Wine Price Analysis/Predictor

Cong Feng

12/15/2020

Introduction

In this project we are going to explore the relation between a wine's price and the features that affects it. As a wine enthusiast, one of the main factors that I look for when trying new wine is their rating, type of wine and where the wine came from. Today I will be working with a data set that includes all of those factors.

The data is downloaded from https://www.kaggle.com/zynicide/wine-reviews , the user scrapped the data from the wineEnthusiast magazine during the week of June 15th 2017.

The goal I have set for this project is to analyze to see how each of the factors correlates with the price of wine. Then use these features to predict the price of each wine. To determine the accuracy of the prediction, we will calculate the "Root Mean Square Error" between the predicted price and actual price.

The formula is defined:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}(\hat{y}_i - y_i)^2}$$

```r
RMSE <- function(predicted_price, actual_price){
  sqrt(mean((predicted_price - actual_price)^2))
}
```

Data Loading and Wrangling

```
## 'data.frame':    129971 obs. of  14 variables:
##  $ X                    : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ country              : chr  "Italy" "Portugal" "US" "US" ...
##  $ description          : chr  "Aromas include tropical fruit, broom, brimstone and dried herb. The p
##  $ designation          : chr  "VulkÃ  Bianco" "Avidagos" "" "Reserve Late Harvest" ...
##  $ points               : int  87 87 87 87 87 87 87 87 87 87 ...
##  $ price                : num  NA 15 14 13 65 15 16 24 12 27 ...
##  $ province             : chr  "Sicily & Sardinia" "Douro" "Oregon" "Michigan" ...
##  $ region_1             : chr  "Etna" "" "Willamette Valley" "Lake Michigan Shore" ...
##  $ region_2             : chr  "" "" "Willamette Valley" "" ...
##  $ taster_name          : chr  "Kerin Oâ\200\231Keefe" "Roger Voss" "Paul Gregutt" "Alexander Peartr
##  $ taster_twitter_handle: chr  "@kerinokeefe" "@vossroger" "@paulgwineÂ " "" ...
##  $ title                : chr  "Nicosia 2013 VulkÃ  Bianco  (Etna)" "Quinta dos Avidagos 2011 Avidag
##  $ variety              : chr  "White Blend" "Portuguese Red" "Pinot Gris" "Riesling" ...
##  $ winery               : chr  "Nicosia" "Quinta dos Avidagos" "Rainstorm" "St. Julian" ...
```

Here we see the first few rows of the data, there are a lot of information available to us. I will tidy the data and only keep the columns I plan to use.

```r
new_dat <- dat %>%
  select(X, country, points, price, title, variety, winery)

head(new_dat)
```

```
##   X  country points price
## 1 0    Italy     87    NA
## 2 1 Portugal     87    15
## 3 2       US     87    14
## 4 3       US     87    13
## 5 4       US     87    65
## 6 5    Spain     87    15
##                                                                                  title
## 1                                                 Nicosia 2013 VulkÃ  Bianco  (Etna)
## 2                                      Quinta dos Avidagos 2011 Avidagos Red (Douro)
## 3                                      Rainstorm 2013 Pinot Gris (Willamette Valley)
## 4                  St. Julian 2013 Reserve Late Harvest Riesling (Lake Michigan Shore)
## 5 Sweet Cheeks 2012 Vintner’s Reserve Wild Child Block Pinot Noir (Willamette Valley)
## 6                              Tandem 2011 Ars In Vitro Tempranillo-Merlot (Navarra)
##              variety              winery
## 1        White Blend             Nicosia
## 2     Portuguese Red Quinta dos Avidagos
## 3         Pinot Gris           Rainstorm
## 4           Riesling          St. Julian
## 5         Pinot Noir        Sweet Cheeks
## 6 Tempranillo-Merlot              Tandem
```

The title column includes the year that the wine is produce. We can extract this important information and store in it’s own column.

```r
pattern <- "\\d{4}"
year <- str_extract(new_dat$title, pattern)

new_dat <- new_dat %>%
  mutate(years = year)

# Filter out NAs from dataset
new_dat <- new_dat %>%
  filter(!is.na(price), !is.na(points), !is.na(country),
         !is.na(variety), !is.na(years), !is.na(winery))


head(new_dat)
```

```
##   X  country points price
## 1 1 Portugal     87    15
## 2 2       US     87    14
## 3 3       US     87    13
## 4 4       US     87    65
## 5 5    Spain     87    15
## 6 6    Italy     87    16
##                                                                                  title
## 1                                       Quinta dos Avidagos 2011 Avidagos Red (Douro)
## 2                                       Rainstorm 2013 Pinot Gris (Willamette Valley)
## 3                 St. Julian 2013 Reserve Late Harvest Riesling (Lake Michigan Shore)
## 4 Sweet Cheeks 2012 Vintner's Reserve Wild Child Block Pinot Noir (Willamette Valley)
## 5                               Tandem 2011 Ars In Vitro Tempranillo-Merlot (Navarra)
## 6                                    Terre di Giurfo 2013 Belsito Frappato (Vittoria)
##              variety              winery years
## 1     Portuguese Red Quinta dos Avidagos  2011
## 2         Pinot Gris           Rainstorm  2013
## 3           Riesling          St. Julian  2013
## 4         Pinot Noir        Sweet Cheeks  2012
## 5 Tempranillo-Merlot              Tandem  2011
## 6           Frappato     Terre di Giurfo  2013
```

The most important rule in machine learning is to not train on data that will be tested on. This is to prevent over training and over smoothing. For the last part of data wrangling is to divide the data into a modeling/training set and a validation set. It will be a 90/10% split.

```r
set.seed(123, sample.kind = "Rounding")

ind <- createDataPartition(new_dat$price, times=1, p=0.1, list=FALSE)
wine_dat <- new_dat[-ind,]
temp_file <- new_dat[ind,]

# Make sure all the features are in both data.
validate_dat <- temp_file %>%
  semi_join(wine_dat, by = "country") %>%
  semi_join(wine_dat, by = "price") %>%
  semi_join(wine_dat, by = "variety") %>%
  semi_join(wine_dat, by = "years") %>%
  semi_join(wine_dat, by = "winery")

rm(temp_file)
```

Data Analysis

To begin the data analysis, first we will take a look at the dimension of the dataset.

```
dim(wine_dat)
```

```
## [1] 105152      8
```

```
##   Country Points Price Title Variety Winery Year
## 1      43     21   379 97485     671  15077   88
```

Here is the breakdown of unique entries in each column. We see there are only 21 unique entries in the points column, which is strange because the point scale is out of 100.

```
c(min(wine_dat$points), max(wine_dat$points))
```

```
## [1]  80 100
```

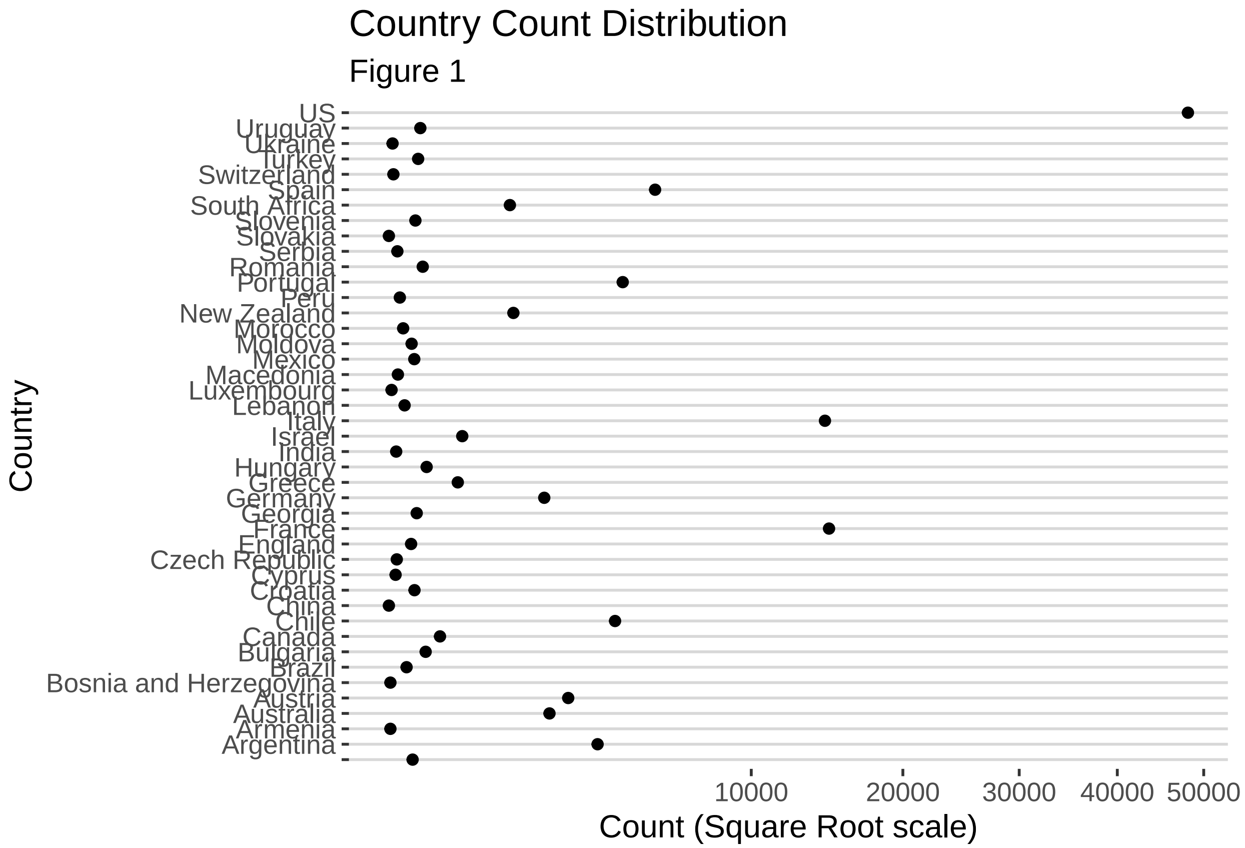
We see the point distribution is actually from 80 to 100, this is an important piece of information for us to proceed with. A consumer might assumes 80 is a relatively high score, however it's actually the lowest score they give.

```
c(min(wine_dat$price), max(wine_dat$price))
```

```
## [1]    4 2500
```

The price covers a wide range of wines as well. In this data, it covers everything from 4USD to 2500USD.

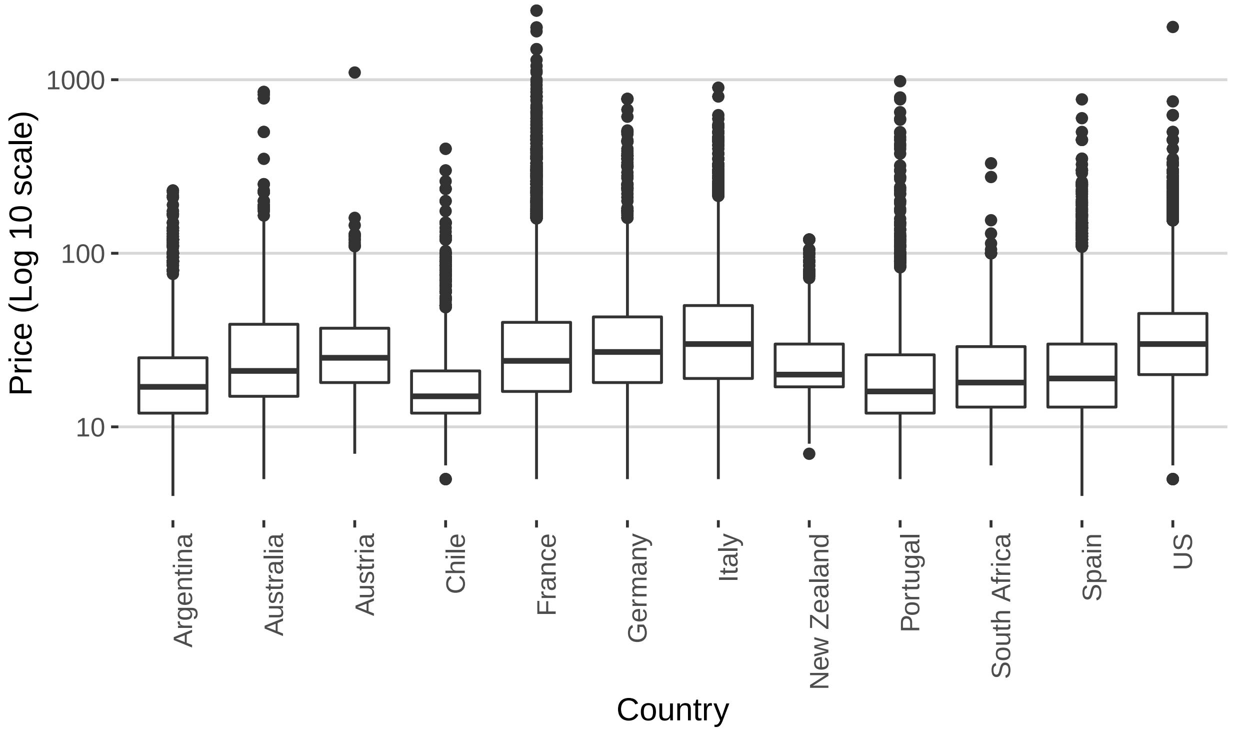
Breakdown by Country



Here is the breakdown of the number of entries each country is represented in the data. We see that majority of the entries are from 4 countries: US, France, Spain and Italy. This is not surprising as these 4 countries are the largest producers of wine.

Price by Country

Figure 2

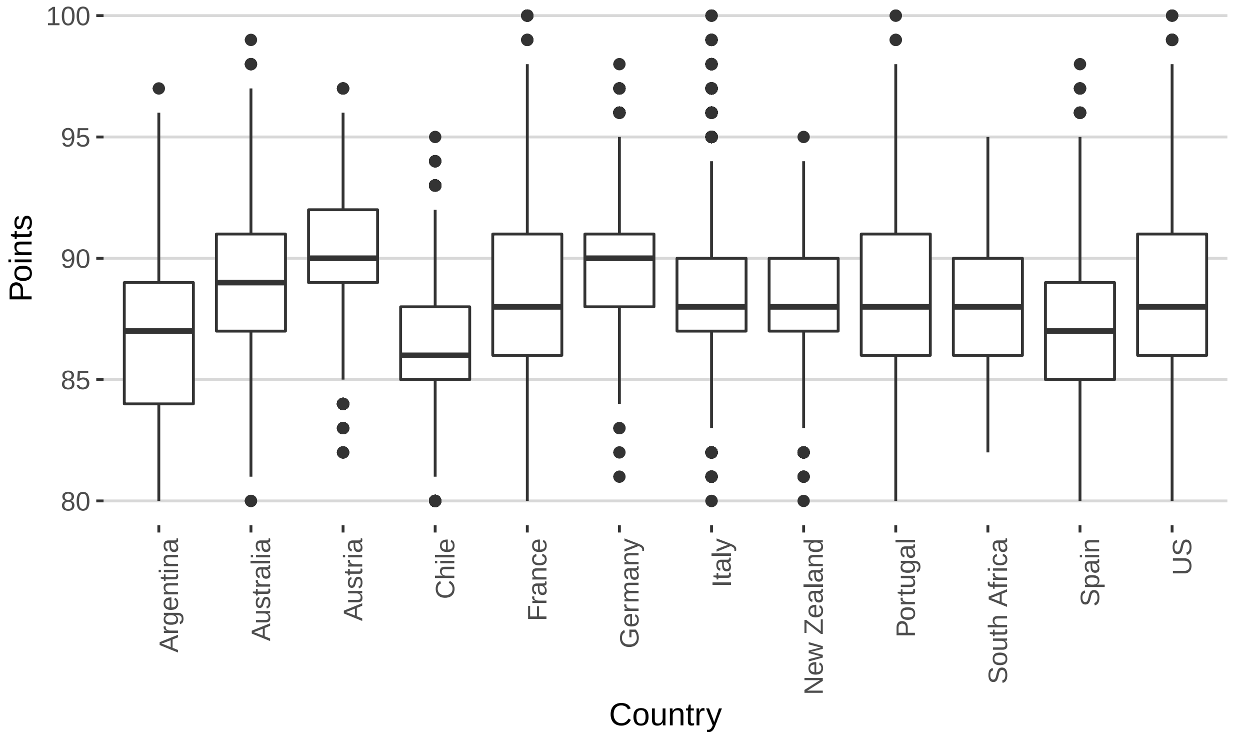


This graph breaks down the price distribution for countries with over 500 entries. Each rectangle represents the 25 to 75% percentile for the price, the line inside the rectangle represents the median price. This means the larger the rectange, the higher variance of the price in that relative country.

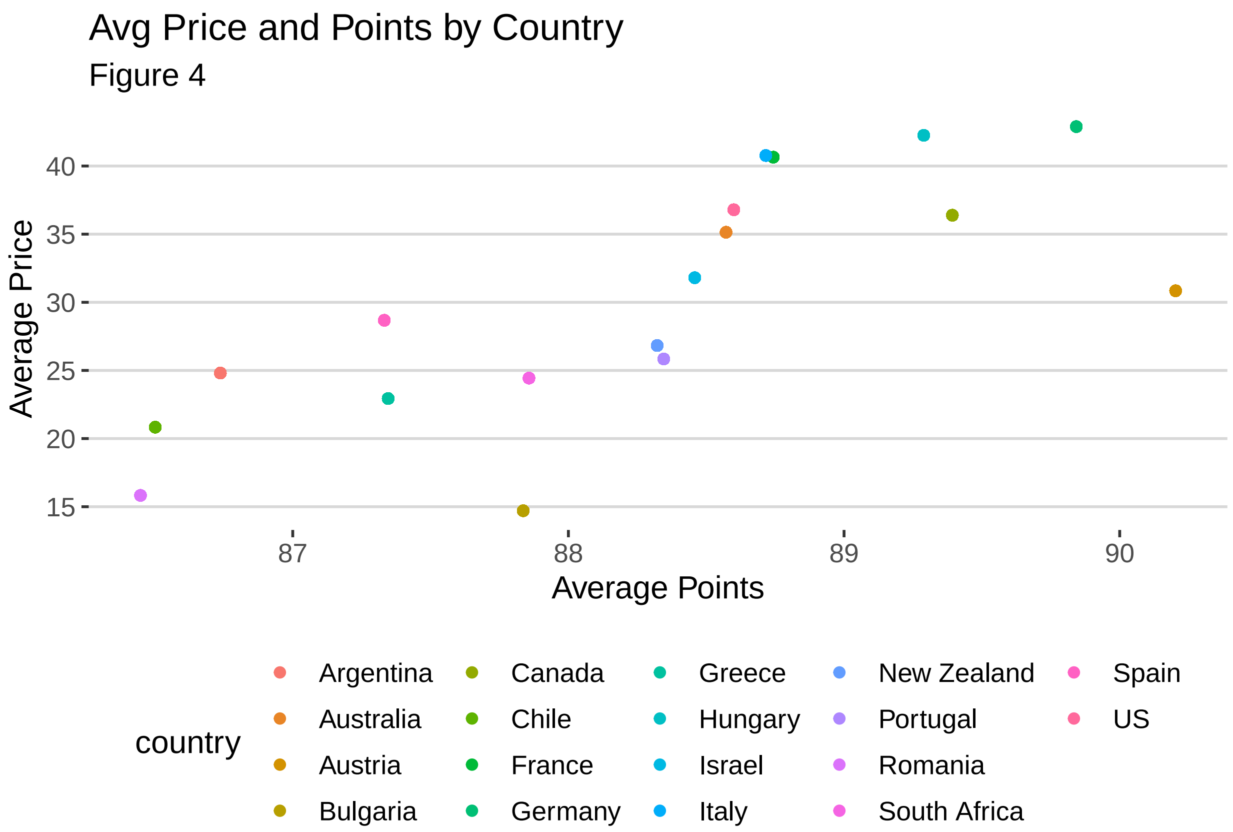
We see from this graph that the US has the highest average price with average size variance. Chile has the lowest variance and the lowest average price. France has the most wines in highest price percentile, however their average wine price is lower than the US.

Points by Country

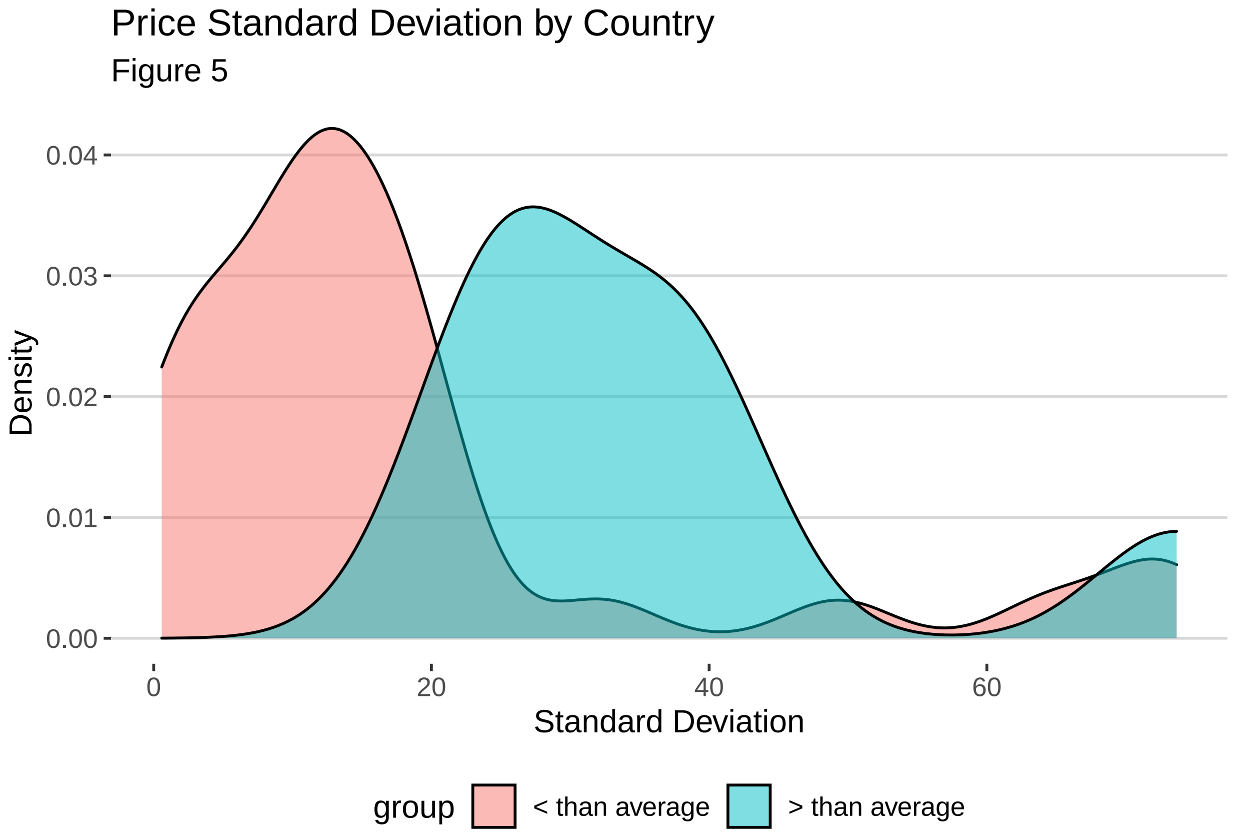
Figure 3



We will do the same breakdown for the points distribution. Germany, Austria and Australia has the highest median rating; Chile with the lowest median price also has the lowest median rating. However, despite having the highest median rating - none of those 3 countries have any wines that has the perfect score. Only US, France, Italy and Portugal have wines that received the perfect rating.



When the average price and points are compared, the points are very scattered. There is not a strong correlation between the average points and price when grouped by country. The country with the highest average points is ranked 9th when it comes to the average price.



When we group the groups into greater and less than average number of ratings to compare the standard deviation; we see that the countries with less ratings has lower variance. This does makes sense; the countries that were rated more has a wider quality range and their producers sell wines at more price points to cater to all clientele.

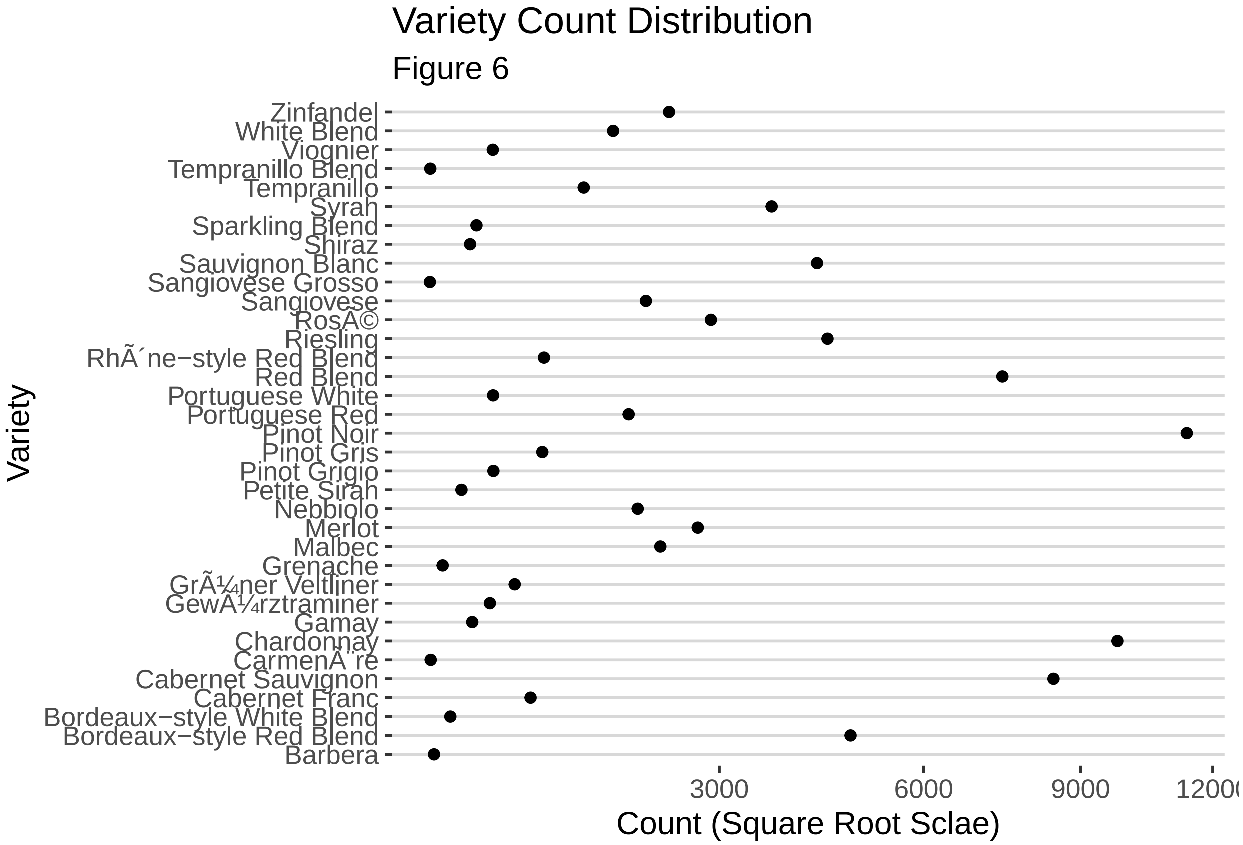
```
country_avg <- wine_dat %>%
  group_by(country) %>%
  summarize(avg_pts = mean(points), avg_p = mean(price))

correlate <- data.frame(Category = "Category",
                        Correlation = "Correlation")

correlate <- bind_rows(Category = "Country",
                       Correlation = cor(country_avg$avg_pts, country_avg$avg_p))
correlate
```

```
## # A tibble: 1 x 2
##   Category Correlation
##   <chr>          <dbl>
## 1 Country        0.468
```

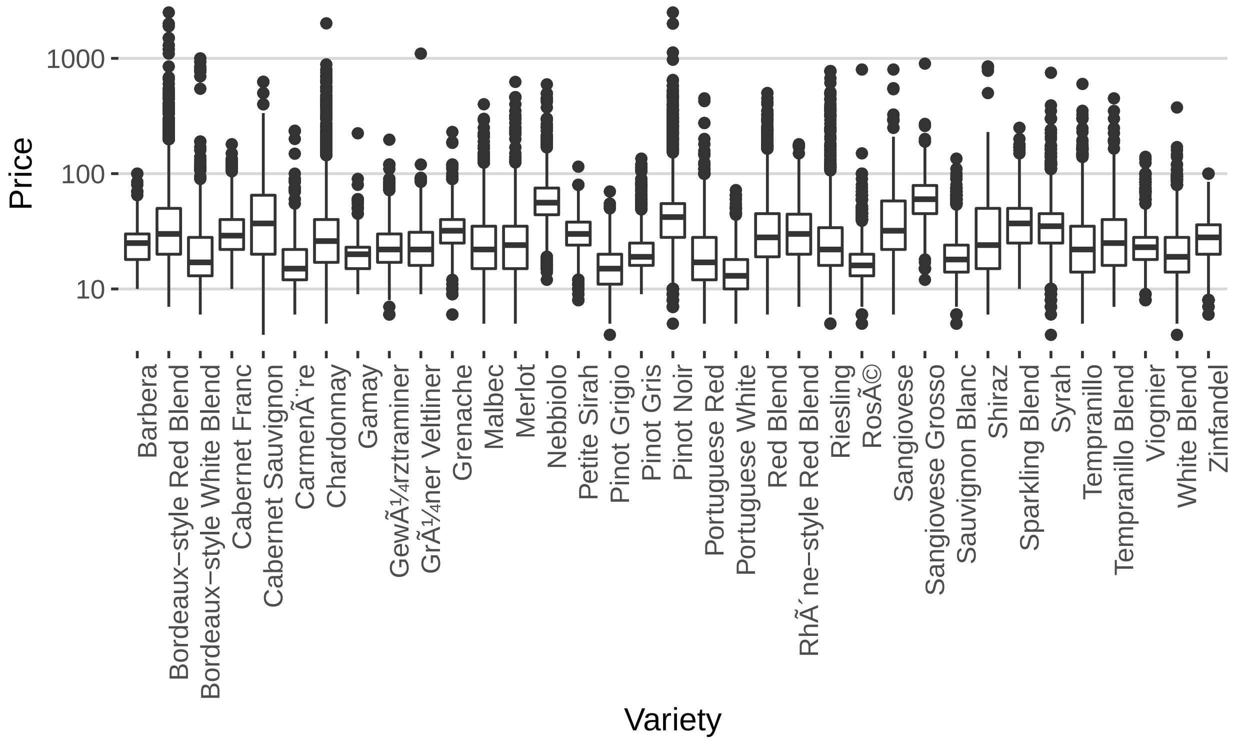
We can use the correlate function to calculate our findings, we will make it into a data tables so we can observe which category has the highest correlation with the wine's price. This function returns a range from -1 to 1, where -1 is negative correlation and 1 is positive.

The country category returned a 0.465, this means that it shows some positive correlation but not overwhelming.

Breakdown by Variety



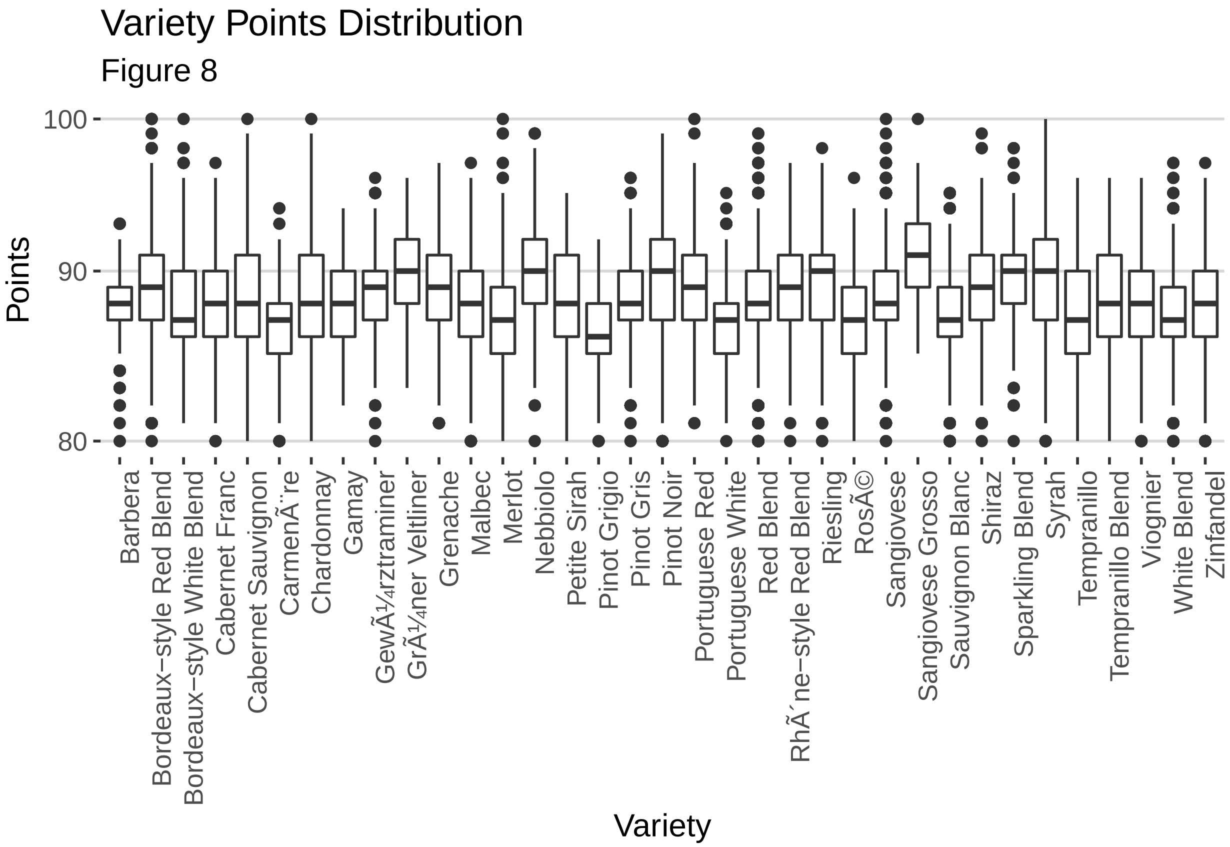
In this graph, I have plotted the variety of wine that has more than 500 entries. From the distribution by variety, the top 5 most rated wine types are: Pinot Noir, Chardonnay, Carbernet Sauvignon, Red Blend and Bordeaux-style Red Blend. Four out of the five most rated wines are red wines, only Chardonnay is a white wine.



Variety Price Distribution

Figure 7

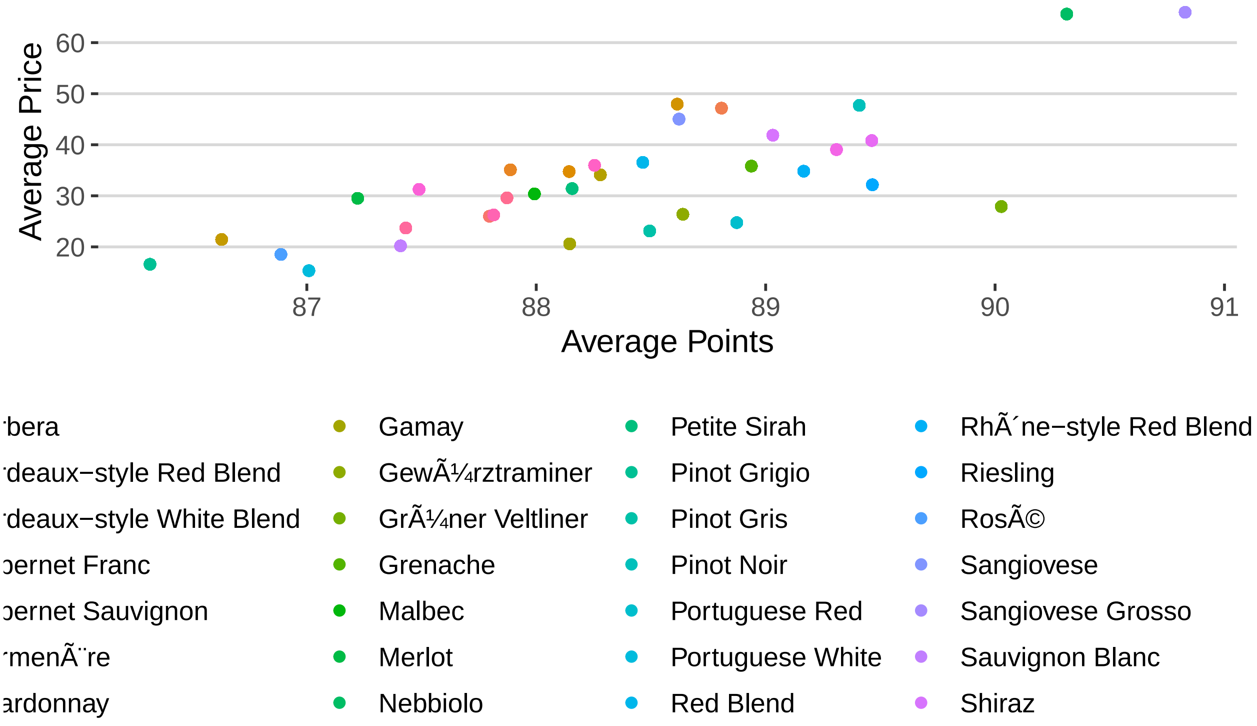
There aren't any distinct pattern emerging from this graph. The different types of wine have wide range of variance in price. Three out of the top five most rated wines are represented in the highest price range of over 1000USD. The Bordeaux-style Red Blend is the variety with the most wines there is that price range, however the median price for this wine is in line with the other varieties.



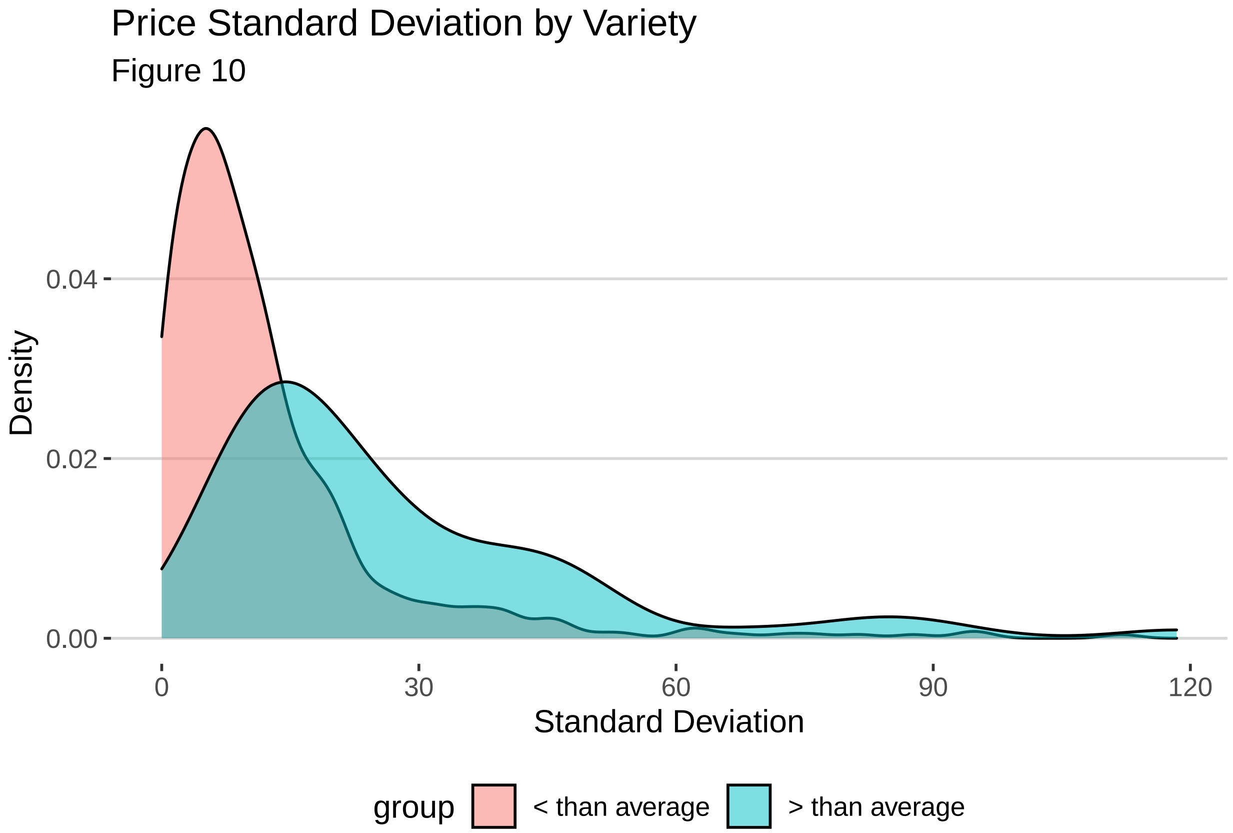
The different variety's median points is all around the 88-90 points range. Pinot Grigio is clearly the variety with the lowest rating. On the other end of the spectrum, Sangivoese Grosso is the clearly leader in the points it receives.

Avg Points and Price by Variety

Figure 9



There is linear distribution at the lower end of the average point scale (< 88). As it moves past that point the distribution becomes much more scattered. When broken down the variety, there is not a distinct pattern.

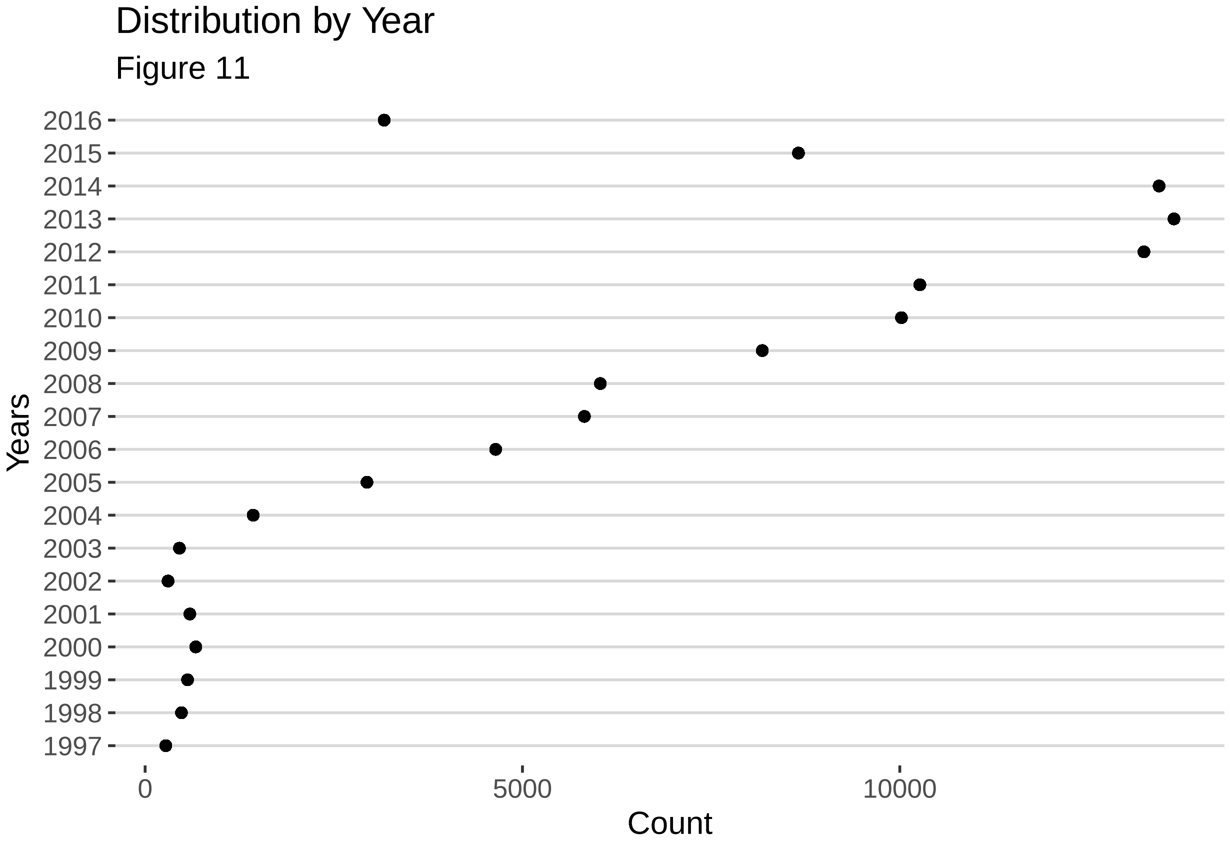


We see lower standard deviation for the group that were rated less than the average amount. The variety of wine that were rated a lot are most likely to be the more popular ones. Therefore the producers is looking to produce both wines that are targets everyday consumers and connoisseurs.

```r
variety_avg <- wine_dat %>%
  group_by(variety) %>%
  summarize(avg_pts = mean(points), avg_p = mean(price))

correlate <- bind_rows(correlate,
                     data.frame(Category = "Variety",
                     Correlation = cor(variety_avg$avg_pts, variety_avg$avg_p)))
```

This is consistent with the finding of our data analysis. The different types of wine does not play a big role in affecting the price and points. Prior to doing this project I had the misconception of the different type of wine has a drastic effect on the price and point.

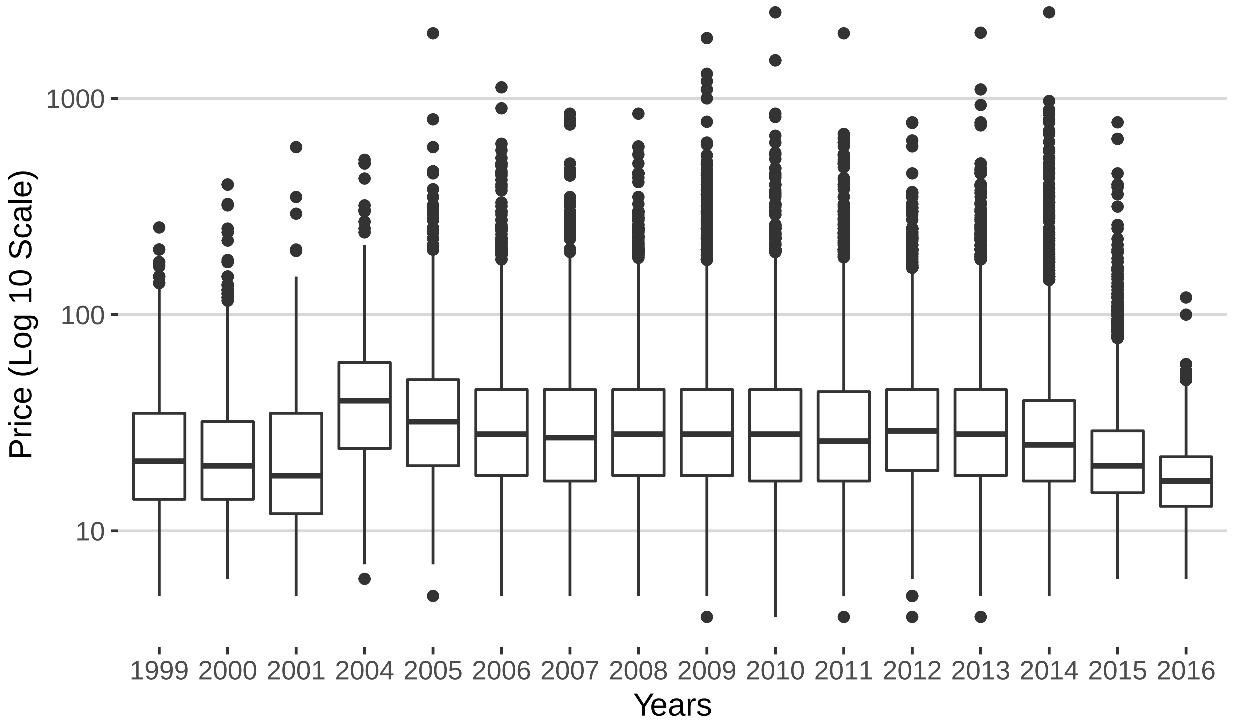
Breakdown by Year



In this graph, we see that majority of the reviews of the wines in the list are for ones that are produced recently. The top 5 years with over 10,000 review each are between the year of 2010 to 2014. We know the climate is a big factor in grape/wine production, let's dig deeper into the data to see if there is any evidence of this affecting the price and points of the wines.

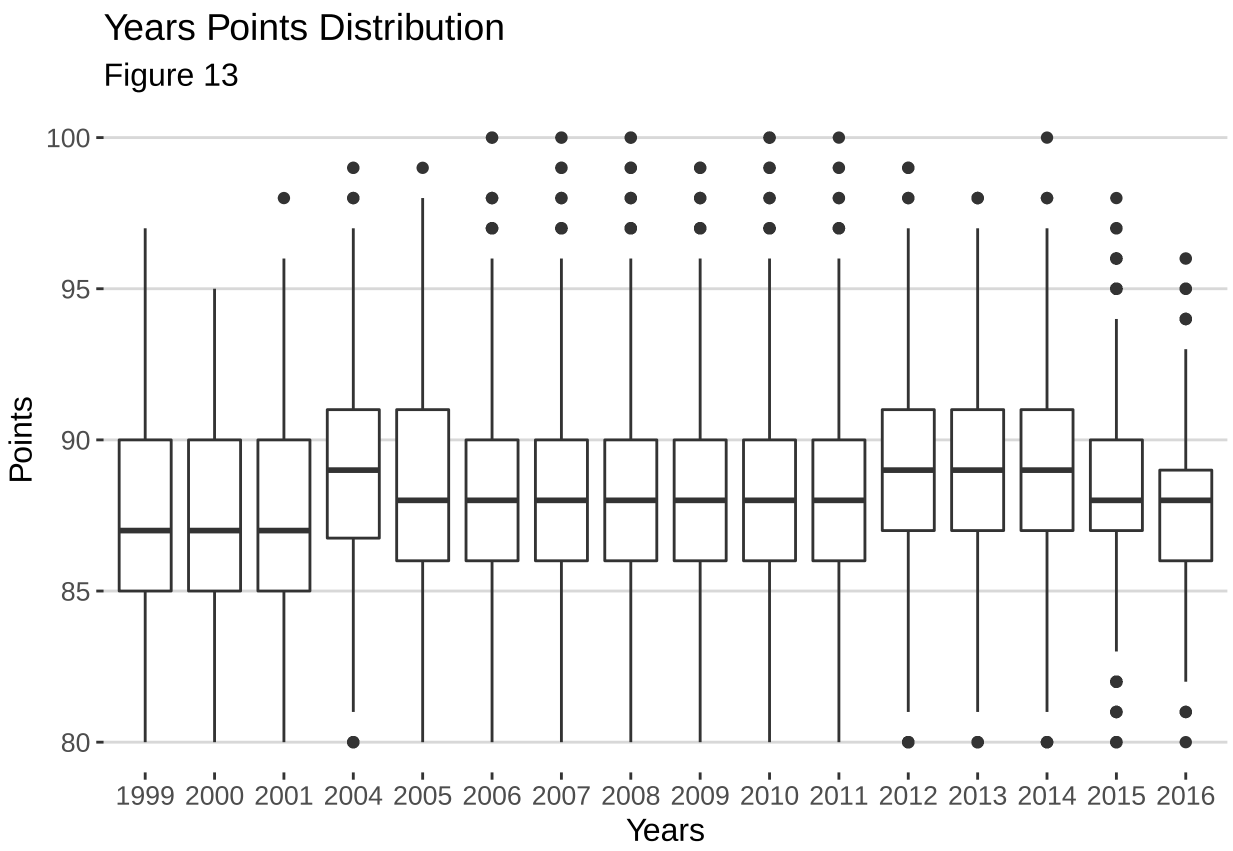
Years Price Distribution

Figure 12



In this graph we can observe that from the year 2006 to 2013, the distribution of the price is nearly identical. The rectangles are very similar to each other in height and the top percentile, however there is more variation at the bottom percentile.

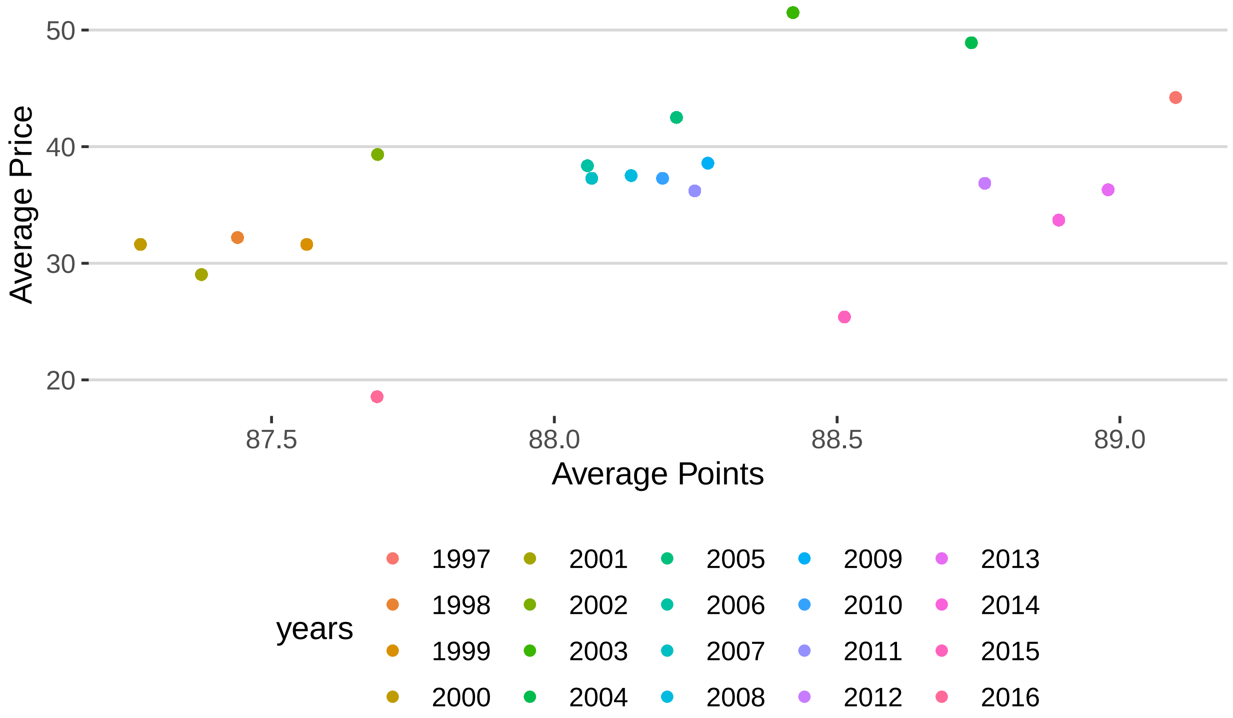
We can also observe that the year 2004 has the highest average price, however it did not have any wines in the in the extreme upper percentile of price. Alternatively, the most recent years in this data set (2016) has by far the lowest variance and price. Only 6 years has wines that are priced at over 1000USD.



In the points distribution breakdown, we see that there are several groups of nearly identical distributions. Year 1999 to 2001, 2006 to 2011 and 2012 to 2014 are the 3 groups that falls into this pattern. When we refer back to Figure 9, we see that years 2012-2014 are the 3 years with the most ratings. We also can observe that there are only 6 years that has wines with perfect rating.

Avg Points and Price by Year

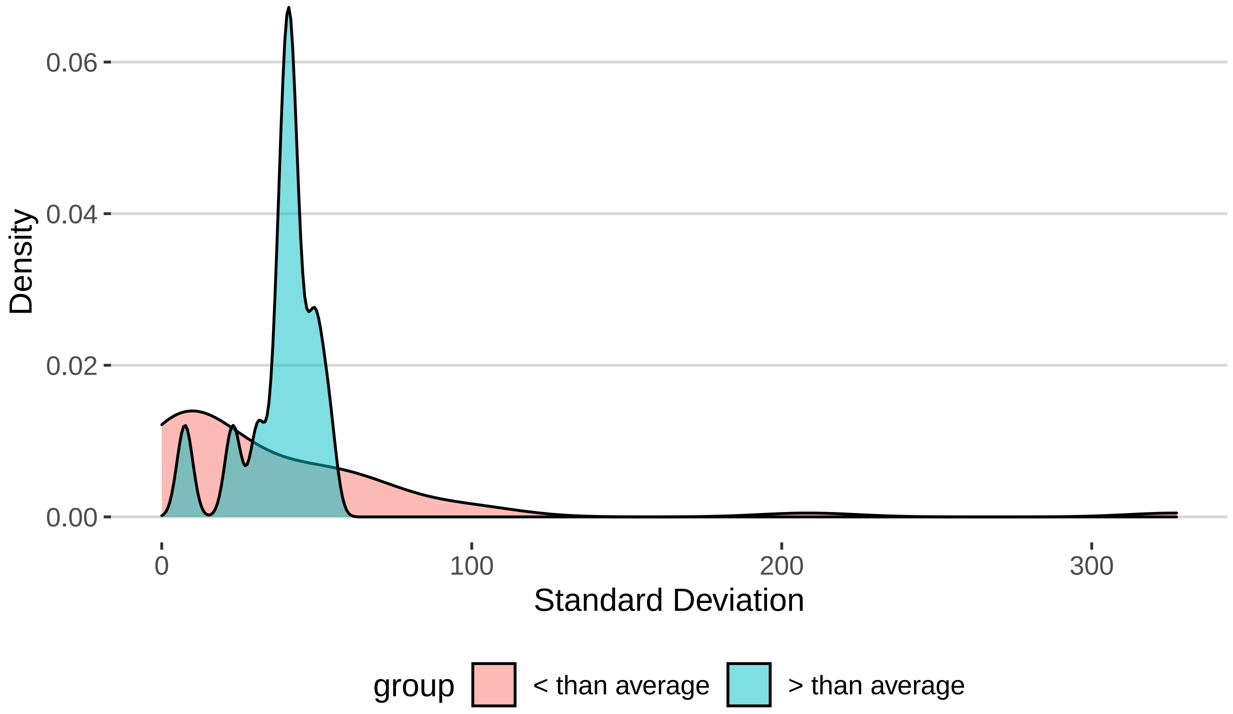
Figure 14



This graph confirms our findings from the previous 2 figures. There are clusters of points around 3 price and rating areas. From these observations we can deduce that the climate is relatively similar in a period of few years. This effects the growth condition of the grapes and the quality of the wine.

Price Standard Deviation by Years

Figure 15



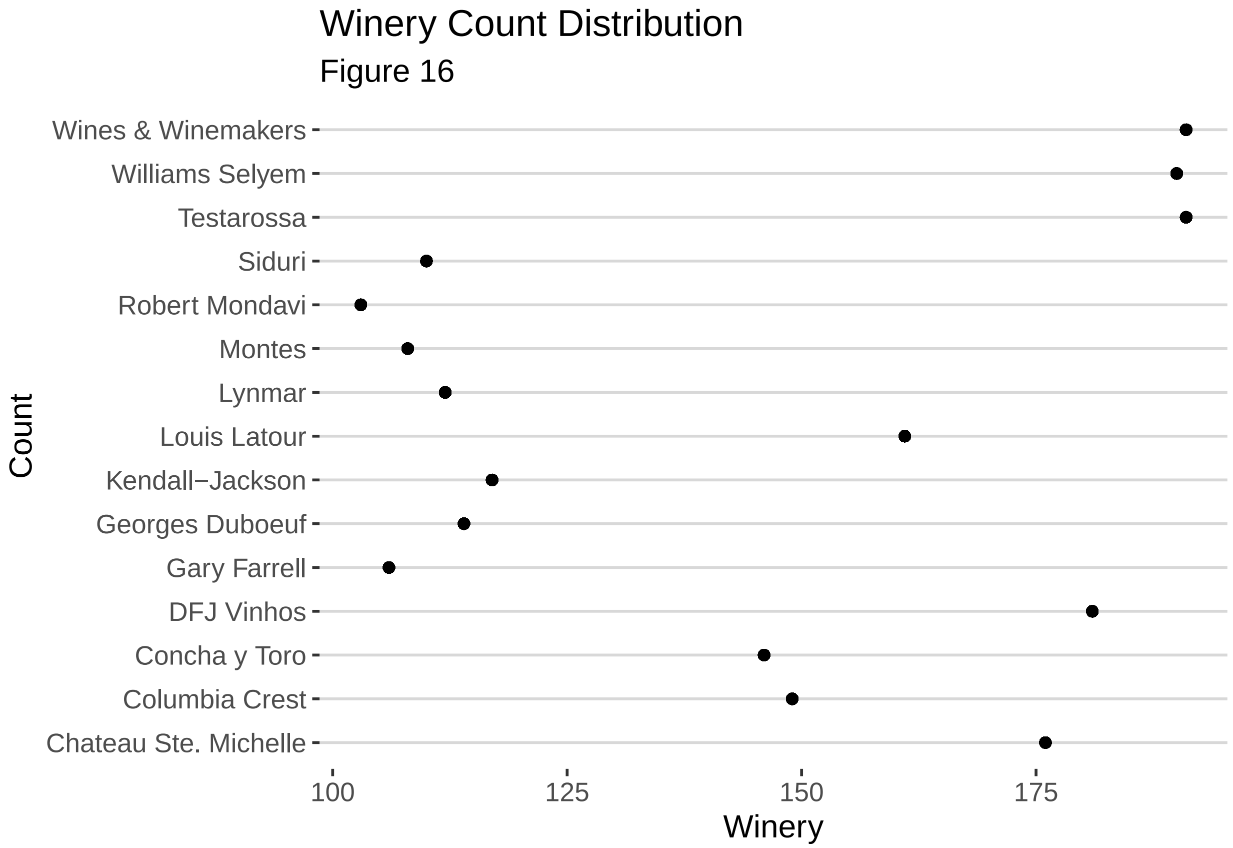
The standard deviation for the years category is complete opposite of the breakdown for the country and variety groups. For the years that were rated more than the average, it is very concentrated. This is in line with our previous observations. Majority of the wines in this data set are from 2010 to 2016, we saw in the box plots that the distribution is quite similar. As the wine gets older it tends to be more rare, and those tend to be more expensive.

```
years_avg <- wine_dat %>%
  group_by(years) %>%
  filter(!is.na(years)) %>%
  summarize(avg_pts = mean(points), avg_p = mean(price))

correlate <- bind_rows(correlate,
                       data.frame(
                         Category = "Years",
                         Correlation = cor(years_avg$avg_pts, years_avg$avg_p)
                       ))

correlate
```

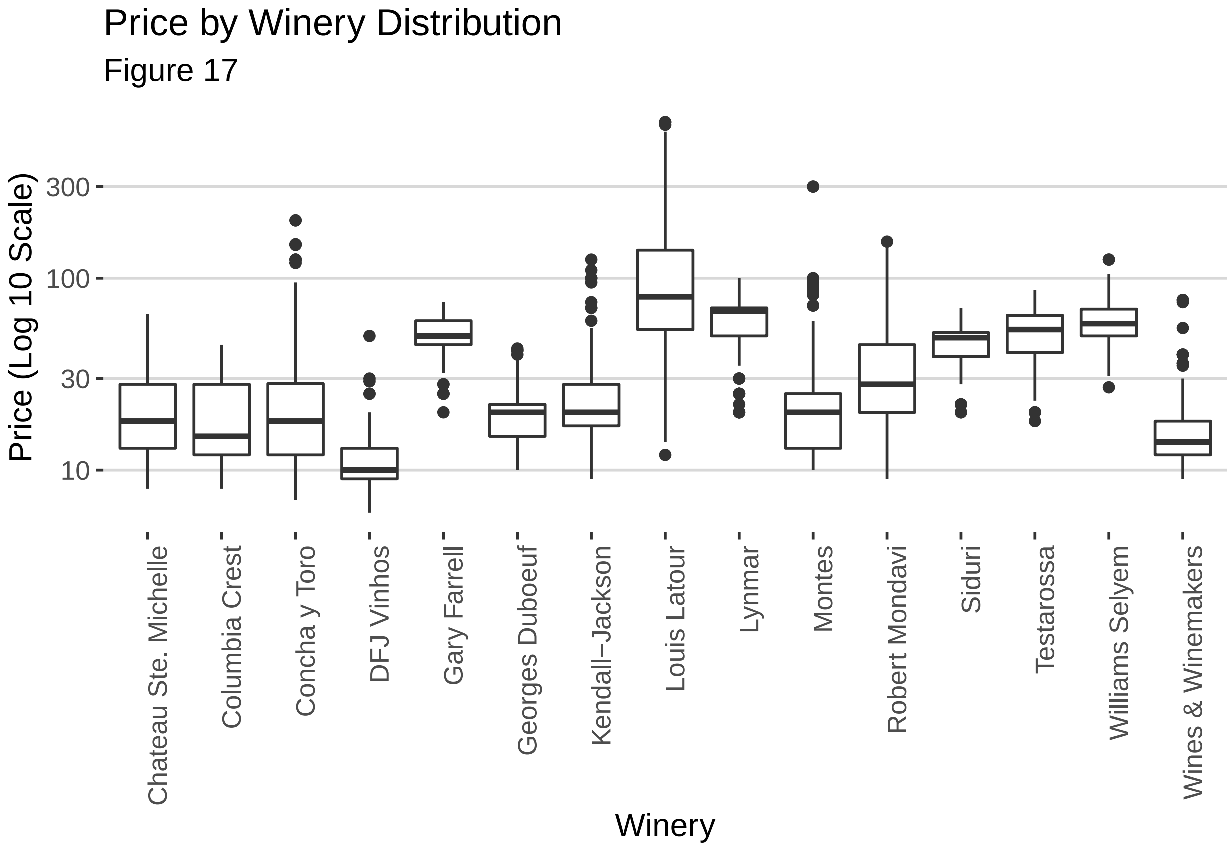
```
## # A tibble: 3 x 2
##   Category Correlation
##   <chr>          <dbl>
## 1 Country        0.468
## 2 Variety        0.425
## 3 Years          0.619
```

When grouped by the years category, the average price and points showed the highest correlation so far. This further reinforce our finding from the visual analysis.

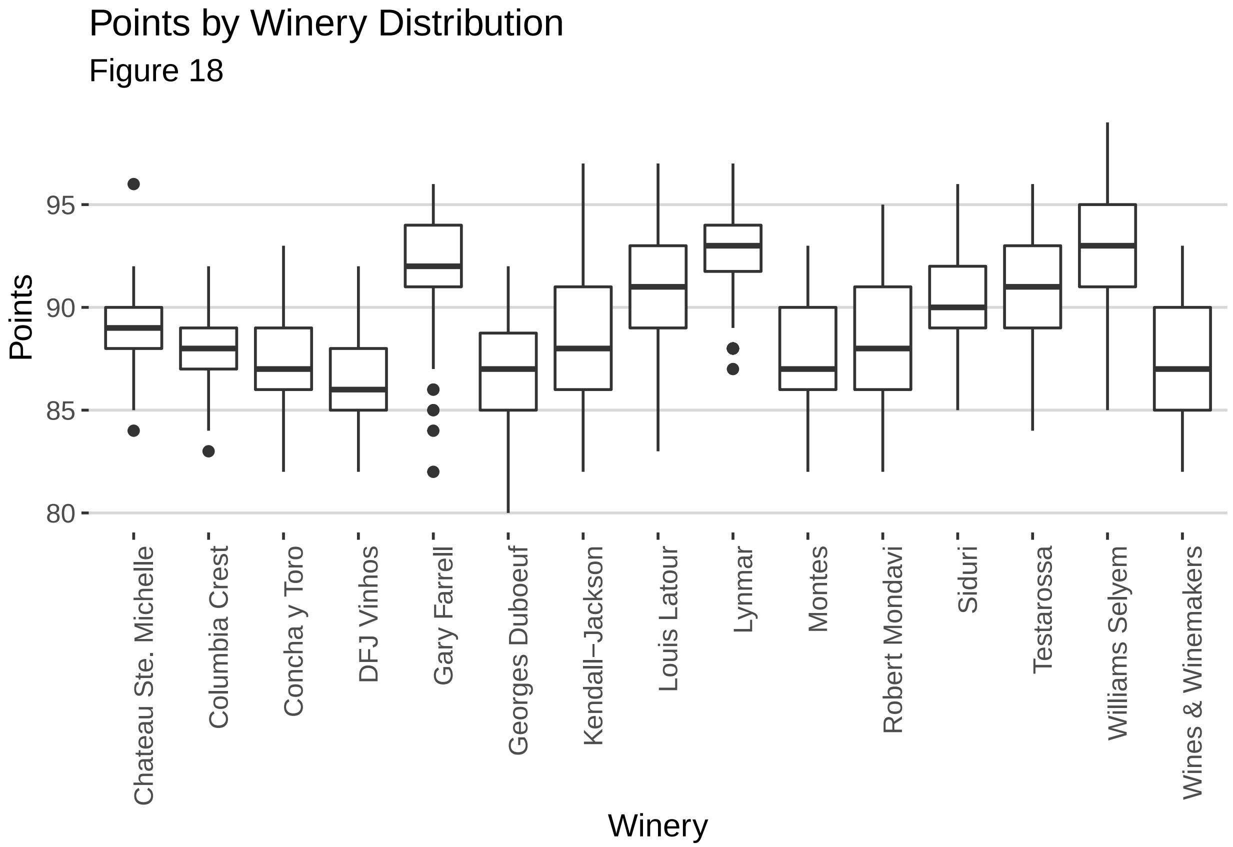
Breakdown by Winery



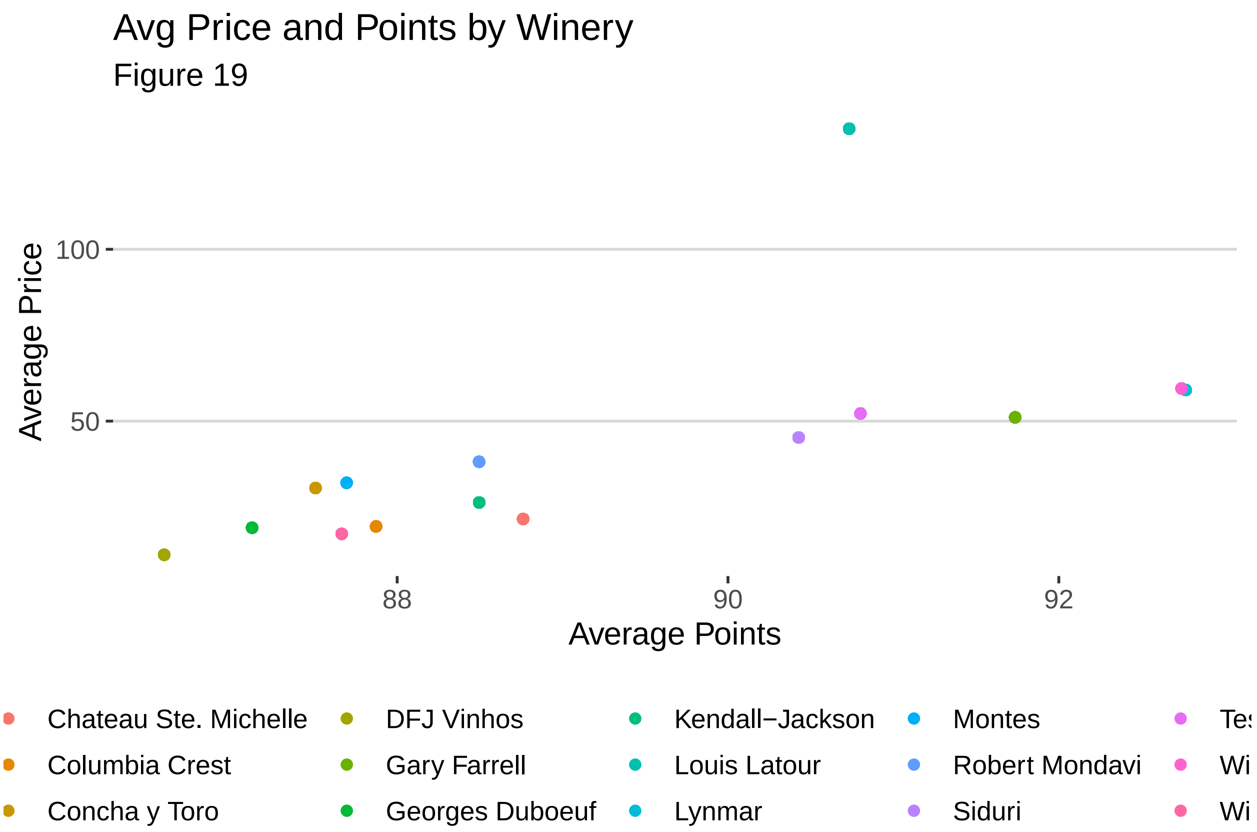
When we observe the breakdown by winery, we see that the distribution is quite even. The most rated wineries only has about 200 observations. The data set is not dominated by a few wineries unlike the country and years categories. Since there are many wineries used in this data set, we are only going to graph the ones were rated over 100 times.



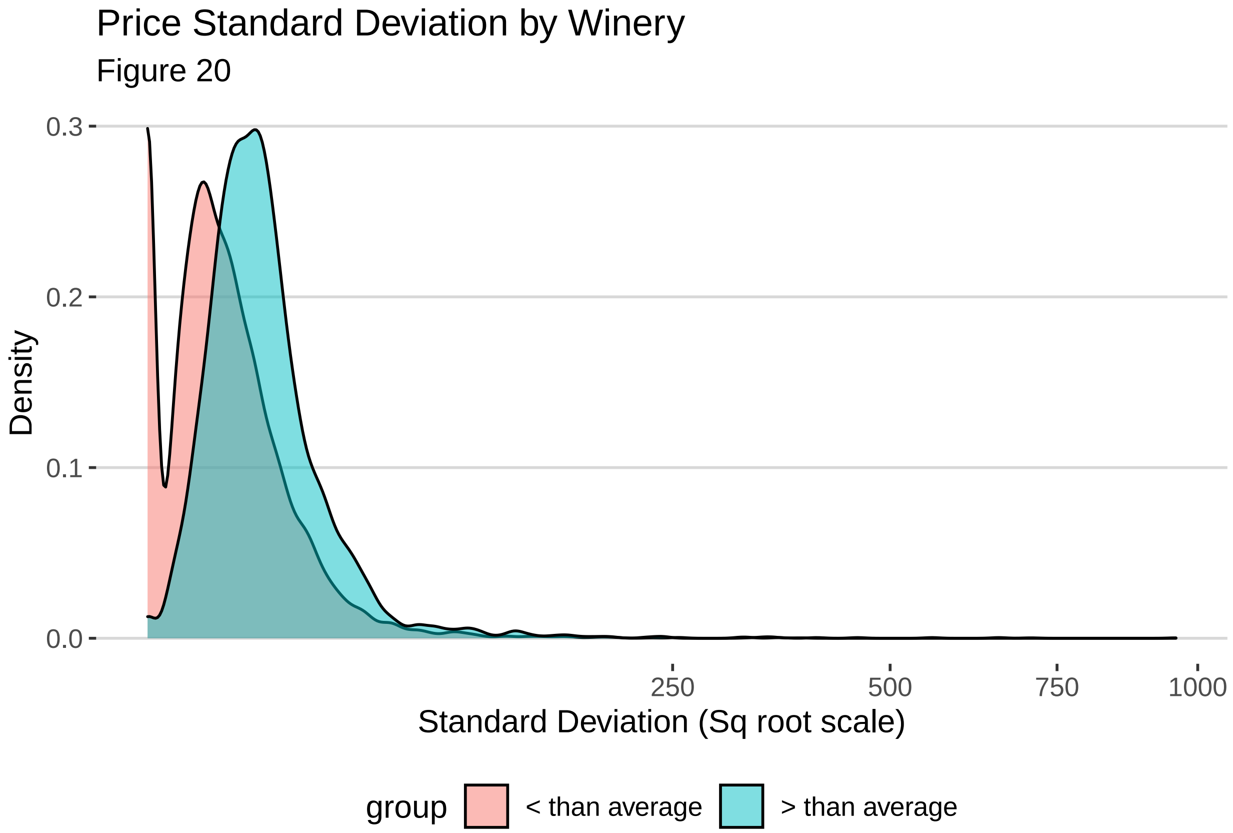
This box plot is the opposite of the distribution for the years category (figure 10). This makes sense as some wineries are very reputable brands and can charge accordingly. The most expensive winery is Louis Latour, the median price for their wine is higher than all of the other winery's upper box limits. On the other end of the spectrum DFJ Vinhos is the cheapest winery, with the median price at 10USD. Next we will take a look at their respective points and see if the price is reflected in their rating.



When looking at the the points distribution, we see that even though Louis Latour is the most expensive winery, they do not have the highest average rating. The two wineries with the highest average ratings are William Selyem and Lynmar. Those 2 wineries are among the highest in median prices, however both of their variance in price are small compared to Louis Latour. On the other end of the spectrum, DFJ Vinhos is the cheapest wineries and they have also received the lowest median. However the Santa Ema winery has higher variance and has more wines rated lower than DFJ Vinhos.



In the winery breakdown, we can observe a linear pattern emerging. All the points besides one (Louis Latour) fits on the linear progression line.



When grouped by winery, the number of times each winery rated does not seem to play a bit part in the price variance. This does make sense as well, as reputable winery generally charge more for all their wines and vice versa.

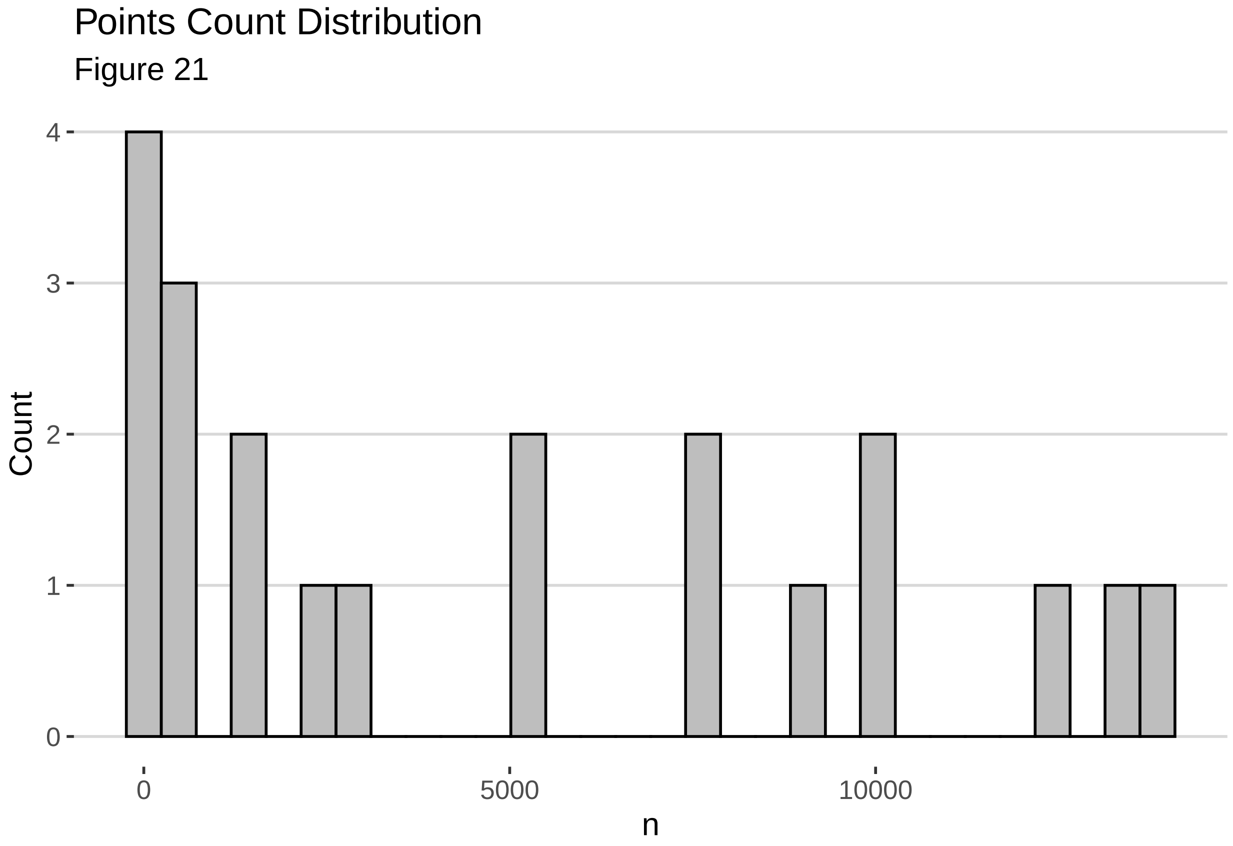
```
winery_avg <- wine_dat %>%
  group_by(winery) %>%
  summarize(winery = winery, avg_pts = mean(points), avg_p = mean(price))

correlate <- bind_rows(correlate,
                       data.frame(Category = "Winery",
                                  Correlation = cor(winery_avg$avg_pts, winery_avg$avg_p)))
correlate
```

```
## # A tibble: 4 x 2
##   Category Correlation
##   <chr>          <dbl>
## 1 Country        0.468
## 2 Variety        0.425
## 3 Years          0.619
## 4 Winery         0.530
```

The winery category has the second highest correlation. Even though the previous graph showed a linear plot, we must remember those are only for wineries that was rated over 100 times. There are over 15000 unique wineries included in this data set. This leads to wineries with small numbers of reviews that skew the data. This is a important factor to consider during the modeling step.

Breakdown by points

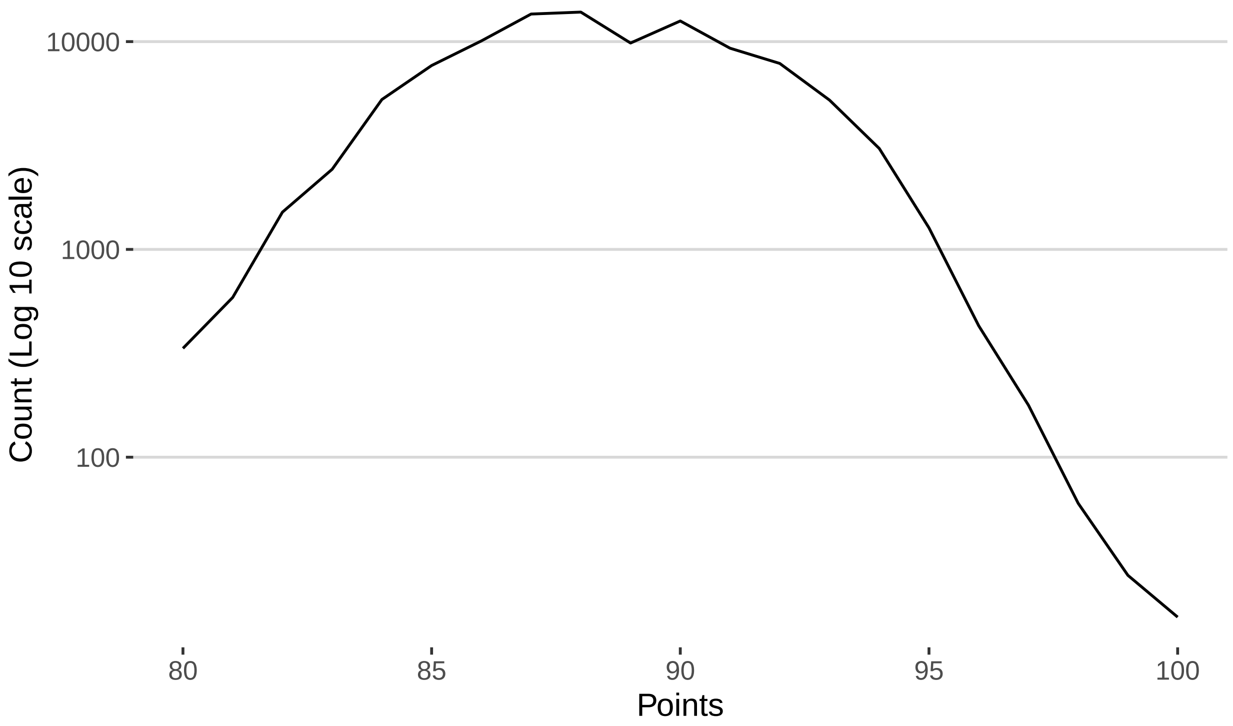
Finally we will dive into the points breakdown to see if we can find any insights.



When we observe the distribution of the number of times each rating has been given, we see that there is a wide disparity between the counts. Out of the 21 different ratings, 10 was rated over 5,000 times - with 5 of those over 10,000 times. Out of the remaining 11, 7 of those has very few ratings and the 4 remaining were rated less than 2,500 times.

Count by Points

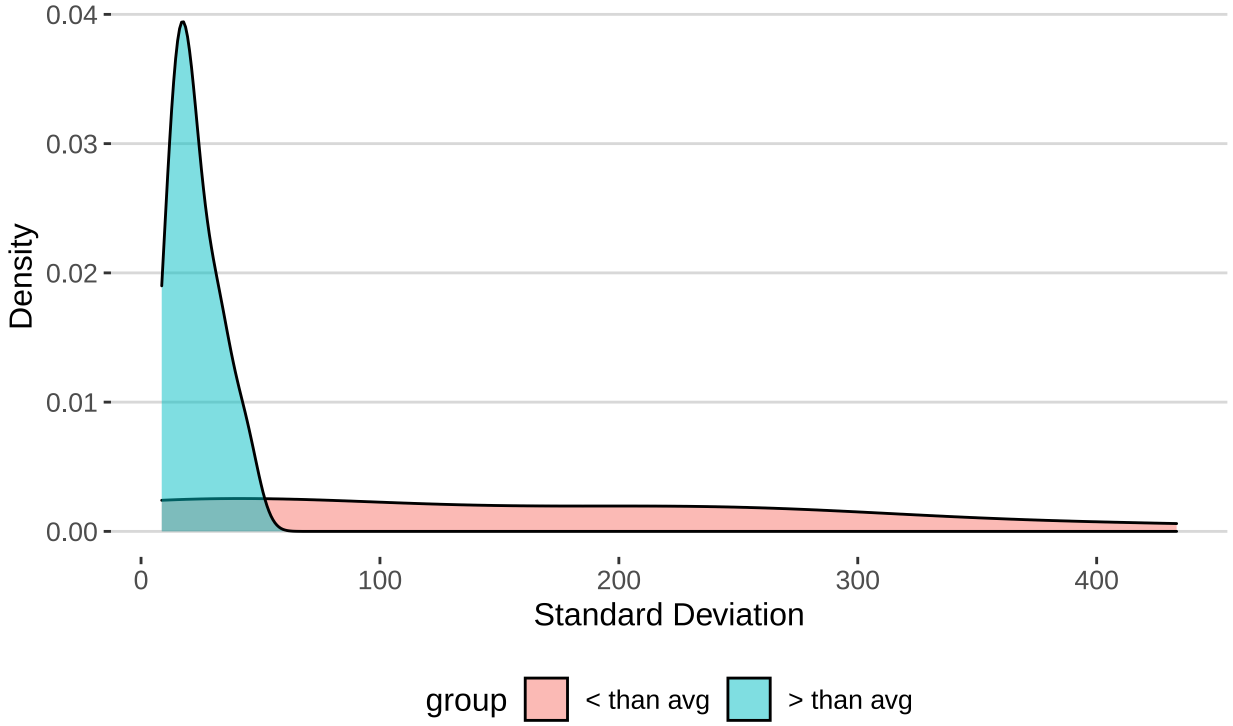
Figure 22



We can observe that from this graph the 5 points that were given the most is from 86-91. This makes sense as the middle in a normal distribution curve is the highest. As we move to the edges the number of times each rating was given goes down. This is especially apparent when we look at the range from 95 to 100, it shows a sharp decline in the number of times each rating was given.

Standard Deviation of Price − Points Group

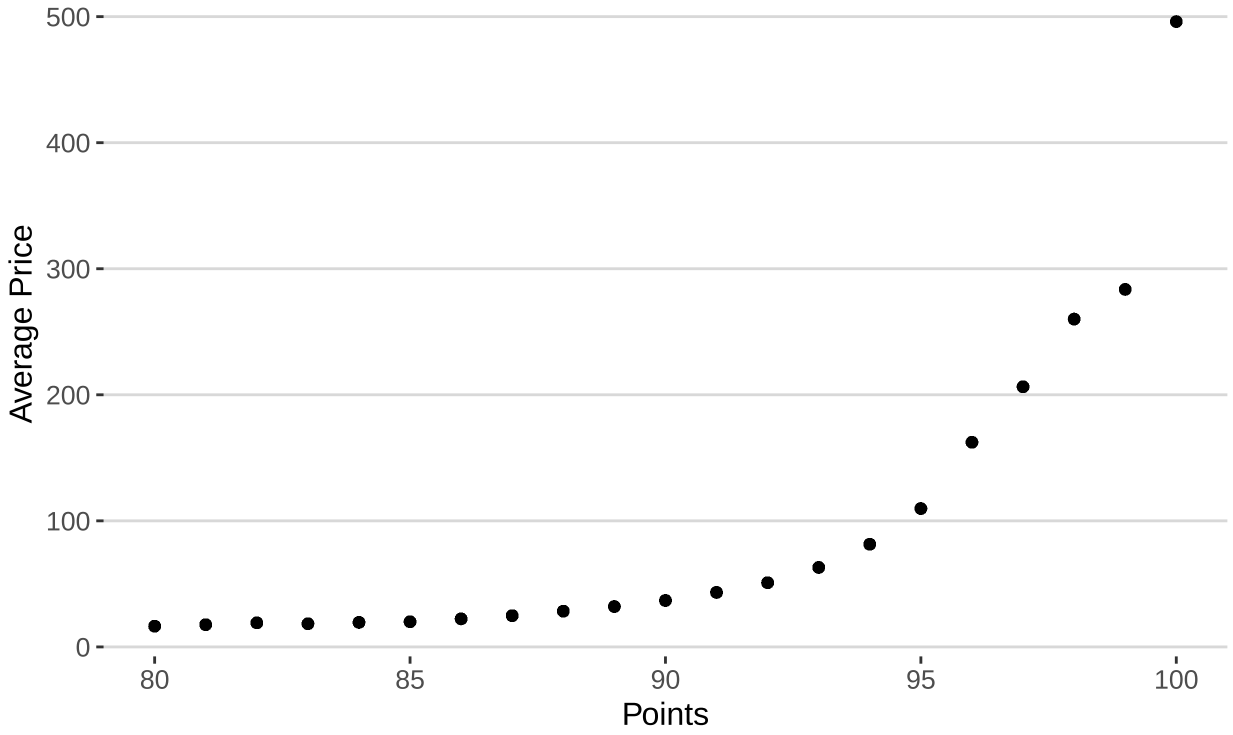
Figure 23



We can see a huge difference with the standard deviation when it was separated by the number of times each point was rated. The points that had was rated more than the average number has a relatively small standard deviation; the other group has a huge variance in the price.

Average Price by Points

Figure 24



There is an exponential curve when we plot the average price against the points. From the rating of 80 - 85 the average price is relatively flat. From 86 - 90 we begin to see increases, from 91+ we can observe exponential increases in the average price as the points increases. The average price of wines that were rated 100 is more than 500USD per bottle.

```r
pts_avg <- wine_dat %>%
  group_by(points) %>%
  summarize(avg_pts = mean(points), avg_p = mean(price))

correlate <- bind_rows(correlate,
                       data.frame(Category = "Points",
                                  Correlation = cor(pts_avg$avg_pts, pts_avg$avg_p)))
correlate
```

```
## # A tibble: 5 x 2
##   Category Correlation
##   <chr>          <dbl>
## 1 Country        0.468
## 2 Variety        0.425
## 3 Years          0.619
## 4 Winery         0.530
## 5 Points         0.794
```

We see there is strong correlation between the average price and it's relative point rating. This confirms our previous graph where we observed the exponential increase in price as the points increased. However, there is a large variance in the rating that were seldom given.

Modeling

For this project, we will use two machine learning models to predict the price of each wine. Our aim is to achieve the lowest possible RMSE. To prevent over training we will further partition the data set. We will only apply the model to the validate set at the very end.

```r
# Partition data into train and test set
set.seed(111, sample.kind = "Rounding")
ind <- createDataPartition(wine_dat$price, times = 1, p = 0.1, list = FALSE)

train_dat <- wine_dat[-ind,]
temp <- wine_dat[ind,]

# Make sure the all the criteria are in both data sets
test_data <- temp %>%
  semi_join(train_dat, by = "country") %>%
  semi_join(train_dat, by = "variety") %>%
  semi_join(train_dat, by = "years") %>%
  semi_join(train_dat, by = "winery")
```

Linear Regression Model

The first model we will explore is the linear regression model. This model takes into account the bias that effects the price in all 4 categories we examined. The formula will be:

$$\hat{y}_i = \mu + b_c + b_v + b_y + b_w + b_p + \varepsilon_i$$

To define the variables:

$\hat{y}_i$ is the Predicted price for wine i.

μ is the average price of all the wines.

b_c is the bias for the country.

b_v is the bias for the variety of wine.

b_y is the bias for the year the wine was produced.

b_w is the bias for the winery that the wine was produced at.

b_p is the bias for the point that each wine was rated.

First we will determine a baseline by predicting the average price to all the wines.

```r
mu <- mean(train_dat$price)

c(min(train_dat$price), max(train_dat$price))
```

```
## [1]    4 2500
```

```r
result <- data.frame(Method = "Baseline",
                     RMSE = RMSE(mu, test_data$price))

result
```

```
##     Method     RMSE
## 1 Baseline 42.55519
```

The RMSE baseline is rather substantial, however this is expected. The price range for the wines is 4USD to 3300USD. This list include rare wines that can skew the numbers quite a bit. Let's factor in the other biases to see how that affects the RMSE.

```
# Country bias
b_c <- train_dat %>%
  group_by(country) %>%
  summarize(b_c = mean(price - mu))

pred_bc <- test_data %>%
  left_join(b_c, by = "country") %>%
  mutate(pred = mu + b_c) %>%
  .$pred

result <- bind_rows(result,
                    data.frame(Method = "Country Bias",
                               RMSE = RMSE(pred_bc, test_data$price)))
result
```

```
##         Method     RMSE
## 1     Baseline 42.55519
## 2 Country Bias 42.16372
```

When the country bias was factored in, there is a slight change in the RMSE. When looking back at our data analysis for the country category, we saw large standard deviation for the countries that were rated the most. Also the number of ratings in those countries made up a large proportion of the data set. Taking that into consideration it does make sense on why this does not have a big affected the RMSE.

```
# Variety bias
b_v <- train_dat %>%
  group_by(variety) %>%
  summarize(b_v = mean(price - mu))

pred_bcv <- test_data %>%
  left_join(b_c, by = "country") %>%
  left_join(b_v, by = "variety") %>%
  mutate(pred = mu + b_c + b_v) %>%
  .$pred

result <- bind_rows(result,
                    data.frame(Method = "Country + Variety",
                               RMSE = RMSE(pred_bcv, test_data$price)))
result
```

```
##              Method     RMSE
## 1          Baseline 42.55519
## 2      Country Bias 42.16372
## 3 Country + Variety 40.61665
```

The variety bias has a decrease in the RMSE when factored in. The standard deviation distribution is similar to the ones for the countries. However, the variety category has 655 unique observations compared to only 43 for countries. This I believe is the reason why there is a larger affect in the variety bias. The few observations with a lot of ratings and high variance does not affect the data as much.

```
# Years Bias
b_y <- train_dat %>%
  group_by(years) %>%
  summarize(b_y = mean(price - mu))

pred_bcvy <- test_data %>%
  left_join(b_c, by = "country") %>%
  left_join(b_v, by = "variety") %>%
  left_join(b_y, by = "years") %>%
  mutate(pred = mu + b_c + b_v + b_y) %>%
  .$pred


result <- bind_rows(result,
                    data.frame(Method = "Country + Variety + Years",
                               RMSE = RMSE(pred_bcvy, test_data$price)))
result
```

```
##                      Method     RMSE
## 1                  Baseline 42.55519
## 2              Country Bias 42.16372
## 3         Country + Variety 40.61665
## 4 Country + Variety + Years 39.99906
```

There is a small decrease in the RMSE when factored in the years category. The standard deviation for the price was very concentrated for the years that were rated the most number of times. There is however large variance for the years that were not rated as much, we can address this when we apply regularization to the algorithm.

```
# Winery Bias
b_w <- train_dat %>%
  group_by(winery) %>%
  summarize(b_w = mean(price - mu))

pred_bcvyw <- test_data %>%
  left_join(b_c, by = "country") %>%
  left_join(b_v, by = "variety") %>%
  left_join(b_y, by = "years") %>%
  left_join(b_w, by = "winery") %>%
  mutate(pred = mu + b_c + b_v + b_y + b_w) %>%
  .$pred

result <- bind_rows(result,
                      data.frame(Method = "Country + Variety + Years + Winery",
                                 RMSE = RMSE(pred_bcvyw, test_data$price)))
result
```

```
##                               Method     RMSE
## 1                              Baseline 42.55519
## 2                          Country Bias 42.16372
## 3                     Country + Variety 40.61665
## 4             Country + Variety + Years 39.99906
## 5 Country + Variety + Years + Winery 35.80823
```

The winery bias provided a large decrease in the RMSE. When we examined the winery category, there was clear differences in both the price and points distribution. Furthermore, the number of times that each winery was rated did not play a large factor in it's standard deviation. All of those factors lead to a more accurate prediction.

```r
# Points bias
b_p <- train_dat %>%
  group_by(points) %>%
  summarize(b_p = mean(price - mu))

pred_bcvywp <- test_data %>%
  left_join(b_c, by = "country") %>%
  left_join(b_v, by = "variety") %>%
  left_join(b_y, by = "years") %>%
  left_join(b_w, by = "winery") %>%
  left_join(b_p, by = "points") %>%
  mutate(pred = mu + b_c + b_v + b_y + b_w + b_p) %>%
  .$pred

result <- bind_rows(result,
                    data.frame(Method = "Country + Variety + Years + Winery + Points",
                               RMSE = RMSE(pred_bcvywp, test_data$price)))
result
```

```
##                                          Method     RMSE
## 1                                      Baseline 42.55519
## 2                                  Country Bias 42.16372
## 3                             Country + Variety 40.61665
## 4                     Country + Variety + Years 39.99906
## 5            Country + Variety + Years + Winery 35.80823
## 6 Country + Variety + Years + Winery + Points 38.74359
```

There is a large increase in the RMSE when accounting for the points bias. In the points standard deviation graph we observed that there is huge variance in the price of the points that were rated less than average. We also observed that there is an exponential increase in the average price as the points went above 95. The same range where there is a sharp decrease in the number of ratings. This high variance with low number of ratings skewed the model, this is another thing we need to keep in mind in the regularization step.

Regularization

In the linear regression model, we observed some biases with low number of occurrence and high variance. The regularization method introduce a tuning parameter to control the variability. This method will shrink the bias towards 0 when there are low number of occurrence, but will be effectively ignored when there is high number of occurrences.

The formula we will use:

$$\sum_i (y_i - \mu - b_v - b_y - b_w - b_p)^2 + \lambda \left(\sum_v b_v^2 + \sum_y b_y^2 + \sum_w b_w^2 + \sum_p b_p^2 \right)$$

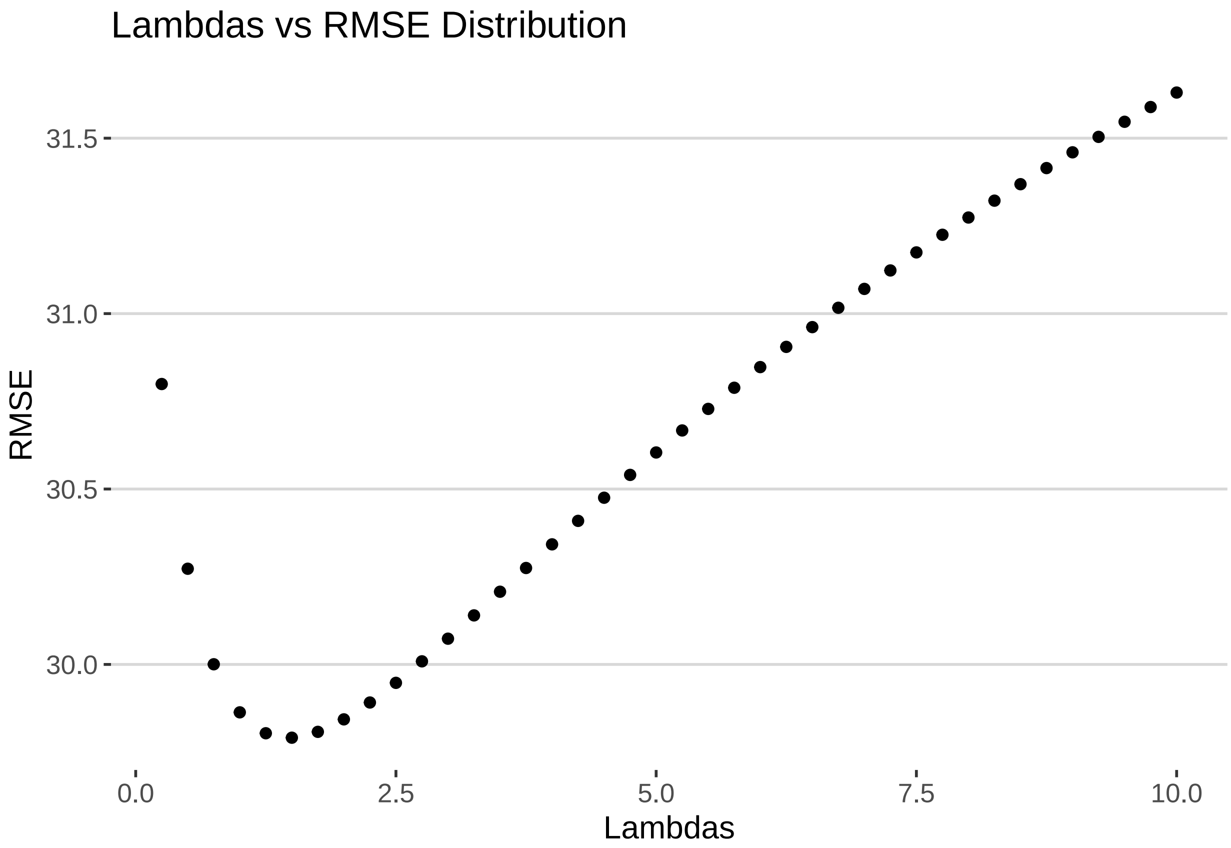
We can solve for each specific bias with this formula:

$$\hat{b}_v(\lambda) = \frac{1}{\lambda + n_v} \sum_{u=1}^{n_v} (Y_i - \hat{\mu})$$

We will not be using the country bias due to the fact that we discovered the most rated countries actually had more variance compared to the seldom rated countries. Regularization will not be able to affect this, this method aims to provide more weight to the variables that appeared more in the data.

Lambda is a tuning parameter, we can use machine learning to determine what the value is to give the lowest RMSE. We will run the lambda at the range of 0.25 to 10 at 0.25 increments.

```r
# establish lambda range
lambdas <- seq(0.25, 10, 0.25)

# regularization function
regularization_rmse <- sapply(lambdas, function(l){
  # variety regularization
  bv_r <- train_dat %>%
    group_by(variety) %>%
    summarize(bv_r = sum(price - mu)/(n() + l))
  # years regularization
  by_r <- train_dat %>%
    left_join(bv_r, by = "variety") %>%
    group_by(years) %>%
    summarize(by_r = sum(price - bv_r - mu)/(n() + l))
  # winery regularization
  bw_r <- train_dat %>%
    left_join(bv_r, by = "variety") %>%
    left_join(by_r, by = "years") %>%
    group_by(winery) %>%
    summarize(bw_r = sum(price - bv_r - by_r - mu)/(n() + l))
  # points regularization
  bp_r <- train_dat %>%
    left_join(bv_r, by = "variety") %>%
    left_join(by_r, by = "years") %>%
    left_join(bw_r, by = "winery") %>%
    group_by(points) %>%
    summarize(bp_r = sum(price - bv_r - by_r - bw_r - mu)/(n() + l))
  # predict algorithm
  pred_reg <- test_data %>%
    left_join(bv_r, by = "variety") %>%
    left_join(by_r, by = "years") %>%
    left_join(bw_r, by = "winery") %>%
    left_join(bp_r, by = "points") %>%
    mutate(pred = mu + bv_r + by_r + bw_r + bp_r) %>%
    .$pred

  return(RMSE(pred_reg, test_data$price))
})

# Graph Lambdas vs RMSE
data.frame(Lambdas = lambdas, RMSE = regularization_rmse) %>%
  ggplot(aes(Lambdas, RMSE)) +
  geom_point() +
  theme_hc() +
  ggtitle("Lambdas vs RMSE Distribution")
```



```
lambda <- lambdas[which.min(regularization_rmse)]
lambda
```

```
## [1] 1.5
```

```
result <- bind_rows(result,
                    data.frame(Method = "Regularization",
                               RMSE = min(regularization_rmse)))
result
```

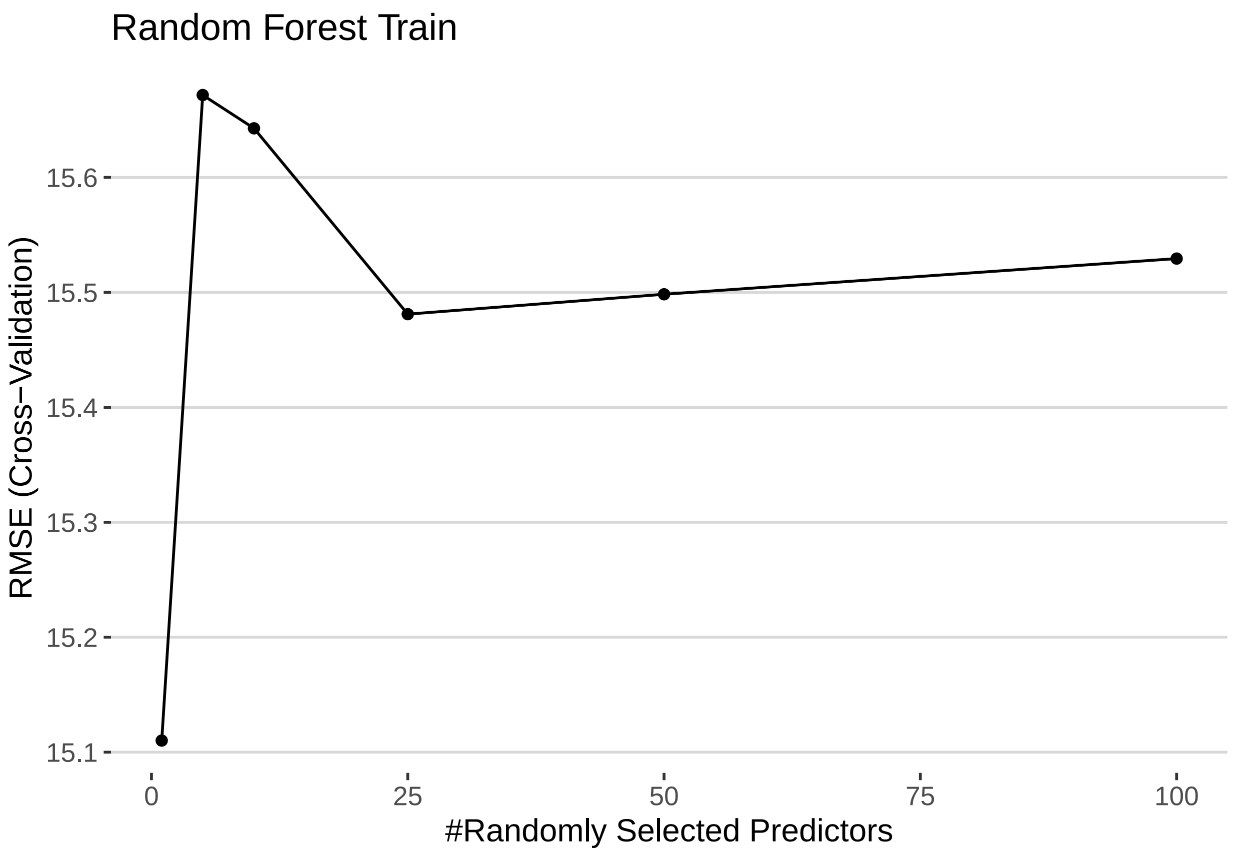
```
##                                        Method     RMSE
## 1                                    Baseline 42.55519
## 2                                Country Bias 42.16372
## 3                           Country + Variety 40.61665
## 4                   Country + Variety + Years 39.99906
## 5          Country + Variety + Years + Winery 35.80823
## 6 Country + Variety + Years + Winery + Points 38.74359
## 7                              Regularization 29.79118
```

The regularization provided a drastic decrease in the RMSE. We observe from the graph that the lambda at 1.5 provided the lowest RMSE. This was anticipated when we were observing the standard deviation differences in the different features.

Random Forest

Random forest is a popular machine learning algorithm. The “forest” that it builds is made up of multiple decision trees to create an ensemble, then it combines them to reduce variance. Another advantage is that factors in variable importance. As we saw from our data analysis, some variables has low sampling with high variance.

```r
# Set column index for model
col_index <- c("points", "winery", "variety", "years", "country")

# set cross validation and RF tuning parameter
control <- trainControl(method = "cv", number = 5)
grid <- data.frame(mtry = c(1,5,10,25,50,100))

# create smaller sample of data
n = 10000

index <- sample(train_dat$price, n)

train_small <- train_dat[index,]

# randomforest training
train_rf <- train(train_small[,col_index], train_small$price,
                method = "rf",
                trControl = control,
                tuneGrid = grid)

ggplot(train_rf) +
  theme_hc() +
  ggtitle("Random Forest Train")
```



```
train_rf$bestTune
```

```
##   mtry
## 1    1
```

In this algorithm, I implemented 5 fold cross validation on a small portion of the file to build the training model. I also tuned the mtry(the number of randomly selected predictors) at 1,5,10,25,50,100. As we can observe from the graph, the mtry of 1 provides the lowest RMSE.

```
imp <- varImp(train_rf)
imp
```

```
## rf variable importance
##
##         Overall
## winery   100.00
## variety   71.52
## years     63.30
## points    32.78
## country    0.00
```

Here we are able to see the decision tree that the determined the modeling. The most important predictor is the winery and the country variable has no effect on the calculation.

```
# Fit the data with lowest mtry
rf_fit <- randomForest(train_small[,col_index], train_small$price,
                       minNode = train_rf$bestTune$mtry)

# predict the test set price
pred_rf <- predict(rf_fit, test_data[,col_index])

result <- bind_rows(result, data.frame(Method = "Random Forest",
                                       RMSE = RMSE(pred_rf, test_data$price)))
result
```

```
##                                            Method     RMSE
## 1                                        Baseline 42.55519
## 2                                    Country Bias 42.16372
## 3                               Country + Variety 40.61665
## 4                       Country + Variety + Years 39.99906
## 5              Country + Variety + Years + Winery 35.80823
## 6 Country + Variety + Years + Winery + Points 38.74359
## 7                                  Regularization 29.79118
## 8                                   Random Forest 39.06898
```

Random Forest did return with a relatively large RMSE, there are a few reasons for this. Random Forest is better at predicting categorical values compared to the continuous values. However, I thought it would be an interesting exercise to observe the variable importance of each predictors.

Another reason is I only trained the algorithm with 10,000 rows of data, this is due to hardware limitations.

Results

The importance of machine learning is to test the algorithm that has not been any part of the training process. We will test our models on the validate set.

```
# variety regularization
bv_r <- wine_dat %>%
  group_by(variety) %>%
  summarize(bv_r = sum(price - mu)/(n() + lambda))
# years regularization
by_r <- wine_dat %>%
  left_join(bv_r, by = "variety") %>%
  group_by(years) %>%
  summarize(by_r = sum(price - bv_r - mu)/(n() + lambda))
# winery regularization
bw_r <- wine_dat %>%
  left_join(bv_r, by = "variety") %>%
  left_join(by_r, by = "years") %>%
  group_by(winery) %>%
  summarize(bw_r = sum(price - bv_r - by_r - mu)/(n() + lambda))
# points regularization
bp_r <- wine_dat %>%
  left_join(bv_r, by = "variety") %>%
  left_join(by_r, by = "years") %>%
  left_join(bw_r, by = "winery") %>%
  group_by(points) %>%
  summarize(bp_r = sum(price - bv_r - by_r - bw_r - mu)/(n() + lambda))
# predict algorithm
pred_val <- validate_dat %>%
  left_join(bv_r, by = "variety") %>%
  left_join(by_r, by = "years") %>%
  left_join(bw_r, by = "winery") %>%
  left_join(bp_r, by = "points") %>%
  mutate(pred = mu + bv_r + by_r + bw_r + bp_r) %>%
  .$pred

validate <- data.frame(Method = "Regularization",
                       RMSE = RMSE(pred_val, validate_dat$price))
validate
```

```
##           Method     RMSE
## 1 Regularization 22.06887
```

The regularized linear modeling provided great result, let's do the same for the random forest model.

```
rf_val <- predict(rf_fit, validate_dat[,col_index])

validate <- bind_rows(Method = "RandomForest",
                      RMSE = RMSE(rf_val, validate_dat$price))
validate
```

```
## # A tibble: 1 x 2
##   Method        RMSE
##   <chr>         <dbl>
## 1 RandomForest  27.8
```

Here we see the RMSE results for both models. In my opinion, I think both of these models produced good results. It would be easier for me to pick a categorical outcome, however I found it more challenging and interesting to try to predict the price. Even though the price range in this data set is quite large, both of the models provided a low RMSE.

Conclusion

In this project we explored the data that was from the wineEthusiast that contained the list of wine with their relative price, points, origin and variety. Through our data analysis we gained insights into how different features affected the price of the wine.

Some of the limitations of this report is hardware related. I would like to have experimented with training with the random forest model on the whole data set with more cross-validation to see it's possible to lower the RMSE even further. However, I am happy with the result that the model provided.

Future work I would like to explore more with the specific reviews that were given by each reviewer. It would be interesting to see if any of their key words had any affect on the wine's points or price. For example: dry vs sweet; or fruity vs herbal. I think it would be a good practice to apply for regex filters.

I hope this report can give readers more knowledge about picking wine. Instead of only looking at it's point rating, as we saw from this project; there are other factors that may affect the price. We also learned a higher rated wine is not necessarily more expensive. I want to encourage the readers to explore different variety and wineries to find new wines that you will enjoy!