In [1]:

```
import pandas as pd
df = pd.read_csv('winequality-red.csv')
df
```

Out[1]:

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

1599 rows × 12 columns

Q1. What are the key features of the wine quality data set? Discuss the importance of each feature in predicting the quality of wine.

In [6]:

```
df.columns
```

Out[6]:

```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual suga
r',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'densit
y',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

Fixed acidity - This refers to the amount of non-volatile acids present in the wine. It is an important feature as it can affect the taste and acidity of the wine, which in turn can impact its overall quality.

Volatile acidity - This refers to the amount of volatile acids present in the wine. Too much volatile acidity can lead to a vinegar-like taste and spoil the wine's quality.

Citric acid - This refers to the amount of citric acid present in the wine. It is an important feature as it can contribute to the wine's freshness and balance its taste.

Residual sugar - This refers to the amount of sugar left over after fermentation. It can impact the sweetness of the wine and its overall balance.

Chlorides - This refers to the amount of salt present in the wine. Too much salt can spoil the wine's taste and reduce its quality.

Free sulfur dioxide - This refers to the amount of sulfur dioxide present in the wine that is not bound to other molecules. It is an important feature as it can act as an antioxidant and help preserve the wine's quality.

Total sulfur dioxide - This refers to the total amount of sulfur dioxide present in the wine, both free and bound. It can impact the wine's overall quality and aging potential.

Density - This refers to the density of the wine. It can be an indicator of the wine's alcohol content and overall body.

pH - This refers to the acidity level of the wine. It can impact the wine's taste and overall balance.

Sulphates - This refers to the amount of sulphates present in the wine. It can act as an antioxidant and help preserve the wine's quality.

Alcohol - This refers to the alcohol content of the wine. It can impact the wine's body, flavor, and overall quality.

Q2. How did you handle missing data in the wine quality data set during the feature engineering process? Discuss the advantages and disadvantages of different imputation techniques.

In [9]:

```
df.isnull().sum()
```

Out[9]:

```
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 dataset has no missing value.

Q3. What are the key factors that affect students' performance in exams? How would you go about analyzing these factors using statistical techniques?

In [10]:

```
df.describe()
```

Out[10]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total d
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.(
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.4
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.8
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.(
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.(
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.(
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.(
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.(

Q4. Describe the process of feature engineering in the context of the student performance data set. How did you select and transform the variables for your model?

Feature engineering is the process of selecting and transforming raw data into features that can be used to train a machine learning model. In the context of the student performance data set, feature engineering involves selecting the most relevant variables (or features) and transforming them into a format that can be used to train a machine learning model.

To select and transform the variables for our model, we can follow the following steps:

Explore the data: Start by exploring the data and gaining an understanding of the variables and their relationships. This can be done by visualizing the data, calculating summary statistics, and looking for correlations between variables.

Select relevant variables: Identify the variables that are most relevant for predicting student performance. This could be based on prior knowledge of the subject matter, domain expertise, or statistical analysis.

Transform variables: Transform the variables into a format that can be used to train a machine learning model. This could involve scaling the variables, converting categorical variables to numeric variables, or creating new variables based on the existing variables.

Feature selection: Once the variables have been transformed, we can use feature selection techniques to identify the most important variables for predicting student performance. This could be done using techniques such as correlation analysis or feature importance scores from a machine learning model.

Model training: Finally, we can train a machine learning model using the selected and transformed variables.

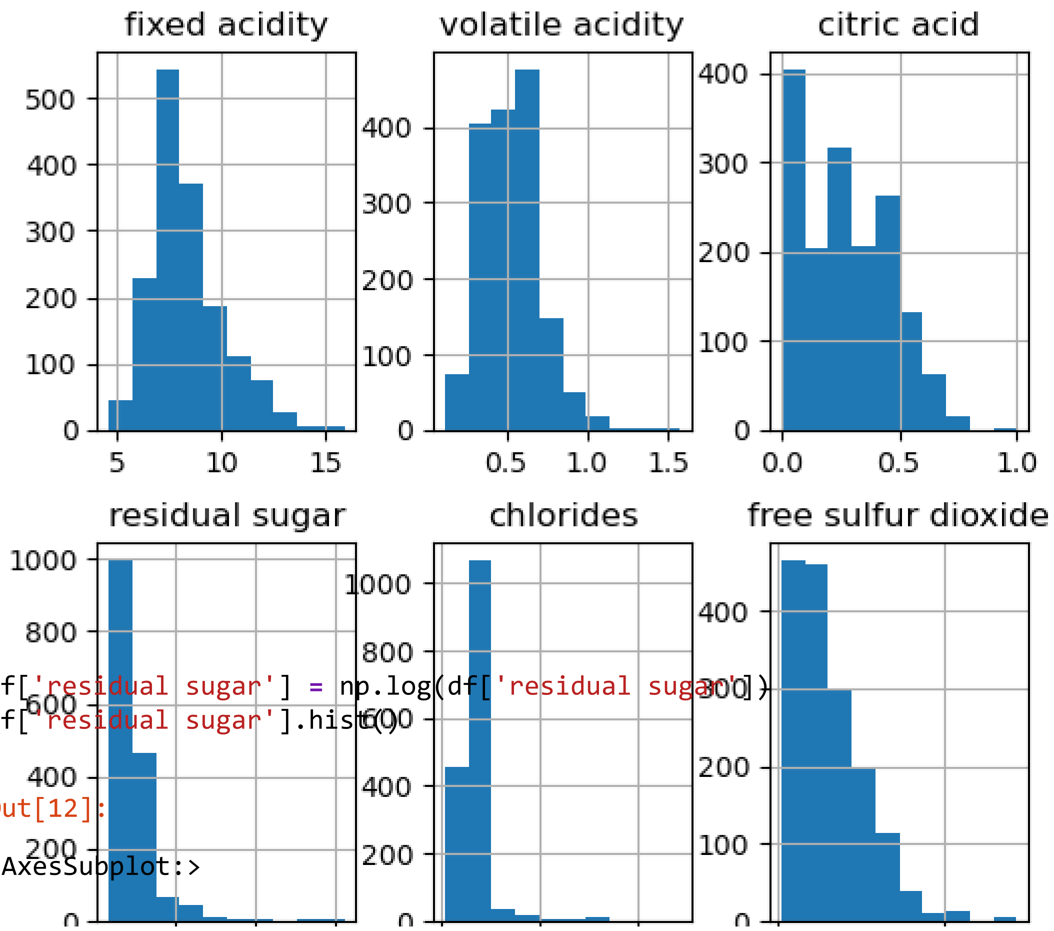
Q5. Load the wine quality data set and perform exploratory data analysis (EDA) to identify the distribution of each feature. Which feature(s) exhibit non-normality, and what transformations could be applied to these features to improve normality?

In [11]:

```
df.hist(figsize = (6,12))
```

Out[11]:

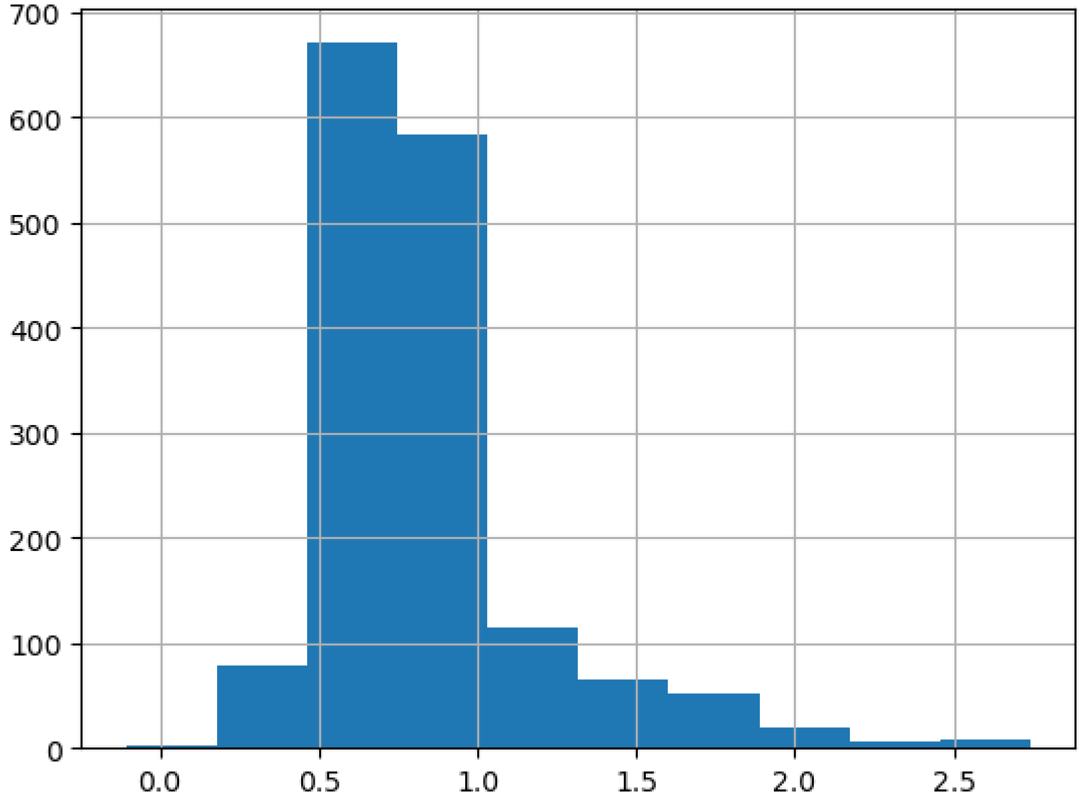
```
array([[<AxesSubplot:title={'center':'fixed acidity'}>,
        <AxesSubplot:title={'center':'volatile acidity'}>,
        <AxesSubplot:title={'center':'citric acid'}>],
       [<AxesSubplot:title={'center':'residual sugar'}>,
        <AxesSubplot:title={'center':'chlorides'}>,
        <AxesSubplot:title={'center':'free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

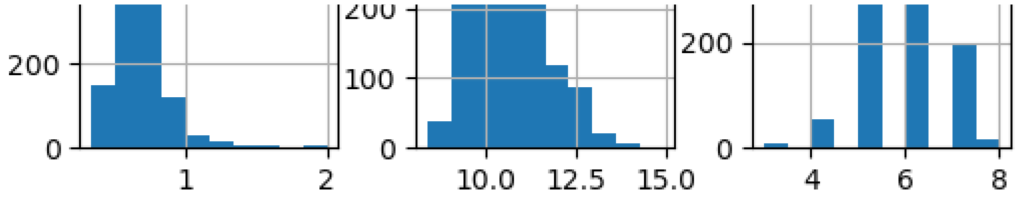


```
df['residual sugar'] = np.log(df['residual sugar'])
df['residual sugar'].hist()
```

Out[12]:

```
<AxesSubplot:>
```



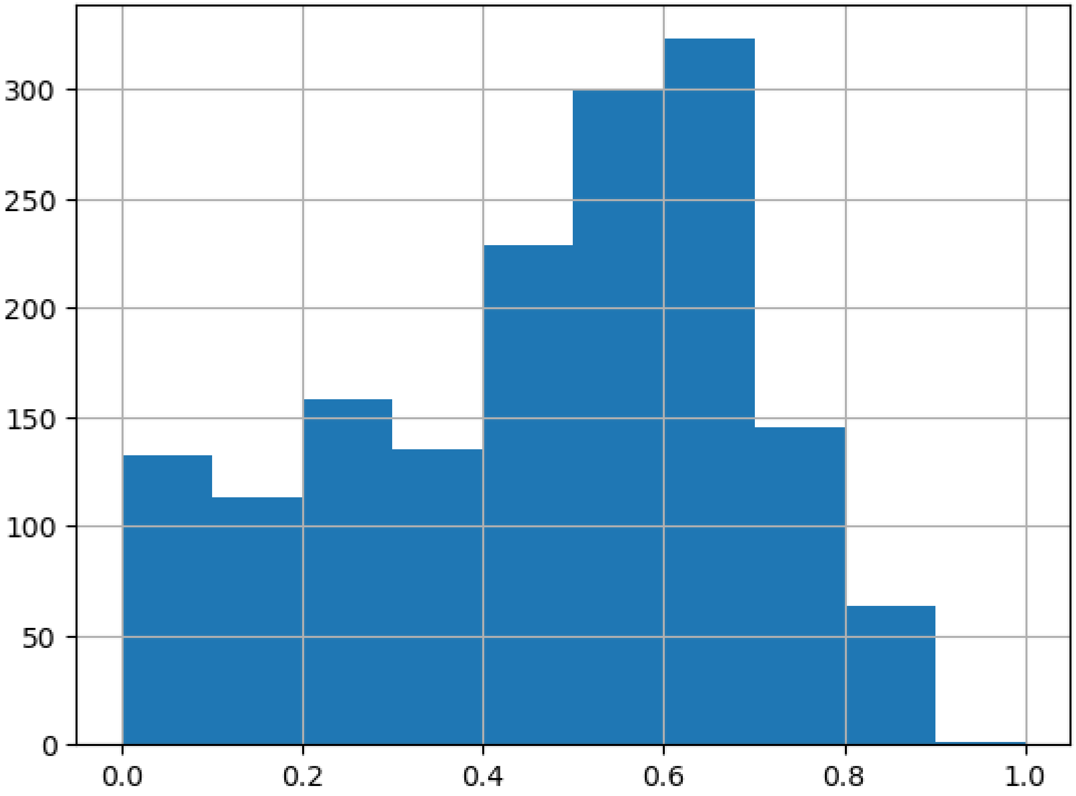


In [13]:

```
df['citric acid'] = np.sqrt(df['citric acid'])
df['citric acid'].hist()
```

Out[13]:

```
<AxesSubplot:>
```



Q6. Using the wine quality data set, perform principal component analysis (PCA) to reduce the number of features. What is the minimum number of principal components required to explain 90% of the variance in the data?

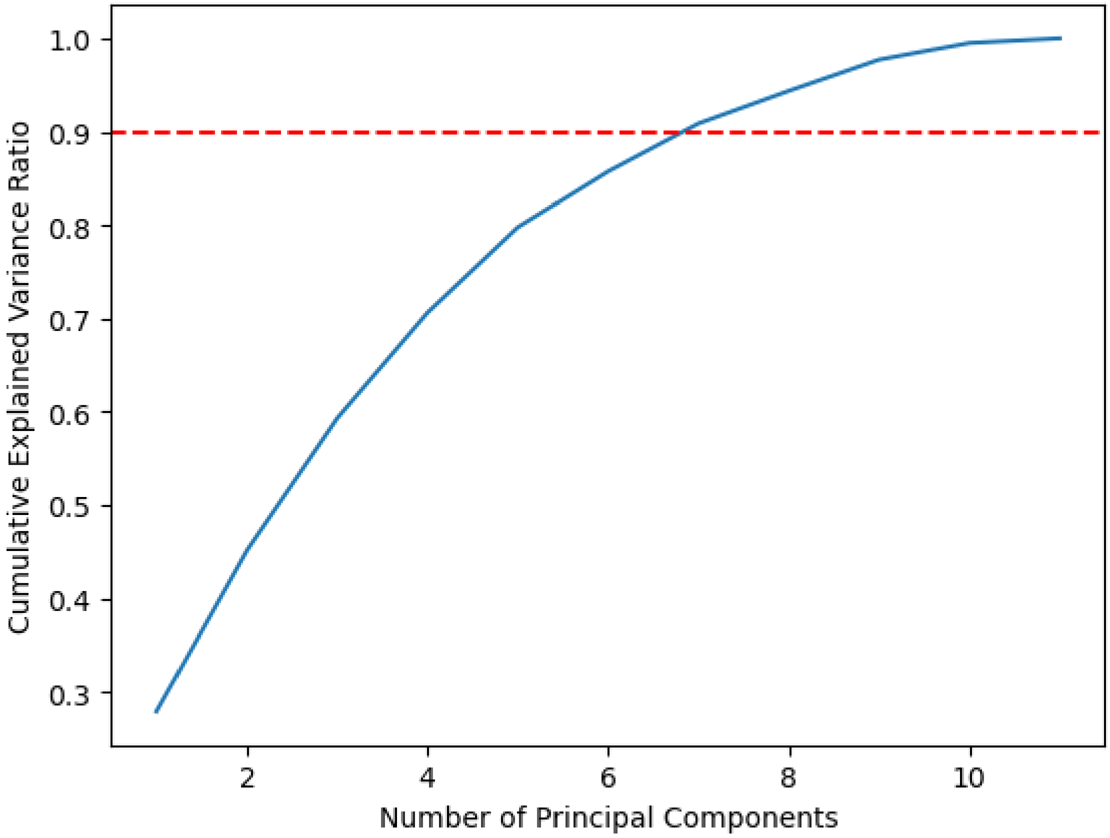
In [15]:

```
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

features = df.drop('quality', axis=1)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(features)

pca = PCA()
pca.fit(X_scaled)

plt.plot(range(1, len(pca.explained_variance_ratio_)+1),
         np.cumsum(pca.explained_variance_ratio_))
plt.axhline(y=0.9, color='r', linestyle='--')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Explained Variance Ratio')
plt.show()
```



In [ ]: