435_final_project

Tiana Noll-Walker

2023-12-05

Project: Wine Quality Prediction

1. Problem Setup:

• Chose the Wine Quality dataset, both red and white wines • Define a problem: Predict the quality of wine based on physio chemical attributes. • Statistical Question: What statistical model can accurately predict the wine quality score?

2. Implementation:

• Used SVM, linear regression, and logistic regression • Evaluated model performance using metrics like mean squared error, R-squared, F1 score, and MSE. Then used cross validation and ROC curves to evaluate models.

3. Depth:

• Explored the importance of different physio chemical attributes in predicting wine quality. • Considered feature selection methods to identify relevant input variables. • Implemented and compared various algorithms.

4. Model Evaluations and Comparisons:

I chose 3 models: linear regression, logistic regression, and SVM. I chose linear regression because it is often used when the target variable is continuous, which made it a good choice for wine quality scores. Linear regression gives coefficients for each predictor, which lets you interpret the impact of each feature on the target variable. This is valuable to understand the relationships between individual physiochemical attributes and wine quality.

Logistic regression is a binary classification algorithm, usually used when the target variable is categorical with two classes. In this project, I transformed the problem into a binary classification task, and predicted whether the wine quality was greater than 5 or not. Logistic regression coefficients represent the log odds of the target variable being in a particular class. The interpretation of coefficients helps identify which features contribute positively or negatively to the likelihood of the wine quality being greater than 5.

SVM is a versatile algorithm that can be used for both regression and classification. It's especially useful when dealing with complex relationships and high dimensional data. SVM can capture nonlinear relationships between features and the target variable by using different kernels. This is beneficial when the relationships between physiochemical attributes and whine quality isn't linear.

Linear Regression Model Summary Red Wine:

• Residual Standard Error: This is an estimate of the standard deviation of the residuals, approximately 0.6441. Smaller values indicate a better fit of the model to the data. • Multiple R-squared: The model explains around 36.62% of the variability in red wine quality. • Adjusted R-squared: A modified R-squared considering the number of predictors in the model, standing at 35.99. • F-statistic: With a p-value < 2.2e-16, the model is statistically significant. • Red Wine MSE: 0.43545

Logistic Regression Model Summary Red Wine:

• Accuracy: The model correctly predicts whether the quality is greater than 5 about 72.86% of the time. • Precision: About 75.82% of the instances predicted as quality greater than 5 are true positives. • Recall: Approximately 72.27% of actual instances with quality greater than 5 are correctly identified by the model. • F1 Score: The harmonic mean of precision and recall is 74%. • Red Wine MSE: 25.345167

Linear Regression Interpretation Red Wine:

• The negative t-value and low p-value for "Volatile Acidity" suggest that higher volatile acidity is associated with lower wine quality. • Other predictors with low p-values (e.g., "Residual Sugar," "Free Sulfur Dioxide," "Total Sulfur Dioxide," "Alcohol") are likely to be significant predictors of red wine quality.

Logistic Regression Red Wine Interpretation:

• Volatile Acidity: A negative coefficient indicates that higher volatile acidity is associated with a lower likelihood of red wine quality being greater than 5. • Free Sulfur Dioxide, Total Sulfur Dioxide, Alcohol: Positive coefficients suggest that higher values of these features are associated with a higher likelihood of red wine quality being greater than 5. • Chlorides, pH, Density: Negative coefficients suggest that higher values of these features are associated with a lower likelihood of red wine quality being greater than 5.

Linear Regression Model Summary White Wine:

• Residual Standard Error: This estimate of the standard deviation of the residuals is approximately 0.7488, indicating a reasonable fit of the model to the data. • Multiple R-squared: The model explains around 28.55% of the variability in white wine quality. • Adjusted R-squared: The adjusted R-squared, accounting for predictors, is 28.32. • F-statistic: The F-statistic with a p-value < 2.2e-16 underscores the model's statistical significance. • White Wine MSE: 0.576736

Logistic Regression Model Summary White Wine:

• Accuracy: The model correctly predicts whether the quality is greater than 5 about 73.32% of the time. • Precision: About 76.62% of the instances predicted as quality greater than 5 are true positives. • Recall: Approximately 86.18% of actual instances with quality greater than 5 are correctly identified by the model. • F1 Score: The harmonic mean of precision and recall is 81%. • White Wine MSE: 27.870901

Linear Regression Interpretation White Wine:

• The negative t-value and low p-value for "Volatile Acidity" suggest that higher volatile acidity is associated with lower white wine quality. • Other predictors with low p-values, such as "Residual Sugar," "Free Sulfur Dioxide," "Total Sulfur Dioxide," and "Alcohol," are likely significant predictors.

Logistic Regression White Wine Interpretation:

• Volatile Acidity: A negative coefficient indicates that higher volatile acidity is associated with a lower likelihood of white wine quality being greater than 5. • Free Sulfur Dioxide, Total Sulfur Dioxide, Alcohol: Positive coefficients suggest that higher values of these features are associated with a higher likelihood of white wine quality being greater than 5. • Chlorides, pH, Density: Negative coefficients suggest that higher values of these features are associated with a lower likelihood of white wine quality being greater than 5.

Support Vector Machine (SVM) Model:

For both red and white wines, a Support Vector Machine model was trained. The mean squared error was used to evaluate the model's performance. The MSE for the red wine SVM model was .381 and for the white wine SVM model was .482.

Cross-Validation:

Cross-validation was used to get a more robust estimate of model performance. The data was divided into 10 folds and the model was trained and evaluated 10 times, with each fold serving as the test set once. For the red wine model, the results of the cross-validation were as follows: • Root Mean Squared Error (RMSE): 0.6519379 • R-Squared: 0.3466255 • Mean Absolute Error (MAE): 0.5081582 For the white wine model, the results of the cross-validation were as follows: • Root Mean Squared Error (RMSE): 0.7483681 • R-Squared: 0.2868018 • Mean Absolute Error (MAE): 0.574295

These results provide a more reliable estimate of the model's performance because they average the performance over multiple different splits of the data. This helps to ensure that the performance estimate is not overly optimistic due to a particularly favorable split of the data.

ROC Curve:

The Receiver Operating Characteristic (ROC) curve was computed for the logistic regression models. The ROC curve is a plot of the true positive rate against the false positive rate, and it provides a useful way to evaluate the performance of a binary classifier. The area under the ROC curve (AUC) was also calculated, with a value closer to 1 indicating a better model. The ROC curve for the red wine model has an Area Under the Curve (AUC) of 0.857, indicating a very good model. The ROC curve for the white wine model has an AUC of 0.755, indicating a reasonably good model. These values suggest that both models are capable of distinguishing between wines with quality greater than 5 and those with quality less than or equal to 5.

Data Visualization:

Scatter plots of volatile.acidity vs quality and boxplots of alcohol by quality were created for both red and white wines. These visualizations provide insights into the relationships between these features and wine quality.

Feature Engineering:

Interaction terms were created between volatile.acidity and alcohol, and a log transformation was applied to residual.sugar. The interaction term captures the combined effect of volatile.acidity and alcohol on wine quality. Log transformation of residual.sugar is applied to handle skewness in the data. It's helpful when the distribution of the variable is right skewed to make it more symmetric.

Conclusion:

In conclusion, this project leveraged linear regression, logistic regression, and support vector machines to predict wine quality based on physical and chemical attributes. Through much evaluation, including mean squared error, ROC curves, and cross-validation, the models demonstrated robust performance. Key predictors, notably volatile acidity, alcohol, and residual sugar, were identified, and their impact on wine quality comprehensively analyzed. Feature engineering, including interaction terms and log transformations, enhanced model accuracy by capturing complex relationships. The project not only provides accurate predictive models but also offers valuable insights into the intricate interplay between physio chemical attributes and wine quality, showcasing a holistic approach to statistical modeling in the context of wine quality prediction.

```r
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(ggplot2)
library(olsrr)
```

```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':
##
##     rivers
```

```r
wine_data_red <- read.csv("/Users/tiananoll-walker/Documents/stat435/wine+quality/winequality-red.csv",

str(wine_data_red)
```

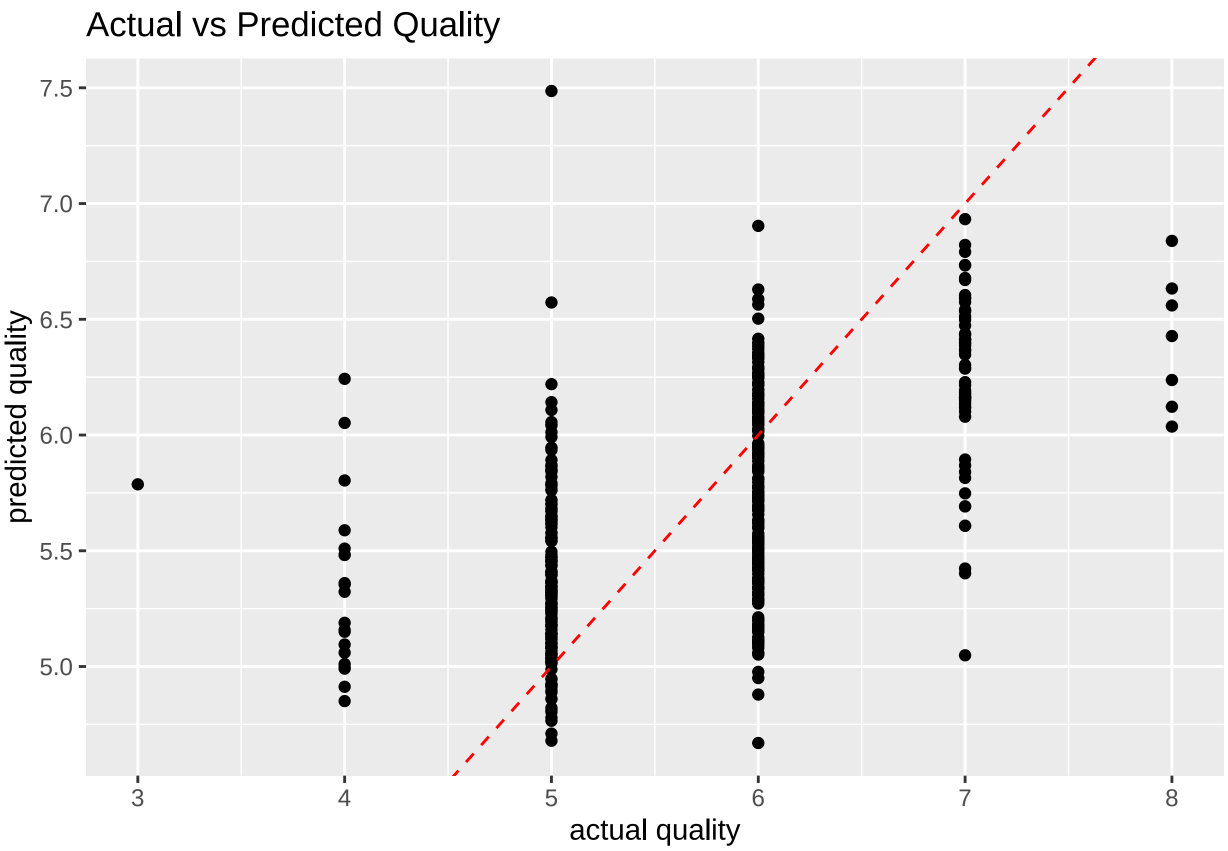
```
## 'data.frame':    1599 obs. of  12 variables:
##  $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : 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 ...
```

```r
#split into training and testing sets
set.seed(123)
train_index <- createDataPartition(wine_data_red$quality, p = 0.7, list = FALSE)
train_data_red <- wine_data_red[train_index, ]
test_data_red <- wine_data_red[-train_index, ]

# linear regression
model_red <- lm(quality ~ ., data = train_data_red)
summary(model_red)
```

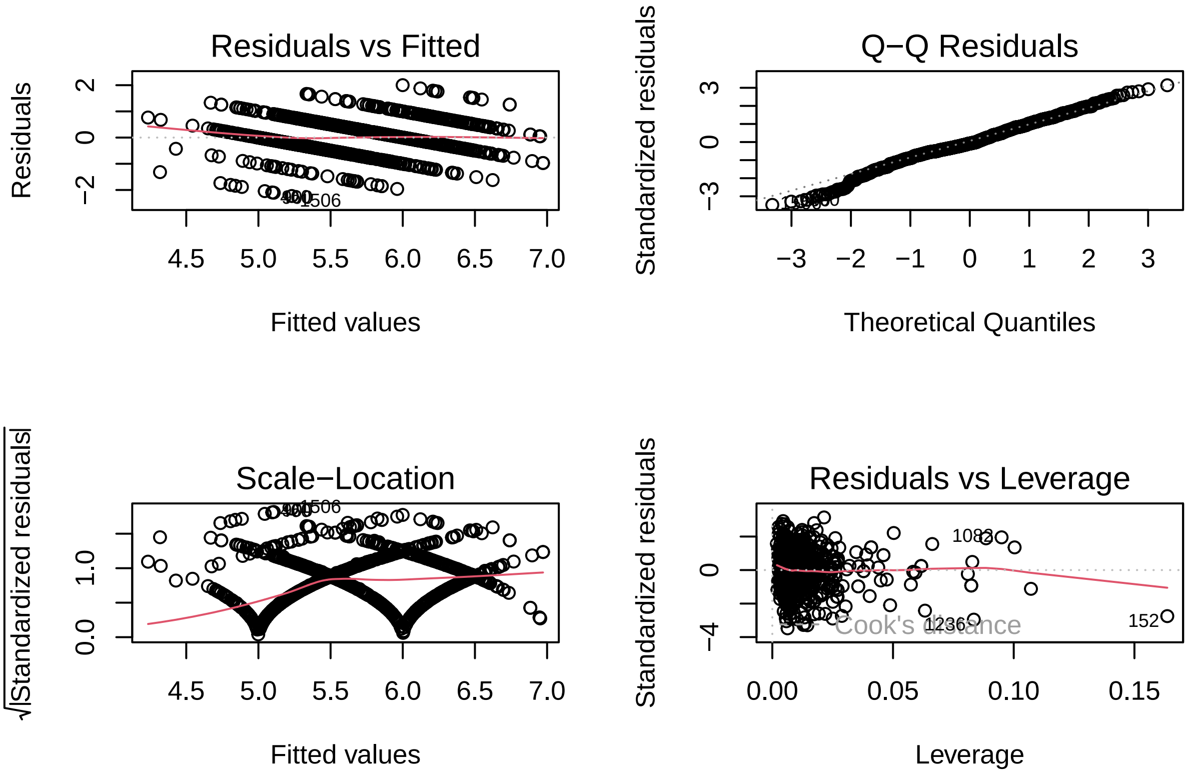
```
##
## Call:
## lm(formula = quality ~ ., data = train_data_red)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.23105 -0.35452 -0.05157  0.43806  2.00060
##
## Coefficients:
```

```
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            2.575e+01  2.530e+01   1.018 0.308978    
## fixed.acidity          4.708e-02  3.131e-02   1.504 0.132936    
## volatile.acidity      -1.125e+00  1.447e-01  -7.770 1.78e-14 ***
## citric.acid           -2.091e-01  1.799e-01  -1.163 0.245193    
## residual.sugar         4.439e-03  1.851e-02   0.240 0.810491    
## chlorides             -1.905e+00  5.478e-01  -3.477 0.000526 ***
## free.sulfur.dioxide    5.416e-03  2.568e-03   2.109 0.035170 *  
## total.sulfur.dioxide -2.864e-03  8.694e-04  -3.295 0.001017 ** 
## density               -2.255e+01  2.584e+01  -0.873 0.383079    
## pH                    -1.825e-01  2.281e-01  -0.800 0.423805    
## sulphates              9.595e-01  1.357e-01   7.070 2.73e-12 ***
## alcohol                2.678e-01  3.184e-02   8.411  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.6441 on 1108 degrees of freedom
## Multiple R-squared:  0.3662, Adjusted R-squared:  0.3599 
## F-statistic:  58.2 on 11 and 1108 DF,  p-value: < 2.2e-16
```

```r
#make predictions on test set
predictions_red <- predict(model_red, newdata = test_data_red)

mean_squared_error_red<-mean((test_data_red$quality - predictions_red)^2)
rsquared <- summary(model_red)$r.squared

#predicted vs actual wine quality scores
ggplot(data = test_data_red, aes(x = quality, y = predictions_red)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Actual vs Predicted Quality",
       x = "actual quality",
       y = "predicted quality")
```



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



#

I've trained a logistic regression model for red wine quality. I've calculated and displayed classification metrics like accuracy, precision, recall and F1 score for evaluation.

```
#logistic regression model
logistic_model_red <- glm(quality > 5 ~ ., data = train_data_red, family = binomial)
summary(logistic_model_red)
```

```
##
## Call:
## glm(formula = quality > 5 ~ ., family = binomial, data = train_data_red)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           41.566782  95.117298   0.437   0.6621
## fixed.acidity          0.159993   0.120092   1.332   0.1828
## volatile.acidity      -3.595677   0.604362  -5.950 2.69e-09 ***
## citric.acid           -1.307661   0.696922  -1.876   0.0606 .
## residual.sugar         0.036986   0.066901   0.553   0.5804
## chlorides             -3.509952   2.085513  -1.683   0.0924 .
## free.sulfur.dioxide    0.022272   0.009889   2.252   0.0243 *
## total.sulfur.dioxide  -0.016017   0.003454  -4.637 3.53e-06 ***
## density              -50.866491  97.217992  -0.523   0.6008
## pH                    -0.044936   0.870611  -0.052   0.9588
## sulphates              2.830951   0.541281   5.230 1.69e-07 ***
## alcohol                0.881228   0.125285   7.034 2.01e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1547.2  on 1119  degrees of freedom
## Residual deviance: 1143.6  on 1108  degrees of freedom
## AIC: 1167.6
##
## Number of Fisher Scoring iterations: 4
```

```
logistic_predictions_red <- predict(logistic_model_red, newdata = test_data_red, type = "response")

logistic_mse_red <- mean((test_data_red$quality - logistic_predictions_red)^2)

predicted_classes_red <- ifelse(logistic_predictions_red > 0.5, 1, 0)

# Convert predicted probabilities to class labels for red wine (e.g., 0 or 1)
predicted_classes_red <- ifelse(logistic_predictions_red > 0.5, 1, 0)
confusion_matrix_red <- table(test_data_red$quality > 5, predicted_classes_red)
accuracy_red <- sum(diag(confusion_matrix_red)) / sum(confusion_matrix_red)
precision_red <- confusion_matrix_red[2, 2] / sum(confusion_matrix_red[, 2])
recall_red <- confusion_matrix_red[2, 2] / sum(confusion_matrix_red[2, ])
f1_score_red <- 2 * (precision_red * recall_red) / (precision_red + recall_red)

print("Red Wine Classification Metrics:")
```

```
## [1] "Red Wine Classification Metrics:"
```

```
print(paste("Accuracy:", accuracy_red))
```

```
## [1] "Accuracy: 0.728601252609603"
```

```
print(paste("Precision:", precision_red))
```

```
## [1] "Precision: 0.758196721311475"
```

```
print(paste("Recall:", recall_red))
```

```
## [1] "Recall: 0.72265625"
```

```
print(paste("F1 Score:", f1_score_red))
```

```
## [1] "F1 Score: 0.74"
```

#I've trained a linear regression model for white wine quality and visualized the predicted vs actual wine quality score. The model, as indicated by the F-statistic and R-squared values, suggests that the set of predictor variables collectively has a significant impact on predicting white wine quality.

```
library(caret)
library(ggplot2)
library(olsrr)

wine_data_white <- read.csv("/Users/tiananoll-walker/Documents/stat435/wine+quality/winequality-white.c

str(wine_data_white)
```

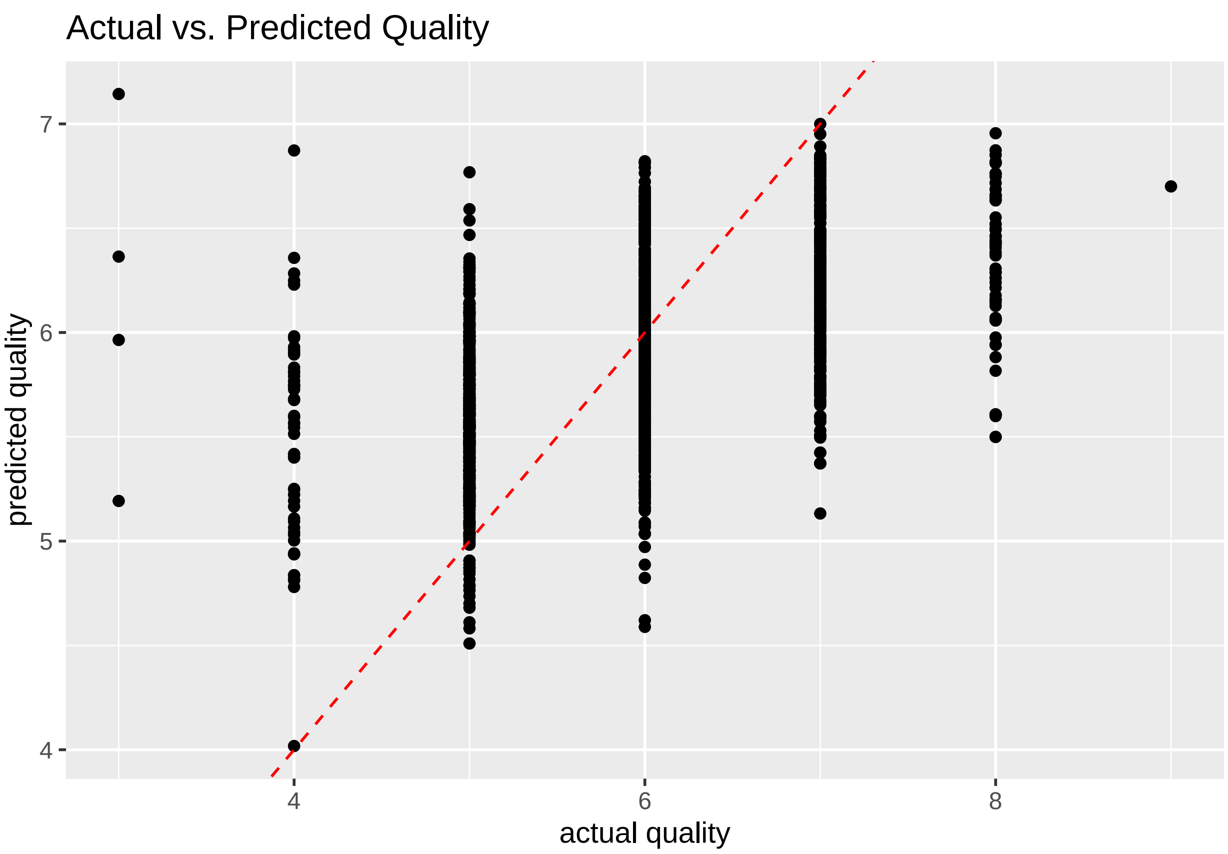
```
## 'data.frame':    4898 obs. of  12 variables:
##  $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
##  $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
##  $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
##  $ 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 ...
```

```
set.seed(123)
train_index <- createDataPartition(wine_data_white$quality, p = 0.7, list = FALSE)
train_data_white <- wine_data_white[train_index, ]
test_data_white <- wine_data_white[-train_index, ]

model_white <- lm(quality ~ ., data = train_data_white)

summary(model_white)
```

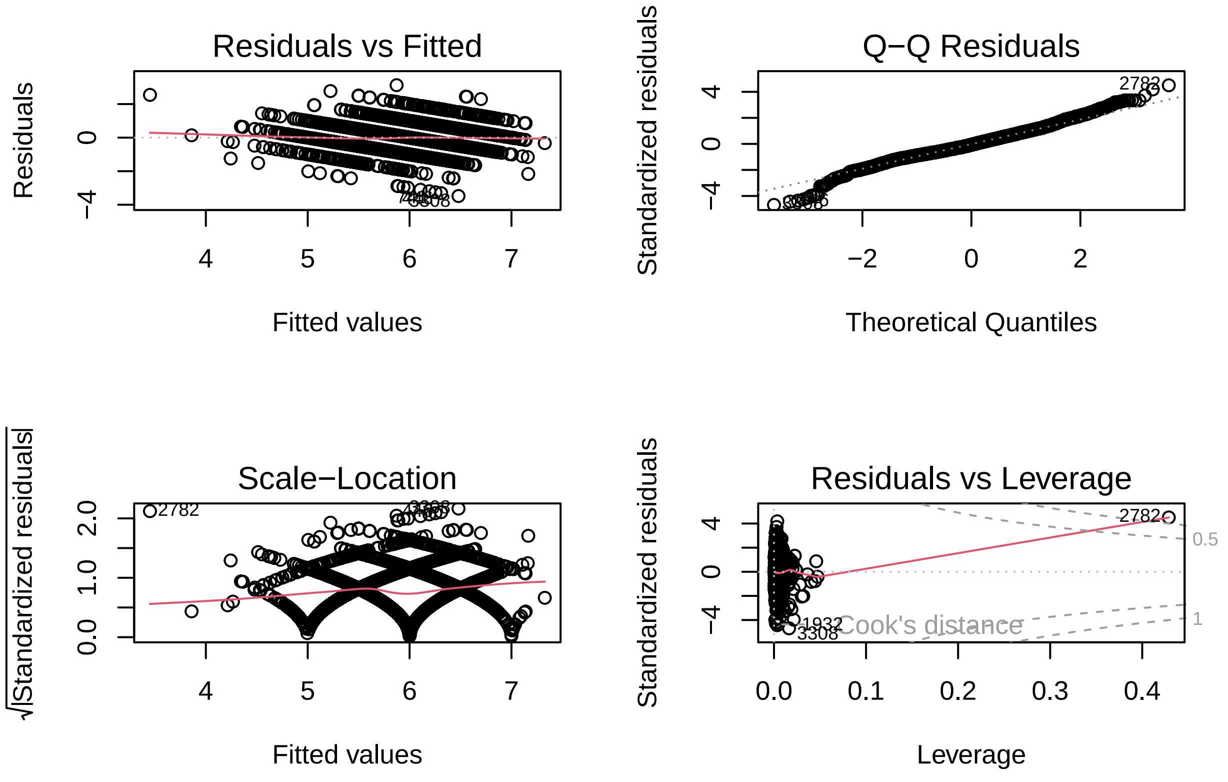
```
##
## Call:
## lm(formula = quality ~ ., data = train_data_white)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4802 -0.4905 -0.0488  0.4641  3.1275
##
## Coefficients:
```

```
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.298e+02  2.086e+01   6.223 5.46e-10 ***
## fixed.acidity          5.471e-02  2.379e-02   2.300   0.0215 *
## volatile.acidity      -1.764e+00  1.328e-01 -13.281  < 2e-16 ***
## citric.acid            7.488e-02  1.136e-01   0.659   0.5098
## residual.sugar         7.074e-02  8.575e-03   8.249 2.24e-16 ***
## chlorides             -5.646e-01  6.428e-01  -0.878   0.3799
## free.sulfur.dioxide    5.183e-03  1.038e-03   4.995 6.17e-07 ***
## total.sulfur.dioxide -5.857e-04  4.477e-04  -1.308   0.1909
## density               -1.300e+02  2.116e+01  -6.142 9.11e-10 ***
## pH                     7.197e-01  1.224e-01   5.880 4.49e-09 ***
## sulphates              6.718e-01  1.178e-01   5.705 1.26e-08 ***
## alcohol                2.084e-01  2.694e-02   7.736 1.34e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7488 on 3417 degrees of freedom
## Multiple R-squared:  0.2855, Adjusted R-squared:  0.2832
## F-statistic: 124.1 on 11 and 3417 DF,  p-value: < 2.2e-16
```

```r
predictions_white <- predict(model_white, newdata = test_data_white)

mean_squared_error_white <- mean((test_data_white$quality - predictions_white)^2)
rsquared <- summary(model_white)$r.squared

ggplot(data = test_data_white, aes(x = quality, y = predictions_white)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Actual vs. Predicted Quality",
       x = "actual quality",
       y = "predicted quality")
```



```
par(mfrow = c(2, 2))
plot(model_white)
```



I’ve trained a logitstic regression model for white wine quality, calculated accuracy and displayed the confusion matrix.

```r
logistic_model_white <- glm(quality > 5 ~ ., data = train_data_white, family = binomial)
summary(logistic_model_white)
```

```
##
## Call:
## glm(formula = quality > 5 ~ ., family = binomial, data = train_data_white)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.993e+02  7.794e+01   2.557  0.01055 *
## fixed.acidity        -2.361e-02  8.135e-02  -0.290  0.77164
## volatile.acidity     -6.285e+00  4.862e-01 -12.928  < 2e-16 ***
## citric.acid           3.826e-01  3.650e-01   1.048  0.29454
## residual.sugar        1.463e-01  3.048e-02   4.798 1.60e-06 ***
## chlorides            -1.920e-01  2.036e+00  -0.094  0.92489
## free.sulfur.dioxide   1.312e-02  3.445e-03   3.808  0.00014 ***
## total.sulfur.dioxide -2.115e-03  1.451e-03  -1.457  0.14504
## density              -2.122e+02  7.901e+01  -2.685  0.00725 **
## pH                    1.164e+00  4.204e-01   2.769  0.00562 **
## sulphates             1.924e+00  4.286e-01   4.489 7.15e-06 ***
## alcohol               8.139e-01  1.036e-01   7.854 4.03e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4372.1  on 3428  degrees of freedom
## Residual deviance: 3420.3  on 3417  degrees of freedom
## AIC: 3444.3
##
## Number of Fisher Scoring iterations: 5
```

```r
logistic_predictions_white <- predict(logistic_model_white, newdata = test_data_white, type = "response
logistic_mse_white <- mean((test_data_white$quality - logistic_predictions_white)^2)

predicted_classes_white <- ifelse(logistic_predictions_white > 0.5, 1, 0)

confusion_matrix_white <- table(test_data_white$quality > 5, predicted_classes_white)
accuracy_white <- sum(diag(confusion_matrix_white)) / sum(confusion_matrix_white)
precision_white <- confusion_matrix_white[2, 2] / sum(confusion_matrix_white[, 2])
recall_white <- confusion_matrix_white[2, 2] / sum(confusion_matrix_white[2, ])
f1_score_white<- 2 * (precision_white * recall_white) / (precision_white + recall_white)

print(confusion_matrix_white)
```

```
##        predicted_classes_white
##           0   1
##   FALSE 235 257
##   TRUE  135 842
```

```r
metrics_white <- data.frame(
  Model = "Logistic Regression - white wine",
  MSE = logistic_mse_white,
  Accuracy = accuracy_white,
  Precision = precision_white,
  Recall = recall_white,
  F1_Score = f1_score_white
)

print("Metrics for white wine logistic regression:")
```

```
## [1] "Metrics for white wine logistic regression:"
```

```r
print(metrics_white)
```

```
##                               Model     MSE  Accuracy Precision    Recall
## 1 Logistic Regression - white wine 27.8709 0.7331518  0.766151 0.8618219
##    F1_Score
## 1 0.8111753
```

```r
print(paste("Accuracy (White Wine):", accuracy_white))
```

```
## [1] "Accuracy (White Wine): 0.733151803948264"
```

```r
print(paste("Precision:", precision_white))
```

```
## [1] "Precision: 0.766151046405823"
```

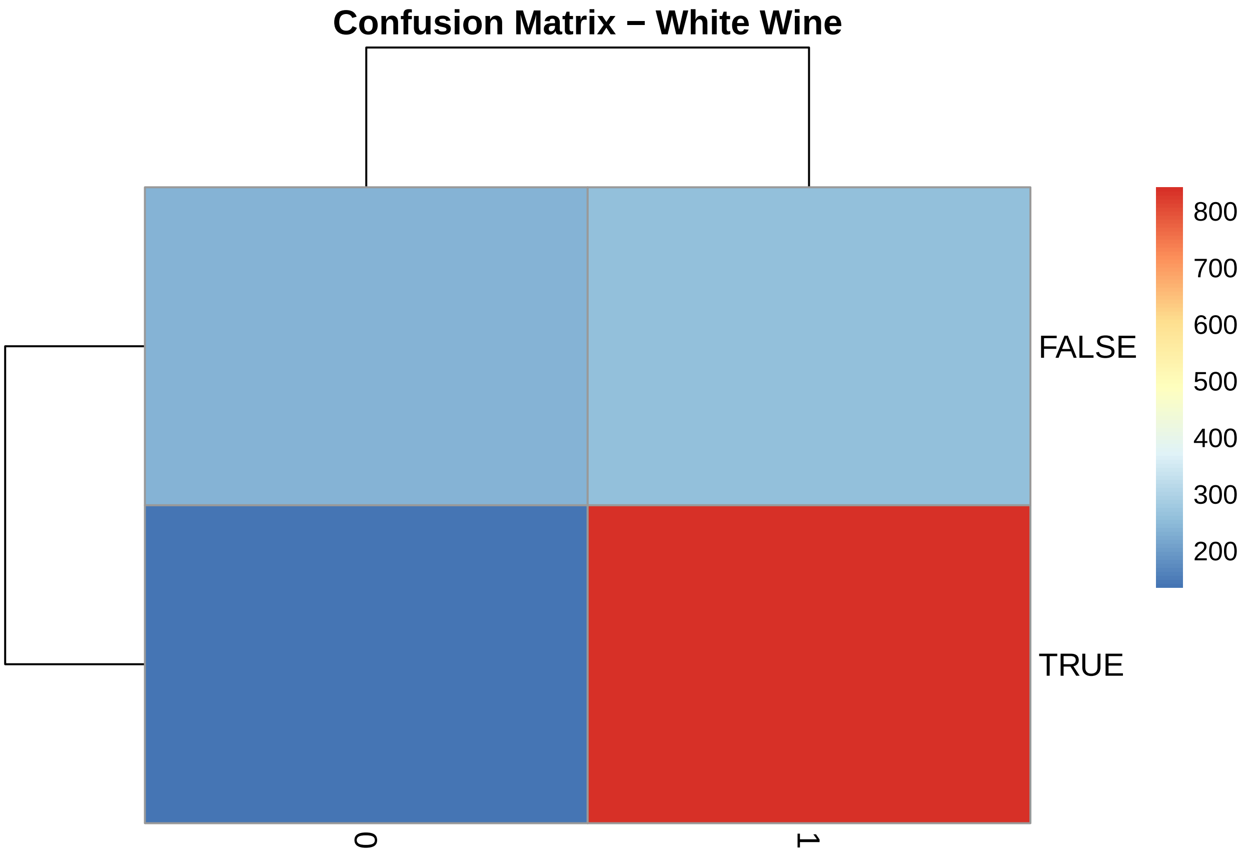
```r
print(paste("Recall:", recall_white))
```

```
## [1] "Recall: 0.861821903787103"
```

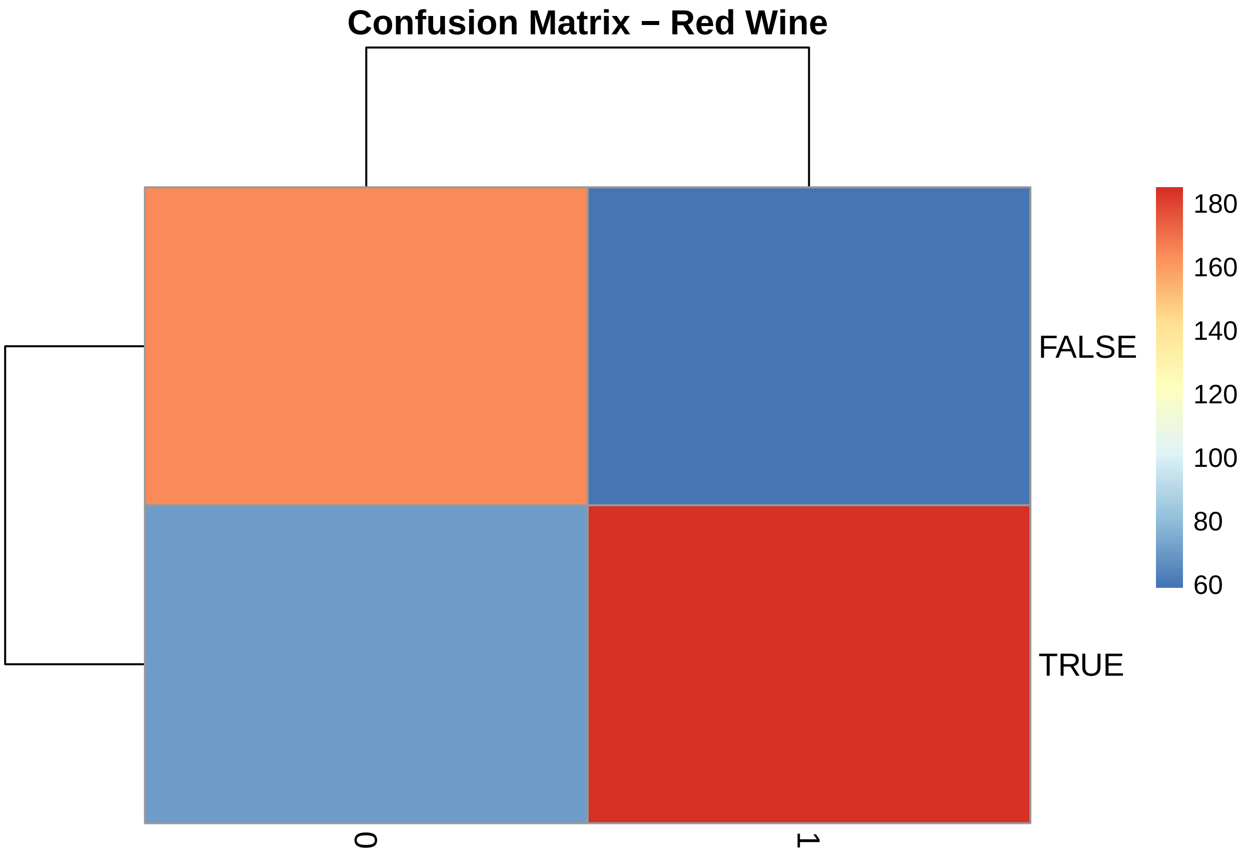
```r
print(paste("F1 Score:", f1_score_white))
```

```
## [1] "F1 Score: 0.811175337186898"
```

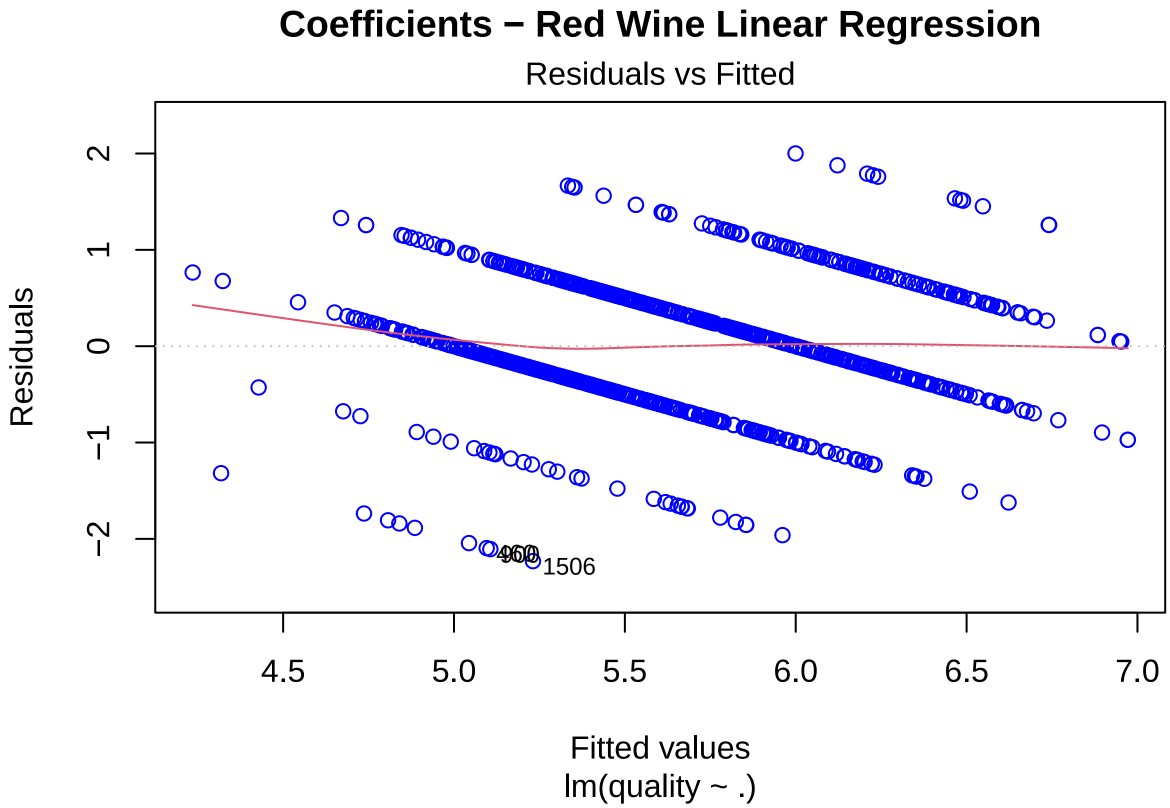
```r
library(pheatmap)

pheatmap(confusion_matrix_white, fontsize = 10,
         main = "Confusion Matrix - White Wine", fontsize_row = 12, fontsize_col = 12)
```

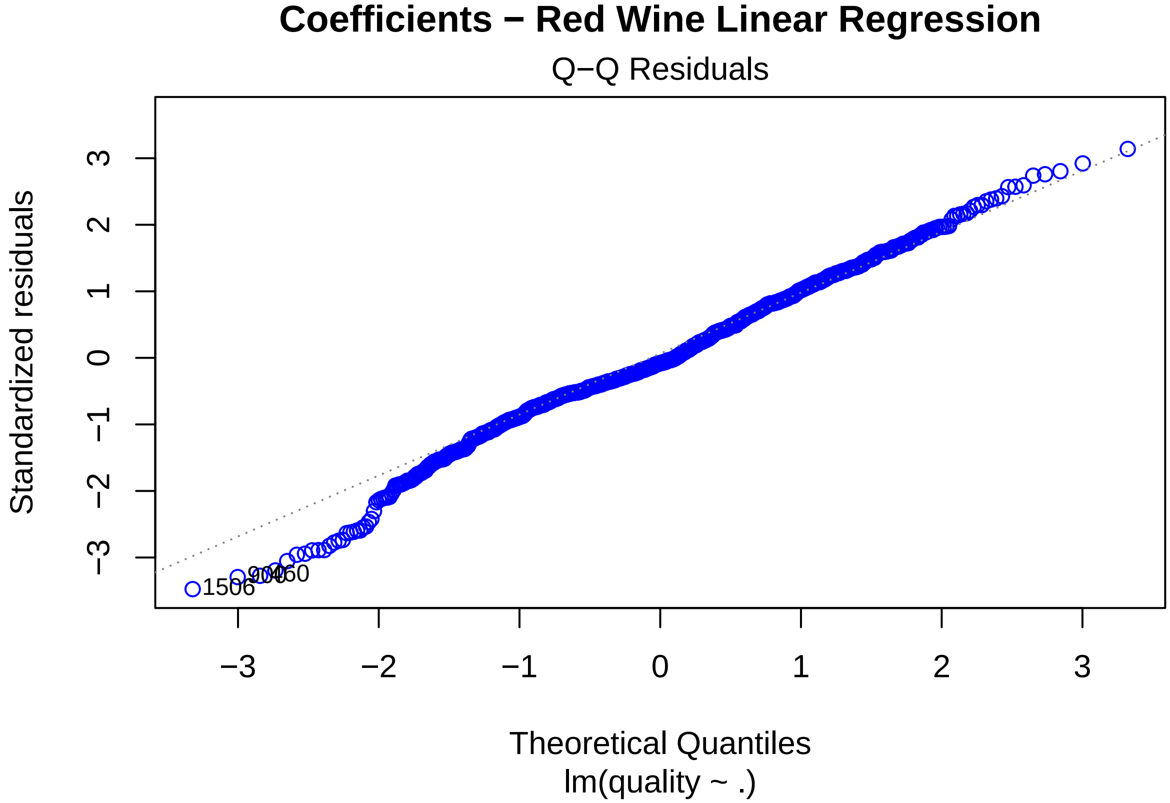


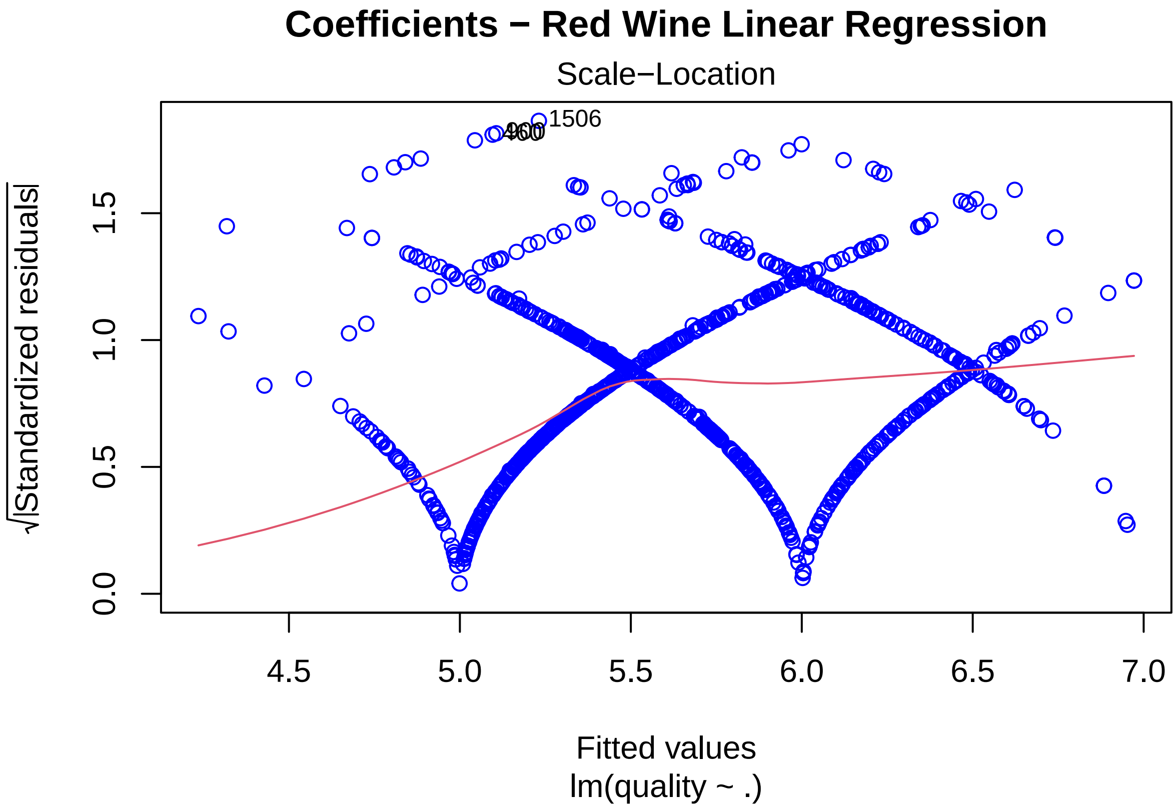
```r
pheatmap(confusion_matrix_red, fontsize = 10,
         main = "Confusion Matrix - Red Wine", fontsize_row = 12, fontsize_col = 12)
```

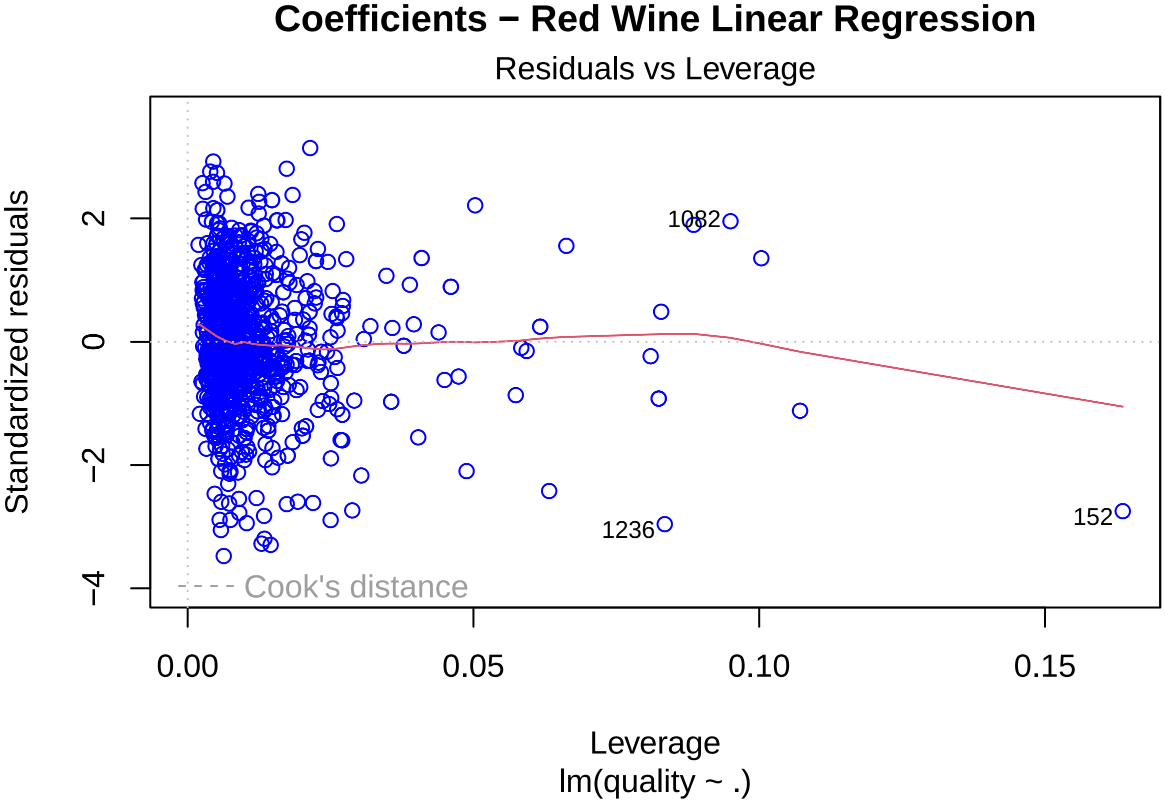


```
#plot coefficients for red and white wine linear regression models

plot(model_red, col = "blue", main = "Coefficients - Red Wine Linear Regression")
```

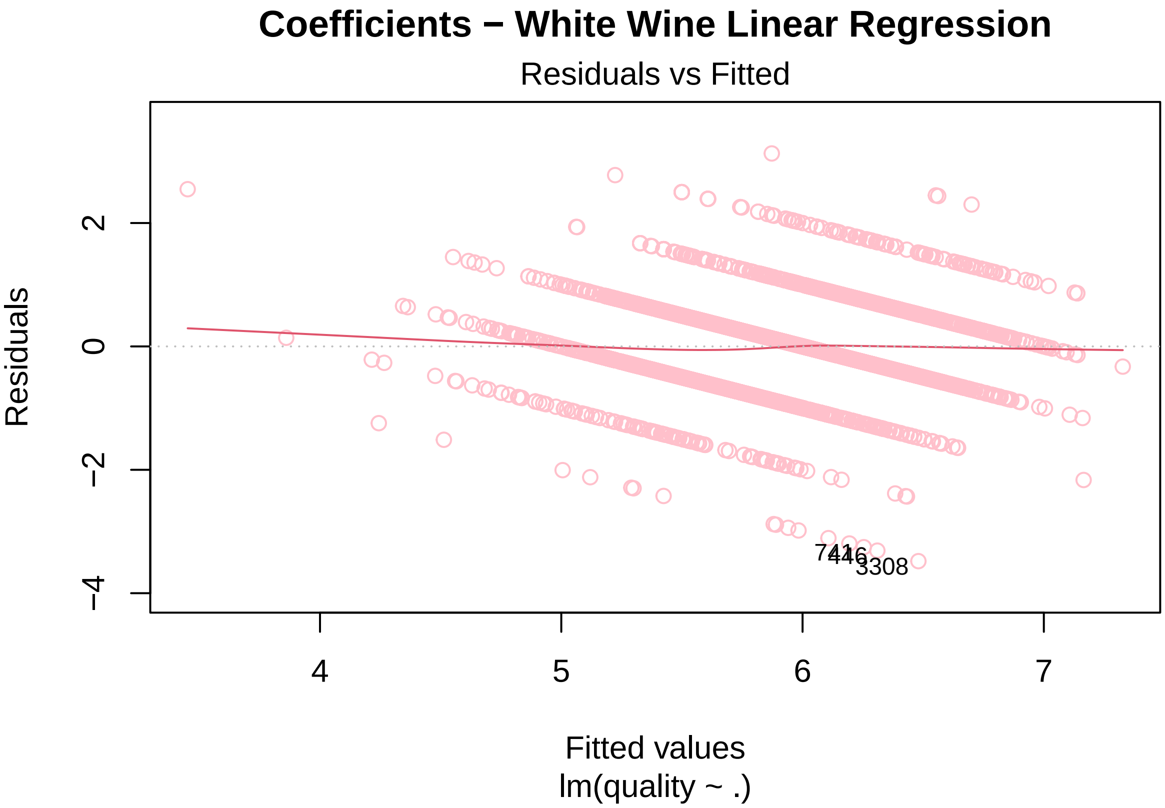

Coefficients − Red Wine Linear Regression
Residuals vs Fitted
Residuals
2
1
0
−1
−2
4.5
5.0
5.5
6.0
6.5
7.0
Fitted values
lm(quality ~ .)
460
900
1506

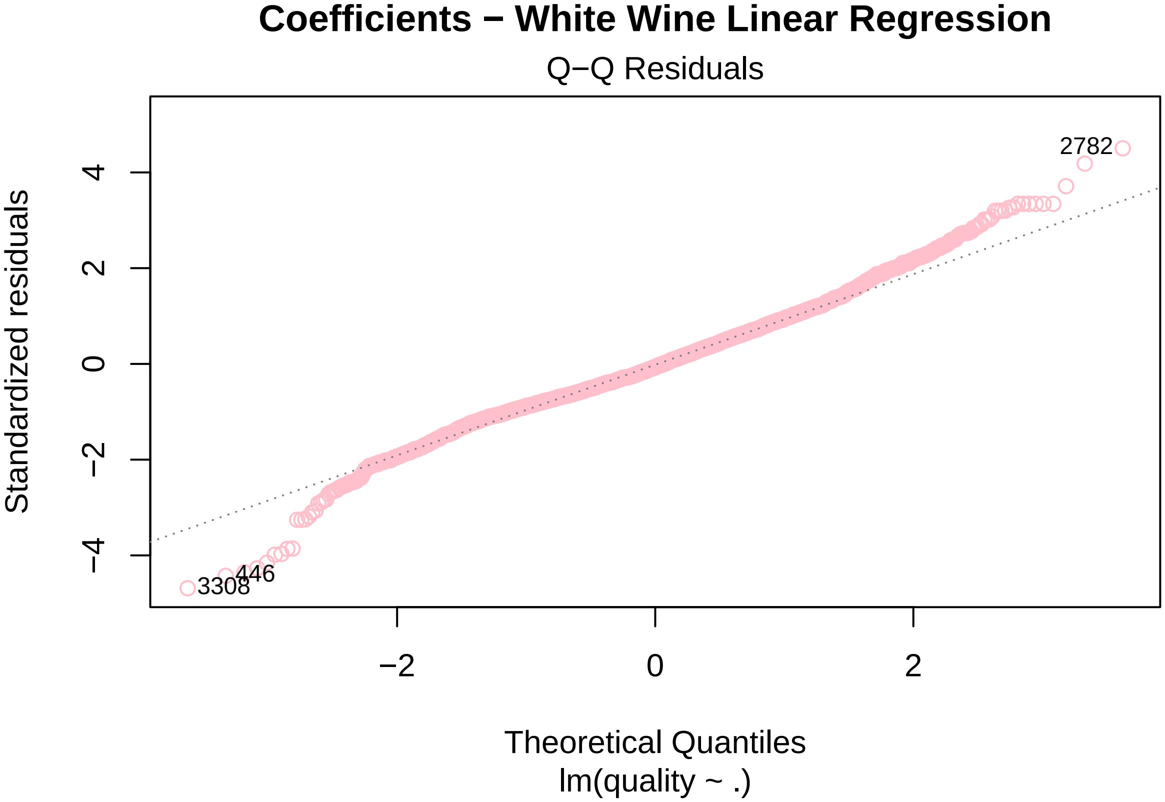

Coefficients − Red Wine Linear Regression
Q−Q Residuals
Standardized residuals
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(quality ~ .)
1506
900
460

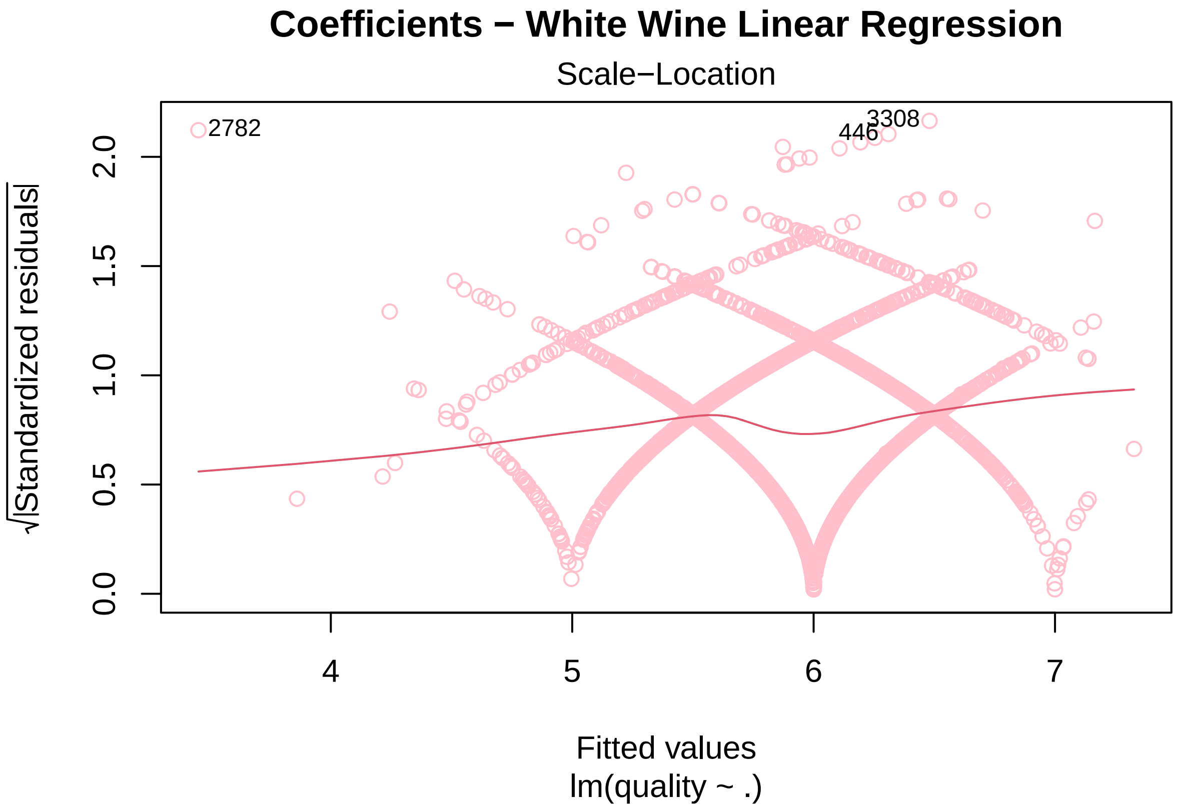


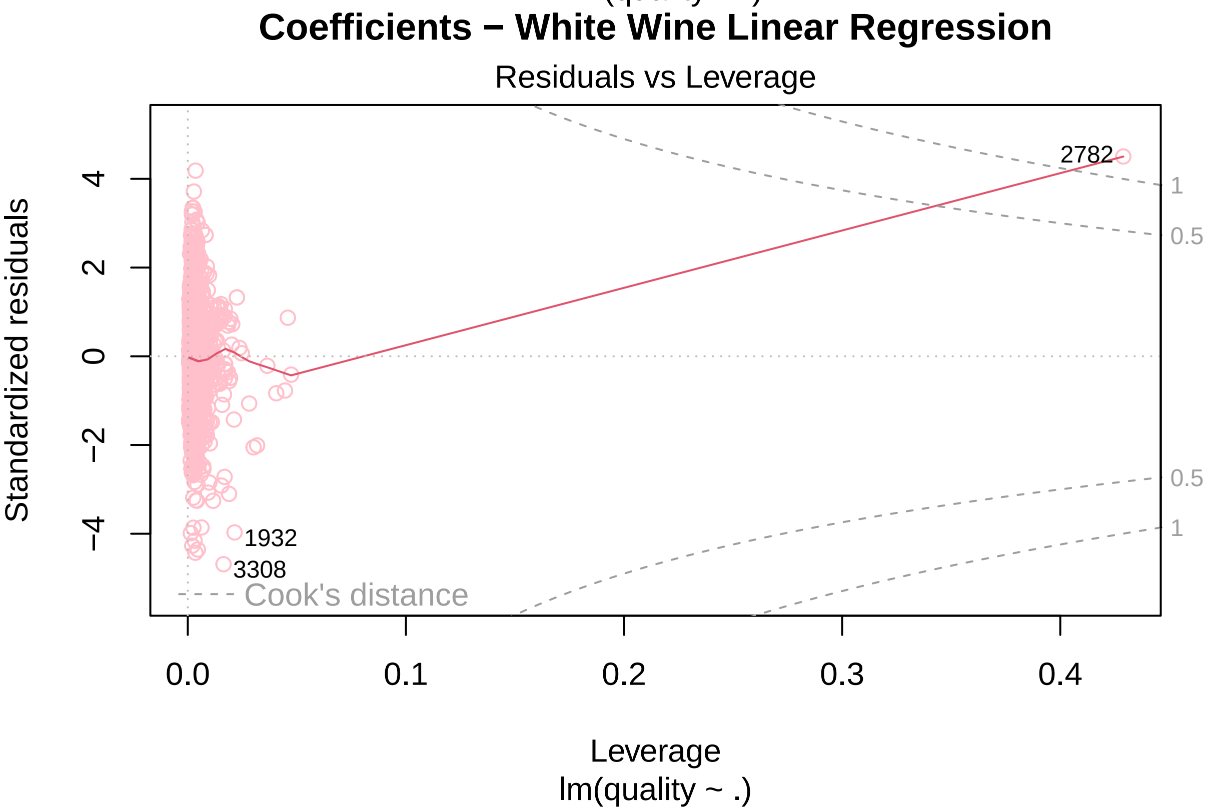


```r
plot(model_white, col = "pink", main = "Coefficients - White Wine Linear Regression")
```


Coefficients − White Wine Linear Regression
Residuals vs Fitted
Residuals
2
0
−2
−4
741
446
3308
4
5
6
7
Fitted values
lm(quality ~ .)


Coefficients − White Wine Linear Regression
Q−Q Residuals
Standardized residuals
4
2
0
−2
−4
2782
3308
446
−2
0
2
Theoretical Quantiles
lm(quality ~ .)





```r
library(e1071)

# train SVM models for red and white wine
```

```r
svm_model_red <- svm(quality ~ ., data = train_data_red)

svm_predictions_red <- predict(svm_model_red, newdata = test_data_red)

svm_mse_red <- mean((test_data_red$quality - svm_predictions_red)^2)
```

```r
svm_model_white <- svm(quality ~ ., data = train_data_white)

svm_predictions_white <- predict(svm_model_white, newdata = test_data_white)

svm_mse_white <- mean((test_data_white$quality - svm_predictions_white)^2)
```

```r
library(caret)

train_control <- trainControl(method = "cv", number = 10)

# train model using cross-validation for red wine
cv_model_red <- train(quality ~ ., data = train_data_red, method = "lm",
                  trControl = train_control)

print(cv_model_red)
```

```
## Linear Regression
##
## 1120 samples
##   11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1007, 1008, 1009, 1009, 1008, 1007, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.6481343  0.3529338  0.5050138
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
# train model using cross-validation for white wine
cv_model_white <- train(quality ~ ., data = train_data_white, method = "lm",
                  trControl = train_control)

print(cv_model_white)
```

```
## Linear Regression
##
## 3429 samples
##   11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 3086, 3086, 3086, 3087, 3085, 3086, ...
## Resampling results:
##
##   RMSE       Rsquared  MAE
##   0.7530688  0.277963  0.5833415
```

```
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

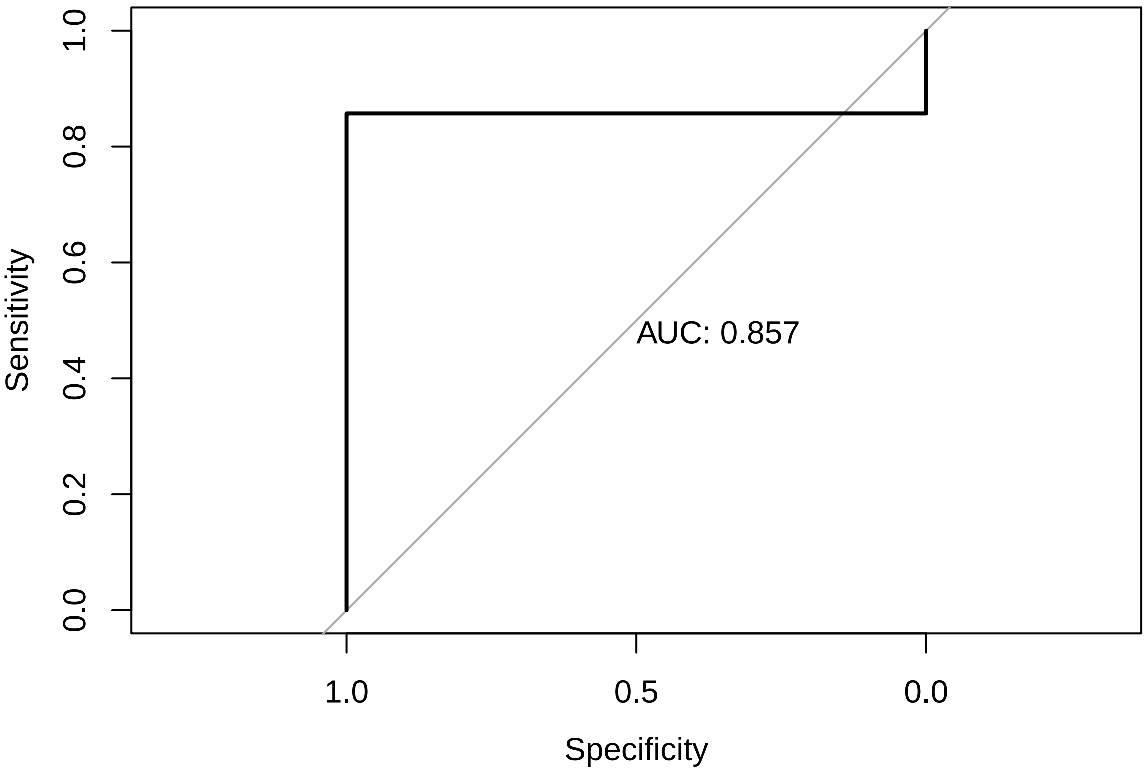
```r
library(pROC)
```

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

##
## Attaching package: 'pROC'

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

```r
#compute and plot ROC curves
roc_obj_red <- roc(test_data_red$quality, logistic_predictions_red)
```

```
## Warning in roc.default(test_data_red$quality, logistic_predictions_red):
## 'response' has more than two levels. Consider setting 'levels' explicitly or
## using 'multiclass.roc' instead

## Setting levels: control = 3, case = 4

## Setting direction: controls > cases
```

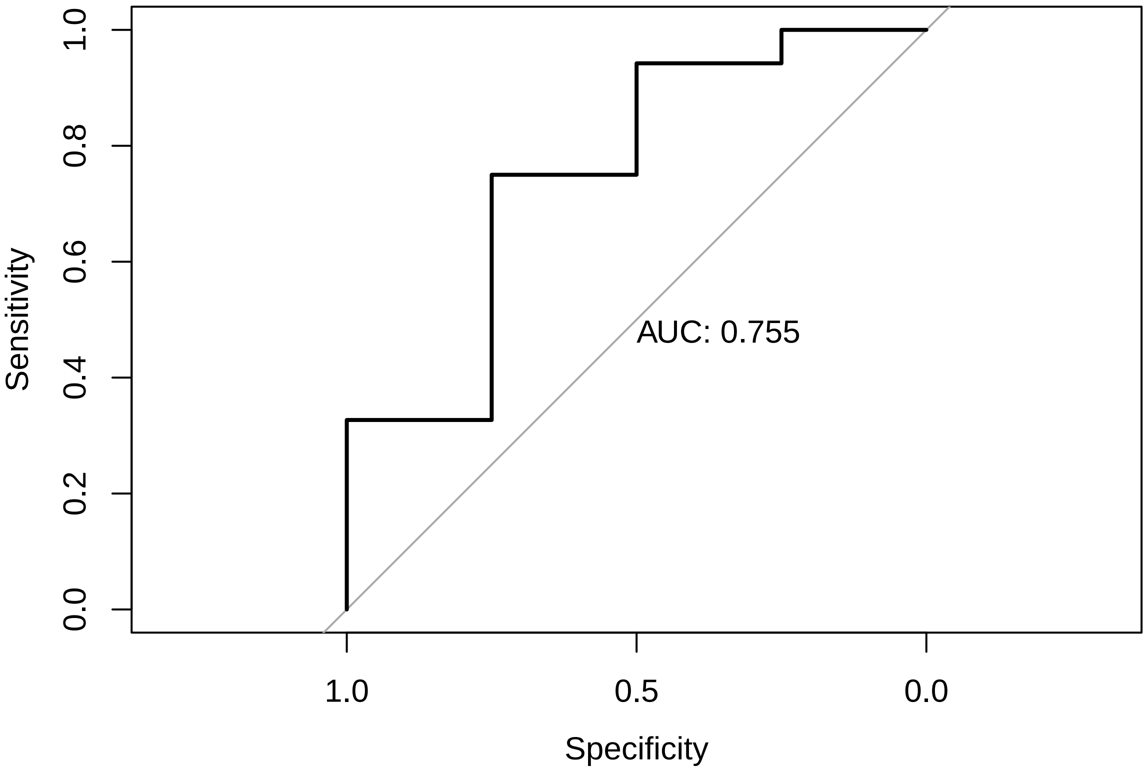
```r
plot(roc_obj_red, print.auc=TRUE)
```



```r
roc_obj_white <- roc(test_data_white$quality, logistic_predictions_white)
```

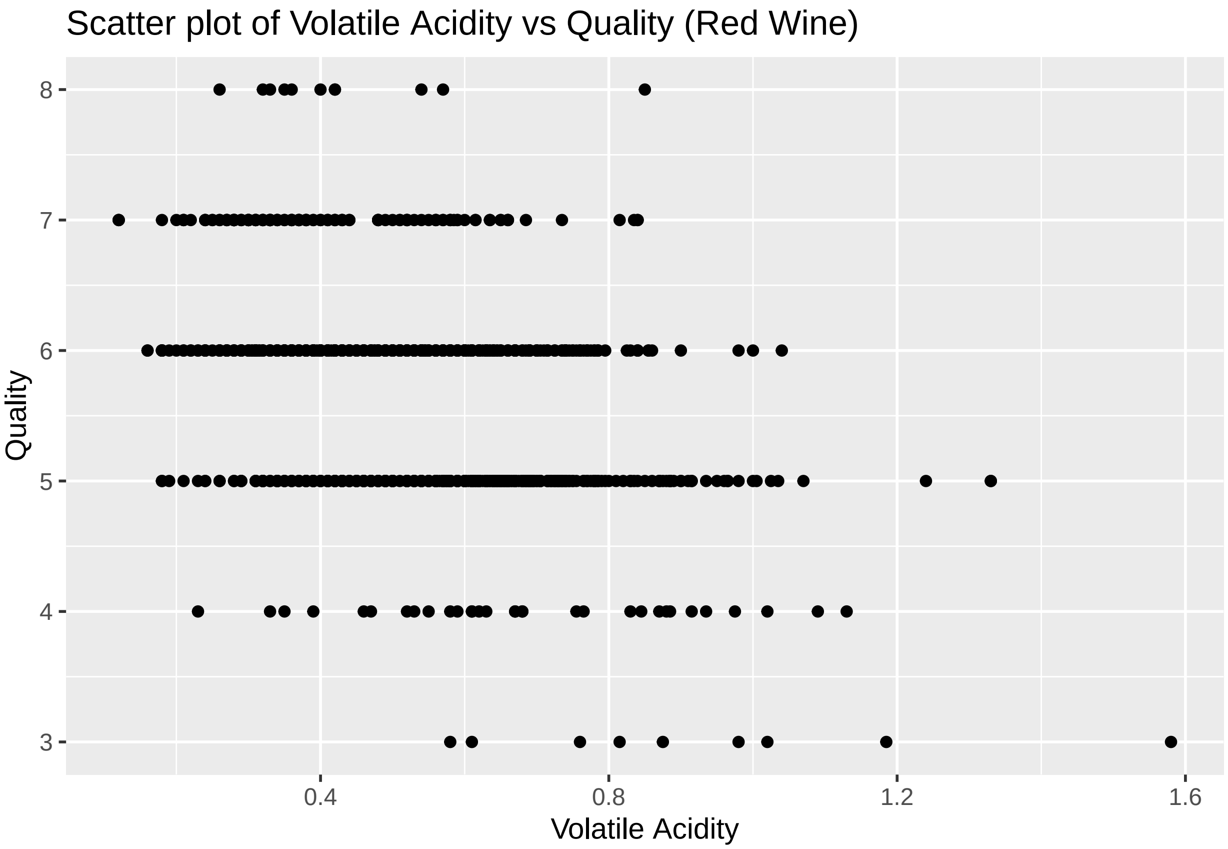
```
## Warning in roc.default(test_data_white$quality, logistic_predictions_white):
## 'response' has more than two levels. Consider setting 'levels' explicitly or
## using 'multiclass.roc' instead

## Setting levels: control = 3, case = 4
## Setting direction: controls > cases
```

```
plot(roc_obj_white, print.auc=TRUE)
```

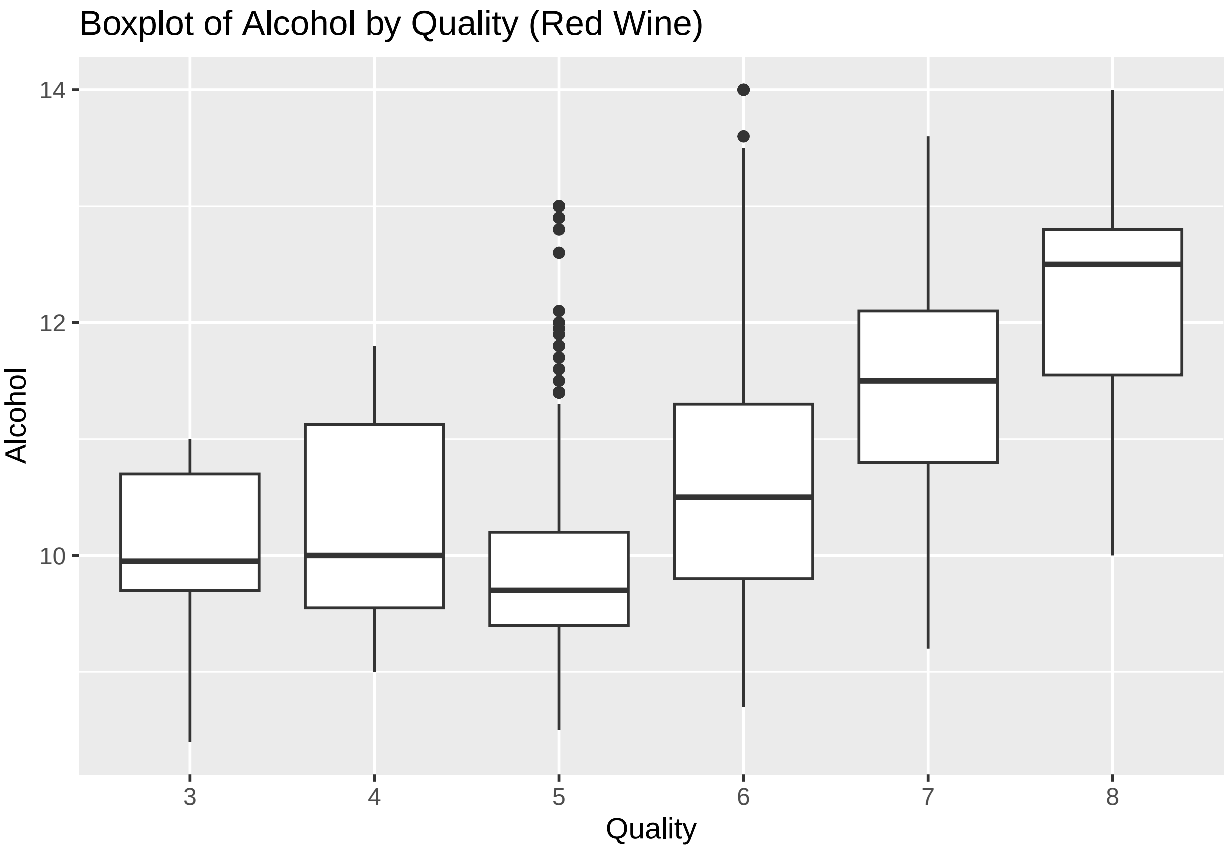


```
#make interaction term between volatile.acidity and alcohol for red & white wine
train_data_red$interaction_term <- train_data_red$volatile.acidity * train_data_red$alcohol

#apply log transformation to residual.sugar for red &white wine
train_data_red$log_residual_sugar <- log(train_data_red$residual.sugar)

train_data_white$interaction_term <- train_data_white$volatile.acidity * train_data_white$alcohol

train_data_white$log_residual_sugar <- log(train_data_white$residual.sugar)

#add squared terms for features
train_data_red$squared_residual_sugar <- train_data_red$residual.sugar^2
train_data_red$squared_alcohol <- train_data_red$alcohol^2

train_data_white$squared_residual_sugar <- train_data_white$residual.sugar^2
train_data_white$squared_alcohol <- train_data_white$alcohol^2
```

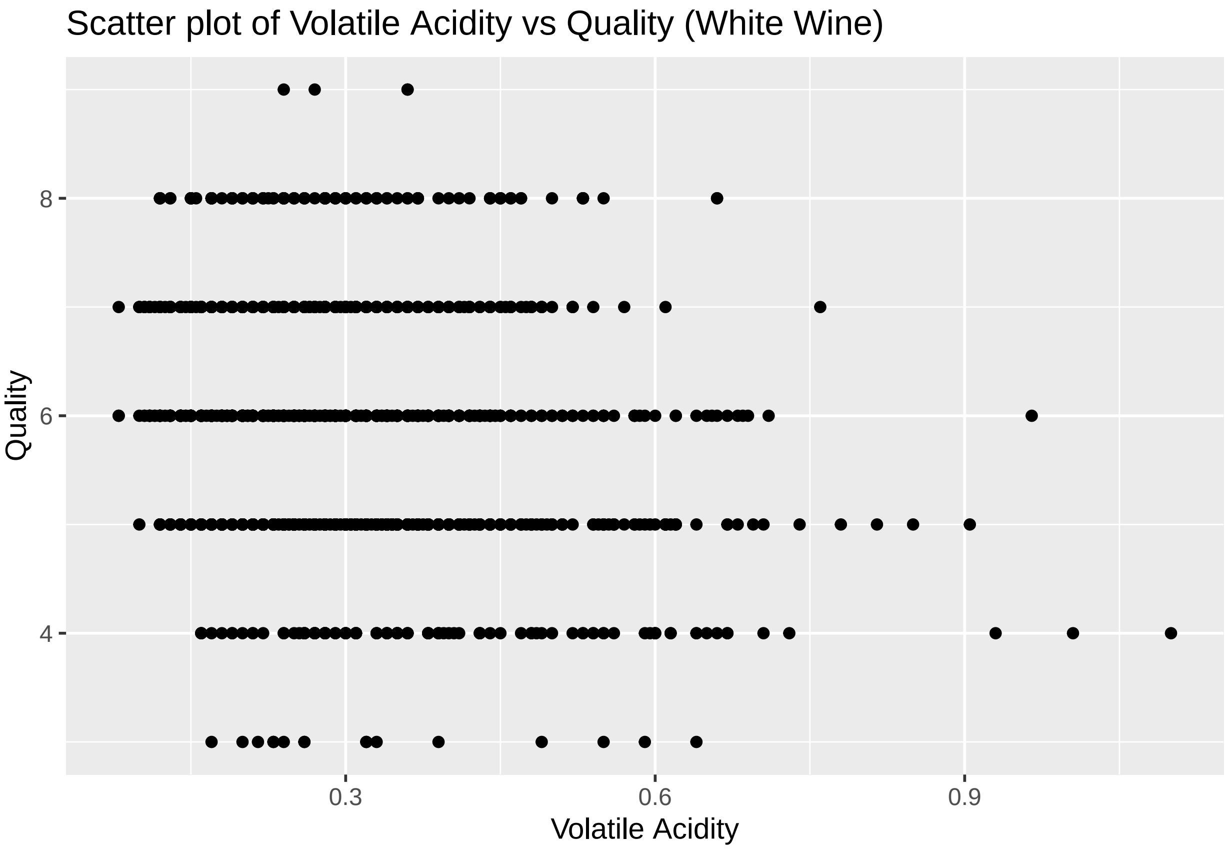
```
library(ggplot2)

ggplot(train_data_red, aes(x = volatile.acidity, y = quality)) +
  geom_point() +
  labs(x = "Volatile Acidity", y = "Quality", title = "Scatter plot of Volatile Acidity vs Quality (Red
```



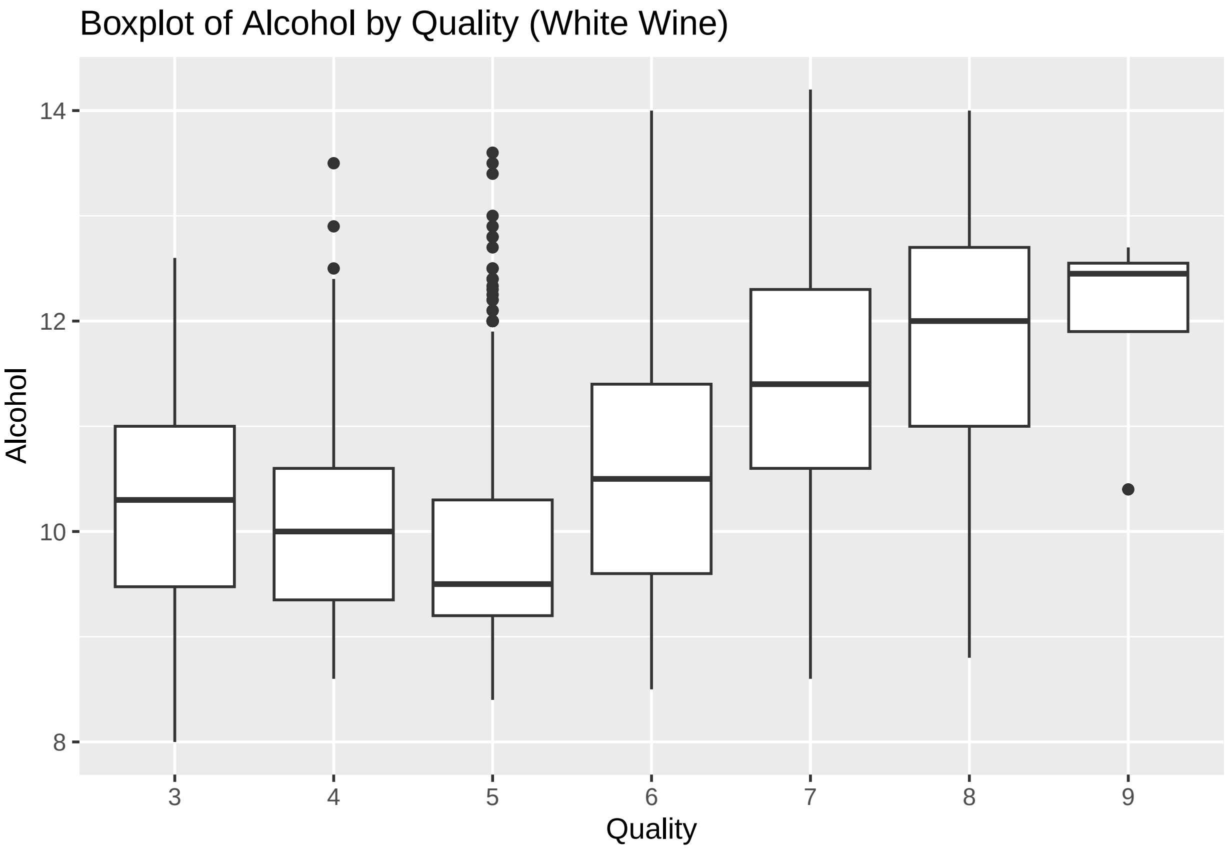
```
ggplot(train_data_red, aes(x = factor(quality), y = alcohol)) +
  geom_boxplot() +
  labs(x = "Quality", y = "Alcohol", title = "Boxplot of Alcohol by Quality (Red Wine)")
```



```
ggplot(train_data_white, aes(x = volatile.acidity, y = quality)) +
  geom_point() +
  labs(x = "Volatile Acidity", y = "Quality", title = "Scatter plot of Volatile Acidity vs Quality (Whi
```



```r
ggplot(train_data_white, aes(x = factor(quality), y = alcohol)) +
  geom_boxplot() +
  labs(x = "Quality", y = "Alcohol", title = "Boxplot of Alcohol by Quality (White Wine)")
```


Boxplot of Alcohol by Quality (White Wine)
Alcohol
14
12
10
8
3
4
5
6
7
8
9
Quality