Ilmonen/ Lietzén/ Voutilainen/ Mellin

1. Computer exercises

A few words about R-programming

We solve the computer exercises of this course using the statistical software R. R is widely used and freely distributed programming language that is particularly suitable for statistical analysis. You should be able to find R from every computer located in the Undergraduate Centre (Otakaari 1) or Maarintalo. The exercises of this course can be solved by using the basic R-software, which you can download free of charge to your personal computer. There exists many different integrated development environments (IDE) for the R programming language. We recommend that you use the one called RStudio. RStudio is also free of charge. Note that, in order to use RStudio, you need the basic R-software installed. The path to RStudio on Aalto computers is:

```
c:\Program_files\RStudio\bin\rstudio.exe
```

Basic commands

Below you can find some basic commands that may help you get started.

Assigning variables

Assigning variables is done as, e.g., in Matlab:

```
x = 5
```

or alternatively

```
x <- 5
```

You can create vectors with c()

```
x <- c(1,2,3)
```

A matrix is created by giving its elements as an vector

```
X <- matrix(c(1,1,1,2,2,2,3,3,3,4), nrow=5, ncol=2, byrow=FALSE)
```

Arguments `nrow` and `ncol` defines the number of columns and rows in the matrix. Argument `byrow` tells how the matrix is filled, in this case one column at a time from left to right. Note that in R lowercase and uppercase letters are not the same! You can comment the code with #.

Setting the working directory

The working directory can be set with `setwd()`, where the path of the directory comes inside the parenthesis. In RStudio this can be done also by choosing from the upper panel: Session-Set Working Directory-Choose Directory.

Scripts

It is recommended to write your solutions as R-scripts. A new script is created from the upper panel: `File- New File- R Script`. You can run a script with the command:

```
source("nameofthescript.R").
```

Single lines of a script can be executed by pressing `ctrl+ENTER`.

Importing data

The first step of statistical analysis is importing your data into the software. Suppose that you have a file `data.txt` in your working directory. You can import the data to R with the command,

```
read.table("data.txt", header=T, sep="\t")
```

The argument `header=T` indicates the first line of the file is a header. If the first line contains data, you can write `header=F` instead. The argument `sep = "\t"` indicates that the observations are separated with a tabulator. Note that the default value is comma.

Searching for commands and using help

The command `help()` is the most useful command in R. For example,

```
help(matrix)
```

describes how to create matrices with `matrix` command. Additionally, `help.search()` is useful, as you can use it search commands. For example

```
help.search("transpose")
```

searches for help-files that contain the word "transpose". However, nowadays web search engines are usually more convenient than `help.search()`.

Other

For a additional information on R programming, see for example,

`http://a-little-book-of-r-for-time-series.readthedocs.org/en/latest/`

The corresponding book contains, for example, installation instructions for your home computer.

Demo exercises

1.1 The file `emissions.txt` contains a study, where the nitrous emissions (NOx) of a diesel engine were measured under different values of humidity (Humidity), temperature (Temp) and pressure (Pressure).

a) Get familiar with the data by creating histograms for the different variables. Do the variables look normally distributed?

b) Estimate a linear model, where the amount of nitrous emissions is explained with variables humidity, temperature and pressure.

c) What is the coefficient of determination of the estimated model?

d) Use the permutation test to determine which regression coefficients are statistically significant at a significance level of 5%. Use 2000 as the number of permutations.

e) Remove explanatory variables that are not significant at a significance level of 5%. Estimate a new linear model, without the omitted variables. Solve the following parts without the omitted variables.

f) Compute standard deviations for the regression coefficient estimators by using formulas from the lecture slides.

g) Assume that the normality assumption holds. Compute 95% confidence intervals for the regression coefficients with the `confint()` command.

h) Assume that the normality assumption holds. Repeat the step (g) by using formulas from the lecture slides.

i) Compute 95% confidence intervals for the regression coefficients by bootstrapping and compare the results to parts (g) ja (h). Use a loop of length 2000 in your bootstrap code. Plot histograms of the bootstrap estimates.

Solution.

a) First, we import the file emissions.txt. In addition, set `seed(123)` to get identical results as in the models.

```
emis=read.table("emissions.txt", header=T, sep="\t")
set.seed(123)
```

The data contains 5 variables: ObsNo, NOx, Humidity, Temp and Pressure. The command `summary()` provides some important statistics.

```
summary(emis)

     ObsNo            NOx             Humidity           Temp
 Min.   : 1.00   Min.   :0.6900   Min.   : 33.85   Min.   :65.44
 1st Qu.: 5.75   1st Qu.:0.7175   1st Qu.: 77.94   1st Qu.:73.50
 Median :10.50   Median :0.7600   Median : 96.22   Median :77.82
 Mean   :10.50   Mean   :0.7910   Mean   : 93.98   Mean   :78.57
 3rd Qu.:15.25   3rd Qu.:0.8275   3rd Qu.:111.55   3rd Qu.:86.03
 Max.   :20.00   Max.   :1.0100   Max.   :139.47   Max.   :89.28
   Pressure
 Min.   :28.87
 1st Qu.:29.03
 Median :29.07
 Mean   :29.15
 3rd Qu.:29.16
 Max.   :29.98
```

Histograms can be plotted with the command

```
hist(emis[,2])
```

or

```
hist(emis[,"NOx"])
```

The correlation matrix is given by

```
cor(emis)
              ObsNo        NOx   Humidity        Temp    Pressure
ObsNo     1.0000000 -0.1260915  0.3686937  0.25908229 -0.24143057
NOx      -0.1260915  1.0000000 -0.8729408  0.65591970  0.27825058
Humidity  0.3686937 -0.8729408  1.0000000 -0.47282672 -0.27050695
Temp      0.2590823  0.6559197 -0.4728267  1.00000000  0.02677886
Pressure -0.2414306  0.2782506 -0.2705070  0.02677886  1.00000000
```

The variable **NOx** correlates negatively with the variable **Humidity** and positively with the variables **Temp** ja **Pressure**. The variables do not seem to be normal. However, as the sample size is very small, one should avoid making strong conclusions.

b) A linear regression model can be estimated with the commands presented below. In the command, we have a "∼"-sign between the response variable and the explanatory variables, and "+-sign separating the explanatory variables. The command,

```
fit1 <- lm(NOx~Humidity+Temp+Pressure, data=emis)
summary(fit1)
```

produces,

```
Call:
lm(formula = NOx ~ Humidity + Temp + Pressure, data = emis)

Residuals:
      Min        1Q    Median        3Q       Max
-0.061616 -0.034795  0.003699  0.029233  0.077782

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.2707790  1.2538066  -0.216   0.8317
Humidity    -0.0025280  0.0004242  -5.959    2e-05 ***
Temp         0.0043960  0.0015320   2.869   0.0111 *
Pressure     0.0327257  0.0418425   0.782   0.4456
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 0.04249 on 16 degrees of freedom
Multiple R-squared:  0.8441,Adjusted R-squared:  0.8149
F-statistic: 28.89 on 3 and 16 DF,  p-value: 1.079e-06
```

c) The coefficient of determination of the model is 84,4%, which corresponds to “Multiple R-square” in the output.

d) In order to test the significance of the explanatory variables, we use the permutation test. When using the permutation test, we do not need any distributional assumptions about the residuals. In general, strong distributional assumptions should be avoided when possible, especially, when the sample size is small.

We permute the the explanatory variables one at a time and see how that affects the coefficient of determination.

```
k <- 2000
y.mean <- mean(emis[,"NOx"])
SST <- sum((emis[,"NOx"]-y.mean)^2)
SSE <- sum((fit1$res)^2)
Rsquare1 <- 1-SSE/SST
perm <- matrix(NA,nrow =2000,ncol=3)

for(i in 1:k){
  tmp1 <- emis
  tmp2 <- emis
  tmp3 <- emis

  tmp1$Pressure <- sample(tmp1$Pressure)
  tmp2$Humidity <- sample(tmp2$Humidity)
  tmp3$Temp <- sample(tmp3$Temp)

  tmpfit1 <- lm(NOx ~ Humidity+Temp+Pressure, data=tmp1)
  tmpfit2 <- lm(NOx ~ Humidity+Temp+Pressure, data=tmp2)
  tmpfit3 <- lm(NOx ~ Humidity+Temp+Pressure, data=tmp3)

  perm[i,1] <- summary(tmpfit1)$r.squared
  perm[i,2] <- summary(tmpfit2)$r.squared
  perm[i,3] <- summary(tmpfit3)$r.squared
}

pre <- 1-sum(perm[,1] < Rsquare1)/k #0.4635
hum <- 1-sum(perm[,2] < Rsquare1)/k #0
temp <- 1-sum(perm[,3] < Rsquare1)/k #0.0115
```

Variables pre, hum and temp are the proportions of permutations where the coefficient of determination is larger than the original one. Note that changing the seed may change the results to some extent. The variables represent the p-value of the null hypothesis (H_0: regression coefficient is not significant, see the lecture slides.)

Alternatively, you could follow the lecture slides and order the coefficient of determination values from the smallest to the largest and calculate the empirical 95th percentile from the sample and then compare the calculated value with the original coefficient of determination. If the original coefficient of determination is larger than the calculated percentile, the null hypothesis is rejected.

```
# Choose the significance level alpha = 5% and order the observations:
preord <- sort(perm[,1])
humord <- sort(perm[,2])
tempord <- sort(perm[,3])

# (note that the sorting is not necessary for the next step)

# The null hypothesis (H_0) of beta_j = 0 is rejected if the
# original R-squared is larger than the calculated 95th percentile
Rsquare1 > quantile(preord,0.95)  # H_0 Accepted
Rsquare1 > quantile(humord,0.95)  # H_0 Rejected
Rsquare1 > quantile(tempord,0.95) # H_0 Rejected
```

By the permutation test, the null hypothesis of the variable Pressure is accepted at a 5% significance level and hence, Pressure is removed from the model. The null hypothesis regarding the other explanatory variables are rejected. Model selection will be a relevant topic in the second exercise session.

e)
```
fit2=lm(NOx~Humidity+Temp, data=emis)
summary(fit2)
Call:
lm(formula = NOx ~ Humidity + Temp, data = emis)

Residuals:
      Min        1Q    Median        3Q       Max
-0.065394 -0.018456 -0.001075  0.032302  0.072869

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.703544   0.140272   5.016 0.000106 ***
```

```
Humidity    -0.002625   0.000401  -6.547 4.98e-06 ***
Temp         0.004253   0.001504   2.829 0.011586 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04201 on 17 degrees of freedom
Multiple R-squared:  0.8382,Adjusted R-squared:  0.8191
F-statistic: 44.03 on 2 and 17 DF,  p-value: 1.891e-07
```

f) First, create a new data matrix, where we have removed some of the original variables. We perform also a data type conversion from data frame to matrix (more about different data types on Introduction to R-programming course) and add a column (intercept) for the constant term of the model.

```
tmp <- as.matrix(emis[,c(3,4)])
n <- nrow(emis)
Intercept <- rep(1,n)
emis2 <- cbind(Intercept,tmp)
p <- ncol(emis2)
```

Next, we find an unbiased estimate s^2 for the variance of the residuals,

```
res <- fit2$res
s2 <- sum(res^2)/(n-p)
```

Unbiased estimators for the variances of the regression coefficient estimators are given by the diagonal entries of the following matrix,

$$D^2(\mathbf{b}) = s^2(\mathbf{X}^\top\mathbf{X})^{-1}.$$

```
stdev <- sqrt(diag( s2*solve(t(emis2)%*%emis2)))
```

Which are identical with the values of the table in part (e).

g) `confinterval <- confint(fit2, level=0.95)`

Gives the following output:

```
                    2.5 %       97.5 %
(Intercept)  0.407595872  0.999491779
Humidity    -0.003471148 -0.001779119
Temp         0.001080766  0.007425475
```

h) The same results can be obtained with the following commands.

```
coef <- fit2$coef
up <- coef + stdev * qt(0.975,n-p)
down <- coef - stdev * qt(0.975,n-p)
```

where `qt()` gives the quantile of t-distribution with $n - p$ degrees of freedom.

i) Confidence intervals by using bootstrapping:

```
# (i)
# The logic in bootstrapping is mainly the same as in permutation test

# Below are steps 1-5 from the lecture slides

k <- 2000
bootmat <- matrix(NA,nrow=k,ncol=3) # Empty matrix for the results

y <- emis$NOx
X <- emis2

set.seed(123)

for(i in 1:(k-1)){
  # 1. Select n data points randomly with replacement from the
  # original observations
  # (Note that in the permutation test the data points are selected
  # without replacement)
  ind <- sample(1:n,replace = TRUE) # Indices for the random rows

  Xtmp <- X[ind,]
  ytmp <- y[ind]

  # 2. Calculate a new parameter vector estimate from the new sample
  btmp <- solve(t(Xtmp)%*%Xtmp)%*%t(Xtmp)%*%ytmp

  # 3. Save the results to bootmat and repeat k-1 times
  bootmat[i,] <- t(btmp)
}

# 4. Include the original estimate and order the obtained estimates
# from the smallest to the largest
boriginal <- solve(t(X)%*%X)%*%t(X)%*%y

bootmat[k,] <- t(boriginal)

boot1 <- sort(bootmat[,1])
boot2 <- sort(bootmat[,2])
boot3 <- sort(bootmat[,3])

# 5. Set alpha=5% and set the lower end of the confidence interval
# to be smaller or equal to the [0.025*2000] ordered estimate.
# Set the upper end to be larger or equal to the [0.975*2000]
# ordered estimate.
```

```
qconst <- quantile(boot1, probs = c(0.025,0.975))
qhum <- quantile(boot2, probs = c(0.025,0.975))
qtemp <- quantile(boot3, probs = c(0.025,0.975))

# Note that step 4 is not necessary in R,
# you would get the same results by inputting the
# columns of bootmat into the quantile-function.

# Here, we have wanted to emphasize the steps of the lecture slides
# and thus have included step 4 separately

# Plot the histograms of the bootstrap estimates
hist(bootmat[,1])
hist(bootmat[,2])
hist(bootmat[,3])
```

Homework

1.2 The file tobacco.txt contains the following information from 11 different countries:
CONSUMPTION = consumption of cigarettes `per capita` in 1930,
ILL = Lung cancer cases `per 100 000 individuals` in 1950.

The numbering of the countries is the following:

1	Iceland
2	Norway
3	Sweden
4	Canada
5	Denmark
6	Austria
7	USA
8	Holland
9	Switzerland
10	Finland
11	England

Note that the file contains additional information that is not relevant from the viewpoint of this exercise.

(a) Formulate a linear regression model, where the variable ILL is explained with the variable CONSUMPTION. Include a constant term in your model.

(b) Estimate the regression coefficients of the model by using the least squares method and give interpretations for the estimated regression coefficients.

(c) What is the coefficient of determination of the model?

(d) Is the model statistically significant according to the F-test? Use 1% as the level of significance.

(e) Is the variable CONSUMPTION statistically significant according to the t-test? Compare the p-value with the p-value obtained in part (d) and explain the connection between them.

(f) Plot the estimated regression line together with the data.

(g) Explain the concept of confidence interval.

(h) Suppose that the normality assumptions holds. Form a 95% confidence interval for the slope of the regression line. Give also the 99% confidence interval. Note that the confidence interval is notably wide for the constant term. Can you give some explanation for this?

(i) Compute 95% confidence intervals for the regression coefficients with bootstrapping (2000 repetitions). Compare to part (h).

(j) What advantages bootstrapping has when compared to the conventional way of determining confidence intervals?