R for EPP research Assignment 2 – Key

Angelos-Miltiadis Krypotos

2016-02-11

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. 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.

Goals of the assignment

- Gain experience with exporting and importing data
- Gain experience with loops
- Gain experience with working with different directories.

Note. I did not include anything regarding if else statements. This was intentional as I think you already have enough on your plate. We will, however, have the chance to work with if else statements later on, especially when we talk about creating custom functions.

Exercise 1

Read the tutorial on loops from DataCamp link, until the title 'The apply family: just hidden loops?'.

Answer 1

Done.

Exercise 2

The official manual for importing and exporting data in R is here. However, it is long and complicated. So, make sure that you read the slides of lecture 3, and you can also check the links here and here.

Answer 2

Done.

Exercise 3

One of the reasons that make R so cool, is that it allows you to easily read your experimental data, reduce them to the variables of interest, and then perform the analyses you want (all in a few lines of code!). Goal of this exercise is to show you how you can read your experimental data in, perform some analyses, and then end up with the dependent variables of interest. In principle, you could use your own data – actually, if you want to do so, I am absolutely fine; just make sure that you do some things in line with what it is asked in the assignment. If you do not want to use your data, then you can just go on and generate some data from R, then read them in, and do whatever it is asked from you. The goals of this exercise are: 1) to make sure that you can export and import data, 2) use loops. When you use loops, you can decide what type of looping you want to choose (i.e., for, while, repeat). If you prefer to use vectorization than looping that is also fine – although I would prefer if you use loops as you should be able to use them.

So, let's begin:

a) Create a folder on your desktop and name it 'fakedata'

b) Change your working directory to the 'fakedata' directory. Note: Make sure that when you provide the path, that every subfolder is seperated with either the "\\" or the "/" characters. Do NOT use the "" character.

c) Create an empty matrix with 1000 rows and 3 columns. The column names should be "Trial", "CS", "RT". You do not need to name the rows.

d) Time to fill in the matrix. The first column should get the numbers from 1 to 1000. For the second column, fill in the numbers in the sequence from 1 to 5, in a random order. The third column should get numbers sampled from a random distribution (?rnorm) with a mean of 600 and a SD of 50.

e) Repeat exercises 3c and 3d 100 times. Every time, save the matrix into a txt file that is saved in your working directory. The txt files that you save should get the names, "fakepart_X", with X having the numbers from 1 to 100. Note: Make sure that your data are saved without quotes, and that the row names are not saved as well.

f) Now, read in all the data and saved them into a list, with every list item referring to the seperate file. Make sure that the list item gets the name of the txt file that is referring to (to get the files names of a directory, check ?list.files).

g) Create a new empty matrix, named "fake.final" with 100 rows and 3 columns. The column names should be "CS1", "CS2", "CS3", "CS4", "CS5".

h) Fill in each row with the median RTs of each stimulus, from each matrix. So, the first row should refer to the first matrix. The first column of the 1st row should include the medians for CS1 from that matrix, the second column of the 1st row should include the medias for CS2 from that matrix etc.

i) Save the fake.final file as a csv file in the fakedata directory.

Answer 7

a)

Done

b)

The exact directory depends on the path you are working on but here is an example:

```
setwd("//soliscom.uu.nl//uu//Users//Krypo001//My Documents//Desktop//fakedata")
```

c)

```
my.matrix <- matrix(-999, nrow = 1000, ncol = 3)
colnames(my.matrix) <- c("Trial", "CS", "RT")
```

d)

```
my.matrix[, 1] <- 1:1000
my.matrix[, 2] <- sample(rep(1:5, nrow(my.matrix)/5), nrow(my.matrix))
my.matrix[, 3] <- rnorm(nrow(my.matrix), mean = 600, sd = 50)
```

e)

```
for (i in 1:100){
  my.matrix <- matrix(-999, nrow = 1000, ncol = 3)
  colnames(my.matrix) <- c("Trial", "CS", "RT")
  my.matrix[, 1] <- 1:1000
  my.matrix[, 2] <- sample(rep(1:5, nrow(my.matrix)/5), nrow(my.matrix))
  my.matrix[, 3] <- rnorm(nrow(my.matrix), mean = 600, sd = 50)
  write.table(my.matrix, paste0("fakepart_", i, ".txt"), row.names = FALSE, quote = FALSE)
}
```

f)

This is one of the ways to read data. It is not the best way!! The idea is to show you a way that sometimes you need so much more code to do stuff with explicit looping. In the next class, we will see how much easier it is to do the same thing in a single line of codign – with implicit looping.

```
my.data.list <- list()
for (i in 1:100){
  my.data.list[[i]] <- read.table(paste0("fakepart_", i, ".txt"), header = TRUE)
  names(my.data.list)[i] <- paste0("fakepart_", i) ## We do not want the .txt file extensio
}
```

g)

```
fake.final <- matrix(-999, nrow = 100, ncol = 5)
colnames(fake.final) <- c("CS1", "CS2", "CS3", "CS4", "CS5")
```

h)

This is not the best the way to go – it is slow and lengthy. I am showing it thought because in the next lecture, we will see how we can achieve exactly what we do here using implicit looping!

```
for (i in 1:length(names(my.data.list))){
  # Just rename it in order to make indexing easier
  tmp = my.data.list[[i]]
  # Now we make new objects that can be filled in the matrix
  cs1 = median(tmp[tmp$CS == 1, ]$RT)
  cs2 = median(tmp[tmp$CS == 2, ]$RT)
  cs3 = median(tmp[tmp$CS == 3, ]$RT)
  cs4 = median(tmp[tmp$CS == 4, ]$RT)
  cs5 = median(tmp[tmp$CS == 5, ]$RT)
  # Fill the matrix
  fake.final[i, ] = c(cs1, cs2, cs3, cs4, cs5)
}
```

i)

```
write.csv(fake.final, "fake.final.csv", row.names = FALSE, quote = FALSE)
```