CS 251 Statistical Computing

HOP1: R for statistical project

2/19/2020 Developed by Aya Khalil

3/30/2020 Reviewed by Apiwat Chuaphan

School of Technology & Computing @City University of Seattle (CityU)

**Before You Start**

* If you already finished this module through any CityU Technology Institute (TI) courses,  
  just skim this module and skip it.
* Version numbers may not match with the guide. But that should be fine.  
  If given the option to choose between stable release (long-term support) or most recent, please choose the stable release.
* This guide targets Windows OS users. So, MacOS users may have different commands to input in the shell/terminal.
* We cannot explain every step. **This cookbook always needs your own creative judgement.**
* **For your working directory, use your course number.** The hands-on tutorial may use a different course number as an example.

**Learning Outcomes**

* Introduction to R programming language
* Arithmetic operations
* Variables
* Data types
* Vectors
* Lists

**Resource**

* Introduction to R: <https://www.r-project.org/about.html>
* Hui, E. G. M. (2019). [*Learn R for applied statistics: With data visualization, regressions, and statistics*](https://login.proxy.cityu.edu/sso/skillport?context=144516). Apress.

**What is R?**

R programming is for statistical computing and is supported by the R Foundation for Statistical

Computing. The R programming language is used by academics and researchers for data analysis and statistical analysis, and R programming's popularity has risen over time. As of June 2018, R is ranked 10th in the TIOBE index.

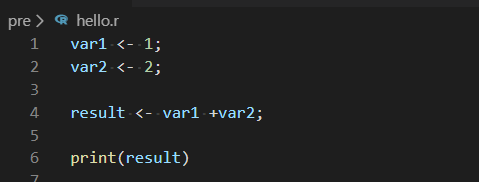
Along with Python, R is used widely in the field of data science, which consists of statistics, machine learning, and domain expertise or knowledge.

Is R high-level language or low-level language?

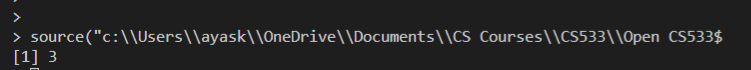
A high-level programming language (HLL) is designed to be used by a human and is closer to the human language. Its programming style is easier to comprehend and implement than a lower-level programming language (LLL). A high-level programming language needs to be converted to machine language before being executed, so a high-level programming language can be slower.

A low-level programming language, on the other hand, is a lot closer to the machine and computer language. A low-level programming language can be executed directly on computer without the need to convert between languages before execution. Thus, a low-level programming language can be faster than a high-level programming language. Low-level programming languages like the assembly language are more inclined towards machine language that deals with bits 0 and 1.

**R is a HLL because it shares many similarities to human languages**. For example, in R programming code,



Output:



The R programming code is more like human language. A low-level programming language like the assembly language is more towards the machine language.

**What Is Statistics?**

Statistics is a collection of mathematics to deal with the organization, analysis, and interpretation of data. Three main statistical methods are used in the data analysis: descriptive statistics, inferential statistics, and regressions analysis.

**Why R?**

When learning data science, many people struggle with choosing which programming languages and data sciences to learn. There are many programming languages available for data science, like R, Python, SAS, Java, and more. There are many data science software packages to learn, such as SPSS Statistics, SPSS Modeler, SAS Enterprise Miner, Tableau, RapidMiner, Weka, GATE, and more

We recommend learning R for statistics because it was developed for statistics in the first place. Python is a real programming language, so you can develop real applications and software via Python programming. Hence, if you want to develop a data product or data application, Python can be a better choice. R programming is very strong in statistics, so it is ideal for data

exploration or data understanding using descriptive statistics, inferential statistics, regression analysis, and data visualizations. R is also ideal for modeling because you can use statistical learning like regressions for predictive analytics. R also has some packages for data mining, text mining, and machine learning like Rattle, CARET, and TM. R programming can also interface

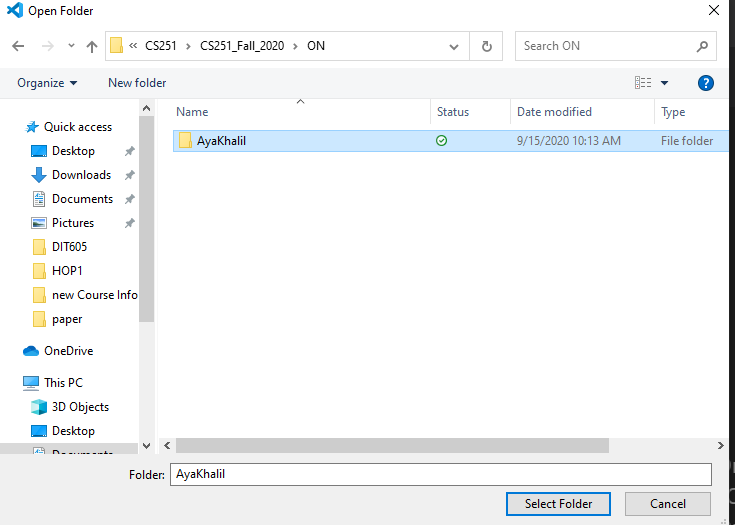
with big data systems like Apache Spark using Sparklyr. SAS programming is commercial, and Java has direct interfaces with GATE, Stanford NLP, and Weka. SPSS Statistics, SPSS Modeler, SAS Enterprise Miner, and Tableau are data science software packages with GUIs and are commercial. RapidMiner, Weka, and GATE are open source software packages for data

science.

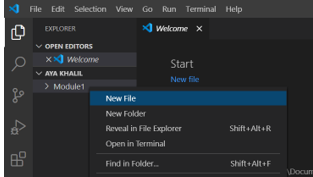
R is also heavily used in many of the companies that hire data scientists. Google and Facebook have data scientists who use R. R is also used in companies like Bank of America, Ford, Uber, Trulia, and more.

**You should be in:**

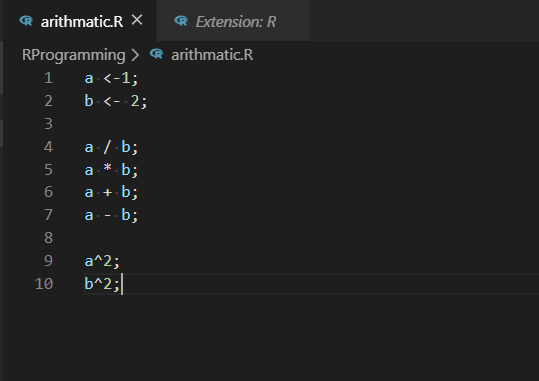
* **onsite student:** Open CS251\_Fall\_2020/**IN**/FirstnameLastname (File> open)
* **online student:** Open CS251\_ Fall \_2020/**ON**/FirstnameLastname (File >Open)



* In Module1 project folder, create new file arithmetic.R
* Right-Click on Module1 and select new file

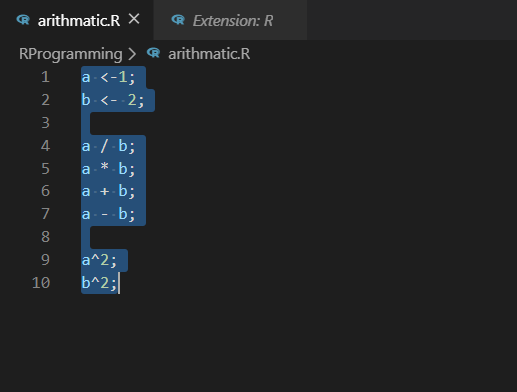


- Type the following code in arithmetic.R file



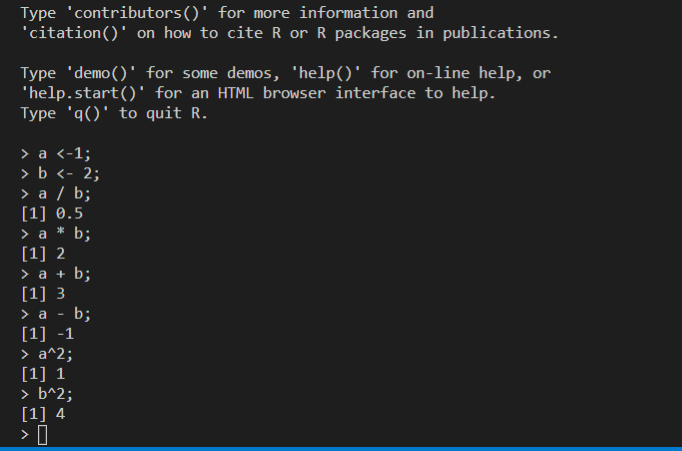
To run the code:

1. Select Terminal > New Terminal
2. In the terminal type R
3. Select your code by ctrl+a



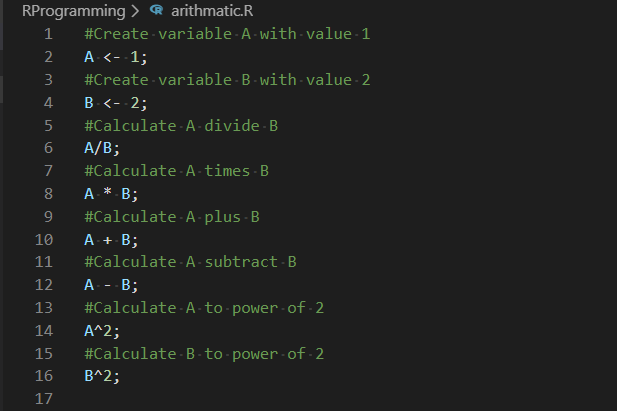
1. Make sure that you selected your code by ctrl+a, then ctrl+enter

-Your output will be :



**Adding Comments to the Code**

You can add comments to the code. Comments are text that will not be run by the R console. You can add in a comment by putting # in front of the text. The comment is for you to describe your code to let anyone read it more easily.



**Variables**

Let's look into the code and scripts we used previously. we actually created two variables, A and B, and assigned some values

to the two variables.

A <- 1

B <- 2

In this code, A is a variable, and B is a variable also. <- means assign. A <- 1 means variable A is assigned a value of 1. 1 is a

numeric type. B <- 2 means variable B is assigned a value of 2. 2 is a numeric type.

If you want to assign text or character values, you add quotations, like

A <- "Hello World"

Variable A is assigned a text value of "Hello World". Character and numeric are data types.

**Data Types**

Data types are the types or kind of information or data a variable is holding.

For example,

A <- "abc"

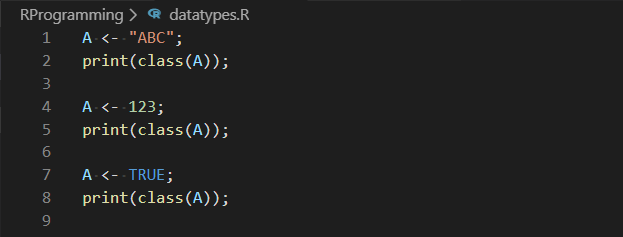
B <- 1.2

In R, data types are automatically determined. Because of the quotations surrounding the values, variable A is of the character data type, while variable B is of the numeric data type.

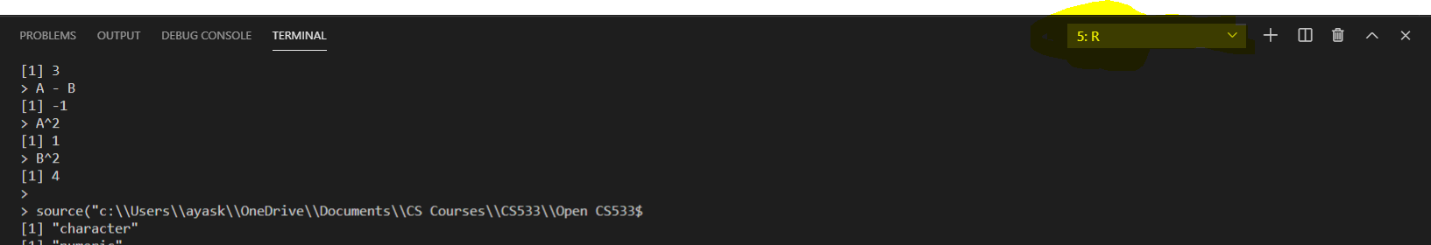
**You should be in:**

* **onsite student:** CS251\_Fall\_2020/**IN**/FirstnameLastname
* **online student:** CS251\_ Fall \_2020/**ON**/FirstnameLastname
* In Module1 project folder, create new file dataTypes.R

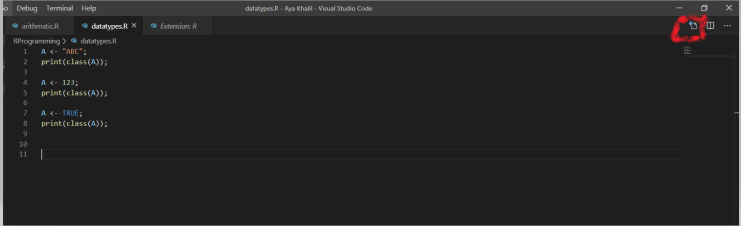
Type the following code in dataTypes.R file



1. Make sure that your terminal is R



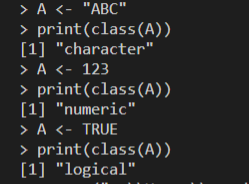
1. You can run your code with different ways
   1. **First way:** Press shift+ctrl+s
   2. **Second way**: press the Run Source button



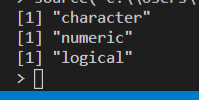
* 1. Third way: ctrl+a , then ctrl+enter

1. Your output will be:

If you used the third way, your output will be:



-If you used the first or second way, your output will be:



**Why is the data type important?**

If you do a math calculation in R and one variable's data type is numeric and one variable's data type is non-numeric, you will get the following error:

> A <- 123;

> B <- "aaa";

> A + B;

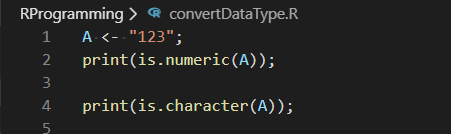
Error in A + B : non-numeric argument to binary operator

**You can also use is.datatype() to determine whether a variable is of a certain data type**

**You should be in:**

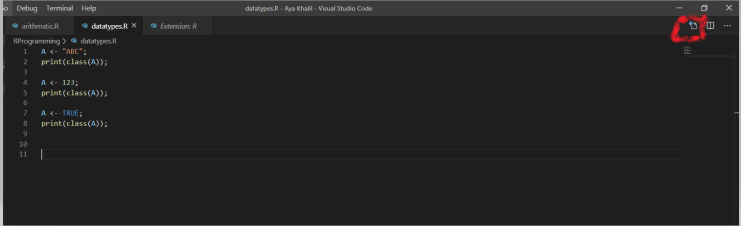
* **onsite student:** CS251\_ Fall \_2020/**IN**/FirstnameLastname
* **online student:** CS251\_ Fall \_2020/**ON**/FirstnameLastname
* In Module1 project folder, create new file converyDataTypes.R

Type the following code in convertDataTypes.R file



**Run your code:**

Select the Run Source button



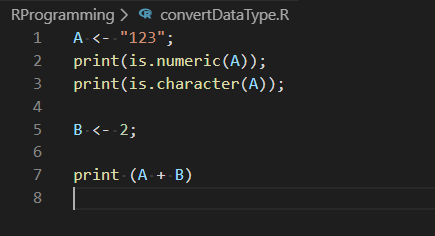
**Output:**



**Explanation:**

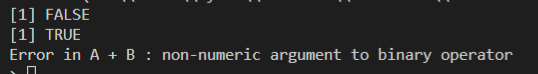
A is not numeric because the value of A “123” between double quotes which makes it a character data type not numeric

* **Type the following to update convertDataType.R file**



**Run your code:** SelectRun Source button

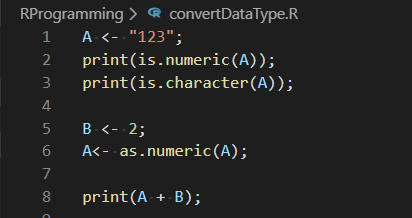
Output:



**To fix that error, we need to convert the value of A into numeric value**

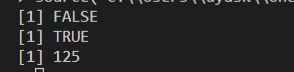
we can use as.datatype() to convert between data types

* **Type the following to update convertDataType.R file**



**Run your code:** SelectRun Source button

Output:



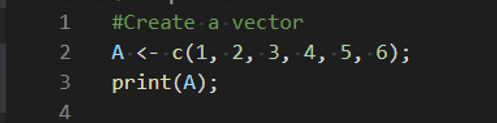
**Vectors**

A vector is a basic data structure or R object for storing a set of values of the same data type. A vector is the most basic and common data structure in R. A vector is used when you want to store and modify a set of values. The data types can be logical, integer, double, and character. The integer data type is used to store number values without a decimal, and the double data type is used to store number values with a decimal. Vectors can be created using the c() function

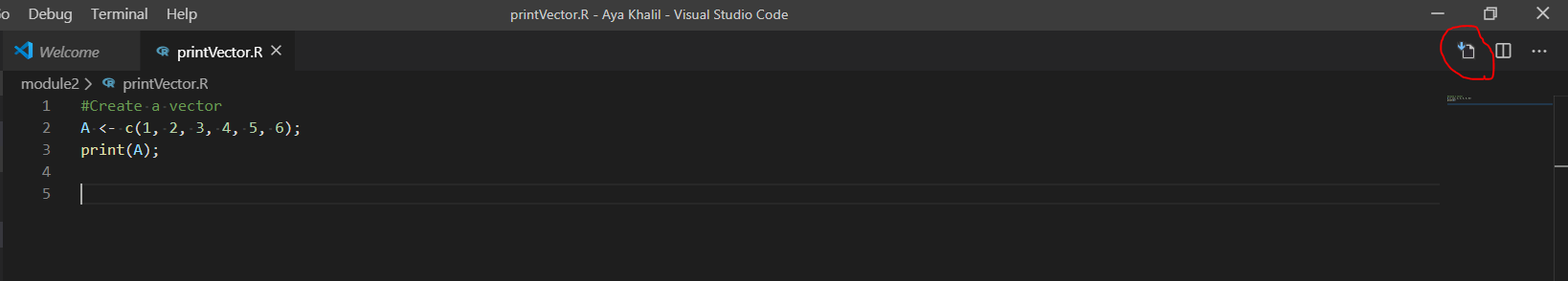
**You should be in:**

* **onsite student:** CS251\_ Fall \_2020/**IN**/FirstnameLastname
* **online student:** CS251\_ Fall \_2020/**ON**/FirstnameLastname
* In Module1 project folder, create printVector.R

**Type the following code in printVector.R**



**-**Run your code: Select Run Source button

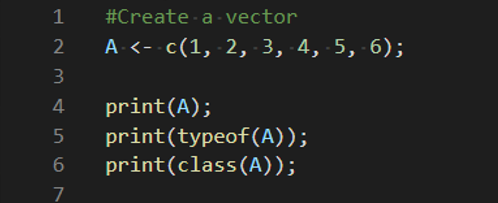


Output:



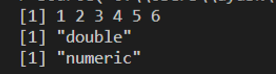
You can check the data type of the vector using typeof() and class():

* **Type the following code to update printVector.R**



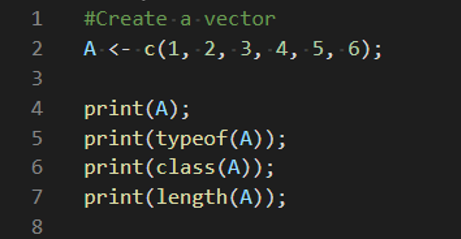
-Save your code, file>save or ctrl+s

-Run your code: Select Run Source button

**Output:** 

You can check the number of elements or values in a vector using the length() function

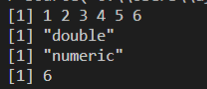
- **Type the following code to update printVector.R**



-Save your code, file>save or ctrl+s

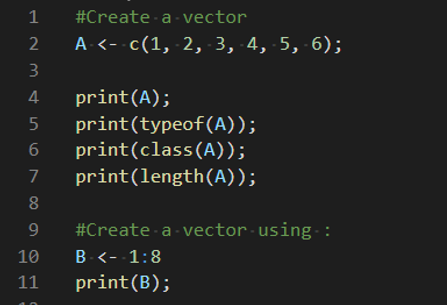
-Run your code: Select Run Source button

**Output**



You can also use the operator : to create a vector

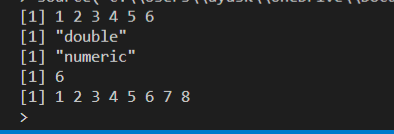
* **Type the following code to update printVector.R**



-Save your code, file>save or ctrl+s

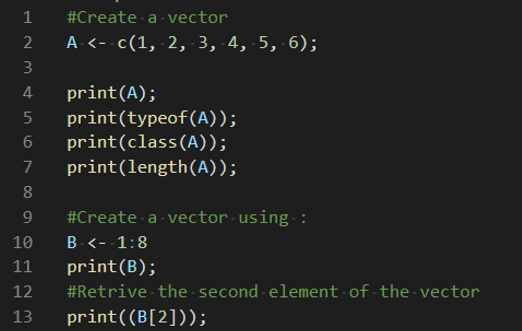
-Run your code: Select Run Source button

**-Output**



To retrieve the second element or value of a vector, use the [] brackets and put in the element number to retrieve

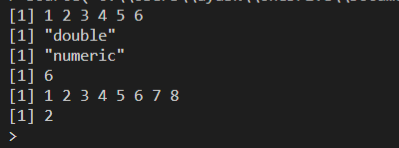
* **Type the following code to update printVector.R**



-Save your code, file>save or ctrl+s

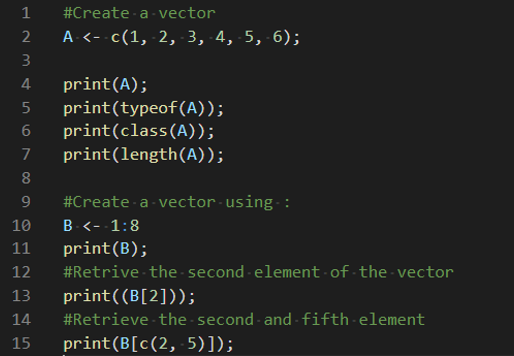
-Run your code: Select Run Source button

**-Output**



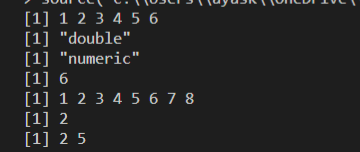
You can also retrieve the elements in the vector using another vector, for example, to retrieve the second and fifth element.

* **Type the following code to update printVector.R**



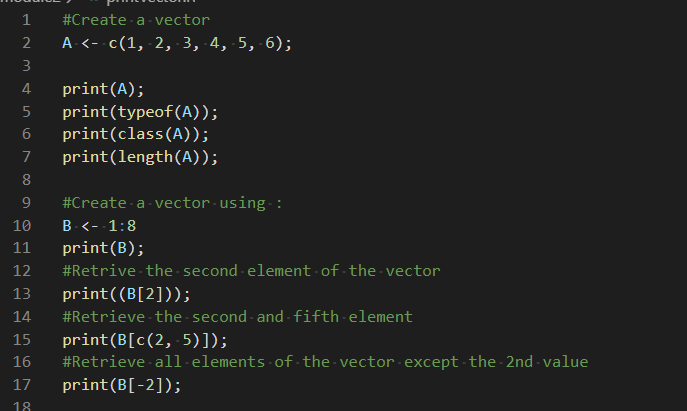
-Run your code: Select Run Source button

**-Output**



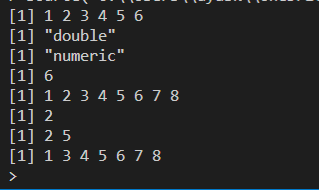
You can retrieve all elements except the second element

- **Type the following code to update printVector.R**



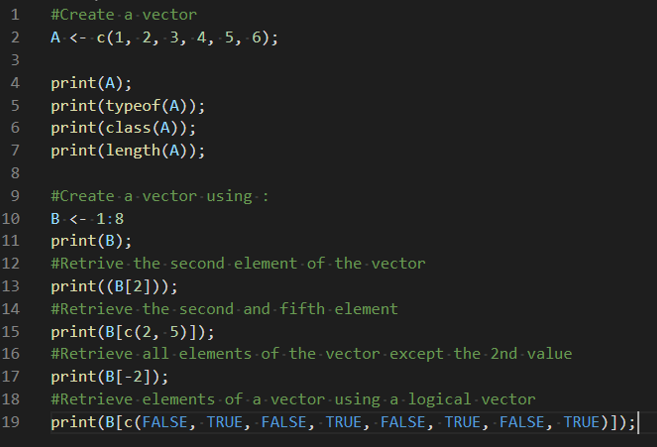
-Run your code: Select Run Source button

**-Output**



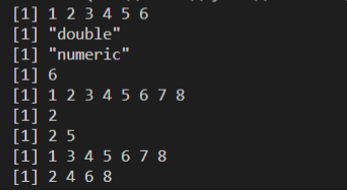
You can also retrieve elements of a vector using a logical vector

- **Type the following code to update printVector.R**



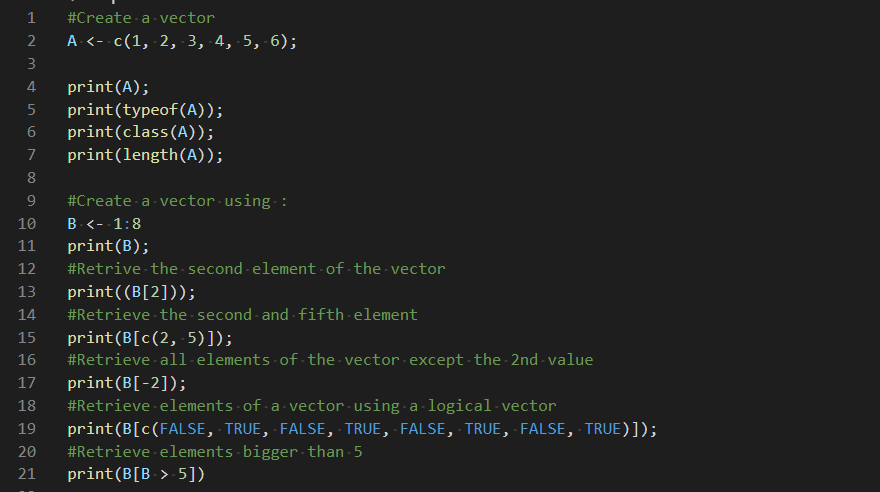
-Run your code: Select Run Source button

**-Output**



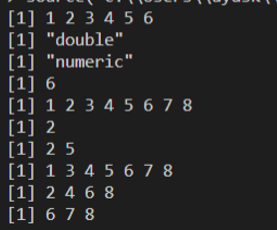
You can also use more than or less than signs to retrieve elements

* **Type the following code to update printVector.R**



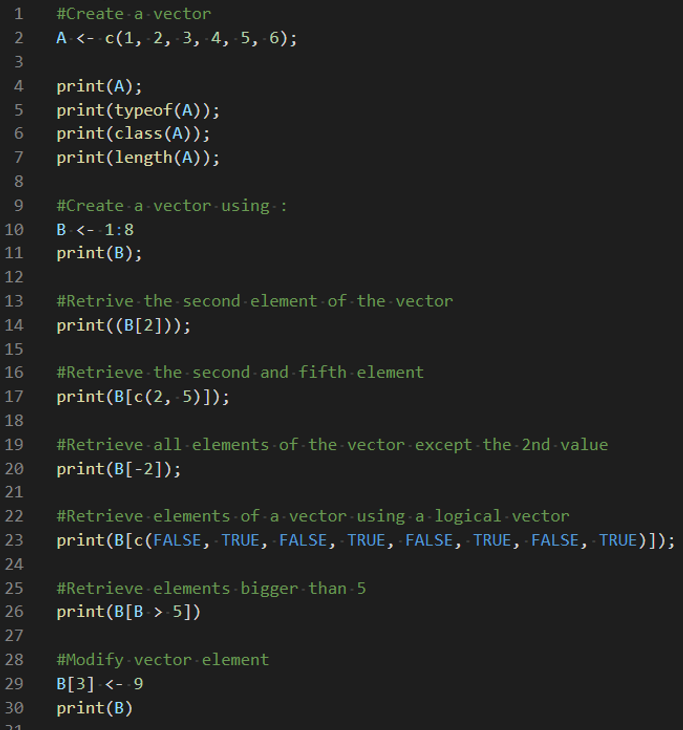
-Run your code: Select Run Source button

**-Output**



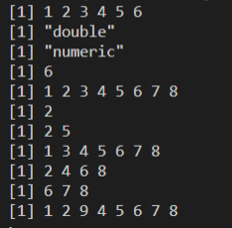
You can modify a vector as follows using assign, <-

* **Type the following code to update printVector.R**



-Run your code: Select Run Source button

**-Output**

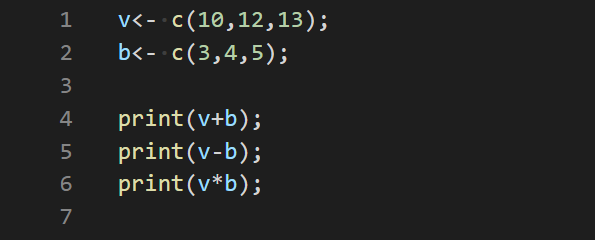


You can perform arithmetic operations on vectors.

**You should be in:**

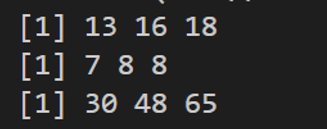
* **onsite student:** CS251\_ Fall \_2020/**IN**/FirstnameLastname
* **online student:** CS251\_ Fall \_2020/**ON**/FirstnameLastname
* In Module1 project folder, create new file arithmaticOperVect.R

Type the following code in arithmaticOperVect.R file



Run your code: Select Run Source button

**Output**

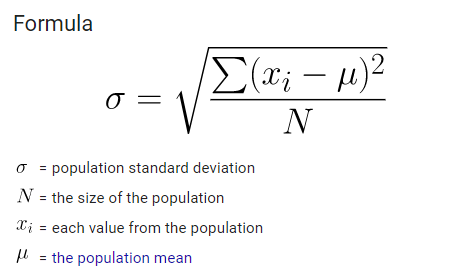


**Let’s use the vector to do some simple statistical calculations**

**Standard deviation:**

In statistics, the standard deviation is a measure of the amount of variation or dispersion of a set of values. A low standard deviation indicates that the values tend to be close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range.

**Standard deviation equation:**

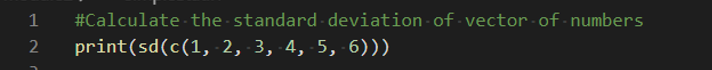


**You should be in:**

* **onsite student:** CS251\_ Fall \_2020/**IN**/FirstnameLastname
* **online student:** CS251\_ Fall \_2020/**ON**/FirstnameLastname

In Module1 project folder, create new file simpleStat.R

Type the following code in simpleStat.R file



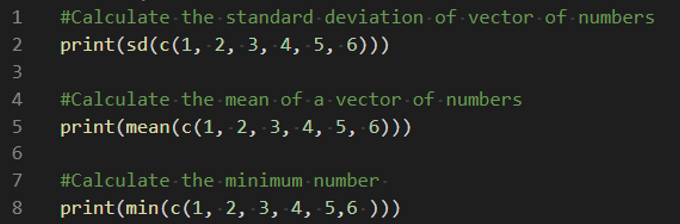
Run your code: Select Run Source button

**-Output** 

**Mean:**

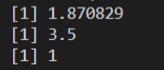
The statistical mean refers to the mean or average that is used to derive the central tendency of the data in question. It is determined by adding all the data points in a population and then dividing the total by the number of points. The resulting number is known as the mean or the average

**- Type the following code to update simpleStat.R file**



-Run your code: Select Run Source button

**-Output**



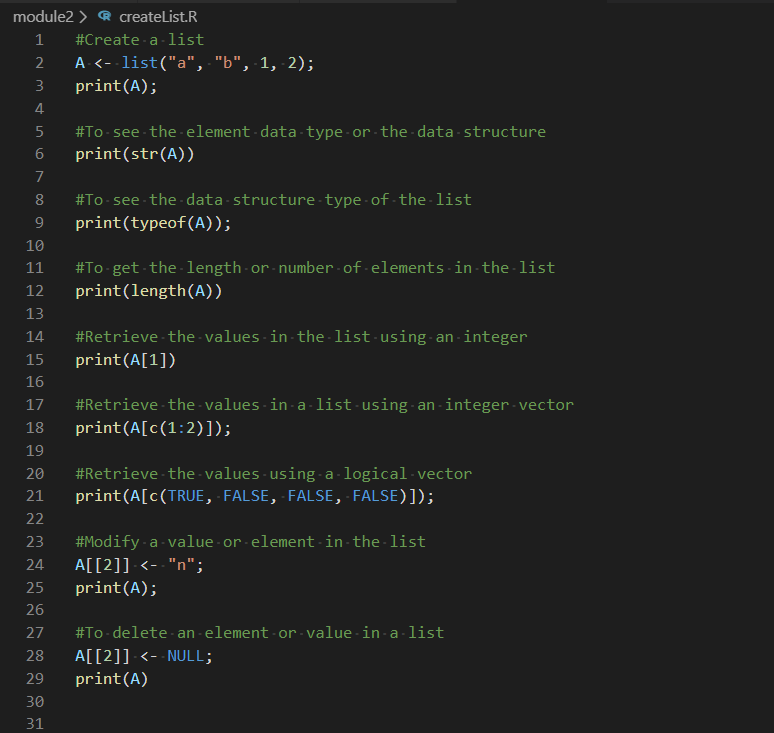
**Lists**

A list is like a vector. It is an R object that can store a set of values or elements, but a list can store values of different data types. A list is also another common data structure in R. You use a list when you want to modify and store a set of values of different data types. A vector can only store values of the same data type.

**You should be in:**

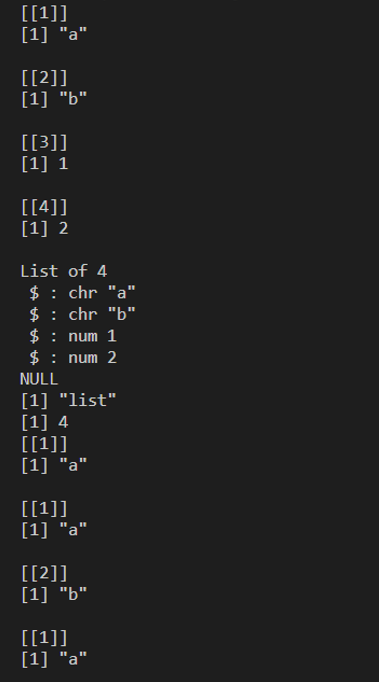
* **onsite student:** CS251\_ Fall \_2020/**IN**/FirstnameLastname
* **online student:** CS251\_ Fall \_2020/**ON**/FirstnameLastname
* In Module1 project folder, create new file createList.R

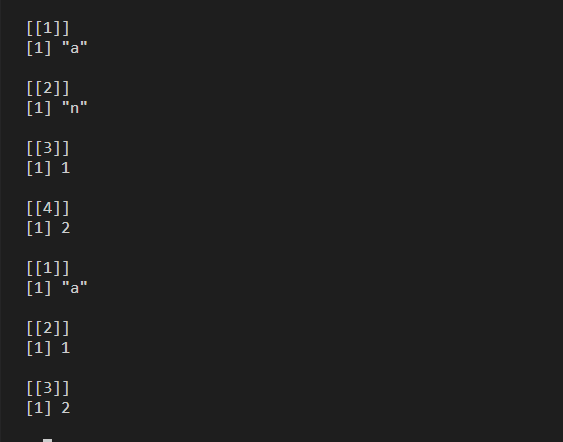
Type the following code in createList.R file



Run your code: Select Run Source button

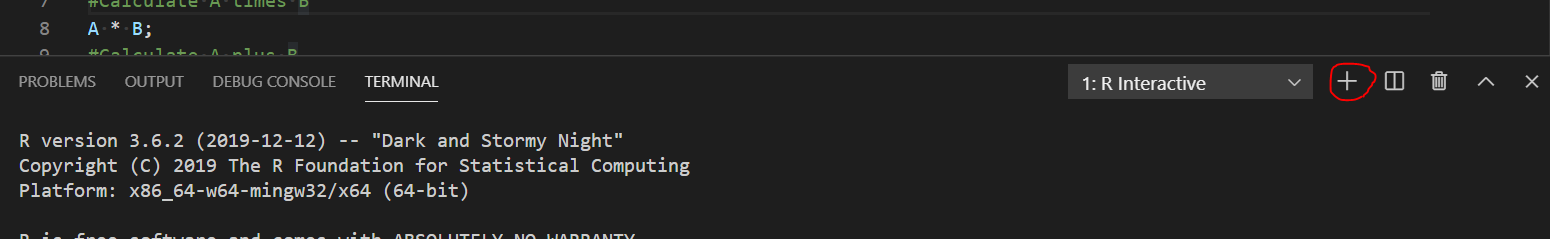
**Output**



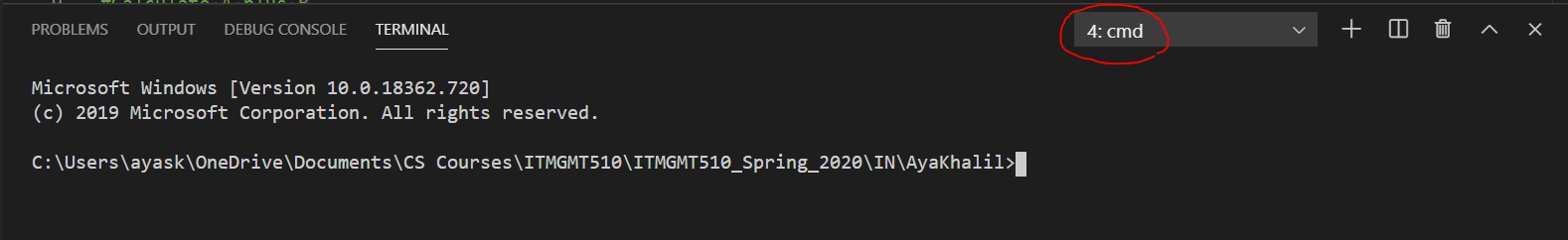


**Push your work to GitHub**

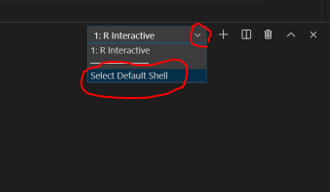
Open the cmd terminal,



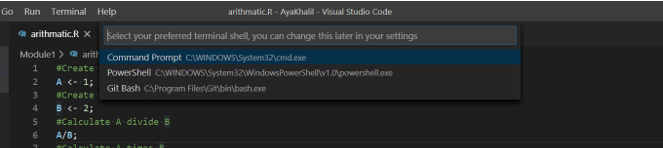
After selecting the + button, make sure the terminal is cmd



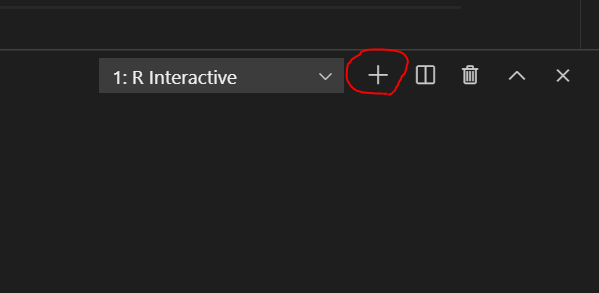
If the terminal not cmd, then you need to add it.



Then select command prompet



Then select the + button





**Make sure you are in**

Onsite students: CS251\_ Fall \_2020/**IN**/FirstnameLastname

Online students: CS251\_ Fall \_2020/**ON**/FirstnameLastname

Run the following commands to push your work to the GitHub repository:

Open the terminal from the VSCode by hit the control + ~ key and type the following command:

>>> git add .

>>> git commit -m “Submission for Module 1”

>>> git push origin YOUR\_BRANCH\_NAME

Note: you should change the YOUR\_BRANCH\_NAME to your own branch name. It should be firstname-lastname (e.g. maria-gracia).

If you cannot remember, run the command “git status” to check