Lab Evaluation Report

UCS654 PREDICTIVE ANALYTICS USING STATISTICS

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Submitted By:

Ratish Jindal (101803004)

COE-1

Submitted To :

Mr. PS rana

Dataset Name: Wine Data Set

Dataset Source: https://archive.ics.uci.edu/ml/datasets/wine

Github Respositories: https://github.com/zeearo/PCA-analysis-for-Wine-dataset

Source:

Original Owners:

Forina, M. et al, PARVUS - An Extendible Package for Data Exploration, Classification and Correlation. Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, 16147 Genoa, Italy.

Data Set Information:

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

The attributes are (donated by Riccardo Leardi, riclea '@' anchem.unige.it)
1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10)Color intensity
11)Hue
12)OD280/OD315 of diluted wines
13)Proline

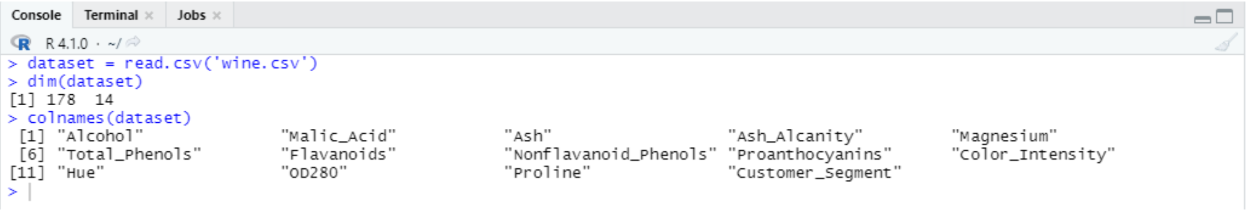
CODE:

Part 1 – Analytics and Visualization

dataset = read.csv('wine.csv')   #import datset

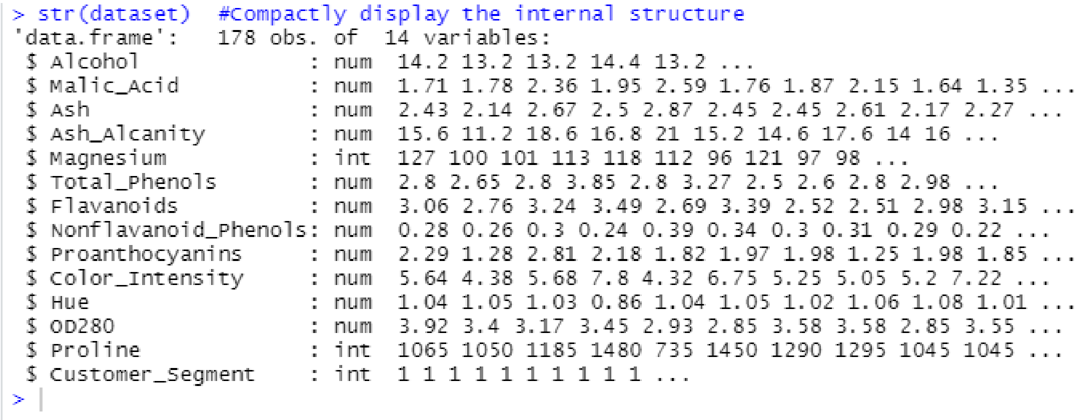
dim(dataset)   #output the size/shape of the dataset

colnames(dataset)   #list the names of the columns of the dataset



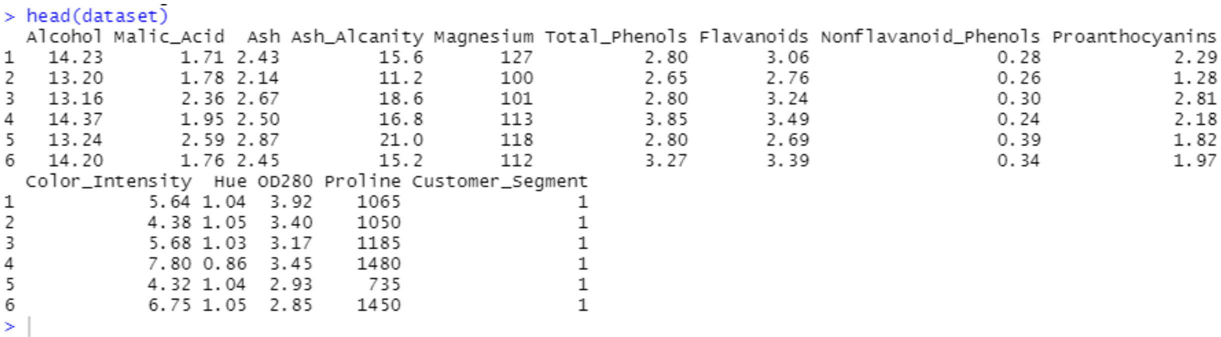
```
> dataset = read.csv('wine.csv')
> dim(dataset)
[1] 178  14
> colnames(dataset)
 [1] "Alcohol"              "Malic_Acid"           "Ash"                  "Ash_Alcanity"         "Magnesium"
 [6] "Total_Phenols"        "Flavanoids"           "Nonflavanoid_Phenols" "Proanthocyanins"      "Color_Intensity"
[11] "Hue"                  "OD280"                "Proline"              "Customer_Segment"
> 
```

str(dataset)   #Compactly display the internal structure



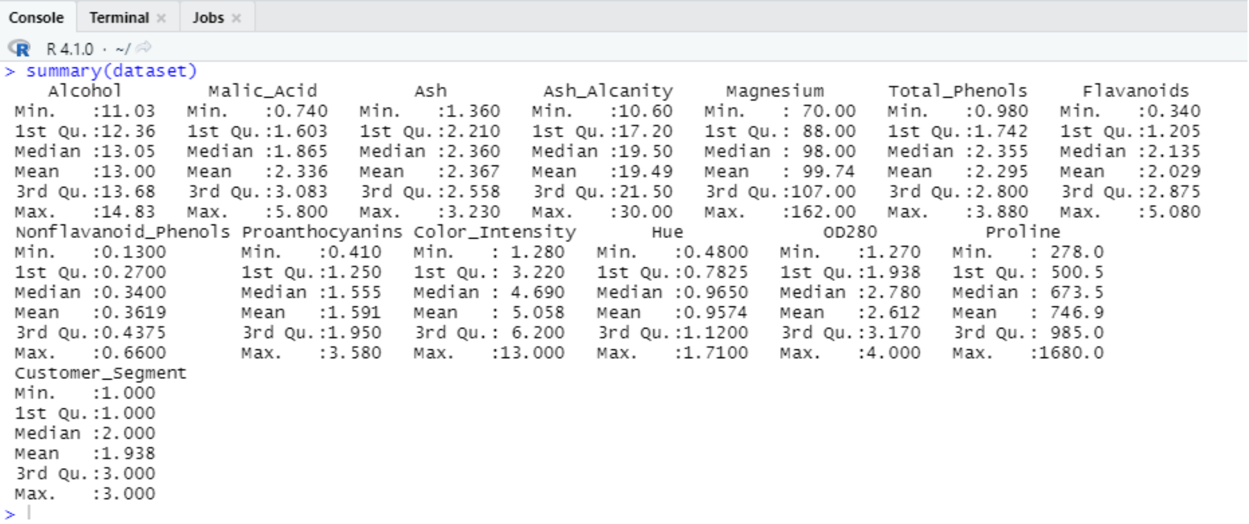
```
> str(dataset)  #Compactly display the internal structure
'data.frame':	178 obs. of  14 variables:
 $ Alcohol             : num  14.2 13.2 13.2 14.4 13.2 ...
 $ Malic_Acid          : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
 $ Ash                 : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
 $ Ash_Alcanity        : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
 $ Magnesium           : int  127 100 101 113 118 112 96 121 97 98 ...
 $ Total_Phenols       : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
 $ Flavanoids          : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
 $ Nonflavanoid_Phenols: num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
 $ Proanthocyanins     : num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
 $ Color_Intensity     : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
 $ Hue                 : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
 $ OD280               : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
 $ Proline             : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
 $ Customer_Segment    : int  1 1 1 1 1 1 1 1 1 1 ...
> 
```

head(dataset)   #Returns the first or last parts



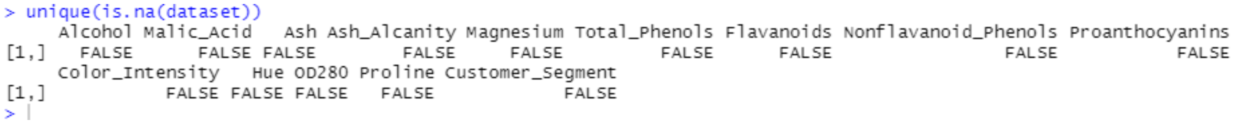
```
> head(dataset)
  Alcohol Malic_Acid  Ash Ash_Alcanity Magnesium Total_Phenols Flavanoids Nonflavanoid_Phenols Proanthocyanins
1   14.23       1.71 2.43         15.6       127          2.80       3.06                 0.28            2.29
2   13.20       1.78 2.14         11.2       100          2.65       2.76                 0.26            1.28
3   13.16       2.36 2.67         18.6       101          2.80       3.24                 0.30            2.81
4   14.37       1.95 2.50         16.8       113          3.85       3.49                 0.24            2.18
5   13.24       2.59 2.87         21.0       118          2.80       2.69                 0.39            1.82
6   14.20       1.76 2.45         15.2       112          3.27       3.39                 0.34            1.97
  Color_Intensity  Hue OD280 Proline Customer_Segment
1            5.64 1.04  3.92    1065                1
2            4.38 1.05  3.40    1050                1
3            5.68 1.03  3.17    1185                1
4            7.80 0.86  3.45    1480                1
5            4.32 1.04  2.93     735                1
6            6.75 1.05  2.85    1450                1
> 
```

summary(dataset)



```
> summary(dataset)
    Alcohol        Malic_Acid         Ash         Ash_Alcanity     Magnesium      Total_Phenols     Flavanoids   
 Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60   Min.   : 70.00   Min.   :0.980   Min.   :0.340  
 1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20   1st Qu.: 88.00   1st Qu.:1.742   1st Qu.:1.205  
 Median :13.05   Median :1.865   Median :2.360   Median :19.50   Median : 98.00   Median :2.355   Median :2.135  
 Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49   Mean   : 99.74   Mean   :2.295   Mean   :2.029  
 3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50   3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.875  
 Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00   Max.   :162.00   Max.   :3.880   Max.   :5.080  
 Nonflavanoid_Phenols Proanthocyanins Color_Intensity      Hue              OD280          Proline      
 Min.   :0.1300       Min.   :0.410   Min.   : 1.280   Min.   :0.4800   Min.   :1.270   Min.   : 278.0  
 1st Qu.:0.2700       1st Qu.:1.250   1st Qu.: 3.220   1st Qu.:0.7825   1st Qu.:1.938   1st Qu.: 500.5  
 Median :0.3400       Median :1.555   Median : 4.690   Median :0.9650   Median :2.780   Median : 673.5  
 Mean   :0.3619       Mean   :1.591   Mean   : 5.058   Mean   :0.9574   Mean   :2.612   Mean   : 746.9  
 3rd Qu.:0.4375       3rd Qu.:1.950   3rd Qu.: 6.200   3rd Qu.:1.1200   3rd Qu.:3.170   3rd Qu.: 985.0  
 Max.   :0.6600       Max.   :3.580   Max.   :13.000   Max.   :1.7100   Max.   :4.000   Max.   :1680.0  
 Customer_Segment
 Min.   :1.000   
 1st Qu.:1.000   
 Median :2.000   
 Mean   :1.938   
 3rd Qu.:3.000   
 Max.   :3.000   
> 
```

unique(is.na(dataset))   #outputs true if there is a missing value in any column



```
> unique(is.na(dataset))
     Alcohol Malic_Acid   Ash Ash_Alcanity Magnesium Total_Phenols Flavanoids Nonflavanoid_Phenols Proanthocyanins
[1,]   FALSE      FALSE FALSE        FALSE     FALSE         FALSE      FALSE                FALSE           FALSE
     Color_Intensity   Hue OD280 Proline Customer_Segment
[1,]           FALSE FALSE FALSE   FALSE            FALSE
> 
```

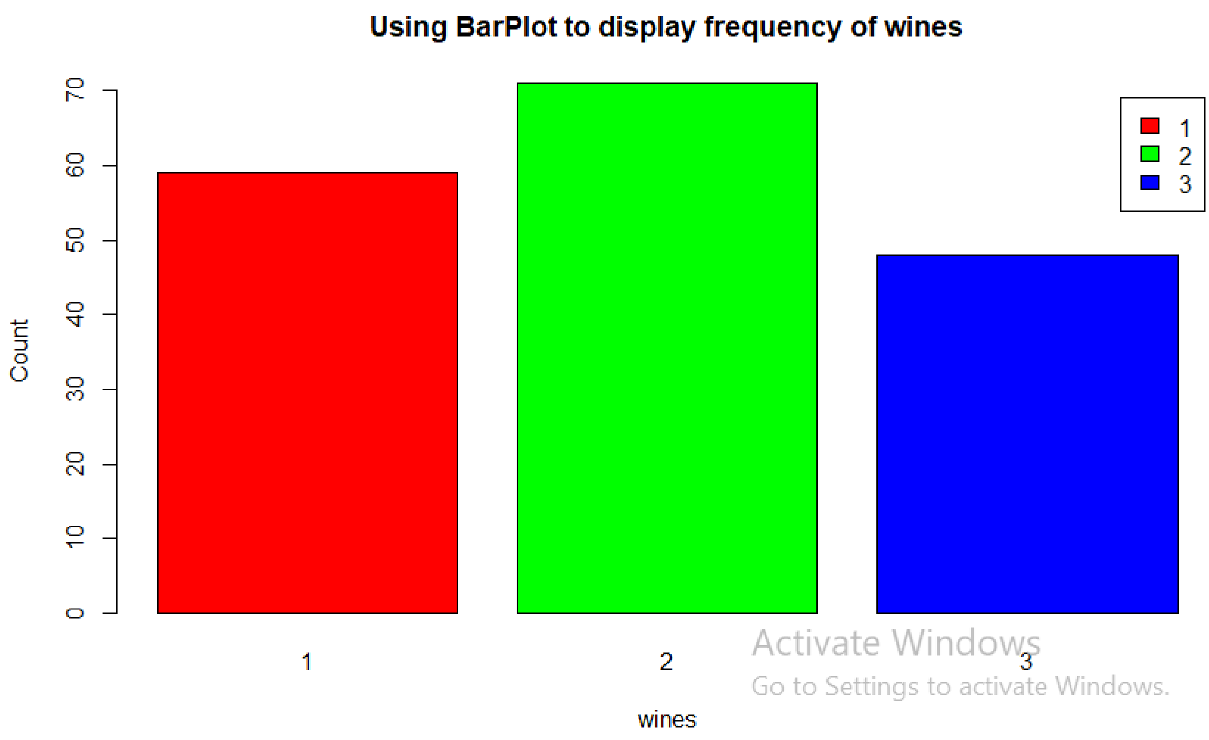
data_1=dataset %>% filter(dataset$Customer_Segment == 1)

data_2=dataset %>% filter(dataset$Customer_Segment == 2)

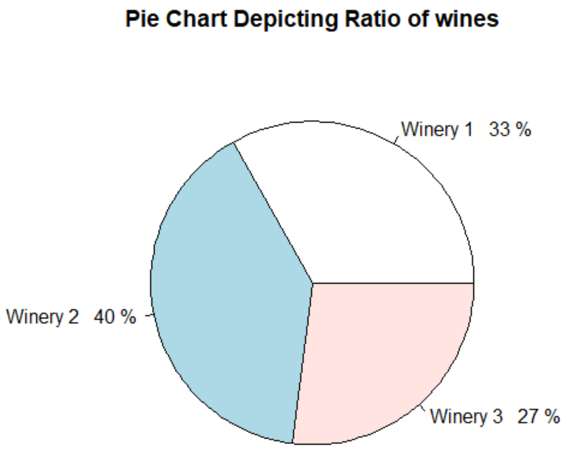
data_3=dataset %>% filter(dataset$Customer_Segment == 3)

this creates a subset of the winery products collected from each winery.

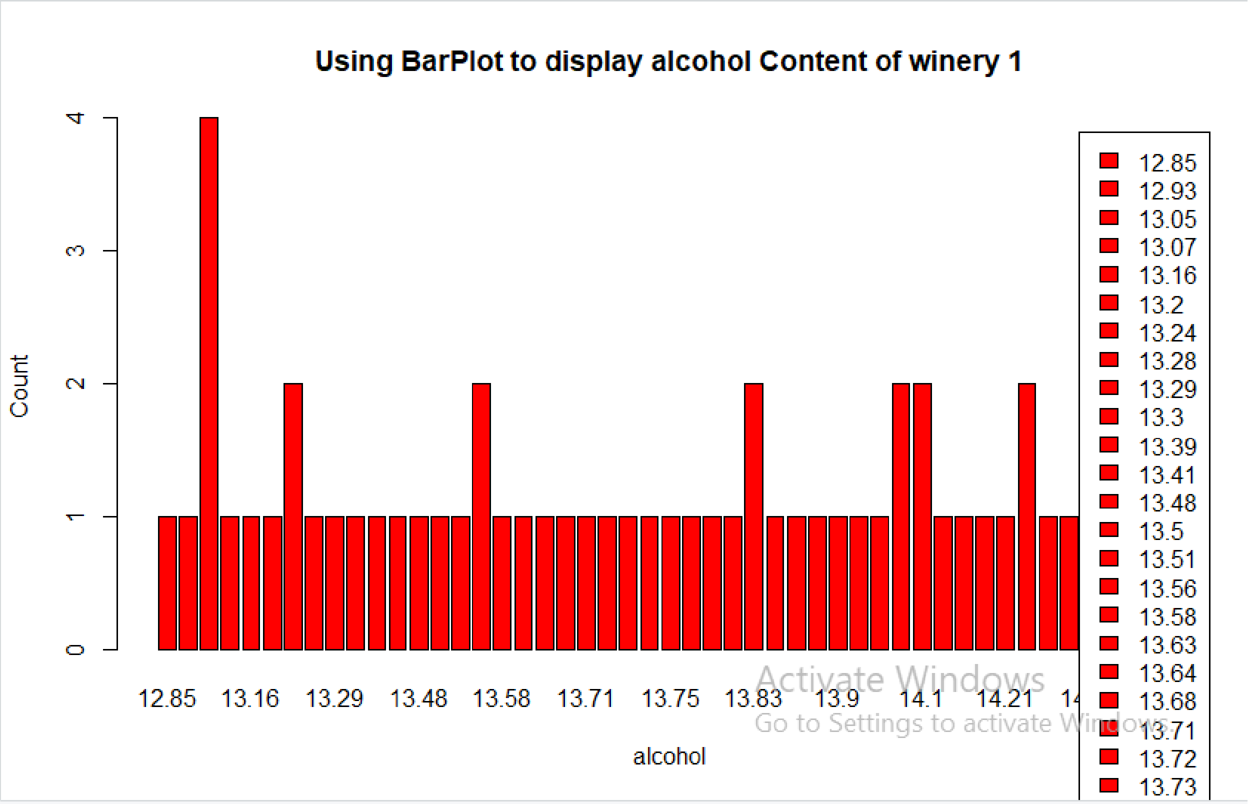
```
a=table(dataset$Customer_Segment)

barplot(a,main="Using BarPlot to display frequency of wines",

        ylab="Count",

        xlab="wines",

        col=rainbow(3),

        legend=rownames(a))
```



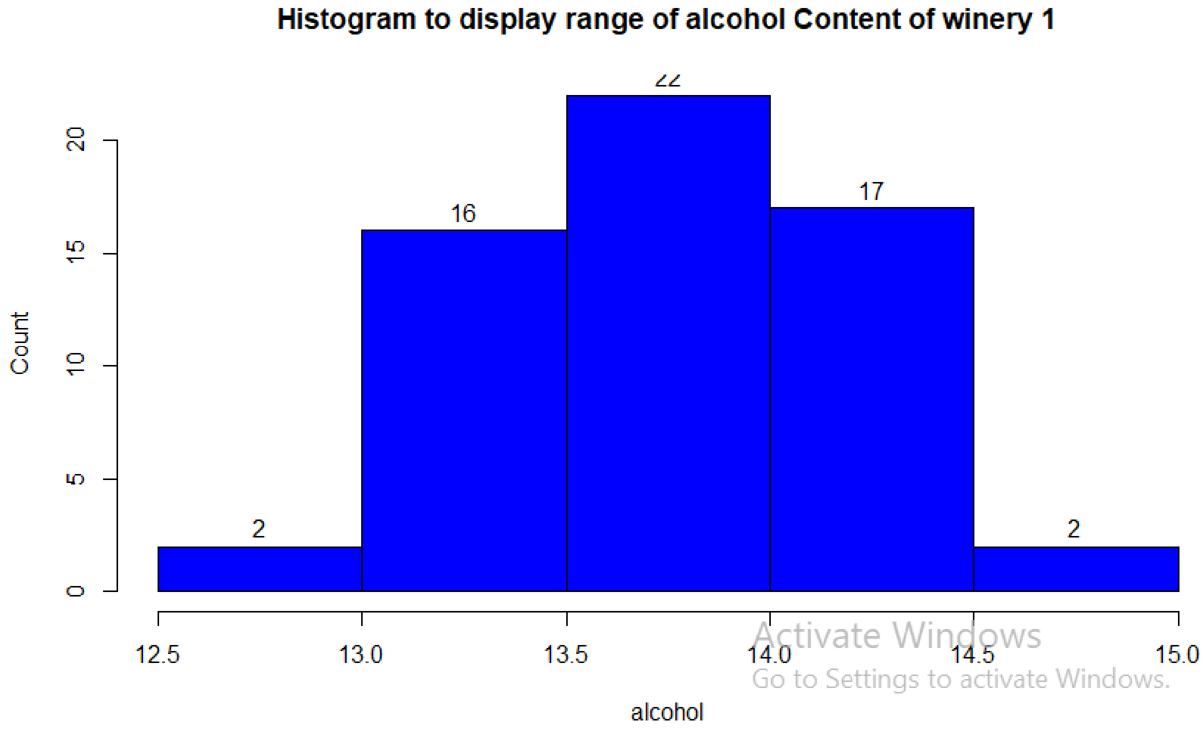
```
pct=round(a/sum(a)*100)

lbs=paste(c("Winery 1","Winery 2","Winery 3")," ",pct,"%",sep=" ")

library(plotrix)

pie(a,labels=lbs,main="Pie Chart Depicting Ratio of wines")
```



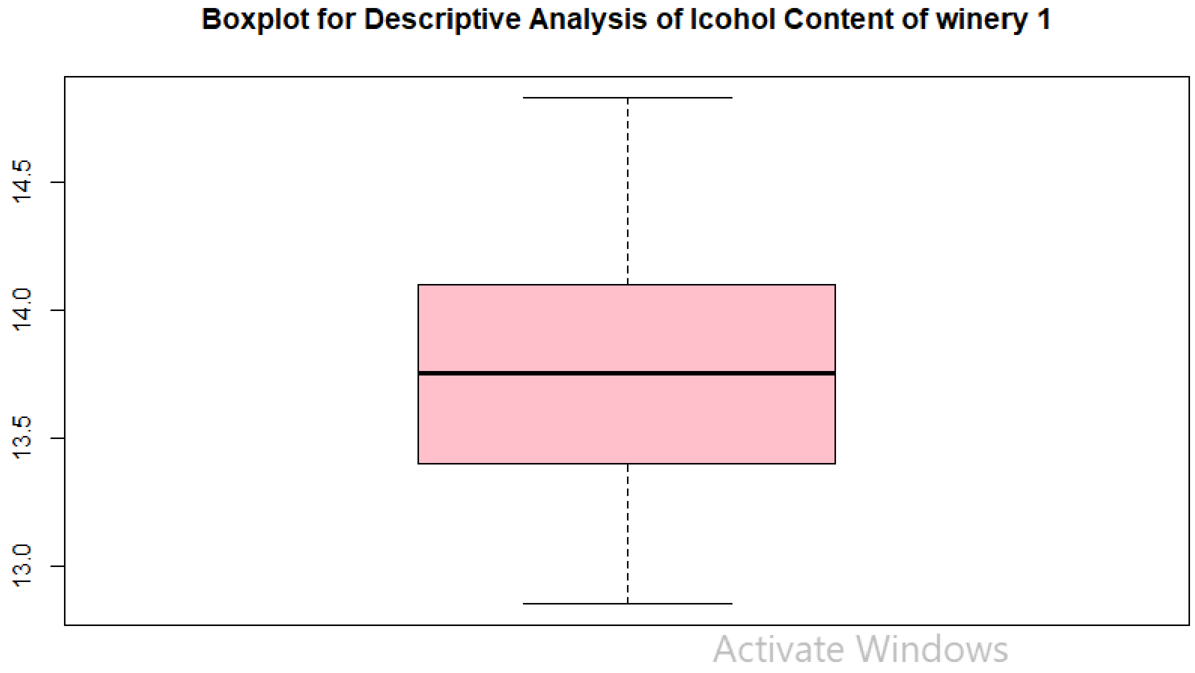
```
a=table(data_1$Alcohol)

barplot(a,main="Using BarPlot to display alcohol Content of winery 1",

    ylab="Count",

    xlab="alcohol",

    col='red',

    legend=rownames(a))
```



```
hist(data_1$Alcohol,

    col="blue",

    main="Histogram to display range of alcohol Content of winery 1",

    xlab="alcohol",

    ylab="Count",

    labels=TRUE)
```

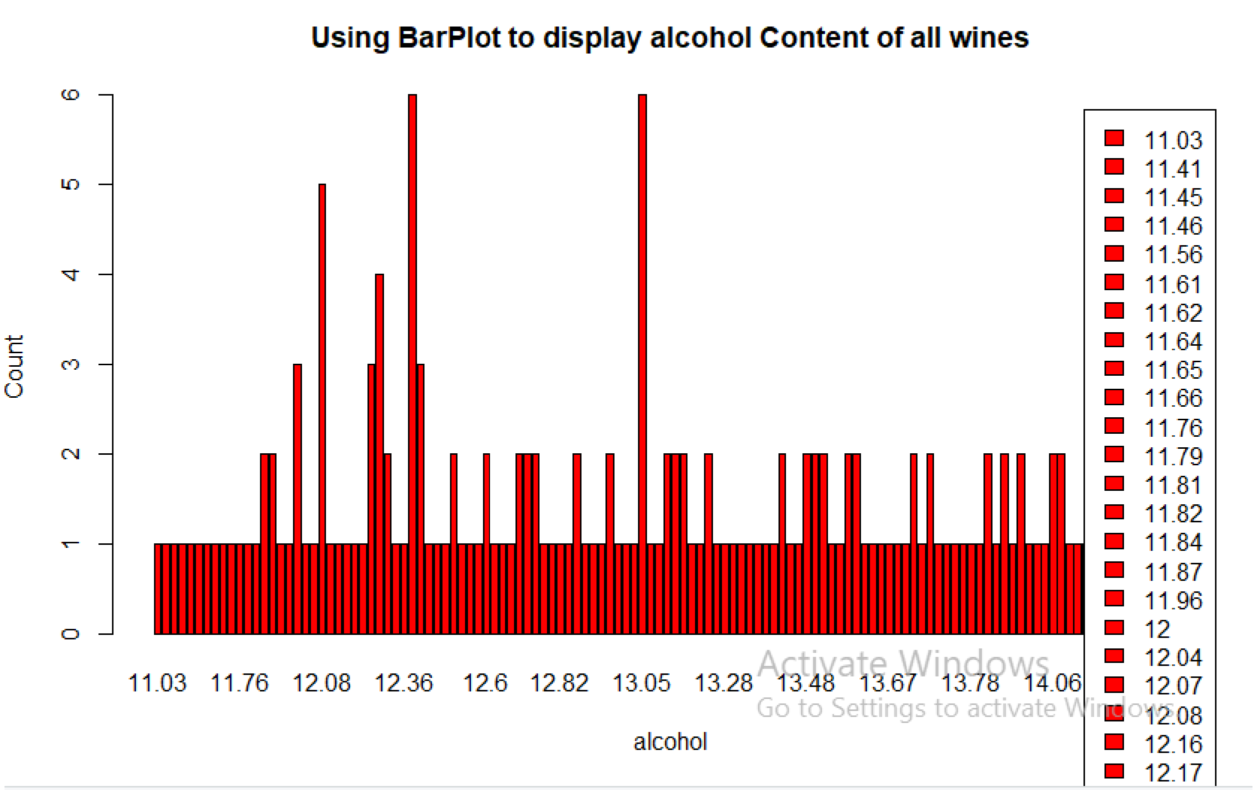


```
boxplot(data_1$Alcohol,

     col="pink",

     main="Boxplot for Descriptive Analysis of lcohol Content of winery 1")
```

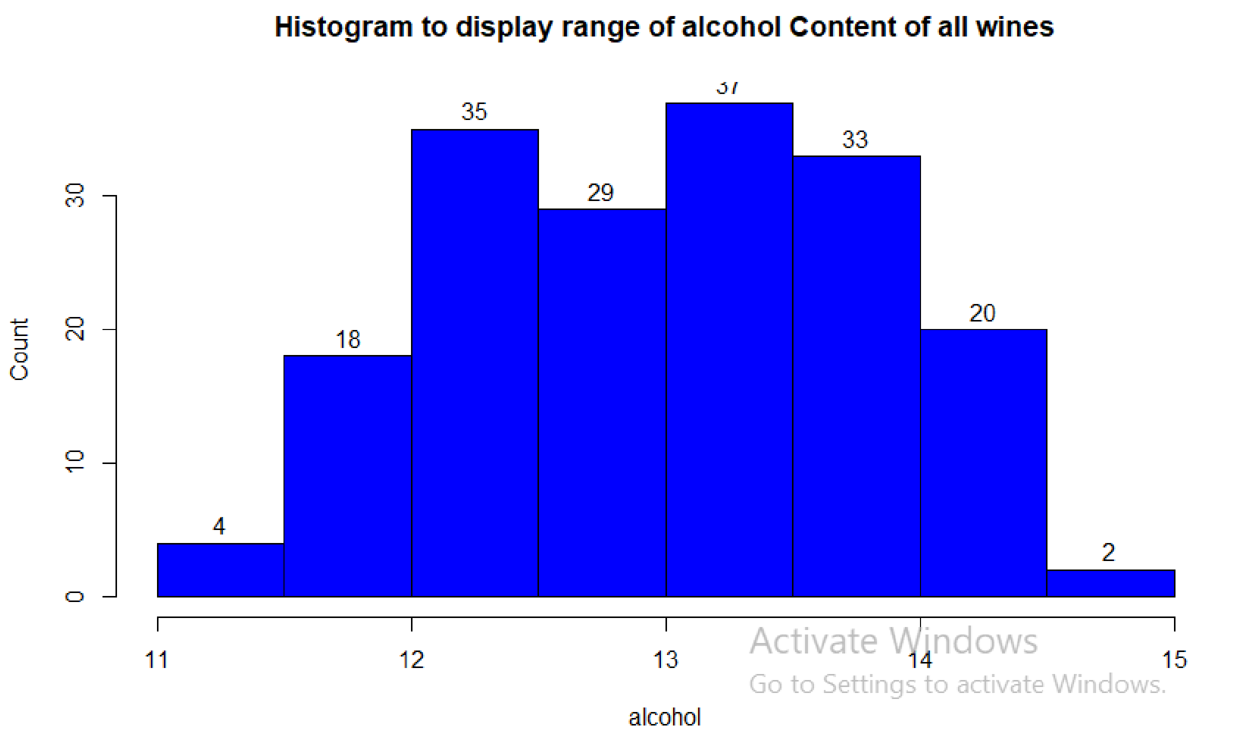


Similarly, we can plot other variables of our dataset and subset to analyze its value range and frequency.

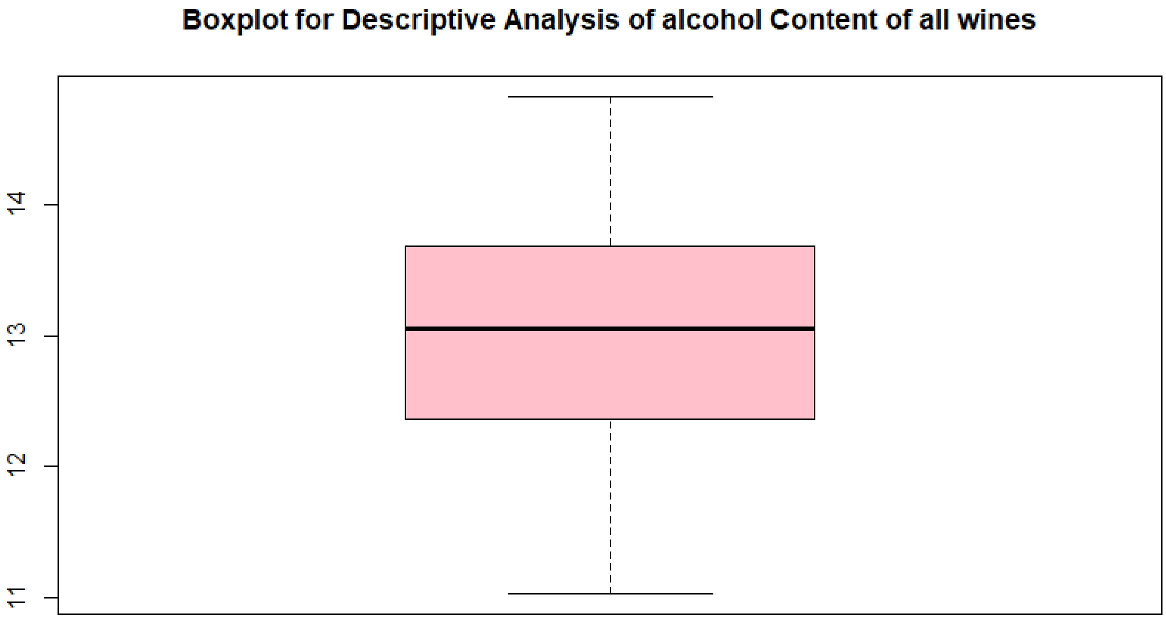
```
a=table(dataset$Alcohol)

barplot(a,main="Using BarPlot to display alcohol Content of all wines",

     ylab="Count",

     xlab="alcohol",

     col='red',

     legend=rownames(a))
```



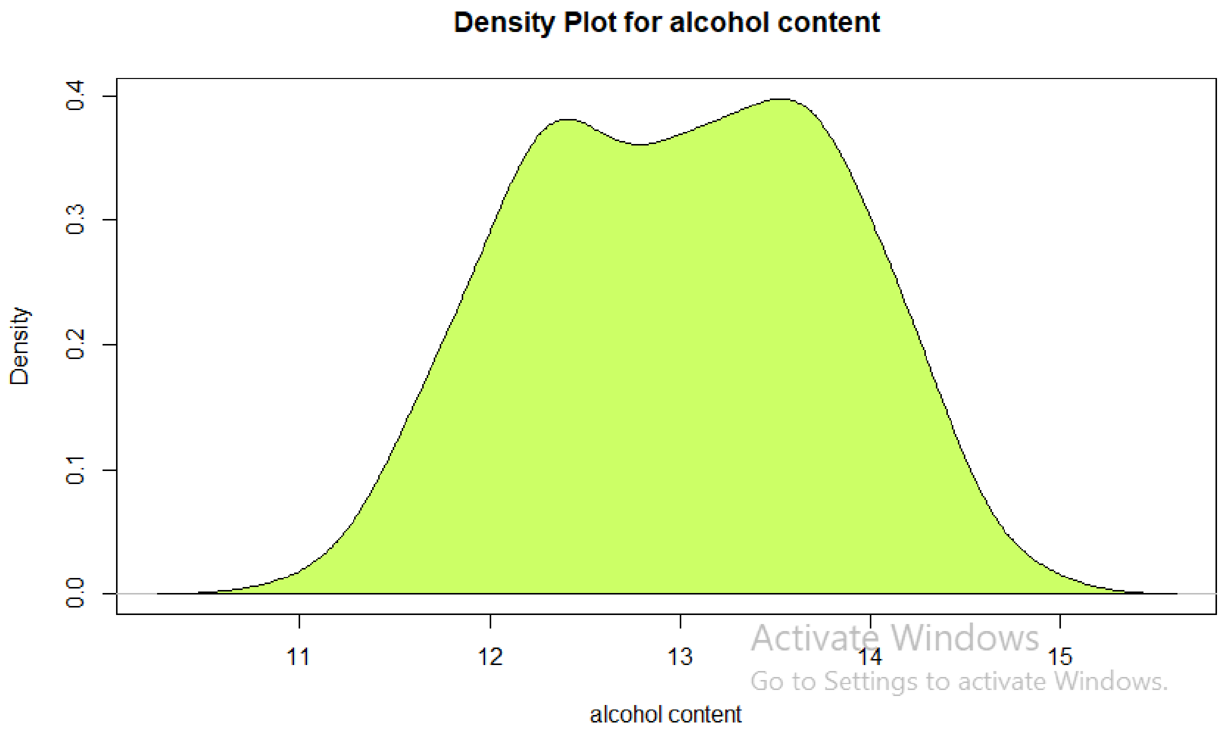
```
hist(dataset$Alcohol,
   col="blue",
   main="Histogram to display range of alcohol Content of all wines",
   xlab="alcohol",
   ylab="Count",
   labels=TRUE)
```



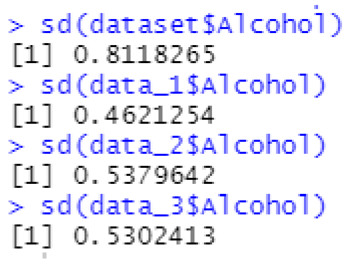
```
boxplot(dataset$Alcohol,
     col="pink",
     main="Boxplot for Descriptive Analysis of alcohol Content of all wines")
```



```
plot(density(dataset$Alcohol),
    main="Density Plot for alcohol content",
    xlab="alcohol content",ylab="Density")
polygon(density(dataset$Alcohol),col="#ccff66")
```



```
sd(dataset$Alcohol)     # computes the standard deviation
sd(data_1$Alcohol)
sd(data_2$Alcohol)
sd(data_3$Alcohol)
```

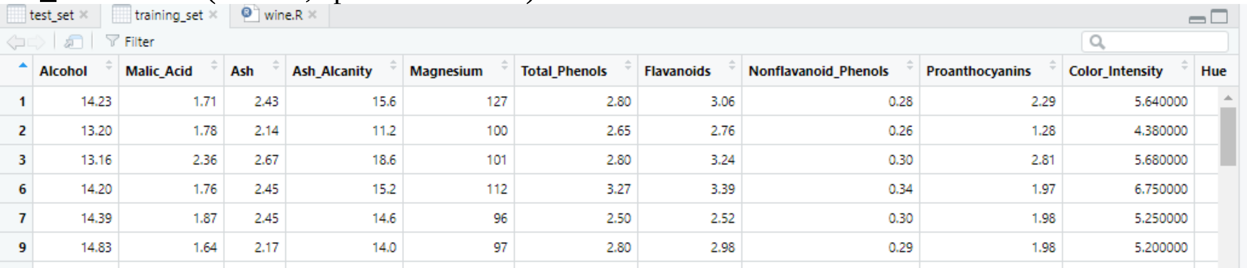


```
> sd(dataset$Alcohol)
[1] 0.8118265
> sd(data_1$Alcohol)
[1] 0.4621254
> sd(data_2$Alcohol)
[1] 0.5379642
> sd(data_3$Alcohol)
[1] 0.5302413
```

Part 2 – Prediction

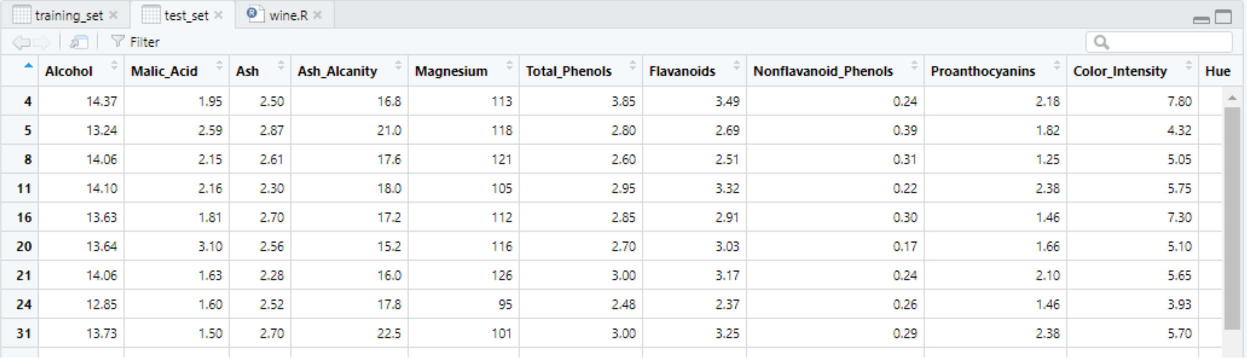
I will use principal component analysis (PCA) on our dataset. I am using PCA for predicting values because goal of PCA is to identify and detect correlation between variables, if there's a strong correlation and it’s found, then you could reduce the dimensionality, which really what PCA is intended for.

```
library(caTools)

set.seed(123)

split = sample.split(dataset$Customer_Segment, SplitRatio = 0.8) #spliting the dataset

training_set = subset(dataset, split == TRUE)

test_set = subset(dataset, split == FALSE)
```



	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue
1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.640000	
2	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.380000	
3	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.680000	
6	14.20	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.750000	
7	14.39	1.87	2.45	14.6	96	2.50	2.52	0.30	1.98	5.250000	
9	14.83	1.64	2.17	14.0	97	2.80	2.98	0.29	1.98	5.200000	

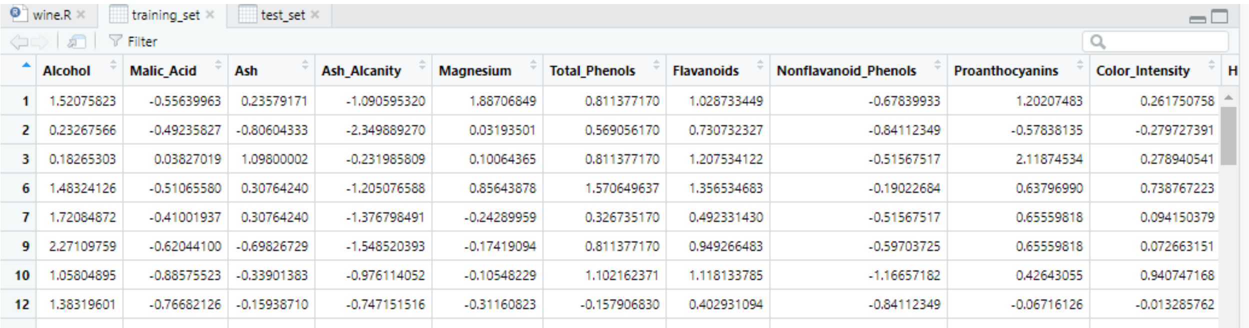
Training set



	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue
4	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	
5	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	
8	14.06	2.15	2.61	17.6	121	2.60	2.51	0.31	1.25	5.05	
11	14.10	2.16	2.30	18.0	105	2.95	3.32	0.22	2.38	5.75	
16	13.63	1.81	2.70	17.2	112	2.85	2.91	0.30	1.46	7.30	
20	13.64	3.10	2.56	15.2	116	2.70	3.03	0.17	1.66	5.10	
21	14.06	1.63	2.28	16.0	126	3.00	3.17	0.24	2.10	5.65	
24	12.85	1.60	2.52	17.8	95	2.48	2.37	0.26	1.46	3.93	
31	13.73	1.50	2.70	22.5	101	3.00	3.25	0.29	2.38	5.70	

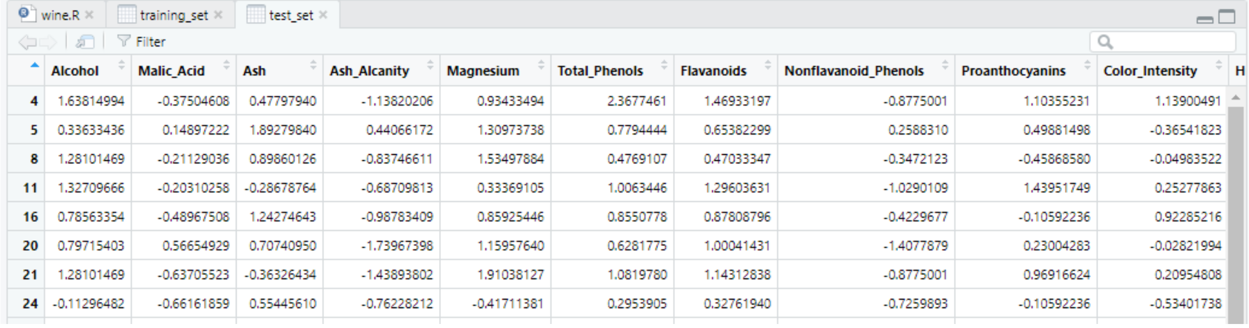
Test set

```
# Feature Scaling

training_set[-14] = scale(training_set[-14])

test_set[-14] = scale(test_set[-14])
```



	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	H
1	1.52075823	-0.55639963	0.23579171	-1.090595320	1.88706849	0.811377170	1.028733449	-0.67839933	1.20207483	0.261750758	
2	0.23267566	-0.49235827	-0.80604333	-2.349889270	0.03193501	0.569056170	0.730732327	-0.84112349	-0.57838135	-0.279727391	
3	0.18265303	0.03827019	1.09800002	-0.231985809	0.10064365	0.811377170	1.207534122	-0.51567517	2.11874534	0.278940541	
6	1.48324126	-0.51065580	0.30764240	-1.205076588	0.85643878	1.570649637	1.356534683	-0.19022684	0.63796990	0.738767223	
7	1.72084872	-0.41001937	0.30764240	-1.376798491	-0.24289959	0.326735170	0.492331430	-0.51567517	0.65559818	0.094150379	
9	2.27109759	-0.62044100	-0.69826729	-1.548520393	-0.17419094	0.811377170	0.949266483	-0.59703725	0.65559818	0.072663151	
10	1.05804895	-0.88575523	-0.33901383	-0.976114052	-0.10548229	1.102162371	1.118133785	-1.16657182	0.42643055	0.940747168	
12	1.38319601	-0.76682126	-0.15938710	-0.747151516	-0.31160823	-0.157906830	0.402931094	-0.84112349	-0.06716126	-0.013285762	

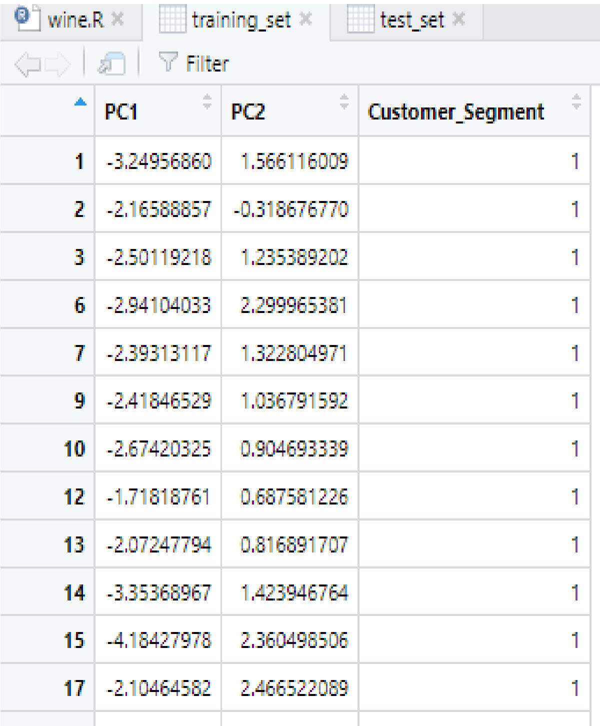
Training set after scaling



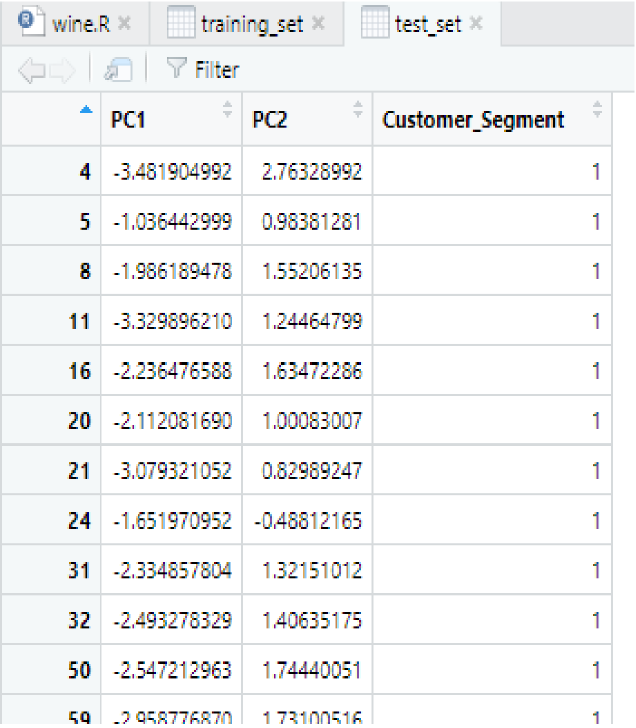
	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	H
4	1.63814994	-0.37504608	0.47797940	-1.13820206	0.93433494	2.3677461	1.46933197	-0.8775001	1.10355231	1.13900491	
5	0.33633436	0.14897222	1.89279840	0.44066172	1.30973738	0.7794444	0.65382299	0.2588310	0.49881498	-0.36541823	
8	1.28101469	-0.21129036	0.89860126	-0.83746611	1.53497884	0.4769107	0.47033347	-0.3472123	-0.45868580	-0.04983522	
11	1.32709666	-0.20310258	-0.28678764	-0.68709813	0.33369105	1.0063446	1.29603631	-1.0290109	1.43951749	0.25277863	
16	0.78563354	-0.48967508	1.24274643	-0.98783409	0.85925446	0.8550778	0.87808796	-0.4229677	-0.10592236	0.92285216	
20	0.79715403	0.56654929	0.70740950	-1.73967398	1.15957640	0.6281775	1.00041431	-1.4077879	0.23004283	-0.02821994	
21	1.28101469	-0.63705523	-0.36326434	-1.43893802	1.91038127	1.0819780	1.14312838	-0.8775001	0.96916624	0.20954808	
24	-0.11296482	-0.66161859	0.55445610	-0.76228212	-0.41711381	0.2953905	0.32761940	-0.7259893	-0.10592236	-0.53401738	

Test set after scaling

```
library(caret)

library(e1071)

pca = preProcess(x = training_set[-14], method = 'pca', pcaComp = 2)  #training the model

training_set = predict(pca, training_set)

training_set = training_set[c(2, 3, 1)]

test_set = predict(pca, test_set)

test_set = test_set[c(2, 3, 1)]
```



	PC1	PC2	Customer_Segment
1	-3.24956860	1.566116009	1
2	-2.16588857	-0.318676770	1
3	-2.50119218	1.235389202	1
6	-2.94104033	2.299965381	1
7	-2.39313117	1.322804971	1
9	-2.41846529	1.036791592	1
10	-2.67420325	0.904693339	1
12	-1.71818761	0.687581226	1
13	-2.07247794	0.816891707	1
14	-3.35368967	1.423946764	1
15	-4.18427978	2.360498506	1
17	-2.10464582	2.466522089	1

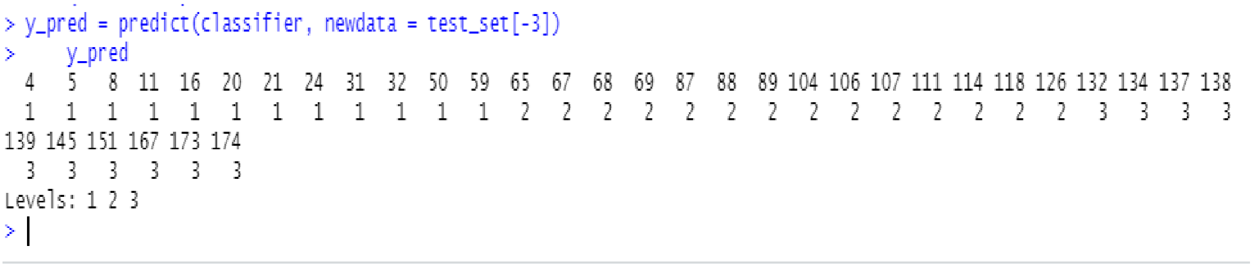


	PC1	PC2	Customer_Segment
4	-3.481904992	2.76328992	1
5	-1.036442999	0.98381281	1
8	-1.986189478	1.55206135	1
11	-3.329896210	1.24464799	1
16	-2.236476588	1.63472286	1
20	-2.112081690	1.00083007	1
21	-3.079321052	0.82989247	1
24	-1.651970952	-0.48812165	1
31	-2.334857804	1.32151012	1
32	-2.493278329	1.40635175	1
50	-2.547212963	1.74440051	1
59	-2.958776870	1.73100516	1

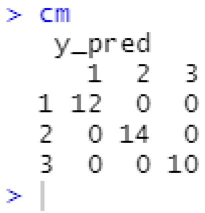
Training and Test set after feature extraction

```
# Fitting SVM to the Training set (I chose svm model)

classifier = svm(formula = Customer_Segment ~ .,

             data = training_set,

             type = 'C-classification',

             kernel = 'linear')
```

```
# Predicting the Test set results

y_pred = predict(classifier, newdata = test_set[-3])
```

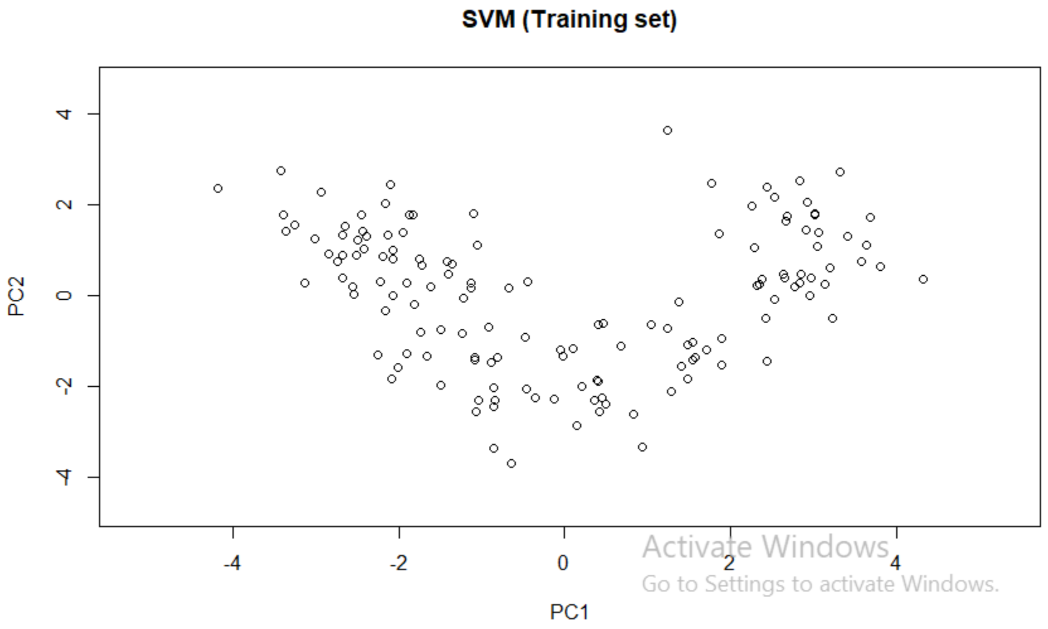


```
> y_pred = predict(classifier, newdata = test_set[-3])
>     y_pred
  4   5   8  11  16  20  21  24  31  32  50  59  65  67  68  69  87  88  89 104 106 107 111 114 118 126 132 134 137 138 
  1   1   1   1   1   1   1   1   1   1   1   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2   3   3   3   3 
139 145 151 167 173 174 
  3   3   3   3   3   3 
Levels: 1 2 3
> |
# Making the Confusion Matrix

cm = table(test_set[, 3], y_pred)

cm
```

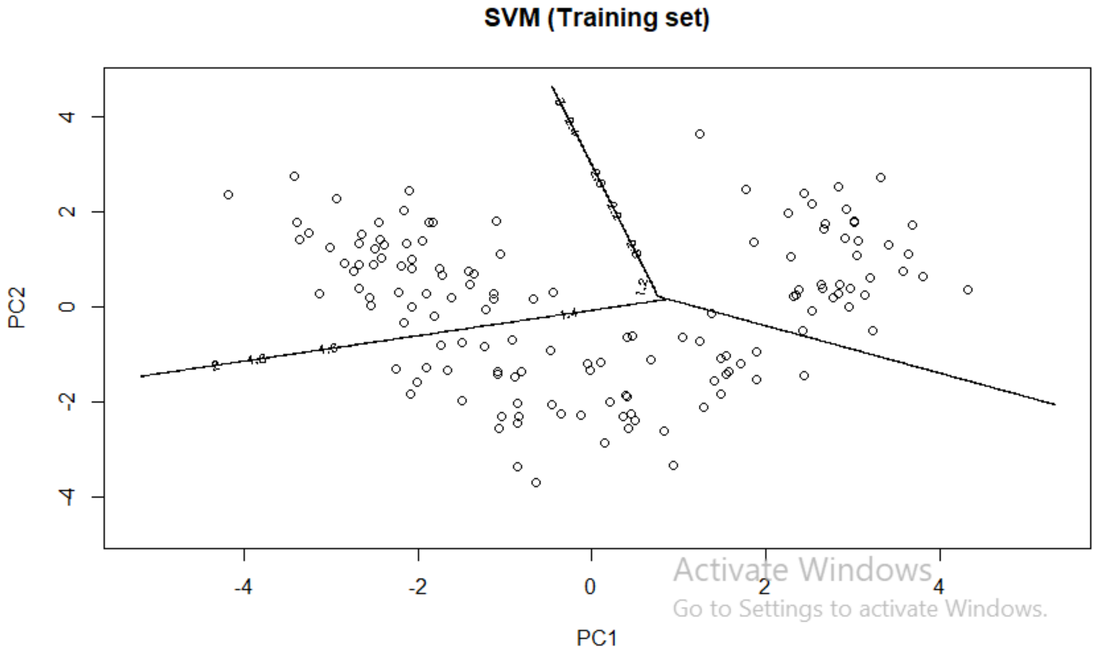


```
> cm
   y_pred
     1  2  3
  1 12  0  0
  2  0 14  0
  3  0  0 10
> 
```

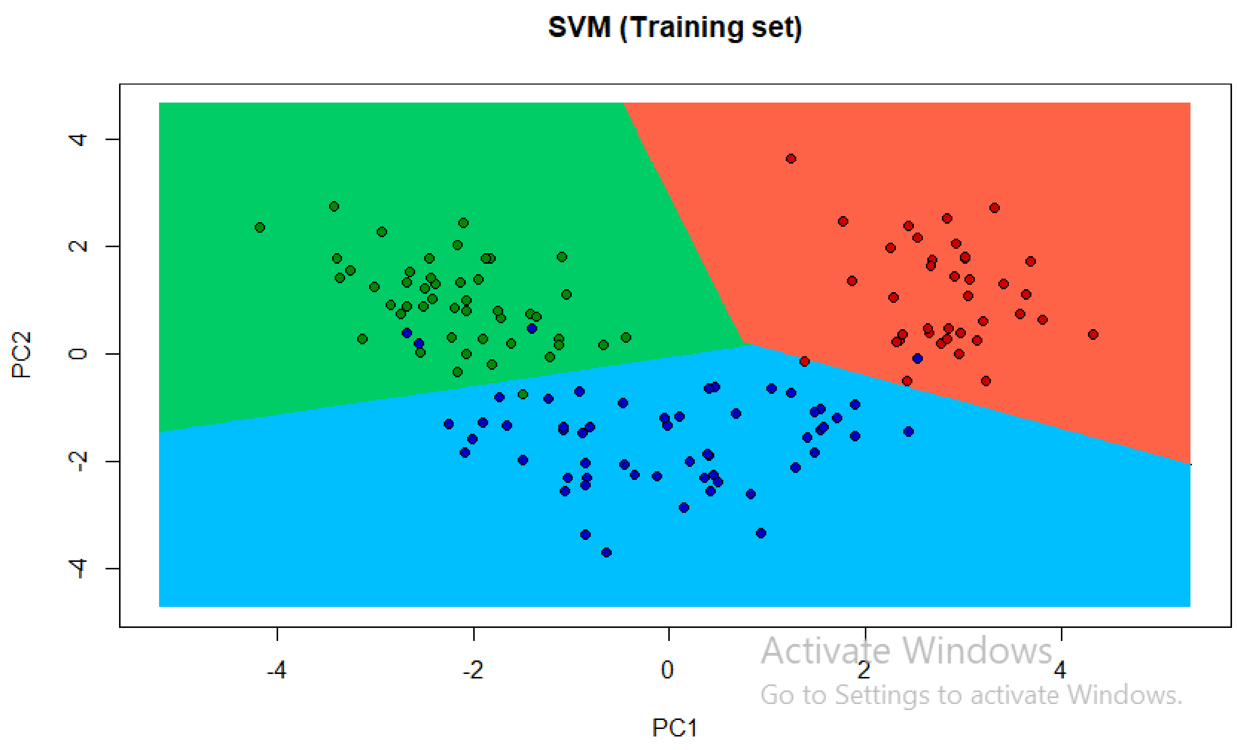
```
# Visualising the Training set results

library(ElemStatLearn)

set = training_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)

X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)

grid_set = expand.grid(X1, X2)

colnames(grid_set) = c('PC1', 'PC2')

y_grid = predict(classifier, newdata = grid_set)


plot(set[, -3],

     main = 'SVM (Training set)',

     xlab = 'PC1', ylab = 'PC2',

     xlim = range(X1), ylim = range(X2))
```



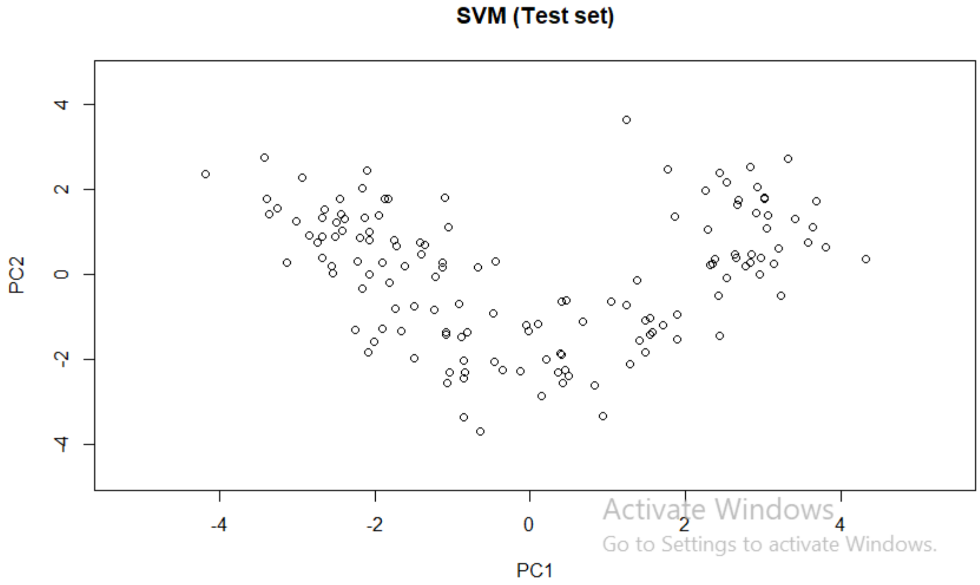
```
contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add = TRUE)
```



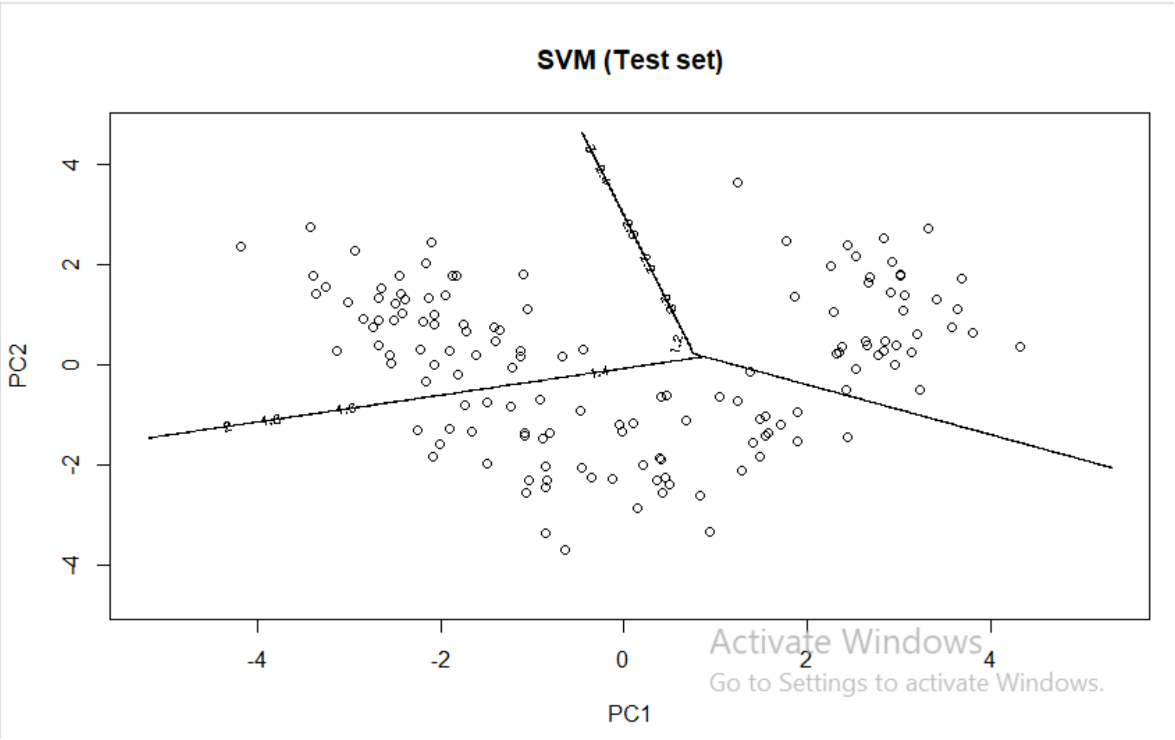
```
points(grid_set,
       pch = '.',
       col = ifelse(y_grid == 2, 'deepskyblue', ifelse(y_grid == 1, 'springgreen3', 'tomato')))
points(set, pch = 21,
    bg = ifelse(set[, 3] == 2, 'blue3', ifelse(set[, 3] == 1, 'green4', 'red3')))
```



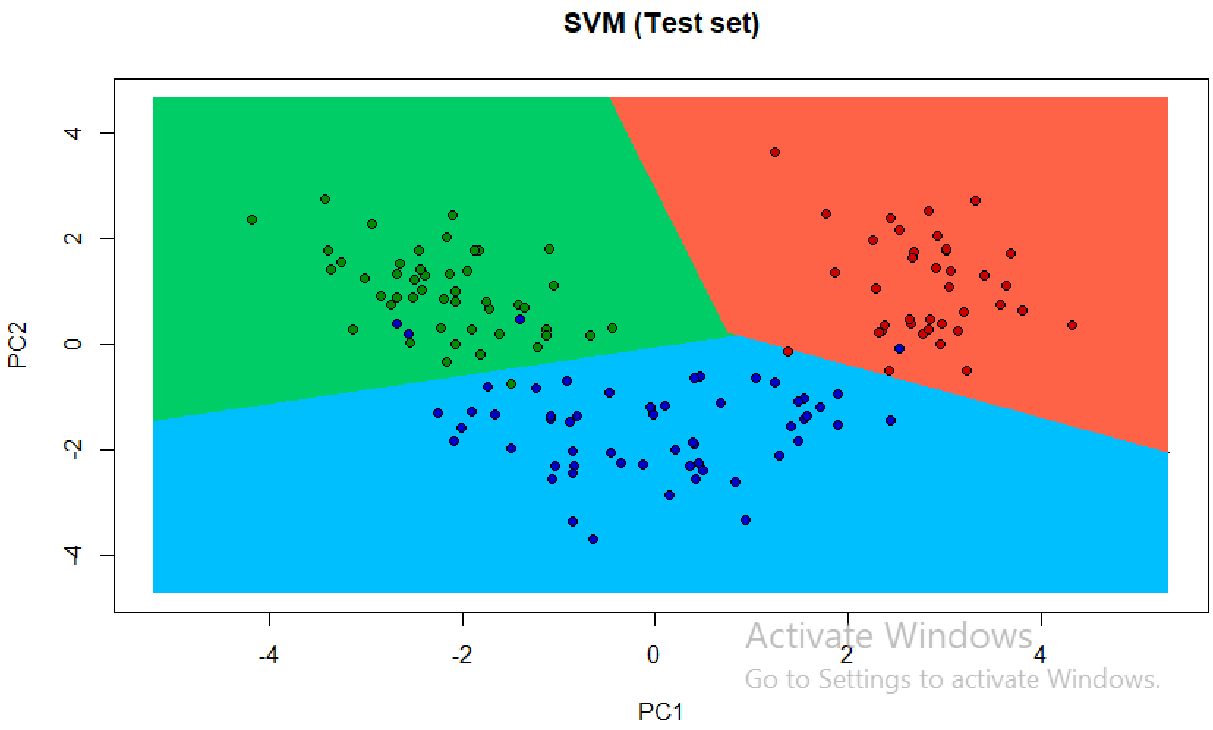
```
# Visualising the Test set results

library(ElemStatLearn)

set = test_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)

X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)

grid_set = expand.grid(X1, X2)

colnames(grid_set) = c('PC1', 'PC2')

y_grid = predict(classifier, newdata = grid_set)


plot(set[, -3], main = 'SVM (Test set)',

     xlab = 'PC1', ylab = 'PC2',

     xlim = range(X1), ylim = range(X2))
```

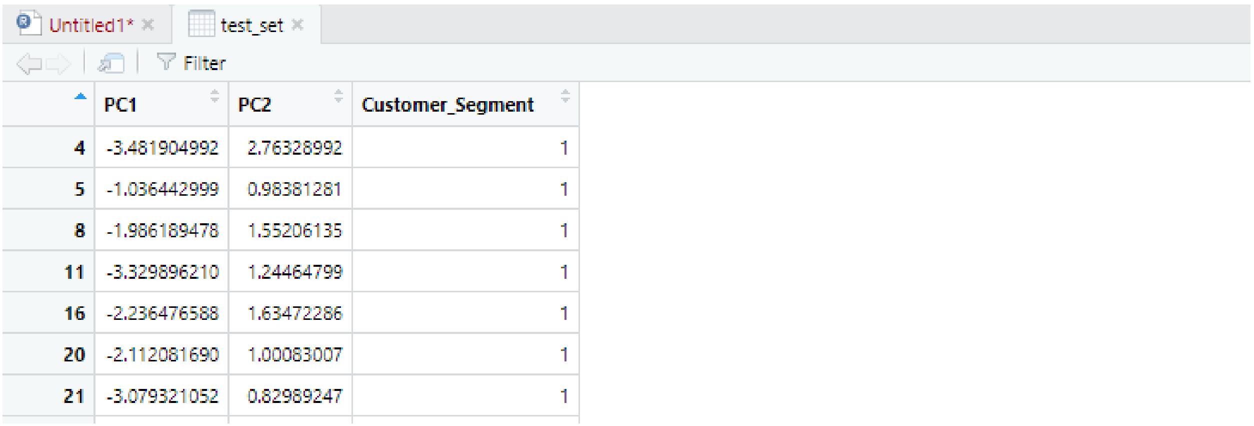


```
contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add = TRUE)
```



```
points(grid_set,
     pch = '.',
     col = ifelse(y_grid == 2, 'deepskyblue', ifelse(y_grid == 1, 'springgreen3', 'tomato')))
points(set,
     pch = 21,

     bg = ifelse(set[, 3] == 2, 'blue3', ifelse(set[, 3] == 1, 'green4', 'red3')))
```





	PC1	PC2	Customer_Segment
4	-3.481904992	2.76328992	1
5	-1.036442999	0.98381281	1
8	-1.986189478	1.55206135	1
11	-3.329896210	1.24464799	1
16	-2.236476588	1.63472286	1
20	-2.112081690	1.00083007	1
21	-3.079321052	0.82989247	1

Test Set Predictions

SUMMARY

In this data science project, we worked on a highly distributed Multivariate dataset. With help of various functions we analyzed our dataset’s component, how the alcohol level and other measures of wines from each winery re distributed, their frequency, count etc. due to high correlation between the dataset, we used PCA model to reduce their dimensionality and then plot its graph and at last predicted the values. I used SVM model for classification and it has given me an accuracy of 100 % on the test set and it could be seen in confusion matrix.

I can use other classification models like naïve bayes alse and similarly other dimension reduction models like lda or kernel PCA instead of PCA.

.