

R for EPP research Assignment 1

Angelos-Miltiadis Krypotos

2016-01-08

Name : [Put your name here]

General instructions.

This first assignment includes a lot of reading. My estimation is that you will need less than 8 hours for completing it – so with the 2 hours of the course that would add to 10 hours. If you want to do parts of the assignment, that is fine, although it is best to do everything. In case you are done with it and you want more exercises, please let me know and I can send you more – I just did not want to overwork anyone in the group as I know that you have many more other things to do.

How to complete the assignment.

Please write your answers below each question, indicating clearly to which question (or subquestion) your answer refers to. I would prefer if you would send your answers as an html document using “knitr” and with all the code having been already executed. You can create html documents in R studio by just clicking the notepad icon on the top – For more info how to create notebooks in R studio see here [<https://support.rstudio.com/hc/en-us/articles/200552276-Creating-Notebooks-from-R-Scripts>]. If you want, you can also send your answers as an .R script. Lastly, writing your assignments in .doc or .docx is also OK, but because of automatised formatting etc., it would be hard for me to check whether the code is running properly or not.

Exercise 1

Install R [<https://cran.r-project.org/>] and R studio [<https://www.rstudio.com/products/rstudio/download/>].

Exercise 2

Read the slideshows of Lectures 1 and 2. If anything is unclear, please mail me or drop by whenever you want.

Exercise 3

Read Google’s R style guide, available here <https://google.github.io/styleguide/Rguide.xml#object> You will need this for the next exercise.

Exercise 4

Based on Google’s R style guide, find any potential flaws in style on each line below and rewrite accordingly.

```
myCool.vector = c(1, 43, 3, 1, 5)
i_Like_Spiders <- c("Yes", "No")
x=1:50
if(exists("x")){
  something.stupid="something.stupid"
  print(something.stupid)
}
```

Exercise 5

For the assignments below we will do a lot of indexing, so you better fist check the R reference card here [\[https://cran.r-project.org/doc/contrib/Baggott-refcard-v2.pdf\]](https://cran.r-project.org/doc/contrib/Baggott-refcard-v2.pdf).

Exercise 6

Create a numeric vector that holds a sequence of numbers from -100 to 100, with steps of 10.

Exercise 7

This exercise is seperated into multiple subexercises. The idea here is to show you how you could create a randomization list, with equal number of participants in each group, and each group having and equal number of males and females.

- a) create a matrix with 60 rows and 2 columns. The row names should have the pattern “IDX”, with X standing for the row number. So, the first row should have the name, ID1, the second row ID2, etc. The columns should have the following names: “Sex”, and “group”. The column sex gets two values “M”, “F”. Put those values into the matrix so that every odd line has the value “M” on the Sex column, and every even line the value “F” on the Sex column. The column group gets two values: “EMDR”, and “Ext”. Put those values into the matrix so that there is an equal number of males and females for any of the two groups.
- b) Check the number of males and females.
- c) Check whether there are equal number of males and females in each group.
- d) Print the first 4 lines of the matrix.
- e) Print the last 4 lines of the matrix.
- f) Randomize the rows and save the new matrix into a new object.
- g) Print the first 4 lines of the matrix.
- h) Print the last 4 lines of the matrix.
- i) Add a new column to the matrix named “PartN”, that holds a sequence of numbers from 1 to 60 with steps of 1.

Exercise 8

The idea behind this exercise is to show you how you can index a data frame and create a new columns based on scores on a table. For example, you will be asked to score people with low ASI scores as 1 and with hight ASI scores as 0. This could be useful when, for example, you would want to see the mean ASI per group etc. Run the code below

```
x <- matrix(c(134, 414, 211, 5523, 12, 43, 12, 55, 11, 22), nrow = 5, ncol = 2)
colnames(x) <- c("RT", "ASI")
rownames(x) <- paste0("ID", 1:5)
```

- a) Convert the matrix into a data frame
- b) Return the RT scores for all participants that have an ASI score below 30.
- c) Create a new column named “LASI”, where you are going to code the participants you selected on question 7b as 1 and then rest as 0.
- d) Convert the column “LASI” into a factor. What are the levels of that factor?
- e) Install the package dplyr.
- f) Load the package.
- g) Check the function filter in the dplyr package.
- h) Repeat what you did on question 6b, but this time by using the function filter.
- i) Check whether the results are equivalent (you need to use a boolean statement for that).

Exercise 9

Just a simple exercise in which we manipulate lists. Run the code below

```
my.List <- list(CS = c("+", "-", "+"), Part = c("1", "2", "3"))
```

- a) Select the Part object and compute its mean.
- b) Add a new list element naming it “RT” that includes 3 random numbers drawn from a normal distribution (see ?rnorm).
- c) Add a new list element naming it “RN” that includes 3 unique random numbers from the sequence 100:200.
- d) Convert my.List to a data frame called “list.2.frame”
- e) Create a new column on the list.2.frame list, naming it “RSum” that includes the row sums of columns RT and RN
- f) [This last one is just for getting you excited]. Run this

```
plot(list.2.frame$Rsum)
```