

```
In [5]: import pandas as pd  
df = pd.read_csv('winequality-red.csv', sep=';')
```

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

Out[6]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.62
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.61
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56

```
In [8]: # Summary of the Data Set  
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 [11]: # Descriptive Summary of the Data Set  
df.describe()
```

Out[11]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	

In [12]: `df.shape`

Out[12]: (1599, 12)

In [13]: `# List Down all the Columns in the Data Set`
`df.columns`

Out[13]: Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
 'pH', 'sulphates', 'alcohol', 'quality'],
 dtype='object')

In [15]: `df['quality'].unique()`

Out[15]: array([5, 6, 7, 4, 8, 3])

In [16]: `# Missing Values in the Data Set`
`df.isnull().sum()`

Out[16]: 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 [18]: `# Duplicate Records in the Data Set`
`df[df.duplicated()]`

Out[18]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulph
4	7.4	0.700	0.00	1.90	0.076	11.0	34.0	0.99780	3.51	
11	7.5	0.500	0.36	6.10	0.071	17.0	102.0	0.99780	3.35	
27	7.9	0.430	0.21	1.60	0.106	10.0	37.0	0.99660	3.17	
40	7.3	0.450	0.36	5.90	0.074	12.0	87.0	0.99780	3.33	
65	7.2	0.725	0.05	4.65	0.086	4.0	11.0	0.99620	3.41	
...
1563	7.2	0.695	0.13	2.00	0.076	12.0	20.0	0.99546	3.29	
1564	7.2	0.695	0.13	2.00	0.076	12.0	20.0	0.99546	3.29	
1567	7.2	0.695	0.13	2.00	0.076	12.0	20.0	0.99546	3.29	
1581	6.2	0.560	0.09	1.70	0.053	24.0	32.0	0.99402	3.54	
1596	6.3	0.510	0.13	2.30	0.076	29.0	40.0	0.99574	3.42	

240 rows × 12 columns



In [19]: `# Removing the duplicates
df.drop_duplicates(inplace=True)`

In [20]: `df.shape`

Out[20]: (1359, 12)

In [21]: `# Correlation
df.corr()`

Out[21]:

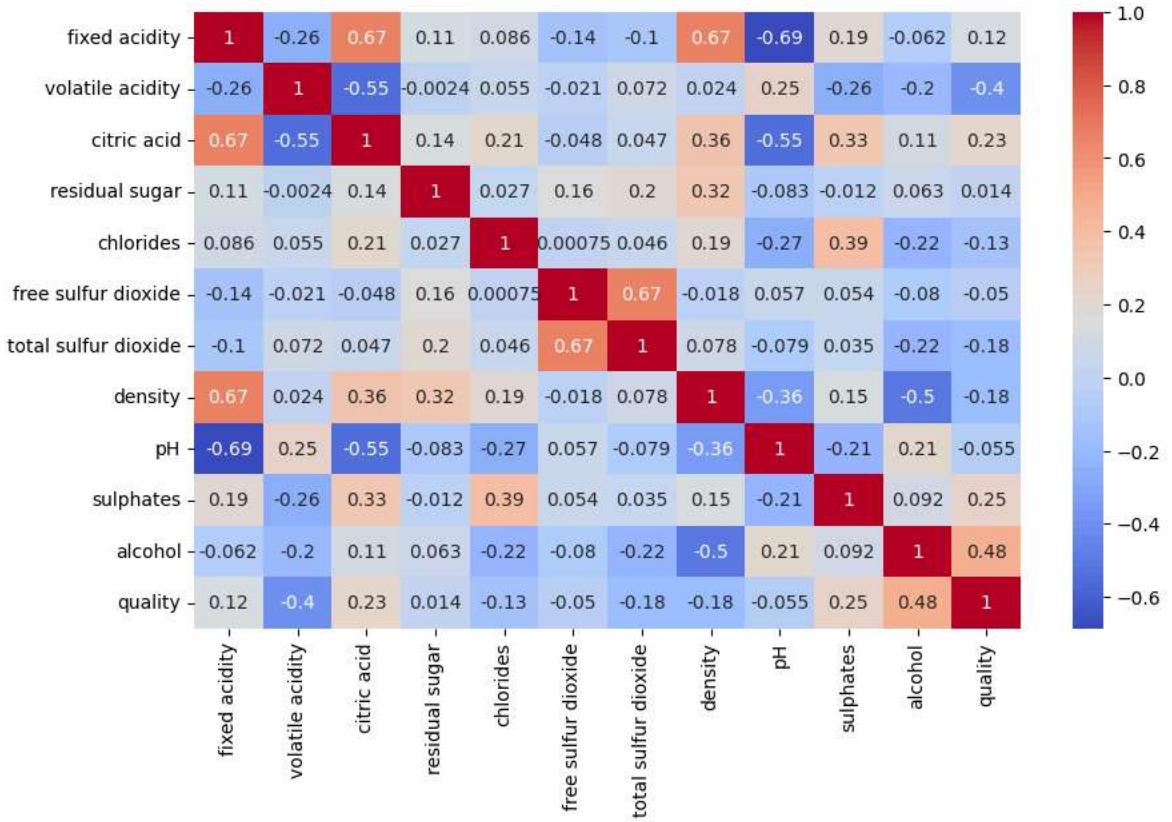
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide
fixed acidity	1.000000	-0.255124	0.667437	0.111025	0.085886	-0.140580	-0.103777
volatile acidity	-0.255124	1.000000	-0.551248	-0.002449	0.055154	-0.020945	0.071701
citric acid	0.667437	-0.551248	1.000000	0.143892	0.210195	-0.048004	0.047358
residual sugar	0.111025	-0.002449	0.143892	1.000000	0.026656	0.160527	0.201038
chlorides	0.085886	0.055154	0.210195	0.026656	1.000000	0.000749	0.045773
free sulfur dioxide	-0.140580	-0.020945	-0.048004	0.160527	0.000749	1.000000	0.667246
total sulfur dioxide	-0.103777	0.071701	0.047358	0.201038	0.045773	0.667246	1.000000
density	0.670195	0.023943	0.357962	0.324522	0.193592	-0.018071	0.078141
pH	-0.686685	0.247111	-0.550310	-0.083143	-0.270893	0.056631	-0.079257
sulphates	0.190269	-0.256948	0.326062	-0.011837	0.394557	0.054126	0.035291
alcohol	-0.061596	-0.197812	0.105108	0.063281	-0.223824	-0.080125	-0.217829
quality	0.119024	-0.395214	0.228057	0.013640	-0.130988	-0.050463	-0.177855

In [23]:

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10,6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```

Out[23]: <Axes: >



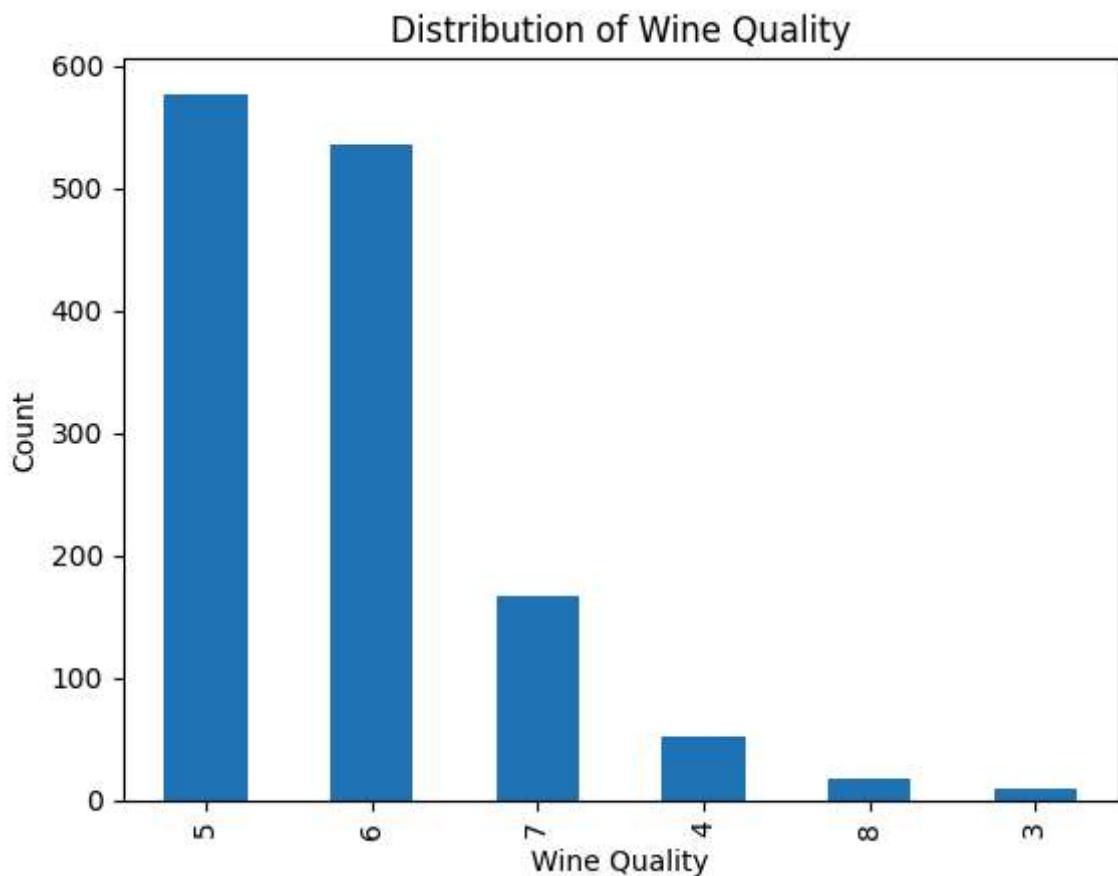
```
In [26]: # Visualization
df.quality.value_counts()

# Conclusion: It is an imbalanced dataset.
```

```
Out[26]: quality
5    577
6    535
7    167
4     53
8     17
3     10
Name: count, dtype: int64
```

```
In [28]: df.quality.value_counts().plot(kind='bar')
plt.xlabel('Wine Quality')
plt.ylabel('Count')
plt.title('Distribution of Wine Quality')
```

```
Out[28]: Text(0.5, 1.0, 'Distribution of Wine Quality')
```



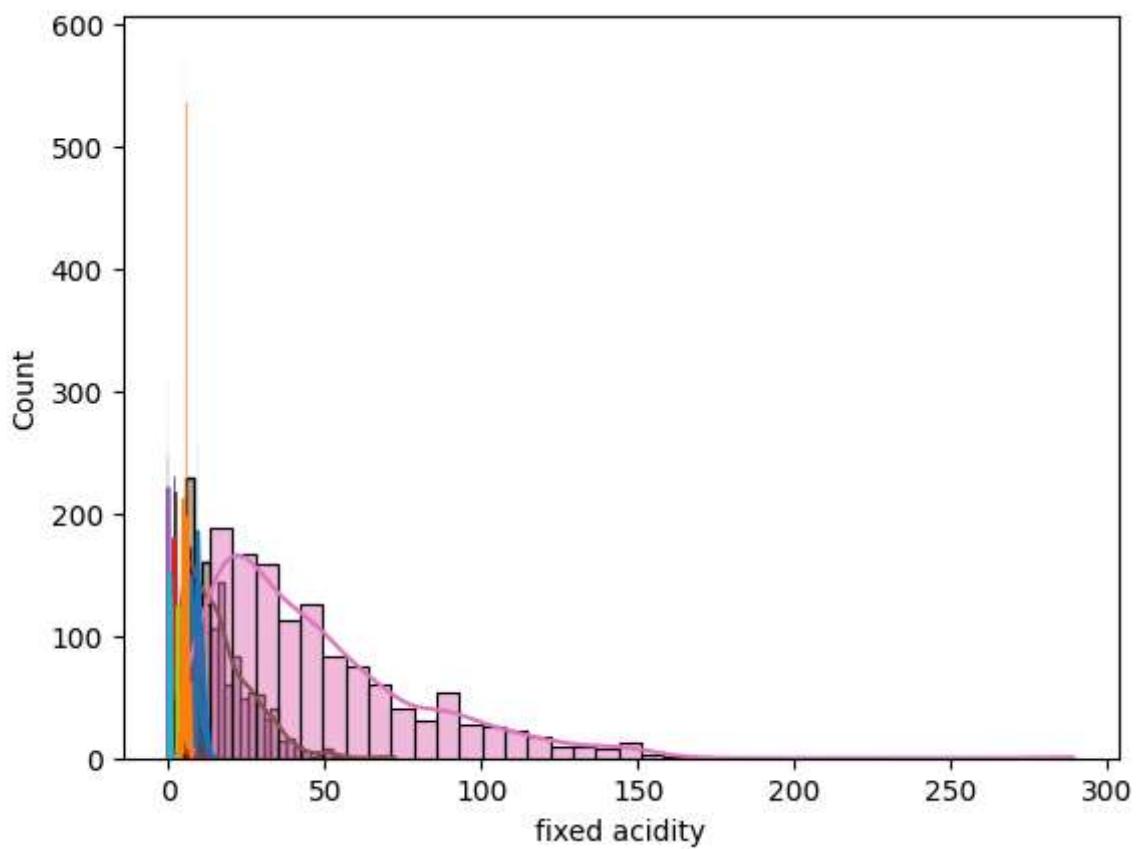
```
In [29]: df.head()
```

Out[29]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.62
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.61
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56

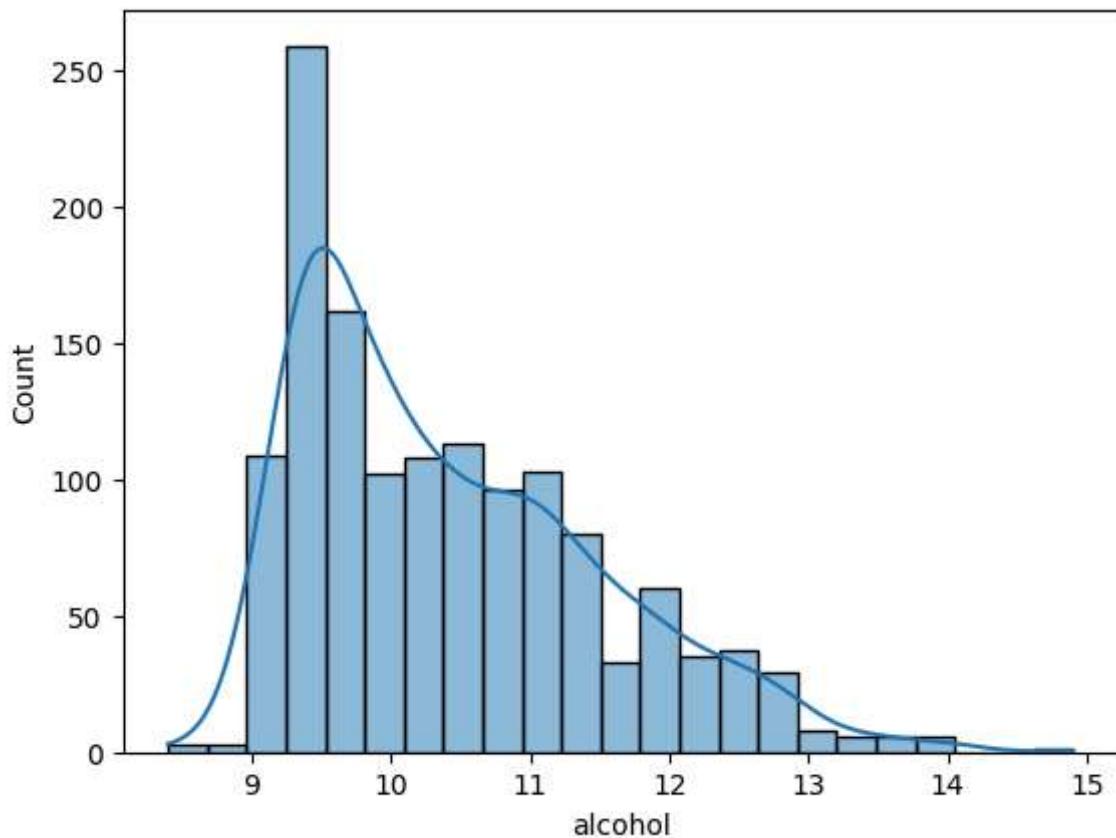


```
In [31]: for column in df.columns:  
    sns.histplot(df[column], kde=True)
```



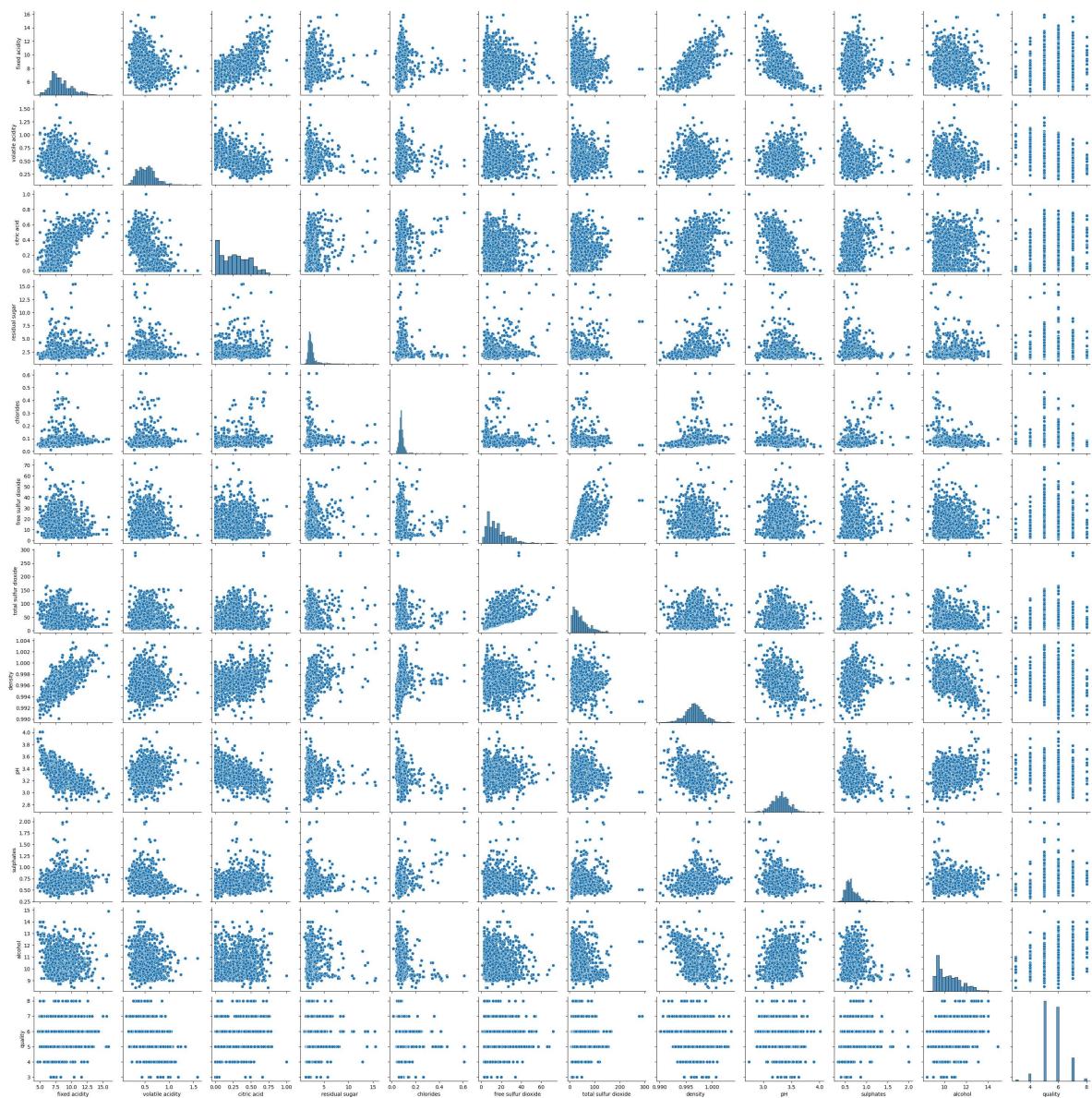
```
In [32]: sns.histplot(df['alcohol'], kde=True)
```

```
Out[32]: <Axes: xlabel='alcohol', ylabel='Count'>
```



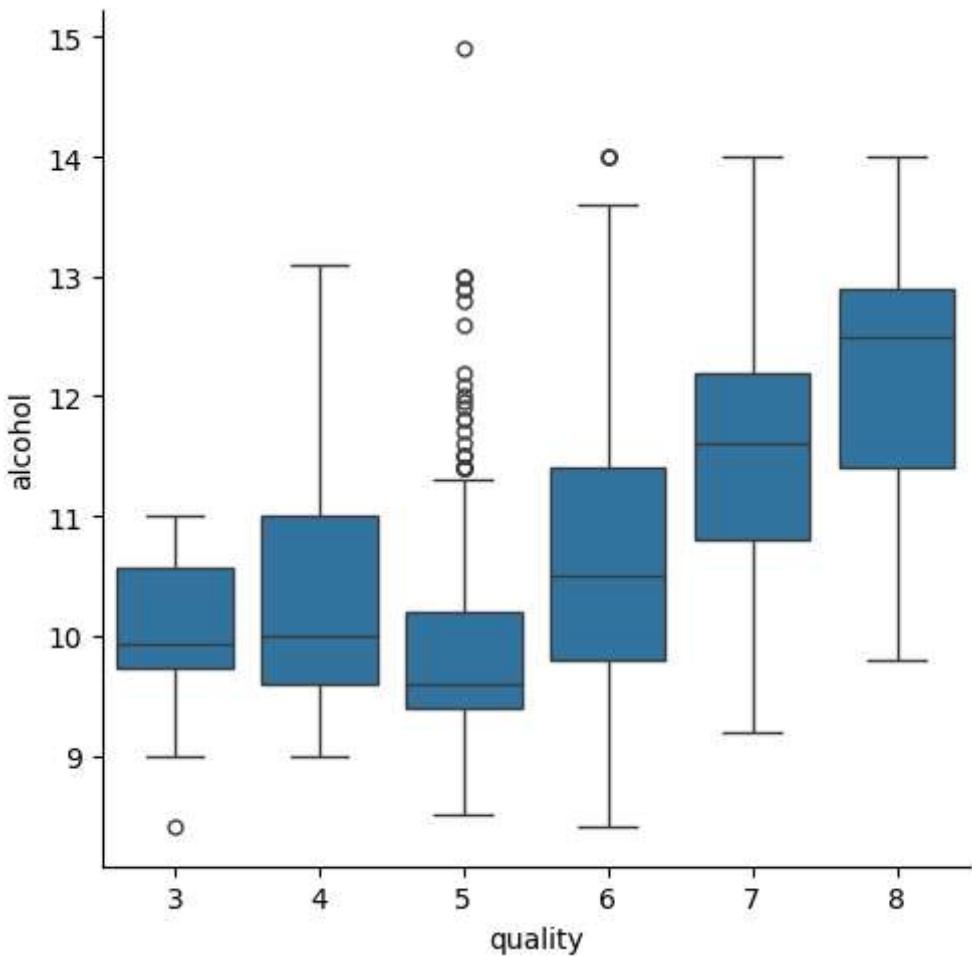
```
In [33]: # Univariate, Bivariate and Multivariate Analysis  
sns.pairplot(df)
```

Out[33]: <seaborn.axisgrid.PairGrid at 0x285b74ca2d0>



In [34]: # Categorical Plot
sns.catplot(x='quality', y='alcohol', data=df, kind='box')

Out[34]: <seaborn.axisgrid.FacetGrid at 0x285c1016840>



```
In [37]: # Numerical Plot  
sns.scatterplot(x='alcohol', y='pH', hue='quality', data=df)
```

```
Out[37]: <Axes: xlabel='alcohol', ylabel='pH'>
```

