Basic exploration of Raisin Data set

Y. A. R. Olivera Under guidance of Prof. R. Srivastava

2022-15-07

Description

Some image attributes of Kecimen and Besni raisin varieties grown in Turkey were downloaded from ('https://archive.ics.uci.edu/ml/datasets/Raisin+Dataset'). The total number of observations/records is 900 (sample size) with equal number of both varities. The images were processed and 7 morphological features were listed in the data base. The article written by Karimi, N., Arabhosseini, A., Kianmehr, M. H., and Khazaei, J. ("Modelling of raisin berries by some physical and statistical characteristics," Int. Agrophys., vol. 25(2), pp. 141-147, April 2011.) considers this data first to the best of our knowledge. Since then several analysis on this data set have been performed. Recently, "Classification of Raisin Grains Using Machine Vision and Artificial Intelligence Methods" by Ilkay CINAR, Murat KOKLUb and Sakir TASDEMIR (published at Gazi Mühendislik Bilimleri Dergisi 2020, 6(3): 200-209 Araştırma Makalesi/Research Article https://dergipark.org.tr/gmbd.) is also a good read.

Our objective is to illustrate the linear discrimant analysis (LDA) and support vector machine (SVM) on this data to students. These classification task on this data may help students to undertand discrimation technique. In this document, we explore the data set and perform basic exploration.

Attribute Information:

Variable Name	Description	Alias Name
ar	Area gives the number of pixels within the boundaries of the raisin	x1
peri	Perimeter measures the environment by calculating the distance between the boundaries of the raisin and the pixels around it	x2
majorAL	Major Axis Length gives the length of the main axis, which is the longest line that can be drawn on the raisin	x3
minorAL	Minor Axis Length gives the length of the small axis, which is the shortest line that can be drawn on the raisin	x4
ecc	Eccentricity gives a measure of the eccentricity of the ellipse, which has the same moments as raisins	x5
conArea	Convex Area gives the number of pixels of the smallest convex shell of the region formed by the raisin	x6
extent	Gives the ratio of the region formed by the raisin to the total pixels in the bounding box	x7
Class	Kecimen and Besni raisin.	y

I-) Loading Libraries

```
options(warn = -1)
suppressMessages(library(readxl)) # Read Excel file
suppressMessages(library(moments)) # Moments computation
suppressMessages(library(corrplot)) # Correlation plot
suppressMessages(library(ggcorrplot)) # Useful package for analysis
suppressMessages(library(caret)) # Useful package for analysis
suppressMessages(library(MVN)) # For Multivariate Normality test
suppressMessages(library(tidyverse)) #  Useful package for analysis
suppressMessages(library(GGally)) #  Extends ggplot2 to a new dimension
suppressMessages(library(cowplot))# Useful package for analysis
suppressMessages(library(psych)) # Useful package for analysis
suppressMessages(library(data.table)) # change the columns name
suppressMessages(library(lemon)) # Display DataFrame data as table

knit_print.data.frame <- lemon_print #
```

II-) Loading Data

```
raisin <- read_xlsx("Raisin.xlsx")
raisin$class <- factor(raisin$class)
colnames(raisin)<- c("x1","x3","x4","x5","x6","x7","x2","class")
raisin<- raisin[,c(1,7,3,4,5,6,2,8)]

class_Kecimen <- subset(raisin,class=='Kecimen')
class_Besni <- subset(raisin,class=='Besni')
```

Summary Statistics

we create a function named *calculate_stats()* that calculates manual summary statistics (mean, standard deviation, min, max, quartiles, skewness, kurtosis) for a given column of the data. Summary_statistics() that takes a dataset (each class data) and a class_name as inputs. Inside the summary_statistics() function, It calculates the summary statistics for each column in the dataset data by applying the calculate_stats() function using *pivot_longer()*, *group_by()*, *summarise()*, and *pivot_wider()* functions.

```
# Function to calculate the manual summary statistics for a given column and class
calculate_stats <- function(column_data) {
  mean_val <- round(mean(column_data), 2)
  sd_val <- round(sd(column_data), 2)
  min_val <- round(min(column_data), 2)
  max_val <- round(max(column_data), 2)
  q1_val <- round(quantile(column_data, probs = 0.25), 2)
  median_val <- round(median(column_data), 2)
  q3_val <- round(quantile(column_data, probs = 0.75), 2)
  skewness_val <- round(skew(column_data), 2)
  kurtosis_val <- round(kurtosi(column_data), 2)

  stats <- c(mean_val, sd_val, min_val, max_val, q1_val, median_val,
```

```r
              q3_val, skewness_val, kurtosis_val)
  return(stats)
}

# Create a DataFrame with the manual summary statistics
summary_statistics <- function(data,class_name){


  summary_df <- data %>%
    pivot_longer(cols = starts_with("x"), names_to = "Column") %>%
    group_by(Column) %>%
    summarise(Value = list(calculate_stats(value))) %>%
    pivot_wider(names_from = Column, values_from = Value) %>%
    unnest(cols = c(x1, x2, x3, x4, x5, x6, x7))

  # Set the row names (index) as the desired statistics
  statistics_names <- c("mean", "sd", "min", "max", "Quartile 1", "median", "Quartile 3",
                        "skewness", "kurtosis")
  summary_df$Statistics<- statistics_names

  #data.frame(summary_df)
  row.names(summary_df)<- summary_df$Statistics
  old <- c("x1" , "x2" , "x3" , "x4",  "x5" , "x6" , "x7"  ,"Statistics")
  new<- c(paste0("x1_", class_name) , paste0("x2_", class_name) , paste0("x3_",class_name),
          paste0("x4_", class_name),  paste0("x5_", class_name) , paste0("x6_",class_name),
          paste0("x7_", class_name)  ,"Statistics")
  setnames(summary_df, old = old, new = new)
  summary_df<-summary_df[,c(8,1,2,3,4,5,6,7)]

  return(summary_df)
}
```

```r
summary_statistics(class_Kecimen,"Kecimen")
```

Table 2: Summary Statistics of class `Kecimen`.

Statistics	x1_Kecimen	x2_Kecimen	x3_Kecimen	x4_Kecimen	x5_Kecimen	x6_Kecimen	x7_Kecimen
mean	63413.47	983.69	352.86	229.35	0.74	65696.36	0.71
sd	17727.77	150.31	59.61	34.06	0.09	19005.89	0.04
min	25387.00	619.07	225.63	143.71	0.35	26139.00	0.45
max	180898.00	2253.56	843.96	326.90	0.92	221396.00	0.84
Quartile 1	50466.25	885.15	313.54	205.20	0.70	52115.00	0.68
median	61420.00	977.93	350.23	228.62	0.77	63826.50	0.71
Quartile 3	75153.00	1073.22	390.36	250.79	0.80	77504.25	0.74
skewness	1.00	2.17	1.74	0.24	-1.33	1.81	-0.63
kurtosis	3.89	16.35	12.02	-0.15	2.04	11.17	2.50

```r
summary_statistics(class_Besni,"Besni")
```

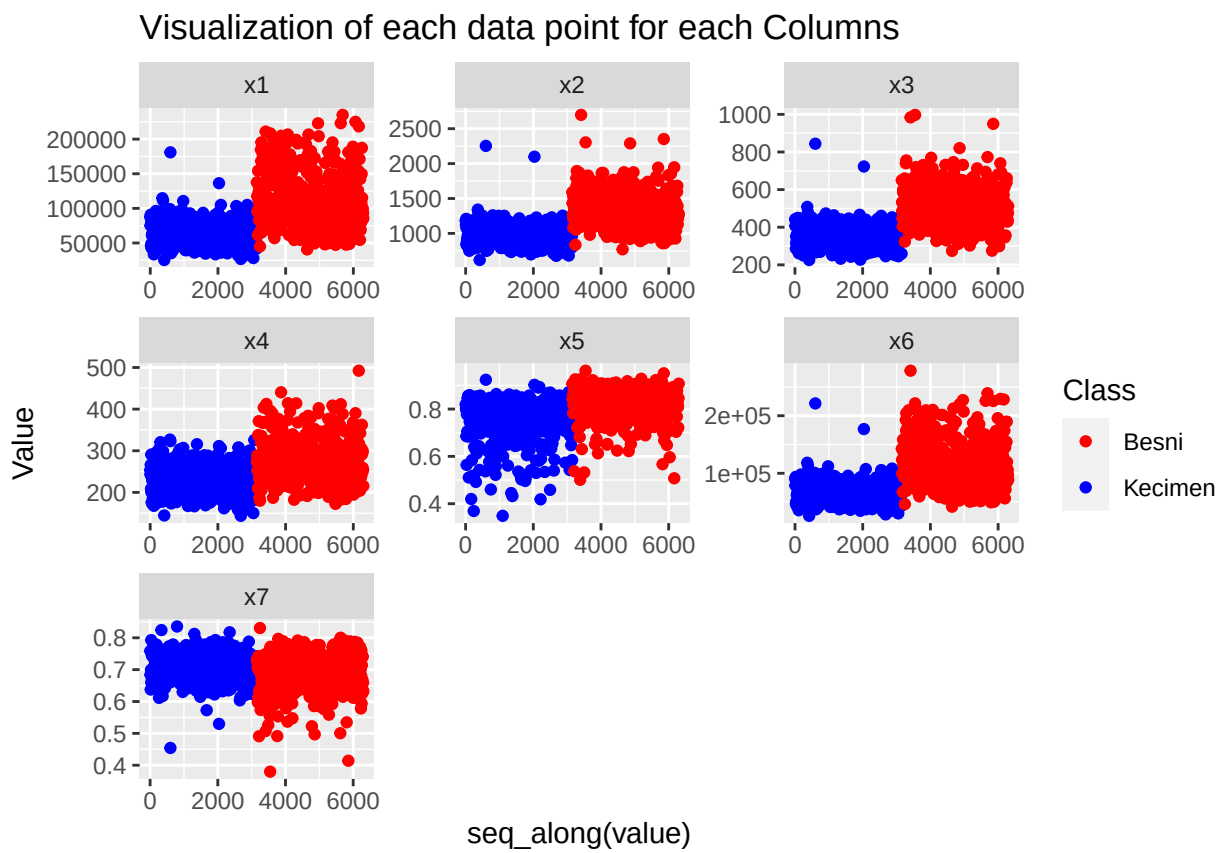
Table 3: Summary Statistics of class `Besni`.

Statistics	x1_Besni	x2_Besni	x3_Besni	x4_Besni	x5_Besni	x6_Besni	x7_Besni
mean	112194.79	1348.13	509.00	279.62	0.82	116675.82	0.69
sd	39229.90	246.80	105.77	50.76	0.07	40797.07	0.06
min	40702.00	771.80	274.17	172.51	0.50	41924.00	0.38
max	235047.00	2697.75	997.29	492.28	0.96	278217.00	0.83
Quartile 1	83598.50	1182.35	436.55	242.64	0.79	87668.00	0.66
median	104426.50	1305.80	493.19	273.37	0.83	108062.50	0.70
Quartile 3	135683.25	1510.48	582.39	308.31	0.87	139779.75	0.73
skewness	0.73	0.89	0.78	0.64	-1.40	0.83	-1.13
kurtosis	-0.04	2.38	1.66	0.45	3.25	0.36	2.44

Visualize each data point

this code efficiently generates a scatter plots for each column in the 'raisin' dataset, with color-coded points based on the "class" variable.

```
# Gather all the columns (x1 to x7) into a single column 'variable' and 'value'
data_gathered <- raisin %>%
  pivot_longer(cols = starts_with("x"), names_to = "variable", values_to = "value")

# Create scatter plots in a 3x3 grid for each column
ggplot(data_gathered, aes(x = seq_along(value), y = value, color = class)) +
  geom_point() +
  # Arrange plots in a 3x3 grid with free scales
  facet_wrap(~ variable, nrow = 3, ncol = 3, scales = "free") +
  labs(title = "Visualization of each data point for each Columns",
       y = "Value",
       color = "Class") +
   # Customize the colors for the two classes (e.g., red and blue)
  scale_color_manual(values = c("red", "blue"))
```



IV-)Data exploration

Note that all the columns of the data matrix are continuous except the class. The main assumption of LDA is that the data matrix is Gaussian with equal covariance matrix. We begin with exploring the univariate behavior of the data columns. We first write a function to report the summary of a column. If data is Gaussian then each column is also Gaussian. Note that otherway round is not true. We plot histogram and qq-plot for visual display and apply Shapiro Wilk test for confirmatory analysis. We subsequnetly write a function using Shaprio.test for a clearer understanding of stundents.

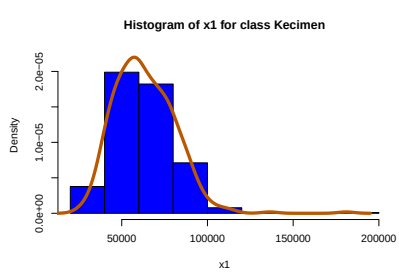
```
Shapiro_Test_Gaussianity <- function(data,column_name){
    # Perform Shapiro-Wilk test
    result <- shapiro.test(data)

    # Set the significance level
    significance_level <- 0.05

    # Null hypothesis (H0): The specific data column is normally distributed.
    # Alternative hypothesis (H1): The specific data column is not.

    # p-value criterion for the decision
    if (result$p.value < 0.05) {
      cat(sprintf("p-value: %.2e < 0.05 (significant level)
We reject the null hypothesis that the specific column is Gaussian. \n",
      result$p.value,column_name))
```

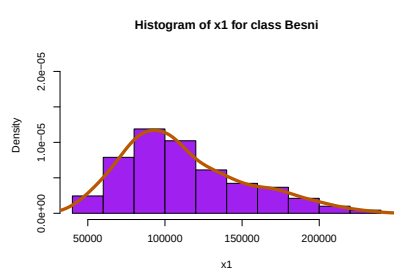
```
    } else {
      cat(sprintf("p-value: %.2e > 0.05 (significant level)
We fail to reject the null hypothesis, therefore the specific data column is Gaussian.\n",
      result$p.value,column_name))

    }
}
```

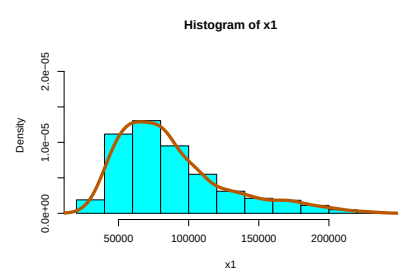
1) Area (x1): We first display the summary statistic of x1. We plot the histogram of x1 for both classes separately along with density estimates. Then we show the qq-plot to visualize the gap from normality. We finally apply the Shapiro Wilk test for confirmation

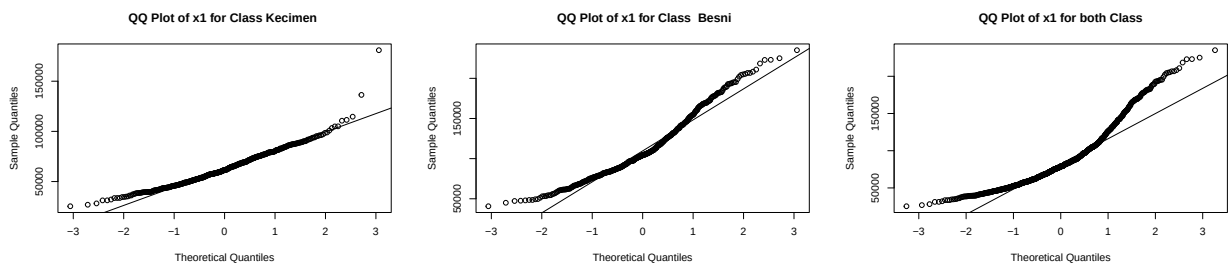
```
x1_Kecimen<-class_Kecimen$x1
hist(x1_Kecimen, main= "Histogram of x1 for class Kecimen", col = "blue",prob = TRUE,
     ylim=c(0,2.2e-05),xlab='x1',breaks=10)
lines(density(x1_Kecimen),
      lwd = 5,
      col = "#BA5801")
x1_Besni<-class_Besni$x1
hist(x1_Besni, main= "Histogram of x1 for class Besni", col = "purple",prob = TRUE,
     ylim=c(0,2.2e-05),xlab='x1',breaks=10)
lines(density(x1_Besni),
      lwd = 5,
      col = "#BA5801")
hist(raisin$x1, main= " Histogram of x1 ", col = "cyan",prob = TRUE,
     ylim=c(0,2.2e-05),xlab='x1',breaks=10)
lines(density(raisin$x1),
      lwd = 5,
      col = "#BA5801")

qqnorm(x1_Kecimen,main="QQ Plot of x1 for Class Kecimen ")
qqline(x1_Kecimen)
qqnorm(x1_Besni,main=" QQ Plot of x1 for Class  Besni")
qqline(x1_Besni)
qqnorm(raisin$x1,main=" QQ Plot of x1 for both Class ")
qqline(raisin$x1)

Shapiro_Test_Gaussianity(raisin$x7,'Area')
```

```
## p-value: 5.45e-18 < 0.05 (significant level)
## We reject the null hypothesis that the specific column is Gaussian.
```







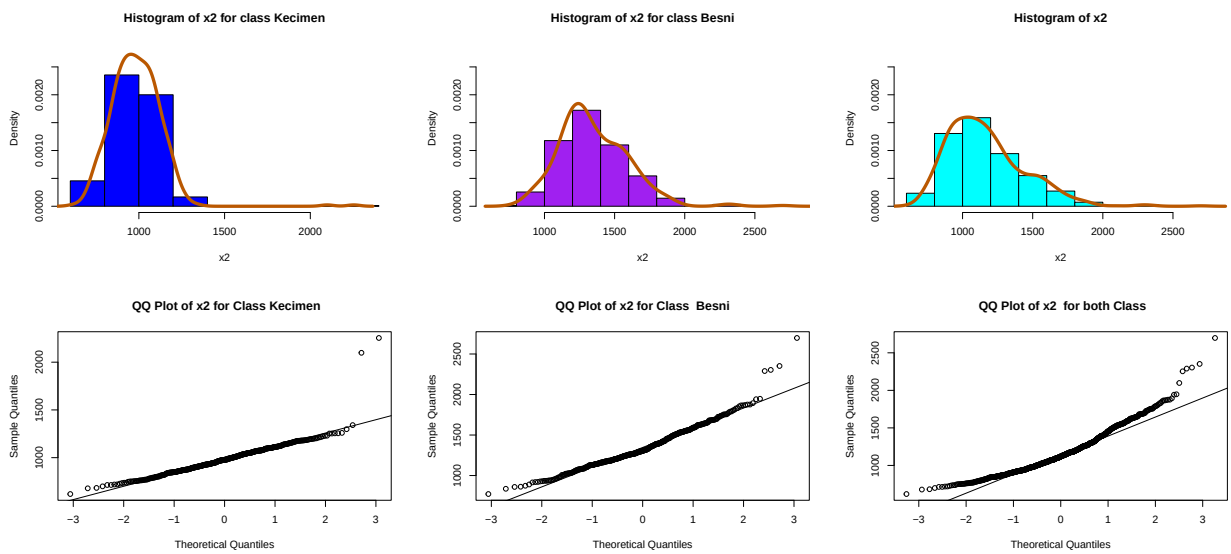


It is visually also observed from the histogram as well as qq-plot that column x1 is not Gaussian. Further the gap is quiet large from normal distribution. The column x1 is slightly positively skewed. Further the spread of both the classes for x1 is very different.

2) Perimeter(X2): We repeat the same exercise for x2 as we applied for x1

```
x2_Kecimen<-class_Kecimen$x2
hist(x2_Kecimen, main= "Histogram of x2 for class Kecimen", col = "blue",prob = TRUE,
     ylim=c(0,0.0028),xlab='x2',breaks=10)
lines(density(x2_Kecimen),
      lwd = 5,
      col = "#BA5801")
x2_Besni<-class_Besni$x2
hist(x2_Besni, main= "Histogram of x2 for class Besni", col = "purple",prob = TRUE,
     ylim=c(0,0.0028),xlab='x2',breaks=10)
lines(density(x2_Besni),
      lwd = 5,
      col = "#BA5801")

hist(raisin$x2, main= "Histogram of x2", col = "cyan",prob = TRUE,
     ylim=c(0,0.0028),xlab='x2',breaks=10)
lines(density(raisin$x2),
      lwd = 5,
      col = "#BA5801")

qqnorm(x2_Kecimen,main=" QQ Plot of x2 for Class Kecimen")
qqline(x2_Kecimen)
qqnorm(x2_Besni,main=" QQ Plot of x2 for Class  Besni ")
qqline(x2_Besni)
qqnorm(raisin$x2,main=" QQ Plot of x2  for both Class")
qqline(raisin$x2)

Shapiro_Test_Gaussianity(raisin$x2,'Perimeter')
```

```
## p-value: 7.34e-18 < 0.05 (significant level)
## We reject the null hypothesis that the specific column is Gaussian.
```

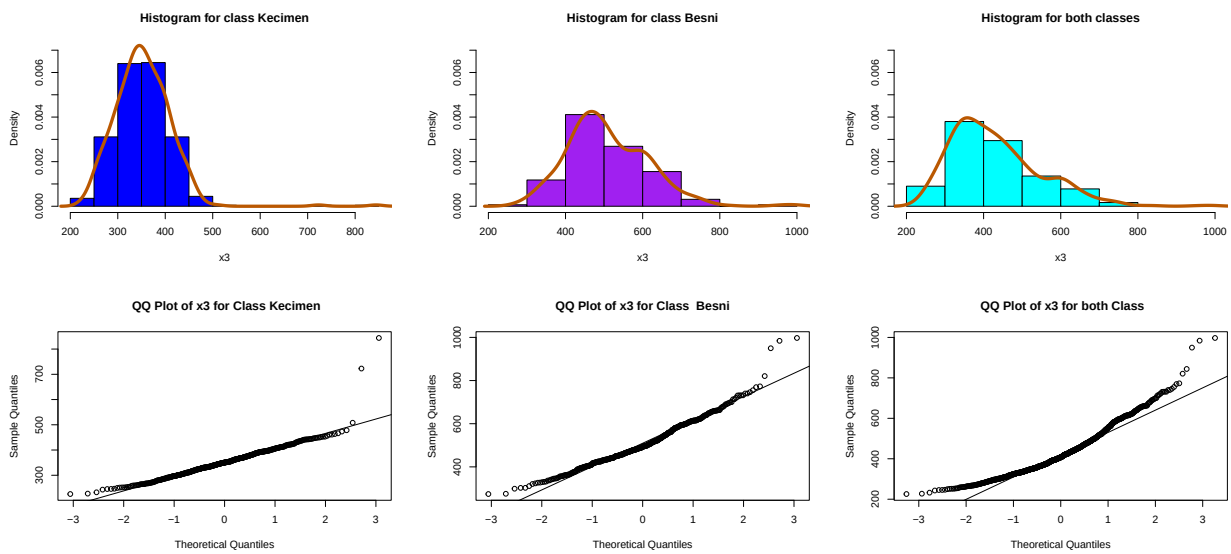


It is visually also observed from the histogram as well as qq-plot that column x1 is not Gaussian. Further the gap is quiet large from normal distribution. The column x2 is slightly positively skewed. Further the spread of both the classes for x2 is very different.

3) Major Axis Length(x3): We repeat the same exercise for x3 as we applied for x1

```
x3_Kecimen<-class_Kecimen$x3
hist(x3_Kecimen, main= "Histogram for class Kecimen", col = "blue",prob = TRUE,
     ylim=c(0,0.007),xlab='x3')
lines(density(x3_Kecimen),
      lwd = 5,
      col = "#BA5801")
x3_Besni<-class_Besni$x3
hist(x3_Besni, main= "Histogram for class Besni", col = "purple",prob = TRUE,
     ylim=c(0,0.007),xlab='x3')
lines(density(x3_Besni),
      lwd = 5,
      col = "#BA5801")
hist(raisin$x3, main= "Histogram for both classes", col = "cyan",prob = TRUE,
     ylim=c(0,0.007),xlab='x3')
lines(density(raisin$x3),
      lwd = 5,
      col = "#BA5801")

qqnorm(x3_Kecimen,main=" QQ Plot of x3 for Class Kecimen")
qqline(x3_Kecimen)
qqnorm(x3_Besni,main=" QQ Plot of x3 for Class  Besni ")
qqline(x3_Besni)
qqnorm(raisin$x3,main=" QQ Plot of x3 for both Class")
qqline(raisin$x3)

Shapiro_Test_Gaussianity(raisin$x3,'Major Axis Length')
```

```
## p-value: 2.74e-18 < 0.05 (significant level)
## We reject the null hypothesis that the specific column is Gaussian.
```

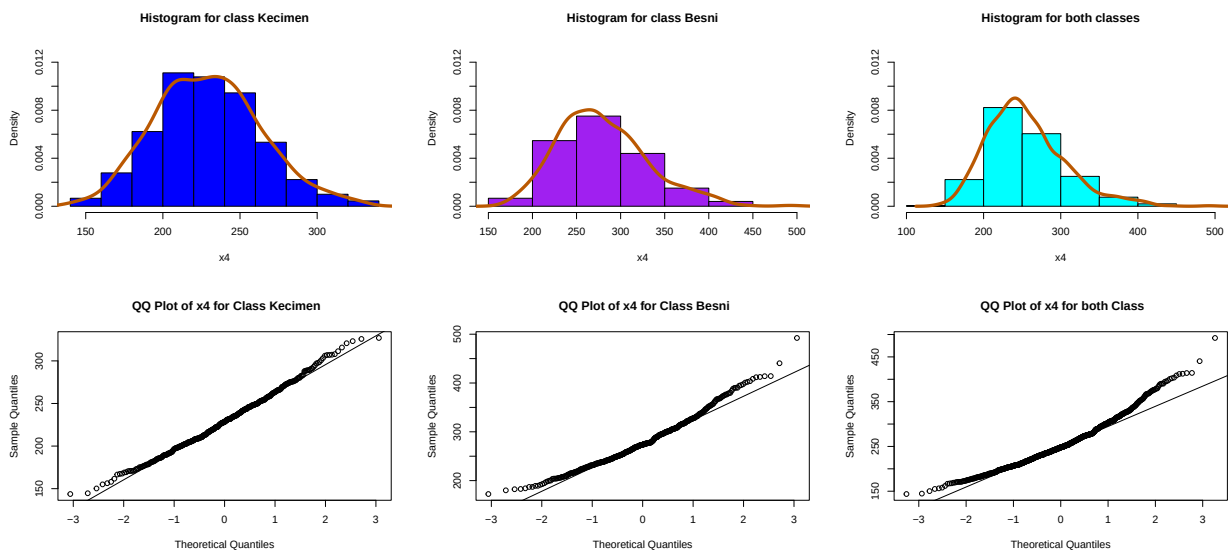


Same interpretation as x1

4) Minor Axis Length(x4): We repeat the same exercise for x4 as we applied for x1

```
x4_Kecimen<-class_Kecimen$x4
hist(x4_Kecimen, main= "Histogram for class Kecimen", col = "blue",prob = TRUE,
     ylim=c(0,0.013),xlab='x4')
lines(density(x4_Kecimen),
      lwd = 5,
      col = "#BA5801")
x4_Besni<-class_Besni$x4
hist(x4_Besni, main= "Histogram for class Besni", col = "purple",prob = TRUE,
     ylim=c(0,0.013),xlab='x4')
lines(density(x4_Besni),
      lwd = 5,
      col = "#BA5801")

hist(raisin$x4, main= "Histogram for both classes", col = "cyan",prob = TRUE,
     ylim=c(0,0.013),xlab='x4')
lines(density(raisin$x4),
      lwd = 5,
      col = "#BA5801")

qqnorm(x4_Kecimen,main=" QQ Plot of x4 for Class Kecimen")
qqline(x4_Kecimen)
qqnorm(x4_Besni,main=" QQ Plot of x4 for Class Besni")
qqline(x4_Besni)
qqnorm(raisin$x4,main=" QQ Plot of x4 for both Class")
qqline(raisin$x4)

Shapiro_Test_Gaussianity(raisin$x4,'Minor Axis Length')
```

```
## p-value: 8.30e-14 < 0.05 (significant level)
## We reject the null hypothesis that the specific column is Gaussian.
```

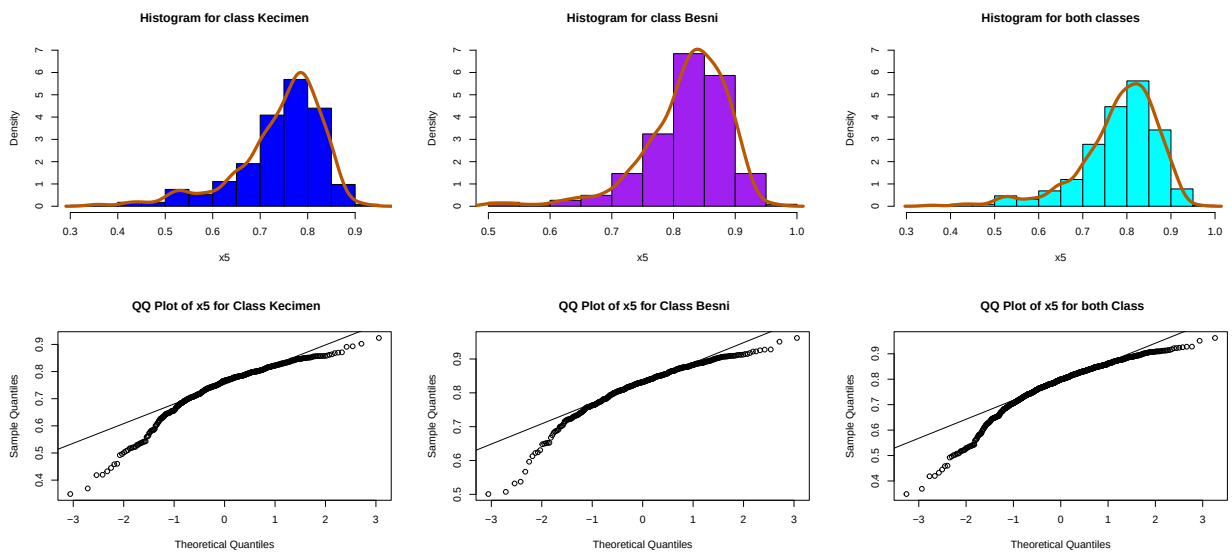


Same interpretation as x1

5) Eccentricity(x5): We repeat the same exercise for x5 as we applied for x1

```
x5_Kecimen<-class_Kecimen$x5
hist(x5_Kecimen, main= "Histogram for class Kecimen", col = "blue",prob = TRUE,
     ylim=c(0,7),xlab='x5')
lines(density(x5_Kecimen),
      lwd = 5,
      col = "#BA5801")

x5_Besni<-class_Besni$x5
hist(x5_Besni, main= "Histogram for class Besni", col = "purple",prob = TRUE,
     ylim=c(0,7),xlab='x5')
lines(density(x5_Besni),
      lwd = 5,
      col = "#BA5801")

hist(raisin$x5, main= "Histogram for both classes", col = "cyan",prob = TRUE,
     ylim=c(0,7),xlab='x5')
lines(density(raisin$x5),
      lwd = 5,
      col = "#BA5801")

qqnorm(x5_Kecimen,main=" QQ Plot of x5 for Class Kecimen")
qqline(x5_Kecimen)
qqnorm(x5_Besni,main=" QQ Plot of x5 for Class Besni")
qqline(x5_Besni)
qqnorm(raisin$x5,main=" QQ Plot of x5 for both Class")
qqline(raisin$x5)

Shapiro_Test_Gaussianity(raisin$x5,'Eccentricity')
```

```
## p-value: 3.08e-22 < 0.05 (significant level)
## We reject the null hypothesis that the specific column is Gaussian.
```

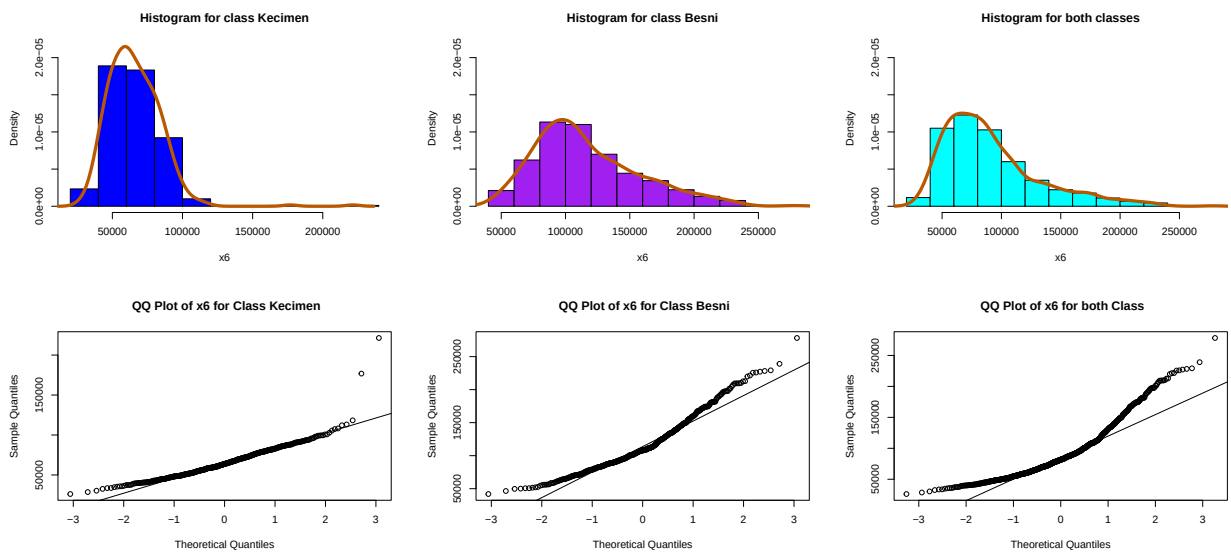


It is visually also observed from the histogram as well as qq-plot that column x5 is not Gaussian. Further the gap is quiet large from normal distribution. The column x5 is slightly negative skewed. Further the spread of both the classes for x5 is slightly same.

6) Convex Area (x6): We repeat the same exercise for x6 as we applied for x1

```
x6_Kecimen<-class_Kecimen$x6
hist(x6_Kecimen, main= "Histogram for class Kecimen", col = "blue",prob = TRUE,
     ylim=c(0,2.1e-05),xlab='x6')
lines(density(x6_Kecimen),
      lwd = 5,
      col = "#BA5801")
x6_Besni<-class_Besni$x6
hist(x6_Besni, main= "Histogram for class Besni", col = "purple",prob = TRUE,
     ylim=c(0,2.1e-05),xlab='x6')
lines(density(x6_Besni),
      lwd = 5,
      col = "#BA5801")

hist(raisin$x6, main= "Histogram for both classes", col = "cyan",prob = TRUE,
     ylim=c(0,2.1e-05),xlab='x6')
lines(density(raisin$x6),
      lwd = 5,
      col = "#BA5801")

qqnorm(x6_Kecimen,main=" QQ Plot of x6 for Class Kecimen")
qqline(x6_Kecimen)
qqnorm(x6_Besni,main=" QQ Plot of x6 for Class Besni")
qqline(x6_Besni)
qqnorm(raisin$x6,main=" QQ Plot of x6 for both Class")
qqline(raisin$x6)

Shapiro_Test_Gaussianity(raisin$x6,'Convex Area')
```

```
## p-value: 1.35e-23 < 0.05 (significant level)
## We reject the null hypothesis that the specific column is Gaussian.
```

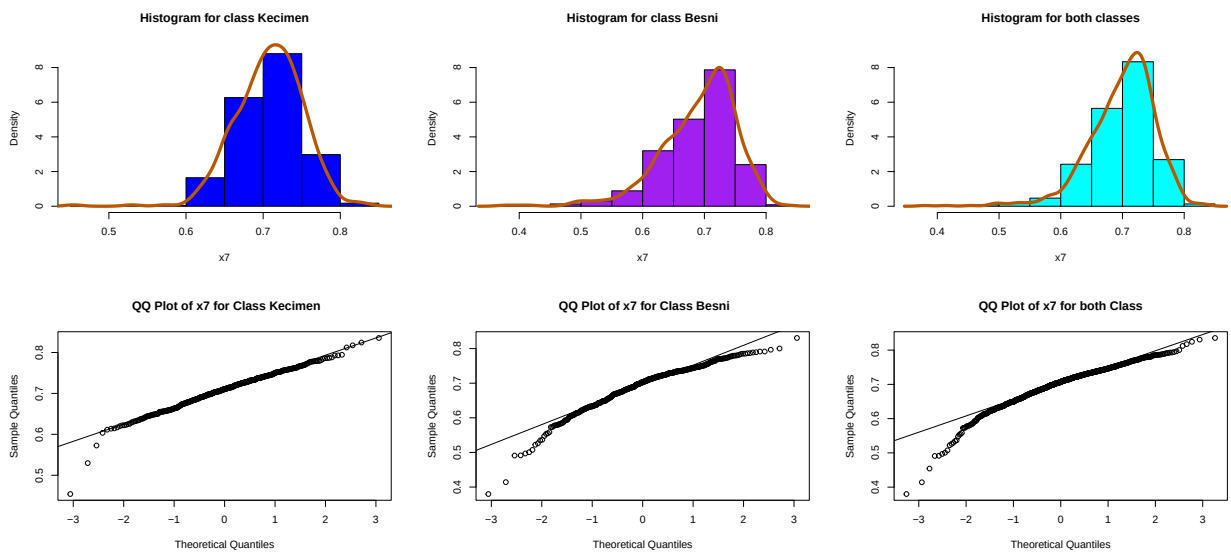


Same interpretation as x1

7) extent (x7): We repeat the same exercise for x7 as we applied for x1

```
x7_Kecimen<-class_Kecimen$x7
hist(x7_Kecimen, main= "Histogram for class Kecimen", col = "blue",prob = TRUE,
     ylim=c(0,9),xlab='x7')
lines(density(x7_Kecimen),
      lwd = 5,
      col = "#BA5801")
x7_Besni<-class_Besni$x7
hist(x7_Besni, main= "Histogram for class Besni", col = "purple",prob = TRUE,
     ylim=c(0,9),xlab='x7')
lines(density(x7_Besni),
      lwd = 5,
      col = "#BA5801")

hist(raisin$x7, main= "Histogram for both classes", col = "cyan",prob = TRUE,
     ylim=c(0,9),xlab='x7')
lines(density(raisin$x7),
      lwd = 5,
      col = "#BA5801")

qqnorm(x7_Kecimen,main=" QQ Plot of x7 for Class Kecimen")
qqline(x7_Kecimen)
qqnorm(x7_Besni,main=" QQ Plot of x7 for Class Besni")
qqline(x7_Besni)
qqnorm(raisin$x7,main=" QQ Plot of x7 for both Class ")
qqline(raisin$x7)

Shapiro_Test_Gaussianity(raisin$x7,'extent')
```

```
## p-value: 5.45e-18 < 0.05 (significant level)
## We reject the null hypothesis that the specific column is Gaussian.
```



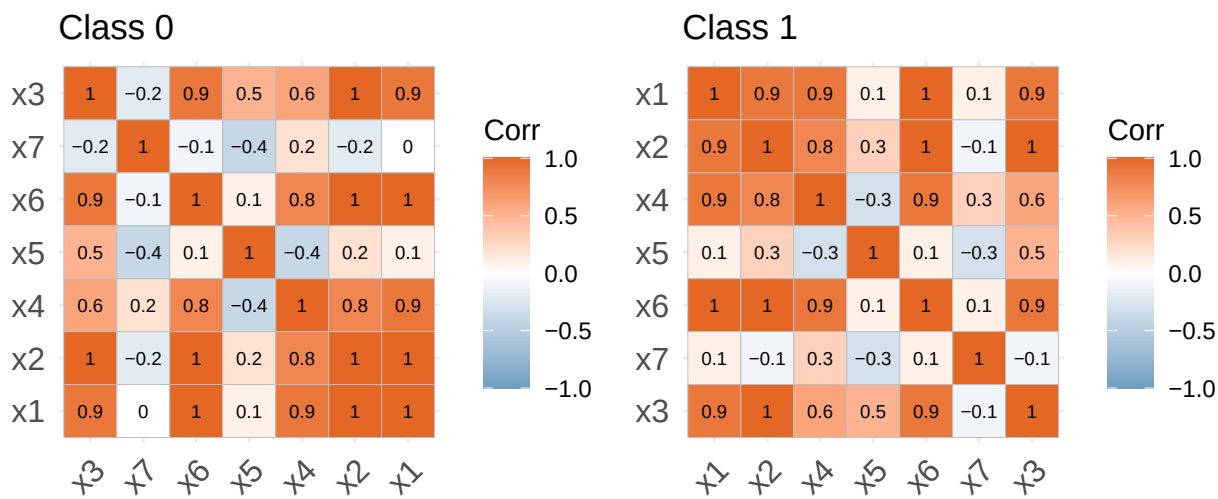
It is visually also observed from the histogram as well as qq-plot that column x7 is not Gaussian. Further the gap is quiet large from normal distribution. The column x7 is slightly Negative skewed. Further the spread of both the classes for x7 is slightly different. ## B-) Multivariate Analysis.

1-) Bivariate Analysis.

1-) Plot Correlation Matrix

```
options(digits = 2)
corr_Kecimen <- cor(class_Kecimen[-8])
corr_Besni <- cor(class_Besni[-8])
corr_all<- cor(raisin[-8])


 # Plot the correlation matrix for corr_class_Kecimen with title
my1<-ggcorrplot(t(corr_Kecimen[,7:1]),
           hc.order = FALSE,
           colors = c("#6D9EC1", "white", "#E46726"),
           lab = TRUE,
           lab_size=2.5,
           digits = 1)+ggtitle("Class 0")


# Plot the correlation matrix for corr_class_Besni with title


my2<-ggcorrplot(corr_Besni[,7:1],
           hc.order = FALSE,
           colors = c("#6D9EC1", "white", "#E46726"),
           lab = TRUE,
           lab_size=2.5,
           digits = 1)+ggtitle("Class 1")
```

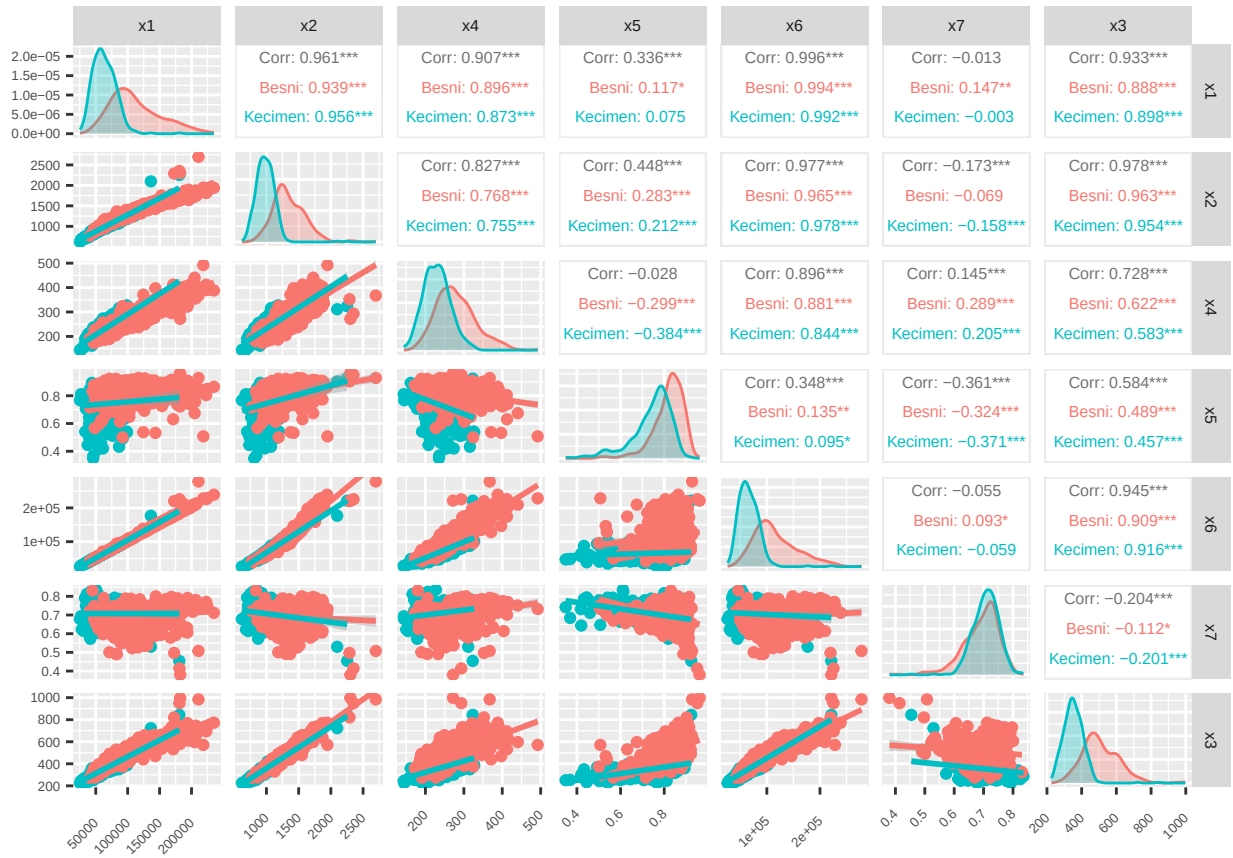
```r
plot_grid(my1, my2,
  label_size = 20,
  axis = "tb",
  align = "hv")
```



1-) Correlation,Scatter plot and Distribution Plot

```r
# Define custom function for the KDE plots on the diagonal
custom_kde <- function(data, mapping, ...){
  ggplot(data = data, mapping = mapping) +
    geom_density(aes(fill = class), alpha = 0.3, ...)
}

# Use ggpairs with custom KDE function
p <- ggpairs(raisin,
             columns = 1:7,
             aes(colour = class),
             lower = list(continuous = "smooth"),
             upper = list(continuous = wrap("cor", method = "pearson",size = 2)),
             diag = list(continuous = custom_kde))  # Specify the custom function here

# Apply theme
p <- p + theme(
  text = element_text(size = 7),  # Overall text size
  # X-axis text size
  axis.text.x = element_text(size = 5, angle = 45, vjust = 0.5, hjust = 1),
```

```
  axis.text.y = element_text(size = 5),  # Y-axis text size
  axis.title.x = element_text(size = 5),  # X-axis title size
  axis.title.y = element_text(size = 5)  # Y-axis title size
)

print(p)
```



2-) Display the Determinant

```
cat(sprintf("the determinant of class 0 is: %.e\nthe determinant of class 1 is: %.e",
            det(corr_Kecimen),det(corr_Besni)))
```

```
## the determinant of class 0 is: 4e-08
## the determinant of class 1 is: 5e-08
```

3-) Display the eigen Value

```
eigen_Kecimen<-eigen(corr_Kecimen)$values
eigen_Besni<-eigen(corr_Besni)$values


variance_proportion_Kecimen <- round(eigen_Kecimen / sum(eigen_Kecimen) *100,2)
variance_proportion_Besni <- round(eigen_Besni / sum(eigen_Besni) * 100,2)
cat(sprintf("the eigen proportion of class 0 is: %s\nthe eigen proportion of
                class 1 is: %s",
            list( variance_proportion_Kecimen),list( variance_proportion_Besni)))
```

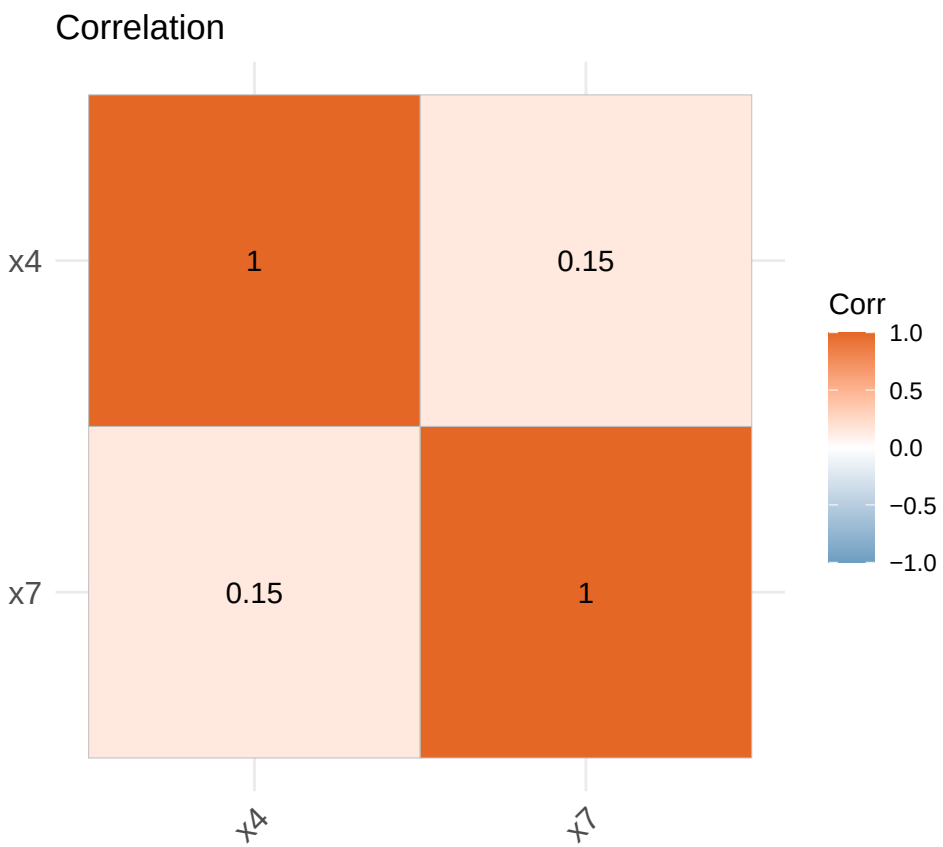
```
## the eigen proportion of class 0 is: c(64.78, 24.24, 9.8, 0.83, 0.22, 0.11, 0.02)
## the eigen proportion of
##                  class 1 is: c(65.39, 23.55, 9.9, 0.77, 0.26, 0.1, 0.02)
```

5-) Feature Selections

```r
det_corr <- function(corr, cutoff){
  #  Find pairs with high correlation
  high_corr <- findCorrelation(corr, cutoff = cutoff)
  # Get the names of the highly correlated variables
  high_corr_vars <- colnames(raisin)[high_corr]
  # Remove the highly correlated variables from the data
  df_new <- raisin[ , !(names(raisin) %in% high_corr_vars)]
  corr_all_new = cor(df_new[-length(names(df_new))])
  print(sprintf("the determinant is: %.2f",det(corr_all_new)))
  #corrplot(corr_all_new, method = "shade")
  print(ggcorrplot(corr_all_new[,ncol(corr_all_new):1],
           hc.order = FALSE,
           colors = c("#6D9EC1", "white", "#E46726"),
           lab = TRUE,
           lab_size=4,
           digits = 2)+ggtitle("Correlation "))
  return (df_new)
}
```

```r
df_new <- det_corr(corr_all,0.30)
```

```
## [1] "the determinant is: 0.98"
```



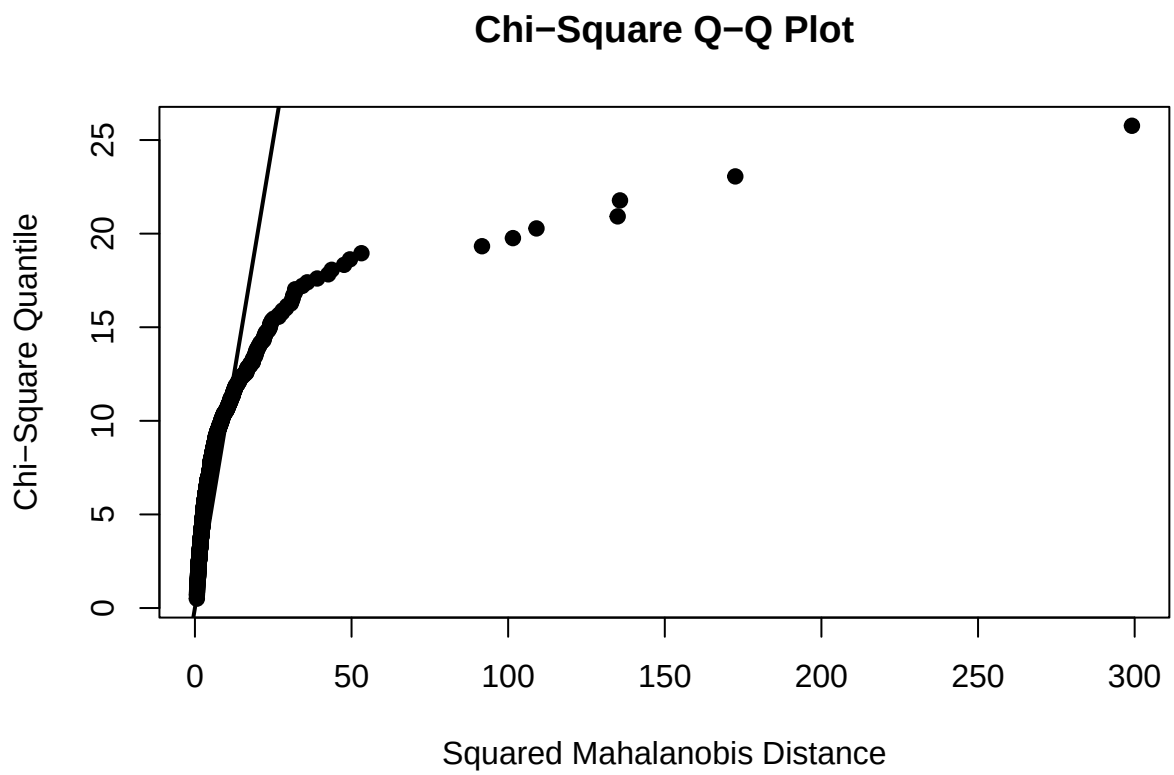
6-) multivariate normality check

mvn function performs multivariate normality tests, including Mardia, Royston, Henze-Zirkler, Dornik-Haansen, and E-Statistics. It also implements multivariate outlier detection and univariate normality of marginal distributions through plots and tests, and performs multivariate Box-Cox transformation.

For moore information about this function click on this link : https://cran.r-project.org/web/packages/MVN/MVN.pdf

```
print(mvn(
  raisin[-8],
  subset = NULL,
  mvnTest = "hz",
  covariance = TRUE,
  tol = 1e-25,
  alpha = 0.5,
  scale = FALSE,
  desc = TRUE,
  transform = "none",
  R = 1000,
  univariateTest = "AD",
  univariatePlot = "none",
  multivariatePlot = "qq",
  multivariateOutlierMethod = "none",
  bc = FALSE,
  bcType = "rounded",
  showOutliers = TRUE,
```

```
  showNewData = FALSE
))
```




```
## $multivariateNormality
##            Test HZ p value MVN
## 1 Henze-Zirkler 18       0  NO
##
## $univariateNormality
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling    x1          25.0  <0.001      NO
## 2 Anderson-Darling    x2          11.2  <0.001      NO
## 3 Anderson-Darling    x4           7.1  <0.001      NO
## 4 Anderson-Darling    x5          18.0  <0.001      NO
## 5 Anderson-Darling    x6          25.5  <0.001      NO
## 6 Anderson-Darling    x7           8.3  <0.001      NO
## 7 Anderson-Darling    x3          12.9  <0.001      NO
##
## $Descriptives
##      n    Mean Std.Dev  Median     Min     Max    25th    75th  Skew Kurtosis
## x1 900 8.8e+04 3.9e+04 7.9e+04 2.5e+04 2.4e+05 5.9e+04 1.1e+05  1.17     1.05
## x2 900 1.2e+03 2.7e+02 1.1e+03 6.2e+02 2.7e+03 9.7e+02 1.3e+03  1.01     1.72
## x4 900 2.5e+02 5.0e+01 2.5e+02 1.4e+02 4.9e+02 2.2e+02 2.8e+02  0.80     0.93
## x5 900 7.8e-01 9.0e-02 8.0e-01 3.5e-01 9.6e-01 7.4e-01 8.4e-01 -1.32     2.46
## x6 900 9.1e+04 4.1e+04 8.2e+04 2.6e+04 2.8e+05 6.2e+04 1.1e+05  1.24     1.40
## x7 900 7.0e-01 5.3e-02 7.1e-01 3.8e-01 8.4e-01 6.7e-01 7.3e-01 -1.15     3.30
```

```
## x3 900 4.3e+02 1.2e+02 4.1e+02 2.3e+02 1.0e+03 3.5e+02 4.9e+02  0.99     1.30
##
## $multivariateOutliers
## NULL
```