```
# import library
import pandas as Nafasa
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.datasets import load_wine

# Load dataset and convert to DataFrame
data = load_wine()

# mengubah data load_wine menjadi dataframe
df = Nafasa.DataFrame(data.data, columns=data.feature_names)

df['target'] = data.target

df.head(10)
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proant
0	14.23	1.71	2.43	15.60	127.00	2.80	3.06	0.28	
1	13.20	1.78	2.14	11.20	100.00	2.65	2.76	0.26	
2	13.16	2.36	2.67	18.60	101.00	2.80	3.24	0.30	
3	14.37	1.95	2.50	16.80	113.00	3.85	3.49	0.24	
4	13.24	2.59	2.87	21.00	118.00	2.80	2.69	0.39	
5	14.20	1.76	2.45	15.20	112.00	3.27	3.39	0.34	
6	14.39	1.87	2.45	14.60	96.00	2.50	2.52	0.30	
7	14.06	2.15	2.61	17.60	121.00	2.60	2.51	0.31	
8	14.83	1.64	2.17	14.00	97.00	2.80	2.98	0.29	
9	13.86	1.35	2.27	16.00	98.00	2.98	3.15	0.22	

Langkah berikutnya: Buat kode dengan df | Lihat plot yang direkomendasikan | New interactive sheet

```
#mengetahui berapa data/class pada target
df['target'].value_counts()
```

target	count
1	71
0	59
2	48

dtype: int64

```
df.shape
```

```
(178, 14)
```

```
# Mengubah format tampilan angka desimal
Nafasa.options.display.float_format = '{:,.2f}'.format
```

```
# Melihat deskripsi statistik
df['alcohol'].describe()
```

	alcohol
count	178.00
mean	13.00
std	0.81
min	11.03
25%	12.36
50%	13.05
75%	13.68
max	14.83

dtype: float64

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   alcohol                       178 non-null    float64
 1   malic_acid                    178 non-null    float64
 2   ash                           178 non-null    float64
 3   alcalinity_of_ash             178 non-null    float64
 4   magnesium                     178 non-null    float64
 5   total_phenols                 178 non-null    float64
 6   flavanoids                    178 non-null    float64
 7   nonflavanoid_phenols          178 non-null    float64
 8   proanthocyanins               178 non-null    float64
 9   color_intensity               178 non-null    float64
 10  hue                           178 non-null    float64
 11  od280/od315_of_diluted_wines  178 non-null    float64
 12  proline                       178 non-null    float64
 13  target                        178 non-null    int64  
dtypes: float64(13), int64(1)
memory usage: 19.6 KB
```

```
df['target'].unique()
```

```
array([0, 1, 2])
```

```
from sklearn.model_selection import train_test_split

# Memuat Dataset
wine = load_wine()

# Split the data into features (X) and target (y)
X = df.drop('target', axis=1) # Fitur (independent variables)
y = df['target']  # Target (dependent variable)

# Split the data into training and test sets (70% latih dan 30% uji)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
from sklearn.preprocessing import StandardScaler

# Menerapkan StandardScaler
scaler = StandardScaler()
```

```
scaler. fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

```
from sklearn.neighbors import KNeighborsClassifier

# Initialize K-Nearest Neighbor Classifier
knn = KNeighborsClassifier (n_neighbors=3)

# Train the model / Melatih model dengan data pelatihan
knn.fit(X_train, y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

```
from sklearn.metrics import accuracy_score

# Predict on the test set
y_pred = knn.predict (X_test)
y_pred
#menentukan probabilitas prediksi
knn.predict_proba(X_test)
```

Tampilkan output tersembunyi

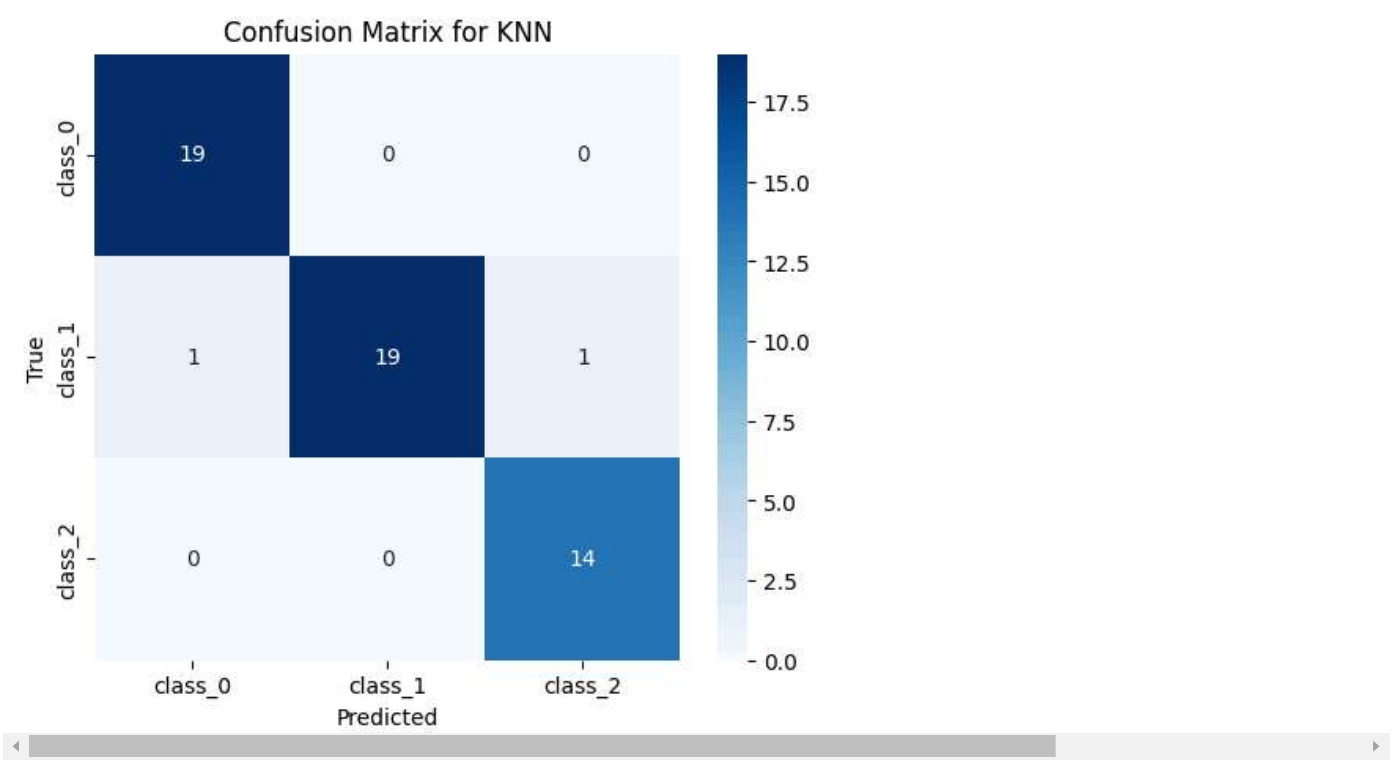
```
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(accuracy * 100)
```

```
96.29629629629629
```

```
from sklearn.metrics import classification_report, confusion_matrix

#Membuat confusion matrix
cm = print(confusion_matrix(y_test, y_pred))

#Membuat laporan klasifikasi
cf = print(classification_report(y_test, y_pred))
```

```
[[19  0  0]
 [ 1 19  1]
 [ 0  0 14]]
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        19
           1       1.00      0.90      0.95        21
           2       0.93      1.00      0.97        14

    accuracy                           0.96        54
   macro avg       0.96      0.97      0.96        54
weighted avg       0.97      0.96      0.96        54
```

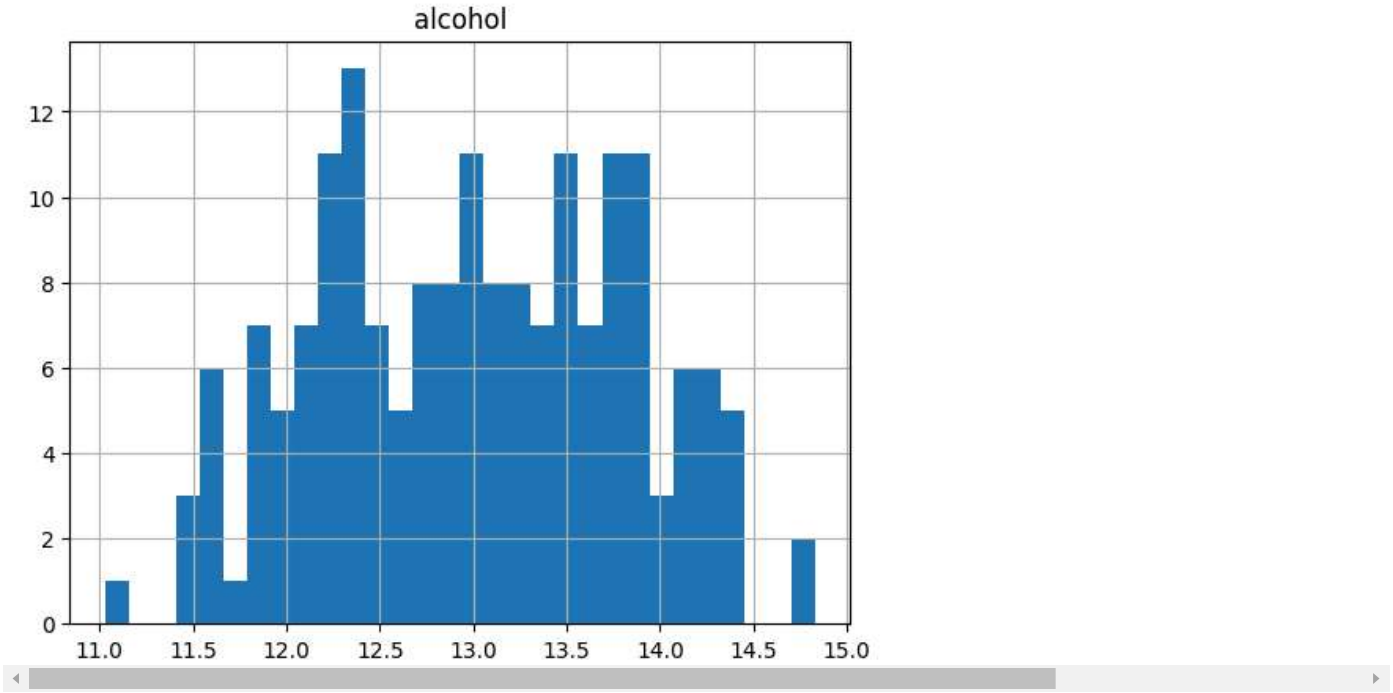
```
#Menampilkan visualisasi confusion matrix
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=data.target_names, yticklabels=data.target_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix for KNN')
plt.show()
```



alcohol

```
# @title alcohol

df.hist(column='alcohol',bins=30)
```

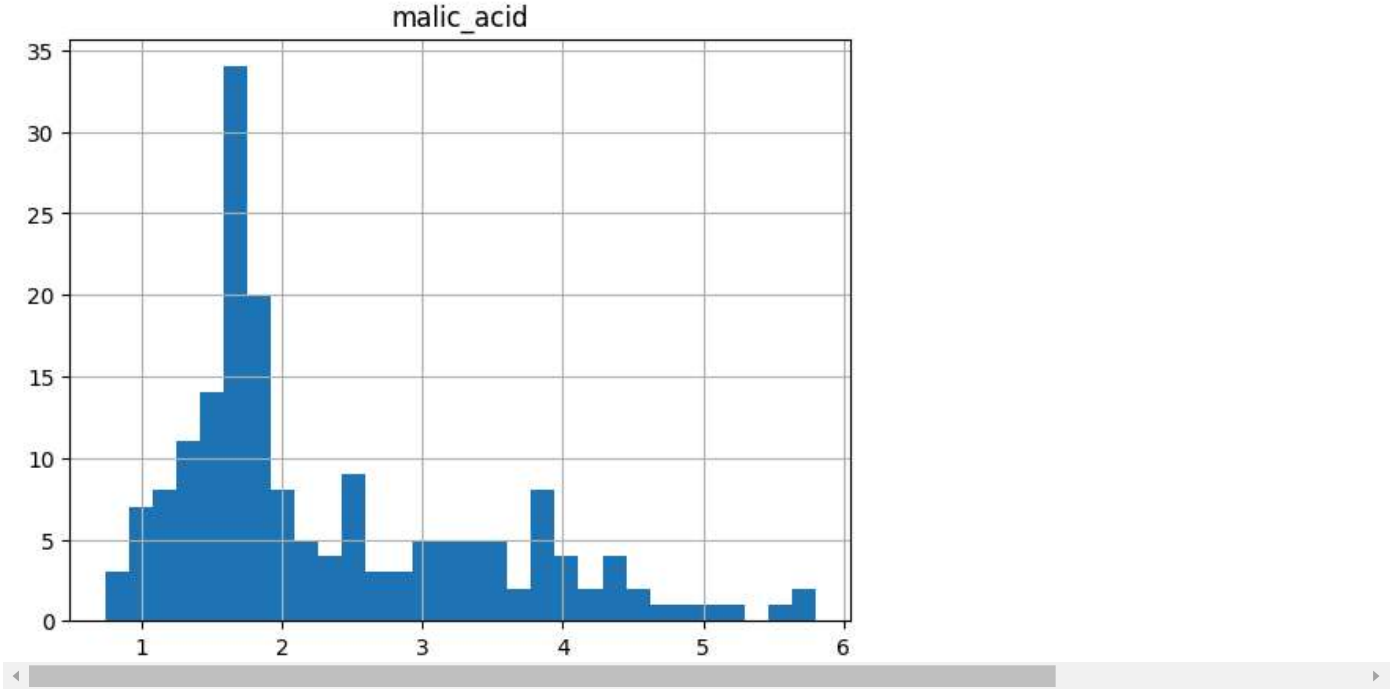
```
array([[<Axes: title={'center': 'alcohol'}>]], dtype=object)
```



malic_acid

```
# @title malic_acid

df.hist(column='malic_acid',bins=30)
```

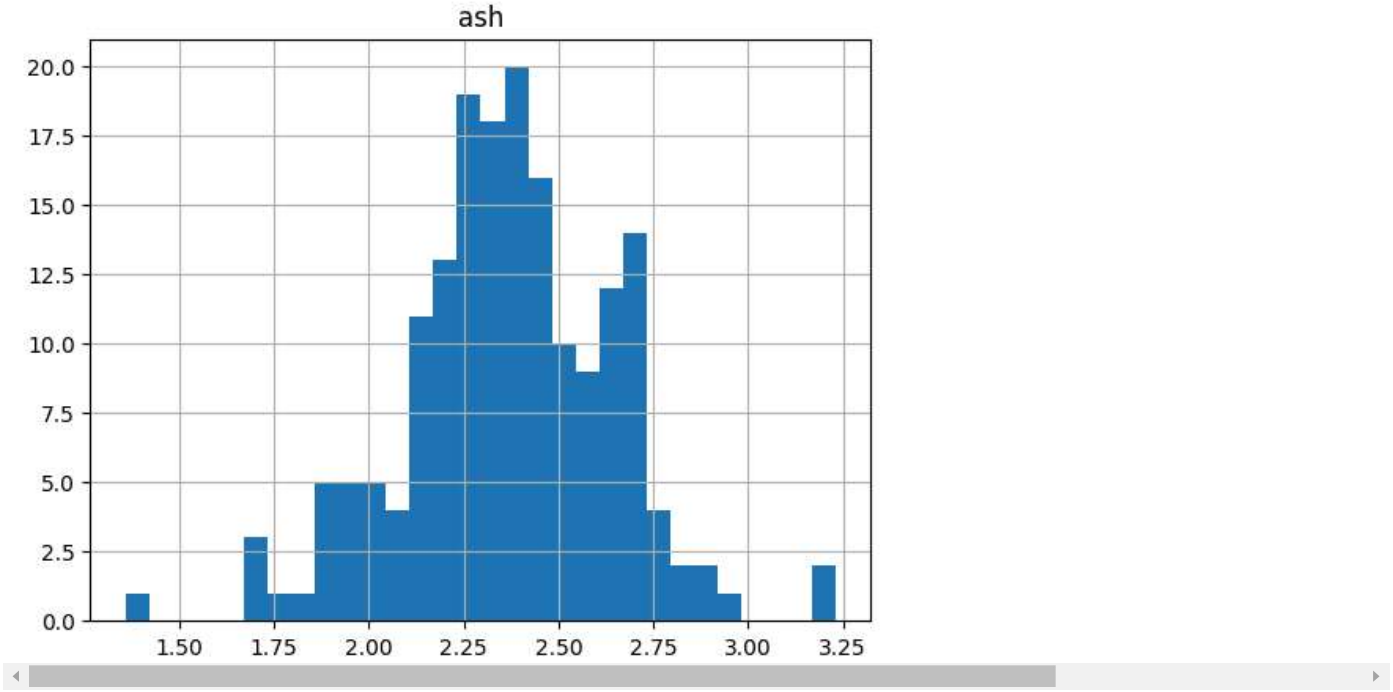
```
array([[<Axes: title={'center': 'malic_acid'}>]], dtype=object)
```



ash

```
# @title ash

df.hist(column='ash',bins=30)
```

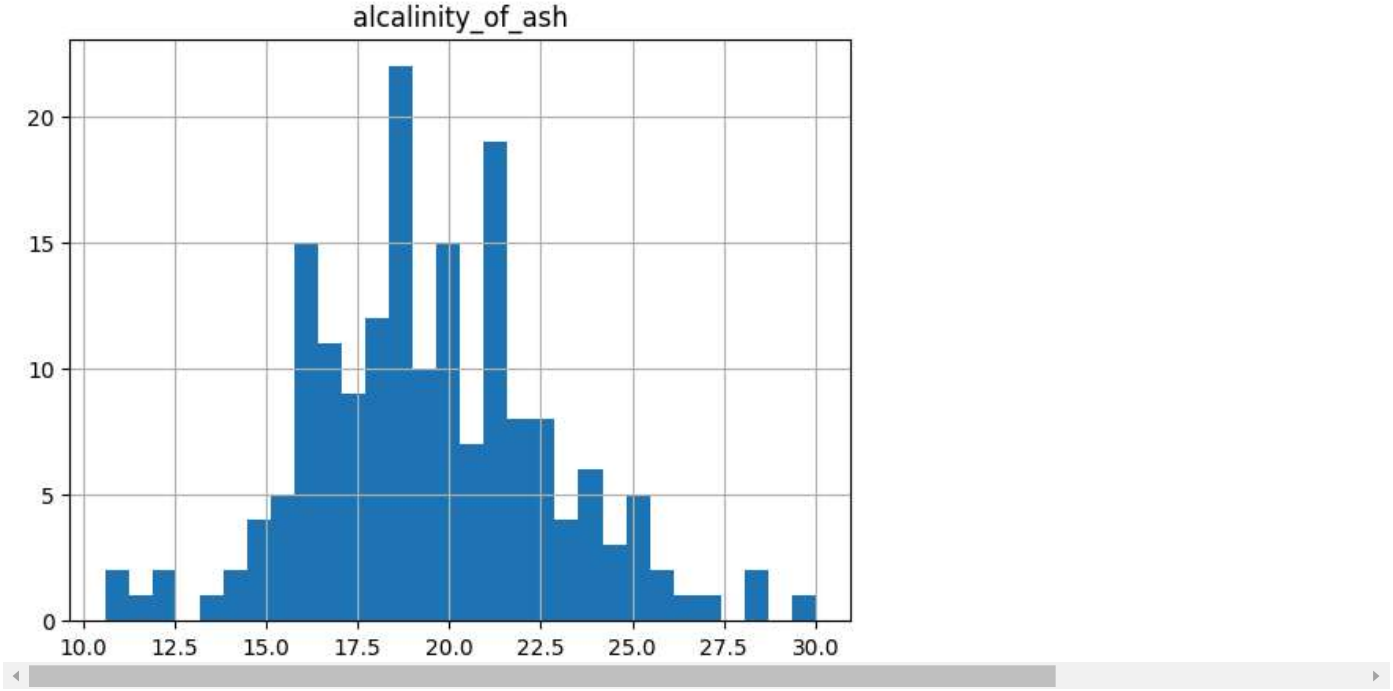
```
array([[<Axes: title={'center': 'ash'}>]], dtype=object)
```



alcalinity_of_ash

```
# @title alcalinity_of_ash

df.hist(column='alcalinity_of_ash',bins=30)
```

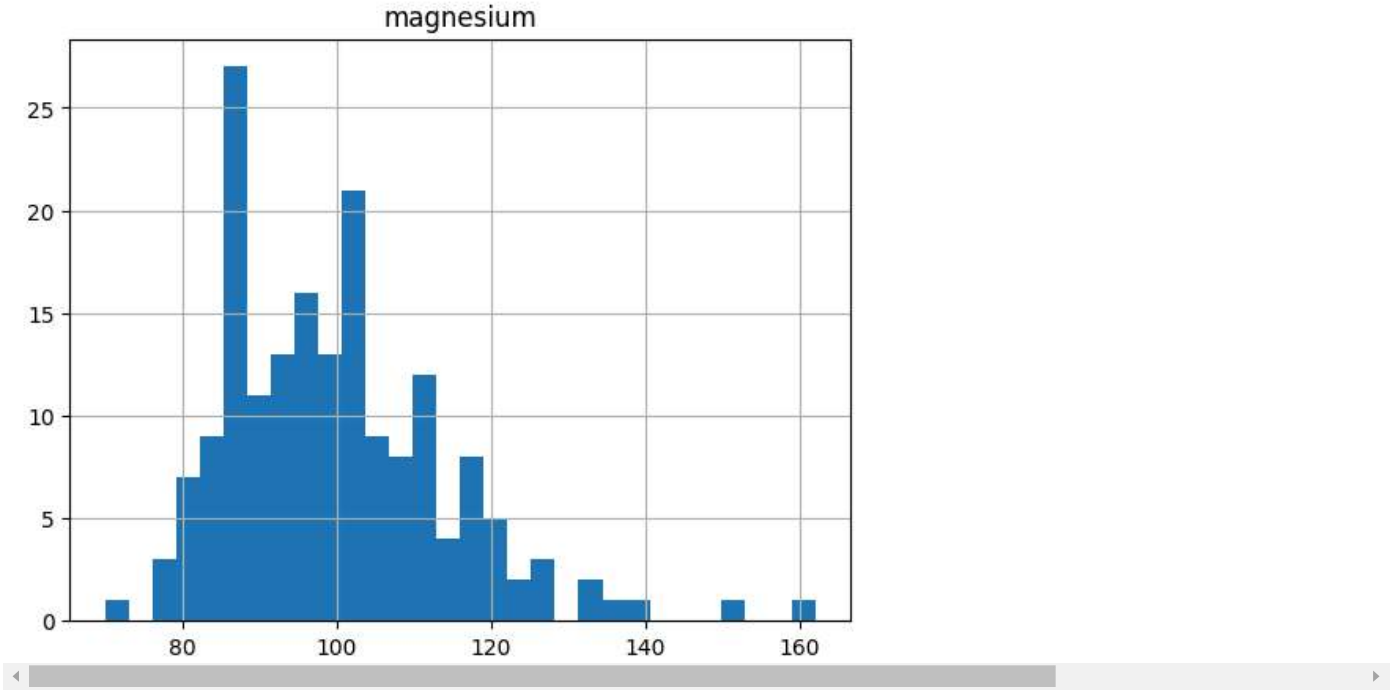
```
array([[<Axes: title={'center': 'alcalinity_of_ash'}>]], dtype=object)
```



magnesium

```
# @title magnesium

df.hist(column='magnesium',bins=30)
```

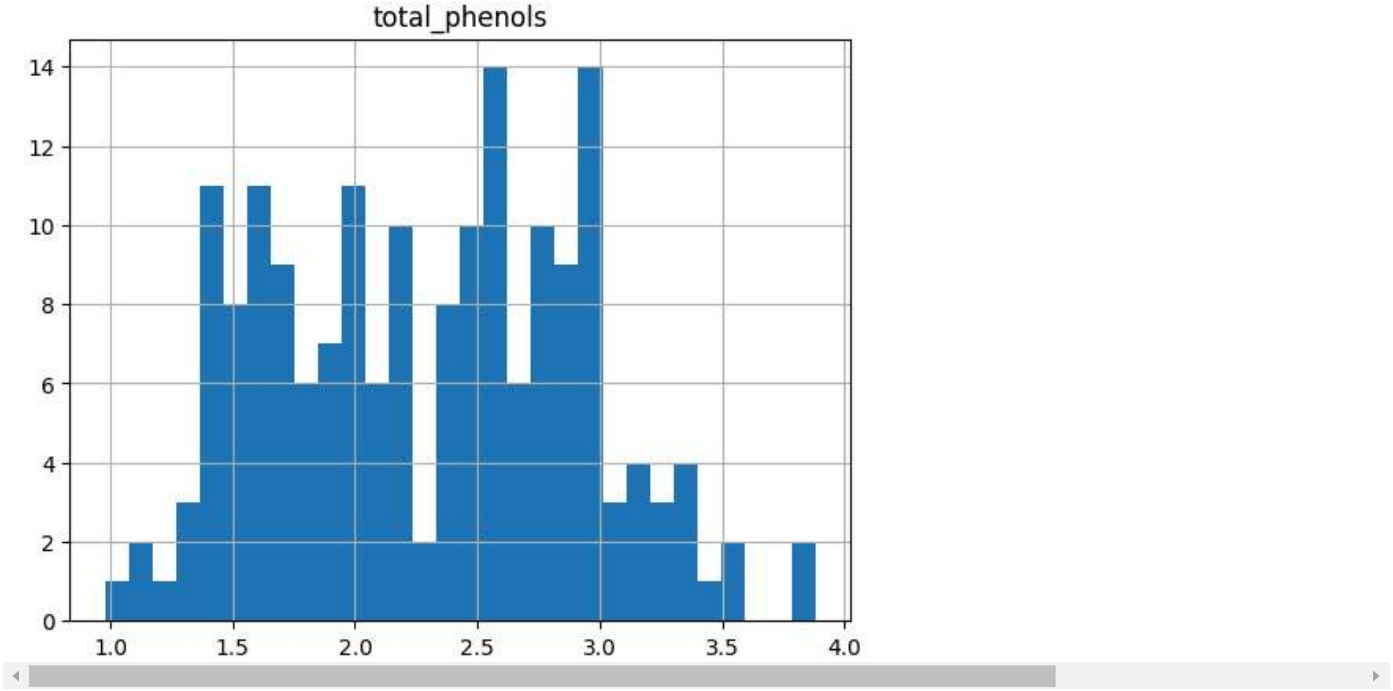
```
array([[<Axes: title={'center': 'magnesium'}>]], dtype=object)
```



total_phenols

```
# @title total_phenols

df.hist(column='total_phenols',bins=30)
```

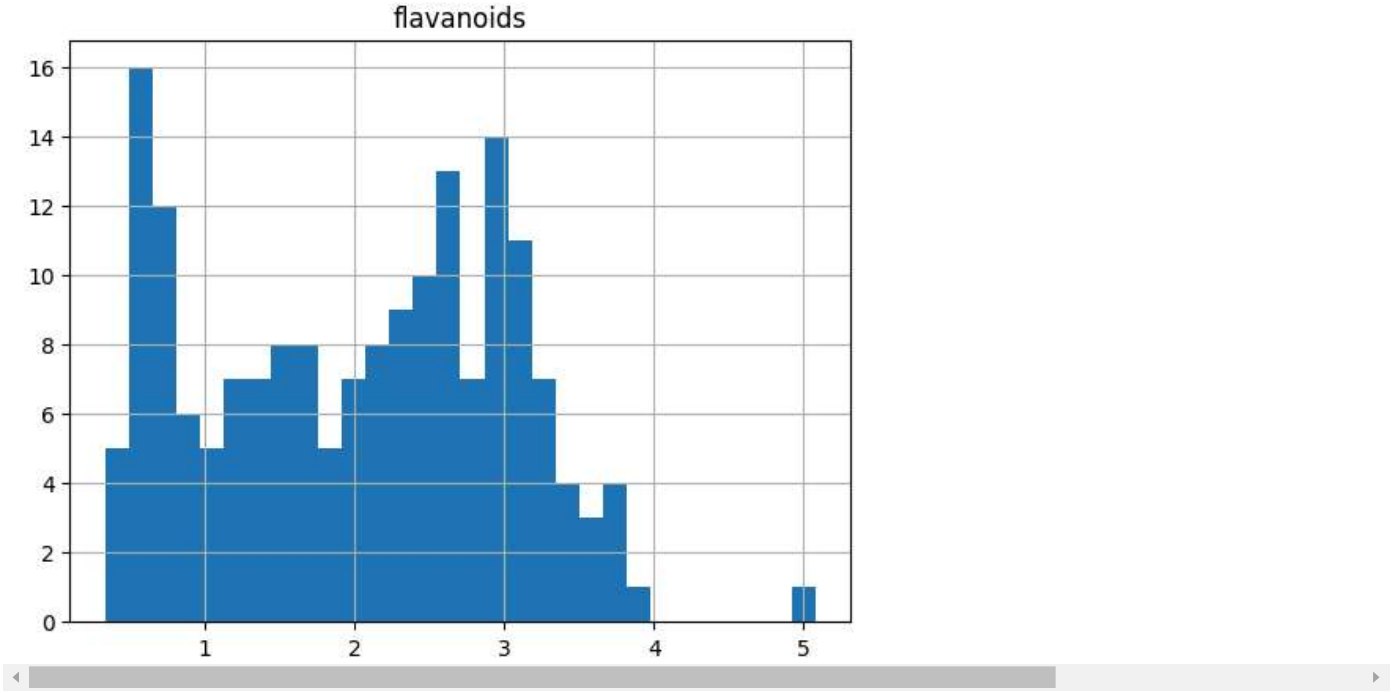
```
array([[<Axes: title={'center': 'total_phenols'}>]], dtype=object)
```



flavanoids

```
# @title flavanoids

df.hist(column='flavanoids',bins=30)
```

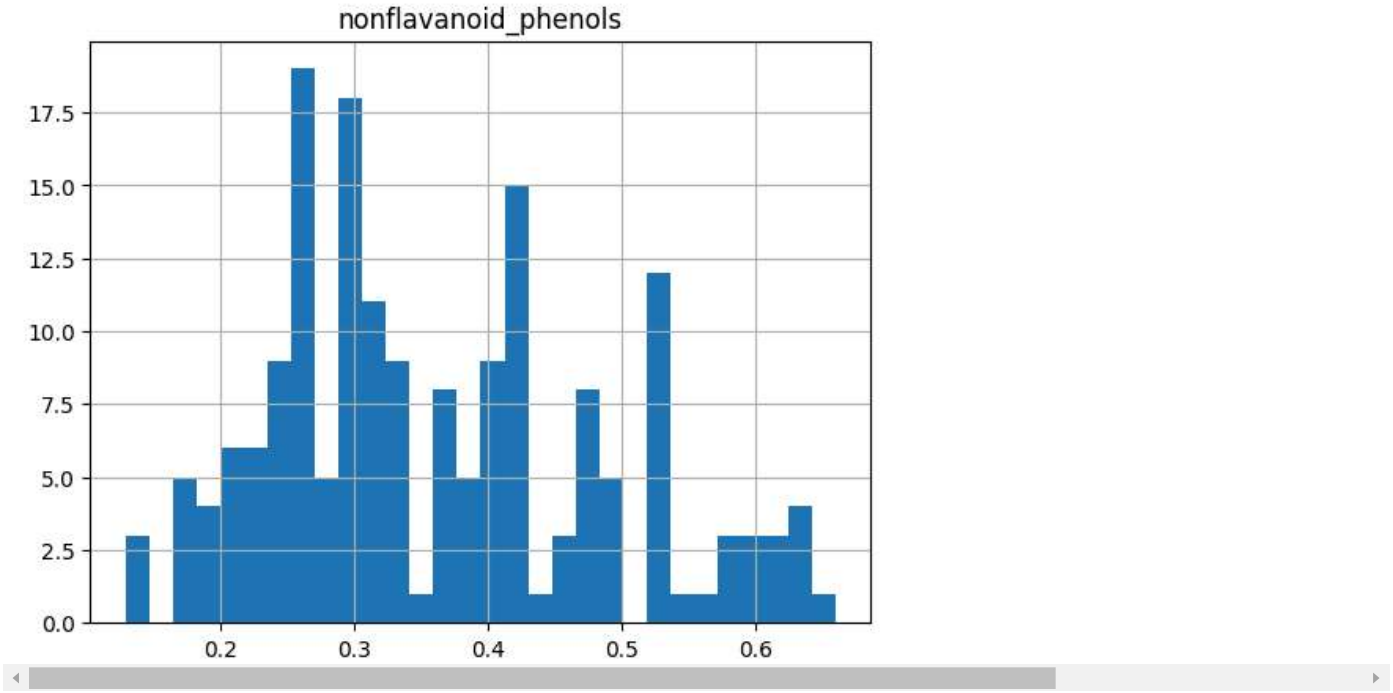
```
array([[<Axes: title={'center': 'flavanoids'}>]], dtype=object)
```



nonflavanoid_phenols

```
# @title nonflavanoid_phenols

df.hist(column='nonflavanoid_phenols',bins=30)
```

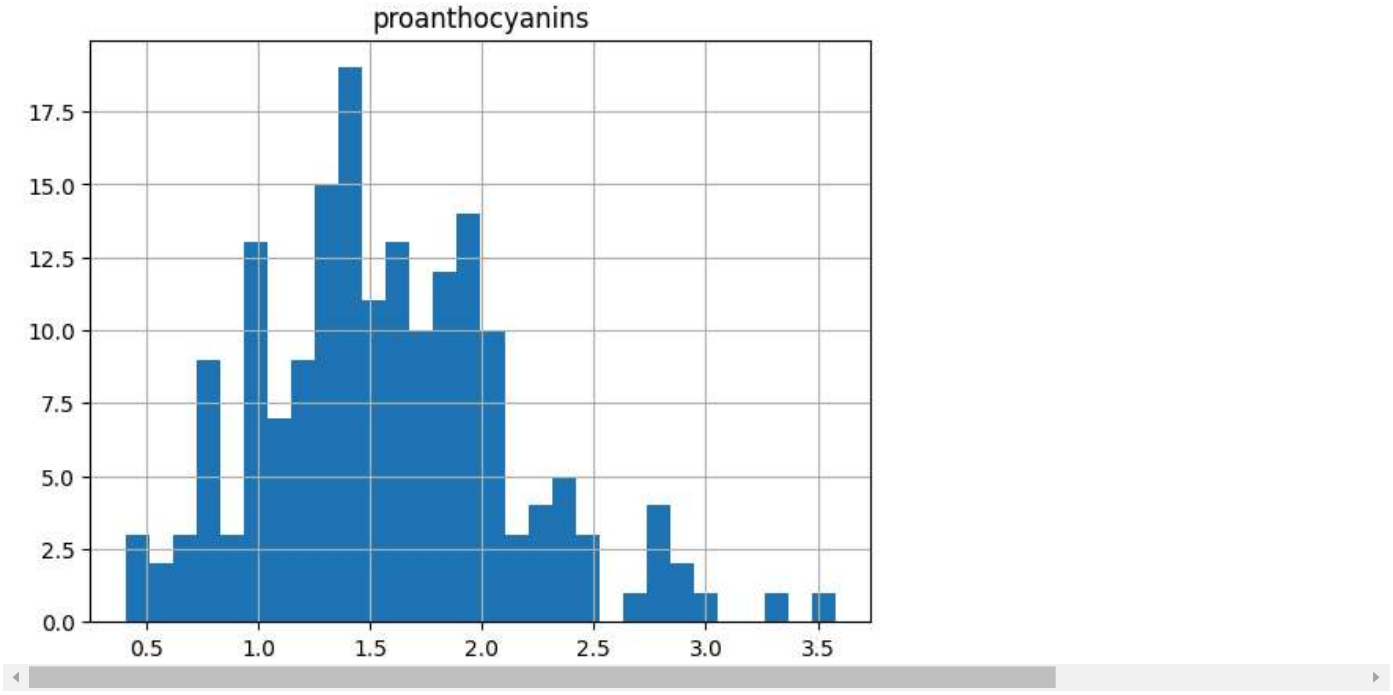
```
array([[<Axes: title={'center': 'nonflavanoid_phenols'}>]], dtype=object)
```



proanthocyanins

```
# @title proanthocyanins

df.hist(column='proanthocyanins',bins=30)
```

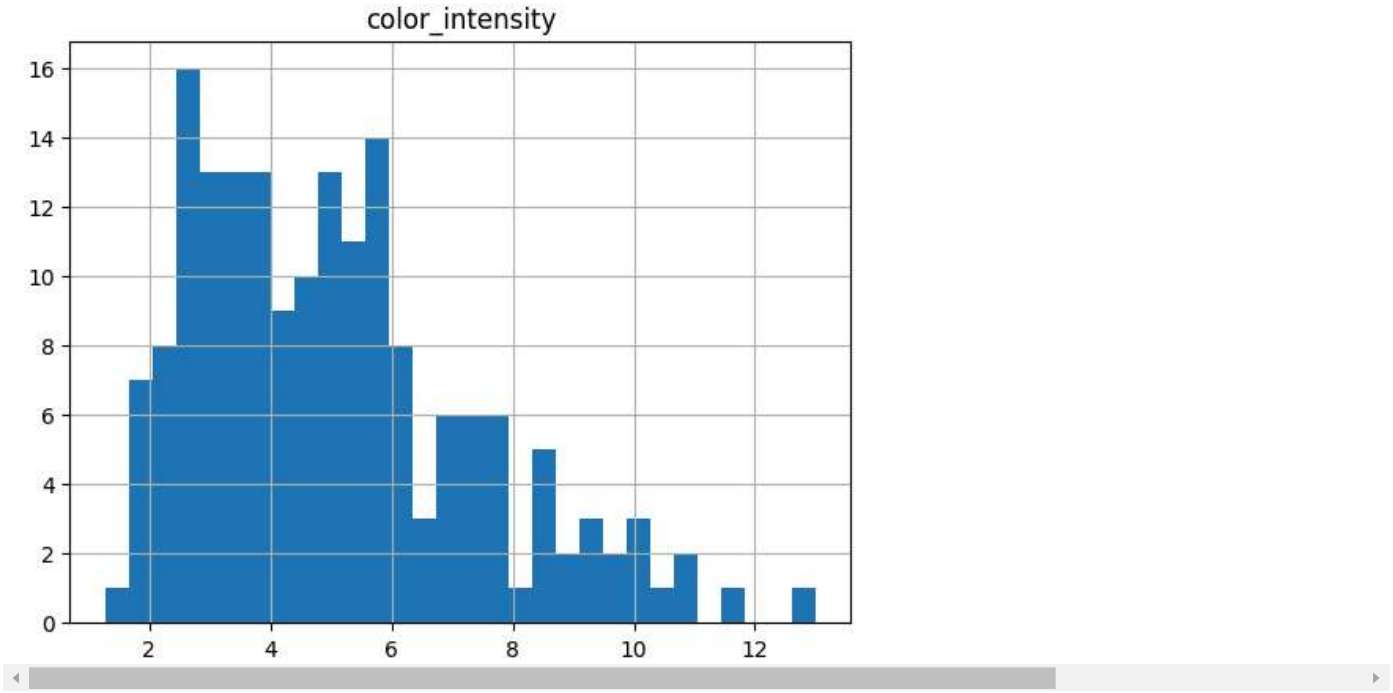
```
array([[<Axes: title={'center': 'proanthocyanins'}>]], dtype=object)
```



color_intensity

```
# @title color_intensity

df.hist(column='color_intensity',bins=30)
```

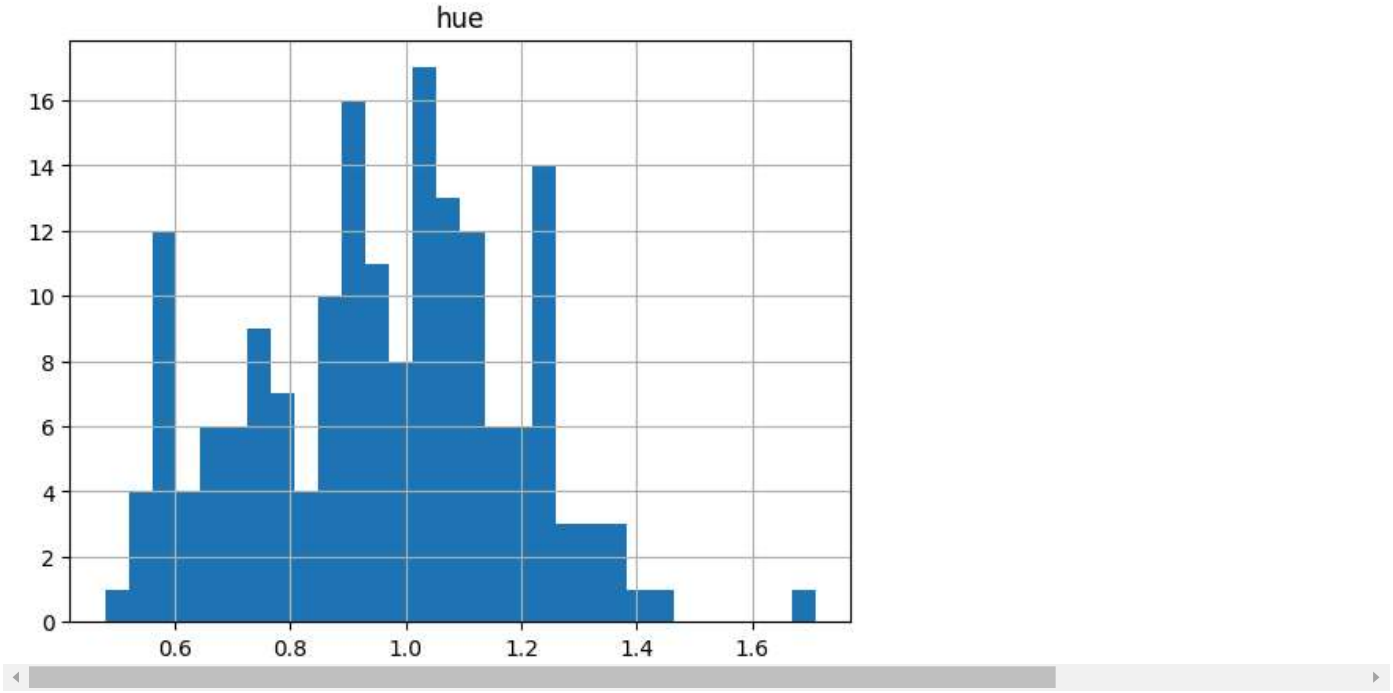
```
array([[<Axes: title={'center': 'color_intensity'}>]], dtype=object)
```



hue

```
# @title hue

df.hist(column='hue',bins=30)
```

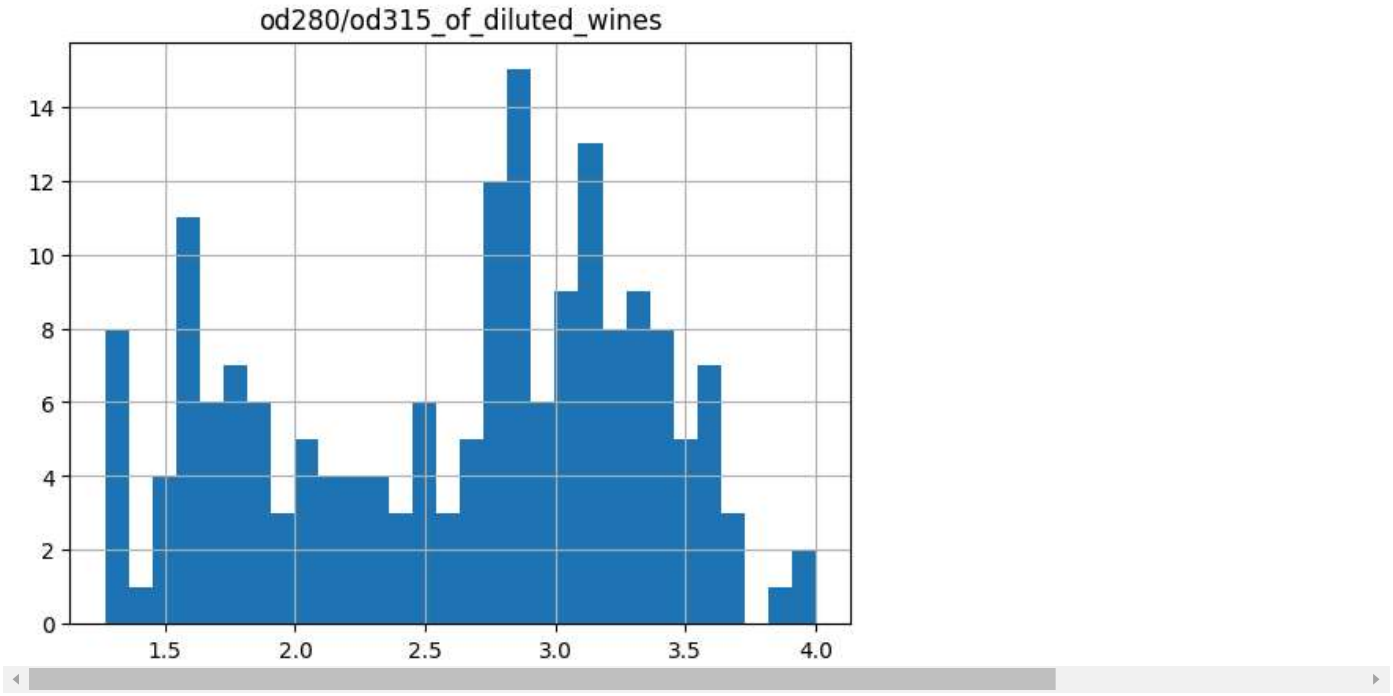
```
array([[<Axes: title={'center': 'hue'}>]], dtype=object)
```



od280/od315_of_diluted_wines

```
# @title od280/od315_of_diluted_wines

df.hist(column='od280/od315_of_diluted_wines',bins=30)
```

```
array([[<Axes: title={'center': 'od280/od315_of_diluted_wines'}>]],
      dtype=object)
```



proline

```
# @title proline

df.hist(column='proline',bins=30)
```

```
array([[<Axes: title={'center': 'proline'}>]], dtype=object)
```