TASK

Exploratory Data Analysis on the Wine Dataset

Visit our website

Introduction

Summary of the data set

The dataset is read into a dataframe, wine_df using the .read_csv method. The initial encoding used is the normal default of UTF-8. A first look of the dataset is gained through using the .head() method. The .columns method gives the column head names.

The data is a set of wine reviews with ratings given, points, prices etc. Areas given are Province, Country, Region_1, Region_2. A summary of the dataset is shown below:

Country - [String] [Categorical] Country of origin

Description - [String] Free text description of the wine

Designation - [String] This is the vineyard where the grapes are grown

Points - [Integer][Continuous] Points awarded to the wine

Price - [Float][Continuous] Price of the wine

Province - [String][Categorical] This province the wine came from

Region_1 - [String][Categorical] This is the region

Region_2 - [String][Categorical] This is the wider region

Variety - [String][Categorical] Grape variety

Winery - [String][Categorical] This is the winery where the wine is made

The countries we have are:

```
['US', 'Spain', 'France', 'Italy', 'New Zealand', 'Bulgaria','Argentina',
'Australia', 'Portugal', 'Israel', 'South Africa','Greece', 'Chile',
'Morocco', 'Romania', 'Germany', 'Canada','Moldova', 'Hungary', 'Austria',
'Croatia', 'Slovenia'],
```

The varieties we have are:

'Cabernet Sauvignon', 'Tinta de Toro', 'Sauvignon Blanc',

```
'Pinot Noir', 'Provence red blend', 'Friulano', 'Tannat',
'Chardonnay', 'Tempranillo', 'Malbec', 'Rosé', 'Tempranillo Blend',
'Syrah', 'Mavrud', 'Sangiovese', 'Sparkling Blend',
'Rhône-style White Blend', 'Red Blend', 'Mencía', 'Palomino',
'Petite Sirah', 'Riesling', 'Cabernet Sauvignon-Syrah',
'Portuguese Red', 'Nebbiolo', 'Pinot Gris', 'Meritage', 'Baga',
'Glera', 'Malbec-Merlot', 'Merlot-Malbec', 'Ugni Blanc-Colombard',
'Viognier', 'Cabernet Sauvignon-Cabernet Franc', 'Moscato',
'Pinot Grigio', 'Cabernet Franc', 'White Blend', 'Monastrell',
'Gamay', 'Zinfandel', 'Greco', 'Barbera', 'Grenache',
'Rhône-style Red Blend', 'Albariño', 'Malvasia Bianca',
'Assyrtiko', 'Malagouzia', 'Carmenère', 'Bordeaux-style Red Blend',
'Touriga Nacional', 'Agiorgitiko', 'Picpoul', 'Godello',
```

```
 'Gewürztraminer', 'Merlot', 'Syrah-Grenache', 'G-S-M', 'Mourvèdre',
 'Bordeaux-style White Blend', 'Petit Verdot', 'Muscat',
 'Chenin Blanc-Chardonnay', 'Cabernet Sauvignon-Merlot',
 'Pinot Bianco', 'Alvarinho', 'Portuguese White', 'Garganega',
 'Sauvignon', 'Gros and Petit Manseng', 'Tannat-Cabernet',
 'Alicante Bouschet', 'Aragonês', 'Silvaner', 'Ugni Blanc',
 'Grüner Veltliner', 'Frappato', 'Lemberger', 'Sylvaner',
 'Chasselas', 'Alsace white blend', 'Früburgunder', 'Kekfrankos',
 'Vermentino', 'Sherry', 'Aglianico', 'Torrontés', 'Primitivo',
 'Semillon-Sauvignon Blanc', 'Portuguese Rosé', 'Grenache-Syrah',
 'Prié Blanc', 'Negrette', 'Furmint', 'Carignane', 'Pinot Blanc',
"Nero d'Avola", 'St. Laurent', 'Blauburgunder', 'Blaufränkisch',
 'Scheurebe', 'Ribolla Gialla', 'Charbono',
 'Malbec-Cabernet Sauvignon', 'Pinot Noir-Gamay', 'Pinot Nero',
 'Gros Manseng', 'Nerello Mascalese', 'Shiraz', 'Negroamaro',
 'Champagne Blend', 'Romorantin', 'Syrah-Cabernet Sauvignon',
 'Tannat-Merlot', 'Duras', 'Garnacha', 'Tinta Francisca',
 'Portuguese Sparkling', 'Chenin Blanc', 'Turbiana',
```

The Provinces we have are:

```
(['California', 'Northern Spain', 'Oregon', 'Provence',
 'Northeastern Italy', 'Southwest France', 'Kumeu', 'Washington',
 'Bulgaria', 'Tuscany', 'France Other', 'Rh√¥ne Valley', 'Galicia',
 'Andalucia', 'Idaho', 'Burgundy', 'Loire Valley', 'New York',
 'Mendoza Province', 'Victoria', 'Alentejano', 'Piedmont',
 'Alentejo', 'Champagne', 'Upper Galilee', 'Beira Atlantico',
 'Veneto', 'Douro', 'Tejo', 'Stellenbosch', 'Levante',
 'Sicily & Sardinia', 'Southern Italy', 'Languedoc-Roussillon',
 'Bordeaux', 'Atalanti Valley', 'Catalonia', 'Santorini', 'Florina',
 'Marchigue', 'Colchagua Valley', 'Curic√≥ Valley', 'Nemea',
 'Maule Valley', 'Alsace', 'Guerrouane', 'Colinele Dobrogei',
 'Central Spain', 'Vinho Verde', 'Mosel', 'Rheinhessen',
 'Golan Heights', 'W√ºrttemberg', 'Ahr', 'British Columbia',
 'Moldova', 'Spain Other', 'Sopron', 'Other', 'Walker Bay', 'D√£o',
 'Italy Other', 'Duriense', 'Ontario', 'Beiras', 'Tokaji', 'Lisboa',
 'Thermenregion', 'Burgenland', 'Carnuntum', 'Rheingau', 'Nahe',
 'South Australia', 'North Dalmatia', 'Thracian Valley',
 'Goriska Brda', 'Western Cape', 'Overberg', 'Robertson', 'Galilee',
 'Maipo Valley', 'Casablanca Valley', 'Cachapoal Valley',
 'Terras do D√£o', 'Leyda Valley', 'Peumo', 'Baden',
 'Limar√≠ Valley', 'Lombardy', 'Peloponnese', 'Judean Hills',
 'Tasmania', 'Bairrada', 'Simonsberg-Paarl',
 'Portuguese Table Wine', 'Beaujolais', 'Pen√≠nsula de Set√∫bal',
 'Aconcagua Valley', 'Virginia', 'Dealurile Munteniei'],
dtype=object)
```

Some of this information can be gain by using the .info() method which shows the data types eg integer, float, string and the numbers of non-null fields.

We also have strange characters in Description, Designation and Winery which highlights an issue with the encoding settings. We have missing values in Designation, Price, Region_1 and Region_2.

Using .drop_duplicates() removes any duplicates, but none show here.

Using the .describe() method shows general statistical figures on the numerical data (integers, floats, doubles):

	Points	Price
count	1103.000000	1046.000000
mean	89.701723	40.242830
std	2.390405	32.588141
min	85.000000	7.000000
25%	88.000000	20.000000
50%	90.000000	31.000000
75%	91.000000	50.000000
max	96.000000	500.000000

From the table above we can easily see that there's 1103 non-empty rows in points and 1046 non-empty rows in price. The table also shows the mean, standard deviation, minimum and maximum values and percentiles for 25%, 50% and 75% of the data.

From this information we can see, for example, that 25% of the points are below or equal to 88. 50% of the prices are below or equal to £31. The maximum price paid for a bottle is $500 and the minimum is $7, the average is $40.24 and the 50% percentile, or median, is $31. The standard deviation is $32.58. The standard deviation measures the spread of the data so the wine price standard deviation of $32.58 means that most of the wine differ from the mean ($40.24) by $32.58. The standard deviation of the points is only 2.39. So there is a much smaller points dispersion than in wine price. The minimum points given is, the maximum is 95 and the mean is 89.70. This is assuming the price of wine and the points follow a normal distribution.

DATA CLEANING

SUMMARY OF THE METHODS AND VISUALISATIONS DONE DURING DATA CLEANING

Correcting the strange characters in Winery

Although we were encoding into utf-8 when we are reading in the dataset, the alien characters weren't being picked up correctly. The solution I found, after trying all the encoding settings, was to take the data, covert it to a string then encode it to MacRoman and then decode it to utf-8. This was done in a for loop to change each entry and appending them to a new list, winery_list. The winery_list was then copied over to the dataframe.

```
winery_list = []
for i in wine_df['winery']:
    i = str(i).encode('MacRoman').decode('utf-8')
    winery_list.append(i)



winery_clean_df = pd.DataFrame (winery_list, columns = ['winery_list'])
wine_df['winery'] = winery_clean_df.copy()
```


```
winery_clean_df = pd.DataFrame (winery_list, columns = ['winery_list'])
wine_df['winery'] = winery_clean_df.copy()
```


The same method is repeated for the description and variety columns which had the same problem.

There was no other cleansing required to columns as the points were already integers and the price was already a float.

MISSING DATA

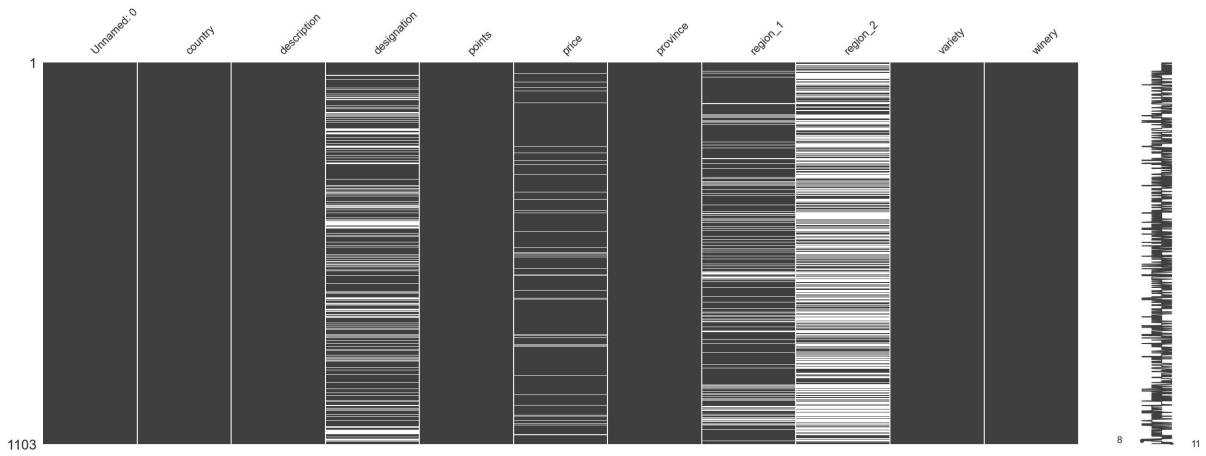
The missing data is revealed by applying, to the wine dataframe, the methods: .isnull().sum(). This shows missing values in Designation, Price, Region_1 and Region_2.

Missing Cells Numbers:

```
Unnamed: 0       0
country          0
description      0
designation    269
```

```
points           0
price           57
province         0
region_1       173
region_2       611
variety          0
winery           0
dtype: int64
```

Visually, missing fields can be shown using missingno.matrix from the missingno module:



The percentage of missing fields is 9.15%. This is calculated by getting number of cells (using the shape of the array - .shape method) and the number of cells with the np.product() function: np.product(wine_df). The empty cells are summed up using the .sum() method on the Missing Cells Table and a percentage is calculated.

Missing Values in the Designation Column

Out of 1103 rows there are 269 designations missing. This is 24% missing which is quite high.

Let's look at Argentina which has missing designations and non-missing designations. If you show the first 12 Argentinian wines, using wine_df[wine_df.country == 'Argentina'].head(12), you can see that designations can't be implied from the other columns, and so, along with the high missing percentage, it might be best to drop the designations column using the drop method.

Missing Values in the Region_1 & Region_2 Columns

Comparing the province, region_1 and region_2 for rows indexes 2 and 1098:

```
wine_df.iloc[2,[6,7,8]]
```

```
province                     California
```

```
region_1    Special Selected Late Harvest
region_2                          Sonoma
Name: 2, dtype: object
```

```
wine_df.iloc[1098,[6,7,8]]
```

```
province          California
region_1                 nan
region_2    California Other
Name: 1098, dtype: object
```

Row 2 and 1098 have California as the Province but their Region_1 and Region_2 are vastly different. Region_1 and Region_2 must be non-compulsory free text fields. Let's drop them and focus on Country and Province as Region_1 and Region_2 don't seem to shed any more light on where the wine came from. This will resolve the missing data for these columns.

Dropping these columns:

```
wine_df.drop(['region_1','region_2', 'designation','description'], axis=1, inplace=True)
```

Missing Values in the Price Column

For the price we can take the average price of the wine and put this into the blanks. The average value can be taken from the .describe table above or extracted using the mean method. First we can look at the NaNs in price by using the .isnull() method:

```
wine_df[wine_df['price'].isnull()].head()
```

As a precaution we create a temporary dataframe, and after checking the result in the temporary dataframe, we can move it into our working dataframe.

```
temp_df[temp_df['price'].isnull()]
```

Here the average price is 40.24 and we will replace the NaNs with this figure:

```
temp_df['price'] = temp_df.price.fillna(40.24)
```

We can check if this has been successful by checking row index 32 which was previously NaN:

```
temp_df.iloc[32:33]
```

DATA STORIES AND VISUALISATIONS

THIS IS THE BULK OF THIS PROJECT. EXTRACT STORIES AND ASSUMPTIONS BASED ON VISUALISATIONS OF THE DATA

Using NLTK Module's Pre-Trained Sentiment Analyzer

Checking the Description column for judge the sentiment of the text and give it a sentiment score. This would give an alternative rating to the points and ought to correspond to the points.

To do this the nltk module was imported along with its sentiment analyser.

```
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')

analyzer = SentimentIntensityAnalyzer()
```

A test sentence was run:

```
analyzer.polarity_scores("Disgusting. This wine is horrible!")
```

This was rated as:

```
{'neg': 0.706, 'neu': 0.294, 'pos': 0.0, 'compound': -0.8016}
```

The neg, pos, neu (neutral) results combined into the compound figure. We will be using this compound figure as our Sentiment_Score which we'll be incorporating into our dataframe. The test negative sentence was rated as -0.8016 which is seems to be a good indication of its negativity. The sentiment analyser is built for gauging posts on social media rather than reviews but these wine reviews are quite similar in length to social media posts.

The below code applies the sentiment analyzer into every field in the Description and creates a Sentiment column in the dataframe.

```
wine_df['sentiment'] = wine_df['description'].apply(analyzer.polarity_scores)
pd.concat([wine_df.drop(['sentiment'], axis=1), wine_df['sentiment'].apply(pd.Series)], axis=1)
```

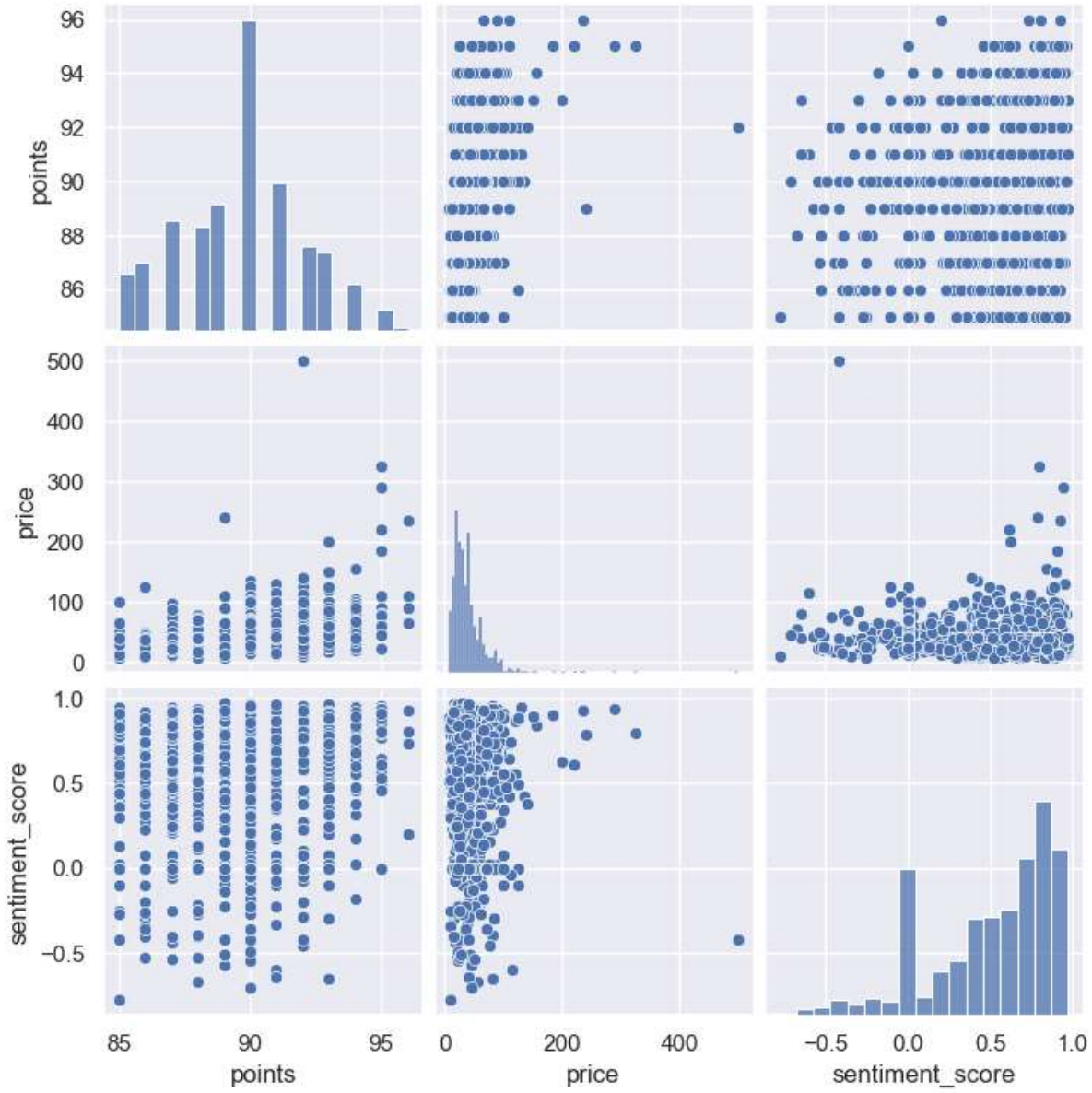
Sentiment is now a column containing a dictionary with neg, neu, pos, compound entries. We need to separate out just the compound for our sentiment score. To do this we apply pd.Series to the sentiment column and move it to a temporary dataframe. Then we drop the columns we don't need and leave only compound. Once this is done we can copy the temporary dataframe back into the main dataframe.

```
compound_temp = wine_df.sentiment.apply(pd.Series)
compound_temp.drop(columns=['neg', 'neu', 'pos'], inplace=True)
wine_df['sentiment_score'] = compound_temp
```

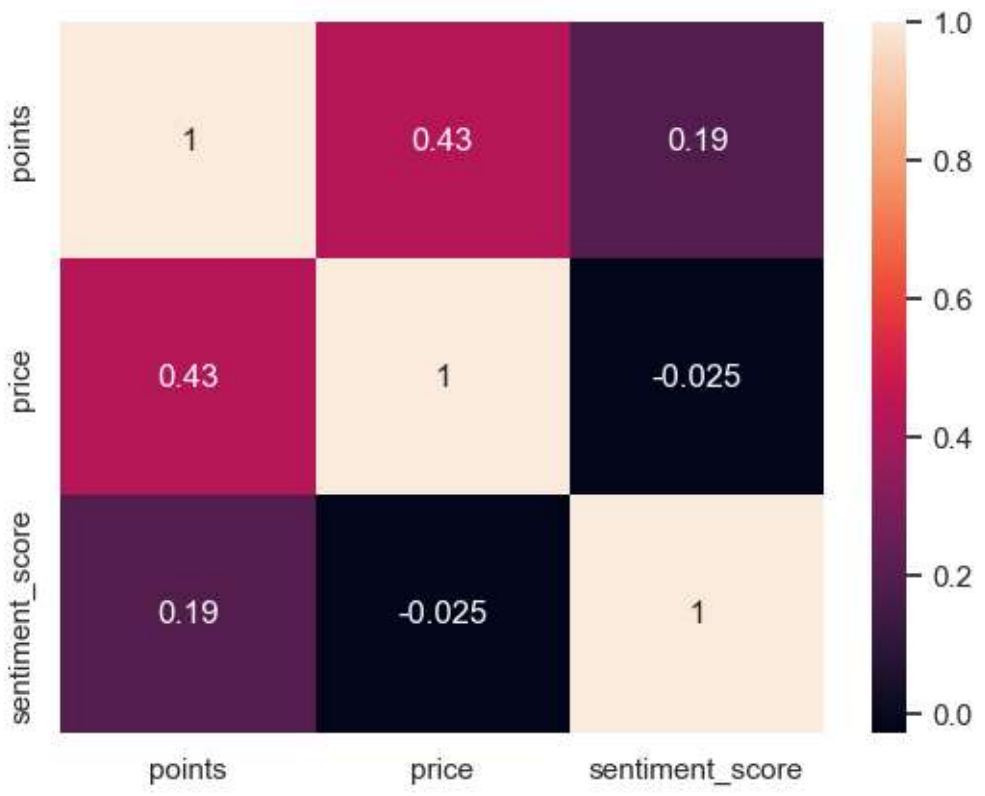
Looking for Correlations in Between Price, Points and Sentiment Score

We can plot a seaborn pairplot to look at the relationships between Price, Points and Sentiment Score

```
sns.pairplot(wine_df.iloc[:,1:13])
```



From the above pair plot the Sentiment_Score histogram shows a large number at 0.0, the neutral mark. There's a smattering of negative scored descriptions but mostly the scores are positive.
On the Sentiment_Score v Points plot, as most of the points are over 85, which seems highly positive, it just seems to suggest that a few of these highly positive point allocation have negatively assessed sentiment scores. The Sentiment_Score v Price suggests that at the higher price most scores are positive. However these is one outlier which has a $500 price but negative sentiment_score around -0.5.

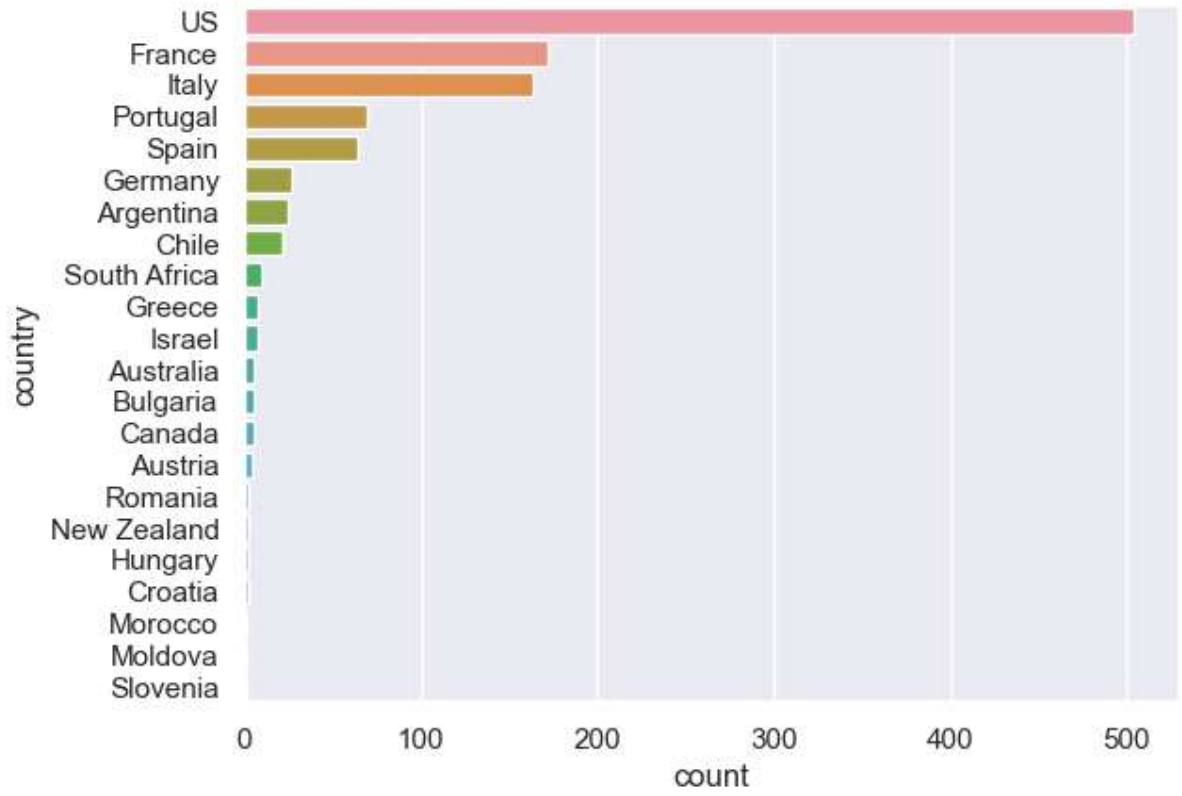


This heatmap would suggest that there is almost no correlation between price and the sentiment_score (-0.025)
This would mean there is hardly any link between price paid and writing a positive review. The actual correlation is negative and very small.

The correlation between price and points is much higher than the correlation between price and sentiment_score.

There is a positive correlation between price and points given (0.43). This means that as the price goes up the wine is rated more highly in the points by the reviewer.

Top Wine Producing Countries

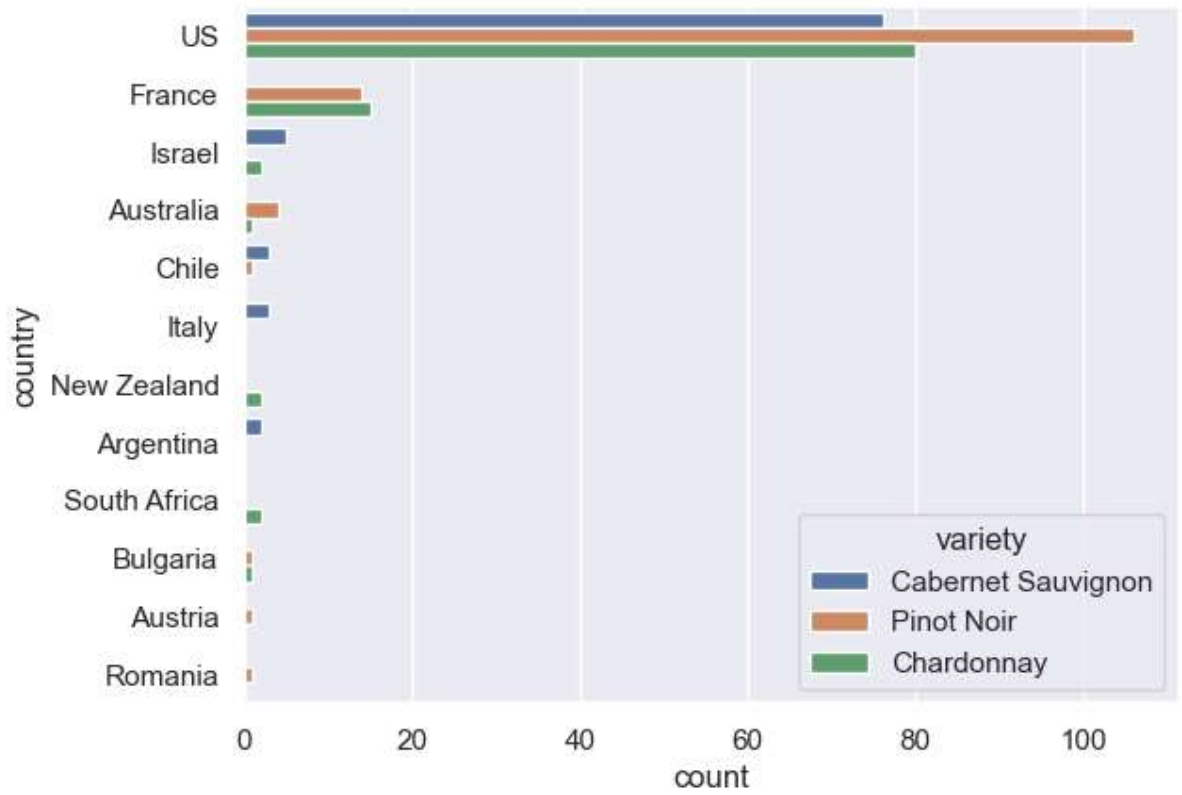


Grouping by variety we can see the top 3 grape varieties represented in the dataset are:

	Frequency
variety	
Pinot Noir	128
Chardonnay	103
Cabernet Sauvignon	89

By grouping by variety and using the mean method we can see the average points for these 3 grapes:

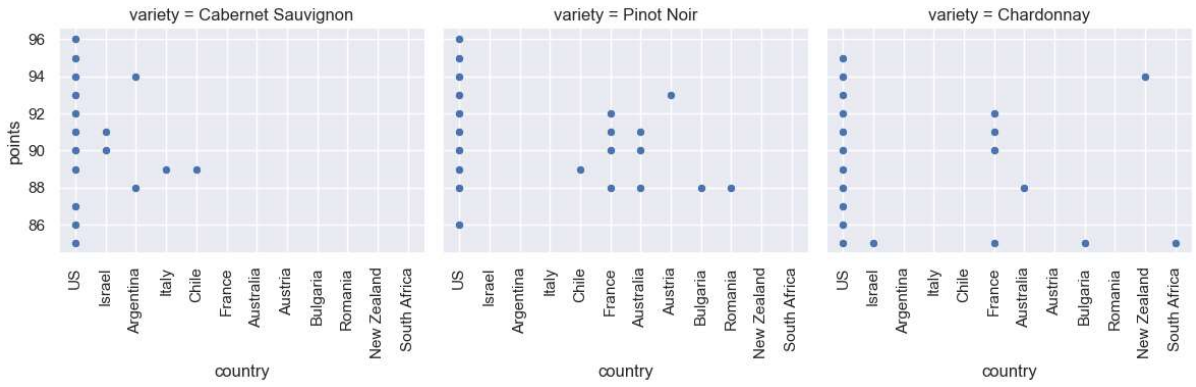
	points
variety	
Cabernet Sauvignon	89.943820
Chardonnay	89.815534
Pinot Noir	90.828125



The above plot show the how much of the top 3 grape varieties are produced by each country. You can see that the US produces the most in the 3 top varieties. Next it is France who only produces Pinot Noir and Chardonnay. Then Israel is next which produce Cabernet Sauvignon and Chardonnay.

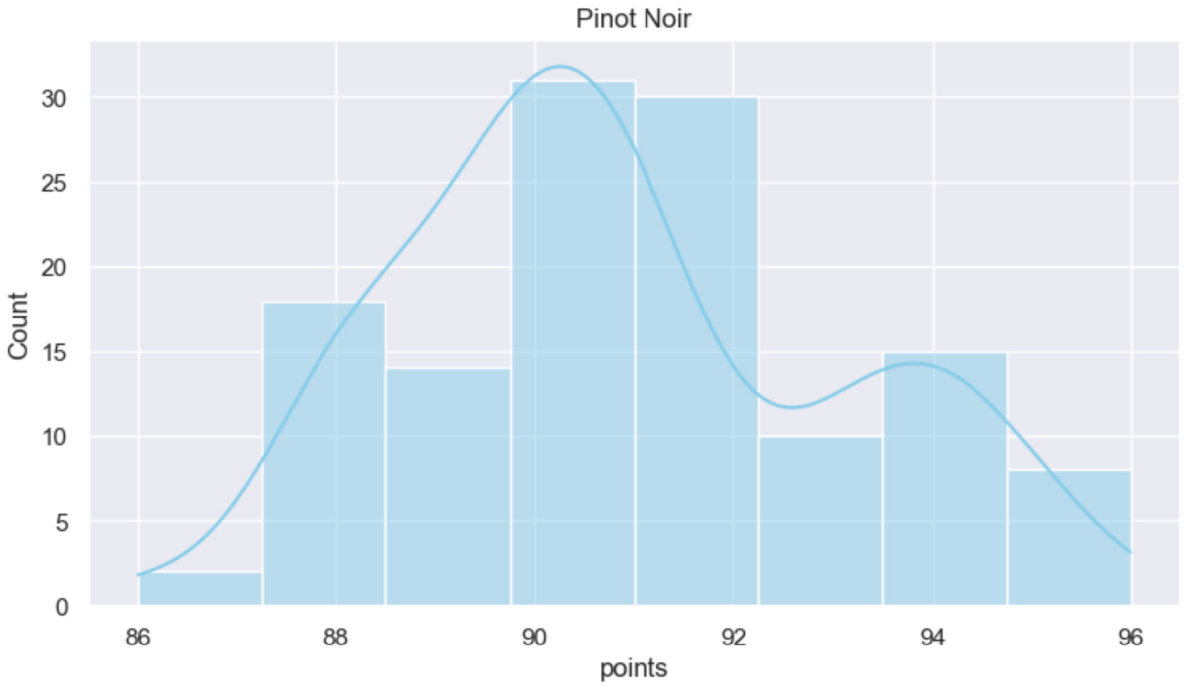
Multi-Plot Grid of the Top 3 Varieties (Cabernet Sauvignon, Pinot Noir and Chardonnay)

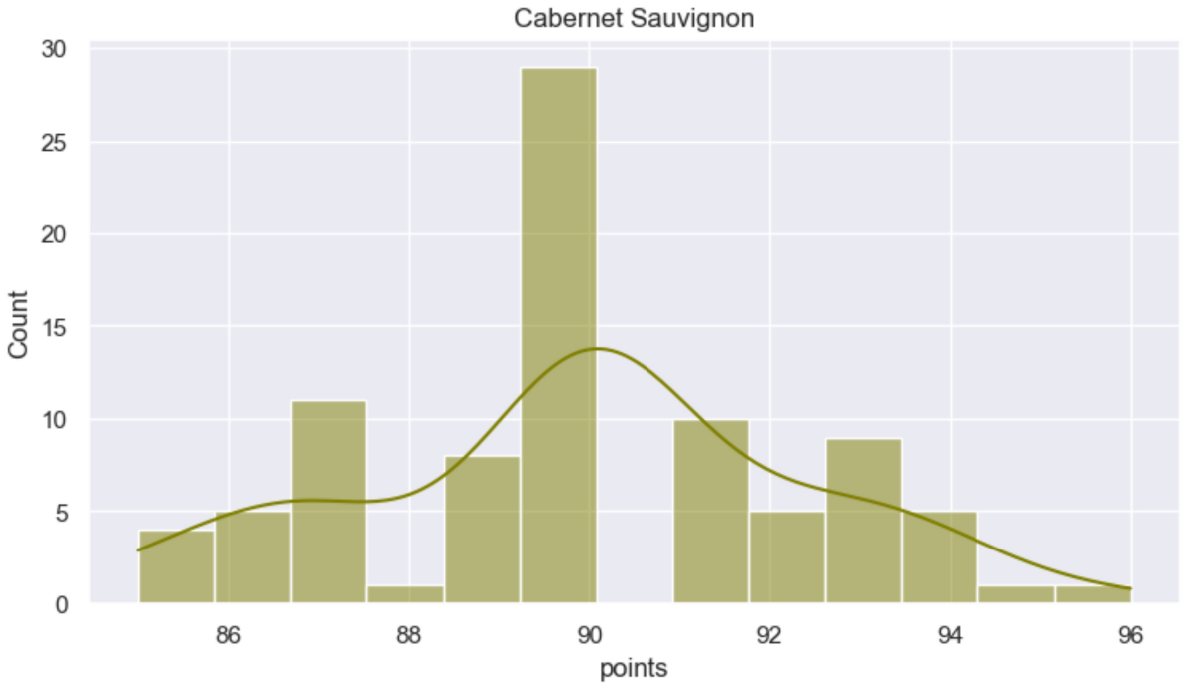
Using Seaborn and FacetGrid we can plot 3 scatterplots on the same plot -

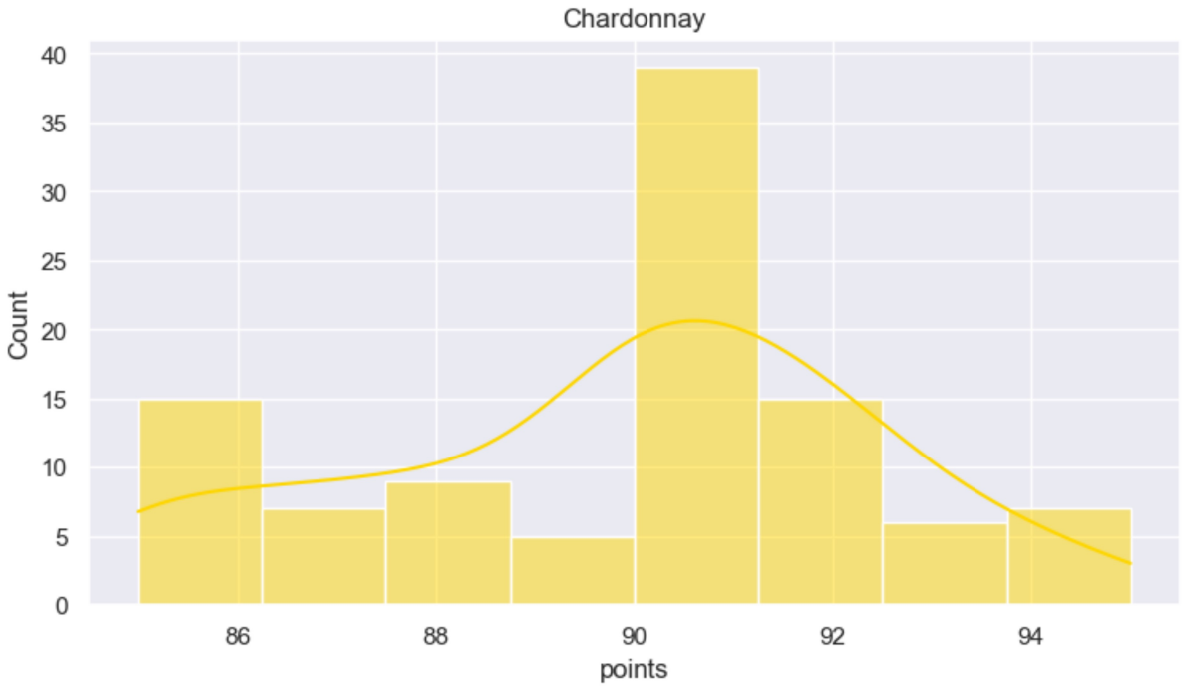


The above multi-plot shows that the US produces the most of these varieties of wines but their points ratings are the more widely dispersed than the other countries.
France produces no Cabernet Sauvignon, but some Pinot Noir and Chardonnay that have a points rating of between 88 and 92, although is one Chardonnay that has a points rating of much lower at below 86.

Cabernet Sauvignon, Pinot Noir and Chardonnay in a Seaborn histogram plot of points

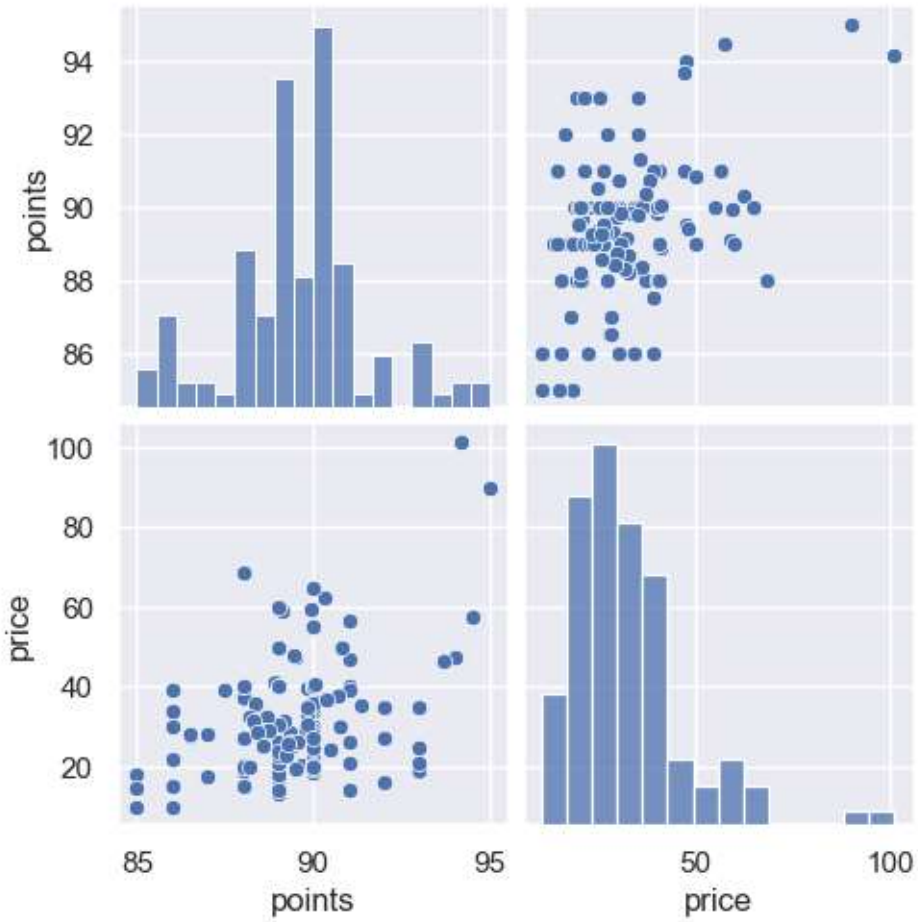

Pinot Noir
Count
points
0
5
10
15
20
25
30
86
88
90
92
94
96


Cabernet Sauvignon
Count
points
0
5
10
15
20
25
30
86
88
90
92
94
96


Chardonnay
Count
points
0
5
10
15
20
25
30
35
40
86
88
90
92
94

The above histograms are close to the Gaussian distribution.

Pair Plot of Points and Price



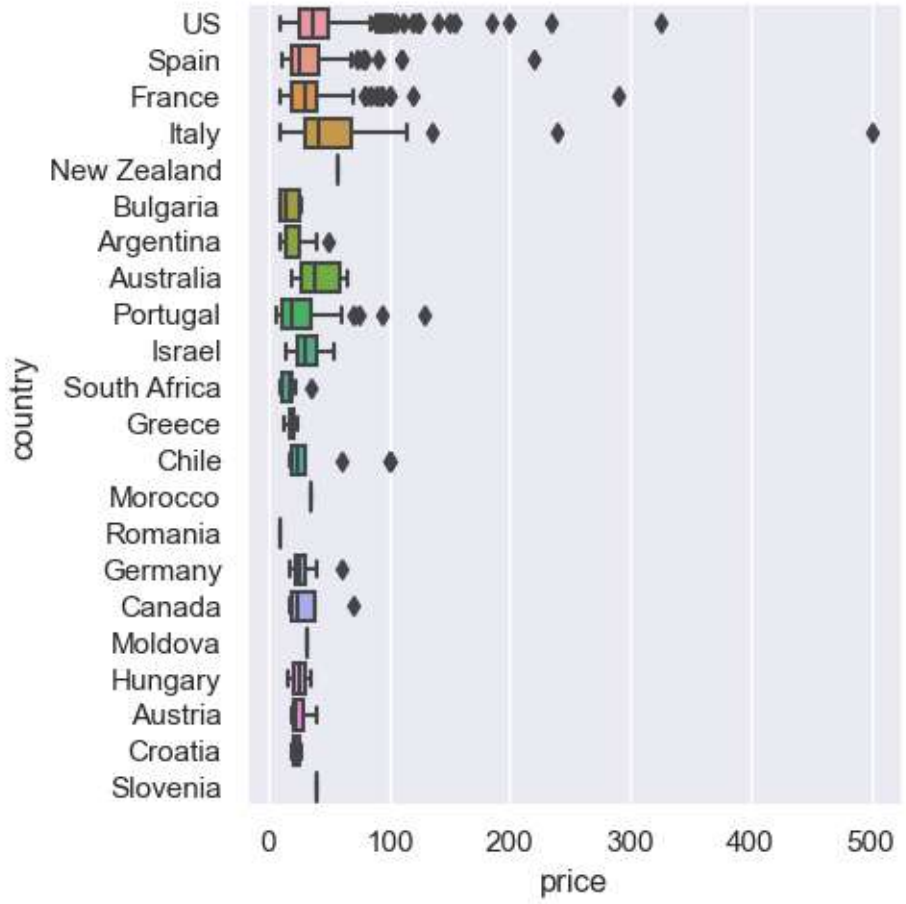
From the above pair plot there seems to be a positive correlation between price and points. Most of the wines in our dataset is less than £40. The points range from around 85 to 95 and are centred around the 90 point mark.

Table of the Top 15 Wines by Country, Ranked by Points

	points	price
country		
New Zealand	94.0	57.0
Austria	93.0	26.0
Morocco	93.0	35.0
Hungary	91.0	26.0
Germany	91.0	28.0
Canada	90.0	34.0
Moldova	90.0	32.0
Slovenia	90.0	40.0
Spain	90.0	38.0
US	90.0	43.0
Italy	90.0	54.0
Chile	90.0	36.0
France	90.0	36.0
Australia	90.0	42.0
Greece	89.0	19.0

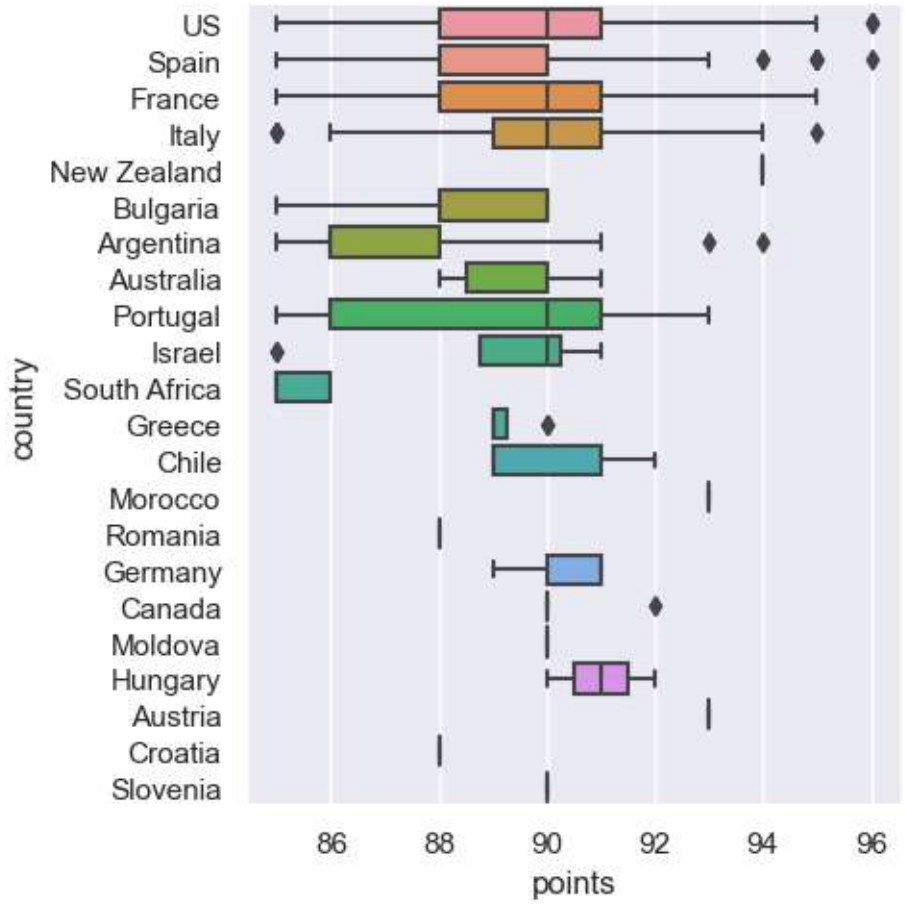
The above table shows that New Zealand has the most points on average. It's wines costs $57 on average. US and France, the top 2 wine producers are not even in the top 10 for points.

Box Plot Wine Price by Country (Seaborn Catplot)



The above boxplot shows a lot variance for US and Italian wines but the median wine prices are below $50. Slovenia seems to have the highest median wine price but there is not large representation of Slovenian wine in the dataset. Italy has the highest priced wine at $500.

Box Plot Wine Points by Country (Seaborn Catplot)



The above box plot shows most countries with a median points rating of 90. Hungary performs well with all its wine's points above 90. America and Spain have the widest spread of wine points shown here.

Word Clouds of US v French Wine

From the above Seaborn count plot we can see that the top 2 countries are the US and France.
Let's look at the words in the reviews of their wines next.

To get the string to use in the Word Cloud we need to use a for loop to go through the description column and concatenate into a string.

French stand out words from the reviews: crisp, aged, perfumed, fresh, character, rich, acidity, bright, softo, wood, great

US stand out words: flavour, cherry, oak, rich, blackberry, fruit, light, texture, apple, dark, tobacco French Word Cloud of Reviews:

French Word Cloud of Reviews:

US Word Cloud of Reviews:

Displaying the top 5 highest priced provinces (average price grouped by provinces)

	points	price
province		
Aconcagua Valley	91.000000	100.000000
Champagne	91.000000	69.500000
Piedmont	90.396552	68.477241
Tasmania	90.000000	65.000000
Maule Valley	89.000000	61.000000

The province with the highest price on average is the Aconcagua Valley, in Chile. Their wine is on average $100. The next province is Champagne in France with an average bottle at $69.50.

Highest Rates Wines by Country

	points	price
country		
New Zealand	94.0	57.0
Austria	93.0	26.0
Morocco	93.0	35.0
Hungary	91.0	26.0
Germany	91.0	28.0
Canada	90.0	34.0
Moldova	90.0	32.0
Slovenia	90.0	40.0
Spain	90.0	38.0
US	90.0	43.0
Italy	90.0	54.0
Chile	90.0	36.0
France	90.0	36.0
Australia	90.0	42.0
Greece	89.0	19.0

The above table shows that New Zealand has the most points on average. It's wines costs $57 on average.

This table shows the best rated, by average points, grape variety

	points	price
variety		
Tannat	95.0	90.0
Friulano	94.0	58.0
Tinta de Toro	94.0	101.0
Provence red blend	94.0	48.0
Tannat-Cabernet	94.0	47.0

This shows that the grape variety Tannat is the highest rated grape variety with 95 points. It costs $90 on average.

Displaying the top 5 highest priced wineries (average points grouped by winery)

	points	price
winery		
Macauley	96.0	90.0
Bodega Carmen Rodríguez	95.5	110.0
Heitz	95.5	129.5
Center of Effort	95.0	60.0
Hall	95.0	325.0

This table shows the Macauley winery as the highest rated by average points. The price for their wine is $90.

Resources:

https://realpython.com/python-nltk-sentiment-analysis/

ENSURE THIS DOCUMENT IS NEAT AND ADD IT TO YOUR PORTFOLIO

THIS REPORT WAS WRITTEN BY : Michael Sullivan