Red Wine Data Exploration

Anuj Nimkar

Project Overview - This is an exploration of 1599 samples of red wine. Our main purpose of carrying out the exploration of the given data set is that, we need to figure out which factors among the given set of factors are the most influential ones in deciding the quality of the wine. Such kind of an analysis could help a decision maker (Management of a Wine selling company) to take a decision on how much should they invest, in which wine ingredients. Also, it can give them an overview about the quality of their current product.

Analysis

Set the working directory and load the data

```
setwd('F:/Anuj/Study & Work/Data Analytics/EDA using R/Final Project')
redWineData <- read.csv('wineQualityReds.csv', sep = ',')
```

Summary of the data set

```
dim(redWineData)
```

```
## [1] 1599   13
```

```
names(redWineData)
```

```
##  [1] "X"                    "fixed.acidity"        "volatile.acidity"
##  [4] "citric.acid"          "residual.sugar"       "chlorides"
##  [7] "free.sulfur.dioxide"  "total.sulfur.dioxide" "density"
## [10] "pH"                   "sulphates"            "alcohol"
## [13] "quality"
```

```
str(redWineData)
```

```
## 'data.frame':    1599 obs. of  13 variables:
##  $ X                   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ 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(redWineData)
```

```
##        X          fixed.acidity    volatile.acidity  citric.acid
##  Min.   :   1   Min.   : 4.60   Min.   :0.120    Min.   :0.000
##  1st Qu.: 400   1st Qu.: 7.10   1st Qu.:0.390    1st Qu.:0.090
##  Median : 800   Median : 7.90   Median :0.520    Median :0.260
##  Mean   : 800   Mean   : 8.32   Mean   :0.528    Mean   :0.271
##  3rd Qu.:1200   3rd Qu.: 9.20   3rd Qu.:0.640    3rd Qu.:0.420
##  Max.   :1599   Max.   :15.90   Max.   :1.580    Max.   :1.000
##  residual.sugar     chlorides       free.sulfur.dioxide total.sulfur.dioxide
##  Min.   : 0.90   Min.   :0.0120   Min.   : 1.0        Min.   :  6.0
##  1st Qu.: 1.90   1st Qu.:0.0700   1st Qu.: 7.0        1st Qu.: 22.0
##  Median : 2.20   Median :0.0790   Median :14.0        Median : 38.0
```

```
##  Mean   : 2.54   Mean   :0.0875   Mean   :15.9        Mean   : 46.5  
##  3rd Qu.: 2.60   3rd Qu.:0.0900   3rd Qu.:21.0        3rd Qu.: 62.0  
##  Max.   :15.50   Max.   :0.6110   Max.   :72.0        Max.   :289.0  
##     density             pH          sulphates        alcohol    
##  Min.   :0.990   Min.   :2.74   Min.   :0.330   Min.   : 8.4  
##  1st Qu.:0.996   1st Qu.:3.21   1st Qu.:0.550   1st Qu.: 9.5  
##  Median :0.997   Median :3.31   Median :0.620   Median :10.2  
##  Mean   :0.997   Mean   :3.31   Mean   :0.658   Mean   :10.4  
##  3rd Qu.:0.998   3rd Qu.:3.40   3rd Qu.:0.730   3rd Qu.:11.1  
##  Max.   :1.004   Max.   :4.01   Max.   :2.000   Max.   :14.9  
##     quality    
##  Min.   :3.00  
##  1st Qu.:5.00  
##  Median :6.00  
##  Mean   :5.64  
##  3rd Qu.:6.00  
##  Max.   :8.00
```

Observations from the summary

-> The amount of citric acid in the red wine varies mostly between 0 and 1.0.

-> 75% of the red wines have residual sugar content less than 2.6 in them but there are a few outliers whose residual sugar content can go right upto 15.5

-> The ingredients which are used in least amounts are sulphates, chlorides, citric acids and volatile acids.

-> The quality mostly hovers between 3 to 8, with the Mean being 5.6.

From the given summary results we have a few quantifiable results but none of them are leading us to any kind of causation yet. In order to surge ahead in that direction, we will need to explore the variables(ingredients) individually in univariate, bi-variate and multi-variate styles.

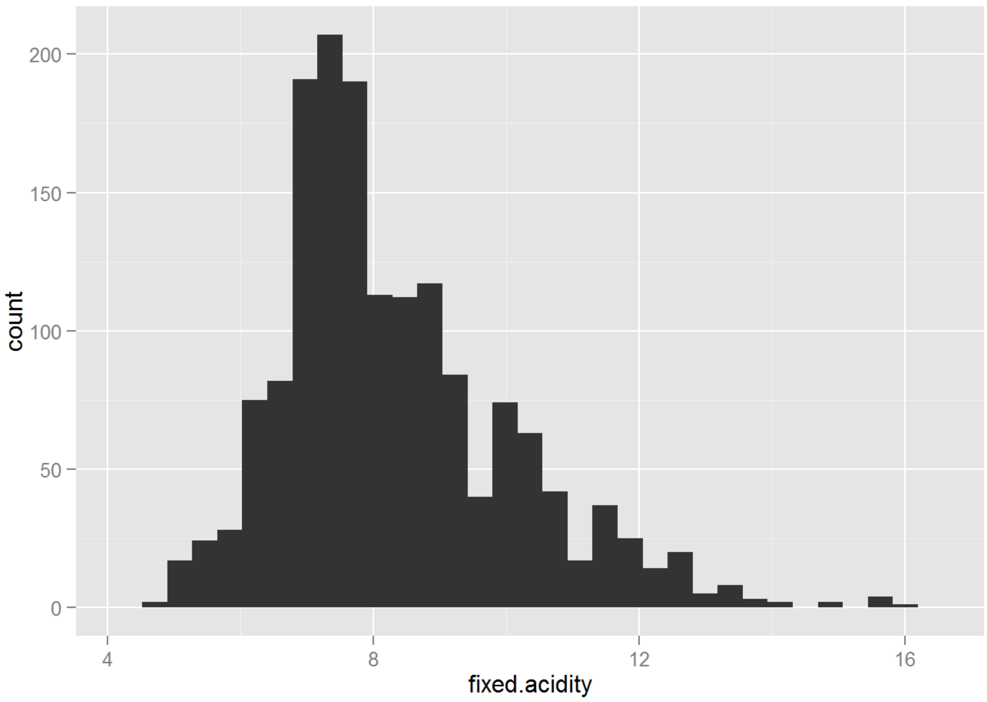
Understanding the distribution of single variables

I am going to analyze a few single variable distributions now. I am majorly going to be using histograms to represent my plot results.

Fixed acidity

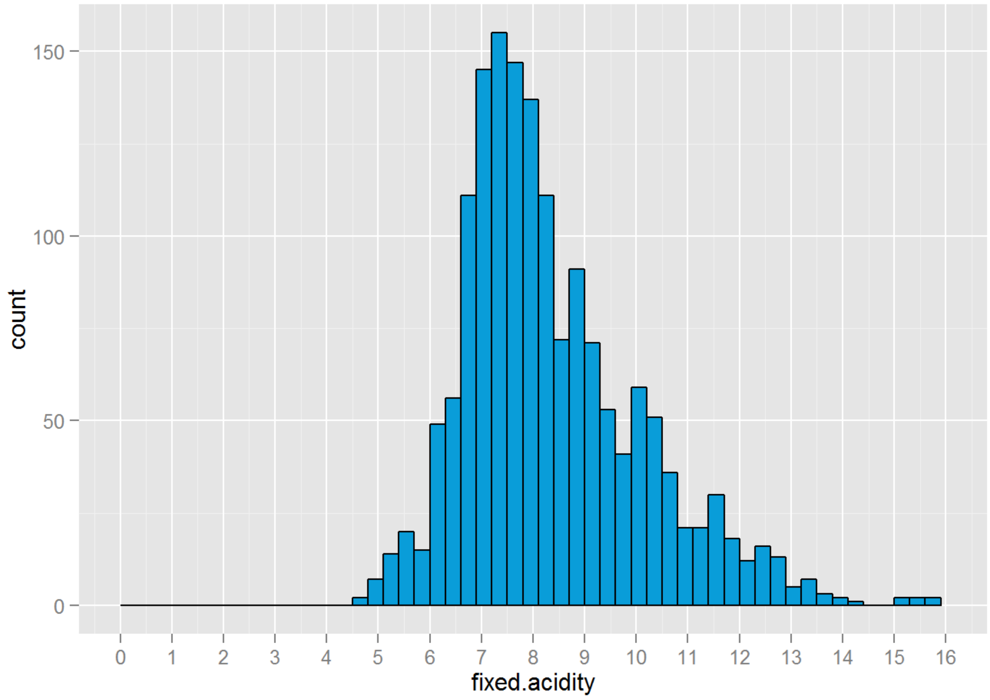
```r
library(ggplot2)
qplot(data = redWineData, fixed.acidity)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



Lets just refine our plot a bit

```
qplot( data = redWineData,
       fixed.acidity,
       binwidth = 0.3,
       fill = I('#099DD9'),
       color = I('black')
       ) +
  scale_x_continuous(breaks = seq(0,16,1), limits = c(0,16))
```



This looks good

```
min(redWineData$fixed.acidity)
```

```
## [1] 4.6
```

```
max(redWineData$fixed.acidity)
```

```
## [1] 15.9
```

Conclusion - Quite big groups of the wine samples have a fixed acidity between 6 to 10.0. There is no sample having a fixed acidity of 0, in fact the least fixed acidity is 4.6 and the highest is 15.9

```
max_fa_redWineData <- subset(redWineData,
                             redWineData$fixed.acidity >= 6.0 &  redWineData$fixed.acidity <= 10.0)

nrow(max_fa_redWineData)
```

```
## [1] 1288
```

```
nrow(redWineData)
```

```
## [1] 1599
```

About 80% of the samples have a fixed acidity between 6.0 to 10.0.

Volatile acidity

```
max(redWineData$volatile.acidity)
```

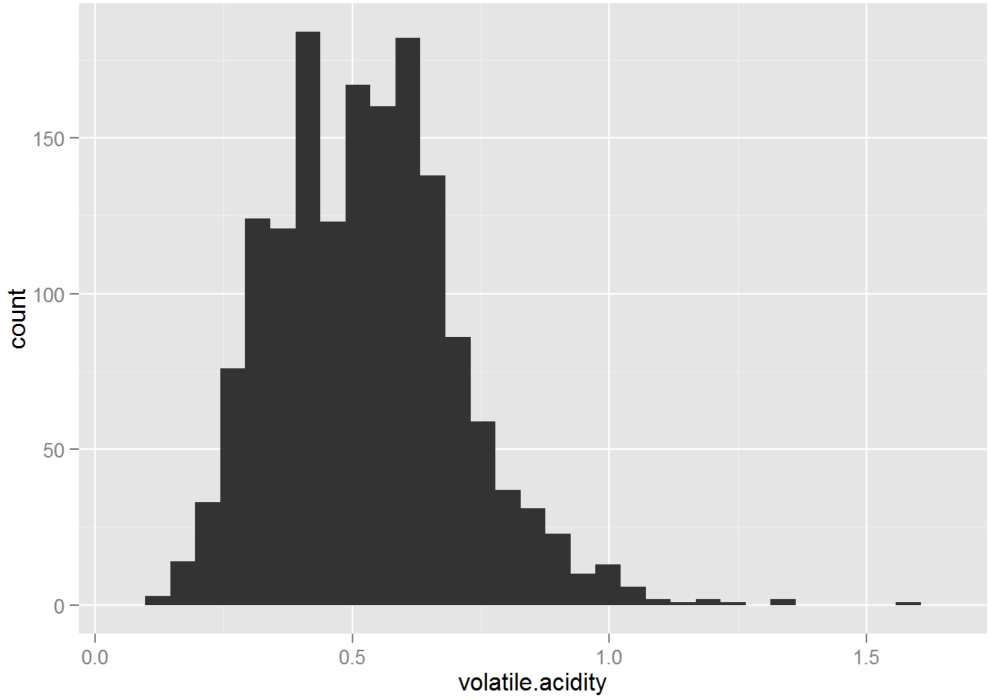
```
## [1] 1.58
```

```
min(redWineData$volatile.acidity)
```

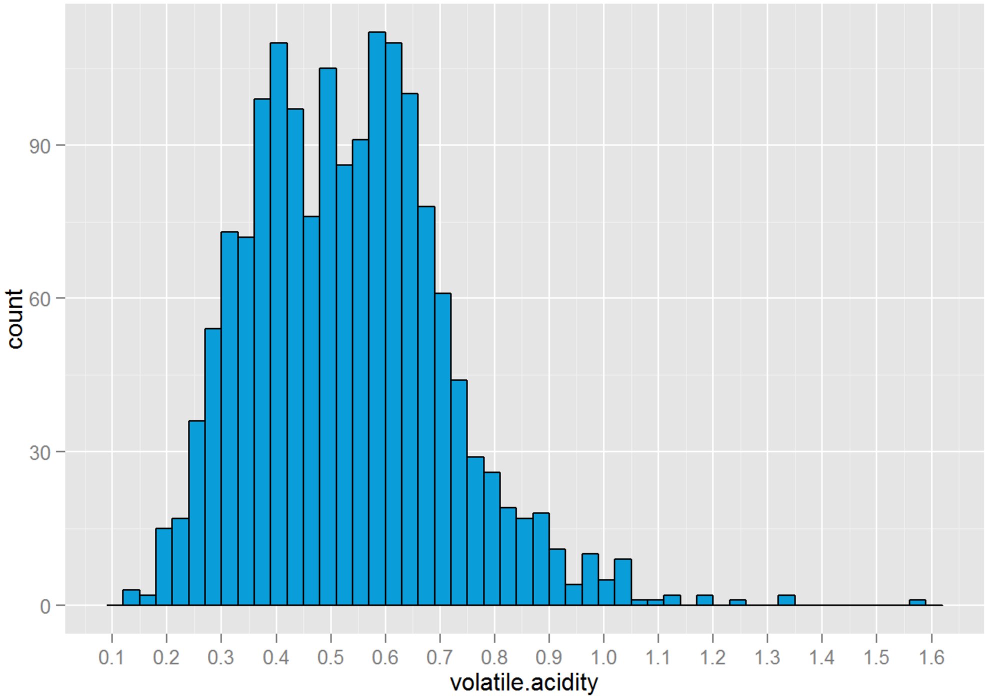
```
## [1] 0.12
```

```
qplot(data = redWineData, volatile.acidity)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
qplot(data = redWineData,
      volatile.acidity,
      color = I('black'),
      fill = I('#099DD9'),
      binwidth = 0.03
      ) +
  scale_x_continuous(breaks = seq(0,2,0.1))
```



Conclusion - Majorityof the wine samples have a volatile acidity between 0.3 to 0.7. There is no sample having a volatile acidity of 0. There are very few samples having volatile acidity above 1. Most of the samples have a volatile acidity less than 1.0.

Let us just verify our claims about volatile acidity.

```
max_va_redWineData <- subset(redWineData,
                             redWineData$volatile.acidity >= 0.3 &  redWineData$volatile.acidity <= 0.7)

nrow(max_va_redWineData)
```

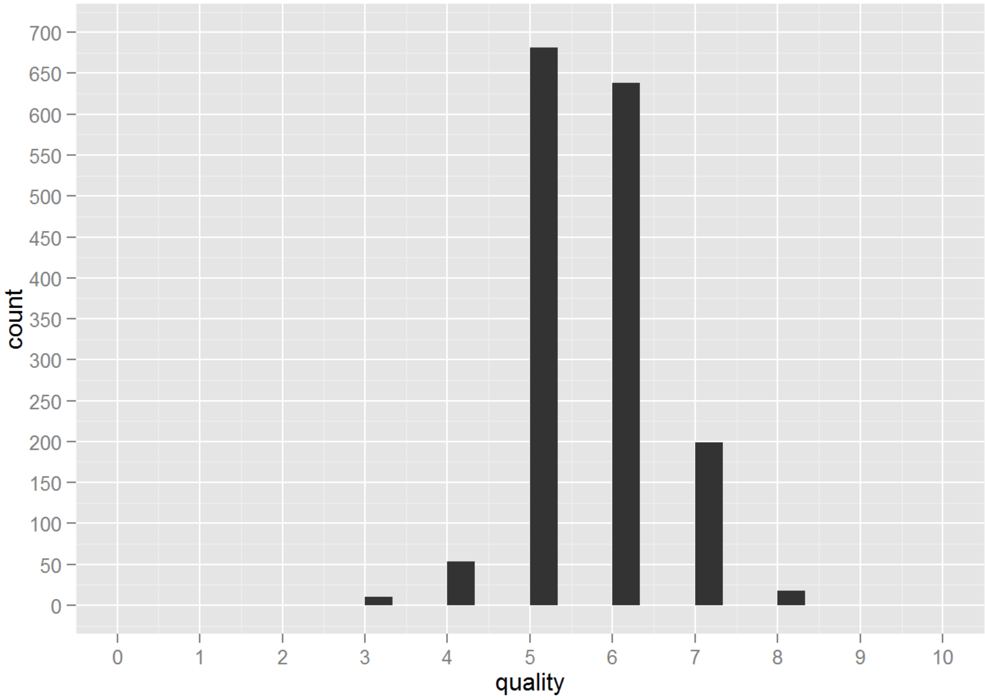
```
## [1] 1249
```

Ok, now the above figures verify our claims. About 79% of the wine samples have a volatile acidity between 0.3 to 0.7.

Ok..Let me just check the quality of the samples

```
qplot( data = redWineData,
       x = quality
     ) +
  scale_x_continuous(breaks = seq(0,10,1), limits = c(0,10)) +
  scale_y_continuous(breaks = seq(0,700,50),limits = c(0,700))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
max_quality_redWineData <- subset(redWineData,
                                  redWineData$quality == 5 | redWineData$quality == 6
                                 )

nrow(max_quality_redWineData)
```

```
## [1] 1319
```

Now this one really answers a few important questions. More than 82% of the wine samples have either a 5 or 6 quality on a scale of 1 to 10. Not even a single sample has a 0,1,2,9,10 quality.

Ideally as a wine company owner I would want majority of my samples to have a quality of more than 8, but then it would also depend a lot upon costing and profit-margins and other business factors

The above results lead us to a correlation between the acidity and the quality of the wines. There is a high correlation between the quality being 5 and 6 when the volatile acidity between 0.3 to 0.7 and fixed acidity between 6.0 to 10.0. But correlation does not necessarily lead to causation. Meaning that, the results of acidity might or might not be responsible for the quality being 5 and 6.

In order to be confident about the above correlation we will need to subset data with volatile acidity 0.3 and 0.7 and fixed acidity between 6.0 to 10.0 and check the quality of that data. Lets do that

```
guess_data1 <- subset(max_va_redWineData,
                      max_va_redWineData$quality == 5 | max_va_redWineData$quality == 6)
guess_data2 <- subset(max_fa_redWineData,
                      max_fa_redWineData$quality == 5 | max_fa_redWineData$quality == 6
                      )

nrow(guess_data1)
```

```
## [1] 1051
```

```
nrow(guess_data2)
```

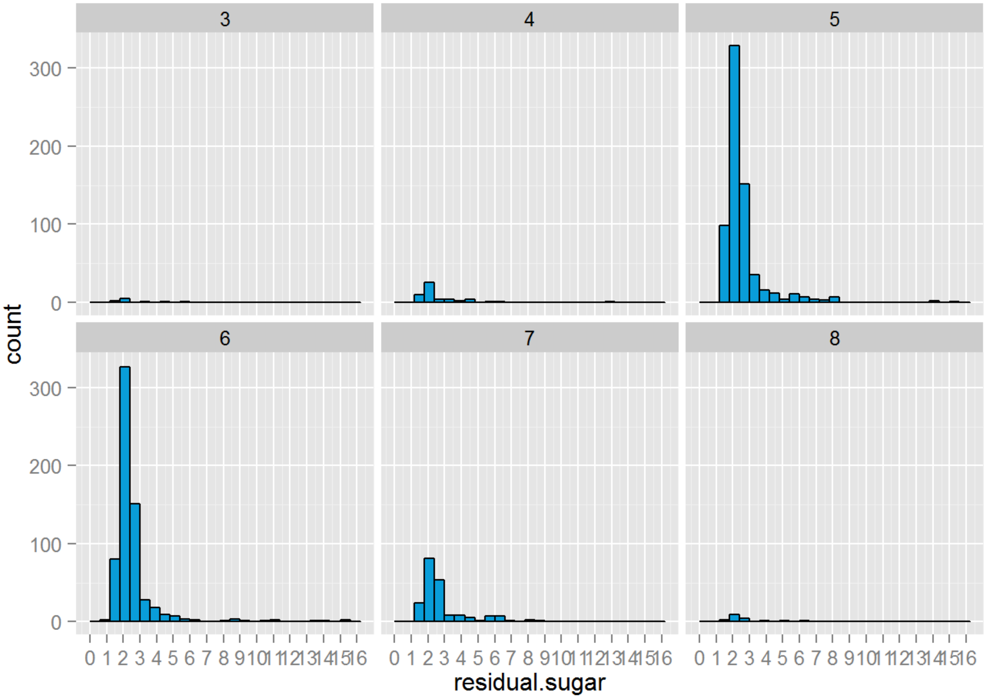
```
## [1] 1094
```

Our assumption is verified. Approximately 80% of the wine samples that have volatile acidity between 0.3 and 0.7 and fixed acidity between 6.0 to 10.0 have a quality that is either 5 or 6.

Faceting

Let’s analyse the residual sugar content in the wines faceted by quality. This would give us a fair idea about the distribution of residual sugar content across the different quality levels.

```
qplot( data = redWineData,
       x = residual.sugar,
       binwidth = 0.6,
       color = I('black'),
       fill = I('#099DD9')
     ) +
  scale_x_continuous(breaks = seq(0,16,1)) +
  facet_wrap(~quality)
```




```
max(redWineData$residual.sugar)
```

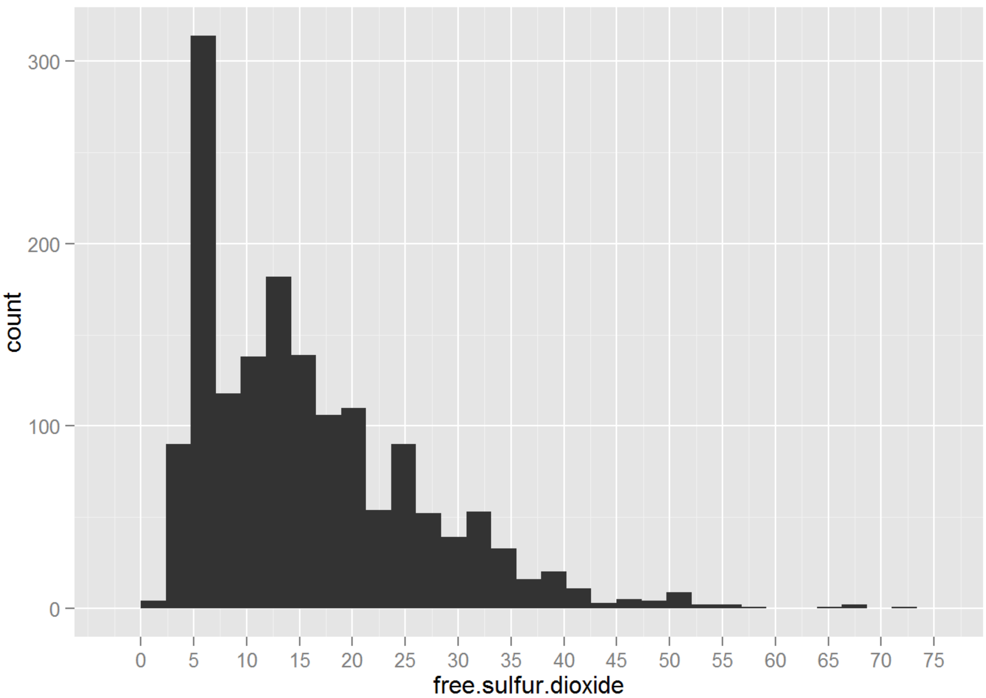
```
## [1] 15.5
```

The distribution of residual sugar is prominently seen in the facets showing quality 5 and 6. Most of these wines (having quality 5 and 6) have residual sugar content under 5.

Exploring free sulphur dioxide.

```
qplot( data = redWineData,
       x = free.sulfur.dioxide,
     ) +
   scale_x_continuous(breaks = seq(0,75,5))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

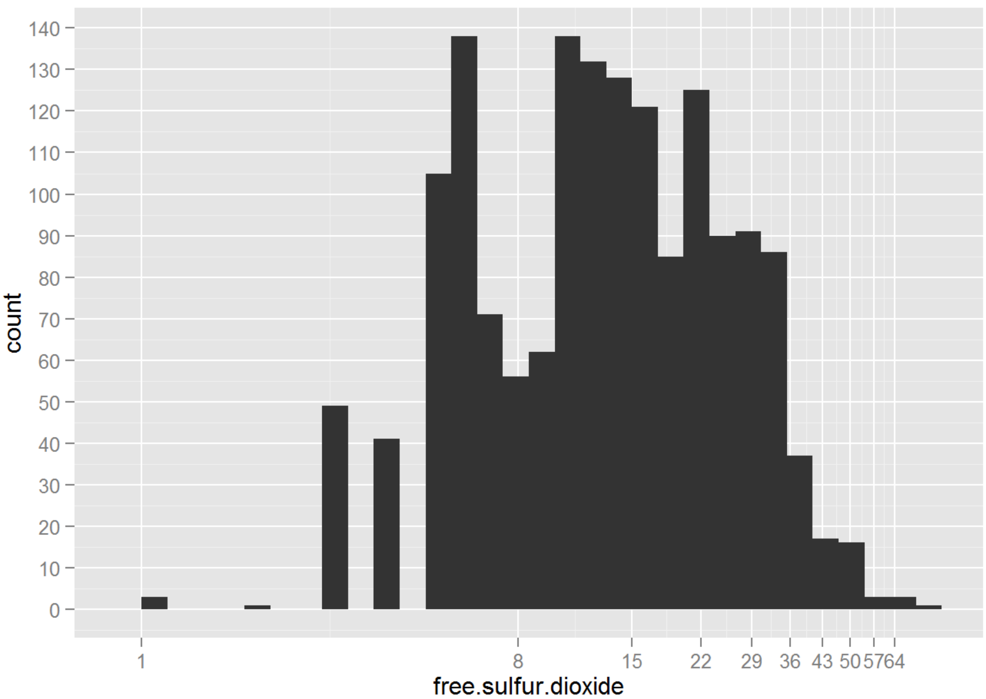


Majority of the samples have a free sulfur dioxide content under 30.

Now we are seeing a long tail here after free sulfur dioxide goes beyond 40. Lets add a log transformation to our code to address this issue.

```
library(scales)

qplot( data = redWineData,
       x = free.sulfur.dioxide,
     ) +
   scale_x_continuous(trans = log10_trans(), breaks = seq(1,70,7)) +
  scale_y_continuous(breaks = seq(0,140,10))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



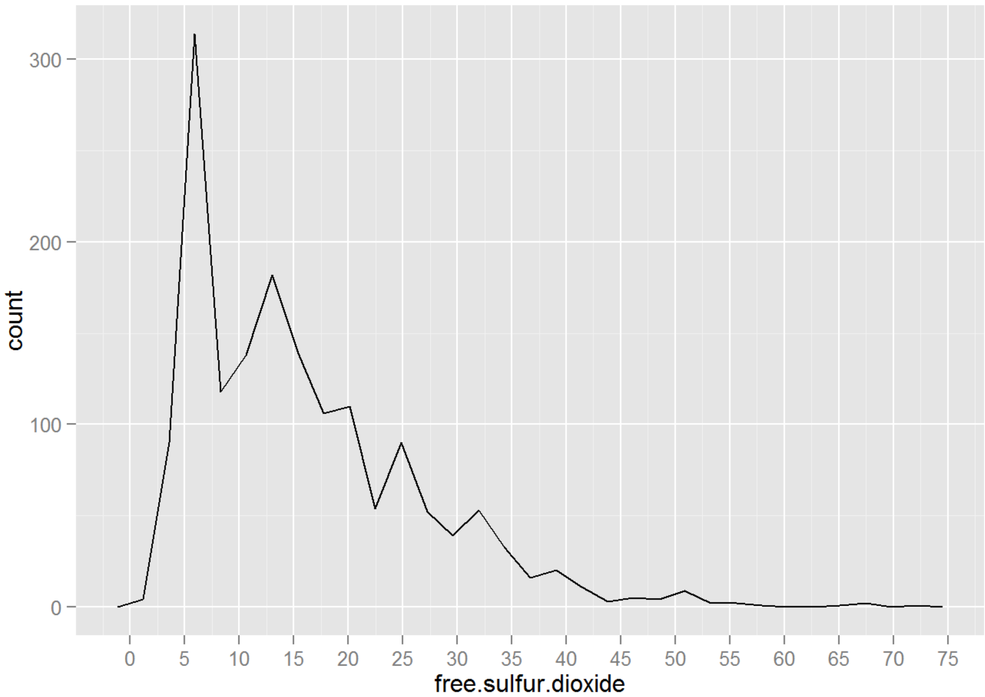
There are very few samples having free sulfur dioxude more than 35

Lets try the same variable with frequency polygon

Frequency polygon

```
qplot( data = redWineData,
       x = free.sulfur.dioxide,
       geom = 'freqpoly',
     ) +
  scale_x_continuous(breaks = seq(0,75,5))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



Conclusion - Majority of the samples have a free sulfur dioxide content under 30.

```
fsd_under30 <- subset(redWineData, free.sulfur.dioxide <= 30)
nrow(fsd_under30)
```

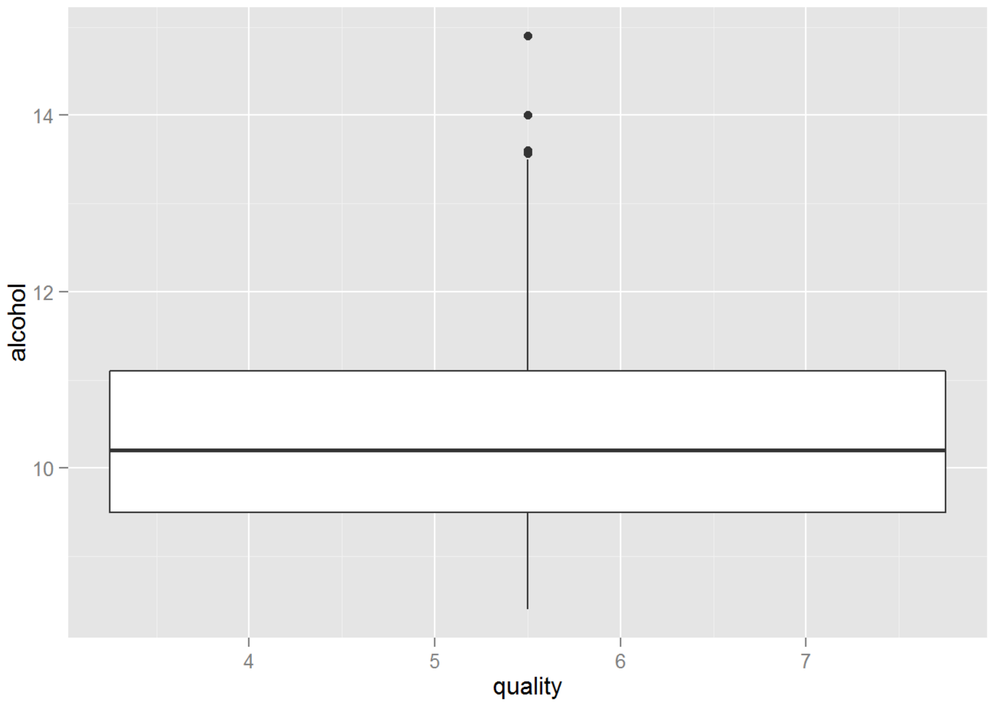
```
## [1] 1436
```

Our freq polygon results are verified. Approximately 90% of the samples have a free sulfur dioxide value under 30.

Box Plots

Lets try to check out the quality of the samples against the alcohol content.

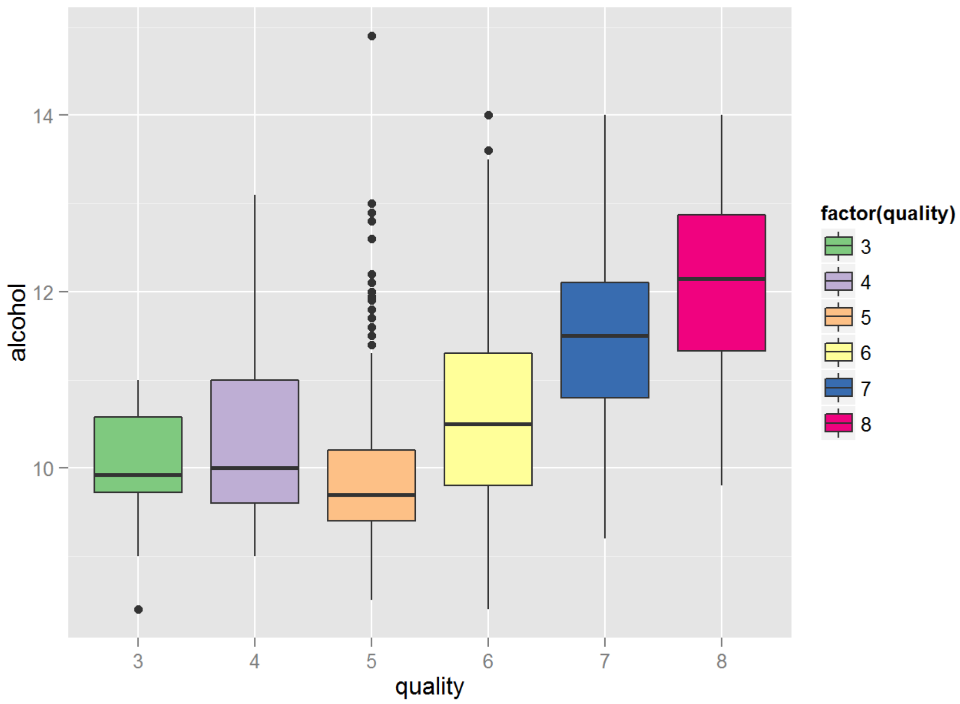
```
qplot(data = redWineData,
      x = quality,
      y = alcohol,
      geom = 'boxplot',
    )
```



Here we will need to factor the variable quality first

Factorisation

```
redWineData$qualityfact <- factor(redWineData$quality,
                                  levels = c('1','2','3','4','5','6','7','8','9','10')
                                 )

library(RColorBrewer)

qplot(data = redWineData,
      x = qualityfact,
      y = alcohol,
      geom = 'boxplot',
      binwidth = 0.1,
      fill = factor(quality),
      xlab = "quality"
      ) +
  scale_fill_brewer(type = "qual")
```



Conclusion : Median alcohol content is highest for the samples with quality 8

Lets verify those results attained through the box plots

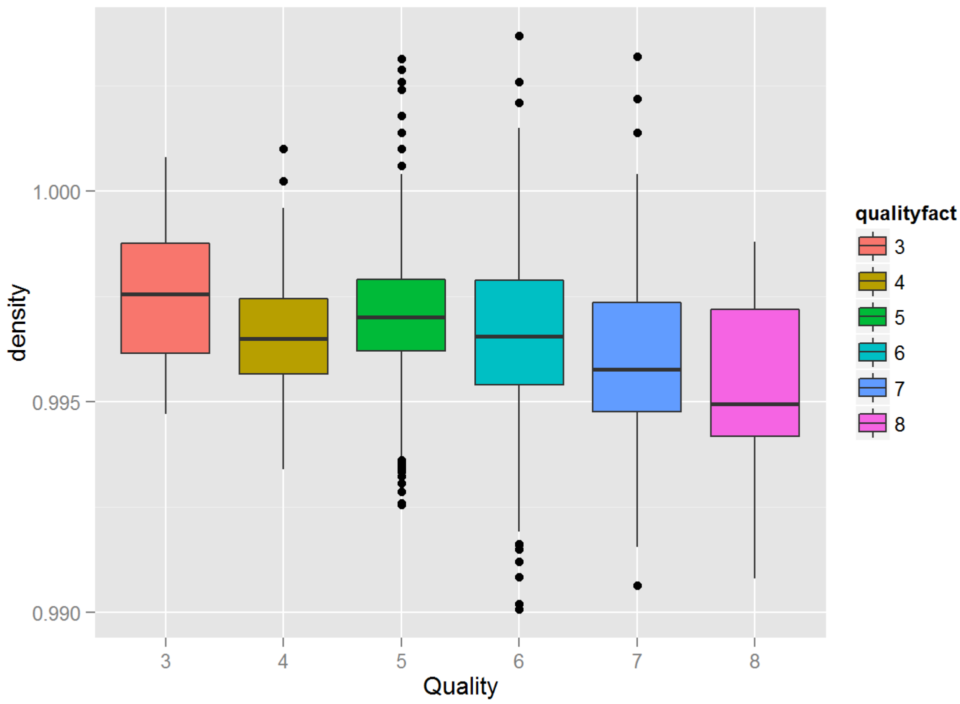
```
by(redWineData$alcohol,redWineData$quality,summary)
```

```
## redWineData$quality: 3
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    8.40    9.72    9.93    9.96   10.60   11.00 
## -------------------------------------------------------- 
## redWineData$quality: 4
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     9.0     9.6    10.0    10.3    11.0    13.1 
## -------------------------------------------------------- 
## redWineData$quality: 5
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     8.5     9.4     9.7     9.9    10.2    14.9 
## -------------------------------------------------------- 
## redWineData$quality: 6
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     8.4     9.8    10.5    10.6    11.3    14.0 
## -------------------------------------------------------- 
## redWineData$quality: 7
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     9.2    10.8    11.5    11.5    12.1    14.0 
## -------------------------------------------------------- 
## redWineData$quality: 8
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     9.8    11.3    12.2    12.1    12.9    14.0
```

Our results from the box-plot analysis are verified. Median alcohol content is indeed on the higher side for wines with high quality.

Lets test the impact of density on quality of the samples

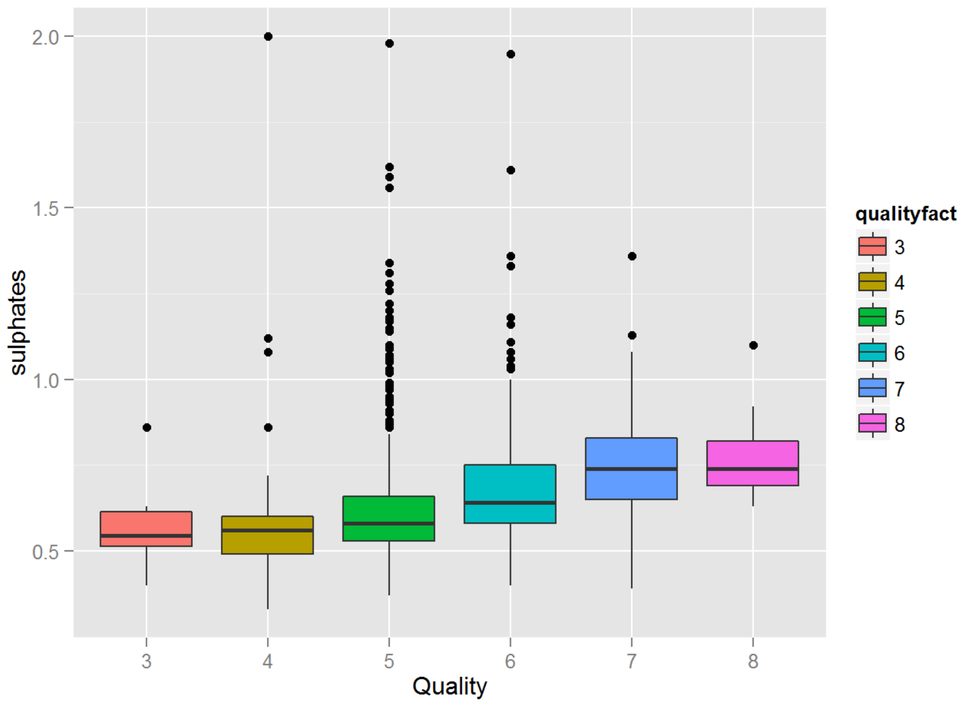
```
ggplot( data = redWineData,
        aes(qualityfact,density, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality")
```



What we observe from the above plots is that, the median density is highest for the quality level 3 and in general, density of the samples goes on decreasing as the quality goes on increasing.

Now, lets test the impact of sulphate proportion on the quality of the samples

```
ggplot( data = redWineData,
        aes(qualityfact,sulphates, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality")
```



What we see here is that as the quality goes on increasing the median content of sulphates goes on increasing.

Revisiting the analysis goals

Let's just get back to our purpose of doing this analysis. Let's put our-self in the shoes of the Product Manager, for a moment. As a product manager, I would be interested in maximizing the quality of my product, the wine in this case and minimizing the cost of production. Such, data analysis of our current set of products and it's ingredients can help me immensely as a Product Manager. I know a few important things

1. Current status of my products(quality, cost etc)

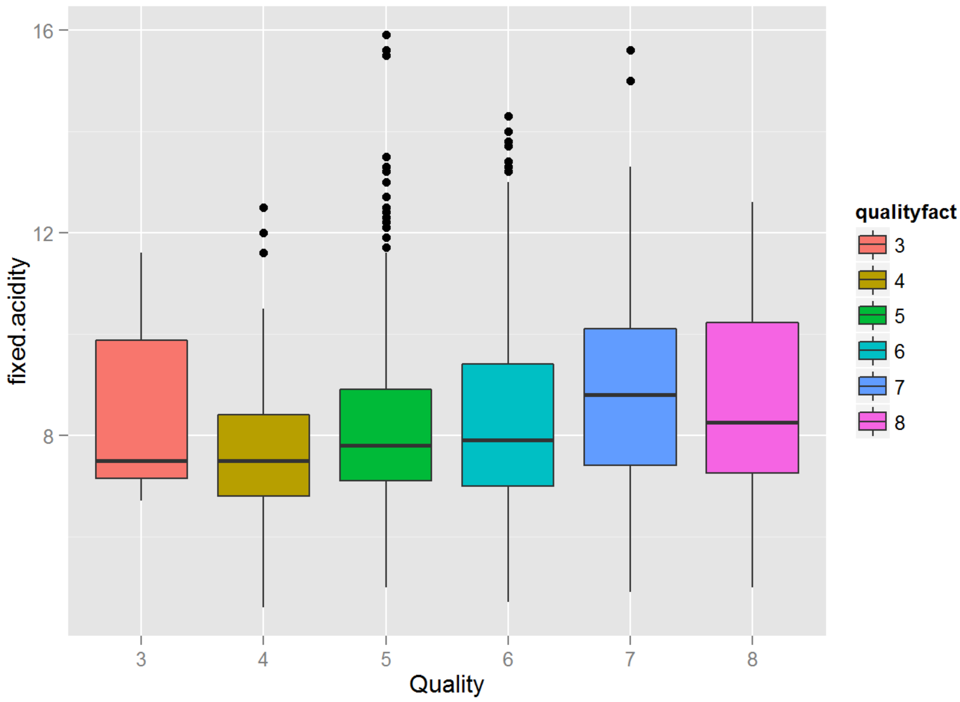
2. Contribution of the ingredients in deriving the kind of quality that they are deriving.
3. Possibilities of cost-cutting, in case we come up with an analysis that shows that too much of attention is being given to a costlier ingredient when, we can do away with cheaper ones, without having to sacrifice the quality much.

We have figured out from our univariate analysis that all the ingredients influence the quality of the redwine in some or the other ways. But in order to find out the ingredients which predominantly affect the quality of the redwine, we need to perform a bivariate analysis of these variables along with the quality

Bivariate analysis using ggplot syntax

Fixed Acidity Vs Quality

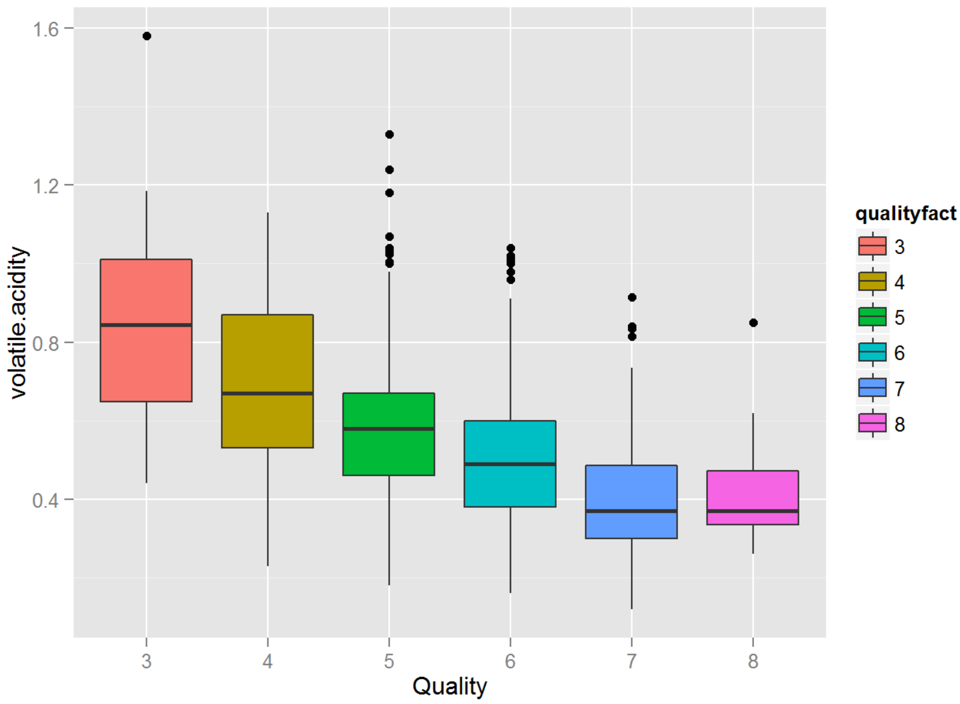
```
ggplot( data = redWineData,
        aes(qualityfact, fixed.acidity, fill = qualityfact),
      ) +
  geom_boxplot() +
  xlab("Quality")
```



From the above box-plots, it is clear that fixed acidity remains fairly constant over all the quality levels.

Volatile acidity VS Quality

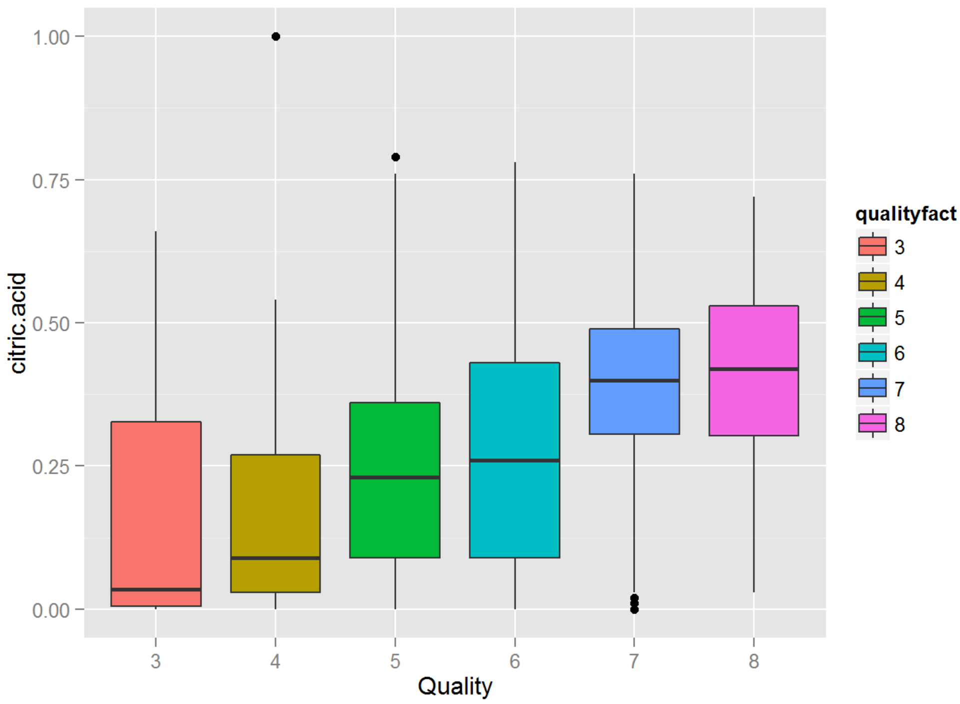
```
ggplot( data = redWineData,
        aes(qualityfact, volatile.acidity, fill = qualityfact),
      ) +
  geom_boxplot() +
  xlab("Quality")
```



As the volatile acidity decreases the quality of the wine goes on increasing.

Citric acid VS Quality

```
ggplot( data = redWineData,
        aes(qualityfact, citric.acid, fill = qualityfact),
      ) +
  geom_boxplot() +
  xlab("Quality")
```

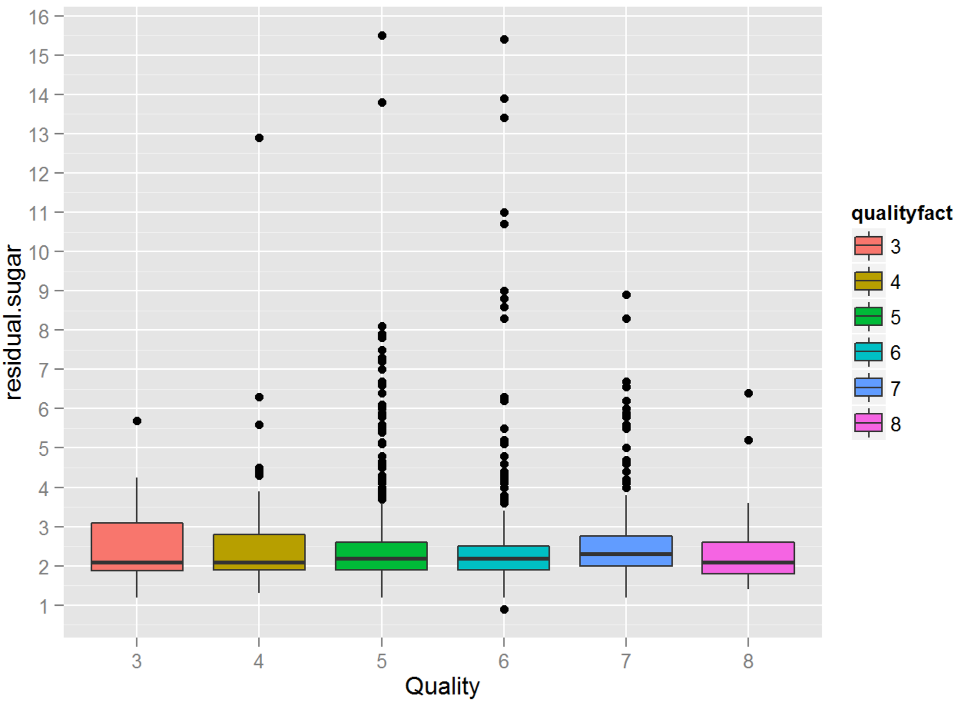


The quality of the redwines tends to have an increasing trend with an increase in citric acid.

Residual sugar VS Quality

```
ggplot( data = redWineData,
        aes(qualityfact, residual.sugar, fill = qualityfact),
      ) +
  geom_boxplot() +
  xlab("Quality") +
```

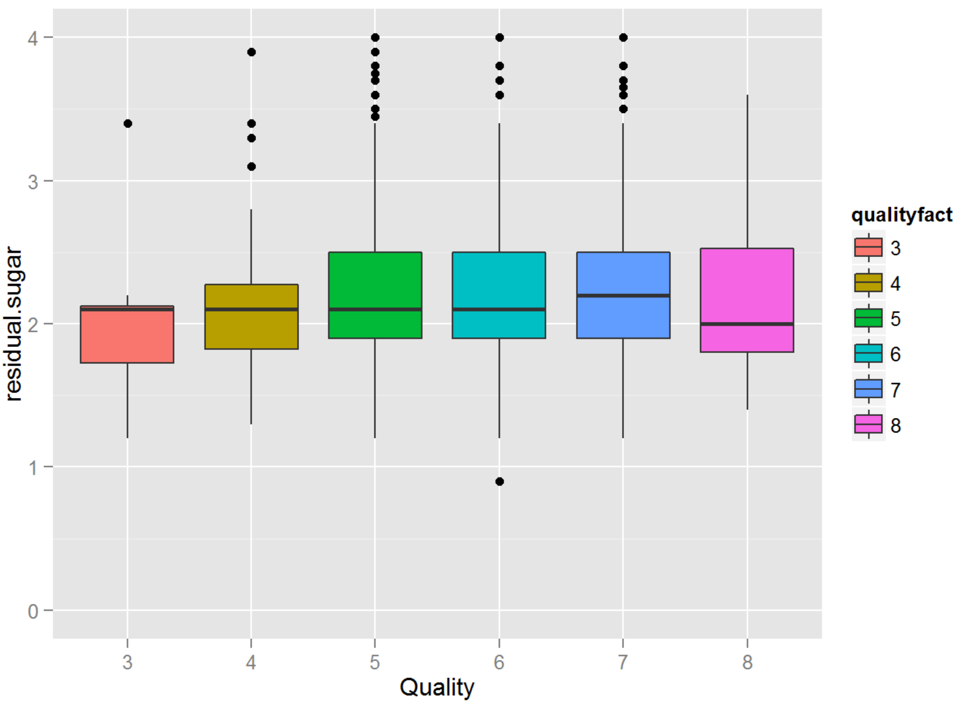
```
  scale_y_continuous(breaks = seq(0,16,1))
```



Here , we can guess that the residual sugar is more or less at the same level but the output is kinda squished because of the large number of outliers. Lets bring the focus on the box plots.

```
ggplot( data = redWineData,
        aes(qualityfact, residual.sugar, fill = qualityfact),
      ) +
  geom_boxplot() +
  xlab("Quality") +
  scale_y_continuous(breaks = seq(0,16,1), limits = c(0,4))
```

```
## Warning: Removed 125 rows containing non-finite values (stat_boxplot).
```



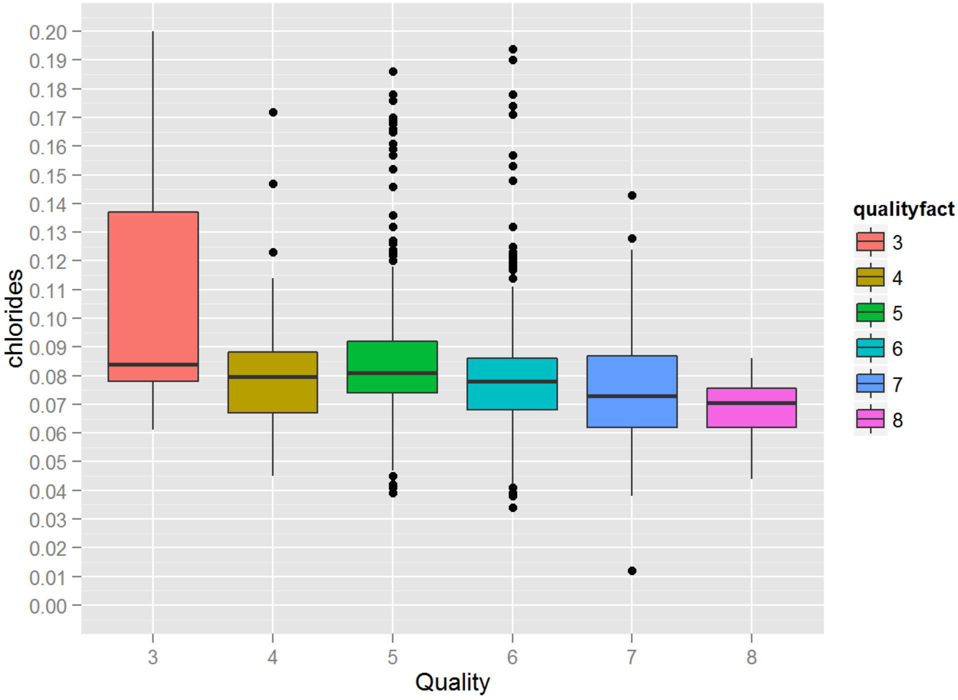
The above box-plots show that, the amount of residual sugar remains fairly constant through all the quality levels.

Chlorides VS Quality

```
ggplot( data = redWineData,
```

```
        aes(qualityfact, chlorides, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality") +
  scale_y_continuous(breaks = seq(0,0.2,0.01), limits = c(0,0.2))
```

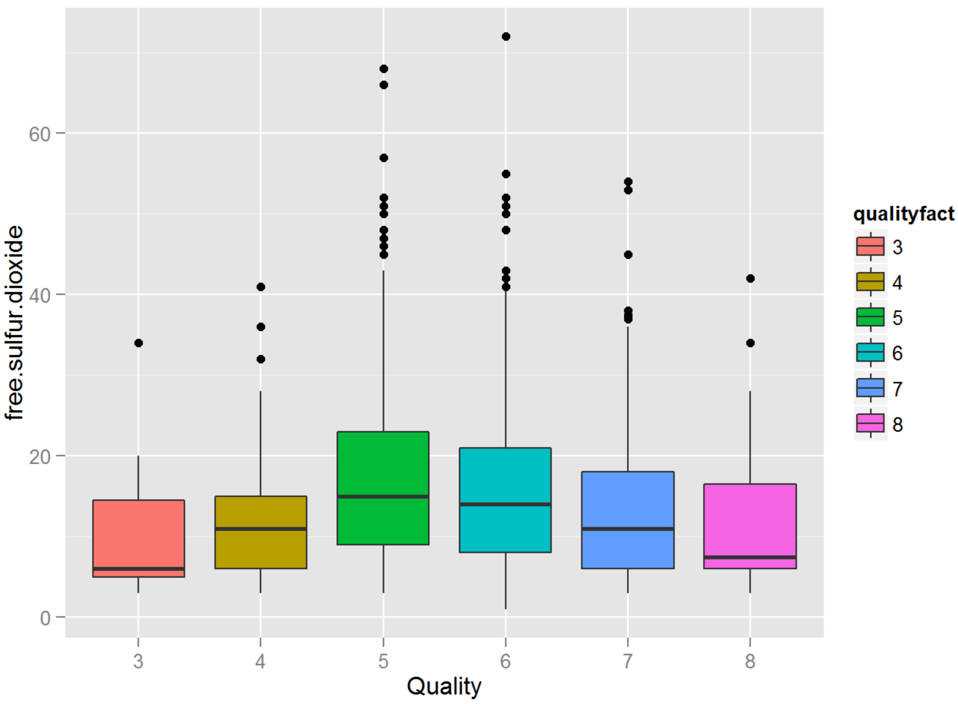
```
## Warning: Removed 41 rows containing non-finite values (stat_boxplot).
```



The above box-plots show that, the amount of chlorides remains fairly constant through all the quality levels.

Free sulphur dioxide VS Quality

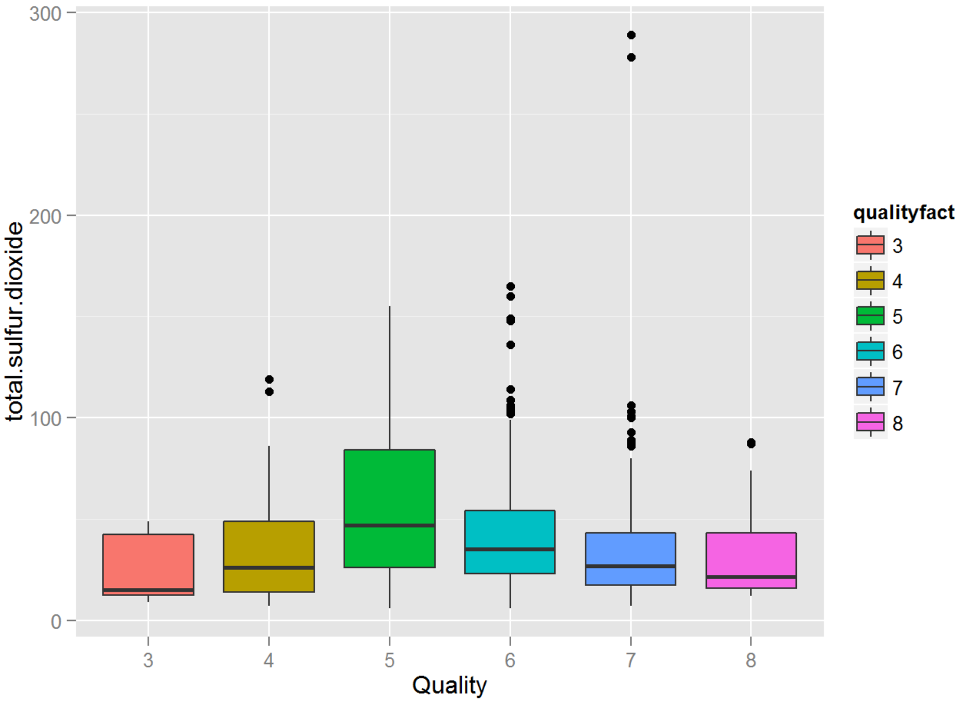
```
ggplot( data = redWineData,
        aes(qualityfact,free.sulfur.dioxide, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality")
```



The amount of free sulphur dioxide varies across different quality levels.

Total Sulphur dioxide VS Quality

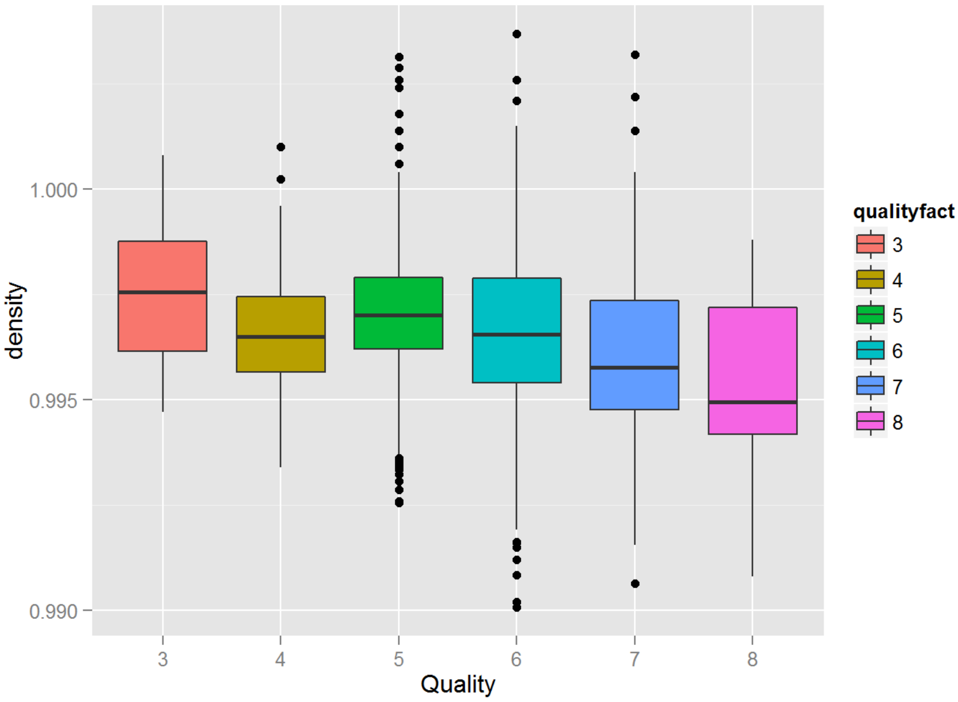
```
ggplot( data = redWineData,
        aes(qualityfact,total.sulfur.dioxide, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality")
```



The trend of total sulphur dioxide is very similar to that of free sulphur dioxide w.r.t quality.

Density VS Quality

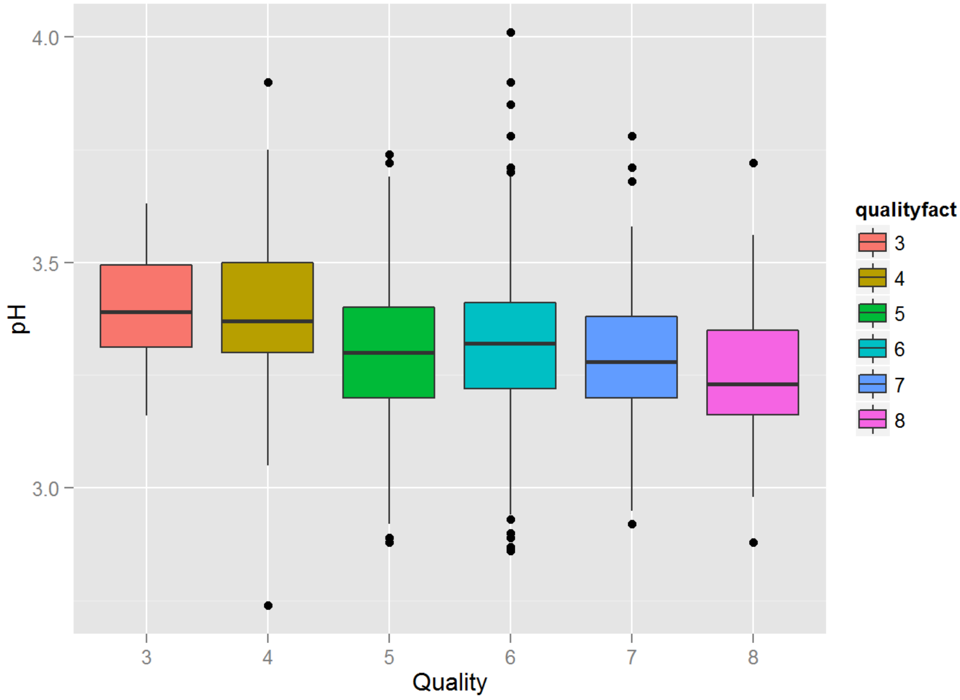
```
ggplot( data = redWineData,
        aes(qualityfact,density, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality")
```



The density gradually decreases, as the quality goes on increasing.

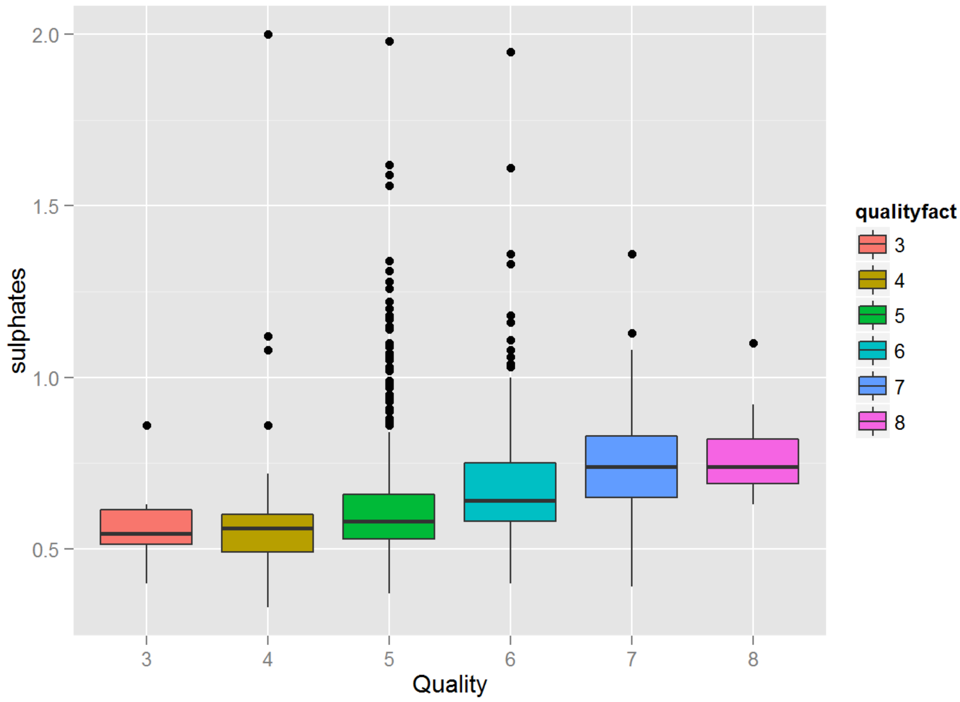
pH vs Quality

```
ggplot( data = redWineData,
        aes(qualityfact,pH, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality")
```



sulphates VS Quality

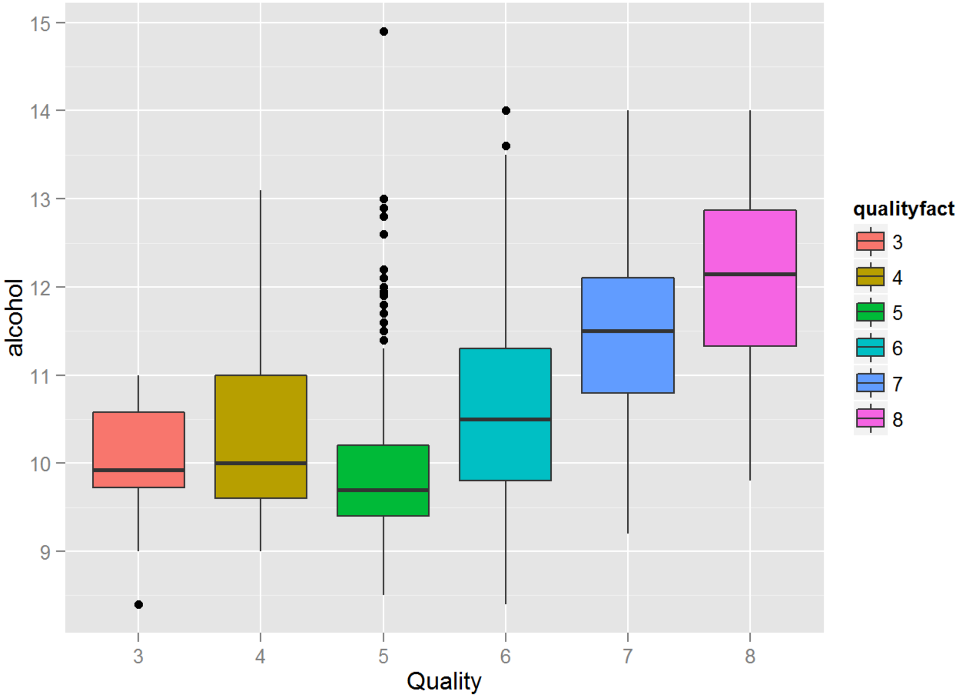
```
ggplot( data = redWineData,
        aes(qualityfact, sulphates, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality")
```



With a steady increase in quality, increase in sulphates.

alcohol VS quality

```
ggplot( data = redWineData,
        aes(qualityfact,alcohol,fill = qualityfact)
      ) +
  geom_boxplot() +
  scale_y_continuous(breaks = seq(0,16,1)) +
  xlab("Quality")
```



What we see from the above plots is that, as the quality goes on improving the median alcohol content goes on increasing

Our analysis so far has been carried out with quality being the response variable and other variables being the predictor variables.

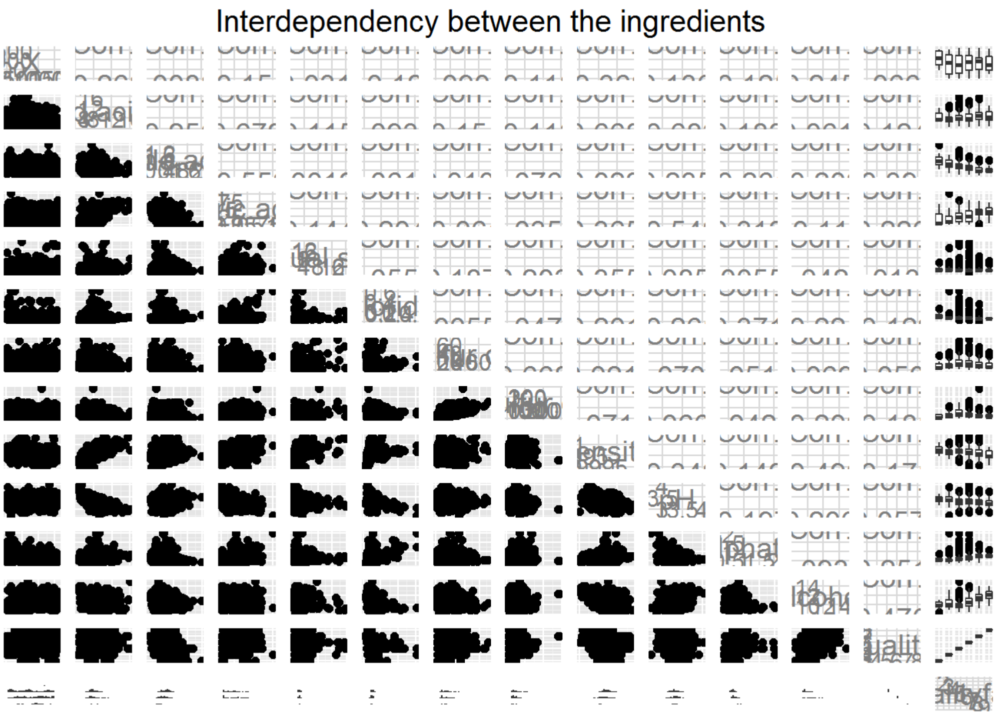
Some of the variables have a strong impact on the quality of redwine while some don't. However, it's not clear from the analysis so far, whether these variables independently have an impact on the quality of the redwine or not. Is it because of the combination with some other variable, that the impact is created or not. We cannot be certain as of now about whether the impact is independent or not.

After doing a bit of research on the internet , I figured that chi-squared test could be a correct way to figure out which amongst the above variables have a dependency between them.

However chi-squared test is more suitable for identifying relationships between samples of the population.

We will have to test the interdependency between the factors affecting the quality of the redwine. A correlation coefficient matrix would come in handy for that purpose.

```
library(GGally)
set.seed(1000)
good_quality_subset <- good_quality[,c(2:12)]

ggpairs(redWineData[sample.int(nrow(redWineData), ), ],
        title = "Interdependency between the ingredients"
       )
```



The following pairs of ingredients have a relatively strong correlation

1. Fixed Acidity and Citric Acid(+ve correlation)
2. Fixed Acidity and Density(+ve correlation)
3. Fixed Acidity and pH(-ve correlation)
4. Free Sulphur Dioxide and Total Sulphur Dioxide(+ve correlation)

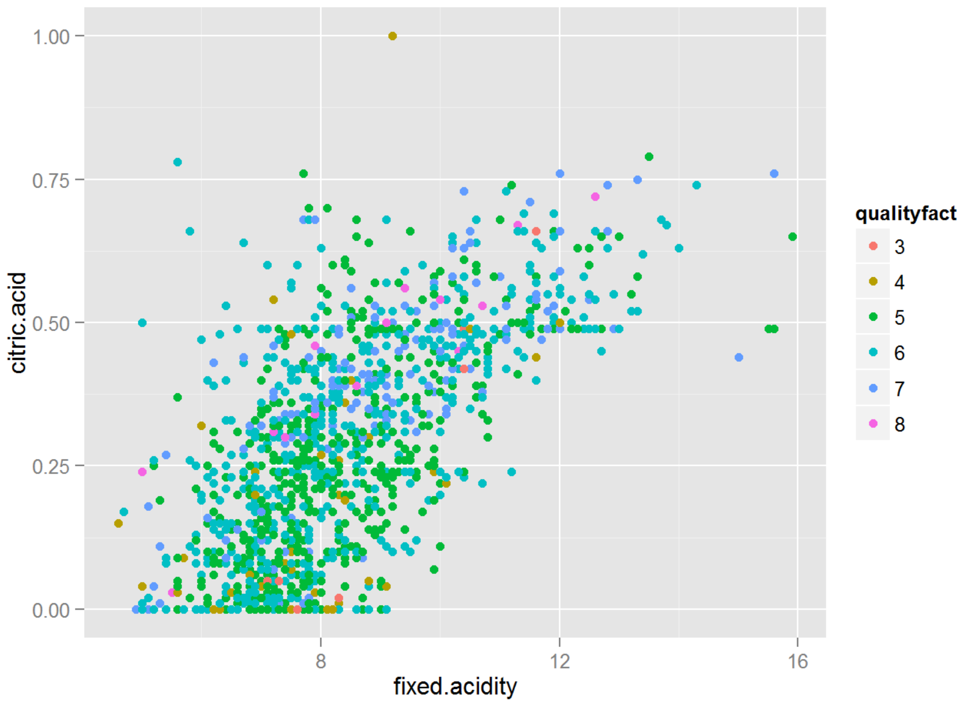
This indicates that if one of the above factors affects the quality of the red wine then its impact is supplemented by the other factor that it has a strong correlation with.

Thus, we reach some of the following conclusions, - In order to improve the quality of the redwine, we need to increase the fixed acidity with a subtle increase in the citric acid. However,I am yet to figure out the proportion of citric acid that needs to be increased with Fixed acidity.

- In order to improve the quality of the redwine, we need to increase the fixed acidity with a subtle increase in the density. However,I am yet to figure out the proportion of density that needs to be increased with Fixed acidity.

Now that we have figured out that there is strong correlation between some of the ingredients, we know that value of one can help us in predicting the value of another. Linear regression could help us the recognize the change that needs to be brought up in one variable given a change in another variable.

```
ggplot( data = redWineData,
        aes(fixed.acidity, citric.acid, color = qualityfact),
      ) +
  geom_point()
```



The above scatterplot clearly indicates a linear dependency between fixed acidity and citric acid.

```
linearModel1  <- lm(redWineData$fixed.acidity ~ redWineData$citric.acid)

p <- ggplot( data = redWineData,
        aes(x = fixed.acidity, y = citric.acid)
      ) +
  geom_point()

p1 <- p + geom_smooth(method = "lm", formula = y~x) + ggtitle('Linear Model for Fixed acidity VS Citric Acid')

summary(linearModel1)
```

```
##
## Call:
## lm(formula = redWineData$fixed.acidity ~ redWineData$citric.acid)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -5.776 -0.815 -0.033  0.806  5.965
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               6.6928     0.0553   121.0   <2e-16 ***
## redWineData$citric.acid   6.0036     0.1657    36.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.29 on 1597 degrees of freedom
## Multiple R-squared:  0.451,  Adjusted R-squared:  0.451
## F-statistic: 1.31e+03 on 1 and 1597 DF,  p-value: <2e-16
```

Since the p-value is less than 0.05(assuming the alpha = 0.05), we reject the null hypothesis that, there is no dependency between fixed acidity and citric acid. In other words we conclude that there is a linear dependency between fixed acidity and citric acid.

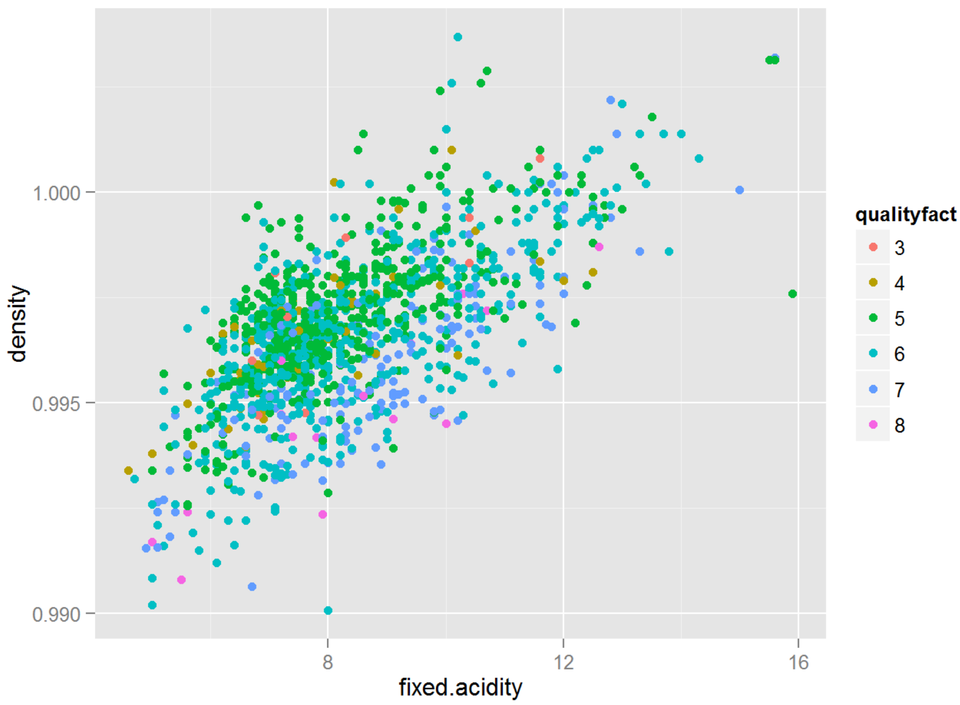
fixed.acidity = 6.692 + citric.acid*6.003

Using the above equation, we can predict the fixed acidity given we have a citric acid content. The above equation can help us add fixed acidity and citric acid in a measured way in order to improve the quality of red wine samples that we have.

Similarly let us build some more linear models based on the results we have from the correlation matrix

```
ggplot( data = redWineData,
```

```
       aes(fixed.acidity, density, color = qualityfact),
     ) +
  geom_point()
```



Here again, we see a similar pattern of linear dependency between fixed acidity and density.

```
linearModel2 <- lm(redWineData$fixed.acidity ~ redWineData$density)

p <- ggplot( data = redWineData,
        aes(x = fixed.acidity, y = density)
      ) +
  geom_point()

p2 <- p + geom_smooth(method = "lm", formula = y~x) + ggtitle('Linear Model for Fixed acidity VS Density')


summary(linearModel2)
```

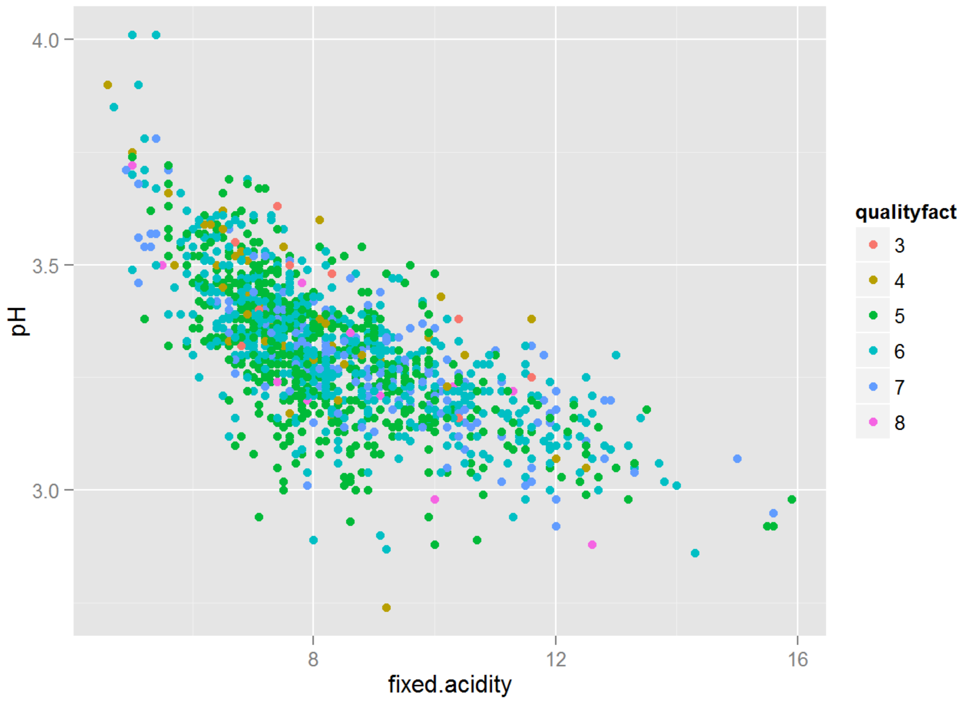
```
##
## Call:
## lm(formula = redWineData$fixed.acidity ~ redWineData$density)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.355 -0.885 -0.241  0.804  7.054
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -606.0       17.1   -35.4   <2e-16 ***
## redWineData$density    616.3       17.2    35.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.3 on 1597 degrees of freedom
## Multiple R-squared:  0.446,  Adjusted R-squared:  0.446
## F-statistic: 1.29e+03 on 1 and 1597 DF,  p-value: <2e-16
```

Since the p-value is less than 0.05(assuming the alpha = 0.05), we reject the null hypothesis that, there is no dependency between fixed acidity and density.

fixed.acidity = -605.96 + density * 616.28

Using the above equation, we can predict the fixed acidity using the density

```
ggplot(data = redWineData,
       aes(fixed.acidity, pH, color = qualityfact)
      ) +
  geom_point()
```



Now here is another interesting trend. We see a linear dependency between fixed acidity and pH. But it is a negative linear dependency.

```
linearModel3 <- lm(redWineData$fixed.acidity ~ redWineData$pH)

p <- ggplot( data = redWineData,
        aes(x = fixed.acidity, y = pH)
      ) +
  geom_point()

p3 <- p + geom_smooth(method = "lm", formula = y~x) + ggtitle('Linear Model for Fixed acidity VS pH')


summary(linearModel3)
```

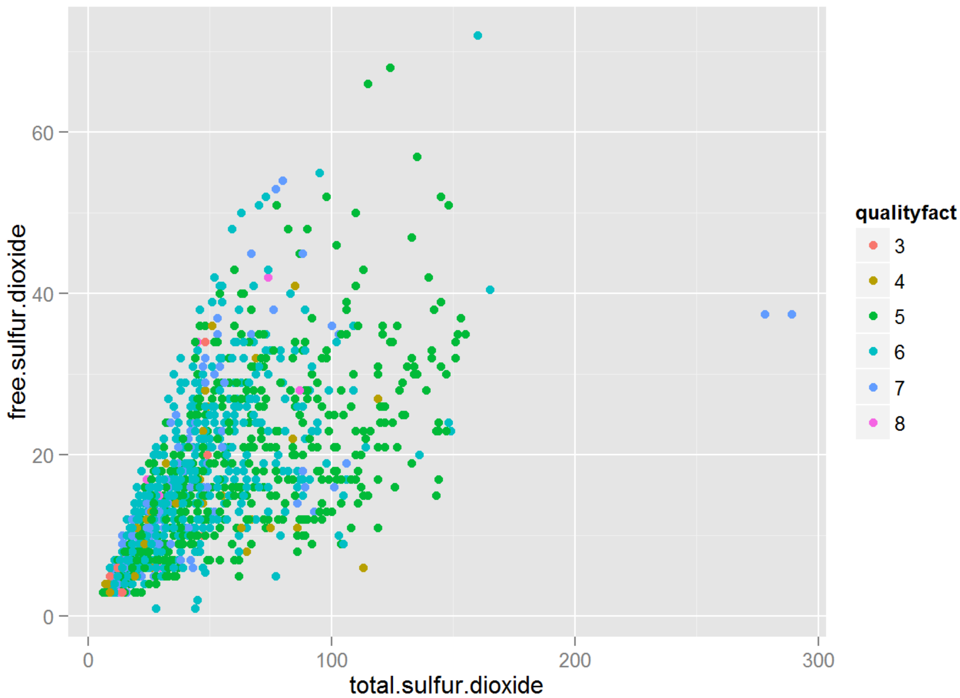
```
##
## Call:
## lm(formula = redWineData$fixed.acidity ~ redWineData$pH)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.078 -0.840 -0.155  0.682  5.030
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      33.823      0.683    49.5   <2e-16 ***
## redWineData$pH   -7.702      0.206   -37.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.27 on 1597 degrees of freedom
## Multiple R-squared:  0.466,  Adjusted R-squared:  0.466
## F-statistic: 1.4e+03 on 1 and 1597 DF,  p-value: <2e-16
```

Since the p-value is less than 0.05(assuming the alpha = 0.05), we reject the null hypothesis that, there is no dependency between fixed acidity and pH.

fixed.acidity = 33.822 + pH*(-7.702)

Above equation indicates a negative linear dependency between fixed acidity and pH. In other words value of fixed acidity can be increasingly predicted with a decreasing value of pH

```
ggplot( data = redWineData,
         aes(total.sulfur.dioxide,free.sulfur.dioxide,color = qualityfact)
       ) +
  geom_point()
```




```
linearModel4 <- lm(redWineData$free.sulfur.dioxide ~ redWineData$total.sulfur.dioxide)

p <- ggplot( data = redWineData,
        aes(x = free.sulfur.dioxide, y = total.sulfur.dioxide)
      ) +
  geom_point()

p4 <- p + geom_smooth(method = "lm", formula = y~x) + ggtitle('Linear Model for Free Sulfur dioxide VS Total Sulfur dioxide'
)



summary(linearModel4)
```

```
##
## Call:
## lm(formula = redWineData$free.sulfur.dioxide ~ redWineData$total.sulfur.dioxide)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -29.87  -4.41  -1.77   3.57  35.66
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       6.00950    0.33722    17.8   <2e-16 ***
## redWineData$total.sulfur.dioxide  0.21231    0.00592    35.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.79 on 1597 degrees of freedom
## Multiple R-squared:  0.446,  Adjusted R-squared:  0.445
```

```
## F-statistic: 1.28e+03 on 1 and 1597 DF,  p-value: <2e-16
```

Since the p-value is less than 0.05(assuming the alpha = 0.05), we reject the null hypothesis that, there is no dependency between free and total sulfur dioxide.

free.sulfur.dioxide = 6.009 + 0.212 * total.sulfur.dioxide

Above equation helps us in predicting the free sulphur dioxde given the total sulfur dioxide.

Let us just review our work so far. We started off, with uni-variate analysis then moved to bi-variate analysis with quality as one of the two variables. Here, we noticed certain ingredients having a strong impact on the quality of the redwines. This led us to a conclusion that the following variables have a strong impact on the quality of the redwines. -> Fixed acidity -> Citric acid -> Density -> pH -> Free sulfur dioxide -> Total sulfur dioxide

In order to figure out how, the impact happens in tandem, we built a correlation coefficient matrix. This led us to the understanding that certain pairs of ingredients have a strong inter-dependency between themselves. This led us further to building predictive linear model equations between these variables. These equations will help us in predicting the amounts of ingredients that we should add in order to improve the quality of the red wines.

Now lets find out the collective impact of the above mentioned ingredients on the quality of the redwines through multiple regression linear model equation

```
linearModelQuality <- lm(redWineData$quality ~ redWineData$fixed.acidity + redWineData$citric.acid + redWineData$density + r
edWineData$pH + redWineData$free.sulfur.dioxide + redWineData$total.sulfur.dioxide)

summary(linearModelQuality)
```

```
##
## Call:
## lm(formula = redWineData$quality ~ redWineData$fixed.acidity +
##     redWineData$citric.acid + redWineData$density + redWineData$pH +
##     redWineData$free.sulfur.dioxide + redWineData$total.sulfur.dioxide)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0496 -0.4823 -0.0699  0.5025  2.3566
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       1.81e+02   1.36e+01   13.30  < 2e-16 ***
## redWineData$fixed.acidity         1.51e-01   2.23e-02    6.75  2.1e-11 ***
## redWineData$citric.acid           1.03e+00   1.30e-01    7.91  4.9e-15 ***
## redWineData$density              -1.80e+02   1.39e+01  -12.93  < 2e-16 ***
## redWineData$pH                    6.90e-01   1.72e-01    4.01  6.4e-05 ***
## redWineData$free.sulfur.dioxide   1.05e-02   2.37e-03    4.41  1.1e-05 ***
## redWineData$total.sulfur.dioxide -5.13e-03   7.92e-04   -6.48  1.2e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.726 on 1592 degrees of freedom
## Multiple R-squared:  0.194,  Adjusted R-squared:  0.191
## F-statistic: 63.9 on 6 and 1592 DF,  p-value: <2e-16
```

The above linear model translates into the following equation which can help us in predicting the values that are needed to be added in order to generate the desired quality level.

Quality = (1.808e + 02) + [(1.506e-01)*fixed.acidity] + [(1.031e+00)*citric.acid] + [(-1.795e + 02) * density] + [(6.900e-01)*pH] +[(1.406e-02)*free.sulfur.dioxide] + [(-5.130e-03)*total.sulfur,dioxide]

The following factors help us in expanding our knowledge of the linear model that we have generated.

```
coefficients(linearModelQuality)
```

```
##                      (Intercept)        redWineData$fixed.acidity
##                        180.84225                          0.15058
##          redWineData$citric.acid              redWineData$density
##                          1.03062                       -179.53467
##                   redWineData$pH  redWineData$free.sulfur.dioxide
##                          0.69001                          0.01046
```

```
## redWineData$total.sulfur.dioxide
##                         -0.00513
```

Coefficents gives us the list of coefficients generated in our linear model.

```r
confint(linearModelQuality, level = 0.95)
```

```
##                                      2.5 %     97.5 %
## (Intercept)                      1.542e+02  2.075e+02
## redWineData$fixed.acidity        1.068e-01  1.944e-01
## redWineData$citric.acid          7.750e-01  1.286e+00
## redWineData$density             -2.068e+02 -1.523e+02
## redWineData$pH                   3.524e-01  1.028e+00
## redWineData$free.sulfur.dioxide   5.809e-03  1.511e-02
## redWineData$total.sulfur.dioxide -6.682e-03 -3.577e-03
```

Confint gives us the confidence interval with an error tolerance of 0.05% because we have specified the confidence level to be 95%.

```r
fitted(linearModelQuality)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## 5.179 5.182 5.191 5.981 5.179 5.170 5.337 5.765 5.397 5.169 5.222 5.169
##    13    14    15    16    17    18    19    20    21    22    23    24
## 5.509 5.443 5.065 5.056 5.836 5.416 5.217 5.567 6.052 5.400 5.426 5.213
##    25    26    27    28    29    30    31    32    33    34    35    36
## 5.447 5.531 5.630 5.426 5.241 5.462 5.211 5.385 5.056 4.969 5.332 5.039
##    37    38    39    40    41    42    43    44    45    46    47    48
## 5.389 5.601 5.727 5.150 5.150 5.564 5.444 5.582 5.431 5.781 5.337 5.888
##    49    50    51    52    53    54    55    56    57    58    59    60
## 5.558 5.282 5.519 5.485 5.481 5.423 5.572 5.238 6.040 5.074 5.384 5.620
##    61    62    63    64    65    66    67    68    69    70    71    72
## 5.710 5.396 5.454 5.222 5.464 5.464 5.400 5.415 6.064 5.537 5.451 5.362
##    73    74    75    76    77    78    79    80    81    82    83    84
## 5.384 5.435 5.742 5.865 5.865 5.363 5.242 5.119 5.509 5.716 5.608 5.317
##    85    86    87    88    89    90    91    92    93    94    95    96
## 5.736 5.548 4.927 5.620 5.437 5.201 5.361 4.927 4.942 5.620 5.795 5.702
##    97    98    99   100   101   102   103   104   105   106   107   108
## 5.306 5.445 5.446 5.449 5.574 5.782 5.449 5.409 5.523 5.409 5.627 5.333
##   109   110   111   112   113   114   115   116   117   118   119   120
## 5.592 5.437 5.410 5.046 5.072 5.782 5.410 5.589 5.490 5.399 5.430 5.258
##   121   122   123   124   125   126   127   128   129   130   131   132
## 5.107 5.430 5.139 5.257 5.411 4.992 5.594 5.586 5.668 5.514 5.526 5.549
##   133   134   135   136   137   138   139   140   141   142   143   144
## 5.549 5.564 5.494 5.359 5.353 5.572 5.276 5.275 5.359 5.353 6.097 5.398
##   145   146   147   148   149   150   151   152   153   154   155   156
## 6.097 5.467 5.387 5.094 5.517 5.775 5.939 5.667 5.431 5.431 5.296 5.290
##   157   158   159   160   161   162   163   164   165   166   167   168
## 5.296 5.290 5.347 5.316 5.398 5.280 5.486 5.094 5.087 5.532 5.199 5.515
##   169   170   171   172   173   174   175   176   177   178   179   180
## 5.531 5.234 5.350 5.431 5.431 5.569 5.515 5.401 5.515 5.731 5.290 5.424
##   181   182   183   184   185   186   187   188   189   190   191   192
## 5.424 5.484 5.241 5.283 5.266 5.841 5.527 5.301 5.043 5.049 5.131 5.427
##   193   194   195   196   197   198   199   200   201   202   203   204
## 4.796 5.508 5.508 5.275 5.553 6.132 5.690 5.730 5.978 5.338 5.496 5.694
##   205   206   207   208   209   210   211   212   213   214   215   216
## 5.685 6.264 6.264 5.401 5.304 6.180 6.363 5.423 6.263 5.783 5.161 5.289
##   217   218   219   220   221   222   223   224   225   226   227   228
## 5.516 5.570 5.623 5.231 5.768 5.433 5.586 5.396 5.560 5.721 5.774 5.410
##   229   230   231   232   233   234   235   236   237   238   239   240
## 5.721 5.472 5.540 5.447 5.348 5.472 5.375 5.252 5.252 5.257 5.252 5.375
##   241   242   243   244   245   246   247   248   249   250   251   252
## 5.354 5.933 5.210 6.111 6.111 5.148 5.252 5.330 5.426 5.148 5.856 5.267
##   253   254   255   256   257   258   259   260   261   262   263   264
## 5.857 5.507 5.267 5.149 5.671 5.122 5.789 5.656 5.635 5.277 5.536 5.651
```

```
##   265   266   267   268   269   270   271   272   273   274   275   276 
## 5.752 6.414 4.925 5.735 5.102 5.824 5.475 5.824 5.729 5.448 4.907 5.475 
##   277   278   279   280   281   282   283   284   285   286   287   288 
## 5.102 5.824 5.968 5.335 5.847 5.914 5.268 5.335 5.098 5.098 5.884 5.510 
##   289   290   291   292   293   294   295   296   297   298   299   300 
## 5.404 5.471 5.404 6.083 5.382 5.574 5.605 5.408 5.719 5.166 4.964 5.140 
##   301   302   303   304   305   306   307   308   309   310   311   312 
## 5.443 6.177 5.193 5.547 5.547 5.476 5.507 5.591 5.612 5.681 5.476 5.337 
##   313   314   315   316   317   318   319   320   321   322   323   324 
## 5.301 5.241 5.600 5.655 5.254 5.330 5.625 5.330 5.625 5.072 5.250 5.726 
##   325   326   327   328   329   330   331   332   333   334   335   336 
## 4.810 4.810 6.225 5.982 6.064 5.852 5.985 5.985 5.142 5.510 5.459 5.942 
##   337   338   339   340   341   342   343   344   345   346   347   348 
## 6.337 5.568 5.594 5.923 5.810 6.076 6.024 6.024 5.558 5.318 5.604 6.397 
##   349   350   351   352   353   354   355   356   357   358   359   360 
## 5.516 5.075 5.340 5.079 5.176 5.871 6.038 5.598 5.828 6.320 5.940 5.787 
##   361   362   363   364   365   366   367   368   369   370   371   372 
## 5.196 5.534 5.901 6.103 5.496 5.921 5.496 5.675 5.397 6.130 5.322 5.779 
##   373   374   375   376   377   378   379   380   381   382   383   384 
## 6.092 5.245 5.712 6.121 5.708 6.130 6.005 5.553 5.648 5.889 5.648 5.648 
##   385   386   387   388   389   390   391   392   393   394   395   396 
## 5.438 5.163 5.438 5.461 5.427 5.513 5.697 5.889 5.757 5.569 5.969 5.926 
##   397   398   399   400   401   402   403   404   405   406   407   408 
## 4.899 5.942 5.942 5.526 4.899 5.976 5.711 5.705 5.253 6.007 5.795 6.032 
##   409   410   411   412   413   414   415   416   417   418   419   420 
## 6.192 6.161 5.410 5.445 5.170 5.817 5.181 4.526 6.311 5.368 5.903 5.349 
##   421   422   423   424   425   426   427   428   429   430   431   432 
## 5.866 5.711 5.649 5.881 5.649 5.711 5.630 5.328 5.589 6.034 5.881 5.433 
##   433   434   435   436   437   438   439   440   441   442   443   444 
## 6.637 5.888 5.639 5.888 5.904 6.120 5.639 5.369 6.081 6.031 5.743 6.229 
##   445   446   447   448   449   450   451   452   453   454   455   456 
## 5.678 5.225 5.974 5.784 5.564 6.056 6.056 5.568 5.319 5.748 5.927 6.102 
##   457   458   459   460   461   462   463   464   465   466   467   468 
## 5.566 5.358 5.748 5.697 6.000 5.322 6.374 5.476 5.861 5.490 5.753 6.320 
##   469   470   471   472   473   474   475   476   477   478   479   480 
## 5.878 5.838 6.027 5.927 5.933 6.116 5.927 5.291 5.733 5.946 5.291 5.228 
##   481   482   483   484   485   486   487   488   489   490   491   492 
## 4.936 6.096 5.831 5.852 6.194 5.839 5.839 5.705 5.898 5.683 5.521 6.372 
##   493   494   495   496   497   498   499   500   501   502   503   504 
## 6.359 5.127 5.585 6.111 5.345 5.493 6.111 5.127 5.345 6.000 6.000 5.938 
##   505   506   507   508   509   510   511   512   513   514   515   516 
## 5.917 6.219 5.976 5.744 5.389 6.452 6.137 5.389 5.769 6.151 6.151 4.778 
##   517   518   519   520   521   522   523   524   525   526   527   528 
## 5.738 5.637 6.129 5.705 5.959 5.664 5.566 5.499 5.503 5.424 5.705 5.797 
##   529   530   531   532   533   534   535   536   537   538   539   540 
## 5.620 5.538 5.324 5.623 5.623 6.105 5.296 5.324 5.538 5.588 5.584 6.015 
##   541   542   543   544   545   546   547   548   549   550   551   552 
## 5.373 5.639 5.668 5.374 6.039 5.485 5.331 5.847 5.869 5.807 5.359 5.446 
##   553   554   555   556   557   558   559   560   561   562   563   564 
## 5.418 5.916 5.582 5.582 5.554 5.597 5.554 5.492 5.954 5.443 5.481 5.715 
##   565   566   567   568   569   570   571   572   573   574   575   576 
## 5.492 5.954 5.293 5.293 5.701 5.905 5.696 5.905 5.871 5.738 5.562 6.055 
##   577   578   579   580   581   582   583   584   585   586   587   588 
## 5.736 5.357 5.344 5.989 5.810 5.810 5.751 6.103 5.985 5.341 5.814 5.161 
##   589   590   591   592   593   594   595   596   597   598   599   600 
## 6.307 5.869 5.548 5.836 5.548 5.468 5.428 5.117 5.845 6.148 5.472 5.756 
##   601   602   603   604   605   606   607   608   609   610   611   612 
## 5.707 5.808 5.184 5.808 4.989 5.202 6.009 5.667 4.815 5.655 5.803 5.781 
##   613   614   615   616   617   618   619   620   621   622   623   624 
## 5.396 5.821 4.991 5.666 5.666 5.981 5.546 5.760 5.085 5.086 5.597 5.618 
##   625   626   627   628   629   630   631   632   633   634   635   636 
## 4.695 4.695 5.441 5.441 5.487 5.329 5.487 5.843 5.532 4.916 5.574 5.781 
##   637   638   639   640   641   642   643   644   645   646   647   648 
## 5.214 5.227 5.504 5.710 5.725 5.593 5.725 5.593 5.725 5.194 5.267 5.431 
##   649   650   651   652   653   654   655   656   657   658   659   660
```

```
## 5.832 5.201 5.486 4.785 6.725 5.855 5.514 4.975 5.486 6.122 5.189 5.274
##   661   662   663   664   665   666   667   668   669   670   671   672
## 5.189 5.377 5.230 5.853 5.636 5.390 5.462 5.703 6.038 5.703 5.662 5.527
##   673   674   675   676   677   678   679   680   681   682   683   684
## 5.226 5.527 5.755 5.613 5.755 5.405 5.391 6.018 5.899 5.665 5.601 5.604
##   685   686   687   688   689   690   691   692   693   694   695   696
## 5.301 5.604 5.405 5.155 5.532 5.695 5.517 5.026 5.445 5.164 5.163 6.168
##   697   698   699   700   701   702   703   704   705   706   707   708
## 5.235 5.235 5.096 6.003 5.580 5.235 5.291 5.524 5.318 5.070 5.587 5.531
##   709   710   711   712   713   714   715   716   717   718   719   720
## 5.564 5.781 5.516 4.959 5.046 5.477 5.534 5.401 5.477 5.229 5.258 5.142
##   721   722   723   724   725   726   727   728   729   730   731   732
## 5.258 5.180 5.234 5.068 5.727 5.525 5.031 5.203 5.203 5.676 5.864 5.233
##   733   734   735   736   737   738   739   740   741   742   743   744
## 5.394 5.319 5.202 5.486 5.486 4.823 5.103 5.158 5.909 5.116 5.178 5.512
##   745   746   747   748   749   750   751   752   753   754   755   756
## 5.379 5.420 5.439 5.502 5.402 5.420 5.257 5.257 5.223 5.257 5.915 5.818
##   757   758   759   760   761   762   763   764   765   766   767   768
## 5.657 5.333 5.333 5.262 5.107 5.562 5.440 5.562 5.166 5.160 5.201 4.892
##   769   770   771   772   773   774   775   776   777   778   779   780
## 5.075 5.352 5.075 5.156 5.166 5.343 5.370 5.493 5.386 5.538 5.464 5.028
##   781   782   783   784   785   786   787   788   789   790   791   792
## 5.122 5.342 5.146 5.342 5.066 5.561 5.561 5.318 5.318 4.878 5.455 4.958
##   793   794   795   796   797   798   799   800   801   802   803   804
## 4.988 5.454 6.354 5.456 5.455 6.252 5.557 5.557 5.079 5.396 5.559 5.216
##   805   806   807   808   809   810   811   812   813   814   815   816
## 5.569 6.399 6.324 6.399 5.310 5.509 5.543 5.878 5.851 5.939 6.074 5.851
##   817   818   819   820   821   822   823   824   825   826   827   828
## 5.425 6.507 5.286 4.930 5.355 6.076 5.398 5.398 5.452 5.482 6.014 5.482
##   829   830   831   832   833   834   835   836   837   838   839   840
## 6.134 5.899 5.508 5.899 6.050 6.249 5.790 5.514 6.398 6.398 5.964 5.276
##   841   842   843   844   845   846   847   848   849   850   851   852
## 6.093 5.409 6.127 5.287 6.259 5.511 5.511 5.340 5.511 5.478 5.930 5.930
##   853   854   855   856   857   858   859   860   861   862   863   864
## 5.067 6.276 6.276 5.757 6.276 5.894 6.305 5.947 5.052 6.246 5.615 5.078
##   865   866   867   868   869   870   871   872   873   874   875   876
## 5.052 5.064 5.933 5.962 5.947 5.676 6.063 5.812 5.829 6.104 6.115 6.013
##   877   878   879   880   881   882   883   884   885   886   887   888
## 5.591 6.063 5.392 5.063 5.810 5.826 5.939 5.063 5.392 5.353 5.319 6.259
##   889   890   891   892   893   894   895   896   897   898   899   900
## 5.618 4.705 5.821 5.046 5.705 5.046 5.043 5.616 6.199 5.616 6.199 5.179
##   901   902   903   904   905   906   907   908   909   910   911   912
## 6.056 5.467 5.467 5.451 5.451 5.054 5.381 5.653 5.813 6.297 6.166 5.509
##   913   914   915   916   917   918   919   920   921   922   923   924
## 6.151 6.082 6.297 6.148 5.641 5.340 5.755 6.080 5.741 5.755 6.080 5.340
##   925   926   927   928   929   930   931   932   933   934   935   936
## 5.947 6.238 6.011 5.181 5.947 6.248 5.269 5.227 5.617 5.227 5.269 6.013
##   937   938   939   940   941   942   943   944   945   946   947   948
## 6.013 6.053 6.269 5.956 6.497 6.438 5.630 5.487 6.212 6.193 6.161 6.366
##   949   950   951   952   953   954   955   956   957   958   959   960
## 6.266 6.266 6.266 6.366 6.083 6.525 6.074 6.050 6.259 6.103 5.730 5.319
##   961   962   963   964   965   966   967   968   969   970   971   972
## 5.905 5.491 5.256 6.094 5.905 6.080 5.967 5.094 6.405 5.351 6.173 6.173
##   973   974   975   976   977   978   979   980   981   982   983   984
## 6.203 6.180 5.953 5.610 5.610 5.116 5.948 6.366 5.680 5.758 6.260 5.680
##   985   986   987   988   989   990   991   992   993   994   995   996
## 6.366 5.606 6.333 5.618 5.557 6.183 5.557 5.571 5.581 5.571 5.356 5.400
##   997   998   999  1000  1001  1002  1003  1004  1005  1006  1007  1008
## 5.802 5.802 5.942 6.182 6.028 6.104 6.134 6.273 5.848 6.273 6.134 6.144
##  1009  1010  1011  1012  1013  1014  1015  1016  1017  1018  1019  1020
## 6.101 5.807 6.525 5.955 5.512 5.495 5.875 6.204 6.221 6.488 6.488 5.582
##  1021  1022  1023  1024  1025  1026  1027  1028  1029  1030  1031  1032
## 6.530 6.530 5.508 6.104 5.577 5.177 6.283 5.822 5.757 5.577 5.678 5.618
##  1033  1034  1035  1036  1037  1038  1039  1040  1041  1042  1043  1044
## 5.088 5.537 5.420 5.900 6.273 5.332 6.166 5.734 5.333 5.690 5.734 5.672
```

```
##  1045  1046  1047  1048  1049  1050  1051  1052  1053  1054  1055  1056 
## 6.051 5.786 5.547 5.638 5.830 5.842 5.638 5.758 6.061 6.345 5.092 5.092 
##  1057  1058  1059  1060  1061  1062  1063  1064  1065  1066  1067  1068 
## 6.132 5.102 6.154 6.132 6.140 6.365 6.018 6.492 5.944 5.547 5.885 6.358 
##  1069  1070  1071  1072  1073  1074  1075  1076  1077  1078  1079  1080 
## 6.358 5.722 6.276 4.887 5.541 5.616 4.887 5.943 6.376 5.848 5.848 5.469 
##  1081  1082  1083  1084  1085  1086  1087  1088  1089  1090  1091  1092 
## 6.556 5.413 5.591 6.099 5.591 5.502 5.788 6.312 6.190 6.190 6.471 6.043 
##  1093  1094  1095  1096  1097  1098  1099  1100  1101  1102  1103  1104 
## 5.779 6.009 5.332 5.750 5.332 5.762 6.272 5.762 6.299 6.129 5.807 6.129 
##  1105  1106  1107  1108  1109  1110  1111  1112  1113  1114  1115  1116 
## 6.314 5.906 6.317 6.195 5.184 5.831 5.522 5.915 6.157 5.725 6.636 5.584 
##  1117  1118  1119  1120  1121  1122  1123  1124  1125  1126  1127  1128 
## 5.584 5.584 6.144 5.912 6.425 5.914 6.150 6.067 5.544 6.245 6.289 5.798 
##  1129  1130  1131  1132  1133  1134  1135  1136  1137  1138  1139  1140 
## 5.499 5.719 5.286 5.790 6.301 5.369 6.066 6.231 5.961 5.961 4.963 5.244 
##  1141  1142  1143  1144  1145  1146  1147  1148  1149  1150  1151  1152 
## 5.565 5.690 5.898 5.815 5.757 6.073 5.542 6.018 6.091 6.278 6.076 5.822 
##  1153  1154  1155  1156  1157  1158  1159  1160  1161  1162  1163  1164 
## 5.577 6.004 5.797 5.577 6.285 5.662 5.841 5.881 6.200 6.115 6.231 5.730 
##  1165  1166  1167  1168  1169  1170  1171  1172  1173  1174  1175  1176 
## 5.730 5.799 5.383 6.140 5.743 5.916 5.840 5.641 6.099 5.607 5.607 5.801 
##  1177  1178  1179  1180  1181  1182  1183  1184  1185  1186  1187  1188 
## 5.555 5.919 5.642 5.733 5.733 6.361 6.181 5.297 5.206 6.005 5.427 6.005 
##  1189  1190  1191  1192  1193  1194  1195  1196  1197  1198  1199  1200 
## 5.206 5.605 6.113 5.550 6.114 5.535 5.275 5.488 5.271 5.833 5.592 5.271 
##  1201  1202  1203  1204  1205  1206  1207  1208  1209  1210  1211  1212 
## 5.833 6.169 6.186 5.335 6.074 6.074 6.074 6.106 6.074 6.115 5.572 5.477 
##  1213  1214  1215  1216  1217  1218  1219  1220  1221  1222  1223  1224 
## 5.572 5.957 6.141 6.336 5.366 6.285 5.776 6.225 6.178 6.178 5.470 6.228 
##  1225  1226  1227  1228  1229  1230  1231  1232  1233  1234  1235  1236 
## 5.953 5.037 5.331 5.567 5.865 5.463 5.954 5.545 5.463 6.108 6.110 5.069 
##  1237  1238  1239  1240  1241  1242  1243  1244  1245  1246  1247  1248 
## 5.330 6.110 5.225 5.599 5.759 5.812 6.311 5.078 5.184 5.628 5.452 5.628 
##  1249  1250  1251  1252  1253  1254  1255  1256  1257  1258  1259  1260 
## 6.001 5.667 5.667 5.483 5.418 5.634 5.587 5.830 5.844 5.371 5.522 5.522 
##  1261  1262  1263  1264  1265  1266  1267  1268  1269  1270  1271  1272 
## 5.959 5.799 5.857 5.265 6.236 5.541 5.541 6.285 5.614 6.080 6.232 6.035 
##  1273  1274  1275  1276  1277  1278  1279  1280  1281  1282  1283  1284 
## 5.751 5.820 5.850 5.255 6.022 5.307 5.255 6.271 5.714 5.714 5.460 5.457 
##  1285  1286  1287  1288  1289  1290  1291  1292  1293  1294  1295  1296 
## 5.841 6.144 6.367 6.094 4.747 4.747 5.632 5.519 5.976 5.332 5.519 5.439 
##  1297  1298  1299  1300  1301  1302  1303  1304  1305  1306  1307  1308 
## 5.439 6.071 6.022 5.814 5.710 5.511 6.011 6.124 5.500 5.478 5.508 5.640 
##  1309  1310  1311  1312  1313  1314  1315  1316  1317  1318  1319  1320 
## 5.508 5.405 5.478 6.000 5.810 5.593 5.618 5.365 6.151 6.011 5.365 5.864 
##  1321  1322  1323  1324  1325  1326  1327  1328  1329  1330  1331  1332 
## 5.408 6.091 6.398 6.118 5.850 5.850 5.850 5.850 5.453 5.360 5.360 5.576 
##  1333  1334  1335  1336  1337  1338  1339  1340  1341  1342  1343  1344 
## 5.494 5.409 5.420 5.748 5.544 5.544 5.544 5.582 5.582 5.582 5.555 5.582 
##  1345  1346  1347  1348  1349  1350  1351  1352  1353  1354  1355  1356 
## 5.913 5.688 5.599 5.480 5.480 5.574 5.848 5.740 5.219 5.219 5.439 5.653 
##  1357  1358  1359  1360  1361  1362  1363  1364  1365  1366  1367  1368 
## 5.663 5.709 5.375 6.035 5.709 5.375 6.035 5.477 5.643 5.179 5.360 4.947 
##  1369  1370  1371  1372  1373  1374  1375  1376  1377  1378  1379  1380 
## 5.504 5.684 5.677 6.150 5.677 5.483 5.591 5.275 5.778 5.623 5.479 5.507 
##  1381  1382  1383  1384  1385  1386  1387  1388  1389  1390  1391  1392 
## 5.507 5.507 5.434 5.434 5.241 5.487 5.437 5.437 5.261 5.138 5.940 5.644 
##  1393  1394  1395  1396  1397  1398  1399  1400  1401  1402  1403  1404 
## 5.658 5.640 5.209 5.417 5.448 5.329 5.517 5.523 5.260 5.260 5.862 5.643 
##  1405  1406  1407  1408  1409  1410  1411  1412  1413  1414  1415  1416 
## 5.440 5.915 5.552 5.693 6.146 5.693 5.836 5.790 5.552 5.487 5.692 5.299 
##  1417  1418  1419  1420  1421  1422  1423  1424  1425  1426  1427  1428 
## 5.692 5.950 5.504 5.382 5.504 5.509 5.616 5.341 5.621 5.621 5.655 5.786 
##  1429  1430  1431  1432  1433  1434  1435  1436  1437  1438  1439  1440 
```

```
## 5.448 5.979 5.732 5.531 5.578 5.546 4.844 4.844 5.354 5.782 5.423 5.350
##  1441  1442  1443  1444  1445  1446  1447  1448  1449  1450  1451  1452
## 5.915 5.009 5.265 6.073 5.350 5.009 5.265 5.244 5.413 5.896 5.915 5.711
##  1453  1454  1455  1456  1457  1458  1459  1460  1461  1462  1463  1464
## 5.872 5.519 5.935 5.665 5.862 5.519 6.018 6.118 5.240 5.318 5.485 5.466
##  1465  1466  1467  1468  1469  1470  1471  1472  1473  1474  1475  1476
## 5.625 5.625 5.549 5.494 5.549 5.230 5.632 5.963 6.035 5.748 5.141 5.841
##  1477  1478  1479  1480  1481  1482  1483  1484  1485  1486  1487  1488
## 5.141 5.841 5.079 5.800 5.604 5.800 5.736 6.116 5.547 5.515 5.671 5.603
##  1489  1490  1491  1492  1493  1494  1495  1496  1497  1498  1499  1500
## 5.786 5.694 6.390 5.786 5.907 5.124 5.708 5.619 5.124 5.632 5.535 5.632
##  1501  1502  1503  1504  1505  1506  1507  1508  1509  1510  1511  1512
## 5.618 5.495 5.375 5.855 6.043 5.506 5.582 6.043 5.711 6.202 5.516 5.369
##  1513  1514  1515  1516  1517  1518  1519  1520  1521  1522  1523  1524
## 5.516 5.835 5.116 5.116 5.445 5.675 5.791 5.367 5.675 5.613 5.445 5.728
##  1525  1526  1527  1528  1529  1530  1531  1532  1533  1534  1535  1536
## 5.593 5.515 5.487 5.632 5.716 5.695 5.754 5.468 5.583 5.572 5.875 5.527
##  1537  1538  1539  1540  1541  1542  1543  1544  1545  1546  1547  1548
## 5.660 5.595 5.650 5.670 5.366 5.614 5.670 6.138 6.039 5.811 5.508 6.047
##  1549  1550  1551  1552  1553  1554  1555  1556  1557  1558  1559  1560
## 5.980 6.064 5.191 5.217 5.489 5.226 5.473 5.817 5.455 5.473 5.391 5.296
##  1561  1562  1563  1564  1565  1566  1567  1568  1569  1570  1571  1572
## 5.296 5.296 5.634 5.634 5.634 5.650 6.173 5.634 5.512 5.854 6.347 5.690
##  1573  1574  1575  1576  1577  1578  1579  1580  1581  1582  1583  1584
## 5.528 5.745 5.642 6.066 6.197 5.514 5.705 5.937 5.993 5.937 5.935 5.427
##  1585  1586  1587  1588  1589  1590  1591  1592  1593  1594  1595  1596
## 6.043 6.015 6.124 5.717 5.873 5.089 6.026 5.853 5.613 5.498 5.729 5.750
##  1597  1598  1599
## 5.613 5.705 5.817
```

Fitted gives us the predicted values of all the variables with out multi-linear predictive modelling equation.

```
residuals(linearModelQuality)
```

```
##          1          2          3          4          5          6
## -0.1793783 -0.1823623 -0.1911943  0.0193231 -0.1793783 -0.1695147
##          7          8          9         10         11         12
## -0.3365372  1.2354553  1.6025854 -0.1689828 -0.2216467 -0.1689828
##         13         14         15         16         17         18
## -0.5090568 -0.4425332 -0.0649758 -0.0563359  1.1641364 -0.4163188
##         19         20         21         22         23         24
## -1.2173025  0.4325516 -0.0515315 -0.3997602 -0.4260881 -0.2131184
##         25         26         27         28         29         30
##  0.5534958 -0.5306969 -0.6301767 -0.4260881 -0.2405676  0.5377816
##         31         32         33         34         35         36
## -0.2113049  0.6153288 -0.0559949  1.0310332 -0.3319252  0.9605886
##         37         38         39         40         41         42
##  0.6105392  1.3988804 -1.7266012 -0.1497261 -0.1497261 -1.5639625
##         43         44         45         46         47         48
##  0.5556874 -0.5816621 -0.4314388 -1.7810217 -0.3365981 -0.8882958
##         49         50         51         52         53         54
## -0.5577205 -0.2818681 -0.5186426  0.5146913  0.5194227 -0.4234826
##         55         56         57         58         59         60
##  0.4275860 -0.2384487 -1.0397258 -0.0738166 -0.3839296  0.3804039
##         61         62         63         64         65         66
## -0.7095487 -0.3960866  1.5455044 -0.2216400 -0.4638300 -0.4638300
##         67         68         69         70         71         72
## -0.3995893 -0.4151924 -1.0641004  0.4632178  0.5488768 -0.3624509
##         73         74         75         76         77         78
## -0.3839147 -1.4345957 -0.7422339 -0.8649018 -0.8649018  0.6367358
##         79         80         81         82         83         84
## -0.2415094 -1.1185319 -0.5086840 -0.7164014 -0.6084099 -0.3174404
##         85         86         87         88         89         90
##  0.2640450 -0.5477701  1.0729528 -0.6196268 -0.4373582 -0.2008217
```

```
##         91         92         93         94         95         96 
## -0.3612018  1.0729528  0.0577149 -0.6196268 -1.7950634  0.2981032 
##         97         98         99        100        101        102 
## -0.3060814 -0.4445996 -0.4463862  0.5511036  0.4261724  0.2179077 
##        103        104        105        106        107        108 
##  0.5511036 -0.4086413 -0.5225691 -0.4086413 -0.6271539 -0.3331973 
##        109        110        111        112        113        114 
##  0.4075070 -0.4367386 -0.4097430 -0.0460603 -0.0723491  0.2177563 
##        115        116        117        118        119        120 
## -0.4097430  0.4108561  0.5101090  0.6007415  0.5698183  0.7416911 
##        121        122        123        124        125        126 
## -0.1068408  0.5698183 -0.1386788 -0.2573305 -0.4109301  0.0076447 
##        127        128        129        130        131        132 
## -0.5941833 -0.5860258  1.3315461 -0.5137089 -0.5258225 -0.5493044 
##        133        134        135        136        137        138 
## -0.5493044  0.4361074  0.5062270 -0.3594404 -0.3530305 -0.5717939 
##        139        140        141        142        143        144 
## -0.2761027 -0.2747280 -0.3594404 -0.3530305 -0.0970813 -0.3979324 
##        145        146        147        148        149        150 
## -0.0970813 -0.4672801 -0.3869784 -0.0935257  0.4828576  0.2251950 
##        151        152        153        154        155        156 
##  0.0611827 -1.6666264 -0.4306226 -0.4306226 -0.2956427 -0.2903152 
##        157        158        159        160        161        162 
## -0.2956427 -0.2903152 -0.3470993  0.6842631 -0.3976112 -1.2797983 
##        163        164        165        166        167        168 
##  0.5138875 -0.0940729 -0.0865115 -0.5320435 -0.1992815 -1.5153198 
##        169        170        171        172        173        174 
##  0.4688849 -0.2342959 -1.3499752  0.5693637  0.5693637  0.4310766 
##        175        176        177        178        179        180 
## -0.5147989 -0.4006949 -0.5147989  0.2685734 -0.2900149 -0.4237554 
##        181        182        183        184        185        186 
## -0.4237554 -0.4844947 -0.2405441 -0.2825108  0.7340700 -0.8411122 
##        187        188        189        190        191        192 
## -0.5265865 -0.3008416 -0.0430975 -0.0485760 -0.1307732  0.5727025 
##        193        194        195        196        197        198 
##  0.2040578 -0.5075186 -0.5075186 -0.2746373 -0.5526459 -0.1316135 
##        199        200        201        202        203        204 
##  1.3101735 -1.7303772  1.0220791 -0.3378466 -0.4961461 -0.6944086 
##        205        206        207        208        209        210 
##  0.3149171  0.7361458  0.7361458 -0.4012839 -0.3038114  0.8195804 
##        211        212        213        214        215        216 
## -0.3634673  0.5768590 -0.2632059 -0.7827667  0.8387955 -0.2892397 
##        217        218        219        220        221        222 
## -0.5159478 -0.5695967 -0.6227210 -0.2309097  0.2321841 -0.4333167 
##        223        224        225        226        227        228 
## -0.5863381  0.6044085 -1.5599182  0.2786942  0.2261832 -0.4095773 
##        229        230        231        232        233        234 
##  0.2786942 -0.4719555  1.4600167  0.5532491  0.6519841 -0.4719555 
##        235        236        237        238        239        240 
##  0.6251667  0.7479742  0.7479742  0.7426467  0.7479742  0.6251667 
##        241        242        243        244        245        246 
## -0.3539670  0.0674417  0.7900442  0.8894854  0.8894854  0.8523054 
##        247        248        249        250        251        252 
## -0.2522868 -0.3303865  0.5743439  0.8523054  0.1440696  0.7333884 
##        253        254        255        256        257        258 
## -0.8569195 -0.5072095  0.7333884 -0.1494970 -0.6708299 -0.1222560 
##        259        260        261        262        263        264 
## -0.7890526  1.3438715 -0.6349592 -1.2771485 -0.5358002 -0.6508497 
##        265        266        267        268        269        270 
## -0.7517591  0.5857059 -0.9253801  2.2654439  0.8979858  0.1762813 
##        271        272        273        274        275        276 
##  0.5247885  0.1762813 -0.7294968 -0.4478911  0.0925906  0.5247885 
##        277        278        279        280        281        282 
##  0.8979858  0.1762813  2.0324877  1.6651983  0.1529162  1.0859421 
##        283        284        285        286        287        288 
```

```
## -0.2683340  1.6651983 -0.0978584 -0.0978584  0.1156936  0.4902373
##        289        290        291        292        293        294
##  1.5963414 -0.4713735  1.5963414 -1.0830837  0.6178230  0.4263175
##        295        296        297        298        299        300
##  0.3947481 -0.4082529 -0.7193783 -0.1660697  0.0363792 -0.1402893
##        301        302        303        304        305        306
##  0.5572151 -0.1765075 -0.1925511 -0.5473559 -0.5473857  0.5235454
##        307        308        309        310        311        312
## -0.5068092  0.4088416  0.3884271  0.3188492  0.5235454  0.6628025
##        313        314        315        316        317        318
##  0.6987875 -0.2408512 -0.5997587  0.3451099 -0.2539912  0.6696853
##        319        320        321        322        323        324
##  1.3750560  0.6696853  1.3750560 -0.0715678 -0.2499167  0.2743180
##        325        326        327        328        329        330
##  1.1899410  1.1899410  0.7750897 -0.9817298 -0.0638340 -0.8522931
##        331        332        333        334        335        336
##  0.0148865  0.0148865  0.8578395 -0.5102591  1.5408024  1.0578946
##        337        338        339        340        341        342
## -0.3372714 -0.5683360  0.4057434  1.0771617  0.1903968 -0.0759691
##        343        344        345        346        347        348
## -0.0243781 -0.0243781  0.4415457 -0.3177937  1.3964971 -0.3970275
##        349        350        351        352        353        354
##  0.4835464  0.9246204  0.6597321  0.9212612 -0.1763359 -0.8711528
##        355        356        357        358        359        360
## -0.0379034  0.4021866 -0.8277382  0.6801960  1.0602300  0.2128947
##        361        362        363        364        365        366
## -0.1961231  0.4663562 -0.9007526 -1.1034459  1.5037233  0.0794129
##        367        368        369        370        371        372
##  1.5037233 -0.6746017 -0.3967117  0.8701003 -0.3221759  0.2205333
##        373        374        375        376        377        378
## -0.0917433 -0.2453552  0.2878282  0.8790673  0.2923158  0.8701003
##        379        380        381        382        383        384
## -0.0053644  0.4468512  0.3515250  0.1114219  0.3515250  0.3515250
##        385        386        387        388        389        390
## -0.4380095  0.8370218  0.5616208  0.5390855  0.5734689  1.4871377
##        391        392        393        394        395        396
##  2.3026857  0.1114219 -0.7566513 -0.5687607 -0.9688962  1.0739402
##        397        398        399        400        401        402
##  0.1009358  0.0584616  0.0584616 -0.5263098  0.1009358  0.0238191
##        403        404        405        406        407        408
##  0.2894330  0.2953365 -0.2531298 -0.0073190  0.2051471  0.9676514
##        409        410        411        412        413        414
## -0.1917388 -2.1609698  0.5904560 -0.4445735 -0.1701965  1.1825240
##        415        416        417        418        419        420
## -0.1808125  0.4738305 -0.3114733 -0.3683433  0.0968904 -0.3490220
##        421        422        423        424        425        426
##  1.1335945  1.2886884 -0.6485605  1.1188369 -0.6485605  1.2886884
##        427        428        429        430        431        432
##  0.3700564  0.6717388 -0.5887638 -0.0335460  1.1188369 -0.4331590
##        433        434        435        436        437        438
## -0.6370173 -0.8879879  0.3611822 -0.8879879  0.0958099 -0.1199015
##        439        440        441        442        443        444
##  0.3611822 -0.3687944  1.9185038 -0.0309142  1.2569696  0.7710758
##        445        446        447        448        449        450
##  1.3220090  0.7750766 -0.9738401 -0.7838326  0.4355301 -0.0555532
##        451        452        453        454        455        456
## -0.0555532  0.4321531  0.6807446  1.2523618 -0.9265778  1.8978377
##        457        458        459        460        461        462
## -0.5657677 -0.3578190  1.2523618 -2.6968584  0.0003594 -0.3219756
##        463        464        465        466        467        468
## -1.3739083 -0.4763134  0.1392902 -0.4903074  0.2470398 -0.3196648
##        469        470        471        472        473        474
##  0.1223691 -0.8382419 -1.0270851  0.0727023  0.0674614 -1.1156813
##        475        476        477        478        479        480
##  0.0730968 -0.2911914 -0.7334657  0.0541098 -0.2911914  0.7715216
```

```
##       481       482       483       484       485       486
##  0.0635657  1.9044595 -0.8311439 -0.8519073 -0.1941772 -0.8391555
##       487       488       489       490       491       492
## -0.8391555  0.2945521  1.1016512  0.3167215  0.4788911  0.6281624
##       493       494       495       496       497       498
##  0.6407246  0.8725718  0.4152276  1.8885836  0.6548054 -0.4927873
##       499       500       501       502       503       504
##  1.8885836  0.8725718  0.6548054  0.9996874  0.9996874  1.0622297
##       505       506       507       508       509       510
##  1.0828009  0.7811069  1.0239038  0.2557336  0.6113624  0.5478693
##       511       512       513       514       515       516
## -1.1366498  0.6113624  0.2312503  0.8490983  0.8490983  0.2221527
##       517       518       519       520       521       522
##  0.2617416 -2.6369425 -0.1292764 -0.7047292  0.0411914 -0.6640080
##       523       524       525       526       527       528
## -0.5658101 -0.4987145 -0.5029770 -0.4238504 -0.7047292  0.2029500
##       529       530       531       532       533       534
##  0.3801159 -0.5380213  0.6755298 -0.6231021 -0.6231021 -0.1052306
##       535       536       537       538       539       540
##  0.7042791  0.6755298 -0.5380213  0.4117708  1.4155483 -1.0154223
##       541       542       543       544       545       546
## -0.3731564  0.3614849 -0.6682318  0.6260664 -0.0390812 -0.4848516
##       547       548       549       550       551       552
##  0.6690630  0.1534579  0.1313182  0.1925576  0.6405416  0.5538851
##       553       554       555       556       557       558
##  0.5819985 -0.9155596 -0.5824045 -0.5824045  0.4463415 -0.5974621
##       559       560       561       562       563       564
##  0.4463415  0.5082625 -0.9540797 -0.4426780 -0.4806469  0.2848020
##       565       566       567       568       569       570
##  0.5082625 -0.9540797  0.7073005  0.7073005  0.2986067  0.0954337
##       571       572       573       574       575       576
##  0.3040395  0.0954337 -0.8710125 -1.7375573  0.4375159 -0.0552179
##       577       578       579       580       581       582
## -1.7361737 -0.3566886 -0.3444610  0.0107483 -0.8103701 -0.8103701
##       583       584       585       586       587       588
## -0.7508014  0.8965366  1.0151719  0.6585558  1.1856168 -0.1610377
##       589       590       591       592       593       594
##  1.6930109  1.1305226 -0.5475766  0.1642152 -0.5475766 -0.4679044
##       595       596       597       598       599       600
## -0.4275666 -0.1170868  0.1546731 -0.1483516  0.5281159  0.2441131
##       601       602       603       604       605       606
## -1.7074567  0.1916296 -0.1838167  0.1916296  1.0108354  0.7978347
##       607       608       609       610       611       612
##  0.9909483  0.3325531  1.1846241  0.3447815 -0.8025691 -0.7807508
##       613       614       615       616       617       618
##  0.6044223 -0.8214182  1.0085019 -0.6659147 -0.6659147  0.0185117
##       619       620       621       622       623       624
## -0.5460966 -0.7598894 -0.0847223 -0.0863814 -0.5965683  0.3815196
##       625       626       627       628       629       630
##  0.3051240  0.3051240 -0.4407489 -0.4407489  0.5130469 -0.3285127
##       631       632       633       634       635       636
##  0.5130469 -0.8429061  0.4675684 -0.9162170 -0.5740273 -0.7814819
##       637       638       639       640       641       642
## -0.2139986 -0.2273048  1.4962303  0.2896977 -0.7249096 -0.5933755
##       643       644       645       646       647       648
## -0.7249096 -0.5933755 -0.7249096  1.8064126 -0.2672004 -1.4314396
##       649       650       651       652       653       654
##  1.1682194  0.7994962 -0.4856756  0.2148115 -1.7246146  0.1449969
##       655       656       657       658       659       660
## -0.5136221  0.0245332 -0.4856756  0.8784686  0.8108564 -1.2744240
##       661       662       663       664       665       666
##  0.8108564 -0.3767168  0.7696876  0.1474920 -0.6360641 -0.3902393
##       667       668       669       670       671       672
##  0.5376429  0.2972578 -1.0382988  0.2972578 -0.6623119 -0.5268804
##       673       674       675       676       677       678
```

```
## -0.2262923 -0.5268804  0.2451577 -0.6130376  0.2451577 -0.4054298
##        679        680        681        682        683        684
## -0.3911641 -1.0178036 -0.8992949  0.3350471 -0.6008846 -0.6035706
##        685        686        687        688        689        690
## -0.3009230 -0.6035706 -0.4051798 -0.1550247 -0.5321714 -0.6949441
##        691        692        693        694        695        696
## -2.5174712 -0.0258020 -0.4451822 -0.1644376 -0.1632139 -0.1680197
##        697        698        699        700        701        702
##  0.7653243  0.7653243 -0.0964460 -0.0033251  0.4201798  0.7653243
##        703        704        705        706        707        708
##  0.7087009 -1.5242973 -1.3182669 -0.0699113 -0.5871598 -0.5309327
##        709        710        711        712        713        714
##  0.4359860  0.2192309 -0.5162991  0.0409451 -0.0463219 -0.4770793
##        715        716        717        718        719        720
## -0.5337801  0.5986257 -0.4770793 -0.2289053 -0.2580823 -0.1419286
##        721        722        723        724        725        726
## -0.2580823 -0.1795105 -0.2344880 -0.0678510 -1.7273767 -0.5245903
##        727        728        729        730        731        732
##  0.9686633 -0.2033230 -0.2033230  0.3243785 -0.8644477 -0.2330602
##        733        734        735        736        737        738
## -0.3943672 -0.3186099 -0.2018306 -0.4859683 -0.4859683  1.1773980
##        739        740        741        742        743        744
## -0.1033674 -0.1575966  0.0914381 -0.1155565 -0.1782397 -0.5120034
##        745        746        747        748        749        750
## -0.3787964  0.5804888  0.5609529 -0.5017019  0.5983680  0.5804888
##        751        752        753        754        755        756
## -0.2568290 -0.2568290 -0.2234903 -0.2568290  0.0851216  0.1820908
##        757        758        759        760        761        762
##  0.3426493 -0.3329340 -0.3329340 -0.2616687 -0.1065570 -0.5615635
##        763        764        765        766        767        768
##  0.5604108 -0.5615635  0.8344696  0.8397795 -0.2010682  0.1082158
##        769        770        771        772        773        774
##  0.9247856 -0.3516809  0.9247856 -0.1561431 -0.1658864  0.6574578
##        775        776        777        778        779        780
##  0.6295379 -0.4928645  0.6140117  0.4616087 -0.4643894 -0.0281124
##        781        782        783        784        785        786
##  0.8781075 -0.3415205 -0.1461075 -0.3415205 -0.0656557 -0.5610553
##        787        788        789        790        791        792
## -0.5610553  0.6823269  0.6823269  0.1218853  0.5454165  0.0420822
##        793        794        795        796        797        798
##  1.0121744 -0.4538879 -0.3541150 -0.4557467 -0.4548106  0.7478044
##        799        800        801        802        803        804
##  0.4425919  0.4425919 -0.0791258 -0.3962462  1.4414195  0.7838122
##        805        806        807        808        809        810
##  0.4309937  0.6009768  0.6764286  0.6009768 -0.3102389  0.4906597
##        811        812        813        814        815        816
## -0.5432586  0.1224673 -0.8505130 -1.9393581 -0.0741972 -0.8505130
##        817        818        819        820        821        822
##  0.5751970 -0.5069677 -0.2860006  0.0697950 -0.3548777  0.9239256
##        823        824        825        826        827        828
## -0.3982927 -0.3982927 -0.4524672 -0.4815040  0.9861408 -0.4815040
##        829        830        831        832        833        834
##  1.8663358  0.1006342 -1.5080875  0.1006342 -3.0496087 -2.2492571
##        835        836        837        838        839        840
## -0.7901563 -0.5137647  0.6016146  0.6016146  1.0358904 -0.2760279
##        841        842        843        844        845        846
##  0.9073197 -0.4085746 -0.1271707 -0.2870862 -0.2591372 -0.5105692
##        847        848        849        850        851        852
## -0.5105692  0.6598607 -0.5105692 -0.4776251 -0.9301498 -0.9301498
##        853        854        855        856        857        858
## -0.0671643 -0.2756119 -0.2756119  1.2432841 -0.2756119  1.1061167
##        859        860        861        862        863        864
##  0.6953904  0.0529991 -0.0523578 -0.2464234 -0.6151616 -0.0784019
##        865        866        867        868        869        870
```

```
## -0.0523578 -0.0644008  0.0670134  0.0382903  0.0529991  0.3240803
##        871        872        873        874        875        876
## -0.0631302 -0.8118822 -1.8286758  0.8963509  0.8848256  0.9870368
##        877        878        879        880        881        882
## -1.5912447 -0.0631302  0.6075189 -0.0633236 -0.8100028  0.1741989
##        883        884        885        886        887        888
##  0.0607955 -0.0633236  0.6075189 -0.3528370  0.6813599  0.7406554
##        889        890        891        892        893        894
##  0.3816021  0.2951848 -0.8209140 -0.0459531  0.2947362 -0.0459531
##        895        896        897        898        899        900
##  0.9574397  0.3835115  0.8013512  0.3835115  0.8013512 -2.1794267
##        901        902        903        904        905        906
## -1.0556445  1.5328483  1.5328483  1.5488017  1.5488017 -0.0537561
##        907        908        909        910        911        912
## -0.3808290  0.3466110  0.1874002 -0.2967232 -0.1657475  0.4909138
##        913        914        915        916        917        918
## -0.1506251  0.9183886 -0.2967232 -0.1478406 -0.6406349  0.6597412
##        919        920        921        922        923        924
##  0.2450347 -0.0802318 -0.7408696  0.2450347 -0.0802318  0.6597412
##        925        926        927        928        929        930
## -0.9467547  0.7618998 -0.0109513 -1.1805481 -0.9467547  0.7515037
##        931        932        933        934        935        936
## -0.2692621 -0.2268310  0.3832732 -0.2268310 -0.2692621 -0.0132467
##        937        938        939        940        941        942
## -0.0132467 -2.0527377  0.7307281 -0.9563938  0.5033132  0.5622683
##        943        944        945        946        947        948
##  1.3702889  1.5128231  0.7881183  0.8072609  0.8389174  0.6344734
##        949        950        951        952        953        954
##  0.7344124  0.7344124  0.7344124  0.6344734  0.9170250  0.4746139
##        955        956        957        958        959        960
## -0.0736100 -1.0499298 -0.2592022 -0.1027840  1.2703978 -0.3192607
##        961        962        963        964        965        966
##  0.0951617 -0.4906602 -0.2560895 -0.0938562  0.0951617 -0.0797192
##        967        968        969        970        971        972
##  1.0325429 -0.0935565 -0.4046083 -0.3512403 -0.1729682 -0.1729682
##        973        974        975        976        977        978
##  0.7974595 -1.1801367  1.0472370 -0.6098208 -0.6098208 -0.1158507
##        979        980        981        982        983        984
##  1.0515171 -1.3664991  0.3199534 -0.7579231 -0.2603521  0.3199534
##        985        986        987        988        989        990
## -1.3664991  0.3944834  0.6666995 -0.6175899 -0.5566307 -0.1828626
##        991        992        993        994        995        996
## -0.5566307 -0.5705129  0.4188310 -0.5705129 -0.3556070  0.5996492
##        997        998        999       1000       1001       1002
##  1.1976037  1.1976037  0.0577910 -0.1817492  0.9724392  0.8958735
##       1003       1004       1005       1006       1007       1008
##  0.8662466  0.7268276 -0.8479813  0.7268276  0.8662466  0.8561383
##       1009       1010       1011       1012       1013       1014
##  0.8992337 -0.8072310  0.4745401  0.0454439 -0.5121813  0.5045220
##       1015       1016       1017       1018       1019       1020
##  0.1253537 -0.2037777  0.7790451 -0.4877546 -0.4877546 -0.5817333
##       1021       1022       1023       1024       1025       1026
## -0.5304031 -0.5304031 -0.5084306 -0.1036054  1.4230993  0.8229599
##       1027       1028       1029       1030       1031       1032
## -0.2834397 -0.8221692  0.2433041  1.4230993  1.3219756  1.3816635
##       1033       1034       1035       1036       1037       1038
## -0.0882748  0.4632125  0.5795261  1.0996486  0.7272442 -0.3321403
##       1039       1040       1041       1042       1043       1044
##  0.8344215  0.2656147 -0.3333127  0.3095266  0.2656147  1.3279401
##       1045       1046       1047       1048       1049       1050
## -0.0509359  0.2139210  0.4532581 -0.6378416  0.1701335  0.1577416
##       1051       1052       1053       1054       1055       1056
## -0.6378416 -0.7577687 -1.0613672  0.6554911  0.9081699  0.9081699
##       1057       1058       1059       1060       1061       1062
##  0.8677810 -0.1020741  0.8455042  0.8677810 -0.1398017  1.6349104
```

```
##       1063       1064       1065       1066       1067       1068 
## -0.0184476 -0.4917091  0.0558230  0.4529086  1.1149468  0.6424794 
##       1069       1070       1071       1072       1073       1074 
##  0.6424794 -0.7215045  0.7238952  0.1132562  0.4594007  0.3835562 
##       1075       1076       1077       1078       1079       1080 
##  0.1132562  1.0565173 -0.3759309 -0.8476550 -0.8476550  1.5309996 
##       1081       1082       1083       1084       1085       1086 
## -0.5561685  1.5874258  0.4086186 -0.0987051  0.4086186 -0.5023785 
##       1087       1088       1089       1090       1091       1092 
##  1.2124532 -0.3116745  0.8098591  0.8098591  1.5286387 -0.0428087 
##       1093       1094       1095       1096       1097       1098 
##  0.2210216  0.9911421  0.6680210 -0.7504101  0.6680210 -0.7618026 
##       1099       1100       1101       1102       1103       1104 
##  0.7276746 -0.7618026 -0.2986317 -0.1285578  0.1926564 -0.1285578 
##       1105       1106       1107       1108       1109       1110 
## -0.3139304 -0.9061960 -0.3173057  0.8054472 -0.1842136  0.1687469 
##       1111       1112       1113       1114       1115       1116 
##  0.4784684  1.0846799 -0.1574150  0.2752473 -0.6362314  0.4164640 
##       1117       1118       1119       1120       1121       1122 
##  0.4164640  0.4164640 -0.1435884 -0.9116368  1.5745301  0.0856255 
##       1123       1124       1125       1126       1127       1128 
## -0.1501823 -0.0672040 -1.5441880  0.7545763 -0.2890079  0.2017980 
##       1129       1130       1131       1132       1133       1134 
## -0.4987986  0.2811615  0.7140227 -0.7899335  0.6993254  1.6312246 
##       1135       1136       1137       1138       1139       1140 
##  0.9343871 -0.2306540  0.0390078  0.0390078  0.0365975  0.7558448 
##       1141       1142       1143       1144       1145       1146 
##  0.4346904  0.3097451  0.1020601  0.1846674 -0.7571733 -0.0726426 
##       1147       1148       1149       1150       1151       1152 
##  0.4578574  0.9820033 -0.0914563 -0.2778538  0.9244414  0.1778034 
##       1153       1154       1155       1156       1157       1158 
## -0.5771517 -0.0040495  0.2031098 -0.5771517  0.7154225  1.3378516 
##       1159       1160       1161       1162       1163       1164 
##  0.1590927 -0.8806239  0.8003829 -0.1146252  0.7688398 -0.7300658 
##       1165       1166       1167       1168       1169       1170 
## -0.7300658 -0.7987469 -0.3827442  0.8604798  0.2565881  0.0837729 
##       1171       1172       1173       1174       1175       1176 
##  0.1600935  0.3591349 -0.0989673  0.3927748  0.3927748  0.1993770 
##       1177       1178       1179       1180       1181       1182 
## -1.5553114  1.0810500 -0.6418606  0.2673870  0.2673870 -1.3605582 
##       1183       1184       1185       1186       1187       1188 
## -0.1813404 -0.2965904 -0.2059677 -0.0053771 -0.4266041 -0.0053771 
##       1189       1190       1191       1192       1193       1194 
## -0.2059677 -1.6054529 -0.1134419 -0.5500607  0.8862029 -0.5350031 
##       1195       1196       1197       1198       1199       1200 
##  0.7246359  0.5120424  0.7288252  0.1673800  0.4080662  0.7288252 
##       1201       1202       1203       1204       1205       1206 
##  0.1673800  0.8314248  1.8138525 -0.3350021  0.9262519  0.9262519 
##       1207       1208       1209       1210       1211       1212 
##  0.9262519 -1.1061374  0.9262519  0.8849186  0.4283773 -0.4769960 
##       1213       1214       1215       1216       1217       1218 
##  0.4283773  0.0426808 -0.1410210 -0.3357714  0.6340155 -0.2852781 
##       1219       1220       1221       1222       1223       1224 
##  0.2238844 -0.2251656 -0.1776479 -0.1776479  0.5299325 -0.2278044 
##       1225       1226       1227       1228       1229       1230 
##  0.0471298 -0.0367403 -0.3305337 -0.5674157  1.1349467 -0.4632915 
##       1231       1232       1233       1234       1235       1236 
##  0.0463658 -0.5453292 -0.4632915 -2.1082584 -0.1096813 -1.0693691 
##       1237       1238       1239       1240       1241       1242 
##  0.6702813 -0.1096813 -1.2245182 -1.5985151 -0.7591831 -0.8115150 
##       1243       1244       1245       1246       1247       1248 
## -0.3108988 -0.0780647  0.8155570 -0.6276425 -0.4515373 -0.6276425 
##       1249       1250       1251       1252       1253       1254 
## -0.0006131  0.3332729  0.3332729 -0.4832660 -0.4177999 -0.6337208 
##       1255       1256       1257       1258       1259       1260 
```

```
## -0.5874227 -0.8295789 -0.8440191  0.6285465  0.4778361  0.4778361
##       1261       1262       1263       1264       1265       1266
## -0.9593014 -1.7987885 -0.8573083 -1.2651898 -0.2364550  0.4590860
##       1267       1268       1269       1270       1271       1272
##  0.4590860 -0.2845345  0.3855725  1.9200560 -0.2324505 -0.0347956
##       1273       1274       1275       1276       1277       1278
## -0.7508792 -0.8203433  0.1498119  0.7448505 -2.0222157  0.6932699
##       1279       1280       1281       1282       1283       1284
##  0.7448505  0.7294128  0.2861173  0.2861173  0.5398429  0.5430108
##       1285       1286       1287       1288       1289       1290
## -0.8408901 -1.1435475 -0.3669042 -1.0938152  0.2528853  0.2528853
##       1291       1292       1293       1294       1295       1296
## -0.6315534  0.4809321  0.0240008 -1.3315981  0.4809321 -0.4387283
##       1297       1298       1299       1300       1301       1302
## -0.4387283 -0.0710982 -0.0215310 -2.8138772  0.2897443  0.4888956
##       1303       1304       1305       1306       1307       1308
## -0.0107679 -1.1239600 -0.4999124 -0.4780544 -0.5083850 -1.6398726
##       1309       1310       1311       1312       1313       1314
## -0.5083850 -0.4052243 -0.4780544 -0.0003347 -0.8095621  0.4066616
##       1315       1316       1317       1318       1319       1320
##  0.3817608  0.6350106 -0.1511312 -0.0107750  0.6350106  0.1356129
##       1321       1322       1323       1324       1325       1326
## -0.4083419 -0.0909005 -1.3982692  0.8818811  0.1496741  0.1496741
##       1327       1328       1329       1330       1331       1332
##  0.1496741  0.1496741 -0.4532509  0.6396742  0.6396742 -0.5760109
##       1333       1334       1335       1336       1337       1338
##  0.5056328 -0.4089399 -0.4200431  0.2517212 -0.5439905 -0.5439905
##       1339       1340       1341       1342       1343       1344
## -0.5439905  0.4176760  0.4176760  0.4176760  0.4454521  0.4176760
##       1345       1346       1347       1348       1349       1350
## -0.9128134  0.3120712 -0.5987813 -0.4804053 -0.4804053 -0.5741341
##       1351       1352       1353       1354       1355       1356
## -0.8476363  0.2601451 -0.2187812 -0.2187812 -0.4393424 -0.6526270
##       1357       1358       1359       1360       1361       1362
## -0.6626885  0.2912081 -0.3753229 -0.0353811 -0.7085093 -0.3745427
##       1363       1364       1365       1366       1367       1368
## -0.0353811 -1.4772329  0.3569964 -0.1788408 -0.3598344  1.0525039
##       1369       1370       1371       1372       1373       1374
##  0.4963581 -1.6842989 -0.6770150 -0.1502140 -0.6770150 -0.4831596
##       1375       1376       1377       1378       1379       1380
## -2.5906267 -0.2753748 -0.7780300  0.3767912  0.5206217  0.4932206
##       1381       1382       1383       1384       1385       1386
##  0.4932206 -0.5067083 -0.4335756 -0.4335756 -0.2413561 -0.4865570
##       1387       1388       1389       1390       1391       1392
## -0.4368829 -0.4368829 -0.2605444 -0.1377374  0.0600407 -0.6438634
##       1393       1394       1395       1396       1397       1398
## -0.6578729 -0.6395108 -0.2085279  0.5828380 -0.4478066 -0.3291336
##       1399       1400       1401       1402       1403       1404
##  1.4831442  0.4770223 -0.2601336 -0.2601336  0.1376351  2.3566085
##       1405       1406       1407       1408       1409       1410
##  0.5603087  1.0849753  0.4478553  0.3073661  0.8535920  0.3073661
##       1411       1412       1413       1414       1415       1416
##  0.1643469  0.2095755  0.4478553 -0.4867972 -0.6915893 -0.2993314
##       1417       1418       1419       1420       1421       1422
## -0.6915893  1.0500041 -0.5043958 -0.3821194 -0.5043958 -0.5088721
##       1423       1424       1425       1426       1427       1428
##  0.3835987 -1.3409523  0.3794083  0.3794083  0.3453318 -0.7858583
##       1429       1430       1431       1432       1433       1434
## -0.4484546 -0.9789397 -0.7316161  0.4687203  0.4224446  1.4544769
##       1435       1436       1437       1438       1439       1440
##  1.1556337  1.1556337 -0.3535862 -0.7823474 -0.4230225  0.6503412
##       1441       1442       1443       1444       1445       1446
##  1.0849782  0.9906606 -0.2646037 -1.0729652  0.6503412  0.9906606
##       1447       1448       1449       1450       1451       1452
## -0.2646037 -0.2443444 -0.4133091  2.1040047  1.0849782  1.2891119
```

```
##       1453       1454       1455       1456       1457       1458 
##  1.1275380 -0.5191544  0.0645558  0.3348075  0.1376127 -0.5191544 
##       1459       1460       1461       1462       1463       1464 
## -1.0176207  0.8819828  0.7602787 -1.3175842  0.5148837  0.5339830 
##       1465       1466       1467       1468       1469       1470 
## -0.6252143 -0.6252143  1.4508752 -1.4942268  1.4508752 -2.2296702 
##       1471       1472       1473       1474       1475       1476 
## -0.6321240 -0.9629101 -0.0349755 -0.7484558 -0.1408675  1.1589926 
##       1477       1478       1479       1480       1481       1482 
## -0.1408675  1.1589926 -2.0785009 -0.8003310 -1.6044850 -0.8003310 
##       1483       1484       1485       1486       1487       1488 
## -1.7356946 -1.1157111 -1.5469765 -0.5146306 -0.6713836 -0.6031702 
##       1489       1490       1491       1492       1493       1494 
## -0.7857716  0.3057158 -0.3898807 -0.7857716 -0.9073023 -0.1243737 
##       1495       1496       1497       1498       1499       1500 
##  1.2916307  0.3809751 -0.1243737  0.3675528  0.4646854  0.3675528 
##       1501       1502       1503       1504       1505       1506 
## -0.6175843 -0.4954243 -0.3752265  0.1451127 -0.0426814 -2.5059163 
##       1507       1508       1509       1510       1511       1512 
##  0.4177441 -0.0426814  0.2892572 -1.2024759  0.4836802 -0.3687780 
##       1513       1514       1515       1516       1517       1518 
##  0.4842627  0.1645455  0.8837252  0.8837252 -0.4454725  0.3246262 
##       1519       1520       1521       1522       1523       1524 
## -0.7911279 -0.3668097  0.3246262 -1.6130166 -0.4454725 -0.7279502 
##       1525       1526       1527       1528       1529       1530 
##  0.4068700 -0.5147615  0.5131951  0.3675848  0.2835096  0.3048074 
##       1531       1532       1533       1534       1535       1536 
##  0.2460318 -0.4679903  0.4173668 -0.5723332  1.1247171  0.4725587 
##       1537       1538       1539       1540       1541       1542 
##  0.3398764  0.4052994 -0.6501824 -0.6701973  0.6335281  1.3862616 
##       1543       1544       1545       1546       1547       1548 
##  0.3296375 -0.1377828  0.9611512  0.1894333 -0.5080849 -1.0468949 
##       1549       1550       1551       1552       1553       1554 
## -0.9801711  1.9355793 -0.1911991 -0.2165567  0.5106447 -0.2262281 
##       1555       1556       1557       1558       1559       1560 
##  0.5268273  1.1832605 -0.4547442  0.5268273 -0.3911666 -0.2957979 
##       1561       1562       1563       1564       1565       1566 
## -0.2957979 -0.2957979 -0.6338245 -0.6338245 -0.6338245  0.3499477 
##       1567       1568       1569       1570       1571       1572 
## -0.1733469 -0.6338245 -0.5122869  0.1464857 -0.3469079  0.3096547 
##       1573       1574       1575       1576       1577       1578 
## -0.5275241  0.2553071  0.3576926 -0.0658289 -0.1967520  0.4855340 
##       1579       1580       1581       1582       1583       1584 
##  0.2948632 -0.9369861  0.0073392 -0.9369861 -0.9353679 -0.4273263 
##       1585       1586       1587       1588       1589       1590 
##  0.9569898 -0.0145155 -0.1238916  0.2833798  0.1274403 -0.0885864 
##       1591       1592       1593       1594       1595       1596 
## -0.0255944  0.1466535  0.3870837  0.5017658 -0.7286901  0.2497746 
##       1597       1598       1599 
##  0.3870837 -0.7052083  0.1829519
```

The residuals function gives us the amount of deviation from the linear model generated.

Let us build some additional density plots in order to gain more information on our dataset

Cutting a variable

Let us cut the variable quality in order to distribute the samples over buckets of variable quality

```
redWineData$quality_bucket <- cut(redWineData$quality,c(1,3,6,8,10),
                                  labels = c('1-3','3-6','6-8','8-10')
                                  )
```

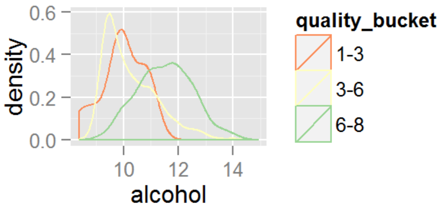
Density plots

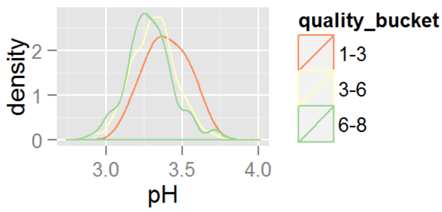
Lets create density plots for a few of the variables

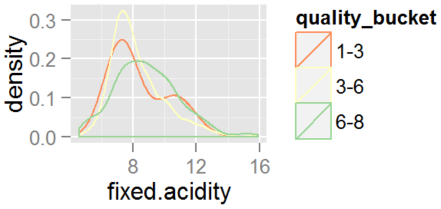
```
library(gridExtra)
```

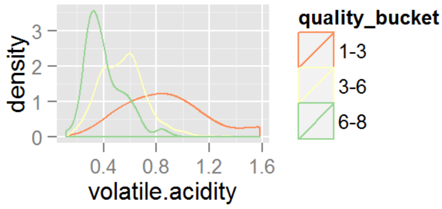
```
## Loading required package: grid
```

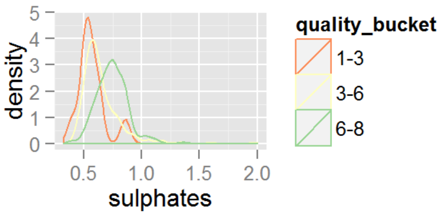
```
p1 <- ggplot( data = redWineData,
              aes(alcohol,color = quality_bucket)
            ) +
  geom_density() +
  scale_color_brewer(palette = "Spectral")

p2 <- ggplot( data = redWineData,
              aes(pH,color = quality_bucket)
            ) +
  geom_density() +
  scale_color_brewer(palette = "Spectral")

p3 <- ggplot( data = redWineData,
              aes(fixed.acidity,color = quality_bucket)
            ) +
  geom_density() +
  scale_color_brewer(palette = "Spectral")

p4 <- ggplot( data = redWineData,
              aes(volatile.acidity,color = quality_bucket)
            ) +
  geom_density() +
  scale_color_brewer(palette = "Spectral")

p5 <- ggplot( data = redWineData,
              aes(sulphates,color = quality_bucket)
            ) +
  geom_density() +
  scale_color_brewer(palette = "Spectral")

p6 <- ggplot( data = redWineData,
              aes(residual.sugar,color = quality_bucket)
            ) +
  geom_density() +
  scale_color_brewer(palette = "Spectral")

grid.arrange(p1,p2,p3,p4,p5,p6,ncol = 2)
```

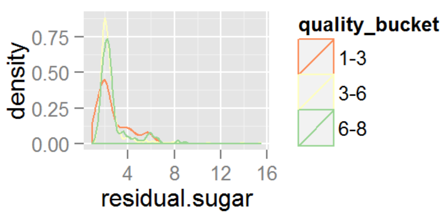












The above plots show the distribution of various ingredients over their respective quality buckets.

Sampling

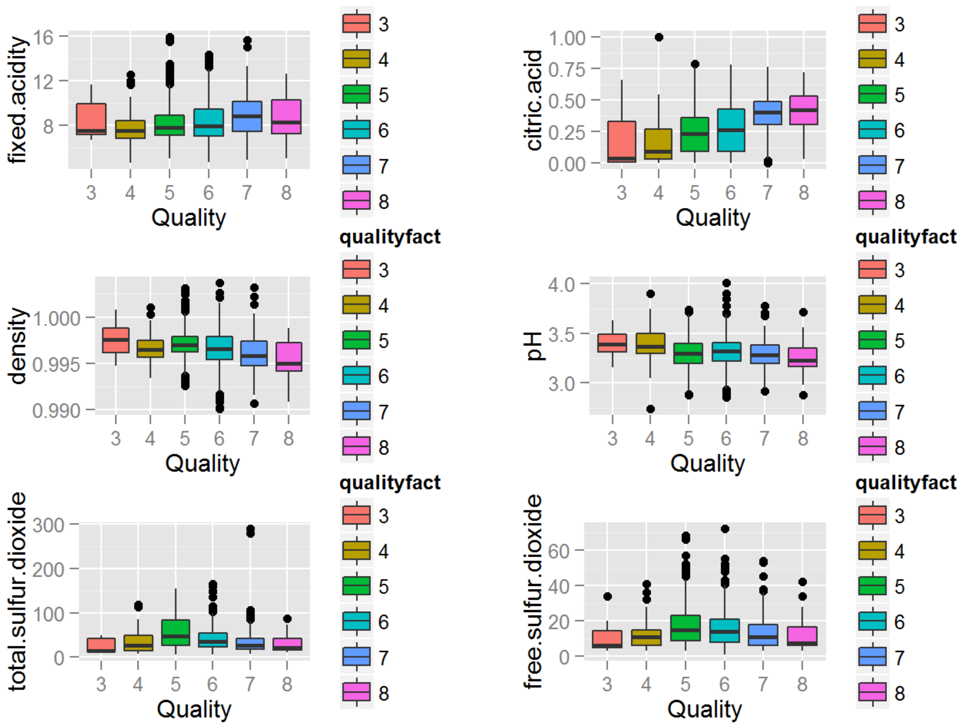
Sampling is'nt going to help us much here, because we don't really have any trends to analyze in this data set. It's just a dataset of 1599 unique wines.

Final Plots and Summary

Plot 1

```
library(gridExtra)

p1 <- ggplot( data = redWineData,
        aes(qualityfact, fixed.acidity, fill = qualityfact),
      ) +
  geom_boxplot() +
  xlab("Quality")



p2 <- ggplot( data = redWineData,
        aes(qualityfact, citric.acid, fill = qualityfact),
      ) +
  geom_boxplot() +
  xlab("Quality")

p3 <- ggplot( data = redWineData,
        aes(qualityfact,density, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality")

p4 <- ggplot( data = redWineData,
        aes(qualityfact,pH, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality")

p5 <- ggplot( data = redWineData,
        aes(qualityfact,total.sulfur.dioxide, fill = qualityfact)
      ) +
```

```
  geom_boxplot() +
  xlab("Quality")

p6 <- ggplot( data = redWineData,
        aes(qualityfact,free.sulfur.dioxide, fill = qualityfact)
      ) +
  geom_boxplot() +
  xlab("Quality")

grid.arrange(p1,p2,p3,p4,p5,p6,ncol=2)
```

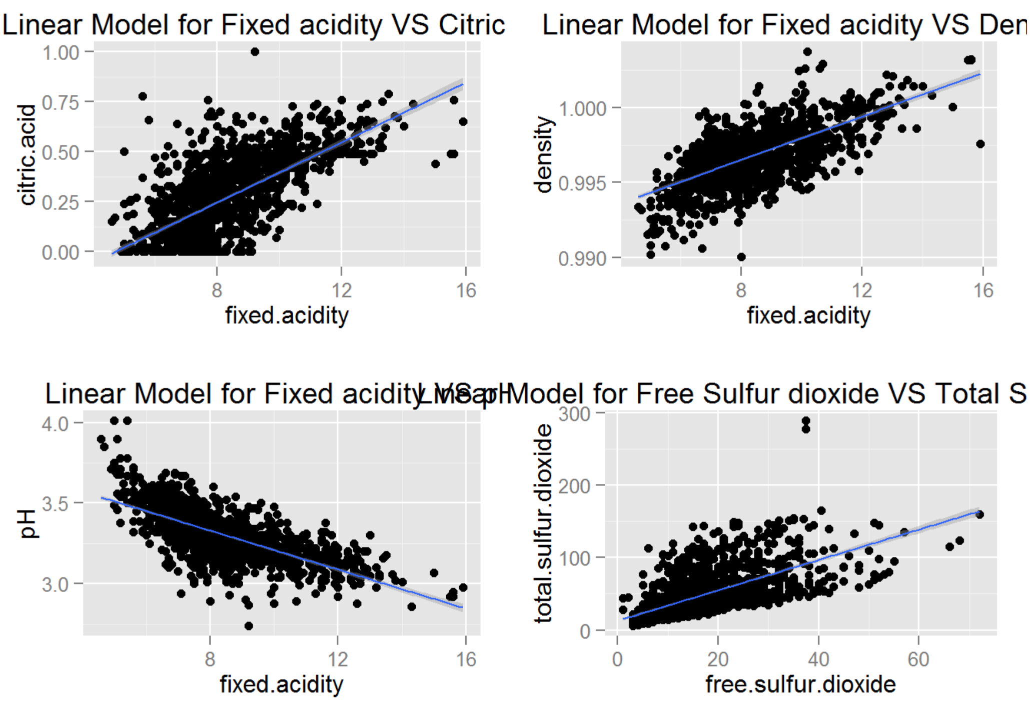


The above box-plots indicate the ingredients which are prominently influencing the quality of the redwines. These results have been derived from the bi-variate analysis of quality vs the other factors.

Plot 2

```
library(gridExtra)

p <- ggplot( data = redWineData,
        aes(x = fixed.acidity, y = citric.acid)
      ) +
  geom_point()

p1 <- p + geom_smooth(method = "lm", formula = y~x) + ggtitle('Linear Model for Fixed acidity VS Citric Acid')




p <- ggplot( data = redWineData,
        aes(x = fixed.acidity, y = density)
      ) +
  geom_point()

p2 <- p + geom_smooth(method = "lm", formula = y~x) + ggtitle('Linear Model for Fixed acidity VS Density')



p <- ggplot( data = redWineData,
        aes(x = fixed.acidity, y = pH)
      ) +
  geom_point()
```

```r
p3 <- p + geom_smooth(method = "lm", formula = y~x) + ggtitle('Linear Model for Fixed acidity VS pH')



p <- ggplot( data = redWineData,
        aes(x = free.sulfur.dioxide, y = total.sulfur.dioxide)
      ) +
  geom_point()

p4 <- p + geom_smooth(method = "lm", formula = y~x) + ggtitle('Linear Model for Free Sulfur dioxide VS Total Sulfur dioxide'
)

grid.arrange(p1,p2,p3,p4,ncol=2)
```



Reflection

Preclude:

The redwine dataset consists of 1599 red wine samples. I had information about various chemical factors involved in determining the quality of the samples, the quantity of these factors. My job as a data analyst is to recognize those factors which impact the quality of the wines the most. Such an analysis could help the senior management in identifying which factors to invest in more and which factors deserve lesser investment.

Analysis:

I started off with uni-variate analysis where in, I identified the amounts of individual ingredients that went into attaining the quality attained by the current dataset. After that, I carried out bi-variate analysis of all the chemical factors individually against the quality of the wine. From this, I identified certain factors that hold a strong impact on the quality of the wine and those that held a lesser impact on the quality of the wine. But, it was still not evident whether this impact was only because of the individual factors or were there other factors contributing as well. In order to identify these dependencies, I built a correlation coefficient matrix. This matrix gave me an idea of which variables held dependencies with the impact variables that were identified earlier from the uni variate analysis. By this point in the analysis, I was sure of which variables made an impact on the quality of the wines however the amount of impact was still not evident. I used single and multiple linear regression techniques to build linear models. These models gave me an idea of what quantities of individual variables are responsible for attainment of the current quality level and which variables need to be increased or decreased in tandem with other variables to improve the quality.

Conclusion and Future work :

I identified the impact variables that are responsible for the current quality levels of the dataset and in what quantities do I need to increase or decrease these variables in order to improve the quality. For future work, I would recommend some experiments to test the results of the above analysis and test the success level. Based on the results of these experiments, we could carry out further iterations of our analysis.

For my future work, I would also like choose a dataset which is in an unstructured form and bring that into an acceptable format (.csv, .xsl etc), apply some machine learning techniques to it and build interesting visualizations of my analysis.