

Computing for Data Science

FALL SEMESTER 2017

INSTRUCTOR: Mr. Atul Nag
atulnag@curaj.ac.in

R Basics Exercise

Test you on a few topics:

- Basic Data Types
- Basic Arithmetic
- Vector
- Vector Operations
- Comparison operators
- Vector Selection and Indexing

Solve the following problems using R

- 1) What is two to the power of five? [32]
- 2) Create a vector called stock.prices with the following data points: 23,27,23,21,34 and print it out.
- 3) Assign names to the price data points relating to the day of the week, starting with Mon, Tue, Wed, etc...
- 4) What was the average (mean) stock price for the week? [25.6]
- 5) Create a vector called over.23 consisting of logicals that correspond to the days where the stock price was more than 23
- 6) Use the over.23 vector to filter out the stock.prices vector and only return the day and prices where the price was over 23
- 7) Use a built-in function to find the day the price was the highest

R Matrix Exercises

Through these exercises we will review the matrix data structure and perhaps introduce you to a few ideas for you to discover on your own!

- 1) Create 2 vectors A and B, where A is (1,2,3) and B is (4,5,6). With these vectors, use the `cbind()` or `rbind()` function to create a 2 by 3 matrix from the vectors. You'll need to figure out which of these binding functions is the correct choice.
- 2) Create a 3 by 3 matrix consisting of the numbers 1-9. Create this matrix using the shortcut `1:9` and by specifying the `nrow` argument in the `matrix()` function call. Assign this matrix to the variable `mat`
- 3) Confirm that `mat` is a matrix using `is.matrix()`
- 4) Create a 5 by 5 matrix consisting of the numbers 1-25 and assign it to the variable `mat2`. The top row should be the numbers 1-5.
- 5) Using indexing notation, grab a sub-section of `mat2` from the previous exercise that looks like this:
[7,8]
[12,13]
- 6) Using indexing notation, grab a sub-section of `mat2` from the previous exercise that looks like this:
[19,20]
[24,25]
- 7) What is the sum of all the elements in `mat2`?
- 8) Find out how to use `runif()` to create a 4 by 5 matrix consisting of 20 random numbers ($4 \times 5 = 20$).

R Data frame exercises

- 1) Recreate the following dataframe by creating vectors and using the `data.frame` function:

	Age	Weight	Sex
Sam	22	150	M
Frank	25	165	M
Amy	26	120	F

- 2) Check if `mtcars` is a dataframe using `is.data.frame()`
- 3) Use `as.data.frame()` to convert a matrix into a dataframe:

```
mat <- matrix(1:25,nrow = 5)
```
- 4) Set the built-in data frame `mtcars` as a variable `df`. We'll use this `df` variable for the rest of the exercises.
- 5) Display the first 6 rows of `df`
- 6) What is the average mpg value for all the cars?
- 7) Select the rows where all cars have 6 cylinders (`cyl` column)
- 8) Select the columns `am`, `gear`, and `carb`.
- 9) Create a new column called `performance`, which is calculated by `hp/wt`.
- 10) Your `performance` column will have several decimal place precision. Figure out how to use `round()` (check `help(round)`) to reduce this accuracy to only 2 decimal places.
- 11) What is the average mpg for cars that have more than 100 hp AND a `wt` value of more than 2.5.
- 12) What is the mpg of the Hornet Sportabout?

Conditional Statements Exercises

- 1) Write a script that prints "Hello" if the variable x is equal to 1:
- 2) Write a script that will print "Even Number" if the variable x is an even number, otherwise print "Not Even":
- 3) Write a script that will print 'Is a Matrix' if the variable x is a matrix, otherwise print "Not a Matrix". Hint: You may want to check out `help(is.matrix)`
- 4) Create a script that given a numeric vector x with a length 3, will print out the elements in order from high to low. You must use if,else if, and else statements for your logic. (This code will be relatively long)
- 5) Write a script that uses if,else if, and else statements to print the max element in a numeric vector with 3 elements.

R Functions Exercise

For these exercises you be given instructions to create functions that take in certain inputs and give certain outputs.

- 1) Create a function that takes in a name as a string argument, and prints out "Hello name"
- 2) Create a function that takes in a name as a string argument and *returns* a string of the form - "Hello name"
- 3) Create a function that will return the product of two integers.
- 4) Create a function that accepts two arguments, an integer and a vector of integers. It returns TRUE if the integer is present in the vector, otherwise it returns FALSE. Make sure you pay careful attention to your placement of the return(FALSE) line in your function!
- 5) Create a function that accepts two arguments, an integer and a vector of integers. It returns the count of the number of occurrences of the integer in the input vector.
- 6) We want to ship bars of aluminum. We will create a function that accepts an integer representing the requested kilograms of aluminum for the package to be shipped. To fulfill these order, we have small bars (1 kilogram each) and big bars (5 kilograms each). Return the least number of bars needed.

For example, a load of 6 kg requires a minimum of two bars (1 5kg bars and 1 1kg bars). A load of 17 kg requires a minimum of 5 bars (3 5kg bars and 2 1kg bars).
- 7) Create a function that accepts 3 integer values and returns their sum. However, if an integer value is evenly divisible by 3, then it does not count towards the sum. Return zero if all numbers are evenly divisible by 3. Hint: You may want to use the append() function.
- 8) Create a function that will return TRUE if an input integer is prime. Otherwise, return FALSE. You may want to look into the any() function. [Primality Test](#)