R Exercises

1. The goal of this exercise is to get acquainted with different abilities of the R statistical software. It is recommended to use the distributed R tutorial as a guide.

R contains more than 50 datasets and more can be loaded using optional packages. The package `VR` is depending on the package `MASS` which contains the dataset `survey`. This dataset comprises of measurements and answers taken from 237 students of statistics at the university of Adelaide. The following variables are available

Sex	gender of student
Wr.Hnd	span width in cm (from thumb to pinky) of the writing hand
NW.Hnd	span width in cm (from thumb to pinky) of the non-writing hand
W.Hnd	writing hand
Fold	When folding your arms - which one is on top?
Pulse	beats per minute
Clap	When clapping your hands - which on is on top?
Exer	How often do you exercise?
Smoke	How often do you smoke?
Height	body length in cm
M.I	Preference of either metric (cm/m) or imperial (feet/inches) units?
Age	age in years

> **library(MASS)**	makes the datasets of the `MASS` package available **PC: Install first the package** `VR`
> **data()**	shows a list of all available datasets
> **help(survey)**	gives a description of the dataset `survey`
> **data(survey)**	makes the dataset `survey` available

Useful functions to get a first overview of the dataset:
str(survey), **summary(survey)**, **table(survey$Sex)**, **table(survey$Sex, survey$Smoke)**
The notation `survey$Smoke` accesses the variable `Smoke` in the dataset `survey`.

> **attach(survey)**	puts the dataset `survey` on level 2 of the list of available objects. The working directory is on level 1. The variables in the dataset `survey` can now be accessed directly with their names, i.e. instead of typing `survey$Smoke` you may access the variable directly with `Smoke`.

Dealing with missing values (`NA`):

> **mean(Pulse)**	result is `NA`
> **mean(Pulse, na.rm=T)**	the missing values are removed from the calculation of the mean
> **na.omit(Pulse)**	all missing values are removed
> **Pulse[!is.na(Pulse)]**	same as above, but generated *by hand*

Useful functions for graphics:

> **hist(Height)**	histogram
> **boxplot(Height)**	boxplot
> **boxplot(split(Height, Sex))**	boxplots of two variables
> **boxplot(Height[Sex=="Female"],Height[Sex=="Male"])**	boxplots
> **plot(Wr.Hnd,NW.Hnd)**	scatter plot
> **plot(Sex,Height)**	?

> **detach(survey)** disconnects the dataset `survey` from level 2, i.e. variables can no longer by accessed directly, but only using `$` or `[·,·]`:
> **plot(survey$Wr.Hnd,survey$NW.Hnd)** or **plot(survey[,2],survey[,3])**.

Selecting observations, i.e. only the first 50:
> **plot(survey[1:50,2],survey[1:50,3])**

Do not forget about the online help:
> **help(survey)**
> **help(plot)**
...

Now analyse the dataset `survey` using descriptive methods. Therefore produce tables and contingency tables of the categorical variables and calculate location and deviation properties for the continuous variables. Provide suitable graphical representations. Comment on the distributions. Are there any outliers?

Answer the following questions:

a) Is the span width of the writing hand in general larger than the span width of the non-writing hand?

b) Do the two oldest students smoke?

c) Which factors might have an influence on the student's pulse?

d) It is generally believed that the pulse of an individual decreases with increasing age. The function `lm` fits a linear regression. Investigate the output of the following code:
> **Agejung <- Age[Age<30]; Pulsejung <- Pulse[Age<30]; plot(Agejung,Pulsejung)**
Comment on the output. What does the above code do?
> **lmobj <- lm(Pulsejung ~ Agejung); plot(Agejung,Pulsejung); abline(lmobj)**

2. Vectors

What is the output of the following commands? Try to predict the solutions before you type in the commands. We define:

```
x <- c(5,2,1,4); xx <- c(1,10,15,18); y <- rep(1,5)
z <- c(TRUE,FALSE,TRUE,TRUE); w <- c("Marie","Betty","Peter")
```

a)
```
sum(x)
range(x)
length(y)
sum(y)
```

b)
```
c(x,y,13)
```

c)
```
xx - x
c(x,12) * y
1:6 + 1
1:9 + 1:2
```

d)
```
x <= 2
x <= 2 & z
```

e) `substring(w,2,4)`
   `paste(substring(w,1,2),substring(w,5,5),sep="..")`

f) `cbind(x,xx)`
   `cbind(2,6:1, rep(c(3,1,4),2), seq(1.1,1.6,by=0.1))`

3. Sequences of Numbers

Create the following sequences. Use the commands `rep` and `seq`.

a) `1 2 3 4 5 6 7 8 9`

b) `"m" "w" "m" "w" "m" "w" "m" "w" "m" "w"`

c) `1 2 3 4 1 2 3 4 1 2 3 4`

d) `4 4 4 3 3 3 2 2 2 1 1 1`
   Hint: Use argument `each` of the function `rep`.

e) `1 2 2 3 3 3 4 4 4 4 5 5 5 5 5`

f) `1 1 3 3 5 5 7 7 9 9 11 11`

4. Matrices.

a) Generate the following matrices.

```
     [,1] [,2] [,3] [,4]
[1,]    1  101  201  301
[2,]    2  102  202  302
[3,]    3  103  203  303
[4,]    4  104  204  304
[5,]    5  105  205  305
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    5    0    0    0    0    0    0    0    0     0
 [2,]    0    5    0    0    0    0    0    0    0     0
 [3,]    0    0    5    0    0    0    0    0    0     0
 [4,]    0    0    0    5    0    0    0    0    0     0
 [5,]    0    0    0    0    5    0    0    0    0     0
 [6,]    0    0    0    0    0    5    0    0    0     0
 [7,]    0    0    0    0    0    0    5    0    0     0
 [8,]    0    0    0    0    0    0    0    5    0     0
 [9,]    0    0    0    0    0    0    0    0    5     0
[10,]    0    0    0    0    0    0    0    0    0     5
```

b) Explore the properties of your generated objects. Which class of R-objects do they belong to? How are they structured?
   Hint: class(), dim(), str(), summary().

5. Lapwings

For various meadows at Zurich Airport we counted the daily number of lapwings on several occasions. For every bird we noted the kind of activity (resting, feeding, flying) as well as the ground conditions (damp, dry, wet). The data was stored in `vogel.dat`:

```
   Datum  Zeit Feld.Nr Anzahl Taetigkeit Boden
1 910903 10.06    1411     22         ru     t
2 910903 10.07    1413     15         ru     t
3 910906 10.01    1411     29         fr     t
4 910906 10.03    1413     44         fr     t
5 910910 15.19    1410     34         ru     n
6 910911 10.00    1413     41         fr     f
7 910912 12.38    1411     10         ru     t
8 911014 15.17    1409      2         fl     t
9 911203 13.05    1413      2         fl     t
```

It contains the columns date (`Datum`), time (`Zeit`), id of meadow (`Feld.Nr`), count (`Anzahl`), activity (`Taetigkeit`) (`ru` resting, `fr` feeding, `fl` flying) and ground condition (`n` wet, `t` dry, `f` damp).

a) To read the data into R type:
`d.vogel <- read.table("http://stat.ethz.ch/Teaching/Datasets/NDK/vogel.dat", sep =";", header=TRUE)`

b) Create a new data frame that only contains the meadow id and the counts. How many birds were counted on average?

c) Create a data frame only with the data of meadow 1413.

d) Create a vector that contains the number of birds of meadow 1413.

e) On how many occasions(days) did we observe feeding birds? How many birds were counted while feeding? What were the corresponding observation numbers?

f) In the data frame change the number of lapwings of the eighth observation (row) to 6. Delete the third and seventh observation from the data frame.

Hint: `mean(), sum(), which()`.

6. Meteo

The data set `meteo70.txt` contains several measurements of weather variables between 1994 and 2007. The mean daily air temperature is stored in the variable `X211`.

a) Read the data into R:
```
fname <- "http://stat.ethz.ch/education/semesters/ss2014/regression/uebungen/meteo70.txt"
d.meteo <- read.table(fname, header=T)
```
The missing values are encoded by the value 32767. Change this value to `NA` and rename the variable `X211` in `temp`.

b) Calculate the mean of `temp` separately for each day of the week.
Hint: aggregate()

c) Plot the mean of `temp` for each year. Why is the mean in 2007 so high?

7. Getting to know Data: Iris blossoms

The data set `iris` contains measurements of the length and the width (in cm) of petals and sepals of three iris species: `1`: Setosa, `2`: Versicolor and `3`: Virginica.
(Source: R. A. Fisher, *The Use of Multiple Measurements in Taxonomic Problems*, Annals of Eugenics, Vol. 7, Part II, 1936, pp. 179-188.)

a) This data set `iris` is already part of the standard R installation. Consider the object `iris`. How is it structured? How many observations (lines) does it contain? How many variables (columns)?
Hint: nrow(), ncol(), dim(), str()

b) To get an overview of the range of values, look at the `summary()` of the data set. Which information on the data set does it provide?

c) For the variable `Sepal.Length` check the results above by using the R-functions `min(), max(), mean(), median(), quantile()`. If necessary, make use of the help functions `?quantile` etc.

8. Missing Values

Statistics needs data. Unfortunately, data often cannot be collected fully. Therefore many data sets contain "gaps", non-existing measurements, so-called NAs (not available). In this exercise you will get to know how R deals with NAs. We work with the data set `iris`. Make a copy of the iris data set by `d.iris <- iris`.

a) Assume that we were unable to take the second observation of `Petal.Length` and `Petal.Width`, and for the fifth observation, the data for `Sepal.Length`, `Sepal.Width` and `Petal.Width` are missing. Replace these five fields by `NA`.
Hint: Replace the values by NAs using e.g. `d.iris[2, 3:4] <- NA`

b) Consider the first eight observations of the modified data set, to observe how the NAs are displayed by R. The commands `class()`, `nrow()`,`ncol()`, `dim()`, `str()` also work for the data set with missing values. What changes in the `summary()`?

c) Try to confirm the given values for the variable `Sepal.Length` using `min()`, `max()`, `mean()`, `median()`, `quantile()`. Is there a difference?

d) There are functions that cannot handle `NA`s (Result '`NA`' or 'Error: missing observations'). There is a trick to make them calculate the correct results: simple functions such as `min()`, `max()`, `mean()`, `median()`, `quantile()`, `range()` etc. can take an argument `na.rm`. When you set its value to `TRUE`, the NAs will not be considered in the calculation.
Try to confirm the values provided by `summary()` again, using this new argument.

e) Why should missing values always be coded by `NA`, and not, for instance, filled with a zero? Explain for the case of the `mean()` function.

f) Experiment with missing values in the statistical functions `var()`, `sd()`, `cor()`. Can you explain the behaviour of R?

g) Select only those observations which have missing values in either `Sepal.Length` or in `Petal.Length`.
Hint: `is.na()`

h) The function `na.omit()` eliminates all observations from the data frame for which **any(!)** variable contains `NA`s. Save the result of `na.omit(d.iris)`. How many observations remain? How many remain using `na.omit(d.iris[ ,1:3])`?

Note:
Higher-level functions such as `t.test()` or `wilcox.test()` have an argument `na.action`, with which the reaction to `NA`s can be determined. `na.action=na.omit` first deletes all lines (observations) with `NA`s before anything is calculated.