Cognorise Infotech Internship

Name: Anuttama Mondal

Task 3: Red Wine Quality

Problem Statement:

The two datasets are related to red and white variants of the Portuguese "VinhoVerde" wine. For more details, consult the reference [Cortez et al., 2009]. Due toprivacy and logistic issues, only physicochemical (inputs) and sensory (the output)variables are available (e.g. there is no data about grape types, wine brand, wineselling price, etc.).These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones)

Import Libraries:

In [1]:
```
import pandas as pd         #For data manipulation and analysis
import numpy as mp          # For numerical computation and handling arrays
import matplotlib.pyplot as plt #For data visualization
import seaborn as sns           # For enhanced data visualization
from sklearn.model_selection import train_test_split      #For Data splitting
from datetime import datetime               #For Working with dates and times
import warnings
warnings.filterwarnings("ignore")           #disable warning
%matplotlib inline
```

In [3]:
```
df= pd.read_csv("winequality-red.csv")
```

In [4]:
```
df
```

Out[4]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1599 rows × 12 columns

Checking with the Rows for storing the Length:

In [5]:
```
df_len=len(df)
df_len
```

Out[5]:
```
1599
```

Displaying first and last rows and columns of the dataset:

In [6]:
```
df.head()
```

Out[6]:

	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

In [7]:
```
df.tail()
```

Out[7]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

View the information:

In [8]:
```
df.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
```

In [9]:
```
df.shape
```

Out[9]:
```
(1599, 12)
```

checking for any null or missing values:

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
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 [11]:
```
df.describe()
```

Out[11]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoho
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.00000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.42298
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.06566
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.40000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.50000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.20000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.10000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.90000

In [16]:
```
duplicates = df[df.duplicated()] #Checking for duplicates
dff=df.drop_duplicates() #Dropping of duplicates
```

In [17]:
```
dff
```

Out[17]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
5	7.4	0.660	0.00	1.8	0.075	13.0	40.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1593	6.8	0.620	0.08	1.9	0.068	28.0	38.0	0.99651	3.42	0.82	9.5	6
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

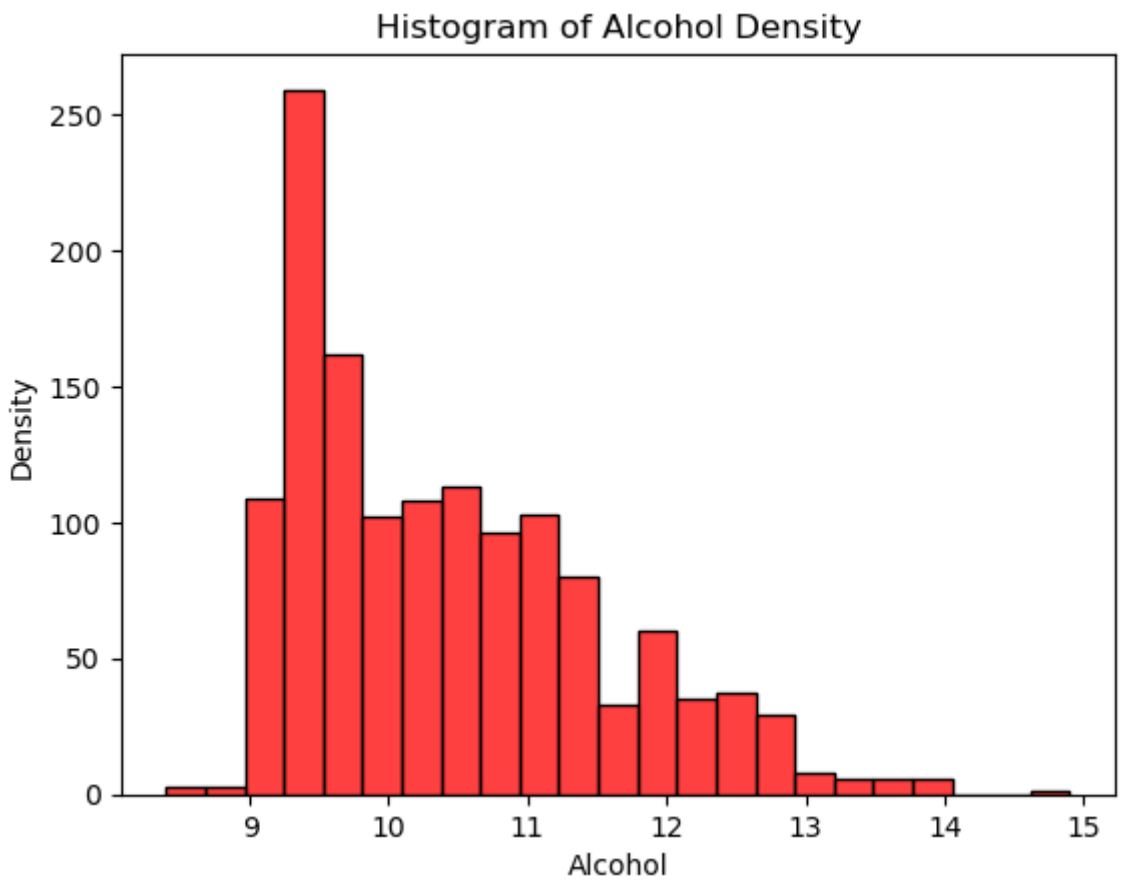
1359 rows × 12 columns

In [18]:
```
dff.nunique()
```

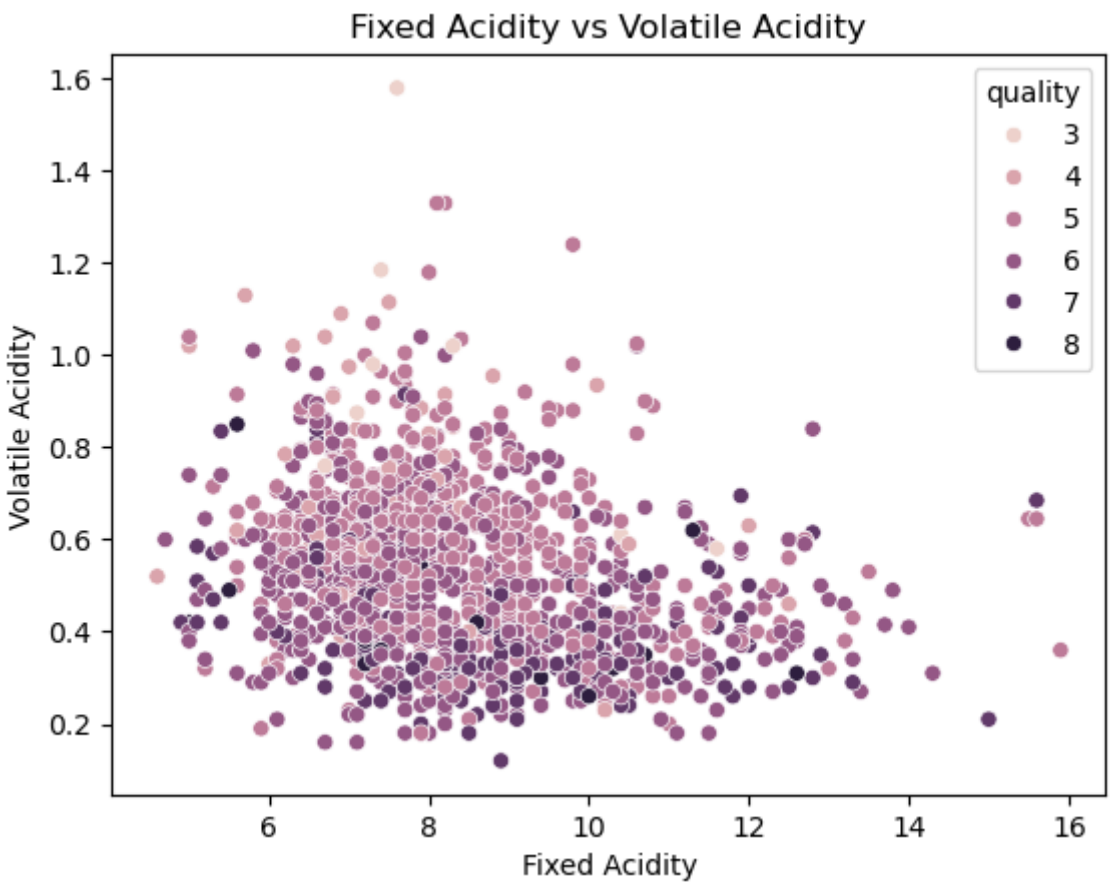
Out[18]:
```
fixed acidity           96
volatile acidity       143
citric acid             80
residual sugar          91
chlorides              153
free sulfur dioxide     60
total sulfur dioxide   144
density                436
pH                      89
sulphates               96
alcohol                 65
quality                  6
dtype: int64
```

Using EDA(Exploratory Data Analysis):

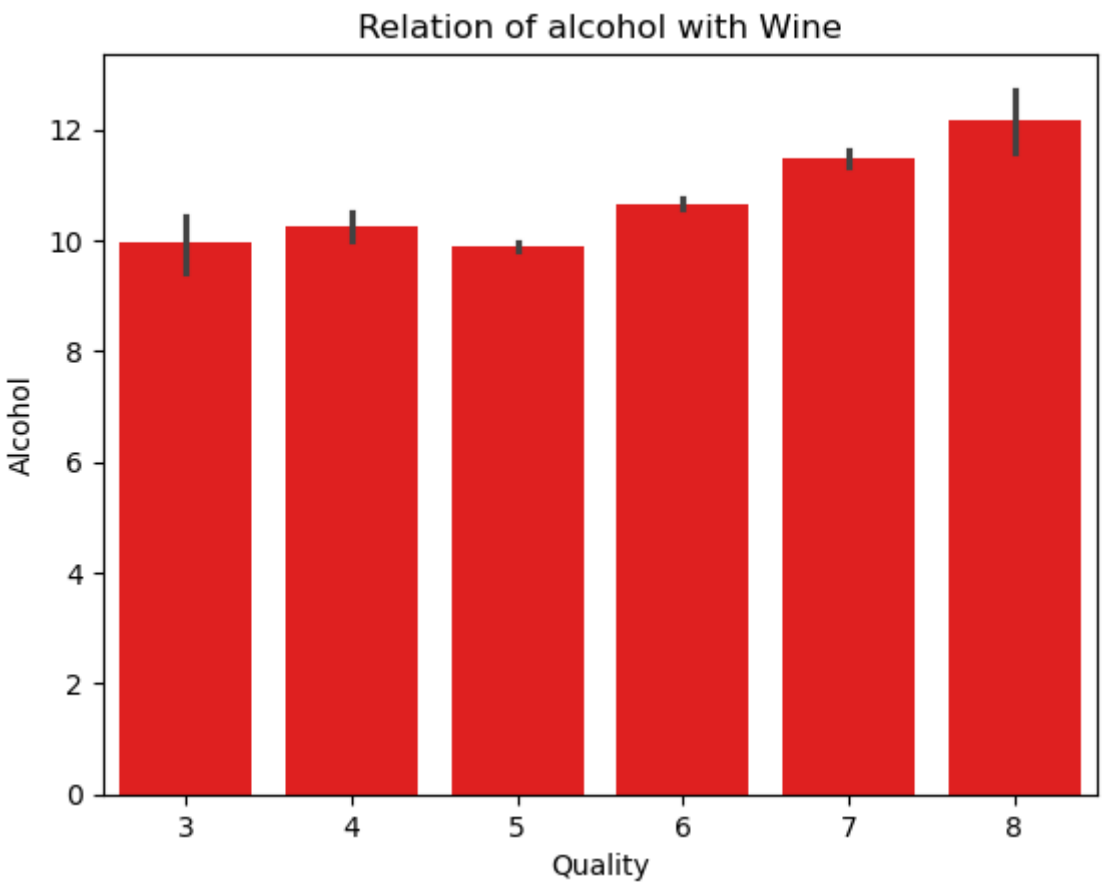
In [19]:
```
#Histogram of Alcohol Density
sns.histplot(dff['alcohol'],color='red');
plt.xlabel("Alcohol")
plt.ylabel("Density")
plt.title("Histogram of Alcohol Density");
```



In [25]:
```
#Fixed Acidity Vs Volatile Acidity
sns.scatterplot(x='fixed acidity', y='volatile acidity', hue='quality',data=dff);
plt.xlabel("Fixed Acidity")
plt.ylabel("Volatile Acidity")
plt.title("Fixed Acidity vs Volatile Acidity");
```

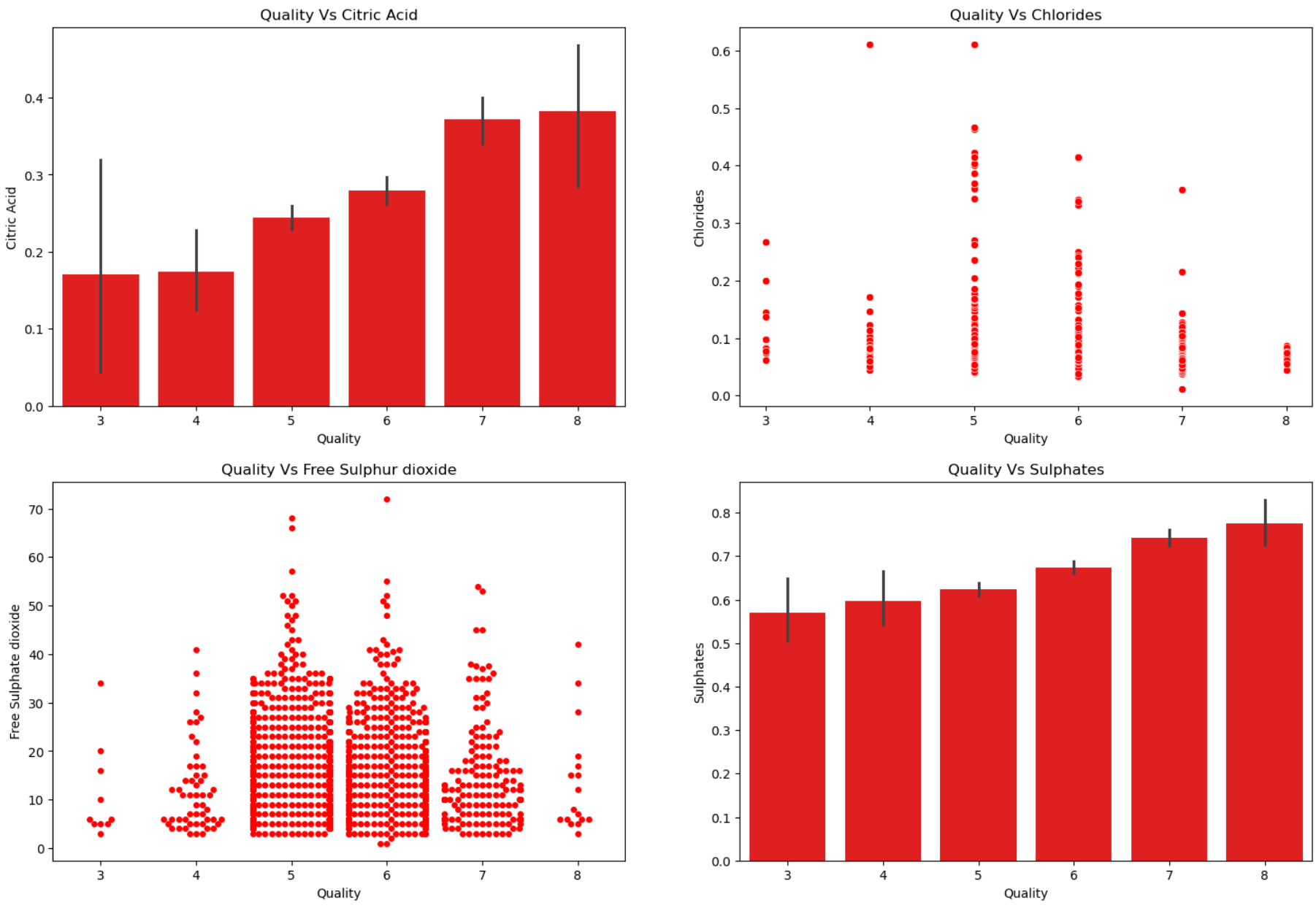


In [24]:
```
#Relation of Alcohol with Quality
sns.barplot(x='quality', y='alcohol', color = 'red',data=dff)
plt.title('Relation of alcohol with Wine')
plt.xlabel('Quality')
plt.ylabel('Alcohol')
plt.show()
```

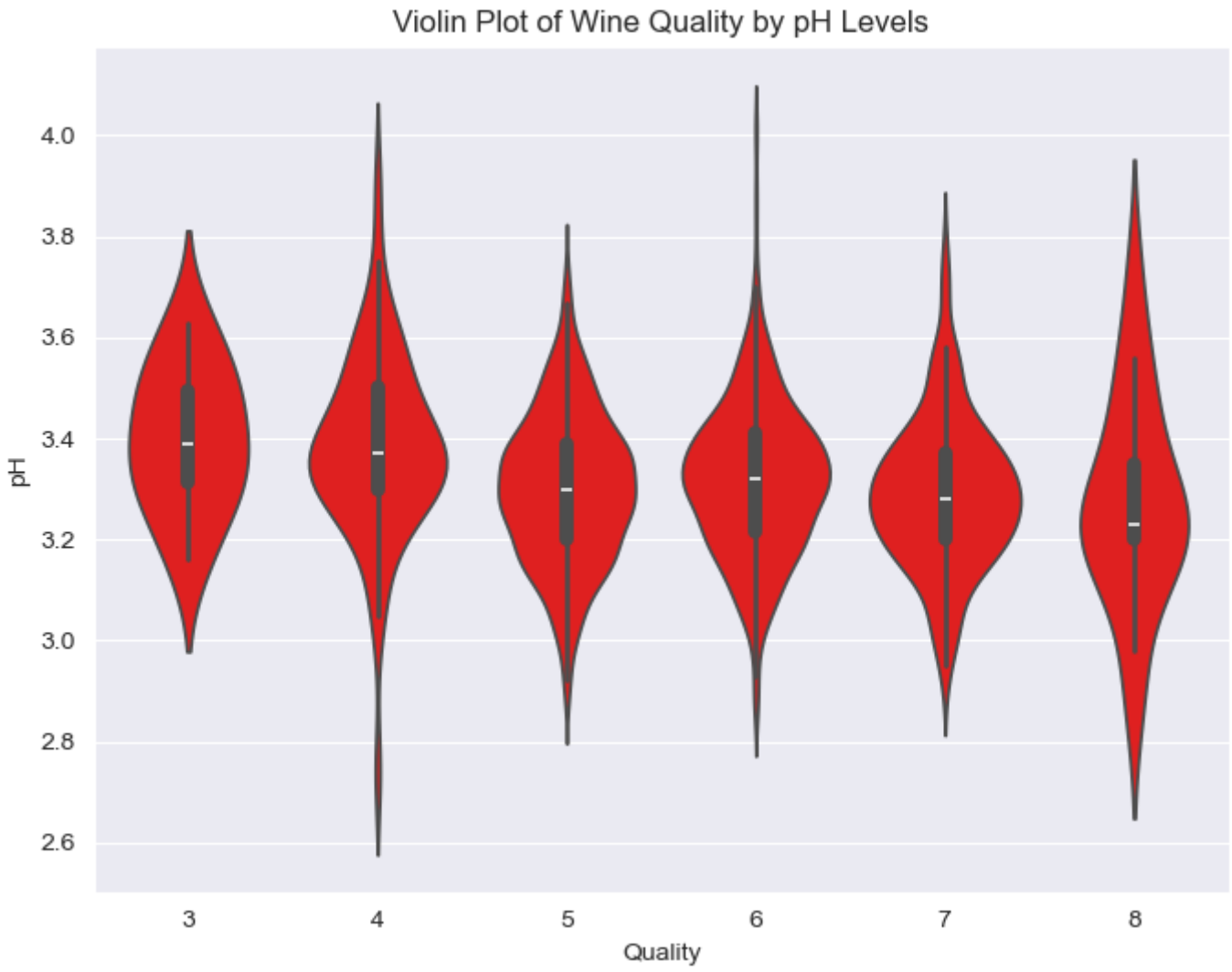


In [28]:
```
plt.figure(figsize=(18,12))
#Quality Vs Citric Acid"
plt.subplot(2,2,1) #2-Rows, 2-column, 1-first
sns.barplot(x = 'quality', y = 'citric acid',color='red', data = dff)
plt.title("Quality Vs Citric Acid")
plt.xlabel("Quality")
plt.ylabel("Citric Acid")

#Quality Vs Chlorides
plt.subplot(2, 2, 2)
sns.scatterplot(x = 'quality', y = 'chlorides',color='red',data = dff)
plt.title("Quality Vs Chlorides")
plt.xlabel("Quality")
plt.ylabel("Chlorides")

#Quality Vs Free Sulphurdioxide
plt.subplot(2,2,3)
sns.swarmplot(x = 'quality', y = 'free sulfur dioxide',color='red', data = dff)
plt.title("Quality Vs Free Sulphur dioxide")
plt.xlabel("Quality")
plt.ylabel("Free Sulphate dioxide")

#Quality Vs Sulphates
plt.subplot(2,2,4)
sns.barplot(x = 'quality', y = 'sulphates',color='red', data = dff)
plt.title("Quality Vs Sulphates")
plt.xlabel("Quality")
plt.ylabel("Sulphates")
```

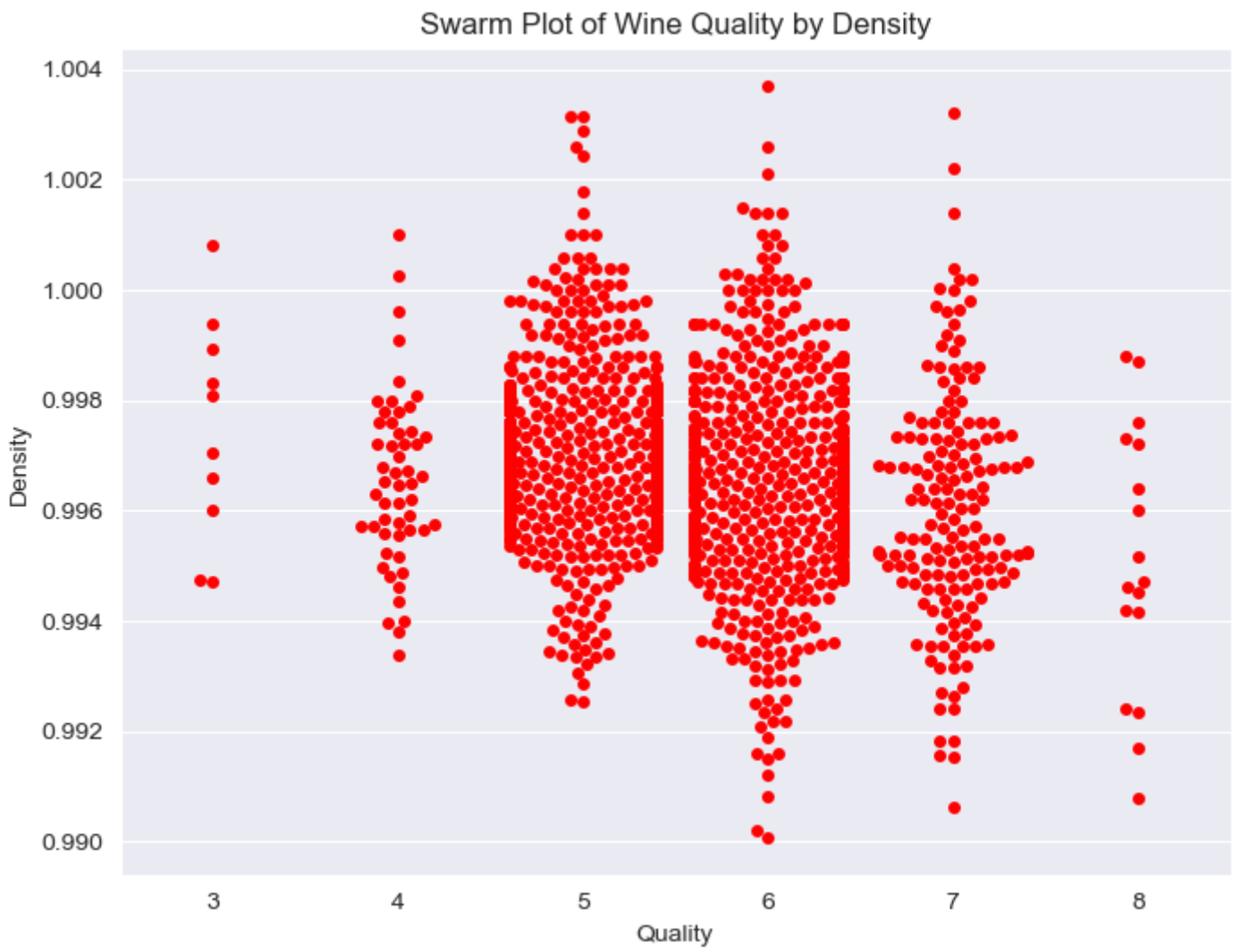
Out[28]: Text(0, 0.5, 'Sulphates')



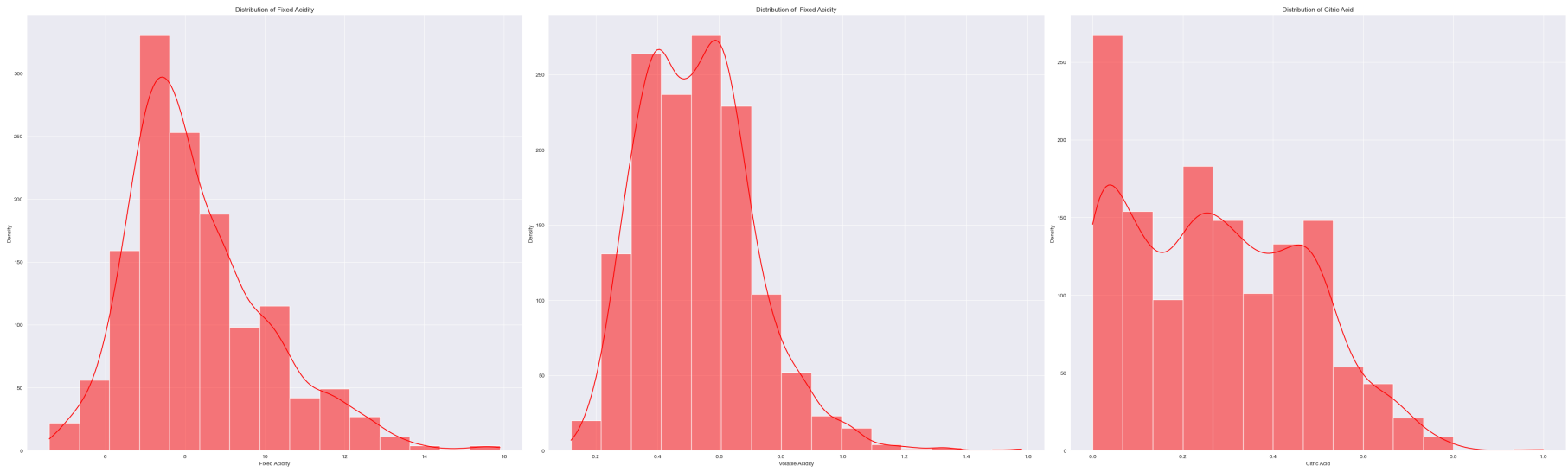
In [32]:
```
#Violin Plot of Wine Quality by pH Levels
sns.set_style('darkgrid')
plt.figure(figsize=(8, 6))
sns.violinplot(x='quality', y='pH', data=dff, color='red')
plt.title('Violin Plot of Wine Quality by pH Levels')
plt.xlabel('Quality')
plt.ylabel('pH')
plt.show()
```



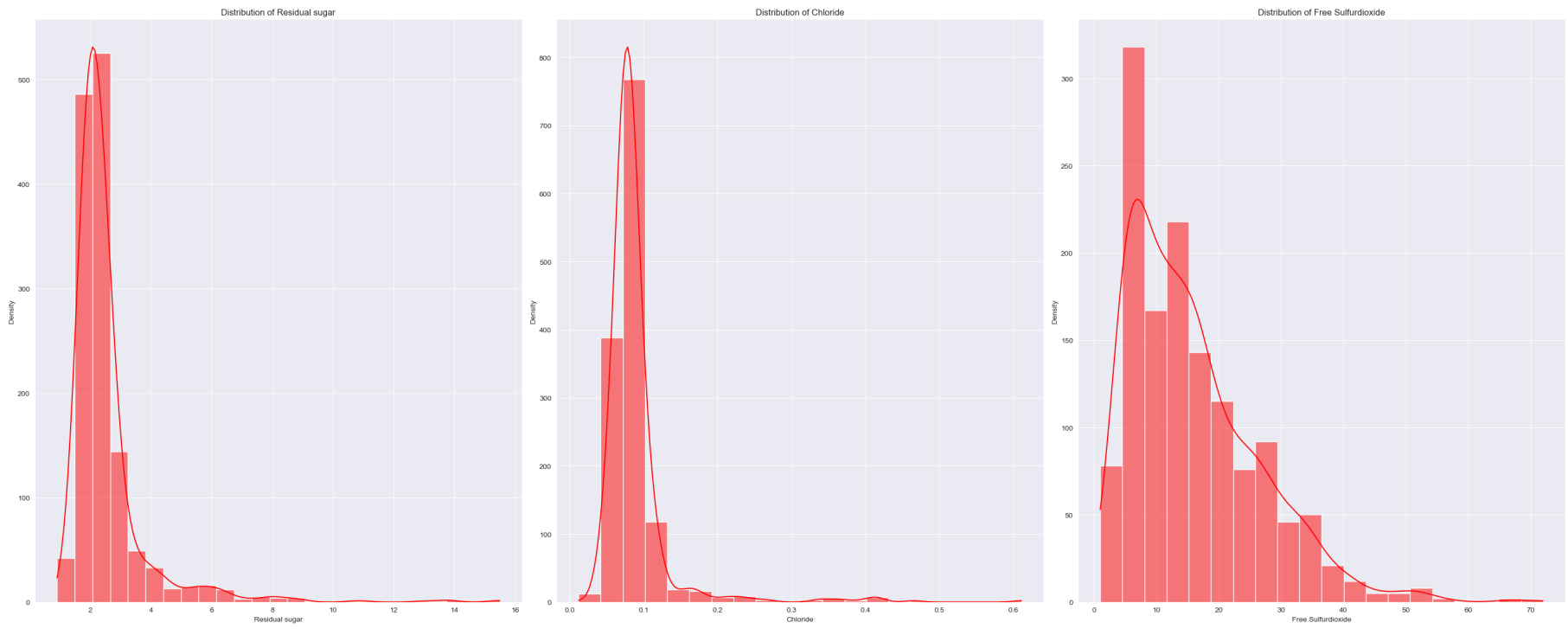
In [33]:
```
#Swarm Plot of Wine Quality by Density
sns.set_style('darkgrid')
plt.figure(figsize=(8, 6))
sns.swarmplot(x='quality', y='density', data=dff, color='red')
plt.title('Swarm Plot of Wine Quality by Density')
plt.xlabel('Quality')
plt.ylabel('Density')
plt.show();
```



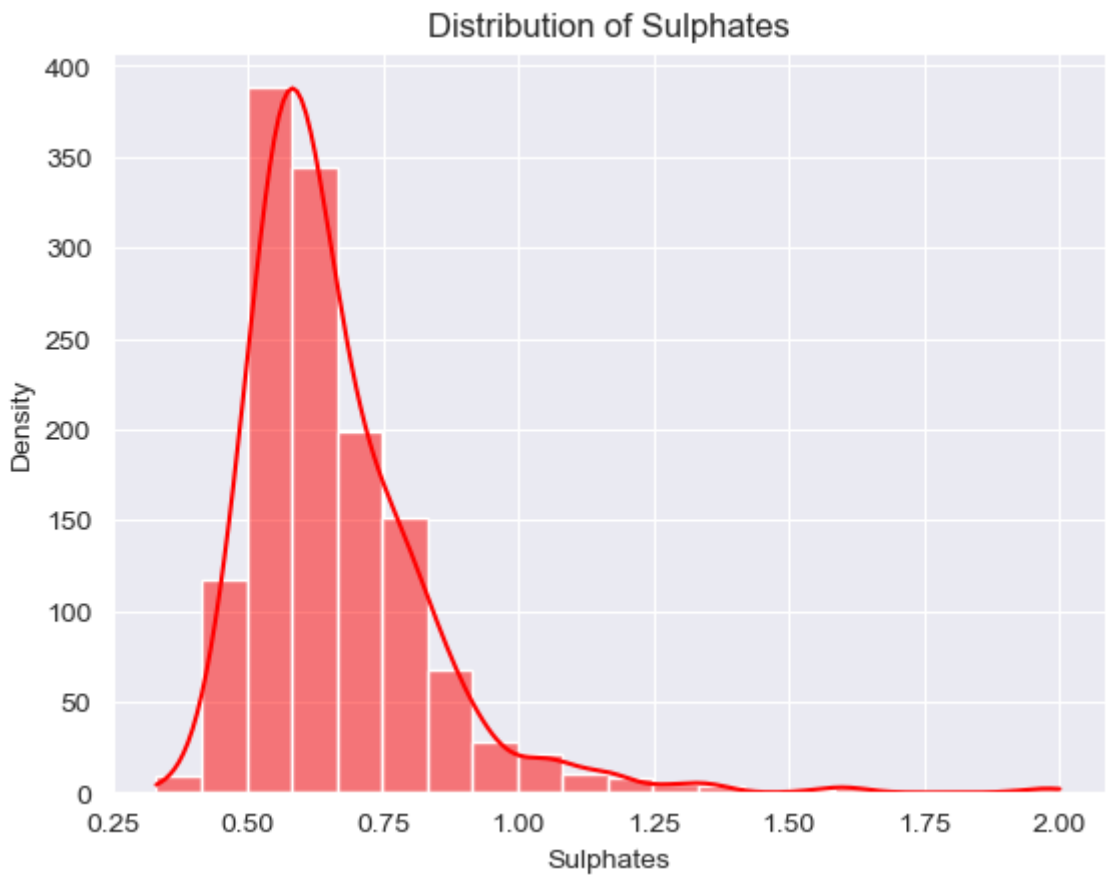
In [34]:
```
#Distribution of Fixed Acidity, Volatile Acidity, Citric Acid
plt.figure(figsize=(40,12))

plt.subplot(1, 3, 1)
sns.histplot(x=dff["fixed acidity"], bins=15, kde=True ,color='red')
plt.xlabel("Fixed Acidity")
plt.ylabel("Density")
plt.title("Distribution of Fixed Acidity");

plt.subplot(1, 3, 2)
sns.histplot(x=dff["volatile acidity"], bins=15,kde=True ,color='red')
plt.xlabel("Volatile Acidity")
plt.ylabel("Density")
plt.title("Distribution of  Fixed Acidity");

plt.subplot(1, 3, 3)
sns.histplot(x=dff["citric acid"],bins=15, kde=True ,color='red')
plt.xlabel("Citric Acid")
plt.ylabel("Density")
plt.title("Distribution of Citric Acid");
plt.tight_layout()
plt.show();
```



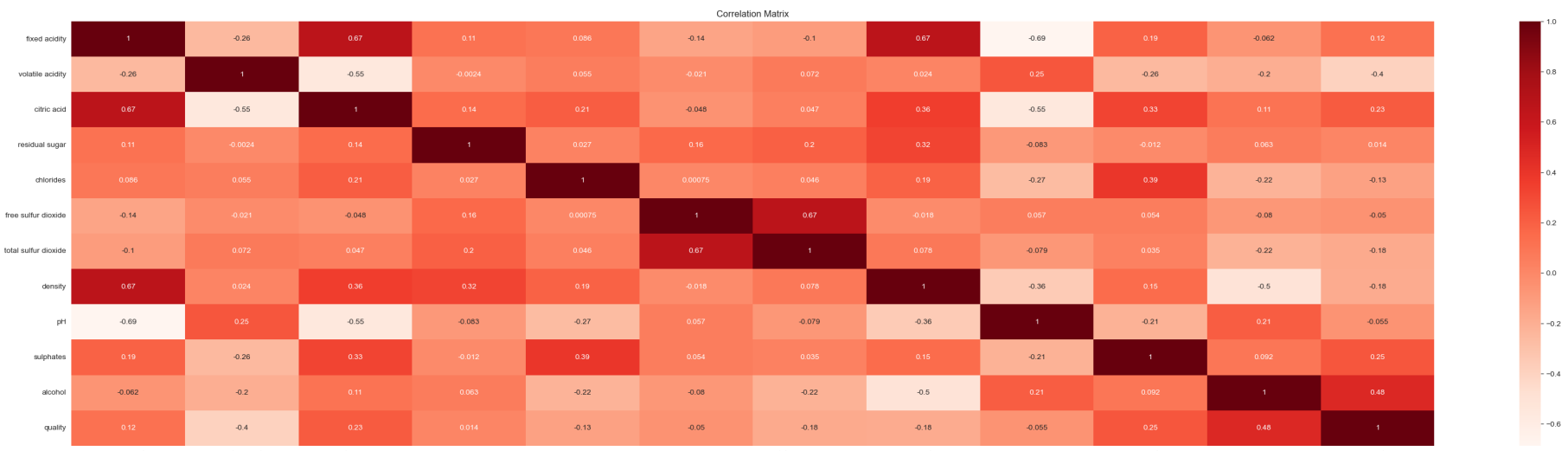
In [35]:
```
#Distribution of Residual sugar,Chlorides,Free sulfurdioxide
plt.figure(figsize=(30,12))

plt.subplot(1, 3, 1)
sns.histplot(x=dff["residual sugar"], bins=25,kde=True ,color='red')
plt.xlabel("Residual sugar")
plt.ylabel("Density")
plt.title("Distribution of Residual sugar");

plt.subplot(1, 3, 2)
sns.histplot(x=dff["chlorides"],bins=20, kde=True ,color='red')
plt.xlabel("Chloride")
plt.ylabel("Density")
plt.title("Distribution of Chloride")

plt.subplot(1, 3, 3)
sns.histplot(x=dff["free sulfur dioxide"],bins=20, kde=True ,color='red')
plt.xlabel("Free Sulfurdioxide")
plt.ylabel("Density")
plt.title("Distribution of Free Sulfurdioxide");
plt.tight_layout()
plt.show();
```



In [36]:
```
#Distribution of Sulphates
sns.histplot(x=dff["sulphates"],bins=20, kde=True ,color='red')
plt.xlabel("Sulphates")
plt.ylabel("Density")
plt.title("Distribution of Sulphates");
```



In [39]:
```
#Correlation Matrix
plt.figure(figsize=(40,10))
plt.title('Correlation Matrix')
sns.heatmap(dff.corr(), annot=True,cmap='Reds')
plt.show()
```



THANK YOU