DS3110 - Data Wrangling

Individual Assignment

Question 01 - "Spirit Quality.csv" dataset

In [ ]:
```
import pandas as pd
data_file = pd.read_csv("D:/Uni/5 Semester/Data Wrangling/In class assignmnet/Sp
```

1. Display variables of the data set and their data types.

In [ ]:
```
for col in data_file.columns:
    print(data_file.dtypes)
```

```
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
wine ID                  object
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
quality index             int64
cost group               object
dtype: object
```

2. How many records are there in the dataset?

In [ ]:
```
print(data_file.shape[0])
```

```
4898
```

3. What is the highest fixed acidity value present in data?

In [ ]:
```
print(data_file['fixed acidity'].max())
```

```
14.2
```

4. Calculate the difference between mean fixed acidity and median volatile acidity.

In [ ]:
```
print(data_file['fixed acidity'].mean() - data_file['volatile acidity'].median()
```

```
6.604787668436097
```

5. Display the count of spirits that falls into each catergory of the quality variable.

In [ ]:
```
print(data_file['quality'].value_counts())
```

```
quality
6    2198
5    1457
7     880
8     175
4     163
3      20
9       5
Name: count, dtype: int64
```

6. Replace the values of the "quality index" variable and rename the as "quality categories".

In [ ]:
```
# Replace the values of the "quality index" variable
data_file['quality categories'] = data_file['quality index'].replace({
    0: 'bad',
    1: 'moderate',
    2: 'good'
})

#Rename the "quality index" column
data_file.rename(columns={'quality index': 'quality categories'}, inplace=True)

print(data_file)
```

```
     wine ID  fixed acidity  volatile acidity  citric acid  residual sugar  \
0       ID01            7.0              0.27         0.36            20.7   
1       ID02            6.3              0.30         0.34             1.6   
2       ID03            8.1              0.28         0.40             6.9   
3       ID04            7.2              0.23         0.32             8.5   
4       ID05            7.2              0.23         0.32             8.5   
...      ...            ...               ...          ...             ...   
4893  ID4894            6.2              0.21         0.29             1.6   
4894  ID4895            6.6              0.32         0.36             8.0   
4895  ID4896            6.5              0.24         0.19             1.2   
4896  ID4897            5.5              0.29         0.30             1.1   
4897  ID4898            6.0              0.21         0.38             0.8   

      chlorides  free sulfur dioxide  total sulfur dioxide  density    pH  \
0         0.045                 45.0                 170.0  1.00100  3.00   
1         0.049                 14.0                 132.0  0.99400  3.30   
2         0.050                 30.0                  97.0  0.99510  3.26   
3         0.058                 47.0                 186.0  0.99560  3.19   
4         0.058                 47.0                 186.0  0.99560  3.19   
...         ...                  ...                   ...      ...   ...   
4893      0.039                 24.0                  92.0  0.99114  3.27   
4894      0.047                 57.0                 168.0  0.99490  3.15   
4895      0.041                 30.0                 111.0  0.99254  2.99   
4896      0.022                 20.0                 110.0  0.98869  3.34   
4897      0.020                 22.0                  98.0  0.98941  3.26   

      sulphates  alcohol  quality quality categories            cost group  \
0          0.45      8.8        6                  1  moderately expensive   
1          0.49      9.5        6                  1  moderately expensive   
2          0.44     10.1        6                  1  moderately expensive   
3          0.40      9.9        6                  1  moderately expensive   
4          0.40      9.9        6                  1  moderately expensive   
...         ...      ...      ...                ...                   ...   
4893       0.50     11.2        6                  1  moderately expensive   
4894       0.46      9.6        5                  1  moderately expensive   
4895       0.46      9.4        6                  1  moderately expensive   
4896       0.38     12.8        7                  1    somewhat expensive   
4897       0.32     11.8        6                  1  moderately expensive   

     quality categories  
0              moderate  
1              moderate  
2              moderate  
3              moderate  
4              moderate  
...                 ...  
4893           moderate  
4894           moderate  
4895           moderate  
4896           moderate  
4897           moderate  

[4898 rows x 16 columns]
```

7. i) Group the data using "cost group" and calculate mean values for alcohol amount, residual sugar, chlorides and sulphates for each category in "cost group".

In [ ]:
```
data_file.groupby('cost group')[['alcohol', 'residual sugar', 'chlorides', 'sulp
```

Out[ ]:

cost group	alcohol	residual sugar	chlorides	sulphates
cheap	10.345000	6.392500	0.054300	0.474500
moderately expensive	10.264797	6.705107	0.047841	0.487069
somewhat expensive	11.412401	5.266919	0.038211	0.500303
very expensive	12.180000	4.120000	0.027400	0.466000

7. ii) Describe the relationship in categories of cost group variable and mean alcohol amount.

    - There is a positive correlation between cost group and mean alcohol amount which means when the cost of wine increase, the mean alcohol amount also increase.
    - The correlation is a weak positive correlation which means besides the cost of the wine, there are other facotrs which influece the mean alcohol amount in wine.
    - The cheap category has 10.35% mean alcohol amount while very expensive category has 12.18% mean alcohol amount.

8. Get the total acidity of spirits by adding "fixed acidity" and "volatile acidity" and store the values in a new column called "total acidity". Save the data set as a csv file having name as "Total acidity".

In [ ]:
```
# Get the total acidity
data_file["total acidity"] = data_file["fixed acidity"] + data_file["volatile ac

# Save as a csv file
data_file.to_csv("Total acidity.csv", index=False)

print(data_file['total acidity'])
```

```
0       7.27
1       6.60
2       8.38
3       7.43
4       7.43
       ...
4893    6.41
4894    6.92
4895    6.74
4896    5.79
4897    6.21
Name: total acidity, Length: 4898, dtype: float64
```

9. Get the records of spirits which has pH value less than 3.0 and Save the data set as a csv file having the name as “Low PH”.

In [ ]:
```
# Get the records of spirits which has pH value less than 3.0
low_ph_data = data_file[data_file["pH"] < 3.0]

# Save as a csv file
low_ph_data.to_csv("Low PH.csv", index=False)

print(low_ph_data)
```

```
     wine ID  fixed acidity  volatile acidity  citric acid  residual sugar  \
10      ID11            8.1              0.27         0.41            1.45
14      ID15            8.3              0.42         0.62           19.25
73      ID74            8.6              0.20         0.46            1.00
78      ID79            7.4              0.20         0.30            8.80
97      ID98            8.6              0.20         0.36            1.20
...      ...            ...               ...          ...             ...
4807  ID4808            6.0              0.17         0.30            7.30
4816  ID4817            6.1              0.41         0.20           12.60
4855  ID4856            7.1              0.23         0.39           13.70
4856  ID4857            7.1              0.23         0.39           13.70
4895  ID4896            6.5              0.24         0.19            1.20

      chlorides  free sulfur dioxide  total sulfur dioxide  density    pH  \
10        0.033                 11.0                  63.0  0.99080  2.99
14        0.040                 41.0                 172.0  1.00020  2.98
73        0.054                  9.0                  72.0  0.99410  2.95
78        0.064                 26.0                 103.0  0.99610  2.94
97        0.034                 15.0                  80.0  0.99130  2.95
...         ...                  ...                   ...      ...   ...
4807      0.039                 39.0                 104.0  0.99252  2.91
4816      0.032                 54.0                 136.0  0.99516  2.91
4855      0.058                 26.0                 172.0  0.99755  2.90
4856      0.058                 26.0                 172.0  0.99755  2.90
4895      0.041                 30.0                 111.0  0.99254  2.99

      sulphates  alcohol  quality quality categories            cost group  \
10         0.56     12.0        5                  1  moderately expensive
14         0.67      9.7        5                  1  moderately expensive
73         0.49      9.1        6                  1  moderately expensive
78         0.56      9.3        5                  1  moderately expensive
97         0.36     11.4        7                  1    somewhat expensive
...         ...      ...      ...                ...                   ...
4807       0.57     11.0        6                  1  moderately expensive
4816       0.43     10.6        6                  1  moderately expensive
4855       0.46      9.0        6                  1  moderately expensive
4856       0.46      9.0        6                  1  moderately expensive
4895       0.46      9.4        6                  1  moderately expensive

     quality categories  total acidity
10             moderate           8.37
14             moderate           8.72
73             moderate           8.80
78             moderate           7.60
97             moderate           8.80
...                 ...            ...
4807           moderate           6.17
4816           moderate           6.51
4855           moderate           7.33
4856           moderate           7.33
4895           moderate           6.74

[437 rows x 17 columns]
```

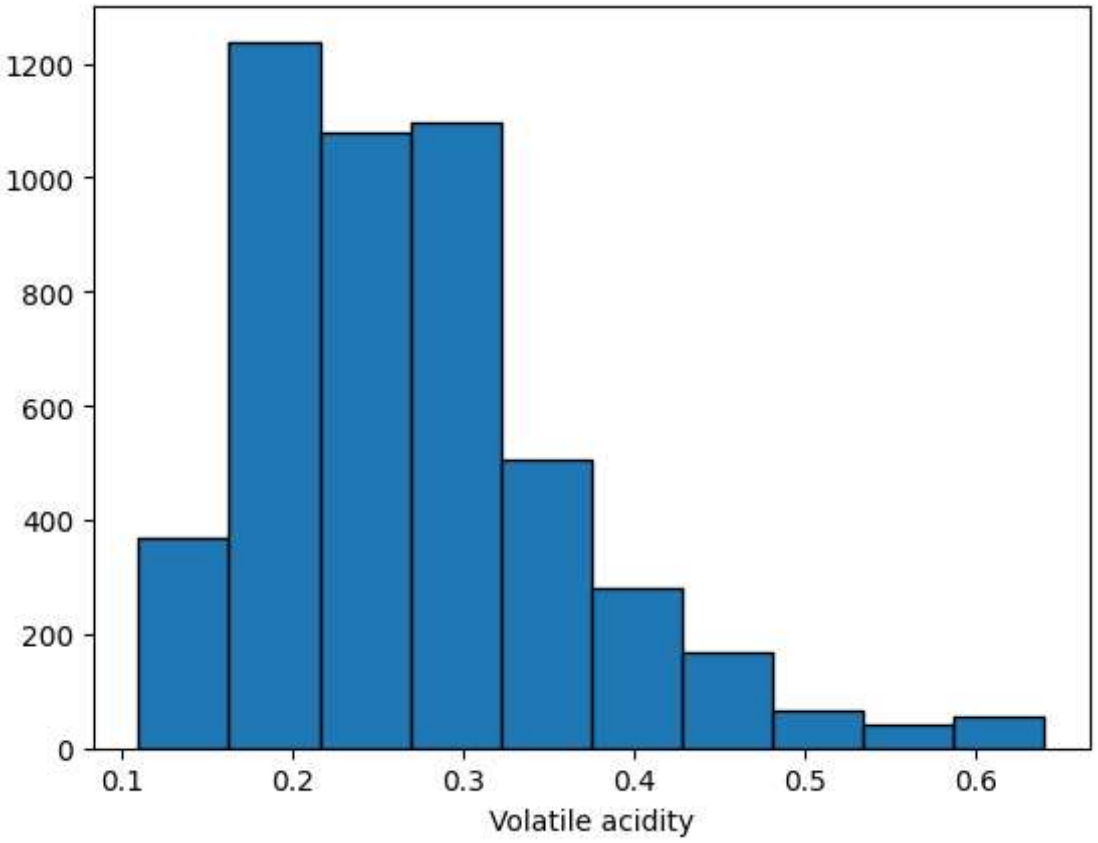
10. Get a subset of data with the columns; fixed acidity, volatile acidity, citric acid, free sulfur dioxide, total sulfur dioxide, sulphates and quality and Save the data set as a csv file having the name as “acid_and_sulpher”

In [ ]:
```
# Get the subset of the data with the columns
subset_data = data_file[["fixed acidity", "volatile acidity", "citric acid", "fr

# Save as a csv file
subset_data.to_csv("acid_and_sulpher.csv", index=False)

print(subset_data)
```

```
      fixed acidity  volatile acidity  citric acid  free sulfur dioxide  \
0               7.0              0.27         0.36                 45.0
1               6.3              0.30         0.34                 14.0
2               8.1              0.28         0.40                 30.0
3               7.2              0.23         0.32                 47.0
4               7.2              0.23         0.32                 47.0
...             ...               ...          ...                  ...
4893            6.2              0.21         0.29                 24.0
4894            6.6              0.32         0.36                 57.0
4895            6.5              0.24         0.19                 30.0
4896            5.5              0.29         0.30                 20.0
4897            6.0              0.21         0.38                 22.0

      total sulfur dioxide  sulphates  quality
0                    170.0       0.45        6
1                    132.0       0.49        6
2                     97.0       0.44        6
3                    186.0       0.40        6
4                    186.0       0.40        6
...                    ...        ...      ...
4893                  92.0       0.50        6
4894                 168.0       0.46        5
4895                 111.0       0.46        6
4896                 110.0       0.38        7
4897                  98.0       0.32        6

[4898 rows x 7 columns]
```

11. Draw a suitable graph to display the distribution of “volatile acidity” with suitable axis labels.

In [ ]:
```
import matplotlib.pyplot as plt

plt.hist(data_file["volatile acidity"], bins=10, edgecolor='black')
plt.xlabel("Volatile acidity")
plt.show()
```



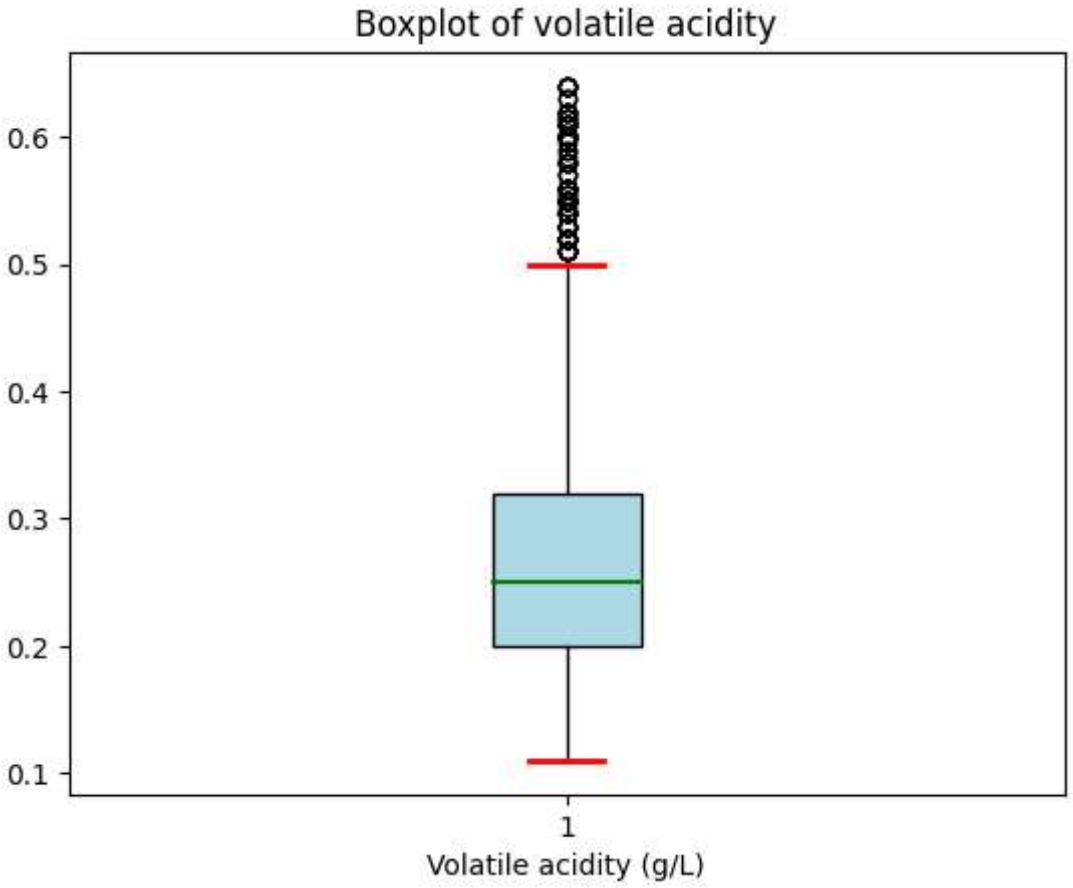
12. i) Which data problem you see in the distribution of "volatile acidity"?

- The distribution of volatile acidity is not normally disributed. It is right-skewed. Which means there are more values in right side of the distribution.
- High volatile acidity in wine is not a good thing this can be happen due to many reasons like;
  - Less sanitation practices.
  - Use ingredients with high level of acetic acid.
  - Oxygen overexposure and so on.

12. ii) Using a suitable method, display the entries having the identified issue.

- Boxplot can be used to show that the entries having right-skewness issue.
- It shows the five number summary of the distribution of volatile acidity of wine (median, quartile 1, quatile 3, min and max) and also the outliers.
- As shown in the boxplot the distribution is a righ-skewed with a median of approximately 2.5 g/L.
- The black circles in the boxplot shows the outliers, the values that fall outside of the inter-quartile range. Those outliers are in falls outside of the upperlimit which means that there are some wine which has more volatile acidity than the normal limit.

In [ ]:
```
plt.boxplot(data_file['volatile acidity'],
            patch_artist = True,
            boxprops = dict(facecolor = "lightblue"),
            medianprops = dict(color = "green", linewidth = 1.5),
            capprops = dict(color = "red", linewidth = 2))
plt.xlabel('Volatile acidity (g/L)')
plt.title('Boxplot of volatile acidity')
plt.show()
```



12. iii) What do you suggest to overcome the problem?

- To overcome this right-skewness issue, we can do to several things like;
  - Collect more data which helps to balance and make the distribution approximately normal.
  - Improve winemaking practices to reduce the amount of volatile acidity in wines.
  - At the same time we should remove these outliers by taking steps to reduce the acidity since it is unsafe to have more volatile acidity than the recommended limit.

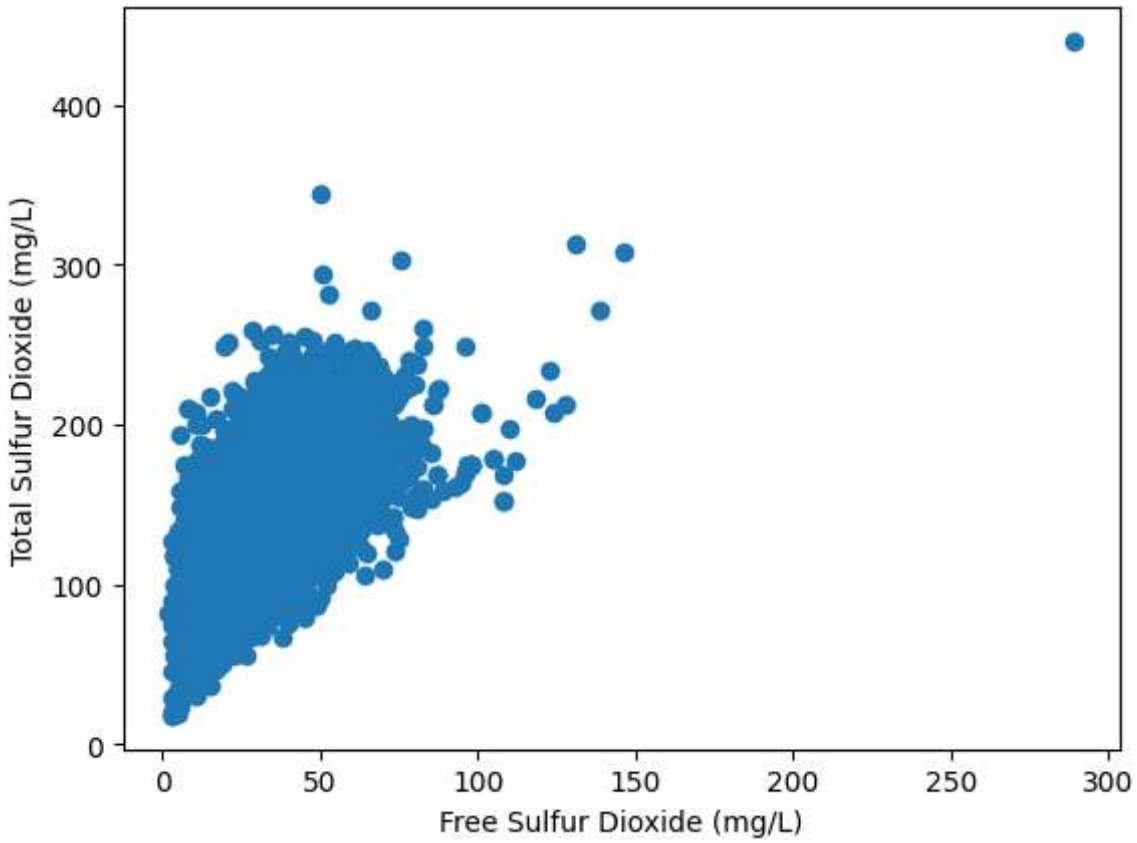
12. iv) Perform your suggestion and store the new ...volatile acidity values in a new column called, "volatile_acidity_new". Save the data as a csv file having the name as "cleaned_data".

In [ ]:
```
import numpy as np
```

```
# Define upper and lower limit for outliers
q1 = np.percentile(data_file['volatile acidity'], 25)
q3 = np.percentile(data_file['volatile acidity'], 75)
iqr = q3 - q1
upper_lim = np.percentile(data_file['volatile acidity'], 75) + 1.5 * iqr
lower_lim = np.percentile(data_file['volatile acidity'], 25) - 1.5 * iqr

# Outliers
outliers = np.where((data_file['volatile acidity'] < lower_lim) | (data_file['vc

# Remove outliers from the data set
data_file = data_file.drop(outliers, axis=0)

# Create new column
data_file['volatile_acidity_new'] = data_file['volatile acidity']

# Save the data as a csv file
data_file.to_csv('cleaned_data.csv', index=False)
```

13. i) Draw a suitable graph to display the relationship between “free sulfur dioxide” and “total sulfur dioxide” with suitable axis labels.

In [ ]:
```
plt.scatter(data_file['free sulfur dioxide'], data_file['total sulfur dioxide'])
plt.xlabel('Free Sulfur Dioxide (mg/L)')
plt.ylabel('Total Sulfur Dioxide (mg/L)')
plt.show()
```



13. ii) What conclusions can you make on this?

- The above satter plot shows a positive relationship between free sulfur dioxide and total sulfur dioxide which means when free sulfur dioxide increase, total

sulfur dioxide also increase.

14. i) Merge "Spirit_Quality.csv" and "Spirit_QualityNew.csv" such that the data loss is the least and the data loss is the most with suitable names for the merged data.

In [ ]:
```
import csv

# least data lost
SQ_data_l = pd.read_csv(r"D:/Uni/5 Semester/Data Wrangling/In class assignmnet/S
SQ_new_data_l = pd.read_csv(r"D:/Uni/5 Semester/Data Wrangling/In class assignmn

def merge_csv_files(SQ_data_l, SQ_new_data_l, output_file_l):
    with open(SQ_data_l, "r") as fl1, open(SQ_new_data_l, "r") as fl2, open(outp
        reader_fl1 = csv.reader(fl1)
        reader_fl2 = csv.reader(fl2)
        writer_fl3 = csv.writer(fl3)

        for row1, row2 in zip(reader_fl1, reader_fl2):
            writer_fl3.writerow(row1 + row2)

merge_csv_files("Spirit_Quality.csv", "Spirit_QualityNew.csv", "Spirit_Quality_L


# most data lost
SQ_data_m = pd.read_csv(r"D:/Uni/5 Semester/Data Wrangling/In class assignmnet/S
SQ_new_data_m = pd.read_csv(r"D:/Uni/5 Semester/Data Wrangling/In class assignmn

def merge_csv_files(SQ_data_m, SQ_new_data_m, output_file_m):
    with open(SQ_data_m, "r") as fm1, open(SQ_new_data_m, "r") as fm2, open(outp
        reader_fm1 = csv.reader(fm1)
        reader_fm2 = csv.reader(fm2)
        writer_fm3 = csv.writer(fm3)

        for row1, row2 in zip(reader_fm1, reader_fm2):
            writer_fm3.writerow([row1[0] + row2[1]])

merge_csv_files("Spirit_Quality.csv", "Spirit_QualityNew.csv", "Spirit_Quality_M
```

14. (ii) Display the number of records in the resulting set of data in each of the above situations.

In [ ]:
```
# Number of records in least data lost
with open("Spirit_Quality_LeastDataLOss.csv", "r") as fl3:
    reader_fl3 = csv.reader(fl3)
    num_records_L = len(list(reader_fl3))

print("The no. of records in least data lost is --->", num_records_L)


# Number of records in most data lost
with open("Spirit_Quality_MostDataLoss.csv", "r") as fm3:
    reader_fm3 = csv.reader(fm3)
    num_records_M = len(list(reader_fm3))

print("The no. of records in most data lost is --->", num_records_L)
```

```
The no. of records in least data lost is ---> 5002
The no. of records in most data lost is ---> 5002
```

Question 02 - "Flights_data.csv" data set

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import csv

data_set = pd.read_csv(r"D:/Uni/5 Semester/Data Wrangling/In class assignmnet/Fl
print(data_set)
```

```
      Month  average number of passengers destination loss of luggage
0    1949-01                     123.99352         USA             yes
1    1949-02                     124.32410     Germany             NaN
2    1949-03                     124.72410       Japan             NaN
3    1949-04                     125.13451         USA             yes
4    1949-05                     125.72341       Japan             NaN
..       ...                           ...         ...             ...
139  1960-08                     479.01240     Germany              no
140  1960-09                     482.45100       Japan             NaN
141  1960-10                     485.15400         USA             NaN
142  1960-11                     488.67700       Dubai             NaN
143  1960-12                     491.68200   Australia              no

[144 rows x 4 columns]
```

1. How many observations are there which contain at least one missing value

In [ ]:
```
missing_nof_obs = sum(data_set.isnull().sum())
print("No of obeservations which have at leat one missing value --->", missing_r
```

```
No of obeservations which have at leat one missing value ---> 112
```

2. List the variables which contain missing values, with the corresponding number of missing values in each variable.

In [ ]:
```
missing_nof_obs = data_set.isnull().sum()
print("List of variables with missing values:\n", missing_nof_obs)
```

```
List of variables with missing values:
 Month                          0
average number of passengers      5
destination                    0
loss of luggage                107
dtype: int64
```

3. Calculate the percentage of missing values (out of rows) for each of the variables.

In [ ]:
```
percent_missing = missing_nof_obs / len(data_set)
print("Percentage of missing values --->", percent_missing)
```

```
Percentage of missing values ---> Month                          0.000000
average number of passengers    0.034722
destination                     0.000000
loss of luggage                 0.743056
dtype: float64
```

4. i) Suggest a remedy for the missing values in “loss of luggage” variable with sufficient reasoning.

    - The best remedy we can use is to impute the missing values with "no" because that is the most likely value for a passenger who had not loss his/ her luggage.

4. ii) Perform your suggestion and Save the data set as a csv file having the name as “Flights_data_new1.csv”

In [ ]:
```
# Replace missing values with "no"
data_set['loss of luggage'] = data_set['loss of luggage'].fillna("no")

# Convert "loss of luggage" column to a categorical variable
data_set['loss of luggage'] = data_set['loss of luggage'].astype('category')

print(data_set)

# Save as a csv file
data_set.to_csv("Flights_data_new1.csv", index=False)
```

```
      Month  average number of passengers destination loss of luggage
0   1949-01                     123.99352         USA             yes
1   1949-02                     124.32410     Germany              no
2   1949-03                     124.72410       Japan              no
3   1949-04                     125.13451         USA             yes
4   1949-05                     125.72341       Japan              no
..      ...                           ...         ...             ...
139 1960-08                     479.01240     Germany              no
140 1960-09                     482.45100       Japan              no
141 1960-10                     485.15400         USA              no
142 1960-11                     488.67700       Dubai              no
143 1960-12                     491.68200   Australia              no

[144 rows x 4 columns]
```

5. i) What would you suggest as a remedy for missing values in “average number of passengers” variable with reasons.

    - The best remedy for the missing values in "average number of passengers" is to impute missing values with median since it is less sensitive to outliers than mean.

5. ii) Perform your suggestion (Hint: You can use any mathematical calculation). Describe the steps used.

A. Find the median value of non-missing values
B. Replace the missing values with the median value

In [ ]:
```
data_set['average number of passengers'] = data_set['average number of passenger

print(data_set)
```

```
      Month  average number of passengers destination loss of luggage
0   1949-01                     123.99352         USA             yes
1   1949-02                     124.32410     Germany              no
2   1949-03                     124.72410       Japan              no
3   1949-04                     125.13451         USA             yes
4   1949-05                     125.72341       Japan              no
..      ...                           ...         ...             ...
139 1960-08                     479.01240     Germany              no
140 1960-09                     482.45100       Japan              no
141 1960-10                     485.15400         USA              no
142 1960-11                     488.67700       Dubai              no
143 1960-12                     491.68200   Australia              no

[144 rows x 4 columns]
```

5. iii) Save the data set with your performance as a csv file having the name as "Flights_data_new2.csv"

In [ ]:
```
data_set.to_csv('Flights_data_new2.csv', index=False)
```

5. iv) Do you think dropping records containing missing values is reasonable in this data? Give reasons.

- No, because the data set is very small and dropping records will reduce the amount of data availabl for the analysis.
- At the same time the missing values are also relatively few and they are not concentrated in any special variable so no need to dropp the records.
- So, it is better to impute values than dopping the records with missing values.