

Practical 3

R and package *Rcmdr* (R Commander): descriptive statistics, graphs, comparison of means, ...

PART 1

--- Basic of *R Commander* ---

1) Running the *R Commander*

Open the *R*.

Open the package *Rcmdr* typing into the command line

```
> library(Rcmdr)
```

or choosing the packages from the additionally installed packages list

(*Packages -> Load package ... -> Rcmdr*)

The menu-based *R Commander* window should appear.



2) Opening the datasets in *R Commander*

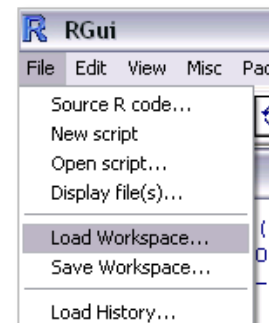
- a) If you have saved the '*R workspace*' in last week, you can just open it to take into use the dataset '*students*' imported into *R* in last week:

File -> Load Workspace...

If you haven't the corresponding Rdata-file, look at the sections 2) b) and c).

NB! Open the RData-file in *R Commander*, and not in *R*!

After that fix the active dataset '*students*' by choosing it in *R Commander*



Data set: <No active dataset>

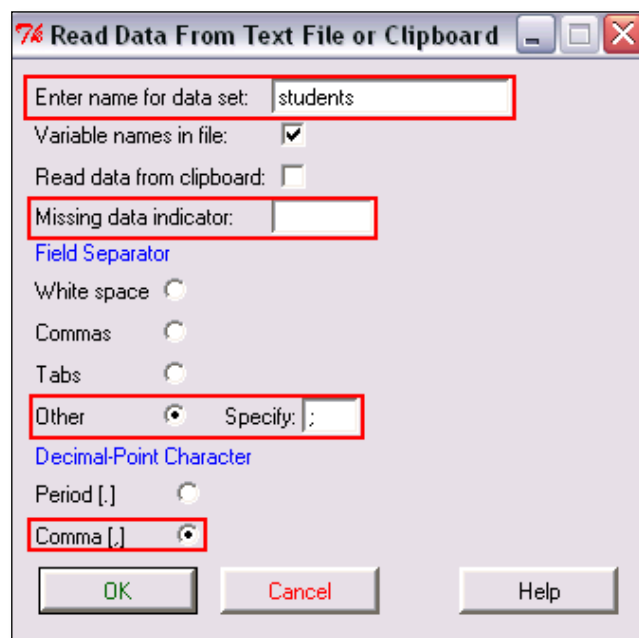
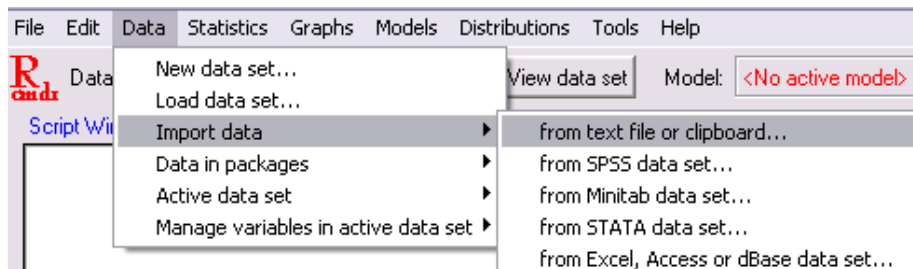


- b) If you haven't the saved *R workspace*, you should import the students database again (also other may try this instead of (or in addition to) opening the previously imported dataset).

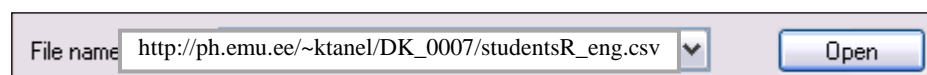
The simplest way is to import the correct comma separated data fail from internet

http://ph.emu.ee/~ktanel/DK_0007/studentsR_eng.csv

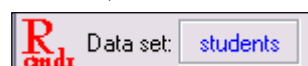
using the commands in *R Commander*:



↓ (type or copy the file address ...)



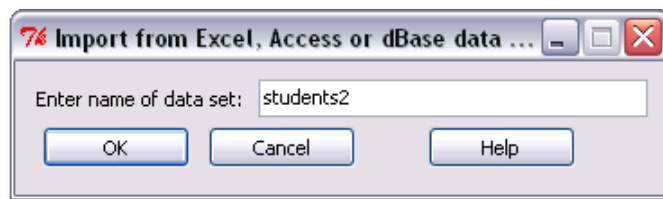
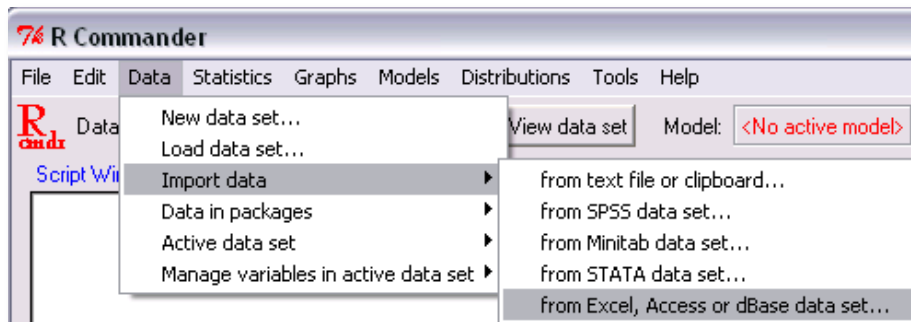
↓ (as the result the dataset is imported and treated as the default dataset in *R Commander*)



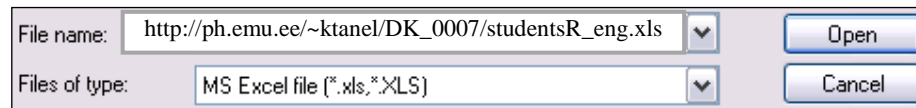
c) But, the R Commander allows also the data import straight from the MS Excel!

Try this variant also

(you may name the dataset in *R* with new name – for example 'students2').

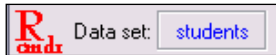


↓ (type or copy the address of the *Excel* file)

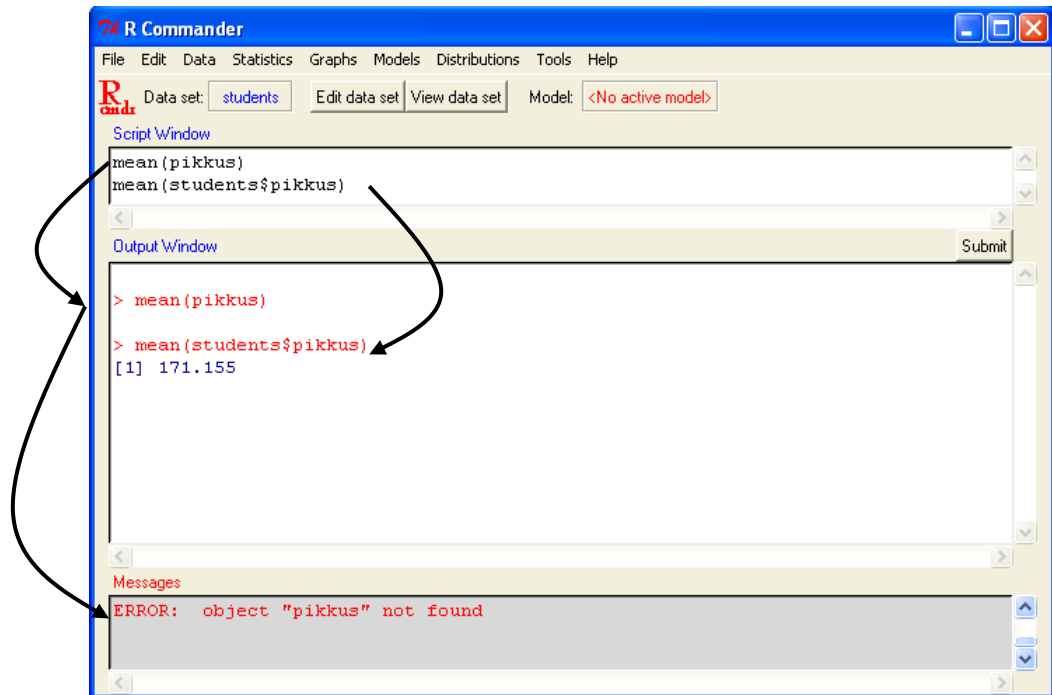


If the imported *Excel* fail contains several worksheets, the *R Commander* will ask which of them to import.

3) Fixing the default dataset

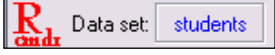
Fixing the active dataset in *R Commander* with button  makes this dataset default for *R Commander* menus but not for *R* or *R Commander* 'Script Window'!

So, typing the desired command into the *R Commander* skript window and running it with the key combination 'Ctrl'+'R' or with button , the *R* assumes that additionally to the trait name also the dataset name is specified:



To use the trait names in 'Script Window' without specifying the dataset name you should type and run the `attach` command:

```
attach(students)
```

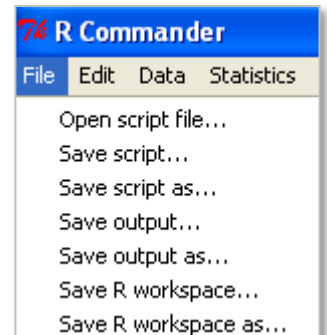
Selection  in *R Commander* is not equivalent with the command `attach(students)!`

4) Saving

a) Saving and opening the script

Content of the *R Commander's* 'Script Window' (where *R Commander* writes the commands based on your choices from menus and where you can add the commands and comments like in usual *R* script window) can be saved as the *R script* (with file extension *.R*, but in nature as a text file).

Also the already written script can be opened in *R Commander*.



b) Saving the analyses results

Content of the 'Output Window', which contains all applied commands and got results, is saved as a text file (with file extension *.txt*).

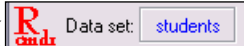
c) Saving the *R* workspace

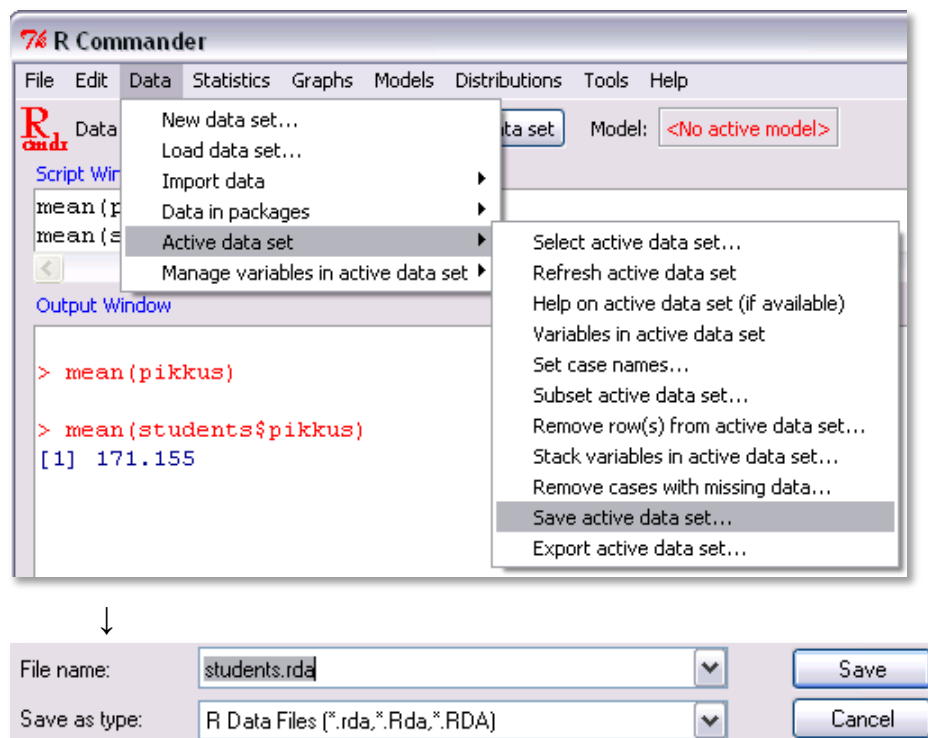
You can save the *R workspace* (as *.RData* file), which contains all used datasets

(it can be convenient to take all desired datasets into use in next *R* session just by opening the corresponding *R workspace* again)

and all defined variables (sometimes useful to continue calculations or modeling).

d) Saving the dataset

The active dataset () can be saved as the *R* datafile:



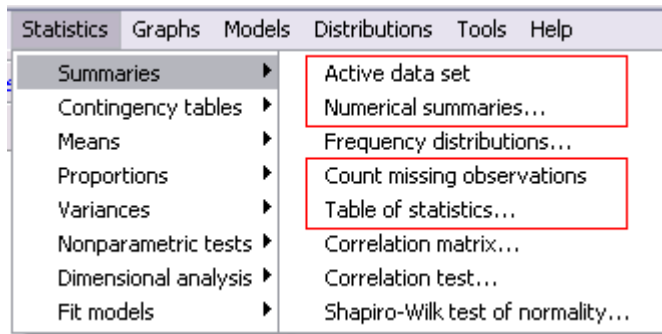
And once saved data file can be just loaded into the *R* (*Data -> Load data set...*).

PART 2**--- Basic descriptive statistics with *R Commander* ---**

1) Try to find some descriptive statistics (as in previous practical) with help of *R Commander*:

Statistics -> Summaries

Apply the commands and try to understand, what they are doing.



- The results are printed into the *R Commander* 'Output Window'
- and the commands' scripts into the *R Commander* 'Script Window'.

If you want, you may change the scripts printed out by *R Commander* and run them again by selecting the desired row(s) and pressing 'Ctrl'+R'

or pressing the button .

PART 3

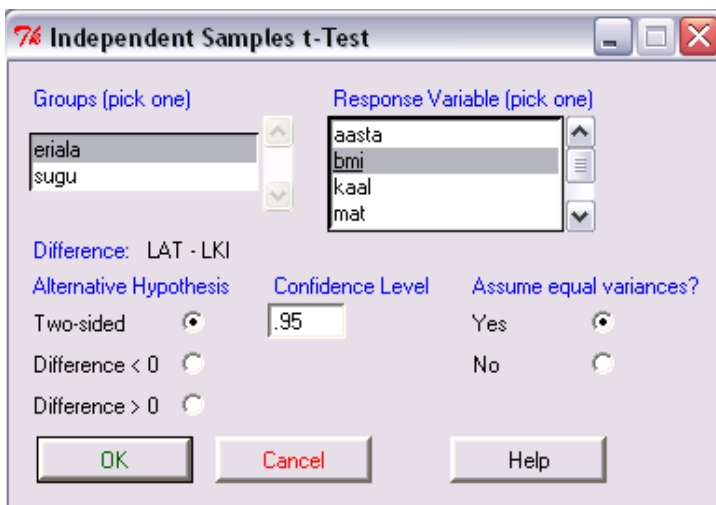
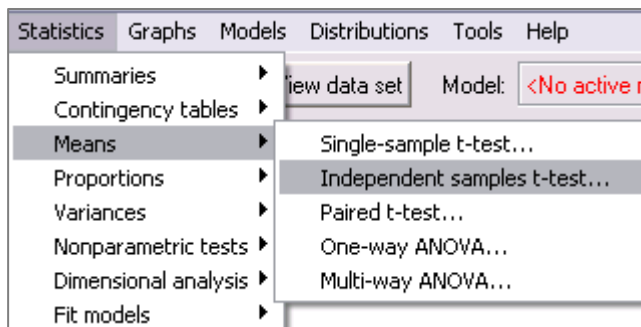
--- Comparison of two groups with *R Commander* ---

Most of the standard tests used in two groups comparison can be found in *R Commander* menus. More specific tests can be performed just by typing the corresponding command into the ‘*Script Window*’ and running it.

❖ Comparison of means with t-test

To compare the average body mass indexes (‘bmi’) of different specialities (‘special’, ‘eriala’ in Estonian: veterinary medicine denoted as ‘LAT’, animal science ‘LKI’) the t-test should be performed:

Statistics -> Means -> Independent samples t-test ...



```
> t.test(bmi~eriala, alternative='two.sided', conf.level=.95, var.equal=TRUE, data=students)

Two Sample t-test

data:  bmi by eriala
t = 1.9781, df = 94, p-value = 0.05084
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.00401474  2.15660806
sample estimates:
mean in group LAT mean in group LKI
      22.3613      21.2850
```

- For grouping variable (‘*Groups*’) the *R Commander* allows select only bivariate nonnumeric traits and for response variable (‘*Response Variable*’) which average values to compare only numeric traits in activ dataset.
- *R Commander* shows also the direction of calculable difference (‘LAT-LKI’; because the values of grouping factor are ordered alphabetically).
- The two-sided and two different one-sided hypothesis can be chosen (writing the script yourself there is the option `alternative='two.sided'` in function `t.test`).
- User can also determine the confidence level for confidence interval (additional option `conf.level=`) and
- make the additional assumption about the equality of variances (additional option `var.equal=TRUE`).

As a result *R* calculates the p-value, estimates the confidence interval of the means’ difference and presents the average body mass indexes for both specialities.

Did you found all these measures? What can you conclude?

From the command written automatically by *R Commander* into the script window follows an alternative and more general way to compare the groups of one numerical variable (for example 'bmi') when the grouping is based on the second variable ('special'):

```
bmi ~ special
```

Also accepts number of functions the dataset specification of the form:

```
data = students
```

So, the standard t-test command to compare the average body mass indexes of LAT- and LKI-students

```
t.test(students$bmi[students$special=='LAT'], students$bmi[students$special=='LKI'],  
var.equal=TRUE)
```

can be alternatively written as

```
t.test(bmi~special, var.equal=TRUE, data=students)
```

(the additional options of function `t.test` written by *R Commander*

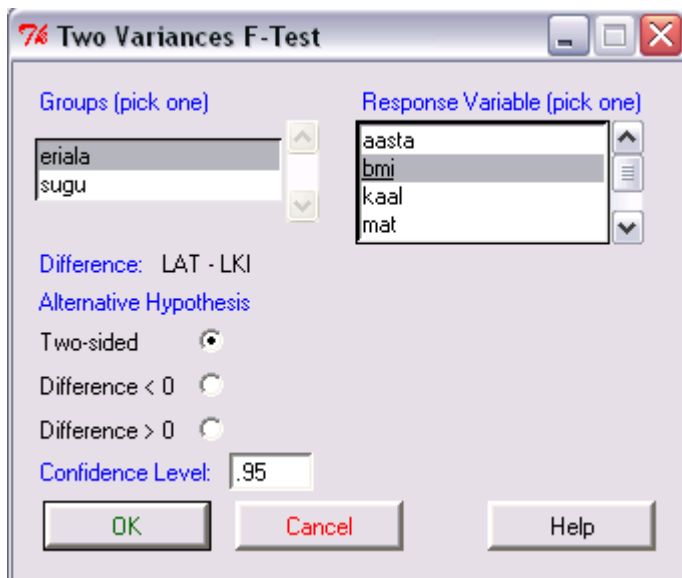
```
alternative='two.sided' ja conf.level=.95
```

are not necessary as these values are the default values, but it is good to know how to test one-sided hypothesis or how to change the default significance level.)

❖ Comparison of variances

To perform the F-test comparing the variances in two groups:

Statistics -> Variances -> Two-variances F-test ...



```
> var.test(bmi ~ eriala, alternative='two.sided', conf.level=.95, data=students)

      F test to compare two variances

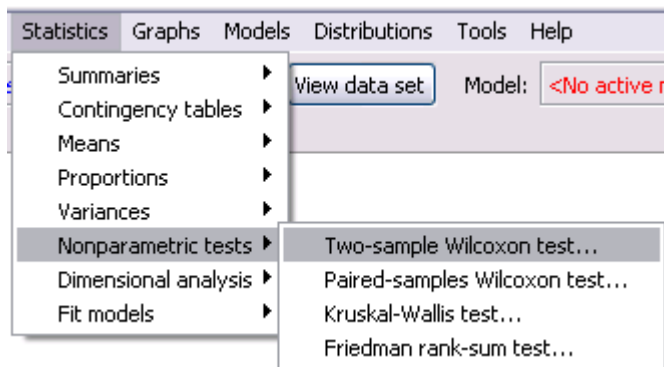
data:  bmi by eriala
F = 1.3446, num df = 44, denom df = 50, p-value = 0.31
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.7569526 2.4146552
sample estimates:
ratio of variances
      1.344644
```

The output contains the p-value, variances' ratio and its confidence interval (if the variances' ratio does not differ from one, the variances are not different :).

What decision can you make? Are the variances of body mass indexes statistically significantly different?

❖ Nonparametric tests

- 1) The mostly used nonparametric test for two groups comparison – Wilcoxon test – can be performed in *R Commander* using menus *Statistics* -> *Nonparametric tests*:



Other nonparametric tests in the same submenu can be used

to compare two dependent groups (*Paired-samples Wilcoxon test*),

to compare more than two independent groups (nonparametric analog to the analysis of variance, *Kruskal-Wallis test*)

and to compare more than two dependent groups (nonparametric analog to the repeated measures analysis of variance, *Friedman rank-sum test*).

As an alternative the Wilcoxon test can be performed by typing into the script window the following command

```
wilcox.test(students$bmi[students$special=='LAT'],students$bmi[students$special=='LKI'])
```

or the command

```
wilcox.test(bmi~special, alternative="two.sided", data=students)
```

```
> wilcox.test(students$bmi[students$seriala=='LAT'],students$bmi[students$seriala=='LKI'])

      Wilcoxon rank sum test with continuity correction

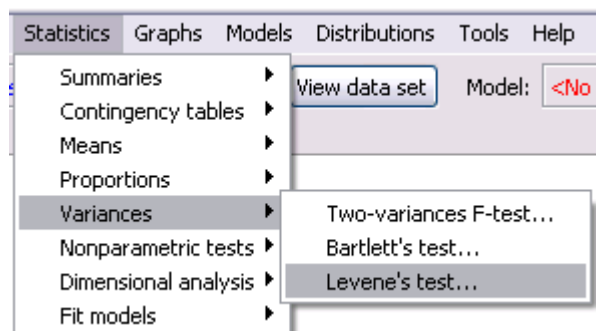
data:  students$bmi[students$seriala == "LAT"] and students$bmi[students$seriala == "LKI"]
W = 1384.5, p-value = 0.08248
alternative hypothesis: true location shift is not equal to 0
```

Are the body mass indexes of LAT and LKI group statistically significantly different?

- 2) The nonparametric test for variances comparison is the Levene's test. It allows compare variances in two groups but also in more than two groups without assuming normality (its parametric normality assuming analog in *R Commander* is the Bartlett's test).

Compare the variances of LAT and LKI speciality students' body mass indexes with Levene's test.

Is the conclusion the same as with F-test (performed earlier) – the variances are not different?



```
> tapply(students$bmi, students$seriala, var, na.rm=TRUE)
      LAT      LKI
8.194334 6.094057

> levene.test(students$bmi, students$seriala)
Levene's Test for Homogeneity of Variance
  Df F value Pr(>F)
group 1 0.0015 0.9696
  94
```

NB! Look at the syntax of the `levene.test`

The presentation of arguments is different from other so far used tests!

To compare the variances without assuming normality also the Fligner-Killeen test can be used. But this test is not realised in *R Commander* menus (like a lot of other tests – in *R* there are more than 100 tests for two groups comparison ...). The Fligner-Killeen test can be performed typing the corresponding command into the script window and submitting it (the syntax of function `fligner.test` is analogous to the function `levene.test` syntax):

```
fligner.test(students$bmi, students$special)
```

```
> fligner.test(students$bmi, students$eriala)

      Fligner-Killeen test of homogeneity of variances

data:  students$bmi and students$eriala
Fligner-Killeen:med chi-squared = 0.0474, df = 1, p-value = 0.8277
```

NB! Actually may the Levene's and Fligner-Killeen tests applied also with commands

```
fligner.test(bmi ~ eriala, data=students)
```

and

```
levene.test(bmi ~ eriala, data=students)
```

Only in some reason differently from other tests *R Commander* does not use for Levene's test such syntax ...

3) To compare the distributions in two groups the Kolmogorov-Smirnov test can be used (without assuming normality). Also this test is not realised in *R Commander*'s menu system and the corresponding command should be run in script window:

```
ks.test(students$bmi[students$eriala=='LAT'],
        students$bmi[students$eriala=='LKI'])
```

```
> ks.test(students$bmi[students$eriala=='LAT'],students$bmi[students$eriala=='LKI'])

      Two-sample Kolmogorov-Smirnov test

data:  students$bmi[students$eriala == "LAT"] and students$bmi[students$eriala == "LKI"]
D = 0.2758, p-value = 0.05264
alternative hypothesis: two-sided
```

As $p = 0,053$, then the distributions of body mass indexes of veterinary medicine (LAT) and animal breeding (LKI) students are not statistically significantly different.

❖ Comparison of distributions

- 1) The Kolmogorov-Smirnov test can be used also for the normality testing (or for comparing the sample distribution with some other theoretical distribution):

```
ks.test(students$bmi, pnorm, mean=mean(students$bmi, na.rm=TRUE),
       sd=sd(students$bmi, na.rm=TRUE))
```

```
> ks.test(students$bmi, pnorm, mean=mean(students$bmi, na.rm=TRUE), sd=sd(students$bmi, na.rm=TRUE))

One-sample Kolmogorov-Smirnov test

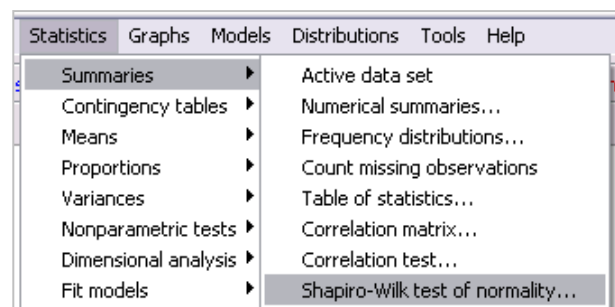
data:  students$bmi
D = 0.0857, p-value = 0.4804
alternative hypothesis: two-sided
```

Conclusion: as $p = 0,48 > 0,05$, then based on the Kolmogorov-Smirnov test there is no reason to reject the null hypothesis (students body mass indexes follow normal distribution).

- 2) The alternative test used in normality testing is the Shapiro-Wilk normality test:

```
shapiro.test(students$bmi)
```

This test can be selected also from the *R Commander* menus:



Result:

```
> shapiro.test(students$bmi)

Shapiro-Wilk normality test

data:  students$bmi
W = 0.962, p-value = 0.00699
```

As visible from the test results the Shapiro-Wilk normality test is considerably more sensitive to the differences from normality. Differently from the Kolmogorov-Smirnov test we can reject the null hypothesis that the students' body mass indexes are normally distributed ($p = 0,007$).

The reason of different results is the different computing algorithms and depending on this different power. As the Shapiro-Wilks test is specially constructed to compare the empirical distribution of data with the theoretical normal distribution, it compares only several aspects related with the normal distribution. The Kolmogorov-Smirnov test is very general and due to this also very robust test, which offers the coparison of empirical data distribution and theoretical normal distribution as one of the lot of options and can't test the details.

PART 4

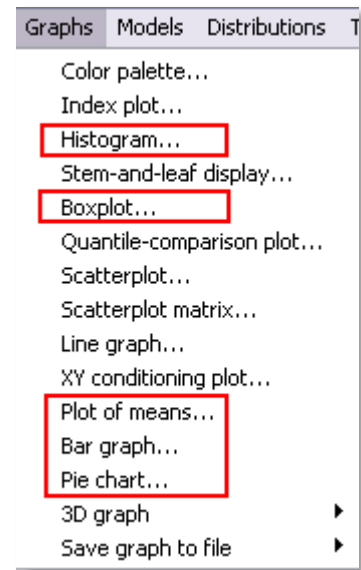
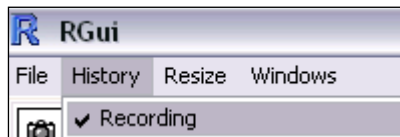
--- Basic graphs with *R Commander* ---

Try to construct some graphs (surrounded with red line in figure) in *R Commander*.

NB! *R Commander* writes the commands of ordered charts into the script window, where user can change and improve them and run again already without menu commands.

The graphs are produced into the graph window in *R* (not in *R Commander*!).

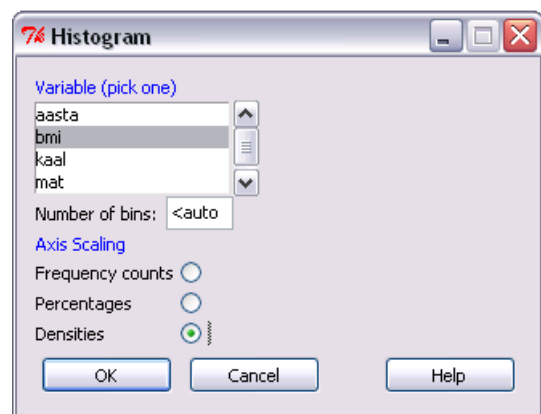
Asking the *R* not to overwrite the old graphs with the newly produced graphs, the selection *Recordings* in the *R* menu *History* (not in *R Commander* menu!) should be made:



Try to change the default *R Commander* graphs using the graphs options applied in last week (there is a very good webpage about basic graphics commands and their additional options: <http://www.ms.ut.ee/mart/R/Rgraafika.html> – even as the webpage is in Estonian, the examples should be understandable without understanding the comments).

Construct the histogram of students' body mass indexes, determining the y-axis scale as density:

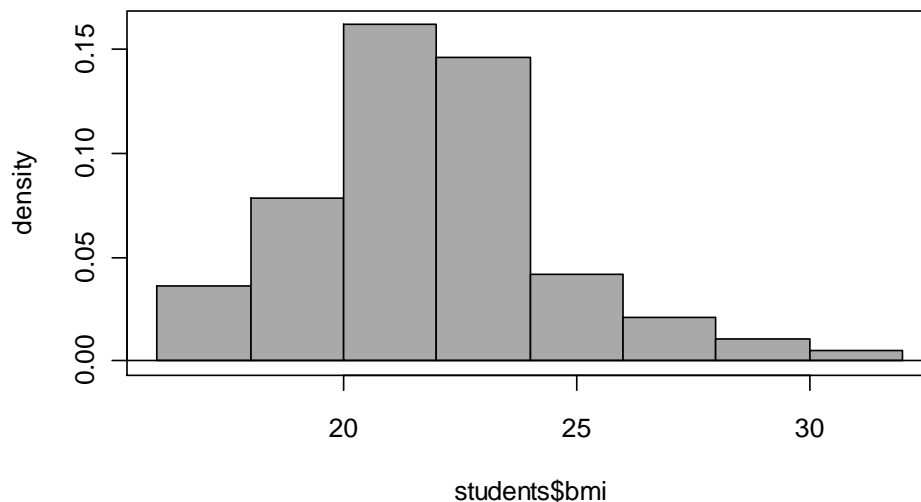
Graphs -> Histogram ->



As a result *R Commander* writes into the skript window the command

```
Hist(students$bmi, scale="density", breaks="Sturges", col="darkgray")
```

and produces into the *R* window the following graph.



R Commander's function `Hist` is the improved version of the base *R* function `hist`¹.

To improve the informativeness and look of the graph you can add arguments into the applied function in script window.

For example modifying the `Hist` command written into the script window by *R Commander* in the following way the result will be the figure where the initial histogram is improved with the empirical density function of 'bmi' (black line) and corresponding (with the same mean and standard deviation) normal density function (red line). Also the axis names and the range of the x-axis is changed.

```
Hist(students$bmi, scale="density", xlab="Kehamassiindeks", ylab="Jaotustihedus", xlim=c(14, 32))
lines(density(na.omit(students$bmi)), lwd=2)
x=seq(12, 32, length=300)
y=dnorm(x, mean=mean(students$bmi, na.rm=T), sd=sd(students$bmi, na.rm=T))
lines(x, y, lwd=2, col="red")
```

¹ Many *R Commander* functions are built on the corresponding *R* base version functions. Often are their names almost the same, the usual difference is that the *R Commander* functions start with capital letter when the *R* base version functions are written in small letters (for example `Hist` versus `hist`).

All *R* base version functions' options can be used also in *R Commander* functions, but additionally the *R Commander* functions can include some options working only with these functions and not with their analogues in *R* base version.

For example the additional options of *R* base version function `hist` – `freq=TRUE` and `freq=FALSE` – can be used also with the *R Commander* function `Hist`. But the analogues options (the result is the same) of the *R Commander* function `Hist` – `scale="frequency"` and `scale="density"` – did not work with the *R* base version function `hist`.

Additionally the alternative options of *R Commander* functions can have additional values. For example the options `scale="percent"` used with *R Commander* function `Hist` orders the scale of the y-axis in percents (for *R* base version function `hist` there is not available similar option).

