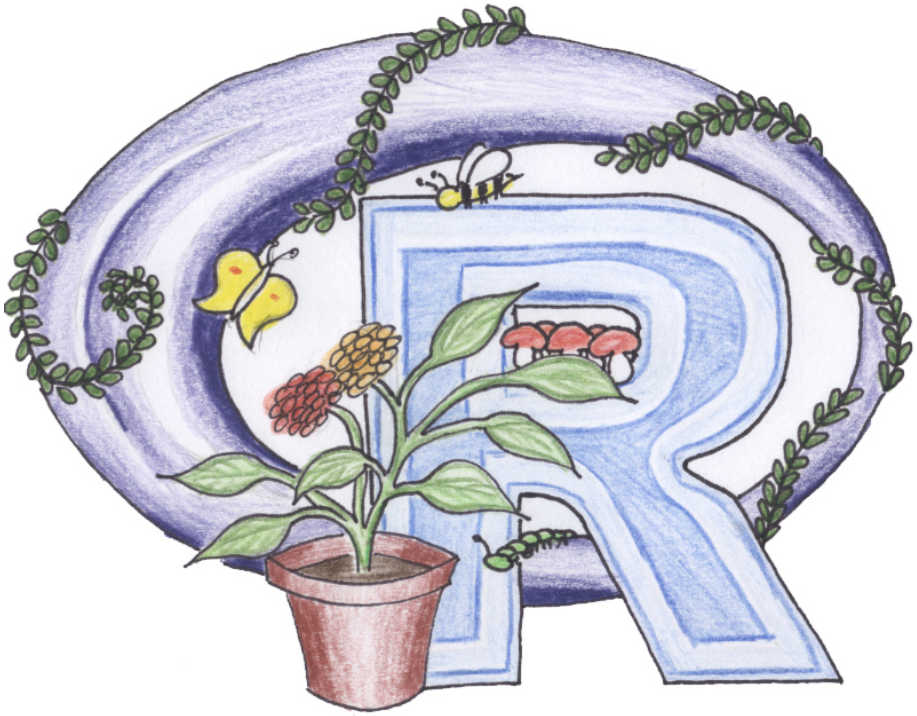
R Handbook for Biostatistics

An Introduction for Students of Horticulture,Plant Biotechnology and Biology

by Katharina J. Hoff



Submitted as a Bachelor Thesis at University of Hannover, 2005.
Modified at University of Greifswald in January 2019.

SUPERVISORS

Prof. Dr. L. A. Hothorn, University of Hannover
Universitetslektor J.-E. Englund, SLU Alnarp

Contents

Chapter 1

Introduction

1.1 History

In 1976, John Chambers and his colleagues (Bell Laboratories) began to develop a programming languages called S. The new language should provide the possibility to program with data. Since then, S has been improved continuously.

The S language has been implemented in several ways. The commercial version, S-Plus, has been commonly used for data analysis by scientists.

Ross Ihaka and Robert Gentleman (University of Auckland, New Zealand) started working on an open source implementation that is similar to S. It is called – referring to the initial letters of their Christian names – R. R (R Development Core Team, 2004a) is covered by the GNU General Public License (Stallman, 1991). That means, access to R as program and source code is free for public, respecting certain conditions[1]. Based on this license, R is permanently improved by a worldwide community. Today, it represents a powerful system that meets the requirements of scientists in Horticulture, Biology and Agriculture on statistics very well. In comparison to S-Plus, there are no license fees to be paid.

1.2 Bachelor Thesis Problem

The topic of my thesis is called *Writing of an R-Manual for Biometry*. This issue has been announced because there exists a demand for a manual considering the special needs of horticultural scientists, biologists and agricultural engineers. Many of the hitherto existing books about R (e.g. Introductory Statistics with R (Dalgaard, 2002)) are very good guidelines showing the functions of basic statistics on general examples. But those examples might be too abstract for a student of horticulture or plant biotechnology. Some functions that are very interesting with regard to field experiments, e.g. for multiple comparison tests, are still missing in most books.

This manual is adapted to the standard of knowledge of an undergraduate student in a biological sciences. Ideally, a lecture in basic statistics should go along with studying this book. Many horticultural and agricultural examples demonstrate the usage of different R functions in scientific practice.

[1] §1 You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

1.3 Reasons to Use R

In comparison to S-Plus, R is for free and powerful in almost the same manner. Big parts of source code written in S-Plus are running on R without any problems. But an undergraduate student of Horticulture might not know S-Plus at all. The typical undergraduate in Plant Sciences is rather used to the Microsoft Office Suite, asking why he should not evaluate his experiments with Excel. There are a number of reasons to move on:

- R does not represent itself with an intuitive graphical surface and is furthermore command line oriented. On the other hand, this gives full control of the actions to the user. All parameters can be set individually and the provided help system assists in keeping a good overview about existing parameters.
- An R test output is far more advanced and comprehensive than the result of any Office Program. Confidence intervals, quantiles et cetera are usually automatically calculated along with p-value, degrees of freedom and many other values.
- In comparison to Office Programs, R is more powerful regarding huge data sets and complicated commands (e.g. nested functions).
- The knowledge of mathematical formulas for statistical procedures is not an imperative necessity for the evaluation of data with R.
- R is an object oriented programming language. This has many advantages. It is e.g. possible to produce a graph with confidence intervals of an object containing the test output of `simint()` using the single, short command `plot(object.simint)`.
- R is platform independent. It may be used on Unix, Linux, Windows and MacOS.
- The usage of R is not more complicated than the usage of a GUI based program. Commands are typed into the command line but the command structure is logical and therefore easy to learn.
- Another advantage is the integration into the text markup language LaTeX by the Sweave tools. LaTeX is increasingly popular among scientists due to its clear structure. Together, LaTeX and R are offering a working platform that contains all tools for evaluation and publication of scientific experiments (Gentleman, 2005).
- R is able to import Microsoft Excel data sheets (`RODBC` package). The package `foreign` is additionally supporting the usage of data created by S, SAS, SPSS, Stata et cetera.

GUI refers to *Graphical User Interface*.

These arguments shall convince students to start working with R.

1.4 Download and Installation

Packages prepared for installation are provided for the operating systems Linux, Windows and Mac OS. References for the self compilation of source code and the installation on Unix, Windows and Mac OS are given in *R Installation and Administration* (R Development Core Team, 2004b).

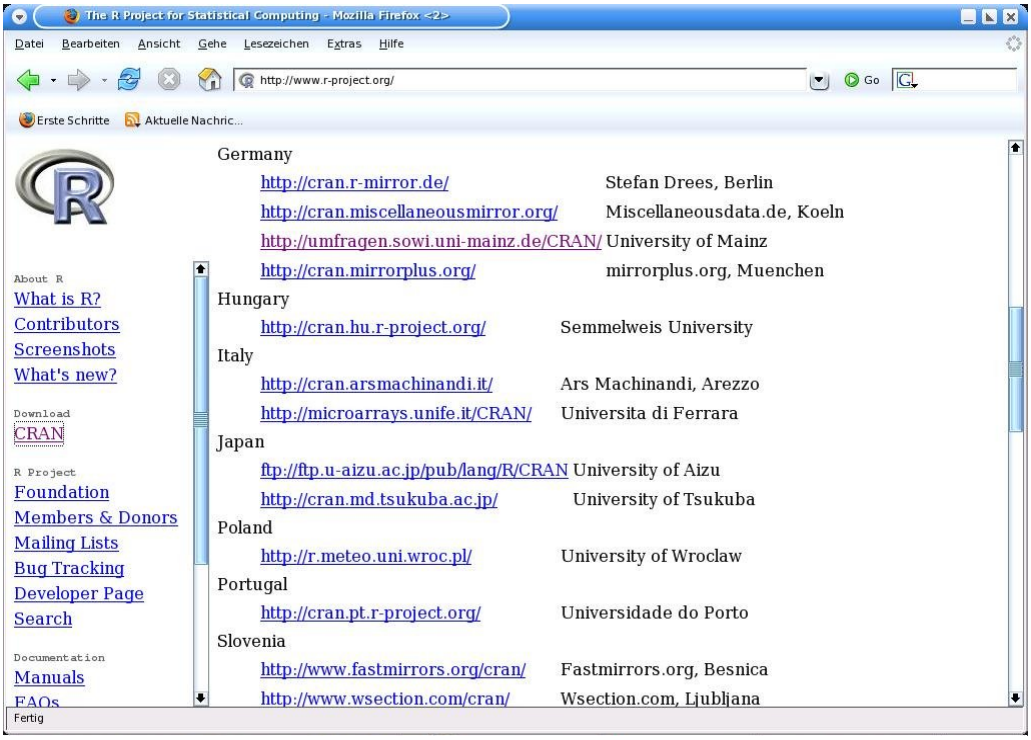


Figure 1.1: Selection of a closely located mirror with CRAN.

1.4.1 Download

If you are not familiar with the installation of programs, please remember the directory where you save the *.exe or *.rpm package!

R is available at CRAN (Comprehensive R Archive Network) on the website http://www.R-project.org. In order to minimise the transfer time, a closely located Mirror should be selected (figure 1.1). Download the newest version of the base package for your respective operating system (an *.exe file for Windows or an *.rpm package for rpm supporting Linux systems) in a directory on your local computer.

1.4.2 Installation on Windows

The installation will be started by a double click on the downloaded *.exe file. The Installation Wizard will ask for the target directory of the installation. The next step is the selection of R components (Figure 1.2). During the further commencing installation, it will be asked in which folder of the start menu an R icon shall be created, which registry entrances shall be written and if a desktop icon is wished. Take a configuration of your choice and click on Next >, finally. R is now being installed on your computer.

Console means the command line inside the running program R.

Subsequently, the program can be called by a click on the desktop icon, the link in the start menu or with a double click on the file `R/bin/Rgui.exe`. End R either by **File** submenu **Exit** or by typing `q()` in the R console.

1.4.2.1 Installation of Add-on Packages

The R base system does not include all packages. I recommend the installation of `pastecs`, `exactRankTests`, `multcomp`, `mvtnorm`, `car`, `Rodbc`, `Biobase` (Linux only, available at http://www.bioconductor.org/repository/release1.5/package/html/index.html) and `multtest` to solve all problems given in this book.

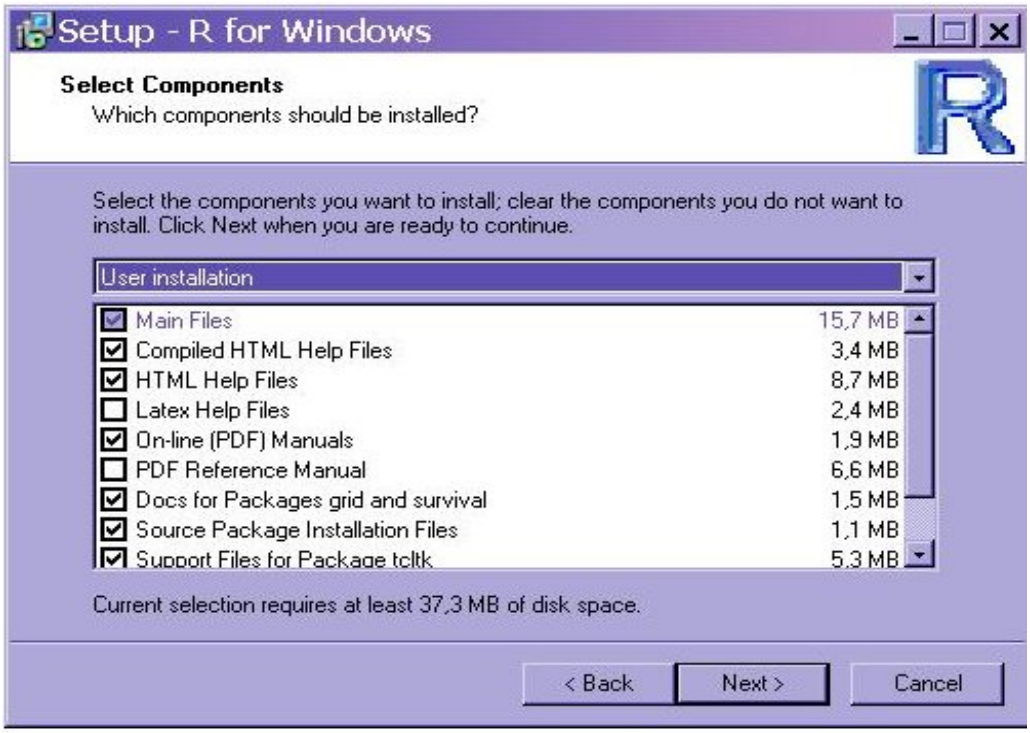


Figure 1.2: Selection of R components on Windows. The standard configuration should be convenient for most users.

An internet connection is required for the installation of add-ons. You can start the installation process by clicking on the subentry **Install package(s) from CRAN...** in the **Packages** menu (Figure 1.3). A popup windows opens, presenting a list of available packages. Select the package of your choice and confirm with **OK**. The respective archive will be downloaded, unpacked and installed automatically. Afterwards, R asks the following question: `Delete downloaded files (y/N)?`. You can delete them with `y` (yes) because those files are only the sources for the preliminarily accomplished installation.

For usage of an add-on, you have to load it with the command `library(package name)` into your running R-system.

The **Command Line** (terminal window) is the **Shell** on Linux. It is a terminal program for executing commands. In most of the cases, you will find it as a Shell-icon on your graphical surface.

1.4.3 Installation on Linux

It is necessary to be logged in as **root**[2] for the R installation on Linux. On Suse-Linux, a click on the *.rpm packages in the Conquerer starts a simple GUI based installation with Yast.

If your Linux-Distribution does not contain a graphical installation manager, you may install R by typing the following command in the Shell:

```
rpm -ih /path/to/package/packagename
```

After a successful installation, R can be called in the terminal window (Shell) by typing `R`. Typing `q()` in the R-Console (= terminal window while R is running) stops the program.

A **Distribution** is a Linux version published by a company or a private association. A distributor is usually selling some kind of service and not the program itself which is open source and covered by the GNU Public License, anyway.

[2] If you install your package via a GUI, the root password will be requested automatically. Using the Shell, you have to change user with the command `su root` manually.

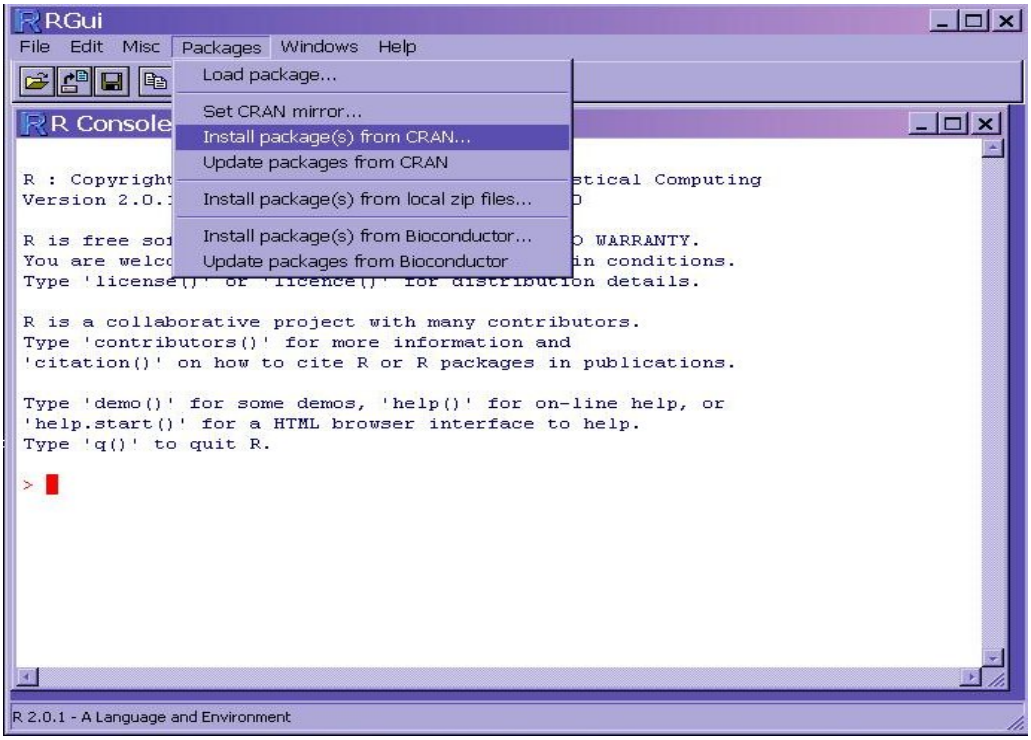


Figure 1.3: Installation of add-on packages with CRAN on Windows.

The R base package does presently not contain an error free running GUI. The package `gnomeGUI` promises to be a new R-Console for GNOME if the appropriate GNOME libraries are installed. However, I was not able to install this package myself (possibly due to an old GNOME system).

1.4.3.1 Installation of Add-ons

As mentioned in section 1.4.2.1, the R base installation does not contain all packages. Add-on packages can be installed easily by using the command line (change user to **root** is necessary).

After downloading the appropriate package from CRAN manually, type the following command in the Shell (**not** into the R-Console!) (R Development Core Team, 2004b):

```
R CMD INSTALL -l /path/to/library /path/to/packagename.tar.gz
```

The path to library depends on your system. On Suse-Linux, it is:

```
/usr/lib/R/library
```

It is possible to leave out the path to the package if you are already inside the correct directory[3]. Indicating the full name of the package is sufficient, then.

There is also the possibility of an installation through the R-Console if the computer is actively connected to internet. Therefore, first set the option `CRAN` as follows:

```
> options(CRAN = "http://cran.us.r-project.org")
```

[3]Change directory with `cd /path/to/downloaded/package/`

The command

```
> install.packages(packagename)
```

installs the appropriate package, afterwards (R Development Core Team, 2004b).

Remember to include the add-on with `library(package.name)` before usage.

1.4.4 Documentation and Help System

Entering `help.start()` in the R-Console will open a Browser window on Linux, presenting different manuals and documentations. On Windows, the help pages are opening within the GUI. Handbooks are usually included in the R installation. If they are missing because you excluded them during a user defined installation, an active internet connection will be required.

The command `?function()` or `help(function)` calls for the help of individual functions.

On **Linux**, most help pages are opening within the terminal window. You navigate there with the arrow keys and return to the R command line by typing `q`.

If you do not know the name of the function you are looking for, try searching for a related word:

```
help.search("search.item")
```

It is possible to call examples for a certain function with `example(function)`. The simple entry of a function name will search for this function and return if it exists on the current system.

1.4.5 Editors

A **text editor** is a computer program for entering, processing and saving plain text. It is reasonable to use an editor while working with R if you want to recall certain preliminarily used functions after a longer period of time without complications.

For the usage of the standard Windows editor or another simple editor, you have to open the editor as well as R and arrange them somehow parallel on the screen. Type your commands into the editor first and copy & paste them into R. Finishing your session, remember to save the editor document as a .txt file somewhere (remember the directory and file name!).

There are many more advanced editors available. Those are able to do much more than only plain text editing. On Windows, WinEdt turned out to be a useful R editor (available at http://www.winedt.com). It can be adjusted in a way that you only have to press a button to hand marked source code over to the R machine. Emacs (available at emacs) combined with ESS (Emacs Speaks Statistics, available at http://ess.r-project.org is offering a similar service which is even platform independent. Both editors provide the user with a colorful highlighting for the source code.

1.5 Basics

This section has been written following the tutorial script for Biometry 1 (Froemke, 2004). A full understanding of the terminology is not required after first reading. Nevertheless, later chapters are built on the content of this section and it might help you to flip back for certain parts.

1.5.1 Handling of the Command Line

Commands are always typed after > in the R command line. A command is verified by pressing the **ENTER** or **RETURN** key. R is calculating the input and gives an output if available. The arrow keys ↑ and ↓ provide a navigation through previously used commands. **POS1** sets the cursor to the beginning of a line, **END** sets the cursor the end of a line.

Comments are used to explain the source code for other people and yourself. Comments will be ignored during compilation.

Comments are marked with Hash (#).

Blanks are usually ignored. `4 + 7` has the same meaning for R as `4+7`. However, blanks are not allowed to be used inside a command: `x <- 3` ⇒ three is alloted to x, but with a blank within the < and - it is getting the meaning " x is smaller than -3?".

Line breaks. If a command is overlapping a single line, `+` will indicate that the same command is continued in the next line. This character does **not** have to be typed! If a command is not complete, there will also show up a `+` in the next line. You have the possibility to complete your command after this sign. In many cases, brackets are missing.

1.5.2 Pocket Calculator, Objects and Functions

R can be used as a simple pocket calculator for addition, subtraction, multiplication and division. Also logarithms et cetera are calculated easily:

```
> 4+7

[1] 11

> log(2)

[1] 0.6931472

> exp(0.6931472)

[1] 2


> 30/6 # Take care with division. Double dots will lead to the output of
 all natural numbers from 30 to 6.

[1] 5

> log(-1)

[1] NaN
Warning message:
NaNs produced in: log(x)
```

Attention! `log()` is calculating the natural logarithm, not the logarithm to the base 10!

`NaN` stands for "not a number". Missing values are indicated by `NA` (not available).

R is writing the result into a **vector** (see section 1.5.4.1), that is containing only one single element at the position [1] in the above mentioned examples. But you can also get a vector with many elements by calling the natural numbers from 30 to 6:

```
> 30:6

 [1] 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10  9  8
 + 7  6
```

A vector can be saved into an object by using the <- command. An object is recalled by its name and it might be used in other calculations and functions directly:

```
> a<-89
> b<-45
> result<-(a+b)^2
> result

[1] 17956
```

Objects will be overwritten without any warning. A definite name avoids this to a certain extent, e.g. *binom.formula.of.a.b* instead of *result*. Even functions can be overwritten with object names easily. The safest method is therefore to enter the name of interest into the R-Console. If there is a function with this name existing, it will be returned. Some more hints for choosing an appropriate object name:

- Object names are not allowed to begin with a number and it is not recommended to start with a dot,
- dot (.) and underline (_) are permitted but other special characters as e.g. ~, @, !, #, %, ^, & are not allowed,
- upper and lower cases have to be considered.

A **function** is the implementation of a method, it gives a result value.

Objects are processed by functions. A function consists of its unique name and the following parentheses which can include different arguments. The function `objects()` for example lists all existing objects. The argument `pattern` can specify a selection criterion, which means that

```
> objects(pattern="example")
```

prints only those objects which contain the character `example` in their name. You can get more information about the function `objects()` by typing `?objects()`.

Section 1.4.4 gives instruction for the R help system.

The function `rm()` deletes objects.

1.5.3 Data Types

Objects in R can contain different types of data. Important for the examples given in this manual are the following types:

Numeric: Numbers. You can only calculate with numeric objects.

Character: Character strings are commonly used for group and variable names.

Logical: has the two values, `TRUE` and `FALSE`. Requests often have a logical output:

```
> a <- 23
> b <- "Keine Zahl"
> is.numeric(a)

[1] TRUE

> is.numeric(b)

[1] FALSE
```

Factor: Categorical data, e.g. traffic lights in the colors red, orange and green. The value of a factor is named *level*. Factors can be generated from numerical and character objects. In the following example, a vector is transformed into a factor. Calling the factor, content and levels are printed. It is also possible to get the levels printed by the function `levels()` .

```
> traffic.lights.vector <- c("green", "red", "green", "yellow", "yellow")
> traffic.lights.factor <- factor(x=traffic.lights.vector)
> traffic.lights.factor

[1] green  red    green  yellow yellow
Levels: green red yellow

> levels(traffic.lights.factor)

[1] "green"  "red"    "yellow"
```

The levels occur in alphabetical order. Nevertheless, it is of importance for certain statistical procedures to sort them by another criterion. A new order can be given with:

```
> affection.factor <- factor(c("none","few","too many", "few", "many",
+ "too many"))
> sorted.affection.factor <- ordered(x=affection.factor,
+ levels=c("none","few","many","too many"))
> sorted.affection.factor

[1] none     few      too many few      many     too many
Levels: none < few < many < too many
```

1.5.4 Data Input and Output

Data might be saved in the following structures in R: vector, matrix, list and data frame. An R output occurs on calling the object or as result of a function (usually a list).

1.5.4.1 Vector

Vectors are a one dimensional data structures containing only one data type, e.g. numeric or character. Vectors with only one element can be created by simple allocation (see section 1.5.2):

```
> vec.1 <- "cucumber"
> vec.1

[1] "cucumber"
```

To create a vector containing more than one element, the function `c()` concatenates several elements. (`c()` can also concatenate only one single element, of course.)

```
> vec.2 <- c(2,3,4,5,6,3.4)
> vec.2

[1] 2.0 3.0 4.0 5.0 6.0 3.4

> vec.3 <- c("cauliflower", "cucumber", "tomato")
> vec.3

[1] "cauliflower" "cucumber"    "tomato"
```

If different data types are posed in one vector, R will convert them all into a common type. In this example, R is changing all numerical entries into characters as soon as a single entry with the type character occurs:

```
> vec.4 <- c(1:4,10.5,"flower")
> vec.4

[1] "1"      "2"      "3"      "4"      "10.5"   "flower"
```

`seq()` generates sequences at constant intervals:

```
> vec.5 <- seq(from =1, to =5, by = 0.5)
> vec.5

[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

`rep()` repeats elements in vectors and lists:

```
> vec.6 <- rep(x=c("A","B","C"), times = 3)
> vec.6

[1] "A" "B" "C" "A" "B" "C" "A" "B" "C"

> vec.7 <- rep(x=c("A","B","C"), each = 3)
> vec.7

[1] "A" "A" "A" "B" "B" "B" "C" "C" "C"
```

It is possible to name vector elements. It is important that the number of names is equal to the number of elements:

```
> vec.8 <- seq(from=1,to=9,by=2)
> vec.8

[1] 1 3 5 7 9

> names(x=vec.8)<-c("a","b","c","d","e")
> vec.8

a b c d e
1 3 5 7 9
```

`length()` and `mode()` return the length and mode of vectors, matrixes, lists and data frames. The function `sort()` sorts a vector by size or alphabetically. Acceding is the default value but the argument `decreasing = TRUE` inverts the order.

1.5.4.2 Matrix

In contrast to a vector, a matrix has two dimensions. However, it can still only contain one data type per matrix. A matrix is created with the functions `cbind()` (column bind), `rbind()` (row bind) or `matrix()`. The arguments `ncol` or rather `nrow` indicate the column/row numbers for the function matrix (data are always entered horizontally into the matrix):

```
> mat.1 <- cbind(1:3, c(4,3,6))
> mat.1

     [,1] [,2]
[1,]    1    4
[2,]    2    3
[3,]    3    6

> mat.2 <- rbind(1:3, c(4,3,6))
> mat.2

     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    3    6

> mat.3 <- matrix(data=c("A","B","C","D","E","F"), nrow=3)
> mat.3

     [,1] [,2]
[1,] "A"  "D"
[2,] "B"  "E"
[3,] "C"  "F"

> mat.4 <- matrix(data=c("A","B","C","D","E","F"), ncol=3)
> mat.4
```

```
     [,1] [,2] [,3]
[1,] "A"  "C"  "E"
[2,] "B"  "D"  "F"
```

Names for columns and rows can be set with the functions `colnames()` or `rownames()` (this is also a useful tool for data frames):

```
> colnames(mat.2) <- c("one","two","three")
> rownames(mat.2) <- c("A","B")
> mat.2

  one two three
A   1   2     3
B   4   3     6
```

A matrix can be transposed with the function `t()`. The function `dim()` returns the dimensions (number of rows and columns).

1.5.4.3 List

A list is an assemblage of objects which contain e.g. a test output. It is possible to combine several data types in one list:

```
> vec.numeric <- c(1:6)
> mat.character <- rbind(c("tomato","cucumber", "iceberg","pepper",
+ "egg fruit","cauliflower"), c(1,4,6,2,7,9), c("D5","A1","E9","G3",
+ "B5","P1"))
> list.1 <- list(example.vec=vec.numeric, example.mat=mat.character)
> list.1

$example.vec
[1] 1 2 3 4 5 6

$example.mat
     [,1]     [,2]       [,3]      [,4]     [,5]        [,6]
[1,] "tomato" "cucumber" "iceberg" "pepper" "egg fruit" "cauliflower"
[2,] "1"      "4"        "6"       "2"      "7"         "9"
[3,] "D5"     "A1"       "E9"      "G3"     "B5"        "P1"
```

Naming and adding of list elements:

```
> names(list.1)[2] <- "new name"
> list.1$new.element <- c(9,8,7,6,5)
> list.1

$example.vec
[1] 1 2 3 4 5 6

$`new name`
     [,1]     [,2]       [,3]      [,4]     [,5]        [,6]
[1,] "tomato" "cucumber" "iceberg" "pepper" "egg fruit" "cauliflower"
[2,] "1"      "4"        "6"       "2"      "7"         "9"
[3,] "D5"     "A1"       "E9"      "G3"     "B5"        "P1"
```

```
$new.element
[1] 9 8 7 6 5
```

The function `names()` returns the names of list and data frame elements.

1.5.4.4 Data Frame

The data frame is a two dimensional data structure that might contain different data types in separated columns. It is most frequently used in biometry. All columns must have the same length:

```
> x <- c(1:6)
> x[2] <- 12
> treatment <- rep(x=c("A","B"), each = 3)
> my.frame <- data.frame(group=treatment, value=x)
> my.frame
```

```
  group value
1     A     1
2     A    12
3     A     3
4     B     4
5     B     5
6     B     6
```

The function `transform()` serves for editing a data frame:

```
> new.frame<- transform(my.frame,evaluation=c("low",NA,"medium","medium",
+ "medium","high"))
> new.frame
```

```
  group value evaluation
1     A     1        low
2     A    12       <NA>
3     A     3     medium
4     B     4     medium
5     B     5     medium
6     B     6       high
```

1.5.4.5 Subsets

The command `vectorname[positionnumber(s)]` allows access to the single values of vectors.

```
> vec.8[2]
```

```
b
3
```

```
> vec.8[2:4]
```

```
b c d
3 5 7
```

```
> vec.8[c(1,3,4)]
```

```
a c d
1 5 7
```

The command can be applied on a matrix similarly but both, row and column numbers, have to be indicated in this case (`matrixname[rownumber(s),columnnumber(s)]`) . The respective matrix data is returned as a vector:

```
> mat.3[1,2]
```

```
[1] "D"
```

```
> mat.3[c(2,3),2]
```

```
[1] "E" "F"
```

The command `listname[elementnumber]` returns a new list containing the appropriate element. The alternative `listname[[elementnumber]]` returns the element in its original data type (e.g. as a vector):

```
> list.1[1]
```

```
$example.vec
[1] 1 2 3 4 5 6
```

```
> list.1[[1]]
```

```
[1] 1 2 3 4 5 6
```

Calling columns, rows and single values from data frames works as described for matrix. `objectname$elementname/columnname` offers another alternative for calling objects from lists and data frames:

```
> list.1$example.vec
```

```
[1] 1 2 3 4 5 6
```

```
> my.frame$group
```

```
[1] A A A B B B
Levels: A B
```

If elements of lists and data frames are called frequently, they can be attached temporarily with the function `attach()`. The element is thereafter called simply by its name or column header. It is of high importance to detach the object afterwards in order to avoid conflicts between different attached data sets (`detach()`):

```
> attach(list.1)
> example.vec

[1] 1 2 3 4 5 6

> detach(list.1)
```

The function `subset()` returns subsets which fulfill defined criteria, e.g. all elements in `my.frame`, that are greater than 3:

```
> subset(x = my.frame, subset = value > 3)

  group value
2     A    12
4     B     4
5     B     5
6     B     6
```

A file written in the **flat file format** contains the entire information for a single entry in each row, e.g. block: A, repetition: 3, plant height: 5.

1.5.5 Import and Export of Data Sets

On Windows, the package RODBC assists in the import of Excel data sheets. The source file, an Excel sheet in this case, should be written in the flat file format:

```
> library(rodbc)
> full.data <- odbcConnectExcel("filename.xls")
> sqlTables(full.data)
> data <- sqlQuery(full.data, 'select * from "Sheet1$"')
> odbcCloseAll()
```

In German versions of excel, a data sheet is indicated with the German word **Tabelle** instead of **Sheet**.

The full directory name to the target file is omitted if the appropriate directory has been set previously by clicking on the submenu **Change Directory** in **File** (`setwd()` serves the same purpose).

Another handy alternative for data import on Windows is the Copy & Paste method. Therefore, the data set is fully marked and copied with **Ctrl C** and afterwards recalled with the following command in the R console:

```
> data <- read.table(file("clipboard"), header = TRUE)
```

`header` defines whether the original dataset has a header (set on `TRUE`) or if there is no header to be imported (default value `FALSE`). If the default value of a parameter is used, the argument does not have to be indicated in the command.

On Linux, neither the import of Excel files nor the Copy & Paste method works properly. An alternative that works on all platforms is therefore the import of *.txt or *.csv files. The excel sheet can either be saved as a *.txt directly from excel or it might be copied into a text editor and be saved as a *.txt from there. The import command is then:

```
> data <- read.table(file = "/path/to/file/filename.txt", header = TRUE,
+ sep = "\t", dec = ",")
```

The argument `sep` specifies the separator for the different columns. Tabulator is the default value.

`dec` defines if a dot or a comma is used as decimal sign. The default value in R is the international dot. In most European countries, commas are commonly used.

The function `write.table()` saves datasets from R in an external *.txt file:

```
> write.table(x = my.frame, file = "/path/to/file/filename.txt",
+ sep = "\t", dec=".", col.names = TRUE)
```

`col.names` has the same function as `header` in `read.table()`, it defines whether there exist column names (default) or not.

1.5.6 Workspace Management

The practical navigation through previously used commands with the arrow keys (section 1.5.1) gets lost with a restart of R if the workspace has not been saved in a known directory. The following functions can be used to save and recall the command history:

```
> savehistory(file = "filename.Rhistory")
> loadhistory(file = "filename.Rhistory")
```

On Windows, the GUI subentry **Save workspace...** in the menu **File** saves all currently used objects. They can be recalled with the subentry **Load workspace...**. On all platforms, the commands `save()` and `load()` serve the same purpose:

```
> save(list = ls(), file = "filename.RData")
> load(file = "filename.RData")
```

On Windows, the produced source code of a session can be saved in a *.txt file by clicking on **Save to file...** in the menu **File**. On all platform the command `save.image()` saves source code in e.g. a *.txt file.

Regarding the process of saving and loading files (also import and export of data sets), the function `setwd()` is important for setting a working directory where files are saved or loaded:

```
setwd("/directory")
```

This functionality is also offered through the GUI on Windows: **File – Change Directory**. The function `getwd()` calls the current directory.

The usage of an editor is very helpful regarding clarity and long term backup (see section 1.4.5).

✎ Exercise 1

1. Calculate in R the second binomial formula

   $$(a - b)^2$$

   using a = 12 and b = 7. Create the objects a and b! Save the result in an object with a definite name!

2. Create an object containing the reverse running numbers from 28 to -34!

3. Call help for the function `objects()` and close it correctly! Use the function `objects()` to see all existing objects! Remove object a!

4. Create a data frame in the flat file format for table 1.1!

Batch	Culture Solution	Plant 1	Plant 2	Plant 3
A	Complete	1172	750	784
B	Lacking magnesium	67	95	59
C	Lacking nitrogen	148	234	92
D	Lacking micro-nutrients	297	243	263

Table 1.1: A plant nutrition experiment with sunflowers in water culture. End point is the dry weight in (mg) (Bishop, 1980, p. 1).

Chapter 2

Descriptive Statistics

2.1 Basic Functions

A vector is created in order to demonstrate the basic functions of descriptive statistics:

```
> data <- c(34,5,23,17,23,19,21,12,25,22,19,19,12,22,17)
```

`mean()` calculates the mean of `data`:

```
> mean(data)

[1] 19.33333
```

The usage of `sd()` for standard deviation, `median()`, `var()` for the variance, `IQR()` for the interquartile_range, `min()` for the minimum, `max()` for the maximum, `range()` for minimum and maximum, `diff()` for the range and `sum()` is identical.

The variation coefficient is returned with the following command:

```
> var.coeff <- sd(data)/mean(data)
> var.coeff

[1] 0.3429134
```

The function `quantile()` calculates per default the 0%, 25%, 50%, 75% and 100% quartile. It is possible to specify the quantiles with `probs`:

```
> quantile(data, probs=c(0.25,0.75)) # calculates the 25 and
+ 75 percent quartiles

 25%  75%
17.0 22.5
```

The function `summary()` returns a summary of the most important statistics for a sample:

```
> summary(data)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00   17.00   19.00   19.33   22.50   34.00
```

2.2 Loops with `tapply()`

The looping function `tapply()` offers the possibility of a fast and easy statistic analysis of flat file datasets with different categories (e.g. treatments).

```
tapply(X, INDEX, FUN = NULL, ...)
```

`X` stands for the response variable, e.g. as a column in a data frame. `INDEX` identifies the grouping column or vector containing the different levels (e.g. treatments).

`FUN` specifies the applied function of descriptive statistics, e.g. `sum`, `mean`, `var` or `IQR`.

`tapply()` returns an array with the calculated results.

2.2.1 Example Soil Respiration (1)

2.2.1.1 Experiment

Plant growth is influenced by the microbial activity in the soil. Soil respiration is an indicator for this activity. Soil samples from two characteristic areas in the forest (gap = "clearing and growth" and growth = "dense tree population") have been analyzed regarding their carbon dioxide output in an experiment. The amount of excreted CO_2 has been measured in mol CO_2 g^{-1} soil hr^{-1} (see data 2.1) (Fierer, 1994) cited according to Samuels and Witmer (2003, p. 289).

Growth	Gap
17	22
20	29
170	13
315	16
22	15
190	18
64	14
	6

Data 2.1: Soil respiration (mol CO_2/g soil/hr).

2.2.1.2 Statistical Analysis

Calculation of mean, standard deviation, median, variance and quartiles with `tapply()`:

```
> soil <- data.frame(treatment = c(rep(c("growth"), times = 7),
+ rep(c("gap"), times = 8)),response = c(17,20,170,315,22,190,64,22,29,13,
+ 16,15,18,14,6))
> tapply(X = soil$response, INDEX = soil$treatment, FUN = mean)

    gap  growth
 16.625 114.000

> tapply(X = soil$response, INDEX = soil$treatment, FUN = sd)

       gap     growth
  6.759913 114.398427

> tapply(X = soil$response, INDEX = soil$treatment, FUN = median)

   gap growth
  15.5   64.0

> tapply(X = soil$response, INDEX = soil$treatment, FUN = var)

        gap      growth
   45.69643 13087.00000

> tapply(X = soil$response, INDEX = soil$treatment, FUN = quantile)
```

```
$gap
   0%   25%   50%   75%  100%
 6.00 13.75 15.50 19.00 29.00

$growth
  0%  25%  50%  75% 100%
  17   21   64  180  315
```

2.3 The Function `stat.desc()`

The add on package `pastecs` comes along with a function called `stat.desc()` which returns a table with many values of descriptive statistics for several variables:

```
stat.desc(x, basic=TRUE, desc=TRUE, p=0.95, ...)
```

`x` is a data frame.

`basic` is set on `TRUE` by default. This means that the values for *number of observations*, *number of values that are zero*, *number of NAs*, *minimum*, *maximum*, *range* and *sum of all not missing values* are returned in the table. If the argument is set on `FALSE`, those values will be missing in the output.

The argument `desc` is responsible for the output of descriptive statistics. If it is set on `TRUE` (which is default), the values median, mean, standard error of mean, confidence interval for the mean according to the set confidence level `p`, variance, standard deviation and variation coefficient will be returned in the output.

2.3.1 Example Soil Respiration (2)

`pastecs` is loaded with the function `library()`:

```
> library(pastecs)
```

`stat.desc()` produces a comprehensive output:

```
> stat.desc(x = soil)
```

```
         treatment   response
nbr.val         NA   15.00000
nbr.null        NA    0.00000
nbr.na          NA    0.00000
min             NA    6.00000
max             NA  315.00000
range           NA  309.00000
sum             NA  931.00000
median          NA   20.00000
mean            NA   62.06667
SE.mean         NA   23.32390
CI.mean         NA   50.02480
var             NA 8160.06667
std.dev         NA   90.33309
coef.var        NA    1.45542
```

✎ Exercise 2

The lettuce varieties *Salad Bowl* and *Bibb* have been grown in a greenhouse under identical conditions for 16 days. Data 2.2 presents the dry weight of leaves from nine plants *Salad Bowl* and six plants *Bibb* (Samuels and Witmer, 2003, p. 226).

Create a data frame in the flat file format!

Calculate for both varieties the mean, standard deviation, median, variance, minimum, maximum, quartiles, sum and IQR respectively by using the function `tapply()`.

Salad Bowl	Bibb
3.06	1.31
2.78	1.17
2.87	1.72
3.52	1.20
3.81	1.55
3.60	1.53
3.30	
2.77	
3.62	

Data 2.2: Dry weight of two lettuce varieties (g).

Chapter 3

Graphics in R

R offers a huge amount of graphical functions. Most of the parameters for plotting functions can be applied universally. The example `boxplot()` points out the difference between a standard (default) plot and a plot with more specified arguments.

3.1 Boxplot

A boxplot shows the distribution of a sample. Therefore, it is often used to check the normal distribution. Several boxplots are helpful to estimate the homogeneity of variances between different samples (see section 5.1).

Some parameters of the function `boxplot()`:

```
boxplot(x, col = NULL, xlab = "...", ylab = "...", main = "...")
```

`x` is either a vector or a list containing several vectors. Alternatively, data might be specified with the `formula` construct:

```
formula = observations ~ grouping factor with two levels,
data = ..., subset = ..., na.action
```

Using the `formula` construct, group names are treated alphabetically (first position in the alphabet = first position in the function, e.g. first boxplot).

`col` specifies the color of the graph. The function `color()` calls all predefined colors.

`xlab` and `ylab` set the axes labels. The group names will be displayed by default (if header of a data frame column).

`main` adds a diagram title. This might be replaced by a separate function called `title()`.

Figure 3.1 shows the difference between the default configuration (specification of the dataset only) and a personalized plot with several arguments.

3.1.1 Example Soil Respiration (3)

Recalling the data from section 2.2.1, boxplots for gaps and dense tree population are drawn (figure 3.2):

```
> boxplot(formula = response~treatment, data = soil, col = "red1",
+ ylab = "Soil Respiration (mol CO2/g soil/hr)")
```

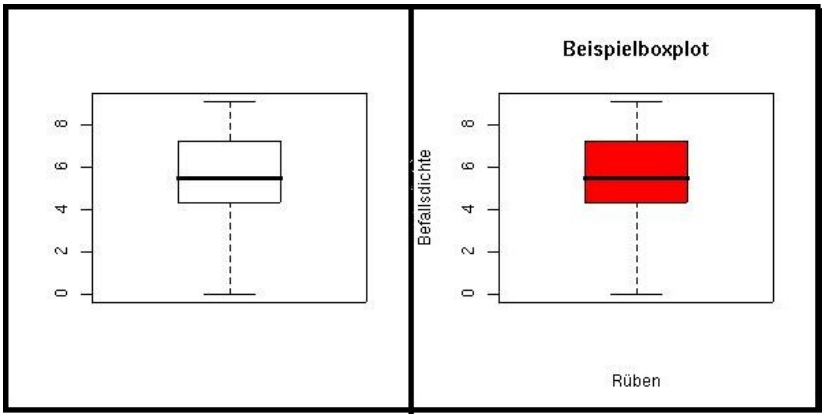


Figure 3.1: The difference between default configuration (**left:** boxplot(x = data)) and the specification of additional arguments (**right** boxplot(x = data, col = "red1", xlab = "Rüben", ylab = "Befallsdichte", main = "Beispielboxplot").

```
> title("Soil Respiration in the Forest")
```

3.2 Histogram

A histogram shows frequency and might also be used to obtain the normal distribution of a sample.

3.2.1 Example Soybeans (1)

3.2.1.1 Experiment

"As part of a study on plant growth, a plant physiologist grew 13 individually potted soybean seedlings of the type Wells II. She raised the plants in a greenhouse under identical environmental conditions (light, temperature, soil and so on). She measured the total stem length (cm) for each plant after 16 days of growth" (Data 3.1) (Pappas and Mitchell, 1984, the actual experiment contained several groups treated with different environmental conditions.), raw data published in Samuels and Witmer (2003, p. 179).

20.2	22.9
23.2	20.0
19.4	22.0
22.1	22.0
21.9	21.5
19.7	21.5
20.9	

Data 3.1: Stem length of soy bean seedlings.

3.2.1.2 Graphical Presentation of Data

The function `hist()` creates a histogram (figure 3.3):

```
> beans <- c(20.2,22.9,23.3,20,19.4,22,22.1,22,21.9,21.5,19.7,
+ 21.5,20.9)
> hist(beans, col = "red1", main = "Histogram of Soybean Seedlings",
+ breaks = 5)
```

The argument `breaks` defines the number of cells displayed in the histogram.

The argument `type` can be set `p` for points, `l` for line or `b` for both line and points.

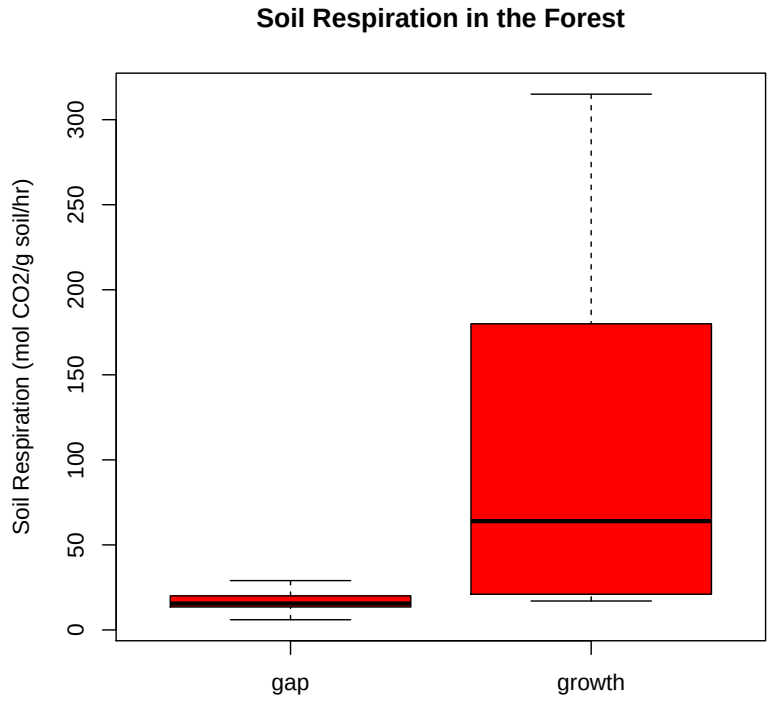


Figure 3.2: Boxplots of soil respiration in the forest.

3.3 Scatterplot

The function `plot()` returns the graph of an empiric cumulative distribution in its basic functionality.

The data about sugar beets (section 9.2.5) are used for visualization (figure 3.4).

```
> beets <- read.table(file = "../text/beets.txt", sep = "\t",
+ header = TRUE)
> plot(yield~water, data = beets, col = "red1", xlab = "irrigation (mm)",
+ ylab = "yield (t/ha)", main = "Sugar Beet Irrigation")
```

Attention! The function `plot` accepts data input in form of a `formula` construct, but only if the part `formula =` is **left out**!

The function `abline()` fits e.g. a horizontal line through the graph:

```
> abline(h = 14, col = "red1")
```

3.4 QQ-Plot

The QQ-plot for normal distribution is created with the function `qqnorm()` (see figure 3.5). `qqline()` fits a straight line through the points:

```
> qqnorm(beans, col = "red1", main = "QQ-Plot of Soybeans")
> qqline(beans, col = "red1")
```

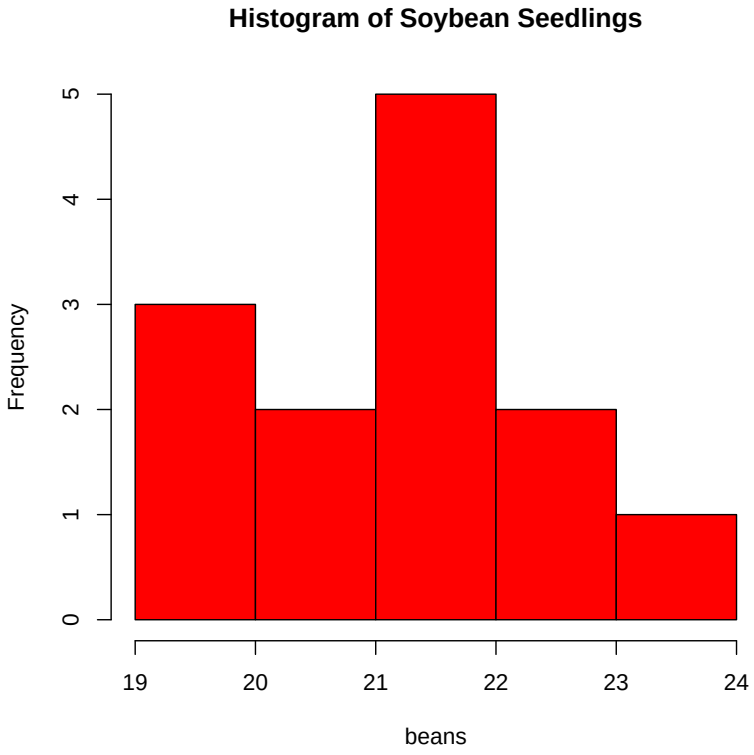


Figure 3.3: Histogram of soybean stem length.

The function `qq.plot()` in the package `car` provides a qq-plot for other distributions.

More examples for the usage of `plot()` and `qqnorm()` are presented in section 9.2.3.

3.5 Other Graphical Functions

R offers the opportunity to plot objects, e.g. confidence intervals, directly (see section 11.2.4.6 and regression diagnostics in sections 9.2.3 and 9.2.5.3).

Frequently used in Biology and Horticulture are in addition the stem-leaf diagram (`stem()`), `barplot()` and the pie diagram (`pie()`).

✎ Exercise 3

Use the data from Exercise 2 (Data 2.2) to plot the boxplots for the different varieties! Define title, axes names and box color!

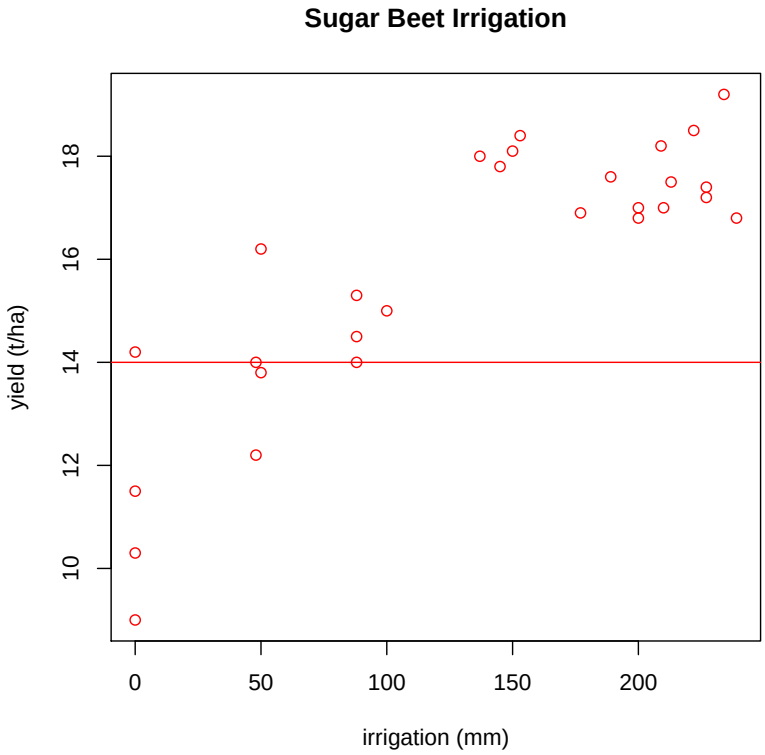


Figure 3.4: Sugar beet data as an example for a scatterplot.

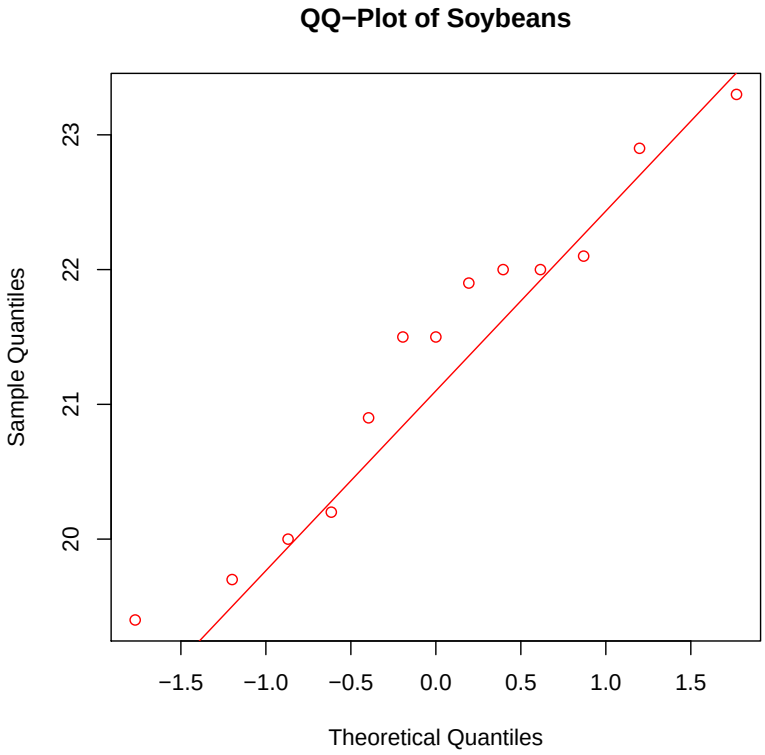


Figure 3.5: QQ-plot of soybean data.

Chapter 4

F-Test

4.1 Assumptions

The F-Test [1] is used in this manual as a tool for the decision which test is used for the comparison of two samples. It checks for heterogeneity of variances. The test result completes the consideration of boxplots as described in section 5.1.

The hypotheses for this test are called:

$$H_0 : \frac{\sigma_A}{\sigma_B} = 1$$
$$H_1 : \frac{\sigma_A}{\sigma_B} \neq 1$$

Normal distribution of both samples is an important assumption for the F-test (see section 5.1).

Attention! A significance in the F-test concludes a heterogeneity in variances. It is not possible to conclude a homogeneity from a non significant test result. I regard a p-value close to 1 accompanied by a look at the boxplots as an indicator for homogeneity of variances in this manual.

4.2 Implementation

4.2.1 The Function `var.test()`

```
var.test(x, y, ratio = 1, alternative = c("two.sided", "less", "greater"),
              conf.level = 0.95, ...)
```

`x` and `y` are two numerical vectors. Alternatively, data can be indicated with a `formula` construct (see section 3.1).

`ratio` refers to the ratio of variances in the working hypotheses. The default value is `1`.

`alternative` specifies a one- or two-sided test. Default value is `two.sided`.

`conf.level` defines the confidence level, `0.95` is default.

Used as a pre-test for a t-Test or Wilcoxon rank sum test, the only obligatory argument are two data vectors or a `formula` construct. The default configuration calculates a two-sided test for the ratio 1 to a confidence level of 0.95.

[1] There exists another F-Test called ANOVA (see chapter 10) which takes advantage of the same distribution obtaining another result. ANOVA checks for differences in two or more samples by analysis of variances.

4.2.2 Example "Wisconsin Fast Plant" (1)

4.2.2.1 Experiment

"The "Wisconsin Fast Plant", *Brassica campestris*, has a very rapid growth cycle that makes it particularly well suited for the study of factors that affect plant growth. In one such study, seven plants were treated with the substance Ancymidol (ancy) and were compared to eight control plants that were given ordinary water. Heights of all of the plants were measured, in cm, after 14 days of growth" (Data 4.1) (Ahern, 1998) cited according to Samuels and Witmer (2003, p. 228, author indicates that this data is only a randomly selected subset of the original data). Ancymidol is a growth suppressor used in agriculture as a herbicide.

Control	Ancy
10.0	13.2
13.2	19.5
19.8	11.0
19.3	5.8
21.2	12.8
13.9	7.1
20.3	7.7
9.6	

Data 4.1: Height of Brassica plants after 14 days (cm).

Are the variances homogeneous?

4.2.2.2 Statistical Analysis

```
> brassica <- read.table("../text/brassica.txt", sep = "\t",
+ header = TRUE)

> var.test(formula = height~group, data = brassica)

        F test to compare two variances

data:  height by group
F = 0.97316, num df = 6, denom df = 7, p-value = 0.9898
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1901215 5.5425762
sample estimates:
ratio of variances
         0.9731551
```

4.2.2.3 Interpretation

Please see section 5.2.3.2 for general interpretation instructions. The p-value is compared to an α-error that has been set a priori. If the p-value is smaller than α then the alternative hypothesis will be accepted.

The F-test checks for heterogeneity of variances. Although the homogeneity of variances is more interesting in this case, there is no test for homogeneity existing as far as I know. There are no general rules how to treat the output of a F-test when looking for homogeneity. I assume the variances to be more or less homogeneous if the p-value is rather big - including the interpretation of the boxplots. The arguments of `var.test()` are described in chapter 5.

A p-value of 0.9898 implies, that there is no significant heterogeneity in variances (comparing with an α of 5%) $\Longrightarrow$ homogeneity of variances.

Chapter 5

t-Test

5.1 Assumptions

The parametric t-Test compares the mean of two samples.

The "classical" **t-Test** is used with the following assumptions:

- **Approximate normal distribution of data** is read from the boxplots: The median lies in the middle of the box and both whiskers have an equal length (see figure 5.1. Watch each boxplot single!) The normal distribution results in continuity of data, e.g. temperatures measured in Kelvin or lengths measured in metres.
- **Homogeneity of variances** is either read from the boxplots: The respective boxes including whiskers have the same length. Or the homogeneity of variances is checked with a statistical test. Chapter 4 describes the F-test for two variances (`var.test()`).
- **Independence of data** is not fulfilled if one has e.g. taken data on the same fruit trees in two consecutive years. In vitro explants that originate in the same mother plant are not allowed to be treated as independent.

The **Welch t-test** is very similar to the "classical" t-Test. Assumptions are normal distribution as well as independence of data. But the Welch t-test is more tolerant to heterogeneity in variances.

A **paired t-Test** implies:

- **Paired data**: A paired sample results from e.g. the investigation of the effect of two insecticides on different branches of the same tree.
- **Normal distribution of the differences in mean** (Boxplot).

5.2 Implementation

5.2.1 The Function `t.test()`

```
t.test(x, y = NULL, alternative = c("two.sided", "less", "greater"),
            var.equal = FALSE, paired = FALSE, conf.level = 0.95, ...)
```

or

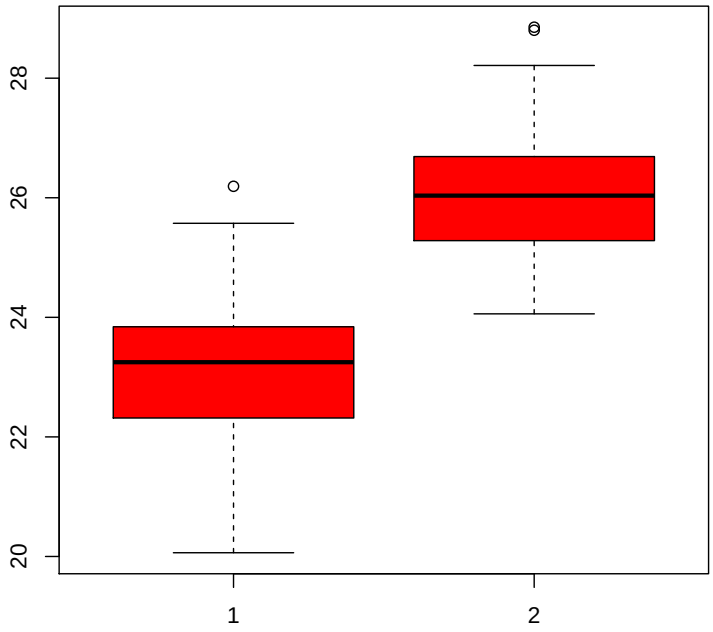


Figure 5.1: Boxplot example for a t-test.

```
t.test(formula, data, subset, na.action, ...)
```

`x` and `y` represent two vectors that will be compared. `x` is the only essential variable while `y` is an optional argument (the function `t.test()` might be used for a one sample t-test). Alternatively, data can be implied with the `formula`-construct (section 3.1).

`data` specifies the data set for a `formula`-construct.

`subset` selects data that will be ex- or included regarding certain criteria (see section 1.5.4.5).

`na.action` defines the treatment for values which are not available. Options for this argument are called with:

```
getOption("na.action").
```

Attention! R sorts variables called with a `formula`-construct alphabetically. That means B > A has to be indicated with `alternative = less`.

`alternative` indicates whether a two-sided (H_1: $\mu_1 \neq \mu_2$), one-sided acceding (H_1: $\mu_1 > \mu_2$) or one-sided seceding (H_1: $\mu_1 < \mu_2$) test is calculated.

`var.equal` declares whether the variances are heterogeneous (`FALSE`) or homogeneous (`TRUE`). The default is `FALSE`, which stands for a t-Welch test. It has to be set on `TRUE` for a classical t-Test.

`conf.level` specifies the confidence level. The α error is calculated from 1 - `conf.level`. 0.95 is the default value (95% $\Rightarrow \alpha = 5\%$).

`paired` is set on `FALSE` by default. A paired t-Test is calculated if it is set on `TRUE`.

5.2.2 The Function `qt()`

`qt()` calculates the quantile for a given p-value and degrees of freedom separately.

```
qt(p, df, lower.tail = TRUE)
```

`p` represents the given p-value, `df` stands for degrees of freedom.

The default argument `lower.tail = TRUE` is used for two-sided and one-sided seceding tests ($X \leq x$). It has to be set on `FALSE` for a one-sided acceding test.

5.2.3 Example "Wisconsin Fast Plant" (2)

Referring to the Data given in section 4.2.2, the question is now whether the two samples differ significantly in means ($\alpha = 5\%$).

5.2.3.1 Statistical Analysis

```
> brassica <- read.table("../text/brassica.txt", sep = "\t",
+ header = TRUE)
> boxplot(formula = height~group, data = brassica, ylab = "height in cm",
+ main="Height of Brassica Plants", col = "red", names = c("control", "ancy"))
```

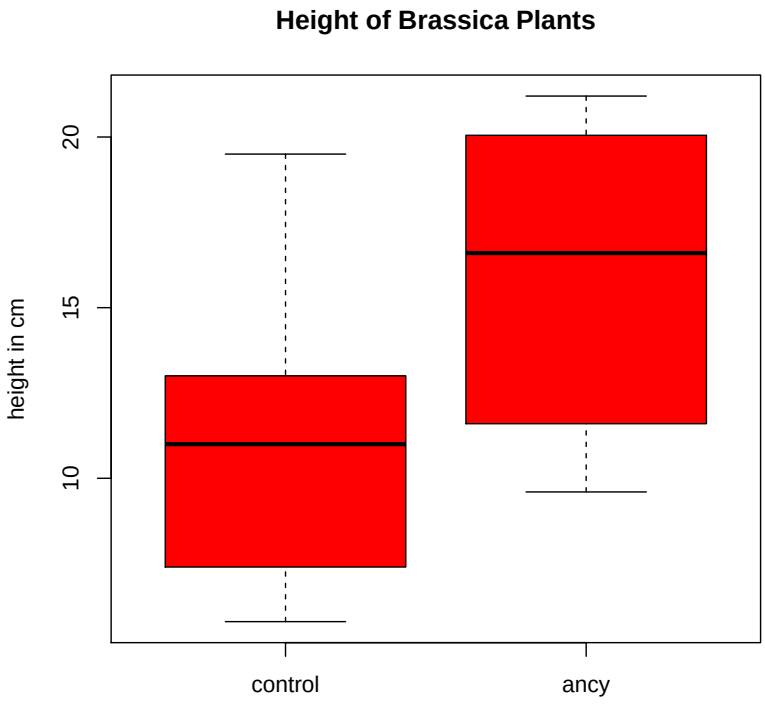


Figure 5.2: Boxplots of Brassica plant height after 14 days.

✔ Approximate **normal distribution** is accepted because the median is located in the middle of both boxes (see figure 5.2).

- ✔ Approximate **homogeneity of variances**, see result of F-test in section 4.2.2.
- ✔ **Continuous data** because height is indicated in cm
- ✔ **Independency of data** because the plants were treated independent from each other.

$\Longrightarrow$ Data is suiting for the analysis with a classical t-Test. Ancymidol is a growth repressor. Therefore, a one-sided test with the expectation that Ancymidol treated plants are smaller than the control group is calculated. Hypotheses:

$$H_0 : \mu_{control} \leq \mu_{ancy}$$
$$H_1 : \mu_{control} > \mu_{ancy}$$

```
> t.test(formula = height~group, data = brassica, var.equal = TRUE,
+ alternative = "less", conf.level = 0.95)

        Two Sample t-test

data:  height by group
t = -1.9919, df = 13, p-value = 0.03391
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -0.543402
sample estimates:
   mean in group ancy mean in group control
             11.01429              15.91250
```

5.2.3.2 Interpretation

```
        Two Sample t-test
```

The line presents the test header. If the variable `var.equal = TRUE` would not have been set, the function would return `Welch Two Sample t-test`.

```
data:  height by group
```

This says that the `formula`-construct compared heights dependent on the group.

```
t = -1.9919, df = 13, p-value = 0.03391
```

The test statistic `t` amounts 1.9919. This value is usually compared to a table value. The comparison of means is called "significant" if the t-value is more extreme than the table value for the respective quantile and degrees of freedom. Degrees of freedom are printed as `df = 13`. The `p-value` is compared to the respective α-error. The test result is significant if the p-value is smaller than α.

α must be set a priori before the test itself is calculated! In R, the default of α is 5%. The plants treated with Ancymidol are significantly shorter than the non treated control group because $0.03391 < 0.5$. The alternative hypothesis is accepted.

The test statistic can be calculated with `qt()` separately:

```
> qt(0.03391, 13, lower.tail = FALSE)
```

```
[1] 1.99187

alternative hypothesis: true difference in means is greater than 0
```

This line returns the alternative hypothesis.

```
95 percent confidence interval:
      -Inf -0.543402
```

The 95% confidence interval for the difference of the true parameters $\mu_{control}$ - μ_{ancy} is displayed. If the experiment was repeated infinite times, the true difference would be located within the respective confidence interval in 95% of all cases . However, there is no statement about the current experiment in it.

Practice: If the confidence interval includes zero, the test result is counted as not significant. If the result is significant (zero not included), the difference to zero represents a measure of rejection of the H_0-hypothesis. The interval width accounts for scattering and the number of observations. In general, confidence intervals are displayed in the original data's dimension: in this example measurements in centimeteres.

The given confidence interval `0.543402 Inf` indicates a significance to a confidence level of 0.95 because zero is excluded: $\mu_{control}$ - μ_{ancy} = 0 can be rejected with an error probability of 5%. More detailed, the confidence interval indicates that the control plants are at least 0.542402 cm higher than the Ancymidol treated plants.

```
sample estimates:
mean in group ancy mean in group control
          11.01429              15.91250
```

Output of the mean values. Plants treated with Ancymidol have an average height of 11.0 cm whereas the control plants have a mean height of 15.9 cm.

The overall conclusion for this experiment is that the alternative hypothesis is accepted with a confidence level of 0.95.

✎ Exercise 4

The infection of strawberries with small white worms leads to a reduction in harvest. It is possible to fight the parasite with disinfectants. An new additive is suspected to extend the effective period but side effects on the strawberry plants are still unknown. Five plots on a field have randomly been chosen to investigate the overall effect of the additive on strawberry plants. Each plot was randomly divided in two parts where one half was treated with the disinfectant without additive and the other half was treated with disinfectant and additive. The strawberry yield in presented in Data 4.2 (Wonnacott and Wonnacott, 1990, p. 273)

Standard	Additive
109	107
68	72
82	88
104	101
93	97

Data 4.2: The effect of a new disinfection additive fighting white small worms on strawberries.

Develop convenient working hypotheses. Is the data normal distributed and homogeneous in variances? Which test do you choose? Interpret the output!

5.2.4 Example: Root Growth of Mustard Seedlings

5.2.4.1 Experiment

The influence of light and darkness on the root growth of mustard seedlings has been investigated in an experiment (Hand et al., 1994, p. 75, this is a subset of the complete dataset). The question is if the length of roots differs for the two treatments (Data 4.3).

light	dark
21	22
39	16
31	20
13	14
52	32
39	28
55	36
50	41
29	17
17	22

Data 4.3: Root growth of mustard seedlings (cm).

5.2.4.2 Statistical Analysis

```
> mustard <- read.table(file = "../text/mustard.txt", sep = "\t",
+ header = TRUE)
> boxplot(formula = response~treatment, data = mustard, col = "red",
+ ylab = "rootlength (cm)")
> title("Root Growth of Mustard Seedlings")
```

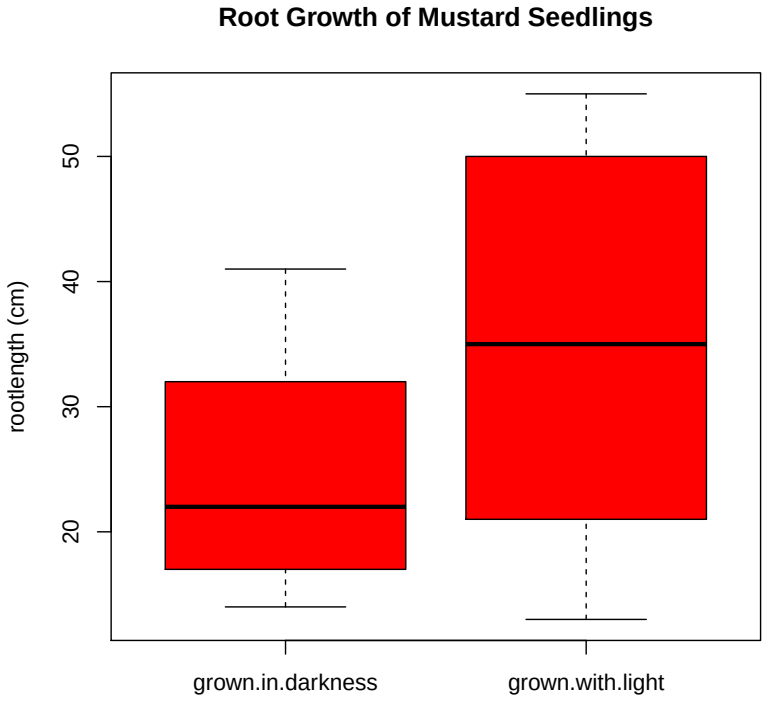


Figure 5.3: Boxplots for root growth of mustard seedlings.

- ✔ Approximate **normal distribution** (figure 5.3) and continuity of data (root length was measured in cm).
- ✔ **Heterogeneity of variances** (figure 5.3, boxes differ in length).
- ✔ The different treatments are assumend to be **independent**.

A two-sided hypothesis is reasonable: the direction of a light effect on mustard roots is unknown. The α-error is set on 5%. Pair of hypotheses:

$$H_0 : \mu_{light} = \mu_{dark}$$
$$H_1 : \mu_{light} \neq \mu_{dark}$$

```
> t.test(formula = response~treatment, data = mustard,
+ alternative = "two.sided", conf.level = 0.95)
```

```
        Welch Two Sample t-test

data:  response by treatment
t = -1.7748, df = 14.879, p-value = 0.09638
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -21.57753   1.97753
sample estimates:
mean in group grown.in.darkness  mean in group grown.with.light
                           24.8                            34.6
```

5.2.4.3 Interpretation

The output is interpreted as shown in section 5.2.3.2.

```
t = -1.7748, df = 14.879, p-value = 0.09638
```

The `p-value` is greater than 0.05. Therefore, the roots of mustard seedlings grown with light and in darkness do not differ significantly with an error probability of 5%. It would have been possible to compare the p-value with another α, e.g. 0.1. In this case, the result would have been significant. But as mentioned before, the α-error has to be set a priori before calculating the test.

Due to the principle of a t-Welch test, the number of degrees of freedom is reduced.

```
95 percent confidence interval:
 -21.577530   1.977530
```

Zero is included in the confidence interval which means that the test result is not significant to a confidence level of 95%.

```
sample estimates:
mean in group grown.in.darkness  mean in group grown.with.light
                           24.8                            34.6
```

Plants grown in darkness have an average root length of 24.8 cm, whereas the group treated with light has an average root length of 34.6 cm.

This test result leads to the conclusion that the null hypothesis cannot be rejected to a confidence level of 0.95. However, this does not assure the equality of the two samples because a t-test is not checking for homogeneity.

✎ Exercise 5

"Two varieties of lettuce were grown for 16 days in a controlled environment. Data 4.4 shows the total dry weight (in g) of the leaves of nine plants of the variety *Salad Bowl* and six plants of the variety *Bibb*." (Knight and Mitchell, 2000, author states that the actual sample sizes were equal; some observations have been omitted.) cited according to Samuels and Witmer (2003, p. 226).

Find adequate hypotheses. Is the data normal distributed and homogeneous in variances? Which test do you choose? Interpret the R-output!

Salad Bowl	Bibb
3.06	1.31
2.78	1.17
2.87	1.72
3.52	1.20
3.81	1.55
3.60	1.53
3.30	
2.77	
3.62	

Data 4.4: Leave dry weight of two lettuce varieties.

5.2.5 Example: Growth Induction

In an experiment, a certain treatment is supposed to initiate growth induction. 20 plants have been divided in two groups by fitting pairs that are as similar as possible. One group was treated, the other was left as a control (Data 4.5) (Mead et al., 2003, p. 72, data has been modified slightly.).

```
> growth <- read.table("../text/growth.txt")
> differences <- growth$height[1:10] - growth$height[11:20]
> boxplot(x = differences, col = "red", ylab = "growth",
+ main = "Pair Differences")
```

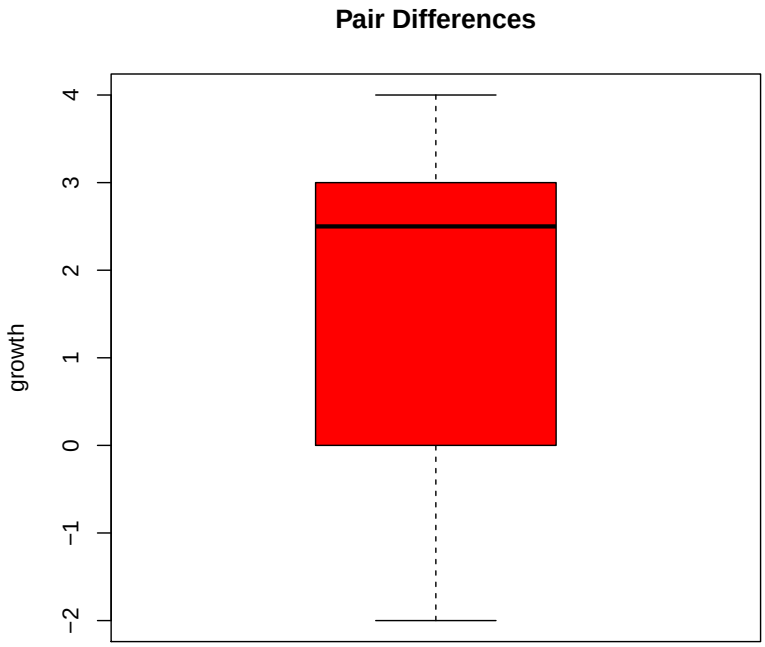


Figure 5.4: Boxplots for growth induction.

- ✔ Approximate **normal distribution** of pair differences (figure 5.4, the test is assumed to be robust to a median which is not perfectly located in the boxes' middle).
- ✔ **Paired data** because plant pairs that are as similar as possible have been formed.

$\Longrightarrow$ Paired one-sided t-test (because it is expected that a growth inductor created taller plants).

```
> t.test(formula = height~treatment, data = growth , paired = TRUE,
+ alternative = "less")

        Paired t-test
```

Treated plant	Control plant
7	4
10	6
9	10
8	8
7	5
6	3
8	10
9	8
12	8
13	10

Data 4.5: Growth induction.

```
data:  height by treatment
t = -2.5468, df = 9, p-value = 0.01568
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
       -Inf -0.4763981
sample estimates:
mean of the differences
                  -1.7
```

The p-value is smaller than 0.05. For this reason, the test result is significant. A treatment for growth induction results in a stronger plant growth.

The analysis of confidence intervals leads to the same result: Zero is not included in the interval which means that the test result is significant to a confidence level of 95%. Plants treated with a growth inductor are at least 0.47 cm taller than the untreated control group.

Chapter 6

Wilcoxon Rank Sum Test

6.1 Assumptions

The t-test is not very tolerant for deviation from the normal distribution. The Wilcoxon Rank Sum Test is used with consideration of an unknown distribution. Assumptions for this test are:

- **Homogeneity in variances**.
- **At least ordinal scaling**.
- **Independent data**.

6.2 Implementation

6.2.1 The Function `wilcox.test()`

```
wilcox.test(x, y, alternative = c("two.sided", "less", "greater"),
                paired = FALSE, correct = TRUE, exact = NULL,
                conf.int = FALSE, conf.level = 0.95, ...)
```

or with a `formula`-construct:

```
wilcox.test(formula, data, subset, na.action, ...)
```

`x` is a numerical vector. `y` represents an optional second numerical vector for the two sample test.

`formula` Alternatively, data might be stated with a `formula`-construct (see section 3.1).

`alternative` indicates whether a two-sided, one-sided acceding or one-sided seceding test is calculated.

`paired` defines whether the data is dependent (see section 5.1). Default value is `FALSE`.

`exact` specifies whether the `p-value` shall be calculated correctly. The default `FALSE` calculates an asymptotic p-value. An exact p-value should be calculated for numbers of observations smaller than 50 in each group without ties. The function `wilcox.test()` is not capable of calculating an exact p-value if the data contains ties. `wilcox.test()`

calculates the asymptotic p-value when the number of observations is low and the data contains ties. The package `exactRankTests` solves this problem (see section 6.2.2).

`conf.int` can be set on `TRUE` which results in the calculation of a Hodges-Lehmann confidence interval.

`conf.level` sets the confidence level. The default value is 0.95.

`correct` states whether a continuity correction is applied. The default value is `TRUE`.

6.2.2 The Function `wilcox.exact()`

The package `exactRankTests` has to be installed and loaded with `library(exactRankTests)` before using the function `wilcox.exact()` (see sections 1.4.2.1 and 1.4.3.1 for installation instructions).

```
wilcox.exact(x, y = NULL, alternative = c("two.sided", "less", "greater"),
                  paired = FALSE, exact = NULL,
                  conf.int = FALSE, conf.level = 0.95, ...)
```

or

```
wilcox.exact(formula, data, subset, na.action, ...)
```

The variables in `wilcox.exact()` are in general the same as described for `wilcox.test()`. The only difference is that this function is able to calculate an exact p-value with tied data. It is therefore reasonable to use this function throughout all Wilcoxon test problems.

6.2.3 Example Mechanical Stress

6.2.3.1 Experiment

"A plant physiologist conducted an experiment to determine whether mechanical stress can retard the growth of soybean plants. Young plants were randomly allocated in two groups of 13 plants each. Plants in one group were mechanically agitated by shaking for 20 minutes twice daily, while plants in the other group were not agitated. After 16 days of growth, the total stem length (cm) of each plant was measured", with the result given in the Data 6.1 (Pappas and Mitchell, 1984), raw data published in Samuels and Witmer (2003, p. 302, the actual experiment included several groups of plants grown under different environmental conditions.).

Control	Stress
25.2	24.7
29.5	25.7
30.1	26.5
30.1	27.0
30.2	27.1
30.2	27.2
30.3	27.3
30.6	27.7
31.1	28.7
31.2	28.9
31.4	29.7
33.5	30.0
34.3	30.6

Data 6.1: Stem length of soybean plants after 16 days of growth in cm.

6.2.3.2 Statistical Analysis

Previous research indicated that mechanically stressed plants tend to be shorter than their non stressed relatives $\Longrightarrow$ one-sided test with the following hypotheses:

$$H_0 : F_{control}(y) \leq F_{stress}(y)$$
$$H_1 : F_{control}(y) > F_{stress}(y)$$

```
> growth.retardant <- read.table(file = "../text/retardant.txt",
+ header = TRUE, sep = "\t")
> boxplot(formula = response~treatment, data = growth.retardant,
```

```
+ col="yellow", ylab ="stem length (cm)", names = c("control","stress"),
+ main = "Stem Length of Soybean Plants")
```

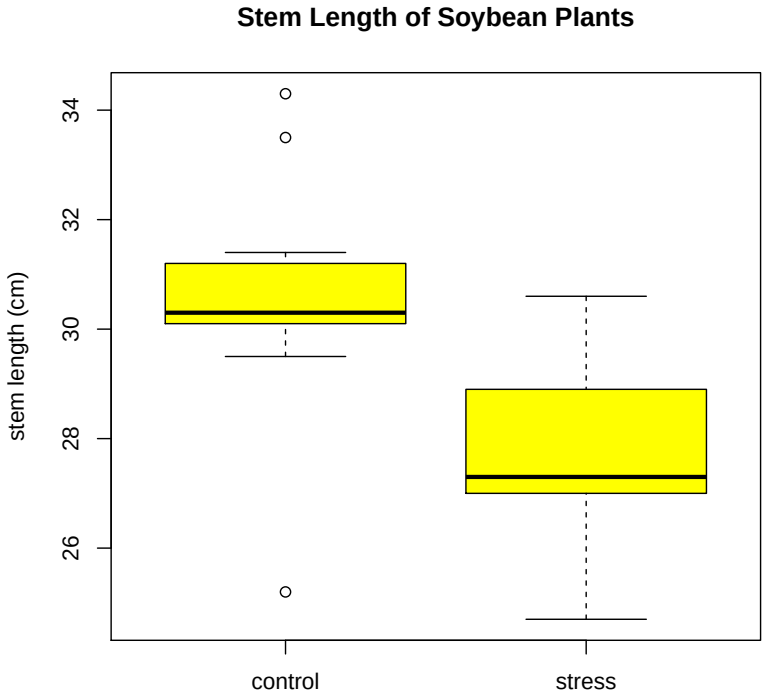


Figure 6.1: Boxplots for stem length of seismically stressed soybean plants. Data is not normal distributed.

- ✔ **Continuous data** (length measured in cm).
- ✔ **Homogeneity of variances** is critical, tolerance is assumed.
- ✔ **Independent data** (no fitted pairs, single plants have been measured independent from each other).

6.2.3.3 Asymptotic p-value with `wilcox.test`

```
> wilcox.test(formula = response~treatment, data = growth.retardant,
+ correct = FALSE, exact = FALSE, alternative = "greater", conf.int = TRUE)

        Wilcoxon rank sum test

data:  response by treatment
W = 148.5, p-value = 0.0005122
alternative hypothesis: true location shift is greater than 0
95 percent confidence interval:
 1.50005     Inf
sample estimates:
difference in location
              3.000042
```

Similar to a t-test output, the header and alternative hypothesis are printed in the beginning.

```
W = 148.5, p-value = 0.0005122
```

`W` is the Wilcoxon test statistic. The extremely small `p-value` of 0.0005122 leads in this case to the conclusion that plants exposed to seismic stress are highly significantly shorter than the nonstressed control plants.[1]

```
95 percent confidence interval:
 1.500050      Inf
```

Zero is included in the confidence interval of the Wilcoxon rank sum test. That means the test result is significant to a confidence level of 95%. Nontreated plants are at least 1.5 cm up to infinite cm longer than plants exposed to seismic stress.

```
sample estimates:
difference in location
              3.000042
```

Output of the sample estimate for the difference in location of both distributions.

6.2.3.4 Exact p-value with the Function `exact.wilcox()`

The number of observations in the respective groups is smaller than 50. Therefore, an exact test is required. For the reason that the dataset contains ties, the exact p-value needs to be calculated with the package exactRankTests:

```
> library(exactRankTests)
> wilcox.exact(formula = response~treatment, data = growth.retardant,
+ exact = TRUE, alternative = "greater", conf.int = TRUE)

        Exact Wilcoxon rank sum test

data:  response by treatment
W = 148.5, p-value = 0.0002604
alternative hypothesis: true mu is greater than 0
95 percent confidence interval:
 1.5 Inf
sample estimates:
difference in location
                     3
```

Test statistic `W`, the exact p-value as well as the confidence interval are returned.

6.2.3.5 Conclusion

Plants treated with seismic stress are significantly shorter than the control group with an error probability of 5%.

Red	Green
8.4	8.6
8.4	5.9
10.0	4.6
8.8	9.1
7.1	9.8
9.4	10.1
8.8	6.0
4.3	10.4
9.0	10.8
8.4	9.6
7.1	10.5
9.6	9.0
9.3	8.6
8.6	10.5
6.1	9.9
8.4	11.1
10.4	5.5
	8.2
	8.3
	10.0
	8.7
	9.8
	9.5
	11.0
	8.0

Data 6.2: Height of soybean plants treated with red and green light two weeks after germination (inches).

[1] Due to the small number of observations, the calculation of an exact p-value would be more correct.

✎ Exercise 6

"A researcher investigated the effect of green and red light on the growth rate of soybean plants. End point was the plant height two weeks after germination (measured in inches). The different light colors were produced by the usage of thin colored plastic as used for e.g. theater spot lights" (Data 6.2) (Gent, 1999), published in Samuels and Witmer (2003, p. 243)).

- Which test is suitable for the evaluation of this data?
- Do you test one- or two-sided?
- Which are your hypotheses?
- Implement the exact test and interpret the output!

Chapter 7

χ^2-Test

7.1 Assumptions

The χ^2-test is a nonparametric test suiting for e.g. dichotomous data. Dichotomous data are a kind of discrete data. For example, Mendel's yellow or green pea color, high or low pest infestation and jagged or round shaped leaves are dichotomous end points.

7.1.1 χ^2 Goodness-of-Fit Test

The χ^2 Goodness-Of-Fit Test compares a measured distribution with a known, theoretical distribution. The classical example is the comparison of an empirical phenotype ratio with a predicted phenotype ratio in genetics . Two-sided hypotheses:

$$H_0 : F_0(x) = F_1(x)$$
$$H_1 : F_0(x) \neq F_1(x)$$

7.1.2 χ^2 Homogeneity Test

The χ^2 Homogeneity Test checks whether the procentual relation of two samples is different (e.g. `infestation` and `no infestation` for the treatments with and without insecticide).

$$H_0 : \pi_0(x) = \pi_1(x)$$
$$H_1 : \pi_0(x) \neq \pi_1(x)$$

Both tests might be calculated one-sided.

7.2 Implementation

7.2.1 χ^2 Goodness-of-Fit Test - `chisq.test()`

The function `chisq.test()` is implemented in the following form:

```
chisq.test(x, p = ...)
```

`x` is a vector containing the observed distribution.

`p` for *probability* is a vector of the same length as `x` containing the expected distribution.

7.2.2 χ^2 Homogeneity Test for 2x2-Tables - `chisq.test()`

```
chisq.test(x, correct = TRUE)
```

`x` represents a matrix in the form of a 2x2-table.

`correct` states whether the Yates-correction shall be used (number of observations smaller than 20) or not. The default configuration (`FALSE`) calculates the original χ^2-test according to Pearson.

7.2.3 Useful Functions for χ^2-Tests

`pchisq()` calculates a p-value for a known quantile for defined degrees of freedom:

```
pchisq(q, df, lower.tail = TRUE)
```

`q` is the χ^2-value, the test statistic.

`df` represents the degrees of freedom.

`lower.tail` indicates the kind of probability. `TRUE` stands for 1 - α, `FALSE` stands for α. `TRUE` is the default value. That means you have to indicate 0.95 for an α-error of 5%.

`qchisq()` calculates the test statistic for a known probability with specific degrees of freedom:

```
qchisq(p, df, lower.tail = TRUE)
```

`p` represents the known probability.

7.2.4 Example Snapdragon

7.2.4.1 Experiment

A geneticist, investigating the Mendelian predictions for F2 generations observed the ratio of phenotypes shown in table 7.1 for the F2 generation (Baur et al., 1931) cited according to Samuels and Witmer (2003, p. 392f).

Red	Pink	White
54	122	58

Table 7.1: Ratio of phenotypes in the F2 of snapdragon plants.

Does the observed result differ from the expected ratio of 1:2:1 for a F2 generation in the intermediate Mendelian heredity (α-error 5%)?

7.2.4.2 Statistical Analysis

No appliance of the Yates-correction because there exist more than 20 observations.

```
> snapdragon <- c(54,122,58)
> mendel.probs <- c(1,2,1)/4
> chisq.test(x = snapdragon, p = mendel.probs)
```

```
        Chi-squared test for given probabilities

data:  snapdragon
X-squared = 0.5641, df = 2, p-value = 0.7542
```

`X-squared` represents the test statistic while `df` gives the degrees of freedom.

`p-value` returns the two-sided p-value (`chisq.test()` is always testing two-sided.)

7.2.4.3 Interpretation

The observed ratio of phenotypes does not differ significantly from the Mendelian ratio for a F2 generation in the intermediate heredity. The H_0 hypothesis cannot be rejected.

✎ Exercise 7

"Researchers studied a mutant type of flax seed that they hoped would produce oil for use in margarine and shortening. The amount of palmitic acid in the flax seed was an important factor in this research; a related factor was whether the seed was brown or variegated. The seeds were classified into six combinations of palmitic acid and color, shown in table 7.2. According to a hypothesized genetic model, the six combinations should occur in a 3:6:3:1:2:1 ratio" (Saedi and Rowland, 1997) cited according to Samuels and Witmer (2003, p. 395).

Color	Acid Level	No
brown	low	15
brown	medium	26
brown	high	15
mottled	low	0
mottled	medium	8
mottled	high	8

Table 7.2: Ratio of phenotypes for flax seeds in the F1 generation.

Does the observed distribution differ from the hypothesized model?

7.2.5 Example Barley

7.2.5.1 Experiment

	surviving	dead
A	64	16
B	34	46

Table 7.3: Survival rate of barley seeds with and without heat treatment.

Researchers investigated the survival rate of barley seeds after a heat treatment. Sample A was used as untreated control group whereas Sample B was exposed to heat. All seeds were cut longitudinal and incubated in 0.1% 2,3,5-triphenyltetrazoliumchloride for half an hour. The breathing, living embryo reduces tetrazoliumchloride to the intensively red colored insoluble substance triphenyl formazan. Surviving seeds were counted according to color (see table 7.3) (Bishop, 1980, p. 76).

7.2.5.2 Statistical Analysis

Does the heat treatment reduce the survival rate of barely seeds? $\alpha = 1\%$.

$$H_0 : \pi_{noheat}(x) \leq \pi_{heat}(x)$$

$$H_1 : \pi_{noheat}(x) > \pi_{heat}(x)$$

Since the number of observations is adequate, no Yates correction is used.

```
> barley <- matrix(c(64,34,16,46), ncol = 2)
> line.names <- c("treatment.A","treatment.B")
> col.names <- c("viable", "not.viable")
> dimnames(barley) <- list(line.names,col.names)
> barley.chi <- chisq.test(barley,correct = FALSE)
> barley.chi

        Pearson's Chi-squared test

data:  barley
X-squared = 23.7, df = 1, p-value = 1.126e-06
```

`chisq.test()` calculates the two-sided p-value as a matter of principle. Therefore, the p-value has to be divided by two or to be compared with a doubled α for a one-sided comparison.

```
> barley.p <- barley.chi$p.value/2
> barley.p

[1] 5.629705e-07
```

Yes, the heat treatment does reduce the survival rate of barley seeds significantly to a confidence level of 0.99.

✎ Exercise 8

Some species occur associated with each other in certain habitats. The reason might be that both are influenced by similar micro climates (e.g. shade plants usually appear together with other shade liking plants), soil conditions (e.g. chalk liking plants will be accompanied by other chalk liking plants), or that one species creates good living conditions for the other one (e.g. host-parasite relationships), or numerous other explanations. (...) A common method for the analysis of such relationships is setting squares in which the respective species are counted. Table 7.4 represents an exemplary dataset (Bishop, 1980, p. 111).

	Presence A	**Absence A**
Presence B	25	75
Absence B	25	75

Table 7.4: Questionable interaction of two species in an ecosystem.

Are those two species associated? $\alpha = 10\%$.

Chapter 8

Analysis of Correlation

8.1 Assumptions

A linear coherence between one or more random variables in a sample is investigated quantitatively by analysis of correlation. However, correlation does not return the mathematical equation. The correlation coefficient r is set between -1 and +1. The closer the absolute value is located to 1, the better is the correlation. A negative coefficient implies that the values of one variable are big while the other variable results in small values. A positive coefficient is returned for data in which both variables are big or small.

The correlation coefficient itself does not state anything about the significance of correlation. Therefore, a test resembling the t-test is used for checking the significance.

8.1.1 Pearson

Assumptions for a correlation according to Pearson are:

- **Normal distributed data**.
- **Independence of observations**.

Pearson's correlation coefficient is named ρ.

8.1.2 Spearman

Correlation according to Spearman is nonparametric and therefore independent from monotone coordinate transformation. Assumptions:

- (Normal distribution of data is not required.)
- **Independency of observations**.

8.2 Implementation

8.2.1 The Function `cor()`

`cor()` is used as follows:

```
cor(x, y = NULL, use = "all.obs",
          method = c("pearson", "spearman"))
```

`x` gives a vector or data frame. `y` is a vector containing the second variable.

The default value for `use` is `all.obs` (= all observations). Missing values produce an error message. `pairwise.complete.obs` uses only complete pair observations.

`method` specifies whether a correlation according to Pearson or Spearman is calculated.

The function produces an output table presenting the coefficients of all possible correlations.

weight (g)	length (cm)
0.7	1.7
1.2	2.2
0.9	2.0
1.4	2.3
1.2	2.4
1.1	2.2
1.0	2.0
0.9	1.9
1.0	2.1
0.8	1.6

Data 8.1: Weight and length of Broad Beans.

8.2.2 The Function `cor.test()`

`cor.test()` tests the significance of a correlation. The hypotheses for a two-sided test are:

$$H_0 : \rho = 0$$

$$H_1 : \rho \neq 0$$

```
cor.test(x, y,
              alternative = c("two.sided", "less", "greater"),
              method = c("pearson", "spearman"),
              conf.level = 0.95, ...)
```

`x, y` represents two vectors. Alternatively, data might be specified with a `formula`-construct:

```
formula = ~var1+var2, data = frame.name
```

`method` specifies whether a correlation according to Pearson or Spearman's rank correlation is calculated.

`conf.level` indicates the test's confidence level (default are 95%).

8.2.3 Example broad beans

8.2.3.1 Experiment

A sample of broad beans classified as the variety *Roger's Emperor* was investigated with regard on length and weight (Data 8.1) (Bishop, 1980, p. 64).

8.2.3.2 Statistical Analysis

```
> broad <- read.table(file = "../text/broad.txt", sep = "\t",
+ header = TRUE)
> plot(length~weight, data = broad, col = "green3",
+  xlab = "length (cm)", ylab = "weight (g)")
> title("Scatterplot of the Broad Bean Data")
> boxplot(x = broad$weight, broad$length, col = "green3",
+ main = "Boxplots of the Broad Bean Data")
```

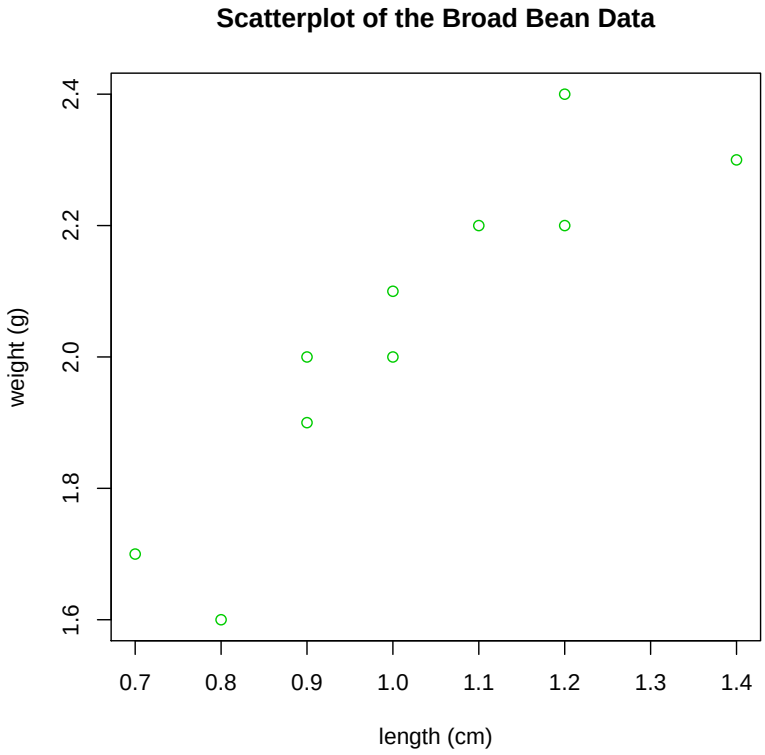


Figure 8.1: Scatterplot of broad bean data.

Figure 8.1 leads to the expectation of a linear correlation with a positive coefficient ($\Rightarrow$ one-sided test).

- ✔ **Normal distribution** of both variables (see figure 8.2)
- ✔ **Indepency of observations** is assumed.

$\Longrightarrow$ Correlation according to Pearson.

`cor` returns all possible correlation coefficients:

```
> cor(broad, method = "pearson")

          weight    length
weight 1.0000000 0.8983172
length 0.8983172 1.0000000
```

`cor.test()` investigates the correlation between length and weight of broad beans with regard to the significance:

```
> cor.test(formula = ~length+weight, data = broad, method = "pearson",
+ alternative = "greater")

        Pearson's product-moment correlation

data:  length and weight
```

leaf area	dry weight
411	2.00
550	2.47
471	2.11
393	1.89
427	2.05
431	2.30
492	2.46
371	2.06
470	2.25
419	2.07
407	2.17
489	2.32
439	2.12

Data 8.2: Leave area (cm^2) and dry weight (g) of soybean seedlings.

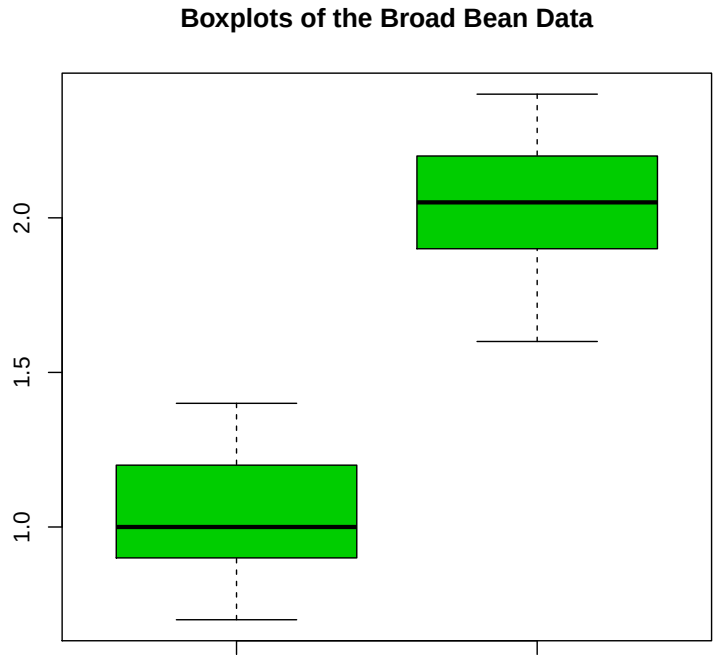


Figure 8.2: Boxplot of broad bean data for an investigation of normal distribution.

```
t = 5.7832, df = 8, p-value = 0.0002065
alternative hypothesis: true correlation is greater than 0
95 percent confidence interval:
 0.6867277 1.0000000
sample estimates:
      cor
0.8983172
```

The Pearson correlation with the coefficient (returned at `cor`) r = 0.0002065 is highly significant with an error probability of 5%. Please see section 5.2.3.2 for confidence interval interpretation instructions.

Ascorbic acid concentration ($\frac{\mu g}{cm^3}$)	Response
150	5.9
300	4.8
450	3.7
600	2.4
750	0.9
900	0.0

Data 8.3: Photometric data of ascorbic acid content.

8.2.4 Example Soybeans (2)

"A plant physiologist grew 13 individually potted soybean seedlings in a greenhouse. Data 8.2 gives measurements of the total leaf area (cm^2) and total plant dry weight (g) for each plant after 16 days of growth" (Pappas and Mitchell, 1984), rawdata published in Samuels and Witmer (2003, p. 563f, one dry weight value differs from the original data.).

```
> bean <- read.table(file = "../text/bean.txt", sep = "\t",
+ header = TRUE)
> plot(area~weight, data = bean, col = "green3", xlab = "area (squarecm)",
+ ylab = "dry weight (g)", main = "Soybean Data")
> boxplot(x = bean$area, col = "green3",
+ main = "Leaf Area of Soybean Seedlings", ylab = "area (squarecm)")
```

```
> boxplot(x = bean$weight, col = "green3",
+ main = "Dry Weight of Soybean Seedlings", ylab = "dry weight (g)")
```

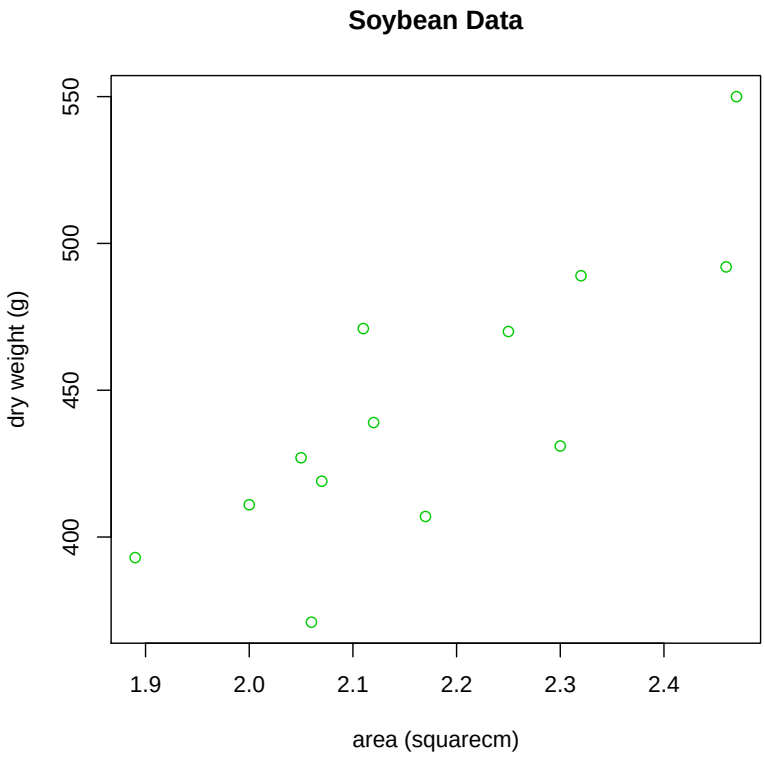


Figure 8.3: Scatterplot of soybean data.

- ✔ **Normal distribution** is rejected because the median does not lie in the box middle (figures 8.4 and 8.5).
- ✔ **Independency of observations** is assumed.

$\Longrightarrow$ Spearman's Rank Correlation. Figure 8.3 implies a positive correlation coefficient. Therefore, a one-sided acceding test is calculated:

```
> cor.test(formula = ~weight+area, data = bean, method = "spearman",
+ alternative = "greater")

        Spearman's rank correlation rho

data:  weight and area
S = 74, p-value = 0.0009218
alternative hypothesis: true rho is greater than 0
sample estimates:
      rho
0.7967033
```

The correlation coefficient ρ is 0.7967022. The correlation is significant with an error probability of 5% because the p-value 0.0008658 is much smaller than 0.05.

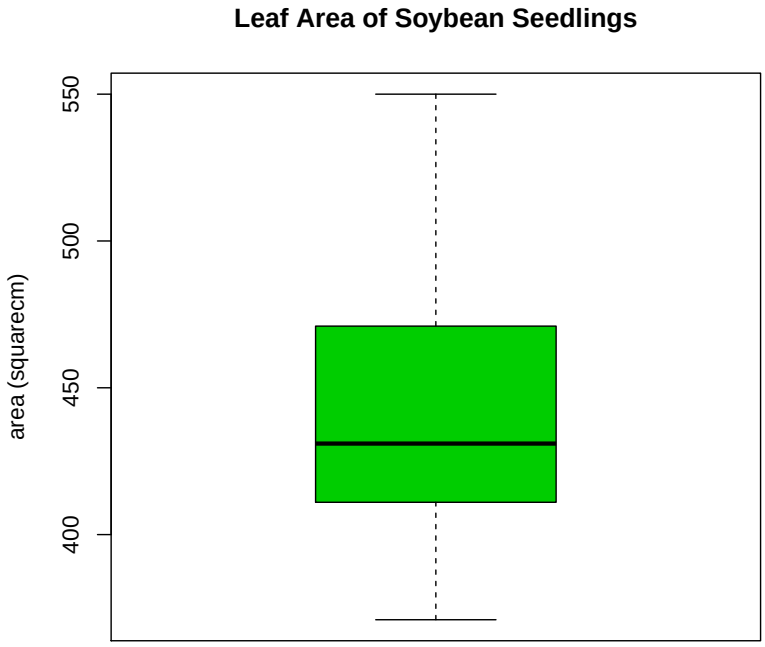


Figure 8.4: Boxplot of soybean seedlings' leaf area (checking for normal distribution).

✎ Exercise 9

The content of ascorbic acid is measured with a photoelectric absorption meter by using the blue starch-iodine complex. In order to standardize this procedure, samples with a known concentration of ascorbic acid are measured, first (Data 8.3) (Bishop, 1980, p. 70).

Are ascorbic acid concentration and metered values correlated significantly?

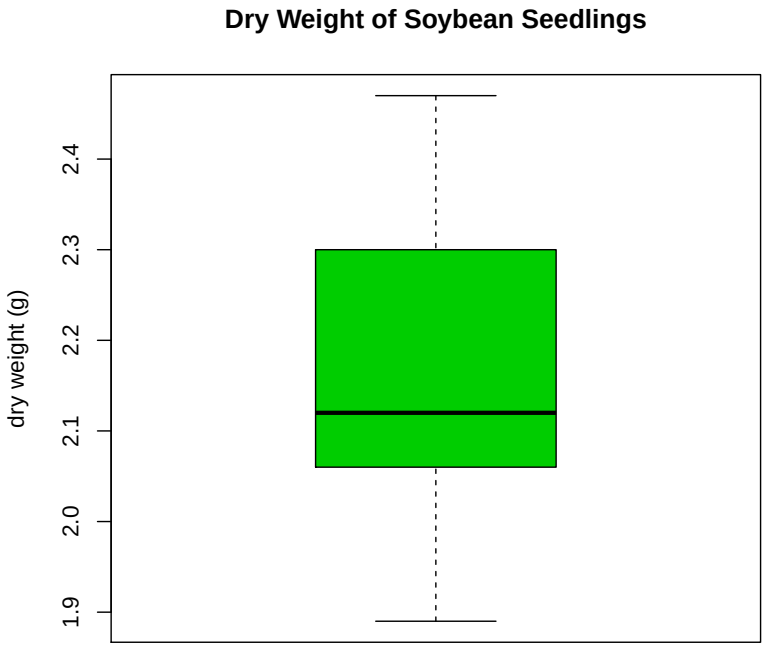


Figure 8.5: Boxplot of soybean seedlings' dry weight (checking for normal distribution).

Chapter 9

Linear Regression

9.1 Assumptions

Correlation analyses checks for a linear coherence between to or more variables. Linear regression calculates the mathematical function for a response variable influenced by one or several predicting variables.

The simplified linear model contains α as y-axis intercept, β standing for the slope and ε for the experimental error (i is the measured value number i):

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

The following assumptions are prerequisites for a linear regression:

- The **number of predicting values (x-values)** must be at least two (preferably more!)
- The **number of repetitions over the entire experiment** must be at least three.
- **Homogeneity of variances of the residuals**: Residuals shall be scattering equally around the zero line in a residual plot. The range should not get smaller in the middle nor on the endings. Homogeneity of variances might be checked with a Levene test (function `leveneTest()` coming along with the `car` package).
- **Normal distribution of residuals:** The residual plot should ideally look like a "sky full of stars" scattering around the horizontal zero line. A boxplot or QQ-Plot might also be helpful for obtaining a normal distribution but this will not offer the possibility to check for homogeneity of variances (because it is only one box present).

9.2 Implementation

9.2.1 The Function `lm()`

`lm()` is used to calculate a linear model.

```
lm(formula, data, subset, na.action, ...)
```

Data is specified with a `formula`-construct (see section 3.1). The linear model function returns intercept and slope of a straight line.

9.2.2 The Function `summary()`

`summary` returns a list containing a lot of useful information about a linear model, e.g. rough distribution of residuals, intercept and slope for a straight line.

```
summary(object, ...)
```

9.2.3 Functions Serving the Analysis of Residuals

`fitted(object, ...)` calls the expected y-values for a linear model on the regression line while `resid(object, ...)` calls the actual residuals of a linear model.

The `plot()` function followed by an `abline()` is used to investigate the distribution of residuals graphically (see section 3.3):

```
plot(x, y, ...)
abline(h = 0)
```

`x` represents a vector containing expected values whereas `y` stands for a vector with the residuals. The points should be scattering equally around the horizontal zero-line (sky full of stars).

A Quantile-Quantile-Plot is another way to visualize residuals (function `qqnorm()` with `x` as a vector containing the residuals):

```
qqnorm(x, ...)
```

`qqline()` applied on a linear model results in a straight line through the QQ-Plot.

Simple plotting of a linear model with `plot(object = lm(...))` returns four different graphs: the residual plot mentioned above, a QQ-Plot, the Scale-Location Plot[1] and Cook's distance Plot[2] .

```
plot(object, ...)
```

9.2.4 The Function `leveneTest()`

The Levene test can be used to verify the assumption of homogeneity in variances for two and more groups while it is more tolerant to deviation from the normal distribution than the F-test (comparing two samples only, `var.test`) and Bartlett's Test for homogeneity in variances (`bartlett.test()`).

The `car` package needs to be installed and loaded with `library()` for the usage of `leveneTest()`!

```
leveneTest(y, group)
```

`y` is a response variable, e.g. residuals, `group` represents a grouping vector, e.g. different treatments (this is similar to the usage of a formula construct). One has to be very careful with the data type of a grouping variable. If the vector contains `numerical` values, the

[1]The Scale-Location Plot (diagram of dispersion) plots the square root of the absolute residuals against the fitted values. It is used to check for non-constant variance.

[2]Cook's Distance is a measure for the influence of a single observation on the regression coefficient. An observation with a huge influence will change the regression coefficient considerably.

water (mm)	root dry weight (t/ha)
0	9
0	10.3
0	11.5
0	14.2
48	12.2
50	13.8
48	14
50	16.2
88	14
88	14.5
100	15
88	15.3
145	17.8
137	18
150	18.1
153	18.4
177	16.9
189	17.6
200	16.8
200	17
209	18.2
210	17
213	17.5
222	18.5
227	17.2
227	17.4
234	19.2
239	16.8

Data 9.1: Sugar beet yield response to different amounts of irrigation.

p-value might be calculated incorrect because the function is based on `anova()`. However, this problem might be solved by redefining the data type with `as.character(group)` or `as.factor(group)`. This problem is exclusively related to the functions `leveneTest()` and `anova()`. It is by the time not possible to enter a "real" `formula`-construct.

A significant p-value in the output indicates heterogeneity in variances.

9.2.5 Example Sugar Beets

9.2.5.1 Experiment

An experiment was designed to find out whether and how irrigation influences the yield of sugar beets. Seven different amounts of water (from 0 up to 250 mm) applied on four plots respectively. The real amount of water varies slightly and Data 9.1 considers only real values (Collins and Seeney, 1999, p. 207f, Dataset was read from figure 6.57 and might therefore differ from the original data.).

9.2.5.2 Statistical Analysis

The dataset is read from a *.txt file in flat file format. One column contains the irrigation, the other column contains the sugar beet yield.

```
> beets <- read.table(file = "../text/beets.txt", sep = "\t",
+ header = TRUE)
> plot(yield~water, data = beets, col = "turquoise3",
+ xlab = "irrigation (mm)", ylab = "yield (t/ha)",
+ main = "Sugar Beet Irrigation")
```

A linear regression model is created with `lm()`:

```
> beetmodel <- lm(formula = yield~water, data = beets)
```

`abline()` applied on this linear model fits a regression line to the scatter plot (see figure 9.1).

```
> abline(reg = beetmodel, col = "turquoise4")
```

9.2.5.3 Analysis of Residuals

Residuals represent the term of error in a regression model!

The following plot is created for the graphical analysis of residuals (figure 9.2):

```
> fitted.values <- fitted(object = beetmodel)
> resid.values <- resid(object = beetmodel)
> plot(x = fitted.values, y = resid.values, col = "turquoise3")
> abline(h = 0, col = "turquoise4")
```

In addition, the QQ-Plot might be used (figure 9.3):

```
> resid.values <- resid(object = beetmodel)
> qqnorm(y = resid.values, col = "black")
> qqline(y = resid.values, col = "turquoise4")
```

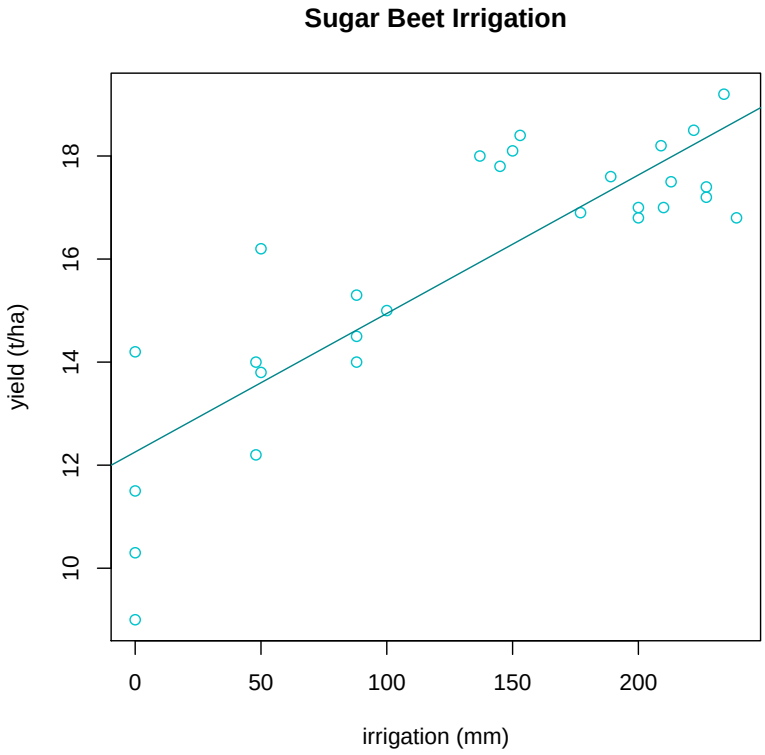


Figure 9.1: Scatterplot of sugar beet data with a fitted regression line.

Both graphs accompanied by a Scale-Location and Cook's Distance Plot are created when the linear model is plotted (figure 9.4):

```
> plot(beetmodel, col = "turquoise3")
```

- ✔ The number of **predicting values** is $7 > 2$.
- ✔ The **number of repetitions** counts four for each predicting value (this is greater than three values for the complete regression).
- ✔ **Homogeneity of variances** for the residuals is accepted (figure 9.2).
- ✔ An approximate **normal distribution** of the residuals is given in figures 9.2 and 9.3).

$\Longrightarrow$ Linear Regression.

```
> summary(object = beetmodel)

Call:
lm(formula = yield ~ water, data = beets)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2555 -0.8490 -0.0286  0.6604  2.6004
```

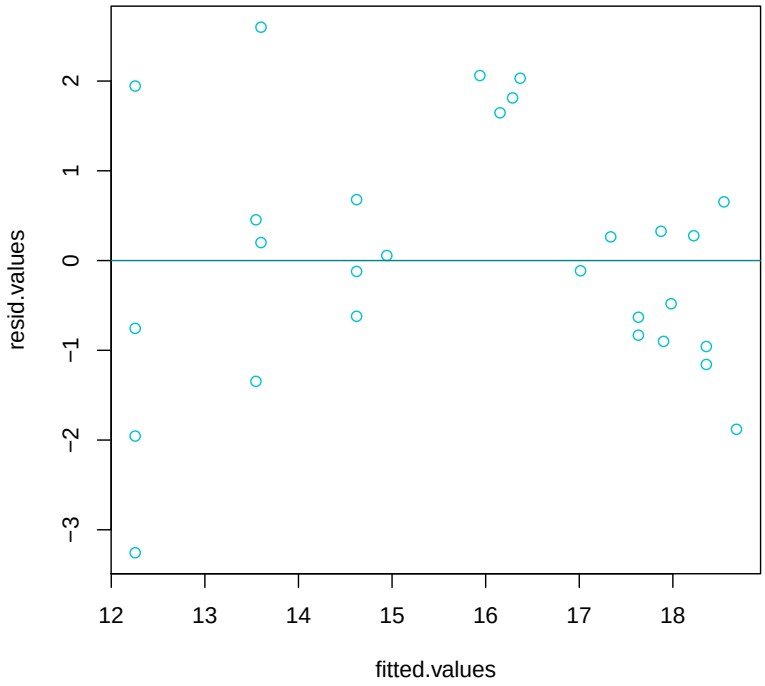


Figure 9.2: Residuals for sugar beet regression model.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.255531   0.505482  24.245   <2e-16 ***
water        0.026881   0.003261   8.244    1e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.407 on 26 degrees of freedom
Multiple R-squared:  0.7233,        Adjusted R-squared:  0.7127
F-statistic: 67.97 on 1 and 26 DF,  p-value: 1.002e-08
```

9.2.5.4 Interpretation

```
Call:
lm(formula = yield ~ water, data = beets)
```

The calculated linear model is printed.

```
Residuals:
     Min       1Q   Median       3Q      Max
-3.25553 -0.84896 -0.02857  0.66045  2.60041
```

This table gives information about the distribution of residuals in a very compact form. A linear regression created with `lm()` is only accepted if the residuals are normal distributed.

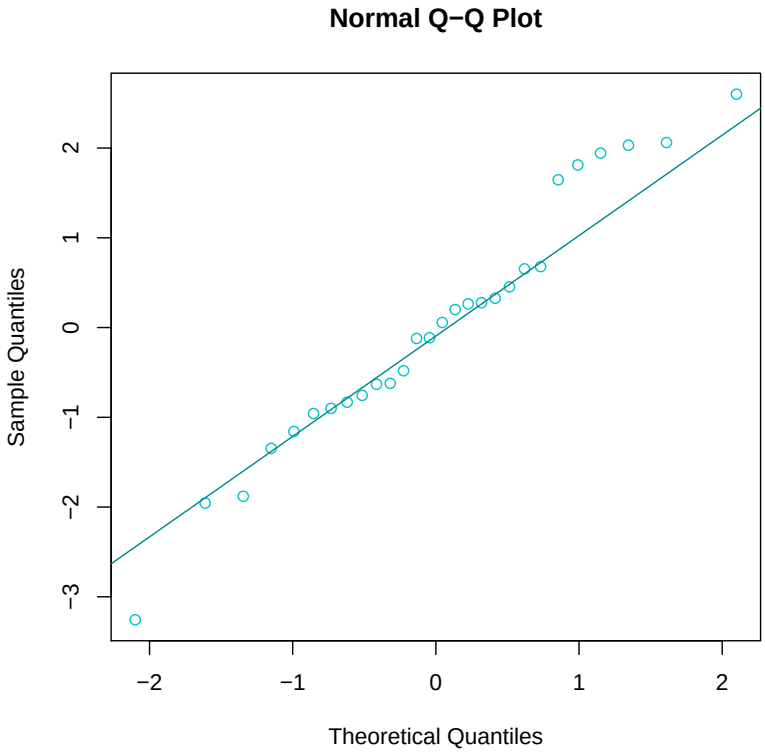


Figure 9.3: QQ-plot of sugar beet data model residuals.

That means the minimum and maximum should have roughly the same absolute value and the median is supposed to be close to zero. This is the case for the sugar beet example.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.255531   0.505482  24.245  < 2e-16 ***
water        0.026881   0.003261   8.244 1.00e-08 ***
```

`Estimate` – `(Intercept)` indicates the intercept, `water` the slope for the fitted regression line. The mathematical function is therefore:

$$y = 12.255531 + 0.026881x$$

`Std. Error` indicates the standard error for intercept and slope, `t value` presents the test statistic and `Pr(>|t|)` holds the p-value. In this example, intercept and slope are highly significant with an error probability of 5%.

```
Residual standard error: 1.407 on 26 degrees of freedom
```

This statement is an expression for the variation of the residuals around the regression line.

```
Multiple R-Squared: 0.7233,     Adjusted R-squared: 0.7127
```

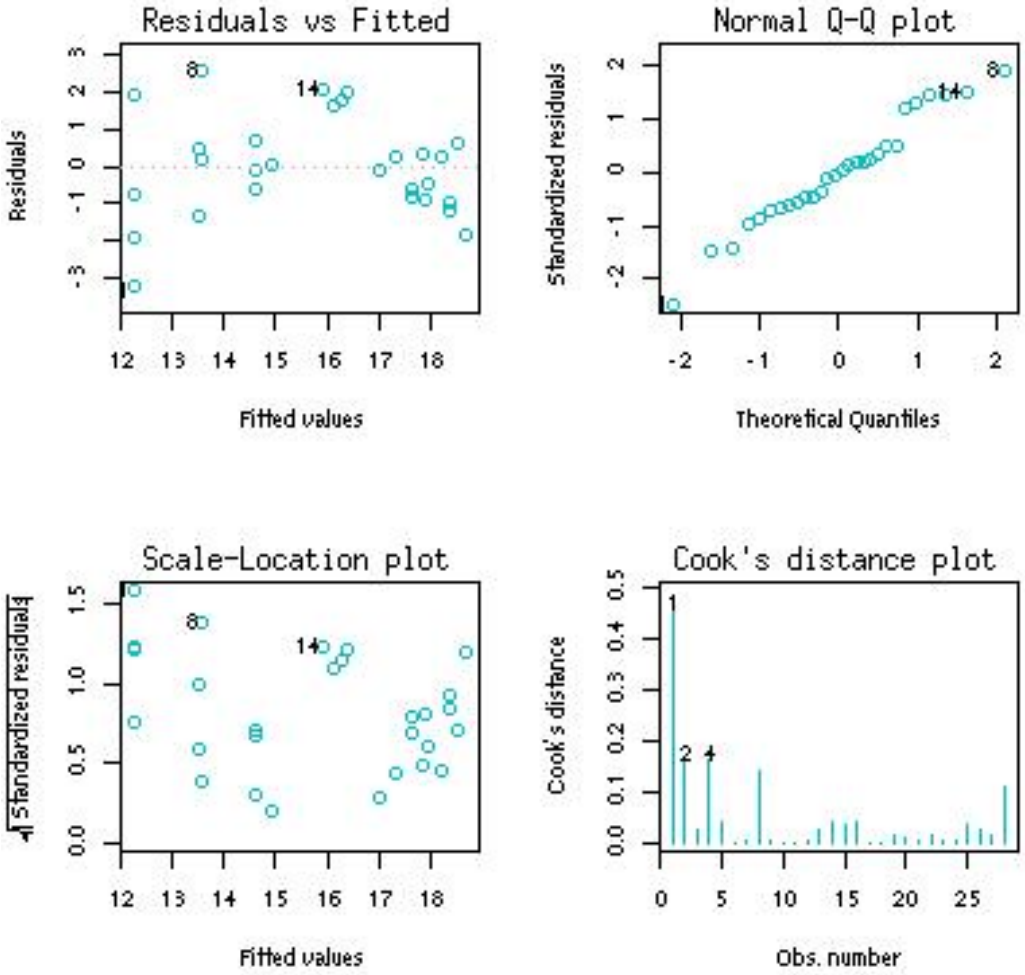


Figure 9.4: Graphs created by `plot(beetmodel)`.

R^2 represents the squared correlation coefficient according to Pearson ($R^2 = r^2$). The adjusted R^2 might be interpreted as *reduction of variance in percentage*.

```
F-statistic: 67.97 on 1 and 26 DF,  p-value: 1.002e-08
```

The F-test is calculated for the hypothesis that the regression coefficient equals zero. In this case, the test is not of interest because it duplicates information which is already present. The result is more interesting when a regression model contains more than one influencing variable.

The starcode shows the kept level of significance for each estimate "at one glance":

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

One star says "p-value smaller 0.05", two stars express "p-value smaller than 0.01" et cetera.

9.2.5.5 Confidence and Prediction Bands

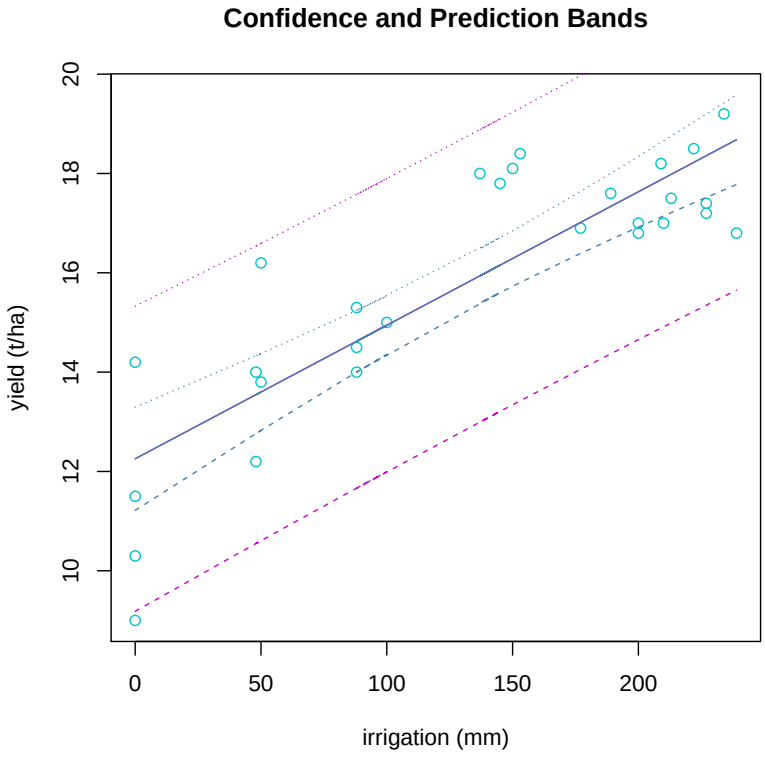


Figure 9.5: Illustration of confidence- and prediction bands for the sugar beet regression. The wide lines are prediction bands while the closer lines represent the confidence bands.

`predict()` allows the calculation of predicting data for a linear model. The parameter `interval` specifies the kind of confidence values: `confidence` stands for bands which include the regression line with a probability of 95%. The option `prediction` creates confidence data for prediction bands include the majority of all observations and show the confidence for the prediction of exact values in the future, based on this regression model.

Fertilizer (lb/acre)	Yield
100	24
100	35
100	42
100	47
100	55
200	31
200	40
200	50
200	54
200	61
300	37
300	43
300	53
300	55
300	62
400	47
400	53
400	62
400	70
400	74
500	52
500	61
500	65
500	70
500	80
600	63
600	68
600	74
600	80
600	90
700	67
700	74
700	80
700	84
700	93

Data 9.2: Yield of bread wheat dependent on different amounts of fertilizer.

Figure 9.5 shows the confidence and prediction bands:

```
> pp <- predict.lm(object = beetmodel, interval = "prediction",
+ data = beets$water)
> pc <- predict.lm(object = beetmodel, interval = "confidence",
+ data = beets$water)
```

`matlines()` plots the confidence bands:

```
> plot(x = beets$water, y = beets$yield, ylim = range(beets$yield, pc),
+ col = "turquoise3", xlab = "irrigation (mm)", ylab = "yield (t/ha)",
+ main = "Confidence and Prediction Bands")
> matlines(x = beets$water, y = pp, tly = c(1,3), col = "magenta3")
> matlines(x = beets$water, y = pc, tly = c(1,2,3), col = "steelblue")
```

`tly` indicates which columns of the `predict`-table are plotted.

9.2.6 Example Bread Wheat

9.2.6.1 Experiment

An experiment was designed to investigate the influence of different amounts of fertilizer on the yield of bread wheat. Concentrations of 100, 200, 300, 400, 500, 600 and 700 lb fertilizer/acre were applied on five randomly chosen plots respectively (Data 9.2) (Wonnacott and Wonnacott, 1990, p. 359, data was read from figure 11-1, it might slightly differ from the original data.).

9.2.6.2 Statistical Analysis

```
> wheat <- read.table(file = "../text/wheat.txt", header = TRUE,
+ sep = "\t")
> plot(yield~fertilizer, data = wheat, col = "turquoise3",
+ xlab = "Fertilizer (lb/acre)",
+ main = "Wheat Yield in a Fertilizer Experiment")
```

`lm()` fits a linear regression model:

```
> wheatmodel <- lm(formula = yield~fertilizer, data = wheat)
```

`abline()` adds a fitted regression line to the scatterplot (figure 9.6):

```
> abline(reg = wheatmodel, col = "turquoise4")
```

A boxplot is created for checking the normal distribution of the residuals (figure 9.7):

```
> resid.values <- resid(object = wheatmodel)
> boxplot(x = resid.values, col = "turquoise3",
+ main = "Boxplot of Residuals")
```

Homogeneity of variances cannot be obtained from a boxplot. Therefore, a Levene test is accomplished. The package `car` has to be installed (add-on) and loaded with `library()` for the usage of `leveneTest()`!

Figure 9.6: Yield of bread wheat dependent on the amount of fertilizer applied to the plot (lb/acre).

```
> library(car)
> lev <- data.frame(res = resid.values,
+ group = as.character(wheat$fertilizer))
> leveneTest(y = lev$res, group = lev$group)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  6  0.0623 0.9989
      28
```

- ✔ The number of **predicting values** is seven (> 2).
- ✔ **Five repetitions** for each x-value fulfill the requirement of at least three repetitions over all predictors.
- ✔ **Homogeneity of variances** is assumed due to a non significant Levene test result.
- ✔ The residuals are approximately **normal distributed** (figure 9.7).

$\Longrightarrow$ `wheatmodel` is fitting well for the regression analysis of bread wheat data.

```
> summary(object = wheatmodel)

Call:
lm(formula = yield ~ fertilizer, data = wheat)
```

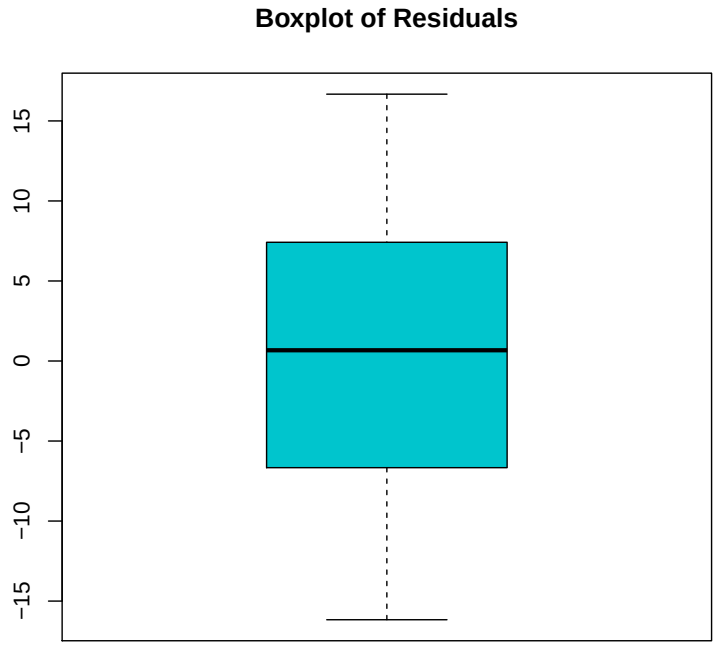


Figure 9.7: Boxplot of residuals for an investigation of distribution.

```
Residuals:
     Min       1Q   Median       3Q      Max
-16.1643  -6.6643   0.6714   7.4179  16.6714

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 33.000000   3.822172   8.634 5.60e-10 ***
fertilizer   0.067214   0.008547   7.864 4.57e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.11 on 33 degrees of freedom
Multiple R-squared:  0.6521,        Adjusted R-squared:  0.6415
F-statistic: 61.85 on 1 and 33 DF,  p-value: 4.574e-09
```

9.2.6.3 Interpretation

The mathematical equation is:

$$y = 0.067214x + 33$$

The intercept as well as the slope are highly significant to a confidence level of 95%.

✎ Exercise 10

Sulphur is efficiently used fighting potato scab. Researchers investigated the effect of different sulphur concentrations on the plant disease. Four concentrations (0, 300, 600 and 1200 pounds/acre) have been applied on four plots respectively. The sum of surface damage by scab has been counted for 100 randomly chosen potatoes from each plot (Data 10.3) (Pearce, 1983, p. 46, Data is not complete, the actual experiment included observations in spring and fall.), original experiment published in Cochran and Cox (1950).

Are the given data fitting for a regression analysis with the linear model? Are the residuals normal distributed?

If so, which are intercept and slope? Is the regression significant with an error probability of 5%?

Plot confidence and prediction bands!

sulphur (pound/ acre)	**scab (%)**
0	18
0	30
0	24
0	29
300	9
300	9
300	16
300	4
600	18
600	10
600	18
600	16
1200	4
1200	10
1200	5
1200	4

Data 10.3: Sulphur treatment of potato scab.

Chapter 10

ANOVA

10.1 Assumptions

Analysis of variances (ANOVA) is used to investigate the effect of one or several categorical predicting variables on one or several random variables, e.g. the influence of different fertilizers and varieties on the variable yield. The ANOVA is not significant when the variances are overlapping each other.

Example for a model – two-factorial ANOVA with interaction:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}$$

Y_{ijk} is the random response variable, μ represents the expected value, α_i stands for the effect of the i^{th} level of factor A, β_j is the effect of the j^{th} level of factor B, $(\alpha\beta)_{ij}$ represents the interaction, ε_{ijk} stands for the experimental error, k is the number of repetitions.

Assumptions for an ANOVA are:

- **Normal distribution of** ε_{ijk} within the respective groups → Plot of residuals, dots should be normal distributed above and below the zero-line for all categories. A boxplot might serve this purpose as well.
- **Homogeneity of variances** of the residuals → Levene-test and/or plot of residuals/boxplot.
- **Independent data.**

An example for the hypotheses of an experiment with three levels of factor A and two levels of factor B is given below:

$\exists$ is read as *there exists*.

$$\begin{array}{lll} H_0^1: & \mu_{A1} = \mu_{A2} = \mu_{A3} & H_0^2: \mu_{B1} = \mu_{B2} \\ H_1^1: & \exists \text{ at least one } \mu_{A_i} \neq \mu_{A_j} & H_1^2: \mu_{B1} \neq \mu_{B2} \end{array}$$

10.2 Implementation

10.2.1 Extension for the Function `lm()`

An introduction to `lm()` is given in section 9.2.1. The `formula`-construct for ANOVA is written as follows:

```
> lm(formula = target~treatment.1+treatment.2+treatment.1:treatment.2,
+ data = dataset)
```

Influencing variables are combined with `+` while `:` forms an interaction term.

10.2.2 The Function `anova()`

`anova()` calculates the table of variances.

```
anova(object, ...)
```

`object` is a linear model. This model can either be saved in an object and used as `anova(objectname)` or it might be integrated into the function, directly: `anova(lm(...))`.

10.2.3 Example Corn

10.2.3.1 Experiment

Do methods of biological plant protection reduce the effect of insects on corn ears efficiently? Researchers compared the ear weight of corn for five different biological treatments: the beneficial nematode *Steinernema carpocapsae*, the wasp *Trichogramma pretiosum*, a combination of those first two, the bacterium *Bacillus thuringiensis* and a non treated control group. Ears of corn were randomly sampled from each plot and weighed (table 10.1) (Martinez, 1998) cited according to Samuels and Witmer (2003, p. 463f, the data presented here are a random sample from a larger study.).

Nematode	Wasp	Nematode & Wasp	Bacterium	Control
16.5	11.0	8.5	16.0	13.0
15.0	15.0	13.0	14.5	10.5
11.5	9.0	12.0	15.0	11.0
12.0	9.0	10.0	9.0	10.0
12.5	11.5	12.5	10.5	14.0
9.0	11.0	8.5	14.0	12.0
16.0	9.0	9.5	12.5	11.0
6.5	10.0	7.0	9.0	18.5
8.0	9.0	10.5	9.0	9.5
14.5	8.5	10.5	9.0	17
7.0	8.0	13.0	6.5	10.0
10.5	5.0	9.0	8.5	11.0

Table 10.1: Weight of corn ears (ounces).

10.2.3.2 Statistical Analysis

```
> corn <- read.table(file = "../text/corn.txt", sep = "\t",
+ header = TRUE)
```

Data is visualized in boxplots (figure 10.1):

```
> boxplot(formula = response~treatment, data = corn, col = "white",
+ main = "Plot of the Corn Data", ylab = "weight (ounces)",
+ names = c("nem","wasp","nem+wasp","bac","control"))
```

This experiment includes only one influencing factor. Therefore, the linear model neither contains additive factors nor an interaction:

```
> corn.model <- lm(formula = response~treatment, data = corn)
```

Graphical visualization of the residuals for `corn.model` (figure 10.2):

```
> fitted.values <- fitted(object = corn.model)
> resid.values <- resid(object = corn.model)
> plot(x = fitted.values, y = resid.values, col = "black")
> abline(h = 0, col = "blue4")
```

A boxplot might also help during the investigation of the distribution (figure. 10.3):

```
> boxplot(x = resid.values, col="white", main="Residuals")
```

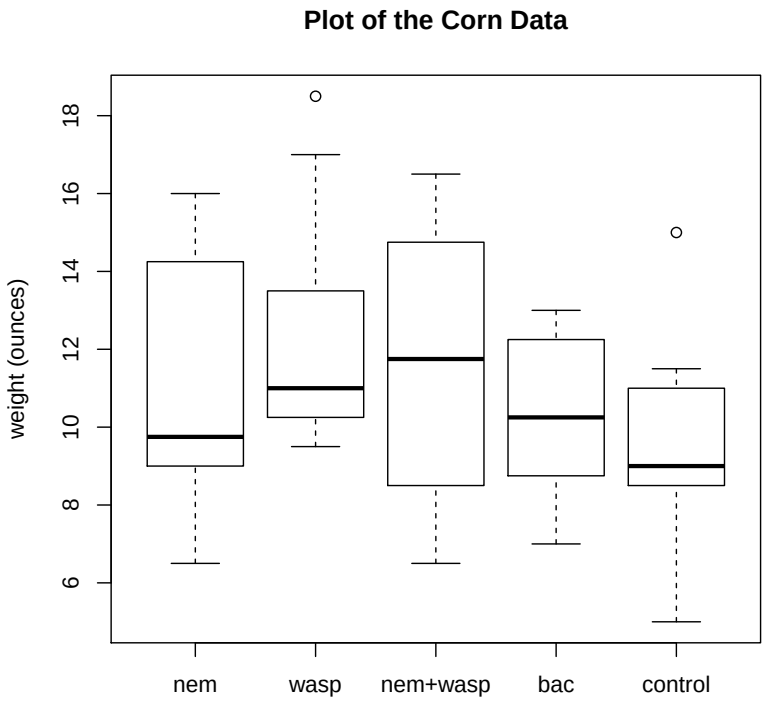


Figure 10.1: Plot of corn data.

A Levene test is used for verification of homogeneity in variances of the residuals (package `car` needs to be installed and loaded!):

```
> library(car)
> lev <- data.frame(res = resid.values, group = corn$treatment)
> leveneTest(y = lev$res, group = lev$group)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  4  1.1028 0.3645
      55
```

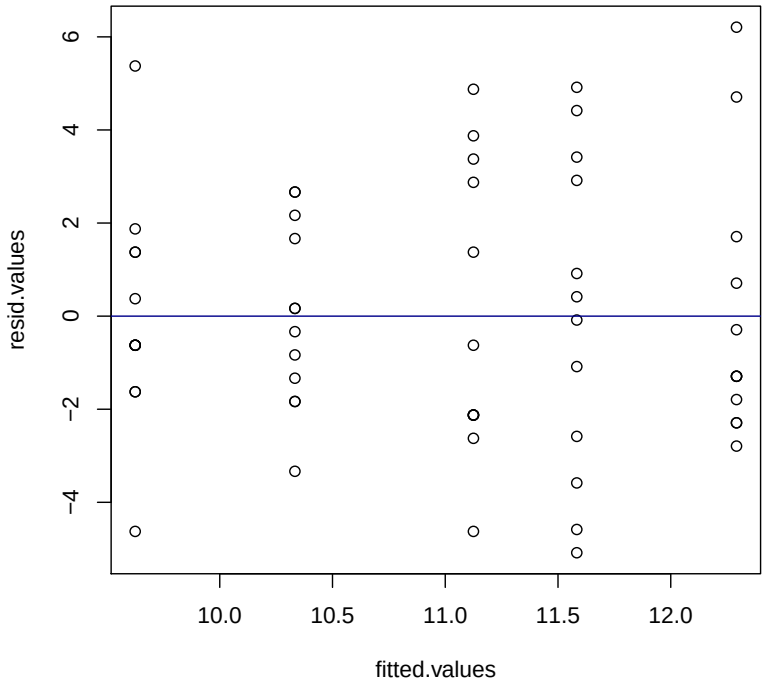


Figure 10.2: Residual plot of `corn.model` for an evaluation of normal distribution.

The null hypothesis - homogeneity of variances - is not rejected.

- ✔ **Homogeneity of variances** (Levene test).
- ✔ **Normal distribution** of residuals (figure 10.2 and 10.3).
- ✔ **Independent data**.

$\Longrightarrow$ ANOVA with one factor. Hypotheses:

$$H_0: \quad \mu_{nem} = \mu_{wasp} = \mu_{nem+wasp} = \mu_{bac} = \mu_{control}$$
$$H_1: \quad \exists \text{ at least one } \mu_{treatment} \neq \mu_{treatment'}$$

```
> anova(object = corn.model)

Analysis of Variance Table

Response: response
          Df Sum Sq Mean Sq F value Pr(>F)
treatment  4  52.31 13.0771  1.6461 0.1758
Residuals 55 436.94  7.9443
```

10.2.3.3 Interpretation

First of all, the variance table's header and the response variable are displayed.

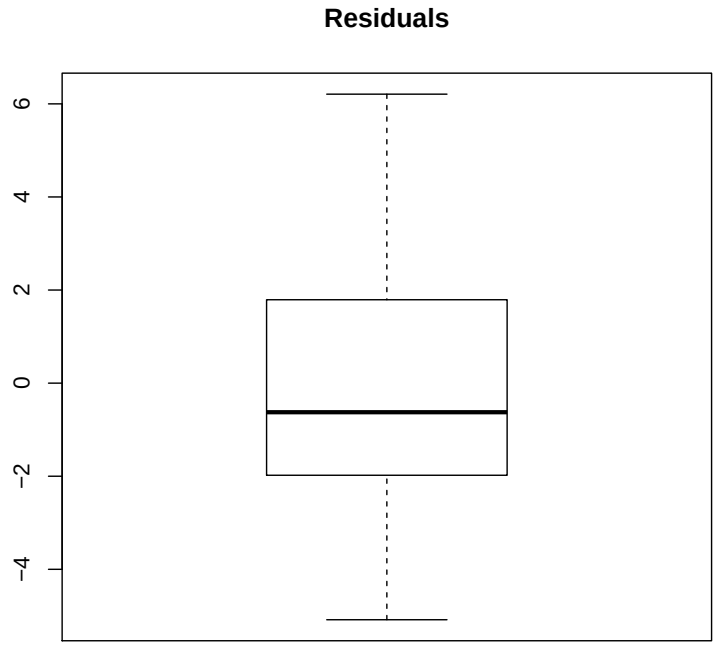


Figure 10.3: Boxplot of residuals.

```
          Df Sum Sq Mean Seq F value Pr(>F)
treatment  4  52.31   13.08  1.6461 0.1758
Residuals 55 436.94    7.94
```

The first column names the rows for the predictor treatment and the Residuals. `Df` presents the degrees of freedom while `Sum Sq` gives the sums of squares for treatment and residuals, `Mean Sq` gives the mean squares and the F-value returns the test statistic (which is the mean square for the factor divided by the mean square for the error) . The p-value is given in the column `Pr(>F)`. For this model, the p-value is greater than 0.05 which leads to the conclusion that the null hypothesis (no difference in biological treatments) is kept: It was not possible to verify a significant difference in yield for different biological treatments for a confidence level of 0.95.

10.2.4 Example Soybeans (3)

10.2.4.1 Experiment

"A plant physiologist investigated the effect of mechanical stress on the growth of soybean plants. Individually potted seedlings were randomly allocated to four treatment groups of 13 seedlings each. Seedlings in two groups were stressed by shaking for 20 minutes twice daily, while two control groups were not stressed. Thus, the first factor in the experiment was presence or absence of stress with two levels. Also, plants were grown in either low or moderate light" $\Longrightarrow$ second factor. The leaf areas of each plant are given in table 10.2 (Pappas and Mitchell, 1984), rawdata published in Samuels and Witmer (2003, p. 491, the author indicates that the original experiment contained more than four treatments.).

10.2.4.2 Statistical Analysis

Control Low Light	Stress Low Light	Control Moderate Light	Stress Moderate Light
264	235	314	283
200	188	320	312
225	195	310	291
268	205	340	259
215	212	299	216
241	214	268	201
232	182	345	267
256	215	271	326
229	272	285	241
288	163	309	291
253	230	337	269
288	255	282	282
230	202	273	257

Table 10.2: Leaf area (cm^2) of the soybean plants.

```
> soybeans <- read.table(file = "../text/soybeans.txt", sep = "\t",
+ header = TRUE)
```

The linear model considers the influence of light and stress as well as an interaction term on the leaf area of soybean plants:

```
> model <- lm(formula = response~treatment.B+treatment.A+treatment.A:
+ treatment.B, data = soybeans)
```

Graphical visualization of residuals (figure 10.4):

```
> fitted.values <- fitted(object = model)
> resid.values <- resid(object = model)
> plot(x = fitted.values, y = resid.values, col = "black")
> abline(h = 0, col = "blue4")
```

Levene test for verification of the residual's homogeneity of variances for the different groups:

```
> library(car)
> lev <- data.frame(res = resid.values, group = rep(c("low.light.c",
+ "low.light.s","mod.light.c","mod.light.s"), each = 13))
> leveneTest(y = lev$res, group = lev$group)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  3  0.1963 0.8984
      48
```

The p-value is greater 0.05. Therefore, the null hypothesis (homogeneity of variances) is not rejected.

✔ **Homogeneity of variances** of the residuals (Levene test).

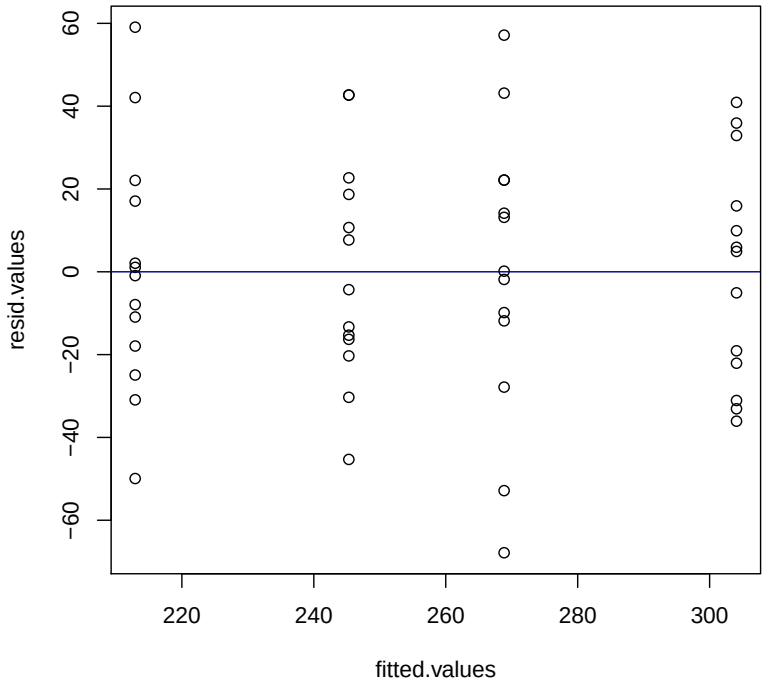


Figure 10.4: Residual plot of soybean data.

✔ Approximate **normal distribution** of residuals (figure 10.4).

✔ **Independent data** (randomized groups).

$\Longrightarrow$ Analysis by ANOVA. Question: Does mechanical stress and different levels of light lead to at least one difference between the experiment groups? Hypotheses (including the interaction):

$$H_0^A : \mu_{stress} = \mu_{nostress} \qquad H_0^B : \mu_{light} = \mu_{dark}$$
$$H_1^A : \mu_{stress} \neq \mu_{nostress} \qquad H_1^B : \mu_{light} \neq \mu_{dark}$$

$$H_0^{A'B}: \quad \mu_{stressfactor,lightfactor} = \mu_{stressfactor} + \mu_{lightfactor} - \mu$$
$$H_1^{A'B}: \quad \mu_{stressfactor,lightfactor} \neq \mu_{stressfactor} + \mu_{lightfactor} - \mu$$

```
> anova(object = model)

Analysis of Variance Table

Response: response
                      Df Sum Sq Mean Sq F value    Pr(>F)
treatment.B            1  14858   14858 16.5954 0.0001725 ***
treatment.A            1  42752   42752 47.7490  1.01e-08 ***
treatment.B:treatment.A  1     26      26  0.0294 0.8645695
Residuals             48  42976     895
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

10.2.4.3 Interpretation

The table of variances (Interpretation instructions are given in the previous example) shows that the factors light treatment and seismic stress have a significant influence on the leaf area of soybean seedlings. There exists no significant interaction. In this experiment, it can be seen "at one glance" where the differences between the groups are located because there are only two respective levels.

10.2.5 Example Alfalfa

10.2.5.1 Experiment

"Researchers were interested in the effect that acid rain has on the growth rate of alfalfa plants. They created three treatment groups in an experiment: low acid, high acid and control. The response variable in their experiment was the average height of the alfalfa plants in a Styrofoam cup after five days of growth. (The observational unit was a cup, rather than individual plants.) They had 5 cups for each of the 3 treatments, for a total of 15 observations. However, the cups were arranged near a window and they wanted to account for the effect of differing amounts of sunlight. Thus, they created 5 blocks and randomly assigned the 3 treatments within each block", as shown in table 10.3. The data is given in table 10.4 (Neumann et al., 2001) cited according to Samuels and Witmer (2003, pp. 487).

	Block 1	**Block 2**	**Block 3**	**Block 4**	**Block 5**
win	high	control	control	control	high
d	control	low	high	low	low
ow	low	high	low	high	control

Table 10.3: Block design of an alfalfa experiment.

	Low acid	**High Acid**	**Control**
Block 1	1.58	1.10	2.47
Block 2	1.15	1.05	2.15
Block 3	1.27	0.50	1.46
Block 4	1.25	1.00	2.36
Block 5	1.00	1.50	1.00

Table 10.4: Alfalfa data: Height of plants for each cup after 5 days measured in cm.

10.2.5.2 Statistical Analysis

```
> alfalfa <- read.table(file = "../text/alfalfa.txt", sep = "\t",
+ header = TRUE)
```

A linear model accounting for the acid treatment and the block design:

```
> alfalfa.model <- lm(formula = height~acid+block, data = alfalfa)
```

Boxplot of residuals (figure 10.5):

```
> resid.values <- resid(object = alfalfa.model)
> boxplot(x = resid.values, col = "white", main="Residuals")
```

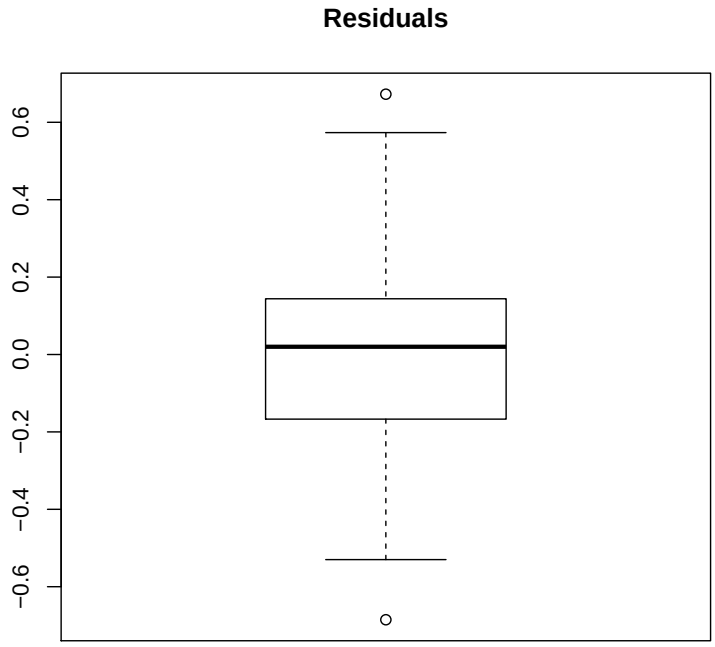


Figure 10.5: Boxplot of residuals for `alfalfa.model`.

Levene test for the verification of homogeneity of variances within the residuals:

```
> library(car)
> lev <- data.frame(res = resid.values, group = alfalfa$acid)
> leveneTest(y = lev$res, group = lev$group)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  2  1.7928 0.2083
      12
```

- ✔ **Homogeneity of variances** (Levene test with a p-value greater than 0.05, null-hypothesis is kept).
- ✔ Approximate **normal distribution** of residuals (figure 10.5).
- ✔ **Independent data** (randomized block design).

$\Longrightarrow$ ANOVA with the following hypotheses:

H_0^1 : $\mu_{low} = \mu_{high} = \mu_{control}$

H_1^1 : $\exists$ at least one $\mu_{acid} \neq \mu_{acid'}$

H_{0^2} : $\mu_{block1} = \mu_{block2} = \mu_{block3} = \mu_{block4} = \mu_{block5}$

H_1^2 : $\exists$ at least one $\mu_{block} \neq \mu_{block'}$

```
> anova(object = alfalfa.model)
```

```
Analysis of Variance Table

Response: height
          Df  Sum Sq Mean Sq F value  Pr(>F)
acid       2 1.98601 0.99301  5.5066 0.02202 *
block      1 0.30805 0.30805  1.7083 0.21787
Residuals 11 1.98363 0.18033
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The table of variances shows that acid influences the height of alfalfa plants significantly with an error probability of 5%. The exact location of the difference cannot be obtained from an ANOVA because there are three treatments compared with each other. A multiple comparison test as described in the next chapter might solve this problem.

10.2.6 Example Cress (1)

10.2.6.1 Experiment

A student experiment was designed to investigate the influence of different light qualities on the growth rate of cress (*Lepidium sativum*). Six new lamps accompanied by the SON-T lamp (widely used in horticulture) were compared. 15 plants were randomly chosen from three blocks per lamp type and the fresh weight was measured after eight days (Norlinger and Hoff, 2004), data is printed in appendix B.

10.2.6.2 Statistical Analysis

ANOVA is chosen to analyse whether there exists a significant difference in weight at different light treatments.

```
> cress <- read.table("../text/cress.txt", sep="\t", dec = ",",
+ header = TRUE)
```

The linear model accounts for the influence of light and block on the fresh weight. The residuals are plotted in figure 10.6 (approximate normal distribution).

```
> cress.model <- lm(formula = weight~light+block, data = cress)
> fitted.values <- fitted(object = cress.model)
> resid.values <- resid(object = cress.model)
> plot(x = fitted.values, y = resid.values, col = "black")
> abline(h = 0, col = "blue4")
```

```
> library(car)
> lev <- data.frame(res = resid.values, group = cress$light)
> leveneTest(y = lev$res, group = lev$group)
```

```
Levene's Test for Homogeneity of Variance (center = median)
       Df F value Pr(>F)
group   5  1.5068 0.1879
      264
```

The Levene test shows that the null hypothesis of homogeneity in variances is not rejected to an error probability of 5%.

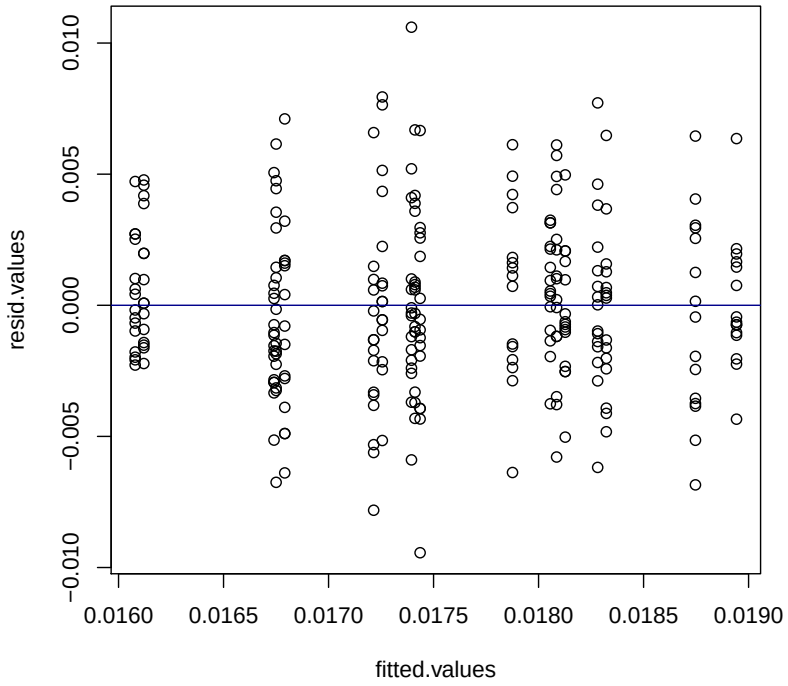


Figure 10.6: Residual plot of `cress.model` for a conclusion about the normal distribution.

- ✔ **Homogeneity in variances** of residuals (Levene test).
- ✔ Approximate **normal distribution** of residuals (figure 10.6).
- ✔ **Independency of data** is assumed.

$\Longrightarrow$ ANOVA with the hypotheses:

H_0^1: $\mu_{red} = \mu_{daylight} = \mu_{SON_T} = \mu_{white} = \mu_{blue} = \mu_{green}$
H_1^1: $\exists$ at least one $\mu_{light_i} \neq \mu_{light_j}$
H_0^2: $\mu_{block1} = \mu_{block2} = \mu_{block3}$
H_1^2: $\exists$ at least one $\mu_{block_i} \neq \mu_{block_j}$

```
> anova(object = cress.model)

Analysis of Variance Table

Response: weight
           Df     Sum Sq    Mean Sq F value  Pr(>F)
light       5 0.00015253 3.0506e-05  2.9121 0.01409 *
block       2 0.00002469 1.2347e-05  1.1787 0.30931
Residuals 262 0.00274459 1.0475e-05
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

10.2.6.3 Conclusion

With an error probability of 5%, there exists at least one significant difference in the fresh weight of cress plants. No significant block influence was obtained.

A multiple comparison test will be used in section 11.2.5 to investigate the location of the difference(s).

✎ Exercise 11

A petroleum gel was applied on Cherry Laurel leaves in order to investigate the effect on leaf transpiration. 16 leaves were chosen and divided randomly in four groups. The first group served as a control while gel was applied on the top side of leaves in the second group, on the lower side of leaves in the fourth group and on both sides of leaves in the third group. The weight of each leaf was measured. The leaves were hanging at a shady place with good air circulation for three days and the weight was measured afterwards, again. The loss of water is presented in table 10.5 (Bishop, 1980, p. 56).

Control	Top	Bottom	Both
86	41	25	13
108	44	35	11
118	40	37	13
79	52	26	13

Table 10.5: Lost of water in Cherry Laurel leaves ($\frac{mg}{cm^3}$) during three days.

Is the data obtained by this experiment suiting for analysis of variances? If so, formulate the hypotheses and do an ANOVA!

Chapter 11

Multiple Comparison Tests

11.1 Assumptions

ANOVA looks for "at least one" significant difference within several levels (treatments). On the other hand, Multiple Comparison Tests (MCP) check the pairwise differences of all indicated groups and show the exact locations. (ANOVA might be used as a pre-test for an MCP but this is not a necessity. If the ANOVA displays a significant interaction, the postulated independency is no longer granted. In this case, the pairwise differences for one factor are calculated for each level of the other factor!)

In principle, MCPs are based on the same assumptions as a common t-Test. Important are:

- **Normal distribution** within the respective groups (boxplots).
- **Homogeneity of variances** between the different treatments (Levene test, boxplots)
- **Independency of data** e.g. no significant interaction in ANOVA, in addition see chapter 5.

11.1.1 Tukey-Procedure

The "all pairs comparison" according to Tukey compares all groups with each other.

11.1.2 Dunnett-Procedure

The "many to one" comparison according to Dunnett compares all groups to one single group, usually the control.

11.2 Implementation

The packages for multiple comparison procedures are currently not included in the R base installation. Therefore, `mvtnorm` and `multcomp` have to be installed and loaded with `library()`.

11.2.1 The Function `glht()`

`glht()` is the abbreviation for *general linear hypothesis*. Among other purposes, this function can be used to do multiple comparison of means tests.

```
glht(model, lnfct = mcp(grouping.varitable = c("Dunnett", "Tukey"),
             alternative=c("two.sided","less", "greater"))
```

`model` is a model that for the purpose of comparing means was e.g. composed using `aov()`.

`linfct` allows to specify the hypothesis. This can be done by specifying e.g. a contrast matrix. If you do not want to bother learning details about contrast matrices, you can use the function `mcp` to compose a contrast matrix automatically. In this case, `grouping.variable` must be the name of the variable that was used as a grouping variable in `model`. (`Tukey` and `Dunnett` are not the only available contrast matrices, consult the help page of `mcp` for further options.)

`type` specifies whether a procedure according to `Dunnett` or `Tukey` is calculated. (There are more methods available which are not discussed here!)

`alternative` should be known from other tests, now. It specifies whether a one- or two-sided test is calculated.

11.2.2 The Function `confint()`

Confidence intervals are calculated by a separate function called `confint()`.

```
confint(object, level=0.95)
```

`object` is the output of `glht`.

`level` sets the confidence level. The default value is 95%.

11.2.3 The Function `summary()`

`summary()` applied on an object containing the output of `glht()` returns the detailed test results.

```
> summary(object)
```

11.2.4 Example Melons (1)

11.2.4.1 Experiment

The yield of four different melon varieties was compared in an experiment. Each variety was planted in six completely randomized blocks (Data 11.1) (Mead et al., 2003, p. 58).

Variety	Yield
A	25.12
A	17.25
A	26.42
A	16.08
A	22.15
A	15.92
B	40.25
B	35.25
B	31.98
B	36.52
B	43.32
B	37.10
C	18.30
C	22.60
C	25.90
C	15.05
C	11.42
C	23.68
D	28.55
D	28.05
D	33.20
D	31.68
D	30.32
D	27.58

Data 11.1: A melon experiment.

11.2.4.2 Statistical Analysis

```
> melon <- read.table(file = "../text/melon.txt", sep = "\t",
+ header = TRUE)
```

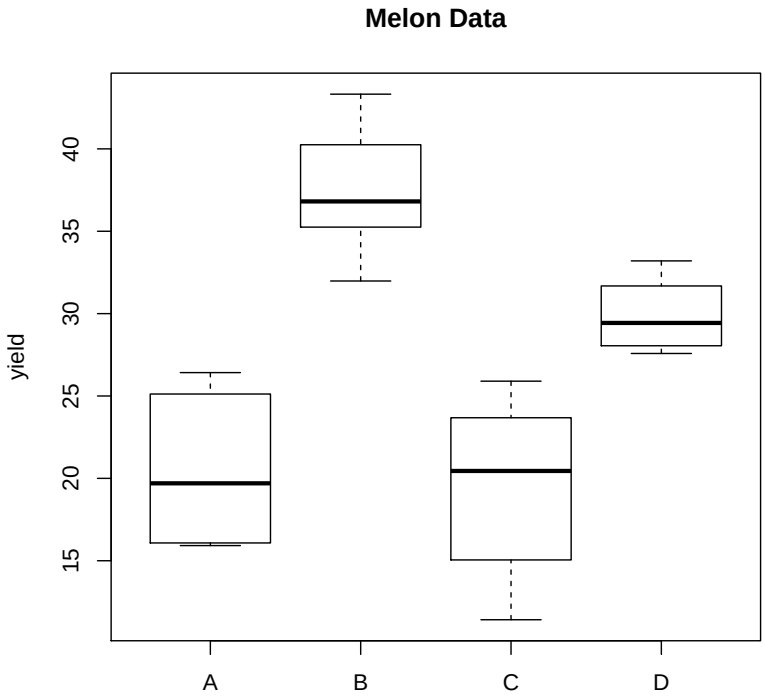


Figure 11.1: Boxplots of melon data.

A boxplot serves the determination of normal distribution and homogeneity in variances of all groups (figure 11.1):

```
> boxplot(formula = yield~variety, data = melon, col = "white",
+ main = "Melon Data", ylab = "yield")
```

Implementation of the Levene test for a verification of homogeneity in variances:

```
> library(car)
> leveneTest(y = melon$yield, group = melon$variety)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  3  2.0901 0.1337
      20
```

- ✔ Approximate **normal distribution** is accepted (figure 11.1).
- ✔ **Homogeneity of variances** is assumed (Levene test and figure 11.1).

$\Longrightarrow$ Data is suiting for the evaluation with a MCP. The question is whether there is a difference between all groups. No control has been nominated $\Longrightarrow$ Tukey procedure. For the reason that no tendency is known, a two-sided test is calculated. Hypotheses:

H_0: $\mu_A = \mu_B = \mu_C = \mu_D$
H_1: $\mu_A \neq \mu_B$
$\mu_A \neq \mu_C$
$\mu_A \neq \mu_D$
$\mu_B \neq \mu_C$
$\mu_B \neq \mu_D$
$\mu_C \neq \mu_D$

11.2.4.3 The Implementation of `glht()`

`glht()` and `summary()` are used to compute p-values:

```
> library(mvtnorm)
> library(multcomp)
> melon.model <- aov(formula = yield~variety, data = melon)
> mcmp <- glht(melon.model, linfct = mcp(variety = "Tukey"))
> summary(mcmp)

         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = yield ~ variety, data = melon)

Linear Hypotheses:
           Estimate Std. Error t value Pr(>|t|)
B - A == 0  16.9133     2.4754   6.833  < 0.001 ***
C - A == 0  -0.9983     2.4754  -0.403  0.97721
D - A == 0   9.4067     2.4754   3.800  0.00569 **
C - B == 0 -17.9117     2.4754  -7.236  < 0.001 ***
D - B == 0  -7.5067     2.4754  -3.033  0.03088 *
D - C == 0  10.4050     2.4754   4.203  0.00242 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

Below the heading of the output, you see the type of contrast that was used. Subsequently, the anova model is printed. The first column of the output table contains the corresponding pair-hypothesis. The second column contains the estimate for the difference in mean between the two samples. The column `Std. Error` returns the standard deviation between estimate and true values. The column `p value` states multiplicity adjusted p values.

The results are interpreted as usual: if the p-value is smaller than or equal to 0.05, the alternative hypothesis is accepted with an α-error of 5%. In this particular case, the samples A and B, A and D, B and C, B and D, C and D are significantly different.

11.2.4.4 The Implementation of `confint`

The implementation of `confint()` creates multiplicity adjusted confidence intervals:

```
> melon.int <- confint(mcmp)
> melon.int
```

```
         Simultaneous Confidence Intervals

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = yield ~ variety, data = melon)

Quantile = 2.7981
95% family-wise confidence level


Linear Hypotheses:
           Estimate lwr      upr
B - A == 0  16.9133   9.9870  23.8397
C - A == 0  -0.9983  -7.9247   5.9280
D - A == 0   9.4067   2.4803  16.3330
C - B == 0 -17.9117 -24.8380 -10.9853
D - B == 0  -7.5067 -14.4330  -0.5803
D - C == 0  10.4050   3.4787  17.3313
```

11.2.4.5 Interpretation

The header is followed by the called test type and model. The subsequent table contains the lower (`lwr`) and upper (`upr`) boundaries of multiplicity adjusted confdence intervals for each pair of samples.

11.2.4.6 Plotting of Confidence Intervals

It is very easy to plot the confidence intervals with `plot(confint())` (figure 11.2):

```
> plot(x = melon.int, col = "purple")
```

11.2.4.7 Conclusion

The question was whether and where significant differences in means are located (with an error probability of 5%). The simultaneous confidence intervals show that variety A differs significantly in yield from the varieties B and D. Furthermore, variety B differs significantly from variety C and D and variety C and D differ also. This result is congruent with the p-values adjusted according to Bonferroni.

A new question arises: Which variety has the highest yield? This can be read from the confidence intervals without calculating further tests. The positive confidence intervals might be read as *greater than* and the negative ones might be read as *smaller than*. This leads to the following conclusion:

B $>$ A, D $>$ A, D $>$ C, C $<$ B and D $<$ B.

A significance for those hypotheses for an confidence level of 95% is kept because the p-values will be divided by two for a one-sided test and that means they are smaller than 0.05, anyway.

This leads to the conclusion that variety B has the highest yield.

However, it is possible to calculate a new one-sided test with new confidence intervals for this purpose, of course.

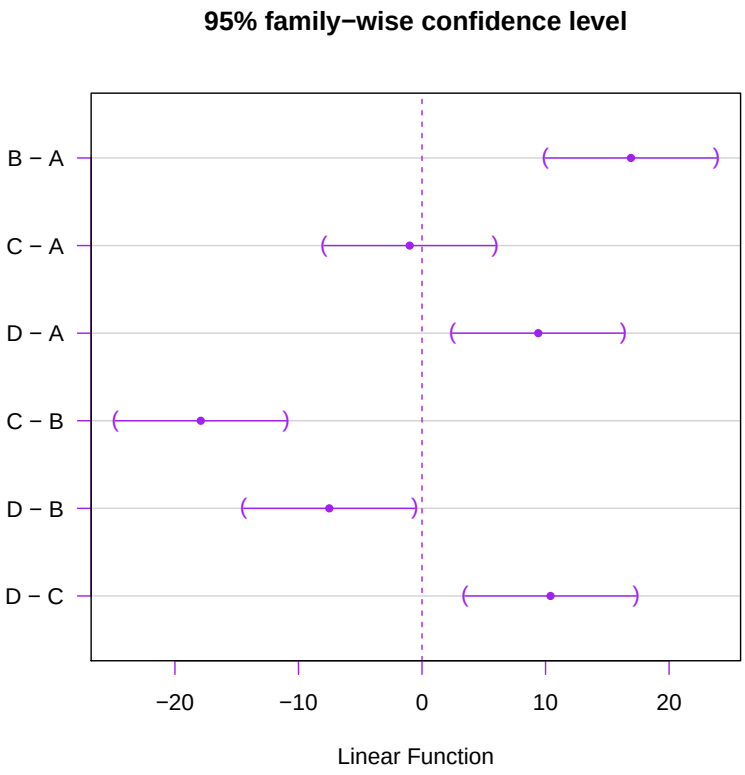


Figure 11.2: Confidence intervals for a Tukey test calculated for the melon experiment.

11.2.5 Example Cress (2)

In section 10.2.6, the conclusion from an ANOVA was that different light qualities affect the fresh weight of cress plants. An MCP according to Tukey is used to locate the difference(s).

The normal distribution of the cress data is determined with boxplots (figure 11.3) and a Levene test is calculated additionally to verify the homogeneity in variances between the different groups:

```
> boxplot(formula = weight~light, data = cress, col = "white",
+ main = "Cress Data", ylab = "fresh weight (mg)")
> library(car)
> leveneTest(y = cress$weight, group = cress$light)

Levene's Test for Homogeneity of Variance (center = median)
       Df F value  Pr(>F)
group   5  1.8985 0.09489 .
      264
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- ✔ Due to outliers in figure 11.3, approximate **normal distribution** is accepted conditionally, only.
- ✔ The hypothesis about **homogeneity of variances** is kept to a confidence level of 95% → homogeneity in variances.

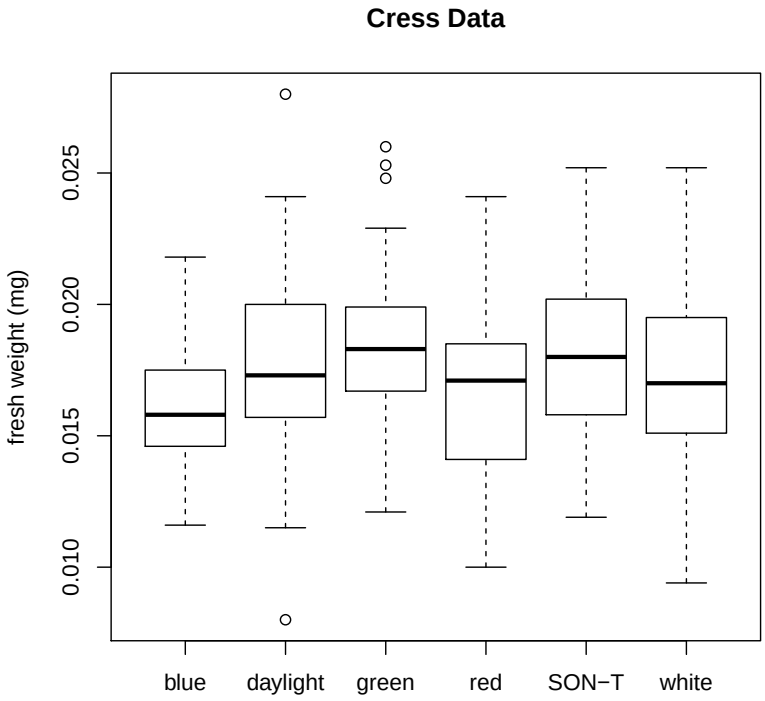


Figure 11.3: Boxplots of cress data for the determination of normal distribution and homogeneity in variances.

✔ **Data is independent**, completely randomized experiment.

The two-sided MCP according to Tukey contains the following hypotheses:

H_0: $\mu_{red} = \mu_{blue} = \mu_{green} = \mu_{white} = \mu_{daylight} = \mu_{SON-T}$

H_1: $\mu_{red} \neq \mu_{blue}$
$\mu_{red} \neq \mu_{green}$
$\mu_{red} \neq \mu_{white}$
$\mu_{red} \neq \mu_{daylight}$
$\mu_{red} \neq \mu_{SON-T}$
$\mu_{blue} \neq \mu_{green}$
$\mu_{blue} \neq \mu_{white}$
$\mu_{blue} \neq \mu_{daylight}$
$\mu_{blue} \neq \mu_{SON-T}$
$\mu_{green} \neq \mu_{white}$
$\mu_{green} \neq \mu_{daylight}$
$\mu_{green} \neq \mu_{SON-T}$
$\mu_{white} \neq \mu_{daylight}$
$\mu_{white} \neq \mu_{SON-T}$
$\mu_{daylight} \neq \mu_{SON-T}$

```
> library(mvtnorm)
> library(multcomp)
> cress.model <- aov(formula = weight~light, data = cress)
> cress.test <- glht(cress.model, linfct = mcp(light = "Tukey"))
> summary(cress.test)
```

```
         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = weight ~ light, data = cress)

Linear Hypotheses:
                       Estimate Std. Error t value Pr(>|t|)
daylight - blue == 0   0.0013156  0.0006828   1.927   0.3883
green - blue == 0      0.0022022  0.0006828   3.225   0.0177 *
red - blue == 0        0.0006711  0.0006828   0.983   0.9231
SON-T - blue == 0      0.0020067  0.0006828   2.939   0.0413 *
white - blue == 0      0.0011356  0.0006828   1.663   0.5575
green - daylight == 0  0.0008867  0.0006828   1.299   0.7857
red - daylight == 0   -0.0006444  0.0006828  -0.944   0.9347
SON-T - daylight == 0  0.0006911  0.0006828   1.012   0.9136
white - daylight == 0 -0.0001800  0.0006828  -0.264   0.9998
red - green == 0      -0.0015311  0.0006828  -2.242   0.2221
SON-T - green == 0    -0.0001956  0.0006828  -0.286   0.9997
white - green == 0    -0.0010667  0.0006828  -1.562   0.6242
SON-T - red == 0       0.0013356  0.0006828   1.956   0.3706
white - red == 0       0.0004644  0.0006828   0.680   0.9840
white - SON-T == 0    -0.0008711  0.0006828  -1.276   0.7980
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

According to the multiplicity adjusted p-value, there exists a significant difference between green and blue as well as between blue and SON-T light treatment in the dry weight with an error probability of 5%.

11.2.5.1 Further Investigations

Previous research concluded that blue light affects the stem elongation negatively in comparison to e.g. red light. Therefore, a blue light treatment results usually in a more compact plant growth and a slightly reduced fresh weight for certain species. Can this be applied on cress? A one-sided seceding test according to Dunnett (control `blue`):

```
> cress.test <- glht(cress.model, linfct = mcp(light="Dunnett"),
+ alternative = "greater")
> summary(cress.test)

         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts


Fit: aov(formula = weight ~ light, data = cress)

Linear Hypotheses:
                        Estimate Std. Error t value  Pr(>t)
daylight - blue <= 0 0.0013156  0.0006828   1.927 0.09934 .
green - blue <= 0    0.0022022  0.0006828   3.225 0.00321 **
```

```
red - blue <= 0      0.0006711  0.0006828   0.983 0.42229
SON-T - blue <= 0    0.0020067  0.0006828   2.939 0.00778 **
white - blue <= 0    0.0011356  0.0006828   1.663 0.16238
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

With an error probability of 5%, plants treated with green and SON-T light have a significantly higher fresh weight than cress plants treated with blue light. This is only partly congruent with previous research results about the effect of light quality on stem elongation.

11.2.6 Example Fertilizer

11.2.6.1 Experiment

Twelve plots were randomly divided in three groups. The first two groups were treated with the fertilizer A and B while the third group was kept as an untreated control (table 11.1) (Wonnacott and Wonnacott, 1990, p. 334).

Fertilizer A	Fertilizer B	Control C
75	74	60
70	78	64
66	72	65
69	68	55

Table 11.1: Yield dependent on different fertilizers.

11.2.6.2 Analysis

```
> fertilizer <- read.table(file = "../text/fertilizer.txt", sep = "\t",
+ header = TRUE)
```

Normal distribution and homogeneity of variances are obtained from boxplot figure 11.4 and a Levene test:

```
> boxplot(formula = yield~fertilizer, data = fertilizer, col = "white",
+ main = "Fertilizer Data")
> library(car)
> leveneTest(y = fertilizer$yield, group = fertilizer$fertilizer)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  2  0.1765 0.8411
       9
```

- ✔ Data is approximately **normal distributed** (figure 11.4).
- ✔ **Homogeneity of variances** is concluded from a non significant Levene test and the boxplots.
- ✔ **Independent data:** randomized block design.

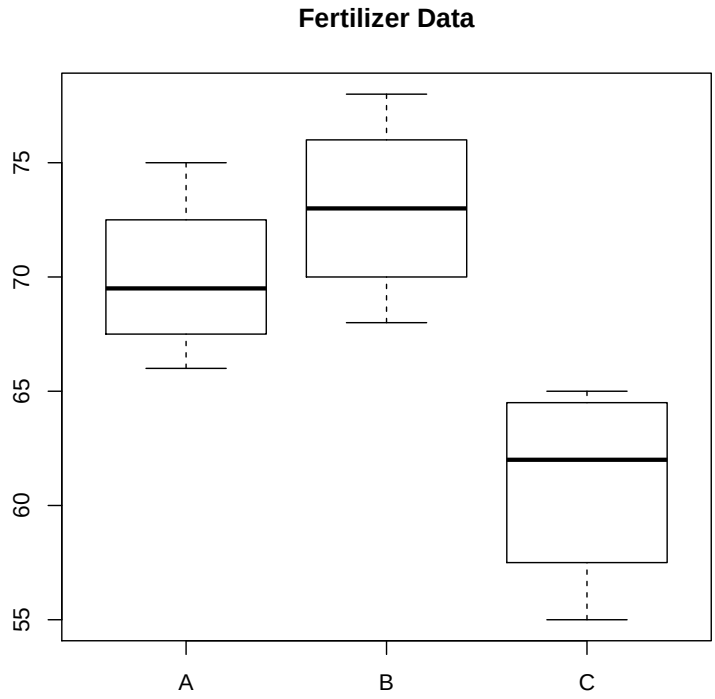


Figure 11.4: Boxplots for fertilizer data.

The question in this experiment is whether the two fertilizers differ significantly from the control. Therefore, a one-sided acceding test with the following hypotheses is calculated:

H_0 : $\mu_C \geq \mu_A$
$\mu_C \geq \mu_B$

H_1 : $\mu_C < \mu_A$
$\mu_C < \mu_B$

```
> fert.model <- aov(formula = yield~fertilizer, data  = fertilizer)
> fert.test <- glht(fert.model, linfct = mcp(fertilizer = "Dunnett"),
+ alternative = "greater")
> summary(fert.test)

         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts


Fit: aov(formula = yield ~ fertilizer, data = fertilizer)

Linear Hypotheses:
           Estimate Std. Error t value Pr(>t)
B - A <= 0    3.000      2.944   1.019  0.265
C - A <= 0   -9.000      2.944  -3.057  0.999
(Adjusted p values reported -- single-step method)
```

With an error probability of 5%, both fertilizers increase the yield highly significant.

11.2.7 Example Melons (2)

The registration of new varieties is based on the fact that the new variety is better than already existing varieties in at least on criterion.

Using the data of section 11.2.4, I assume that A is a new variety that has to be compared to the already existing varieties B, C and D. A one-sided acceding MCP with `Dunnett` procedure is calculated. The implementation equals the description in section 11.2.4 except that `type` is set on `Dunnett`.

```
> melon.model <- aov(formula = yield~variety, data = melon)
> mcmp <- glht(melon.model, linfct = mcp(variety = "Dunnett"),
+ alternative = "greater")
> summary(mcmp)

         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts


Fit: aov(formula = yield ~ variety, data = melon)

Linear Hypotheses:
           Estimate Std. Error t value  Pr(>t)
B - A <= 0  16.9133     2.4754   6.833 < 0.001 ***
C - A <= 0  -0.9983     2.4754  -0.403 0.87049
D - A <= 0   9.4067     2.4754   3.800 0.00147 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)

> plot(confint(mcmp), col = "purple")
```

The confidence intervals lead to the conclusion that variety A is significantly better in yield with an error probability of 5% than the varieties B and D. The new variety would probably not be accepted for registration because it is not significantly better than variety C.

11.2.8 Elementary Calculation of p-values According to Holm

The local p-values (`p raw`) are adjusted as follows:

Bonferroni: The raw p-value is multiplied with the number of comparisons.

Bonferroni-Holm: The raw p-values are sorted by increasing size. The first p-value is multiplied with the full number z of comparisons. If this p-value is significant, the next one is multiplied with z-1 et cetera. The procedure stops when a p-value has not been significant (Holm, 1979).

Table 11.2 shows the calculation exemplarily for the melon test problem.

11.2.8.1 Implementation in R

Adjusting p-values according to several methods is implemented in the function `p.adjust()`:

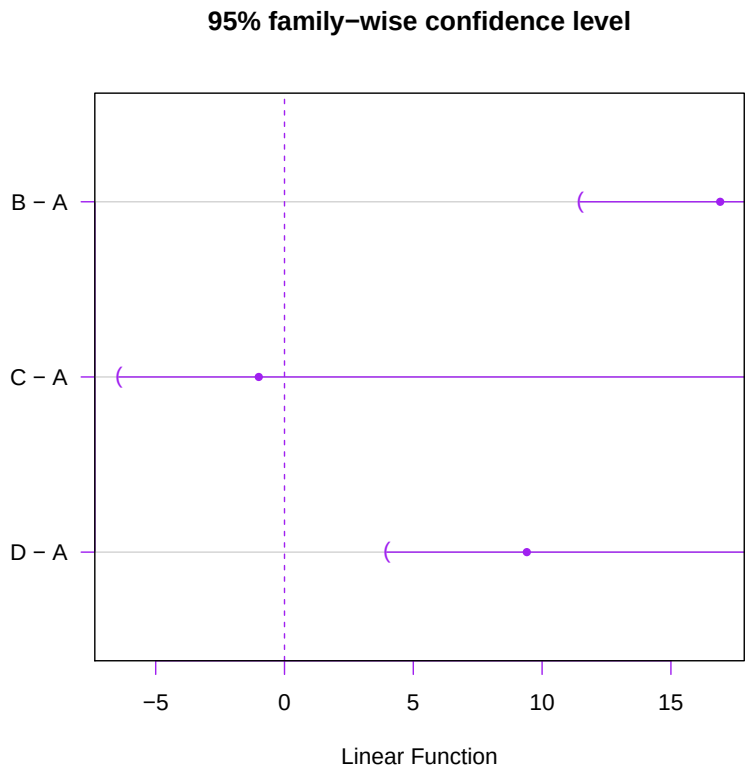


Figure 11.5: One-sided confidence intervals for a Dunnett test of the melon data.

Hypoth.	$\mathbf{p}_{raw}$	$\mathbf{p}_{Bonf}$	$\mathbf{p}_{Bonf-Holm}$	**Sign.**
B - A	0.000	0.000*3 = 0	0.000*3 = 0	yes/yes
D - A	0.001	0.001*3 = 0.003	0.001*2 = 0.002	yes/yes
C - A	0.654	0.654*3 $\Rightarrow$ 1	0.654*1 = 0.654, Stop	no/no

Table 11.2: Elementary p-value adjustment according to Bonferroni and Bonferroni-Holm.

```
p.adjust(raw.p.vector, method = "holm", "hochberg", "hommel",
         "bonferroni", "BH", "BY", "fdr", "none")
```

```
> raw.p.values <- c(0, 0.001, 0.654)
> bonf.p.values <- p.adjust(raw.p.values, method = "bonferroni")
> bonf.p.values

[1] 0.000 0.003 1.000

> holm.p.values <- p.adjust(raw.p.values, method = "holm")
> holm.p.values

[1] 0.000 0.002 0.654
```

✎ Exercise 12

The humidity in four soil types was measured for ten samples respectively (table 11.3) (Mead et al., 2003, p. 62).

Soil A	Soil B	Soil C	Soil D
12.8	8.1	9.8	16.4
13.4	10.3	10.6	8.2
11.2	4.2	9.1	15.1
11.6	7.8	4.3	10.4
9.4	5.6	11.2	7.8
10.3	8.1	11.6	9.2
14.1	12.7	8.3	12.6
11.9	6.8	8.9	11.0
10.5	6.9	9.2	8.0
10.4	6.4	6.4	9.8

Table 11.3: Humidity content of four different soil types.

Is this dataset suiting for the analysis with a MCP? Which procedure do you choose? Which are the hypotheses? Implement the test and plot the confidence intervals. Interpret the output!

Summary

This manual was written to help students of introductory biostatistics courses in understanding and using R as a tool for the evaluation of scientific experiments.

Amoung hundreds of functions in R, a couple of very helpful functions have been chosen and explained in detail. Real data sets keep up with the practical, horticultural basis. Parametric and non-paramentric two sample tests, correlation, linear regression, ANOVA and Multiple Comparison Tests have been discussed.

The R-Manual is prepared for extensions. All document sources are available and appendix C gives usage instructions.

Appendix A

Answers to Exercises

✍ Answer 1

1.
```
> a <- 12
> b <- 7
> result.2.binom <- (a-b)^2
> result.2.binom

[1] 25
```

2.
```
> zahlenkette <- (28:-34)
> zahlenkette

 [1]  28  27  26  25  24  23  22  21  20  19  18  17  16  15  14
[16]  13  12  11  10   9   8   7   6   5   4   3   2   1   0  -1
[31]  -2  -3  -4  -5  -6  -7  -8  -9 -10 -11 -12 -13 -14 -15 -16
[46] -17 -18 -19 -20 -21 -22 -23 -24 -25 -26 -27 -28 -29 -30 -31
[61] -32 -33 -34
```

3.
```
> ?objects()
```

   On Windows, the help window is closed as usual. On Linux, you have to type q to return to the command line.

```
> objects()

[1] "a"              "b"              "result.2.binom"
[4] "zahlenkette"

> rm(object = a)
```

```
> sunflowers <- data.frame(solution = rep(c("complete","l.Mg","l.N","l.mn"),
+ each = 3), dry.weight = c(1172,750,784,67,95,59,148,234,92,297,243,263))
> sunflowers

   solution dry.weight
1  complete       1172
2  complete        750
3  complete        784
4      l.Mg         67
5      l.Mg         95
```

```
6      l.Mg         59
7       l.N        148
8       l.N        234
9       l.N         92
10     l.mn        297
11     l.mn        243
12     l.mn        263
```

✍ Answer 2

```
> salad <- data.frame(weight = c(3.06, 2.78, 2.87, 3.52, 3.81, 3.60, 3.3,
+ 2.77, 3.62, 1.31, 1.17, 1.72, 1.20, 1.55, 1.53), group = c(rep(c("bowl"),
+ times = 9), rep(c("bibb"), times = 6)))
> tapply(X = salad$weight, INDEX = salad$group, FUN = mean)

    bibb     bowl
1.413333 3.258889

> tapply(X = salad$weight, INDEX = salad$group, FUN = sd)

     bibb      bowl
0.2198788 0.3999201

> tapply(X = salad$weight, INDEX = salad$group, FUN = median)

bibb bowl
1.42 3.30

> tapply(X = salad$weight, INDEX = salad$group, FUN = var)

      bibb       bowl
0.04834667 0.15993611

> tapply(X = salad$weight, INDEX = salad$group, FUN = min)

bibb bowl
1.17 2.77

> tapply(X = salad$weight, INDEX = salad$group, FUN = max)

bibb bowl
1.72 3.81

> tapply(X = salad$weight, INDEX = salad$group, FUN = quantile)

$bibb
    0%    25%    50%    75%   100%
1.1700 1.2275 1.4200 1.5450 1.7200

$bowl
  0%  25%  50%  75% 100%
2.77 2.87 3.30 3.60 3.81
```

```
> tapply(X = salad$weight, INDEX = salad$group, FUN = sum)

 bibb  bowl
 8.48 29.33

> tapply(X = salad$weight, INDEX = salad$group, FUN = IQR)

  bibb   bowl
0.3175 0.7300
```

✍ Answer 3

Boxplot figure A.1:

```
> boxplot(formula = weight~group, data = salad, col = "white",
+ main = "Dry Weight of Lettuce Varieties", ylab = "dry weight (g)")
```

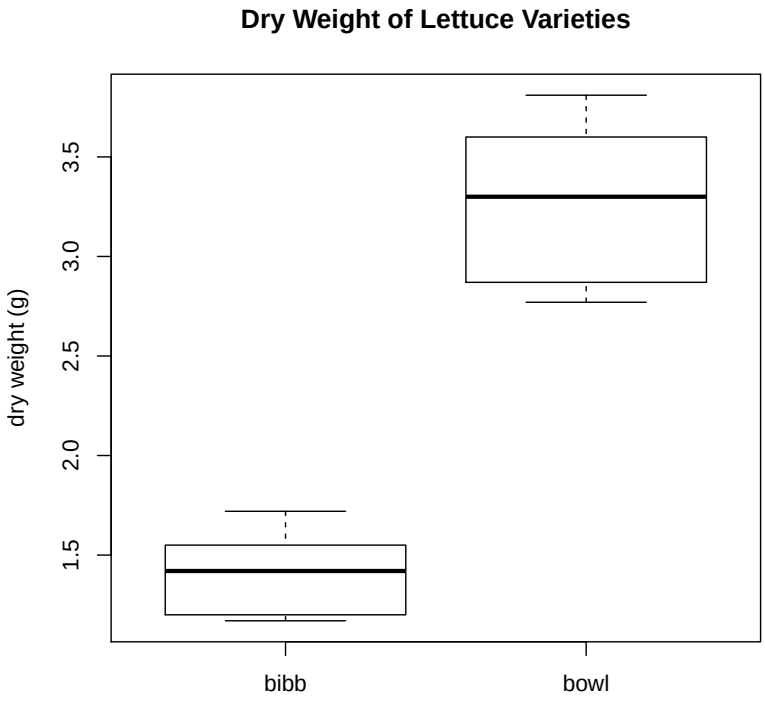


Figure A.1: Boxplots showing the leaf dry weight of two lettuce varieties.

✍ Answer 4

The additive effect on the plants is unknown. Two-sided hypotheses:

$$H_0 : \mu_{standard} = \mu_{additiv}$$
$$H_1 : \mu_{standard} \neq \mu_{additiv}$$

```
> strawberry<- read.table(file = "../text/strawberry.txt", sep = "\t",
+ header = TRUE)
```

Boxplots for the determination of normal distribution (figure A.2):

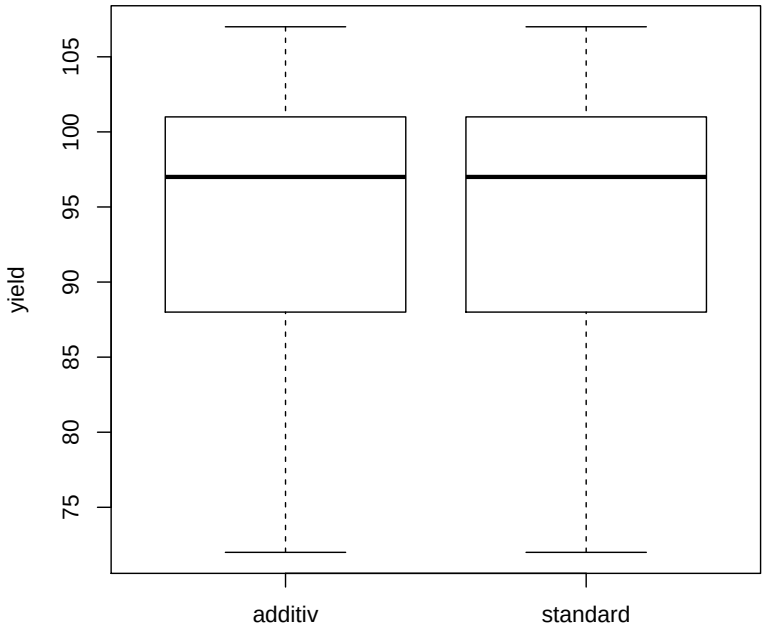


Figure A.2: Boxplots for the yield of strawberries with and without an additive fighting small white worms.

```
> boxplot(formula = yield ~ treatment, data = strawberry, col = "white",
+ ylab = "yield")
```

Normal distribution is assumed.

F-test to verify homogeneity in variances:

```
> var.test(formula = yield ~ treatment, data = strawberry)

        F test to compare two variances

data:  yield by treatment
F = 1, num df = 4, denom df = 4, p-value = 1
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1041175 9.6045299
sample estimates:
ratio of variances
                 1
```

Variances to not differ significantly.

Data is suiting for an analysis with a classical t-test.

```
> t.test(formula = yield ~ treatment, data = strawberry,
+ alternative = "two.sided", var.equal = TRUE)


        Two Sample t-test

data:  yield by treatment
t = 0, df = 8, p-value = 1
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -19.86379  19.86379
sample estimates:
 mean in group additiv mean in group standard
                    93                     93
```

To a confidence level of 95%, there is no significant difference. The very high p-value might rather be used as an indicator for equality which is a success for this experiment (looking for no effect on the strawberry plants).

✍ Answer 5

Two-sided test (because no tendency is expected):

$$H_0 : \mu_{bowl} \neg \mu_{bibb}$$
$$H_1 : \mu_{bowl} \neq \mu_{bibb}$$

```
> read.table(file = "../text/lettuce.txt", sep = "\t", header = TRUE)

   variety weight
1     bowl   3.06
2     bowl   2.78
3     bowl   2.87
4     bowl   3.52
5     bowl   3.81
6     bowl   3.60
7     bowl   3.30
8     bowl   2.77
9     bowl   3.62
10    bibb   1.31
11    bibb   1.17
12    bibb   1.72
13    bibb   1.20
14    bibb   1.55
15    bibb   1.53
```

Figure A.3 shows the boxplots:

```
> boxplot(formula = weight~variety, data = lettuce, col = "white")
```

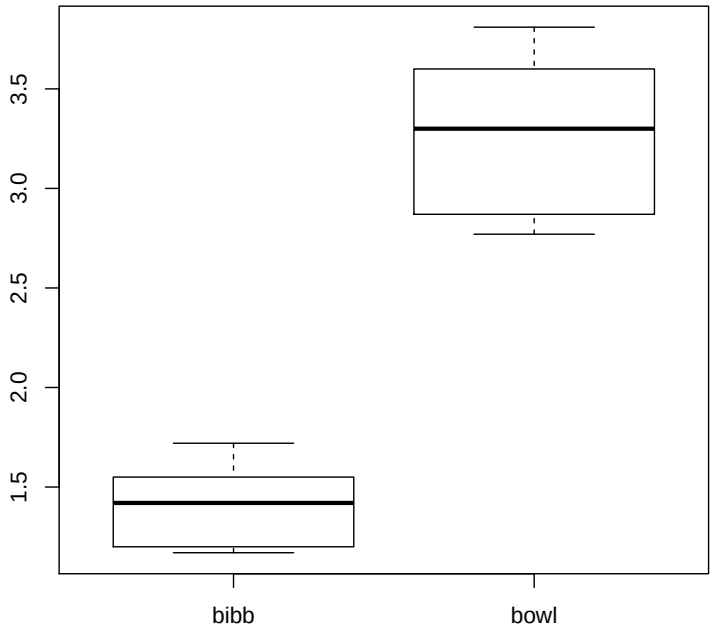


Figure A.3: Boxplots for the dry weight of two lettuce varieties' leaves.

- ✔ Approximate normal distribution (figure A.3).
- ✔ Boxes are different in length (figure A.3), therefore heterogeneity in variances is concluded.
- ✔ Independent data.

$\Longrightarrow$ t-Welch test.

```
> t.test(formula = weight~variety, data = lettuce)

        Welch Two Sample t-test

data:  weight by variety
t = -11.484, df = 12.716, p-value = 4.422e-08
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.193542 -1.497569
sample estimates:
mean in group bibb mean in group bowl
          1.413333           3.258889
```

The lettuce varieties differ significantly in dry weight with an error probability of 5%. According to the confidence interval, leaves of the variety Bibb are at least 1.4 up to 2.1 g lighter than leaves of the variety Bowl.

✍ Answer 6

Boxplots are shown in figure A.4:

```
> light <- read.table(file = "../text/light.txt", sep = "\t", header = TRUE)
> boxplot(formula = height~color, data = light, col = "white",
+ ylab = "height in inches", main ="Effect of Light on Soybeans")
> var.test(formula = height~color, data = light)

        F test to compare two variances

data:  height by color
F = 1.4026, num df = 24, denom df = 16, p-value = 0.4892
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.5342844 3.3809995
sample estimates:
ratio of variances
          1.402585
```

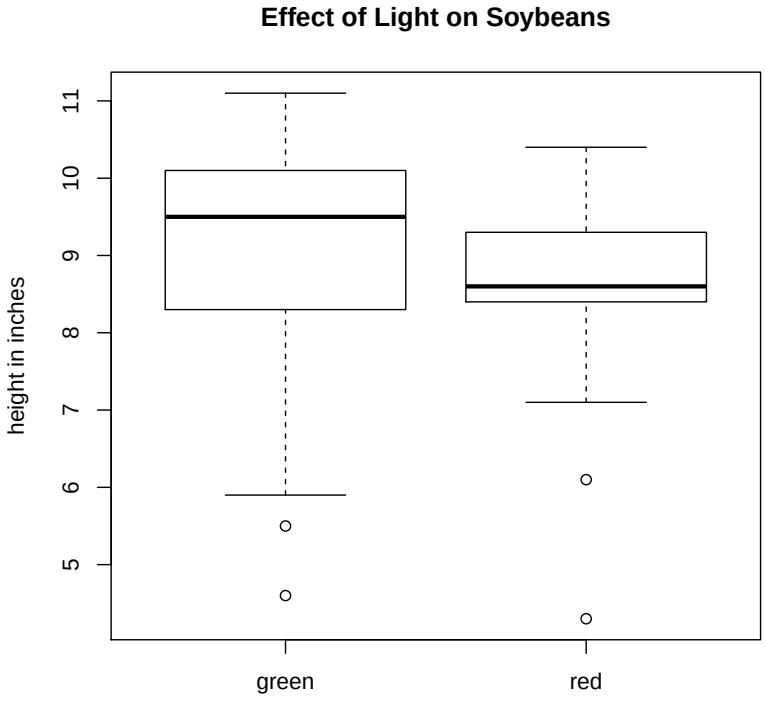


Figure A.4: Boxplots showing the effect of two different light colors on the growth of soybean plants.

- ✔ Unknown distribution (data is not normal distributed due to outliers and an asymmetric median in figure A.4).
- ✔ Homogeneity of variances is critical. F-test does not detect a heterogeneity.

✔ Continuous data (height measured in inches).

$\Longrightarrow$ Two-sided Wilcoxon Rank Sum Test, usage of `wilcox.exact()` due to ties.

$$H_0 : F_{red}(y) = F_{green}(y)$$
$$H_1 : F_{red}(y) \neq F_{green}(y)$$

```
> library(exactRankTests)
> wilcox.exact(formula = height~color, data = light,
+ alternative = "two.sided", correct = FALSE, exact = TRUE)

        Exact Wilcoxon rank sum test

data:  height by color
W = 272, p-value = 0.1296
alternative hypothesis: true mu is not equal to 0
```

Soybean plants treated with red light differ significantly in height from plants treated with green light with an error probability of 10%.

✍ Answer 7

χ^2 Goodness-of-Fit Test according to Pearson (number of observations greater than 20).

```
> flax <- c(15,26,15,0,8,8)
> genetic.model <- c(3,6,3,1,2,1)/16
> chisq.test(x = flax, p = genetic.model)

        Chi-squared test for given probabilities

data:  flax
X-squared = 7.7037, df = 5, p-value = 0.1733
```

The observed distribution does not differ significantly from the expected distribution to a confidence level of 90%. H_0 is not rejected.

✍ Answer 8

χ^2 Homogeneity Test according to Pearson (number of observation greater than 20).

```
> biotope <- matrix(c(25,25,75,75), ncol = 2)
> chisq.test(biotope, correct = FALSE)

        Pearson's Chi-squared test

data:  biotope
X-squared = 0, df = 1, p-value = 1
```

H_0 is not rejected. No significant differences in the percentage distribution of species A dependent on species B could be detected. The species are therefore not associated.

✍ Answer 9

Scatterplot is shown in figure A.5, boxplots in figures A.6 and A.7:

```
> ascorbic.acid <- read.table(file = "../text/ascorbic.txt", sep = "\t",
+ header = TRUE)
> plot(acid~response, data = ascorbic.acid, col = "black",
+ xlab = "Ascorbic acid concentration (mug/cm^3)", ylab = "response",
+ main = "Photometric Data")
> boxplot(x = ascorbic.acid$acid, col = "green3",
+ ylab = "concentration (mug/cm^3)", main = "Boxplot of Acid Concentration")
> boxplot(ascorbic.acid$response, col = "green3", main = "Photometer Response")
```

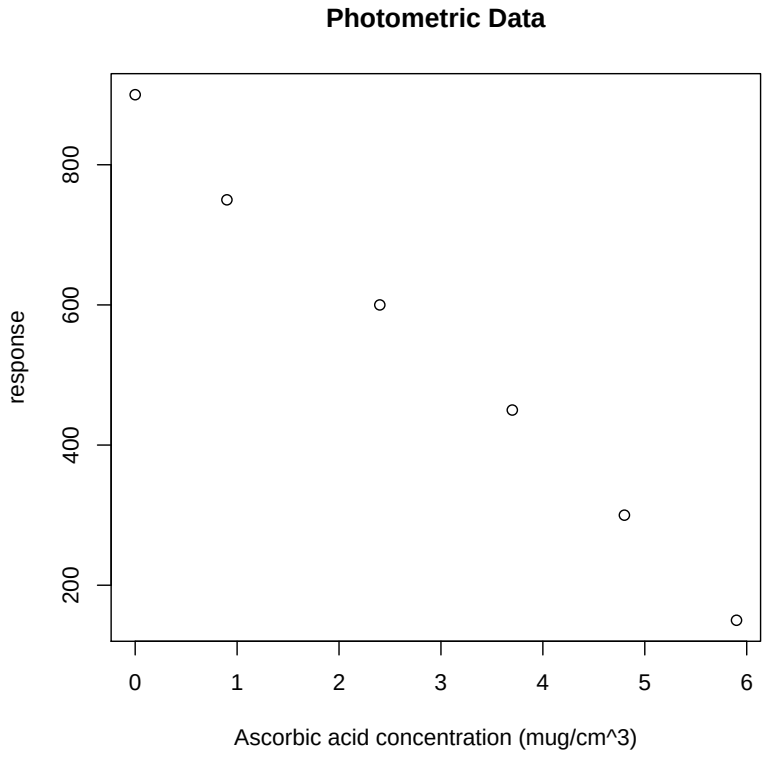


Figure A.5: Scatterplot of photometric data for the ascorbic acid content.

The scatterplot (figure A.5) takes us to the expectation of a negative correlation coefficient. Figures A.6 and A.7 show the normal distribution of both variables. Therefore, a correlation according to Pearson with a one-sided seceding test is calculated.

```
> cor.test(formula = ~acid+response, data = ascorbic.acid, method = "pearson",
+ alternative = "less")

        Pearson's product-moment correlation

data:  acid and response
t = -33.599, df = 4, p-value = 2.34e-06
alternative hypothesis: true correlation is less than 0
```

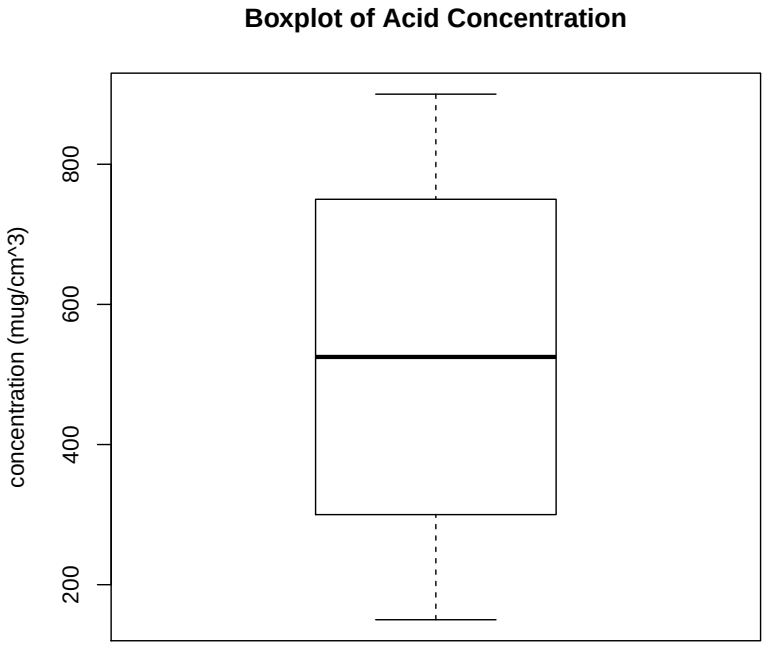


Figure A.6: Boxplot of ascorbic acid concentration for determination of the normal distribution.

```
95 percent confidence interval:
 -1.0000000 -0.9882535
sample estimates:
       cor
-0.9982331
```

The coefficient of -0.998233 shows an almost perfect correlation. The small p-value verifies the significance to a confidence level of 95%.

✍ Answer 10

The Scatterplot is shown in figure A.8:

```
> sulphur <- read.table(file = "../text/sulphur.txt", sep = "\t",
+ header = TRUE)
> plot(scab~concentration, data = sulphur, col = "black",
+ xlab = "sulphur (pounds/acre)", ylab = "percentage scab damage",
+ main = "Scab Treatment with Sulphur")
> scabmodel <- lm(formula = scab~concentration, data = sulphur)
> abline(reg = scabmodel, col = "black")
```

The following functions visualize the residuals in figure A.9. They are accepted as normal distributed and homogeneous in variances:

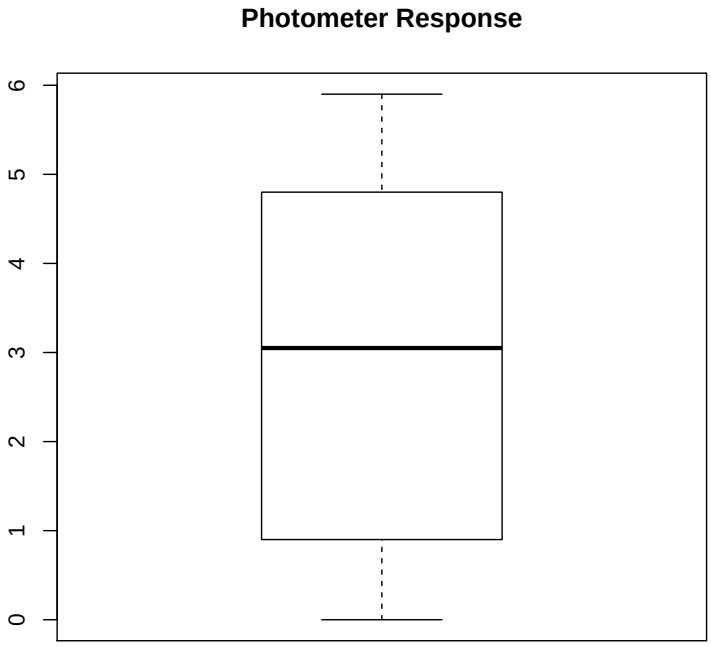


Figure A.7: Boxplot of photometric response data for determination of normal distribution.

```
> fitted.values <- fitted(object = scabmodel)
> resid.values <- resid(object = scabmodel)
> plot(x = fitted.values, y = resid.values, col = "black")
> abline(h = 0, col = "black")
```

- ✔ The **Number of predictor levels** is greater than two.
- ✔ The **Number of observations** over all x-values is greater than three.
- ✔ **Homogeneity of variances** of residuals (figure A.9)
- ✔ **Normal distribution** of residuals (figure A.9)

$\Longrightarrow$ Data is fitting for a regression with scabmodel.

```
> summary(object = scabmodel)

Call:
lm(formula = scab ~ concentration, data = sulphur)

Residuals:
     Min       1Q   Median       3Q      Max
-12.9571  -2.9286  -0.5429   4.9000   9.1000

Coefficients:
```

Figure A.8: Scatterplot of potato scab data.

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   20.90000    2.44048   8.564 6.14e-07 ***
concentration -0.01314    0.00355  -3.702  0.00237 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.301 on 14 degrees of freedom
Multiple R-squared:  0.4946,        Adjusted R-squared:  0.4586
F-statistic:  13.7 on 1 and 14 DF,  p-value: 0.002369
```

The equation for the straight line is:

$$y = 20.9 - 0.01314x$$

The intercept as well as the slope are highly significant.

The following commands plot confidence and prediction bands (figure A.10):

```
> pp <- predict(object = scabmodel, interval = "prediction",
+ data = sulphur$concentration)
> pc <- predict(object = scabmodel, interval = "confidence",
+ data = sulphur$concentration)
> plot(x = sulphur$concentration, y = sulphur$scab,
+ ylim = range(sulphur$scab, pc), col = "black",
+ xlab = "application (pounds/acre)", ylab = "percentage scab damage",
+ main = "Confidence and Prediction Bands")
```

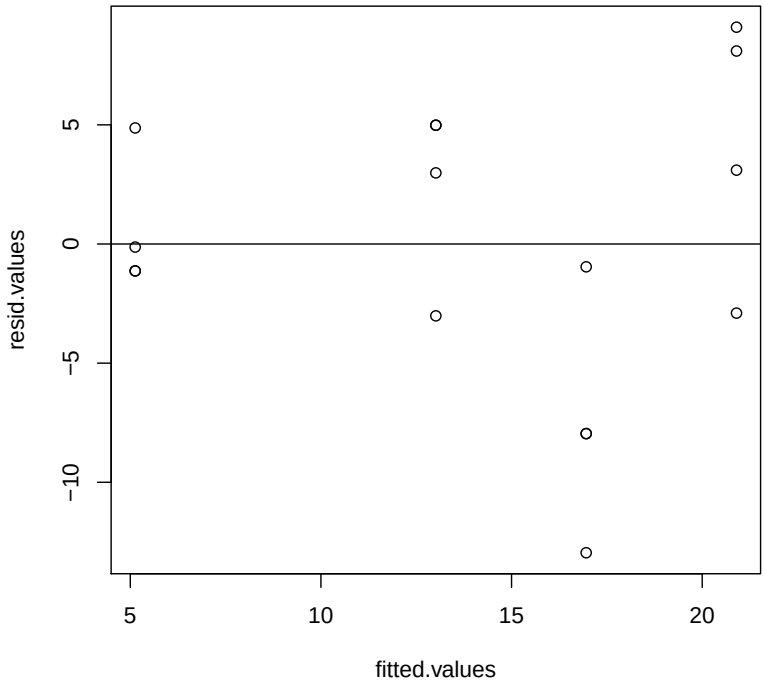


Figure A.9: Residual plot of potato scab data.

```
> matlines(x = sulphur$concentration, pp, tly = c(1,3), col = "magenta3")
> matlines(x = sulphur$concentration, pc, tly = c(1,2,3), col = "steelblue")
```

✍ Answer 11

```
> cherry <- read.table(file = "../text/cherry.txt", sep = "\t", header = TRUE)
> cherry.model <- lm(formula = response~treatment, data = cherry)
```

Residual plot (figure A.11):

```
> fitted.values <- fitted(object = cherry.model)
> resid.values <- resid(object = cherry.model)
> plot(x = fitted.values, y = resid.values, col = "black")
> abline(h = 0, col = "black")
```

Levene test for homogeneity of variances between the groups:

```
> library(car)
> lev <- data.frame(res = resid.values, group = cherry$treatment)
> attach(lev)
> leveneTest(y = res, group = group)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value    Pr(>F)
```

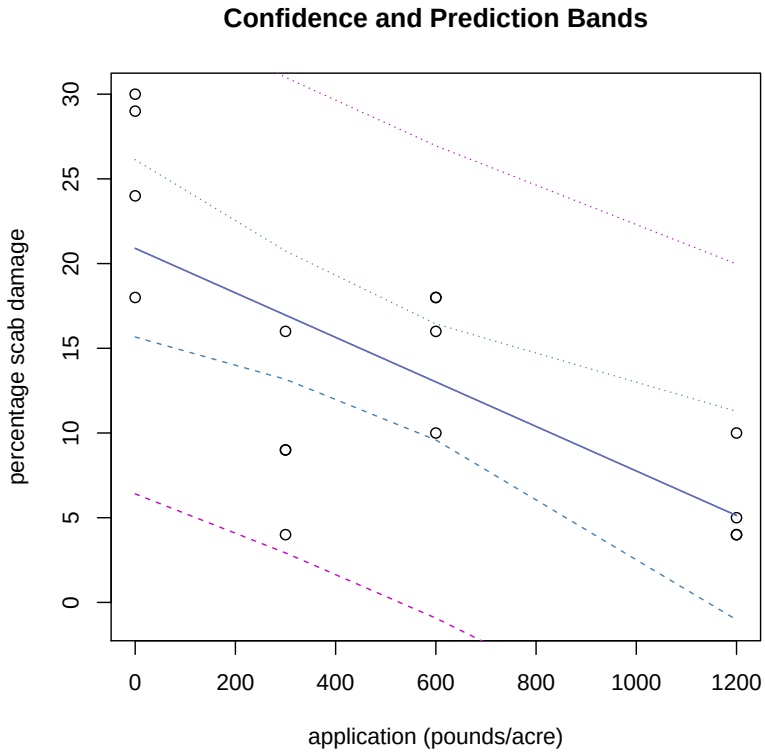


Figure A.10: Confidence and prediction bands for scabmodel.

```
group  3  14.912 0.0002376 ***
      12
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> detach(lev)
```

- ✔ **Homogeneity of variances** is accepted due to the levene test result.
- ✔ **Normal distribution** of residuals is critical, I assume robustness.
- ✔ **Independent data**.

$\Longrightarrow$ ANOVA. Hypotheses:

H_0: $\mu_{control} = \mu_{top}\mu_{bottom}\mu_{both}$
H_1: $\exists$ at least one $\mu_{location} \neq \mu_{location'}$

```
> anova(object = cherry.model)

Analysis of Variance Table

Response: response
          Df  Sum Sq Mean Sq F value    Pr(>F)
treatment  3 16278.2  5426.1  53.801 3.124e-07 ***
Residuals 12  1210.3   100.9
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

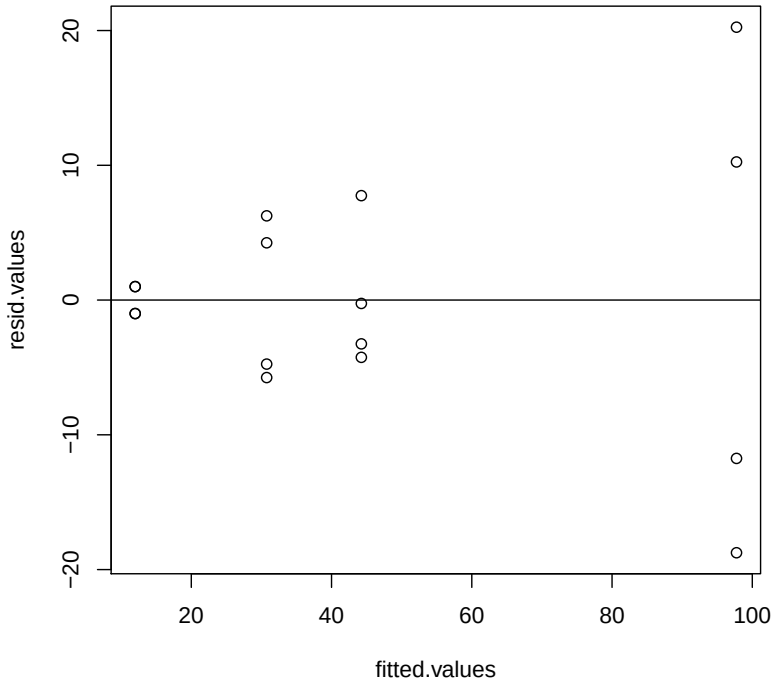


Figure A.11: Residual plot of cherry.model.

There exists at least one significant difference in transpiration for the different treatments (to a confidence level of 95%).

✍ Answer 12

The boxplots are shown in figure A.12:

```
> soil <- read.table(file = "../text/soil.txt", sep = "\t", header = TRUE)
> boxplot(formula = moisture~treatment, data = soil, col = "white",
+ ylab = "moisture", main = "Soil Moisture in Different Plots")


> library(car)
> leveneTest(y = soil$moisture, group = soil$treatment)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  3  0.8003 0.5019
      36
```

- ✔ Approximate **normal distribution** (figure A.12) is assumed (although there are outliers).
- ✔ **Homogeneity in variances** is accepted due to the levene test result.

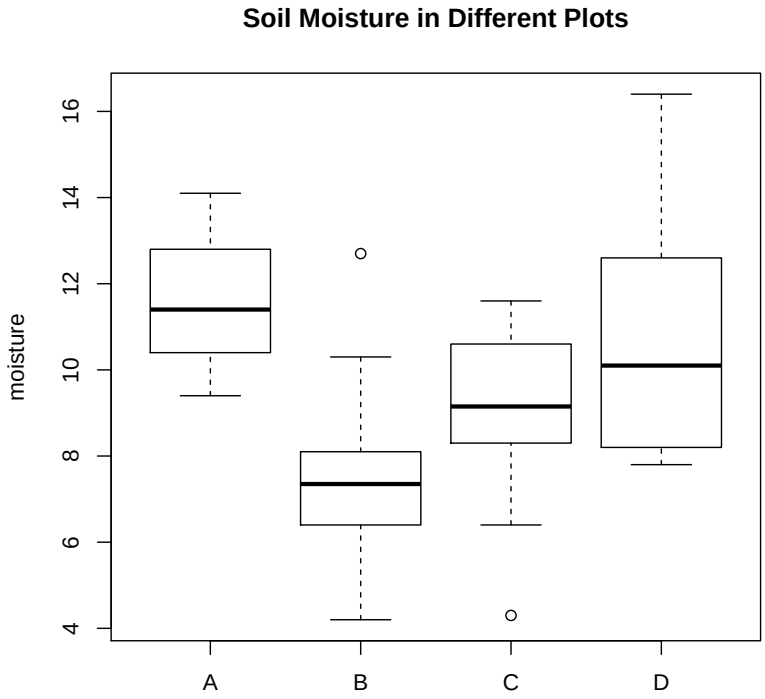


Figure A.12: Boxplots for humidity in different soil types.

$\Longrightarrow$ Multiple Comparison Test with Tukey procedure (no control was nominated). Two-sided hypotheses:

H_0: $\mu_A = \mu_B = \mu_C = \mu_D$

H_1: $\mu_A \neq \mu_B$
$\mu_A \neq \mu_C$
$\mu_A \neq \mu_D$
$\mu_B \neq \mu_C$
$\mu_B \neq \mu_D$
$\mu_C \neq \mu_D$

The confidence intervals are plot in figure A.13:

```
> library(mvtnorm)
> library(multcomp)
> soil.model <- aov(formula = moisture~treatment, data = soil)
> soil.test <- glht(soil.model, linfct = mcp(treatment = "Tukey"))
> summary(soil.test)

         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = moisture ~ treatment, data = soil)
```

```
Linear Hypotheses:
           Estimate Std. Error t value Pr(>|t|)
B - A == 0   -3.870      1.046  -3.701  0.00389 **
C - A == 0   -2.620      1.046  -2.506  0.07587 .
D - A == 0   -0.710      1.046  -0.679  0.90438
C - B == 0    1.250      1.046   1.195  0.63365
D - B == 0    3.160      1.046   3.022  0.02272 *
D - C == 0    1.910      1.046   1.827  0.27793
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)

> plot(confint(soil.test), col = "purple")
```

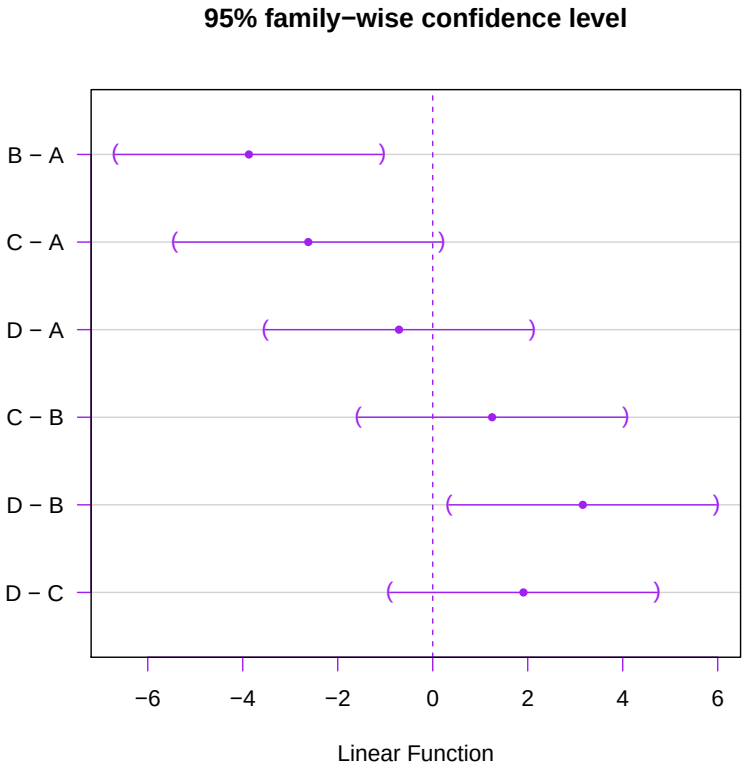


Figure A.13: Confidence intervals about humidity in different soil types.

With an error probability of 5%, the soil types A and B as well as D and B differ significantly in humidity.

Figure A.13 shows that the humidity in soil D is higher than in C and that the humidity in A is higher than in B.

Appendix B

Cress Data

Light	Block	Height	Light	Block	Height	Light	Block	Height
red	A	2,1	SON-T	A	2,5	blue	A	2,15
red	A	2,2	SON-T	A	2,6	blue	A	2,4
red	A	2,7	SON-T	A	2,5	blue	A	2,15
red	A	1,85	SON-T	A	2,6	blue	A	2
red	A	2,4	SON-T	A	2,25	blue	A	2,1
red	A	2,2	SON-T	A	2,6	blue	A	2,1
red	A	2,55	SON-T	A	2,75	blue	A	2,35
red	A	2,55	SON-T	A	2,7	blue	A	1,95
red	A	2,6	SON-T	A	2,5	blue	A	2,2
red	A	3,05	SON-T	A	3,2	blue	A	2,4
red	A	2,45	SON-T	A	2,1	blue	A	2,1
red	A	2,75	SON-T	A	3,15	blue	A	2,2
red	A	2,55	SON-T	A	2,55	blue	A	2,1
red	A	2,65	SON-T	A	2,75	blue	A	2,45
red	A	2,6	SON-T	A	1,85	blue	A	2
red	B	2,2	SON-T	B	2,95	blue	B	1,95
red	B	2,5	SON-T	B	3,4	blue	B	2
red	B	2,4	SON-T	B	2,35	blue	B	2,3
red	B	2,95	SON-T	B	3,1	blue	B	1,8
red	B	2,8	SON-T	B	3,25	blue	B	2,5
red	B	3,2	SON-T	B	2,9	blue	B	2,4
red	B	2,25	SON-T	B	2,6	blue	B	2,1
red	B	2,7	SON-T	B	2,45	blue	B	2,15
red	B	2,4	SON-T	B	2,95	blue	B	2,3
red	B	2,35	SON-T	B	3,05	blue	B	2,5
red	B	2,6	SON-T	B	3,5	blue	B	2,1
red	B	2,8	SON-T	B	3,4	blue	B	2,3
red	B	2,1	SON-T	B	2,7	blue	B	2,35
red	B	2,75	SON-T	B	2,9	blue	B	2,3
red	B	2,3	SON-T	B	2,5	blue	B	1,95
red	C	2,5	SON-T	C	2,4	blue	C	2,05
red	C	2,9	SON-T	C	2,6	blue	C	2,35
red	C	2,8	SON-T	C	3,15	blue	C	2,1

red	C	2,5	SON-T	C	2,6	blue	C	2,1
red	C	2,7	SON-T	C	2,7	blue	C	1,75
red	C	3,05	SON-T	C	2,8	blue	C	1,95
red	C	2,5	SON-T	C	2,7	blue	C	2,35
red	C	2	SON-T	C	3,35	blue	C	2,2
red	C	2,7	SON-T	C	2,4	blue	C	2,6
red	C	2,7	SON-T	C	2,8	blue	C	1,65
red	C	2,8	SON-T	C	2,85	blue	C	1,75
red	C	2,6	SON-T	C	2,5	blue	C	2,2
red	C	2,9	SON-T	C	2,9	blue	C	2,1
red	C	3,1	SON-T	C	2,7	blue	C	1,9
red	C	2,5	SON-T	C	2,8	blue	C	2,25
daylight	A	2,5	white	A	2,5	green	A	2,55
daylight	A	2,4	white	A	2,8	green	A	2,35
daylight	A	2,3	white	A	2,2	green	A	2,8
daylight	A	2,15	white	A	3	green	A	2,55
daylight	A	1,6	white	A	2,7	green	A	2,9
daylight	A	2,35	white	A	2,7	green	A	2,4
daylight	A	1,95	white	A	2,75	green	A	2,3
daylight	A	2,5	white	A	2,7	green	A	2,75
daylight	A	2,7	white	A	2,35	green	A	3,1
daylight	A	2,75	white	A	2,8	green	A	2,9
daylight	A	2,6	white	A	2,5	green	A	2,8
daylight	A	2,8	white	A	2,8	green	A	2,75
daylight	A	2,4	white	A	3,4	green	A	2,5
daylight	A	2,15	white	A	3,3	green	A	2,4
daylight	A	2,25	white	A	2,55	green	A	3
daylight	B	2,4	white	B	2,65	green	B	2,5
daylight	B	2,55	white	B	2,2	green	B	2,7
daylight	B	2,3	white	B	2,7	green	B	3,2
daylight	B	2,9	white	B	2,2	green	B	2,9
daylight	B	2,75	white	B	2,5	green	B	2,6
daylight	B	2,85	white	B	2,5	green	B	3,4
daylight	B	2,3	white	B	1,5	green	B	3
daylight	B	2,85	white	B	2,25	green	B	2,2
daylight	B	2,2	white	B	2,15	green	B	2,5
daylight	B	2,45	white	B	2,5	green	B	2,15
daylight	B	2,2	white	B	1,85	green	B	2,8
daylight	B	2,55	white	B	3	green	B	2,8
daylight	B	2,3	white	B	2,2	green	B	2,8
daylight	B	2,3	white	B	2,1	green	B	2,75
daylight	B	2,2	white	B	2,7	green	B	3,2
daylight	C	2,55	white	C	3	green	C	2,95
daylight	C	2,45	white	C	2,95	green	C	2,9
daylight	C	2,35	white	C	2,25	green	C	2,6
daylight	C	2,35	white	C	2,95	green	C	3,2
daylight	C	2,7	white	C	2,7	green	C	2,8
daylight	C	2,6	white	C	2,7	green	C	2,8

daylight	C	2,1	white	C	2,5	green	C	2,9
daylight	C	2,15	white	C	2,3	green	C	2,95
daylight	C	2,55	white	C	2,2	green	C	2,75
daylight	C	2,45	white	C	2,8	green	C	2,75
daylight	C	2,5	white	C	3	green	C	2,8
daylight	C	2,65	white	C	2,5	green	C	2,4
daylight	C	2,65	white	C	2,9	green	C	2,6
daylight	C	2,65	white	C	1,8	green	C	2,6
daylight	C	2,2	white	C	2,45	green	C	3,2

Appendix C

Editing the R-Manual

If you are planning to elaborate on this R-Manual, you should get familiar with the usage of R and LaTeX first.

This document has been generated by Sweave (R 2.1.1) and pdfLaTeX. It is written in the unicode-format (utf8). This means you cannot transfer it to Windows easily, except you find a unicode supporting editor. I strongly recommend you to elaborate on this document on Linux or another Unix-System. Although there is a tool called **GNU recode** (Free Software Foundation Inc., 1998) which is able to transpose utf8 to Latin-1, you will still have to change all path references inside the different collaborating documents on Windows - and probably some of the LaTeX libraries, too.

The source of the R-Manual for Biometry is provided in a folder called **BSc**.

C.1 Structure

The folder **BSc** has five subdirectories: **Bilder**, which contains all pictures that are not automatically generated (Screenshots ect.), **excel**, which contains all data sets as ∗.xls-files, **Snw_files**, which contains the source of the document, **text**, which contains all data sets as a ∗.txt-files and **windows**, which contains an R source code file for windows and the data-set cress.txt (not automatically generated).

Additional obligatory files in the BSc directory are: *RHandbuch.tex*, *RManual_English.tex*, *boxplot.jpg*, *danksagung.tex*, *danksagunge.tex*, *bibnames.sty*, *plotbeetmodel.jpg*, *cress.tex*, *khoff.bib*, *titlebar.jpg*, *whitebox.jpg* and *Sweave_Linux_Howtoe.tex*.

RHandbuch.tex and *RManual_English.tex* are the LaTeX master documents that will be used for pdf-LaTeX Compilation of the R-Manual for Biometry in German and English. The output files are named *RHandbuch.pdf* and *RManual_English.pdf*. They will be found in the same directory by default. This file does not necessarily need to be changed very much, except you want to use different LaTeX libraries or change the self defined commands. It might be of help for you to have a look at the commented self-defined commands in this document to elaborate on the Snw.files.

khoff.bib contains the references for the R-Manual. Include your additional references in this file to use them with the command `citep` in the LaTeX environment.

The file *boxplot.jpg* is a default boxplot picture used in chapter five (t-Test), *titlebar.jpg* and the other jpgs are pictures (in the wrong directory) or tools used for formatting certain parts of the documents manually. *danksagung.tex* contains my personal thanks to people who helped developing this manual. *Sweave_Linux_Howto.tex* contains this appendix on how to elaborate on the document.

Don't change any other files that might occur in the BSc directory during Sweave or LaTeX compilation!

C.2 Working Environment

You need to have the texmf LaTeX environment for Linux including pdfLaTeX and ucs to be installed.

I used the KDE LaTeX editor Kile for editing the source files. You can basically use any other Linux editor. Kile is convenient regarding the user friendly buttons for LaTeX compilation and the management of several documents opened at the same time.

In addition, you will need to have R running on your computer.

C.3 Where to Start?

If you decide that you would like to include a new chapter, the first step is to create a $*$.Snw-file in the BSc/Snw_files/ directory. Those source files are named by numbers (kap1.Snw, kap2.Snw...) but you can choose another name if you like to. The English file version gehts the identical name except that an "e" is added on the end.

Note: This newly created file is NOT a LaTeX file. You do not need to include the begin and end document tags or anything else. It is the R source for a LaTeX chapter of the R-manual.

Enter a LaTeX chapter tag and start writing your document as if it was a LaTeX file.

C.4 A Short Summary on Sweave

Whenever there occurs a R-source code part you would like to include in your chapter, use the Sweave tags.

The most simple tag that will provide the entered source code and the result in the document and does not display pictures is:

```
<<>>=
R code
@
```

The option `echo = FALSE` provides a nice tool if you want to enter source code NOT displayed in the document (hidden chunks):

```
<<echo = FALSE>>=
R code
@
```

For displayed figures, you should set the argument `fig = TRUE`. You can also combine this with the `echo = FALSE` argument if you ONLY want the figure to be displayed:

```
<<fig = TRUE, echo = FALSE>>=
R code
@
```

Please have a look at the *Sweave User Manual* (Leisch, 2005) for further information on the usage of Sweave.

C.5 How to Proceed

When you think that you have finished your chapter including all the R-tags you save it and open R. Set the correct directory by hand the first time:

```
setwd("/wherever/you/keep/BSc")
```

In R, you call the ∗.Snw chapter with the command:

```
Sweave("Snw_files/yourfile.Snw")
```

Elaborate on your R source code if you get any error messages.

If the source code is correct, R will create a ∗.tex-file in the **BSc** directory, named the same as your ∗.Snw-file. It will also create all figures you included in your source code with `fig = TRUE`.

The next step is then to open the file *RHandbuch.tex* / *RManual_English.tex* and edit a line at the bottom (before end document):

```
\include{yourfile}
```

Make sure that you compile all other Snw-files one time before you run pdf-LaTeX on RHandbuch.tex the first time on your computer. (The *.tex-files and figures need to be created one time.)

C.6 How to Treat LaTeX Errors

If you get any LaTeX errors while compiling with pdf-LaTeX, go back to your ∗.Snw-file and do the corrections. Compile the ∗.Snw again with Sweave and THEN run pdf-LaTeX!

I hope you are able to work on the R-Manual with those comments.

Acknowledgements

I would like to thank Prof. Dr. A. L. Hothorn and Universitetslektor Jan-Eric Englund for supporting and supervising my bachelor thesis.

A special thanks goes to Dr. Frank Bretz who announced the topic and spend a lot of time on supervision in the beginning phase of my work.

I thank Cornelia Froemke, Alexandra Hoff, Xuefei Mi and Barbara Zinck for patient proofreading.

Without Brian Fynn who lent me a spare notebook when my own computer was damaged, I would not have been able to work on my thesis for quite a while. Thank you very much, Brian.

Grateful acknowledgements for discussion, support and motivation are also made to Prof. Dr. Klaus Hoff, Linus Masumbuko, Prof. Dr. Jan Petersen and Richard Zinck.

Bibliography

Ahern, T. (1998). *Statistical analysis of EIN plants treated with ancymidol and $H_2 0$.* Oberlin College. Unpublished manuscript.

Baur, E., Fischer, E., and Lenz, F. (1931). *Human Heredity, 3rd edition.* Macmillan, New York.

Bishop, O. N. (1980). *Statistics for biology - A practical guide fo the experimental biologist, 3rd edition.* Longman, Longman House, Burnt Mill, Harlow, Essex.

Cochran, W. G. and Cox, G. M. (1950). *Experimental designs.* John Wiley & Sons, Ltd, New York, Second Edition 1957.

Collins, C. and Seeney, F. (1999). *Statistical Experiment Design and Interpretation - An Introduction with Agricultural Examples.* John Wiley & Sons, Ltd, Baffins Lane, Chichester, West Sussex PO19 1UD, England.

Dalgaard, P. (2002). *Introductory Statistics with R.* Springer Verlag.

Fierer, N. (1994). *Statistical analysis of soil respiration rates in a light gap and surrounding old-growth forest.* Oberlin College. Unpublished manuscript.

Free Software Foundation Inc. (1998). *GNU recode.* 59 Temple Place - Suite 330, Boston, MA 02111, USA. http://www.gnu.org/software/recode/recode.html, 19. Juli 2005.

Froemke, C. (2004). *Einführung in die Biometrie für Gartenbauer.* Lehrgebiet für Bioinformatik, Universität Hannover. Unveröffentlichtes Übungsskript.

Gent, A. (1999). Oberlin College. Unpublished data collected at Oberlin College.

Gentleman, R. (2005). Reproducible research: A bioinformatics case study. *bepress (http://www.bepress.com/sagmb)*, 4, Issue 1.

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., and Ostrowski, E. (1994). *A Handbook of Small Data Sets.* Chapman & Hall, Great Britain.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70.

Knight, S. L. and Mitchell, C. A. (2000). Enhancement of lettuce yield by manipulation of light and nitrogen nutrition. *Journal of the American Society for Horticultural Science*, 108:750 – 754.

Leisch, F. (2005). *Sweave User Manual.* http://www.ci.tuwien.ac.at/~leisch/Sweave/, 11. Juni 2005.

Martinez, J. (1998). *Organic practices for the cultivation of sweet corn.* Oberlin College. Unpublished manuscript.

Mead, R., Curnow, R. N., and Hasted, A. M. (2003). *Statistical Methods in Agriculture and Experimental Biology.* Chapman & Hall/CRC, CRC Press LLC, 2000 N. W. Corporate Blvd., Boca Raton, Florida 33431.

Neumann, A., Richards, A.-L., and Randa, J. (2001). *Effects of acid rain on alfalfa plants*. Oberlin College. Unpublished manuscript.

Norlinger, C. and Hoff, K. J. (2004). *The effect of light quality on garden cress*. Swedish University of Agricultural Sciences. Unpublished project report.

Pappas, T. and Mitchell, C. A. (1984). Effects of seismic stress on the vegetative growth of glycine max (l.) merr. cv. wells ii. *Plant, Cell and Environment*, 8:143 – 148.

Pearce, S. C. (1983). *The Agricultural Field Experiment*. John Wiley & Sons, Ltd, Chicester, New York, Brisbane, Toronto, Singapore.

R Development Core Team (2004a). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3.

R Development Core Team (2004b). *R Installation and Administration (Version 2.0.1., 2004-11-15)*. 17.01.2004 http://www.r-project.org.

Saedi, G. and Rowland, G. G. (1997). The inheritance of variegated seed color and palmitic acid in flax. *Journal of Heredity*, 88:466 – 468.

Samuels, M. L. and Witmer, J. A. (2003). *Statistics for the Life Sciences, 3rd edition*. Pearson Education, Inc., Upper Saddle River, New Jersey 07458.

Stallman, R. (1991). *GNU General Public License, 2nd edition 1991*. 59 Temple Place, Suite 330, Boston, USA.

Wonnacott, T. H. and Wonnacott, R. J. (1990). *Introductory Statistics*. John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore. 5th edition.