Group M
Matt Thomas, Kevin Finity, Kevin Lennon, Jae Hyun Lee

Introduction

For our analysis we selected the Wine Reviews dataset (www.kaggle.com/zynicide/wine-reviews). This is a compilation of reviews from Wine Enthusiast magazine scraped from the web. The csv file contains about 130,000 reviews of wines from all over the world. The data comes with the following columns: country, province, wine region (two columns if the wine belongs to a sub-appellation), price, point rating (out of 100), taster name, taster twitter handle, description, designation (vineyard name), variety, and winery. Our analysis is divided up into two main parts. First, some general data analysis revolving around the price and points (the only numeric columns), and second analysis of the text used in descriptions. In the first part we wanted to learn what the various designations, such as country of origin or grape variety, could tell us about how a wine is priced or how high it's rated (or both), and especially how the tasters differed in their awarding of points. In the second part, we wanted to learn what various keywords and descriptors of the wine could tell us about how high a wine is rated, where it comes from, or any other insight we could glean.

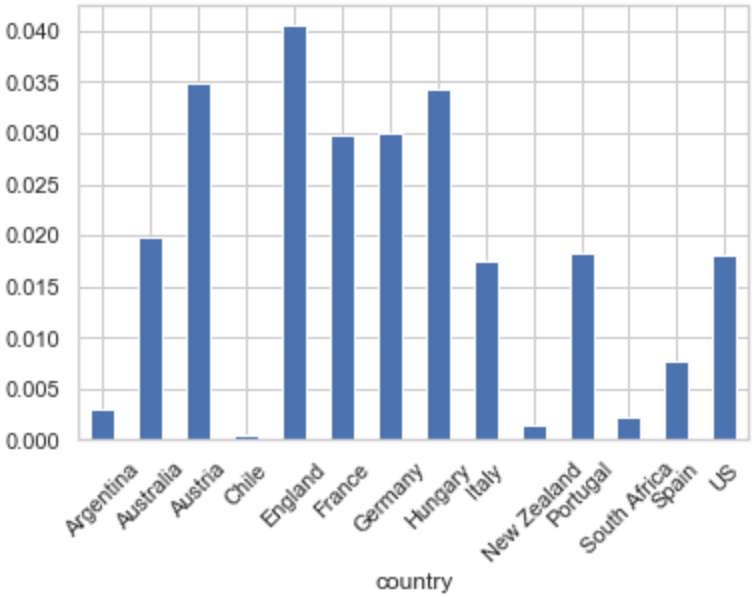
The Data

We got our dataset from Kaggle and we chose it not only because we were interested in the subject, but because with its limited numeric columns we knew we would have to get creative with the analysis and not just rely on statistics. The data doesn't require much cleaning, but there was a redundant column that needed to be deleted (an extra numeric index) and one of the taster names had extra strings in it. For some subsets, rows with missing pricing had to be deleted because otherwise it was not possible to plot a points/price comparison.

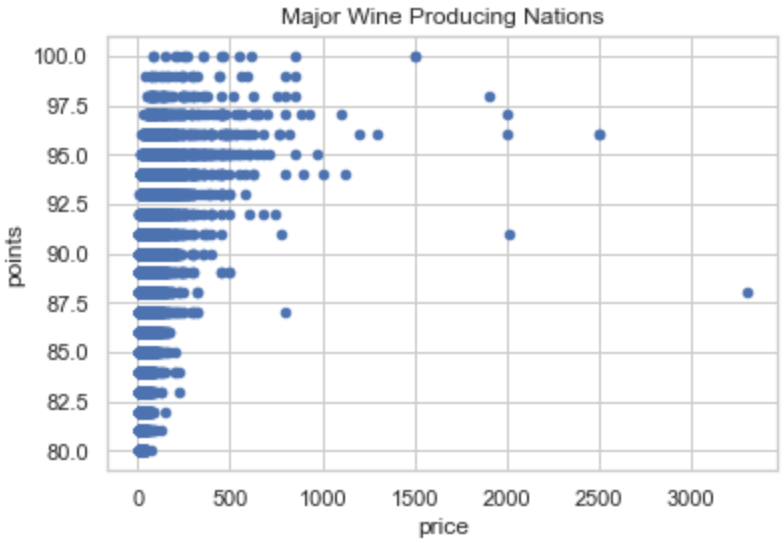
Uses of the data

Region Exploration

There are many ways to query this dataset so to start we chose a few that interested us. We created a new category called "Quality" that binned the wines into four categories based on rating: okay (80-84), good (85-89), excellent (90-94), and superlative (95-100). We wanted to see which countries had to most superlative wines:

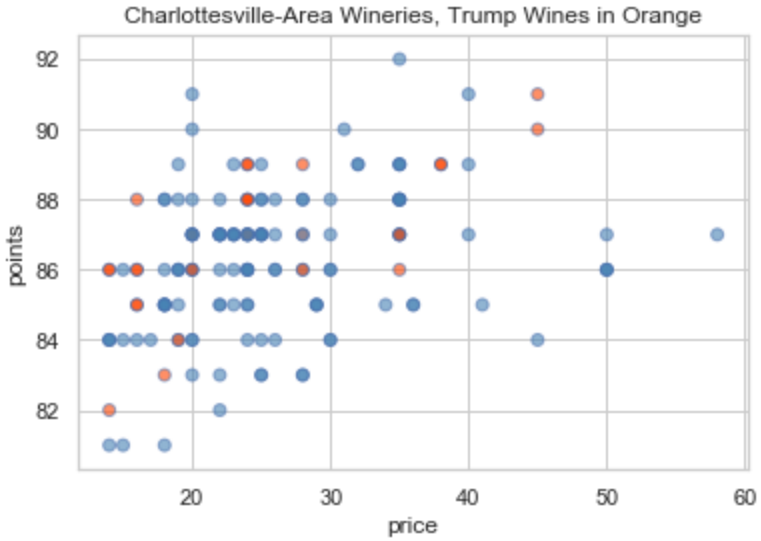


The winner is England, but England produces very little wine so we looked only at the major wine producing countries: US, France, Italy, Spain, Australia, South Africa, Chile, and Argentina. These choices are somewhat arbitrary but they make up the majority of what you would find in any wine store. We made a boxplot showing the distribution of reviews (see Jupyter notebook, this graph is interactive) and it looked like the reviewers slightly favor France. We also did several group-bys to find the highest rated wine per country, as well as grouping them into “old world” or “new world” countries to compare statistics. Although many of the stats were close, we found that old world wine prices were much more variable: a standard deviation of 56 versus 29 for new world. We also plotted price versus rating:



We wanted to find out what those outliers were, and all but two are from the Bordeaux region of France, which is not too surprising because those wines can be expensive for the quality.

We took a closer look at Virginia wines since that's where most of us live, to look at stats by sub-region and the highest rated wines. The various sub-regions rated similarly. We analyzed the wines of the Monticello region because it's probably the most important single region in Virginia and it's also very close to UVa. There are several 90+ point wines that we found in this area, and we also did a comparison of all wines in Monticello versus Trump wines:



Yes, the President owns a winery near UVa for some reason. We also created an interactive version of the scatterplot above so that you can hover over any point to see the winery name, variety, price and points (see jupyter notebook).

Taster Insights

We also tried to characterize the individual tasters in the dataset. We started by calculating a few aggregate values and adding them to the dataframe. A "num_reviews" field groups by taster and counts the number of reviews that each taster has done. A "top_country" field groups by both taster and country, calculating the country which the taster has written the most reviews for. And a "fav_country" field looks at the average scores per taster name and country, and picks the country which each taster has given the highest average score to.

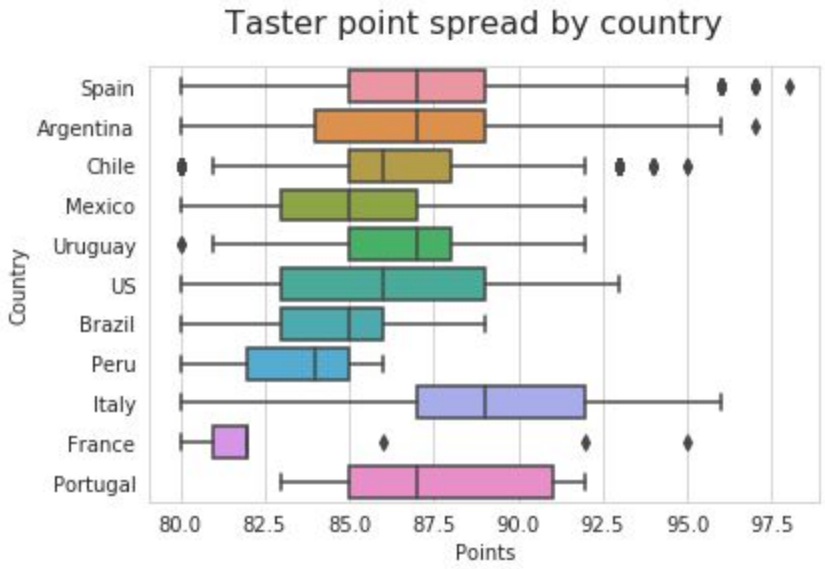
Using these aggregate values, we wrote an interactive "taster_profile()" function to display some basic stats about a selected taster. Two additional helper functions were written to produce a list of tasters ("get_taster_list()") and to interactively let the user select a taster from the list ("choose_taster()").

Example taster_profile() output

Taster Profile Name: Michael Schachner

15134 reviews

Most-reviewed country: Spain
Favorite country: Italy

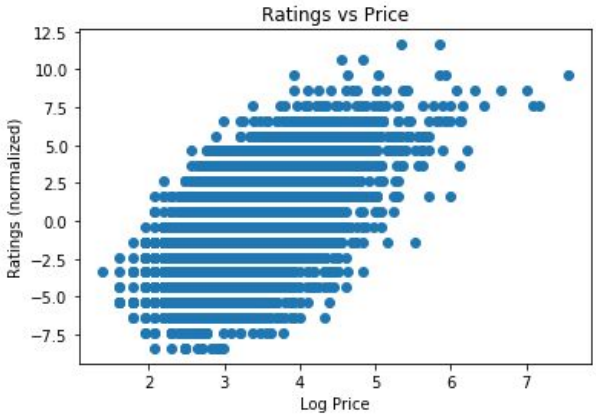


Top varieties for this taster:

Variety	# Reviews	Average Score	Score Std. Dev.
Malbec	1652	87.417070	3.141296
Red Blend	1496	88.397059	3.007254
Tempranillo	1439	87.492008	3.151005
Cabernet Sauvignon	1358	86.462445	2.728380
Chardonnay	877	84.990878	2.330052

Wine Value Tool

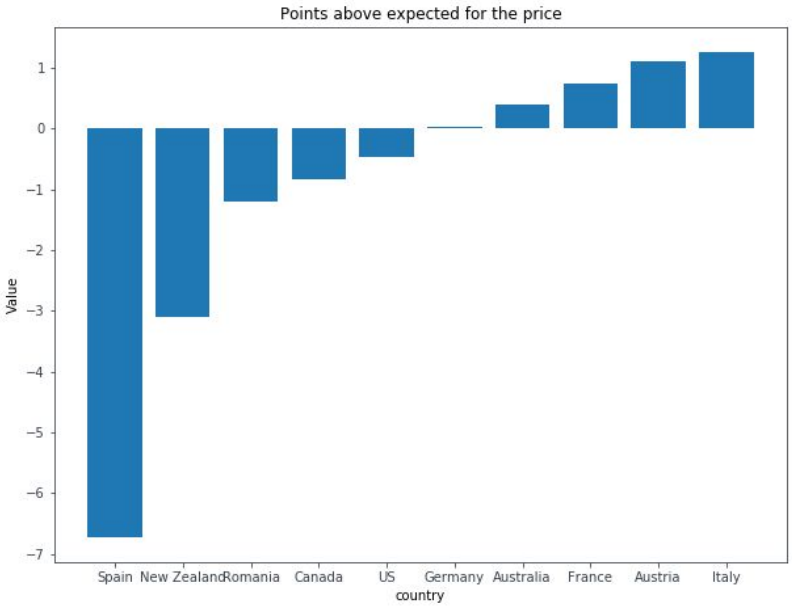
We created a tool to help people who don't know much about wine to help them find good value wine for the money. First, we had to define what a good value wine was. In order to do this, we normalized the ratings since they were on a scale of 80-100 by subtracting the mean and dividing the standard deviation. This transformed the wines to a more natural scale, with positive values being above average wine and negative values indicating below average wine. From there, we used the log of price to predict the quality of the wine with simple linear regression.

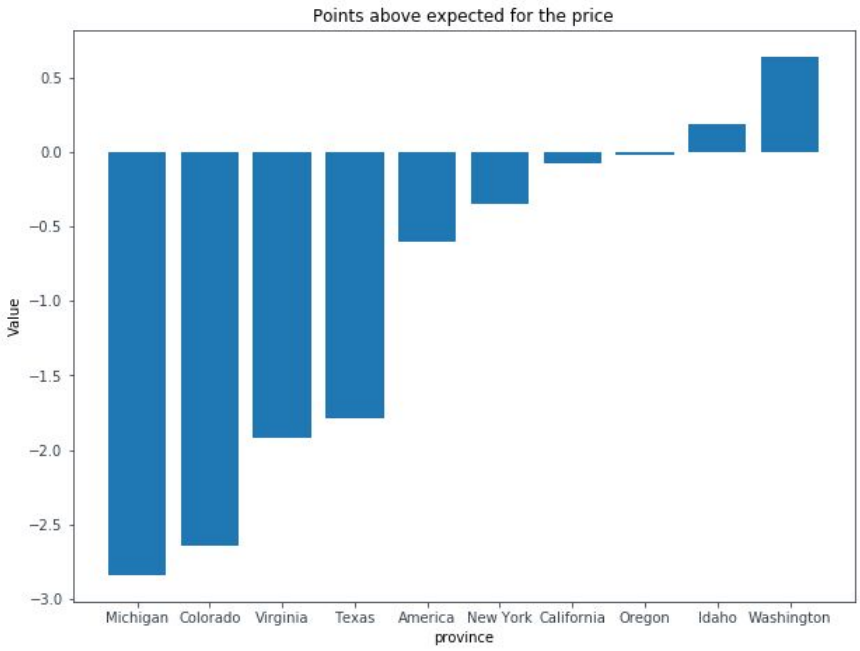


We used the log of price since this transformation created a linear relationship between price and ratings. From there, the residual errors of the model were treated as the “Value” variable. We would expect a 60 dollar wine to have a normalized rating of 0.26175 (roughly an 88.7 rating), but if it actually had a rating of 3.26, it would have a value of +3.00.

Once we had a measure for value relative to price, we then created a dynamic model that would predict based on a selected factor. If someone wanted to see what the best value wines were by country or variety, the tool would show you which types on average were better value.

While most of these factors were readily available in the model, the descriptions were in the form of text and needed a bit of processing. We used the natural language toolkit to remove common words unrelated from wine from the dataset, and added our own list of common words in wine reviews to remove as well. From there we used one-hot-encoding to create a matrix of the most common descriptors in wine and used binary values to indicate which words were contained in the review. We then used multiple regression to determine which factors were most predictive of good value wine.





The tool also allows you to drill down within a factor and find value within a specified field. For example, if you're going to a dinner party and you know that your friend enjoys a nice Sauvignon blanc, you can use this tool to filter by country which will show you the top value countries.

From here you can see that New Zealand has the best value for this type of wine. From there, you can filter on New Zealand and search by the description of the wine and see that wines described as silky, long, and creamy are all indicate especially good value for New Zealand Sauvignon blancs.

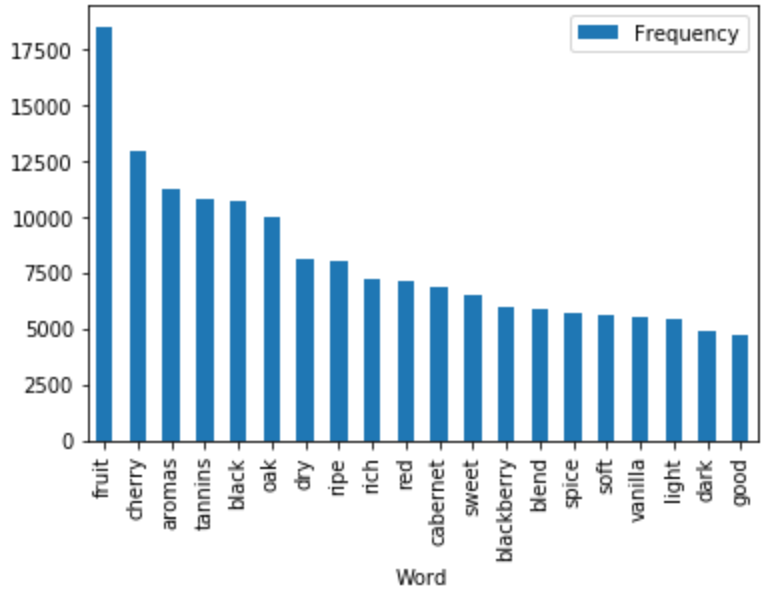
This program can help the consumer make an educated guess on what type of wine to get without having to look up the reviews for every individual bottle of wine that looks appealing like you have to using other wine apps (Vivino, Delectable, and many others). It lets you find value indicators for given filters and dynamically calculates the new indicators. The dynamic calculations are important since certain indicators may be positive for one set of filters, but negative for other sets of data. Portugal, for example, may be a strong indicator of value for a red wine, but indicates poor value for white wines so it is very important to recalculate the importance for the indicators with every filter.

The only downside to this program is the way it was trained. Because the dataset had a limited number of reviewers, it could have bias built in since they each may be biased towards a specific type of wine. Additionally, the descriptions are not the descriptions displayed by the winemaker, so there may be bias in the word choice as well. Using this model on a more diverse set of reviews and with the winemakers' descriptions would help reduce potential biases and create a better model.

Text Analysis

The text analysis focuses on the most frequently appearing words in the descriptions of the wine review overall, by country, and by quality. The countries studied is top 8 wine producing countries (US, France, Italy, Spain, Australia, South Africa, Argentina, and Chile). The quality was divided into three groups based on the points – Good (80-84), Better (85-89), and Best

(90+). Overall, the most frequent word used to describe wines was “aromas,” followed by “tannins” and “cherry.” By country, “aromas” was the top word for five of the eight countries and almost all countries shared a word such as “ripe”, “red”, “rich”, and “dry”. By quality, it is interesting to note that the descriptive words change from “light” and “soft” to “tannins” and “ripe” to “rich” and “ripe” as the quality (point) increases. The following shows how frequent the most common words were used:



See Jupyter Notebook for the full breakdown.

Beyond The Original Specifications

We went beyond the original specifications in a number of ways. We have quite a few interactive parts. In part 1, we have two interactive functions. The first one asks the user for a country, then a grape variety, and returns a dataframe with all of the wines rated 90 or above for that country and variety. The second asks the user for a price range, a variety (the user can type ‘none’ if none), and wine region. It returns an interactive plot showing all wines with the specifications by price and rating, and the user only has to hover over the plot to see the wine names. This can be used to find highly rated wines to a user’s preferences. It works on sub-regions too, e.g. you can search under Sonoma, or you can search by Russian River Valley, which is a sub-region within Sonoma.

In part 2, the taster profile demonstrated earlier is also interactive. You can specify which taster you want to see and that taster’s statistics will come up.

In part 3, we made a program that is dynamic and interactive program to help users find the best value wine for any given factors. It will recalculate the important factors for each new filter added to the dataset since the indicators will change depending on the filters. It makes use of machine learning, natural language processing, and user queries. The user can use any combination of factors which makes this a complete self serve insights tool.

As far as advanced queries, that part is subjective but both parts 3 and 4 use text analysis to query specific words used as descriptors in the dataframe.

Unit Testing

Our unit testing revolves mainly around the functions that we wrote, as the regular pandas queries themselves aren't testable. In the Jupyter notebook you will find unit tests for the functions, including for the two interactive functions in part 1. The test asks the user to input a certain answer, then tests whether the dataframe query and the user's answer matches.

In part 2, we test "get_taster_list()", since the list itself is hard to verify, we just test that the returned list has >0 items. To test "choose_taster", we created one test for the interactive mode (using the default blank argument), and one with a provided argument (of "Roger Voss", one of the tasters). Both tests verify the function output against the expected dataframe. We did not write a unit test for "taster_profile()", since it only prints to standard out, and doesn't return any output.

In part 3, we tested that the dataframe was initially manipulated correctly. We tested the "calc value" function since it is the basis of the whole analysis. We created one test which tested that both the value column was created, and it has a mean of roughly 0. We then tested to make sure the ratings were normalized through the calc value function as well . The ratings would have a mean of 0 after we normalized them, so we also tested that they had a mean of roughly 0.

Testing output Example:
(TESTING) Please type 'Riesling' and 'France'
Enter a grape variety (capitalize first letter):Riesling
Enter a country:France

/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:6: PerformanceWarning:

indexing past lexsort depth may impact performance.

/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:5: PerformanceWarning:

indexing past lexsort depth may impact performance.

```
.
(TESTING) Please enter '0', then '100', then 'Malbec', then 'Cafayate'
Enter your minimum price (0 if none). Do not include dollar sign:0
Enter your maximum price:100
Enter a grape variety (capitalize first letter), or 'none' if none:Malbec
Enter a wine region (e.g. Napa)Cafayate

.
----------------------------------------------------------------------
Ran 5 tests in 21.127s

OK
```

For text analysis, we test two functions to find the total frequency, “get_frequency()”, and mean, “get_average()”, of a particular word the user would like to look up in one of the three groups. Both tests proceed with provided statements to check whether the function correctly returns a numeric value rounded to two decimal points. Both tests yielded an expected output.

Conclusion

Our findings aren’t easy to summarize as they include many different queries on the dataframe, but mostly we looked at comparison between the major wine producing countries, Virginia wines, and general queries in part 1. In part 2 we looked at how the taster affects ratings and preferences. In part 3, we found many different insights about what predicts good value wine. One of the main insights was that Washington is actually the best province in the world for wine value. We also found that the best value winery in the world is the famous “Chateau Saint Michelle” winery located in Washington. These insights were both a pleasant surprise to the group member who worked on this section and is also from Washington. In part 4, we used text analysis to find the most frequent wine descriptors and saw how they related to different countries and rating levels.

In general, the best use of our analysis is that it helps users make an informed choice about wine. By querying the data by preference, region, or other factors, our functions allow you to find any wine that’s been reviewed based on different criteria, and our text analysis shows you what descriptors or other factors will likely get you the best quality wine.

One interesting thing that we were not able to get to would be to look at vintage. Vintage was not a separate variable but most of the wines had it listed in the description or the title. It would be possible to use a regular expression to find the vintages, although the hard part would be to match them up to the correct wine because not every wine has a vintage. If you were to make

the vintage into a separate column, it would be possible to see how that affects the rest of the analysis or compare the same wine across different vintages.

A second interesting exploration we could carry out in the future is working on a more diverse data set that contains reviews from more wine reviewers. The current set of wine reviewers may have their own individual preferences for wine and biases, so working with a dataset from a larger variety of wine reviewers could help avoid any biases in the dataset. Perhaps a scraping project from another wine magazine, like Wine Advocate, would make the functions we wrote even more useful.