

GEO 4830

Analytical methods in geochemistry Short introduction to use of the statistical software R, paper 1

Instructor: Knut-Endre Sjøstad
E-Mail: knutesj@geo.uio.no
Phone: 98437541
Office: [Location]
Office Hours: [Hours]

Overview

This short paper will give you an introduction to the statistical software R. R is a Open Source software under [GNU](#), and it works with Microsoft Windows, OSX and different distributions of LINUX.

Goals

In this paper you will learn how to install R on a computer running Windows or OSX (mac). For those using Linux, R will be found in the program center.

Further you will learn how to make some simple calculations of mean values, standard deviations standard error etc.

Download

Go to <http://www.r-project.org/> and choose a CRAN-mirror (middle on the web-page) and choose a preferred location. Alternatively go directly to <http://cran.uib.no/> and choose your preferred platform. On the next page choose base and then “[Download R 2.13.1 for Windows](#)”. If you are using a mac, please choose “[Download R for MacOS X](#)”, click R-2.1.3.pkg and follow the instructions. For those using LINUX, please follow the instructions on <http://cran.uib.no/bin/linux/>.

Materials


You need access to a PC running either Microsoft Windows, mac OSx or Linux

Report

Wednesday 07.09.2011

Groups of two students shall deliver a script file that contain code that perform some statistical calculation

A first look

You may be a bit surprised when you open the R-program by (double) click on the -icon. What appears on your screen is nearly a blank window, and only few menus and buttons. It seems rather cryptic and old fashion. Don't worry, that is normal (both the screen and your confusion). Lets start at the beginning. The screen look like in fig. 1. This is the opening screen on a mac OSX-system. There are some small differences on different platforms, but they are of no significance for us.

Well, now what? First of all, what is R? It's a huge calculator and a plotting device for those that like to play around with numbers and statistics. The nice thing about R is that it is a very flexible system that can do calculations for the non-statistician that wants to do simple tasks. It's also meant for those making advanced models and heavy statistics. Why not using EXCEL you may ask. It is not so obvious why you shouldn't (at the beginning), but when you learn to know R a bit more, you will see that it is tenfold as flexible as EXCEL when it comes to statistics and statistical analysis. Another thing is of course that R is "hot" at the moment. "Everyone" are using R in academia, governmental institutions, research institutes etc, AND it is free (quite important some times).

So let us start with making a complex calculation. What is $2+2$? Four you say? Sure? Let's see if R agrees with that. Place your mouse-pointer in the R-screen (or console as it is called by the pros). Click there, and you will see a cursor starting to blink by this sign ">". That is R waiting for your assignment (like a dog waiting for you throwing a stick).

Then write $2+2$. This will look a bit like this

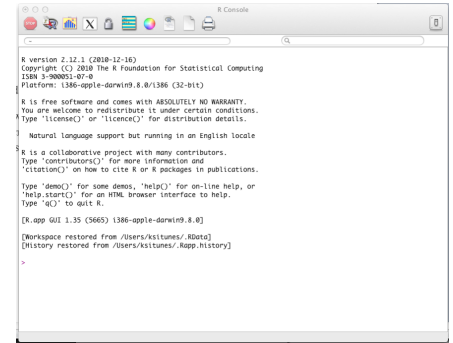
```
Type 'q()' to quit R.
[R.app GUI 1.35 (5665) i386-apple-darwin9.8.0]
[Workspace restored from /Users/ksitunes/.RData]
[History restored from /Users/ksitunes/.Rapp.history]
> 2+2
```

After you have pressed enter, your screen will look like this

```
[R.app GUI 1.35 (5665) i386-apple-darwin9.8.0]
[Workspace restored from /Users/ksitunes/.RData]
[History restored from /Users/ksitunes/.Rapp.history]
> 2+2
[1] 4
>
```

Now you see **[1] 4** which means that $2+2=4$. The squared bracket is a bit confusing, but it indicates the output line. Let us say that you make a calculation that give hundreds or even thousand of numbers as answer, then having R to

Fig 1 The opening screen as seen on a mac



count the lines is helpful. But for us, it is just a number in a bracket that we don't give a thought.

Arithmetic operators are +, -, /, *, %%%, and ^. By using the following syntax in the console you get help (be sure that you are connected to internet)

```
>?"+"
```

This notation is valid only for operators. To get help on other topics just write

```
>?mean
```

```
>?sd
```

etc.

Another method to get help is to write

```
>help(mean)
```

```
>help(sd)
```

```
>help("?")
```

EXCERSISE

Find out what the different arithmetic operators do and calculate

$2+3+5+6$

$2/3$

Find the integer quotient of $2/3$

What is the difference between

$2+2*4$

and

$(2+2)*4$.

Now you have seen a bit of the simple tasks you can solve with R. Not much you may think. It's easier with a calculator. Yes it is, but these simple tasks are not what R is meant for. You will now learn how to make calculations with a bit more complicated dataset. Our first task is to calculate the mean value for person A. This is done by adding all the measurements and divide by the number of measurements. In R this is simple. Let us say that the following measurements are done $H1=190$ cm, $H2=191$ cm, $H3=190$ cm, $H4=190$ cm, $H5=187$ cm and $H6 =192$ cm. What is the mean value. The first thing we have to do is to define the DATA belonging to person A. These are H1 to H6. We put them together in

Matrix1

The data: Make measurement of your fellow students. The instructor will "name" each student A, B, C...etc. you will finally get a matrix like this

A B C D

A -
B -
C -
D -

something we call a vector

```
>A<-c(190,191,190,190,187,192)
```

This may seem a bit odd, but what we have done is to join all the measurements of the **person A**'s height into a **unit** called A. Let us look at this statement in detail

A is the name of the vector that contains the data. The `<—` means that we assign the values that is in the parenthesis to the name A. This is actually not the same as the “=” sign, but for now you can think of it as the equal sign. Then you have the strange letter “c”. That is a short hand notation for “contains”. In plain language we can say the statement as follows:

A contains the values 190,191,190,190,187,192.

What happens if you write

```
>A
```

in the R-console. Try to add two A-values. What happens? Try to multiply A by 4, what happens?

To calculate the mean value of a, simply write

```
>mean(A)
```

We will now look into a new problem, and that is to construct a matrix. Say that one student(B) measures A to 190,192 and 193 cm and that another student (C) measures A to 189,190 and 190 cm.

$$A_{tot} = \begin{pmatrix} 190 & 192 & 193 \\ 189 & 190 & 190 \end{pmatrix}$$

We assign the measurements as follow

```
>AmB<-c(190,192,193)
```

```
>AmC<-c(189,190,190)
```

We do now want the measurements to be collected in a matrix like the one to the right

To construct this matrix we uses the command “matrix” in R.

```
> AmB<-c(190,192,193)
```

```
> AmC<-c(189,190,190)
```

```
> Atot<-matrix(c(AmB,AmC), nrow=2, byrow=T)
```

This code is straight forward, but it may seem a bit complicated. The first thing we do is to assign the measurements done by student B and C to different identifiers (AmB and AmC, m is a to remind you that A is Measured by B etc). The next step is to make a matrix of the measurements of A were the rows are the measurements done by the student B and C. To do this we use the word *matrix*. Next, the measurements we want in Atot are those made by B and C. This is done by making an vector containing the values by writing `c(AmB, AmC)`. Then we tell R to help us making two rows, one for each student (B and

C). We want the matrix to be sorted by its rows and not the columns, that's why we add the `byrow=T` (`T=True`, a boolean expression). What is the difference between `c(AmB, AmC)` and `Atot`? What if we set `byrow=F`

If we want to get the hold of row number 1, we can write

```
>AR1<-Atot[1,]
```

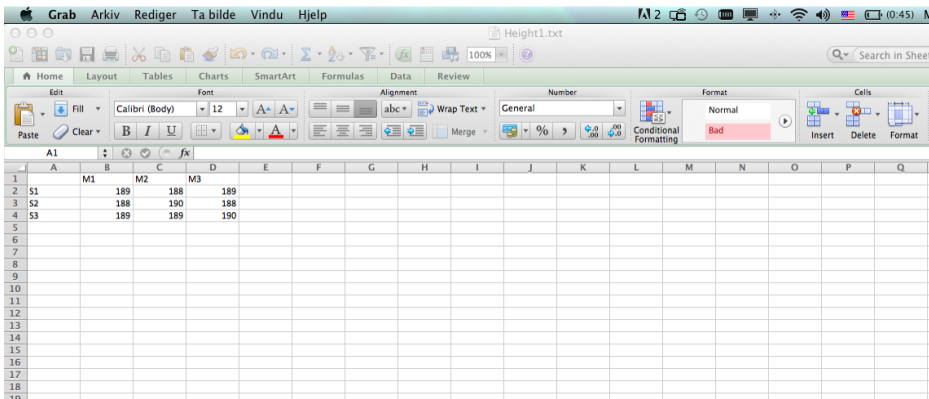
	M1	M2	M3
S1	189	188	189
S2	188	190	188
S3	189	189	190

To get column 1, write

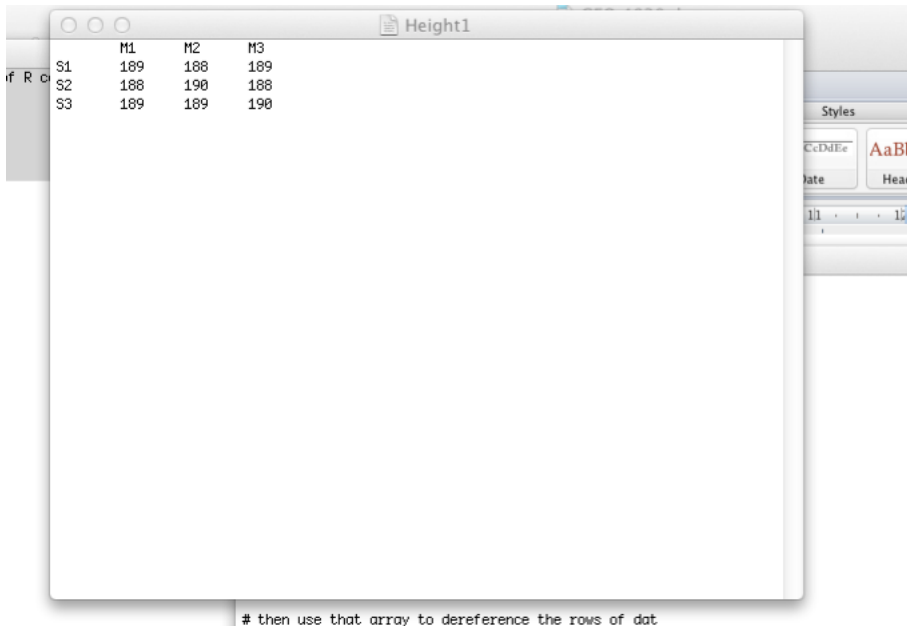
```
>AR1<-Atot[,1]
```

We will now move to the next step, loading data from a file. R is not EXCEL with respect to entering data by the keyboard. To load a dataset from a file, the best thing to do is to make the table in a spreadsheet like EXCEL, save the data as a tab delimited file (*.txt) and then load into R by the command `read.table()`

When you are calling this command, the least you have to provide to R is a file name, and that should be a file containing data acceptable to R. Further it would be wise to assign the table to a name. So, let us say that the class have measured the height of the instructor. Each student has a name and each student have measured the height three times. A table may seem like this. We have made this in EXCEL first, and it looks like this



We have then exported this file as an ascii-file, so when you open this file in a text editor like Notepad (Windows), TextEdit (Mac OSX) or Emacs (Linux), the file will look like this:



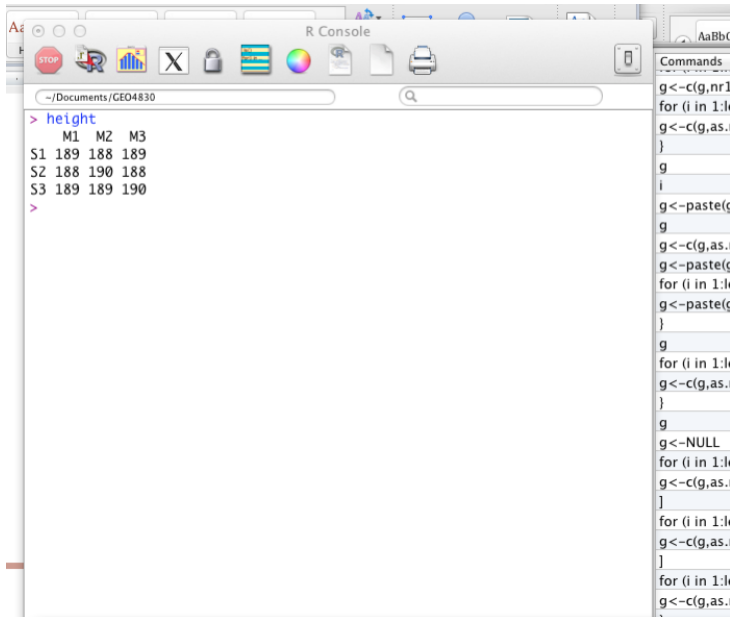
Finally, to get this file into R write the following statement (Windows)

```
>height<-read.table("C:\Dokumeter\Height1.txt", header=T)
```

Now, typing

```
>height
```

your console, you will get a screen like this



To get the first v value in the first cell you write

```
>cell1<-height[1,1]
```

To get the first row write

```
>cell_row_1<-height[1,]
```

To get the first column write

```
>cell_column_1<-height[,1]
```

Until the next week, make a group of four persons. All in the group measure the height according to Matrix 1. Then, make a text file with the data that you import into R. Calculate the mean value and the standard deviation for each person in the group.

One way to make plots is by using the plot() command. Use the help system and see if you can make a plot of the mean values of the participants in the group.

Save the plot by using the menu system. Edit the code in a text editor and save it. You can copy and paste text directly into R. This is shown in the lecture. If you were not at the first lecture, see example in powerpoint presentation on Fronter