LAB 01

DOCUMENTATION

**PART A.**

* Firstly, to be totally able to import data from yearly\_request.csv file, you have to point your R workspace to directory contains that yearly\_request.csv file.

Use command: setwd(“/path/of/directory/contains/ yearly\_request.csv”)

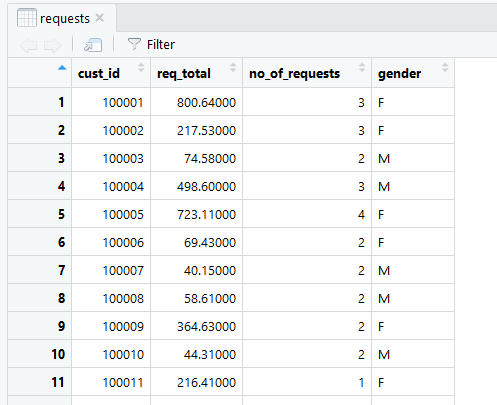


* Secondly, Import data in .csv file to **requests** data frame in R

Use command: requests <- read.csv(“yearly\_request.csv”)



When we execute the code above, it produces the following results…



* Next, to export the histogram of **number of requests** to **jpeg** file
* Create a jpeg file to store histogram

Use command: 

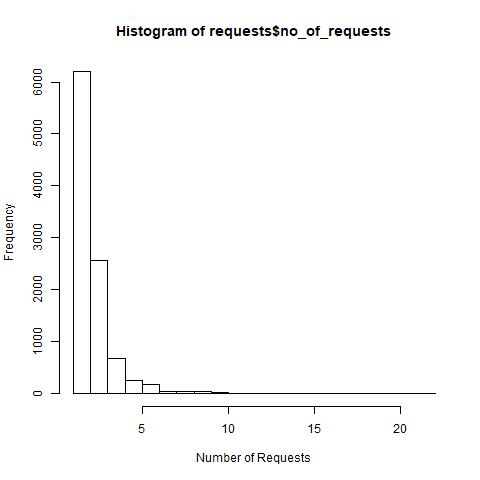
* R creates histogram using **hist()** function. This function takes a vector as an input and uses some more parameters to plot histograms.



* Save the file

Use command: 

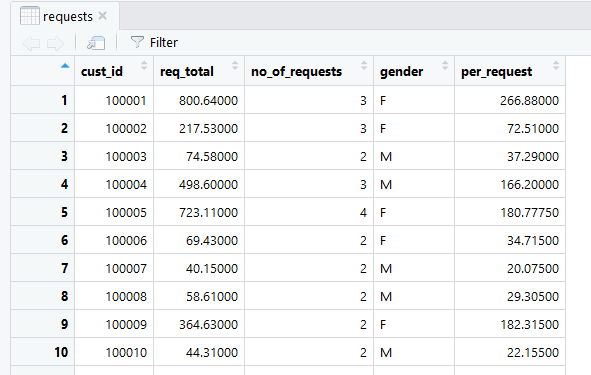
* The histogram shown



* Next, to add a column **per\_request** to dataset and exports the modified dataset to “requests\_modified.txt” file
* Assuming that new column **per\_request** data is caculated by column **req\_total** divide **no\_of\_request** so the command would be:



* A new column “**per\_request**” is added with suitable data



* Export edited data to txt file “ **requests\_modified.txt** ”

Use command:



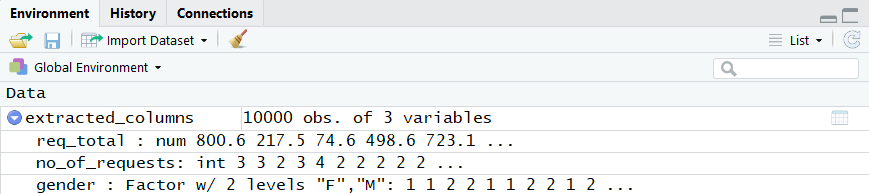
The data in the above command is added to file which is modified that data is separated by **tab spaces** and **row.names=TRUE** means and row numbers(oder numbers) would be written into file too.

* Extract the second, third, and fourth column of the **requests** data frame

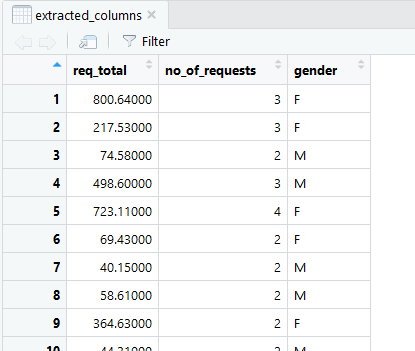
Corresponding to the question, the required columns needs extracting in current **requests** data frame are **req\_total**, **no\_of\_requests**, **gender.**

Use command: 

The results of the corresponding data :



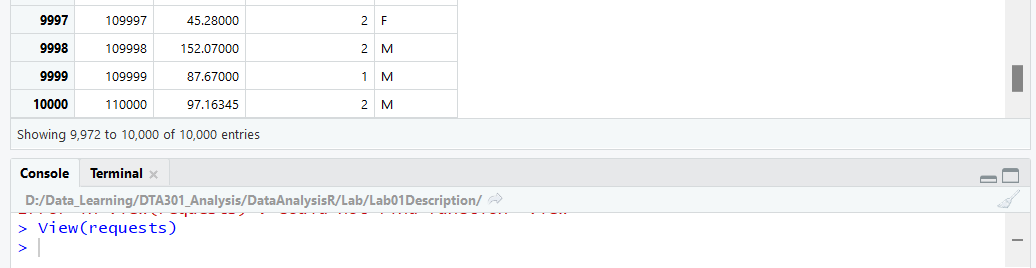
Data as a Table



* Come to next requirements, build an empty character vector of the same length as **requests** and **group the customers** according to the request amount
* Before **create** an empty character which has the same length as **requests** data frame, check the current length of rows in **requests.**

Use command: 

And we can see the maximum length of **requests** data frame is 10000 rows



* Next is to create the empty vector which has the same length as **requests (10000 rows)**

Use command: 

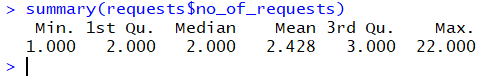
Check if the **group\_vector** has the length as 10000 rows



* Suppose to use **no\_of\_requests** to group the costumers

Step1: In order to choose the appropriate range of requests to determine the Costumers in 3 groups (**LOW, MEDIUM, HIGH),** I have to check the **descriptive statistics** of **no\_of\_requests** columns.

Use command: summary(data-frame$checked-column)



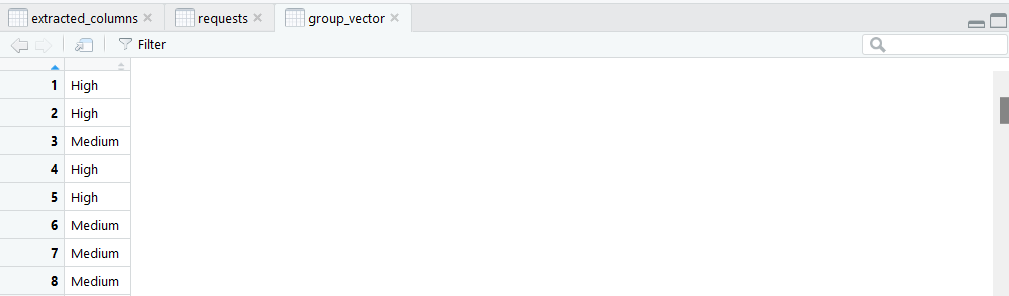
Step2: After having a summary data results of required column, I choose:

* 1st quartile as a **LOW** requests: 2.0
* 3rd quartile as a **HIGH** requests: 3.0

Step3: Group **the costumers** on the quartile percentages

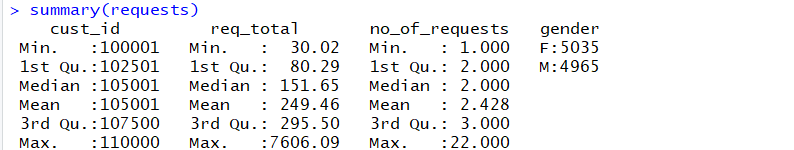


Command to view the **group\_vector: **



* Summary of **requests**

By **summarize** the **requests** data frame, we will explore each column data with **descriptive statistics** like Min, Median, Mean, Max, 1st Quartile and 3rd Quartile.



* Calculate and show total of request value

We use methods **sum** in R to calculate the total of request value in **requests** dataset on **no\_of\_requests** column

Command: total\_requests <- sum(**requests**$**no\_of\_requests**)



View the **total** value:



**PART B.**

* 1. Create a vector **v** contain values from 1 to 50

Use command: 

**Results:** 

* 1. Calculate the total value of all elements in **v**



* 1. List all elements which has value greater than 10 and less than 40

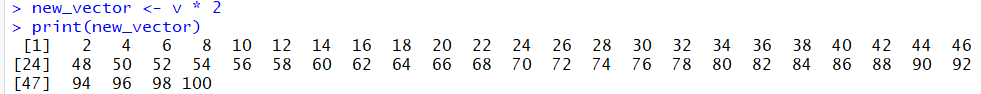
I create a **filtered\_value** vector to store data from 10 to 40



Print the results to check if data is from 11 to 39



* 1. Create a new vector from vector **v**. Element value in the new vector is two times of element value in **v**

****

* 1. Write the code to generate data (**with norm=60, mean=0, sd=0.6 )** and show this data like the figure
* Create random vector x, y with the data given



**\*Notice:** The command above **rnorm** create a vector of normally distributed random variable.   
How ever the **random value** will be changed if we re-run that command. In order to keep the old value, we have to use the command : **set.seed()**

* Create scatterplot which **name** is **“Scatterplot of X and Y”, x-label is “x” , y-label is “y”,**and **x and y** is 2 vector contains data that you want to draw

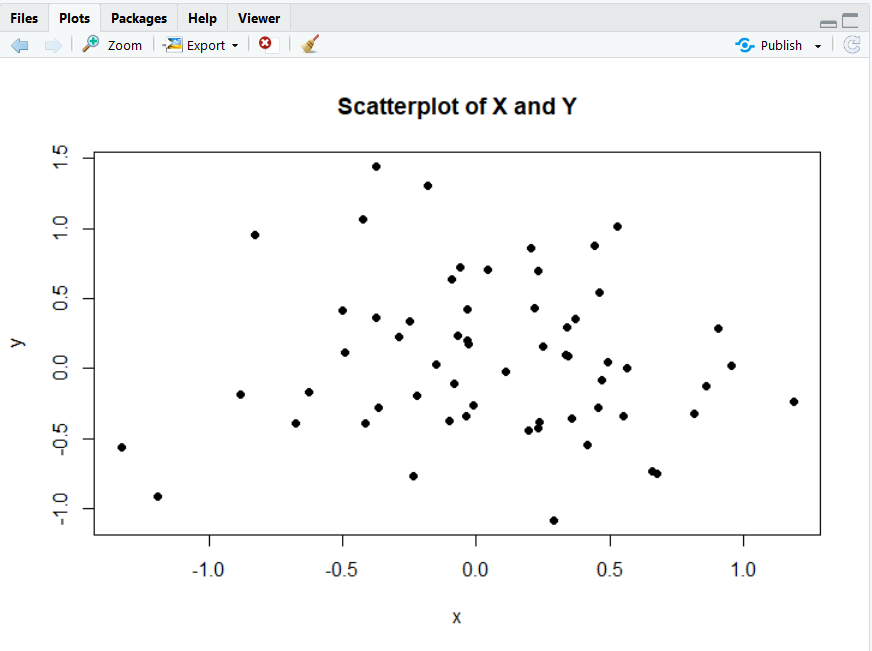
****

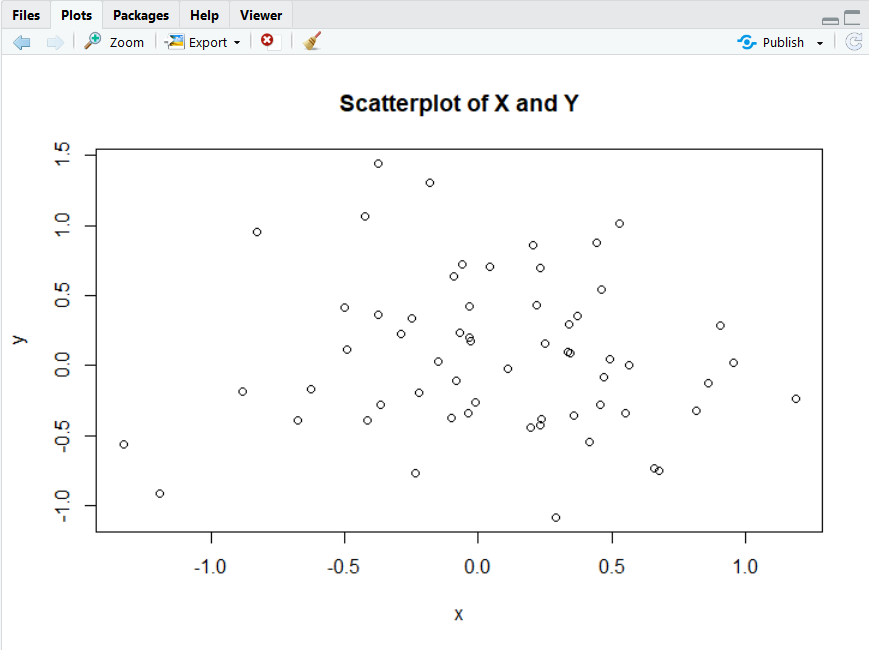
**\*Notice:** With the parameter **pch** is short for **plotting character**

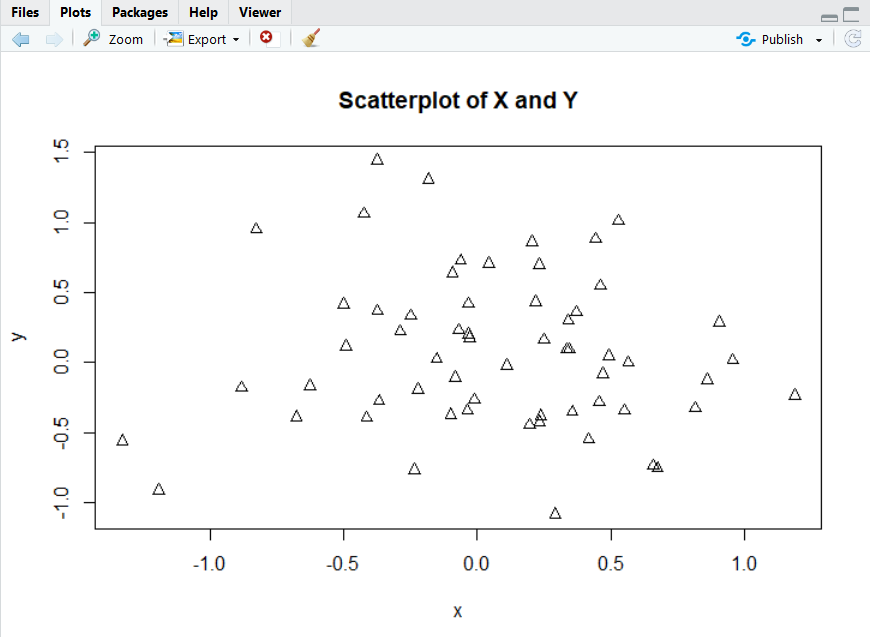
**- pch** is a parameter that use symbol type to specify the **data point** in which kind of shapes. And some common values for **pch** is:

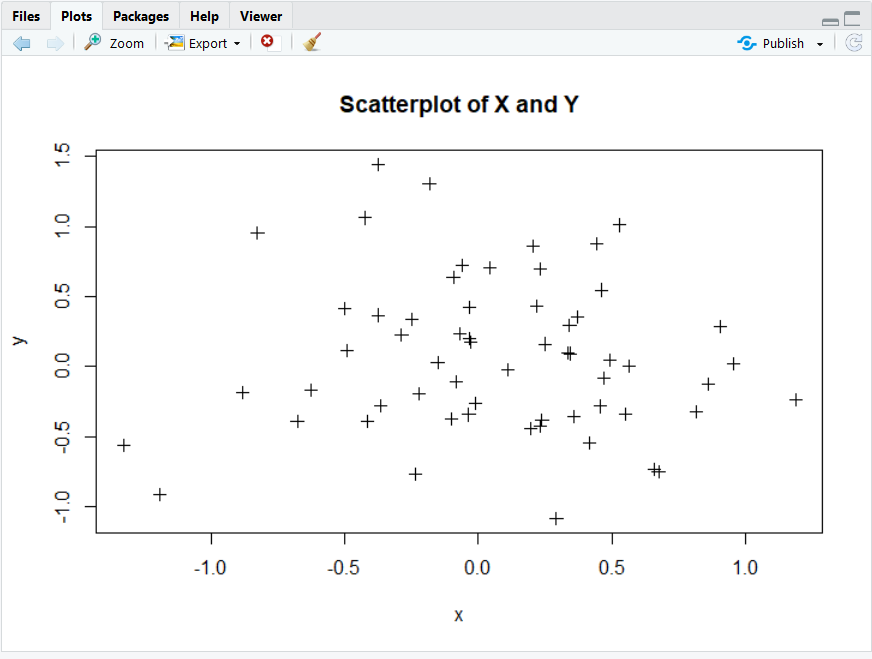
* pch = 16: Filled circles
* pch = 1: Hollow circles
* pch = 2: hollow triangles
* pch = 3: plus signs

Results for those **4 symbol types:**

****

****

****

****