Data Analysis – Wine Quality Dataset

July 17, 2020

#

Udacity Data Analysis Nanodegree

##

Project: Wine Quality Dataset

###

Noaman Mangera, June 2020

0.1 Table of Contents

[1]:
```
#import modules
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
#display visuals in notebook
%matplotlib inline
```

[3]:
```
#set theme for visualisations
sns.set_style('darkgrid')
```

0.2 Data Wrangling

0.2.1 General Properties

[4]:
```
#read in red_wine dataset
df_r = pd.read_csv(r"C:\Users\noama\OneDrive\My␣
↪Documents\OneDrive\Python\Projects\Wine Quality\winequality-red.csv", sep=';
↪')
```

[5]:
```
#validate first 5 elements of red_wine dataset
df_r.head(5)
```

[5]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

[6]:
```
#read in white_wine dataset
df_w = pd.read_csv(r"C:\Users\noama\OneDrive\My␣
↪Documents\OneDrive\Python\Projects\Wine Quality\winequality-white.csv",␣
↪sep=';')
```

[7]:
```
#validate first 5 elements of white_wine dataset
df_w.head(5)
```

[7]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
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		

```
0      8.8        6
1      9.5        6
2     10.1        6
3      9.9        6
4      9.9        6
```

[8]:
```
#inspect number of columns and rows in the red_wine dataset
df_r.shape
```

[8]:
```
(1599, 12)
```

[9]:
```
#inspect name of columns in the red_wine dataset
df_r.columns
```

[9]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

[10]:
```
#inspect number of columns and rows in the white_wine dataset
df_w.shape
```

[10]:
```
(4898, 12)
```

[11]:
```
#inspect name of columns in the white_wine dataset
df_w.columns
```

[11]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

[12]:
```
#inspect data types of in the red_wine dataset
df_r.dtypes
```

[12]:
```
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
```

[13]:
```
#inspect data types of in the white_wine dataset
df_w.dtypes
```

[13]:
```
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
```

[14]:
```
#inspect summary information of ther red_wine dataset
df_r.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

[15]:
```
#inspect summary information of ther white_wine dataset
df_w.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4898 entries, 0 to 4897
Data columns (total 12 columns):
```

```
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         4898 non-null   float64
 1   volatile acidity      4898 non-null   float64
 2   citric acid           4898 non-null   float64
 3   residual sugar        4898 non-null   float64
 4   chlorides             4898 non-null   float64
 5   free sulfur dioxide   4898 non-null   float64
 6   total sulfur dioxide  4898 non-null   float64
 7   density               4898 non-null   float64
 8   pH                    4898 non-null   float64
 9   sulphates             4898 non-null   float64
 10  alcohol               4898 non-null   float64
 11  quality               4898 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 459.3 KB
```

[16]:
```
#generate summary statistics of numerical columns of red_wine dataset
df_r.describe()
```

[16]:

	fixed acidity	volatile acidity	citric acid	residual sugar	\
count	1599.000000	1599.000000	1599.000000	1599.000000	
mean	8.319637	0.527821	0.270976	2.538806	
std	1.741096	0.179060	0.194801	1.409928	
min	4.600000	0.120000	0.000000	0.900000	
25%	7.100000	0.390000	0.090000	1.900000	
50%	7.900000	0.520000	0.260000	2.200000	
75%	9.200000	0.640000	0.420000	2.600000	
max	15.900000	1.580000	1.000000	15.500000	

	chlorides	free sulfur dioxide	total sulfur dioxide	density	\
count	1599.000000	1599.000000	1599.000000	1599.000000	
mean	0.087467	15.874922	46.467792	0.996747	
std	0.047065	10.460157	32.895324	0.001887	
min	0.012000	1.000000	6.000000	0.990070	
25%	0.070000	7.000000	22.000000	0.995600	
50%	0.079000	14.000000	38.000000	0.996750	
75%	0.090000	21.000000	62.000000	0.997835	
max	0.611000	72.000000	289.000000	1.003690	

	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000
mean	3.311113	0.658149	10.422983	5.636023
std	0.154386	0.169507	1.065668	0.807569
min	2.740000	0.330000	8.400000	3.000000
25%	3.210000	0.550000	9.500000	5.000000
50%	3.310000	0.620000	10.200000	6.000000

```
75%       3.400000      0.730000     11.100000      6.000000
max       4.010000      2.000000     14.900000      8.000000
```

[17]:
```
#generate summary statistics of numerical columns of white_wine dataset
df_w.describe()
```

[17]:
```
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    4898.000000       4898.000000  4898.000000     4898.000000
mean        6.854788          0.278241     0.334192        6.391415
std         0.843868          0.100795     0.121020        5.072058
min         3.800000          0.080000     0.000000        0.600000
25%         6.300000          0.210000     0.270000        1.700000
50%         6.800000          0.260000     0.320000        5.200000
75%         7.300000          0.320000     0.390000        9.900000
max        14.200000          1.100000     1.660000       65.800000

         chlorides  free sulfur dioxide  total sulfur dioxide      density  \
count  4898.000000          4898.000000           4898.000000  4898.000000
mean      0.045772            35.308085            138.360657     0.994027
std       0.021848            17.007137             42.498065     0.002991
min       0.009000             2.000000              9.000000     0.987110
25%       0.036000            23.000000            108.000000     0.991723
50%       0.043000            34.000000            134.000000     0.993740
75%       0.050000            46.000000            167.000000     0.996100
max       0.346000           289.000000            440.000000     1.038980

                pH    sulphates      alcohol      quality
count  4898.000000  4898.000000  4898.000000  4898.000000
mean      3.188267     0.489847    10.514267     5.877909
std       0.151001     0.114126     1.230621     0.885639
min       2.720000     0.220000     8.000000     3.000000
25%       3.090000     0.410000     9.500000     5.000000
50%       3.180000     0.470000    10.400000     6.000000
75%       3.280000     0.550000    11.400000     6.000000
max       3.820000     1.080000    14.200000     9.000000
```

[18]:
```
#count number of unique values in red_wine dataset
df_r.nunique()
```

[18]:
```
fixed acidity           96
volatile acidity       143
citric acid             80
residual sugar          91
chlorides              153
free sulfur dioxide     60
total sulfur dioxide   144
density                436
```

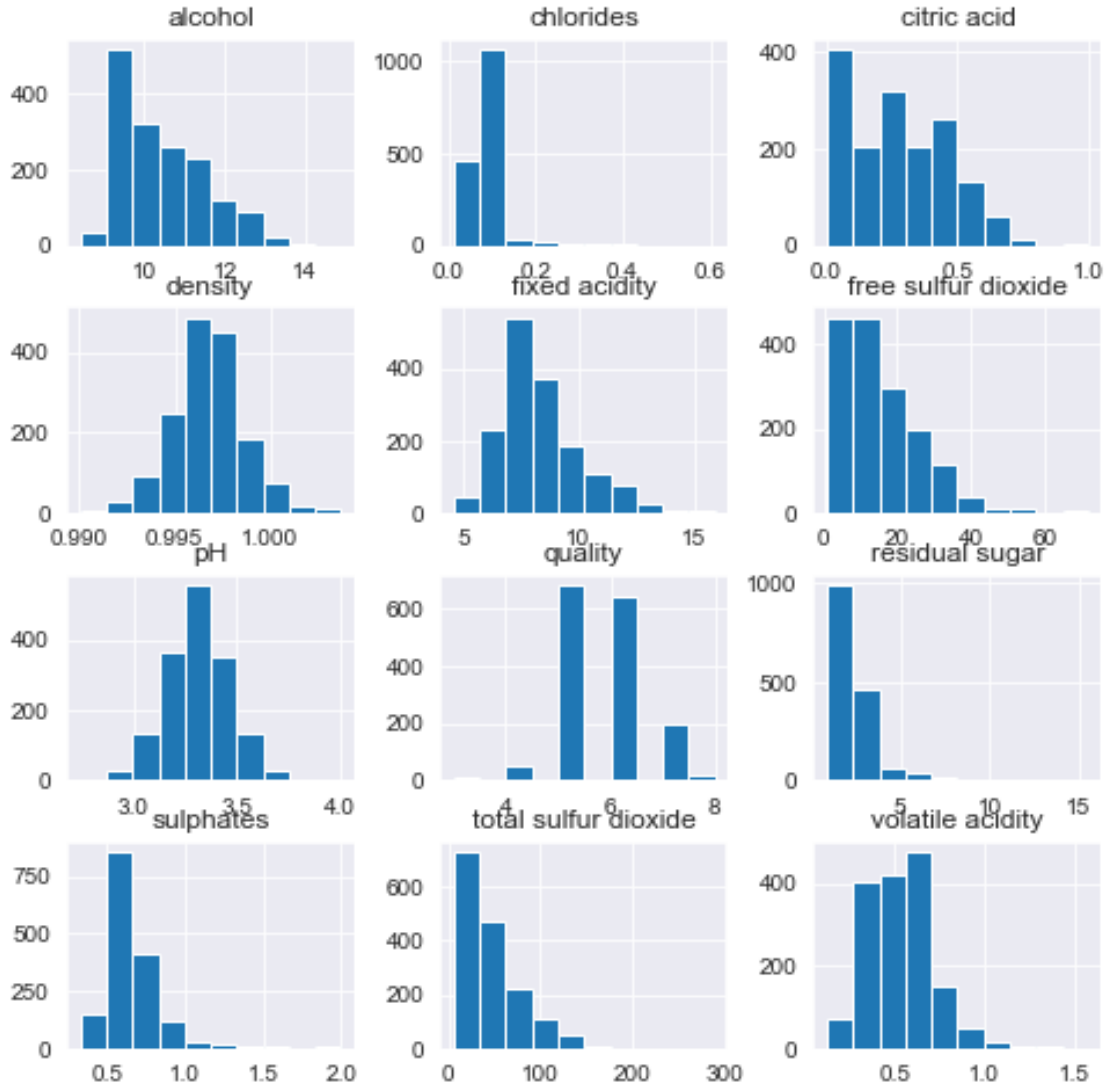
```
pH                      89
sulphates               96
alcohol                 65
quality                  6
dtype: int64
```

[19]:
```
#count number of unique values in white_wine dataset
df_w.nunique()
```

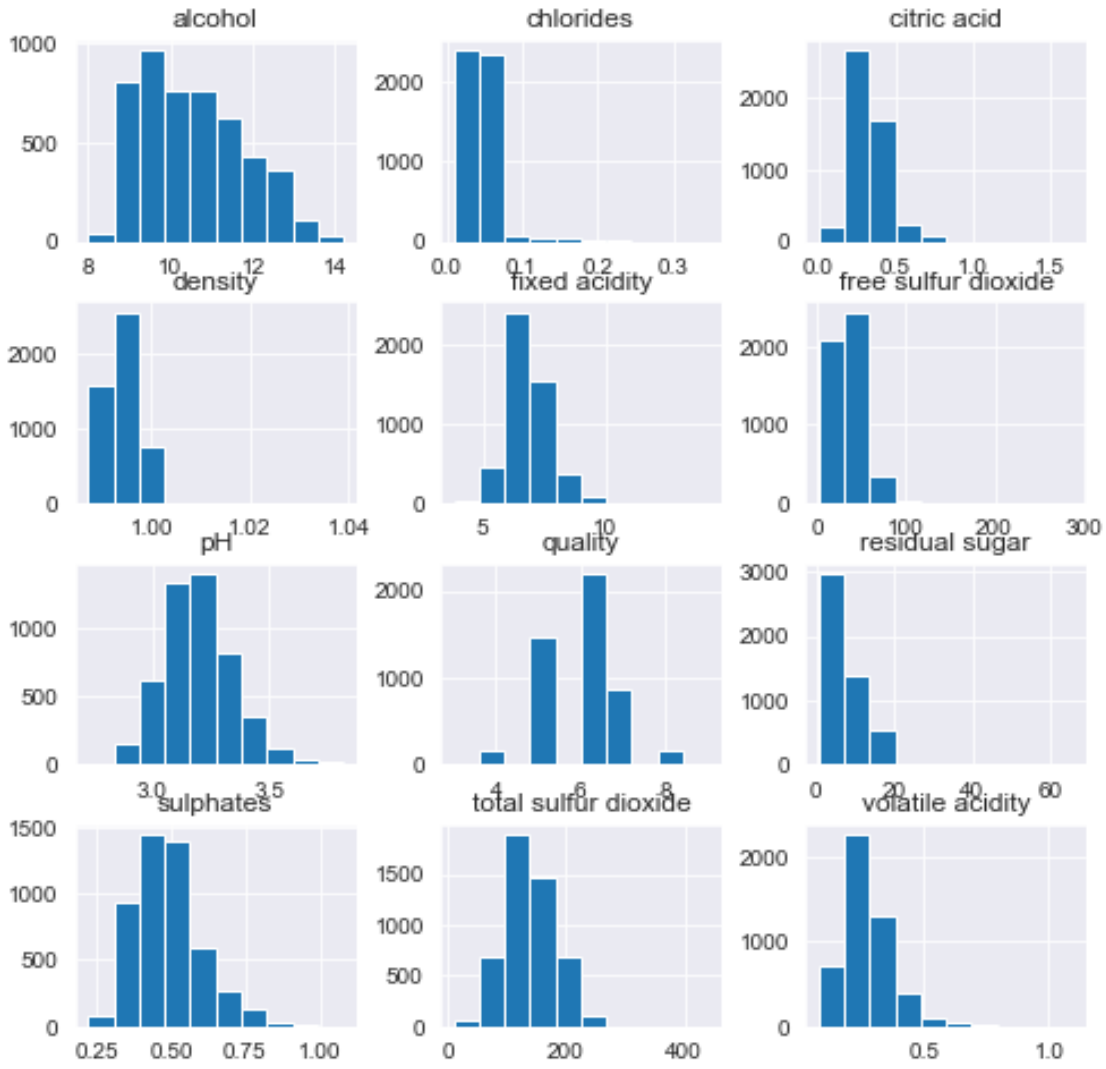
[19]:
```
fixed acidity           68
volatile acidity       125
citric acid             87
residual sugar         310
chlorides              160
free sulfur dioxide    132
total sulfur dioxide   251
density                890
pH                     103
sulphates               79
alcohol                103
quality                  7
dtype: int64
```

[20]:
```
#univarate analysis of varibales for red_wine dataset
df_r.hist(figsize=(8,8));
```

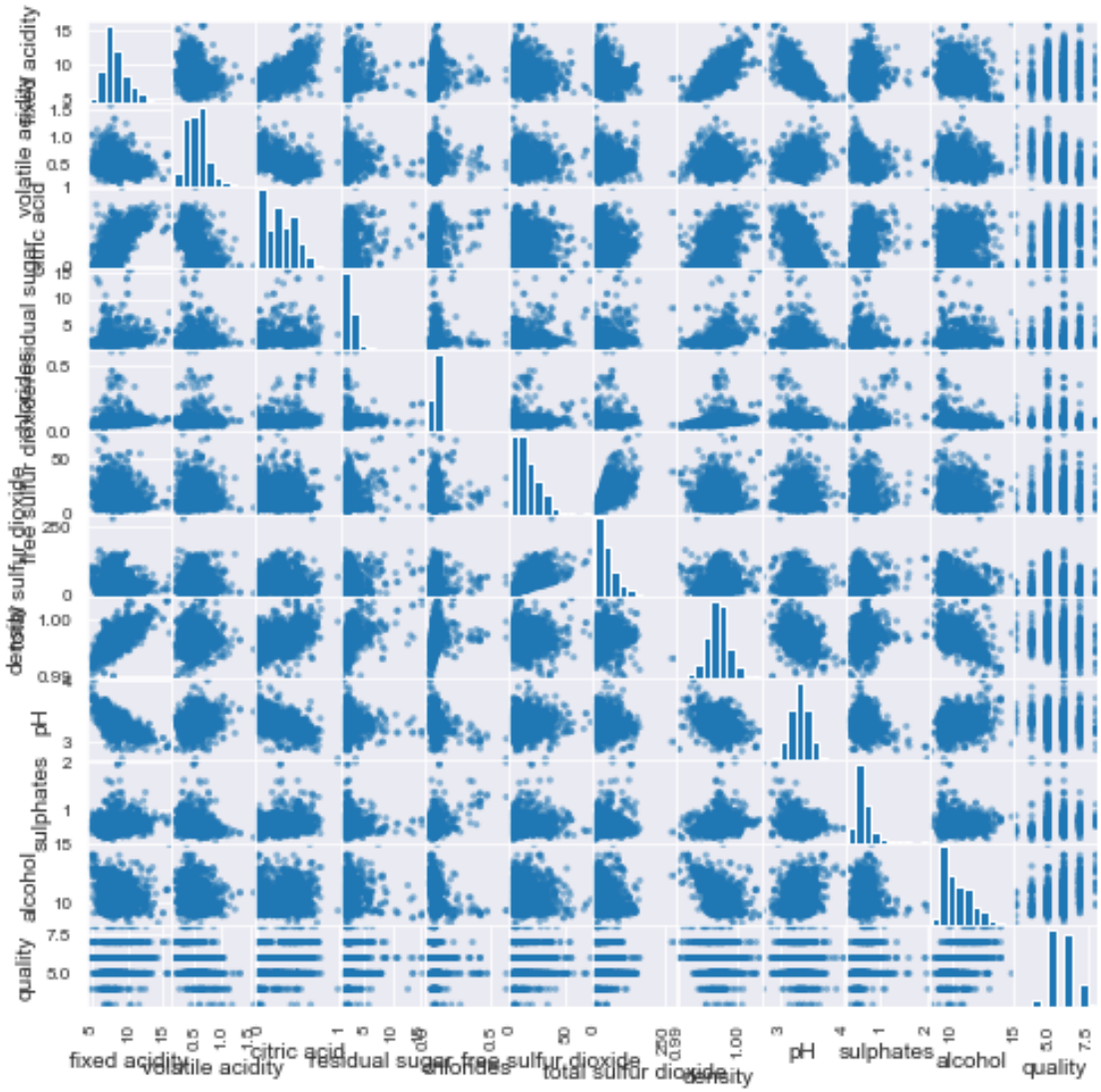
```
C:\Users\noama\anaconda3\lib\site-
packages\pandas\plotting\_matplotlib\tools.py:298: MatplotlibDeprecationWarning:
The rowNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().rowspan.start instead.
  layout[ax.rowNum, ax.colNum] = ax.get_visible()
C:\Users\noama\anaconda3\lib\site-
packages\pandas\plotting\_matplotlib\tools.py:298: MatplotlibDeprecationWarning:
The colNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().colspan.start instead.
  layout[ax.rowNum, ax.colNum] = ax.get_visible()
C:\Users\noama\anaconda3\lib\site-
packages\pandas\plotting\_matplotlib\tools.py:304: MatplotlibDeprecationWarning:
The rowNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().rowspan.start instead.
  if not layout[ax.rowNum + 1, ax.colNum]:
C:\Users\noama\anaconda3\lib\site-
packages\pandas\plotting\_matplotlib\tools.py:304: MatplotlibDeprecationWarning:
The colNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().colspan.start instead.
  if not layout[ax.rowNum + 1, ax.colNum]:
```



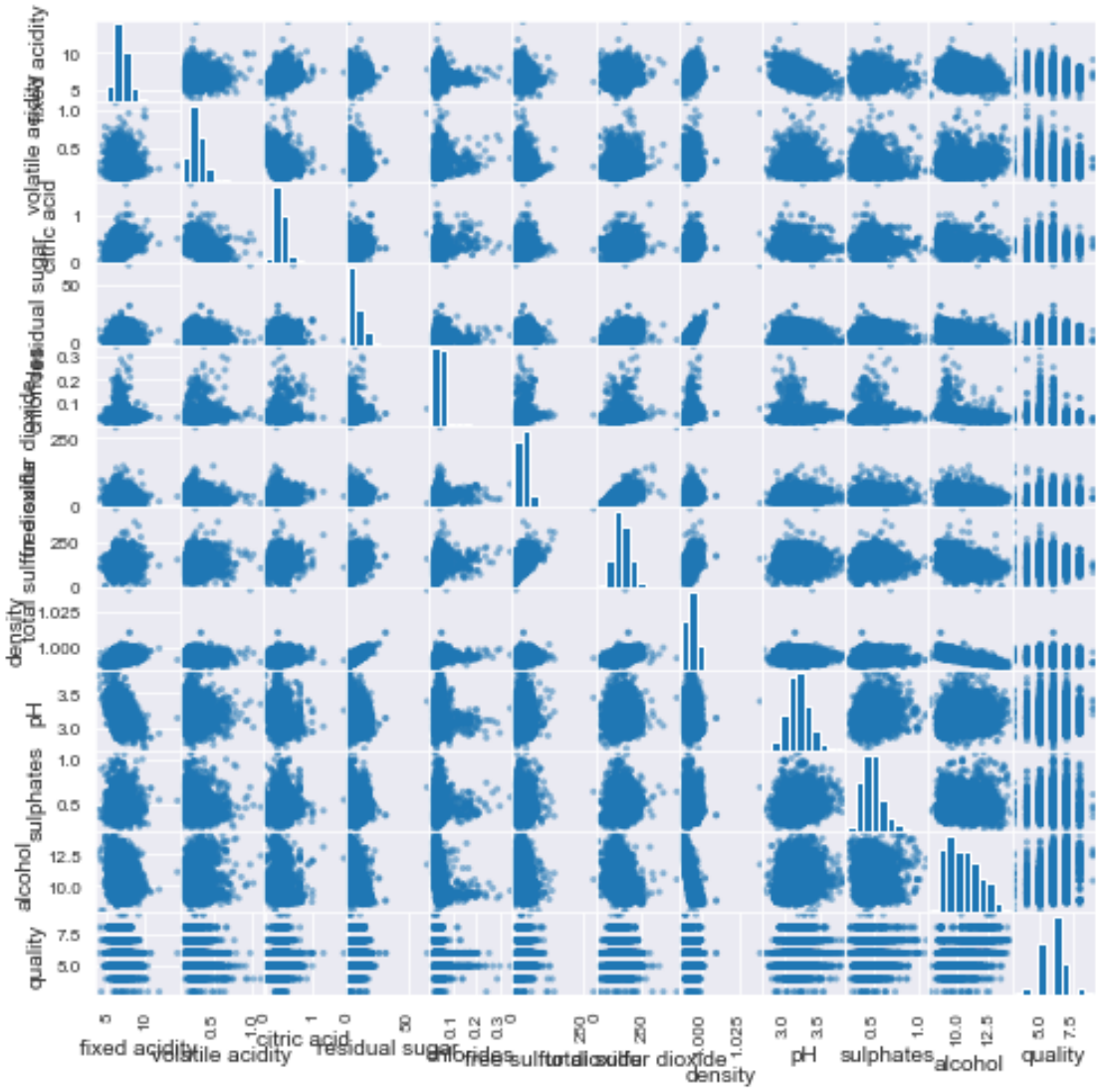
[21]:
```
#univarate analysis of varibales for white_wine dataset
df_w.hist(figsize=(8,8));
```



[22]:
```
#bivariate analysis of variables in the red_wine dataset
pd.plotting.scatter_matrix(df_r, figsize=(8,8));
```



[23]:
```
#bivariate analysis of variables in the white_wine dataset
pd.plotting.scatter_matrix(df_w, figsize=(8,8));
```



0.2.2 Verify Data Quality

Examine the quality of the data, addressing questions such as: >1. Is the data complete (does it cover all the cases required)?

2. Is it correct, or does it contain errors and, if there are errors, how common are they?

3. Are there missing values in the data? If so, how are they represented, where do they occur, and how common are they?

0.2.3 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.

[24]:
```
#create user define function to capture number of missing values
def missing_values(df):
    miss_val = df.isnull().sum()
    #print(miss_val)
    miss_val_percent = 100 * df.isnull().sum() / len(df)
    #print(miss_val_percent)
    miss_val_table = pd.concat([miss_val, miss_val_percent], axis=1)
    #print(miss_val_table)
    miss_val_table_rename_col = miss_val_table.rename(columns = {0 : 'Missing␣
 ↪Values', 1 : '% of Total Values'})
    #print(miss_val_table_rename_col)
    miss_val_table_sorted = miss_val_table_rename_col[miss_val_table_rename_col.
 ↪iloc[:,1] != 0].sort_values('% of Total Values', ascending=False).round(1)
    #print(miss_val_table_sorted)
    print ("Your selected dataframe has " + str(df.shape[1]) + " columns.\n"
           "There are " + str(miss_val_table_sorted.shape[0]) +
           " columns that have missing values.")
    return miss_val_table_sorted
```

[25]:
```
#apply misisng values function to red_wine dataset
missing_values(df_r)
```

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

[25]:
```
Empty DataFrame
Columns: [Missing Values, % of Total Values]
Index: []
```

[26]:
```
#apply misisng values function to white_wine dataset
missing_values(df_w)
```

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

[26]:
```
Empty DataFrame
Columns: [Missing Values, % of Total Values]
Index: []
```

[27]:
```
#inspect number of duplicates in red_wine dataset
sum(df_r.duplicated())
```

[27]:
```
240
```

[28]:
```
#inspect number of duplicates in white_wine dataset
sum(df_w.duplicated())
```

[28]:
```
937
```

[29]:
```
#How many samples and variables are there in the red wine dataset?
df_r.shape
```

[29]:
```
(1599, 12)
```

[30]:
```
#How many samples and variables are there in the white wine dataset?
df_w.shape
```

[30]:
```
(4898, 12)
```

[31]:
```
#which features have missing values in the red wine dataset?
df_r.isnull().sum()
```

[31]:
```
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
```

[32]:
```
#which features have missing values in the white wine dataset?
df_w.isnull().sum()
```

[32]:
```
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
```

0.3 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.

[33]:
```
#how many duplicate records are there in the red wine dataset?
df_r.duplicated().sum()
```

[33]:
```
240
```

[34]:
```
#how many duplicate records are there in the white wine dataset?
df_w.duplicated().sum()
```

[34]:
```
937
```

[35]:
```
#how many unique values of quality are there in the red wine dataset?
df_r.nunique()
```

[35]:
```
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
```

[36]:
```
#how many unique values of quality are there in the white wine dataset?
df_w.nunique()
```

[36]:
```
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
```

Exploratory Data Analysis

[37]:
```
#what is the mean density of the red wine dataset?
df_r['density'].mean()
```

[37]:
```
0.9967466791744833
```

[38]:
```
##is a certain type of wine (red or white) associated with higher quality?
#Column name differences between the files, so change to a matching name
df_r = df_r.rename(columns = {'total_sulfur-dioxide':'total_sulfur_dioxide'})

# 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])

#assign newly created coloured arrays to respective dataframes
df_r['color'] = color_red
df_w['color'] = color_white

#combine red wine dataset with white wine dataset
wine_df = df_r.append(df_w)
```

[39]:
```
#inspect first five rows of merged dataframe
wine_df.head()
```

[39]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	\
0	7.4	0.70	0.00	1.9	0.076	
1	7.8	0.88	0.00	2.6	0.098	
2	7.8	0.76	0.04	2.3	0.092	
3	11.2	0.28	0.56	1.9	0.075	
4	7.4	0.70	0.00	1.9	0.076	

	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	\
0	11.0	34.0	0.9978	3.51	0.56	
1	25.0	67.0	0.9968	3.20	0.68	
2	15.0	54.0	0.9970	3.26	0.65	
3	17.0	60.0	0.9980	3.16	0.58	
4	11.0	34.0	0.9978	3.51	0.56	

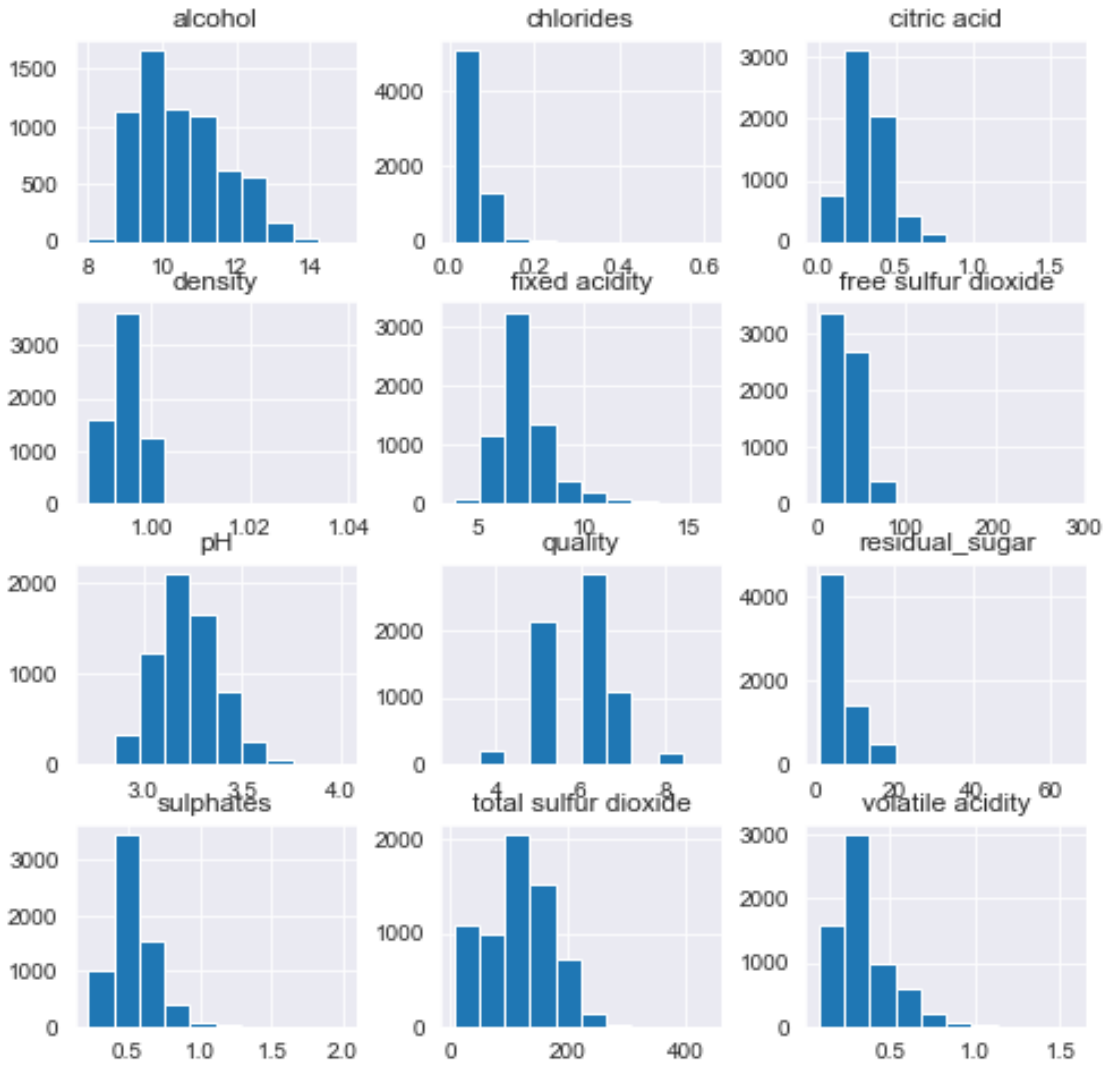
	alcohol	quality	color
0	9.4	5	red
1	9.8	5	red

```
2      9.8        5   red
3      9.8        6   red
4      9.4        5   red
```

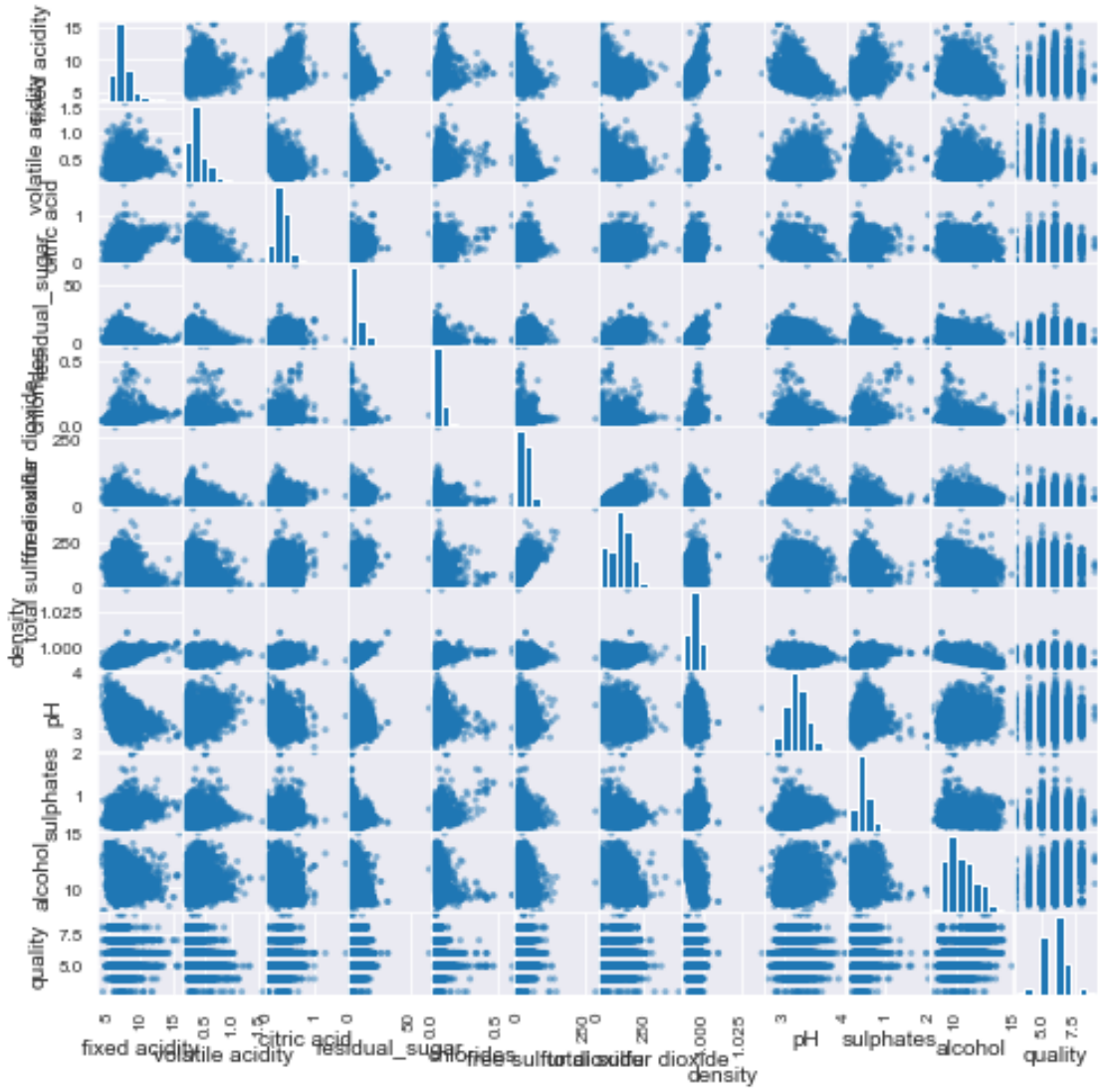
[40]:
```
#rename column
wine_df = wine_df.rename(columns = {'residual sugar':'residual_sugar'})
```

[41]:
```
#univarate analysis of varibales of merged dataset
wine_df.hist(figsize=(8,8));
```

```
C:\Users\noama\anaconda3\lib\site-
packages\pandas\plotting\_matplotlib\tools.py:298: MatplotlibDeprecationWarning:
The rowNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().rowspan.start instead.
  layout[ax.rowNum, ax.colNum] = ax.get_visible()
C:\Users\noama\anaconda3\lib\site-
packages\pandas\plotting\_matplotlib\tools.py:298: MatplotlibDeprecationWarning:
The colNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().colspan.start instead.
  layout[ax.rowNum, ax.colNum] = ax.get_visible()
C:\Users\noama\anaconda3\lib\site-
packages\pandas\plotting\_matplotlib\tools.py:304: MatplotlibDeprecationWarning:
The rowNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().rowspan.start instead.
  if not layout[ax.rowNum + 1, ax.colNum]:
C:\Users\noama\anaconda3\lib\site-
packages\pandas\plotting\_matplotlib\tools.py:304: MatplotlibDeprecationWarning:
The colNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().colspan.start instead.
  if not layout[ax.rowNum + 1, ax.colNum]:
```



[42]:
```
#bivariate analysis of variables in the merged dataset
pd.plotting.scatter_matrix(wine_df, figsize=(8,8));
```



[43]:
```
# Find the mean quality of each wine type (red and white) with groupby
wine_df.groupby('color').quality.mean()
```

[43]:
```
color
red      5.636023
white    5.877909
Name: quality, dtype: float64
```

[44]:
```
##what level of acidity (PH value) receives the highest average rating?
wine_df.pH.describe()
```

[44]:
```
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
```

[45]:
```
# Bin edges that will be used to "cut" the data into groups
bin_edges = [2.72, 3.11, 3.21, 3.32, 4.01]
```

[46]:
```
# Labels for the four acidity level groups
bin_names = ['high', 'mod_high', 'medium', 'low']
```

[47]:
```
# Creates acidity_levels column
wine_df['acidity_levels'] = pd.cut(wine_df['pH'], bin_edges, labels=bin_names)
```

[48]:
```
# Checks for successful creation of this column
wine_df.head()
```

[48]:

	fixed acidity	volatile acidity	citric acid	residual_sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	color	acidity_levels
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	red	low
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	red	mod_high
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	red	medium
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	red	mod_high
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	red	low

[49]:
```
# Find the mean quality of each acidity level with groupby
wine_df.groupby('acidity_levels').quality.mean()
```

[49]:
```
acidity_levels
high        5.783343
mod_high    5.784540
```

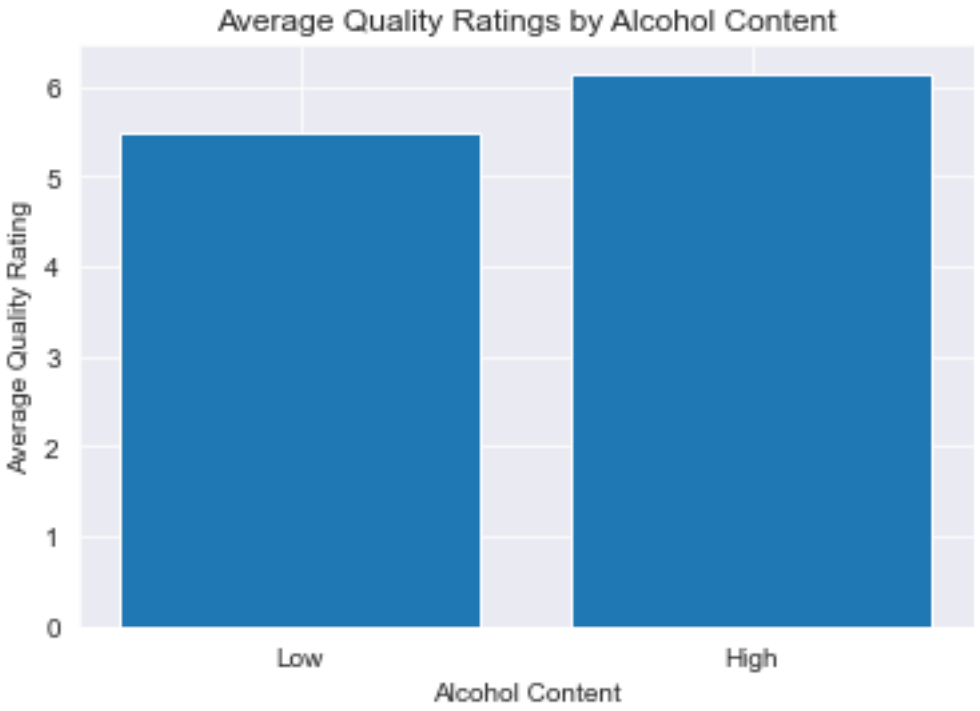
```
medium      5.850832
low         5.859593
Name: quality, dtype: float64
```

[50]:
```
##Do wines with higher alcohol content receive higher ratings?
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()
```

[51]:
```
#create bar chart with labels of preceeding analysis
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');
```



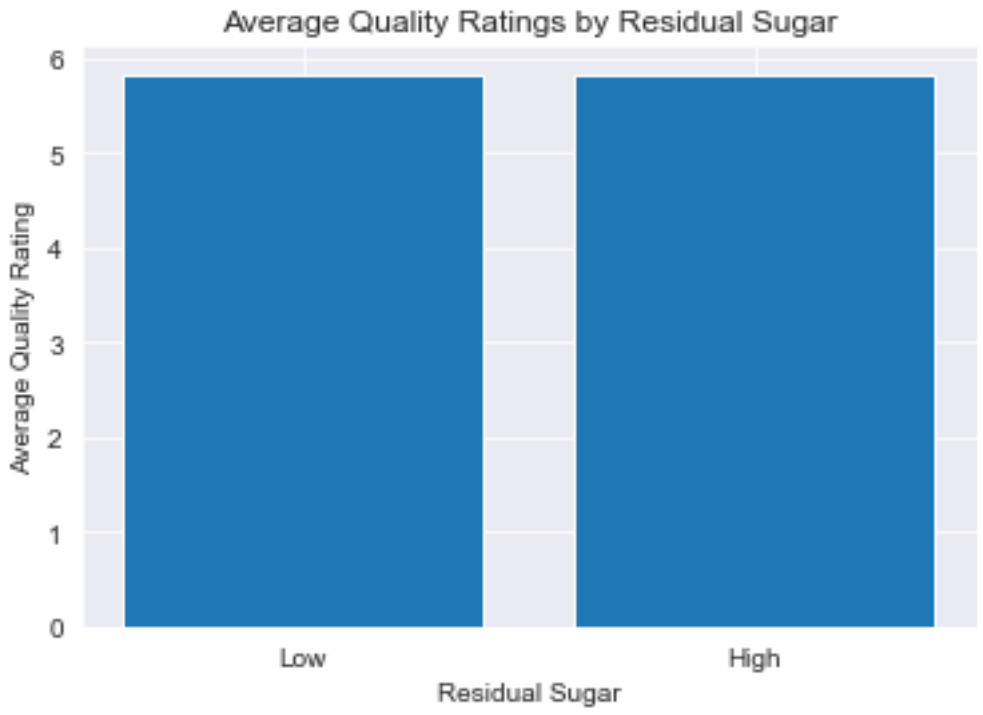
[52]:
```
##do sweeter wines (more resiudal sugar) recieve higher ratings?
# Use query to select each group and get its mean quality
median = wine_df['residual_sugar'].median()

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()
```

[53]:
```
#create bar chart with labels of preceeding analysis
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')
```

[53]:
```
Text(0, 0.5, 'Average Quality Rating')
```



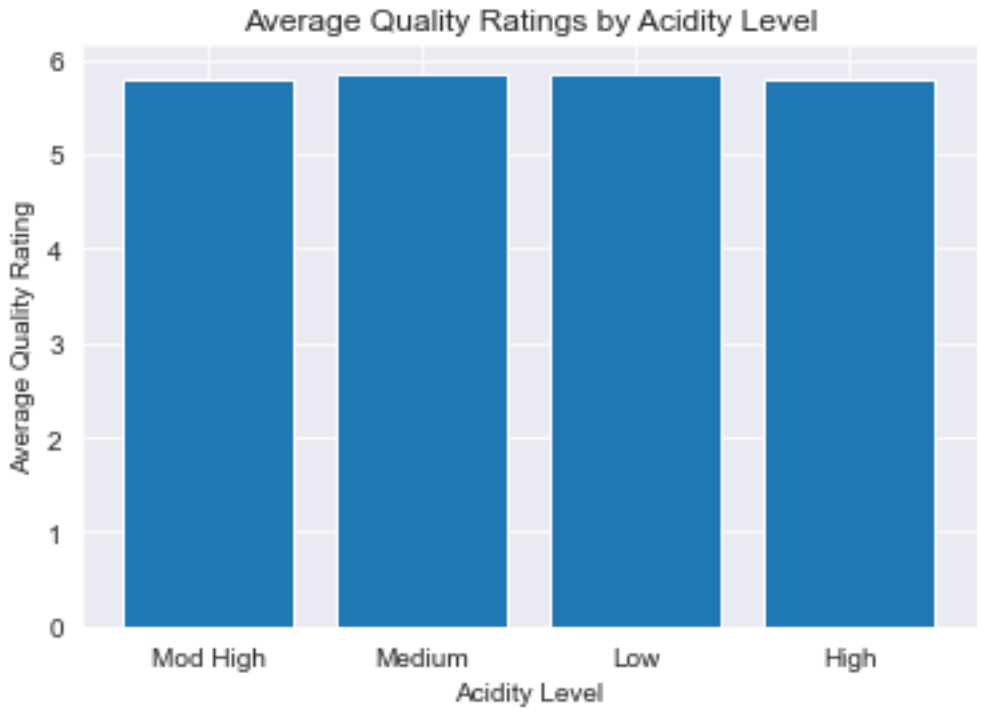
[54]:
```
##what level of acidity receives the highest average rating?
acidity_level_quality_means = wine_df.groupby('acidity_levels').mean().quality
acidity_level_quality_means
```

[54]:
```
acidity_levels
high        5.783343
mod_high    5.784540
medium      5.850832
low         5.859593
Name: quality, dtype: float64
```

[55]:
```
# reorder values above to go from low to high
locations = [4, 1, 2, 3]
heights = acidity_level_quality_means

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

#create bar chart with labels for title and axis
plt.bar(locations, heights, tick_label=labels)
plt.title('Average Quality Ratings by Acidity Level')
plt.xlabel('Acidity Level')
plt.ylabel('Average Quality Rating');
```



[56]:
```
# get counts for each rating and color
color_counts = wine_df.groupby(['color', 'quality']).count()['pH']
color_counts
```

[56]:
```
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
```

[57]:
```
#get count by color
color_totals = wine_df.groupby('color').count()['pH']
color_totals
```

[57]:
```
color
red      1599
white    4898
Name: pH, dtype: int64
```

[58]:
```
# get proportions by dividing red rating counts by total # of red samples
red_proportions = color_counts['red'] / color_totals['red']
red_proportions
```

[58]:
```
quality
3    0.006254
4    0.033146
5    0.425891
6    0.398999
7    0.124453
8    0.011257
Name: pH, dtype: float64
```

[59]:
```
#create additional layer of quality
red_proportions['9'] = 0
red_proportions
```

[59]:
```
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
```

[60]:
```
# get proportions by dividing white rating counts by total # of white samples
white_proportions = color_counts['white'] / color_totals['white']
white_proportions
```

[60]:
```
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
```

[61]:
```
# the x locations for the groups
ind = np.arange(len(red_proportions))

# the width of the bars
width = 0.35
```

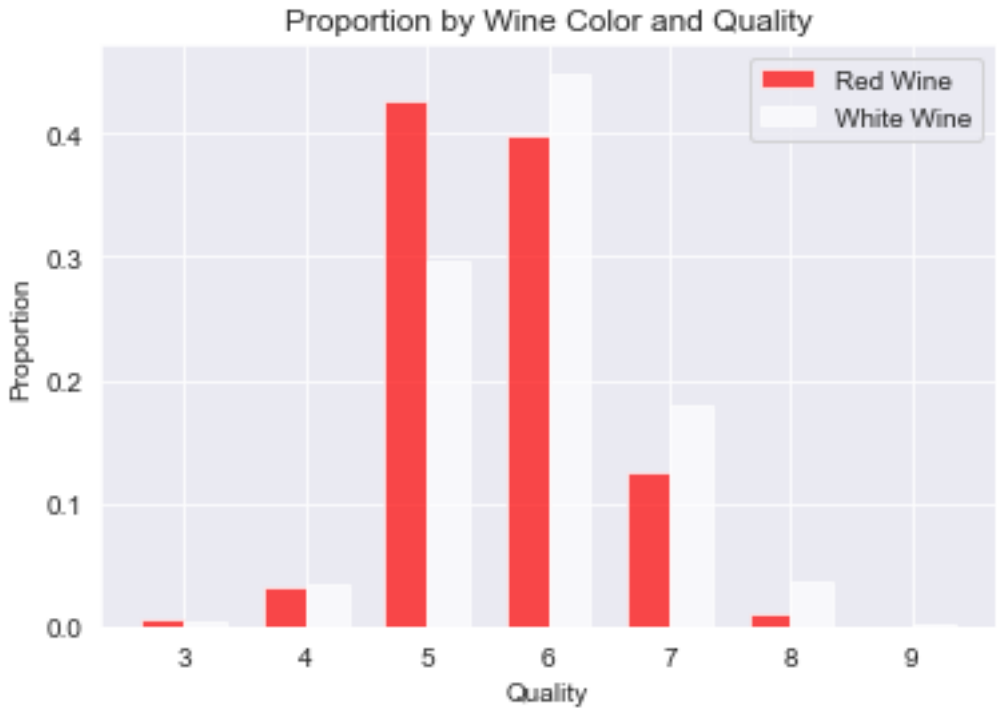
[62]:
```
# plot bars
red_bars = plt.bar(ind, red_proportions, width, color='r', alpha=.7, label='Red␣
 ↪Wine')
white_bars = plt.bar(ind + width, white_proportions, width, color='w', alpha=.
 ↪7, label= 'White Wine')

# title and labels
plt.ylabel('Proportion')
plt.xlabel('Quality')
plt.title('Proportion by Wine Color and Quality')

# xtick locations
locations = ind + width / 2
# xtick labels
labels = ['3', '4', '5', '6', '7', '8', '9']
plt.xticks(locations, labels)

# legend
plt.legend()
```

[62]:
```
<matplotlib.legend.Legend at 0x1c288df2248>
```



Conclusion

Question:How many samples of red wine are there? >Answer - 1559

Question:How many samples of white wine are there? >Answer - 4898

Question:How many columns are in each dataset? >Answer - 12

Question:Which features have missing values? >Answer - None

Question:How many duplicate rows are in the white wine dataset? >Answer - 937

Question: Are duplicate rows in these datasets significant/ need to be dropped? >Answer - Not necessarily

Question:How many unique values of quality are in the red wine dataset? >Answer - 6

Question:How many unique values of quality are in the white wine dataset? >Answer - 7

Question:What is the mean density in the red wine dataset? >Answer - 0.996747

Question:Is a certain type of wine (red or white) associated with higher quality? >Answer - White

Question:What level of acidity (pH value) receives the highest average rating? >Answer - Low

Do wines with higher alcoholic content receive better ratings? >Answer - High

Do sweeter wines (more residual sugar) receive better ratings? >Answer - Yes

Question:What level of acidity receives the highest average rating? >Answer: - Low