```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv('Wine Quality Dataset.csv')

data.head()

   fixed acidity  volatile acidity  citric acid  residual sugar
chlorides  \
0            7.0              0.27         0.36            20.7
0.045
1            6.3              0.30         0.34             1.6
0.049
2            8.1              0.28         0.40             6.9
0.050
3            7.2              0.23         0.32             8.5
0.058
4            7.2              0.23         0.32             8.5
0.058

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates
\
0                 45.0                 170.0   1.0010  3.00       0.45

1                 14.0                 132.0   0.9940  3.30       0.49

2                 30.0                  97.0   0.9951  3.26       0.44

3                 47.0                 186.0   0.9956  3.19       0.40

4                 47.0                 186.0   0.9956  3.19       0.40


   alcohol  quality
0      8.8        6
1      9.5        6
2     10.1        6
3      9.9        6
4      9.9        6

data.shape

(4898, 12)

data.index

RangeIndex(start=0, stop=4898, step=1)

data.columns
```

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

```
data.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
```

Observations from Task 1

There are 4898 rows and 12 columns in the data.Each row contains the details of the types of acids present in white-wine and the quality

The features in the data set are:

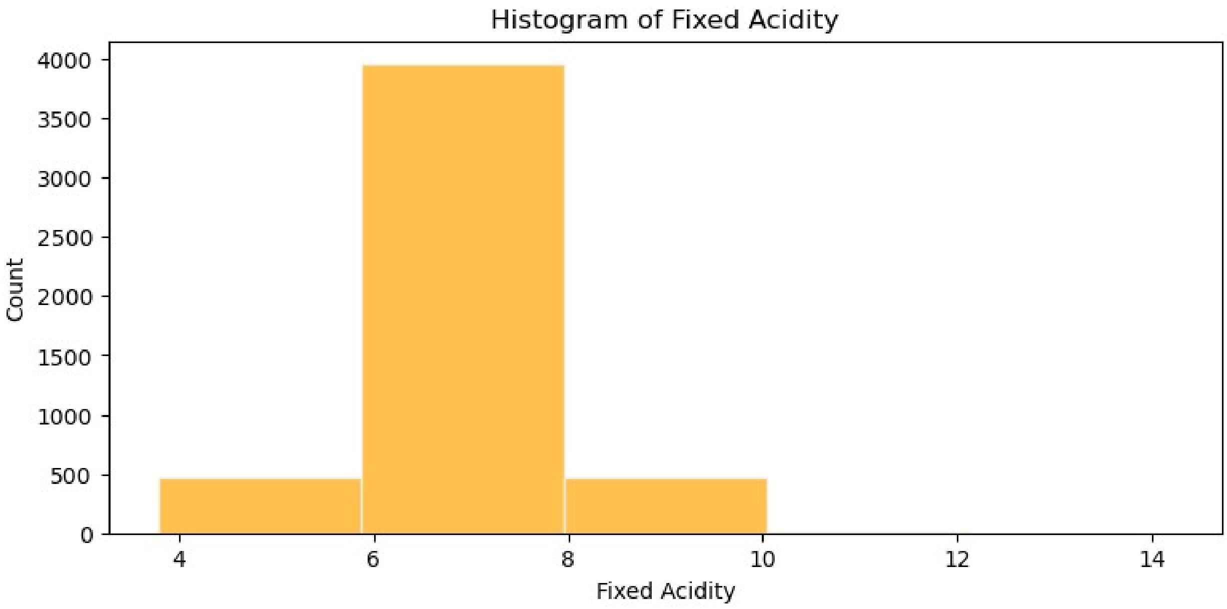
- Different acids and their Quality

Task 2 - View the distributions of the various features in the data set and calculate their central tendencies

#We will now look at the distributions of the various features in the data set

#We will also calculate appropriate measures of central tendency for these features

```
# Create a histogram of the "Fixed acidity" feature

plt.figure(figsize = (9,4))

sns.histplot(data = data ,x = 'fixed acidity', color = 'orange',
                 edgecolor = 'linen', alpha = 0.7, bins = 5)

plt.title("Histogram of Fixed Acidity")
plt.xlabel('Fixed Acidity')
plt.ylabel('Count')
plt.show()
```



Observations

We observe that the histogram is normally distributed.

The maximum count of values for fixed acidity lies in between 6 to 8.

Let's see the measures of central tendency in working!

1. Mean
2. Median
3. Mode

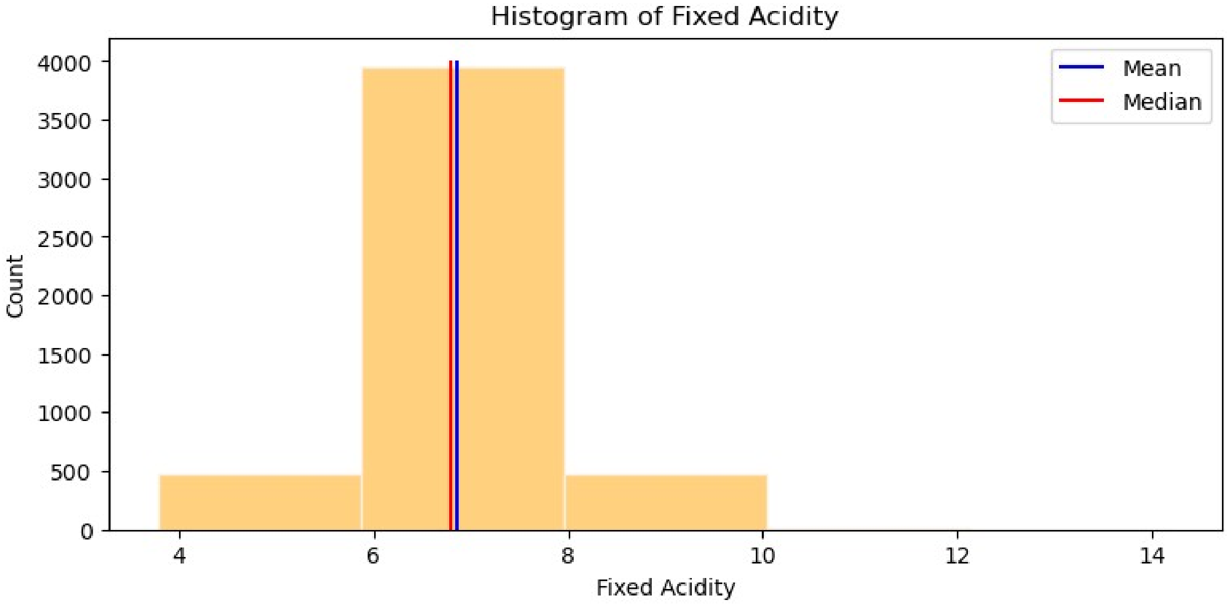
```
round(data['fixed acidity'].mean(),2)
```

```
6.85
```

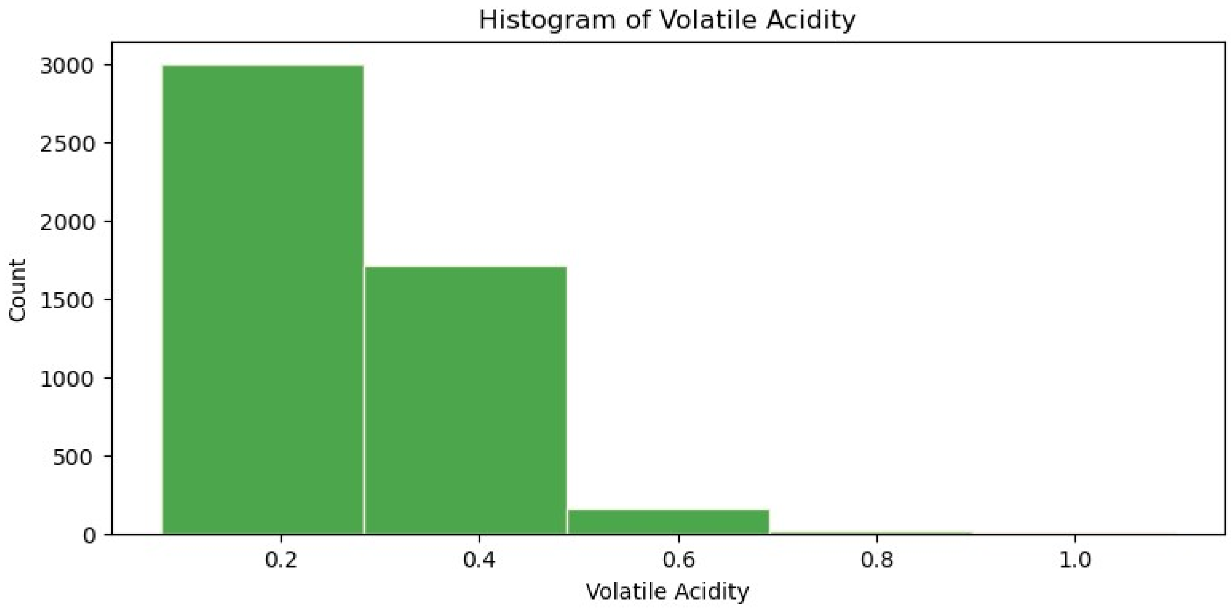
```
data['fixed acidity'].median()
```

```
6.8
```

```
plt.figure(figsize = (9,4))

sns.histplot(data = data ,x = 'fixed acidity', color = 'orange',
             edgecolor = 'linen', alpha = 0.5, bins = 5)

plt.title("Histogram of Fixed Acidity")
plt.xlabel('Fixed Acidity')
plt.ylabel('Count')
plt.vlines(data['fixed acidity'].mean(), ymin = 0, ymax = 4000,
colors='blue', label='Mean')
plt.vlines(data['fixed acidity'].median(), ymin = 0, ymax = 4000,
colors='red', label='Median')
plt.legend()
plt.show()
```

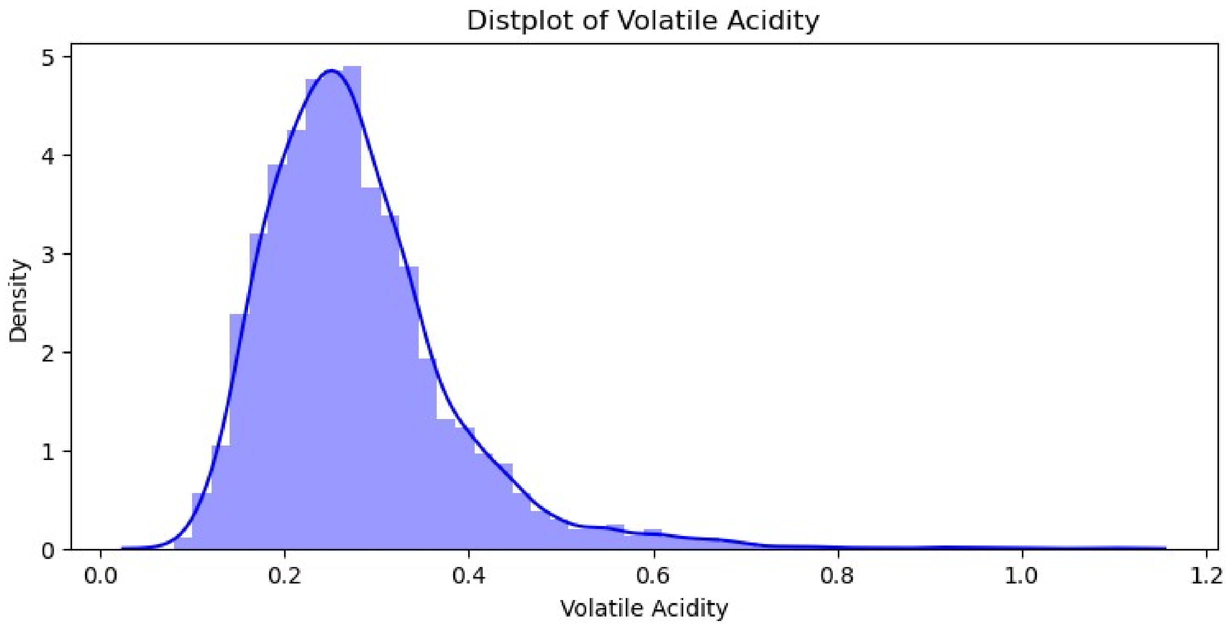


```
plt.figure(figsize = (9,4))

sns.histplot(data = data ,x = 'volatile acidity', color = 'green',
             edgecolor = 'linen', alpha = 0.7, bins = 5)

plt.title("Histogram of Volatile Acidity")
plt.xlabel('Volatile Acidity')
plt.ylabel('Count')
plt.show()
```



```python
# Plot distplot using 'Volatile acidity' feature

plt.figure(figsize = (9,4))

sns.distplot(data['volatile acidity'], color = 'blue')

plt.title("Distplot of Volatile Acidity")
plt.xlabel('Volatile Acidity')
plt.ylabel('Density')
plt.show()
```

```
C:\Users\Bharath\AppData\Local\Temp\ipykernel_19512\3796699363.py:5:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['volatile acidity'], color = 'blue')
```



Observation:

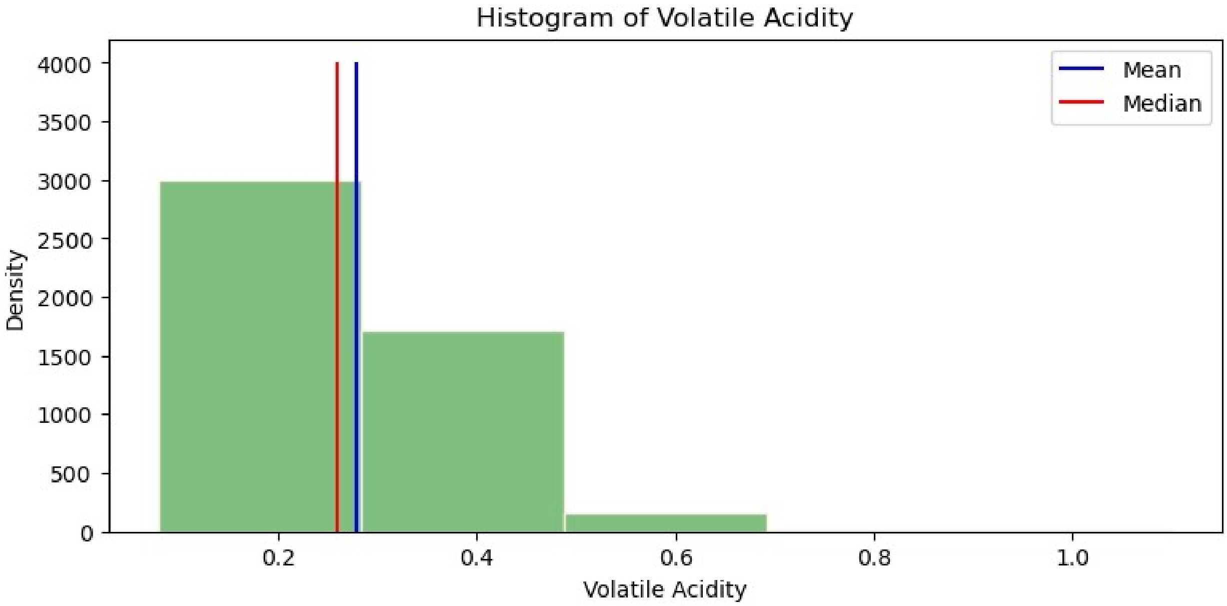
The above plot shows the normal distribution.

The normal distribution is described by the mean and the standard deviation.

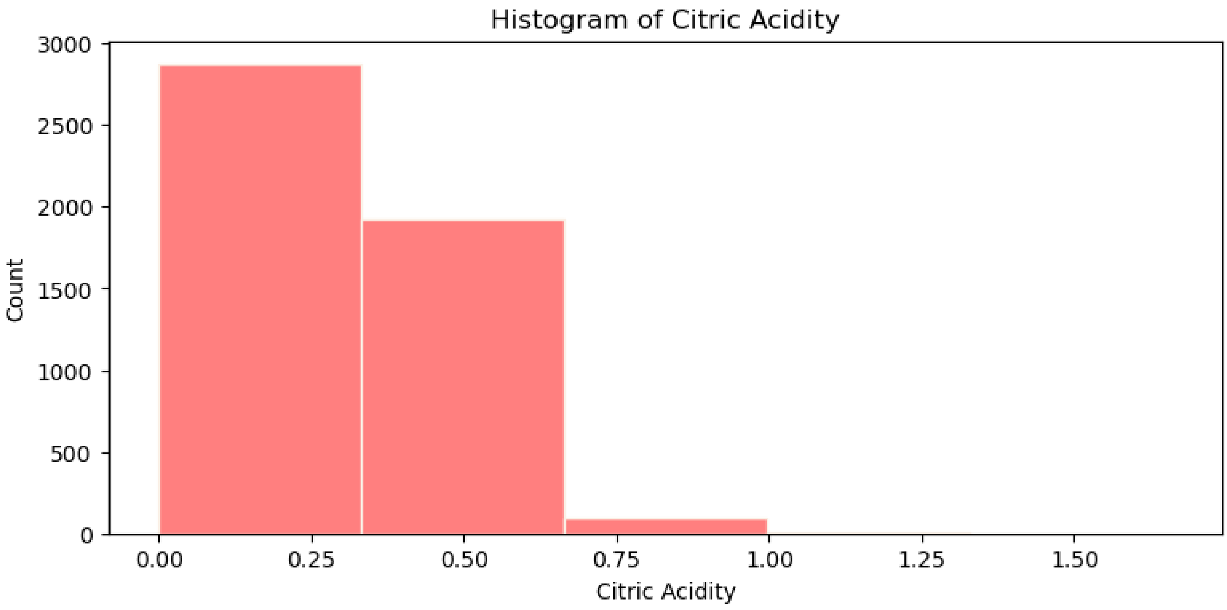
The normal distribution is often referred to as a 'bell curve' because of it's shape:

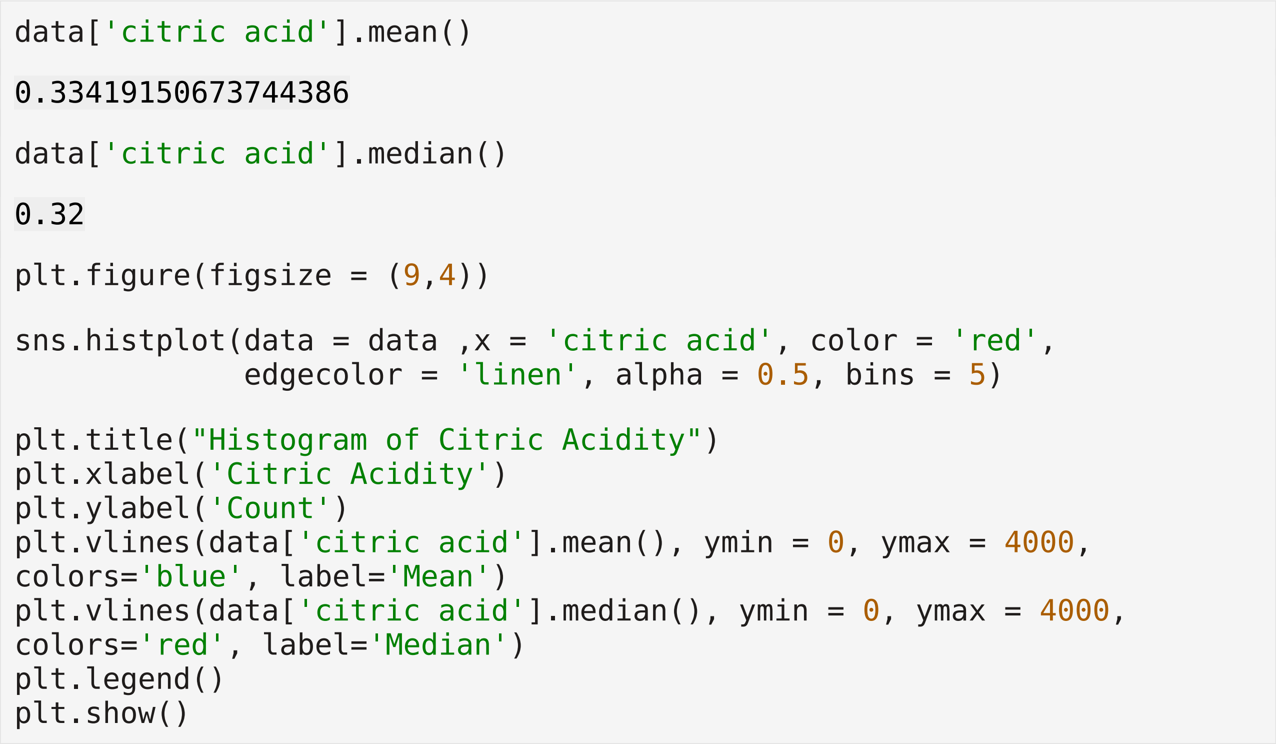
- The median and mean are equal
- It has only one mode
- It is symmetric, meaning it decreases the same amount on the left and the right of the centre

```
data['volatile acidity'].skew()

1.5769795029952025

data['volatile acidity'].mean()

0.27824111882400976

data['volatile acidity'].median()

0.26

plt.figure(figsize = (9,4))

sns.histplot(data = data ,x = 'volatile acidity', color = 'green',
             edgecolor = 'linen', alpha = 0.5, bins = 5)

plt.title("Histogram of Volatile Acidity")
plt.xlabel('Volatile Acidity')
plt.ylabel('Density')
plt.vlines(data['volatile acidity'].mean(), ymin = 0, ymax = 4000,
colors='blue', label='Mean')
plt.vlines(data['volatile acidity'].median(), ymin = 0, ymax = 4000,
colors='red', label='Median')
plt.legend()
plt.show()
```



```
plt.figure(figsize = (9,4))

sns.histplot(data = data ,x = 'citric acid', color = 'red',
             edgecolor = 'linen', alpha = 0.5, bins = 5)

plt.title("Histogram of Citric Acidity")
plt.xlabel('Citric Acidity')
plt.ylabel('Count')
plt.show()
```





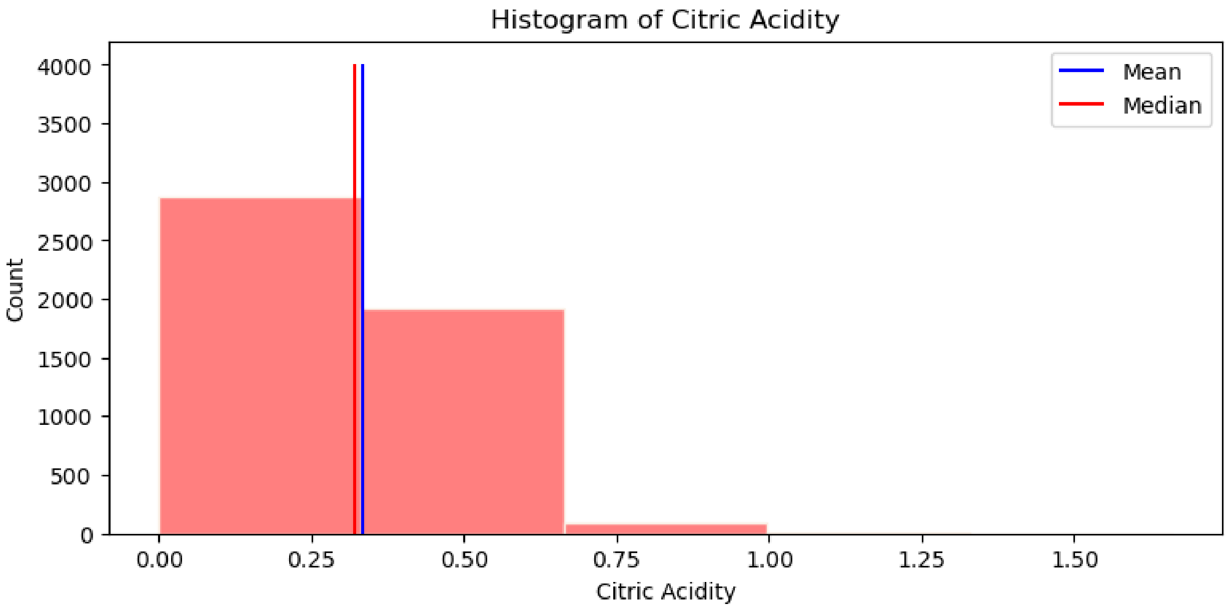
```
data['citric acid'].mean()
```

```
0.33419150673744386
```

```
data['citric acid'].median()
```

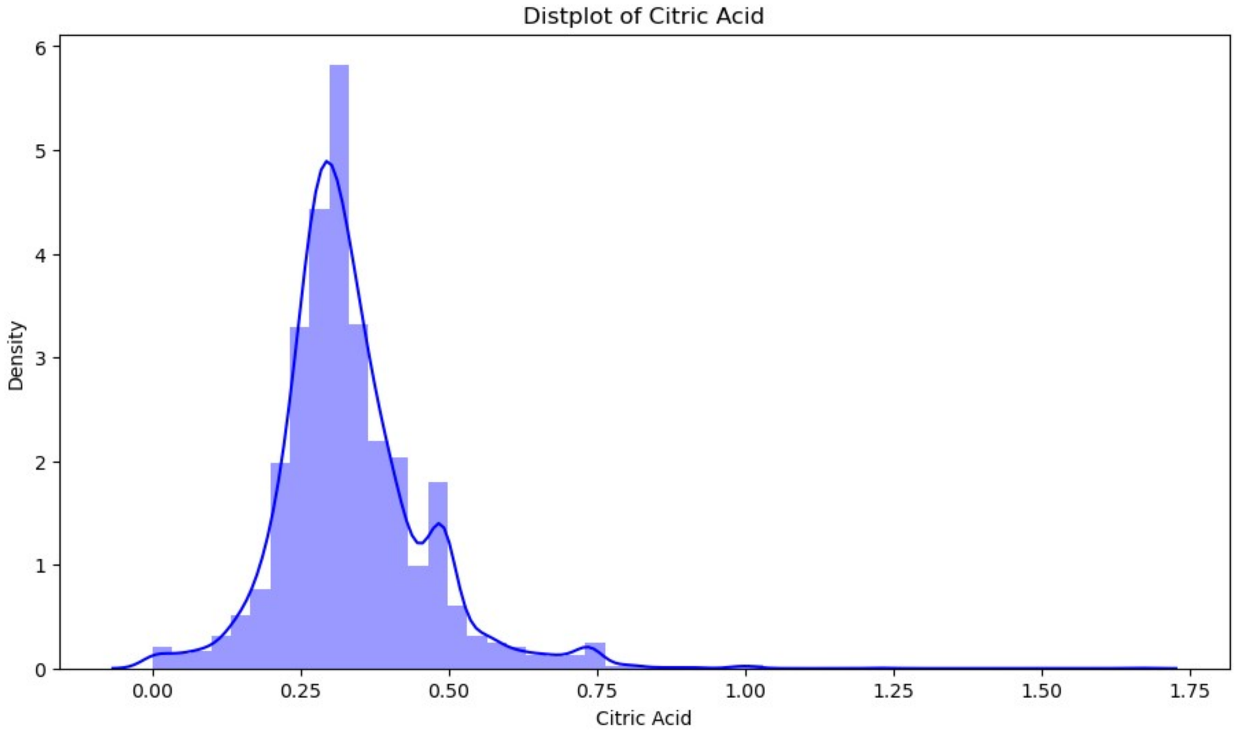
```
0.32
```

```
plt.figure(figsize = (9,4))

sns.histplot(data = data ,x = 'citric acid', color = 'red',
             edgecolor = 'linen', alpha = 0.5, bins = 5)

plt.title("Histogram of Citric Acidity")
plt.xlabel('Citric Acidity')
plt.ylabel('Count')
plt.vlines(data['citric acid'].mean(), ymin = 0, ymax = 4000,
colors='blue', label='Mean')
plt.vlines(data['citric acid'].median(), ymin = 0, ymax = 4000,
colors='red', label='Median')
plt.legend()
plt.show()
```



```
# Calculate distplot using 'Citric Acidity' feature

plt.figure(figsize = (11,6))

sns.distplot(data['citric acid'], color = 'blue')

plt.title("Distplot of Citric Acid")
plt.xlabel('Citric Acid')
plt.ylabel('Density')
plt.show()
```

```
C:\Users\Bharath\AppData\Local\Temp\ipykernel_19512\409944871.py:5:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['citric acid'], color = 'blue')
```



```
quality = pd.DataFrame(data['quality'].value_counts())
```

```
quality.index
```

```
Index([6, 5, 7, 8, 4, 3, 9], dtype='int64', name='quality')
```

```
data['quality'].value_counts()
```

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

```
data['quality'].value_counts().index[0]
```

```
6
```

```
# Create a new Pandas Series called "rep_acid" that contains the 
details of the representative quality for the different types of acids

rep_acid = pd.DataFrame(index = ['fixed acidity','volatile 
acidity','citric acid','quality'],
                     data = [data['fixed 
acidity'].mean(),data['volatile acidity'].mean(),
                             data['citric
acid'].mean(),data['quality'].value_counts().index[0]])

rep_acid
```

```
                          0
fixed acidity     6.854788
volatile acidity  0.278241
citric acid       0.334192
quality           6.000000
```

Final Conclusions

- From the given data, we can use simple visualisations to get a sense of how data are distributed.
- We can use various measures of central tendency such as mean, median and mode to represent a group of observations.
- The type of central tendency measure to use depends on the type and the distribution of the data