



```
import pandas as pd
import numpy as np

np.random.seed(42)
data = {
    'Station_ID': range(1, 31),
    'Temperature': np.random.normal(25, 5, 30),
    'Humidity': np.random.uniform(40, 90, 30),
    'WindSpeed': np.random.uniform(5, 25, 30),
    'Rainfall': np.random.uniform(0, 200, 30)
}
df = pd.DataFrame(data)
df.head()
```

	Station_ID	Temperature	Humidity	WindSpeed	Rainfall	
0	1	27.483571	74.211651	6.491013	5.083825	
1	2	24.308678	62.007625	24.737739	21.578285	
2	3	28.238443	46.101912	20.444895	6.285837	
3	4	32.615149	64.758846	8.974314	127.282082	
4	5	23.829233	41.719426	5.110442	62.871196	

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
from sklearn.preprocessing import StandardScaler

df = df.dropna()
X = df[['Temperature', 'Humidity', 'WindSpeed', 'Rainfall']]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
from sklearn.cluster import DBSCAN

db = DBSCAN(eps=1.2, min_samples=3)
labels = db.fit_predict(X_scaled)
df['Cluster'] = labels
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8,6))
```

```
plt.scatter(df['Temperature'], df['Humidity'], c=df['Cluster'], cmap='viridis')
outliers = df[df['Cluster'] == -1]
plt.scatter(outliers['Temperature'], outliers['Humidity'], color='red', marker='x', s=100)
plt.title('DBSCAN Clustering of Weather Stations')
plt.xlabel('Temperature')
plt.ylabel('Humidity')
plt.show()
```

