Computational Methods for Statistic with Applications
Computer Exercise no. 1: Getting acquainted with R

NGSSC, LU, UU

September-October, 2008

The Lab contains tasks of varying difficulty. Depending on your experience with $\mathbb{R}$ you may skip those, which seem 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 (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

Exercise 2 (Functions in $\mathbb{R}$)
Write your own function in $\mathbb{R}$.
Example: The function below computes a logarithm to the base b of a number x

```
> logbb = function(x,b) log(x)/log(b)
> logbb(8,2)
[1] 3
> logbb(81,3)
[1] 4
```

1. Write a function, named largerV(), which takes two vectors (assumed here to be of the same length) as arguments, compares them element by element, and returns a vector that consists of elements that are the larger in magnitude of the corresponding elements in the two vectors.

2. Write a recursive function which computes the factorial of an integer n, $n!$

3. Choose your own example. Save the function in a file with extension '.R'. Call the function from your $\mathbb{R}$ session.

Exercise 3 (Linear algebra stuff)
Create a symmetric positive definite (spd) matrix A of size 8 and compute its Cholesky factor. Do you know some famous spd matrices?

Exercise 4 (One Way ANOVA)
Execute the following commands and comment on those (borrowed from http://www.personality-project.org/r/r.guide.html).

```
#  One Way ANOVA

#tell where the data come from
datafilename="http://personality-project.org/r/datasets/R.appendix1.data"
data.ex1=read.table(datafilename,header=T)   #read the data into a table

aov.ex1 = aov(Alertness~Dosage,data=data.ex1) #do the analysis of variance
summary(aov.ex1)                              #show the summary table
#report the means and the number of subjects/cell
print(model.tables(aov.ex1,"means"),digits=3)
#graphical summary appears in graphics windowboxplot(Alertness~Dosage,data=data.ex1)
```

Exercise 5 (Two Way ANOVA)
Execute the following commands and comment on those (borrowed from http://www.personality-project.org/r/r.guide.html).

```
# Two Way (between subjects) Analysis of Variance (ANOVA)

datafilename="http://personality-project.org/R/datasets/R.appendix2.data"
data.ex2=read.table(datafilename,header=T)    #read the data into a table
data.ex2                                      #show the data
                                              #do the analysis of variance
aov.ex2 = aov(Alertness~Gender*Dosage,data=data.ex2)
summary(aov.ex2)                              #show the summary table
#report the means and the number of subjects/cell
```

```
print(model.tables(aov.ex2,"means"),digits=3)
#graphical summary of means of the 4 cells
boxplot(Alertness~Dosage*Gender,data=data.ex2)
attach(data.ex2)
interaction.plot(Dosage,Gender,Alertness)     #another way to graph the means
detach(data.ex2)
```

Exercise 6 (Linear regression)

Execute the following commands and comment on those (borrowed from http://www.personality-project.org/r/r.guide.html).

What is `par`?

```
# Linear regression
#  y~x or y~1+x are both examples of simple linear regression
#  with an implicit or explicit intercept.
# y~0+x or y~ -1 +x or y~ x-1 linear regression through the origin
# y ~ A where A is a matrix of categorical factors is a classic ANOVA model.
# y ~ A + x is ANOVA with x as a covariate
# y ~A*B or y~ A + B + A*B ANOVA with interaction terms

#where are the data
datafilename="http://personality-project.org/r/datasets/maps.mixx.epi.bfi.data"
#read the data file
personality.data =read.table(datafilename,header=TRUE)
names(personality.data)   #what variables are in the data set?
attach(personality.data)  #make the variables easier to use
#simple regression of beck depression on Neuroticism
model1 = lm(bdi~epiNeur)
summary(model1)           # basic statistical summary

                          #pass parameters to the graphics device
op <- par(mfrow = c(2, 2),# 2 x 2 pictures on one plot
               pty = "s") # square plotting region,
                          # independent of device size
#diagnostic plots in the graphics window
plot(model1)

model2=lm(bdi~epiNeur+traitanx)   #add in trait anxiety
summary(model2)                   #basic output
plot(model2)

#compare the difference between the two models
anova(model1,model2)
#test for the interaction, note that the main effects are incorrect
#because we need to 0 center the data
model2.5=lm(bdi~epiNeur*traitanx)
summary(model2.5)
anova(model2,model2.5)          #compare the two models

                                #rescale the data to do the analysis
cneur=scale(epiNeur,scale=F)    #0 center epiNeur
```

```
zneur=scale(epiNeur,scale=T)    #standardize epiNeur
ctrait = scale(traitanx,scale=F)#0 center traitAnx

model3=lm(bdi~cneur+ctrait+cneur*ctrait)
#explicitly list the additive and interactive terms
summary(model3)
plot(model3)
```

Exercise 7 ()
One hundred units of a mechanical component are tested for life and the following data are obtained:

Life (hours)	Number of units failed
0-999	22
1000-1999	25
2000-2999	18
3000-3999	12
4000-4999	23

Determine whether the life of the component follows exponential distribution with constant failure rate at 15% significance.

Exercise 8 (Data structures)
Imagine the following scenario.
Suppose you collect observations during some period of time, say, every week. You start with n_1 observations during the first week, then you obtain n_2 new observations during the next week, etc. Thus, we have a sequence of observations $n_1, n_2, \cdots, n_k$ and n_i may or may not be equal. For simplicity, suppose that these observations follows a linear model

$$y = X\beta + \epsilon$$

with a LS-estimator $\hat{\beta} = (X^T X)^{-1} X^T y$.
After each week, an estimator can be computed, using the data of all previous observations. To complicate the setting, we allow to exclude some of the earlier observations, thus, n_i may decrease.
What data structure would you suggest to keep the data (in $\mathbb{R}$ and or in Matlab)?
Note: for those of you, who have experience with solving discrete PDEs, the above setting might remind the bookkeeping of a mesh, which is locally refined and derefined.