```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import drive
```

```
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
df = pd.read_csv('/content/drive/MyDrive/python-Saylani/WineQT.csv')
```

```
df
```

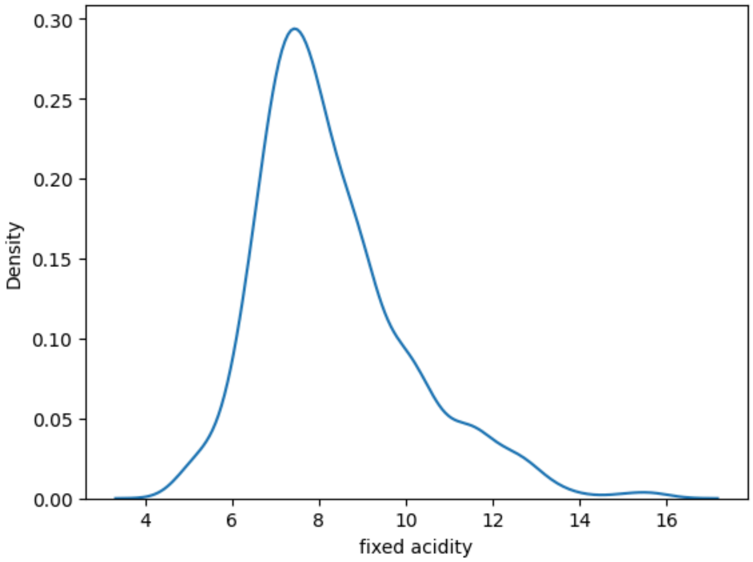
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Id
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	0
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5	1
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5	2
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6	3
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1138	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6	1592
1139	6.8	0.620	0.08	1.9	0.068	28.0	38.0	0.99651	3.42	0.82	9.5	6	1593
1140	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5	1594
1141	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6	1595
1142	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5	1597

```
df = df[['fixed acidity', 'residual sugar', 'free sulfur dioxide']]
df.head()
```

	fixed acidity	residual sugar	free sulfur dioxide
0	7.4	1.9	11.0
1	7.8	2.6	25.0
2	7.8	2.3	15.0
3	11.2	1.9	17.0
4	7.4	1.9	11.0

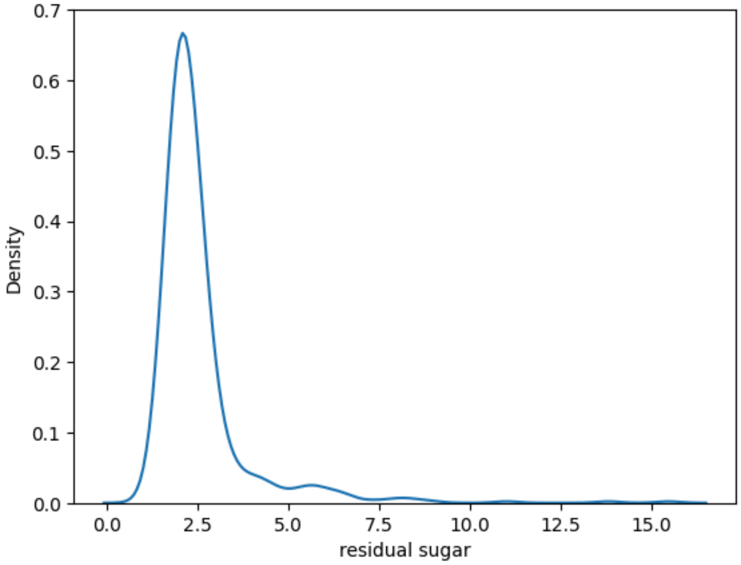
```
sns.kdeplot(df['fixed acidity'])
```

```
<Axes: xlabel='fixed acidity', ylabel='Density'>
```



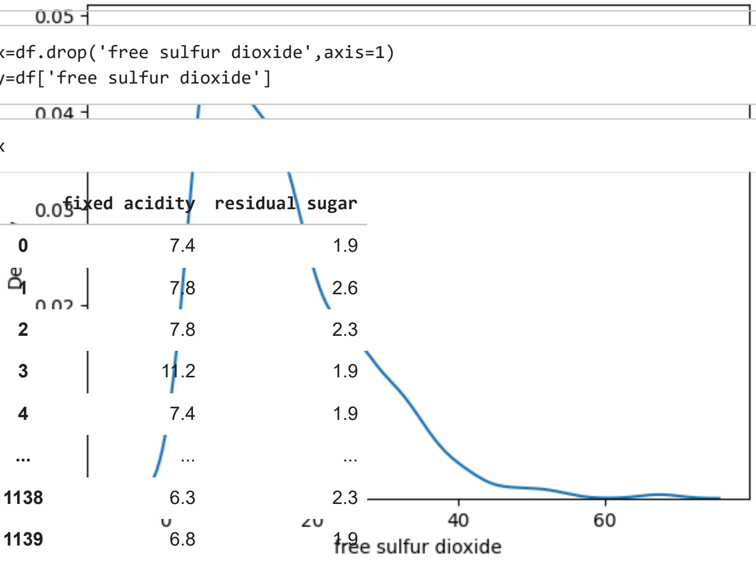
```
sns.kdeplot(df['residual sugar'])
```

```
<Axes: xlabel='residual sugar', ylabel='Density'>
```



```
sns.kdeplot(df['free sulfur dioxide'])
```

```
<Axes: xlabel='free sulfur dioxide', ylabel='Density'>
```



```
x=df.drop('free sulfur dioxide',axis=1)
y=df['free sulfur dioxide']
```

```
x
```

	fixed acidity	residual sugar
0	7.4	1.9
1	7.8	2.6
2	7.8	2.3
3	11.2	1.9
4	7.4	1.9
...	...	...
1138	6.3	2.3
1139	6.8	1.9
1140	6.2	2.0
1141	5.9	2.2
1142	5.9	2.0

1143 rows × 2 columns

```
y
```

	free sulfur dioxide
0	11.0
1	25.0
2	15.0
3	17.0
4	11.0
...	...
1138	29.0
1139	28.0
1140	32.0
1141	39.0
1142	32.0

1143 rows × 1 columns

dtype: float64

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3)

x_train.shape, x_test.shape
```

```
((800, 2), (343, 2))
```

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

# fit the scaler to the train set, it will learn the parameters
scaler.fit(x_train)

# transform train and test sets
x_train_scaled = scaler.transform(x_train)
x_test_scaled = scaler.transform(x_test)
```

```
x_train_scaled = pd.DataFrame(x_train_scaled, columns=x_train.columns)
x_test_scaled = pd.DataFrame(x_test_scaled, columns=x_test.columns)
```

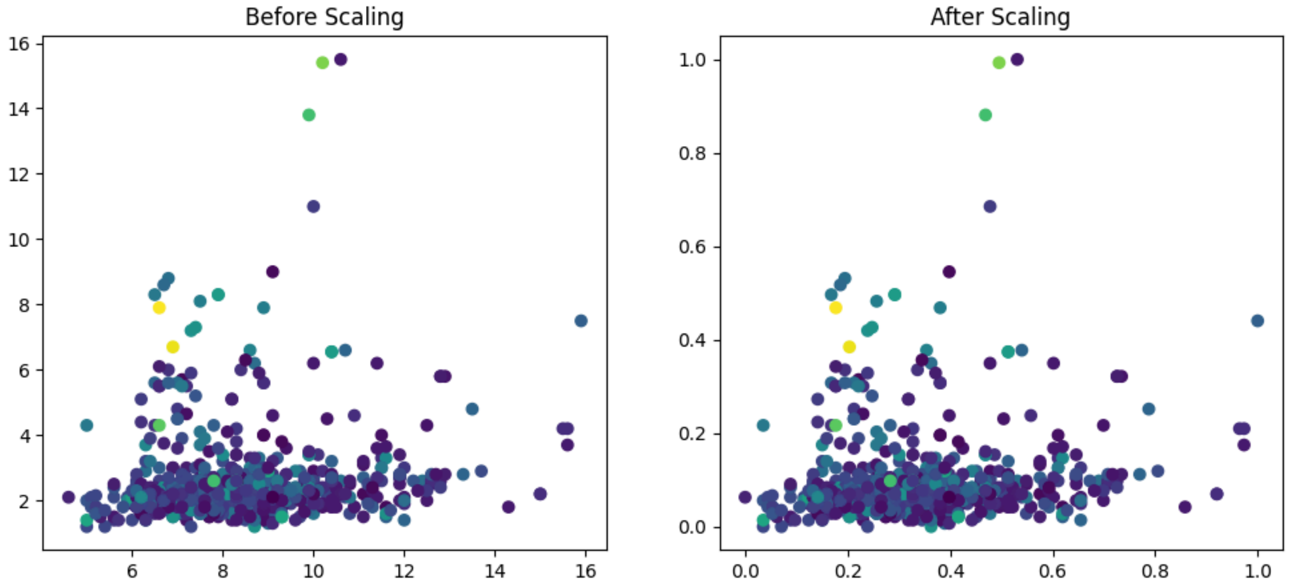
```
x_train.describe()
```

	fixed acidity	residual sugar
count	800.000000	800.000000
mean	8.300000	2.554688
std	1.741858	1.403507
min	4.600000	1.200000
25%	7.100000	1.900000
50%	7.900000	2.200000
75%	9.100000	2.600000
max	15.900000	15.500000

```
x_train_scaled.describe()
```

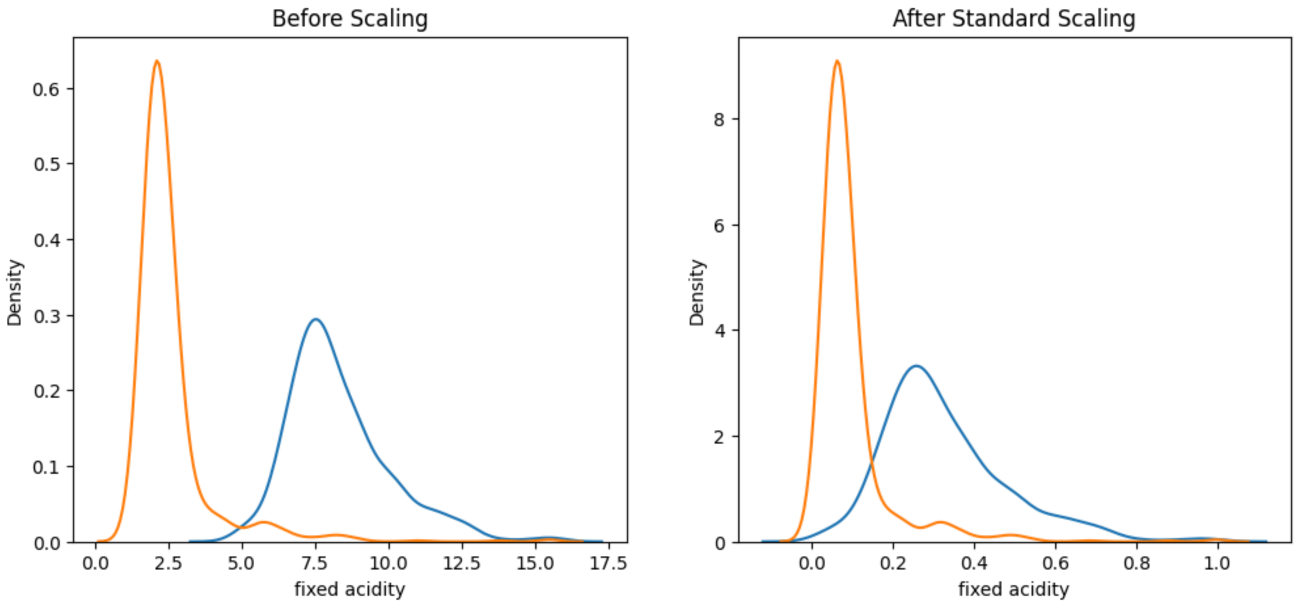
	fixed acidity	residual sugar
count	800.000000	800.000000
mean	0.327434	0.094733
std	0.154147	0.098147
min	0.000000	0.000000
25%	0.221239	0.048951
50%	0.292035	0.069930
75%	0.398230	0.097902
max	1.000000	1.000000

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))

ax1.scatter(x_train['fixed acidity'], x_train['residual sugar'],c=y_train)
ax1.set_title("Before Scaling")
ax2.scatter(x_train_scaled['fixed acidity'], x_train_scaled['residual sugar'],c=y_train)
ax2.set_title("After Scaling")
plt.show()
```

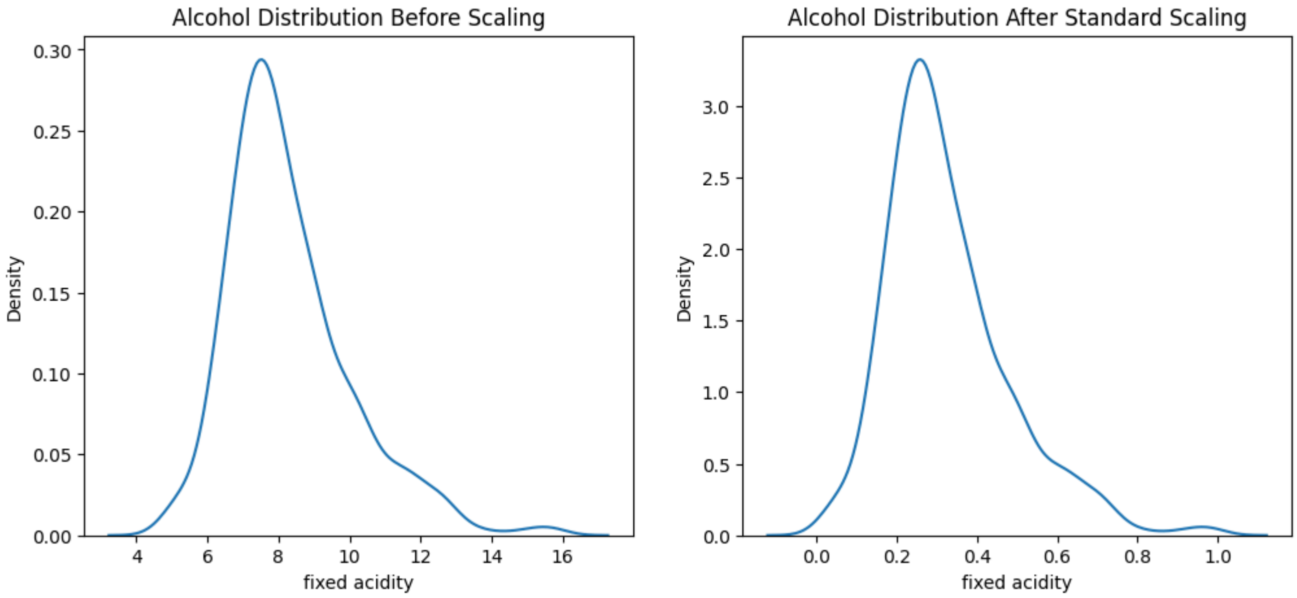


```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))
```

```
# before scaling
ax1.set_title('Before Scaling')
sns.kdeplot(x_train['fixed acidity'], ax=ax1)
sns.kdeplot(x_train['residual sugar'], ax=ax1)

# after scaling
ax2.set_title('After Standard Scaling')
sns.kdeplot(x_train_scaled['fixed acidity'], ax=ax2)
sns.kdeplot(x_train_scaled['residual sugar'], ax=ax2)
plt.show()
```

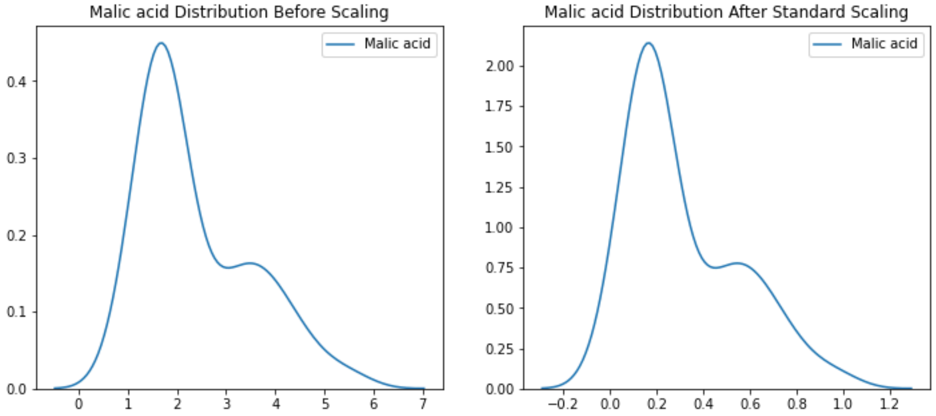


```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))

# before scaling
ax1.set_title('Alcohol Distribution Before Scaling')
sns.kdeplot(x_train['fixed acidity'], ax=ax1)

# after scaling
ax2.set_title('Alcohol Distribution After Standard Scaling')
sns.kdeplot(x_train_scaled['fixed acidity'], ax=ax2)
plt.show()
```



```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))
```

```
# before scaling
ax1.set_title('Malic acid Distribution Before Scaling')
sns.kdeplot(x_train['residual sugar'], ax=ax1)

# after scaling
ax2.set_title('Malic acid Distribution After Standard Scaling')
sns.kdeplot(x_train_scaled['residual sugar'], ax=ax2)
plt.show()
```



Start coding or generate with AI.