Exercises for Day 1

Used datasets and R scripts can be downloaded in a ZIP archive from the Absalon page (Applied Statistics) or from

`http://www.math.ku.dk/~pdq668/SmB/material/day1.zip`

Exercise 1.1: *Significance vs. importance.*

In the lecture I propose the following four main questions[1] to be answered by the statistical analysis of a dataset:

1. Is there an effect?
2. Where is the effect?
3. What is the effect?
4. Can the conclusions be trusted?

The founder of modern statistics R.A. Fisher once wrote:

> "It is the magnitude of treatment differences that is of primary importance, not their statistical significance"

Which of the four questions listed above are concerned with *significance* and with *magnitude* respectively? Do you agree with Fisher?

Exercise 1.2: *Datasets, variables and observations.*

Often data is organized in tables in a laboratory diary or in an Excel sheet. Below you see four examples from the biosciences. For each example discuss the following questions, and summarize your conclusions in a *Table-of-Variables*:

(a) How many observations have been made?

(b) What are the variables in the experiment?

(c) What are the variable types (nominal, ordinal, interval, ratio)?

(d) What do you think is the relevant question to be answered by the statistical analysis?

(e) Which variable would you use as the response?

[1] In some situations the word *"effect"* should be replaced by *"association"* in these questions.

Data example 1: In an experiment concerning the effect of antibiotic and vitamin additives on growth 12 rats were given two different levels of antibiotic and two different levels of vitamin in their diet, and the growth was measured over some time period. The following table shows the measurements for all 12 rats.

	Level of vitamin	
Level of antibiotic	0	5
0	1.30 1.19 1.08	1.26 1.21 1.19
40	1.05 1.00 1.05	1.52 1.56 1.55

Hint: There are three variables in this example.

Data example 2: An experiment made by Anders Juel Møller (KVL) compared two chilling methods (tunnel-chilling and fast-chilling) of pork meat. 24 porks were sampled from two pH groups (high and low pH). After slaughtering the 24 porks were divided into two sides. One side was tunnel-chilled, the other fast-chilled. After some time the tenderness of the 48 meat pieces was measured. The measurements are displayed in the following table.

Pork	pH	Tunnel	Fast
1	low	7.22	5.56
2	low	3.11	3.33
3	low	7.44	7.00
4	low	4.33	4.89
5	low	6.78	6.56
6	low	5.56	5.67
7	low	7.33	6.33
8	low	4.22	5.67
9	low	3.89	4.00
10	low	5.78	5.56
11	low	6.44	5.67
12	low	8.00	5.33
13	high	8.44	8.44
14	high	7.11	6.00
15	high	6.00	5.78
16	high	7.56	7.67
17	high	5.11	4.56
18	high	8.67	8.00
19	high	5.78	7.67
20	high	6.11	5.67
21	high	7.44	7.56
22	high	7.67	6.11
23	high	8.00	8.22
24	high	8.78	8.44

Data example 3: In an experiment comparing the difference between two different diets 20 persons participated. By randomization 10 persons were assigned to each diet and every week a weight gain or weight loss was observed. The observations are the number of weeks where the diet resulted in a weight loss for each of the 20 persons in the experiment. The table below displays the results for a period of eight weeks showing the number of persons for each combination of diet and weeks with weight loss.

	Weeks with weight loss								
	0	1	2	3	4	5	6	7	8
Diet 1	1	0	2	0	1	1	2	0	3
Diet 2	2	1	0	1	2	1	2	1	0

Hint: The observations are perhaps not what they seem at first sight. How many observations are there here?

Data example 4: In an experiment concerning the influence of stress on metabolism in rats the regulation of 96 genes were measured using the qPCR method. A total of 47 rats were allocated to 8 groups as shown in the following table.

Group	1	2	3	4	5	6	7	8
Number of rats	6	5	6	6	6	6	6	6
Sex	male	male	male	male	female	female	female	female
Stabeling	single	single	group	group	single	single	group	group
Food additive	no	yes	no	yes	no	yes	no	yes

In each group the average gene regulation was measured on a logarithmic scale. The following table shows the measurements for 8 genes.

	Group							
Gene	1	2	3	4	5	6	7	8
Abcb1b	5.554	4.49	4.85	5.076	7.416	6.684	7.524	6.894
Abcb1	5.334	5.55	5.53	4.656	3.456	3.134	3.894	3.004
Abcb4	1.134	1.19	1.51	1.406	1.916	1.454	2.054	1.684
Abcc1	8.114	8.01	8.86	8.466	8.316	7.104	7.884	6.644
Abp1	8.224	8.68	9.24	8.676	11.406	8.504	10.604	8.214
Adh1	−2.996	−3.38	−2.92	−3.214	−3.964	−4.216	−3.766	−4.416
Adh4	2.944	3.10	3.24	3.786	2.456	2.474	2.154	2.494
Ahr	3.624	3.62	4.56	4.976	3.136	3.334	3.014	3.294

The experiment was made by Tina Vicky Alstrup Hansen (UCPH-LIFE).

Exercise 1.3: *Simple but excellent features in RStudio.*

The purpose of this exercise is to show some small details in RStudio as well as my favourite method of starting RStudio. I have a Windows 10 laptop, and the RStudio icon is attached to my "process bar" in the lower-left corner of my desktop. Alternatively I might have had the RStudio icon on the desktop itself, or in the "programs folder" in the start-menu. I'll assume that you have similar access to RStudio on your Windows, Mac, or Linux laptop. Now please start RStudio, and do the following step-by-step (I hope it works, let's see...).

1. Look at the *Environment*[2] in the upper-right window. Are the any variables?

2. In any case, let's make two new variables and a data frame by executing the following 3 lines in the *Console* in the lower-left window:

   ```
   x <- rnorm(100)
   y <- x+rnorm(100)
   z <- data.frame(x1=x,x2=y)
   ```

   Now you should have (at least) 3 objects in your *Environment*: Two *Values* called "x" and "y", and one *Data* called "z".

3. Click on the object called "z" in order to get a display in the *Editor* in the upper-left window. You may always do this later on to see the data you have inside R. Pretty neat[3], right?

4. Clear the *Environment* by executing the following line in the *Console*:

   ```
   rm(list=ls())
   ```

   Alternatively, you can also clear the *Environment* by clicking the "broom" icon in the upper-right window.

5. Now let save an empty(!) R-dataset to the folder you are using for the computer exercises today: Click on the *Files*-menu in the lower-right window, and browse the file system to locate the folder (sometimes

[2]In early versions of RStudio this was called the *Workspace*. Essentially its the same thing, but now you have the option of choosing different environments.

[3]In my view as a teacher this feature matches the major pedagogical point of Excel and JMP (easy to use statistical software, available from www.kunet.dk), namely the possibility to *see* the datasets.

also called a "directory") you want to use. E.g. it might be a subfolder called "Day 1" in a folder called "Statistics course" (you, of course, may have used different names).

6. Click on "Set As Working Directory" inside the *More*-submenu in the lower-right window.

7. Now close RStudio. Since the workspace has been changed (in the steps you did above), RStudio will ask you whether you want to save the workspace. Click "Save" (or possibly "Yes") to do so.

   Of course you can also save your data without closing RStudio. Either by clicking the "floppy-disk" icon in the *Environment* window, or by executing the `save.image()` command in the *Console*.

8. Now, use your operating system (Windows, Mac, or Linux) to browse to the folder you chose above. Then there should be a file called something like ".RData". This file contains the empty workspace you just saved.

9. If you (double) click on this file, then it hopefully activates RStudio[4] at the present folder on your laptop.

Of course this way of saving the working environment makes even more sense when you save a non-empty environment with the variables you are working on. In my daily work I have relevant R datasets for each of the projects I'm working on. And to continue my work I simple double click on the R datasets in order to resume my work inside the correct folder.

Exercise 1.4: *Getting data into R.*

In this exercise we continue practising the usage of RStudio. The first step in a statistical analysis is to get the data into R. So let's try this using the second data example from Exercise 1.2.

If you have data in a plain text file (ASCII file), then it is straightforward to import the data. Let's try this using the file `dataExample2.txt` available from the ZIP archive `day1.zip`[5]

1. Click on "From Text (base)..." in the *Import Dataset*-submenu in the *Environment* window.

[4] Alternatively it might start the classical R console (RGui). If so, then you should use the operating system to associate .RData files with RStudio instead.

[5] You need to "unzip" the ZIP archive, i.e. extract the files inside the archive, before you can import the files into R. Ok?

2. Browse your file system to locate the file `dataExample2.txt` and click "Open" (or something similar) to open the file.

3. A window with an *Import Wizard* should appear. The file `dataExample2.txt` contains a heading with the variable names, data entries are separated by *Tab(-ulator)* signs, and decimals are given by commas.

4. Try out the different possibilities in the left part of the Wizard, and see how the data frame in the lower part changes.

5. Click "Import" when things look right.

Please notice that the import Wizard actually generates some R code in the *Console* (which is also available in the *History* in the upper-right window). Thus, you may insert this R code in your R program instead of using the import Wizard. In the long run this is much easier[6].

In practice it is more common to have data recorded in an Excel sheet. So let's try to read the Excel file named `dataExample2.xls`, which is assumed to be extracted from the zip archive `day1.zip`.

1. Make sure that the possibility "From Excel..." is available in the *Import Dataset*-submenu in the *Environment* window. If this is not the case, then you probably should update RStudio.

2. Import `dataExample2.xls` by using the "From Excel..." interface. And again, for future usage you might want to use the generated R code instead of using the import Wizard.

3. Now that we have a dataset inside R let's also try to make some graphics. Execute the following R commands[7] in the *Console*:

```
library(ggplot2)
ggplot(dataExample2,aes(x=Fast,y=Tunnel,col=pH)) +
  geom_point() +
  geom_abline(intercept=0,slope=1) +
  ggtitle("Tenderness of pork meat")
```

   Can you decipher the purpose of the R code? And what can you infer about the data from the resulting plot?

[6] And also necessary when you knit R Markdown documents!

[7] If the `ggplot2`-package is not available, then you need install it via "Install" inside *Packages* in the lower-right window.

Exercise 1.5: *Hypertension in diabetic patients.*

Before commencing on the statistical methods we introduce yet another R technicality. So far we have seen data encoded in text-files, Excel sheets, and R scripts. But of course R also has a format for saving data, namely in RData-files[8]. If RStudio is already open, then you may read RData files using the "open file" icon in the *Environment* window, or by using the `load()` function from the *Console*. If RStudio is not open, then you may open RStudio together with the RData file by double clicking on the file (in Windows).

The data for this exercise is available in the file `hypertension.RData`, and also in an Excel sheet (just in case you need it, which you shouldn't).

An experiment on 19 diabetic patients was conducted in order to compare the effects of two drugs called *Drug E* and *Drug N* on the treatment of high blood pressure. The experiment is a cross-over study. This means that all patients try both drugs in two different study periods. Both study periods lasted for 14 days. In between the two study periods was a wash-out period, which also lasted for 14 days. The patients were randomly assigned to two groups called *E/N* and *N/E*. The patients in the E/N-group received drug E in the first study period and drug N in the second study period. The patients in the N/E-group received drug N in the first study period and drug E in the second study period.

The systolic and the diastolic blood pressure was measured for all the patients at the beginning and the end of both study periods. In this exercise we will only analyse the observations of the systolic blood pressure. These observations are shown in the table on the next page.

[8] We already have worked with RData-files in Exercise 1.3.

		Systolic blood pressure			
Patient id	Treatment order	Baseline 1	End 1	Baseline 2	End 2
9	Drug E, Drug N	124	136	120	145
21	Drug E, Drug N	120	132	138	126
8	Drug E, Drug N	115	96	111	91
12	Drug E, Drug N	134	118	123	123
16	Drug E, Drug N	131	106	111	123
19	Drug E, Drug N	119	108	113	112
20	Drug E, Drug N	124	112	108	112
24	Drug E, Drug N	127	113	121	143
13	Drug N, Drug E	113	113	107	97
17	Drug N, Drug E	132	109	122	119
18	Drug N, Drug E	129	133	139	130
23	Drug N, Drug E	124	120	127	118
25	Drug N, Drug E	112	103	112	121
10	Drug N, Drug E	124	112	128	122
11	Drug N, Drug E	144	154	156	137
14	Drug N, Drug E	134	118	122	109
15	Drug N, Drug E	119	118	115	114
22	Drug N, Drug E	123	123	114	108
26	Drug N, Drug E	122	123	124	120

The R dataset `hypertension.RData` contains the dataset. Beside the raw observations encoded in the variables *patient, order, baseline1, end1, baseline2* and *end2* five new variables called *change1, change2, average, diff* and *E_diff_N* have been defined.

- The variable *change1* contains the change of blood pressure over study period 1.
- The variable *change2* contains the change of blood pressure over study period 2.
- The variable *average* contains the average change of the blood pressure over both study periods.
- The variable *diff* contains the difference of the changes of blood pressure between study period 1 and study period 2.
- The variable *E_diff_N* contains the difference of the changes of the blood pressure between the study periods given drug E and drug N.

To analyse the dataset for the cross-over study the following four T-tests may be performed:

- Two sample T-test comparing *E_diff_N* in the E/N-group against the N/E-group.
- Two sample T-test comparing *average* in the E/N-group against the N/E-group.
- Two sample T-test comparing *diff* in the E/N-group against the N/E-group.
- One sample T-test comparing *E_diff_N* against the mean value 0.

Two of these T-tests do the actual comparison between the effects of drug E and drug N. These tests, however, are only valid when the following two problems do not occur:

Problem 1: A spill-over (also called a carry-over) from study period 1 to study period 2. A possible explanation for such an effect is that the drug given in study period 1 still has an effect in study period 2.

Problem 2: An interaction between the effects of the drugs and the study periods. For instance that the effect of drug E for some strange reason is larger in study period 1 than in study period 2.

The two remaining T-tests are done to validate that these two problems have not occurred.

a) Which of the four T-tests listed above do the drug comparison, and which T-tests validates against problem 1 and 2?

   Help to get started: If the drugs have different effects and if there is a spill-over from period 1 to period 2, then the difference between the changes in the E- and the N-period will depend on the order the drugs were given.

b) Perform the relevant T-tests. Remember to validate the underlying normality assumption before you make the T-tests. What is the conclusion from these tests?

Remark: I would probably not do the statistical analysis using all these T-tests. Instead I would do the analysis using a random effect model. We will return to this on course day 5.

Reference: Bradstreet, T.E. (1994) "Favorite Data Sets from Early Phases of Drug Research - Part 3." *Proceedings of the Section on Statistical Education of the American Statistical Association.*

End of Exercises.