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

Display first five records of file

```
head(ww)
```

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

Find dimensions of ww dataframe

```
dim(ww)
```

```
## [1] 4898   12
```

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

```
str(ww)
```

```
## '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 ...
```

Display summary statistics for each variable

```
summary(ww)
```

```
##  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
```

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

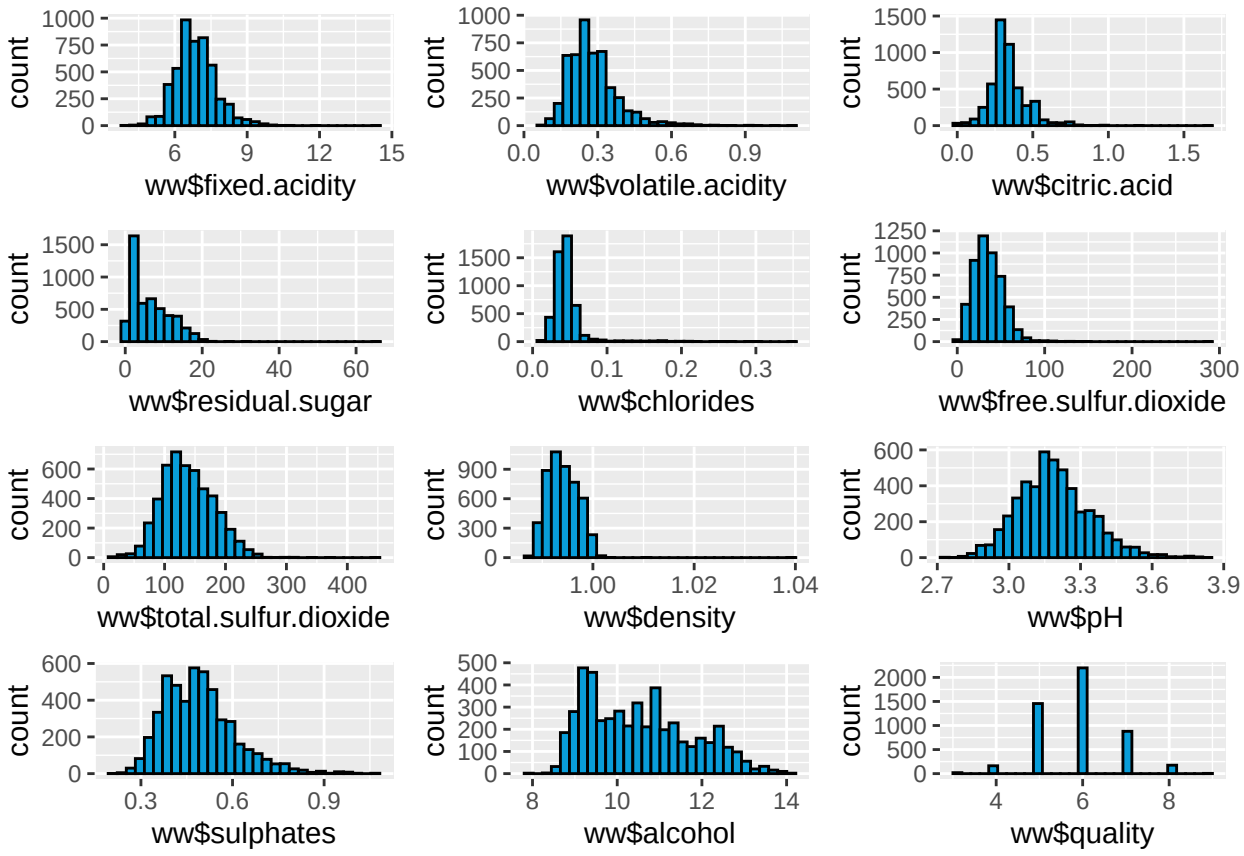
```
colSums(is.na(ww))
```

```
##        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
```

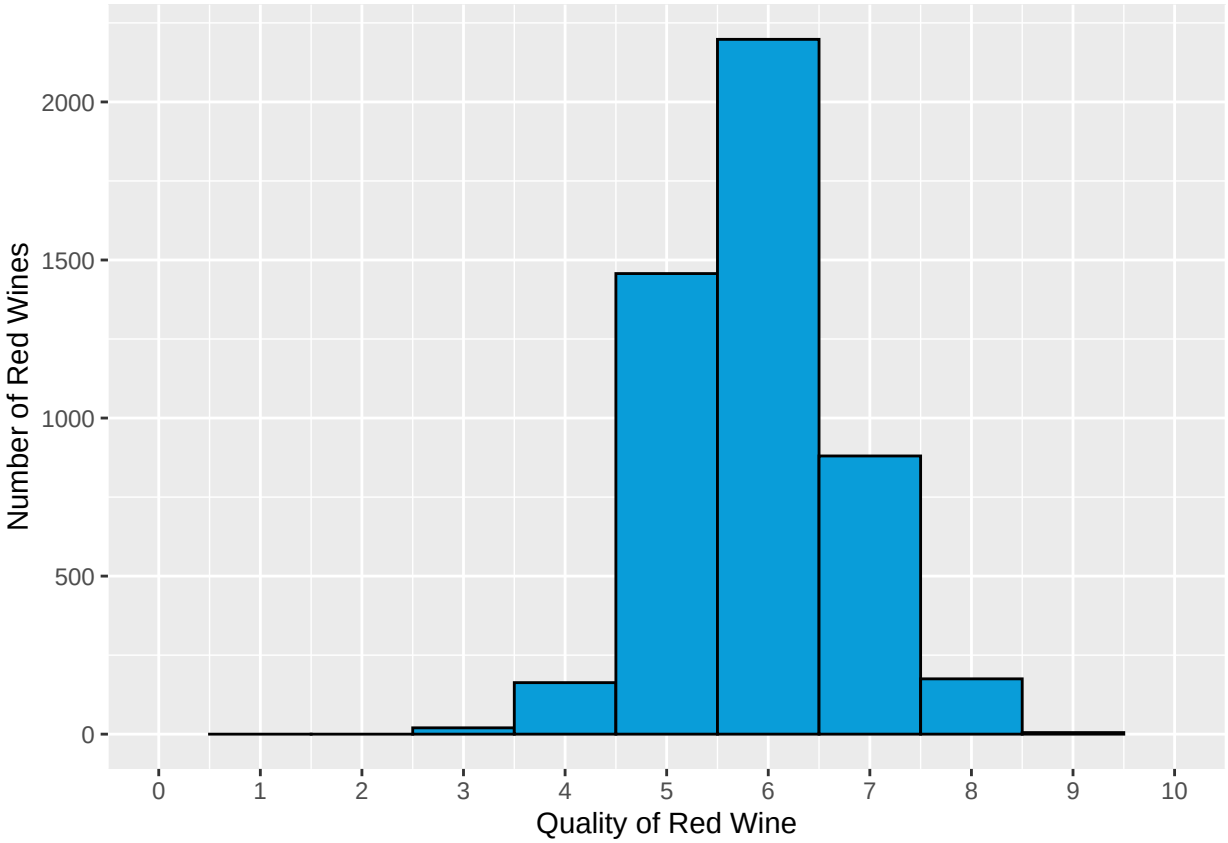
```r
# Draw a histogram for a given dataframe and variable
# Use deparse() and substitute() functions to decode column name from
# a variable passed as an argument to the function, to be displayed
# on x axis (xlab())
draw_hist <- function(dataframe, variable)
{
  # Save histogram definition to the plot 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(ww, ww$fixed.acidity),
             draw_hist(ww, ww$volatile.acidity),
             draw_hist(ww, ww$citric.acid),
             draw_hist(ww, ww$residual.sugar),
             draw_hist(ww, ww$chlorides),
             draw_hist(ww, ww$free.sulfur.dioxide),
             draw_hist(ww, ww$total.sulfur.dioxide),
             draw_hist(ww, ww$density),
             draw_hist(ww, ww$pH),
             draw_hist(ww, ww$sulphates),
             draw_hist(ww, ww$alcohol),
             draw_hist(ww, ww$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 = ww, 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).
```



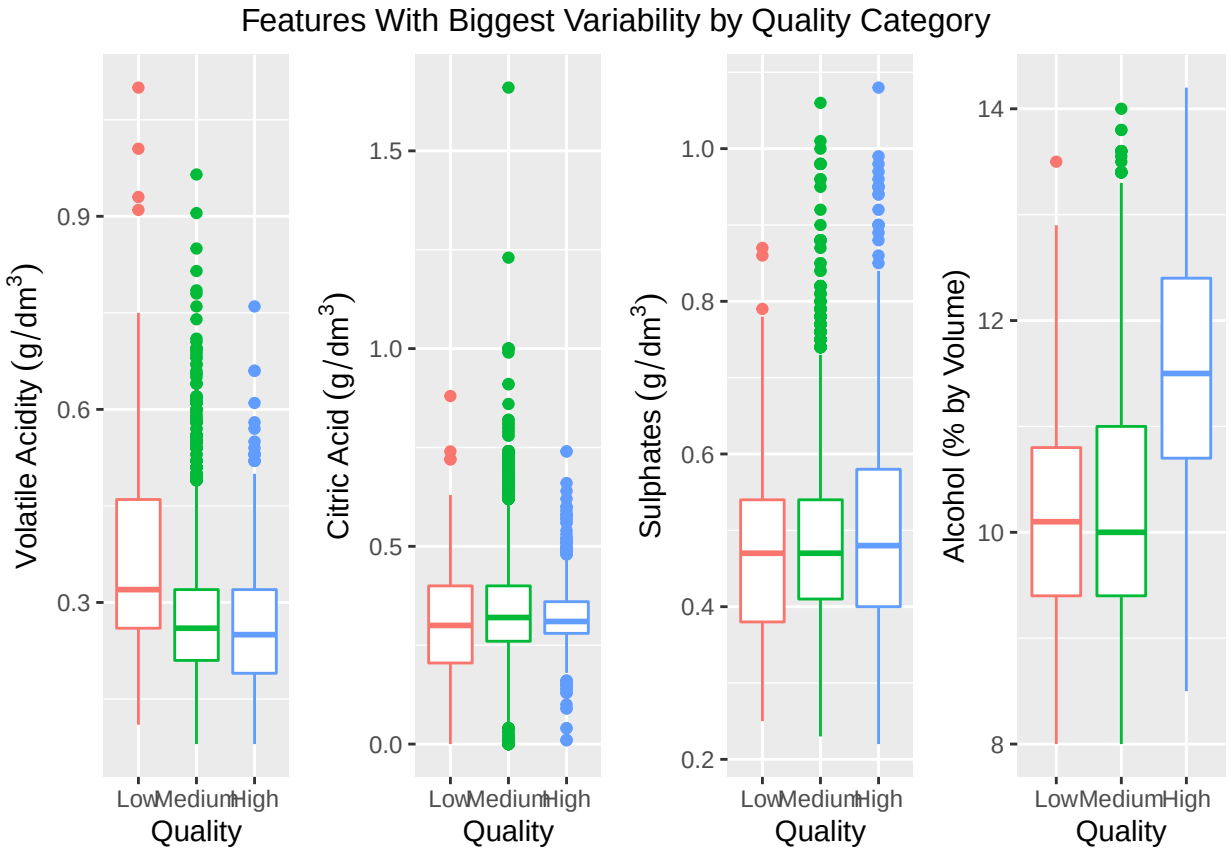

```
# Set boundaries for intervals
breaks <- c(0, 5, 7, 10)

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

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

```
##  [0,5)  [5,7) [7,10]
##    183   3655   1060
```

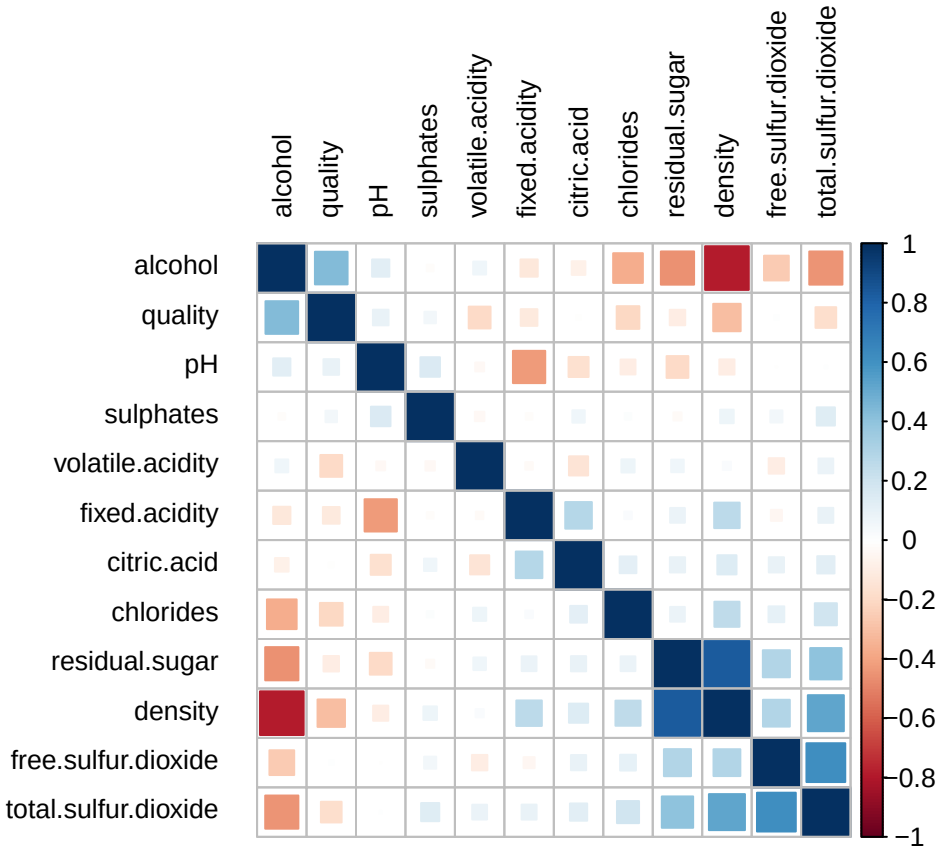
```
# Add labels to intervals
labels <- c("Low", "Medium", "High")
ww$quality.category <- cut(ww$quality, breaks, include.lowest = TRUE, right = FALSE, labels=labels)

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

```
##
##    Low Medium   High
##    183   3655   1060
```

```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(ww, ww$volatile.acidity, expression(Volatile~Acidity~(g/dm^{3}))),
               draw_boxplot(ww, ww$citric.acid, expression(Citric~Acid~(g/dm^{3}))),
               draw_boxplot(ww, ww$sulphates, expression(Sulphates~(g/dm^{3}))),
               draw_boxplot(ww, ww$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 ww variables
wwcor <- cor(ww[c(1:11, 12)])
# Draw a correlation matrix
corrplot(wwcor, method = 'square', order = "hclust",
         tl.col = "black", tl.cex = 0.8, tl.offset = 1)
```



Regession models using binomial

```r
# create categorical variables

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

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


head(ww)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.0             0.27        0.36           20.7     0.045
## 2           6.3             0.30        0.34            1.6     0.049
## 3           8.1             0.28        0.40            6.9     0.050
## 4           7.2             0.23        0.32            8.5     0.058
```

```
## 5           7.2             0.23        0.32            8.5     0.058
## 6           8.1             0.28        0.40            6.9     0.050
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  45                 170  1.0010 3.00      0.45     8.8
## 2                  14                 132  0.9940 3.30      0.49     9.5
## 3                  30                  97  0.9951 3.26      0.44    10.1
## 4                  47                 186  0.9956 3.19      0.40     9.9
## 5                  47                 186  0.9956 3.19      0.40     9.9
## 6                  30                  97  0.9951 3.26      0.44    10.1
##   quality quality.category category quality2
## 1       6           Medium        1        6
## 2       6           Medium        1        6
## 3       6           Medium        1        6
## 4       6           Medium        1        6
## 5       6           Medium        1        6
## 6       6           Medium        1        6
```

Split data into Train Test sets

```
set.seed(3000)

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

wwtrain = subset(ww, spl==TRUE)
wwtest = subset(ww, spl==FALSE)

head(wwtrain)
```

```
##    fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 2            6.3             0.30        0.34            1.6     0.049
## 5            7.2             0.23        0.32            8.5     0.058
## 6            8.1             0.28        0.40            6.9     0.050
## 8            7.0             0.27        0.36           20.7     0.045
## 9            6.3             0.30        0.34            1.6     0.049
## 10           8.1             0.22        0.43            1.5     0.044
##    free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 2                   14                 132  0.9940 3.30      0.49     9.5
## 5                   47                 186  0.9956 3.19      0.40     9.9
## 6                   30                  97  0.9951 3.26      0.44    10.1
## 8                   45                 170  1.0010 3.00      0.45     8.8
## 9                   14                 132  0.9940 3.30      0.49     9.5
## 10                  28                 129  0.9938 3.22      0.45    11.0
##    quality quality.category category quality2
## 2        6           Medium        1        6
## 5        6           Medium        1        6
## 6        6           Medium        1        6
## 8        6           Medium        1        6
## 9        6           Medium        1        6
## 10       6           Medium        1        6
```

Create model

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

Stepwise model

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

```
## Start:  AIC=2985.57
## 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
## - chlorides             1   2957.6 2983.6
## - citric.acid           1   2957.8 2983.8
## - fixed.acidity         1   2957.8 2983.8
## - total.sulfur.dioxide  1   2958.0 2984.0
## - pH                    1   2958.1 2984.1
## <none>                      2957.6 2985.6
## - density               1   2960.9 2986.9
## - free.sulfur.dioxide   1   2963.8 2989.8
## - sulphates             1   2965.3 2991.3
## - residual.sugar        1   2970.4 2996.4
## - alcohol               1   3003.0 3029.0
## - volatile.acidity      1   3046.7 3072.7
## - quality.category      2   3475.8 3499.8
##
## Step:  AIC=2983.57
## category ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     free.sulfur.dioxide + total.sulfur.dioxide + density + pH +
##     sulphates + alcohol + quality.category
##
##                        Df Deviance    AIC
## - citric.acid           1   2957.8 2981.8
## - fixed.acidity         1   2957.9 2981.9
## - total.sulfur.dioxide  1   2958.0 2982.0
## - pH                    1   2958.1 2982.1
## <none>                      2957.6 2983.6
## - density               1   2961.0 2985.0
## - free.sulfur.dioxide   1   2963.9 2987.9
## - sulphates             1   2965.3 2989.3
## - residual.sugar        1   2970.9 2994.9
## - alcohol               1   3003.1 3027.1
## - volatile.acidity      1   3047.2 3071.2
## - quality.category      2   3475.8 3497.8
##
## Step:  AIC=2981.81
## category ~ fixed.acidity + volatile.acidity + residual.sugar +
##     free.sulfur.dioxide + total.sulfur.dioxide + density + pH +
##     sulphates + alcohol + quality.category
##
##                        Df Deviance    AIC
## - fixed.acidity         1   2958.1 2980.1
```

```
## - total.sulfur.dioxide  1   2958.3 2980.3
## - pH                    1   2958.3 2980.3
## <none>                      2957.8 2981.8
## - density               1   2961.1 2983.1
## - free.sulfur.dioxide   1   2964.3 2986.3
## - sulphates             1   2965.6 2987.6
## - residual.sugar        1   2971.0 2993.0
## - alcohol               1   3004.2 3026.2
## - volatile.acidity      1   3050.1 3072.1
## - quality.category      2   3476.1 3496.1
##
## Step:  AIC=2980.05
## category ~ volatile.acidity + residual.sugar + free.sulfur.dioxide +
##     total.sulfur.dioxide + density + pH + sulphates + alcohol +
##     quality.category
##
##                         Df Deviance    AIC
## - total.sulfur.dioxide  1   2958.5 2978.5
## - pH                    1   2959.7 2979.7
## <none>                      2958.1 2980.1
## - free.sulfur.dioxide   1   2964.5 2984.5
## - density               1   2966.1 2986.1
## - sulphates             1   2966.2 2986.2
## - residual.sugar        1   2983.4 3003.4
## - alcohol               1   3038.1 3058.1
## - volatile.acidity      1   3050.9 3070.9
## - quality.category      2   3476.1 3494.1
##
## Step:  AIC=2978.5
## category ~ volatile.acidity + residual.sugar + free.sulfur.dioxide +
##     density + pH + sulphates + alcohol + quality.category
##
##                        Df Deviance    AIC
## - pH                    1   2960.1 2978.1
## <none>                      2958.5 2978.5
## - free.sulfur.dioxide   1   2965.7 2983.7
## - sulphates             1   2966.2 2984.2
## - density               1   2968.0 2986.0
## - residual.sugar        1   2985.8 3003.8
## - alcohol               1   3038.1 3056.1
## - volatile.acidity      1   3059.1 3077.1
## - quality.category      2   3476.2 3492.2
##
## Step:  AIC=2978.1
## category ~ volatile.acidity + residual.sugar + free.sulfur.dioxide +
##     density + sulphates + alcohol + quality.category
##
##                        Df Deviance    AIC
## <none>                      2960.1 2978.1
## - free.sulfur.dioxide   1   2967.5 2983.5
## - density               1   2968.2 2984.2
## - sulphates             1   2968.4 2984.4
## - residual.sugar        1   2985.9 3001.9
## - alcohol               1   3053.4 3069.4
```

```
## - volatile.acidity     1   3062.5 3078.5
## - quality.category     2   3482.1 3496.1
```

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

```
##         2         5         6         8         9        10
## 0.3055928 0.6111362 0.5462353 0.4965432 0.3055928 0.6881878
```

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

```
##           2           5           6           8           9          10
## -0.82080506  0.45209071  0.18547122 -0.01382746 -0.82080506  0.79166055
```

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

```
##         2         5         6         8         9        10
## 0.3055928 0.6111362 0.5462353 0.4965432 0.3055928 0.6881878
```

Categorize wine

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

```
##           2           5           6           8           9          10
##  "Bad Wine" "Good Wine" "Good Wine"  "Bad Wine"  "Bad Wine" "Good Wine"
```

Confusion matrix

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

```
##            actual
## predicted      0    1
##   Bad Wine   664  295
##   Good Wine  484 1986
```

Checking accuracy of the training set.

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

```
## [1] 0.7728201
```

Confusion matrix for the test data.

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

```
##            actual
## predicted     0   1
##   Bad Wine  294 120
##   Good Wine 198 857
```

Checking accuracy for the test data.

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

```
## [1] 0.7835262
```