ST 557
Final Project
Elena Volpi
12/11/2021

Goal 1: Distinguish white wine from red wine

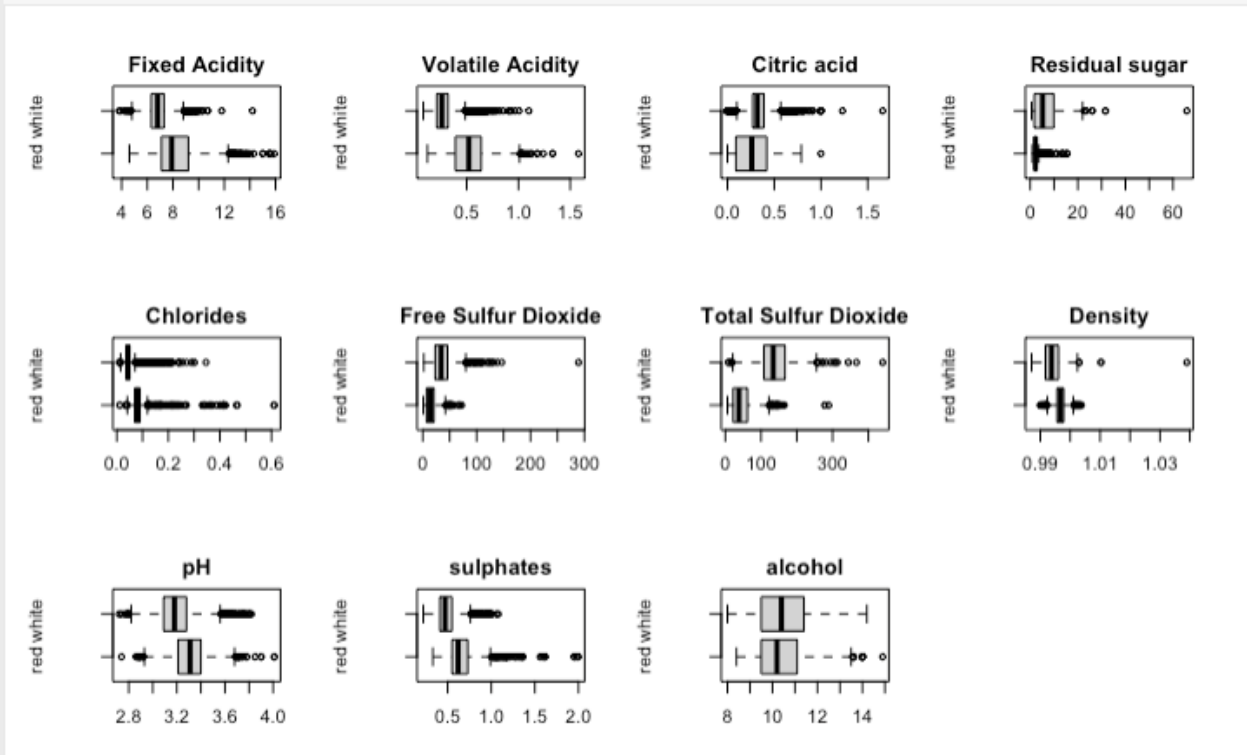
a)

To determine if there was a difference in mean vectors of the attributes between red and white wine, I implemented Hotelling's two sample test with hypotheses:

$H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = \mu_7 = \mu_8 = \mu_9 = \mu_{10} = \mu_{11} = 0$

$H_A: \mu_i \neq \mu_j \; for \; some \; i, j.$

With 95% confidence, we reject the null hypothesis that all mean vectors of the 11 chemical attributes of white wine and red wine are equal with a p-value very near zero (p-value < 2.2e-16). Therefore, at least one chemical attribute differs between red wine and white wine.



The most notable differences in averages of attributes are with free and total sulfur dioxide, red wine has lower free sulfur dioxide by 19.43 and lower total sulfur dioxide by 91.89 than white wine. Red wine has higher fixed acidity then white wine by 1.465 (units) and lower residual sugar by 3.85. We do have some notable outliers in white wine for density, residual sugar, citric acid, and free sulfur dioxide, as well as others. Our datasets are not equal in size, we have approximate 3:1 ratio of white wine to red wine. Standardization would help to do further exploratory analysis of the dataset.

b)

I decided to implement linear discriminant analysis because each attribute is approximately normal and LDA would handle the imbalance of having significantly more white wine observations then red wine. I also appreciated that LDA is computationally easy and standardization doesn't have a benefit. Although, the covariance matrixes are not exactly equal.

To implement LDA, I first added a label of '1' for white wine, and '2' for red wine and then created a data frame will all observations of white wine and red wine. After implementing LDA, our confusion matrix and coefficients are as follows:

Predicted	Actual: White	Actual: Red
Predicted: white	4882	16
Predicted: red	19	1580

```
Coefficients of linear discriminants:
                              LD1
`fixed acidity`         -0.32334347
`volatile acidity`       3.06028749
`citric acid`           -0.87495758
`residual sugar`        -0.35118041
chlorides                5.08835703
`free sulfur dioxide`    0.01921302
`total sulfur dioxide`  -0.02006459
density                910.43476356
pH                      -1.10607299
sulphates                0.86513227
alcohol                  0.82519241
```

We see that a wine was predicted as white wine and classified as red wine 16 times. Similarly, a white was inaccurately classified as red wine 19 times. Our error rate is 0.005387102, meaning our classification model has prediction accuracy of ~99.47%. The probability that we would correctly classify a red wine if drawn from the same population of red wines is 98.99%.

A new observation, X_0 will be classified as belonging to white wine if :

$$a^T X_0 > \frac{a^T \bar{X}_{white} + a^T \bar{X}_{red}}{2}, \text{ where}$$

$$a^T = [1.87, -17.72, 5.07, 2.03, -29.47, -0.11, 0.12, -5272.45, 6.41, -5.10, -4.78]^T$$

c)

I have chosen to use k-means because of it prioritizing variance minimization (using Euclidean distance) and using the mean values in its algorithm. Using the data frame that has all wines, I first use scale() to standardize. Our confusion matrix is as follows:

Predicted	Actual: White	Actual: Red
Predicted: white	4830	68
Predicted: red	24	1575

Our error rate is 0.01416038, meaning we classify with 98.58396% accuracy.

Goal 2:Better understand which of these variables is most important to wine quality. (Focus on red wine)

a)

Using Wilk's test/One way MANOVA, we reject the null hypothesis that the mean vectors for all quality scores would be the same in favor of the alternative hypothesis that at least one mean vector is different at a 95% confidence level with p-value < 2.2e-16.

If we group quality by low, medium, and high and use one way MANOVA, we come to the same conclusion that there is evidence that at least one mean vector for quality grouping is the not the same with a p-value near zero (p-value< 2.2e-16).

b)

I chose to use knn as my classification. Using the standardized red wine data, I created a training set of size 1000 and testing set with the remaining observations. First, I wanted to see how well it would predict quality with k=3. Our error rate for predicting quality is 0.1419032, we see from our confusion matrix that most errors occurred when the wines were close in quality, particularly those of medium quality.

	level_knn.3		
	1	2	3
1	11	3	0
2	0	505	6
3	0	13	61

Confusion matrix for grouped quality

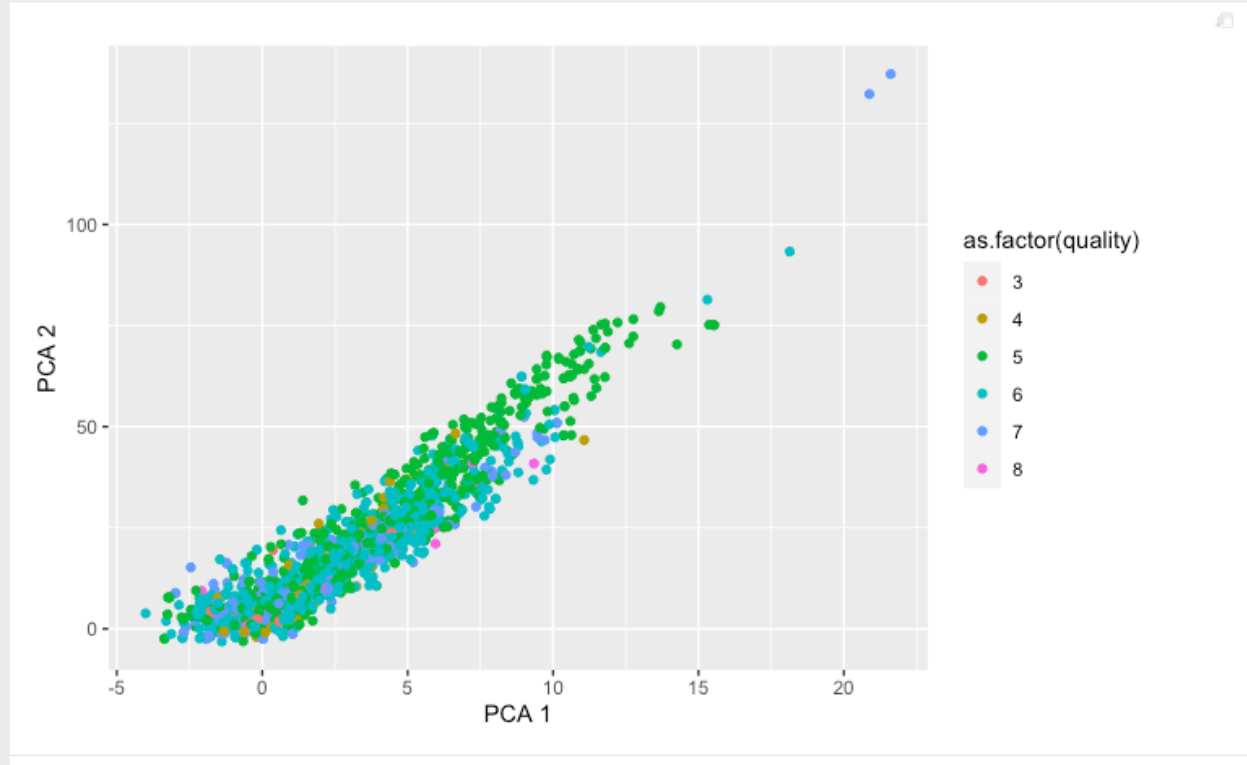
	quality_knn.3					
	3	4	5	6	7	8
3	1	0	0	1	0	0
4	0	10	2	0	0	0
5	0	0	234	36	0	0
6	0	0	19	215	7	0
7	0	0	0	13	54	3
8	0	0	0	0	4	0

Confusion matrix for non-grouped quality

Using knn with k=3 to predict grouped quality (Low =1, Medium=2, High=3) was more successful with an error rate of 0.03672788.

c)

I did not standardize first since I used my correlation matrix for PCA and there was no need. The first two components account for 69.81% of cumulative variance.



Using the first two PCA components, I used knn for quality prediction. The prediction error rate was 0.008347245 or predicting with 99% accuracy. This was a better prediction rate than of kmeans and knn implemented above.

```
  pca_knn.3
     3   4   5   6   7   8
3    2   1   0   0   0   0
4    0  17   0   0   0   0
5    0   0 236   0   0   0
6    0   0   0 258   0   0
7    0   0   0   0  75   0
8    0   0   0   0   4   6
```

Final Project: ST 557

Elena Volpi

12/8/2021

##Goal 1: Distinguish white wine from red wine

```
red_wine <- read_csv("winequality-red.csv")
```

```
## Rows: 1599 Columns: 12

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (12): fixed acidity, volatile acidity, citric acid, residual sugar, chlo...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
white_wine <- read_csv("winequality-white.csv")
```

```
## Rows: 4898 Columns: 12

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (12): fixed acidity, volatile acidity, citric acid, residual sugar, chlo...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Is there a difference in mean vectors between red and white wines for these 11 chemical attributes?

```
##
##  Two-sample Hotelling test
##
## data:  white_wine[, -12] and red_wine[, -12]
## T2 = 40427.9, F = 3669.6, df1 = 11, df2 = 6485, p-value < 2.2e-16
## alternative hypothesis: true difference in mean vectors is not equal to (0,0,0,0,0,0,0,0,0,0,0)
## sample estimates:
##               fixed acidity volatile acidity citric acid residual sugar
## mean x-vector      6.854788        0.2782411   0.3341915       6.391415
```

```
## mean y-vector      8.319637        0.5278205   0.2709756       2.538806
##                 chlorides free sulfur dioxide total sulfur dioxide   density
## mean x-vector 0.04577236            35.30808            138.36066 0.9940274
## mean y-vector 0.08746654            15.87492             46.46779 0.9967467
##                       pH sulphates  alcohol
## mean x-vector 3.188267 0.4898469 10.51427
## mean y-vector 3.311113 0.6581488 10.42298
```

##Which attributes seem to differ most between red and white wine?

```
#compute the covariance matrices without quality parameter
#(cov_white <- cov(white_wine[,1:11]))
#(cov_red <- cov(red_wine[,1:11]))  #Maybe not need this

standardize <- function(mean,sd,x){
  standard_wine <- x
  for(i in 1:nrow(x)) {
    standard_wine[i,] <- (x[i,] - mean)/sd
  }
  return(standard_wine)
}

#Standardize data
white_sd <- map_dbl(white_wine[,-12],sd)
red_sd <- map_dbl(red_wine[,-12],sd)
Mean_White <- apply(white_wine[,-12],2,mean)
Mean_Red <- apply(red_wine[,-12],2,mean)

#Also add label for type to distinguish when combinging data frames.
white_standardized <- standardize(Mean_White,white_sd,white_wine[,-12]) %>% mutate(wine_type = rep(1,nr

red_standardized <- standardize(Mean_Red,red_sd,red_wine[,-12]) %>% mutate(wine_type = rep(2,nrow(red_w

st_all_wine <- rbind(white_standardized, red_standardized)

st_white_cov <- cov(white_standardized[,-12])
st_red_cov <- cov(red_standardized[,-12])
st_dif_cov <- st_white_cov-st_red_cov
st_dif_cov
```

```
##                      fixed acidity volatile acidity   citric acid
## fixed acidity           0.00000000      0.233433605 -3.825227e-01
## volatile acidity        0.23343360      0.000000000  4.030239e-01
## citric acid            -0.38252274      0.403023874 -3.330669e-16
## residual sugar         -0.02575602      0.062368178 -4.936554e-02
## chlorides              -0.07061954      0.009213799 -8.945847e-02
## free sulfur dioxide     0.10439833     -0.086508112  1.550554e-01
## total sulfur dioxide    0.20425120      0.012790499  8.559777e-02
## density                -0.40271628      0.005087613 -2.154446e-01
## pH                      0.25711990     -0.266852663  3.781559e-01
## sulphates              -0.20014865      0.225258538 -2.504391e-01
## alcohol                -0.05921285      0.270005970 -1.856320e-01
##                      residual sugar    chlorides free sulfur dioxide
```

```
## fixed acidity           -0.02575602 -0.070619543        1.043983e-01
## volatile acidity         0.06236818  0.009213799       -8.650811e-02
## citric acid             -0.04936554 -0.089458465        1.550554e-01
## residual sugar           0.00000000  0.033075001        1.120494e-01
## chlorides                0.03307500  0.000000000        9.583021e-02
## free sulfur dioxide      0.11204936  0.095830205       -2.220446e-16
## total sulfur dioxide     0.19841143  0.151509831       -5.216549e-02
## density                  0.48368308  0.056578994        3.161562e-01
## pH                      -0.10848103  0.174586675       -7.099529e-02
## sulphates               -0.03219149 -0.354497598        7.559674e-03
## alcohol                 -0.49270666 -0.139048167       -1.806956e-01
##                      total sulfur dioxide      density            pH
## fixed acidity               2.042512e-01 -0.402716278  2.571199e-01
## volatile acidity            1.279050e-02  0.005087613 -2.668527e-01
## citric acid                 8.559777e-02 -0.215444605  3.781559e-01
## residual sugar              1.984114e-01  0.483683084 -1.084810e-01
## chlorides                   1.515098e-01  0.056578994  1.745867e-01
## free sulfur dioxide        -5.216549e-02  0.316156242 -7.099529e-02
## total sulfur dioxide       -4.440892e-16  0.458611848  6.881553e-02
## density                     4.586118e-01  0.000000000  2.481078e-01
## pH                          6.881553e-02  0.248107841 -2.220446e-16
## sulphates                   9.161553e-02 -0.074013263  3.525991e-01
## alcohol                    -2.432382e-01 -0.283957851 -8.420041e-02
##                         sulphates       alcohol
## fixed acidity       -2.001486e-01 -5.921285e-02
## volatile acidity     2.252585e-01  2.700060e-01
## citric acid         -2.504391e-01 -1.856320e-01
## residual sugar      -3.219149e-02 -4.927067e-01
## chlorides           -3.544976e-01 -1.390482e-01
## free sulfur dioxide  7.559674e-03 -1.806956e-01
## total sulfur dioxide 9.161553e-02 -2.432382e-01
## density             -7.401326e-02 -2.839579e-01
## pH                   3.525991e-01 -8.420041e-02
## sulphates            1.110223e-16 -1.110275e-01
## alcohol             -1.110275e-01  1.110223e-16
```

```
un_white_cov <- cov(white_wine[,-12])
un_red_cov <- cov(red_wine[,-12])
diff_cov <- un_white_cov- un_red_cov
```

```
#LDA

#First put together in one dataset, with labels for 1 for white wine and 2 for red wine.
white <- white_wine[,-12] %>% mutate(wine_type = rep(1, nrow(white_wine)))
red <- red_wine[,-12] %>% mutate(wine_type = rep(2, nrow(red_wine)))
all_wine <- rbind(white, red)
all_wine_sc <- cbind(scale(all_wine[,-12]),all_wine[,12])


#No scaling
wine_lda <-  lda(wine_type~., data=all_wine)
wine_lda
```

```
## Call:
```

```
## lda(wine_type ~ ., data = all_wine)
##
## Prior probabilities of groups:
##         1         2
## 0.7538864 0.2461136
##
## Group means:
##   `fixed acidity` `volatile acidity` `citric acid` `residual sugar`  chlorides
## 1        6.854788          0.2782411     0.3341915         6.391415 0.04577236
## 2        8.319637          0.5278205     0.2709756         2.538806 0.08746654
##   `free sulfur dioxide` `total sulfur dioxide`   density       pH sulphates
## 1              35.30808              138.36066 0.9940274 3.188267 0.4898469
## 2              15.87492               46.46779 0.9967467 3.311113 0.6581488
##    alcohol
## 1 10.51427
## 2 10.42298
##
## Coefficients of linear discriminants:
##                                 LD1
## `fixed acidity`         -0.32334347
## `volatile acidity`       3.06028749
## `citric acid`           -0.87495758
## `residual sugar`        -0.35118041
## chlorides                5.08835703
## `free sulfur dioxide`    0.01921302
## `total sulfur dioxide`  -0.02006459
## density                910.43476356
## pH                      -1.10607299
## sulphates                0.86513227
## alcohol                  0.82519241
```

```
Predicted <- predict(wine_lda, newdata=all_wine[,-12])
actual <- as.factor(all_wine$wine_type)
confusionMatrix(actual, Predicted$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    2
##          1 4882   16
##          2   19 1580
##
##                Accuracy : 0.9946
##                  95% CI : (0.9925, 0.9962)
##     No Information Rate : 0.7543
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9855
##
##  Mcnemar's Test P-Value : 0.7353
##
##             Sensitivity : 0.9961
##             Specificity : 0.9900
##          Pos Pred Value : 0.9967
```

```
##          Neg Pred Value : 0.9881
##              Prevalence : 0.7543
##          Detection Rate : 0.7514
##    Detection Prevalence : 0.7539
##       Balanced Accuracy : 0.9930
##
##        'Positive' Class : 1
##
```

```
error1 <- (19+16)/nrow(all_wine)
error1
```

```
## [1] 0.005387102
```

```
#scaling
wine_lda_2 <-  lda(wine_type~., data=all_wine_sc)
wine_lda
```

```
## Call:
## lda(wine_type ~ ., data = all_wine)
##
## Prior probabilities of groups:
##         1         2
## 0.7538864 0.2461136
##
## Group means:
##   `fixed acidity` `volatile acidity` `citric acid` `residual sugar`  chlorides
## 1        6.854788          0.2782411     0.3341915         6.391415 0.04577236
## 2        8.319637          0.5278205     0.2709756         2.538806 0.08746654
##   `free sulfur dioxide` `total sulfur dioxide`   density       pH sulphates
## 1              35.30808              138.36066 0.9940274 3.188267 0.4898469
## 2              15.87492               46.46779 0.9967467 3.311113 0.6581488
##    alcohol
## 1 10.51427
## 2 10.42298
##
## Coefficients of linear discriminants:
##                                  LD1
## `fixed acidity`         -0.32334347
## `volatile acidity`       3.06028749
## `citric acid`           -0.87495758
## `residual sugar`        -0.35118041
## chlorides                5.08835703
## `free sulfur dioxide`    0.01921302
## `total sulfur dioxide`  -0.02006459
## density                910.43476356
## pH                      -1.10607299
## sulphates                0.86513227
## alcohol                  0.82519241
```

```
Predicted <- predict(wine_lda_2, newdata=all_wine_sc[,-12])
actual <- as.factor(all_wine_sc$wine_type)
confusionMatrix(actual, Predicted$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    2
##          1 4882   16
##          2   19 1580
##
##                Accuracy : 0.9946
##                  95% CI : (0.9925, 0.9962)
##     No Information Rate : 0.7543
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9855
##
##  Mcnemar's Test P-Value : 0.7353
##
##             Sensitivity : 0.9961
##             Specificity : 0.9900
##          Pos Pred Value : 0.9967
##          Neg Pred Value : 0.9881
##              Prevalence : 0.7543
##          Detection Rate : 0.7514
##    Detection Prevalence : 0.7539
##       Balanced Accuracy : 0.9930
##
##        'Positive' Class : 1
##
```

```
error2 <- (19+16)/nrow(all_wine_sc)
error2
```

```
## [1] 0.005387102
```

```
n1 <- nrow(white_wine)
n2 <- nrow(red_wine)

all_wine_sp <- ((n1-1)*un_white_cov + (n2-1)*un_red_cov)/(n1+n2-2)
all_wine_s <- cov(all_wine[,-12])

Wmat <- (n1+n2-2)*all_wine_sp
Tmat <- (nrow(all_wine)-1)*all_wine_s
Bmat <- Tmat - Wmat

WinvB <- solve(Wmat) %*% Bmat
WinvB.eigen <- eigen(WinvB)

a_T <- t(Mean_White - Mean_Red)%*% solve(all_wine_sp)
#rule <- (a_T %*% as.matrix(Mean_White) + a_T %*% as.matrix(Mean_Red))/2

a_T
```

```
##      fixed acidity volatile acidity citric acid residual sugar chlorides
## [1,]      1.872525        -17.72253    5.066995       2.033732 -29.46735
```

```
##      free sulfur dioxide total sulfur dioxide   density       pH sulphates
## [1,]          -0.1112651            0.1161967 -5272.448 6.405415 -5.010095
##        alcohol
## [1,] -4.778798
```

c) Clustering: Cluster these wines into two clusters using the 11 chemical attributes.Think carefully about the distance measure and the clustering method you chose, and justify your choices in your report. If you split these data into two clusters, how well do these clusters reflect the red/white classification?

```
#kmeans
wine.km2 <- kmeans(scale(all_wine[,-12]),center=2)
actual <- as.factor(all_wine[,12])

cluster_df <- cbind(all_wine[,12], wine.km2$cluster)

table(cluster_df$wine_type, wine.km2$cluster)
```

```
##
##        1    2
##   1 2208 2690
##   2   18 1581
```

```
error <- (24+68)/nrow(cluster_df)
error
```

```
## [1] 0.01416038
```

##Goal 2:Better understand which of these variables is most important to wine quality. (Focus on red wine)

a) Is there a difference in mean vectors between wines with different quality scores?

```
#compute the covariance matrices without quality parameter
cov_white <- cov(white_wine[,1:11])
cov_red <- cov(red_wine[,1:11])
#covaraince matrices are not equal

Wilks.test(red_wine[,1:11],grouping=as.factor(red_wine$quality)) #assumes equal covariance?
```

```
##
##  One-way MANOVA (Bartlett Chi2)
##
## data:  x
## Wilks' Lambda = 0.54965, Chi2-Value = 951.28, DF = 55.00, p-value <
## 2.2e-16
## sample estimates:
##   fixed acidity volatile acidity citric acid residual sugar  chlorides
## 3      8.360000        0.8845000   0.1710000       2.635000 0.12250000
## 4      7.779245        0.6939623   0.1741509       2.694340 0.09067925
## 5      8.167254        0.5770411   0.2436858       2.528855 0.09273568
## 6      8.347179        0.4974843   0.2738245       2.477194 0.08495611
## 7      8.872362        0.4039196   0.3751759       2.720603 0.07658794
```

```
## 8      8.566667        0.4233333   0.3911111       2.577778 0.06844444
##   free sulfur dioxide total sulfur dioxide   density       pH sulphates
## 3            11.00000             24.90000 0.9974640 3.398000 0.5700000
## 4            12.26415             36.24528 0.9965425 3.381509 0.5964151
## 5            16.98385             56.51395 0.9971036 3.304949 0.6209692
## 6            15.71160             40.86991 0.9966151 3.318072 0.6753292
## 7            14.04523             35.02010 0.9961043 3.290754 0.7412563
## 8            13.27778             33.44444 0.9952122 3.267222 0.7677778
##     alcohol
## 3  9.955000
## 4 10.265094
## 5  9.899706
## 6 10.629519
## 7 11.465913
## 8 12.094444
```

What if you group the wines into Low (Quality 3 – 4), Medium (Quality 5 – 6), and High (Quality 7 - 8)?

```
grouped<- red_wine%>% mutate(Level = ifelse(quality == 3 | quality == 4,"Low",ifelse(quality == 5 | qual

Wilks.test(grouped[,1:11],grouping=as.factor(grouped$Level))
```

```
##
##  One-way MANOVA (Bartlett Chi2)
##
## data:  x
## Wilks' Lambda = 0.70562, Chi2-Value = 554.75, DF = 22.00, p-value <
## 2.2e-16
## sample estimates:
##        fixed acidity volatile acidity citric acid residual sugar  chlorides
## High        8.847005        0.4055300   0.3764977       2.708756 0.07591244
## Low         7.871429        0.7242063   0.1736508       2.684921 0.09573016
## Medium      8.254284        0.5385595   0.2582638       2.503867 0.08897271
##        free sulfur dioxide total sulfur dioxide   density       pH sulphates
## High              13.98157             34.88940 0.9960303 3.288802 0.7434562
## Low               12.06349             34.44444 0.9966887 3.384127 0.5922222
## Medium            16.36846             48.94693 0.9968673 3.311296 0.6472631
##         alcohol
## High   11.51805
## Low    10.21587
## Medium 10.25272
```

b) Classification/Prediction: come up with a rule for predicting wine quality based on the 11 chemical attributes. You could consider methods like linear regression, nearest neighbors, classification and regression trees, etc.

```
#Knn

#I couldn't get Knn to run, after some googling, it turned out that the C code didn't like that my "Lev
grouped2<- red_wine%>% mutate(Level = ifelse(quality == 3 | quality == 4,1,ifelse(quality == 5 | qualit

set.seed(1234567)
#standardize the data
```

```
red_group_standardized <- cbind(red_standardized[,-12],grouped2[,12:13])


sample <- sample(nrow(red_group_standardized),size = 1000)
sorted_sample <- sort(sample)

training_set <- red_group_standardized[sorted_sample,]
testing_set <- red_group_standardized[-sorted_sample,]

#k=3
level_knn.3 <- knn(training_set, testing_set, cl=training_set$Level, prob=TRUE, k=3)
#knn.3

table(testing_set$Level,level_knn.3)
```

```
##    level_knn.3
##       1   2   3
##   1  11   3   0
##   2   0 505   6
##   3   0  13  61
```

```
sum(testing_set$Level!= level_knn.3)/length(level_knn.3)
```

```
## [1] 0.03672788
```

```
#k=3
quality_knn.3 <- knn(training_set, testing_set, cl=training_set$quality, prob=TRUE, k=3)

table(testing_set$quality,quality_knn.3)
```

```
##    quality_knn.3
##       3   4   5   6   7   8
##   3   1   0   0   1   0   0
##   4   0  10   2   0   0   0
##   5   0   0 234  36   0   0
##   6   0   0  19 215   7   0
##   7   0   0   0  13  54   3
##   8   0   0   0   0   4   0
```

```
mean(testing_set$quality != quality_knn.3)
```

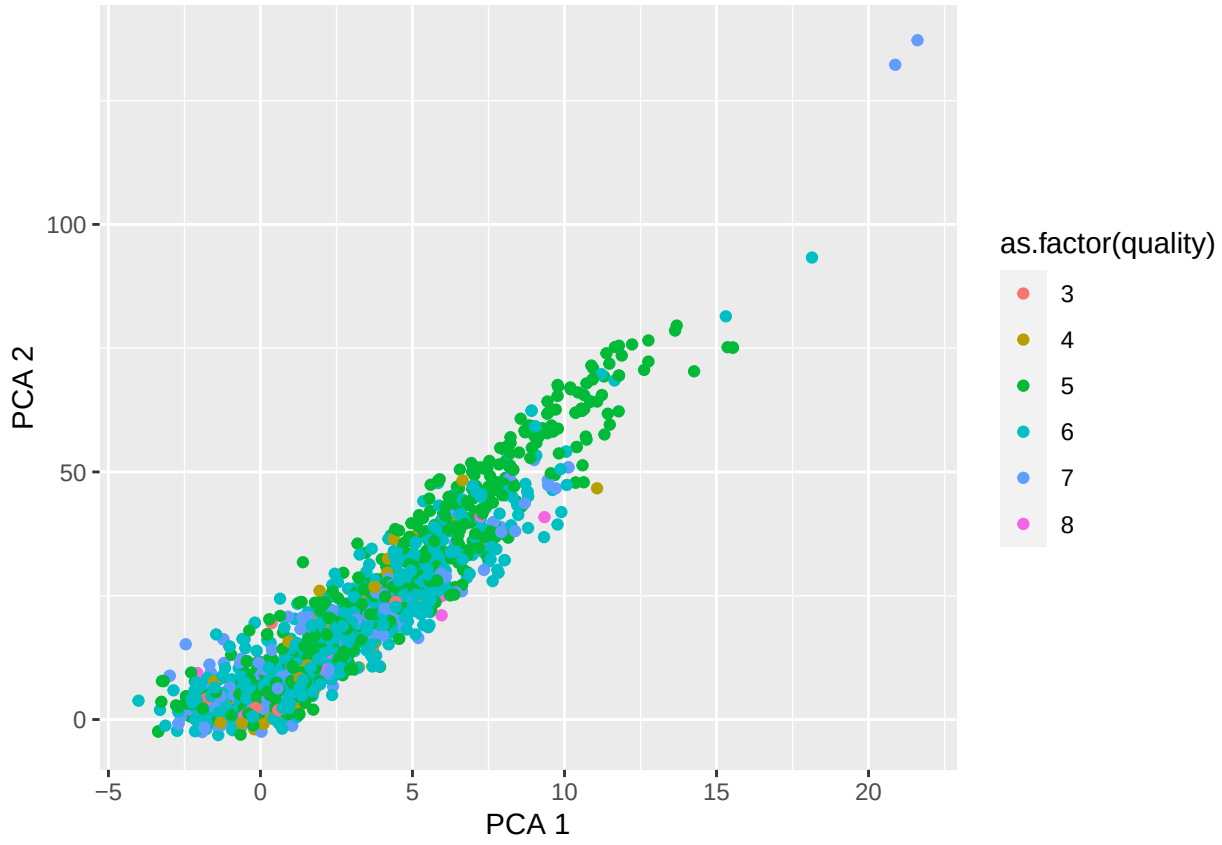
```
## [1] 0.1419032
```

c) Perform PCA on the 11 chemical attributes, and plot the wines color coded by quality score. Did you standardize the data before performing PCA? Why or why not?

```
#PCA
red_cor <- cor(red_wine[,-12])
red_pca <- prcomp(red_cor)
summary(red_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4    PC5    PC6     PC7
## Standard deviation     0.9296 0.5658 0.4645 0.38260 0.2526 0.2084 0.16296
## Proportion of Variance 0.5094 0.1887 0.1272 0.08629 0.0376 0.0256 0.01565
## Cumulative Proportion  0.5094 0.6981 0.8253 0.91156 0.9492 0.9748 0.99041
##                           PC8     PC9    PC10      PC11
## Standard deviation     0.10992 0.05744 0.02968 8.501e-17
## Proportion of Variance 0.00712 0.00194 0.00052 0.000e+00
## Cumulative Proportion  0.99754 0.99948 1.00000 1.000e+00
```

```
p1 <- ggplot(red_wine, aes(x=as.matrix(red_wine[,-12])%*%as.matrix(red_pca$rotation[,1]), y=as.matrix(r
  geom_point()+
  xlab("PCA 1") +ylab("PCA 2")
p1
```



Come up with another rule for predicting wine quality, but now use only the scores for the first two principal components. How does the performance of this new classifier compare to the classifier that used all the predictor variables? Explain how you compared performance.

```
pca_dataframe <- data.frame(cbind(red_wine$quality,red_pca$x[,1], red_pca$x[,2]))
```

```
## Warning in cbind(red_wine$quality, red_pca$x[, 1], red_pca$x[, 2]): number of
## rows of result is not a multiple of vector length (arg 2)
```

```
colnames(pca_dataframe) <- c('quality', 'pca.x1', 'pca.x2')
sample2 <- sample(nrow(pca_dataframe), 1000)
sorted_sample2 <- sort(sample2)

training_set2 <- pca_dataframe[sorted_sample2,]
testing_set2 <- pca_dataframe[-sorted_sample2,]

pca_knn.3 <- knn(training_set2, testing_set2, cl=training_set2$quality, prob=TRUE, k=3)

table(testing_set2$quality,pca_knn.3)
```

```
##    pca_knn.3
##       3   4   5   6   7   8
##   3   1   5   0   0   0   0
##   4   0  17   0   0   0   0
##   5   0   0 254   0   0   0
##   6   0   0   0 245   0   0
##   7   0   0   0   0  72   0
##   8   0   0   0   0   1   4
```

```
mean(testing_set2$quality != pca_knn.3)
```

```
## [1] 0.01001669
```

#Code Section 2: Plots

```
par(mfrow=c(3,4))
 boxplot(red_wine$fixed.acidity, white_wine$`fixed acidity`, horizontal = TRUE, main="Fixed Acidity", yl
```

```
## Warning: Unknown or uninitialised column: ‘fixed.acidity‘.
```

```
boxplot(red_wine$volatile.acidity, white_wine$`volatile acidity`, horizontal = TRUE, main="Volatile Acid
```

```
## Warning: Unknown or uninitialised column: ‘volatile.acidity‘.
```

```
 boxplot(red_wine$citric.acid, white_wine$`citric acid`, horizontal = TRUE, main="Citric acid", ylab="re
```

```
## Warning: Unknown or uninitialised column: ‘citric.acid‘.
```

```
boxplot(red_wine$residual.sugar, white_wine$`residual sugar`, horizontal = TRUE, main="Residual sugar",
```

```
## Warning: Unknown or uninitialised column: ‘residual.sugar‘.
```

```
boxplot(red_wine$chlorides, white_wine$chlorides, horizontal = TRUE, main="Chlorides", ylab="red white"]

boxplot(red_wine$fsd, white_wine$`free sulfur dioxide`, horizontal = TRUE, main="Free Sulfur Dioxide", y
```

```
## Warning: Unknown or uninitialised column: ‘fsd‘.
```

```
boxplot(red_wine$tsd, white_wine$`total sulfur dioxide`, horizontal = TRUE, main="Total Sulfur Dioxide"
```

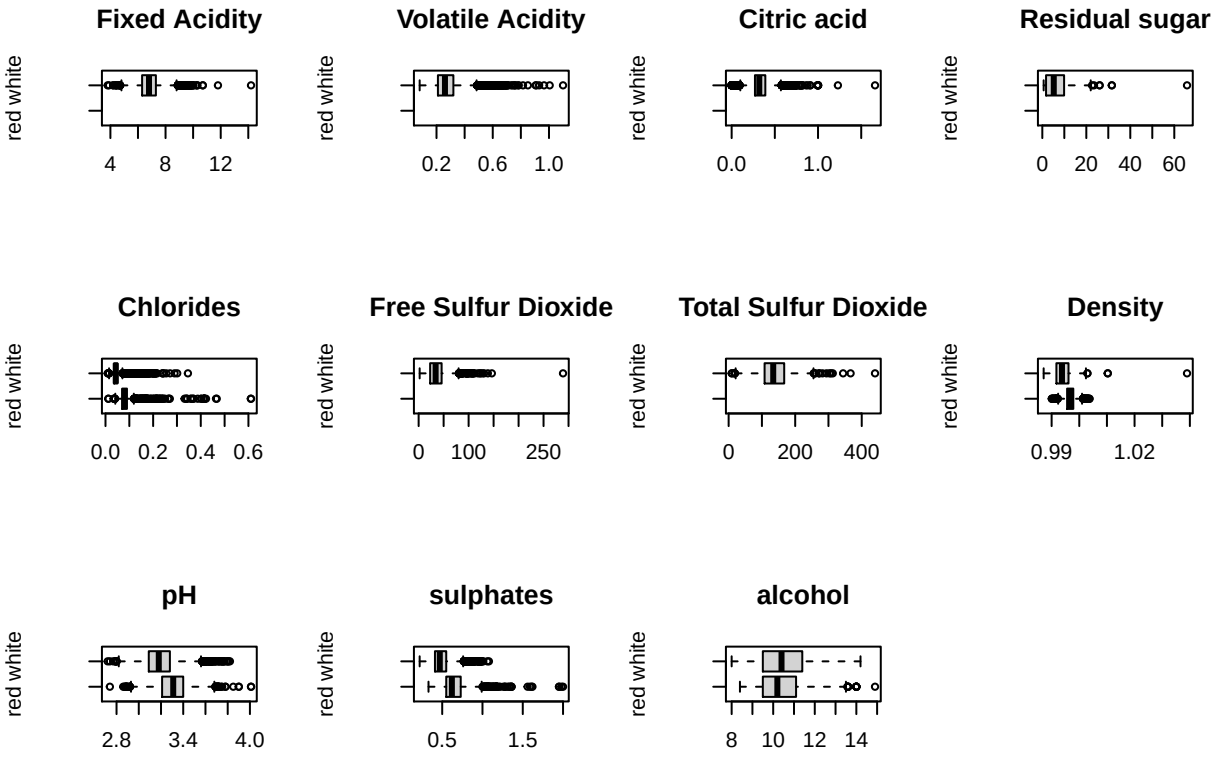
```
## Warning: Unknown or uninitialised column: 'tsd'.
```

```
boxplot(red_wine$density, white_wine$density, horizontal = TRUE, main="Density", ylab="red white")
```

```
boxplot(red_wine$pH, white_wine$pH, horizontal = TRUE, main="pH", ylab="red white")
```

```
boxplot(red_wine$sulphates, white_wine$sulphates, horizontal = TRUE, main="sulphates", ylab="red white"
```

```
boxplot(red_wine$alcohol, white_wine$alcohol, horizontal = TRUE, main="alcohol", ylab="red white")
```




```
par(mfrow=c(2,4))
```

```
boxplot(red_wine$tsd, white_wine$`total sulfur dioxide`, horizontal = TRUE, main="Total Sulfur Dioxide"
```

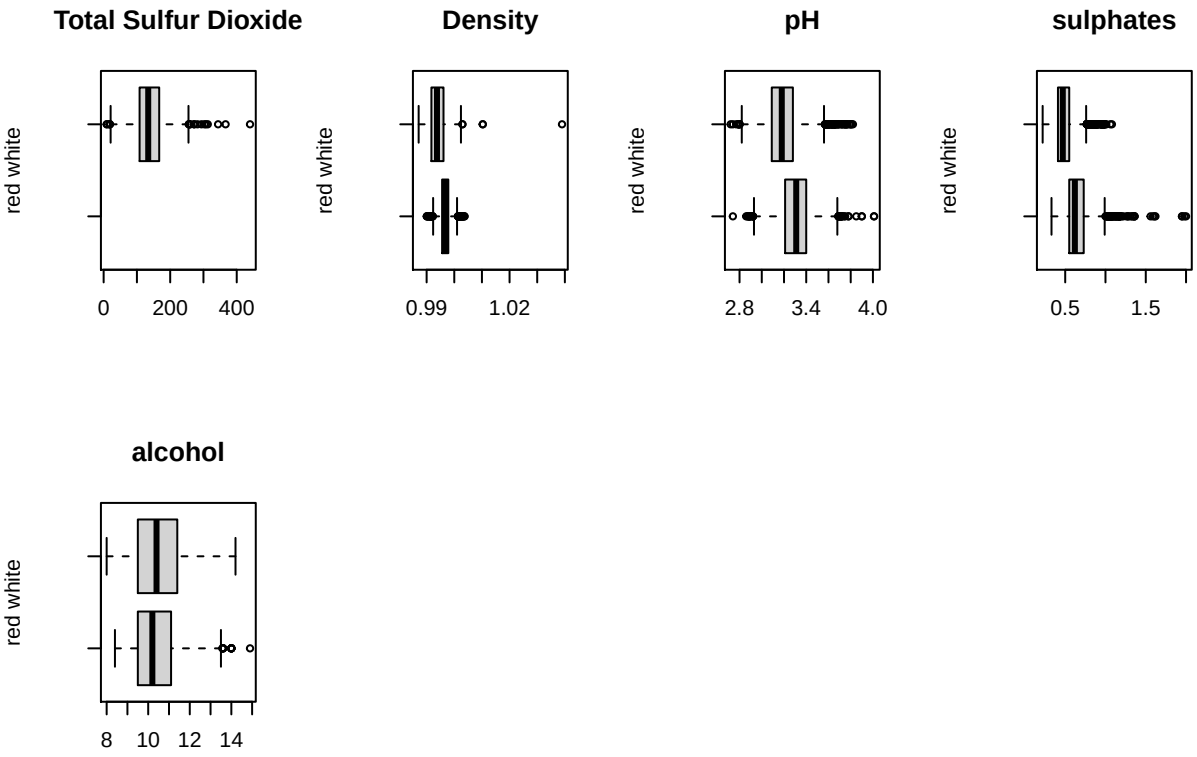
```
## Warning: Unknown or uninitialised column: 'tsd'.
```

```
boxplot(red_wine$density, white_wine$density, horizontal = TRUE, main="Density", ylab="red white")
```

```
boxplot(red_wine$pH, white_wine$pH, horizontal = TRUE, main="pH", ylab="red white")
```

```
boxplot(red_wine$sulphates, white_wine$sulphates, horizontal = TRUE, main="sulphates", ylab="red white"
```

```
boxplot(red_wine$alcohol, white_wine$alcohol, horizontal = TRUE, main="alcohol", ylab="red white")
```


Total Sulfur Dioxide
red white
0
200
400
Density
red white
0.99
1.02
pH
red white
2.8
3.4
4.0
sulphates
red white
0.5
1.5
alcohol
red white
8
10
12
14