Stat\_fp\_white.R

pravallikaavula

2022-12-08

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 0.3.5   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(dplyr)  
library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(corrplot)

## corrplot 0.92 loaded

library(rpart)  
library(rpart.plot)  
library(DMwR2) # Contains rt.prune

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(ISLR)   
library(MASS)

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select

library(randomForest)

## randomForest 4.7-1.1  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine  
##   
## The following object is masked from 'package:ggplot2':  
##   
## margin

library(cluster) # clustering algorithms  
library(factoextra) # clustering algorithms & visualization

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(maditr)

##   
## To select columns from data: columns(mtcars, mpg, vs:carb)  
##   
##   
## Attaching package: 'maditr'  
##   
## The following objects are masked from 'package:dplyr':  
##   
## between, coalesce, first, last  
##   
## The following object is masked from 'package:purrr':  
##   
## transpose  
##   
## The following object is masked from 'package:readr':  
##   
## cols

library(ggpubr)  
library(Metrics)  
library(glmnet)

## Loading required package: Matrix  
##   
## Attaching package: 'Matrix'  
##   
## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack  
##   
## Loaded glmnet 4.1-4

library(mlbench)  
library(caret) # use createDataPartition() function

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following objects are masked from 'package:Metrics':  
##   
## precision, recall  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(gridExtra)

##   
## Attaching package: 'gridExtra'  
##   
## The following object is masked from 'package:randomForest':  
##   
## combine  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine

#Data reading  
white\_df <- read.csv(file = "winequality-white.csv", as.is = TRUE,sep = ";", header = TRUE)  
head(white\_df)

## fixed.acidity volatile.acidity citric.acid residual.sugar chlorides  
## 1 7.0 0.27 0.36 20.7 0.045  
## 2 6.3 0.30 0.34 1.6 0.049  
## 3 8.1 0.28 0.40 6.9 0.050  
## 4 7.2 0.23 0.32 8.5 0.058  
## 5 7.2 0.23 0.32 8.5 0.058  
## 6 8.1 0.28 0.40 6.9 0.050  
## free.sulfur.dioxide total.sulfur.dioxide density pH sulphates alcohol  
## 1 45 170 1.0010 3.00 0.45 8.8  
## 2 14 132 0.9940 3.30 0.49 9.5  
## 3 30 97 0.9951 3.26 0.44 10.1  
## 4 47 186 0.9956 3.19 0.40 9.9  
## 5 47 186 0.9956 3.19 0.40 9.9  
## 6 30 97 0.9951 3.26 0.44 10.1  
## quality  
## 1 6  
## 2 6  
## 3 6  
## 4 6  
## 5 6  
## 6 6

dim(white\_df)

## [1] 4898 12

#Data overview  
str(white\_df)

## 'data.frame': 4898 obs. of 12 variables:  
## $ fixed.acidity : num 7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...  
## $ volatile.acidity : num 0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...  
## $ citric.acid : num 0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...  
## $ residual.sugar : num 20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...  
## $ chlorides : num 0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...  
## $ free.sulfur.dioxide : num 45 14 30 47 47 30 30 45 14 28 ...  
## $ total.sulfur.dioxide: num 170 132 97 186 186 97 136 170 132 129 ...  
## $ density : num 1.001 0.994 0.995 0.996 0.996 ...  
## $ pH : num 3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...  
## $ sulphates : num 0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...  
## $ alcohol : num 8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...  
## $ quality : int 6 6 6 6 6 6 6 6 6 6 ...

summary(white\_df)

## fixed.acidity volatile.acidity citric.acid residual.sugar   
## Min. : 3.800 Min. :0.0800 Min. :0.0000 Min. : 0.600   
## 1st Qu.: 6.300 1st Qu.:0.2100 1st Qu.:0.2700 1st Qu.: 1.700   
## Median : 6.800 Median :0.2600 Median :0.3200 Median : 5.200   
## Mean : 6.855 Mean :0.2782 Mean :0.3342 Mean : 6.391   
## 3rd Qu.: 7.300 3rd Qu.:0.3200 3rd Qu.:0.3900 3rd Qu.: 9.900   
## Max. :14.200 Max. :1.1000 Max. :1.6600 Max. :65.800   
## chlorides free.sulfur.dioxide total.sulfur.dioxide density   
## Min. :0.00900 Min. : 2.00 Min. : 9.0 Min. :0.9871   
## 1st Qu.:0.03600 1st Qu.: 23.00 1st Qu.:108.0 1st Qu.:0.9917   
## Median :0.04300 Median : 34.00 Median :134.0 Median :0.9937   
## Mean :0.04577 Mean : 35.31 Mean :138.4 Mean :0.9940   
## 3rd Qu.:0.05000 3rd Qu.: 46.00 3rd Qu.:167.0 3rd Qu.:0.9961   
## Max. :0.34600 Max. :289.00 Max. :440.0 Max. :1.0390   
## pH sulphates alcohol quality   
## Min. :2.720 Min. :0.2200 Min. : 8.00 Min. :3.000   
## 1st Qu.:3.090 1st Qu.:0.4100 1st Qu.: 9.50 1st Qu.:5.000   
## Median :3.180 Median :0.4700 Median :10.40 Median :6.000   
## Mean :3.188 Mean :0.4898 Mean :10.51 Mean :5.878   
## 3rd Qu.:3.280 3rd Qu.:0.5500 3rd Qu.:11.40 3rd Qu.:6.000   
## Max. :3.820 Max. :1.0800 Max. :14.20 Max. :9.000

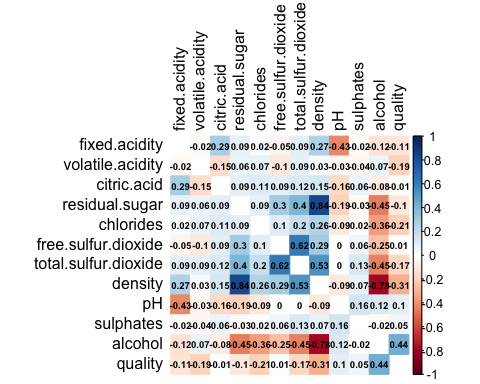
#Checking missing values  
which(is.na(white\_df))

## integer(0)

colSums(is.na(white\_df))

## fixed.acidity volatile.acidity citric.acid   
## 0 0 0   
## residual.sugar chlorides free.sulfur.dioxide   
## 0 0 0   
## total.sulfur.dioxide density pH   
## 0 0 0   
## sulphates alcohol quality   
## 0 0 0

#Build correlation and order by decreasing  
set.seed(123)  
  
white\_dfcor <- cor(white\_df)  
corrplot(white\_dfcor, method = "color", addCoef.col = "black",number.cex = .6,  
 tl.col = "black", tl.srt = 90, diag = FALSE)



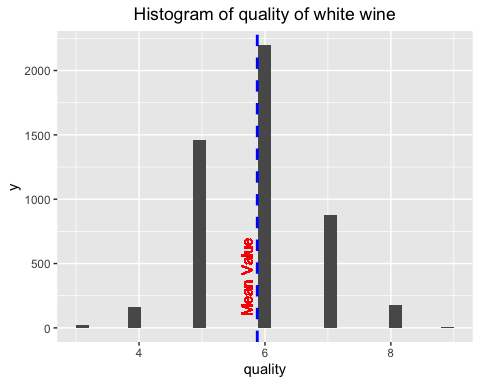
#Build correlation atts with Quality variable  
dfcor <- cor(white\_df)  
quality\_cor <- dfcor[,12]  
absoutcome\_cor <- abs(quality\_cor)  
head(absoutcome\_cor[order(absoutcome\_cor, decreasing = TRUE)],12)

## quality alcohol density   
## 1.000000000 0.435574715 0.307123313   
## chlorides volatile.acidity total.sulfur.dioxide   
## 0.209934411 0.194722969 0.174737218   
## fixed.acidity pH residual.sugar   
## 0.113662831 0.099427246 0.097576829   
## sulphates citric.acid free.sulfur.dioxide   
## 0.053677877 0.009209091 0.008158067

#Analysing the overall quality  
ggplot(white\_df, aes(quality))+   
 geom\_histogram() +   
 labs(title = "Histogram of quality of white wine") +   
 theme(plot.title=element\_text(hjust=0.5)) +  
 geom\_vline(aes(xintercept=mean(quality)), color="blue", linetype="dashed", size=1) +  
 geom\_text(aes(x=5.6, label="Mean Value", y=400), colour="red", angle=90, vjust = 1.2)

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.

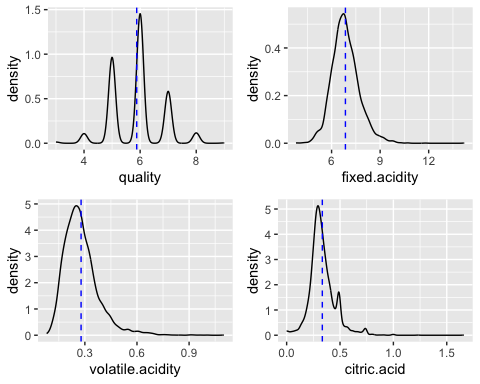
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



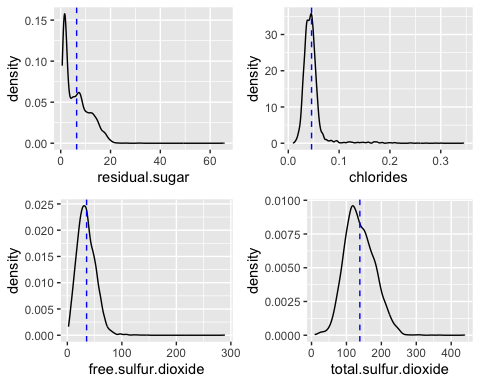
mean(white\_df$quality)

## [1] 5.877909

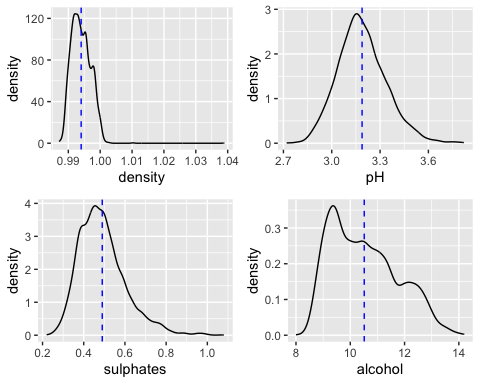
#Plotting the marginal distributions of numerical quantities of interest using density plots  
  
d0 <- ggplot(white\_df, aes(x=quality)) +   
 geom\_density()  
d0 <- d0 + geom\_vline(aes(xintercept=mean(quality)),  
 color="blue", linetype="dashed")  
d1 <- ggplot(white\_df, aes(x=fixed.acidity)) +   
 geom\_density()  
d1 <- d1 + geom\_vline(aes(xintercept=mean(fixed.acidity)),  
 color="blue", linetype="dashed")  
d2 <- ggplot(white\_df, aes(x=volatile.acidity)) +   
 geom\_density()  
d2 <- d2 + geom\_vline(aes(xintercept=mean(volatile.acidity)),  
 color="blue", linetype="dashed")  
d3 <- ggplot(white\_df, aes(x=citric.acid)) +   
 geom\_density()  
d3 <- d3 + geom\_vline(aes(xintercept=mean(citric.acid)),  
 color="blue", linetype="dashed")  
d4 <- ggplot(white\_df, aes(x=residual.sugar)) +   
 geom\_density()  
d4 <- d4 + geom\_vline(aes(xintercept=mean(residual.sugar)),  
 color="blue", linetype="dashed")  
d5 <- ggplot(white\_df, aes(x=chlorides)) +   
 geom\_density()  
d5 <- d5 + geom\_vline(aes(xintercept=mean(chlorides)),  
 color="blue", linetype="dashed")  
d6 <- ggplot(white\_df, aes(x=free.sulfur.dioxide)) +   
 geom\_density()  
d6 <- d6 + geom\_vline(aes(xintercept=mean(free.sulfur.dioxide)),  
 color="blue", linetype="dashed")  
d7 <- ggplot(white\_df, aes(x=total.sulfur.dioxide)) +   
 geom\_density()  
d7 <- d7 + geom\_vline(aes(xintercept=mean(total.sulfur.dioxide)),  
 color="blue", linetype="dashed")  
d8 <- ggplot(white\_df, aes(x=density)) +   
 geom\_density()  
d8 <- d8 + geom\_vline(aes(xintercept=mean(density)),  
 color="blue", linetype="dashed")  
d9 <- ggplot(white\_df, aes(x=pH)) +   
 geom\_density()  
d9 <- d9 + geom\_vline(aes(xintercept=mean(pH)),  
 color="blue", linetype="dashed")  
d10 <- ggplot(white\_df, aes(x=sulphates)) +   
 geom\_density()  
d10 <- d10 + geom\_vline(aes(xintercept=mean(sulphates)),  
 color="blue", linetype="dashed")  
d11 <- ggplot(white\_df, aes(x=alcohol)) +   
 geom\_density()  
d11 <- d11 + geom\_vline(aes(xintercept=mean(alcohol)),  
 color="blue", linetype="dashed")  
ggarrange(d0, d1, d2, d3, nrow = 2, ncol =2)



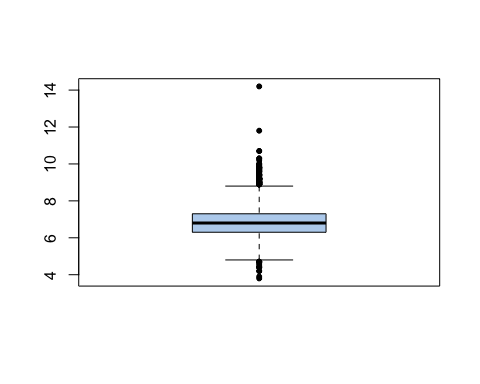
ggarrange(d4, d5, d6, d7, nrow = 2, ncol =2)



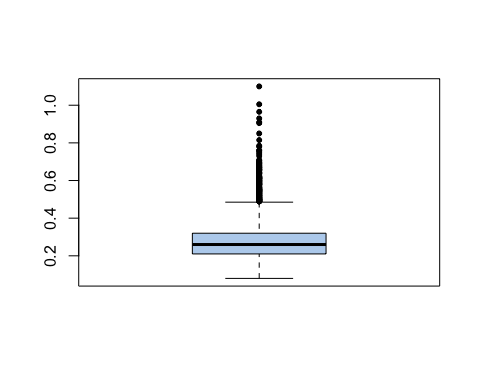
ggarrange(d8, d9, d10, d11, nrow = 2, ncol =2)



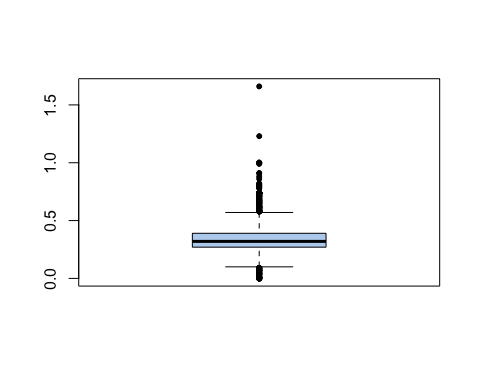
#Plotting the marginal distributions of numerical quantities of interest using box plots  
b1 <- boxplot(white\_df$fixed.acidity, col="slategray2", pch=20)



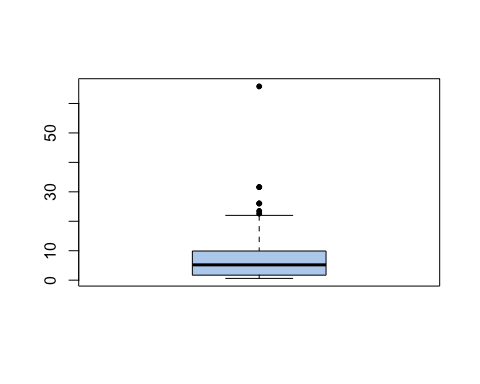
b2 <- boxplot(white\_df$volatile.acidity, col="slategray2", pch=20)



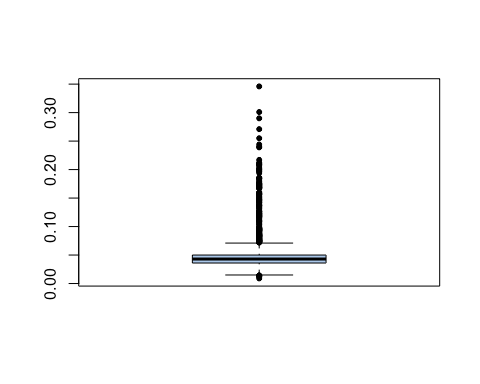
b3 <- boxplot(white\_df$citric.acid, col="slategray2", pch=20)



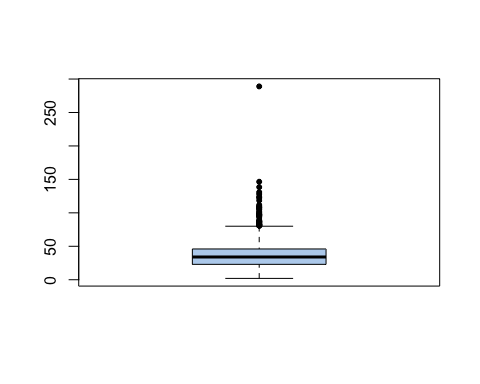
b4 <- boxplot(white\_df$residual.sugar, col="slategray2", pch=20)



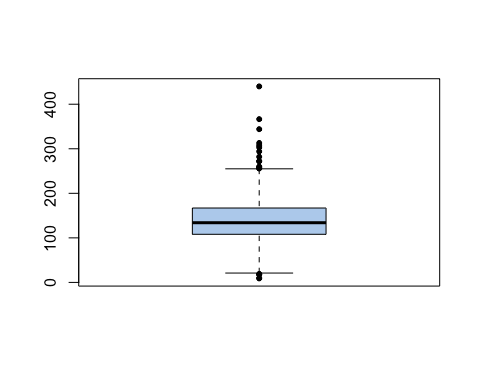
b5 <- boxplot(white\_df$chlorides, col="slategray2", pch=20)



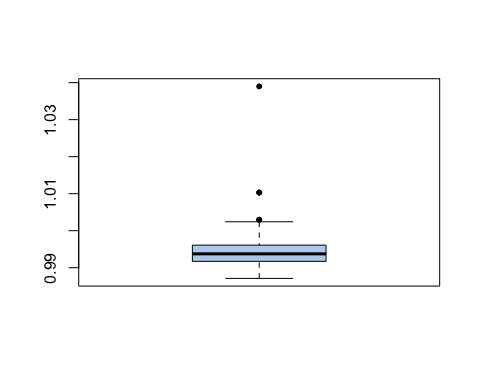
b6 <- boxplot(white\_df$free.sulfur.dioxide, col="slategray2", pch=20)



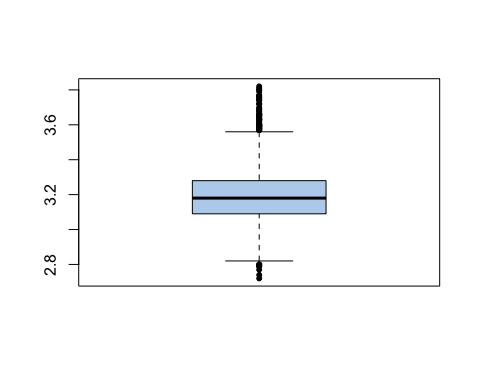
b7 <- boxplot(white\_df$total.sulfur.dioxide, col="slategray2", pch=20)



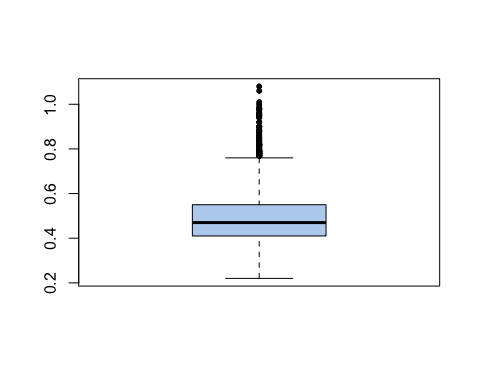
b8 <- boxplot(white\_df$density, col="slategray2", pch=20)



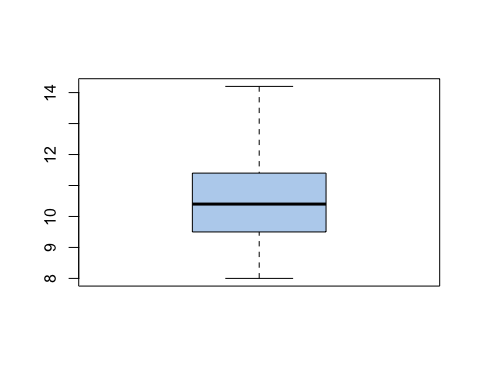
b9 <- boxplot(white\_df$pH, col="slategray2", pch=20)



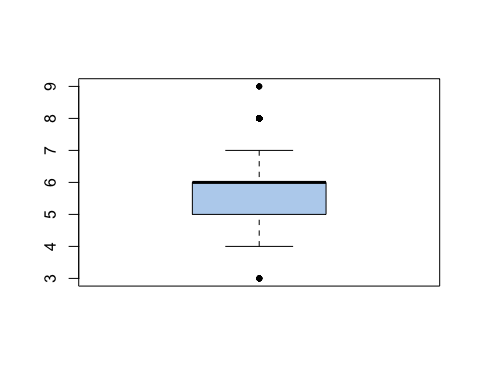
b10 <- boxplot(white\_df$sulphates, col="slategray2", pch=20)



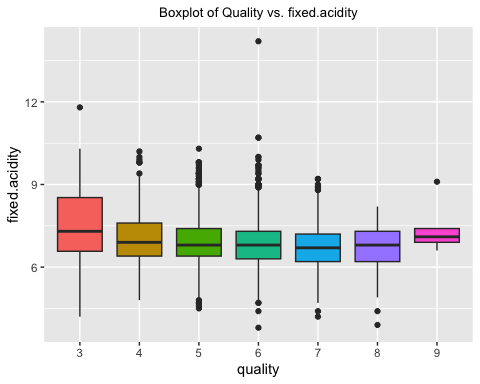
b11 <- boxplot(white\_df$alcohol, col="slategray2", pch=20)



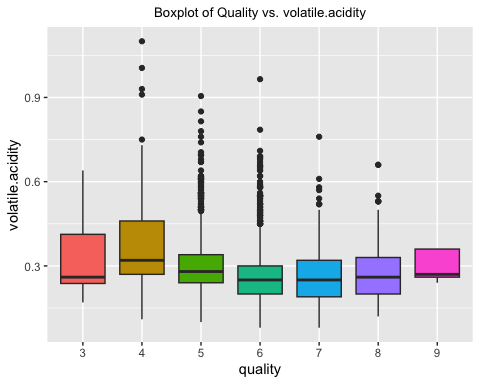
b12 <- boxplot(white\_df$quality, col="slategray2", pch=20)



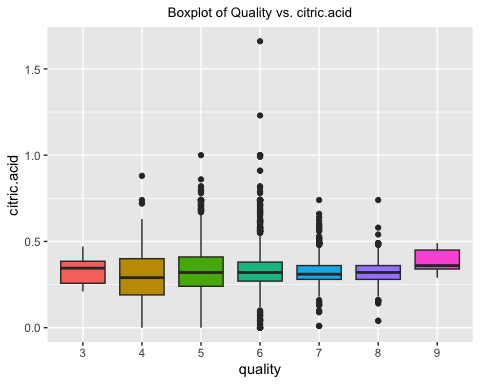
#Plotting quality against numerical variables  
bp1 <- ggplot(white\_df, aes(factor(quality), fixed.acidity, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "quality", y = "fixed.acidity", title = "Boxplot of Quality vs. fixed.acidity") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp1



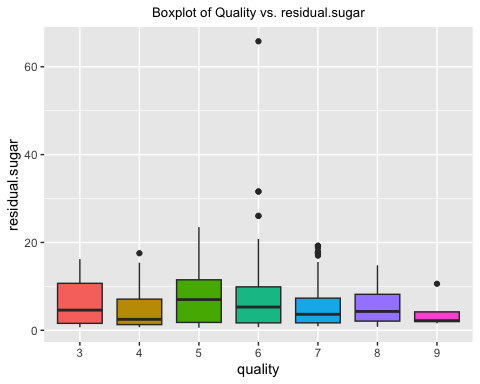
bp2 <- ggplot(white\_df, aes(factor(quality), volatile.acidity, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "quality", y = "volatile.acidity", title = "Boxplot of Quality vs. volatile.acidity") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp2



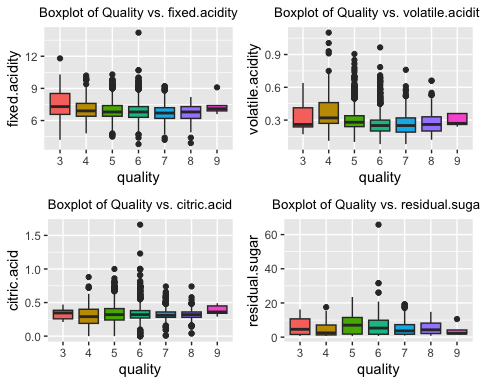
bp3 <- ggplot(white\_df, aes(factor(quality), citric.acid, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "quality", y = "citric.acid", title = "Boxplot of Quality vs. citric.acid") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp3



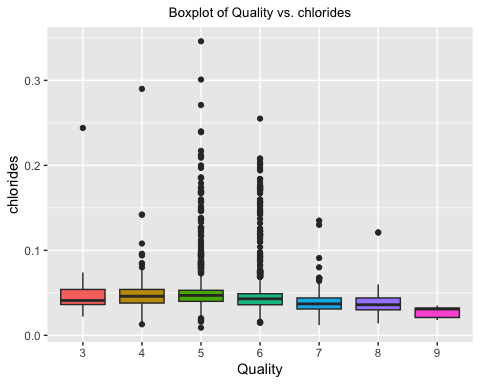
bp4 <- ggplot(white\_df, aes(factor(quality), residual.sugar, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "quality", y = "residual.sugar", title = "Boxplot of Quality vs. residual.sugar") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp4



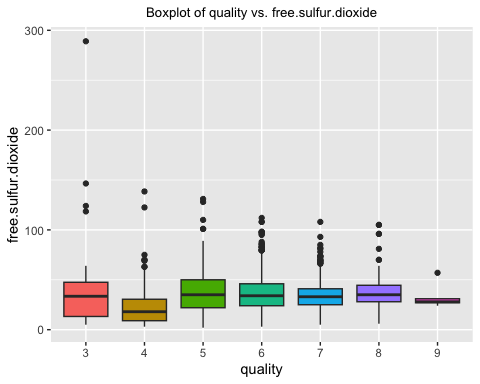
ggarrange(bp1, bp2, bp3, bp4, nrow = 2, ncol =2)



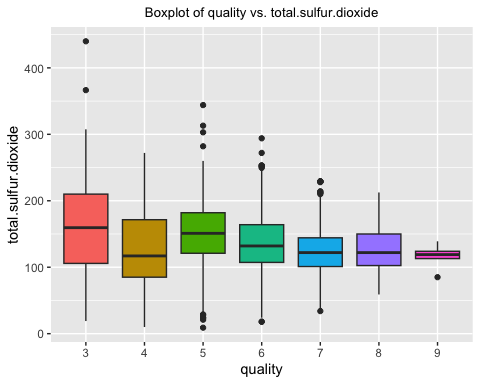
bp5 <- ggplot(white\_df, aes(factor(quality), chlorides, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "Quality", y = "chlorides", title = "Boxplot of Quality vs. chlorides") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp5



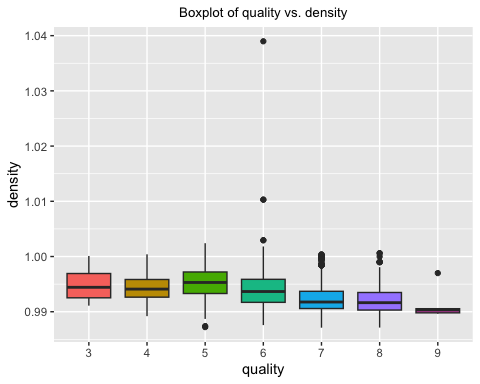
bp6 <- ggplot(white\_df, aes(factor(quality), free.sulfur.dioxide, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "quality", y = "free.sulfur.dioxide", title = "Boxplot of quality vs. free.sulfur.dioxide") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp6



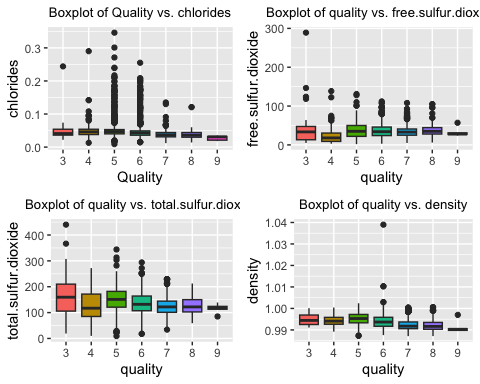
bp7 <- ggplot(white\_df, aes(factor(quality), total.sulfur.dioxide, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "quality", y = "total.sulfur.dioxide", title = "Boxplot of quality vs. total.sulfur.dioxide") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp7



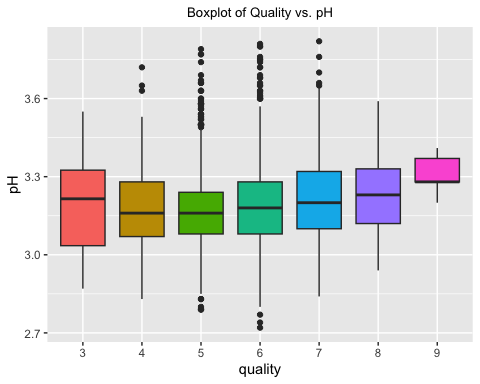
bp8 <- ggplot(white\_df, aes(factor(quality), density, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "quality", y = "density", title = "Boxplot of quality vs. density") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp8



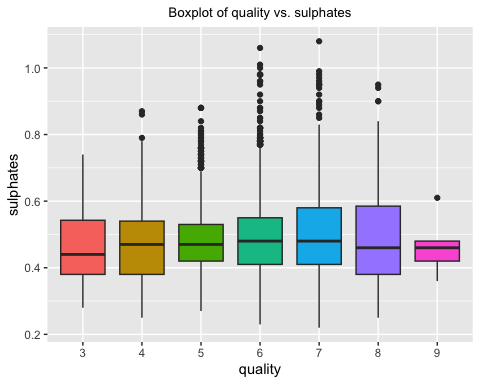
ggarrange(bp5, bp6, bp7, bp8, nrow = 2, ncol =2)



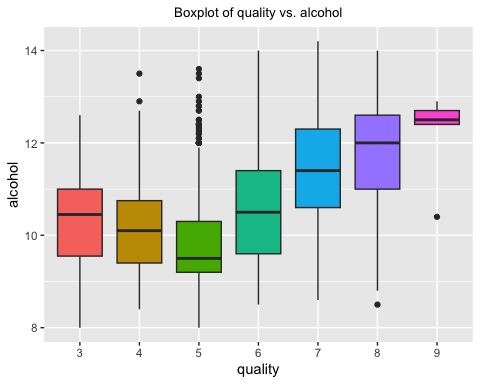
bp9 <- ggplot(white\_df, aes(factor(quality), pH, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "quality", y = "pH", title = "Boxplot of Quality vs. pH") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp9



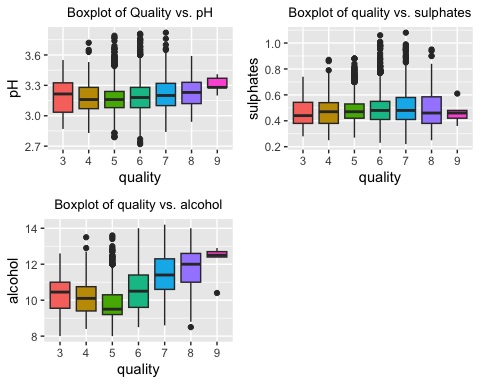
bp10 <- ggplot(white\_df, aes(factor(quality), sulphates, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "quality", y = "sulphates", title = "Boxplot of quality vs. sulphates") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp10



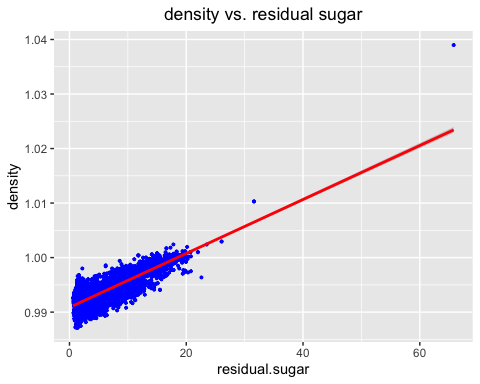
bp11 <- ggplot(white\_df, aes(factor(quality), alcohol, fill=factor(quality))) +   
 geom\_boxplot() +  
 labs(x = "quality", y = "alcohol", title = "Boxplot of quality vs. alcohol") +   
 theme(legend.position = 'none', plot.title = element\_text(size = 10, hjust=0.5))  
bp11



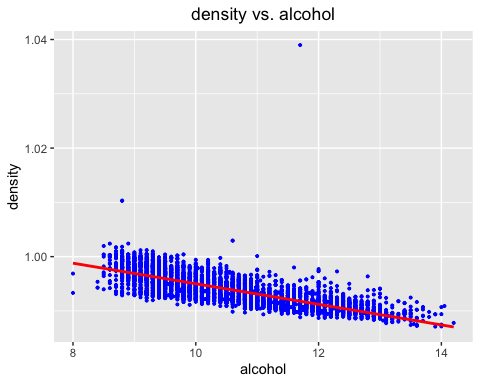
ggarrange(bp9, bp10, bp11, nrow = 2, ncol =2)



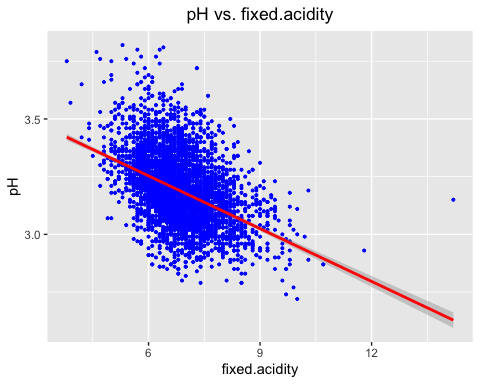
#Analysing relationship among numerical variables  
gg1 <- ggplot(white\_df, aes(x=residual.sugar, y=density)) +  
 geom\_point(color="blue",size=0.7) +   
 labs(title="density vs. residual sugar") +  
 geom\_smooth(formula=y~x,method=lm, color="red") +  
 theme(plot.title=element\_text(hjust=0.5))  
gg1



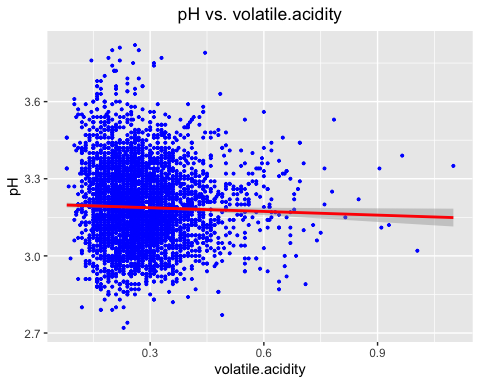
gg2 <- ggplot(white\_df, aes(x=alcohol, y=density)) +  
 geom\_point(color="blue",size=0.7) +   
 labs(title="density vs. alcohol") +  
 geom\_smooth(formula=y~x,method=lm, color="red") +  
 theme(plot.title=element\_text(hjust=0.5))  
gg2



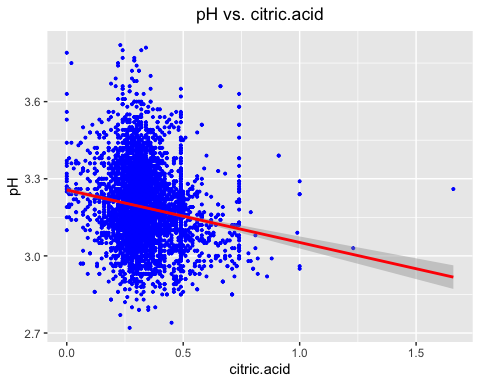
gg3 <- ggplot(white\_df, aes(x=fixed.acidity, y=pH)) +  
 geom\_point(color="blue",size=0.7) +   
 labs(title="pH vs. fixed.acidity") +  
 geom\_smooth(formula=y~x,method=lm, color="red") +  
 theme(plot.title=element\_text(hjust=0.5))  
gg3



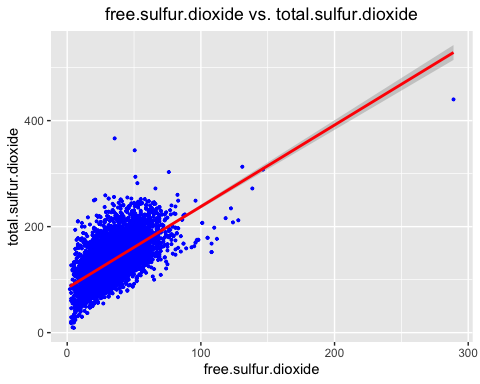
gg4 <- ggplot(white\_df, aes(x=volatile.acidity, y=pH)) +  
 geom\_point(color="blue",size=0.7) +   
 labs(title="pH vs. volatile.acidity") +  
 geom\_smooth(formula=y~x,method=lm, color="red") +  
 theme(plot.title=element\_text(hjust=0.5))  
gg4



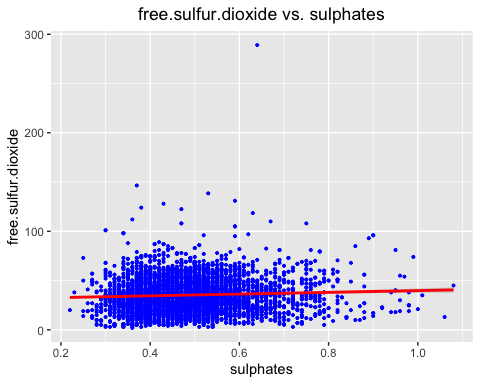
gg5 <- ggplot(white\_df, aes(x=citric.acid, y=pH)) +  
 geom\_point(color="blue",size=0.7) +   
 labs(title="pH vs. citric.acid") +  
 geom\_smooth(formula=y~x,method=lm, color="red") +  
 theme(plot.title=element\_text(hjust=0.5))  
gg5



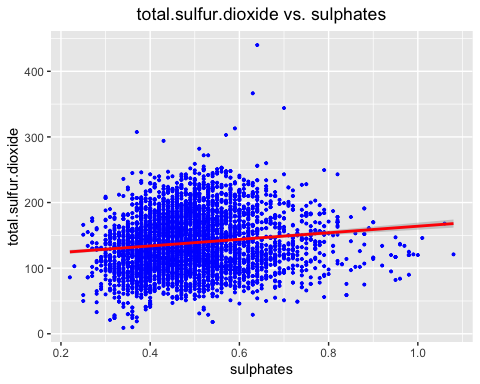
gg6 <- ggplot(white\_df, aes(x=free.sulfur.dioxide, y=total.sulfur.dioxide)) +  
 geom\_point(color="blue",size=0.7) +   
 labs(title="free.sulfur.dioxide vs. total.sulfur.dioxide") +  
 geom\_smooth(formula=y~x,method=lm, color="red") +  
 theme(plot.title=element\_text(hjust=0.5))  
gg6



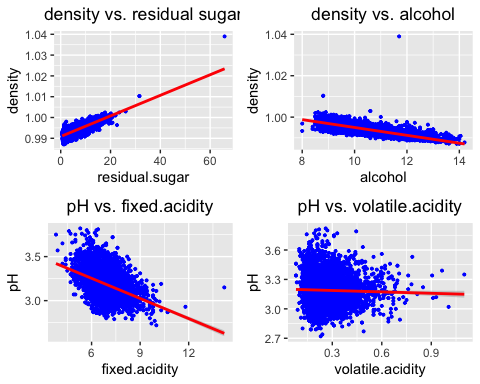
gg7 <-ggplot(white\_df, aes(x=sulphates, y=free.sulfur.dioxide)) +  
 geom\_point(color="blue",size=0.7) +   
 labs(title="free.sulfur.dioxide vs. sulphates") +  
 geom\_smooth(formula=y~x,method=lm, color="red") +  
 theme(plot.title=element\_text(hjust=0.5))  
gg7



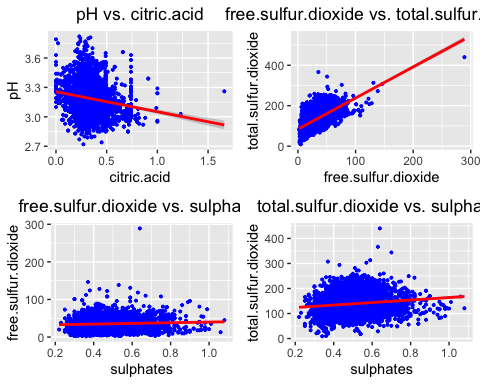
gg8 <-ggplot(white\_df, aes(x=sulphates, y=total.sulfur.dioxide)) +  
 geom\_point(color="blue",size=0.7) +   
 labs(title="total.sulfur.dioxide vs. sulphates") +  
 geom\_smooth(formula=y~x,method=lm, color="red") +  
 theme(plot.title=element\_text(hjust=0.5))  
gg8



ggarrange(gg1, gg2, gg3, gg4, nrow = 2, ncol =2)



ggarrange(gg5, gg6, gg7, gg8, nrow = 2, ncol =2)



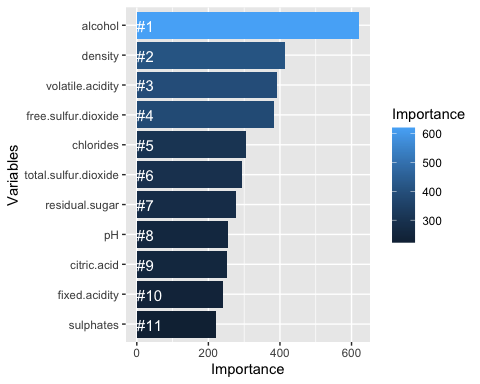
#Baseline Random Forest Model  
whitewineRF<-randomForest(quality ~ .,white\_df,ntree=150)  
  
whitewineRF

##   
## Call:  
## randomForest(formula = quality ~ ., data = white\_df, ntree = 150)   
## Type of random forest: regression  
## Number of trees: 150  
## No. of variables tried at each split: 3  
##   
## Mean of squared residuals: 0.3470637  
## % Var explained: 55.74

# Get importance  
Importance <- importance(whitewineRF)  
varImportance <- data.frame(Variables = row.names(Importance),   
 Importance = (Importance))  
  
# Create a rank variable based on importance  
rankImportance <- varImportance %>%  
 mutate(Rank = paste0('#',dense\_rank(desc(Importance))))  
importance(whitewineRF)

## IncNodePurity  
## fixed.acidity 239.9680  
## volatile.acidity 392.2521  
## citric.acid 252.4018  
## residual.sugar 277.1518  
## chlorides 305.7604  
## free.sulfur.dioxide 383.3018  
## total.sulfur.dioxide 293.4699  
## density 414.0035  
## pH 255.4234  
## sulphates 222.4904  
## alcohol 620.8299

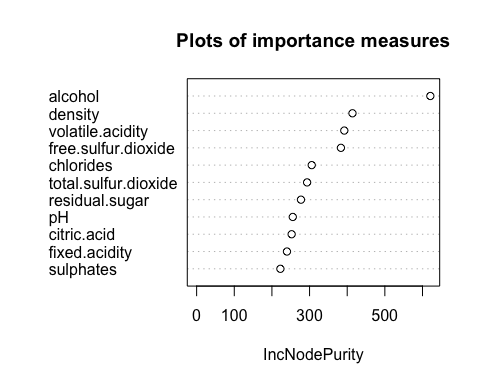
# Use ggplot2 to visualize the relative importance of variables  
ggplot(rankImportance, aes(x = reorder(Variables, Importance),   
 y = Importance, fill = Importance)) +  
 geom\_bar(stat='identity') +   
 geom\_text(aes(x = Variables, y = 0.5, label = Rank),  
 hjust=0, vjust=0.55, size = 4, colour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



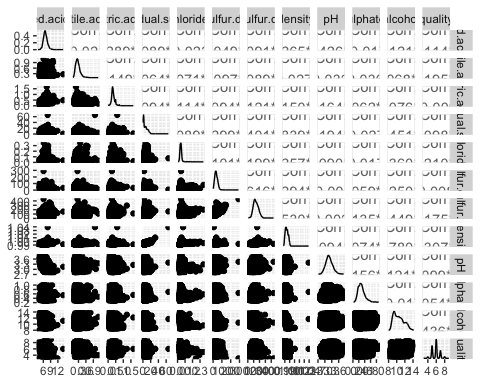
#importance plot  
importance(whitewineRF)

## IncNodePurity  
## fixed.acidity 239.9680  
## volatile.acidity 392.2521  
## citric.acid 252.4018  
## residual.sugar 277.1518  
## chlorides 305.7604  
## free.sulfur.dioxide 383.3018  
## total.sulfur.dioxide 293.4699  
## density 414.0035  
## pH 255.4234  
## sulphates 222.4904  
## alcohol 620.8299

varImpPlot(whitewineRF, main="Plots of importance measures")



ggpairs(white\_df)



#Linear regression model with top 4 highest correlation with Total Incidents  
white\_lm <- lm(quality ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide, data = white\_df)  
summary(white\_lm)

##   
## Call:  
## lm(formula = quality ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide, data = white\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.3107 -0.4984 -0.0413 0.4897 3.1713   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.5096598 0.1346811 18.634 < 2e-16 \*\*\*  
## alcohol 0.3424202 0.0100918 33.930 < 2e-16 \*\*\*  
## volatile.acidity -2.0210739 0.1105420 -18.283 < 2e-16 \*\*\*  
## sulphates 0.3616881 0.0975247 3.709 0.000211 \*\*\*  
## total.sulfur.dioxide 0.0011067 0.0002955 3.745 0.000183 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7697 on 4893 degrees of freedom  
## Multiple R-squared: 0.2453, Adjusted R-squared: 0.2446   
## F-statistic: 397.5 on 4 and 4893 DF, p-value: < 2.2e-16

# Define training control  
set.seed(555)  
train <- white\_df[1:800, ]  
test <- white\_df[801:nrow(white\_df), ]  
# Train the model  
white\_model <- lm(quality ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide, data=train)  
summary(white\_model)

##   
## Call:  
## lm(formula = quality ~ alcohol + volatile.acidity + sulphates +   
## total.sulfur.dioxide, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.3798 -0.5022 0.0072 0.5062 3.0868   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.5810459 0.3697099 6.981 6.18e-12 \*\*\*  
## alcohol 0.3310033 0.0288349 11.479 < 2e-16 \*\*\*  
## volatile.acidity -2.1264214 0.2772213 -7.670 5.01e-14 \*\*\*  
## sulphates 1.1123045 0.2597845 4.282 2.08e-05 \*\*\*  
## total.sulfur.dioxide -0.0003852 0.0007323 -0.526 0.599   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7743 on 795 degrees of freedom  
## Multiple R-squared: 0.2507, Adjusted R-squared: 0.2469   
## F-statistic: 66.48 on 4 and 795 DF, p-value: < 2.2e-16

#calculate MSE  
mean(white\_model$residuals^2)

## [1] 0.5958007

#calculate RMSE  
sqrt(mean(white\_model$residuals^2))

## [1] 0.7718813

#calculate MAE  
predValues <- predict(white\_model,test)  
#MAE for the model  
mean(abs(test$quality -predValues))

## [1] 0.6127622

#Lasso regression  
x <- model.matrix(quality~., white\_df)[,-1]  
y <- white\_df$quality  
mod <- cv.glmnet(as.matrix(x), y, alpha=1)  
  
#coefficients with the minimum cross-validation error  
as.matrix(coef(mod, mod$lambda.min))

## s1  
## (Intercept) 1.218573e+02  
## fixed.acidity 3.847004e-02  
## volatile.acidity -1.874168e+00  
## citric.acid 1.373124e-03  
## residual.sugar 6.983052e-02  
## chlorides -3.572286e-01  
## free.sulfur.dioxide 3.630384e-03  
## total.sulfur.dioxide -2.448370e-04  
## density -1.213902e+02  
## pH 5.588171e-01  
## sulphates 5.733381e-01  
## alcohol 2.241061e-01

#coefficients with the "largest value of lambda such that error is   
#within 1 standard error of the minimum  
as.matrix(coef(mod, mod$lambda.1se))

## s1  
## (Intercept) 2.672428715  
## fixed.acidity -0.040800606  
## volatile.acidity -1.773726362  
## citric.acid 0.000000000  
## residual.sugar 0.016479681  
## chlorides -0.591302671  
## free.sulfur.dioxide 0.002676398  
## total.sulfur.dioxide 0.000000000  
## density 0.000000000  
## pH 0.044969055  
## sulphates 0.218520816  
## alcohol 0.338160233

#Lasso regression model using above independent variables  
white\_lm1 <- lm(quality ~ fixed.acidity + volatile.acidity + chlorides + total.sulfur.dioxide   
 + sulphates + alcohol, data=white\_df)  
summary(white\_lm1)

##   
## Call:  
## lm(formula = quality ~ fixed.acidity + volatile.acidity + chlorides +   
## total.sulfur.dioxide + sulphates + alcohol, data = white\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.3885 -0.4967 -0.0518 0.4807 3.3119   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.2132033 0.1764150 18.214 < 2e-16 \*\*\*  
## fixed.acidity -0.0707338 0.0131049 -5.398 7.08e-08 \*\*\*  
## volatile.acidity -2.0011868 0.1106875 -18.080 < 2e-16 \*\*\*  
## chlorides -1.6508700 0.5409552 -3.052 0.002287 \*\*   
## total.sulfur.dioxide 0.0012086 0.0002949 4.098 4.24e-05 \*\*\*  
## sulphates 0.3507270 0.0972098 3.608 0.000312 \*\*\*  
## alcohol 0.3274520 0.0106497 30.748 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7669 on 4891 degrees of freedom  
## Multiple R-squared: 0.251, Adjusted R-squared: 0.2501   
## F-statistic: 273.2 on 6 and 4891 DF, p-value: < 2.2e-16

# Define training control  
set.seed(555)  
train <- white\_df[1:800, ]  
test <- white\_df[801:nrow(white\_df), ]  
# Train the model  
white\_model1 <- lm(quality ~ fixed.acidity + volatile.acidity + chlorides + total.sulfur.dioxide   
 + sulphates + alcohol, data=train)  
summary(white\_model1)

##   
## Call:  
## lm(formula = quality ~ fixed.acidity + volatile.acidity + chlorides +   
## total.sulfur.dioxide + sulphates + alcohol, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.4447 -0.4958 0.0039 0.4992 3.2031   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.2088704 0.5013010 6.401 2.64e-10 \*\*\*  
## fixed.acidity -0.0578413 0.0378383 -1.529 0.127   
## volatile.acidity -2.1456664 0.2791806 -7.686 4.50e-14 \*\*\*  
## chlorides -1.4501201 1.1781069 -1.231 0.219   
## total.sulfur.dioxide -0.0003389 0.0007329 -0.462 0.644   
## sulphates 1.0800241 0.2602024 4.151 3.67e-05 \*\*\*  
## alcohol 0.3163131 0.0301153 10.503 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7735 on 793 degrees of freedom  
## Multiple R-squared: 0.2541, Adjusted R-squared: 0.2485   
## F-statistic: 45.03 on 6 and 793 DF, p-value: < 2.2e-16

#calculate MSE  
mean(white\_model1$residuals^2)

## [1] 0.5930486

#calculate RMSE  
sqrt(mean(white\_model1$residuals^2))

## [1] 0.7700965

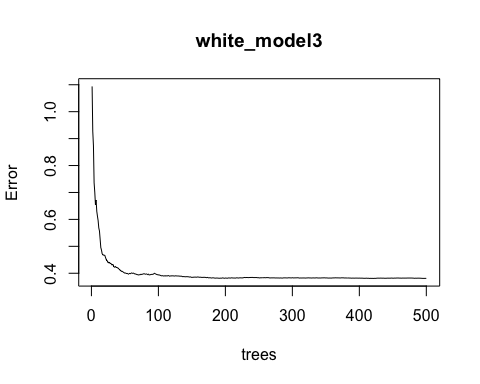
#calculate MAE  
predValues <- predict(white\_model1,test)  
#MAE for the model  
mean(abs(test$quality -predValues))

## [1] 0.6097995

#Random forest  
set.seed(555)  
train <- white\_df[1:800, ]  
test <- white\_df[801:nrow(white\_df), ]  
white\_model3 <- randomForest(quality ~ ., train, mtry = 3,   
 importance = TRUE, na.action = na.omit)  
print(white\_model3)

##   
## Call:  
## randomForest(formula = quality ~ ., data = train, mtry = 3, importance = TRUE, na.action = na.omit)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## Mean of squared residuals: 0.3809709  
## % Var explained: 52.09

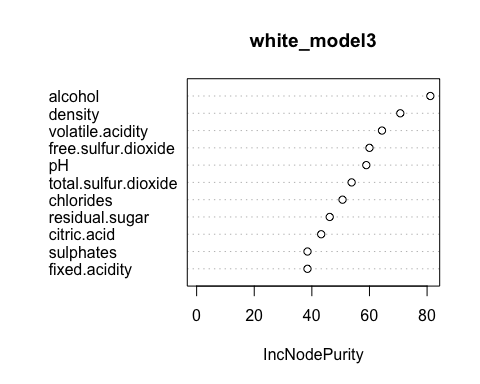
#Plot the error vs the number of trees graph   
plot(white\_model3)



importance(white\_model3)

## %IncMSE IncNodePurity  
## fixed.acidity 19.00341 38.47732  
## volatile.acidity 37.85046 64.34346  
## citric.acid 29.22456 43.24203  
## residual.sugar 23.68386 46.19656  
## chlorides 25.12812 50.64720  
## free.sulfur.dioxide 29.94043 60.00146  
## total.sulfur.dioxide 23.71018 53.80431  
## density 27.98187 70.70141  
## pH 28.44408 58.90013  
## sulphates 26.12448 38.48484  
## alcohol 25.15161 81.12908

varImpPlot(white\_model3,type=2)



# MSE for the model  
white\_model3$mse[length(white\_model3$mse)]

## [1] 0.3809709

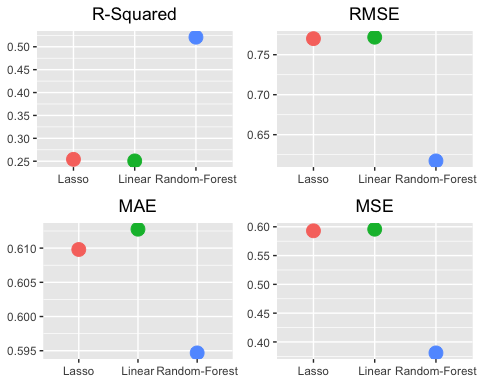
# RMSE for the model  
sqrt(white\_model3$mse[length(white\_model3$mse)])

## [1] 0.6172284

predValues <- predict(white\_model3,test)  
#MAE for the model  
mean(abs(test$quality -predValues))

## [1] 0.5946836

#Comparing models  
  
Model <- c("Linear", "Lasso", "Random-Forest")  
R\_squared <- c(0.2507, 0.2541, 0.5209)  
RMSE <- c(0.7718813, 0.7700965, 0.6172284)  
MAE <- c(0.6127622, 0.6097995, 0.5946836)  
MSE <- c(0.5958007, 0.5930486, 0.3809709)  
ml <- data.frame(Model, R\_squared, RMSE, MAE, MSE)  
  
  
p1 <- ggplot(ml, aes(Model, RMSE)) + geom\_point(aes(colour = factor(Model), size = 4)) + labs(title="RMSE") + theme(plot.title=element\_text(hjust=0.5), axis.title.y = element\_blank(),axis.title.x = element\_blank(), legend.position="none")  
p2 <- ggplot(ml, aes(Model, R\_squared)) + geom\_point(aes(colour = factor(Model), size = 4)) + labs(title="R-Squared") + theme(plot.title=element\_text(hjust=0.5), axis.title.y = element\_blank(),axis.title.x = element\_blank(), legend.position="none")  
p3 <- ggplot(ml, aes(Model, MAE)) + geom\_point(aes(colour = factor(Model), size = 4)) + labs(title="MAE") + theme(plot.title=element\_text(hjust=0.5), axis.title.y = element\_blank(),axis.title.x = element\_blank(), legend.position="none")  
p4 <- ggplot(ml, aes(Model, MSE)) + geom\_point(aes(colour = factor(Model), size = 4)) + labs(title="MSE") + theme(plot.title=element\_text(hjust=0.5), axis.title.y = element\_blank(),axis.title.x = element\_blank(), legend.position="none")  
ggarrange(p2,p1,p3,p4, nrow=2, ncol=2)



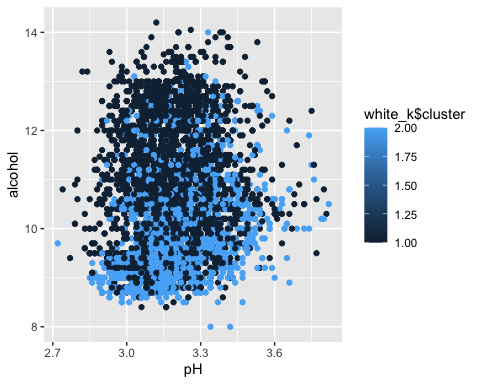
#clustering to find relation between predictors  
set.seed(1941)  
white\_k <- kmeans(white\_df, 2, nstart = 25)  
names(white\_k)

## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

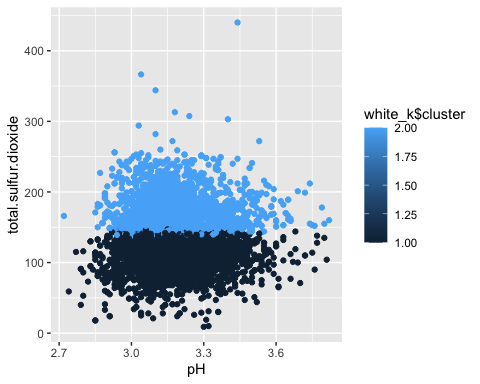
white\_k$centers

## fixed.acidity volatile.acidity citric.acid residual.sugar chlorides  
## 1 6.785150 0.2728681 0.3218799 4.793352 0.04217691  
## 2 6.947571 0.2854000 0.3505952 8.520643 0.05056286  
## free.sulfur.dioxide total.sulfur.dioxide density pH sulphates  
## 1 27.16976 108.5960 0.9927779 3.188867 0.4793531  
## 2 46.15143 178.0186 0.9956921 3.187467 0.5038286  
## alcohol quality  
## 1 10.962258 6.009650  
## 2 9.917373 5.702381

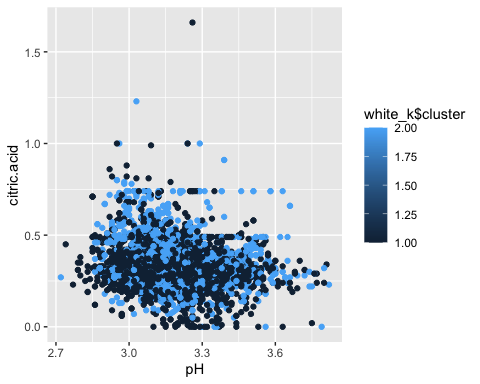
c1 <- ggplot(white\_df)+geom\_point(aes(x=pH,y=alcohol ,color=white\_k$cluster))  
c1



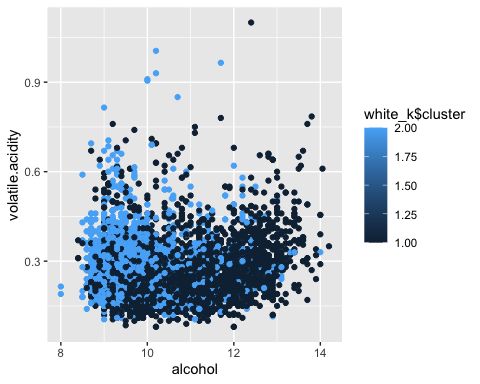
c2 <- ggplot(white\_df)+geom\_point(aes(x=pH,y=total.sulfur.dioxide ,color=white\_k$cluster))  
c2



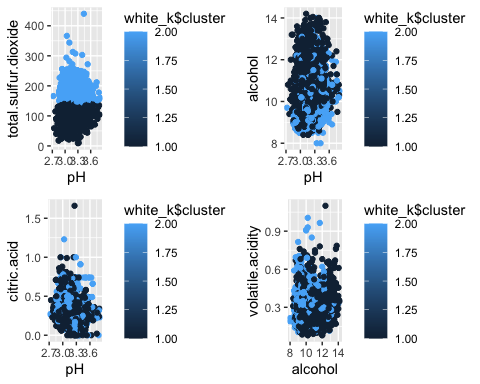
c3 <- ggplot(white\_df)+geom\_point(aes(x=pH,y=citric.acid ,color=white\_k$cluster))  
c3



c4 <- ggplot(white\_df)+geom\_point(aes(x=alcohol,y=volatile.acidity ,color=white\_k$cluster))  
c4



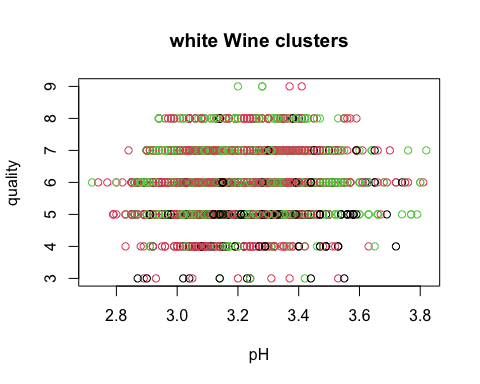
ggarrange(c2,c1,c3,c4, nrow=2, ncol=2)



#clustering  
set.seed(1406)  
kmeans.re <- kmeans(white\_df, centers = 3, nstart = 25)  
kmeans.re

## K-means clustering with 3 clusters of sizes 1125, 1797, 1976  
##   
## Cluster means:  
## fixed.acidity volatile.acidity citric.acid residual.sugar chlorides  
## 1 6.970133 0.2934800 0.3533956 9.356089 0.05179556  
## 2 6.778075 0.2730662 0.3174791 4.170924 0.04090095  
## 3 6.858882 0.2742713 0.3384565 6.722874 0.04677328  
## free.sulfur.dioxide total.sulfur.dioxide density pH sulphates  
## 1 50.69689 196.93911 0.9963027 3.183013 0.5149511  
## 2 23.93684 96.05175 0.9923317 3.187863 0.4798164  
## 3 36.88790 143.48634 0.9942740 3.191624 0.4846761  
## alcohol quality  
## 1 9.707156 5.555556  
## 2 11.109507 6.002226  
## 3 10.432463 5.948381  
##   
## Clustering vector:  
## [1] 1 3 2 1 1 2 3 1 3 3 2 2 2 3 1 2 2 2 3 3 2 2 3 3 3 1 3 3 3 2 3 2 2 3 1 3 3  
## [38] 3 3 3 3 3 3 3 3 3 1 1 3 3 3 3 2 2 3 1 1 3 2 2 3 3 3 2 3 2 3 1 2 2 1 1 1 2  
## [75] 2 2 2 2 2 2 2 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 2 2 3 1 3 3 1 1 2 3 1 1 1 3  
## [112] 1 1 3 3 2 3 2 1 1 2 3 3 3 3 3 3 1 1 1 2 1 1 1 1 1 3 3 2 2 2 3 2 2 2 2 3 2  
## [149] 2 2 3 3 2 2 2 1 1 3 3 3 3 3 2 1 1 1 1 2 3 2 3 2 2 3 1 3 2 1 2 1 1 1 3 1 1  
## [186] 1 3 3 2 1 1 3 3 3 1 1 1 1 1 1 1 1 1 3 2 3 3 3 2 3 2 2 3 3 2 3 3 3 1 3 3 3  
## [223] 2 3 3 3 1 1 1 3 3 1 1 1 1 1 1 1 3 3 1 2 2 1 3 1 3 2 2 2 1 1 3 2 3 3 2 2 2  
## [260] 2 2 3 2 1 3 3 1 3 3 3 1 3 3 3 1 1 3 2 2 2 2 2 1 1 1 1 1 1 1 1 1 3 1 3 1 1  
## [297] 3 1 3 3 2 2 2 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 2 2 3 3 3 1 1 1 3 1 2 2 3 2  
## [334] 3 2 2 2 1 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 1 1 1 3 3 3 3 2 3 1 2 2 3 3 1 2  
## [371] 3 1 1 3 2 2 2 2 3 2 2 1 3 3 3 2 2 1 3 1 1 2 3 2 3 1 2 2 1 2 2 3 2 1 3 1 2  
## [408] 3 2 2 3 3 2 2 3 3 2 1 2 3 3 1 1 1 3 1 1 1 2 1 1 2 1 3 2 2 1 1 1 2 2 3 3 1  
## [445] 3 2 3 2 2 3 3 3 3 2 3 2 2 2 1 1 2 1 3 2 1 3 3 2 1 1 3 1 2 3 2 1 2 2 3 3 3  
## [482] 2 3 3 1 2 3 2 1 1 2 2 1 3 2 3 1 3 3 1 1 3 1 1 3 3 3 3 3 3 3 3 3 2 2 3 3 3  
## [519] 2 2 3 3 2 2 2 3 2 2 2 2 2 3 1 1 1 1 1 1 2 1 3 1 3 3 3 3 1 2 3 1 3 2 2 3 2  
## [556] 2 3 3 3 3 3 3 1 3 3 2 2 3 3 1 1 2 1 3 3 1 1 3 2 3 1 2 3 2 3 2 3 3 3 3 3 3  
## [593] 3 3 3 3 3 3 2 2 3 2 3 3 3 3 3 3 3 2 2 2 3 3 3 3 2 1 1 3 1 1 3 2 3 3 1 1 1  
## [630] 2 3 3 3 1 3 3 3 3 1 1 3 1 1 3 3 3 3 3 1 1 1 1 1 3 3 2 2 3 1 1 2 3 1 2 1 3  
## [667] 1 1 1 1 1 2 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 1 1 3 3 1 1 1 1 1 2 1 1 1 1 3 2  
## [704] 2 2 2 1 3 3 2 1 3 3 1 1 3 1 3 3 3 1 1 3 2 3 3 2 2 3 1 3 1 2 1 1 3 1 1 3 3  
## [741] 2 2 3 3 3 2 1 1 1 1 1 1 1 1 3 2 1 1 3 3 1 1 1 3 3 3 1 2 2 2 2 3 3 1 3 2 2  
## [778] 1 1 3 2 1 1 3 1 2 2 2 2 2 2 2 3 2 1 3 1 3 2 2 3 1 1 3 2 3 1 1 1 1 1 2 3 3  
## [815] 1 3 2 3 3 3 2 1 2 2 2 3 3 3 2 2 2 3 2 2 2 2 1 1 1 3 3 3 3 2 2 2 3 2 3 2 2  
## [852] 3 2 3 2 2 3 1 3 3 2 1 3 2 3 2 2 3 1 2 3 3 3 2 2 3 3 2 2 2 3 2 3 3 1 2 1 2  
## [889] 1 2 3 2 2 3 2 2 1 2 2 1 3 2 1 1 3 2 2 3 3 1 3 3 3 2 2 2 3 3 2 2 3 3 3 1 3  
## [926] 2 2 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1 2 3 3 2 1 3 2 2 3 3 2 2 3 3 3 2 2 1 1 2  
## [963] 1 2 3 2 1 3 2 2 2 2 3 2 2 2 3 1 3 2 2 3 2 2 3 3 3 3 3 2 3 2 1 2 1 3 2 3 1  
## [1000] 3 2 1 1 3 3 2 3 2 1 3 2 1 2 1 3 2 3 2 3 1 3 3 1 1 1 3 3 2 3 1 3 1 1 1 1 3  
## [1037] 2 2 2 2 3 2 2 1 2 2 2 2 2 2 3 3 2 2 2 2 2 1 2 3 3 1 1 1 3 3 1 3 2 3 3 3 2  
## [1074] 1 1 3 1 3 1 1 2 3 3 1 3 3 3 3 1 3 1 1 3 3 1 2 3 3 3 2 3 2 3 2 3 1 3 2 2 3  
## [1111] 2 2 1 2 2 2 2 1 2 3 2 2 2 2 3 1 1 2 2 2 3 3 3 2 3 3 1 1 3 2 2 3 3 3 3 1 3  
## [1148] 1 2 1 1 1 2 2 3 3 3 3 1 3 3 3 3 1 2 3 2 2 2 3 2 2 2 2 1 1 1 1 3 2 2 3 2 3  
## [1185] 3 1 1 2 3 2 2 3 2 3 1 3 3 3 2 2 2 2 1 2 2 1 1 1 3 3 2 1 3 2 2 3 2 1 3 2 1  
## [1222] 3 2 3 2 2 2 2 2 2 1 2 2 3 3 1 2 2 3 3 1 1 3 3 3 1 3 2 2 1 3 3 3 3 3 2 3 1  
## [1259] 1 1 1 3 3 1 2 1 2 3 3 1 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 1 2 2 2  
## [1296] 1 3 3 3 3 3 1 1 3 1 2 2 3 3 2 3 2 3 1 3 3 1 1 2 1 2 2 3 3 1 2 3 3 2 2 2 2  
## [1333] 1 1 2 1 3 1 1 2 3 3 2 1 3 2 3 3 3 2 2 3 1 1 1 3 1 3 2 3 3 1 2 2 3 3 2 2 1  
## [1370] 1 2 3 1 1 2 3 3 3 3 2 3 2 2 2 3 3 2 2 3 1 3 2 2 2 2 2 3 1 1 2 1 1 3 3 2 2  
## [1407] 2 2 1 2 2 2 3 3 3 2 3 1 3 2 3 2 2 1 3 3 3 3 2 2 2 2 2 2 3 2 1 2 1 1 3 2 2  
## [1444] 3 3 3 2 3 3 1 3 3 3 3 1 3 2 3 3 3 3 3 3 3 3 3 1 2 2 3 3 3 2 3 2 3 1 2 3 3  
## [1481] 3 3 1 2 3 3 2 3 1 1 2 2 1 1 1 1 2 3 2 3 3 3 2 3 3 3 1 3 1 3 3 3 3 3 3 3 3  
## [1518] 3 2 3 3 2 3 1 3 3 3 1 3 3 3 3 1 2 1 2 2 3 2 2 1 2 2 2 2 3 3 3 2 2 3 3 3 3  
## [1555] 3 3 1 3 2 2 3 2 3 3 2 2 3 2 1 3 3 1 3 2 2 1 2 3 1 1 2 2 2 1 1 3 3 1 3 3 3  
## [1592] 3 2 2 3 1 3 1 2 2 3 1 2 2 3 2 2 3 1 1 2 2 2 3 3 3 1 3 1 3 2 3 3 2 3 3 1 3  
## [1629] 2 2 2 2 2 2 1 3 3 3 1 2 3 3 3 3 3 3 3 2 2 3 3 1 2 1 1 3 2 3 1 1 1 1 2 1 1  
## [1666] 2 2 3 2 2 3 2 3 3 1 1 2 3 3 2 1 1 1 1 1 1 3 1 1 3 3 1 1 1 3 3 1 1 1 1 2 1  
## [1703] 2 3 3 3 3 1 3 1 2 3 1 3 2 3 2 3 1 2 1 1 3 3 3 3 2 1 2 3 2 1 2 3 1 1 2 1 2  
## [1740] 2 3 3 3 3 3 1 2 3 2 3 1 3 1 3 2 3 1 1 2 1 1 3 2 2 1 1 1 1 3 2 1 3 1 1 2 3  
## [1777] 3 3 3 3 3 1 3 2 3 3 2 3 3 1 1 2 3 1 2 3 3 3 1 3 1 2 1 3 1 2 2 1 2 1 1 2 2  
## [1814] 2 2 3 3 2 2 3 3 2 2 1 3 2 3 1 1 3 3 3 3 3 3 1 2 2 1 2 3 2 1 3 2 3 1 3 1 1  
## [1851] 1 2 3 2 1 1 1 3 3 1 3 1 2 1 3 3 1 3 3 3 3 3 3 3 3 3 3 2 1 2 1 2 1 1 3 2 2  
## [1888] 1 1 2 1 1 3 3 3 1 3 3 2 2 3 2 1 2 1 3 2 3 2 3 1 2 2 3 3 2 1 2 1 1 3 3 3 2  
## [1925] 2 2 2 3 1 1 1 1 2 1 2 1 1 3 2 3 1 3 1 1 1 3 3 1 3 3 1 1 3 1 1 1 2 2 1 2 2  
## [1962] 3 2 1 1 3 2 1 3 3 2 3 3 3 3 1 1 3 2 1 1 1 1 1 1 1 3 1 1 2 2 1 2 1 3 1 3 1  
## [1999] 1 1 3 3 3 1 1 3 1 2 3 2 2 3 3 3 2 2 2 2 3 3 3 3 1 3 1 3 2 1 3 1 3 3 2 1 2  
## [2036] 2 3 3 2 2 1 2 2 2 2 2 3 3 3 3 2 3 3 2 2 3 3 3 1 3 1 2 3 3 1 3 3 3 2 3 3 3  
## [2073] 3 1 1 3 2 1 2 2 2 2 2 2 3 3 2 3 3 3 1 1 2 3 3 3 3 3 1 3 3 2 1 3 3 3 1 3 1  
## [2110] 1 3 3 3 2 1 3 2 2 3 2 3 2 1 3 2 3 3 1 3 2 3 3 3 3 3 1 3 1 2 2 2 2 1 2 3 3  
## [2147] 2 3 2 2 2 2 2 2 1 1 2 2 2 2 3 2 2 2 3 3 3 1 1 1 1 1 2 3 1 1 2 3 3 2 3 2 3  
## [2184] 3 2 2 2 3 2 3 2 3 3 3 2 3 2 2 1 1 3 3 1 3 3 1 3 3 3 3 3 3 3 2 3 3 2 3 3 3  
## [2221] 3 3 3 3 3 3 1 1 3 3 3 3 2 3 2 3 1 3 3 3 1 1 1 1 1 3 3 3 2 1 1 2 1 1 1 3 3  
## [2258] 3 1 3 1 3 2 3 3 3 3 3 3 1 2 3 2 2 2 1 1 2 1 3 2 2 1 1 1 1 3 3 1 2 2 3 1 2  
## [2295] 2 3 3 1 2 2 3 3 1 3 3 3 3 3 3 3 3 3 3 2 3 3 2 3 3 2 3 3 2 3 1 2 2 2 2 1 1  
## [2332] 3 1 2 1 3 1 1 3 2 3 3 2 3 2 1 1 2 3 1 1 1 3 2 2 3 3 2 1 3 3 2 3 1 1 3 1 1  
## [2369] 1 3 2 1 2 2 1 3 1 2 1 1 1 3 2 2 2 3 1 1 2 2 3 3 3 3 1 1 1 2 2 2 2 2 1 3 3  
## [2406] 1 2 2 1 2 1 1 1 2 1 2 1 1 2 1 2 1 1 2 1 2 3 1 3 1 1 1 1 1 1 3 1 3 1 3 3 3  
## [2443] 1 1 1 1 1 3 2 1 3 3 3 3 1 1 3 3 1 3 3 2 2 1 3 3 3 3 2 2 1 3 2 3 2 2 3 2 1  
## [2480] 3 2 1 1 1 1 1 3 3 3 1 2 1 1 3 3 3 2 3 3 1 3 1 2 2 1 1 1 3 1 3 1 1 2 3 3 2  
## [2517] 3 1 2 1 1 3 3 1 1 3 3 2 2 3 3 3 3 3 2 3 2 3 3 2 1 3 3 2 2 3 1 3 1 1 3 1 2  
## [2554] 3 3 3 1 3 3 3 2 3 2 3 3 1 2 3 1 3 3 2 2 1 3 1 1 1 2 2 1 3 3 3 3 3 3 2 3 3  
## [2591] 3 3 3 3 2 3 1 3 1 1 3 1 3 2 2 2 3 1 1 2 1 1 2 3 2 3 3 3 3 3 3 3 3 3 2 1 3  
## [2628] 3 1 1 2 2 1 1 1 2 1 1 2 2 2 3 2 3 1 2 2 3 3 1 3 3 2 3 1 1 1 3 3 2 3 1 2 2  
## [2665] 2 2 3 3 2 1 3 3 3 2 2 2 2 3 3 2 3 3 3 3 3 2 3 1 3 2 3 2 3 3 2 3 1 2 3 3 2  
## [2702] 3 3 3 1 1 1 3 1 1 1 3 1 1 1 1 1 3 2 3 2 3 2 3 3 3 2 3 1 2 1 3 3 2 3 1 3 2  
## [2739] 3 2 3 3 3 2 2 2 3 3 1 3 1 2 3 2 2 1 1 2 2 3 3 3 3 2 3 2 2 3 2 3 1 3 3 2 3  
## [2776] 3 3 2 2 2 2 3 1 1 1 3 2 3 1 1 1 1 1 2 3 2 2 3 2 3 1 1 2 2 2 3 3 3 1 3 2 3  
## [2813] 2 3 2 2 3 2 3 3 3 1 3 2 1 3 1 3 3 3 3 1 2 2 3 3 1 3 2 2 2 2 2 2 2 2 2 2 3  
## [2850] 1 3 2 3 2 2 3 3 2 3 2 3 2 2 2 2 2 3 2 3 2 2 2 1 2 3 2 3 3 3 2 2 2 3 2 2 2  
## [2887] 2 2 2 3 3 3 1 3 2 1 1 1 2 3 2 2 3 2 2 3 1 2 2 2 1 3 2 1 2 2 3 2 2 3 2 1 3  
## [2924] 3 3 1 2 3 3 3 3 2 1 3 2 2 2 1 2 2 3 2 1 2 3 3 2 2 1 3 2 2 2 2 2 3 2 2 2 2  
## [2961] 2 2 2 2 3 2 2 2 3 2 3 2 3 2 2 3 3 3 3 3 2 2 1 1 2 2 2 2 2 1 1 3 3 2 2 3 2  
## [2998] 3 2 2 2 3 3 2 2 3 1 1 1 1 1 2 2 2 2 2 2 3 3 2 2 2 3 2 2 3 2 3 3 3 3 3 2 1  
## [3035] 3 2 1 1 2 1 3 1 1 2 3 2 2 3 3 1 1 1 1 2 2 2 2 2 1 2 1 1 2 1 1 3 1 3 2 2 2  
## [3072] 2 1 3 3 2 2 1 2 2 2 1 2 2 2 2 2 1 2 2 2 1 3 3 2 2 2 3 3 2 3 2 2 3 3 3 3 3  
## [3109] 3 1 2 2 3 3 2 2 2 2 3 2 1 3 2 2 3 3 3 2 2 3 3 1 2 3 2 1 2 3 2 3 1 2 3 2 3  
## [3146] 3 3 3 3 3 2 3 1 2 2 3 3 3 2 3 2 3 3 2 1 3 3 2 2 2 3 2 2 3 3 3 2 3 2 2 2 2  
## [3183] 3 2 2 3 2 3 3 3 2 3 2 3 2 3 3 3 3 3 3 3 2 3 1 3 2 3 3 3 2 3 3 3 2 2 2 2 2  
## [3220] 2 2 2 2 3 3 2 3 1 1 3 3 2 2 2 3 3 3 3 2 2 2 2 2 2 2 2 2 3 3 3 3 1 2 2 1 1  
## [3257] 1 1 1 1 1 2 1 2 1 3 2 3 3 1 2 2 2 3 3 2 3 3 1 3 2 3 3 3 2 3 3 3 3 1 2 2 1  
## [3294] 2 2 1 1 1 2 2 2 2 2 2 3 2 3 1 3 2 3 3 2 2 1 2 2 2 2 3 3 2 2 2 2 3 1 2 2 2  
## [3331] 1 3 3 3 2 1 1 1 2 2 3 2 2 1 1 1 1 3 2 2 3 2 2 2 3 2 3 2 2 2 2 2 3 2 2 2 2  
## [3368] 2 3 2 2 2 3 3 3 3 3 1 3 1 3 3 3 3 1 3 3 1 2 2 2 2 2 2 1 1 3 1 1 2 3 3 3 2  
## [3405] 2 2 2 1 1 2 3 3 3 1 3 3 1 2 1 3 3 2 2 3 2 3 3 3 2 3 3 3 2 2 2 2 2 2 1 2 2  
## [3442] 2 2 2 1 3 1 2 3 3 2 2 3 2 2 2 1 3 2 3 2 1 2 2 1 2 2 1 2 3 1 2 3 2 1 1 2 3  
## [3479] 1 2 2 3 2 2 2 2 2 1 2 2 3 2 3 3 3 2 3 3 3 3 3 3 1 2 3 2 3 2 1 1 1 2 2 2 2  
## [3516] 2 1 2 3 3 1 3 1 1 3 3 3 3 2 3 1 1 2 3 1 1 2 2 3 2 2 3 3 3 3 1 3 1 3 3 3 2  
## [3553] 2 2 2 3 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 3 3 2 2 2 2 3 2 2  
## [3590] 2 2 1 3 3 2 3 2 3 1 1 2 3 3 3 2 3 3 2 3 3 3 2 2 3 3 2 1 2 3 3 1 3 2 3 2 1  
## [3627] 2 1 1 1 3 3 3 3 3 2 2 3 2 2 3 3 3 3 3 3 2 2 3 3 2 1 2 1 2 3 1 3 3 3 2 3 2  
## [3664] 3 3 3 3 2 2 2 3 2 2 2 2 3 2 3 2 2 1 3 3 1 3 1 1 2 1 3 3 2 2 3 2 3 3 1 2 1  
## [3701] 3 3 1 1 1 1 2 2 1 3 2 1 3 1 2 1 1 2 3 1 3 3 3 3 3 2 2 3 2 3 1 3 3 2 3 2 2  
## [3738] 3 3 3 3 3 3 3 3 3 3 1 3 3 3 2 1 1 3 3 2 1 1 3 2 2 2 2 3 1 1 1 3 1 3 3 3 2  
## [3775] 1 2 3 2 2 2 1 2 2 1 3 2 1 1 1 1 1 1 3 1 2 3 2 2 3 1 2 2 3 2 2 2 2 2 2 2 3  
## [3812] 3 3 1 3 2 3 3 3 3 3 3 1 1 2 2 3 2 3 2 2 3 3 2 3 2 2 2 3 2 2 1 2 2 2 1 2 1  
## [3849] 2 2 2 3 2 2 2 3 2 2 3 1 1 3 1 1 2 2 3 1 1 3 3 1 1 1 1 2 3 2 1 2 2 2 3 3 2  
## [3886] 2 3 2 2 3 2 3 3 2 3 2 1 1 2 3 2 2 2 2 2 2 2 2 3 3 3 2 2 2 3 2 3 1 2 2 3 1  
## [3923] 2 2 2 2 2 2 3 3 2 2 2 2 2 1 2 3 2 3 3 3 2 2 3 3 3 3 2 3 3 3 2 3 3 2 3 2 2  
## [3960] 2 3 1 3 2 1 2 2 1 1 3 3 3 2 1 1 3 3 2 1 1 3 3 3 2 3 2 2 1 3 2 1 2 2 2 2 2  
## [3997] 2 3 2 2 2 2 2 2 2 2 2 2 2 3 3 3 2 1 1 2 3 2 1 2 2 2 3 3 3 2 3 3 2 3 2 2 2  
## [4034] 3 2 1 1 1 2 2 1 3 3 3 3 3 3 3 3 2 3 2 3 3 3 2 3 2 3 2 2 2 2 3 2 2 3 3 3 3  
## [4071] 3 2 3 2 2 3 3 2 2 2 3 2 3 1 2 2 2 2 2 3 1 2 2 2 2 2 3 2 3 2 2 3 3 2 2 3 3  
## [4108] 3 2 3 1 2 2 2 2 2 3 3 3 3 3 3 3 2 2 3 3 1 1 2 3 1 1 2 2 3 3 1 2 3 3 3 3 3  
## [4145] 3 3 3 3 2 2 1 1 2 1 1 1 1 3 3 3 3 3 3 2 2 2 3 3 2 3 3 2 3 2 1 3 3 2 3 1 3  
## [4182] 2 3 2 2 1 2 2 2 2 2 3 2 2 2 2 2 3 3 2 2 2 2 1 3 2 3 2 2 2 3 1 3 2 1 1 1 2  
## [4219] 1 1 2 3 2 2 2 1 1 2 1 3 2 2 2 2 2 2 2 2 2 2 3 3 3 2 3 2 2 3 2 3 2 2 2 2 2  
## [4256] 2 2 3 3 2 3 2 3 3 2 3 2 2 3 1 1 1 2 2 2 2 2 1 2 3 2 3 3 2 2 2 2 2 2 2 3 1  
## [4293] 2 1 2 1 2 2 2 3 3 3 1 2 2 2 3 2 2 2 3 2 2 2 2 2 2 3 2 2 1 3 3 3 2 1 3 1 3  
## [4330] 1 1 3 3 3 3 3 3 3 3 2 3 3 3 2 2 2 2 3 3 1 2 3 3 3 3 3 1 1 1 1 3 2 3 3 3 3  
## [4367] 2 3 2 3 2 2 2 2 1 1 3 2 3 3 3 3 2 2 1 1 1 2 2 2 3 1 3 3 3 3 1 1 1 3 1 3 1  
## [4404] 1 1 1 3 2 1 3 1 2 2 2 2 3 1 3 3 3 3 3 3 3 3 3 3 1 3 3 3 2 2 1 3 3 2 3 3 3  
## [4441] 3 2 3 3 2 2 2 2 3 3 3 1 1 2 3 1 1 1 3 2 3 3 2 2 3 3 3 3 2 2 2 2 2 3 3 2 2  
## [4478] 1 1 3 3 1 3 2 3 2 2 2 3 3 2 2 2 3 2 1 2 3 2 2 1 1 3 2 3 3 2 2 3 2 3 2 2 2  
## [4515] 1 3 2 2 2 1 1 1 2 3 3 1 1 2 2 3 3 3 3 3 2 3 3 2 3 2 2 2 2 2 2 2 2 2 1 2 2  
## [4552] 2 2 2 2 2 3 1 3 2 2 2 3 2 2 3 1 3 3 3 2 2 2 2 2 3 2 2 2 3 2 3 3 3 2 3 3 2  
## [4589] 3 3 3 1 3 2 3 3 2 2 2 2 2 3 3 3 3 2 2 1 2 2 2 2 3 3 3 3 2 3 3 1 3 2 3 3 2  
## [4626] 3 1 2 2 2 3 2 1 1 3 2 1 3 1 1 3 3 2 1 3 3 2 2 1 2 2 2 1 2 3 3 3 3 3 2 2 3  
## [4663] 2 2 2 2 1 2 2 2 3 1 1 2 3 3 3 2 3 1 3 2 2 3 1 2 3 3 3 3 3 3 3 3 3 2 2 2 3  
## [4700] 1 1 3 3 2 1 2 2 2 2 3 2 2 2 2 2 2 2 3 3 2 2 2 3 1 2 2 2 3 3 2 2 3 3 3 2 2  
## [4737] 2 3 2 2 1 3 3 3 2 1 3 2 1 1 3 1 2 2 2 2 2 2 3 3 3 3 2 3 2 2 3 1 3 3 3 3 1  
## [4774] 2 3 3 3 2 2 2 3 3 3 3 3 2 3 2 3 2 3 3 3 3 3 3 3 2 2 2 3 3 3 2 2 2 3 2 2 3  
## [4811] 3 2 2 1 2 2 3 3 2 2 1 3 3 3 3 3 3 2 3 2 3 3 3 2 2 2 2 1 2 2 3 1 2 1 1 2 3  
## [4848] 2 3 3 3 3 2 2 2 3 3 2 1 2 2 2 2 2 3 2 2 2 3 2 3 2 1 3 3 2 2 2 2 1 1 2 3 3  
## [4885] 1 1 3 2 3 2 2 3 2 2 1 2 2 2  
##   
## Within cluster sum of squares by cluster:  
## [1] 1026873.8 872531.1 849211.8  
## (between\_SS / total\_SS = 73.6 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

plot(white\_df[c("pH", "quality")],   
 col = kmeans.re$cluster,   
 main = "white Wine clusters")



#Classification trees  
library(party)

## Loading required package: grid  
## Loading required package: mvtnorm  
## Loading required package: modeltools  
## Loading required package: stats4  
## Loading required package: strucchange  
## Loading required package: zoo  
##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric  
##   
## Loading required package: sandwich  
##   
## Attaching package: 'strucchange'  
##   
## The following object is masked from 'package:stringr':  
##   
## boundary

tree1=ctree(quality~alcohol+volatile.acidity,data=white\_df) #~target variable,predicting   
tree1

##   
## Conditional inference tree with 17 terminal nodes  
##   
## Response: quality   
## Inputs: alcohol, volatile.acidity   
## Number of observations: 4898   
##   
## 1) alcohol <= 10.8; criterion = 1, statistic = 929.085  
## 2) volatile.acidity <= 0.25; criterion = 1, statistic = 412.098  
## 3) volatile.acidity <= 0.205; criterion = 1, statistic = 52.154  
## 4) alcohol <= 9; criterion = 0.99, statistic = 7.887  
## 5)\* weights = 92   
## 4) alcohol > 9  
## 6) alcohol <= 10.1; criterion = 0.952, statistic = 5.057  
## 7) alcohol <= 9.9; criterion = 0.964, statistic = 5.582  
## 8)\* weights = 277   
## 7) alcohol > 9.9  
## 9)\* weights = 85   
## 6) alcohol > 10.1  
## 10)\* weights = 277   
## 3) volatile.acidity > 0.205  
## 11) alcohol <= 9.8; criterion = 1, statistic = 28.599  
## 12)\* weights = 438   
## 11) alcohol > 9.8  
## 13)\* weights = 306   
## 2) volatile.acidity > 0.25  
## 14) volatile.acidity <= 0.3; criterion = 1, statistic = 88.408  
## 15) alcohol <= 10.3; criterion = 1, statistic = 25.5  
## 16) alcohol <= 9; criterion = 0.97, statistic = 5.915  
## 17)\* weights = 115   
## 16) alcohol > 9  
## 18)\* weights = 435   
## 15) alcohol > 10.3  
## 19)\* weights = 152   
## 14) volatile.acidity > 0.3  
## 20) volatile.acidity <= 0.46; criterion = 1, statistic = 39.963  
## 21) alcohol <= 9.7; criterion = 1, statistic = 29.619  
## 22)\* weights = 482   
## 21) alcohol > 9.7  
## 23)\* weights = 283   
## 20) volatile.acidity > 0.46  
## 24)\* weights = 143   
## 1) alcohol > 10.8  
## 25) alcohol <= 11.73333; criterion = 1, statistic = 115.528  
## 26) volatile.acidity <= 0.46; criterion = 1, statistic = 35.936  
## 27) volatile.acidity <= 0.18; criterion = 0.995, statistic = 9.253  
## 28)\* weights = 168   
## 27) volatile.acidity > 0.18  
## 29)\* weights = 697   
## 26) volatile.acidity > 0.46  
## 30)\* weights = 16   
## 25) alcohol > 11.73333  
## 31) alcohol <= 12.75; criterion = 1, statistic = 18.802  
## 32)\* weights = 699   
## 31) alcohol > 12.75  
## 33)\* weights = 233

library(rpart)  
mytree=rpart(quality~alcohol+volatile.acidity,data=white\_df,method="class")  
mytree

## n= 4898   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 4898 2700 6 (0.0041 0.033 0.3 0.45 0.18 0.036 0.001)   
## 2) alcohol< 10.85 3085 1732 6 (0.0039 0.041 0.42 0.44 0.089 0.012 0.00032)   
## 4) volatile.acidity>=0.2375 1936 926 5 (0.0041 0.055 0.52 0.37 0.044 0.0036 0.00052)   
## 8) alcohol< 9.85 1230 482 5 (0.0041 0.05 0.61 0.31 0.025 0.00081 0) \*  
## 9) alcohol>=9.85 706 370 6 (0.0042 0.062 0.37 0.48 0.076 0.0085 0.0014) \*  
## 5) volatile.acidity< 0.2375 1149 515 6 (0.0035 0.017 0.24 0.55 0.17 0.026 0) \*  
## 3) alcohol>=10.85 1813 968 6 (0.0044 0.02 0.097 0.47 0.33 0.076 0.0022)   
## 6) alcohol< 12.55 1458 732 6 (0.0048 0.023 0.11 0.5 0.29 0.064 0.0014) \*  
## 7) alcohol>=12.55 355 180 7 (0.0028 0.011 0.025 0.34 0.49 0.13 0.0056) \*

library(rattle)

## Loading required package: bitops  
##   
## Attaching package: 'bitops'  
##   
## The following object is masked from 'package:Matrix':  
##   
## %&%  
##   
## Rattle: A free graphical interface for data science with R.  
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.  
##   
## Attaching package: 'rattle'  
##   
## The following object is masked from 'package:randomForest':  
##   
## importance

library(RColorBrewer)  
fancyRpartPlot(mytree,caption="white wine Classification")

