Pumpkin Price Dataset

Import Libraries

```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import LinearRegression
```

Import Data

```
# Get dataset
df_wth = pd.read_csv('/content/weather1.csv')
df_wth.head()
```

	Pressure (millibars)	Humidity
0	1014.40	0.62
1	1014.20	0.66
2	1014.47	0.79
3	1014.45	0.82
4	1014.49	0.83

Next steps: Generate code with df_wth View recommended plots

Analyze the Data

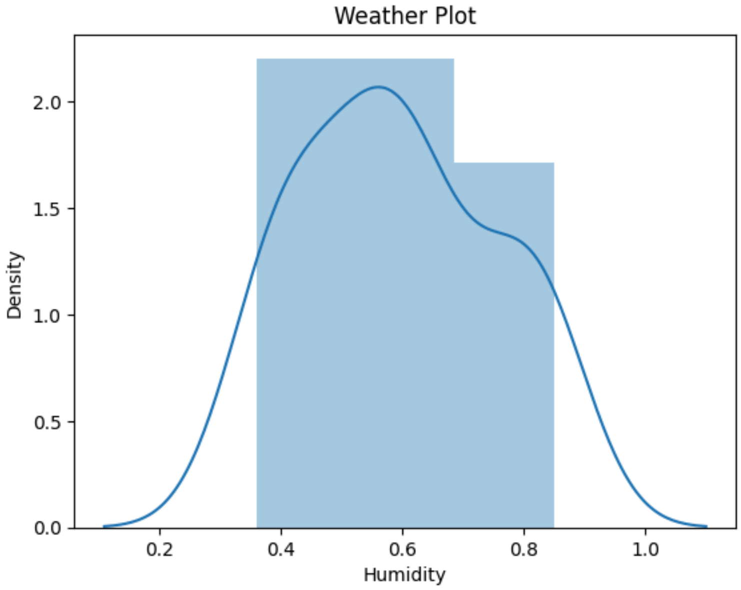
```
# Describe data
df_wth.describe()
```

	Pressure (millibars)	Humidity
count	25.000000	25.0000
mean	1011.481600	0.5932
std	2.873799	0.1590
min	1007.260000	0.3600
25%	1008.360000	0.4600
50%	1012.220000	0.5900
75%	1014.240000	0.7200
max	1014.520000	0.8500

Distribution

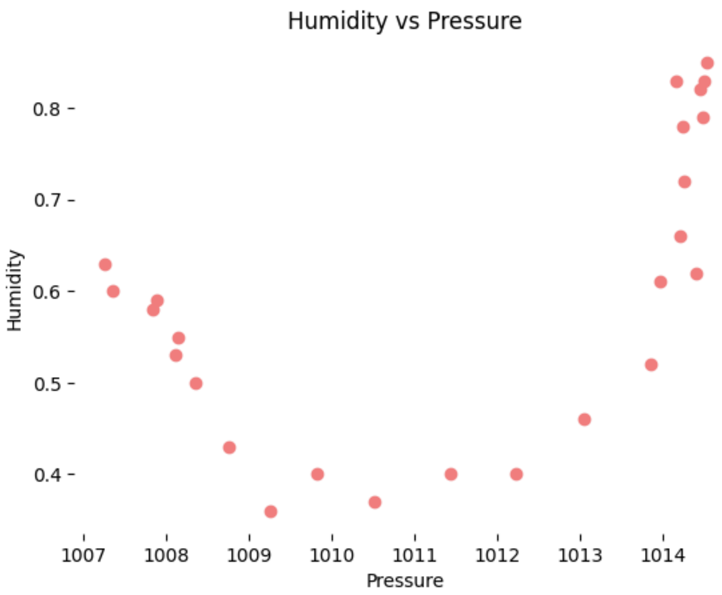
```
# Data distribution
plt.title('Weather Plot')
sns.distplot(df_wth['Humidity'])
plt.show()
```

```
<ipython-input-14-c49fa2e4408f>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_wth['Humidity'])
```



Relation between Pressure and Humidity

```
# Relationship between Pressure and Humidity
plt.scatter(df_sal['Pressure (millibars)'], df_sal['Humidity'], color = 'lightcoral')
plt.title('Humidity vs Pressure')
plt.xlabel('Pressure')
plt.ylabel('Humidity')
plt.box(False)
plt.show()
```



Split data into Independent/Dependent variables

```
# Splitting variables
X = df_wth.iloc[:, 0:-1].values   # independent variables
y = df_wth.iloc[:, -1].values     # dependent variable
```

Train model

Linear Regression

```
# Train linear regression model on whole dataset
lr = LinearRegression()
lr.fit(X, y)
```

```
▾ LinearRegression
LinearRegression()
```

Polynomial Regression

```
# Train polynomial regression model on the whole dataset
pr = PolynomialFeatures(degree = 4)
X_poly = pr.fit_transform(X)
lr_2 = LinearRegression()
lr_2.fit(X_poly, y)
```

```
▾ LinearRegression
LinearRegression()
```

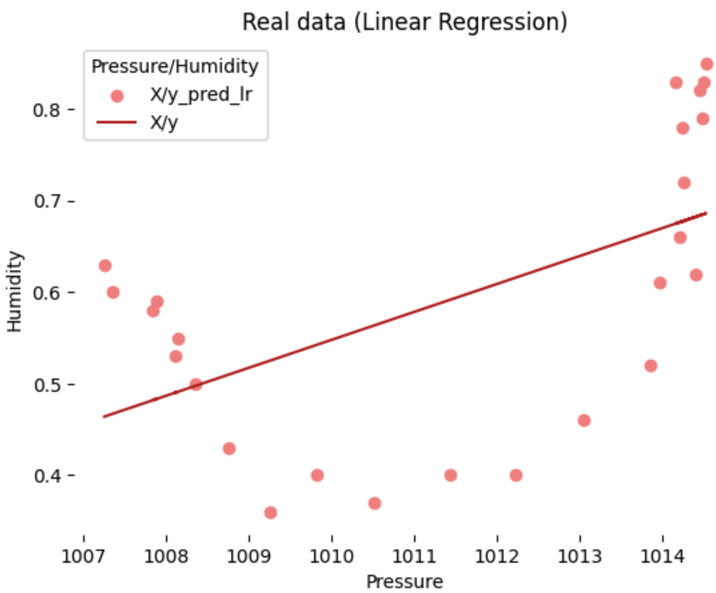
Predict results

```
# Predict results
y_pred_lr = lr.predict(X)           # Linear Regression
y_pred_poly = lr_2.predict(X_poly)  # Polynomial Regression
```

Visualize predictions

Prediction with Linear Regression

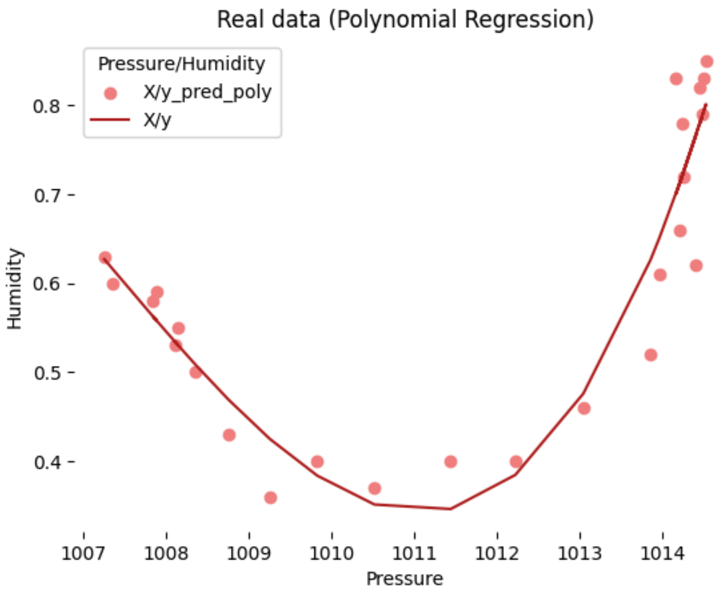
```
# Visualize real data with linear regression
plt.scatter(X, y, color = 'lightcoral')
plt.plot(X, lr.predict(X), color = 'firebrick')
plt.title('Real data (Linear Regression)')
plt.xlabel('Pressure')
plt.ylabel('Humidity')
plt.legend(['X/y_pred_lr', 'X/y'], title = 'Pressure/Humidity', loc='best', facecolor='white')
plt.box(False)
plt.show()
```



Prediction with Polynomial Regression

```
# Visualize real data with polynomial regression
X_grid = np.arange(min(X), max(X), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'lightcoral')
plt.plot(X, lr_2.predict(X_poly), color = 'firebrick')
plt.title('Real data (Polynomial Regression)')
plt.xlabel('Pressure')
plt.ylabel('Humidity')
plt.legend(['X/y_pred_poly', 'X/y'], title = 'Pressure/Humidity', loc='best', facecolor='white')
plt.box(False)
plt.show()
```

```
<ipython-input-28-61ee569b0a27>:2: DeprecationWarning: Conversion of an array with nd
  X_grid = np.arange(min(X), max(X), 0.1)
```



Test with an example

```
# Predict a new result with linear regression
print(f'Linear Regression result : {lr.predict([[6.5]])}')
# Predict a new result with polynomial regression
print(f'Polynomial Regression result : {lr_2.predict(pr.fit_transform([[6.5]]))}')
```

```
Linear Regression result : [-30.13472872]
Polynomial Regression result : [-675201.81699724]
```

Conclusion : the linear regression model is able to explain much more of the variance in the data than the polynomial regression model. Therefore, it is more likely that the linear regression model will make accurate predictions on new data.