IST 687 Final Project Technical Analysis

Wine Tasting Analysis

○ Introduction

Description of Dataset

Team #2 chose a dataset of wine reviews and ratings collected and made publicly available on Kaggle to analyze for this project. The dataset comprised two Microsoft Excel Comma Separated Values (CSV) files that were combined for a total of over 97,000 values. The combined dataset included the following categories:

- Country of origin,
- Province of origin,
- Region of origin,
- Winery of origin,
- Description of the wine,
- Variety of the wine,
- Points (wine rating),
- Price of the wine, and
- Name of the wine tasters.

Some categories were not included in the combined data set and / or used for analysis, including designation of the wine (the vintage), secondary region (often blank or a duplicate of the primary region), title of the wine, and Twitter handle of the wine taster.

There are some obvious biases to the dataset that cannot be overcome for this analysis. The wine ratings come from a United States-based publication, so a large number of the rated wines come from the United States, California in particular. There was no way to avoid a significant bias towards United States and California wines without combining data from foreign sources, which we did not have access to.

○ Data Wrangling

This section details the process of transforming the dataset used for analysis. The R code is provided with explanations of each step in combining and cleaning the dataset.

The first step was reading in the two separate CSV files into data frames.

```
#Import the first csv file
wine1 <- read.csv(file = 'winemag-data_first150k.csv')

#Import the second csv file
wine2 <- read.csv(file = 'winemag-data-130k-v2.csv')
```

The next steps were cleaning the two data frames by removing unnecessary columns and removing duplicate entries.

```
# Remove X column from wine1, set rownames to NULL, remove duplicate rows
wine1 <- wine1[,-1]
rownames(wine1) <- NULL
wine1[!duplicated(wine1$description),]

# Remove X column from wine 2, set rownames to NULL, remove duplicate rows
wine2 <- wine2[,-1]
rownames(wine2) <- NULL
```

```
wine2[!duplicated(wine2$description),]

# Remove taster, and variety columns from wine2 so it can be combined with wine1
wine2 <- wine2[,-9:-11]
```

Once the data frames were prepared, the next step was to merge them into one data frame.

```
# Merge data frames into one data frame
wineData <- merge(wine1, wine2,by=c('country', 'description', 'designation', 'points', 'price', 'province', 'region_1', 'region_2', 'variety', 'winery'), all.x=T)
```

Lastly, duplicate entries as a result of the merge were removed, blank countries were removed, and periods from the descriptions were removed.

```
# Remove duplicate rows from description
cleanWine <- wineData[!duplicated(wineData$description),]

# Remove periods
cleanWine$description <- gsub("\\.","",cleanWine$description)

# Remove blank country values
cleanWine <- cleanWine[-1:-3,]
```

This process created the main data frame from which analysis was accomplished. For certain analysis techniques, custom data frames were created to for easier processing of specific categories and relationships. These custom data frames are detailed in the following section as they apply to the business questions.

o Data Analysis & Results

This section details the business questions that were selected for this dataset, the types of analyses performed to answer the business questions, and the associated R code for the analysis techniques.

Each business question includes a description of the question, the analysis techniques performed, the associated R code, and any visualizations generated from the results.

ARE WINE RATERS BIASED?

- **Description**

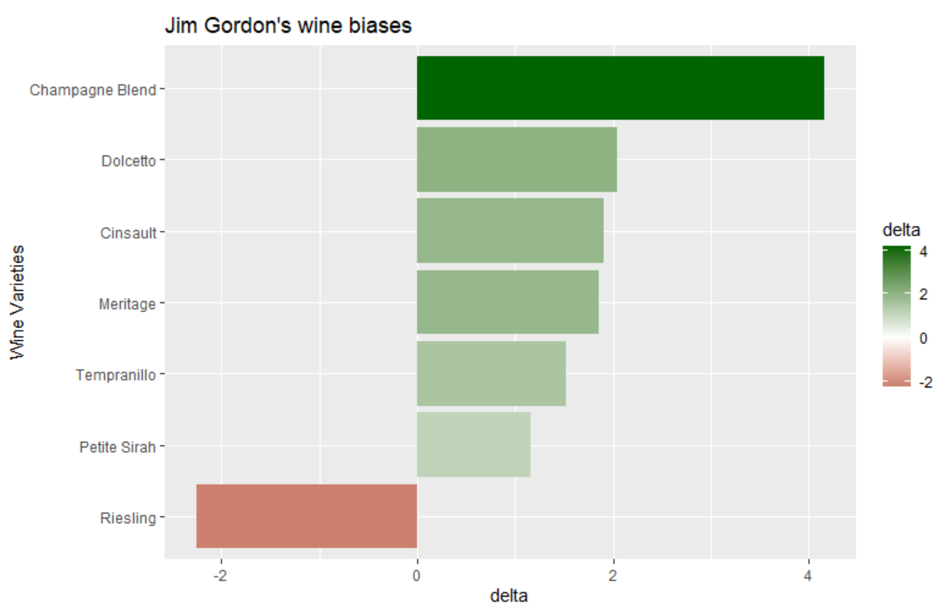
In this analysis, we investigated whether the wine tasters prefered certain varieties of wines by awarding higher points than the average taster.

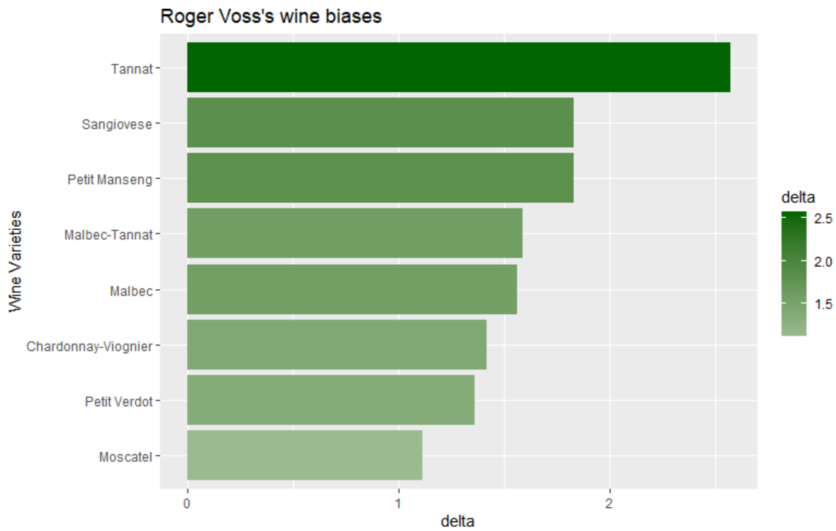
- **Analysis**

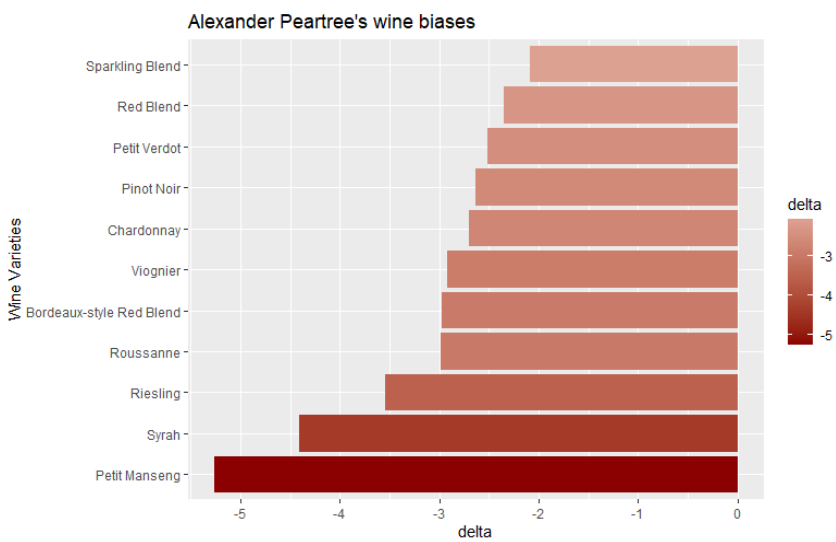
```
#=====================
# Taster Bias Analysis
#=====================
#Group the reviews by taster and variety
taster_df <- sqldf("SELECT taster_name, variety, avg(points) as avg_points, avg(price) as avg_price, count(*) as num_reviews
```

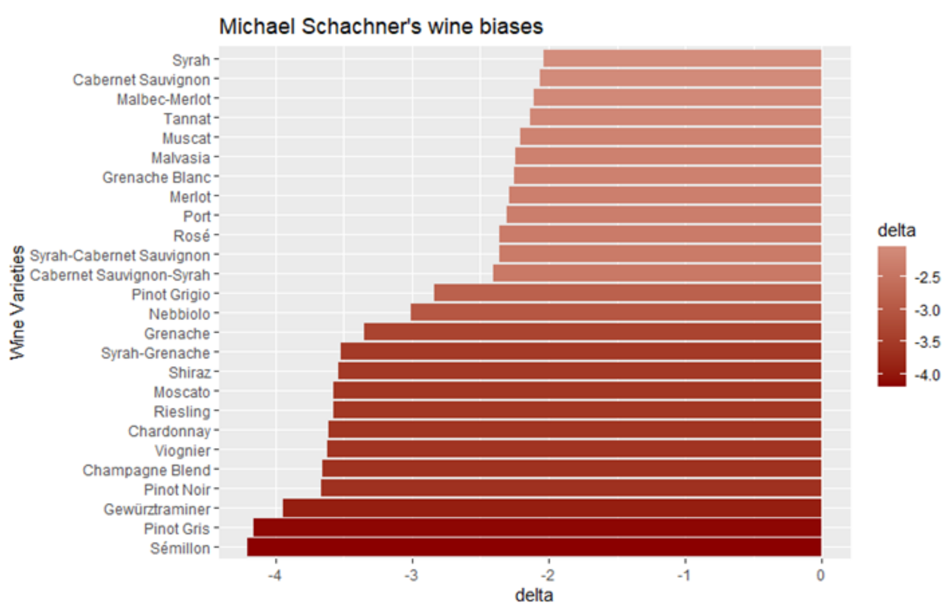
```
                     from df group by taster_name, variety")
variety_df <- sqldf("SELECT variety, avg(points) as avg_points, avg(price) as
avg_price, count(*) as num_reviews
                     from df group by variety")

#Filter to varieties with at least 10 reviews and join with the taster df
variety_df <- variety_df[variety_df$num_reviews > 10,]
taster_df <- taster_df[taster_df$variety %in% variety_df$variety,]
taster_df <- sqldf("SELECT t.*, v.avg_points as v_avg_points
                     from taster_df as t
                     join variety_df as v
                     on t.variety = v.variety")
#Calculate the delta between a taster's ratings
#and the average for that variety, only keep the most biased
taster_df$delta <- taster_df$avg_points - taster_df$v_avg_points
t_df <- t_df[(t_df$delta > 1 | t_df$delta < -2),]

#only look at varieties the tasters have tried at least 5 times
t_df <- taster_df[taster_df$num_reviews > 5,]

#Print out bias charts for tasters with clear biases
ggplot(t_df[t_df$taster_name == 'Jim Gordon',], aes(x = delta, y =
reorder(variety, delta), fill=delta)) +
  geom_bar(stat="identity") +
  ylab("Wine Varieties") + ggtitle("Jim Gordon's wine biases") +
  scale_fill_gradient2(low="darkred", high="darkgreen", midpoint=0)
```

- **Visualization**




Roger Voss's wine biases
Wine Varieties
Tannat
Sangiovese
Petit Manseng
Malbec-Tannat
Malbec
Chardonnay-Viognier
Petit Verdot
Moscatel
0
1
2
delta
delta
2.5
2.0
1.5


Alexander Peartree's wine biases
Wine Varieties
Sparkling Blend
Red Blend
Petit Verdot
Pinot Noir
Chardonnay
Viognier
Bordeaux-style Red Blend
Roussanne
Riesling
Syrah
Petit Manseng
-5
-4
-3
-2
-1
0
delta
delta
-3
-4
-5


Michael Schachner's wine biases
Wine Varieties
Syrah
Cabernet Sauvignon
Malbec-Merlot
Tannat
Muscat
Malvasia
Grenache Blanc
Merlot
Port
Rosé
Syrah-Cabernet Sauvignon
Cabernet Sauvignon-Syrah
Pinot Grigio
Nebbiolo
Grenache
Syrah-Grenache
Shiraz
Moscato
Riesling
Chardonnay
Viognier
Champagne Blend
Pinot Noir
Gewürztraminer
Pinot Gris
Sémillon
-4
-3
-2
-1
0
delta
delta
-2.5
-3.0
-3.5
-4.0

WHAT KINDS OF TERMS ARE WINE RATERS USING TO DESCRIBE DIFFERENT TYPES OF WINE?

- **Description**

To determine the most important terms the raters used to describe the different types of wine, we created a word cloud using the mean points scores of each wine. We analyzed the top three wines by mean points to see how tasters described the “best” wines.

- **Analysis**

```
> library (wordcloud)
> library(RColorBrewer)
> library(wordcloud2)
> library(tm)
> library(NLP)
#---------------------------#
#Text Mining/Clean Data - Top wine in points - Sangiovese Grosso 90.32644
> Grosso_df <- cleanWine[cleanWine$variety=="Sangiovese Grosso",]
> words.vec1 <- Grosso_df[["description"]]
> wordsVecOne <- VectorSource(words.vec1)
> words.corpusOne <- Corpus(VectorSource(Grosso_df$description))
> words.corpusOne
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 870
> words.corpusOne <- tm_map(words.corpusOne, content_transformer(tolower))
Warning message:
In tm_map.SimpleCorpus(words.corpusOne, content_transformer(tolower)) :
  transformation drops documents
> words.corpusOne <- tm_map(words.corpusOne, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(words.corpusOne, removePunctuation) :
  transformation drops documents
> words.corpusOne <- tm_map(words.corpusOne, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(words.corpusOne, removeNumbers) :
  transformation drops documents
> words.corpusOne <- tm_map(words.corpusOne, removeWords,
stopwords("english"))
Warning message:
In tm_map.SimpleCorpus(words.corpusOne, removeWords, stopwords("english")) :
  transformation drops documents
> tdmOne <- TermDocumentMatrix(words.corpusOne)
> tdmOne
<<TermDocumentMatrix (terms: 2317, documents: 870)>>
Non-/sparse entries: 21642/1994148
Sparsity           : 99%
Maximal term length: 23
Weighting          : term frequency (tf)
#-----------------------------------------------------------#
#Creating word cloud - Sangiovese Grosso
> mOne <- as.matrix(tdmOne)
> wordCountsOne <- rowSums(mOne)
> wordCountsOne <- sort(wordCountsOne, decreasing=TRUE)
> head(wordCountsOne)
    wine    spice brunello
     610      448      437
```

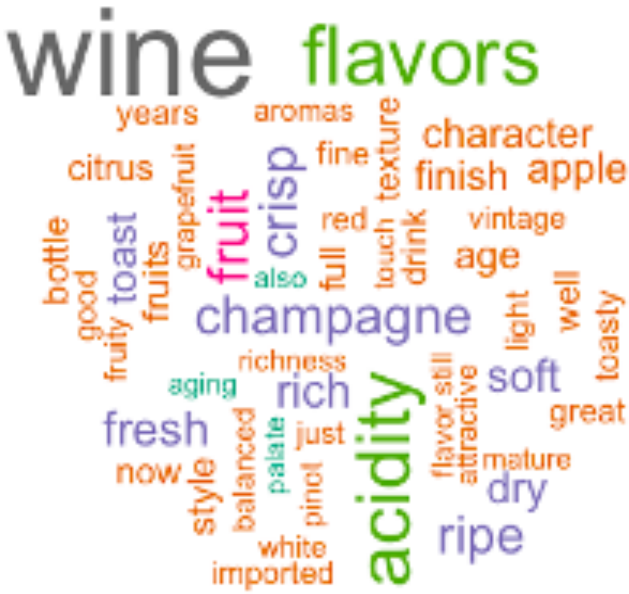
```
   fruit   cherry   aromas 
     430      425      414 
> cloudFrameOne <- data.frame(word=names(wordCountsOne), freq=wordCountsOne)
> wordcloud(names(wordCountsOne), wordCountsOne, min.freq=2, max.words=50, 
rot.per=0.35, colors=brewer.pal(8, "Dark2"))
#----------------------------------------#
#Text Mining/Clean Data - 2nd highest points - Nebbiolo 90.23152
> Nebb_df <- cleanWine[cleanWine$variety=="Nebbiolo",]
> words.vec <- Nebb_df[["description"]]
> wordsVec <- VectorSource(words.vec)
> words.corpus <- Corpus(VectorSource(Nebb_df$description))
> words.corpus <- tm_map(words.corpus, content_transformer(tolower))
Warning message:
In tm_map.SimpleCorpus(words.corpus, content_transformer(tolower)) :
  transformation drops documents
> words.corpus <- tm_map(words.corpus, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(words.corpus, removePunctuation) :
  transformation drops documents
> words.corpus <- tm_map(words.corpus, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(words.corpus, removeNumbers) :
  transformation drops documents
> words.corpus <- tm_map(words.corpus, removeWords, stopwords("english"))
Warning message:
In tm_map.SimpleCorpus(words.corpus, removeWords, stopwords("english")) :
  transformation drops documents
> tdm <- TermDocumentMatrix(words.corpus)
> tdm
<<TermDocumentMatrix (terms: 3444, documents: 1339)>>
Non-/sparse entries: 36206/4575310
Sparsity           : 99%
Maximal term length: 25
Weighting          : term frequency (tf)
#Creating word cloud - Nebbiolo
> m <- as.matrix(tdm)
> wordCounts <- rowSums(m)
> wordCounts <- sort(wordCounts, decreasing=TRUE)
> head(wordCounts)
 aromas    wine tannins  cherry 
    839     782     718     583 
  fruit   spice 
    522     510 
> cloudFrame <- data.frame(word=names(wordCounts), freq=wordCounts)
> wordcloud(names(wordCounts), wordCounts, min.freq=2, max.words=50, 
rot.per=0.35, colors=brewer.pal(8, "Dark2"))
#----------------------------------------#

#Text Mining/Clean Data - 3rd - Champagne Blend 89.62515
Champ_df <- cleanWine[cleanWine$variety=="Champagne Blend",]
> words.vec <- Champ_df[["description"]]
> wordsVec <- VectorSource(words.vec)
> words.corpus <- Corpus(VectorSource(Champ_df$description))
> words.corpus <- tm_map(words.corpus, content_transformer(tolower))
Warning message:
In tm_map.SimpleCorpus(words.corpus, content_transformer(tolower)) :
  transformation drops documents
> words.corpus <- tm_map(words.corpus, removePunctuation)
```

```
Warning message:
In tm_map.SimpleCorpus(words.corpus, removePunctuation) :
  transformation drops documents
> words.corpus <- tm_map(words.corpus, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(words.corpus, removeNumbers) :
  transformation drops documents
> words.corpus <- tm_map(words.corpus, removeWords, stopwords("english"))
Warning message:
In tm_map.SimpleCorpus(words.corpus, removeWords, stopwords("english")) :
  transformation drops documents
> tdm <- TermDocumentMatrix(words.corpus)
> tdm
<<TermDocumentMatrix (terms: 2853, documents: 811)>>
Non-/sparse entries: 18456/2295327
Sparsity           : 99%
Maximal term length: 21
Weighting          : term frequency (tf)
#Creating word cloud - Champagne Blend
> m <- as.matrix(tdm)
> wordCounts <- rowSums(m)
> wordCounts <- sort(wordCounts, decreasing=TRUE)
> head(wordCounts)
     wine   flavors   acidity
      621       385       353
    fruit     crisp champagne
      239       223       222
> cloudFrame <- data.frame(word=names(wordCounts), freq=wordCounts)
> wordcloud(names(wordCounts), wordCounts, min.freq=2, max.words=50,
rot.per=0.35, colors=brewer.pal(8, "Dark2"))
```

- **Visualization**

Sangiovese Grosso - 90.3 Average Points

Nebbiolo - 90.2 Average Points



Sangiovese Grosso - 90.3 Average Points

DO CERTAIN VARIETIES OF WINE HAVE A HIGH RATION OF RATING / PRICE?

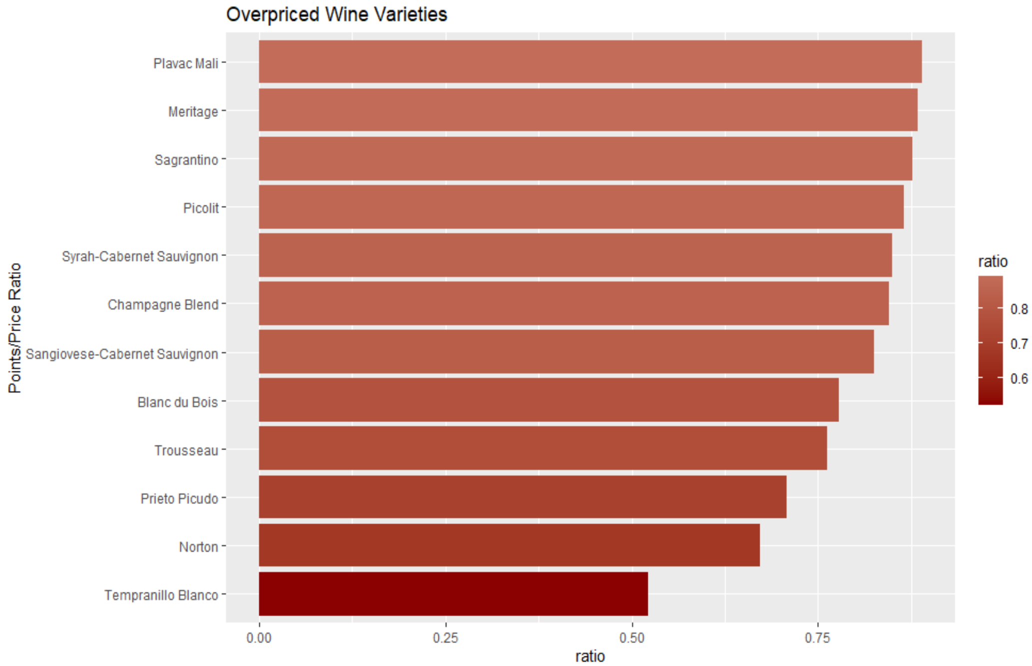
- **Description**

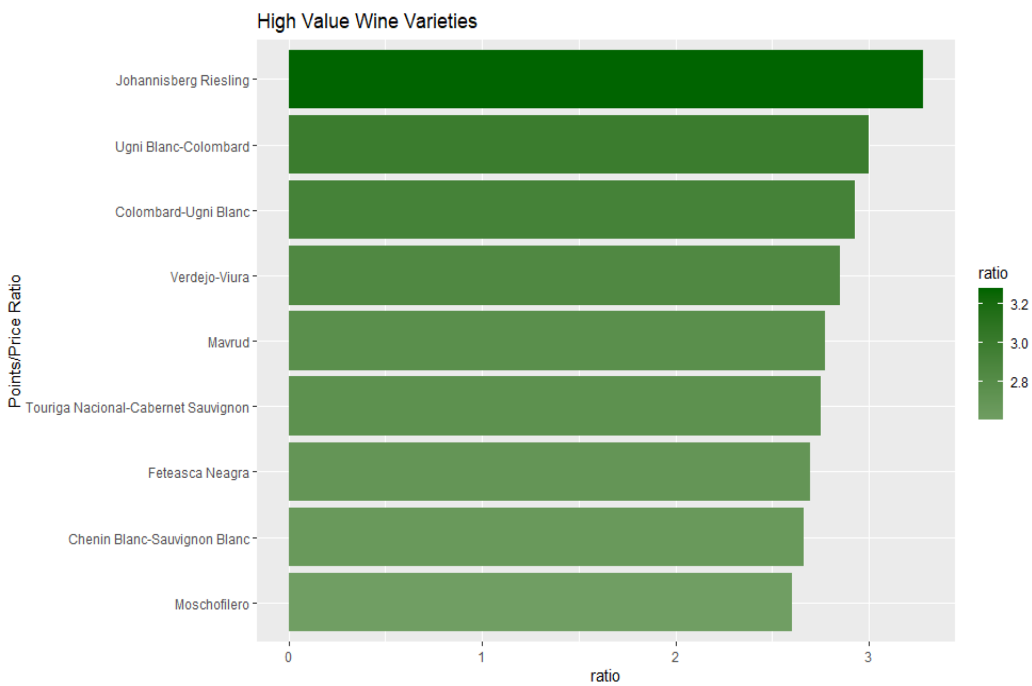
The purpose of this investigation is to determine if some wine varieties are overpriced. Though we show later in our analysis that points can be predicted by price, the wines in this section represent significant outliers.

- **Analysis**

```
#=============================
# Finding points/price ratios
#=============================

#Scale points and price
df$points <- (df$points-min(df$points))/(max(df$points) - min(df$points))
df$price <- (df$price-min(df$price))/(max(df$price) - min(df$price))

variety_df <- sqldf("SELECT variety, avg(price) as avg_price,
                    avg(points) as avg_points, avg(points)/avg(price) as
ratio,
                    count(*) As num_wines FROM df GROUP BY variety
                    HAVING num_wines > 5")

high_df <- variety_df[variety_df$ratio > 2.5,]
low_df <- variety_df[variety_df$ratio < 0.9,]

ggplot(high_df, aes(x = ratio, y = reorder(variety, ratio), fill=ratio)) +
  geom_bar(stat="identity") +
  ylab("Points/Price Ratio") + ggtitle("High Value Wine Varieties") +
  scale_fill_gradient2(low="darkred", high="darkgreen", midpoint=1.5)

ggplot(low_df, aes(x = ratio, y = reorder(variety, ratio), fill=ratio)) +
  geom_bar(stat="identity") +
  ylab("Points/Price Ratio") + ggtitle("Overpriced Wine Varieties") +
  scale_fill_gradient2(low="darkred", high="darkgreen", midpoint=1.5)
```

- **Visualization**





WHICH REGIONS / COUNTRIES HAVE THE MOST / LEAST VARIETIES OF WINE?

- **Description**

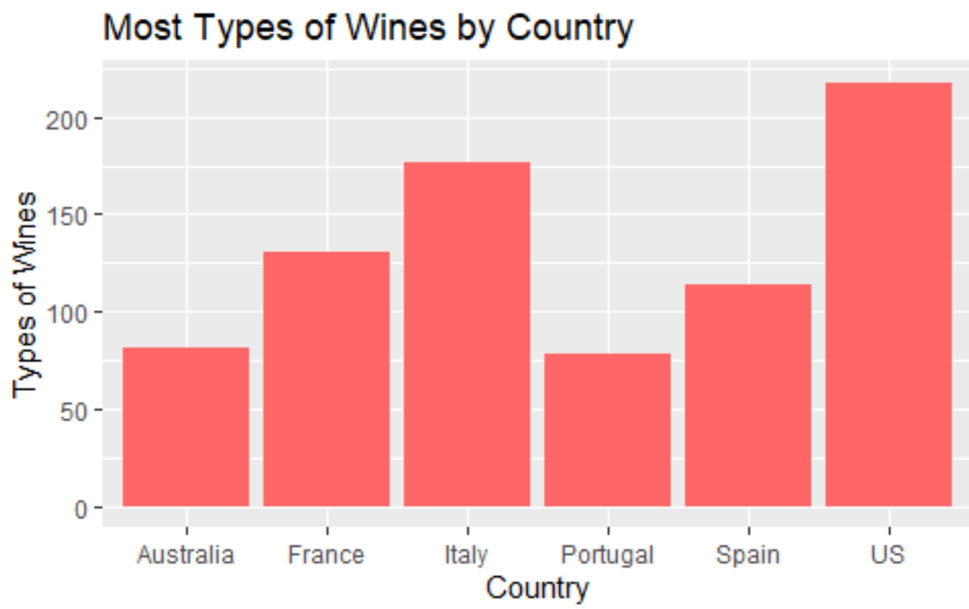
To determine the most and least wine varieties by country and state/province, we group the wine varieties by country to get a column of individual countries and another column of the number of wine varieties for each corresponding country in our dataset. Then, we sorted the dataset in descending order by number of wine varieties to get the countries with the most varieties and in ascending order by number of wine varieties to get the countries with the least varieties. We repeated the same steps for state/province.

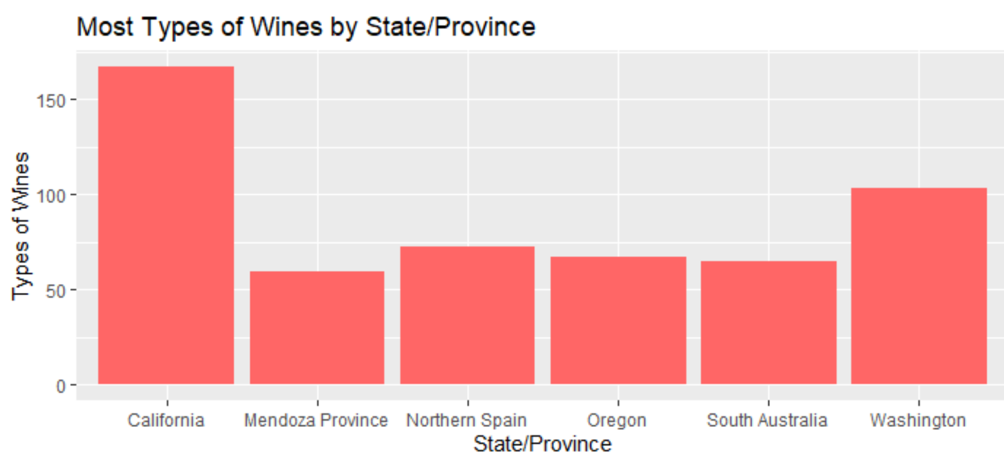
An additional question was asked within this business question: what is the distribution of price and points by country? A custom data frame was created that grouped price and points, then counted those groupings by country. A heat map was created to visualize the distribution of grouped price / points by country with a color scale. Based on this visualization, we can see the United States, Spain, Portugal, Italy, France, Chile, Australia, and Argentina have the highest numbers of highly priced *and* rated wines. The visualization also allows us to see the overlap in price / point pairings for each country. The prices vary considerably by points, but there is a visible pattern that as points increase, so does price.

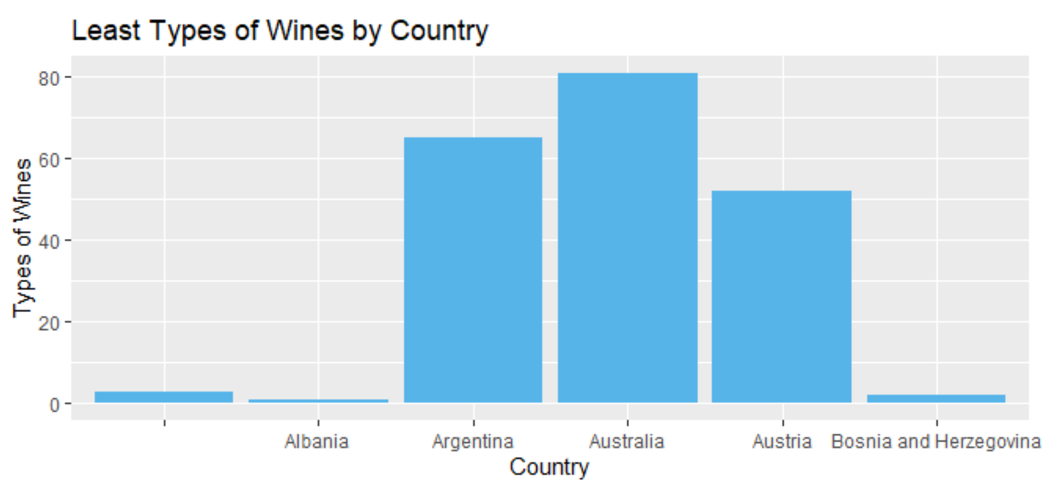
- **Analysis**

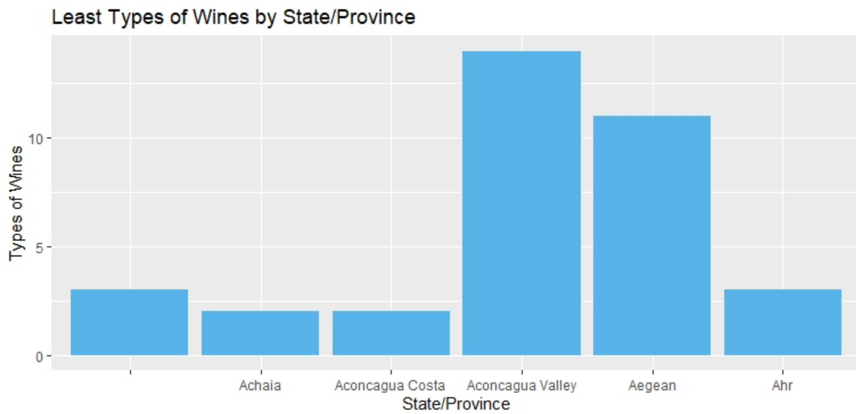
```
#Using Tidyverse, I aggregated the data to find the countries with the most
and least varieties of wines
library(tidyverse)

df1 <- cleanWine %>% group_by(country) %>% count(variety)
df2 <- df1 %>% count(country)
df6 <- df2 %>% arrange(desc(n))#Most Varieties Country
df2 %>% arrange(n) #Least Varieties Country

#Also using Tidyverse, I aggregated the data to find the provinces or states
with the most and least varieties of wines
df3 <- cleanWine %>% group_by(province) %>% count(variety)
df4 <- df3 %>% count(province)
dfA <- df4 %>% arrange(desc(n))#Most Varieties Province
df4 %>% arrange(n) #Least Varieties Province

#Using ggplot, I plotted the top 6 countries and provinces/states with the
most varieties of wines
library(dplyr)
library(ggplot2)

df7 <- head(df6)
dfB <- head(dfA)
df8 <- as.data.frame(df7)
dfC <- as.data.frame(dfB)

ggplot(df8, aes(x=country, y=n)) +
    xlab("Country") + ylab("Types of Wines") +
    ggtitle("Most Types of Wines by Country") +
    geom_bar(stat = "identity", fill = “#FF6666”)
ggplot(dfC, aes(x=province, y=n)) +
    xlab("State/Province") + ylab("Types of Wines") +
    ggtitle("Most Types of Wines by State/Province") +
    geom_bar(stat = "identity", fill = “#FF6666”)
```

- **Visualization**







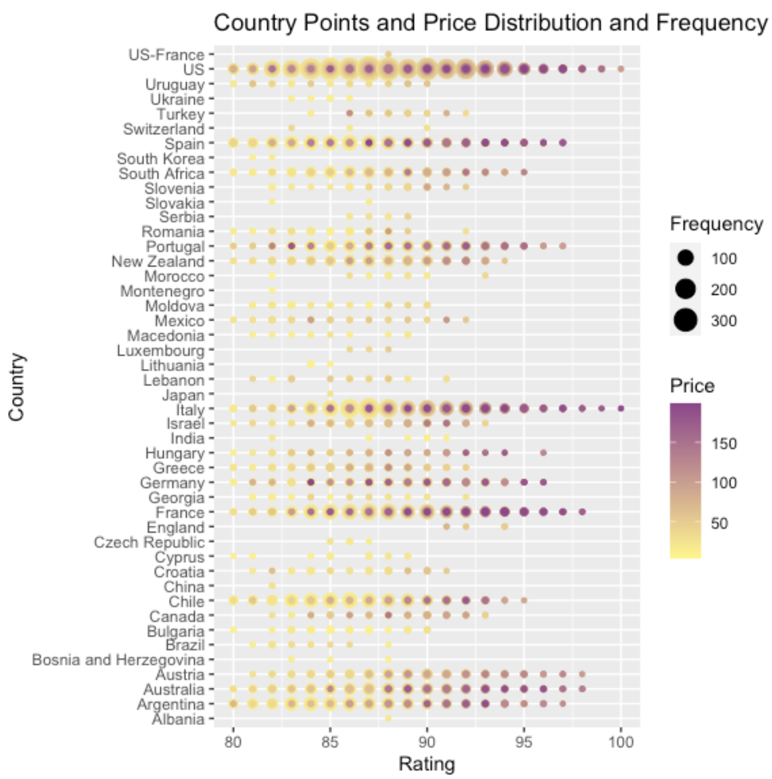


- ## Additional Analysis

```
test2 <- ppNew %>% group_by(country,points,price) %>% count()
test2 <- test2[test2$price < 200,]

plot1 <- ggplot(data = test2, aes(x=points, y=country)) +
geom_point(aes(color=price, size=n))
plot1 <- plot1 + labs(title = "Country Points and Price Distribution and
Frequency", x = "Rating", y = "Country", color = "Price", size = "Frequency")
plot1 <- plot1 + scale_colour_gradient(low = "khaki1", high = "orchid4")
```

- ## Visualization

Distribution of grouped price / points by country.



WHICH WINERIES HAVE THE HIGHEST / LOWEST MEAN POINTS SCORE?

- **Description**

To determine which wineries have the highest and lowest point scores, we take the mean points score of the wineries in our dataset. To avoid a skewed result, we narrowed our search to wineries that have more than 100 occurrences in our dataset.

- **Analysis**

```
#Which winery has the highest mean points score? Which has the lowest?
wineries = cleanWine %>%
  group_by(winery) %>%
  count()

#Looking at wineries that have more than 100 occurences
Biggest_Wineries = wineries %>%
  filter(n>100)
# Looking at points score for each winery
Best_Wineries = cleanWine %>%
  filter(winery %in% Biggest_Wineries$winery) %>%
  select(winery,points)
# Ranking mean points score per winery
Best_Wineries %>%
  group_by(winery) %>%
  summarise(Mean_Score = mean(points)) %>%
  arrange(desc(Mean_Score))%>%
  kable()
|winery                 | Mean_Score|
|:----------------------|----------:|
|Williams Selyem        |   92.28511| #Winery w/ highest mean points
|Testarossa             |   90.85965|
|Bouchard Père & Fils   |   90.68519|
|Gary Farrell           |   90.39216|
|Joseph Drouhin         |   90.37190|
|Louis Latour           |   89.70339|
|De Loach               |   88.97321|
|Chateau Ste. Michelle  |   88.40000|
|Kendall-Jackson        |   88.20000|
|Robert Mondavi         |   88.18966|
|Concha y Toro          |   87.99242|
|Wines & Winemakers     |   87.79646|
|Columbia Crest         |   87.67105|
|Trapiche               |   87.33333|
|Georges Duboeuf        |   86.91473|
|DFJ Vinhos             |   86.81081|
|Hogue                  |   86.74510|
|Cameron Hughes         |   86.73267|
|Kenwood                |   86.66087| #Winery w/ lowest mean points

#Visualizing winery vs points scatter plot
#creating a new df to pull from for our winery scatter plot
top_winery <- cleanWine %>%
  group_by(winery) %>%
  count()
#only looking at wineries with more than 100 occurrences
```

```
top_winery <- top_winery[top_winery$n > 100,]
Popular <- testData %>%
  group_by(winery, points) %>%
  count()
Popular
View(Popular)

Popular <- Popular[Popular$winery %in% top_winery$winery,]

w <- ggplot(Popular) + geom_point(aes(x=points, y=winery, color=points,
size=n, alpha=.1))
w <- w + labs(title = "Winery Points Distribution and Frequency", x =
"points", y = "winery", size = "frequency")
W
```

- **Visualization**

WHICH WINE VARIETIES HAVE THE HIGHEST / LOWEST MEAN POINTS SCORE?

- **Description**

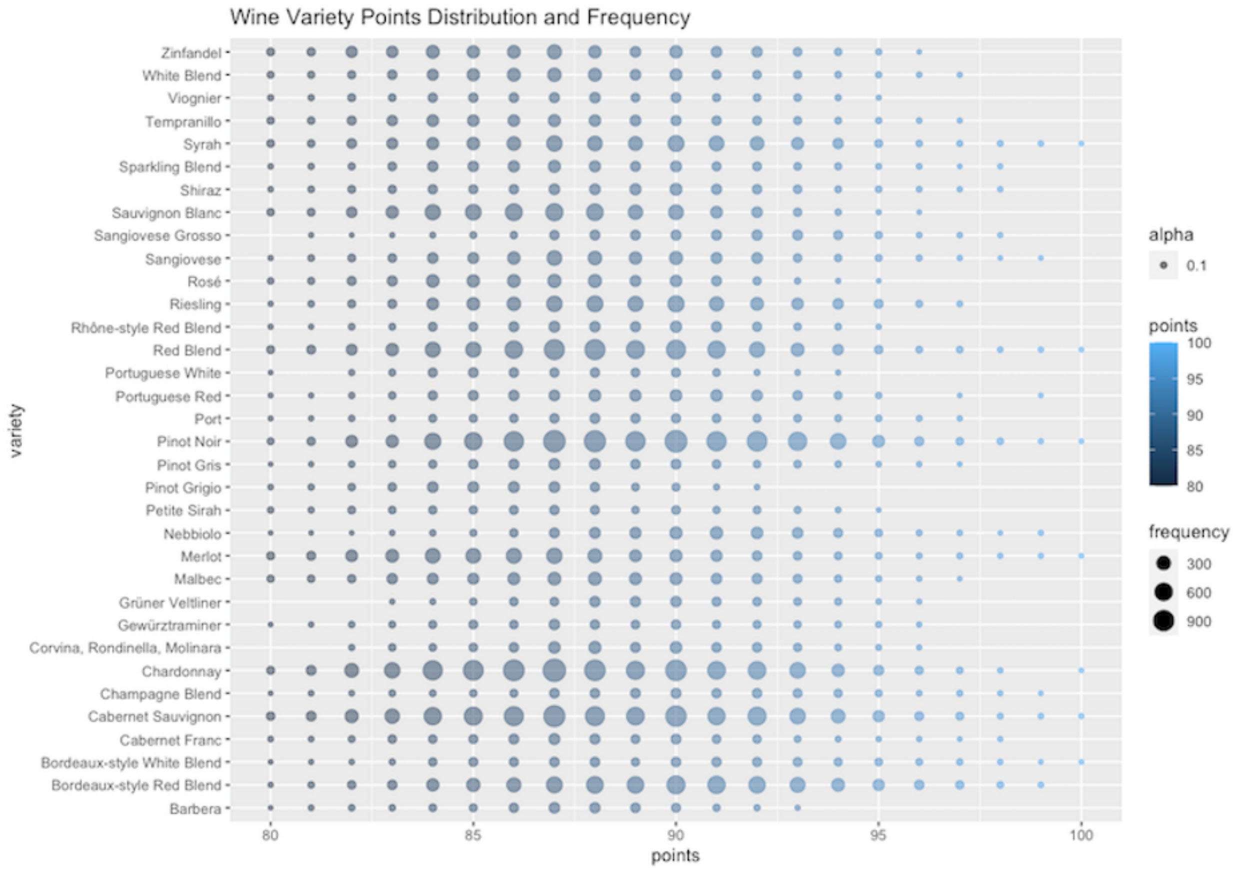
To determine which wine varieties have the highest and lowest point scores, we take the mean points score of the wine varieties in our dataset. To avoid a skewed result, we narrowed our search to varieties that have more than 500 occurrences in our dataset.

- **Analysis**

```
#grouping the varieties
varieties = cleanWine %>%
  group_by(variety) %>%
  count()

#Varieties with over 500 occurrences
Popular_Varieties = varieties %>%
  filter(n>500)

#Looking at points score for each variety
Variety_Data = cleanWine %>%
  filter(variety %in% Popular_Varieties$variety) %>%
  select(variety,points)

#Ranking mean points score per variety
MeanPointsVariety <- Variety_Data %>%
  group_by(variety) %>%
  summarise(Mean_Score = mean(points)) %>%
  arrange(desc(Mean_Score))%>%
  kable()
MeanPointsVariety
|variety                        | Mean_Score|
|:------------------------------|----------:|
|Sangiovese Grosso              |   90.32644| #Variety w/ highest mean points
|Nebbiolo                       |   90.23152|
|Champagne Blend                |   89.62515|
|Bordeaux-style Red Blend       |   89.48627|
|Bordeaux-style White Blend     |   89.35981|
|Grüner Veltliner               |   89.27413|
|Pinot Noir                     |   88.84323|
|Portuguese Red                 |   88.72014|
|Riesling                       |   88.67755|
|Corvina, Rondinella, Molinara  |   88.54204|
|Port                           |   88.53607|
|Syrah                          |   88.40809|
|Rhône-style Red Blend          |   88.26004|
|Shiraz                         |   88.25758|
|Cabernet Sauvignon             |   88.17499|
|Red Blend                      |   88.09272|
|Gewürztraminer                 |   88.06902|
|Sangiovese                     |   87.98559|
|Pinot Gris                     |   87.82536|
|Chardonnay                     |   87.76722|
|Cabernet Franc                 |   87.53001|
|Sparkling Blend                |   87.48821|
|Malbec                         |   87.46240|
|Barbera                        |   87.37638|
```

```
|Viognier                      |    87.21886|
|White Blend                   |    87.05813|
|Petite Sirah                  |    87.02397|
|Portuguese White              |    87.00484|
|Sauvignon Blanc               |    86.86660|
|Tempranillo                   |    86.75277|
|Zinfandel                     |    86.75156|
|Rosé                          |    86.65340|
|Merlot                        |    86.49497|
|Pinot Grigio                  |    85.82070| #Variety w/ lowest mean points
#Visualizing variety vs points scatter plot
#creating a new df to pull from for our variety scatter plot
top_variety <- cleanWine %>%
  group_by(variety) %>%
  count()

#only looking at varieties with more than 500 occurrences
top_variety <- top_variety[top_variety$n > 500,]
varietiesTest <- testData %>%
  group_by(variety, points) %>%
  count()
varietiesTest
View(varietiesTest)
varietiesTest <- varietiesTest[varietiesTest$variety %in%
top_variety$variety,]

#Creating the scatter plot
p <- ggplot(varietiesTest) + geom_point(aes(x=points, y=variety,
color=points, size=n, alpha=.1))
p <- p + labs(title = "Wine Variety Points Distribution and Frequency", x =
"points", y = "variety", size = "frequency")
p
```

- **Visualization**



CAN WE PREDICT THE RATING OF A WINE BASED ON SOME VARIABLES?

- **Description**

In order to predict the rating of a wine, multiple variables were considered, such as price, variety, and taster. The main predictor of wine rating was price. Variety and taster were also able to predict the rating of a wine.

- **Analysis**

To analyze the relationship of points and price, a custom data frame was created with just the price and points categories. There were 8,713 blank prices and these were removed from the data frame, since replacing these blanks with the mean price would have skewed the data too much towards the mean.

```
# Points and Price analysis

# Moving them into their own data frame for easier analysis
pp <- data.frame(cleanWine$points, cleanWine$price)

# Renaming columns
names(pp)[names(pp) == "cleanWine.points"] <- "points"
names(pp)[names(pp) == "cleanWine.price"] <- "price"

# Counting NAs
sum(is.na(pp$points))
[1] 0
sum(is.na(pp$price))
[1] 8713

# Omitting blanks, since it's too many to replace with the mean
pp <- na.omit(pp)
```

Once the data frame was cleaned, the next step was to look at the measures of central tendency and create distributions of the points and prices. The distributions helped to identify the outliers that needed to be removed in order to continue preparing the data for a prediction model.

```
# Setting the values as numbers to run statistics
as.numeric(pp$points)
as.numeric(pp$price)

# Measures of central tendency
mean(pp$points)
[1] 87.86846
median(pp$points)
[1] 88
sd(pp$points)
[1] 3.222009
min(pp$points)
[1] 80
max(pp$points)
1] 100

mean(pp$price)
```

```
[1] 33.65857
median(pp$price)
[1] 25
sd(pp$price)
[1] 37.6679
min(pp$price)
[1] 4
max(pp$price)
> max(pp$price)
[1] 2300
```

The following libraries were installed in order to create the histogram and scatter plot diagrams displayed in the visualization section.

```
library(ggpubr)
library(RColorBrewer)
library(plyr)
library(dplyr)
library(knitr)
library(ggplot2)
```

A histogram was created to view the distribution of points. The distribution of points closely resembled a normal distribution.

```
# Fancy points histogram

pointsHist <- ggplot(pp, aes(x=points)) + geom_histogram(fill = "red4", color
= "black")
pointsHist <- pointsHist + ggtitle("Points Distribution")
pointsHist <- pointsHist+ xlab("Points")
pointsHist <- pointsHist+ ylab("Frequency")
pointsHist
```

Since outliers significantly affected the price distribution, a custom data frame was created that removed any data points with prices over $200. This custom data frame was used to create a price histogram. Even with extreme outliers of prices over $200 removed, the distribution of prices was left skewed since the majority of prices fell below $90.

```
# Create a new data frame removing all rows with prices over 200
pp2 <- pp[!rowSums(pp > 200),]

adjpriceHist <- ggplot(pp2, aes(x=price)) + geom_histogram(fill = "red4",
color = "black")
adjpriceHist <- adjpriceHist + ggtitle("Price Distribution")
adjpriceHist <- adjpriceHist+ xlab("Price")
adjpriceHist <- adjpriceHist+ ylab("Frequency")
adjpriceHist
```

After the distributions were assessed, a scatter plot was created to view points as a function of price to identify any noticeable pattern that may indicate a relationship between the two variables. The data frame with outliers removed was used for the scatter plot.

```
# Fancy scatter plot with price outliers removed
```

```
pPlot2 <- ggplot(data = pp2, aes(x=price, y=points)) + geom_point(color =
"red4") + geom_smooth(method = "lm", color = "dodgerblue2")
pPlot2 <- pPlot2 + ggtitle("Price vs Points")
pPlot2 <- pPlot2 + xlab("Price")
pPlot2 <- pPlot2 + ylab("Points")
pPlot2
```

A linear relationship was identifiable from the scatter plot. Based on this, a linear regression model was created using the lm() function.

```
# Linear model of points as a function of price
priceLM <-  lm(points ~ price, data=pp)
summary(priceLM)

Call:
lm(formula = points ~ price, data = pp)

Residuals:
    Min      1Q  Median      3Q     Max 
-75.581  -1.963  -0.040   2.080  10.932 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 8.658e+01  1.295e-02  6687.6   <2e-16 ***
price       3.826e-02  2.563e-04   149.3   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.882 on 89103 degrees of freedom
Multiple R-squared:  0.2001,	Adjusted R-squared:  0.2001 
F-statistic: 2.229e+04 on 1 and 89103 DF,  p-value: < 2.2e-16
```

The p-values are statistically significant for this model, however only 20% of the variability in points is explained by price. Considering the inconsistency in the application of wine ratings, this can be considered a good percentage for explaining wine ratings based solely on price.

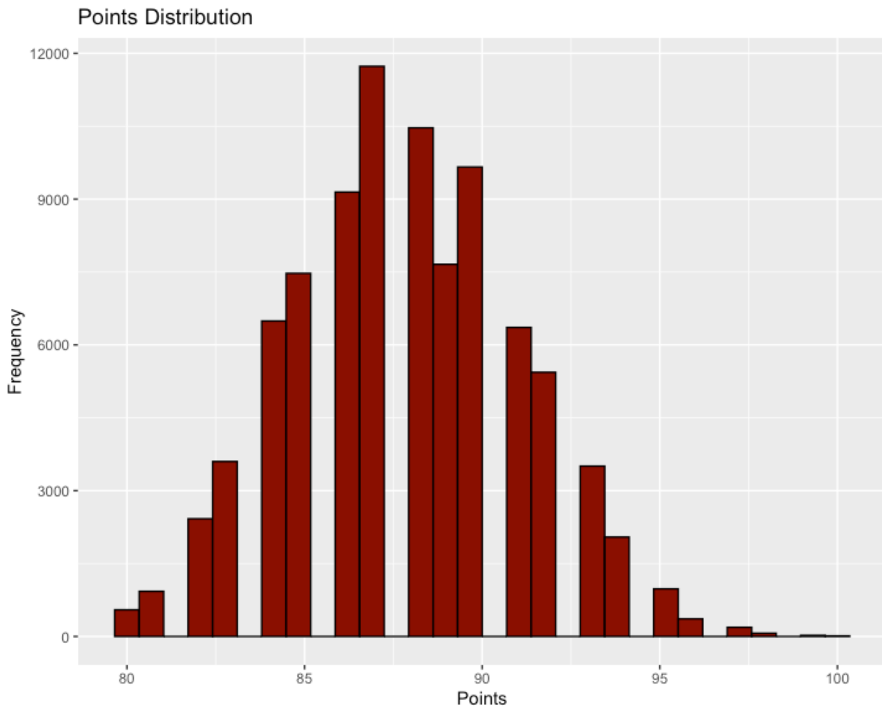
The following are some points predictions using the linear model.

```
predict(priceLM, data.frame(price = 10))
[1]
86.96326

predict(priceLM, data.frame(price = 25))
[1]
87.53717

predict(priceLM, data.frame(price = 50))
[1]
88.4937

predict(priceLM, data.frame(price = 75))
[1]
89.45022
predict(priceLM, data.frame(price = 100))
[1]
```

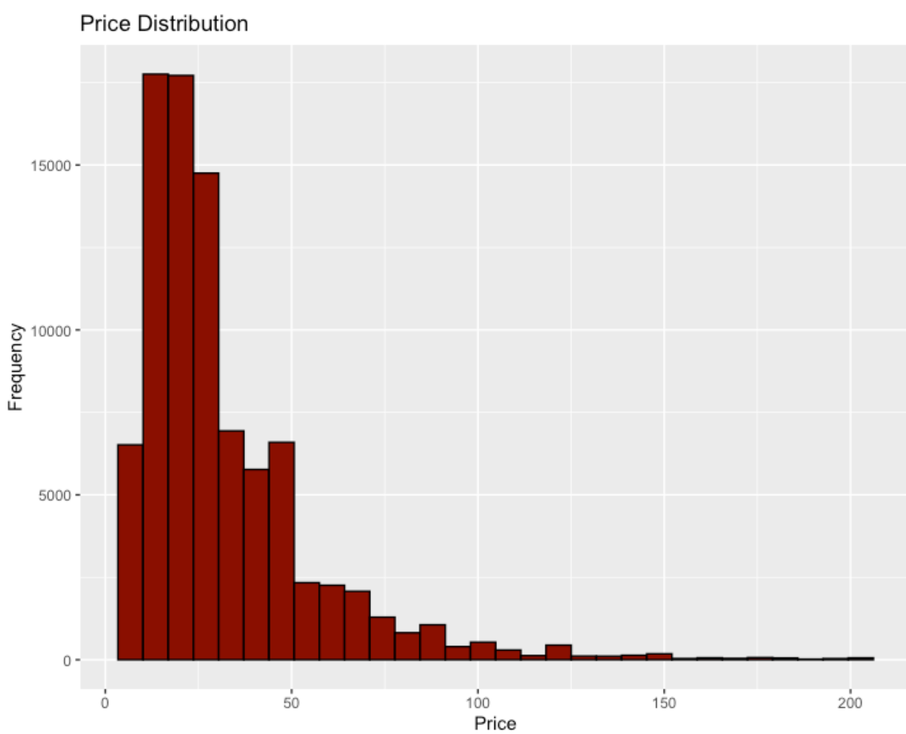
```
90.40674
```

- **Visualization**

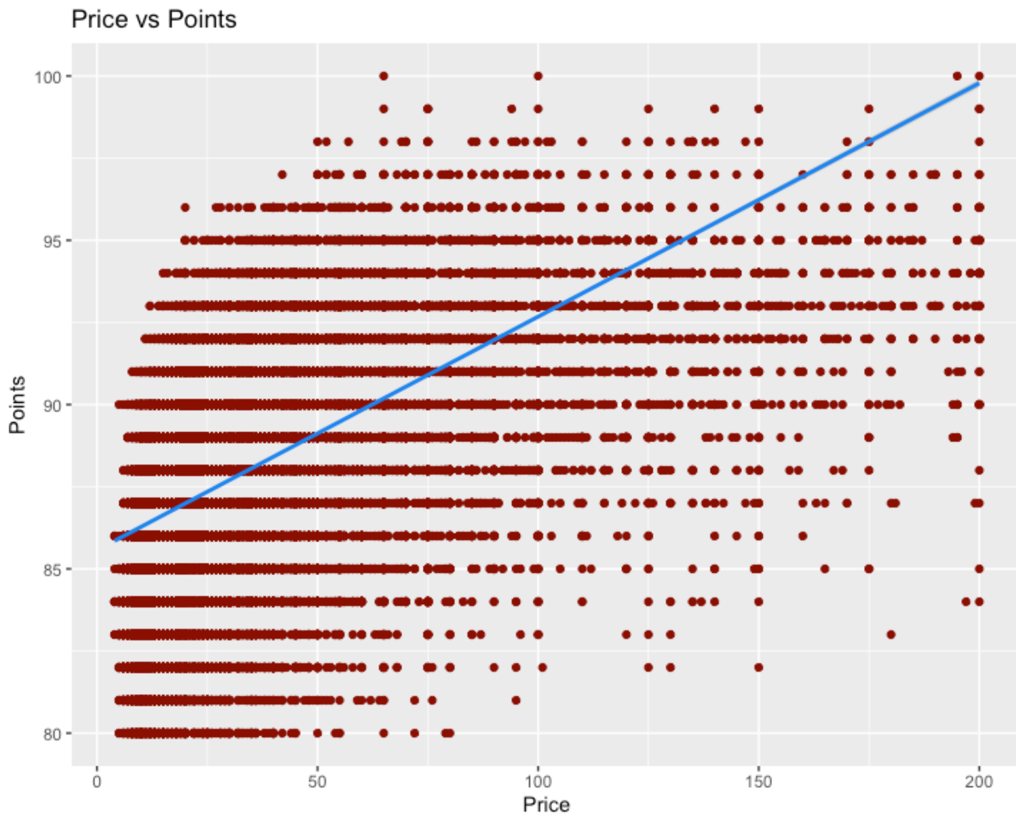
Histogram of points distribution, including all price values.



Histogram of price distribution with outliers removed (prices over $200).



Scatter plot of points as a function of price with outliers removed.



Can we predict the region / country of a wine based on the rating?

- **Description**

While we found that the most significant predictor of the points was the price, we also analyzed more complex models that used other factors in an attempt to improve our prediction accuracy.

While analyzing the significance of Country, Region, State/Province, we found that these were not strong predictors of points, and reduced our adjusted r2. We also found that the Winery and the Designation had too many unique values, and was prone to overfit our models.

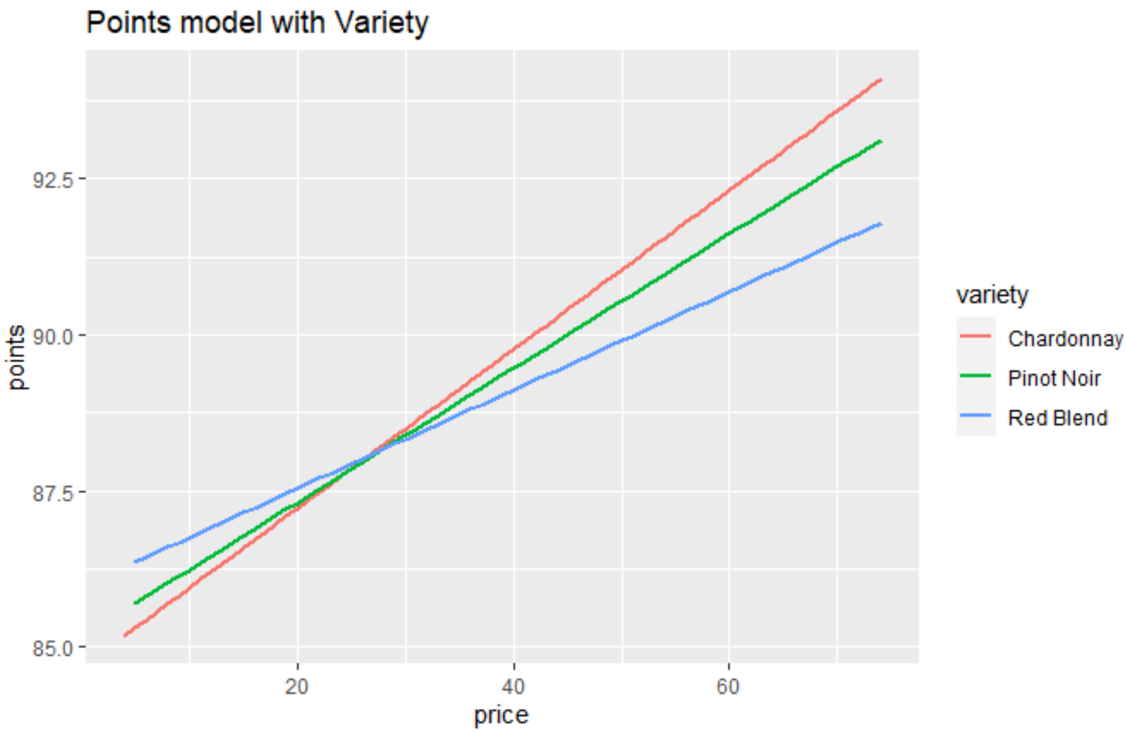
Below is the best model found, which uses the Price, Taster, and Variety. By using the Taster and Variety as factors in a linear model, R partitions the model to contain separate coefficients for Price for each combination of Taster/Variety. While this increases model complexity, it also doubled our r2 from using Price alone.

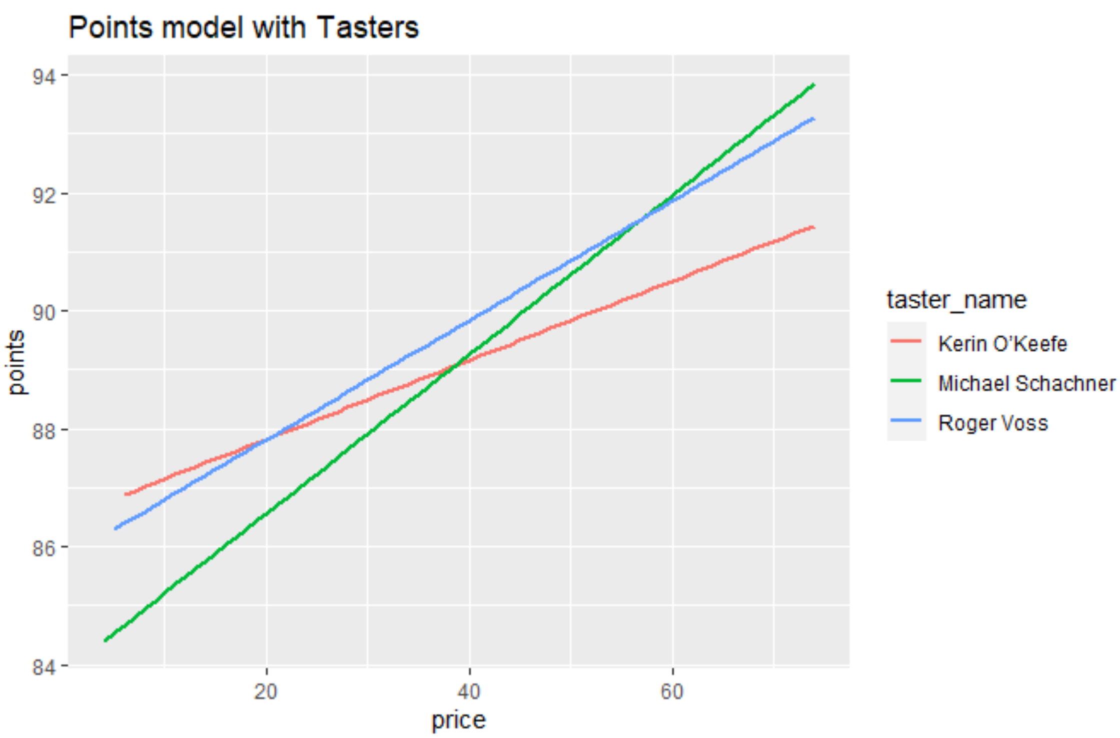
- **Analysis**

R code and notes.

```
#========================================================
# Regression Model using points, taster_name, and variety
#========================================================

library(sqldf)
library(ggplot2)
library(broom)

#For simple visual, take a subset of the tasters
#This will demonstrate the different coefficents for Price

taster_df <- sqldf("SELECT taster_name, count(*) as qty from df
group by taster_name order by count(*) desc")
df <- df[df$taster_name %in% taster_df$taster_name[0:5],]

wine_model <- lm(points ~ price +
factor(taster_name) +
factor(variety) ,
data=df)
ggplot(augment(cross_model), aes(x=price, y=points, color=taster_name)) +
        geom_line(aes(y = .fitted), size=1)
```

- **Visualization**





○ Conclusions

Summary of Results

Wine raters are definitely biased!

California had the most varieties of wine (at 200) and the United States was the country with the most varieties.

California wineries hold the 1^{st}, 2^{nd} and 4^{th} highest mean ratings. France holds the 3^{rd} and 5^{th} highest.

Red wines hold the 1^{st}, 2^{nd} and 4^{th} highest mean ratings. White wines hold the 3^{rd} and 5^{th} highest.

The lowest mean ratings included an equal amount of white, red, and blush wine varieties.

The United States, Spain, Portugal, Italy, France, Chile, Australia, and Argentina have the highest numbers of highly priced and rated wines.

Prediction models can accurately predict the rating of a new wine using multiple variables.

The top 3 words used to describe wine were "aromas", "fruit", and "fresh". The types of fruit were also frequently mentioned.

Lessons Learned

Sentiment analysis may have been more useful than text mining with regards to the wine descriptions.

We had issues with R not allowing us to remove specific values, blanks, or rows from the dataset. Multiple different methods were used to try and removed these values / blanks / rows, but nothing worked.

We had difficulty formatting axes using the ggplot package. Multiple team members had issues when trying to rotate the axis labels. Many methods were tried, including multiple packages, but nothing worked to rotate the text. The "fix" was to swap the x and y axes so that the text was readable. We do not know why the text rotation did not work.

A dataset with more numerical variables would have been much easier to analyze and use for prediction. The only numbers in this dataset were price and points, which made some aspects of correlation difficult.

A more balanced dataset which included more data from other countries would have helped mitigate biases in the dataset and make a more robust prediction model.