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

In [ ]:
```python
#the given data set is a set of parameters which determine the quality of the wine.
#the key incredients and how they affect the quality of wine
#here we are using the random forest classifier as the quality of wine is based on several parameters.
```

In [2]:
```python
data_set=pd.read_csv("winequality-red.csv")
```

In [3]:
```python
data_set.head()
```

Out[3]:

	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 [4]:
```python
data_set.describe()
```

Out[4]:

	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

In [5]:
```python
data_set.shape
```

Out[5]:
```
(1599, 12)
```

In [6]:
```python
data_set['quality'].value_counts()
```

Out[6]:
```
5    681
6    638
7    199
4     53
8     18
3     10
Name: quality, dtype: int64
```

In [ ]:
```python
#the quality 7 and ablove can be considered as good quality
```

In [8]:
```python
data_set.isnull().sum()
```

Out[8]:
```
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 [10]:
```python
x=data_set.drop(columns=['quality'],axis=1)
```

In [11]:
```python
print(x)
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0               7.4             0.700         0.00             1.9      0.076
1               7.8             0.880         0.00             2.6      0.098
2               7.8             0.760         0.04             2.3      0.092
3              11.2             0.280         0.56             1.9      0.075
4               7.4             0.700         0.00             1.9      0.076
...             ...               ...          ...             ...        ...
1594            6.2             0.600         0.08             2.0      0.090
1595            5.9             0.550         0.10             2.2      0.062
1596            6.3             0.510         0.13             2.3      0.076
1597            5.9             0.645         0.12             2.0      0.075
1598            6.0             0.310         0.47             3.6      0.067

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                    11.0                  34.0  0.99780  3.51       0.56
1                    25.0                  67.0  0.99680  3.20       0.68
2                    15.0                  54.0  0.99700  3.26       0.65
3                    17.0                  60.0  0.99800  3.16       0.58
4                    11.0                  34.0  0.99780  3.51       0.56
...                   ...                   ...      ...   ...        ...
1594                 32.0                  44.0  0.99490  3.45       0.58
1595                 39.0                  51.0  0.99512  3.52       0.76
1596                 29.0                  40.0  0.99574  3.42       0.75
1597                 32.0                  44.0  0.99547  3.57       0.71
1598                 18.0                  42.0  0.99549  3.39       0.66

      alcohol
0         9.4
1         9.8
2         9.8
3         9.8
4         9.4
...       ...
1594     10.5
1595     11.2
1596     11.0
1597     10.2
1598     11.0

[1599 rows x 11 columns]
```

In [15]:
```python
y=data_set['quality'].apply(lambda value: 1 if value>=7 else 0)
```

In [16]:
```python
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
```

In [17]:
```python
y.value_counts()
```

Out[17]:
```
0    1382
1     217
Name: quality, dtype: int64
```

In [18]:
```python
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=1,test_size=0.2,stratify=y)
```

In [19]:
```python
model=RandomForestClassifier()
```

In [20]:
```python
model.fit(x_train,y_train)
```

Out[20]:
```
RandomForestClassifier()
```

In [21]:
```python
train_prediction = model.predict(x_train)
train_accuraacy = accuracy_score(train_prediction,y_train)
```

In [22]:
```python
print(train_accuraacy)
```

```
1.0
```

In [23]:
```python
test_prediction = model.predict(x_test)
test_accuraacy = accuracy_score(test_prediction,y_test)
```

In [24]:
```python
print(test_accuraacy)
```

```
0.925
```

In [ ]:
```python
#we are checking the accuracy score of the test data and they came out fine. we are checking with new values from the
#data set and cross checking the vlaidity of the results
```

In [26]:
```python
values_to_predict = (12.8,0.3,0.74,2.6,0.095,9,28,0.9994,3.2,0.77,10.8)
value_array = np.asarray(values_to_predict)
value_reshaped=value_array.reshape(1,-1)
predict = model.predict(value_reshaped)
if (predict[0]==1):
    print('The quality is good')
else:
    print('The quality is bad')
```

```
The quality is good
C:\Users\jowin\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```

In [27]:
```python
data_set.describe()
```

Out[27]:

	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

In [ ]:
```python
# we analyse the data further for the relation among various parameters and how they affect our results
```

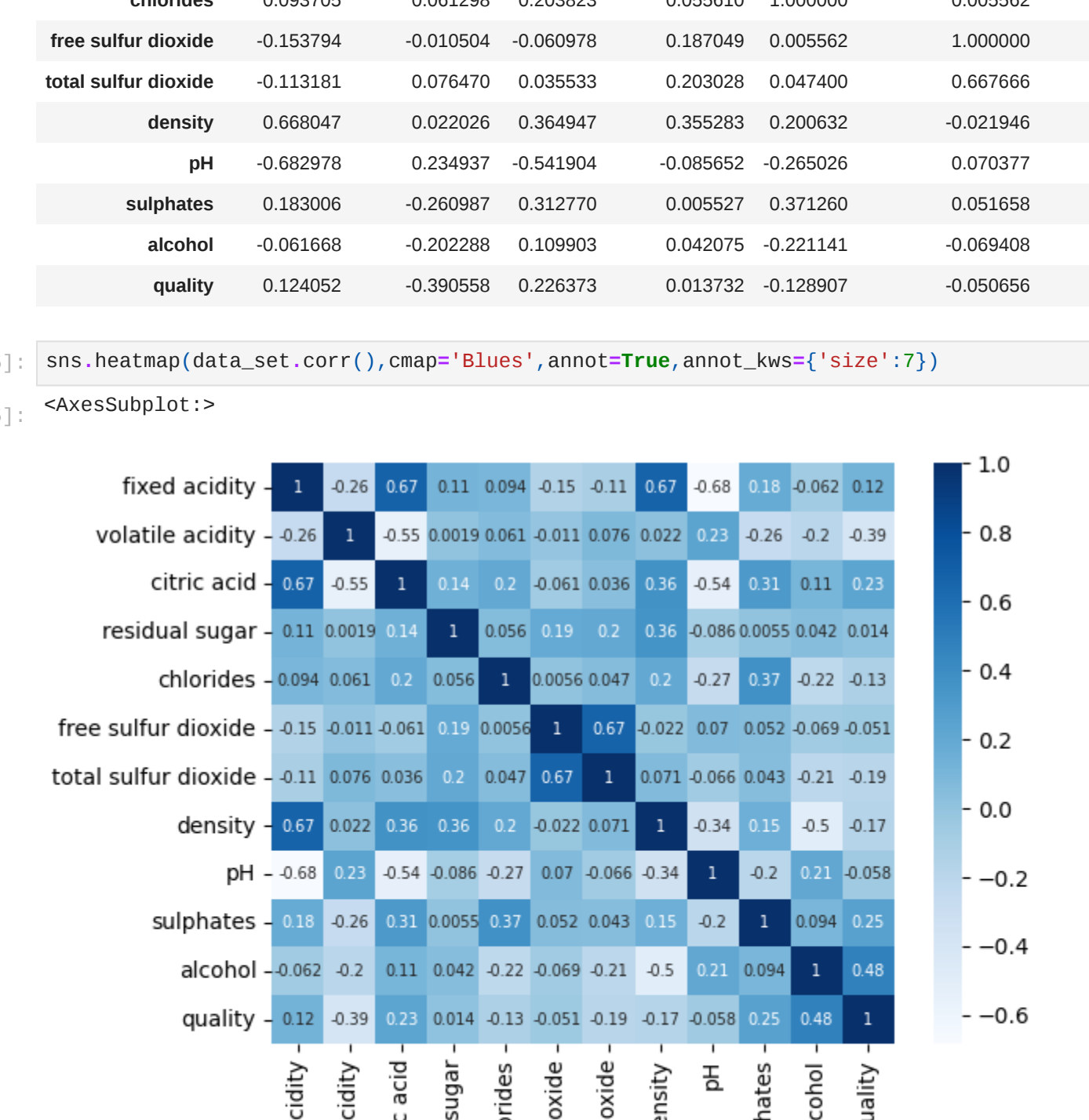
In [28]:
```python
data_set.corr()
```

Out[28]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.668047	-0.682978	0.183006	-0.061668	0.124052
volatile acidity	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.022026	0.234937	-0.260987	-0.202288	-0.390558
citric acid	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.364947	-0.541904	0.312770	0.109903	0.226373
residual sugar	0.114777	0.001918	0.143577	1.000000	0.055610	0.187049	0.203028	0.355283	-0.085652	0.005527	0.042075	0.013732
chlorides	0.093705	0.061298	0.203823	0.055610	1.000000	0.005562	0.047400	0.200632	-0.265026	0.371260	-0.221141	-0.128907
free sulfur dioxide	-0.153794	-0.010504	-0.060978	0.187049	0.005562	1.000000	0.667666	-0.021946	0.070377	0.051658	-0.069408	-0.050656
total sulfur dioxide	-0.113181	0.076470	0.035533	0.203028	0.047400	0.667666	1.000000	0.071269	-0.066495	0.042947	-0.205654	-0.185100
density	0.668047	0.022026	0.364947	0.355283	0.200632	-0.021946	0.071269	1.000000	-0.341699	0.148506	-0.496180	-0.174919
pH	-0.682978	0.234937	-0.541904	-0.085652	-0.265026	0.070377	-0.066495	-0.341699	1.000000	-0.196648	0.205633	-0.057731
sulphates	0.183006	-0.260987	0.312770	0.005527	0.371260	0.051658	0.042947	0.148506	-0.196648	1.000000	0.093595	0.251397
alcohol	-0.061668	-0.202288	0.109903	0.042075	-0.221141	-0.069408	-0.205654	-0.496180	0.205633	0.093595	1.000000	0.476166
quality	0.124052	-0.390558	0.226373	0.013732	-0.128907	-0.050656	-0.185100	-0.174919	-0.057731	0.251397	0.476166	1.000000

In [45]:
```python
sns.heatmap(data_set.corr(),cmap='Blues',annot=True,annot_kws={'size':7})
```

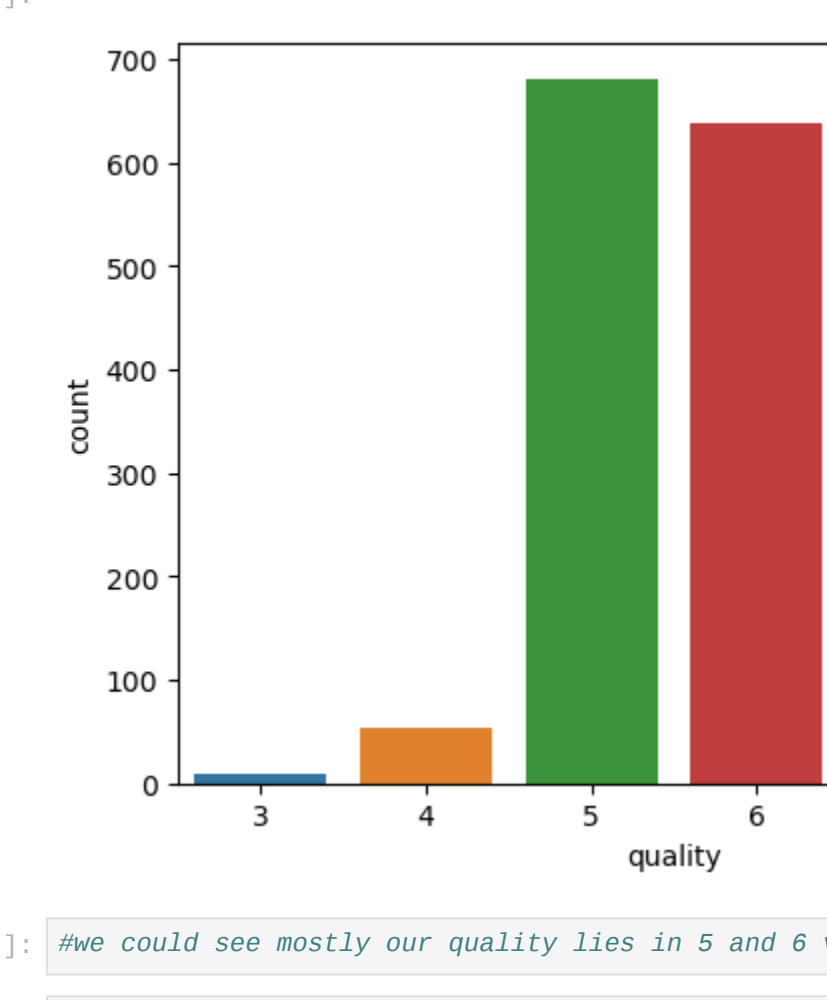
Out[45]:
```
&lt;AxesSubplot:&gt;
```



In [ ]:
```python
#wecould see that alcohol, fixed acid,citric acid, residual sugar, sulphates have positive relation which means if the
#quantity of these items increase the quality of our wine increases and all others have negative corelation
```

In [49]:
```python
sns.countplot(data=data_set,x='quality')
```

Out[49]:
```
<AxesSubplot:xlabel='quality', ylabel='count'>
```

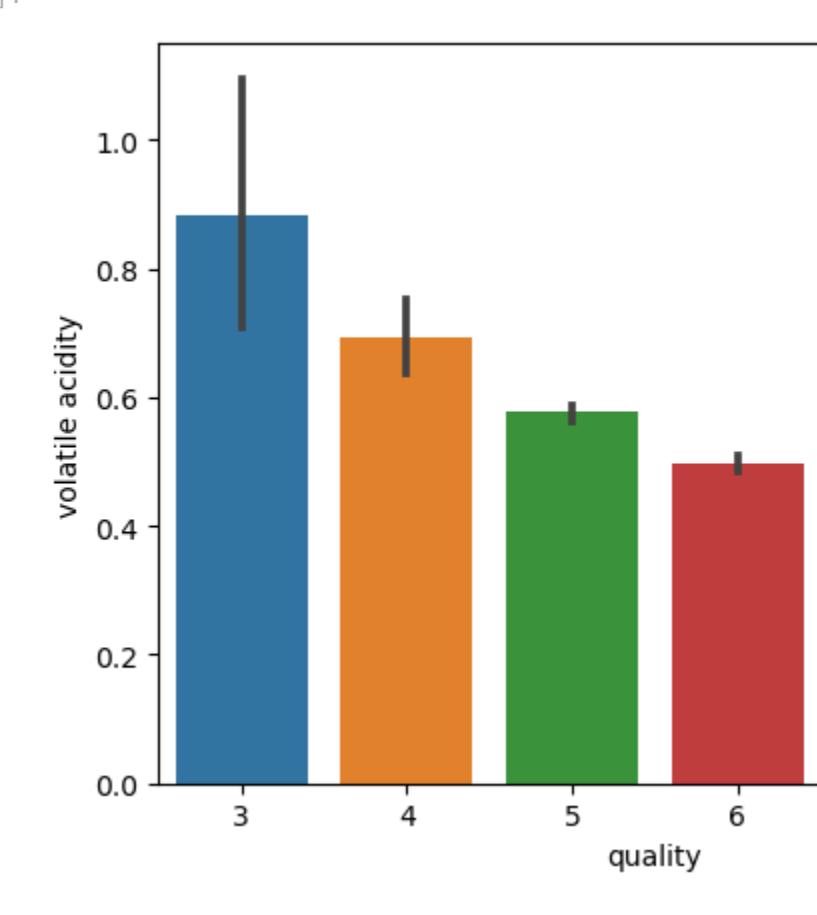


In [ ]:
```python
#we could see mostly our quality lies in 5 and 6 values.
```

In [ ]:
```python
# finding values of fields having good wine quality
```

In [50]:
```python
sns.barplot(x='quality',y='volatile acidity',data=data_set)
```

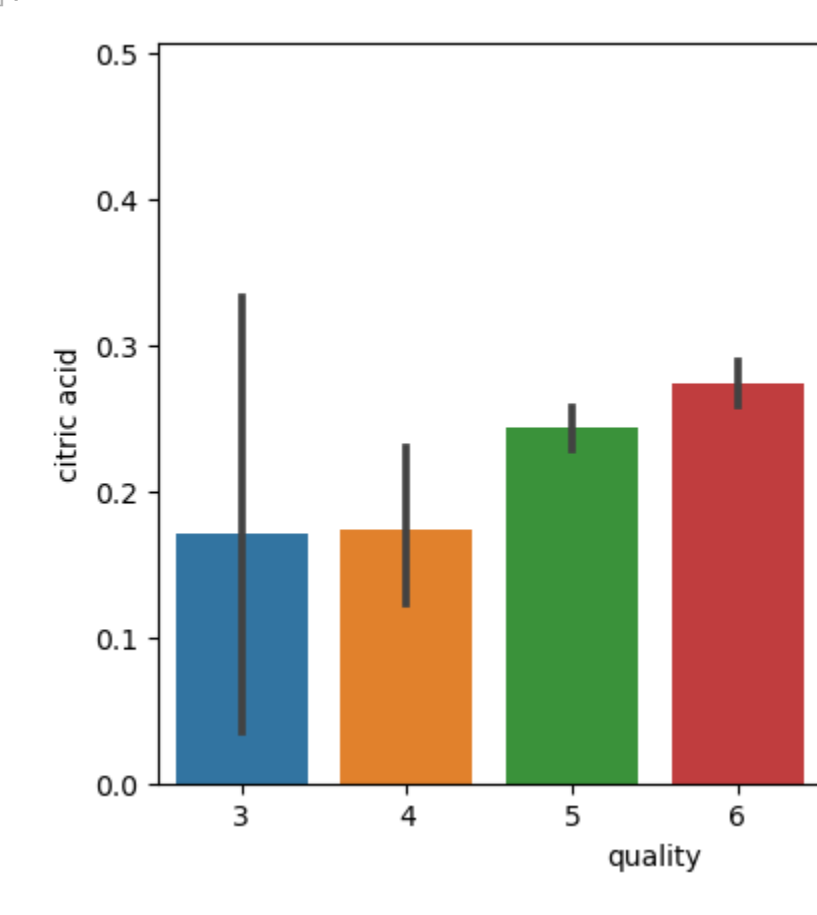
Out[50]:
```
<AxesSubplot:xlabel='quality', ylabel='volatile acidity'>
```



In [ ]:
```python
#the volatile acidity should be in the range of 0.4 for good wine quality
```

In [60]:
```python
sns.barplot(x='quality',y='citric acid',data=data_set)
```

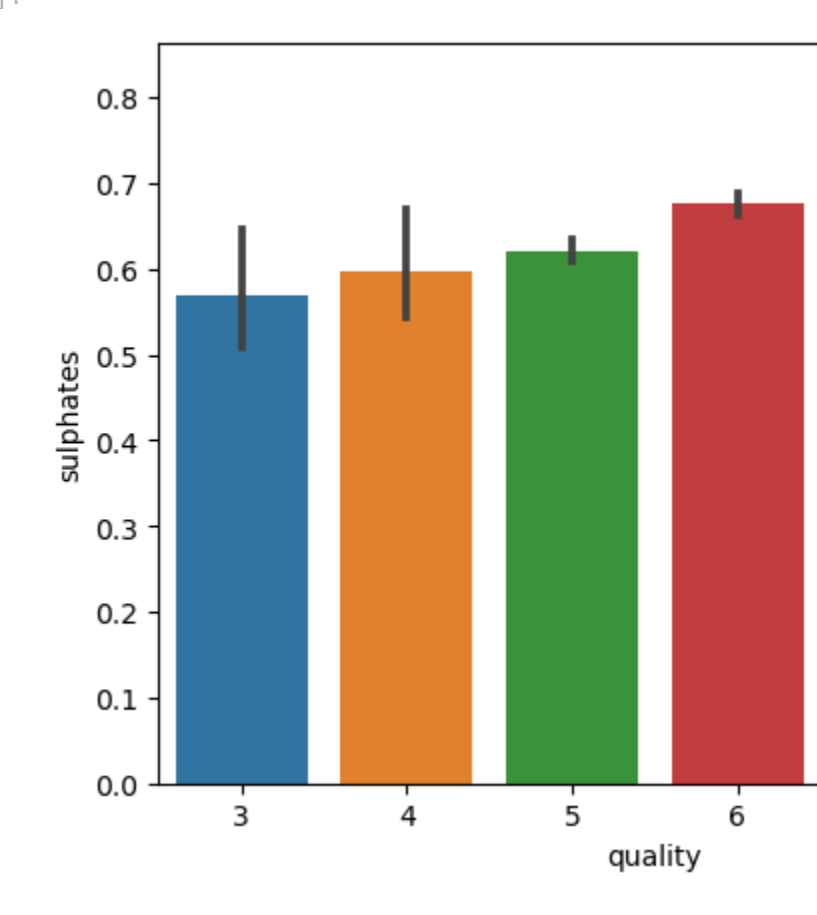
Out[60]:
```
<AxesSubplot:xlabel='quality', ylabel='citric acid'>
```



In [ ]:
```python
#the citric acid should be in the range of 0.38 and above for good wine quality
```

In [52]:
```python
sns.barplot(x='quality',y='sulphates',data=data_set)
```

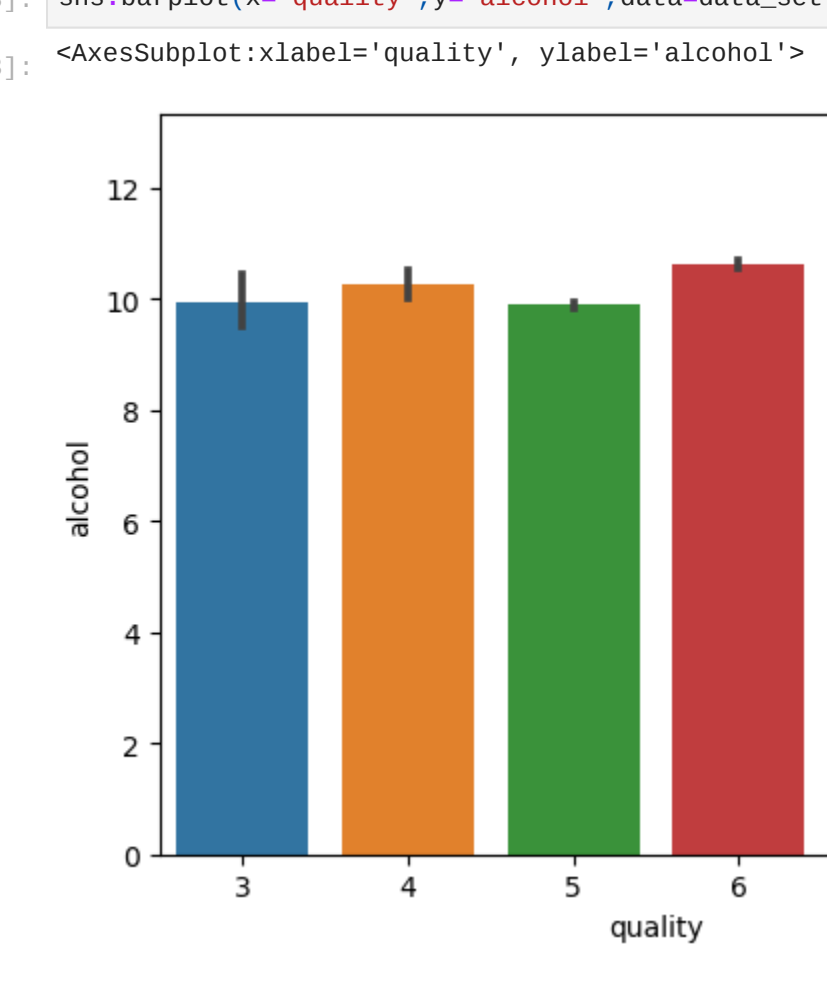
Out[52]:
```
<AxesSubplot:xlabel='quality', ylabel='sulphates'>
```



In [ ]:
```python
#the sulphates should be in the range of 0.7 and above for good wine quality
```

In [53]:
```python
sns.barplot(x='quality',y='alcohol',data=data_set)
```

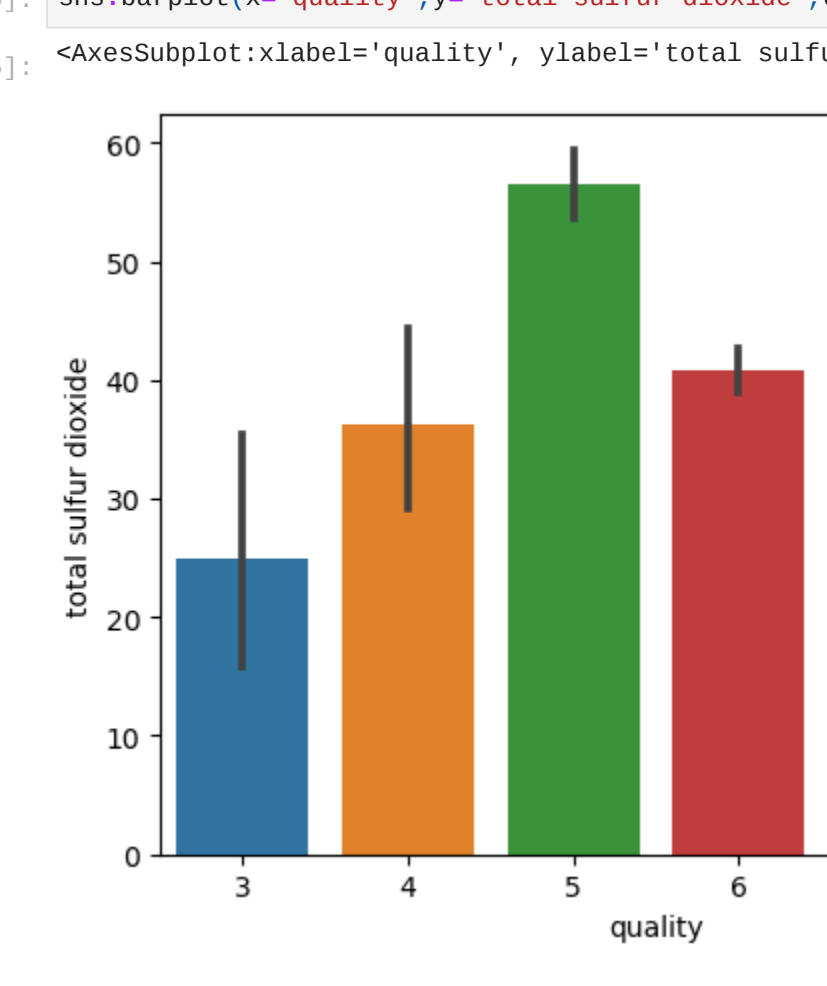
Out[53]:
```
<AxesSubplot:xlabel='quality', ylabel='alcohol'>
```



In [ ]:
```python
#the alcohol should be in the range of 11 and above for good wine quality
```

In [55]:
```python
sns.barplot(x='quality',y='total sulfur dioxide',data=data_set)
```

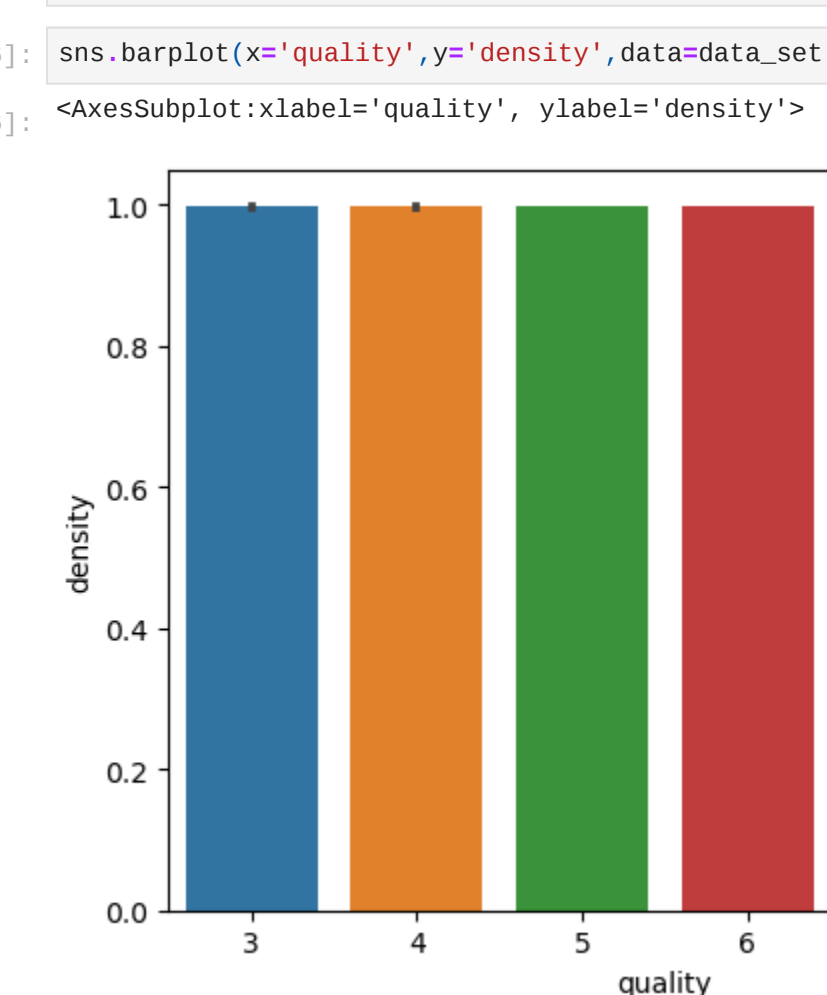
Out[55]:
```
<AxesSubplot:xlabel='quality', ylabel='total sulfur dioxide'>
```



In [ ]:
```python
#the total sulfur dioxide should be in the range of 35 and above for good wine quality
```

In [56]:
```python
sns.barplot(x='quality',y='density',data=data_set)
```

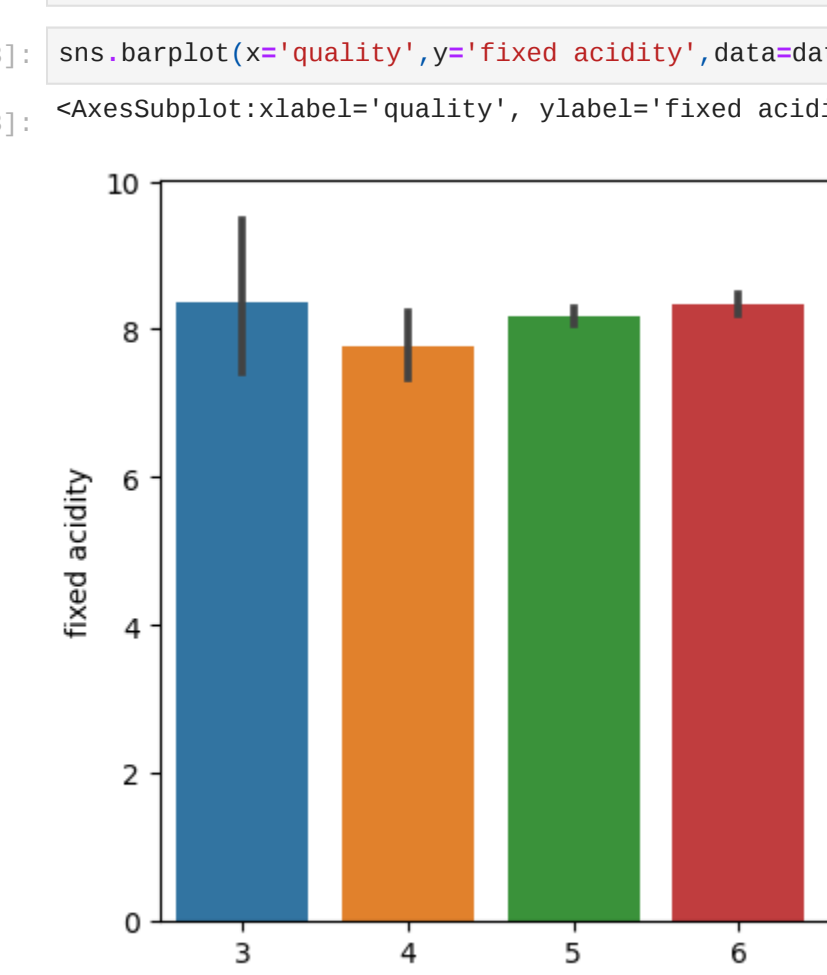
Out[56]:
```
<AxesSubplot:xlabel='quality', ylabel='density'>
```



In [ ]:
```python
#the density should be in the range of 1 for good wine quality
```

In [63]:
```python
sns.barplot(x='quality',y='fixed acidity',data=data_set)
```

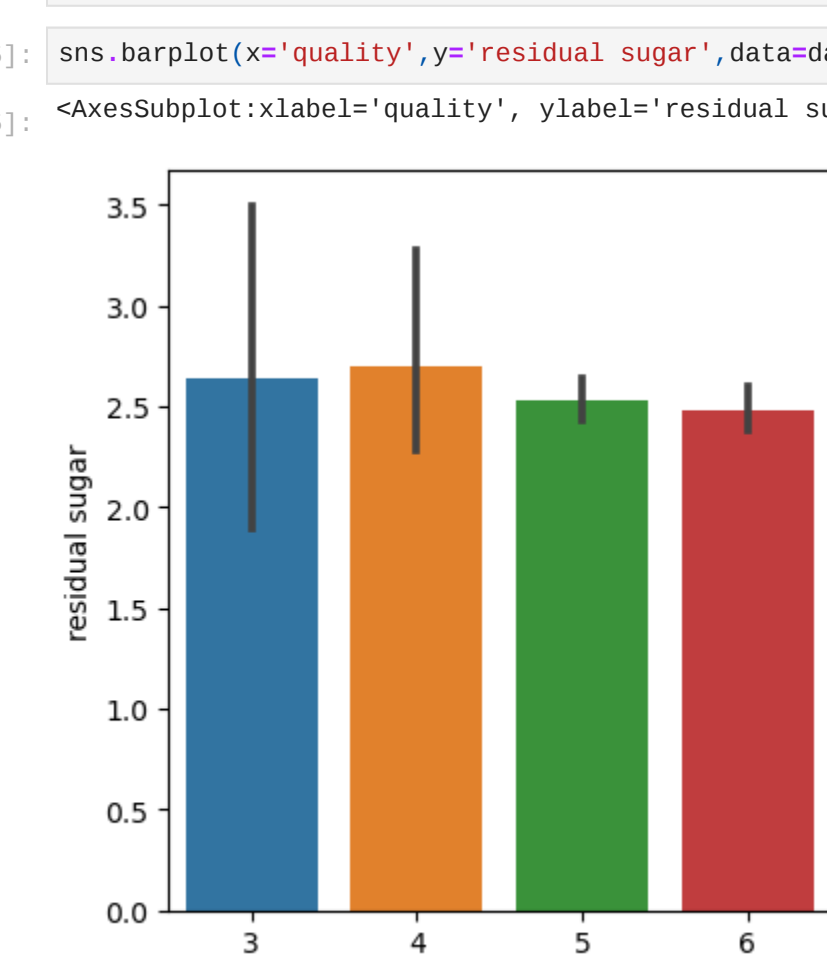
Out[63]:
```
<AxesSubplot:xlabel='quality', ylabel='fixed acidity'>
```



In [ ]:
```python
#the fixed acidity should be in the range of 8.8 and above for good wine quality
```

In [65]:
```python
sns.barplot(x='quality',y='residual sugar',data=data_set)
```

Out[65]:
```
<AxesSubplot:xlabel='quality', ylabel='residual sugar'>
```



In [ ]:
```python
#the residual sugar should be in the range of 2.7 for good wine quality
```

In [61]:
```python
data_set
```

Out[61]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1599 rows × 12 columns

In [ ]:
```python
# here i am taking the value of the first row whose wine quality is 5 and is of bad quality
```

In [72]:
```python
values_to_predict = (7.4,0.7,0,1.9,0.076,11,34,0.9978,3.51,0.56,9.4)
value_array = np.asarray(values_to_predict)
value_reshaped=value_array.reshape(1,-1)
predict = model.predict(value_reshaped)
if (predict[0]==1):
    print('The quality is good')
else:
    print('The quality is bad')
```

```
The quality is bad
C:\Users\jowin\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```

In [ ]:
```python
#the result came as bad quality and i am changing positive correlation factors to the obseverd values from the diagram
#fixed acidity=8.8, citric acid=0.38,alcohol = 11.7, sulphates to 0.74 and reduced the negative correlation factor values such as volatile acidity to 0.4
#residual sugar to 2.7,
```

In [82]:
```python
values_to_predict = (8.8,0.4,0.38,2.7,0.076,11,34,0.9978,3.51,0.74,11.7)
value_array = np.asarray(values_to_predict)
value_reshaped=value_array.reshape(1,-1)
predict = model.predict(value_reshaped)
if (predict[0]==1):
    print('The quality is good')
else:
    print('The quality is bad')
```

```
The quality is good
C:\Users\jowin\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```

In [ ]:
```python
#changing these parametr values increases the quality of alcohol from bad to good
```