CS 251 Statistical Computing

HOP2A: R for statistical project

2/19/2020 Developed by Aya Khalil

04/02/2020 Reviewed by Apiwat Chuaphan

10/16/2020 Reviewed by Shanshan Yu

School of Technology and 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**

* Matrix
* Data Frame

**Resource**

* 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.
* Data Science and Machine Learning BootCamp with R online course

**Matrix**

A matrix is like a vector, but it has two dimensions. You usually use a matrix to modify and store values from a data set because a matrix has two dimensions. A matrix is good when you plan to do linear algebra types or mathematical operations. For a data set with different types, you need to use a data frame.

To create a matrix, you can use the following syntax:

**variable <- matrix(vector, nrow=n, ncol=i)**

**Setup Working Environment for Module2**

1. Open VS Code.
2. **You should be in:**

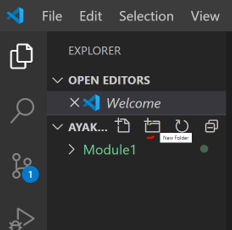
Desktop/CS251/hop02-YourUserName/

1. Then, Create “**Module2**” directory in the VSCode.

>>>mkdir Module2

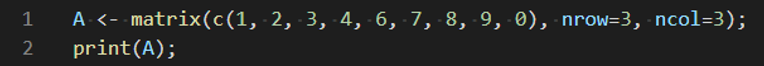
**OR**

Click on New Folder button that behind your name, and name the folder Module2



4)In Module2 project folder, create createMatrix.R

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

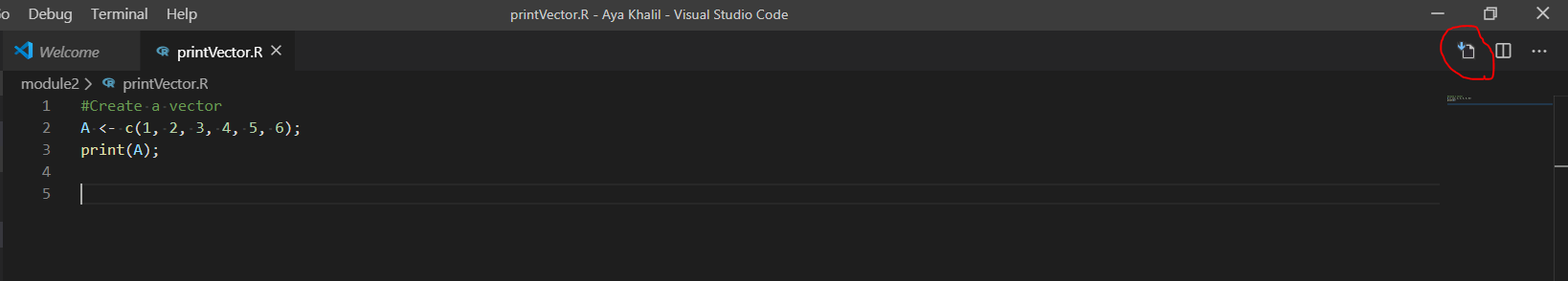


-Save your code, file>save or ctrl+s

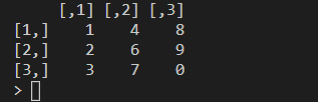
-Run your code: Select Run Source button:

-When you select the button for first time, it will open the R terminal

- Select the same button one more time to run and display the output of your code.

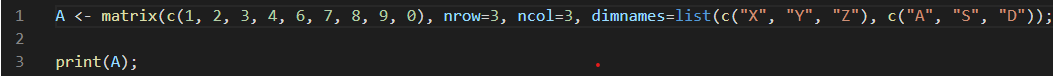


Output:



**You can use dimnames to rename the rows and columns**

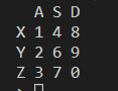
* Type the following to update createMatrix.R



Save your code, file>save or ctrl+s

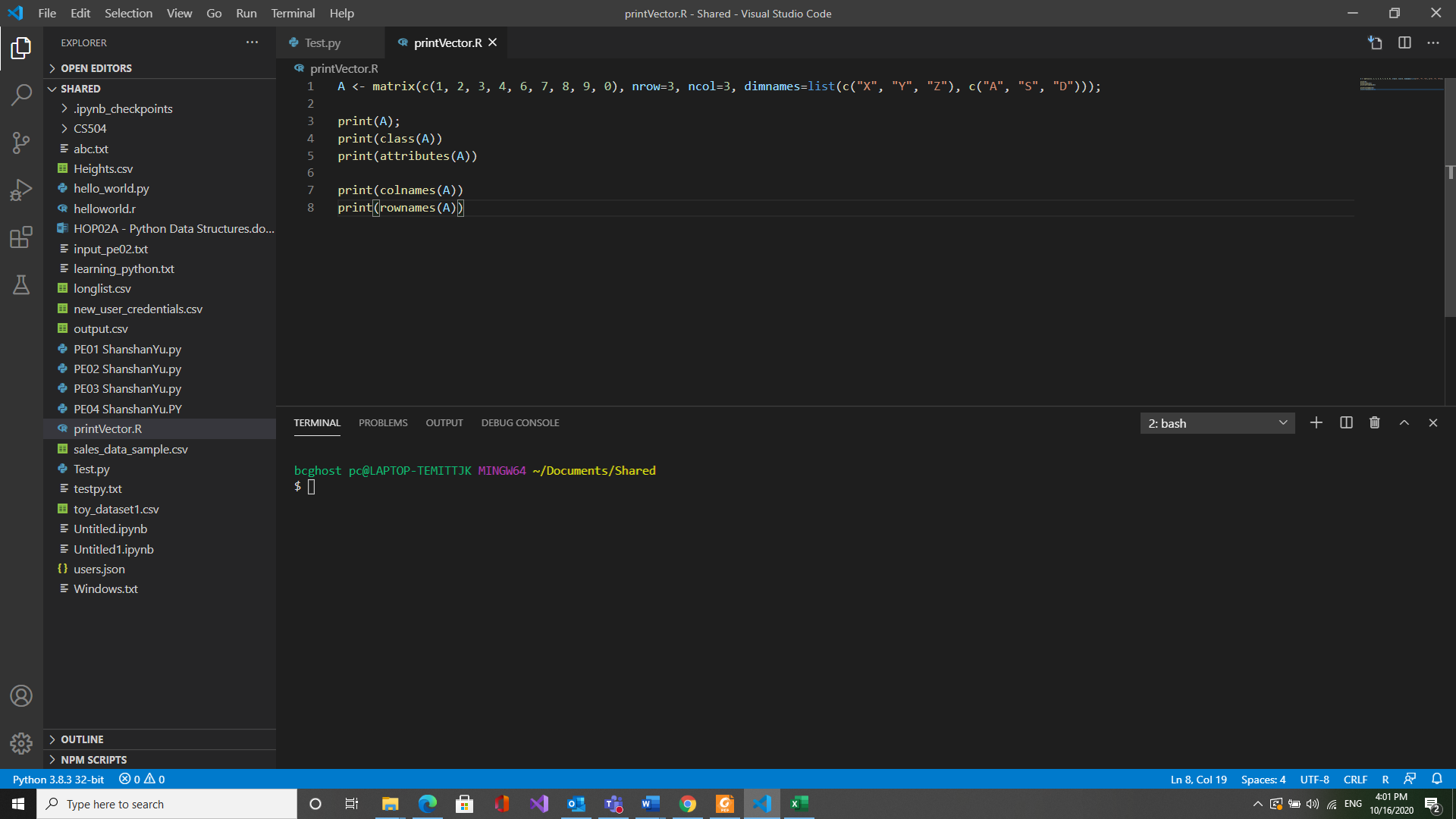
-Run your code: Select Run Source button

Output:



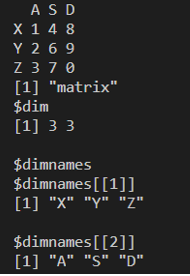
You can check the dimension of the matrix using the attributes() function and whether a variable is a matrix using the class()function

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



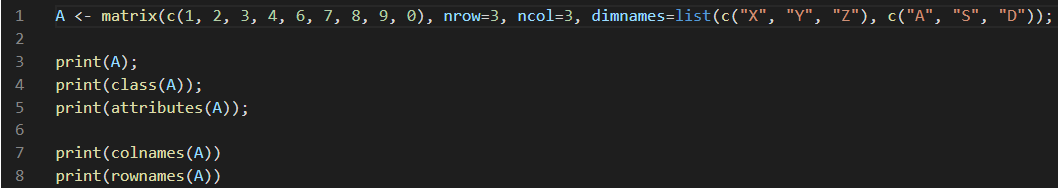
-Run your code: Select Run Source button

Output:



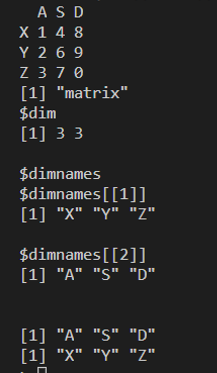
You can get column names and row names using the colnames() and rownames()

* Type the following to update createMatrix.R



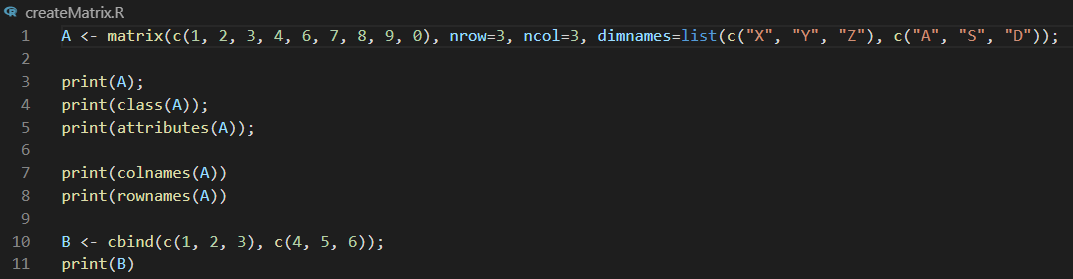
-Run your code: Select Run Source button

Output:



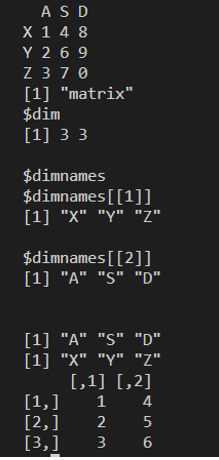
You can also create a matrix by using column binding and row binding functions

* Type the following to update createMatrix.R

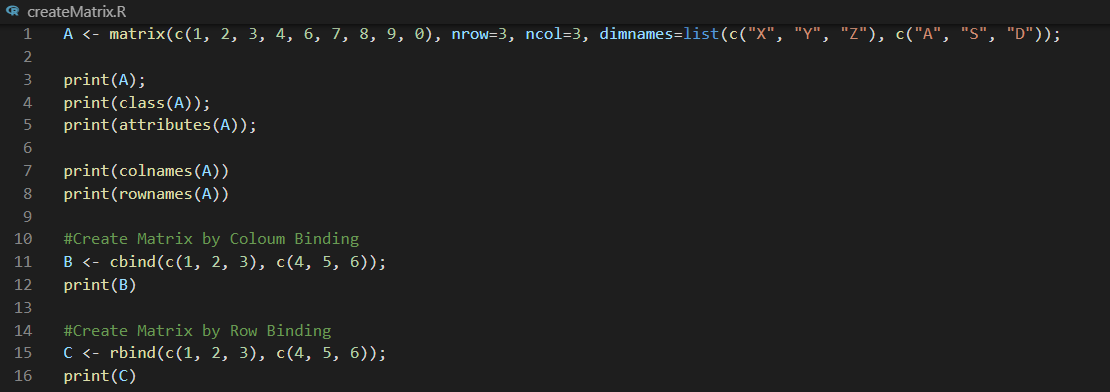


-Run your code: Select Run Source button

Output:

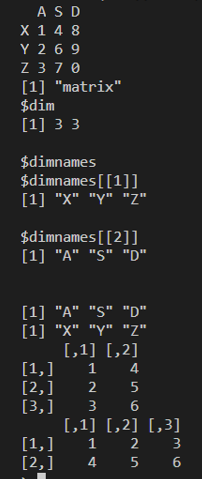


* Type the following to update createMatrix.R



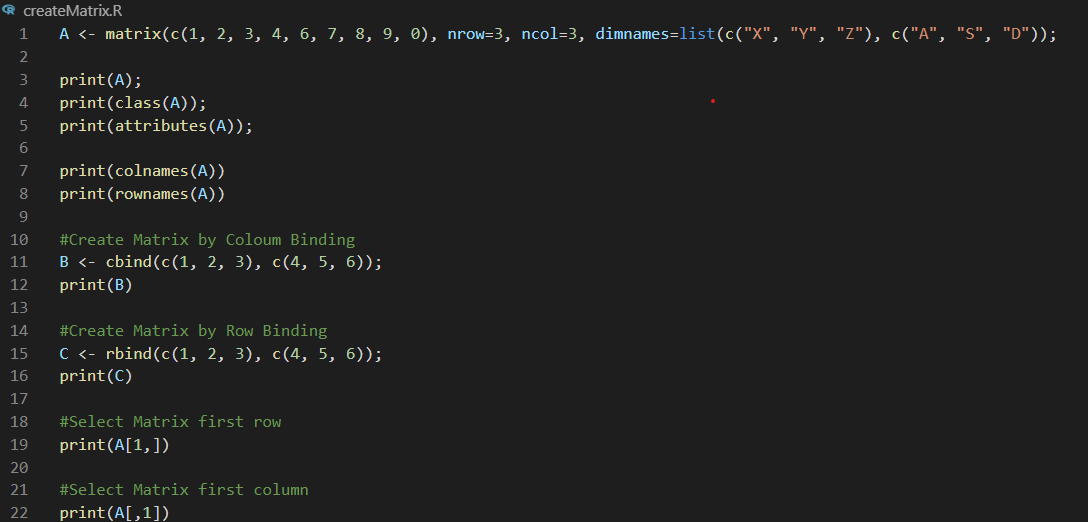
-Run your code: Select Run Source button

Output:



You can select first row or first column of the matrix

* Type the following to update createMatrix.R



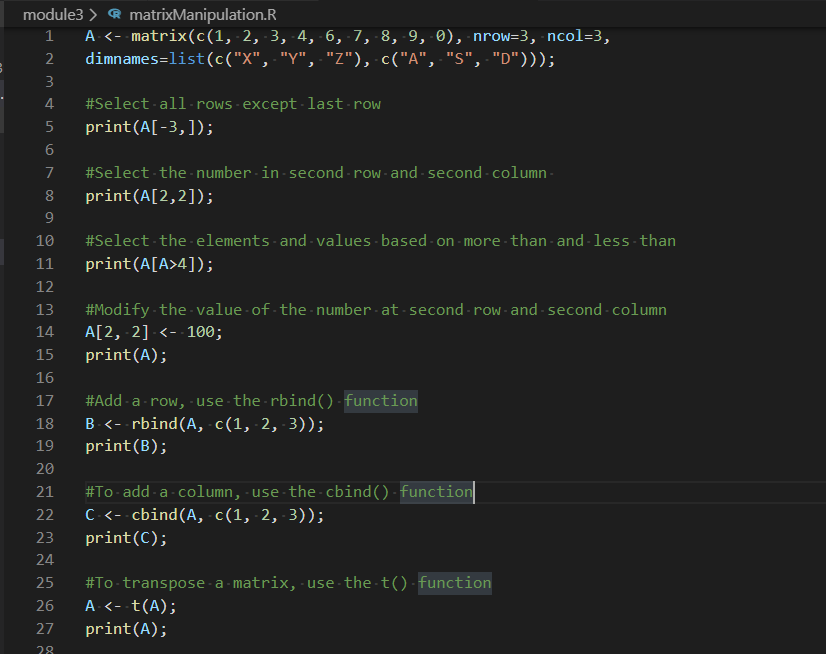
-Run your code: Select Run Source button

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file matrixManipulation.R

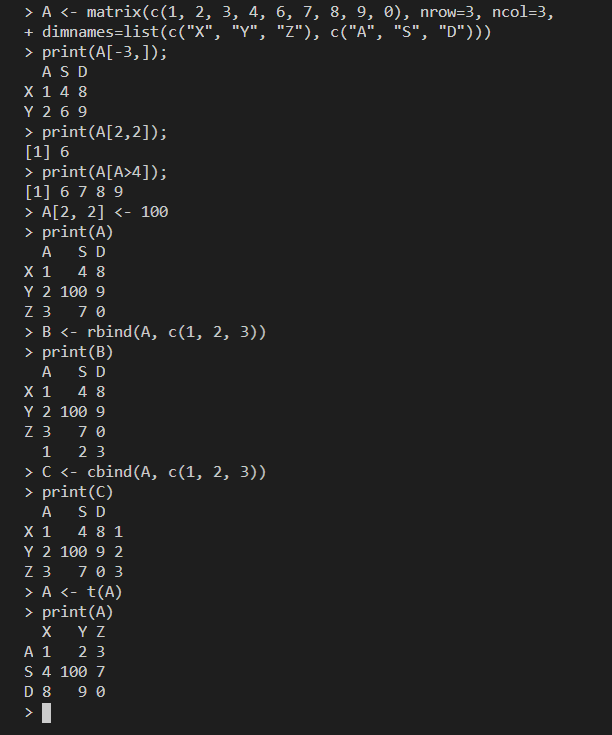
Type the following code in matrixManipulation.R file

***\*Please Run your code after each print to see the output***



Run your code: ctrl+A, then ctrl+enter

Output:



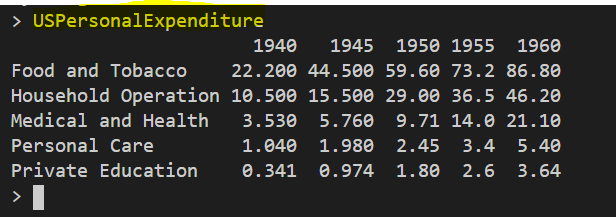
**Data Frame:**

A data frame is a special list or R object that is multidimensional and is usually used to store data read from an Excel or .csv file. A matrix can only store values of the same type, but a data frame can store values of different types.

R has built in Data Frames for quick reference to play around with! Check out the following data frames that are built-in

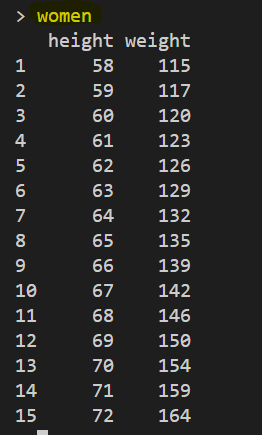
For example, Type in the R console

USPersonalExpenditure



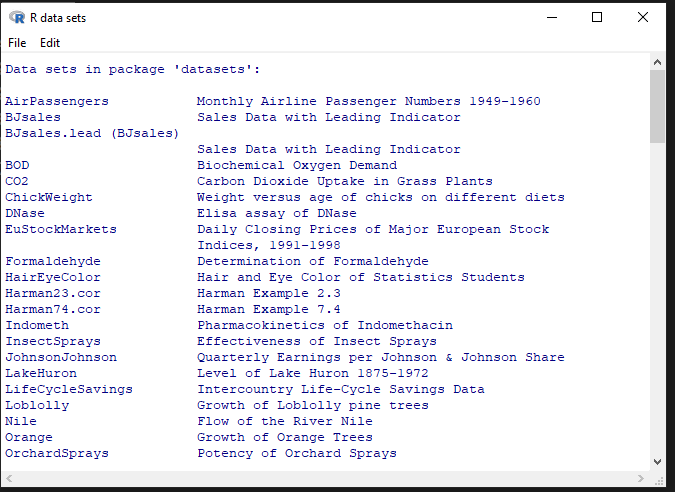
Or Type

women



To get a list of all available built-in dataframes, type in the R console  **data()**

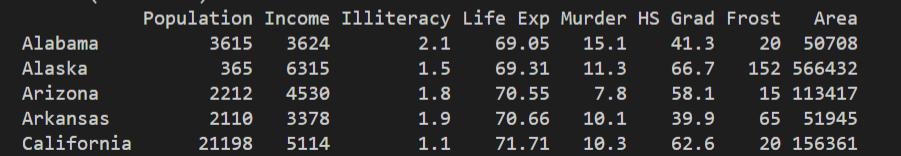
You will see a list of all available built-in dataframes



the R console, type

state.x77

You will see the below

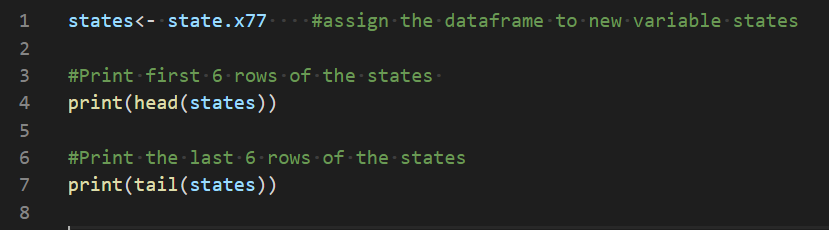


You will notice that the dataframe is too big, so  we can use the **head()** and **tail()** functions to view the first and last 6 rows respectively.

**You should be in:**

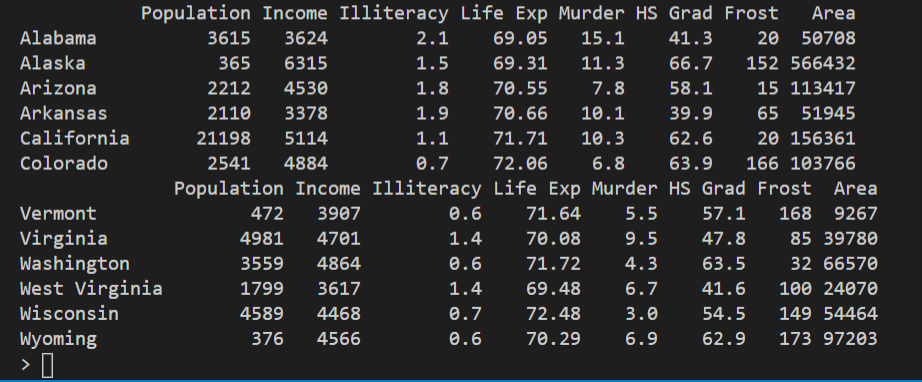
* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file WorkWithDFrame.R

Type the following in WorkWithDFrame.R



-Run your code: Select Run Source button

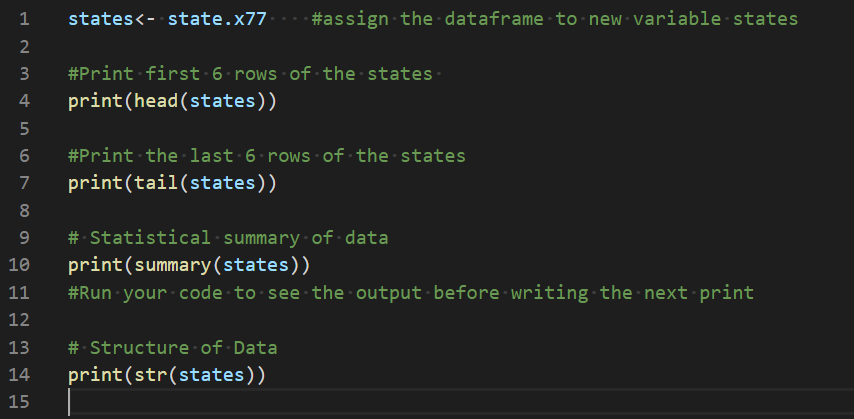
-Output:



**DataFrames - Overview of information**

We can use the **str()** to get the structure of a dataframe, which gives information on the structure of the dataframe and the data it contains, such as variable names and data types. We can use **summary()** to get a quick statistical summary of all the columns of a DataFrame, depending on the data.

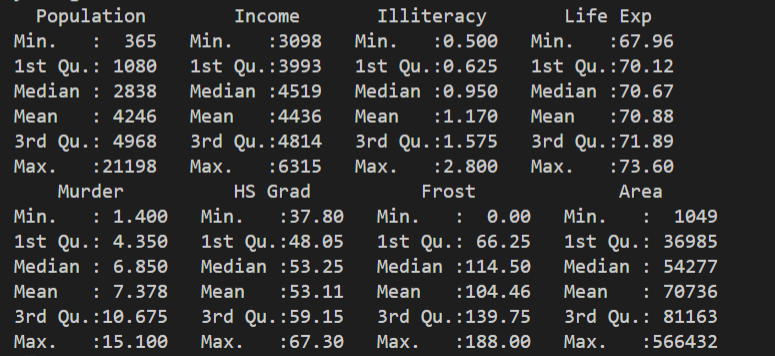
-Add the following to update WorkWithDFrame.R



Save & Run your code: Select Run Source button

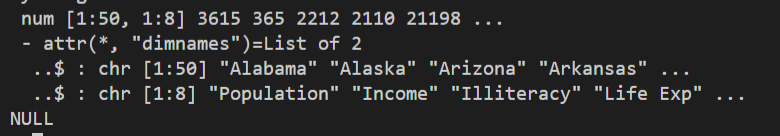
-Output:

**This will be the summary of the states data frame**



**The summary of the dataset shows the minimum, maximum, mean and median numbers in each column.**

**This will be the structure of the data frame**



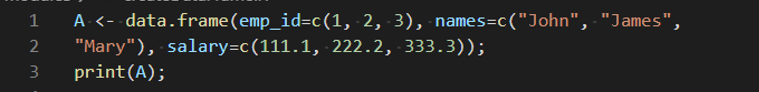
**Create Data Frame**

We can create data frames using the **data.frame()** function and pass vectors as arguments, which will then convert the vectors into columns of the data frame.

**You should be in:**

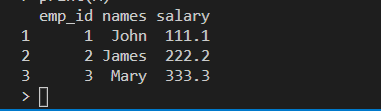
* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file createDataFrame.R

Type the following code in createDataFrame.R file



-Run the code

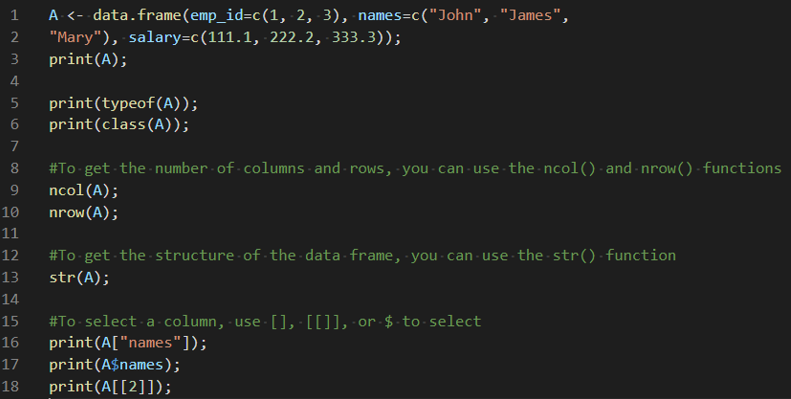
-output:



**You can use the typeof() and class() functions to check whether a variable is of the data frame type**

- Type the following code to update createDataFrame.R file

**\*Please run your code after each print to see the output**

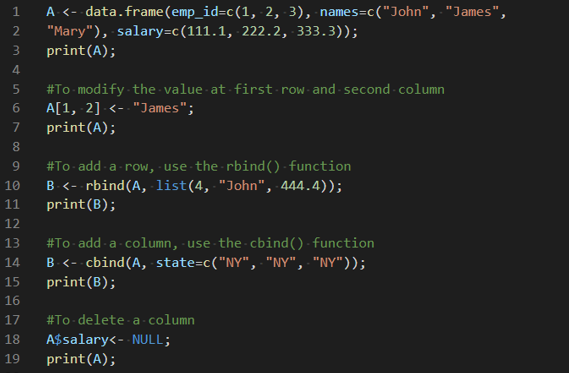


-Save and Run your code

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file DFrameManipulation.R

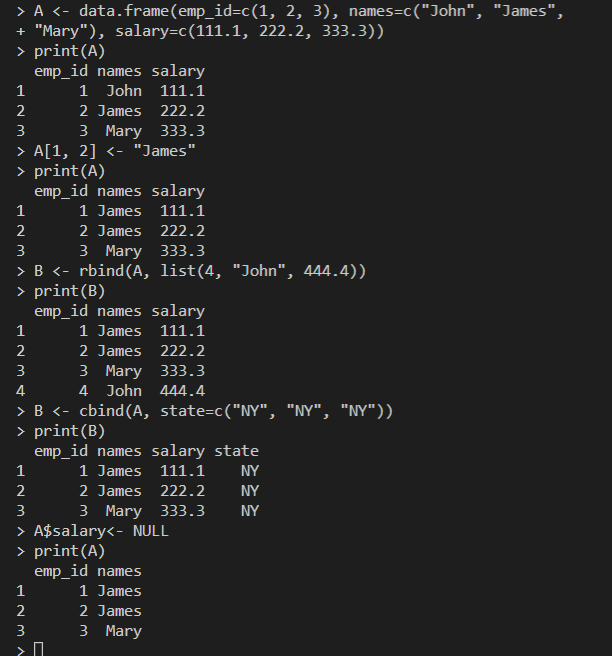
Type the following code in DFrameManipulation.R file



-Save the file: ctrl+s, or file>save

- Run your code: ctrl+ a , then ctrl+enter

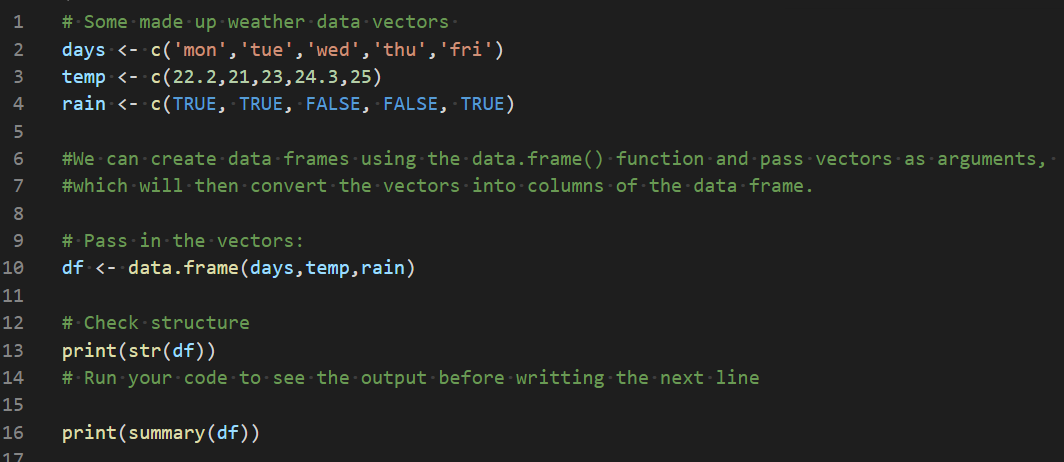
**\*run your code after each print to see the output**



**You should be in:**

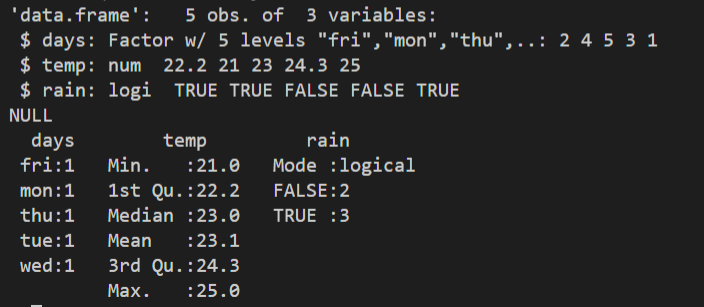
* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file DFrameCreation.R

Type the following in DFrameCreation.R file



-Save & Run your code: Select Run Source button

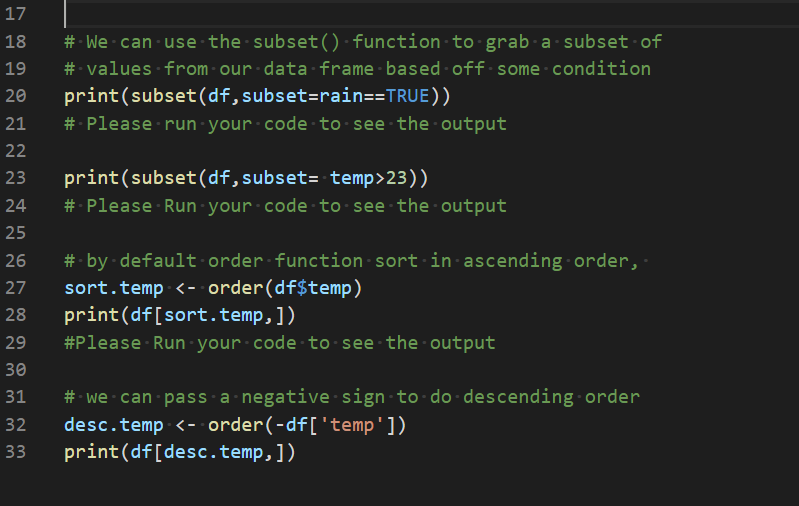
-Output:



## **Filtering with a subset condition**

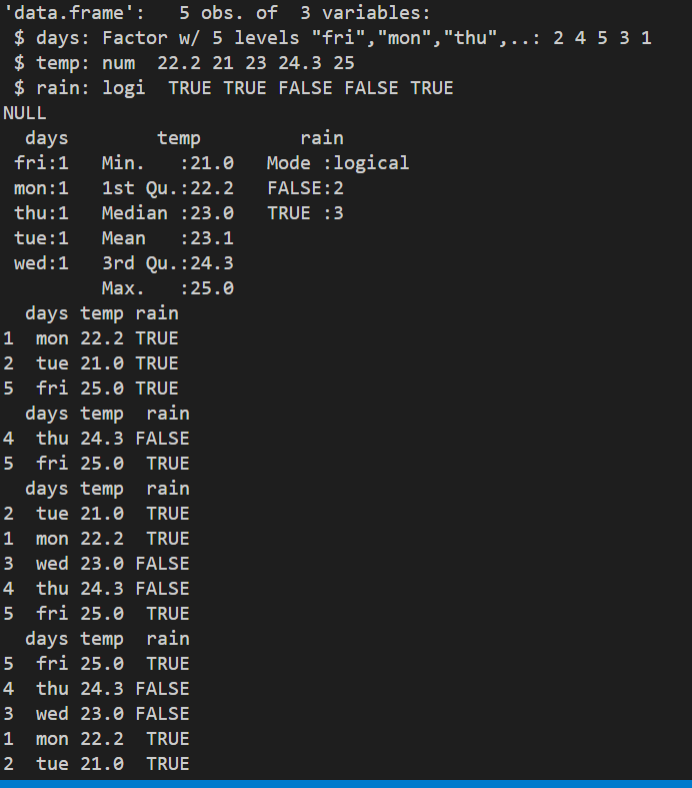
We can use the subset() function to grab a subset of values from our data frame based off some condition. So for example, imagine we wanted to grab the days where it rained (rain=True), we can use the subset() function

* Add the following to update DFrameCreation.R



-Save & Run your code: Select Run Source button

-Output:



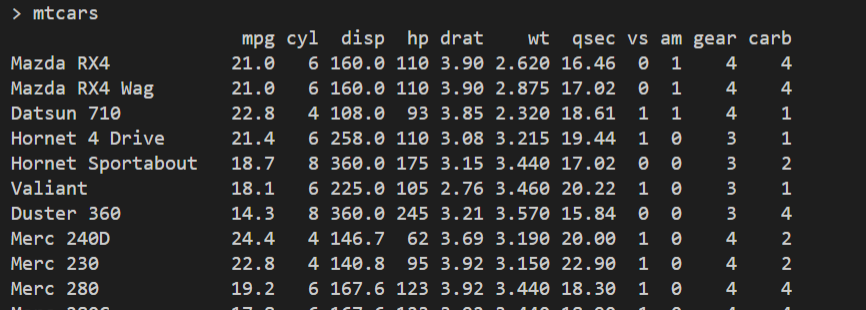
**Referencing Rows**

Usually you'll use the [row,] format

**Now, we will use another built-in dataset called mtcars**

**mtcars is the Motor Trend Car Road Tests dataset**

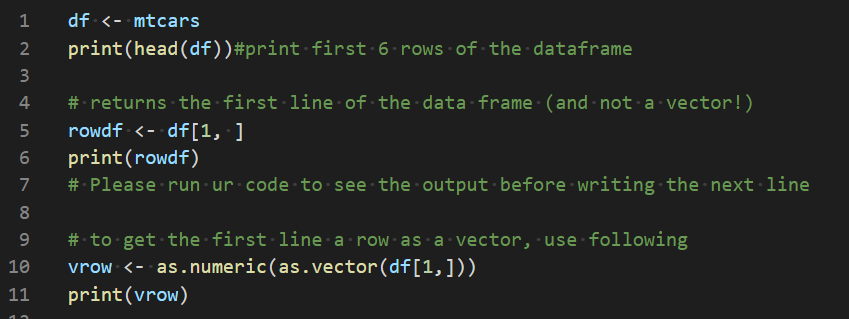
**-Type mtcars in the R console, you will see**



**You should be in:**

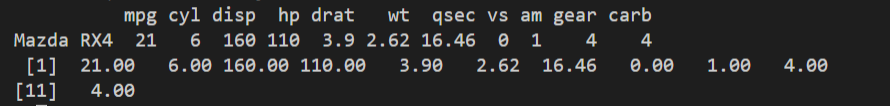
* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file DFRowRef.R

Type the following in DFRowRef.R file



**Save & Run your code**

**Output:**



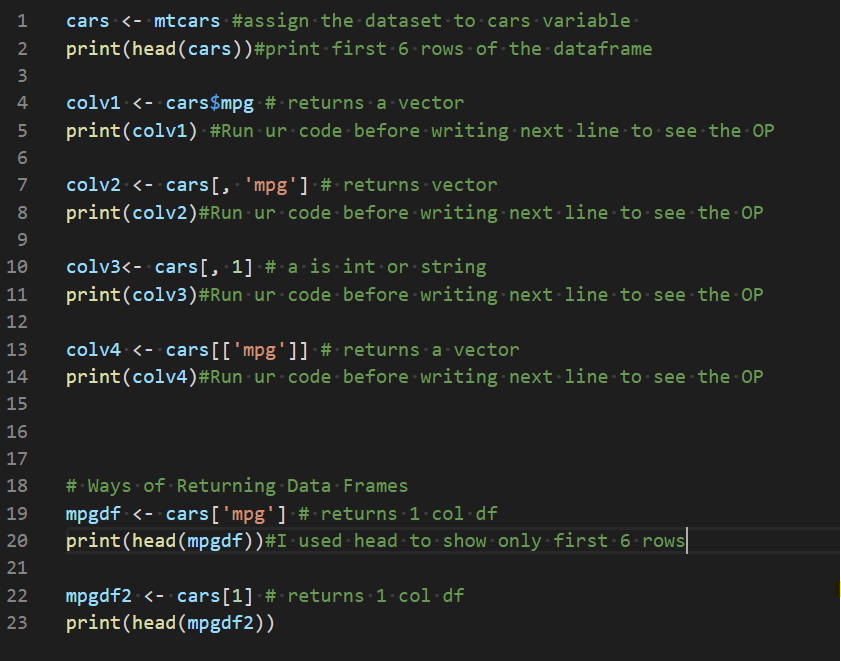
**Referencing Columns**

Most column references return a vector

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file DFColRef.R

Type the following in DFColRef.R file



**-Save & Run your code**

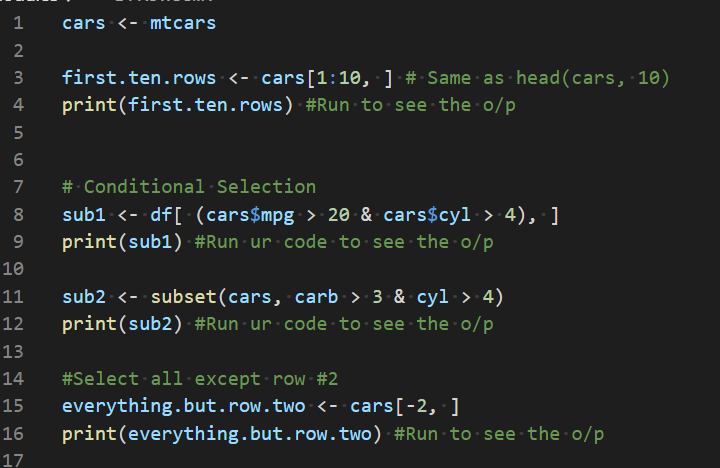
# **Selecting Multiple Rows**

Now, we will see how we can select more than one row

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file DFRowSel.R

Type the following in DFRowSel.R file



**-Challenge**

**Add to the previous code one line that Select the rows where all cars have 6 cylinders (cyl column)**

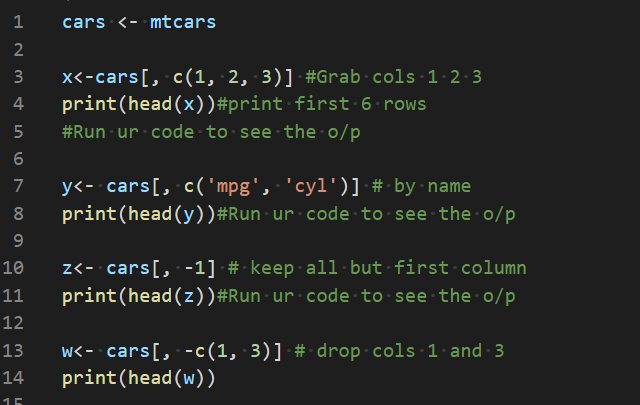
**Selecting Multiple Columns**

We can select more than one column

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file DFColSel.R

Type the following in DFColSel.R file



**-Save & Run your code**

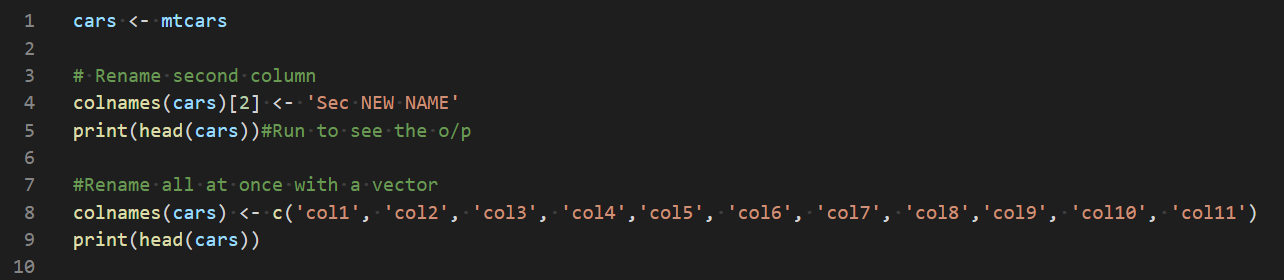
**Column Name**

We can assign names to the columns

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file DFColName.R

Type the following in DFColName.R file

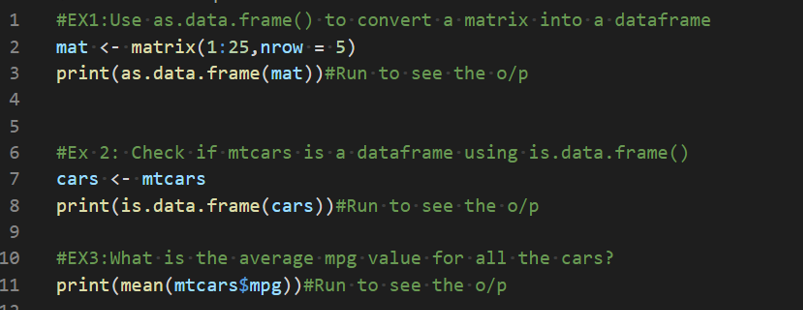


-Save & Run your code

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file DFmoreExamples.R

Type the following in DFmoreExamples.R file



Run your code.

**Try to solve the following questions. You do not have to submit it. These questions only to see if you understand the HOP.**

**Question1:**

Create three vectors  x,y,z  with integers and each vector has 3 elements. Combine the three vectors to become a 3×3 matrix  A  where each column represents a vector. Change the row names to  a,b,c.

**Question2:**

Select the columns am, gear, and carb from mtcars data frame

**Question3**:

Create a new column called performance which is calculated by hp/wt columns from mtcars data frame