```python
In [2]:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```python
In [31]:
df=pd.read_csv('C:/Users/HP/OneDrive/Desktop/WineQT.csv')
```

```python
In [32]:
df.head(5)
```

Out[32]:

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

```python
In [33]:
df.shape
```

```
Out[33]:
(1143, 13)
```

```python
In [34]:
#Check for any missing values in the dataset.
df.isna().sum()
```

```
Out[34]:
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
```

Distribution of Wine Quality: a. Explore the distribution of wine quality ratings. b. Visualize it using a bar chart or histogram.

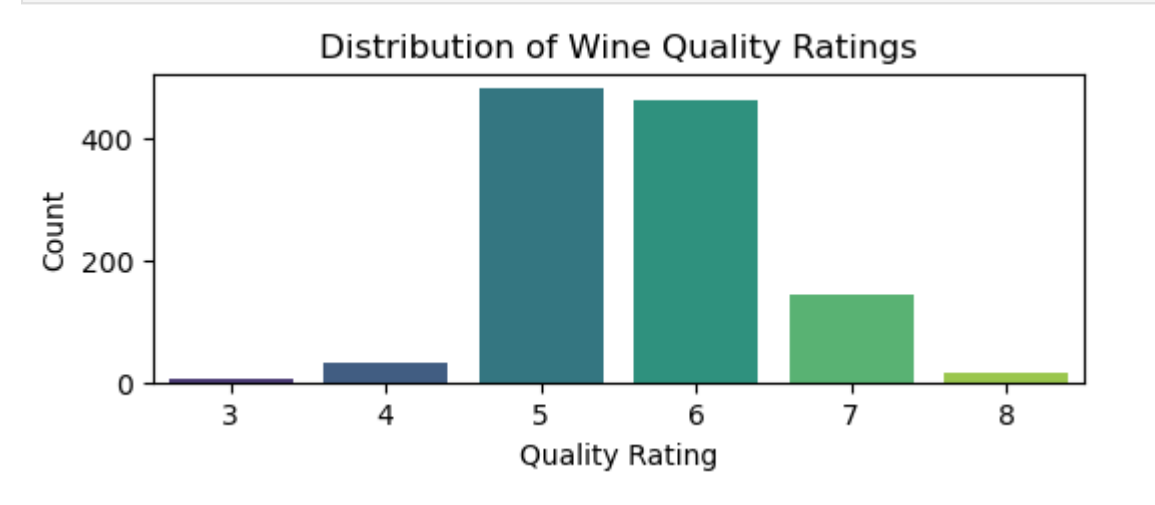
```python
In [ ]:
# a. Explore the distribution of wine quality ratings
```

```python
In [35]:
df['quality'].value_counts()
```

```
Out[35]:
5    483
6    462
7    143
4     33
8     16
3      6
Name: quality, dtype: int64
```

```python
In [ ]:
# b. Visualize it using a bar chart or histogram
```

```python
In [8]:
plt.figure(figsize=(6, 2))
sns.countplot(x='quality', data=df, palette='viridis')
plt.title('Distribution of Wine Quality Ratings')
plt.xlabel('Quality Rating')
plt.ylabel('Count')
plt.show()
```



3. Correlation Analysis: a. Examine the correlation between different attributes and wine quality. b. Create a correlation matrix and visualize it.

```python
In [ ]:
#a. Examine the correlation between different attributes and wine quality.
```

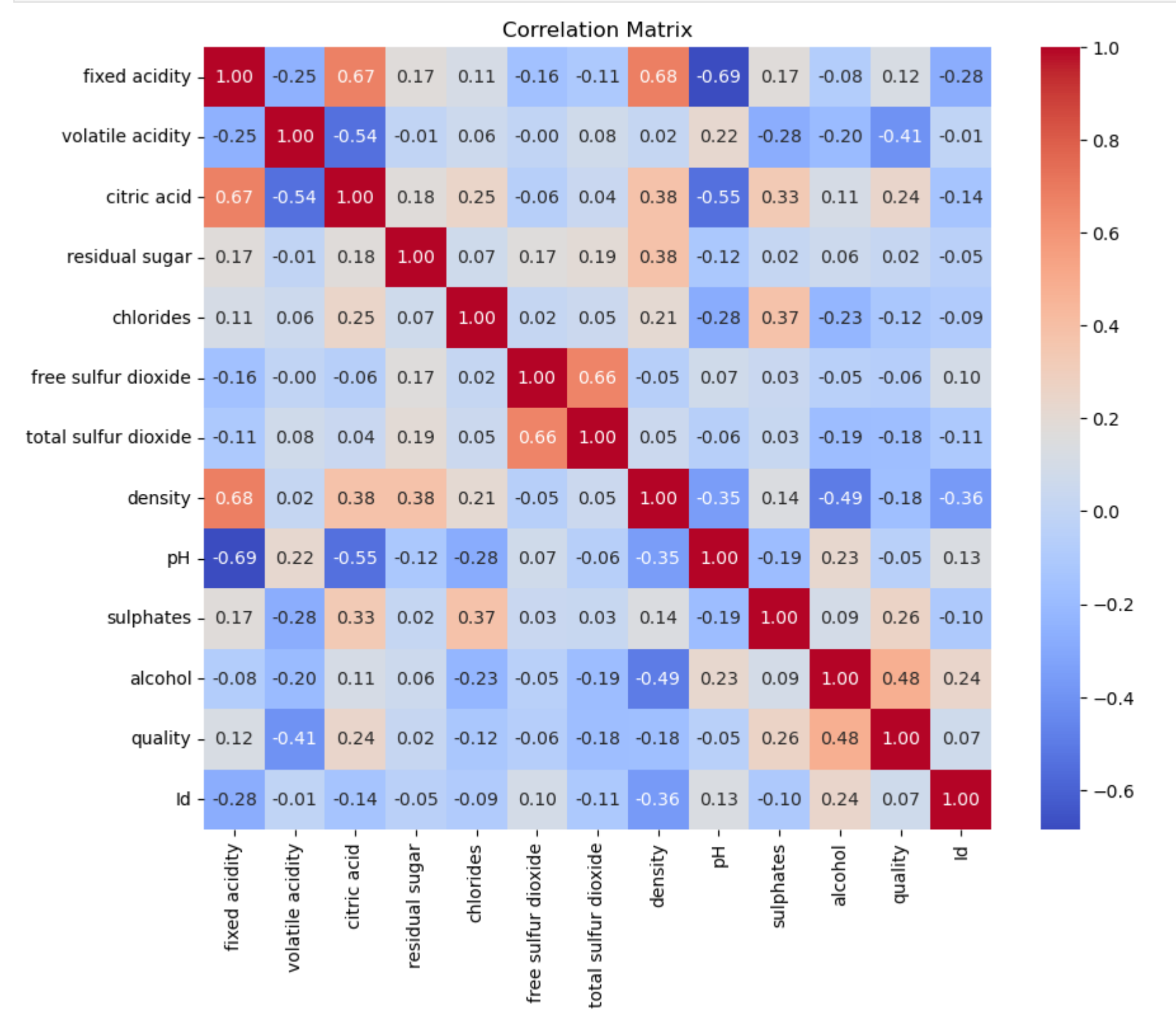
```python
In [10]:
df.corr()
```

Out[10]:

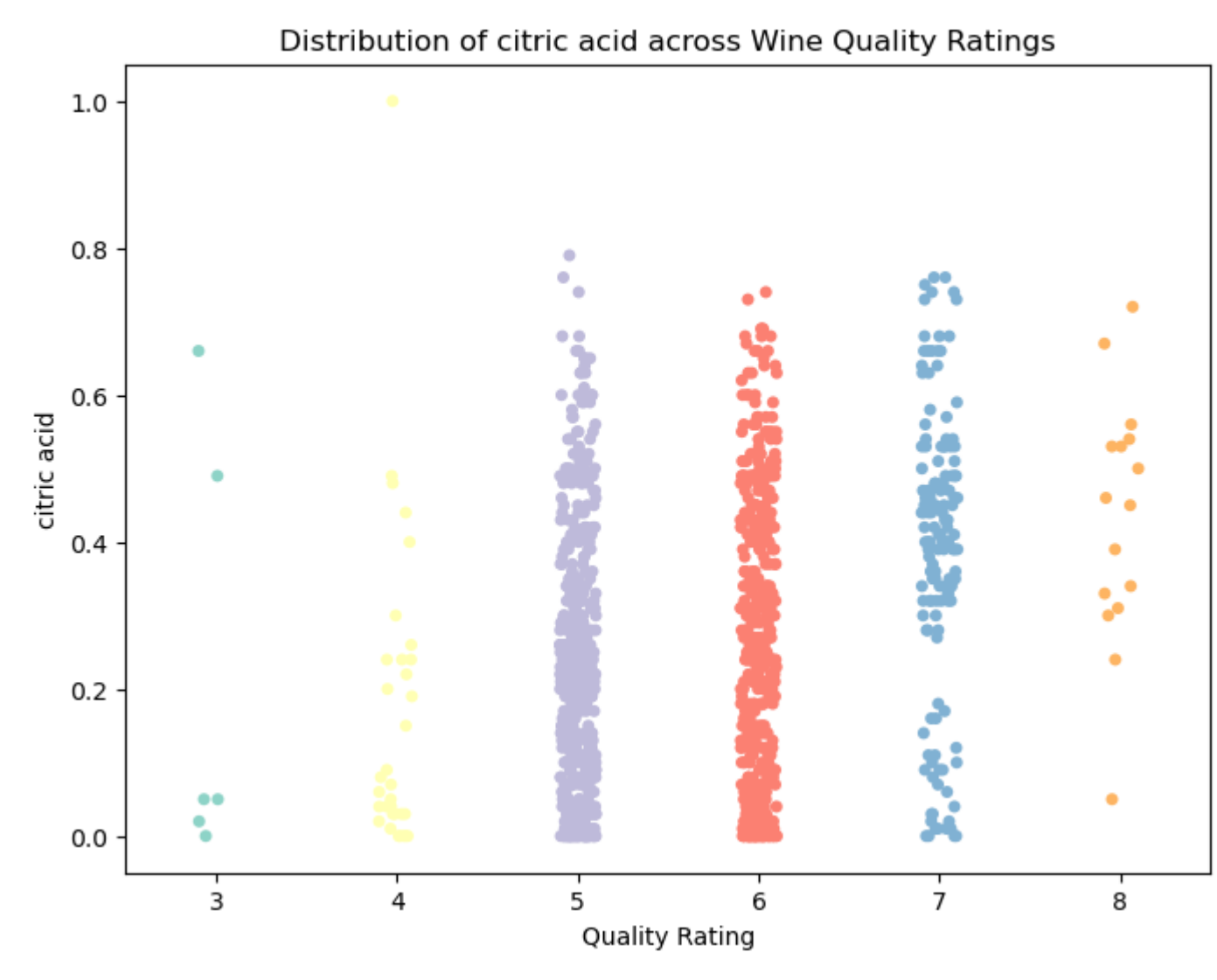
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Id
fixed acidity	1.000000	-0.250728	0.673157	0.171831	0.107889	-0.164831	-0.110628	0.681501	-0.685163	0.174592	-0.075055	0.121970	-0.275826
volatile acidity	-0.250728	1.000000	-0.544187	-0.005751	0.056336	-0.001962	0.077748	0.016512	0.221492	-0.276079	-0.203909	-0.407394	-0.007892
citric acid	0.673157	-0.544187	1.000000	0.175815	0.245312	-0.057589	0.036871	0.375243	-0.546339	0.331232	0.106250	0.240821	-0.139011
residual sugar	0.171831	-0.005751	0.175815	1.000000	0.070863	0.165339	0.190790	0.380147	-0.116959	0.017475	0.058421	0.022002	-0.046344
chlorides	0.107889	0.056336	0.245312	0.070863	1.000000	0.015280	0.048163	0.208901	-0.277759	0.374784	-0.229917	-0.124085	-0.088099
free sulfur dioxide	-0.164831	-0.001962	-0.057589	0.165339	0.015280	1.000000	0.661093	-0.054150	0.072804	0.034445	-0.047095	-0.063260	0.095268
total sulfur dioxide	-0.110628	0.077748	0.036871	0.190790	0.048163	0.661093	1.000000	0.050175	-0.059126	0.026894	-0.188165	-0.183339	-0.107389
density	0.681501	0.016512	0.375243	0.380147	0.208901	-0.054150	0.050175	1.000000	-0.352775	0.143139	-0.494727	-0.175208	-0.363926
pH	-0.685163	0.221492	-0.546339	-0.116959	-0.277759	0.072804	-0.059126	-0.352775	1.000000	-0.185499	0.225322	-0.052453	0.132904
sulphates	0.174592	-0.276079	0.331232	0.017475	0.374784	0.034445	0.026894	0.143139	-0.185499	1.000000	0.094421	0.257710	-0.103954
alcohol	-0.075055	-0.203909	0.106250	0.058421	-0.229917	-0.047095	-0.188165	-0.494727	0.225322	0.094421	1.000000	0.484866	0.238087
quality	0.121970	-0.407394	0.240821	0.022002	-0.124085	-0.063260	-0.183339	-0.175208	-0.052453	0.257710	0.484866	1.000000	0.069708
Id	-0.275826	-0.007892	-0.139011	-0.046344	-0.088099	0.095268	-0.107389	-0.363926	0.132904	-0.103954	0.238087	0.069708	1.000000

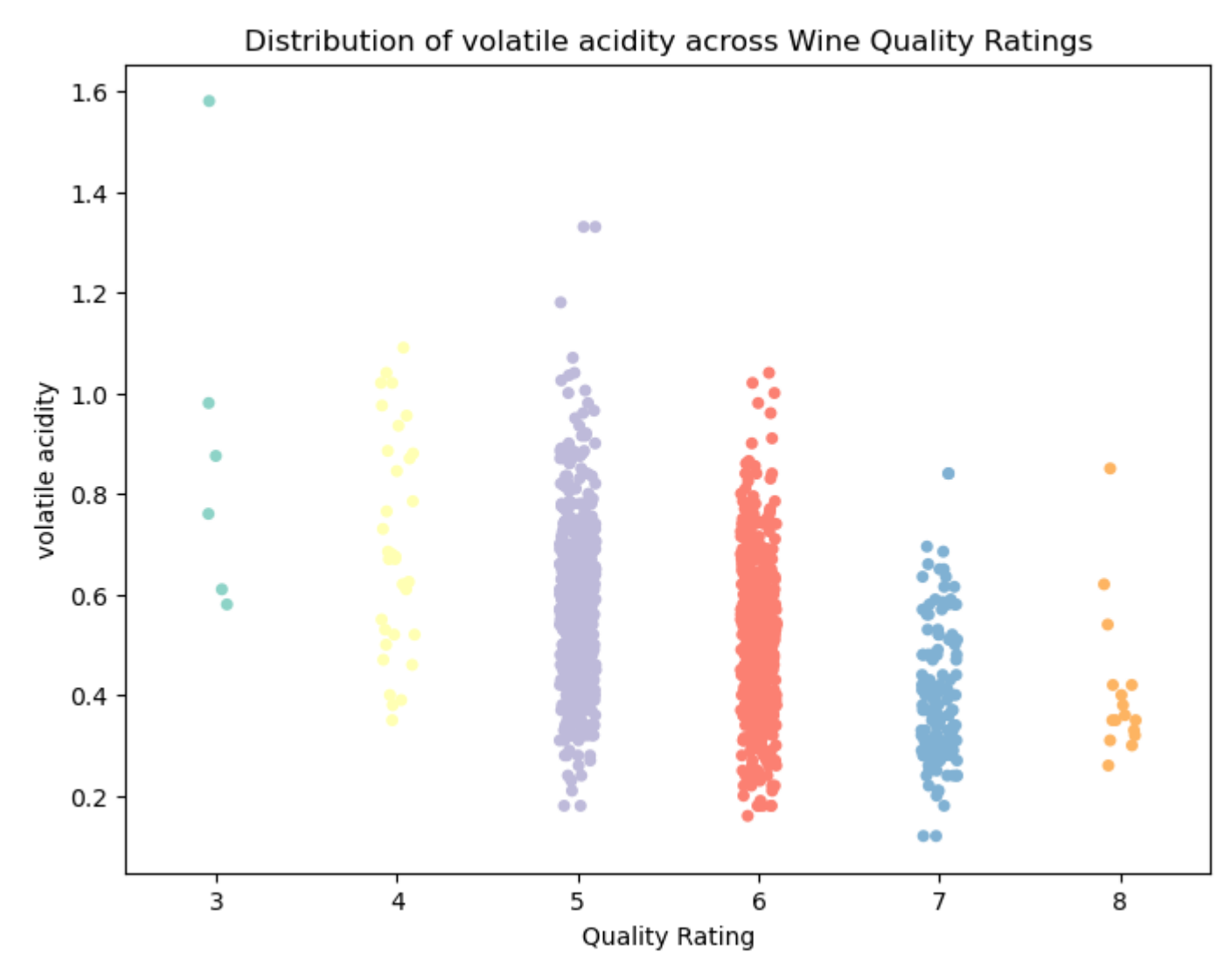
```python
In [ ]:
#b. Create a correlation matrix and visualize it
```

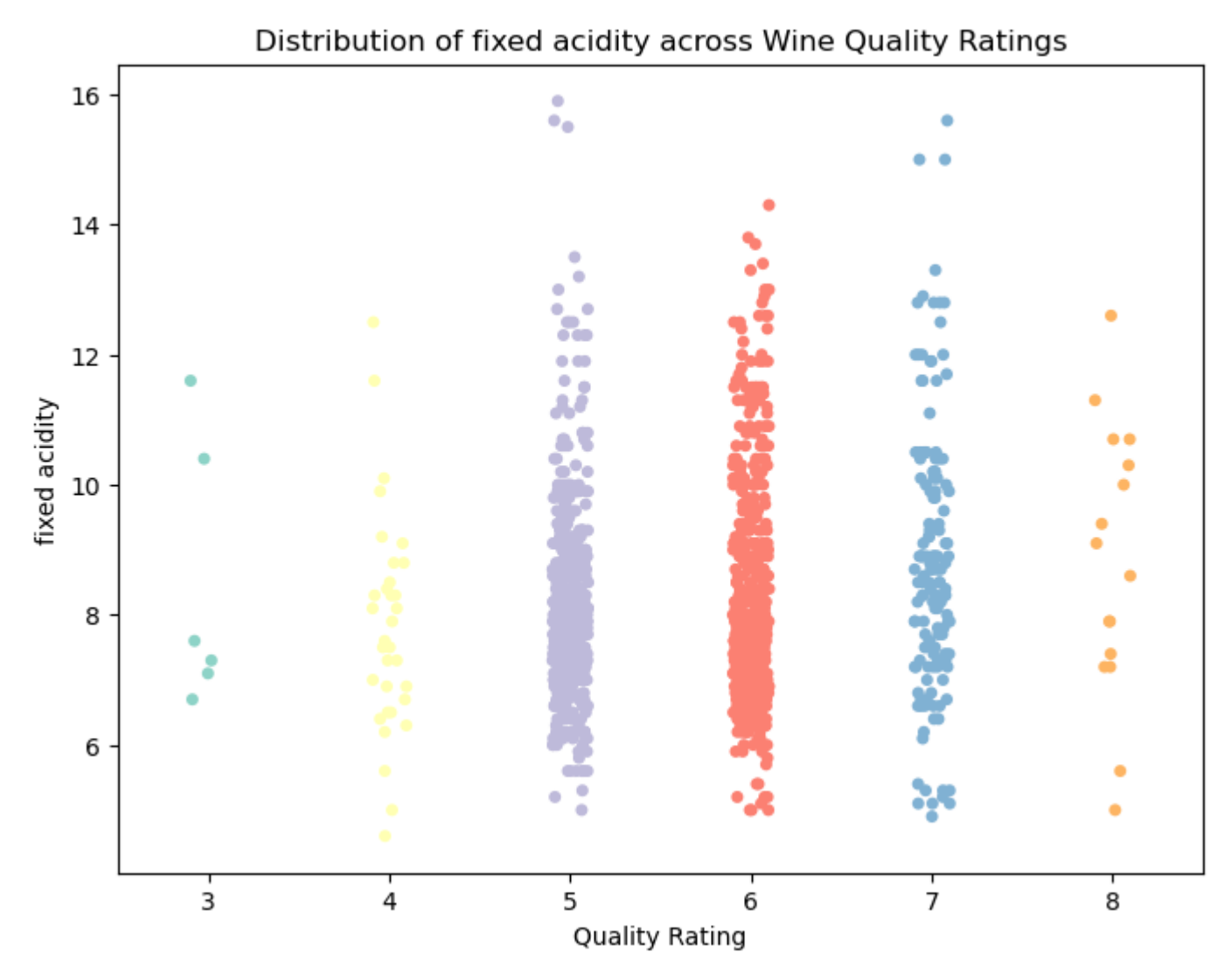
```python
In [11]:
plt.figure(figsize=(10,8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```

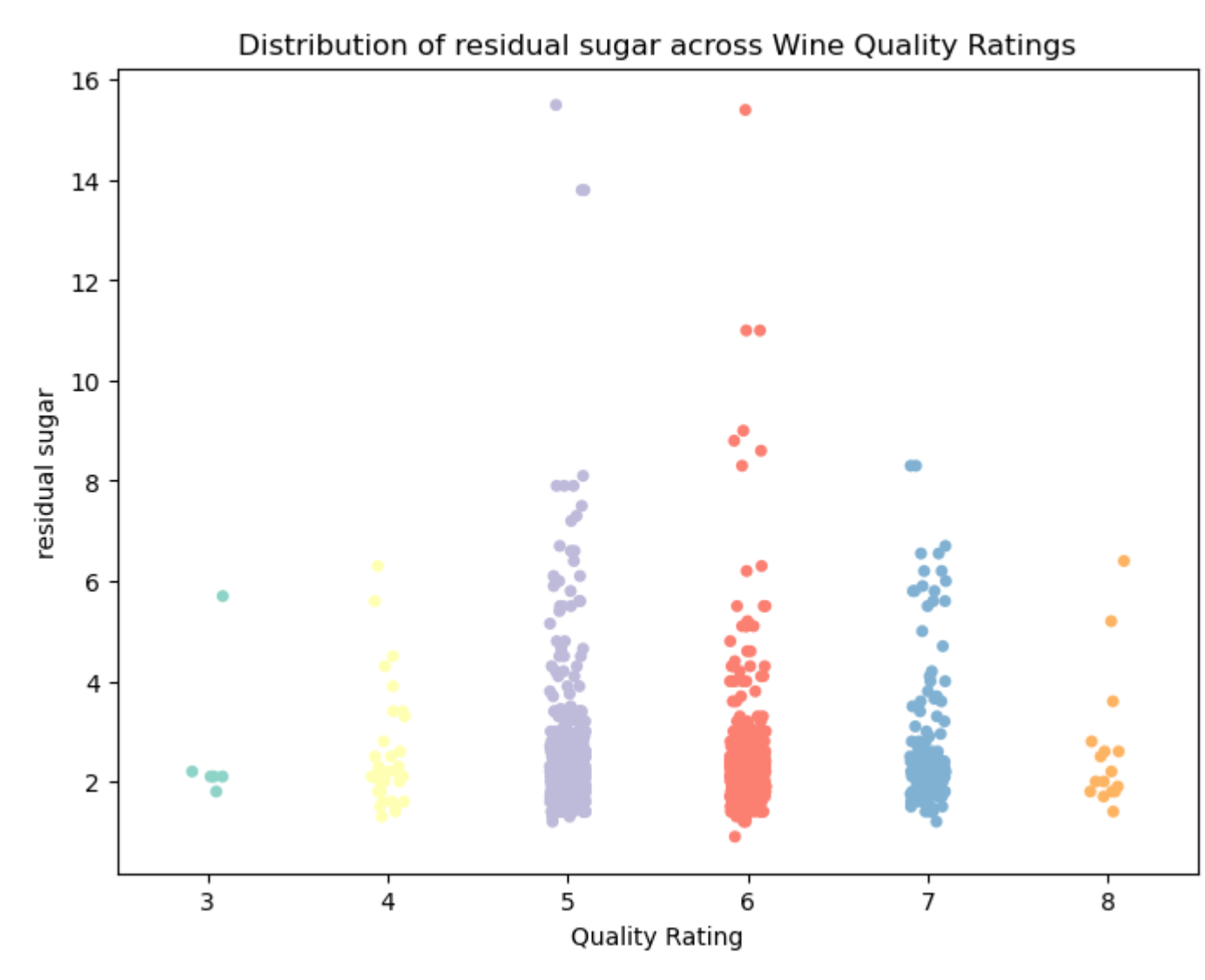



```python
In [26]:
# 4. Feature Analysis
## a. Analyze the distribution of key features such as alcohol content, acidity, etc.
## b. Compare the features across different quality ratings

features_of_interest = ['alcohol', 'citric acid', 'volatile acidity', 'fixed acidity', 'residual sugar']

# Loop through each feature
for feature in features_of_interest:
    # Create a new figure (plot) with a specified size
    plt.figure(figsize=(8, 6))

    # Create a strip plot (similar to a scatter plot) for the current feature
    sns.stripplot(x='quality', y=feature, data=df, palette='Set3')

    # Set the title of the plot
    plt.title(f'Distribution of {feature} across Wine Quality Ratings')

    # Label the x-axis as 'Quality Rating'
    plt.xlabel('Quality Rating')

    # Label the y-axis as the current feature
    plt.ylabel(feature)

    # Show the plot
    plt.show()
```

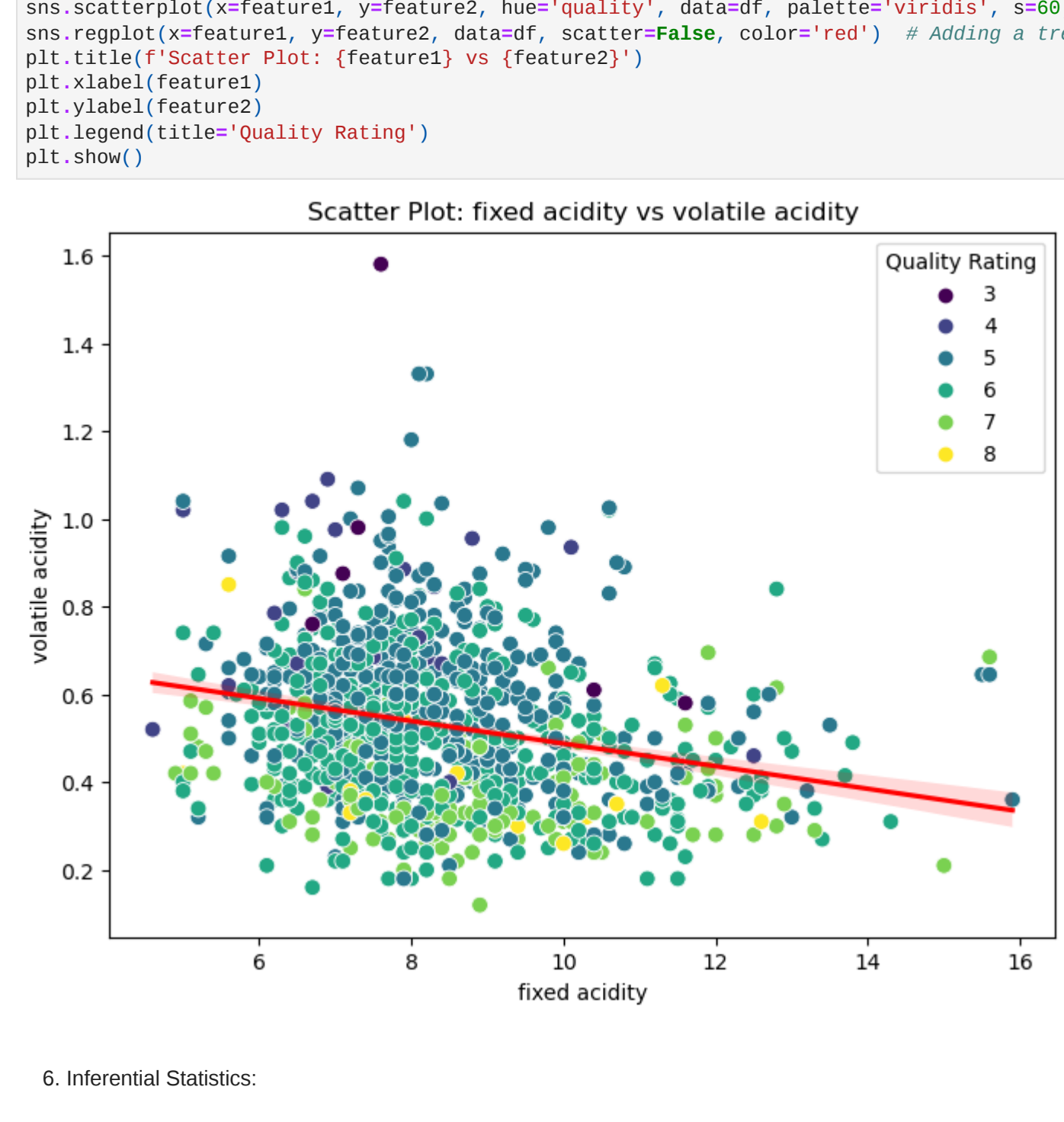









```python
In [28]:
# 5. Visualizing Relationships - Simple Scatter Plot
## Create scatter plots to visualize relationships between two features
## Use color-coding to distinguish between different quality ratings

# Selected two features for the scatter plot
feature1 = 'fixed acidity'
feature2 = 'volatile acidity'

# Create a scatter plot
plt.figure(figsize=(8, 6))
sns.scatterplot(x=feature1, y=feature2, hue='quality', data=df, palette='viridis', s=60)
sns.regplot(x=feature1, y=feature2, data=df, scatter=False, color='red')  # Adding a trend line
plt.title(f'Scatter Plot: {feature1} vs {feature2}')
plt.xlabel(feature1)
plt.ylabel(feature2)
plt.legend(title='Quality Rating')
plt.show()
```



5. Inferential Statistics

a. Calculate summary statistics (mean, median, standard deviation) for a chosen feature

```python
In [ ]:
# Chosen feature
chosen_feature = 'alcohol'

# Calculate summary statistics
mean_value = df[chosen_feature].mean()
median_value = df[chosen_feature].median()
std_dev_value = df[chosen_feature].std()

# Display summary statistics
print(f"Mean {chosen_feature}: {mean_value:.2f}")
print(f"Median {chosen_feature}: {median_value:.2f}")
print(f"Standard deviation {chosen_feature}: {std_dev_value:.2f}")
```

```
Mean alcohol: 10.44
Median alcohol: 10.20
Standard deviation alcohol: 1.08
```