COMPLETED SAMPLE OF WORK DATA SCIENCE

(Data Visualization and Storytelling)

NORIE JEANNE PEREIRA

Data visualization and storytelling has always been one of the most important phases of any data science pipeline involving extracting meaningful insights from data, regardless of the complexity of the data or the project.

The main goal of data visualization is to have the reader quickly digest the data, including possible trends, relationships, and more. Ideally, a reader will not have to spend more than 5-6 seconds digesting a single visualization. For this reason, we must make visuals very seriously and ensure that we are making a visual as effective as possible.

The Challenge

Visualizing Structured Multi-Dimensional Data

Workflow goals

First, you'll want to answer a set of basic questions about the dataset:

1. How many observations do I have?
2. How many features?
3. What are the data types of my features? Are they numeric? Categorical?
4. Do I have a target variable

To describe any visualization with one or more dimensions, we can use the components as follows.

1. Data: Always start with the data, identify the dimensions you want to visualize.
2. Aesthetics: Confirm the axes based on the data dimensions, positions of various data points in the plot. Also check if any form of encoding is needed including size, shape, color and so on which are useful for plotting multiple data dimensions.
3. Scale: Do we need to scale the potential values, use a specic scale to represent multiple values or a range?
4. Geometric objects: These are popularly known as 'geoms'. This would cover the way we would depict the data points on the visualization. Should it be points, bars, lines and so on?
5. Statistics: Do we need to show some statistical measures in the visualization like measures of central tendency, spread, condence intervals?
6. Facets: Do we need to create subplots based on specic data dimensions?
7. Coordinate system: What kind of a coordinate system should the visualization be based on — should it be cartesian or polar?

1. We'll start by loading up the following necessary dependencies for our analyses.

```
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib as mpl
import numpy as np
import seaborn as sns

%matplotlib inline
```

2. We will mainly be using matplotlib and seaborn as our visualization frameworks here .

```
#basic data pre-processing steps
white_wine = pd.read_csv('C:/5 Data Science Project Tuturial/wine/winequality-white.csv', sep=';')
red_wine = pd.read_csv('C:/5 Data Science Project Tuturial/wine/winequality-red.csv', sep=';')

# store wine type as an attribute
red_wine['wine_type'] = 'red'
white_wine['wine_type'] = 'white'

# bucket wine quality scores into qualitative quality labels
red_wine['quality_label'] = red_wine['quality'].apply(lambda value: 'low'
                                            if value <= 5 else 'medium'
                                              if value <= 7 else 'high')
red_wine['quality_label'] = pd.Categorical(red_wine['quality_label'],
                                   categories=['low', 'medium', 'high'])
white_wine['quality_label'] = white_wine['quality'].apply(lambda value: 'low'
                                            if value <= 5 else 'medium'
                                              if value <= 7 else 'high')
white_wine['quality_label'] = pd.Categorical(white_wine['quality_label'],
                                   categories=['low', 'medium', 'high'])

# merge red and white wine datasets
wines = pd.concat([red_wine, white_wine])

# re-shuffle records just to randomize data points
wines = wines.sample(frac=1, random_state=42).reset_index(drop=True)
```

3.
#create a single data frame wines by merging both the datasets
create a new categorical variable quality_label based on the quality attribute of wine samples

wines.head()

We create a single data frame wines by merging both the datasets pertaining to red and white wine samples. We also create a new categorical variable quality_label based on the quality attribute of wine samples.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	wine_ty
0	7.0	0.17	0.74	12.8	0.045	24.0	126.0	0.99420	3.26	0.38	12.2	8	white
1	7.7	0.64	0.21	2.2	0.077	32.0	133.0	0.99560	3.27	0.45	9.9	5	red
2	6.8	0.39	0.34	7.4	0.020	38.0	133.0	0.99212	3.18	0.44	12.0	7	white
3	6.3	0.28	0.47	11.2	0.040	61.0	183.0	0.99592	3.12	0.51	9.5	6	white
4	7.4	0.35	0.20	13.9	0.054	63.0	229.0	0.99888	3.11	0.50	8.9	6	white

It is quite evident that we have several numeric and categorical attributes for wine samples. Each observation belongs to a red or white wine sample and the attributes are specific attributes or properties measured and obtained from physicochemical tests.

4. Let's do a quick basic descriptive summary statistics.

#Exploratory Data Analysis and Visualizations
#Descriptive Statistics

subset_attributes = ['residual sugar', 'total sulfur dioxide', 'sulphates', 'alcohol', 'volatile acidity', 'quality']
rs = round(red_wine[subset_attributes].describe(),2)
ws = round(white_wine[subset_attributes].describe(),2)
pd.concat([rs, ws], axis=1, keys=['Red Wine Statistics', 'White Wine Statistics'])

	Red Wine Statistics						White Wine Statistics				
	residual sugar	total sulfur dioxide	sulphates	alcohol	volatile acidity	quality	residual sugar	total sulfur dioxide	sulphates	alcohol	volatile acidity
count	1599.00	1599.00	1599.00	1599.00	1599.00	1599.00	4898.00	4898.00	4898.00	4898.00	4898.00
mean	2.54	46.47	0.66	10.42	0.53	5.64	6.39	138.36	0.49	10.51	0.28
std	1.41	32.90	0.17	1.07	0.18	0.81	5.07	42.50	0.11	1.23	0.10
min	0.90	6.00	0.33	8.40	0.12	3.00	0.60	9.00	0.22	8.00	0.08
25%	1.90	22.00	0.55	9.50	0.39	5.00	1.70	108.00	0.41	9.50	0.21
50%	2.20	38.00	0.62	10.20	0.52	6.00	5.20	134.00	0.47	10.40	0.26
75%	2.60	62.00	0.73	11.10	0.64	6.00	9.90	167.00	0.55	11.40	0.32
max	15.50	289.00	2.00	14.90	1.58	8.00	65.80	440.00	1.08	14.20	1.10

5.
subset_attributes = ['alcohol', 'volatile acidity', 'pH', 'quality']
ls = round(wines[wines['quality_label'] == 'low'][subset_attributes].describe(),2)
ms = round(wines[wines['quality_label'] == 'medium'][subset_attributes].describe(),2)
hs = round(wines[wines['quality_label'] == 'high'][subset_attributes].describe(),2)
pd.concat([ls, ms, hs], axis=1, keys=['Low Quality Wine', 'Medium Quality Wine', 'High Quality Wine'])

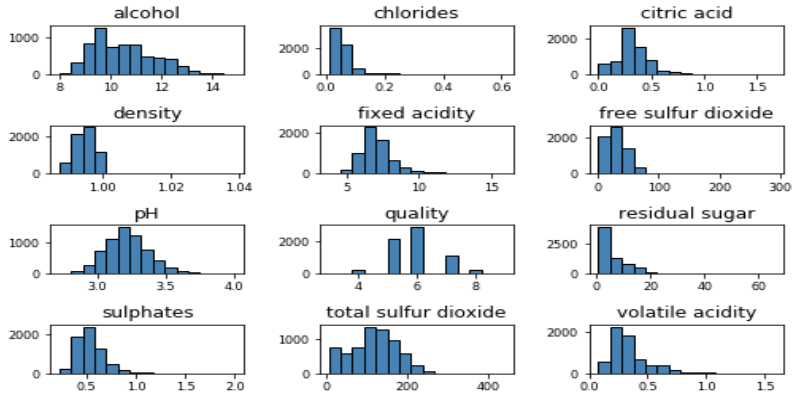
	Low Quality Wine				Medium Quality Wine				High Quality Wine			
	alcohol	volatile acidity	pH	quality	alcohol	volatile acidity	pH	quality	alcohol	volatile acidity	pH	quality
count	2384.00	2384.00	2384.00	2384.00	3915.00	3915.00	3915.00	3915.00	198.00	198.00	198.00	198.00
mean	9.87	0.40	3.21	4.88	10.81	0.31	3.22	6.28	11.69	0.29	3.23	8.03
std	0.84	0.19	0.16	0.36	1.20	0.14	0.16	0.45	1.27	0.12	0.16	0.16
min	8.00	0.10	2.74	3.00	8.40	0.08	2.72	6.00	8.50	0.12	2.88	8.00
25%	9.30	0.26	3.11	5.00	9.80	0.21	3.11	6.00	11.00	0.21	3.13	8.00
50%	9.60	0.34	3.20	5.00	10.80	0.27	3.21	6.00	12.00	0.28	3.23	8.00
75%	10.40	0.50	3.31	5.00	11.70	0.36	3.33	7.00	12.60	0.35	3.33	8.00
max	14.90	1.58	3.90	5.00	14.20	1.04	4.01	7.00	14.00	0.85	3.72	9.00

6. Univariate analysis is basically the simplest form of data analysis or visualization where we are only concerned with analyzing one data attribute or variable and visualizing the same (one dimension).

Visualizing data in One Dimension (1-D) One of the quickest and most effective ways to visualize all numeric data and their distributions, is to leverage histograms using pandas

```
#Univariate Analysis
#Visualizing one dimension
wines.hist(bins=15, color='steelblue', edgecolor='black', linewidth=1.0,
           xlabelsize=8, ylabelsize=8, grid=False)
plt.tight_layout(rect=(0, 0, 1.2, 1.2))
```

The plots below give a good idea about the basic data distribution of any of the attributes.

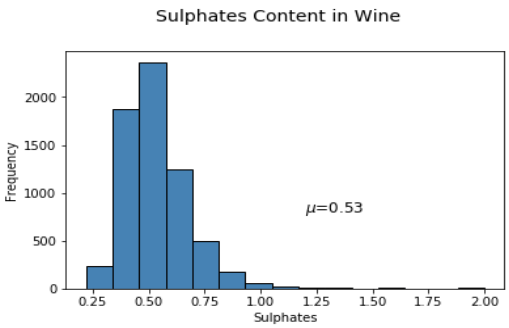


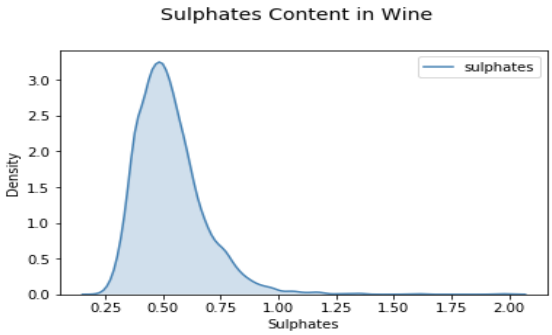
7. visualizing one of the continuous, numeric attributes. Essentially a histogram or a density plot works quite well in understanding how the data is distributed for that attribute.

```
#Continuous, numeric attribute in 1-D
fig = plt.figure(figsize = (6,4))
title = fig.suptitle("Sulphates Content in Wine", fontsize=14)
fig.subplots_adjust(top=0.85, wspace=0.3)

ax = fig.add_subplot(1,1, 1)
ax.set_xlabel("Sulphates")
ax.set_ylabel("Frequency")
ax.text(1.2, 800, r'$\mu$='+str(round(wines['sulphates'].mean(),2)),
        fontsize=12)
freq, bins, patches = ax.hist(wines['sulphates'], color='steelblue', bins=15,
                              edgecolor='black', linewidth=1)
```

8.

```
fig = plt.figure(figsize = (6, 4))
title = fig.suptitle("Sulphates Content in Wine", fontsize=14)
fig.subplots_adjust(top=0.85, wspace=0.3)

ax1 = fig.add_subplot(1,1, 1)
ax1.set_xlabel("Sulphates")
ax1.set_ylabel("Density")
sns.kdeplot(wines['sulphates'], ax=ax1, shade=True, color='steelblue')
```

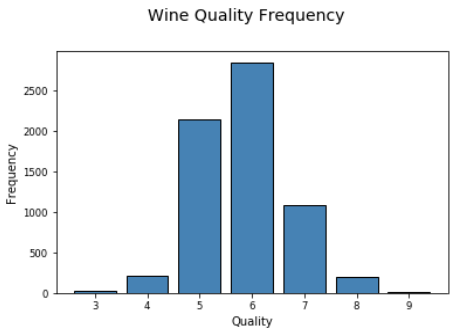




Tthere is a defenite right skew in the distribution for wine sulphates .

9 #Discrete, categorical attribute in 1-D
fig = plt.figure(figsize = (6, 4))
title = fig.suptitle("Wine Quality Frequency", fontsize=14)
fig.subplots_adjust(top=0.85, wspace=0.3)

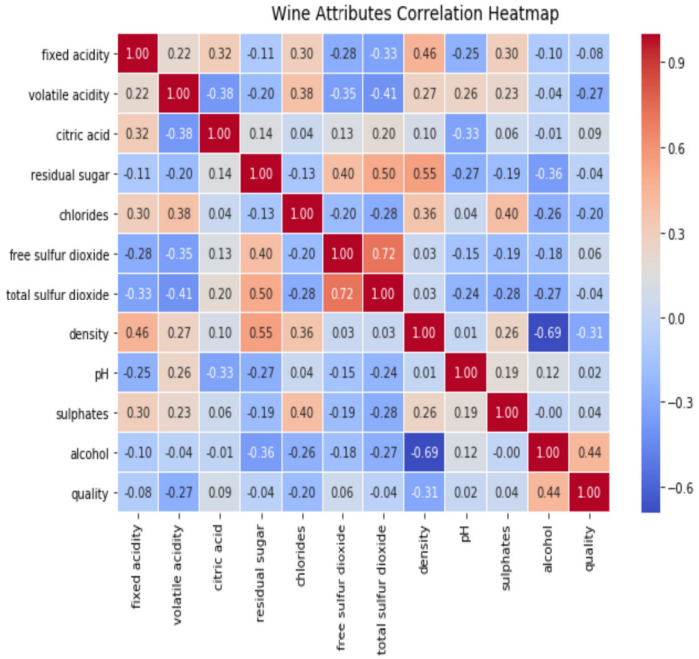
ax = fig.add_subplot(1,1, 1)
ax.set_xlabel("Quality")
ax.set_ylabel("Frequency")
w_q = wines['quality'].value_counts()
w_q = (list(w_q.index), list(w_q.values))
ax.tick_params(axis='both', which='major', labelsize=8.5)
bar = ax.bar(w_q[0], w_q[1], color='steelblue',
edgecolor='black', linewidth=1)



10. Multivariate analysis not only involves just checking out distributions but also potential relationships, patterns and correlations amongst these attributes. You can also leverage inferential statistics and hypothesis testing if necessary based on the problem to be solved at hand to check out statistical significance for different attributes, groups and so on.

#Multivariate Analysis
#Visualizing two dimensions
f, ax = plt.subplots(figsize=(10, 6))
corr = wines.corr()
hm = sns.heatmap(round(corr,2), annot=True, ax=ax, cmap="coolwarm",fmt='.2f',
linewidths=.05)
f.subplots_adjust(top=0.93)
t= f.suptitle('Wine Attributes Correlation Heatmap', fontsize=14)

One of the best ways to check out potential relationships or correlations amongst the different data attributes is to leverage a pair-wise correlation matrix and depict it as a heatmap.



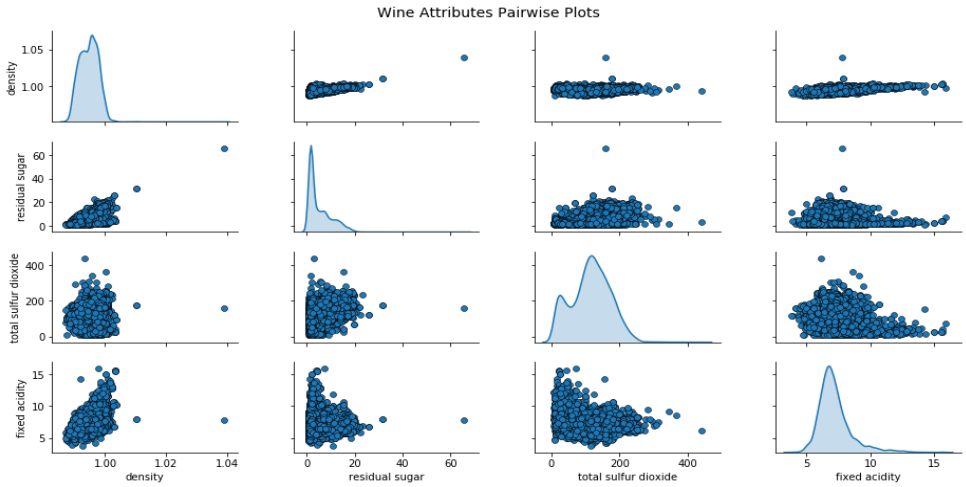
The gradients in the heatmap vary based on the strength of the correlation and you can clearly see it is very easy to spot potential attributes having strong correlations amongst themselves.

In general, you should look out for:

Which features are strongly correlated with the target variable? Are there interesting or unexpected strong correlations between other features Again, your aim is to gain intuition about the data, which will help you throughout th of the workflow.

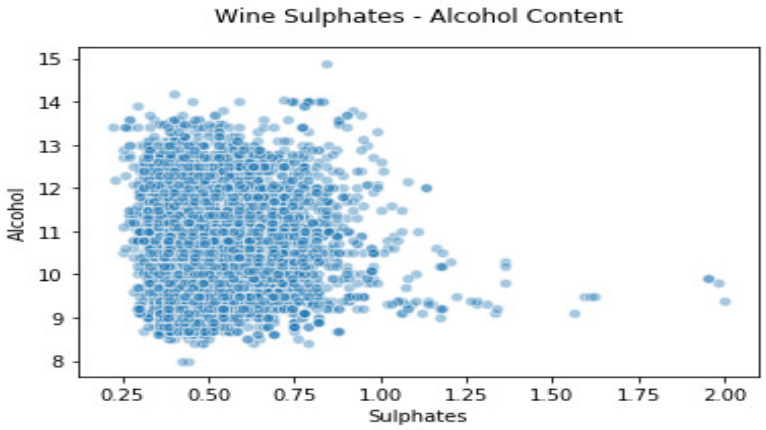
11. We will now visualize the same is to use pair-wise scatter plots amongst attributes of interest.

```
cols = ['density', 'residual sugar', 'total sulfur dioxide', 'fixed acidity']
pp = sns.pairplot(wines[cols], size=1.8, aspect=1.8,
                  plot_kws=dict(edgecolor="k", linewidth=0.5),
                  diag_kind="kde", diag_kws=dict(shade=True))

fig = pp.fig
fig.subplots_adjust(top=0.93, wspace=0.3)
t = fig.suptitle('Wine Attributes Pairwise Plots', fontsize=14)
```



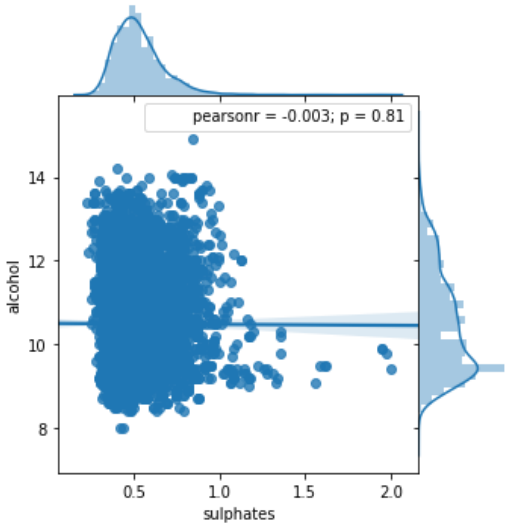
12. Let's look at some ways in which we can visualize two continuous, numeric attributes. Scatter plots and joint plots in particular are good ways to not only check for patterns, relationships but also see the individual distributions for the attributes.

```
#Two Continuous Numeric attributes
plt.scatter(wines['sulphates'], wines['alcohol'],
            alpha=0.4, edgecolors='w')

plt.xlabel('Sulphates')
plt.ylabel('Alcohol')
plt.title('Wine Sulphates - Alcohol Content',y=1.05)
```



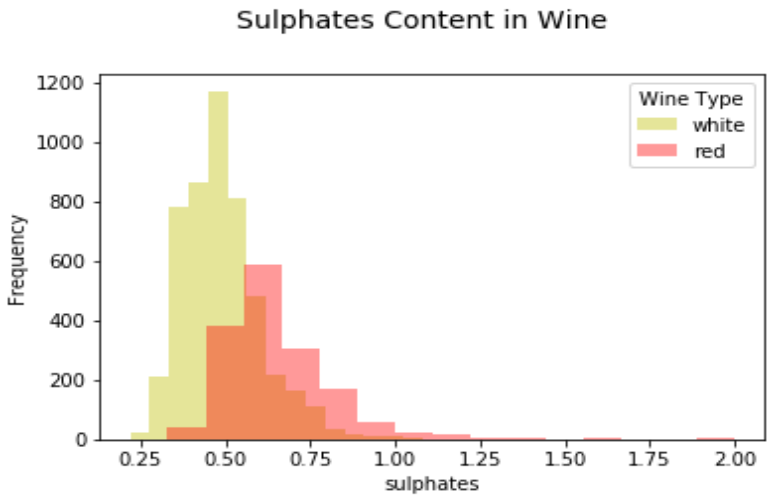
13.

```
jp = sns.jointplot(x='sulphates', y='alcohol', data=wines,
                   kind='reg', space=0, size=5, ratio=4)
```



14. Using multiple histograms, we will use box plots to effectively depict groups of numeric data based on the different values in the categorical attribute. Box plots are a good way to know the quartile values in the data and also potential outliers. We will use seaborn and even depict the plots in one single chart.

```
fig = plt.figure(figsize = (6, 4))
title = fig.suptitle("Sulphates Content in Wine", fontsize=14)
fig.subplots_adjust(top=0.85, wspace=0.3)
ax = fig.add_subplot(1,1, 1)
ax.set_xlabel("Sulphates")
ax.set_ylabel("Frequency")

g = sns.FacetGrid(wines, hue='wine_type', palette={"red": "r", "white": "y"})
g.map(sns.distplot, 'sulphates', kde=False, bins=15, ax=ax)
ax.legend(title='Wine Type')
plt.close(2)
```



You can see the plot generated above is clear and concise and we can easily compare across the distributions easily.

15. Visualizing data in Six Dimensions (6-D)

We will leverage depth, hue, size and shape besides our regular two axes to depict all the six data dimensions

Interpreting this might seem a bit taxing but consider a couple of components at a time when trying to understand what's going on.

1. Considering shape & y-axis, we have high and medium quality wines having higher alcohol levels as compared to low quality wines.
2. Considering hue and size, we have higher content of total sulfur dioxide for white wines as compared to red wines.
3. Considering depth and hue, we have white wines having lower fixed acidity levels as compared to red wines.
4. Considering hue and x-axis, we have red wines having lower levels of residual sugar as compared to white wines.
5. Considering hue and shape, white wines seem to have more high quality wines as compared to red wines (possibly due to larger sample size of white wines).