UDACITY Data Analysis Nanodegree

Project:- Wine Analysis

Learning about: Appending, Renaming Columns, Visuals, Pandas Groupby, Pandas Query

Grant Patience, 26th June 2019

Table of Contents

1. Problem understanding
   A. Outline of Steps
   B. What are the Desired Outputs
   C. What Questions Are We Trying to Answer?
   D. What Resources are Available?
2. Data Wrangling and Understanding
   A. Data Extraction
   B. Describe the Data's General Properties
   C. Verify Data Quality
      a. Missing Data
      b. Outliers
      c. Duplicates
      d. Data Quality Report
3. Exploratory Data Analysis
4. References

1. Determine Objectives and Assess the Situation

The task is to answer several questions regarding wine quality, using this data set
https://archive.ics.uci.edu/ml/datasets/Wine+Quality (https://archive.ics.uci.edu/ml/datasets/Wine+Quality)

1.1 Outline of Steps

- We discuss what it is we wish to achieve, and decide which questions we want to ask of the data
- We will extract the data we need
- Import the data into Python for analysis
- Perform some rudimentary exploratory data analysis to help understand our data
- Perform Exploratory Data Analysis
- Create visualisations to aid exploration
- Draw our conclusions based on the data

1.2 What are the desired outputs of the project?

- Accurate project submission
- Sucesfully answer all queries
- Learn about - Appending, Renaming Columns, Visuals, Pandas Groupby, Pandas Query

1.3 What Questions Are We Trying To Answer?

- How many samples of red wine are there?
- How many samples of white wine are there?
- How many columns are in each dataset?
- Which features have missing values?
- How many duplicate rows are in the white wine dataset?
- Are duplicate rows in these datasets significant/ need to be dropped?
- How many unique values of quality are in the red wine dataset?
- How many unique values of quality are in the white wine dataset?
- What is the mean density in the red wine dataset?
- Is a certain type of wine (red or white) associated with higher quality?
- What level of acidity (pH value) receives the highest average rating?
- Do wines with higher alcoholic content receive better ratings?
- Do sweeter wines (more residual sugar) receive better ratings
- What level of acidity receives the highest average rating?

1.4 What Resources are Available?

- dataset located at https://archive.ics.uci.edu/ml/datasets/Wine+Quality (https://archive.ics.uci.edu/ml/datasets/Wine+Quality)
- Jupyter Python Notebook

2. Data Wrangling and Understanding

The second stage of the process is where we acquire the data listed in the project resources. Describe the methods used to acquire them and any problems encountered. Record problems you encountered and any resolutions achieved. This initial collection includes extraction details and source details, and subsequently loaded into Python and analysed in Jupyter notebook.

2.1 Data Extraction

Simple download from https://archive.ics.uci.edu/ml/datasets/Wine+Quality (https://archive.ics.uci.edu/ml/datasets/Wine+Quality)

2.2 Describe Data's General Properties

Data description report - Describe the data that has been acquired including its format, its quantity (for example, the number of records and fields in each table), the identities of the fields and any other surface features which have been discovered. Evaluate whether the data acquired satisfies requirements.

In [54]:

```
import numpy as np
import pandas as pd
%matplotlib inline

import matplotlib.pyplot as plt

import seaborn as sns
sns.set_style('darkgrid')
```

In [55]:

```
df_r = pd.read_csv('winequality-red.csv', sep=';')
```

In [56]:

```
df_r.head(5)
```

Out[56]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alc
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	

In [57]:

```
df_w = pd.read_csv('winequality-white.csv', sep=';')
```

In [58]:

```
df_w.head(5)
```

Out[58]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alc
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	

In [59]:

```
df_r.shape
```

Out[59]:

```
(1599, 12)
```

In [60]:

```
df_r.columns
```

Out[60]:

```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual suga
r',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'densit
y',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [61]:

```
df_w.shape
```

Out[61]:

```
(4898, 12)
```

In [62]:

```
df_w.columns
```

Out[62]:

```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual suga
r',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'densit
y',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [63]:

```
df_r.dtypes
```

Out[63]:

```
fixed acidity           float64
volatile acidity        float64
citric acid             float64
residual sugar          float64
chlorides               float64
free sulfur dioxide     float64
total sulfur dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
quality                   int64
dtype: object
```

In [64]:

```
df_w.dtypes
```

Out[64]:

```
fixed acidity           float64
volatile acidity        float64
citric acid             float64
residual sugar          float64
chlorides               float64
free sulfur dioxide     float64
total sulfur dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
quality                   int64
dtype: object
```

In [65]:

```
df_r.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
fixed acidity           1599 non-null float64
volatile acidity        1599 non-null float64
citric acid             1599 non-null float64
residual sugar          1599 non-null float64
chlorides               1599 non-null float64
free sulfur dioxide     1599 non-null float64
total sulfur dioxide    1599 non-null float64
density                 1599 non-null float64
pH                      1599 non-null float64
sulphates               1599 non-null float64
alcohol                 1599 non-null float64
quality                 1599 non-null int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

In [66]:

```
df_w.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4898 entries, 0 to 4897
Data columns (total 12 columns):
fixed acidity           4898 non-null float64
volatile acidity        4898 non-null float64
citric acid             4898 non-null float64
residual sugar          4898 non-null float64
chlorides               4898 non-null float64
free sulfur dioxide     4898 non-null float64
total sulfur dioxide    4898 non-null float64
density                 4898 non-null float64
pH                      4898 non-null float64
sulphates               4898 non-null float64
alcohol                 4898 non-null float64
quality                 4898 non-null int64
dtypes: float64(11), int64(1)
memory usage: 459.3 KB
```

In [67]:

```
df_r.describe()
```

Out[67]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.(
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.4
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.8
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.(
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.(
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.(
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.(
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.(

In [68]:

```
df_w.describe()
```

Out[68]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.(
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.3
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.4
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.(
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.(
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.(
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.(
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.(

In [69]:

```
df_r.nunique()
```

Out[69]:

```
fixed acidity            96
volatile acidity        143
citric acid              80
residual sugar           91
chlorides               153
free sulfur dioxide      60
total sulfur dioxide    144
density                 436
pH                       89
sulphates                96
alcohol                  65
quality                   6
dtype: int64
```

In [70]:

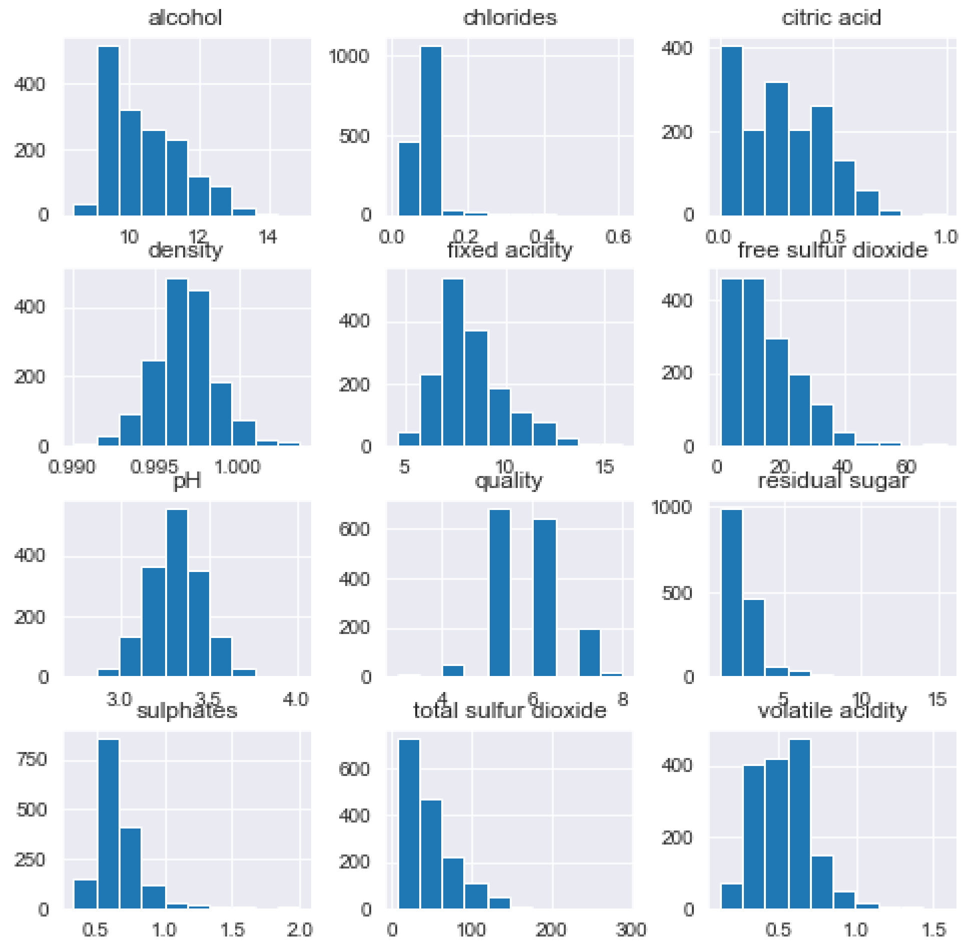
```
df_w.nunique()
```

Out[70]:

```
fixed acidity            68
volatile acidity        125
citric acid              87
residual sugar          310
chlorides               160
free sulfur dioxide     132
total sulfur dioxide    251
density                 890
pH                      103
sulphates                79
alcohol                 103
quality                   7
dtype: int64
```

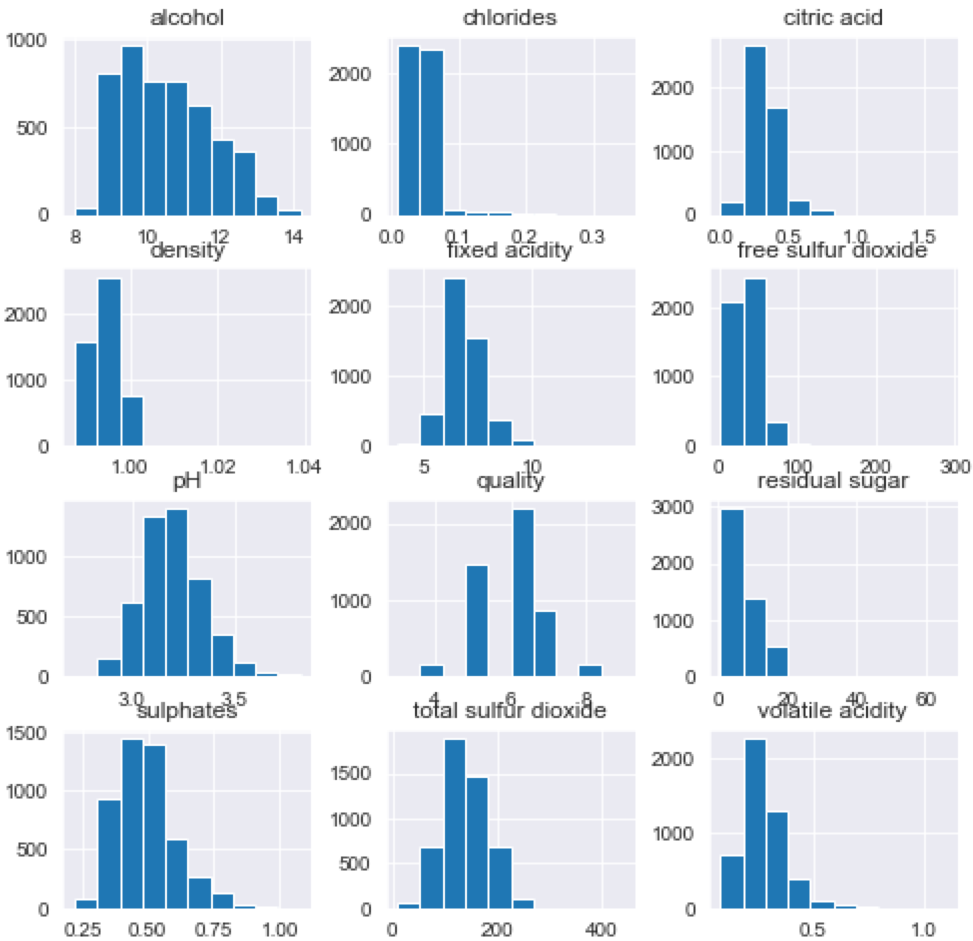
In [114]:

```
df_r.hist(figsize=( 8,8));
```



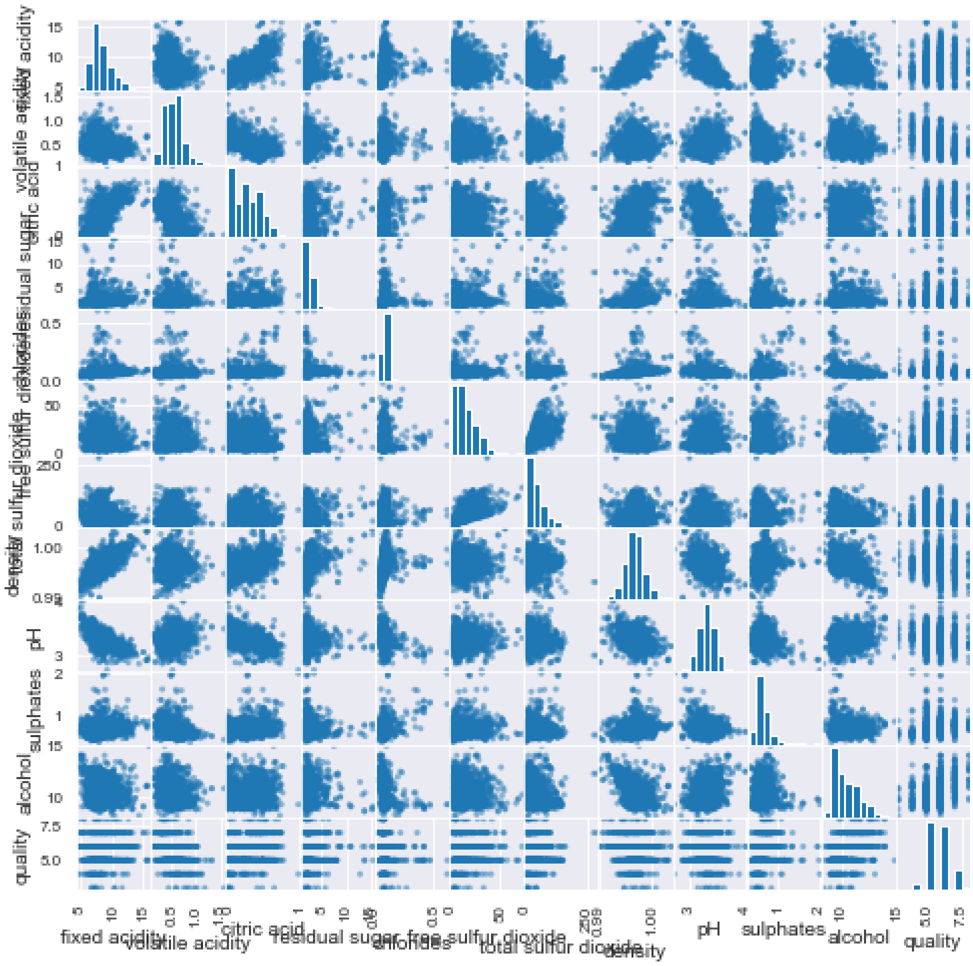
In [71]:

```
df_w.hist(figsize=(8,8));
```



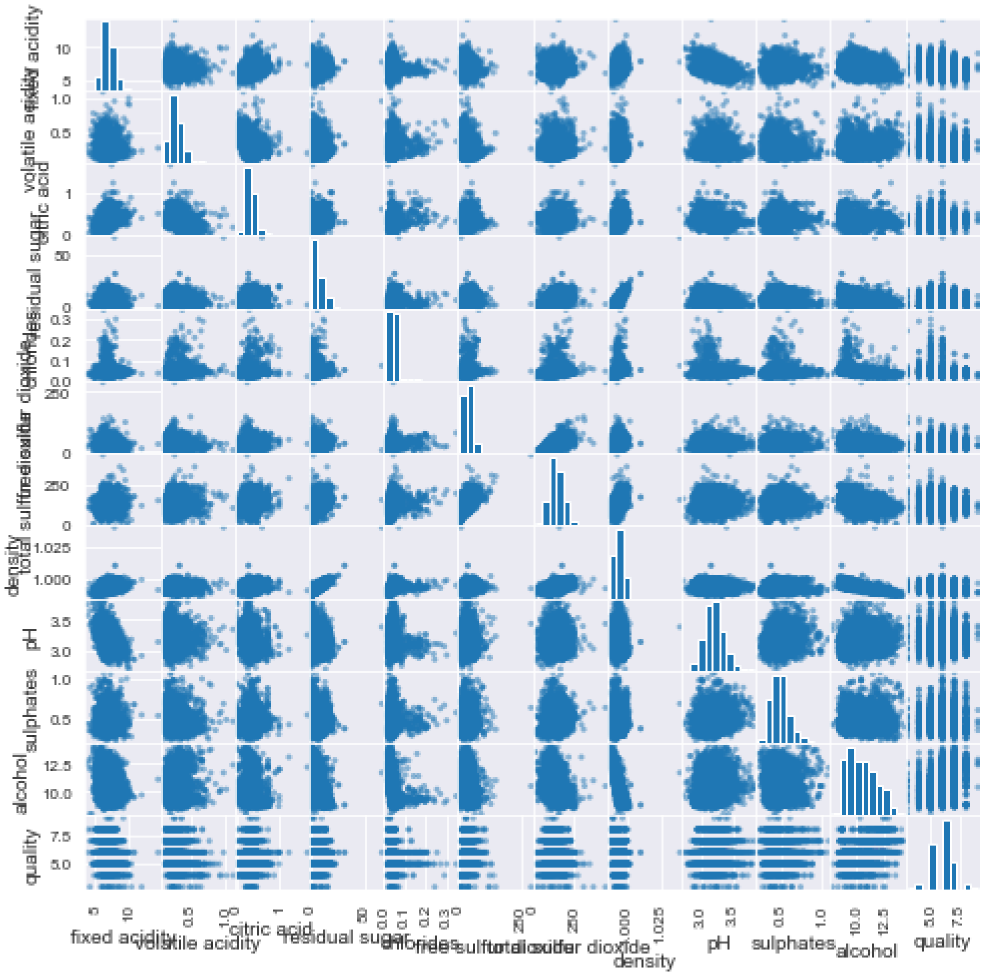
In [116]:

```
pd.plotting.scatter_matrix(df_r, figsize=(8,8));
```



In [117]:

```
pd.plotting.scatter_matrix(df_w, figsize=(8,8));
```



2.3 Verify Data Quality

Examine the quality of the data, addressing questions such as:

- Is the data complete (does it cover all the cases required)?
- Is it correct, or does it contain errors and, if there are errors, how common are they?
- Are there missing values in the data? If so, how are they represented, where do they occur, and how common are they?

2.3.1. Missing Data

In addition to incorrect datatypes, another common problem when dealing with real-world data is missing values. These can arise for many reasons and have to be either filled in or removed before we train a machine learning model. First, let's get a sense of how many missing values are in each column

While we always want to be careful about removing information, if a column has a high percentage of missing values, then it probably will not be useful to our model. The threshold for removing columns should depend on the problem

In [72]:

```
def missing_values_table(df):
        mis_val = df.isnull().sum()
        mis_val_percent = 100 * df.isnull().sum() / len(df)
        mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)
        mis_val_table_ren_columns = mis_val_table.rename(
        columns = {0 : 'Missing Values', 1 : '% of Total Values'})
        mis_val_table_ren_columns = mis_val_table_ren_columns[
            mis_val_table_ren_columns.iloc[:,1] != 0].sort_values(
        '% of Total Values', ascending=False).round(1)
        print ("Your selected dataframe has " + str(df.shape[1]) + " columns.\n"
            "There are " + str(mis_val_table_ren_columns.shape[0]) +
              " columns that have missing values.")
        return mis_val_table_ren_columns
```

In [73]:

```
df_r.isnull().sum()
```

Out[73]:

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

In [74]:

```
missing_values_table(df_r)
```

```
Your selected dataframe has 12 columns.
There are 0 columns that have missing values.
```

Out[74]:

Missing Values	% of Total Values

In [75]:

```
missing_values_table(df_w)
```

```
Your selected dataframe has 12 columns.
There are 0 columns that have missing values.
```

Out[75]:

Missing Values	% of Total Values

Decision

- We may want to remove null rows entirely from the dataset. To do so we would run the following

  ```
  df.dropna()
  ```

- We may want to drop the columns if they appear to be predominantly NA. To do so we would run the following

  ```
  # Get the columns with > 50% missing
  missing_df = missing_values_table(df);
  missing_columns = list(missing_df[missing_df['% of Total Values'] > 50].inde
  x)
  print('We will remove %d columns.' % len(missing_columns))
  df = df.drop(list(missing_columns))
  ```

- We may want to fill the missing values with the mean values from the dataset. To do so we would run the following

  ```
  mean = df['x'].mean()
  df['x'].fillna(mean, inplace=True)
  ```

2.3.2. Outliers

At this point, we may also want to remove outliers. These can be due to typos in data entry, mistakes in units, or they could be legitimate but extreme values. For this project, we will remove anomalies based on the definition of extreme outliers:

In [ ]:

```

```

2.3.3. Duplicates

There may be duplicates in the data. However, these may be legitimate new rows depending on the structure of the data. We need to discover them, then decide what to do with them

In [76]:

```
sum(df_r.duplicated())
```

Out[76]:

```
240
```

In [77]:

```
sum(df_w.duplicated())
```

Out[77]:

```
937
```

Decision We may want to remove duplicate rows entirely from the dataset. To do so we would run the following

```
df.drop_duplicates(inplace=True)
```

Data Quality Report

Category	Issue	Decision
Missing Values	N/A	None
Outliers	N/A	None
Duplicates	Duplicates found	None

---

3. Exploratory Data Analysis

- How many samples of red wine are there?

In [78]:

```
df_r.shape
```

Out[78]:

```
(1599, 12)
```

- How many samples of white wine are there?

In [79]:

```
df_w.shape
```

Out[79]:

```
(4898, 12)
```

- How many columns are in each dataset?

- Which features have missing values?

In [80]:

```
df_r.isnull().sum()
```

Out[80]:

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

In [81]:

```
df_w.isnull().sum()
```

Out[81]:

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

- How many duplicate rows are in the white wine dataset?

In [82]:

```
sum(df_r.duplicated())
```

Out[82]:

```
240
```

In [83]:

```
sum(df_w.duplicated())
```

Out[83]:

```
937
```

- Are duplicate rows in these datasets significant/ need to be dropped?

In [ ]:

```

```

- How many unique values of quality are in the red wine dataset?

In [84]:

```
df_r.nunique()
```

Out[84]:

```
fixed acidity            96
volatile acidity        143
citric acid              80
residual sugar           91
chlorides               153
free sulfur dioxide      60
total sulfur dioxide    144
density                 436
pH                       89
sulphates                96
alcohol                  65
quality                   6
dtype: int64
```

- How many unique values of quality are in the white wine dataset?

In [85]:

```
df_w.nunique()
```

Out[85]:

```
fixed acidity            68
volatile acidity        125
citric acid              87
residual sugar          310
chlorides               160
free sulfur dioxide     132
total sulfur dioxide    251
density                 890
pH                      103
sulphates                79
alcohol                 103
quality                   7
dtype: int64
```

- What is the mean density in the red wine dataset?

In [86]:

```
df_r.density.mean()
```

Out[86]:

```
0.9967466791744833
```

- Is a certain type of wine (red or white) associated with higher quality?

First, lets combine datasets

In [87]:

```
#Column name differences between the files, so change to a matching name
df_r=df_r.rename(columns = {'total_sulfur-dioxide':'total_sulfur_dioxide'})
```

In [88]:

```
# create color array for red dataframe
color_red = np.repeat('red', df_r.shape[0])

# create color array for white dataframe
color_white = np.repeat('white', df_w.shape[0])
```

In [89]:

```
df_r['color'] = color_red
df_w['color'] = color_white
```

In [90]:

```
wine_df = df_r.append(df_w)
```

In [91]:

```
wine_df.head()
```

Out[91]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alc
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	

In [92]:

```
wine_df=wine_df.rename(columns = {'residual sugar':'residual_sugar'})
```

In [93]:

```
# Find the mean quality of each wine type (red and white) with groupby
wine_df.groupby('color').mean().quality
```

Out[93]:

```
color
red      5.636023
white    5.877909
Name: quality, dtype: float64
```

- What level of acidity (pH value) receives the highest average rating?

In [94]:

```
wine_df.describe().pH
```

Out[94]:

```
count    6497.000000
mean        3.218501
std         0.160787
min         2.720000
25%         3.110000
50%         3.210000
75%         3.320000
max         4.010000
Name: pH, dtype: float64
```

In [95]:

```
# Bin edges that will be used to "cut" the data into groups
bin_edges = [2.72, 3.11, 3.21, 3.32, 4.01]
```

In [96]:

```
# Labels for the four acidity level groups
bin_names = ['high', 'mod_high', 'medium', 'low']
```

In [97]:

```
# Creates acidity_levels column
wine_df['acidity_levels'] = pd.cut(wine_df['pH'], bin_edges, labels=bin_names)

# Checks for successful creation of this column
wine_df.head()
```

Out[97]:

	fixed acidity	volatile acidity	citric acid	residual_sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphat
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.!
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.(
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.(
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.!
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.!

In [98]:

```
# Find the mean quality of each acidity level with groupby
wine_df.groupby('acidity_levels').mean().quality
```

Out[98]:

```
acidity_levels
high        5.783343
mod_high    5.784540
medium      5.850832
low         5.859593
Name: quality, dtype: float64
```

- Do wines with higher alcoholic content receive better ratings?

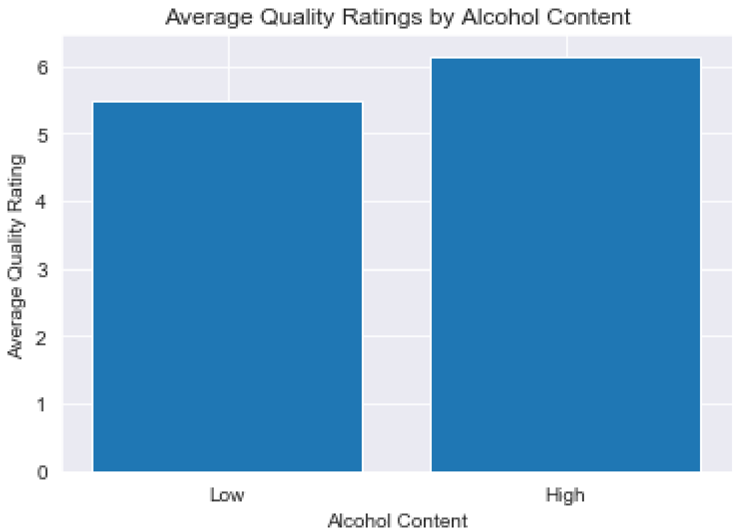
In [99]:

```
median = wine_df['alcohol'].median()
low = wine_df.query('alcohol < {}'.format(median))
high = wine_df.query('alcohol >= {}'.format(median))

mean_quality_low = low['quality'].mean()
mean_quality_high = high['quality'].mean()
```

In [100]:

```
locations = [1, 2]
heights = [mean_quality_low, mean_quality_high]
labels = ['Low', 'High']
plt.bar(locations, heights, tick_label=labels)
plt.title('Average Quality Ratings by Alcohol Content')
plt.xlabel('Alcohol Content')
plt.ylabel('Average Quality Rating');
```



- Do sweeter wines (more residual sugar) receive better ratings

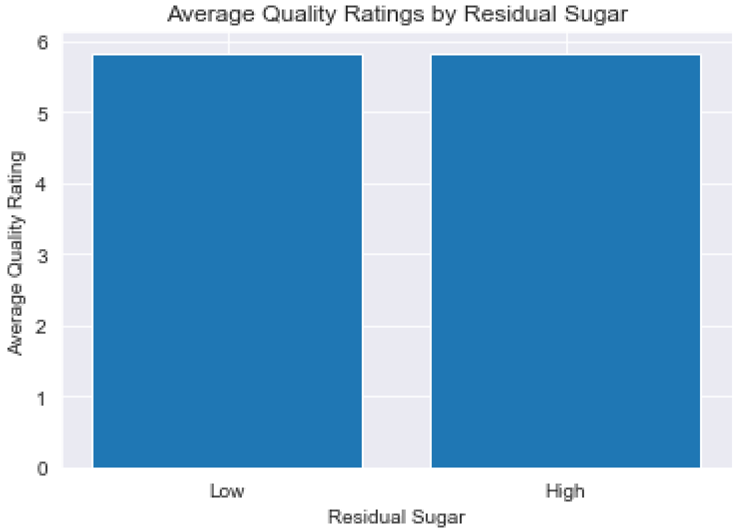
In [101]:

```
# Use query to select each group and get its mean quality
median = wine_df['residual_sugar'].median()
low = wine_df.query('residual_sugar < {}'.format(median))
high = wine_df.query('residual_sugar >= {}'.format(median))

mean_quality_low = low['quality'].mean()
mean_quality_high = high['quality'].mean()
```

In [102]:

```
# Create a bar chart with proper labels
locations = [1, 2]
heights = [mean_quality_low, mean_quality_high]
labels = ['Low', 'High']
plt.bar(locations, heights, tick_label=labels)
plt.title('Average Quality Ratings by Residual Sugar')
plt.xlabel('Residual Sugar')
plt.ylabel('Average Quality Rating');
```



- What level of acidity receives the highest average rating?

In [103]:

```
acidity_level_quality_means = wine_df.groupby('acidity_levels').mean().quality
acidity_level_quality_means
```

Out[103]:

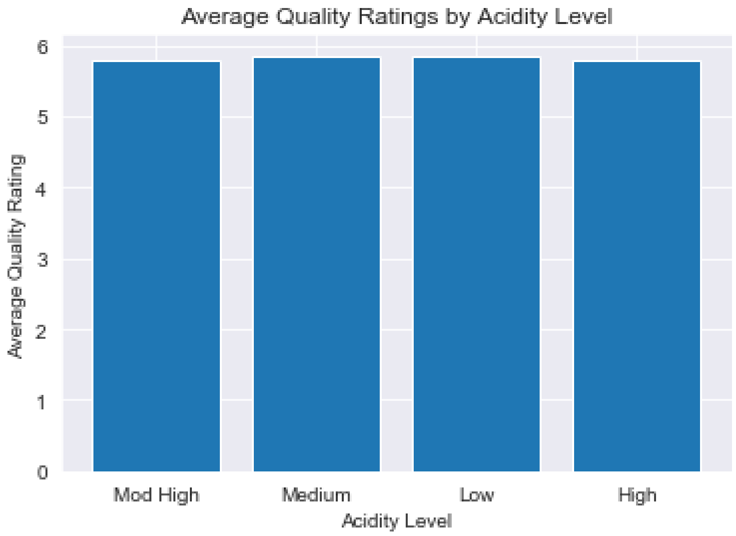
```
acidity_levels
high        5.783343
mod_high    5.784540
medium      5.850832
low         5.859593
Name: quality, dtype: float64
```

In [104]:

```
locations = [4, 1, 2, 3]  # reorder values above to go from low to high
heights = acidity_level_quality_means

# labels = ['Low', 'Medium', 'Moderately High', 'High']
labels = acidity_level_quality_means.index.str.replace('_', ' ').str.title() # alternat
ive to commented out line above

plt.bar(locations, heights, tick_label=labels)
plt.title('Average Quality Ratings by Acidity Level')
plt.xlabel('Acidity Level')
plt.ylabel('Average Quality Rating');
```



In [105]:

```
# get counts for each rating and color
color_counts = wine_df.groupby(['color', 'quality']).count()['pH']
color_counts
```

Out[105]:

```
color  quality
red    3            10
       4            53
       5           681
       6           638
       7           199
       8            18
white  3            20
       4           163
       5          1457
       6          2198
       7           880
       8           175
       9             5
Name: pH, dtype: int64
```

In [106]:

```
color_totals = wine_df.groupby('color').count()['pH']
color_totals
```

Out[106]:

```
color
red      1599
white    4898
Name: pH, dtype: int64
```

In [107]:

```
# get proportions by dividing red rating counts by total # of red samples
red_proportions = color_counts['red'] / color_totals['red']
red_proportions
```

Out[107]:

```
quality
3    0.006254
4    0.033146
5    0.425891
6    0.398999
7    0.124453
8    0.011257
Name: pH, dtype: float64
```

In [108]:

```
red_proportions['9'] = 0
red_proportions
```

Out[108]:

```
quality
3    0.006254
4    0.033146
5    0.425891
6    0.398999
7    0.124453
8    0.011257
9    0.000000
Name: pH, dtype: float64
```

In [109]:

```
white_proportions = color_counts['white'] / color_totals['white']
white_proportions
```

Out[109]:

```
quality
3    0.004083
4    0.033279
5    0.297468
6    0.448755
7    0.179665
8    0.035729
9    0.001021
Name: pH, dtype: float64
```

In [110]:

```
ind = np.arange(len(red_proportions))  # the x locations for the groups
width = 0.35       # the width of the bars
```

In [111]:

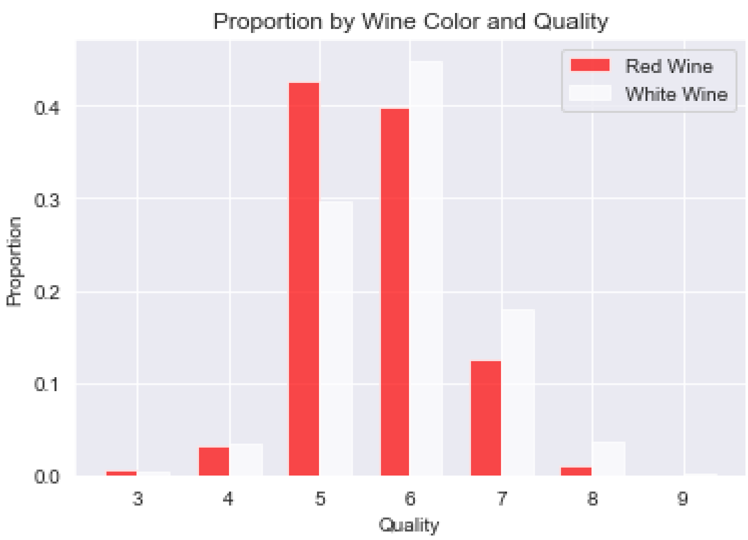
```
# plot bars
red_bars = plt.bar(ind, red_proportions, width, color='r', alpha=.7, label='Red Wine')
white_bars = plt.bar(ind + width, white_proportions, width, color='w', alpha=.7, label=
'White Wine')

# title and labels
plt.ylabel('Proportion')
plt.xlabel('Quality')
plt.title('Proportion by Wine Color and Quality')
locations = ind + width / 2  # xtick locations
labels = ['3', '4', '5', '6', '7', '8', '9']  # xtick labels
plt.xticks(locations, labels)

# legend
plt.legend()
```

Out[111]:

```
<matplotlib.legend.Legend at 0x2905d7d0a90>
```



4. Observations and Conclusion

4. Observations and Conclusion

- **How many samples of red wine are there?**
  - A - 1559
- **How many samples of white wine are there?**
  - A - 4898
- **How many columns are in each dataset?**
  - A - 12
- **Which features have missing values?**
  - A - None
- **How many duplicate rows are in the white wine dataset?**
  - A - 937
- **Are duplicate rows in these datasets significant/ need to be dropped?**
  - A - Not necessarily
- **How many unique values of quality are in the red wine dataset?**
  - A - 6
- **How many unique values of quality are in the white wine dataset?**
  - A - 7
- **What is the mean density in the red wine dataset?**
  - A - 0.996747
- **Is a certain type of wine (red or white) associated with higher quality?**
  - A - White
- **What level of acidity (pH value) receives the highest average rating?**
  - A - Low
- **Do wines with higher alcoholic content receive better ratings?**
  - A - High
- **Do sweeter wines (more residual sugar) receive better ratings?**
  - A - Yes
- **What level of acidity receives the highest average rating?**
  - A - Low

References

- UCI Wine Quality Data Set: [https://archive.ics.uci.edu/ml/datasets/Wine+Quality (https://archive.ics.uci.edu/ml/datasets/Wine+Quality)](https://archive.ics.uci.edu/ml/datasets/Wine+Quality)
- UDACITY Data Analyst Nanodegree: [https://eu.udacity.com/course/data-analyst-nanodegree--nd002?v=a (https://eu.udacity.com/course/data-analyst-nanodegree--nd002?v=a)](https://eu.udacity.com/course/data-analyst-nanodegree--nd002?v=a)