Statistical Modeling - Final Project

Deshpande, Vedant Sunil

6/18/2021

### Load required packages

library(dplyr)  
library(ggplot2)  
library(patchwork)  
library(car)  
library(corrplot)  
library(ROCR)  
library(GGally)  
library(VGAM)  
library(faraway)

# Part A. Background and Data Exploration (EDA):

## A.4. Import the red wine data set into your R system.

# Load dataset  
red\_wine <- read.csv("winequality-red.csv", sep = ";",header = T)  
head(red\_wine)

## fixed.acidity volatile.acidity citric.acid residual.sugar chlorides  
## 1 7.4 0.70 0.00 1.9 0.076  
## 2 7.8 0.88 0.00 2.6 0.098  
## 3 7.8 0.76 0.04 2.3 0.092  
## 4 11.2 0.28 0.56 1.9 0.075  
## 5 7.4 0.70 0.00 1.9 0.076  
## 6 7.4 0.66 0.00 1.8 0.075  
## free.sulfur.dioxide total.sulfur.dioxide density pH sulphates alcohol  
## 1 11 34 0.9978 3.51 0.56 9.4  
## 2 25 67 0.9968 3.20 0.68 9.8  
## 3 15 54 0.9970 3.26 0.65 9.8  
## 4 17 60 0.9980 3.16 0.58 9.8  
## 5 11 34 0.9978 3.51 0.56 9.4  
## 6 13 40 0.9978 3.51 0.56 9.4  
## quality  
## 1 5  
## 2 5  
## 3 5  
## 4 6  
## 5 5  
## 6 5

## 

## A.5. In your report, write a section to introduce the data set and give basic statistics.

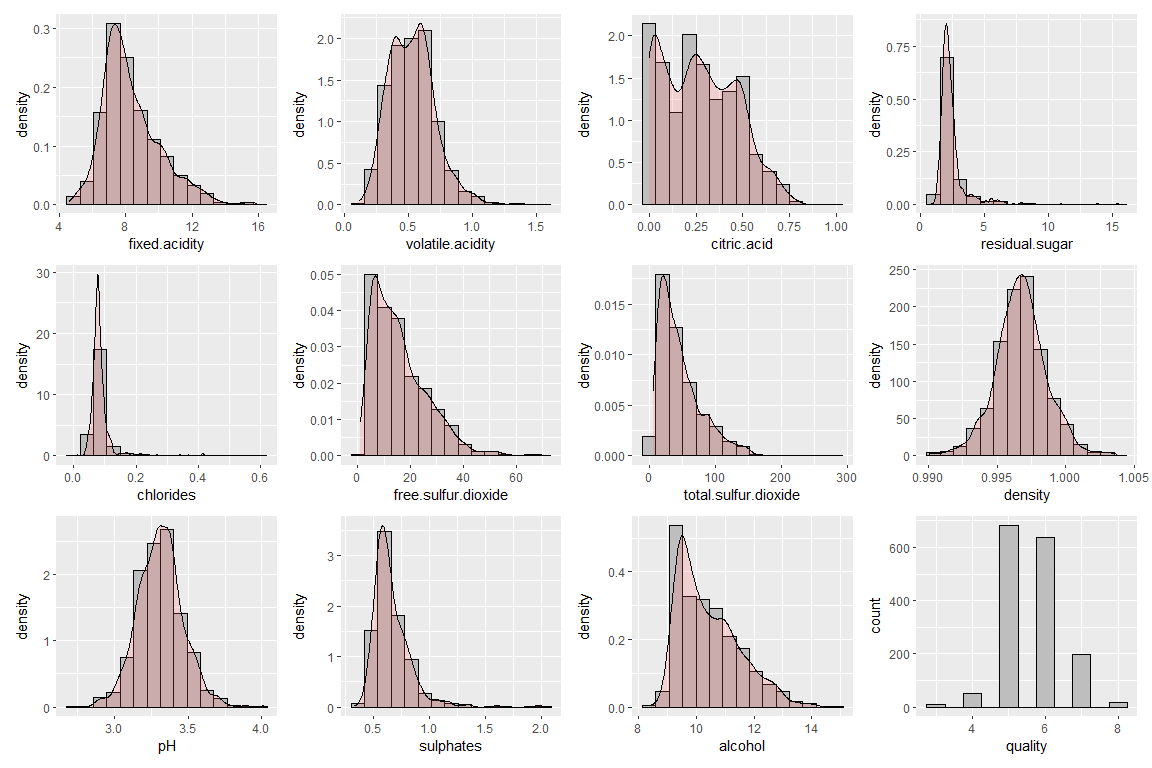
# Column structure and basic summary statistics  
str(red\_wine)

## 'data.frame': 1599 obs. of 12 variables:  
## $ fixed.acidity : num 7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...  
## $ volatile.acidity : num 0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...  
## $ citric.acid : num 0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...  
## $ residual.sugar : num 1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...  
## $ chlorides : num 0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...  
## $ free.sulfur.dioxide : num 11 25 15 17 11 13 15 15 9 17 ...  
## $ total.sulfur.dioxide: num 34 67 54 60 34 40 59 21 18 102 ...  
## $ density : num 0.998 0.997 0.997 0.998 0.998 ...  
## $ pH : num 3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...  
## $ sulphates : num 0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...  
## $ alcohol : num 9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...  
## $ quality : int 5 5 5 6 5 5 5 7 7 5 ...

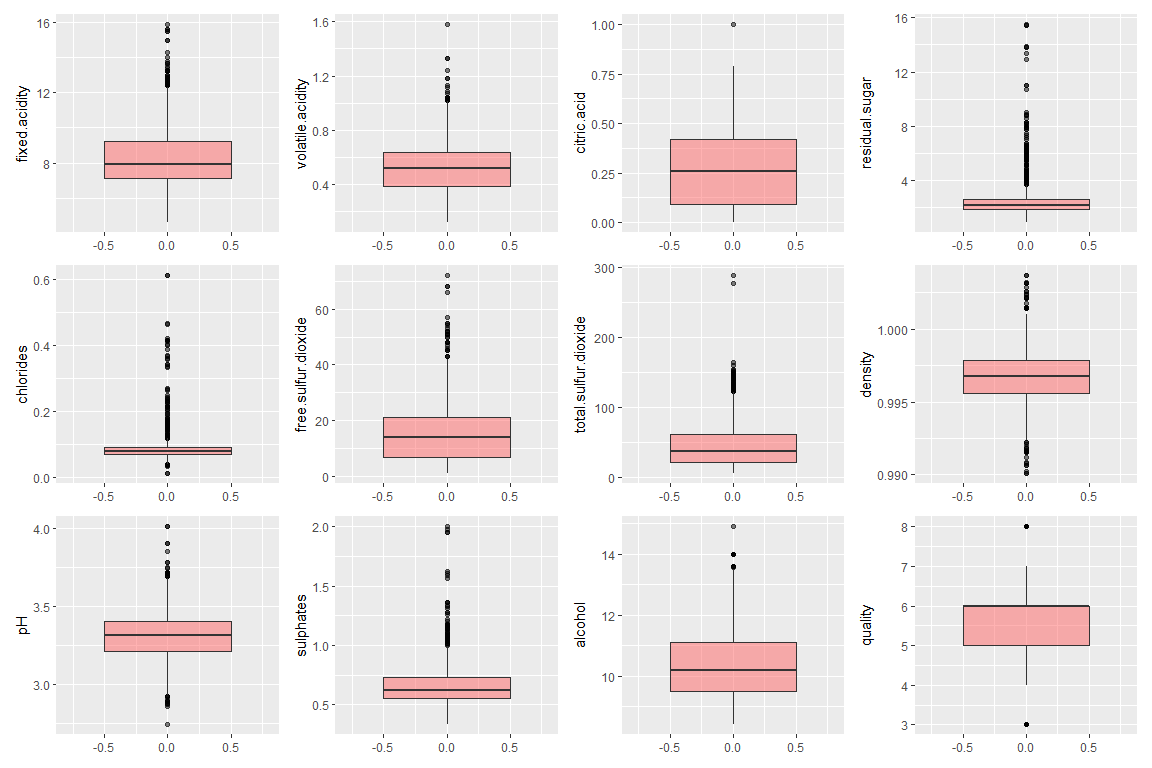
summary(red\_wine)

## fixed.acidity volatile.acidity citric.acid residual.sugar   
## Min. : 4.60 Min. :0.1200 Min. :0.000 Min. : 0.900   
## 1st Qu.: 7.10 1st Qu.:0.3900 1st Qu.:0.090 1st Qu.: 1.900   
## Median : 7.90 Median :0.5200 Median :0.260 Median : 2.200   
## Mean : 8.32 Mean :0.5278 Mean :0.271 Mean : 2.539   
## 3rd Qu.: 9.20 3rd Qu.:0.6400 3rd Qu.:0.420 3rd Qu.: 2.600   
## Max. :15.90 Max. :1.5800 Max. :1.000 Max. :15.500   
## chlorides free.sulfur.dioxide total.sulfur.dioxide density   
## Min. :0.01200 Min. : 1.00 Min. : 6.00 Min. :0.9901   
## 1st Qu.:0.07000 1st Qu.: 7.00 1st Qu.: 22.00 1st Qu.:0.9956   
## Median :0.07900 Median :14.00 Median : 38.00 Median :0.9968   
## Mean :0.08747 Mean :15.87 Mean : 46.47 Mean :0.9967   
## 3rd Qu.:0.09000 3rd Qu.:21.00 3rd Qu.: 62.00 3rd Qu.:0.9978   
## Max. :0.61100 Max. :72.00 Max. :289.00 Max. :1.0037   
## pH sulphates alcohol quality   
## Min. :2.740 Min. :0.3300 Min. : 8.40 Min. :3.000   
## 1st Qu.:3.210 1st Qu.:0.5500 1st Qu.: 9.50 1st Qu.:5.000   
## Median :3.310 Median :0.6200 Median :10.20 Median :6.000   
## Mean :3.311 Mean :0.6581 Mean :10.42 Mean :5.636   
## 3rd Qu.:3.400 3rd Qu.:0.7300 3rd Qu.:11.10 3rd Qu.:6.000   
## Max. :4.010 Max. :2.0000 Max. :14.90 Max. :8.000

# Histogram and density plots  
par(mfrow = c(4,3))  
h1 <- ggplot(red\_wine, aes(x=fixed.acidity)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey',bins = 15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h2 <- ggplot(red\_wine, aes(x=volatile.acidity)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey',bins = 15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h3 <- ggplot(red\_wine, aes(x=citric.acid)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey', bins = 15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h4 <- ggplot(red\_wine, aes(x=residual.sugar)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey', bins = 15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h5 <- ggplot(red\_wine, aes(x=chlorides)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey', bins = 15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h6 <- ggplot(red\_wine, aes(x=free.sulfur.dioxide)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey', bins =15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h7 <- ggplot(red\_wine, aes(x=total.sulfur.dioxide)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey', bins = 15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h8 <- ggplot(red\_wine, aes(x=density)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey', bins = 15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h9 <- ggplot(red\_wine, aes(x=pH)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey', bins = 15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h10 <- ggplot(red\_wine, aes(x=sulphates)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey', bins = 15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h11 <- ggplot(red\_wine, aes(x=alcohol)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = 'grey', bins = 15) +  
 geom\_density(alpha = 0.2, fill = "#FF6666")  
h12 <- ggplot(red\_wine, aes(x=quality)) +   
 geom\_histogram(aes(), color='black', fill = 'grey', bins = 6, binwidth = 0.5)  
  
h1 + h2 + h3 + h4 + h5 + h6 + h7 + h8 + h9 + h10 + h11 + h12



# Boxplots to determine outliers  
par(mfrow=c(4,3))  
b1 <- ggplot(red\_wine, aes(y=fixed.acidity)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b2 <- ggplot(red\_wine, aes(y=volatile.acidity)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b3 <- ggplot(red\_wine, aes(y=citric.acid)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b4 <- ggplot(red\_wine, aes(y=residual.sugar)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b5 <- ggplot(red\_wine, aes(y=chlorides)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b6 <- ggplot(red\_wine, aes(y=free.sulfur.dioxide)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b7 <- ggplot(red\_wine, aes(y=total.sulfur.dioxide)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b8 <- ggplot(red\_wine, aes(y=density)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b9 <- ggplot(red\_wine, aes(y=pH)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b10 <- ggplot(red\_wine, aes(y=sulphates)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b11 <- ggplot(red\_wine, aes(y=alcohol)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b12 <- ggplot(red\_wine, aes(y=quality)) +   
 geom\_boxplot(fill='#FF6666',outlier.colour = 'black',width=1,alpha=0.5) + xlim(-0.8,0.8)  
b1 + b2 + b3 + b4 + b5 + b6 + b7 + b8 + b9 + b10 + b11 + b12



# Part B. Visualization and initial models for a binary response

## B.1. Create a new binary variable “excellent” following this rule:

1. excellent = 1, if the quality rating >= 7
2. excellent = 0, otherwise

red\_wine$excellent <- as.factor(ifelse(red\_wine$quality>6,1,0))  
red\_wine <- subset(red\_wine,select = -c(quality))

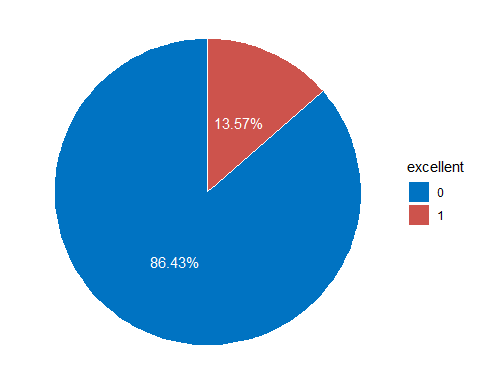
## 

## B.2. Calculate the proportion of excellent red wine (i.e., excellent == 1). Make a pie chart and write down your interpretation of the proportion.

pie\_l <- count(red\_wine,excellent)  
pie\_l <- pie\_l %>%  
 arrange(desc(excellent)) %>%  
 mutate(lab.ypos = cumsum(n) - 0.5\*n)  
pie\_l$p <- round(pie\_l$n/sum(pie\_l$n),6)  
  
Excellent <- pie\_l$excellent  
Count <- pie\_l$n  
Proportion <- pie\_l$p  
proportion <- data.frame(Excellent,Count,Proportion)  
proportion

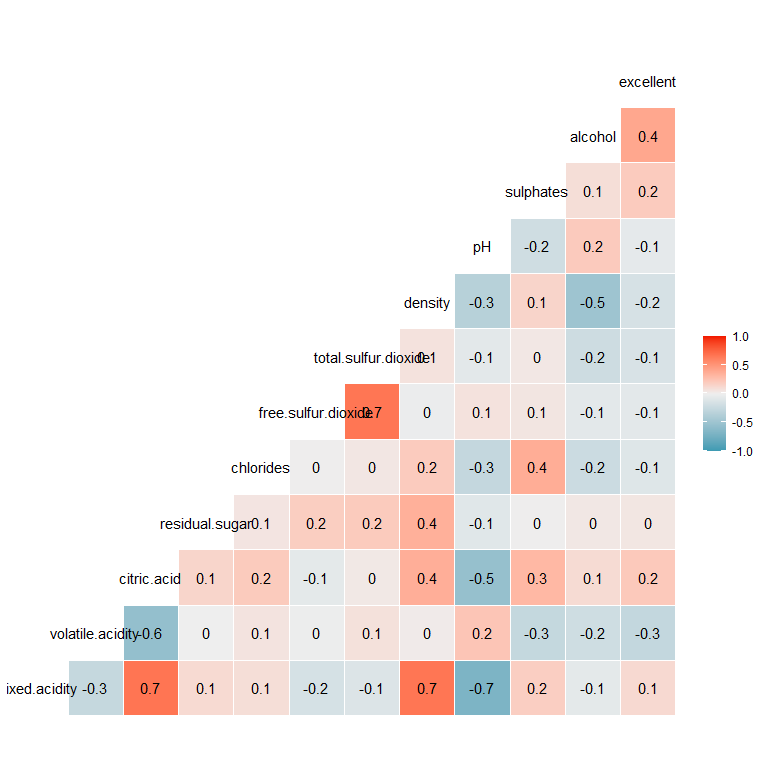
## Excellent Count Proportion  
## 1 1 217 0.13571  
## 2 0 1382 0.86429

mycols <- c("#0073C2FF", "#CD534CFF")  
  
ggplot(pie\_l, aes(x=103.5, y=n, fill=excellent)) +  
 geom\_bar(stat="identity", color='white') +  
 coord\_polar("y", start=0) + geom\_text(aes(y = lab.ypos, label = scales::percent(p,accuracy=0.01)), color = "white") + scale\_fill\_manual(values = mycols) + theme\_void()



**B.3. Make use of various visualization tools to inspect the association between the binary response “excellent” and each of the explanatory variables/predictors. In your report, include those for which you do see an association.**

# Corrplot  
red\_wine\_adj <- red\_wine  
red\_wine\_adj$excellent <- as.numeric(red\_wine\_adj$excellent)  
ggcorr(red\_wine\_adj, label = TRUE)



We see high correlation for alcohol, volatile acidity, sulphates, citric acidity, density.

par(mfrow=c(3,2))  
s1 <- ggplot(red\_wine, aes(x=excellent, y=alcohol)) +  
 geom\_point(size=1,color='#990521')  
s2 <- ggplot(red\_wine, aes(x=excellent, y=volatile.acidity)) +  
 geom\_point(size=1,color='#990521')  
s3 <- ggplot(red\_wine, aes(x=excellent, y=sulphates)) +  
 geom\_point(size=1,color='#990521')  
s4 <- ggplot(red\_wine, aes(x=excellent, y=citric.acid)) +  
 geom\_point(size=1,color='#990521')  
s5 <- ggplot(red\_wine, aes(x=excellent, y=density)) +  
 geom\_point(size=1,color='#990521')  
  
s1 + s2 + s3 + s4 + s5



## B.4. Use “excellent” as the response, fit a linear model. What are your findings? Which variables are significant? Among those significant ones, which has a positive association with the quality? Summarize your findings in your report

# Fit Linear Regression model  
model\_lm <- lm(as.numeric(excellent)~.,red\_wine)  
summary(model\_lm)

##   
## Call:  
## lm(formula = as.numeric(excellent) ~ ., data = red\_wine)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.92396 -0.17446 -0.04220 0.05006 0.99838   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.643e+01 9.792e+00 3.721 0.000206 \*\*\*  
## fixed.acidity 3.400e-02 1.199e-02 2.836 0.004620 \*\*   
## volatile.acidity -1.783e-01 5.595e-02 -3.187 0.001468 \*\*   
## citric.acid 8.683e-02 6.799e-02 1.277 0.201774   
## residual.sugar 2.520e-02 6.931e-03 3.635 0.000286 \*\*\*  
## chlorides -6.556e-01 1.937e-01 -3.385 0.000730 \*\*\*  
## free.sulfur.dioxide -5.526e-04 1.003e-03 -0.551 0.581817   
## total.sulfur.dioxide -6.658e-04 3.367e-04 -1.977 0.048159 \*   
## density -3.668e+01 9.994e+00 -3.670 0.000251 \*\*\*  
## pH 1.725e-02 8.852e-02 0.195 0.845516   
## sulphates 3.515e-01 5.282e-02 6.655 3.89e-11 \*\*\*  
## alcohol 7.618e-02 1.224e-02 6.226 6.11e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2994 on 1587 degrees of freedom  
## Multiple R-squared: 0.2416, Adjusted R-squared: 0.2363   
## F-statistic: 45.96 on 11 and 1587 DF, p-value: < 2.2e-16

All the variables are statistically significant with a p-value < 0.05, except for citric.acid, free.sulfur.dioxide and ph. Following variables have a positive coefficient, implying that increase in these variables will improve the chances of wine quality being excellent. fixed.acidity, residual.sugar, sulphates and alcohol Following variables have a negative coefficient, implying that increase in these variables will decrease the chances of wine quality being excellent. volatile.acidity, chlorides, total.sulfur.dioxide and density.

## 

## B.5. Repeat Item 4 but use a logistic regression model. Are your new findings the same as those in Item 4? Is there any difference? Give a summary in your report

model\_lg <- glm(excellent~.,data=red\_wine,family='binomial')  
summary(model\_lg)

##   
## Call:  
## glm(formula = excellent ~ ., family = "binomial", data = red\_wine)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.9878 -0.4351 -0.2207 -0.1222 2.9869   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.428e+02 1.081e+02 2.247 0.024660 \*   
## fixed.acidity 2.750e-01 1.253e-01 2.195 0.028183 \*   
## volatile.acidity -2.581e+00 7.843e-01 -3.291 0.000999 \*\*\*  
## citric.acid 5.678e-01 8.385e-01 0.677 0.498313   
## residual.sugar 2.395e-01 7.373e-02 3.248 0.001163 \*\*   
## chlorides -8.816e+00 3.365e+00 -2.620 0.008788 \*\*   
## free.sulfur.dioxide 1.082e-02 1.223e-02 0.884 0.376469   
## total.sulfur.dioxide -1.653e-02 4.894e-03 -3.378 0.000731 \*\*\*  
## density -2.578e+02 1.104e+02 -2.335 0.019536 \*   
## pH 2.242e-01 9.984e-01 0.225 0.822327   
## sulphates 3.750e+00 5.416e-01 6.924 4.39e-12 \*\*\*  
## alcohol 7.533e-01 1.316e-01 5.724 1.04e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1269.92 on 1598 degrees of freedom  
## Residual deviance: 870.86 on 1587 degrees of freedom  
## AIC: 894.86  
##   
## Number of Fisher Scoring iterations: 6

The results obtained using Logistic Regression Model are similar to Linear Regression Model in terms of significance of variables. citric.acid, free.sulfur.dioxide and pH continue to be the variables with a p-value > 0.05. The sign is different for free.sulfur.dioxide (negative for Linear vs positive for Logistic). But since the coefficient is statistically insignificant, this discrepancy should not matter. Hence, we can conclude that both models provide similar results in terms of variable significance and positive/negative association.

# 

# Part C. Variable selection, interpretation, and prediction for a logistic model

## C.1 Start with the logistic regression model you have built in Part B. Are you able to “trim” it to achieve a smaller set of predictors? In this variable selection process, you may need to consider these techniques and issues: AIC/BIC, Chi-squared test, collinearity, and the domain knowledge for each variable (references: Lecture 3, pages 8-29). Present the methods you use and the final model in your report

model\_null <- glm(excellent ~ 1,family='binomial',red\_wine)

# Stepwise selecting using AIC  
model\_aic <- step(model\_null,scope=list(lower=model\_null,upper=model\_lg), direction='both',trace=F)  
summary(model\_aic)

##   
## Call:  
## glm(formula = excellent ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar +   
## density, family = "binomial", data = red\_wine)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.0158 -0.4314 -0.2220 -0.1255 2.9883   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.268e+02 9.163e+01 2.475 0.013336 \*   
## alcohol 7.823e-01 1.120e-01 6.983 2.88e-12 \*\*\*  
## volatile.acidity -2.913e+00 6.467e-01 -4.504 6.66e-06 \*\*\*  
## sulphates 3.699e+00 5.287e-01 6.997 2.62e-12 \*\*\*  
## total.sulfur.dioxide -1.360e-02 3.447e-03 -3.946 7.95e-05 \*\*\*  
## chlorides -8.441e+00 3.259e+00 -2.590 0.009593 \*\*   
## fixed.acidity 2.812e-01 8.029e-02 3.502 0.000462 \*\*\*  
## residual.sugar 2.328e-01 7.009e-02 3.322 0.000893 \*\*\*  
## density -2.409e+02 9.202e+01 -2.618 0.008835 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1269.92 on 1598 degrees of freedom  
## Residual deviance: 872.08 on 1590 degrees of freedom  
## AIC: 890.08  
##   
## Number of Fisher Scoring iterations: 6

The final model recommended by stepwise selection using AIC values is as follows: excellent ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar + density

pH, citric.acid and free.sulfur.dioxide have been eliminated.

# Stepwise selection using BIC  
model\_bic <- step(model\_null,scope=list(lower=model\_null,upper=model\_lg), direction='both', k=log(nrow(red\_wine)), trace=FALSE)  
summary(model\_bic)

##   
## Call:  
## glm(formula = excellent ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide + chlorides + fixed.acidity, family = "binomial",   
## data = red\_wine)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6332 -0.4386 -0.2313 -0.1270 3.0411   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -13.283277 1.316313 -10.091 < 2e-16 \*\*\*  
## alcohol 0.988930 0.086710 11.405 < 2e-16 \*\*\*  
## volatile.acidity -3.345588 0.619146 -5.404 6.53e-08 \*\*\*  
## sulphates 3.323236 0.520367 6.386 1.70e-10 \*\*\*  
## total.sulfur.dioxide -0.012355 0.003596 -3.436 0.000591 \*\*\*  
## chlorides -8.802351 3.413297 -2.579 0.009913 \*\*   
## fixed.acidity 0.133235 0.047596 2.799 0.005121 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1269.92 on 1598 degrees of freedom  
## Residual deviance: 883.17 on 1592 degrees of freedom  
## AIC: 897.17  
##   
## Number of Fisher Scoring iterations: 6

The final model recommended by stepwise selection using BIC values is as follows: excellent ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity

Stepwise selection using BIC always provided a more parsimonious model as it penalizes more in case of higher number of covariates. Thus, less number of variables are being recommended in this case as compared to stepwise selection using AIC.

# Model selection based on p-values  
drop1(model\_lg,test="Chi")

## Single term deletions  
##   
## Model:  
## excellent ~ fixed.acidity + volatile.acidity + citric.acid +   
## residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +   
## density + pH + sulphates + alcohol  
## Df Deviance AIC LRT Pr(>Chi)   
## <none> 870.86 894.86   
## fixed.acidity 1 875.67 897.67 4.807 0.0283520 \*   
## volatile.acidity 1 882.52 904.52 11.660 0.0006387 \*\*\*  
## citric.acid 1 871.32 893.32 0.457 0.4991231   
## residual.sugar 1 880.02 902.02 9.153 0.0024826 \*\*   
## chlorides 1 880.85 902.85 9.983 0.0015801 \*\*   
## free.sulfur.dioxide 1 871.64 893.64 0.780 0.3771127   
## total.sulfur.dioxide 1 884.49 906.49 13.625 0.0002231 \*\*\*  
## density 1 876.34 898.34 5.475 0.0192895 \*   
## pH 1 870.91 892.91 0.050 0.8224870   
## sulphates 1 915.26 937.26 44.391 2.689e-11 \*\*\*  
## alcohol 1 904.51 926.51 33.643 6.620e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Model given by stepwise selection using AIC is same as the one above, if we remove variables with p-value > 0.05.

Thus, we will be going ahead with the following model for further analysis. excellent ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar + density

# Final model  
model\_final <- glm(excellent ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar + density,family='binomial',red\_wine)  
summary(model\_final)

##   
## Call:  
## glm(formula = excellent ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar +   
## density, family = "binomial", data = red\_wine)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.0158 -0.4314 -0.2220 -0.1255 2.9883   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.268e+02 9.163e+01 2.475 0.013336 \*   
## alcohol 7.823e-01 1.120e-01 6.983 2.88e-12 \*\*\*  
## volatile.acidity -2.913e+00 6.467e-01 -4.504 6.66e-06 \*\*\*  
## sulphates 3.699e+00 5.287e-01 6.997 2.62e-12 \*\*\*  
## total.sulfur.dioxide -1.360e-02 3.447e-03 -3.946 7.95e-05 \*\*\*  
## chlorides -8.441e+00 3.259e+00 -2.590 0.009593 \*\*   
## fixed.acidity 2.812e-01 8.029e-02 3.502 0.000462 \*\*\*  
## residual.sugar 2.328e-01 7.009e-02 3.322 0.000893 \*\*\*  
## density -2.409e+02 9.202e+01 -2.618 0.008835 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1269.92 on 1598 degrees of freedom  
## Residual deviance: 872.08 on 1590 degrees of freedom  
## AIC: 890.08  
##   
## Number of Fisher Scoring iterations: 6

# Checking collinearity in the final model  
red\_wine\_adj <- red\_wine  
red\_wine\_adj$excellent <- as.numeric(red\_wine\_adj$excellent)  
ggcorr(red\_wine\_adj, label = TRUE)



We see high collinearity between the following pairs: 1. fixed.acidity - citric.acid 2. fixed.acidity - density 3. free.sulfur.dioxide - total.sulfur.dioxide

Since citric.acid and free.sulfur.dioxide are not part of final mode, #1 and #3 is not an issue. For #2, we will now conduct a more sophisticated test to identify if this high collinearity is a concern.

# Check multicollinearity  
car::vif(model\_final)

## alcohol volatile.acidity sulphates   
## 1.849496 1.209405 1.261923   
## total.sulfur.dioxide chlorides fixed.acidity   
## 1.187110 1.157832 3.515531   
## residual.sugar density   
## 1.526325 5.541689

As we can see, the VIF (variance inflation factor) is < 10 for all variables, the model does not possess any multi-collinearity issues.

## 

## C.2 Quantify and explain the influence of “alcohol”, “residual.sugar”, and “pH” on the wine quality, if they are in your final model.

summary(model\_final)

##   
## Call:  
## glm(formula = excellent ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar +   
## density, family = "binomial", data = red\_wine)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.0158 -0.4314 -0.2220 -0.1255 2.9883   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.268e+02 9.163e+01 2.475 0.013336 \*   
## alcohol 7.823e-01 1.120e-01 6.983 2.88e-12 \*\*\*  
## volatile.acidity -2.913e+00 6.467e-01 -4.504 6.66e-06 \*\*\*  
## sulphates 3.699e+00 5.287e-01 6.997 2.62e-12 \*\*\*  
## total.sulfur.dioxide -1.360e-02 3.447e-03 -3.946 7.95e-05 \*\*\*  
## chlorides -8.441e+00 3.259e+00 -2.590 0.009593 \*\*   
## fixed.acidity 2.812e-01 8.029e-02 3.502 0.000462 \*\*\*  
## residual.sugar 2.328e-01 7.009e-02 3.322 0.000893 \*\*\*  
## density -2.409e+02 9.202e+01 -2.618 0.008835 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1269.92 on 1598 degrees of freedom  
## Residual deviance: 872.08 on 1590 degrees of freedom  
## AIC: 890.08  
##   
## Number of Fisher Scoring iterations: 6

pH is not part for the final model, but alcohol and residual.sugar are. For every unit increase in alcohol, the log odds of wine being excellent increases by 0.78. For every unit increase in residual.sugar, the log odds of wine being excellent increases by 0.23.

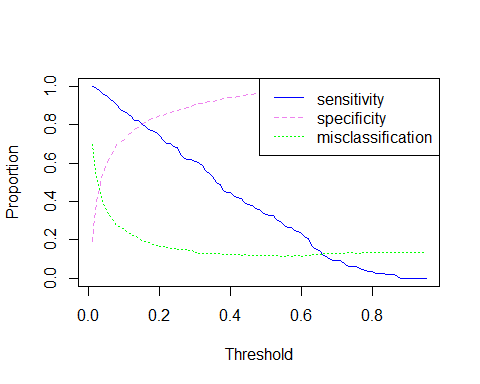
Log odds can be defined as log(p/(1-p)) where p is the probability of the event happening (wine being excellent in our case).

## C.3 If we use the probability 0.5 as the threshold to determine whether or not a bottle of wine is “excellent”, produce a 2 by 2 similar to Slide 31 in Lecture 3. For this particular classification method, calculate its specificity and sensitivity. Should we move the threshold up or down? Please give your comments and insights

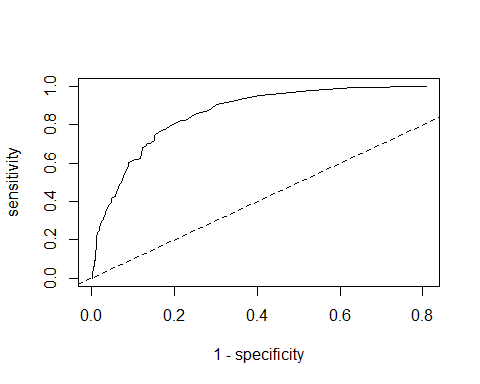
# Predict probabilities  
predprob<-predict(model\_final,type="response")  
  
# Outcome using 0.5 as cutoff probability  
predout<-ifelse(predprob<0.5,0,1)  
  
mtx <- data.frame(red\_wine,predprob,predout)  
conf\_mtx <- xtabs(~excellent+predout,mtx)  
conf\_mtx

## predout  
## excellent 0 1  
## 0 1336 46  
## 1 145 72

# Varying threshold  
thresh<-seq(0.01,0.95,0.01)  
sensitivity<-specificity<-misc<-rep(NA,length(thresh))  
for(j in seq(along=thresh)){  
pp<-ifelse(mtx$predprob<thresh[j],0,1)  
  
xx<-xtabs(~excellent+pp,mtx)  
specificity[j]<-xx[1,1]/(xx[1,1]+xx[1,2])  
sensitivity[j]<-xx[2,2]/(xx[2,1]+xx[2,2])  
misc[j]=(xx[1,2]+xx[2,1])/sum(xx)  
  
}  
matplot(thresh,cbind(sensitivity,specificity,misc),type="l", xlab ="Threshold", ylab ="Proportion",lty =1:3,col=c("blue","violet","green"))  
legend("topright",legend = c("sensitivity","specificity","misclassification"),col = c("blue","violet","green"), lty=c(1,2,3))

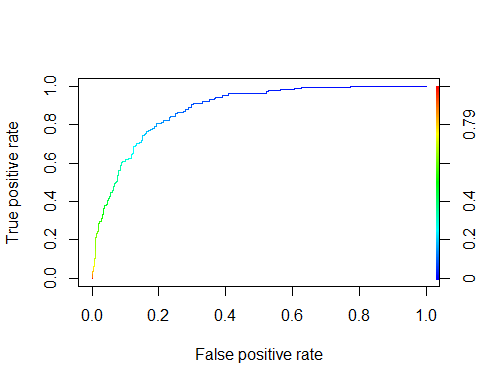


plot(1-specificity,sensitivity,type="l");abline(0,1,lty=2)



As cut-off probability increases, the misclassification rate decreases upto a certain point (~0.5) and then flattens out. Hence, we can continue using 0.5 as cut-off probability. ## C.4 Produce the ROC curve for your final model and calculate its AUC (area under the curve). You may use one of the R packages described in this webpage, or you may use other tools as well

# ROC Curve  
pred\_lg <- prediction(predprob, red\_wine$excellent)  
perf <- performance(pred\_lg, "tpr", "fpr")  
plot(perf, colorize=TRUE)



# AUC  
unlist(slot(performance(pred\_lg, "auc"), "y.values"))

## [1] 0.881985

An AUC value of 0.88 implies that the model does a good job of predicting whether a particular wine is excellent or not.

## 

## C.5 Predict the chance that the 1st bottle (the 1st row in the data set) is excellent and give its 95% confidence interval. Do the same for the 268th bottle (the 268th row in the data set). Compare your prediction with the observed quality ratings.

# Create sample for 1st and 268th rows  
sample\_1 <- red\_wine[1,]  
sample\_268 <- red\_wine[268,]

# Predicting outcome for sample with 1st row  
predict(model\_final,sample\_1,type='response')

## 1   
## 0.007680065

pred\_1 <- ifelse(predict(model\_final,sample\_1)>0.5,1,0)  
pred\_1[[1]]

## [1] 0

# Probability and link prediction  
pred\_l <- predict(model\_final,sample\_1,type='link',se=T)  
pred\_p <- predict(model\_final,sample\_1,type='response',se=T)  
  
# Confidence interval  
round(ilogit(c(pred\_l$fit-1.96\*pred\_l$se,pred\_l$fit+1.96\*pred\_l$se)),4)

## 1 1   
## 0.0045 0.0131

# Predicting outcome for sample with 1st row  
predict(model\_final,sample\_268,type="response")[[1]]

## [1] 0.6286813

pred\_268 <- ifelse(predict(model\_final,sample\_268)>0.5,1,0)  
pred\_268[[1]]

## [1] 1

# Probability and link prediction  
pred\_l <- predict(model\_final,sample\_268,type='link',se=T)  
pred\_p <- predict(model\_final,sample\_268,type='response',se=T)  
  
# Confidence interval  
round(ilogit(c(pred\_l$fit-1.96\*pred\_l$se,pred\_l$fit+1.96\*pred\_l$se)),4)

## 268 268   
## 0.4853 0.7525

# Comparing the predicted quality with observed quality  
red\_wine[1,12]

## [1] 0  
## Levels: 0 1

red\_wine[268,12]

## [1] 1  
## Levels: 0 1

Using the model generated above, we predicted the value of excellent quality for 1st row sample to be 0 and 1 for 268th sample. This is consistent with the observed qualities for both these samples.

The 95% confidence interval for 1st row sample is: 0.0045 - 0.0131 For 268th row sample: 0.4853 - 0.7525

# 

# Part D. Link functions and dispersion parameter

## D.1 Start with the final logistic regression model you have selected in Part C. A wine expert may ask: why do you think the logit link is appropriate? Is it possible that other link functions give better result? To answer these questions, do the following.

### D.1.a Fit another two models using the probit and complementary log-log links (with the same set of variables in your final logistic regression model).

# Fit models using all links  
model\_l <- glm(excellent ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar + density,family='binomial'(link=logit),red\_wine)

summary(model\_l)

##   
## Call:  
## glm(formula = excellent ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar +   
## density, family = binomial(link = logit), data = red\_wine)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.0158 -0.4314 -0.2220 -0.1255 2.9883   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.268e+02 9.163e+01 2.475 0.013336 \*   
## alcohol 7.823e-01 1.120e-01 6.983 2.88e-12 \*\*\*  
## volatile.acidity -2.913e+00 6.467e-01 -4.504 6.66e-06 \*\*\*  
## sulphates 3.699e+00 5.287e-01 6.997 2.62e-12 \*\*\*  
## total.sulfur.dioxide -1.360e-02 3.447e-03 -3.946 7.95e-05 \*\*\*  
## chlorides -8.441e+00 3.259e+00 -2.590 0.009593 \*\*   
## fixed.acidity 2.812e-01 8.029e-02 3.502 0.000462 \*\*\*  
## residual.sugar 2.328e-01 7.009e-02 3.322 0.000893 \*\*\*  
## density -2.409e+02 9.202e+01 -2.618 0.008835 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1269.92 on 1598 degrees of freedom  
## Residual deviance: 872.08 on 1590 degrees of freedom  
## AIC: 890.08  
##   
## Number of Fisher Scoring iterations: 6

model\_p <- glm(excellent ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar + density,family='binomial'(link=probit),red\_wine)

summary(model\_p)

##   
## Call:  
## glm(formula = excellent ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar +   
## density, family = binomial(link = probit), data = red\_wine)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.07806 -0.45118 -0.20474 -0.08787 3.12995   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.205e+02 5.068e+01 2.378 0.01741 \*   
## alcohol 4.297e-01 6.178e-02 6.955 3.52e-12 \*\*\*  
## volatile.acidity -1.567e+00 3.459e-01 -4.529 5.91e-06 \*\*\*  
## sulphates 2.039e+00 2.974e-01 6.858 6.96e-12 \*\*\*  
## total.sulfur.dioxide -7.457e-03 1.861e-03 -4.006 6.17e-05 \*\*\*  
## chlorides -4.403e+00 1.644e+00 -2.678 0.00740 \*\*   
## fixed.acidity 1.442e-01 4.388e-02 3.286 0.00102 \*\*   
## residual.sugar 1.237e-01 3.905e-02 3.168 0.00154 \*\*   
## density -1.283e+02 5.087e+01 -2.522 0.01168 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1269.92 on 1598 degrees of freedom  
## Residual deviance: 868.86 on 1590 degrees of freedom  
## AIC: 886.86  
##   
## Number of Fisher Scoring iterations: 7

model\_c <- glm(excellent ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar + density,family='binomial'(link=cloglog),red\_wine)

summary(model\_c)

##   
## Call:  
## glm(formula = excellent ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar +   
## density, family = binomial(link = cloglog), data = red\_wine)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -7.1486 -0.4995 -0.2983 -0.1888 2.7585   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.618e+02 7.700e+01 2.101 0.035652 \*   
## alcohol 5.715e-01 9.096e-02 6.283 3.33e-10 \*\*\*  
## volatile.acidity -2.438e+00 5.079e-01 -4.800 1.59e-06 \*\*\*  
## sulphates 2.804e+00 4.108e-01 6.826 8.73e-12 \*\*\*  
## total.sulfur.dioxide -1.136e-02 3.156e-03 -3.600 0.000318 \*\*\*  
## chlorides -6.664e+00 2.859e+00 -2.331 0.019747 \*   
## fixed.acidity 2.133e-01 6.400e-02 3.332 0.000861 \*\*\*  
## residual.sugar 2.175e-01 7.308e-02 2.976 0.002924 \*\*   
## density -1.725e+02 7.715e+01 -2.236 0.025347 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1269.92 on 1598 degrees of freedom  
## Residual deviance: 934.91 on 1590 degrees of freedom  
## AIC: 952.91  
##   
## Number of Fisher Scoring iterations: 25

### 

### D.1.b Compare the three models with respect to the significance, sign, and size of each variable

# Comparing coefficients  
variables <- c("Intercept","alcohol",'volatile.acidity','sulphates','total.sulfur.dioxide', 'chlorides', 'fixed.acidity', 'residual.sugar', 'density')  
coefs <- data.frame("Variables"=variables,"Logit"=NA,"Probit"=NA,"Cloglog"=NA)  
for (i in 1:9) {  
 coefs[i,2]= round(coef(model\_l)[[i]],3)  
 coefs[i,3]= round(coef(model\_p)[[i]],3)  
 coefs[i,4]= round(coef(model\_c)[[i]],3)  
}  
coefs

## Variables Logit Probit Cloglog  
## 1 Intercept 226.750 120.515 161.766  
## 2 alcohol 0.782 0.430 0.571  
## 3 volatile.acidity -2.913 -1.567 -2.438  
## 4 sulphates 3.699 2.039 2.804  
## 5 total.sulfur.dioxide -0.014 -0.007 -0.011  
## 6 chlorides -8.441 -4.403 -6.664  
## 7 fixed.acidity 0.281 0.144 0.213  
## 8 residual.sugar 0.233 0.124 0.217  
## 9 density -240.946 -128.268 -172.507

All the variables in all 3 models are significant with a p-value < 0.05.

The signs of all coefficients are consistent across all 3 models. alcohol, sulphates, fixed.acidity and residual.sugar having positive association while other negative.

The size of coefficients vary across the models. The absolute value of coefficients seem to follow the pattern: logit > cloglog > probit.

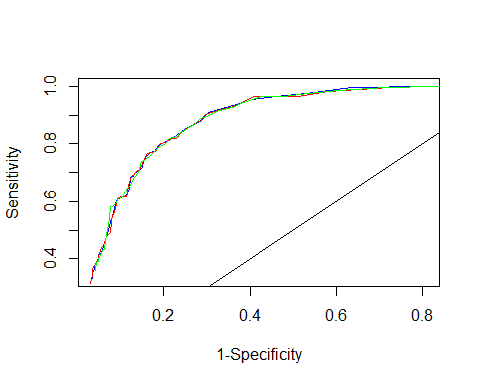
### 

### Compare the three models with respect to their predictive power. For example, you can compare the three ROCs and AUCs. You can also visually compare their predictions for those excellent wines as we are particularly interested in identifying them

# Predict probabilities for all models  
red\_wine\_n <- red\_wine  
  
# Predict values for final selected models  
  
# Logit  
red\_wine\_n$predprob\_l <- predict(model\_l,type='response')  
red\_wine\_n$pred\_l <- ifelse(red\_wine\_n$predprob\_l > 0.50, 1, 0)  
red\_wine\_n$pred\_l <- as.factor(red\_wine\_n$pred\_l)  
pred\_roc\_l <- prediction(red\_wine\_n$predprob\_l, red\_wine\_n$excellent)  
  
# Probit  
red\_wine\_n$predprob\_p <- predict(model\_p,type='response')  
red\_wine\_n$pred\_p <- ifelse(red\_wine\_n$predprob\_p > 0.50, 1, 0)  
red\_wine\_n$pred\_p <- as.factor(red\_wine\_n$pred\_p)  
pred\_roc\_p <- prediction(red\_wine\_n$predprob\_p, red\_wine\_n$excellent)  
  
# Complementary log-log  
red\_wine\_n$predprob\_c <- predict(model\_c, type='response')  
red\_wine\_n$pred\_c <- ifelse(red\_wine\_n$predprob\_c > 0.50, 1, 0)  
red\_wine\_n$pred\_c <- as.factor(red\_wine\_n$pred\_c)  
pred\_roc\_c <- prediction(red\_wine\_n$predprob\_c, red\_wine\_n$excellent)

# Specificity and sensitivity  
thresh = seq(0.01,0.5,0.01)  
  
sensitivity\_l<- specificity\_l<- rep( NA,length (thresh))  
sensitivity\_p<- specificity\_p<- rep( NA,length (thresh))  
sensitivity\_c<- specificity\_c<- rep( NA,length (thresh))  
  
for (j in seq(along=thresh)){  
  
pp\_l=ifelse(red\_wine\_n$predprob\_l<thresh[j],0,1)  
xx\_l= xtabs(~excellent+pp\_l,red\_wine\_n)  
specificity\_l[j]= xx\_l[1,1]/(xx\_l[1,1]+xx\_l[1,2])  
sensitivity\_l[j]= xx\_l[2,2]/(xx\_l[2,1]+xx\_l[2,2])  
   
pp\_p=ifelse(red\_wine\_n$predprob\_p<thresh[j],0,1)  
xx\_p= xtabs(~excellent+pp\_p,red\_wine\_n)  
specificity\_p[j]= xx\_p[1,1]/(xx\_p[1,1]+xx\_p[1,2])  
sensitivity\_p[j]= xx\_p[2,2]/(xx\_p[2,1]+xx\_p[2,2])  
  
pp\_c=ifelse(red\_wine\_n$predprob\_c<thresh[j],0,1)  
xx\_c= xtabs(~excellent+pp\_c,red\_wine\_n)  
specificity\_c[j]= xx\_c[1,1]/(xx\_c[1,1]+xx\_c[1,2])  
sensitivity\_c[j]= xx\_c[2,2]/(xx\_c[2,1]+xx\_c[2,2])  
}

# ROC curves  
plot(1-specificity\_l,sensitivity\_l,type="l",,lty=1,col="blue",xlab ="1-Specificity", ylab ="Sensitivity"); abline (coef=c(0,1))  
points(1-specificity\_p,sensitivity\_p,type="l",lty=1,col="red")  
points(1-specificity\_c,sensitivity\_c,type="l",lty=1,col="green")



perf\_l <- performance(pred\_roc\_l, "tpr", "fpr")  
perf\_p <- performance(pred\_roc\_p, "tpr", "fpr")  
perf\_c <- performance(pred\_roc\_c, "tpr", "fpr")

# AUC  
unlist(slot(performance(pred\_roc\_l, "auc"), "y.values"))

## [1] 0.881985

unlist(slot(performance(pred\_roc\_p, "auc"), "y.values"))

## [1] 0.8820483

unlist(slot(performance(pred\_roc\_c, "auc"), "y.values"))

## [1] 0.881178

As we can see, the ROC curves and AUC values are almost same for all models. We will now look at predictive power of each model.

# Misclassification rate for all models  
misc\_l <- table(red\_wine\_n$excellent, red\_wine\_n$pred\_l, dnn=c("Truth","Predicted"))  
miscr\_l <- (misc\_l[1,2] + misc\_l[2,1])/sum(misc\_l)  
  
misc\_p <- table(red\_wine\_n$excellent, red\_wine\_n$pred\_p, dnn=c("Truth","Predicted"))  
miscr\_p <- (misc\_p[1,2] + misc\_p[2,1])/sum(misc\_p)  
  
misc\_c <- table(red\_wine\_n$excellent, red\_wine\_n$pred\_c, dnn=c("Truth","Predicted"))  
miscr\_c <- (misc\_c[1,2] + misc\_c[2,1])/sum(misc\_c)  
  
miscr\_l

## [1] 0.1194497

miscr\_p

## [1] 0.1194497

miscr\_c

## [1] 0.1219512

The misclassification rate is also very similar for all 3 models.

On comparing various metrics for 3 models, we could not find stark difference between any of those and all of them seem to perform equally well. Thus, we will go ahead wit the default link option of logit for better interpretability.

## 

## D.2. Do you think the default value of the dispersion parameter being 1 can be reasonably used in the wine analysis? If not, do the following.

### D.2.a Adjust the dispersion parameter to give a better fit.

# Calculate dispersion paramter  
sigma.squared<-sum(residuals(model\_l, type="pearson")^2)/(nrow(red\_wine)-9)  
sigma.squared

## [1] 0.8415362

The updated dispersion parameter is 0.841

### 

### D.2.b Report the final model with the updated dispersion parameter.

# Updated model with dispersion parameter  
summary(model\_l,dispersion=sigma.squared)

##   
## Call:  
## glm(formula = excellent ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar +   
## density, family = binomial(link = logit), data = red\_wine)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.0158 -0.4314 -0.2220 -0.1255 2.9883   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.268e+02 8.406e+01 2.698 0.006984 \*\*   
## alcohol 7.823e-01 1.028e-01 7.612 2.69e-14 \*\*\*  
## volatile.acidity -2.913e+00 5.933e-01 -4.910 9.11e-07 \*\*\*  
## sulphates 3.699e+00 4.850e-01 7.627 2.41e-14 \*\*\*  
## total.sulfur.dioxide -1.360e-02 3.162e-03 -4.301 1.70e-05 \*\*\*  
## chlorides -8.441e+00 2.989e+00 -2.824 0.004750 \*\*   
## fixed.acidity 2.812e-01 7.365e-02 3.817 0.000135 \*\*\*  
## residual.sugar 2.328e-01 6.430e-02 3.621 0.000293 \*\*\*  
## density -2.409e+02 8.442e+01 -2.854 0.004314 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 0.8415362)  
##   
## Null deviance: 1269.92 on 1598 degrees of freedom  
## Residual deviance: 872.08 on 1590 degrees of freedom  
## AIC: 890.08  
##   
## Number of Fisher Scoring iterations: 6

### D.2.c Explain to a wine expert what inference has changed and what remains the same.

On using the dispersion parameter calculated earlier, the standard error for all coefficients has reduced (since the dispersion paramter is < 1). p-value has also changed but all variables are statistically significant in new model also.

The coefficient estimates have not changed, in the new model. They remain constant, irrespective of the dispersion factor.

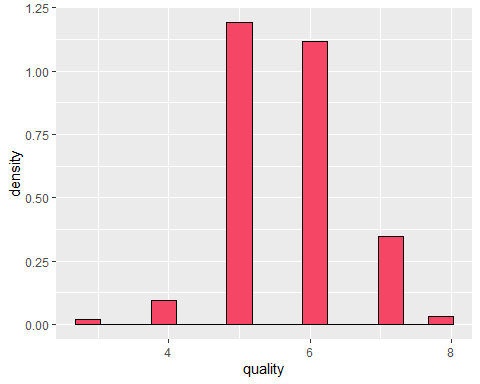
# 

# Part E. Modeling the wine quality as a multinomial variable with order

## E.1 In Part E, we will use the original wine quality variable as our response variable. How many categories does this variable contain? How many categories do we actually observe? Plot a histogram showing the distribution of this quality variable.

# Load wine dataset to get back quality variable  
red\_wine <- read.csv("winequality-red.csv", sep = ";",header = T)

# Plot histogram  
ggplot(red\_wine, aes(x=quality)) +   
 geom\_histogram(aes(y=..density..), color='black', fill = '#F64666',bins = 15)



The variable quality has 6 categories which can also be seen from the above plot.

## E.2 What distribution should we use to model this quality variable? Is it continuous, binary, or multinomial? Is there any order between the different ratings?

Quality variable follows approximately normal distribution. The variable quality is multinomial and ordered with 6 levels..

## 

## E.3 Calculate the correlations between the quality variable and each of the predictors. Note that Pearson correlation may not be a good measure in this situation. Consider using Kendall’s tau.

variables <- c('fixed.acidity','volatile.acidity','citric.acid', 'residual.sugar', 'chlorides', 'free.sulfur.dioxide','total.sulfur.dioxide', 'density', 'pH', 'sulphates', 'alcohol')  
  
kendalls <- data.frame("Variables"=variables,'Kendall'=NA)  
  
for (i in 1:length(variables)) {  
 kendalls[i,2]= cor(red\_wine$quality,red\_wine[,i],method = 'kendall')  
}  
  
# Sorting the table based on Kenall Correlation value  
kendalls$kendall\_abs <- abs(kendalls$Kendall)  
kendalls <- kendalls[order(-kendalls$kendall\_abs),]  
kendalls <- subset(kendalls,select = -c(kendall\_abs))  
kendalls

## Variables Kendall  
## 11 alcohol 0.38036731  
## 2 volatile.acidity -0.30077867  
## 10 sulphates 0.29926962  
## 3 citric.acid 0.16731793  
## 7 total.sulfur.dioxide -0.15661215  
## 5 chlorides -0.14891936  
## 8 density -0.13661135  
## 1 fixed.acidity 0.08796609  
## 6 free.sulfur.dioxide -0.04564591  
## 9 pH -0.03423504  
## 4 residual.sugar 0.02574400

## 

## E.4 Hand pick a subset of predictors using the ranking in Step 3 and the knowledge you have built in Parts A-D.

Based on the above table and information we have from Parts A-D, we would remove fixed.acidity and residual.sugar in addition to initially removed variables.

We are dropping residual.sugar and fixed.acidity since it has a correlation value < 0.1.

Final list of covariates: alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + density

## 

## E.5 Use the predictors you picked in Step 4 to fit a model for the quality response. What model should we use? Why? Present the model fitting result and discuss the implications.

We should use a proportional odds model since our response variable is multinomial ordinal.

# Convert quality to ordinal  
red\_wine$quality <- ordered(red\_wine$quality, levels=3:8)

# Fitting a model  
model\_m <- vglm(formula = quality ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + density, family=cumulative(parallel=T),red\_wine)  
summary(model\_m)

##   
## Call:  
## vglm(formula = quality ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide + chlorides + density, family = cumulative(parallel = T),   
## data = red\_wine)  
##   
## Coefficients:   
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept):1 29.659824 31.382674 0.945 0.345   
## (Intercept):2 31.575441 31.381674 1.006 0.314   
## (Intercept):3 35.272314 31.384897 1.124 0.261   
## (Intercept):4 38.091588 31.386736 NA NA   
## (Intercept):5 41.091729 31.387081 1.309 0.190   
## alcohol -0.874864 0.062100 -14.088 < 2e-16 \*\*\*  
## volatile.acidity 3.392940 0.311887 10.879 < 2e-16 \*\*\*  
## sulphates -2.768300 0.346282 -7.994 1.30e-15 \*\*\*  
## total.sulfur.dioxide 0.008106 0.001602 5.061 4.18e-07 \*\*\*  
## chlorides 4.949119 1.227150 4.033 5.51e-05 \*\*\*  
## density -27.313856 31.208891 -0.875 0.381   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Number of linear predictors: 5   
##   
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2]),   
## logitlink(P[Y<=3]), logitlink(P[Y<=4]), logitlink(P[Y<=5])  
##   
## Residual deviance: 3096.75 on 7984 degrees of freedom  
##   
## Log-likelihood: -1548.375 on 7984 degrees of freedom  
##   
## Number of Fisher scoring iterations: 6   
##   
## Warning: Hauck-Donner effect detected in the following estimate(s):  
## '(Intercept):4'  
##   
##   
## Exponentiated coefficients:  
## alcohol volatile.acidity sulphates   
## 4.169185e-01 2.975331e+01 6.276864e-02   
## total.sulfur.dioxide chlorides density   
## 1.008139e+00 1.410506e+02 1.373230e-12

For a proportional odds model, we get (k-1) intercepts, where k = number of orders in response variable Y (k=6 in our case).

The coefficients of proportional odds model can be interpreted using following example: Consider the variable alcohol. For unit increase in alcohol, the log odds of probability of quality <= (ord)i will increase by Ci, where (ord)i is ith order. Quality = 3 will be 1st order, quality = 4 will be 2nd order. Ci is the ith intercept from the model summary.

For alcohol, probability of quality <= 3 will be equal to ilogit(-130.70433 - 0.776x), where x is alcohol level. Using other intercepts from model, we can gauge the impact of other variables on our response variable quality.

## 

## E.6 Compare the conclusions you draw in Step 5 and those you drew from the logistic regression you built for the variable “excellent”. Are there any similarities and differences?

On comparing above model with the logistic regression model, we observe that the sign of all coefficients has changed. This is because in vlgm model, beta gives the change in probability of outcome being less than each level in ordered data.

Since we are working on a proportional odds model, the beta is same for all levels.

For example, in an ordered data with possible outcomes in (1,2,3), assume beta for some predictor variable x, then unit increase in x will increase the log odds of outcome y <= 1 , and eventually its probability. It will also increase the probability of outcome y <= 2. Probability of y <=3 = 1 - other 2 probabilities. Thus by increasing the 2 probabilities, we are decreasing the probability of y <= 3.

Similarly, if the beta is negative for x, unit increase in x will decrease the probabilities for y<=1, y<=2 and thus increase the probability of y<=3.

This explains the change in signs of coefficients between logistic and ordered multinomial regression models.

In logistic regression, positive coefficient implies that increase in the variable will lead to an increase in probability of desired outcome. The outcome is reversed in case of proportional odds model due to the interpretation of its coefficients.

## 

## E.7 Can we use the model you built in Step 5 to predict whether or not a bottle of wine is “excellent” (rating is at least 7)? If yes, how? How do you evaluate the prediction result? How does the result compare to that based on the logistic regression model (Part C)?

Yes, we can use the model built in step 5 to predict whether or not a bottle of wine is excellent. First, we will use the proportional odds model to predict probabilities of each level and then use it to determine the quality.

Once we have the quality of a particular wine bottle, we can determine whether it is excellent or not depending on if it is less than greater than equal to 7.

To evaluate the prediction result, we can predict data for given set of samples and form a confusion matrix using observed values of qualities or with outcomes of logistic regression model prediction. This matrix can then be used to calculate the misclassification rate.

To evaluate the prediction result, we can predict data for given set of samples and form a confusion matrix using observed values of qualities or with outcomes of logistic regression model prediction. This matrix can then be used to calculate the misclassification rate.

We can also predict the quality for 2 samples we used earlier and compare with their observed values.

# Sample 1  
pred\_1 <- predict(model\_m, newdata = red\_wine[1,], "response")   
ifelse(sum(pred\_1[,1:4]) > 0.5,0,1)

## [1] 0

# Sample 2  
pred\_268 <- predict(model\_m, newdata = red\_wine[268,], "response")   
ifelse(sum(pred\_268[,1:4]) > 0.5,0,1)

## [1] 1

The results obtained are consistent with the observed values of these wine bottles.

We can also compare predictions for logistic and proportional odds model.

red\_wine <- read.csv("winequality-red.csv", sep = ";",header = T)  
red\_wine$quality <- ordered(red\_wine$quality, levels=3:8)  
  
# Converting quality to excellent  
red\_wine$excellent <- as.factor(ifelse(red\_wine$quality>6,1,0))  
red\_wine <- subset(red\_wine,select = -c(quality))  
red\_wine\_n <- red\_wine  
  
# Predict outcome using logistic regression  
red\_wine\_n$predprob\_l <- predict(model\_final,red\_wine,type='response')  
red\_wine\_n$pred\_l <- ifelse(red\_wine\_n$predprob\_l>0.5,1,0)  
  
# Predict outcome using proportional odds model  
predprob\_m <- predict(model\_m,red\_wine,type='response')  
predprob\_m <- as.data.frame(predprob\_m)  
  
n <- nrow(predprob\_m)  
  
for (i in 1:n) {  
 predprob\_m$max[i] = max(predprob\_m[i,1:6])  
 predprob\_m$exc[i] <- ifelse(sum(predprob\_m[i,1:4]) > 0.5,0,1)  
}  
  
  
misc <- table(red\_wine\_n$pred\_l, predprob\_m$exc, dnn=c("Logistic","VGLM"))  
miscr <- (misc[1,2]+misc[2,1])/sum(misc)  
miscr

## [1] 0.02626642

There is a difference of 2.62% between the predictions by proportional odds model and earlier logistic regression model. Thus we can conclude that the new model performs well in terms of prediction result when compared to logistic model.