Computational Methods for Statistic with Applications
Computer Exercise no. 1: Getting acquainted with $\mathbb{R}$ and Matlab

NGSSC, LU,, SLU, UU

September, 2011

The Lab contains tasks of varying difficulty. Depending on your experience with $\mathbb{R}$ you may skip those, which seem too elementary for you. You are also free to extend some of the exercises with additional tasks, relevant to the theme of that exercise.
At the end, the results of three exercises have to be sketched and commented, and given to the lab-consultant. It is recommended to use the publishing facilities, provided in $\mathbb{R}$ and Matlab.
Some of the tasks require some data files which can be downloaded via
`http://http://user.it.uu.se/~maya/Courses/NGSSC/STAT_Labs/MN/`.
Make a copy of these files in some of your directories.

Exercise 1 (First steps in $\mathbb{R}$) .

1. Explore the possibilities for obtaining information about $\mathbb{R}$ via the help command.
2. Ask $\mathbb{R}$ about objects and commands such as 'distribution', 'plot', 'data.frame', 'lm', 'var', 'multivariate', ...

Exercise 2 (Random variables, test, plot) .

1. ```
   > sample=rnorm(100)
   > mu=mean(sample)
   > sigma=sd(sample)
   > x=seq(-2,2,length=1000)
   > y=dnorm(x,mu,sigma)
   > plot(x,y,type='l')
   ```
2. Go through the help for the command 'plot'. Test it on some data. You can create your own test data or use some of the available data sets.
   Example:
   ```

```
>>R
> ls()
character(0)
> library(MASS)
> data()
Data sets in package 'datasets':

AirPassengers           Monthly Airline Passenger Numbers 1949-1960
BJsales                 Sales Data with Leading Indicator
BJsales.lead (BJsales)
                        Sales Data with Leading Indicator
BOD                     Biochemical Oxygen Demand
CO2                     Carbon Dioxide uptake in grass plants
ChickWeight             Weight vs age of chicks on different diets
....
> data(ChickWeight)
> plot(ChickWeight[1:578,1])
```

Use some of the features of the 'plot' command to put a title and some other information to the plot.

Test the examples from the help for 'plot':

```
> plot(sin, -pi, 2*pi)
> plot(x <- sort(rnorm(47)), type = "s",
       main = "plot(x, type = \"s\")")
       points(x, cex = .5, col = "dark red")
```

Exercise 3 (More on graphics in $\mathbb{R}$)

Consider the data set `htwt_R.dat`. It contains data of 237 children, their sex ('m' or 'f'), age (months), heigh (cm) and weight (kg).

1. Load the data set `T=read.table("htwt_R.dat",header=T)`
2. See what is inside `> T`
3. Plot the data by `> plot(T)`. What do you see?
4. Attach T for convenience
5. Plot on separate plots age versus height, age versus weight:

   ```
   > plot(age, height)
   ```

6. Do regression and plot the regression line

```
> plot(age, height, main="Age vs height")
> linfit1<-lm(height~age)
> abline(linfit1,col="red")
```

7. Multiple plots in a single graphical window

```
par(mfrow=c(2,2))
plot(age,weight,type="o",title("Age vs Weight"))
plot(age,height,col=2,title("Age vs Height"))
plot(age,height,col=1,title("Age vs Height (a)"))
plot(age,height/weight,type="o", lty=2, title("Just a trial"))
par(mfrow=c(1,1))
```

8. Extract a subset of a data set, subject to a criteria (for example, plot data only related to female persons)

```
> subsF=T[T$sex=="f",,,]
> subsF
```

Plot in the same graphics window age vs height separately for boys and girls.

9. Overlaying plots
We run three different models for the relation 'age-height' and plot the resulting LS fit on one and the same window with different colors, to visually see which produces a better fit.

```
> plot(age, height, main="Age vs height")
> linfit1<-lm(height~age)
> abline(linfit1,col="red")
> sqrfit1<-lm(height~age+I(age^2))
#Plotting the quadratic curve is less simple.
# One solution: To obtain the plot, we'll first create a sequence of x values,
# then apply the linear combination implied by the regression model using
# matrix multiplication:
> xx <- seq(min(age),max(age),len=50)
> yy <- sqrfit1$coef %*% rbind(1,xx,xx^2)
> lines(xx,yy,lwd=2,col=3)
```

10. Saving into Postscript files/Xfig files (The dev.off() function writes the file.)

```
postscript("test.ps")
plot(x)
dev.off()
#######
xfig("test.fig")
plot(x)
dev.off()
```

11. Some examples

```
## Setup up coordinate system (with x==y aspect ratio):
  plot(c(-2,3), c(-1,5), type = "n", xlab="x", ylab="y", asp = 1)
  ## the x- and y-axis, and an integer grid
  abline(h=0, v=0, col = "gray60")
  text(1,0, "abline( h = 0 )", col = "gray60", adj = c(0, -.1))
  abline(h = -1:5, v = -2:3, col = "lightgray", lty=3)
  abline(a=1, b=2, col = 2)
  text(1,3, "abline( 1, 2 )", col=2, adj=c(-.1,-.1))

 ## Simple Regression Lines:
  require(stats)
  sale5 <- c(6, 4, 9, 7, 6, 12, 8, 10, 9, 13)
  plot(sale5)
  abline(lsfit(1:10,sale5))
  # less fitting
  abline(lsfit(1:10,sale5, intercept = FALSE), col= 4)

  z <- lm(dist ~ speed, data = cars)
  plot(cars)
  abline(z) # equivalent to abline(reg = z) or
  abline(coef = coef(z))

  ## trivial intercept model
  abline(mC <- lm(dist ~ 1, data = cars)) ## the same as
  abline(a = coef(mC), b = 0, col = "blue")
```

Exercise 4 (Package installation, libraries)

1. Find out how to see which packages are currently available.
2. Check the data sets and functions in the package `MASS`
3. Load the package `ISwR`
4. Check the available data sets (`data()`)
5. Choose a data set and analyse it as much as you can - plot, print the data, plot a histogram.

```
datafilename="http://personality-project.org/r/datasets/maps.mixx.epi.bfi.data"
person.data =read.table(datafilename,header=TRUE)  #read the data file
#print out the min, max, range, mean, median, etc.
summary(person.data)
```

```
#make the variables inside of data available to be called by name
 attach(person.data)
 max(epiImp)
 min(epiImp)
 mean(epiImp)
 median(epiImp)
 var(epiImp)
 sqrt(var(epiImp))
 sum(epiImp)
 sd(epiImp)
 mad(epiImp)
 fivenum(epiImp)
```

6. Try to install the package `psych`

Exercise 5 (Input and output of data in $\mathbb{R}$, editing)
Fifty samples of the yield stress of steel are available. The first 40 are already arranged in a table (file `Steel.dat`). The rest are listed in Table 1.
Tasks:

1. Read `Steel.dat` in $\mathbb{R}$. Use an appropriate function.
2. Use the facilities in $\mathbb{R}$ to edit data and add the rest of the samples.
3. Plot the histogram of the data.
4. Plot the cumulative distribution of the data.

41	29.7
42	29.2
43	27.8
44	31.7
45	30.6
46	29.1
46	29.1
47	30.2
48	29.4
49	30.3
50	27.2

Table 1: Yield stress of steel (in *kpsi*)

Exercise 6 (Statistical characteristics: mean, mode, median)

1. Find the mean, median, standard deviation and histogram corresponding to the data in the file `SChar_1.dat`.
2. Ask $\mathbb{R}$ to find out more about variance and deviation. Test those on some of the available data.

Exercise 7
The number of automobiles passing a border checkpoint during 9 am and 10 am on 50 different days is found in the file `SChar_2.dat`.

1. Read the data in a proper way in $\mathbb{R}$.
2. Plot the histogram and the cumulative distribution function.
3. Determine the mean, mode, median, variance, standard deviation, coefficient of variation and skewness coefficient of the given data.
4. Repeat the experiment in `Matlab`.

Hint: `mean, hist, var, median, dev, sd, mad, pooledSD`

Exercise 8 (Vectors, matrices, arrays and operations with those)

1. Create a vector of length 100 and entries, normally distributed over the interval (-2,3).
2. Create a diagonal matrix (D2) of size 40 with all entries equal to 2.
3. Create a diagonal matrix (DR) of size 20 with entries (1,2,...20).
4. Create a matrix (R) of size 10×2 with randomly generated entries.
5. Test the operations 'matrix-times-vector', 'matrix-times-matrix', 'matrix-transpose'.

Some hints:

```
m2<-matrix(c(1,3,2,5,-1,2,2,3,9),nrow=3);m2
m2<-matrix(c(1,3,2,5,-1,2,2,3,9),ncol=3,byrow=T);m2
m2[2,]
m2[-1,-1]
solve(m2)
diag(3)
diag(c(2,3,3))
eigen(m2)
```

Exercise 9 (Installation of packages)
Install the following packages, which will be needed during later labs.

- On the Unix machines you are working on:
  - multicore
  - Rcmdr
- On kalkyl:
  First execute:

  module unload pgi openmpi
  module load gcc openmpi

  Then install
  - multicore
  - Rmpi
  - snow