Import Libraries

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

```
data=pd.read_csv("redwine.csv")
data.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH
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
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51

```
data.shape
```

```
(1599, 12)
```

```
data.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
dtype: int64
```

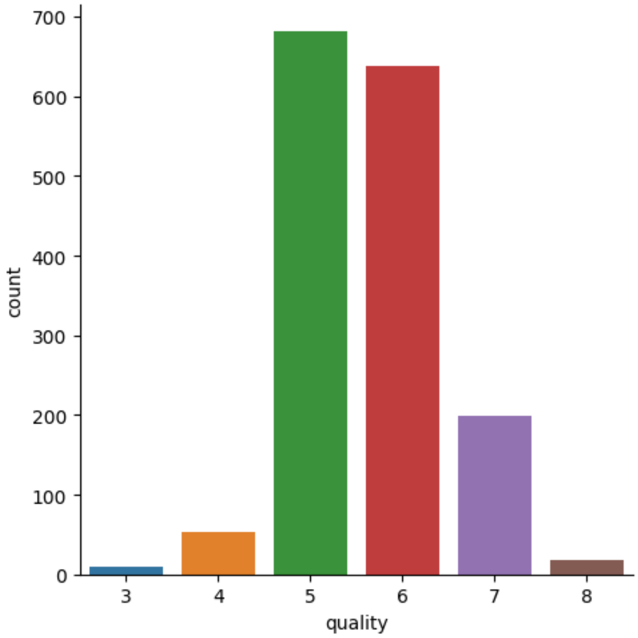
Data Analysis & Data Visualization

```
data.describe()
```

	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
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

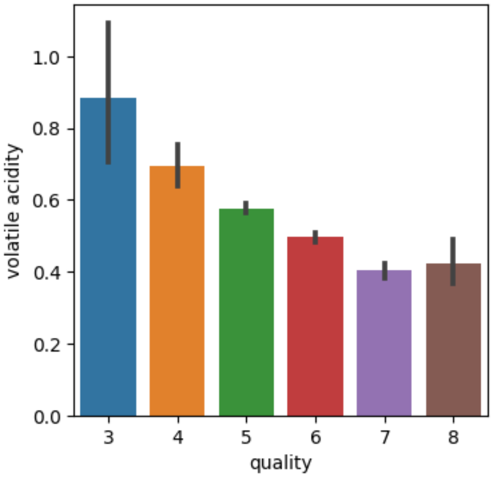
```
sns.catplot(x='quality',data=data,kind='count')
```

```
<seaborn.axisgrid.FacetGrid at 0x7a41e1e258d0>
```



```
#volatile acidity vs Quality
plot=plt.figure(figsize=(4,4))
sns.barplot(x='quality',y='volatile acidity',data=data)
```

```
<Axes: xlabel='quality', ylabel='volatile acidity'>
```



```
#citric acid vs quality
sns.barplot(x='quality', y='citric acid',data=data)
```

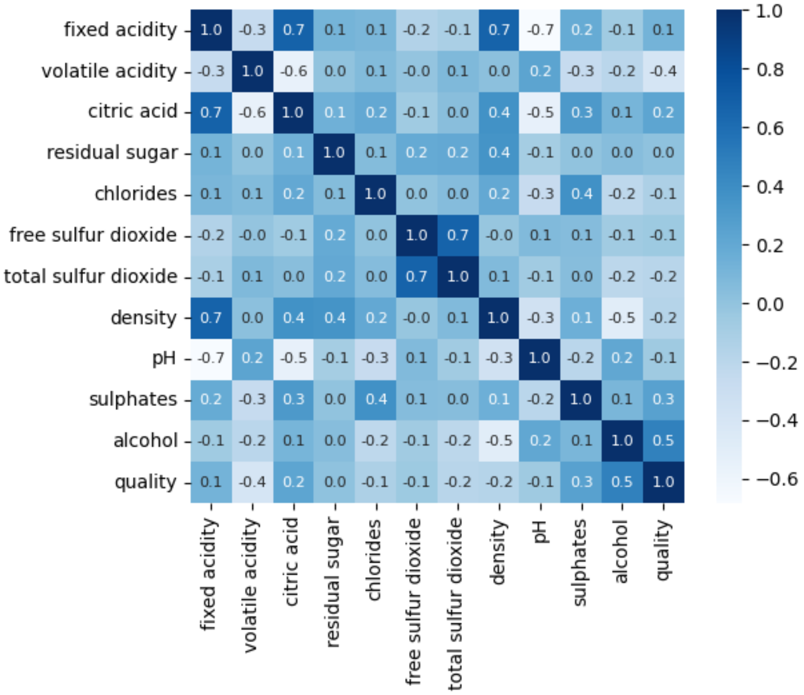
```
<Axes: xlabel='quality', ylabel='citric acid'>
```

Correlations 1.Positive and Negative Correlations

```
correlations=data.corr()
```

```
sns.heatmap(correlations,cbar=True,square=True,fmt='.1f',annot=True,annot_kws={'size':8},cmap='Blues')
```

```
<Axes: >
```



Data Preprocessing

```
X = data.drop(['quality'], axis=1)
```

```
Y=data['quality'].apply(lambda value:1 if value>=7 else 0)
print(Y)
```

```
0       0
1       0
2       0
3       0
4       0
       ..
1594    0
1595    0
1596    0
1597    0
1598    0
Name: quality, Length: 1599, dtype: int64
```

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=3)
print(Y.shape,Y_train.shape,Y_test.shape)
```

```
(1599,) (1279,) (320,)
```

Model Training

```
model=RandomForestClassifier()
model.fit(X_train,Y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

Model Evaluation

```
prediction=model.predict(X_test)
test_accuracy=accuracy_score(prediction,Y_test)
```

```
print("Accuracy is: ",test_accuracy)
```

```
Accuracy is:  0.93125
```

Building a Predictive System

```
input_data=(7.5,0.5,0.36,6.1,0.071,17.0,102.0,0.9978,3.35,0.8,10.5)
input_data_num=np.asarray(input_data)
input_data_reshape=input_data_num.reshape(1,-1)
predict_result=model.predict(input_data_reshape)
print(predict_result)
if (predict_result[0]==1):
  print('Good Quality Wine')
else:
    print('Bad Quality Wine')
```

```
[0]
Bad Quality Wine
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but RandomForestClassifie
  warnings.warn(
```

✓ 0s completed at 7:54 AM