INTRODUCTION:

This project delves into a dataset featuring 600 Portuguese wines, examining various chemical and sensory attributes. From acidity to alcohol content and quality assessments, the dataset provides a comprehensive overview. The primary focus is on understanding patterns and relationships within the data, with tasks including summarizing variables, identifying correlations, and visualizing trends. Key objectives involve differentiating red and white wines based on continuous measurements as well as predicting quality distinctions. Through this analysis, we aim to unveil insights that resonate with wine enthusiasts and industry professionals, briefly exploring the diverse world encapsulated in each bottle of Portuguese wine.

Data Summary:

The data used in this project is “wine.csv”. The original dataset consists of 600 observations and 12 variables which are shown below:

Fixed Acidity
Volatile Acidity
Citric Acid
Residual Sugar
Chlorides
Free Sulphur dioxide
Total sulphur dioxide
Density
pH
Sulphates
Alcohol
Quality
red(indicator:1 if red, 0 if white)

Based on the original data the continuous variables are Fixed Acidity, Volatile Acidity, Citric Acid, Residual Sugar, Chlorides, Free Sulphur dioxide, Total sulphur dioxide, Density, pH, Sulphates, and Alcohol. Grouping variables are red, indicating red if 1 and white if 0 and quality.

Materials and Methods:

First, we looked into the summary of the continuous variables, and made plots to learn more about each variable and their relationship, we used a correlation matrix to know which variables are most and least correlated. We also use PCA to find the combination of variables that explain the most variance. We compared the mean difference of the continuous variables between red and white wine, as well as the quality, obtaining the simultaneous confidence intervals.
Then, we assess the multi-variate normality and check for MVN plausiblity.

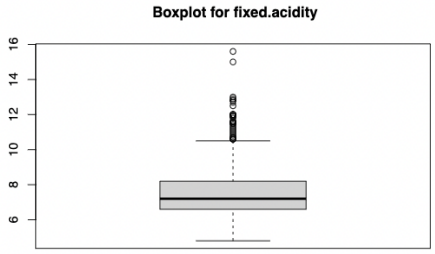
Finally, we perform Fisher's discriminate analysis for classification using the transformed data.

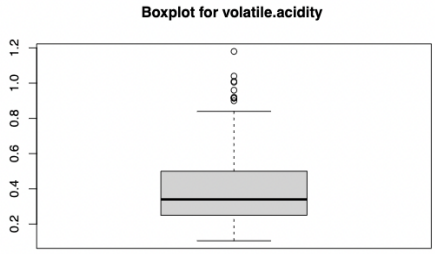
Analysis and Results:
Based on the original data we created a red_wine_subset and a white_wine_subset to understand those two variables better. The contains 300 samples each and the summary is as follows:

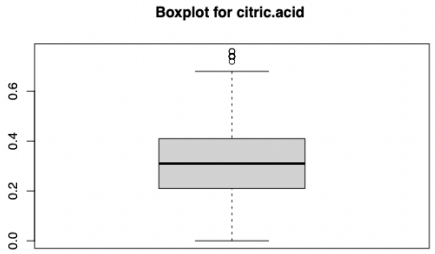
```
## Summary for Overall Data set t:

##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 4.800   Min.   :0.1050   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.600   1st Qu.:0.2500   1st Qu.:0.2100   1st Qu.: 1.900
##  Median : 7.200   Median :0.3400   Median :0.3100   Median : 2.400
##  Mean   : 7.585   Mean   :0.3865   Mean   :0.3086   Mean   : 4.436
##  3rd Qu.: 8.200   3rd Qu.:0.5000   3rd Qu.:0.4100   3rd Qu.: 5.525
##  Max.   :15.600   Max.   :1.1800   Max.   :0.7600   Max.   :18.950
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.01200   Min.   :  3.00      Min.   :  7.00       Min.   :0.9885
##  1st Qu.:0.04200   1st Qu.: 13.00      1st Qu.: 38.00       1st Qu.:0.9931
##  Median :0.05900   Median : 24.00      Median : 89.00       Median :0.9957
##  Mean   :0.06411   Mean   : 26.27      Mean   : 92.03       Mean   :0.9952
##  3rd Qu.:0.08000   3rd Qu.: 36.00      3rd Qu.:138.00       3rd Qu.:0.9973
##  Max.   :0.40300   Max.   :108.00      Max.   :240.00       Max.   :1.0032
##        pH          sulphates         alcohol
##  Min.   :2.870   Min.   :0.2900   Min.   : 8.50
##  1st Qu.:3.140   1st Qu.:0.4700   1st Qu.: 9.50
##  Median :3.250   Median :0.5500   Median :10.50
##  Mean   :3.249   Mean   :0.5838   Mean   :10.61
##  3rd Qu.:3.360   3rd Qu.:0.6700   3rd Qu.:11.50
##  Max.   :3.900   Max.   :1.9500   Max.   :14.00
```

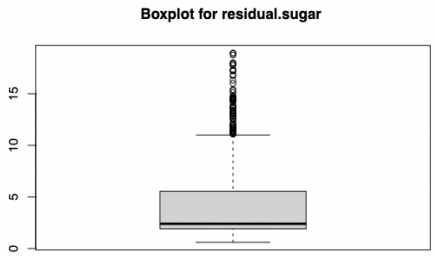
```
## Summary for Red Wine Subset:

##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 4.900   Min.   :0.1200   Min.   :0.0000   Min.   : 0.900
##  1st Qu.: 7.175   1st Qu.:0.3600   1st Qu.:0.1100   1st Qu.: 1.900
##  Median : 8.000   Median :0.4800   Median :0.2800   Median : 2.200
##  Mean   : 8.372   Mean   :0.4948   Mean   :0.2845   Mean   : 2.473
##  3rd Qu.: 9.400   3rd Qu.:0.6200   3rd Qu.:0.4425   3rd Qu.: 2.600
##  Max.   :15.600   Max.   :1.1800   Max.   :0.7600   Max.   :13.400
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.01200   Min.   : 3.00       Min.   :  7.00       Min.   :0.9901
##  1st Qu.:0.06875   1st Qu.: 8.00       1st Qu.: 23.00       1st Qu.:0.9954
##  Median :0.07900   Median :14.00       Median : 38.00       Median :0.9966
##  Mean   :0.08229   Mean   :16.38       Mean   : 44.42       Mean   :0.9966
##  3rd Qu.:0.08800   3rd Qu.:23.00       3rd Qu.: 56.00       3rd Qu.:0.9976
##  Max.   :0.40300   Max.   :72.00       Max.   :160.00       Max.   :1.0032
##        pH          sulphates         alcohol
##  Min.   :2.890   Min.   :0.3900   Min.   : 9.00
##  1st Qu.:3.217   1st Qu.:0.5600   1st Qu.: 9.70
##  Median :3.310   Median :0.6300   Median :10.50
##  Mean   :3.315   Mean   :0.6763   Mean   :10.65
##  3rd Qu.:3.402   3rd Qu.:0.7600   3rd Qu.:11.50
##  Max.   :3.900   Max.   :1.9500   Max.   :14.00

## Summary for White Wine Subset:

##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 4.800   Min.   :0.1050   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.800
##  Median : 6.750   Median :0.2600   Median :0.3200   Median : 4.950
##  Mean   : 6.798   Mean   :0.2783   Mean   :0.3327   Mean   : 6.398
##  3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.:10.625
##  Max.   :10.700   Max.   :0.8150   Max.   :0.7400   Max.   :18.950
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.02000   Min.   :  4.00      Min.   : 45.0        Min.   :0.9885
##  1st Qu.:0.03500   1st Qu.: 25.75      1st Qu.:111.0        1st Qu.:0.9915
##  Median :0.04200   Median : 35.00      Median :137.0        Median :0.9934
##  Mean   :0.04593   Mean   : 36.16      Mean   :139.6        Mean   :0.9939
##  3rd Qu.:0.04900   3rd Qu.: 46.00      3rd Qu.:166.0        3rd Qu.:0.9963
##  Max.   :0.20900   Max.   :108.00      Max.   :240.0        Max.   :1.0004
##        pH          sulphates         alcohol
##  Min.   :2.870   Min.   :0.2900   Min.   : 8.50
##  1st Qu.:3.080   1st Qu.:0.4100   1st Qu.: 9.40
##  Median :3.170   Median :0.4700   Median :10.50
##  Mean   :3.183   Mean   :0.4913   Mean   :10.58
##  3rd Qu.:3.290   3rd Qu.:0.5400   3rd Qu.:11.50
##  Max.   :3.770   Max.   :1.0600   Max.   :13.70
```

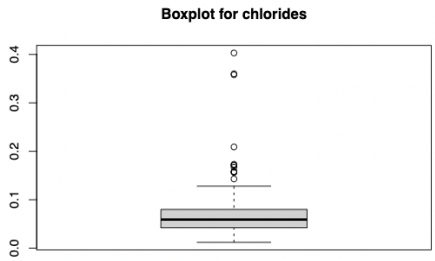
From the summary for red wine subset, white wine subset, and overall wine. We can see that the mean value for Fixed acidity, Volatile Acidity, chlorides, and sulfates is higher for red wine compared to white and overall wine. Similarly, the mean value for residual sugar, free sulphur dioxide, and total sulphur dioxide is more for white wine compared to red and overall wine. We will explore the boxplot and correlation matrix to learn more.

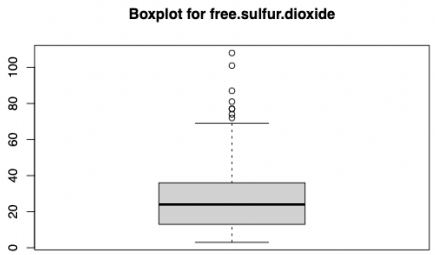


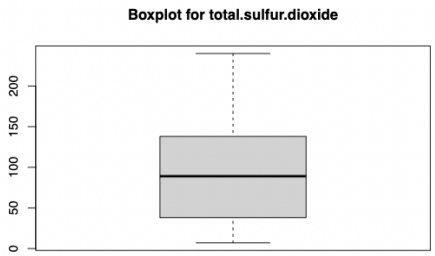


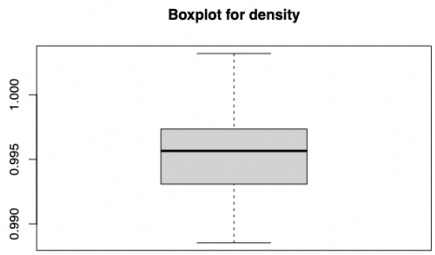


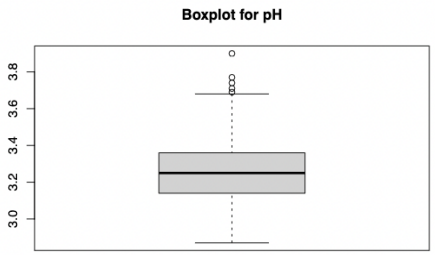


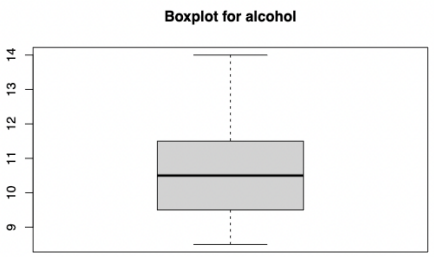




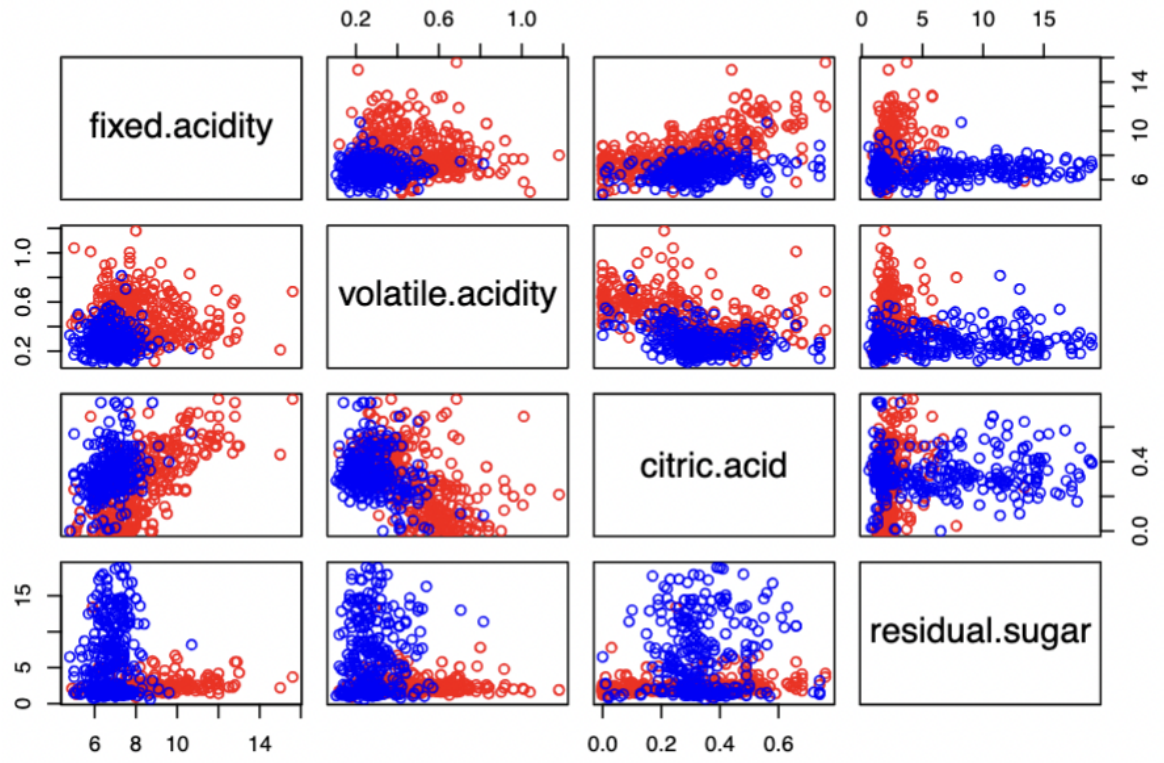


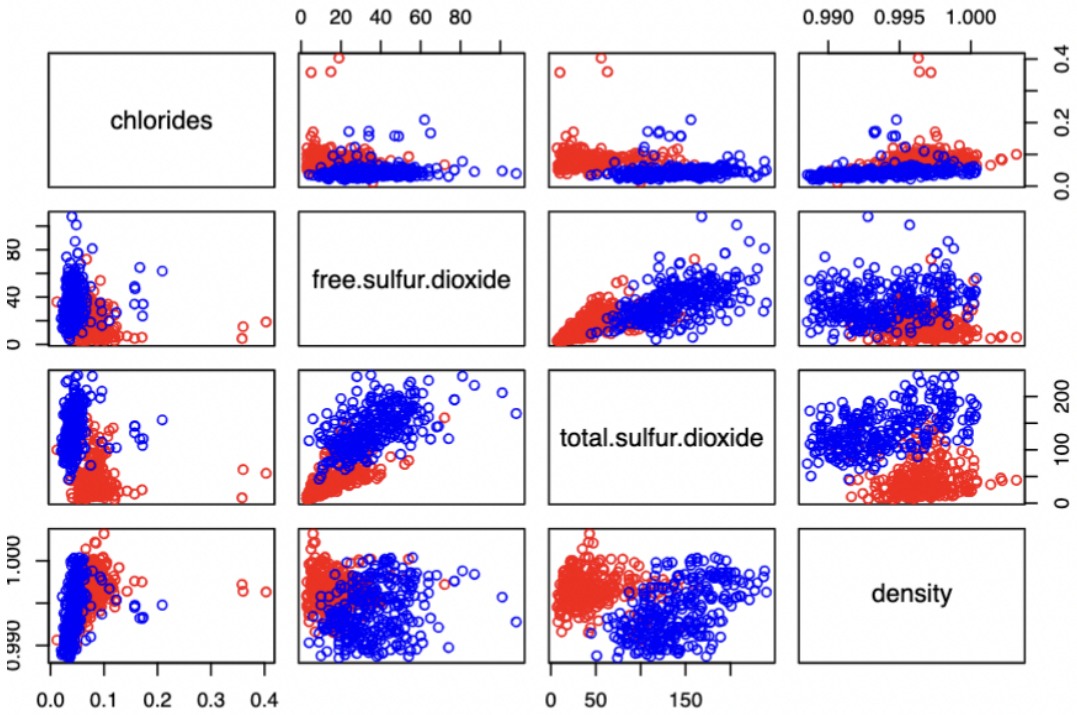


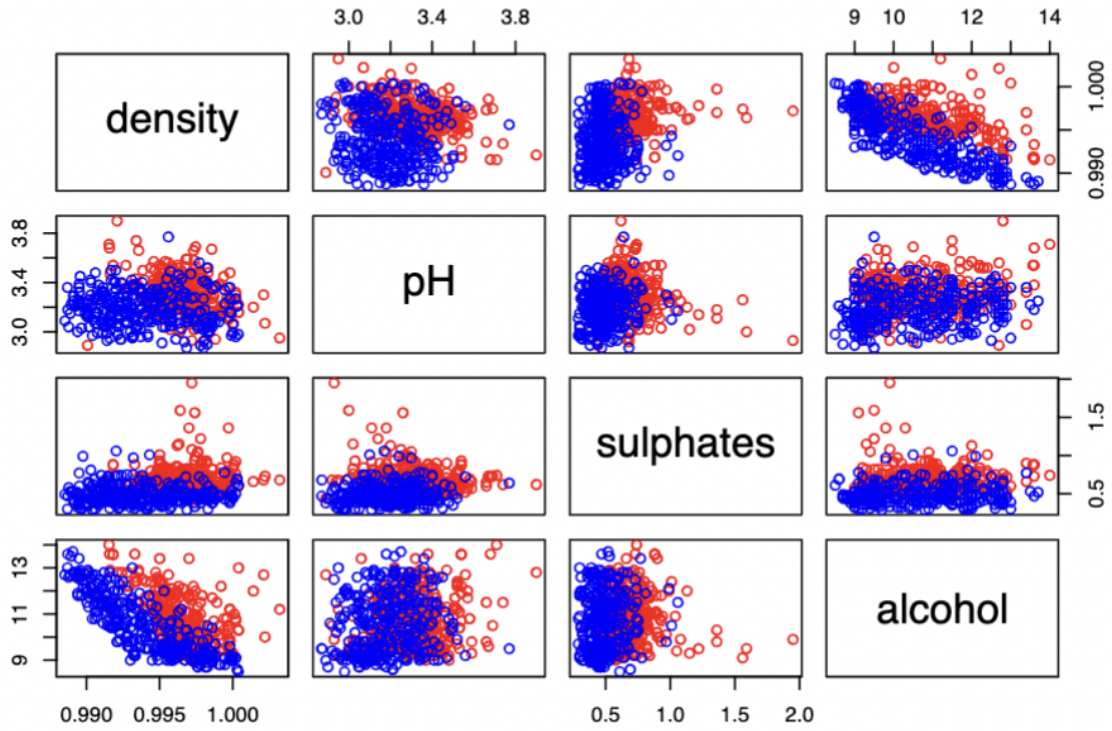




From the boxplot, we can see that there are outliers in fixed. acidity, volatile.acidity, citric.acid, residual.sugar, chlorides, free.sulfur.dioxide, pH, and sulfates.


fixed.acidity
volatile.acidity
citric.acid
residual.sugar


chlorides
free.sulfur.dioxide
total.sulfur.dioxide
density



From the plots, we can see that there is a lot of overlapping in data points between the variables. The variable with less no of overlapping is the total. sulfur. dioxide and density. We can see a positive relationship between the variable total. sulfur. dioxide and free.sulfur.dioxide.We can see a negative relationship between density and alcohol. We will further investigate the relationship using a correlation matrix.

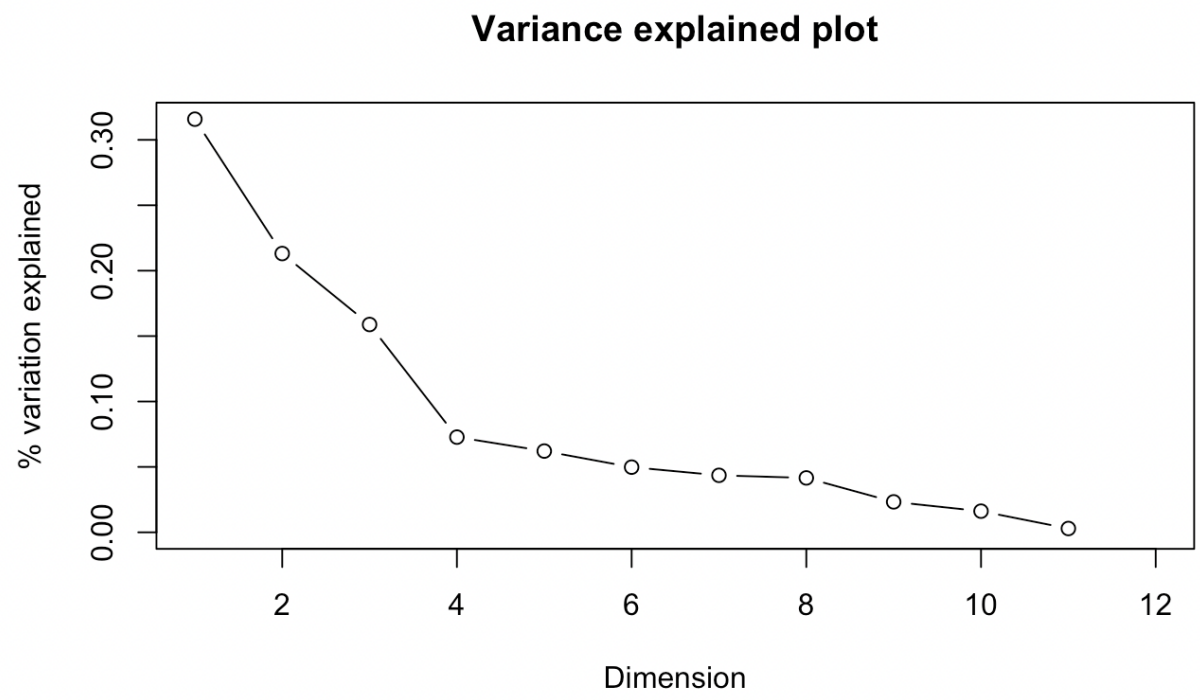
```
> correlation_matrix
                      fixed.acidity volatile.acidity citric.acid residual.sugar   chlorides free.sulfur.dioxide
fixed.acidity            1.00000000       0.15452574  0.42840905     -0.1728437  0.34762644         -0.39822236
volatile.acidity         0.15452574       1.00000000 -0.45403620     -0.2517687  0.37474328         -0.34364135
citric.acid              0.42840905      -0.45403620  1.00000000      0.1211949  0.07990265          0.05618164
residual.sugar          -0.17284375      -0.25176873  0.12119486      1.0000000 -0.20135147          0.45288180
chlorides                0.34762644       0.37474328  0.07990265     -0.2013515  1.00000000         -0.30365440
free.sulfur.dioxide     -0.39822236      -0.34364135  0.05618164      0.4528818 -0.30365440          1.00000000
total.sulfur.dioxide    -0.44037858      -0.41896624  0.10117308      0.5642441 -0.39966614          0.76649931
density                  0.57615026       0.31119807  0.11527440      0.3598633  0.42097585         -0.15757001
pH                      -0.23633014       0.35226788 -0.39165557     -0.3395116  0.06960826         -0.23922718
sulphates                0.38946835       0.16100799  0.13783211     -0.2788560  0.41369560         -0.31306455
alcohol                 -0.06086882      -0.08934422  0.06993635     -0.3342381 -0.23500330         -0.17698076
                      total.sulfur.dioxide     density          pH  sulphates     alcohol
fixed.acidity                  -0.4403786  0.57615026 -0.23633014  0.3894683 -0.06086882
volatile.acidity               -0.4189662  0.31119807  0.35226788  0.1610080 -0.08934422
citric.acid                     0.1011731  0.11527440 -0.39165557  0.1378321  0.06993635
residual.sugar                  0.5642441  0.35986329 -0.33951156 -0.2788560 -0.33423813
chlorides                      -0.3996661  0.42097585  0.06960826  0.4136956 -0.23500330
free.sulfur.dioxide             0.7664993 -0.15757001 -0.23922718 -0.3130646 -0.17698076
total.sulfur.dioxide            1.0000000 -0.18907149 -0.33622513 -0.4242543 -0.22622292
density                        -0.1890715  1.00000000  0.01731977  0.2712817 -0.61587691
pH                             -0.3362251  0.01731977  1.00000000  0.1493122  0.14946617
sulphates                      -0.4242543  0.27128171  0.14931225  1.0000000  0.08759460
alcohol                        -0.2262229 -0.61587691  0.14946617  0.0875946  1.00000000
>
```

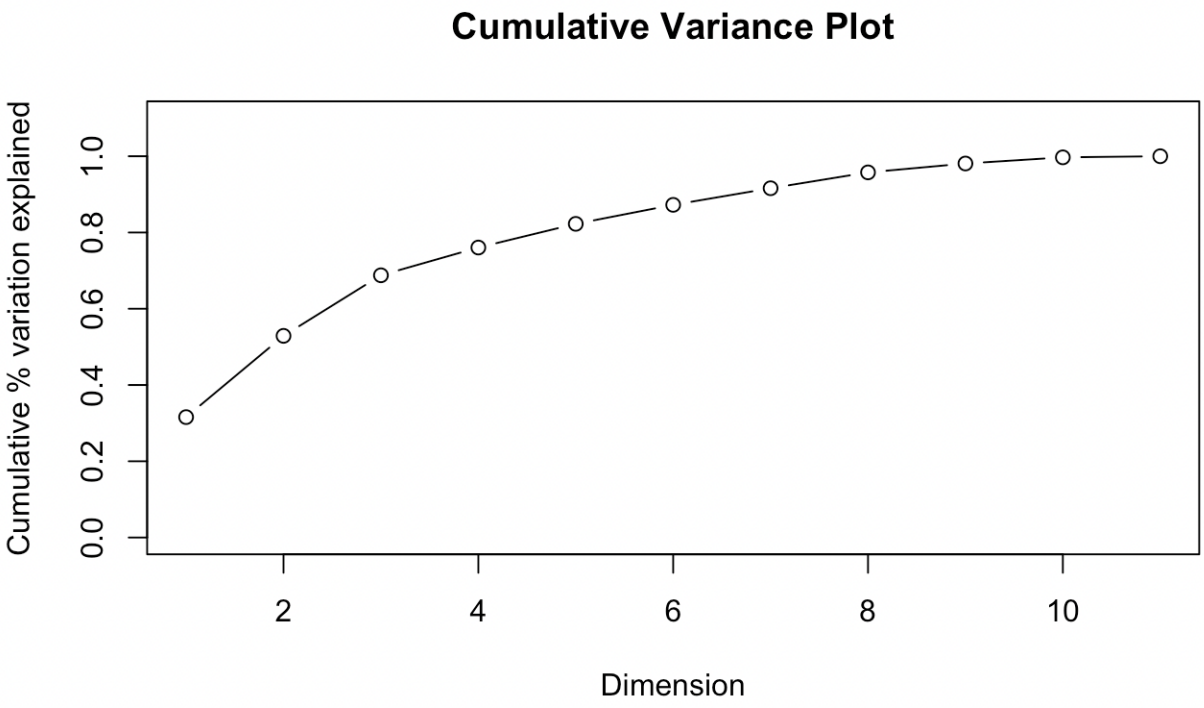
From the correlation matrix above The most positively correlated variables are total. sulfur. dioxide and free.sulfur.dioxide. Meanwhile, the most negatively correlated variables are alcohol and density. The negative correlation between alcohol and density explains that the higher alcohol has lower density. The positive correlation explains that higher total free sulfur dioxide means higher free sulfur dioxide and vice versa.

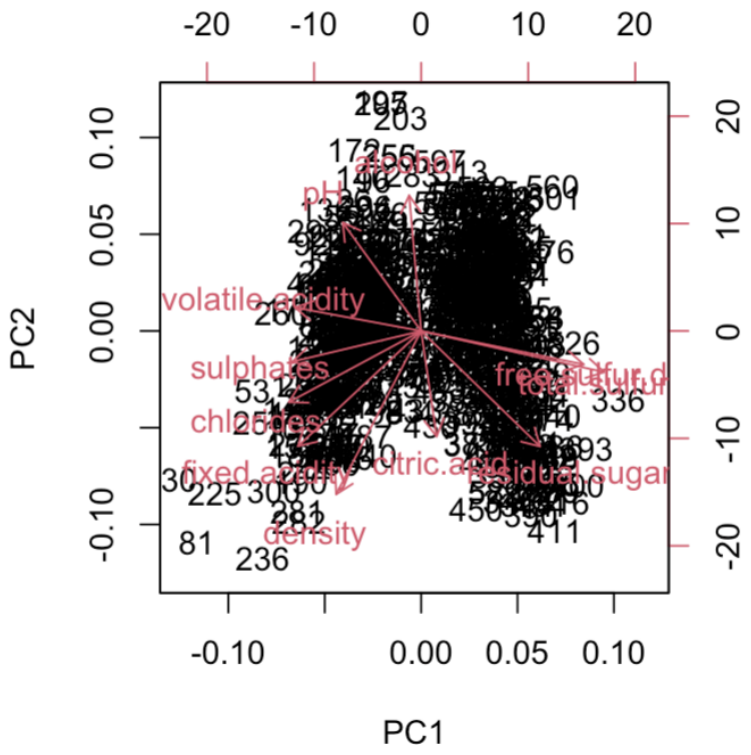
Principal component analysis:

The result of PCA is:

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.8638 1.5311 1.3219 0.89475 0.82658 0.74028 0.69232
## Proportion of Variance 0.3158 0.2131 0.1589 0.07278 0.06211 0.04982 0.04357
## Cumulative Proportion  0.3158 0.5289 0.6877 0.76053 0.82265 0.87246 0.91604
##                           PC8     PC9    PC10    PC11
## Standard deviation     0.67637 0.50605 0.42182 0.17915
## Proportion of Variance 0.04159 0.02328 0.01618 0.00292
## Cumulative Proportion  0.95763 0.98091 0.99708 1.00000
```


Variance explained plot
% variation explained
0.00
0.10
0.20
0.30
2
4
6
8
10
12
Dimension


Cumulative Variance Plot
Cumulative % variation explained
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
Dimension



In analyzing wine characteristics, Principal Component Analysis (PCA) was performed to identify the key components capturing the variance in our dataset. The goal was to determine the optimal number of principal components for retaining essential information. The cumulative proportion of variance analysis indicates that to preserve 95% of the variance, only the first 8 principal components are needed. If a 90% retention is acceptable, 7 principal components suffice.

The scree plot visually supports these findings, showing a noticeable decrease in the significance of variance beyond the 8th principal component. However, we must be careful not to cut off the number of PCAs too soon as it will cause problems in the future.

The biplot reveals relationships between variables and observations. We can see that pH and alcohol positively correlate, while citric. acid and alcohol show a negative correlation. This visualization helps us have a greater understanding of the specific relationships between the variables in the dataset.

Confidence Interval:

Confidence intervals are used by statisticians to find the measure derived from sample data, offering a projected range of values expected to contain an unknown population parameter. It's mostly meant to serve as a means to gauge the level of uncertainty or variation inherent in estimating a population parameter using information obtained from a sample of that population. In this instance, we are observing every variable from our wine.csv dataset to look at where the true population parameter is found within a range of values.
The confidence interval percentage typically ranges from 90%, 95%, and 99%. In this case, we will use a 95% confidence test for every single one of our numeric variables.

```
##  [1] "fixed.acidity"        "volatile.acidity"     "citric.acid"
##  [4] "residual.sugar"       "chlorides"            "free.sulfur.dioxide"
##  [7] "total.sulfur.dioxide" "density"              "pH"
## [10] "sulphates"            "alcohol"              "quality"
## [13] "red"
```

Confidence Interval for Fixed.acidity of the two wines:

Red Wine:

```
##  num [1:300] 8.1 9.6 7.7 7.1 8.3 8.8 6.2 7 7.7 6.6 ...
```

```
## [1] 8.172435 8.571565
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 6.6 6.4 7.3 6.8 7.5 7.3 7.8 5.9 7.4 7.4 ...
```

```
## [1] 6.709790 6.886877
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Volatile.acidity of the two wines:

Red Wine:

```
##  num [1:300] 0.67 0.68 1 0.34 0.65 ...
```

```
## [1] 0.4746139 0.5149194
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 0.41 0.24 0.25 0.26 0.705 0.32 0.26 0.29 0.41 0.21 ...
```

```
## [1] 0.2667781 0.2898219
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Citric.acid of the two wines:

Red Wine:

```
##  num [1:300] 0.55 0.24 0.15 0.28 0.1 0.26 0.08 0 0.26 0.03 ...
```

```
## [1] 0.2624848 0.3064485
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 0.24 0.29 0.29 0.22 0.1 0.48 0.44 0.16 0.66 0.3 ...
```

```
## [1] 0.3186175 0.3468492
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Residual.sugar of the two wines:

Red Wine:

```
##  num [1:300] 1.8 2.2 2.1 2 2.9 1.6 2 1.7 1.9 7.8 ...
```

```
## [1] 2.343332 2.603002
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 4.9 11.4 7.5 7.7 13 13.3 1.3 7.9 10.8 7.9 ...
```

```
## [1] 5.826165 6.970501
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Chlorides of the two wines:

Red Wine:

```
##  num [1:300] 0.117 0.087 0.102 0.082 0.089 0.088 0.09 0.052 0.062 0.079 ...
```

```
## [1] 0.07834326 0.08623674
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 0.158 0.051 0.049 0.047 0.044 0.06 0.037 0.044 0.051 0.039 ...
```

```
## [1] 0.04329032 0.04856968
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Free.sulfur.dioxide of the two wines:

Red Wine:

```
##  num [1:300] 32 5 11 31 17 16 32 3 9 6 ...
```

```
## [1] 15.21798 17.54869
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 47 32 38 57 44 57 43 48 77 14 ...
```

```
## [1] 34.38674 37.92659
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Total.sulfur.dioxide of the two wines:

Red Wine:

```
##  num [1:300] 141 28 32 68 40 23 44 8 31 12 ...
```

```
## [1] 41.06958 47.77709
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 144 166 158 210 214 196 132 197 194 118 ...
```

```
## [1] 135.2539 144.0027
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Density of the two wines:

Red Wine:

```
##  num [1:300] 0.997 0.999 0.996 0.997 0.998 ...
```

```
## [1] 0.9963310 0.9967732
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 0.995 0.997 0.997 0.996 0.997 ...
```

```
## [1] 0.9935691 0.9942679
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for pH of the two wines:

Red Wine:

```
##  num [1:300] 3.17 3.14 3.23 3.45 3.29 3.32 3.45 3.41 3.39 3.52
```

```
## [1] 3.297434 3.331700
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 3.17 3.31 3.43 3.1 3.1 3.04 3.18 3.21 3.05 2.96 ..
```

```
## [1] 3.166980 3.199286
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Sulphates of the two wines:

Red Wine:

```
##  num [1:300] 0.62 0.6 0.48 0.48 0.55 0.47 0.58 0.47 0.64 0.5 ...
```

```
## [1] 0.6554214 0.6971119
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 0.49 0.45 0.38 0.47 0.5 0.5 0.65 0.36 0.46 0.34 ...
```

```
## [1] 0.4773138 0.5052862
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Alcohol of the two wines:

Red Wine:

```
##  num [1:300] 9.4 10.2 10 9.4 9.5 9.4 10.5 10.3 9.6 12.2 ...
```

```
## [1] 10.51888 10.77568
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

```
##  num [1:300] 9.4 9.5 9.6 9 9.1 9.2 10 9.4 8.7 10.4 ...
```

```
## [1] 10.43397 10.72532
## attr(,"conf.level")
## [1] 0.95
```

Confidence Interval for Quality of the two wines:

Red Wine:

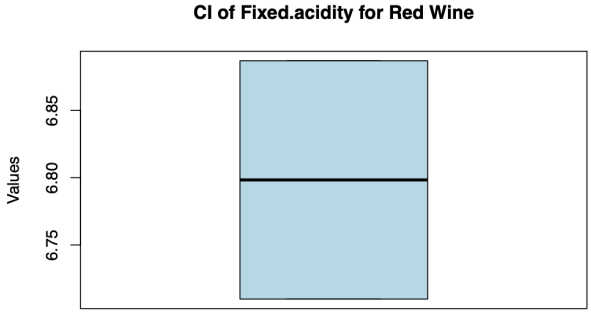
```
##  int [1:300] 5 5 5 5 5 5 5 5 5 5 ...
```

```
## [1] 5.907076 6.092924
## attr(,"conf.level")
## [1] 0.95
```

White Wine:

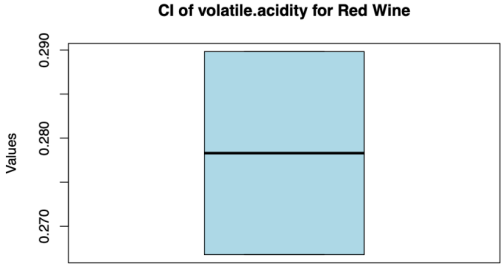
```
##  int [1:300] 5 5 5 5 5 5 5 5 5 5 ...
```

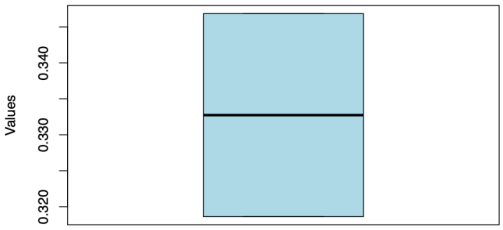
```
## [1] 5.907076 6.092924
## attr(,"conf.level")
## [1] 0.95
```

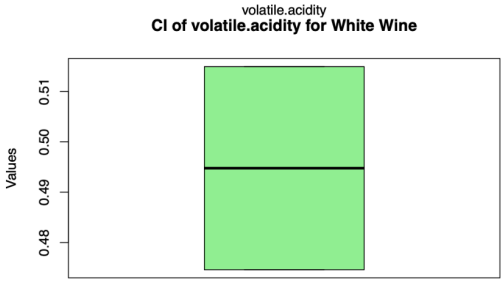

CI of Fixed.acidity for Red Wine
Values
6.75
6.80
6.85
Fixed.acidity

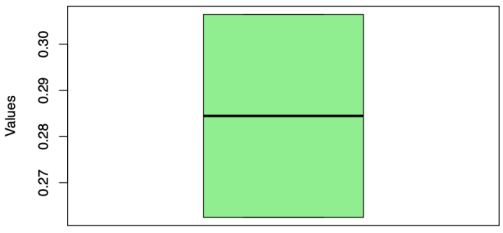

CI of Fixed.acidity for White Wine
Values
8.2
8.3
8.4
8.5
Fixed.acidity

```
## [1] "--------------------------------------------------------------------------------"
```


CI of volatile.acidity for Red Wine
Values
0.270
0.280
0.290
volatile.acidity

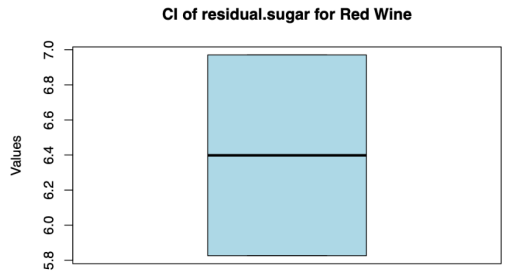

CI of citiric.acid for Red Wine
Values
0.320
0.330
0.340
citric.acid

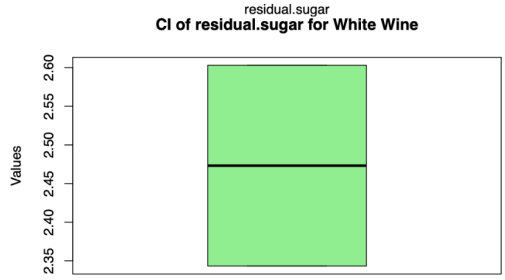

CI of volatile.acidity for White Wine
Values
0.48
0.49
0.50
0.51
volatile.acidity


CI of citiric.acid for White Wine
Values
0.27
0.28
0.29
0.30
citiric.acid

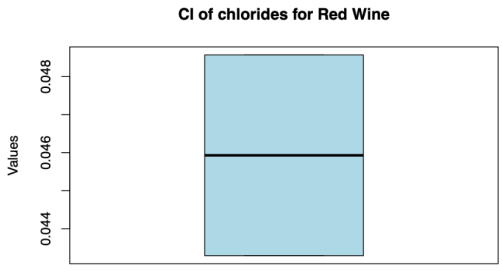
```
## [1] "--------------------------------------------------------------------------------"
```

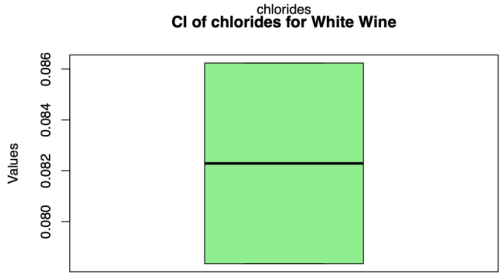
```
## [1] "--------------------------------------------------------------------------------"
```


CI of residual.sugar for Red Wine
Values
5.8
6.0
6.2
6.4
6.6
6.8
7.0
residual.sugar

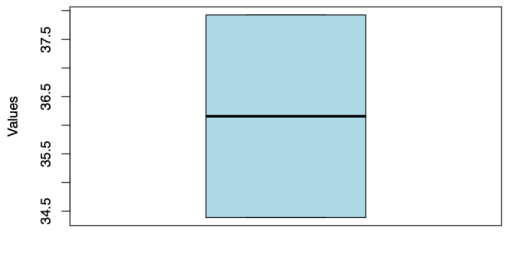

CI of residual.sugar for White Wine
Values
2.35
2.40
2.45
2.50
2.55
2.60
residual.sugar

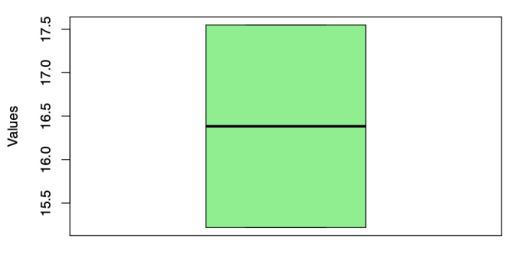
[1] "----------"


CI of chlorides for Red Wine
Values
0.044
0.046
0.048
chlorides

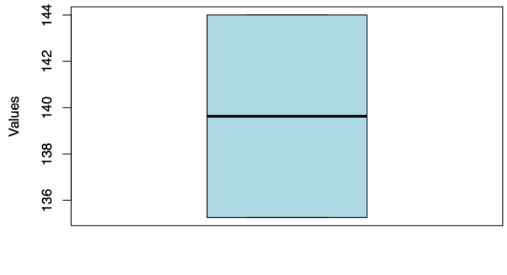

CI of chlorides for White Wine
Values
0.080
0.082
0.084
0.086
chlorides

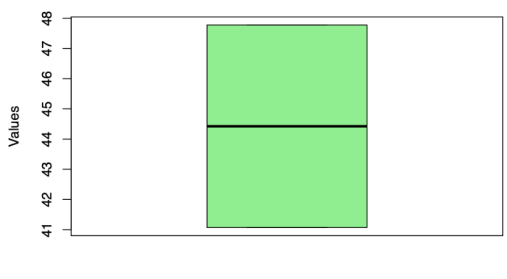
[1] "----------"


CI of free.sulfur.dioxide for Red Wine
Values
34.5
35.5
36.5
37.5
free.sulfur.dioxide

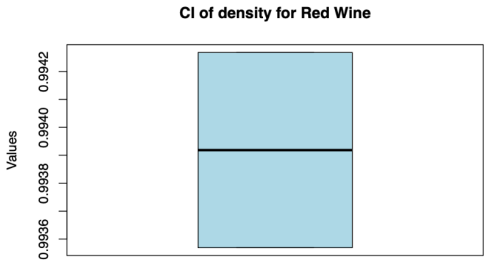

CI of free.sulfur.dioxide for White Wine
Values
15.5
16.0
16.5
17.0
17.5
free.sulfur.dioxide

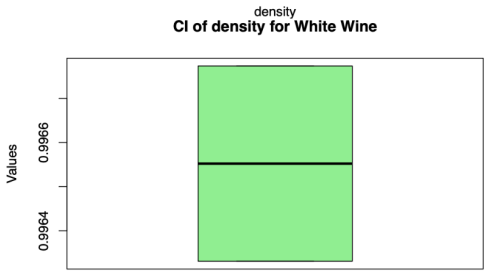
[1] "----------"


CI of total.sulfur.dioxide for Red Wine
Values
136
138
140
142
144
total.sulfur.dioxide


CI of total.sulfur.dioxide for White Wine
Values
41
42
43
44
45
46
47
48
total.sulfur.dioxide

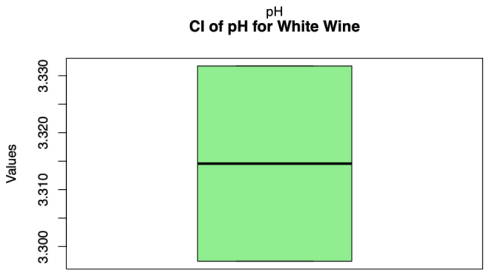
[1] "----------"


CI of density for Red Wine
Values
0.9936
0.9938
0.9940
0.9942
density

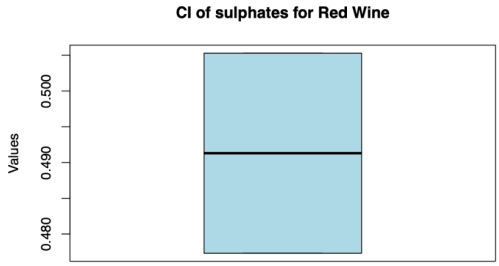

CI of density for White Wine
Values
0.9964
0.9966
density

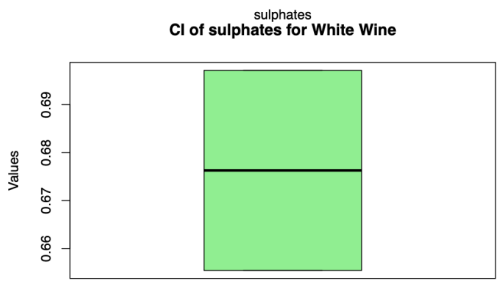
```
## [1] "--------------------------------------------------------------------------------"
```


CI of pH for Red Wine
Values
3.170
3.180
3.190
3.200
pH

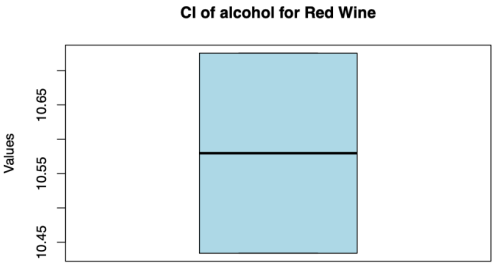

CI of pH for White Wine
Values
3.300
3.310
3.320
3.330
pH

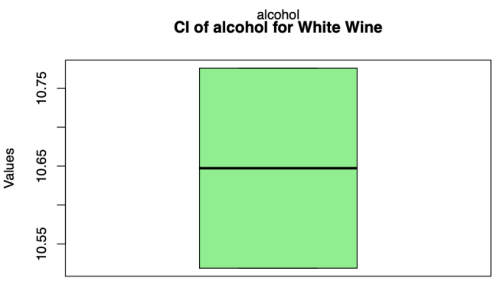
```
## [1] "--------------------------------------------------------------------------------"
```


CI of sulphates for Red Wine
Values
0.480
0.490
0.500
sulphates

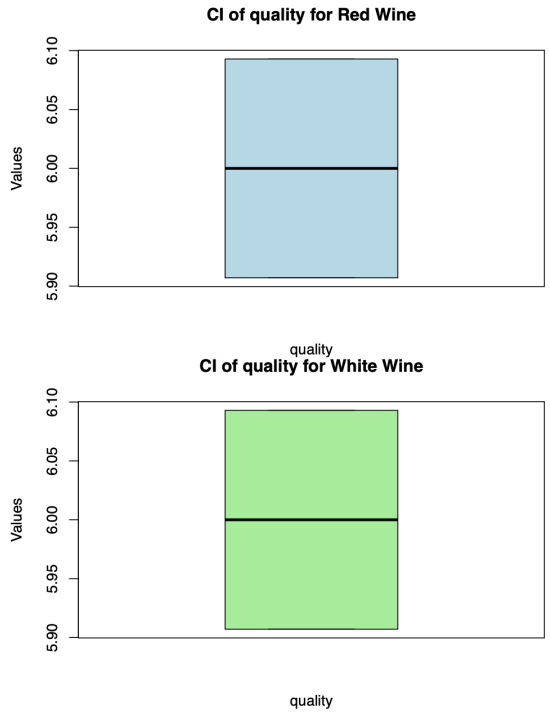

CI of sulphates for White Wine
Values
0.66
0.67
0.68
0.69
sulphates

```
## [1] "--------------------------------------------------------------------------------"
```


CI of alcohol for Red Wine
Values
10.45
10.55
10.65
alcohol


CI of alcohol for White Wine
Values
10.55
10.65
10.75
alcohol

```
## [1] "--------------------------------------------------------------------------------"
```



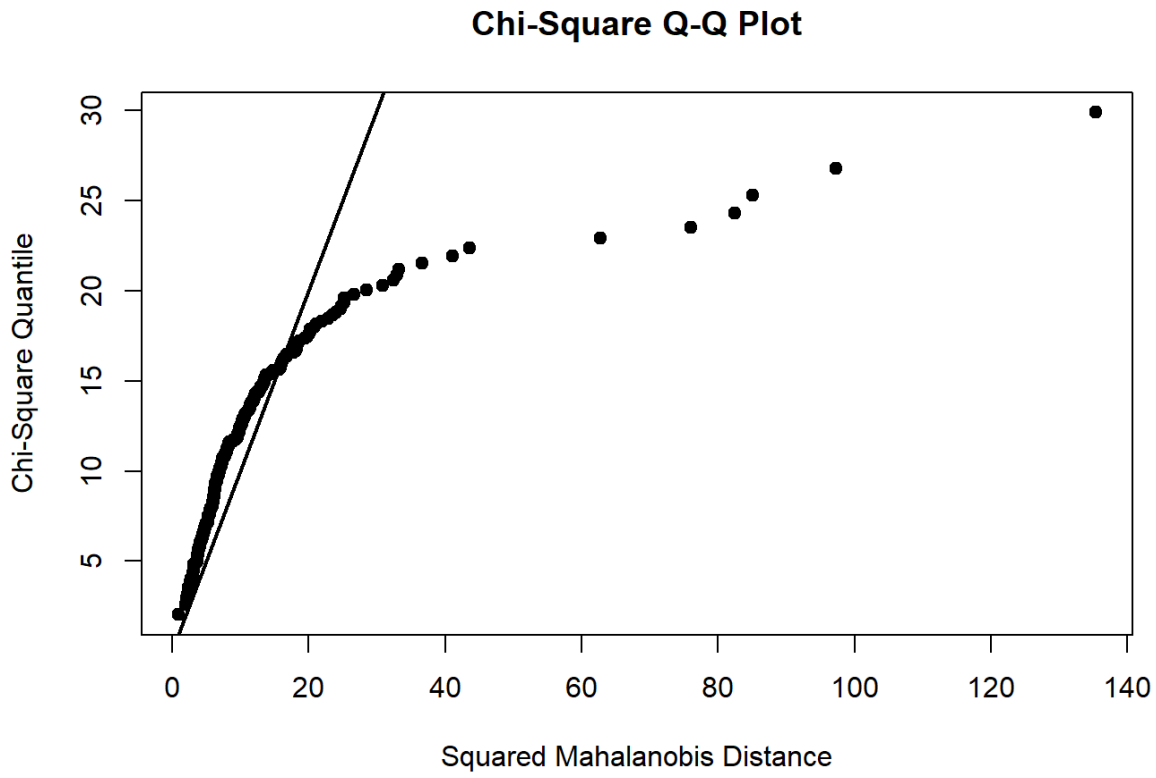
When looking completely at the confidence intervals of the two wines, we notice that red wine had a higher true population value over white wine in fixed. acidity, volatile.acidity, chlorides, and sulfates. White wine had the upper hand in citric. acid, residual.sugar, free.sulfur.dioxide, and total.sulfur.dioxide. The other variable columns saw an overlap between the two wines in density, pH, alcohol, and quality. So it is interesting to note however that the gap in true population value between the two types of sulfur dioxide saw a much more noticeable gap from white wine to red wine compared to any other variable column.

Multivariate normality Assumption and Transformation:

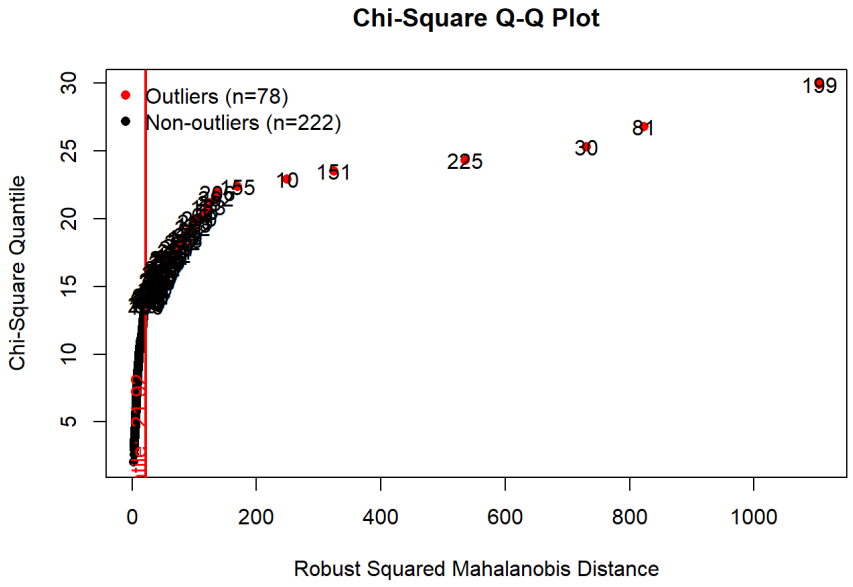
To explore the potential difference between red and white wine, access the multi-variate normality of the dataset “wine”. We break down the whole dataset “wine” based on the category variable “red.” The two new datasets are remanded as “red_wine” and “white_wine.”

Red Wine

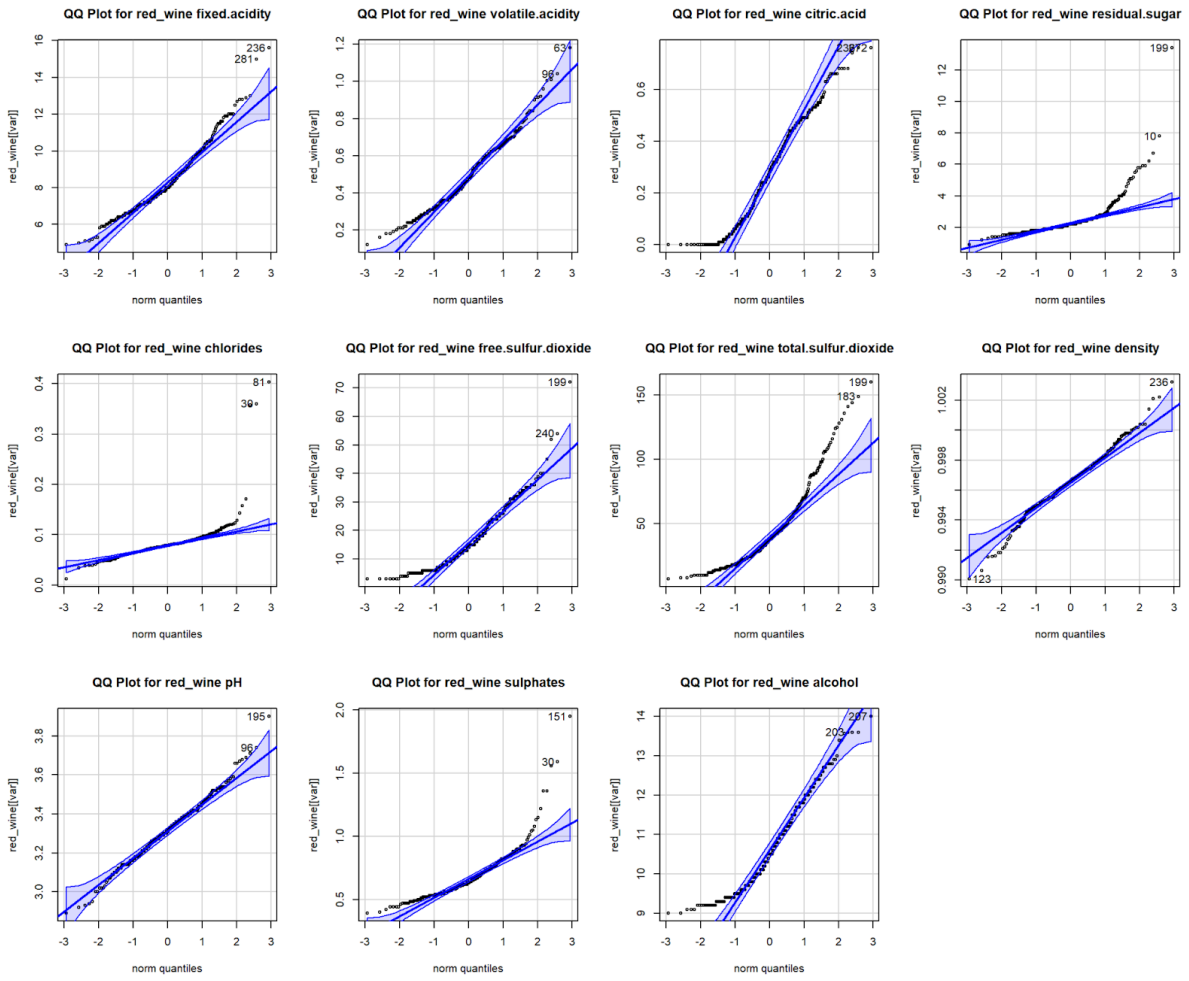
To access the multi-variate normality of the dataset “red_wine”, we need to visualize the distribution by Chi-Square QQ-plot.



The first Chi-Square QQ plot shows the squared Mahalanobis distances of our observations against the expected Chi-Square quantiles. Points that lie on the line are consistent with the Chi-Square distribution. However, points that deviate significantly from the line could be regarded as outliers.



The second Chi-Square QQ plot shows a more detailed version of the previous plot. This plot uses robust squared Mahalanobis distances, potentially more accurately identifying outliers. Also, the plot labels red points as outliers to visualize them. The result visualizes a huge amount of outliers. To look deeper at the effect of outliers, we need to plot a QQ plot for each variable.



To supplement my finding regarding the red_wine, we plot all the continuous variables in the QQ plot to see if any distribution is normally distributed. Most variables have a heavy tail skew distribution, which is not normally distributed. We suspect the outliers were the cause.

```
##              Test        Statistic p value Result
## 1 Mardia Skewness 5946.68533618899       0     NO
## 2 Mardia Kurtosis 80.7433242334724       0     NO
## 3             MVN              <NA>    <NA>     NO
```

```
##              Test             Variable Statistic   p value Normality
## 1  Anderson-Darling    fixed.acidity        3.9031  <0.001      NO
## 2  Anderson-Darling   volatile.acidity      1.9729  <0.001      NO
## 3  Anderson-Darling     citric.acid         3.0079  <0.001      NO
## 4  Anderson-Darling    residual.sugar      27.2558  <0.001      NO
## 5  Anderson-Darling      chlorides         27.0585  <0.001      NO
## 6  Anderson-Darling free.sulfur.dioxide     6.3067  <0.001      NO
## 7  Anderson-Darling total.sulfur.dioxide   10.6027  <0.001      NO
## 8  Anderson-Darling       density           1.0678  0.0083      NO
## 9  Anderson-Darling          pH             0.4745  0.2393      YES
## 10 Anderson-Darling      sulphates          9.7339  <0.001      NO
## 11 Anderson-Darling       alcohol           5.0542  <0.001      NO
```

After visualizing the data with the QQ-plot, we tried to reaccess multi-variate normality using Mardia's and univariate normality tests. The result is clear: the red_wine dataset is not normally distributed with a huge stat value and zero p-values to conclude non-normal. Only the variable "pH" is normally distributed.

Conclusion:
The dataset "red_wine" is not normally distributed with huge numbers of outliers. Thus, it is required to remove outliers and perform a transformation to make it more closely aligned with normal distribution for later analysis.

Transformation(red_wine)

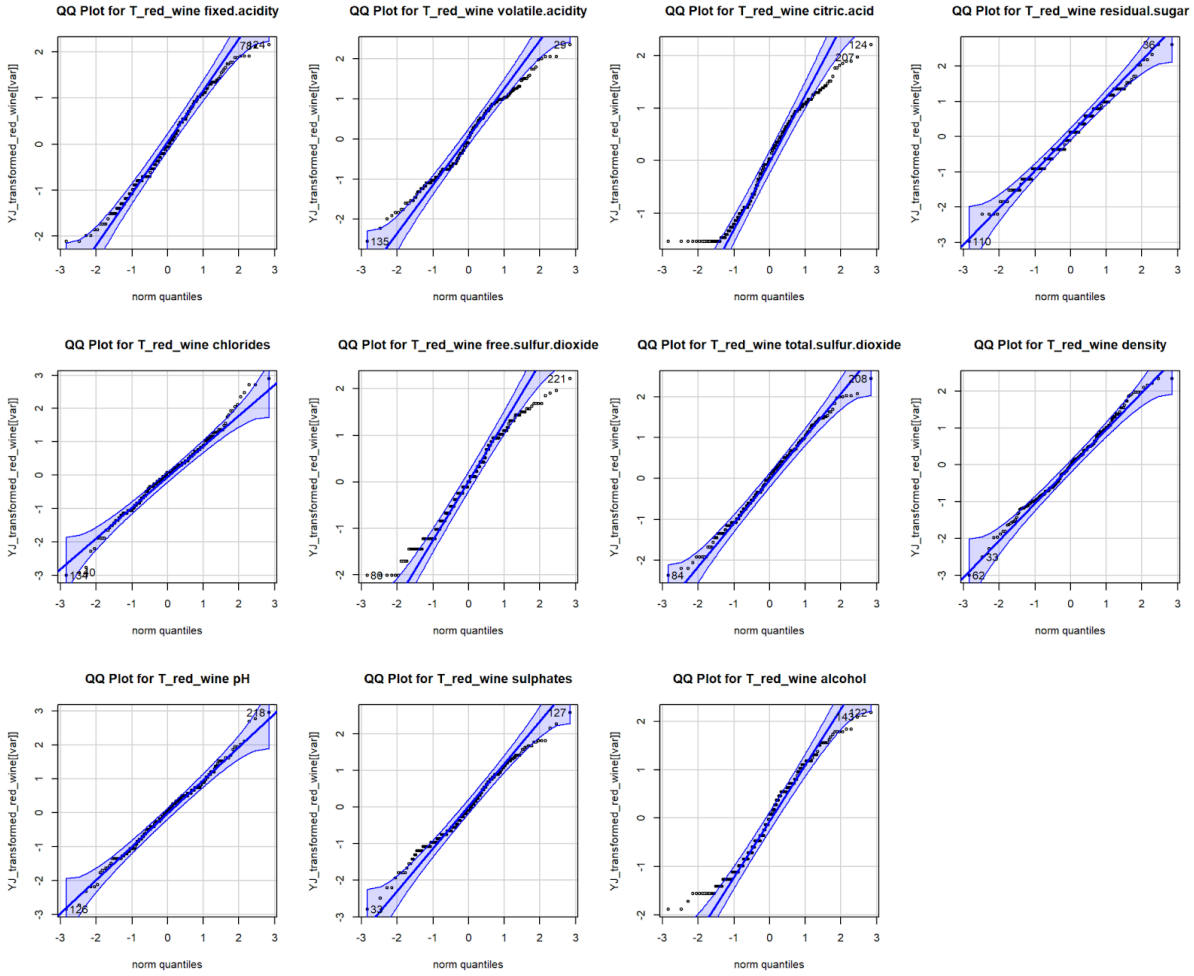
For transformation, we choose the Yeo-Johnson transformation because this transformation can handle both positive and negative values. It also handles zero values.

```
##              Test         Statistic              p value Result
## 1 Mardia Skewness  481.595739152274 3.69644097344349e-12     NO
## 2 Mardia Kurtosis -1.75635142261963   0.0790284219357107    YES
## 3             MVN              <NA>                 <NA>     NO
```

```
##               Test             Variable Statistic   p value Normality
## 1  Anderson-Darling    fixed.acidity        0.9275  0.0183      NO
## 2  Anderson-Darling   volatile.acidity      1.9564   1e-04      NO
## 3  Anderson-Darling     citric.acid         2.7901  <0.001      NO
## 4  Anderson-Darling    residual.sugar       0.7399  0.0532      YES
## 5  Anderson-Darling      chlorides          0.6978  0.0676      YES
## 6  Anderson-Darling free.sulfur.dioxide     1.3638  0.0015      NO
## 7  Anderson-Darling total.sulfur.dioxide    0.3782  0.4046      YES
## 8  Anderson-Darling       density           0.4464  0.2792      YES
## 9  Anderson-Darling          pH             0.4603  0.2584      YES
## 10 Anderson-Darling      sulphates          1.0488  0.0092      NO
## 11 Anderson-Darling       alcohol           2.1559  <0.001      NO
```

After removing outliers and performing the Yeo-Johnson transformation, we tried to reaccess multivariate normality using Mardia's and univariate normality tests. The result is much better this time. The Mardia's test confirms Mardia's Kurtosis and half of the variables are normally distributed.

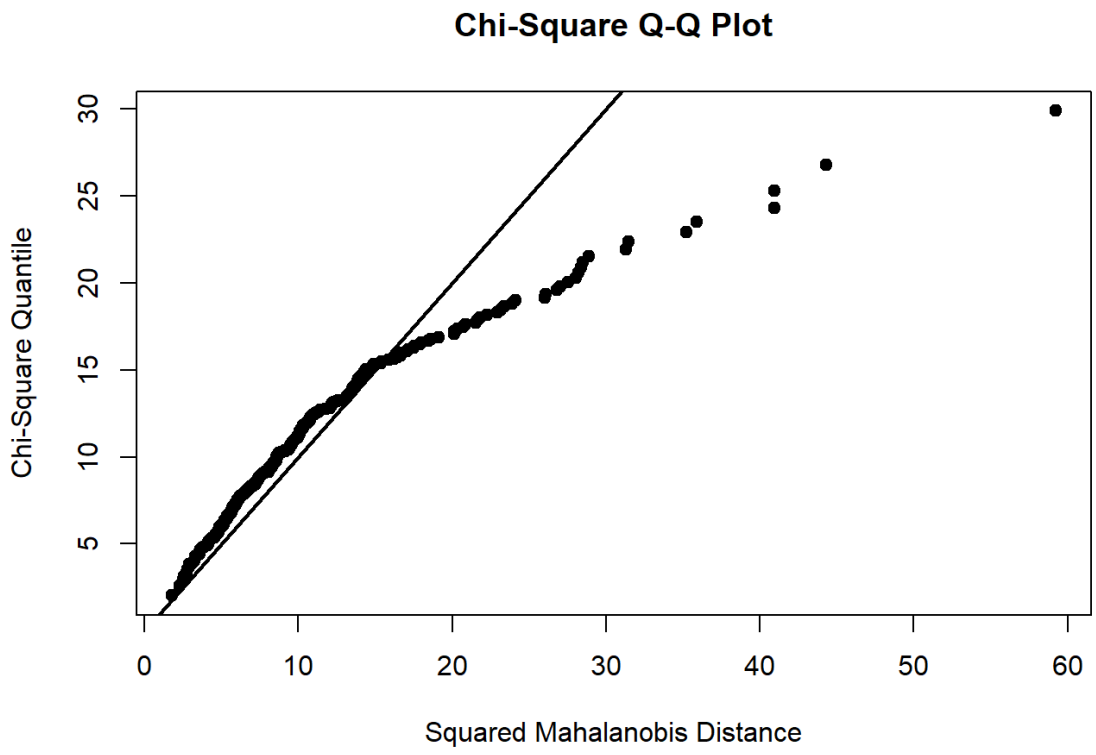
To supplement my result after transformation, we plot a QQ plot for each variable to visualize each distribution.



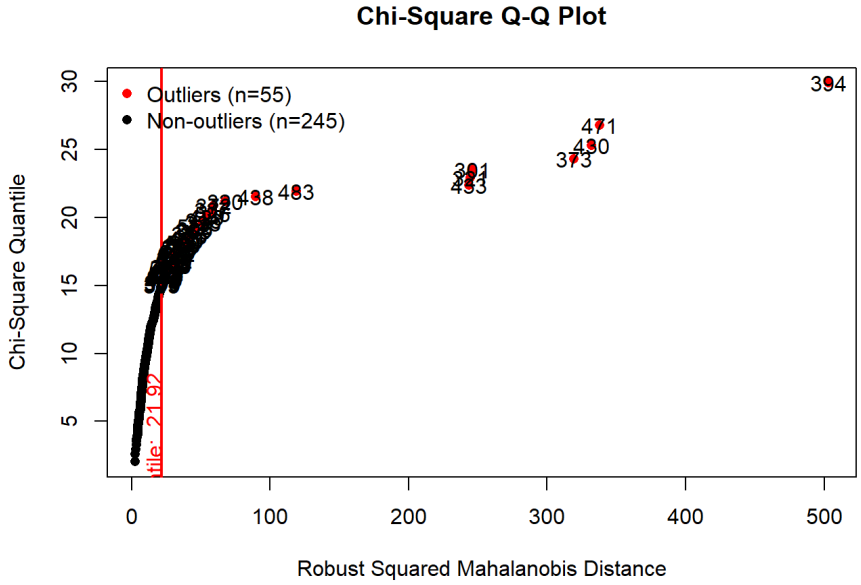
The result is much better; we can see most heavy-tailed behavior disappearing. While slight or mild skew remains, the newly transformed data is more closely aligned with a normal distribution.

White Wine

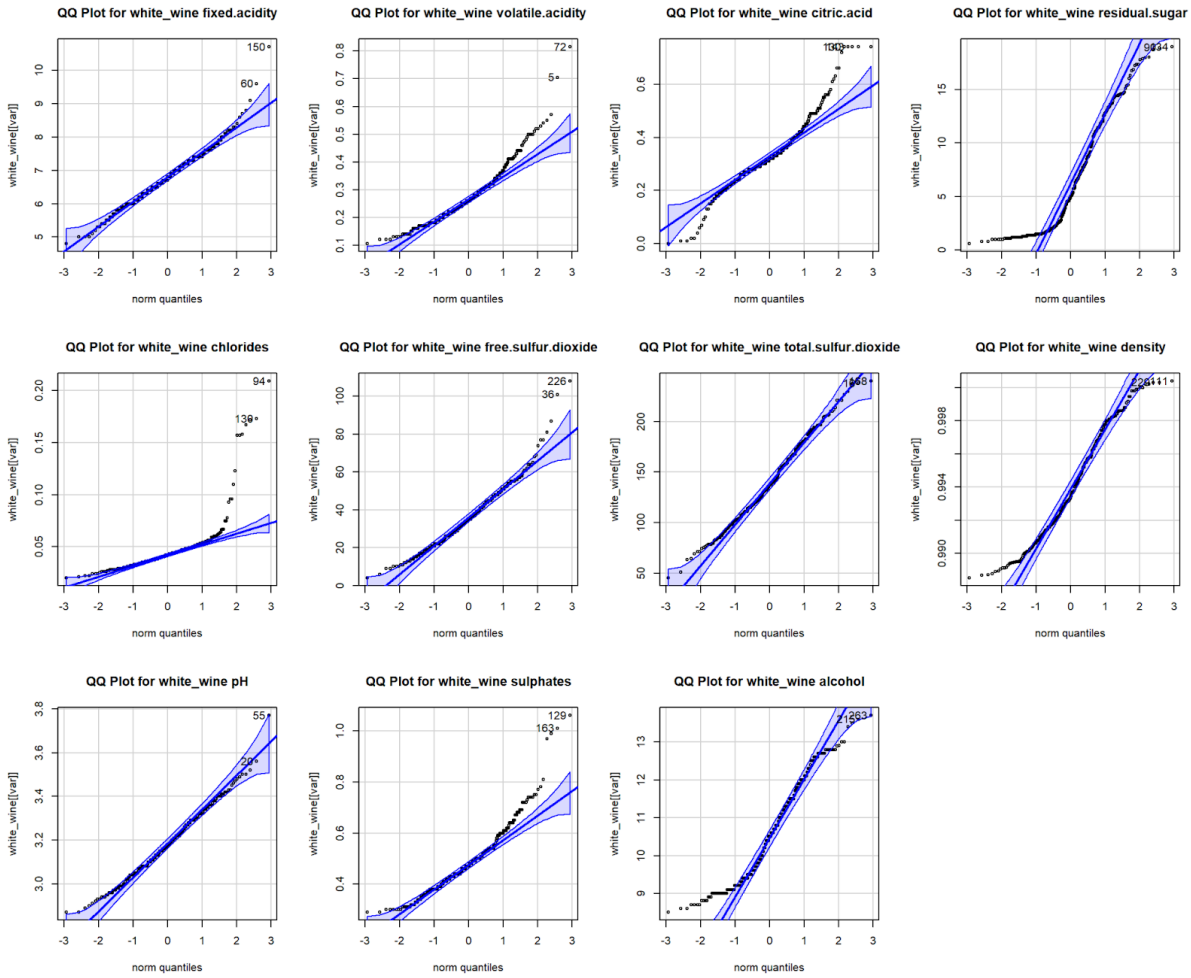
We repeat the same steps to explore the multi-variate normality. First, to access the multi-variate normality of the dataset “white_wine”, we need to visualize the distribution by Chi-Square QQ-plot.



The first Chi-Square QQ plot of White_wine shows the squared Mahalanobis distances of our observations against the expected Chi-Square quantiles. Points that lie on the line are consistent with the Chi-Square distribution. However, points that deviate significantly from the line could be regarded as outliers.



The second Chi-Square QQ plot shows a more detailed version of the previous plot. This plot uses robust squared Mahalanobis distances, potentially more accurately identifying outliers. Also, the plot labels red points as outliers to visualize them. The result visualizes a huge amount of outliers. Unlike the red_wine, white_wine has less outliers. To look deeper at the effect of outliers, we need to plot a QQ plot for each variable.



To supplement my finding regarding the white_wine, we plotted all the continuous variables in the QQ plot to see if any distribution was normally distributed. Although they have fewer outliers, most variables have a heavy tail skew distribution, which is not normally distributed. Again, we suspect the outliers were the cause.

```
##              Test        Statistic                p value Result
## 1 Mardia Skewness 1924.98124866913 1.89038893744207e-240     NO
## 2 Mardia Kurtosis 17.3440159640091                      0     NO
## 3             MVN             <NA>                   <NA>     NO
```

```
##                Test             Variable Statistic   p value Normality
## 1  Anderson-Darling    fixed.acidity        0.8985  0.0216      NO
## 2  Anderson-Darling   volatile.acidity      5.0718  <0.001      NO
## 3  Anderson-Darling     citric.acid         4.9575  <0.001      NO
## 4  Anderson-Darling    residual.sugar      12.0003  <0.001      NO
## 5  Anderson-Darling      chlorides         30.5849  <0.001      NO
## 6  Anderson-Darling free.sulfur.dioxide     1.4737   8e-04      NO
## 7  Anderson-Darling total.sulfur.dioxide    0.9299  0.0181      NO
## 8  Anderson-Darling       density           3.1541  <0.001      NO
## 9  Anderson-Darling          pH             0.4826  0.2286      YES
## 10 Anderson-Darling      sulphates         4.2702  <0.001      NO
## 11 Anderson-Darling       alcohol          4.9706  <0.001      NO
```

After visualizing the data with the QQ-plot, we tried to reaccess multi-variate normality using Mardia's and univariate normality tests. The result is clear: the white_wine dataset is also not normally distributed with a huge stat value and zero p-values to conclude non-normal. Like the red_wine, only the variable "pH" is normally distributed.

Conclusion:
The dataset "white_wine" has the same issue as the "red_wine". They were not normally distributed with the impact of too many outliers. Thus, it is required to remove outliers and perform a transformation to make it more closely aligned with normal distribution for later analysis.

Transformation(white_wine)

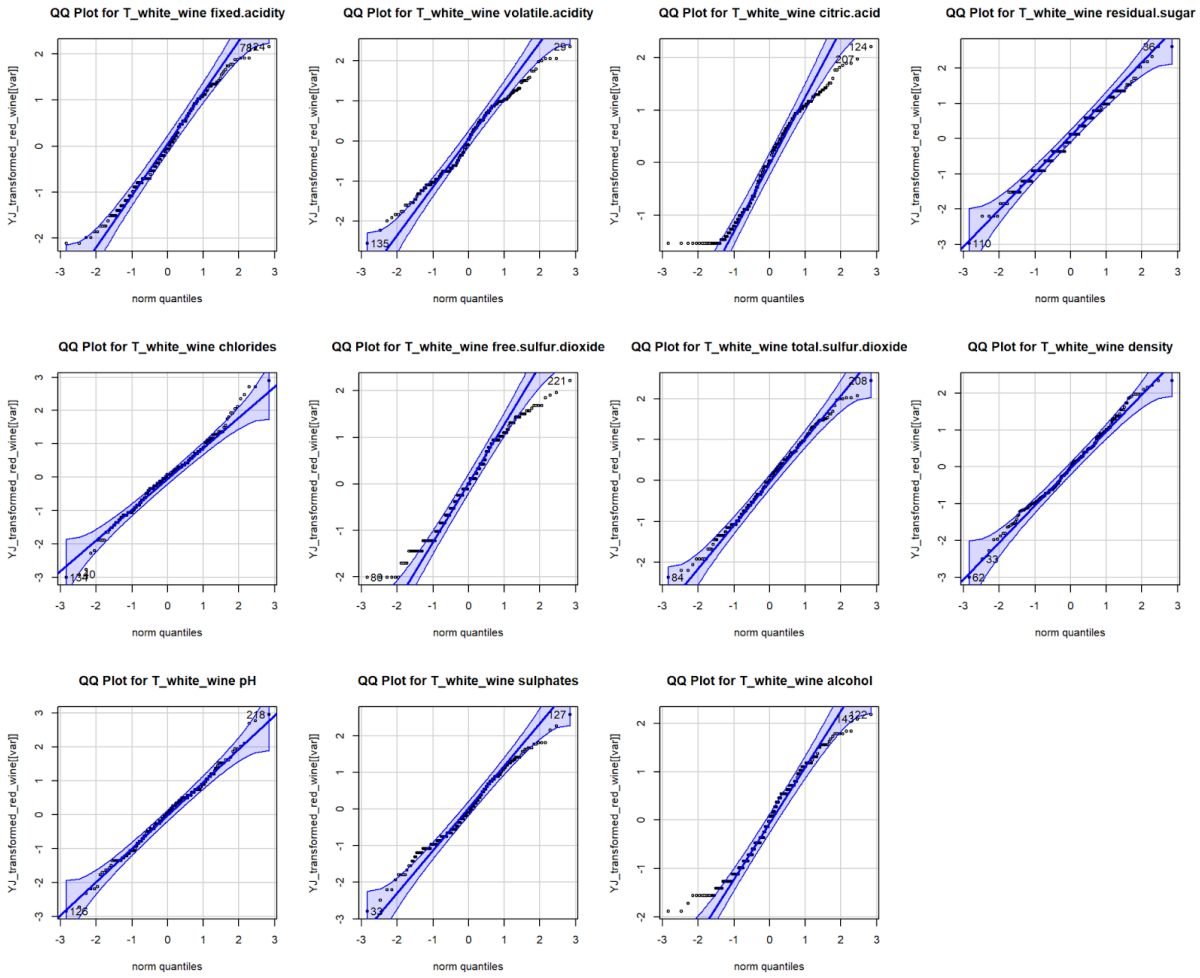
For transformation, we choose the Yeo-Johnson transformation because this transformation can handle both positive and negative values. It also handles zero values.

```
##              Test          Statistic              p value Result
## 1 Mardia Skewness   447.378260570484 3.16862943855091e-09     NO
## 2 Mardia Kurtosis -0.742542632912486    0.457758631364035    YES
## 3             MVN               <NA>                 <NA>     NO
```

```
##                Test             Variable Statistic   p value Normality
## 1  Anderson-Darling    fixed.acidity        0.5235  0.1809      YES
## 2  Anderson-Darling   volatile.acidity      0.4324  0.3018      YES
## 3  Anderson-Darling     citric.acid         1.9394   1e-04      NO
## 4  Anderson-Darling    residual.sugar       6.5998  <0.001      NO
## 5  Anderson-Darling      chlorides          0.4200   0.323      YES
## 6  Anderson-Darling free.sulfur.dioxide     0.6446  0.0916      YES
## 7  Anderson-Darling total.sulfur.dioxide    0.5579  0.1482      YES
## 8  Anderson-Darling       density           3.5806  <0.001      NO
## 9  Anderson-Darling          pH             0.2791  0.6447      YES
## 10 Anderson-Darling      sulphates          0.3964  0.3671      YES
## 11 Anderson-Darling       alcohol           3.1319  <0.001      NO
```

After removing outliers and performing the Yeo-Johnson transformation, we tried to reaccess multi-variate normality using Mardia's and univariate normality tests. The result is very similar to the red_wine. The Mardia's test confirms Mardia's Kurtosis and half of the variables are normally distributed.

To supplement my result after transformation, we plot a QQ plot for each variable to visualize each distribution.



The result is much better; we can see most heavy-tailed behavior disappearing. While slight or mild skew remains, the newly transformed data is more closely aligned with a normal distribution.

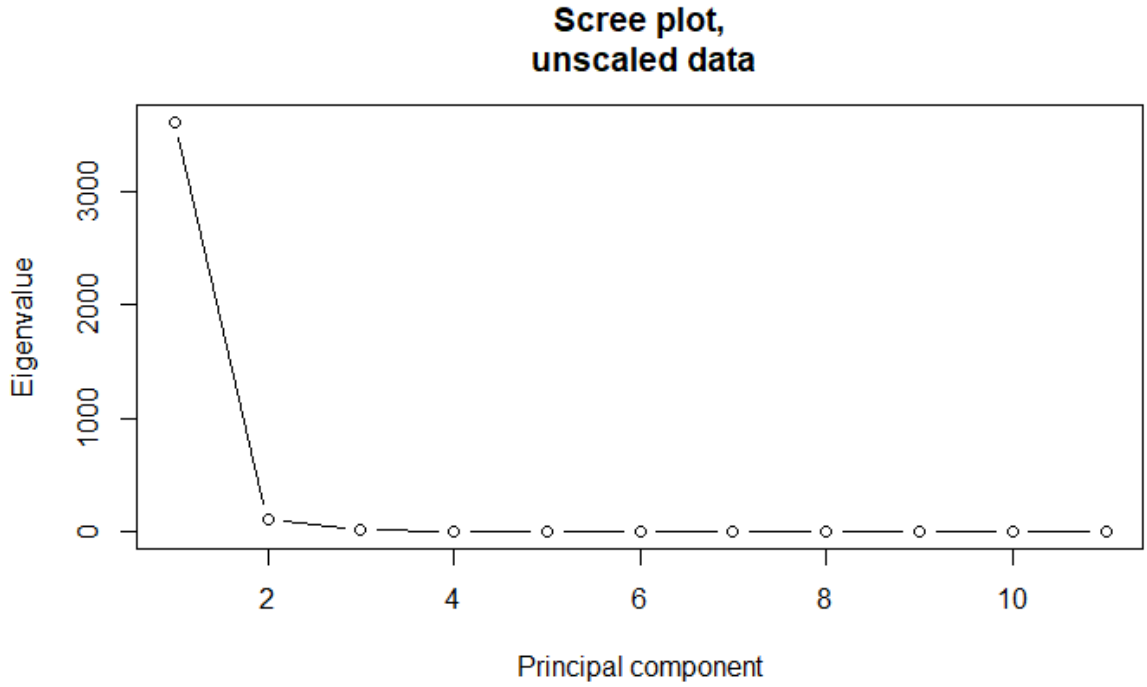
Mean vector comparison:
For red and white wine:

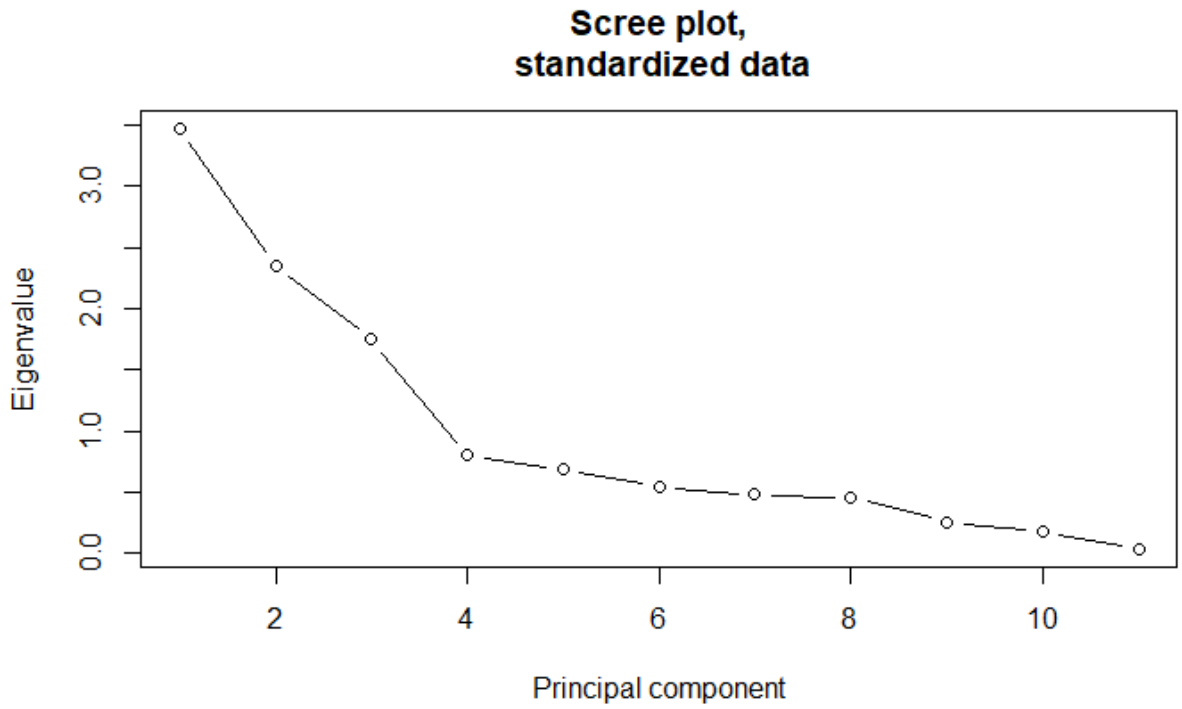
```
> #H test with unscaled data
> subject1 <- data[data$red == 0, -c(12,13)]
> subject2 <- data[data$red == 1, -c(12,13)]
> H <- hotelling.test(subject1, subject2)
> H
Test stat:  4367.4
Numerator df:  11
Denominator df:  588
P-value:  0
>
> PCA <- prcomp(data[,-c(12,13)])
> pc1 <- PCA$x[data$red == 0, 1:2]
> pc2 <- PCA$x[data$red == 1, 1:2]
> H <- hotelling.test(pc1, pc2)
> H
Test stat:  1161.7
Numerator df:  2
Denominator df:  597
P-value:  0
>
> #H test with scaled data
> stddata <- data
> stddata[, -c(12,13)] <- scale(stddata[, -c(12,13)])
> stdSubject1 <- stddata[data$red == 0, -c(12,13)]
> stdSubject2 <- stddata[data$red == 1, -c(12,13)]
> H <- hotelling.test(stdSubject1, stdSubject2)
> H
Test stat:  4367.4
Numerator df:  11
Denominator df:  588
P-value:  0
>
> stdPCA <- prcomp(stddata[,-c(12,13)])
> stdPc1 <- stdPCA$x[data$red == 0, 1:2]
> stdPc2 <- stdPCA$x[data$red == 1, 1:2]
> H <- hotelling.test(stdPc1, stdPc2)
> H
Test stat:  2273.5
Numerator df:  2
Denominator df:  597
P-value:  0
```

So we understand in statistics that data is a very integral part of finding this information out as it carries all the necessities to solve. On a basic level, we can use mean, median, mode, etc. to find the basic information that we would need to tell if one flavor of wine makes a difference over the other. However, in order to

truly understand the difference we have to use a lot of other information that can appear a bit complicated but have a very simple way of figuring out our more complex answers. For example, we can use histograms, box plots, QQ plots, etc. to give ourselves a visual representation of how everything connects such as the difference red wine may have over white wine and vice versa. They may not explain everything but they do a good job of pointing us in the right direction.

We subset the data into two parts: red wine and white wine. Then we test for the difference between them. Since we get the p-value equal to zero, we have strong evidence to show that the two groups have differences from each other. After standardizing the data and applying the hotelling test again, we got the same p-value. Additionally, we did the same test with the first eight principal components (explanation below), the p-value is 0 before and after scaling the data.





From the scree plot with unscaled data, we can see a large proportion of the variation in the data is explained by the first principle component, then a small proportion by the second, and a tiny proportion by the third, fourth, and fifth, and so on. The first two components explained most of the data in this plot.

Additionally, in the scree plot with standardized data, we can see that a large proportion of the variation is explained by the four components. Then, it begins to level out after the eighth component. This indicates that we should be using eight principal components instead of two.

```
Analysis of Variance Table

             Df   Wilks  approx F num Df den Df    Pr(>F)
(Intercept)   1 0.00000 129775123     11    588 < 2.2e-16 ***
red           1 0.12043       390     11    588 < 2.2e-16 ***
Residuals   598
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the ANOVA test, since the p-value is nearly zero, we can safely reject the null hypothesis and conclude that red wine's mean vector differs from the one in white wine.

PREDICTION OF WINE USING LDA

BEFORE transformation:

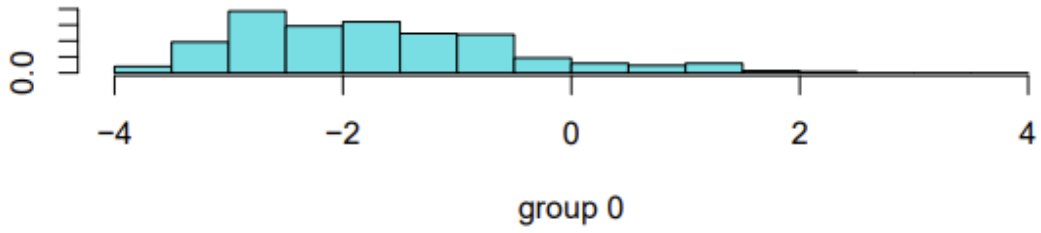
```
Group means:
  total.sulfur.dioxide   density
0             44.42333 0.9965521
1            139.62833 0.9939185

Coefficients of linear discriminants:
                              LD1
total.sulfur.dioxide    0.02973833
density              -251.97347829
[1] 0.05833333
Call:
lda(Xtraining, ytraining)

Prior probabilities of groups:
        0         1
0.4966667 0.5033333

Group means:
  total.sulfur.dioxide   density
0             48.10738 0.9965836
1            137.10265 0.9937476

Coefficients of linear discriminants:
                              LD1
total.sulfur.dioxide    0.02757745
density              -253.16693379
[1] 0.05
```

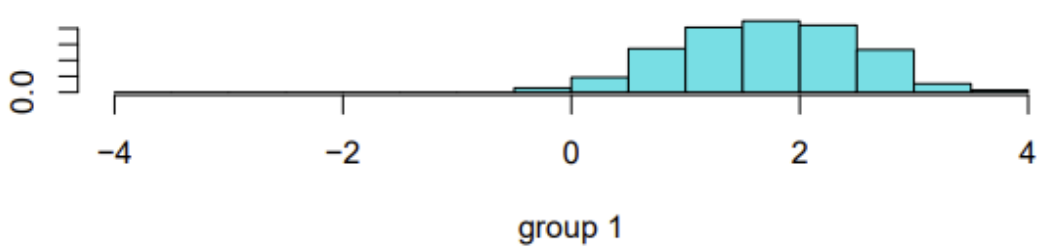
AFTER TRANSFORMATION

```
## Call:
## lda(Xtraining, ytraining)
##
## Prior probabilities of groups:
##         0         1
## 0.4966667 0.5033333
##
## Group means:
##   total.sulfur.dioxide    density
## 0           -0.7321808  0.4661951
## 1            0.7833989 -0.5140986
##
## Coefficients of linear discriminants:
##                              LD1
## total.sulfur.dioxide  1.6527039
## density              -0.6423847

## [1] 0.03666667
```

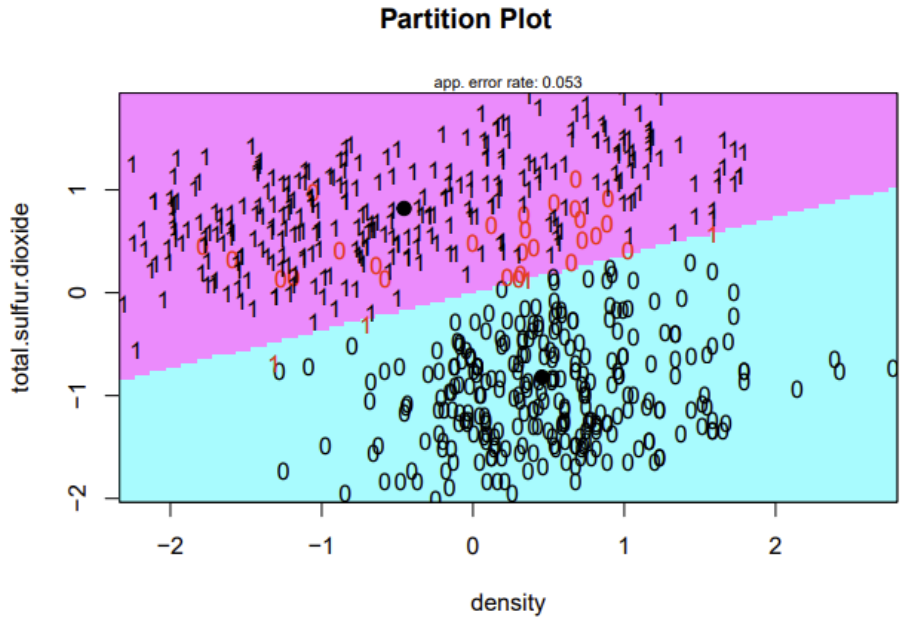
Something we can use in order to explain which wine has the higher quality would be to perform a hypothesis test. This will require us to come up with both a null hypothesis and an alternate hypothesis. When creating these tests, our ultimate goal is for us to reject the null hypothesis. We can start by saying that both the red and white wine are equal to each other, when our alternate one will be that they are indeed not equal to each other and one of them actually has a higher quality than the other. The key thing here will be to use a p-test, so we must use a variable percentage such as 5% to test the significance of our two wines and to see if one of them is indeed better than the other.

We are going to use columns 7 (total.sulfur.dioxide) and 8 (density) here to predict whether the resulting wine is red or white. The probability that p1 and p2 are both 0.5 means that half of the wine is red and the other half is white. The coefficients of linear discriminant tell us we should put more weight on density rather than total.sulfur.dioxide to classify these two variables. We check for the misclassification rate of the original data, which is around 5.33%. Then we will use testing and training sets to evaluate the misclassification rate, the results of probability and linear discriminant are quite similar to the original data, and the misclassification rate lowered a bit to 3.67%.





From the histogram on untransformed data, we can see the separation of group 0 (white wine) and group 1 (red wine) have a certain amount of overlapping. But it is sufficient enough that we can classify with only a 5% chance of misclassification.



From the partition plot on untransformed data, the points are labeled as 0s and 1s according to whether they are red or white wine. We can see a clear separation line despite a few points overlapping, so the linear discriminant is working well here.

PREDICTION OF WINE QUALITY USING LDA

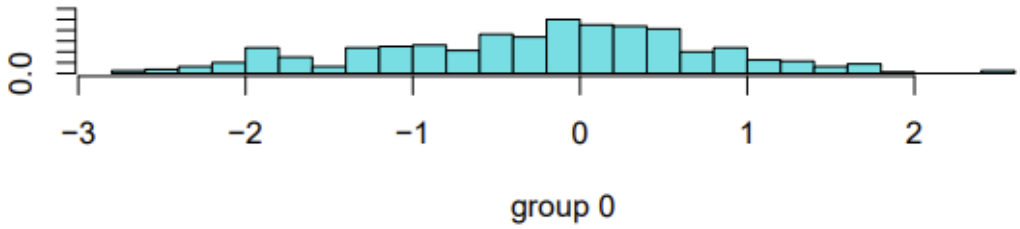
Before transformation:

```
Group means:
  total.sulfur.dioxide   density
0              97.5700 0.9957776
1              80.9375 0.9941508

Coefficients of linear discriminants:
                                LD1
total.sulfur.dioxide   -0.01056445
density              -339.40752617
[1] 0.2833333
Call:
lda(Xtraining1, ytraining1)

Prior probabilities of groups:
        0         1
0.6633333 0.3366667

Group means:
  total.sulfur.dioxide   density
0             97.84925 0.9957669
1             83.15347 0.9939528

Coefficients of linear discriminants:
                                LD1
total.sulfur.dioxide -9.949939e-03
density              -3.376899e+02
[1] 0.2866667
```

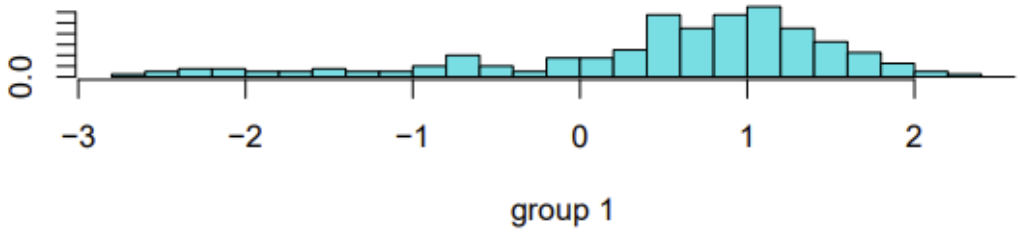
AFTER TRANSFORMATION

```
Call:
lda(Xtraining1, ytraining1)

Prior probabilities of groups:
        0         1
0.6633333 0.3366667

Group means:
  total.sulfur.dioxide    density
0            0.1143762  0.1834583
1           -0.1342829 -0.4423171

Coefficients of linear discriminants:
                            LD1
total.sulfur.dioxide -0.6506189
density              -0.9938917
>
> #misclassification rate
> prd3 <- predict(ldaFit3, Xtesting1)
> tab <- table(Predicted = prd3$class, Actual = ytesting1)
> 1-sum(diag(tab))/sum(tab)
[1] 0.2633333
>
```

We will be using the same columns to predict the wine quality. The probability p1 and p2 are 0.67 and 0.33, respectively, which tells us that the highest quality wine is 1/3 of the total proportion. The coefficients of linear discriminant tell us we should put more weight on density rather than total.sulfur.dioxide to classify these two variables. The misclassification is relatively high in this group, as it is around 28.83%. After using testing and training sets to evaluate the misclassification rate, the misclassification rate is a bit higher than the original one which is 28.33%. After using the transformed data we got a misclassification rate of 26.33% which is still high.

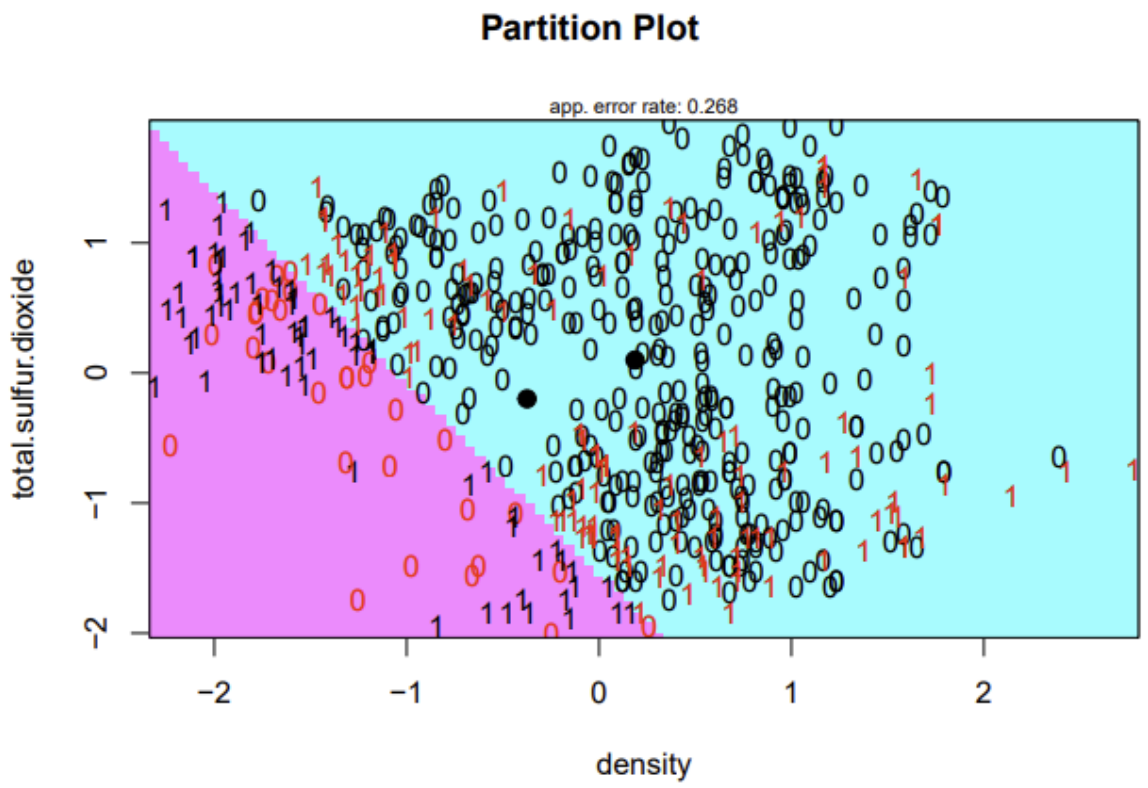


group 0



group 1

From the histogram on untransformed data, we can see the separation of group 0 and group 1 has a large amount of overlapping, which indicates it is not a good predicting model.



From the partition plot of untransformed data, the points are labeled as 0s and 1s according to the highest quality or other quality. We can see a separation line, but many points overlap. The linear discriminant is not doing well here. So we conclude that it's not accurate to predict the quality of wine based on continuous measurements.

Appendix

12/15/2023

```
library(MASS)
library(mvtnorm)
library(car)
library(MVN)
library(moments)
library(GGally)
library(bestNormalize)
library(klaR)
library(psych)
library(dplyr)

#load the data
wine=read.csv("wine.csv")
data=read.csv("wine.csv")
wine_data=read.csv("wine.csv")

#creating a new column wine representing red and white wine
data$wine=ifelse(data$red==1,"Red","White")
data<- na.omit(data)

#creating subset for red wine and white wine
Red_subset=subset(data,wine=="Red")
White_subset=subset(data,wine=="White")

#finding the means and sd of red and white wine subset
cat("\\newpage")

cat("Summary for Red Wine Subset:\n")
summary(Red_subset[,1:11])
cat("Summary for White Wine Subset:\n")
summary(White_subset[,1:11])

cat("Summary for Overall Wine:\n")
summary(data[,1:11])

#Finding Boxplot for each variable
boxplot(data[,1],main=" Boxplot for fixed.acidity")
boxplot(data[,2],main="Boxplot for volatile.acidity ")
boxplot(data[,3],main="Boxplot for citric.acid ")
boxplot(data[,4],main="Boxplot for residual.sugar ")
boxplot(data[,5],main="Boxplot for chlorides")
boxplot(data[,6],main="Boxplot for free.sulfur.dioxide")
boxplot(data[,7],main="Boxplot for total.sulfur.dioxide")
boxplot(data[,8],main="Boxplot for density" )
```

```r
boxplot(data[,9],main="Boxplot for pH" )
boxplot(data[,10],main="Boxplot for sulphates")
boxplot(data[,11],main="Boxplot for alcohol")

#Drawing plots to see any relation
plot(data[, 1:4], col = ifelse(data$red == 1, "red", "blue"))
plot(data[,5:8],col = ifelse(data$red == 1, "red", "blue"))
plot(data[,8:11],col = ifelse(data$red == 1, "red", "blue"))

cat("\nCorrelation matrix to find most and least correlated values:\n")
correlation_matrix=cor(data[,1:11])
correlation_matrix
#Most positively Correlated Variables: total.sulfur.dioxide free.sulfur.dioxide
#Most negatively Correlated Variables: alcohol density

# Perform PCA on standardzied data
pca_result <- prcomp(data[, 1:11], scale. = TRUE,center = TRUE)
summary(pca_result)

#spreeplot
plot(pca_result$sdev^2/sum(pca_result$sdev^2),type="b",
     xlab="Dimension",
     ylab="% variation explained",
     ylim=c(0, max(pca_result$sdev^2/sum(pca_result$sdev^2))),xlim=c(1,12),main="Variance explained pl

#cumulative sum and it's plot
cumsum(pca_result$sdev^2)/sum(pca_result$sdev^2)
plot(cumsum(pca_result$sdev^2)/sum(pca_result$sdev^2),
     type="b",
     xlab="Dimension",
     ylab="Cumulative % variation explained",
     ylim=1.1*c(0, 1),main="Cumulative Variance Plot")
#biplot
biplot(pca_result)

library(readr)

str(wine_data$fixed.acidity)
fixed.acidity_CI <- t.test(wine_data$fixed.acidity, conf.level = 0.95)$conf.int
print(fixed.acidity_CI)

str(wine_data$volatile.acidity)
volatile.acidity_CI <- t.test(wine_data$volatile.acidity, conf.level = 0.95)$conf.int
print(volatile.acidity_CI)

str(wine_data$citric.acid)
citric.acid_CI <- t.test(wine_data$citric.acid, conf.level = 0.95)$conf.int
print(citric.acid_CI)

str(wine_data$residual.sugar)
residual.sugar_CI <- t.test(wine_data$residual.sugar, conf.level = 0.95)$conf.int
print(residual.sugar_CI)
```

```r
str(wine_data$chlorides)
chlorides_CI <- t.test(wine_data$chlorides, conf.level = 0.95)$conf.int
print(chlorides_CI)

str(wine_data$free.sulfur.dioxide)
free.sulfur.dioxide_CI <- t.test(wine_data$free.sulfur.dioxide, conf.level = 0.95)$conf.int
print(free.sulfur.dioxide_CI)

str(wine_data$total.sulfur.dioxide)
total.sulfur.dioxide_CI <- t.test(wine_data$total.sulfur.dioxide, conf.level = 0.95)$conf.int
print(total.sulfur.dioxide_CI)

str(wine_data$density)
density_CI <- t.test(wine_data$density, conf.level = 0.95)$conf.int
print(density_CI)

str(wine_data$pH)
pH_CI <- t.test(wine_data$pH, conf.level = 0.95)$conf.int
print(pH_CI)

str(wine_data$sulphates)
sulphates_CI <- t.test(wine_data$sulphates, conf.level = 0.95)$conf.int
print(sulphates_CI)

str(wine_data$alcohol)
alcohol_CI <- t.test(wine_data$alcohol, conf.level = 0.95)$conf.int
print(alcohol_CI)

str(wine_data$quality)
quality_CI <- t.test(wine_data$quality, conf.level = 0.95)$conf.int
print(quality_CI)

boxplot(fixed.acidity_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Fixed.acidity", ylab = "Values",
        main = "CI of Fixed.acidity",
        col = c("lightblue"), border = "black")

print("-------------------------------------------------------------------------------")

boxplot(volatile.acidity_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Volatile.acidity", ylab = "Values",
        main = "CI of Volatile.acidity",
        col = c("lightblue"), border = "black")

print("-------------------------------------------------------------------------------")

boxplot(citric.acid_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Citric.acid", ylab = "Values",
        main = "CI of Citric.acid",
        col = c("lightblue"), border = "black")
```

```
print("-------------------------------------------------------------------------------")

boxplot(residual.sugar_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Residual.sugar", ylab = "Values",
        main = "CI of Residual.sugar",
        col = c("lightblue"), border = "black")

print("-------------------------------------------------------------------------------")

boxplot(chlorides_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Chlorides", ylab = "Values",
        main = "CI of Chlorides",
        col = c("lightblue"), border = "black")

print("-------------------------------------------------------------------------------")

boxplot(free.sulfur.dioxide_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Free.sulfur.dioxide", ylab = "Values",
        main = "CI of Free.sulfur.dioxide",
        col = c("lightblue"), border = "black")

print("-------------------------------------------------------------------------------")

boxplot(total.sulfur.dioxide_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Total.sulfur.dioxide", ylab = "Values",
        main = "CI of Total.sulfur.dioxide",
        col = c("lightblue"), border = "black")

print("-------------------------------------------------------------------------------")

boxplot(density_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Density", ylab = "Values",
        main = "CI of Density",
        col = c("lightblue"), border = "black")

print("-------------------------------------------------------------------------------")

boxplot(pH_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "pH", ylab = "Values",
        main = "CI of pH",
        col = c("lightblue"), border = "black")

print("-------------------------------------------------------------------------------")

boxplot(sulphates_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Sulphates", ylab = "Values",
```

```
        main = "CI of Sulphates",
        col = c("lightblue"), border = "black")

print("-------------------------------------------------------------------------------------")

boxplot(alcohol_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Alcohol", ylab = "Values",
        main = "CI of Alcohol",
        col = c("lightblue"), border = "black")

print("-------------------------------------------------------------------------------------")

boxplot(quality_CI, notch = FALSE, outline = FALSE,
        names = c("Group 1"),
        xlab = "Quality", ylab = "Values",
        main = "CI of Quality",
        col = c("lightblue"), border = "black")

# subet the data to base on category variable "red"
red_wine = subset(wine, red == "1")
white_wine = subset(wine, red == "0")

# quality and red variables are not continuous
variables <- c("fixed.acidity", "volatile.acidity", "citric.acid",
               "residual.sugar", "chlorides", "free.sulfur.dioxide", "total.sulfur.dioxide",
               "density", "pH", "sulphates", "alcohol")
# Mardia's Test for Multivariate Normality
# Show multivariate Skewness and Kurtosis
# Show Anderson-Darling test for univariateNormality
# also show outliers and chi-square
mvnRed <- mvn(red_wine[variables], mvnTest = "mardia",
                                multivariatePlot = "qq",
                                multivariateOutlierMethod = "quan",
                                showOutliers = "True",
                                showNewData = "True")

print(mvnRed$multivariateNormality)
print(mvnRed$univariateNormality)
print(mvnRed$Descriptives)

# Create QQ plots for each new variable
# Apply qqPlot to each variable
lapply(variables, function(var) {
    qqPlot(red_wine[[var]], main=paste("QQ Plot for red_wine", var))
})

# get clean data from mvn function
new_red_wine <- mvnRed$newData
head(new_red_wine)

new_mvnRed <- mvn(new_red_wine[variables], mvnTest = "mardia")
print(new_mvnRed$multivariateNormality)
```

```r
print(new_mvnRed$univariateNormality)
print(new_mvnRed$Descriptives)

# Create QQ plots for each new variable
# Apply qqPlot to each variable
lapply(variables, function(var) {
    qqPlot(new_red_wine[[var]], main=paste("QQ Plot for red_wine_new", var))
})

# Select columns for transformation, excluding 'red' and 'quality'
columns_to_transform <- setdiff(names(wine), c("red", "quality"))

# Apply Yeo-Johnson transformation
# can handle both positive and negative
# also handle zero
YJ_transformed_red_wine <- new_red_wine
for (col in columns_to_transform) {
  transformation_result <- yeojohnson(new_red_wine[[col]])
  YJ_transformed_red_wine[[col]] <- transformation_result$x.t
}

# View the transformed data
head(YJ_transformed_red_wine)

transformed_mvnRed <- mvn(YJ_transformed_red_wine[variables], mvnTest = "mardia")
print(transformed_mvnRed$multivariateNormality)
print(transformed_mvnRed$univariateNormality)
print(transformed_mvnRed$Descriptives)

# Create QQ plots for each new variable
# Apply qqPlot to each variable
lapply(variables, function(var) {
    qqPlot(YJ_transformed_red_wine[[var]], main=paste("QQ Plot for Transformed_red_wine", var))
})

# Mardia's Test for Multivariate Normality
# Show multivariate Skewness and Kurtosis
# Show Anderson-Darling test for univariateNormality
# also show outliers and chi-square
mvnWhite <- mvn(white_wine[variables], mvnTest = "mardia",
                                multivariatePlot = "qq",
                                multivariateOutlierMethod = "quan",
                                showOutliers = "True",
                                showNewData = "True")

print(mvnWhite$multivariateNormality)
print(mvnWhite$univariateNormality)
print(mvnWhite$Descriptives)

# Create QQ plots for each new variable
# Apply qqPlot to each variable
lapply(variables, function(var) {
    qqPlot(white_wine[[var]], main=paste("QQ Plot for white_wine", var))
```

```
})

# get clean data from mvn function
new_white_wine <- mvnWhite$newData
head(new_white_wine)

new_mvnWhite <- mvn(new_white_wine[variables], mvnTest = "mardia")
print(new_mvnWhite$multivariateNormality)
print(new_mvnWhite$univariateNormality)
print(new_mvnWhite$Descriptives)

# Create QQ plots for each new variable
# Apply qqPlot to each variable
lapply(variables, function(var) {
    qqPlot(new_white_wine[[var]], main=paste("QQ Plot for white_wine_new", var))
})

# Select columns for transformation, excluding 'red' and 'quality'
columns_to_transform <- setdiff(names(wine), c("red", "quality"))

# Apply Yeo-Johnson transformation
# can handle both positive and negative
# also handle zero
YJ_transformed_white_wine <- new_white_wine
for (col in columns_to_transform) {
  transformation_result <- yeojohnson(new_white_wine[[col]])
  YJ_transformed_white_wine[[col]] <- transformation_result$x.t
}

# View the transformed data
head(YJ_transformed_white_wine)

transformed_mvnWhite <- mvn(YJ_transformed_white_wine[variables], mvnTest = "mardia")
print(transformed_mvnWhite$multivariateNormality)
print(transformed_mvnWhite$univariateNormality)
print(transformed_mvnWhite$Descriptives)

# Create QQ plots for each new variable
# Apply qqPlot to each variable
lapply(variables, function(var) {
    qqPlot(YJ_transformed_red_wine[[var]], main=paste("QQ Plot for Transformed_white_wine", var))
})

library(Hotelling)
library(corpcor)

#H test with unscaled data
subject1 <- data[data$red == 0, -c(12,13)]
subject2 <- data[data$red == 1, -c(12,13)]
H <- hotelling.test(subject1, subject2)
H

PCA <- prcomp(data[,-c(12,13)])
```

```
pc1 <- PCA$x[data$red == 0, 1:2]
pc2 <- PCA$x[data$red == 1, 1:2]
H <- hotelling.test(pc1, pc2)
H

#H test with scaled data
stddata <- data
stddata[, -c(12,13)] <- scale(stddata[, -c(12,13)])
stdSubject1 <- stddata[data$red == 0, -c(12,13)]
stdSubject2 <- stddata[data$red == 1, -c(12,13)]
H <- hotelling.test(stdSubject1, stdSubject2)
H

stdPCA <- prcomp(stddata[,-c(12,13)])
stdPc1 <- stdPCA$x[data$red == 0, 1:2]
stdPc2 <- stdPCA$x[data$red == 1, 1:2]
H <- hotelling.test(stdPc1, stdPc2)
H

plot(PCA$sdev^2,
     type="b",
     xlab="Principal component",
     ylab="Eigenvalue",
     main="Scree plot,\n unscaled data")

plot(stdPCA$sdev^2,
     type="b",
     xlab="Principal component",
     ylab="Eigenvalue",
     main="Scree plot,\n standardized data")

#one-way MANOVA
Y <- as.matrix(data[,-c(12,13)])
red <- data$red
fittedModel <- manova(Y ~ red)
anova(fittedModel, test="Wilks")

library(Hotelling)
library(corpcor)

#H test with unscaled data
subject1 <- data[data$quality == 7, -c(12,13)]
subject2 <- data[data$quality %in% c(5, 6), -c(12, 13)]
H <- hotelling.test(subject1, subject2)
H

PCA <- prcomp(data[,-c(12,13)])
pc1 <- PCA$x[data$red == 0, 1:2]
pc2 <- PCA$x[data$red == 1, 1:2]
H <- hotelling.test(pc1, pc2)
H

#H test with scaled data
```

```r
stddata <- data
stddata[, -c(12,13)] <- scale(stddata[, -c(12,13)])
stdSubject1 <- stddata[data$quality == 7, -c(12,13)]
stdSubject2 <- stddata[data$quality %in% c(5, 6), -c(12, 13)]

H <- hotelling.test(stdSubject1, stdSubject2)
H

stdPCA <- prcomp(stddata[, -c(12, 13)])

# Subset for quality == 7
stdPc1 <- stdPCA$x[data$quality == 7, 1:2]

# Subset for quality == 5 or 6
stdPc2 <- stdPCA$x[data$quality %in% c(5, 6), 1:2]  # Assuming you want the first two principal compor

# Perform Hotelling's T-squared test
H <- hotelling.test(stdPc1, stdPc2)
H

plot(PCA$sdev^2,
     type="b",
     xlab="Principal component",
     ylab="Eigenvalue",
     main="Scree plot,\n unscaled data")

plot(stdPCA$sdev^2,
     type="b",
     xlab="Principal component",
     ylab="Eigenvalue",
     main="Scree plot,\n standardized data")

#LDA
X <- YJ_transformed_wine[,7:8] ##using total.sulfur.dioxide and density here for lowest misclassificat
y <- as.numeric(data[13]=='0')

ldaFit <- lda(X, y)
ldaFit

prd <- predict(ldaFit, X)

#histogram
ldahist(data = prd$x, g = y)

#misclassification rate
tab <- table(Predicted = prd$class, Actual = y)
1-sum(diag(tab))/sum(tab)

## plotting
partimat(X, as.factor(y), method = "lda")

#using testing/training sets to evaluate classification error
set.seed(498022)
```

```r
ind <- sample(dim(X)[1], round(dim(X)[1]/2))
Xtraining <- X[ind,]
Xtesting <- X[-ind,]
ytraining <- y[ind]
ytesting <- y[-ind]

ldaFit1 <- lda(Xtraining, ytraining)
ldaFit1

#misclassification rate
prd <- predict(ldaFit1, Xtesting)
tab <- table(Predicted = prd$class, Actual = ytesting)
1-sum(diag(tab))/sum(tab)

#LDA
X1 <- YJ_transformed_wine[,7:8]
y1 <- as.numeric(data[12]=='7')

ldaFit2 <- lda(X1, y1)
ldaFit2

prd2 <- predict(ldaFit2, X1)

#histogram
ldahist(data = prd2$x, g = y1)

#misclassification rate
tab <- table(Predicted = prd2$class, Actual = y1)
1-sum(diag(tab))/sum(tab)

## plotting
partimat(X1, as.factor(y1), method = "lda")

#using testing/training sets to evaluate classification error
set.seed(498022)
ind1 <- sample(dim(X1)[1], round(dim(X1)[1]/2))
Xtraining1 <- X1[ind,]
Xtesting1 <- X1[-ind,]
ytraining1 <- y1[ind]
ytesting1 <- y1[-ind]

ldaFit3 <- lda(Xtraining1, ytraining1)
ldaFit3

#misclassification rate
prd3 <- predict(ldaFit3, Xtesting1)
tab <- table(Predicted = prd3$class, Actual = ytesting1)
1-sum(diag(tab))/sum(tab)
```