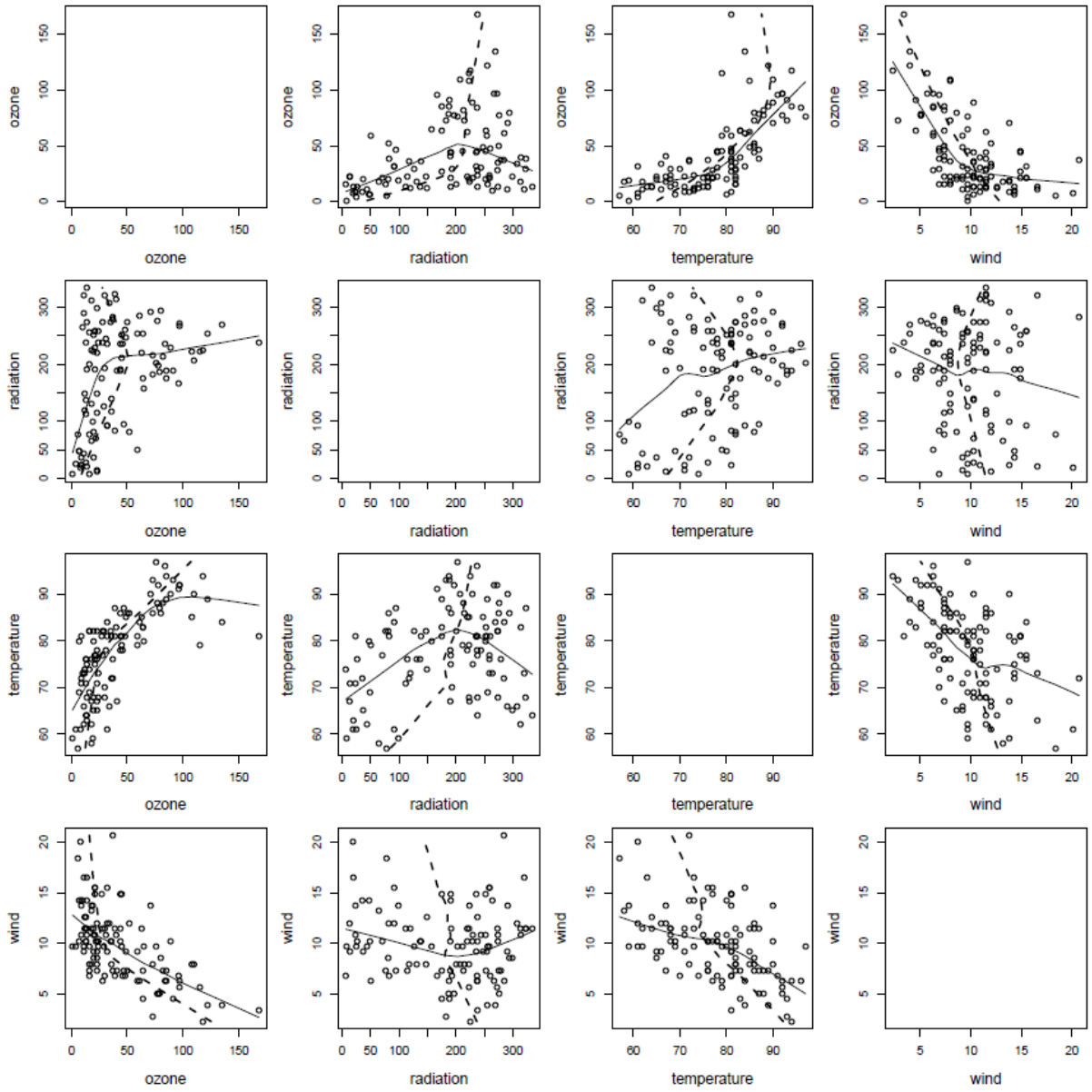


Figure 6.6: Pairs of Middle Smoothing of the environmental data from the library (lattice). To make it easier to distinguish, the curves $(x_i, \hat{y}_i)$ have been drawn with solid lines and the curves $(\hat{x}_i, y_i)$ with dashed lines.

Chapter 7

Time Series Analysis - An Introduction

The aim of this chapter is to get an overview of the procedures with time series, and to impart knowledge of how basic analyzes are carried out in practice. The name time series analysis already says what you are getting at, namely not only displaying time series visually but also analyzing them. Important questions in this context are:

- Are there trends or seasonal fluctuations? If so, can this be quantified?
- What is the structure of the residuals that remain after deducting the trend and the seasonal fluctuation?
- Does the time series follow a certain pattern that can be modeled?
- Is it possible to forecast the future?
- Are there structural breaks in the time series or outliers?

Some of these questions are given here an insight.
We are only concerned here with *univariate time series*, i.e. with values x_t, for $t = 1, \ldots, T$.

As an example of such a time series we consider the monthly volume of beer produced in Australia, in the period from January 1956 to August 1995. The data is available on the page http://134.76.173.220/beer.zip. You can read it into R, convert it as a time series object, and display it graphically with:

```
beer <- read.csv2("beer.csv")
beer <- ts(beer[,1],start=1956,freq=12)
plot(beer)
```

Figure 7.1 (above) shows the resulting time series. You can see a clear trend, but also clear seasonal fluctuations. Apart from that, there are always larger local deviations (outliers). If you look closely at the time series in Figure 7.1 (above), you notice that the upward deviations are greater than the downward deviations. This could have disadvantages in the later estimate. As a simple way out, one can work with the logarithmized data shown in Figure 7.1 (below). The local deviations of the log data now seem to be a little more symmetrical. Differences from the logarithmized neighboring values are:

$$\ln(x_t) - \ln(x_{t-1}) = \ln\left(\frac{x_t}{x_{t-1}}\right) \approx \frac{x_t - x_{t-1}}{x_{t-1}}$$

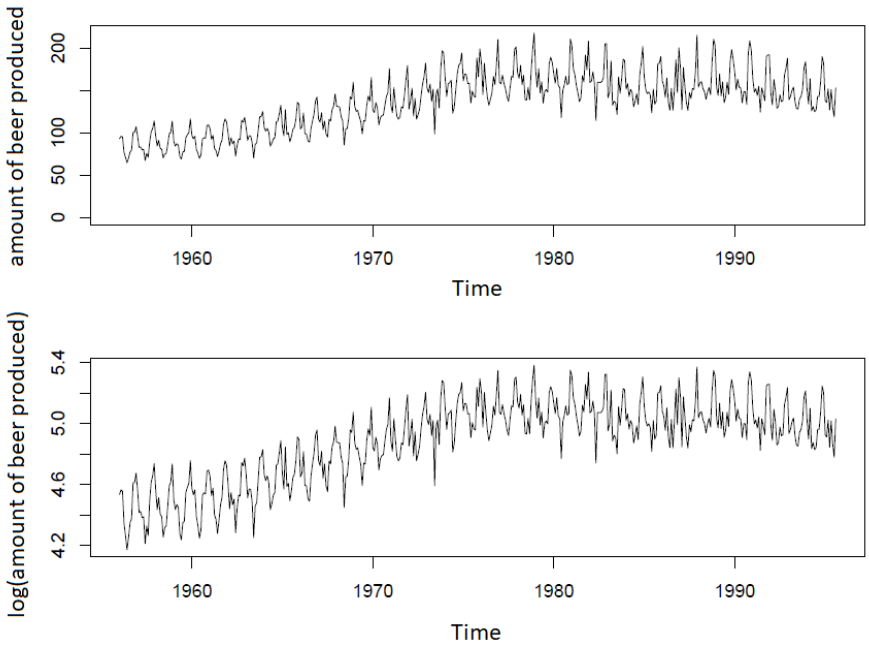


Figure 7.1: Original (above) and logarithmized (below) time series of the beer data.

The latter relationship holds because the log differences between neighbors are generally small numbers. With the logarithmic values one is only interested in relative differences and not in absolute ones. A log representation is also often made for financial time series; the technical term in this context is log-returns. We will now work with these log-transformed values in the following.

7.1 Breakdown of the time series into components

The trend of a time series means that the time series tends to increase or decrease over a longer period of time. Seasonality means that certain fluctuations exist, such as seasonal fluctuations. Often both occur together, but one would still like to appreciate each component individually. So we want to break down the time series into the following components:

$$x_t = \tau_t + \delta_t + e_t \quad \text{für } t = 1, \ldots, T,$$

with the *trend component* τ_t, the *seasonal component* δ_t, and the *residual component* e_t. We also know that the time series has a periodicity of P. For the beer time series, P = 12 because monthly values are available. This results in C = ⌊T/P⌋ cycles. In Chapter 6, methods were already dealt with which a signal can be smoothed or with which non-linear trends were estimated. The R function *stl* uses such methods, especially the function *loess*, see Section 6.3. *stl* works according to the following iterative scheme:

In the kth iteration, k = 1, 2, …, the estimates of the components $\tau_t^{(k)}$ and $\delta_t^{(k)}$ of iteration (k + 1) have the form:

(1) Trend adjustment (detrending): $x_t - \tau_t^{(k)}$

(2) loess applied to (1) for the time points $t_i = t + i * P$, with $i = 0, \ldots, C$, in sequence for all $t \in \{1, \ldots, P\}$. E.g. all January values are smoothed, then all February values, etc.

(3) Applying a (linear) filter to the values of (2), see Chapter 6, results in $\delta_t^{(k+1)}$
(4) Seasonal adjustment: $x_t - \delta_t^{(k+1)}$
(5) loess applied to (4) gives $\tau_t^{(k+1)}$

Similar to the LOWESS algorithm, see Section 6.3, weights are calculated from the residual component $e_t^{(k+1)} = x_t - \tau_t^{(k+1)} - \delta_t^{(k+1)}$ (residuals), which are intended to weigh down outliers in the iterations of the above procedure.

Figure 7.2 shows the result of `stl` applied to the logarithmized beer time series, generated with:

```
Plot(stl(log(beer),s.window="periodic"))
```

If the remainder has no structure, this would be called white noise. If this is not the case, they still contain important information to be modeled. In the results in Figure 7.2, the residuals contain not a conspicuous structure, but "peaks" of different heights, which can also be of interest.

7.2 Regression models for time series

An important goal when modeling time series is the forecast for future values. You could now try to use regression to forecast the future. We'll use a few simple models for this purpose here.

7.2.1 Linear model

In chapter 5 methods for the (robust) estimation of linear trends were presented. For logarithmic time series the linear model has the form:

$$\ln(x_t) = \beta_0 + \beta_1 t + e_t$$

with the coefficients β_0 and β_1, and the error term e_t. The coefficients could be estimated using the methods from Chapter 5. In the following we use LTS regression. For the beer data we would proceed as follows:

```
t <- (1:length(beer))/12+1956   # Zeitachse entsprechend erzeugen
logbeer <- log(beer)            # modelliere logarithmierte Werte
library(rrcov)                  # fuer LTS-Regression
plot(logbeer)
res <- ltsReg(logbeer~t)        # LTS-Regression mit linearem Modell
lines(t,res$fit)
```

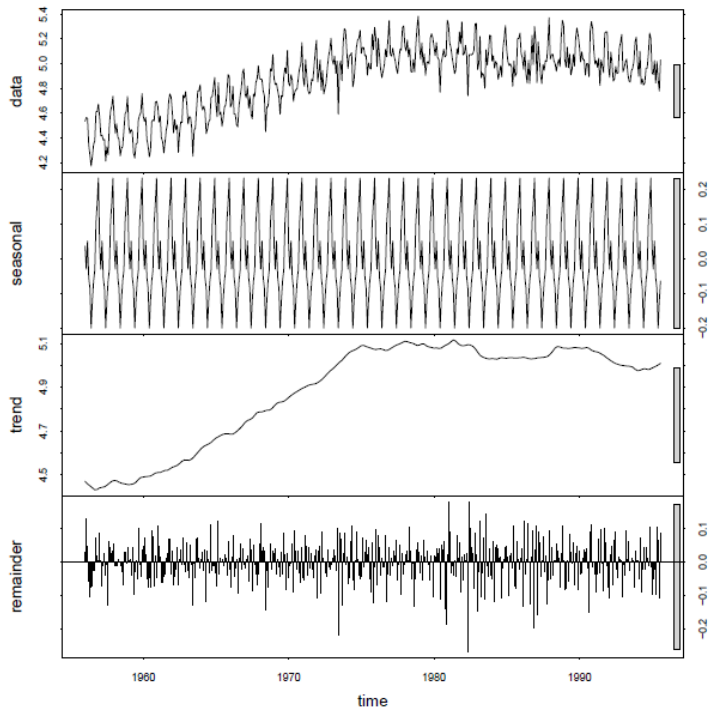


Figure 7.2: Breakdown of the time series into the individual components. (stl)

Figure 7.3 (left) shows the result. The linear model is, of course, too simple and we should move on to a more complex one.

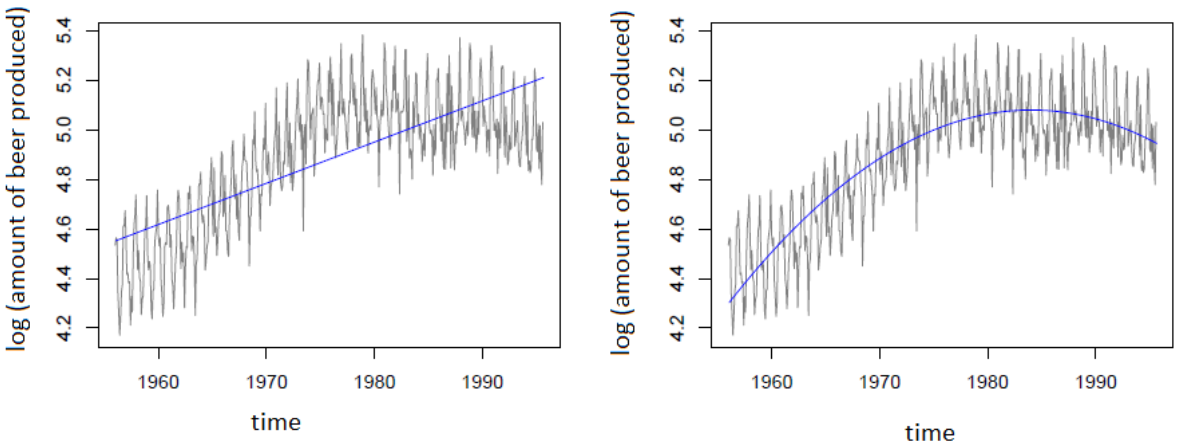


Figure 7.3: Linear model (left) and quadratic model (right) for the beer time series.

7.2.2 Regression with a quadratic term

The above linear model can be easily extended with a quadratic term:

$$\ln(x_t) = \beta_0 + \beta_1 t + \beta_2 t^2 + e_t$$

For our example it looks like this:

```
t2 <- t^2                          # quadratischer Term
plot(logbeer)
res <- ltsReg(logbeer~t+t2)        # LTS-Regression fuer neues Modell
lines(t,res$fit)
```

The result is shown in Figure 7.3 (right). This gives you a (smooth) estimate of the trend. Future values could now be forecast very easily. If $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\beta}_2$ are the estimated regression parameters, so is

$$\hat{x}_t = \exp\left(\hat{\beta}_0 + \hat{\beta}_1 t + \hat{\beta}_2 t^2\right)$$

the estimate for future points in time t = T + 1 , . . . , N. However, since the model is relatively simple, the prognosis will only make limited sense and, at best, only be applicable for the near future.

7.2.3 Regression with Fourier coefficients

A time signal f(t) with period length P could also be represented with (finite) Fourier series:

$$f(t) = a_0 + \sum_{j=1}^{J}\Big(a_j \cos(\omega_j t) + b_j \sin(\omega_j t)\Big), \qquad \text{mit } \omega_j = \frac{2\pi j}{P}.$$

The summands in the series correspond to the frequencies with increasing oscillation, starting with the fundamental oscillation with period length P. If we added the first summand to our model, the seasonality could be modeled as well. So we have the model:

$$\ln(x_t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 \cos\left(\frac{2\pi}{P}\cdot t\right) + \beta_4 \sin\left(\frac{2\pi}{P}\cdot t\right) + e_t.$$

For our example it looks like this (attention, we have already divided t by P = 12):

```
cos.t <- cos(2*pi*t)
sin.t <- sin(2*pi*t)
plot(logbeer)
res <- ltsReg(lbeer~t+t2+cos.t+sin.t) # LTS-Regression fuer neues Modell
lines(t,res$fit)
```

Figure 7.4 shows the result, which of course is still not predicting "perfectly". One could also include higher order terms to improve the fit.

R also provides an inference table for the model with `summary(res)`:

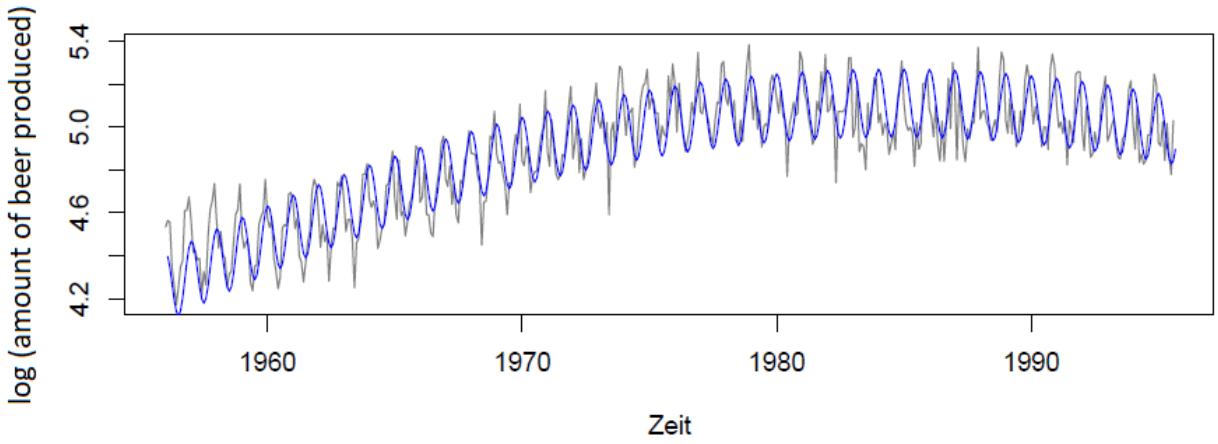


Figure 7.4: Model with Fourier representation for the beer data.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
Intercept -4.161e+03  1.411e+02 -29.480   <2e-16 ***
t          4.198e+00  1.429e-01  29.387   <2e-16 ***
t2        -1.058e-03  3.615e-05 -29.260   <2e-16 ***
cos.t      1.582e-01  6.014e-03  26.308   <2e-16 ***
sin.t      7.136e-03  5.966e-03   1.196    0.232
```

Accordingly, all coefficients except 4 are significantly different from 0. However, one would still include this term for a Fourier representation, because cosine and sine should only appear as a pair.

7.3 Exponential smoothing

The forecast of future values could be made in such a way that the observed values are weighted according to their relevance. If the time series x_t for t = 1,…, T is observed, the smoothed values are obtained by $\tilde{x}_t$:

$$\tilde{x}_t = \alpha x_t + (1-\alpha)\tilde{x}_{t-1},$$

with the smoothing factor $0 < \alpha < 1$. The smaller α is taken, the less the most recent values are taken into account, and it follows that the sequence of $\tilde{x}_t$ becomes smoother. The starting value $\tilde{x}_m$ can be chosen as the arithmetic mean of the first *m* values of x_t. α can practically be determined in such a way that the sum of the squared residuals (or a more robust criterion) is minimized for the available data:

$$\sum_{t=m}^{T} (x_t - \tilde{x}_t)^2 \longrightarrow \min$$

It's easy to see that this type of weighting is equivalent to a recursive calculation:

$$\tilde{x}_t = \alpha x_t + \alpha(1-\alpha)x_{t-1} + \alpha(1-\alpha)^2 x_{t-2} + \alpha(1-\alpha)^3 x_{t-3} + \ldots$$

The weights for past values drop *exponentially*, which also justifies the name of the method.

If one now wants to estimate *h* time steps in advance at time *t*, denoted by $\hat{x}_{t+h|t}$, then one can use the current smoothed value for this, i.e.

$$\hat{x}_{t+h|t} = \tilde{x}_t.$$

The prognosis of all future values does not depend on the horizon *h*, up to which one would like to forecast. This is useful for *stationary time series* that show no trend and no seasonal dependencies and that always return relatively quickly to their mean value, but not for time series with a trend. **When smoothing according to Holt-Winters**, a trend variable b_t is taken into account:

$$\tilde{x}_t = \alpha x_t + (1-\alpha)(\tilde{x}_{t-1} + b_{t-1})$$
$$b_t = \beta(\tilde{x}_t - \tilde{x}_{t-1}) + (1-\beta)b_{t-1}$$

At each point in time *t*, b_t records the local increase in the time series. α and β are again in (0, 1) and regulate the degree of smoothing. The prognosis with Holt-Winters is then by *h* time steps forward

$$\hat{x}_{t+h|t} = \tilde{x}_t + hb_t.$$

The prognosis is thus determined according to a straight line equation with the most recent increase b_t, starting from $\tilde{x}_t$.

A seasonal component can also be modeled in a similar way.

For the beer time series, the Holt-Winters method in R is used as follows:

```
plot(beer)
beer.hw <- HoltWinters(beer)          # Holt-Winters
lines(beer.hw$fitted[,1])             # geglaettete Linie
```

Both a trend and a seasonal component are taken into account. The result of the smoothing is shown in Figure 7.5 (above). The estimated parameters are:

```
Smoothing parameters:
 alpha:  0.07532444
 beta :  0.07434971                # Parameter fuer den Trend
 gamma:  0.143887                  # Parameter fuer die Saison
```

With this model you can now forecast the future, here for the next 48 months:

```
plot(beer,xlim=c(1956,1999))
lines(predict(beer.hw,n.ahead=48))
```

Figure 7.5 (below) shows the result.

7.4 Modeling of time series

7.4.1 Parameters

Time series models take into account dependencies that are given when the time series is systematically shifted by *k* steps. So we consider the values x_t and the values x_{t-k}, where *t* varies. One speaks in this context of a *lag k*. Dependencies are analyzed with the **autocovariance**. The autocovariance of the *order k* is defined as $Cov(x_t, x_{t-k})$ and it can be estimated with:

$$c_k = \frac{1}{T} \sum_{t=k+1}^{T} (x_t - \bar{x})(x_{t-k} - \bar{x}),$$

with the arithmetic mean

$$\bar{x} = \frac{1}{T} \sum_{t=1}^{T} x_t$$

The autocovariance for *lag 0*, i.e. c_0, is the variance of x_t. Thus the **autocorrelation of order** k can be defined as:

$$\rho_k = \mathrm{Corr}(x_t, x_{t-k}) = \mathrm{Cov}(x_t, x_{t-k}) / \mathrm{Var}(x_t)$$

and estimate with $r_k = c_k / c_0$.

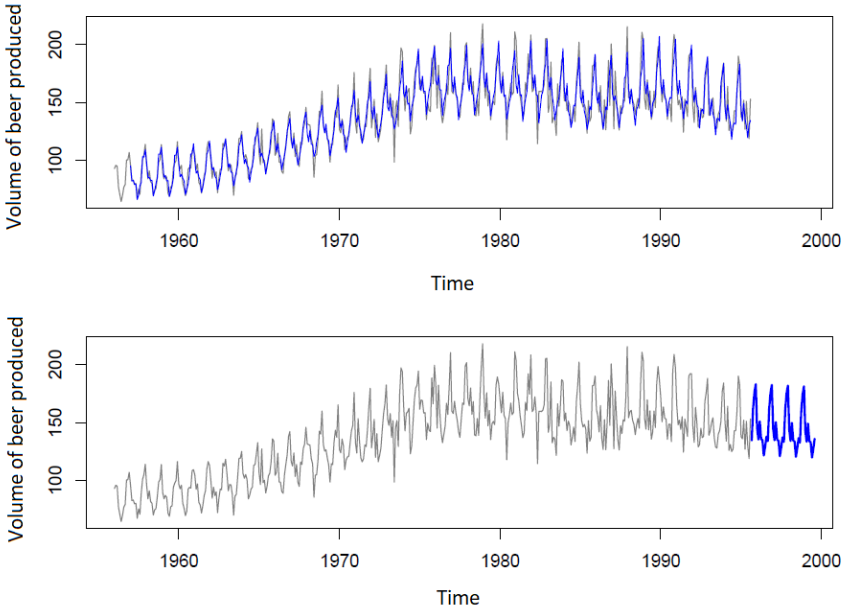


Figure 7.5: Exponential smoothing according to Holt Winters (above) and prognosis (below).

We spoke of *stationary time series* above. These are characterized by the fact that they have the same expected value (mean) and the same variance for all *t*, and that their autocovariance is the same for all *t* and every *k > 0*. Stationary time series with zero autocorrelation for *k > 0* are called *white noise*.

You can test for the property *white noise* with the Q-statistic, also called **Ljung-Box statistic**. The hypothesis $H_0: \rho_1 = \rho_2 = \ldots = \rho_{kmax} = 0$ is tested with a fixed value *kmax.*

Figure 7.6 shows the water level (annual mean) of Lake Huron in the period 1875-1972 (in feet). Figure 7.7 (left) shows the autocorrelations for several values of *k*. This representation is also called a **correlogram**. The dashed horizontal lines correspond to the limits for significance and they are determined by uncorrelated time series.

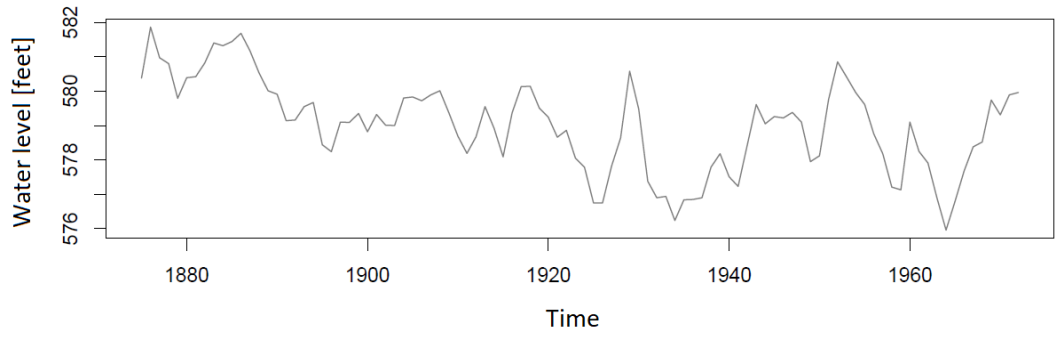


Figure 7.6: Annual water level from Lake Huron

In the case of uncorrelated and stationary time series, all autocorrelations for k > 0 had to be within these limits. In this example, you can see pronounced correlations that only subside after a large *lag*.

Correlations between neighboring points in time can be transferred, i.e. from x_t to x_{t-1}, from x_{t-1} to x_{t-2}, etc. The resulting correlation between x_t and x_{t-k} is therefore influenced by the observations in between. In order to take this effect out of the equation, another gutter for assessing the dependency is defined, namely the **partial autocorrelation** of order k:

$$\mathrm{Corr}(x_t, x_{t-k} | x_{t-1}, \ldots, x_{t-k+1}) \quad \text{für} \quad k = 0, 1, 2, \ldots$$

This corresponds to the autocorrelation between the residuals of a regression from x_t to $x_{t-1}, \ldots, x_{t-k+1}$ and the residuals of a regression of x_{t-k} on the same variables $x_{t-1}, \ldots, x_{t-k+1}$. Figure 7.7 (right) shows the partial autocorrelations for the data from Lake Huron. So up to 2 you still get pronounced partial correlations. This representation is called a partial correlogram.

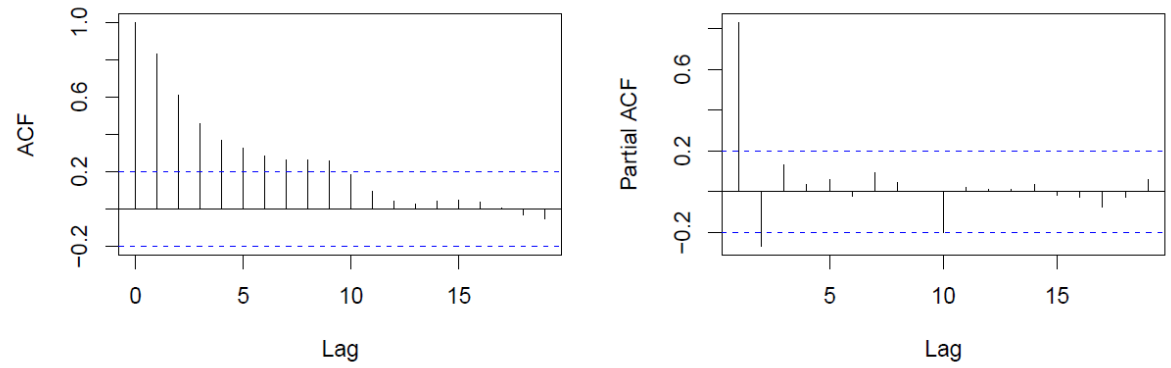


Figure 7.7: Correlogram (left) and partial correlogram (right) of the water level data from Lake Huron.

7.4.2 Basic time series models

Moving Average (MA) model

A *stationary* time series follows a *moving average* process of order 1, or MA (1) for short, if

$$x_t = a + u_t - \theta u_{t-1}$$

with the unknown parameters a and θ. Here u_t stands for white noise. Similarly, *moving averages* of order q, i.e. MA (q), can be defined as

$$x_t = a + u_t - \theta_1 u_{t-1} - \theta_2 u_{t-2} - \ldots - \theta_q u_{t-q}$$

with the unknown parameters a and $\theta_1, \ldots, \theta_q$.

The autocorrelations of MA (q) are 0 for *lags* greater than q. If the correlations in the correlogram drop sharply and after lag q are no longer significant, this is an indication of an MA (q) process. An example of MA (2) can be seen in Figure 7.8 (above).

Autoregressive (AR) model

A *stationary* time series follows an *autoregressive* process of order 1, AR (1) for short, if

$$x_t = a + \phi x_{t-1} + u_t$$

with the unknown parameters a and ϕ. Analogously, an autoregressive process of order p, i.e. AR (p), can be defined as

$$x_t = a + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + u_t$$

with the unknown parameters a and $\phi_1, \ldots, \phi_p$. The partial autocorrelations of AR (p) are 0 for *lags* greater than p. The autocorrelations approach 0 more slowly, and sometimes they have a sinusoidal structure. An example of AR (2) is shown in Figure 7.8 (below).

ARMA (Autoregressive Moving Average) model

If neither the correlogram nor the partial correlogram shows an "implosion" to 0 from a certain *lag*, then a mixture of AR (p) and MA (q) can make sense. This is called ARMA (p; q) and is defined as

$$x_t = a + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + u_t - \theta_1 u_{t-1} - \theta_2 u_{t-2} - \ldots - \theta_q u_{t-q}$$

Preference should be given to simple models, i.e. *p* and *q* should be chosen small.

ARIMA (Autoregressive Integrated Moving Average) model

The ARMA model assumes a *stationary* time series. If there is a trend, one speaks of a non-stationary or *integrated*, i.e. a trend-affected time series (seasonally included). Trends can be eliminated by forming *differences*. To do this, we define the "difference operator" Δ as:

$$\Delta x_t = x_t - x_{t-1}$$

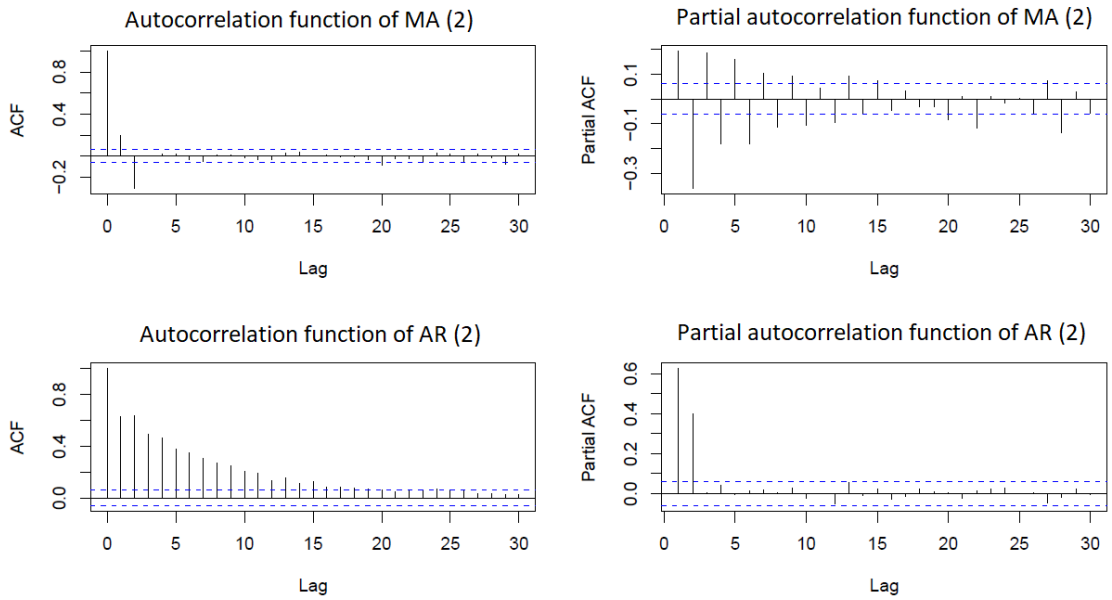


Figure 7.8: Typical structure of MA (2) or AR (2) processes

Linear trends can be eliminated by applying Δ once, quadratic trends can be eliminated by applying Δ twice, that is:

$$\Delta^2 x_t := \Delta(\Delta x_t) = \Delta(x_t - x_{t-1}) = x_t - 2x_{t-1} + x_{t-2}$$

Definition: a time series $x_t,\ t = 1, \dots, T$ follows an $ARIMA(p, d, q)$ model if $\Delta^d x_t$ follows an $ARMA(p, q)$ model.

Because of the preferred simplicity of models, d is usually chosen as 0 or 1.

Example: The model ARIMA (1,1,0) has the shape:

$$\Delta x_t = a + \phi \Delta x_{t-1} + u_t$$

So:

$$x_t = x_{t-1} + a + \phi(x_{t-1} - x_{t-2}) + u_t$$

7.4.3 Estimation of the parameters

The parameter estimation for the models described above is based on the principle of least squares, i.e. by minimizing $\Sigma_{t=1}^{T}(x_t - \hat{x}_t)^2$. However, we do not want to go into the technical details of this estimate here. In R the parameters can be estimated with:

```
arima(data,order=c(p,d,q))
```

for the data *data*, with the corresponding orders of the *ARIMA (p, d, q)* model. For the data from Lake Huron, we try the following model based on Figure 7.7:

```
data(LakeHuron)
fit <- arima(LakeHuron,order=c(1,0,1))
```

So this is an *ARMA (1,1)* model of the shape:

$$x_t = a + \phi x_{t-1} + u_t - \theta u_{t-1}$$

The estimates of the parameters ϕ, θ, and a are:

```
> fit$coef
        ar1         ma1   intercept
  0.7448993   0.3205891 579.0554556
```

7.4.4 Diagnosis of time series models

A very important step in the search for a model is diagnostics. The ARMA (1,1) model for the data from Lake Huron gives estimates of the coefficients, but we do not know whether the model is suitable at all. Diagnostics can be made with:

```
tsdiag(fit)
```

The result is shown in Figure 7.9. At the top you can see the (standardized) residuals, which should be *white noise* in a correctly specified model, in the middle the correlogram of the residuals, where no structure should be visible, and at the bottom the Ljung-box statistics up to lag 10, where all p-values should be beyond the limit 0.05, i.e. not significant. So our model seems to be well specified.

7.4.5 Prediction

Now that we have a reasonably specified model, we can move on to the most interesting point, namely the prediction. However, a sensibly specified model does not necessarily have to be a good prediction model. The fit result also provides information about the model properties, such as the variance of the estimates. This is used to obtain a concentration interval in addition to the forecast.

```
plot(LakeHuron,xlim=c(1875,1980))
LH.pred <- predict(fit,n.ahead=8)  # Prognose fuer die nachsten 8 Jahre
lines(LH.pred$pred,col="blue")
lines(LH.pred$pred+2*LH.pred$se,col="blue",lty=2)
lines(LH.pred$pred-2*LH.pred$se,col="blue",lty=2)
```

Figure 7.10 shows the prediction for the next 8 years, together with a 95% confidence interval. The size of this interval clearly shows that our model - although it makes sense in terms of content - is not very valuable as a prediction model.

You can also go on the internet:

http://www.glerl.noaa.gov/data/now/wlevels/lowlevels/plot/data/Michigan-Huron-1860-.csv

to find historical and current data on Lake Huron water levels. Figure 7.11 shows this data as a red line, in addition to the information from Figure 7.10. The data from R do not exactly agree with the data from the Internet, which may be due to the fact that measurements were taken at another point in the lake. Here the forecast has now been continued up to 2011, just to have a comparison with the current data. The confidence interval spans practically the entire data range, and the prognosis quickly settles towards the mean value. The prognosis can only be used for a very short time, if at all. Our model is generally either too simple, or no better model can be found.

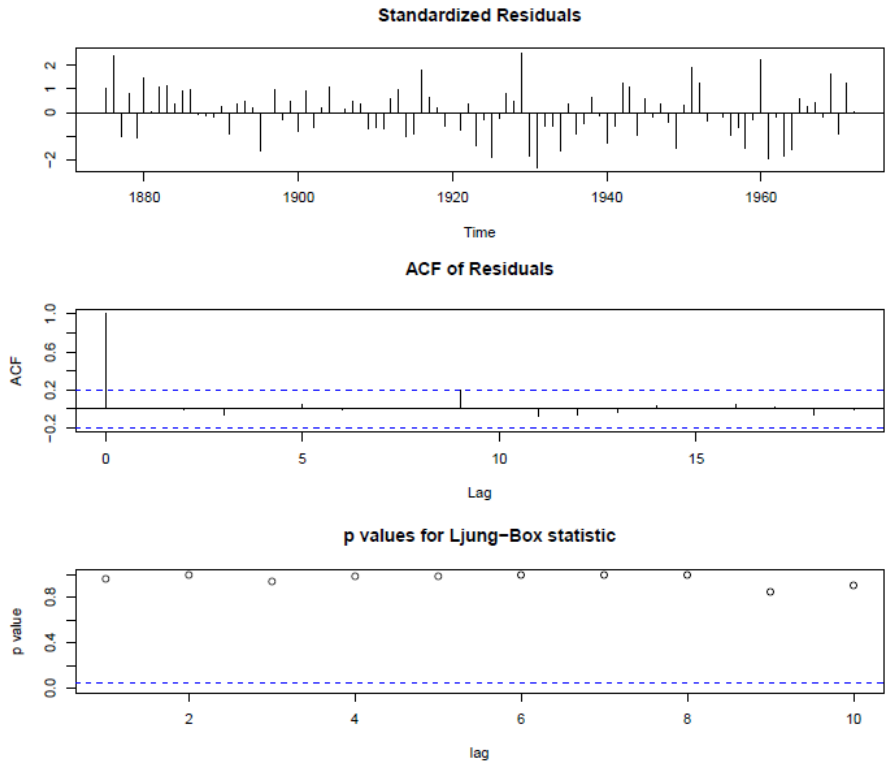


Figure 7.9: Diagnostics for the ARMA (1,1) model of the data from Lake Huron.

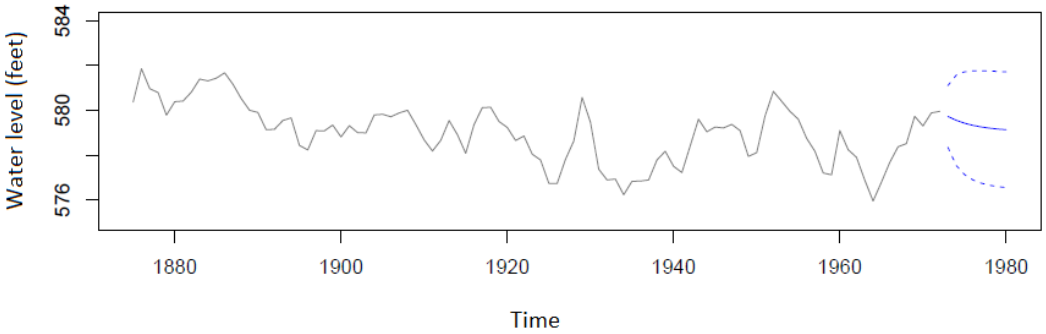


Figure 7.10: Forecast of the water level of Lake Huron for the next 8 years, including the confidence interval.

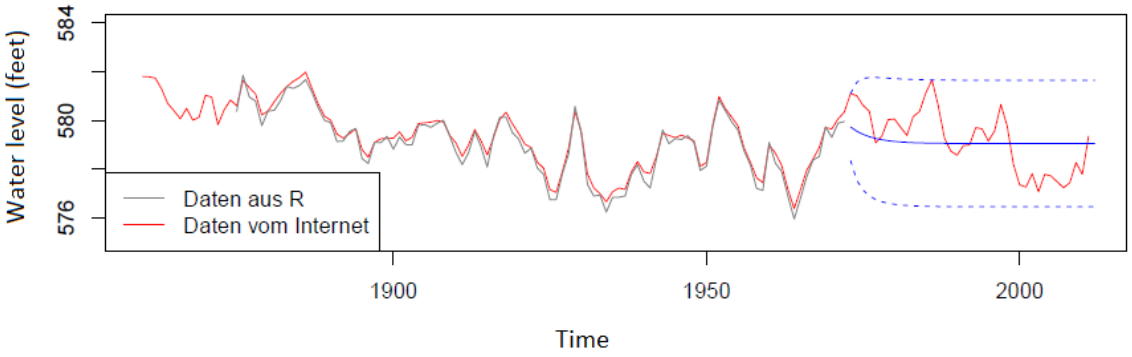


Figure 7.11: Comparison with current data from Lake Huron from the Internet.

Chapter 8
Multivariate graphics

Multivariate data is usually presented in a rectangular table (matrix) consisting of n rows and p columns, with each cell containing a numerical value. Each row contains information about an *object*, and each column contains information about a variable. The matrix is referred to as $\boldsymbol{X}$ in the following. The rows or samples are noted with $x_1, \ldots, x_n$. The i-th sample is thus $x_i = (x_{i1}, \ldots, x_{ip})^T$ (column vector!).

8.1 Scatterplots

In scatter diagrams of low-dimensional data, the data must also be represented by **varying the symbol** (marker) by which the data values are represented. The 4-dimensional iris data are given as an example in Figure 8.1. The first two dimensions *Sepal Length* and *Sepal Width* form the horizontal and vertical axes in the plot. The variable *petal length* is represented by the size of the symbols. The variable *Petal Width* is represented in the plot by different shades of gray.

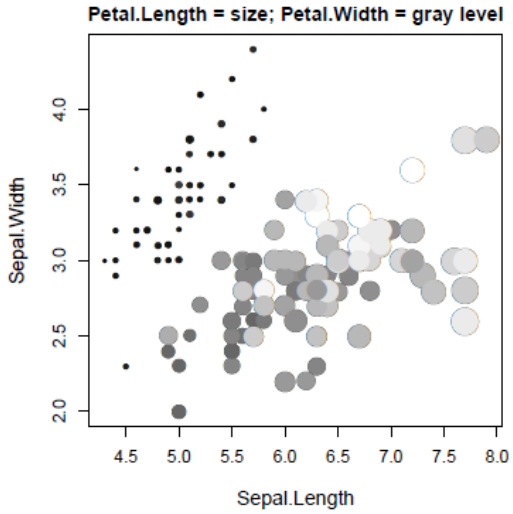


Figure 8.1: Iris data: 4 dimensions are represented by symbol sizes and gray levels.

Another possibility is to display all $\binom{p}{2}$ 2-dimensional views of the p variables (**Draftsman's display**), as was done with the 4-dimensional iris data in Figure 8.2. In addition, the information about the group membership of the 3 types of irises was visualized by different selection of gray levels.

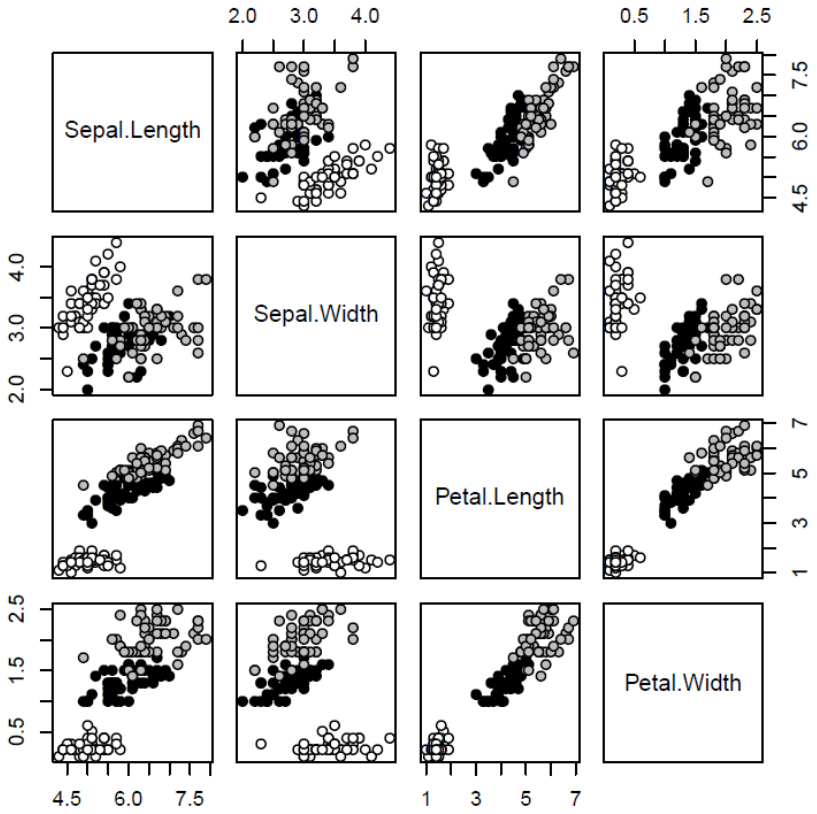


Figure 8.2: Iris data: Draftman's display with additional group information

This representation of all pairs of 2-dimensional views can even be used for data of higher dimensions, provided the data have distinctive structures. However, one must always be aware that one sees "only" 2-dimensional projections of the p-dimensional data.

Of course, multidimensional data could also be displayed in 3-dimensional scatter diagrams (different symbols, colors, etc. had to be selected for the remaining dimensions). However, this only makes sense with a corresponding interactive visualization.

Another variation are the C*asement Displays* - a layered representation of the data, as well as the *Multi-window Plots*, in which one or two of the p variables are used to decompose the sample and the resulting subsets are shown in scatter diagrams.

8.2 Profiles, stars, segments, Chernoff faces

8.2.1 Profiles

Each data value x_i is represented by p bars or lines. The length of the j-th bar (line) of the k-th point is proportional to x_{kj}.

As an example, consider the data in Table 8.1 for Virginia death rates. The "variables" are different population groups (columns), and the "objects" are different age groups (rows). Figure 8.3 shows profiles for the individual objects (left), but also the equivalent representation of the profiles in relation to the individual variables (right).

Table 8.1: Death rates (in%) in Virginia in 1940. The data are broken down into age groups (rows) and population groups (columns).

Age group	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

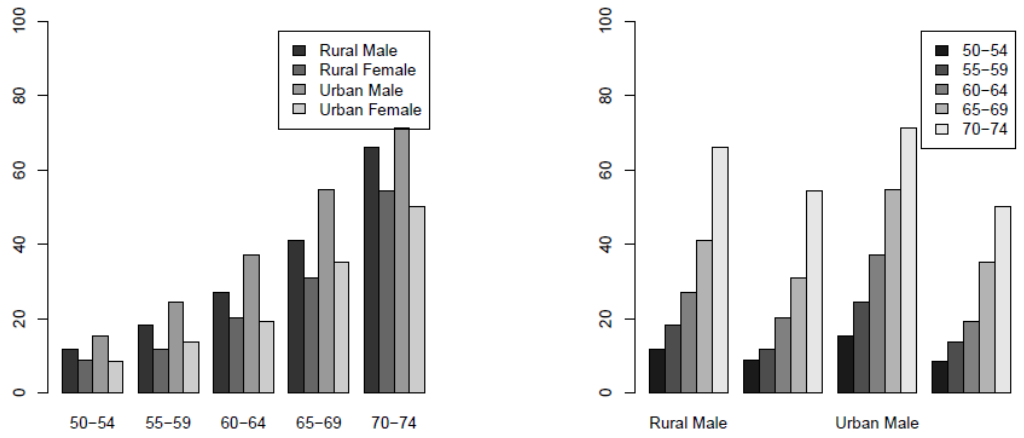


Figure 8.3: Representation of the data from Table 8.1 with profiles for objects (left) and variables (right).

The representation with profiles is suitable for data with a relatively high dimension (around 20), but there is undoubtedly a limitation with regard to the number of objects. Although the profiles can be displayed without a coordinate system, side by side and one below the other, the display is no longer clear even for a three-digit number of objects.

If variables with very different scales are considered, the data should be normalized beforehand, otherwise the clarity can be lost. Many possibilities are conceivable for normalization, e.g. normalization of each variable to

- the interval [0, 1] (or another interval),
- (robust) variance 1,
- Norm 1, i.e. $\Sigma_{i=1}^{n} {x_{ij}}^2 = 1$ for all j.

Alternatively (or additionally) the variables can also be transformed (e.g. with logarithm).

8.2.2 Stars

Depending on the type of variation, they are also called "webs", "polygons" and "circular plots". The data points are "normalized", e.g.

$$\tilde{\boldsymbol{x}}_i := \left(\frac{x_{i1}-\bar{x}_{.1}}{s_1}, \frac{x_{i2}-\bar{x}_{.2}}{s_2}, \ldots, \frac{x_{ip}-\bar{x}_{.p}}{s_p}\right)^T$$

with:

$$\begin{aligned} \bar{x}_{.j} &:= \text{Mean of the n values } x_{kj} \text{ for variable } j = 1, \ldots, p \\ s_j &:= \text{Scatter of the n values } x_{kj} \text{ for variable } j = 1, \ldots, p \end{aligned}$$

and then the $\tilde{x}_{ki}$ plotted radially in the p directions $\frac{2\pi}{p}k \;\; (k = 0, \ldots, p-1)$

As an example we consider the vehicle data from Table 8.2. These various characteristics of 32 vehicles are taken from American motor journals from 1974. (The entire data is available in R under *data(mtcars)*.)

In Figure 8.4 these 7-dimensional data are shown with stars, with each of the 32 vehicles representing a star. Due to the similarity of some stars, the vehicles can be visually divided into groups (clusters).

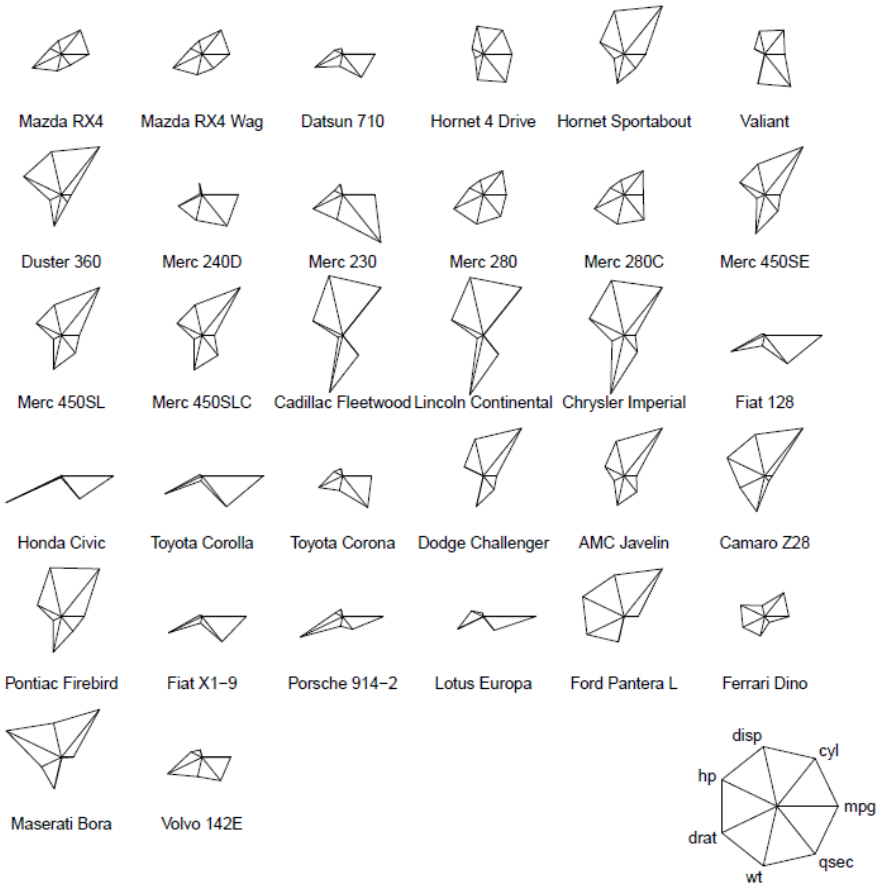


Figure 8.4: Presentation of the vehicle data from Table 8.2 with stars. (*stars*)

Table 8.2: Vehicle data: Various characteristics were collected from 32 vehicles. The data comes from American motor journals from 1974

Car type	Miles per gallon	No. of cylinders	Displace-ment	Horse-power	Rear axle ratio	Weight (lb/1000)	Time for 1/4 mile
Abbreviation	*mpg*	*cyl*	*disp*	*hp*	*drat*	*wt*	*qsec*
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02
Valiant	18.1	6	225.0	105	2.76	3.460	20.22
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60

8.2.3 Segments

The representation with segments is very similar to that with stars. The data are first standardized accordingly. Then the entire angle of 2π is regularly subdivided into p parts. Now the value of each variable of an object is entered in a segment, whereby the area of the circle segment corresponds to the value on the variable.

Figure 8.5 shows the vehicle data above with segments.

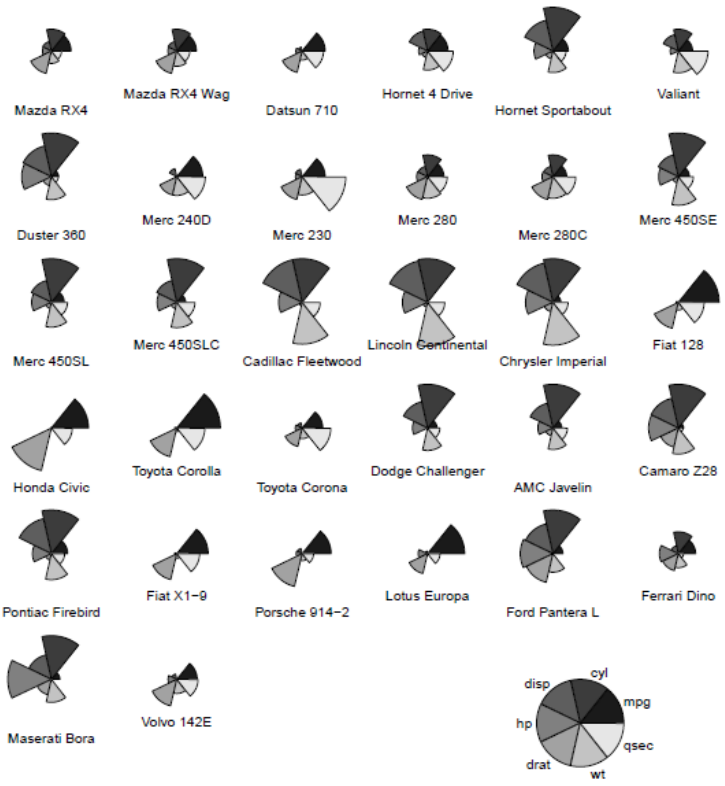


Figure 8.5: Presentation of the vehicle data from Table 8.2 with segments. (stars)

8.2.4 Chernoff Faces

Each variable is represented by the size, direction or curvature of a part of the face. Here it is hardly possible to draw conclusions from the representation of a data point on the values of its components (i.e. on the x_{ij}). In addition, a certain subjectivity is always stored in this representation, because the analyst pays more attention to some parts of the face and some appear less important. Figure 8.6 shows this representation for the vehicle data.

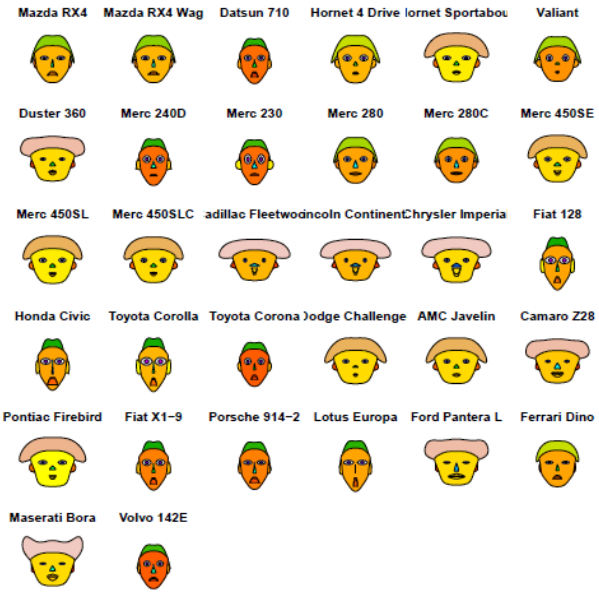


Figure 8.6: Representation of the vehicle data from Table 8.2 with Chernoff faces. (*faces* from *library(aplpack)*)

8.2.5 Boxes

The variables are divided into 3 groups using a cluster analysis. The decisive factor here is the similarity between the variables, which can be determined using the correlation matrix. Then each group of variables is represented as a separate page in a box. The relative proportions of each observation in the variables ultimately determines the size of the box. Figure 8.7 shows the representation with boxes for the above vehicle data.

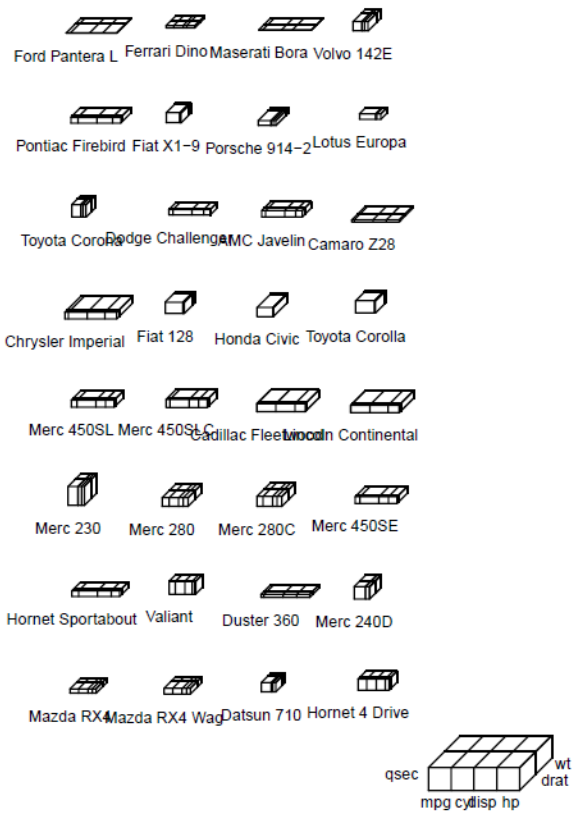


Figure 8.7: Representation of the vehicle data from Table 8.2 with boxes. (*boxes* from *library(StatDA)*)

8.3 Trees

This is a representation of the data in the form of a tree structure. An attempt is made to express the correlation between the variables in the presentation of the data. This is decisive for the sequence of the branches.

The following algorithm describes the construction of a Hartigan tree. It is about the thickness and length of the branches, the angles between the branches and between the trunk and the branches:

1. The thickness of a branch is proportional to the number of branches above the branch (the cluster tree indicates what is above).
2. Angle between 2 branches: A minimum angle and a maximum angle are specified. The range of the angles increases with the number of variables.
3. The branches should be directed in such a way that they do not overlap.
4. Angle:
   (a) The angle of a branch with the vertical is proportional to the thickness of the branch.
   (b) The angle of the trunk with the vertical is inversely proportional to the thickness.
   (c) The sum of the two angles between the branches of a branch and the vertical is given by point 2).
5. The length of a branch is proportional to the mean length of all variables over the branch.

Figure 8.8 shows the representation of the vehicle data using trees.

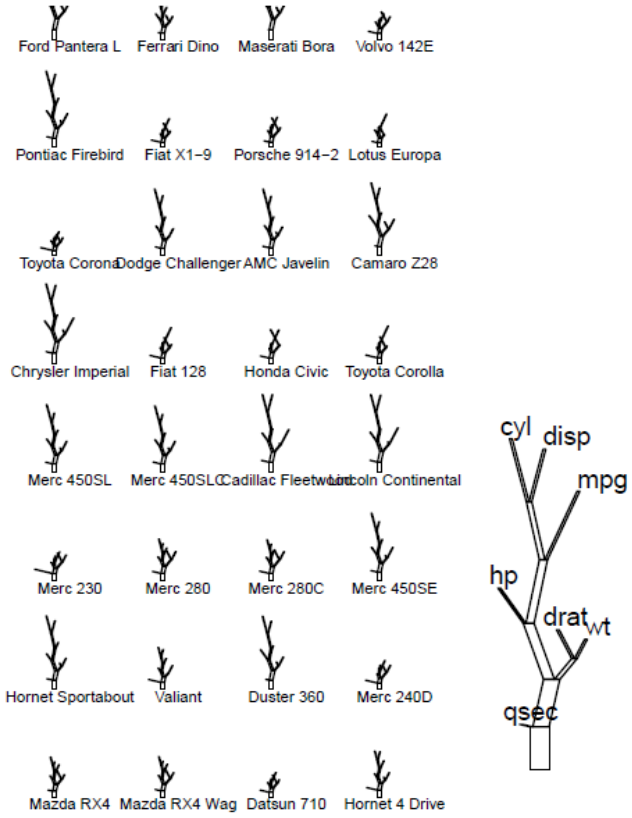


Figure 8.8: Representation of the vehicle data from Table 8.2 with trees. (*tree* from *library(StatDA)*)

8.4 Castles

As an alternative to the trees, the representation form "castles" can also be selected. The principle of the construction is analogous to trees, only that here the angles of the branches are equal to zero. Figure 8.9 shows the representation of the vehicle data using castles.

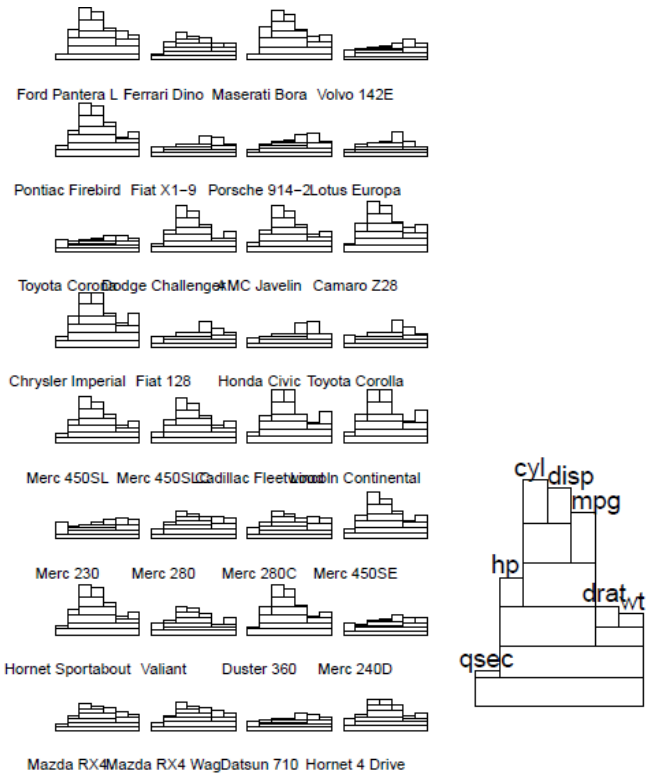


Figure 8.9: Representation of the vehicle data from Table 8.2 with castles. (*tree* from *library(StatDA)*)

8.5 Plot with parallel coordinates

The idea here is that the individual variables are placed side by side as coordinate axes. The values of the individual variables are brought to the same range,

$$x_{ij}^{*} := \frac{x_{ij} - \min_i(x_{ij})}{\max_i(x_{ij}) - \min_i(x_{ij})} \qquad \text{für} \qquad j = 1, \ldots, p.$$

Then objects are plotted in the form of lines according to their coordinates. You can integrate further information in the plot by using different colors or shades of gray, as well as different line thicknesses or line types.

Figure 8.10 shows the iris data with parallel coordinates. The group information is visualized by different colors.

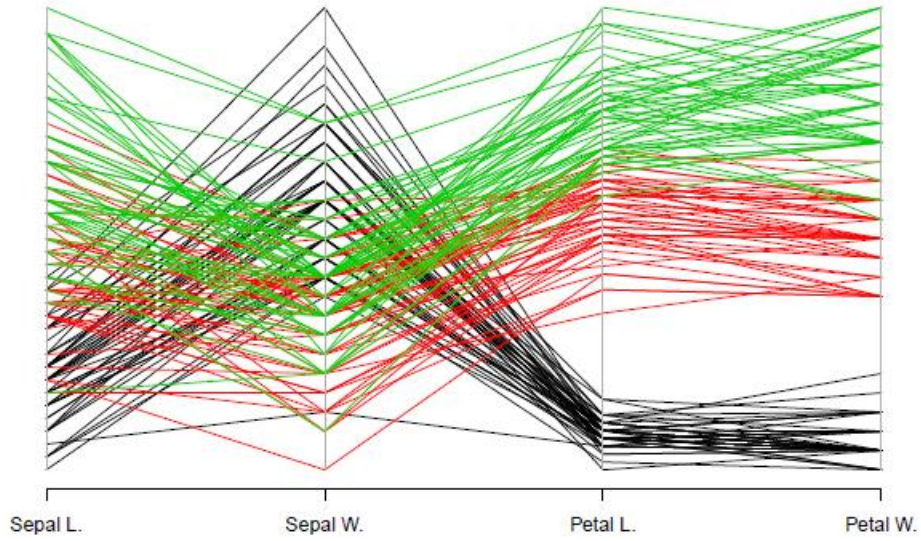


Figure 8.10: Representation of the iris data with parallel coordinates. (parcoord)

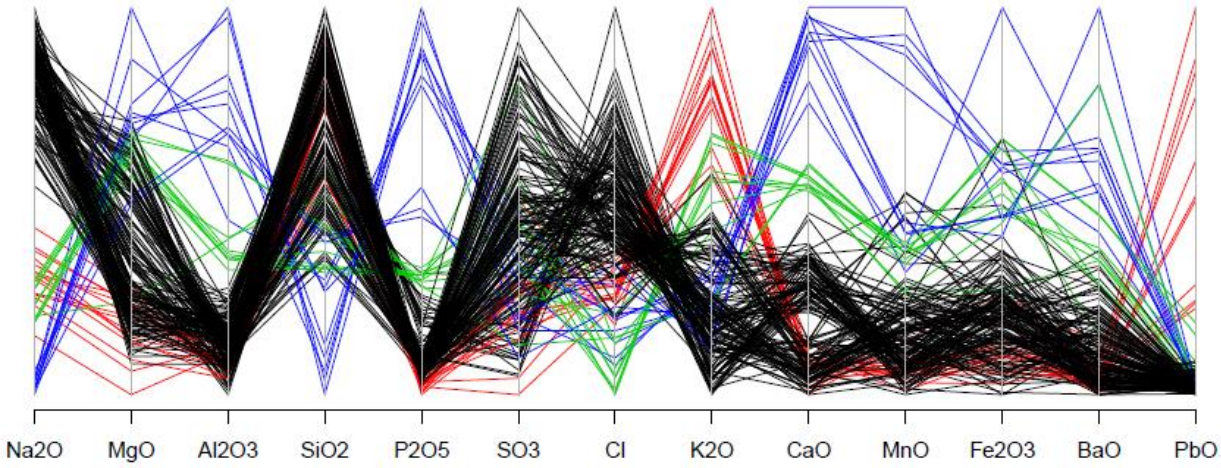


Figure 8.11: Representation of the glass data with parallel coordinates.

As a further example of a plot with parallel coordinates, the data glass from the *library(chemometrics)* is shown in Figure 8.11. Concentrations of various oxides were measured. The colors correspond to different types of glass.

Comments: This type of visualization has the following properties, among others:

- The data quality (rounding effects, extreme outliers) is recognized at a glance. However, categorical variables often lead to a subjective clustering.
- The order of the variables (coordinates) in the representation can be decisive for the clarity.
- This plot "can withstand" a very large number of objects, but many dimensions can also be represented.
- Trends or clusters are relatively easy to recognize, as are values that deviate completely from the data structure (these have a completely different line course).

Chapter 9
Parameter Estimation in the Multidimensional

In Chapter 3, important estimators for univariate data were listed. Here we would like to make an extension to the multivariate case. We assume that measurements of p variables (features) $x_1, \dots, x_p$, and not just a univariate quantity x. Of course, we shall need n observations again later for each of these variables, namely the concrete multivariate data.

9.1 Covariance and Correlation

The covariance and the correlation represent a measure of the relationship between these variables. Usually, only dimensions are given for the linear relationship. The correlation can be interpreted better than the covariance because it is a standardized quantity.

We first assume that $x_1, \dots, x_p$ are random variables, the realizations of which form the $n \times p$ data matrix **X** later. The **covariance** between the pair x_j and x_k $(j, k \in \{1, \dots, p\})$ is defined as:

$$\sigma_{jk} = \mathrm{E}[(x_j - \mathrm{E}(x_j))(x_k - \mathrm{E}(x_k))]$$

The symbol "E" describes the usual expected value, which for a specific sample is usually estimated using the arithmetic mean. For $j = k$ we get $\sigma_{jj} = \sigma_{kk}$, the *variance*.

If one has now given concrete data, i.e. the observations were measured simultaneously on each variable, then there are, in particular, the values $x_{1j}, \dots, x_{nj}$ for x_j and the values $x_{1k}, \dots, x_{nk}$ for x_k. The *classical estimate of the covariance* σ_{jk} is called *sample covariance* and it is defined as:

$$s_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

with the arithmetic means

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \quad \text{and} \quad \bar{x}_k = \frac{1}{n} \sum_{i=1}^{n} x_{ik}$$

If one sets $j = k$ again, one obtains the sample variance.

The correlation coefficient between the random variables x_j and x_k is defined as:

$$\rho_{jk} = \frac{\sigma_{jk}}{\sqrt{\sigma_{jj}\sigma_{kk}}}$$

and is a dimensionless measure for the linear relationship between these two features. This quantity always takes a value in the interval [-1, 1]. With a value of 1 or -1, x_j and x_k contain the same information, with -1 there is an inverse relationship. With a value of 0 there is no linear relationship between x_j and x_k.

The classical estimate of the correlation ρ_{jk} is called the *sample correlation* and it is defined as

$$r_{jk} = \frac{s_{jk}}{\sqrt{s_{jj} s_{kk}}}.$$

This correlation measure measures the *linear relationship* between x_j and x_k. So if r_{jk} = 0, there is no linear relationship between x_j and x_k, but there could very well be a non-linear relationship.

If the covariance or the correlation is determined for all pairs of variables, matrices of order $p \times p$ are obtained. The (theoretical) *covariance matrix* is denoted by Σ, it contains the elements σ_{jk}. If one estimates these elements with the sample covariance s_{jk}, one obtains the $p \times p$ sample covariance matrix **S**. Analogously, if one estimates the correlations ρ_{jk} with the sample correlations r_{jk}, then the matrix **R** is obtained.

```
R:   S <- cov(X)   # Sample Covariance matrix
R:   R <- cor(X)   # Sample Correlation Matrix
```

Figure 9.1 should give a feeling of what a theoretically defined covariance matrix could look like in practice. The theoretical covariance matrices below in the figures result in the theoretical correlations 0.8, 0, and -0.8, and the point cloud shown comes from 200 randomly generated values with this covariance matrix. If the sample correlations were to be determined from the points, the value had to be very close to the theoretical value.

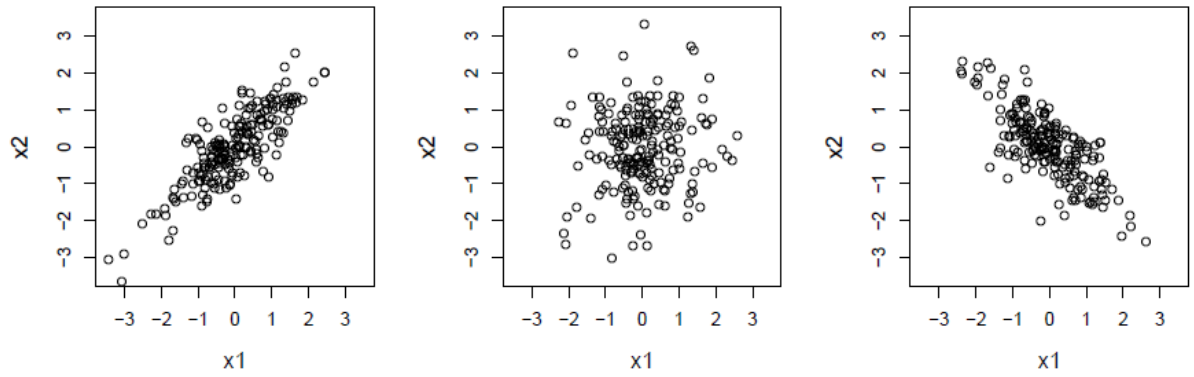


Figure 9.1: Point clouds with realizations of the theoretical covariance matrices, corresponding to the correlations 0.8, 0, and -0.8.

9.1.1 More robust estimates of covariance and correlation

As stated in previous chapters, the arithmetic mean and sample variance are very sensitive to outliers. More robust estimators are, for example, median or MAD. Similarly, the sample covariance is sensitive to outliers because an observation that is (arbitrarily) far away can have a very large influence on the result.

More robust estimates of covariance or correlation can be obtained by giving less weight to observations that deviate from the "bulk" of the common data structure. In the extreme case, the weight could even be 0, which means that the observation is no longer directly entered.

One approach to robustification is the **Spearman rank correlation**, where not the data values themselves, but only the range of the values of the individual variables are taken. The sample correlation is then calculated from these ranges. The rank is always a number from {1, ..., n} (with n observations), and it does not matter how extreme the values of outliers are. Ranges can also map nonlinearities well, so that the resulting measure is a measure not only for the linear but also for the *nonlinear* relationship.

```
R:   cor(X,method="spearman")
```

A more robust variant is obtained, for example, with the **MCD** (*Minimum Covariance Determinant*) **estimator**. As the name suggests, the determinant of the covariance matrix is minimized. However, not all n, but only $h < n$ observations are taken from which the sample covariance matrix is calculated. Their determinant then defines the target criterion, and an algorithm is used to search for those observations that lead to the smallest determinant. For h it is recommended to take about half or 3/4 of the observations. The robust covariance estimate is thus the empirical covariance matrix of these h observations (multiplied by a factor for consistency with normal distribution). A robust location estimate with the arithmetic mean of the h observations is also obtained as a "by-product".

```
R:   library(robustbase)
R:   covMcd(X)
```

The robust covariance estimate using MCD can be used directly to obtain a robust estimate of the correlation matrix. Let c_{jk} be the element (j,k) of the MCD covariance matrix, for $j,k = 1,\dots,p$. The robust variances are then given by c_{jj}. The correlation is defined as covariance through the roots of the variances, i.e. is:

$$\frac{c_{jk}}{\sqrt{c_{jj}}\sqrt{c_{kk}}}$$

the element (j,k) of the robust correlation matrix.

9.2 Distance and Similarity

While the relationships between the variables were in the foreground in the last section, the relationships between the objects will now be examined. We again consider observations in p-dimensional space, in particular the observations $\boldsymbol{x}_A = (x_{A1},\dots,x_{Ap})^T$ and $\boldsymbol{x}_B = (x_{B1},\dots,x_{Bp})^T$.

The **Euclidean distance** between $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ is defined as:

$$d_E(\boldsymbol{x}_A,\boldsymbol{x}_B) = \left(\sum_{j=1}^{p}(x_{Bj}-x_{Aj})^2\right)^{1/2} = [(\boldsymbol{x}_B-\boldsymbol{x}_A)^T(\boldsymbol{x}_B-\boldsymbol{x}_A)]^{1/2} = \|\boldsymbol{x}_B-\boldsymbol{x}_A\|$$

```
R:   dist(X,method="euclidean")
```

The **Manhattan distance** (also *city block* distance), on the other hand, takes the absolute coordinate-wise distances:

$$d_M(\boldsymbol{x}_A, \boldsymbol{x}_B) = \sum_{j=1}^{p} |x_{Bj} - x_{Aj}|$$

```
R:   dist(X,method="manhattan")
```

The above two distance measures are generalized with the **Minkowski distance**, defined as:

$$d_{Mink}(\boldsymbol{x}_A, \boldsymbol{x}_B) = \left(\sum_{j=1}^{p} (x_{Bj} - x_{Aj})^m \right)^{1/m}$$

```
R:   dist(X,method="minkowski",p=m)
```

The **cosine of the angle** α between the object vectors is a measure of similarity. It is independent of the length of the vectors and therefore only takes into account the relative values of the variables:

$$\cos\alpha = \frac{\boldsymbol{x}_A^T \boldsymbol{x}_B}{\sqrt{(\boldsymbol{x}_A^T \boldsymbol{x}_A)(\boldsymbol{x}_B^T \boldsymbol{x}_B)}} = \frac{\boldsymbol{x}_A^T \boldsymbol{x}_B}{\|\boldsymbol{x}_A\| \cdot \|\boldsymbol{x}_B\|}$$

In general, a measure of distance d can easily be converted into a measure of similarity by calculating, for example, $1 - d \,/\, d_{max}$, where d_{max} is the maximum distance between all pairs of objects.

Figure 9.2 shows these different distance measures for observations in two dimensions.

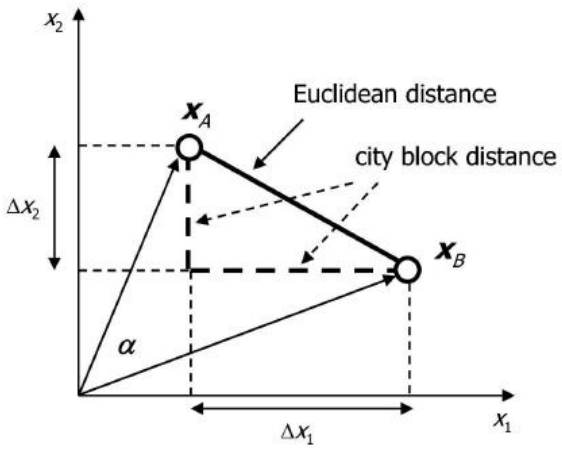


Figure 9.2: Different distance measures, visualized in two dimensions.

The **Mahalanobis distance** is a particularly important statistical measure of distance because it takes the covariance structure into account. It does not depend on the scaling of the variable and is defined as:

$$d_{Mahal}(\boldsymbol{x}_A, \boldsymbol{x}_B) = [(\boldsymbol{x}_B - \boldsymbol{x}_A)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_B - \boldsymbol{x}_A)]^{1/2}$$

If the inverse covariance matrix was set equal to the identity matrix $\boldsymbol{I}$, the Euclidean distance would be obtained. The distance of an observation to the center $\boldsymbol{\mu}$ of the distribution is often of interest here. You then get the formula:

$$d_{Mahal}(\boldsymbol{x}_i, \boldsymbol{\mu}) = [(\boldsymbol{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})]^{1/2} \qquad \text{für } i = 1, \ldots, n.$$

Figure 9.3 (right) shows different levels of the Mahalanobis distance to the center of the distribution represented in the form of ellipses. So in the center there are small distances that grow towards the edge of the point cloud. In comparison, the Euclidean distance is visualized on the left, which results with $\boldsymbol{\Sigma} = \boldsymbol{I}$. Clearly, this distance measure cannot describe how far one can get to the edge of the point cloud - unless one has a spherically symmetrical distribution.

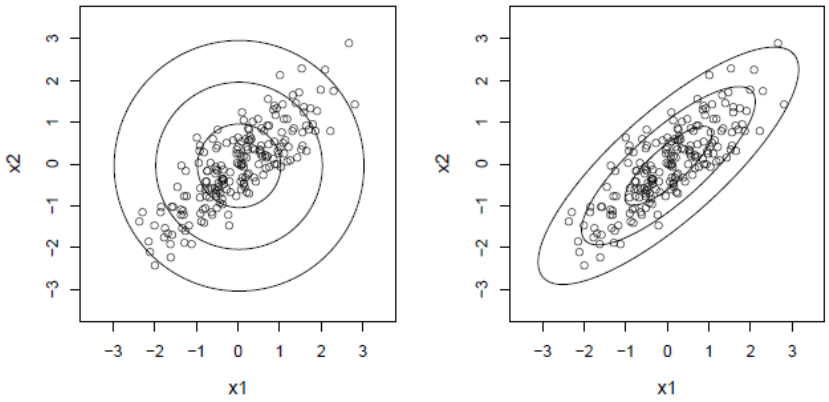


Figure 9.3: Euclidean distance (left) and Mahalanobis distance (right) to the center of the distribution, shown in the form of ellipses.

9.3 Multivariate outlier detection

The concept of Mahalanobis distance can be used well to identify outliers in multivariate data. Outliers can therefore be those observations that are extreme in one dimension (here they were easy to recognize in univariate form) and those that are hidden in different dimensions. Both types should be recognized with the following methodology.

The Mahalanobis distance of an observation to the center of the distribution,

$$d_{Mahal}(\boldsymbol{x}_i, \boldsymbol{\mu}) = [(\boldsymbol{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})]^{1/2} \qquad \text{für } i = 1, \ldots, n,$$

indicates how far the observation is from this center, relative to the underlying covariance structure. Observations with bigger Mahalanobis distance were therefore candidates as outliers. But what does "big" mean? Under certain conditions (e.g. multivariate normal distribution) one can show that the squared Mahalanobis distances follow a chi-square distribution with p degrees of freedom, χ_p^2 . If one

defines a quantile of this distribution as a limit, e.g. $\chi^2_{p;0.975}$, an outlier rule can be created: observations whose squared Mahalanobis distance is greater than this limit are declared as potential outliers.

Since both the center $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ are included in the formula for the Mahalanobis distance, both quantities must be estimated from the data. If you were to use the classical estimators, arithmetic mean and sample covariance, then you would have a bad tool for outlier detection, because these estimators are precisely influenced by the outliers. So, one needs robust estimators for location and covariance. The MCD estimator delivers both.

```
R:   library(robustbase)
R:   plot(covMcd(X))
```

Figure 9.4 shows the difference that results when calculating the Mahalanobis distance when using classical and robust estimators. In order to be able to visualize this well, only 2-dimensional data were used, namely two variables from the data *glass* of the *library(chemometrics)*. The bound for outliers corresponds to $\sqrt{\chi^2_{2;0.975}} = 2.72$, and this can be seen as an ellipse in the plots. The graphic on the left uses the classic estimator, while the graphic on the right uses the MCD. With a robust estimate you get a whole group of outliers that correspond to a certain type of glass (symbol). Figure 9.5 now also shows the obtained values of the Mahalanobis distances for the classical (left) and robust (right) estimation.

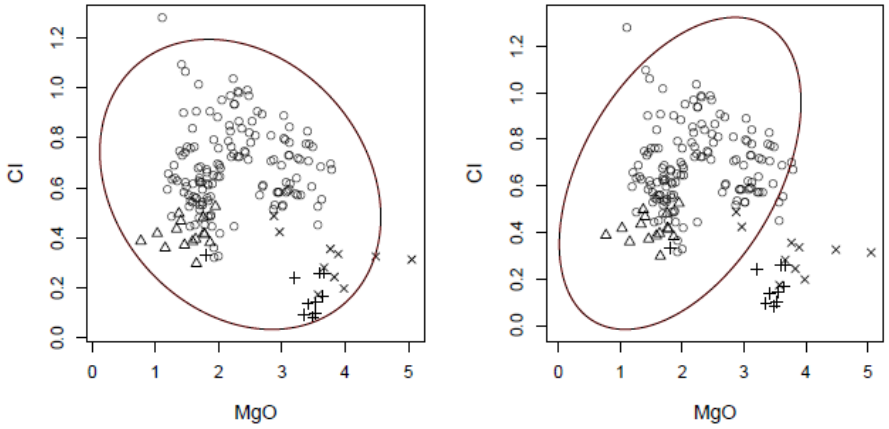


Figure 9.4: Limits for outliers visualized with an ellipse, determined with classical estimators on the left and with robust estimators on the right.

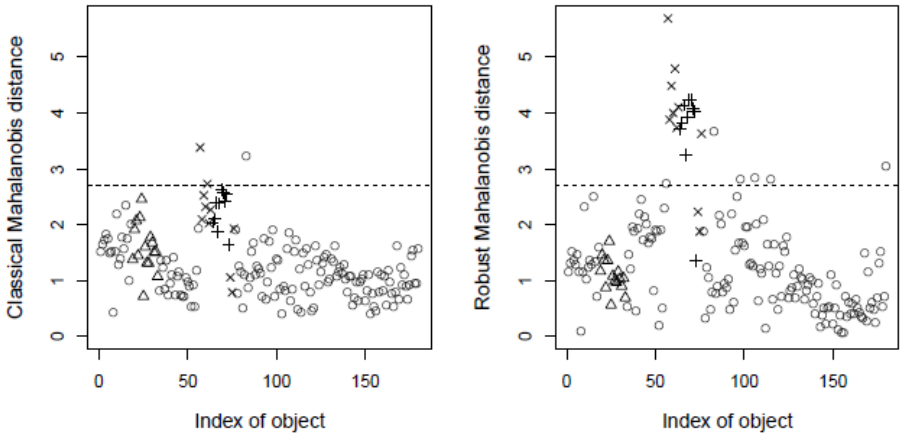


Figure 9.5: Mahalanobis distances based on classical (left) and robust (right) estimates.

Chapter 10
Projections of Multidimensional Data

This chapter deals with the reduction of the dimension of multivariate data through projections onto subspaces. An attempt is made to find projections of the data that still express certain properties of the data and thus cause as little loss of information as possible with regard to these properties. The aim is a 2-D or 3-D representation of the data with the help of these transformations.

10.1 Linear combinations of variables

We again assume that there are p variables $x_1, \ldots, x_p$. Instead of considering all p dimensions separately, one could get a single variable through linear combination. Such a linear combination has the form

$$u = b_1 x_1 + b_2 x_2 + \ldots + b_p x_p$$

with coefficient $b_1, \ldots, b_p$. So, all variables are linked linearly and combined into a new variable u. For example, if $b_1 = 1$ and $b_2 = \ldots = b_p = 0$, then $u = x_1$. For $b_1 = \ldots = b_p = 1$, each variable makes the same contribution to u. The coefficients can thus be understood as the weight with which the corresponding variable contributes to u.

Linear combinations can also be *interpreted geometrically*. After the variables $x_1, \ldots, x_p$ make up the dimensions (the axes in a graphic), these dimensions are weighted with the coefficients $b_1, \ldots, b_p$ and combined to form a new direction in the p-dimensional space. This direction corresponds to a linear projection. If for $x_1, \ldots, x_p$ finally n concrete observations are available, u also consists of n observations that were projected from the p-dimensional space in this direction.

The coefficients $b_1, \ldots, b_p$ are called *loadings*, and u is called the *score*.

With the notation $\boldsymbol{x} = (x_1, \ldots, x_p)^T$ and $\boldsymbol{b} = (b_1, \ldots, b_p)^T$ one also gets u through:

$$u = \boldsymbol{x}^T \boldsymbol{b}$$

The length of the direction vector $\boldsymbol{b}$ is usually normalized to 1, so $\boldsymbol{b}^T \boldsymbol{b} = 1$ applies.

In general, you will not only be interested in one projection direction (linear combination), but in several. This can be achieved simply by defining a matrix $\boldsymbol{B}$ of coefficients consisting of the columns $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$. Each of these k vectors contain p coefficients that define the k directions. One thus obtains k linear combinations:

$$u_j = \boldsymbol{x}^T \boldsymbol{b}_j \qquad \text{for } j = 1, \ldots, k$$

each of which gives different "insights" into the p-dimensional space. One often demands that one want to get as different "insights" as possible, that the directions are normal to one another. This is called *orthogonal* and is expressed mathematically as $\boldsymbol{b}_i^T \boldsymbol{b}_j = 0$, for $i, j = 1, \ldots, p, \; i \neq j$.

So far, no conditions have been placed on the projection directions. With such conditions, however, one can achieve that the coefficients can be clearly determined. For example, one could be interested in a projection direction that allows two groups in the data to be distinguished as well as possible. The direction obtained is thus clearly defined. Another criterion could be that the variance of the projected points is maximal (the direction is therefore as informative as possible). This leads to the first principal component, which is the subject of the next section.

10.2 Principal components

Principal Component Analysis (PCA) is one of the most important methods of multivariate statistics. Principal components are often used as an exploratory tool that provides an overview of the data.

10.2.1 Definition of the principal components

We assume an $n \times p$ data matrix $\boldsymbol{X}$ again. The principal components are defined using linear combinations as follows:

$$\boldsymbol{U} = \boldsymbol{X}\,\boldsymbol{B}$$

The $p \times p$ matrix $\boldsymbol{B}$ contains coefficients that define p new directions. $\boldsymbol{U}$ is the *score* matrix of dimension $n \times p$, i.e. the same dimension as **X**, which contains the projected data values. The individual columns $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p$ of the *score* matrix are called *principal components*.

The individual columns $\boldsymbol{b_1}, \ldots, \boldsymbol{b_p}$ of the matrix **B** are clearly determined using the following criteria:

- The coefficients $\boldsymbol{b}_1$ for determining the first principal component $\boldsymbol{u}_1$ are chosen so that the variance of $\boldsymbol{u}_1$ is maximal. In addition, $\boldsymbol{b}_1^T\boldsymbol{b}_1 = 1$ is required.
- The coefficients $\boldsymbol{b}_2$ for determining the second principal component $\boldsymbol{u}_2$ are chosen so that the variance of $\boldsymbol{u}_2$ is maximal. In addition, $\boldsymbol{b}_1^T\boldsymbol{b}_2 = 0$ and $\boldsymbol{b}_2^T\boldsymbol{b}_2 = 1$ are required.
- The coefficients $\boldsymbol{b}_j$ for determining the j-th principal component $\boldsymbol{u}_j$ $(2 < j \leq p)$ are chosen so that the variance of $\boldsymbol{u}_j$ is maximal. In addition, $\boldsymbol{b}_j^T\boldsymbol{b}_l = 0\ for\ (1 \leq l < j\ )$ and $\boldsymbol{b}_j^T\boldsymbol{b_j} = 1$ are required.

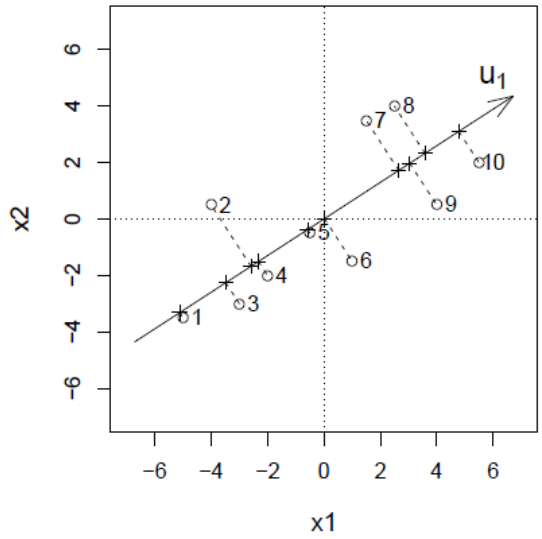


Figure 10.1: Determination of the first principal component with maximum variance.

Figure 10.1 shows for 10 data points in two dimensions what the first principal component could look like. The points projected onto this component therefore have maximum variance. You cannot find any other direction with a higher variance.

From a geometrical point of view, $\boldsymbol{U} = \boldsymbol{XB}$ results in nothing other than a representation of the data in a new coordinate system that is even orthogonal (the columns of $\boldsymbol{B}$ are orthogonal). Thus $\boldsymbol{U}$ contains exactly the same information as $\boldsymbol{X}$, only in a different representation. The representation is special because, according to the construction, the information content (variance) of the first principal component is greatest, and this information content decreases with increasing number of principal components. One can therefore assume that the last principal components are uninformative and can therefore be omitted. This would result in a reduction in dimensions because the first few principal components are "sufficient". More precise criteria for the relevant number of principal components are given in Section 10.2.3.

10.2.2 Algorithm for determining the principal components

From a mathematical point of view, the determination of the principal components is a maximization problem under secondary conditions. We formulate this problem with random variables $x_1, \ldots, x_p$. The reason for this is because it is about variance maximization and so we can work with the theoretical variance "Var". If one were to take one's "favorite estimator" for the variance, the properties of the resulting principal components would depend to a large extent on the properties of this estimator.

According to the above definitions, the j-th principal component is the linear combination:

$$u_j = x_1 b_{1j} + \ldots + x_p b_{pj}$$

for $1 \leq j \leq p$. The coefficient vector $\boldsymbol{b}_j = (b_{1j}, \ldots, b_{pj})^T$ is normalized with $\boldsymbol{b}_j^T \boldsymbol{b}_j = 1$, and is orthogonal to "earlier" directions $\boldsymbol{b}_j^T \boldsymbol{b}_l = 0$ $(for\ j > l,\ if\ j \geq 2)$. The goal is to maximize variance, so $Var(u_j)$ should be maximized. But this is:

$$\mathrm{Var}(u_j) = \mathrm{Var}(x_1 b_{1j} + \ldots + x_p b_{pj}) = \boldsymbol{b}_j^T \mathrm{Cov}(x_1, \ldots, x_p) \boldsymbol{b}_j = \boldsymbol{b}_j^T \boldsymbol{\Sigma} \boldsymbol{b}_j$$

The matrix $\boldsymbol{\Sigma}$ is the theoretical covariance matrix related to the population. A maximization of this expression under constraints is formulated as a Lagrange problem:

$$\phi_j = \boldsymbol{b}_j^T \boldsymbol{\Sigma} \boldsymbol{b}_j - \lambda_j (\boldsymbol{b}_j^T \boldsymbol{b}_j - 1) \qquad \text{for } j = 1, \ldots, p$$

with the Lagrange parameters λ_j. The partial derivatives according to the unknown parameters $\boldsymbol{b}_j$ are set to zero, i.e.

$$\frac{\partial \phi_j}{\partial \boldsymbol{b}_j} = 2\boldsymbol{\Sigma} \boldsymbol{b}_j - 2\lambda_j \boldsymbol{b}_j = 0$$

This is equivalent to:

$$\boldsymbol{\Sigma} \boldsymbol{b}_j = \lambda_j \boldsymbol{b}_j \qquad \text{for } j = 1, \ldots, p$$

One recognizes from this that one arrives at an *eigenvalue problem*: $\boldsymbol{b}_j$ are the *eigenvectors* of Σ to the *eigenvalues* λ_j.

The role of the eigenvalues λ_j is also important:

$$\mathrm{Var}(u_j) = \boldsymbol{b}_j^T \boldsymbol{\Sigma} \boldsymbol{b}_j = \boldsymbol{b}_j^T \lambda_j \boldsymbol{b}_j = \lambda_j \boldsymbol{b}_j^T \boldsymbol{b}_j = \lambda_j$$

The eigenvalues are thus the variances of the principal components that have been maximized. The eigenvectors (columns of ***B***) and eigenvalues are ordered such that: $\lambda_1 \geq \ldots \geq \lambda_p$.

Hint: The directions of the principal components are the eigenvectors of the covariance matrix Σ. Imagine that the variance of x_1 is a thousand times the variance of the rest of the variables. If a search is now made for a direction that maximizes the variance, this direction will more or less exactly match x_1. If this effect is not desired, i.e., regardless of the variance of the individual variables, one would like that each variable can contribute equally to the main components, then scaling must first be carried out. In the simplest case, the variances of the scaled variables should be 1. However, this means that one does not take eigenvectors from the covariance matrix but from the correlation matrix. If one uses classical estimators, one does not determine eigenvectors (and eigenvalues) of ***S***, but rather from the empirical correlation matrix ***R***.

```
R:   X.pca <- princomp(X,cor=TRUE)
```

10.2.3 Number of relevant principal components

The aim of the principal component analysis is to reduce the dimensions with as little loss of information as possible. After the principal components have been ordered according to descending variance $\lambda_1 \geq \ldots \geq \lambda_p$, one looks for a number k, with $1 \leq k \leq p$, of principal components that represent the "essential" information of the data.

Since the total variance of the data can be printed out as $\lambda_1 + \ldots + \lambda_p$, the information content of the first k main components corresponds to $\lambda_1 + \ldots + \lambda_k$. The proportion of the total variance that is expressed by the first k principal components is therefore suitable as a criterion:

$$\frac{\lambda_1 + \ldots + \lambda_k}{\lambda_1 + \ldots + \lambda_p}$$

This part is often visualized. The resulting *scree plot* ("Geroll plot") then gives an indication of the "optimal" k: Figure 10.2 (left) shows a schematic of such a *scree plot*. One would like to define *k* where the line "flattens out". All points that are roughly on a straight line correspond to principal components with irrelevant information content and therefore these can be omitted. In the graph one would therefore decide for $k = 2$. Figure 10.2 (right) shows the cumulative variances of the principal components.

As a rule of thumb, you could choose k in such a way that at least 80% of the total variance is explained. However, this proportion strongly depends on what you want to do with the first k principal components (visualization or modeling).

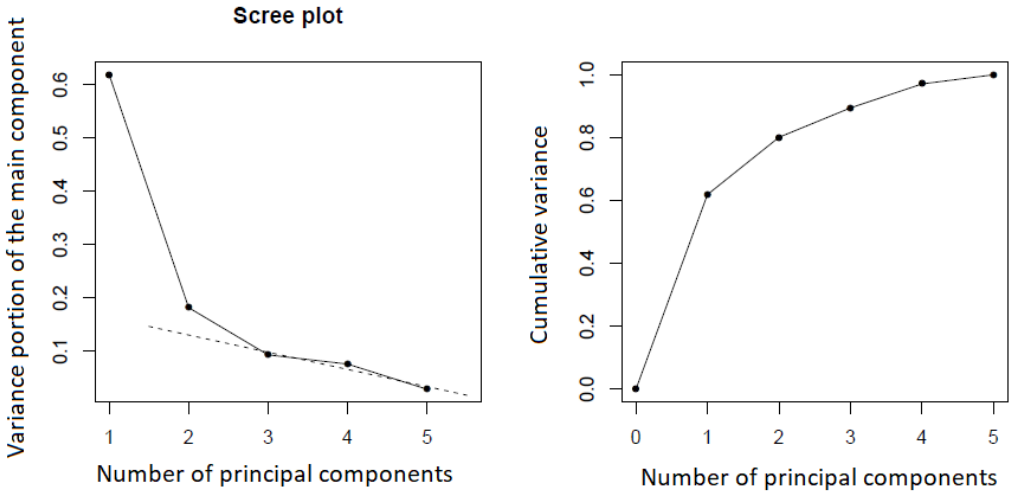


Figure 10.2: Scree plot (left) and proportion of the declared total variance (right). You get the data with *data (scor, package = "bootstrap")*.

10.2.4 Centering and scaling the data

As discussed above, you will generally get different principal components if you take the unscaled or the scaled data. But the centering of the data will also play a role.

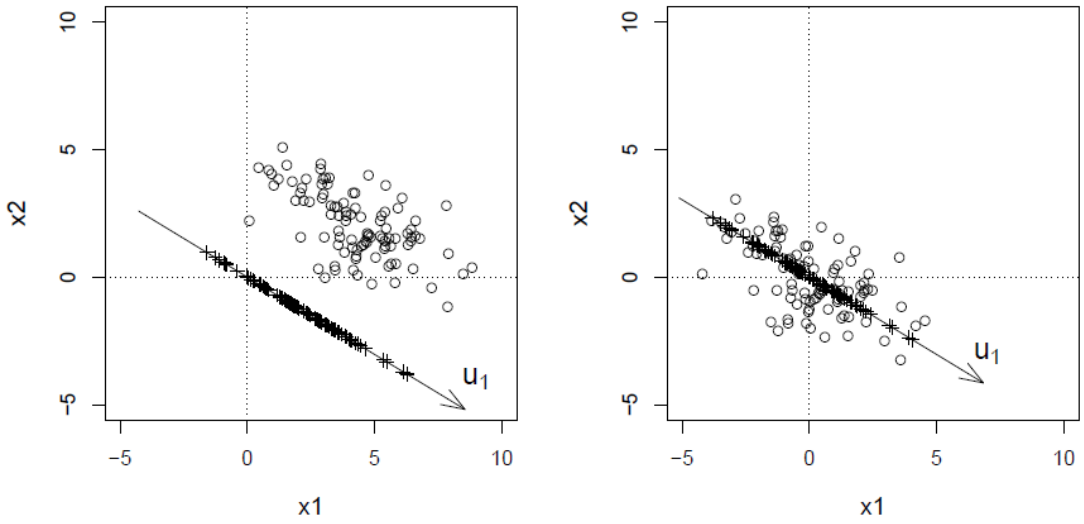


Figure 10.3: Effect of centering the data matrix on column mean zero when calculating the main components: left uncentered, right centered

Centering refers to the means of the original variables. With normal measurements, the mean of the individual variables will not be 0, so the data are *not centered*. If these mean values were brought to 0, one speaks of *centered* data. Centering is achieved by subtracting the column mean (arithmetic mean, median, etc.) from the respective column of the matrix. Figure 10.3 shows the effect of centering in connection with the calculation of the principal components. Uncentered data were used on the left, centered on the right. The direction of the first principal component has not changed.

The ratio of the projected points (*scores*) to one another is also unchanged. However, in the uncentered case the scores are also uncentered, in the centered case they are centered.

In general, one should always use the centered data for the determination of the principal components, because one is generally only interested in the reconstruction of the structure of the data, but not in the reconstruction of the location.

Scaling refers to the variances of the original variables. With normal measurements, the variance of the individual variables will not be 1, so the data are *unscaled*. If these variances were brought to 1, one speaks of *scaled data*. Scaling is achieved by dividing the values of the respective column by the standard deviation of the column values (root of the sample variance, MAD, etc.). Figure 10.4 shows the effect of scaling in connection with the calculation of the principal components. Obviously the variance of the variable x_1 is much higher than that of x_2. The (centered) unscaled data were used on the left, scaled on the right. The direction of the first principal component changes, and thus the scores of the first principal component also change. The graph on the left shows that x_1 contributes significantly more to the determination of the principal component direction than x_2. If you don't want to have this effect, that variables with higher variance automatically make a higher contribution, you have to scale first.

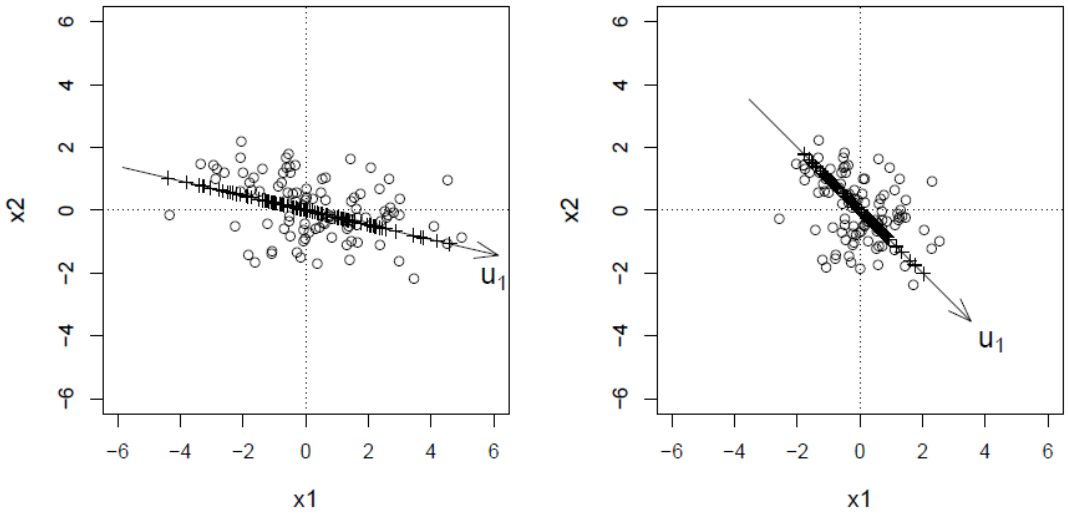


Figure 10.4: Effect of scaling the data matrix on column variance 1 when calculating the principal components: unscaled on the left, scaled on the right.

10.2.5 Normal distribution and outliers

The definition of the principal components did not speak of any distribution assumption. So do you need a multivariate normal distribution? The mathematical derivation manages without this requirement.

Figure 10.5 (left) shows bivariate data, where especially the variable x_2 is very skewed (skewed to the right). The data has been centered and scaled here, and the direction of the first principal component seems to make sense. However, you will not be able to get by with one principal component in order to adequately describe the structure in the data; you also need the second principal component. So we have not achieved a dimensional reduction.

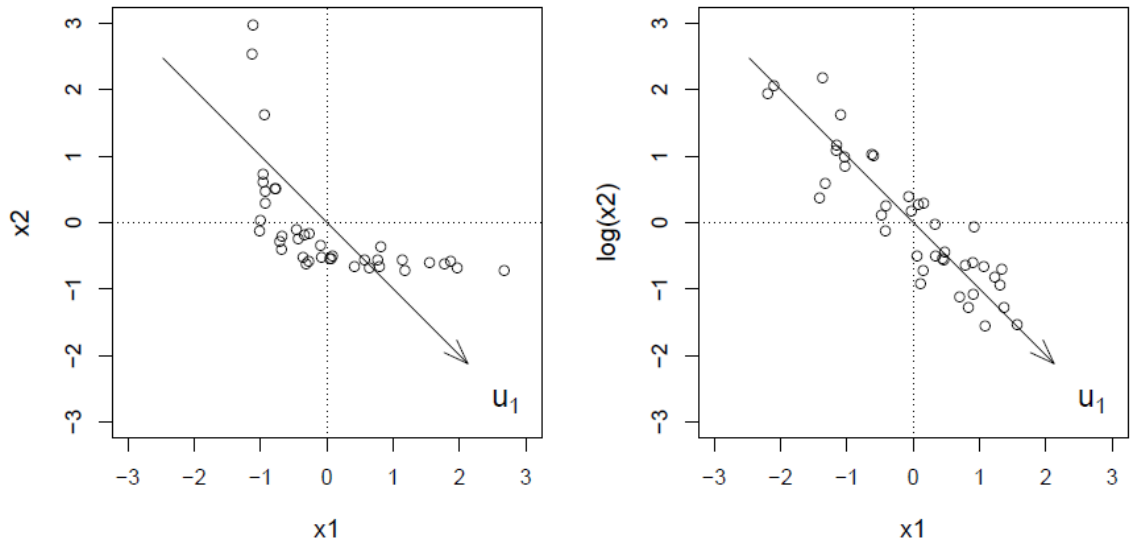


Figure 10.5: Asymmetrical distribution (left) and symmetrization through transformation (right).

In Figure 10.5 (right) the variable $log(x_2)$ was used instead of x_2, and thus symmetry was achieved - possibly even bivariate normal distribution. x_1 and $log(x_2)$ have a strong linear relationship, which is expressed by the direction of the first principal component. A second component will not be necessary here, and thus the dimension reduction could be used successfully. Elliptical symmetry of the data thus seems to be essential for efficient dimension reduction.

Another question is whether outliers can influence the direction of the principal components. In Figure 10.6, two groups of outliers are visible in the data. In the graphic on the left, the "classic" estimation of the first principal component was carried out, in the graphic on the right a robust variant. The direction is obviously changing. Classical estimation means here that the directions correspond to the eigenvectors of the "classical" empirical correlation matrix ***R*** (or the empirical covariance matrix). In the robust case, eigenvectors are taken from the robust estimate of the covariance or correlation matrix. Such a robust estimate can be obtained, for example, from the MCD estimator, see Section 9.1.1.

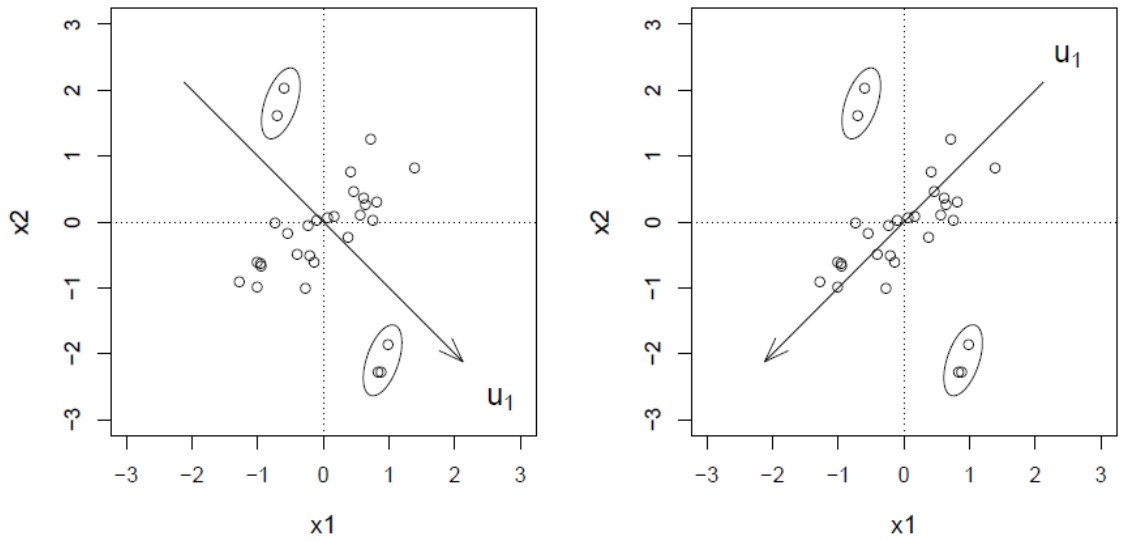


Figure 10.6: Effect of outliers: on the left the solution of classical, on the right of robust principal component analysis.

```
R:   library(robustbase)
R:   X.mcd <- covMcd(X,cor=TRUE)      # also provides a robust correlation matrix
R:   X.pca <- princomp(X,covmat=X.mcd,cor=TRUE)      # robust analysis
```

10.2.6 Presentation of the results, biplot

In addition to a *scree plot*, other results of the principal component analysis are also essential for a visualization, namely the obtained *scores* ***U*** and the *loadings* ***B***. One is particularly interested in the first k columns of these matrices, which carry the relevant information. The matrix of the *scores* is thus of dimension $n \times k$, and that of the *loadings* has dimension $p \times k$ .The *scores* contain the information about the objects and the *loadings* contain the information about the variables.

Example: The examination results of 88 students in the subjects Mechanics, Analytical Geometry, Linear Algebra, Analysis and Elementary Statistics were recorded. Of the 100 achievable points there were the results listed in Table 10.1. The data is available with *data(scor, package = "bootstrap").*

Table 10.1: Exam results of 88 students in the subjects Mechanics (ME), Analytical Geometry (AG), Linear Algebra (LA), Analysis (AN) and Elementary Statistics (ES); 100 points were achievable.

Student	ME	AG	LA	AN	ES	Student	ME	AG	LA	AN	ES
1	77	82	67	67	81	45	46	61	46	38	41
2	63	78	80	70	81	46	40	57	51	52	31
3	75	73	71	66	81	47	49	49	45	48	39
4	55	72	63	70	68	48	22	58	53	56	41
5	63	63	65	70	63	49	35	60	47	54	33
6	53	61	72	64	73	50	48	56	49	42	32
7	51	67	65	65	68	51	31	57	50	54	34
8	59	70	68	62	56	52	17	53	57	43	51
9	62	60	58	62	70	53	49	57	47	39	26
10	64	72	60	62	45	54	59	50	47	15	46
11	52	64	60	63	54	55	37	56	49	28	45
12	55	67	59	62	44	56	40	43	48	21	61
13	50	50	64	55	63	57	35	35	41	51	50
14	65	63	58	56	37	58	38	44	54	47	24
15	31	55	60	57	73	59	43	43	38	34	49
16	60	64	56	54	40	60	39	46	46	32	43
17	44	69	53	53	53	61	62	44	36	22	42
18	42	69	61	55	45	62	48	38	41	44	33
19	62	46	61	57	45	63	34	42	50	47	29
20	31	49	62	63	62	64	18	51	40	56	30
21	44	61	52	62	46	65	35	36	46	48	29
22	49	41	61	49	64	66	59	53	37	22	19
23	12	58	61	63	67	67	41	41	43	30	33
24	49	53	49	62	47	68	31	52	37	27	40
25	54	49	56	47	53	69	17	51	52	35	31
26	54	53	46	59	44	70	34	30	50	47	36
27	44	56	55	61	36	71	46	40	47	29	17
28	18	44	50	57	81	72	10	46	36	47	39
29	46	52	65	50	35	73	46	37	45	15	30
30	32	45	49	57	64	74	30	34	43	46	18
31	30	69	50	52	45	75	13	51	50	25	31
32	46	49	53	59	37	76	49	50	38	23	9
33	40	27	54	61	61	77	18	32	31	45	40
34	31	42	48	54	68	78	8	42	48	26	40
35	36	59	51	45	51	79	23	38	36	48	15
36	56	40	56	54	35	80	30	24	43	33	25
37	46	56	57	49	32	81	3	9	51	47	40
38	45	42	55	56	40	82	7	51	43	17	22
39	42	60	54	49	33	83	15	40	43	23	18
40	40	63	53	54	25	84	15	38	39	28	17
41	23	55	59	53	44	85	5	30	44	36	18
42	48	48	49	51	37	86	12	30	32	35	21
43	41	63	49	46	34	87	5	26	15	20	20
44	46	52	53	41	40	88	0	40	21	9	14

We would now like to represent these 5-dimensional data by means of principal components. In Figure 10.2, the same data have already been used for the *scree plot* and the plot of the cumulative proportions of variance. It could be seen from this that k = 2 makes sense and that 2 principal components represent about 80% of the total variability. Figure 10.7 now also shows the *scores* (left) and *loadings* (right) of these first two principal components. You can see that the scores in the direction of the 1st principal component

have a certain order in the numbers (number of students). The "top students" can be found on the left, those with the poorer results on the right. If you look at the *loadings* plot, you can also interpret this direction of the 1st principal component: The charges of the 5 variables are roughly the same, and thus the direction of the 1st principal component corresponds to an "average power". The smaller the *score*, the better the average performance. The second principal component differentiates between the more geometry-heavy objects (positive) and the more analytical-heavy objects (negative). This is also expressed in the position of the students in the scores of the 2nd principal component.

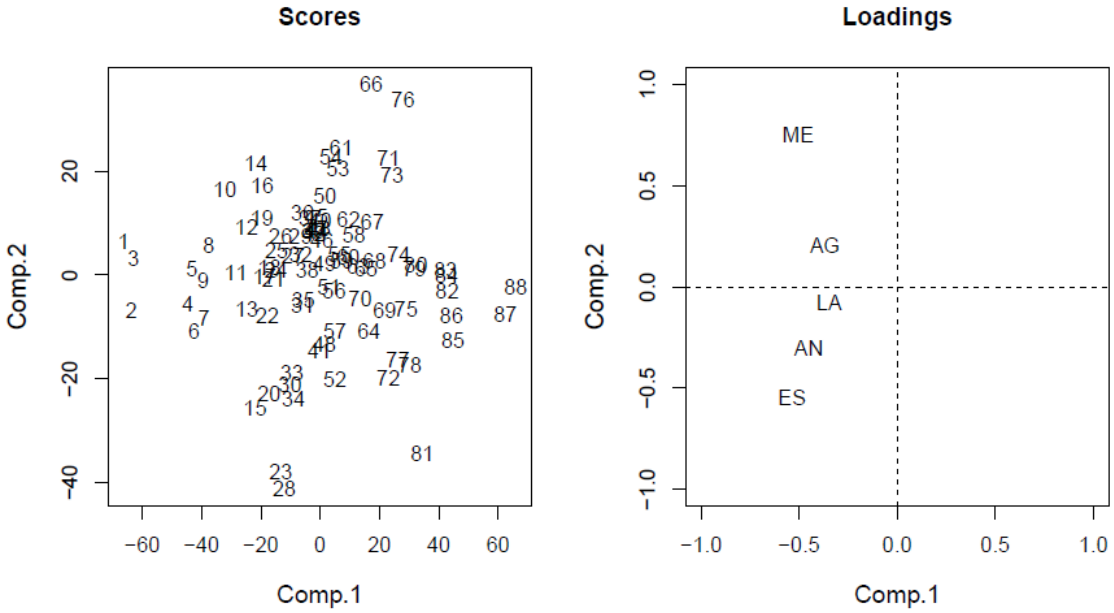


Figure 10.7: *Scores* (left) and *loadings* (right) of the first two principal components. The data is obtained with *data (scor, package = "bootstrap")*, see Table 10.1.

The two figures in 10.7 for *scores* and *loadings* can also be combined in one graphic. The result is the biplot shown in Figure 10.8. The "Bi" does not refer to 2-dimensional, but to the fact that both *scores* and *loadings* are displayed at the same time. You can see that the scaling was chosen differently here than in Figure 10.7. The reason for this lies in the interpretation of the relationship between *scores* and *loadings* in this plot:

- The orthogonal projections of the observations onto the variables (arrows) approximate the original (centered) data values.
- The cosine of the angle between the variables (arrows) approximates the correlation between the variables.
- The Euclidean distances between the observations approximate the Mahalanobis distances of these observations.

When the term "approximation" is used here, it refers to the information content of these two principal components, which here represent 80% of the information.

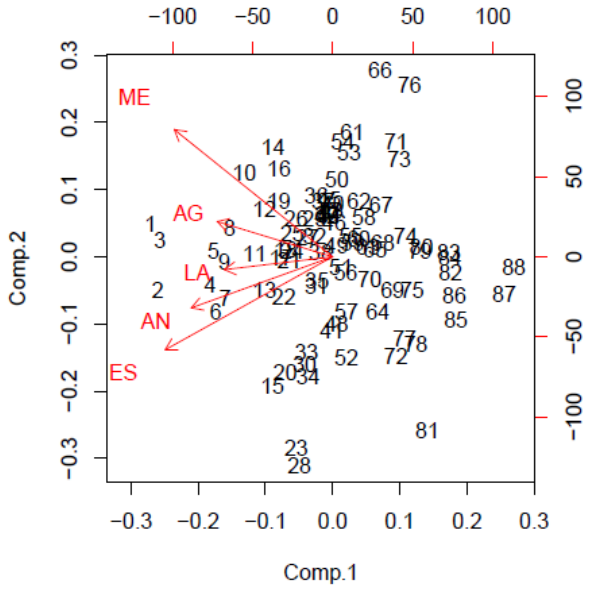


Figure 10.8: Biplot of the first two principal components. The data is obtained with *data (scor, package = "bootstrap")*, see Table 10.1.

```
R:   data(scor,package="bootstrap")
R:   biplot(princomp(scor))
```

10.3 Projection Pursuit

This procedure aims to find projections of the p-dimensional observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ onto lower dimensions so that interesting structures (nonlinearities, clusters, ...) are expressed through these projections. *Pursue* thus prints out that interesting projections are "followed". This is done by choosing a projection index at whose extreme values the structure is most clearly recognizable.

The projection index should be affine invariant: It should show the same projection, regardless of whether the original data $\boldsymbol{x}_i (i = 1, \ldots, n)$ or transformed data

$$\boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{b}$$

are used. Here $\boldsymbol{A}$ is a regular $p \times p$ matrix that is orthonormal (i.e. $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$) and $\boldsymbol{b}$ is a p-dimensional vector.

Projection pursuit solutions (i.e. the projection directions found, projection planes,...) are rarely unambiguous because the projection index usually has many local extreme values. Since every projection can contribute to the recognition of the structure, one should let the projection pursuit algorithm iterate with as many initial values as possible in order to arrive at different local optima.

10.3.1 Projection index

It should be designed in such a way that it shows structures in the data that are not described by the correlation matrix. This is achieved by a projection index that is invariant to all non-singular affine transformations in $\mathbb{R}^p$.

The index should be large if the projection is interesting, otherwise small. When is a projection interesting? One wants to be able to recognize groups or non-linear behavior. Thus, projections that follow a normal distribution are viewed as uninteresting because neither groups nor asymmetries are visible here.
It follows from this that the projection index is constructed in such a way that with increasing deviations from normally distributed projections, the value of the index also increases.

10.3.2 Calculation of the projection index

In order to meet the requirement of affine invariance, the first step is to display the data in the form of standardized principal components. If we use the notation of random variables again, then the original variables $x_1, \ldots, x_p$ are represented as principal components $u_1, \ldots, u_p$ up, see Section 10.2.2. The variances of these main components are $x_1, \ldots, x_p$. So you define

If we take the notation of random variables again, then the original variables $x_1, \ldots, x_p$ are shown as principal components $u_1, \ldots, u_p$, see Section 10.2.2. The variances of these principal components are $\lambda_1, \ldots, \lambda_p$. So you define

$$z_j = \frac{u_j}{\sqrt{\lambda_j}} \qquad \text{für } j = 1, \ldots, p.$$

then $Var(z_j) = 1$, and we have standardized principal components.

Note: If the last λ_j were zero, then these components were simply left out, and we had all the information presented in a subspace.

We are first looking for univariate projection directions $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)^T$ in p-dimensional space. To ensure that the direction is unambiguous, $\boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1$ is required. This has the projection

$$Y = \alpha_1 z_1 + \ldots + \alpha_p z_p$$

also variance 1,

$$\mathrm{Var}(Y) = \boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1.$$

The projection index should now get a high value if the density of *Y* is heavily structured. We denote this density with $p_\alpha(Y)$. The algorithm for the projection index has the following form:

1. Transformation of *Y* into the interval (-1, 1) by
   $$R = 2\Phi(Y) - 1$$
   with ϕ distribution function of the standard normal distribution.
   If Y is normally distributed, R is continuously uniformly distributed in (-1, 1), and the value of the density function in this interval is $\frac{1}{2}$.
2. As a measure of the deviation of the density $p_R(r)$ from *R* from the uniform distribution, the projection index $I(\alpha)$ is defined as

$$I(\boldsymbol{\alpha}) = \int_{-1}^{1} \left(p_R(R) - \frac{1}{2} \right)^2 dR = \int_{-1}^{1} p_R^2(R) dR - \frac{1}{2}.$$

For the practical calculation this integral is approximated by polynomials.

In Figure 10.9 the functionality of this algorithm is illustrated with examples. The five images above assume that the density function of the projection is standard normally distributed. The next row of figures is based on the χ^2-distribution with 2 degrees of freedom, and the lower five images show how the algorithm transforms a bimodal distribution (composition of two normal distributions).

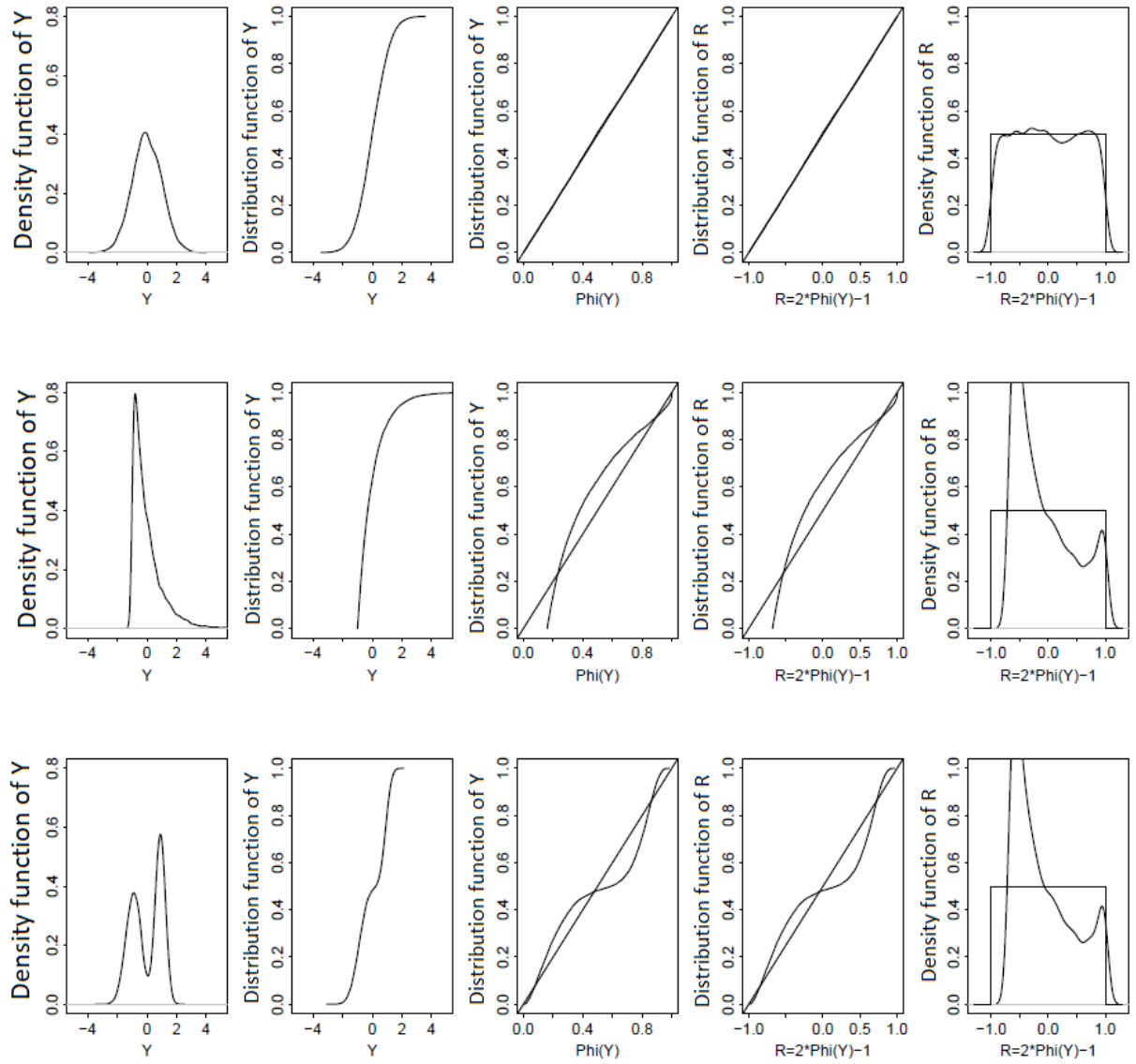


Figure 10.9: Illustration of how the projection pursuit algorithm works.

For 2-dimensional projections, 2 directions $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)^T$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ are sought, so that the common density of

$$Y_1 = \alpha_1 z_1 + \ldots + \alpha_p z_p$$
$$Y_2 = \beta_1 z_1 + \ldots + \beta_p z_p$$

is heavily structured. *Y1* and *Y2* must be uncorrelated, which is equivalent to $\boldsymbol{\alpha}^T \boldsymbol{\beta} = 0$. Furthermore, $\boldsymbol{\alpha}^T \boldsymbol{\alpha} = \boldsymbol{\beta}^T \boldsymbol{\beta} = 1$ must apply. A procedure analogous to the previous one then provides a projection index $\boldsymbol{I}(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

Example: We consider the Iris data, which is known to have a relatively strong structure because there are 3 groups in the data. Figure 10.10 compares the results for 1-dimensional projections when using principal components (left) and when using the projection pursuit method (right). The solution with Projection Pursuit is much better structured, the groups overlap less.

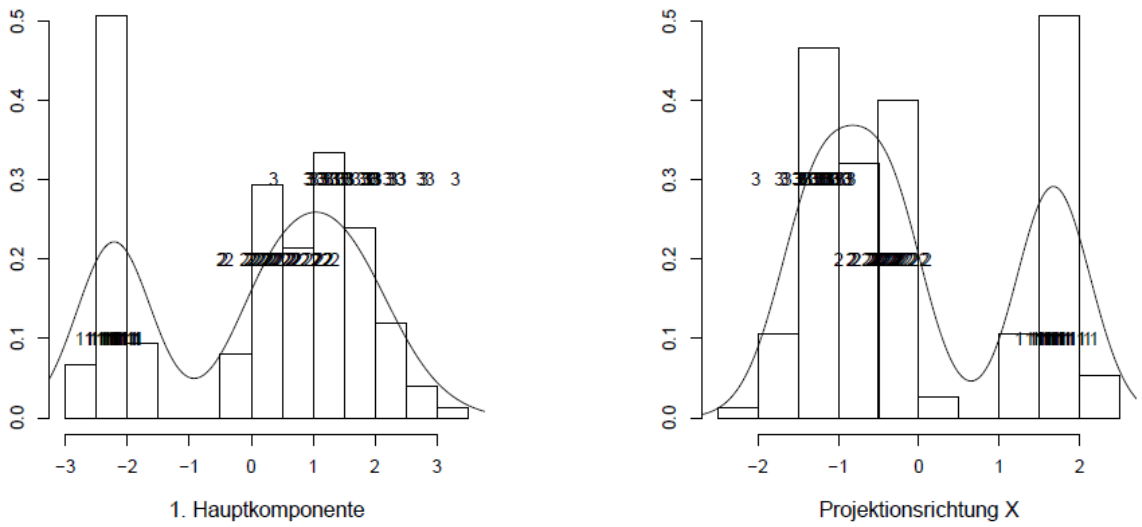


Figure 10.10: Iris data: 1-dimensional projection onto the first principal component (LEFT) and projection pursuit for a 1-dimensional projection direction (RIGHT).

Figure 10.11 shows the Projection Pursuit solution for a two-dimensional projection. The graph on the left is the projection of the data onto the desired projection directions, the graph on the right shows an estimate of the density of the solution.

10.3.3 Structure elimination

If you have found an interesting projection direction, you can continue the search for further projection directions. However, in order to prevent the projection index from being influenced by the direction already found, the structure that is most clearly expressed in this direction must be eliminated. Since the projection index evaluates deviations from the normal distribution most strongly, but normal distribution results in an index value of 0, an attempt must be made to find a transformation that generates a normal distribution in the projection direction that has already been found.

If the projection direction α was found after using the projection pursuit method (one-dimensional version), the structure elimination is carried out as follows:

1. Transform $\boldsymbol{z} = (z_1, \ldots, z_p)^T$ by an orthonormal transformation such that the first coordinate *Y* of the transformed variable is the projection in the direction $\boldsymbol{\alpha}$ (i.e. $\boldsymbol{Y} = \boldsymbol{\alpha}^T \boldsymbol{z}$).
2. Transform this first coordinate to normal distribution

   $$Y' = \Phi^{-1}(F_{\boldsymbol{\alpha}}(Y))$$

   with F_{α} distribution function of Y.
   Since one does not know F_{α}, one uses the empirical distribution function instead of the distribution function.
3. Reverse the orthonormal transformation. Apply the projection pursuit algorithm to the variable z' obtained in this way.

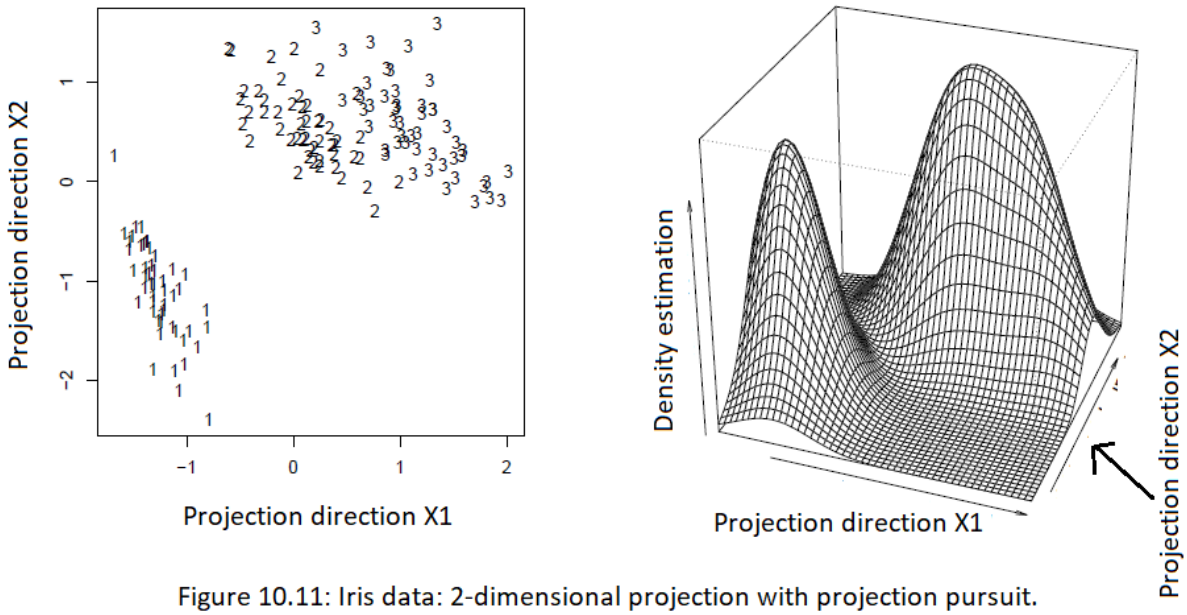


Figure 10.11: Iris data: 2-dimensional projection with projection pursuit.

Structure elimination in the 2-dimensional case is a little more difficult, since a transformation is required that transforms any 2-dimensional distribution to a 2-dimensional normal distribution. Theoretical solutions cannot be carried out arithmetically. However, there is an approximate solution that is not discussed here.

Chapter 11

Other multivariate statistical methods - an overview

11.1 Cluster analysis

The term "cluster" has the meaning of "concentrated" group. The aim of cluster analysis is to divide observations into homogeneous groups. Observations that are very similar should be in the same group (in the same cluster), and observations that are dissimilar to one another should be divided into different clusters. In special cases, the clustering of variables is not interested in the clustering of observations (see boxes or trees for multivariate graphs, Section 8).

Similarities of observations are determined over distances. All distance measures from Section 9.2 can be used here. *Caution*: Since distances depend on the scaling of the variables, it will be important in most clustering procedures to first standardize the variables to mean 0 and variance 1 in order to ensure that the variables can be compared.

The "real" number k of clusters in multivariate data is usually unknown and has to be estimated by criteria to be chosen later. Usually there are no clearly separated groups in the data, which makes it even more difficult to estimate the number of clusters and to assign observations to the clusters. There are basically several approaches to building clusters:

- **Partitioning methods**: Each observation is assigned to exactly one cluster. A disjoint grouping of the observations into a number k of clusters is thus obtained.
- **Hierarchical clustering methods**: A hierarchy of partitionings is constructed in which the number of clusters varies from 1 to n (= number of observations). This allows an overview of different numbers of groupings, and this overview is visualized by means of a *dendrogram*. Hierarchies can be constructed agglomeratively (from an n-cluster to a 1-cluster partitioning) or divisively (a cluster is split up step by step until an n-cluster solution is created), whereby agglomerative is the most useful.
- **Fuzzy clustering**: Each observation is assigned to each cluster via an association coefficient. The coefficients are from the interval [0, 1], and the sum of the coefficients for an observation over all clusters is 1.
- **Model-based clustering**: This is a partitioning method, but the shape of a cluster is parameterized by a model distribution. The typical model distribution is a multivariate normal distribution with a certain expected value and a certain covariance.

We again assume an $n\ x\ p$ data matrix **X**, and want to cluster the observations $x_1, x_2, \ldots, x_n$.

11.1.1 Partitioning methods

Let k be the number of clusters into which the observations are to be disjointly divided. Furthermore, let I_j be an index set that contains the indices of the observations of the j-th cluster, and n_j the number of elements in I_j ($j = 1,\ldots,k$). The index set has the form $I_j = \{i_1, i_2, \ldots, i_j\}$, and accordingly the j-th cluster contains the observations $x_1, x_2, \ldots, x_j$. The following applies to partitioning $n_1 + n_2 + \ldots + n_k = n$.

The best-known algorithm for the construction of a partitioning is the **k-means** algorithm. It needs the desired number k of clusters as an input parameter. *k-means* uses so-called centroids or cluster centers, which can simply be taken as the arithmetic mean of the observations of a cluster:

$$\bar{\boldsymbol{x}}_j = \frac{1}{n_j} \sum_{i \in I_j} \boldsymbol{x}_i \quad \text{for} \quad j = 1, \ldots, k \tag{11.1}$$

The objective function to be minimized for k-means is

$$\sum_{j=1}^{k} n_j \sum_{i \in I_j} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j\|^2 \longrightarrow \min . \tag{11.2}$$

An iterative algorithm is used for minimization, the index sets I_j changing to a greater or lesser extent in each step. After convergence one obtains the final index sets $I_1^*, \ldots, I_k^*$, and thus the final allocation of the n observations to the desired k clusters. The iterative algorithm is usually initialized via a random selection of k observations, which initially form the k cluster centers. Depending on the random selection, you can therefore come up with different cluster solutions!

```
# R code k-means
Xs <- scale(X)                # X is the data matrix that is standardized here
res <- kmeans(Xs,3)           # k = 3 is the number of clusters
str(res)                      # shows the contents of the output object
res$cluster                   # shows the allocation of the objects to the clusters
```

A very similar algorithm to *k-means* is **PAM** (Partitioning Around Medoids), only here the cluster centroids are robustly determined via medians.

```
# R code PAM
Xs <- scale(X)          # X is the data matrix that is standardized here
library(cluster)
res <- pam(Xs,3)        # k = 3 is the number of clusters
str(res)                # shows the contents of the output object
res$clustering          # shows the allocation of the objects to the clusters
plot(res)               # Diagnostic plots
```

A decision about a "sensible" number k of clusters can actually only be made to a degree that will be defined later. You can then execute *k-means* or *PAM* with different values of k, and choose the best solution based on the measure of good. This is also helpful if there are different solutions for the same k.

11.1.2 Hierarchical clustering methods

We describe here the agglomerative approach to constructing a hierarchy of partitions. At the beginning, each of the n observations forms its own cluster - such clusters are called singletons. Then those singletons with the smallest distance are combined into a cluster, and $n-1$ clusters are obtained. If larger clusters are to be combined instead of singletons, then a definition of the distance between clusters is required. Let two different clusters be given by the index sets I_j and $I_{j'}$. The following definitions are used, which also denote the names of the corresponding algorithms:

- **Complete linkage:** $\max_{i\in I_j, i'\in I_{j'}} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$
- **Single linkage:** $\min_{i\in I_j, i'\in I_{j'}} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$
- **Average linkage:** $\text{average}_{i\in I_j, i'\in I_{j'}} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$
- **Centroid Methode:** $d(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}}_{j'})$
- **Ward Methode:** $d(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}}_{j'})\frac{\sqrt{2n_j n_{j'}}}{\sqrt{n_j+n_{j'}}}$

A distance measure from Section 9.2 can be taken as the distance $d(\cdot,\cdot)$, e.g. the Euclidean distance. The different algorithms usually also cause different clustering. For example, chain-shaped clusters are typically formed with *single linkage*.

Figure 11.1 illustrates the distances *complete* and *single linkage* for two clusters (indicated by ellipses). In one step of the hierarchical clustering, those two clusters with the smallest distance to one another are connected.

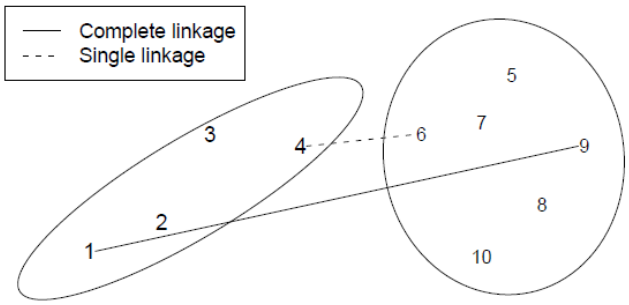


Figure 11.1: Illustration of the complete and single linkage distance (here based on Euclidean distance) for two clusters.

For the example data in Figure 11.1, Figure 11.2 now shows the entire hierarchy of the clusters as generated using *complete linkage*. In contrast to k-means, there is no randomness here - the result is always unambiguous due to the given minimization problem.

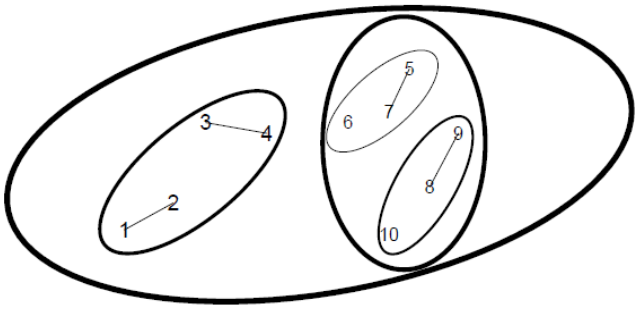


*Figure 11.2: Result of **complete linkage** on the example data from Figure 11.1. First of all, each observation forms its own cluster. These are then gradually combined until all observations are in a single cluster.*

The result in Figure 11.2 can be better represented with the **dendrogram**. The distance is plotted in the vertical direction (e.g. *complete linkage*), which increases gradually. Horizontal lines mean that the respective clusters are connected at the corresponding distance. The observations are arranged in such a way that the lines do not overlap. The dendrogram is therefore to be read from bottom to top.

Figure 11.3 shows the dendrogram of *complete linkage* on the left and that of *single linkage* on the right. While *complete linkage* shows two clusters very clearly (these were only united at a very large distance), this is not so obvious with *single linkage*. This representation can therefore be used to identify a "reasonable" number k of clusters in the data. The assignment of the observations to the k clusters can be obtained with a horizontal cut at the corresponding distance.

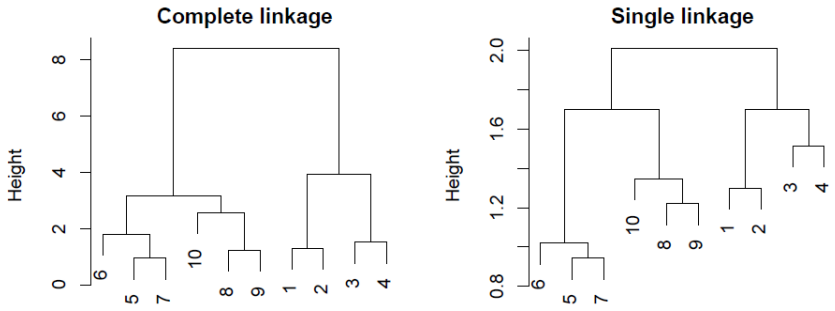


*Figure 11.3: Dendrogram for the sample data from Figure 11.1. On the left the solution for **complete linkage**, see also Figure 11.2, on the right that for **single linkage**.*

```
# R code hierarchische Clusterung
Xs <- scale(X)              # X is the data matrix that is standardized here
res <- hclust(dist(Xs))     # the default is complete linkage and Euclidean distance
plot(res)                   # shows the dendrogram
cl <- cutree(res,3)         # provides the allocation of the objects to 3 clusters
```

11.1.3 Fuzzy Clustering

Instead of "hard" allocation of the observations to the clusters, a "soft" (fuzzy) allocation is made here. During partitioning, each of the n observations is thus assigned to each of the k clusters. This is done using a membership coefficient u_{ij}, for $i = 1,\ldots,n$ and $j = 1,\ldots,k$, which lies in the interval [0,1]. It is also required that $\sum_{j=1}^{k} u_{ij} = 1$, for all i, and thus the coefficients can be interpreted as the proportional distribution of an observation to the clusters.

The cluster algorithm must therefore estimate the matrix with the coefficients u_{ij}. The number of clusters k is specified by the user analogously to k-means. One can always make a "hard" assignment of the result of fuzzy clustering by assigning an observation to the cluster for which the membership coefficient is highest.

The most common algorithm is the *fuzzy c-means* algorithm. The objective function is similar to k-means:

$$\sum_{j=1}^{k}\sum_{i=1}^{n} u_{ij}^2 \|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_j\|^2 \longrightarrow \min, \tag{11.3}$$

with the cluster centroids

$$\tilde{\boldsymbol{x}}_j = \frac{\sum_{i=1}^{n_j} u_{ij}^2 \boldsymbol{x}_i}{\sum_{i=1}^{n_j} u_{ij}^2} \quad \text{für} \quad j = 1,\ldots,k. \tag{11.4}$$

Again, the problem is solved using an iterative algorithm, the coefficients u_{ij} being re-estimated in each step. Similar to k-means, the algorithm is randomly initialized, which can lead to different results

```
# R code fuzzy clustering
Xs <- scale(X)              # X is the data matrix that is standardized here
library(e1071)
res <- cmeans(Xs,3)         # fuzzy c-means with 3 clusters
str(res)                    # shows the content of the output object
res$cluster                 # hard cluster allocation
res$membership              # Membership coefficient matrix
```

11.1.4 Model-based clustering

The shape of the clusters is "modeled" here. Usually, one assumes as a model that the j-th cluster comes from a p-dimensional normal distribution with mean $\boldsymbol{\mu_j}$ and covariance $\sum_{\boldsymbol{j}}$. The cluster algorithm must then not only determine the association of the observations with the clusters (partitioning), but also estimate the parameters $\boldsymbol{\mu_j}$ and $\sum_{\boldsymbol{j}}$. In particular, the estimation of the $p\ x\ p$ matrices $\sum_{\boldsymbol{j}}$, for $j = 1,\ldots,k$, is complex and requires a lot of information (data). One is

therefore interested in a simplification by assuming, for example, that the covariances of the clusters are all the same and perhaps even have a very simple structure.

The simplest variant would be $\sum_j = \boldsymbol{\sigma^2 I}$ for $j = 1, \ldots, k$. Here $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{\sigma^2}$ is a parameter for the variance. Thus, all clusters were circular, with the same radius in all dimensions. A less strict assumption would be $\sum_j = \boldsymbol{\sigma_j^2 I}$ for $j = 1, \ldots, k$. The clusters were still circular, but had different sizes, according to the variance $\boldsymbol{\sigma_j^2}$. Figure 11.4 illustrates different variants.

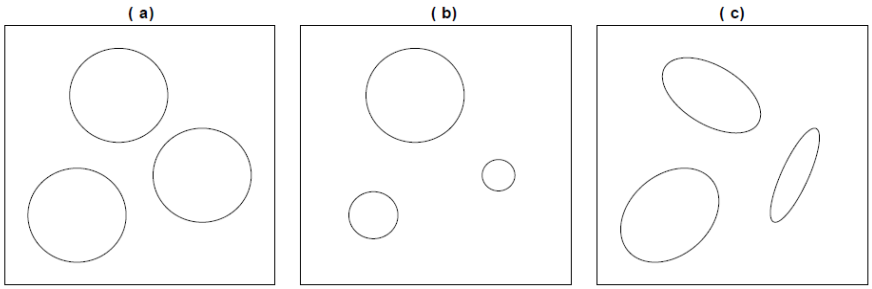


Figure 11.4: Different covariances for the three clusters: (a) $\sum_1 = \sum_2 = \sum_3 = \sigma^2 I$; (b) $\sum_j = \sigma_j^2 I$ for $j = 1, \ldots, k$; (c) all $\sum_j$ different and of no special structure.

One advantage of model-based cluster analysis is that the data does not have to be scaled beforehand (because different variances can be used in the model).

In the following example the iris data is clustered with the algorithm **Mclust()** from the package **mclust**. With this algorithm you can specify a range for the number of clusters, here from 3 to 9:

```
# R code modellbasierte Clusterung
data(iris)                      # iris data, only take columns 1-4!
library(mclust)
res <- Mclust(iris[,1:4],3:9)   # Solutions for 3 to 9 clusters
plot(res)                       # various diagnostic plots
str(res)                        # Contents of the output object
res$classification              # Cluster allocation based on the best solution
```

Results are shown in Figure 11.5. The left figure shows the BIC (Bayesian Information Criterion) values for the cluster solutions with different numbers of clusters (Number of Components). The lines correspond to different cluster models, see legend. Model "EII" is the simplest model, with covariances $\sum_j = \boldsymbol{\sigma^2 I}$. The models then become increasingly complex, up to "VVV", which corresponds to individual covariances $\sum_j$ for the individual clusters. The best model and the best number of clusters is then reached at the maximum BIC value, in this case for $k = 3$ clusters and model “VEV”.

The right figure in 11.5 shows a scatter plot of the data with the assignments of the observations to the $k = 3$ clusters, as well as the structure of the covariances drawn in with ellipses.

11.1.5 Quality measures

A measure of good should support the decision for a good choice of the number k of clusters, but it can also be used for the selection of a suitable cluster algorithm. Unfortunately, there are at least as many different good measures as there are cluster algorithms. The BIC criterion of model-based clustering is one possibility.

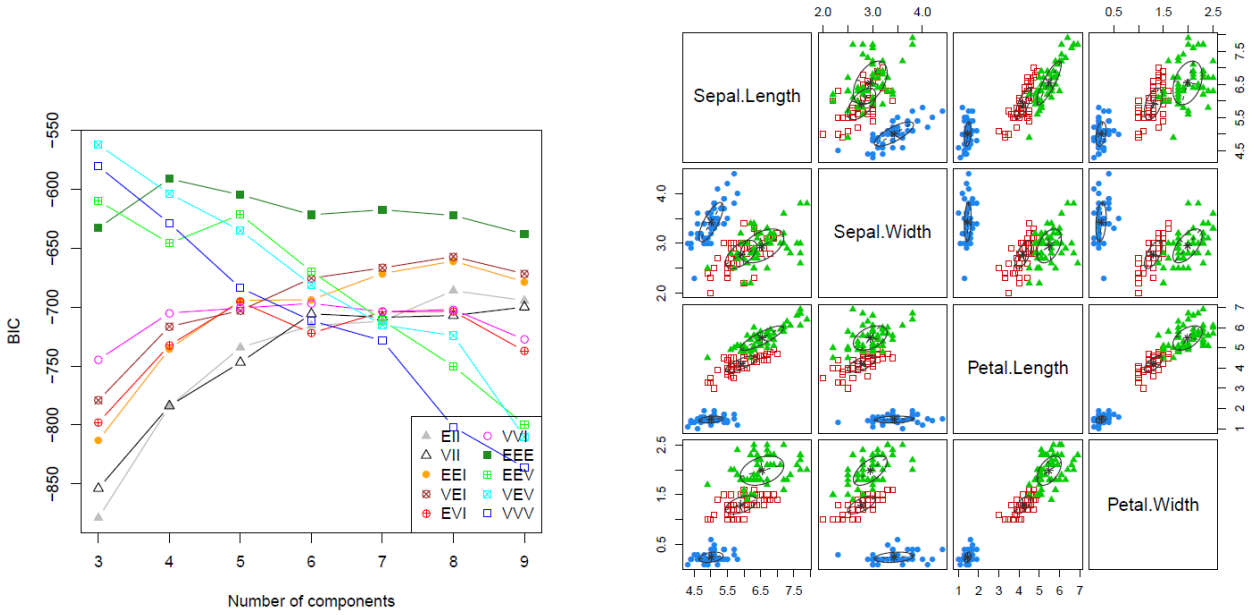


*Figure 11.5: Results of model-based clustering with the algorithm **Mclust()**: left for the desired number of clusters (horizontal axis) the BIC values for the different cluster models; on the right is the solution for the optimal model according to BIC.*

The aim of a cluster analysis is that observations within a cluster are similar to one another (homogeneous) and observations from different clusters are as different as possible (heterogeneous). Measures for homogeneity can be based on the maximum, minimum, or average distance of all observations of a cluster, or on the spread (variance) of the observations of a cluster. The latter is e.g. printed out by the *Within-Cluster Sum-of-Squares*,

$$W_k = \sum_{j=1}^{k} \sum_{i \in I_j} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j\|^2, \tag{11.5}$$

see also formulas (11.1) and (11.2). The value of W_k should be as small as possible, but of course this also depends on k; the higher k is chosen, the smaller the scatter of the points within a cluster can be.

On the other hand, one can define the heterogeneity via the distance measures of *complete linkage*, *single linkage*, etc., or can be expressed using the *Between-Cluster Sum-of-Squares*.

$$B_k = \sum_{j=1}^{k} \|\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}}\|^2, \quad \text{mit} \quad \bar{\boldsymbol{x}} = \frac{1}{k} \sum_{j=1}^{k} \bar{\boldsymbol{x}}_j \tag{11.6}$$

The value of B_k should be as large as possible, but this again depends on k.

The **Calinski-Harabasz-Index** is a normalized ratio of the two quantities,

$$\mathrm{CH}_k = \frac{B_k/(k-1)}{W_k/(n-k)}.$$

The **Hartigan-Index** is defined as

$$\mathrm{H}_k = \log \frac{B_k}{W_k}.$$

The optimal number of clusters is chosen for the biggest value of one of these indices over all k considered.

11.2 Discriminant Analysis

Like cluster analysis, this multivariate method aims to group data. There is, however, one essential difference to cluster analysis: With cluster analysis, it is not known which observations belong to which groups; it is not even clear how many groups exist in the data, and whether there is any group structure at all. In the case of discriminant analysis, on the other hand, one knows the class affiliations of the observations, at least those observations from a training data set. You know, for example, that certain measurements come from sick and healthy people, and thus you know both the number of groups and the group membership. Now one would like to learn a rule that allows new observations to be assigned to the classes on the basis of the same measured characteristics (variables). These new observations form the so-called test data set.

Given are the training data **X**, an $n\,x\,p$ data matrix with the observations $x_1, x_2, \ldots, x_n$. The observations come from k groups, the group memberships are known, and the numbers of observations in the groups are $n_1, \ldots, n_k$, with $n_1 + \ldots + n_k = n$. The variable **G** describes the predicted group membership, and can thus assume a value from the set $\{1, \ldots, k\}$.

It is also assumed that the underlying distribution of the groups is known. Here we assume multivariate normal distribution with parameters μ_j and $\sum_j$, for $j = 1, \ldots, k$; the density function is:

$$\boldsymbol{\phi}_j(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^p \det(\boldsymbol{\Sigma}_j)}} \exp\left\{ -\frac{(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)}{2} \right\}. \qquad (11.7)$$

Finally, we also assume an *a-priori* probability p_j for each group, where $p_1 + \ldots + p_k = 1$. For example, if group 1 describes sick people and group 2 healthy people, then typically $p1 < p2$ because it is (hopefully) less likely to encounter this disease in the population.

With this information, the conditional (*a-posteriori*) probability can be determined according to Bayes' theorem: Given the observation x, the conditional probability that the variable **G** takes on the value j is given by

$$\mathrm{P}(G = j|\boldsymbol{x}) = \frac{\boldsymbol{\phi}_j(\boldsymbol{x})p_j}{\sum_{l=1}^{k} \boldsymbol{\phi}_l(\boldsymbol{x})p_l}. \tag{11.8}$$

The denominator in equation (11.8) is the same for every group, and therefore one can directly compare the *a-posterior* probabilities for two groups. If $P(G = j|x) > P(G = l|x)$, then $\boldsymbol{x}$ would be assigned to the j-th group. In other words: $\boldsymbol{x}$ is assigned to the j-th group (and not the l-th group) if applies

$$\log \frac{\mathrm{P}(G = j|\boldsymbol{x})}{\mathrm{P}(G = l|\boldsymbol{x})} = \log \frac{\boldsymbol{\phi}_j(\boldsymbol{x})p_j}{\boldsymbol{\phi}_l(\boldsymbol{x})p_l} = \log \frac{\boldsymbol{\phi}_j(\boldsymbol{x})}{\boldsymbol{\phi}_l(\boldsymbol{x})} + \log \frac{p_j}{p_l} > 0. \tag{11.9}$$

11.2.1 Linear Discriminant Analysis (LDA)

We make another assumption, namely that $\sum_{\mathbf{1}} = \ldots = \sum_{\boldsymbol{k}}$ holds. The covariances of all groups are thus assumed to be the same, and we now denote this covariance by $\sum$. One can now insert equation (11.7) into (11.9) and finally get the **linear one**

Discriminant rule:

$\boldsymbol{x}$ is assigned to the j-th group (and not to the l-th group) if $\delta_j(\boldsymbol{x}) > \delta_l(\boldsymbol{x})$, where

$$\delta_j(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j - \frac{1}{2} \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j + \log p_j \tag{11.10}$$

the so-called *linear discriminant function* represents the j-th group.

If $k > 2$ groups are given, then for a new test observation $\boldsymbol{x}$ the linear discriminant function is calculated for each group, and $\boldsymbol{x}$ is then assigned to the group for which the value of the linear discriminant function is the greatest.

One immediately recognizes that the resulting decision rules are linear, because the discriminant function in (11.10) is linear in $\boldsymbol{x}$.

In practice, the parameters in equation (11.10) must first be estimated from the training data. As an estimate for p_j one can take n_j/n, provided that the training data reflect the population. Let I_j be the index set for the observations of the j-th group of the training data (see cluster analysis). The arithmetic mean (vector!) of the training data of the j-th group can be used as an estimate for $\boldsymbol{\mu}_j$,

$$\hat{\boldsymbol{\mu}}_j = \bar{\boldsymbol{x}}_j = \frac{1}{n_j} \sum_{i \in I_j} \boldsymbol{x}_i \quad \text{für } j = 1, \ldots, k, \tag{11.11}$$

see also equation (11.1). The common covariance matrix can be estimated by a "pooled" covariance,

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{S}_{pooled} = \frac{1}{n-k} \sum_{j=1}^{k} \sum_{i \in I_j} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)^T. \tag{11.12}$$

11.2.2 Quadratic Discriminant Analysis (QDA)

If one cannot (and would not like to) assume the equality of the covariances, then when inserting equation (11.7) into (11.9) the resulting expression becomes more complicated. The so-called *quadratic discriminant functions* are then obtained

$$\delta_j^{(q)}(\boldsymbol{x}) = -\frac{1}{2}\log(\det(\boldsymbol{\Sigma}_j)) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j) + \log p_j \qquad (11.13)$$

for $j = 1, \ldots, k$, which are quadratic in $\boldsymbol{x}$. The new observation $\boldsymbol{x}$ is then assigned to the group for which the value of the quadratic discriminant function is the greatest.

Again, for the practical application of the rule, the parameters must first be estimated from the training data. In contrast to the LDA, you do not need a joint estimate of the covariance, but rather individual estimates, e.g. through the sample covariances

$$\hat{\boldsymbol{\Sigma}}_j = \boldsymbol{S}_j = \frac{1}{n_j - 1}\sum_{i \in I_j}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)^T, \qquad (11.14)$$

but robust estimates using the MCD estimator were also possible. Note that, compared to LDA, you have to estimate significantly more parameters here, which can lead to an "over-adaptation" to the training data and lead to a poorer prognosis of group membership.

As an example, let's look at the iris data again. In contrast to the cluster analysis, this time the group information for the training data is used to create the discriminant functions, and then used to evaluate the method on the basis of the test data. LDA can be carried out as follows:

```
# R code LDA (QDA)
data(iris)                                  # iris data
X <- iris[,1:4]                             # Measurements
grp <- iris[,5]                             # group
grpn <- as.numeric(grp)                     # Group numbers
set.seed(123)                               # always causes the same random selection
n <- nrow(iris)                             # Number of observations
train <- sample(n,round(n*2/3))             # Training data indexes
test <- (1:n)[-train]                       # Test data indexes
library(MASS)
res <- lda(X[train,],grp[train])            # LDA on training data
### for QDA just use the command “qda()”
res.pred <- predict(res,X[test,])           # Forecast for test data
table(grp[test],res.pred$class)             # Comparison with reality
#               setosa      versicolor      virginica
# setosa        14          0               0
# versicolor    0           17              2
# virginica     0           0               17
#
plot(res,abbrev=2,col=grpn[train])          # linear discriminant functions
points(res.pred$x,col=grpn[test],pch=as.numeric(res.pred$class))        # Test data
```

Figure 11.6 shows a projection of the data into the space of the linear discriminant functions. The symbols with the texts are the training data, and the color corresponds to their group membership. The test data are represented by non-text symbols; the color is the real group, the symbol the predicted group. If you have two red observations, you can see a wrong symbol.

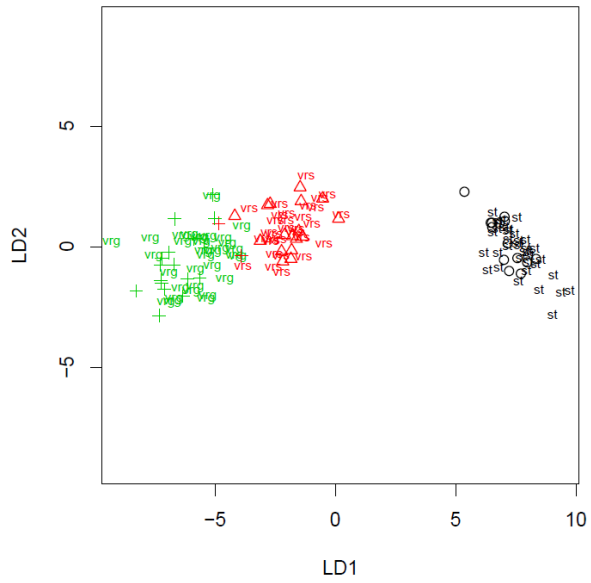


Figure 11.6: Results of LDA on the iris data.