In [1]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

In [2]:
```python
wine_dataset = pd.read_csv('WineQT.csv')
wine_dataset.head()
```

Out[2]:

	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
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	4

In [3]:
```python
### number of rows and columns
wine_dataset.shape
```

Out[3]:
```
(1143, 13)
```

In [4]:
```python
##### checking for missing values
wine_dataset.isnull().sum()
```

Out[4]:
```
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
```

In [5]:
```python
##### Data Analysis and Visualisation
wine_dataset.describe()
```

Out[5]:

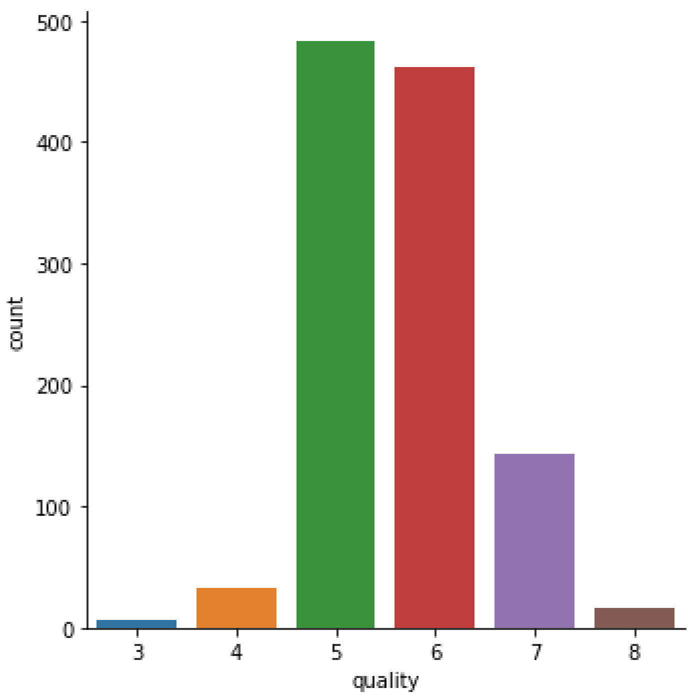
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density
count	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000
mean	8.311111	0.531339	0.268364	2.532152	0.086933	15.615486	45.914698	0.996730
std	1.747595	0.179633	0.196686	1.355917	0.047267	10.250486	32.782130	0.001925

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070
25%	7.100000	0.392500	0.090000	1.900000	0.070000	7.000000	21.000000	0.995570
50%	7.900000	0.520000	0.250000	2.200000	0.079000	13.000000	37.000000	0.996680
75%	9.100000	0.640000	0.420000	2.600000	0.090000	21.000000	61.000000	0.997845
max	15.900000	1.580000	1.000000	15.500000	0.611000	68.000000	289.000000	1.003690

In [6]:

```
#### number of values for each quality
sns.catplot(x='quality',data = wine_dataset, kind = 'count')
```

Out[6]: <seaborn.axisgrid.FacetGrid at 0x1f5ff2d93a0>



In [7]:

```
wine_dataset['quality'].value_counts()
```

Out[7]:

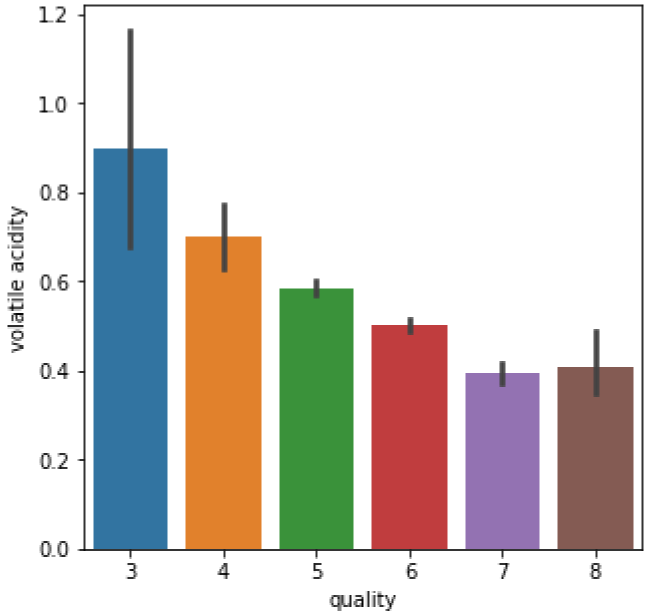
```
5    483
6    462
7    143
4     33
8     16
3      6
Name: quality, dtype: int64
```

In [8]:

```
##### volatile acidity vs Quality

plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality',y='volatile acidity',data = wine_dataset)
```

Out[8]: <AxesSubplot:xlabel='quality', ylabel='volatile acidity'>



In [9]:

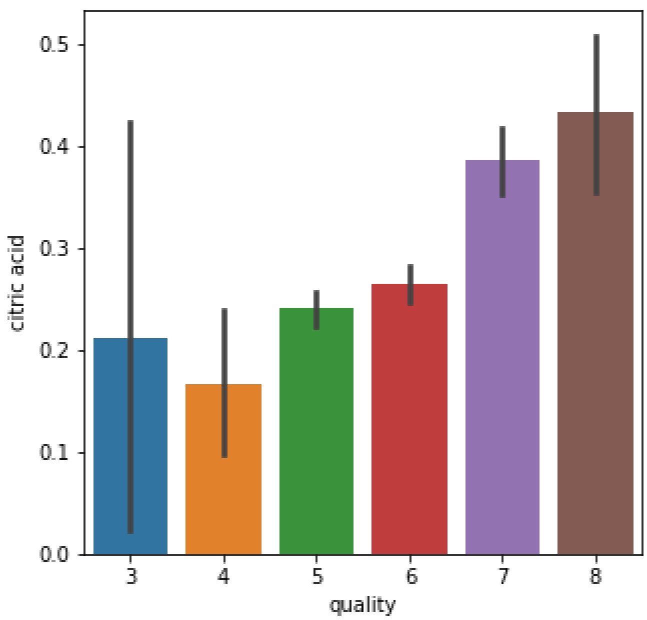
```
## Observsation
#1. If the valatile acidity is high then wine quality is low viceversa.
```

In [10]:

```
##### citric acid vs Quality

plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality',y='citric acid',data = wine_dataset)
```

Out[10]: <AxesSubplot:xlabel='quality', ylabel='citric acid'>



In [11]:

```
#### Observation
#1. If the citric acid quantitiy  is high then the wine quality also high and viceversa.
```

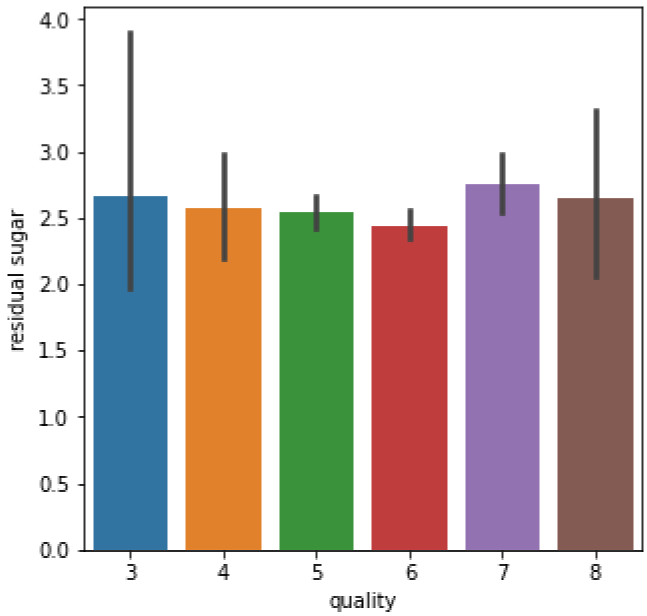
In [12]:

```
##### residual sugar vs Quality

plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality',y='residual sugar',data = wine_dataset)
```

<AxesSubplot:xlabel='quality', ylabel='residual sugar'>

Loading [MathJax]/extensions/Safe.js



In [13]:

```
wine_dataset = wine_dataset.drop('Id',axis = 1)
wine_dataset.head()
```

Out[13]:

	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

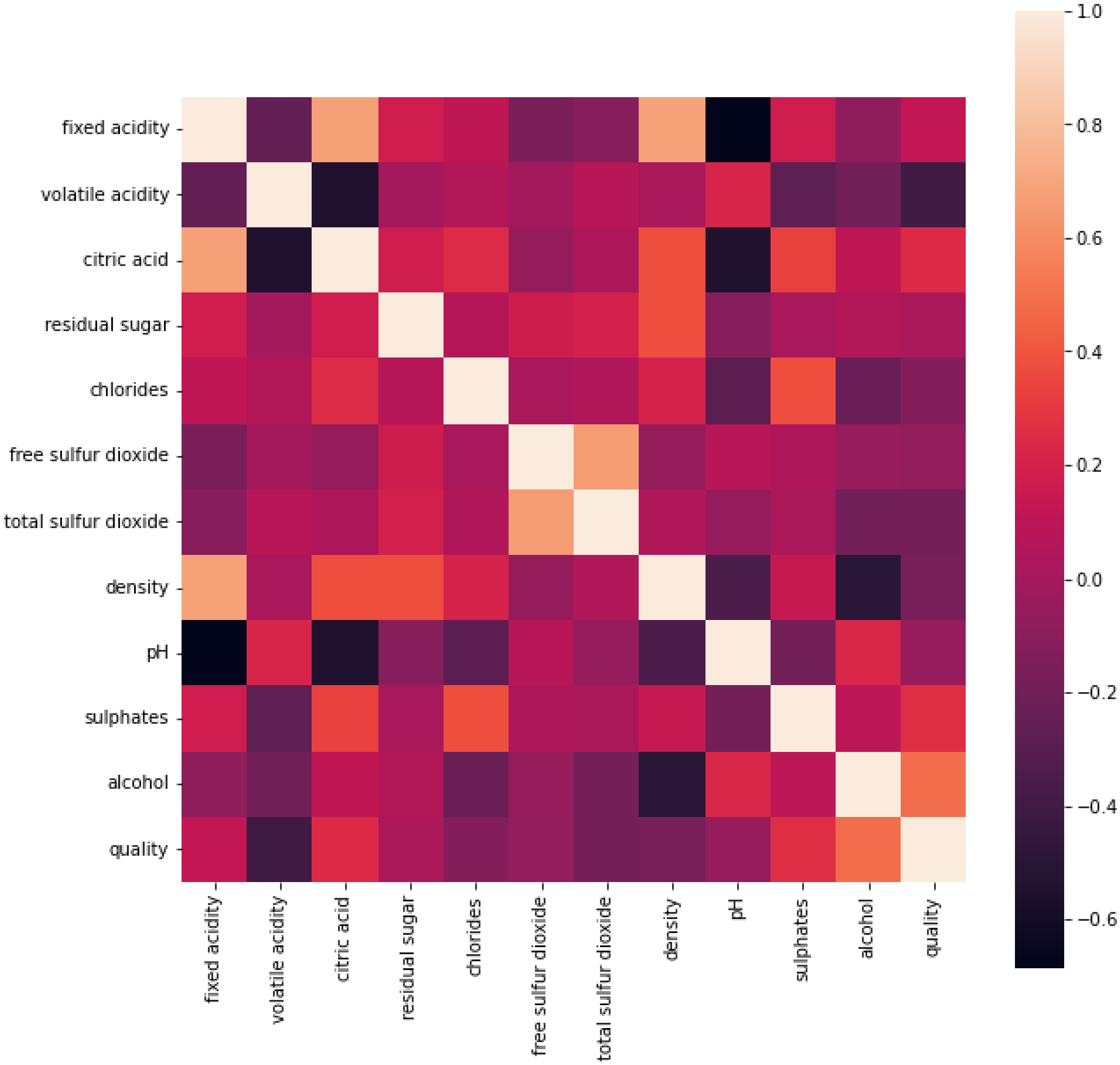
In [14]:

```
##### Correlation
correlation = wine_dataset.corr()
```

In [15]:

```
##### Constructing the heatmap to understand the correlation between the columns
plt.figure(figsize=(10,10))
sns.heatmap(correlation, cbar=True ,square=True)
```

Out[15]: <AxesSubplot:>



In [ ]:

In [16]:

```
wine_dataset.head()
```

Out[16]:

	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

In [17]:

```
#### Observation
#1. By these we can see how these variables are correlated with each other.
```

In [18]:

```
##### Data Preprocessing
```

```
### separating data and labels
X = wine_dataset.drop('quality',axis=1)
```

In [19]:
```
#### Label Binarisation
Y = wine_dataset['quality'].apply(lambda y_value: 1 if y_value>=7 else 0)
```

In [20]:
```
print(Y)
```

```
0       0
1       0
2       0
3       0
4       0
       ..
1138    0
1139    0
1140    0
1141    0
1142    0
Name: quality, Length: 1143, dtype: int64
```

In [21]:
```
#### Train and Test Split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify = Y, ra
```

In [22]:
```
print(X.shape,X_train.shape,X_test.shape)
```

```
(1143, 11) (914, 11) (229, 11)
```

In [23]:
```
print(Y.shape,Y_train.shape,Y_test.shape)
```

```
(1143,) (914,) (229,)
```

In [24]:
```
##### Model Training
```

In [25]:
```
#### Random Forest Classifier

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

Out[25]:
```
RandomForestClassifier()
```

In [26]:
```
#### Model Evaluation

#### Accuacy on training data
X_train_prediction = model.predict(X_train)
trainig_data_accu = accuracy_score(X_train_prediction, Y_train)
trainig_data_accu
```

Out[26]:
```
1.0
```

In [27]:
```
#### Accuacy on test data
X_test_prediction = model.predict(X_test)
test_data_accu = accuracy_score(X_test_prediction, Y_test)
test_data_accu
```

Out[27]: 0.8558951965065502

In [28]:

```
###### Building a Predictive System
```

In [29]:

```
input_data = (8.8,0.41,0.64,2.2,0.093000000000000001,9.0,42.0,0.9986,3.54,0.66,10.5)

#### Changing the input data to np array
input_data_as_np_array = np.asarray(input_data)

### Reshaping the data as we are predicting the label for only one instance
reshaped_input_data = input_data_as_np_array.reshape(1,-1)

prediction = model.predict(reshaped_input_data)
print(prediction)

if (prediction[0]==1):
    print('Good Quality Wine')
else:
    print('Bad Quality Wine')
```

```
[0]
Bad Quality Wine
```

In [ ]: