DS740\_FinalProject\_BHOWMICK

Gautam

December 8, 2017

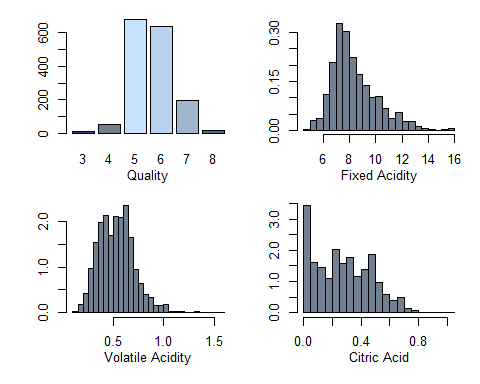
## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

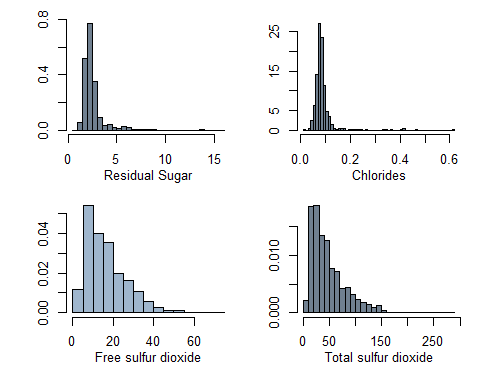
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

## load the data and analyze the data  
  
redwine<-read.csv("winequality-red.csv")  
attach(redwine)  
n=dim(redwine)[1]  
redwine$good <- rep(0, length(redwine$quality))  
redwine$good <- ifelse(redwine$quality >= 7,1,0)  
redwine$good <- as.factor(redwine$good)  
factQuality <- as.factor(quality)  
redwine <- cbind(redwine, factQuality)  
temp <- recode(redwine$factQuality, "c('3','4','5','6')='10';else='40'")  
p.temp <- recode(temp, "c('10')='okay'; else='good'")  
redwine$factQuality <- p.temp

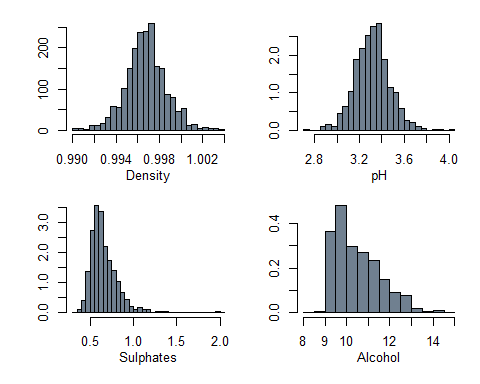
# Types of variables and how they relate - Data Visualization  
#Data Distribution for Quality, Fixed Acidity and Volatile Acidity  
par(mfrow=c(2,2), oma = c(1,1,0,0) + 0.1, mar = c(3,3,1,1) + 0.1)  
barplot((table(quality)), col=c("slateblue4", "slategray", "slategray1", "slategray2", "slategray3", "skyblue4"))  
mtext("Quality", side=1, outer=F, line=2, cex=0.8)  
truehist(fixed.acidity, col="slategray")  
mtext("Fixed Acidity", side=1, outer=F, line=2, cex=0.8)  
truehist(volatile.acidity, col="slategray")  
mtext("Volatile Acidity", side=1, outer=F, line=2, cex=0.8)  
truehist(citric.acid, col="slategray")  
mtext("Citric Acid", side=1, outer=F, line=2, cex=0.8)



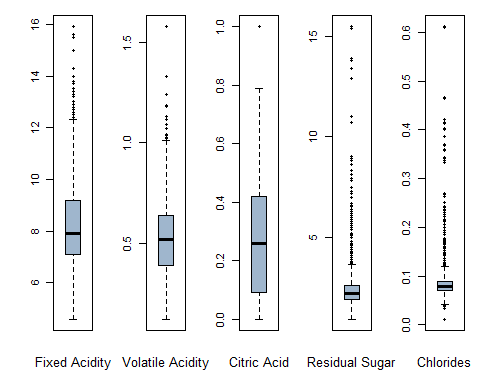
#Data Distribution for Residual Sugar,Chlorides,Free sulfur dioxide & Total sulfur dioxide  
par(mfrow=c(1,1))  
par(mfrow=c(2,2), oma = c(1,1,0,0) + 0.1, mar = c(3,3,1,1) + 0.1)  
truehist(residual.sugar,col="slategray")  
mtext("Residual Sugar", side=1, outer=F, line=2, cex=0.8)  
truehist(chlorides, col="slategray")  
mtext("Chlorides", side=1, outer=F, line=2, cex=0.8)  
truehist(free.sulfur.dioxide, col="slategray3")  
mtext("Free sulfur dioxide", side=1, outer=F, line=2, cex=0.8)  
truehist(total.sulfur.dioxide,col="slategray")  
mtext("Total sulfur dioxide", side=1, outer=F, line=2, cex=0.8)



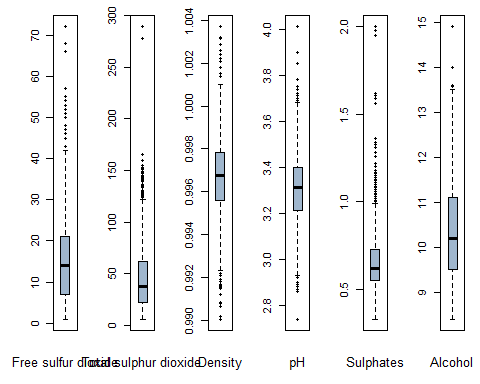
#Data Distribution for Density,pH,Sulphates & Alcohol  
par(mfrow=c(1,1))  
par(mfrow=c(2,2), oma = c(1,1,0,0) + 0.1, mar = c(3,3,1,1) + 0.1)  
truehist(density, col="slategray")  
mtext("Density", side=1, outer=F, line=2, cex=0.8)  
truehist(pH, col="slategray")  
mtext("pH", side=1, outer=F, line=2, cex=0.8)  
truehist(sulphates, col="slategray")  
mtext("Sulphates", side=1, outer=F, line=2, cex=0.8)  
truehist(alcohol,col="slategray")  
mtext("Alcohol", side=1, outer=F, line=2, cex=0.8)



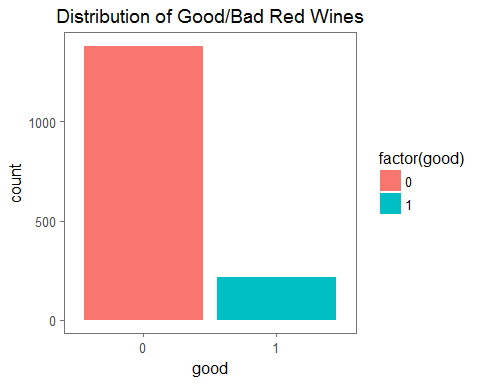
#Boxplots for each of the variables as another indicator of spread.  
par(mfrow=c(1,5), oma = c(1,1,0,0) + 0.1, mar = c(3,3,1,1) + 0.1)  
boxplot(fixed.acidity, col="slategray3", pch=19)  
mtext("Fixed Acidity", cex=0.8, side=1, line=2)  
boxplot(volatile.acidity, col="slategray3", pch=19)  
mtext("Volatile Acidity", cex=0.8, side=1, line=2)  
boxplot(citric.acid, col="slategray3", pch=19)  
mtext("Citric Acid", cex=0.8, side=1, line=2)  
boxplot(residual.sugar, col="slategray3", pch=19)  
mtext("Residual Sugar", cex=0.8, side=1, line=2)  
boxplot(chlorides, col="slategray3", pch=19)  
mtext("Chlorides", cex=0.8, side=1, line=2)



par(mfrow=c(1,1))  
par(mfrow=c(1,6), oma = c(1,1,0,0) + 0.1, mar = c(3,3,1,1) + 0.1)  
boxplot(free.sulfur.dioxide, col="slategray3", pch=19)  
mtext("Free sulfur dioxide", cex=0.8, side=1, line=2)  
boxplot(total.sulfur.dioxide, col="slategray3", pch=19)  
mtext("Total sulphur dioxide", cex=0.8, side=1, line=2)  
boxplot(density, col="slategray3", pch=19)  
mtext("Density", cex=0.8, side=1, line=2)  
boxplot(pH, col="slategray3", pch=19)  
mtext("pH", cex=0.8, side=1, line=2)  
boxplot(sulphates, col="slategray3", pch=19)  
mtext("Sulphates", cex=0.8, side=1, line=2)  
boxplot(alcohol, col="slategray3", pch=19)  
mtext("Alcohol", cex=0.8, side=1, line=2)



#Distribution of good/bad red wines  
ggplot(redwine,aes(x=good,fill=factor(good)))+geom\_bar(stat = "count",position = "dodge")+scale\_x\_discrete(breaks = seq(0,1,1))+  
theme\_few()+ggtitle("Distribution of Good/Bad Red Wines")+theme(plot.title = element\_text(hjust = 0.5))



##Check for corelation between the predictors  
describe(redwine[,-12:-14])

## vars n mean sd median trimmed mad min  
## fixed.acidity 1 1599 8.32 1.74 7.90 8.15 1.48 4.60  
## volatile.acidity 2 1599 0.53 0.18 0.52 0.52 0.18 0.12  
## citric.acid 3 1599 0.27 0.19 0.26 0.26 0.25 0.00  
## residual.sugar 4 1599 2.54 1.41 2.20 2.26 0.44 0.90  
## chlorides 5 1599 0.09 0.05 0.08 0.08 0.01 0.01  
## free.sulfur.dioxide 6 1599 15.87 10.46 14.00 14.58 10.38 1.00  
## total.sulfur.dioxide 7 1599 46.47 32.90 38.00 41.84 26.69 6.00  
## density 8 1599 1.00 0.00 1.00 1.00 0.00 0.99  
## pH 9 1599 3.31 0.15 3.31 3.31 0.15 2.74  
## sulphates 10 1599 0.66 0.17 0.62 0.64 0.12 0.33  
## alcohol 11 1599 10.42 1.07 10.20 10.31 1.04 8.40  
## max range skew kurtosis se  
## fixed.acidity 15.90 11.30 0.98 1.12 0.04  
## volatile.acidity 1.58 1.46 0.67 1.21 0.00  
## citric.acid 1.00 1.00 0.32 -0.79 0.00  
## residual.sugar 15.50 14.60 4.53 28.49 0.04  
## chlorides 0.61 0.60 5.67 41.53 0.00  
## free.sulfur.dioxide 72.00 71.00 1.25 2.01 0.26  
## total.sulfur.dioxide 289.00 283.00 1.51 3.79 0.82  
## density 1.00 0.01 0.07 0.92 0.00  
## pH 4.01 1.27 0.19 0.80 0.00  
## sulphates 2.00 1.67 2.42 11.66 0.00  
## alcohol 14.90 6.50 0.86 0.19 0.03

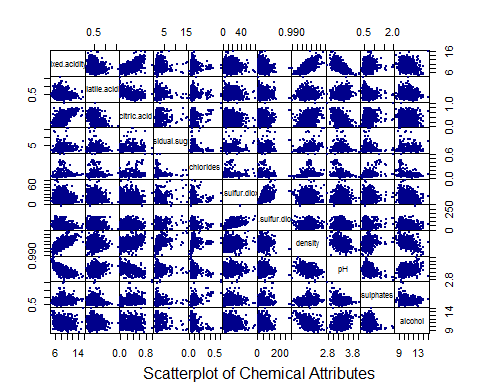
round(cor(redwine[,-12:-14]),2)

## fixed.acidity volatile.acidity citric.acid  
## fixed.acidity 1.00 -0.26 0.67  
## volatile.acidity -0.26 1.00 -0.55  
## citric.acid 0.67 -0.55 1.00  
## residual.sugar 0.11 0.00 0.14  
## chlorides 0.09 0.06 0.20  
## free.sulfur.dioxide -0.15 -0.01 -0.06  
## total.sulfur.dioxide -0.11 0.08 0.04  
## density 0.67 0.02 0.36  
## pH -0.68 0.23 -0.54  
## sulphates 0.18 -0.26 0.31  
## alcohol -0.06 -0.20 0.11  
## residual.sugar chlorides free.sulfur.dioxide  
## fixed.acidity 0.11 0.09 -0.15  
## volatile.acidity 0.00 0.06 -0.01  
## citric.acid 0.14 0.20 -0.06  
## residual.sugar 1.00 0.06 0.19  
## chlorides 0.06 1.00 0.01  
## free.sulfur.dioxide 0.19 0.01 1.00  
## total.sulfur.dioxide 0.20 0.05 0.67  
## density 0.36 0.20 -0.02  
## pH -0.09 -0.27 0.07  
## sulphates 0.01 0.37 0.05  
## alcohol 0.04 -0.22 -0.07  
## total.sulfur.dioxide density pH sulphates alcohol  
## fixed.acidity -0.11 0.67 -0.68 0.18 -0.06  
## volatile.acidity 0.08 0.02 0.23 -0.26 -0.20  
## citric.acid 0.04 0.36 -0.54 0.31 0.11  
## residual.sugar 0.20 0.36 -0.09 0.01 0.04  
## chlorides 0.05 0.20 -0.27 0.37 -0.22  
## free.sulfur.dioxide 0.67 -0.02 0.07 0.05 -0.07  
## total.sulfur.dioxide 1.00 0.07 -0.07 0.04 -0.21  
## density 0.07 1.00 -0.34 0.15 -0.50  
## pH -0.07 -0.34 1.00 -0.20 0.21  
## sulphates 0.04 0.15 -0.20 1.00 0.09  
## alcohol -0.21 -0.50 0.21 0.09 1.00

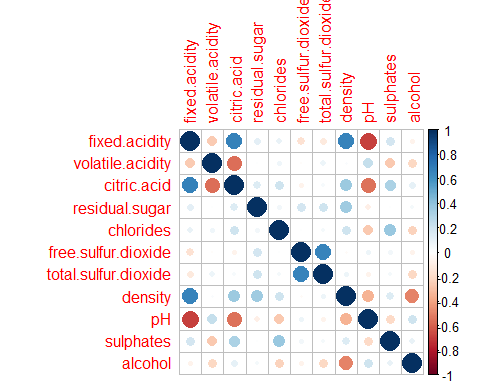
round(cor(redwine[,-12:-14], method="spearman"),2)

## fixed.acidity volatile.acidity citric.acid  
## fixed.acidity 1.00 -0.28 0.66  
## volatile.acidity -0.28 1.00 -0.61  
## citric.acid 0.66 -0.61 1.00  
## residual.sugar 0.22 0.03 0.18  
## chlorides 0.25 0.16 0.11  
## free.sulfur.dioxide -0.18 0.02 -0.08  
## total.sulfur.dioxide -0.09 0.09 0.01  
## density 0.62 0.03 0.35  
## pH -0.71 0.23 -0.55  
## sulphates 0.21 -0.33 0.33  
## alcohol -0.07 -0.22 0.10  
## residual.sugar chlorides free.sulfur.dioxide  
## fixed.acidity 0.22 0.25 -0.18  
## volatile.acidity 0.03 0.16 0.02  
## citric.acid 0.18 0.11 -0.08  
## residual.sugar 1.00 0.21 0.07  
## chlorides 0.21 1.00 0.00  
## free.sulfur.dioxide 0.07 0.00 1.00  
## total.sulfur.dioxide 0.15 0.13 0.79  
## density 0.42 0.41 -0.04  
## pH -0.09 -0.23 0.12  
## sulphates 0.04 0.02 0.05  
## alcohol 0.12 -0.28 -0.08  
## total.sulfur.dioxide density pH sulphates alcohol  
## fixed.acidity -0.09 0.62 -0.71 0.21 -0.07  
## volatile.acidity 0.09 0.03 0.23 -0.33 -0.22  
## citric.acid 0.01 0.35 -0.55 0.33 0.10  
## residual.sugar 0.15 0.42 -0.09 0.04 0.12  
## chlorides 0.13 0.41 -0.23 0.02 -0.28  
## free.sulfur.dioxide 0.79 -0.04 0.12 0.05 -0.08  
## total.sulfur.dioxide 1.00 0.13 -0.01 0.00 -0.26  
## density 0.13 1.00 -0.31 0.16 -0.46  
## pH -0.01 -0.31 1.00 -0.08 0.18  
## sulphates 0.00 0.16 -0.08 1.00 0.21  
## alcohol -0.26 -0.46 0.18 0.21 1.00

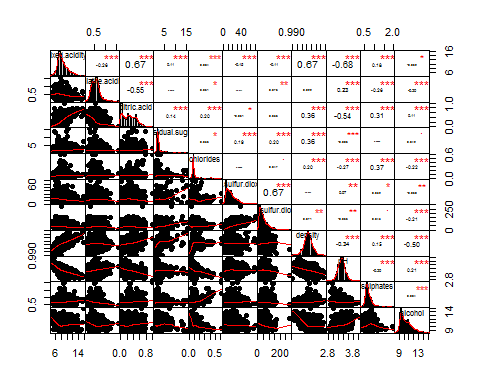
pairs(redwine[,-12:-14], gap=0, pch=19, cex=0.4, col="darkblue")  
title(sub="Scatterplot of Chemical Attributes", cex=0.8)



corrplot(cor(redwine[,-12:-14]))



suppressWarnings(warning(chart.Correlation(redwine[,-12:-14], histogram=TRUE, pch=19)))



#Find out if there are other predictors with high VIF, but they were not removed at this step.  
fit.one <- glm(good ~.-quality-factQuality, data=redwine, family= "binomial")  
summary(fit.one)

##   
## Call:  
## glm(formula = good ~ . - quality - factQuality, family = "binomial",   
## data = redwine)  
##   
## 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

vif(fit.one)

## fixed.acidity volatile.acidity citric.acid   
## 8.548951 1.789987 3.455691   
## residual.sugar chlorides free.sulfur.dioxide   
## 1.714940 1.252465 2.086780   
## total.sulfur.dioxide density pH   
## 2.305224 7.948391 3.409585   
## sulphates alcohol   
## 1.319694 2.536675

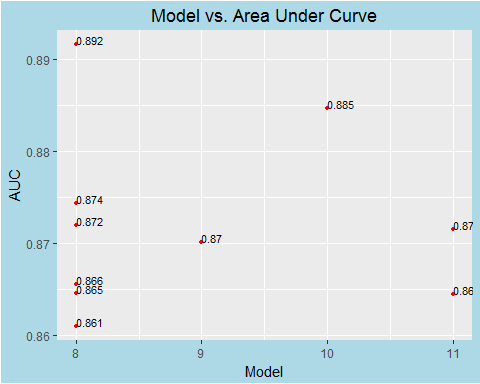
#Not all predictors are significant. A forward selection method is employed to build a working model.   
fit.two <- step(glm(good ~ 1, data=redwine,family= "binomial"), scope=list(lower=~1, upper = ~fixed.acidity +volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol),direction="forward")

## Start: AIC=1271.92  
## good ~ 1  
##   
## Df Deviance AIC  
## + alcohol 1 1027.9 1031.9  
## + volatile.acidity 1 1130.6 1134.6  
## + citric.acid 1 1197.2 1201.2  
## + sulphates 1 1218.1 1222.1  
## + total.sulfur.dioxide 1 1232.5 1236.5  
## + density 1 1233.2 1237.2  
## + chlorides 1 1239.5 1243.5  
## + fixed.acidity 1 1248.4 1252.4  
## + free.sulfur.dioxide 1 1261.1 1265.1  
## + pH 1 1264.6 1268.6  
## + residual.sugar 1 1266.7 1270.7  
## <none> 1269.9 1271.9  
##   
## Step: AIC=1031.89  
## good ~ alcohol  
##   
## Df Deviance AIC  
## + volatile.acidity 1 948.48 954.48  
## + citric.acid 1 975.02 981.02  
## + sulphates 1 975.76 981.76  
## + fixed.acidity 1 987.08 993.08  
## + pH 1 991.30 997.30  
## + total.sulfur.dioxide 1 1013.65 1019.65  
## + density 1 1019.13 1025.13  
## + free.sulfur.dioxide 1 1023.45 1029.45  
## <none> 1027.89 1031.89  
## + chlorides 1 1026.32 1032.32  
## + residual.sugar 1 1026.50 1032.50  
##   
## Step: AIC=954.48  
## good ~ alcohol + volatile.acidity  
##   
## Df Deviance AIC  
## + sulphates 1 917.26 925.26  
## + fixed.acidity 1 932.01 940.01  
## + total.sulfur.dioxide 1 936.34 944.34  
## + pH 1 937.82 945.82  
## + citric.acid 1 941.11 949.11  
## + density 1 941.30 949.30  
## + free.sulfur.dioxide 1 944.00 952.00  
## <none> 948.48 954.48  
## + residual.sugar 1 946.90 954.90  
## + chlorides 1 947.52 955.52  
##   
## Step: AIC=925.26  
## good ~ alcohol + volatile.acidity + sulphates  
##   
## Df Deviance AIC  
## + total.sulfur.dioxide 1 899.55 909.55  
## + fixed.acidity 1 905.71 915.71  
## + free.sulfur.dioxide 1 910.42 920.42  
## + chlorides 1 911.20 921.20  
## + pH 1 911.70 921.70  
## + citric.acid 1 914.11 924.11  
## + density 1 914.82 924.82  
## + residual.sugar 1 915.24 925.24  
## <none> 917.26 925.26  
##   
## Step: AIC=909.55  
## good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide  
##   
## Df Deviance AIC  
## + chlorides 1 890.90 902.90  
## + fixed.acidity 1 893.48 905.48  
## + residual.sugar 1 894.74 906.74  
## + pH 1 895.19 907.19  
## <none> 899.55 909.55  
## + citric.acid 1 897.60 909.60  
## + density 1 898.65 910.65  
## + free.sulfur.dioxide 1 899.31 911.31  
##   
## Step: AIC=902.9  
## good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide +   
## chlorides  
##   
## Df Deviance AIC  
## + fixed.acidity 1 883.17 897.17  
## + pH 1 883.36 897.36  
## + residual.sugar 1 884.50 898.50  
## + citric.acid 1 884.85 898.85  
## <none> 890.90 902.90  
## + density 1 889.04 903.04  
## + free.sulfur.dioxide 1 890.74 904.74  
##   
## Step: AIC=897.17  
## good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide +   
## chlorides + fixed.acidity  
##   
## Df Deviance AIC  
## + residual.sugar 1 878.99 894.99  
## <none> 883.17 897.17  
## + density 1 881.60 897.60  
## + pH 1 881.79 897.79  
## + free.sulfur.dioxide 1 882.71 898.71  
## + citric.acid 1 882.72 898.72  
##   
## Step: AIC=894.99  
## good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide +   
## chlorides + fixed.acidity + residual.sugar  
##   
## Df Deviance AIC  
## + density 1 872.08 890.08  
## <none> 878.99 894.99  
## + pH 1 877.59 895.59  
## + free.sulfur.dioxide 1 878.15 896.15  
## + citric.acid 1 878.86 896.86  
##   
## Step: AIC=890.08  
## good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide +   
## chlorides + fixed.acidity + residual.sugar + density  
##   
## Df Deviance AIC  
## <none> 872.08 890.08  
## + free.sulfur.dioxide 1 871.33 891.33  
## + citric.acid 1 871.78 891.78  
## + pH 1 872.01 892.01

#Regression model list specification  
nmodels = 11  
Model1 = (good ~ alcohol)  
Model2 = (good ~ alcohol + volatile.acidity)  
Model3 = (good ~ alcohol + volatile.acidity + sulphates)  
Model4 = (good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide)  
Model5 = (good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides)  
Model6 = (good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity)  
Model7 = (good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar)  
Model8 = (good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar + density)  
Model9 = (good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar + density + free.sulfur.dioxide)  
Model10 = (good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar + density + free.sulfur.dioxide + citric.acid)   
Model11 = (good ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide + chlorides + fixed.acidity + residual.sugar + density + free.sulfur.dioxide + citric.acid + pH)  
allModels = list(Model1,Model2,Model3,Model4,Model5,Model6,Model7,Model8,Model9,Model10,Model11)

##### General Regression model assessment OUTER CV (with model selection INNER CV as part of model-fitting) #####  
  
fulldata.out = redwine  
k.out = 10   
n.out = dim(fulldata.out)[1]  
#define the cross-validation splits   
groups.out = c(rep(1:k.out,floor(n.out/k.out)),1:(n.out%%k.out)) #produces list of group labels  
set.seed(8)  
cvgroups.out = sample(groups.out,n.out) #orders randomly, with seed (8)   
allpredictedCV.glm.out=rep(NA,n.out)  
glm.best.model.list = data.frame( Model=rep(0, k.out), AUC=rep(0,k.out))  
medel.list<-c("Model","AUC")  
for (j in 1:k.out) { #be careful not to re-use loop indices  
 groupj.out = (cvgroups.out == j)  
 traindata.out = redwine[!groupj.out,]  
 testdata.out = redwine[groupj.out,]  
 ### entire model-fitting process ###  
 fulldata.in = traindata.out # only input the data used to fit the model  
 k.in = 10   
 n.in = dim(fulldata.in)[1]  
 groups.in = c(rep(1:k.in,floor(n.in/k.in)),1:(n.in%%k.in)) #produces list of group labels  
 cvgroups.in = sample(groups.in,n.in) #orders randomly, with seed (8)  
 allpredictedCV.glm.in = matrix(rep(NA,n.in\*nmodels),ncol=nmodels)   
 for (i in 1:k.in) {  
 groupi.in = (cvgroups.in == i)  
 for (m in 1:nmodels) {  
 glmfitCV.in = glm(formula = allModels[[m]],data=redwine,subset=!groupi.in,family= "binomial")  
 allpredictedCV.glm.in[groupi.in,m] = predict(glmfitCV.in,fulldata.in[groupi.in,],type="response")  
 }  
 }  
 allmodelCV.glm.in = rep(NA,nmodels) #place-holder for results  
 for (m in 1:nmodels) {   
 allmodelCV.glm.in[m] = auc(fulldata.in$good,allpredictedCV.glm.in[,m])   
 }  
 bestmodel.in = (1:nmodels)[order(allmodelCV.glm.in)[11]] # actual selection  
 glmfitCV.out = glm(allModels[[bestmodel.in]],traindata.out,family= "binomial")  
 allpredictedCV.glm.out[groupj.out] = predict(glmfitCV.out,testdata.out,type="response")  
 wine.auc=auc(redwine$good,allpredictedCV.glm.out)  
 glm.best.model.list[j, ] = c(bestmodel.in,wine.auc)  
}

##### best model selection #####  
fulldata.selection = redwine  
k.selection = 10   
ggplot(glm.best.model.list,aes(glm.best.model.list$Model,glm.best.model.list$AUC))+geom\_point(colour = "red",size = 1)+ geom\_text(aes(label = paste(label=round(glm.best.model.list$AUC,3)), hjust=0.001, vjust=0.001),size = 3)+scale\_x\_continuous(breaks = seq(1,11,1)) +xlab(label = "Model")+ylab(label = "AUC")+theme(plot.background = element\_rect(fill = "lightblue"))+ggtitle("Model vs. Area Under Curve")+theme(plot.title = element\_text(hjust = 0.5))



glm.best.model.list.sort<-glm.best.model.list[order(glm.best.model.list$AUC),]  
glm.best.model.list.sort

## Model AUC  
## 3 8 0.8609706  
## 6 11 0.8644425  
## 4 8 0.8645456  
## 7 8 0.8655557  
## 10 9 0.8700708  
## 9 11 0.8715466  
## 5 8 0.8719526  
## 8 8 0.8743458  
## 2 10 0.8846655  
## 1 8 0.8916736

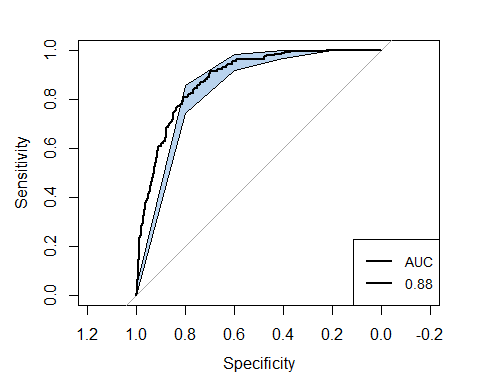
bestmodel.selection = glm.best.model.list.sort[10,1] # actual selection  
glm.fit=glm(formula = allModels[[bestmodel.selection]], data=fulldata.selection,family= "binomial")  
glm.fit.predictvals=predict(glm.fit,data=fulldata.selection)  
myroc=roc(response=redwine$good,predictor=glm.fit.predictvals)  
myroc

##   
## Call:  
## roc.default(response = redwine$good, predictor = glm.fit.predictvals)  
##   
## Data: glm.fit.predictvals in 1382 controls (redwine$good 0) < 217 cases (redwine$good 1).  
## Area under the curve: 0.882

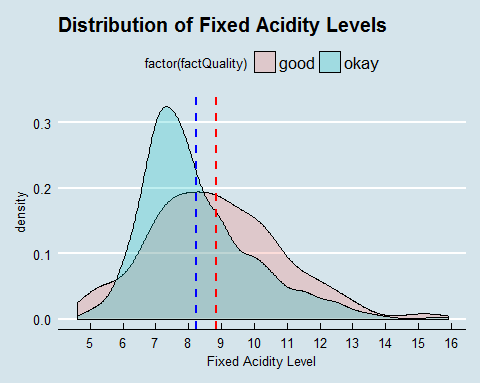
plot.roc.one<-plot.roc(myroc)  
roc.se.one <- ci.se(plot.roc.one,specificities=seq(0,1,0.2))  
plot(roc.se.one,type="shape", col="slategray2")

## Warning in plot.ci.se(roc.se.one, type = "shape", col = "slategray2"): Low  
## definition shape.

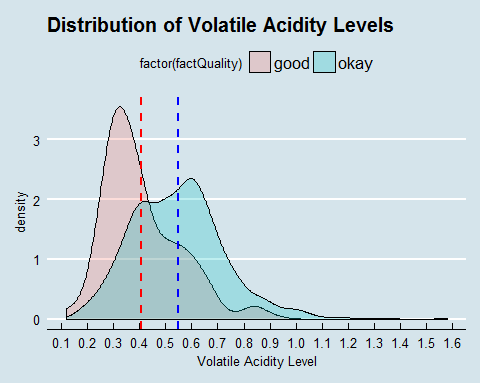
auc.wine<-auc(response=redwine$good,predictor=glm.fit.predictvals)  
legend("bottomright", legend=c("AUC",round(auc.wine,2)),cex=0.90,lty=1:1,lwd=2)



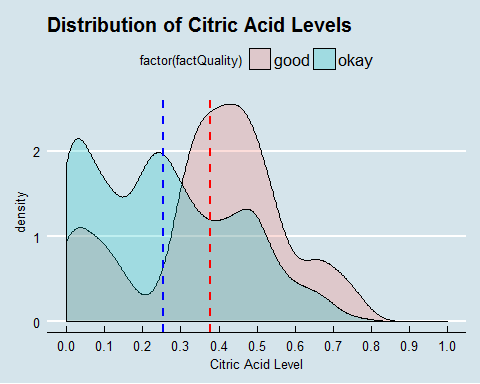
#Fixed Acidity and Wine Quality  
ggplot(redwine,aes(x=fixed.acidity,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(fixed.acidity[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(fixed.acidity[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(4,16,1))+  
 xlab(label = "Fixed Acidity Level")+  
 ggtitle("Distribution of Fixed Acidity Levels")+  
 theme\_economist()



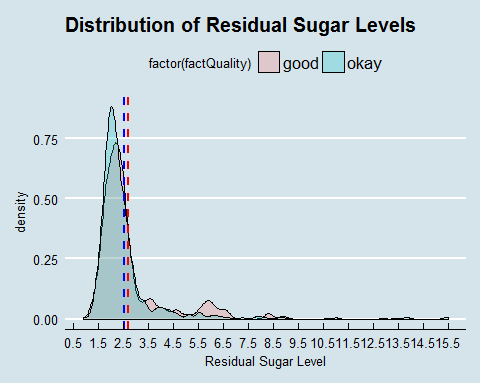
#Volatile Acidity and Wine Quality  
ggplot(redwine,aes(x=volatile.acidity,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(volatile.acidity[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(volatile.acidity[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(0,1.6,0.1))+  
 xlab(label = "Volatile Acidity Level")+  
 ggtitle("Distribution of Volatile Acidity Levels")+  
 theme\_economist()



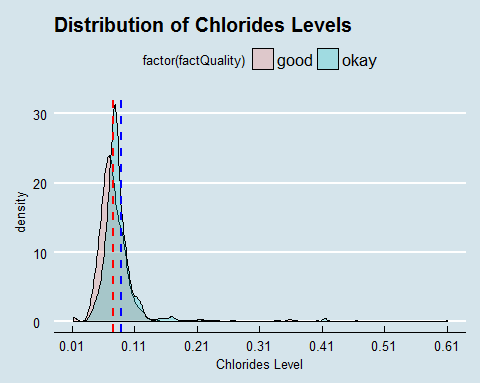
#Citric Acid and Wine Quality  
ggplot(redwine,aes(x=citric.acid,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(citric.acid[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(citric.acid[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(0,1,0.1))+  
 xlab(label = "Citric Acid Level")+  
 ggtitle("Distribution of Citric Acid Levels")+  
 theme\_economist()



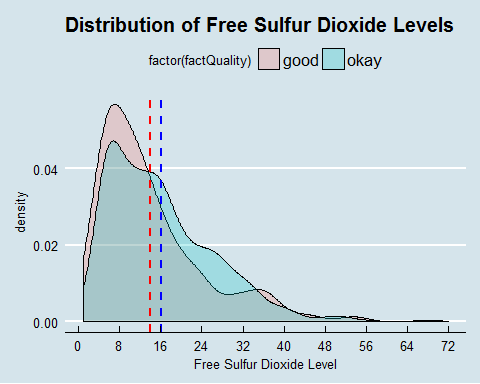
#Residual Sugar and Wine Quality  
ggplot(redwine,aes(x=residual.sugar,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(residual.sugar[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(residual.sugar[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(0.5,15.5,1))+  
 xlab(label = "Residual Sugar Level")+  
 ggtitle("Distribution of Residual Sugar Levels")+  
 theme\_economist()



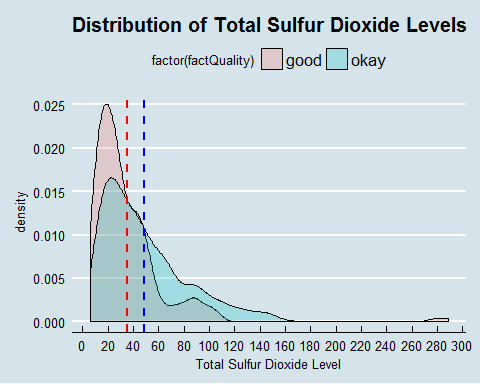
#Chlorides and Wine Quality  
ggplot(redwine,aes(x=chlorides,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(chlorides[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(chlorides[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(0.01,0.62,0.1))+  
 xlab(label = "Chlorides Level")+  
 ggtitle("Distribution of Chlorides Levels")+  
 theme\_economist()



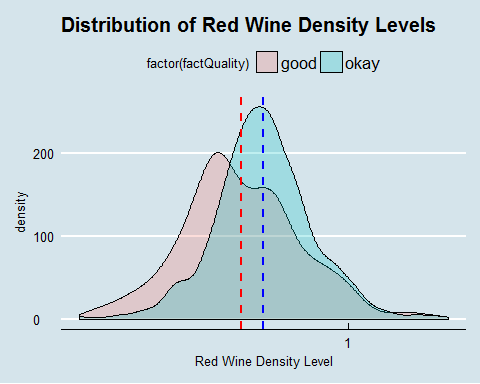
#Free Sulfur Dioxide and Wine Quality  
ggplot(redwine,aes(x=free.sulfur.dioxide,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(free.sulfur.dioxide[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(free.sulfur.dioxide[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(0,72,8))+  
 xlab(label = "Free Sulfur Dioxide Level")+  
 ggtitle("Distribution of Free Sulfur Dioxide Levels")+  
 theme\_economist()



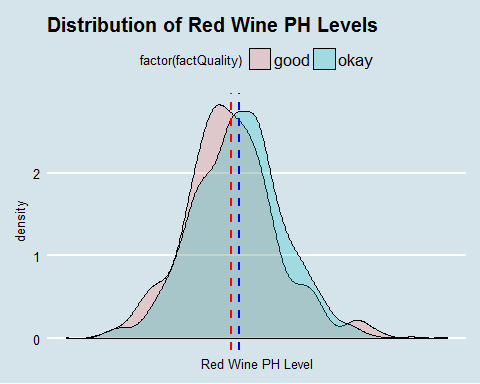
#Total Sulfur Dioxide and Wine Quality  
ggplot(redwine,aes(x=total.sulfur.dioxide,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(total.sulfur.dioxide[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(total.sulfur.dioxide[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(0,300,20))+  
 xlab(label = "Total Sulfur Dioxide Level")+  
 ggtitle("Distribution of Total Sulfur Dioxide Levels")+  
 theme\_economist()



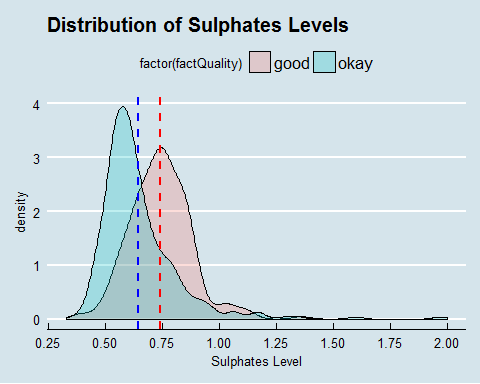
#Density and Wine Quality  
ggplot(redwine,aes(x=density,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(density[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(density[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(0.8,1.2,0.025))+  
 xlab(label = "Red Wine Density Level")+  
 ggtitle("Distribution of Red Wine Density Levels")+  
 theme\_economist()



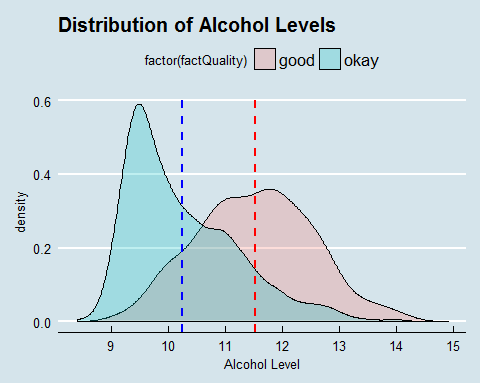
#PH and Wine Quality  
ggplot(redwine,aes(x=pH,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(pH[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(pH[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(0.8,1.2,0.025))+  
 xlab(label = "Red Wine PH Level")+  
 ggtitle("Distribution of Red Wine PH Levels")+  
 theme\_economist()



#Sulphates and Wine Quality  
ggplot(redwine,aes(x=sulphates,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(sulphates[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(sulphates[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(0,2,0.25))+  
 xlab(label = "Sulphates Level")+  
 ggtitle("Distribution of Sulphates Levels")+  
 theme\_economist()



#Alcohol and Wine Quality  
ggplot(redwine,aes(x=alcohol,fill=factor(factQuality)))+geom\_density(alpha=0.25)+  
 geom\_vline(aes(xintercept=mean(alcohol[factQuality=='good'],na.rm=T)),color="red",linetype="dashed",lwd=1)+  
 geom\_vline(aes(xintercept=mean(alcohol[factQuality=='okay'],na.rm=T)),color="blue",linetype="dashed",lwd=1)+  
 scale\_x\_continuous(breaks = seq(8,15,1))+  
 xlab(label = "Alcohol Level")+  
 ggtitle("Distribution of Alcohol Levels")+  
 theme\_economist()



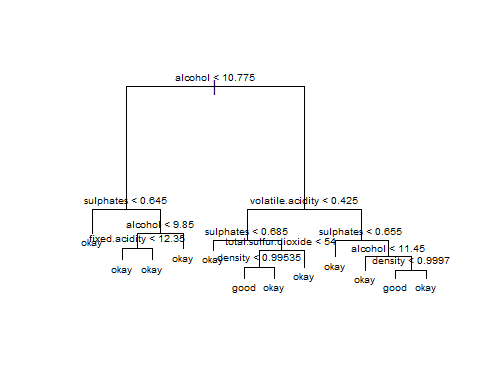
# Decesion tree with 10 cross validation  
  
prop.table(table(redwine$factQuality))

##   
## good okay   
## 0.1357098 0.8642902

# set the random seed equal to 8, and take a random sample of 1280 rows of the data.  
set.seed(8)  
train=sample(1:dim(redwine)[1],1280,replace=F)  
redwineTree.one <- tree(factQuality ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+pH+sulphates+alcohol+density, data=redwine[train,], method="class")  
#What is the error rate on the training set?  
summary(redwineTree.one)

##   
## Classification tree:  
## tree(formula = factQuality ~ fixed.acidity + volatile.acidity +   
## citric.acid + residual.sugar + chlorides + free.sulfur.dioxide +   
## total.sulfur.dioxide + pH + sulphates + alcohol + density,   
## data = redwine[train, ], method = "class")  
## Variables actually used in tree construction:  
## [1] "alcohol" "sulphates" "fixed.acidity"   
## [4] "volatile.acidity" "total.sulfur.dioxide" "density"   
## Number of terminal nodes: 12   
## Residual mean deviance: 0.4984 = 632 / 1268   
## Misclassification error rate: 0.1023 = 131 / 1280

#Plot the decision tree with category labels.  
plot(redwineTree.one)  
text(redwineTree.one, cex=0.6)



#Compute the confusion matrix for the validation set. What is the validation set accuracy rate?  
Treefit.one=predict(redwineTree.one,redwine[-train,],type="class")  
tree.table.one<-table(Treefit.one,redwine$factQuality[-train])  
tree.table.one

##   
## Treefit.one good okay  
## good 14 11  
## okay 33 261

tree.accuracy.one<-(tree.table.one[1,1]+tree.table.one[2,2])/sum(tree.table.one)  
tree.accuracy.one

## [1] 0.862069

#Use 10-fold cross-validation on the training data to choose the number of leaves that minimizes the classification error rate and find out the optimal #numbers of leaves  
redwineTree.one.cv=cv.tree(redwineTree.one,FUN=prune.misclass)  
redwineTree.one.cv$size[which(redwineTree.one.cv$dev==min(redwineTree.one.cv$dev))]

## [1] 8 5

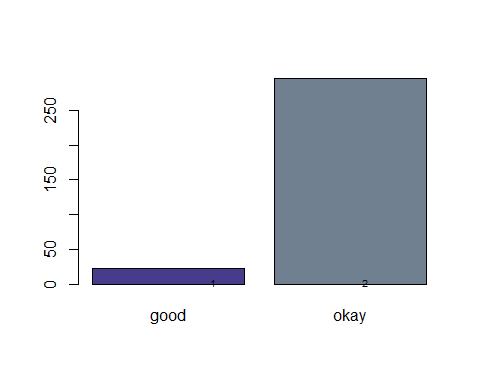
#Create a pruned tree with 5 leaves. What is the error rate of the pruned tree on the validation set?  
prune.redwineTree=prune.misclass(redwineTree.one,best=5)  
summary(prune.redwineTree)

##   
## Classification tree:  
## snip.tree(tree = redwineTree.one, nodes = c(2L, 26L, 7L))  
## Variables actually used in tree construction:  
## [1] "alcohol" "volatile.acidity" "sulphates"   
## [4] "total.sulfur.dioxide"  
## Number of terminal nodes: 5   
## Residual mean deviance: 0.5985 = 763.1 / 1275   
## Misclassification error rate: 0.1094 = 140 / 1280

prune.item.pred.one=predict(prune.redwineTree,redwine[-train,],type="class")  
prune.tree.table.one<-table(prune.item.pred.one,redwine$factQuality[-train])  
prune.tree.table.one

##   
## prune.item.pred.one good okay  
## good 16 7  
## okay 31 265

plot(prune.item.pred.one,col=c("slateblue4", "slategray"))  
text(prune.item.pred.one,cex=0.6)



tree.accuracy.two<-(prune.tree.table.one[1,1]+prune.tree.table.one[2,2])/sum(prune.tree.table.one)  
tree.accuracy.two

## [1] 0.8808777

# Random Forest to find importance of variables for model creation  
  
redwine.RF<-randomForest(factQuality ~ ., data=redwine[,-12:-13], importance=T, proximity=T)  
redwine.RF.pred <- predict(redwine.RF, redwine, type="class")  
table(redwine.RF.pred, redwine$factQuality)

##   
## redwine.RF.pred good okay  
## good 217 0  
## okay 0 1382

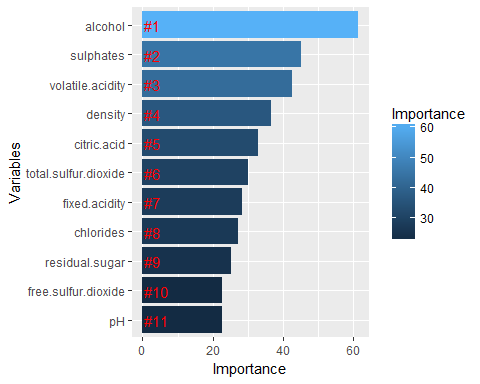
# Get importance of predicted variables   
redwine.importance <- importance(redwine.RF)  
var.Importance <- data.frame(Variables = row.names(redwine.importance), Importance = round(redwine.importance[ ,'MeanDecreaseGini'],2))  
var.Importance

## Variables Importance  
## fixed.acidity fixed.acidity 28.31  
## volatile.acidity volatile.acidity 42.65  
## citric.acid citric.acid 32.86  
## residual.sugar residual.sugar 25.15  
## chlorides chlorides 27.36  
## free.sulfur.dioxide free.sulfur.dioxide 22.76  
## total.sulfur.dioxide total.sulfur.dioxide 30.08  
## density density 36.60  
## pH pH 22.69  
## sulphates sulphates 45.17  
## alcohol alcohol 61.53

# Create a rank variable based on importance of predicted variables  
rank.Importance <- var.Importance %>%mutate(Rank = paste0('#',dense\_rank(desc(Importance))))  
rank.Importance

## Variables Importance Rank  
## 1 fixed.acidity 28.31 #7  
## 2 volatile.acidity 42.65 #3  
## 3 citric.acid 32.86 #5  
## 4 residual.sugar 25.15 #9  
## 5 chlorides 27.36 #8  
## 6 free.sulfur.dioxide 22.76 #10  
## 7 total.sulfur.dioxide 30.08 #6  
## 8 density 36.60 #4  
## 9 pH 22.69 #11  
## 10 sulphates 45.17 #2  
## 11 alcohol 61.53 #1

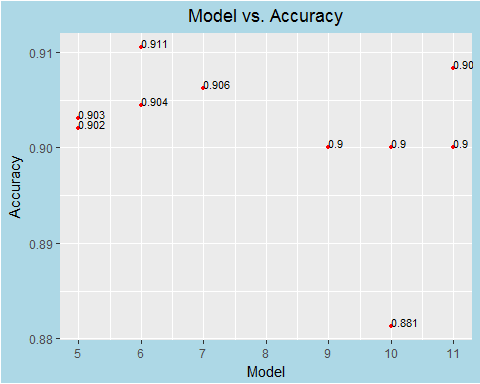
# Use ggplot2 to visualize the relative importance of all predict variables  
ggplot(rank.Importance, aes(x = reorder(Variables, Importance), y = Importance, fill = Importance)) +geom\_bar(stat='identity') + geom\_text(aes(x = Variables, y = 0.5, label = Rank),hjust=0, vjust=0.55, size = 4, colour = 'red') +labs(x = 'Variables') + coord\_flip() + theme\_grey()



#Rendom Forest model list specification based on importance of predict variables  
  
rm.models = 11  
Model1 = (factQuality ~ alcohol)  
Model2 = (factQuality ~ alcohol + sulphates)  
Model3 = (factQuality ~ alcohol + sulphates + volatile.acidity )  
Model4 = (factQuality ~ alcohol + sulphates + volatile.acidity + density)  
Model5 = (factQuality ~ alcohol + sulphates + volatile.acidity + density + citric.acid)  
Model6 = (factQuality ~ alcohol + sulphates + volatile.acidity + density + citric.acid + total.sulfur.dioxide)  
Model7 = (factQuality ~ alcohol + sulphates + volatile.acidity + density + citric.acid + total.sulfur.dioxide + fixed.acidity)  
Model8 = (factQuality ~ alcohol + sulphates + volatile.acidity + density + citric.acid + total.sulfur.dioxide + fixed.acidity + chlorides)  
Model9 = (factQuality ~ alcohol + sulphates + volatile.acidity + density + citric.acid + total.sulfur.dioxide + fixed.acidity + chlorides + residual.sugar)  
Model10 = (factQuality ~ alcohol + sulphates + volatile.acidity + density + citric.acid + total.sulfur.dioxide + fixed.acidity + chlorides + residual.sugar + free.sulfur.dioxide)  
Model11 = (factQuality ~ alcohol + sulphates + volatile.acidity + density + citric.acid + total.sulfur.dioxide + fixed.acidity + chlorides + residual.sugar + free.sulfur.dioxide + pH)  
allModels = list(Model1,Model2,Model3,Model4,Model5,Model6,Model7,Model8,Model9,Model10,Model11)

##### Random Forest model assessment OUTER CV (with model selection INNER CV as part of model-fitting) #####  
  
fulldata.out = redwine  
k.out = 10   
n.out = dim(fulldata.out)[1]  
#define the cross-validation splits   
groups.out = c(rep(1:k.out,floor(n.out/k.out)),1:(n.out%%k.out)) #produces list of group labels  
set.seed(8)  
cvgroups.out = sample(groups.out,n.out) #orders randomly, with seed (8)   
allpredictedCV.rf.out = rep(NA,n.out)  
rf.best.model.list = data.frame( Model=rep(0, k.out), Accuracy=rep(0,k.out))  
for (j in 1:k.out) { #be careful not to re-use loop indices  
 groupj.out = (cvgroups.out == j)  
 traindata.out = redwine[!groupj.out,]  
 testdata.out = redwine[groupj.out,]  
 ### entire model-fitting process ###  
 fulldata.in = traindata.out # only input the data used to fit the model  
 k.in = 10   
 n.in = dim(fulldata.in)[1]  
 groups.in = c(rep(1:k.in,floor(n.in/k.in)),1:(n.in%%k.in)) #produces list of group labels  
 cvgroups.in = sample(groups.in,n.in) #orders randomly, with seed (8)  
 allpredictedCV.rf.in = matrix(rep(NA,n.in\*rm.models),ncol=rm.models)   
 for (i in 1:k.in) {  
 groupi.in = (cvgroups.in == i)  
 for (m in 1:rm.models) {  
 rffitCV.in = randomForest(formula = allModels[[m]], data=redwine,subset=!groupi.in,mtry = m,importance = T)  
 allpredictedCV.rf.in[groupi.in,m] = predict(rffitCV.in, newdata=fulldata.in[groupi.in,],mtry = m,type="response")  
 }  
 }  
 allmodelCV.rf.in = rep(NA,rm.models) #place-holder for results  
 for (m in 1:rm.models) {   
 rf.table.in<-table(fulldata.in$factQuality,allpredictedCV.rf.in[,m])  
 accuracy.in<-(rf.table.in[1,1]+rf.table.in[2,2])/sum(rf.table.in)  
 allmodelCV.rf.in[m] = accuracy.in   
 }  
 bestmodel.in = (1:rm.models)[order(allmodelCV.rf.in)[11]] # actual selection  
 rffitCV.out = randomForest(allModels[[bestmodel.in]],traindata.out,mtry = bestmodel.in,importance = T)  
 allpredictedCV.rf.out[groupj.out] = predict(rffitCV.out,testdata.out,mtry = bestmodel.in,type="response")  
 rf.table.out<-table(redwine$factQuality,allpredictedCV.rf.out)  
 accuracy.out<-(rf.table.out[1,1]+rf.table.out[2,2])/sum(rf.table.out)  
 rf.best.model.list[j, ] = c(bestmodel.in,accuracy.out)  
}

##### best model selection #####  
fulldata.selection = redwine  
k.selection = 10   
ggplot(rf.best.model.list,aes(rf.best.model.list$Model,rf.best.model.list$Accuracy))+geom\_point(colour = "red",size = 1)+ geom\_text(aes(label = paste(label=round(rf.best.model.list$Accuracy,3)), hjust=0.001, vjust=0.001),size = 3)+scale\_x\_continuous(breaks = seq(1,11,1)) +xlab(label = "Model")+ylab(label = "Accuracy")+theme(plot.background = element\_rect(fill = "lightblue"))+ggtitle("Model vs. Accuracy")+theme(plot.title = element\_text(hjust = 0.5))



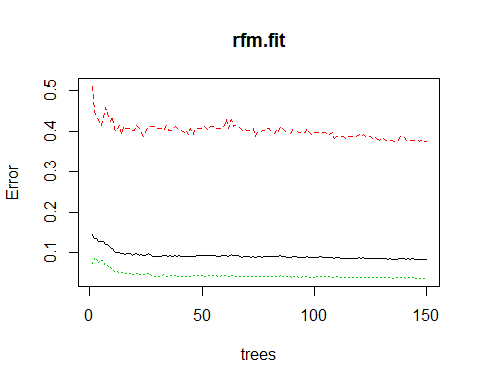
rf.best.model.list.sort<-rf.best.model.list[order(rf.best.model.list$Accuracy),]  
rf.best.model.list.sort

## Model Accuracy  
## 1 10 0.8812500  
## 3 9 0.9000000  
## 4 11 0.9000000  
## 5 10 0.9000000  
## 6 5 0.9020833  
## 2 5 0.9031250  
## 7 6 0.9044643  
## 8 7 0.9062500  
## 9 11 0.9083333  
## 10 6 0.9105691

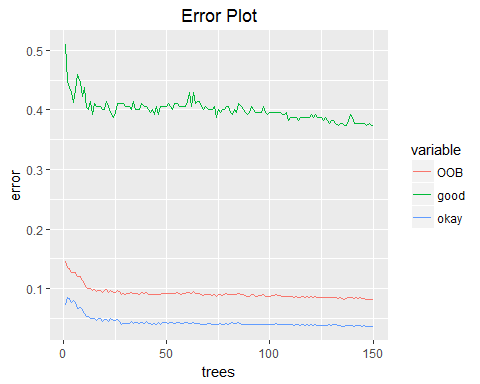
bestmodel.selection = rf.best.model.list.sort[10,1] # actual selection  
rfm.fit=randomForest(formula = allModels[[bestmodel.selection]], data=fulldata.selection,mtry = bestmodel.selection,ntree = 150,importance = T)  
rfm.fit.predictvals=predict(rfm.fit,data=fulldata.selection,mtry = m,type="response")  
rfm.fit.table<-table(rfm.fit.predictvals, redwine$factQuality)  
rfm.fit.table

##   
## rfm.fit.predictvals good okay  
## good 136 51  
## okay 81 1331

# Get OOB data from plot and coerce to data.table  
oobData = as.data.table(plot(rfm.fit))



# Define trees as 1:ntree  
oobData[, trees := .I]  
# Cast to long format  
oobData2 = melt(oobData, id.vars = "trees")  
setnames(oobData2, "value", "error")  
# Plot using ggplot  
ggplot(data = oobData2, aes(x = trees, y = error, color = variable)) + geom\_line()+ggtitle("Error Plot")+theme(plot.title = element\_text(hjust = 0.5))



rfm.fit.accuracy<-(rfm.fit.table[1,1]+rfm.fit.table[2,2])/sum(rfm.fit.table)  
# Get importance of all predict variables   
rfm.fit.importance <- importance(rfm.fit)  
rfm.fit.var.Importance <- data.frame(Variables = row.names(rfm.fit.importance), Importance = round(rfm.fit.importance[ ,'MeanDecreaseGini'],2))  
# Create a rank variable based on importance of all predict variable (11)  
rfm.fit.rank.Importance <- rfm.fit.var.Importance %>%mutate(Rank = paste0('#',dense\_rank(desc(Importance))))  
# Use ggplot2 to visualize the relative importance of all predict variables  
ggplot(rfm.fit.rank.Importance, aes(x = reorder(Variables, Importance), y = Importance, fill = Importance)) +geom\_bar(stat='identity') + geom\_text(aes(x = Variables, y = 0.5, label = Rank),hjust=0,vjust=0.55, size = 4, colour = 'red') +labs(x = 'Variables') + coord\_flip() + theme\_grey()

