TMA4255 Applied Statistics
R-intro, integrated with Exercise 2

1. **Start R or Rstudio** on your laptop, or remote desktop to cauchy.math.ntnu.no and log in with

   ```
   win-ntnu-no\yourusername
   ```

   and password.

   If you use R directly (not Rstudio) you have a R console window, and need to start a separate text editor if you want to make a nice set of commands to be readable for you later (instead of the script window below).

   With Rstudio you already have four windows:

   - Console (where the R commands are executed),
   - Source (the script window),
   - Workspace/History (the objects that you have in your workspace, and the commands you have executed),
   - Files/Plots/Packages/Help (yes, exactly).

   Look at the Rstudio-pages for lists of shortcuts. Most importantly, ESC, terminates a command (in R this is ctrl-c).

2. **An oversized calculator** Try writing the following in the console window, one command at a time:

   ```
   2+3
   2*6
   3*10^4-3*5^2
   sqrt(9)
   log(3,base=10)
   ?log
   log
   exp(34)
   gamma(3)
   factorial(2)
   choose(10,4)
   1:4
   c(1,2,3,4)
   sum(1:5)
   ```

3. **Objects** Can both use "=" and "$<-$" for assigning content to an object. List all objects with `ls()` or just look in the Workspace window. There also `mode(obj)` is shown.

   The function `c` combines values into a vector (concatenate).

```
heights <- c(192,185,174,195,173)
shoes <- c(46,43,40,45,40)
ratio <- height/shoes
ratio
ls()
```

4. **Reading data** Data for Exerc2Problem1 are found in the Lectures tab from the WWW-page as `dataE2P1.txt`, and can be read using
`ds <- read.table("dataE2P1.txt",header=TRUE)` after saving the file to the appropriate place, or directly from the www using
`ds <- read.table("http://www.math.ntnu.no/~mettela/TMA4255/Data/dataE2P1.txt",header=TRUE`

   Data for Execr2Problem2 are found in `dataE2P2.txt` with the 10+8 numbers as one long vector. Read them all (after you have saved the file) and divide into to vectors by

   ```
   dss <- scan("http://www.math.ntnu.no/~mettela/TMA4255/Data/dataE2P2.txt")
   dsA <-dss[1:10]
   dsB <- dss[11:18]
   ```

   Why `scan` and not `read.table`? R does not like non-rectangular data sets (unequal number of rows).

   Other commands for reading data are `scan, read.csv`. Remeber, R use "." as decimal separator (while MINITAB use ",").

   ```
   dim(ds)
   length(dss)
   ```

5. **Data frames and matrices** Entries in a data frame or matrix can be indexed using [row,col], and a column by [,col] and a row by [row,].

   ```
   dif <- ds[,1]-ds[,2]
   ds[1,] # observation for first bus
   ds[1,1] # first bus, the G tyre
   ```

   A data frame is a list structure, and `names(ds)` gives names of objects in the list.

6. **Plots**

   ```
   boxplot(ds)
   plot(ds[,1],ds[,2])

   plot(1:length(dif),dif)
   qqnorm(dif)
   qqline(dif)
   ```

   Plots can be saved using the menues on the plot window, pdf or png recommended. In a script that you would like to run without needing to use menus you may use `pdf("box.pdf"); boxplot(ds); dev.off()` to make a file named `box.pdf` with the boxplot.

7. **Statistical tests** For the z-test wait a bit, and do the t-test first (then read the next two items for two possible solutions to the z-test).

   First, for Problem 1, we need t-test.

```
summary(dif)
mean(dif)
sd(dif)

?t.test
t.test(dif,alternative="greater")
t.test(ds[,1],ds[,2],paired=TRUE,alternative="greater")
t.test(ds[,1],ds[,2],var.equal=TRUE,alternative="greater")

qt(0.05,8,lower.tail=FALSE)
```

   Observe that `paired=TRUE` is the same as making differences and calling the one-sample t-test.

   Then, for Problem 2, we need F-test.

```
dss <- scan("http://www.math.ntnu.no/~mettela/TMA4255/Data/dataE2P2.txt")
dsA <-dss[1:10]
dsB <- dss[11:18]

var(dsA)
var(dsB)

?var.test
var.test(dsA,dsB)
varres <- var.test(dsA,dsB,conf.level=0.9)
names(varres)
sqrt(varres$conf.int)

qf(0.025,9,7,lower.tail=TRUE)
qf(0.025,9,7,lower.tail=FALSE)
1/qf(0.025,7,9,lower.tail=TRUE)
```

8. **Plotting distributions** The t-distribution from Problem 1 and the F-distribution from Problem 2 can be plotted and critical values marked. This to better understand the CI and hypothesis test.

```
qt(0.005,8) # need at least this far - ok go to -4,4
x <- seq(-4,4,length=500)
y <- dt(x,8)
plot(x,y,type="l", main="T-distribution with 8 df")
abline(v=qt(0.025,5))
abline(v=qt(0.975,5))

qf(0.005,9,7) # go from 0
qf(0.995,9,7) # go to 9
```

```
x <- seq(0,9,length=500)
y <- df(x,9,7)
plot(x,y,type="l", main="F-distribution with 9 and 7 df")
abline(v=qf(0.025,9,7))
abline(v=qf(0.975,9,7))
abline(v=var.test(dsA,dsB)$statistic,col=2)
```

Alternatively, use `curve`:

```
curve(dt(x,df=8),-4,4,n=500)
```

9. **Installing and using packages** A z-test (known sigma) is not available in R (since this situation practically never arises except with the purpose of teaching...), so you can either program your own function, or download the TeachingDemos library and use the z.test in this library.

   To install the TeachingDemos library you write `install.packages("TeachingDemos")` and choose Norway for downloading. Then start the library by typing `library(TeachingDemos)`. Then `?z.test` will give you help about the z-test. If you are in command of you laptop, drop the lib argument, else create a folder names Rlibs shomewhere to put R libraries in `install.packages("TeachingDemos",lib="M://Rlibs/")`. Remember to use "/" for folders in Windows.

   To use the Anderson-Darling test for normality use the function `ad.test` in `library(nortest)`

10. **Writing a simple Z-test as a function** To not make it too complicated let us assume that we are testing $\mu = 0$ in one sample and that we use a two-sided test with significance level 0.05. Then we need a function with the data and given standard deviation as input.

```
myz.test <- function(x,sd)
{
n <- length(x)
xbar <- mean(x)
zobs <- xbar/(sd/sqrt(n))
print(paste("Zobs",zobs))
pval<- 2*pnorm(abs(zobs),lower.tail=FALSE)
print(paste("P-value",pval))
return(list("statistic"=zobs,"p.value"=pval))
}

myz.test(dif,sd=6)
```

   For Problem 1 the test is one-sided, how can you change the function to handle this?

11. **Quitting R** `q()` to quit R, if you answer yes to "Save workspace image" all the objects you have created are found in a `.RData` file. Remember to save the script.

12. **R souce file for exercises** For Exercise 1 and 2 a source file for R is available at the course www-page.

    You may run it using
    `source("http://www.math.ntnu.no/ mettela/TMA4255/V2013/E1.r",echo=TRUE)`,

and the same with `E2.r`. The echo argument makes the commands also printed to the screen.

If you want to save commands and output to a text-file called E1allres.txt, you may use sink:

```
sink("E1allres.txt")
source("http://www.math.ntnu.no/~mettela/TMA4255/V2013/E1.r",echo=TRUE)
sink()
```

In E1 I have used a setting where R pauses whenever a new plot is made, `par(ask=TRUE)`.