Second_Excercises

Joe

2018-07-29

Objectives

1. Understand the value of defining your own functions
2. Understand the rationale behind the Don’t Repeat Yourself (DRY) Philosophy
3. Be able to implement your own function to make a simulation easier
4. Be able to implement apply

Objectives 1 and 2

This part of the lesson will be inductive. Open WhyFunctions.R, and look at the scripts provided by ‘Serious Student.’ List what is wrong with both scripts. Do not move no until you feel you have completed objectives 1.

Open WhyApply.R and notice how painful it is to use method 1. See ?apply and describe what it does in your own words.

Objective 3

As can be seen in the bottom of WhyFunctions.R, defining your own functions help deal with the problems mentioned in 1 and 2. You can define your own functions by:

a. Beginning with a function name (e.g., ‘MyFunction =’)
b. type ‘function’
c. type inputs of function in parentheses (e.g., ‘(Data)’). Note that functions will ignore everything in your variable workspace. They only have access to inputs!
d. Define the function using curly brackets (e.g., ’{NewData=Data*10’)
e. Specify the output of the function (e.g., (‘return(NewData)’)
f. End function definition (e.g., ‘}’)

For example,

```
MyFunction=function(Data){
  NewData=Data*10
  return(NewData)
}
MyFunction(5)
```

```
## [1] 50
```

Test your understanding with these two exercises.

1. Consider this thought experiment from Tvsersky & Kahneman (1971).

‘Imagine a psychologist who studies the correlation between need for Achievement and grades. When deciding on sample size, he may reason as follows: “What correlation do I expect? r ~ .35. What N do I need to make the result significant? (Looks at table.) N = 33. Fine, that’s my sample.”’ (p.107)

Complete the code below in order to test whether 33 participants is a large enough sample for this psychologist. The code below is adapted from a tutorial on https://www.r-bloggers.com/easily-generate-correlated-variables-from-any-distribution-without-copulas/ (https://www.r-bloggers.com/easily-generate-correlated-variables-from-any-distribution-without-copulas/). Read it for details.

```
#load necessary package
library(MASS)

# Script for generating random data

#specifies population mean, variance, and covariance
mu <- rep(0,2)
Sigma <- matrix(.35, nrow=2, ncol=2) + diag(2)*.65

#randomly draws data with prespecified correlation
rawvars <- mvrnorm(n=33, mu=mu, Sigma=Sigma)

#TODO: perform a correlation test with something like cor.test(x,y)$p.value

#TODO: Create a function that runs this code 1 time

#TODO: use replicate to run this function 1000 times
```

2. What would be an appropriate sample size?

Exercise: Apply apply()

1. Reviewer X insists that you must apply the following transformation to your data before your paper is accepted. REVIEWER X DOES NOT NEED YOU TO UNDERSTAND STEP 1-3 OF THIS CODE. You only need to turn this code (which essentially grabs only the first and last digits in a number) into a wacky function called BizzaroTransform(). BizzaroTransform takes the argument MyNumber as input

```
#number input
MyNumber=152

#Step 1: convert number to string
MyString=as.character(MyNumber)

#Step 2: extract first and last character
MyStringShort=paste0(substr(MyString,start=1,stop=1), substr(MyString,start=nchar(MyStri
ng),stop=nchar(MyString)))

#step 3: convert string back to a number
MyFinalNumber=as.numeric(MyStringShort)

#step 4: Display number
MyFinalNumber
```

```
## [1] 12
```

2. Apply BizzaroTransform() to each row of this data frame.

```
Data=data.frame(rnorm(n=100,mean=100,sd=1))
```