


```
# Import necessary packages and load `winequality_edited.csv`
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline

import requests
import pandas as pd
import io



# Correct URL to download raw CSV from GitHub
url = 'https://raw.githubusercontent.com/satyanarayanan102/OIBSIP/main/Wine%20Quality%20Analysis/wine_quality_analysis.csv'
response = requests.get(url)

data = pd.read_csv(io.StringIO(response.text))

data.head()
```




	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

Next steps: [Generate code with data](#) [View recommended plots](#) [New interactive sheet](#)

```
# Import necessary packages and load `winequality_edited.csv`
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline

# Assuming 'wine_data' is the DataFrame from your previous cell
wine_data.info()
```

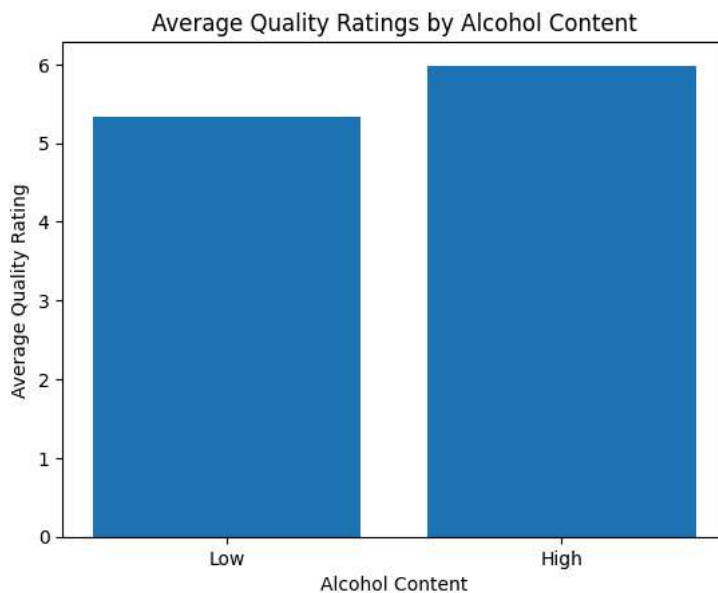


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1143 entries, 0 to 1142
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   fixed acidity          1143 non-null   float64
1   volatile acidity       1143 non-null   float64
2   citric acid            1143 non-null   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
```

```
# Use query to select each group and get its mean quality
# Assuming 'wine_data' is the DataFrame from your previous cell
median = wine_data['alcohol'].median()
low = wine_data.query('alcohol < {}'.format(median))
high = wine_data.query('alcohol >= {}'.format(median))

mean_quality_low = low['quality'].mean()
mean_quality_high = high['quality'].mean()
```

```
# Create a bar chart with proper labels
locations = [1, 2]
heights = [mean_quality_low, mean_quality_high]
labels = ['Low', 'High']
plt.bar(locations, heights, tick_label=labels)
plt.title('Average Quality Ratings by Alcohol Content')
plt.xlabel('Alcohol Content')
plt.ylabel('Average Quality Rating');
```



```
# Use query to select each group and get its mean quality
median_sugar = wine_data['residual sugar'].median() # Corrected column name
low_sugar = wine_data.query('`residual sugar` < {}'.format(median_sugar)) # Use backticks for column name with spaces
high_sugar = wine_data.query('`residual sugar` >= {}'.format(median_sugar))

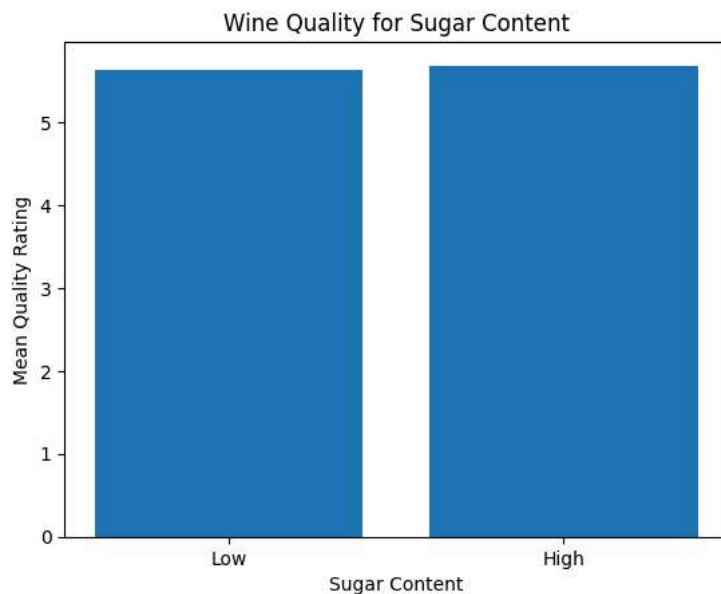
low_sugar_mean_quality = low_sugar['quality'].mean()
low_sugar_mean_quality
high_sugar_mean_quality = high_sugar['quality'].mean()
```

```
high_sugar_mean_quality
```



```
5.682804674457429
```

```
# Create a bar chart with proper labels
locations_sugar = [1,2]
points = [low_sugar_mean_quality, high_sugar_mean_quality]
labels_sugar = ['Low', 'High']
plt.bar(locations_sugar, points, tick_label=labels_sugar)
plt.title('Wine Quality for Sugar Content')
plt.xlabel('Sugar Content')
plt.ylabel('Mean Quality Rating');
```



```
# Use groupby to get the mean quality for each acidity level
wine_data.describe() # Assuming 'wine_data' is the DataFrame you want to describe
```



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide
count	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000
mean	8.311111	0.531339	0.268364	2.532152	0.086933	15.615486	45.900000
std	1.747595	0.179633	0.196686	1.355917	0.047267	10.250486	32.700000
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000
25%	7.100000	0.392500	0.090000	1.900000	0.070000	7.000000	21.000000
50%	7.900000	0.520000	0.250000	2.200000	0.079000	13.000000	37.000000
75%	9.100000	0.640000	0.420000	2.600000	0.090000	21.000000	61.000000

```
import pandas as pd
```

```
# Assuming 'wine_data' is your DataFrame, replace 'df' with 'wine_data'
wine_data['acidity_levels'] = pd.cut(wine_data['pH'], bin_edges, labels=bin_names)
```

```
# Checks for successful creation of this column
wine_data.head()
```



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulph
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	

```
# Find the mean quality of each acidity level with groupby
quality_acidity_mean = []
quality_acidity_mean = wine_data.groupby('acidity_levels').mean()['quality'] # Use 'wine_data' instead of 'df'
quality_acidity_mean
```



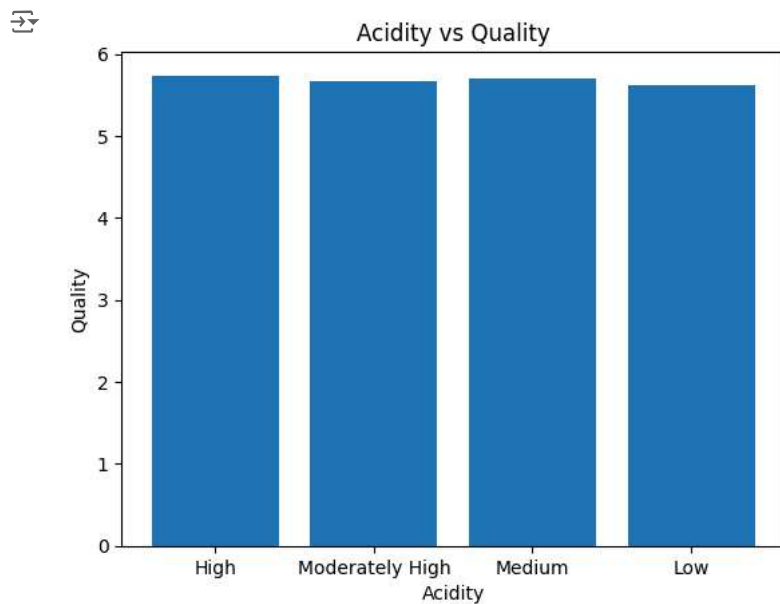
```
acidity_levels
High          5.735849
Moderately High  5.666667
Medium        5.704180
Low           5.609195
Name: quality, dtype: float64
```

```
acidity_mean = wine_data.groupby('acidity_levels').mean()['pH']
```

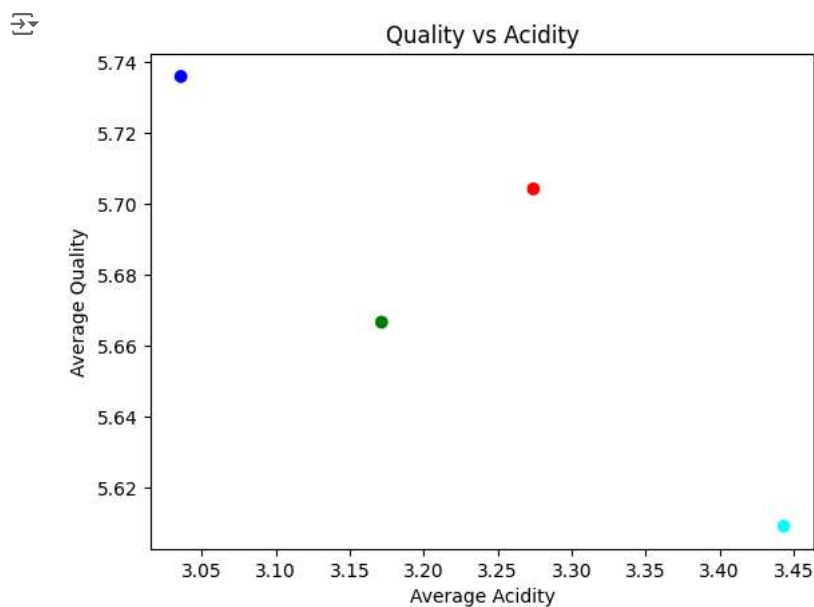
```
# Create a bar chart with proper labels
locations_pH = [1,2,3,4]
```

```
#points = [low_sugar_mean_quality, high_sugar_mean_quality]
#plt.bar(locations_sugar, points, tick_label=labels_sugar)
```

```
plt.bar(locations_pH,quality_acidity_mean, tick_label=bin_names)
plt.title('Acidity vs Quality')
plt.xlabel('Acidity')
plt.ylabel('Quality');
```



```
colors = ['blue', 'green', 'red', 'cyan', 'magenta', 'yellow', 'black', 'white']
plt.scatter(x=acidity_mean, y=quality_acidity_mean, color=[colors[i%len(colors)] for i in range(len(acidity_mean))])
plt.xlabel('Average Acidity')
plt.ylabel('Average Quality')
plt.title('Quality vs Acidity')
plt.show()
```



```
quality_acidity_mean
```

```
acidity_levels
High 5.735849
```

```
Moderately High    5.666667  
Medium              5.704180  
Low                 5.609195  
Name: quality, dtype: float64
```

```
import matplotlib.pyplot as plt  
plt.plot([5.7833, 5.7845, 5.8508, 5.8595])  
plt.ylabel('Quality Mean for Acidity')  
plt.xlabel('Acidity Levels')  
plt.show();
```

