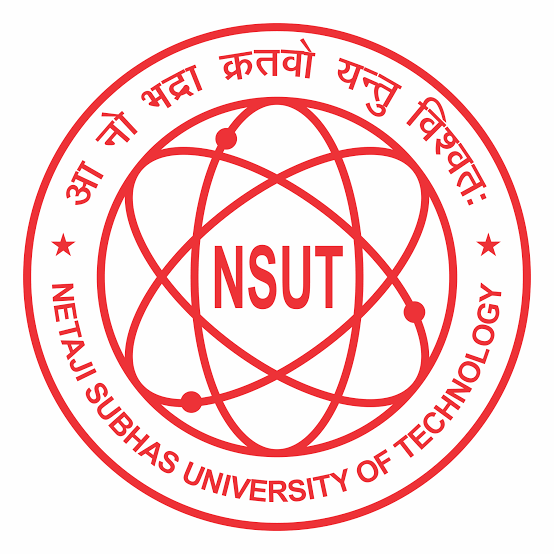
**COMPUTER HARDWARE SOFTWARE WORKSHOP ( COCSC19 )**

**Title: R programming and data visualisation**



Submitted by :

Amar Kumar Tarei: 2021UCS1614

Shreyansh Agrawal: 2021UCS1612

Ayushi Singh: 2021UCS1616

**Task:** Use any R simulator (R studio, Google colab, etc.)  
Perform following task  
  
1. Explain Basic Data Structure in R.  
2. Implement Linear Regression in R and Visualize the results.  
3. Implement Logistic Regression in R and Visualize the results.  
4. Implement any Machine learning Algorithm along with feature selection and data visualization on any dataset of your choice.

**PART 1 – DATA STRUCTURES IN R**

In R, a programming language and environment for statistical computing and graphics, there are several basic data structures that are commonly used. These data structures include vectors, matrices, arrays, lists, data frames, and factors. Let's briefly explain each of these:

**Vectors:**

* A vector is a basic data structure in R and can be thought of as a one-dimensional array.
* It can hold elements of the same data type (numeric, character, logical, etc.).
* Vectors can be created using the **c()** function.

# Example of creating a numeric vector

numeric\_vector <- c (1, 2, 3, 4, 5)

# Example of creating a character vector

character\_vector <- c ("apple", "banana", "orange")

**Matrices:**

* A matrix is a two-dimensional data structure where elements are arranged in rows and columns.
* All elements in a matrix must be of the same data type.
* Matrices can be created using the **matrix ()** function.

# Example of creating a matrix

matrix\_example <- matrix(1:6, nrow = 2, ncol = 3)

**Arrays:**

* An array is a multi-dimensional extension of a matrix. It can have more than two dimensions.
* Arrays can be created using the **array()** function.

# Example of creating a 3-dimensional array

array\_example <- array(1:24, dim = c(2, 3, 4))

**Lists:**

* A list is a versatile data structure in R that can hold elements of different data types.
* Each element in a list can be a vector, matrix, array, or even another list.
* Lists can be created using the **list()** function.

# Example of creating a list

list\_example <- list(numeric\_vector, matrix\_example, character\_vector)

**Data Frames:**

* A data frame is a two-dimensional tabular data structure, similar to a matrix, but with additional features.
* Columns in a data frame can have different data types.
* Data frames can be created using the **data.frame()** function.

# Example of creating a data frame

data\_frame\_example <- data.frame(

Name = c("Alice", "Bob", "Charlie"),

Age = c(25, 30, 22),

Score = c(90, 85, 95)

)

**Factors:**

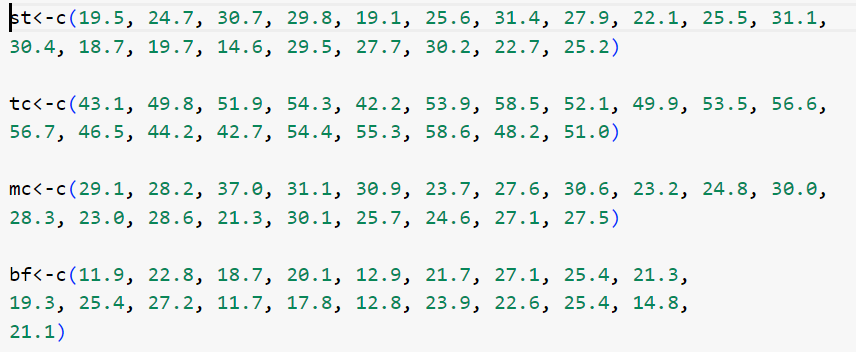
* Factors are used to represent categorical data in R.
* They are created using the **factor()** function.

# Example of creating a factor

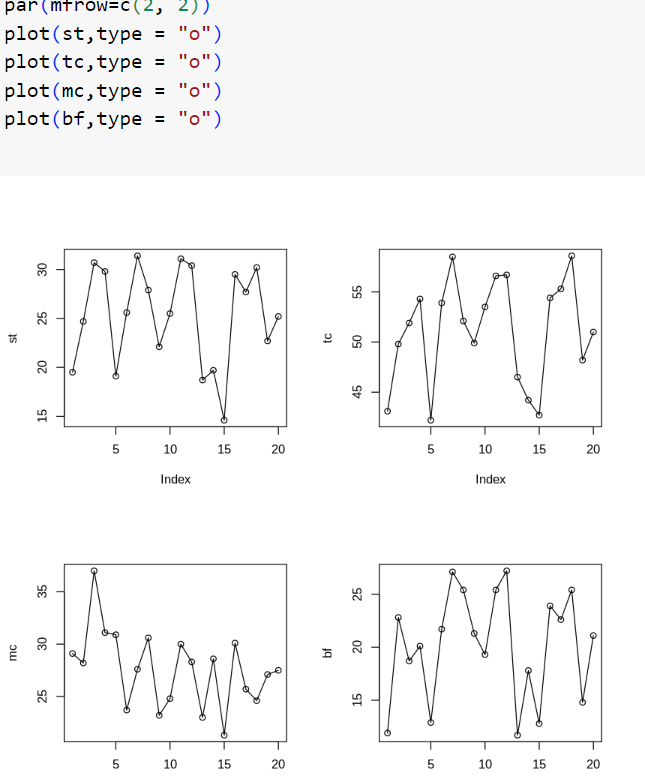
gender <- factor(c("Male", "Female", "Male", "Female"))

**PART 2 – LINEAR REGRESSION**

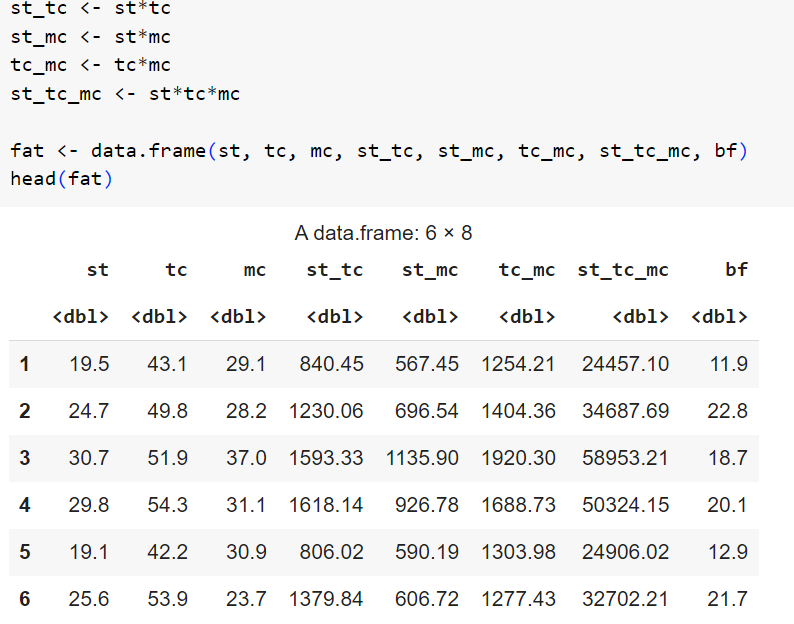
Consider the data set below, which contains data about various **body measurements, as well as body fat**. The goal is to check whether the independent variables **Skinfold Thickness (ST), Thigh Circumference (TC), and Midarm Circumference (MC)** predict the independent variable **Body Fat (BF)**

****

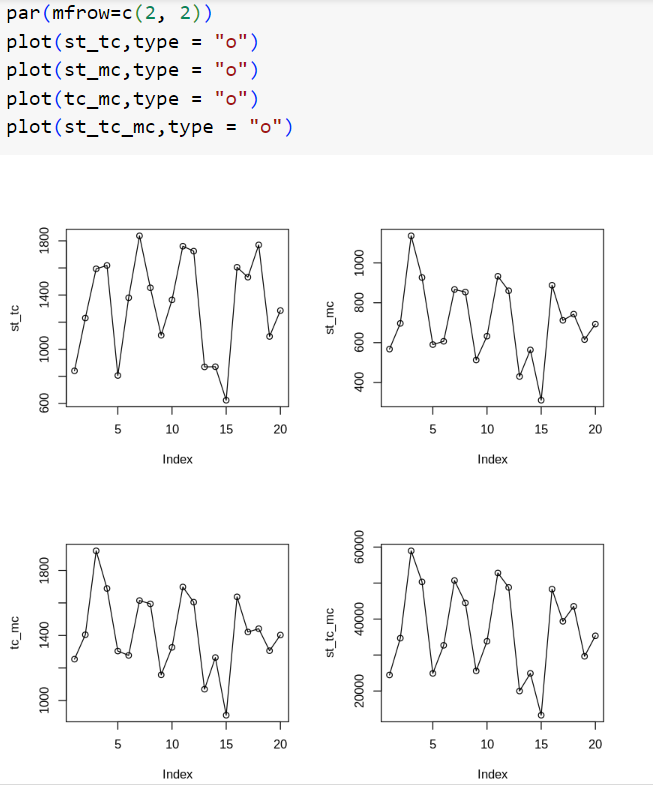
See the varied distribution of skinfold thickness, thigh circumference, midarm circumferenceand bodyfat.

****

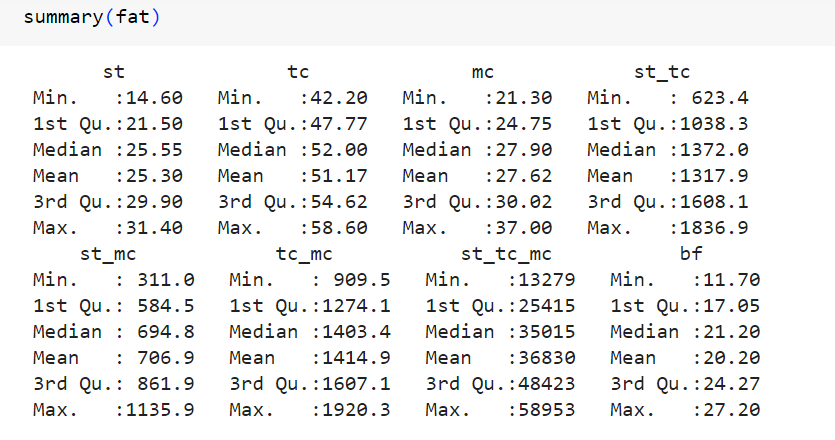
**Using interacting terms and independent terms as features as follows :-**

****

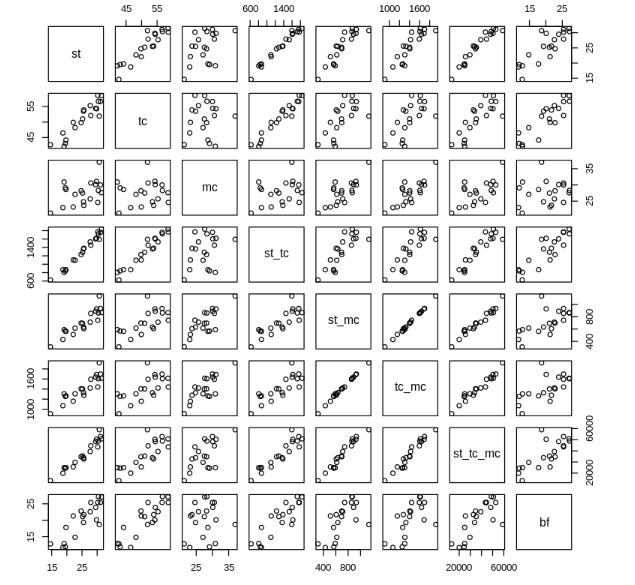
**Graphical visualization of the same is :-**

****

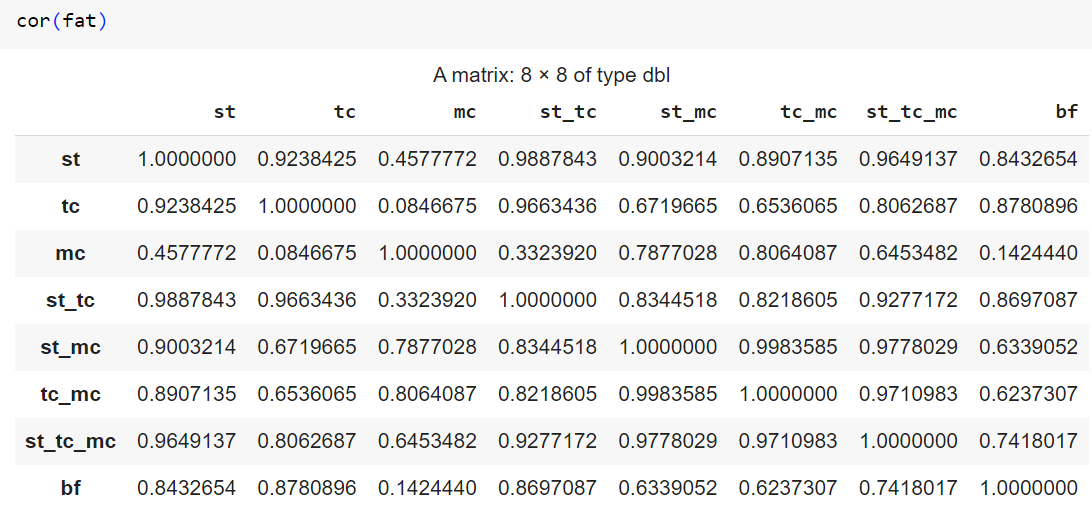
Some mathematical calculations for the data are: -

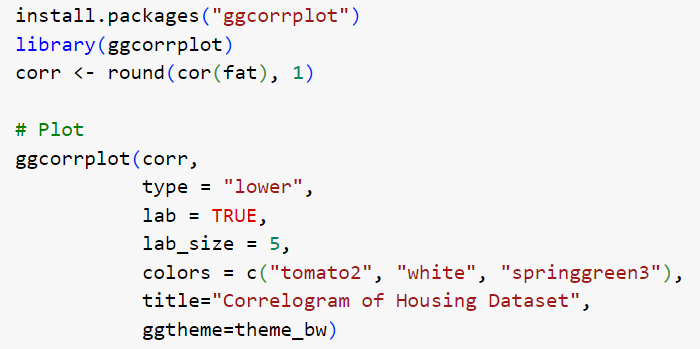


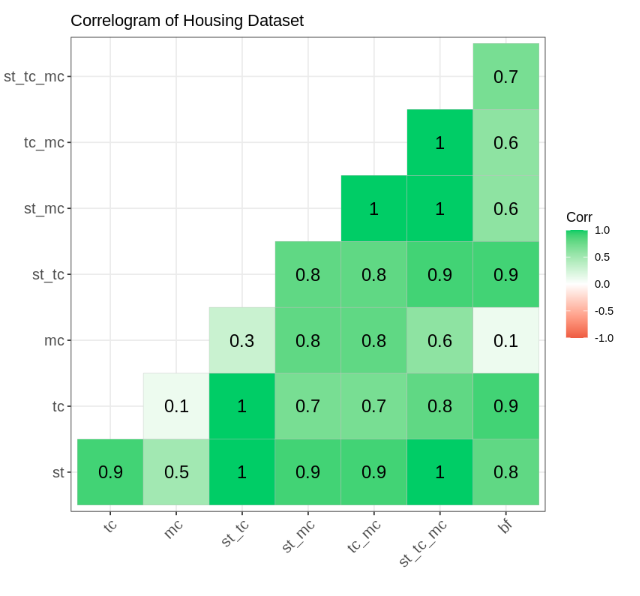
Scatter plot matrix of the data :-

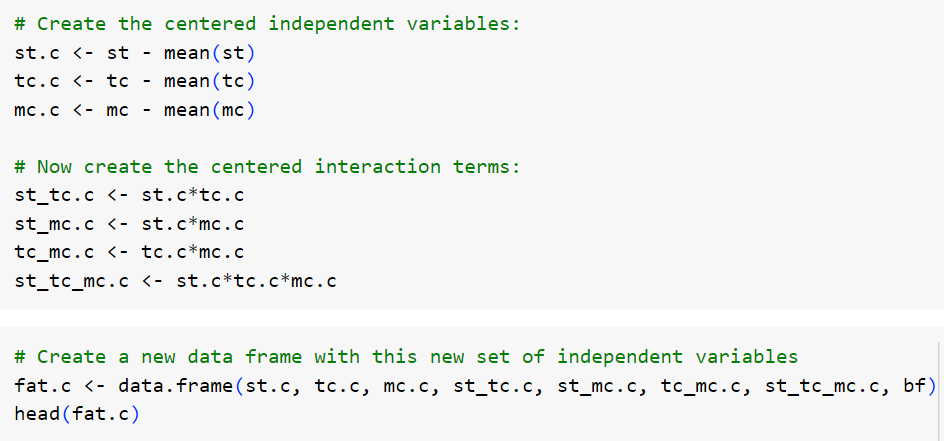


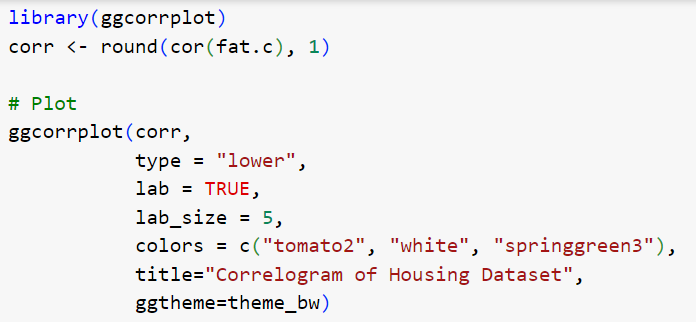
Correlation between numerical :-

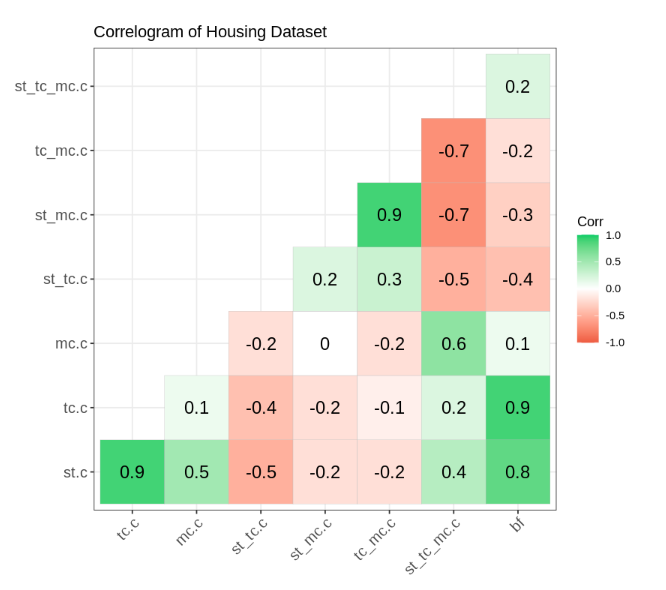


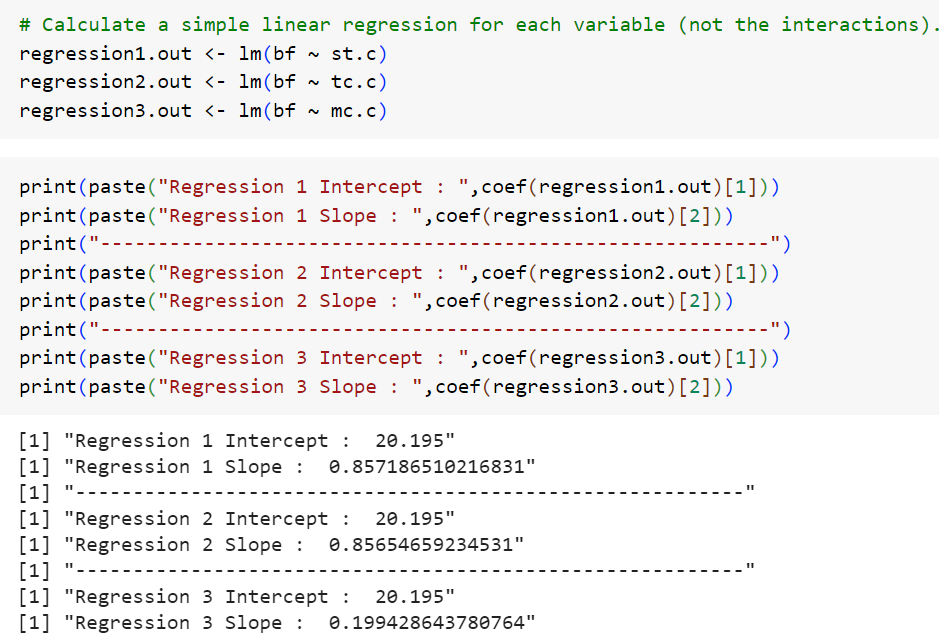


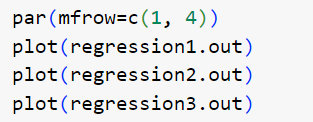


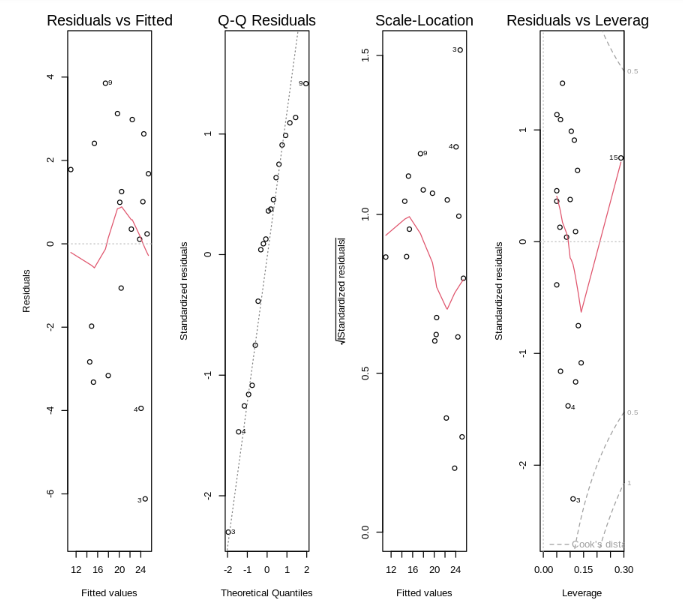


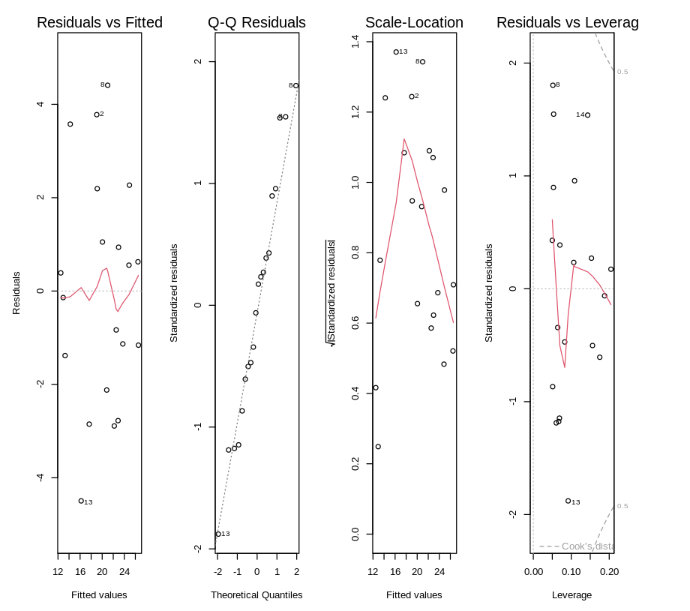


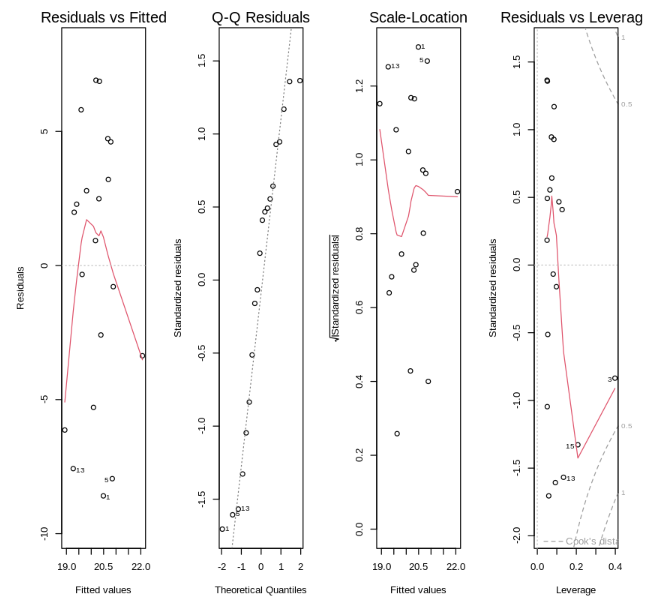


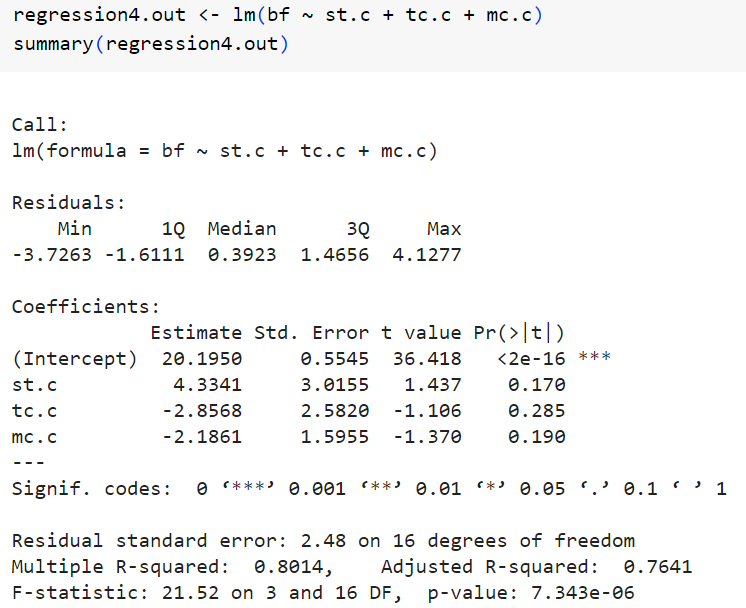


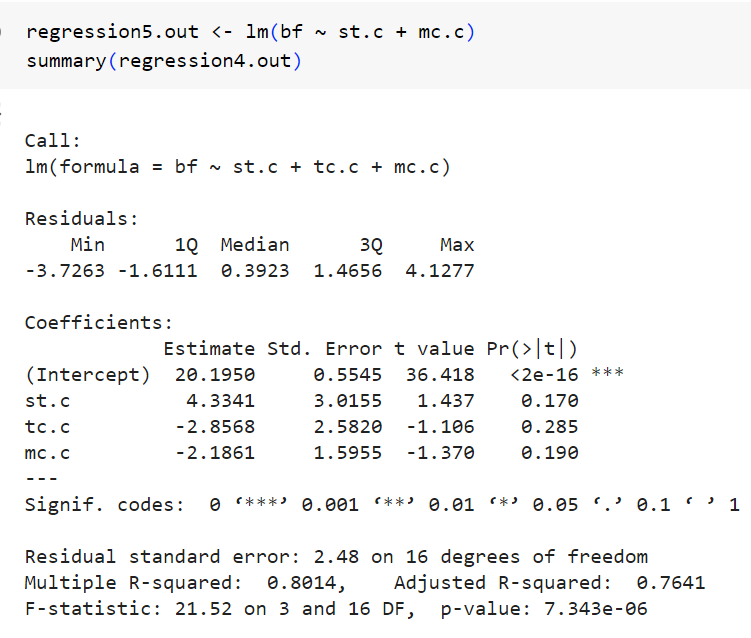


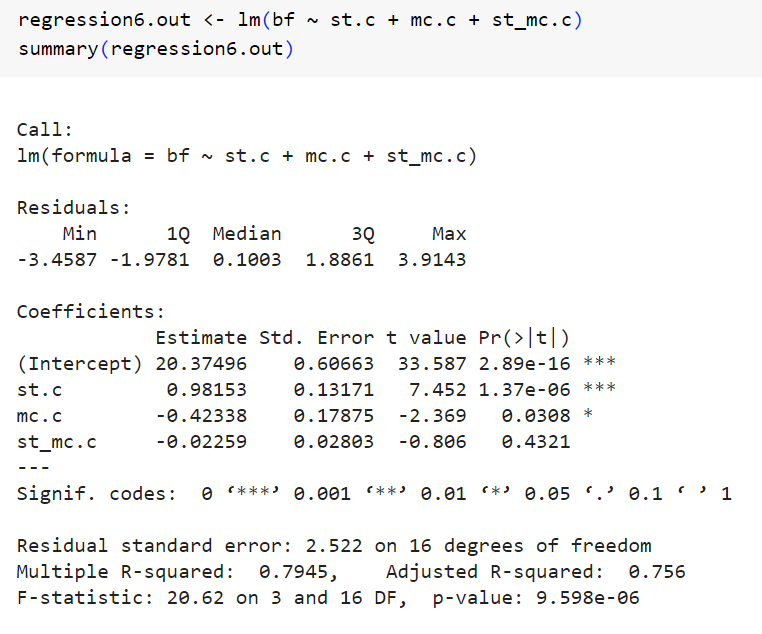


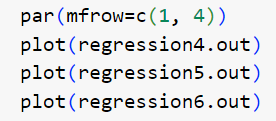


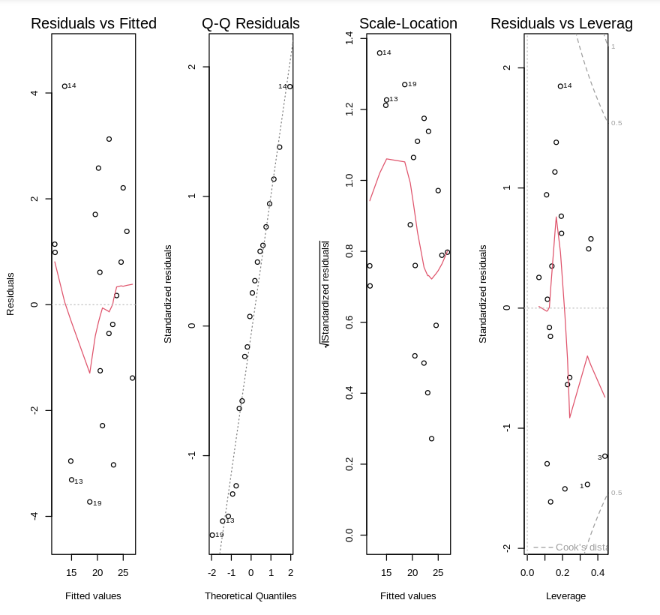


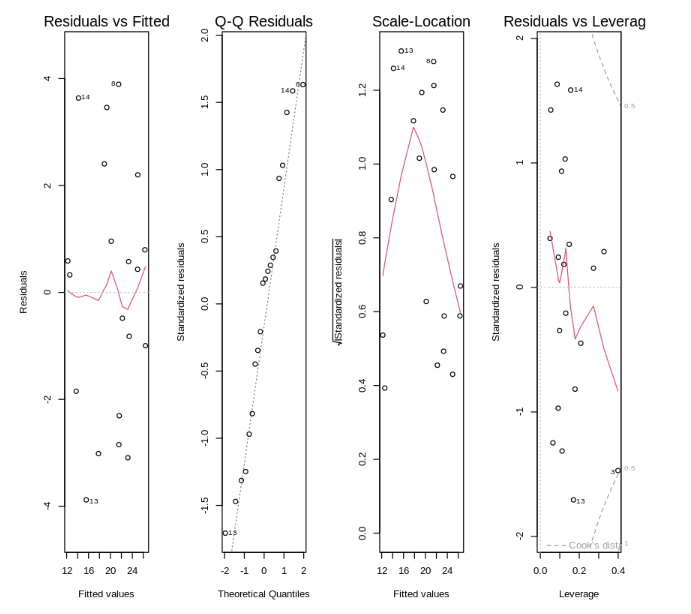


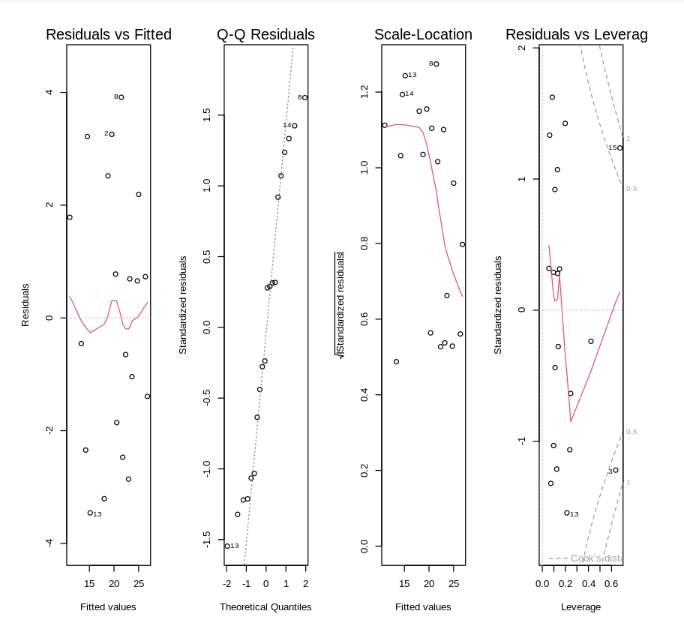


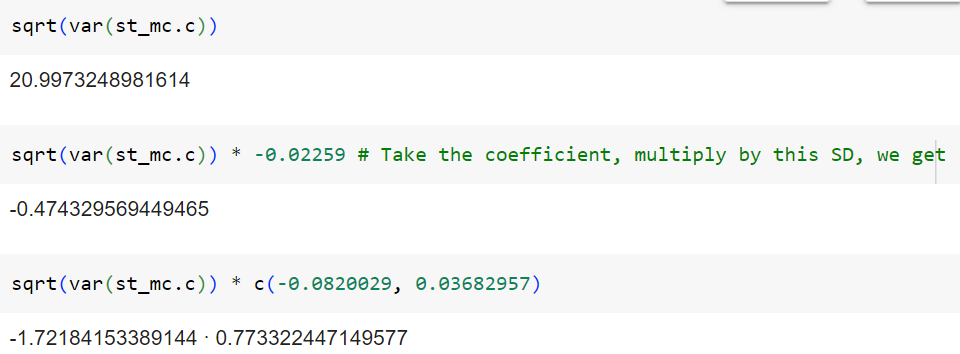








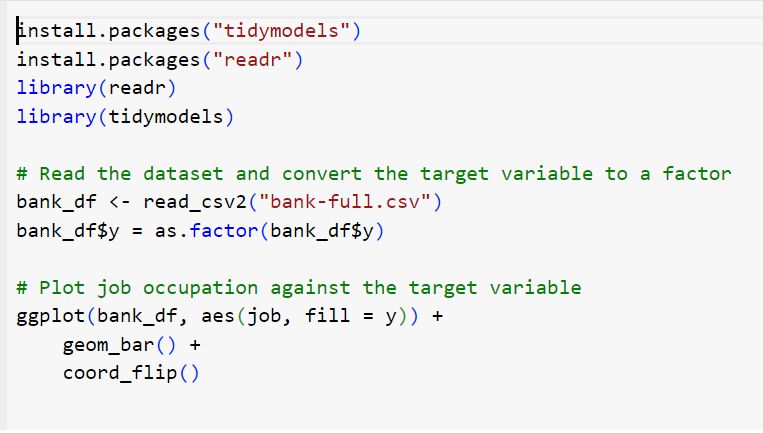


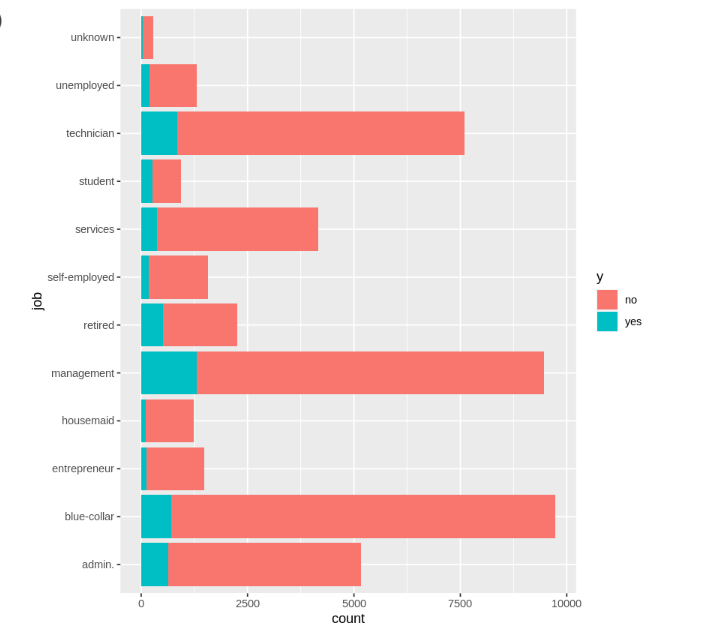


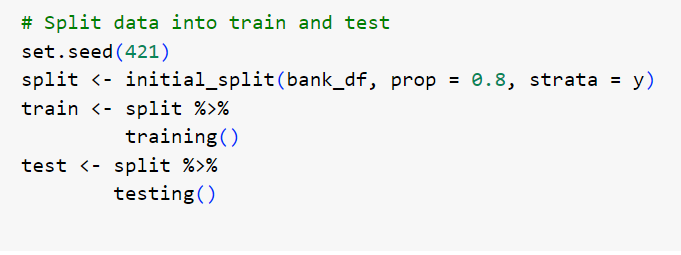
**PART 3 – LOGISTIC REGRESSION ON BANK DATA**

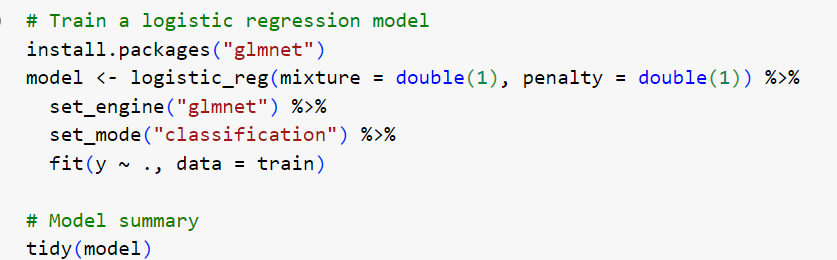
We would use a [**direct marketing campaign dataset**](https://archive.ics.uci.edu/ml/datasets/bank+marketing) by a Portuguese banking institution using phone calls. The campaign aims to sell subscriptions of a bank term deposit represented by the variable y (subscription or no subscription). The objective of the logistic regression model is to predict whether a customer would buy a subscription or not based on the predictor variables, aka attributes of the customer, such as demographic information.

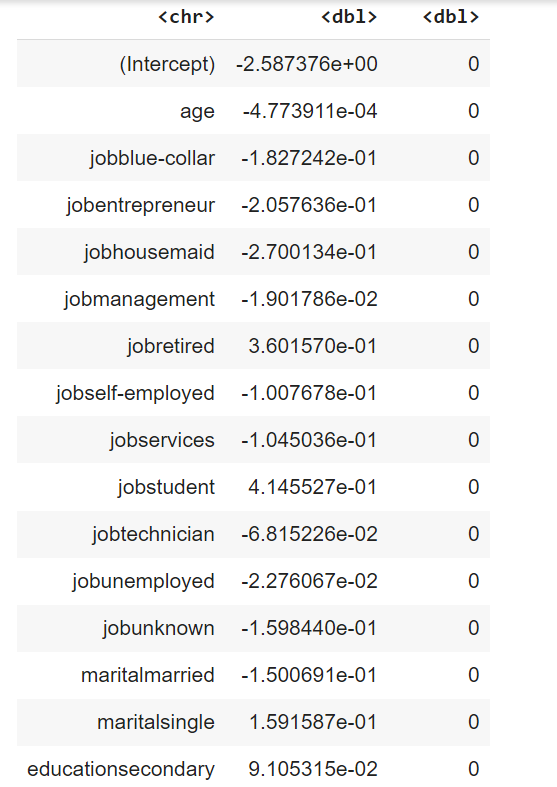
* Splitting the data into training and testing sets
* Fitting (training) the model
* Making predictions
* Evaluating the model performance
* Hyperparameter tuning

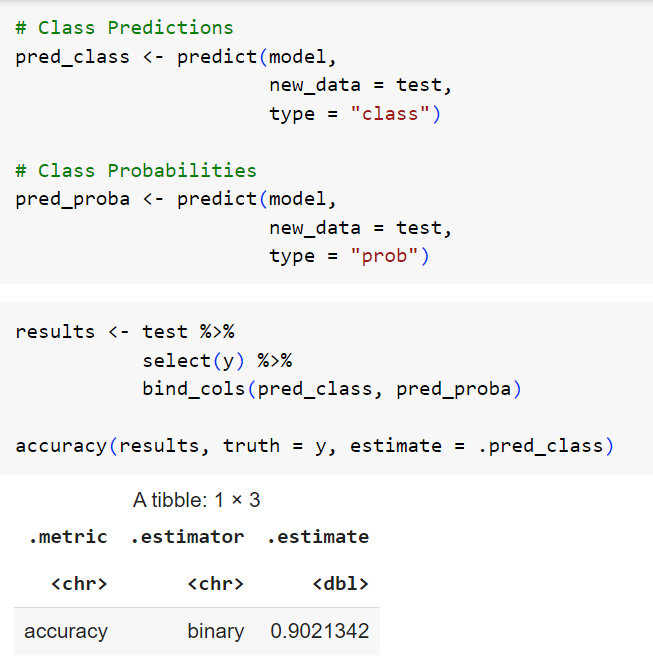


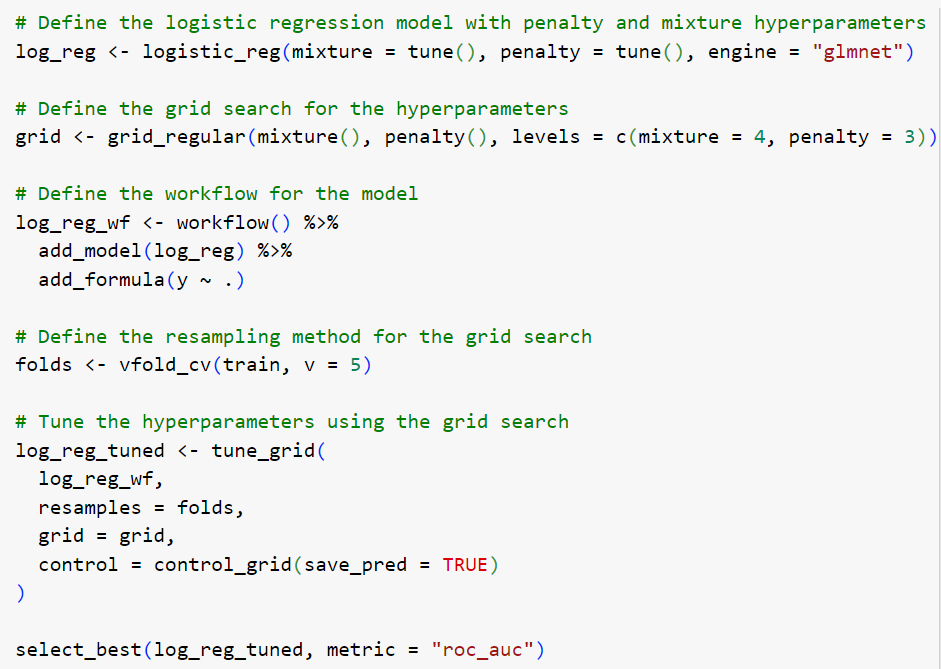


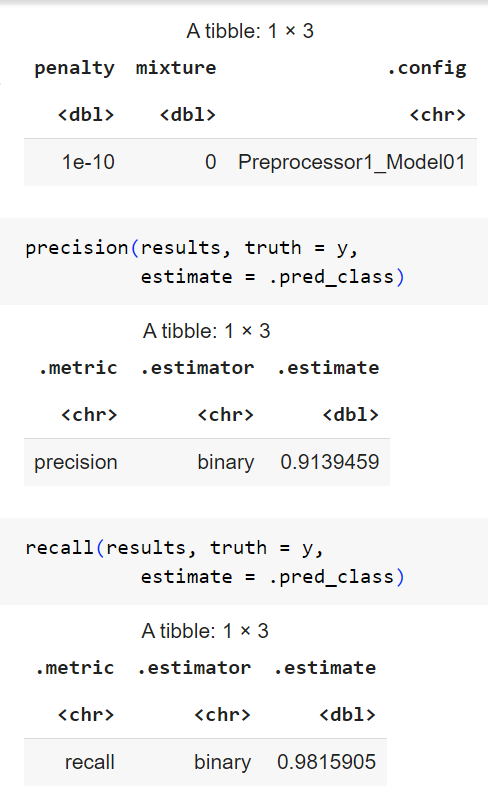


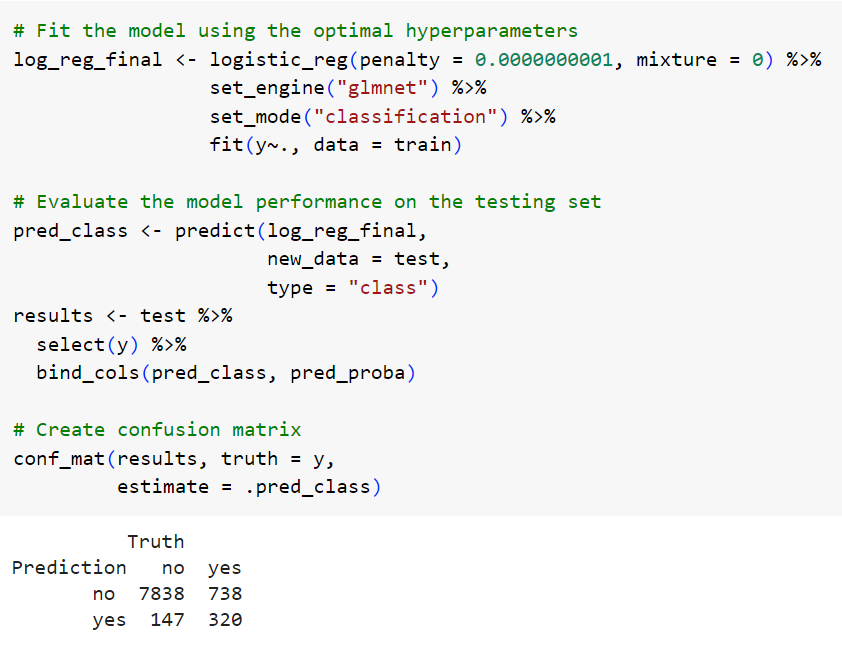


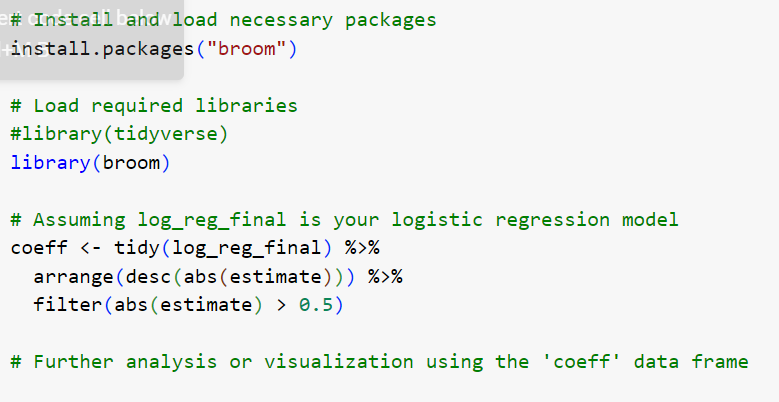


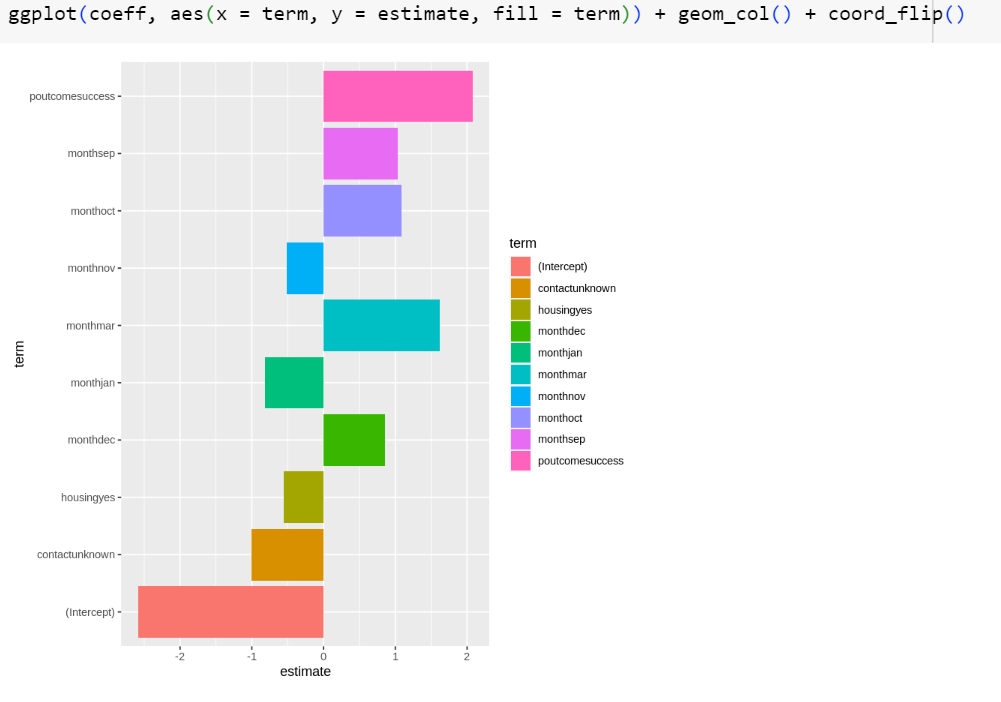


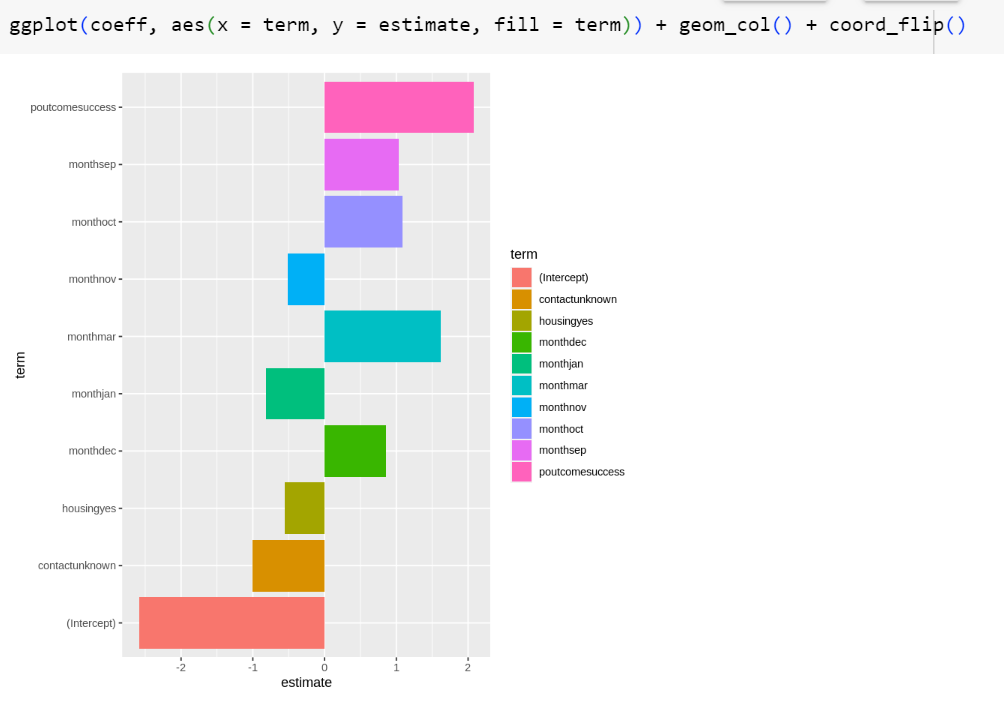




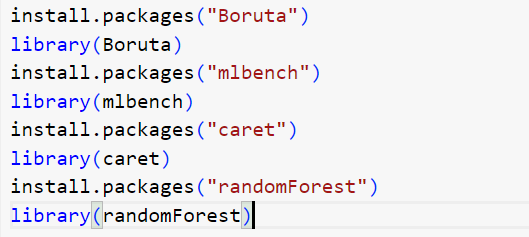


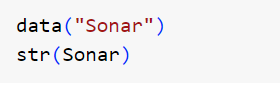


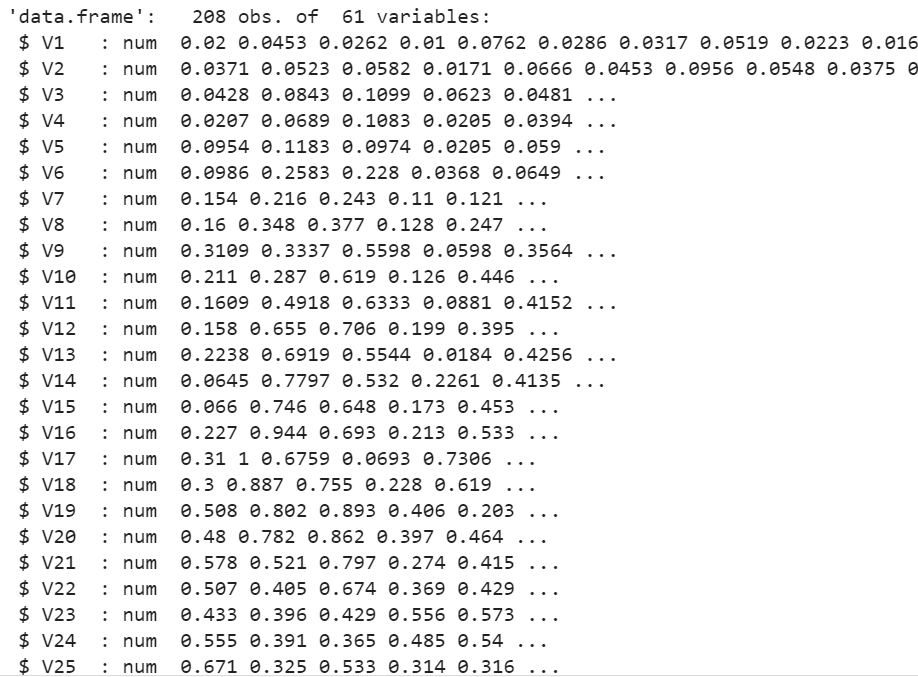


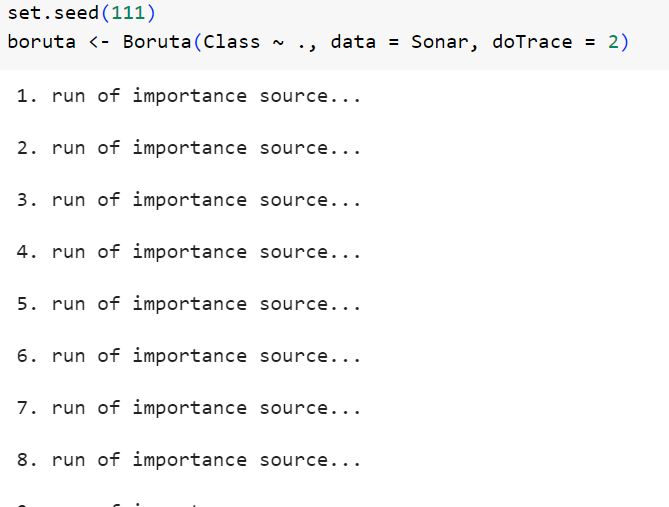


**PART 4 – FEATURE SELECTION AND RANDOM FOREST**

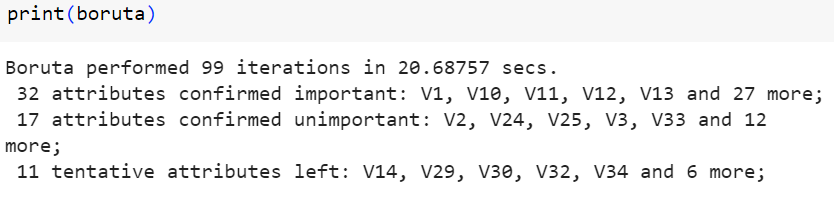


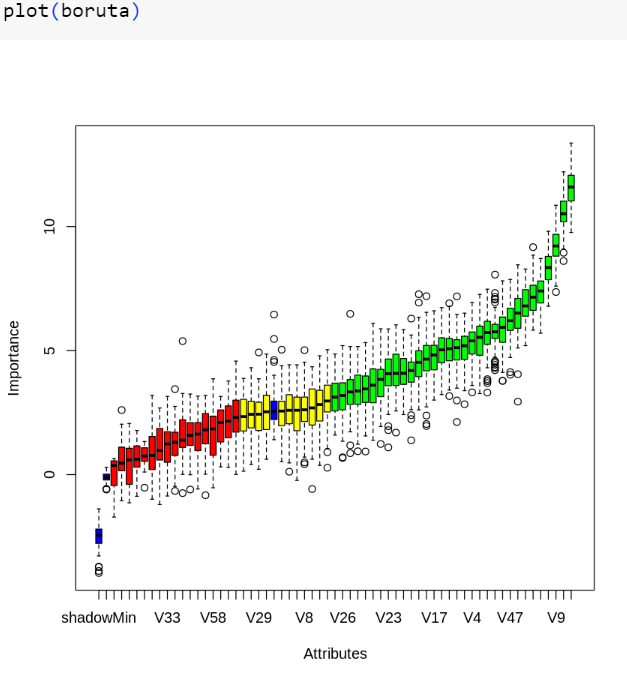


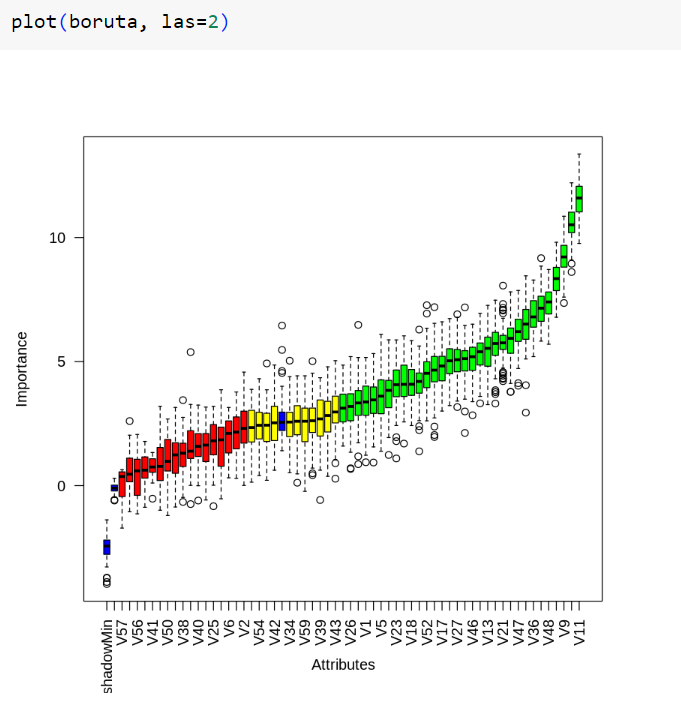


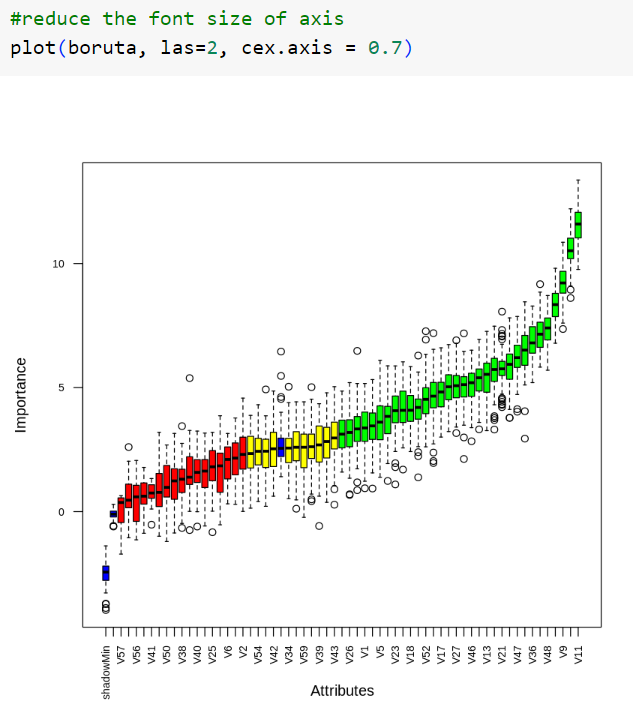


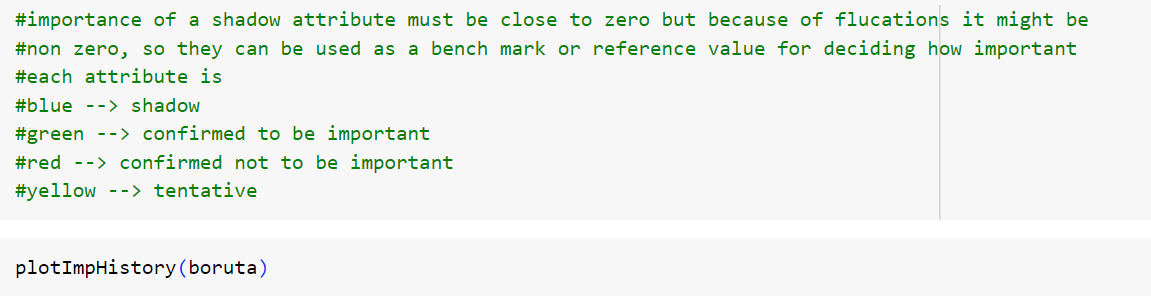
After many iterations it will come to stop listing out important features.

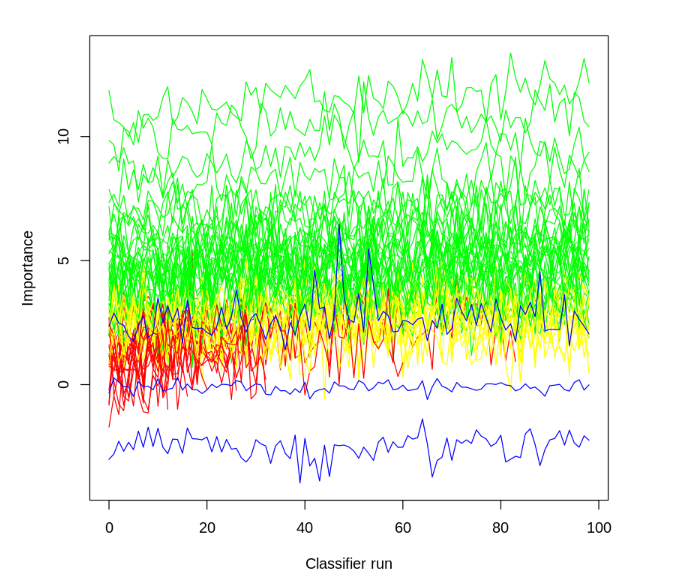


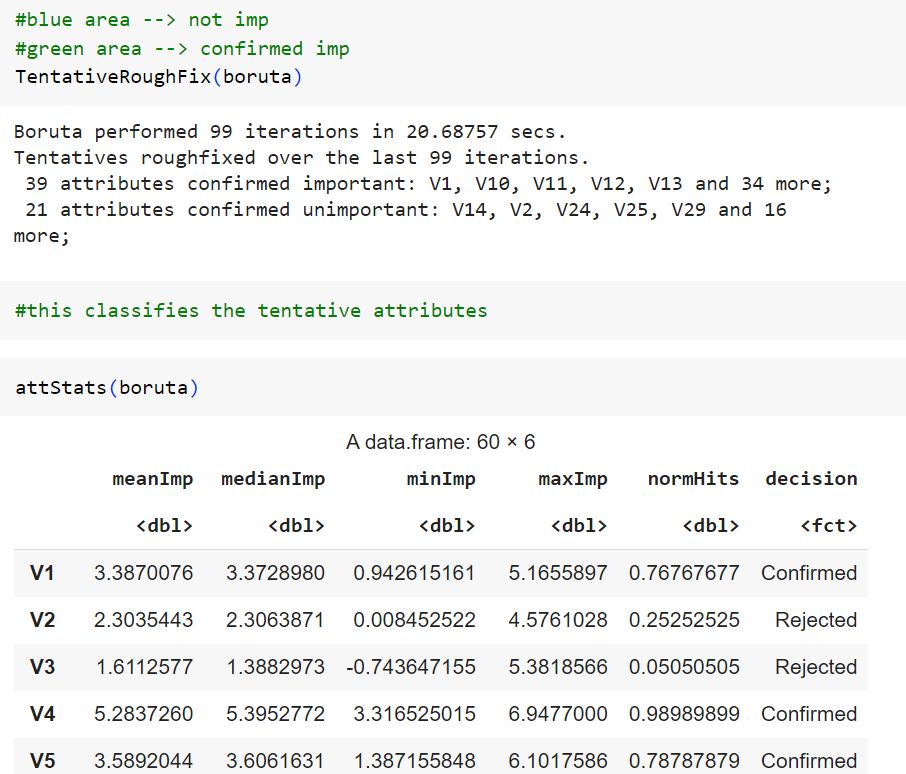




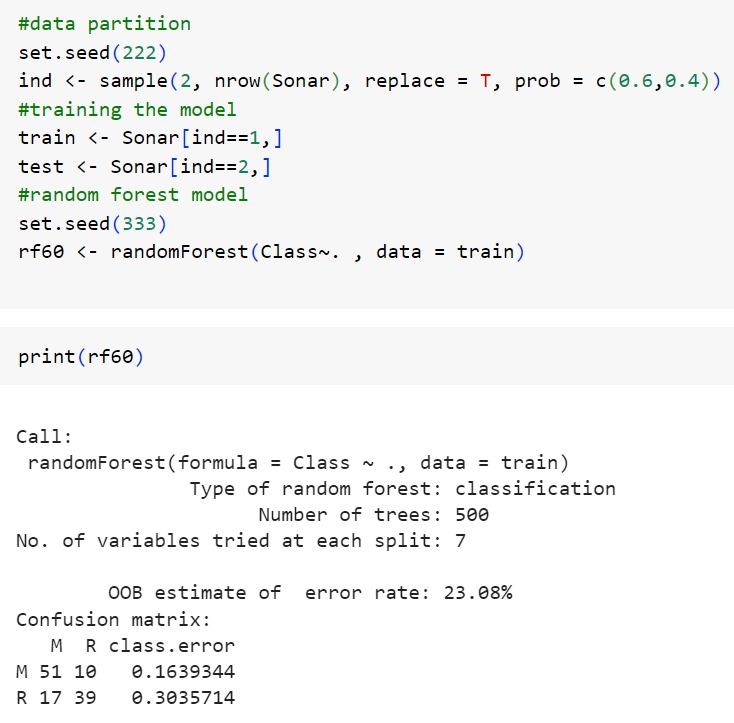


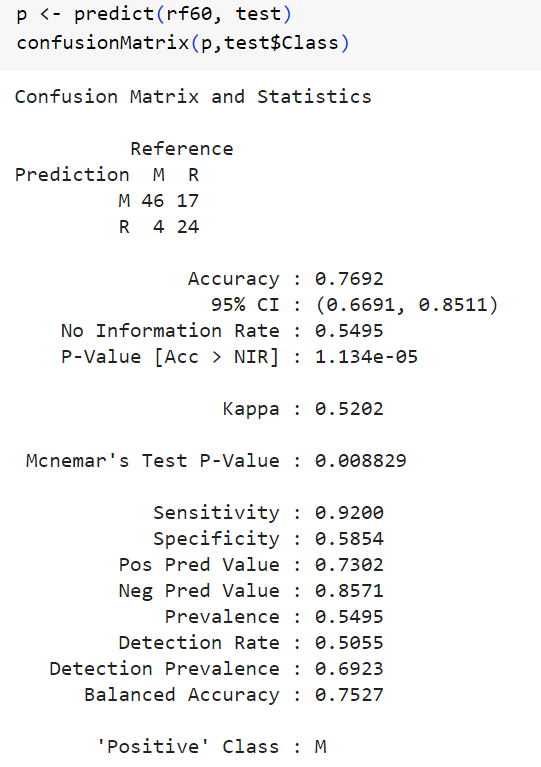






Applying random forest on solar data so got





\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_