```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#simple imputer
array = np.array([[1,2],[np.nan,3],[7,6]])
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan,strategy="mean")
imputed_array = imputer.fit_transform(array)
print(imputed_array)
```

```
[[1. 2.]
 [4. 3.]
 [7. 6.]]
```

```python
#minmaxscaler
from sklearn.preprocessing import MinMaxScaler
array_2 = np.array([[1,2],[6,7],[3,4]])
scaler = MinMaxScaler()
scaled_array = scaler.fit_transform(array_2)
print(scaled_array)
```

```
[[0.  0. ]
 [1.  1. ]
 [0.4 0.4]]
```

```python
data = pd.read_csv("iris.csv")
df = pd.DataFrame(data)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   sepallength  150 non-null    float64
 1   sepalwidth   150 non-null    float64
 2   petallength  150 non-null    float64
 3   petalwidth   150 non-null    float64
 4   class        150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```python
df_nonull = df[df.isnull()==False]
df_nonull
```

```
     sepallength  sepalwidth  petallength  petalwidth           class
0            5.1         3.5          1.4         0.2     Iris-setosa
1            4.9         3.0          1.4         0.2     Iris-setosa
2            4.7         3.2          1.3         0.2     Iris-setosa
3            4.6         3.1          1.5         0.2     Iris-setosa
```

```
4            5.0         3.6          1.4         0.2     Iris-setosa
..           ...         ...          ...         ...             ...
145          6.7         3.0          5.2         2.3  Iris-virginica
146          6.3         2.5          5.0         1.9  Iris-virginica
147          6.5         3.0          5.2         2.0  Iris-virginica
148          6.2         3.4          5.4         2.3  Iris-virginica
149          5.9         3.0          5.1         1.8  Iris-virginica

[150 rows x 5 columns]
```

```
df_nonull.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   sepallength  150 non-null    float64
 1   sepalwidth   150 non-null    float64
 2   petallength  150 non-null    float64
 3   petalwidth   150 non-null    float64
 4   class        150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
df.head(5)
```

```
   sepallength  sepalwidth  petallength  petalwidth        class
0          5.1         3.5          1.4         0.2  Iris-setosa
1          4.9         3.0          1.4         0.2  Iris-setosa
2          4.7         3.2          1.3         0.2  Iris-setosa
3          4.6         3.1          1.5         0.2  Iris-setosa
4          5.0         3.6          1.4         0.2  Iris-setosa
```

```
df.tail(5)
```

```
     sepallength  sepalwidth  petallength  petalwidth           class
145          6.7         3.0          5.2         2.3  Iris-virginica
146          6.3         2.5          5.0         1.9  Iris-virginica
147          6.5         3.0          5.2         2.0  Iris-virginica
148          6.2         3.4          5.4         2.3  Iris-virginica
149          5.9         3.0          5.1         1.8  Iris-virginica
```

```
df["petallength"].dtype
```

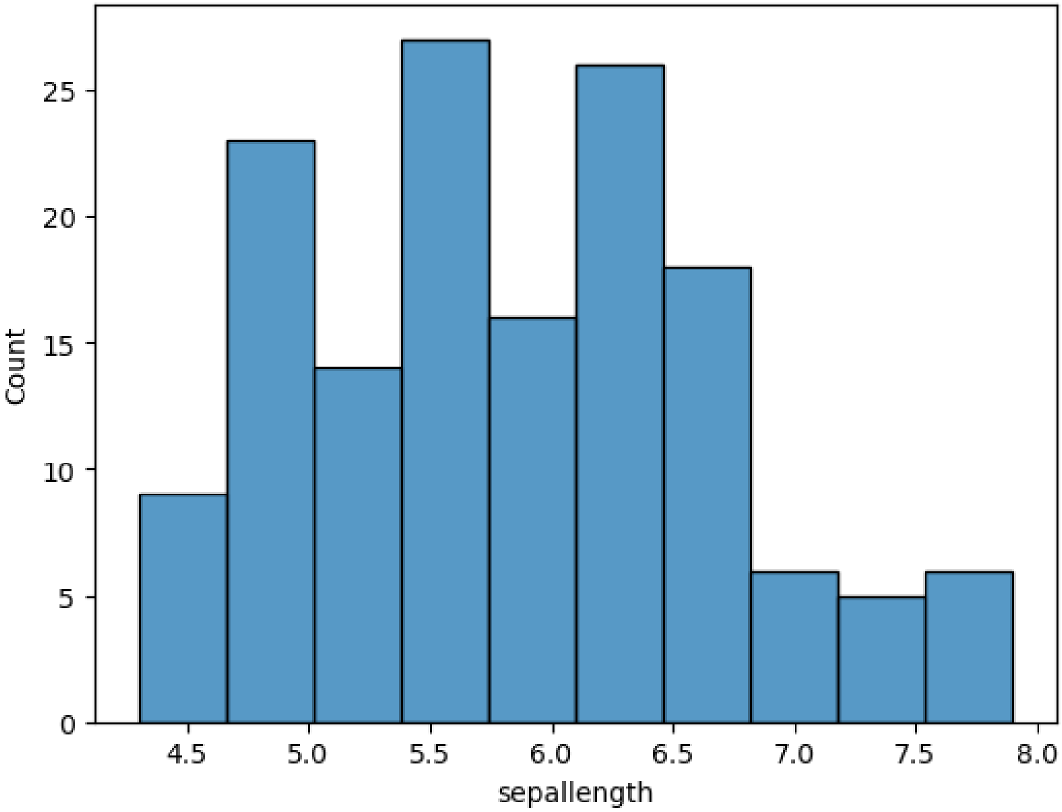
```
dtype('float64')
```

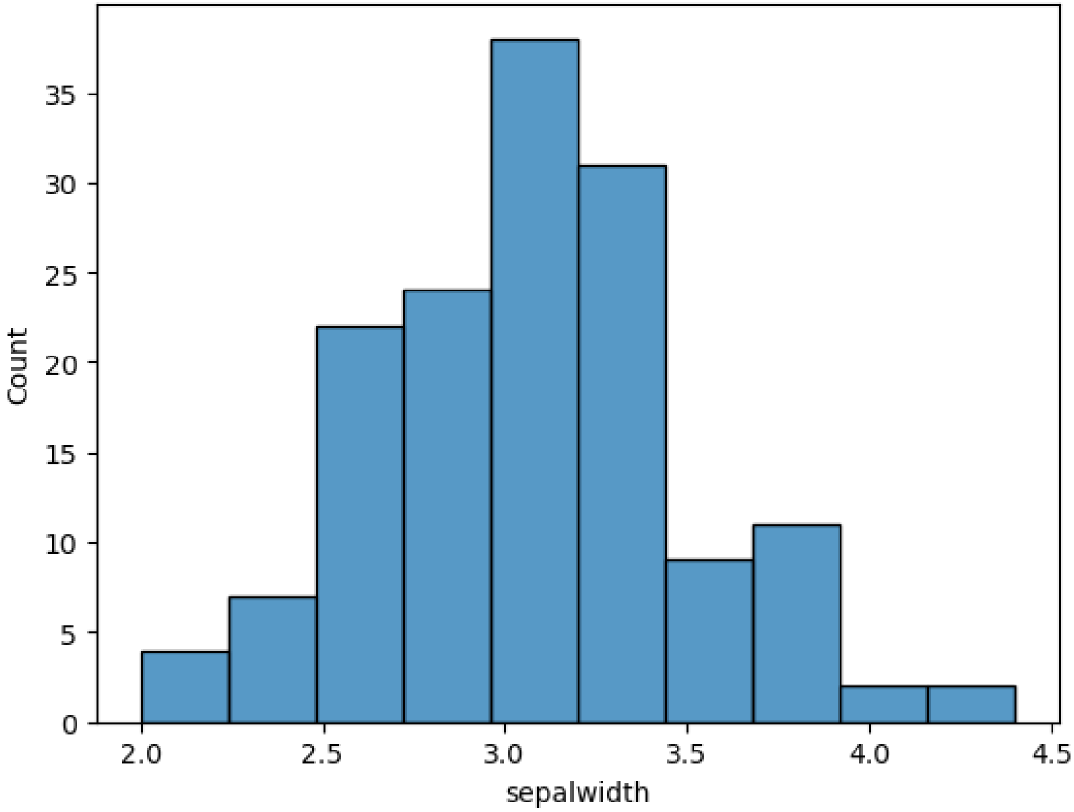
```
df["sepallength"].mean()
```

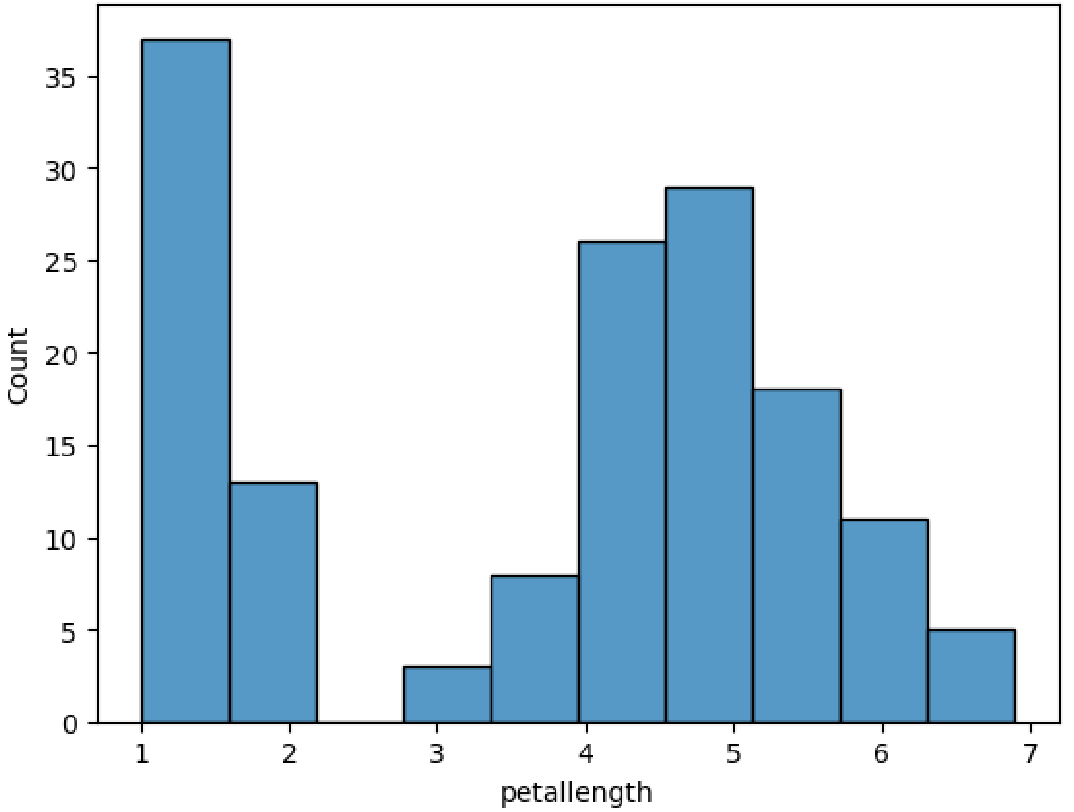
```
5.843333333333334
```

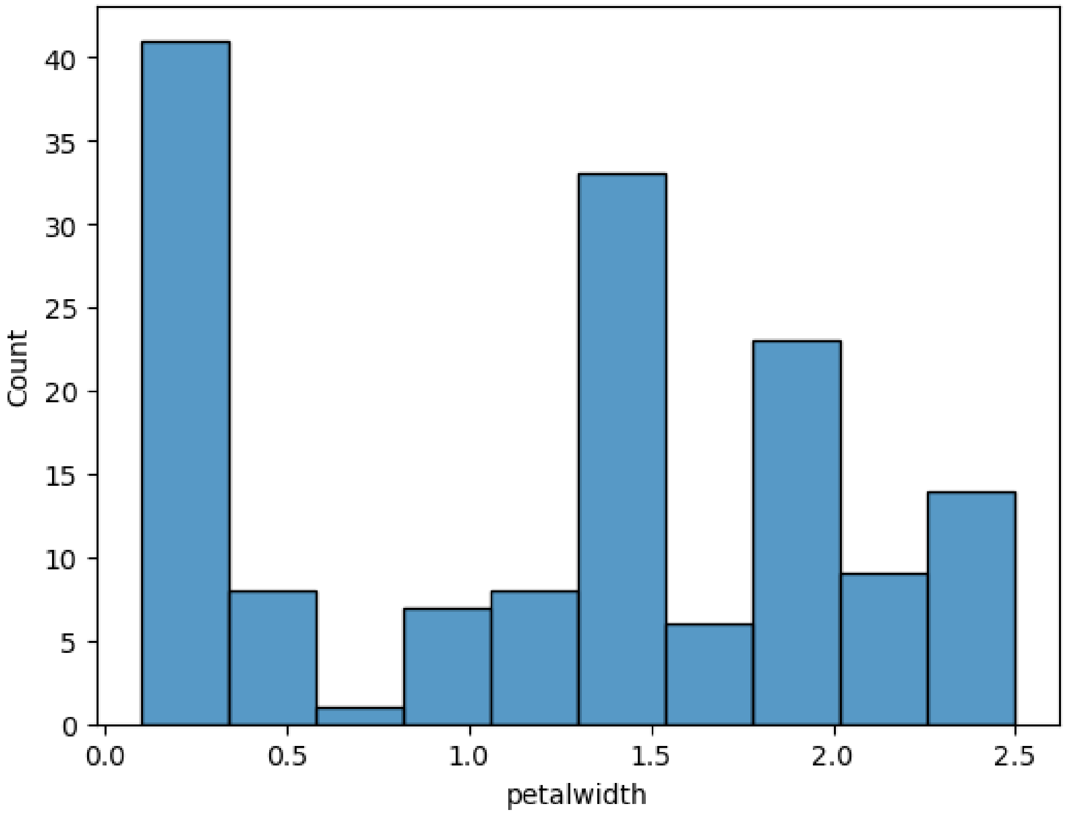
```
df["petalwidth"].std()
```

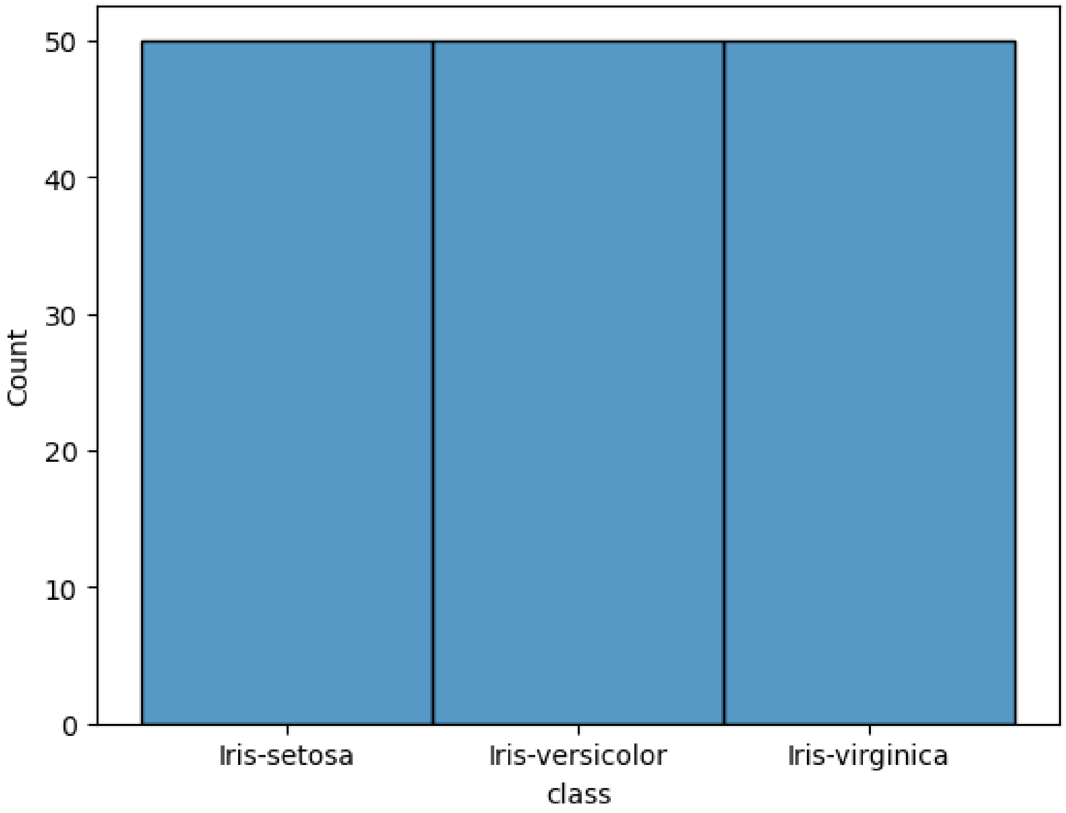
```
0.7631607417008414

import seaborn as sns

for i in list(df.columns):
    sns.histplot(df[i],bins=10)
    plt.show()
```




Count
0
5
10
15
20
25
30
35
2.0
2.5
3.0
3.5
4.0
4.5
sepalwidth

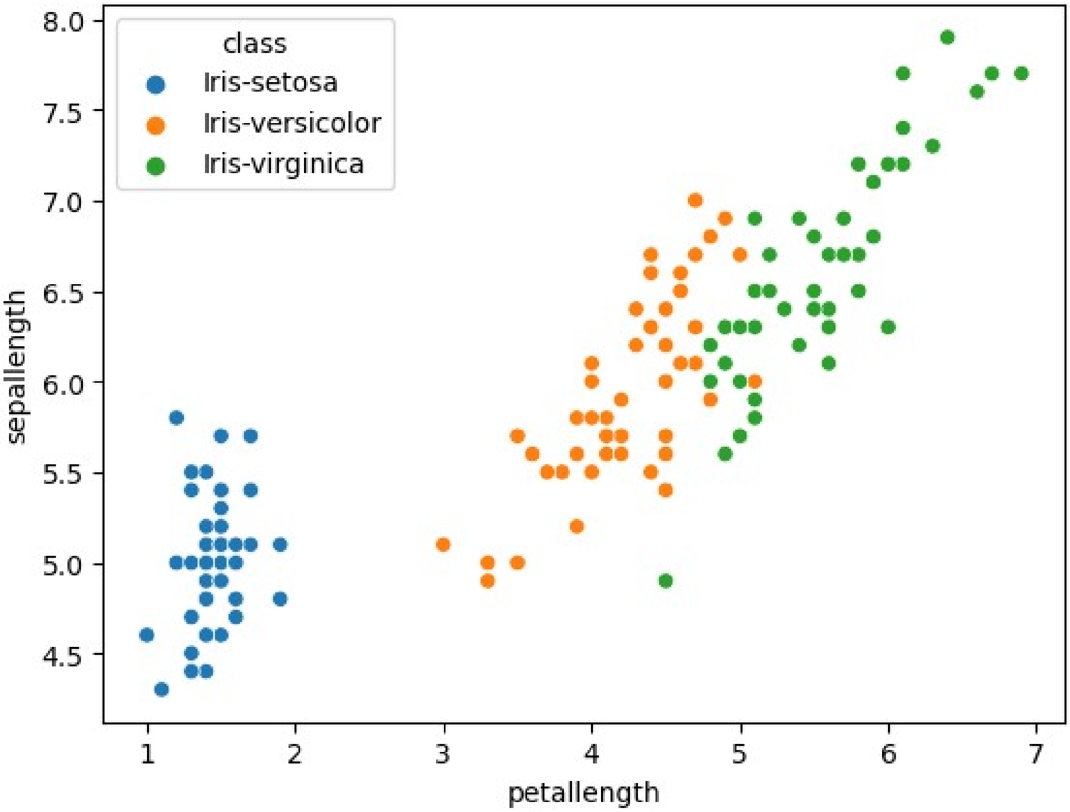

Count
0
5
10
15
20
25
30
35
1
2
3
4
5
6
7
petallength


Count
0
5
10
15
20
25
30
35
40
0.0
0.5
1.0
1.5
2.0
2.5
petalwidth


Count
0
10
20
30
40
50
Iris-setosa
Iris-versicolor
Iris-virginica
class

```
sns.scatterplot(data=df,x=df["petallength"],y=df["sepallength"],hue=df
["class"])
```

```
<Axes: xlabel='petallength', ylabel='sepallength'>
```



```
wine = pd.read_csv("WineQT.csv")
df_wine  = pd.DataFrame(wine)
df_wine.head()
```

```
   fixed_acidity  volatile_acidity  citric_acid  residual_sugar
chlorides  \
0            7.4              0.70         0.00             1.9
0.076
1            7.8              0.88         0.00             2.6
0.098
2            7.8              0.76         0.04             2.3
0.092
3           11.2              0.28         0.56             1.9
0.075
4            7.4              0.70         0.00             1.9
0.076

   free_sulfur_dioxide  total_sulfur_dioxide  density    pH  sulphates
\
0                 11.0                  34.0   0.9978  3.51       0.56

1                 25.0                  67.0   0.9968  3.20       0.68

2                 15.0                  54.0   0.9970  3.26       0.65

3                 17.0                  60.0   0.9980  3.16       0.58

4                 11.0                  34.0   0.9978  3.51       0.56
```

```
   alcohol  quality  Id
0      9.4        5   0
1      9.8        5   1
2      9.8        5   2
3      9.8        6   3
4      9.4        5   4
```

```
df_wine.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
```

```
df_wine.shape
```

```
(1143, 13)
```

```
df_wine.isnull().sum()
```

```
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
```

```python
for j in list(df_wine.columns):
    print("1st quartile:",np.percentile(df_wine[j],25))
    print("2nd quartile:",np.percentile(df_wine[j],75))
    print()
```

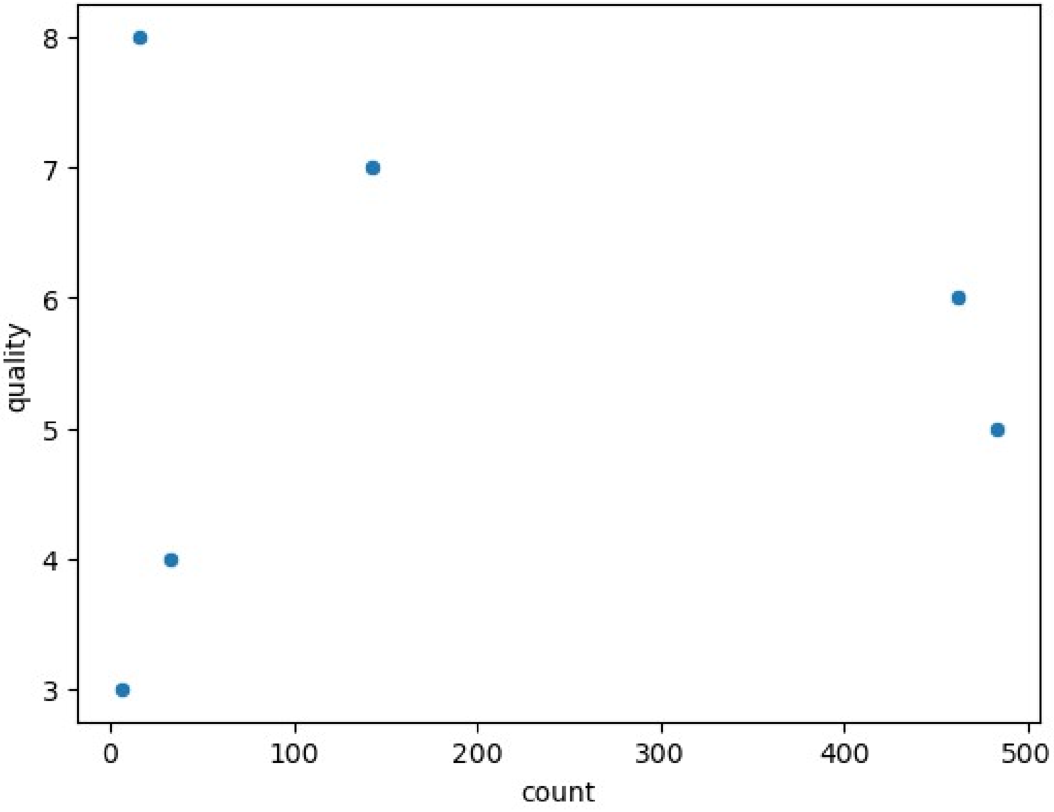
```
1st quartile: 7.1
2nd quartile: 9.1

1st quartile: 0.3925
2nd quartile: 0.64

1st quartile: 0.09
2nd quartile: 0.42

1st quartile: 1.9
2nd quartile: 2.6

1st quartile: 0.07
2nd quartile: 0.09

1st quartile: 7.0
2nd quartile: 21.0

1st quartile: 21.0
2nd quartile: 61.0

1st quartile: 0.99557
2nd quartile: 0.997845

1st quartile: 3.205
2nd quartile: 3.4

1st quartile: 0.55
2nd quartile: 0.73

1st quartile: 9.5
2nd quartile: 11.1

1st quartile: 5.0
2nd quartile: 6.0
```

```
1st quartile: 411.0
2nd quartile: 1209.5
```

```
df_plot = df_wine["quality"].value_counts().sort_index().reset_index()
df_plot.columns = ["quality","count"]
df_plot
```

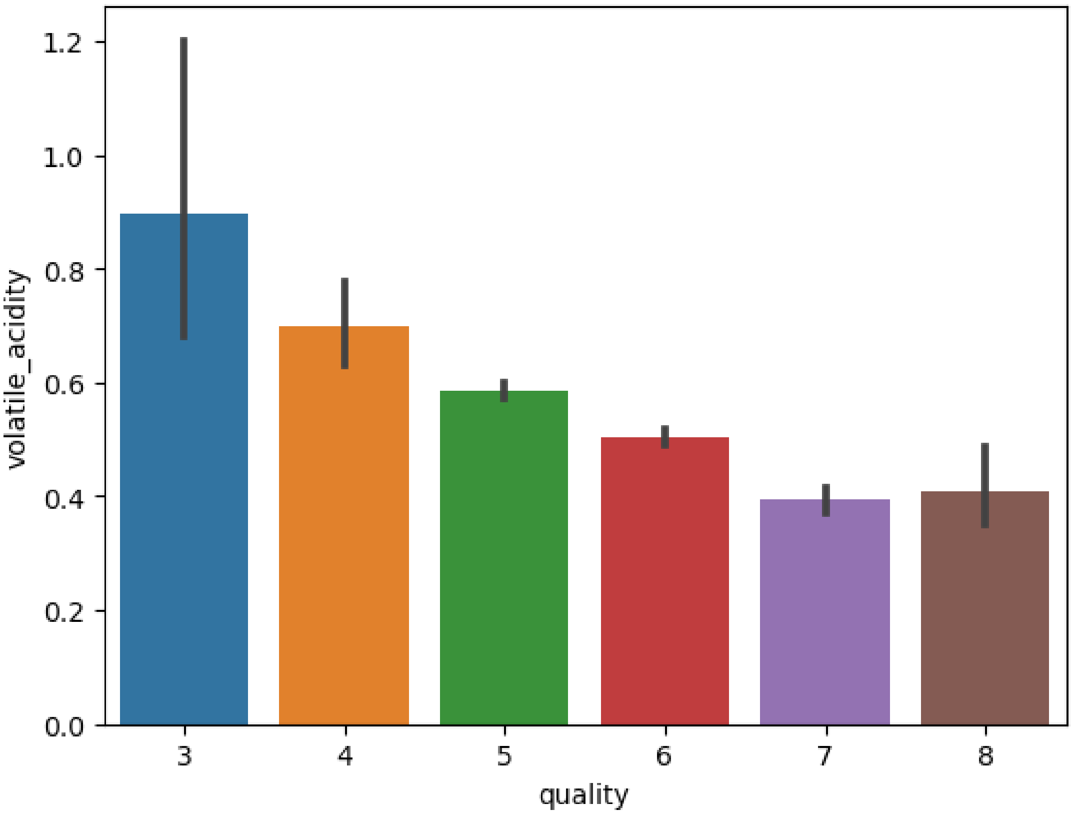
```
   quality  count
0        3      6
1        4     33
2        5    483
3        6    462
4        7    143
5        8     16
```

```
sns.scatterplot(data =
df_plot,x=df_plot["count"],y=df_plot["quality"])
plt.show()
```



```
sns.barplot(data=df_wine,x=df_wine["quality"],y=df_wine["volatile_acid
ity"])
```

```
<Axes: xlabel='quality', ylabel='volatile_acidity'>
```


1.2
1.0
0.8
0.6
0.4
0.2
0.0
volatile_acidity
3
4
5
6
7
8
quality