Assignment 1.1

2022 Programming in Psychological Science

Contents

Q1.1.1 . . . 2
Q1.1.2 . . . 2
Q1.1.3 . . . 2
Q1.1.4 . . . 2
Q1.1.5 . . . 2
Q1.1.6 . . . 2
Q1.1.7 . . . 3
Q1.1.8 . . . 3
Q1.1.9 . . . 3
Q1.1.10 . . . 3
Q1.1.11 . . . 3
Q1.1.12 . . . 3
Q1.1.13 . . . 3
Q1.1.14 . . . 3
Q1.1.15 . . . 4
Q1.1.16 . . . 4
Q1.1.17 . . . 4
Q1.1.18 . . . 4
Q1.1.19 . . . 4
Q1.1.20 . . . 4
Q1.1.21 . . . 4
Q1.1.22 . . . 5
Q1.1.23 . . . 5
Q1.1.24 . . . 5
Q1.1.25 . . . 5
Q1.1.26 . . . 6
Q1.1.27 . . . 6
Q1.1.28 . . . 6
Q1.1.29 . . . 6
Q1.1.30 . . . 7
Q1.1.31 . . . 7
Q1.1.32 . . . 7

Assignment description

Work through the following 32 R problems. This assignment is worth 8 points (.25 points per problem). You should also complete Assignment 1.2P OR Assignment 1.2R for an additional 2 points. The total of Assignments 1.1 and your choice of either 1.2P (Python) or 1.2R (R challenge) are worth 10 points and 15% of your final course grade. You must work individually, but feel free to ask questions in class and on Slack!

You must submit the problems as a R script (.R file). All answers must have comments and be labeled with the problem number. Some answers will only be comments without associated R code. For questions about

output, report your output in a comment under the relevant line of code. Note that all comments have # in front of them so that R does not run this line of code.

```
#Q1.1.0
#This is the answer the example problem Q1.1.0 in a comment
ThisVariable <- 1
ThisVariable
#[1] 1

#Q1.1.1
#Answer
```

Problems

Q1.1.1

Open RStudio or another interactive R environment.

Type `getwd()` in the Console Window. Then type `?getwd` in the Console Window.

When Michael types type "getwd()" in his R Console, Michael sees:

```
[1] "C:/Users/Michael/Desktop/memes"
```

Where is Michael's current working directory? Reminder: Write your answer as a comment in your R script.

Q1.1.2

Michael wants to load a data file into R from his "PIPS" folder that is on his Desktop. But he knows that he must set his working directory to his "PIPS" folder first. What is the R command should he should enter in the R Console?

(Tip: you can also use the menu in R-studio. However this question is asking you about the specific R code!)

Q1.1.3

You want to know what the "rbinom" function does in R. What command can you run in the R Console to know what it does?

Q1.1.4

Create a variable called `NumericVar` and assign it the number `2022.2`. Then create a variable called `StringVar` and assign it the string `'This is my string'`. Can you add the two variables? Why or why not? Report code with output in a comment.

Q1.1.5

```
a <- b <- ab <- ba <- c <- 1
```

What was done is this code? How many additional values are in your workspace (e.g. "Environment") after running this code?

Q1.1.6

Create a string variable titled `FamousCode` with the following sentence: "Hello world." Return the same string by writing the variable name without an assignment. Also try the print() function.

Q1.1.7

Create a vector of numbers 2, FALSE, -1.243, and the string 'test' in that order. See help of the `c()` function if needed.

Q1.1.8

Read the help function of the function `rm()`. Remove all objects from your workspace with R code. Note that you can also use RStudio to do this.

Q1.1.9

```
SampleScores <- c(1,6,7,8,9,NA)
mean(SampleScores)
```

This gives NA, why? Can you compute the mean with the missing value removed? Check `?mean`.

Q1.1.10

Type `?mvrnorm` into the R Console. I got an error saying "No documentation for 'mvrnorm' in specified packages and libraries." But according to THIS WEBSITE this function exists in the MASS package. What two commands do I need to run to install and then load the MASS package?

Q1.1.11

```
apples <- 5; oranges <- 3; pineapples <- 6 # Try:
apples + pineapples
apples + Oranges
```

Why can't we add apples and oranges?

Q1.1.12

See `help(mode)`. Note that "double" is a specific type of numeric variable.

```
a1 <- "1"; a2 <- 1; a3 <- TRUE
b1 <- c(a1,a2); b2 <- c(a1,a3); b3 <- c(a2,a3); b4 <- c(a1,a2,a3)
```

What are the modes/types of a1, a2, and a3? What are the modes/types of b1, b2, b3, and b4? Explain how R converts values in vectors.

Q1.1.13

```
TheseNumbers <- seq(-10, 0)
```

I want to return the third number to the sixth number of TheseNumbers using R indexing. Write code to do this.

Q1.1.14

Look at `?letters`. `letters` is a special built in variable in R.

Return the `length` of `letters` with R code.

Q1.1.15

```
EqualIt = letters
EqualIt <- letters
letters -> EqualIt
EqualIt == letters
```

What is the difference between the four lines?

Q1.1.16

```
TestIt <- 'wow'
TestIt[10] <- 'doge'
```

What are the values of `TestIt[2]` to `TestIt[9]`? Why does this occur?

Q1.1.17

Write R code to compute the following. Given a general triangle with sides a=4, b=5, and the angle alpha=pi/3 between a and b, write R code to compute the length of the third side c. Hint: Use the LAW OF COSINES

Q1.1.18

What is the product of the first 100 odd positive integers `(1,3,5,...)`, e.g all odd positive integers below 200? Find a solution that does not involve typing out all the numbers.

Q1.1.19

Run this entire code before answering the question:

```
set.seed(1234)
NRows <- sample(2:200,1) #Find a random number of rows
NCols <- sample(2:200,1) #Find a random number of columns
MyArray <- matrix(data=seq(1,NRows*NCols), nrow = NRows, ncol=NCols) #Make a matrix (array)
```

Return the dimension of the matrix using only a R function on "MyArray"

Q1.1.20

Set the random seed to your student ID. Then immediately run the following lines and return the output. Run these lines, including the random seed line, multiple times. Why do you get the same output?

```
MySpecialNum <- sample(1:10^9,1) #Find a random number from 1 to 10^(9) = 1000000000
MySpecialNum
```

Now run only these lines multiple times without the random seed line. Why do you get different output?

```
NotSpecial <- sample(1:10^9,1) #Find a random number from 1 to 10^(9) = 1000000000
NotSpecial
```

Q1.1.21

Type:

```
set.seed(1234) #Set the random seed
NTimepoints <- 1325
NElectrodes <- 201
EEGData1 <- matrix(runif(NTimepoints*NElectrodes), ncol=NElectrodes, nrow=NTimepoints)
EEGData2 <- matrix(runif(NTimepoints*NElectrodes), NTimepoints, NElectrodes)
EEGData3 <- matrix(runif(NTimepoints*NElectrodes), NElectrodes, NTimepoints)
EEGData4 <- matrix(ncol=NElectrodes, nrow=NTimepoints, data=runif(NTimepoints*NElectrodes))
```

See the help documentation for `matrix`. We want the 1325 time points in the rows and 201 electrodes in columns. Which of the last four lines is incorrect? What does this tell you about function calls in R?

Q1.1.22

Type:

```
set.seed(1234) #Set the random seed
NRows <- 20 -> NCols  #This is not a recommended way of writing R code, but fun
M <- matrix(runif(NRows*NCols),NRows,NCols)
```

`mean(M)` returns one number. How can we get the mean of each column? Hint: The answer can be found in the help function for `mean`, but not where you might expect it.

Use base R and not `dplyr`. If you don't know what `dplyr` is, you will learn about `dplyr` in this course!

Q1.1.23

Make this array (`matrix`) of strings without typing out each element of the array.

```
matrix(c('string1','string2'), 4, 4)
```

```
##      [,1]      [,2]      [,3]      [,4]
## [1,] "string1" "string1" "string1" "string1"
## [2,] "string2" "string2" "string2" "string2"
## [3,] "string1" "string1" "string1" "string1"
## [4,] "string2" "string2" "string2" "string2"
```

Q1.1.24

```
set.seed(1234) #Set the random seed
NBrainFuncs <- 8
BetaSims <- rbeta(NBrainFuncs^2, shape1=2, shape2=10)
InteractionMatrix <- matrix(BetaSims, NBrainFuncs, NBrainFuncs)
```

How can we make the above code only 3 lines? Reduce the last two lines of R code to only one line of R code

Q1.1.25

```
set.seed(1234) #Set the random seed
NBrainFuncs <- 8
BetaSims <- rbeta(NBrainFuncs^2, shape1=2, shape2=10)
InteractionMatrix <- matrix(BetaSims, NBrainFuncs, NBrainFuncs)
```

My "InteractionMatrix" gives the influence of brain function j on brain function i. For instance the 3rd row, 8th column element gives the influence of the 8th brain function on the development of the 3rd brain function. How do I index the InteractionMatrix to return the influence of the 2nd brain function on the 7th?

Q1.1.26

```
set.seed(1234) #Set the random seed
NBrainFuncs <- 8
BetaSims <- rbeta(NBrainFuncs^2, shape1=2, shape2=10)
InteractionMatrix <- matrix(BetaSims, NBrainFuncs, NBrainFuncs)
DiagIndex <- diag(NBrainFuncs)
InteractionMatrix[DiagIndex] <- 0  #Set this index to zero in the InteractionMatrix
```

I want to set the diagonal of the interaction matrix to zero. Fix the above code to do this. Remember that logicals can be indices.

Q1.1.27

See the help function for `rep()`.

I want to make the vector with a length of 8 in the following order: "This", "This", "is", "is", "a", "a", "test", "test" with the `rep` function but `rep(letters[1:5],2)` does not work. What do I have to change in the `rep` function?

```
ThisIsATest <- c("This","is", "a", "test")
rep(ThisIsATest,2)
```

Q1.1.28

We used this data to simulate the speed of R and Python on various operations.

```
set.seed(1234) #Set the random seed
SpeedSec <- rep(0,10)
SimSpeed <- runif(5)  #Speed simulation in seconds
SpeedSec[seq(1,10,by=2)] <- SimSpeed*10
SpeedSec[seq(0,10,by=2)] <- SimSpeed
```

Make the following `data.frame` without typing out the first two columns. You can use the variable `SpeedSec` for the last column.

```
##    Language CodeType  SpeedSec
## 1         R forloop1 1.1370341
## 2    Python forloop1 0.1137034
## 3         R forloop2 6.2229940
## 4    Python forloop2 0.6222994
## 5         R forloop3 6.0927473
## 6    Python forloop3 0.6092747
## 7         R forloop4 6.2337944
## 8    Python forloop4 0.6233794
## 9         R forloop5 8.6091538
## 10   Python forloop5 0.8609154
```

Q1.1.29

Type:

```
set.seed(1234)
NStudents <- 50
Grades <- data.frame(1:NStudents,matrix(sample(8:20,NStudents*2,TRUE)/2,,2))
names(Grades) <- c('student','grade1','grade2')
```

Students pass the course when the average grade (i.e., the average of 'gradeA' and 'gradeB') is at least 5.5 and both 'gradeA' and 'gradeB' are larger than 5. How would you find a vector of only the students who passed the course?

Q1.1.30

Type:

```
FruitVec <- c("bananas","apples")
```

Change all a's in `FruitVec` to dots (so, the bananas becomes b.n.n.s) Hint: `?gsub`.

Q1.1.31

Make a four dimensional `array` of 2 by 2 by 2 by 2 with as elements the squares of 1 to 16 (e.g., 1^2, 2^2, 3^2, 4^2 etc).

Q1.1.32

In R one can use already available data sets by simply typing their name. Look at the `InsectSprays` data set and export it as a .csv file. How do you make sure that row names are not included?