Red wine Data Analysis

Introduction to the Wine Quality Dataset

Here's a polished version of your introduction that flows clearly and presents your work in the right order:

Introduction

I obtained the **Wine Quality dataset** from the **UCI Machine Learning Repository**, a widely used online resource for research and practice. This dataset has been particularly helpful in improving my **data analysis skills**, as it provides real-world physicochemical and sensory data related to Portuguese *Vinho Verde* wines.

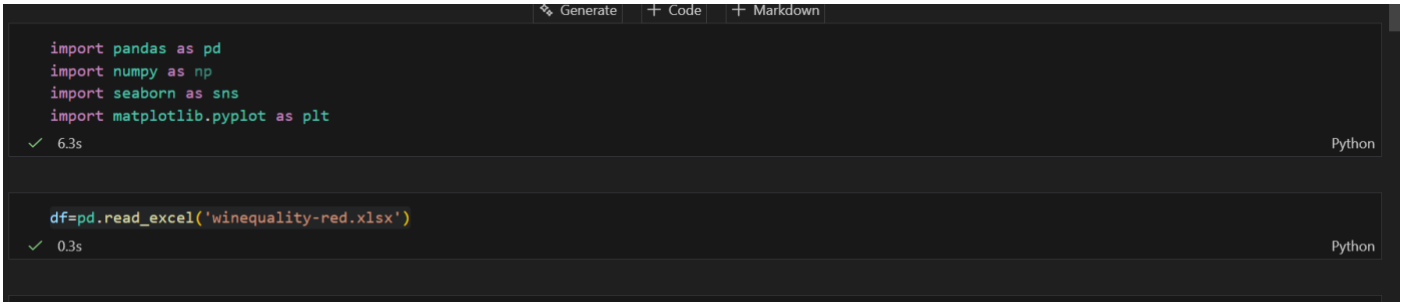
In my exploratory analysis tasks, I applied techniques such as **removing duplicates** and examining the relationships among the key attributes. The dataset includes **11 physicochemical variables** and one **output variable (quality score between 0 and 10)**. My analysis focused on the following attributes:

1. Fixed acidity
2. Volatile acidity
3. Citric acid
4. Residual sugar
5. Chlorides
6. Free sulfur dioxide
7. Total sulfur dioxide
8. Density
9. pH
10. Sulphates
11. Alcohol
12. Quality (sensory score)

By exploring these variables, I was able to practice techniques in **data cleaning, descriptive statistics, and correlation analysis**, which are essential for building predictive models and gaining insights into wine quality. This dataset not only strengthened my technical skills but also gave me practical exposure to handling real-world, imbalanced data suitable for regression and classification tasks.

Would you like me to also **expand this into a research-style introduction** (like one you'd put in a project report or thesis), where I connect your analysis to the original study by Cortez et al. (2009) and highlight its academic relevance?

Step 1: first I had to import python libraries such as pandas, NumPy, matplotlib, seaborn.



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

6.3s Python

```
df=pd.read_excel('winequality-red.xlsx')
```

0.3s Python

Step 2: read the excel file in python. Using pandas to create a data frame as df variable.

```
df=pd.read_excel('winequality-red.xlsx')
```

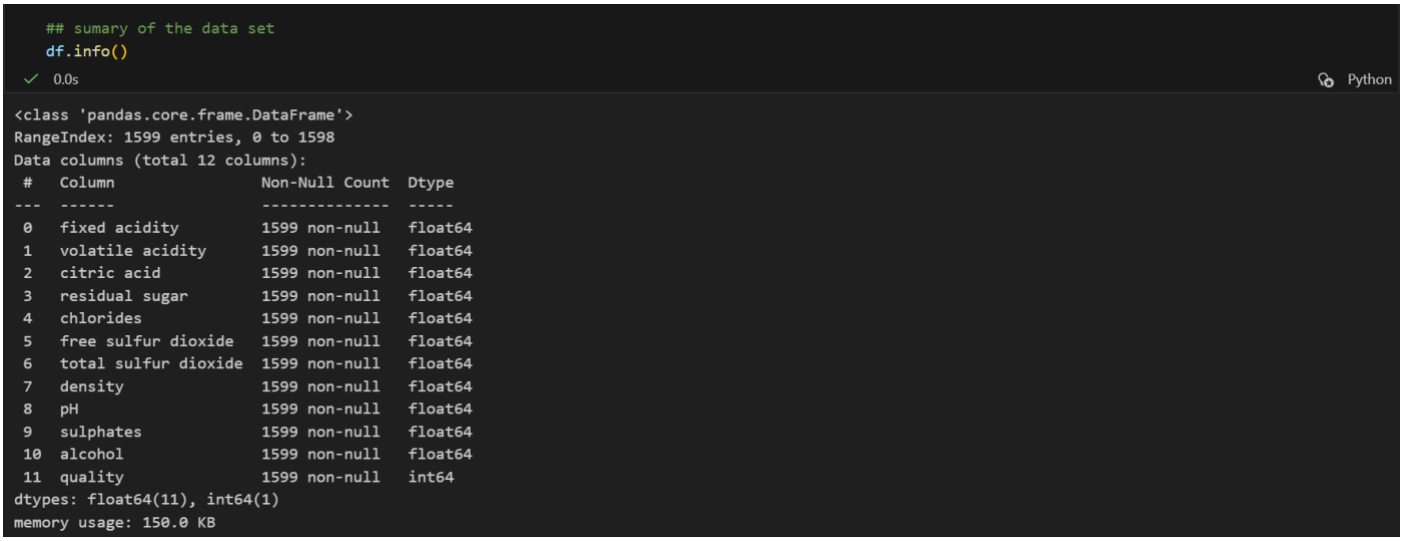
0.3s

```
df.head()
```

0.0s

	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

Step 3: I had been checking the data types of variables. Before making any changes.



```
## sumary of the data set
df.info()
```

0.0s Python

```
<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
```

Step 4: checking the descriptive analysis of stats in the data.

```
## descreptive summary  of the dataset
df.describe()
```

0.0s Python

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

Step 5: checking the records and columns

```
df.shape
```

✓ 0.0s

```
(1599, 12)
```

Step 7: list down the all columns and the check the Quality column Unique values.

```
## list Down All the columns names
df.columns
```

✓ 0.0s

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

```
df['quality'].unique()
```

✓ 0.0s

```
array([5, 6, 7, 4, 8, 3])
```

Step 8: I did check the null values but there is no null values in the data set

```
## missing values inthe data set
df.isnull().sum()
```

✓ 0.0s

```
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
```

Step 9: I had checked the duplicates of data set this data had 240 records duplicates .

```
## Duplicate records
df[df.duplicated()]
```

✓ 0.0s

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
4	7.4	0.700	0.00	1.90	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
11	7.5	0.500	0.36	6.10	0.071	17.0	102.0	0.99780	3.35	0.80	10.5	5
27	7.9	0.430	0.21	1.60	0.106	10.0	37.0	0.99660	3.17	0.91	9.5	5
40	7.3	0.450	0.36	5.90	0.074	12.0	87.0	0.99780	3.33	0.83	10.5	5
65	7.2	0.725	0.05	4.65	0.086	4.0	11.0	0.99620	3.41	0.39	10.9	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1563	7.2	0.695	0.13	2.00	0.076	12.0	20.0	0.99546	3.29	0.54	10.1	5
1564	7.2	0.695	0.13	2.00	0.076	12.0	20.0	0.99546	3.29	0.54	10.1	5
1567	7.2	0.695	0.13	2.00	0.076	12.0	20.0	0.99546	3.29	0.54	10.1	5
1581	6.2	0.560	0.09	1.70	0.053	24.0	32.0	0.99402	3.54	0.60	11.3	5
1596	6.3	0.510	0.13	2.30	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6

240 rows × 12 columns

Step 10: I had decided to remove duplicates

```
## remove duplicates
df.drop_duplicates(inplace=True)
```

✓ 0.0s

```
df.shape
```

✓ 0.0s

```
(1359, 12)
```

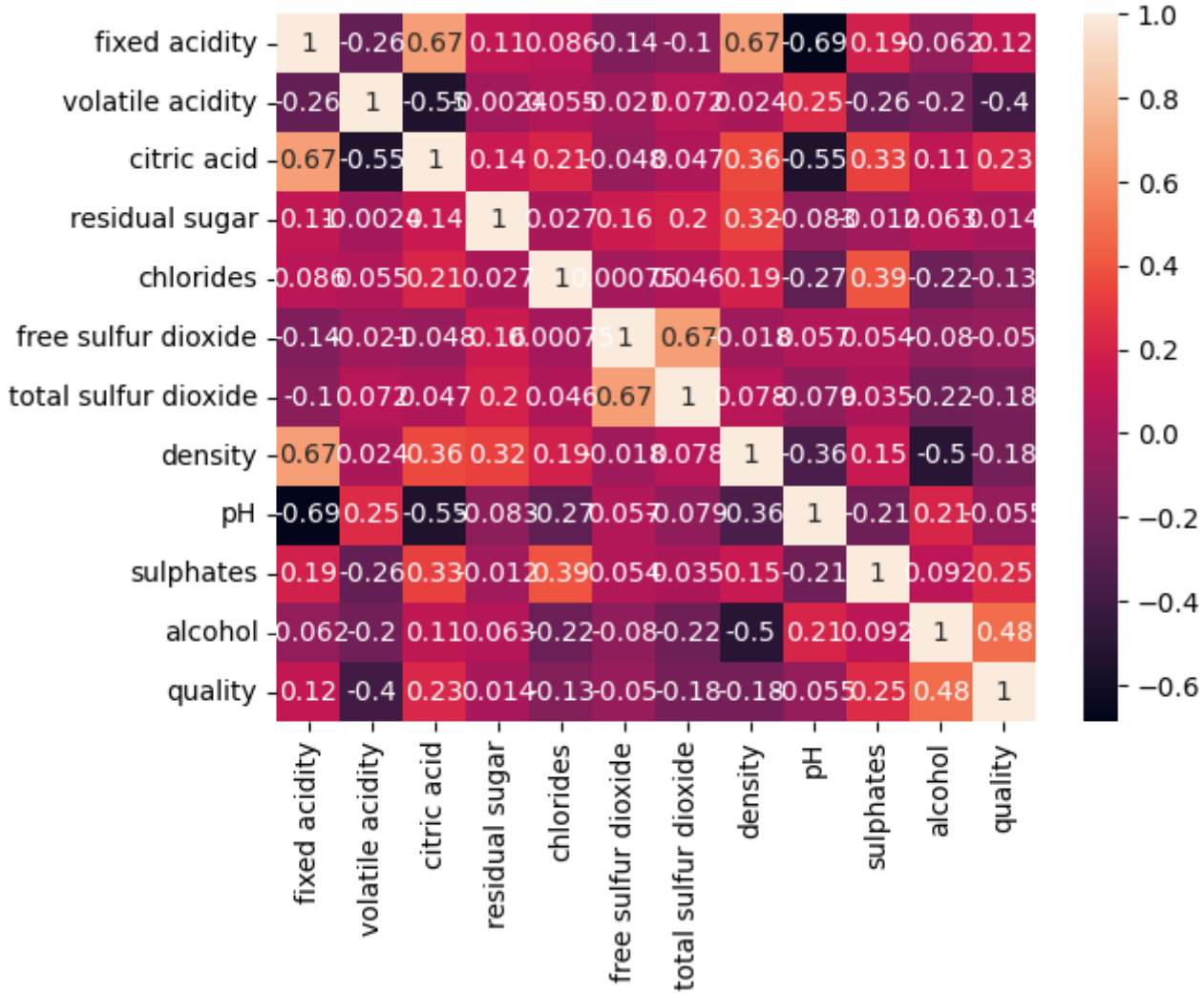
Step 11: I had to check the linear relationships of the data set using correlation function.

```
df.corr()
```
✓ 0.0s

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1.000000	-0.255124	0.667437	0.111025	0.085886	-0.140580	-0.103777	0.670195	-0.686685	0.190269	-0.061596	0.119024
volatile acidity	-0.255124	1.000000	-0.551248	-0.002449	0.055154	-0.020945	0.071701	0.023943	0.247111	-0.256948	-0.197812	-0.395214
citric acid	0.667437	-0.551248	1.000000	0.143892	0.210195	-0.048004	0.047358	0.357962	-0.550310	0.326062	0.105108	0.228057
residual sugar	0.111025	-0.002449	0.143892	1.000000	0.026656	0.160527	0.201038	0.324522	-0.083143	-0.011837	0.063281	0.013640
chlorides	0.085886	0.055154	0.210195	0.026656	1.000000	0.000749	0.045773	0.193592	-0.270893	0.394557	-0.223824	-0.130988
free sulfur dioxide	-0.140580	-0.020945	-0.048004	0.160527	0.000749	1.000000	0.667246	-0.018071	0.056631	0.054126	-0.080125	-0.050463
total sulfur dioxide	-0.103777	0.071701	0.047358	0.201038	0.045773	0.667246	1.000000	0.078141	-0.079257	0.035291	-0.217829	-0.177855
density	0.670195	0.023943	0.357962	0.324522	0.193592	-0.018071	0.078141	1.000000	-0.355617	0.146036	-0.504995	-0.184252
pH	-0.686685	0.247111	-0.550310	-0.083143	-0.270893	0.056631	-0.079257	-0.355617	1.000000	-0.214134	0.213418	-0.055245
sulphates	0.190269	-0.256948	0.326062	-0.011837	0.394557	0.054126	0.035291	0.146036	-0.214134	1.000000	0.091621	0.248835
alcohol	-0.061596	-0.197812	0.105108	0.063281	-0.223824	-0.080125	-0.217829	-0.504995	0.213418	0.091621	1.000000	0.480343
quality	0.119024	-0.395214	0.228057	0.013640	-0.130988	-0.050463	-0.177855	-0.184252	-0.055245	0.248835	0.480343	1.000000

Step 12: it had difficult to understand the data correlaton data so I had been using seaborn and matplotlib libraries to visualise the data.

```
plt.Figure(figsize=(10,6))
sns.heatmap(df.corr(),annot=True)
```
✓ 0.3s

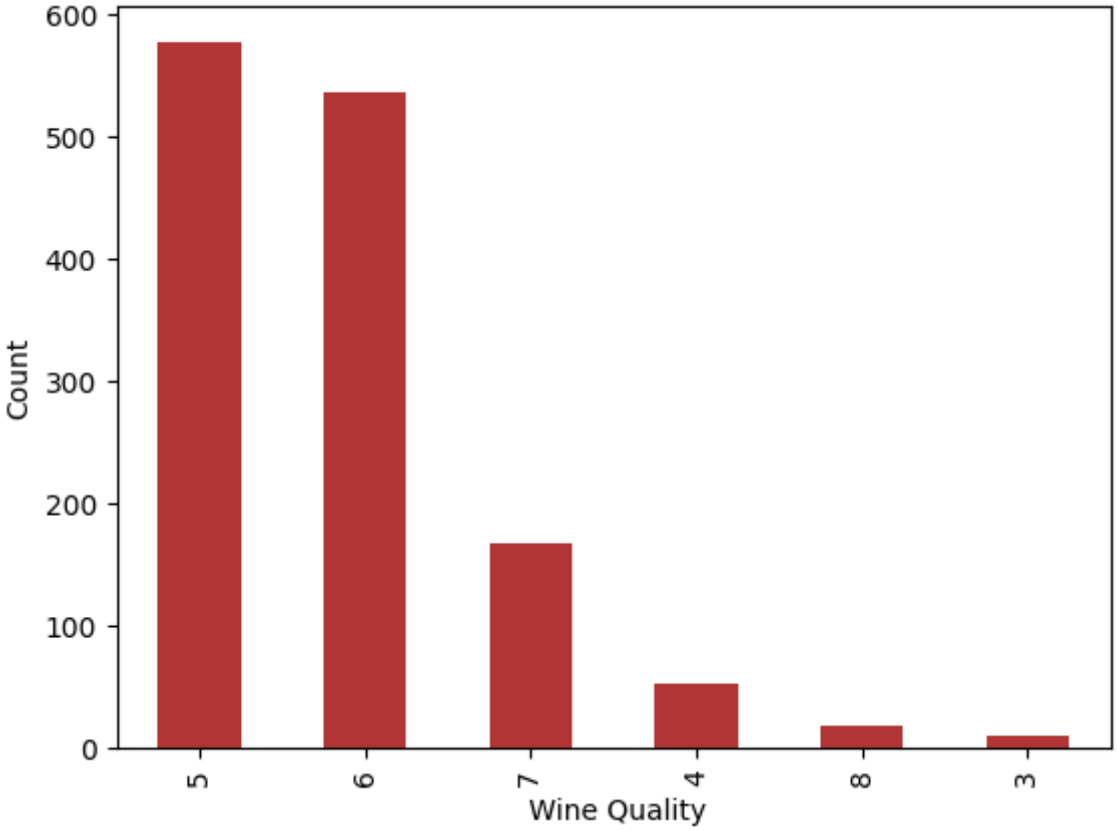


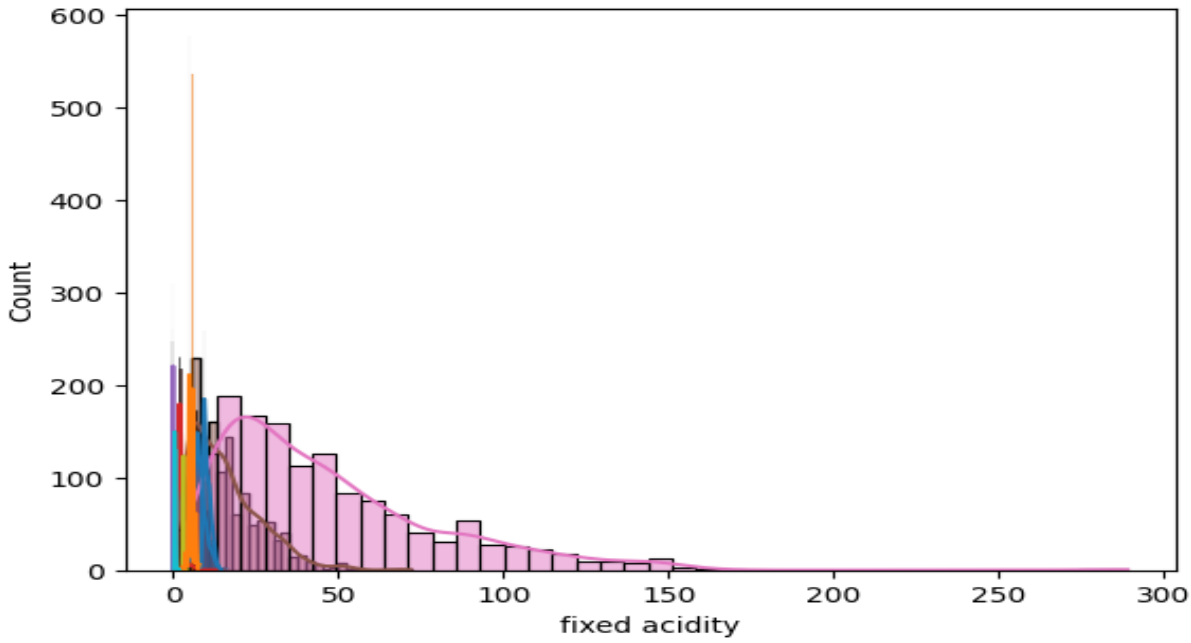
Now data is easy to understanding the relationships.

Step 13: I check the data distribution on quality column

```
## Visualization
## Qualityy counts
## conclusion it is a imbalance data set
df['quality'].value_counts().plot(kind='bar',color="#b13535")
plt.xlabel(" Wine Quality")
plt.ylabel('Count')
plt.show()
```

✓ 0.0s

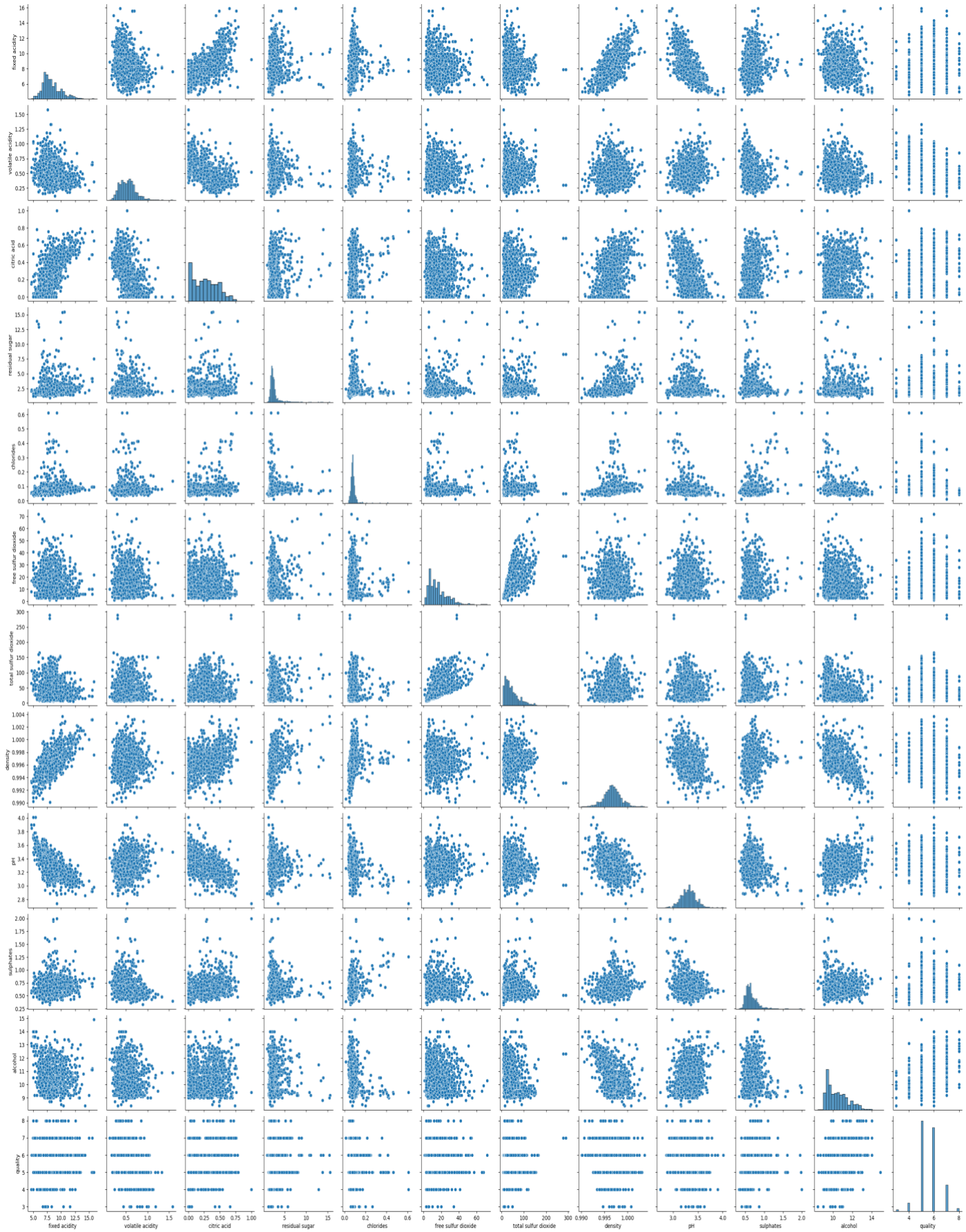




Step 14: univariate, bivariate, multivariate analysis

```
# univariate,bivariate,multivariate analysis
sns.pairplot(df)
```

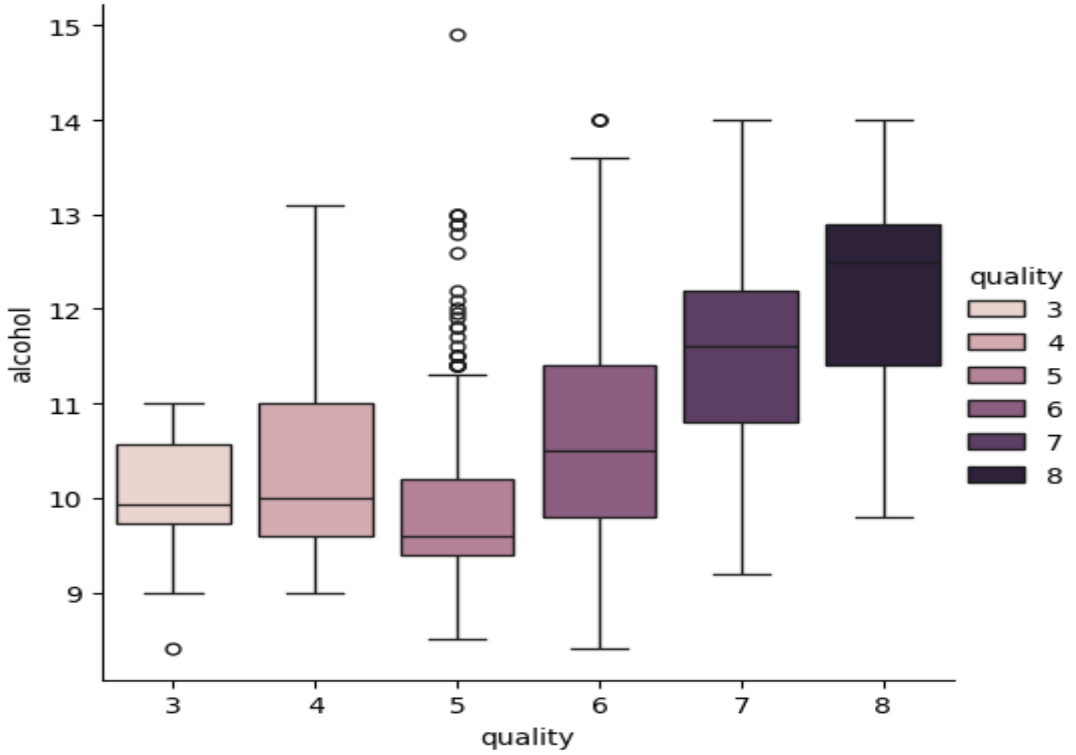
✓ 13.6s



Step 15: check the outliers on Quality to Alcohol

```
## categorical plot
sns.catplot(x='quality',y='alcohol',data=df,kind='box',hue='quality')
```

✓ 0.2s



Step 16: check the correlation relationship on Alcohol and pH.

```
sns.scatterplot(x='alcohol',data=df,y='pH',hue='quality')
```

✓ 0.2s