In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [2]:

```
# Load the dataset
wine_data = pd.read_csv("WineQT.csv")
```

In [3]:

```
# Display the first few rows of the dataset
wine_data.head()
```

Out[3]:

	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 [4]:

```
# Performing data exploration
wine_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1143 entries, 0 to 1142
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1143 non-null   float64
 1   volatile acidity      1143 non-null   float64
 2   citric acid           1143 non-null   float64
 3   residual sugar        1143 non-null   float64
 4   chlorides             1143 non-null   float64
 5   free sulfur dioxide   1143 non-null   float64
 6   total sulfur dioxide  1143 non-null   float64
 7   density               1143 non-null   float64
 8   pH                    1143 non-null   float64
 9   sulphates             1143 non-null   float64
 10  alcohol               1143 non-null   float64
 11  quality               1143 non-null   int64  
 12  Id                    1143 non-null   int64  
dtypes: float64(11), int64(2)
memory usage: 116.2 KB
```

In [5]:

```
# check for any missing or null data
missing_data = wine_data.isnull().sum()
print("Missing Data:\n", missing_data)
```

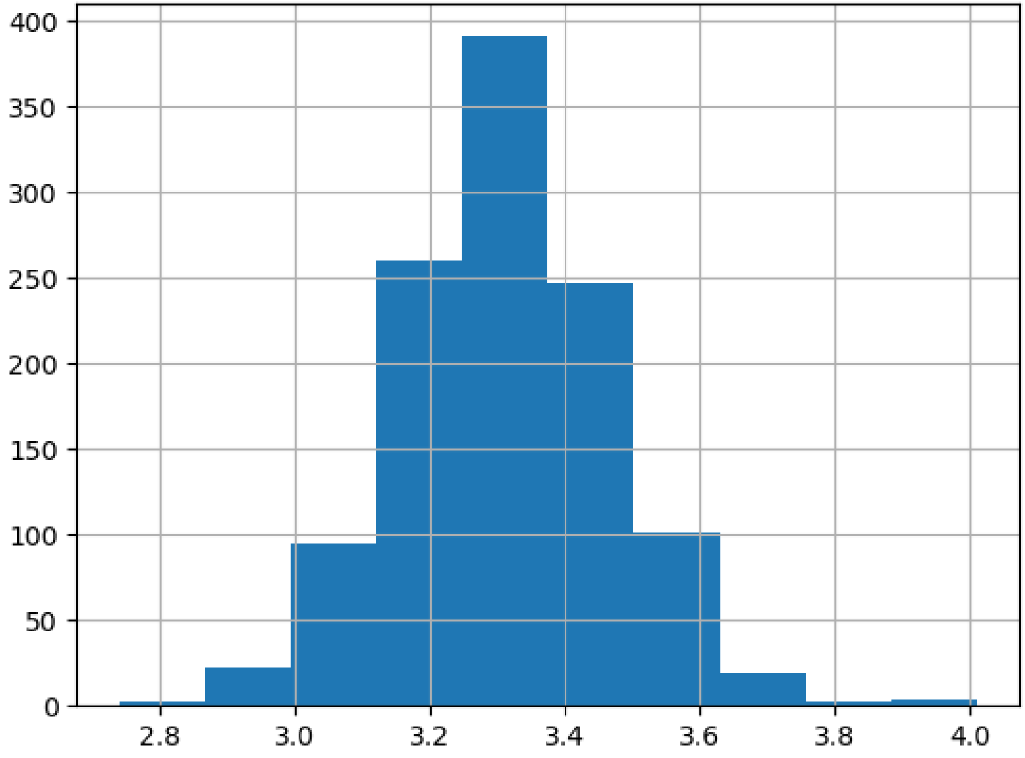
```
Missing Data:
 fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
Id                      0
dtype: int64
```

In [14]:

```
wine_data['pH'].hist()
```

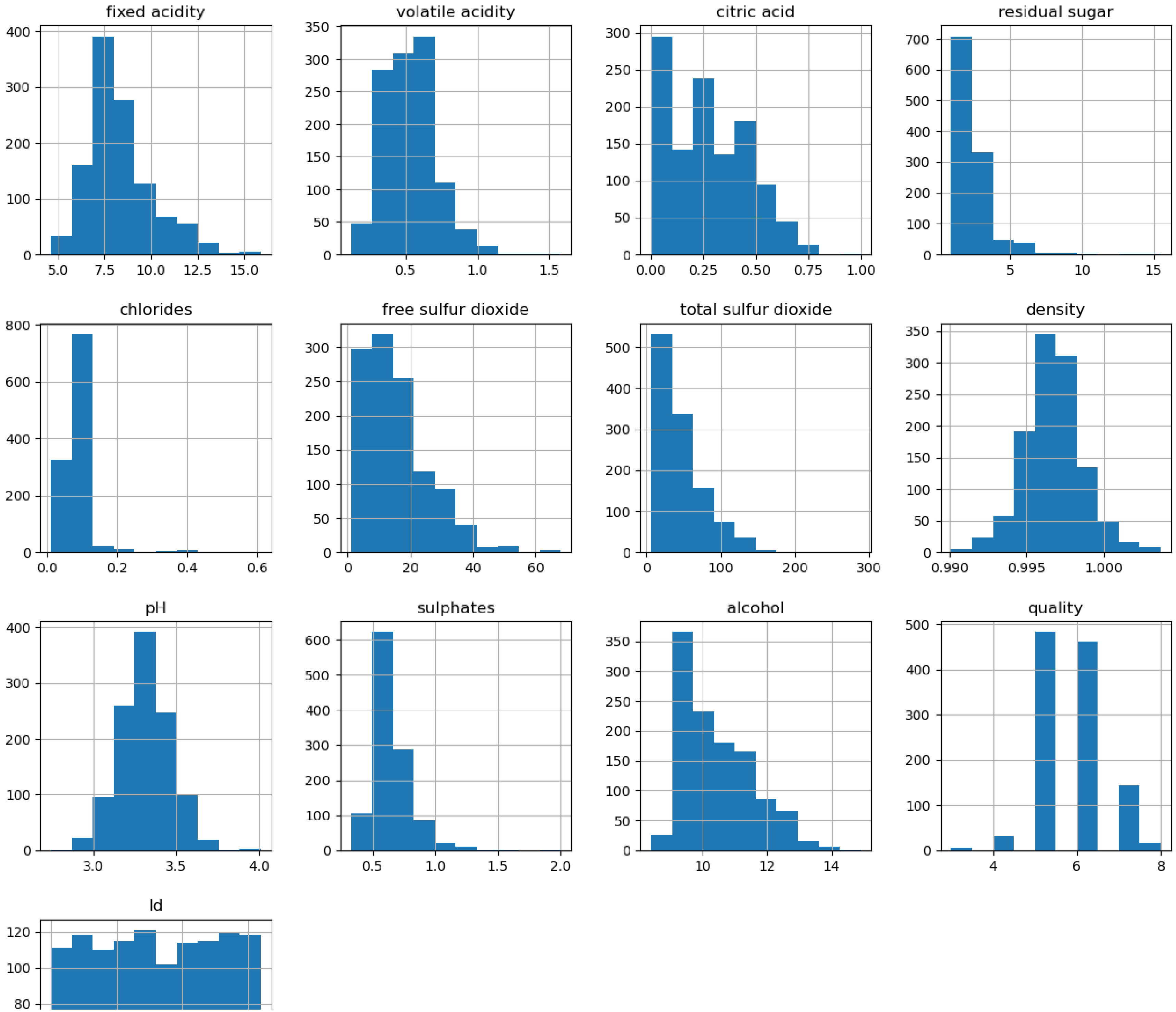
Out[14]:

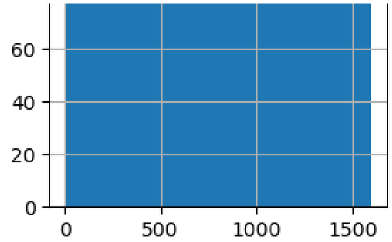
```
<Axes: >
```



```
In [15]: wine_data.hist(figsize=(15,15))
```

```
Out[15]: array([[<Axes: title={'center': 'fixed acidity'}>,
                 <Axes: title={'center': 'volatile acidity'}>,
                 <Axes: title={'center': 'citric acid'}>,
                 <Axes: title={'center': 'residual sugar'}>],
                [<Axes: title={'center': 'chlorides'}>,
                 <Axes: title={'center': 'free sulfur dioxide'}>,
                 <Axes: title={'center': 'total sulfur dioxide'}>,
                 <Axes: title={'center': 'density'}>],
                [<Axes: title={'center': 'pH'}>,
                 <Axes: title={'center': 'sulphates'}>,
                 <Axes: title={'center': 'alcohol'}>,
                 <Axes: title={'center': 'quality'}>],
                [<Axes: title={'center': 'Id'}>, <Axes: >, <Axes: >, <Axes: >]],
               dtype=object)
```


fixed acidity
400
300
200
100
0
5.0
7.5
10.0
12.5
15.0
volatile acidity
350
300
250
200
150
100
50
0
0.5
1.0
1.5
citric acid
300
250
200
150
100
50
0
0.00
0.25
0.50
0.75
1.00
residual sugar
700
600
500
400
300
200
100
0
5
10
15
chlorides
800
600
400
200
0
0.0
0.2
0.4
0.6
free sulfur dioxide
300
250
200
150
100
50
0
0
20
40
60
total sulfur dioxide
500
400
300
200
100
0
0
100
200
300
density
350
300
250
200
150
100
50
0
0.990
0.995
1.000
pH
400
300
200
100
0
3.0
3.5
4.0
sulphates
600
500
400
300
200
100
0
0.5
1.0
1.5
2.0
alcohol
350
300
250
200
150
100
50
0
10
12
14
quality
500
400
300
200
100
0
4
6
8
Id
120
100
80

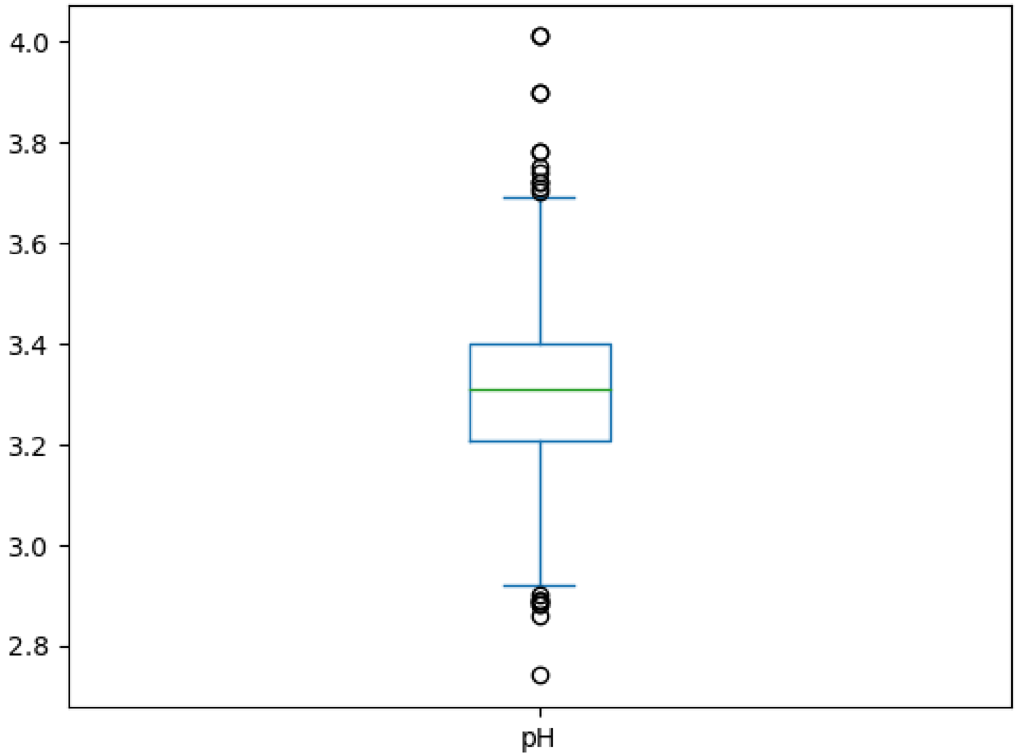


In [22]:

```
wine_data['pH'].plot(kind='box')
```

Out[22]:

```
<Axes: >
```



In [16]:

```
wine_data.corr()
```

Out[16]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Id
fixed acidity	1.000000	-0.250728	0.673157	0.171831	0.107889	-0.164831	-0.110628	0.681501	-0.685163	0.174592	-0.075055	0.121970	-0.275826
volatile acidity	-0.250728	1.000000	-0.544187	-0.005751	0.056336	-0.001962	0.077748	0.016512	0.221492	-0.276079	-0.203909	-0.407394	-0.007892
citric acid	0.673157	-0.544187	1.000000	0.175815	0.245312	-0.057589	0.036871	0.375243	-0.546339	0.331232	0.106250	0.240821	-0.139011
residual sugar	0.171831	-0.005751	0.175815	1.000000	0.070863	0.165339	0.190790	0.380147	-0.116959	0.017475	0.058421	0.022002	-0.046344
chlorides	0.107889	0.056336	0.245312	0.070863	1.000000	0.015280	0.048163	0.208901	-0.277759	0.374784	-0.229917	-0.124085	-0.088099
free sulfur dioxide	-0.164831	-0.001962	-0.057589	0.165339	0.015280	1.000000	0.661093	-0.054150	0.072804	0.034445	-0.047095	-0.063260	0.095268
total sulfur dioxide	-0.110628	0.077748	0.036871	0.190790	0.048163	0.661093	1.000000	0.050175	-0.059126	0.026894	-0.188165	-0.183339	-0.107389
density	0.681501	0.016512	0.375243	0.380147	0.208901	-0.054150	0.050175	1.000000	-0.352775	0.143139	-0.494727	-0.175208	-0.363926
pH	-0.685163	0.221492	-0.546339	-0.116959	-0.277759	0.072804	-0.059126	-0.352775	1.000000	-0.185499	0.225322	-0.052453	0.132904
sulphates	0.174592	-0.276079	0.331232	0.017475	0.374784	0.034445	0.026894	0.143139	-0.185499	1.000000	0.094421	0.257710	-0.103954
alcohol	-0.075055	-0.203909	0.106250	0.058421	-0.229917	-0.047095	-0.188165	-0.494727	0.225322	0.094421	1.000000	0.484866	0.238087
quality	0.121970	-0.407394	0.240821	0.022002	-0.124085	-0.063260	-0.183339	-0.175208	-0.052453	0.257710	0.484866	1.000000	0.069708
Id	-0.275826	-0.007892	-0.139011	-0.046344	-0.088099	0.095268	-0.107389	-0.363926	0.132904	-0.103954	0.238087	0.069708	1.000000

In [17]:

```
# data pre-processing
# Split the data into features (X) and target variable (y)
X = wine_data.drop("quality", axis=1)
y = wine_data["quality"]
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [18]:

```
# Create a linear regression model
linear_reg = LinearRegression()
```

```
# Train the model on the training data
linear_reg.fit(X_train, y_train)
```

Out[18]:

```
▾ LinearRegression
LinearRegression()
```

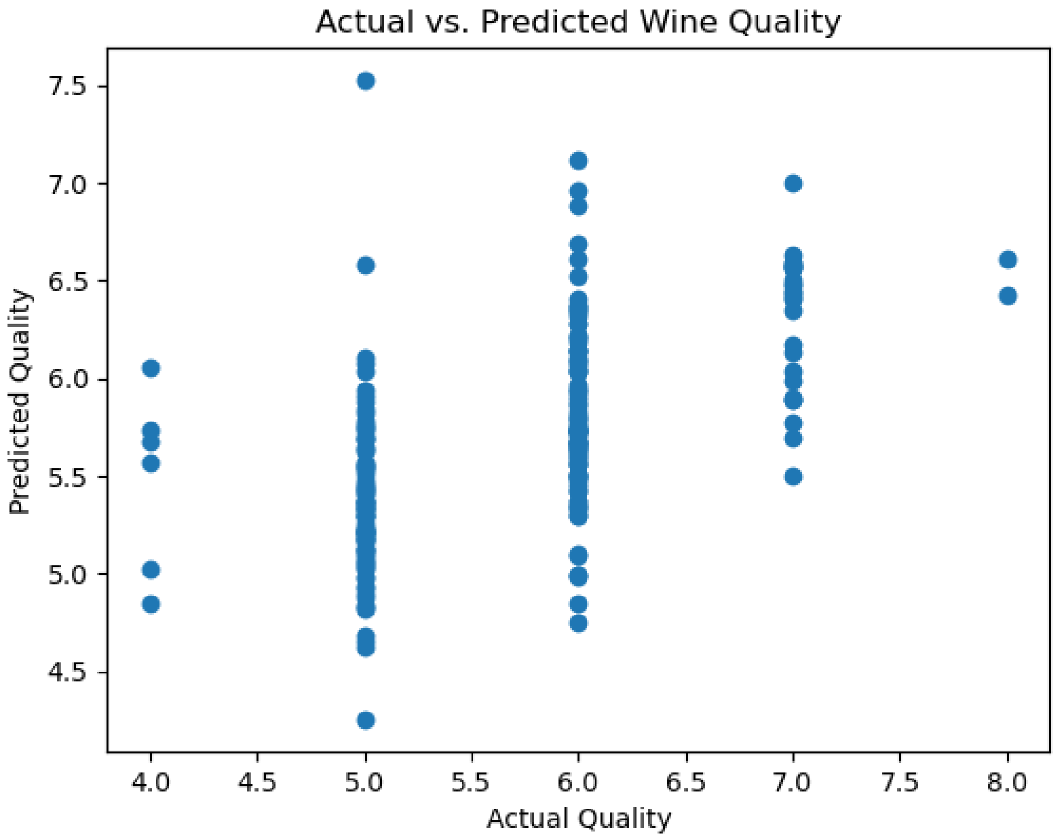
In [20]:

```
# Make predictions on the test data
y_predict = linear_reg.predict(X_test)
# Evaluate the model
mse = mean_squared_error(y_test, y_predict)
r2 = r2_score(y_test, y_predict)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

```
Mean Squared Error: 0.38242835212919696
R-squared: 0.31276385395081874
```

In [21]:

```
# Plot actual vs. predicted values
plt.scatter(y_test, y_predict)
plt.xlabel("Actual Quality")
plt.ylabel("Predicted Quality")
plt.title("Actual vs. Predicted Wine Quality")
plt.show()
```


Actual vs. Predicted Wine Quality
Predicted Quality
7.5
7.0
6.5
6.0
5.5
5.0
4.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Actual Quality

```
In [ ]:
```