

Course on R for linguists

Organizer: Bjørn-Helge Mevik

Research Computing Services, USIT

Date: 15 November, 10-13,

17 November, 9-12,

Place: Henrik Wergelands Hus, PC stue 301

Basic instructions for the practical hands-on, part I

1. Log in to the machine
2. Enter R
3. Fetch data into R
4. Save data for future work

1. Log in to the machine

Participants will log to one UiO statistics server (statwin1), which has R installed as well as the materials for the course.

Log in from a machine with Windows

Use (en) Remote Desktop connection (Start menu -> All programs -> Accessories) or (no) Tilkobling til ekstrent skrivebord (Start -> Alle programmer -> Tilbehør) to the machine **statwin1.uio.no** with user **UIO\username** and your ordinary UiO password. You will find yourself in a Windows environment with access to your M: disk, as well as to the DATA (F:) with a directory named **R_kurs_BHM**.

Log in from a machine with Mac OS

Use Remote Desktop Connection for Mac (you can get it from the Web) the machine **statwin1.uio.no** with user **UIO\username** and your ordinary UiO password. You will find yourself in a Windows environment with access to your UiO home directory as the M: disk, as well as to the DATA (F:) with a directory named **R_kurs_BHM**.

Log in from a machine with Linux

Linux users have to log in to another machine first (**win.uio.no**, which is running Windows), and then do the process described above for Windows users. Use

```
rdesktop -k no -d uio -g 70% win.uio.no
```

and enter with your usual UiO username and password, and then follow the instructions above for accessing the **statwin.uio.no** machine from within Windows.

To go out of the **statwin1** machine, select the keylock icon in the

bottom and choose **Log out**.

2. Enter R

Click on the *R x64 2.11.0* icon on the left of the desktop, and you enter a window called Rgui with a smaller window R Console in it.

You are now inside the R interpreter (signalled by the > sign in red) and can start issuing commands.

Asking for help:

```
> help()
```

Exiting R:

```
> q()
```

3. Work with files

Checking what is R's default folder

```
> getwd()
[1] "e:/work/dssantos"
```

Create your own folder for R, for example **Rwork** in Windows, and set it as your default folder for R

```
> setwd("m:Rwork/")
```

Copy some files there, and read them into R, or simply input them from the data folder (F:) directly

```
> simple <- read.table("simple.txt")
```

OR

```
> simple <- read.table("f:R_kurs_BHM/Other/simple.txt")
```

Check whether it read what you expected

```
> simple
```

```
> simple <- read.table("f:R_kurs_BHM/Other/simple.txt",
header=TRUE)
```

```
> simple ## looks better
```

Try out some simple plots.

```
> plot(colour ~ variety, data = simple)
```

Try to save them in your folder. ("Save as postscript" with righthand click)

Create the same picture as a pdf file

```
> pdf("simple.pdf")
```

```
> plot(colour ~ theme, data = simple)
```

```
> devoff() ## Needed to finalize the plot
```

4. Calculate and manipulate data

Calculate statistics like mean, min, max and sd of the numeric variables in the data frame `simple`. For instance

```
> mean(simple$colour)
```

The `summary()` function prints out summary information about a variable. Try it on some of the columns of `simple`. What happens if you say `summary(simple)`?

Use the `fix()` function to edit `simple`, changing some of the values. Recalculate the statistics.

5. Save/preserve your work in R

List your variables:

```
> ls()
```

To go out of R and back again, without losing your work

```
> save.image("everything.Rdata")
```

Go out of R and back again

```
> load("everything.Rdata")
```

```
> ls() # The variables are back
```

In case you want to keep a log of what you do for subsequent reference, you can do, before you leave R

```
> savehistory()
```

OR

```
> savehistory("myhist.Rhistory")
```

The commands you type are then logged in that file for future reference or repetition.

You can also fetch previous histories for ease of command repetition, by doing at any time in your R session

```
> loadhistory
```

OR

```
> loadhistory("myhist.Rhistory")
```

Finally, you may want to simply write datafiles that your processing created, such as tables or matrices

```
> onlyfb <- subset(simple, theme=="football", 4:5)
```

```
> write.table(onlyfb, file="fbcol.txt")
```

6. Plotting and examining data

Read in the two data sets

```
f:/R_kurs_BHM/Gries/03-1_uh(m).txt
```

and

```
f:/R_kurs_BHM/Gries/03-2-3_reactiontimes.txt
```

Test the different summary functions on the columns. Try out the different plots on the data.