# week2

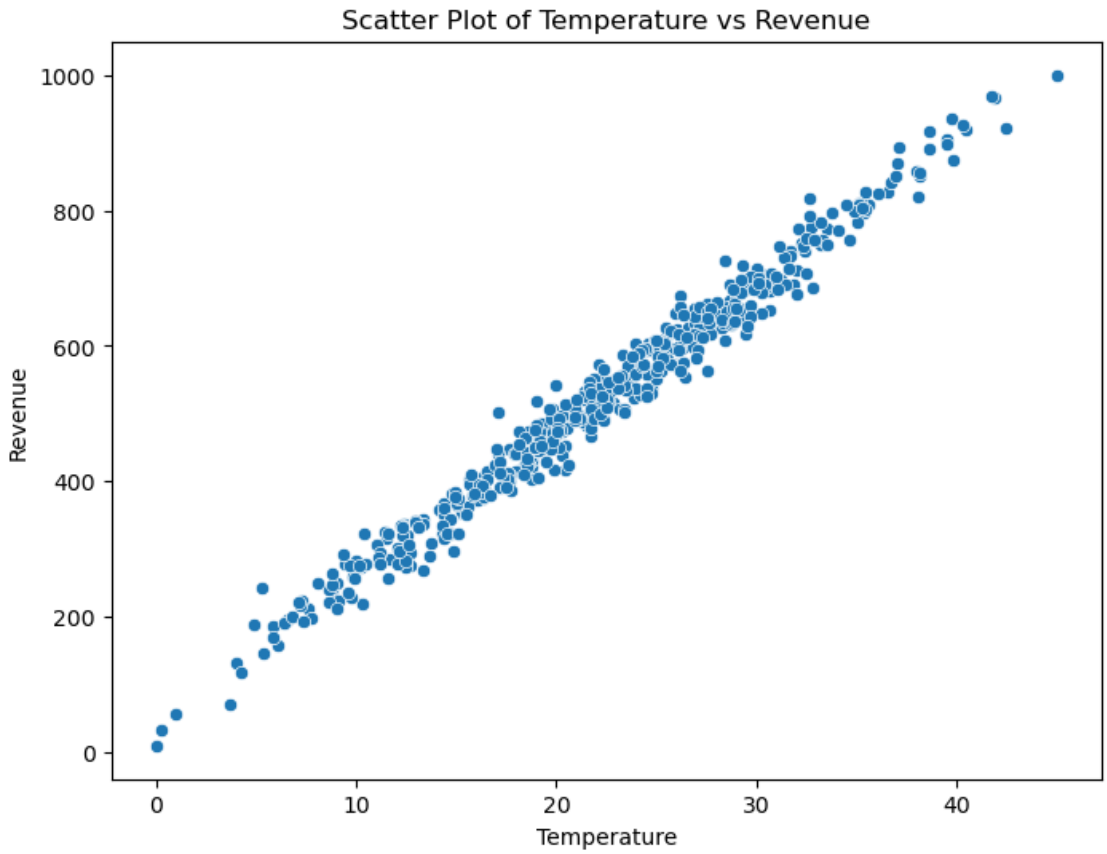
August 16, 2024

[25]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

file_path = 'IceCreamData.csv'
ice_cream_data = pd.read_csv(file_path)

plt.figure(figsize=(8,6))
sns.scatterplot(x='Temperature', y='Revenue', data=ice_cream_data)
plt.title('Scatter Plot of Temperature vs Revenue')
plt.xlabel('Temperature')
plt.ylabel('Revenue')
plt.show()
```

```
c:\Users\hp\anaconda3\envs\env\Lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed in a
future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\hp\anaconda3\envs\env\Lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed in a
future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```




[26]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import linear_model
import statsmodels.api as sm

file_path = 'IceCreamData.csv'
ice_cream_data = pd.read_csv(file_path)

x=ice_cream_data['Temperature']
y=ice_cream_data['Revenue']
x=sm.add_constant(x)
modal=sm.OLS(y,x).fit()
print(modal.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Revenue   R-squared:                       0.980
Model:                            OLS   Adj. R-squared:                  0.980
Method:                 Least Squares   F-statistic:                 2.404e+04
```

```
Date:                Fri, 16 Aug 2024   Prob (F-statistic):               0.00
Time:                        10:36:34   Log-Likelihood:                -2318.1
No. Observations:                 500   AIC:                             4640.
Df Residuals:                     498   BIC:                             4649.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          44.8313      3.272     13.703      0.000      38.403      51.259
Temperature    21.4436      0.138    155.057      0.000      21.172      21.715
==============================================================================
Omnibus:                        3.595   Durbin-Watson:                   1.985
Prob(Omnibus):                  0.166   Jarque-Bera (JB):                4.192
Skew:                           0.032   Prob(JB):                        0.123
Kurtosis:                       3.444   Cond. No.                         69.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[32]:
```python
import pandas as pd
import matplotlib.pyplot as plt

file_name = 'IceCreamData.csv'
data = pd.read_csv(file_name)

x = data['Temperature']
y = data['Revenue']

x_mean = x.mean()
y_mean = y.mean()

numerator = ((x - x_mean) * (y - y_mean)).sum()
denominator = ((x - x_mean) ** 2).sum()
slope = numerator / denominator

intercept = y_mean - slope * x_mean

print(f"Intercept: {intercept}")
print(f"Slope: {slope}")

import matplotlib.pyplot as plt

plt.scatter(x, y, label='Data Points')
regression_line = intercept + slope * x
```

```
plt.plot(x, regression_line, color='orange', label='Regression Line')
plt.xlabel('Temperature')
plt.ylabel('Revenue')
plt.title('Linear Regression: Temperature vs Revenue')
plt.legend()
plt.show()
```

```
Intercept: 44.831267090563585
Slope: 21.44362551068026
```