Statistics for Business

With applications in R and RStudio

Koen Derks
Maurice Hoenderdos
Jacques de Swart
Ruud Wetzels

Version 1.0 (April 30, 2020)

This workbook was originally created for the lectures of the bachelor course *Statistics for Business* at Nyenrode Business University (Breukelen, the Netherlands).

The authors want to give credit to the following people or affiliations for their (open-source) contributions:

Hand icon created by Darius Dan from www.graphicburger.com
Computer icon created by Darius Dan from www.graphicburger.com
Hint icon created by Flat Icons from www.flaticon.com

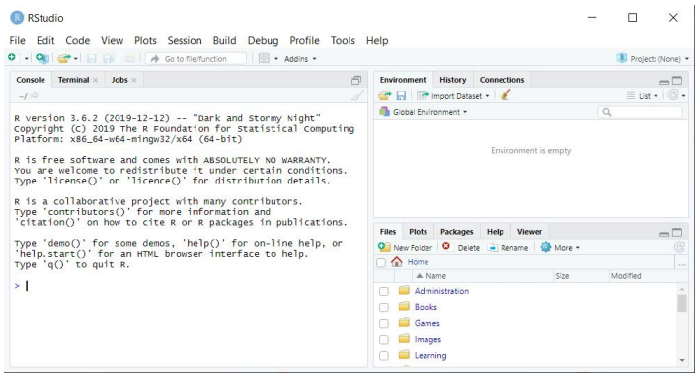
Installing and setting up R and RStudio

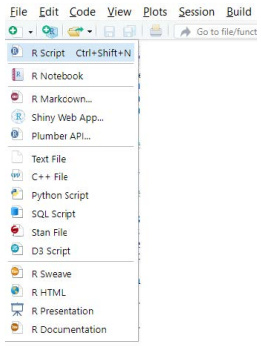
Before you can start to learn statistics using this workbook, it is required that you have installed both R and RStudio on your computer. R itself is just a bare-bones programming language, and can be hard to learn and navigate without guidance. RStudio is a shell around the R language which offers some guidance and visualizes most of the functionality that R generally hides for users.

R is open-source and can be downloaded free-of-charge from https://cran.r-project.org. While working through this workbook, you will not actually open this program at any time, but you are required to have it installed since it is the engine that RStudio runs on.

RStudio is a user-friendly coding environment that makes your life while working with R a little bit easier. RStudio can be freely downloaded from https://rstudio.com. This is the program that you will be using to write and run your own R code.

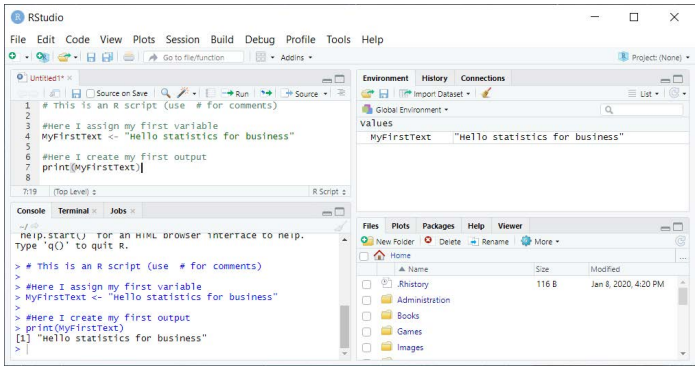
1. If you open RStudio for the first time you will encounter the following screen:





2. Press the "New" menu button and choose "R Script", or press Ctrl (⌘) + Shift + N to open a new R *script.*

3. You can now write and run your own code in the newly opened R *script. Select all code you want to execute and press Ctrl (⌘) + Enter or "Run" to run.*



Probability theory is nothing but common sense reduced to calculation.

- Pierre-Simon Laplace (1749 - 1827)

Content

Page

Introduction to the workbook

In this introduction some basic principles are defined that are followed throughout the workbook to guide you through discovering statistics using R. Here follows a summation of the different sections that you will encounter while working through the assignments of the various chapters.

Learning objectives

Every chapter has specific learning objectives that give you an initial impression about what you will learn in the assignments in that chapter. The learning objectives also provide a good indication of your ability after finishing the assignments in a chapter, since you can go back and check whether you master the learning objectives.

The learning objectives are shown at the beginning of each chapter in a `yellow` box.

Learning objectives of this chapter:

Exercises by hand

Fancy some mathematical exercise? When you see the icon to the right you know that you will be expected to perform calculations by hand (or by the use of a simple calculator). For example, you may get asked to calculate the mean of a small data set in an assignment.

Exercises in R

The icon to the right is shown when it is time to start programming in R. You may warm up your programming fingers, since you will be asked to write and run your own code in the assignments that follow.

Hints

When you see the icon to the right it means that you will be given a hint for that question. Hints can be useful if you do not know how to get started with an assignment, or if you are a little bit lost in the middle of one. Pay close attention to these hints, as they may also guide your attention to important aspects that can be easily overlooked.

Statistical concepts

Various important statistical concepts will present itself during the assignments. These statistical concepts are highlighted in **red** and can be found all over the text of the assignments. When these concepts can be calculated, their formulas can often be found in the formula sheet on page 111. An example of such an important statistical concept can be the mean.

mean

Data sets

The color **green** is used to indicate the name of a data set that can be found in the online resources that accompany this workbook. For example, you might be asked to read a data set into R for further use in an assignment.

example.csv

R code

R code in the assignments is displayed in **blue** . These short lines of code are generally used in the assignments to highlight objects or variables that exist in the current R session. They might also highlight certain functions that you can use on an R object.

```
View(dataset)
```

R code blocks

R code blocks are used throughout the assignments to indicate R code that should be run together. Copy this code into your script in RStudio and select all of the code to run at it at once. Be sure to do this at all times when you see a code block, as running some lines of code individually may influence your results drastically.

```
x <- c(1, 5, 8, 3)
mean(x)
```

Chapter 1: Descriptive statistics

Learning objectives of this chapter:

- Calculating the mean, mode, median, quartiles and range by hand
- Get started with R: set the working directory, inspect and load data
- Calculating the mean, mode, median and range in R

Assignment 1.1: Descriptive statistics by hand

Calculate by hand (or by the use of a pocket calculator) the descriptive statistics that are listed below.

- **mean**
- **mode**
- **median**
- **range**
- **lower and upper quartiles**
- **interquartile range**

1.1 a) 2 7 4 5 8 10 10 7 9 2 8 8 9 4 6

Answer 1.1a:

Mean: _____ Range: _____

Mode: _____ Lower quartile: _____

Median: _____ Upper quartile: _____

Interquartile range: _____

1.1 b) 7 7 6 5 2 1 3 7 5 9 9 10

Answer 1.1b:

Mean: _____ Range: _____

Mode: _____ Lower quartile: _____

Median: _____ Upper quartile: _____

Interquartile range: _____

1.1 c) Describe which question (1.1a or 1.1b) was more difficult to calculate and why.

Answer 1.1c:

1.1 d) Are these data sets **positively skewed** or **negatively skewed** ? Circle the correct answer and explain why you chose this answer using the relation between the **mean** , the **median** , and the **mode** .

Answer 1.1d:

These data sets are **positively** / **negatively** skewed.

Explanation:

Assignment 1.2: Descriptive statistics of small data sets in R

This assignment assumes you opened a new script in the code editor in RStudio. Write and run your own code to answer the following questions.

In R, a one-dimensional row of numbers is represented by a **vector** .

1.2 a) Use the **c()** function to enter the numbers from assignment 1.1a in a new vector called **dataset1** . Next, run the following code in R and explain what you see.

```
View(dataset1)
```

Hint 1.1: Check Part I of the R help on page 118 for more information on how to make a **vector** .

R code 1.2a:

Answer 1.2a:

1.2 b) Write and run your own code in R to find the **mean** , **mode** , **median** , and **range** for the vector **dataset1** . Compare your answers with those of assignment 1.1a.

Hint 1.2: Check part IV of the R help on page 124 for descriptive statistics functions.

Hint 1.3: There is no **mode** *function in R, but you can find the mode in a frequency* **table** .

R code 1.2b:

Answer 1.2b:

Mean: _____ Median: _____

Mode: _____ Range: _____

1.2 c) Run the following code in R and explain what you see.

```
quantile(dataset1, type = 6)
```

Answer 1.2c:

_____

_____

1.2 d) Use the `c()` function to create a vector `dataset2` with the data from assignment 1.1b and find the `mean`, `mode`, `median`, `range`, and `quartiles` for these data. Compare your answers with the answers of assignment 1.1b.

R code 1.2d:

Answer 1.2d:

Mean: _____ Range: _____

Mode: _____ Lower quartile: _____

Median: _____ Upper quartile: _____

Assignment 1.3: Descriptive statistics of larger data sets in R

For this assignment, you need the **`bloodPressure.csv`** data set that you can download from the online resources. This data set contains measurements of the age, blood pressure, cholesterol level, gender, and description of a random selection of people. It is normally used to look for relationships between these variables. Note that this is fake data and does not contain actual measurements.

Let's start with importing the data set (which is available in the online resources) into R.

1.3 a) Inspect and run the following code in R to import the blood pressure data and store it in the object **`dataset3`** . Explain how the code works and describe what the **`bloodPressure.csv`** file contains.

```
dataset3 <- read.csv(file.choose())
```

Answer 1.3a:

This method of importing data can be a lot of work if there are many files or if the script will be run many times. Faster methods exist though, for example by providing the full file path.

1.3 b) Describe and test ways this code can be improved to make importing a file easier.

Hint 1.4: Look at Part I of the R help at page 118 to find out more functions for importing data.

Answer 1.3b:

R code 1.3b:

The functions you used for descriptive statistics on the small data sets in assignment 1.1 can also be applied to the data set that is currently stored in **dataset3** .

1.3 c) Find the **mean** , **mode** , **median** , **range** , and **quartiles** for the column **Age** in **dataset3** . Describe this variable in running text using these statistics.

Hint 1.5: First find out how to extract (index) a specific column in a data frame using the **$** *sign.*

R code 1.3c:

Answer 1.3c:

For large data sets, it becomes a lot of work finding the **mode** in a frequency table each time. It is possible to import a package into the R session that contains a function for calculating the **mode** automatically. However, it is also possible to create a function that calculates the **mode** ourselves.

Run the following lines of R code together:

```
getMode <- function(x){
   uniqx <- unique(x)
   uniqx[which.max(tabulate(match(x, uniqx)))]
}
```

You have now created your first R function and you will see it displayed separately in the R environment. This function will give you the **mode** for any **numeric** vector or column. It works by first extracting all unique values, counting their frequency, and then selecting the value with the highest frequency. Note that you can use this function, but will not be required to understand or make functions like this. However, for the interested reader, part III of the R help contains more information on how to create your own functions.

1.3 d) Use the new **getMode()** function to determine the **mode** for column **Age** in **dataset3** and check if it is consistent with your answer for assignment 1.3c.

R code 1.3d:

Answer 1.3d:

Mode: _____

1.3 e) Find the **mean**, **mode**, **median**, **range**, and **quartiles** for the column **BloodPressure** in **dataset3**. Also use the new **getMode()** function.

R code 1.3e:

Answer 1.3e:

Mean: _____ Range: _____

Mode: _____ Lower quartile: _____

Median: _____ Upper quartile: _____

1.3 f) Is the distribution of the values in the **BloodPressure** column **positively skewed** or **negatively skewed**? Explain your answer using the relation between the **mean**, **median**, and **mode**.

Answer 1.3f:

These data sets are **positively** / **negatively** / **not** skewed.

Explanation:

_____

_____

_____

1.3 g) Determine the **variance** and **standard deviation** for the column **Cholestrol** in **dataset3** .

R code 1.3g:

Answer 1.3g:

Variance: _____

Standard deviation: _____

1.3 h) Validate the relation between the **variance** and the **standard deviation** by performing a calculation in R.

R code 1.3h:

Chapter 2: Creating graphs from data

Learning objectives of this chapter:

- Creating a histogram by hand and in R
- Creating and modifying a scatter plot in R
- Creating and modifying a line plot in R
- Creating a box plot in R

Assignment 2.1: Creating a histogram by hand

Suppose that you have the following set of numbers:

1.5 5.5 1.7 7.2 1.2 7.9 1.4 3.6 3.1 3.8 5.9 3.6 5.1 3.2 7.1

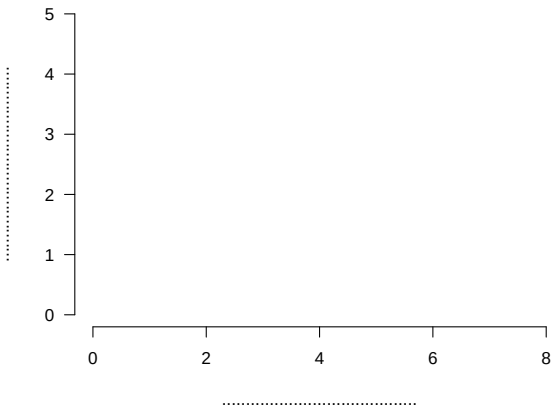
2.1 a) For each of the ranges given below, find out their **frequency** in the data above.

Answer 2.1a:

0 to 2	2 to 4	4 to 6	6 to 8
Frequency: ____	Frequency: ____	Frequency: ____	Frequency: ____

2.1 b) Draw the histogram for these data by hand. The name of the *x-axis* should represent the range of values, and the name of the *y-axis* should represent the **frequency** of the values that fall in the ranges of assignment 2.1a.

Answer 2.1b:



Assignment 2.2: Creating and modifying a histogram in R

2.2 a) Take the numbers from assignment 2.1a and use the `c()` function to enter these numbers in a vector called `dataset4`. Next, run the following code in R and explain what you see.

```
hist(dataset4)
```

Answer 2.2a:

2.2 b) The graph resulting from the command in 2.2a is not the same as the histogram that you drew in assignment 2.1b. What is the difference between the graph from 2.2a and your graph from 2.1b?

Answer 2.2b:

2.2 c) Try to replicate your exact histogram from 2.1b in R. This requires that you specify the `breaks` argument after a comma in the `hist()` function.

Hint 2.1: You can find more information on the `hist()` *function by running* `?hist`.

R code 2.2c:

2.2 d) Give your histogram a different color by adding and changing the `col` argument after the comma. Improve your histogram further by changing the *x-axis* name, the main title, and the rotation of the *y-axis* labels by adding more arguments.

Hint 2.2: Check Part II of the R help on page 121 for additional arguments to the `hist()` *function.*

R code 2.2d:

2.2 e) Calculate the **mean** and **median** of the values of **dataset4** . Is the distribution of **dataset4** **negatively skewed** or **positively skewed** ? Explain your answer using the relation between the **mean** and the **median** .

Answer 2.2e:

These data sets are **positively** / **negatively** / **not** skewed.

Explanation:

Assignment 2.3: Creating and modifying a scatter plot in R

In this assignment you are going to work with a data set that is built into R.

The **`data()`** function gives you access to all the data sets that are built into R, or that are included with downloaded R packages. For example, you can load a data set called **`swiss`** into the environment by running the R code below:

```
data(swiss)
```

The data are imported as an object called **`swiss`** , which you can now also see in the environment. This particular data set contains some socio-economic indicators for each of 47 French-speaking provinces of Switzerland. In this assignment, you will focus on the column **`Education`** (the percentage education beyond primary school) and the column **`Agriculture`** (the percentage of males involved in agriculture as an occupation).

2.3 a) Extract the values of the two columns using the **`$`** sign and store them in two new variables called **`education`** and **`agriculture`** .

Hint 2.3: You can check the R environment to see the names of the objects that you have currently stored.

R code 2.3a:

```
```

2.3 b) Use the **`plot()`** function to create a scatter plot of the two variables. Place the percentage of education beyond primary school on the *x-axis* and the percentage of males involved in agriculture as an occupation on the *y-axis*. Rename your axis labels to match the content of the graph.

R code 2.3b:

```
```

2.3 c) Looking at the scatter plot, what can you globally say about the relation between the percentage of education beyond primary school and the percentage of males involved in agriculture as an occupation in the French-speaking provinces of Switzerland?

Answer 2.3c:

2.3 d) Give the points in your scatter plot a different color by changing the `col` argument after the comma. Improve the graph further by changing the main title, the rotation of the *y-axis* labels, the shape of the points, and removing the outer lines.

R code 2.3d:

Assignment 2.4: Creating and modifying a line plot in R

In this assignment, you are going to create a line plot of the daily closing prices of some major European stock indices: Germany DAX, Switzerland SMI, France CAC, and UK FTSE.

You can load this data set by running the code below:

```
data(EuStockMarkets)
stockData <- data.frame(EuStockMarkets)
```

The data is now loaded into the environment as the `EuStockMarkets` data set and immediately transformed to the `stockData` data set (this is because of technical reasons as the original data is in a time-series format). In this assignment, you will work with the `stockData` data set.

2.4 a) Create a line plot where the closing price of the DAX stock is displayed over time. Give your plot appropriate *x-axis* and *y-axis* names.

R code 2.4a:

```

```

2.4 b) Find a function to add a separate line for the closing price of the SMI stock in red. You may try to add the other stocks as well using this function, but remember to adjust the *y-axis* accordingly so that all the lines are all displayed decently.

Hint 2.4: Do not use the `plot()` *function to add a line to an already existing plot.*

R code 2.4b:

```

```

Assignment 2.5: Creating a box plot in R

For this next assignment, let's return to the **swiss** data set. More specifically, you will now have to focus on the values that are stored in the separate variable **agriculture** .

2.5 a) Find out the **minimum** , **lower quartile** , **median** , **upper quartile** , and **maximum** of the **agriculture** variable.

R code 2.5a:

Answer 2.5a:

Minimum: _____ Upper quartile: _____

Median: _____ Lower quartile: _____

Maximum: _____

2.5 b) Create a box plot of the **agriculture** variable.

R code 2.5b:

2.5 c) Run the following code in R and explain the table output. How does the code work?

```
educationLevel <- rep('2.Medium', 47)
educationLevel[education <= 6] = '1.Low'
educationLevel[education >= 12] = '3.High'
table(educationLevel)
```

Answer 2.5c:

2.5 d) Create a box plot using the R code below and explain what you see.

```
boxplot(agriculture ~ educationLevel)
```

Answer 2.5d:

Chapter 3: Confidence intervals and hypothesis testing

Learning objectives of this chapter:

- Confidence interval and hypothesis testing by hand
- Testing the homogeneity of variance by hand
- Calculating confidence intervals for the mean and testing for normality
- Inspect and test the normality of a sample in R
- Using R to inspect and test the homogeneity of variance

Assignment 3.1: Confidence interval and hypothesis testing by hand

A call center with 38 employees handles thousands of calls a day. The company wants to get more insights into the duration of calls and the workload of the employees. On a day with a total of 4513 calls, the company decides to randomly pick 100 calls and measure how long they take. The **mean** call duration for this sample is 145 seconds with a **standard deviation** of 25 seconds. The call center wants to use this information to estimate the **mean** call duration for that day.

3.1 a) Write down the relevant information from this case. Use the symbols N, n, $\bar{x}$, s, σ, and μ.

Hint 3.1: Not all symbols are known and some should be left empty.

Answer 3.1a:

N: _____ s: _____

n: _____ σ: _____

$\bar{x}$: _____ μ: _____

3.1 b) What is the best estimate of the **mean** call duration (μ) based on this sample?

Answer 3.1b:

$\mu =$ _____

Explanation:

_____

3.1 c) Calculate the **standard error** of the mean (SE_μ) based on this sample.

Hint 3.2: Check the formula sheet on page 111 to find out how to calculate SE_μ.

Answer 3.1c:

$SE_\mu =$ _____

Calculation:

_____

3.1 d) Calculate the 99% **confidence interval** for the mean call duration.

*Hint 3.3: You can find the formula for the **lower bound**, **upper bound** and **z-value** in the formula sheet on page 111.*

Answer 3.1d:

z-value: _____

Calculation: _____

Lower bound: _____

Calculation: _____

Upper bound: _____

Calculation: _____

The manager of the call center does not really care about the **lower bound** and he does not require such high confidence. He just wants to make sure the average workload per employee is not too high, because otherwise he is forced by employment laws to hire more people. He calculated that if the **mean** call duration is below 150 seconds the workload is acceptable, and wants to use this sample to show with 95% confidence that the **mean** call duration is below 150 seconds.

3.1 e) Write down the hypotheses H_0 and H_1 for the manager's test. What is the value of μ_0? Also describe μ_0 for this case.

Hint 3.4: This is a one-sided test. Consider this when you formulate the hypotheses.

Answer 3.1e:

μ_0: _____

H_0: _____ H_1: _____

3.1 f) Do you need the **lower bound** or the **upper bound** of the **confidence interval** for this test? Calculate this bound.

Answer 3.1f:

`........` `bound:` _____

`Calculation:` _____

3.1 g) Draw the conclusion for the manager. Include the following four elements:

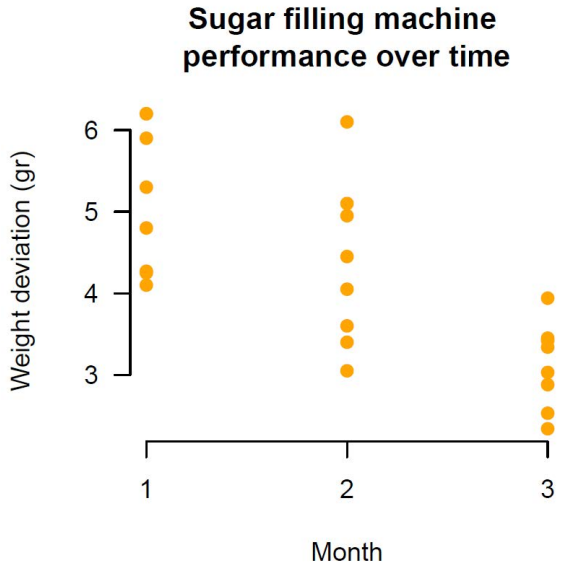
- ☐ Show how the **confidence interval** relates to μ_0.
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about μ and μ_0.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 3.1g:

Assignment 3.2: Testing for homogeneity of variance by hand

A manufacturer of machines for food processing has created a machine that fills bags with sugar. A machine like this is never perfect and when a 1-kilo bag of sugar is filled there are always small deviations. The actual content of a bag is often a few grams more or less than a kilo. The manufacturer knows these deviations are bigger for new machines but become smaller as the machine is longer used due to better calibration and the smoothing out of the moving parts. To be able to show this to his clients they take a random sample of eight bags of sugar on the first day of the three months the machine is in use and plots the (absolute) weight deviations.

The manufacturer wants to use the chart below for a model that shows how much the machine improves over time, but this requires **homogeneity of variances**. The **variance** in month 1 is 0.801 gr, the **variance** in month 2 is 1.113 gr. The **variance** for month 3 has not been calculated yet.



i	x_i	$(x_i - \bar{x})$	$(x_i - \bar{x})^2$
1	3.03		
2	3.45		
3	3.94		
4	2.34		
5	3.34		
6	2.53		
7	2.88		
8	3.42		

$\sum x_i =$ ..... $\sum (x_i - \bar{x})^2 =$ .....

$\bar{x} =$ ..... $s^2 =$ .....

3.2 a) Use the table to the right of the figure to calculate the **variance** s^2 for month 3 by filling in the calculations beneath the table.

Hint 3.5: you can find the formula for the **variance** *in the formula sheet on page 111.*

Answer 3.2a:

Variance: _____

3.2 b) Calculate **Hartley's F** (the variance ratio) for the sugar machine data set.

Hint 3.6: you can find the formula for **Hartley's F** *in the formula sheet on page 111.*

Answer 3.2b:

Hartley's F: _____

3.2 c) Formulate the **null hypothesis** H_0 and **alternative hypothesis** H_1 to test homogeneity of variance for this case.

Answer 3.2c:

H_0: _____ H_1: _____

3.2 d) Determine the **critical value** for Hartley's F ($F_{\max}$) for the sugar machine data set.

Hint 3.7: Check Table 1 on page 113 for the critical values of ***Hartley's F*** .

Answer 3.2d:

Hartley's $F_{\max}$: _____

3.2 e) Draw the conclusion on the homogeneity of variance for the sugar machine data set. Include the following elements:

- ☐ Show how the calculated **Hartley's F** relates to the critical value F_{max}.
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about the homogeneity of the three variances.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 3.2e:

Assignment 3.3: Calculating a confidence interval and testing for normality

For this assignment, you will need the **`populations.csv`** data file, which contains four different **`populations`** of 10,000 observations called **`P1 - P4`** .

3.3 a) Use the **`read.csv()`** function (and **`setwd()`** function if you prefer) to import the data set into an object named **`dataset5`** .

R code 3.3a:

3.3 b) Run the following code in R and explain what it does. Why do you have to use a seed?

```
set.seed(54321) # You can replace 54321 with your own seed number

sample1 <- sample(dataset5$P1, size = 90)
sample2 <- sample(dataset5$P2, size = 90)
sample3 <- sample(dataset5$P3, size = 90)
sample4 <- sample(dataset5$P4, size = 90)
```

Hint 3.8: Use R's help function (the **`?`** *symbol) to find out more information about the* **`sample()`** *and* **`set.seed()`** *functions.*

Answer 3.3b:

3.3 c) Calculate the **`mean`** and **`standard deviation`** for each **`sample`** and save them into variables **`x1`** - **`x4`** and **`s1`** - **`s4`** . Use these results to also calculate the **`standard errors`** and same them into variables **`se1`** - **`se4`** .

R code 3.3c:

Answer 3.3c:

```
Standard error sample 1:  _____________

Standard error sample 2:  _____________

Standard error sample 3:  _____________

Standard error sample 4:  _____________
```

The R function **qnorm(p, mean, sd)** returns the **z-value** for **quantile p** from a **normal distribution** with a certain **mean** μ (**mean**) and **standard deviation** σ (**sd**). If you do not specify a mean or standard deviation for the function it will assume the standard normal distribution $N(\mu = 0, \sigma = 1)$.

3.3 d) Run the following code in R and explain the value that you see.

```
qnorm(p = 0.95)
```

Answer 3.3d:

The **qnorm()** function cannot return the **z-value** for the two-sided **confidence interval** , but you can work around that by realizing that in a two-sided interval you have to split the risk over both sides of the standard normal distribution.

3.3 e) Run the following code in R and explain why you can use the value in **z** for a two-sided 95% **confidence interval** .

```
z <- qnorm(p = 0.975)
```

Answer 3.3e:

3.3 f) Calculate the 95% **confidence interval** for each sample. Store the lower bounds **lb1** - **lb4** and store the upper bounds in **ub1** - **ub4** .

R code 3.3f:

This is a rare case in which you actually have the full **populations**, so you can check if the estimates based on your **samples** are actually close to the real value in the populations.

3.3 g) Calculate the actual **population means**, call them **mu1** - **mu4**. Next, fill the schema below with all known values for **ub**, **x**, **lb**, and **mu**.

R code 3.3g:

Answer 3.3g:

	sample1	sample2	sample3	sample4
ub				
x				
lb				

	P1	P2	P3	P4
mu				
mu in interval?	YES / NO	YES / NO	YES / NO	YES / NO

3.3 h) Are all **population means** inside the interval you calculated?

Answer 3.3h:

YES / NO

3.3 i) How often do you expect this to happen given the confidence level (95%) used?

Answer 3.3i:

Assignment 3.4: Inspecting and testing the normality of a sample in R

In this assignment you are going to work with an R package called **car** . Packages are extensions for R created by its community. They contain functions that are not available in the basic R version and can be really useful. There are many packages on the internet, so only use the one you really need and always make sure you use packages from a reliable source (like the official **CRAN** servers). For clarity, it will be indicated what functions come from what packages in the chapters. If there is no package mentioned, functions come from base R.

A package has to be installed once. Run the following code in R to install the car package:

```
install.packages('car')
```

A package needs to be loaded in **every script** that uses functions from this package.

```
library(car)
```

This assignment assumes you have done assignment 3.3 and therefore have four $n = 90$ samples called **sample1** - **sample4** from populations **P1** - **P4** from **dataset5** and calculated the **confidence intervals** for the population means.

3.4 a) Create a histogram for each sample in **sample1** - **sample4** .

R code 3.4a:

```

```

3.4 b) Which samples look like they could have been taken from a **normal distribution** ?

Answer 3.4b:

3.4 c) Use this R code (from the **car** package) to create four **qq-plots** for the four **samples**.

```
qqPlot(sample1, distribution = 'norm') # qqPlot for sample 1
```

R code 3.4c:

3.4 d) Evaluate each **qq-plot** and explain the deviations from the diagonal by referencing features of the histograms. Do the **qq-plots** confirm your answer for question 3.4b?

Answer 3.4d:

3.4 e) Formulate the **null hypothesis** H_0 and **alternative hypothesis** H_1 for a test of normality in these samples.

Answer 3.4e:

H_0: _____ H_1: _____

3.4 f) Use the following R code to perform a **Shapiro-Wilk** normality test on the four samples. Use a confidence level of 99% and write down the conclusion for each sample.

```
shapiro.test(sample1) # Normality test for sample 1
```

Hint 3.9: First think about what ***p-values*** *lead to rejecting the* ***null hypothesis*** H_0.

R code 3.4f:

Answer 3.4f:

In assignment 3.3 you used these **samples** and the **normal distribution** to estimate the **confidence interval** for the **population mean** . You also tested whether the actual population means were inside these intervals, which they almost always were.

Now you found out that most of these samples are actually not normally distributed at all.

3.4 g) If a sample is not normally distributed, does that mean you cannot use it to estimate the population mean? Explain your answer.

Answer 3.4g:

When the sample is not normally distributed, the sample **can** / **cannot** be used to estimate the population mean.

Explanation:

Assignment 3.5: Using R to inspect and test the homogeneity of variance

In this assignment, you will use the **iris** data set that is built into R. It is assumed that you have also installed the **car** package (from assignment 4.4).

Run the following code in R to load the **car** library and load the **iris** data set into the environment.

```
library(car)
data(iris)
```

3.5 a) Use the following code to create a **box plot** for the sepal length (**Sepal.Length**) per flower species (**Species**). Then rewrite the code to do the same for the sepal width (**Sepal.Width**) per flower species.

```
plot(x = iris$Species, y = iris$Sepal.Length,
     col = 'grey', main = 'Sepal Length')
```

R code 3.5a:

3.5 b) Visually inspect the graphs for **homogeneity of variance** . What can you say about the spread of the sepal length within the different iris species?

Answer 3.5b:

3.5 c) Formulate the **null hypothesis** H_0 and **alternative hypothesis** H_1 to test for **homogeneity of variance** in these samples.

Answer 3.5c:

H_0 : _____ H_1 : _____

3.5 d) Use the (**car** package) function **leveneTest()** to test the **homogeneity of variance** for the sepal length and width of the three species. Evaluate the hypotheses with a 90% confidence. Include the following elements:

- ☐ Discuss what the **p-value** is for this test.
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about the homogeneity of the three variances.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Hint 3.10: Look up **?leveneTest** *in R's help function (it is now also present because you have loaded the* **car** *package).*

R code 3.5d:

Answer 3.5d:

Chapter 4: Correlation and regression

Learning objectives of this chapter:

- Calculating covariance and correlation by hand
- Testing hypotheses about a correlation by hand
- Calculating and testing a correlation in R
- Testing hypotheses using linear regression in R
- Making predictions using linear regression in R

Assignment 4.1: Calculating covariance and correlation by hand

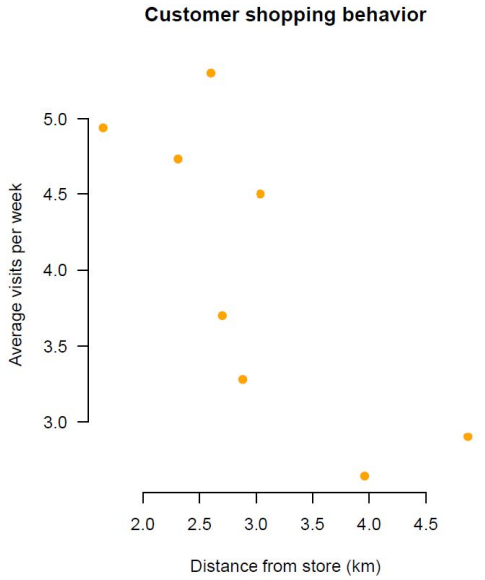
The manager of a local branch of a supermarket chain in the Netherlands has had a bad year last year and wants to improve their sales. Her co-worker has brought up the idea to spend more money on advertising in the neighborhoods that are further away from the store. The co-worker thinks that customers that live further away might visit the supermarket less frequently. He believes that making an effort to attract these people may prove to be profitable for the store. Realizing that her branch currently does not spend a notable amount of money on advertising at all, the manager is interested to know about the **relationship** between the distance a customer lives from the store and the average number of times they visit the store per week.

The manager requests her co-worker to ask the customers at his counter some questions when they check out at his counter. He is instructed to ask them a) how many times, on average, they visit the store per week and b) the number of kilometers they live from the store. The co-worker asks eight customers and reports the findings to his manager.

4.1 a) What kind of **relationship** do you expect to see? Formulate your answer in terms of the direction of the **relationship** .

Answer 4.1a:

The manager sits down with her co-worker, writes down the numbers, and creates a scatter plot of the data. She displays the distance a customer lives from the store (in kilometers) on the *x-axis* and their average number of visits per week on the *y-axis*.



i	x_i	y_i	$(x_i - \bar{x})$	$(y_i - \bar{y})$	$(x_i - \bar{x})(y_i - \bar{y})$
1	4.87	2.90			
2	3.04	4.50			
3	1.65	4.94			
4	2.88	3.28			
5	2.31	4.73			
6	3.96	2.64			
7	2.70	3.70			
8	2.60	5.30			

$\bar{x} =$ ..... $\sum(x_i - \bar{x})(y_i - \bar{y}) =$ .....

$\bar{y} =$ ..... $n - 1 =$ .....

$s_{xy} =$ .....

4.1 b) Use the table above to calculate and fill in the **covariance** s_{xy} between the customer's distance from the store and their average number of visits per week.

Hint 4.1: You can find the complete formula for the ***covariance*** *in the formula sheet.*

4.1 c) Interpret the **covariance** from these data. What can you say about the **relationship** between the distance from the store and the average number of visits per week on the basis of this measure?

Answer 4.1c:

The **covariance** s_{xy} is a useful measure to get an idea about the extent to which two **variables** change together, but it also has some disadvantages when using it as a measure of the strength of a **relationship** .

4.1 d) Explain the disadvantage of using the **covariance** s_{xy} as a measure for the strength of this **relationship** . Consider in your answer what would have happened if the co-worker asked the customers how many meters they lived from the store.

Answer 4.1d:

To reliably measure the strength of the **relationship** , the manager wants to **standardize** the **covariance** s_{xy} to find out the **correlation** coefficient r_{xy}. To do this, she first needs to calculate the sample **standard deviation** of the distance from the store in kilometers (s_x) and the sample **standard deviation** of the average number of visits per week(s_y).

4.1 e) Calculate s_x and s_y for the data that the co-worker collected.

Hint 4.2: You can find the formula for the ***standard deviation*** *in the formula sheet.*

Answer 4.1e:

s_x : s_y :

4.1 f) Use the sample **standard deviations** and the sample **covariance** to calculate the sample **correlation** r_{xy}.

Hint 4.3: You can find the formula for the ***correlation coefficient*** r_{xy} *in the formula sheet.*

Answer 4.1f:

r_{xy} :

4.1 g) Interpret the **correlation coefficient** . Is this a strong **relationship** ?

Answer 4.1g:

Assignment 4.2: Testing hypotheses about a correlation by hand

The store manager looks at the results of her calculations and sees some potential for increasing the sales of her branch. However, before she starts spending more money on advertising in these to hard-to-reach neighborhoods, she wants to gain reasonable assurance that her results can be generalized to all of her potential customers. Therefore, she wants to test the hypothesis that, with 95% confidence, this **correlation** is truly negative in the entire **population** of customers.

4.2 a) Formulate the **null hypothesis** H_0 and **alternative hypothesis** H_1 for the manager's test of the **population correlation coefficient** ρ_{xy}.

Hint 4.4: Think about whether this is a one-sided test or a two-sided test?

Answer 4.2a:

H_0 : _____ H_1 : _____

4.2 b) Write down the relevant information that appeared in assignment 4.1. Use the symbols N, n, r_{xy}, ρ_{xy}

Hint 4.5: Not all symbols are known.

Answer 4.2b:

N : _____ r_{xy} : _____

n : _____ ρ_{xy} : _____

4.2 c) Find out the **critical z-value** that is required to reject H_0 with 95% confidence.

Answer 4.2c:

$z_{critical}$: _____

4.2 d) Calculate the sample **z-value** for the **correlation coefficient** r_{xy}.

*Hint 4.6: You can find the formula for the **z-value** of a **correlation** in the formula sheet.*

Answer 4.2d:

z_{xy} : _____

4.2 e) Draw the conclusion for the manager. Include the following four elements:

- ☐ Show how the calculated **z-value** relates to the **critical z-value** .
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about ρ_{xy}.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 4.2e:

Assignment 4.3: Calculating and testing a correlation in R

The supermarket manager believes she has found a weakness in the distribution policy of the supermarket and believes that more stores should be built nationwide to be more active in other areas. To strengthen her case for the board of directors, she has asked permission to execute her survey in 100 stores in the Netherlands to gather data of 1000 customers. She wants to confirm her **hypothesis** that in the entire Netherlands, there is a negative **relationship** between the distance a customer lives from their supermarket, and the number of times they visit their supermarket.

For this assignment, you need the data file **localSupermarket.csv**, which contains a population of 1,000 observations.

4.3 a) Use the **read.csv()** function (and **setwd()** function if you prefer) to import the data set into a data frame called **dataset6**.

R code 4.3a:

```
```

4.3 b) Use the **cov()** and **cor()** functions to calculate the **covariance** s_{xy} and the **correlation** r_{xy} of the columns **Distance** and **AvgVisits**.

R code 4.3b:

```
```

Answer 4.3b:

Covariance: _____

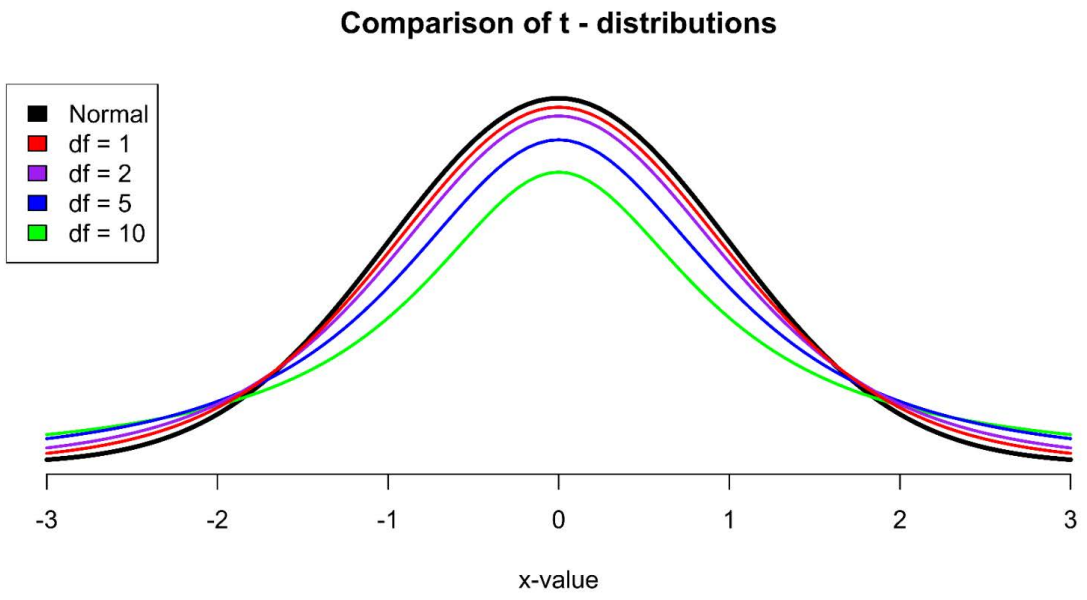
Correlation: _____

4.3 c) Use the function **cor.test()** to test for a negative **relationship** in this **population**. Use **?cor.test** to get help about the function arguments. Can you confirm the value of the **correlation** that you found in assignment 4.3b?

R code 4.3c:

```
```

Instead of using the **standard normal distribution** $N(\mu = 0, \sigma = 1)$ and a **z-value** for testing ρ_{xy} against any value, you can simplify the procedure when you are testing a **correlation coefficient** testing against the value zero (which is almost always the case). In such a case, you can use the **t-distribution** for calculating a **t-value** for the significance of ρ_{xy} against a value of zero. The **t-distribution** is almost identical to the **normal distribution** . However, where the **normal distribution** is defined by its **mean** (μ) and its **standard deviation** (σ), the **t-distribution** is defined by its degrees of freedom (df_{n-1}).



Use the following R code to create a figure of the **t-distribution** with $df = 3$.

```
curve(dnorm(x, mean = 0, sd = 1), from = -3, to = 3, ylab = 'Density')
curve(dt(x, df = 3), from = -3, to = 3, add = TRUE, col = 'red')
```

4.3 d) Which line represents the **normal distribution** in the figure that was drawn? And which line represents the **t-distribution** ? Can you explain what the difference between the two distributions is in terms of their shape? What happens when you increase the **degrees of freedom** (**df**) in the **dt()** function?

Answer 4.3d:

4.3 e) Calculate the **degrees of freedom** and the **t-value** for the **correlation coefficient** between **Distance** and AvgVisits.

*Hint 4.7: You can find the formula for the **degrees of freedom** and the **t-value** of a **correlation** in the formula sheet on page 111.*

R code 4.3e:

Answer 4.3e:

df : _____ t_{xy} : _____

4.3 f) Where do you find the **t-value** that you calculated in assignment 4.3e in the output of the **cor.test()** function from assignment 4.3c?

Answer 4.3f:

_____

4.3 g) What is the **p-value** for this hypothesis test? Interpret this **p-value** with respect to the confidence used and give a conclusion on the hypotheses. Include the following elements:

- ☐ Discuss what the **p-value** is for this test.
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about ρ_{xy}.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 4.3g:

_____

_____

_____

_____

_____

_____

Assignment 4.4: Testing hypotheses using linear regression in R

A manager from a local supermarket wants to increase the sustainability of her store. Since the supermarket is located in a farm village that has easy access to milk, many milk cartons are not sold, expire, and have to be thrown away. From common sense, the manager suspects that decreasing the price of milk will result in more sales, and thus fewer milk cartons that get thrown away. In order to increase her sustainability through this method, the manager sets out to find out if a decrease in the price per carton of milk will result in fewer milk cartons thrown away on an average day. She also wants to predict her decrease in waste when she lowers the price of a milk carton by 30 cents.

The manager goes to the company headquarters and asks the prices of milk (they vary) in every branch of the supermarket chain in the Netherlands. She also asks the average number of milk cartons thrown away per day of each branch.

For this assignment, you need the data file **`nationalSupermarket.csv`**, which contains the full **`population`** of 200 stores of the supermarket chain in the Netherlands.

4.4 a) Use the **`read.csv()`** function (and **`setwd()`** function if you prefer) to import the data set into a data frame called **`dataset7`**.

R code 4.4a:

```
```

4.4 b) Create a scatter plot of the data. Put the price per milk carton (**`Price`**) on the *x-axis* and the average number of milk cartons that was thrown away (**`AvgWasted`**) on the *y-axis*.

R code 4.4b:

```
```

The **`lm()`** function is used to fit a linear model (linear regression) in R. It requires that you specify a **`formula`** that tells the function what the **`regression equation`** is. If you want to fit a **`model`** where you predict an outcome variable **`Y`** on the basis of one predictor variable **`X`**, the formula is as follows:

Formula on paper: $Y = \beta_0 + \beta_1 \times X$ **Formula in R:** `Y ~ 1 + X`

4.4 c) Write down the **`regression equation`** for a linear model where you predict the average number of thrown away milk cartons per day (**`AvgWasted`**) on the basis of the price per milk carton (**`Price`**).

Answer 4.4c:

`AvgWasted =` _____

A linear **model** with these variables can be fitted in R by calling the **lm()** function with the **formula** and the **dataset** (see the R code below). The variables **X** and **Y** should correspond to the names of the corresponding variables in your **dataset** .

```
lm(formula = Y ~ 1 + X, data = dataset)
```

4.4 d) Fit a linear **model** where you use the data in **dataset7** to predict the average number of thrown away milk cartons per day (**AvgWasted**) on the basis of the price per milk carton (**Price**). Store this **model** in an object called **lmfit** .

R code 4.4d:

Now run the following code in R:

```
summary(lmfit)
```

4.4 e) Use the summary to find out the β_0 and $_1$ parameters and write them down in the **regression equation** below.

Answer 4.4e:

AvgWasted = β_0 (_____) + β_1 (_____) × Price

4.4 f) Using the **abline()** function, paste the **regression equation** line into your scatter plot from assignment 4.4b.

R code 4.4f:

4.4 g) What is the R^2 of the **lmfit** regression **model** ? What is the interpretation of this value?

Answer 4.4g:

R^2 = _____

Interpretation:

_____

_____

From the regression line, the manager observes that there is indeed a positive **relationship** between the price of a carton of milk and the average number of milk cartons thrown away per day. To be sure she wants to test the hypothesis that, with 95% confidence, the price of milk is a good predictor of the average number of milk cartons that are thrown away per day, and that this **relationship** is truly positive.

4.4 h) Formulate the **null hypothesis** H_0 and the **alternative hypothesis** H_1 for the manager's test of the population coefficient β_1 for the price of a carton of milk.

Answer 4.4h:

H_0: _____ H_1: _____

4.4 i) Use the **summary** in R to draw the conclusion for the manager. Include the following four elements:

- ☐ Discuss what the **p-value** is for this test.
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about β_1.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 4.4i:

Assignment 4.5: Making predictions using linear regression in R

The manager now wants to know what will happen to the number of milk cartons that she has to throw away when she lowers the price of a milk carton by 30 cents, from \$1.00 to \$0.70. Currently, the supermarket throws away 4 cartons of milk each day.

4.5 a) Create a new data frame that has only one column that includes the new value for the price of milk (the column has to be named exactly the same as in `dataset7`).

R code 4.5a:

```
```

The `predict()` function can be used to predict new data according to a **`linear model`**.

4.5 b) Use the `predict()` function to predict the number of milk cartons that the supermarket will have to throw away when the `Price` is \$0.70.

R code 4.5b:

```
```

Answer 4.5b:

```
Predicted value:  _____________
```

4.5 c) Confirm this estimate by writing out the regression equation of the **`linear model`** from assignment 4.4e and filling in the new value of the price of milk.

Answer 4.5c:

```
Predicted value:  ___________________________
```

The `predict()` function can also be used to construct a **`confidence interval`** for the predicted number of cartons thrown away by using `interval = 'prediction'`.

4.5 d) Create a 90% **confidence interval** for the predicted number of milk cartons thrown away.

R code 4.5d:

4.5 e) When the manager lowers her price from \$1.00 to \$0.70, will the supermarket throw away fewer cartons of milk? Incorporate the 90% **confidence interval** for the predicted value in your answer.

Answer 4.5e:

The supermarket **will** / **will not** throw away fewer cartons of milk.

Explanation:

Chapter 5: Comparing one or two means

Learning objectives of this chapter:

- Hypothesis testing by hand using the z- and t-score
- Generating and interpreting z-scores and t-scores
- Independent two-sample t-test by hand and in R
- Dependent two-sample t-test in R

Assignment 5.1: Hypothesis testing by hand using the z- and t-score

The maximum allowed amount of PFAS chemicals in soil is 0.9 microgram per kg of dry soil. In a regular soil check, researchers take 49 soil **samples** in a town near a plastic factory and find a **mean** PFAS level of 0.918 microgram/kg with a **standard deviation** of 0.071. They want to show with 95% confidence that the PFAS levels in the town are significantly above the norm.

5.1 a) Write down the relevant information from this case.

Answer 5.1a:

_____

_____

5.1 b) Write down the **null hypothesis** H_0 and the **alternative hypothesis** H_1 for the researchers' test of the **mean** PFAS level against the norm.

Answer 5.1b:

H_0: _____ H_1: _____

The **critical z-value** for a one-sided 95% confidence interval is equal to 1.645 (see also Table 1 on page 113)

5.1 c) Calculate the **lower bound** of the one-sided **confidence interval** for the PFAS level **population mean** with 95% confidence.

Answer 5.1c:

Lower bound: _____

Calculation:

_____

5.1 d) Draw the conclusion for the researchers. Include the following four elements:

- ☐ Show how μ_0 relates to the **confidence interval.**
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about μ and μ_0.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 5.1d:

_____

_____

_____

_____

_____

_____

5.1 e) Calculate the **z-score** for this situation.

*Hint 5.1: You can find the formula for the **z-score** in the formula sheet on page 111. Use μ_0 in this formula.*

Answer 5.1e:

z-score: _____

Calculation:

_____

5.1 f) Compare the calculated **z-score** with the **critical z-value**. What does this tell you about the **p-value**?

*Hint 5.2: Use the absolute value of the calculated **z-score**.*

Answer 5.1f:

5.1 g) Draw the conclusion again, but now using the **z-score**, is it the same? Include the following elements:

- ☐ Show how the calculated **z-score** relates to the **critical z-score**.
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about μ and μ_0.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 5.1g:

The researchers want to quickly check another town near the plastic factory, but have less time available. They take 16 **samples** in the second town and find a much higher **mean** PFAS level of 0.930 microgram/kg with a comparable **standard deviation** of 0.070.

5.1 h) Calculate the **t-score** for this situation.

*Hint 5.3: You can find the formula for the **t-score** in the formula sheet on page 111. Use μ_0 in this formula.*

Answer 5.1h:

t-score: _____

Calculation:

_____

The **critical t-value** for a one-sided 95% **confidence interval** with 15 **degrees of freedom** is equal to 1.753.

5.1 i) Draw the conclusion for the researchers using the (absolute) t-score. Include the following elements:

- ☐ Show how the calculated **t-score** relates to the **critical t-score** .
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about μ and μ_0.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 5.1i:

_____

_____

_____

_____

_____

_____

5.1 j) Also calculate the **lower bound** of the one-sided **confidence interval** for the PFAS level **population mean** with 95% confidence.

Answer 5.1j:

Lower bound: _____

Calculation:

_____

5.1 k) If you would draw the conclusion based on this **lower bound** would you get the same result? Explain why.

Answer 5.1k:

The conclusion **would** / **would not** be the same.

Explanation:

5.1 l) Why is it that even though the second town shows much higher PFAS level in the **sample** (0.930 vs 0.918) H_0 cannot be rejected?

Answer 5.1l:

Assignment 5.2: Generating and interpreting z-scores and t-scores

In assignment 5.1 you have seen that to calculate a 95% one-sided **confidence interval** for a **sample** with $n \geq 30$ you need to use the correct **z-value** . Now you are going to find out how to get that number from R using the **qnorm()** function. The **qnorm()** function returns the **x** -value for a certain probability for a specified **normal distribution** , if you do not specify the **mean** (**mean**) and **standard deviation** (**sd**) arguments it returns the **z-value** for the **standard normal distribution** .

Run the following code in R:

```
qnorm(p = 0.95, mean = 0, sd = 1, lower.tail = TRUE)
# Or because the standard normal distribution is the default simply use:
qnorm(0.95)
```

5.2 a) Round the number to 3 decimals. Do you recognise this **z-value** ?

Answer 5.2a:

Result: _____

_____

In assignment 5.1 you have also seen that for the 95% two-sided **confidence interval** for a **sample** with $n \geq 30$ you can use z = 1.960. Now you are going to learn to reproduce that number in R.

Run the following code in R:

```
qnorm(p = 0.975)
```

5.2 b) Why do you have to use the 97.5 percentile to get the z-value for the 95% two-sided **confidence interval** ?

Answer 5.2b:

_____

_____

Run the following code in R:

```
pnorm(q = 1.645, mean = 0, sd = 1)
pnorm(1.960)
```

5.2 c) Based on the results of this R code explain what the **pnorm()** function does.

Answer 5.2c:

In assignment 5.1 it was stated that the **critical t-value** for a one-sided 95% **confidence interval** with 15 **degrees of freedom** is equal to 1.753.

Run the following code in R:

```
qt(p = 0.95, df = 15)
```

5.2 d) Based on the results of this R code explain what the **qt()** function does.

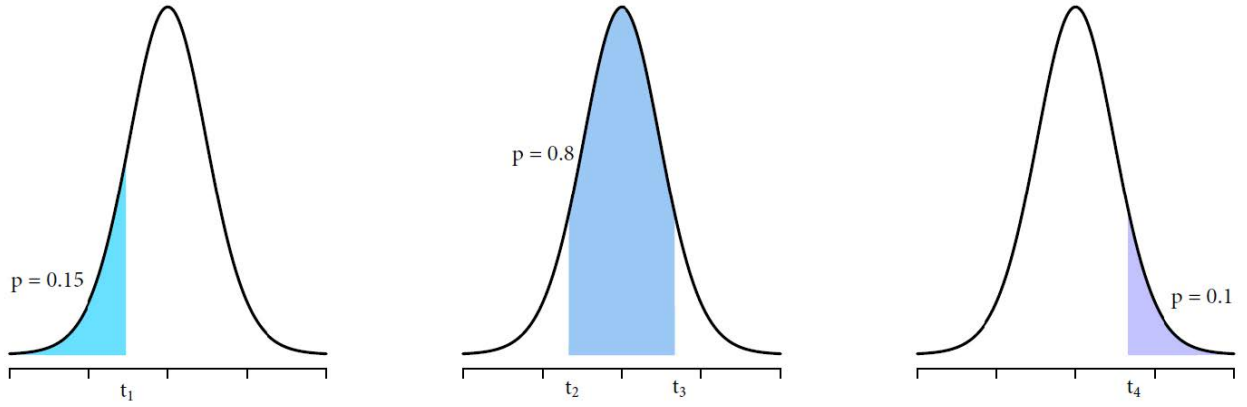
Answer 5.2d:

Run the following code in R:

```
pt(q = 1.753, df = 15)
```

5.2 e) Based on the results of this R code explain what the **pt()** function does.

Answer 5.2e:



5.2 f) For the situation where there are 19 degrees of freedom, use the **`qt()`** function to get the **`t-values`** (t_1, t_2, t_3, t_4) for the three situations shown in the picture above.

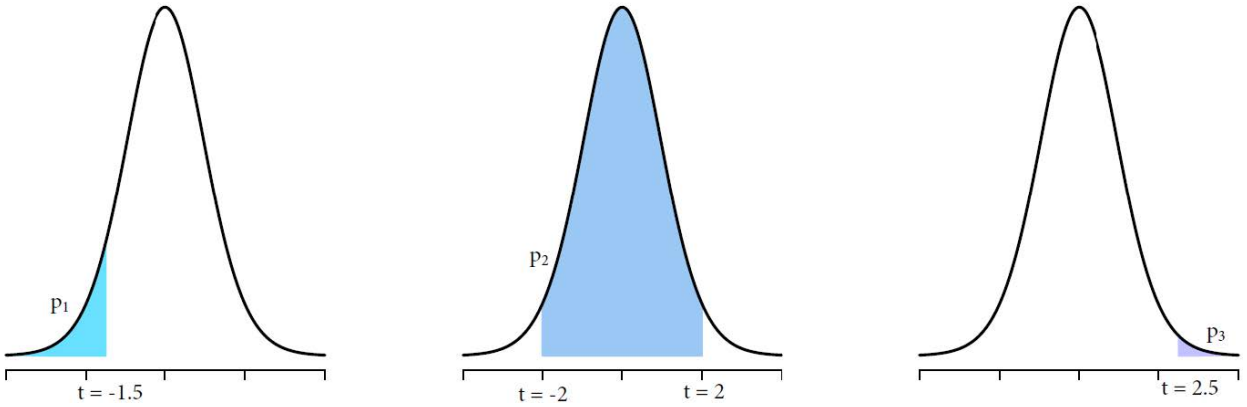
Hint 5.4: For the picture on the right keep in mind that the **`qt()`** *function by default returns the value for the left tail of the distribution.*

R code 5.2f:

Answer 5.2f:

t_1: _____ t_2: _____

t_3: _____ t_4: _____



5.2 g) For the situation where there are 11 **degrees of freedom**, use the **pt()** function to get the probabilities p_1, p_2, and p_3 (area under the curve) for the three situations shown in the figures above.

Hint 5.5: For the picture in the middle and the right keep in mind that the **pt()** *function by default uses the left tail and the total area under the curve is 1.*

R code 5.2g:

Answer 5.2g:

p_1: _____ p_2: _____ p_3: _____

When you test a value for a **population mean**, there exist three types of tests:

Two-tailed inequality test:	One-tailed right sided test:	One-tailed left sided test:
$H_0 : \mu = \mu_0$	$H_0 : \mu \leq \mu_0$	$H_0 : \mu \geq \mu_0$
$H_1 : \mu \neq \mu_0$	$H_1 : \mu > \mu_0$	$H_1 : \mu < \mu_0$

5.2 h) Give a critical **t-value** for the three tests described above using a confidence of 98% and 28 **degrees of freedom**.

Hint 5.6: Use the **qt()** *function in* R.

R code 5.2h:

Answer 5.2h:

```
Two-tailed inequality test:    _____

One-tailed right sided test:   _____

One-tailed left sided test:    _____
```

Three $n = 29$ **samples** were done on three different populations and the **t-score** based on these **samples** was calculated:

Two-tailed inequality test:	One-tailed right sided test:	One-tailed left sided test:
$H_0 : \mu = 50$ $H_1 : \mu \neq 0$	$H_0 : \mu \leq 0$ $H_1 : \mu > 0$	$H_0 : \mu \geq 0$ $H_1 : \mu < 0$
$t = 2.6$	$t = -2.3$	$t = 1.6$

5.2 i) Using a confidence of 98%, find out whether the **null hypothesis** H_0 is rejected for each of these three cases.

Hint 5.7: Use the ***critical t-values*** *from the previous assignment and compare these with the sample* ***t-scores*** *to draw your conclusion for each of these outcomes.*

Answer 5.2i:

Two-tailed inequality test: H_0 **rejected** / H_0 **not rejected**

Explanation: _____

One-tailed right sided test: H_0 **rejected** / H_0 **not rejected**

Explanation: _____

One-tailed left sided test: H_0 **rejected** / H_0 **not rejected**

Explanation: _____

Assignment 5.3: Independent two-sample t-test by hand and in R

To test the effect of caffeine on the respiratory exchange ratio (RER), 9 men get caffeine and 9 different men get a placebo, their RER is measured while doing sports[1]:

RER measurement	n	Mean ($\bar{x}$)	Standard deviation (s_x)
Caffeine	9	94.22	5.61
Placebo	9	100.56	7.70

Researchers want to use this experiment to show with 95% confidence that caffeine reduces the RER in men. They are going to evaluate the results by comparing the **mean** RER using a **two-sample t-test** .

5.3 a) Why is this test also called an **independent samples t-test** ?

Answer 5.3a:

_____

_____

5.3 b) Write down the **null hypothesis** H_0 and the **alternative hypothesis** H_1 for a one-sided test where the researchers want to show that the placebo **mean** RER is higher than the caffeine **mean** RER.

Answer 5.3b:

H_0: _____ H_1: _____

In the formula sheet on page 111 you can see that for **independent samples t-tests** :

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - D_0}{\sqrt{s_p^2(\frac{1}{n_1} + \frac{1}{n_2})}} \qquad s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

5.3 c) Using the formula provided calculate the **pooled standard deviation** s_p.

[1]These data are taken from http://learntech.uwe.ac.uk/da/Default.aspx?pageid=1438

Answer 5.3c:

s_p: _____

Calculation:

_____

5.3 d) Using the formula provided calculate the **t-score** .

Answer 5.3d:

t-score: _____

Calculation:

_____

The **critical t-value** for 16 **degrees of freedom** (n_1+n_2-2) for a one-sided test is 1.746.

5.3 e) Draw the conclusion for this test. Include the following elements:

- ☐ Show how the calculated **t-score** relates to the **critical t-score** .
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about μ and μ_0.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 5.3e:

_____

_____

_____

_____

_____

_____

Now you will learn how to do the same independent t-test in R.

Run the following code in R:

```
# These are the values for the RER test data set
placebo <- c(105, 119, 100, 97, 96, 101, 94, 95, 98)
caffeine <- c(96, 99, 94, 89, 96, 93, 88, 105, 88)
```

5.3 f) Calculate the **mean** and **standard deviation** for the **placebo** and **caffeine** variables, confirm that your results match with the previous assignment.

R code 5.3f:

Answer 5.3f:

placebo		**caffeine**	
Mean:	_____	Mean:	_____
Standard deviation:	_____	Standard deviation:	_____

Run the following code in R:

```
t.test(x = placebo, y = caffeine, alternative = 'greater', mu = 0,
       paired = FALSE, var.equal = TRUE, conf.level = 0.95)
```

5.3 g) Interpret the R code above and check if the results match your conclusion for assignment 5.3e.

Answer 5.3g:

_____

_____

The **t-test** assumes equal **variances** . In this case the **standard deviations** are actually a bit different.

5.3 h) Change the **var.equal** argument in the R code above to do a **Welch t-test** to account for unequal **variances** . Check whether the results differ.

R code 5.3h:

Answer 5.3h:

5.3 i) How can you check if you need to do a **Welch t-test** instead of a normal **t-test** ?

Answer 5.3i:

Assignment 5.4: Dependent two-sample t-test in R

To test for a difference in blood pressure depending on their position, 12 people's blood pressure was measured twice, first standing up and then lying down. Researchers want to use this experiment to show with 92.5% confidence that the blood pressure lying down is significantly higher than standing up. They are going to evaluate the results by comparing the **mean** using a **two-sample t-test** .

5.4 a) Why is this a dependent (also called paired) **t-test** ?

Answer 5.4a:

Run the following code in R:

```
# These are the values for the blood pressure data set
standing <- c(132, 146, 135, 141, 139, 162, 128, 137, 145, 151, 131, 143)
lying <- c(136, 145, 140, 147, 142, 160, 137, 136, 149, 158, 120, 150)
```

5.4 b) Calculate the **mean** of the **standing** and **lying** data sets, also calculate the differences and the **mean** of the differences.

R code 5.4b:

Answer 5.4b:

Mean standing: _____

Mean lying: _____

Mean difference: _____

5.4 c) Write down the **null hypothesis** H_0 and **alternative hypothesis** H_1 for a one-sided test where the researchers want to show with 92.5% confidence that the **mean** blood pressure is higher lying down than standing up.

Answer 5.4c:

H_0: _____ H_1: _____

Run the following code in R:

```
t.test(x = standing, y = lying, alternative = 'greater', mu = 0 ,
       paired = TRUE, conf.level = 0.925)
```

5.4 d) Check the results of this test. The outcome is a bit strange, can you explain what is wrong here?

Answer 5.4d:

_____

5.4 e) Fix the R code to do a correct test and draw the correct (four part) conclusion.

R code 5.4e:

Answer 5.4e:

_____

_____

_____

_____

_____

_____

Chapter 6: Comparing more than two means

Learning objectives of this chapter:

- Implementing a t-test as a regression in R
- Performing an ANOVA in R
- Implementing an ANOVA as a regression in R
- Introducing a covariate in ANCOVA in R

Assignment 6.1: Implementing a t-test as a regression in R

Many of the statistical tests that we have seen are actually equivalent to a specific form of **linear regression**. To understand how a **t-test** can be implemented as a **linear regression**, let's look at an example. Suppose you work in advertising and show four groups of people an advertisement, where the only difference is in the color/position of the eyes of the model (blue eyes, brown eyes, green eyes, or downward-looking eyes), and you ask them how they rate your brand after seeing this advertisement. For this assignment, we will use the **eyeColor.csv** data file that contains 222 participants' ratings for one of the four groups. As a first question, you want to assess whether there is a difference in the ratings if the model shown in the advertisement has blue eyes rather than brown eyes.

6.1 a) Read in the file **eyeColor.csv** and store the data in an object called **dataset8**.

R code 6.1a:

```
```

Note that this data set contains all four groups (**Blue**, **Brown**, **Green**, **Down**). Run the following R code to isolate the scores of the groups that were shown advertisements with **Blue** and **Brown** eyes and store them in the new **ttestData** object.

```
ttestData <- subset(dataset8,
                    dataset8$Group == 'Blue'|
                    dataset8$Group == 'Brown')
```

6.1 b) Write down the **null hypothesis** H_0 and **alternative hypothesis** H_1 for testing whether the **mean** score of the group that was shown **Blue** eyes is equal to the **mean** score of the group that was shown **Brown** eyes. Remember that these are independent **samples**.

Answer 6.1b:

H_0: _____ H_1: _____

6.1 c) Use the **t.test()** function to test the equality of the two means.

Hint 6.1: Make sure to specify **var.equal = TRUE** *to perform a* ***two-sample t-test*** *instead of the non-parametric* ***Welch's t-test*** .

R code 6.1c:

6.1 d) What is your conclusion on the basis of these results? Include the following elements:

- ☐ Discuss what the **p-value** is for this test.
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about μ_{Blue} and μ_{Brown}.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 6.1d:

Now, instead of using the **t.test()** function to test the equality of the two **means** , you can test the equality of the two **means** using a **regression model** . Therefore, you need to add a variable to our data that says whether a participant saw one of the two groups (e.g., only the people that were shown **Brown** eyed models).

Run the following code in R:

```
dummyBrown <- as.numeric(ttestData$Group == 'Brown')
ttestData <- cbind(ttestData, dummyBrown)
```

6.1 e) What are the contents of **dummyBrown** ? What do we call this kind of variable?

Answer 6.1e:

6.1 f) Create a **linear model** in R where you predict the (outcome) variable **Score** using only the (predictor) variable **dummyBrown** . Store the fitted model in an object called **ttestreg** .

R code 6.1f:

```
```

6.1 g) Use the **summary()** function to inspect the results of the **ttestreg** model. How does this output correspond to the **t-test** that you performed in assignment 6.1c? Where do you find the **p-value** that you calculated using the **t.test()** function?

R code 6.1g:

```
```

Answer 6.1g:

Assignment 6.2: Performing an ANOVA in R

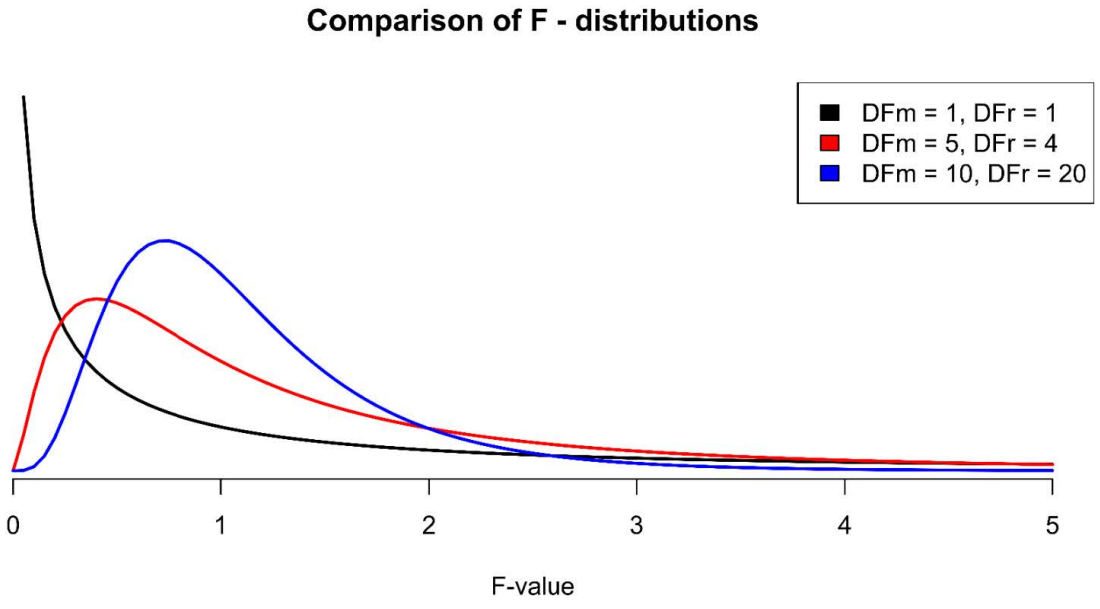
Now let's extend the analysis from assignment 6.1 by comparing all four groups (**Blue** , **Brown** , **Green** , **Down**) instead of only the **Blue** and **Brown** groups. When you are testing more than two **means** , you can use an **ANOVA** test (a specific form of regression). Since you want to compare all four groups, you can leave the **ttestData** from the previous assignment and focus on the data in **dataset8** . Remember that you are interested in testing the effect of the model's eye color on the rating of your brand.

6.2 a) Write down the **null hypothesis** H_0 and the **alternative hypothesis** H_1 for testing whether the **mea** n score of the four groups (**Blue** , **Brown** , **Green** , **Down**) are equal.

Answer 6.2a:

H_0: _____ H_1: _____

The **ANOVA** uses the **F-distribution** to test for a significant difference between all four **means** . Using the (two types of) **degrees of freedom** of this **F-distribution** , you can calculate the **critical F-value** that is required to reject the **null hypothesis** that the **means** of the four groups are equal. Table 5 on page 117 contains the **critical F-values** for a confidence of 95%.



6.2 b) Calculate the **degrees of freedom** df_M and df_R of the **F-distribution** for the data in **dataset8** .

Hint 6.2: You can find the formulas for df_M and df_R in the formula sheet on page 111.

Answer 6.2b:

df_M : _____ df_R : _____

6.2 c) Using the **qf()** function, calculate the **critical F-value** that is required to reject the **null hypothesis** H_0 for these data with 95% confidence.

R code 6.2c:

```
```

Answer 6.2c:

```
Critical F-value: _____
```

The **aov()** function in R is a wrapper for the **lm()** function. The difference between these two functions is that the **lm()** function can only handle **categorical** predictors with two levels (e.g., a **dummy variable**). The **aov()** function can handle **categorical** predictor variables with more than two levels, since it automatically rewrites the **formula** to include the **dummy variables** .

6.2 d) Use the **aov()** function to perform an **ANOVA** with the dependent (outcome) variable **Score** and the independent (predictor) variable **Group** and store the result in an object named **anovaResult** .

*Hint 6.3: You can check more information on the **aov()** function with **?aov** .*

R code 6.2d:

```
```

6.2 e) Use the **summary()** function to inspect the results of the **ANOVA** in **anovaResult**. What is the **F-value** that is calculated from the **sample**? What is the **p-value** calculated from the **sample**?

R code 6.2e:

```
```

Answer 6.2e:

F-value: _____ p-value: _____

6.2 f) What is your conclusion on the basis of these results? Include the following elements:

- ☐ Discuss what the **p-value** is for this test.
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about μ_{Blue}, μ_{Brown}, μ_{Green}, and μ_{Down}.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 6.2f:

Assignment 6.3: Implementing an ANOVA as a regression in R

Now that you have seen the results of the **ANOVA**, let's try to replicate these by implementing the same **ANOVA** as a **linear regression**. Remember that this is exactly what you did for the **t-test** in assignment 6.1 by adding one **dummy variable** to your model that isolated the **Brown** group. For the **ANOVA**, you are going to have three **dummy variables** in your model, one that represents **Brown** eyes, one that represents **Blue** eyes, and one that represents **Green** eyes. You first have to add these dummy variables to your data set.

Run the following code in R that adds a **dummy variable** for **Brown** eyes to the data set:

```
dummyBrown <- as.numeric(dataset8$Group == 'Brown')
dataset8 <- cbind(dataset8, dummyBrown)
```

6.3 a) Add two more **dummy variables** to the data in **dataset8**, one for **Blue** eyes and one for **Green** eyes. Name these variables **dummyBlue** and **dummyGreen**.

R code 6.3a:

```

```

6.3 b) Create a **linear model** in R where you predict the (outcome) variable **Score** using the (predictor) variables **dummyBrown**, **dummyGreen**, and **dummyBlue**. Store the fitted **linear model** in an object called **anovaReg**.

R code 6.3b:

```

```

6.3 c) Use the **summary()** function to inspect the results of the **linear model** stored in **anovaReg**. What is the **F-value** of the model? What is the **p-value** of this model?

R code 6.3c:

```

```

Answer 6.3c:

F-value: _____ p-value: _____

6.3 d) Do the **F-value** and **p-value** of this **linear model** match those of the **ANOVA** in assignment 6.2?

Answer 6.3d:

YES / NO

Assignment 6.4: Introducing a covariate in ANCOVA in R

There might be other determinants that influence people's ratings of your brand that you have not captured by varying the eye color in the advertisements. An example of this might be people's initial rating of your brand. These kinds of variables are called **covariates** and you can incorporate them in our **ANOVA**, very smoothly resulting in an **ANCOVA**. In our scenario, we want to incorporate the **covariate** for the initial score that our raters gave by adding the **initialScore** variable to our **linear model**.

6.4 a) Create a **linear model** in R where you predict the (outcome) variable **Score** using the (predictor) variables **dummyBrown**, **dummyGreen**, **dummyBlue**, and the variable **initialScore**. Store the fitted model in an object called **ancovaReg**.

R code 6.4a:

6.4 b) Use the **summary()** function to inspect the results of the **linear model** stored in **ancovaReg**. What is the **F-value** of the model? What is the **p-value** of this model?

R code 6.4b:

Answer 6.4b:

F-value: _____ p-value: _____

6.4 c) What is your conclusion on the basis of these results? Include the following elements:

- ☐ Discuss what the **p-value** is for this test.
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about μ_{Blue}, μ_{Brown}, μ_{Green}, and μ_{Down}, given the covariate.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 6.4c:

6.4 d) Can you tell whether **initialScore** is a good predictor of the **Score** ? On what value can you base your conclusion?

Hint 6.4: First consider which results you would expect if $\beta_3 \neq 0$.

Answer 6.4d:

_____

_____

To find out whether adding this **covariate** is an improvement over the **linear model** in assignment 6.3, we can compare the two **linear models** **anovaReg** (without **initialScore**) and **ancovaReg** (with **initialScore**) with respect to their proportion of **explained variance** (their R^2).

6.4 e) What is the (multiple) R^2 of the **anovaReg** **model** ? What is the (multiple) R^2 of the **ancovaReg** **model** ? Which **model** explains more variation in the outcome variable **Score** ?

Answer 6.4e:

R^2 **anovaReg** : _____ R^2 **ancovaReg** : _____

The **anovaReg** / **ancovaReg** regression model explains more variation in the outcome variable score.

6.4 f) Interpret the R^2 for the best model.

Answer 6.4f:

_____

_____

The R^2 statistic will always increase when you add more (predictor) variables to our **model** , since you are adding more information. To reliably compare our two **models** , you have to look at a measure that penalizes a **model** for including more (predictor) variables. You can use the **AIC** value for that. The rule of thumb for the **AIC** value is that the **model** with the lower **AIC** value is the preferred model.

6.4 g) Use the **AIC()** function to calculate the **AIC** value of the **anovaReg** and the **ancovaReg** models.

R code 6.4g:

Answer 6.4g:

AIC **anovaReg** : _____ AIC **ancovaReg** : _____

6.4 h) What is the preferred model? How can you use the **AIC** statistic to validate you answer in assignment 6.4d?

Answer 6.4h:

_____

_____

Assignment 6.5: Using post-hoc tests in R to find differences in means

The state of Iowa in the USA receives many invoices for services that they buy. In turn, these invoices need to be paid in a timely manner. The questions has been raised whether the state of Iowa pays all invoices equally timely. To investigate this you are requested to perform an audit. In this audit you set out to statistically check if you can find differences -between the various services that are bought- in the time that it takes for Iowa to pay an invoice.

For this assignment you will have to download the data file **`iowa.RData`** from the online resources[2]. **`.RData`** files are compressed R objects, and are useful when dealing with very large data sets such as this one.

The **`iowa.RData`** file contains payment transactions recorded in the State of Iowa's central accounting system for the Executive Branch and is real data.

6.5 a) Load the **`iowa.RData`** data file into the environment using the **`load()`** function.

R code 6.5a:

```
```

The data set is now stored in object called **`iowa`** .

6.5 b) Give a short description of the data in **`iowa`** .

Hint 6.5: Search the internet for the source of these data to find out what the columns represent.

Answer 6.5b:

_____

_____

_____

6.5 c) How many rows and columns does the **`iowa`** data set have?

Answer 6.5c:

`Rows:` _____ `Columns:` _____

[2]These data are taken from https://data.iowa.gov/State-Government-Finance/State-of-Iowa-Checkbook/cyqb-8ina.

6.5 d) How many unique services are there? Make a **frequency table** of these services.

R code 6.5d:

```
```

Answer 6.5d:

```
Unique services: _____
```

6.5 e) Which service has the most rows? How many rows does this service have?

Answer 6.5e:

```
Service: _____ Rows: _____
```

6.5 f) How many rows show a difference in invoice date and payment date?

R code 6.5f:

```
```

Answer 6.5f:

```
Number of rows that show a difference: _____
```

6.5 g) Create a new data set that consists of these differences, and name the new data set **dataDif** .

R code 6.5g:

```
```

6.5 h) Create an extra column named **dif.days** in **dataDif** that contains the number of days between invoice and payment.

Hint 6.6: Make sure that the column **dif.days** *is numeric.*

R code 6.5h:

```
```

6.5 i) Calculate the **minimum**, **maximum**, **mean**, **quartiles**, and **standard deviation** of the column **dif.days**.

R code 6.5i:

```
```

Answer 6.5i:

Minimum:	_____	Upper quartile:	_____
Mean:	_____	Lower quartile:	_____
Maximum:	_____	Standard deviation:	_____

6.5 j) Create a histogram of the column **dif.days**. Describe what you see in the histogram.

R code 6.5j:

```
```

Answer 6.5j:

_____

6.5 k) Again, create a histogram, but now only use the subset of **dif.days** that is in the 5-95% quantile range (so you cut off the bottom and top 5%).

Hint 6.7: Hint: use the **quantile()** *function.*

R code 6.5k:

```
```

You don't trust the negative values in **dif.days** as you cannot interpret them, and therefore you will not include them in your investigation. Moreover, you also don't want to include value in **dif.days** that are higher than 365 days.

6.5 l) Create a new data set in which these values are removed and name this data set **dataDif2** .

R code 6.5l:

```
```

6.5 m) Create a scatter plot with **dif.days** on the *y-axis* and **Amount** on the *x-axis*.

R code 6.5m:

```
```

6.5 n) Compute the **correlation** between the time between invoice and payment, and the amount that is paid.

R code 6.5n:

```
```

Answer 6.5n:

Correlation: _____

6.5 o) Elaborate on the **correlation coefficient** and it's significance. What does this imply?

Answer 6.5o:

6.5 p) Compute the **mean** `dif.days` per expense category.

Hint 6.8: Use the `aggregate()` *function (for more help on this function see* `?aggregate` *).*

R code 6.5p:

```
```

Answer 6.5p:

6.5 q) Use the `aov()` function to test whether the **means** that you computed in assignment 6.5p are statistically different.

Answer 6.5q:

p-value: _____

Conclusion:

R code 6.5q:

```
```

6.5 r) Use **Tukey's Honest Significant Differences** to find out which group **means** are truly different.

Hint 6.9: Use the `TukeyHSD()` *function to find* ***Tukey's Honest Significant Differences*** .

R code 6.5r:

```
```

Answer 6.5r:

In assignment 6.5q you have computed an **ANOVA** , but this is statistically not completely sound.

6.5 s) Can you formulate why the **ANOVA** in assignment 6.5q was not statistically sound? What would be an appropriate analysis?

Answer 6.5s:

6.5 t) Why do you think it is a good or bad idea to calculate **p-values** if the number of rows in the data is large?

Answer 6.5t:

Chapter 7: Comparing proportions and distributions

Learning objectives of this chapter:

- Chi-square testing by hand
- Performing a Chi-square test in R
- Proportion estimation and testing by hand and in R

Assignment 7.1: Chi-square testing by hand

A small bed and breakfast wants to know if the **distribution** of the number of guests is changing. They compare their historical **distribution** with the number of guests in 2019 and want to show, with 95% confidence, that the shape of the 2019 **distribution** has changed with respect to the past years.

7.1 a) Formulate the **null hypothesis** H_0 and the **alternative hypothesis** H_1 for this test in words.

Answer 7.1a:

H_0: _____

H_1: _____

The table below shows the historical **distribution** of the bed and breakfast guests alongside the number of guests for 2019.

	Historical	Observed (O)	Expected (E)	$O - E$	$\frac{(O-E)^2}{E}$
Spring	4.87	2.90			
Summer	3.04	4.50			
Fall	1.65	4.94			
Winter	2.88	3.28			
Total	2.31	4.73			

7.1 b) Fill in the expected (E) column in the table and show how you calculated it below.

Answer 7.1b:

_____

_____

7.1 c) Are you allowed to do a **chi-square test** using these data? Explain why.

Hint 7.1: Take into account the assumptions of a ***chi-square test*** .

Answer 7.1c:

7.1 d) Fill in the rest of the table to calculate the **chi-square value** (X^2) for this sample.

Answer 7.1d:

X^2: _____

The critical **chi-square value** using a confidence of 95% and 3 **degrees of freedom** is $X^2_{0.05}[df = 3] = 7.815$.

7.1 e) What is your conclusion on the basis of these results? Include the following elements:

- ☐ Show how the calculated **Chi-square value** relates to the **critical Chi-square value** .
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about the historical **distribution** and the 2019 **distribution** .
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 7.1e:

7.1 f) Taking into account the confidence of our analysis, explain in what range the **p-value** of this test must be.

Answer 7.1f:

The bed and breakfast owner does not believe in your calculations and wants you to recalculate it in R.

Run the following code in R to import the data:

```
Observed <- c(37, 27, 12, 8)
Historical <- c(0.3, 0.4, 0.15, 0.15)
Expected <- c(25.2, 33.6, 12.6, 12.6)
```

7.1 g) Calculate the **Chi-square value** for these data using R.

R code 7.1g:

Answer 7.1g:

X^2: _____

Run the following code in R to find out the **critical Chi-square value** :

```
qchisq(p = 0.95, df = 3)
```

7.1 h) Does your conclusion hold up when you recalculate the **Chi-square value** in R?

Answer 7.1h:

YES / NO

Use the following code in R to perform a **Chi-square test** :

```
# chi-square test: x = observations p = model distribution
# rescale makes sure the model distribution adds up to 100%
chisq.test(x = Observed, p = Historical, rescale.p = TRUE)
```

7.1 i) Is the result what you expected?

Answer 7.1i:

As it turns out the function **chisq.test()** will always calculate the **expected values** by itself. You therefore do not need to make sure your total amounts sum to one. Therefore, in the code after **p =** you can supply a distribution that adds up to one (and set **rescale.p = FALSE**), or expected amounts that R will recalculate into a distribution anyway. If you do not supply the **p** argument, R will test against the **uniform distribution** .

Run the following code in R to store the results of the **Chi-square test** in an object called **chisq** :

```
chisq <- chisq.test(x = Observed, p = Historical)
```

7.1 j) Find out how to extract the **expected values** from the **chisq** object. Do they match the **expected values** you wrote down in the table above?

R code 7.1j:

Answer 7.1j:

Assignment 7.2: Chi-square testing in R

A gift company in the Netherlands wants to apply statistics to gain insight into their sales activity. This gift company normally generates most of its revenue in the period before summer and on Christmas holidays. Recently they opened a new store in a different country. The gift company wants to check, using a **Chi-square** (X^2) test, whether the new store has a different seasonal pattern than the stores in the Netherlands.

Run the following code in R to create the data set and store it in an object named **sales** .

```
# These are the values for the sales data set
sales <- data.frame(month = seq(from = 1, to = 12, by = 1),
                    historical = c(5.1, 5.1, 6.7, 10, 11.4, 10,
                                   6.7, 5.1, 6.7, 10, 11.7, 11.7),
                    newstore = c(5.6, 6.2, 9.4, 8.6, 6.8, 4.8,
                                 5.6, 4.8, 8.8, 12.6, 13.1, 13.7))
```

7.2 a) Explore the **sales** object and describe what it contains.

Hint 7.2: You can use the **summary()** *function to find out some quick information.*

Answer 7.2a:

_____

_____

Run the following code in R to create a graphical representation of the data:

```
# Create a barplot
barplot(t(matrix(c(sales$historical, sales$newstore), ncol = 2)),
        beside = TRUE, names.arg = sales$month, las = 1, xlab = 'Month',
        ylab = 'Percentage of yearly sales',
        main = 'Seasonal sales',
        col = c('aquamarine3','coral'))

# Add a legend
legend('topleft', bty = 'n',
       legend = c('Historical', 'New store'),
       fill = c('aquamarine3','coral'))
```

The company wants to perform a **Chi-square test** , with 90% confidence, on these data using the historical sales as the baseline values and the new store sales as the observed values.

7.2 b) Looking at the graph and considering the values in each month, can you perform a Chi-square test on these data?

Answer 7.2b:

7.2 c) Formulate the **null hypothesis** H_0 and the **alternative hypothesis** H_1 for this test in words.

Answer 7.2c:

H_0 :

H_1 :

7.2 d) Perform a **Chi-square test** in R using these data and draw the conclusion for the hypotheses. Include the following elements:

- ☐ Show how the calculated **Chi-square value** relates to the **critical value** .
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about the historical and the new **distribution** .
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

R code 7.2d:

Answer 7.2d:

Assignment 7.3: Proportion testing by hand and in R

The P.H.O.N.E. company sells subscriptions to magazines by phone. The commercial director wants to know what **proportion** of calls actually lead to a subscription and whether that depends on the time of day. He therefore takes two samples (n_1 and n_2), one in the afternoon shift and one in the evening shift, and records the number of calls that lead to a subscription (k_1 and k_2):

Afternoon: $n_1 = 71, k_1 = 8$ Evening: $n_1 = 111, k_1 = 16$

7.3 a) What is the best estimate for the **population proportions** π_1 and π_2?

Answer 7.3a:

π_1: _____ π_2: _____

7.3 b) Calculate the two-sided 95% **confidence interval** for the population proportion π for both samples.

*Hint 7.3: You can find the formula for a **confidence interval** for a **proportion** in the formula sheet on page 111. Use $z = 1.960$ (see also Table 4 on page 116).*

Answer 7.3b:

`Confidence interval sample 1:` _____

`Confidence interval sample 2:` _____

The success rate appears to be higher in the evening than in the afternoon. Unfortunately you cannot deduce from these estimated intervals for the population proportions whether this difference is significant. However, you can do this using a two-sample **z-test** for a **proportion** .

7.3 c) Write down the **null hypothesis** H_0 and **alternative hypothesis** H_1 for a test to find out of the evening success rate is higher than the afternoon success rate.

Answer 7.3c:

H_0: _____ H_1: _____

7.3 d) Calculate the **combined success probability** for both samples together.

Hint 7.4: You can find the formulas for the following assignments in the formula sheet on page 111.

Answer 7.3d:

Combined success probability: _____

7.3 e) Calculate the **combined standard error** for the two proportion z-test.

Answer 7.3e:

Combined standard error: _____

7.3 f) Calculate the **z-score** for the two proportion z-test.

Answer 7.3f:

z-score: _____

7.3 g) Drawn the conclusion based on the **z-score** using a **critical z-value** of 1.645 for a one-sided proportion test with 95% confidence. Include the following four elements:

- ☐ Show how the calculated **z-value** relates to the **critical z-value** .
- ☐ Discuss whether H_0 is rejected or not.
- ☐ Describe what this tells us about π_1 and π_2.
- ☐ Describe what type of error is relevant *(type-I or type-II)*.

Answer 7.3g:

_____

_____

_____

_____

_____

_____

The commercial director does not believe your result and wants you to recalculate it in R.

Run the following code in R:

```
n <- c(71, 111)
k <- c(8, 16)
prop.test(x = k, n = n)
```

7.3 h) Interpret the results and compare the outcome with your answer for assignment 7.3g. Do your results match?

Answer 7.3h:

Chapter 8: Bayesian statistics

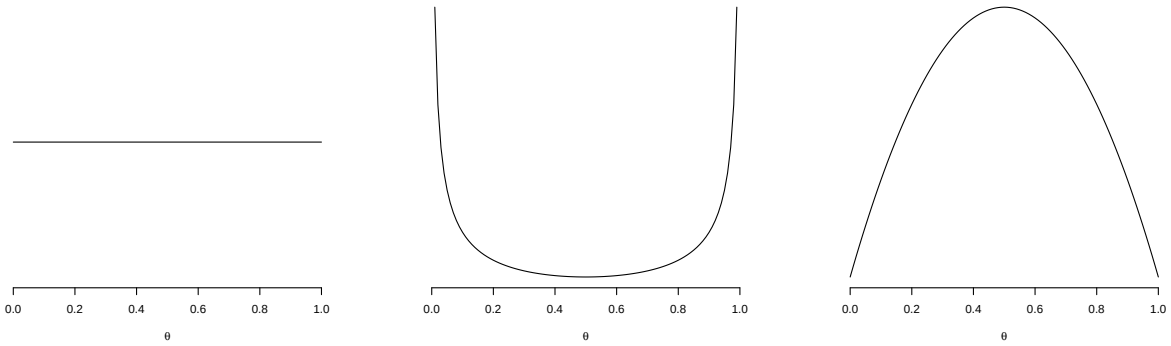
Learning objectives of this chapter:

- Specifying and interpreting a prior distribution
- Updating a prior distribution to a posterior distribution
- Quantifying evidence for a model using the Bayes factor

Assignment 8.1: Specifying and interpreting a prior distribution

In this final chapter, we stray away from the traditional statistical methods that you have learned (**confidence intervals** and **p-values**), and introduce a different methodology; Bayesian statistics. Bayesian statistics expresses the most basic idea in learning, namely, that your prior belief in an event is updated after observing data to a posterior belief. By first specifying your prior beliefs, you allow yourself to learn from the data that you have collected, thereby updating your prior belief to a more informed one. Bayesian statistics is a powerful tool and can help you to update your existing beliefs in the context of new business information.

In Bayesian inference your prior beliefs about a **parameter** are expressed through a **probability distribution** . For example, in tossing a coin, the probability of heads occurring may be defined as a **parameter** denoted by θ. Your initial probability distribution (i.e., beliefs) on θ is called the **prior distribution** and can be interpreted quite visually. The figure below shows three possible prior distributions for the probability of heads in a coin toss. The area under the curve represents probability.



8.1 a) Interpret the three **prior distributions** above and discuss what information about the probability of the coin occurring on heads θ they incorporate.

Answer 8.1a:

When estimating a proportion (like the probability of head in a coin toss) you can use the common $Beta(\alpha, \beta)$ distribution as a prior distribution on θ since it is restricted to be within the range of zero to one. The $Beta(\alpha, \beta)$ distribution has two parameters, α and β, that have an effect on its shape. By playing around with these values, you can create the **prior distribution** that reflects your beliefs as accurately as possible.

Run the following code in R to create the **prior distribution** that is displayed on the left of the figure:

```
alpha <- 1
beta <- 1
curve(dbeta(x, alpha, beta), xlab = expression(theta), ylab = '', yaxt = 'n')
```

8.1 b) Recreate the middle and right **prior distributions** by changing the values for **alpha** and **beta** and running the code again. What are the values of α and β for these distributions?

R code 8.1b:

Answer 8.1b:

Middle: Right

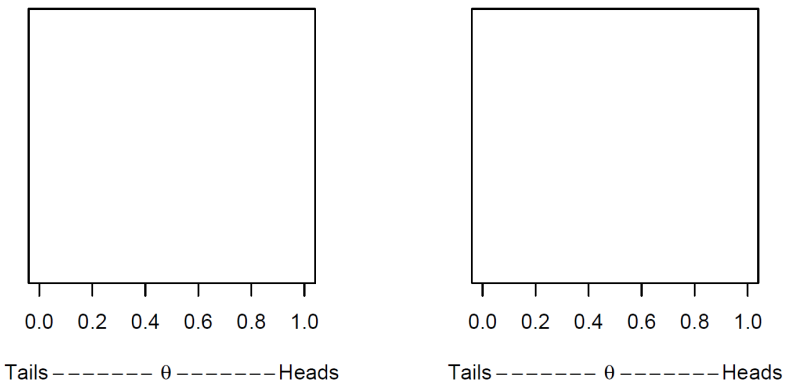
α: _____ α: _____

β: _____ β: _____

8.1 c) How would you draw a **prior distribution** for θ if you believed the coin was definitely biased towards heads? And for one that is biased towards tails? Draw your **prior distributions** below.

Answer 8.1c:

Biased towards heads: Biased towards tails:



Assignment 8.2: Updating a prior distribution to a posterior distribution

Suppose you are the manager of a small store and want to estimate what proportion of your customers leaves the store with a feeling of satisfaction. You give out a questionnaire to 40 people in the store and ask them whether they felt satisfied or not. In a previous enquiry done by you, it was already shown that, on average, 60% of your customers feels good about the store, with a **standard deviation** of ten percent. Using Bayesian statistics, you are now going to update the information from your previous enquiry with the information from the new questionnaire.

8.2 a) What **parameter** does θ represent in this scenario?

Hint 8.1: Think about what question you are interested in answering.

8.2 b) Think of your own α and β parameters for the $Beta(\alpha, \beta)$ **prior distribution** on θ. Take into account the average percentage from your previous enquiry and incorporate this into your **prior distribution.**

Answer 8.2b:

α: _____ β: _____

8.2 c) Use the **curve()** function to create a figure of your **prior distribution** in R.

R code 8.2c:

By updating the $Beta(\alpha, \beta)$ prior distribution with information that you have collected, you create a **posterior distribution** , which contains both the prior information and the information from the sample. Having observed k successes in a sample of n observations, the **prior parameters** α and β are updated to the **posterior parameters** $\alpha + k$ and $\beta + n - k$.

In your sample of $n = 40$ questioned customers, $k = 33$ said they felt satisfied when leaving the store.

8.2 d) Write down the **parameters** of the **posterior distribution** .

Answer 8.2d:

α: _____ β: _____

Run the following code in R to create a figure of the **posterior distribution** . Fill in your own values of **alpha** and **beta** from assignment 8.2b.

```
alpha <- 7
beta <- 5
n <- 40
k <- 33

curve(dbeta(x, alpha + k, beta + n - k),
      xlab = expression(theta), ylab = '', yaxt = 'n')
```

8.2 e) Find out the **posterior probability** that more than 60 percent of your customers left the store feeling satisfied, given the prior information and the information from the sample.

Hint 8.2: Use the `qbeta()` *function to find the cumulative probability under a beta distribution.*

R code 8.2e:

Answer 8.2e:

Probability: _____

8.2 f) Find out the **posterior probability** that the percentage of customers that left the store feeling satisfied, given the prior information and the information from the sample, lies between 70 and 90 percent.

R code 8.2f:

Answer 8.2f:

Probability: _____

8.2 g) Change the values of `alpha` and `beta` in the code above so that you have a different **prior distribution** and run the code again. Describe how robust the **posterior distribution** is to changes in the prior distribution.

Answer 8.2g:

_____

_____

Assignment 8.3: Finding the posterior distribution using MCMC sampling

In assignment 8.2 you have seen that updating a **prior distribution** to a **posterior distribution** is easy when you are counting successes and failures. That is because the **prior distribution** and **posterior distribution** are in the same family, they are both $Beta$ distributions. However, many prior distributions are not so easy to find when updated by the data. In this assignment, you are going to use a Markov chain Monte Carlo sampling method to approximate the posterior distribution.

For this exercise you will be using the Stan coding language with the **rstan** R package. **rstan** performs MCMC sampling using an Hamiltonian Monte Carlo (HMC) algorithm, which is known to have a very high quality. Install and load the package using:

```
install.packages(rstan)
library(rstan)
```

Stan is very specific. It requires that you specify all **data** , all **parameters** , and the complete statistical **model** [3].

Copy and run the following code in R to set up the Stan model for the scenario in exercise 8.2b and store it in an object called **modelCode** . The code then compiles the model using the **stan_model()** function and starts sampling using the **sampling()** function. The result of the sampling is stored in the **stanFit** object.

Hint 8.3: You can change the **theta ~ beta(7, 5)** *part of the code to specify your own prior distribution from assignment 8.2b.*

```
modelCode <- '
data {
     int n;
     int k;
}
parameters {
     real<lower=0,upper=1> theta;
}
model {
     theta ~ beta(7, 5);
     k ~ binomial(n, theta);
}
'

# Note: The following line can take a while to execute
compiledModel <- stan_model(model_code = modelCode, model_name = 'model')

stanFit <- sampling(compiledModel, data = list(n = 40, k = 33),
                    iter = 5000, warmup = 500, chains = 4)
```

[3] For a complete introduction to Stan, you can visit https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started

8.3 a) Extract the samples of the **posterior distribution** on θ from the **stanFit** object using the **extract()** function and store them in an object called **samples** .

R code 8.3a:

8.3 b) Create a histogram of the **samples** . Next, run the following code in R to plot the analytical **posterior distribution** over the histogram. Do the samples of the **posterior distribution** resemble the actual **posterior distribution** in any way?

```
curve(dbeta(x, shape1 = 40, shape2 = 12), add = TRUE)
```

Hint 8.4: Be sure to specify a larger value for the **breaks** *argument since you have many samples. Also set* **probability = TRUE** *in the* **hist()** *function.*

R code 8.3b:

Answer 8.3b:

As you have seen in assignment 8.2, this **posterior distribution** can be calculated by hand, since it is a $Beta$ distribution. To represent your prior information (remember: you have conducted a questionnaire that showed you 60 percent of customers left your store satisfied), let's give θ a $Normal(\mu, \sigma)$ distribution as a prior distribution. You use your prior estimate of θ as the **mean** of the prior distribution, and the **standard deviation** of the sample ($s = 0.10$) as the **standard deviation** of the **prior distribution** .

8.3 c) Change the prior distribution in the **modelCode** to the $Normal$ distribution described in the text above. Run all appropriate code again.

Hint 8.5: The line **theta ~ beta(7, 5)** *specifies the prior distribution.*

R code 8.3c:

8.3 d) Again, find out the **posterior probability** that more than 60 percent of your customers left the store feeling satisfied, given the prior information and the information from the sample.

Hint 8.6: Do not use the **quantile()** *function or the* **pbeta()** *function.*

R code 8.3d:

Answer 8.3d:

Probability: _____

8.3 e) Find out the **posterior probability** that the percentage of customers that left the store feeling satisfied, given the prior information and the information from the sample, lies between 70 and 90 percent.

R code 8.3e:

Answer 8.3e:

Probability: _____

8.3 f) Do your answers for assignment 8.3d and assignment 8.3e differ substantially from your answers at assignment 8.2e and assignment 8.2f. What does this tell you about the robustness of your outcomes to the choice of **prior distribution** ?

Answer 8.3f:

Assignment 8.4: Comparing simple models using the Bayes factor

Bayesian statistics does not use the **p-value** to test hypotheses and yield conclusions in an all-or-none fashion. Instead it uses a continuous measure of evidence, the **Bayes factor**. The **Bayes factor** quantifies the relative predictive performance of two competing hypotheses; the **null hypothesis** H_0 and the **alternative hypothesis** H_1. The subscript in the **Bayes factor** indicates for which hypothesis supported is quantified. BF_{10} indicates the Bayes factor in favor of H_1 over H_0, whereas BF_{01} indicates the **Bayes factor** in favor of H_0 over H_1. Larger values of BF_{10} indicate more support for H_1. A **Bayes factor** can range from 0 to ∞, and a Bayes factor of 1 indicates that both hypotheses are supported equally well by the data. A $BF_{10} = 10$ indicates that the data is 10 times more likely to occur under H_1 than under H_0. This way, the Bayes factor is a continuous measure for the strength of evidence.

Suppose you are a data scientist at a stock trading company. You have written an algorithm that trades stocks automatically. Your algorithm either makes a profitable trade, or a losing trade. You want to make sure that your algorithm actually does something intellectual, and want to disprove the hypothesis that your algorithm makes a profitable trade with a probability other than if it were to trade stocks randomly.

8.4 a) Specify the **null hypothesis** H_0 and the **alternative hypothesis** H_1 for the test of a proportion against a chance value.

Answer 8.4a:

H_0: _____ H_1: _____

8.4 b) Specify a $Beta$ prior distribution for the **parameter** θ. Let your prior distribution depend on how much confidence you have in your programming skills.

Hint 8.7: Think about what the θ represents in this case.

Answer 8.4b:

$\theta \sim Beta($ _____ , _____ $)$

You monitor the algorithm for 30 trades, and each time mark a trade as a success (profit) or a failure (loss). After data collection you see that, of the 30 trades, your algorithm made 22 profitable trades.

8.4 c) What are the α and β parameters of your **posterior distribution**?

Answer 8.4c:

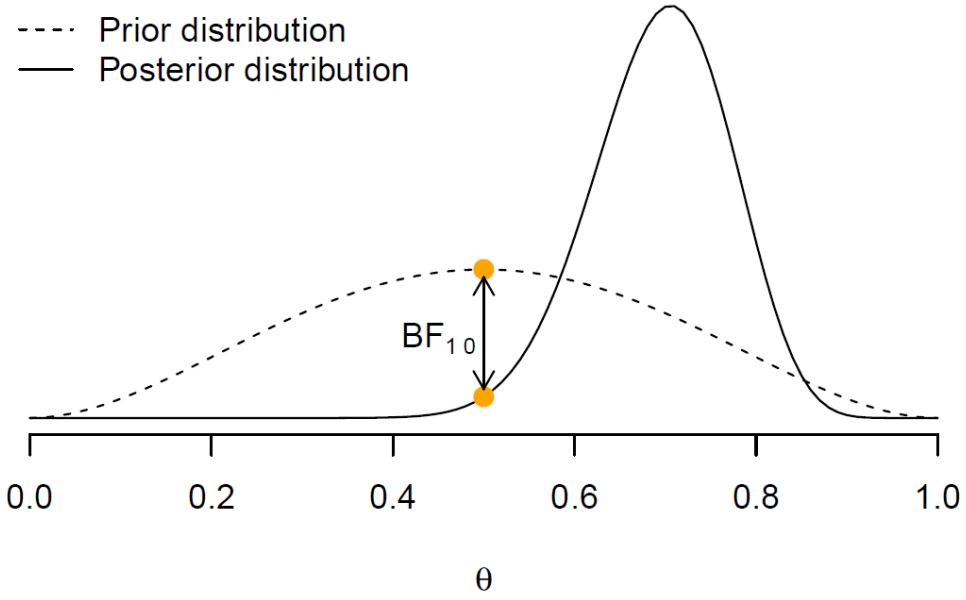
α: _____ β: _____

8.4 d) Use the `curve()` function to create a figure that has both the prior and the posterior distribution in it.

Hint 8.8: The `curve()` *function has an argument* `add` *that controls whether to add a line to an already existing plot.*

In this scenario, the **Bayes factor** is intuitively to calculate by only looking at the posterior distribution for H_1. The **Bayes factor** BF_{10} tells you how much more likely the **alternative hypothesis** H_1 is in comparison with to the **null hypothesis** H_0. The **Bayes factor** BF_{10} in favor of the unrestricted **alternative hypothesis** $H_1 : \theta \neq 0.5$ is simply the height of the **posterior distribution** at the point $\theta = 0.5$ divided by the height of the **prior distribution** at the point $\theta = 0.5$.

Please note that this is a trick that only works for **nested** hypotheses and models, in which only one of the **parameters** is fixed to a specific value.



8.4 e) Use the `dbeta()` function to find out the height of your **prior distribution** at the point $\theta = 0.5$ and store it in an object called `heightPrior` .

R code 8.4e:

```
```

8.4 f) Use the `dbeta()` function to find out the height of your **posterior distribution** at the point $\theta = 0.5$ and store it in an object called `heightPosterior` .

R code 8.4f:

```
```

Run the following code in R to compute the **Bayes factor** in favor of the **alternative hypothesis** H_1.

```
BF10 <- heightPrior / heightPosterior
```

8.4 g) What is the value of the **Bayes factor** `BF10` ? Interpret this **Bayes factor** with respect to the alternative hypothesis H_1.

Answer 8.4g:

To make interpretation of the **Bayes factor** more simple, the labels in the table below have been proposed. Please note that you shout not get too hung up on the specific labels and breakpoints in this table, since convincing evidence is different for every situation.

Bayes factor	Evidence
1 - 3	Anecdotal
3 - 10	Moderate
10 - 30	Strong
30 - 100	Very strong
$>$ 100	Extreme

8.4 h) Considering the labels from the table above, can you convincingly disprove the hypothesis that your algorithm makes a profitable trade with a probability other than if it were to trade stocks randomly?

Answer 8.4h:

8.4 i) Re-specify the α and β parameters of the **prior distribution** so that you express a different prior belief. Recalculate the **Bayes factor** by running your code again. What is the new value of `BF10` . How robust in your **Bayes factor** to changes in the prior distribution?

Answer 8.4i:

Assignment 8.5: Using bridge sampling to calculate the Bayes factor

You are not convinced by the evidence from assignment 8.4 and want to perform additional testing. This time, however, you want to find out whether your algorithm makes more than 75 percent of the trades with profit. That is, you want to test whether θ is higher or lower than 0.75.

8.5 a) Formulate the **models** M_1 (restricts θ to be below 0.75) and M_2 (restricts θ to be higher than 0.75). Choose the restriction ($\leq$, $=$, or $\geq$) that applies to each model.

Answer 8.5a:

M_1: $\theta \leq / = / \geq$ _____ M_2: $\theta \leq / = / \geq$ _____

For these more complicated (e.g., non-nested) models, the computation of the **Bayes factor** becomes more difficult since it involves computing the **marginal likelihoods** of the **models**. To make your life simple, you can use the **bridgesampling** package together with the **rstan** package to let R calculate these values for you.

Run the following code in R to install the **bridgesampling** package. Then load the **bridgesampling** and the **rstan** package.

```
install.packages('bridgesampling')

library(bridgesampling)
library(rstan)
```

Copy and run the following code in R to set up the Stan model for M_1, the model that restricts θ to be lower than 0.75 and store it in an object called **model1**. The code then compiles the model using the **stan_model()** function.

```
model1code <- '
data {
     int n;
     int k;
}
parameters {
     real<lower=0,upper=0.75> theta;
}
model {
     theta ~ beta(1, 1)T[0, 0.75];
     k ~ binomial(n, theta);
}
'

# Note: The following line can take a while to execute
model1 <- stan_model(model_code = model1code, model_name = 'model1')
```

8.5 b) Create the Stan model for M_2, the model that restricts θ to be higher than 0.75, on the basis of the lines in `model1code` . Name the compiled model `model2` .

Hint 8.9: First find out what elements were added to the code since assignment 8.3.

R code 8.5b:

8.5 c) Locate and describe the **prior distributions** for θ in M_1 and M_2.

Answer 8.5c:

Now you start collecting data again. Suppose that you monitor your algorithm for $n = 156$ more trades, and find that $k = 123$ trades resulted in a profit.

8.5 d) Use the `sampling()` function to sample for the **models** M_1 and M_2 using these data. Store the fitted Stan models in objects named `stanFitM1` and `stanFitM2` .

R code 8.5d:

8.5 e) Use the `bridge_sampler()` function to calculate the **marginal likelihoods** of the models M_1 and M_2 and store them in `mLike1` and `mLike2` .

R code 8.5e:

8.5 f) Use the `bf()` function to calculate the **Bayes factor** BF_{21} in favor of model M_2 over model M_1 and store it in an object called `BF21` .

Hint 8.10: Be sure to insert the two marginal likelihoods into the `bf()` *function in the correct order so that the result is* BF_{21}.

8.5 g) What is the value of `BF21` ? Interpret this **Bayes factor** with respect to the model that restricts θ to be higher than 0.75.

Answer 8.5g:

8.5 h) What is the strength of evidence associated with this **Bayes factor** ? Are you convinced by this evidence?

Answer 8.5h:

Assignment 8.6: Performing a Bayesian linear regression

Suppose you work as an analyst for an insurance firm. Your firm wants to assess which variables can predict the height of of a customer insurance charges. For this assignment you will need the **insurance.csv** [4] from the online resources. The data set contains a sample of 1338 health insurance customer and their recorded variables. For the current assignment, you can focus on the columns **charges** (the total charges of the customer), **age** (the customer age), **bmi** (the customer BMI score), and **neighbors** (the number of neighbors of the customer) columns.

8.6 a) Use the **read.csv()** function (and **setwd()** function if you prefer) to import the data set into an object named **insurance** .

R code 8.6a:

```
```

8.6 b) Write down the regression equation for a **linear model** where you predict the variable **charges** on the basis of the predictor variables **age** , **bmi** , and **neighbors** .

Answer 8.6b:

charges = _____

Run the following code in R to create a Stan model M_1 that corresponds to this linear model with all **coefficients** . Note that you do not specify **prior distributions** (under the **model** section) for the β_0, β_1, and β_2 parameters. That is because, when you do not specify the prior distributions in **Stan** , the posterior distributions will be estimated solely from the data.

[4] These data are adapted from the original open-source data set which can be found at: https://www.kaggle.com/mirichoi0218/insurance/data.

```
regmodelcode <- '
data {
     int<lower=0> N;
     vector[N] x1;
     vector[N] x2;
     vector[N] x3;
     vector[N] y;
}
parameters {
     real beta0;
     real beta1;
     real beta2;
     real beta3;
     real<lower=0> sigma;
}
model {
     y ~ normal(beta0 + beta1 * x1 + beta2 * x2 + beta3 * x3, sigma);
}
'

# Note: The following line can take a while to execute
regressionModel <- stan_model(model_code = regmodelcode, model_name = 'Regression')
```

8.6 c) Use the `sampling()` function to sample from the `regressionModel` model and store the samples in an object named `regressionSamples`. Next, use the `summary()` function to find out the **posterior means** of the β_0, β_1, β_2, and β_3 coefficients.

Answer 8.6c:

β_0: _____ β_1: _____ β_2: _____ β_3: _____

R code 8.6c:

8.6 d) Use the `lm()` function to fit a regular **linear model** to the data. Use the `summary()` function to find out the estimates of the coefficients when done this way. Do they match the **posterior means** from assignment 8.6b?

R code 8.6d:

Answer 8.6d:

YES / NO

Run the following code in R to create a figure of the posterior distribution of β_1, the regression coefficient for the variable **age** :

```
plot(density(regressionSamples$beta1), xlab = '', ylab = '',
     yaxt = 'n', bty = 'n', main = expression(beta[1]))
```

8.6 e) Also create figures of the posterior distributions for β_0, β_2, and β_3.

R code 8.6e:

8.6 f) Find out the probability that the average increase in **charges** as a function of 1 point on the **bmi** scale is higher than 200 dollars.

Hint 8.11: Use the approach that you applied in assignment 8.3d and assignment 8.3e.

R code 8.6f:

Answer 8.6f:

Probability: _____

8.6 g) Find out the probability that the value of β_1 is lower than 250.

R code 8.6g:

Answer 8.6g:

Probability: _____

8.6 h) Find out between which values of β_3 90 percent of the **posterior samples** lie.

R code 8.6h:

Answer 8.6h:

`Lower bound:` _____ `Upper bound:` _____

As you can see, the posterior distribution for the β_3 coefficient centers around zero, and it's contribution to the accuracy of the prediction can be questioned. To test how much more likely a model is that includes the β_3 predicts the data compared to a model that excludes the coefficient, you can use the procedure that you have learned in assignment 8.5.

8.6 i) Write code for a new `Stan` model M_2 where you remove the predictor variable **neighbors** .

R code 8.6i:

8.6 j) Using the procedure in assignment 8.5, calculate the **Bayes factor** BF_{12} in favor of M_1 (the model that includes β_3) over M_2 (the model that excludes β_3).

R code 8.6j:

8.6 k) What is the value of BF_{12}. Interpret this **Bayes factor** with respect to the coefficient β_0. Incorporate the strength of the evidence in your answer.

Answer 8.6k:

Keep in mind that basically all statistical tests are a specific form of regression (like you have learned in chapter 6). This means that you can extend the Stan models that you have seen in this chapter to answer any question that you want to investigate.

Formula sheet and tables

Central tendency

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

Median (Q_2) $\frac{(n+2)}{2} th\, number$

Lower quartile (Q_1) $\frac{(n+2)}{4} th\, number$

Upper quartile (Q_1) $\frac{(n+2)}{4} \times 3\, th\, number$

Range $max_x - min_x$

Interquartile range $Q_3 - Q_1$

Dispersion

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

Standard deviation $s = \sqrt{s^2}$

Hartley's F $F = \frac{var_{max}}{var_{min}} = \frac{s^2_{max}}{s^2_{min}}$

Confidence intervals

Standard error $SE_\mu = \frac{s}{\sqrt{n}}$
(of the sample mean)

Confidence interval $\bar{x} \pm z_\alpha \times SE_\mu$
(for the population mean with $n \geq 30$)

Confidence interval $\bar{x} \pm t_\alpha \times SE_\mu$
(for the population mean with $n < 30$)

Confidence interval $p \pm t_\alpha \times \sqrt{\frac{p(1-p)}{n}}$
(for the population proportion)

Tests based on the normal distribution

z-score $z = \frac{x-\mu}{\sigma}$
(for a value x in a normal distribution)

z-score $z = \frac{\bar{x}-\mu}{s/\sqrt{n}}$
(for a test of a population mean based on a sample)

Tests based on the t-distribution

t-score: $t = \frac{\bar{x}-\mu}{s/\sqrt{n}}$
(for a test of a population mean based on sample $n < 30$)

Degrees of freedom: $df = n - 1$
(for the t-distribution in a one sample t-test)

t-score: $t = \frac{(\bar{x_1} - \bar{x_2}) - D_0}{\sqrt{s_p^2(\frac{1}{n_1} + \frac{1}{n_2})}}$
(for an independent samples t-test)

Standard error: $s_p^2 = \frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}$
(for an independent samples t-test)

Degrees of freedom: $df = n_1 + n_2 - 2$
(for an independent samples t-test)

t-score: $t = \frac{\bar{D}-\mu_0}{s_D/\sqrt{n}}$
(for a dependent samples t-test)

Degrees of freedom: $df = n - 1$
(for a dependent samples t-test)

Correlation

Covariance $s_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$

Correlation $r_{xy} = \frac{s_{xy}}{s_x \times s_y}$

Sampling distribution: $z_r = \frac{1}{2} \times log_e(\frac{1+r}{1-r})$
(for the population correlation)

Standard error: $SE_r = \frac{1}{\sqrt{n-3}}$
(of the population correlation)

z-score: $z_{xy} = \frac{z_r}{SE_r}$
(for the test of a correlation against any value)

Degrees of freedom: $df = n - 2$
(for the t-distribution in correlation test against zero)

t-score: $t_{xy} = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}$
(for the test of a correlation against zero)

Regression and AN(C)OVA

R-squared $R^2 = \frac{SS_M}{SS_R}$
(of a regression or AN(C)OVA)

F-score $F = \frac{MS_M}{MS_R}$
(of a regression or AN(C)OVA)

Degrees of freedom $df_M = k - 1$
(for the AN(C)OVA model)

Degrees of freedom $df_R = n - k$
(for the AN(C)OVA residuals)

Mean squared error $MS_M = \frac{SS_M}{df_M}$
(for the AN(C)OVA model)

Mean squared error $MS_R = \frac{SS_R}{df_R}$
(for the AN(C)OVA residuals)

Proportions

Combined success probability $p^* = \frac{k_1+k_2}{n_1+n}$
(for the outcomes of two samples)

Standard error $s_p = \sqrt{p^*(1-p^*)(\frac{1}{n_1} + \frac{1}{n_2})}$
(for the proportion of two samples)

z-score $z = \frac{p_1-p_2}{s_p}$
(for a test comparing two proportions)

Expected value $E = p \times n$
(for a proportion)

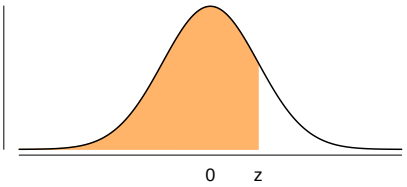
Chi-square score $X^2 = \sum_{i=1}^{k} \frac{(O-E)^2}{E}$
(for a test comparing k proportions)

Table 1: Critical values for Hartley's F (F_{max})

Level of significance $\alpha = 0.05$

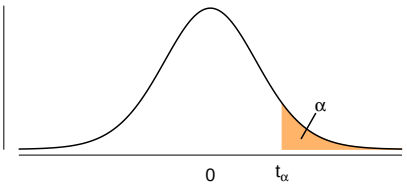
	Number of variances to compare (k)										
$n-1$	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
2	39.0	87.5	142	202	266	333	403	475	550	626	704
3	15.4	27.8	39.2	50.7	62.0	72.9	83.5	93.9	104	114	124
4	9.6	15.5	20.6	25.2	29.5	33.6	37.5	41.1	44.6	48.0	51.4
5	7.15	10.8	13.7	16.3	18.7	20.8	22.9	24.7	26.5	28.2	29.9
6	5.82	8.38	10.4	12.1	13.7	15.0	16.3	17.5	18.6	19.7	20.7
7	4.99	6.94	8.44	9.70	10.8	11.8	12.7	13.5	14.3	15.1	15.8
8	4.43	6.00	7.18	8.12	9.03	9.78	10.5	11.1	11.7	12.2	12.7
9	4.03	5.34	6.31	7.11	7.80	8.41	8.95	9.45	9.91	10.3	10.7
10	3.72	4.85	5.67	6.34	6.92	7.42	7.87	8.28	8.66	9.01	9.34
12	3.28	4.16	4.79	5.30	5.72	6.09	6.42	6.72	7.00	7.25	7.48
15	2.86	3.54	4.01	4.37	4.68	4.95	5.19	5.40	5.59	5.77	5.93
20	2.46	2.95	3.29	3.54	3.76	3.94	4.10	4.24	4.37	4.49	4.59
30	2.07	2.40	2.61	2.78	2.91	3.02	3.12	3.21	3.29	3.36	3.39
60	1.67	1.85	1.96	2.04	2.11	2.17	2.22	2.26	2.30	2.33	2.36
∞	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 2: z - values for significance level α



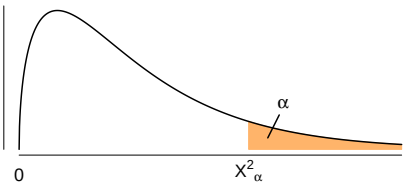
z	**0.00**	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7518	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.99865	0.99869	0.99874	0.99878	0.99882	0.99886	0.99889	0.99893	0.99897	0.99900
3.1	0.99903	0.99906	0.99910	0.99913	0.99916	0.99918	0.99921	0.99924	0.99926	0.99929
3.2	0.99931	0.99934	0.99936	0.99938	0.99940	0.99942	0.99944	0.99946	0.99948	0.99950
3.3	0.99952	0.99953	0.99955	0.99957	0.99958	0.99960	0.99961	0.99962	0.99964	0.99965
3.4	0.99966	0.99968	0.99969	0.99970	0.99971	0.99972	0.99973	0.99974	0.99975	0.99976
3.5	0.99977	0.99978	0.99978	0.99979	0.99980	0.99981	0.99981	0.99982	0.99983	0.99983
3.6	0.99984	0.99985	0.99985	0.99986	0.99986	0.99987	0.99987	0.99988	0.99988	0.99989
3.7	0.99989	0.99990	0.99990	0.99990	0.99991	0.99991	0.99992	0.99992	0.99992	0.99992
3.8	0.99993	0.99993	0.99993	0.99994	0.99994	0.99994	0.99994	0.99995	0.99995	0.99995

Table 3: t - values for significance level α



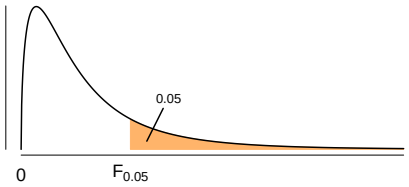
df_{n-1}	$t_{.100}$	$t_{.05}$	$t_{.025}$	$t_{.01}$	$t_{.005}$	$t_{.001}$	$t_{.0005}$
1	3.078	6.314	12.706	31.821	63.657	318.309	636.619
2	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	1.310	1.697	2.042	2.457	2.750	3.385	3.646
31	1.309	1.696	2.040	2.453	2.744	3.375	3.633
32	1.309	1.694	2.037	2.449	2.738	3.365	3.622
33	1.308	1.692	2.035	2.445	2.733	3.356	3.611
34	1.307	1.691	2.032	2.441	2.728	3.348	3.601
35	1.306	1.690	2.030	2.438	2.724	3.340	3.591
36	1.306	1.688	2.028	2.434	2.719	3.333	3.582
37	1.305	1.687	2.026	2.431	2.715	3.326	3.574
38	1.304	1.686	2.024	2.429	2.712	3.319	3.566
39	1.304	1.685	2.023	2.426	2.708	3.313	3.558
40	1.303	1.684	2.021	2.423	2.704	3.307	3.551

Table 4: X^2 ***- values for significance level*** α



df	$X^2_{.100}$	$X^2_{.05}$	$X^2_{.025}$	$X^2_{.01}$	$X^2_{.001}$
1	2.706	3.841	5.024	6.635	10.828
2	4.605	5.991	7.378	9.210	13.816
3	6.251	7.815	9.348	11.345	16.266
4	7.779	9.488	11.143	13.277	18.467
5	9.236	11.070	12.833	15.086	20.515
6	10.645	12.592	14.449	16.812	22.458
7	12.017	14.067	16.013	18.475	24.322
8	13.362	15.507	17.535	20.090	26.125
9	14.684	16.919	19.023	21.666	27.877
10	15.987	18.307	20.483	23.209	29.588
11	17.275	19.675	21.920	24.725	31.264
12	18.549	21.026	23.337	26.217	32.910
13	19.812	22.362	24.736	27.688	34.528
14	21.064	23.685	26.119	29.141	36.123
15	22.307	24.996	27.488	30.578	37.697
16	23.542	26.296	28.845	32.000	39.252
17	24.769	27.587	30.191	33.409	40.790
18	25.989	28.869	31.526	34.805	42.312
19	27.204	30.144	32.852	36.191	43.820
20	28.412	31.410	34.170	37.566	45.315
21	29.615	32.671	35.479	38.932	46.797
22	30.813	33.924	36.781	40.289	48.268
23	32.007	35.172	38.076	41.638	49.728
24	33.196	36.415	39.364	42.980	51.179
25	34.382	37.652	40.646	44.314	52.620
26	35.563	38.885	41.923	45.642	54.052
27	36.741	40.113	43.195	46.963	55.476
28	37.916	41.337	44.461	48.278	56.892
29	39.087	42.557	45.722	49.588	58.301
30	40.256	43.773	46.979	50.892	59.703
31	41.422	44.985	48.232	52.191	61.098
32	42.585	46.194	49.480	53.486	62.487
33	43.745	47.400	50.725	54.776	63.870
34	44.903	48.602	51.966	56.061	65.247
35	46.059	49.802	53.203	57.342	66.619
36	47.212	50.998	54.437	58.619	67.985
37	48.363	52.192	55.668	59.893	69.347
38	49.513	53.384	56.896	61.162	70.703
39	50.660	54.572	58.120	62.428	72.055
40	51.805	55.758	59.342	63.691	73.402

Table 5: F - values for significance level 0.05



df_R \ df_M	1	2	3	4	5	6	7	8	9
1	161.4476	199.5000	215.7073	224.5832	230.1619	233.9860	236.7684	238.8827	240.5433
2	18.5128	19.0000	19.1643	19.2468	19.2964	19.3295	19.3532	19.3710	19.3848
3	10.1280	9.5521	9.2766	9.1172	9.0135	8.9406	8.8867	8.8452	8.8123
4	7.7086	6.9443	6.5914	6.3882	6.2561	6.1631	6.0942	6.0410	5.9988
5	6.6079	5.7861	5.4095	5.1922	5.0503	4.9503	4.8759	4.8183	4.7725
6	5.9874	5.1433	4.7571	4.5337	4.3874	4.2839	4.2067	4.1468	4.0990
7	5.5914	4.7374	4.3468	4.1203	3.9715	3.8660	3.7870	3.7257	3.6767
8	5.3177	4.4590	4.0662	3.8379	3.6875	3.5806	3.5005	3.4381	3.3881
9	5.1174	4.2565	3.8625	3.6331	3.4817	3.3738	3.2927	3.2296	3.1789
10	4.9646	4.1028	3.7083	3.4780	3.3258	3.2172	3.1355	3.0717	3.0204
11	4.8443	3.9823	3.5874	3.3567	3.2039	3.0946	3.0123	2.9480	2.8962
12	4.7472	3.8853	3.4903	3.2592	3.1059	2.9961	2.9134	2.8486	2.7964
13	4.6672	3.8056	3.4105	3.1791	3.0254	2.9153	2.8321	2.7669	2.7144
14	4.6001	3.7389	3.3439	3.1122	2.9582	2.8477	2.7642	2.6987	2.6458
15	4.5431	3.6823	3.2874	3.0556	2.9013	2.7905	2.7066	2.6408	2.5876
16	4.4940	3.6337	3.2389	3.0069	2.8524	2.7413	2.6572	2.5911	2.5377
17	4.4513	3.5915	3.1968	2.9647	2.8100	2.6987	2.6143	2.5480	2.4943
18	4.4139	3.5546	3.1599	2.9277	2.7729	2.6613	2.5767	2.5102	2.4563
19	4.3807	3.5219	3.1274	2.8951	2.7401	2.6283	2.5435	2.4768	2.4227
20	4.3512	3.4928	3.0984	2.8661	2.7109	2.5990	2.5140	2.4471	2.3928
21	4.3248	3.4668	3.0725	2.8401	2.6848	2.5727	2.4876	2.4205	2.3660
22	4.3009	3.4434	3.0491	2.8167	2.6613	2.5491	2.4638	2.3965	2.3419
23	4.2793	3.4221	3.0280	2.7955	2.6400	2.5277	2.4422	2.3748	2.3201
24	4.2597	3.4028	3.0088	2.7763	2.6207	2.5082	2.4226	2.3551	2.3002
25	4.2417	3.3852	2.9912	2.7587	2.6030	2.4904	2.4047	2.3371	2.2821
26	4.2252	3.3690	2.9752	2.7426	2.5868	2.4741	2.3883	2.3205	2.2655
27	4.2100	3.3541	2.9604	2.7278	2.5719	2.4591	2.3732	2.3053	2.2501
28	4.1960	3.3404	2.9467	2.7141	2.5581	2.4453	2.3593	2.2913	2.2360
29	4.1830	3.3277	2.9340	2.7014	2.5454	2.4324	2.3463	2.2783	2.2229
30	4.1709	3.3158	2.9223	2.6896	2.5336	2.4205	2.3343	2.2662	2.2107
40	4.0847	3.2317	2.8387	2.6060	2.4495	2.3359	2.2490	2.1802	2.1240
60	4.0012	3.1504	2.7581	2.5252	2.3683	2.2541	2.1665	2.0970	2.0401
120	3.9201	3.0718	2.6802	2.4472	2.2899	2.1750	2.0868	2.0164	1.9588
∞	3.8415	2.9957	2.6049	2.3719	2.2141	2.0986	2.0096	1.9384	1.8799

R help

Part I: Basic functionality

1.1 Types of data

`numeric`	Numbers
`character`	Words (text)
`logical`	TRUE or FALSE
`factor`	One of the above with predefined categories
`Inf`	Infinite
`NaN`	Not a number
`NA`	Not available

1.2 Assigning values to variables

There are multiple ways to assign a value to a variable. For example, these all do the same, which is assigning the value `1` to the variable `a`.

`a = 1` or `a <- 1` `assign('a', 1)` `a = b = 1`

1.3 Data structures

`vector`	A one-dimensional data structure
`matrix`	A two-dimensional data structure
`array`	A multi-dimensional data structure
`data frame`	A data structure containing different types of data
`list`	A collection of different data structures

1.4 Vectors

`c(1, 2, 3)`	Combine the numbers 1, 2, and 3 in a vector
`seq(1, 6, 2)`	Create sequence from 1 to 6 in increments of 2
`rep(1:3, 2)`	Repeat 1 to 3, and do that 2 times
`1:4`	Vector of 1 t/m 4 (`:` is therefore making a vector)
`paste(x, y)`	Paste vectors `x` and y together
`letters[1:5]`	Vector of first 5 letters of the alphabet
`sample(x, 5)`	Gives a random sample of size 5 from of data `x`
`length(x)`	Indicates the length of `x`
`cut(x, 5)`	Divide `x` in vectors with length 5
`append(x, c(4, 5))`	Add the numbers 4 and 5 to vector `x`
`x <- numeric()`	Creates an empty vector `x`
`sort(x)`	Order vector `x` from low to high (default)

1.5 Matrices

`matrix(1:9, 3, 3)`	Create a matrix of 1 t/m 9 with 3 rows and 3 columns
`diag(x)`	Get the diagonal from matrix `x`
`head(x, 2)`	Gives the first two rows of matrix or data frame `x`
`t(x)`	Gives transpose of matrix `x`
`rbind(x, y)`	Add rows from matrix `x` and `y` together
`cbind(x, y)`	Add columns from matrix `x` and `y` together

1.6 Data frames

`data.frame('X'= x)`	Create a data frame with data `x` (`X` is the column name)
`head(x, 2)`	Look at first two rows of data frame `x`
`names(x)`	Names of data frame `x`
`colnames(x)`	Column names of data frame `x`
`rownames(x)`	Row names of data frame `x`

1.7 Lists

`x <- list()`	Create an empty list `x`
`x[['title']] <- m`	Insert structure `m` into list `x` (`title` is the new title)

1.8 Indexing

`x[2]`	Get the second element from the vector `x`
`x[1:5]`	Get the first to the fifth element from vector `x`
`x[-1]`	Get all elements except first element from vector `x`
`x[x > 5]`	Get all elements greater than 5 from vector `x`
`x[x > 3 & x < 6]`	Get all values from `x` greater than 3 and less than 6
`x[1:3, 1]`	Get first 3 values from the first column from data `x`
`x[1:3, 1:4]`	Get first 3 values from the first four columns from `x`
`x$h or x['h']`	Select element `h` from data frame `x`

1.9 Operators

`x == y`	Check if `x` equals `y`
`x != y`	Check if `x` does not equal `y`
`%%`	The remainder of a division (e.g. `36%%5` $= 1$)

1.10 Importing data and files

`data('x')`	Import data set `x`
`file.choose()`	Get access to interface to select a file
`read.table('x')`	Read in a `.txt` file `x` from the working directory
`write.table(x)`	Writes data `x` to a `.txt` file from the working directory
`read.csv('x')`	Reads in a `.csv` file `x` from the working directory
`write.csv(x)`	Writes data `x` to a `.csv` file from the working directory

1.11 Basic functions

`TAB`	Scrolling through functions beginning with that letter
`ls()`	See all variables available in the environment
`rm(list = ls())`	Delete all variables in your environment
`getwd()`	See the location of your working directory
`setwd()`	Set the location of your working directory
`help(x)`	Read help about the function `x`
`str(x)`	Finds out the structure of data `x`
`summary(x)`	Gives summary of the object `x`
`print('Hello')`	Print `Hello` to the output
`round(x, digits = 2)`	Round number(s) in `x` to a specified number of digits
`which.max(x)`	Indicates the place of the highest value of data `x`
`which(x == 10)`	Shows the place of each object in `x` that equals 10
`unique(x)`	Gives only the unique values in `x`
`length(x)`	Gives the number of elements in `x`
`nrow(x)`	Gives number of rows of matrix/data frame `x`
`ncol(x)`	Gives number of columns of matrix/data frame `x`

1.12 Installing an add-on package

`install.packages('x')`	Installs package with name `x`

Part II: Commands for creating graphics

2.1 Creating a plot

`plot(x, y, ...)`	The basic plot function. Can be extended by adding arguments in `...`
`points(x, y)`	Adds points with `x` and `y` values to an existing plot
`lines(x, y)`	Adds a line with `x` and `y` values to an existing plot
`abline()`	Adds a linear line to plot (see `?abline`)
`text(x, y, 'text')`	Add text to plot on `x` and `y` coordinates
`legend('bottomright')`	Inserts legend in the bottom right of the plot
`pdf('x')`	When called, writes all images to `.pdf` file `x`
`dev.off()`	Ends writing all created images to a file
`layout(matrix(1:6, 2, 3))`	Create the layout (for multiple figures)
`layout(1)`	Put every plot on one page
`curve(dnorm(x, 0, 1), -3, 3)`	Plot the curve of the standard normal distribution

2.2 Quick plot functions

`hist(x, ...)`	Creates a histogram of `x`
`dotchart(x, ...)`	Creates a point chart of `x`
`pairs(x, ...)`	Compares all variables of `x` with multiple plots
`boxplot(x, ...)`	Creates a box plot of `x`
`barplot(x, ...)`	Creates a bar plot of `x`

2.3 Additional plot arguments (to be entered in `...`)

`axes = FALSE`	Disable axis in plot
`add = TRUE`	Adds the created plot to the previous plot
`las = 1`	Rotates the labels on the `x` and `y` axes
`xlim = c(0, 1)`	Set the range of the `x` axis between 0 and 1
`ylim = c(0, 1)`	Set the range of the `y` axis between 0 and 1
`xlab = 'x-axis'`	Set the name of the `x` axis to `x-axis`
`ylab = 'y-axis'`	Set the name of the `y` axis to `y-axis`
`type = 'p'`	Creates a plot containing the data as points
`type = 'l'`	Creates a plot containing the data as line
`type = 'b'`	Creates both points and lines
`lty = 1`	Set the line type in the plot
`col = 'blue'`	Set the color in the plot to `blue`

Part III: If-statements, loops, and functions

3.1 If-statements

`If` -statements have the following structure:

```
if (condition){
    # Perform an action
}
```

Translation: If this `condition` is satisfied, then perform this action.

Note that `length(condition) == 1` must evaluate to `TRUE` .

3.2 If-else-statements

`If-else` -statements have the following structure:

```
if (condition){
    # Perform an action
} else {
    # Perform a different action
}
```

Translation: If this `condition` is satisfied, then perform this action. If this condition is not satisfied, then perform a different action.

Another `condition` can be added by putting another `if (condition)` after `else` .

```
if (condition){
    # Perform an action
} else if (other condition){
    # Perform a different action
}
```

Translation: If this `condition` is satisfied, then perform this action. If not, then check whether the `other condition` is satisfied. If the `other condition` is satisfied, perform a different action.

3.3 For-loops

`for` -loops have the following structure:

```
for(i in 1:4){
    # Perform an action that needs to repeated
}
```

Translation: For a specified number of times (`1:4`), perform this action on each iteration (`i`).

3.4 While-loops

`while` -loops have the following structure:

```
while(condition){
    # Perform an action that needs to be repeated
}
```

Translation: For an unspecified number of times, perform this action as long as the `condition` is `TRUE` .

3.5 Functions

Functions have the following structure:

```
myFunction <- function(x, ...){

    # Perform action on input x to get result

    return(result)
}
```

Translation: Take the input `x` , perform some actions, and return the resulting `outcome` .

The function `myFunction` can then be called with the data in `x` using:

```
myFunction(x)
```

Part IV: Descriptive statistics and hypothesis testing

4.1 Descriptive statistics

`mean(x)`	Gives the **average** of `x`
`median(x)`	Gives the **median** of `x`
`sum(x)`	Gives the **sum** of `x`
`min(x)`	Gives the **minimum** of `x`
`max(x)`	Gives the **maximum** of `x`
`sd(x)`	Gives the **standard deviation** of `x`
`var(x)`	Gives the **variance** of `x`
`cor(x, y)`	Gives the **correlation** between `x` and `y`
`table(x)`	Gives a **frequency table** of `x`

4.2 Simple hypothesis testing

`t.test(x, y)`	Performs a **t-test** on the data `x` (`y` is optional)
`cor.test(x, y)`	Performs a **correlation test** on the data `x` and `y`
`binom.test(x, n, p)`	Performs a **binomial test** on the data
`chisq.test(x, y)`	Performs a **chi-square test** on the data
`aov(formula, x)`	Performs an **ANOVA** on the data in `x` using formula

4.3 Regression

`lm(y ~ 1 + x)`	Regression model $y = \beta_0 + \beta_1 \times x$
`lm(y ~ 1 + x + z)`	Regression model $y = \beta_0 + \beta_1 \times x + \beta_2 \times z$
`lm(y ~ 0 + x)`	Regression model $y = \beta_1 \times x$
`lm(y ~ x:z)`	Regression model $y = \beta_0 + \beta_1 \times x \times z$
`lm(y ~ x * z)`	Regression model $y = \beta_0 + \beta_1 \times x + \beta_2 \times z + \beta_3 \times x \times z$
`lm(y ~ poly(x, 2))`	Regression model $y = \beta_0 + \beta_1 \times x^2$
`coef(x)`	Gives **coefficients** of regression model in `x`
`residuals(x)`	Gives **residuals** of regression model in `x`
`AIC(x)`	Gives the **AIC** value of regression model in `x`
`BIC(x)`	Gives the **BIC** value of regression model in `x`
`summary(x)`	Gives a summary of **regression model** in `x`
`confint(x)`	Gives the **confidence interval** of model `x`
`abline(x)`	When added to a plot, plots the regression line
`predict(x, newdata)`	Use the **regression model** in `x` to predict new data

Beginner R exercises

Creating and removing objects in the environment

B 1) Run the following code in R:

```
a = b = 1
a = 2
```

What happened? How do you know (which command)? How many objects did you create in your environment? Try assigning the values using the `<-` operator. Does the result differ from the result when you are using `=` to create the variables?

B 2) Remove the variables `a` and `b` from your environment using the `rm()` function.

B 3) Run the following code in R:

```
apples <- 5; pears <- 3; pineapples <- 6
```

Then try the following code:

```
apples + pineapples
apples + Pears
```

Why can't you add `apples` and `Pears` ?

B 4) Use R to compute the square root of 81 and store the result in `t1` .

B 5) Use R to compute 81 to the power a half and store the result in `t2` .

B 6) Use the `==` operator to check whether the contents of `t1` and `t2` are the same.

Working directory and help functionality

B 7) With the command `getwd()` you can find the current working directory of the R process. Explain what the working directory means and why it is important.

B 8) Change the working directory of the R session to a folder of your preference. What function do you use? How can you check whether your change has worked?

B 9) Run the following code in R:

```
demo()
```

What does this code do? How do you run the demo `persp` (in package `graphics`)?

B 10) Run the following code in R:

```
a <- c(1, 6, 7, 8, 9, NA)
mean(a)
```

This gives `NA`, why? Check `?mean` to find out what arguments the `mean()` function takes as input. Can you find a way to compute the mean with the missing value removed?

B 11) According to Google there exists an R-function called `mvrnorm()`. Typing `mvrnorm` or `?mvrnorm`, however, gives an error. Why? Which library do you have to load first? How?

B 12) What does `%%` do? Can you ask help with `?%%`, like in `?mean`? If not, how? What is `%%` doing?

B 13) The `cor()` function computes correlations. As an example, run the following code in R:

```
cor(c(1, 2, 3, 4), c(1, 4, 7, 15))
```

How can you find out (which function) whether this correlation is statistically significant? How can you find such a function? What is the confidence interval of the correlation? How can you find all functions that have 'cor' in their name?

Data types of objects

B 14) Run the following code in R:

```
a1 <- '1'; a2 <- 1; a3 <- TRUE
```

What are the data types of `a1` , `a2` , and `a3` ? Now run the following code in R:

```
b1 <- c(a1, a2); b2 <- c(a1, a3); b3 <- c(a2, a3); b4 <- c(a1, a2, a3)
```

What are the modes of `b1` , `b2` , `b3` , and `b4` ? Can you explain?

B 15) Why does `TRUE/TRUE` yield 1, `FALSE/TRUE` yield 0, `FALSE/FALSE` yield `NaN` , and `TRUE/FALSE` yields `Inf` ?

B 16) Convert the logical `TRUE` to a numerical. Convert this numerical to a character. Next, convert the logical `TRUE` to a character. Why do these results differ? (Hint: look at as.numeric())

B 17) How can you check whether a vector is numeric? Is the vector `c(1,0)` a numeric vector? And the vector `c(TRUE, FALSE)` ? And the vector `c(TRUE, FALSE, 1, 0)` ? Why? (Hint: look at is.numeric())

Object types: Vectors

B 18) Use the `c()` function (or `:`) to create the following vector:

```
-2 -1 0 1 2 3 4 5 6 7 8
```

What is the length of this vector?

B 19) Make a vector that starts at -5 and goes to 5 in steps of 0.5. Can you find out more ways to construct this vector than by using the `c()` function?

B 20) Create the following vector:

```
13 15 17 19 21 23 25 27 29 31 33
```

B 21) Run the following code in R:

```
a <- 1:5
b <- -a
```

Combine the vectors `a` and `b` into one vector.

B 22) What is the result of `rep(2, 10)`? What function argument receives the value 10?

B 23) Use the `rep()` function to create the following vector:

```
3 3 3 4 4 4 5 5 5 6 6 6 7 7 7
```

B 24) Suppose you want to make the vector `"a" "a" "b" "b" "c" "c" "d" "d" "e" "e"` with the `rep()` function but `rep(letters[1:5], 2)` does not work. What do you have to change in this command?

B 25) The following code gives the first 10 uneven numbers. What is the sum of these numbers?

```
(1:10) * 2 - 1
```

B 26) Make a vector with the first 10 even numbers. Next, make a vector with the first 10 numbers divided by 5. Finally, create the vector `5 8 11 14 17 20 23 26 29 32`.

B 27) `logical(5)` makes a logical vector of length 5. How can you make the same vector with the `vector()` function?

B 28) Run the following code in R:

```
paste(rep(c('a', 'b'), each = 5), 1:5, sep = '.')
```

Next, create the following vector:

```
"x1m" "x1f" "x2m" "x2f" "y1m" "y1f" "y2m" "y2f"
```

B 29) Run the following code in R:

```
s <- c(5, 7, 2, 8)
```

First sort `s` from highest to lowest. What option in the `sort()` function do you use? Next, sort `s` from lowest to highest. Why don't you have to use the same option here?

B 30) Run the following code in R and explain why the second line gives a warning:

```
1:10 + 1:2 - 1
1:10 + 1:3 - 1
```

Object types: Matrices

B 31) Use the `matrix()` function to create the following matrix:

```
25  24  23  22  21
20  19  18  17  16
15  14  13  12  11
10   9   8   7   6
 5   4   3   2   1
```

B 32) Create the following matrix:

```
0  0  0  0
1  1  1  1
0  0  0  0
1  1  1  1
```

B 33) Run the following code in R:

```
m1 <- matrix(1:20, , 4)
```

Why don't you have to give R the number of rows for the matrix?

B 34) With what function can you transpose a matrix?

B 35) `mean(m1)` returns one number. How can you get the mean of each column in `m1` without calculating them each in turn?

B 36) Find out the values of the diagonal in matrix `m1` . Can you make a new matrix of 0's with on the diagonal the diagonal of `m1` ?

B 37) Add a new row to matrix `m1` with the sum of each row of the matrix. Use the `rowSums()` function.

B 38) Run the following code in R:

```
m2 <- scale(m1)
```

What are the means of the columns in matrix `m2` ? And the standard deviations (hint: use the R function apply)? Given these values, can you find out what the `scale()` function does?

Object types: Data frames

B 39) Create the following data frame without typing out the first two columns:

	subject	time	score
1	1	t1	7
2	1	t2	8
3	2	t1	8
4	2	t2	8
5	3	t1	9
6	3	t2	8
7	4	t1	9
8	4	t2	7
9	5	t1	7
10	5	t2	6

B 40) Run the following code in R:

```
a <- matrix(1:10, , 2, TRUE)
```

Convert the matrix `a` to a data frame.

Object types: Lists

B 41) Create a list consisting of the following entries:

- A numeric vector named `subject` containing a sequence of 1 to 5,
- A character vector named `time` containing the characters `t1` , and `t2` ,
- A numeric matrix named `score` with five rows and five columns, containing the numbers 1 to 25.

Object types: Factors

B 42) Run the following code in R:

```
g <- gl(4, 3)
```

Is `g` a vector? (If not, what is it?) How can you check?

B 43) Run the following code in R:

```
v <- rep(c('m', 'f'), 3)
```

Encode the variable `v` as factor levels. How do you check whether you succeeded?

B 44) Run the following code in R:

```
x <- 0:10%%3
```

Convert `x` to a factor. How many factor levels has `x` ?

Indexing

B 45) Suppose you have `x <- 3` . Now type `x[3] <- 5` . What is the value of `x[2]` ?

B 46) Run the following code in R:

```
a <- c(9, 2, 4, 5, 2, 7, 5)
```

Change the first element into an 8. Next, change the 2's in the vector into zero's.

B 47) Run the following code in R:

```
l <- matrix(c(1, 4, 6, 7, 4, 7), ncol = 3)
```

Use the square brackets to find out the second value of the third column of the matrix `l` .

B 48) Run the following code in R:

```
m <- matrix(sample(1:100, 100, replace = T), 10, 10)
```

Select the third and fifth row of the matrix `m` .

B 49) Select those rows from matrix `m` for which the value in the first column is less than 5.

B 50) Select all but the fourth column of matrix `m` .

B 51) Change the matrix `m` to a data frame. Give the columns of the data frame the names `'trial.1'`, `'trial.2'`, etc.

B 52) Select the second to the fifth element of column `'trial.1'`. Select the first 2 elements of column `'trial.4'`.

B 53) Run the following code in R:

```
b <- c(1, 2, 2, 1, 2, 1, 1, 2, 2, 1)
```

Suppose you want to change all ones into twos and all twos to ones. To try out, run the following code in R. It doesn't work. Can you find a method so that this is done correctly?

```
b[b==1] <- 2
b[b==2] <- 1
```

B 54) Run the following code in R:

```
n <- c('bananas', 'apples')
```

Use the `gsub()` function to change all a's in `n` to dots (e.g., bananas becomes b.n.n.s).

B 55) Run the following code in R:

```
set.seed(1)
grades <- data.frame(1:30, matrix(sample(4:10, 60, TRUE), , 2))
names(grades) <- c('student', 'exer', 'exam')
```

Students pass a course when the average grade is at least 5.5 and both grades are larger than 5. Select (index) which students passed the course. Also find out how can you select which students did not pass the course.

Conditions

B 56) What does `(a < 0 | b < 0) (a * b < 0)` mean? If this condition is `TRUE` and `a` is negative, what do you know about `b` ?

B 57) Why does `!sum(is.na(c(1, 2, 3, 4, NA, 7))) <= 2` return `FALSE` ?

B 58) Run the following code in R and explain the result:

```
x <- 1:24
which(24%%x == 0)
```

B 59) Run the following code in R:

```
x <- y <- 1:10
sum(x==y) == length(x==y)

x <- y <- 1:10
y[1] <- 0
sum(x==y) == length(x==y)
```

Explain the result of this code. Can you make a test whether `x` and `y` are identical using the `min()` function? And with the `mean()` or `prod()` function? Is there a special function to test whether two vectors are identical?

Sampling and simulating data

B 60) You can throw a die 100 times by typing the following code in R:

```
throws <- sample(1:6, 100, TRUE)
```

How can you see how often each number shows up? How often did each number show up?

B 61) Sample, with replacement, 4 cards from the set `'jack'`, `'queen'`, `'king'` and `'ace'` with equal probabilities.

B 62) Create 20 uniform random numbers between 0 and 100. Why is `sample(1:100, 20)` not correct?

B 63) Put 21 uniform random numbers in a vector. What is the median of this vector? Sort the vector from highest to lowest. What is the 11th element of the sorted vector?

B 64) Create 100 normally distributed numbers with `mean = 100` and `sd = 15`. What is the variance of these numbers? Run the code that you used for this again. Why is the variance not exactly the same the second time? What can you use so that, if you run your code again, the results are the same?

B 65) Run the following code in R:

```
x <- matrix(rnorm(100), 25, 4)
```

Find out the covariance matrix of matrix `x`. Also find out the correlation matrix of matrix `x`.

Reading and writing data

B 66) What does the `read.csv()` function do? How would you read in Excel files? What package contains the function `read.spss()`? Can you find other file types that R has read functions for?

B 67) Read in the file `example.csv` from the online resources using the `read.csv()` function. How many observations does this data contain?

B 68) Run the following code in R:

```
d <- data.frame(sex = c('m', 'f'),
                Age = c(6.7, 6.5, 5.6, 5.4),
                var1 = c(9, 5, 4, 4),
                var2 = c(10, 5, 8, 4))
```

Write the data frame `d` to a file named `example.xlsx` so that Excel is able to open it. What options did you use?

Advanced R exercises

A 1) *Let's sort*

Difficulty: Easy

Sorting is easy in R. You can just use `sort(x)` . Let's try to build your own sort function. One straightforward algorithm is called bubble-sort, check the 'Bubblesort' entry on Wikipedia for more information (https://en.wikipedia.org/wiki/Bubble_sort).

☐ Create a function that can sort a vector using the bubble-sort approach.

A 2) *T-test*

Difficulty: Easy

R had a built-in function `t.test()` that performs a t-test. Read the Wikipedia entry for the T-test (https://en.wikipedia.org/wiki/Student%27s_t-test) to find out exactly what is going on in this function.

☐ Program a function that can be used to perform a t-test. Think about what results the user would like to get and how the user should use the function.

A 3) *Mann-Whitney U test*

Difficulty: Easy

Read the Wikipedia entry for the Mann-Whitney U Test for some first information (https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test). The Mann-Whitney U test is the non-parametric equivalent of the two sample t-test and is used when the assumptions of the parametric t-test are violated.

☐ Program your own Mann-Whitney U function and compare it to the built-in Mann-Whitney U test in R to see whether you get the same results.

A 4) *Caesar cipher*

Difficulty: Medium

For some initial information, read the Wikipedia entry on the Caesar cipher encryption/decryption method (https://en.wikipedia.org/wiki/Caesar_cipher).

☐ Program a function that can be used to encrypt a character string using the Caesar cipher and program another function that can be used to decrypt a character string using the Caesar cipher.

☐ Use the decrypt function to decrypt the following string that is encrypted using a Caesar cipher and a key of 13:

```
Nznmvat jbex! Lbh qrpelcgrq guvf zrffntr pbeerpgyl.
```

A 5) *Infinite monkeys, infinite typewriters*

Difficulty: Medium

The infinite monkey theorem says that a monkey hitting keys at random on keyboard for an infinite amount of time will almost surely type any given text, such as the entire contents of this workbook. In fact, the monkey would almost surely type every possible finite text an infinite number of times. For further information, read the Wikipedia entry for the Infinite Monkey Theorem (https://en.wikipedia.org/wiki/Infinite_monkey_theorem).

- ☐ Simulate one monkey, typing random letters on a typewriter in sequences of 5 letters (so, typing 5 letters, then a space, and so on). Turn this into a function that returns the number of letters typed before making a coherent 5-letter word. Use a file that contains all English 5 letter words to check whether a word is valid (e.g., http://www-cs-faculty.stanford.edu/~knuth/sgb-words.txt).

- ☐ Run the function 500 times and make a nice plot with the results (WARNING: a 500 times will take approximately 30 minutes, so start with a much smaller number for testing your code).

A 6) *Hangman*

Difficulty: Hard

Hangman is a paper and pencil guessing game for two or more players. One player thinks of a word, phrase or sentence and the other(s) tries to guess it by suggesting letters within a certain number of guesses. Read the Wikipedia entry on "Hangman" for some information about the rules of the game (https://en.wikipedia.org/wiki/Hangman_%28game%29).

- ☐ Play Hangman against the computer by programming the game in R. For an additional challenge, try to include graphics for wrong answers.

A 7) *Blackjack*

Difficulty: Hard

Blackjack is a casino card game between one or more players and a dealer, where each player in turn competes against the dealer. Read the Wikipedia entry for the "Blackjack" casino game for some more information about the rules of the game (https://en.wikipedia.org/wiki/Blackjack).

- ☐ Program the Blackjack game. For an additional challenge, try to include a betting system.

Answers to the assignments

Chapter 1: Descriptive statistics

1.1 a)
```
Mean:   6.6                    Range:   8
Mode:   8                      Lower quartile:   4
Median:   7                    Upper quartile:   9
                               Interquartile range:   5
```

1.1 b)
```
Mean:   5.91                   Range:   9
Mode:   7                      Lower quartile:   3.5
Median:   6.5                  Upper quartile:   8.5
                               Interquartile range:   6
```

1.1 c) Assignment 1.1b was probably harder to do as you had to take the middle of two numbers to find the median, and the lower and upper quartiles.

1.1 d) These data sets are negatively skewed.

Explanation: The mean is lower than the median and mode, and so more values are concentrated on the right side (tail) of the distribution graph while the left tail of the distribution graph is longer.

1.2 a) The `View()` command opens a window in which you can inspect the data.

```
dataset1 <- c(2, 7, 4, 5, 8, 10, 10, 7, 9, 2, 8, 8, 9, 4, 6)
View(dataset1)
```

1.2 b)
```
Mean:   6.6                    Median:   7
Mode:   8                      Range:   8
```

```
mean(dataset1) # Mean: 6.6
table(dataset1) # Mode: 8 is the most occurring number (3 times)
median(dataset1) # Median: 7
range(dataset1) # Range: 2 to 10 = 8
```

1.2 c) The **quantile()** command returns the minimum (0%), lower quartile (25%), median (50%), upper quartile (75%), and maximum (100%).

```
quantile(dataset1, type = 6)
```

1.2 d)

Mean: 5.91	Range: 9
Mode: 7	Lower quartile: 3.5
Median: 6.5	Upper quartile: 8.5

```
dataset2 <- c(7, 7, 6, 5, 2, 1, 3, 7, 5, 9, 9, 10)

mean(dataset2) # Mean: 5.91
table(dataset2) # Mode: 7 is the most occurring number (3 times)
median(dataset2) # Median: 6.5
range(dataset2) # Range: 1 to 10 = 9
```

1.3 a) The code opens a new window in which you can select the **.csv** file that you want to read into your R session. Using the **colnames()** function, you can see that the **bloodPressure.csv** file contains 6 columns named **Number** , **Age** , **BloodPressure** , **Cholestrol** , **Gender** , **Description** .

1.3 b) You can use a relative path to the file on your computer by providing it directly to the **read.csv()** function (in quotes **'bloodPressure.csv'**). Remember to set your working directory correctly, since R will look inside the working directory folder when it receives such a path. You can also specify a full path (like **'C://path/to/file/bloodPressure.csv'**). With full file paths, R will know exactly where to look, and the location of your working directory does not matter.

```
dataset3 <- read.csv('bloodPressure.csv')
```

1.3 c) The mean age of the respondents ($n = 60$) in the data set is 45.15 years. The minimum age of the respondents is 17, and the maximum age is 69. The age of the respondents spans 52 years. The most occurring age is 39. Twenty-five percent of the respondents is aged below 34.5, fifty percent is aged below 46, and 75 percent is aged below 58.5.

```
mean(dataset3$Age) # Mean: 45.15
table(dataset3$Age) # Mode: 39 is the most occurring number (4 times)
median(dataset3$Age) # Median: 46
range(dataset3$Age) # Range: 17 to 69 = 52

quantile(dataset3$Age, type = 6)
# Minimum: 17
# Lower quartile: 34.5
# Median: 46
# Upper quartile: 58.5
# Maximum: 69
```

1.3 d) Mode: 39

```
getMode <- function(x){
   uniqx <- unique(x)
   uniqx[which.max(tabulate(match(x, uniqx)))]
}

getMode(dataset3$Age) # Mode: 39
```

1.3 e) Mean: 130.62 Range: 121
Mode: 129 Lower quartile: 118.75
Median: 131.5 Upper quartile: 145.75

```
mean(dataset3$BloodPressure) # Mean: 130.62
getMode(dataset3$BloodPressure) # Mode: 129
median(dataset3$BloodPressure) # Median: 131.5
range(dataset3$BloodPressure) # Range: 45 to 166

quantile(dataset3$BloodPressure, type = 6)
# Minimum: 45
# Lower quartile: 118.75
# Median: 131.5
# Upper quartile: 145.75
# Maximum: 166
```

1.3 f) These data sets are not skewed.

Explanation: The mean is lower than the median but not than the mode, and so you cannot conclude that the distribution is skewed in any direction.

1.3 g) Variance: 1.059
Standard deviation: 1.029

```
var(dataset3$Cholestrol) # Variance: 1.059
sd(dataset3$Cholestrol) # Standard deviation: 1.029
```

1.3 h)

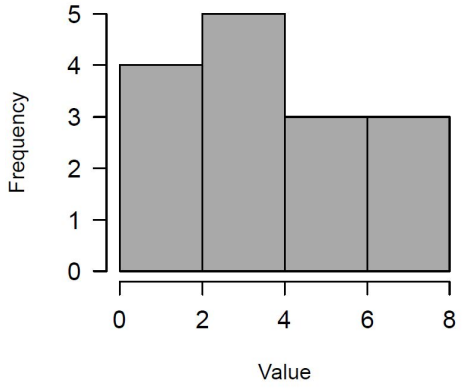
```
# The standard deviation is the square root of the variance
sd(dataset3$Cholestrol) == sqrt(var(dataset3$Cholestrol))
```

Chapter 2: Creating graphs from data

2.1 a)

0 to 2	2 to 4	4 to 6	6 to 8
Frequency: 4	Frequency: 5	Frequency: 3	Frequency: 3

2.1 b)



2.2 a) In the plot window you can see that a histogram for the values in the variable **dataset4** has been drawn by R.

```
dataset4 <- c(1.5, 5.5, 1.7, 7.2, 1.2, 7.9, 1.4, 3.6,
3.1, 3.8, 5.9, 3.6, 5.1, 3.2, 7.1)
hist(dataset4)
```

2.2 b) The histogram drawn by R has four bars for the ranges 1–2, 3–4, 5–6, and 7–8. The histogram from assignment 2.1b has four bars for the ranges 0–2, 2–4, 4–6, and 6–8. The difference in these two histograms is that they have a different number of bars for different ranges.

2.2 c)

```
# The breaks argument determines the number of bars in the histogram
# see ?hist for more help on the hist() function
hist(dataset4, breaks = 4)
```

2.2 d)

```
hist(dataset4, breaks = 4, col = 'blue', # col sets the bar color
xlab = 'x-axis', # xlab sets the x-axis name
ylab = 'Frequency', # ylab sets the y-axis name
main = 'My histogram') # main sets the title
```

2.2 e) These data are positively skewed.

Explanation: The mean is higher than the median, and so more values are concentrated on the left side (tail) of the distribution graph while the right tail of the distribution graph is longer.

```
mean(dataset4) # Mean: 4.12
median(dataset4) # Median: 3.6
```

2.3 a)

```
data(swiss) # Import the swiss data

education <- swiss$Education
agriculture <- swiss$Agriculture
```

2.3 b)

```
plot(x = education, y = agriculture,
     xlab = 'Percentage of education beyond primary school',
     ylab = 'Percentage of males involved in agriculture')
```

2.3 c) Looking at the scatter plot, there seems to be a tendency for provinces that have a high percentage of education beyond primary school to also have a low percentage of males involved in agriculture.

2.3 d)

```
plot(x = education, y = agriculture,
     xlab = 'Percentage of education beyond primary school',
     ylab = 'Percentage of males involved in agriculture',
     col = 'blue',
     main = 'Provinces in Switzerland',
     las = 1, # las sets the rotation of the axis labels
     bty = 'n') # bty = 'n'removes the outer border lines
```

2.4 a)

```
data(EuStockMarkets)
stockData <- data.frame(EuStockMarkets)

plot(stockData$DAX,
     type = 'l', # type = 'l'creates lines instead of dots
     xlab = 'Time',
     ylab = 'Price')
```

2.4 b)

```
# The lines() function adds a line to an existing plot
lines(stockData$SMI, col = 'red') # Add a line for the SMI stock
lines(stockData$CAC, col = 'blue') # Add a line for the CAC stock
lines(stockData$FTSE, col = 'green') # Add a line for the FTSE stock
```

2.5 a)

```
Minimum:  1.2                    Lower quartile:  35.3
Median:  54.2                    Upper quartile:  67.8
Maximum:  89.7
```

```
quantile(agriculture, type = 6)
# Minimum: 1.2
# Lower quartile: 35.3
# Median: 54.1
# Upper quartile: 67.8
# Maximum: 89.7
```

2.5 b)

```
boxplot(agriculture)
```

2.5 c) The first line creates a vector called **educationLevel** that contains 47 times **'2.Medium'** . The second line changes the **'2.Medium'** to **'1.Low'** for the provinces that have a percentage of education beyond primary school lower than 6. The third line changes the **'2.Medium'** to **'3.High'** for the provinces that have a percentage higher than 12. The resulting table shows how many provinces had a percentage lower than 6, between 6 and 12, and higher than 12.

```
educationLevel <- rep('2.Medium', 47)
educationLevel[education <= 6] = '1.Low'
educationLevel[education >= 12] = '3.High'
table(educationLevel)
```

2.5 d) The code creates a box plot of the agriculture variable for each education level `'1.Low'`, `'2.Medium'`, and `'3.High'`.

```
boxplot(agriculture ~ educationLevel)
```

Chapter 3: Confidence intervals and hypothesis testing

3.1 a) $N = 4513$ $s = 25$
$n = 100$ $\sigma =$ unknown
$\bar{x} = 145$ $\mu =$ unknown

3.1 b) $\mu =$ 145 seconds

Explanation: The best estimate for the population mean μ is the sample mean $\bar{x}$.

3.1 c) $SE_\mu = \frac{s}{\sqrt{n}} = \frac{25}{\sqrt{100}} = 2.5$

3.1 d) z-value: 2.576

Explanation: In table 2 of the formula sheet, the cumulative probability in that lies the closest to 0.995 (split the risk over two sides) is 0.9949. That value can be found at a z-value of 2.576.

Lower bound: $\bar{x} - z_{0.995} \times SE_\mu = 145 - 2.567 \times 2.5 = 138.56$
Upper bound: $\bar{x} + z_{0.995} \times SE_\mu = 145 - 2.567 \times 2.5 = 151.44$

3.1 e) $\mu_0 = 150$ (the to be tested limit of 150 seconds for the actual population mean call duration).

$H_0 : \mu \geq \mu_0$ $H_0 : \mu < \mu_0$

3.1 f) Upper bound: $\bar{x} + z_{0.99} \times SE_\mu = 145 - 1.645 \times 2.5 = 149.11$

3.1 g) The upper bound of the confidence interval for μ is lower than μ_0. H_0 is rejected with 99% confidence. μ is shown to be significantly lower than 150 seconds. There is a risk of 5% for a type-I error.

3.2 a)

i	x_i	$(x_i - \bar{x})$	$(x_i - \bar{x})^2$
1	3.03	-0.0863	0.0074
2	3.45	0.3338	0.1114
3	3.94	0.8238	0.6786
4	2.34	-0.7763	0.6026
5	3.34	0.2238	0.0501
6	2.53	-0.5863	0.3437
7	2.88	-0.2363	0.0558
8	3.42	0.3038	0.0923

$\sum x_i = 24.93 \qquad \sum (x_i - \bar{x})^2 = 1.9418$

$\bar{x} = 3.116 \qquad s^2 = 0.277$

3.2 b) Hartley's F: $\frac{s^2_{max}}{s^{min}} = \frac{1.113}{0.227} = 4.018$

3.2 c) $H_0 : \sigma_1^2 = \sigma_2^2 = \sigma_3^2 \qquad H_0 : \sigma_1^2 \neq \sigma_2^2 \neq \sigma_3^2$

3.2 d) Hartley's F_{max}: 6.94

3.2 e) The value $F = 4.018$ is lower than critical value $F_{max} = 6.94$. H_0 is not rejected. There is no indication the variance for these months is not homogeneous. There is a risk of a type-II error.

3.3 a)

```
# Be sure to set your working directory when providing a relative path
dataset5 <- read.csv('populations.csv')
```

3.3 b) The code creates four random samples of size 90 from the columns **P1** - **P4** in the data frame called **dataset5** . The seed makes sure that you can recreate the same random samples again, so that if you close R and continue tomorrow we get the same samples.

```
set.seed(54321) # You can replace 54321 with your own seed number

sample1 <- sample(dataset5$P1, size = 90)
sample2 <- sample(dataset5$P2, size = 90)
sample3 <- sample(dataset5$P3, size = 90)
sample4 <- sample(dataset5$P4, size = 90)
```

3.3 c) Standard error sample 1: 29.24
Standard error sample 2: 7.64
Standard error sample 3: 30.61
Standard error sample 4: 18.09

```
# Means
x1 <- mean(sample1) # Mean: 456.78
x2 <- mean(sample2) # Mean: 511.02
x3 <- mean(sample3) # Mean: 790.32
x4 <- mean(sample4) # Mean: 533.37

# Standard deviations
sd1 <- sd(sample1) # Standard deviation: 277.38
sd2 <- sd(sample2) # Standard deviation: 72.43
sd3 <- sd(sample3) # Standard deviation: 290.46
sd4 <- sd(sample4) # Standard deviation: 171.58

# Standard errors se = sd / sqrt(n)
se1 <- sd1 / sqrt(length(sample1)) # Standard error: 29.24
se2 <- sd2 / sqrt(length(sample2)) # Standard error: 7.64
se3 <- sd3 / sqrt(length(sample3)) # Standard error: 30.61
se4 <- sd4 / sqrt(length(sample4)) # Standard error: 18.09
```

3.3 d) The value 1.644854 comes from the standard normal distribution with $\mu = 0$ and $\sigma = 1$. This is the z-value for a 95% one-sided confidence interval (or a 90% two-sided confidence interval).

3.3 e)

```
# Gives the left-tailed probability (z-value) for 95% confidence
qnorm(p = 0.95) # 1.645
```

3.3 f) In a 95% confidence interval there is 2.5% of the risk at the lower bound and 2.5% of the risk at the upper bound. You can therefore use the 97.5% quantile of the standard normal distribution to get the z-value for a two-sided 95% confidence interval and use it for both upper and lower bound.

```
z <- qnorm(p = 0.975)
```

3.3 g)

```
# Lower bounds
lb1 <- x1 - z * se1 # Lower bound: 399.48
lb2 <- x2 - z * se2 # Lower bound: 496.06
lb3 <- x3 - z * se3 # Lower bound: 730.31
lb4 <- x4 - z * se4 # Lower bound: 497.92

# Upper bounds
ub1 <- x1 + z * se1 # Upper bound: 514.10
ub2 <- x2 + z * se2 # Upper bound: 525.99
ub3 <- x3 + z * se3 # Upper bound: 850.33
ub4 <- x4 + z * se4 # Upper bound: 568.81
```

3.3 h)

	sample1	sample2	sample3	sample4
ub	514.10	525.99	850.33	568.81
x	456.78	511.01	790.32	533.37
lb	399.48	496.06	730.31	497.92

	P1	P2	P3	P4
mu	495.54	500.08	748.96	556.43
mu in interval?	YES	YES	YES	YES

```
# Population means
mu1 <- mean(dataset5$P1) # Mean: 495.54
mu2 <- mean(dataset5$P2) # Mean: 500.08
mu3 <- mean(dataset5$P3) # Mean: 748.96
mu4 <- mean(dataset5$P4) # Mean: 556.43
```

3.3 i) Yes, in this case all population means are inside the intervals.

3.3 j) The population mean will fall inside the interval 95 out of a 100 times (95% confidence). This means that about 1 in 20 confidence intervals will not have the true population mean between their lower and upper bound.

3.4 a)

```
# install.packages('car')
library(car)

# This create a 2x2 layout
layout(matrix(c(1, 2, 3, 4), byrow = TRUE, nrow = 2))

hist(sample1, col = 'gray')
hist(sample2, col = 'gray')
hist(sample3, col = 'gray')
hist(sample4, col = 'gray')

# This resets the layout to the default (1 plot)
layout(1)
```

3.4 b) `Sample 2 looks like it might come from a normal distribution.`

3.4 c)

```
# This create a 2x2 layout
layout(matrix(c(1, 2, 3, 4), byrow = TRUE, nrow = 2))

qqPlot(sample1, distribution = 'norm')
qqPlot(sample2, distribution = 'norm')
qqPlot(sample3, distribution = 'norm')
qqPlot(sample4, distribution = 'norm')

# This resets the layout to the default (1 plot)
layout(1)
```

3.4 d) The **sample1** histogram looks normal in the middle, but has too many low and too many high values. This can be seen in the qqplot by the dots on the left of the diagonal at the bottom and the dots on the right at the top.

The **sample2** histogram looks normal so the dots in the qqplot are almost everywhere on the diagonal. Only in the right part of the middle the histogram frequency is a bit too low, this is reflected in the dots below the diagonal around zero in the qqplot.

The **sample3** histogram is too high on the sides (or too low in the middle) to be normal. This can be seen in the qqplot by the dots on the left of the diagonal at the bottom and the dots on the right at the top.

sample4 is negatively skewed. This can be seen in the qqplot from the arch shape; the left of the histogram is too low causing the dots on the right of the diagonal and the right of the histogram is too high also causing dots to the right of the diagonal.

3.4 e) H_0: The sample is normally distributed
H_1: The sample is not normally distributed

3.4 f) Samples 1, 3, and 4 are not normally distributed, since the p-value is lower than 1% (for 99% confidence). Sample 2 is normally distributed, since the p-value is higher than 1%.

```
shapiro.test(sample1) # p-value: 0.0022
shapiro.test(sample2) # p-value: 0.949
shapiro.test(sample3) # p-value: 0.0068
shapiro.test(sample4) # p-value: 0.0001
```

3.4 g) When the sample is not normally distributed, the sample can be used to estimate the population mean.

Explanation: This method to estimate the population mean assumes the distribution of sample means to be normally distributed. It does not assume a normally distributed sample. The Central Limit Theorem states that any sample large enough ($n \geq 30$) will have a normal distribution of sample means, so you can use this method here ($n = 90$) without problems.

3.5 a)

```
library(car)
data(iris)

plot(x = iris$Species, y = iris$Sepal.Length,
    col = 'grey', main = 'Sepal Length')

plot(x = iris$Species, y = iris$Sepal.Width,
    col = 'grey', main = 'Sepal Width')
```

3.5 b) Looking at the width of the range and quartile ranges: For sepal length the variance for setosa looks much smaller than for the other two species, for sepal width all variances look similar.

3.5 c) $H_0 : \sigma_1^2 = \sigma_2^2 = \sigma_3^2$ $H_0 : \sigma_1^2 \neq \sigma_2^2 \neq \sigma_3^2$

3.5 d) The p-value for the sepal length is lower than 10%. This implies that H_0 is rejected. This means that the variance of the sepal length over the species is not homogeneous. There is a 5% change of a type-I error.

The p-value for the sepal with is not lower than 10%. This implies that H_0 is not rejected. This means that there is no indication that the variance of the sepal width over the species is not homogeneous. There is a risk of a type-II error.

```
# Levene's Test for Homogeneity of Variance
leveneTest(y = iris$Sepal.Length,
group = iris$Species) # p-value: 0.002259

leveneTest(y = iris$Sepal.Width,
group = iris$Species) # p-value: 0.5555
```

Chapter 4: Correlation and regression

4.1 a) Two variables can either be positively related, not related, or negatively related. The most logical relationship is that the distance a customer lives from the store is negatively related to how many times they visit the store.

4.1 b)

i	x_i	y_i	$(x_i - \bar{x})$	$(y_i - \bar{y})$	$(x_i - \bar{x})(y_i - \bar{y})$
1	4.87	2.90	1.868	-1.09	-2.053
2	3.04	4.50	0.038	0.501	0.0194
3	1.65	4.94	-1.351	0.941	-1.272
4	2.88	3.28	-0.121	-0.718	0.087
5	2.31	4.73	0.691	0.731	-0.505
6	3.96	2.64	0.958	-1.358	-1.303
7	2.70	3.70	-0.301	-0.298	0.089
8	2.60	5.30	-0.401	1.301	-0.522

$\bar{x} = 3.001$ $\sum(x_i - \bar{x})(y_i - \bar{y}) = -5.458$

$\bar{y} = 3.998$ $n - 1 = 7$

$s_{xy} = -0.779$

4.1 c) The covariance is negative. A negative covariance indicates that that as one variable deviates from the mean, the other variable deviates in the other direction. This means that, when a customer's distance from the store in kilometers is higher than the mean, their average visits per week will likely be lower than the mean.

4.1 d) The disadvantage of using the covariance as a measure for the strength of this relationship is that it depends on the measurement unit (kilometers vs. meters) that the co-worker asks the questions in. If the co-worker would have asked the question in meters the covariance would have increased by a 1000 times, namely -779.84.

4.1 e) $s_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{6.977}{7}} = 0.998$ $s_y = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}} = \sqrt{\frac{7}{7}} = 1$

4.1 f) $r_{xy} = \frac{s_{xy}}{s_x \times s_y} = \frac{-0.779}{0.998 \times 1} = -0.779$

4.1 g) The coefficient is -0.779, which represents is a relatively strong negative relationship.

4.2 a) H_0: $\rho_{xy} \geq 0$ H_1: $\rho_{xy} < 0$

4.2 b) $N =$ unknown $r_{xy} = -0.779$
$n = 8$ $\rho_{xy} =$ unknown

4.2 c) $z_{0.05} = -1.645$

4.2 d) $z_r = \frac{1}{2} \times log_e(\frac{1+r}{1-r}) = \frac{1}{2} \times log_e(\frac{1-0.779}{1+0.779}) = -1.043$

$SE_r = \frac{1}{\sqrt{n-3}} = \frac{1}{\sqrt{8-3}} = 0.447$

$z_{xy} = \frac{z_r}{SE_r} = \frac{-1.043}{0.447} = -2.33$

4.2 e) The observed z-value is more extreme (lower) than the critical z-value. H_0 is rejected with 95% confidence. You can be 95% confident that ρ_{xy} is negative in the population. There is a 5% change of a type-I error.

4.3 a)

```
# Be sure to set your working directory when providing a relative path
dataset6 <- read.csv('localSupermarket.csv')
```

4.3 b)

```
Covariance:  -0.30
Correlation:  -0.30
```

```
cov(dataset6$Distance, dataset6$AvgVisits) # Covariance -0.3000603
cor(dataset6$Distance, dataset6$AvgVisits) # Correlation: -0.3000382
```

4.3 c)

```
cor.test(dataset6$Distance, dataset6$AvgVisits, alternative = 'less')
# Correlation: -0.30
# t-value: -9.936
# p-value: < 2.2e-16
```

4.3 d) The black line represents the normal distribution. The red line represents the t-distribution. The difference between the two distributions, in terms of their shape, is that the t-distribution has slightly thicker tails. When you increase the degrees of freedom of the t-distribution, it will start to look more like the normal distribution.

```
curve(dnorm(x, mean = 0, sd = 1), from = -3, to = 3, ylab = 'Density')
curve(dt(x, df = 3), from = -3, to = 3, add = TRUE, col = 'red')
```

4.3 e) $df = 998$ $t_{xy} = -9.936$

```
n <- nrow(dataset6)
dft <- n - 2 # 998

r <- cor(dataset6$Distance, dataset6$AvgVisits)
tscore <- r * sqrt(n - 2) / sqrt(1 - r^2)
# t-score: -9.936 so you can confirm the value in 4.3c
```

4.3 f) You can find the t-value in the bottom line of the output in the console.

4.3 g) The p-value is < 2.2e-16, which is lower than the significance level of 5%. This means that H_0 can be rejected with 95% confidence.

4.4 a)

```
# Be sure to set your working directory when providing a relative path
dataset7 <- read.csv('nationalSupermarket.csv')
```

4.4 b)

```
plot(x = dataset7$Price,
     y = dataset7$AvgWasted,
     main = 'Scatter plot of Price vs. AvgWasted',
     ylab = 'Average number of cartons wasted',
     xlab = 'Price of a carton of milk',
     las = 1,
     col = 'orange',
     pch = 19,
     bty = 'n')
```

4.4 c) $\texttt{AvgWasted} = \beta_0 + \beta_1 \times \texttt{Price}$

4.4 d)

```
lmfit <- lm(formula = AvgWasted ~ Price, data = dataset7)
```

4.4 e) $\texttt{AvgWasted} = 0.236 + 2.995 \times \texttt{Price}$

```
summary(lmfit)
# b0 = 0.236
# b1 = 2.995
# R-squared: 0.64
```

4.4 f)
```
abline(lmfit)
```

4.4 g) $R^2 = 0.64$

Interpretation: The multiple R^2 is 0.64, meaning that 64% of the variation in the number of milk cartons that are thrown away each day can be explained by the price of the milk cartons.

4.4 h) H_0: $\beta_1 \leq 0$ H_1: $\beta_1 > 0$

4.4 i) The p-value for the regression coefficient is < 2e-16, which is lower than the significance level of 5%. H_0 can be rejected with 95% confidence. You can be 95% sure that β_1 is positive in the population. The price contributes significantly to the average number of milk cartons thrown away. There is a 5% risk of a type-I error.

4.5 a)
```
newdata <- data.frame(Price = 0.70)
```

4.5 b) Predicted value: 2.33

```
predict(object = lmfit,
        newdata = newdata) # Prediction: 2.33
```

4.5 c) Predicted value: $0.236 + 2.995 \times 0.70 = 2.33$

4.5 d)
```
predict(object = lmfit, newdata = newdata,
        interval = 'prediction', level = 0.90)
# Lower bound: 0.734
# Upper bound: 3.931
```

4.5 e) The supermarket will throw away fewer cartons of milk.

Explanation: The current number of milk cartons thrown away (4) lies outside the bounds of the 90% confidence interval for the prediction.

Chapter 5: Comparing one or two means

5.1 a) $n = 49$
$\bar{x} = 0.918$
$s = 0.071$
$\mu_0 = 0.9$

5.1 b) H_0: $\mu_0 \leq 0.9$ H_1: $\mu_0 > 0.9$

5.1 c) Lower bound: $\bar{x} - z_\alpha \times SE_\mu = 0.918 - 1.645 \times \frac{0.071}{\sqrt{49}} = 0.901$

5.1 d) The lower bound of the confidence interval for μ is higher than μ_0. H_0 is rejected with 95% confidence. The PFAS level in town is significantly higher than 0.9 microgram/kg dry soil. There is a risk of 5% for a type-I error.

5.1 e) z-score: $\frac{\bar{x}-\mu}{s/\sqrt{n}} = \frac{0.918-0.9}{0.071/\sqrt{49}} = 1.775$

5.1 f) The critical z-value for 95% confidence is 1.645. The p-value (the probability that H_0 is true) is 0.05. For any z-value higher than 1.645 the p-value is lower than 0.05. Since for this test the (absolute of the) calculated z-score = 1.775 and this is higher than 1.645, this means that the p-value is lower than 0.05.

Note: For right sided tests the z-score is negative, but since the standard normal distribution is symmetric we can simply use the positive value for comparing it with the critical z-value.

5.1 g) The calculated z-score is more extreme (higher) than the critical z-value of 1.645. H_0 is rejected with 95% confidence. The PFAS level in town is significantly higher than 0.9 microgram/kg dry soil. There is a risk of 5% for a type-I error. The two methods produce the same answer. It cannot be different; the methods are equivalent.

5.1 h) t-score: $\frac{\bar{x}-\mu}{s/\sqrt{n}} = \frac{0.930-0.9}{0.070/\sqrt{16}} = 1.714$

5.1 i) The calculated t-score of 1.714 is lower than the critical t-value of 1.753. H_0 is not rejected. The PFAS level in town is significantly below the norm. There is a risk type-II error.

5.1 j) Lower bound: $\bar{x} - t_{\alpha(df=15)} \times SE_\mu = 0.930 - 1.753 \times \frac{0.070}{\sqrt{16}} = 0.899$

5.1 k) Yes. The lower bound for the population mean μ is 0.899, it can therefore not be ruled out that the mean PFAS level is below 0.9.

5.1 l) The sample is so much smaller (16 vs. 49) that there is too much uncertainty in the result.

5.2 a) Result: 1.645

This is the z-value for a 95% one-sided confidence interval.

```
qnorm(p = 0.95, mean = 0, sd = 1, lower.tail = TRUE)
# Or because the standard normal distribution is the default simply use:
qnorm(0.95) # 1.645
```

5.2 b) Because for a two-sided interval you spread the risk over the two tails. There is 2.5% of risk on the left tail and 2.5% of risk on the right tail.

```
qnorm(p = 0.975) # 1.960
```

5.2 c) The **pnorm()** is the inverse of the **qnorm()** function: It returns the cumulative probability for a given value **q** in a specified normal distribution.

```
pnorm(q = 1.645, mean = 0, sd = 1) # 0.95
pnorm(1.960) # 0.975
```

5.2 d) The **qt()** function returns the t-value for a given cumulative probability **p** and given number of degrees of freedom **df** .

```
qt(p = 0.95, df = 15) # 1.753
```

5.2 e) The **pt()** is the inverse of the **qt()** function: It returns the cumulative probability for a given value **q** in a specified t-distribution.

```
pt(q = 1.753, df = 15) # 0.95
```

5.2 f) t_1: -1.06 t_2: -1.328
t_3: 1.328 t_4: 1.328

```
qt(p = 0.15, df = 19) # -1.06
qt(p = 0.10, df = 19) # -1.328
qt(p = 0.90, df = 19) # 1.328
qt(p = 0.10, df = 19, lower.tail = FALSE) # 1.328
```

5.2 g) p_1: 0.081 p_2: 0.929 p_2: 0.015

```
pt(q = -1.5, df = 11) # 0.081
diff(pt(q = c(-2,2), df = 11)) # 0.929
pt(q = 2.5, df = 11, lower.tail = FALSE) # 0.015
```

5.2 h) Two tailed inequality test: 2.467
One-tailed right sided test: 2.153
One-tailed left sided test: -2.153

```
# Two-tailed inequality test
qt(p = 0.99, df = 28) # 2.467

# One-tailed right sided test
qt(p = 0.98, df = 28) # 2.153

# One-tailed left sided test
qt(p = 0.98, df = 28, lower.tail = FALSE) # -2.153
```

5.2 i) Two-tailed inequality test: H_0 rejected (2.6 > 2.476)
One-tailed right sided test: H_0 rejected (-2.3 < -2.153)
One-tailed left sided test: H_0 not rejected (1.6 < 2.153)

5.3 a) Because different men get the caffeine and the placebo. There are 18 unique test subjects. Everyone gets tested once and every measurement is therefore independent.

5.3 b) H_0: $\mu_1 \leq \mu_2$ H_1: $\mu_1 > \mu_2$

5.3 c) $\frac{2}{p} = \frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2)} = \frac{(9-1)\times 5.61^2+(9-1)\times 7.70^2}{9+9-2} = 45.38$

5.3 d) $t = \frac{(x_1-x_2)-D_0}{\sqrt{s_p^2(\frac{1}{n_1}+\frac{1}{n_2})}} = \frac{(94.22-100.56)}{\sqrt{45.38\times(\frac{1}{9}+\frac{1}{9})}} = 1.995$

5.3 e) The calculated t-score of 1.995 is more extreme (higher) than the critical value of 1.746. H_0 is rejected with 95% confidence. The mean placebo RER level is significantly higher than the mean caffeine RER level. There is a risk of 5% for type-I error.

5.3 f) **placebo**
Mean: 100.55
Standard deviation: 7.699

caffeine
Mean: 94.22
Standard deviation: 5.607

```
# These are the values for the RER test data set
placebo <- c(105, 119, 100, 97, 96, 101, 94, 95, 98)
caffeine <- c(96, 99, 94, 89, 96, 93, 88, 105, 88)


mean(placebo) # Mean: 100.55
sd(placebo) # Standard deviation: 7.699

mean(caffeine) # Mean: 94.22
sd(caffeine) # Standard deviation: 5.607
```

5.3 g) The code runs an independent (not paired) samples t-test with a confidence of 95% for **placebo** and **caffeine** , with the alternative hypothesis H_1 that the difference in means is greater than 0. The variances are assumed equal.

```
t.test(x = placebo, y = caffeine, alternative = 'greater', mu = 0,
       paired = FALSE, var.equal = TRUE, conf.level = 0.95)
```

5.3 h) The Welch Two Sample t-test leads to a slightly different p-value of 0.03252, but the same conclusion: rejection of H_0.

```
t.test(x = placebo, y = caffeine, alternative = 'greater', mu = 0,
       paired = FALSE, var.equal = FALSE, conf.level = 0.95)
```

5.3 i) You can use Hartley's F or Levene's test.

5.4 a) The observations are not independent because the same twelve people are tested twice. The two blood pressure measurements for one person are connected/dependent: a person with high blood pressure will have higher values in both experiments. That is why in a dependent t-test you look at the difference between the two measurements.

5.4 b) Mean standing: 140.83
Mean lying: 143.33
Mean difference: 2.5

```
# These are the values for the blood pressure data set
standing <- c(132, 146, 135, 141, 139, 162, 128, 137, 145, 151, 131, 143)
lying <- c(136, 145, 140, 147, 142, 160, 137, 136, 149, 158, 120, 150)

mean(standing) # Mean standing: 140.83
mean(lying) # Mean lying: 143.33
differences <- lying - standing
mean(differences) # Mean difference: 2.5
```

5.4 c) H_0: $\mu_D \leq 0$ H_1: $\mu_D > 0$

5.4 d) The test is done with `alternative = 'greater'`, which means that now R will test for standing greater than lying, which is quite improbable given the sample results.

```
t.test(x = standing, y = lying, alternative = 'greater', mu = 0,
       paired = TRUE, conf.level = 0.925)
```

5.4 e) The p-value for this sample outcome is 0.07189, which is below the limit of 0.075 (92.5% confidence). H_0 is rejected with 92.5% confidence. The blood pressure is significantly higher lying down than standing up. There is a risk of 7.5% for type-I error.

```
# Correct: alternative = 'less'
t.test(x = standing, y = lying, alternative = 'less', mu = 0,
       paired = TRUE, conf.level = 0.925)
```

Chapter 6: Comparing more than two means

6.1 a)

```
# Be sure to set your working directory when providing a relative path
dataset8 <- read.csv('eyeColor.csv')

ttestData <- subset(dataset8,
                    dataset8$Group == 'Blue'|
                    dataset8$Group == 'Brown')
```

6.1 b) H_0: $\mu_1 = \mu_2$ H_1: $\mu_1 \neq \mu_2$

6.1 c)

```
blue <- subset(ttestData$Score, ttestData$Group == 'Blue')
brown <- subset(ttestData$Score, ttestData$Group == 'Brown')
t.test(blue, brown, var.equal = TRUE) # p-value: 0.1401
```

6.1 d) The p-value is 0.1401, which is higher than the 0.05 required to reject H_0. H_0 is not rejected with 95% confidence. You can be 95% confident that the mean of the blue group is the same as the mean of the brown group. There is a risk of a type-II error.

6.1 e) The content of `dummyBrown` is a 0 for blue eyes, and a 1 for brown eyes. This kind of variable is called a dummy variable.

```
dummyBrown <- as.numeric(ttestData$Group == 'Brown')
ttestData <- cbind(ttestData, dummyBrown)
```

6.1 f)

```
ttestreg <- lm(formula = Score ~ dummyBrown, data = ttestData)
```

6.1 g)

```
summary(ttestreg) # p-value dummyBrown: 0.14
```

6.2 a) H_0: $\mu_1 = \mu_2 = \mu_3 = \mu_4$ H_1: $\mu_1 \neq \mu_2 \neq \mu_3 \neq \mu_4$

6.2 b) $df_M = k - 1 = 4 - 1 = 3$ $df_R = n - k = 222 - 4 = 218$

6.2 c) Critical F-value: 2.646

```
df1 <- 4 - 1 # 3
df2 <- nrow(dataset8) - 4 # 218
qf(p = 0.95, df1 = df1, df2 = df2) # 2.646
```

6.2 d)

```
anovaResult <- aov(formula = Score ~ Group, data = dataset8)
```

6.2 e) F-value: 2.894 p-value: 0.0362

```
summary(anovaResult)
# F-value: 2.894
# p-value: 0.0362
```

6.2 f) The p-value is 0.0362, which is lower than the 0.05 required to reject H_0. The sample F-value is 2.894, which is higher than the critical F-value of 2.646. H_0 is rejected with 95% confidence. The means of the four groups are not equal to each other. There is a risk of 5% for a type-I error.

6.3 a)

```
dummyBrown <- as.numeric(dataset8$Group == 'Brown') # Brown eyes
dataset8 <- cbind(dataset8, dummyBrown)

dummyBlue <- as.numeric(dataset8$Group == 'Blue') # Blue eyes
dataset8 <- cbind(dataset8, dummyBlue)

dummyGreen <- as.numeric(dataset8$Group == 'Green') # Green eyes
dataset8 <- cbind(dataset8, dummyGreen)
```

6.3 b) anovaReg <- lm(formula = Score ~ dummyBrown + dummyBlue + dummyGreen, data = dataset8)

6.3 c) F-value: 2.894 p-value: 0.0362

```
summary(anovaReg)
# F-value: 2.894
# p-value: 0.0362
# R-squared: 0.0383
```

6.3 d) Yes, the results are the same.

6.4 a)

```
ancovaReg <- lm(formula = Score ~ dummyBrown + dummyBlue +
                                  dummyGreen + initialScore,
                data = dataset8)
```

6.4 b) F-value: 2.252 p-value: 0.064

```
summary(ancovaReg)
# F-value: 2.252
# p-value: 0.064
# R-squared: 0.0398
```

6.4 c) The p-value is 0.0645, which is higher than the 0.05 required to reject H_0. H_0 is not rejected with 95% confidence. The means are equal to each other if you consider the initial score as a covariate. There is a 5% change of a type-I error.

6.4 d) The p-value is of the coefficient of initial score is 0.5539, which means that it is not significantly different from zero. This means that the initial score is not a good predictor of the actual score.

6.4 e) R^2 **anovaReg** : 0.0383 R^2 **ancovaReg** : 0.0398

The **ancovaReg** regression model explains more variation in the outcome variable score.

6.4 f) The groups, together with the initial score, explain 3.98% of the variance in the dependent variable score.

6.4 g) AIC **anovaReg** : 865.54 AIC **ancovaReg** : 867.18

```
AIC(anovaReg) # AIC: 865.54
AIC(ancovaReg) # AIC: 867.18
```

6.4 h) The AIC value of the **anovaReg** regression model is the lowest, which means that the **anovaReg** model fits the data better than the **ancovaReg** regression model. This means that the model without the covariate is a better model. You may have already seen this, since the covariate in the **ancovaReg** model was not a good predictor of the score.

6.5 a)

```
#  Be sure to set your working directory when providing a relative path
load('iowa.RData')
```

6.5 b) The **iowa** data consists of payments made by the state of Iowa. Payments are assigned to fiscal years that run from July 1 through June 30, and are numbered for the calendar year in which they end. The fiscal year is divided into fiscal periods with 1 being July and 12 being June. The fiscal year also includes a hold-over period for payments made after year end for good and services received on or before June 30.

6.5 c) Rows: 12279009 Columns: 22

```
nrow(iowa) # 12279009 rows
ncol(iowa) # 22 columns
```

6.5 d) Unique services: 8

```
unique(iowa$Service) # 8 unique services
table(iowa$Service)
```

6.5 e) Service: Human Services Rows: 6682159

6.5 f) Number of rows that show a difference: 2624607

```
iowa$Payment.Issue.Date <- as.Date(iowa$Payment.Issue.Date,format= '%m/%d/%Y')
iowa$Invoice.Date <- as.Date(iowa$Invoice.Date,format = '%m/%d/%Y')

length(which(iowa$Payment.Issue.Date != iowa$Invoice.Date)) # 2624607
```

6.5 g)

```
dataDif <- data[which(iowa$Payment.Issue.Date != data$Invoice.Date),]
```

6.5 h)

```
dataDif$dif.days <- dataDif$Payment.Issue.Date - dataDif$Invoice.Date
dataDif$dif.days <- as.numeric(dataDif$dif.days)
```

6.5 i)
```
Minimum:  -3651                    Upper quartile:  33
Mean:  21.815                      Lower quartile:  4
Maximum:  36529                    Standard deviation:  89.24
```

```
min(dataDif$dif.days)
max(dataDif$dif.days)
mean(dataDif$dif.days)
quantile(dataDif$dif.days)
sd(dataDif$dif.days)
```

6.5 j) The default histogram does not provide much information due to the fact that R specifies a very wide *x-axis*.

```
hist(dataDif$dif.days)
```

6.5 k)
```
hist(dataDif$dif.days[dataDif$dif.days > quantile(dataDif$dif.days, 0.05) &
     dataDif$dif.days < quantile(dataDif$dif.days, 0.95)], breaks = 100)
```

6.5 l)
```
dataDif2 <- dataDif[(dataDif$dif.days > (-1)) & (dataDif$dif.days <= 365),]
```

6.5 m)
```
plot(dataDif2$Amount,dataDif2$dif.days)
```

6.5 n) Correlation: -0.0025

```
cor.test(dataDif2$Amount,dataDif2$dif.days)
```

6.5 o) The p-value of the correlation test against the value zero is 3.148e-05, which is sufficient enough to reject H_0 with 95% confidence. This implies that there is, with 95% certainty, a correlation between the the time between invoice and payment, and the amount that is paid.

6.5 p) Administration and regulation: 26.452
Agriculture and natural resources: 28.275
Capital: 35.490
Economic development: 25.815
Education: 23.868
Human services: 18.508
Justice system: 25.478

```
aggregate(dataDif2$dif.days, by = list(dataDif2$Service), FUN = mean)
```

6.5 q) p-value: < 2e-16

Conclusion: The p-value is lower than 0.05, so you can reject H_0 with 95% confidence. This means that the means of all expense categories are not equal. There is a 5% type change of a type-I error.

```
aovRes <- aov(dif.days ~ Service, data = dataDif2)
summary(aovRes)
```

6.5 r) All means, except for the means of the justice system expenses and the economic development expenses, show a p-value below 5% and can thus be regarded to differ from each other.

```
tukeyRes <- TukeyHSD(aovRes)
```

6.5 s) An ANOVA assumes the dependent variable to be continuous. Some examples of appropriate analyses could be:

```
# Poisson regression and then interpret the predictors
poissonReg <- glm(dif.days ~ Expense.Category, family = poisson, data = dataDif2)
summary(poissonReg)

# Kruskal Wallis test and Dunn test to compare individual groups
kruskRes <- kruskal.test(dif.days ~ Service, data = dataDif2)
# install.packages('FSA'); library(FSA)
dunn.res <-dunnTest(dataDif2$dif.days, dataDif2$Service)
```

6.5 t) It is a bad idea, the p-value is affected by the number of samples. The higher the sample, the lower the p-value gets. In other words, p-values lose their meaning quite quickly (unless they are non-significant).

Chapter 7: Comparing proportions and distributions

7.1 a) H_0: The 2019 distribution is equal to the historical distribution
H_1: The 2019 distribution is not equal to the historical distribution

7.1 b) Expected number = Historical % × Observed number

	Historical	Observed (O)	Expected (E)	$O - E$	$\frac{(O-E)^2}{E}$
Spring	4.87 (30%)	2.90	25.2	11.8	5.525
Summer	3.04 (40%)	4.50	33.6	-6.6	1.296
Fall	1.65 (15%)	4.94	12.6	-0.6	0.029
Winter	2.88 (15%)	3.28	12.6	-4.6	1.679
Total	2.31 (100%)	4.73	84		8.530

7.1 c) Since every season has an expected value above 5 you can do a chi-square test.

7.1 d) $X^2 = 8.530$

7.1 e) The calculated chi-square value of 8.530 is higher than the critical chi-square value of 7.8. H_0 is rejected. The 2019 distribution is significantly different from the historical distribution. There is a 5% risk of a type-I error.

7.1 f) You rejected the null hypothesis H_0 with 95% confidence, and so the p-value must be lower than 0.05.

7.1 g) $X^2 = 8.530$

```
Observed <- c(37, 27, 12, 8)
Historical <- c(0.3, 0.4, 0.15, 0.15)
Expected <- c(25.2, 33.6, 12.6, 12.6)

chi <- sum((Observed - Expected)^2 / Expected) # 8.530
```

7.1 h) Yes.

```
qchisq(p = 0.95, df = 3) # 7.185
```

7.1 i) R returns the same chi-squared as calculated, so the answer was correct. It also shows a p-value of below 0.05, as expected.

```
# chi-square test: x = observations p = model distribution
# rescale makes sure the model distribution adds up to 100%
chisq.test(x = Observed, p = Historical, rescale.p = TRUE)

# Chi-squared value: 8.5298
# p-value: 0.0362
```

7.1 j) The expected values are 25.2, 33.6, 12.6, and 12.6. R shows the same expected values.

```
chisq <- chisq.test(x = Observed, p = Historical)
chisq$expected # Extract expected values with $expected
# 25.2 33.6 12.6 12.6
```

7.2 a) The **sales** data frame contains 3 columns: **month**, **historical** and **newstore**. It contains the 'Historical' and 'New Store' distribution of sales over the months.

```
# These are the values for the sales data set
sales <- data.frame(month = seq(from = 1, to = 12, by = 1),
                    historical = c(5.1, 5.1, 6.7, 10, 11.4, 10,
                                   6.7, 5.1, 6.7, 10, 11.7, 11.7),
                    newstore = c(5.6, 6.2, 9.4, 8.6, 6.8, 4.8,
                                 5.6, 4.8, 8.8, 12.6, 13.1, 13.7))

summary(sales)
```

7.2 b) The chi-square test requires every cell in the expected distribution to have a value of at least 5 and it requires the total of both groups to be equal. Since the historical distribution contains more than 5 in every cell and both observed and expected values add up to the same number (100) we can use this for a chi-squared test.

7.2 c) H_0: The new distribution is equal to the historical distribution
H_1: The new distribution is not equal to the historical distribution

7.2 d) The p-value of 0.6963 is higher than the critical p-value of 0.10. H_0 is not rejected. The new store distribution is not significantly different from the historical distribution. There is a risk a type-II error.

```
chisq.test(x = sales$newstore, p = sales$historical, rescale.p = TRUE)
# p-value: 0.6963
```

7.3 a) The best estimate of the population proportion π is the sample proportion p.

$\pi_1 = \frac{k}{n} = \frac{8}{71} = 0.113$

$\pi_2 = \frac{k}{n} = \frac{16}{111} = 0.144$

7.3 b) Confidence interval sample 1:

$p \pm z_\alpha \times \sqrt{\frac{p(1-p)}{n}} = 0.113 \pm 1.960 \times \sqrt{\frac{0.113 \times (1-0.113)}{71}} = [0.039;\ 0.187]$

Confidence interval sample 2:

$p \pm z_\alpha \times \sqrt{\frac{p(1-p)}{n}} = 0.144 \pm 1.960 \times \sqrt{\frac{0.144 \times (1-0.144)}{111}} = [0.078;\ 0.210]$

7.3 c) H_0: $\pi_2 \leq \pi_1$ H_1: $\pi_2 > \pi_1$

Where π_2 and π_1 are the success proportions for the evening and afternoon calls respectively.

7.3 d) Combined success probability:

$p^* = \frac{k_1 + k_2}{n_1 + n_2} = \frac{8+16}{71+111} = 0.132$

7.3 e) Combined standard error:

$s_p = \sqrt{p^*(1-p^*)(\frac{1}{n_1} + \frac{1}{n_2})} = \sqrt{0.132(1-0.132)(\frac{1}{71} + \frac{1}{111})} = 0.0514$

7.3 f) z-score: $\frac{p_1 - p_2}{s_p} = \frac{0.144 - 0.113}{0.514} = 0.612$

Note that p_1 and p_2 were switched because in the hypotheses π_1 and π_2 were also switched.

7.3 g) The calculated z-score of 0.612 is lower than than the critical z-value of 1.645. H_0 is not rejected. The evening success rate is not shown to be significantly higher than the afternoon success rate. There is a risk of a type-II error.

7.3 h) The code creates a vector of successes `k` and a vector of sample sizes `n`. The proportion test then tests the equality. It shows the proportions you calculated earlier, and a p-value of 0.6984 which supports your conclusion if you do not reject H_0.

```
n <- c(71, 111)
k <- c(8, 16)
prop.test(x = k, n = n)
# p-value: 0.6984
```

Note that R actually returns a chi-squared value. Because it actually does a chi-square test it can in fact be used to test more than two proportions at the same time.

Chapter 8: Bayesian statistics

8.1 a) The prior distribution on the left of the figure assigns equal mass to every value of θ, and therefore does not incorporate relevant information about which values of θ are more likely to occur. The prior distribution in the middle reflects the belief that the probability of heads is either very close to zero, or very close to one. The prior distribution on the right reflects the information that it is most likely that the coin is a fair coin (highest probability at 0.5), but that there is some uncertainty about θ around this value.

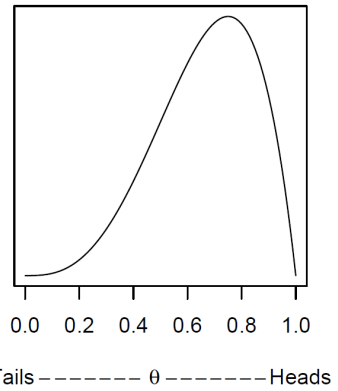
8.1 b)

Middle:		Right	
α:	0.5	α:	2
β:	0.5	β:	2

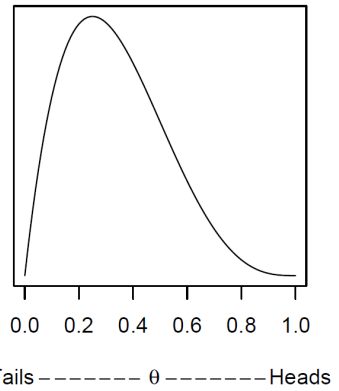
```
alpha <- 0.5
beta <- 0.5
curve(dbeta(x, alpha, beta), xlab = expression(theta), ylab = '', yaxt = 'n')

alpha <- 2
beta <- 2
curve(dbeta(x, alpha, beta), xlab = expression(theta), ylab = '', yaxt = 'n')
```

8.1 c)

Biased towards heads:



Biased towards tails:



8.2 a) The Greek letter θ represents the proportion of customers that leaves the store feeling satisfied in this case.

8.2 b) An example of prior parameters can be $\alpha = 7$ and $\beta = 5$. This combination yields a prior distribution that assigns most mass at the point $\theta = 0.6$, but expresses relatively much uncertainty about θ. However, other values of the parameters can also be correct, since the prior reflects your own beliefs. Higher values of α and β give prior distributions that are less wide, and this reflect more certain prior beliefs.

8.2 c) The answer depends on your own choices of α and β in assignment 8.2b. In the following answers, $alpha = 7$ and $beta = 5$ are used.

```
curve(dbeta(x, shape1 = 7, shape2 = 5),
      xlab = expression(theta), ylab = '', yaxt = 'n')
```

8.2 d) α: $7 + 33 = 40$ β: $5 + 40 - 33 = 12$

8.2 e) Probability: 0.995

```
pbeta(0.6, shape1 = 40, shape2 = 12, lower.tail = FALSE) # 0.995
```

8.2 f) Probability: 0.875

```
diff(pbeta(c(0.7, 0.9), shape1 = 40, shape2 = 12)) # 0.875
```

8.2 g) The posterior distribution is fairly robust to changes in the prior distribution. In Bayesian inference, the data quickly overwhelm the prior distribution when it comes to parameter estimation. Therefore, the more data you have the more robust the posterior distribution generally is.

8.3 a)

```
samples <- extract(stanFit)
```

8.3 b) The histogram approximates the analytical posterior pretty well. The more samples you draw, the better the histogram will resemble the actual posterior distribution.

```
hist(samples, breaks = 100, probability = TRUE)
curve(dbeta(x, shape1 = 40, shape2 = 12), add = TRUE)
```

8.3 c)

```
modelCode <- '
data {
    int n;
    int k;
}
parameters {
    real<lower=0,upper=1> theta;
}
model {
    theta ~ normal(0.6, 0.1);
    k ~ binomial(n, theta);
}
'

# Note: The following line can take a while to execute
compiledModel <- stan_model(model_code = modelCode, model_name = 'model')

stanFit <- sampling(compiledModel, data = list(n = 40, k = 33),
                    iter = 5000, warmup = 500, chains = 4)
```

8.3 d) Probability: 0.990

```
length(which(samples$theta > 0.6)) / length(samples$theta) # 0.990
```

8.3 e) Probability: 0.796

```
length(which(samples$theta >= 0.7 & samples$theta <= 0.9)) /
length(samples$theta) # 0.796
```

8.3 f) The answers of assignments 8.3d and 8.3e do not differ substantially from those of assignments 8.2e and 8.2f. Overall you can conclude that your posterior distribution is fairly robust to changes in the prior distribution. You can therefore be highly sure that the percentage of satisfied customers is higher than 60 percent, and reasonably sure that the percentage lies between 70 and 90 percent.

8.4 a) H_0: $\theta = 0.5$ H_0: $\theta \neq 0.5$

8.4 b) An example of a possible answer can be $\alpha = 3$ and $\beta = 3$. This prior distribution is fairly wide, and thus expresses not much confidence in your programming skills.

$\theta \sim Beta(3, 3)$

8.4 c) Assuming the values $\alpha = 2$ and $\beta = 2$:

α: `3 + 22 = 25` β: `3 + 30 - 22 = 11`

8.4 d)

```
alpha <- 3
beta <- 3
n <- 30
k <- 22

# Plot the posterior distribution first so no adjustment of the axes is
needed curve(dbeta(x, alpha + k, beta + n - k),
      bty = 'n', las = 1, lty = 1, yaxt = 'n', ylab = '', xlab = expression(theta))
curve(dbeta(x, alpha, beta), add = TRUE, lty = 2)
```

8.4 e)

```
heightPrior <- dbeta(0.5, alpha, beta)
```

8.4 f)

```
heightPosterior <- dbeta(0.5, alpha + k, beta + n - k)
```

8.4 g) The value of the Bayes factor `BF10` is 7.02. This implies that the data are 7.18 times more likely to occur under the alternative hypothesis $H_1 : \theta \neq 0.5$ than under the null hypothesis $H_0 : \theta = 0.5$.

8.4 h) The strength of the evidence is moderate, and so you cannot be very confident that your algorithm performs better than change. You will need to collect more data if your want a more decisive Bayes factor.

8.5 a) M_1: $\theta \leq 0.75$ M_2: $\theta \geq 0.75$

8.5 b)

```
model2code <- '
data {
    int n;
    int k;
}
parameters {
    real<lower=0.75,upper=1> theta;
}
model {
    theta ~ beta(1, 1)T[0.75, 1];
    k ~ binomial(n, theta);
}
'

# Note: The following line can take a while to execute
model2 <- stan_model(model_code = model2code, model_name = 'model2')
```

8.5 c) The prior distribution for θ in M_1 is a $Beta(1, 1)$ prior distribution truncated to the [0, 0.75] interval (`beta(1, 1)T[0, 0.75]`). The prior distribution for θ in M_2 is a $Beta(1, 1)$ prior distribution truncated to the [0.75, 1] interval (`beta(1, 1)T[0.75, 1]`).

8.5 d)

```
stanFitM1 <- sampling(model1, data = list(n = 156, k = 123),
                      iter = 5000, warmup = 500, chains = 4)

stanFitM2 <- sampling(model2, data = list(n = 156, k = 123),
                      iter = 5000, warmup = 500, chains = 4)
```

8.5 e)

```
mLike1 <- bridge_sampler(stanFitM1)
mLike2 <- bridge_sampler(stanFitM2)
```

8.5 f)

```
BF12 <- bf(mLike2, mLike1)
```

8.5 g) The value of `BF12` is around 17.77. This indicates that the data is 17.77 times more likely to occur under model M_2 : $\theta \geq 0.75$ than under model $M_2 : \theta \leq 0.75$.

8.5 h) The evidence in favor of M_2 is moderate, and so additional testing might be required to come to a reasonable conclusion.

8.6 a)

```
# Be sure to set your working directory when providing a relative path
insurance <- read.csv('insurance.csv')
```

8.6 b) $\texttt{charges} = \beta_0 + \beta_1 \times \texttt{age} + \beta_2 \times \texttt{bmi} + \beta_3 \times \texttt{neighbors}$

8.6 c) β_0: -6670.25 β_1: 333.92 β_2: 241.56 β_3: 45.84

```
regressionFit <- sampling(object = regressionModel,
                          data = list(N = nrow(insurance),
                                      x1 = insurance$bmi,
                                      x2 = insurance$age,
                                      x3 = insurance$neighbors,
                                      y = insurance$charges),
                          iter = 5000, warmup = 500, chains = 4)

regressionSamples <- extract(regressionFit)
summary(regressionSamples)
```

8.6 d) YES

```
summary(lm(charges ~ 1 + age + bmi + neighbors, data = insurance))
```

8.6 e)

```
layout(matrix(1:4, nrow = 2))

plot(density(regressionSamples$beta1), xlab = '', ylab = '',
     yaxt = 'n', bty = 'n', main = expression(beta[1]))
plot(density(regressionSamples$beta0), xlab = '', ylab = '',
     yaxt = 'n', bty = 'n', main = expression(beta[0]))
plot(density(regressionSamples$beta2), xlab = '', ylab = '',
     yaxt = 'n', bty = 'n', main = expression(beta[2]))
plot(density(regressionSamples$beta3), xlab = '', ylab = '',
     yaxt = 'n', bty = 'n', main = expression(beta[3]))

layout(1)
```

8.6 f) Probability: 0.966

```
length(which(regressionSamples$beta2 > 200)) / length(regressionSamples$beta2)
```

8.6 g) `Probability: 0.052`

```
length(which(regressionSamples$beta1 < 250)) / length(regressionSamples$beta1)
```

8.6 h) `Lower bound: -79.98` `Upper bound: 169.74`

```
quantile(regressionSamples$beta3, probs = c(0.10, 0.90))
```

8.6 i)

```
regressionmodelcode2 <- '
data {
    int<lower=0> N;
    vector[N] x1;
    vector[N] x2;
    vector[N] y;
}
parameters {
    real beta0;
    real beta1;
    real beta2;
    real<lower=0> sigma;
}
model {
    y ~ normal(beta0 + beta1 * x1 + beta2 * x2, sigma);
}
'

# Note: The following line can take a while to execute
regressionModel2 <- stan_model(model_code = regressionmodelcode2,
                               model_name = 'Regression2')
```

8.6 j)

```
regressionFit2 <- sampling(object = regressionModel2,
                                data = list(N = nrow(insurance),
                                            x1 = insurance$bmi,
                                            x2 = insurance$age,
                                            y = insurance$charges),
                                iter = 5000, warmup = 500, chains = 4)

mLike1 <- bridge_sampler(regressionFit)
mLike2 <- bridge_sampler(regressionFit2)

BF12 <- bf(mLike1, mLike2)
```

8.6 k) The value of **BF12** is 258.65. This implies that the data are 258.65 times as likely to be observed under the model where β_3 is not included in the regression equation under the model where β_3 is included. The evidence against the number of **neighbors** adding to the prediction accuracy is extreme.

Beginner R exercises

B 1) The code created two variables (**a** and **b**) in your environment. The value of variable **a** is 2 (although at first it is specified as 1) and the value of variable **b** is 1. It does not matter whether variables are assigned with the **=** or the **<-** operator.

B 2)
```
rm(a); rm(b)
```

B 3) Adding up **apples** and **pears** equals 11.
The variable **Pears** does not exist. Instead, you should add **apples** and **pears** together (notice the lack of a capital P).

B 4)
```
t1 <- sqrt(81)
```

B 5)
```
t2 <- 81^0.5
```

B 6)
```
t1 == t2
```

B 7) The working directory is a file path on your computer that sets the default location of any files you read into R, or save out of R. In other words, a working directory is like a little flag somewhere on your computer which is tied to a specific analysis project. If you ask R to import a dataset from a text file, or save a data frame as a text file, it will assume that the file is inside of your working directory.

B 8) You can change the working directory using the **setwd()** function. You can check whether your change worked by calling the **getwd()** function again and checking whether it is now at the folder you specified in **setwd()** . Note that RStudio also has the option to change the working directory from the "Session" dropdown menu, then select "Set Working Directory".

B 9) The code shows all the available demos that are built into R. The **persp** demo can be viewed with:

```
demo(persp)
```

B 10) It gives `NA` because the `mean()` function does not handle `NA`'s by default. You can compute the mean without the missing value by setting `na.rm = TRUE`.

B 11) Just typing `mvrnorm()` gives an error because the package `MASS` (from which the function comes) is not loaded yet. You can load the `MASS` package by typing:

```
library(MASS)
```

B 12) The `%%` operator gives the modulo (the remaining number after division) of the first number divided by the last number. You cannot find help by typing `?%%`, but you can search Google for help on how to use this operator. Note that in RStudio you can also type `%%` into the search bar in the Help section (in the Files and Plots part of RStudio, usually displayed at the bottom right of your screen).

B 13) You can test the correlation and find the confidence interval for the correlation by using the `cor.test()` function. You can find all function that have 'cor' in their name by typing `apropos('cor')`.

```
cor.test(c(1, 2, 3, 4), c(1, 4, 7, 15))
```

B 14) The modes of `a1`, `a2`, and `a3` are character, numeric, and logical respectively. The modes of `b1`, `b2`, `b3`, and `b4` are character, character, numeric, and character respectively. When vectors of different modes are combined R converts the vectors to one and the same mode, because elements in a vector can only be of one mode.

B 15) As a standard, R converts `TRUE` to a 1 and `FALSE` to a 0.
$1/1 = 1$ $0/1 = 1$ $0/0 =$ undefined $1/0 = \infty$

B 16) In the first case the numeric `1` is converted to character mode, while in the last case the logical `TRUE` is converted to character mode.

```
as.numeric(TRUE)
as.character(TRUE)
```

B 17) You can check whether a vector is numeric by using the `is.numeric()` function. `c(1, 0)` is numeric; `c(TRUE, FALSE)` is not; `c(TRUE, FALSE, 1, 0)` is numeric because R automatically converts `TRUE` and `FALSE` to a `1` and a `0` because vectors have to be of one mode.

B 18) The length of this vector is 11.

```
v1 <- c(-2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8)
# or
v1 <- -2:8
length(v1)
```

B 19)

```
v2 <- seq(from = -5, to = 5, by = 0.5)
v2 <- (-10:10)/2
```

B 20)

```
v3 <- seq(from = 13, to = 33, by = 2)
```

B 21)

```
c(a, b)
```

B 22) The result of `rep(2, 10)` is a repeated vector containing 10 times the number 2. The function argument `times = 10` is implicitly set.

B 23)

```
rep(3:7, each = 3)
```

B 24) You have to explicitly specify that the 2 refers to the `each` argument. By default, the second argument of the `rep()` function is times, see also `?rep` .

B 25) The sum of these numbers is 100.

```
sum((1:10)*2-1)
```

B 26)

```
v3 <- (1:10) * 2
v4 <- v3 / 5
v5 <- seq(5, 32, 3)
```

B 27)
```
vector('logical', 5)
```

B 28)
```
paste(rep(c('x', 'y'), each = 4), rep(1:2, each = 2), c('m', 'f'), sep = '')
```

B 29) The options `decreasing = TRUE` can be set to sort the numbers from highest to lowest. The default is `FALSE`, so sorting from lowest to highest does not require specification of this option.

```
sort(s, decreasing = TRUE) # Highest to lowest
sort(s) # Lowest to highest
```

B 30) R is actually pretty smart and will recycle values for you. 10 is not divisible by 3, hence full recycling of all numbers is not possible.

B 31)
```
matrix(25:1, nrow = 5, byrow = TRUE)
```

B 32)
```
matrix(rep(0:1, 8), nrow = 4)
```

B 33) You do not have to specify the number of rows (only the number of columns) because, given the number of values and the number of columns, the number of rows for the matrix is known.

B 34) You can transpose a matrix using the `t()` function.

B 35)
```
colMeans(m1)
```

B 36) You can select the diagonal of a matrix using the `diag()` function.

```
m2 <- matrix(rep(0, 16), nrow = 4)
diag(m2) <- diag(m1)

# or:
m2 <- diag(diag(m1))
```

B 37)
```
m1 <- rbind(m1, rowSums(m1))
```

B 38) The means of the columns are all zero (calculate with rowSums(m1)), and the standard deviations are all one (calculate with apply(m2,2,sd). The **scale()** function therefore transforms the values in the columns so that their mean is zero and their standard deviation is one.

B 39)
```
data.frame(subject = rep(1:5, each = 2),
           time = c('t1', 't2'),
           score = c(7, 8, 8, 8, 9, 8, 9, 7, 7, 6))
```

B 40)
```
as.data.frame(a)
```

B 41)
```
list(subject = 1:5,
       time = c('t1', 't2'),
     score = matrix(1:25,5,5))
```

B 42) No, **g** is not a true vector but a factor. That is because **g** consist of factor levels. You can check this using **is.vector(g)** or **is.factor(g)** . Note that you can also find this out by using the 'str' function: str(g).

B 43)
```
v <- as.factor(v)
```

B 44) The factor **x** has three levels, which you can find out using the **levels()** function.

```
x <- as.factor(x)
levels(x)
```

B 45) The value of `x[2]` is `NA`.

B 46)
```
a[1] <- 8
a[a == 2] <- 0
```

B 47)
```
l[2, 3]
```

B 48)
```
l[c(3, 5), ]
```

B 49)
```
m[m[, 1] < 5, ]
```

B 50)
```
m[ , -4]
```

B 51)
```
m <- as.data.frame(m)
colnames(m) <- paste('trial', 1:10, sep = '.')
```

B 52)
```
m$'trial.1'[2:5]
m$'trial.4'[1:2]
```

B 53)
```
b <- ifelse(b == 1, yes = 2, no = 1)
```

B 54)
```
gsub('a', '.', n)
```

B 55)
```
passed <- grades[rowMeans(grades[, c('exer','exam')]) > 5.5 &
                 grades[, 'exer'] >= 5 &
                 grades[, 'exam'] >= 5 , 1]
failed <- grades$student[-passed]
```

B 56) The condition `(a < 0 | b < 0) (a * b < 0)` means that `a` is smaller than `0` or `b` is smaller than `0`, and `a * b` is smaller than `0`. If this condition is `TRUE` and `a` is negative, then `b` must be positive.

B 57) The number of `NA`'s in the vector is 1, hence it is smaller than or equal to 2. So, `sum(is.na(c(1, 2, 3, 4, NA, 7))) <= 2` returns `TRUE`, and its logical negation (`!`) is therefore `FALSE`.

B 58) These are the numbers by which the number 24 can be divided (24 is divisible by 1, 2, 3, 4, 6, 8, 12, and 24).

B 59) This code tests whether two vectors are identical. The `identical()` function does this automatically.

```
min(x==y) == 1
mean(x==y) == 1
prod(x==y) == 1
identical(x, y)
```

B 60) You can use the `table()` function to create a frequency table of the throws.

B 61)
```
sample(c('jack', 'queen', 'king', 'ace'), replace = TRUE)
```

B 62) You can use the `runif()` function to sample uniform random numbers. `sample(1:100, 20)` is not correct in this case because the sampling vector starts at 1.

```
runif(n = 20, min = 0, max = 100)
```

B 63)
```
r1 <- runif(n = 21, min = 0, max = 100)
median(r1)
r2 <- sort(r1, decreasing = TRUE)
r2[11]
```

B 64) Simulating data using the same code twice does not result in the same samples because of the inherent randomness of a simulation. You can use the **set.seed()** function to make your code reproducible.

```
set.seed(123)
r3 <- rnorm(n = 100, mean = 100, sd = 15)
var(v3)
```

B 65)

```
cov(x)
cor(x)
```

B 66) The **read.csv()** function reads in **.csv** files. You can read in Excel files (**.xlsx**) using the **read.xslx()** function (from the **xlsx** package). The function **read.spss()** is featured in the package **foreign** (see **?read.spss**) and reads **.spss** files.

B 67)

```
# Be sure to set your working directory when providing a relative path
dataset <- read.csv('example.csv')
```

B 68)

```
# Be sure to set your working directory when providing a relative path
write.csv(d, file = 'example.csv')
```

The subject of statistics involves the study of how to collect, analyze and interpret data. These data and the information it conveys are at the basis of most business decisions. Given that decisions are usually taken in order to influence the future, one can only presume that the collected data represent the future in some way. In this workbook, we teach you how statistical inference can deal with quantifying the uncertainty that is induced by using data to make decisions.

R is a powerful coding language that is ideally suited for doing statistics. Though R may be difficult to understand at first, this workbook will guide you through gaining an understanding of how to use R for estimation and hypothesis testing in the most common business research designs.