|  |  |  |  |
| --- | --- | --- | --- |
| Date: | **28-07-2020** | Name: | **Varun G Shetty** |
| Course: | **Coursera** | USN: | **4AL17EC093** |
| Topic: | Basic Statestics | Semester & Section: | **6th & ‘B’** |
| GitHub Repository: | **Varunshetty4** |  |  |

**Report-**

R Language:

Dataframes:  A data frame has the variables of a data set as columns and the observations as rows. This will be a familiar concept for those coming from different statistical software packages such as SAS or SPSS.

There are several functions you can use to inspect your dataframe. To name a few

* head: this by default prints the first 6 rows of the dataframe
* tail: this by default prints the last 6 rows to the console
* str: this prints the structure of your dataframe
* dim: this by default prints the dimensions, that is, the number of rows and columns of your dataframe
* colnames: this prints the names of the columns of your dataframe

As a first goal, you want to construct a data frame that describes the main characteristics of eight planets in our solar system. The main features of a planet are:

* The type of planet (Terrestrial or Gas Giant).
* The planet's diameter relative to the diameter of the Earth.
* The planet's rotation across the sun relative to that of the Earth.
* If the planet has rings or not (TRUE or FALSE).

You construct a data frame with the data.frame() function. As arguments, you should provide the above mentioned vectors as input that should become the different columns of that data frame. Therefore, it is important that each vector used to construct a data frame has an equal length. But do not forget that it is possible (and likely) that they contain different types of data.

Indexing: now have multiple dimensions: rows and columns. That's why you can use a comma in the middle of the brackets to differentiate between rows and columns. For instance, the following code planet\_df[1,2] would select the element in the first row and the second column from the dataframe planet\_df.

You can also use the $ operator to select an entire column from a dataframe. For instance, planet\_df$planets would select the entire planets column from the dataframe planet\_df.

List(): A list in R is similar to your to-do list at work or school: the different items on that list most likely differ in length, characteristic, type of activity that has to do be done. A list in R allows you to gather a variety of objects under one name (that is, the name of the list) in an ordered way.

These objects can be matrices, vectors, data frames, even other lists, etc. It is not even required that these objects are related to each other. You can easily construct a list using the list() function. In this function you can wrap the different elements like so: list(item1, item2, item3).

Indexing: For example, to "grab" the first component of my\_list you type my\_list[[1]]. Another way to check is to refer to the names of the components: my\_list[["my\_vector"]] selects the my\_vector vector.

A last way to grab an element from a list is using the $ sign. The following code would select my\_df from my\_list: my\_list$my\_df. Besides selecting components, you often need to select specific elements out of these components. For example, with my\_list[[1]][1] you select from the first component of my\_list the first element. This would select the number 1.