DSC 680 Project 3 Red Wine R

Christine Hathaway

May 10, 2020

```
#Set the working directory
setwd("C:/Users/Christine/Documents/Bellevue/DSC 680/Project 3")
```

Import data from file

```
# Read data to rw dataframe
# row.names = 1 to avoid an index column creation upon dataset reading into a dataframe
rw <- read.csv('C:/Users/Christine/Documents/Bellevue/DSC 680/Project 3/winequality-red.csv', sep = ';'
```

Display first five records of file

```
head(rw)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                   34  0.9978 3.51      0.56     9.4
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
## 5                  11                   34  0.9978 3.51      0.56     9.4
## 6                  13                   40  0.9978 3.51      0.56     9.4
##   quality
## 1       5
## 2       5
## 3       5
## 4       6
## 5       5
## 6       5
```

Find dimensions of rw dataframe

```
dim(rw)
```

```
## [1] 1599   12
```

List rw dataframe's column names, types and a subset of values

```
str(rw)
```

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

Display summary statistics for each variable

```
summary(rw)
```

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

Check how many missing values (NA) are in each column/variable, sum them up per column

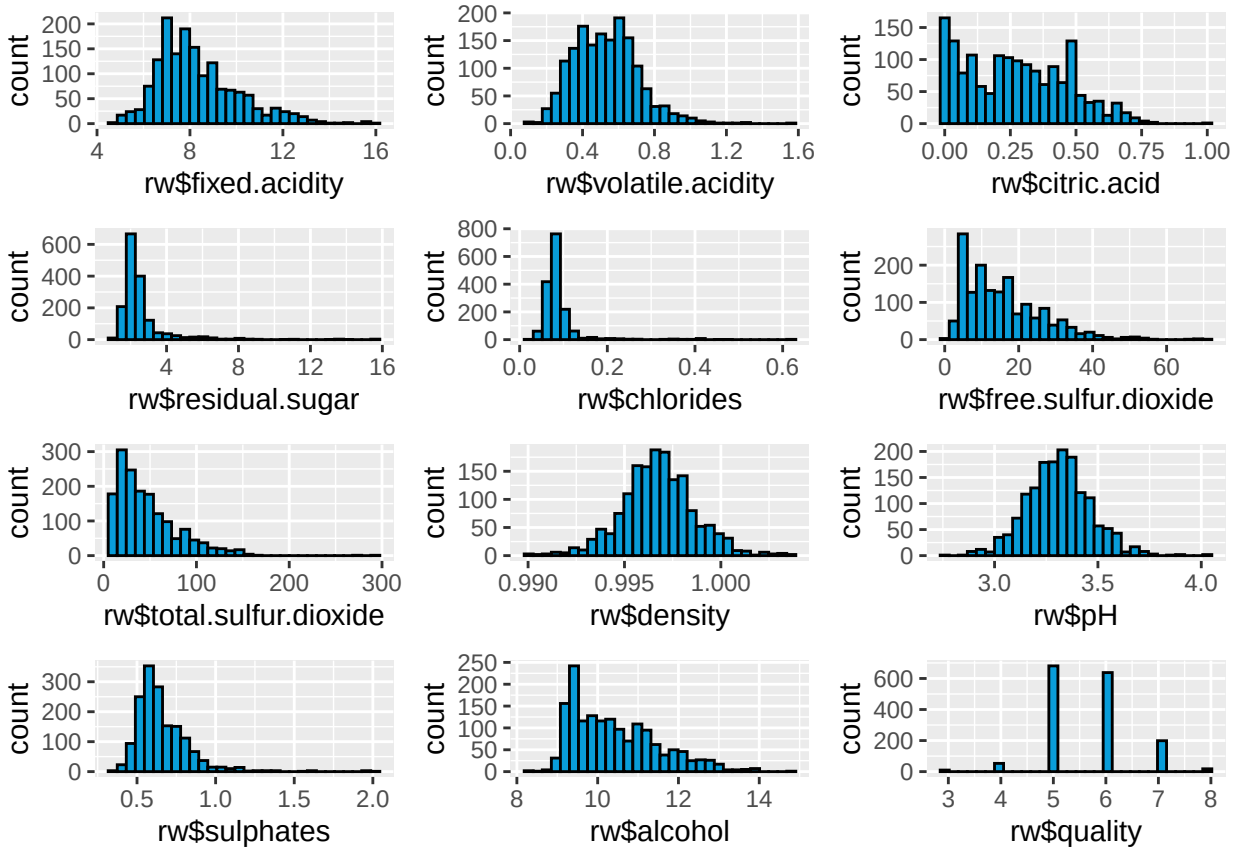
```
colSums(is.na(rw))
```

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

```
##                        0                      0                      0
##             sulphates                alcohol                quality
##                        0                      0                      0
```

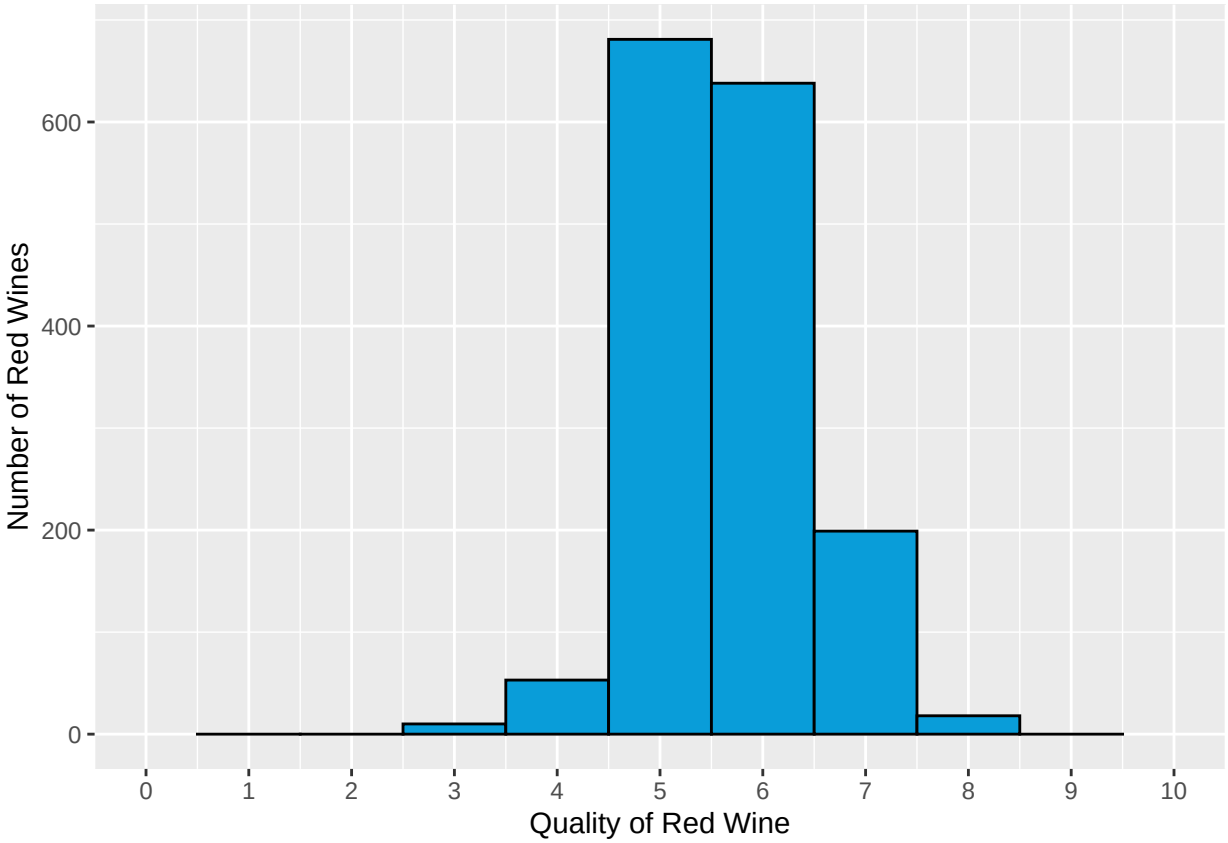
```r
# Draw a histogram for a given dataframe and variable
# Use deparse() and substitute() functions to decode column name from
# a variable passed as an argument to the function, to be displayed
# on x axis (xlab())
draw_hist <- function(dataframe, variable)
{
  # Save histogram definition to the plot variable
  plot <- ggplot(data = dataframe, aes(x = variable)) +
           geom_histogram(color = 'black', fill = '#099DD9') +
           xlab(deparse(substitute(variable)))
  return(plot)
}

# Build a matrix of small histograms with 3 columns
# using customly defined draw_hist() function
grid.arrange(draw_hist(rw, rw$fixed.acidity),
             draw_hist(rw, rw$volatile.acidity),
             draw_hist(rw, rw$citric.acid),
             draw_hist(rw, rw$residual.sugar),
             draw_hist(rw, rw$chlorides),
             draw_hist(rw, rw$free.sulfur.dioxide),
             draw_hist(rw, rw$total.sulfur.dioxide),
             draw_hist(rw, rw$density),
             draw_hist(rw, rw$pH),
             draw_hist(rw, rw$sulphates),
             draw_hist(rw, rw$alcohol),
             draw_hist(rw, rw$quality),
             ncol = 3)
```

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




```
# Plot a histogram of quality values
ggplot(data = rw, aes(x = quality)) +
  geom_histogram(color = 'black', fill = '#099DD9', binwidth = 1) +
  # Used to show 0-10 range, even if there are no values close to 0 or 10
  scale_x_continuous(limits = c(0, 10), breaks = seq(0, 10, 1)) +
  xlab('Quality of Red Wine') +
  ylab('Number of Red Wines')
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



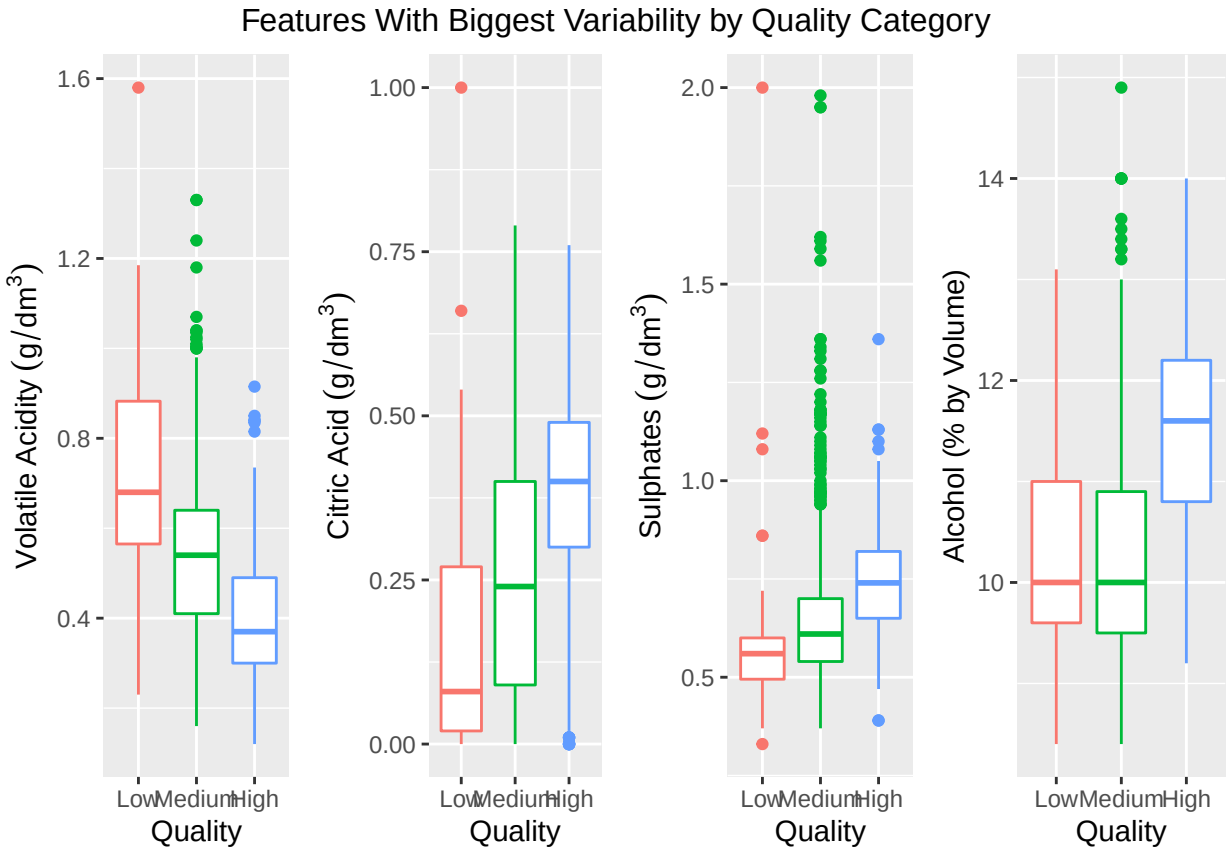

```
# Set boundaries for intervals
breaks <- c(0, 5, 7, 10)

# Bucket data points into intervals
rw$quality.category <- cut(rw$quality, breaks, include.lowest = TRUE, right = FALSE)

# Check intervals
summary(rw$quality.category)
```

```
##  [0,5)  [5,7) [7,10]
##     63   1319    217
```

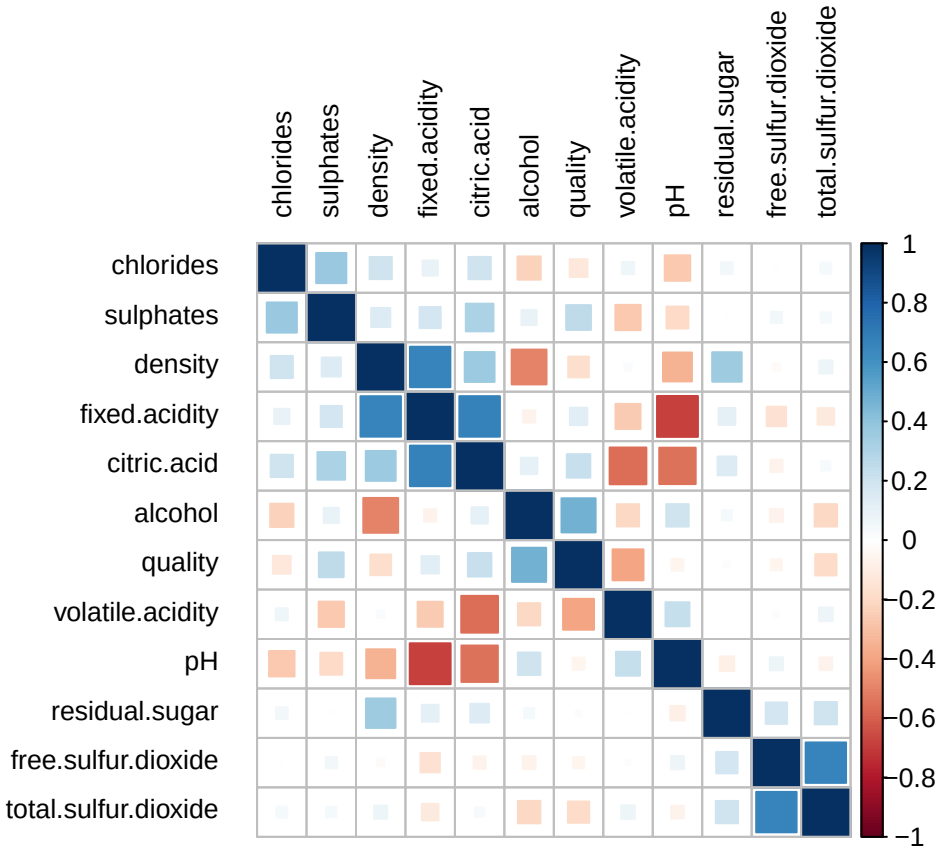
```
# Add labels to intervals
labels <- c("Low", "Medium", "High")
rw$quality.category <- cut(rw$quality, breaks, include.lowest = TRUE, right = FALSE, labels=labels)

# Check if labels are applied properly
table(rw$quality.category)
```

```
##
##    Low Medium   High
##     63   1319    217
```

```r
draw_boxplot <- function(dataframe, variable, ylab)
{
  plot <- ggplot(data = dataframe, aes(x = quality.category, y = variable, color = quality.category)) +
    geom_boxplot() +
    xlab('Quality') +
    #ylab(deparse(substitute(variable))) +
    ylab(ylab) +
    theme(legend.position = "none")
  return(plot)
}

# Build 4 boxplots summarizing distributions of 4 selected features
draw_univ_summary <- function()
{
  grid.arrange(draw_boxplot(rw, rw$volatile.acidity, expression(Volatile~Acidity~(g/dm^{3}))),
               draw_boxplot(rw, rw$citric.acid, expression(Citric~Acid~(g/dm^{3}))),
               draw_boxplot(rw, rw$sulphates, expression(Sulphates~(g/dm^{3}))),
               draw_boxplot(rw, rw$alcohol, 'Alcohol (% by Volume)'),
               ncol = 4,
               top = 'Features With Biggest Variability by Quality Category')
}

draw_univ_summary()
```



```r
# Create a new dataframe and calculate correlations
# between rw variables
rwcor <- cor(rw[c(1:11, 12)])
# Draw a correlation matrix
corrplot(rwcor, method = 'square', order = "hclust",
         tl.col = "black", tl.cex = 0.8, tl.offset = 1)
```



Regession models using binomial

```r
# create categorical variables

rw$category[rw$quality <= 5] <- 0
rw$category[rw$quality > 5] <- 1
rw$quality2 <- as.factor(rw$quality)

rw$category <- as.factor(rw$category)


head(rw)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
```

```
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                  34  0.9978 3.51      0.56     9.4
## 2                  25                  67  0.9968 3.20      0.68     9.8
## 3                  15                  54  0.9970 3.26      0.65     9.8
## 4                  17                  60  0.9980 3.16      0.58     9.8
## 5                  11                  34  0.9978 3.51      0.56     9.4
## 6                  13                  40  0.9978 3.51      0.56     9.4
##   quality quality.category category quality2
## 1       5           Medium        0        5
## 2       5           Medium        0        5
## 3       5           Medium        0        5
## 4       6           Medium        1        6
## 5       5           Medium        0        5
## 6       5           Medium        0        5
```

Split data into Train Test sets

```
set.seed(3000)

spl = sample.split(rw$category, SplitRatio = 0.7)

rwtrain = subset(rw, spl==TRUE)
rwtest = subset(rw, spl==FALSE)

head(rwtrain)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 2           7.8             0.88        0.00            2.6     0.098
## 4          11.2             0.28        0.56            1.9     0.075
## 6           7.4             0.66        0.00            1.8     0.075
## 7           7.9             0.60        0.06            1.6     0.069
## 8           7.3             0.65        0.00            1.2     0.065
## 9           7.8             0.58        0.02            2.0     0.073
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 2                  25                  67  0.9968 3.20      0.68     9.8
## 4                  17                  60  0.9980 3.16      0.58     9.8
## 6                  13                  40  0.9978 3.51      0.56     9.4
## 7                  15                  59  0.9964 3.30      0.46     9.4
## 8                  15                  21  0.9946 3.39      0.47    10.0
## 9                   9                  18  0.9968 3.36      0.57     9.5
##   quality quality.category category quality2
## 2       5           Medium        0        5
## 4       6           Medium        1        6
## 6       5           Medium        0        5
## 7       5           Medium        0        5
## 8       7             High        1        7
## 9       7             High        1        7
```

Create model

```r
model_glm <- glm(category ~ . - quality - quality2, data = rwtrain, family=binomial(link = "logit"))
```

Stepwise model

```r
model_gl <- step(model_glm)
```

```
## Start:  AIC=1027.09
## category ~ (fixed.acidity + volatile.acidity + citric.acid +
##     residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol + quality + quality.category +
##     quality2) - quality - quality2
##
##                        Df Deviance    AIC
## - pH                    1   999.74 1025.7
## - density               1  1000.67 1026.7
## <none>                      999.09 1027.1
## - chlorides             1  1001.49 1027.5
## - fixed.acidity         1  1001.59 1027.6
## - free.sulfur.dioxide   1  1001.61 1027.6
## - citric.acid           1  1001.90 1027.9
## - residual.sugar        1  1003.19 1029.2
## - sulphates             1  1021.74 1047.7
## - total.sulfur.dioxide  1  1021.86 1047.9
## - alcohol               1  1027.04 1053.0
## - volatile.acidity      1  1030.75 1056.8
## - quality.category      2  1118.48 1142.5
##
## Step:  AIC=1025.74
## category ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + sulphates + alcohol + quality.category
##
##                        Df Deviance    AIC
## - density               1  1000.67 1024.7
## <none>                      999.74 1025.7
## - fixed.acidity         1  1001.90 1025.9
## - citric.acid           1  1002.42 1026.4
## - free.sulfur.dioxide   1  1002.71 1026.7
## - chlorides             1  1002.97 1027.0
## - residual.sugar        1  1003.20 1027.2
## - sulphates             1  1021.74 1045.7
## - total.sulfur.dioxide  1  1025.20 1049.2
## - volatile.acidity      1  1031.07 1055.1
## - alcohol               1  1044.50 1068.5
## - quality.category      2  1118.48 1140.5
##
## Step:  AIC=1024.67
## category ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     sulphates + alcohol + quality.category
##
##                        Df Deviance    AIC
## - fixed.acidity         1   1001.9 1023.9
```

```
## <none>                      1000.7 1024.7
## - residual.sugar        1   1003.2 1025.2
## - citric.acid           1   1003.4 1025.3
## - chlorides             1   1003.6 1025.6
## - free.sulfur.dioxide   1   1003.7 1025.7
## - sulphates             1   1021.7 1043.7
## - total.sulfur.dioxide  1   1026.0 1048.0
## - volatile.acidity      1   1034.2 1056.2
## - alcohol               1   1085.3 1107.3
## - quality.category      2   1120.8 1140.8
##
## Step:  AIC=1023.9
## category ~ volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     sulphates + alcohol + quality.category
##
##                         Df Deviance    AIC
## - citric.acid           1   1003.4 1023.4
## <none>                      1001.9 1023.9
## - residual.sugar        1   1004.5 1024.5
## - free.sulfur.dioxide   1   1005.0 1025.0
## - chlorides             1   1005.8 1025.8
## - sulphates             1   1023.6 1043.6
## - total.sulfur.dioxide  1   1031.0 1051.0
## - volatile.acidity      1   1034.4 1054.4
## - alcohol               1   1085.8 1105.8
## - quality.category      2   1124.9 1142.9
##
## Step:  AIC=1023.35
## category ~ volatile.acidity + residual.sugar + chlorides + free.sulfur.dioxide +
##     total.sulfur.dioxide + sulphates + alcohol + quality.category
##
##                         Df Deviance    AIC
## - residual.sugar        1   1005.3 1023.3
## <none>                      1003.4 1023.4
## - free.sulfur.dioxide   1   1007.8 1025.8
## - chlorides             1   1008.7 1026.7
## - sulphates             1   1024.3 1042.3
## - total.sulfur.dioxide  1   1035.6 1053.6
## - volatile.acidity      1   1041.1 1059.1
## - alcohol               1   1086.2 1104.2
## - quality.category      2   1125.2 1141.2
##
## Step:  AIC=1023.34
## category ~ volatile.acidity + chlorides + free.sulfur.dioxide +
##     total.sulfur.dioxide + sulphates + alcohol + quality.category
##
##                         Df Deviance    AIC
## <none>                      1005.3 1023.3
## - chlorides             1   1010.0 1026.0
## - free.sulfur.dioxide   1   1010.9 1026.9
## - sulphates             1   1025.3 1041.3
## - total.sulfur.dioxide  1   1036.3 1052.3
## - volatile.acidity      1   1043.1 1059.1
```

```
## - alcohol               1   1090.0 1106.0
## - quality.category      2   1127.4 1141.4
```

```
head(fitted(model_gl))
```

```
##         2         4         6         7         8         9
## 0.1721425 0.5122556 0.2379190 0.1831529 1.0000000 0.9999999
```

```
head(predict(model_gl))
```

```
##           2           4           6           7           8           9
## -1.57051835  0.04903217 -1.16412280 -1.49513042 16.86038629 16.78646357
```

```
head(predict(model_gl, type = "response"))
```

```
##         2         4         6         7         8         9
## 0.1721425 0.5122556 0.2379190 0.1831529 1.0000000 0.9999999
```

Categorize wine

```
trn_pred <- ifelse(predict(model_gl, type = "response") > 0.5,"Good Wine", "Bad Wine")
head(trn_pred)
```

```
##           2           4           6           7           8           9
##  "Bad Wine" "Good Wine"  "Bad Wine"  "Bad Wine" "Good Wine" "Good Wine"
```

Confusion matrix

```
trn_tab <- table(predicted = trn_pred, actual = rwtrain$category)
trn_tab
```

```
##            actual
## predicted     0   1
##   Bad Wine  410 145
##   Good Wine 111 453
```

Checking accuracy of the training set.

```
sum(diag(trn_tab))/length(rwtrain$category)
```

```
## [1] 0.7712243
```

Confusion matrix for the test data.

```
# Making predictions on the test set.
tst_pred <- ifelse(predict(model_gl, newdata = rwtest, type = "response") > 0.5, "Good Wine", "Bad Wine
tst_tab <- table(predicted = tst_pred, actual = rwtest$category)
tst_tab
```

```
##            actual
## predicted     0   1
##   Bad Wine  171  68
##   Good Wine  52 189
```

Checking accuracy for the test data.

```
sum(diag(tst_tab))/length(rwtest$category)
```

```
## [1] 0.75
```