DATA 1 Practical 1 - Model Answers

Simos Gerasimou

Wine Exploration

WineEnthusiast is a website for buying wine products and in which customers can also review products. The company has collected reviews for a wide variety of their products on November 22nd, 2017. The company wants to analyse this data to extract insights from its products and answer questions including:

- explore how its products are rated by its customers
- identify patterns that might increase its revenue and/or profit.

Your tasks are to explore this dataset and generade some actionable knowledge.

This Jupyter Notebook will be presented to the WineEnthusiast main stakeholders who have limited knowledge about data science. So, your findings should be complemented by a suitable justification explaining what you observe and, when applicable, what does this observation mean and, possibly, why it occurs.

- For each question (task) a description is provided accompanied (most of the time) by two cells: one for writing the Python code and another for providing the justification. Feel free to add more cells if you feel they are needed, but keep the cells corresponding to the same question close by.

---

1) Reading dataset

The wine review dataset is available on the following links:

- A small (10K) dataset (4MBs): https://drive.google.com/file/d/1XvdXCtqmuwdZZUkBKwd4Ml4oMsBsy7T7/view?usp=sharing (https://drive.google.com/file/d/1XvdXCtqmuwdZZUkBKwd4Ml4oMsBsy7T7/view?usp=sharing)
- A bigger (100K) dataset(42MBs): https://drive.google.com/file/d/12ne4Tu1XHW86IcWB0dMd-SBH2C8upBP9/view?usp=sharing (https://drive.google.com/file/d/12ne4Tu1XHW86IcWB0dMd-SBH2C8upBP9/view?usp=sharing)

To save time with reading, loading and analysing the dataset, I suggest to start with the small dataset and use the big dataset once you have completed all the tasks for the small dataset.

T1) Use the Python Standard Library or Numpy to read the dataset

Hint: This question was included in the exam of SOF1.

In [2]:

```
# Write your code here
# See: https://docs.python.org/3/library/csv.html
import csv
data_path = "wine-data-filtered-10K.csv"
with open(data_path, encoding="utf-8") as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=',')
    winedata = list(csv_reader)
```

T2) Explore the dataset and try to understand the meaning of each column. For each column, write its meaning and its data type

- If you find it difficult (only then), you can go to https://docs.google.com/spreadsheets/d/1w9B_u50z6Oi703qi9UdISRhPZj_dFkrCKY9dA_mhWz4/edit?usp=sharing (https://docs.google.com/spreadsheets/d/1w9B_u50z6Oi703qi9UdISRhPZj_dFkrCKY9dA_mhWz4/edit?usp=sharing) for help.

Answer for T2

- **Country**: the country where the wine is from
- **Description**: a few sentences from a sommelier describing the wine's taste, feel etc
- **Points**: the number of points WineEnthusiast rated the wine on a scale of 1–100
- **Price**: the cost for a bottle of the wine.
- **Province**: the province or state that the wine is from.
- **Taster_name**: Name of a person who taste it.
- **Title**: A brief title of a wine.
- **Variety**: the type of grapes used to make the wine
- **Winery**: the winery that made the wine

---

2) Which countries and wine varieties have received most of the reviews?

T3) Find the 5 most popular countries in terms of number of reviews and their percentage over the total reviews

For instance, if the reviews for the 5 most popular countries are 10 + 11 + 12 + 13 + 14 + 15 = 75 and the total reviews are 100, then the percentage is 75/100.

In [3]:

```
# Write your answer here
def mostPopularCountries():
    reviewsPerCountry = {}
    for i in range(1,len(winedata)):
        if (winedata[i][1] in reviewsPerCountry):
            v = reviewsPerCountry[winedata[i][1]]
            v = v+1
            reviewsPerCountry.update({winedata[i][1]: v})
        else:
            reviewsPerCountry[winedata[i][1]] = 1
    return reviewsPerCountry

reviewsPerCountry=mostPopularCountries()

#Sort dictionary by value Method 1: sorted function and list comprehension
reviewsPerCountrySorted1 = sorted(((v, k) for (k,v) in reviewsPerCountry.items
()), reverse=True)
print("Most popuplar countries method 1:", reviewsPerCountrySorted1[0:5])

#Sort dictionary by value Method 2: sorted and lamda functions
reviewsPerCountrySorted2 = sorted(reviewsPerCountry.items(), key=lambda item: it
em[1], reverse=True)
print("Most popuplar countries method 2:", reviewsPerCountrySorted2[0:5])

#Sort dictionary by value Method 3: sorted function and operator module
import operator
reviewsPerCountrySorted3 = sorted(reviewsPerCountry.items(), key=operator.itemge
tter(1), reverse=True)
print("Most popuplar countries method 3:", reviewsPerCountrySorted3[0:5])

#I opt for method 2
reviewsForPopular5 = 0
for country in reviewsPerCountrySorted2[0:5]:
    reviewsForPopular5 += country[1]

print("Total reviews for popular 5 countries:", reviewsForPopular5)
print("Percentage of reviews for popular 5 countries:", reviewsForPopular5/len(w
inedata)*100)
```

```
Most popuplar countries method 1: [(3873, 'US'), (1741, 'France'),
(1072, 'Italy'), (656, 'Spain'), (525, 'Portugal')]
Most popuplar countries method 2: [('US', 3873), ('France', 1741),
('Italy', 1072), ('Spain', 656), ('Portugal', 525)]
Most popuplar countries method 3: [('US', 3873), ('France', 1741),
('Italy', 1072), ('Spain', 656), ('Portugal', 525)]
Total reviews for popular 5 countries: 7867
Percentage of reviews for popular 5 countries: 78.66213378662134
```

T4) Find the most 5 popular wine varieties and their percentage over the total reviews

In [4]:

```
# Write your answer here
def mostPopularVarieties():
    reviewsPerVariety = {}
    for i in range(1,len(winedata)):
        if (winedata[i][8] in reviewsPerVariety):
            v = reviewsPerVariety[winedata[i][8]]
            v = v+1
            reviewsPerVariety.update({winedata[i][8]: v})
        else:
            reviewsPerVariety[winedata[i][8]] = 1
    return (reviewsPerVariety)

reviewsPerVariety = mostPopularVarieties()

#Sort dictionary by value Method 1: sorted function and list comprehension
reviewsPerVarietySorted = sorted(((v, k) for (k,v) in reviewsPerVariety.items
()), reverse=True)
print("Most popuplar wine varieties:", reviewsPerVarietySorted[0:5])

reviewsForPopular5 = 0
for variety in reviewsPerVarietySorted[0:5]:
    reviewsForPopular5 += variety[0]

print("Total reviews for popular 5 wine varieties:", reviewsForPopular5)
print("Percentage of reviews for popular 5:", reviewsForPopular5/len(winedata)*1
00)
```

```
Most popuplar wine varieties: [(978, 'Pinot Noir'), (835, 'Chardonna
y'), (728, 'Red Blend'), (702, 'Cabernet Sauvignon'), (505, 'Bordeau
x-style Red Blend')]
Total reviews for popular 5 wine varieties: 3748
Percentage of reviews for popular 5: 37.476252374762524
```

T5) Which is the most widely reviewed winery?

Hint: In which format should the data be transformed so that you can apply some statistical metrics we have seen?

In [5]:

```
# Write your answer here
# You might consider helpful using scipy
# Check the various functions of scipy.stats https://docs.scipy.org/doc/scipy/reference/stats.html
import numpy as np
from scipy import stats
wineDataArray = np.array(winedata[1:]) #transform list into numpy array

mode, count = stats.mode(wineDataArray[:,9], nan_policy='omit')

print('The most reviewed winery is ', mode, 'with ', count, 'reviews')
```

```
/usr/local/lib/python3.6/site-packages/scipy/stats/stats.py:245: RuntimeWarning: The input array could not be properly checked for nan values. nan values will be ignored.
  "values. nan values will be ignored.", RuntimeWarning)

The most reviewed winery is  ['Wines & Winemakers'] with  [28] reviews
```

---

3) How do the wine prices look like?

T6) Calculate the range and standard deviation of wine prices for the entire dataset

In [6]:

```
# Write your answer here
#You might consider helpful to use Numpy
#Check the statistical Numpy functions: https://docs.scipy.org/doc/numpy/reference/routines.statistics.html
winePrices = wineDataArray[:,4].astype(np.float) #transform string to float
min=np.amin(winePrices)
max=np.amax(winePrices)
print('Min wine price:', min)
print('Max wine price:', max)
print('Range:', max-min)


print('Std:', np.std(winePrices))
```

```
Min wine price: 4.0
Max wine price: 1900.0
Range: 1896.0
Std: 43.728817389794564
```

T7) Calculate the mean and median prices for all the wines

In [7]:

```
# Write your answer here
#You might consider helpful to use Numpy
#Check the statistical Numpy functions: https://docs.scipy.org/doc/numpy/referen
ce/routines.statistics.html
print('Mean wine price:',   np.mean(winePrices))
print('Median wine price:', np.median(winePrices))
```

```
Mean wine price: 34.7813
Median wine price: 25.0
```

T8) What insights can you extract from these values? Which metric of central tendency should we use?

Answer for T8

There is a significant difference between the mean and median wine price. This is a potential sign that some very expensive wines exist in the dataset.

---

4) What do the reviewers think about the quality of wines?

T9) Calculate the range of wine ratings (point) for the entire dataset

In [8]:

```
# Write your answer here
#You might consider helpful to use Numpy
wineRatings = wineDataArray[1:,3].astype(np.float)
minR = np.amin(wineRatings)
maxR = np.amax(wineRatings)
print('Min wine rating:', minR)
print('Max wine rating:', maxR)
print('Range:', maxR-minR)
```

```
Min wine rating: 80.0
Max wine rating: 100.0
Range: 20.0
```

T10) Calculate the mean and median ratings for all the wines

In [9]:

```
# Write your answer here
#You might consider helpful to use Numpy
print('Mean rating:', np.mean(wineRatings))
print('Median rating:', np.median(wineRatings))
```

```
Mean rating: 88.46864686468646
Median rating: 88.0
```

T11) What insights can you extract from these values? Which metric of central tendency should we use?

Answer for T11

The mean and median wine ratings are very close indicating the potential absence of very high or very low values.

---

5) How do the price and rating values vary?

T12) Calculate the interquartile range for the price of all reviewed wines

In [10]:

```
# Write your answer here
#You might consider helpful to use Numpy
Q3P = np.percentile(winePrices, 75) #Third quartile
Q1P = np.percentile(winePrices, 25) #First quartile
IQRP = Q3P - Q1P #Inter Quartile Range
print('Price IQR:', IQRP)
```

```
Price IQR: 24.0
```

T13) Calculate the interquartile range for the ratings of all reviewed wines

In [11]:

```
# Write your answer here
#You might consider helpful to use Numpy
Q3R = np.percentile(wineRatings, 75) #Third quartile
Q1R = np.percentile(wineRatings, 25) #First quartile
IQRR = Q3R - Q1R #Inter Quartile Range
print('Rating IQR:', IQRR)
```

```
Rating IQR: 5.0
```

T14) Plot the Box whisker plot showing the wine price for the 10 most reviewed countries

In [12]:

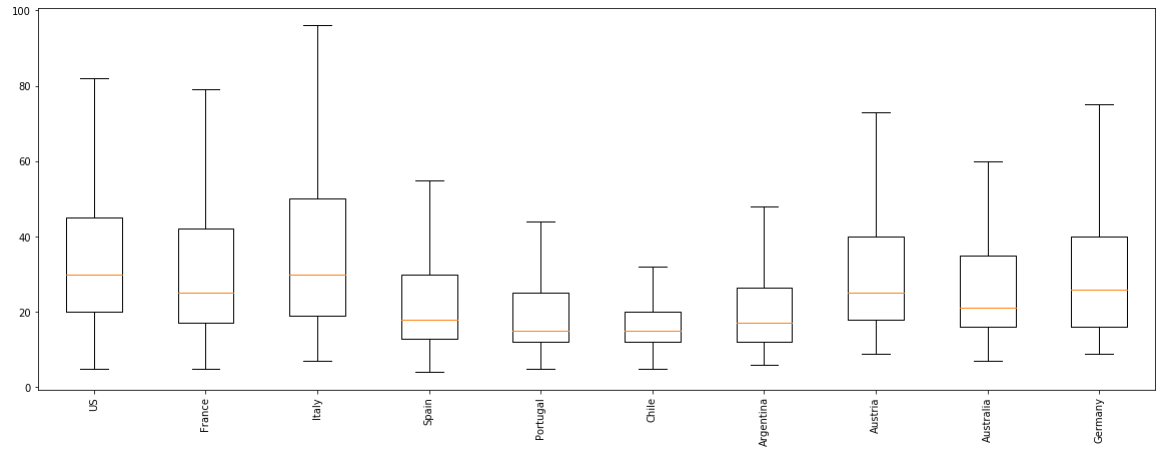
```
# Write your answer here
#You might consider helpful to use Matplot lib: https://matplotlib.org/3.1.1/gal
lery/statistics/boxplot_demo.html
#Also, you might reuse data from T3

%matplotlib inline
import matplotlib.pyplot as plt

pricePerCountry = {}
uniqueCountries = np.unique(wineDataArray[1:,1])
for country in reviewsPerCountrySorted2[0:10]:
    pricePerCountry[country[0]]= wineDataArray[wineDataArray[:,1]==country[0]]
[:,4].astype(np.float)

# Multiple box plots on one axis
fig, ax = plt.subplots(figsize = (20,7))
ax.boxplot(pricePerCountry.values(), showfliers=False)
ax.set_xticklabels(pricePerCountry.keys())
plt.xticks(rotation = 90)


plt.show()
```



T15) Plot the Box whisker plot showing the wine rating for the 10 most reviewed countries

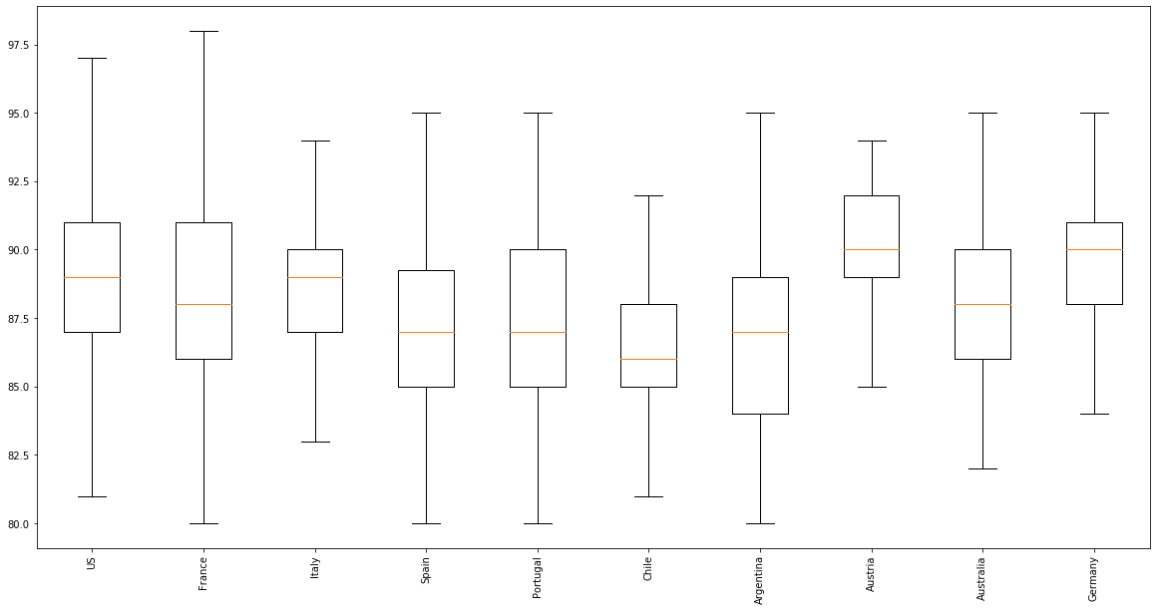
In [13]:

```
# Write your answer here
#You might consider helpful to use Matplot lib: https://matplotlib.org/3.1.1/gal
lery/statistics/boxplot_demo.html
#Also, you might reuse data from T3

ratingPerCountry = {}
uniqueCountries = np.unique(wineDataArray[1:,1])
for country in reviewsPerCountrySorted2[0:10]:
    ratingPerCountry[country[0]]= wineDataArray[wineDataArray[:,1]==country[0]]
[:,3].astype(np.float)

# Multiple box plots on one axis
fig, ax = plt.subplots(figsize = (20,10))
ax.boxplot(ratingPerCountry.values(), showfliers=False)
ax.set_xticklabels(ratingPerCountry.keys())
plt.xticks(rotation = 90)


plt.show()
```



T16) Discuss your findings

Answer for T16

- Prices in general between 20-40
- Ratings in general between 85-92
- ...

Should you finish earlier/want to practice at home, you could

- Plot the box whisker plots of wine rating/price per variety and winery (4 more boxplots in total)
- Find the tasters (sommellier) who provided the most reviews, the higher or the lowest ratings
- Calculate the length of each description and investigate whether there is a pattern in the length.
- Any other analysis that you might could generate some useful insight.