

Stat_fp_red.R

pravallikaavula

2022-12-08

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   0.3.5
## v tibble   3.1.8     v dplyr    1.0.10
## v tidyr    1.2.1     v stringr  1.4.1
## v readr    2.1.3     v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x 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)

##
## Use magrittr pipe '%>%' to chain several operations:
##   mtcars %>%
##     let(mpg_hp = mpg/hp) %>%
##     take(mean(mpg_hp), by = am)
##
## 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
red_df <- read.csv(file = "winequality-red.csv", as.is = TRUE, sep = ";", header = TRUE)
head(red_df)

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

```

```

## 6           13          40  0.9978 3.51      0.56      9.4
##   quality
## 1       5
## 2       5
## 3       5
## 4       6
## 5       5
## 6       5

dim(red_df)

## [1] 1599    12

#Data overview
str(red_df)

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

summary(red_df)

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

```

```
#Checking missing values
which(is.na(red_df))
```

```
## integer(0)
```

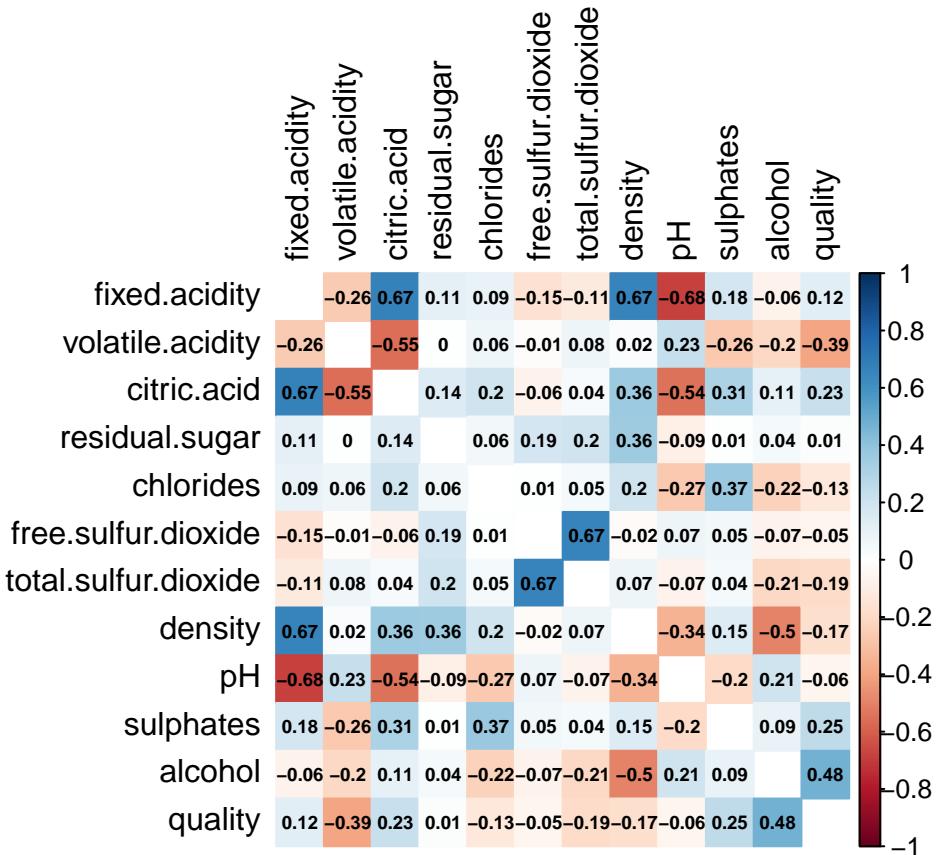
```
colSums(is.na(red_df))
```

```
##      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)
```

```
red_dfcor <- cor(red_df)
corrplot(red_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(red_df)
quality_cor <- dfcor[,12]
absoutcome_cor <- abs(quality_cor)
head(absoutcome_cor[order(absoutcome_cor, decreasing = TRUE)],12)

##           quality          alcohol      volatile.acidity
## 1.000000000 0.47616632 0.39055778
##       sulphates      citric.acid total.sulfur.dioxide
## 0.25139708 0.22637251 0.18510029
##       density      chlorides      fixed.acidity
## 0.17491923 0.12890656 0.12405165
##          pH free.sulfur.dioxide residual.sugar
## 0.05773139 0.05065606 0.01373164

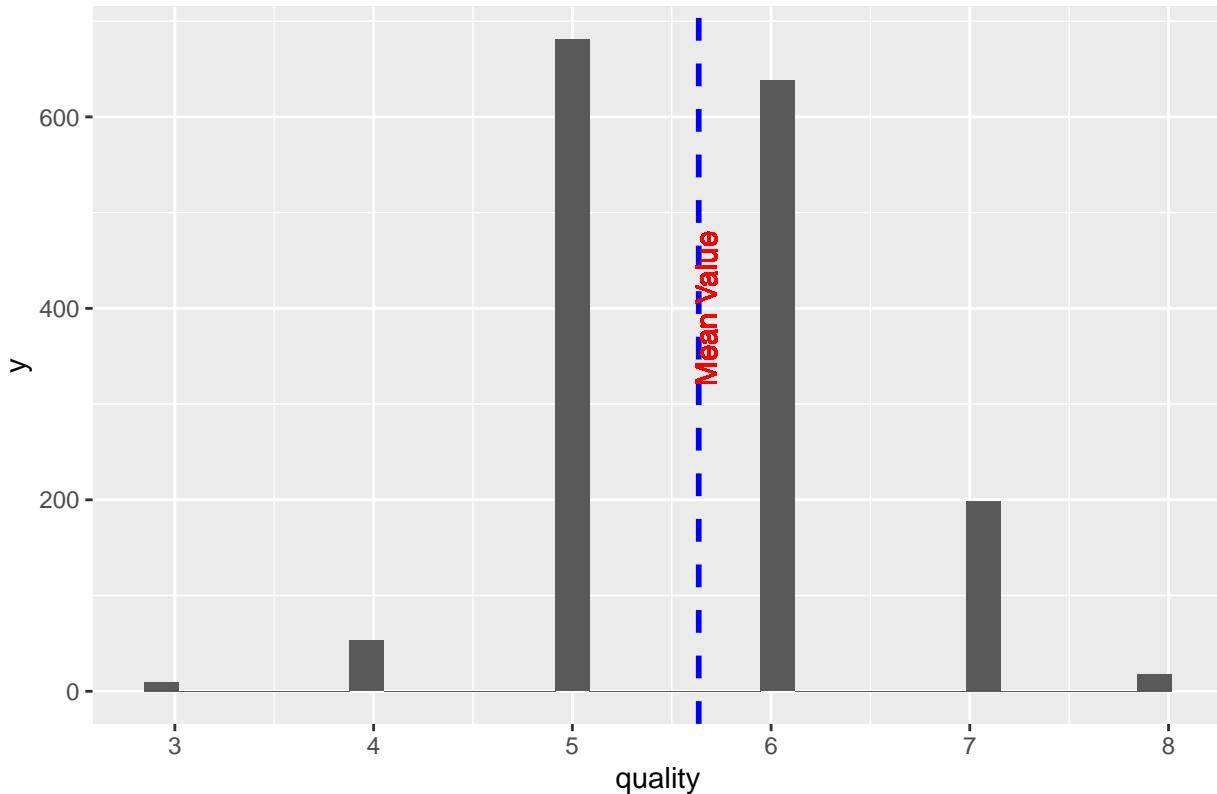
#Analysing the overall quality
ggplot(red_df, aes(quality))+
  geom_histogram() +
  labs(title = "Histogram of quality of Red wine") +
  theme(plot.title=element_text(hjust=0.5)) +
  geom_vline(aes(xintercept=mean(quality)), color="blue", linetype="dashed", size=1) +
  geom_text(aes(x=5.6, label="Mean 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'.

```

Histogram of quality of Red wine



```
mean(red_df$quality)

## [1] 5.636023

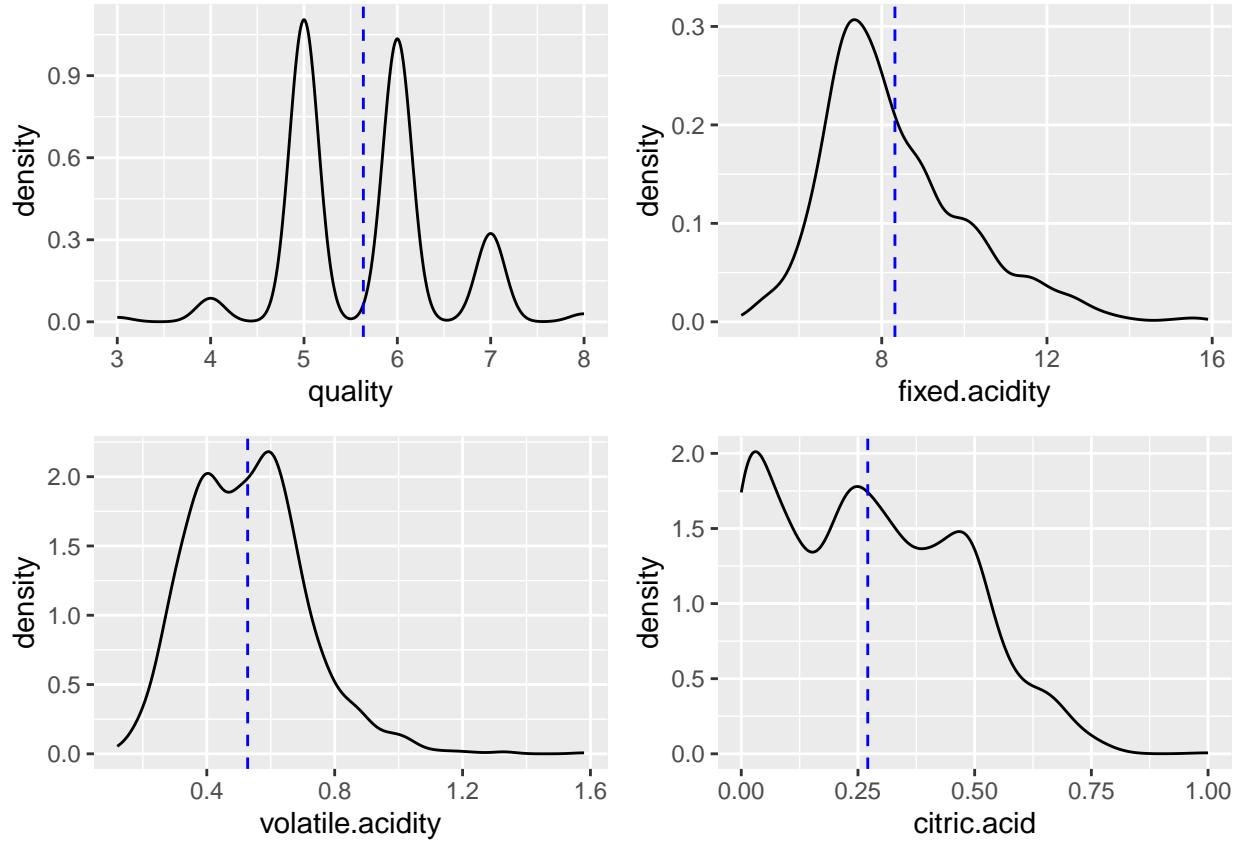
#Plotting the marginal distributions of numerical quantities of interest using density plots

d0 <- ggplot(red_df, aes(x=quality)) +
  geom_density()
d0 <- d0 + geom_vline(aes(xintercept=mean(quality)),
                      color="blue", linetype="dashed")
d1 <- ggplot(red_df, aes(x=fixed.acidity)) +
  geom_density()
d1 <- d1 + geom_vline(aes(xintercept=mean(fixed.acidity)),
                      color="blue", linetype="dashed")
d2 <- ggplot(red_df, aes(x=volatile.acidity)) +
  geom_density()
d2 <- d2 + geom_vline(aes(xintercept=mean(volatile.acidity)),
                      color="blue", linetype="dashed")
d3 <- ggplot(red_df, aes(x=citric.acid)) +
  geom_density()
d3 <- d3 + geom_vline(aes(xintercept=mean(citric.acid)),
                      color="blue", linetype="dashed")
d4 <- ggplot(red_df, aes(x=residual.sugar)) +
  geom_density()
d4 <- d4 + geom_vline(aes(xintercept=mean(residual.sugar)),
```

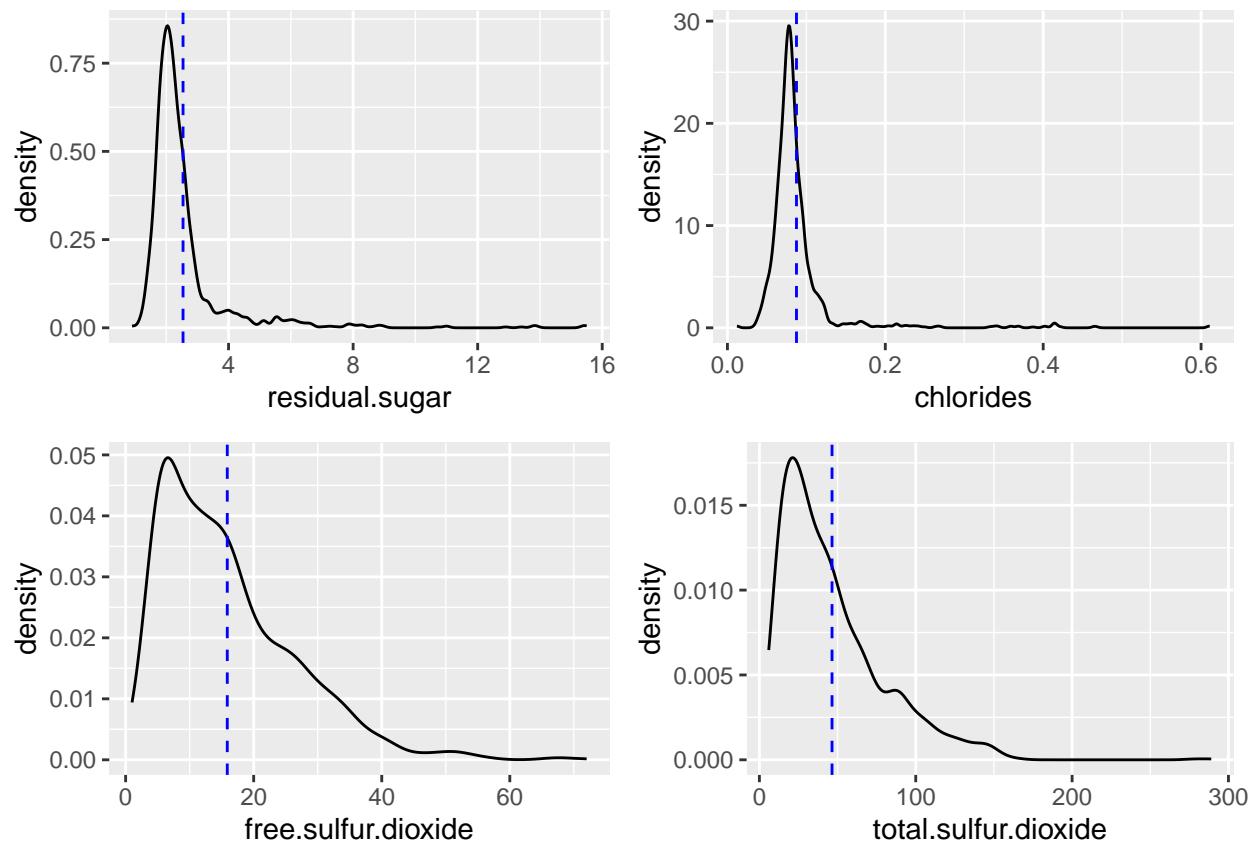
```

        color="blue", linetype="dashed")
d5 <- ggplot(red_df, aes(x=chlorides)) +
  geom_density()
d5 <- d5 + geom_vline(aes(xintercept=mean(chlorides)),
                      color="blue", linetype="dashed")
d6 <- ggplot(red_df, aes(x=free.sulfur.dioxide)) +
  geom_density()
d6 <- d6 + geom_vline(aes(xintercept=mean(free.sulfur.dioxide)),
                      color="blue", linetype="dashed")
d7 <- ggplot(red_df, aes(x=total.sulfur.dioxide)) +
  geom_density()
d7 <- d7 + geom_vline(aes(xintercept=mean(total.sulfur.dioxide)),
                      color="blue", linetype="dashed")
d8 <- ggplot(red_df, aes(x=density)) +
  geom_density()
d8 <- d8 + geom_vline(aes(xintercept=mean(density)),
                      color="blue", linetype="dashed")
d9 <- ggplot(red_df, aes(x=pH)) +
  geom_density()
d9 <- d9 + geom_vline(aes(xintercept=mean(pH)),
                      color="blue", linetype="dashed")
d10 <- ggplot(red_df, aes(x=sulphates)) +
  geom_density()
d10 <- d10 + geom_vline(aes(xintercept=mean(sulphates)),
                        color="blue", linetype="dashed")
d11 <- ggplot(red_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)

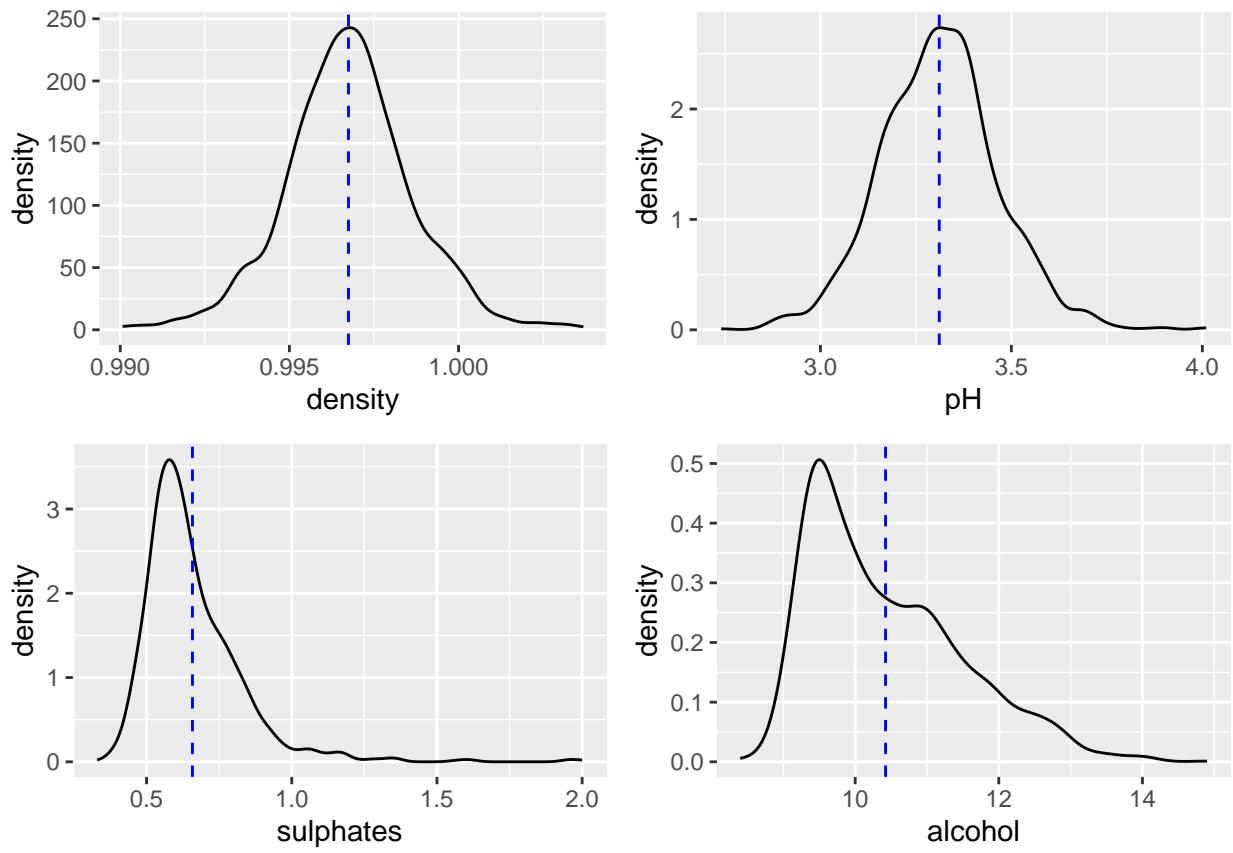
```



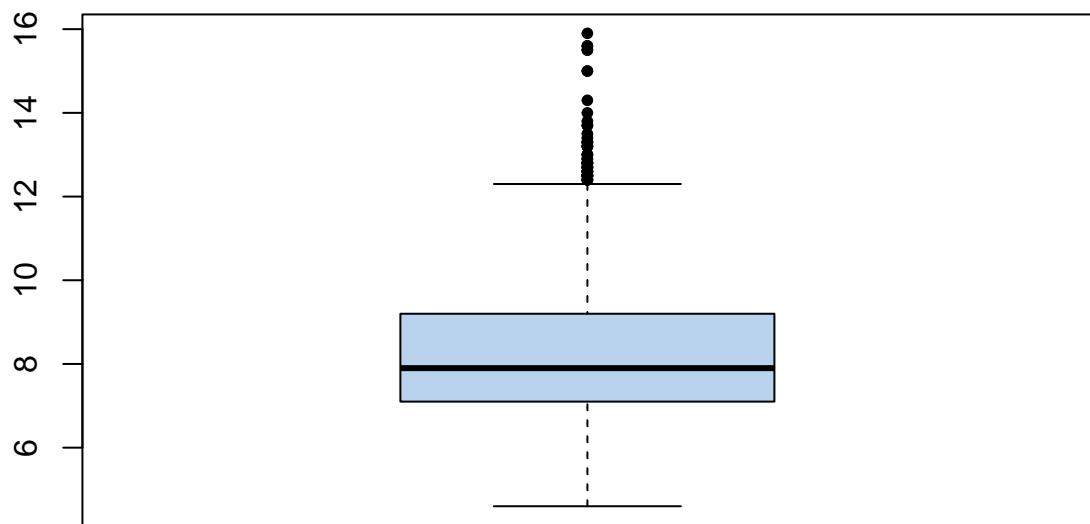
```
ggarrange(d4, d5, d6, d7, nrow = 2, ncol =2)
```



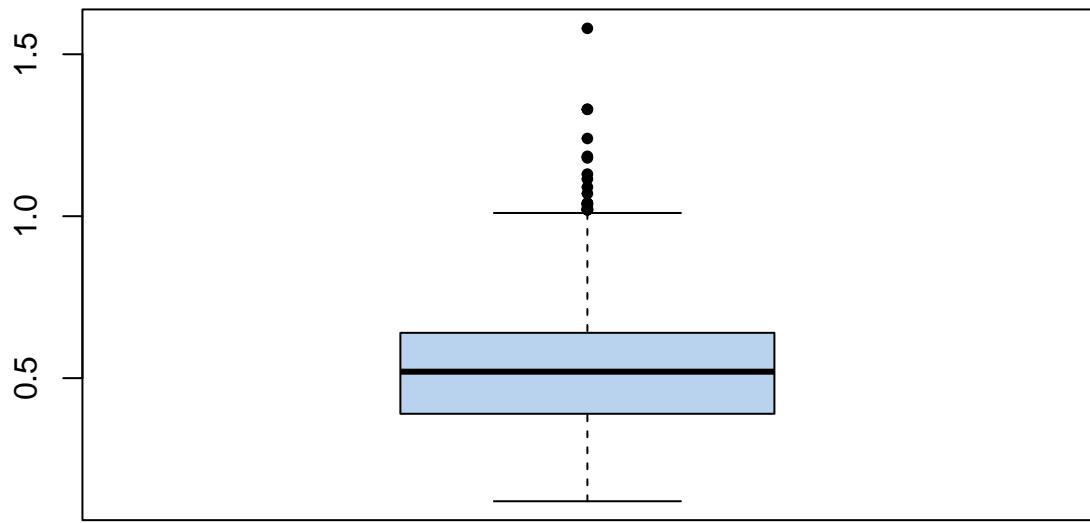
```
ggarrange(d8, d9, d10, d11, nrow = 2, ncol =2)
```



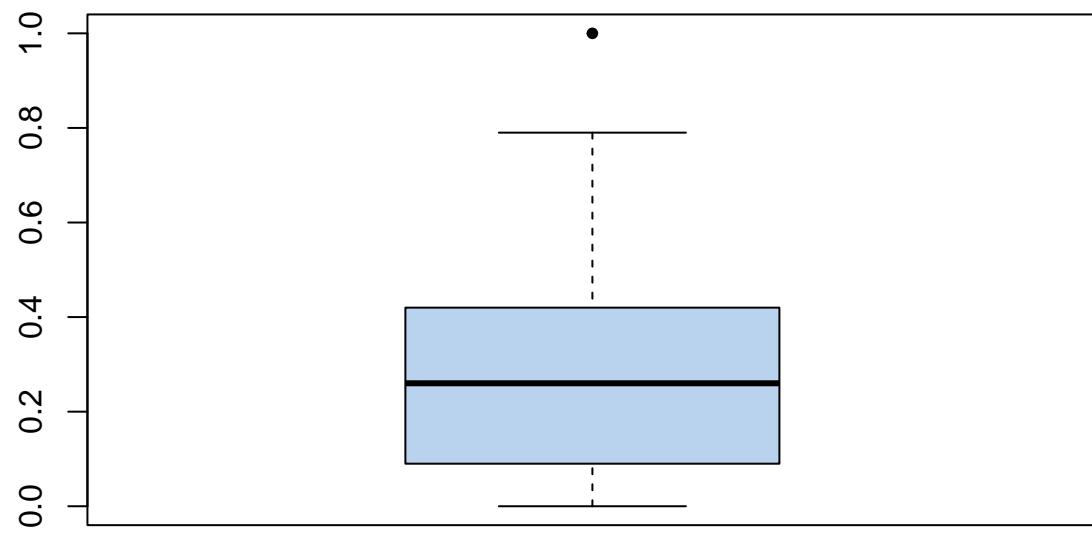
```
#Plotting the marginal distributions of numerical quantities of interest using box plots
b1 <- boxplot(red_df$fixed.acidity, col="slategray2", pch=20)
```



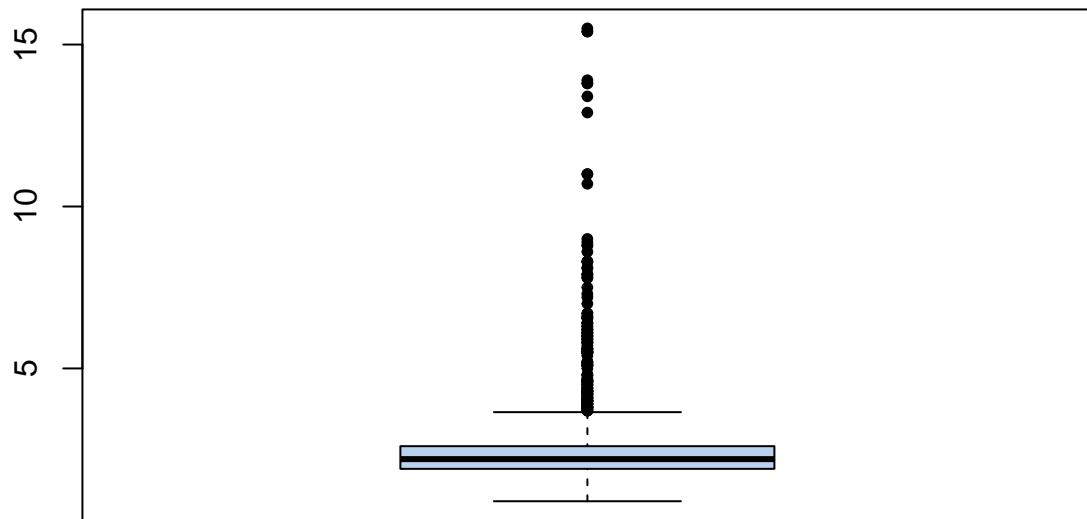
```
b2 <- boxplot(red_df$volatile.acidity, col="slategray2", pch=20)
```



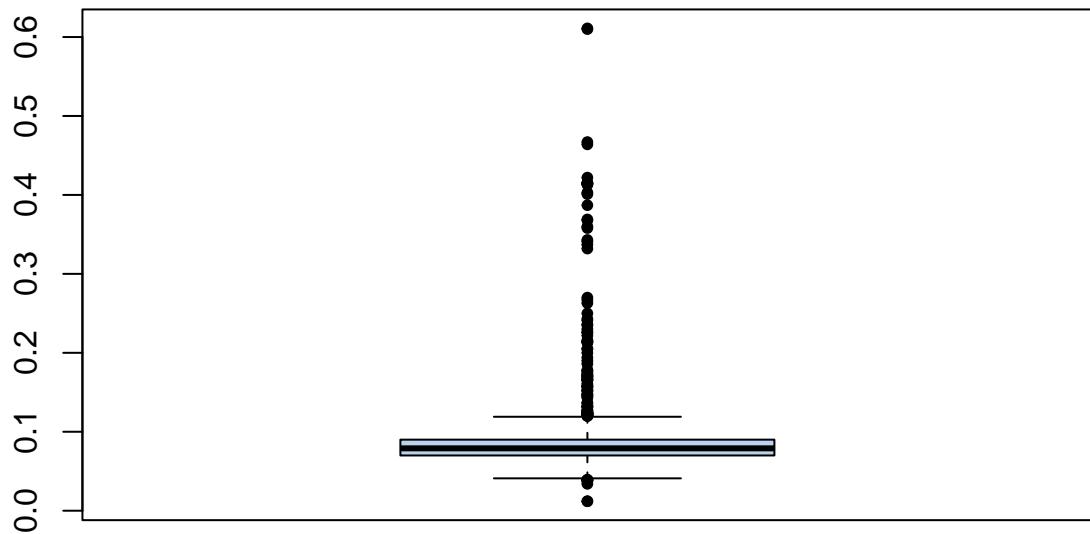
```
b3 <- boxplot(red_df$citric.acid, col="slategray2", pch=20)
```



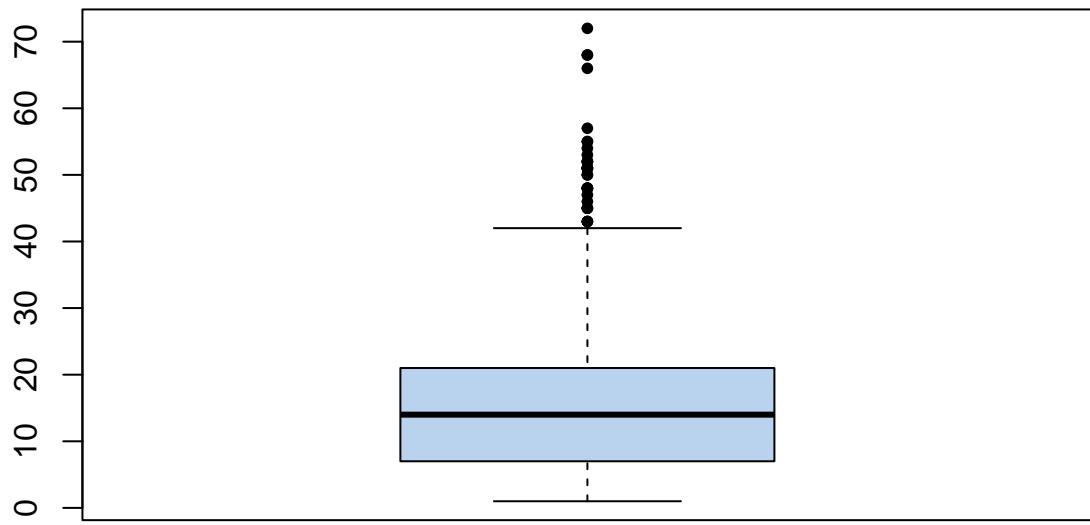
```
b4 <- boxplot(red_df$residual.sugar, col="slategray2", pch=20)
```



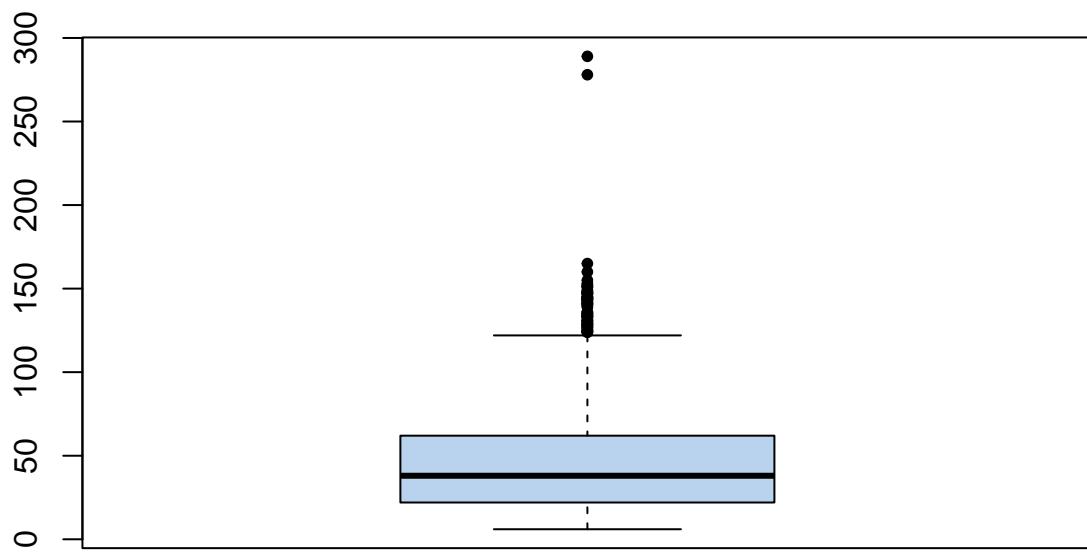
```
b5 <- boxplot(red_df$chlorides, col="slategray2", pch=20)
```



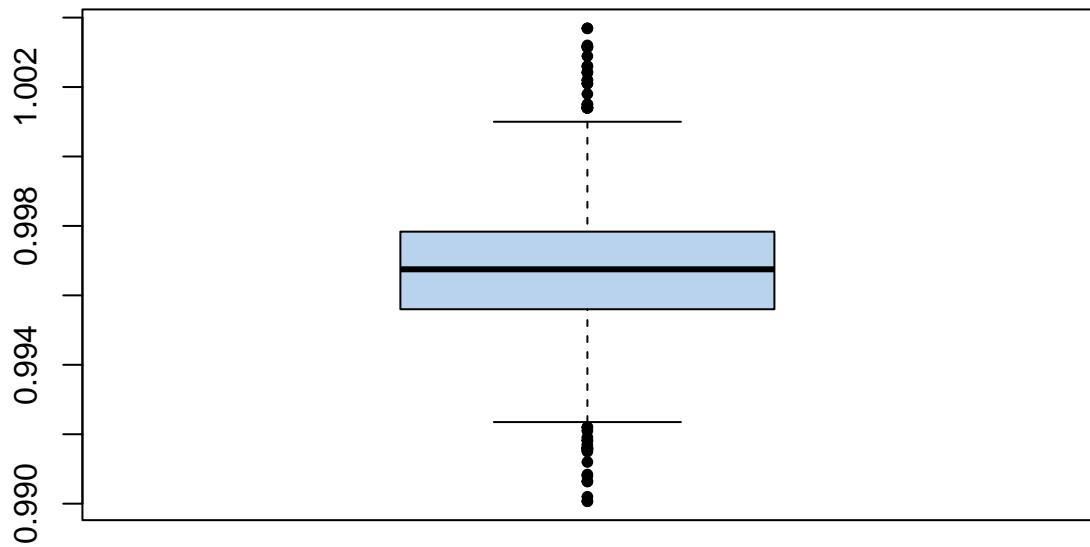
```
b6 <- boxplot(red_df$free.sulfur.dioxide, col="slategray2", pch=20)
```



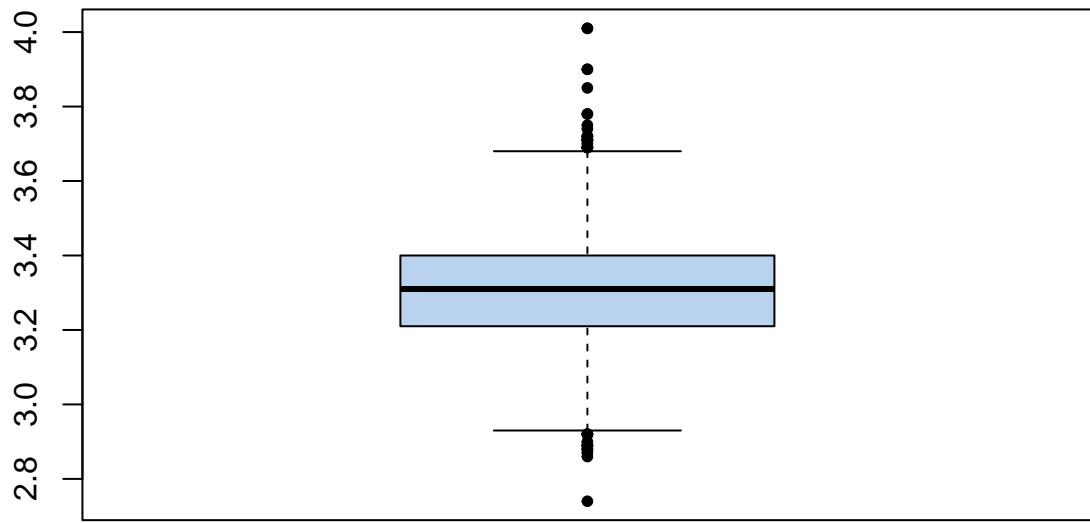
```
b7 <- boxplot(red_df$total.sulfur.dioxide, col="slategray2", pch=20)
```



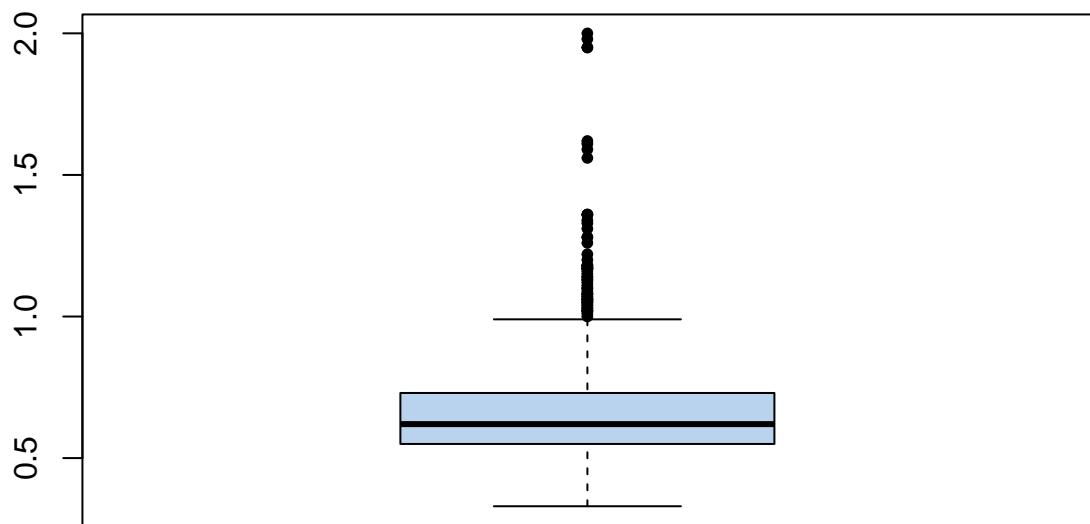
```
b8 <- boxplot(red_df$density, col="slategray2", pch=20)
```



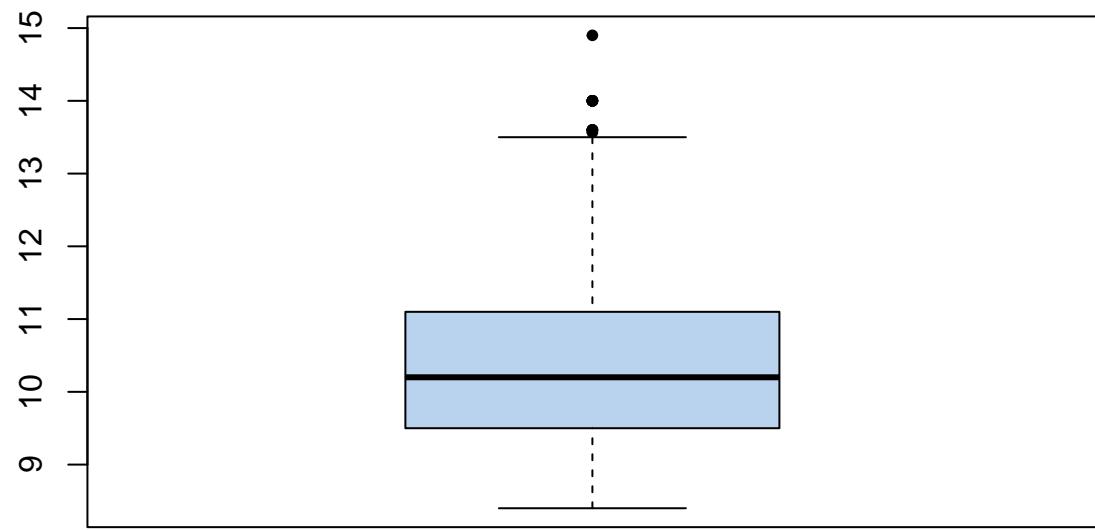
```
b9 <- boxplot(red_df$pH, col="slategray2", pch=20)
```



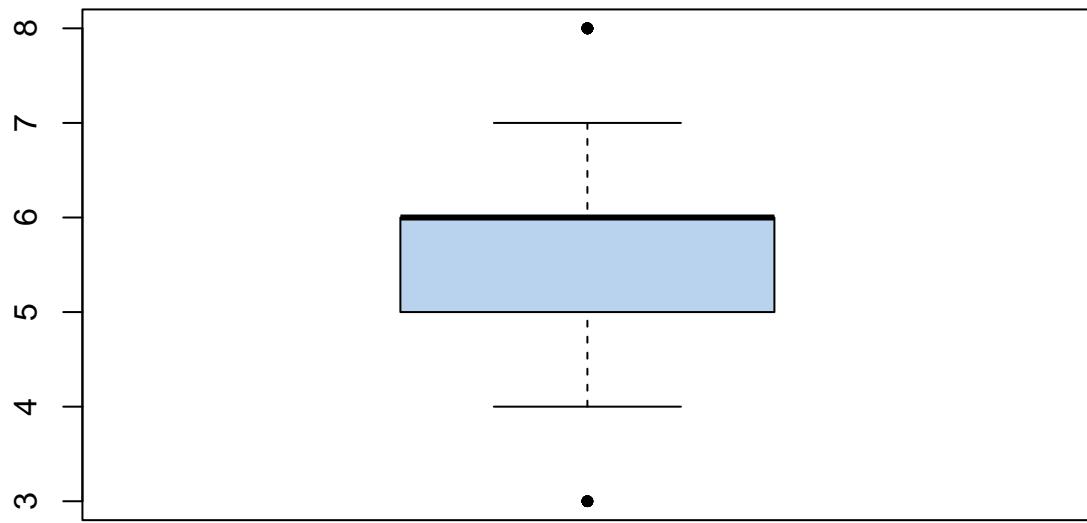
```
b10 <- boxplot(red_df$sulphates, col="slategray2", pch=20)
```



```
b11 <- boxplot(red_df$alcohol, col="slategray2", pch=20)
```

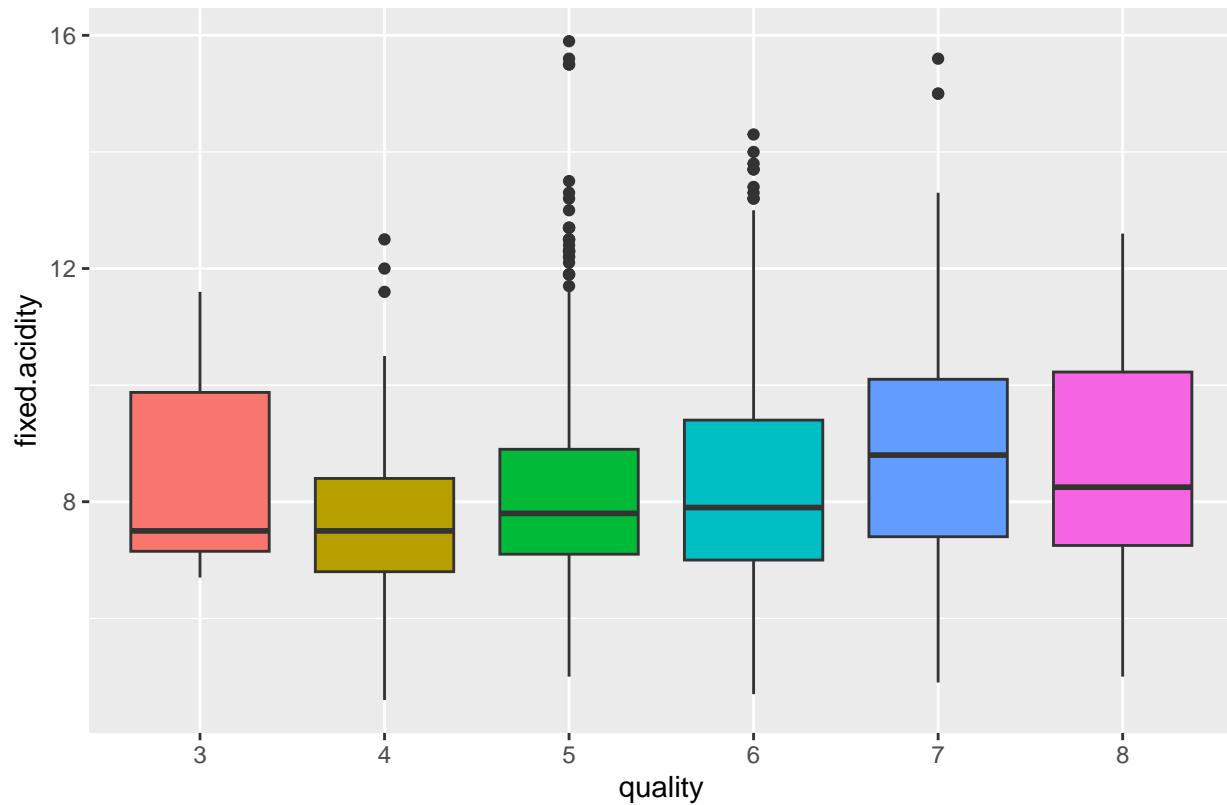


```
b12 <- boxplot(red_df$quality, col="slategray2", pch=20)
```



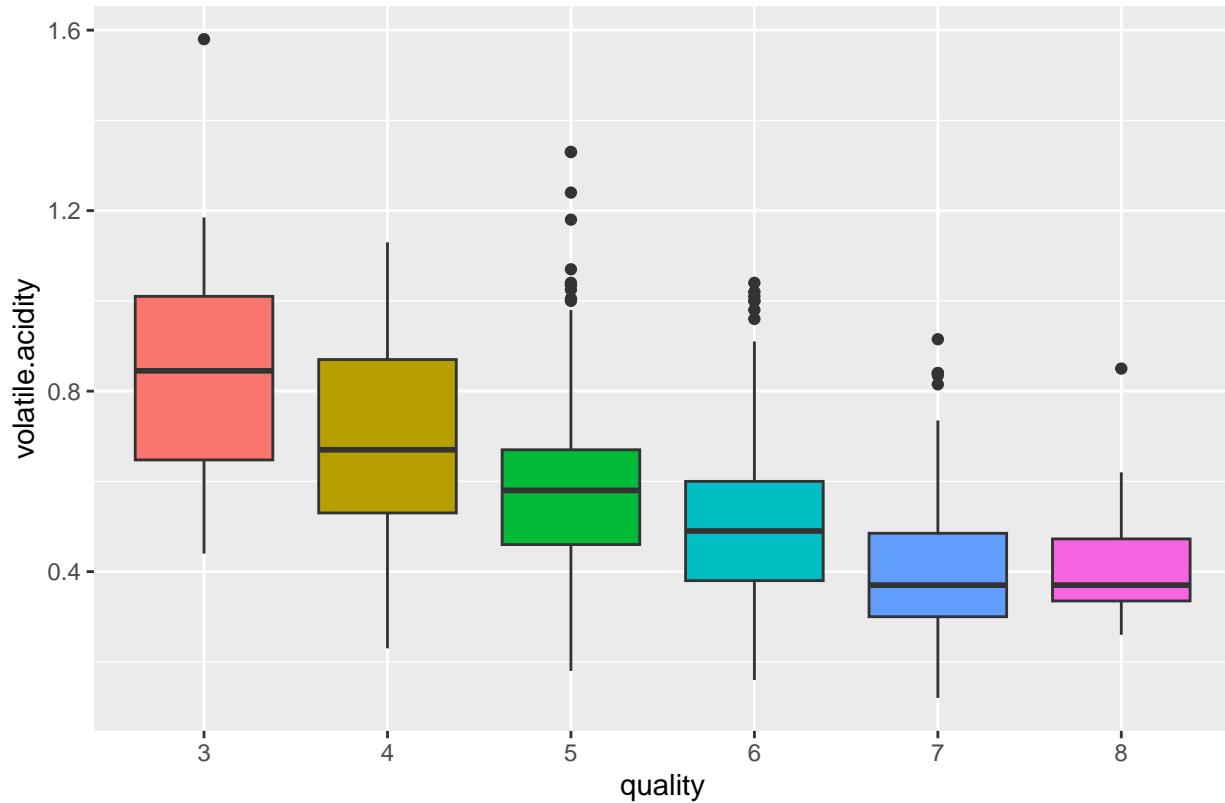
```
#Plotting quality against numerical variables
bp1 <- ggplot(red_df, aes(factor(quality), fixed.acidity, fill=factor(quality))) +
  geom_boxplot() +
  labs(x = "quality", y = "fixed.acidity", title = "Boxplot of Quality vs. fixed.acidity") +
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))
bp1
```

Boxplot of Quality vs. fixed.acidity



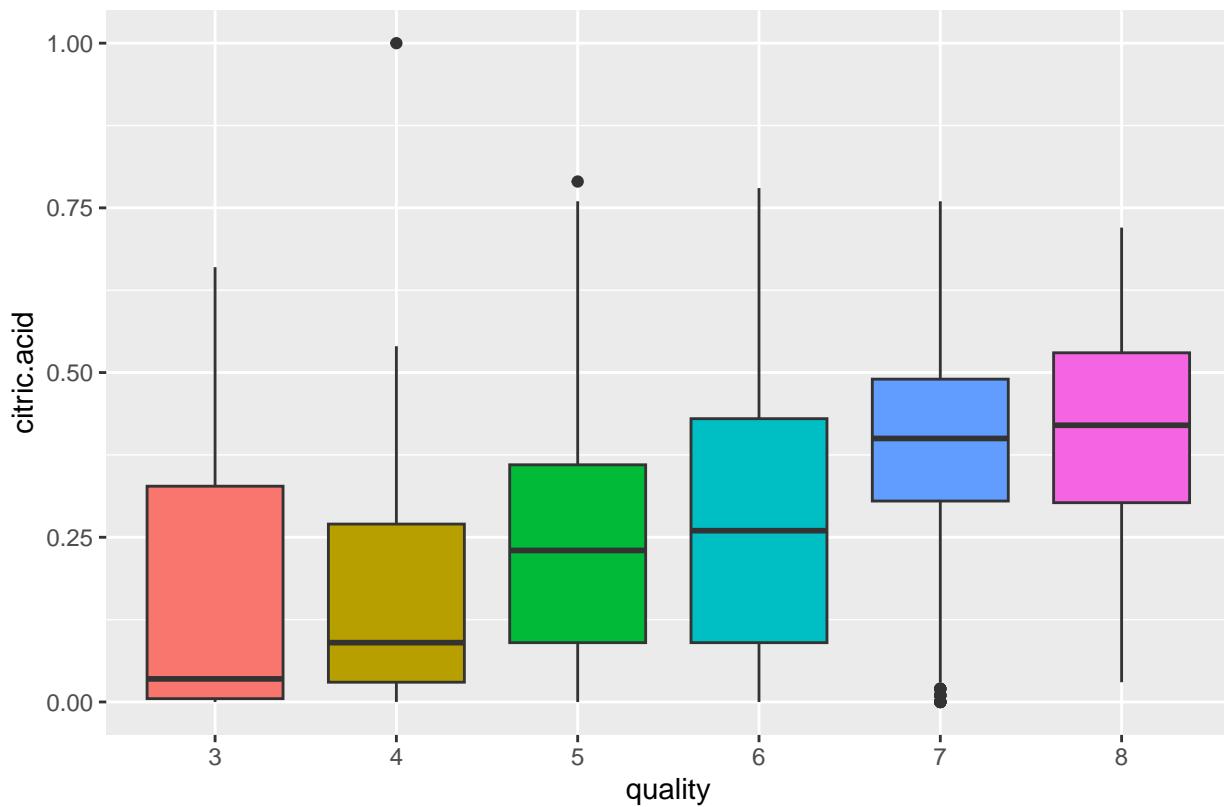
```
bp2 <- ggplot(red_df, aes(factor(quality), volatile.acidity, fill=factor(quality))) +  
  geom_boxplot() +  
  labs(x = "quality", y = "volatile.acidity", title = "Boxplot of Quality vs. volatile.acidity") +  
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))  
bp2
```

Boxplot of Quality vs. volatile.acidity



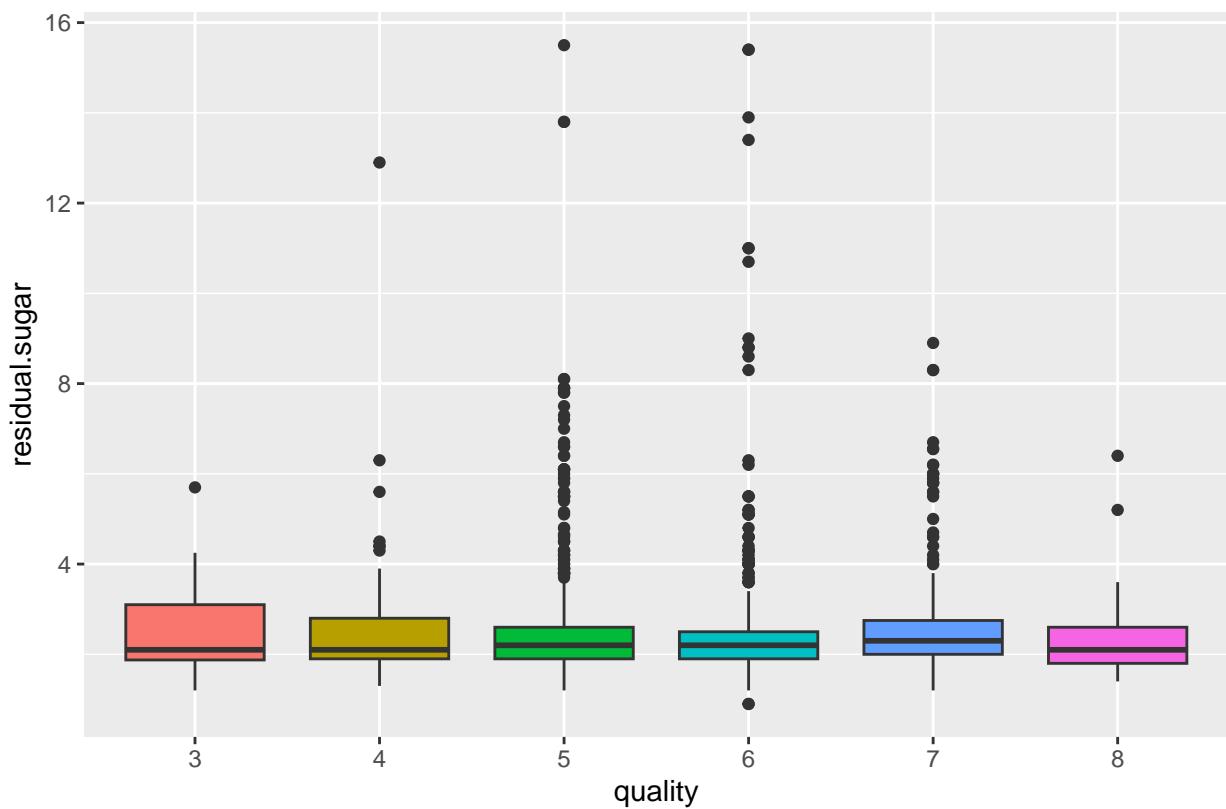
```
bp3 <- ggplot(red_df, aes(factor(quality), citric.acid, fill=factor(quality))) +  
  geom_boxplot() +  
  labs(x = "quality", y = "citric.acid", title = "Boxplot of Quality vs. citric.acid") +  
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))  
bp3
```

Boxplot of Quality vs. citric.acid

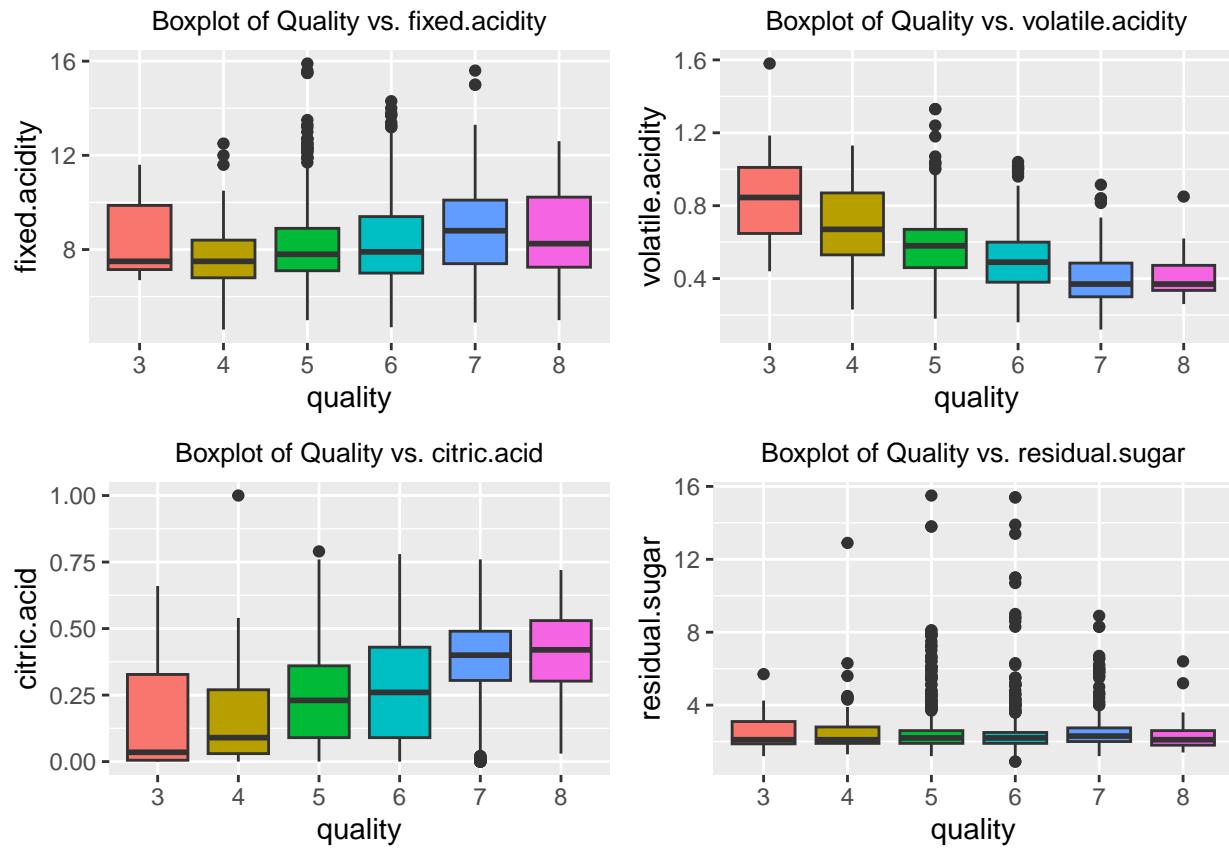


```
bp4 <- ggplot(red_df, aes(factor(quality), residual.sugar, fill=factor(quality))) +  
  geom_boxplot() +  
  labs(x = "quality", y = "residual.sugar", title = "Boxplot of Quality vs. residual.sugar") +  
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))  
bp4
```

Boxplot of Quality vs. residual.sugar

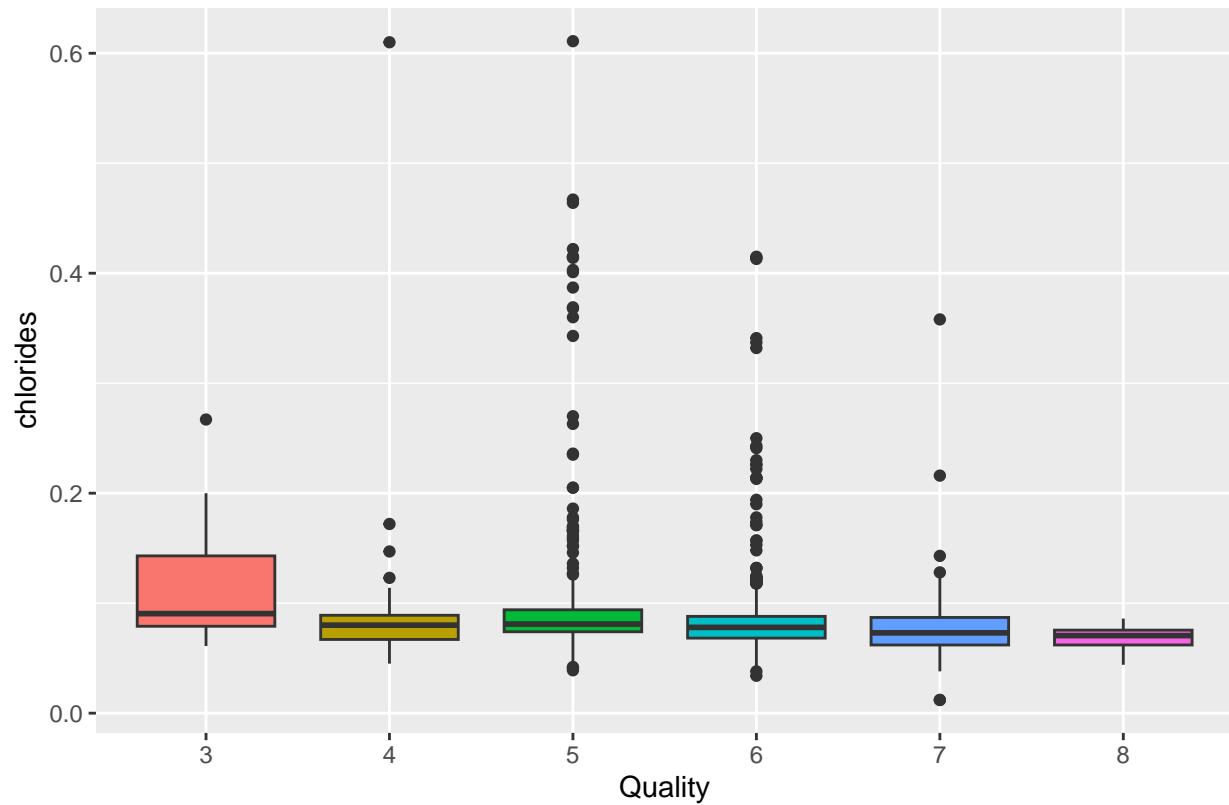


```
ggarrange(bp1, bp2, bp3, bp4, nrow = 2, ncol =2)
```



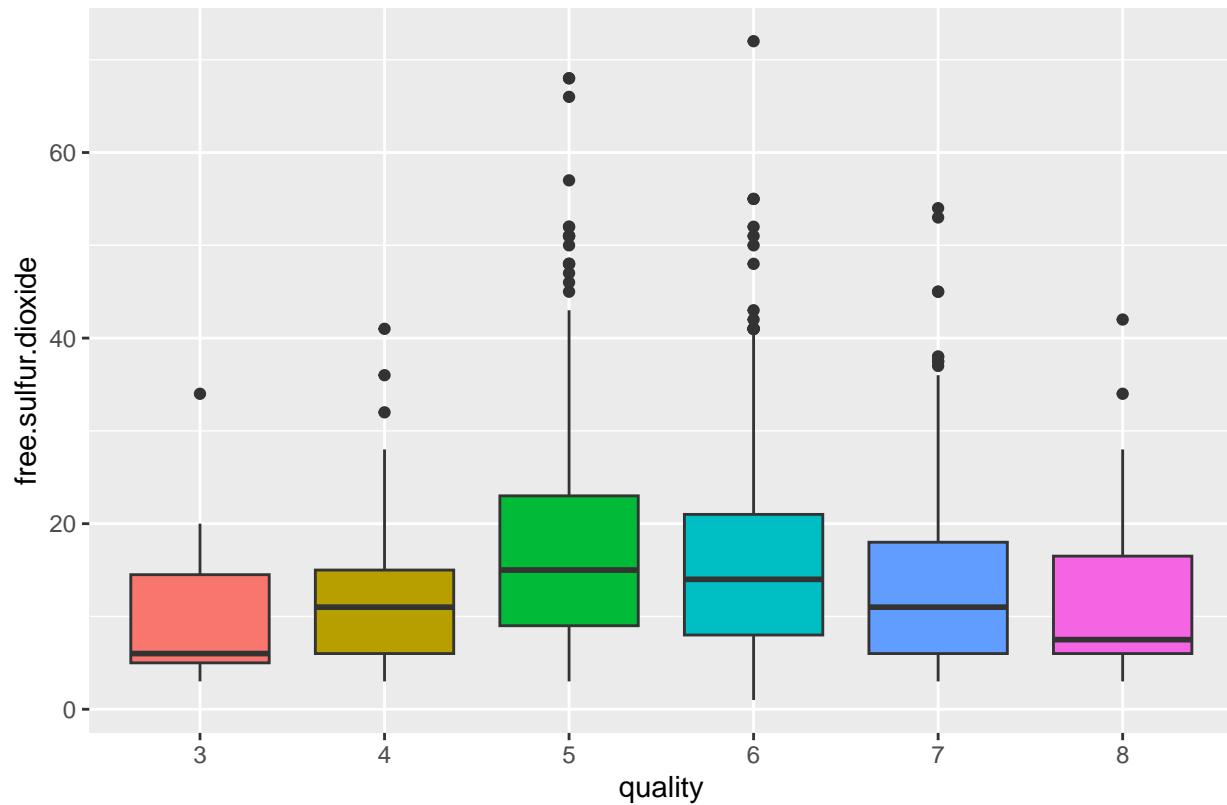
```
bp5 <- ggplot(red_df, aes(factor(quality), chlorides, fill=factor(quality))) +
  geom_boxplot() +
  labs(x = "Quality", y = "chlorides", title = "Boxplot of Quality vs. chlorides") +
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))
bp5
```

Boxplot of Quality vs. chlorides

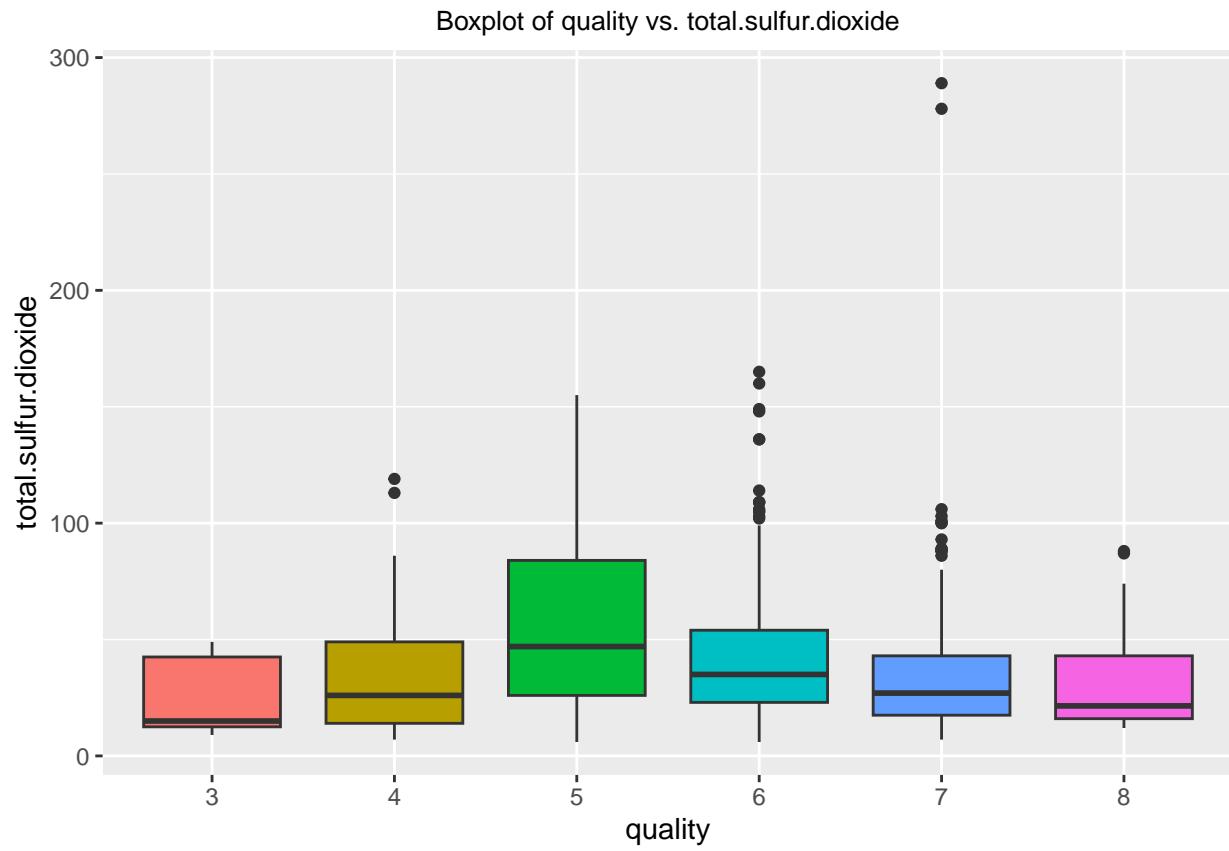


```
bp6 <- ggplot(red_df, aes(factor(quality), free.sulfur.dioxide, fill=factor(quality))) +  
  geom_boxplot() +  
  labs(x = "quality", y = "free.sulfur.dioxide", title = "Boxplot of quality vs. free.sulfur.dioxide") +  
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))  
bp6
```

Boxplot of quality vs. free.sulfur.dioxide

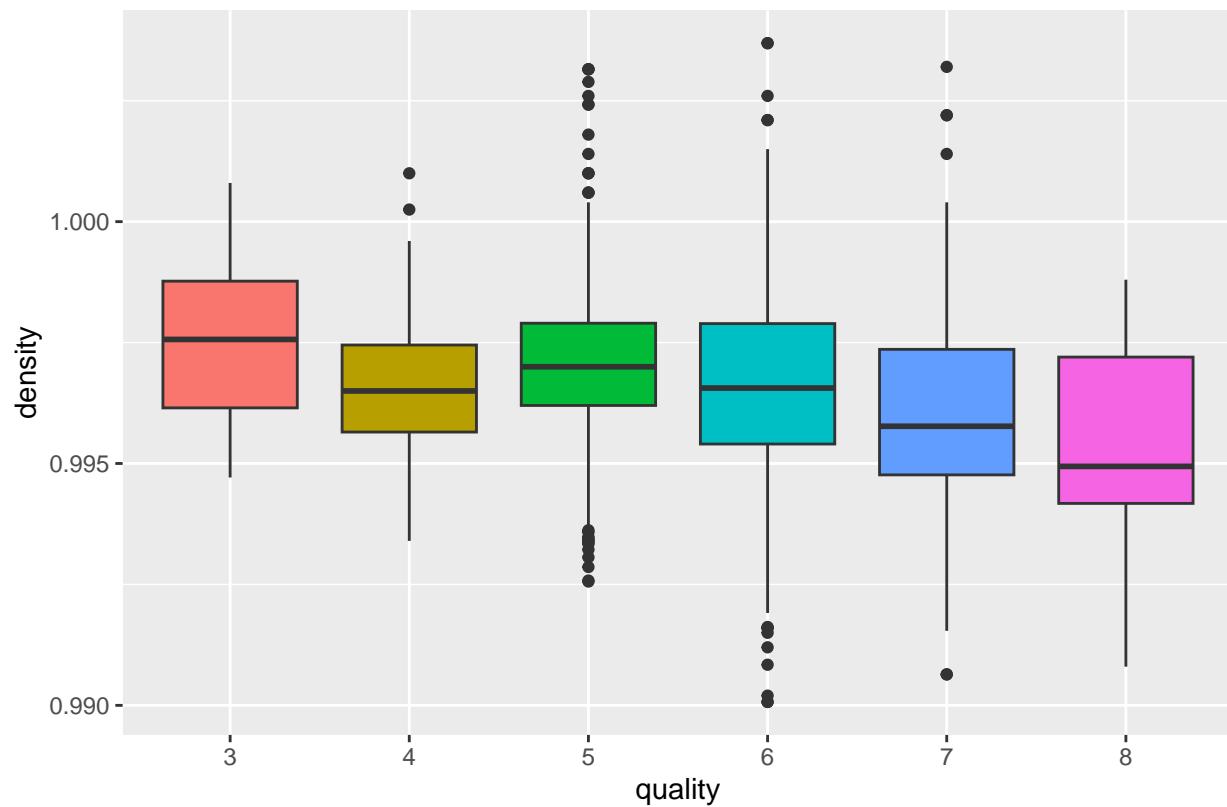


```
bp7 <- ggplot(red_df, aes(factor(quality), total.sulfur.dioxide, fill=factor(quality))) +  
  geom_boxplot() +  
  labs(x = "quality", y = "total.sulfur.dioxide", title = "Boxplot of quality vs. total.sulfur.dioxide")  
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))  
bp7
```

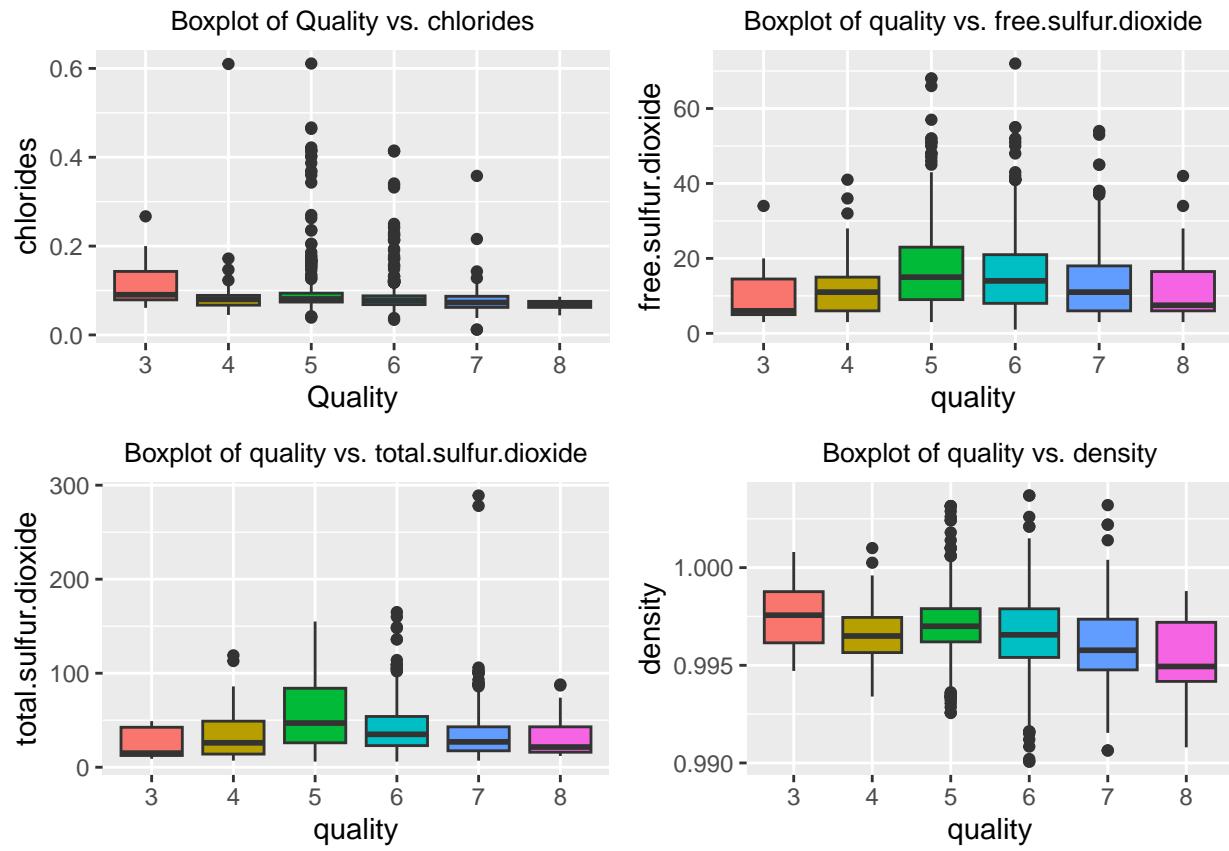


```
bp8 <- ggplot(red_df, aes(factor(quality), density, fill=factor(quality))) +
  geom_boxplot() +
  labs(x = "quality", y = "density", title = "Boxplot of quality vs. density") +
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))
bp8
```

Boxplot of quality vs. density

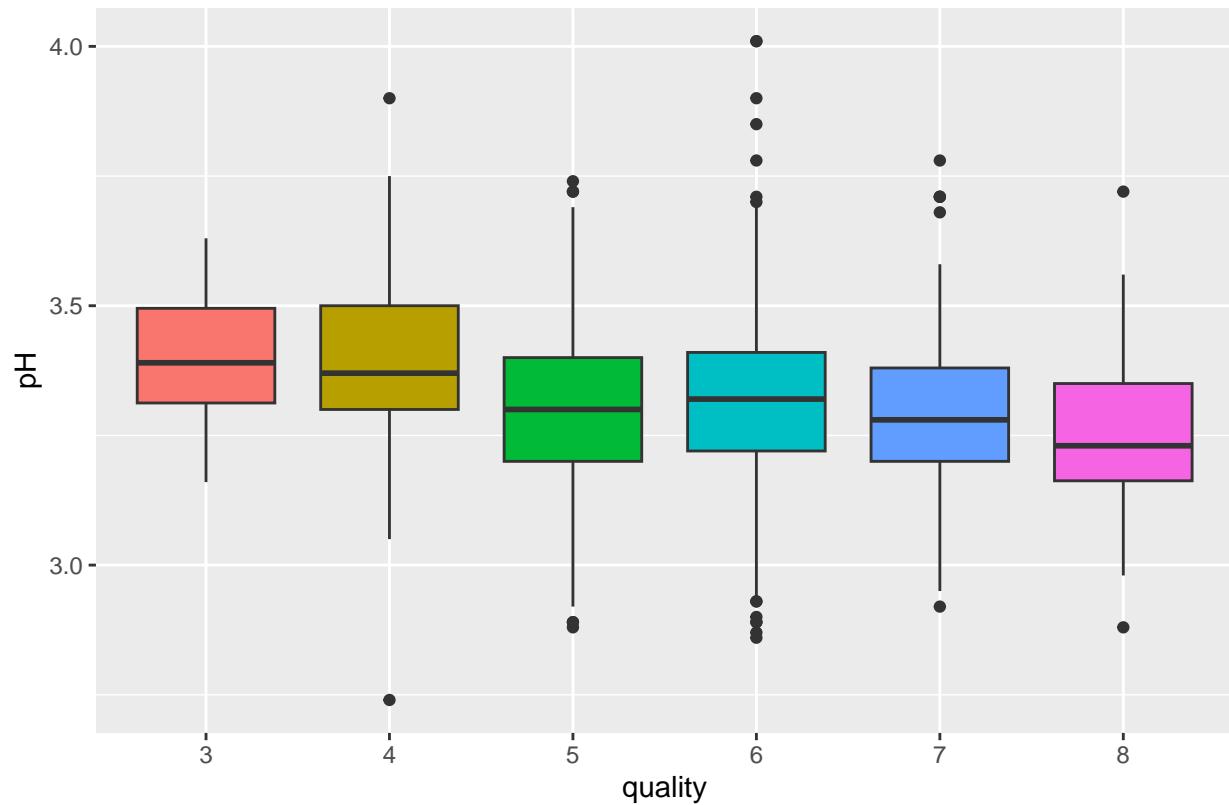


```
ggarrange(bp5, bp6, bp7, bp8, nrow = 2, ncol =2)
```



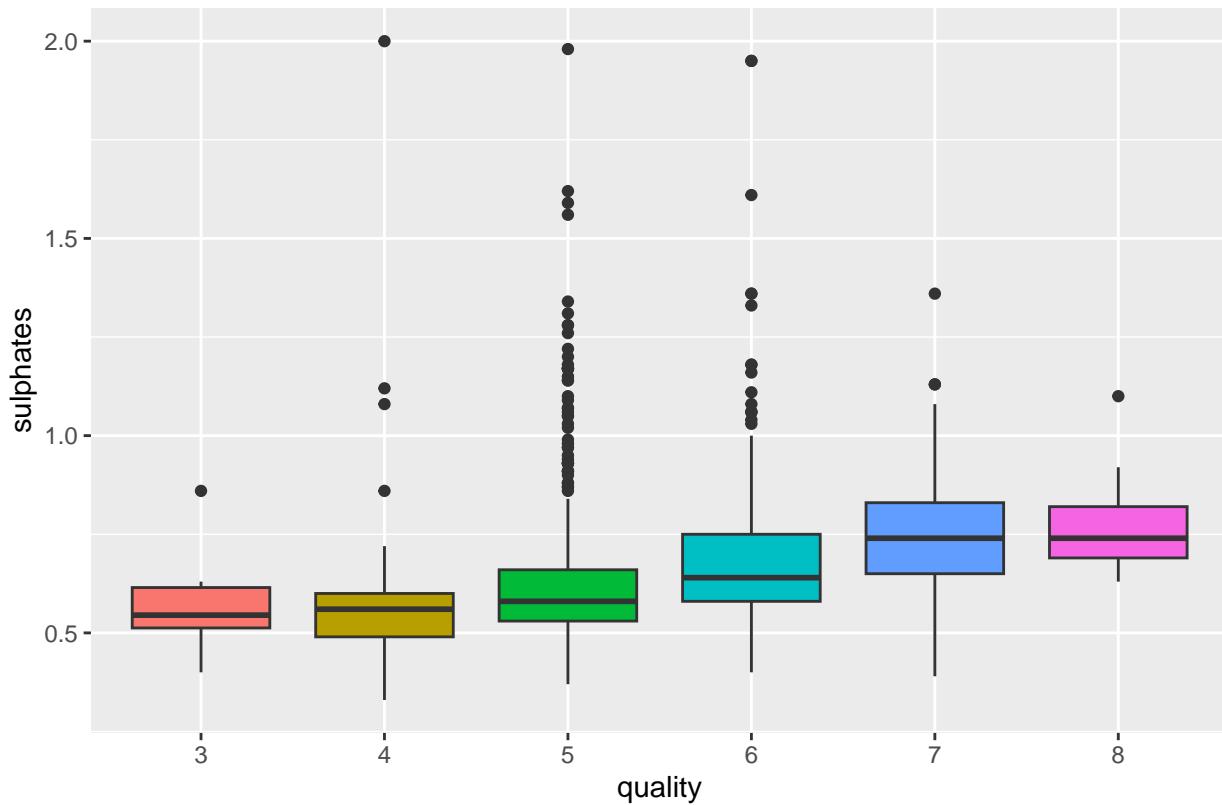
```
bp9 <- ggplot(red_df, aes(factor(quality), pH, fill=factor(quality))) +
  geom_boxplot() +
  labs(x = "quality", y = "pH", title = "Boxplot of Quality vs. pH") +
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))
bp9
```

Boxplot of Quality vs. pH



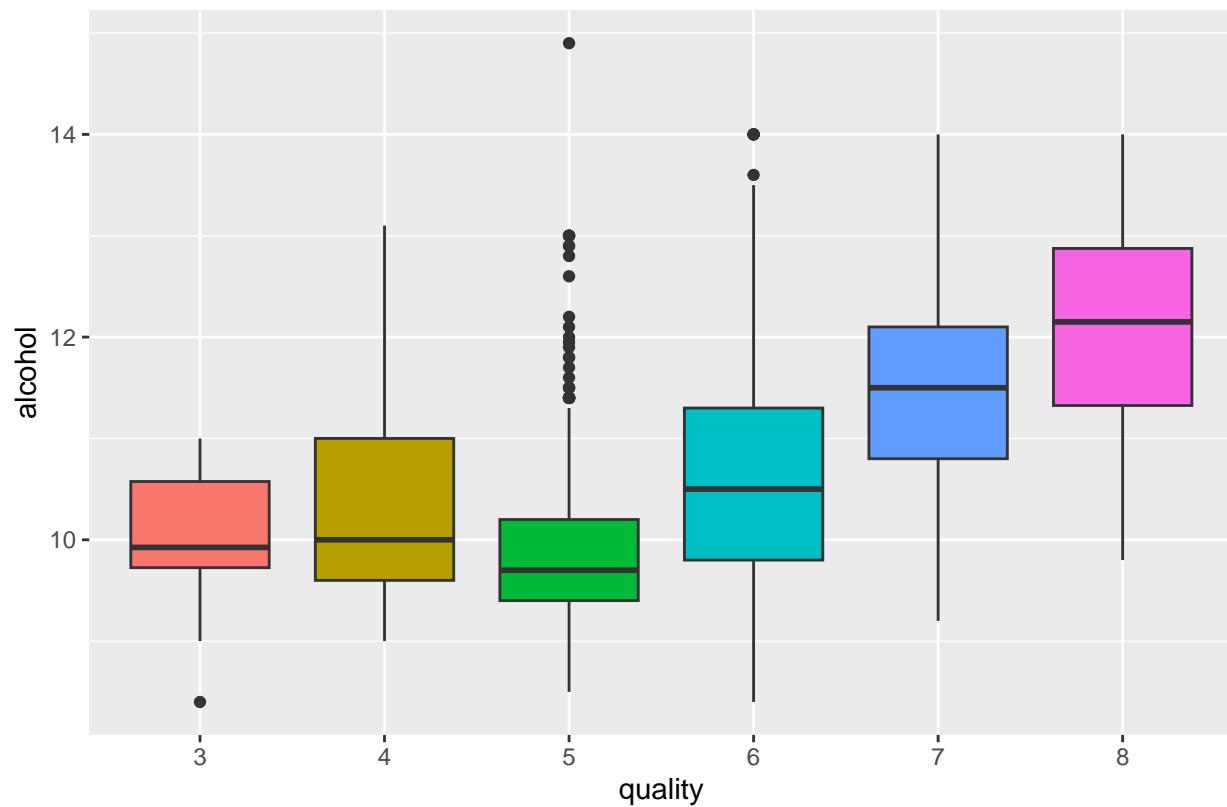
```
bp10 <- ggplot(red_df, aes(factor(quality), sulphates, fill=factor(quality))) +  
  geom_boxplot() +  
  labs(x = "quality", y = "sulphates", title = "Boxplot of quality vs. sulphates") +  
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))  
bp10
```

Boxplot of quality vs. sulphates

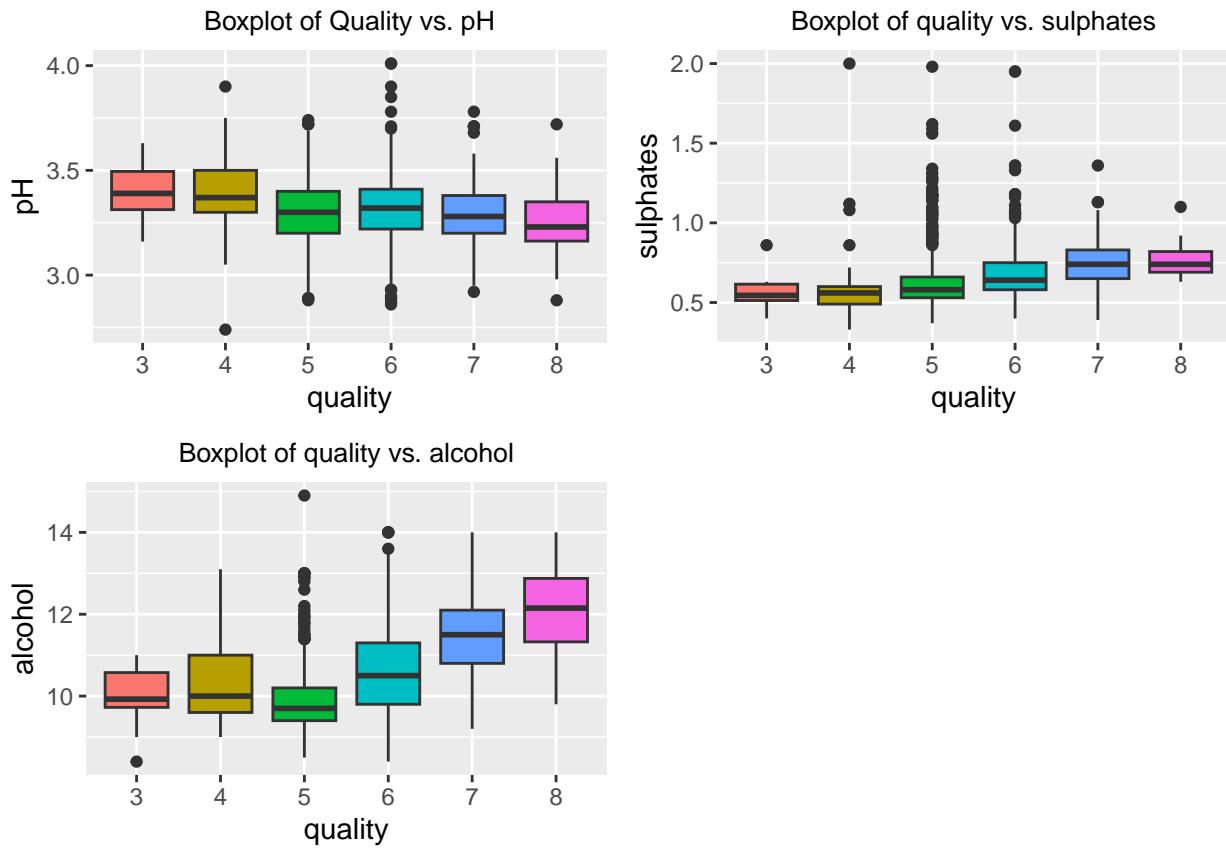


```
bp11 <- ggplot(red_df, aes(factor(quality), alcohol, fill=factor(quality))) +  
  geom_boxplot() +  
  labs(x = "quality", y = "alcohol", title = "Boxplot of quality vs. alcohol") +  
  theme(legend.position = 'none', plot.title = element_text(size = 10, hjust=0.5))  
bp11
```

Boxplot of quality vs. alcohol

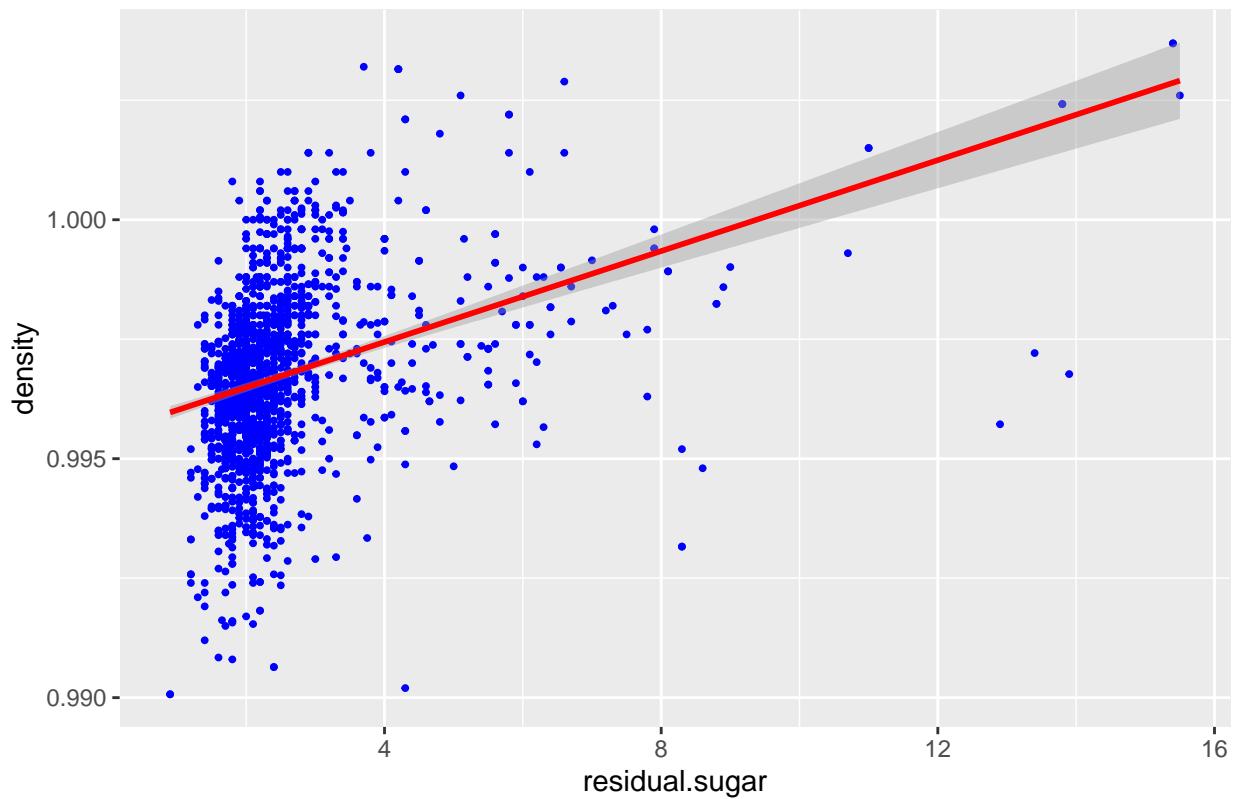


```
ggarrange(bp9, bp10, bp11, nrow = 2, ncol =2)
```



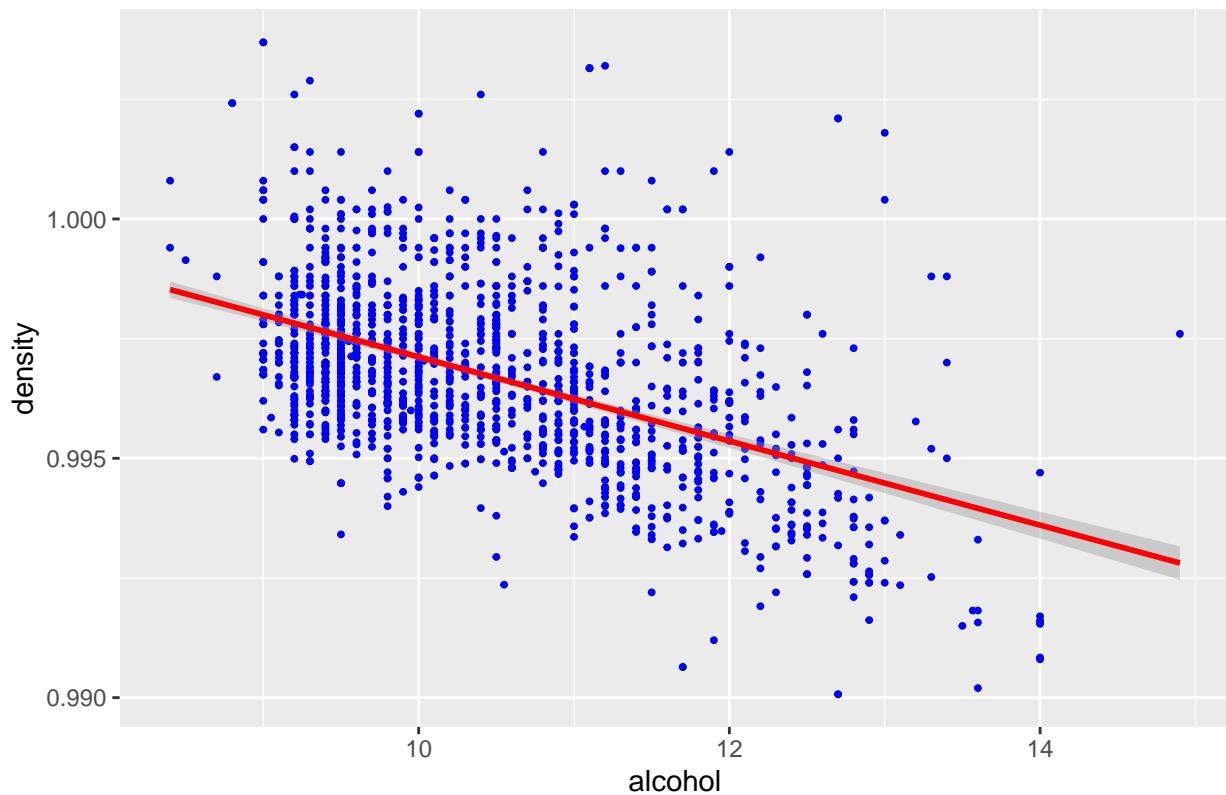
```
#Analysing relationship among numerical variables
gg1 <- ggplot(red_df, aes(x=residual.sugar, y=density)) +
  geom_point(color="blue", size=0.7) +
  labs(title="density vs. residual sugar") +
  geom_smooth(formula=y~x, method=lm, color="red") +
  theme(plot.title=element_text(hjust=0.5))
gg1
```

density vs. residual sugar

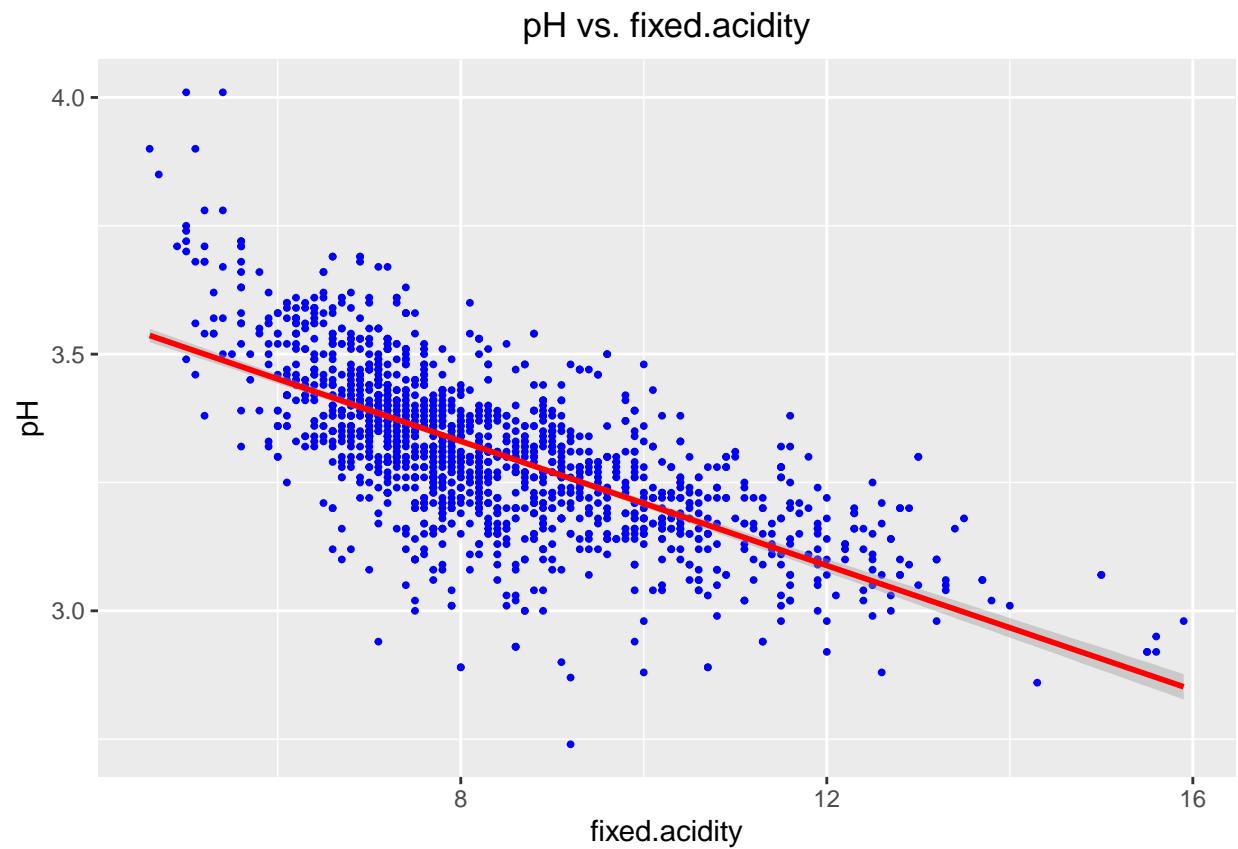


```
gg2 <- ggplot(red_df, aes(x=alcohol, y=density)) +  
  geom_point(color="blue", size=0.7) +  
  labs(title="density vs. alcohol") +  
  geom_smooth(formula=y~x, method=lm, color="red") +  
  theme(plot.title=element_text(hjust=0.5))  
gg2
```

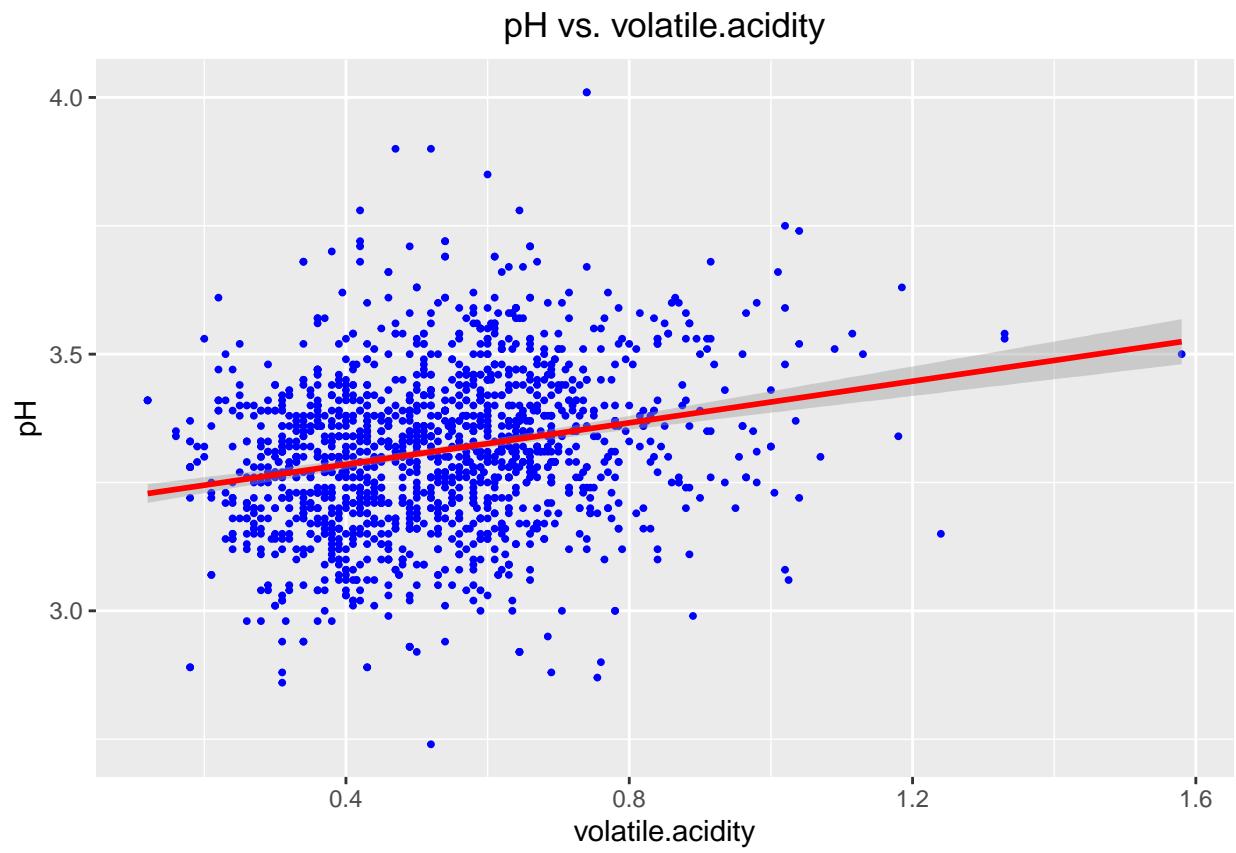
density vs. alcohol



```
gg3 <- ggplot(red_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
```

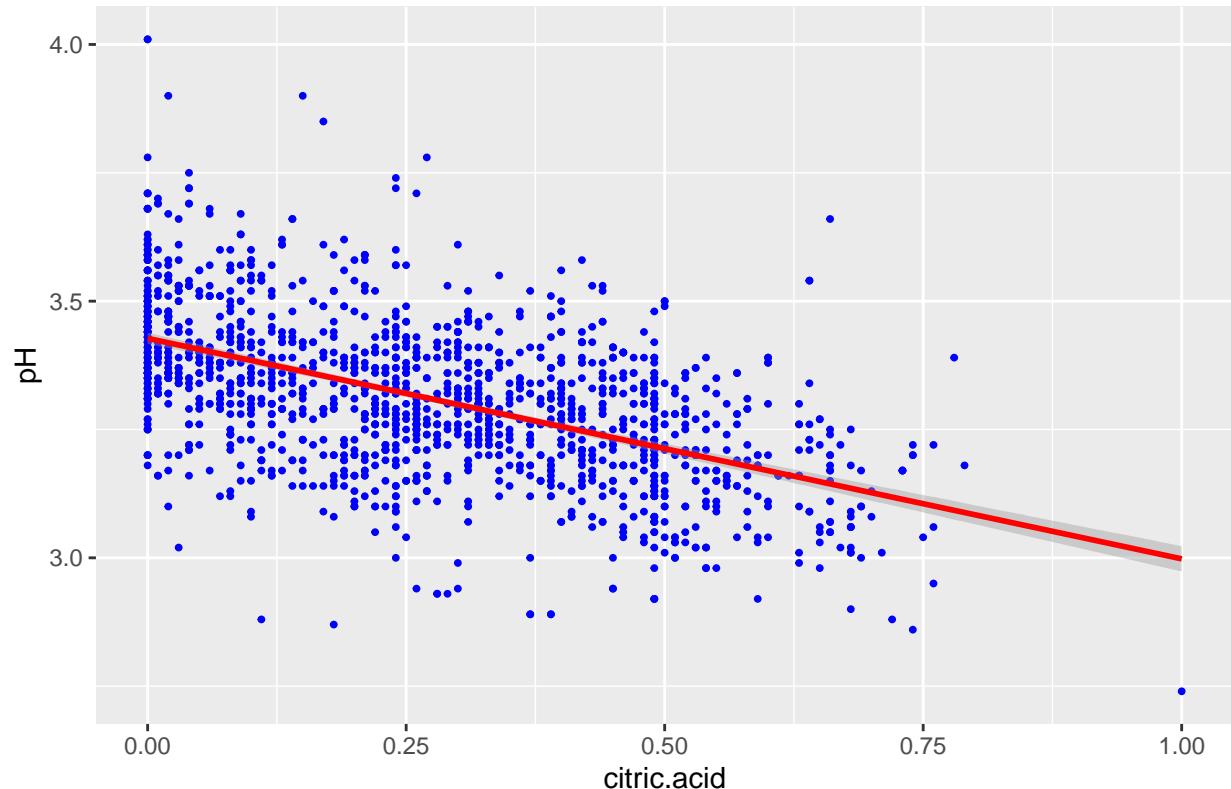


```
gg4 <- ggplot(red_df, aes(x=volatile.acidity, y=pH)) +  
  geom_point(color="blue", size=0.7) +  
  labs(title="pH vs. volatile.acidity") +  
  geom_smooth(formula=y~x, method=lm, color="red") +  
  theme(plot.title=element_text(hjust=0.5))  
gg4
```



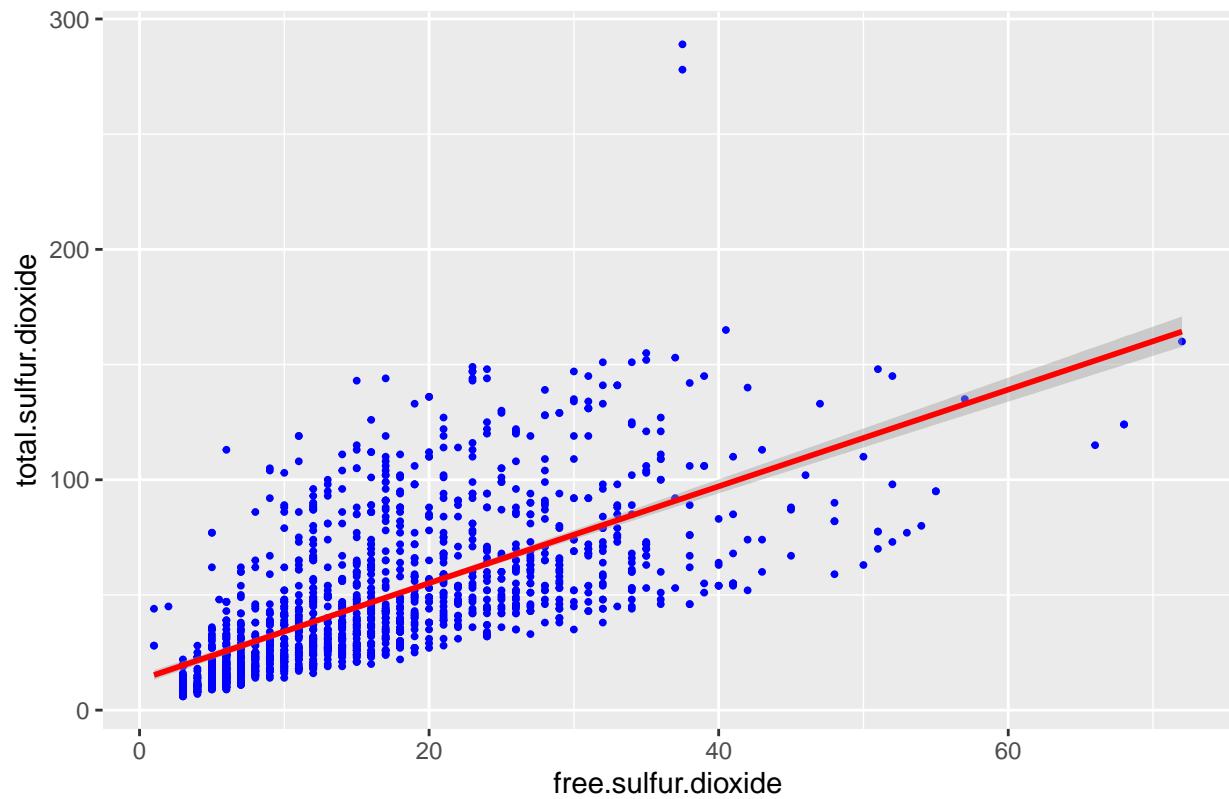
```
gg5 <- ggplot(red_df, aes(x=citric.acid, y=pH)) +  
  geom_point(color="blue", size=0.7) +  
  labs(title="pH vs. citric.acid") +  
  geom_smooth(formula=y~x, method=lm, color="red") +  
  theme(plot.title=element_text(hjust=0.5))  
gg5
```

pH vs. citric.acid



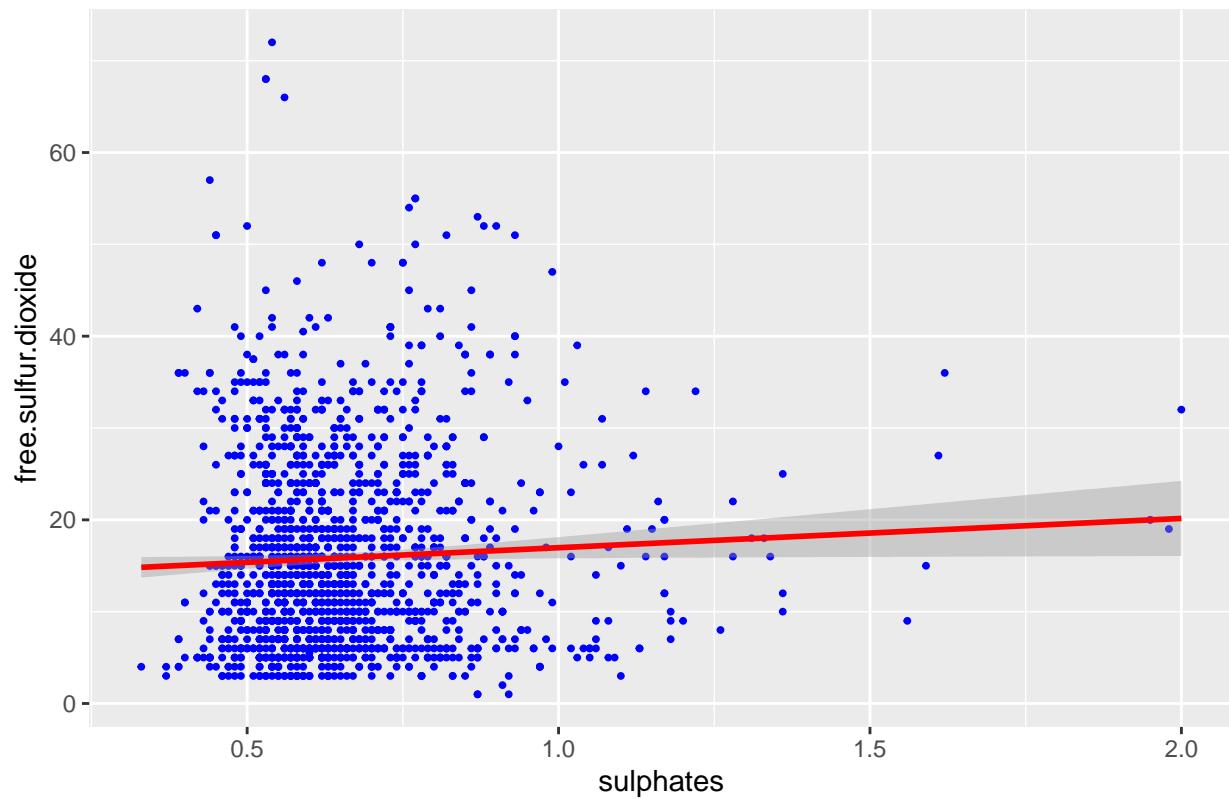
```
gg6 <- ggplot(red_df, aes(x=free.sulfur.dioxide, y=total.sulfur.dioxide)) +  
  geom_point(color="blue", size=0.7) +  
  labs(title="free.sulfur.dioxide vs. total.sulfur.dioxide") +  
  geom_smooth(formula=y~x, method=lm, color="red") +  
  theme(plot.title=element_text(hjust=0.5))  
gg6
```

free.sulfur.dioxide vs. total.sulfur.dioxide



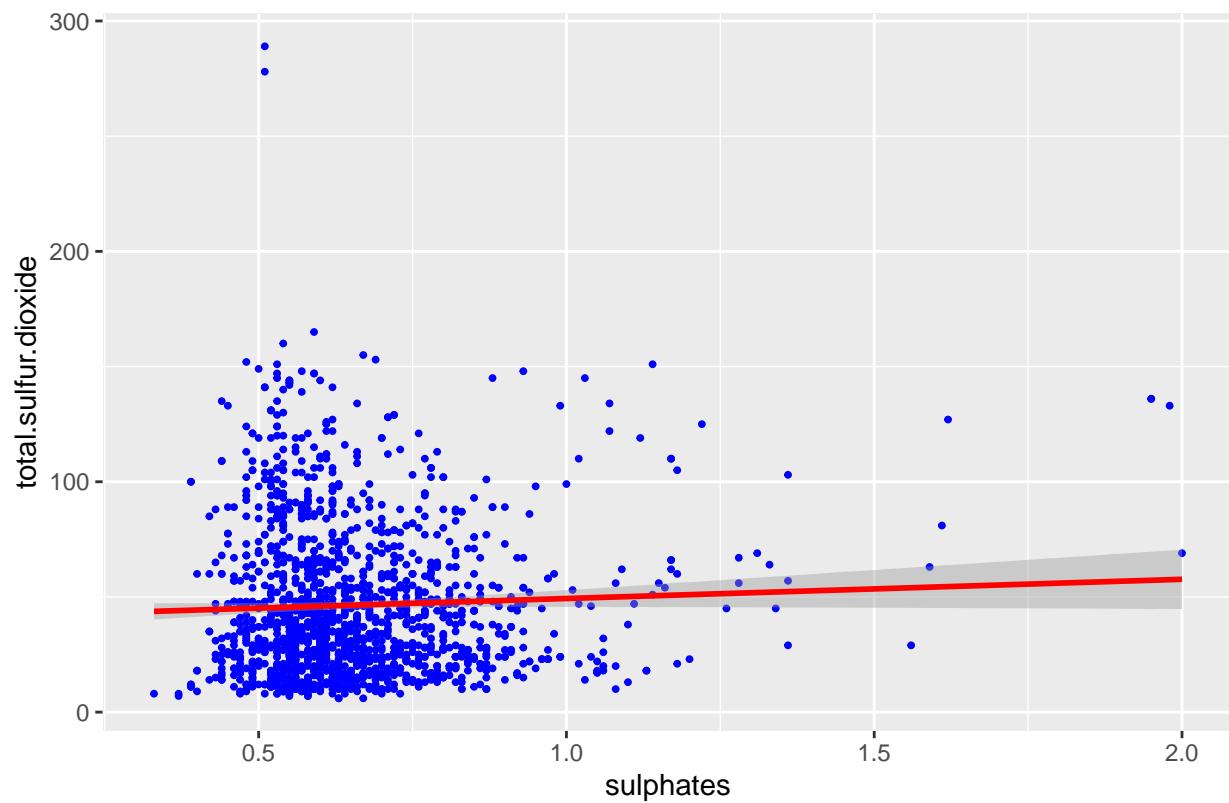
```
gg7 <- ggplot(red_df, aes(x=sulphates, y=free.sulfur.dioxide)) +  
  geom_point(color="blue", size=0.7) +  
  labs(title="free.sulfur.dioxide vs. sulphates") +  
  geom_smooth(formula=y~x, method=lm, color="red") +  
  theme(plot.title=element_text(hjust=0.5))  
gg7
```

free.sulfur.dioxide vs. sulphates

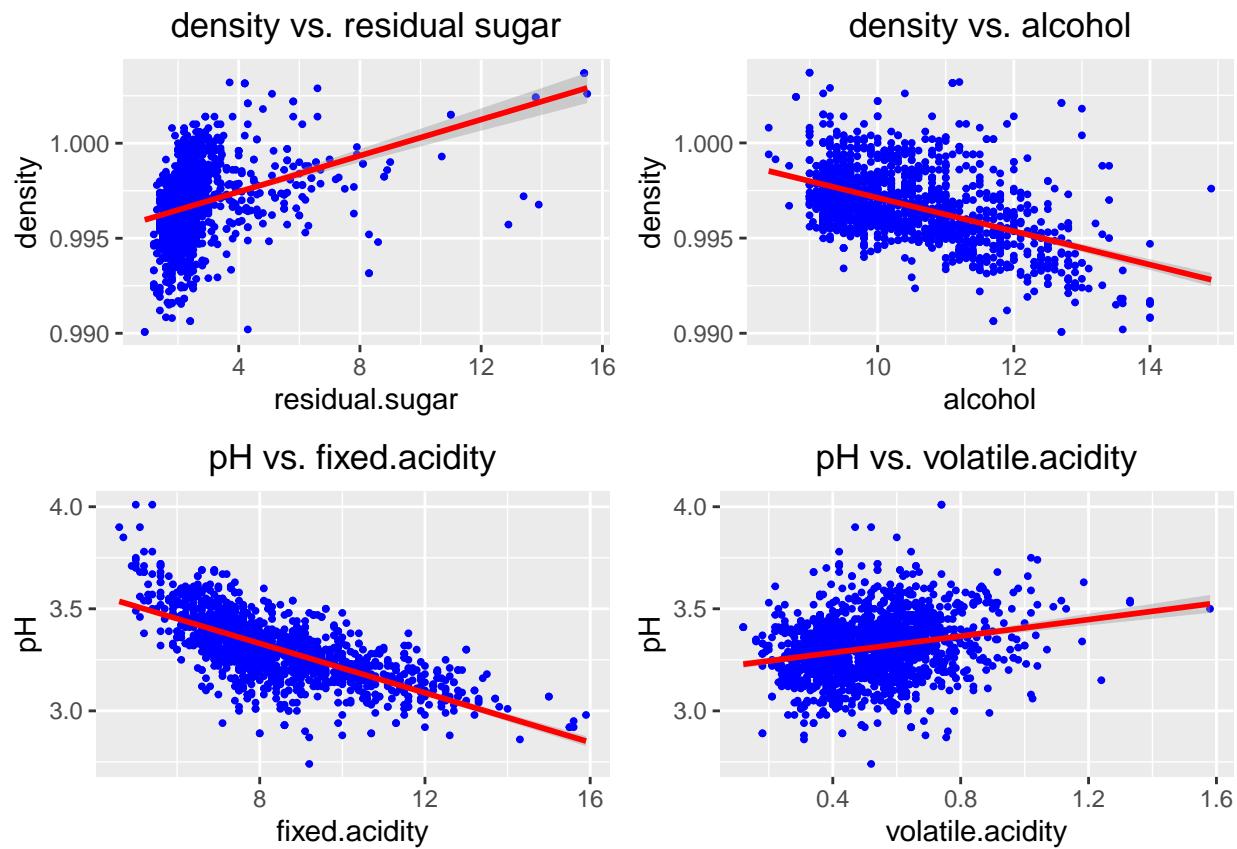


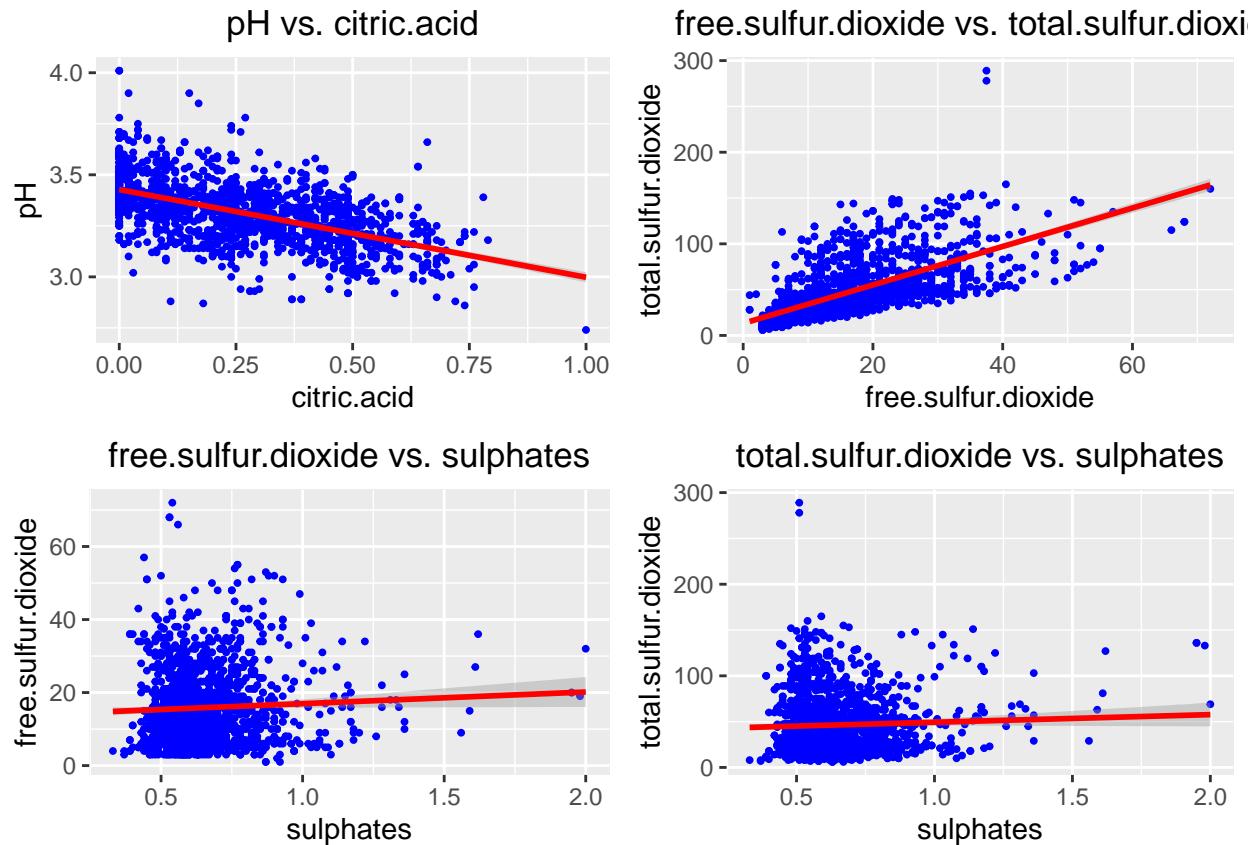
```
gg8 <- ggplot(red_df, aes(x=sulphates, y=total.sulfur.dioxide)) +  
  geom_point(color="blue", size=0.7) +  
  labs(title="total.sulfur.dioxide vs. sulphates") +  
  geom_smooth(formula=y~x, method=lm, color="red") +  
  theme(plot.title=element_text(hjust=0.5))  
gg8
```

total.sulfur.dioxide vs. sulphates



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





```
#Baseline Random Forest Model
redwineRF<-randomForest(quality ~ .,data=red_df,ntree=150)
```

```
redwineRF
```

```
##
## Call:
##   randomForest(formula = quality ~ ., data = red_df, ntree = 150)
##   Type of random forest: regression
##   Number of trees: 150
##   No. of variables tried at each split: 3
##
##   Mean of squared residuals: 0.3197216
##   % Var explained: 50.94
```

```
# Get importance
Importance      <- importance(redwineRF)
varImportance <- data.frame(Variablenames = row.names(Importance),
                           Importance = (Importance))
```

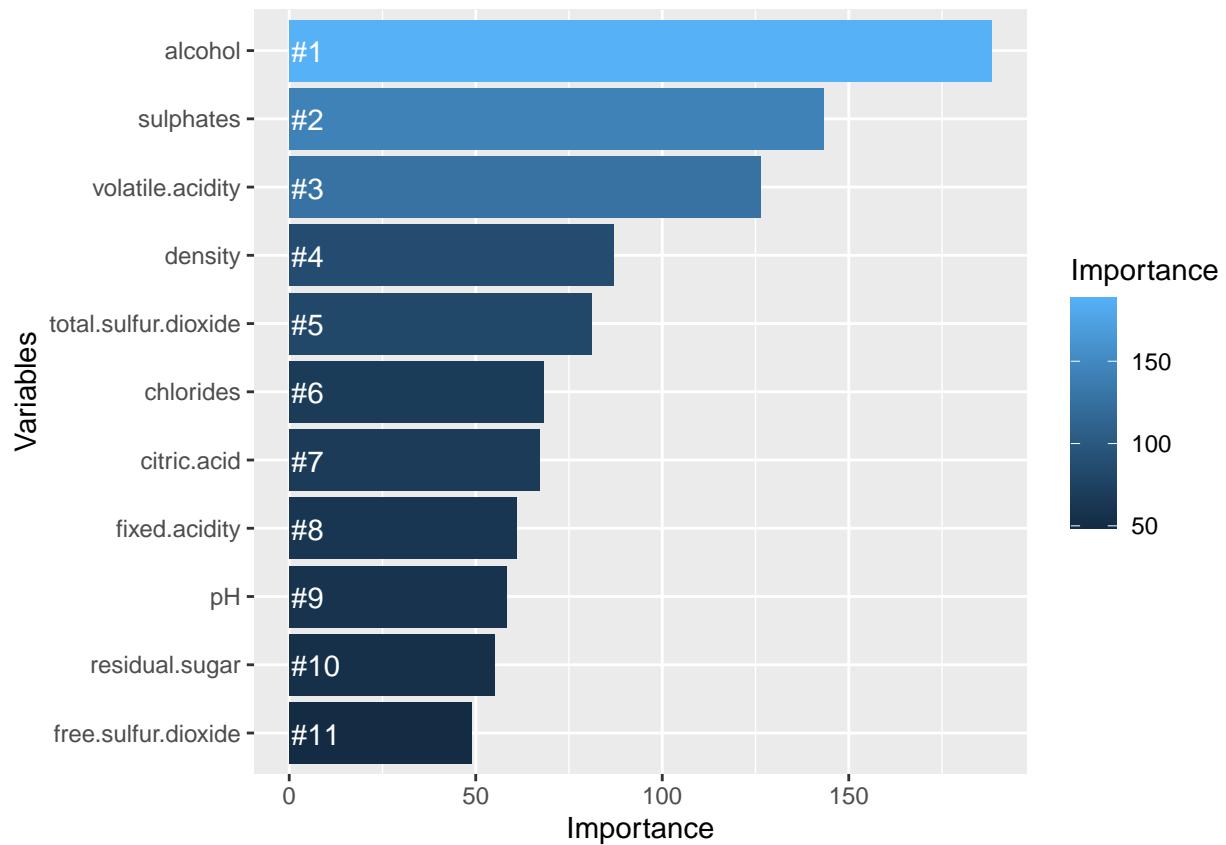
```
# Create a rank variable based on importance
rankImportance <- varImportance %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))
```

```
# 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(redwineRF)

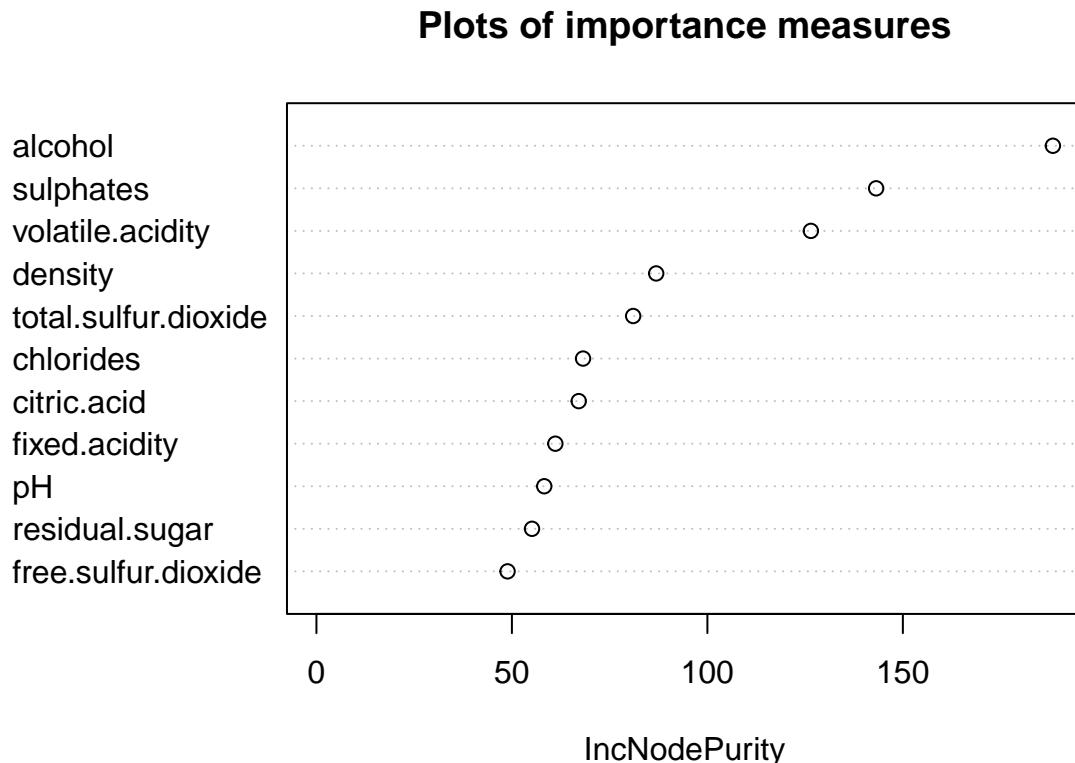
```

```

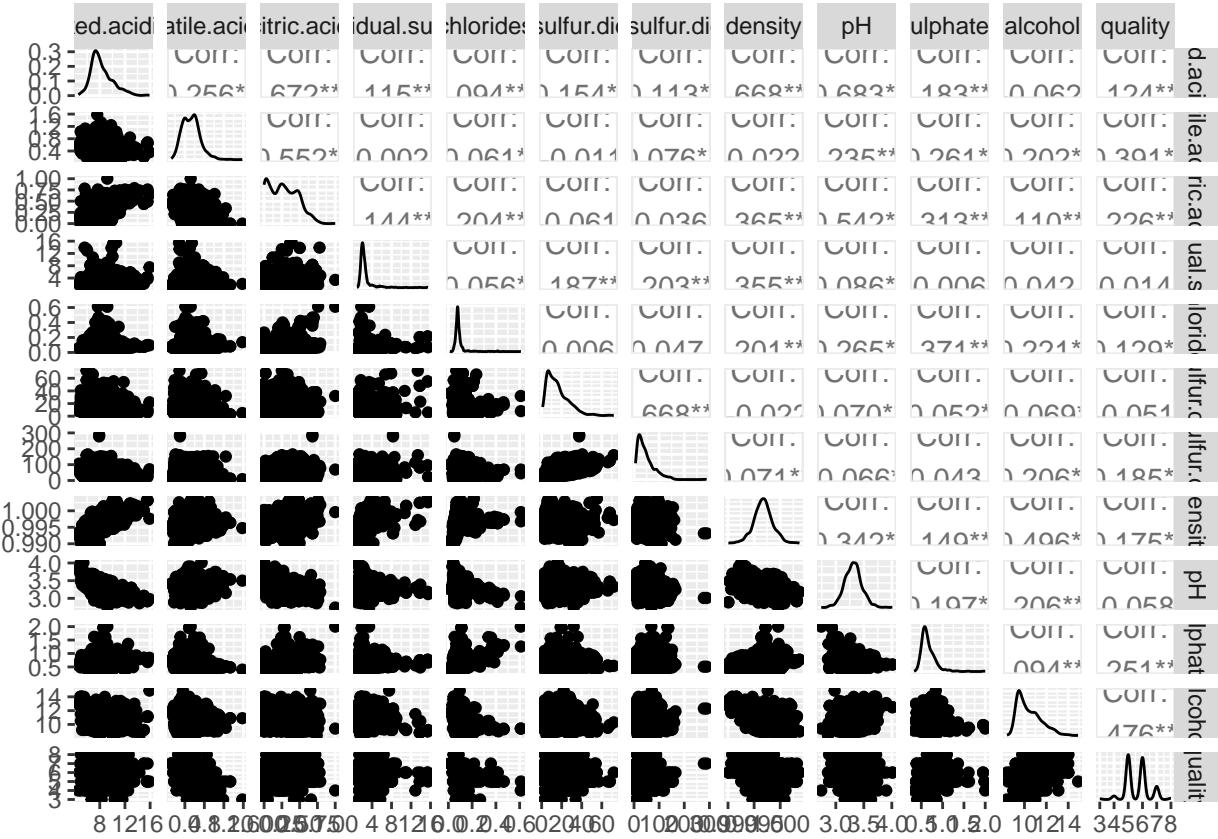
##           IncNodePurity
## fixed.acidity      61.11190
## volatile.acidity   126.46765
## citric.acid        67.09380
## residual.sugar     55.13986
## chlorides          68.19421
## free.sulfur.dioxide 48.89042
## total.sulfur.dioxide 81.01511
## density            86.90260
## pH                 58.26993
## sulphates          143.15981
## alcohol             188.37172

```

```
varImpPlot(redwineRF, main="Plots of importance measures")
```



```
ggpairs(red_df)
```



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

```
##
## Call:
## lm(formula = quality ~ alcohol + volatile.acidity + sulphates +
##     total.sulfur.dioxide, data = red_df)
##
## Residuals:
##      Min       1Q       Median       3Q      Max 
## -2.72716 -0.38486 -0.06503  0.44980  2.13257 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 2.8258128  0.2006892 14.081 < 2e-16 ***
## alcohol      0.2953105  0.0160331 18.419 < 2e-16 ***
## volatile.acidity -1.1985632  0.0966011 -12.407 < 2e-16 ***
## sulphates     0.7121396  0.1005146  7.085 2.08e-12 ***
## total.sulfur.dioxide -0.0022354  0.0005108 -4.376 1.28e-05 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.655 on 1594 degrees of freedom
## Multiple R-squared:  0.3438, Adjusted R-squared:  0.3421 
## F-statistic: 208.8 on 4 and 1594 DF,  p-value: < 2.2e-16
```

```

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

##
## Call:
## lm(formula = quality ~ alcohol + volatile.acidity + sulphates +
##     total.sulfur.dioxide, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.29129 -0.41439 -0.06258  0.44028  2.13996
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.0940601  0.2658198 11.640 < 2e-16 ***
## alcohol                  0.2826331  0.0232910 12.135 < 2e-16 ***
## volatile.acidity      -1.0201384  0.1302268 -7.834 1.52e-14 ***
## sulphates                 0.5282305  0.1206150  4.379 1.35e-05 ***
## total.sulfur.dioxide -0.0041631  0.0006662 -6.249 6.73e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6369 on 795 degrees of freedom
## Multiple R-squared:  0.3058, Adjusted R-squared:  0.3023
## F-statistic: 87.55 on 4 and 795 DF,  p-value: < 2.2e-16

#calculate MSE
mean(red_model$residuals^2)

## [1] 0.403121

#calculate RMSE
sqrt(mean(red_model$residuals^2))

## [1] 0.6349181

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

## [1] 0.5233042

#Lasso regression
x <- model.matrix(quality~., red_df)[,-1]
y <- red_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)      4.113819e+00
## fixed.acidity   0.000000e+00
## volatile.acidity -1.026243e+00
## citric.acid    0.000000e+00
## residual.sugar 8.124684e-05
## chlorides       -1.643697e+00
## free.sulfur.dioxide 1.928718e-03
## total.sulfur.dioxide -2.500755e-03
## density         0.000000e+00
## pH               -3.635424e-01
## sulphates        8.060690e-01
## alcohol          2.845865e-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)      3.1816088105
## fixed.acidity   0.0000000000
## volatile.acidity -1.0152991147
## citric.acid    0.0000000000
## residual.sugar 0.0000000000
## chlorides       -0.0882356401
## free.sulfur.dioxide 0.0000000000
## total.sulfur.dioxide -0.0008119338
## density         0.0000000000
## pH               0.0000000000
## sulphates        0.4577249926
## alcohol          0.2623533623

#Lasso regression model using above independent variables
red_lm1 <- lm(quality ~ fixed.acidity + volatile.acidity + chlorides + total.sulfur.dioxide +
              + sulphates + alcohol, data=red_df)
summary(red_lm1)

## 
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + chlorides +
##     total.sulfur.dioxide + sulphates + alcohol, data = red_df)
## 
## Residuals:
##      Min      1Q      Median      3Q      Max 
## -2.70812 -0.37181 -0.06238  0.45933  1.99472 
## 
## Coefficients:
```

```

##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.7365412  0.2325021 11.770 < 2e-16 ***
## fixed.acidity         0.0236576  0.0099187  2.385  0.0172 *
## volatile.acidity      -1.0856214  0.0996323 -10.896 < 2e-16 ***
## chlorides              -1.7376885  0.3913566 -4.440 9.61e-06 ***
## total.sulfur.dioxide -0.0021460  0.0005121 -4.191 2.93e-05 ***
## sulphates              0.8846921  0.1108310  7.982 2.72e-15 ***
## alcohol                 0.2825603  0.0166180 17.003 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6504 on 1592 degrees of freedom
## Multiple R-squared:  0.3538, Adjusted R-squared:  0.3514
## F-statistic: 145.3 on 6 and 1592 DF, p-value: < 2.2e-16

# Define training control
set.seed(555)
train <- red_df[1:800, ]
test <- red_df[801:nrow(red_df), ]
# Train the model
red_model1 <- lm(quality ~ fixed.acidity + volatile.acidity + chlorides + total.sulfur.dioxide
                  + sulphates + alcohol, data=train)
summary(red_model1)

##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + chlorides +
##     total.sulfur.dioxide + sulphates + alcohol, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.23168 -0.39914 -0.07055  0.45416  2.04570
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.0774808  0.2912846 10.565 < 2e-16 ***
## fixed.acidity          0.0177703  0.0127942  1.389  0.16524
## volatile.acidity       -0.9332460  0.1344362 -6.942 8.05e-12 ***
## chlorides              -1.3764407  0.4649166 -2.961  0.00316 **
## total.sulfur.dioxide -0.0040524  0.0006758 -5.997 3.06e-09 ***
## sulphates              0.6873536  0.1345356  5.109 4.06e-07 ***
## alcohol                 0.2655413  0.0237330 11.189 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6333 on 793 degrees of freedom
## Multiple R-squared:  0.3153, Adjusted R-squared:  0.3101
## F-statistic: 60.86 on 6 and 793 DF, p-value: < 2.2e-16

#calculate MSE
mean(red_model1$residuals^2)

## [1] 0.3976104

```

```

#calculate RMSE
sqrt(mean(red_model1$residuals^2))

## [1] 0.6305636

#calculate MAE
predValues <- predict(red_model1,test)
#MAE for the model
mean(abs(test$quality -predValues))

## [1] 0.5179756

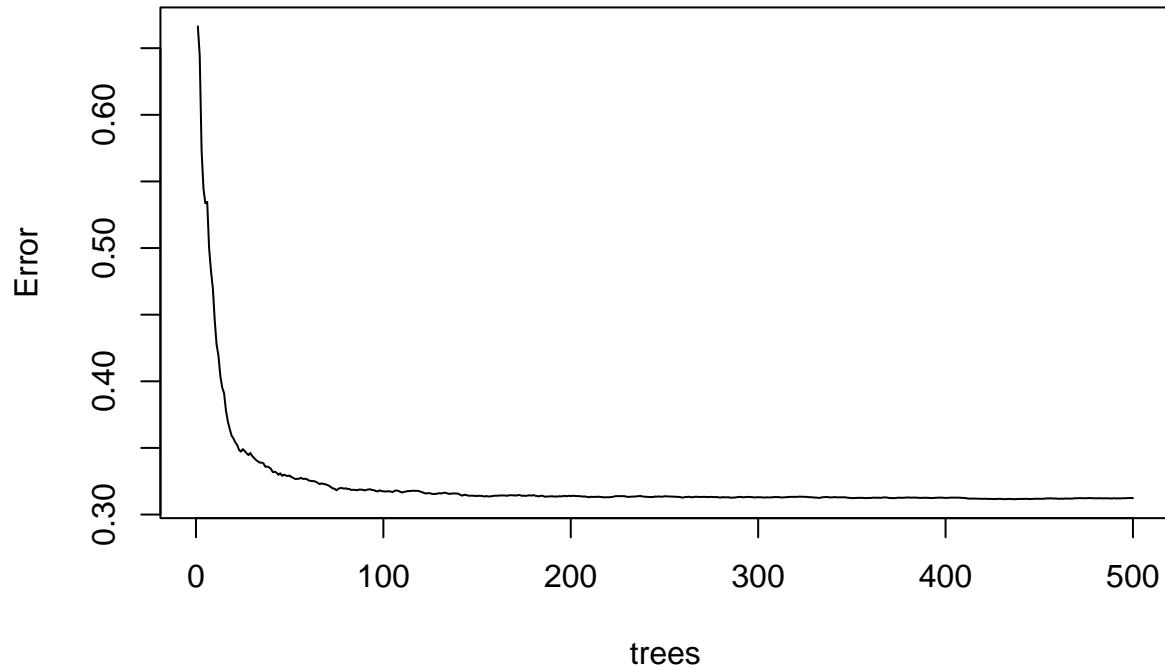
#Random forest
set.seed(555)
train <- red_df[1:800, ]
test <- red_df[801:nrow(red_df), ]
red_model3 <- randomForest(quality ~ ., train, mtry = 3,
                           importance = TRUE, na.action = na.omit)
print(red_model3)

##
## 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: 0.3123345
##   % Var explained: 46.21

#Plot the error vs the number of trees graph
plot(red_model3)

```

red_model3

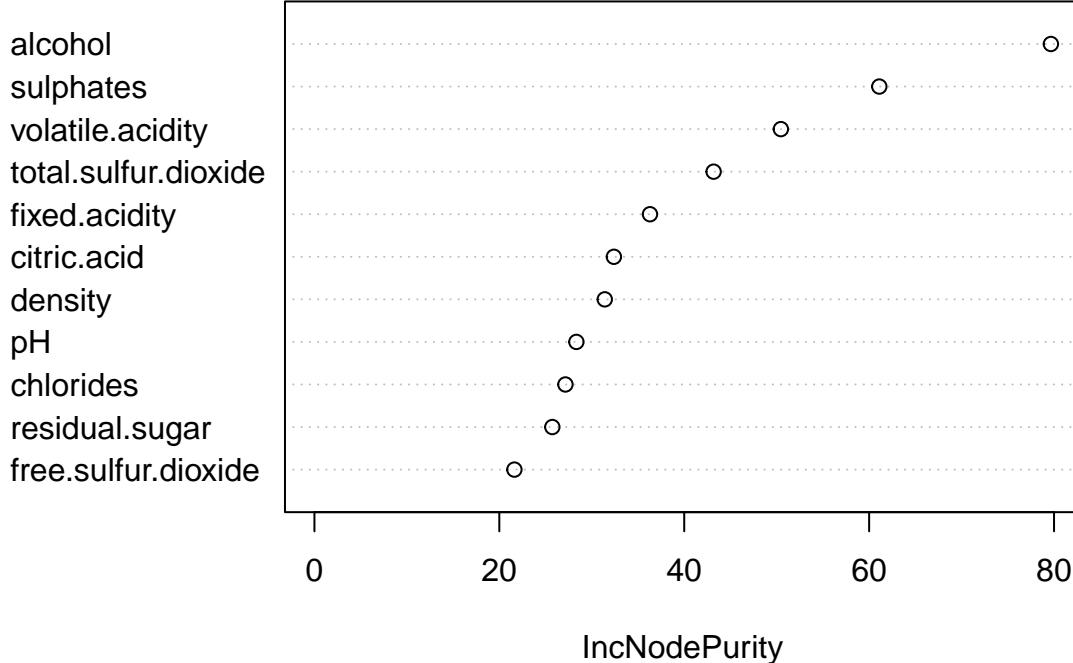


```
importance(red_model3)
```

```
##                                     %IncMSE IncNodePurity
## fixed.acidity             21.90628    36.29757
## volatile.acidity          28.48120    50.45388
## citric.acid              21.39634    32.39123
## residual.sugar            17.38764    25.73327
## chlorides                  19.30311    27.15030
## free.sulfur.dioxide       17.97617    21.62821
## total.sulfur.dioxide      34.88990    43.17906
## density                   21.37384    31.40464
## pH                         17.57445    28.33563
## sulphates                 36.74733    61.10011
## alcohol                   43.73199    79.66182
```

```
varImpPlot(red_model3,type=2)
```

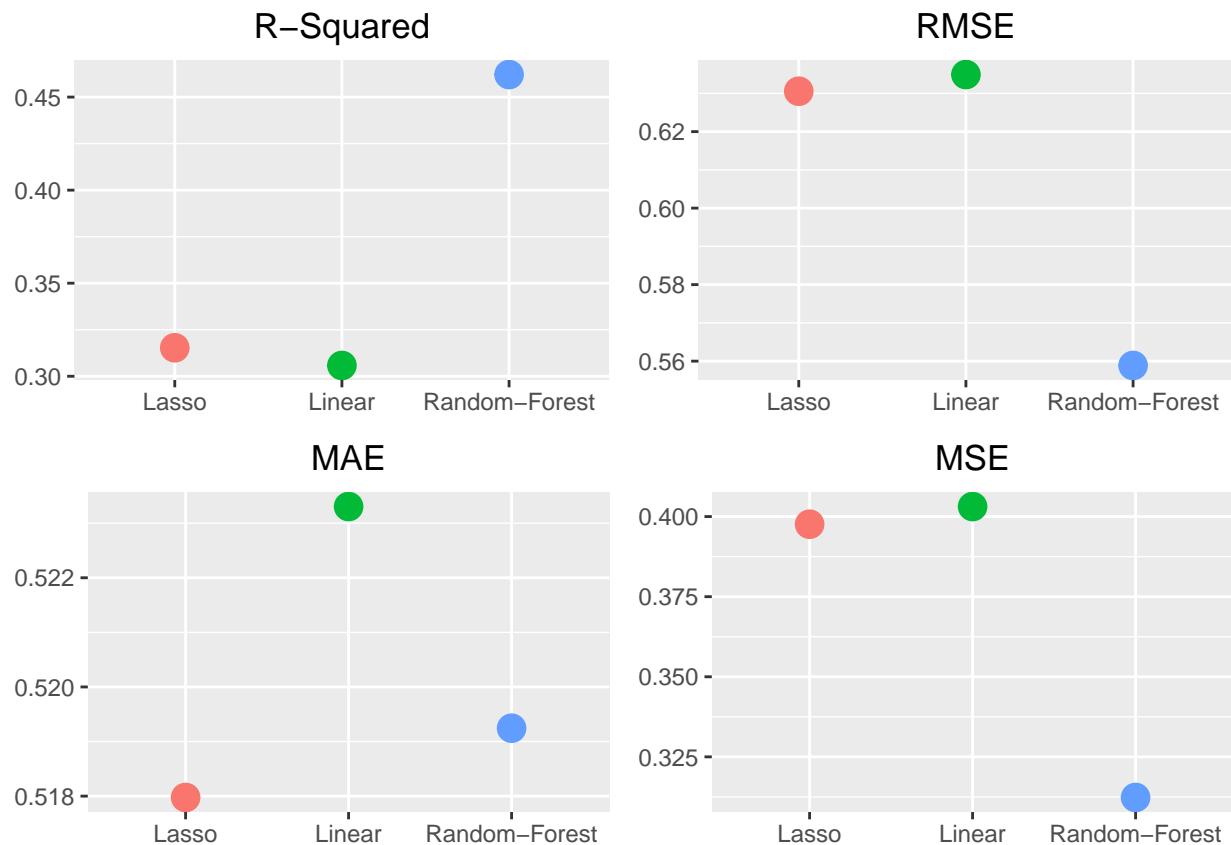
red_model3



```
# MSE for the model
red_model3$mse[length(red_model3$mse)]  
  
## [1] 0.3123345  
  
# RMSE for the model
sqrt(red_model3$mse[length(red_model3$mse)])  
  
## [1] 0.558869  
  
predValues <- predict(red_model3,test)
#MAE for the model
mean(abs(test$quality -predValues))  
  
## [1] 0.5192442  
  
#Comparing models  
  
Model <- c("Linear", "Lasso", "Random-Forest")
R_squared <- c(0.3058, 0.3153, 0.4621)
RMSE <- c(0.6349181, 0.6305636, 0.558869)
MAE <- c(0.5233042, 0.5179756, 0.5192442)
MSE <- c(0.403121, 0.3976104, 0.3123345)
```

```
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")
p2 <- ggplot(ml, aes(Model, R_squared)) + geom_point(aes(colour = factor(Model), size = 4)) + labs(title="R-Squared")
p3 <- ggplot(ml, aes(Model, MAE)) + geom_point(aes(colour = factor(Model), size = 4)) + labs(title="MAE")
p4 <- ggplot(ml, aes(Model, MSE)) + geom_point(aes(colour = factor(Model), size = 4)) + labs(title="MSE")
ggarrange(p2,p1,p3,p4, nrow=2, ncol=2)
```



```
#clustering to find relation between predictors
```

```
set.seed(1941)
red_k <- kmeans(red_df, 2, nstart = 25)
names(red_k)
```

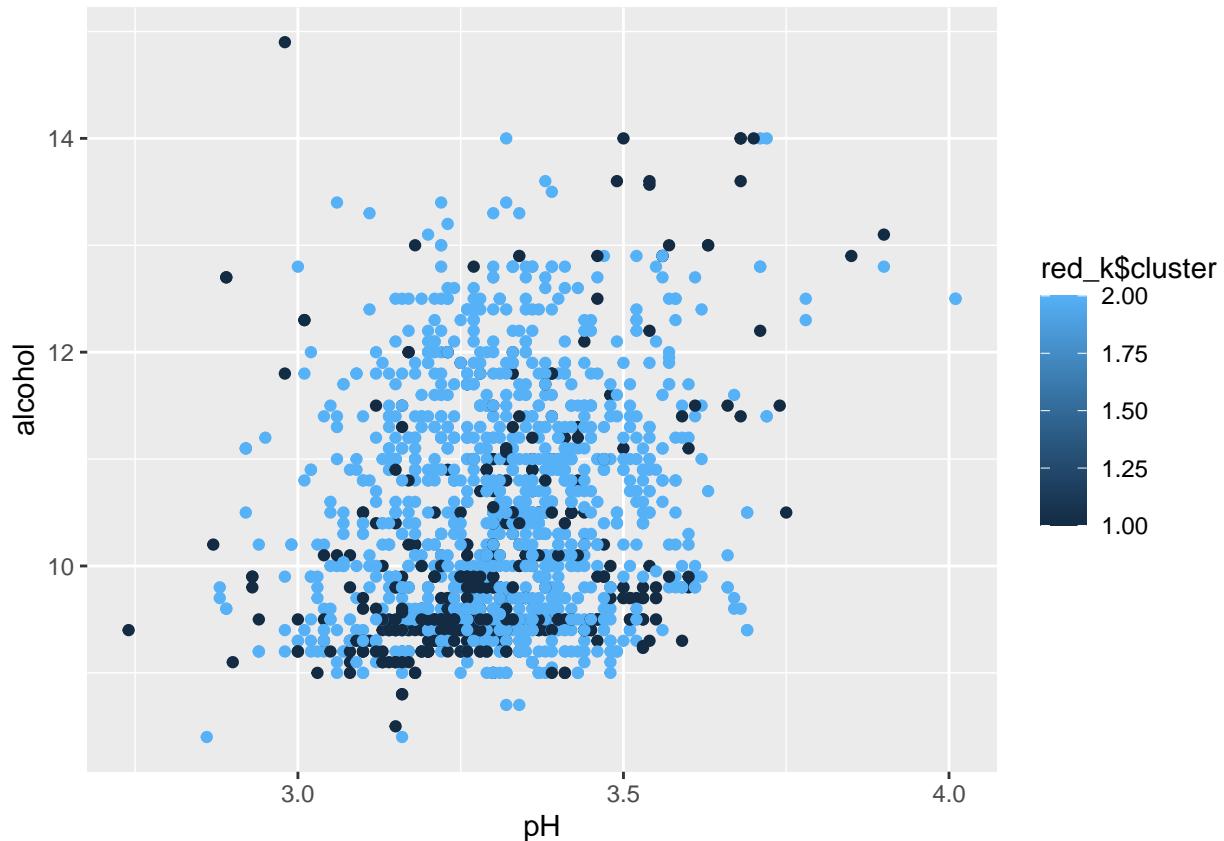
```
## [1] "cluster"      "centers"       "totss"        "withinss"      "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
red_k$centers
```

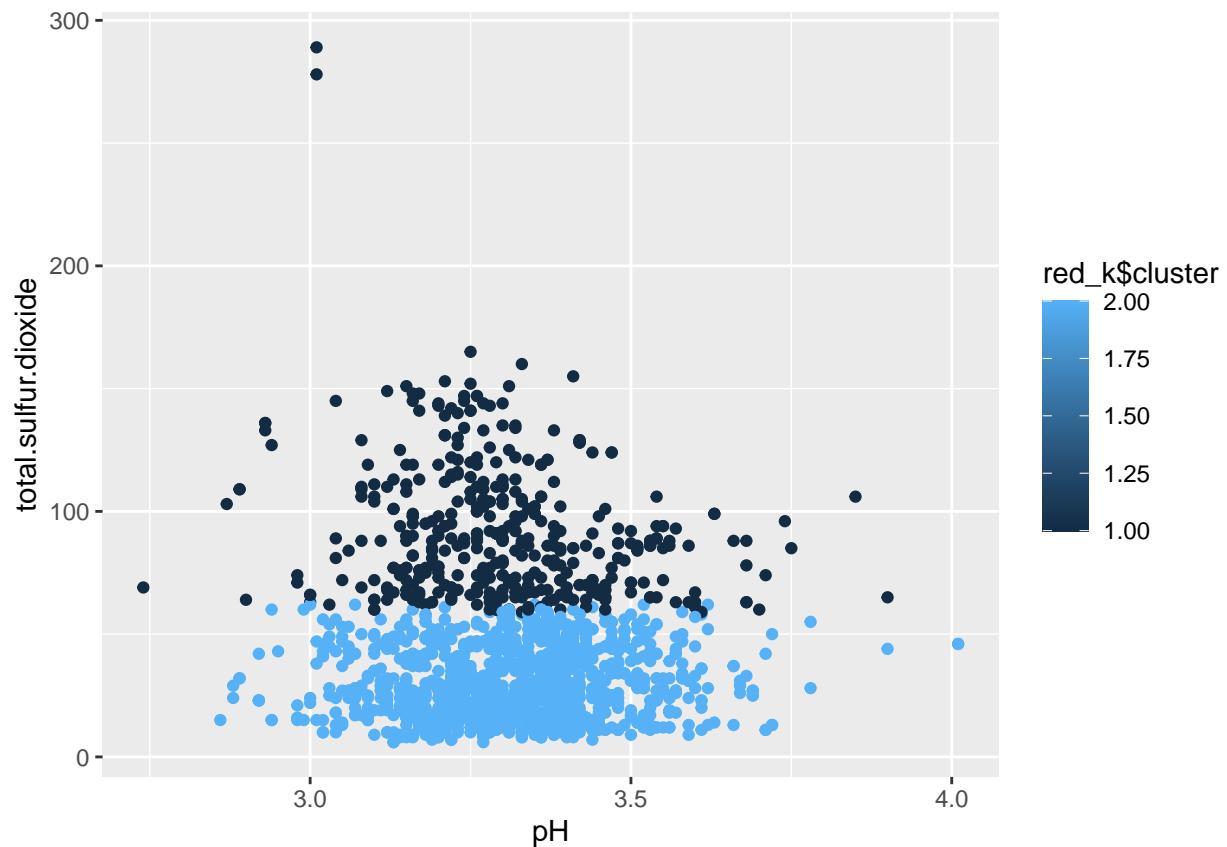
```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1     8.025952          0.5516429   0.2834286     2.944524 0.09313810
## 2     8.424258          0.5193342   0.2665394     2.394275 0.08544614
##   free.sulfur.dioxide total.sulfur.dioxide density      pH sulphates
## 1      25.70833           91.72857 0.9969427 3.298738 0.6626905
```

```
## 2           12.37193      30.34436 0.9966768 3.315522 0.6565310
##   alcohol   quality
## 1 10.09389 5.388095
## 2 10.54022 5.724343
```

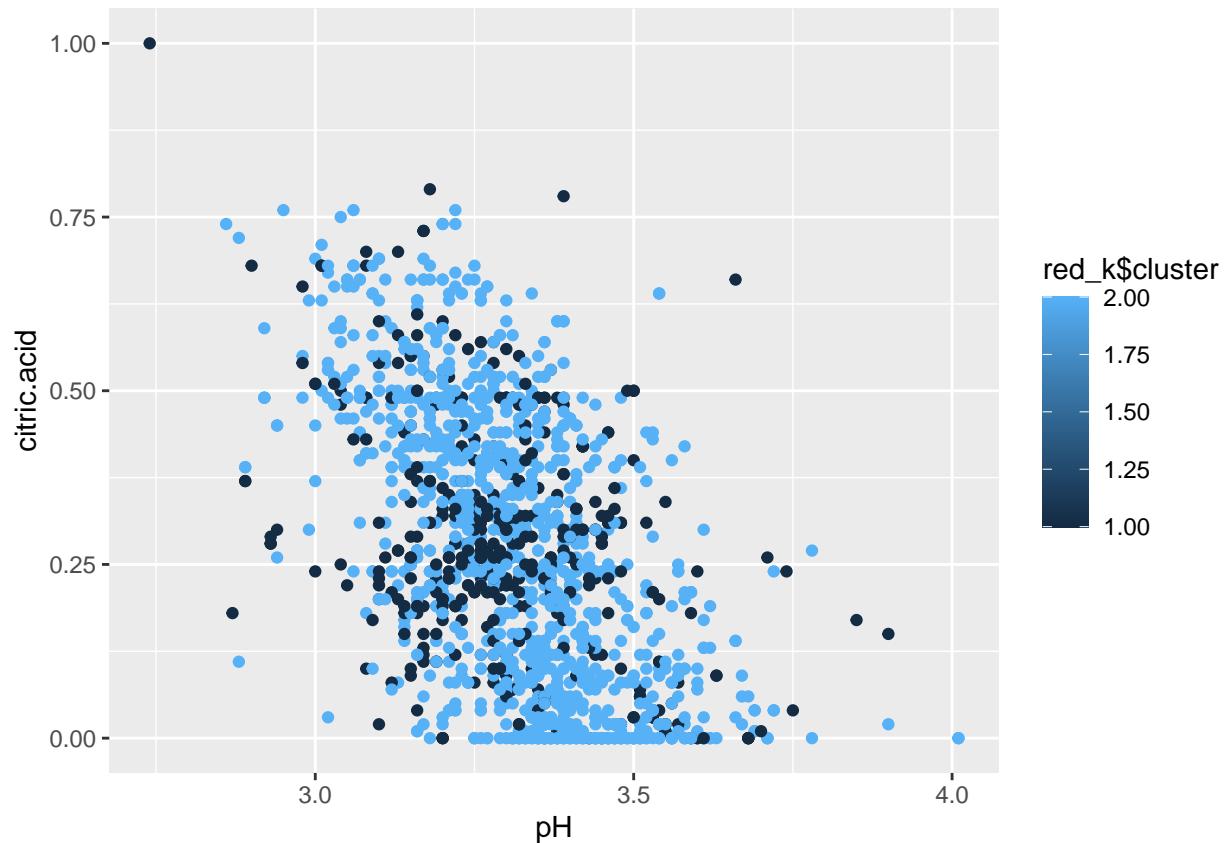
```
c1 <- ggplot(red_df)+geom_point(aes(x=pH,y=alcohol ,color=red_k$cluster))
c1
```



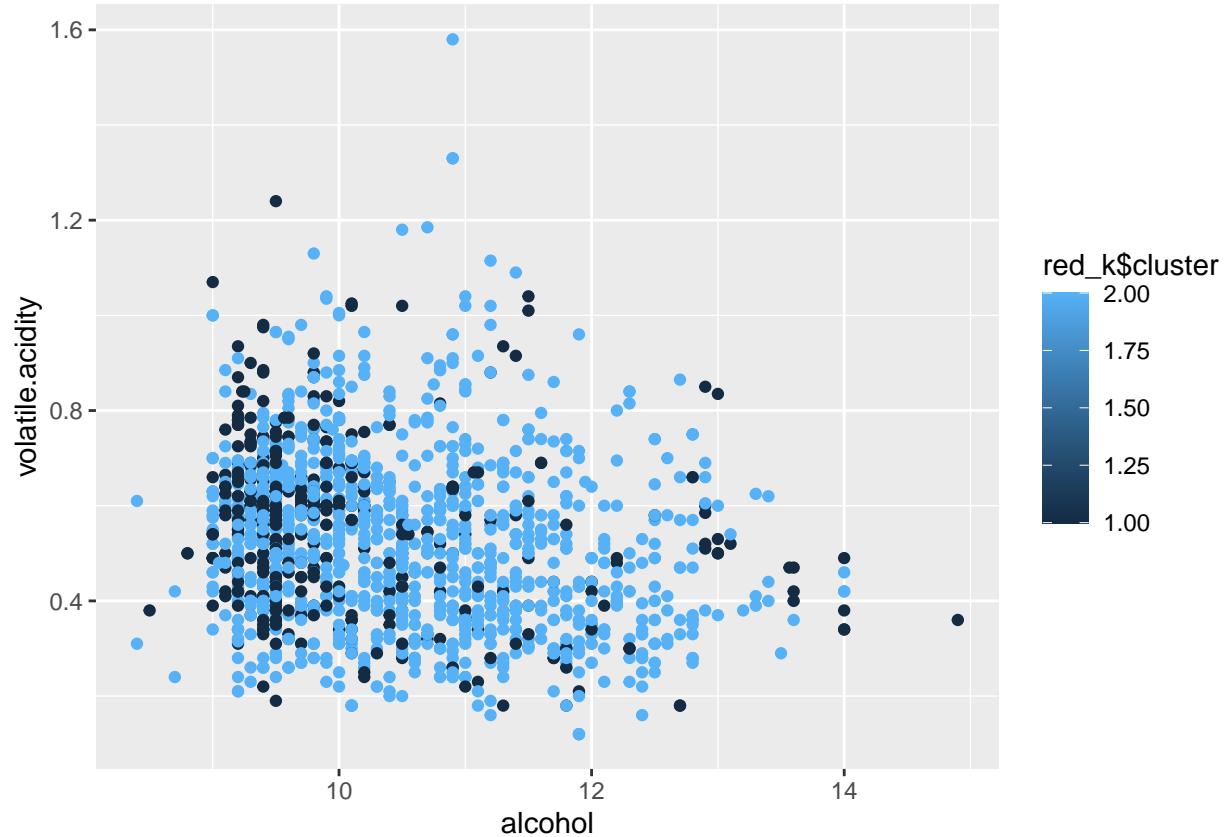
```
c2 <- ggplot(red_df)+geom_point(aes(x=pH,y=total.sulfur.dioxide ,color=red_k$cluster))
c2
```



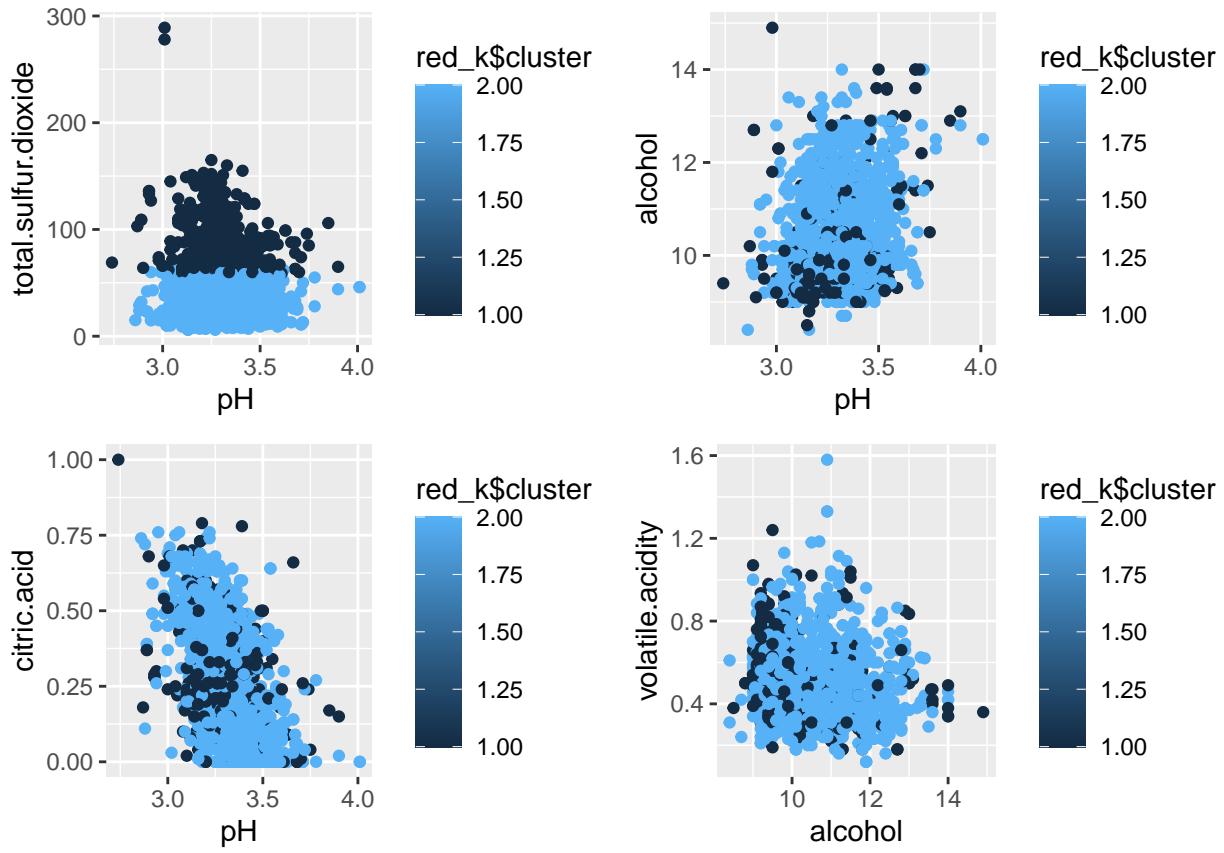
```
c3 <- ggplot(red_df)+geom_point(aes(x=pH,y=citric.acid ,color=red_k$cluster))  
c3
```



```
c4 <- ggplot(red_df)+geom_point(aes(x=alcohol,y=volatile.acidity ,color=red_k$cluster))  
c4
```



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



```

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

## K-means clustering with 3 clusters of sizes 519, 839, 241
##
## Cluster means:
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1     8.258189      0.5210019   0.2688632     2.488054 0.09184586
## 2     8.481764      0.5192431   0.2709416     2.370977 0.08448987
## 3     7.887552      0.5723651   0.2756432     3.232365 0.08839834
##   free.sulfur.dioxide total.sulfur.dioxide density      pH sulphates
## 1           21.939306                  55.69750 0.9968552 3.322428 0.6767630
## 2            9.171633                  22.93802 0.9966256 3.310286 0.6483909
## 3           26.151452                 108.50622 0.9969347 3.289627 0.6520332
##   alcohol    quality
## 1 10.34399 5.603083
## 2 10.59078 5.750894
## 3 10.00892 5.307054
##
## Clustering vector:
## [1] 2 1 1 1 2 2 1 2 2 3 1 3 1 2 3 3 1 2 1 1 2 1 1 2 2 2 2 1 1 3 3 1 2 2
## [38] 2 2 3 3 1 2 2 2 1 3 2 2 3 2 2 2 3 1 1 2 3 1 1 1 3 2 3 2 3 3 2 3 3 2 2 2 2 1 2 2 2 1 2 1 1 1 3 3 1 1 2 1 3 3 1 1 1 2 1 3 3 1
## [75] 3 1 1 2 1 3 2 1 1 1 1 3 2 3 2 3 3 2 3 3 2 2 2 2 1 2 2 1 2 1 1 1 3 3 1 1 1 2 1 3 3 1 1 1 2 1 3 3 1
## [112] 3 3 1 1 1 2 2 1 3 3 1 2 2 3 3 2 2 2 3 3 3 2 2 1 1 1 3 3 1 1 1 2 1 3 3 1 1 1 2 1 3 3 1

```

```

## [149] 2 1 2 1 3 3 3 3 3 2 3 2 2 2 3 3 3 3 1 1 1 2 2 2 1 2 1 2 2 2 1 1 3 1 1 1
## [186] 3 1 2 3 3 3 1 3 2 2 3 1 2 3 2 2 3 1 2 2 2 2 3 3 2 2 1 2 1 2 3 1 2 1 3 1 1
## [223] 2 2 1 1 1 2 1 2 3 1 1 2 2 2 2 2 2 1 2 3 2 2 2 1 1 2 2 2 2 3 2 3 2 1 1
## [260] 2 1 1 2 1 2 2 1 2 2 2 1 2 1 2 3 1 2 2 2 1 2 2 1 1 1 1 1 1 3 1 2 1 2 1 2
## [297] 1 2 2 2 1 2 2 2 3 2 1 2 2 2 2 3 3 1 1 3 1 1 1 1 3 2 1 1 1 2 2 2 2 2 2 3
## [334] 1 2 2 2 1 3 2 2 2 2 2 1 2 2 1 2 2 2 1 3 2 1 2 2 2 1 1 2 2 2 2 2 2 1 2
## [371] 1 2 1 1 1 2 1 2 2 1 2 1 2 2 1 2 1 1 1 3 1 2 3 2 2 3 1 1 1 3 1 1 2 2 2 2
## [408] 2 2 1 3 1 1 2 3 3 2 3 2 1 1 1 1 2 1 1 1 2 2 2 2 1 2 2 1 2 1 2 1 1 2 2 2 2
## [445] 2 2 1 2 2 2 2 2 2 1 2 2 1 2 1 2 2 2 3 2 2 1 2 2 1 1 1 2 2 2 2 2 2 2 2 2
## [482] 2 2 2 2 2 2 2 1 2 1 2 2 1 3 2 2 3 2 1 2 1 1 2 2 2 2 2 1 1 1 1 2 2 3 2 2
## [519] 2 3 2 1 3 3 3 1 3 3 1 2 2 1 1 2 2 2 2 2 2 1 2 2 1 2 3 2 2 1 2 2 1 1 3 2
## [556] 2 2 2 2 1 2 3 3 1 1 2 2 2 2 1 2 1 2 1 1 2 3 3 1 2 2 2 2 3 2 1 3 1 2 1 3
## [593] 1 1 1 3 2 2 2 2 2 2 2 1 2 2 1 1 2 1 2 2 2 3 1 1 2 2 2 3 3 2 1 1 1 2 2 2
## [630] 3 2 2 1 3 3 2 3 3 2 1 1 1 1 1 2 2 2 2 3 2 3 1 2 2 1 2 1 2 2 2 1 2 2 1 1
## [667] 2 2 2 2 1 2 3 2 2 2 2 2 3 1 1 2 1 2 3 2 2 2 2 2 3 1 3 3 1 2 2 1 2 3 2 2
## [704] 3 2 1 2 2 2 2 3 3 2 1 1 2 1 1 2 1 2 3 1 3 2 2 1 2 2 1 2 2 2 1 2 2 2 1 3 1
## [741] 2 3 2 3 3 2 1 1 2 2 1 1 1 1 2 1 2 2 2 3 3 2 2 2 1 1 1 3 3 2 3 3 3 2 1 2 1
## [778] 2 2 3 1 2 3 2 3 1 1 1 1 3 1 3 3 2 2 1 1 1 2 2 3 2 3 1 2 2 2 2 2 2 2 2 1 2
## [815] 1 1 2 2 1 1 2 1 2 2 2 2 2 1 2 2 2 1 1 2 1 3 3 2 2 2 2 1 3 1 2 2 2 2 2 2
## [852] 2 3 1 1 2 1 1 2 2 3 3 2 3 3 3 2 2 2 1 1 1 1 2 2 2 2 1 1 3 2 1 2 3 1 2 2 1
## [889] 1 3 1 3 2 3 3 1 1 1 1 2 2 2 2 2 2 3 1 2 1 2 2 2 2 2 2 1 1 1 1 2 1 1 1 2
## [926] 1 1 1 2 2 2 2 1 2 2 1 1 2 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2
## [963] 2 2 2 1 2 3 1 2 2 2 2 2 2 1 1 3 2 2 2 2 1 2 2 2 2 1 2 2 2 1 1 1 3 1 2 2 2
## [1000] 2 2 1 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 3 3 2 2 2 2 2 2 1 2 1 1 2 2 2 2 2 1 2
## [1037] 2 1 1 1 2 2 1 1 1 1 1 2 2 1 1 2 2 1 1 2 3 1 2 2 2 2 2 2 2 2 2 1 2 3 1
## [1074] 2 3 1 2 2 2 3 2 3 1 1 1 1 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1
## [1111] 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 3 1 2 3 2 2 2 1 2 2 3 3 1 1 2 2 1 1 2
## [1148] 2 2 2 2 1 2 2 1 2 3 3 1 2 2 2 2 2 2 2 1 1 1 2 2 1 2 1 1 1 1 2 1 1 1 2 1 1
## [1185] 3 2 2 2 3 2 2 2 2 2 1 1 3 2 1 3 2 2 2 1 1 1 1 1 2 2 1 2 2 2 2 1 1 2 1 1
## [1222] 1 3 2 2 1 1 2 3 1 1 3 1 2 2 3 2 2 2 2 1 1 1 3 3 2 1 2 2 1 1 1 2 2 2 1 1 1
## [1259] 2 2 1 2 1 2 1 2 2 2 1 3 1 1 1 1 2 1 2 2 1 2 1 1 2 1 2 1 2 2 3 3 2 1 2 2 1
## [1296] 1 1 2 2 2 2 1 2 1 1 3 3 2 3 1 3 2 1 1 1 3 1 2 3 1 3 1 2 1 2 2 2 2 2 3 3 1
## [1333] 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 2 2 1 3 2 2 1 2 1 2 1 1 3 1
## [1370] 2 1 2 1 3 2 3 2 1 2 2 2 2 3 3 3 3 2 2 1 3 2 2 2 1 1 2 2 3 2 2 3 3 2 2 1 2
## [1407] 2 1 1 1 2 2 2 3 2 2 2 2 2 3 2 1 1 2 2 2 1 2 1 1 1 2 2 3 3 3 2 2 3 2 3 2
## [1444] 2 3 3 2 1 1 2 2 2 1 3 2 2 3 3 2 2 1 2 2 2 1 1 1 2 1 1 2 2 1 1 3 3 3 3 2 2
## [1481] 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 3 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2 2 1 1 1 1
## [1518] 1 2 2 1 2 1 1 1 2 2 2 1 1 2 2 2 3 2 2 1 2 2 1 2 1 1 2 2 2 2 2 1 2 2 2 1 2
## [1555] 2 2 2 2 3 3 3 3 2 2 2 1 1 2 1 2 2 2 3 1 3 2 2 2 2 2 2 2 2 3 2 1 1 2 3 1 2
## [1592] 2 1 1 1 1 1 1 1 1

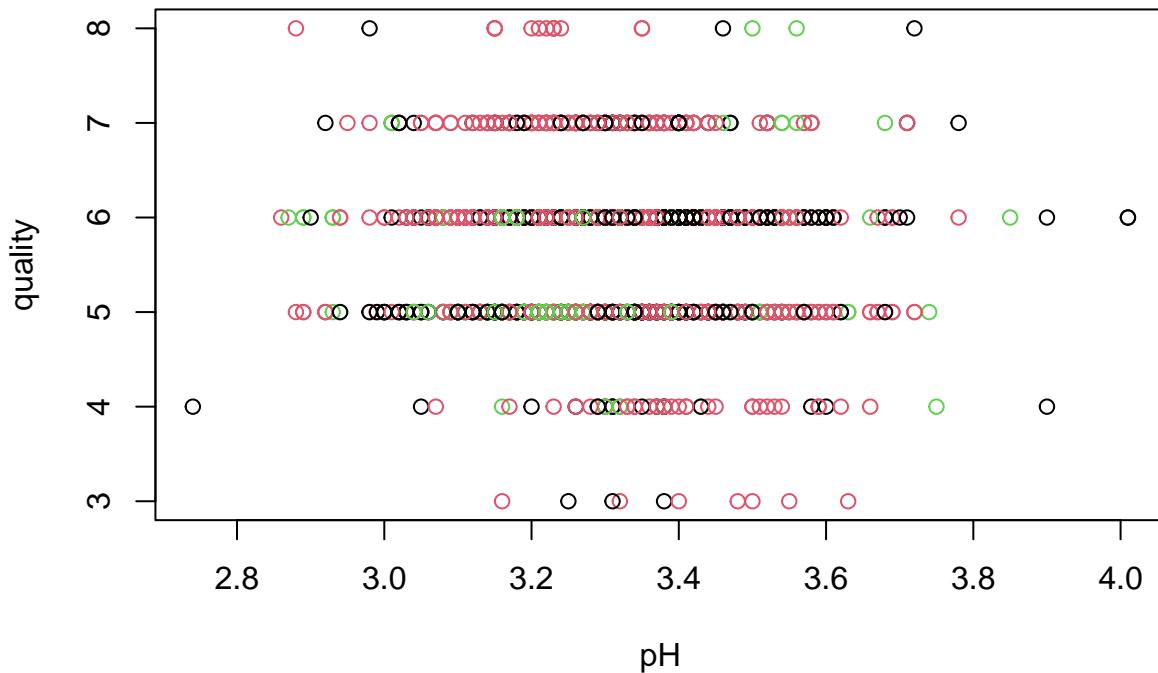
##
## Within cluster sum of squares by cluster:
## [1] 113849.5 88801.2 193638.5
##   (between_SS / total_SS = 79.3 %)

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

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

```

Red 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=red_df) #~target variable,predicting
tree1

## Conditional inference tree with 10 terminal nodes
##
## Response: quality
## Inputs: alcohol, volatile.acidity
## Number of observations: 1599
##
## 1) alcohol <= 10.5; criterion = 1, statistic = 362.322
##   2) volatile.acidity <= 0.33; criterion = 1, statistic = 88.693
##     3) alcohol <= 9.7; criterion = 0.963, statistic = 5.545
##       4)* weights = 37
##     3) alcohol > 9.7
##       5)* weights = 48
##   2) volatile.acidity > 0.33
##     6) volatile.acidity <= 0.65; criterion = 1, statistic = 43.251
##       7) alcohol <= 9.8; criterion = 1, statistic = 31.696
##         8)* weights = 429
##       7) alcohol > 9.8
##         9)* weights = 228
##     6) volatile.acidity > 0.65
##       10)* weights = 241
## 1) alcohol > 10.5
##   11) volatile.acidity <= 0.865; criterion = 1, statistic = 105.187
##     12) volatile.acidity <= 0.44; criterion = 1, statistic = 44.243
##       13) alcohol <= 11.5; criterion = 0.999, statistic = 12.315
##         14)* weights = 156
##       13) alcohol > 11.5
##         15)* weights = 145
##     12) volatile.acidity > 0.44
##       16) alcohol <= 11.4; criterion = 1, statistic = 22.228
##         17)* weights = 167
##       16) alcohol > 11.4
##         18)* weights = 121
##     11) volatile.acidity > 0.865
##       19)* weights = 27

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

## n= 1599
##
## node), split, n, loss, yval, (yprob)
##      * denotes terminal node
##
## 1) root 1599 918 5 (0.0063 0.033 0.43 0.4 0.12 0.011)
##   2) alcohol< 10.25 842 323 5 (0.0083 0.034 0.62 0.31 0.03 0.0024)
##     4) volatile.acidity>=0.3175 788 280 5 (0.0089 0.036 0.64 0.29 0.022 0.0013) *
##     5) volatile.acidity< 0.3175 54 21 6 (0 0.019 0.2 0.61 0.15 0.019) *
##   3) alcohol>=10.25 757 379 6 (0.004 0.032 0.21 0.5 0.23 0.021) *

```

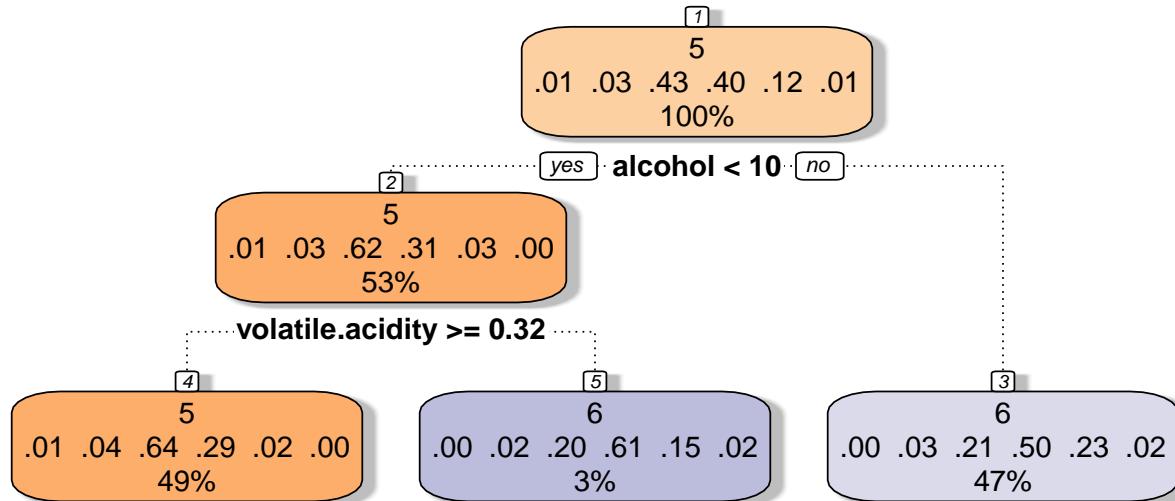
```

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="Red wine Classification")

```



Red wine Classification