Using predictive modeling to estimate the quality rating of a wine sample

Final Project IST 707

Nick Waine

Dr. Gregory Block

Introduction

Viticulturists understand that the market for wine is vast, with countless producers all jockeying for position in the highly competitive industry. Wines are scrutinized, compared, and judged by many different authorities on wine quality, such as Wine Spectator. These authorities are looked to by the consumer as a way to discern which ones are likely to be a hit at the next dinner party, or which bottles offer the bast bang for their buck. Wine producers are at the mercy of these judging authorities and are aware that their brand's reputation can be made or broken by a high- or low-quality rating. There are numerous characteristics of wine, both qualitative and quantitative, that can affect a particular batch's quality rating. Soil, geography, and climate are examples of qualitative characteristics that a producer has no control over once they've chosen a location for their vineyard. Storage temperature, harvest time, and barrel type are examples of variables that the producer does have control over, but how is a producer to know which aspects are affecting quality score? A blind taste-tested opinion of a wine connoisseur surely can't always fairly sum up the quality of a wine, so wine producers need as much information as possible about their concoctions before releasing them into the market to ensure a chance of a favorable rating. Machine learning can provide the answers to these questions for wine producers before releasing a product, so they can improve their chances of a better quality rating, and more revenue.

Analysis & Models

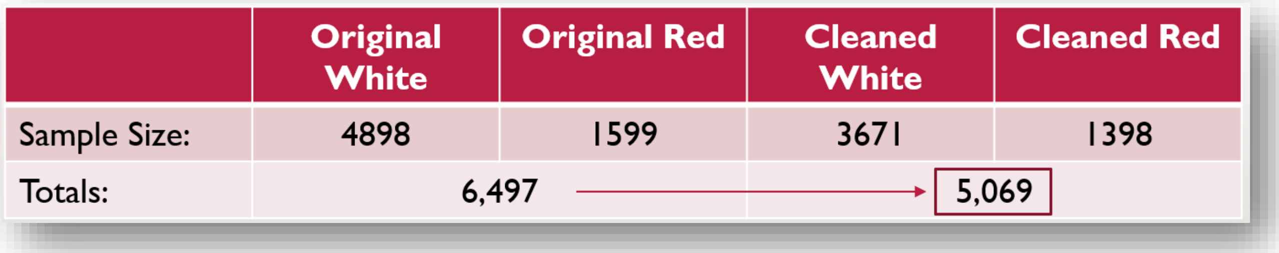
About The Data

The data used in this classification experiment was obtained from Kaggle and contains chemical composition data for over 6,000 types of wine, in both red and white grape variety. White wines make up around 75% of the data set. Each observation representing a type of wine includes a quality rating, with possible ratings from 1 to 10. The attributes included in the dataset are the following:

Attribute	**Description**
Type	Broad category of wine—red or white
Fixed Acidity	Fixed acidity content of the wine
Volatile Acidity	Volatile acidity content of the wine
Citric Acid	Citric acid content of the wine
Residual Sugar	Residual sugar content of the wine
Chlorides	Chloride content of the wine
Free Sulfur Dioxide	Free sulfur dioxide content of the wine
Total sulfur Dioxide	Total sulfur dioxide content of the wine
Density	Density property of the wine
pH	pH level of the wine
Sulphates	Total sulphate content of the wine
Alcohol	Alcohol by volume content of the wine
Quality (target variable)	Quality rating judged by Wine Spectator

Data Preparation

Cleaning steps were necessary to prepare the data for analysis and reduce noise within the data. Bad data points containing extreme outliers and missing data were removed. Once this was completed, the dataset had been reduced from 6,497 rows to 5,069 rows.



	Original White	Original Red	Cleaned White	Cleaned Red
Sample Size:	4898	1599	3671	1398
Totals:	6,497		5,069	

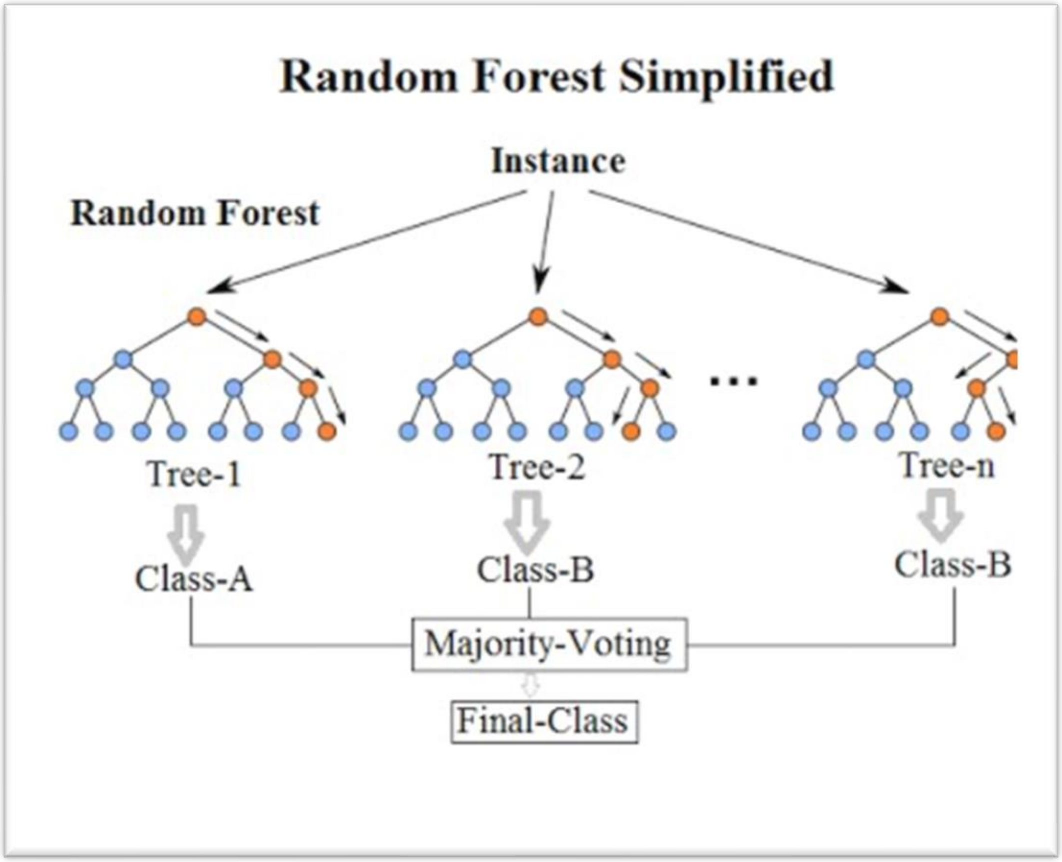
Analysis

Exploratory data analysis is necessary once any cleaning and processing steps have been completed, to get familiar with the nature of the data and become aware of any obvious trends or patterns. Becoming aware of significant patterns in the data in the early stages of analysis is important for the formulation of strong hypotheses and avoiding surprises at the modeling stage.

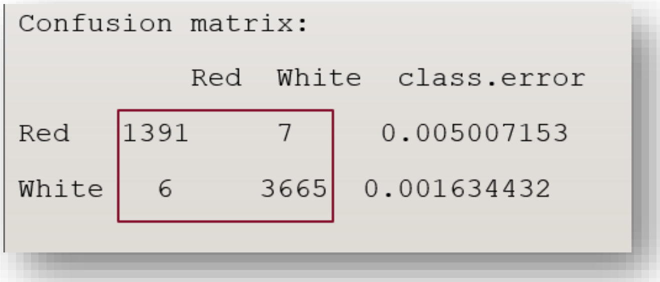
Models

The modeling technique selected to answer the question of predicting a wine's quality score needed to meet the criteria for a classification problem. The model needed to be able to sort wine samples without a quality score (test data) into "quality" buckets by comparing the attributes to samples with known quality ratings (training data).

The Random Forest algorithm uses the functions of a simple decision tree algorithm but has a greater capacity for large amounts of data with multiple variables at each split. It is powerful enough to handle data with larger numbers of attributes, which can be a prohibiting factor for other models.



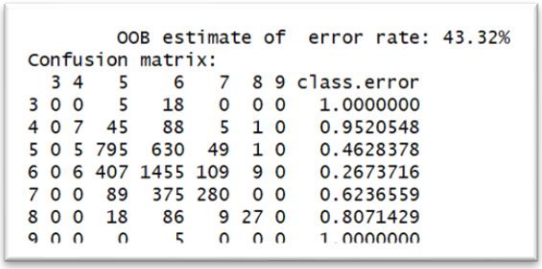
To test the functionality of the Random Forest, an initial test was performed to check the algorithm's ability to classify wine as "red" or "white". The algorithm performed admirably in this first test, only misclassifying 7 of 1398 red wines, and 6 of 3671 white wines. The "color" test inspired confidence in the algorithm ahead of running quality score predictions.



```
Confusion matrix:
          Red  White  class.error
Red      1391      7  0.005007153
White       6   3665  0.001634432
```

Results

Results of the first iteration of the Random Forest model did not inspire the same amount of confidence as the wine "color" test, as there was an observed out-of-bag error rate of 43.32%. The out-of-bag (OOB) estimate is an estimation of how well the model will perform with future unseen data, giving wine producers ample reason to worry that their wine samples will be misjudged nearly one-third of the time, perhaps causing them to be over-confident in a mediocre product, or less confident in wine with good potential; this would be an ineffective model. The confusion matrix for the first run is shown below:

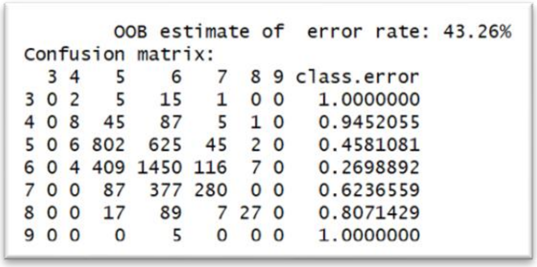


```
        OOB estimate of  error rate: 43.32%
Confusion matrix:
  3 4   5    6   7  8 9 class.error
3 0 0   5   18   0  0 0   1.0000000
4 0 7  45   88   5  1 0   0.9520548
5 0 5 795  630  49  1 0   0.4628378
6 0 6 407 1455 109  9 0   0.2673716
7 0 0  89  375 280  0 0   0.6236559
8 0 0  18   86   9 27 0   0.8071429
9 0 0   0    5   0  0 0   1.0000000
```

Results of the initial Random Forest test.

The first iteration of the model used default hyperparameters (500 trees, 1 variable at each split), and only factored in three variables for prediction: Residual Sugar, pH, and Citric Acid levels.

A classification error rate of over 43% meant that improvements to the model needed to be made to see a higher accuracy rate. Considering number of trees in the following test, a more complex model with 1000 trees would be tested next:



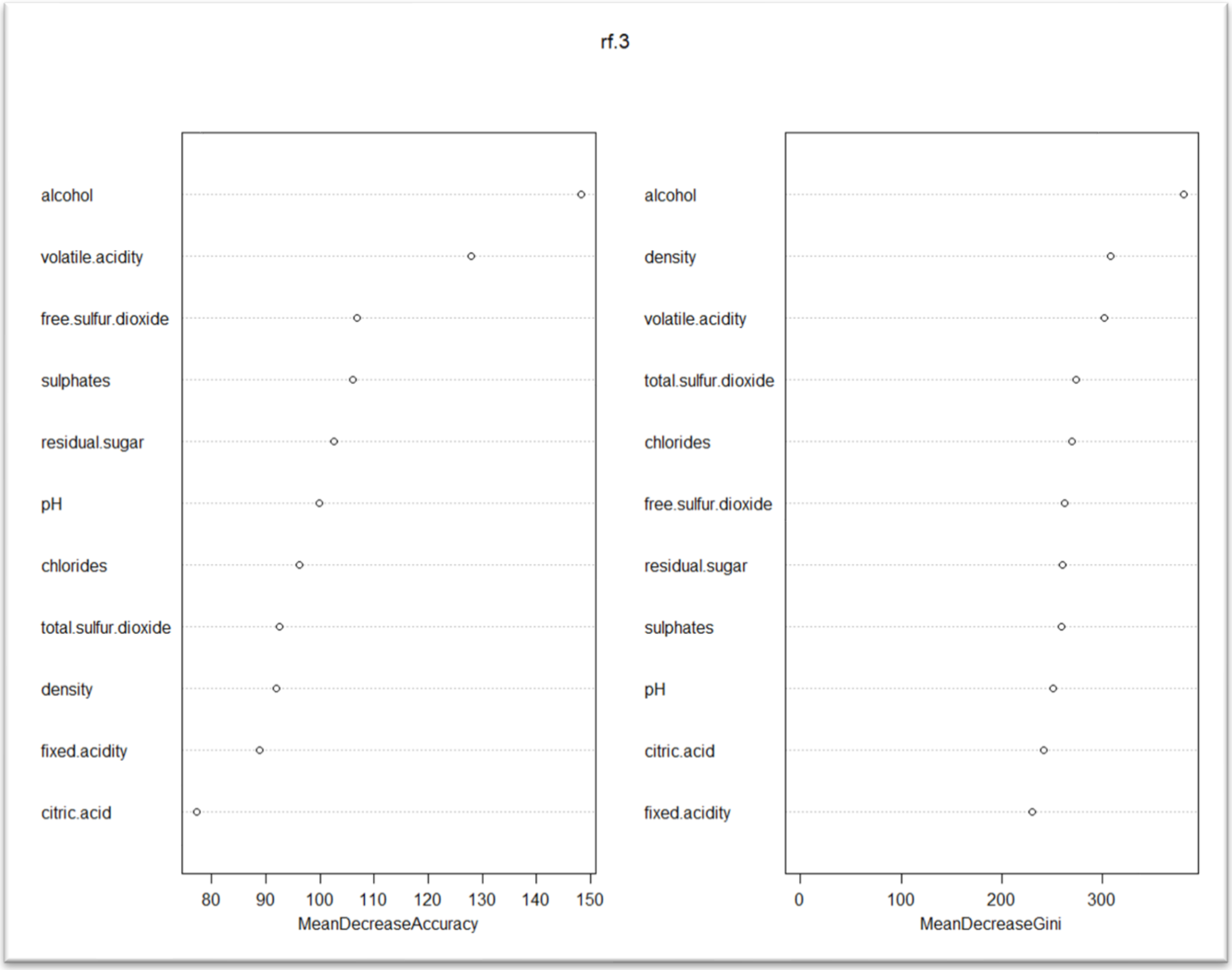
```
        OOB estimate of  error rate: 43.26%
Confusion matrix:
  3 4   5    6   7  8 9 class.error
3 0 2   5   15   1  0 0   1.0000000
4 0 8  45   87   5  1 0   0.9452055
5 0 6 802  625  45  2 0   0.4581081
6 0 4 409 1450 116  7 0   0.2698892
7 0 0  87  377 280  0 0   0.6236559
8 0 0  17   89   7 27 0   0.8071429
9 0 0   0    5   0  0 0   1.0000000
```

Results of the second Random Forest test.

Results showed virtually the same error rate, with a negligible improvement of less than 0.1 percent. The next model would need to include more attributes to achieve a higher accuracy rating.

Additionally, further testing was necessary to help determine which attributes held the most predictive power.

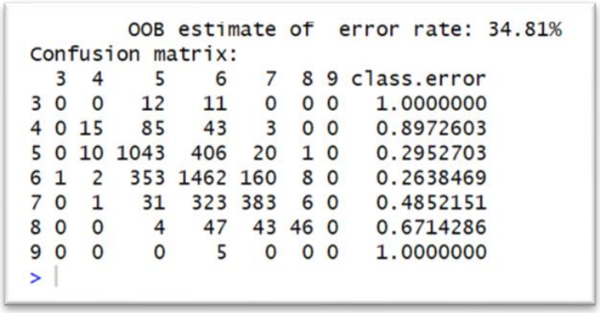
After including the rest of the attributes and changing parameters to train 1000 trees and three variables at each split in the next model, the result was an error rate of around 32.71%, a significant improvement from the initial 43%. Predictions were getting stronger, so it was time to identify which attributes held the most predictive power. Random Forest's feature selection could be visualized using a variable importance plot.



RF 3's Variable Importance Plot.

Random Forest's feature selection showed that alcohol level and volatile acidity had the most predictive power in estimating a wine's quality level. A gradual decrease in predictive power is seen beginning with sulfur dioxide and sulphates.

With the additional information about variable importance, another iteration of the model was done which only included the five variables with the strongest predictive power, to reduce noise in the model:



```
        OOB estimate of  error rate: 34.81%
Confusion matrix:
  3  4    5    6   7  8 9 class.error
3 0  0   12   11   0  0 0   1.0000000
4 0 15   85   43   3  0 0   0.8972603
5 0 10 1043  406  20  1 0   0.2952703
6 1  2  353 1462 160  8 0   0.2638469
7 0  1   31  323 383  6 0   0.4852151
8 0  0    4   47  43 46 0   0.6714286
9 0  0    0    5   0  0 0   1.0000000
> 
```

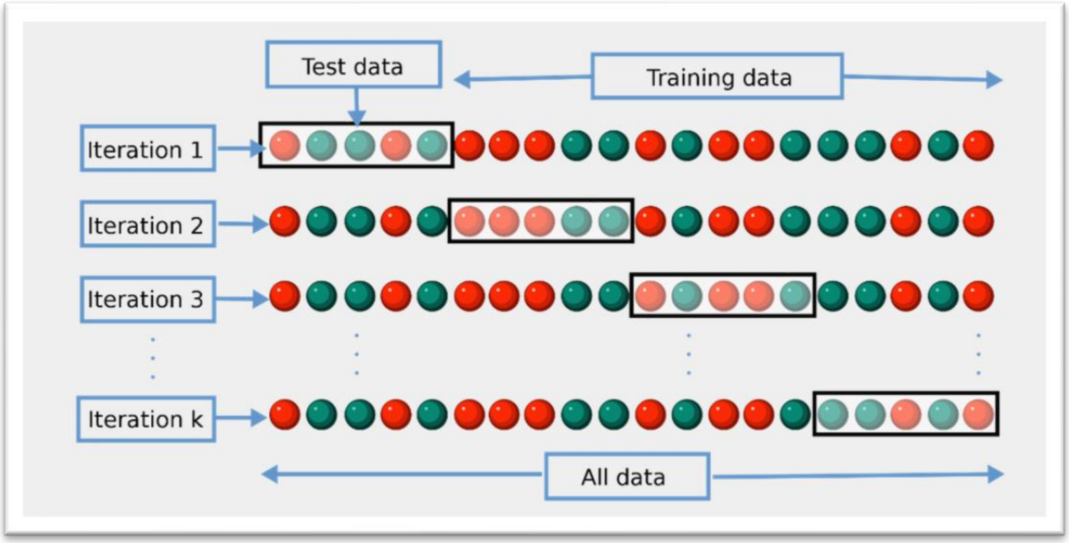
Results of fourth iteration of Random Forest model.

Results of the fourth model had a worse error rate than expected, and the effect of removing variables that were lower on the "importance" scale, according to the algorithm, was less information available for prediction and thus a lower accuracy.

The moral of the story from this is to provide as many training inputs to the algorithm as possible to obtain a better accuracy score. However, overfitting is a potential hazard of machine learning techniques that can lead to unreliable or misleading results, and Cross Validation is a way to combat this.

Cross Validation

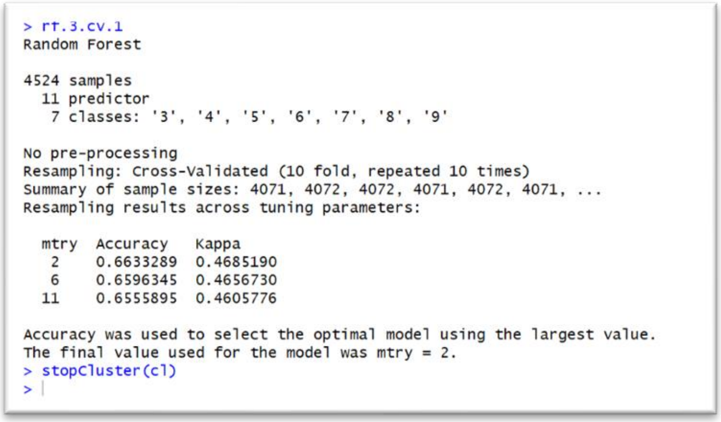
Cross Validation is arguably the most important concept to leverage when applying machine learning algorithms to data science questions. It is a way to maximize the value of the limited amount of training data available, and acts as another method of testing how accurate the model will perform against unseen data.



Diagrammatical Cross Validation theory.

Cross Validation was used to check if the estimated Out-of-Bag (OOB) error rate was a fair reflection of the model in practice against unseen data. N-fold Cross Validation works by splitting the training data into N logical subsets, and iteratively holding out one of the subsets as a test set, to train against the remaining subsets, and running predictions against the subset that was held out to obtain a result set of accuracy. This is done n-times, until each of the n subsets has been used as the holdout set. The error rate is then averaged to produce a "real world accuracy" score to evaluate how successful the model is likely to be at prediction using unseen data. Ten-fold Cross Validation has the ability to effectively boost the power of a random forest algorithm with 1000 trees and make it as if it had 100,000 trees (10 x 10 x 1000). This can become a resource intensive computation when dealing with datasets containing numerous variables and many observations.

Using trainControl, hyperparameters can be tuned and CPU cores can be allocated to provide additional computing power. The process is timely when the model includes several attributes, but the output is highly reflective of a model's performance. Using the third iteration of the random forest model, which included all 10 attributes for predicting quality:



```
> rf.3.cv.1
Random Forest

4524 samples
  11 predictor
   7 classes: '3', '4', '5', '6', '7', '8', '9'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 4071, 4072, 4072, 4071, 4072, 4071, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.6633289  0.4685190
   6    0.6596345  0.4656730
  11    0.6555895  0.4605776

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
> stopCluster(cl)
> 
```

Results of 10-fold Cross Validation.

Accuracy levels between 65-66% were determined by the 10-fold cross validation. This confirmed that the out-of-bag error rate of around 32% for the random forest model would hold up against future unseen data.

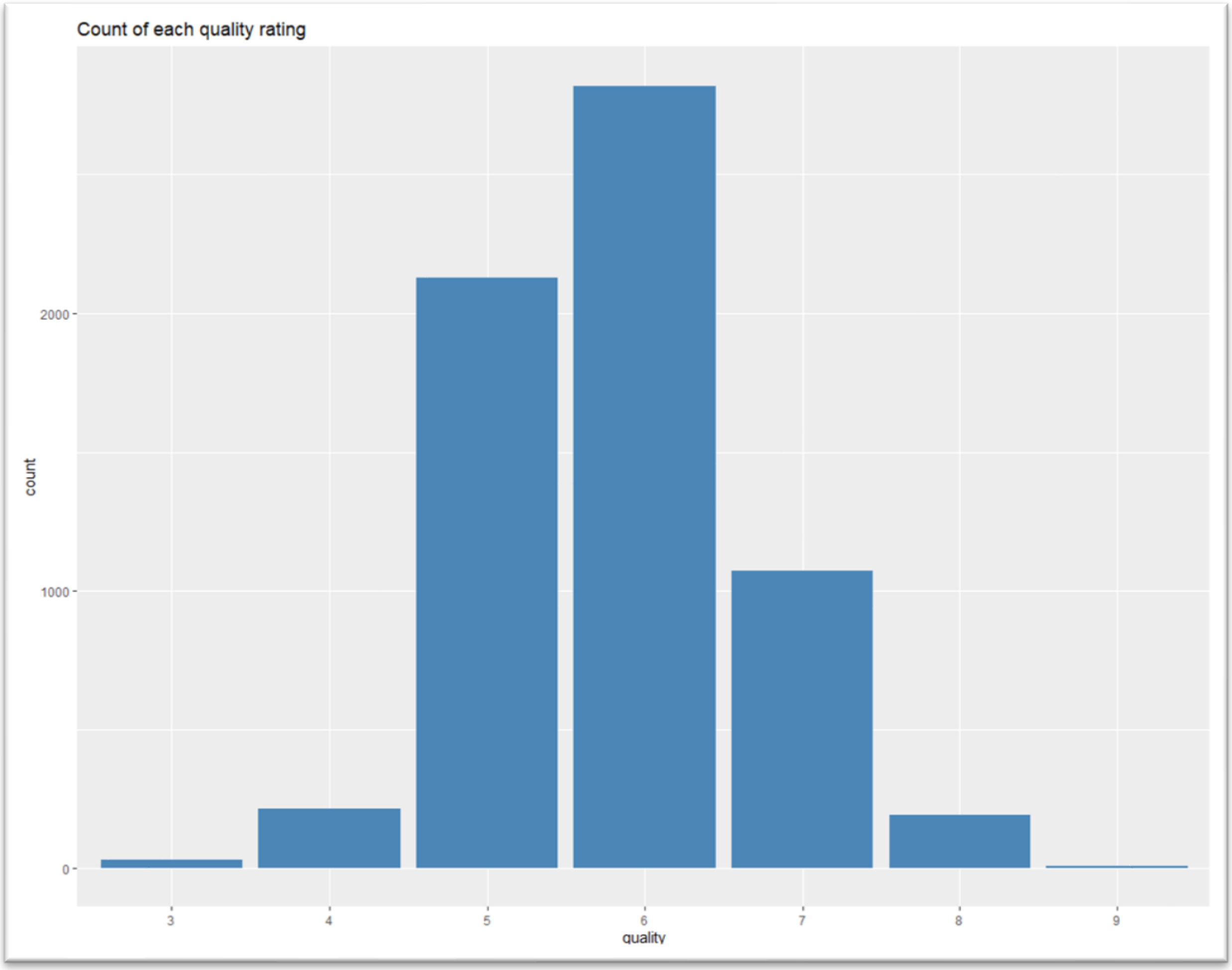
Conclusions

Exploratory data analysis showed that no recommendations could be made which applied to the production processes for both red and white wine varieties.

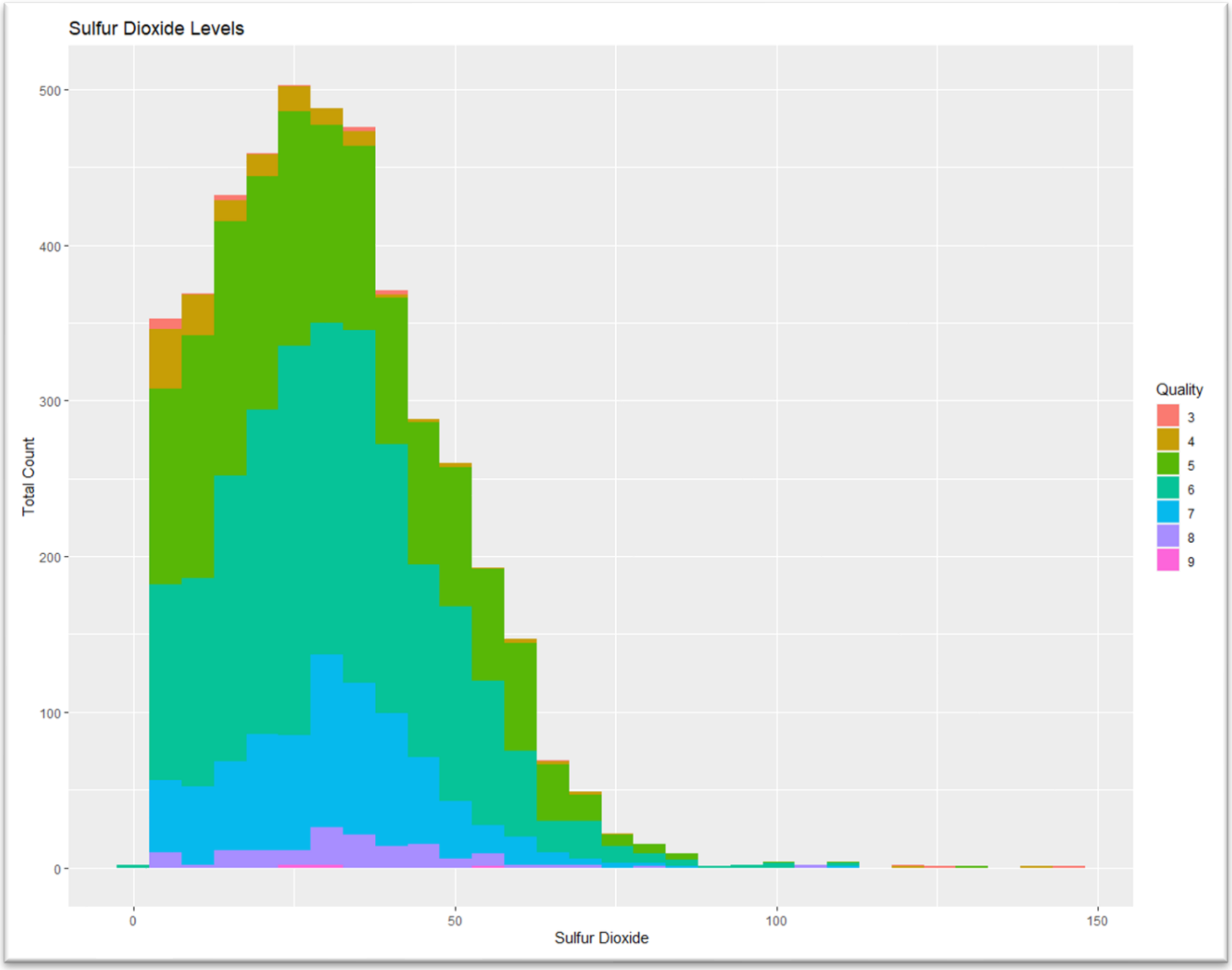
The most useful information that winemakers could come away with from this application, despite the model's ultimate failure to predict wine quality to a precise degree, is the knowledge of which attributes to focus on when considering future production processes. What could they do to optimize these numbers in future batches? If, for example, it could be determined that a lower volatile acidity usually predicted a good quality wine, what could be changed in the production process to reduce volatile acidity?

Data would be more useful if it included more categorical variables to be used as inputs. Examples of categorical inputs would be climate type, grape family, and barrel type. Producers may find that the algorithm would identify a categorical variable as the most important feature associated with a quality score and would be able to implement the necessary changes to ensure higher scores on future batches.

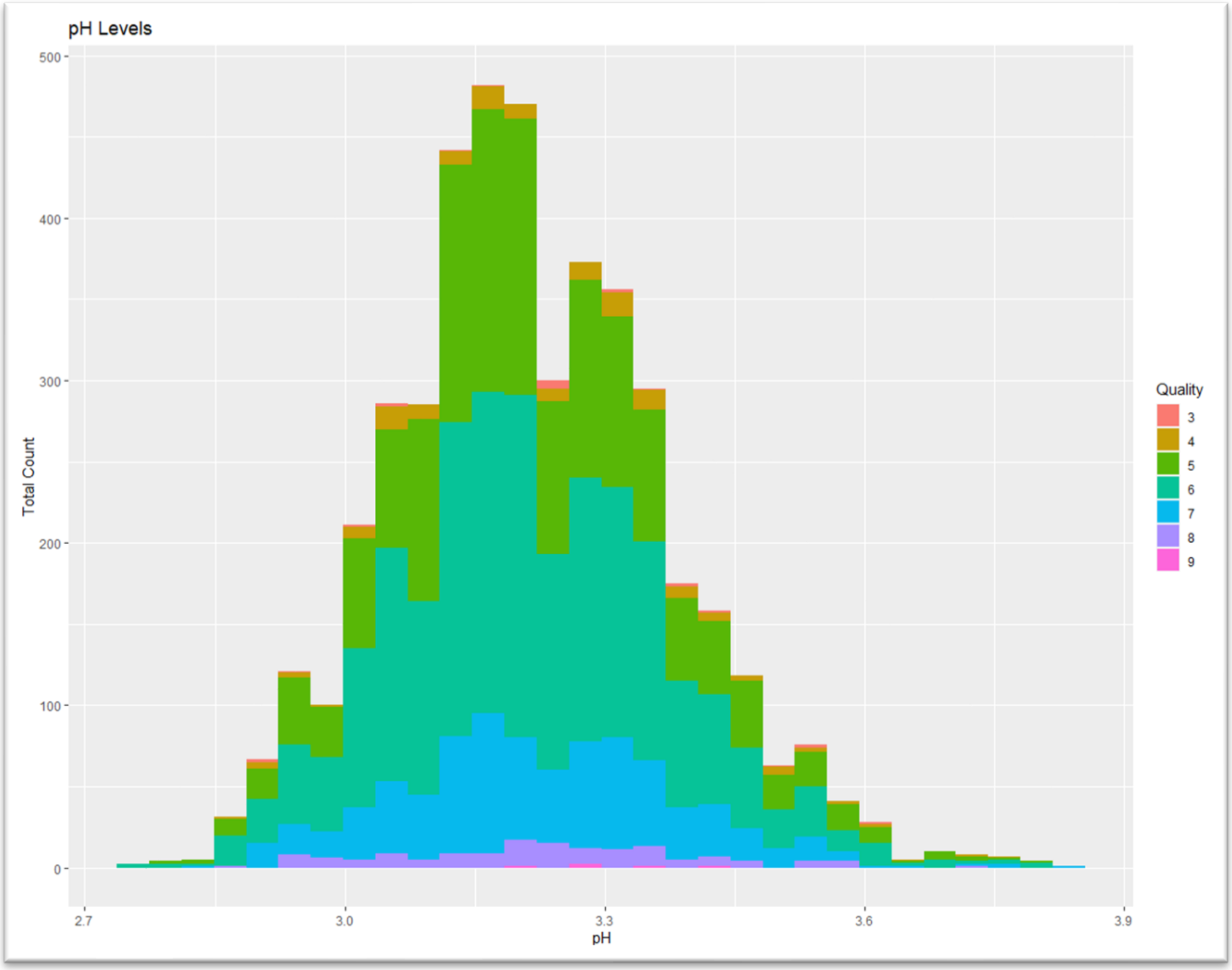
EDA - Visualizations



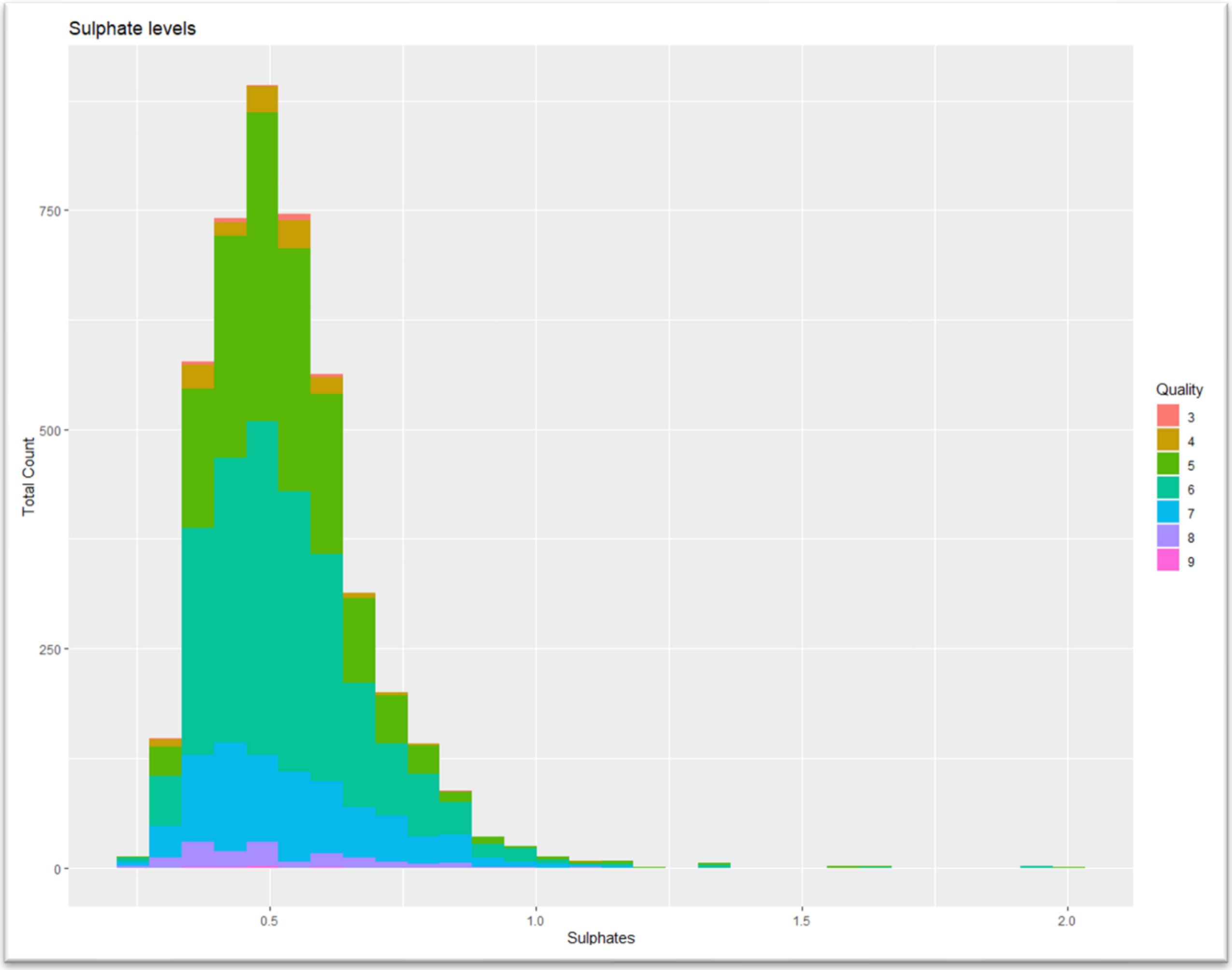
Frequency of quality ratings fit a normal distribution.



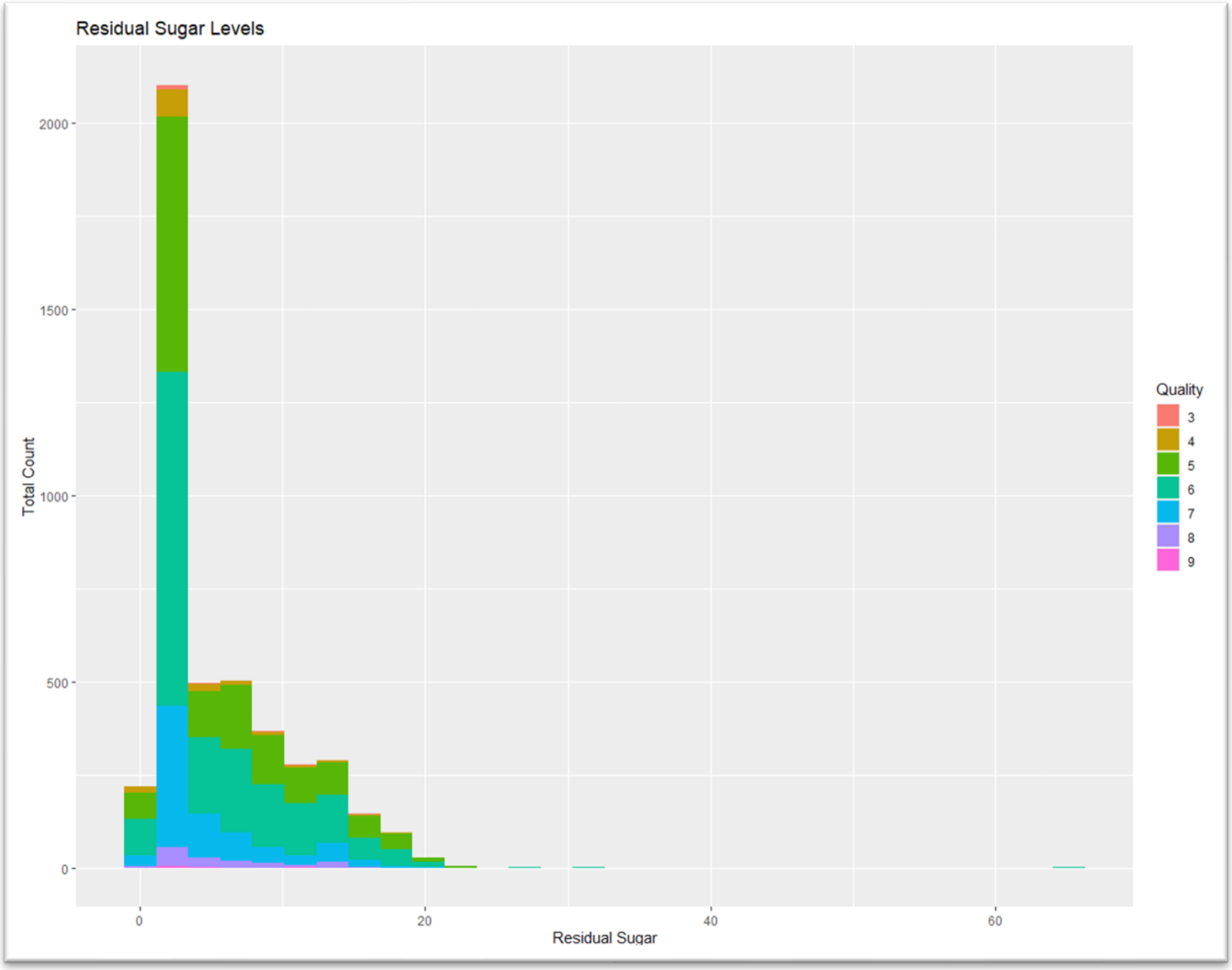
Count of wines across all levels of sulfur dioxide.



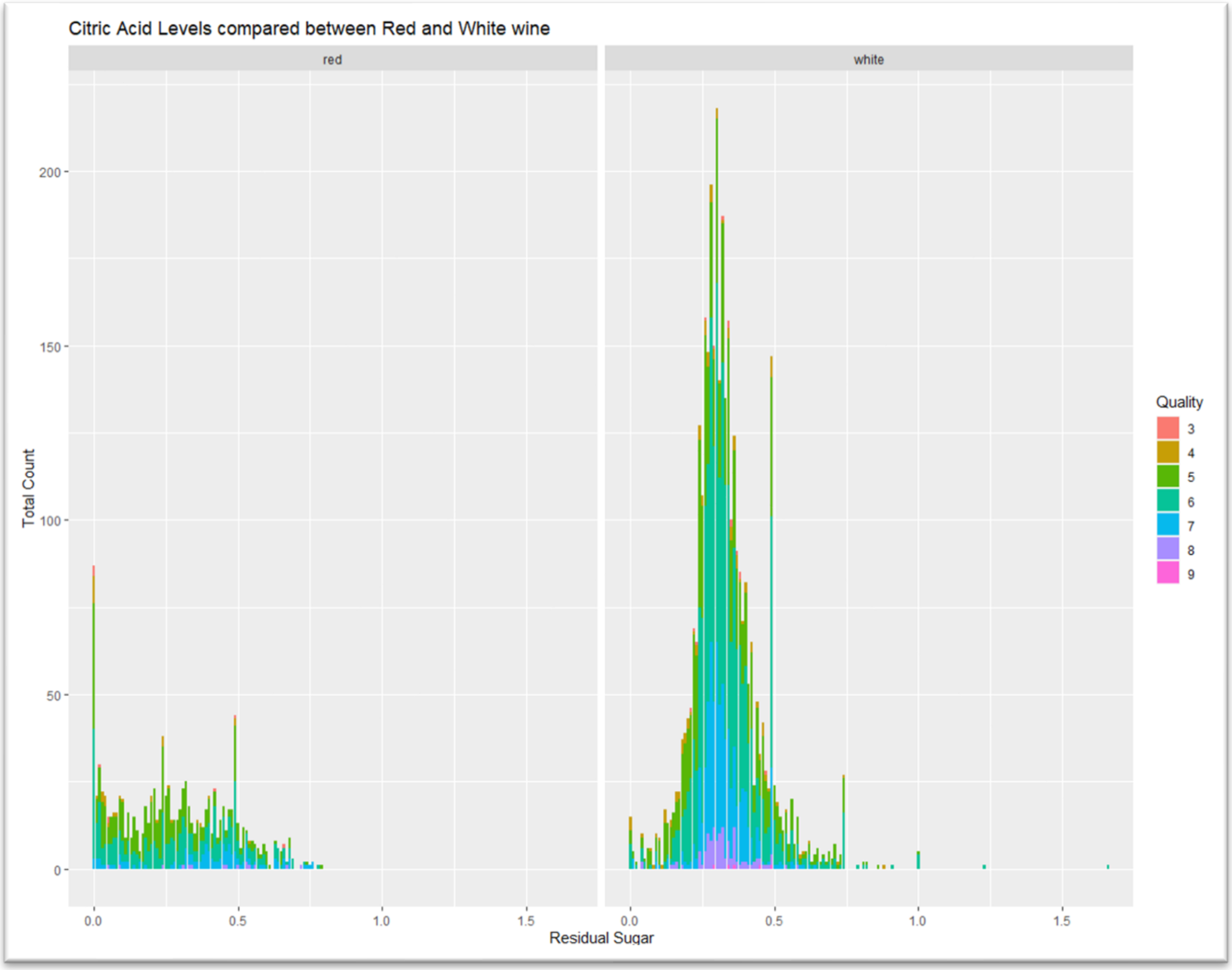
Count of wines across all pH Levels.



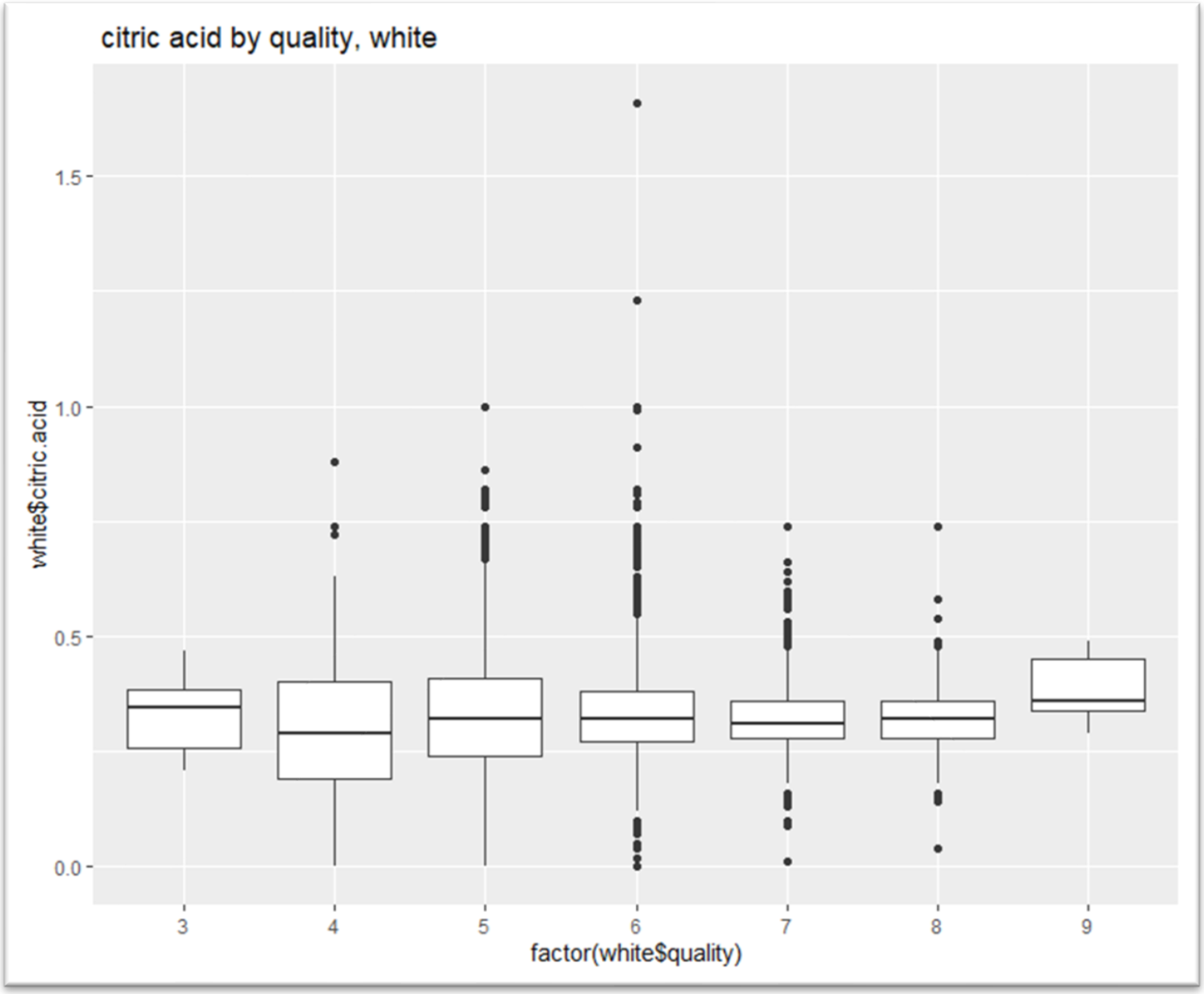
Count of wines across all sulphate levels.



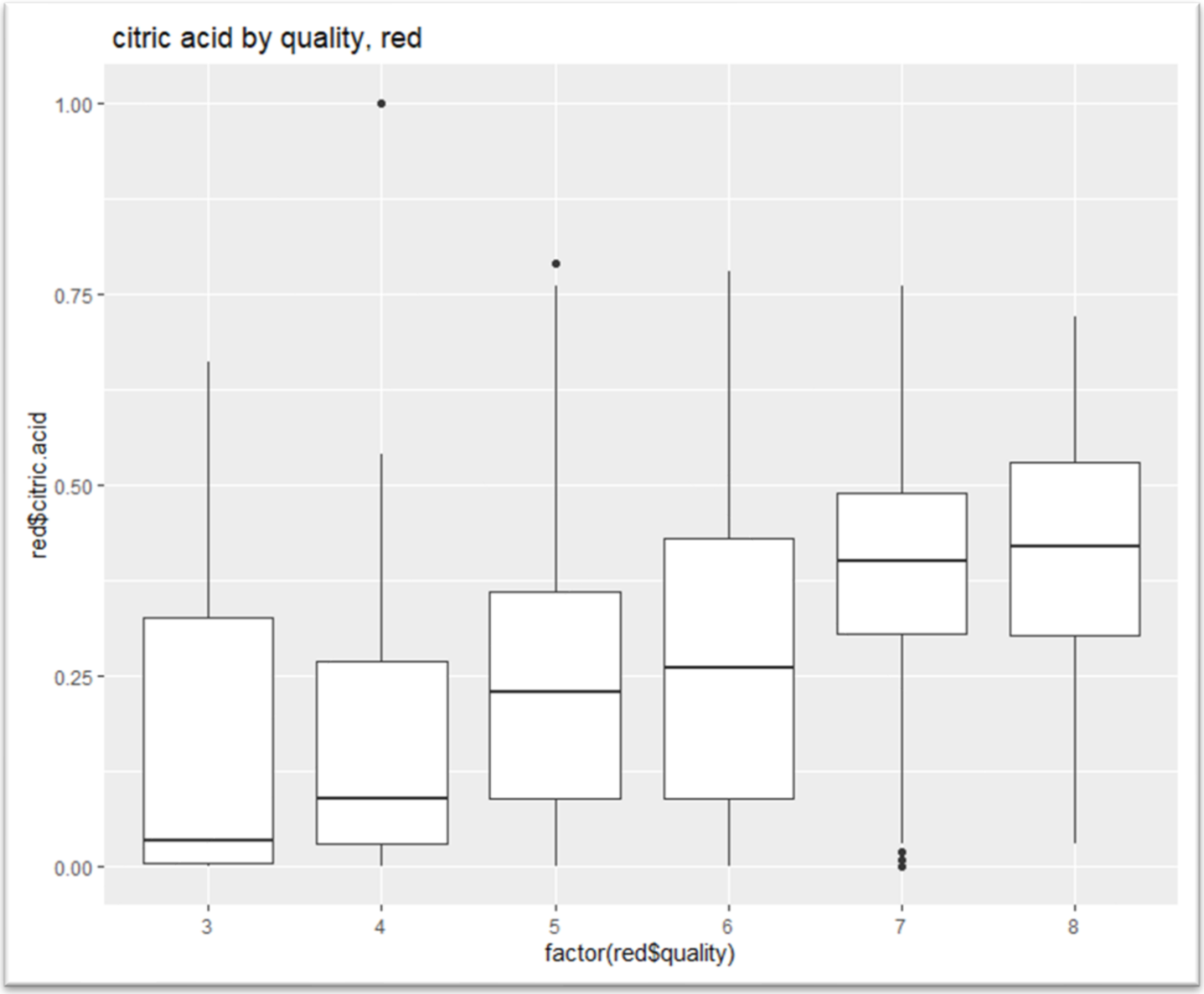
Count of wines across all levels of Residual Sugar.



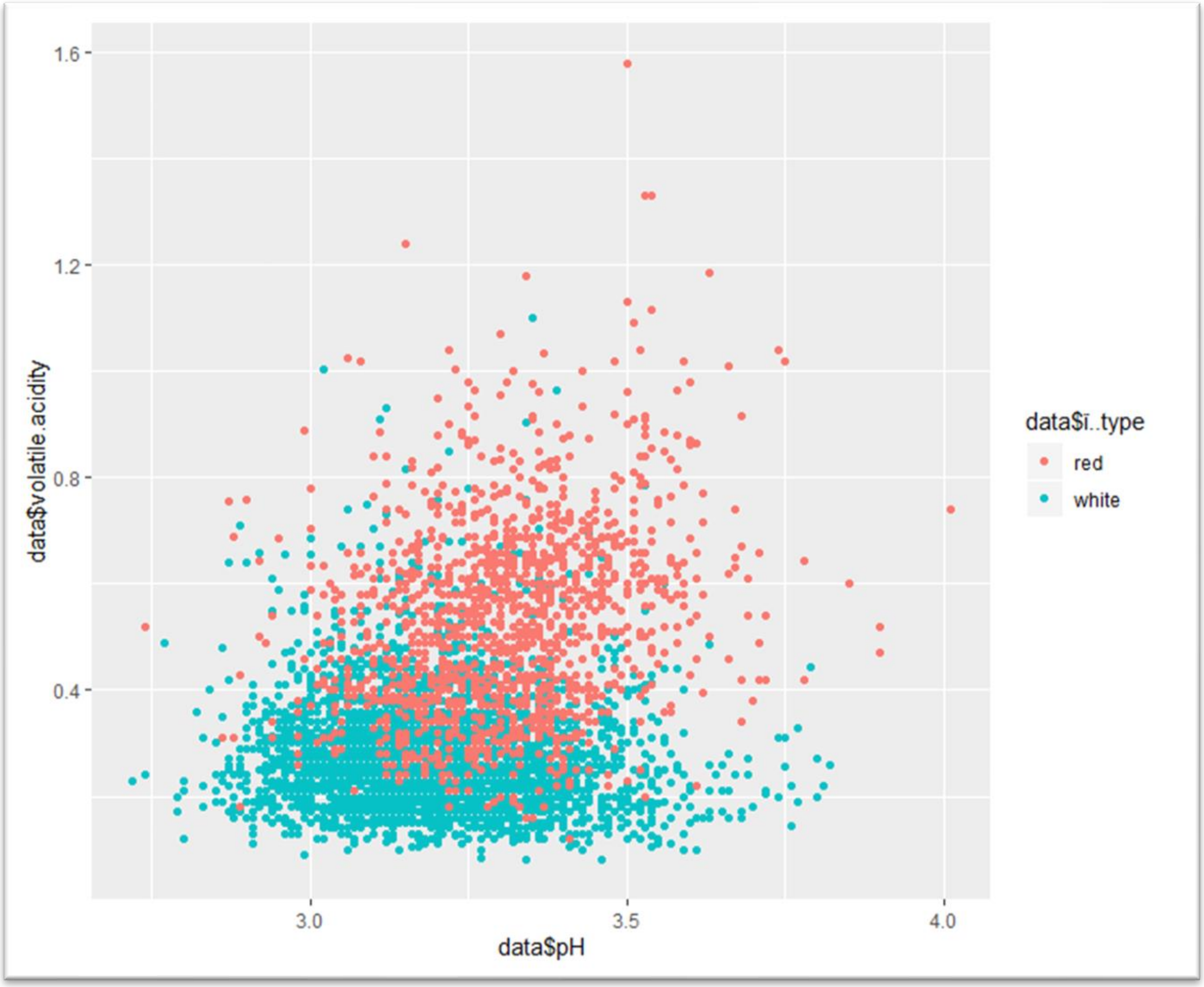
Citric Acid levels between Red and White wines.



Citric Acid level of white wines by quality score.



Citric acid level of red wines by quality score.



Volatile Acidity vs. PH level for red and white wines.