# Topic: K - Means Clustering

1.) Perform clustering (Both hierarchical and K means clustering) for the airlines data to obtain optimum number of clusters.

Draw the inferences from the clusters obtained.

**Ans:**

**Data Preprocessing:**

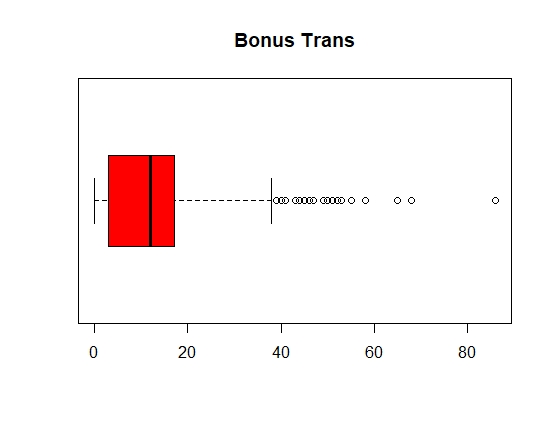
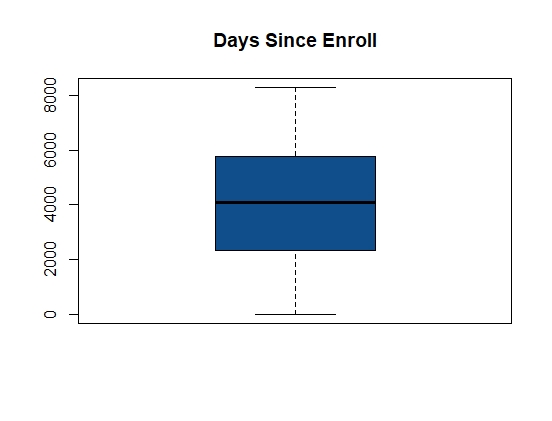
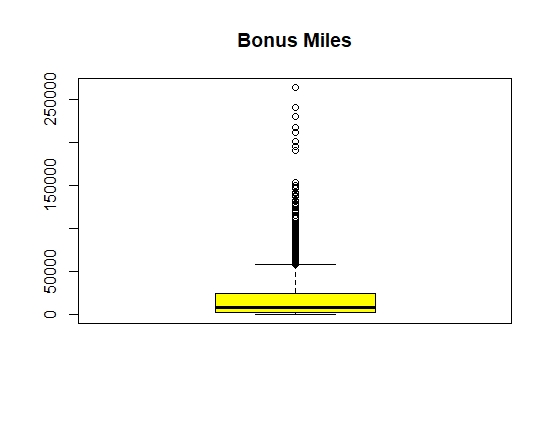
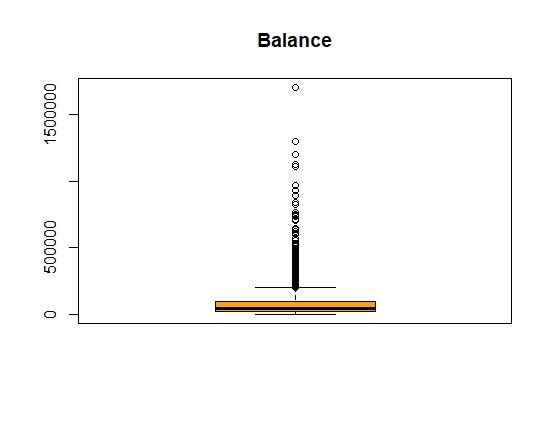
* Removing unnecessary columns.

**Normalizing the data:**

* Using the normalization function, the data is normalized and the data will come under same level.

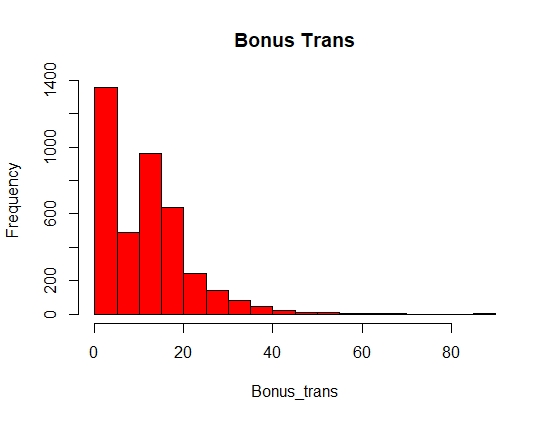
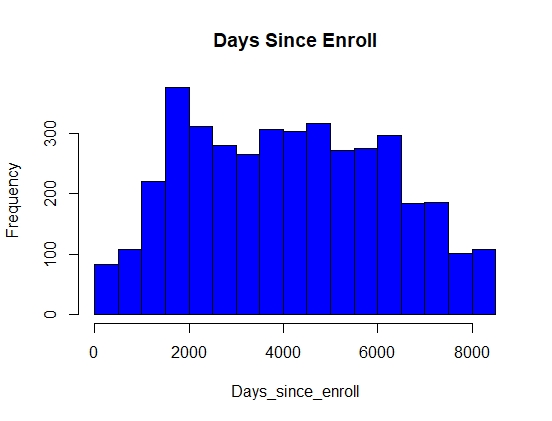
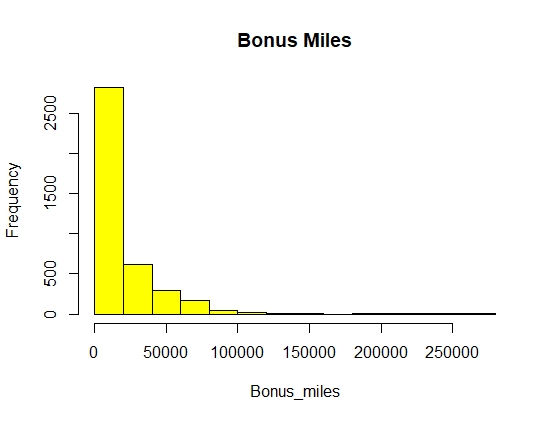
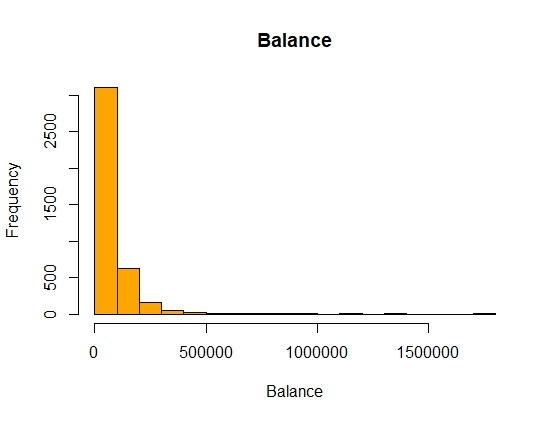
**Exploratory data analysis:**

**Box Plot Representation:**

****

* From the above graphical representation, its clearly shows that outliers exist in individual variables except in days to be enrolled.

**Histogram Representation:**

****

* From the above graphs all the variables are right skewed in nature except from days since enroll.

**Elbow curve to decide the k value:**

str(fit) --🡪 function to fit the structure / information of K means

List of 9

$ cluster : int [1:3999] 2 2 2 2 3 2 3 2 1 3 ...

$ centers : num [1:3, 1:11] 1.1948 -0.2982 0.4068 0.7186 -0.0593 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:3] "1" "2" "3"

.. ..$ : chr [1:11] "Balance" "Qual\_miles" "cc1\_miles" "cc2\_miles" ...

$ totss : num 43978

$ withinss : num [1:3] 9724 12404 8874

$ tot.withinss: num 31001

$ betweenss : num 12977

$ size : int [1:3] 192 2522 1285

$ iter : int 7

$ ifault : int 0

- attr(\*, "class")= chr "kmeans"

1. The str () function gives the structure of the kmeans which includes various parameters like withinss, betweenss, etc, analyzing which you can find out the performance of kmeans.
2. betweenss: Between sum of squares i.e. Intracluster similarity
3. withinss: Within sum of square i.e. Intercluster similarity
4. totwithinss: Sum of all the withinss of all the clusters i.e. Total intra-cluster similarity
5. A good clustering, will have a lower value of withinss and higher value of betweenss which depends on the number of clusters ‘k’ chosen initially. Let us see how we can find the optimal value of ‘k’.

**Finding the optimal value of ‘k’**

An optimal value of ‘k’ is the value which gives us a converged set of clusters with minimum distortion. Greater the distortion, worse will be the clusters formed

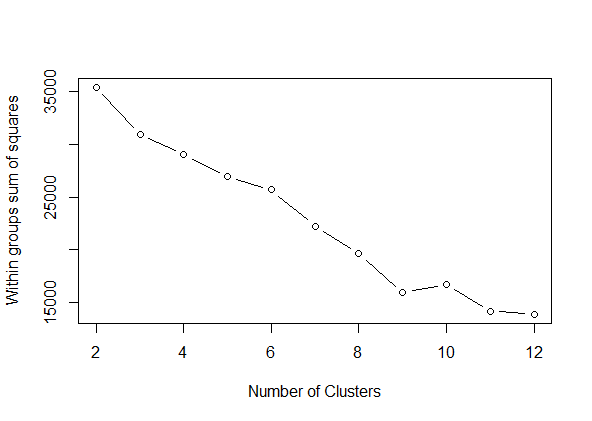
twss <- NULL

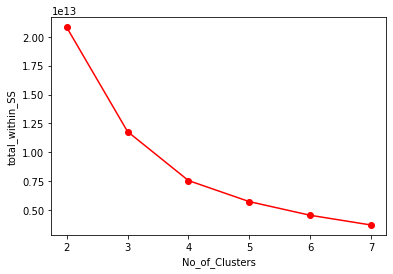
for (i in 2:8) {

twss <- c(twss, kmeans(normalized\_data, centers = i)$tot.withinss)

}

**TWSS (Total Within Sum of Square root)**





As we can see that the point at no 7 is very near to the x-axis which means it has very lesser variance whereas, till no 3 the difference of each point is very less.

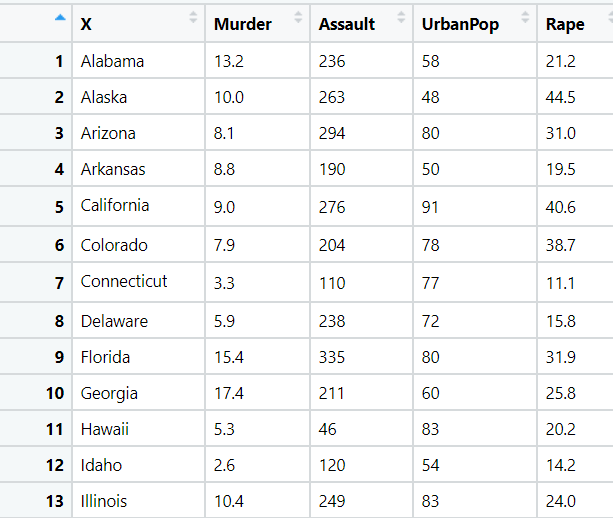
So, as per the graph, the steep difference was at no 3 so we have taken 3 Clusters.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Clusters** | **Bal** | **Qual.miles** | **Cc1. miles** | **Cc2. miles** | **Cc3. miles** | **Bonus. miles** | **Bonus.**  **trans** | **flight.**  **miles** | **fligtht.**  **trans** | **Days.**  **Since enroll** | **awards** |
| **1** | 195200.23 | 783.20 | 2.15 | 1.03 | 1.00 | 31379.963 | 26.99 | 5412.00 | 15.79 | 4715.94 | 0.81 |
| **2** | 44370.73 | 96.72 | 1.23 | 1.01 | 1.00 | 4837.90 | 7.01 | 200.46 | 0.62 | 3707.35 | 0.20 |
| **3** | 117627.58 | 157.88 | 3.73 | 1.00 | 1.03 | 40499.72 | 18.99 | 346.26 | 1.02 | 4883.02 | 0.65 |

**INFERENCES DRAWN FROM ABOVE ANALYSIS**

1. Cluster 1 has max no of balance, bonus miles, flight miles, flight transactions and also with highest percentage of awards will definitely shows they are the Premium customers.
2. Cluster 2 seems to be the customers comes under average category as per the above data in terms of balance, bonus miles, flight miles, flight trans and the awards comparing both Cluster 1 & Cluster 2.
3. Cluster 3 are the customers who are to be the next set of Premium customers compare to Cluster 1.

2.) Perform K-Means Clustering for the crime data and identify the number of clusters formed and draw inferences.



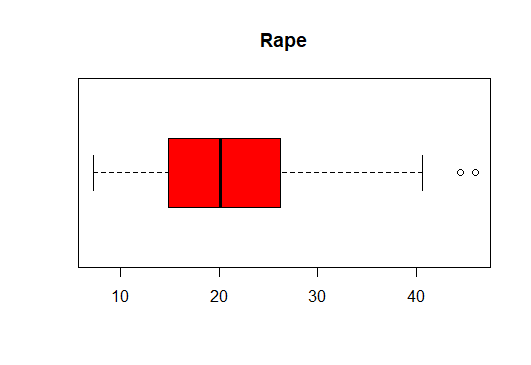
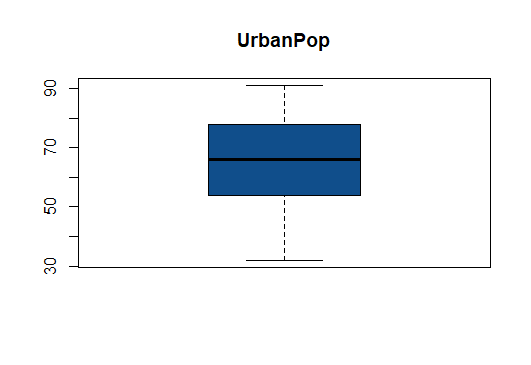
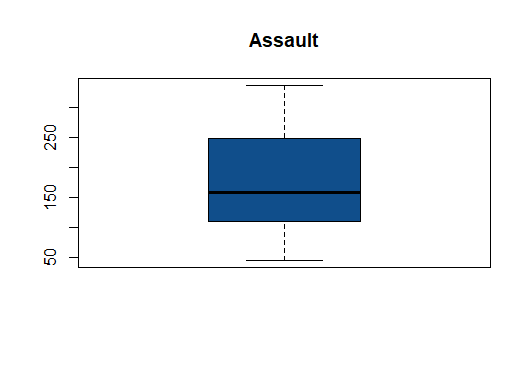
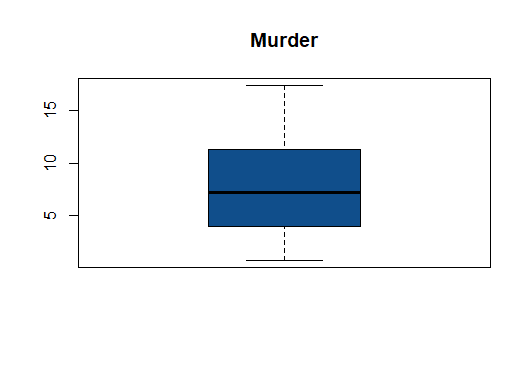
**Ans:**

**Normalizing the data:**

* Using the normalization function, the data is normalized and the data will come under same level.

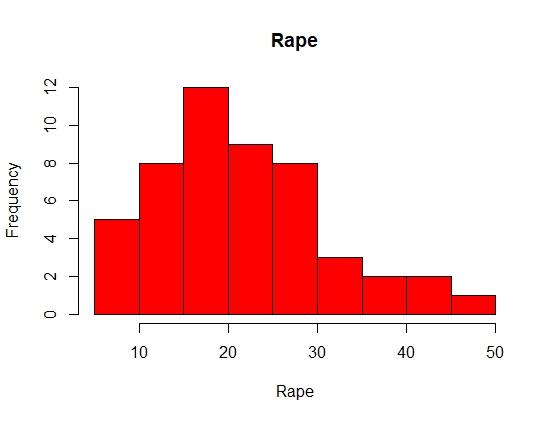
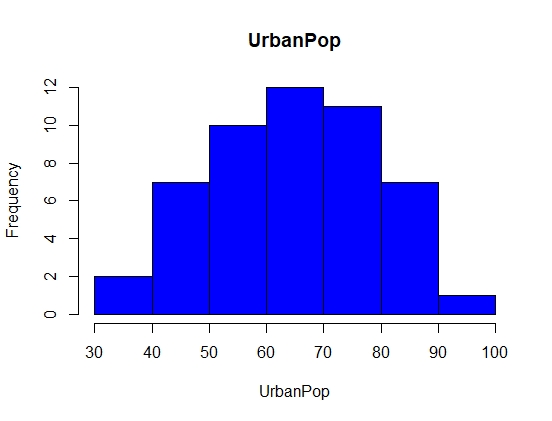
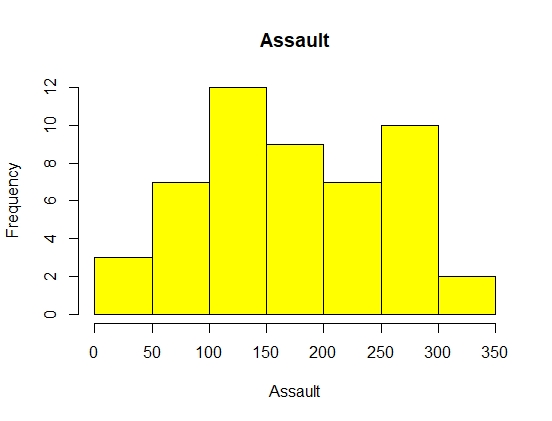
**Exploratory Data Analysis:**

**Box Plot Representation:**

****

* From the above graphical Representation, its clearly shows there exists no outliers

**Histogram Representation:**

****

* From the above histogram, its clearly shows the data is normally distributed.

**Elbow curve to decide the k value:**

str(fit) --🡪 function to fit the structure / information of K means

List of 9

$ cluster : int [1:50] 1 3 3 1 3 3 2 2 3 1 ...

$ centers : num [1:3, 1:4] 1.412 -0.701 0.695 0.874 -0.707 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:3] "1" "2" "3"

.. ..$ : chr [1:4] "Murder" "Assault" "UrbanPop" "Rape"

$ totss : num 196

$ withinss : num [1:3] 8.32 53.35 19.92

$ tot.withinss: num 81.6

$ betweenss : num 114

$ size : int [1:3] 8 29 13

$ iter : int 3

$ ifault : int 0

- attr(\*, "class")= chr "kmeans"

1. The str () function gives the structure of the kmeans which includes various parameters like withinss, betweenss, etc, analyzing which you can find out the performance of kmeans.
2. betweenss: Between sum of squares i.e. Intracluster similarity
3. withinss: Within sum of square i.e. Intercluster similarity
4. totwithinss: Sum of all the withinss of all the clusters i.e. Total intra-cluster similarity
5. A good clustering, will have a lower value of withinss and higher value of betweenss which depends on the number of clusters ‘k’ chosen initially. Let us see how we can find the optimal value of ‘k’.

**Finding the optimal value of ‘k’**

An optimal value of ‘k’ is the value which gives us a converged set of clusters with minimum distortion. Greater the distortion, worse will be the clusters formed

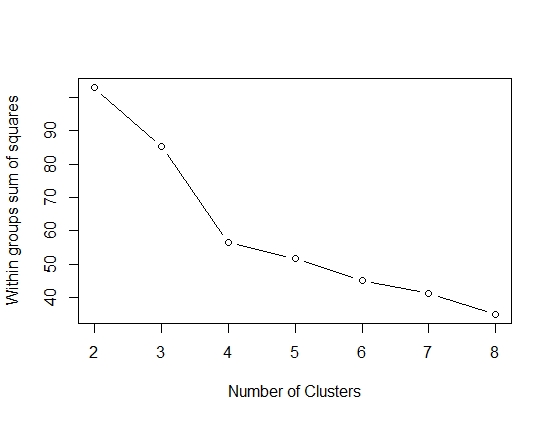
twss <- NULL

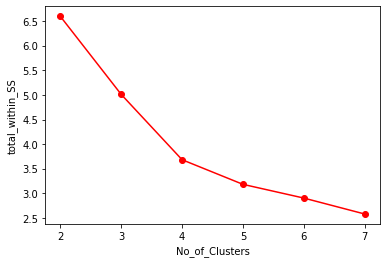
for (i in 2:8) {

twss <- c(twss, kmeans(normalized\_data, centers = i)$tot.withinss)

}

Twss



****

This is the plot between ‘k’, the number of clusters and the ‘totwithinss’ (or distortion) for each value of k. You can see when the number of cluster is less, there is a gradual decrease in distortion but as we keep on increasing the value of k, the rate of reduction of distortion values becomes constant.

This value of k beyond which the distortion rate becomes constant is the optimal value. Here k=3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clusters** | **Murder** | **Assault** | **Urban Pop** | **Rape** |
| 1 | 13.93 | 243.62 | 53.75 | 21.41 |
| 2 | 4.73 | 111.82 | 64.10 | 15.82 |
| 3 | 10.81 | 257.38 | 76.00 | 33.19 |

**INFERENCES DRAWN FROM ABOVE ANALYSIS**

1. Cluster 1 has a huge crime rate with the lesser population.
2. Cluster 2 has a low crime rate with the moderate population compare to Cluster 1 & Cluster 3.
3. Cluster 3 with high population the crime rate is also moderately high even though the murder rate is low when compared to cluster 1.

3.) Analyze the information given in the following ‘Insurance Policy dataset’ to

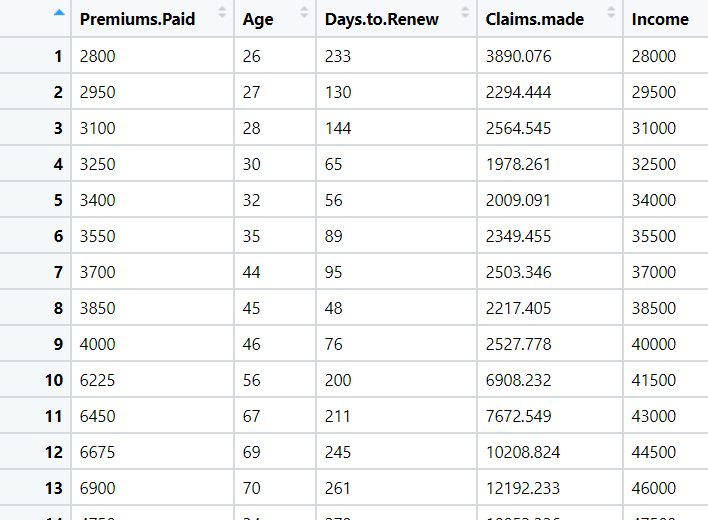
create clusters of persons falling in the same type.

The description of the attributes in the dataset are as follows: Premium Paid – Amount paid by the person

Age – Indicates the age of the person

Days to renew – Days remaining to renew the policy

Claims made – Indicates the claims already made by the person Income – Net income of the person



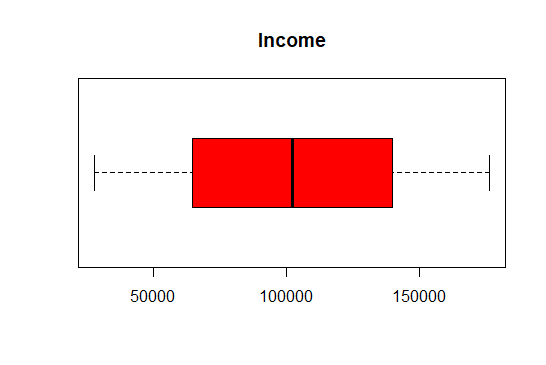
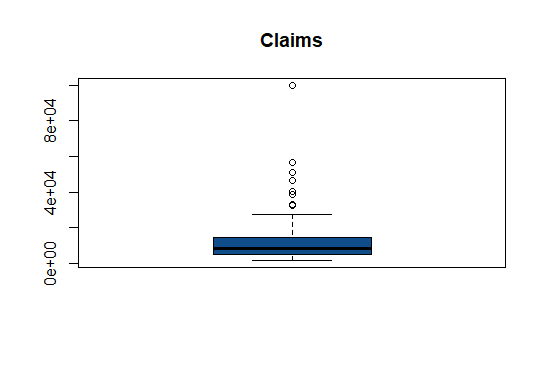
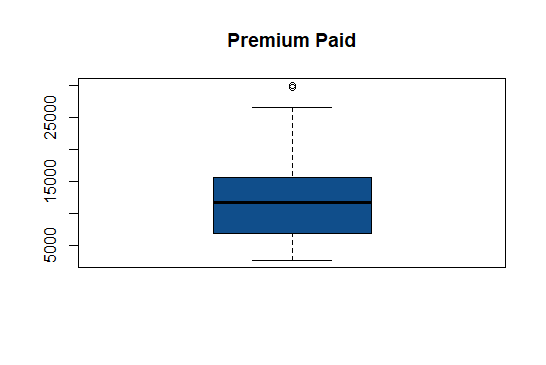
**Ans:**

**Normalizing the data:**

* Using the normalization function, the data is normalized and the data will come under same level.

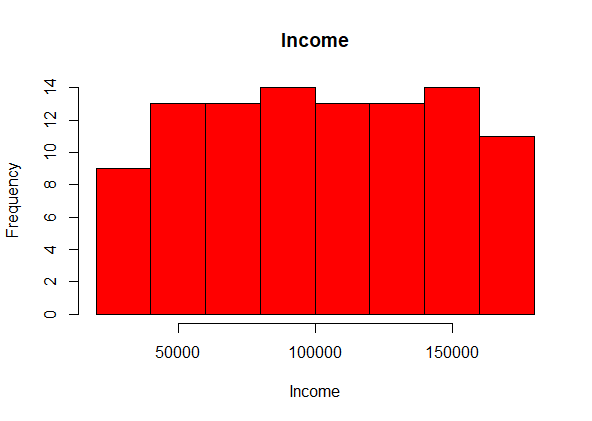
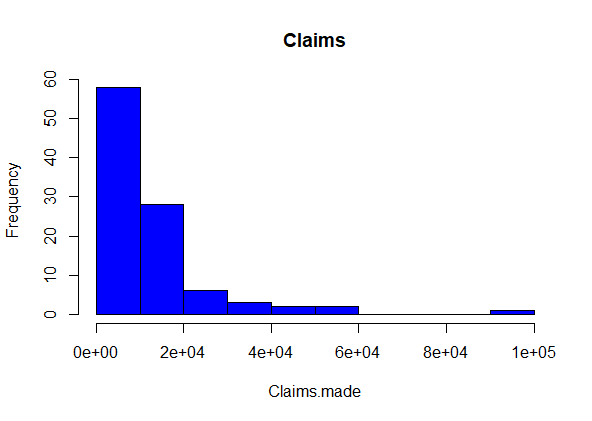
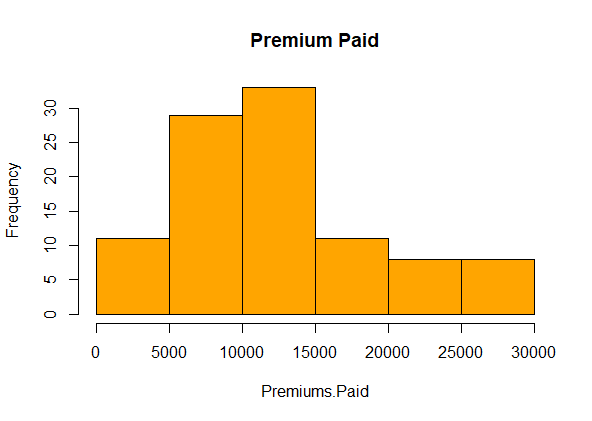
**Exploratory Data Analysis:**

**Box Plot Representation:**



* From the above graphical Representation, its clearly shows there exists outliers in premium paid and claims.

**Histogram Representation:**

****

* From the above histogram we can clearly see that the data is normally distributed for premium paid and the income whereas, data is right skewed in case of claims made.

**Elbow curve to decide the k value:**

str(fit) --🡪 function to fit the structure / information of K means

List of 9

$ cluster : int [1:100] 1 2 2 2 2 2 2 2 2 1 ...

$ centers : num [1:3, 1:5] -0.5201 -0.3828 1.5642 0.0717 -0.4004 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:3] "1" "2" "3"

.. ..$ : chr [1:5] "Premiums.Paid" "Age" "Days.to.Renew" "Claims.made" ...

$ totss : num 495

$ withinss : num [1:3] 64.9 110.1 83.5

$ tot.withinss: num 258

$ betweenss : num 237

$ size : int [1:3] 19 60 21

$ iter : int 3

$ ifault : int 0

- attr(\*, "class")= chr "kmeans"

* The str () function gives the structure of the kmeans which includes various parameters like withinss, betweenss, etc, analyzing which you can find out the performance of kmeans.
* betweenss: Between sum of squares i.e. Intracluster similarity
* withinss: Within sum of square i.e. Intercluster similarity
* totwithinss: Sum of all the withinss of all the clusters i.e. Total intra-cluster similarity
* A good clustering, will have a lower value of withinss and higher value of betweenss which depends on the number of clusters ‘k’ chosen initially. Let us see how we can find the optimal value of ‘k’.

**Finding the optimal value of ‘k’**

An optimal value of ‘k’ is the value which gives us a converged set of clusters with minimum distortion. Greater the distortion, worse will be the clusters formed

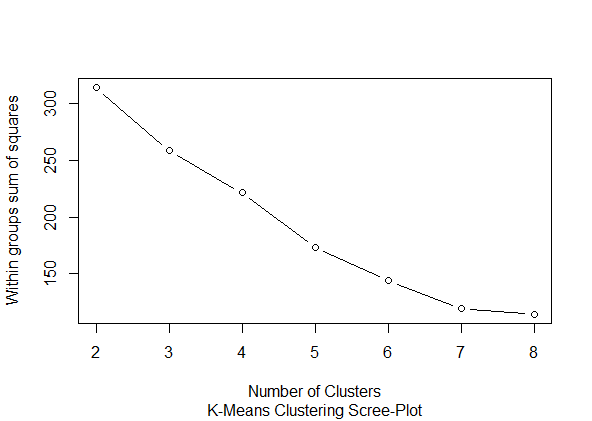
twss <- NULL

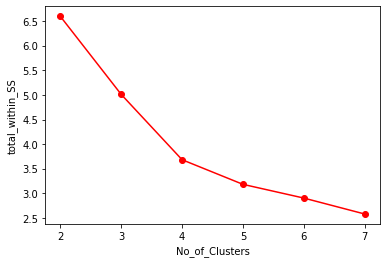
for (i in 2:8) {

twss <- c(twss, kmeans(normalized\_data, centers = i)$tot.withinss)

}

Twss



****

This is the plot between ‘k’, the number of clusters and the ‘totwithinss’ (or distortion) for each value of k. You can see when the number of cluster is less, there is a gradual decrease in distortion but as we keep on increasing the value of k, the rate of reduction of distortion values becomes constant.

This value of k beyond which the distortion rate becomes constant is the optimal value. Here k=3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clusters** | **Age** | **Days to renew** | **Claims made** | **Income** |
| 1 | 47.10 | 244.52 | 17312.25 | 77263.16 |
| 2 | 40.55 | 69.83 | 6329.06 | 91525.00 |
| 3 | 61.09 | 152.57 | 26153.45 | 155500.00 |

**INFERENCES DRAWN FROM ABOVE ANALYSIS**

1. Cluster 3 has the highest income with an age group of above 60 who have made highest claims compare to Cluster 1 & Cluster 3
2. Cluster 2 has the moderate income with an age group of 40 who made the lowest claims.
3. Cluster 1 with a moderate age group of above 45 who has the lowest income of all and the claims made by them are the 2nd highest after Cluster 3.

**Hints:**

1. Business Problem
   1. Objective
   2. Constraints (if any)
2. Data Pre-processing

2.1 Data cleaning, Feature Engineering, EDA etc.

1. Model Building
   1. Partition the dataset
   2. Model(s) - Reasons to choose any algorithm
   3. Model(s) Improvement steps
   4. Model Evaluation
   5. Python and R codes
2. Deployment

4.1 Deploy solutions using R shiny and Python Flask.

1. Result Share the benefits/impact of the solution - how or in what way the business (client) gets benefit from the solution provided.

**Note:**

1. For each assignment the solution should be submitted in the format
2. Research and Perform all possible steps for improving the model(s) accuracy Ex: Feature Engineering, Hyper Parameter tuning, etc.
3. All the codes (executable programs) are running without errors
4. Documentation of the module should be submitted along with R & Python codes, elaborating on every step mentioned here