Wine Type Classification using Principle Component Analysis

Dae Hyeun Cheong
Hee Won Ahn

December 15, 2023

I Introduction

In this report, we delve into a dataset comprising six hundred distinct Portuguese wines, each characterized by thirteen different features. Employing meticulous introductory analysis and principal component analysis, our objective is to categorize the wines into red or white types. Furthermore, we aim to assess the efficacy of principal component analysis in addressing and resolving such classification problems.

II Exploratory Data Analysis

The wine.csv data consists of 13 different characteristics of six-hundred different wines.

Table 1: Data Dictionary

Wine Characteristics	***Unit***	***Type***
fixed.acidity		num (continuous)
volatile.acidity		num (continuous)
citric.acid	g/dm^3	num (continuous)
residual.sugar		num (continuous)
chlorides		num (continuous)
free.sulfur.dioxide	mg/dm^3	num (continuous)
total.sulfur.dioxide	mg/dm^3	num (continuous)
density	g/dm^3	num (continuous)
pH	N/A	num (continuous)
sulphates	g/dm^3	num (continuous)
alcohol	$vol.\%$	num (continuous)
quality	0 $(very\ bad) to$ 10 $(excellent)$	int (qualitative)
red	1 red / 0 white	int (qualitative)

We examined the dataset and found no missing values (refer to Code 1 in the appendix), eliminating the need for imputation. The distribution of wine quality and wine type across the dataset is even, with each quality representing 33% and each wine type representing 50% (refer to Figures 1 and 2 in the appendix). This balanced distribution of qualitative data suggests a systematic sampling approach from the population.

Inspecting the histograms of each continuous variable (refer to Figure 3 in the appendix), we observe that many of them are right skewed, except for density and pH, which appear to be approximately normal.

The correlation matrix (refer to Code 3 in the appendix) reveals notable correlations. Notably, free.sulfer.dioxide and total.sulfer.dioxide exhibit the highest correlation, which is intuitively expected. Additionally, there is a discernible correlation between fixed acidity/density and residual sugar/total sulfer.dioxide.

III Principal Component Analysis and Result

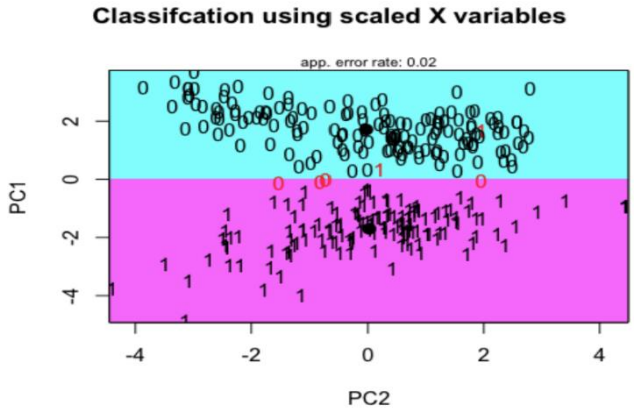
We conducted principal component analysis on the unscaled dataset and generated the scree plot (refer to Figure 4 in the appendix), revealing that the 1st principal component accounted for the majority of the variance (approximately 96% of the entire dataset, refer to Code 4 in the appendix). Using the first two unscaled principal components, we plotted wine quality and wine type. Surprisingly, while it wasn't effective in clustering wine quality, it proved useful in grouping wine types (refer to Figure 5 and 6 in the appendix).

Recognizing the efficacy of these unscaled principal components in classifying wine types, we applied one clustering and one classification technique—K-means and Linear Discriminant Analysis (LDA). Both methods performed well, with LDA achieving accuracy comparable to K-means, even though the assumption of multi-normality of predictor variables was not fully justified (refer to Figure 7, 8, and 9 in the appendix). The overall error rate for both methods, when applied to the entire dataset without splitting into train and test sets, was approximately 9% (refer to Code 5, 6, and 7 in the appendix).

Transitioning to scaled principal components yielded interesting findings:

1. The scree plot for scaled principal components showed that each component explained a fair amount of variance (31% for the 1st principal component, 20.7% for the 2nd principal component, etc., refer to Code 8 in the appendix).
2. Analyzing the loadings of unscaled and scaled principal components separately, we observed that for unscaled principal components, sulfur dioxide was the primary contributor to the 1st component explaining the most variance (refer to Figure 12 in the appendix). Conversely, the 1st principal component of scaled data displayed a more even distribution of loadings across predictor variables (refer to Figure 13 in the appendix).

Upon applying LDA to the first two scaled principal components, we achieved remarkably impressive results in classifying wine types, with a misclassification rate of only 2%.



IV Discussion

The analysis above yields several crucial insights into the dataset:

1. **Scaling for PCA:**

To effectively utilize PCA on the wine dataset, it is advisable to scale the data before applying PCA. This is particularly important because the units across predictors vary significantly, and the quantity of sulfur dioxide stands out as being significantly larger (about 1000 times) than the other predictors. The substantial magnitude of one predictor has the potential to distort the overall analysis of the data, emphasizing the importance of scaling for a more accurate representation.

2. **Choosing Principal Components:**

Despite the first two unscaled principal components explaining more variance than the first two scaled principal components, the latter proved more effective for our specific objective. This observation underscores the importance of carefully examining the overall composition of loadings for each principal component, rather than solely relying on the total variance explained by the components. Depending on the type of analysis and objective, the choice of principal components should be guided by a nuanced understanding of the loading patterns for optimal results.

V Conclusion

By leveraging PCA and LDA on the wine dataset, we achieved a remarkable success in classifying wine types among 600 different wines, achieving a mere 2% misclassification rate. This notable performance can be attributed to the careful scaling of the data and the strategic selection of principal components characterized by a well-balanced composition of loadings. The analytical approach adopted, emphasizing appropriate preprocessing and thoughtful consideration of component characteristics, played a pivotal role in the successful classification of wine types.

Appendix

Figure 1

```
pie(table(data$quality),labels=lab,col=c('blue','purple','green'),
    main='Distribution of wine quality')
```

Figure 2

```
pie(table(data$red),
    labels = lab,
    col = lbls,
    main = 'Distribution of wine type',
    cex.main = 1.2,  # Increase the main title font size
    cex = 0.8,      # Adjust label font size
    radius = 0.8     # Adjust the size of the pie chart
)
```

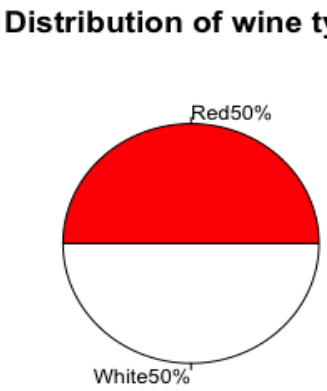


Figure 3

```
par(mfrow = c(2, 6))
for (i in 1:11) {
  hist(data[, i],
       main = paste(names(data)[i]),
       xlab = paste('Wine', names(data)[i]),
       col = 'skyblue',  # Change color to sky blue
       border = 'black', # Add a black border
       breaks = 20        # Adjust the number of bins
  )
}
```

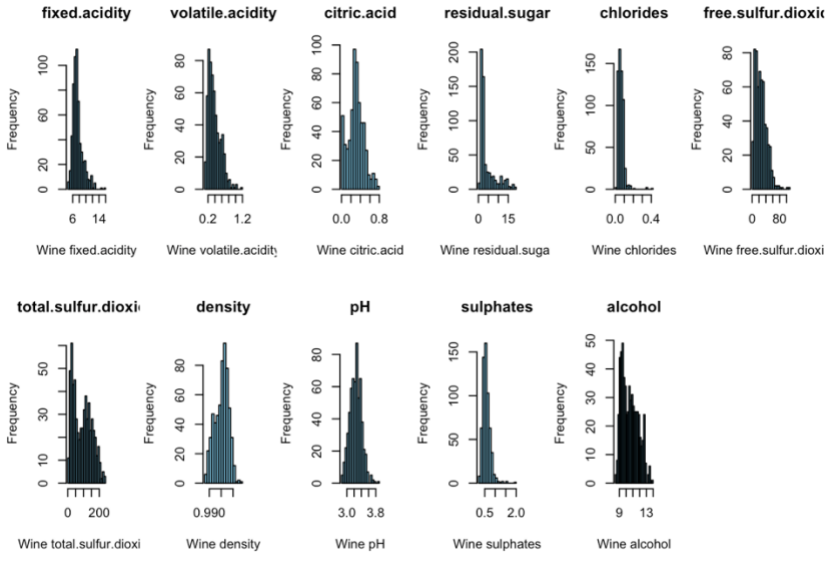


Figure 4

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

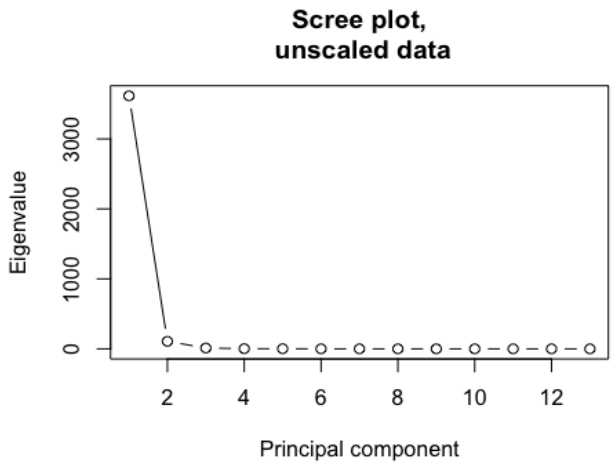


Figure 5

```r
plot(PCA$x[,1],PCA$x[,2],col=as.factor(data$quality),xlab="PC1",ylab="PC2",main='Wine quality on PCA plot')
```

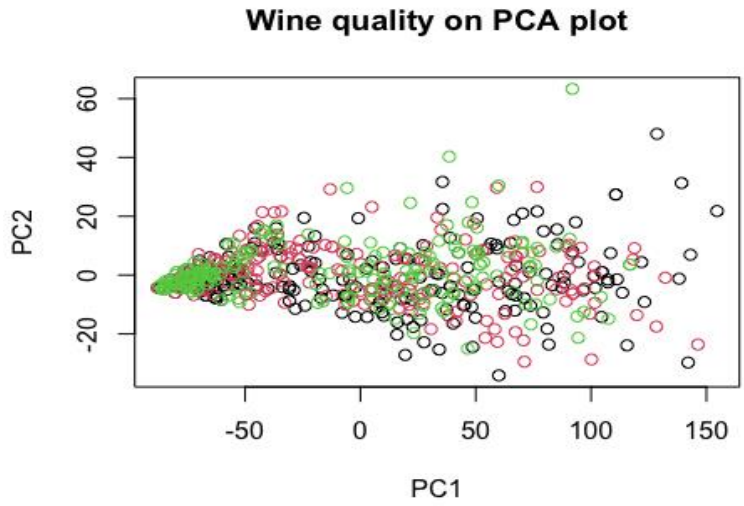


Figure 6

```r
plot(PCA$x[,1],PCA$x[,2],col=as.factor(data$red),xlab="PC1",ylab="PC2",main='Wine type on PCA plot')
```

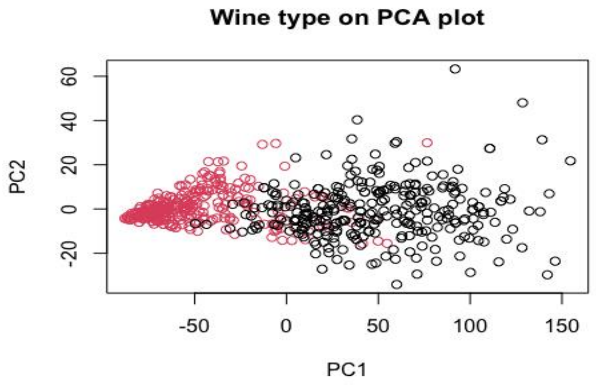


Figure 7

```r
cls=kmeans(PCA$x[,1:2],2)
plot(PCA$x[,1:2],col=cls$cluster,main='Clustering on PC1 and PC2')
```

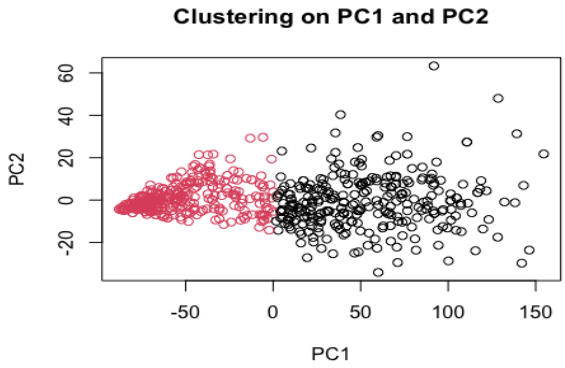


Figure 8

```r
par(mfrow=c(1,2))
for (i in 1:2){
  qqnorm(PCA$x[,i])
}
```

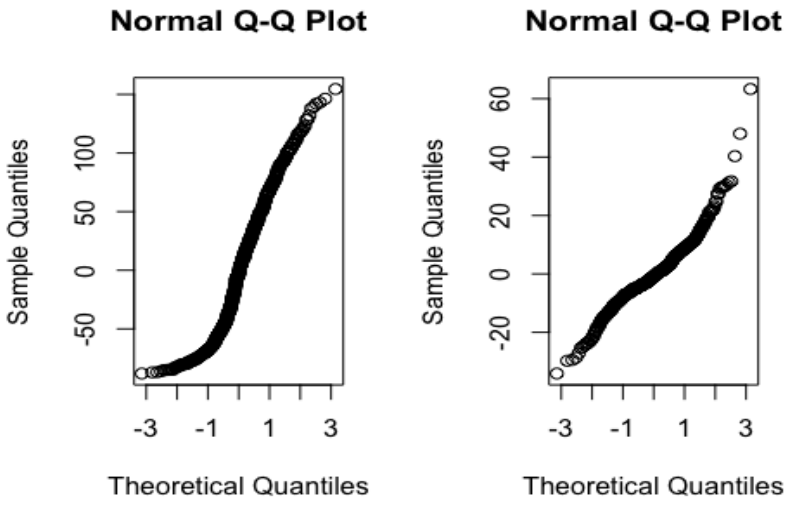


Figure 9

```
partimat(X, as.factor(y), method = "lda",main='Clssification using unscaled X variables')
```

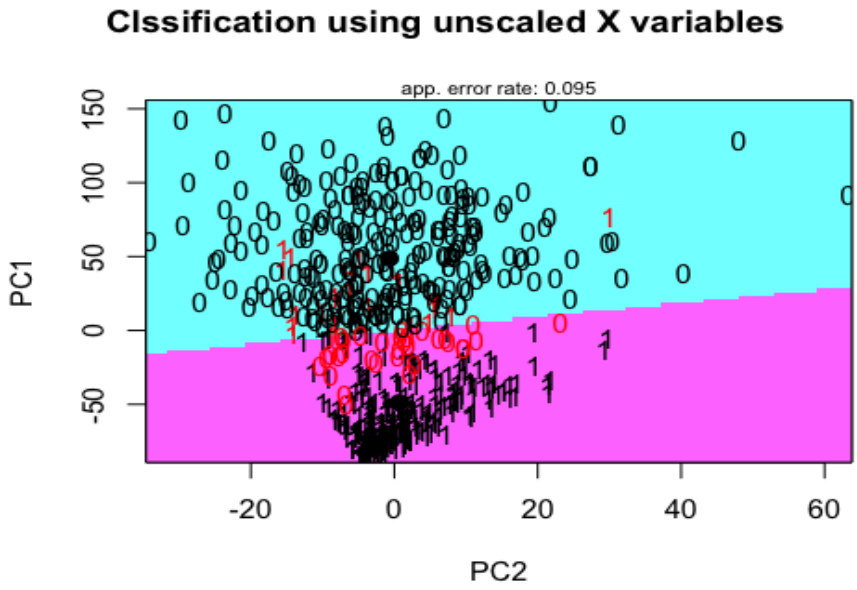


Figure 10

```
plot(PCA$sdev^2,
     type="b",
     xlab="Principal component",
```

```
     ylab="Eigenvalue",
     main="unscaled data")
```

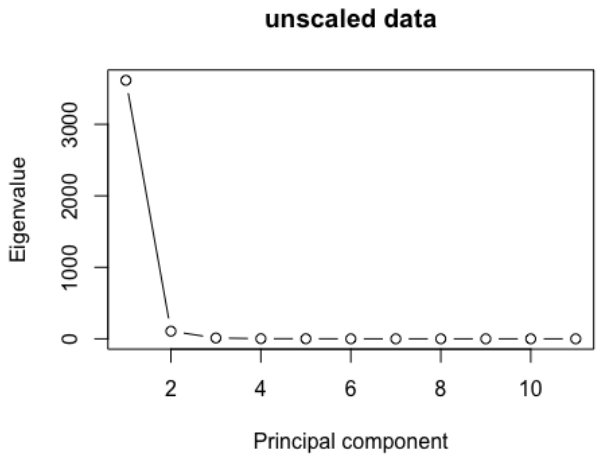


Figure 11

```
plot(stdPCA$sdev^2,
     type="b",
     xlab="Principal component",
     ylab="Eigenvalue",
     main="scaled data")
```

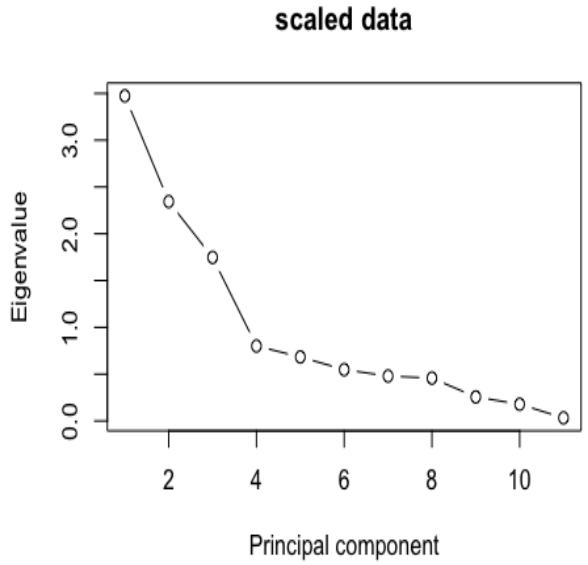


Figure 12

```
barplot(e1/PCA$sdev[1], main="Loadings for the 1st PC,\n unscaled data",cex.names = 0.8,las=2)
```

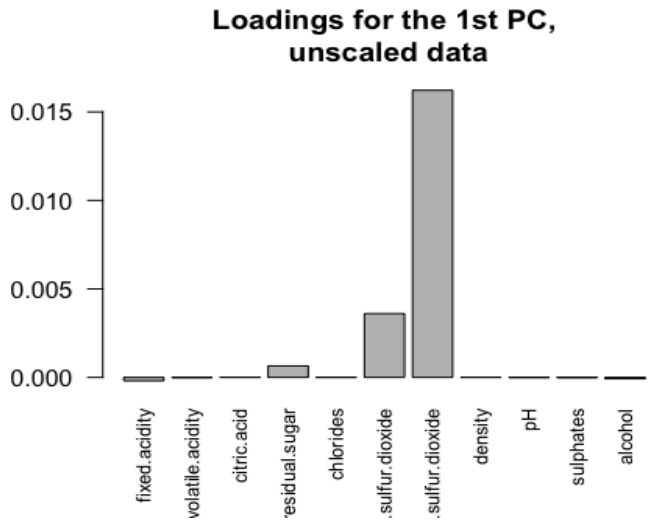


Figure 13

```
barplot(re1/stdPCA$sdev[1], main="Loadings for the 1st PC, \n rescaled data",cex.names = 0.8,
las=2)
```

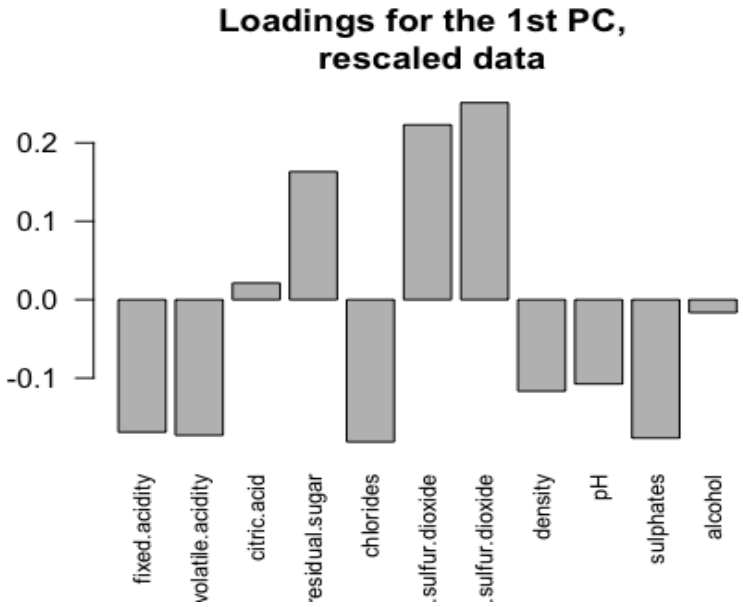
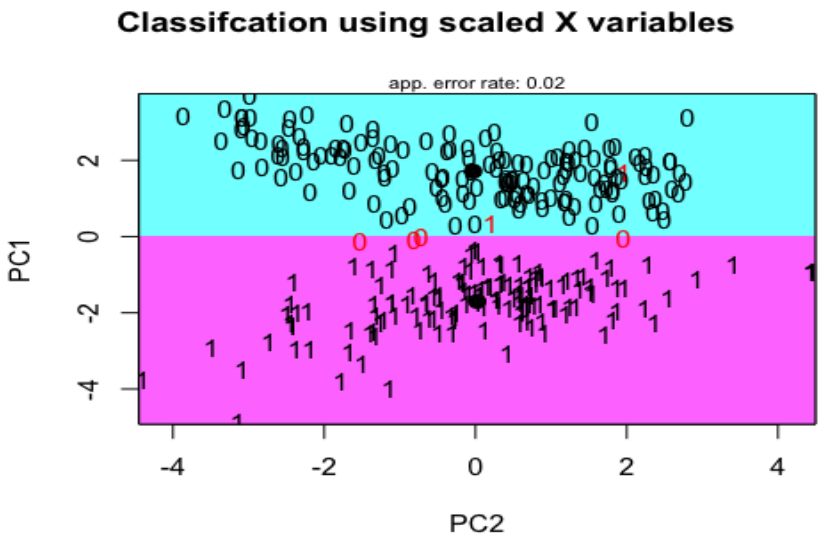


Figure 14

```
partimat(scale_Xtesting, as.factor(ytesting), method = "lda",main='Classifcation using scaled
X variables')
```



Code 1

```
str(data)
```

```
## 'data.frame':    600 obs. of  13 variables:
##  $ fixed.acidity       : num  8.1 9.6 7.7 7.1 8.3 8.8 6.2 7 7.7 6.6 ...
##  $ volatile.acidity    : num  0.67 0.68 1 0.34 0.65 ...
##  $ citric.acid         : num  0.55 0.24 0.15 0.28 0.1 0.26 0.08 0 0.26 0.03 ...
##  $ residual.sugar      : num  1.8 2.2 2.1 2 2.9 1.6 2 1.7 1.9 7.8 ...
##  $ chlorides           : num  0.117 0.087 0.102 0.082 0.089 0.088 0.09 0.052 0.062 0.079 ..
.
##  $ free.sulfur.dioxide : num  32 5 11 31 17 16 32 3 9 6 ...
##  $ total.sulfur.dioxide: num  141 28 32 68 40 23 44 8 31 12 ...
##  $ density             : num  0.997 0.999 0.996 0.997 0.998 ...
##  $ pH                  : num  3.17 3.14 3.23 3.45 3.29 3.32 3.45 3.41 3.39 3.52 ...
##  $ sulphates           : num  0.62 0.6 0.48 0.48 0.55 0.47 0.58 0.47 0.64 0.5 ...
##  $ alcohol             : num  9.4 10.2 10 9.4 9.5 9.4 10.5 10.3 9.6 12.2 ...
##  $ quality             : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ red                 : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
sum(is.na(data)) # there is no missing value.
```

```
## [1] 0
```

```
n = nrow(data) #there are total 600 data
```

Code 2

```
lbls <- c('quality 5','quality 6','quality 7')
pct <- round(100*table(data$quality)/n)
lab <- paste(lbls,pct)
lab <- paste(lab,'%',sep='')
pie(table(data$quality),labels=lab,col=c('blue','purple','green'),
    main='Distribution of wine quality')
```

```r
# distribution of quality of wines are equal with frequency of 33% each
lbls <- c('Red', 'White')
pct <- round(100 * table(data$red) / nrow(data))
lab <- paste(lbls, pct, '%', sep='')
pie(table(data$red),
    labels = lab,
    col = lbls,
    main = 'Distribution of wine type',
    cex.main = 1.2,  # Increase the main title font size
    cex = 0.8,      # Adjust label font size
    radius = 0.8     # Adjust the size of the pie chart
)
```

Code 3

```r
# distribution of type of wines are equal with frequency of 50% each

par(mfrow = c(2, 6))
for (i in 1:11) {
  hist(data[, i],
       main = paste(names(data)[i]),
       xlab = paste('Wine', names(data)[i]),
       col = 'skyblue',  # Change color to sky blue
       border = 'black', # Add a black border
       breaks = 20        # Adjust the number of bins
  )
}

# it seems from histogram that that fixed.acidity, volatitle.acidity, ciric.acid, residual.sug
ar, chlorides, free.sulfur.dioxide, total.sulfur.dioxide,sulphates, alcohol are skewed to the
right
# it seems from histogram that PH and density  normally distributed

cor(data[1:10])
```

```
##                       fixed.acidity volatile.acidity citric.acid residual.sugar
## fixed.acidity             1.0000000        0.1545257  0.42840905     -0.1728437
## volatile.acidity          0.1545257        1.0000000 -0.45403620     -0.2517687
## citric.acid               0.4284090       -0.4540362  1.00000000      0.1211949
## residual.sugar           -0.1728437       -0.2517687  0.12119486      1.0000000
## chlorides                 0.3476264        0.3747433  0.07990265     -0.2013515
## free.sulfur.dioxide      -0.3982224       -0.3436413  0.05618164      0.4528818
## total.sulfur.dioxide     -0.4403786       -0.4189662  0.10117308      0.5642441
## density                   0.5761503        0.3111981  0.11527440      0.3598633
## pH                       -0.2363301        0.3522679 -0.39165557     -0.3395116
## sulphates                 0.3894683        0.1610080  0.13783211     -0.2788560
##                        chlorides free.sulfur.dioxide total.sulfur.dioxide
## fixed.acidity         0.34762644         -0.39822236           -0.4403786
## volatile.acidity      0.37474328         -0.34364135           -0.4189662
## citric.acid           0.07990265          0.05618164            0.1011731
## residual.sugar       -0.20135147          0.45288180            0.5642441
## chlorides             1.00000000         -0.30365440           -0.3996661
## free.sulfur.dioxide  -0.30365440          1.00000000            0.7664993
## total.sulfur.dioxide -0.39966614          0.76649931            1.0000000
## density               0.42097585         -0.15757001           -0.1890715
## pH                    0.06960826         -0.23922718           -0.3362251
## sulphates             0.41369560         -0.31306455           -0.4242543
##                          density          pH  sulphates
## fixed.acidity         0.57615026 -0.23633014  0.3894683
## volatile.acidity      0.31119807  0.35226788  0.1610080
## citric.acid           0.11527440 -0.39165557  0.1378321
```

```
## residual.sugar        0.35986329 -0.33951156 -0.2788560
## chlorides             0.42097585  0.06960826  0.4136956
## free.sulfur.dioxide  -0.15757001 -0.23922718 -0.3130646
## total.sulfur.dioxide -0.18907149 -0.33622513 -0.4242543
## density               1.00000000  0.01731977  0.2712817
## pH                    0.01731977  1.00000000  0.1493122
## sulphates             0.27128171  0.14931225  1.0000000
```

```
#correlation between free.sulfur.dioxide and total.sulfur.dioxide is 0.77 which shows high lin
earity.
#correlation between fixed.acidity and density is 0.58 which shows some linearity.
#correlation between residual.sugar and total.sulfur.dixoide is 0.58 which shows some linearit
y.

# correlation between other variables are between -0.5 and 0.5 which mean there are not much l
inearity.
```

Code 4

```
par(mfrow = c(1,1))

PCA=prcomp(data)

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

cumsum(PCA$sdev^2/sum(PCA$sdev^2)) # first principal component explain 96% of total variance
```

```
##  [1] 0.9671576 0.9957596 0.9989170 0.9994657 0.9998646 0.9999588 0.9999818
##  [8] 0.9999890 0.9999935 0.9999971 0.9999998 1.0000000 1.0000000
```

```
plot(PCA$x[,1],PCA$x[,2],col=as.factor(data$quality),xlab="PC1",ylab="PC2",main='Wine quality
on PCA plot') #first principal component and second principal component cannot explan quality
of wines pretty well

plot(PCA$x[,1],PCA$x[,2],col=as.factor(data$red),xlab="PC1",ylab="PC2",main='Wine type on PCA
plot') #plot of first principal component and second principal compent  an explan quality of w
ines pretty well
```

Code 5

```
cls=kmeans(PCA$x[,1:2],2)
plot(PCA$x[,1:2],col=cls$cluster,main='Clustering on PC1 and PC2')

tab_clus = table(Predicted = cls$cluster, Actual = data$red)
1-sum(diag(tab_clus))/sum(tab_clus)  #classifications rate=9%
```

```
## [1] 0.09
```

```
# even though clustering is not classification method, we can distinguish type of wine by usin
g K-means clustering because data points of red and white wine are close to each other when pl
otted by two principal components.
# using PCA, we may be able to classify type of wine.
```

Code 6

```r
par(mfrow=c(1,2))
for (i in 1:2){
  qqnorm(PCA$x[,i])
}
```

Code 7

```r
#  two principal components does not perfectly follow normal distribution,but we may try to fi
sher's lda classifciaton model.

X = PCA$x[,c(1,2)]
y = data$red
ind <- sample(dim(X)[1], round(dim(X)[1]/2))
Xtraining <- X[ind,]
Xtesting <- X[-ind,]
ytraining <- y[ind]
ytesting <- y[-ind]
ldaFit <- lda(Xtraining, ytraining)
prd <- predict(ldaFit, Xtesting)
tab <- table(Predicted = prd$class, Actual = ytesting)
1-sum(diag(tab))/sum(tab) # classifications rate 8%
```

```
## [1] 0.09666667
```

```r
partimat(X, as.factor(y), method = "lda",main='Clssification using unscaled X variables')



# even though principle of components does not folloow normal distriubtion perfectly, LDA can
still
# classify type of wines using two pricniple of componetns significantly
```

Code 8

```r
PCA= prcomp(data[,1:11])
stdPCA <- prcomp(scale(data[,1:11]))

## scree plots for unscaled vs. standardized
plot(PCA$sdev^2,
     type="b",
     xlab="Principal component",
     ylab="Eigenvalue",
     main="unscaled data")

# cumulative proportion of variance explained

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

cumsum(stdPCA$sdev^2/sum(stdPCA$sdev^2))
```

```
##  [1] 0.3157960 0.5288978 0.6877531 0.7605328 0.8226450 0.8724644 0.9160372
##  [8] 0.9576259 0.9809063 0.9970823 1.0000000
```

```r
# we need more than two principal of components to explain scaled data
```

```r
e1 = PCA$rotation[,1]
re1 = stdPCA$rotation[,1]
par(mfrow=c(1,1))

barplot(e1/PCA$sdev[1], main="Loadings for the 1st PC,\n unscaled data",cex.names = 0.8,las=2)

# for unsealed data, total.sulfur.dioxide plays significant role
barplot(re1/stdPCA$sdev[1], main="Loadings for the 1st PC, \n re`scaled data",cex.names = 0.8,
 las=2)
```

Code 9

```r
# for scaled data, all the variables plays similar role
scale_X = stdPCA$x[,c(1,2)]
y = data$red
ind <- sample(dim(scale_X)[1], round(dim(scale_X)[1]/2))
scale_Xtraining =  scale_X[ind,]
scale_Xtesting = scale_X[-ind,]
ytraining = y[ind]
ytesting = y[-ind]
scale_ldaFit = lda(scale_Xtraining, ytraining)
scale_prd = predict(scale_ldaFit, scale_Xtesting)
scale_tab = table(Predicted = scale_prd$class, Actual = ytesting)
1-sum(diag(scale_tab))/sum(scale_tab) # classifications rate 10%

## [1] 0.02

partimat(scale_Xtesting, as.factor(ytesting), method = "lda",main='Classifcation using scaled
X variables')

# scaling data does improve performance of classification model.
```