Group 17: Project Milestone 3

Hannah Douglas (hcd6tc) Akhil Havaldar (ash2sfp) Skye Jung (hsj5sn)
Nathan Park (ngp7ce)

10/21/2022

Contents

0.1 Section 1: Introduction

0.1.1 The question of interest your group is trying to answer using classification techniques.

From a regression standpoint, our group is trying to see if the alcohol percentage of a wine can be determined by the predictors in the dataset? Specifically, which predictors are most significant in determining the percentage of the wine? From a classification standpoint we are trying to see if we can classify whether or not a wine observation has good or bad quality based on the predictors in the dataset.

0.1.2 A short paragraph to explain why this question is worth exploring.

These questions are worth exploring because we are interested in the factors that go into wine quality and alcohol percentage. The results of our study will demonstrate which predictors are most significant in both predicting the alcohol percentage, and classifying a wine as good or bad. The classification question specifically will be useful for wine makers to know which components of wine will have the biggest impact on wine quality, in addition to predictors that are associated with higher or lower levels of alcohol in their wine.

0.1.3 A short paragraph about where you obtained the data set from.

Our dataset was obtained from Kaggle and was prepossessed and downloaded from the UC Irvine Machine Learning Repository. The data itself was originally from a study by Paulo Cortez in Portugal at the University of Minho being used as a data mining approach to predict human wine taste preferences. The dataset used for Cortez's study is related to red variants of the Portuguese "Vinho Verde" wine.

0.1.4 A description of the variables used in exploratory data analysis and classification models, so the reader understands how these were measured. Do not include descriptions for variables not used.

i) Fixed acidity (g(tartaric acid)/dm): content of tartaric acid
ii) Volatile acidity (g(acetic acid)/dm3): content of acetic acid
iii) Citric acid (g/dm3): citric acid content
iv) Residual sugar (g/dm3): natural grape sugars leftover in a wine after the alcoholic fermentation finishes
v) Chlorides (g(sodium chloride)/dm3)
vi) Free sulfur dioxide (mg/dm3)
vii) Total sulfur dioxide (mg/dm3)
viii) Density (g/cm3)
ix) pH: pH (1-14 scale)
x) Sulphates (g(potassium sulphate)/dm3)
xi) Alcohol (vol.%): a measurement of the amount of alcohol in the wine
xii) Quality: quality of wine labeled as good or bad

0.2 Section 2: Exploratory Data Analysis

0.2.1 A description of any data cleaning / processing that needed to be performed in order to produce your graphical summaries.

We turned our quality variable into factors by using the factor() method. This ensures that our classification is done correctly. Additionally, we are choosing to treat free and total sulfur dioxide as continuous variables even though they are discrete. We are doing this because each of them has a wide range of variables, so there would be a lot of partitions if we treated them as discrete. Additionally, because there is such a wide range, they will behave similarly to continuous variables. Besides factoring our quality, our dataset already was very clean and did not need any processing to be done before producing our graphical summaries.

0.2.2 Graphical summaries and contextual interpretations

1) Plot 1
    - From the graph it appears as though the strongest relationships are present between fixed acidity and citric acid, fixed acidity and density, fixed acidity and pH. Bad is represented by black dots, while good wine is represented with red dots. In general, better wines are categorized with higher alcohol content, lower densities, and lower volatile acidity.

```
library(readr)
library(ggplot2)
library(reshape2)
library(dplyr)
library(tidyr)
library(ipred)
library(ROCR)
library(MASS)
library(boot)

Data<-read.csv("wine.csv", header=T)
Data$quality<- factor(Data$quality)

set.seed(4630)
```

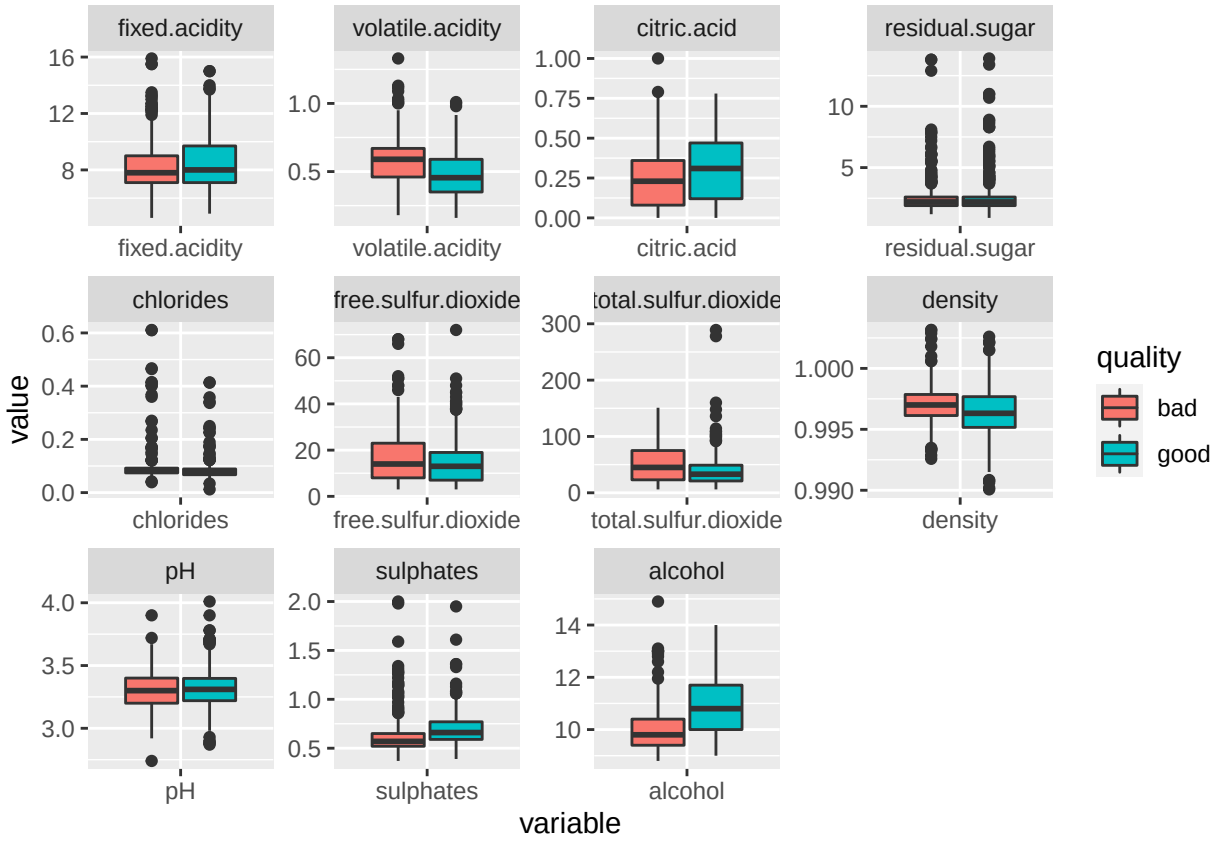
```
sample.data<-sample.int(nrow(Data), floor(.50*nrow(Data)), replace = F)
train<-Data[sample.data, ]
test<-Data[-sample.data, ]

pairs(train[,1:11], col = c(1,2)[train$quality], lower.panel=NULL)
```

2) Plot 2

- The boxplots give us a better understanding of the relationships between each of the predictors and the response variable. We get some background information on what the classification might result in based on a value of the predictor. For example, good wines tend to have higher alcohol content when looking at the median. We can also see that pH remains constant between bad and good quality wines.

```
train2 <- melt(train,id="quality")
ggplot(data = train2, aes(x=variable, y=value)) + geom_boxplot(aes(fill=quality))+
  facet_wrap( ~ variable, scales="free")
```



3) Summary Statistics

- Here are some basic summary statistics. For classification, it is important to note that this dataset is balanced.

```
summary(train)
```

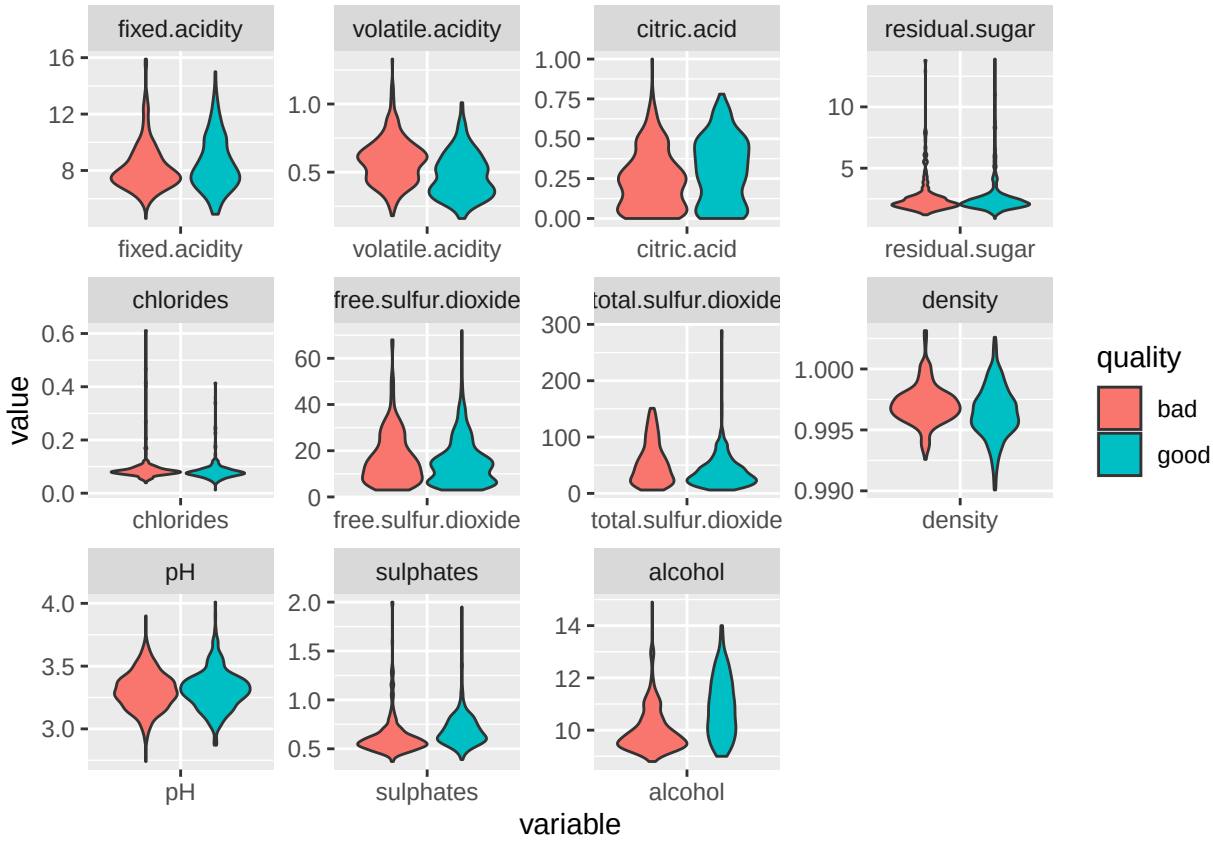
```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 4.600   Min.   :0.1600   Min.   :0.0000   Min.   : 0.900
##  1st Qu.: 7.100   1st Qu.:0.3900   1st Qu.:0.1000   1st Qu.: 1.900
##  Median : 7.900   Median :0.5200   Median :0.2600   Median : 2.200
##  Mean   : 8.342   Mean   :0.5253   Mean   :0.2751   Mean   : 2.615
##  3rd Qu.: 9.250   3rd Qu.:0.6400   3rd Qu.:0.4200   3rd Qu.: 2.600
##  Max.   :15.900   Max.   :1.3300   Max.   :1.0000   Max.   :13.900
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.01200   Min.   : 3.00       Min.   :  6.00       Min.   :0.9901
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 21.00       1st Qu.:0.9956
##  Median :0.07900   Median :14.00       Median : 38.00       Median :0.9968
##  Mean   :0.08899   Mean   :15.77       Mean   : 46.46       Mean   :0.9967
##  3rd Qu.:0.09050   3rd Qu.:21.00       3rd Qu.: 61.00       3rd Qu.:0.9978
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00       Max.   :1.0031
##        pH          sulphates         alcohol      quality
##  Min.   :2.740   Min.   :0.3700   Min.   : 8.80   bad :373
##  1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   good:426
##  Median :3.300   Median :0.6100   Median :10.20
```

```
##  Mean   :3.308   Mean   :0.6575   Mean   :10.47
##  3rd Qu.:3.400   3rd Qu.:0.7200   3rd Qu.:11.20
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90
```

4) Plot 3

- The violin plots give us another way of looking at the distributions of the predictors against the response variable. We can see where majority of observations lie for each predictor vs. response based on how wide the distributions are. In line with the boxplots, bad wines are concentrated around low values. Chlorides and residual sugars have very similar distributions, with majority of observations, for both bad and good wines, concentrating at the lower end.

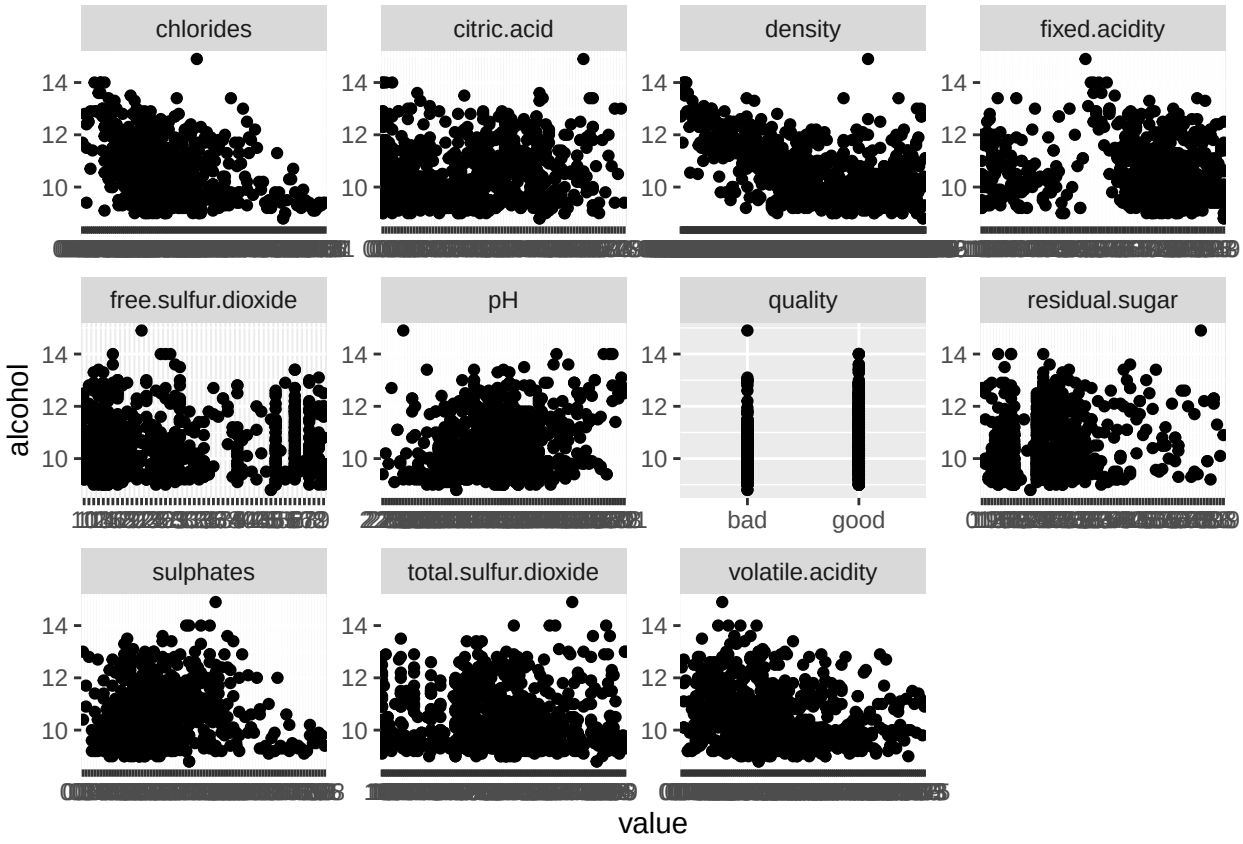
```
ggplot(data = train2, aes(x=variable, y=value)) + geom_violin(aes(fill=quality))+
  facet_wrap( ~ variable, scales="free")
```



5) Plot 4 and Correlations

- From the scatterplots and the correlation list, we can see that density appears to have a strong relationship with alcohol and could be a significant predictor. However, in the scatterplot it appears to be curved slightly which could signify it has an exponential relationship. We can also see that the amount of chlorides may have a relationship with the alcohol % although not as strong as density. There are a number of variables that do not appear to have a strong if at all relationship with the alcohol % including fixed acidity, residual sugar, free sulfur dioxide, and sulphates.

```
train %>%
  gather(-alcohol, key = "var", value = "value") %>%
  ggplot(aes(x=value, y=alcohol)) + geom_point() + facet_wrap(~var, scales = "free")
```



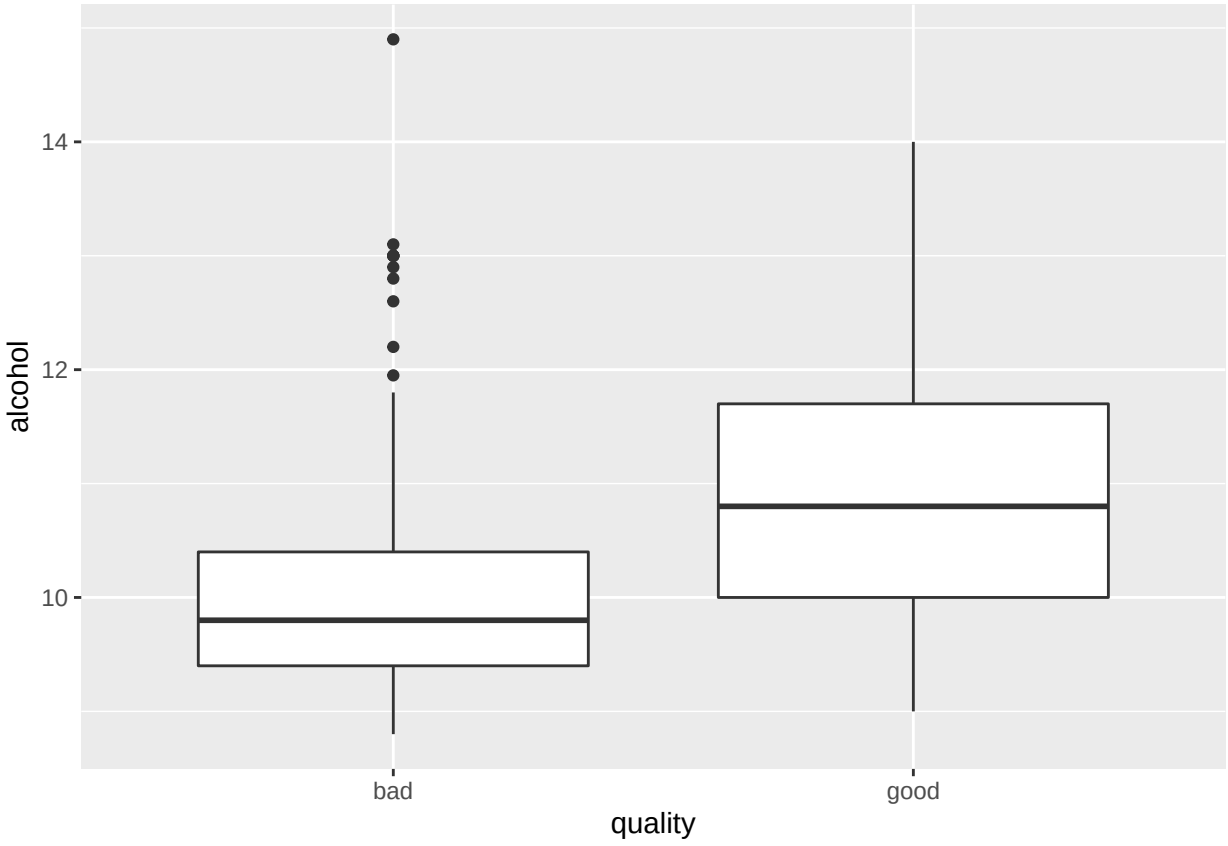

```
cor(train$alcohol, train[,1:10])
```

```
##      fixed.acidity volatile.acidity citric.acid residual.sugar  chlorides
## [1,]   -0.06668969       -0.2158227   0.1091253     0.07130022 -0.2404019
##      free.sulfur.dioxide total.sulfur.dioxide    density        pH  sulphates
## [1,]         -0.07047378           -0.1719911 -0.4989929 0.2230569 0.06384465
```

6) Plot 5

- From the boxplot comparing quality of the alcohol and its % we can see that the median alcohol % for wines categorized as "good" is higher than that for "bad" wines. We can also see that there appear to be a number of outliers for wines that are "bad".

```
ggplot(data = train, aes(x=quality, y=alcohol)) + geom_boxplot()
```



0.3 Section 3.1: Classification

0.3.1 a) A description of any data cleaning / processing that needed to be performed in order to build your models.

We turned our quality variable into factors by using the factor() method. This resulted in 2 levels being "good" and "bad" to categorize quality.

0.3.2 b) Explain which predictors your group decided to include in logistic regression and LDA. You should use the same predictors for both models.

We included all predictors in logistic regression and LDA as none of the predictors were categorical variables.

0.3.3 c) Compare the performance of both logistic regression and LDA, using the following:

i) ROC curve (please see the document for Labs 4a and 4c),

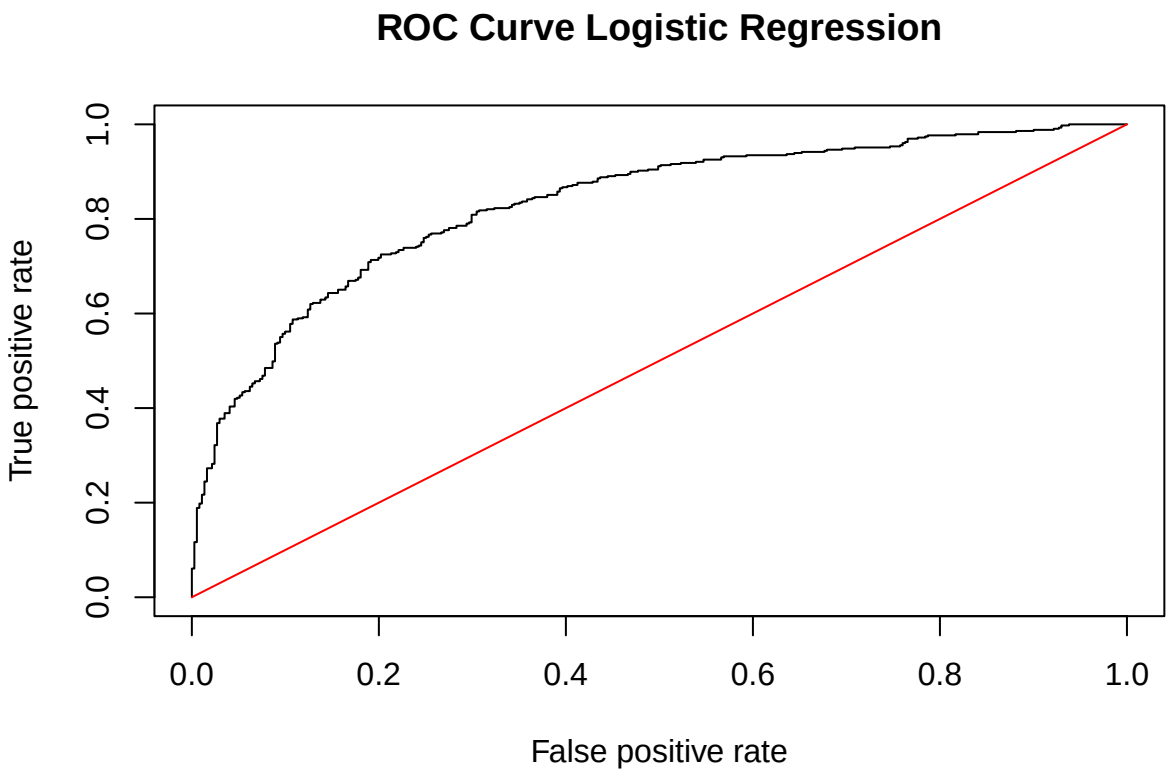
```
contrasts(train$quality)
```

```
##      good
## bad     0
## good    1
```

```r
result_train <- glm(quality~., family="binomial", data=train)
summary(result_train)
```

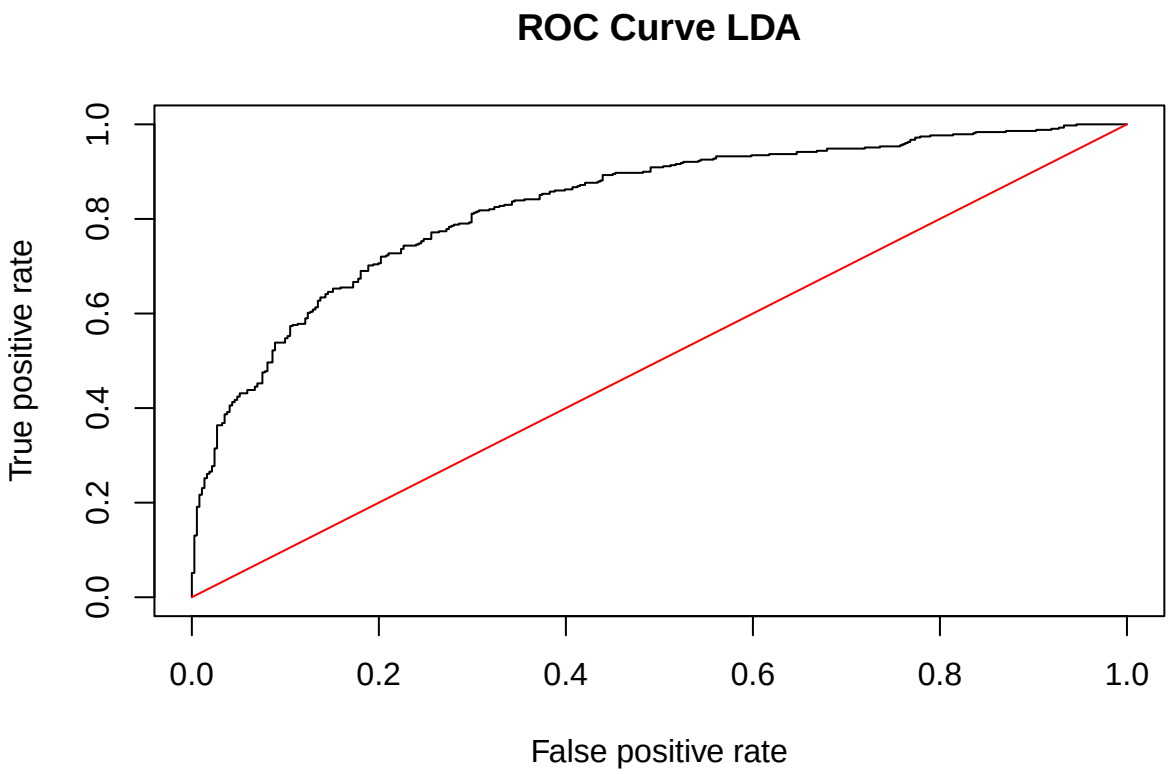
```
##
## Call:
## glm(formula = quality ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0833  -0.9036   0.3434   0.8667   2.1536
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.571e+02  1.087e+02   1.446   0.1482
## fixed.acidity         1.318e-01  1.371e-01   0.961   0.3365
## volatile.acidity     -2.619e+00  6.733e-01  -3.890   0.0001 ***
## citric.acid          -1.282e-01  7.617e-01  -0.168   0.8664
## residual.sugar        1.259e-01  7.056e-02   1.784   0.0744 .
## chlorides            -5.032e+00  2.137e+00  -2.355   0.0185 *
## free.sulfur.dioxide   1.755e-02  1.155e-02   1.519   0.1287
## total.sulfur.dioxide -1.777e-02  3.959e-03  -4.487 7.22e-06 ***
## density              -1.659e+02  1.111e+02  -1.493   0.1355
## pH                    3.684e-01  1.019e+00   0.362   0.7177
## sulphates             3.161e+00  6.439e-01   4.909 9.16e-07 ***
## alcohol               5.825e-01  1.373e-01   4.241 2.22e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1104.13  on 798  degrees of freedom
## Residual deviance:  858.16  on 787  degrees of freedom
## AIC: 882.16
##
## Number of Fisher Scoring iterations: 4
```

```r
preds <- predict(result_train, newdata=test, type="response")
rates <- ROCR::prediction(preds, test$quality)
roc_result <- ROCR::performance(rates, measure="tpr", x.measure="fpr")
plot(roc_result, main="ROC Curve Logistic Regression")
lines(x = c(0,1), y = c(0,1), col="red")
```




```
lda.wine <- MASS::lda(quality~., data=train)
lda.test <- predict(lda.wine, test)

preds2 <- lda.test$posterior[,2]
rates2 <- ROCR::prediction(preds2, test$quality)
roc_result2 <- ROCR::performance(rates2, measure="tpr", x.measure="fpr")
plot(roc_result2, main="ROC Curve LDA")
lines(x = c(0,1), y = c(0,1), col="red")
```



ii) AUC (please see the document for Labs 4a and 4c),

```
auc <- ROCR::performance(rates, measure="auc")
auc@y.values
```

```
## [[1]]
## [1] 0.8310243
```

```
auc2 <- ROCR::performance(rates2, measure = "auc")
auc2@y.values
```

```
## [[1]]
## [1] 0.8308987
```

Logistic regression: 0.8232 LDA: 0.8220

iii) estimated test error rate using k fold cross-validation with k = 5 and k = 10. (please see the document for Lab 5a).

```
result <- glm(quality ~., family="binomial", data=Data)
five.fold <- boot::cv.glm(Data, result, K=5)
five.fold$delta[1]
```

```
## [1] 0.1738929
```

```
ten.fold <- boot::cv.glm(Data, result, K=10)
ten.fold$delta[1]
```

```
## [1] 0.174818
```

```
cv.da <- function(object, newdata)
{
  return(predict(object, newdata = newdata)$class)
}

five.fold2 <- ipred::errorest(quality~., data=Data, model=lda, estimator="cv",
                est.para=control.errorest(k=5), predict=cv.da)$err
five.fold2
```

```
## [1] 0.260788
```

```
ten.fold2 <- ipred::errorest(quality~., data=Data, model=lda, estimator="cv",
                est.para=control.errorest(k=10), predict=cv.da)$err
ten.fold2
```

```
## [1] 0.2557849
```

Logistic Regression: K=5: 0.1752, K=10: 0.1758

LDA: K=5: 0.2564, K=10: 0.2514

iv) actual test error rate by using the test data,

```
confusion.mat <- table(test$quality, preds > 0.5)
error <- (105 + 100) / (281 + 100 + 105 + 314)
error
```

```
## [1] 0.25625
```

```
confusion.mat2 <- table(test$quality, lda.test$class)
error2 <- (107+98) / (283+98+107+312)
error2
```

```
## [1] 0.25625
```

Logistic regression: 0.2563

LDA: 0.2563

v)

```
confusion.mat
```

```
##
##        FALSE TRUE
##   bad    280   91
##   good   107  322
```

```
fpr <- 100 / (281+100)
fnr <- 105 / (105+314)
fpr
```

```
## [1] 0.2624672
```

```
fnr
```

```
## [1] 0.2505967
```

```
confusion.mat2
```

```
##
##        bad good
##   bad  283   88
##   good 110  319
```

```
fpr2 <- 98 / (98+283)
fnr2 <- 107 / (107+312)
fpr2
```

```
## [1] 0.2572178
```

```
fnr2
```

```
## [1] 0.2553699
```

Logistic Regression: FPR: 0.2625, FNR: 0.2506

LDA: FPR: 0.2572, FNR: 0.2554

0.3.4 d) Attempt to improve your logistic regression model (e.g. adding categorical predictor(s), dropping insignificant predictor(s), adding interaction terms). Be sure to include:

i) We are choosing to improve the model after recognizing that a few coefficients are insignificant in the Wald test. We are hoping that by removing some of the more insignificant predictors, our model will be easier to interpret, and will only contain significant predictors. We will drop predictors with a significance level of 0.05.

ii)

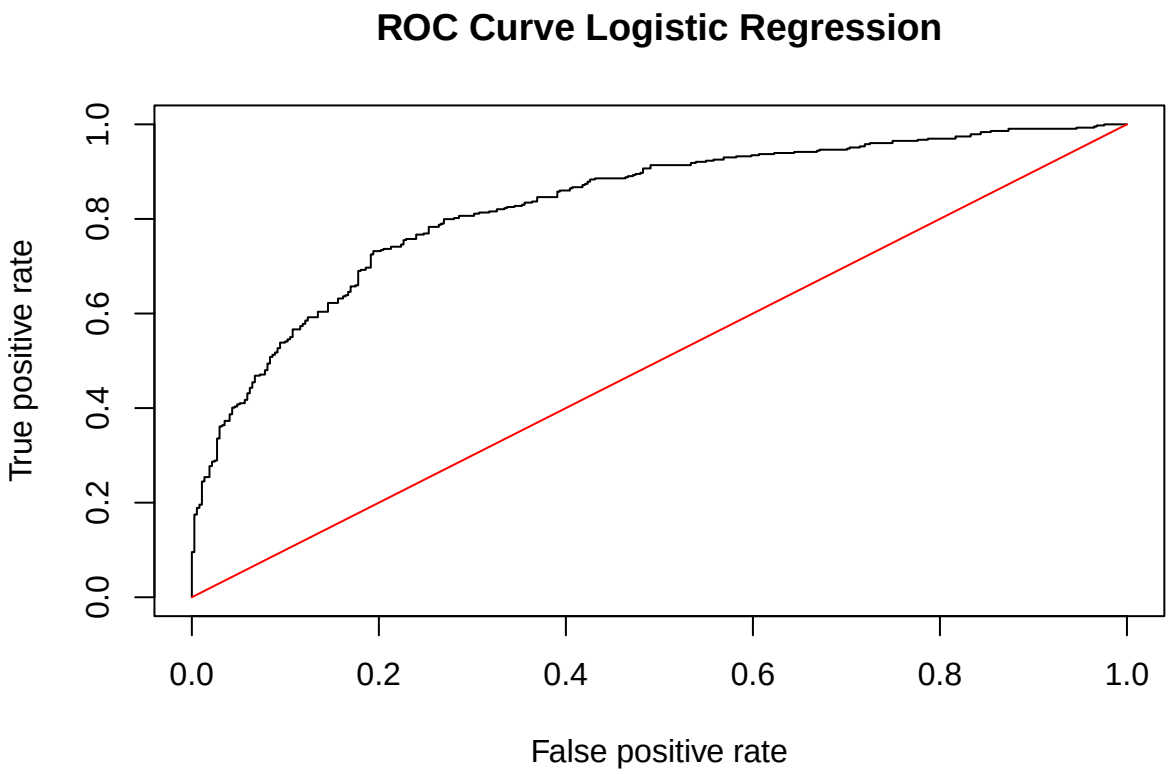
```
summary(result_train)
```

```
##
## Call:
## glm(formula = quality ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -3.0833  -0.9036   0.3434   0.8667   2.1536
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.571e+02  1.087e+02   1.446   0.1482
## fixed.acidity         1.318e-01  1.371e-01   0.961   0.3365
## volatile.acidity     -2.619e+00  6.733e-01  -3.890   0.0001 ***
## citric.acid          -1.282e-01  7.617e-01  -0.168   0.8664
## residual.sugar        1.259e-01  7.056e-02   1.784   0.0744 .
## chlorides            -5.032e+00  2.137e+00  -2.355   0.0185 *
## free.sulfur.dioxide   1.755e-02  1.155e-02   1.519   0.1287
## total.sulfur.dioxide -1.777e-02  3.959e-03  -4.487 7.22e-06 ***
## density              -1.659e+02  1.111e+02  -1.493   0.1355
## pH                    3.684e-01  1.019e+00   0.362   0.7177
## sulphates             3.161e+00  6.439e-01   4.909 9.16e-07 ***
## alcohol               5.825e-01  1.373e-01   4.241 2.22e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1104.13  on 798  degrees of freedom
## Residual deviance:  858.16  on 787  degrees of freedom
## AIC: 882.16
##
## Number of Fisher Scoring iterations: 4
```

```
result_train2 <- glm(quality~volatile.acidity+chlorides+total.sulfur.dioxide+sulphates+alcohol, family=

preds2 <- predict(result_train2, newdata=test, type="response")
rates2 <- ROCR::prediction(preds2, test$quality)
roc_result <- ROCR::performance(rates2, measure="tpr", x.measure="fpr")
plot(roc_result, main="ROC Curve Logistic Regression")
lines(x = c(0,1), y = c(0,1), col="red")
```



iii)

```
auc2_log <- ROCR::performance(rates2, measure="auc")
auc2_log@y.values
```

```
## [[1]]
## [1] 0.8302892
```

AUC: 0.8302892

iv)

```
result2 <- glm(quality ~volatile.acidity+chlorides+total.sulfur.dioxide+sulphates+alcohol, family="binor
five.fold2 <- boot::cv.glm(Data, result2, K=5)
five.fold2$delta[1]
```

```
## [1] 0.1763333
```

```
ten.fold2 <- boot::cv.glm(Data, result2, K=10)
ten.fold2$delta[1]
```

```
## [1] 0.1759326
```

```
confusion.mat3 <- table(test$quality, preds2 > 0.5)
error3 <- (104 + 86) / (285 + 104 + 86 + 325)
error3
```

```
## [1] 0.2375
```

K=5: 0.1758058, K=10: 0.1759179 Actual test error rate: 0.2375

v)

```
confusion.mat3
```

```
##
##        FALSE TRUE
##   bad    285   86
##   good   104  325
```

```
fpr3 <- 86 / (86+285)
fnr3 <- 104 / (104+325)
fpr3
```

```
## [1] 0.2318059
```

```
fnr3
```

```
## [1] 0.2424242
```

FPR: 0.2318, FNR: 0.2424

0.3.5 Compare the results from this model with the models you considered in part c. While your group may consider a few different ways to improve the model, present the improved model which your group is most satisfied with. Therefore, this section should only have R output for two logistic regression models.

From our reduced model, we see that the accuracy was improved, the error rate decreased, and the FPR and FNR also slightly decreased. The AUC however, is ever so slightly smaller than the AUC for the full model, though this difference is almost negligent. The K fold validation expected test error was also higher for the reduced model. Overall, the models have very similar performance statistics, but the reduced model did have better accuracy, FPR, FNR, smaller error, thus we would use the reduced model for classification.

0.3.6 e)

```
summary(result_train)
```

```
##
## Call:
## glm(formula = quality ~ ., family = "binomial", data = train)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0833  -0.9036   0.3434   0.8667   2.1536
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.571e+02  1.087e+02   1.446   0.1482
## fixed.acidity         1.318e-01  1.371e-01   0.961   0.3365
## volatile.acidity     -2.619e+00  6.733e-01  -3.890   0.0001 ***
## citric.acid          -1.282e-01  7.617e-01  -0.168   0.8664
## residual.sugar        1.259e-01  7.056e-02   1.784   0.0744 .
## chlorides            -5.032e+00  2.137e+00  -2.355   0.0185 *
## free.sulfur.dioxide   1.755e-02  1.155e-02   1.519   0.1287
## total.sulfur.dioxide -1.777e-02  3.959e-03  -4.487 7.22e-06 ***
## density              -1.659e+02  1.111e+02  -1.493   0.1355
## pH                    3.684e-01  1.019e+00   0.362   0.7177
## sulphates             3.161e+00  6.439e-01   4.909 9.16e-07 ***
## alcohol               5.825e-01  1.373e-01   4.241 2.22e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1104.13  on 798  degrees of freedom
## Residual deviance:  858.16  on 787  degrees of freedom
## AIC: 882.16
##
## Number of Fisher Scoring iterations: 4
```

```
summary(result_train2)
```

```
##
## Call:
## glm(formula = quality ~ volatile.acidity + chlorides + total.sulfur.dioxide +
##     sulphates + alcohol, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8735  -0.8991   0.3623   0.8792   2.1765
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -6.862381   1.078401  -6.363 1.97e-10 ***
## volatile.acidity      -2.809427   0.521210  -5.390 7.04e-08 ***
## chlorides             -4.893399   1.889679  -2.590  0.00961 **
## total.sulfur.dioxide -0.012786   0.002622  -4.877 1.08e-06 ***
## sulphates              2.829919   0.614857   4.603 4.17e-06 ***
## alcohol                0.737354   0.092911   7.936 2.09e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1104.13  on 798  degrees of freedom
```

```
## Residual deviance:  864.86  on 793  degrees of freedom
## AIC: 876.86
##
## Number of Fisher Scoring iterations: 4
```

In the first model, there are many insignificant predictors. On the other hand, in the reduced model, eliminating the insignificant predictors, all predictors are significant from the Wald test. We were surprised that residual sugar was not more significant, as we thought that the sweetness of a wine was a strong factor in whether somebody liked or disliked a wine.

0.3.7 f)

```
lda.wine
```

```
## Call:
## lda(quality ~ ., data = train)
##
## Prior probabilities of groups:
##       bad      good
## 0.4668335 0.5331665
##
## Group means:
##      fixed.acidity volatile.acidity citric.acid residual.sugar  chlorides
## bad        8.20429        0.5812332   0.2424397       2.609383 0.09484182
## good       8.46338        0.4763146   0.3037089       2.619249 0.08386854
##      free.sulfur.dioxide total.sulfur.dioxide   density       pH sulphates
## bad             16.61126             54.28820 0.9970827 3.304129 0.6211260
## good            15.02934             39.61268 0.9964334 3.311244 0.6893427
##        alcohol
## bad   9.993834
## good 10.892332
##
## Coefficients of linear discriminants:
##                                LD1
## fixed.acidity           0.09269391
## volatile.acidity       -2.20942484
## citric.acid             0.03980897
## residual.sugar          0.09169678
## chlorides              -4.41803920
## free.sulfur.dioxide     0.01580010
## total.sulfur.dioxide   -0.01529985
## density              -117.78478874
## pH                      0.29982147
## sulphates               2.62841200
## alcohol                 0.49638580
```

The large coefficients for density, chlorides,sulphates, and volatile.acidity demonstrate that these predictors all have a larger influence on the classification, especially density. Additionally, the negative signs of volatile.acidity, chlorides, and density show that lower values of each of these variables, and higher values of the rest of the variables mean a wine is more likely to be classified as a good wine. We were surprised to see that density had such a large factor in a wine's classification.

0.4 Section 4: Conclusions

0.4.1 a) A discussion on how your model(s) answers your group's question of interest.

- To restate the question of interest, we are interested in determining whether or not a wine can be classified as good or bad based on certain predictors in the data set. The usefulness of an analysis like this can be seen with wine makers. By looking at the results of our model, wine makers can modify their process to ensure the highest quality wine. We believe that our model does fit this goal for a few reasons. One, looking at the reduced model, we observe low levels of FPR, FNR, and error rate. This is a good sign, as it proves our model is reliable in that our model does not make many errors when classifying a wine. Second, because we are using a reduced model, our results are much easier to interpret. In theory, this would make it easier to explain the results to someone who does not come from a statistics background. Overall, our model does a good job predicting whether or not an observation is a good or bad wine.

0.4.2 b) Any interesting insights gained about the data.

- We took note of a few interesting insights about the data. One, we were extremely surprised by the LDA analysis of the data. The output gave us an extremely large negative number for density (-117.78), but density was not included in our reduced model. This could be a good thing as we don't want one predictor dominating the interpretation of the results. Something else we found interested was that our AUC value dropped a little with our reduced model. This is most likely due to random error, but it was still interesting to think through why with all other summary statistics dropped (FNR, FPR, error), AUC still slightly decreased.

0.4.3 c) Any challenges your group faced.

- Our dataset is clean, so we had no problems with initial data cleaning/manipulation. Our biggest challenge, was deciding whether to move forward with the reduced model, but after seeing the results of both the models, it was clear that the reduced model gave us better results.