Statistics for Data Science / Exercise 02

Pierpaolo Brutti

Just follow the step–by–step instructions and feel free to make mistakes. Remember that any time something wrong happens and you end up with an unwelcome `+` at the beginning of the command line, just press `esc` and everything will be ok. OK, time to start `RStudio`...

Part I: How to make a plot using `R`

Getting started

a. Always a good idea to start a new working session my creating a new project (`File` > `New Project` etc).

b. As you know, instead of typing weird commands directly at the prompt, use a `.R` script file: from `File` menu select `New File` and then `R Script`. Save it right away choosing any file name you like. For your convenience, use "sectioning" whenever you can to organize your script.

Manipulating Data

In `base R` you can find several example datasets. For the sake of this exercise we just pick one to play with. Try...

```
data()
```

Now you see the list of all available datasets.

To load the dataset `trees` in your current workspace type

```
data(trees)
```

Now check what's new in your workspace by looking at the tab `Environment` in the upper right panel of the `RStudio` interface, or simply use

```
?ls
ls()
```

A variable named `trees` should be there. Have a look in the simplest possible way...

```
trees
```

Sometimes, this is not a good idea, because you may ignore how *big your dataset is...so try with

```
# Show only the first 6 rows
head(trees)
# Show only the last 6 rows
tail(trees)
# Show the structure of the object
str(trees)
```

So, try to complete the following with the informations you get:

Number of observations: ..

Number of variables: ..

Type of variables: ..

Searching the R help file, can you find specific functions to answering *programmatically* (i.e. without eyeballing the output of `str()`) these questions?

You may already have an idea of what the data are about, but always remember you can ask R for

```
help(trees)
?trees
```

To get the names of the variables

```
colnames(trees)
```

What do you think is the result of this command? ..

Do you remember how to assign a name to this output? In other words, create an object, named `variables`, containing the result of the command above. ..

Now, type

```
variables[2]
```

and try to explain what the square bracket seems to do. ..

A step more: to extract a row (for example the 3rd tree) of the dataset you need

```
trees[3,]
```

What are the observed variable values for the 9th tree?

Girth: ..

Height: ..

Volume: ..

You already know, how to build a sequence using R. How would you select the dataset rows corresponding to the odd trees? ..

Finally, in order to extract the variables you can either use the square bracket (how? ..) or

```
trees$Girth
trees$Height
```

Summary Statistics

You can use the function `min()`, `max()`, `mean()` to complete the following:

Minimum Height: ..

Maximum Volume: ..

Average Girth: ..

Play also with `summary()`

Can you identify the tallest tree? ..

Select the trees with Volume > 30 ..

Create a new object containing the corresponding rows ..

Compute the average height of this subset of trees ..

Break!

Have you ever saved??? Sometimes it helps...

Save the script, and use `save()` or `save.image()` (what's the difference between these two? .....) to store your results.

Plots

Scatterplots

Lets start with some graphical representation of the data. To produce a scatterplot of the `Volume` just type

```
plot(trees$Volume)
```

Can you explain whats on the x-axis? .....

Change the y-axis label

```
plot(trees$Volume, ylab = "volume")
```

Now try with the x-axis .....

Set a title for your first graph.

```
plot(trees$Volume, ylab = "volume", main = "Trees volume")
```

We can draw a horizontal (option `h`) line to separate the trees we have selected in the previous point from the others

```
abline(h = 30)
```

Play with the following options:

```
abline(h = 30, col = 2, lty = 2)
```

What does the `col` argument do? .....

What does the `lty` argument do? .....

Now try to plot the volume of the trees as a function of their height. .....

You can add *something* on the previous plot using the function `points()`.

Try to overwrite on the same plot the points corresponding to the trees with volume larger than 30, using a different color (and/or a different type of point with the option '`pch = 3`).

Lets try something more *brutal*. Can you give your interpretation of what happens if you directly apply the function `plot()` to the whole dataset `tree`? .....

Histograms

Plot a histogram of each variable in the dataset using the function `hist()`. .....

Play with graphical options (`xlab`, `ylab`, `main`, `col`) .....

Check the help file of `hist()`: can you understand what is the option `breaks` for? Give an example. .....

Check the help file of `hist()`: can you understand why there's an option `plot` defaulted to `TRUE`? I mean, if you set it to `FALSE` and the function `hist()` does not produce a plot (try and check), what is its purpose then? (Hint: try to store its output in a variable, check its type, and notice the existence of a function named `?plot.histogram()`...) Explain. .....
