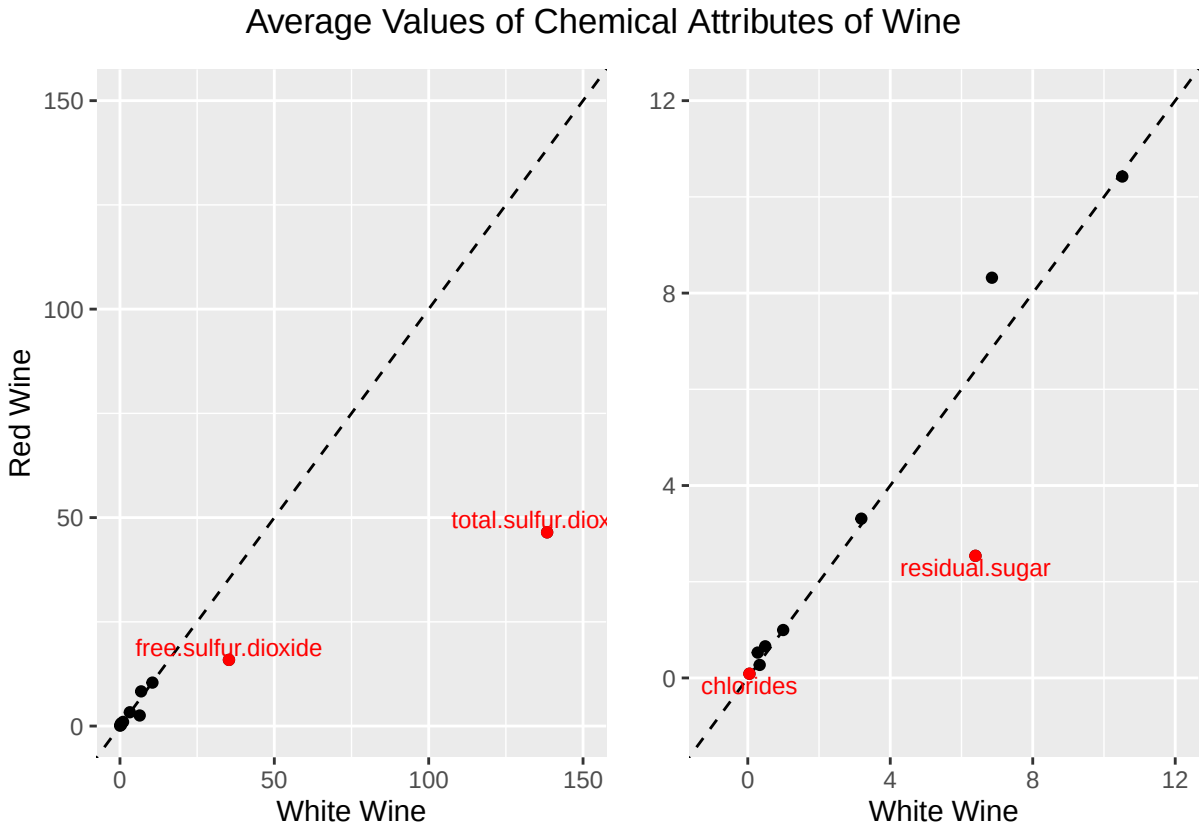
Final Project - Shannon Leiss
Distinguishing Between Red and White Wines

The 11 chemical attributes that contribute to the quality score of Red and White Wines are the following: Fixed Acidity(F.Acid), Volatile Acidity(V.Acid), Citric Acid(C.Acid), Residual Sugars(R.Sugars), Chlorides(Chol), Free Sulfur Dioxide(F.Sulf), Total Sulfur Dioxide(T.Sulf), Density(Dens), pH, Sulphates(Sulf), and Alcohol(Alc). Each observation of both Red and White wines was measured on each of these 11 chemicals and then given a quality score from 3-9. The mean vector for each of the chemical attributes for the White and Red Wines, as well as the difference of the means, is as follows:

	F.Acid	V.Acid	C.Acid	R.Sugars	Chol	F.Sulf	T.Sulf	Dens	pH	Sulf	Alc
White	6.8548	0.2782	0.3342	6.3914	0.0458	35.3081	138.3607	0.9940	3.1883	0.4898	10.5143
Red	8.3196	0.5278	0.2710	2.5388	0.0875	15.8749	46.4678	0.9967	3.3111	0.6581	10.4230
Difference	-	-	0.0632	3.8526	-	19.4332	91.8929	-	-	-	0.0913
	1.4648	0.2496			0.0417			0.0027	0.1228	0.1683	

Using the Two Sample Hotelling's T^2 test for the null hypothesis that the mean vectors between Red and White Wines were the same and the alternative hypothesis that the mean chemical vectors between the two wines were not the same, a test statistic of $T^2 = 40427.9$ was found, which produced a p-value of $p < 0.00000001$. Additional information that was produced by this test was an F statistic of 3669.6, with degrees of freedom equal to 11 and 6485. From this p-value, there is significant evidence that the mean vectors for the 11 chemical elements are not the same between White and Red wine, meaning that the chemical make up of the two wines are different.

Using the Bonferroni multiple testing method, it was found that all chemicals means were significantly different between the two types of wine besides the alcohol variable. To investigate which chemicals differed the most between the two types of wine, the below graphics can be used. The below graph shows the average chemical values for Red wine against the average chemical values for White wine, along with a dotted line that shows where the points would land if the chemical averages were the same between the two wines.



From the above graph, it can be seen that the following four chemical aspects are highlighted: Total Sulfur Dioxide, Free Sulfur Dioxide, Residual Sugar, and Chlorides. These four chemicals were highlighted since the mean for these chemicals for one wine type were almost/over double the mean value of the corresponding chemical in the other wine type - meaning that these chemicals are the ones that differ the most between wine types. Of the four chemicals, only Chlorides had a greater mean for Red wine, while the other three were greater for the White wines.

Since it was found that all chemicals - but alcohol - were significantly different between the two types of wine, a classification rule was able to be made to group new wine observations into the correct color. Comparing the covariance structure of the chemical values for both Red and White wine, it was found that the structure was similar enough to use Linear Discriminant Analysis - for simplicity and interpretation advantage over Quadratic Discriminant Analysis. Using the `lda` function in R, it was found that the type of a new wine can be classified using the following formula:

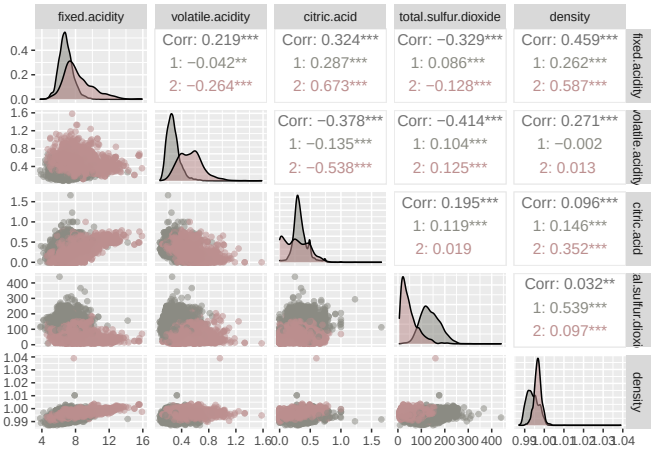
a^T = (-0.3233 3.0603 -0.8750 -0.3512 5.0884 0.0192 -0.0201 910.4347 -1.1061 0.8651 0.8252)

Where the new wine will be classified as a White Wine if the separation value is less than 907.9 and will be classified as a Red Wine if the separation is greater than 907.9. The performance of this classification can be seen in the below confusion matrix.

	White Wine	Red Wine
White Wine	4882	19
Red Wine	16	1580

It can also be noted that the apparent error rate of this classification method is 0.5387%, meaning that this classification method will assign a new wine the wrong type .5% of the time. From the above confusion matrix, it can also be found that the probability of classifying a new Red Wine correctly is 98.81176% when using this classification rule.

Since the data set of all wines has over 5000 observations measured on 11 different chemical attributes, the best way to cluster these observations would be by using k-means clustering. K-means clustering works best when the data set is large as the pairwise distances did not need to be stored. Using this clustering method, with 2 centers(clusters), the following graph shows the pairs plots for a selected number of chemical attributes - the pairs plot with all attributes can be found in the appendix -with the color of the points indicating which cluster that observation was assigned to - grey equating to cluster 1 and light red equating to cluster 2.



Looking at the clusters, it can be seen that cluster 1 is the White Wine cluster and cluster 2 is the Red Wine cluster. In the k-means clustering, 4854 of the wines were placed in the first cluster and 1643 of the wines were placed in the second cluster, Which is similar to the amount of wines that were White(4898) and Red(1599). The clustering can be generally thought of as a White Wine and Red wine cluster since the first cluster houses 98.6% of all White Wines and the second cluster houses 98.5% of the Red Wines. Overall, Red and White wines can be distinctly identified over 95% of the time based off of the 11 chemical attribute measurements.

Understanding Chemical Contribution to Quality Score

For this section, the Red Wine data will be used. The range of quality scores for Red Wines ranges from 3 to 8, with 3 being the lower quality score and 8 being the higher quality wine. The mean values of each chemical broken down by each quality score are as follows:

	3	4	5	6	7	8
fixed.acidity	8.3600	7.7792	8.1673	8.3472	8.8724	8.5667
volatile.acidity	0.8845	0.6940	0.5770	0.4975	0.4039	0.4233
citric.acid	0.1710	0.1742	0.2437	0.2738	0.3752	0.3911
residual.sugar	2.6350	2.6943	2.5289	2.4772	2.7206	2.5778
chlorides	0.1225	0.0907	0.0927	0.0850	0.0766	0.0684
free.sulfur.dioxide	11.0000	12.2642	16.9838	15.7116	14.0452	13.2778
total.sulfur.dioxide	24.9000	36.2453	56.5140	40.8699	35.0201	33.4444
density	0.9975	0.9965	0.9971	0.9966	0.9961	0.9952
pH	3.3980	3.3815	3.3049	3.3181	3.2908	3.2672
sulphates	0.5700	0.5964	0.6210	0.6753	0.7413	0.7678
alcohol	9.9550	10.2651	9.8997	10.6295	11.4659	12.0944

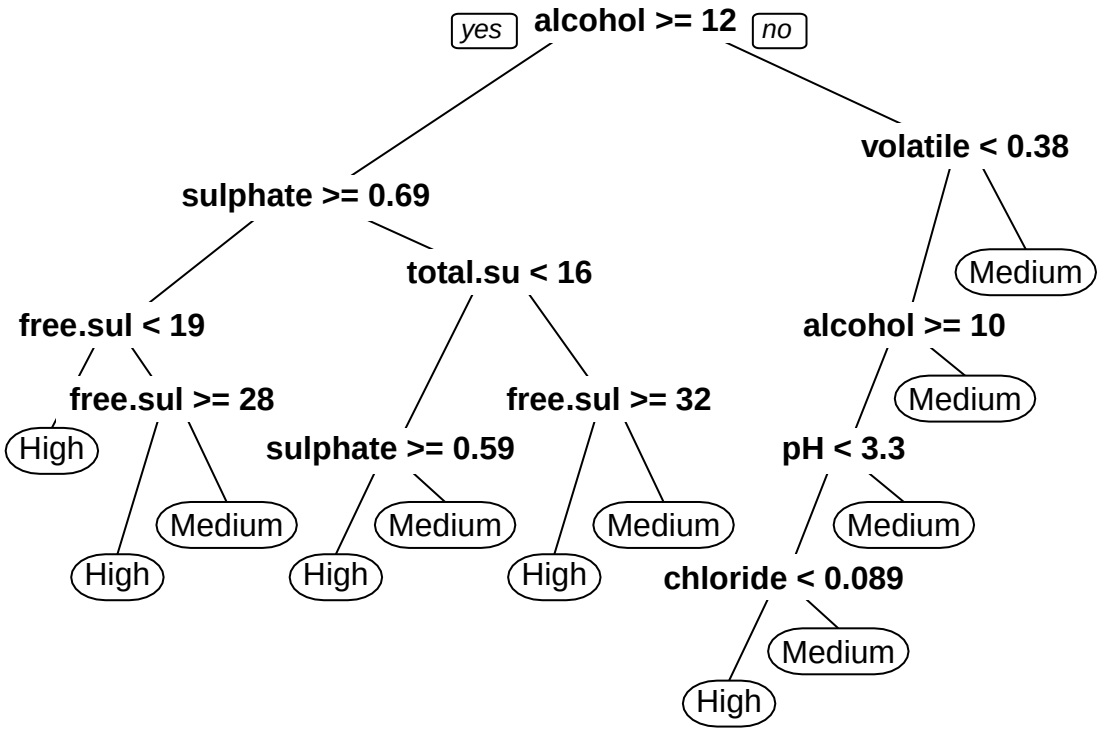
To test if there is a difference between the mean values of all the chemical between each quality score, a MANOVA using Wilks test was run for the null hypothesis that the mean chemical values for each quality score were equal. The test produced a Wilks' Lambda value of $\Lambda = 0.54965$ with a p-value of $p < 0.00001$. Since both of these values are small, there is significant evidence that the mean chemical vectors are not equal between the different quality scores. The MANOVA also produced a $\chi^2 = 951.28$ with 55 degrees of freedom. Running simultaneous univariate ANOVA's and comparing the p-values to the α^* level of 0.004545, it was found that all chemical mean vectors were significantly different between the quality scores except residual sugars.

The quality of wines can also be grouped into levels of Low(3-4), Medium(5-6), and High(7-8). Below are the chemical mean vectors for each of the quality levels:

	F.Acid	V.Acid	C.Acid	R.Sugars	Chol	F.Sulfur	T.Sulfur	Density	pH	Sulph	Alcohol
Low	7.8714	0.7242	0.1737	2.6849	0.0957	12.0635	34.4444	0.9967	3.3841	0.5922	10.2159
Medium	8.2543	0.5386	0.2583	2.5039	0.0890	16.3685	48.9469	0.9969	3.3113	0.6473	10.2527
High	8.8470	0.4055	0.3765	2.7088	0.0759	13.9816	34.8894	0.9960	3.2888	0.7435	11.5180

To test if there is a difference between the mean values of all the chemical between each quality level, a MANOVA using Wilks test was run for the null hypothesis that the mean chemical values for each quality level were equal. The test produced a Wilks' Lambda value of $\Lambda = 0.70562$ with a p-value of $p < 0.00001$. Since both of these values are small, there is significant evidence that the mean chemical vectors are not equal between the different quality levels. The MANOVA also produced a $\chi^2 = 554.75$ with 22 degrees of freedom. Running simultaneous univariate ANOVA's and comparing the p-values to the α^* level of 0.004545, it was found that all chemical mean vectors were significantly different between the quality levels except residual sugars, which was the same conclusion for when the quality of wine was a score instead of a level.

Just as a classification was able to be created to group a wine into its type based on the chemical attributes, classifications can also be made to group a wine into its quality based on the chemical attributes. To classify the wines, the quality levels - Low, Medium, and High - were used to create the classification rules; classification rules based on the quality score are explored in the appendix section, but were left out due to worse performance. One possible classification rule for the wine quality level was found using a classification tree. The pruned version of the tree is found below, which was pruned by taking the values that minimized the error rates.



From the plot, it can be seen that the alcohol content is the "most important" variable is deciding what quality the wine has, then the next most important variable is density. Using the Red wine data, it was found that the above classification had an error rate of 10.569%. The confusion matrix for this classification is provided below, with the columns representing a wines true quality level and the rows representing the predicted quality level.

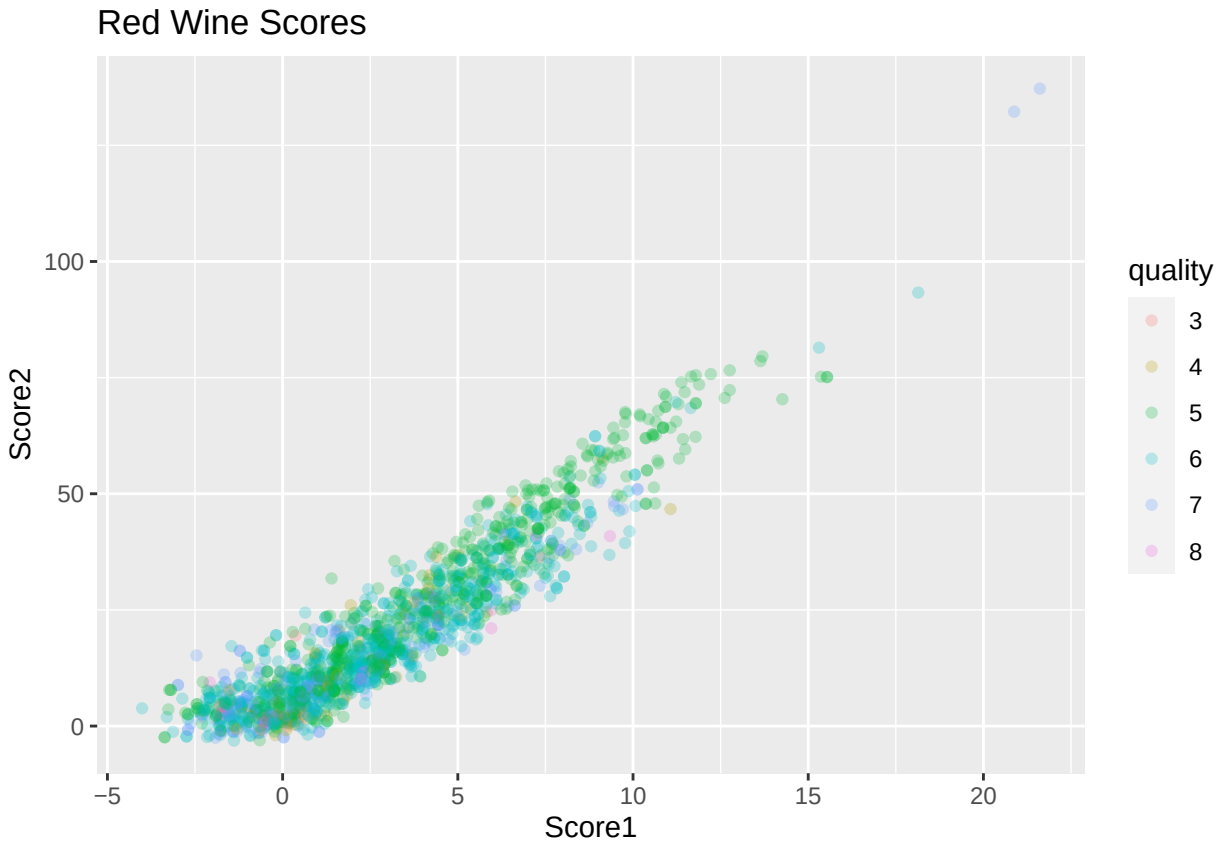
	High	Low	Medium
High	160	2	61
Low	0	17	5
Medium	57	44	1253

Another way the quality level of Red wines can be defined is using the k-nearest neighbors approach. For this approach, the Red Wine values were standardized and k was set to 3. This approach is included since it reduces the prediction error rate of the classification to 9.255%. The confusion matrix for this approach is given below.

	High	Low	Medium
High	157	2	39
Low	0	19	5
Medium	60	42	1275

Performing Principal Component Analysis on the correlation matrix of Red Wine - correlation matrix was used since it standardizes the data and is equal to the covariance matrix of the standardized data - the score values for the first two components of the Red Wine data are graphed below. The first two components had the following values:

	F.Acid	V.Acid	C.Acid	R.Sugars	Chol	F.Sulf	T.Sulf	Dens	pH	Sulf	Alc
PC1	-0.5003	0.2761	-0.4616	-0.0764	-0.1748	0.1281	0.0667	-0.3633	0.4632	-0.2072	0.1172
PC2	-0.0196	0.3123	-0.1729	0.1773	0.1400	0.3491	0.4503	0.3366	-0.1135	-0.1514	-0.5894



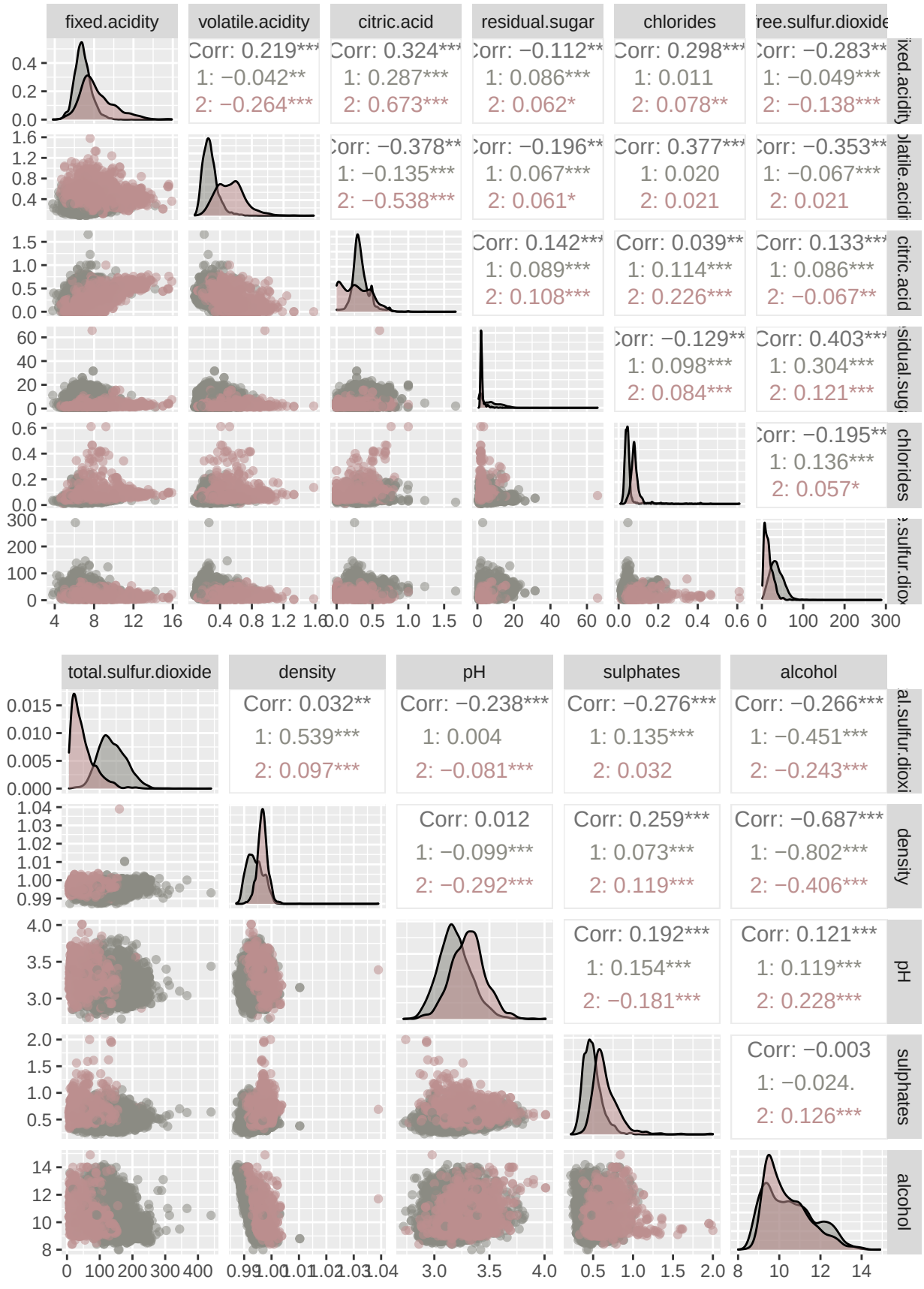
From the principal component analysis, it was found that the first two components account for 69.81% of the variability in the data. Using the quality levels to create classifications - see appendix for exploration of using quality scores - a classification tree and k nearest neighbors classifications were performed. The classification tree can be found in the appendix - due to the unpruned tree being very large and the pruned tree having a single branch that predicted ever wine to be of Medium quality. The confusion matrix for this classification is as follows:

	High	Low	Medium
High	80	4	47
Low	4	10	5
Medium	133	49	1267

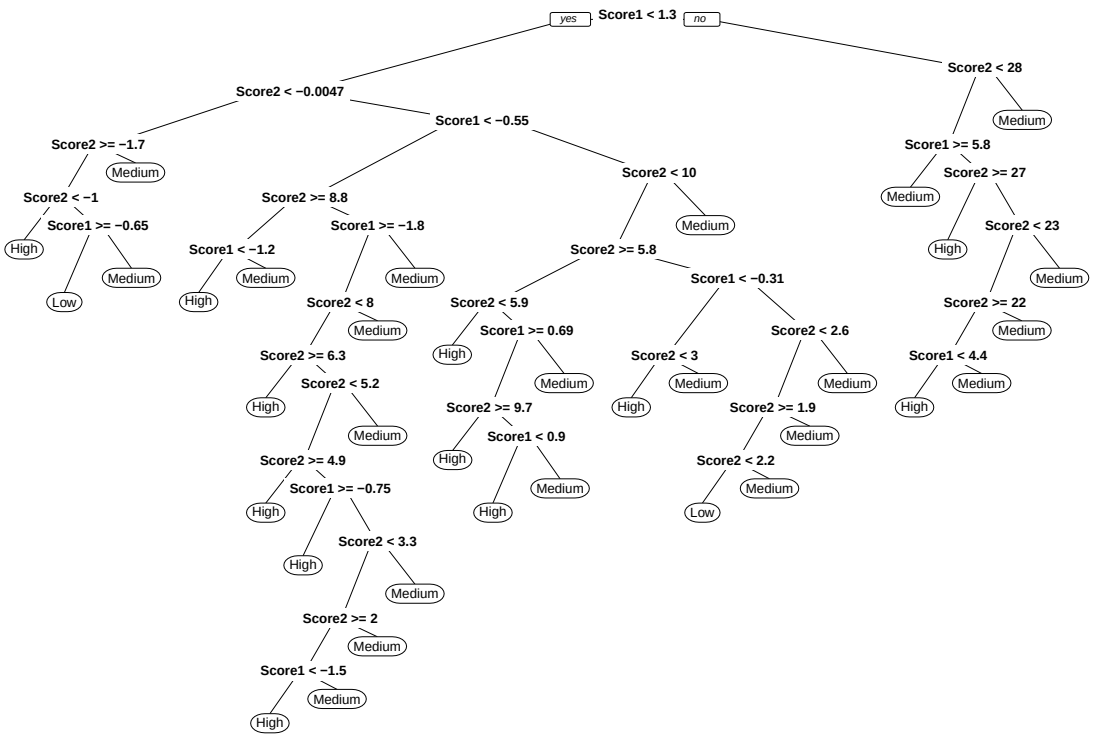
The error rate of this classifier is 15.134%. This is classifier is about 69.83% worse than the classifier created using all of the predictor variables. This makes sense since the first two principal components only account for about 69.81% of the variation found in the data. This relationship can also be explored by finding the k-nearest neighbors, when k=3, for the first two principal components. Using the standardized data to find the first two score values and then using those values for the k-nearest neighbors, it was found that the classifier had an error rate of 11.006%, which is greater than the error rate when all predictors are used. This error rate is about 84% worse than the k-nearest neighbors with all predictors, bolstering the conclusion that the classifiers created using only the first two principal components preform worse than the classifiers created using all the explanatory variables.

Appendix

Pairs Plot



Classification Tree for PCA Quality Levels



R Code, Including Comments

```
##1a
##Mean Vectors for each Wine Type
Mean_White <- colMeans(White[,-12])
Mean_Red <- colMeans(Red[,-12])
##Hottelings two sample T^2 test
T2.test(x=White[,-12],y=Red[,-12])
## Standardize - Used in part 2
SD_White <- map_dbl(White,sd)
SD_Red <- map_dbl(Red,sd)
White.std <- White
for(i in 1:nrow(White)){
  White.std[i,] <- (White[i,]-Mean_White)/SD_White
}
Red.std <- Red
for(i in 1:nrow(Red)){
  Red.std[i,] <- (Red[i,]-Mean_Red)/SD_Red
}
##Differences
Dif_Means <- Mean_White-Mean_Red
Dif_Means
##Simultaneous t tests
.05/22
```

```r
for(i in 1:11){
  cat(colnames(Red[i]),":",(t.test(x=White[,i],y=Red[,i], conf.level = 1-(.05/22),alternative = "two.sided
}

##Rule for large difference: about double
means <- as.data.frame(cbind(Mean_White,Mean_Red))
p1 <- ggplot(data=means,aes(x=Mean_White,y=Mean_Red))+
  geom_point()+
  xlim(c(0,150))+
  ylim(c(0,150))+
  geom_point(data=means[c(6,7),], aes(x=Mean_White,y=Mean_Red),col="red")+
  geom_abline(intercept = 0,slope=1,linetype = "dashed")+
  geom_text(data=means[c(6,7),],label = rownames(means[c(6,7),]),nudge_y = 3,size=3,col="red")+
  xlab("White Wine")+
  ylab("Red Wine")
p2 <- ggplot(data=means[-c(6,7),],aes(x=Mean_White,y=Mean_Red))+
  geom_point()+
  xlim(c(-1,12))+
  ylim(c(-1,12))+
  geom_abline(slope = 1,intercept=0,linetype = "dashed")+
  geom_point(data=means[c(4,5),], aes(x=Mean_White,y=Mean_Red),col="red")+
  geom_text(data=means[c(4,5),],label = rownames(means[c(4,5),]),nudge_y = -.25,size=3,col="red")+
  xlab("White Wine")+
  ylab(element_blank())
ggp_all <- (p1 + p2) +
  plot_annotation(title = "Average Values of Chemical Attributes of Wine") &
  theme(plot.title = element_text(hjust = 0.5))
ggp_all

##1b

##Use lda when covariance matrices are similar
White.class <- White %>%
  mutate(Type = rep(1,nrow(White)))
Red.class <- Red %>%
  mutate(Type= rep(2,nrow(Red)))
##Data Set of all Wines with a Class Column
Wine <- as.data.frame(rbind(White.class,Red.class))
Wine <- Wine[,-12]##Removing Quality
##Compare Covariance Matrices
RW <- cov(White[,1:11])
RR <- cov(Red[,1:11])
RW-RR
##LDA to classify
Wine.lda <- lda(Type~., data=Wine)
Wine.lda ##Scaled Output
Predicted <- as.numeric(predict(Wine.lda, newdata=Wine[,-12])$class)
Wine.Updated <- Wine %>%
  mutate(Predicted = Predicted, Seperation = as.matrix(Wine[,-12])%*%as.matrix(Wine.lda$scaling))
##Confusion Matrix
White.Up <- Wine.Updated[Wine.Updated$Type == 1,]
Red.Up <- Wine.Updated[Wine.Updated$Type==2,]
True_W <- nrow(White.Up[White.Up$Predicted==1,])
True_R <- nrow(Red.Up[Red.Up$Predicted==2,])
Wrong_W <- nrow(White.Up[White.Up$Predicted==2,])
Wrong_R <- nrow(Red.Up[Red.Up$Predicted==1,])
```

```
Confusion <-data.frame("White Wine" = c(True_W,Wrong_W), "Red Wine" = c(Wrong_R,True_R))
rownames(Confusion) <- c("White Wine", "Red Wine")
colnames(Confusion) <- c("White Wine", "Red Wine")
knitr::kable(Confusion)
##Error Rate
APER <- (Wrong_W+Wrong_R)/(nrow(Wine.Updated))
APER
##Probability of correctly predicting Red Wine
Red.Draw <- True_R/nrow(Wine.Updated[Wine.Updated$Type == 2,])
Red.Draw

##1c

## Pairs PLots broken on Type
ggpairs(Wine, columns = 1:6,aes(color=as.factor(Type), alpha=.5))
ggpairs(Wine, columns = 7:11,aes(color=as.factor(Type), alpha=.5))
##K-Means Clustering = 2
set.seed(101214)
Winek2 <- kmeans(scale(Wine[,-12]),center=2)
Wine.Cluster <- Wine
Wine.Cluster$K2 <- Winek2$cluster
Wine.Cluster$Correct <- ifelse(Wine.Cluster$K2 == Wine.Cluster$Type,1,0)
##Pairs Plots of Clusters
ggpairs(Wine.Cluster, columns = 1:6,aes(color=as.factor(K2), alpha=.5))+
  scale_color_manual(values=c("ivory4","rosybrown"))+
  scale_fill_manual(values=c("ivory4","rosybrown"))
ggpairs(Wine.Cluster, columns = 7:11,aes(color=as.factor(K2), alpha=.5))+
  scale_color_manual(values=c("ivory4","rosybrown"))+
  scale_fill_manual(values=c("ivory4","rosybrown"))
##Confusion Matrix
table(Wine.Cluster$Type,Wine.Cluster$K2)
##Accuracy of Clusters
92/nrow(Wine.Cluster)##1.4% or 98.58% accuracy

##2a

##Quality Scores Means
Quality <- unique(Red$quality)
Red.Mean <- matrix(nrow=6,ncol=12)
for(i in 1:length(Quality)){
  Red.Mean[i,] <- Red %>% subset(quality == Quality[i])%>%colMeans()
}
colnames(Red.Mean) <- colnames(Red)
Red.Mean <- Red.Mean[order(Red.Mean[,12]),]
## Quality Score MANOVA
Wilks.test(Red[,1:11],grouping=as.factor(Red$quality))
### Univariate ANOVASto see which variables are different
a_star <- .05/11
(anova(lm(fixed.acidity~as.factor(quality),data=Red))$`Pr(>F)` ) < a_star
(anova(lm(volatile.acidity~as.factor(quality),data=Red))$`Pr(>F)`)< a_star
(anova(lm(citric.acid~as.factor(quality),data=Red))$`Pr(>F)`)< a_star
(anova(lm(residual.sugar~as.factor(quality),data=Red))$`Pr(>F)`)< a_star
(anova(lm(chlorides~as.factor(quality),data=Red))$`Pr(>F)`)< a_star
(anova(lm(free.sulfur.dioxide~as.factor(quality),data=Red))$`Pr(>F)`)< a_star
(anova(lm(total.sulfur.dioxide~as.factor(quality),data=Red))$`Pr(>F)`)< a_star
(anova(lm(density~as.factor(quality),data=Red))$`Pr(>F)`)< a_star
```

```
(anova(lm(pH~as.factor(quality),data=Red))$`Pr(>F)`)< a_star
(anova(lm(sulphates~as.factor(quality),data=Red))$`Pr(>F)`)< a_star
(anova(lm(alcohol~as.factor(quality),data=Red))$`Pr(>F)`)< a_star
##Grouping Quality
Red.grouped<- Red%>% mutate(Level = ifelse(quality == 3 | quality == 4,"Low",ifelse(quality == 5 | quality
Red.Mean.Level <- matrix(nrow=3,ncol=11)
Levels <- c("Low", "Medium", "High")
for(i in 1:3){
  place_hold <- Red.grouped %>% subset(Level == Levels[i])
  Red.Mean.Level[i,] <- place_hold[,1:11]%>%colMeans()
}
colnames(Red.Mean.Level) <- colnames(Red[,1:11])
rownames(Red.Mean.Level) <- Levels
##Grouped MANOVA
Wilks.test(Red.grouped[,1:11],grouping=as.factor(Red.grouped$Level))
##Univariate Anovas to see which variables are different
a_star <- .05/11
(anova(lm(fixed.acidity~as.factor(Level),data=Red.grouped))$`Pr(>F)` ) < a_star
(anova(lm(volatile.acidity~as.factor(Level),data=Red.grouped))$`Pr(>F)`)< a_star
(anova(lm(citric.acid~as.factor(Level),data=Red.grouped))$`Pr(>F)`)< a_star
(anova(lm(residual.sugar~as.factor(Level),data=Red.grouped))$`Pr(>F)`)< a_star
(anova(lm(chlorides~as.factor(Level),data=Red.grouped))$`Pr(>F)`)< a_star
(anova(lm(free.sulfur.dioxide~as.factor(Level),data=Red.grouped))$`Pr(>F)`)< a_star
(anova(lm(total.sulfur.dioxide~as.factor(Level),data=Red.grouped))$`Pr(>F)`)< a_star
(anova(lm(density~as.factor(Level),data=Red.grouped))$`Pr(>F)`)< a_star
(anova(lm(pH~as.factor(Level),data=Red.grouped))$`Pr(>F)`)< a_star
(anova(lm(sulphates~as.factor(Level),data=Red.grouped))$`Pr(>F)`)< a_star
(anova(lm(alcohol~as.factor(Level),data=Red.grouped))$`Pr(>F)`)< a_star

##2b

##CART For Quality Score
redtree <- rpart(as.factor(quality) ~ ., data=Red, control =rpart.control(cp = 0.0001))
redtree.testPredCl <- predict(redtree,Red[,1:11], type="class")
bestcp <- redtree$cptable[which.min(redtree$cptable[,"xerror"]),"CP"]
tree.pruned <- prune(redtree, cp = bestcp)
prp(tree.pruned)
table(Red$quality, redtree.testPredCl)
mean(Red$quality != redtree.testPredCl)
##K Nearest for Quality Score - Using Standardized data:
Red.std <- Red.std %>% mutate(quality = Red$quality)
set.seed(101214)
knnRslt.3 <- knn(Red.std[,-12], Red.std[,-12], cl=Red$quality, prob=TRUE, k=3)
summary(knnRslt.3)
table(Red$quality,knnRslt.3)
mean(Red$quality!= knnRslt.3)
## CART FOR Level
Red.grouped <- Red.grouped[,-12]
redtree <- rpart(as.factor(Level) ~ ., data=Red.grouped, control =rpart.control(cp = 0.0001))
bestcp <- redtree$cptable[which.min(redtree$cptable[,"xerror"]),"CP"]
tree.pruned <- prune(redtree, cp = bestcp)
prp(tree.pruned)
redtree.testPredCl <- predict(redtree,Red.grouped[,1:11], type="class")
table(Red.grouped$Level, redtree.testPredCl)
mean(Red.grouped$Level != redtree.testPredCl)
##K nearest for Level: Using Standardized Data
```

```
Red.grouped.std<- Red.std %>% mutate(Levels = Red.grouped$Level)
Red.grouped.std <- Red.grouped.std[,-12]
set.seed(101214)
knnRslt.3 <- knn(Red.grouped.std[,1:11], Red.grouped.std[,1:11], cl=Red.grouped.std$Levels, prob=TRUE, k=3
summary(knnRslt.3)
table(Red.grouped.std$Levels,knnRslt.3)
mean(Red.grouped.std$Levels!= knnRslt.3)

##2c

##Use correlation since it is unstandardized
Red.Cor <- Red[,-12] %>% cor()
Red.PCA<- prcomp(Red.Cor)
summary(Red.PCA) ##First two components only account for 69% Variance
Red$Score1 <- as.matrix(Red[,-12])%*%as.matrix(Red.PCA$rotation[,1])
Red$Score2 <- as.matrix(Red[,-c(12,13)])%*%as.matrix(Red.PCA$rotation[,2])
p1 <- ggplot(Red, aes(x=Score1, y=Score2,col=as.factor(quality)))+
  geom_point(alpha=0.25)
##CART
redtree <- rpart(as.factor(quality) ~ Score1+Score2, data=Red, control =rpart.control(cp = 0.001))
prp(redtree)
redtree.testPredCl <- predict(redtree,Red[,13:14], type="class")
table(Red.grouped$quality, redtree.testPredCl)
mean(Red.grouped$quality != redtree.testPredCl)
##Nearest Score
Red$Score1.std <- as.matrix(Red.std[,-12])%*%as.matrix(Red.PCA$rotation[,1])
Red$Score2.std <- as.matrix(Red.std[,-c(12,13)])%*%as.matrix(Red.PCA$rotation[,2])
set.seed(101214)
knnRslt.3 <- knn(Red[,15:16], Red[,15:16], cl=Red$quality, prob=TRUE, k=3)
summary(knnRslt.3)
table(Red$quality, knnRslt.3)
mean(Red$quality!= knnRslt.3)
##CART Group
Red.grouped$Score1 <- as.matrix(Red.grouped[,-12])%*%as.matrix(Red.PCA$rotation[,1])
Red.grouped$Score2 <- as.matrix(Red.grouped[,-c(12,13)])%*%as.matrix(Red.PCA$rotation[,2])
redtree <- rpart(as.factor(Level) ~ Score1+Score2, data=Red.grouped, control =rpart.control(cp = 0.001))
prp(redtree)
redtree.testPredCl <- predict(redtree,Red.grouped[,13:14], type="class")
table(Red.grouped$Level, redtree.testPredCl)
mean(Red.grouped$Level != redtree.testPredCl)
## Nearest Neighbors Grouped Standardized
Red.grouped.std$Score1 <- as.matrix(Red.grouped.std[,-c(12)])%*%as.matrix(Red.PCA$rotation[,1])
Red.grouped.std$Score2 <- as.matrix(Red.grouped.std[,-c(12,13,14)])%*%as.matrix(Red.PCA$rotation[,2])
##Does worse than
set.seed(101214)
knnRslt.3 <- knn(Red.grouped.std[,c(13,14)], Red.grouped.std[,c(13,14)], cl=Red.grouped.std$Levels, prob=T
summary(knnRslt.3)
table(Red.grouped.std$Levels, knnRslt.3)
mean(Red.grouped.std$Levels!= knnRslt.3)
```