wine-eda

August 22, 2025

EDA - Exploratory Data Analysis in Python

Exploratory Data Analysis (EDA) is a important step in data analysis which focuses on understanding patterns, trends and relationships through statistical tools and visualizations

[1]:

```
#Step 1:Importing Required Libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings as wr
wr.filterwarnings('ignore')
```

[3]:

```
import os

print(os.listdir("C:/Users/Tanya Raj/OneDrive/Desktop"))
```

```
['02_numpy(linear algebra).ipynb', '1.xlsx', '2(wine).xlsx', 'ADHAAR CARD.pdf',
'aditya', 'Arduino IDE.lnk', 'Canva.lnk', 'desktop.ini', 'FITA', 'major_project
doc', 'myself info', 'python basics', 'python function-2']
```

[4]:

```
pip install openpyxl
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: openpyxl in c:\users\tanya
raj\appdata\roaming\python\python313\site-packages (3.1.5)
Requirement already satisfied: et-xmlfile in c:\users\tanya
raj\appdata\roaming\python\python313\site-packages (from openpyxl) (2.0.0)
Note: you may need to restart the kernel to use updated packages.


[notice] A new release of pip is available: 25.1.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[5]:

```
import pandas as pd

dataframe = pd.read_excel(r"C:\Users\Tanya Raj\OneDrive\Desktop\2(wine).xlsx")
print(dataframe.head())
```

```
   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  Id
0      9.4        5   0
1      9.8        5   1
2      9.8        5   2
3      9.8        6   3
4      9.4        5   4
```

[7]:

```
#step 3:Analyzing the data

#1.df.shape():used to understand the number of rows and columns in the dataset.

dataframe.shape
```

[7]:

```
(1143, 13)
```

[8]:

```
dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1143 entries, 0 to 1142
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1143 non-null   float64
 1   volatile acidity      1143 non-null   float64
 2   citric acid           1143 non-null   float64
 3   residual sugar        1143 non-null   float64
 4   chlorides             1143 non-null   float64
 5   free sulfur dioxide   1143 non-null   float64
 6   total sulfur dioxide  1143 non-null   float64
 7   density               1143 non-null   float64
 8   pH                    1143 non-null   float64
 9   sulphates             1143 non-null   float64
 10  alcohol               1143 non-null   float64
 11  quality               1143 non-null   int64
 12  Id                    1143 non-null   int64
dtypes: float64(11), int64(2)
memory usage: 116.2 KB
```

[9]:
```
dataframe.describe()
```

[9]:
```
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    1143.000000       1143.000000  1143.000000     1143.000000
mean        8.311111          0.531339     0.268364        2.532152
std         1.747595          0.179633     0.196686        1.355917
min         4.600000          0.120000     0.000000        0.900000
25%         7.100000          0.392500     0.090000        1.900000
50%         7.900000          0.520000     0.250000        2.200000
75%         9.100000          0.640000     0.420000        2.600000
max        15.900000          1.580000     1.000000       15.500000

         chlorides  free sulfur dioxide  total sulfur dioxide      density  \
count  1143.000000          1143.000000           1143.000000  1143.000000
mean      0.086933            15.615486             45.914698     0.996730
std       0.047267            10.250486             32.782130     0.001925
min       0.012000             1.000000              6.000000     0.990070
25%       0.070000             7.000000             21.000000     0.995570
50%       0.079000            13.000000             37.000000     0.996680
75%       0.090000            21.000000             61.000000     0.997845
max       0.611000            68.000000            289.000000     1.003690

                pH    sulphates      alcohol      quality           Id
count  1143.000000  1143.000000  1143.000000  1143.000000  1143.000000
mean      3.311015     0.657708    10.442111     5.657043   804.969379
std       0.156664     0.170399     1.082196     0.805824   463.997116
min       2.740000     0.330000     8.400000     3.000000     0.000000
25%       3.205000     0.550000     9.500000     5.000000   411.000000
50%       3.310000     0.620000    10.200000     6.000000   794.000000
75%       3.400000     0.730000    11.100000     6.000000  1209.500000
max       4.010000     2.000000    14.900000     8.000000  1597.000000
```

[10]:
```
dataframe.columns.tolist()
```

[10]:
```
['fixed acidity',
 'volatile acidity',
 'citric acid',
 'residual sugar',
 'chlorides',
 'free sulfur dioxide',
 'total sulfur dioxide',
 'density',
 'pH',
 'sulphates',
 'alcohol',
 'quality',
 'Id']
```

[11]:

```
#Checking Missing Values

dataframe.isnull().sum()
```

[11]:

```
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
Id                      0
dtype: int64
```

[12]:

```
#Step 5:Checking for the duplicate values

dataframe.nunique()
```

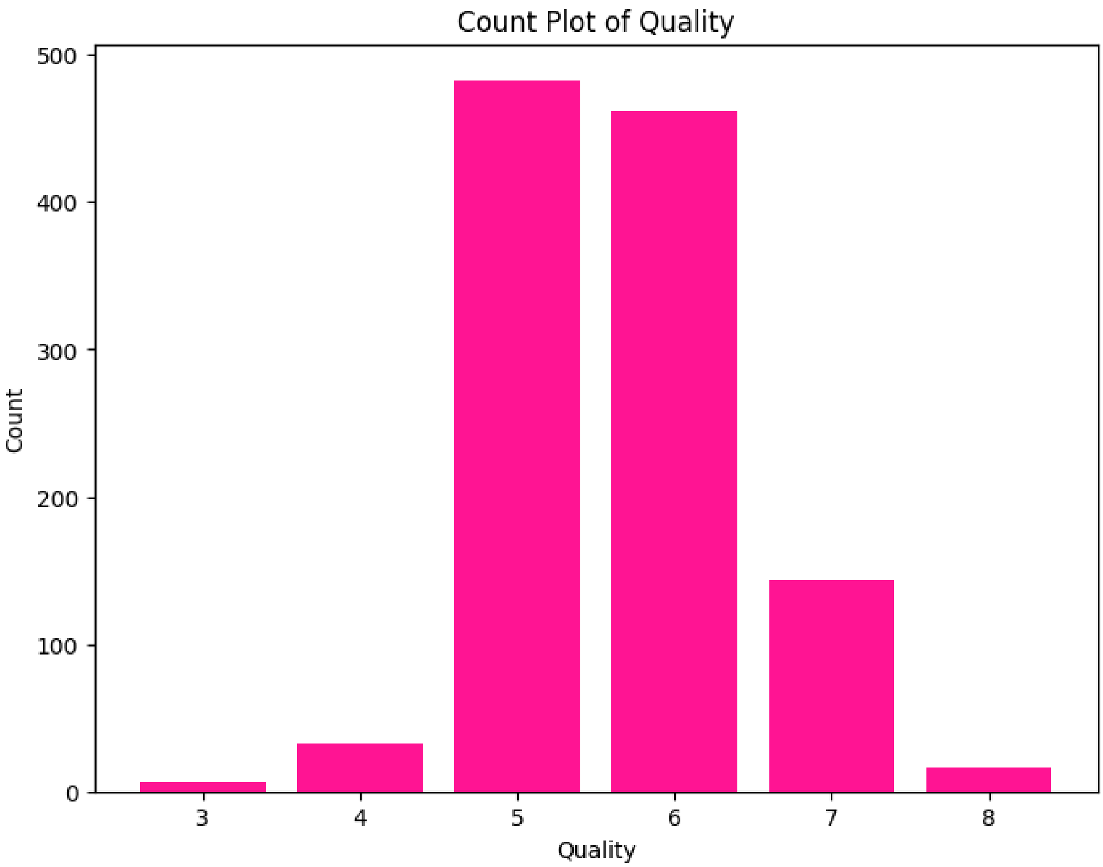
[12]:

```
fixed acidity             91
volatile acidity         135
citric acid               77
residual sugar            80
chlorides                131
free sulfur dioxide       53
total sulfur dioxide     138
density                  388
pH                        87
sulphates                 89
alcohol                   61
quality                    6
Id                      1143
dtype: int64
```

[14]:

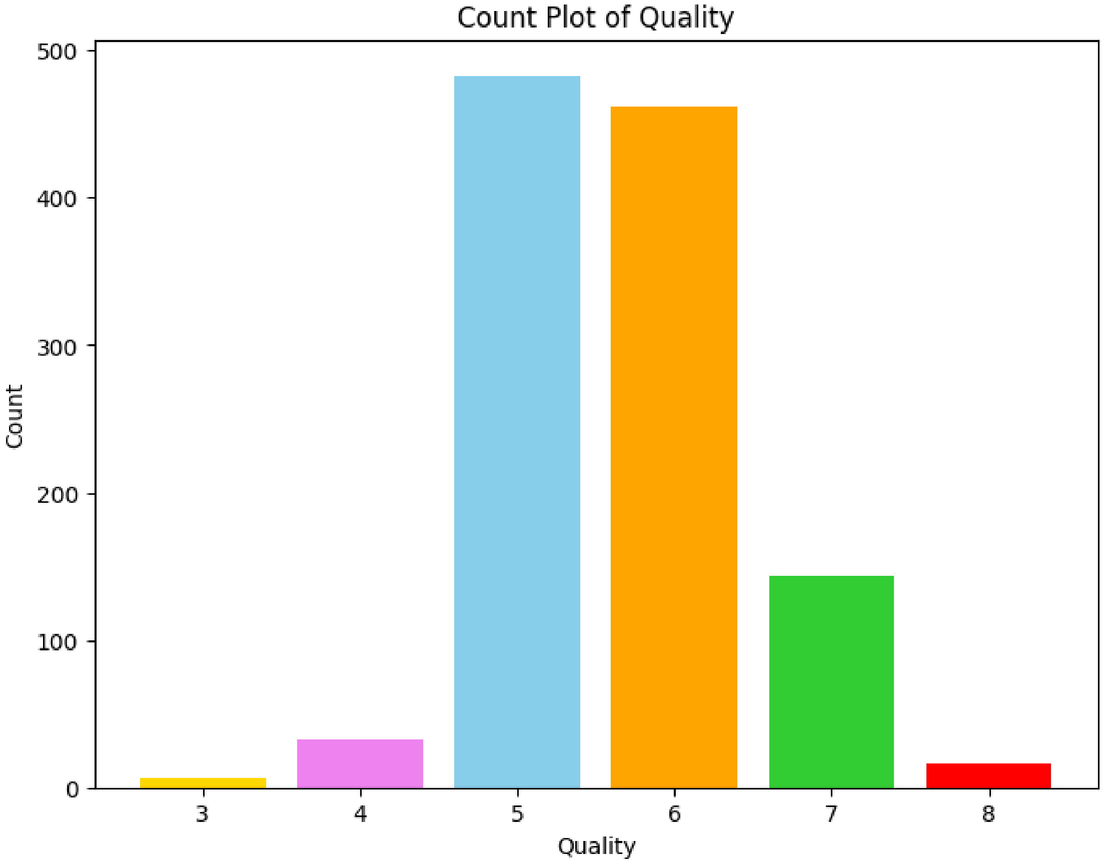
```
#step 6:Univariate Analysis

quality_counts = dataframe['quality'].value_counts()
```

```
plt.figure(figsize=(8, 6))
plt.bar(quality_counts.index, quality_counts, color='deeppink')
plt.title('Count Plot of Quality')
plt.xlabel('Quality')
plt.ylabel('Count')
plt.show()
```



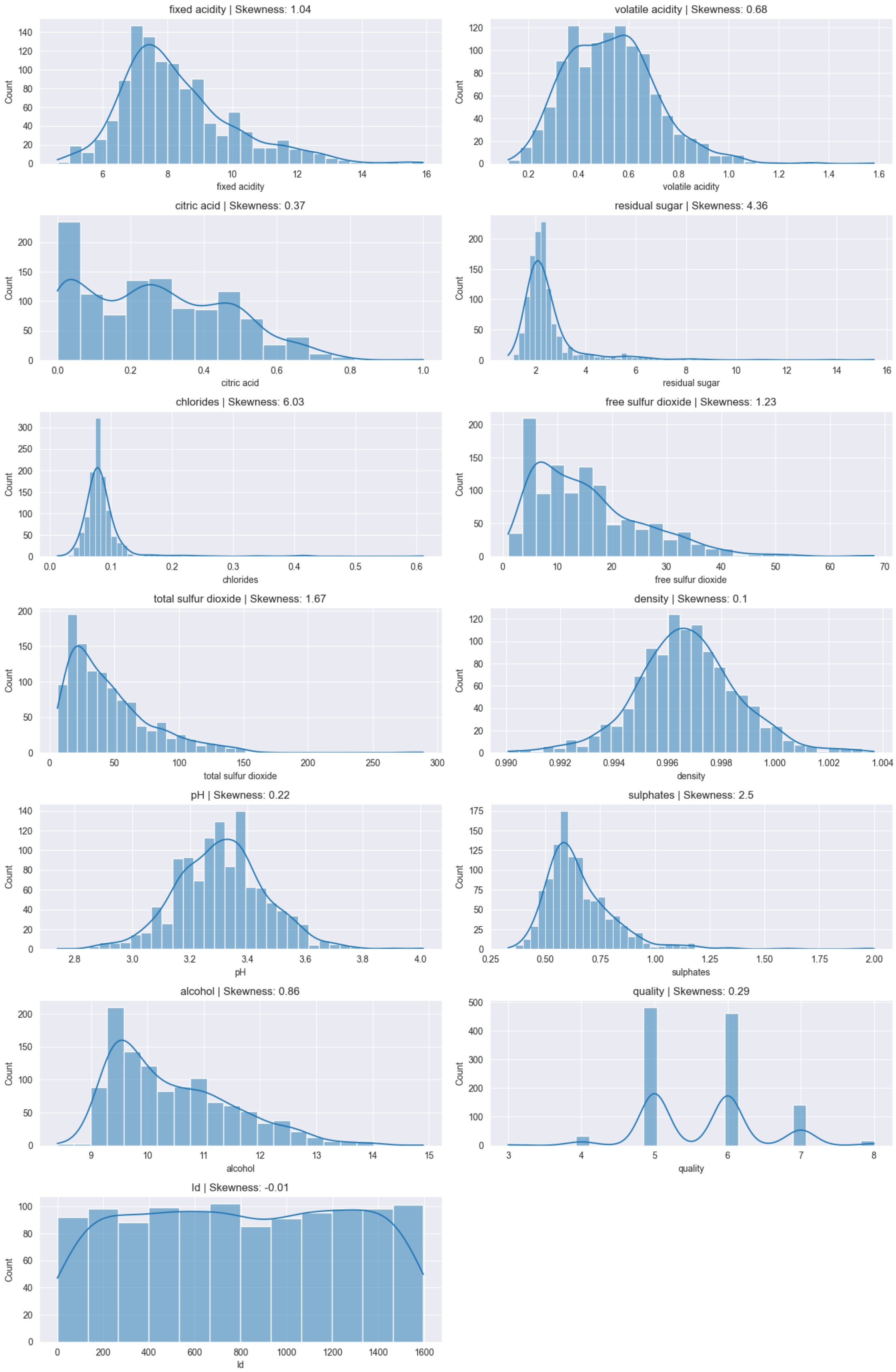
[15]:

```
# step 6: Univariate Analysis
quality_counts = dataframe['quality'].value_counts()

plt.figure(figsize=(8, 6))

# give a list of colors (length must match number of bars)
colors = ['skyblue', 'orange', 'limegreen', 'violet', 'red', 'gold']

plt.bar(quality_counts.index, quality_counts, color=colors[:
 ↪len(quality_counts)])
plt.title('Count Plot of Quality')
plt.xlabel('Quality')
plt.ylabel('Count')
plt.show()
```



KERNEL DENSITY PLOT : for understanding variance in the dataset.

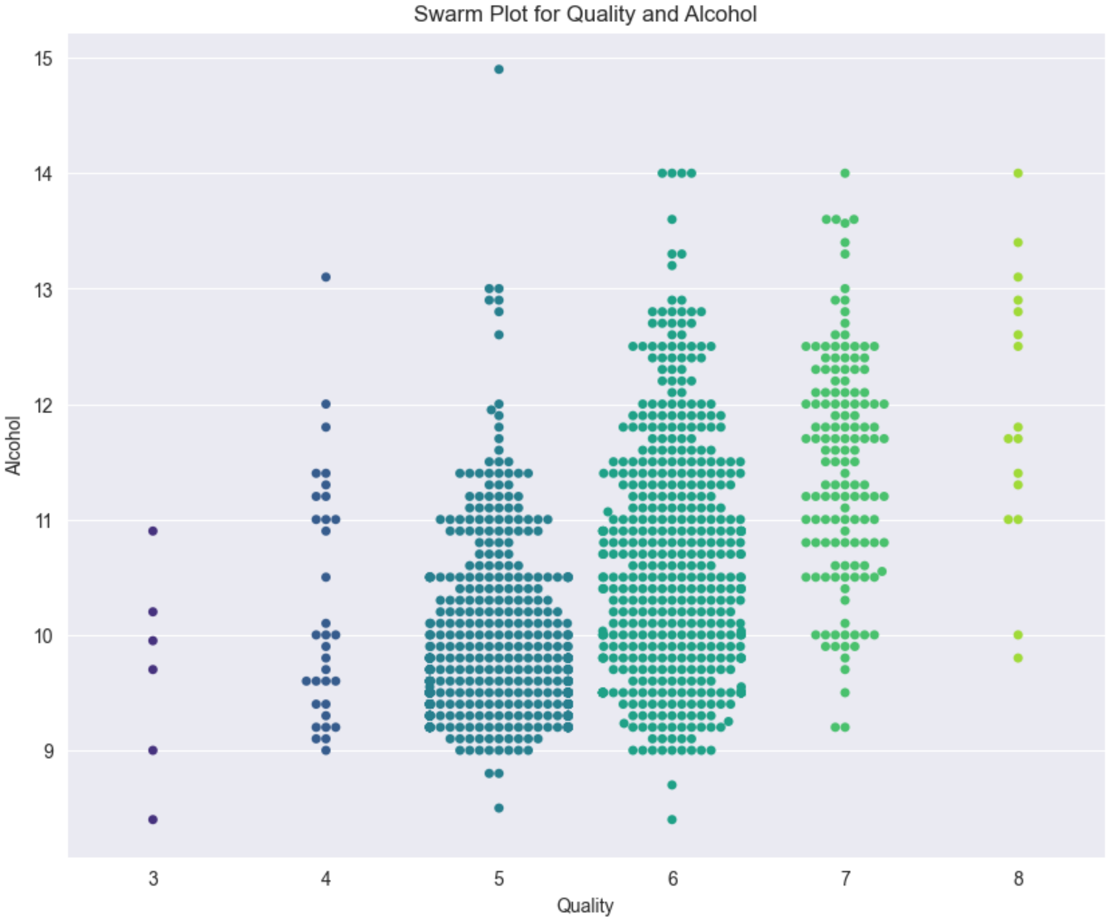
[19]:
```
sns.set_style("darkgrid")

numerical_columns = dataframe.select_dtypes(include=["int64", "float64"]).
 ↪columns

plt.figure(figsize=(14, len(numerical_columns) * 3))
for idx, feature in enumerate(numerical_columns, 1):
    plt.subplot(len(numerical_columns), 2, idx)
    sns.histplot(dataframe[feature], kde=True)
    plt.title(f"{feature} | Skewness: {round(dataframe[feature].skew(), 2)}")

plt.tight_layout()
plt.show()
```


fixed acidity | Skewness: 1.04
Count
0
20
40
60
80
100
120
140
6
8
10
12
14
16
fixed acidity
volatile acidity | Skewness: 0.68
Count
0
20
40
60
80
100
120
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
volatile acidity
citric acid | Skewness: 0.37
Count
0
50
100
150
200
0.0
0.2
0.4
0.6
0.8
1.0
citric acid
residual sugar | Skewness: 4.36
Count
0
50
100
150
200
2
4
6
8
10
12
14
16
residual sugar
chlorides | Skewness: 6.03
Count
0
50
100
150
200
250
300
0.0
0.1
0.2
0.3
0.4
0.5
0.6
chlorides
free sulfur dioxide | Skewness: 1.23
Count
0
50
100
150
200
0
10
20
30
40
50
60
70
free sulfur dioxide
total sulfur dioxide | Skewness: 1.67
Count
0
50
100
150
200
0
50
100
150
200
250
300
total sulfur dioxide
density | Skewness: 0.1
Count
0
20
40
60
80
100
120
0.990
0.992
0.994
0.996
0.998
1.000
1.002
1.004
density
pH | Skewness: 0.22
Count
0
20
40
60
80
100
120
140
2.8
3.0
3.2
3.4
3.6
3.8
4.0
pH
sulphates | Skewness: 2.5
Count
0
25
50
75
100
125
150
175
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
sulphates
alcohol | Skewness: 0.86
Count
0
50
100
150
200
9
10
11
12
13
14
15
alcohol
quality | Skewness: 0.29
Count
0
100
200
300
400
500
3
4
5
6
7
8
quality
Id | Skewness: -0.01
Count
0
20
40
60
80
100
0
200
400
600
800
1000
1200
1400
1600
Id

SWARM PLOT: for showing the outlier in the data

[20]:

```
plt.figure(figsize=(10, 8))

sns.swarmplot(x="quality", y="alcohol", data=dataframe, palette='viridis')

plt.title('Swarm Plot for Quality and Alcohol')
plt.xlabel('Quality')
plt.ylabel('Alcohol')
plt.show()
```

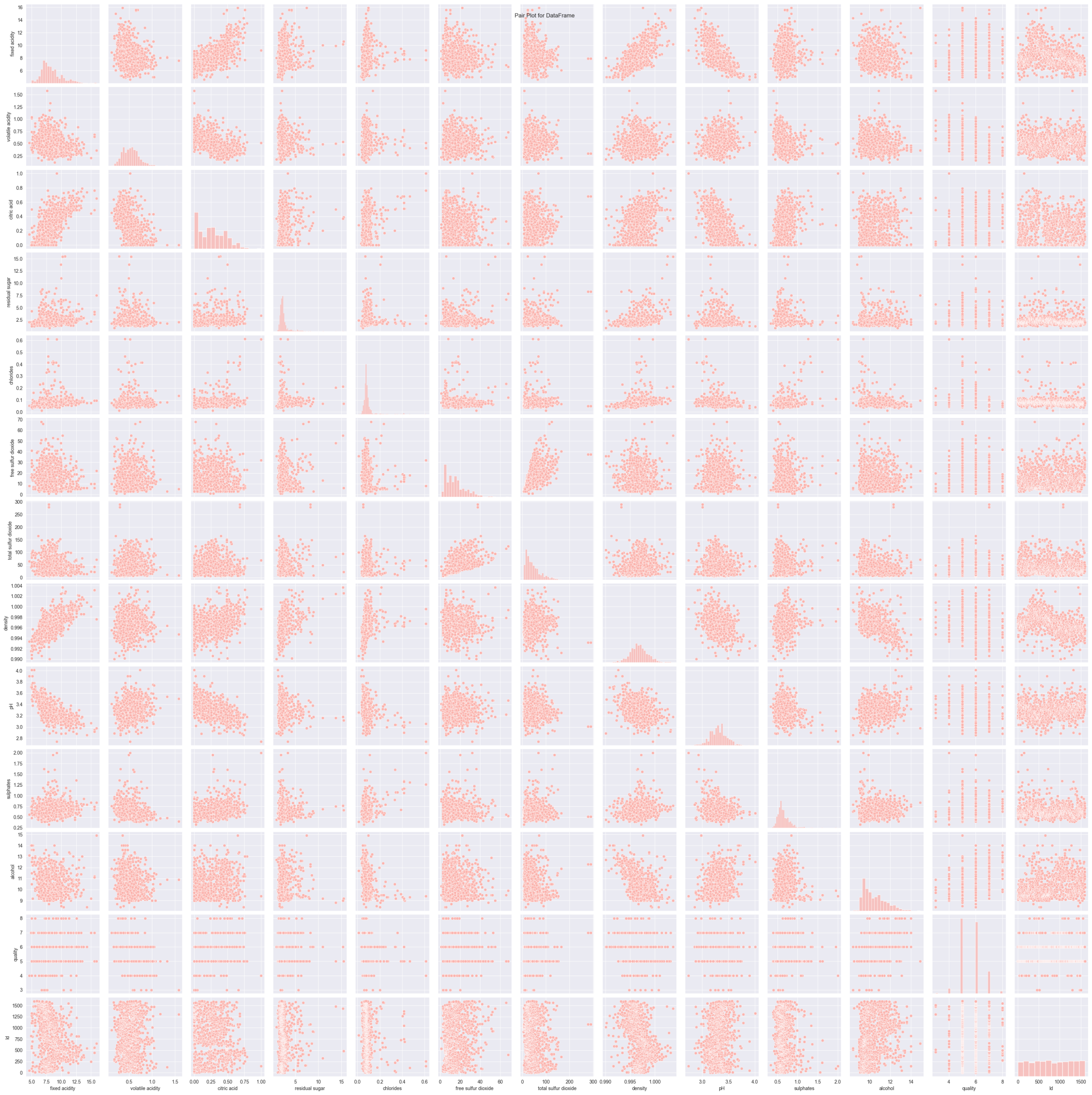


[22]:

```
#step 7: Bivariate Analysis

#1.Pair Plot for showing the distribution of the individual variables.

sns.set_palette("Pastel1")
```

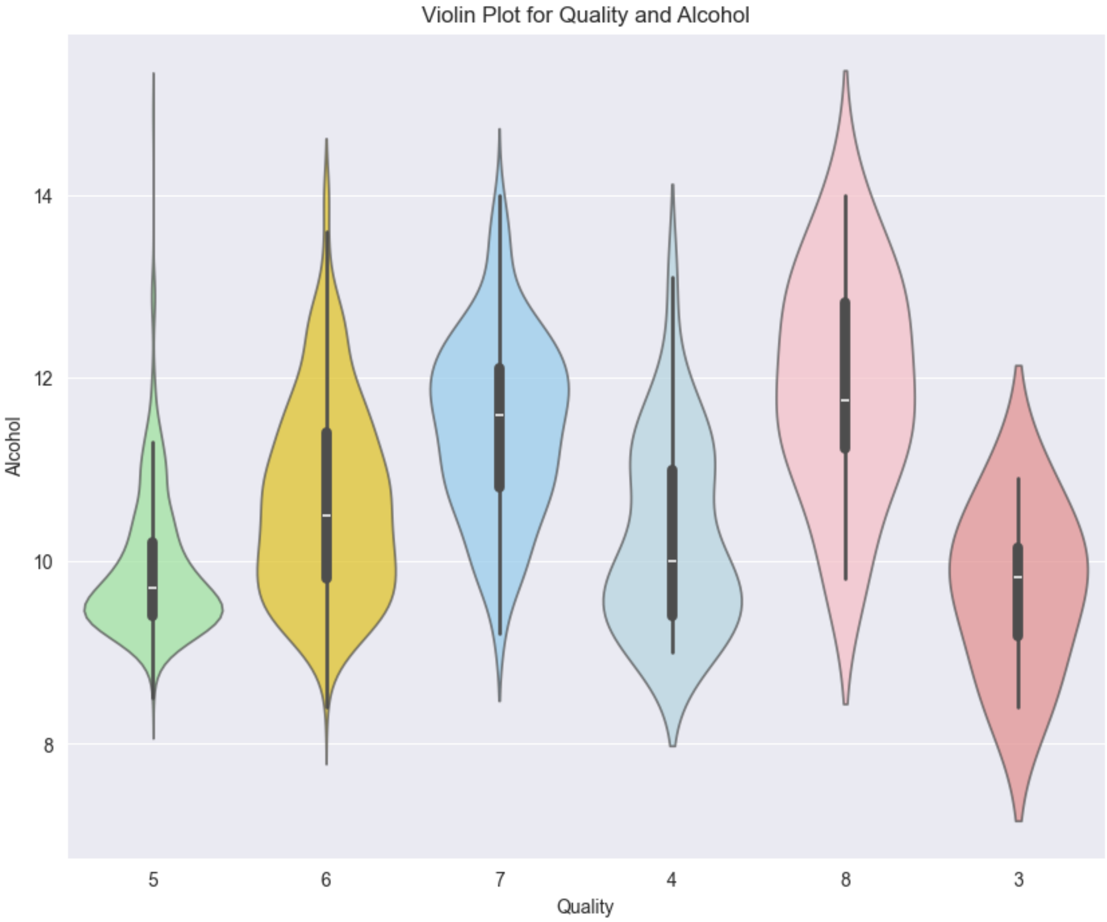
```
plt.figure(figsize=(10, 6))

sns.pairplot(dataframe)

plt.suptitle('Pair Plot for DataFrame')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



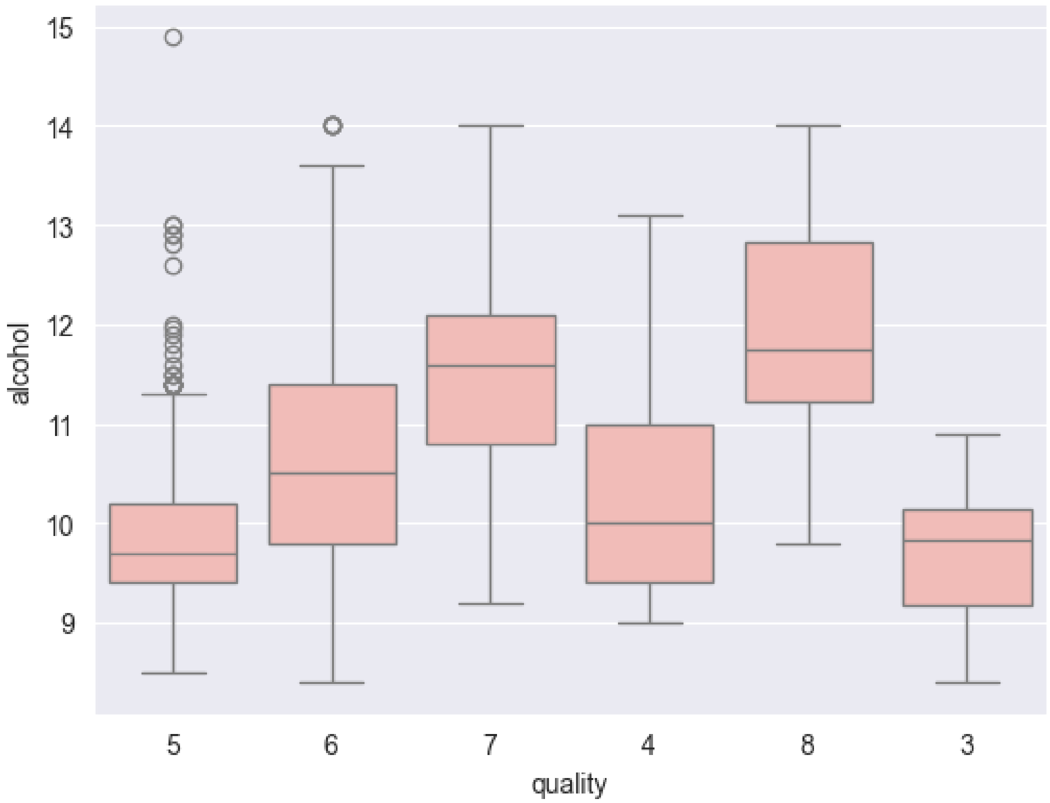
[23]:
```
#2.violine Plot : for examining the relationship between alcohol and quality.

dataframe['quality'] = dataframe['quality'].astype(str)
```

```
plt.figure(figsize=(10, 8))

sns.violinplot(x="quality", y="alcohol", data=dataframe, palette={
               '3': 'lightcoral', '4': 'lightblue', '5': 'lightgreen', '6':␣
 ↪'gold', '7': 'lightskyblue', '8': 'lightpink'}, alpha=0.7)

plt.title('Violin Plot for Quality and Alcohol')
plt.xlabel('Quality')
plt.ylabel('Alcohol')
plt.show()
```

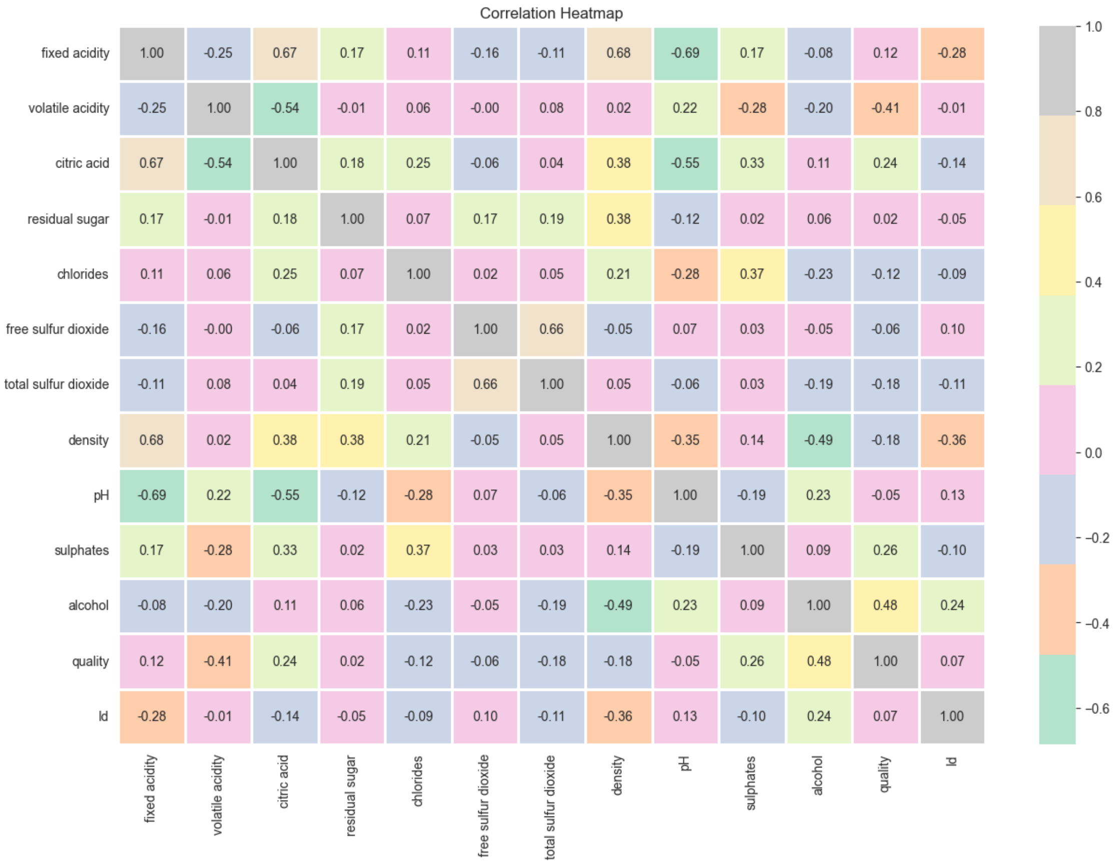


[25]:
```
3. #Box Plot for examining the relationship between alcohol and Quality

sns.boxplot(x='quality', y='alcohol', data=dataframe)
```

[25]:
```
<Axes: xlabel='quality', ylabel='alcohol'>
```



```
[26]: #step 8:Multivariate Analysis

      plt.figure(figsize=(15, 10))

      sns.heatmap(dataframe.corr(), annot=True, fmt='.2f', cmap='Pastel2',␣
       ↪linewidths=2)

      plt.title('Correlation Heatmap')
      plt.show()
```


Correlation Heatmap
fixed acidity 1.00 -0.25 0.67 0.17 0.11 -0.16 -0.11 0.68 -0.69 0.17 -0.08 0.12 -0.28
volatile acidity -0.25 1.00 -0.54 -0.01 0.06 -0.00 0.08 0.02 0.22 -0.28 -0.20 -0.41 -0.01
citric acid 0.67 -0.54 1.00 0.18 0.25 -0.06 0.04 0.38 -0.55 0.33 0.11 0.24 -0.14
residual sugar 0.17 -0.01 0.18 1.00 0.07 0.17 0.19 0.38 -0.12 0.02 0.06 0.02 -0.05
chlorides 0.11 0.06 0.25 0.07 1.00 0.02 0.05 0.21 -0.28 0.37 -0.23 -0.12 -0.09
free sulfur dioxide -0.16 -0.00 -0.06 0.17 0.02 1.00 0.66 -0.05 0.07 0.03 -0.05 -0.06 0.10
total sulfur dioxide -0.11 0.08 0.04 0.19 0.05 0.66 1.00 0.05 -0.06 0.03 -0.19 -0.18 -0.11
density 0.68 0.02 0.38 0.38 0.21 -0.05 0.05 1.00 -0.35 0.14 -0.49 -0.18 -0.36
pH -0.69 0.22 -0.55 -0.12 -0.28 0.07 -0.06 -0.35 1.00 -0.19 0.23 -0.05 0.13
sulphates 0.17 -0.28 0.33 0.02 0.37 0.03 0.03 0.14 -0.19 1.00 0.09 0.26 -0.10
alcohol -0.08 -0.20 0.11 0.06 -0.23 -0.05 -0.19 -0.49 0.23 0.09 1.00 0.48 0.24
quality 0.12 -0.41 0.24 0.02 -0.12 -0.06 -0.18 -0.18 -0.05 0.26 0.48 1.00 0.07
Id -0.28 -0.01 -0.14 -0.05 -0.09 0.10 -0.11 -0.36 0.13 -0.10 0.24 0.07 1.00
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality
Id
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6