Glass classification project report

Data science

Name std

Shareef Tarek ragab 20160010

Name teacher

Eng. Bushra Alqarout

**Introduction** The motivation behind this project was to learn more about the quick application of ML algorithms by experimenting with various classification models and the tuning parameters associated with each model. Caret is a powerful R library which is used for Classification and Regression Training, and I wanted to get familiar with this package and different classification algorithms.

**Data** This dataset consists of the:

* Refractive Index
* Measures of various elements in the glass
* The Glass Type (1-7) e.g. *for headlamps, containers, building windows, vehicle windows etc. \*\** (Target Variable) \*\*

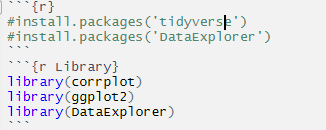
**Aim**

* The aim was to identify patterns between variables in the dataset through Principle Component Analysis (PCA)
* Apply and evaluate different classification models and see how each one performs based on the 'Accuracy' and 'Kappa' metrics

*I expected the Random Forest model to perform the best, so just felt like exploring different parameter tuning methods (random Search through caret)*

* Call the libraries

We called the following libraries as shown in the picture



To perform operations, graphics display data resulting from the following commands

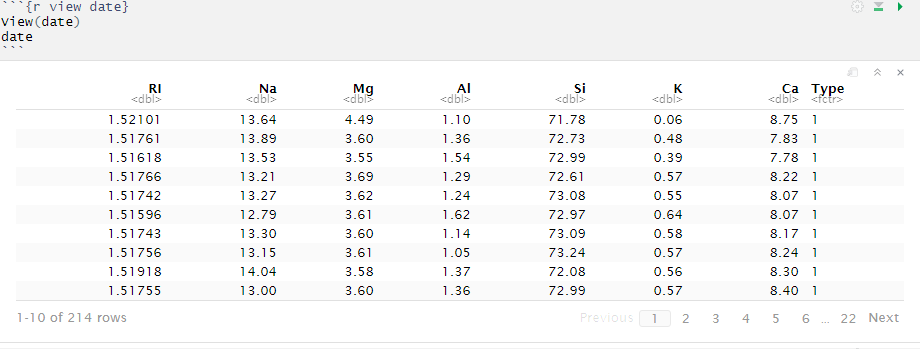
* Initialize of data frame



I booted the date value with the data from the CSV file reader and the reader called the file containing the data and gave the header true value

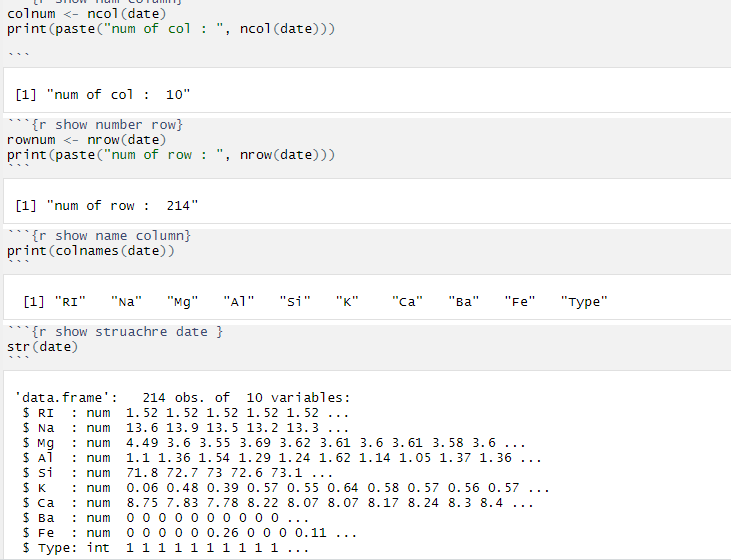
File: glass.csv

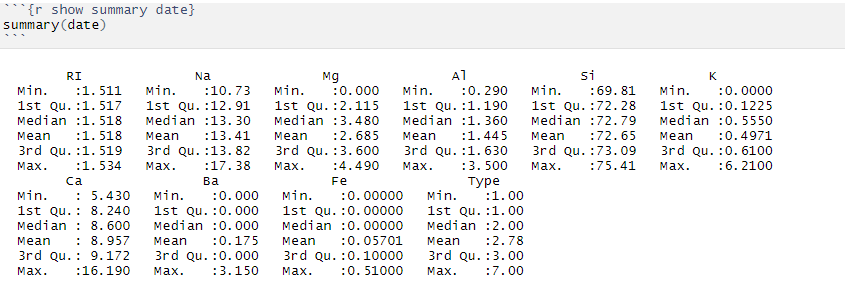
* Display data



We display the data by calling the value in which the data was initialized and containing the data

* View details





1. It has been implemented that displays how many columns the table contains

And the result of the number of columns as shown in the picture is equal 8

1. It has been implemented that displays how many rows the table contains

The result was that the number of rows as shown on the image was equal 214

1. It has been implemented that displays how many name columns the table contains

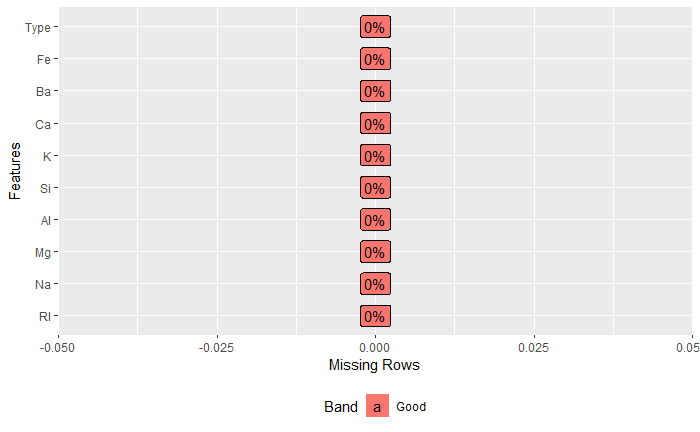
And the result of the name of columns as shown in the picture is equal

"RI" "Na" "Mg" "Al" "Si" "K" "Ca" "Type"

The mayors' names refer to the following:

* + - * 1. RI: refractive index
        2. Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)
        3. Mg: Magnesium
        4. Al: Aluminum
        5. Si: Silicon
        6. K: Potassium
        7. Ca: Calcium
        8. Ba: Barium
        9. Fe: Iron
        10. Type of glass: (class attribute)
* building\_windows\_float\_processed (1)
* building\_windows\_non\_float\_processed (2)
* vehicle\_windows\_float\_processed (3)
* vehicle\_windows\_non\_float\_processed (none in this database) (4)
* containers (5)
* tableware (6)
* headlamps (7)

1. A function has been called that shows the structure and elements of the existing table and the types of values that the table contains from the elements and properties.
2. The function has been called which summarizes the structure and elements of the existing table and the types of values that the table contains from the elements and properties from which the elements in it refer to the following:
   1. Min: Symbolizes the lowest value in a column
   2. 1st Qu: Symbolizes the first value in a column
   3. Median: the median value of a range of values.
   4. Mean: Refers to the mean value (average) in the column
   5. 3rd Qu.: It denotes the third value in the column
   6. Max.: Symbolizes the largest value in a column
3. Show missing values

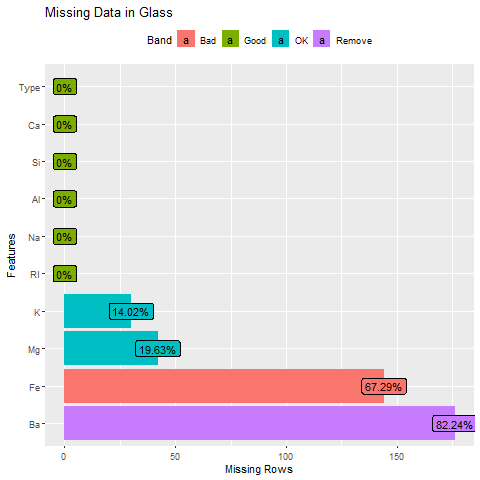


Where the following values appear when the dataExplorer library's plot\_missing () function is called and the date scale is passed to it where the data was initialized

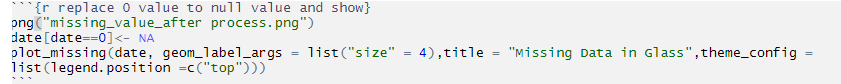
.

Resulting values: Zeros contain an empty value, so no null values appear in the table.

It becomes plot\_missing(date)



It is caused by this code

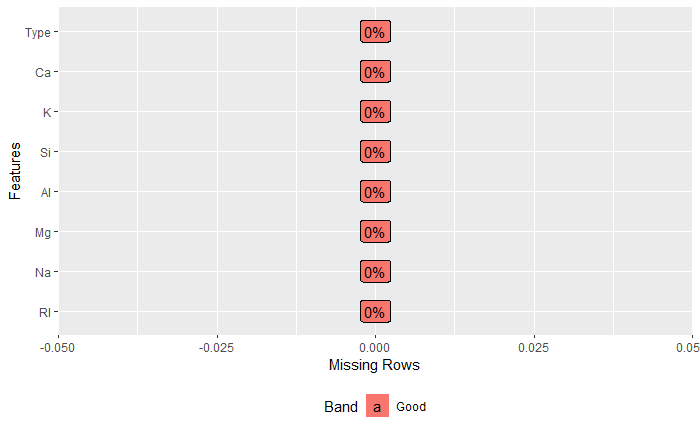


Where all zero-valued values are converted to null values and the function of displaying empty values is called back.

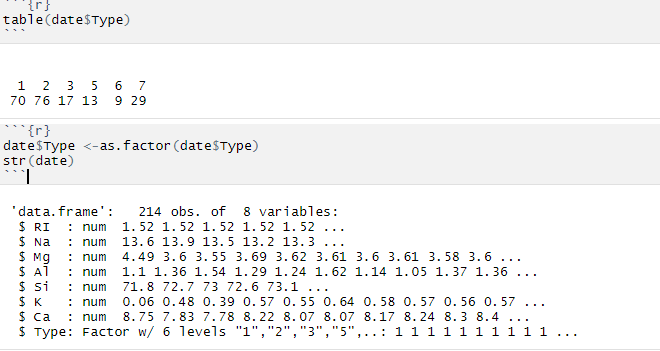


The actions suggested by the drawing were taken and the two columns were deleted (Fe, Ba)

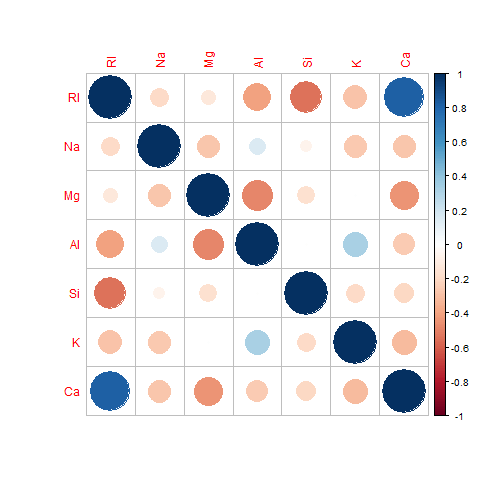
The null values are returned and converted to zeros again to complete the operations on the data



1. Convert values

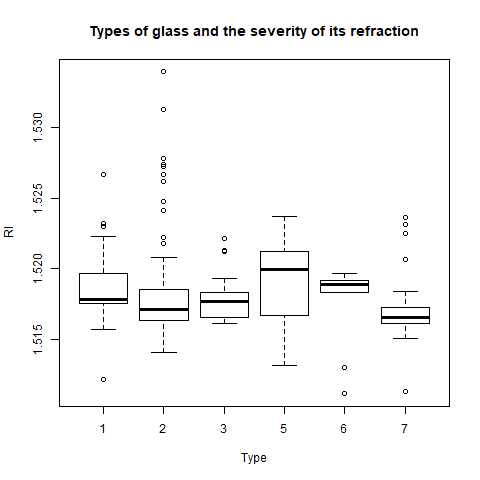


Based on the description of the data, the Type property is a value on which the glass is classified, so it must be converted from a numeric value to Factor.

1. Relationships between the elements: 

Relationships between items in date frame without Type is a property

Where I calculated the relationships of the date frame without the type property via the cor () function and the date [, - 8] parameter was passed to it and passed the result to the corplot () function directly to it to draw the relationships of the elements and this function is found in the corplot library.

1. The relationship between the type of glass and RI:s and refraction:

Explain the relationship between the types of glass and the intensity of ginksar for each of these types.

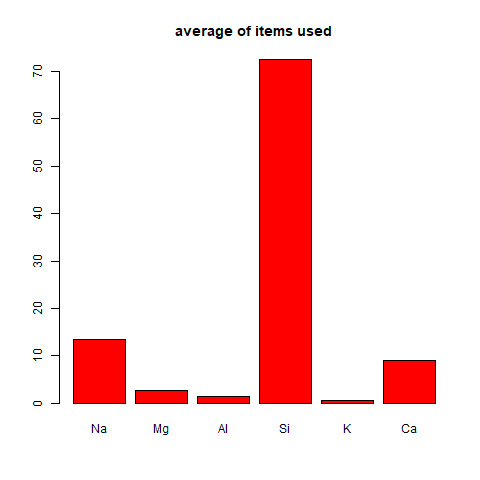
As the glass type contains a number of its constituent elements, the illustration shows that some elements are outside a frame.

And this was done through this matter.

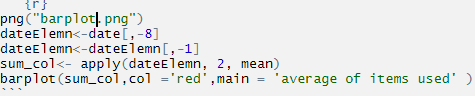


Where I chose the x-coordinate as the type and y-events as the refractive intensity from date frame.

1. average of items used

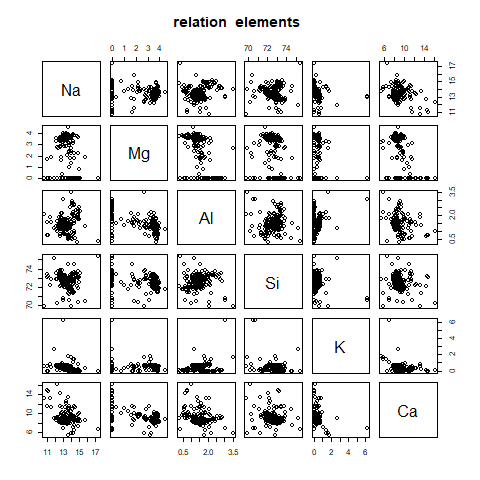


Here a chart is drawn showing the arithmetic mean percentage of the elements present in all samples taken.



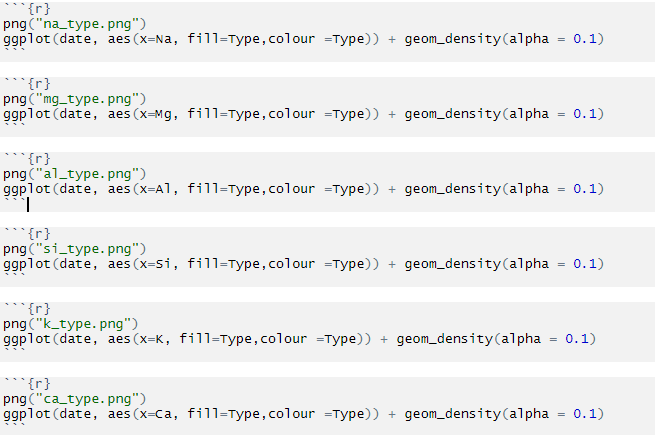
Here we removed the first and last columns of the dataframe and applied the intermediate function to them by supply and passed the output to the barplot () function.

1. The relationship of the elements to each other

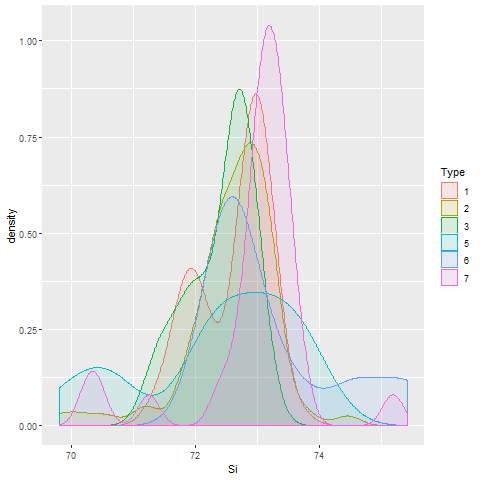
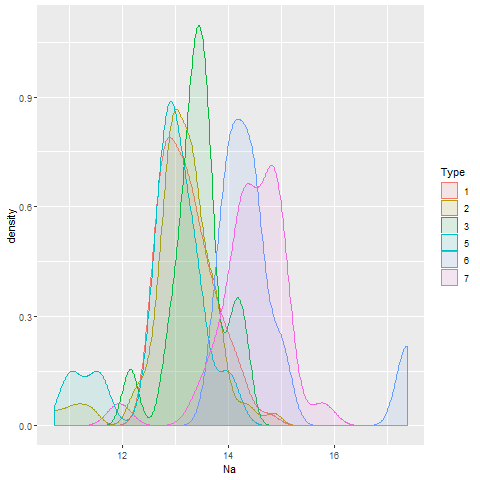
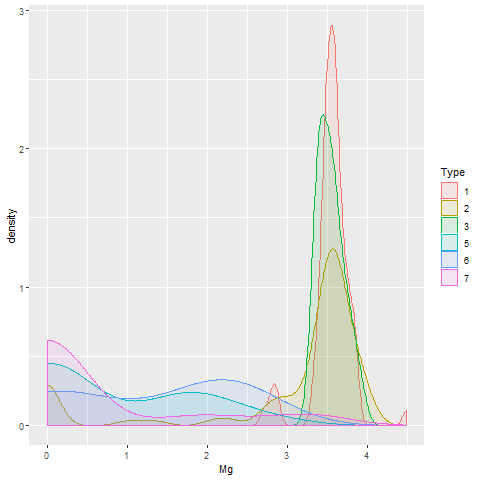
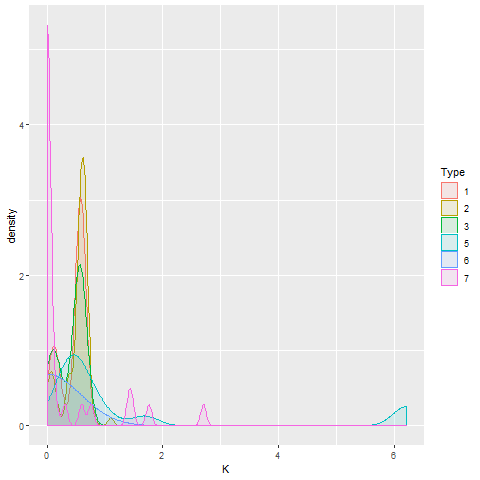
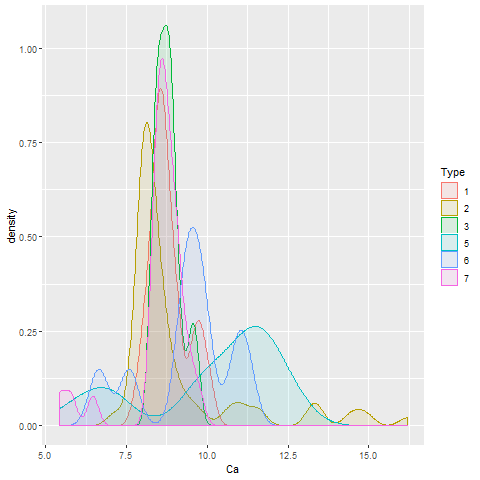
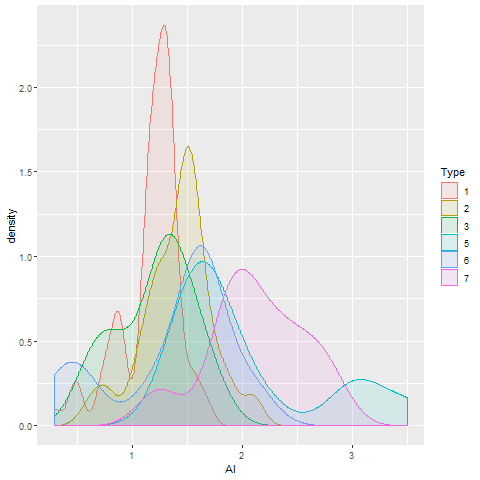


Here, the chemical elements, some of them in the glass industry and the role they play in it, are clarified.

plot (dateElemn, main ="relation elements “)

1. The relationship of the Element with the typeelement to the type

Here the ggplot2 library was called drawing the element and its relationship to the glass manufacturing types



1. Reflection grading classification

I have classified the degrees of reflection by the operations of cluster and we have classified them to 7 degrees according to the existing types of glass, where each element of the types of glass will get a classification according to the severity of its reflection.

And clustering is by the following code:

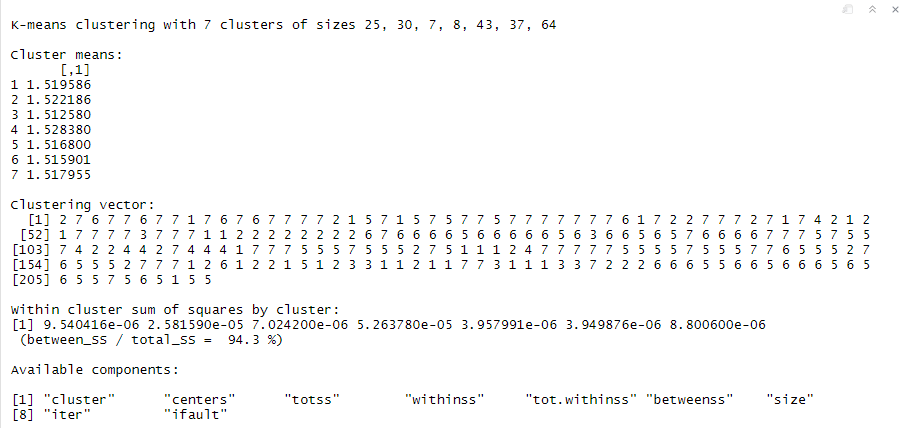


This was done via the kmeans function of clustering

K-means clustering is a simple unsupervised learning algorithm that is used to solve clustering problems. It follows a simple procedure of classifying a given data set into a number of clusters, defined by the letter "k," which is fixed beforehand. The clusters are then positioned as points and all observations or data points are associated with the nearest cluster, computed, adjusted and then the process starts over using the new adjustments until a desired result is reached.

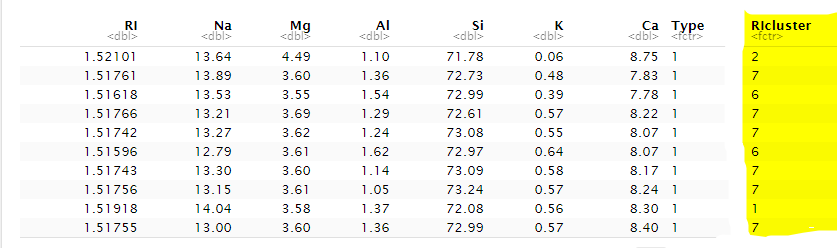
K-means clustering has uses in search engines, market segmentation, statistics and even astronomy.

The output was:



The practical output has been added to dataframe.



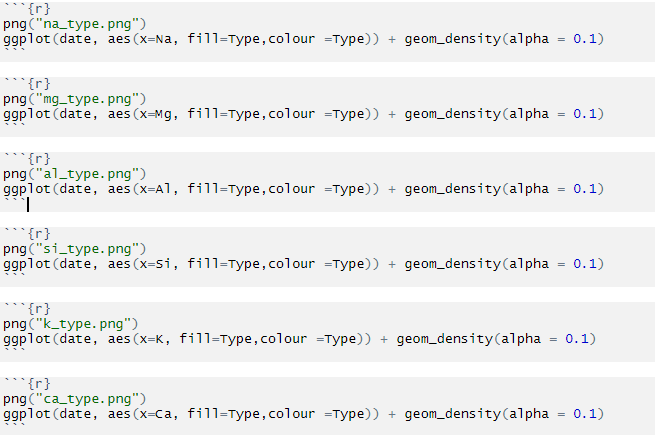


It was added as a factor within the data

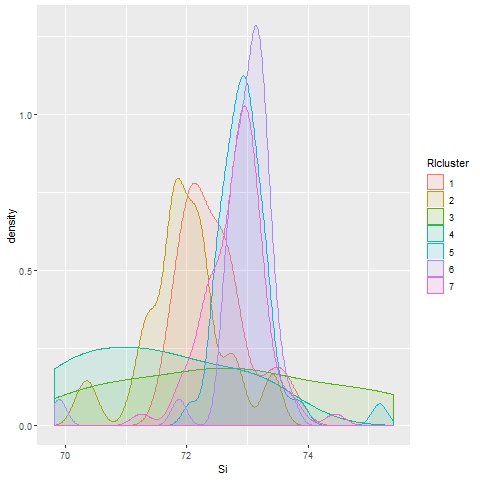
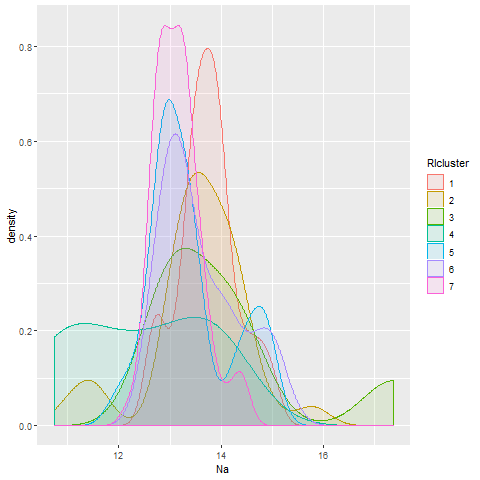
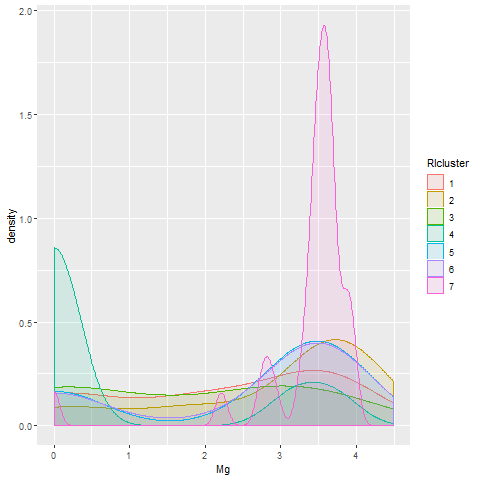
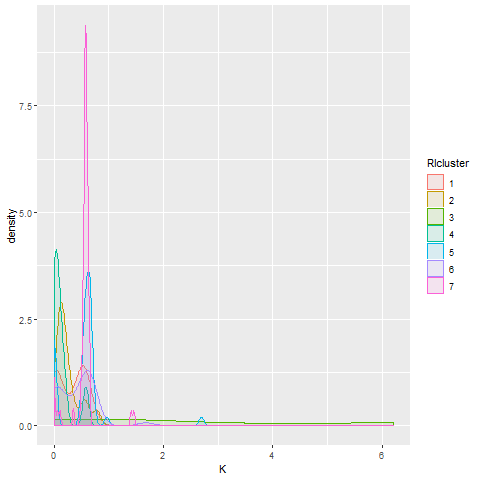
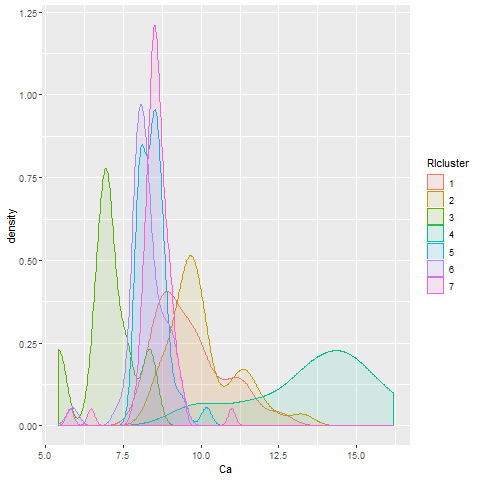
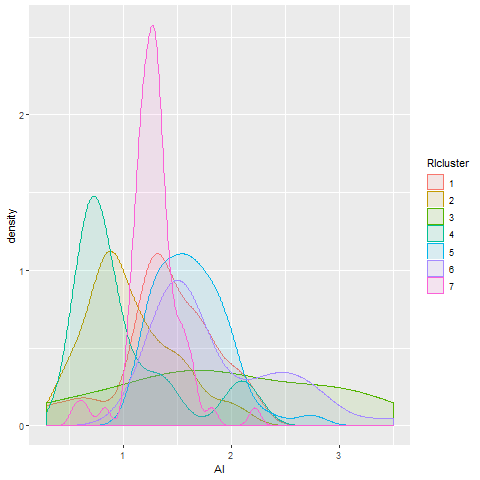
1. relationship of degrees and reflection to the elements

Here, the degrees of reflection for each element are explained.

And this was done through these codes:



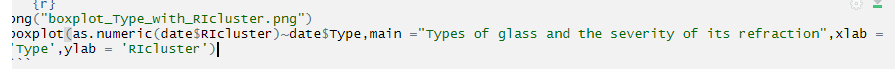
The output is:



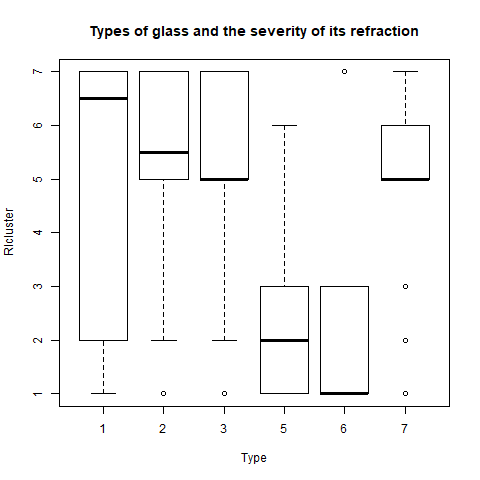
1. The relationship of type to degrees of reflection:

The relationship was made for each type of glass in relation to the degrees of reflection of each type of glass.

This was done via this code:



And its result



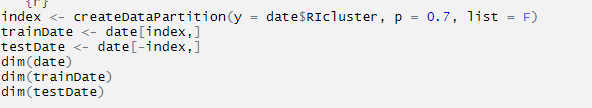
1. Training algorithms:

* Split the samples:

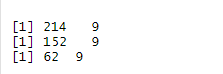
The data was divided into two:

* test samples: Where the test sample is at 30% and is separated from the training samples and used to verify the success and training of the network in a proportional way to verify the performance of the algorithm
* data training samples: Where the training sample is 70% separated from the test samples and used to train the algorithm used

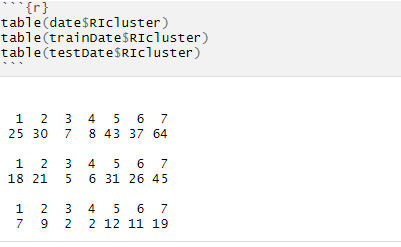
The process code is



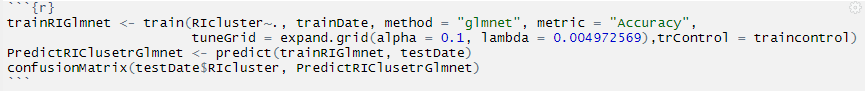
The output is



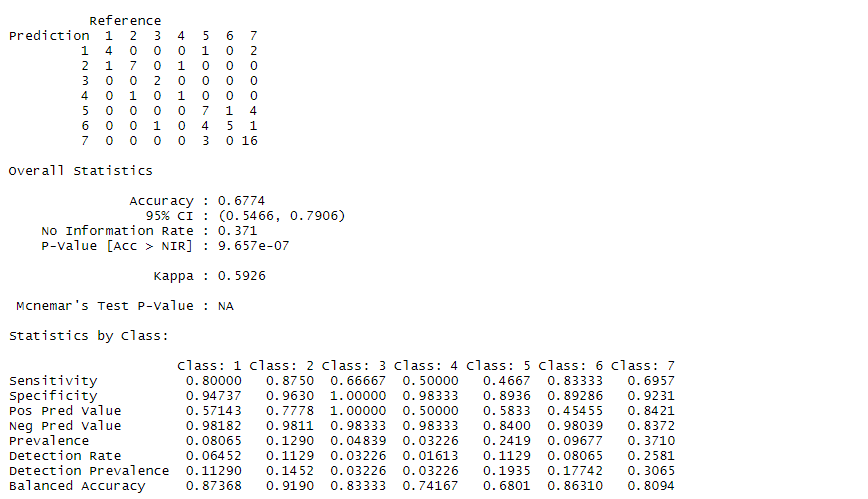
Display sample ratios for each degree of reflection



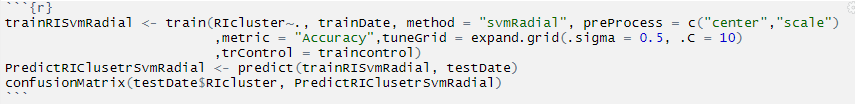
* Algorithms:
* **Multinomial Logistic Regression** This is a classifaction technique quite similar to logistic regression however the dependent (target) variable consists of more than two level. It is used to describe the data and to expalin the relationship between one dependent nominal variable and multiple independent variables. In our case, we will attempt to explain the relationship between Glass Type and independent variables such as Refractive Index and elements quantities.



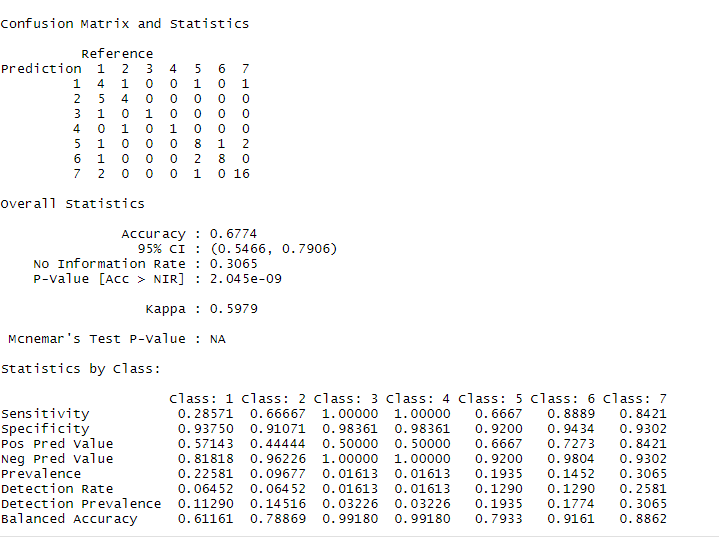
Output is:



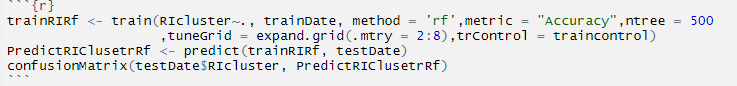
* **Support Vector Machine:** The SVM algorithm is implemented using a kernel. We plot the data as a point in a mutli-dimensional space with the value of each feature being the value of a particular co-ordinate. The algorithm outputs an optimal hyperplane which categorizes new data samples. The optimal hyperplane is found through support vectors, which are co-ordinates of individual observations and it's a frontier which best segregates the classes.



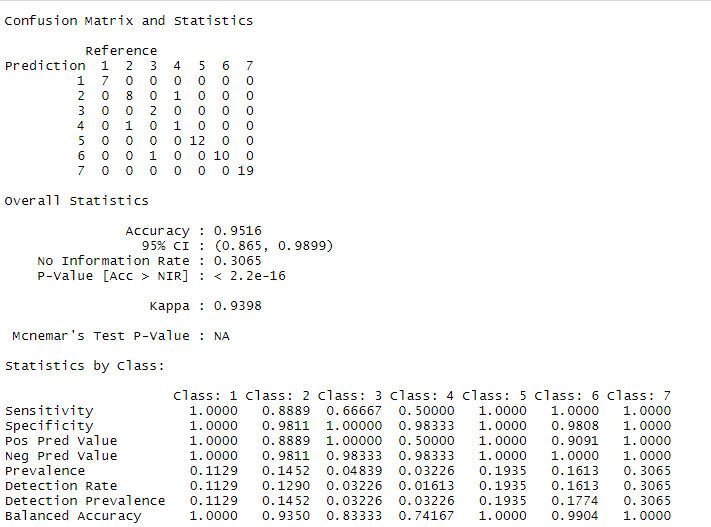
Output is:



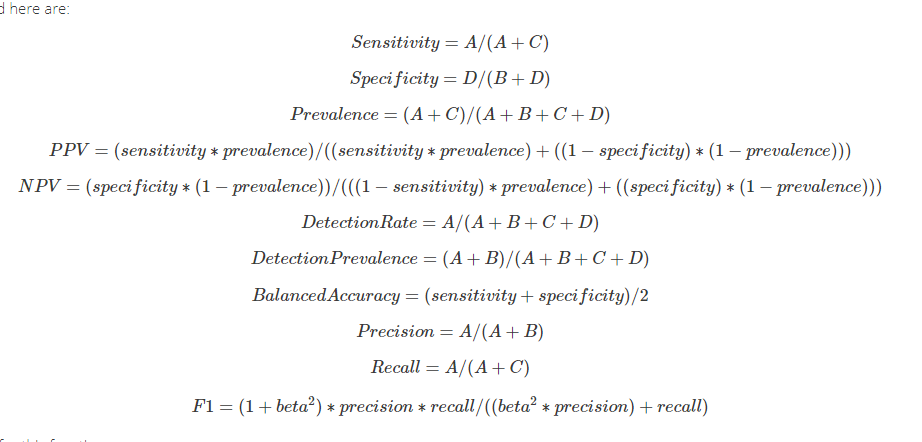
* **Random Forest** These are an ensemble learning method for classification and regression, that operate by constructing a multitude of decision trees at training time and outputing the class that is the mode of the classes in classification.



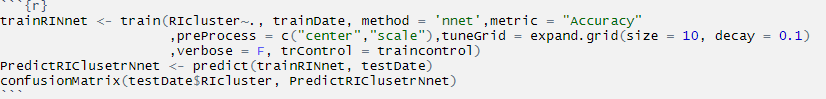
Output is:

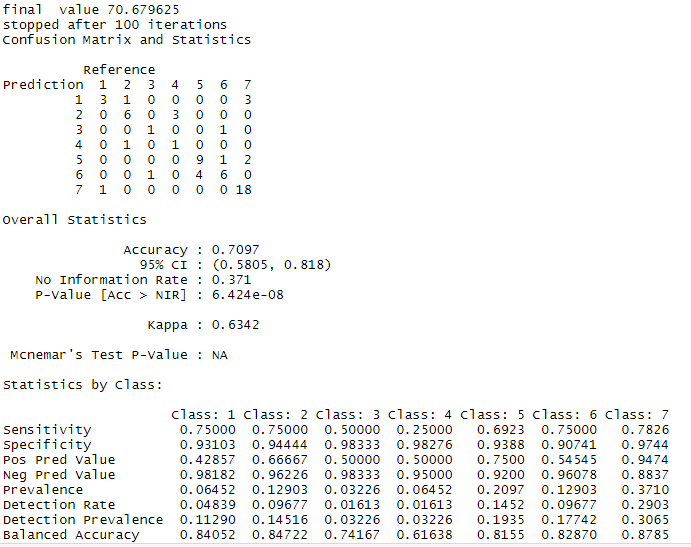
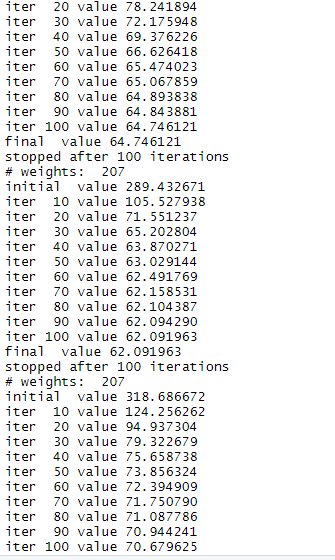
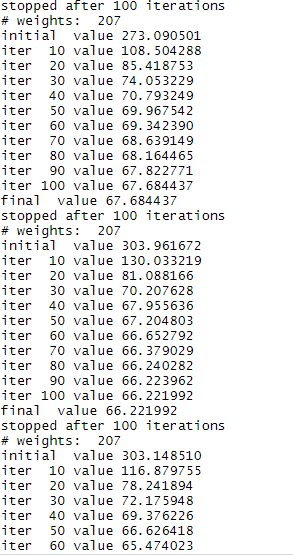
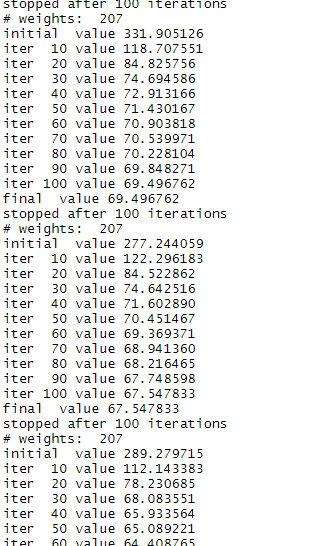
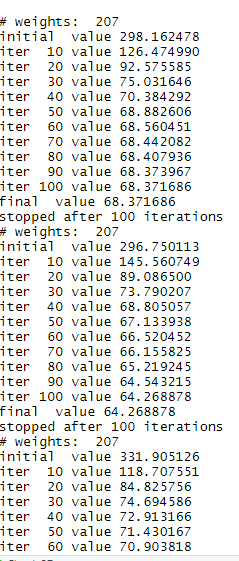


* accuracy = (true positive + true negatve) / all (100 times this is the same as percentCorrect)
* sensitivity = true pasitive rate = true positive / all positive (sensitivity is also called recall)
* specificity = true negative rate = true negative / all negative
* precision = positive predictive velue = true positive rate.



* **Neural Network** Neural networks consist of inut and output layers, as well as a hidden layer consisting of units that transform the input into something the output layer can use.





1. Arranging algorithms

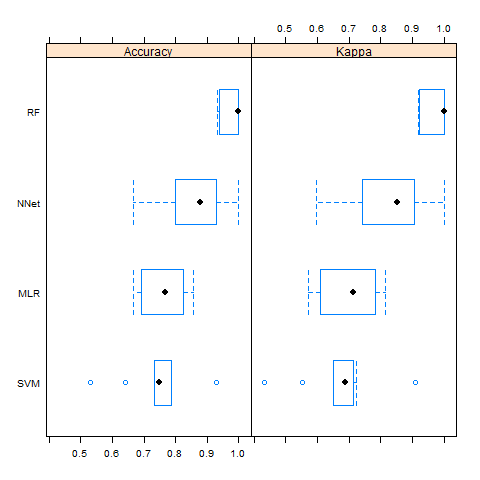
I invoke the training algorithms and review their comparison results, whichever results are better in performance

By this code:



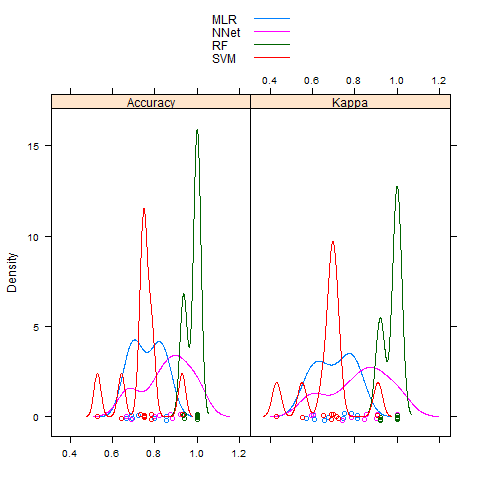
Boxplot algorithms and this code and its results





Densityplot mapping algorithms and this code and its results :





Accordingly, we decide that the algorithm **Random Forest** has achieved much better results than the rest of the algorithms