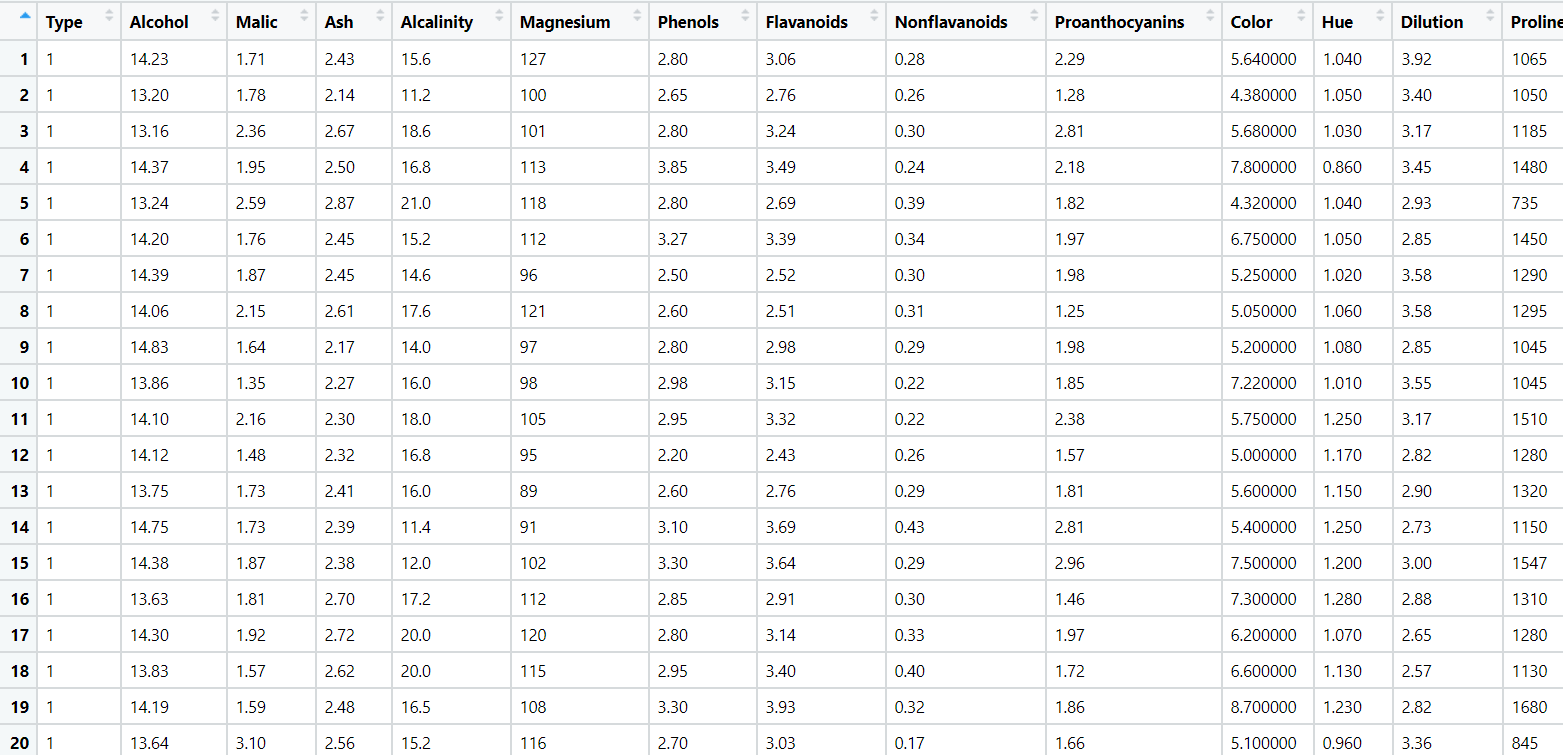
# Topic: Dimension Reduction (PCA)

Perform Principal component analysis and perform clustering using first 3 principal component scores (both Hierarchical & K-Mean clustering). Use Scree plot or elbow curve and obtain optimum number of clusters and check whether we have obtained same number of clusters with the original data



**Data Preprocessing:**

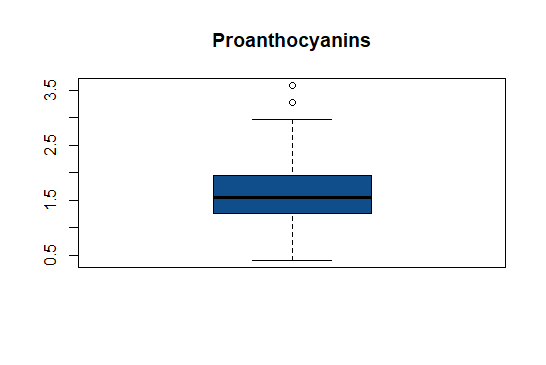
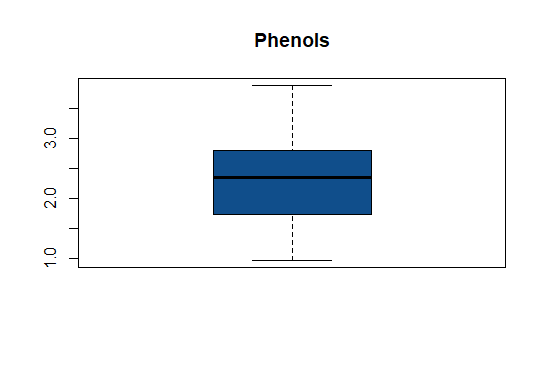
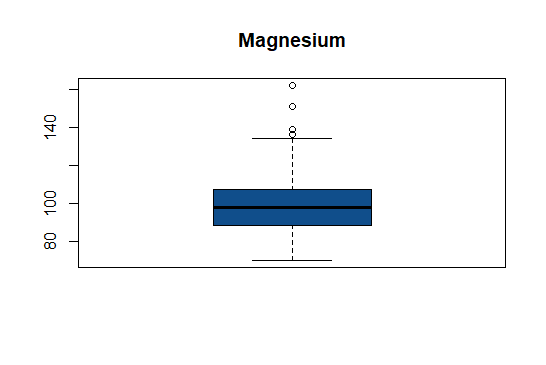
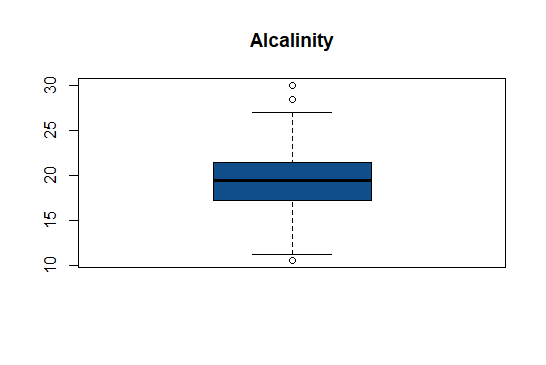
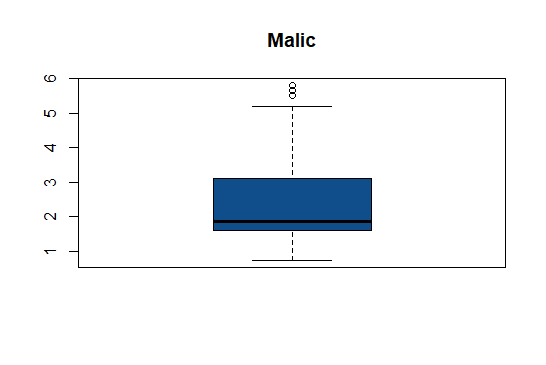
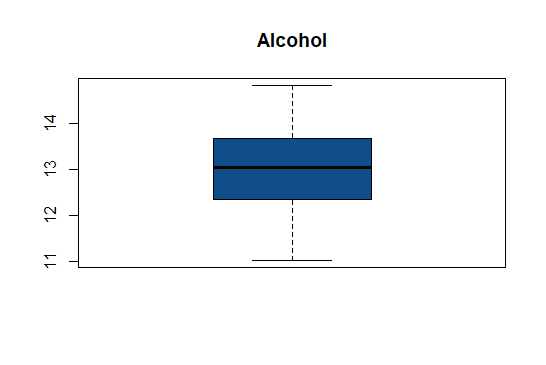
* Importing the dataset using read.csv.
* Removing unnecessary columns.

**Normalizing the data:**

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

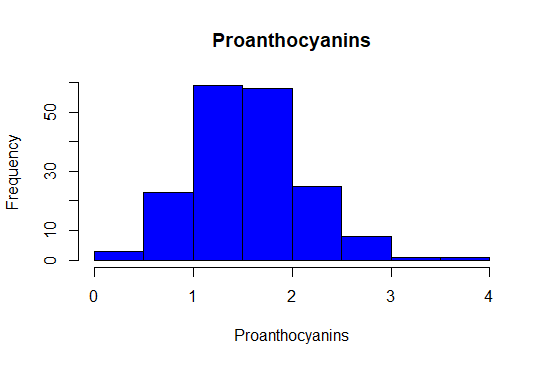
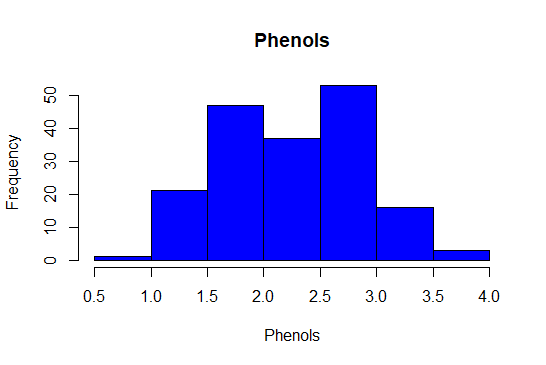
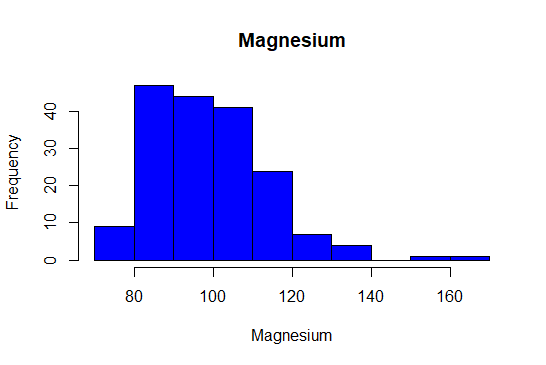
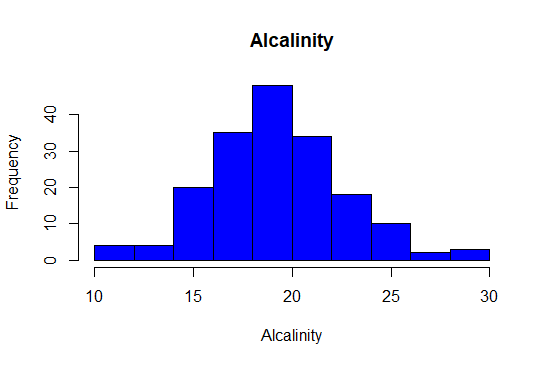
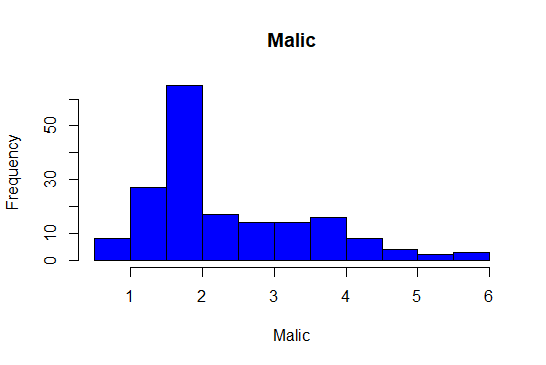
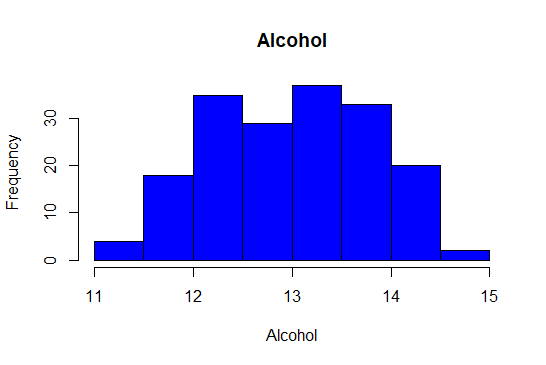
**Exploratory Data Analysis:**

**Box Plot Representation:**



* From the above graphical representation, It clearly show that there exist outliers in Malic, Magnesium, Proanthocyanins

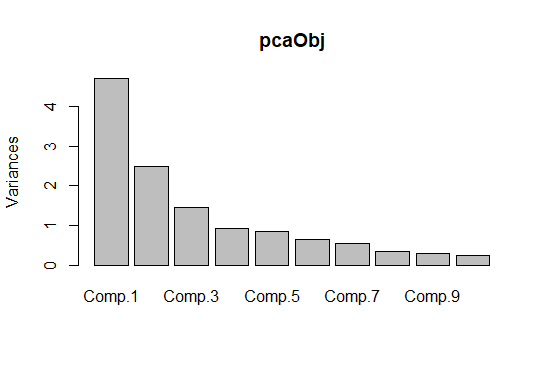
**Histogram Representation:**



* From the above graphical representation, It clearly show that the data is normally distributed in all the variables except Malic, the data is right skewed.

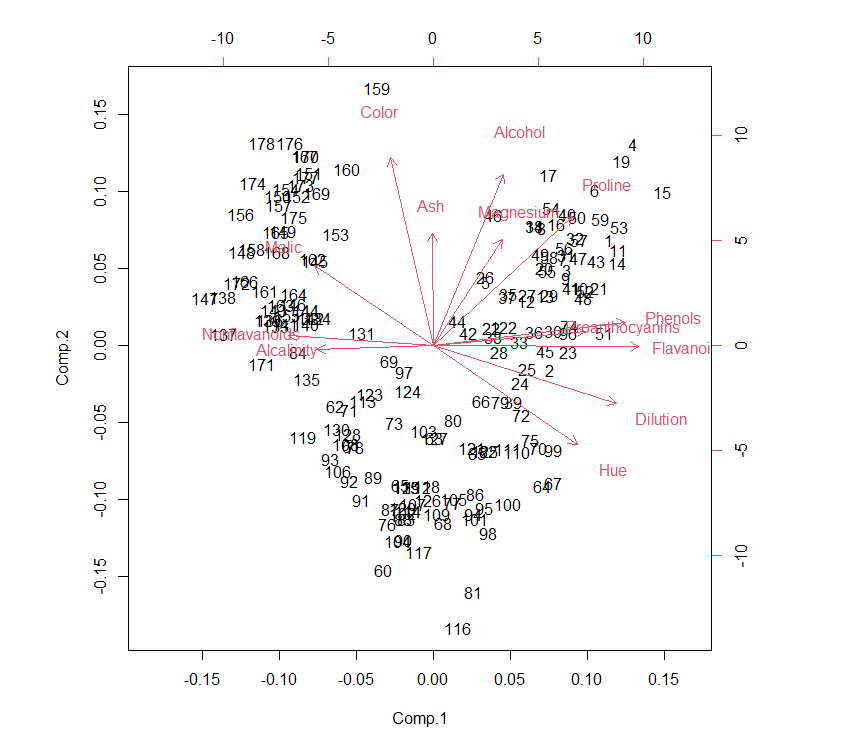
**DIMENSION REDUCTION (PCA) ANALYSIS:**

* Two main goals of dimensionality reduction
  + find structure in features
  + aid in visualization
* PCA has 3 goals
  + find a linear combination of variables to create principle components
  + maintain as much variance in the data as possible
  + principal components are uncorrelated
* intuition
  + with an x y correlation scatter plot, the best 1 dimension to explain the variance in the data is the linear regression line
  + this is the first principal component
  + then the distance of the points from the line is the component score



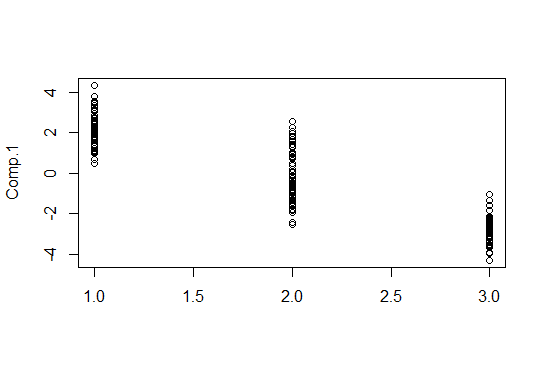
* The above graph shows that the first three components explain the majority of the variance in our data.

**Biplot Analysis of PCA:**

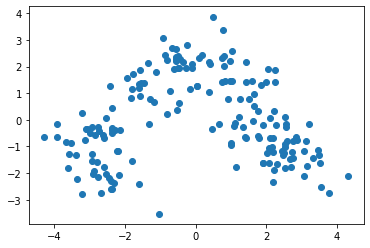
****

* The graph shows at least two clearly distinguishable clusters. This factoid tells us that the observations in the dataset can be grouped.
* It’s important to note that we do not have a target variable by which to label these groups, so we do not know exactly what these labels are.
* Biplot does not show all the meaningful principal components, however. To visualize the rest of the reduced dataset with much greater granularity, we will use k-means clustering.

**Scatter Plot:**

****

* The above graph represents how the points of 3 components are scattered.

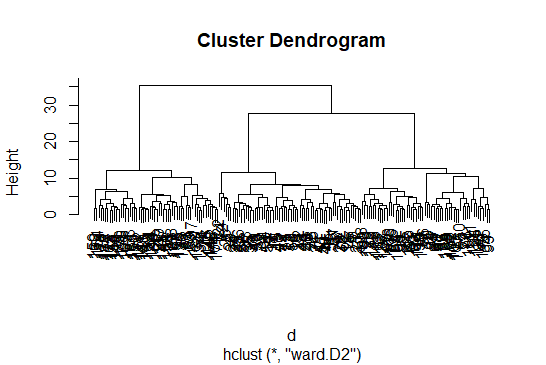
****

* The above graph represents how the points of 2 components are scattered and forming clusters.

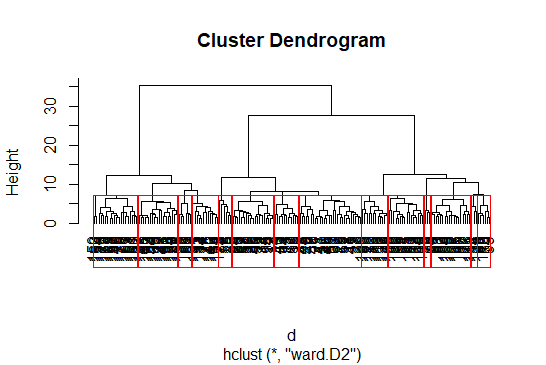
**HIERARCHICAL CLUSTERING:**

**Dendrogram:**

* Standardization/Normalization of data is not done then below issues will influence the model.
* Distance measure will be wrongly calculated, if all the variables are not with equal weight.
* Largest scale dominating the measure
* The below dendogarm generated with the given data set with standardization is as below.
* Using **ward.D2** method the dendogarm is prepared

****

* Now the dendogram is divided into clusters and the aggregate of each cluster is compared with the other clusters.



* Now the output is stored in the csv file into working directory.

**K MEANS CLUSTERING:**

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

$ cluster : int [1:178] 2 2 2 2 4 2 2 2 2 2 ...

$ centers : num [1:4, 1:13] 0.186 0.958 -0.905 -0.787 0.902 ...

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

.. ..$ : chr [1:4] "1" "2" "3" "4"

.. ..$ : chr [1:13] "Alcohol" "Malic" "Ash" "Alcalinity" ...

$ totss : num 2301

$ withinss : num [1:4] 303 269 290 307

$ tot.withinss: num 1169

$ betweenss : num 1132

$ size : int [1:4] 49 56 45 28

$ 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

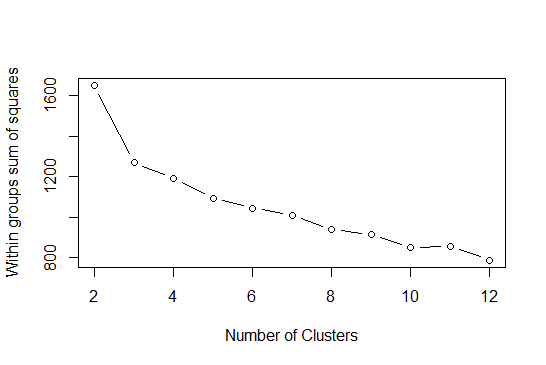
twss <- NULL

for (i in 2:8) {

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

}

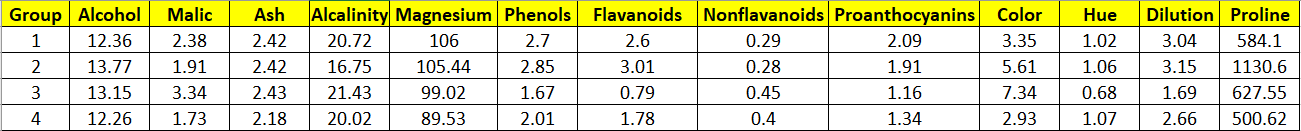
twss



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=4.

**Inferences:**



* Cluster 1 has high values of Magnesium, proanthocyanins and all other factors are moderate in nature
* Cluster 2 has high value of Alcohol, phenols, flavonoids, dilution and proline and others factors are less compared to other clusters
* Cluster 3 has high values of Malic, Ash, Alcalinity, Nonflavonoids and color and other factors are low and moderate while comparing with other clusters
* Cluster 4 has high value of hue and other factors are low and moderate while comparing with other clusters

# 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