Wine data analysis

Tarun Kumar

2024-02-01

Introduction

White wine is a popular beverage enjoyed by millions around the world for its refreshing taste and versatility. Understanding the factors that contribute to the quality of white wine is of great interest to winemakers, consumers, and researchers alike. In this report, we analyze a dataset containing information on various chemical properties of white wine, as well as its quality rating.

The dataset comprises 4898 observations and 12 variables, providing a comprehensive overview of key characteristics that may influence white wine quality. These variables include measures such as fixed acidity, volatile acidity, citric acid content, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH level, sulphates, alcohol content, and the quality rating assigned to each wine.

Our objective is to explore the relationships between these chemical properties and wine quality, with a particular focus on identifying significant factors that contribute to the perception of high-quality white wine. By leveraging statistical analysis and visualization techniques, we aim to uncover patterns, trends, and insights within the dataset that can inform both industry practices and consumer preferences.

univariate analysis

```
#load the CSV data into a data frame
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
df <- read.csv("Sem2/downloads/white_wine_data_1.csv")
str(df)
```

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

```
summary(df)
```

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

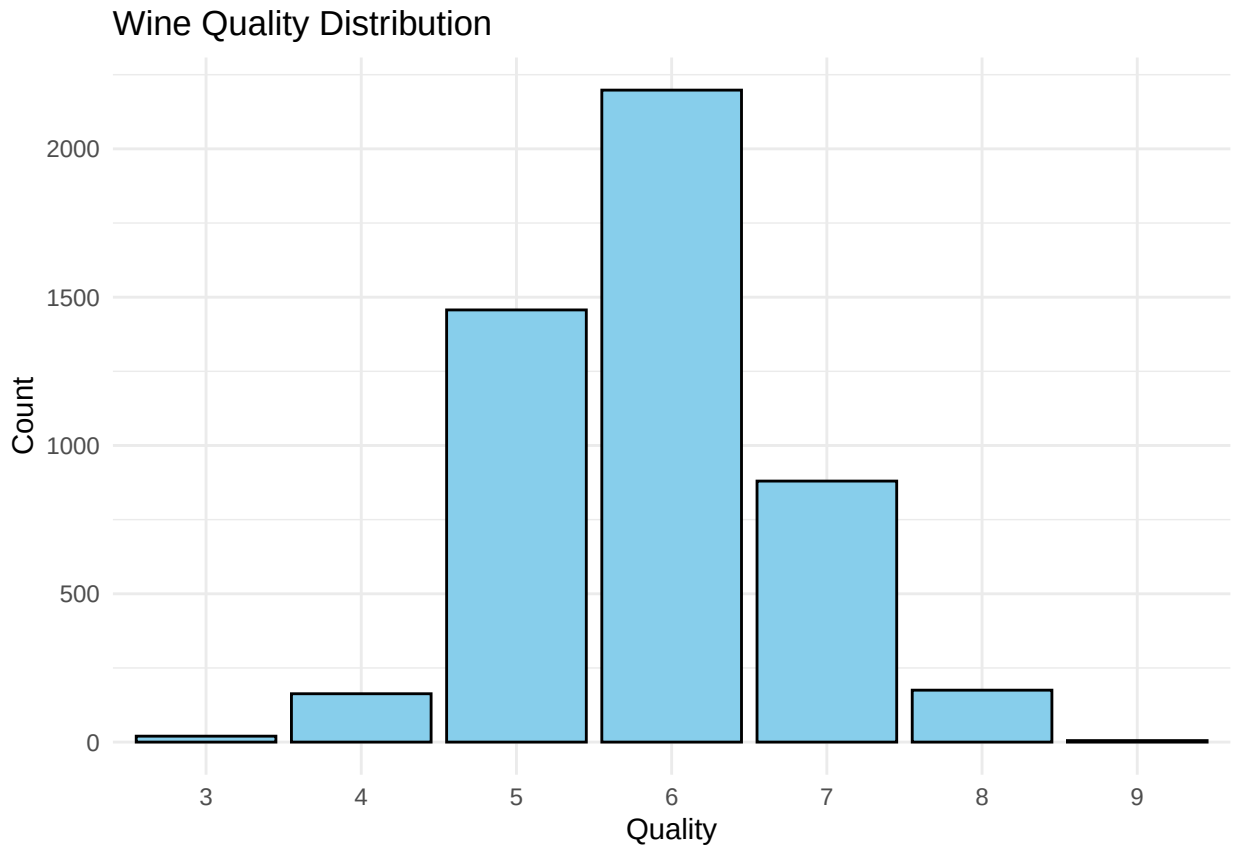
By presenting the summary statistics for the dataset, we can gain an initial understanding of the varying value ranges associated with each attribute. It becomes apparent that numerous characteristics display considerable outliers, given that the maximum values deviate significantly from their respective third quantiles.

```
column_names <- names(df)
print(column_names)
```

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

These are the features in the wine dataset. Upon close examination of the dataset, I have observed that the most suitable choice for the target vector is the wine quality. The reason behind this decision is straightforward: the wine quality is a pivotal feature in this dataset. Companies are keen to understand the factors influencing quality to enhance their product, which, in turn, has a direct impact on sales. Therefore, I will select the quality of wine as the target feature.

```
df %>%
  ggplot(aes(x = factor(quality))) +
  geom_bar(fill = "skyblue", color = "black") +
  labs(title = "Wine Quality Distribution",
       x = "Quality",
       y = "Count") +
  theme_minimal()
```

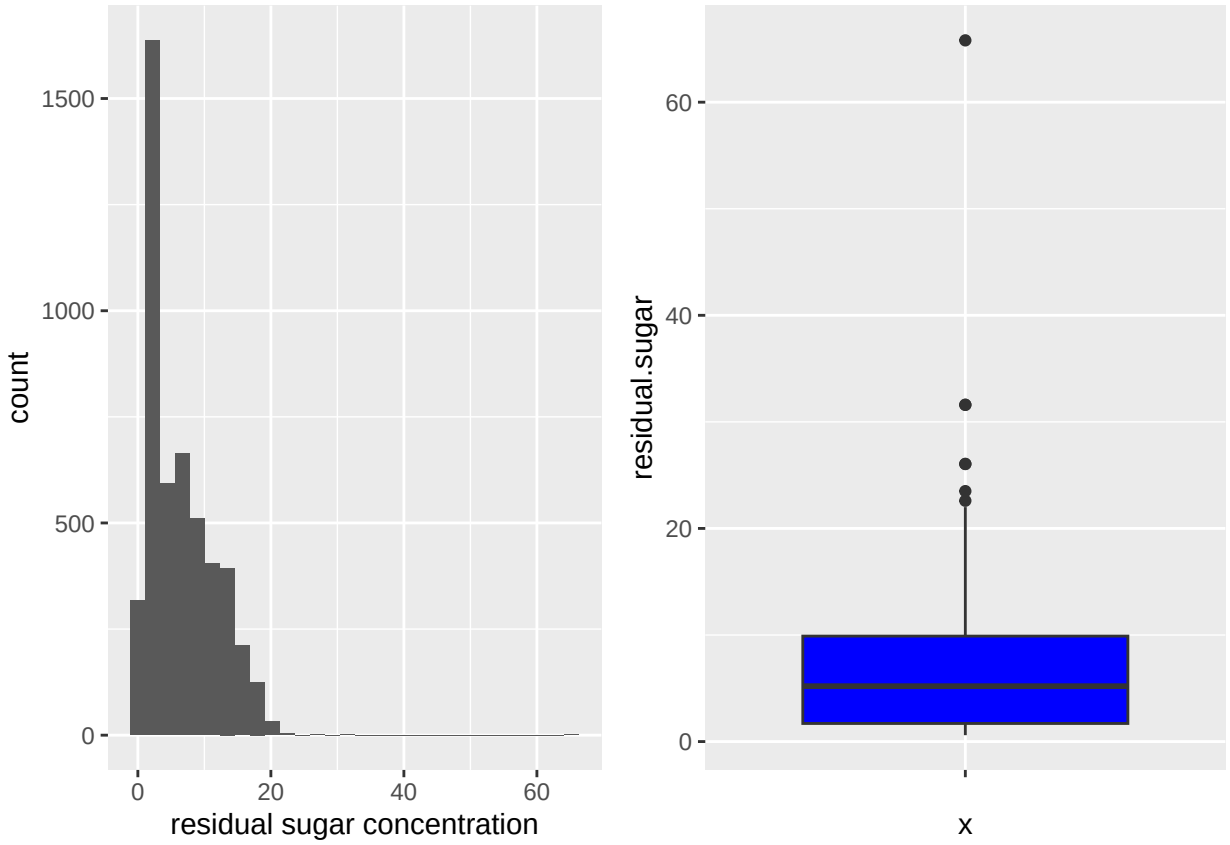


The distribution of wine quality appears to be relatively symmetrical. The majority of wines are rated with a quality score of 6. None of the wines received the maximum score of 10, while the lowest-rated wines were assigned a score of 3.

Residual sugar

```
p1 <- ggplot(df, aes(x='',y = residual.sugar)) + geom_boxplot(fill = "blue")
p2 <- ggplot(df, aes(x = residual.sugar)) + geom_histogram() +labs(x="residual sugar concentration")

grid.arrange(p2, p1, nrow = 1)
```

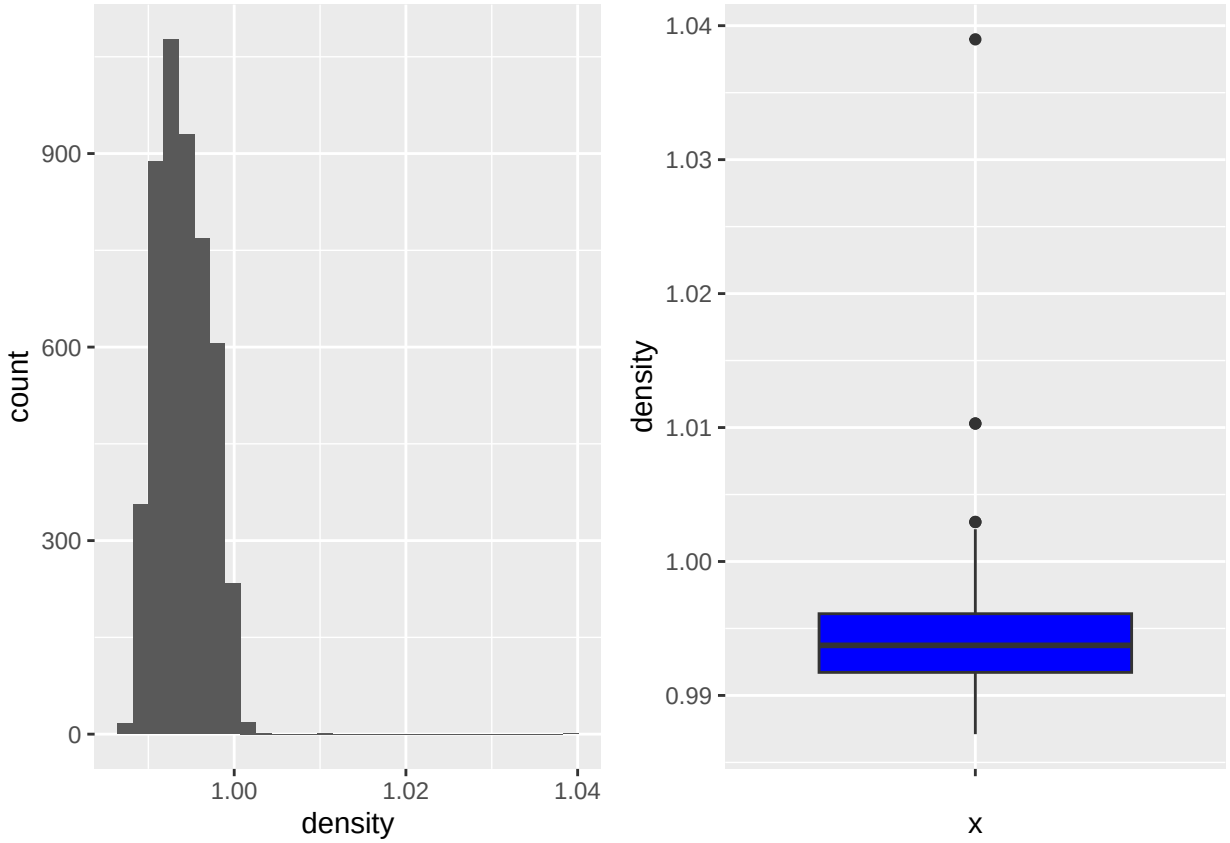
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



In general, the wines in the dataset seem to exhibit low concentrations of residual sugar. The positive skewness in the data results in a mean value (5.4) that is higher than the median (3.0). Notably, there is an extreme outlier with a residual sugar concentration of around 65 g/L.

Density

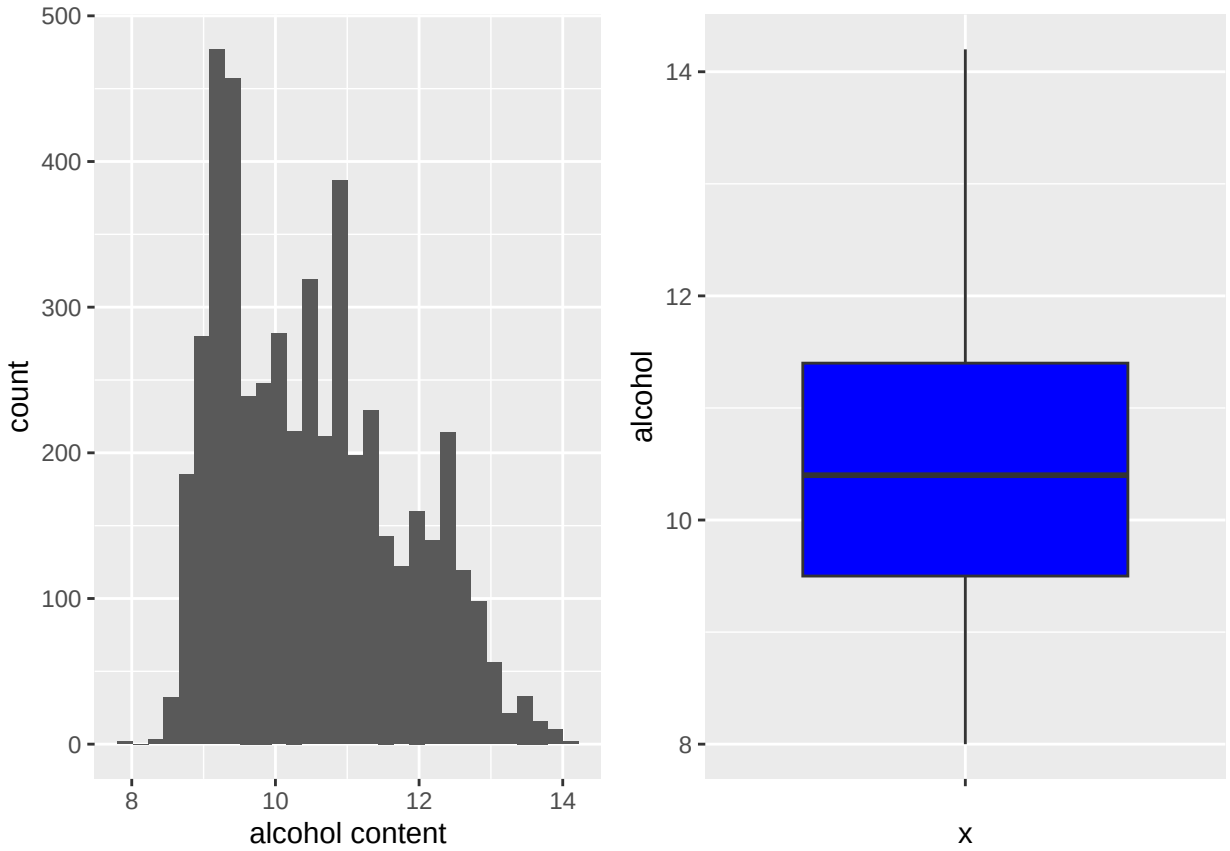
```
p1 <- ggplot(df, aes(x='',y = density)) + geom_boxplot(fill = "blue")
p2 <- ggplot(df, aes(x = density)) + geom_histogram(bins=30) +labs(x="density")

grid.arrange(p2, p1, nrow = 1)
```



density of wine have a narrow distribution with very low variance. While a few outliers exist around 1.01 and 1.04 g/cm3, the majority of wines exhibit a density ranging between 0.99 and 1.00 g/cm3.

Alcohol

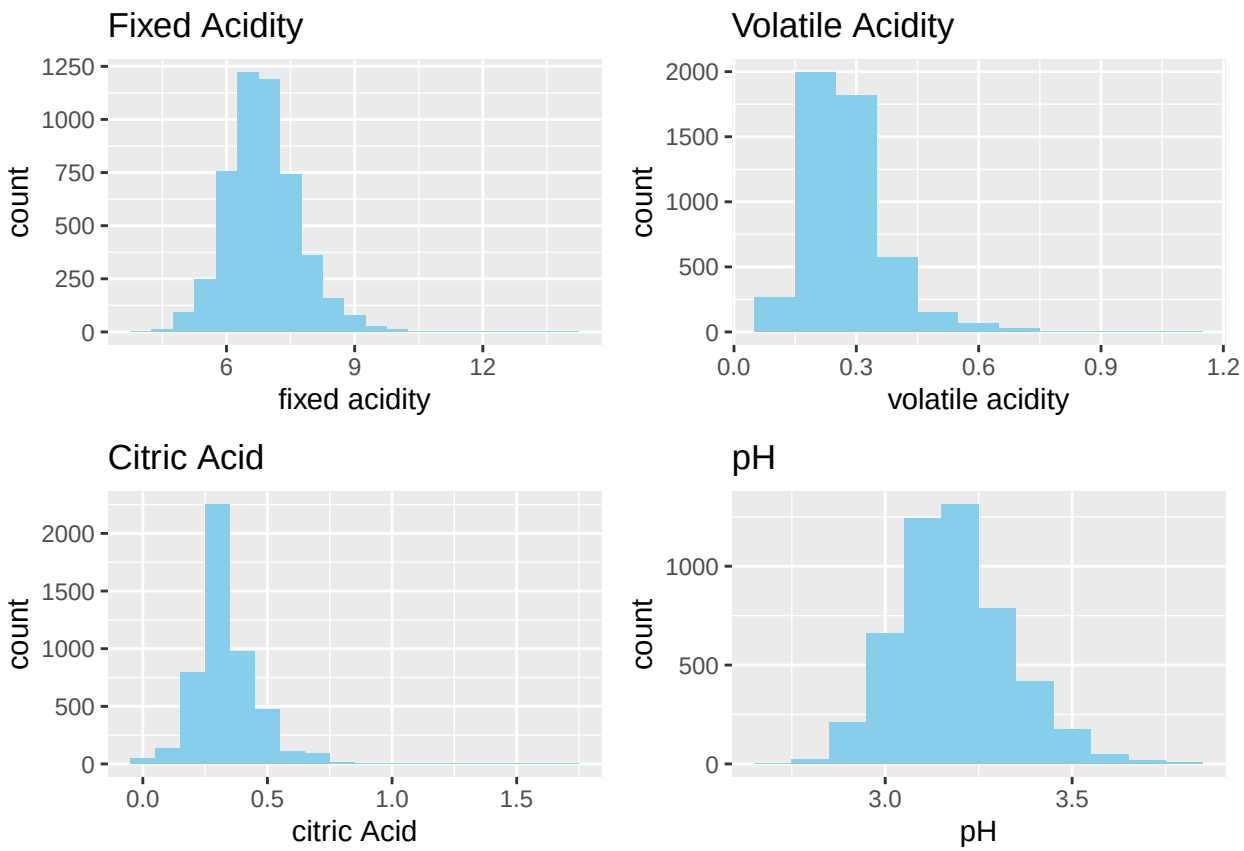
```
p1 <- ggplot(df, aes(x='',y = alcohol)) + geom_boxplot(fill = "blue")
p2 <- ggplot(df, aes(x = alcohol)) + geom_histogram(bins=30) +labs(x="alcohol content")

grid.arrange(p2, p1, nrow = 1)
```



The alcohol content of the wines in the dataset spans from 8 to 15 vol%. The median is approximately 10 vol%. The distribution is notably broad, indicating positive skewness in the data.

Acidity

```
p1 <- ggplot(df, aes(x = fixed.acidity)) +
  geom_histogram(binwidth = 0.5, fill = "skyblue") + labs(title = "Fixed Acidity", x = "fixed acidity")

p2 <- ggplot(df, aes(x = volatile.acidity)) +geom_histogram(binwidth = 0.1, fill = "skyblue") +labs(tit

p3 <- ggplot(df, aes(x = citric.acid)) +geom_histogram(binwidth = 0.1, fill = "skyblue") +labs(title =
p4 <- ggplot(df, aes(x = pH)) + geom_histogram(binwidth = 0.1, fill = "skyblue") +labs(title = "pH", x =
grid.arrange(p1, p2, p3, p4, ncol = 2)
```

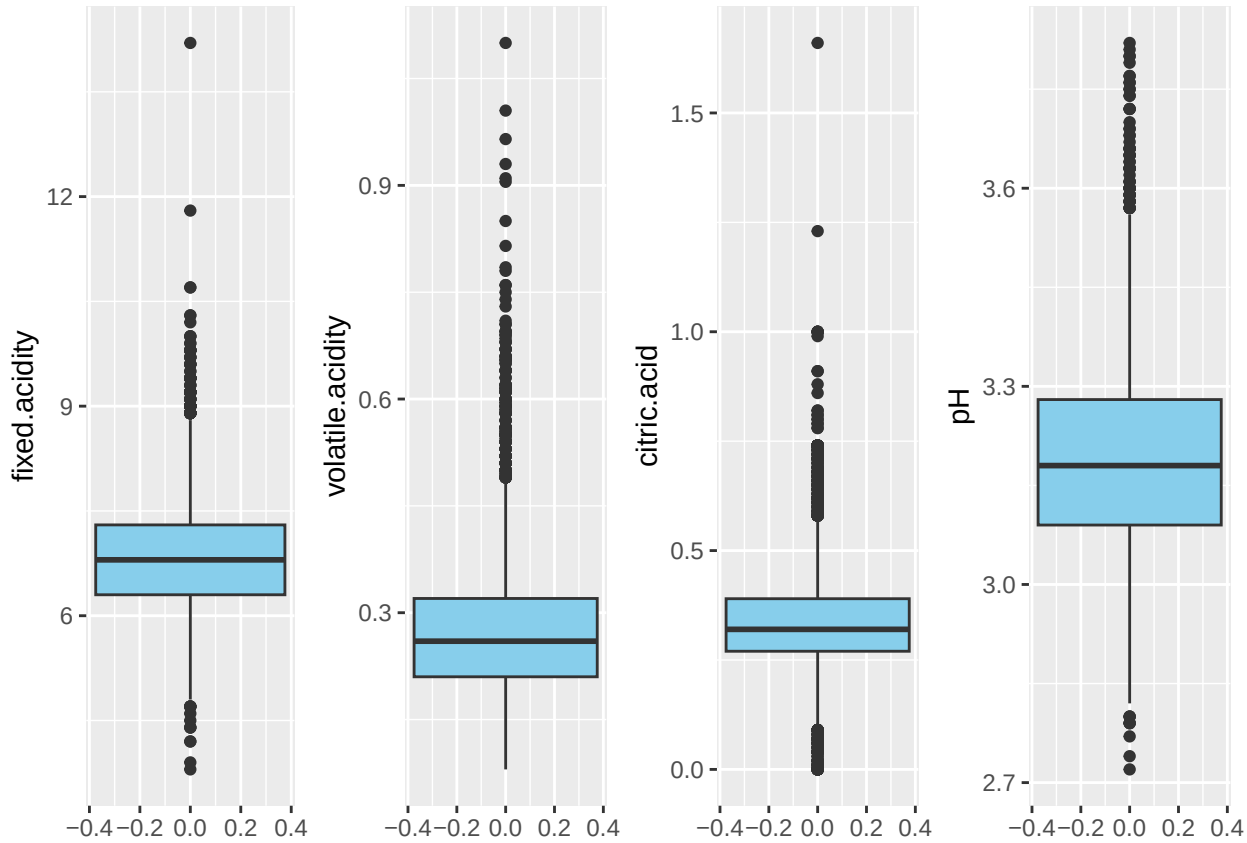


Fixed and volatile acidity: Both distributions exhibit positive skewness, indicating that most wines have lower acidity values, with a smaller tail extending towards higher values.

Citric acid: The distribution shows an "edge peak", suggesting a significant portion of wines have very low citric acid concentrations near 0, with a smaller number spread across higher values.

pH: The distribution appears relatively symmetrical, suggesting a more even spread of pH values across the dataset. This aligns with pH ranges observed in wines.

```
p1 <- ggplot(df, aes(y = fixed.acidity)) + geom_boxplot(fill = "skyblue")
p2 <- ggplot(df, aes(y = volatile.acidity)) + geom_boxplot(fill = "skyblue")
p3 <- ggplot(df, aes(y = citric.acid)) +geom_boxplot(fill = "skyblue")
p4 <- ggplot(df, aes(y = pH)) +geom_boxplot(fill = "skyblue")
grid.arrange(p1, p2, p3, p4, nrow = 1)
```



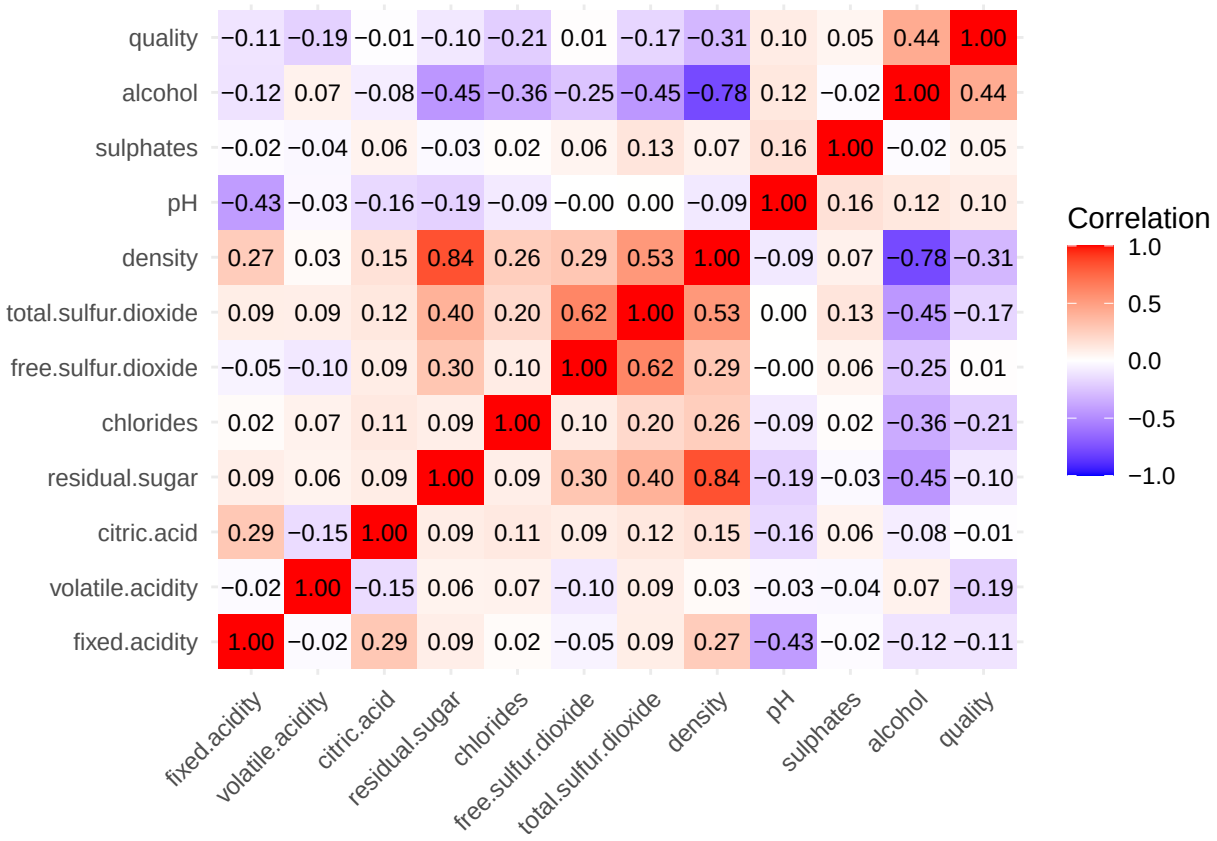
When examining the acidity parameters through boxplots, a similar pattern emerges. Both fixed and volatile acidity exhibit long positive tails in their distributions, indicating a significant spread of concentrations. In contrast, the distributions for citric acid and pH appear narrower, suggesting a more concentrated range of values for these parameters.

Bivariate analysis and Linear Regression

Correlation Matrix

```r
# Load the required libraries
library(ggplot2)
library(reshape2)  # For melting the correlation matrix



# Compute the correlation matrix
correlation_matrix <- cor(df)
#print(correlation_matrix)

# Melt the correlation matrix for plotting
correlation_data <- melt(correlation_matrix)

# Plot the correlation matrix as a heatmap
ggplot(data = correlation_data, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1,1), space =
```

```
  geom_text(aes(label=sprintf("%.2f", value)), color="black", size=3) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1),
        axis.title = element_blank(),
        legend.title.align=0.5) +
  labs(fill="Correlation")
```



Observation from correlation matrix and plot:

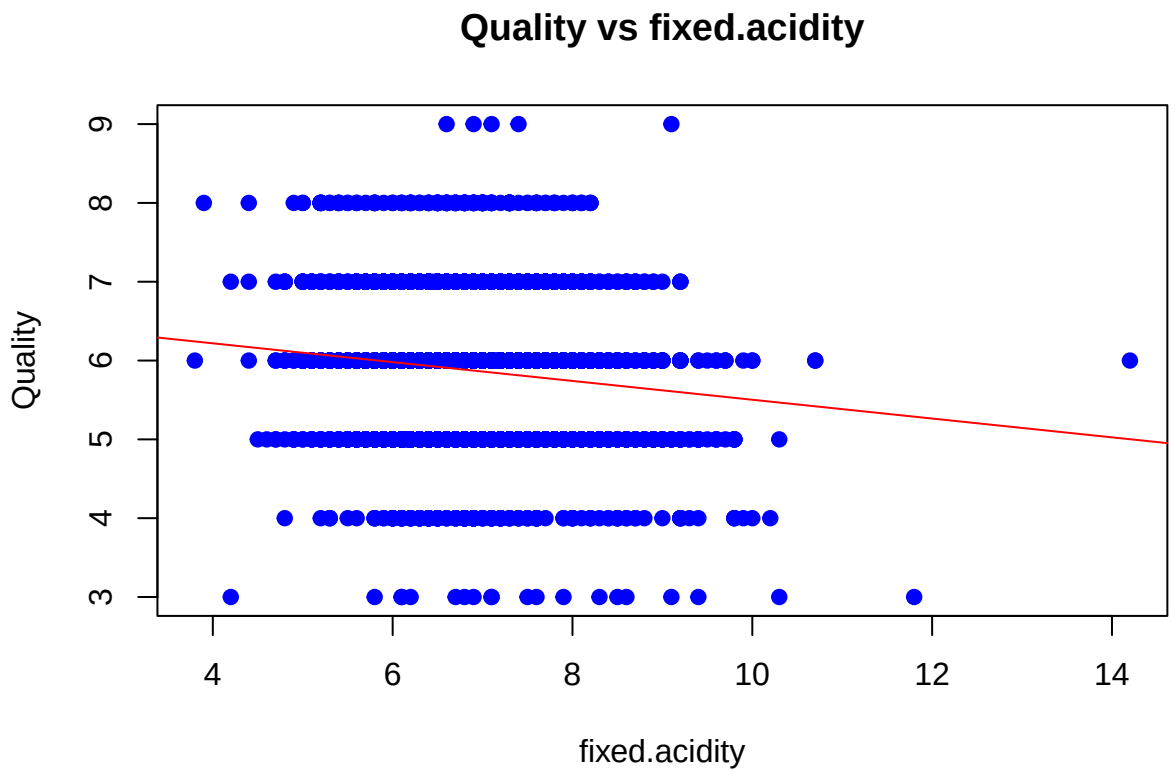
- There is a positive correlation between alcohol content and wine quality and negative with density.
- The correlation between wine quality and citric acis, as well as between wine quality and sulfur dioxide ratio, is very low.
- Wine quality exhibits a slight negative correlation with volatile acidity.
- There appears to be a relationship between sulfur dioxide and residual sugar in wines.
- We anticipate that alcohol content and residual sugar concentration will influence wine density.
- There is a correlation between fixed acidity and total fixed acidity, as the former is a component of the latter.
- Similarly, there is a relationship between free sulfur dioxide, total sulfur dioxide, and the sulfur dioxide ratio.
- Color demonstrates relationships with density, residual sugar, total sulfur dioxide, and volatile acidity.

Simple Linear Regression

we have familiarized ourselves with the features, we can proceed to analyze the correlation between each feature and the target vector. We'll employ a simple linear regression approach and visualize the relationships

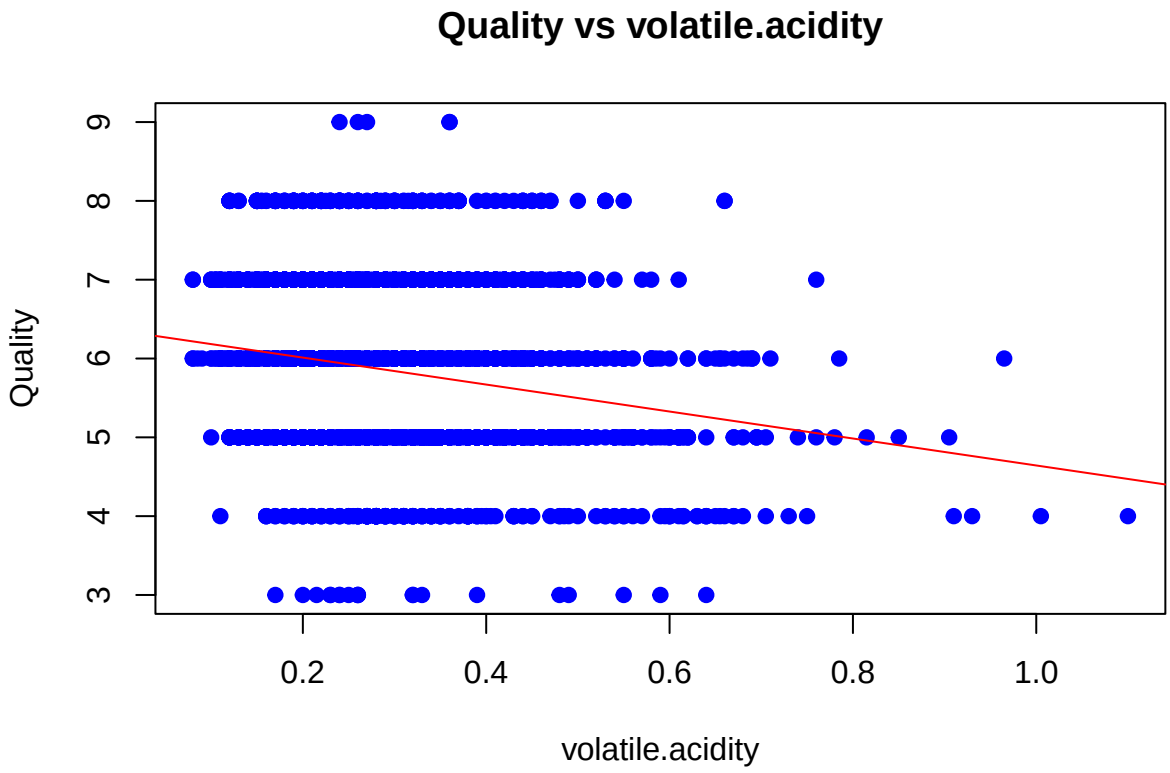
to gain insights into the dependency of the target vector on the features.

```
# Perform simple linear regression for each feature with quality
features <- c("fixed.acidity", "volatile.acidity", "citric.acid", "residual.sugar",
              "chlorides", "free.sulfur.dioxide", "total.sulfur.dioxide",
              "density", "pH", "sulphates", "alcohol")

# Loop through each feature and fit a linear regression model
for (feat in features) {
  model <- lm(quality ~ ., data = df[, c(feat, "quality")])

   # Plot the data points
  plot(df[[feat]], df$quality, main = paste("Quality vs", feat),
       xlab = feat, ylab = "Quality", pch = 19, col = "blue")

  # Add the fitted line to the plot
  abline(model, col = "red")

  # Print the summary of the linear regression model
  cat("Feature:", feat, "\n")
  print(summary(model))
  cat("\n")
}
```



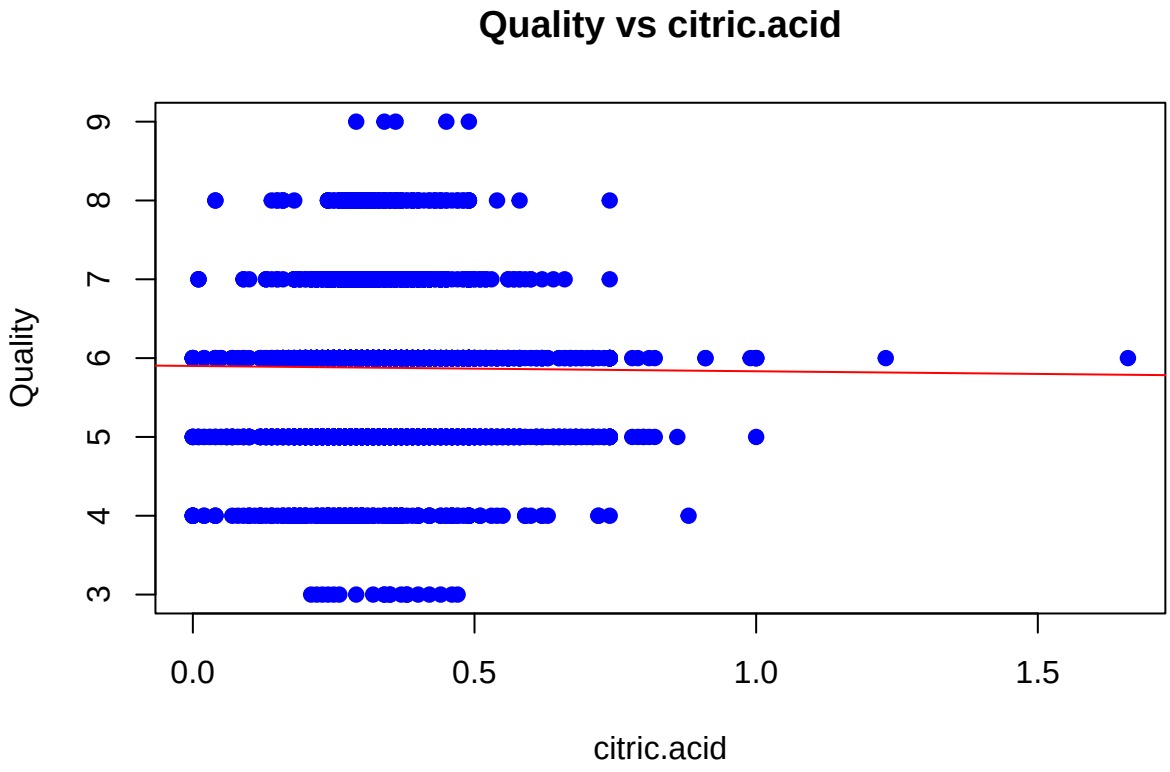

```
## Feature: fixed.acidity
##
```

```
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1946 -0.8248  0.0798  0.2706  3.3899
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.6956     0.1029  65.057  < 2e-16 ***
## fixed.acidity  -0.1193     0.0149  -8.005 1.48e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.88 on 4896 degrees of freedom
## Multiple R-squared:  0.01292,    Adjusted R-squared:  0.01272
## F-statistic: 64.08 on 1 and 4896 DF,  p-value: 1.48e-15
```



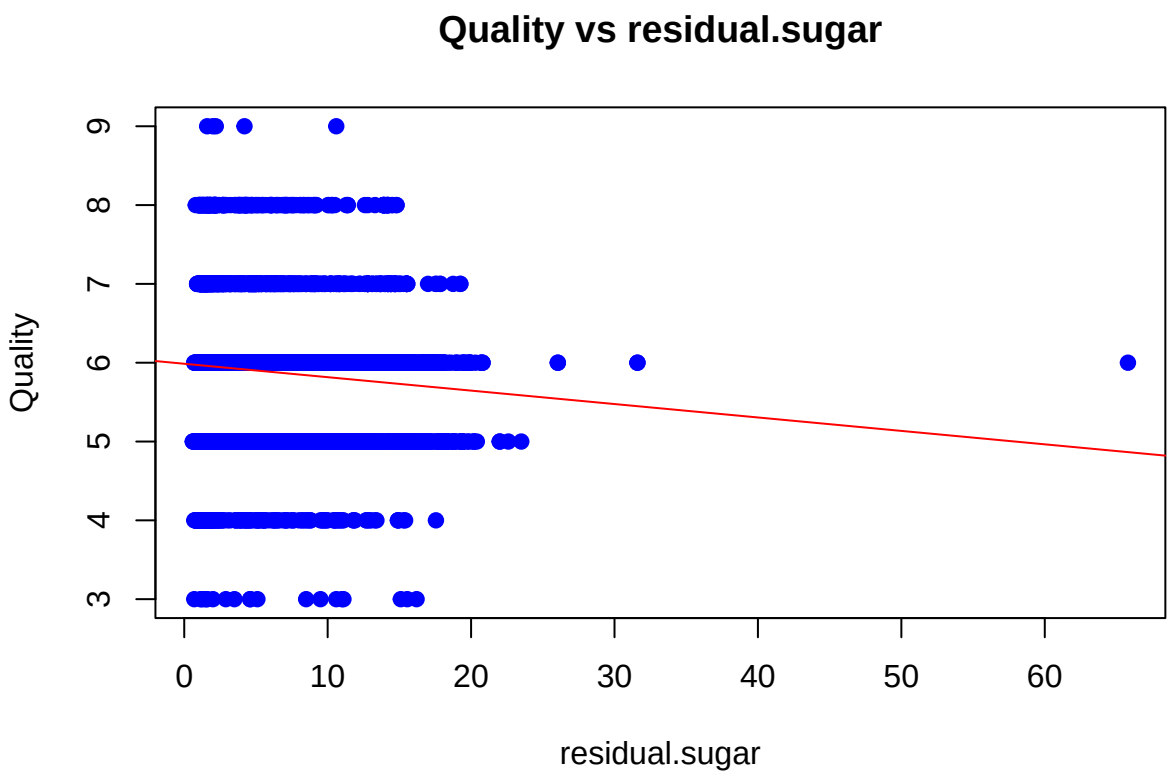

```
## Feature: volatile.acidity
##
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.0631 -0.7894  0.0224  0.3133  3.2620
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.35397    0.03645  174.32   <2e-16 ***
## volatile.acidity -1.71095    0.12317  -13.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8688 on 4896 degrees of freedom
## Multiple R-squared:  0.03792,    Adjusted R-squared:  0.03772
## F-statistic:   193 on 1 and 4896 DF,  p-value: < 2.2e-16
```



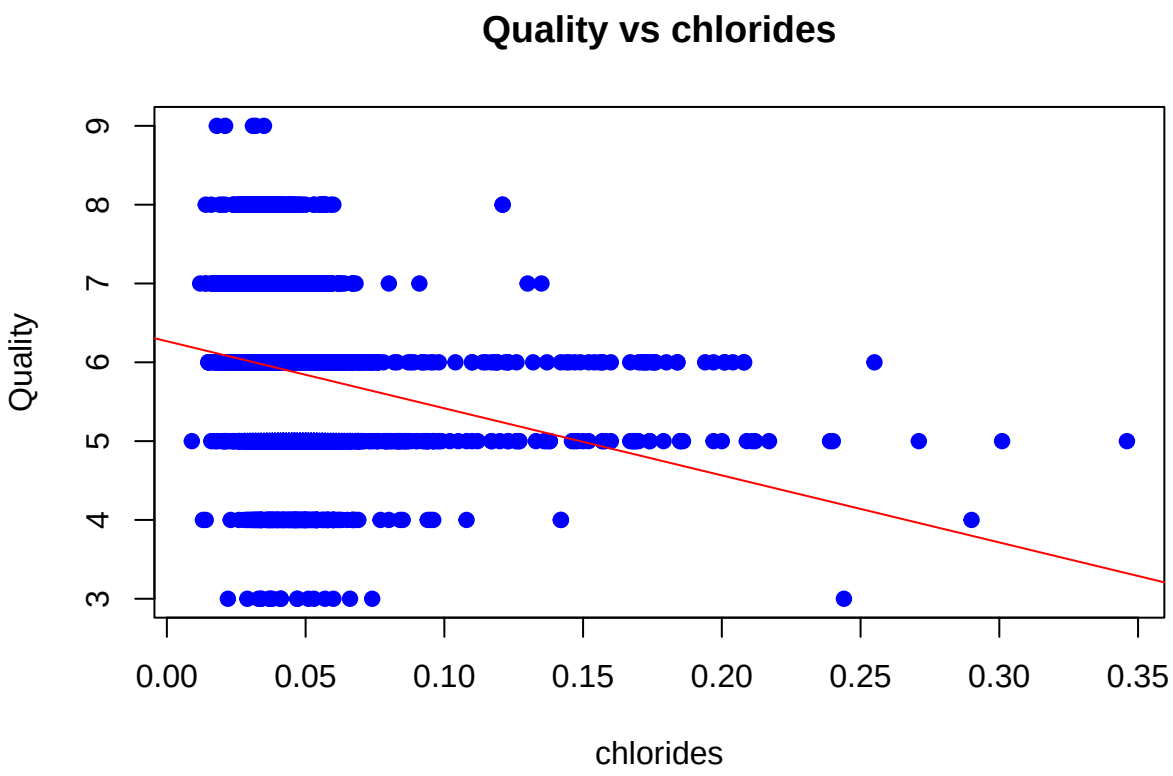

```
## Feature: citric.acid
##
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8863 -0.8735  0.1191  0.1326  3.1326
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.90043    0.03717 158.736   <2e-16 ***
```

```
## citric.acid -0.06739    0.10458  -0.644    0.519
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8857 on 4896 degrees of freedom
## Multiple R-squared:  8.481e-05,  Adjusted R-squared:  -0.0001194
## F-statistic: 0.4153 on 1 and 4896 DF,  p-value: 0.5193
```

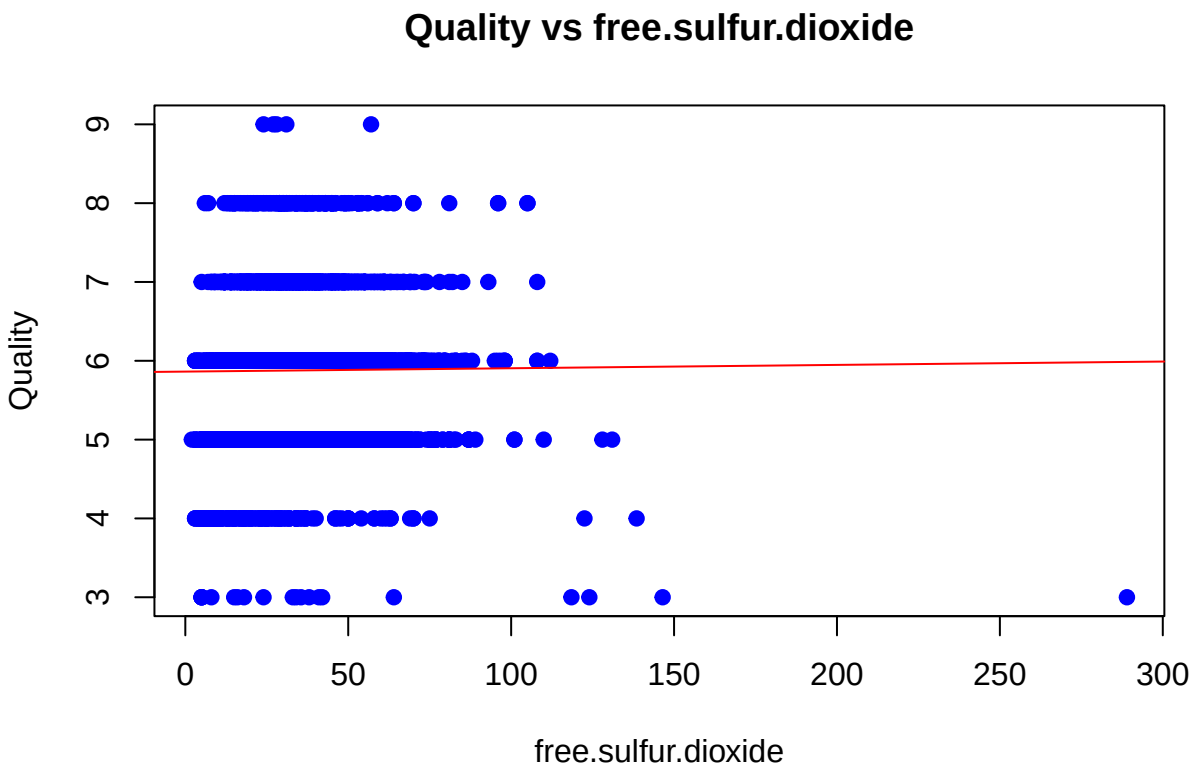



```
## Feature: residual.sugar
##
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9749 -0.8058  0.0609  0.2611  3.1938
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.986806   0.020264  295.45  < 2e-16 ***
## residual.sugar -0.017038   0.002484   -6.86 7.72e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8815 on 4896 degrees of freedom
```

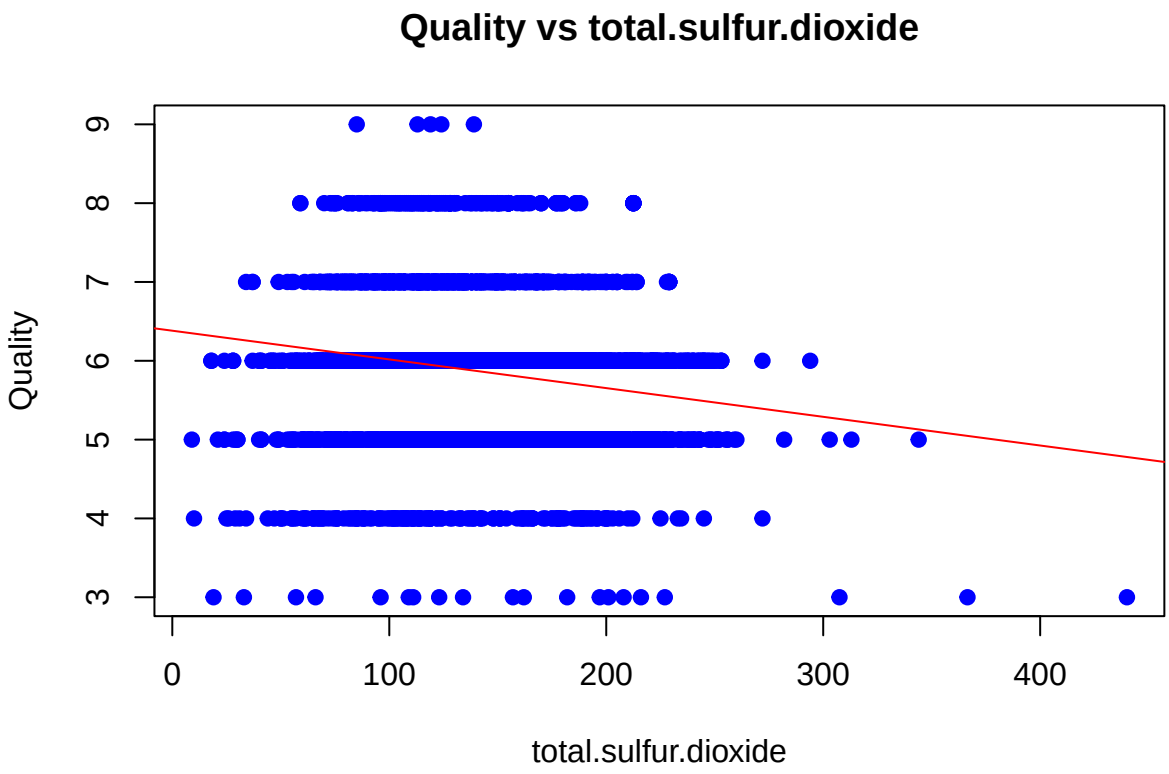
```
## Multiple R-squared:  0.009521,   Adjusted R-squared:  0.009319
## F-statistic: 47.06 on 1 and 4896 DF,  p-value: 7.724e-12
```



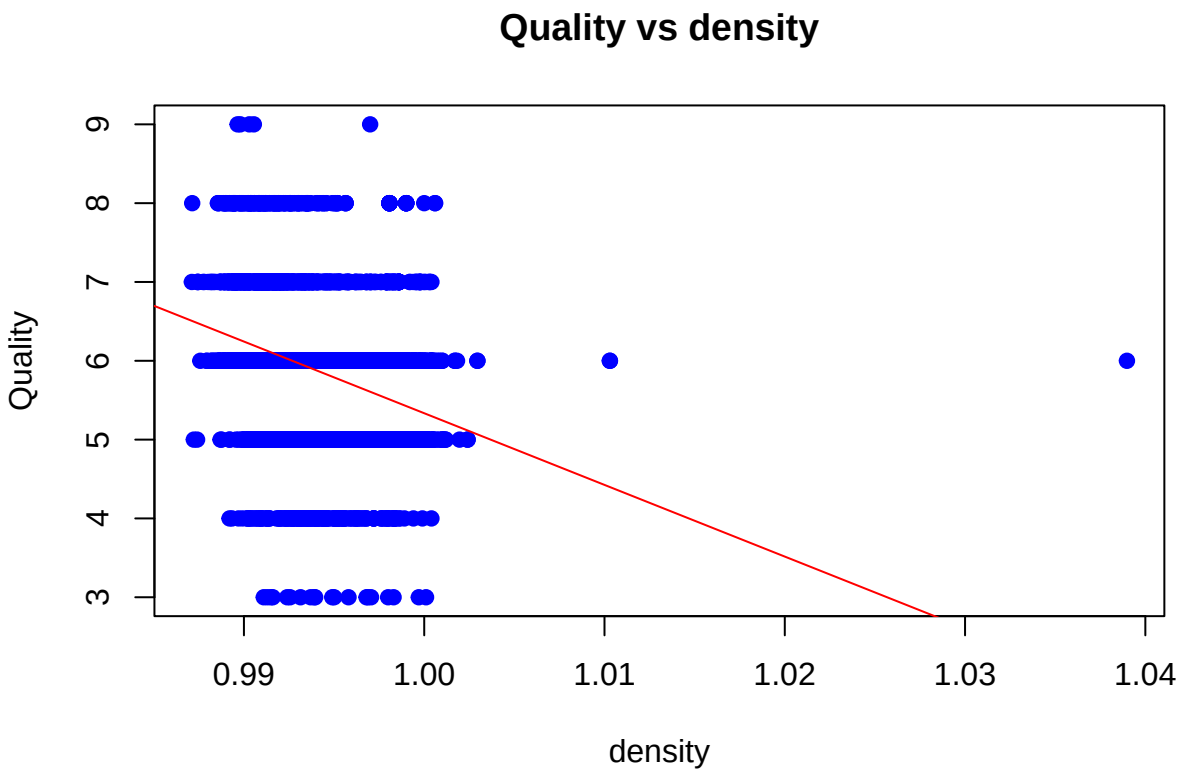

```
## Feature: chlorides
##
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.08021 -0.82491  0.06446  0.24317  3.03042
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.26743    0.02873  218.17   <2e-16 ***
## chlorides   -8.50999    0.56642  -15.02   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.866 on 4896 degrees of freedom
## Multiple R-squared:  0.04407,    Adjusted R-squared:  0.04388
## F-statistic: 225.7 on 1 and 4896 DF,  p-value: < 2.2e-16
```



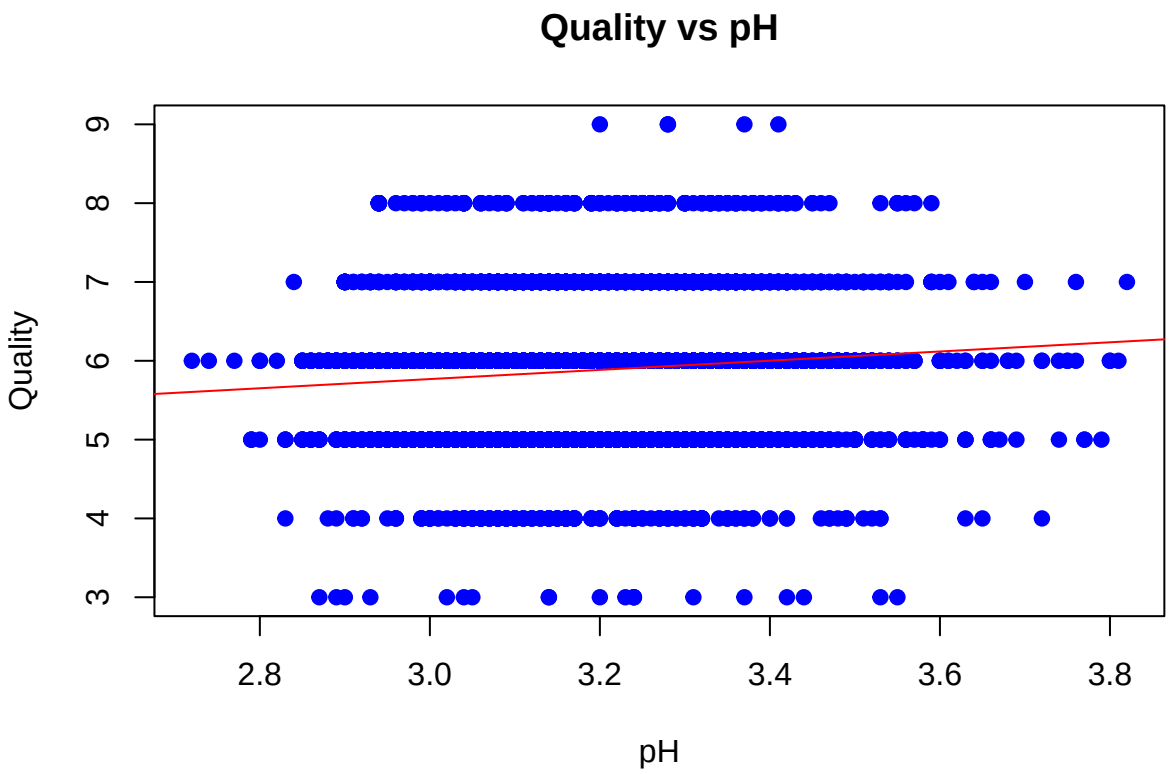

```
## Feature: free.sulfur.dioxide
##
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9857 -0.8731  0.1205  0.1307  3.1269
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.8629095  0.0291651 201.025   <2e-16 ***
## free.sulfur.dioxide 0.0004248  0.0007442   0.571    0.568
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8857 on 4896 degrees of freedom
## Multiple R-squared:  6.655e-05,  Adjusted R-squared:  -0.0001377
## F-statistic: 0.3259 on 1 and 4896 DF,  p-value: 0.5681
```



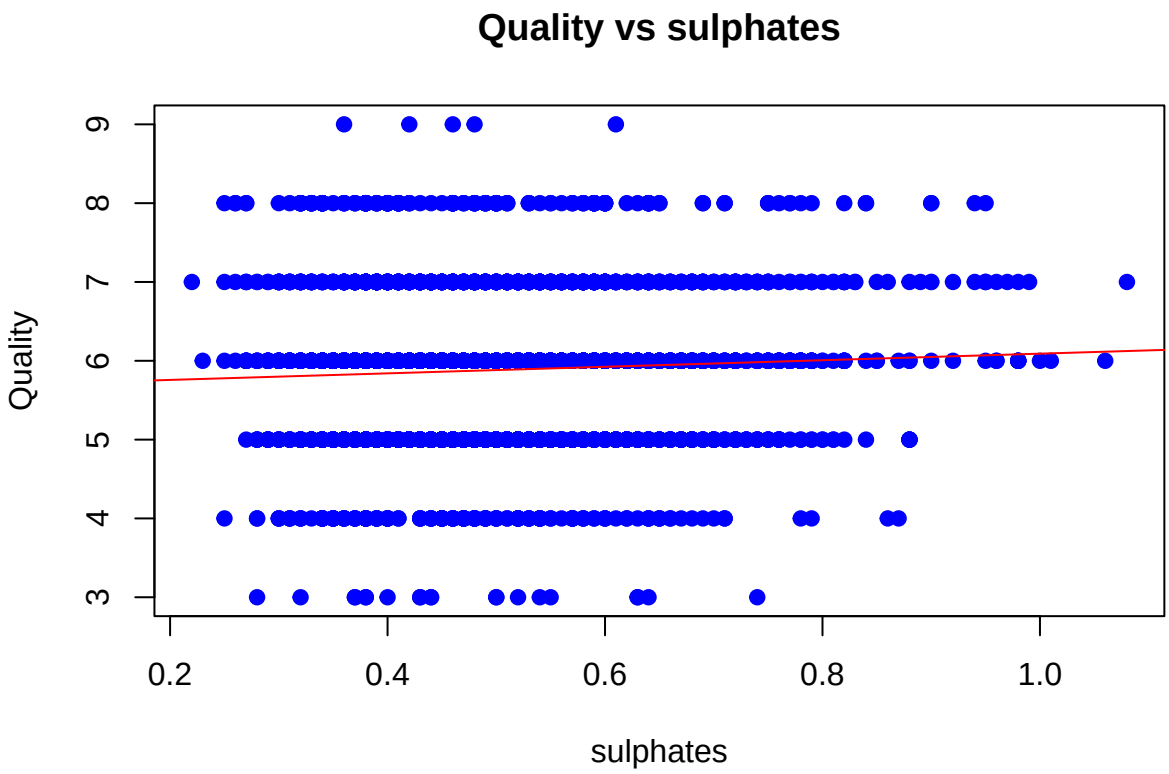

```
## Feature: total.sulfur.dioxide
##
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3126 -0.7336  0.0479  0.3465  3.1244
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           6.3817410  0.0424442  150.36   <2e-16 ***
## total.sulfur.dioxide -0.0036414  0.0002932  -12.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8721 on 4896 degrees of freedom
## Multiple R-squared:  0.03053,    Adjusted R-squared:  0.03034
## F-statistic: 154.2 on 1 and 4896 DF,  p-value: < 2.2e-16
```



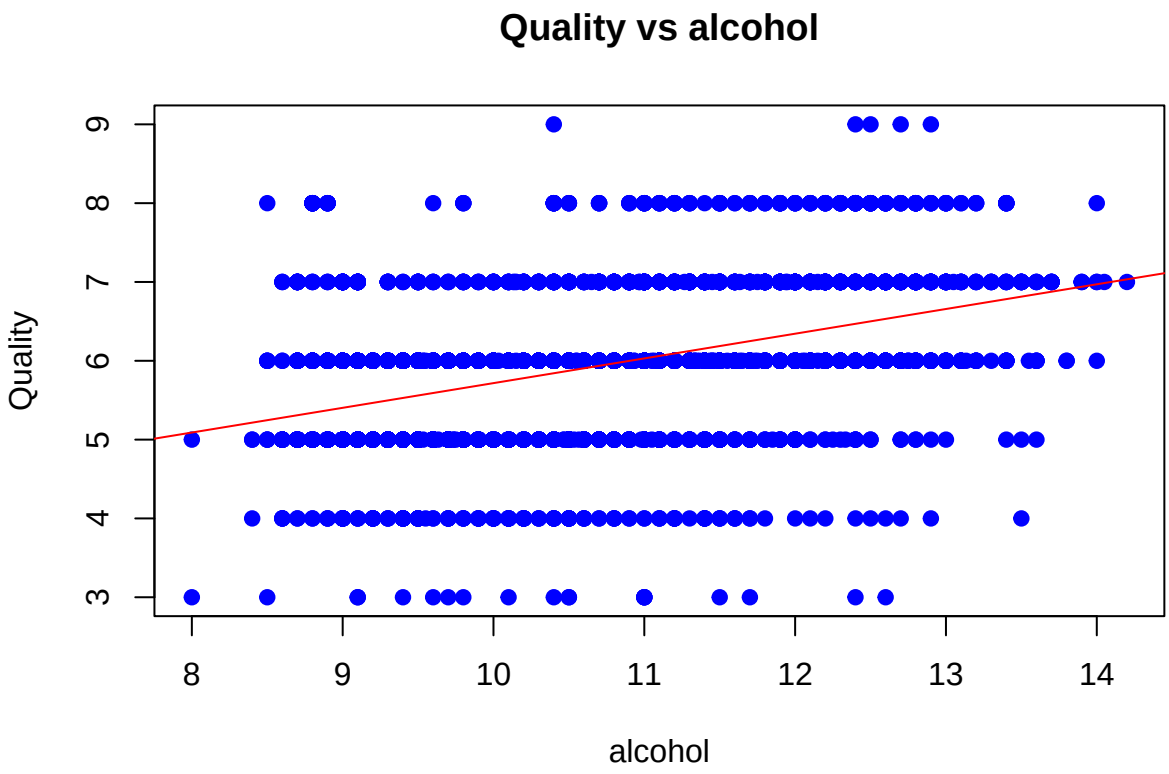

```
## Feature: density
##
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1441 -0.6258  0.0005  0.5162  4.2102
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   96.277      4.003   24.05   <2e-16 ***
## density      -90.942      4.027  -22.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8429 on 4896 degrees of freedom
## Multiple R-squared:  0.09432,    Adjusted R-squared:  0.09414
## F-statistic: 509.9 on 1 and 4896 DF,  p-value: < 2.2e-16
```




```
## Feature: pH
##
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.08886 -0.82060  0.09775  0.23771  3.11525
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.01866    0.26622  15.095  < 2e-16 ***
## pH           0.58315    0.08341   6.992 3.08e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8813 on 4896 degrees of freedom
## Multiple R-squared:  0.009886,   Adjusted R-squared:  0.009684
## F-statistic: 48.88 on 1 and 4896 DF,  p-value: 3.081e-12
```




```
## Feature: sulphates
##
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9821 -0.8488  0.1137  0.1803  3.1762
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.6739     0.0557 101.863  < 2e-16 ***
## sulphates     0.4165     0.1108   3.761 0.000171 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8845 on 4896 degrees of freedom
## Multiple R-squared:  0.002881,   Adjusted R-squared:  0.002678
## F-statistic: 14.15 on 1 and 4896 DF,  p-value: 0.000171
```




```
## Feature: alcohol
##
## Call:
## lm(formula = quality ~ ., data = df[, c(feat, "quality")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5317 -0.5286  0.0012  0.4996  3.1579
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.582009   0.098008   26.34   <2e-16 ***
## alcohol     0.313469   0.009258   33.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7973 on 4896 degrees of freedom
## Multiple R-squared:  0.1897, Adjusted R-squared:  0.1896
## F-statistic:  1146 on 1 and 4896 DF,  p-value: < 2.2e-16
```

Simple Linear Regression Analysis

Above are the result of simple linear regression between each feature and target feature(wine quality), apart from this i have plotted both feature and wine quality and showed regression lines.

Significance Testing with T-Statistics and P-Values: In each result of linear regression we can observe t-statistics and p-value.t-statistics are used to determine the statistical significance of each independent variable in relation to the dependent variable while controlling for the other independent variables in the model. For example a 0.05 significance level t-value is 3.182, so if the t-value is greater than 3.182 or the corresponding p-value is smaller than 0.05 we will reject the null hypothesis (no relationship held between the independent variable and target vector).

Criteria for Rejection of Null Hypothesis: Set threshold at a t-value greater than 3.182 or a corresponding p-value less than 0.05 to reject the null hypothesis. Indicates a lack of relationship between the independent variable and the target vector.

Acceptance or Rejection of Null Hypothesis: Despite the relatively small values of the coefficient of determination (R-squared) for all features, the significant t-statistics compel us to reject the null hypothesis, suggesting that the coefficient of the predicted model (i.e., the regression line) is not equal to zero. However, there are exceptions where the t-statistics are notably low, indicating an inability to reject the null hypothesis. For instance, in the cases of citric acid and free sulfur dioxide, the t-statistics are -0.644 and 0.571 respectively, with corresponding p-values of 0.519 and 0.568. Notably, only one feature, alcohol concentation, achieves an R-squared value exceeding 0.1, reaching 0.18

Overall Conclusion: - No single feature alone sufficiently explains the variability of wine quality. - Relying solely on simple linear regression may not be an effective approach for modeling the wine data.

Multiple Linear regression

Based on past experiences with simple linear regression, it has become evident that this method did not yield satisfactory results. Therefore, we have decided to proceed with multiple linear regression for our analysis.

```
# Fit a multiple linear regression model
model <- lm(quality ~ ., data = df)

# Print the summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = quality ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8348 -0.4934 -0.0379  0.4637  3.1143
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.502e+02  1.880e+01   7.987 1.71e-15 ***
## fixed.acidity         6.552e-02  2.087e-02   3.139  0.00171 **
## volatile.acidity     -1.863e+00  1.138e-01 -16.373  < 2e-16 ***
## citric.acid           2.209e-02  9.577e-02   0.231  0.81759
## residual.sugar        8.148e-02  7.527e-03  10.825  < 2e-16 ***
## chlorides            -2.473e-01  5.465e-01  -0.452  0.65097
## free.sulfur.dioxide   3.733e-03  8.441e-04   4.422 9.99e-06 ***
## total.sulfur.dioxide -2.857e-04  3.781e-04  -0.756  0.44979
## density              -1.503e+02  1.907e+01  -7.879 4.04e-15 ***
## pH                    6.863e-01  1.054e-01   6.513 8.10e-11 ***
## sulphates             6.315e-01  1.004e-01   6.291 3.44e-10 ***
```

```
## alcohol               1.935e-01  2.422e-02   7.988 1.70e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7514 on 4886 degrees of freedom
## Multiple R-squared:  0.2819, Adjusted R-squared:  0.2803
## F-statistic: 174.3 on 11 and 4886 DF,  p-value: < 2.2e-16
```

Observation

Estimated coefficients of multiple linear regression provide the information that: A one-unit increase in particular feature is associated with a increase/decrease(depending on sign) of estimated coefficient(beta) units in wine quality.

Model Evaluation:

- **Residual Standard Error:** The residual standard error of 0.7514 indicates the average difference between the observed and predicted values. Lower values suggest a better fit of the model to the data.
- **Multiple R-squared:** The coefficient of determination is 28.19%, signifying that 28.19% of the variability in wine quality is explained by the model. A higher R-squared indicates a better fit, but in this case, a significant portion of the variability remains unexplained.
- **Adjusted R-squared:** The adjusted R-squared, at 28.03%, accounts for the number of predictors in the model. It is slightly lower than the multiple R-squared, suggesting that the inclusion of predictors may not be contributing substantially to the explanatory power.
- **F-statistic:** The F-statistic of 174.3 with a p-value less than 2.2e-16 indicates that the overall model is statistically significant. This implies that at least one predictor variable is significantly related to the response variable.

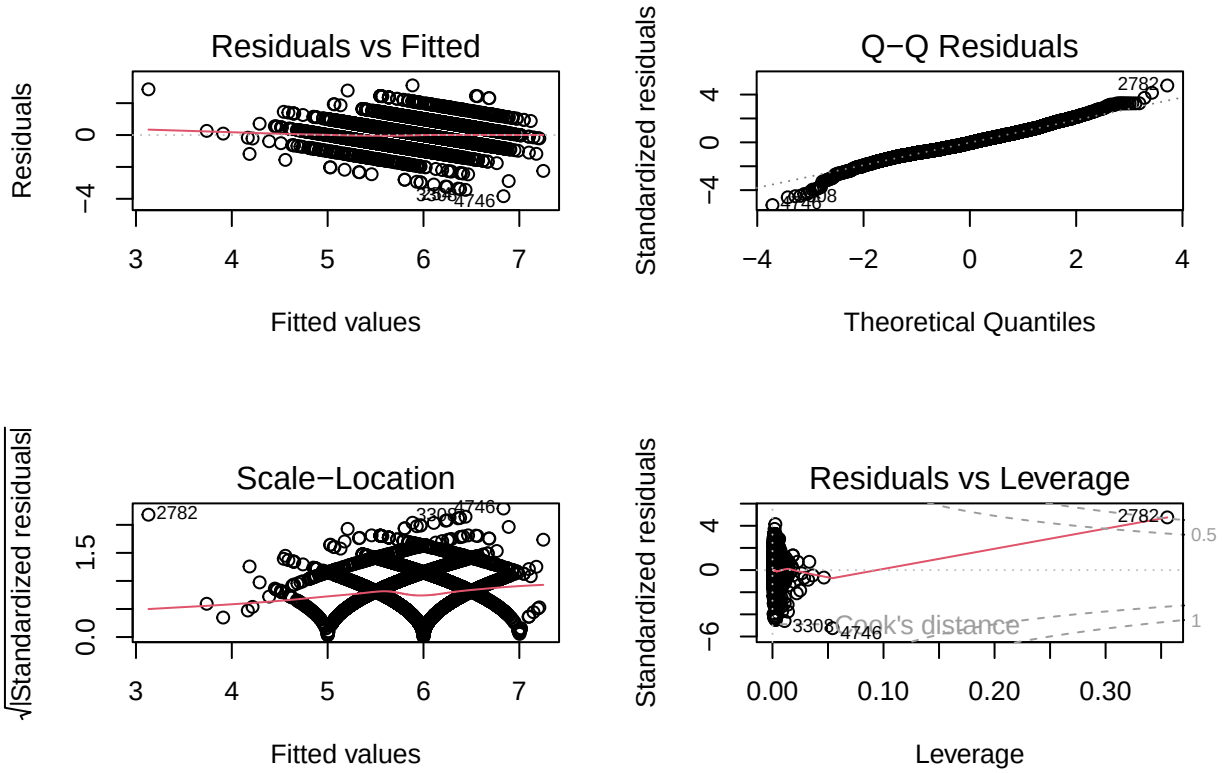
Conclusion:

- The model identifies volatile acidity, residual sugar, free sulfur dioxide, density, pH, sulphates, and alcohol as statistically significant predictors of wine quality. Changes in these variables are associated with changes in wine quality.
- Citric acid and total sulfur dioxide, however, do not appear to have a significant impact on wine quality according to the model.
- Despite statistical significance, the model only explains a moderate proportion of the variability in wine quality. This suggests that other factors not included in the current model may influence wine quality. Also it provide the insight that linear regression is not suitable for modelling this data set.

Regression Diagnostic plots

```r
par(mfrow = c(2, 2))

# Plot the diagnostics
# Residuals vs Fitted Values Plot
plot(model, which = 1)
# Normal Q-Q Plot of Residuals
plot(model, which = 2)
```

```r
# Scale-Location Plot
plot(model, which = 3)
# Residuals vs Leverage Plot:
plot(model, which = 5)
```



Observations and Inferences:

1. Residuals vs Fitted Plot:
   - Observation: Residuals are not random along the fitted values (x-axis). There seems to be a pattern or trend in the residuals, suggesting that the spread of the residuals does not remain consistent across the range of fitted values.
   - Inference: The assumption of constant variance (homoscedasticity) may be violated. This indicates that the variability of the residuals changes as the predicted wine quality values change.
2. Normal Q-Q Plot of Residuals:
   - Observation: Residuals are not perfectly aligned with the y=x line. They are deviated on both extremes, indicating departures from normality in the distribution of residuals.
   - Inference: The assumption of normality of residuals may be violated. This suggests that the residuals do not follow a perfectly normal distribution, which can impact the accuracy of statistical inferences made using the regression model.
3. Scale-Location Plot (Spread-Location Plot):
   - Observation: Similar to the Residuals vs Fitted plot, the spread of residuals is not consistent across the range of fitted values.

- Inference: The observation suggest the indication of heteroscedasticity, where the spread of residuals varies systematically with the predicted values. This strengthens the need to address the violation of the constant variance assumption in the regression model.

4. Residuals vs Leverage Plot:
   - Observation: There are many points that are far from the x-axis (leverage = 0), indicating potentially influential observations.
   - Inference: The presence of outliers or influential points suggests that certain observations disproportionately influence the estimated regression coefficients. These influential points may have a significant impact on the regression model's predictions and should be carefully examined to assess their validity and potential effects on the model's performance.

Logistic Regression

Previously, our attempts at predicting wine quality using linear models didn't work well. So, we're changing our approach. Instead of predicting the exact quality score, we're now simplifying the problem to just two categories: good wine and bad wine. We've created a new feature called "wine_quality" that labels each wine as either good (quality score above 6) or bad (quality score 6 or below). We'll use logistic regression, a type of statistical model, to analyze this new classification problem and predict whether a wine is good or bad based on its features.

```
# Load the required libraries
library(caret)
```

```
## Loading required package: lattice
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
data <- df %>%
  mutate(wine_quality = ifelse(quality > 6, 1, 0)) %>%
  select(-quality)  # Remove the original quality column

# Perform logistic regression
logistic_model <- glm(wine_quality ~ ., data = data, family = binomial)
summary(logistic_model)
```

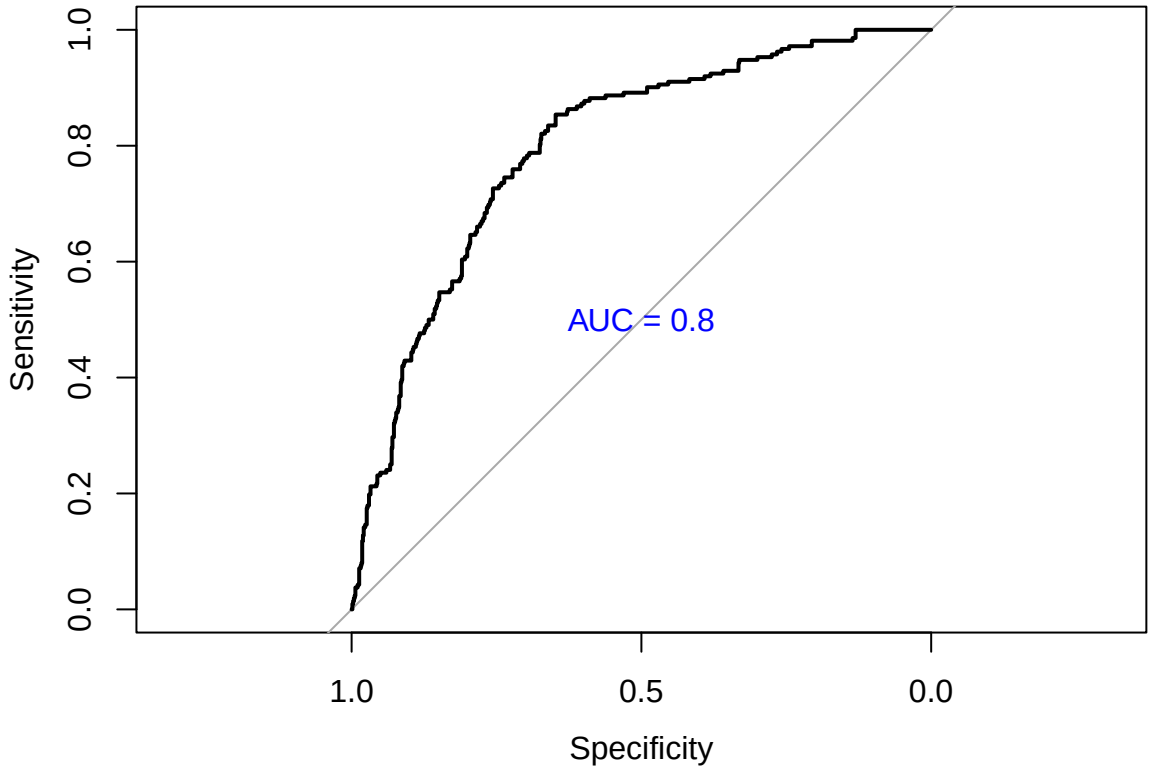
```
##
## Call:
## glm(formula = wine_quality ~ ., family = binomial, data = data)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)           6.362e+02  9.412e+01   6.759 1.39e-11 ***
## fixed.acidity         5.521e-01  9.053e-02   6.099 1.07e-09 ***
## volatile.acidity     -3.785e+00  4.885e-01  -7.749 9.28e-15 ***
## citric.acid          -7.378e-01  4.010e-01  -1.840 0.065776 .
## residual.sugar        2.952e-01  3.564e-02   8.283  < 2e-16 ***
## chlorides            -1.264e+01  3.816e+00  -3.312 0.000926 ***
## free.sulfur.dioxide   8.645e-03  3.130e-03   2.762 0.005749 **
## total.sulfur.dioxide -2.696e-04  1.506e-03  -0.179 0.857936
## density              -6.591e+02  9.540e+01  -6.909 4.89e-12 ***
## pH                    3.343e+00  4.268e-01   7.832 4.81e-15 ***
## sulphates             2.168e+00  3.475e-01   6.238 4.42e-10 ***
## alcohol               1.423e-01  1.139e-01   1.250 0.211334
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5116.8  on 4897  degrees of freedom
## Residual deviance: 4143.2  on 4886  degrees of freedom
## AIC: 4167.2
##
## Number of Fisher Scoring iterations: 6
```

```
# Split the data into training and testing sets
set.seed(123)  #
train_index <- createDataPartition(data$wine_quality, p = 0.8, list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]

# Predict on the test data
predicted_probabilities <- predict(logistic_model, newdata = test_data, type = "response")
predicted_classes <- ifelse(predicted_probabilities > 0.5, 1, 0)

# Compute AUC
roc_obj <- roc(test_data$wine_quality, predicted_probabilities)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```
auc_value <- auc(roc_obj)

# Plot ROC curve with AUC value
plot(roc_obj)
text(0.5, 0.5, paste("AUC =", round(auc_value, 2)), adj = c(0.5, 0.5), col = "blue")
```




```
# Create confusion matrix
conf_matrix <- confusionMatrix(factor(predicted_classes), factor(test_data$wine_quality))
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 729 162
##          1  38  50
##
##                Accuracy : 0.7957
##                  95% CI : (0.7691, 0.8206)
##     No Information Rate : 0.7835
##     P-Value [Acc > NIR] : 0.1865
##
##                   Kappa : 0.2363
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9505
##             Specificity : 0.2358
##          Pos Pred Value : 0.8182
##          Neg Pred Value : 0.5682
##              Prevalence : 0.7835
##          Detection Rate : 0.7446
```

```
##    Detection Prevalence : 0.9101
##       Balanced Accuracy : 0.5932
##
##        'Positive' Class : 0
##
```

Observations and Inferences for Logistic Regression Model:

After modelling we got estimate of parameters, lets observe them one by one:

Coefficients: Intercept: The intercept coefficient indicates the log odds of the response variable being in the "good" category when all predictor variables are zero. In this case, the intercept is significant ($p < 0.001$), suggesting that even when all predictor variables are zero, there's a substantial probability of a wine being classified as "good."

Among the predictor variables, several coefficients are statistically significant at conventional levels (indicated by the asterisks **,** ,). These include fixed.acidity, volatile.acidity, residual.sugar, chlorides, free.sulfur.dioxide, density, pH, and sulphates.These significant coefficients suggest that changes in these predictors have a noticeable impact on the odds of a wine being classified as "good" or "bad".

coefficient for total sulfur dioxide is not statistically significant (p-value > 0.05). This suggests that this predictor may not contribute significantly to the model, similarly alcohol p-value is 0.211, suggesting its lesser impact on wine quality.

Model fit: The null deviance (5116.8) is like a baseline measure of how much unexplained variation there is when no predictors are used. The residual deviance (4143.2) is a measure of the unexplained variation when predictors are included in the model. If the residual deviance is lower than the null deviance, it suggests that the model with predictors does a better job of explaining the variation in the response variable compared to a model with no predictors.

Confusion Matrix:

- The confusion matrix provides a summary of the model's predictions compared to the actual values.
- There are 729 true negatives (TN), 162 false positives (FP), 38 false negatives (FN), and 50 true positives (TP).
- Sensitivity, also known as the true positive rate or recall, measures the proportion of actual positive cases correctly identified by the model. In this case, sensitivity is 0.9505, indicating that the model correctly identifies 95.05% of the actual positive cases (wines classified as "good").
- Specificity measures the proportion of actual negative cases correctly identified by the model. In this case, specificity is 0.2358, indicating that the model correctly identifies only 23.58% of the actual negative cases (wines classified as "bad").
- The AUC value of 0.8 suggests that the model has good discriminatory power in distinguishing between positive and negative cases.
- An AUC of 0.8 indicates that there is an 80% chance that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

Observations from confusion matrix:

- The model performs well in terms of sensitivity, correctly identifying the majority of positive cases.
- However, the model's specificity is relatively low, indicating that it struggles to accurately identify negative cases.

- The AUC value of 0.8 indicates that the model has good discriminatory power overall, but there may be room for improvement, particularly in terms of specificity.
- The imbalance in the confusion matrix (higher count of true negatives compared to true positives) suggests that the model may be biased towards predicting negative cases.
- While the model shows good performance in terms of sensitivity and AUC, there is a clear trade-off with specificity.

Overall model performance: In comparison to linear regression, logistic regression provides far better performance in this case. Based on the significant coefficients and the model fit statistics, it seems like the logistic regression model developed has good overall performance. This suggests that the predictors included in the model are meaningful in predicting whether a wine is good or bad, and the model fits the data well.

Conclusion

In this Wine data analysis, we began by delving into the dataset's characteristics through univariate and bivariate analyses, exploring distributions and relationships between features. Initial attempts with simple linear regression, associating each feature individually with wine quality, yielded unsatisfactory results, highlighting the complexity of the relationship.

Moving forward, we embraced multiple linear regression to build a comprehensive model, aiming to capture the combined influence of multiple features on wine quality. Despite our efforts, achieving a high accuracy model remained elusive. Nonetheless, we meticulously conducted regression diagnostics to evaluate model assumptions and identify areas for improvement.

Subsequently, we approached the problem as a classification task, employing logistic regression. This methodology provided a significant improvement, demonstrating promising results in model accuracy. Our analysis encompassed the evaluation of ROC curves and confusion matrices, illuminating the model's discriminative abilities and its efficacy in correctly classifying wines.

In conclusion, logistic regression emerged as a viable approach, offering enhanced predictive capabilities compared to earlier models. While the current results are encouraging, there remains potential for further refinement and enhancement, potentially through the exploration of more sophisticated modeling techniques.