In [238…]
```
# Data Cleaning and Analysis

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

import seaborn as sns
```

In [239…]
```
# read in the data
raw_data = pd.read_csv('winequality-red.csv', sep=';')
```

In [240…]
```
# remove repeated data
raw_data = raw_data.drop_duplicates()
raw_data.info()
raw_data.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1359 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1359 non-null   float64
 1   volatile acidity      1359 non-null   float64
 2   citric acid           1359 non-null   float64
 3   residual sugar        1359 non-null   float64
 4   chlorides             1359 non-null   float64
 5   free sulfur dioxide   1359 non-null   float64
 6   total sulfur dioxide  1359 non-null   float64
 7   density               1359 non-null   float64
 8   pH                    1359 non-null   float64
 9   sulphates             1359 non-null   float64
 10  alcohol               1359 non-null   float64
 11  quality               1359 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 138.0 KB
```

Out[240]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	
count	1359.000000	1359.000000	1359.000000	1359.000000	1359.000000	1359.000000	13
mean	8.310596	0.529478	0.272333	2.523400	0.088124	15.893304	
std	1.736990	0.183031	0.195537	1.352314	0.049377	10.447270	
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	0.430000	2.600000	0.091000	21.000000	
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	2

In [241…

```
# remove outliers
for col in raw_data.columns:
    if col != 'quality':
        iqr = raw_data[col].quantile(0.75) - raw_data[col].quantile(0.25)
        upper_bound = raw_data[col].quantile(0.75) + 3 * iqr
        lower_bound = raw_data[col].quantile(0.25) - 3 * iqr
        raw_data = raw_data[(raw_data[col] < upper_bound) & (raw_data[col] >

raw_data.info()
raw_data.describe()
```

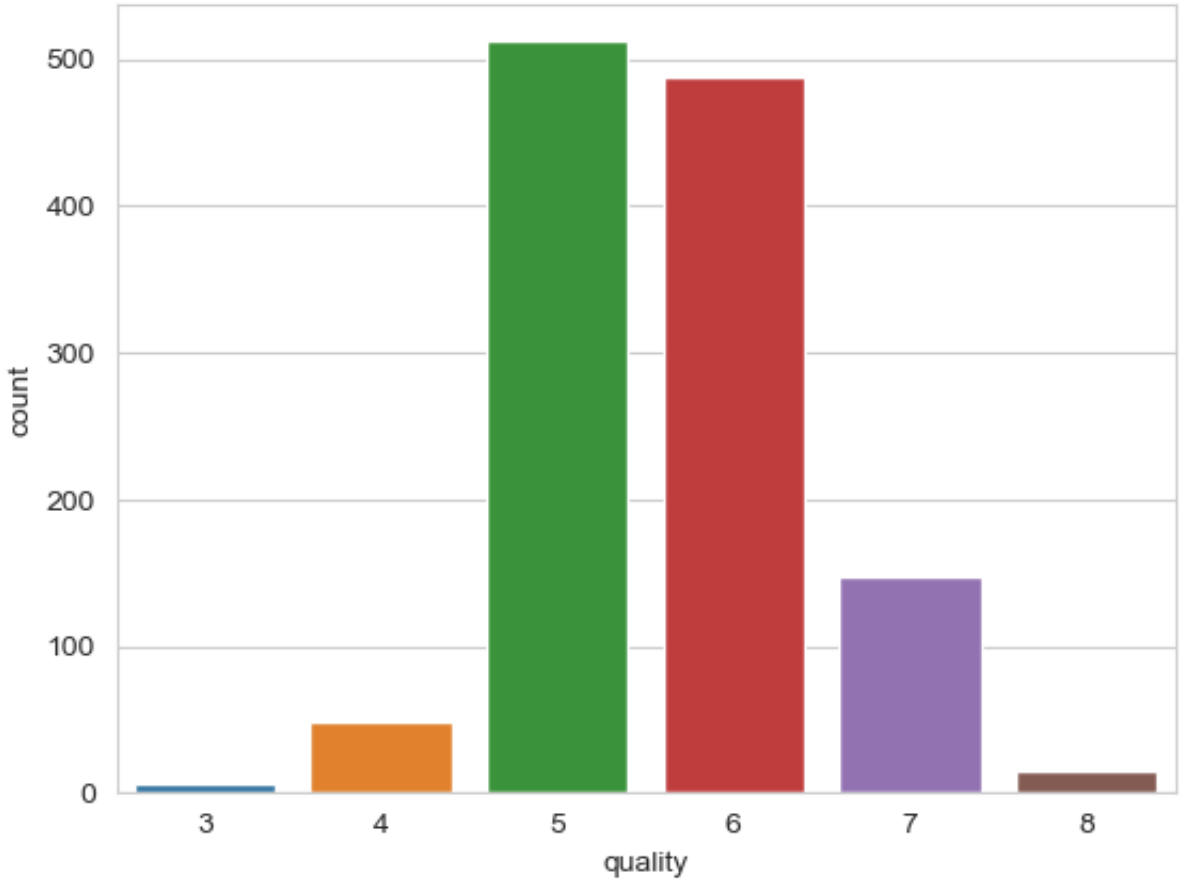
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1217 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1217 non-null   float64
 1   volatile acidity      1217 non-null   float64
 2   citric acid           1217 non-null   float64
 3   residual sugar        1217 non-null   float64
 4   chlorides             1217 non-null   float64
 5   free sulfur dioxide   1217 non-null   float64
 6   total sulfur dioxide  1217 non-null   float64
 7   density               1217 non-null   float64
 8   pH                    1217 non-null   float64
 9   sulphates             1217 non-null   float64
 10  alcohol               1217 non-null   float64
 11  quality               1217 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 123.6 KB
```

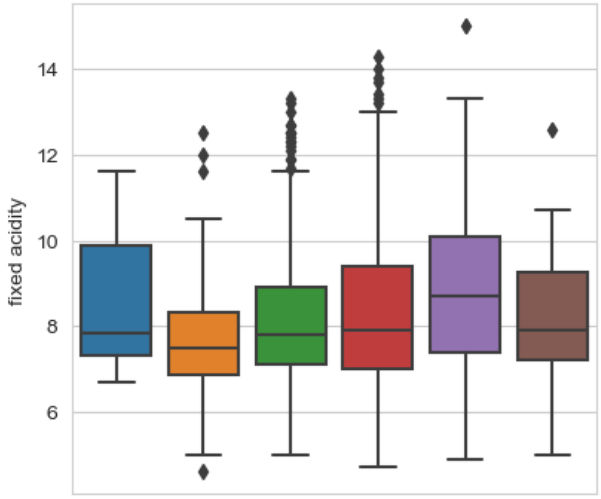
Out[241]:

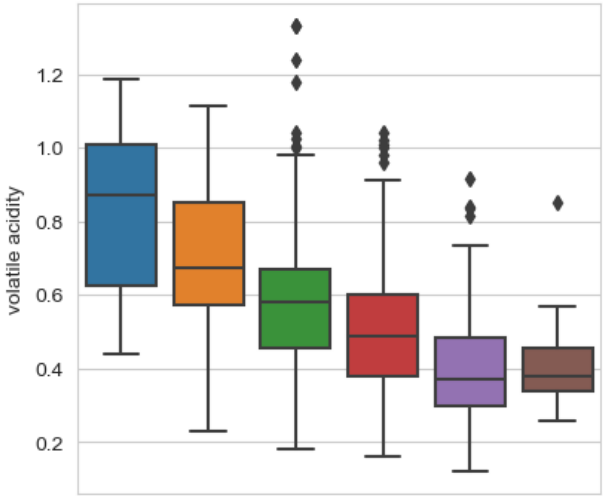
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total d
count	1217.000000	1217.000000	1217.000000	1217.000000	1217.000000	1217.000000	1217.0
mean	8.286360	0.526652	0.262021	2.278554	0.079132	15.759655	45.4
std	1.701773	0.180310	0.190105	0.593928	0.016570	9.894299	30.9
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.0
25%	7.100000	0.390000	0.090000	1.900000	0.069000	8.000000	22.0
50%	7.900000	0.520000	0.250000	2.200000	0.078000	14.000000	37.0
75%	9.200000	0.640000	0.420000	2.500000	0.088000	21.000000	60.0
max	15.000000	1.330000	0.760000	4.700000	0.147000	57.000000	165.0

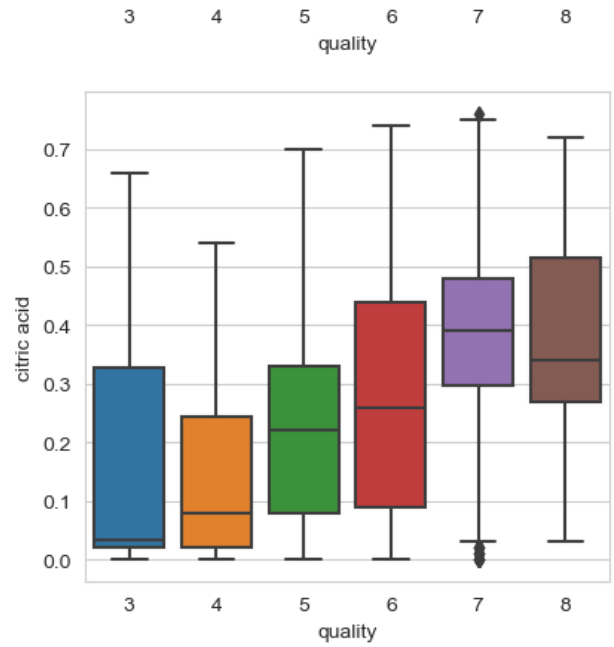
In [242…

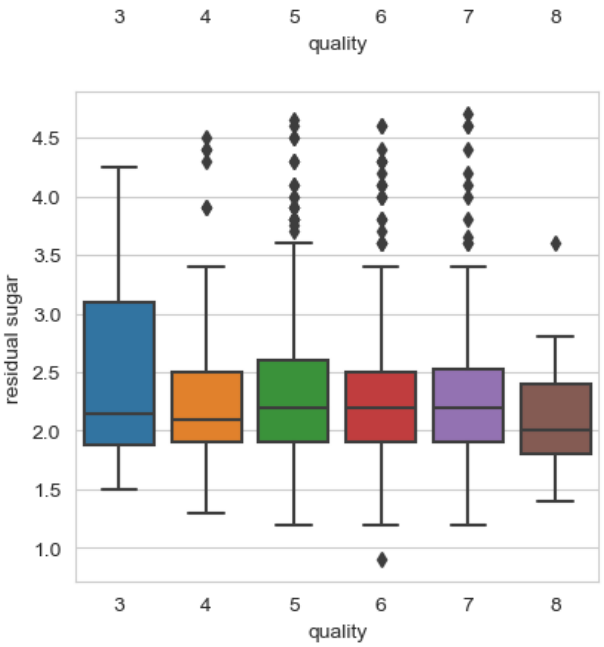
```
# visualization
sns.countplot(x='quality', data=raw_data)

# box plot for each feature
fig, ax = plt.subplots(6, 2, figsize=(10, 30))
for i, col in enumerate(raw_data.columns):
    if col != 'quality':
        sns.boxplot(x='quality', y=col, data=raw_data, ax=ax[i//2][i%2])

plt.show()
# heatmap correlation matrix
fig = plt.figure(figsize=(10, 8))
corr = raw_data.corr()
sns.heatmap(corr, annot=True, cmap='BrBG')
```

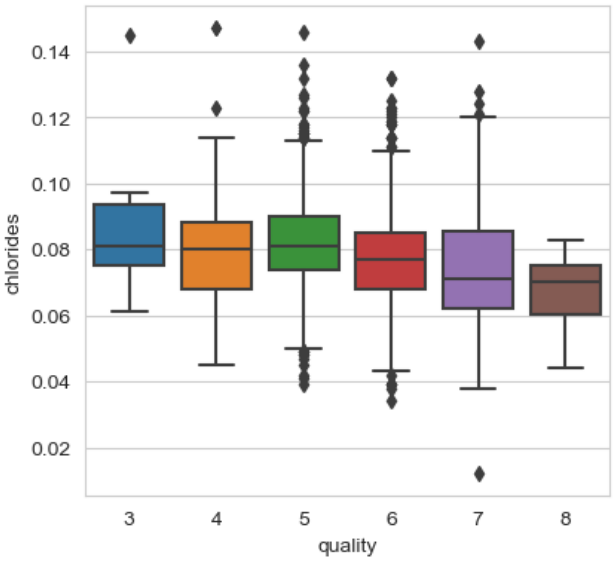


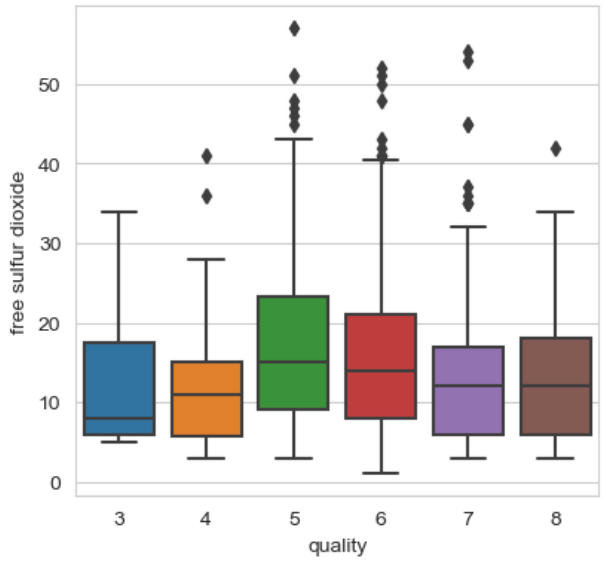


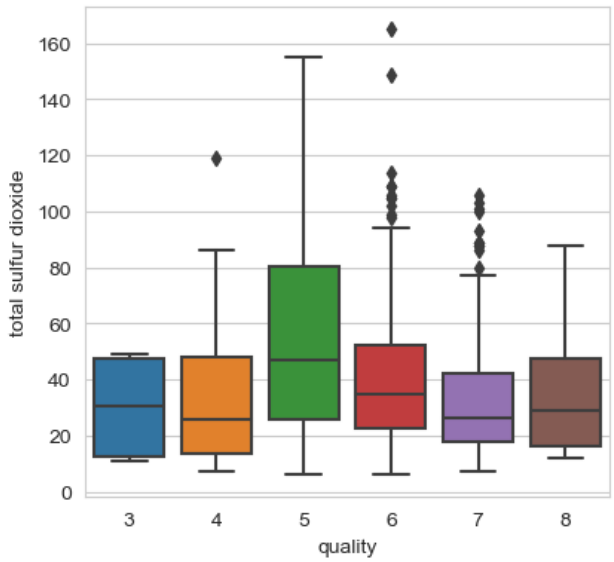


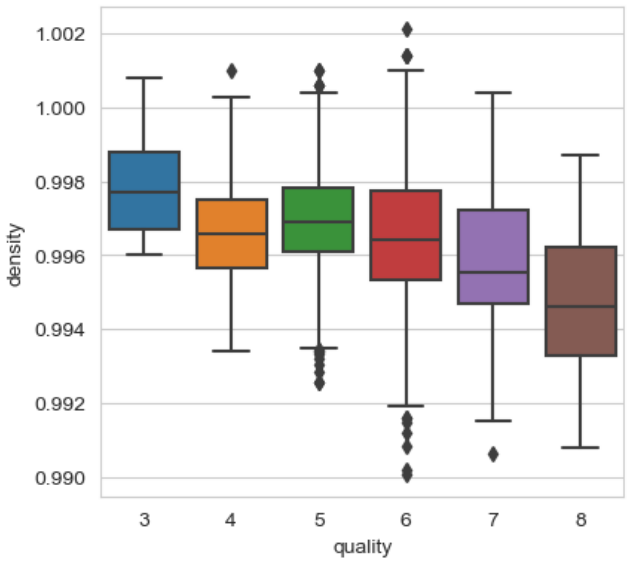

3 4 5 6 7 8
quality
citric acid
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7


3 4 5 6 7 8
quality
residual sugar
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5

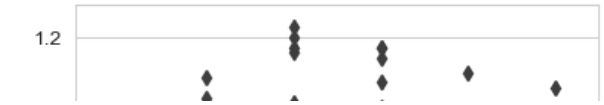

3 4 5 6 7 8
quality
chlorides
0.02 0.04 0.06 0.08 0.10 0.12 0.14

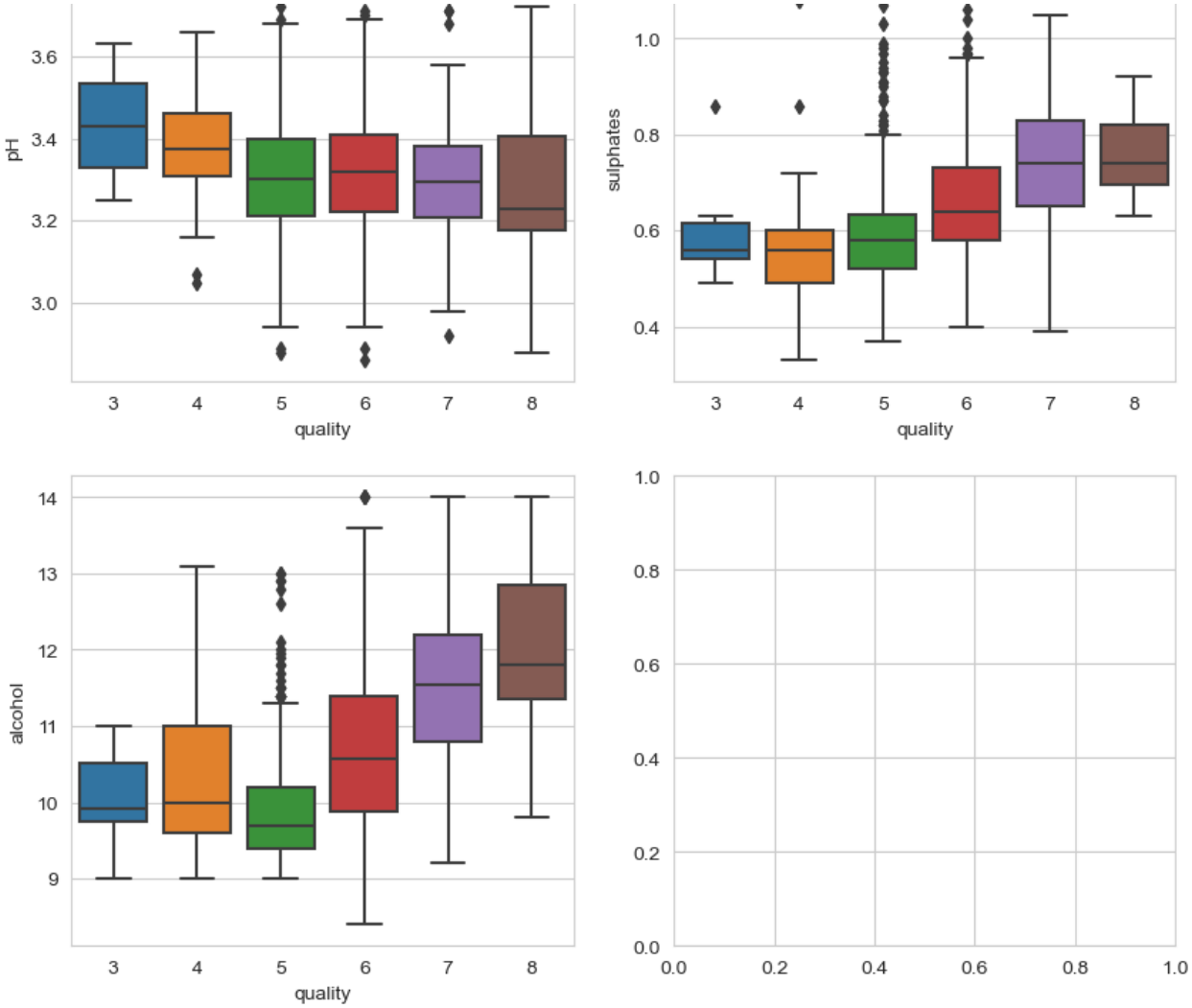

3 4 5 6 7 8
quality
free sulfur dioxide
0 10 20 30 40 50


3 4 5 6 7 8
quality
total sulfur dioxide
0 20 40 60 80 100 120 140 160

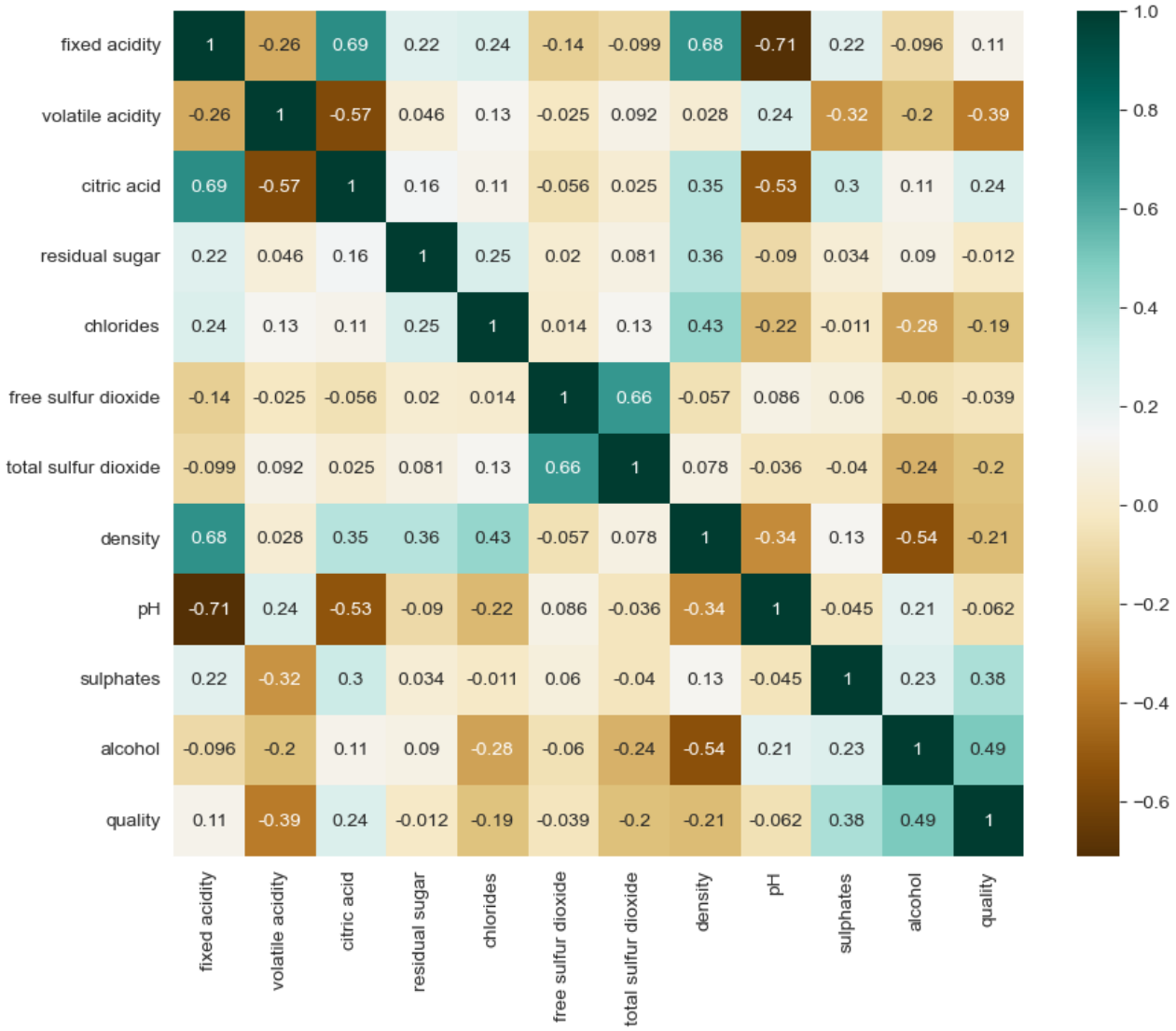

3 4 5 6 7 8
quality
density
0.990 0.992 0.994 0.996 0.998 1.000 1.002


3.8


1.2



Out[242]: <AxesSubplot: >



In [243…

```
# normalize features except quality
from sklearn.preprocessing import StandardScaler
input_features = raw_data.drop('quality', axis=1)

scaler = StandardScaler()
input_features = scaler.fit_transform(input_features)

# contact quality to input features
processed_data = np.concatenate((input_features, raw_data['quality'].values.
```

In [244…]
```
# show processed data
processed_data = pd.DataFrame(processed_data, columns=raw_data.columns)
processed_data.info()
processed_data.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1217 entries, 0 to 1216
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1217 non-null   float64
 1   volatile acidity      1217 non-null   float64
 2   citric acid           1217 non-null   float64
 3   residual sugar        1217 non-null   float64
 4   chlorides             1217 non-null   float64
 5   free sulfur dioxide   1217 non-null   float64
 6   total sulfur dioxide  1217 non-null   float64
 7   density               1217 non-null   float64
 8   pH                    1217 non-null   float64
 9   sulphates             1217 non-null   float64
 10  alcohol               1217 non-null   float64
 11  quality               1217 non-null   float64
dtypes: float64(12)
memory usage: 114.2 KB
```

Out[244]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free d
count	1.217000e+03	1.217000e+03	1217.000000	1.217000e+03	1.217000e+03	1.21700
mean	-2.802469e-16	-4.028550e-16	0.000000	-1.634774e-16	-4.437243e-16	-5.25463
std	1.000411e+00	1.000411e+00	1.000411	1.000411e+00	1.000411e+00	1.00041
min	-2.167078e+00	-2.256216e+00	-1.378863	-2.322031e+00	-4.053072e+00	-1.49234
25%	-6.974183e-01	-7.581809e-01	-0.905247	-6.376348e-01	-6.117312e-01	-7.84577
50%	-2.271271e-01	-3.690493e-02	-0.063261	-1.323158e-01	-6.836149e-02	-1.77918
75%	5.370962e-01	6.288883e-01	0.831348	3.730032e-01	5.353826e-01	5.29850
max	3.946708e+00	4.457199e+00	2.620567	4.078676e+00	4.097473e+00	4.16980

In [245…]
```
print(raw_data['quality'].value_counts())
```

```
5    512
6    488
7    148
4     48
8     15
3      6
Name: quality, dtype: int64
```

In [246…]
```
# save processed data to csv
processed_data.to_csv('processed_data.csv', index=False)
```