In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')
```

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

In [3]:
```
df.head()
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	a
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	

In [4]:
```
df.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
dtype: int64
```

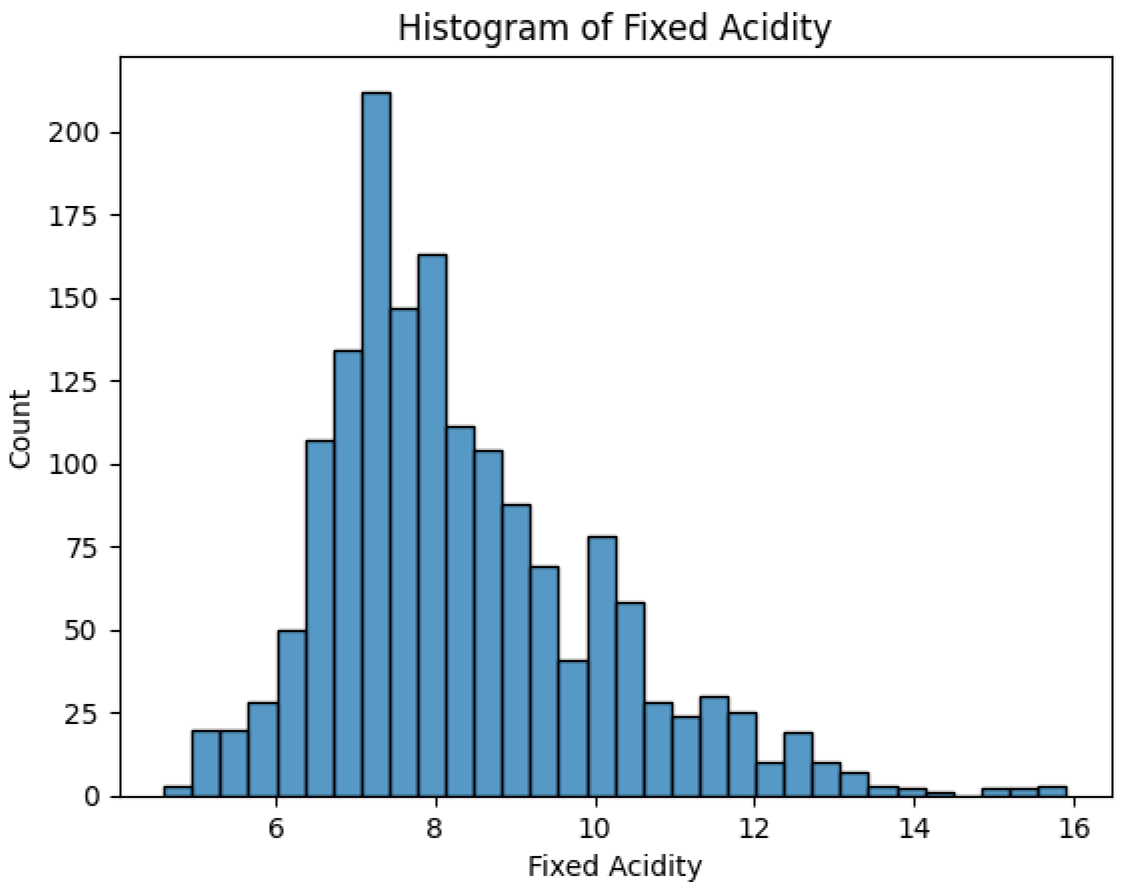
In [5]:
```
df.columns
```

Out[5]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

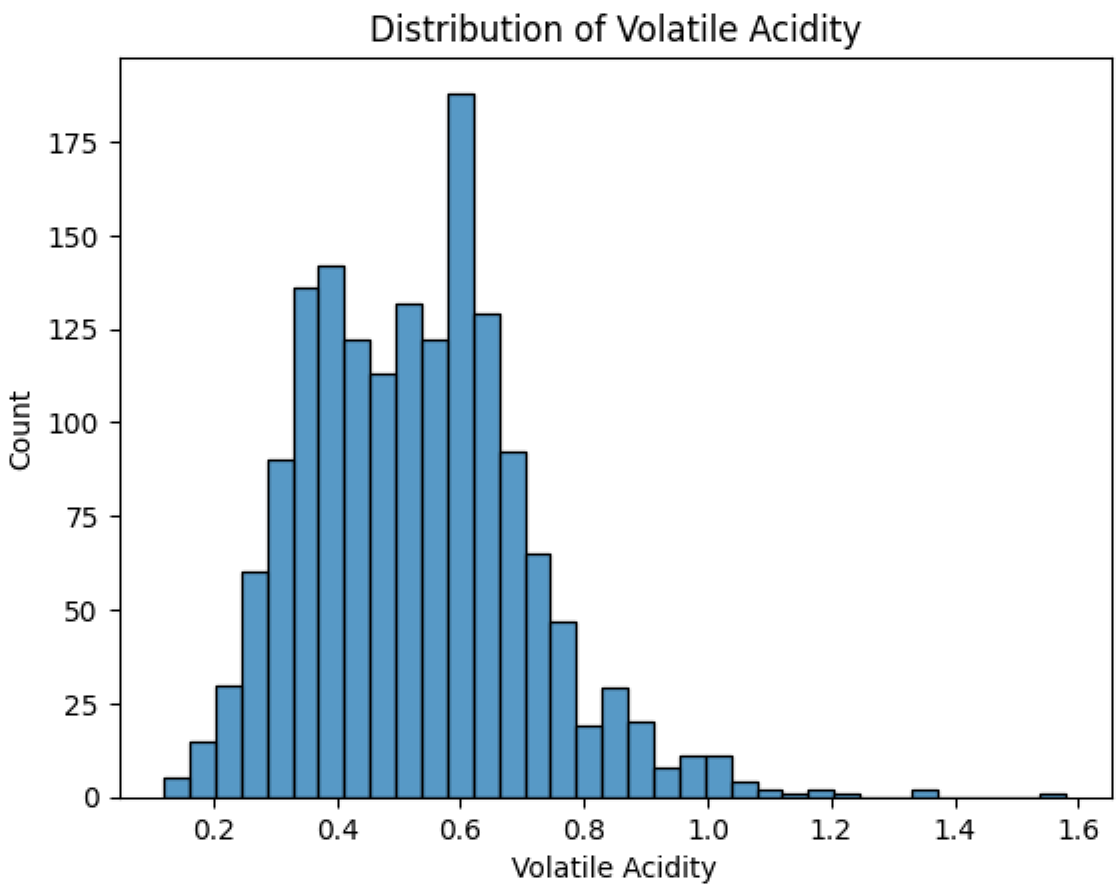
In [6]:
```
df['quality'].unique()
```

Out[6]:
```
array([5, 6, 7, 4, 8, 3], dtype=int64)
```

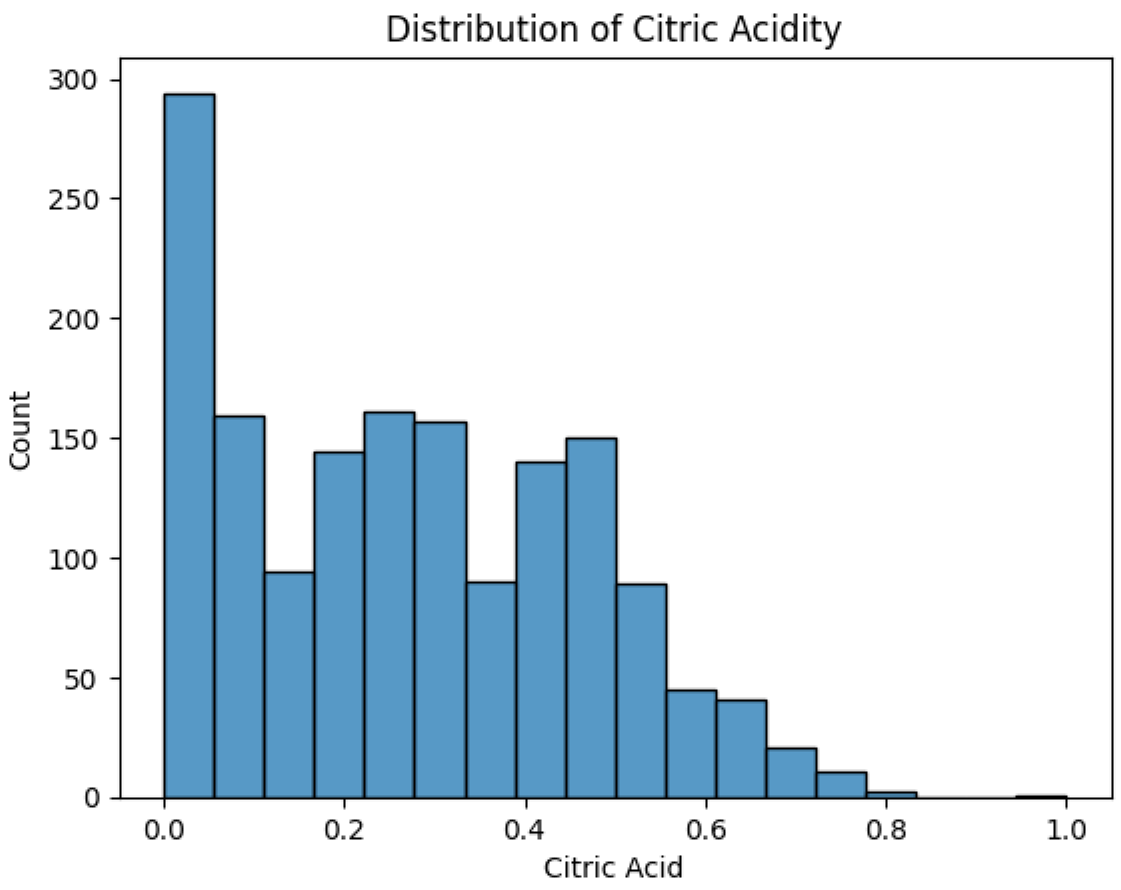
In [7]:
```python
sns.histplot(df['fixed acidity'])
plt.xlabel('Fixed Acidity')
plt.title('Histogram of Fixed Acidity')
plt.show()
```



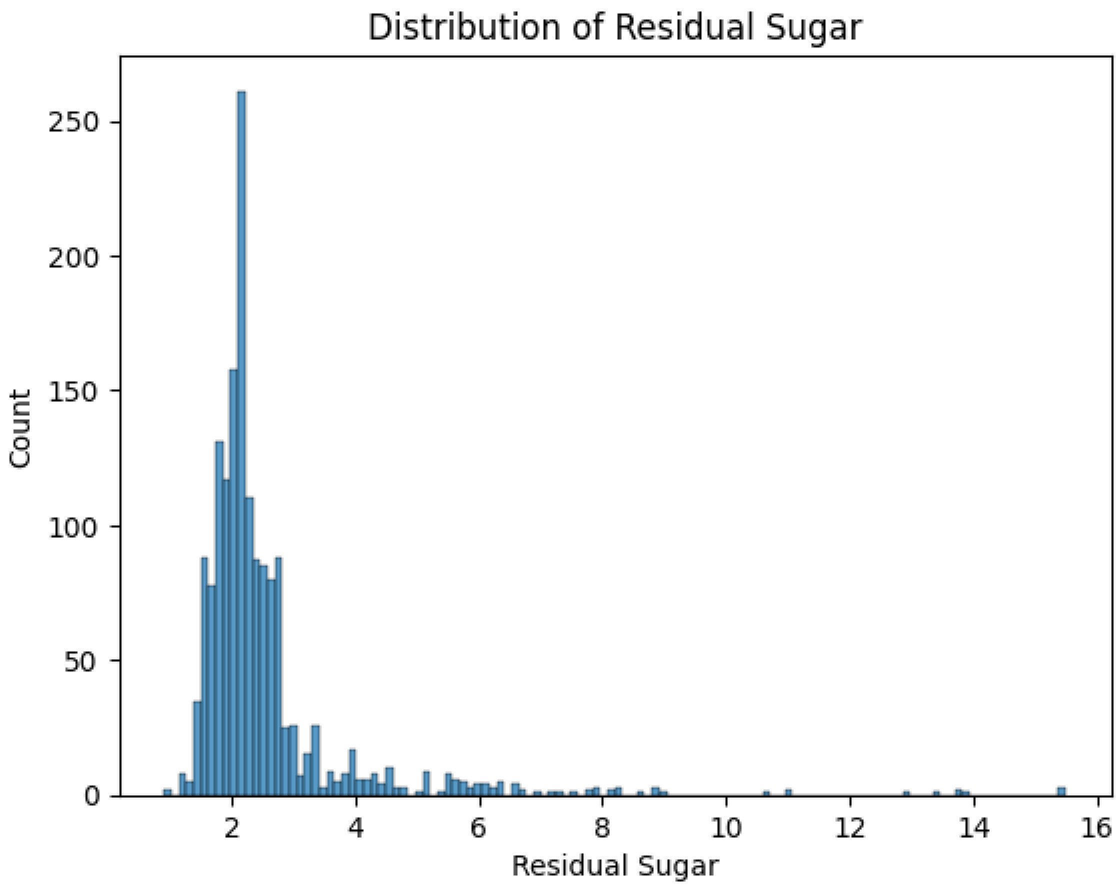
In [8]:
```python
sns.histplot(df['volatile acidity'])
plt.xlabel('Volatile Acidity')
plt.title('Distribution of Volatile Acidity')
plt.show()
```



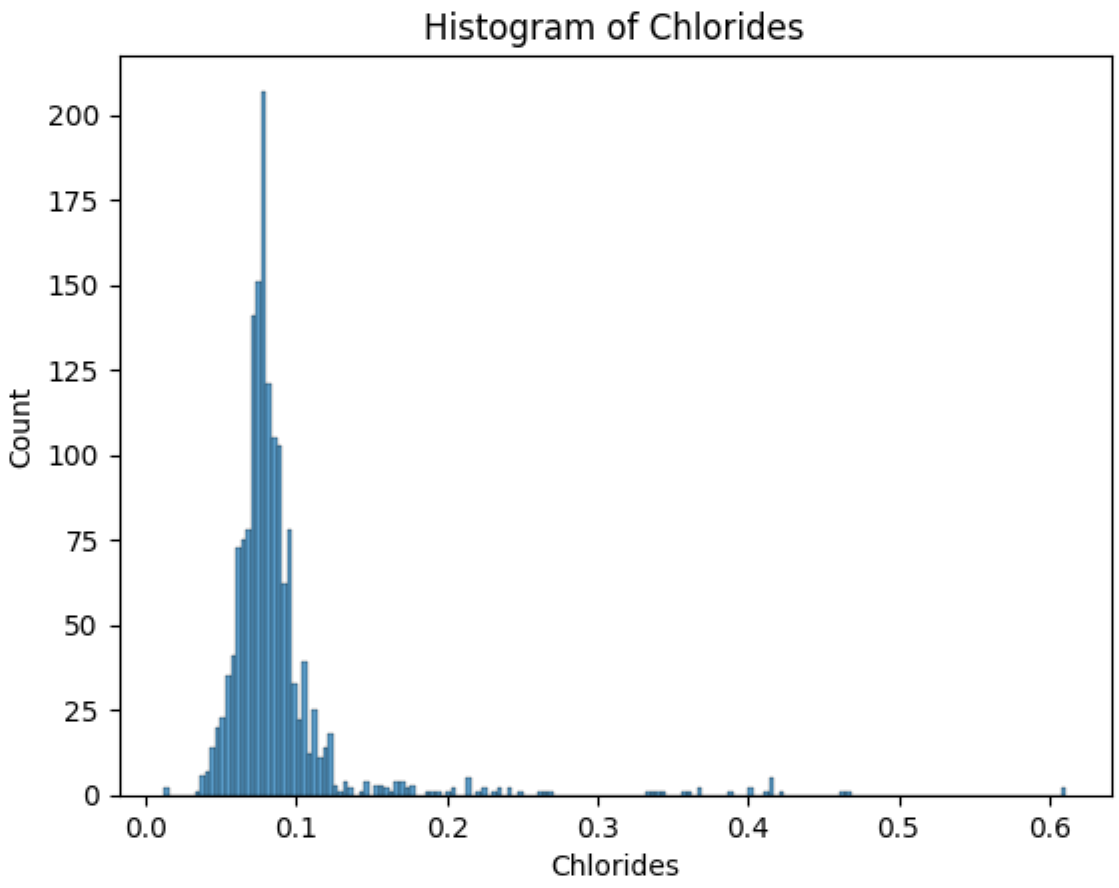
```
In [9]: sns.histplot(df['citric acid'])
        plt.xlabel('Citric Acid')
        plt.title('Distribution of Citric Acidity')
        plt.show()
```



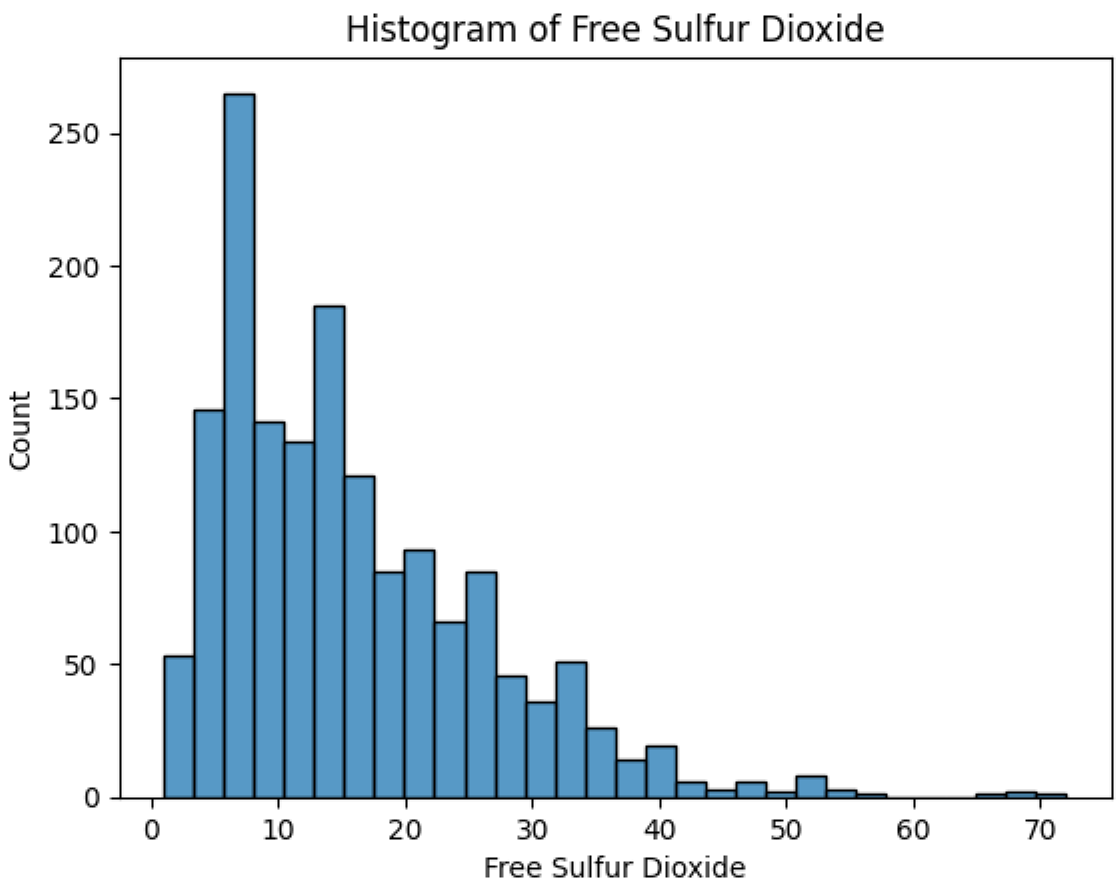
In [10]:
```
sns.histplot(df['residual sugar'])
plt.xlabel('Residual Sugar')
plt.title('Distribution of Residual Sugar')
plt.show()
```



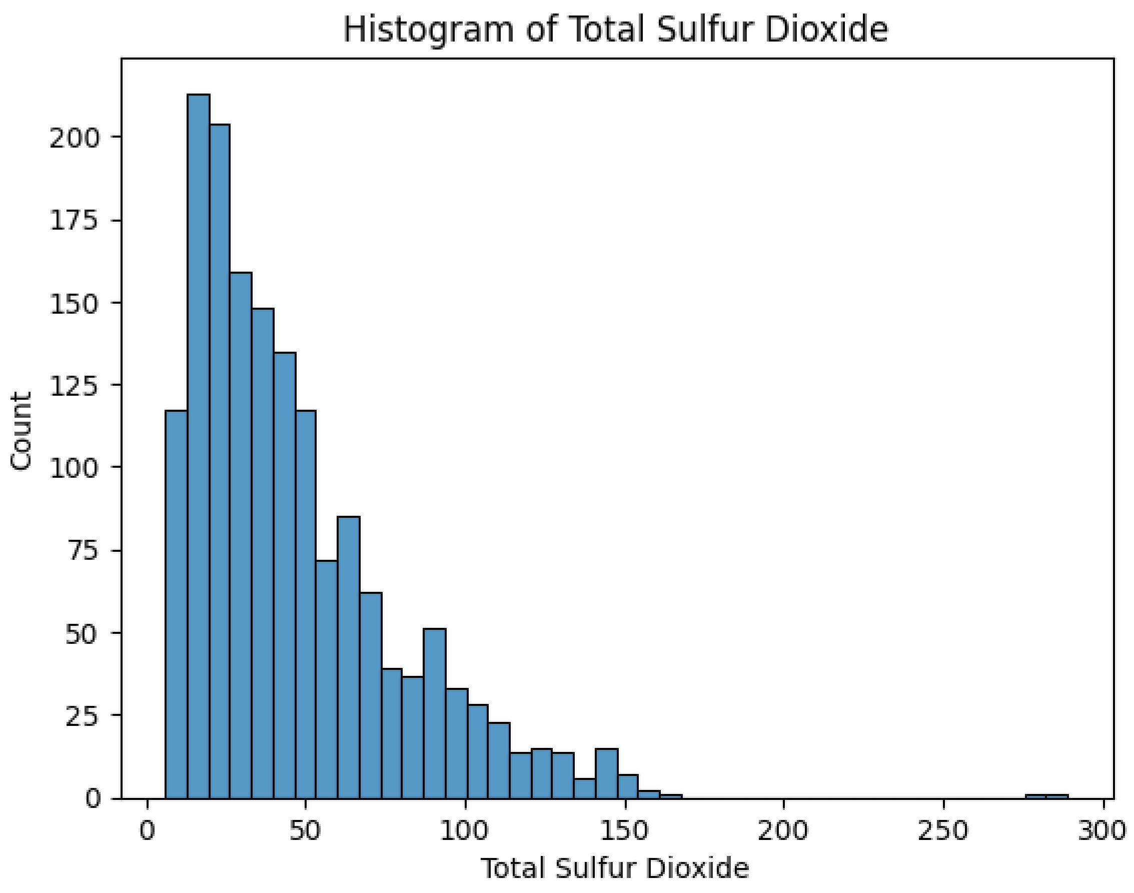
In [11]:
```
sns.histplot(df['chlorides'])
plt.xlabel('Chlorides')
plt.title('Histogram of Chlorides')
plt.show()
```



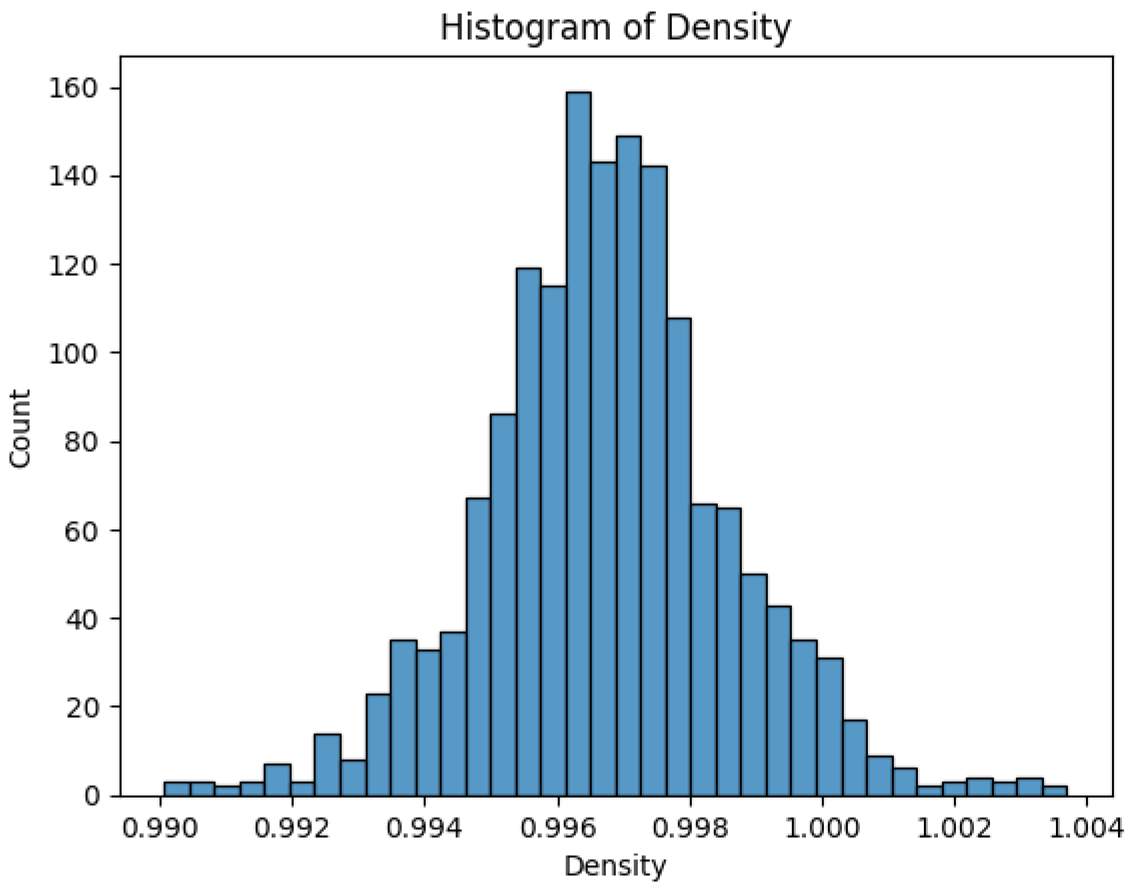
In [12]:
```
sns.histplot(df['free sulfur dioxide'])
plt.xlabel('Free Sulfur Dioxide')
plt.title('Histogram of Free Sulfur Dioxide')
plt.show()
```



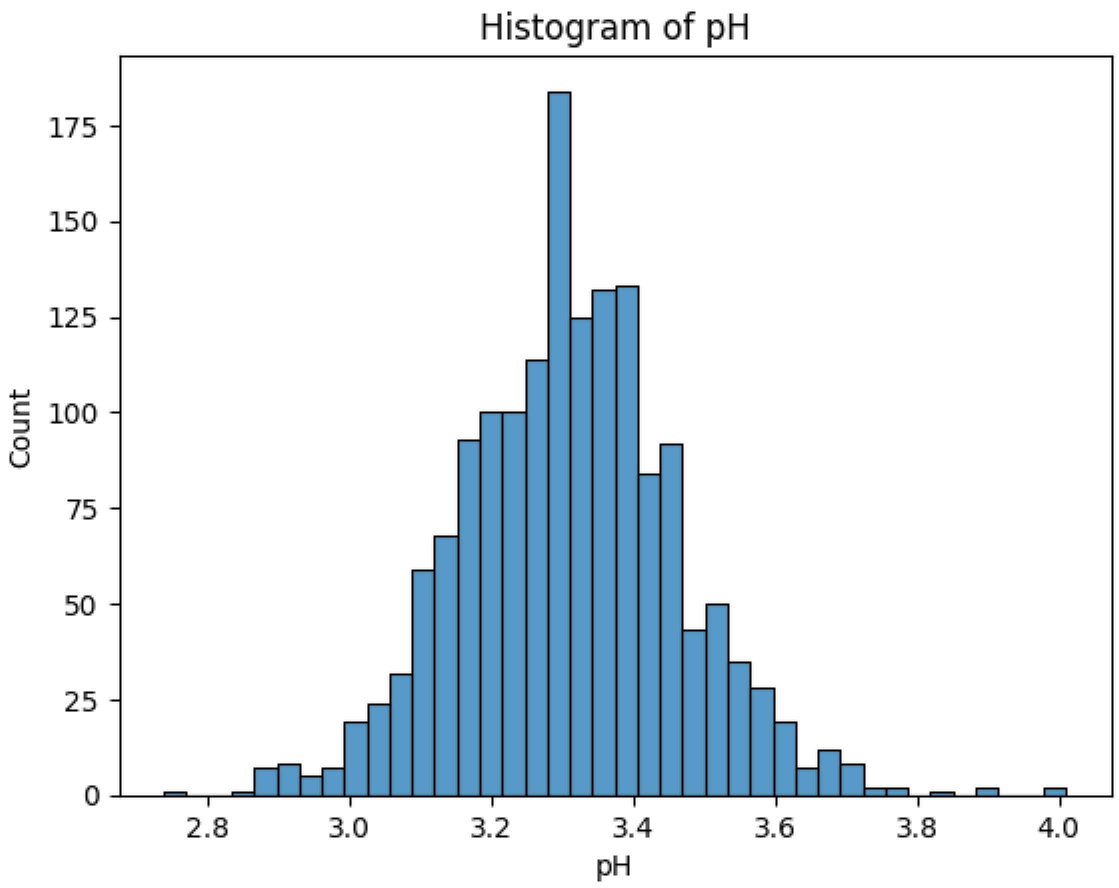
In [13]:
```
sns.histplot(df['total sulfur dioxide'])
plt.xlabel('Total Sulfur Dioxide')
plt.title('Histogram of Total Sulfur Dioxide')
plt.show()
```



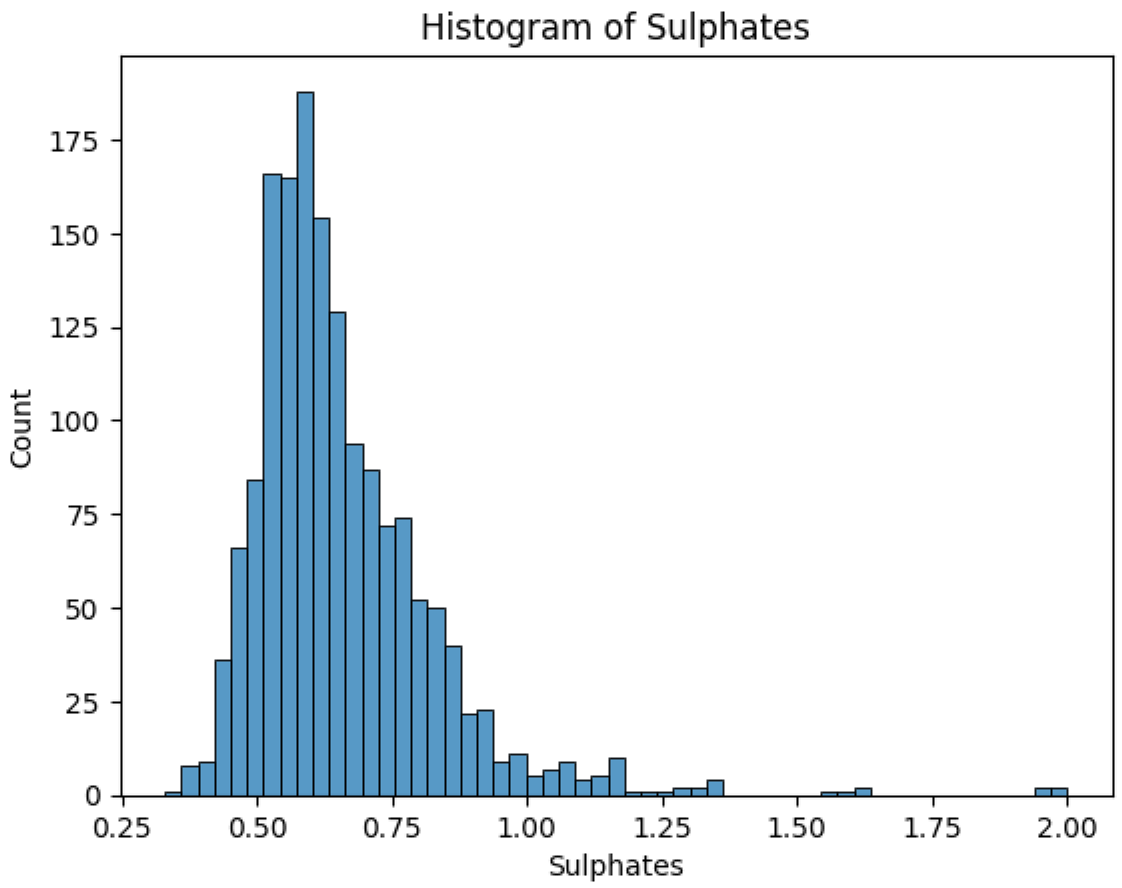
In [14]:
```
sns.histplot(df['density'])
plt.xlabel('Density')
plt.title('Histogram of Density')
plt.show()
```



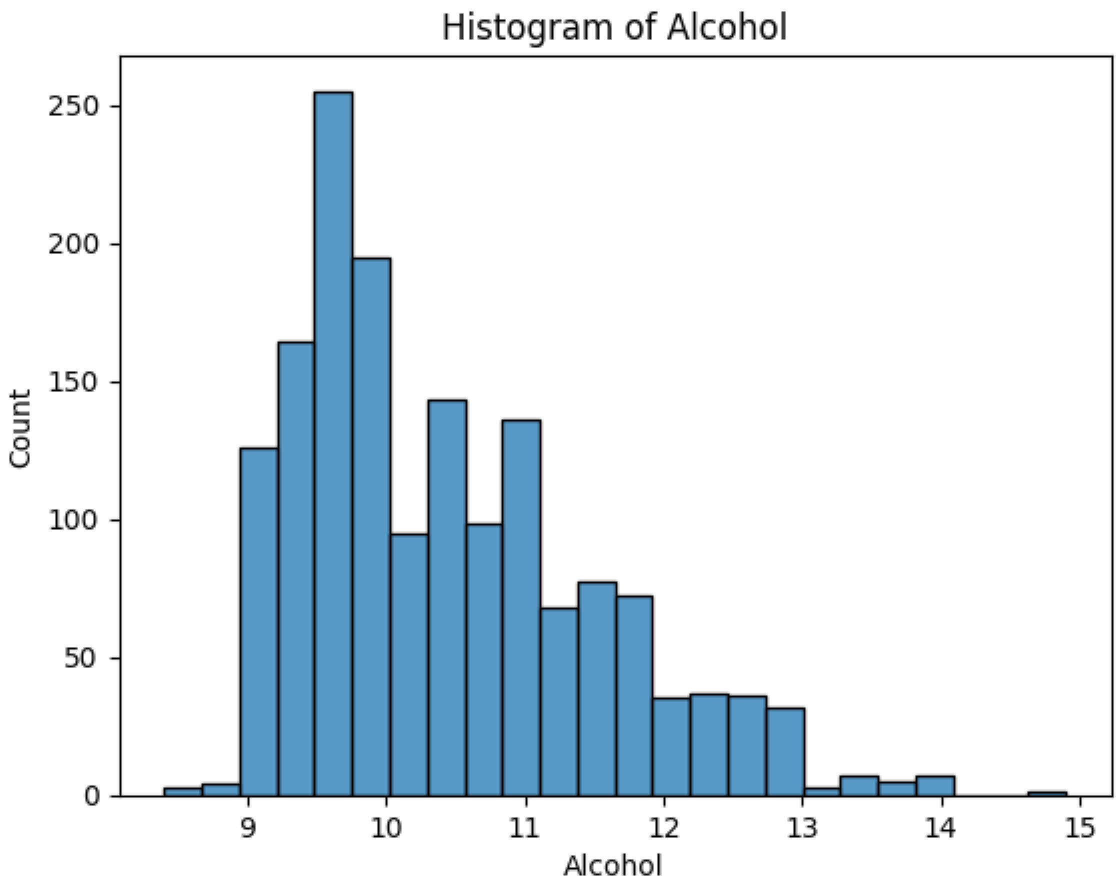
In [15]:
```
sns.histplot(df['pH'])
plt.xlabel('pH')
plt.title('Histogram of pH')
plt.show()
```



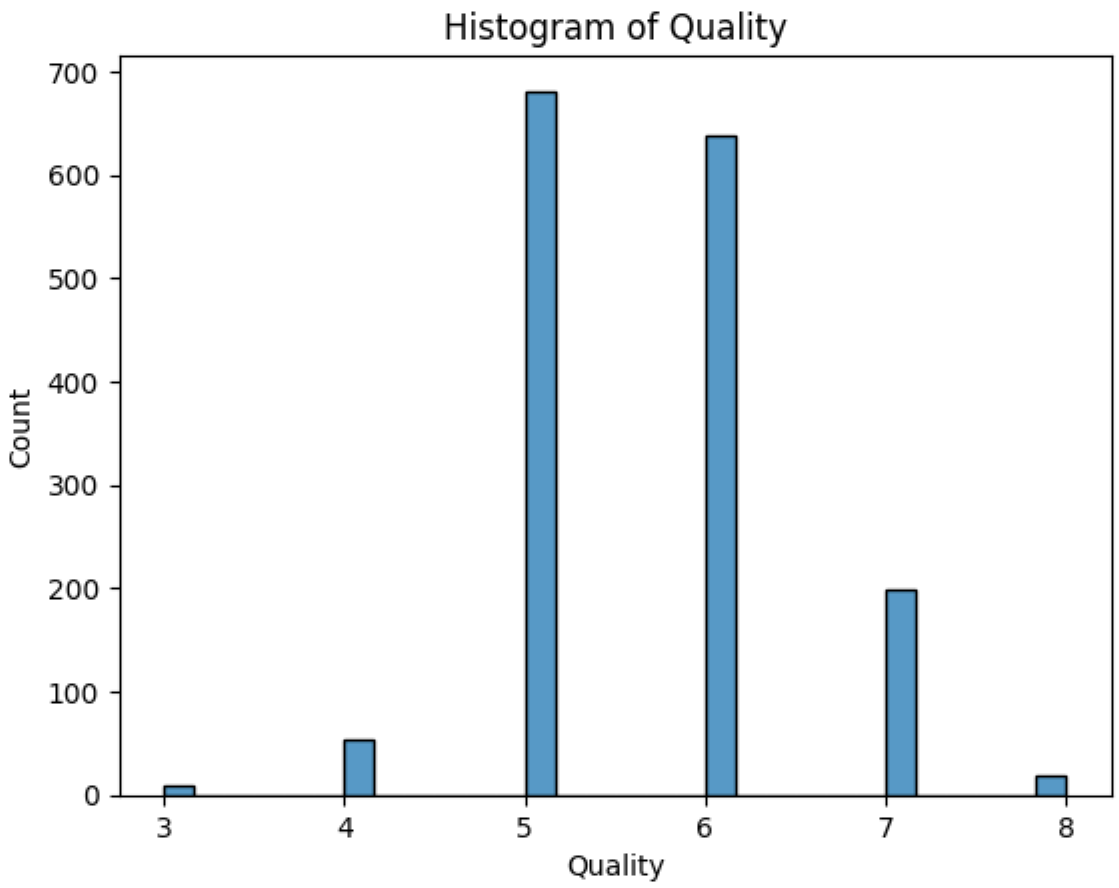
In [16]:
```
sns.histplot(df['sulphates'])
plt.xlabel('Sulphates')
plt.title('Histogram of Sulphates')
plt.show()
```



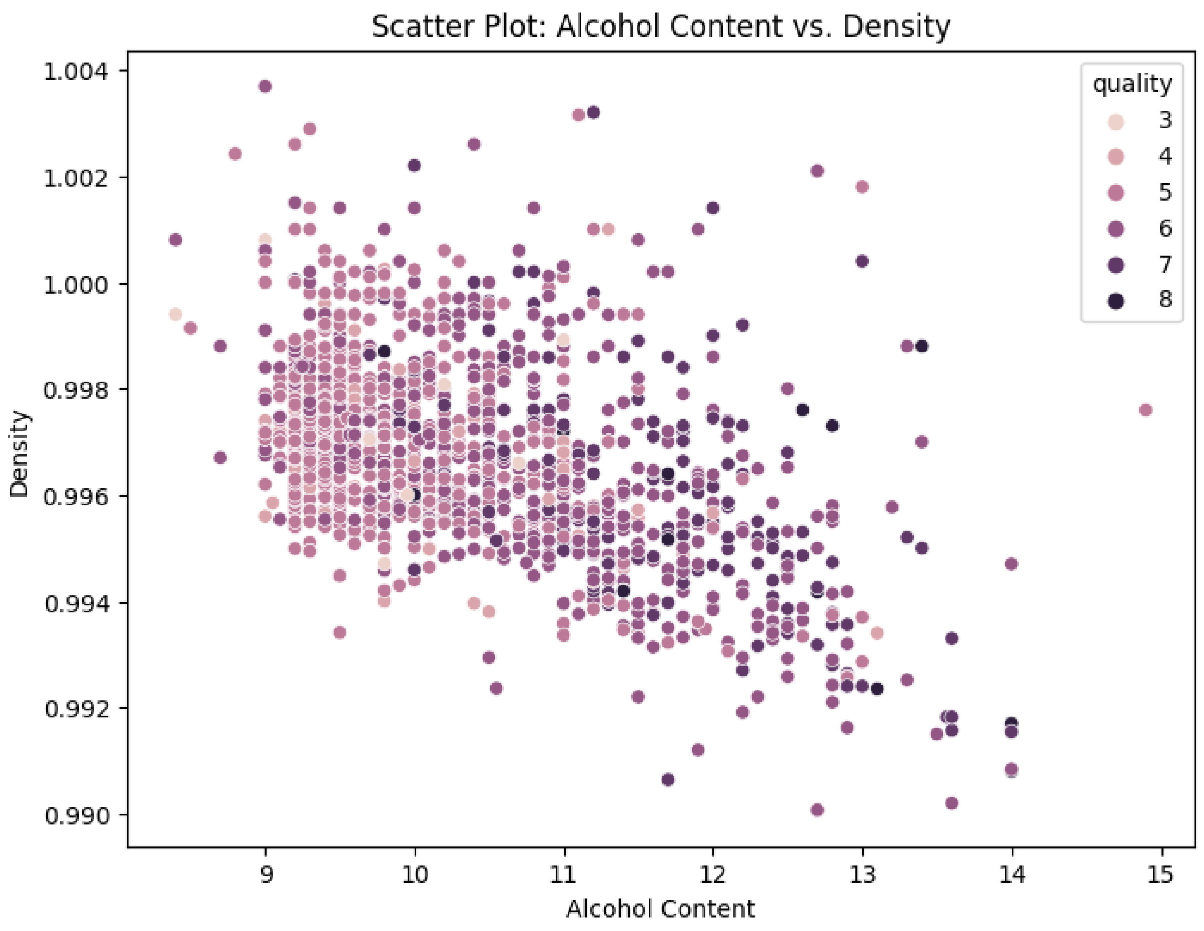
In [17]:
```
sns.histplot(df['alcohol'])
plt.xlabel('Alcohol')
plt.title('Histogram of Alcohol')
plt.show()
```



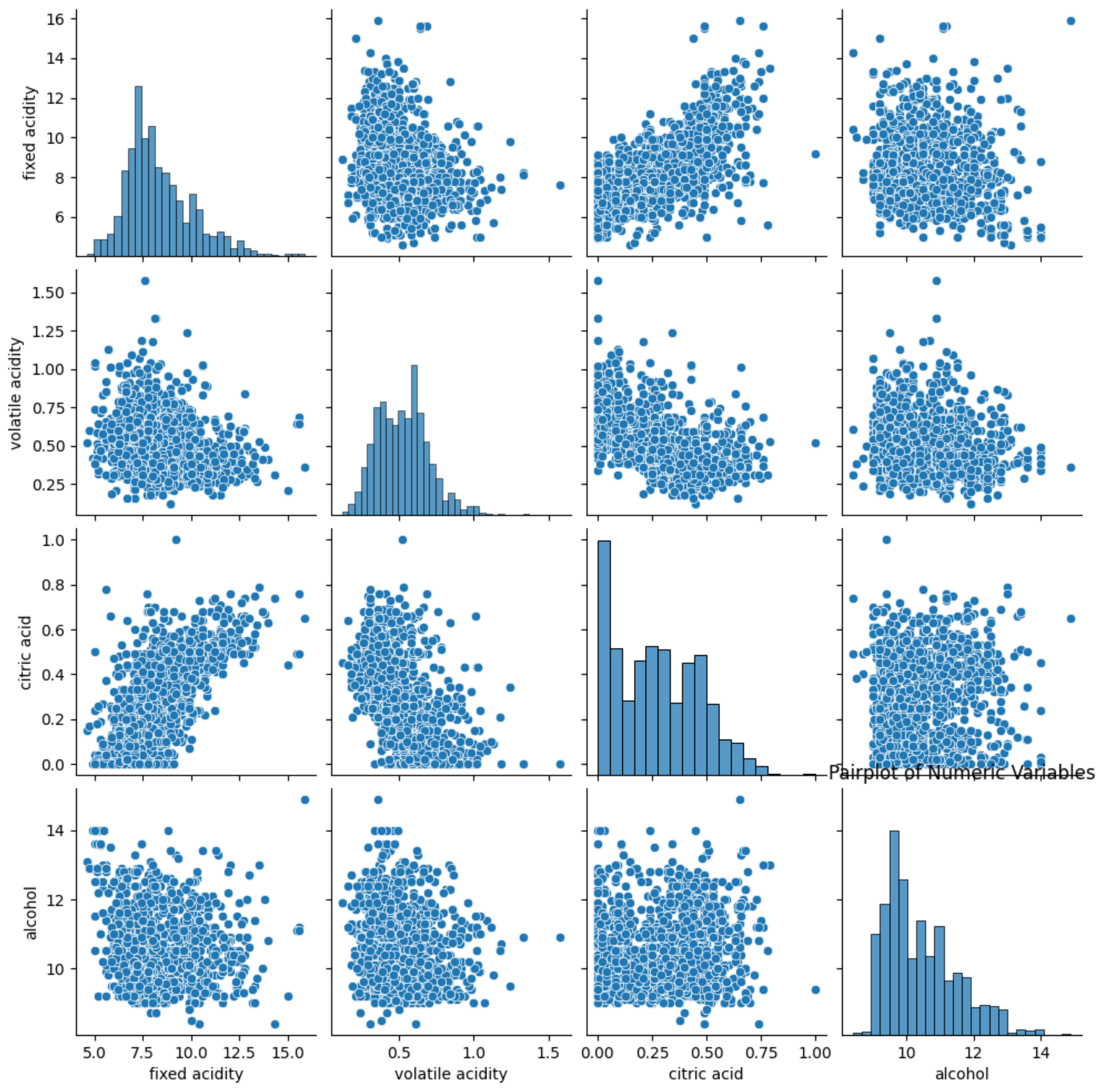
In [18]:
```
sns.histplot(df['quality'])
plt.xlabel('Quality')
plt.title('Histogram of Quality')
plt.show()
```



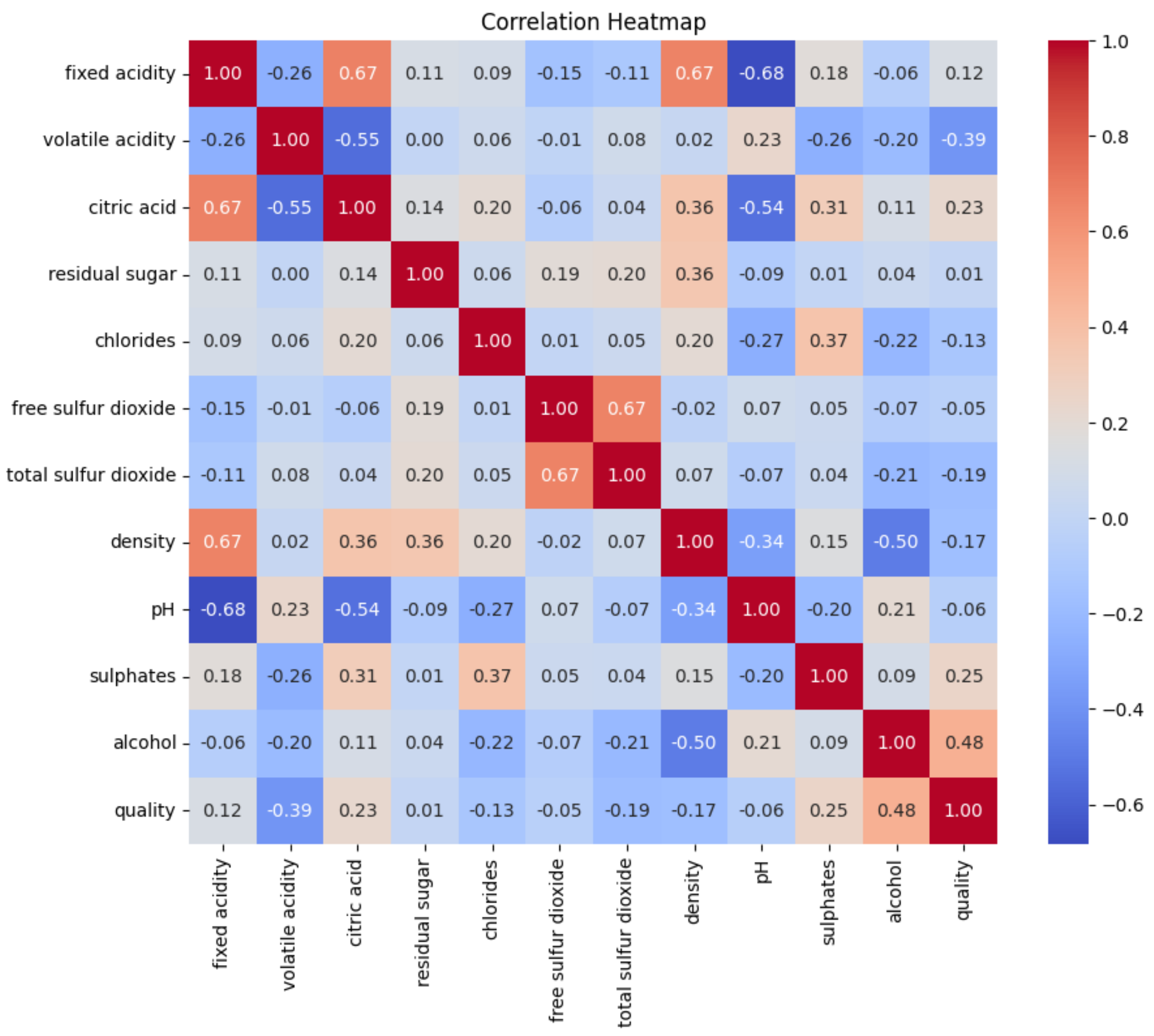
```
In [19]: plt.figure(figsize=(8, 6))
         sns.scatterplot(x='alcohol', y='density', hue='quality', data=df)
         plt.xlabel('Alcohol Content')
         plt.ylabel('Density')
         plt.title('Scatter Plot: Alcohol Content vs. Density')
         plt.show()
```



In [20]:
```
sns.pairplot(df[['fixed acidity', 'volatile acidity', 'citric acid', 'alcohol']])
plt.title('Pairplot of Numeric Variables')
plt.show()
```

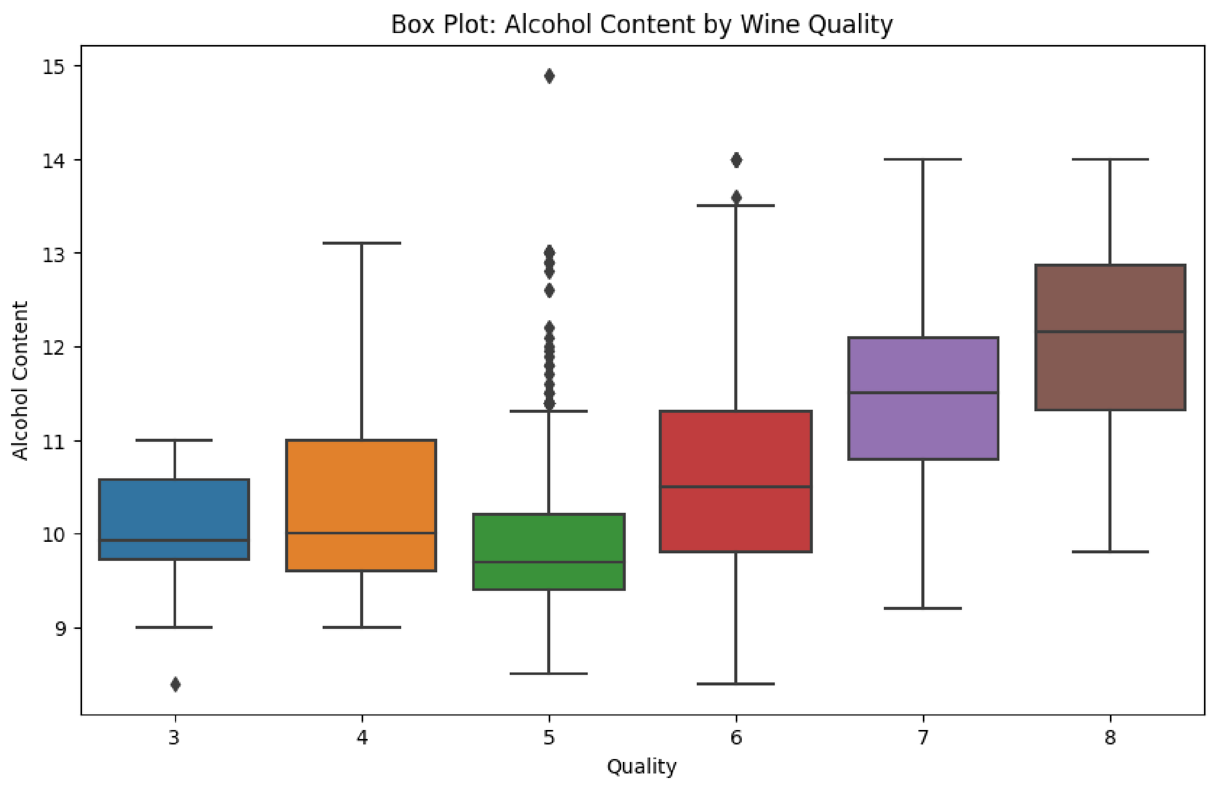


In [21]:
```
correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```

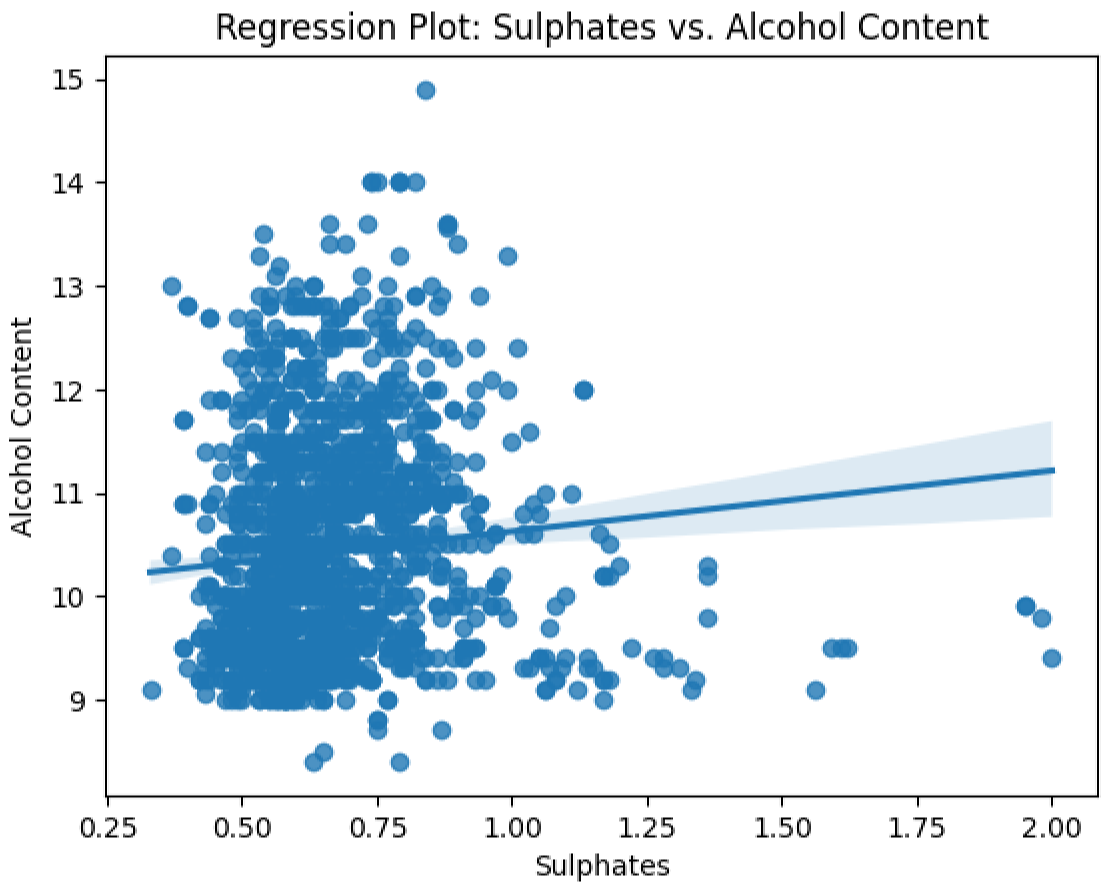


	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1.00	-0.26	0.67	0.11	0.09	-0.15	-0.11	0.67	-0.68	0.18	-0.06	0.12
volatile acidity	-0.26	1.00	-0.55	0.00	0.06	-0.01	0.08	0.02	0.23	-0.26	-0.20	-0.39
citric acid	0.67	-0.55	1.00	0.14	0.20	-0.06	0.04	0.36	-0.54	0.31	0.11	0.23
residual sugar	0.11	0.00	0.14	1.00	0.06	0.19	0.20	0.36	-0.09	0.01	0.04	0.01
chlorides	0.09	0.06	0.20	0.06	1.00	0.01	0.05	0.20	-0.27	0.37	-0.22	-0.13
free sulfur dioxide	-0.15	-0.01	-0.06	0.19	0.01	1.00	0.67	-0.02	0.07	0.05	-0.07	-0.05
total sulfur dioxide	-0.11	0.08	0.04	0.20	0.05	0.67	1.00	0.07	-0.07	0.04	-0.21	-0.19
density	0.67	0.02	0.36	0.36	0.20	-0.02	0.07	1.00	-0.34	0.15	-0.50	-0.17
pH	-0.68	0.23	-0.54	-0.09	-0.27	0.07	-0.07	-0.34	1.00	-0.20	0.21	-0.06
sulphates	0.18	-0.26	0.31	0.01	0.37	0.05	0.04	0.15	-0.20	1.00	0.09	0.25
alcohol	-0.06	-0.20	0.11	0.04	-0.22	-0.07	-0.21	-0.50	0.21	0.09	1.00	0.48
quality	0.12	-0.39	0.23	0.01	-0.13	-0.05	-0.19	-0.17	-0.06	0.25	0.48	1.00

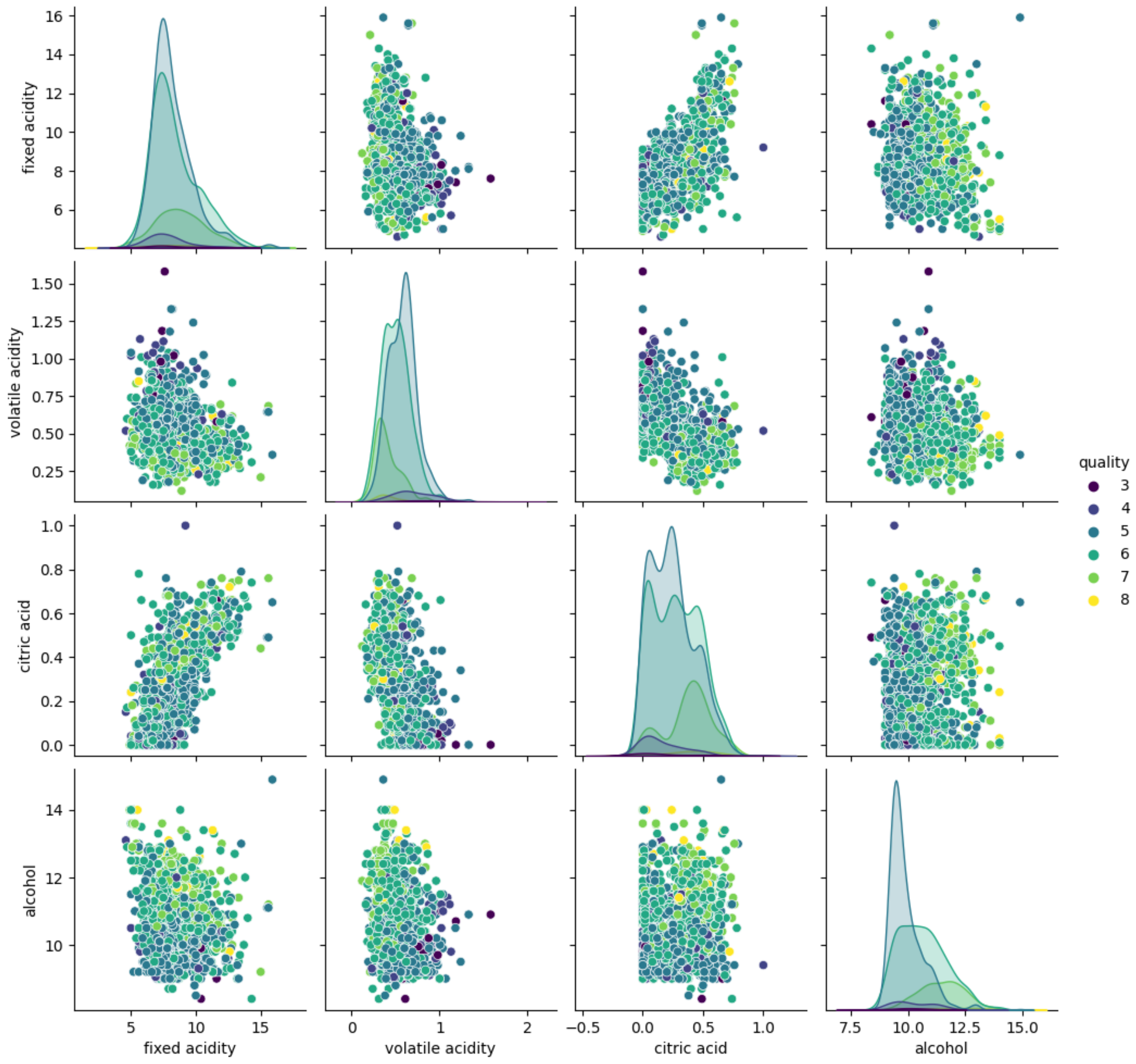
```
In [22]: plt.figure(figsize=(10, 6))
         sns.boxplot(x='quality', y='alcohol', data=df)
         plt.xlabel('Quality')
         plt.ylabel('Alcohol Content')
         plt.title('Box Plot: Alcohol Content by Wine Quality')
         plt.show()
```



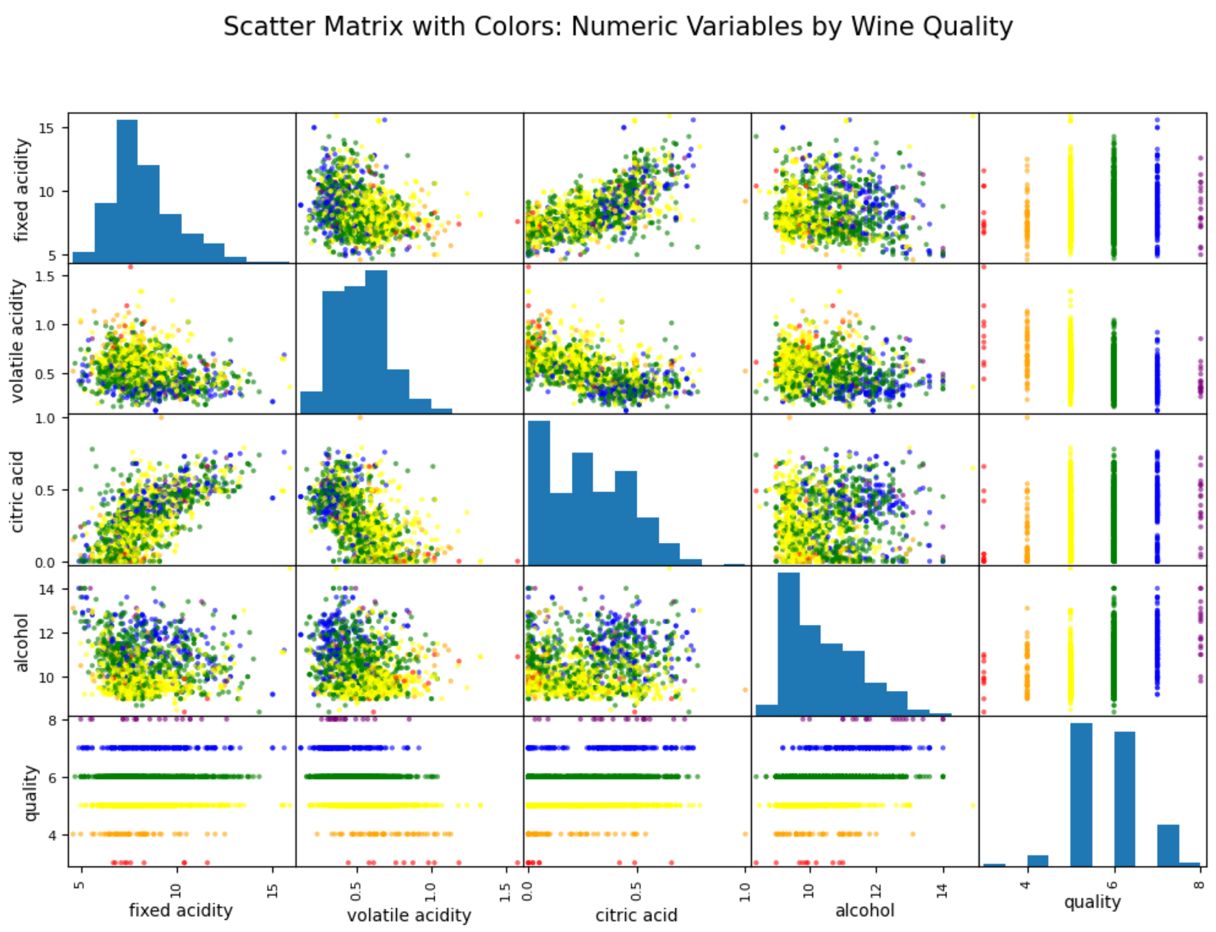
In [23]:
```
sns.regplot(x='sulphates', y='alcohol', data=df)
plt.xlabel('Sulphates')
plt.ylabel('Alcohol Content')
plt.title('Regression Plot: Sulphates vs. Alcohol Content')
plt.show()
```



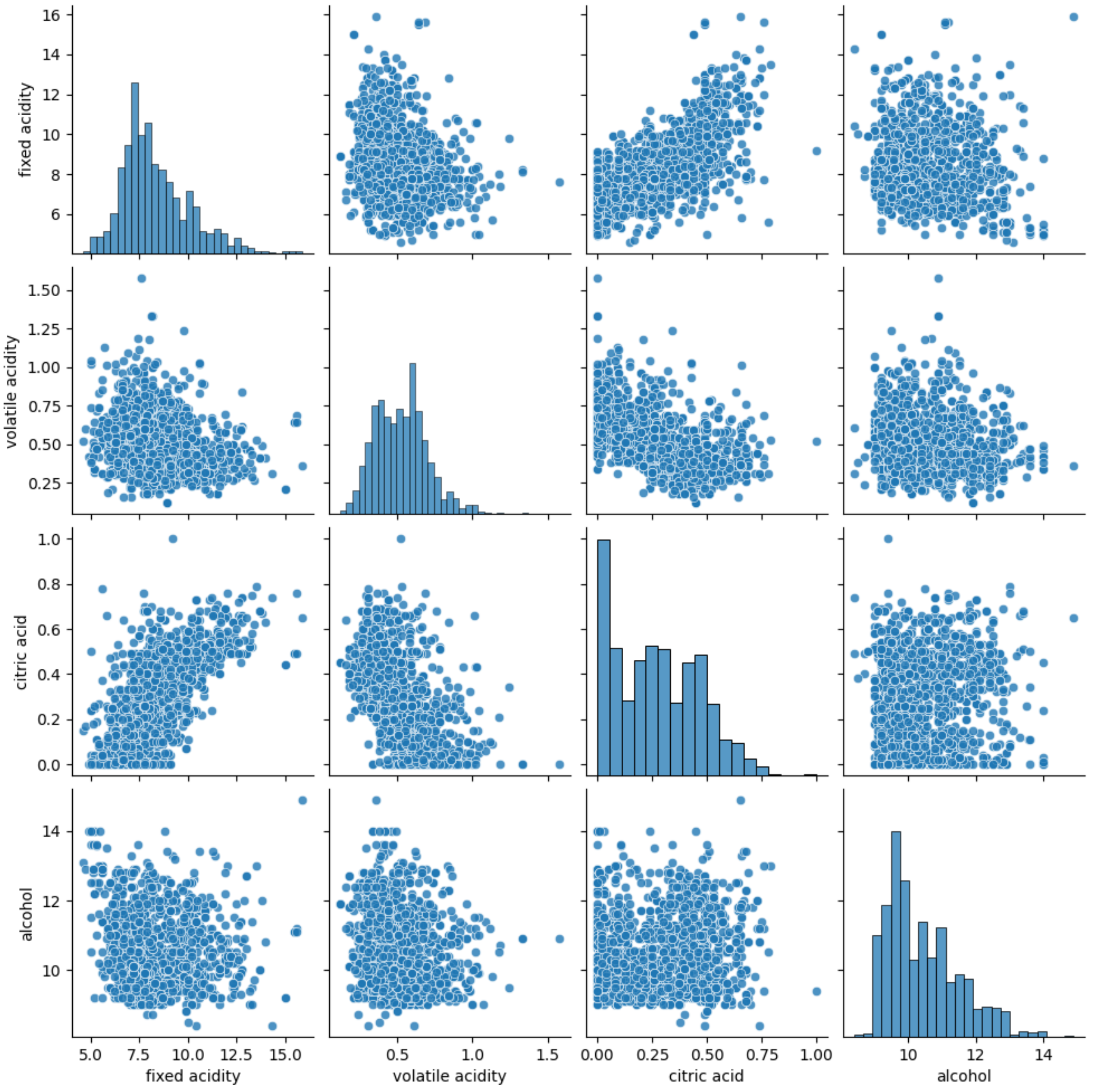
In [24]:
```
sns.pairplot(df[['fixed acidity', 'volatile acidity', 'citric acid', 'alcohol', 'qu
# plt.title('Pairplot with Hue: Numeric Variables by Wine Quality')
plt.show()
```



In [25]:
```
from pandas.plotting import scatter_matrix

numeric_vars = df[['fixed acidity', 'volatile acidity', 'citric acid', 'alcohol']]
numeric_vars['quality'] = df['quality']

colors = {3: 'red', 4: 'orange', 5: 'yellow', 6: 'green', 7: 'blue', 8: 'purple'}

scatter_matrix(numeric_vars, c=numeric_vars['quality'].map(colors), figsize=(12, 8)
plt.suptitle('Scatter Matrix with Colors: Numeric Variables by Wine Quality', size=
plt.show()
```



In [26]:
```
correlation_matrix = df[['fixed acidity', 'volatile acidity', 'citric acid', 'alcoh

sns.pairplot(df[['fixed acidity', 'volatile acidity', 'citric acid', 'alcohol']], p
# plt.title('Pairplot with Correlation Colors: Numeric Variables')
plt.show()
```



In [27]:
```
df.head()
```

Out[27]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	a
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	

In [28]:
```
df['quality'] = df['quality'] >= 7
```

In [29]:
```
df.head()
```

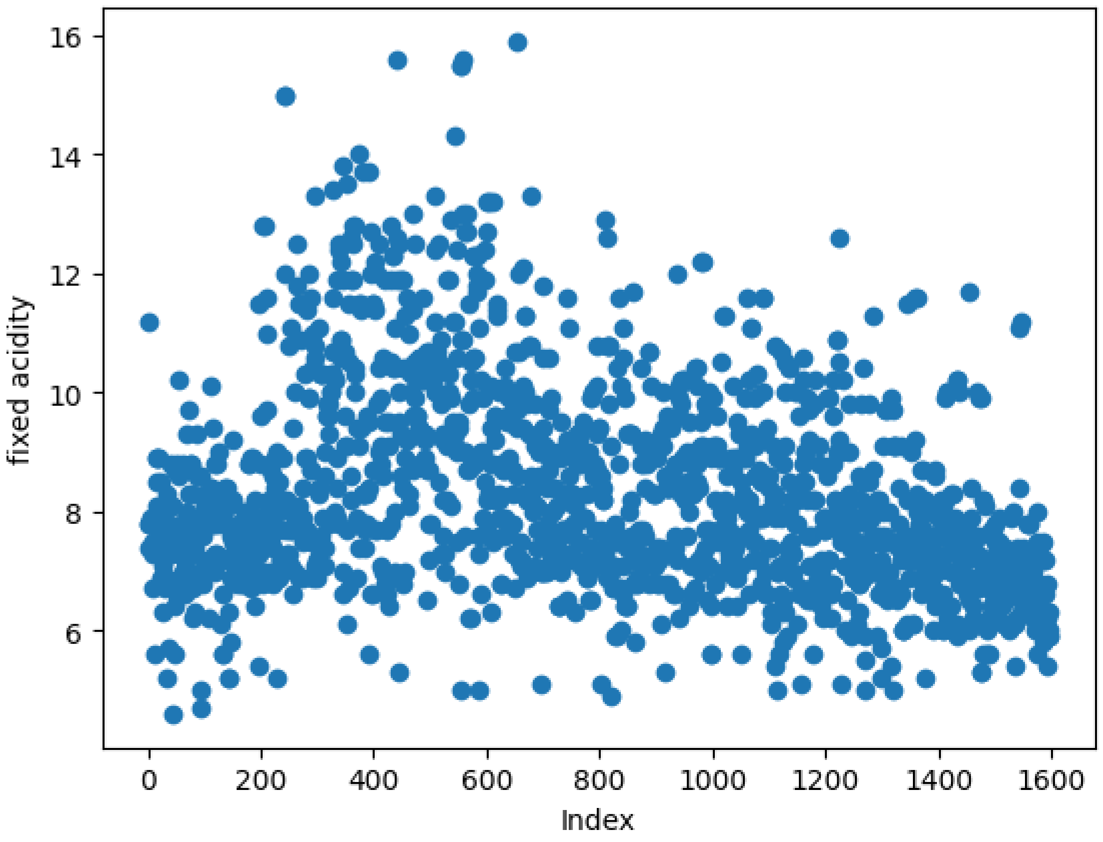
Out[29]:

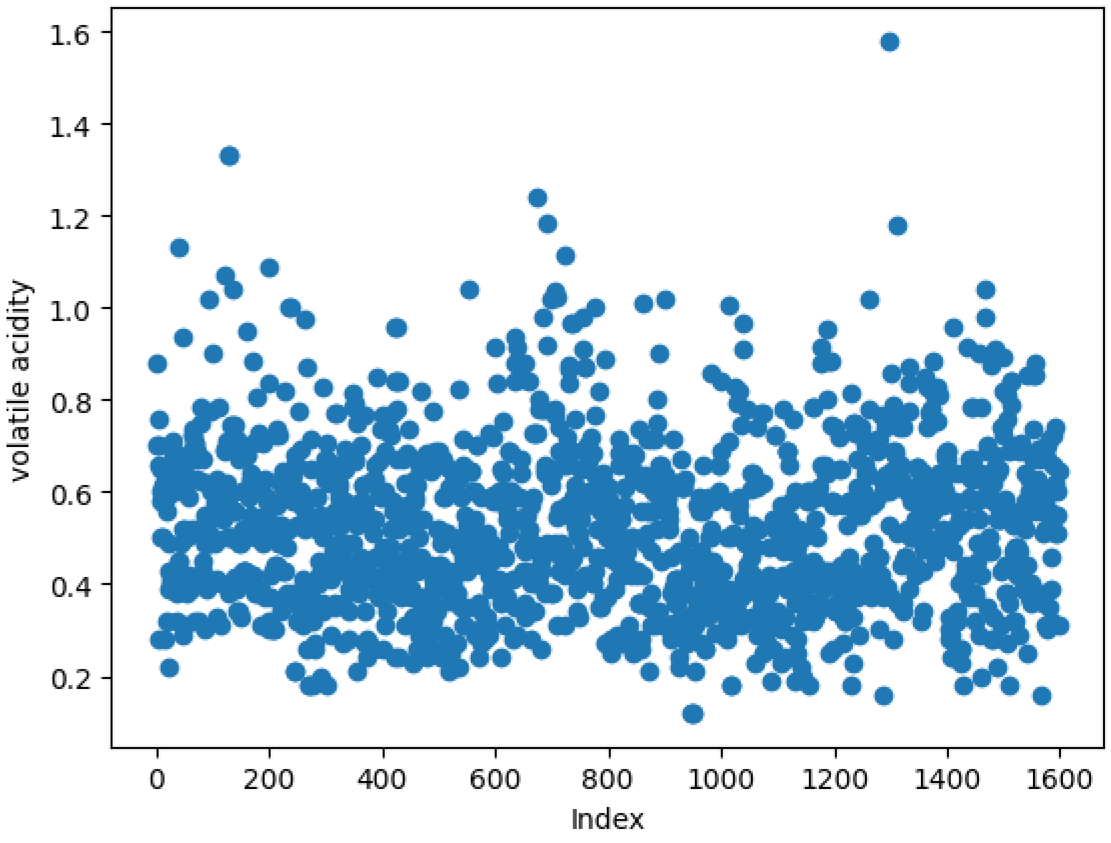
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	a
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	

Checking for Outliers

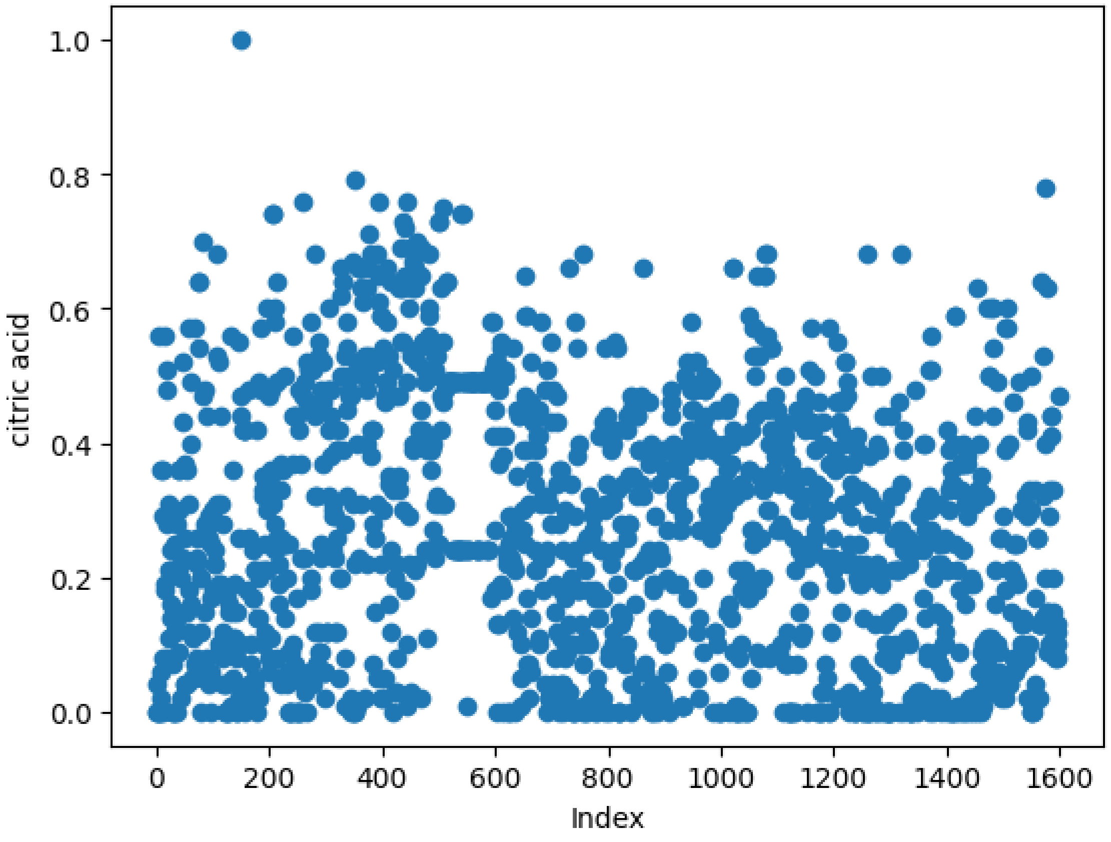
In [30]:
```
cols = df.columns
```

In [31]:
```
for col in cols:
    plt.scatter(df.index, df[col])
    plt.xlabel('Index')
    plt.ylabel(col)
    plt.show()
```

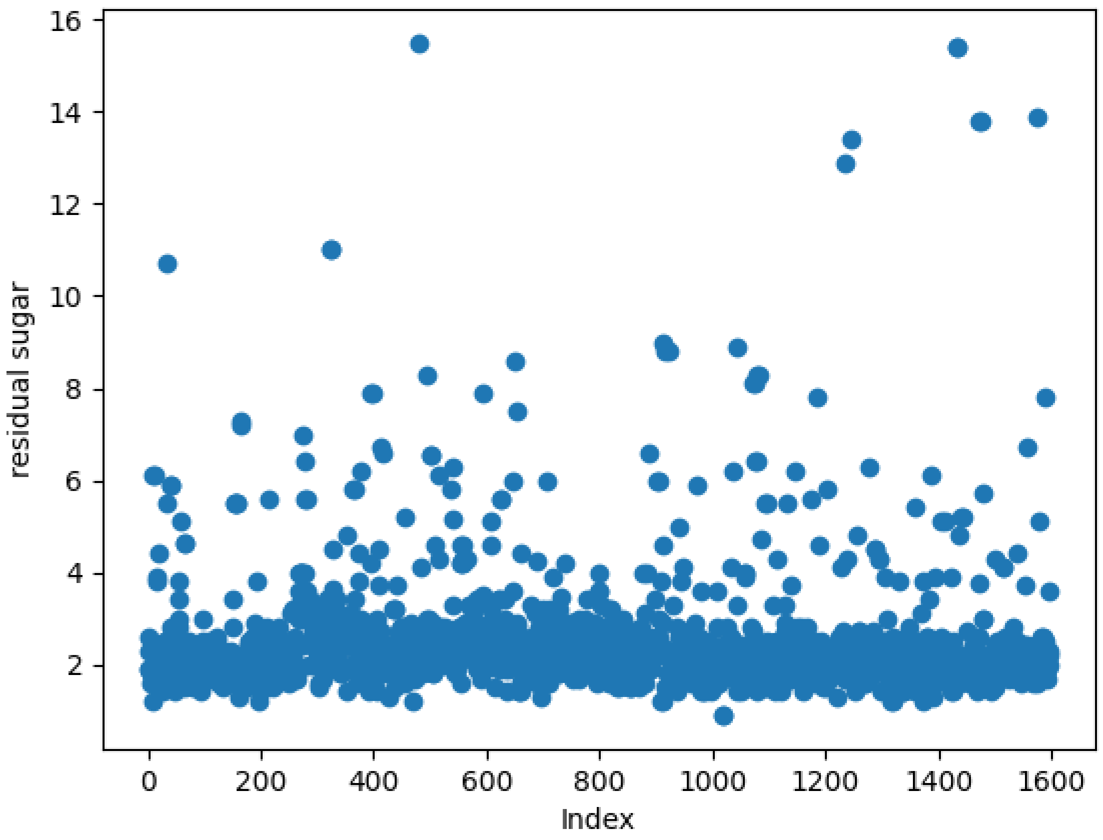


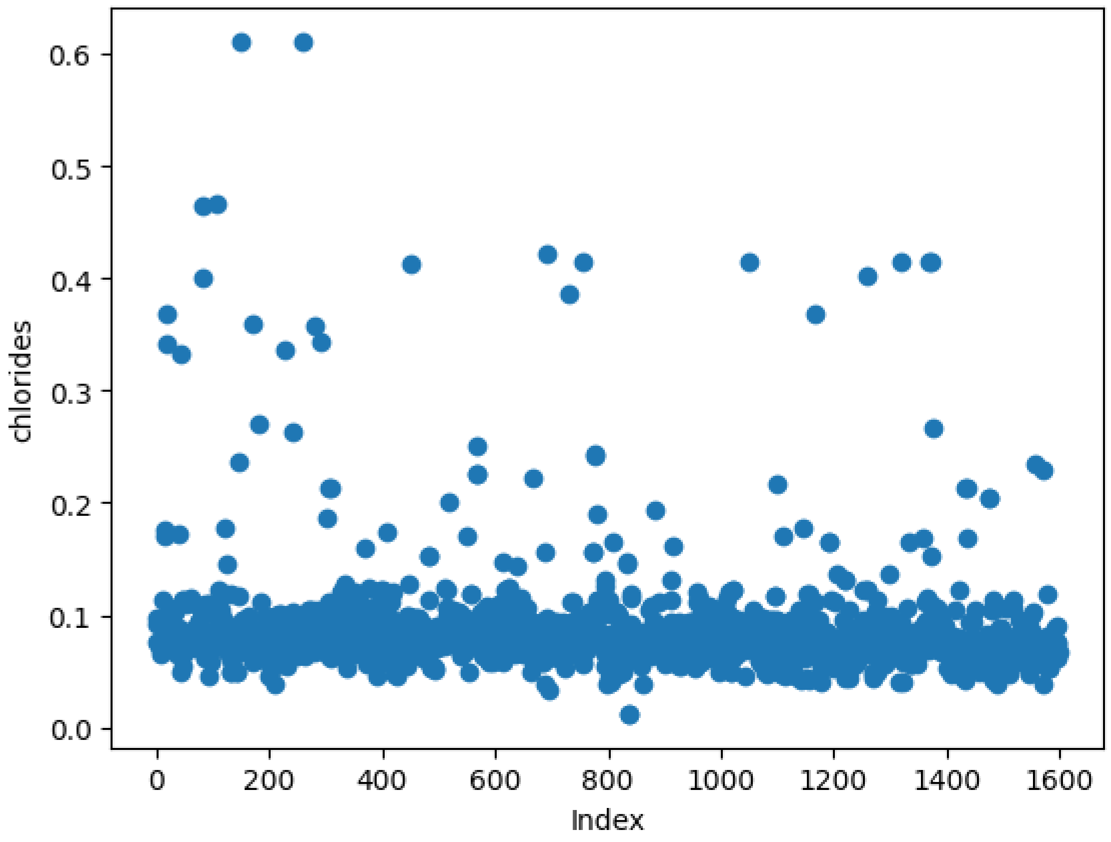


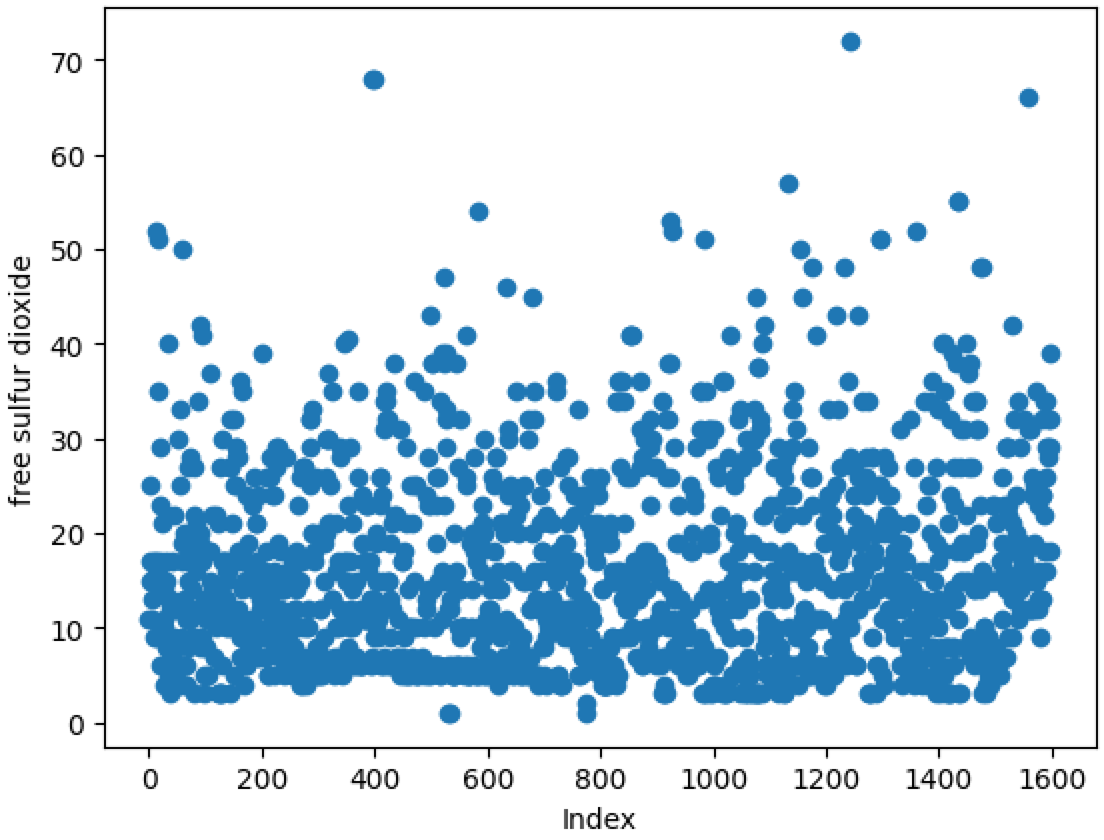
volatile acidity
Index
0 200 400 600 800 1000 1200 1400 1600
0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6

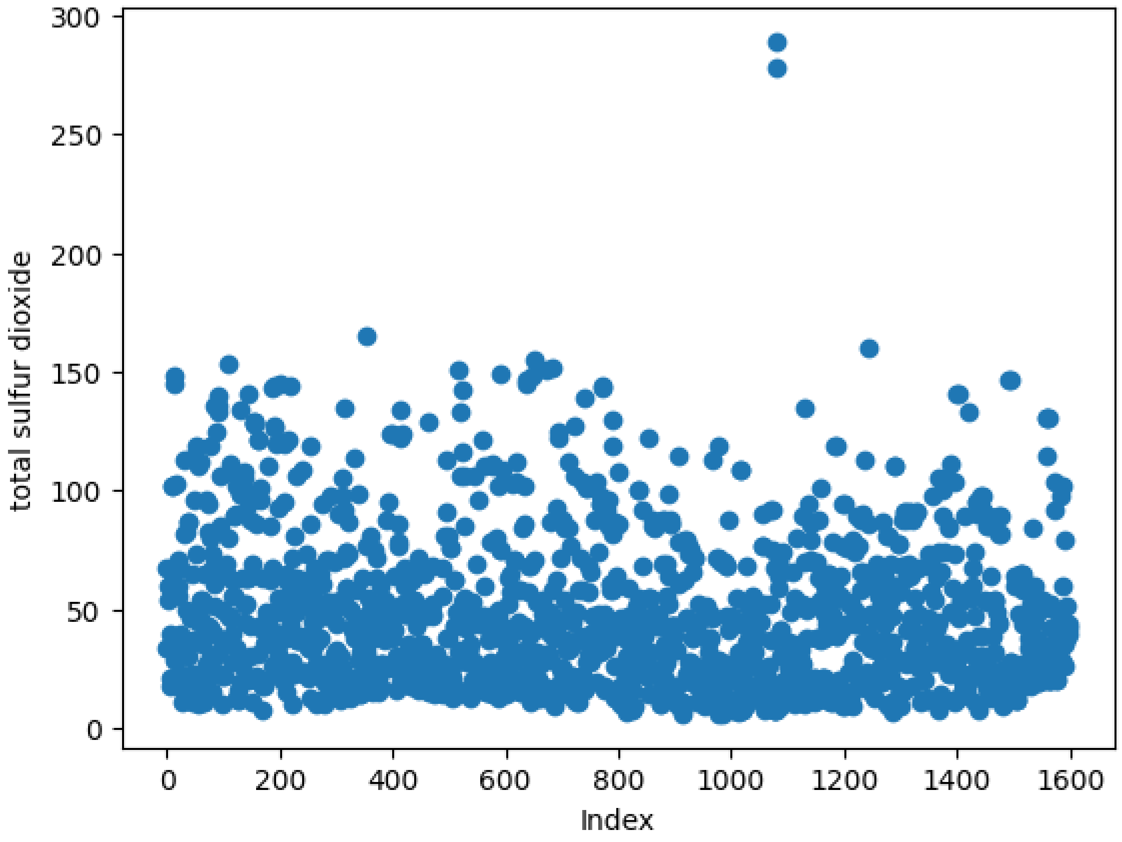


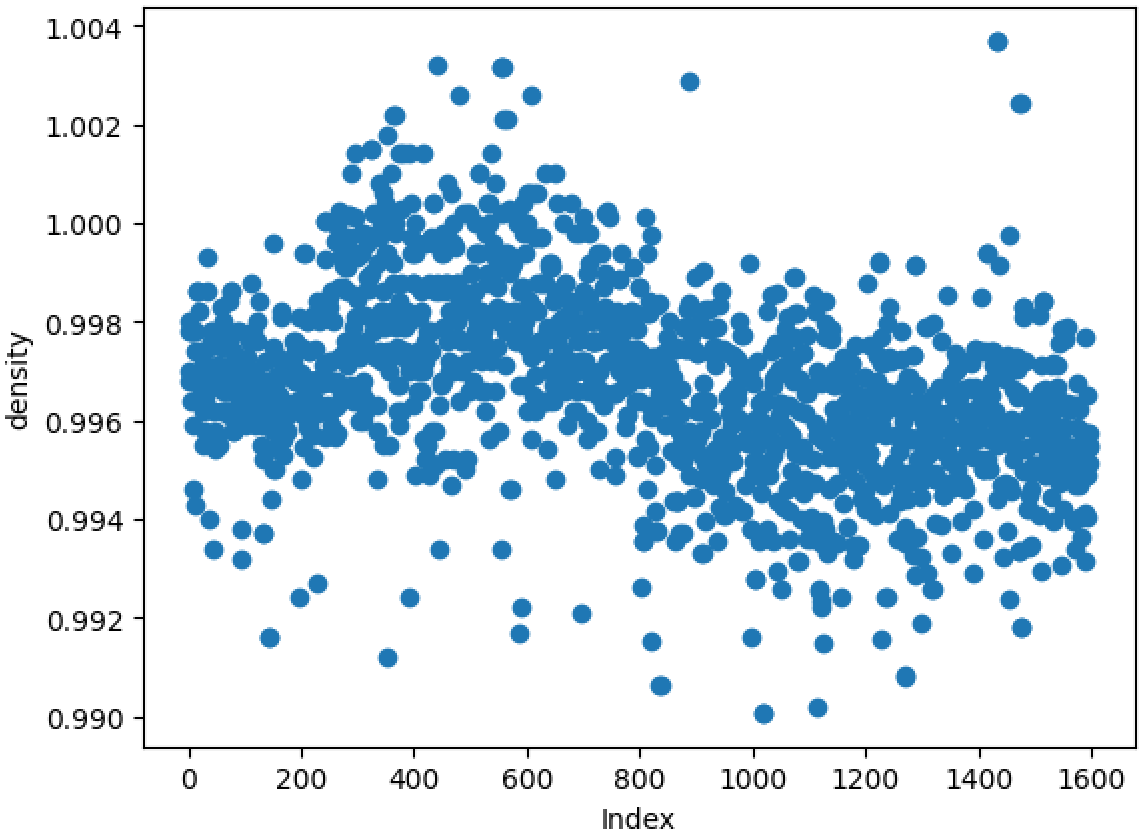
citric acid
Index
0 200 400 600 800 1000 1200 1400 1600
0.0 0.2 0.4 0.6 0.8 1.0

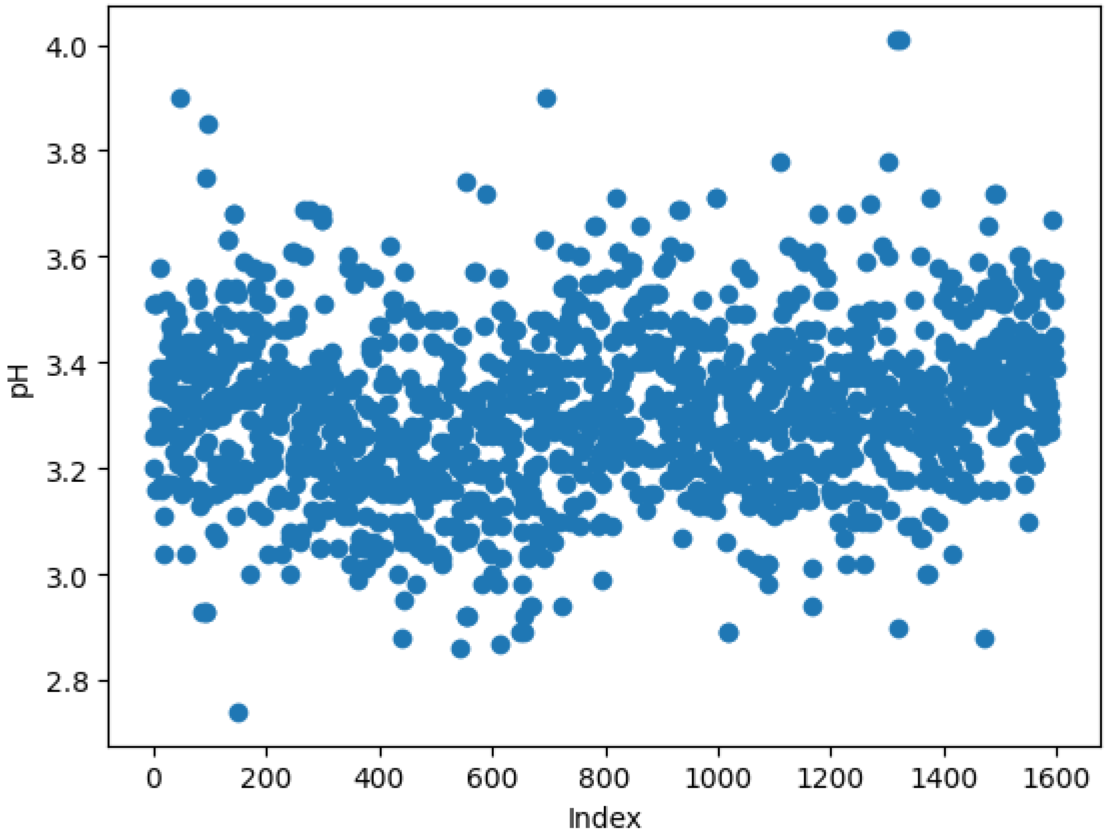

residual sugar
Index
16
14
12
10
8
6
4
2
0
200
400
600
800
1000
1200
1400
1600

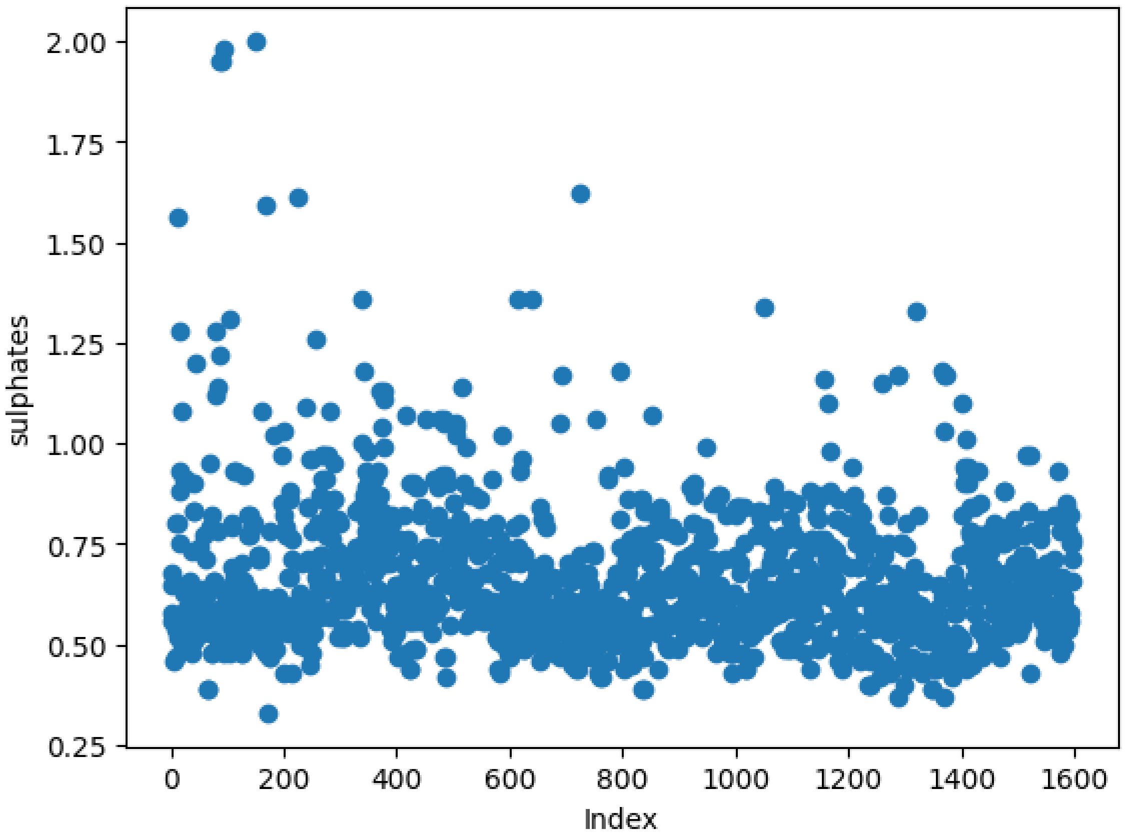

chlorides
Index
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
200
400
600
800
1000
1200
1400
1600

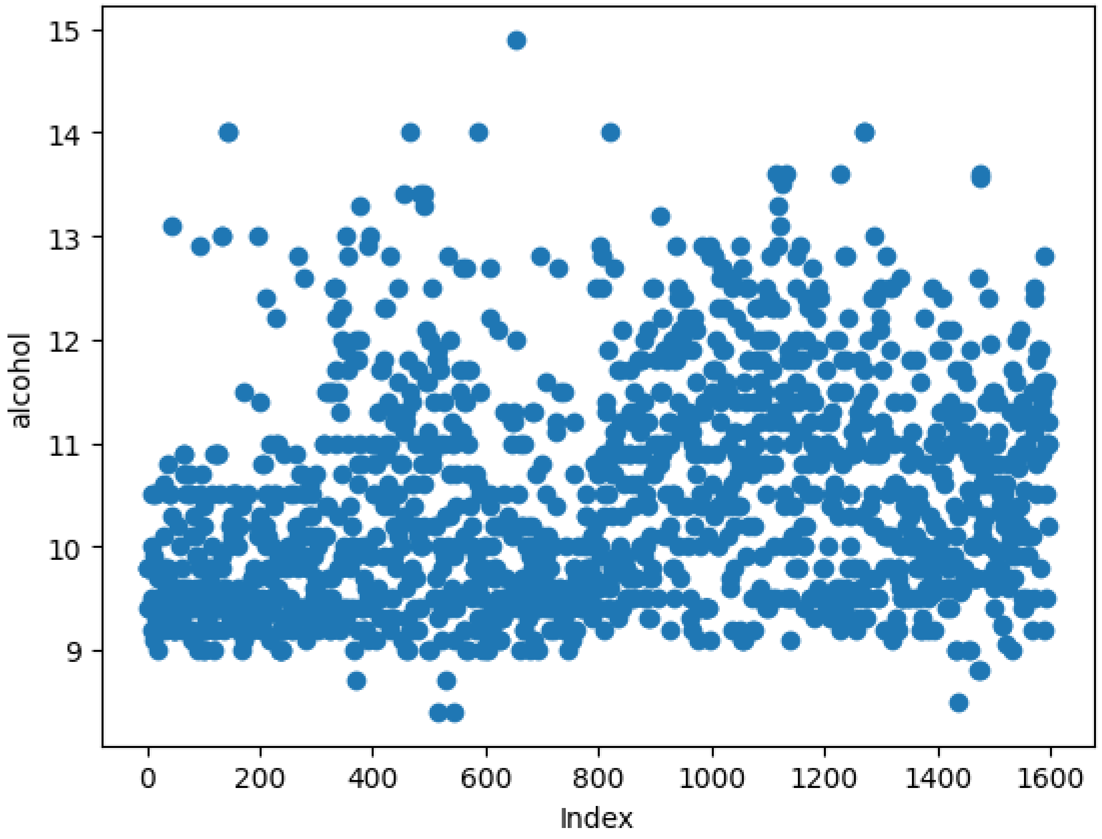

free sulfur dioxide
Index

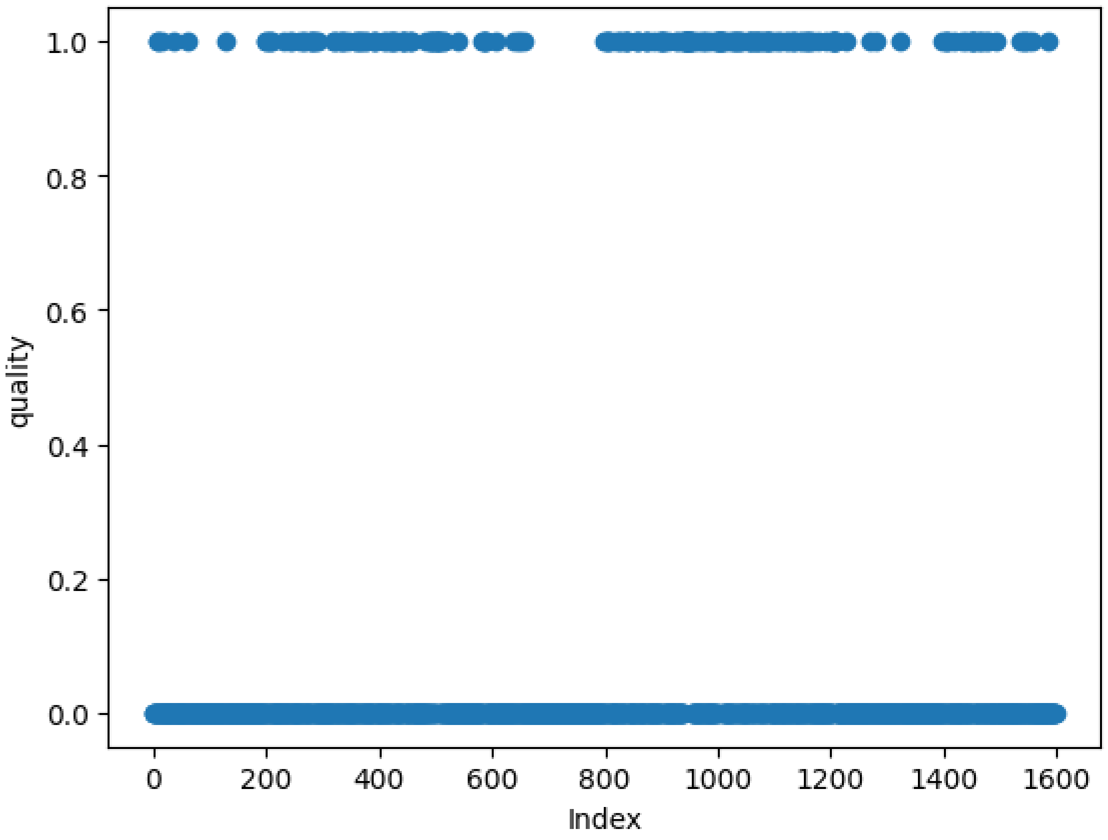

total sulfur dioxide
Index


density
Index
1.004
1.002
1.000
0.998
0.996
0.994
0.992
0.990
0
200
400
600
800
1000
1200
1400
1600


pH
Index
4.0
3.8
3.6
3.4
3.2
3.0
2.8
0
200
400
600
800
1000
1200
1400
1600


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
sulphates
0
200
400
600
800
1000
1200
1400
1600
Index


15
14
13
12
11
10
9
alcohol
0
200
400
600
800
1000
1200
1400
1600
Index



```
In [32]: cols = list(cols)
```

```
In [33]: cols.remove('quality')
```

```
In [34]: cols
```

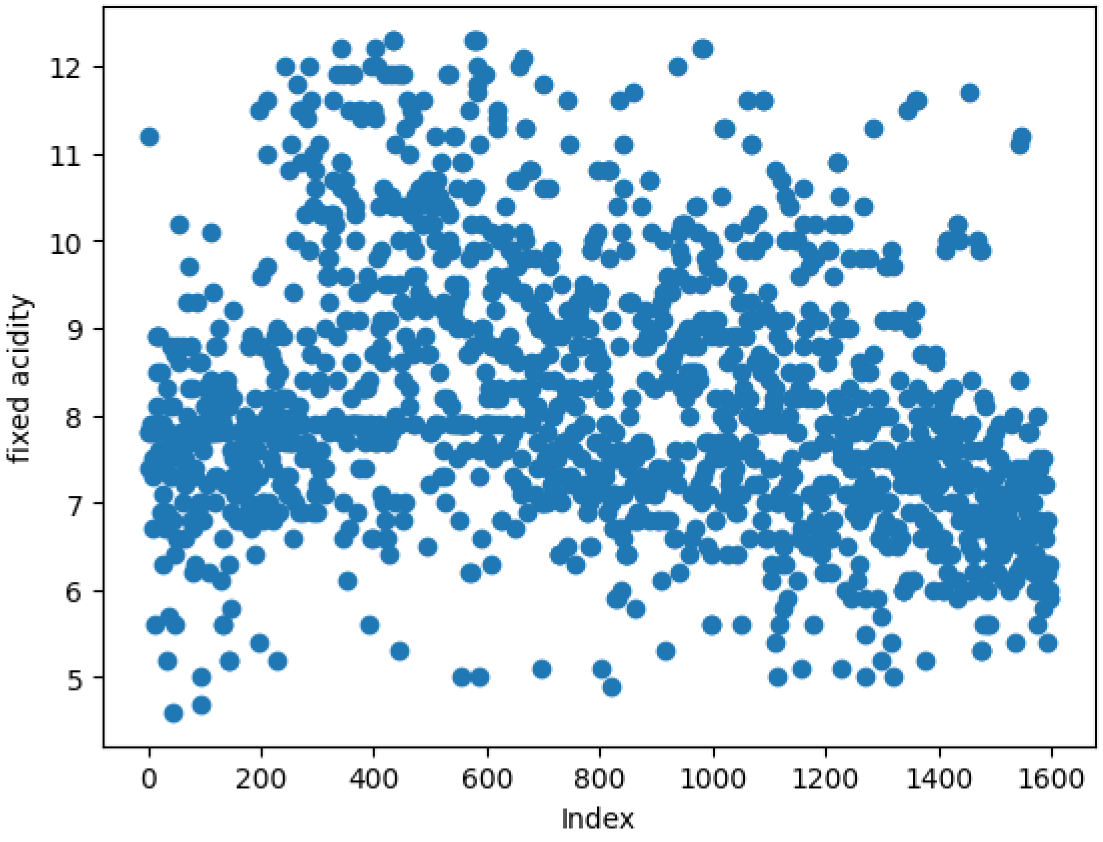
```
Out[34]: ['fixed acidity',
          'volatile acidity',
          'citric acid',
          'residual sugar',
          'chlorides',
          'free sulfur dioxide',
          'total sulfur dioxide',
          'density',
          'pH',
          'sulphates',
          'alcohol']
```

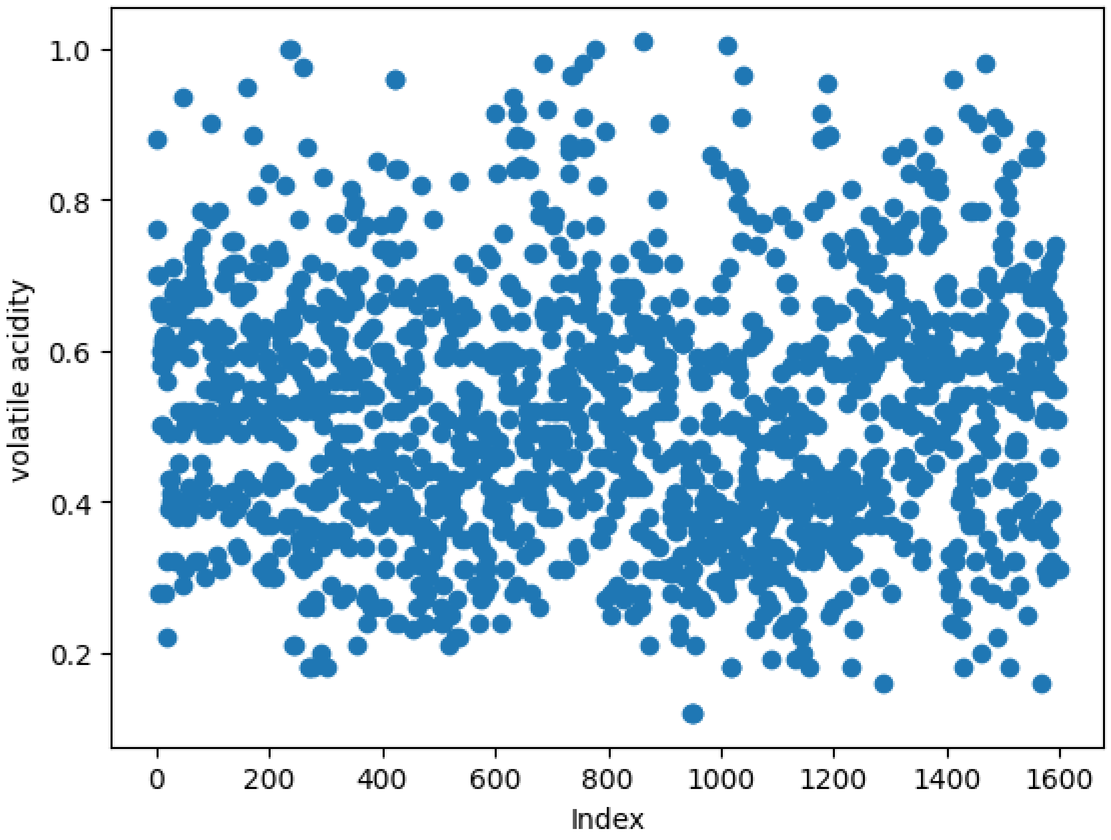
```
In [35]: df_clean = df.copy()
```

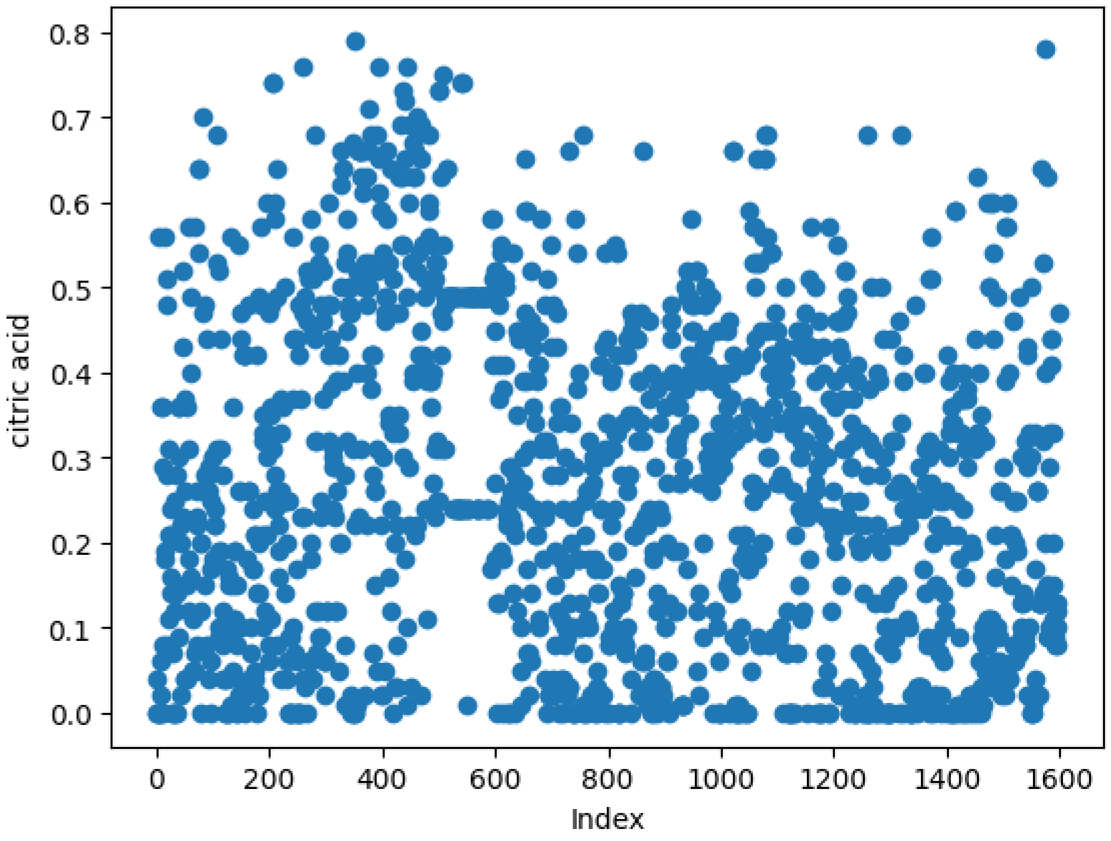
```
In [36]: for col in cols:
             Q1 = df_clean[col].quantile(0.25)
             Q3 = df_clean[col].quantile(0.75)
             IQR = Q3 - Q1

             lower_bound = Q1 - 1.5 * IQR
             upper_bound = Q3 + 1.5 * IQR
```

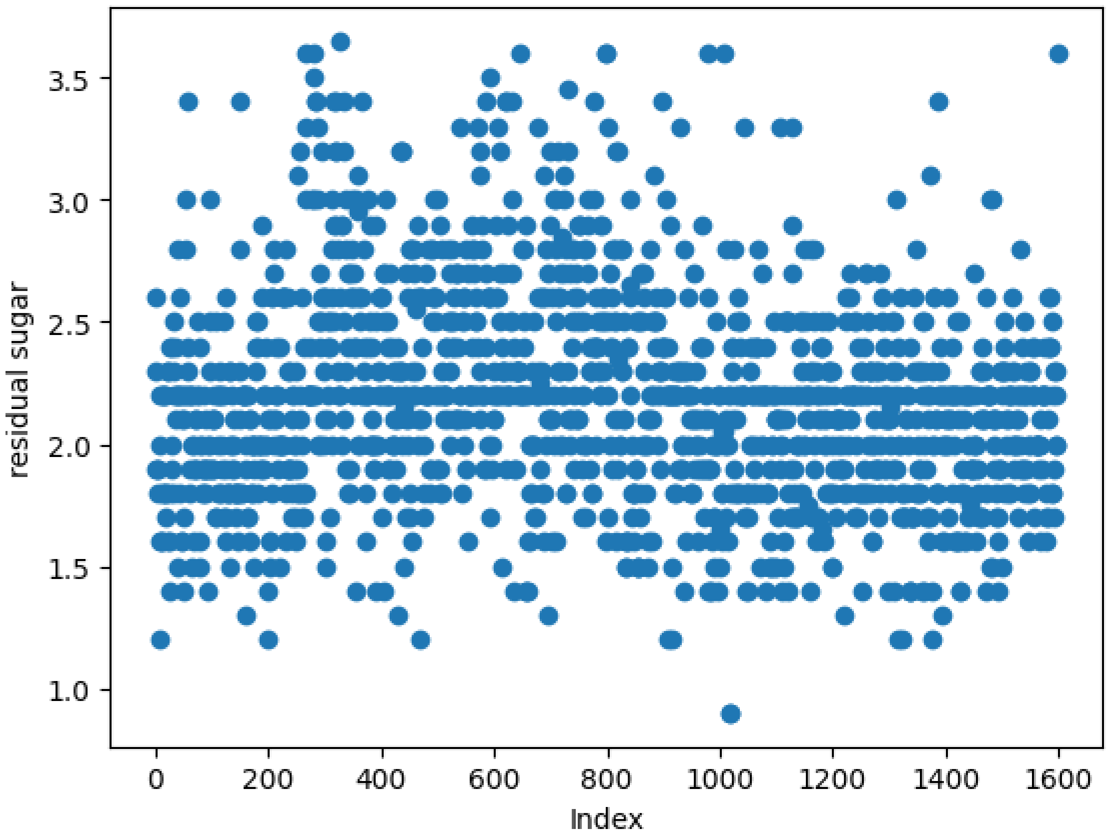
```
    median_value = df_clean[col].median()
    df_clean[col] = df_clean[col].apply(lambda x: x if lower_bound <= x <= upper_bo
```

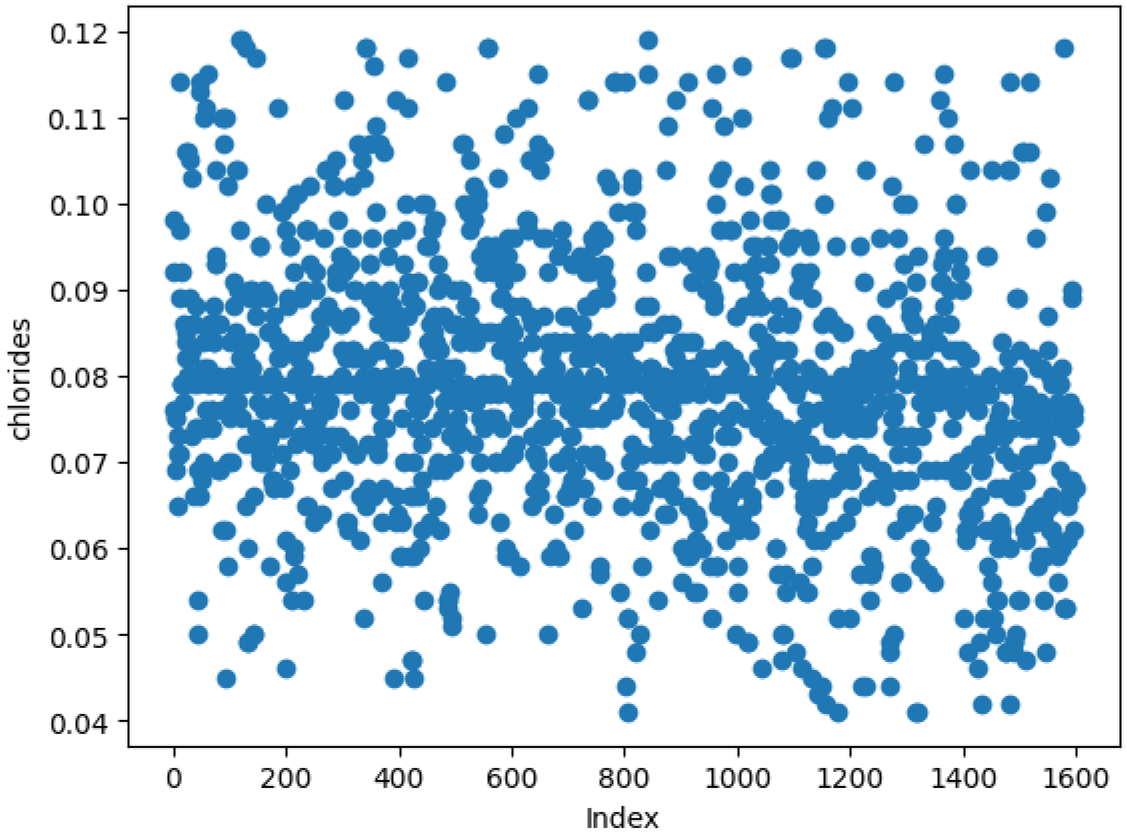
In [37]:
```
for col in cols:
    plt.scatter(df_clean.index, df_clean[col])
    plt.xlabel('Index')
    plt.ylabel(col)
    plt.show()
```

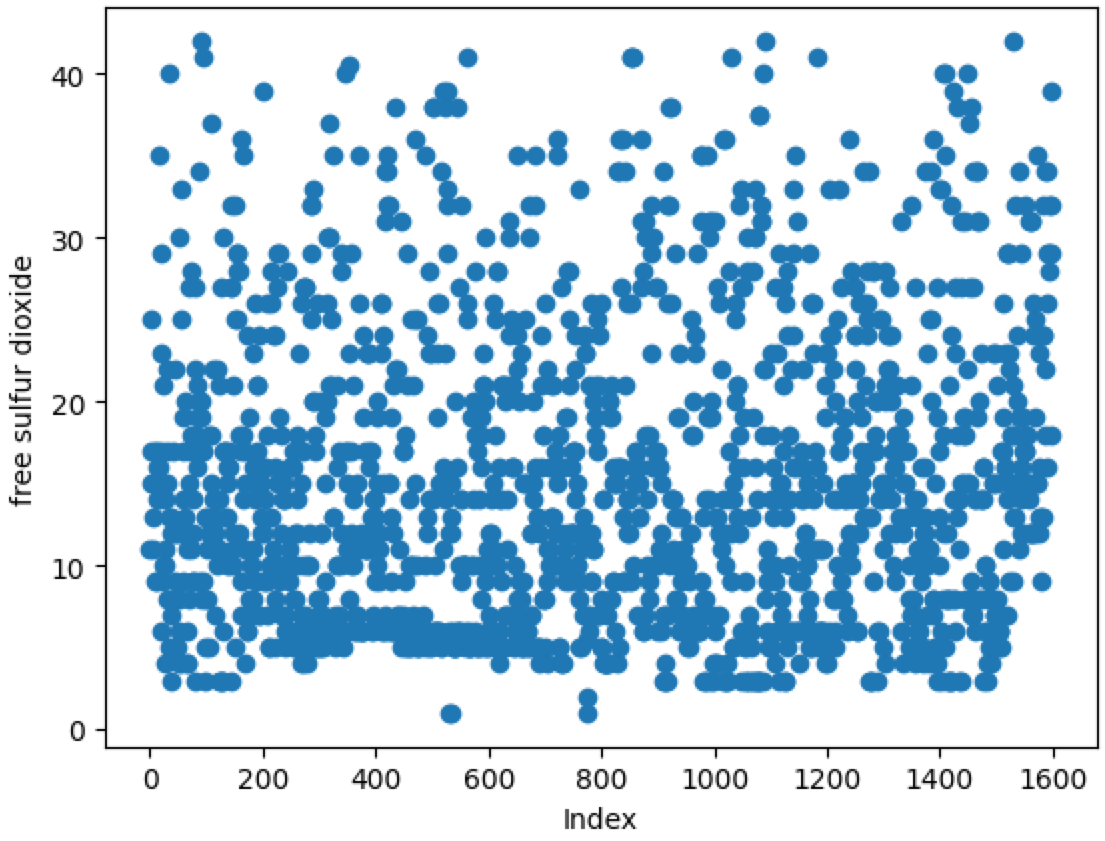


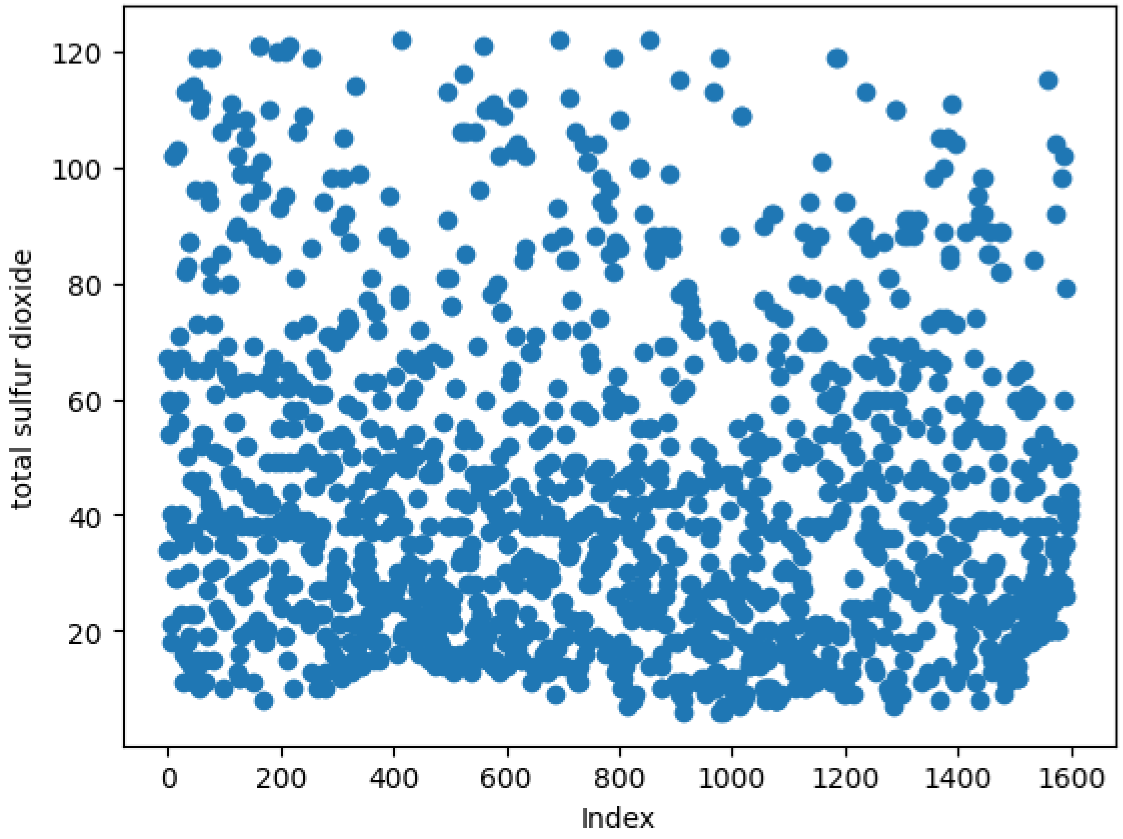

1.0
0.8
0.6
0.4
0.2
volatile acidity
0
200
400
600
800
1000
1200
1400
1600
Index

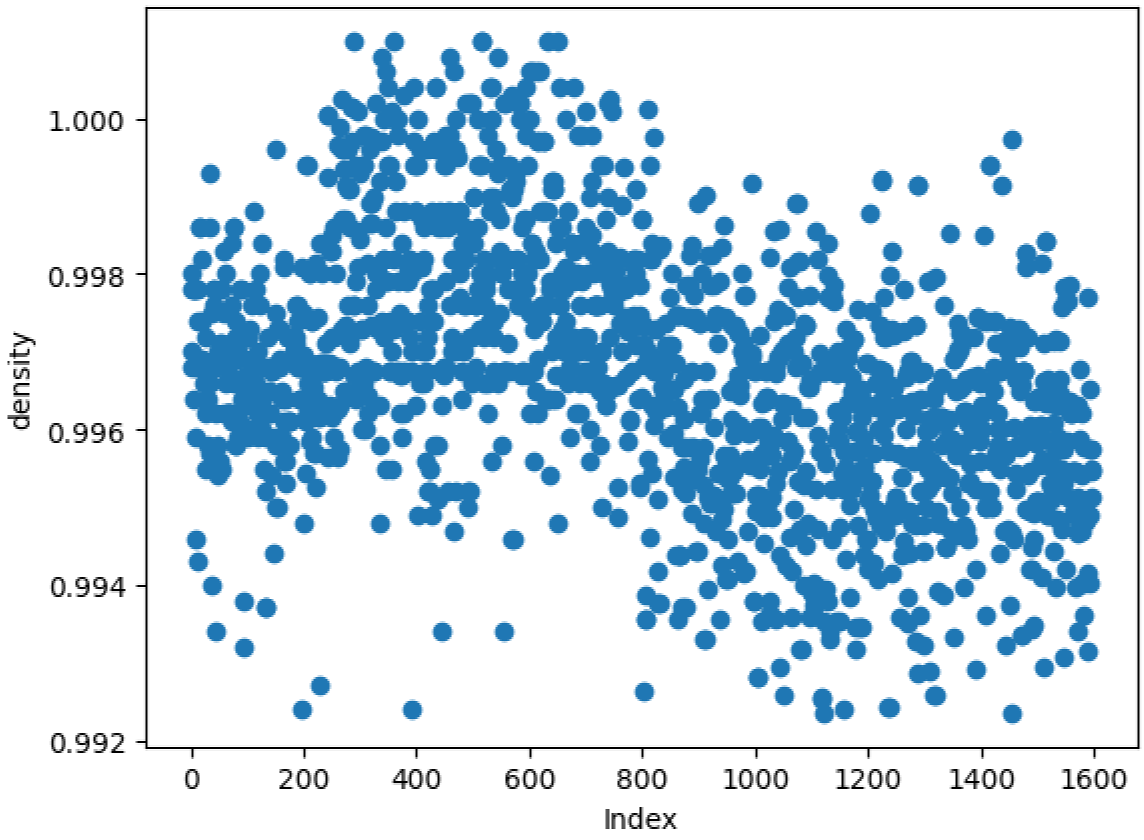

0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
citric acid
0
200
400
600
800
1000
1200
1400
1600
Index

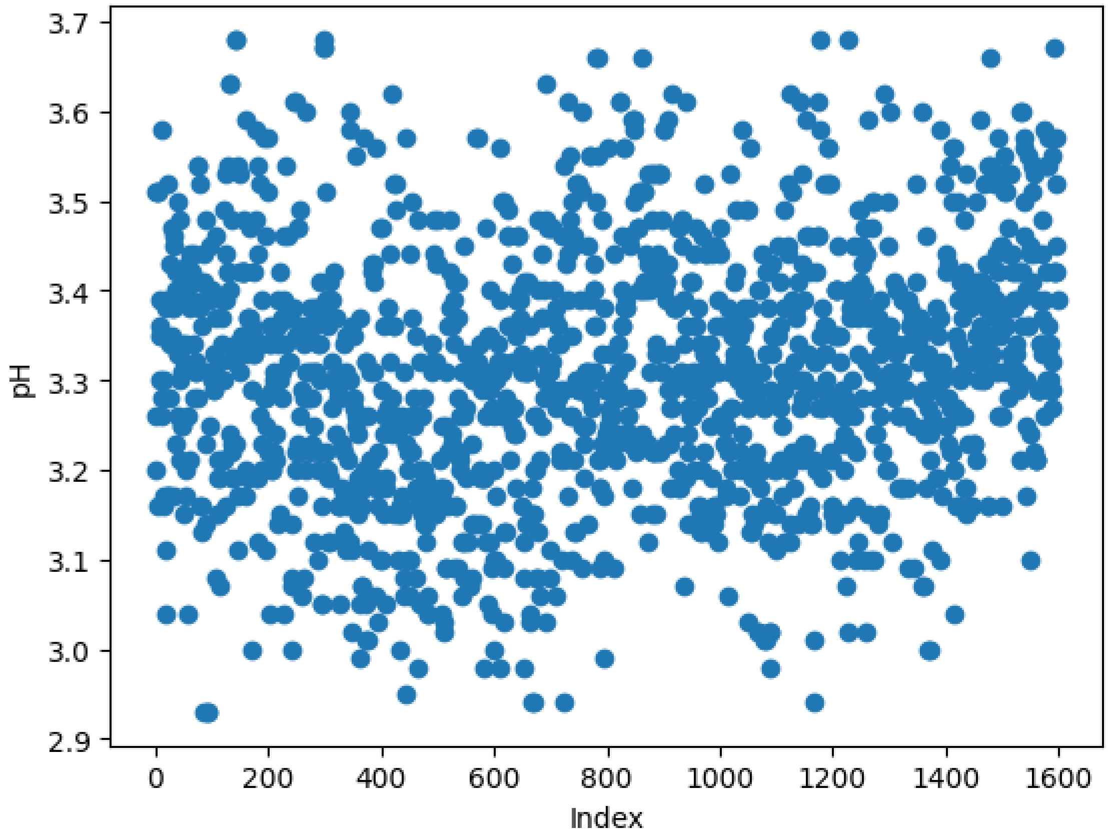

residual sugar
3.5
3.0
2.5
2.0
1.5
1.0
0
200
400
600
800
1000
1200
1400
1600
Index

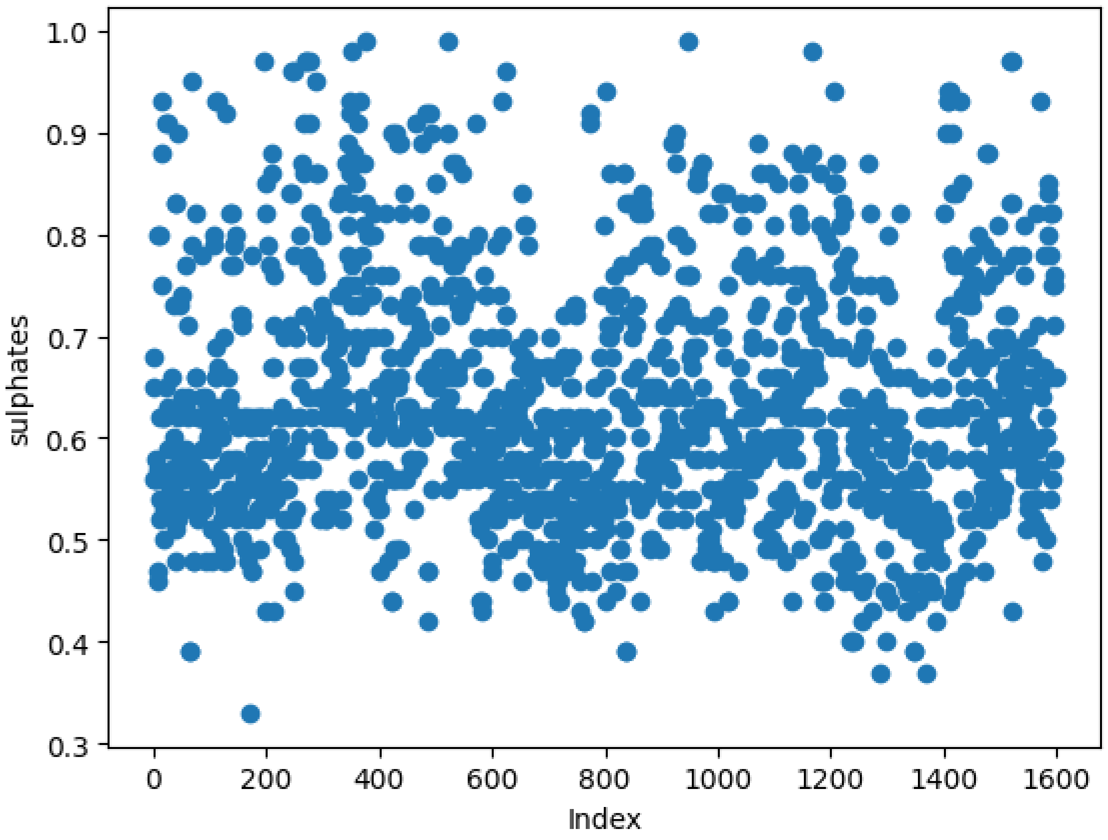

chlorides
0.12
0.11
0.10
0.09
0.08
0.07
0.06
0.05
0.04
0
200
400
600
800
1000
1200
1400
1600
Index

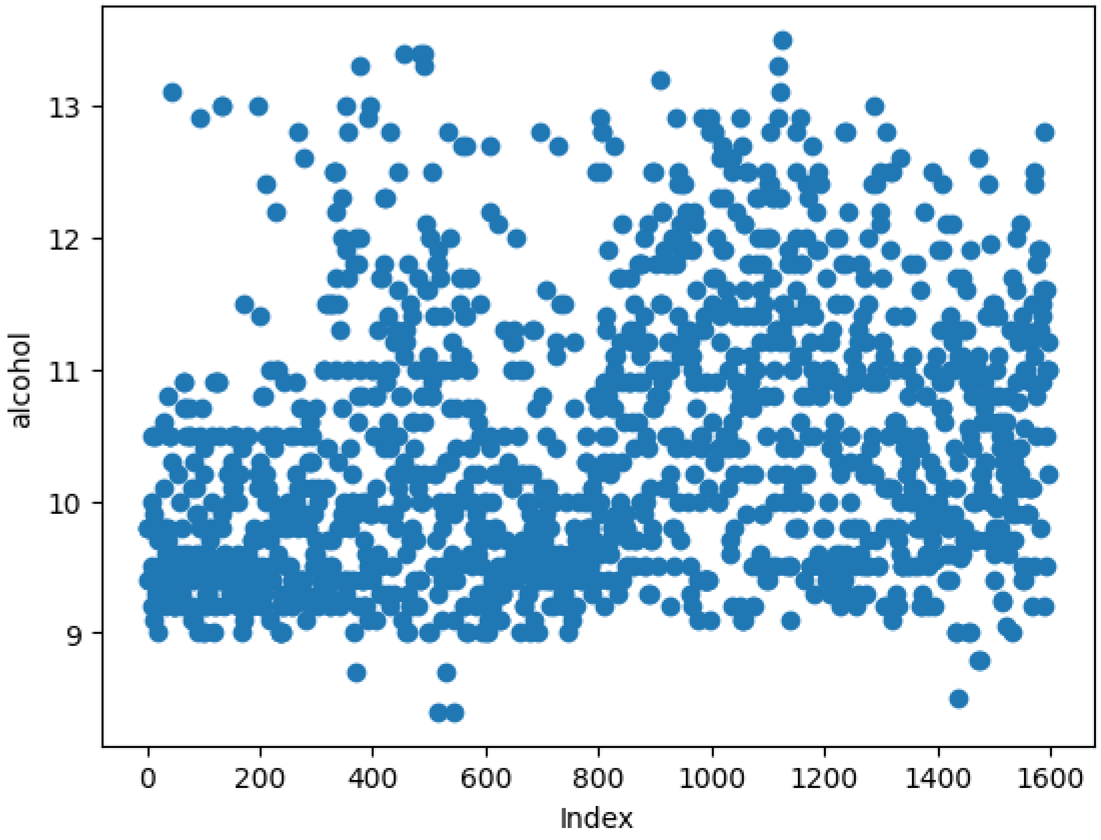

free sulfur dioxide
Index
0
10
20
30
40
0
200
400
600
800
1000
1200
1400
1600


total sulfur dioxide
Index
20
40
60
80
100
120
0
200
400
600
800
1000
1200
1400
1600


density
1.000
0.998
0.996
0.994
0.992
0
200
400
600
800
1000
1200
1400
1600
Index


pH
3.7
3.6
3.5
3.4
3.3
3.2
3.1
3.0
2.9
0
200
400
600
800
1000
1200
1400
1600
Index





In [38]:
```
from sklearn.model_selection import train_test_split
```

In [39]:
```
features = ['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', '
X = df_clean[features]

y = df_clean['quality']
```

In [40]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [41]:
```
from sklearn.linear_model import LogisticRegression
```

In [42]:
```
model = LogisticRegression()
```

In [43]:
```
model.fit(X_train, y_train)
```

Out[43]:
```
▾ LogisticRegression
LogisticRegression()
```

In [46]:
```
from sklearn.metrics import accuracy_score
```

In [47]:
```
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.8625
```

In [ ]:

In [ ]:

In [ ]: