Wine Classification Capstone Project

José Caloca

15/02/2021

Introduction

The purpose of this project is to determine the Class (quality) of wine from 13 attributes. The data that is examined in this project is provided by UCI Machine Learning Repository. Each wine was grown in the same region in Italy although they were processed by three different cultivars. The cultivars are represented by three Classes: 1, 2, or 3. The columns of this dataset are as follows:

- **Class**: This is what we are attempting to predict. Factor
- **Alcohol**: Numeric
- **Malic Acid**: Numeric
- **Ash**: Numeric
- **Alcalinity** of Ash: Numeric
- **Magnesium**: Integer
- **Total Phenols**: Numeric
- **Flavanoids**: Numeric
- **Nonflavanoids Phenols**: Numeric
- **Proanthocyanins**: Numeric
- **Color Intensity**: Numeric
- **Hue**: Numeric
- **Proteine Concentration**: Numeric
- **Proline**: Numeric

Exploring the Data Set

We load the following libraries:

- **Tidyverse**: The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.
- **Caret**: The caret package (short for Classification And REgression Training) contains functions to streamline the model training process for complex regression and classification problems. The package utilizes a number of R packages but tries not to load them all at package start-up (by removing formal package dependencies, the package startup time can be greatly decreased). The package "suggests" field includes 30 packages. caret loads packages as needed and assumes that they are installed. If a modeling package is missing, there is a prompt to install it.
- **Datatable**: is a package is used for creating graphics with details from statistical tests included in the information-rich plots themselves.

- **Ggstatsplot**: This package is already contained in the Tidyverse package but it provides very usefull tools for data visualisation.
- **Rpart.plot**: Used for plotting rpart decision trees models.

First, we look at some main summary statistics of our dataset and get a picture of the distribution of each variables.

	Class	Alcohol	Malic_Acid	Ash	Alcalinity_of_Ash
Min.	59	11.03000	0.740000	1.360000	10.60000
1st Qu.	71	12.36250	1.602500	2.210000	17.20000
Median	48	13.05000	1.865000	2.360000	19.50000
Mean	59	13.00062	2.336348	2.366517	19.49494
3rd Qu.	71	13.67750	3.082500	2.557500	21.50000
Max.	48	14.83000	5.800000	3.230000	30.00000

	Magnesium	Total_Phenols	Flavanoids	Nonflavanoids_Phenols	Proanthocyanins
Min.	70.00000	0.980000	0.34000	0.1300000	0.410000
1st Qu.	88.00000	1.742500	1.20500	0.2700000	1.250000
Median	98.00000	2.355000	2.13500	0.3400000	1.555000
Mean	99.74157	2.295112	2.02927	0.3618539	1.590899
3rd Qu.	107.00000	2.800000	2.87500	0.4375000	1.950000
Max.	162.00000	3.880000	5.08000	0.6600000	3.580000

	Color_Intensity	Hue	Proteine_Concentration	Proline
Min.	1.28000	0.4800000	1.270000	278.0000
1st Qu.	3.22000	0.7825000	1.937500	500.5000
Median	4.69000	0.9650000	2.780000	673.5000
Mean	5.05809	0.9574494	2.611685	746.8933
3rd Qu.	6.20000	1.1200000	3.170000	985.0000
Max.	13.00000	1.7100000	4.000000	1680.0000

We want to get an idea of the percentage of wines that are Class 1,2, or 3.

```
## The percentage of Class 1 is:  0.3314607
```

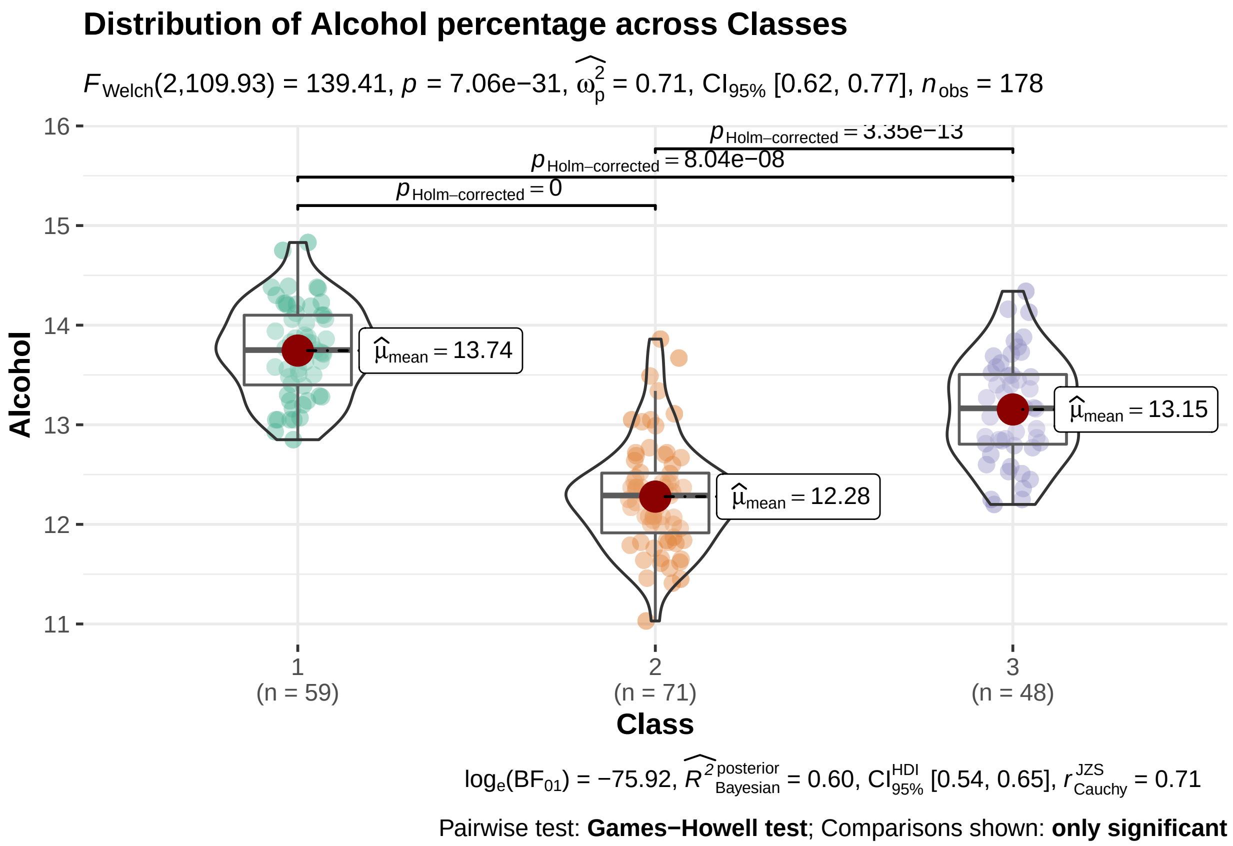
```
## The percentage of Class 2 is:  0.3988764
```

```
## The percentage of Class 3 is:  0.2696629
```

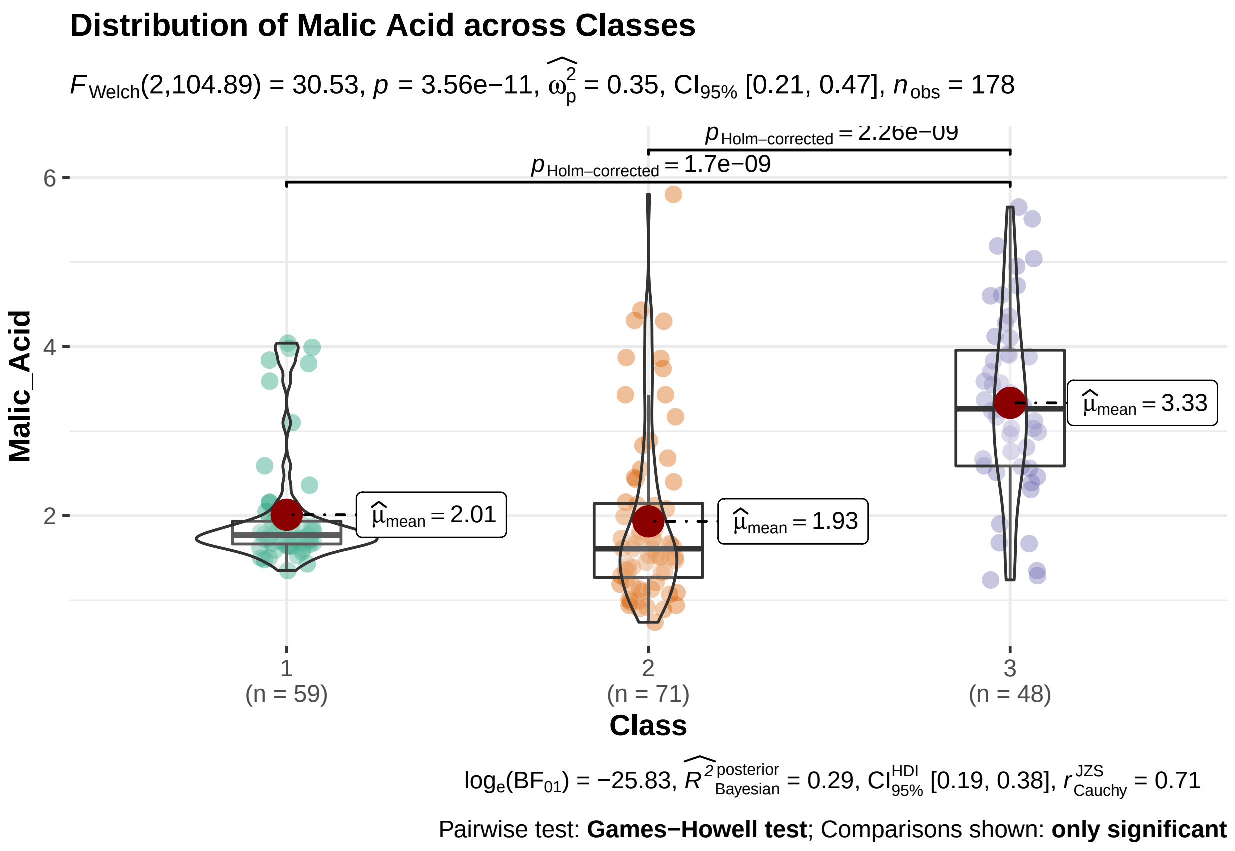
In order to understand our dataset we are interested in visualizing the variables by class. To do this, we will make violin and box plots to analyze the distribution of each variable using the *ggstatsplot* package. By default we will obtain the following information in our graphs:

- Raw data + distributions
- Descriptive statistics
- Statistic + p-value

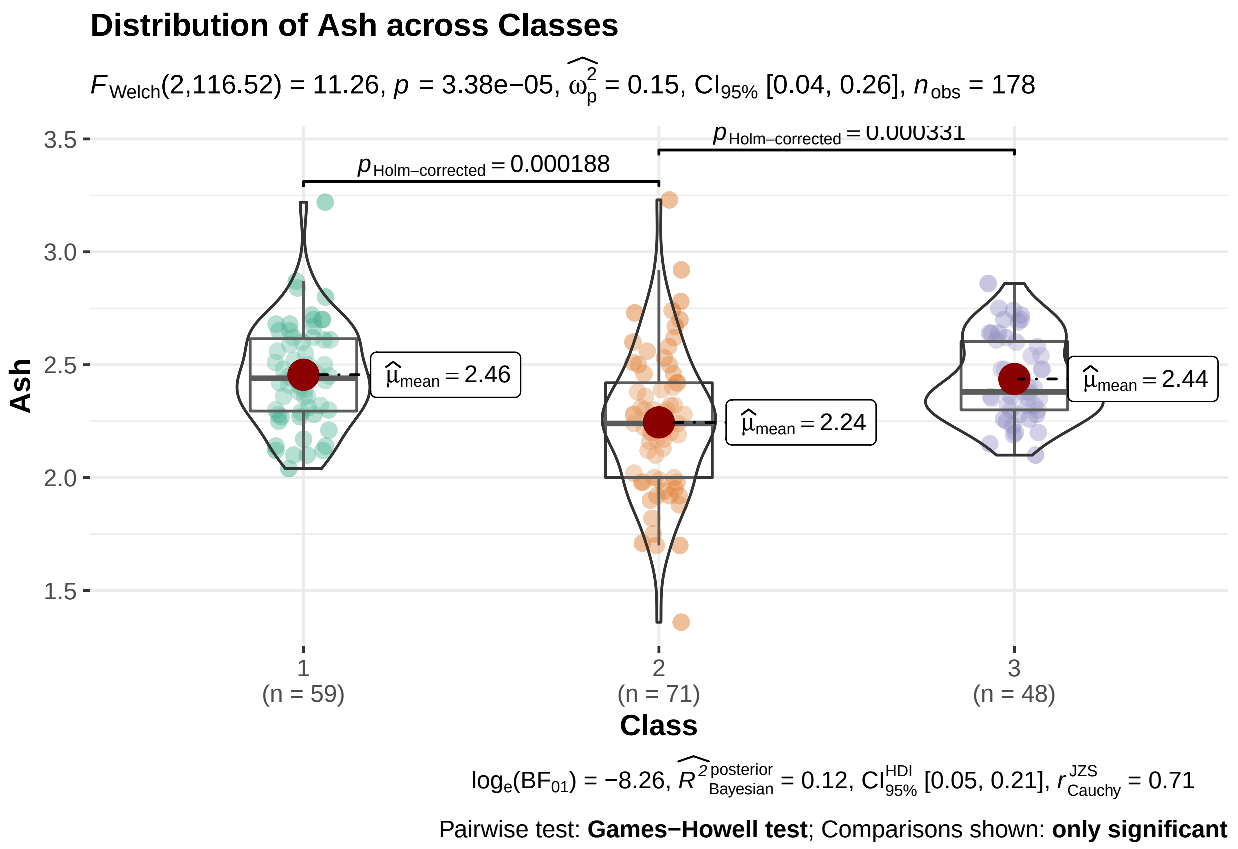
- Effect size + CIs
- Pairwise comparisons
- Bayesian hypothesis-testing
- Bayesian estimation



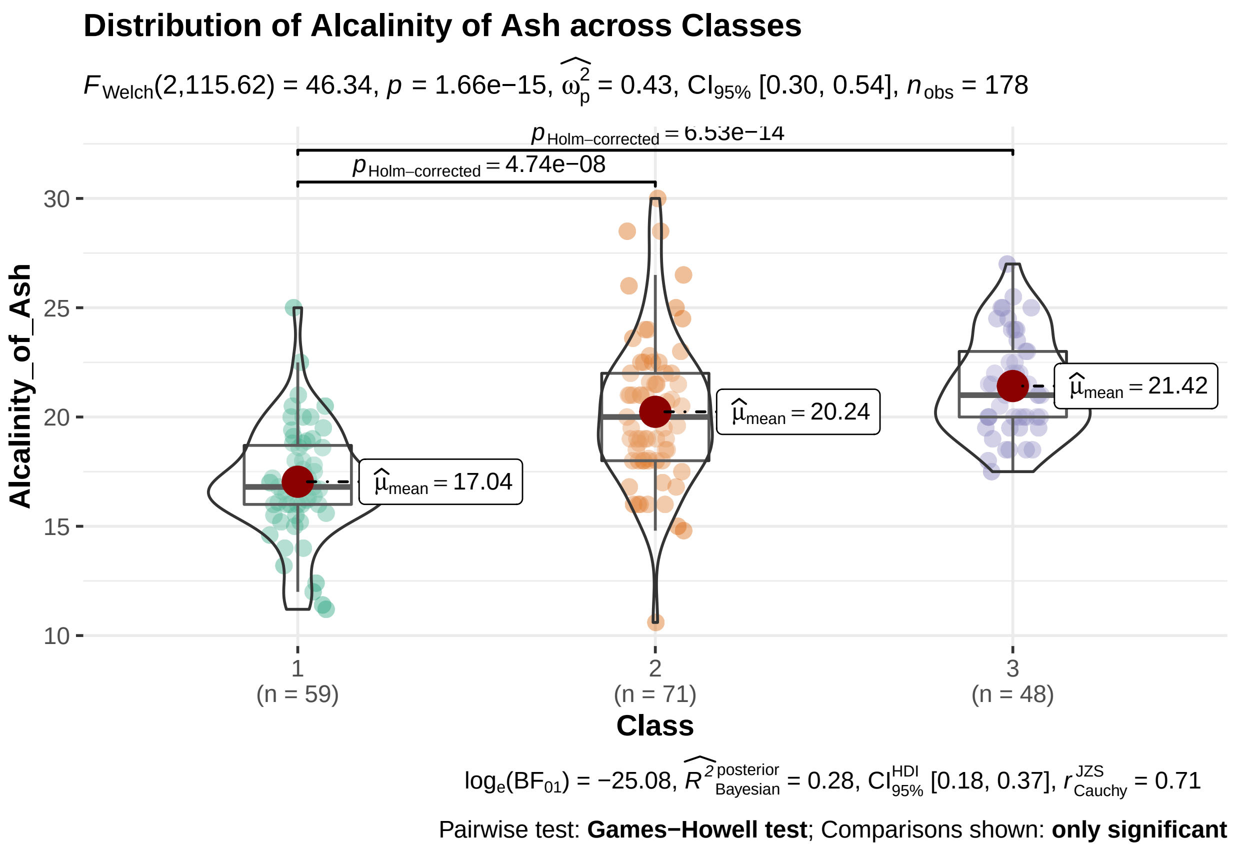
We can observe from comparing variable alcohol % by class that wines of class 1 generally contain the highest percentage of alcohol as the observations fall right most on the interval and are also most concentrated around 13.75 % alcohol. Class 2 wines lie in the middle of the alcohol % interval, concentrated between 12.75% and 13.5% as there are two peaks. Class 3 wines are the lowest in alcohol % as the observations lie on the left most side of the interval, these wines typically contain about 12.25% alcohol.



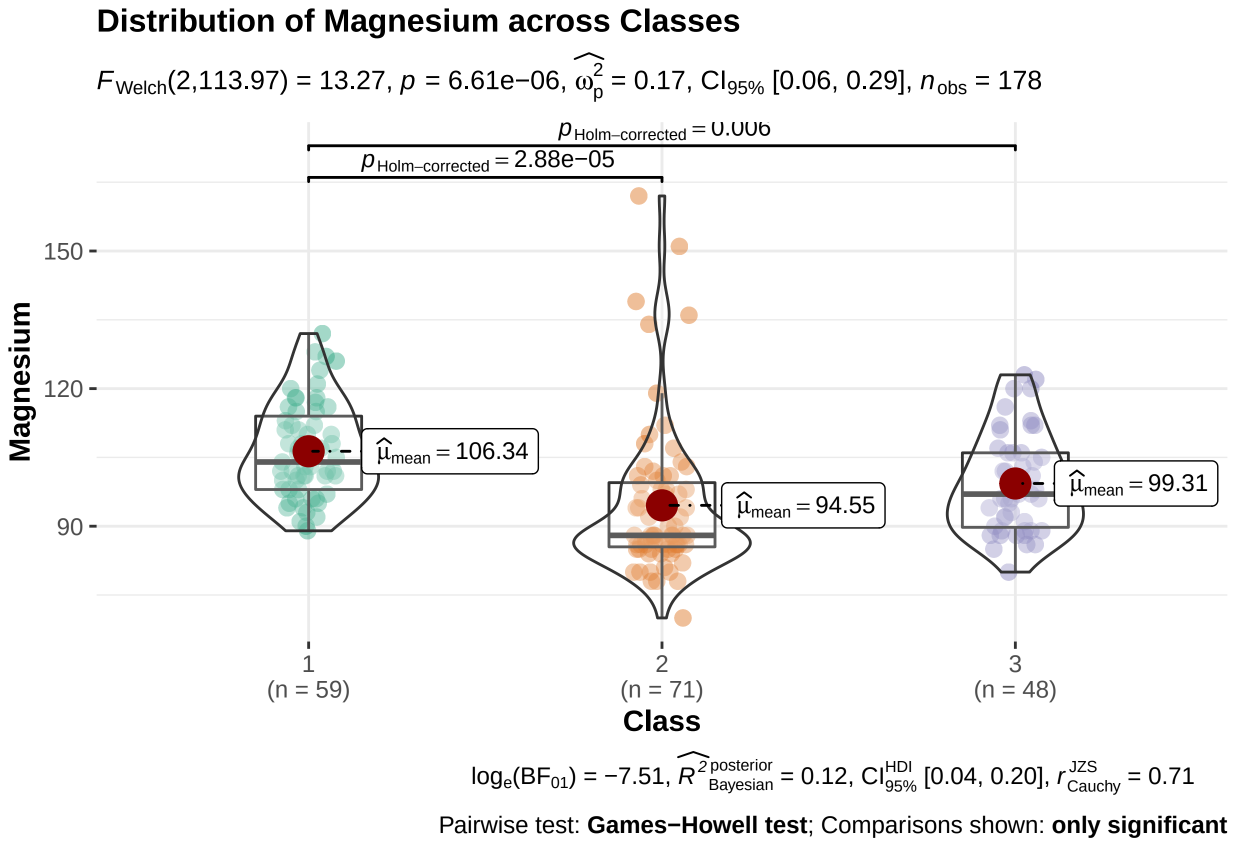
Malic acid contributes to the sour taste of fruits and is used as a food additive, the more malic acid the sourer the wine. All three classes of wine contain malic acid varying from .5 to 5.75 g/l on the interval. Conveniently, the density plot allows us to easily spot the largest concentrations for each of the classes. Class 1 wines have the greatest concentration at around 1.75 g/l malic acid, class 2 with 1.5 g/l and lastly class 3 are most likely to contain 3.25 g/l malic acid. Class 3 wines are typically more sour.



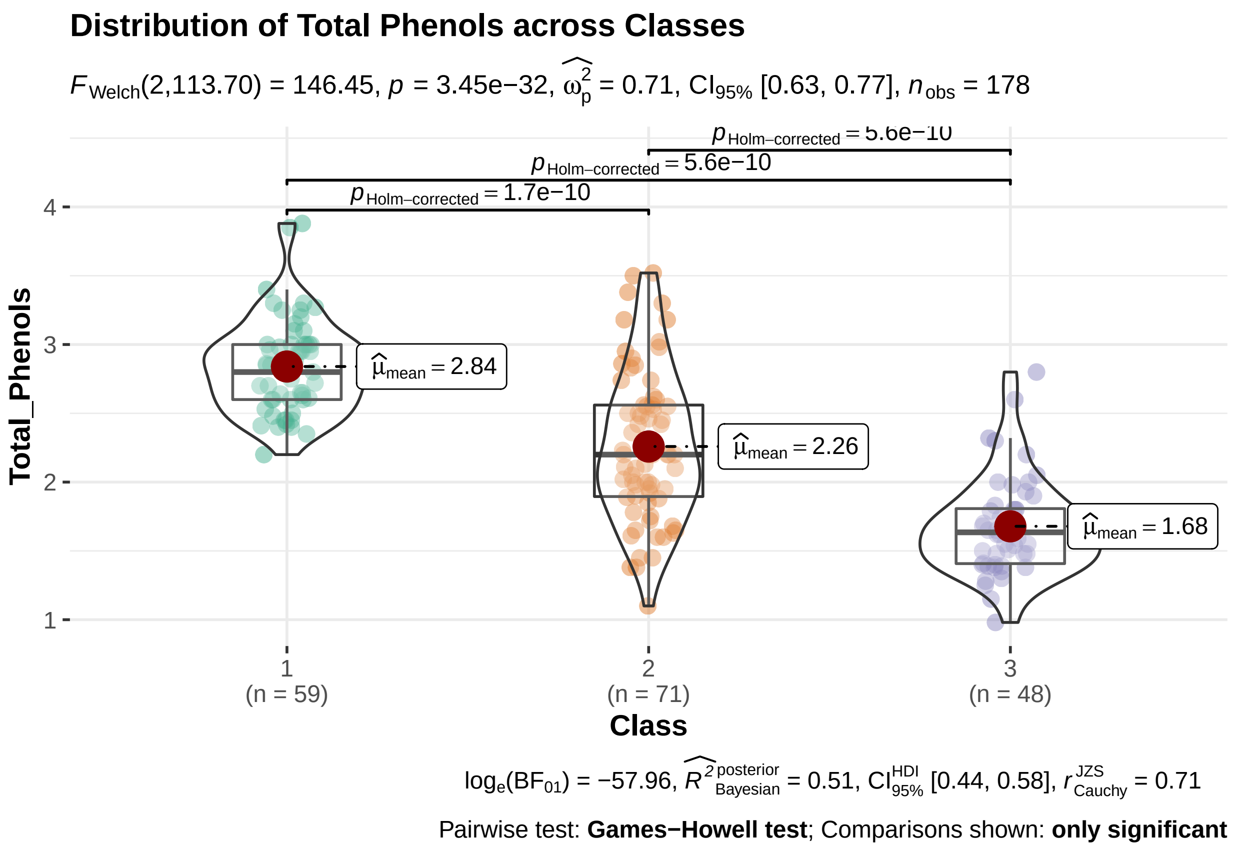
Ash content is an important indicator in wine quality because of the link between minerals and trace elements. All three classes of wine are normally distributed around the center of the ash interval ranging from 1.25mg/l to 3.75 mg/l. Class 1 wines are most likely to contain 2.4mg/l of ash, 2.25mg/l for class 2 and 2.35 mg/l for class 3 according to the peaks on the density plot.



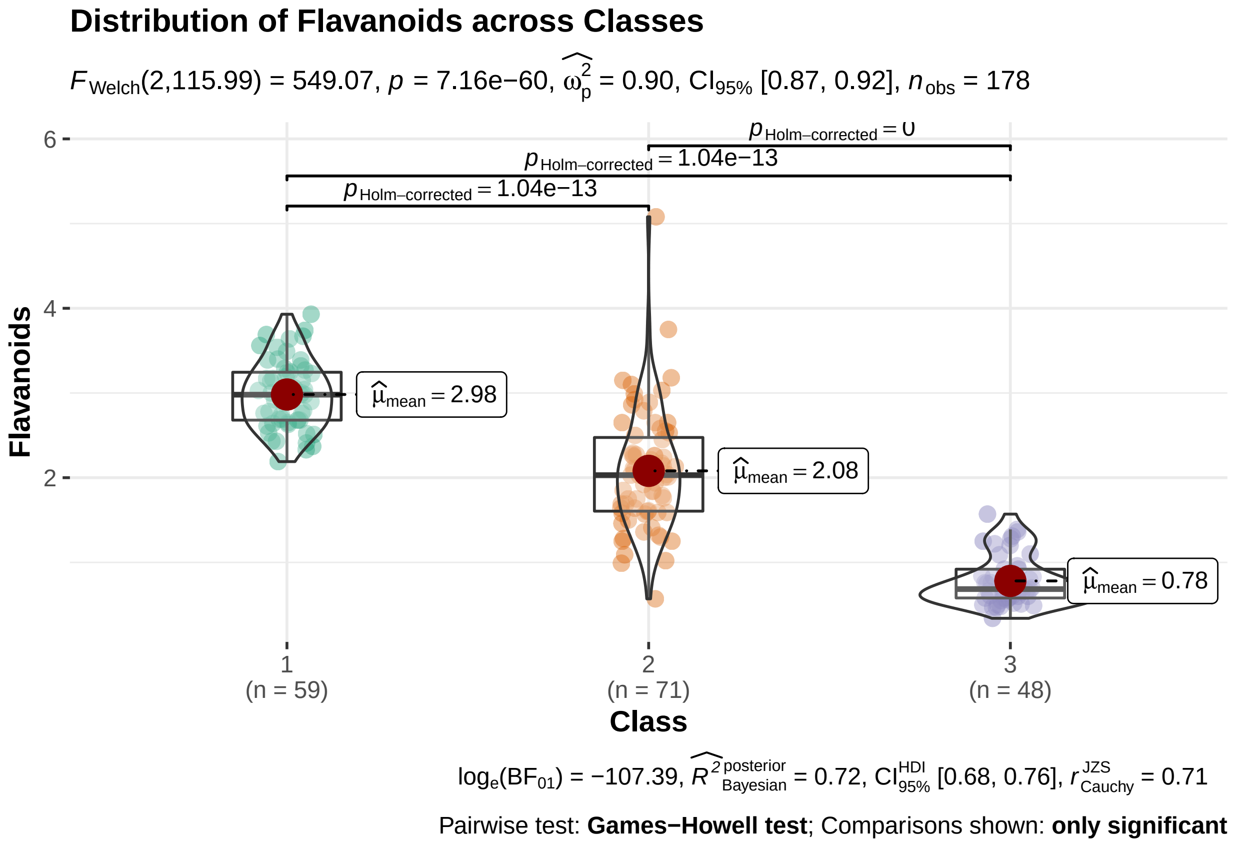
The alcalinity of the ash is defined as the sum of cations, other than the ammonium ion, combined with the organic acids dissolved in water measured by ml/l. Observations for class 1 wines lie on the left side of the interval for alcalinity of ash, the majority of wines under this class have an alcalinity of 17. Class 2 wine observations span between 11 and 30, the entire interval, but the greatest concentration of wines under category 2 contain alcalinity of around 19. Lastly, class 3 wines contain the greatest amount of alcalinity on average with the largest concentration at around 21, the observations for class 3 wines lie between 15 and 29.



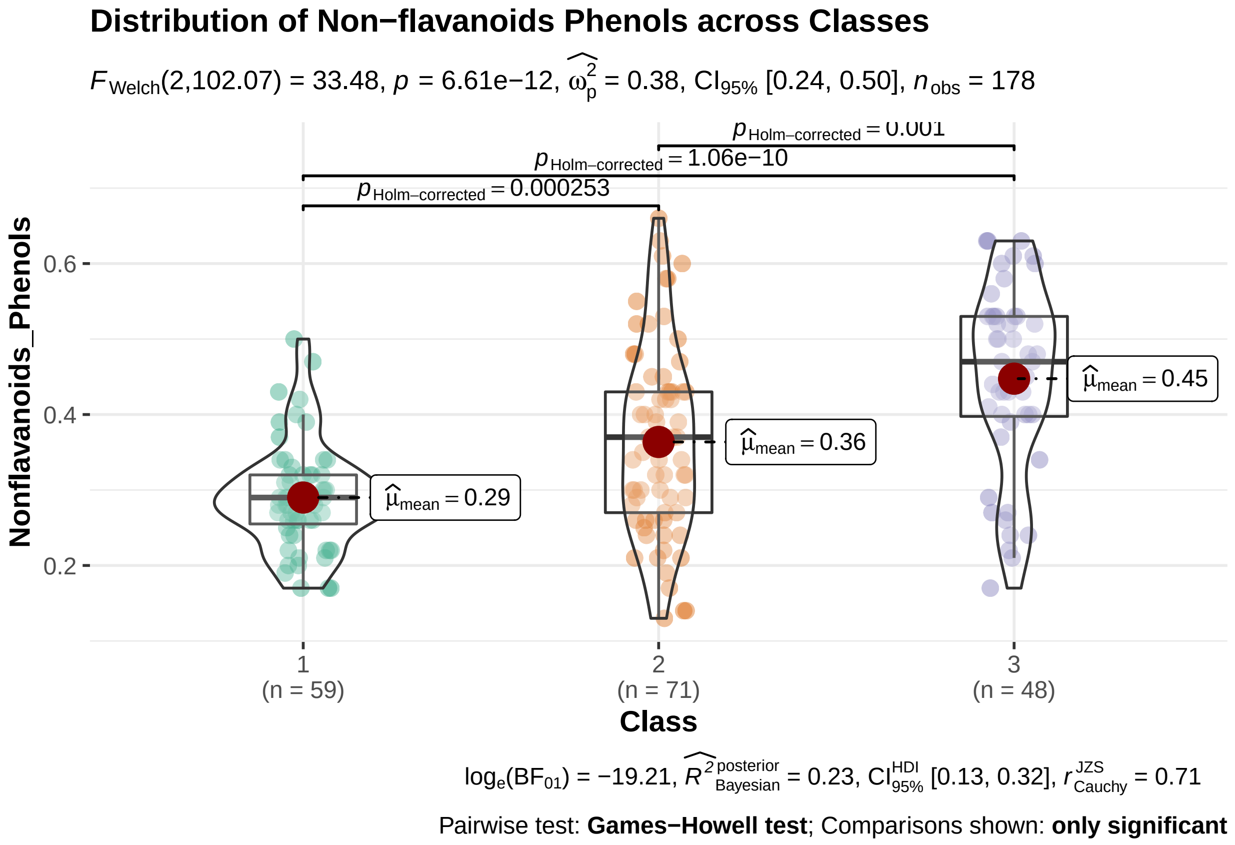
Magnesium is an essential mineral in the human body that promotes energy metabolism and is weakly alkaline. All three classes contain magnesium contents that typically lie on the lower half of the interval between 70 and 162. In order of highest concentrations from least to greatest is class 2 with magnesium amounts around 85, followed by class 3 at 95 then lastly class 1 at 105. Class 1 wines contain the largest amount of magnesium amidst the three classes.



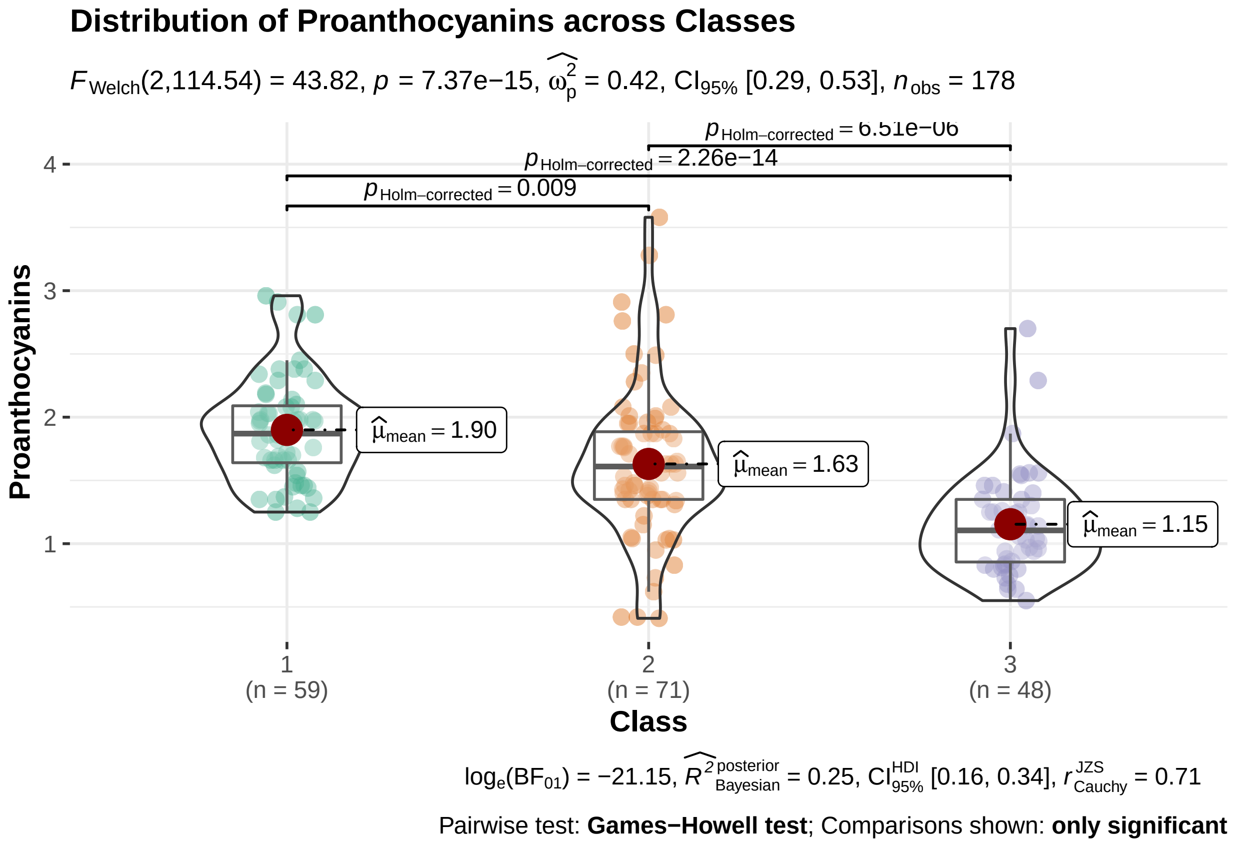
Total phenols account for molecules containing polyphenolic substances, which have a bitter taste and affect the taste, color and taste of the wine, and contribute to the nutritional value of the wine. Class 2 wines are distributed throughout the entire interval between 1 and 3.9. Class 3 wines are skewed towards the left on the interval, while class 1 are the opposite, mostly lying on the right side of the interval. The highest concentrations pertaining to each class is as follows: class 1 is around 2.9, class 2 is around 2.1 while class 3 is around 1.5. Class 1 wines typically have the greatest number of total phenols.



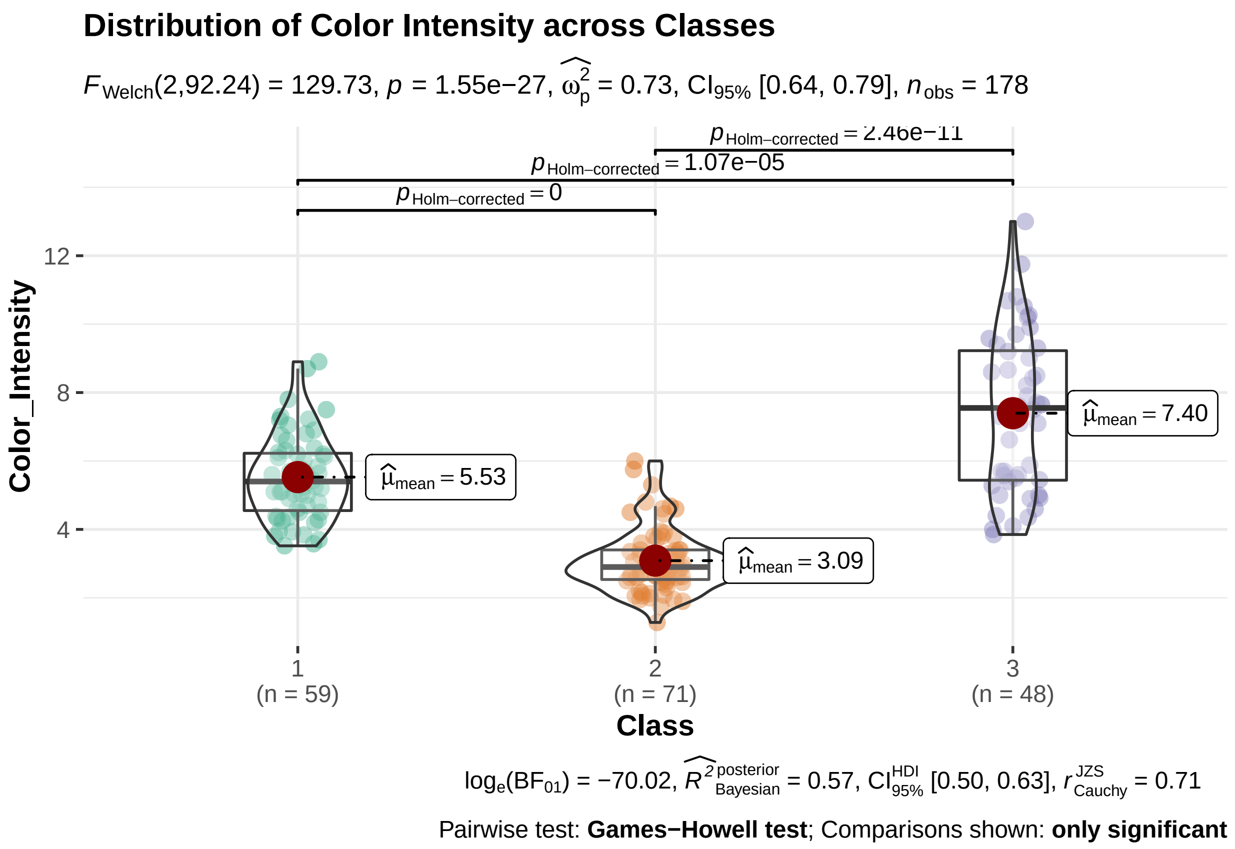
Flavanoids are antioxidants promote anti-aging and are beneficial for the heart, they are rich in aroma and bitter. Class 3 wine observations lie entirely on the far-left side of the interval, the highest concentration is situated amongst a rating of .6 flavanoids. Wines categorized under class 2 contain between 0 to 5, the entire length of the interval but the highest concentration is observed at around 2. Class 1 observations are normally distributed towards the middle of the interval, the peak is situated at 2.9. Class 1 wines are most likely to have the greatest amount of flavanoids.



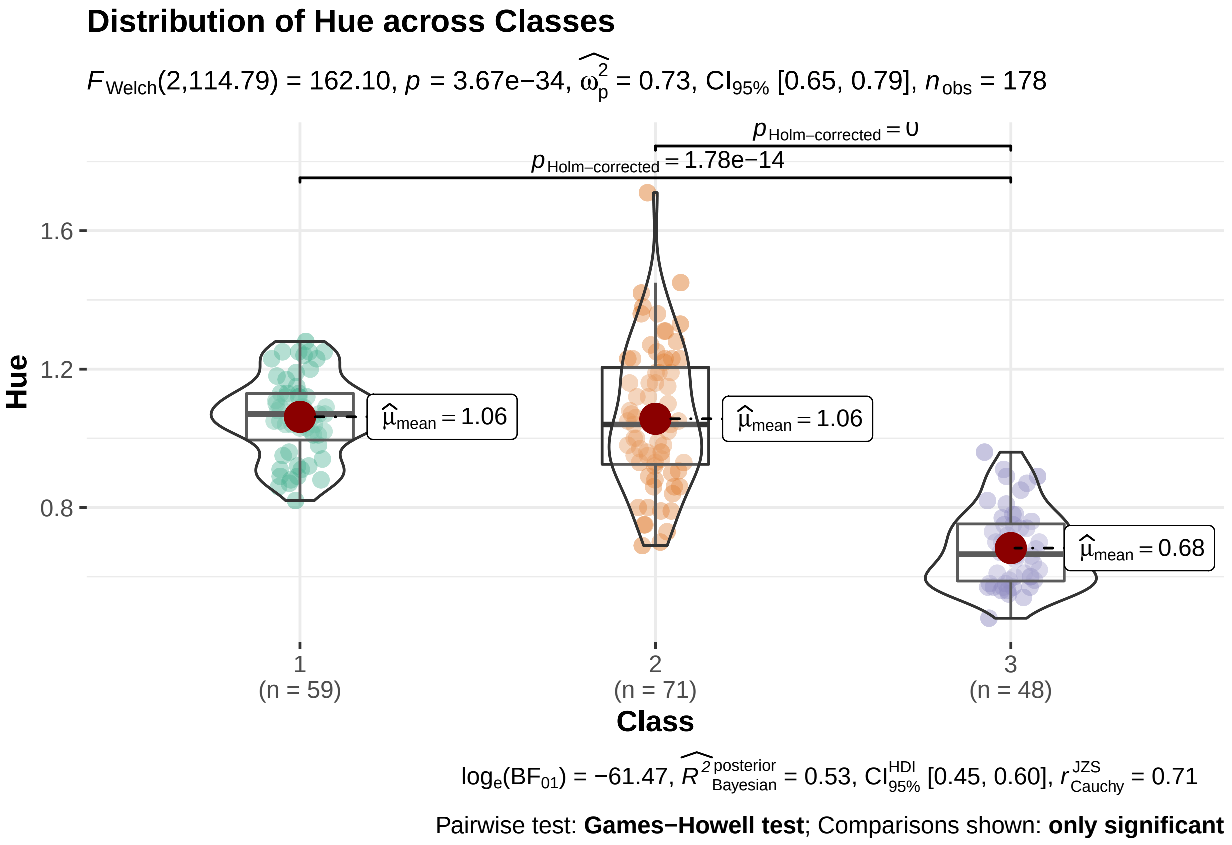
Nonflavanoid phenols is an aromatic gas with oxidation resistance and is weakly acidic. Class 1 wines are largely distributed towards the left side of the interval while both class 2 and 3 wines are spread across the entire interval. The largest concentration of class 1 wines lies at a measurement of about .28 nonflavanoid phenols. The concentration for class 2 wines is a plateau that spans between .28 to .40 while class 3 wines are most concentrated around a rating of .5. It is evident that class 3 wines are most likely to contain the greatest amount of Nonflavanoid phenols.



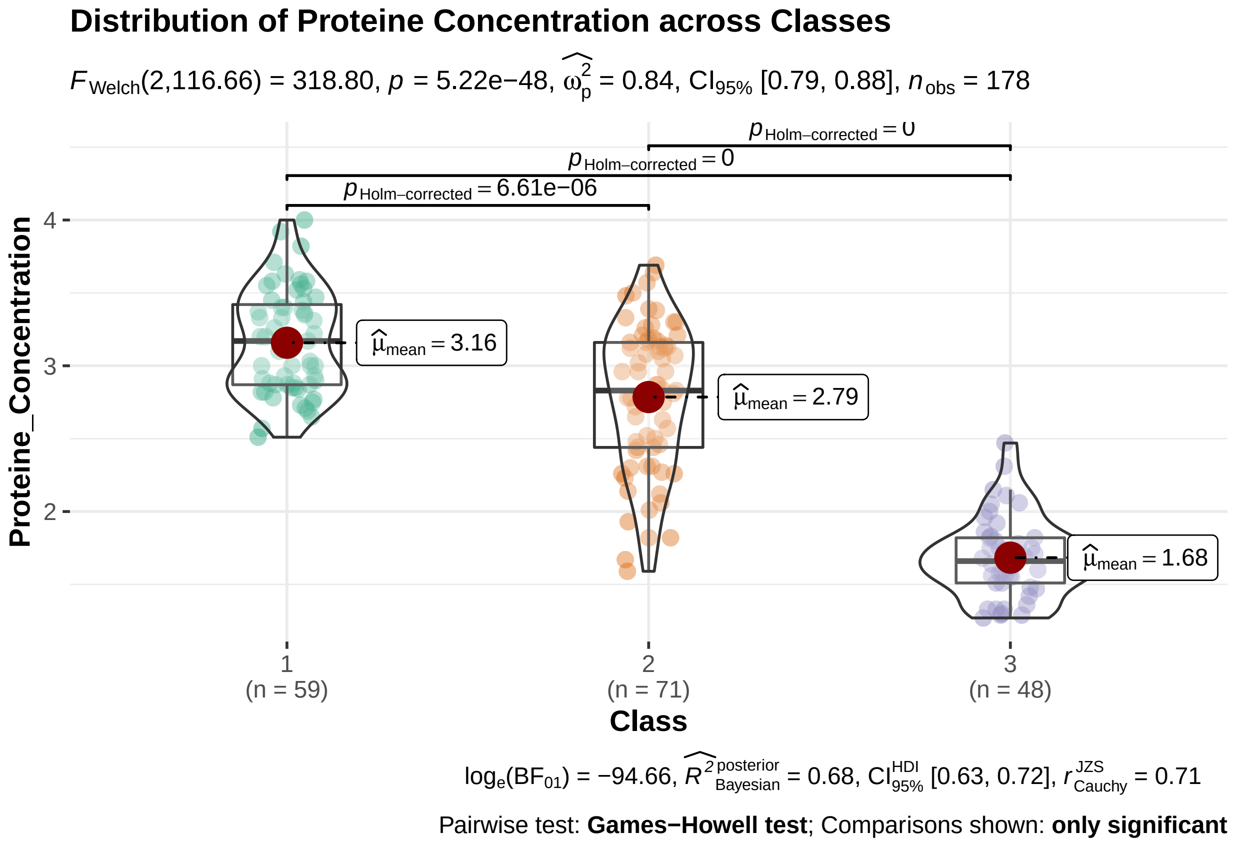
Proanthocyanins are a bioflavonoid compound which is a natural antioxidant with a slightly bitter aroma. All three classes are almost entirely distributed throughout the interval. Class 3 wines typically have the smallest amount of Proanthocyanins while class 1 wines contain the largest amount on average when observing their distribution peaks.



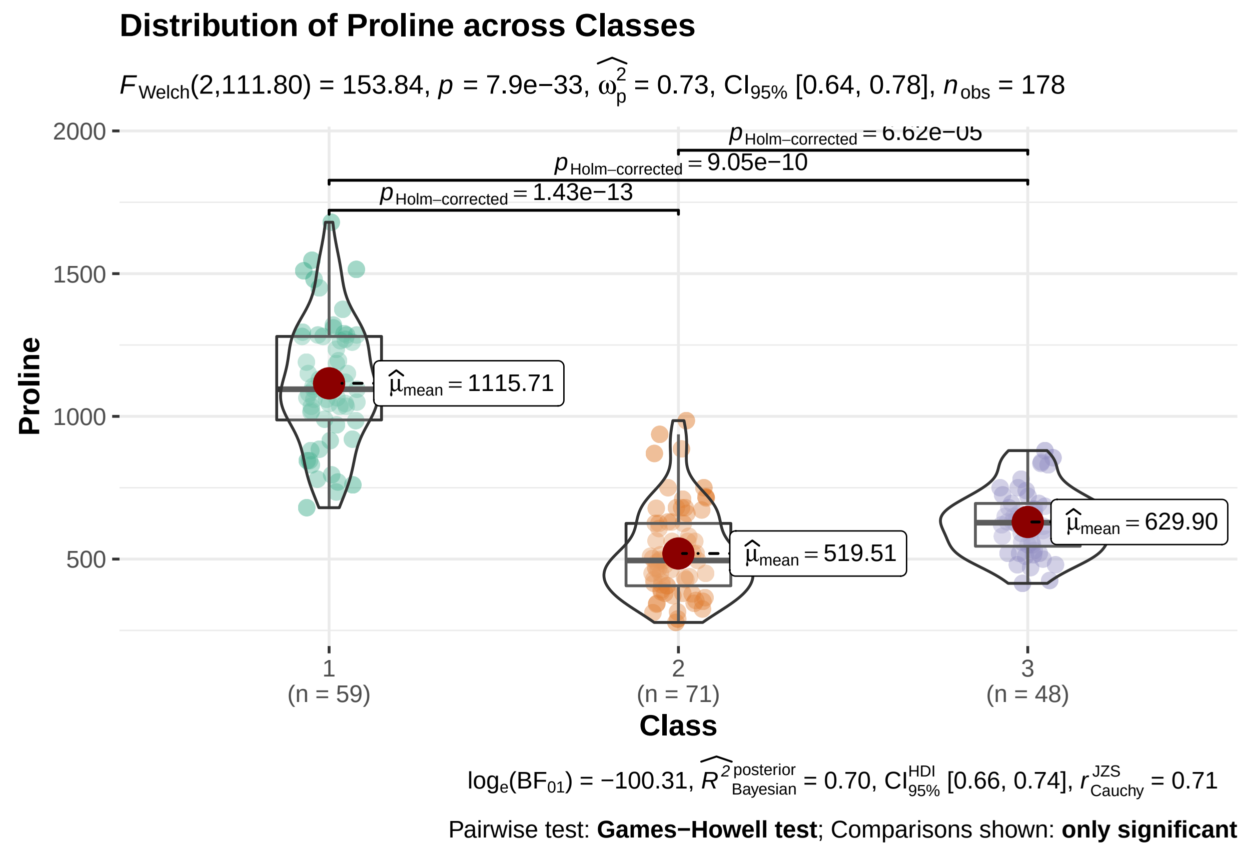
Color intensity measures the degree of color shade, lower numbers correspond to a lighter intensity of wine while higher numbers represent thicker shades. The longer the wine and grape juice are involved during the wine making process, the thicker the taste. Class 3 wines are normally distributed throughout the interval of color intensity from 0 to 13, there are two peaks situated between ratings 5 and 8, half of the wines from this class have the greatest color intensity of the three classes. Observations for class 2 wines are almost entirely concentrated around a 2 rating. Class 3 wines are greatly concentrated around a 6 rating, the wines under this class fall between 2 and 10.



Hue variable measures color vividness and degree or warmth and coldness, hue is used to measure the age and variety of the wine. Red wines that are aged longer have a yellow hue and increased transparency. Class 3 wines have the lowest amount of hue as many of the observations are skewed to the left, the majority lie at about .60 hue. Observations for class 2 wines are distributed all across the interval, the tallest peak of class 2 wines is observed at .90 hue. Class 1 wines are distributed in the middle of the interval, most class 1 wines contain a rating of about 1.10 hue.



OD280/OD315 is the method for determining the protein content of wine. Class 3 wines contain the smallest amount of protein content, the highest concentration of class 3 wines have a reading of about 1.6. Class 2 wines have protein contents varying drastically between 1.25 to 4, but the majority have protein contents of about 3.25. Class 1 wines have the greatest amount of protein content, there are two large concentrations, the higher peak is located at about 2.8 while the smaller peak lies at 3.4.



Proline is the main amino acid in red wine and imperative to the flavor and nutrition of wine. Class 1 wines have the highest amount of proline as the majority of observations lie on the right side of the interval, their density peak lies at around the 1100 measure. Both class 2 and class 3 wines are situated on the left side of the interval. The greatest concentration of class 2 wines lies around 475 proline, while for class 3 wines, it is about 625 when we observe each peak.

The rest of the variables have similar distributions by class. There is nothing that strikes out as different.

Methods/Analysis

To create a predictive model, we create three simple (and naive) models followed by two machine learning models. For the machine learning algorithms we will create our *train_set* that contains 75% of the observations, and *test_set* contains the remaining 25%.

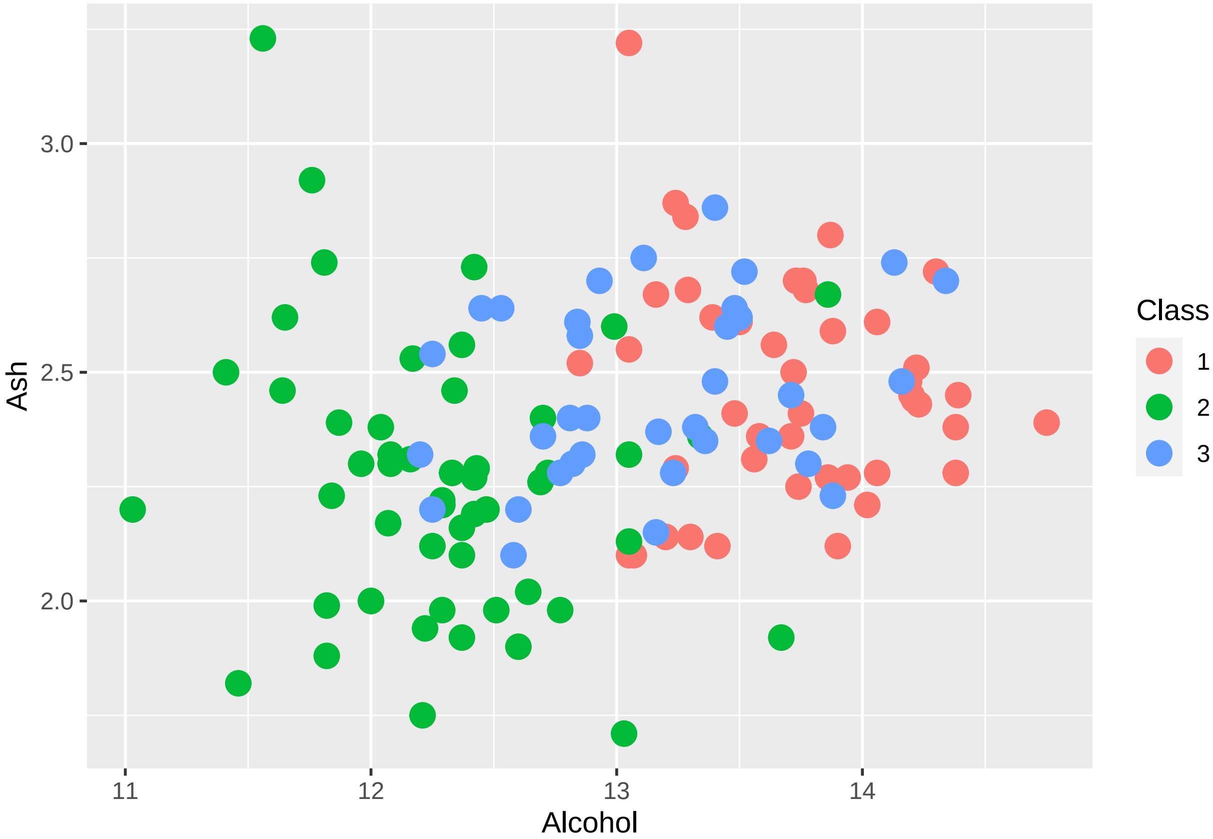
```
#the names for the columns I specified were bad
colnames(wine_data) <- make.names(colnames(wine_data))
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(wine_data$Class, times = 1, p = 0.25, list = FALSE)

train_set <- wine_data[-test_index]
test_set <- wine_data[test_index,]
```

If/Else Models

Alcohol Vs Ash

First, let's plot Alcohol vs Ash. Ash's distribution by class showed very little differences. This initial model is meant to be simple.

```
train_set %>%
  ggplot(aes(x = Alcohol, y = Ash, col = Class)) +
  geom_point(size = 4)
```



There appears to be groups in the data. The groups are not exclusive, which implies that the If/Else model cannot be 100% accurate.

The first model is built as follows: Wines with alcohol less than or equal to 12.75 are classified as "Class 2". Otherwise, Wines with Alcohol less than or equal to 13.75 are classified as "Class 3". Otherwise, the wine is classified as "Class 1". These cut-off values are visually inspected.

```
alcohol_model <- function(val){
    if(val <= 12.75){
        2
    } else if(val <= 13.75){
        3
    } else 1
}

y_hat_alc <- sapply(test_set$Alcohol, alcohol_model)
y_hat_alc <- as.factor(y_hat_alc)
```

```
method1 <- mean(y_hat_alc == test_set$Class)*100
method1
```

```
## [1] 77.77778
```

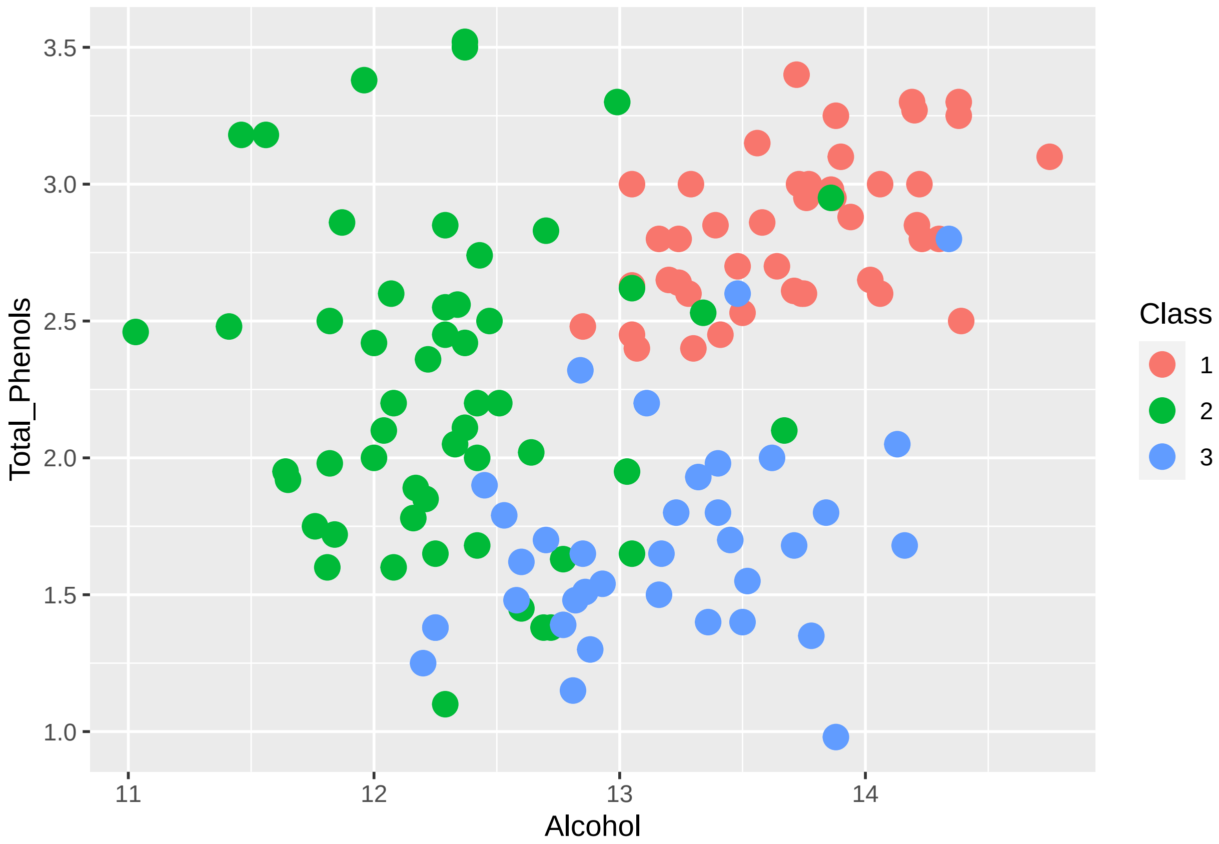
Methods	Accuracy
Just Alcohol	77.77778

With this simple model, we get an accuracy of about 77.7777778‘%.

Alcohol Vs Total Phenols

For the next model, we consider Alcohol and Total Phenols. We can plot the relationship and identify clusters.

```
train_set %>%
  ggplot(aes(x = Alcohol, y = Total_Phenols, col = Class)) +
  geom_point(size = 4)
```



We can identify some clusters in the graph. The groups are a bit more different compared to the Alcohol vs Ash analysis so we will consider Total Phenols when constructing the If/Else model.

The next model is built as follows: Wines with alcohol less than or equal to 12.75 are classified as “Class 2”. Otherwise, Wines with Total Phenols greater than or equal to 2.5 are classified as “Class 1”. Otherwise, the Wine is classified as “Class 3”. These cut-off values are visually inspected.

```
alc_phen <- function(alc, phen){
  if (alc <= 12.75){
    2
  } else if(phen >= 2.5){
    1
  } else 3
}

y_hat_alc_phen <- mapply(alc_phen, test_set$Alcohol, test_set$Total_Phenols)
y_hat_alc_phen <- as.factor(y_hat_alc_phen)
method2 <- mean(y_hat_alc_phen == test_set$Class)*100
method2
```

```
## [1] 80
```

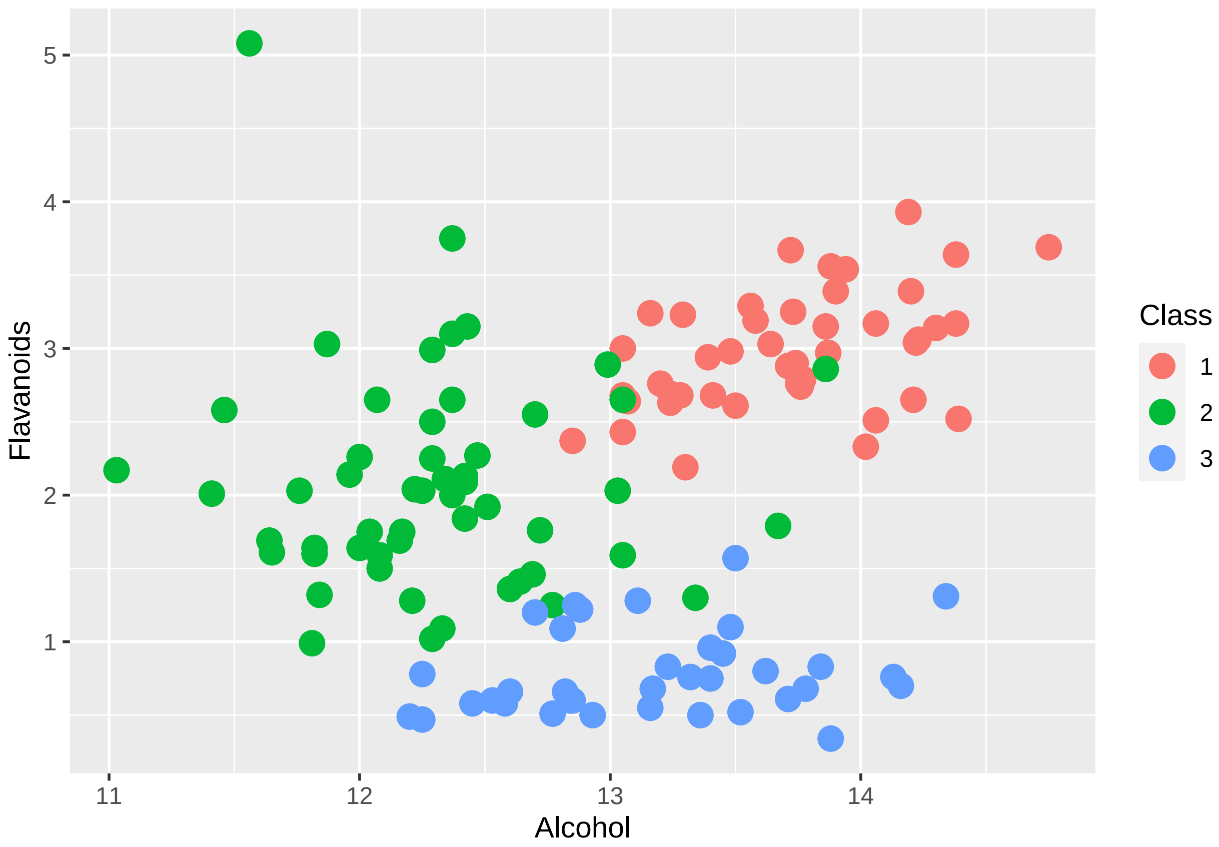
Methods	Accuracy
Just Alcohol	77.77778
Alcohol and Total Phenols	80.00000

With this improved model, our accuracy increases to 80%.

Alcohol Vs Flavanoids

For the final If/Else model, we consider Alcohol and Flavanoids. We plot the relationship as follows:

```
train_set %>%
  ggplot(aes(x = Alcohol, y = Flavanoids, col = Class)) +
  geom_point(size = 4)
```



In the above graph we can appreciate some clusters. For this If/Else model, we consider both Alcohol and Flavanoids.

The model is built as follows: Wines with flavanoids greater than 2.5 are classified as "Class 1". Otherwise, Wines with alcohol less than or equal to 12.5 are classified as "Class 2". Otherwise, the wine is labeled as "Class 3".

```
alc_flav <- function(alc, flav){
  if(flav > 2.5){
    1
  } else if(alc <= 12.5){
    2
  } else
    3
}

y_hat_alc_flav <- mapply(alc_flav, test_set$Alcohol, test_set$Flavanoids)
y_hat_alc_flav <- as.factor(y_hat_alc_flav)
method3 <- mean(y_hat_alc_flav == test_set$Class)*100
method3
```

```
## [1] 77.77778
```

Methods	Accuracy
Just Alcohol	77.77778
Alcohol and Total Phenols	80.00000
Alcohol and Flavanoids	77.77778

This method gave an accuracy of 77.7777778%, similar to the first model.

Machine Learning Models

Decision Tree

The first machine learning model is a **Decision Tree**.

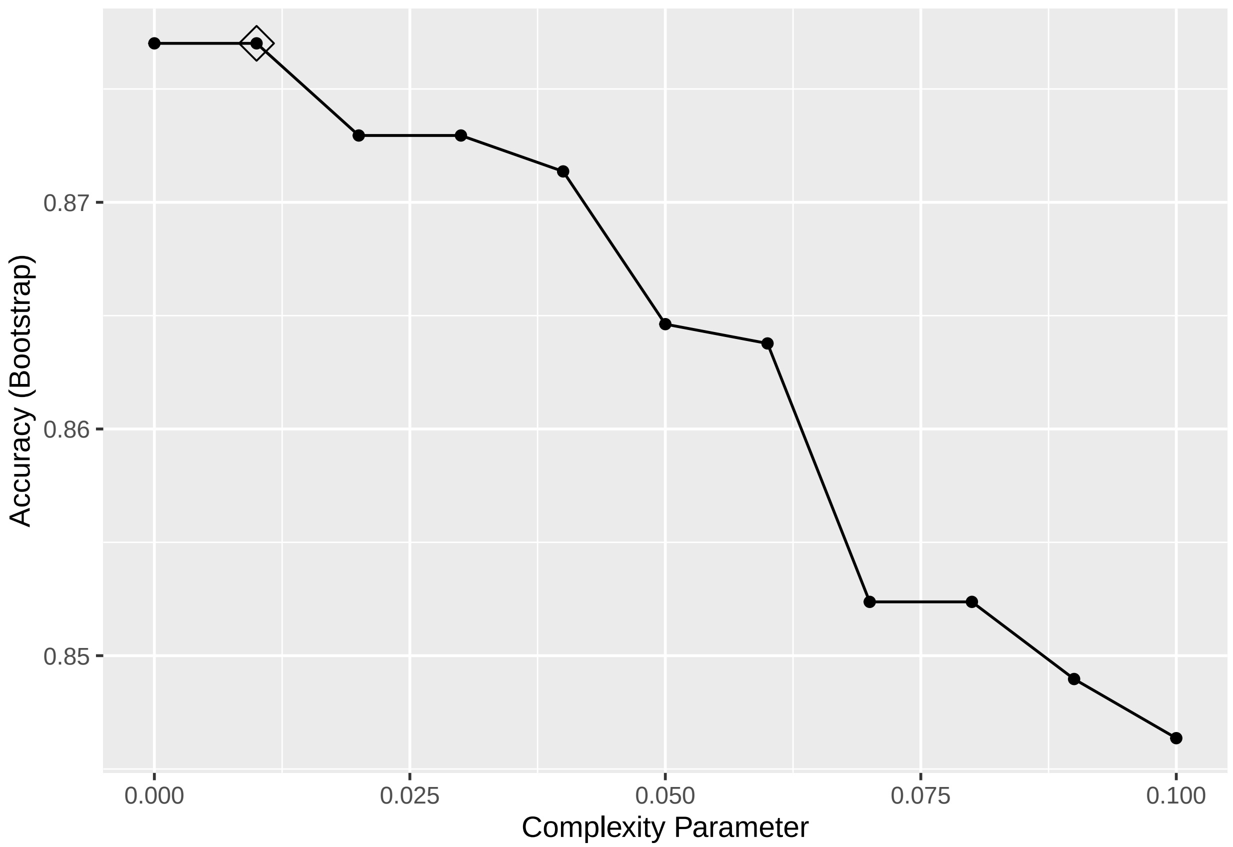
A decision tree is a machine learning algorithm that partitions the data into subsets. The partitioning process starts with a binary split and continues until no further splits can be made. Various branches of variable length are formed.

The goal of a decision tree is to encapsulate the training data in the smallest possible tree. The rationale for minimizing the tree size is the logical rule that the simplest possible explanation for a set of phenomena is preferred over other explanations. Also, small trees produce decisions faster than large trees, and they are much easier to look at and understand. There are various methods and techniques to control the depth, or prune, of the tree.

Main ideas of a decision tree: + Decision trees are popular among non-statisticians as they produce a model that is very easy to interpret. Each leaf node is presented as an if/then rule. Cases that satisfy the if/then statement are placed in the node. + Are non-parametric and therefore do not require normality assumptions of the data. Parametric models specify the form of the relationship between predictors and a response. An example is a linear relationship for regression. In many cases, however, the nature of the relationship is unknown. This is a case in which non-parametric models are useful. + Can handle data of different types, including continuous, categorical, ordinal, and binary. Transformations of the data are not required. + Can be useful for detecting important variables, interactions, and identifying outliers. + Handle missing data by identifying surrogate splits in the modeling process. Surrogate splits are splits highly associated with the primary split. In other models, records with missing values are omitted by default.

The model is constructed using the **train()** function that is part of the *caret* package. The train function uses Cross-Validation to find the optimal Complexity Parameter. The complexity parameter is the minimum improvement in the model needed at each node. The Complexity Parameter value is a stopping parameter. It helps speed up the search for splits because it can identify splits that doesn't meet this criteria and prune them before going too far.

```
set.seed(3, sample.kind = "Rounding") #so that we get consistent results everytime.
train_rpart <- train(Class ~ .,
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0,0.1,0.01)),
                     data = train_set)

ggplot(train_rpart, highlight = TRUE)
```



```
y_hat_rpart <- predict(train_rpart, test_set) #we make a prediction

method4 <- mean(y_hat_rpart == test_set$Class)*100
method4
```

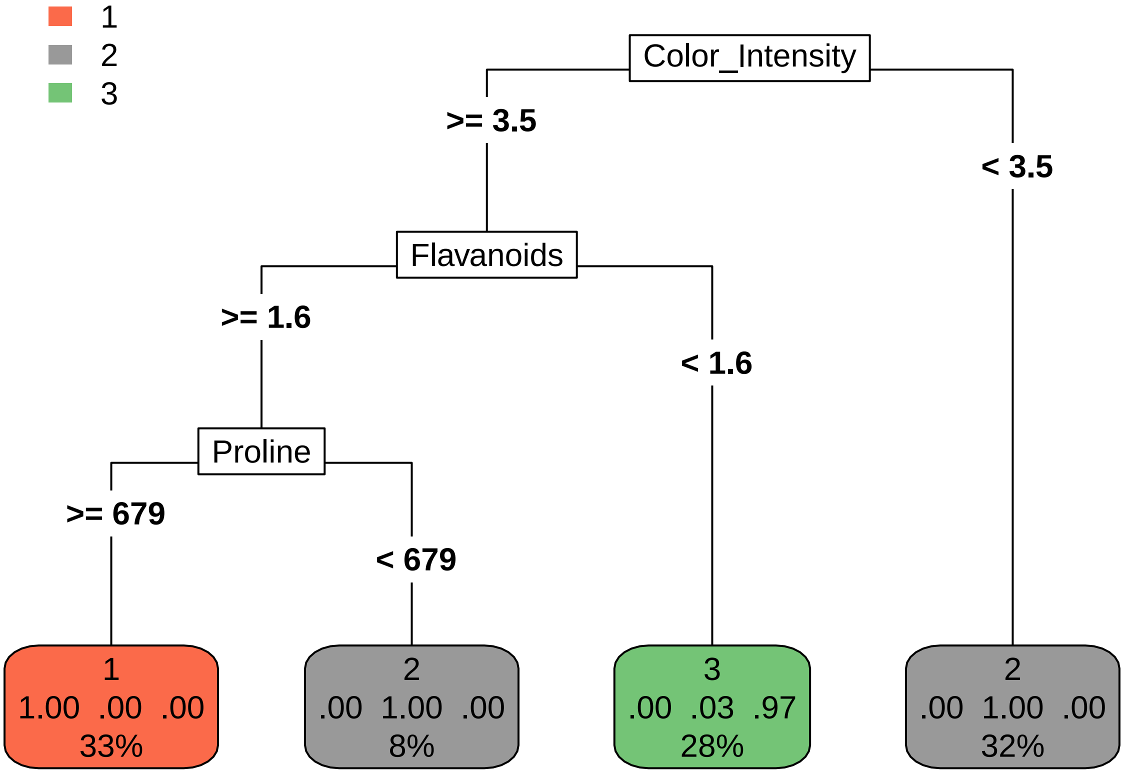
```
## [1] 93.33333
```

By using this method, we obtain an accuracy of 93.3333333%. these results are validated in the following confusion matrix by checking the sensitivity and specificity:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 15  1  0
##          2  0 15  0
##          3  0  2 12
##
## Overall Statistics
##
##                Accuracy : 0.9333
##                  95% CI : (0.8173, 0.986)
##     No Information Rate : 0.4
##     P-Value [Acc > NIR] : 6.213e-14
##
```

```
##                    Kappa : 0.8898
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            1.0000   0.8333   1.0000
## Specificity            0.9667   1.0000   0.9394
## Pos Pred Value         0.9375   1.0000   0.8571
## Neg Pred Value         1.0000   0.9000   1.0000
## Prevalence             0.3333   0.4000   0.2667
## Detection Rate         0.3333   0.3333   0.2667
## Detection Prevalence   0.3556   0.3333   0.3111
## Balanced Accuracy      0.9833   0.9167   0.9697
```

A decision tree will be rendered based on the results from the trained model and the trained data based on the corresponding class. The following plot shows that Flavanoids, Color Intensity, and Proline were the most important variables for this model.



Now we can compare the accuracy of this model with the previous ones in the following table

Methods	Accuracy
Just Alcohol	77.77778
Alcohol and Total Phenols	80.00000
Alcohol and Flavanoids	77.77778

Methods	Accuracy
Decision Tree	93.33333

Random Forest

Random forest (RF) modeling has emerged as an important statistical learning method due to its exceptional predictive performance. Random Forest use a technique called feature bagging, which has the advantage of significantly decreasing the correlation between each DT and thus increasing its predictive accuracy, on average. Feature bagging works by randomly selecting a subset of the feature dimensions at each split in the growth of individual decision trees. This may sound counterintuitive, after all it is often desired to include as many features as possible initially in order to gain as much information for the model. However it has the purpose of deliberately avoiding (on average) very strong predictive features that lead to similar splits in trees (and thus increase correlation).

That is, if a particular feature is strong in predicting the response value then it will be selected for many trees. Hence a standard bagging procedure can be quite correlated. Random forests avoid this by deliberately leaving out these strong features in many of the grown trees. If all values are chosen in splitting of the trees in a random forest ensemble then this simply corresponds to standard bagging.

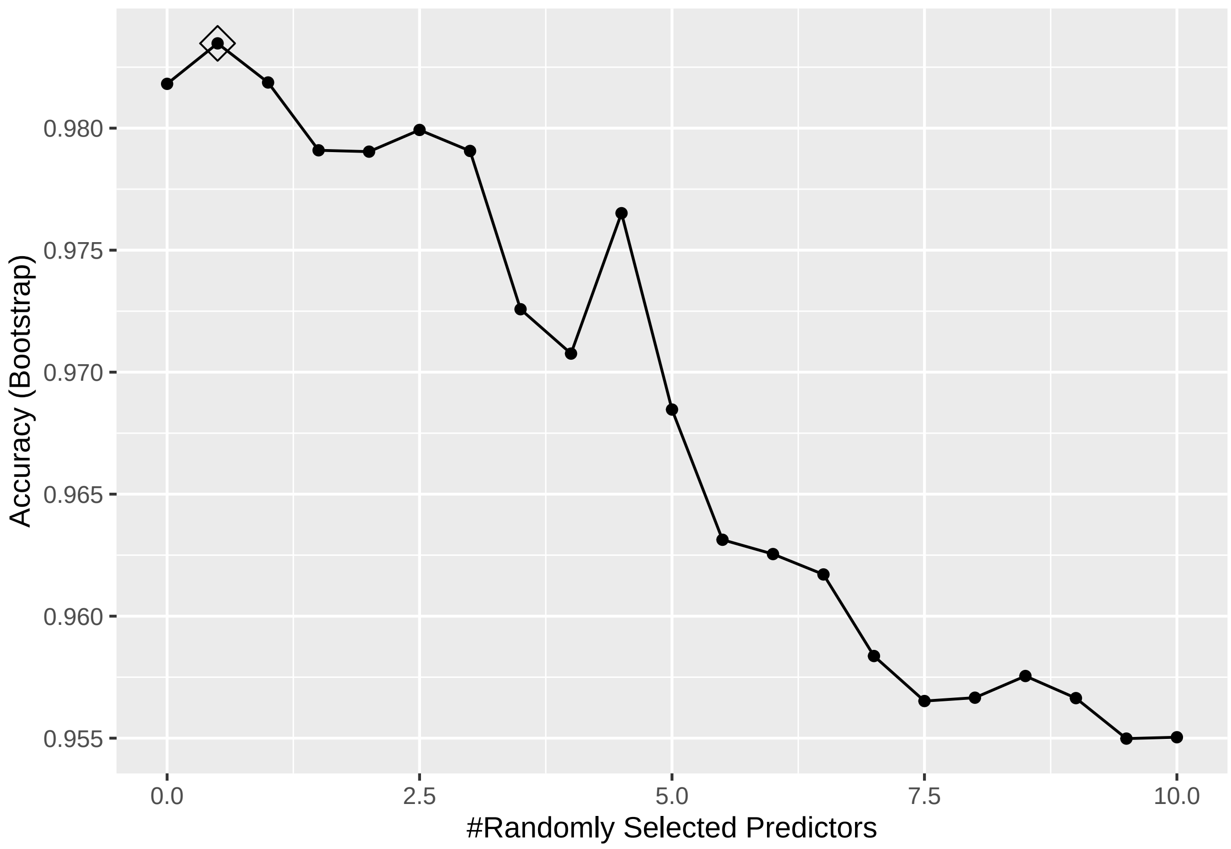
First we try to look for the best parameter that fits for our model. First we use modelLookup to see what parameter is needed and we find out that "mtry" seems to be the best parameter.

```
modelLookup("rf")
```

```
##   model parameter                         label forReg forClass probModel
## 1    rf      mtry #Randomly Selected Predictors   TRUE     TRUE      TRUE
```

The next model is an extension of a Decision Tree. We implement the Random Forest model through the caret's train() function. The train function uses Cross-Validation to find the optimal mtry Parameter. first we will use modelLookup to see what parameter is needed

```
train_rf <- train(Class ~ .,
                  data = train_set,
                  model = "rf",
                  tuneGrid = data.frame(mtry = seq(0,10,0.5)))

ggplot(train_rf, highlight = TRUE)
```



```
y_hat_rf <- predict(train_rf, test_set)
method5 <- mean(y_hat_rf == test_set$Class)*100
method5
```

```
## [1] 100
```

This model has drastically improved the accuracy of the model to 100%! And the information is confirmed by the following confusion matrix.

```
confusionMatrix(y_hat_rf, test_set$Class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 15  0  0
##          2  0 18  0
##          3  0  0 12
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9213, 1)
##     No Information Rate : 0.4
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                  Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            1.0000      1.0   1.0000
## Specificity            1.0000      1.0   1.0000
## Pos Pred Value         1.0000      1.0   1.0000
## Neg Pred Value         1.0000      1.0   1.0000
## Prevalence             0.3333      0.4   0.2667
## Detection Rate         0.3333      0.4   0.2667
## Detection Prevalence   0.3333      0.4   0.2667
## Balanced Accuracy      1.0000      1.0   1.0000
```

Methods	Accuracy
Just Alcohol	77.77778
Alcohol and Total Phenols	80.00000
Alcohol and Flavanoids	77.77778
Decision Tree	93.33333
Random Forest	100.00000

From this model, our accuracy jumps to 100%! We can observe the variable importance as follows.

	MeanDecreaseGini
Alcohol	9.149007
Malic_Acid	4.722658
Ash	4.018700
Alcalinity_of_Ash	6.064171
Magnesium	5.762170
Total_Phenols	6.588014
Flavanoids	9.021562
Nonflavanoids_Phenols	4.137313
Proanthocyanins	4.824190
Color_Intensity	9.345227
Hue	6.649037
Proteine_Concentration	7.660042
Proline	8.858598

Here we can see the Flavanoids, Color Intensity, and Alcohol had the highest variable importance.

Results

Below is a table summarizing the accuracies of each method by using the train and test sets.

Methods	Accuracy
Alcohol and Flavanoids	77.77778
Just Alcohol	77.77778
Alcohol and Total Phenols	80.00000
Decision Tree	93.33333
Random Forest	100.00000

We would like to test the Random Forest model with the entire dataset as follows.

```r
final_preds <- predict(train_rf, wine_data)
final_method <- mean(final_preds == wine_data$Class)*100
cat("The accuracy of Random Forest on the entire dataset is: ",final_method,"%")
```

```
## The accuracy of Random Forest on the entire dataset is:  100 %
```

Conclusion

By applying a Random Forest algorithm, it is possible to provide very robust and reliable results. The results of this project can be extended to classify unknown wine's by their chemical compounds. In order to do that, this data set must be much more detailed. All of the wines that were explored in this project were from the same region in Italy; it is possible that wines from a different country could have similar chemical compounds as these wines which would require a more sophisticated predictive model.

References

Aeberhard, Stefan. et al (2020). UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/wine]. Irvine, CA: University of California, School of Information and Computer Science. 17

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2020). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-4. https://CRAN.R-project.org/package=e1071

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Max Kuhn (2020). caret: Classification and Regression Training. R package version 6.0-86. https://CRAN.R-project.org/package=caret

Matt Dowle and Arun Srinivasan (2020). data.table: Extension of `data.frame`. R package version 1.13.2. https://CRAN.R-project.org/package=data.table

Stephen Milborrow (2020). rpart.plot: Plot 'rpart' Models: An Enhanced Version of 'plot.rpart'. R package version 3.0.9. https://CRAN.R-project.org/package=rpart.plot

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686