

Intro to Data Science - Lab 1

Copyright 2022, Jeffrey Stanton and Jeffrey Saltz Please do not post online.

Week 1 - Introduction to R

```
# Enter your name here: HENDI KUSHTA
```

Please include nice comments.

Attribution statement: (choose only one and delete the rest)

```
# 1. I did this lab assignment by myself, with help from the book and the professor.
```

Each student should run R-Studio on their computer (or via <https://rstudio.cloud/>). Remember that R should always be installed before R-Studio (if you are not using rstudio.cloud or Google Colab)

1. Add together all the numbers between 1 and 10 (inclusive). Take note of the result. Remember, every student should type and run the code on their machine.

```
sum(1:10)
```

```
## [1] 55
```

2. Now create a vector of data that contains the numbers between 1 and 10 (inclusive). Here is a line of code to do that:

```
myNumbers <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

```
myNumbers <- c(1:10)
```

3. Now add together all of the numbers that are in the vector **myNumbers**. There is a built-in function within R that can do this for you in one step: Take a guess as to the name of that function and run it on **myNumbers**. Check your result against the results of question 1.

```
sum(myNumbers)
```

```
## [1] 55
```

4. R can do a powerful operation called **** vector math **** in which a calculation runs on every element of a vector. Try vector math on **myNumbers** by adding 10 to each element of myNumbers, and storing the result in **myNewNumbers**. Print out **myNewNumbers**.

```
myNewNumbers <- myNumbers + 10
```

```
myNewNumbers
```

```
## [1] 11 12 13 14 15 16 17 18 19 20
```

5. Efficiently calculate a sum of the numbers between 11 and 20 (inclusive), using techniques from the problems above. **Hint:** use `c(11:20)`

```
sum(c(11:20))
```

```
## [1] 155
```

6. Calculate a sum of all of the numbers between 1 and 100 (inclusive), using techniques from the problems above.

```
sum(1:100)
```

```
## [1] 5050
```

7. Make sure you have a variable **myNumbers**, that is a vector of 10 numbers (1,2,3,4,5,6,7,8,9,10)

```
myNumbers
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

8. Type the following commands in the code cells below and run each one:

```
mean(myNumbers) median(myNumbers) max(myNumbers) min(myNumbers) length(myNumbers)
```

```
mean(myNumbers)
```

```
## [1] 5.5
```

```
median(myNumbers)
```

```
## [1] 5.5
```

```
max(myNumbers)
```

```
## [1] 10
```

```
min(myNumbers)
```

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

```
length(myNumbers)
```

```
## [1] 10
```

9. Repeat the commands from above, this time adding a comment to each line of code in your file explaining what it does. The comment character is # .

```
mean(myNumbers)
```

```
## [1] 5.5
```

```
# Finds the average of myNumbers vector, wich in this case is 5.5
```

```
median(myNumbers)
```

```
## [1] 5.5
```

```
# Finds the median f myNumbers vector.
```

```
# There are even numbers of values in the vector. (5+6)/2 in this case is 5.5.
```

```
max(myNumbers)
```

```
## [1] 10
```

```
# Finds the maximum value in myNumbers vector wich is 10.
```

```
min(myNumbers)
```

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

```
# # Finds the minimum value in myNumbers vector wich is 1.
```

```
length(myNumbers)
```

```
## [1] 10
```

```
# There are 10 numbers in myNumbers vector.
```

10. In a comment, explain the output of the following command: `myNumbers > 5`

```
# It will produce a boolean results of FALSEs and TRUEs for each value in the vector  
# myNumbers.  
# If the value is greater than 5, it will give TRUE, else it will give FALSE.  
# From 1-5 will give FALSE and from 6-10 it will give TRUE.
```

11. Explain what in is **bigNum** after executing the following command:

```
bigNum <- myNumbers[myNumbers > 5]  
bigNum <- myNumbers[myNumbers > 5]  
# bigNum will contain a subset of myNumbers vector based on the conditional expression  
# myNumbers > 5.  
# It will contain 6, 7, 8, 9, 10.
```

12. Whenever you need R to explain what a command does and how it works, use the ? command or the **help()** command. Add and run these commands: `?mean` `help("mean")`

```
?mean
```

```
help(mean)
```