In [20]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [21]:

```
sales_df = pd.read_csv(r"C:\Users\Maj Mortuza\Downloads\IceCreamData.csv")
sales_df
```

Out[21]:

| | Temperature | Revenue |
|---|---|---|
| **0** | 24.566884 | 534.799028 |
| **1** | 26.005191 | 625.190122 |
| **2** | 27.790554 | 660.632289 |
| **3** | 20.595335 | 487.706960 |
| **4** | 11.503498 | 316.240194 |
| **...** | ... | ... |
| **495** | 22.274899 | 524.746364 |
| **496** | 32.893092 | 755.818399 |
| **497** | 12.588157 | 306.090719 |
| **498** | 22.362402 | 566.217304 |
| **499** | 28.957736 | 655.660388 |

500 rows × 2 columns

In [22]:

```
sales_df.head(10)
```

Out[22]:

| | Temperature | Revenue |
|---|---|---|
| **0** | 24.566884 | 534.799028 |
| **1** | 26.005191 | 625.190122 |
| **2** | 27.790554 | 660.632289 |
| **3** | 20.595335 | 487.706960 |
| **4** | 11.503498 | 316.240194 |
| **5** | 14.352514 | 367.940744 |
| **6** | 13.707780 | 308.894518 |
| **7** | 30.833985 | 696.716640 |
| **8** | 0.976870 | 55.390338 |
| **9** | 31.669465 | 737.800824 |

In [23]:

```
sales_df.tail(10)
```

Out[23]:

| | Temperature | Revenue |
|---|---|---|
| **490** | 23.824922 | 584.399945 |
| **491** | 34.472169 | 809.352519 |
| **492** | 23.056214 | 552.819351 |
| **493** | 14.931506 | 377.430928 |
| **494** | 25.112066 | 571.434257 |
| **495** | 22.274899 | 524.746364 |
| **496** | 32.893092 | 755.818399 |
| **497** | 12.588157 | 306.090719 |
| **498** | 22.362402 | 566.217304 |
| **499** | 28.957736 | 655.660388 |

In [24]:

```
sales_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Temperature  500 non-null    float64
 1   Revenue      500 non-null    float64
dtypes: float64(2)
memory usage: 7.9 KB
```

In [25]:

```
sales_df.describe()
```

Out[25]:

| | Temperature | Revenue |
|---|---|---|
| **count** | 500.000000 | 500.000000 |
| **mean** | 22.232225 | 521.570777 |
| **std** | 8.096388 | 175.404751 |
| **min** | 0.000000 | 10.000000 |
| **25%** | 17.122258 | 405.558681 |
| **50%** | 22.392791 | 529.368565 |
| **75%** | 27.740674 | 642.257922 |
| **max** | 45.000000 | 1000.000000 |

In [26]:

```
sales_df.mean()
```

Out[26]:

```
Temperature     22.232225
Revenue        521.570777
dtype: float64
```

In [27]:

```
sales_df.max()
```

Out[27]:

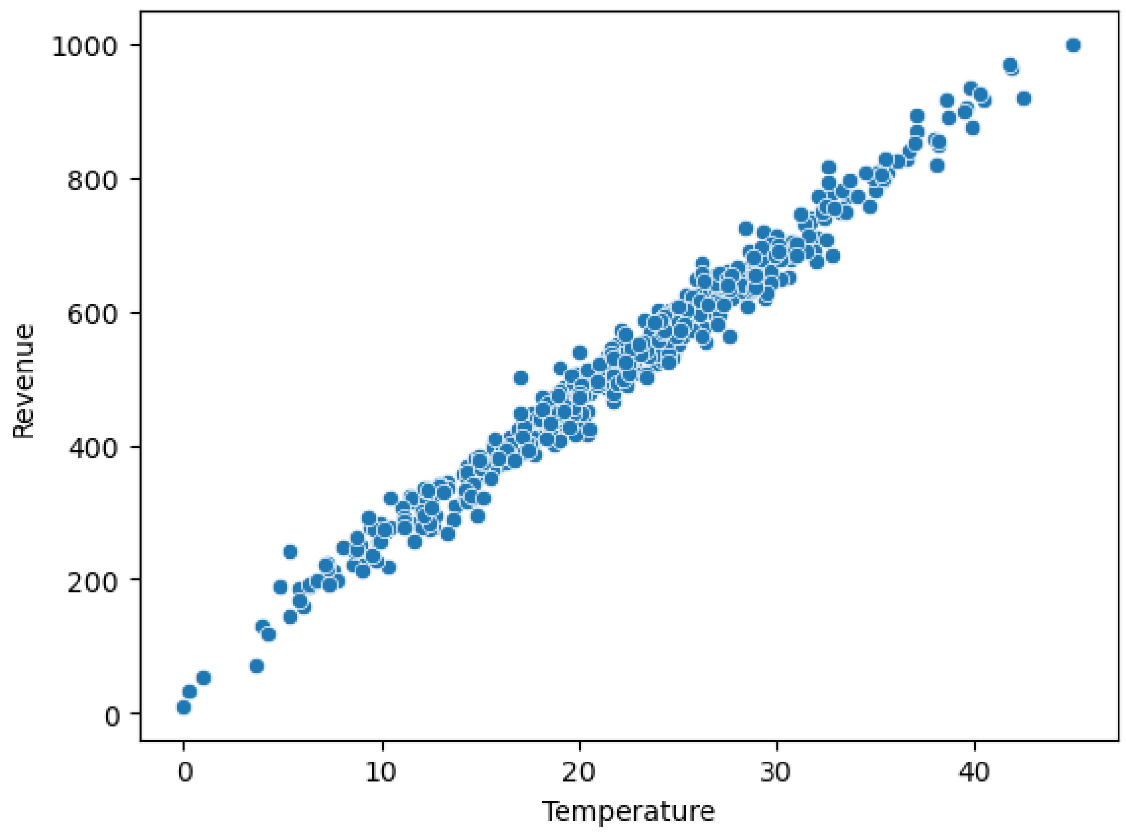
```
Temperature      45.0
Revenue        1000.0
dtype: float64
```

In [30]:

```
sns.scatterplot(x = 'Temperature', y = 'Revenue', data = sales_df)
```

Out[30]:

```
<Axes: xlabel='Temperature', ylabel='Revenue'>
```

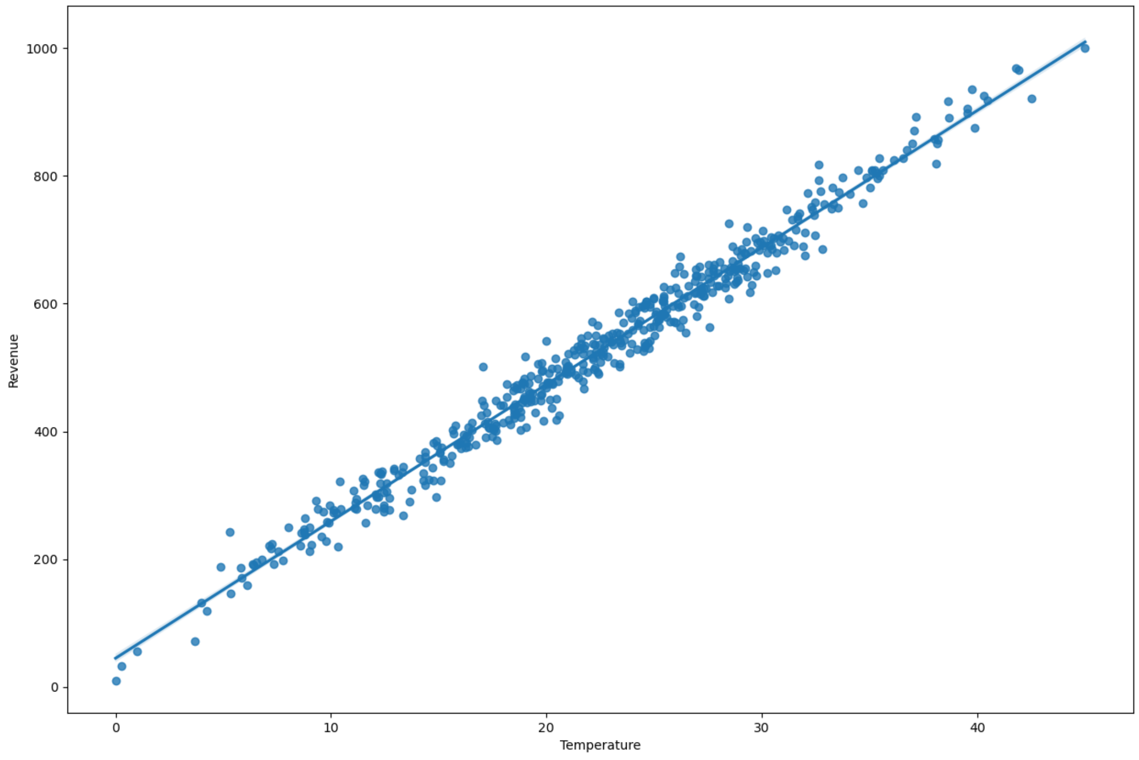


In [18]:

```
plt.figure(figsize = (15, 10))
sns.regplot(x = 'Temperature', y = 'Revenue', data = sales_df)
```

Out[18]:

```
<Axes: xlabel='Temperature', ylabel='Revenue'>
```



In [31]:

```
#To eveluate the model, test & train the dataset
X = sales_df['Temperature']
y = sales_df['Revenue']
```

In [32]:

```
#Convert x & y into numpy array
X = np.array(X)
y = np.array(y)
```

In [35]:

```
#Reshape the subsets
X = X.reshape(-1, 1)
print(X.shape)

y = y.reshape(-1, 1)
print(y.shape)
```

```
(500, 1)
(500, 1)
```

In [36]:

```
#Split data into train and test for 80% and 20% respectively
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2)
```

In [37]:

```
X_train.shape
```

Out[37]:

```
(400, 1)
```

In [38]:

```
y_train.shape
```

Out[38]:

```
(400, 1)
```

In [39]:

```
X_test.shape
```

Out[39]:

```
(100, 1)
```

In [40]:

```
X
y_test.shape
```

Out[40]:

```
(100, 1)
```

In [41]:

```
#Train the dataset
from sklearn.linear_model import LinearRegression
SimpleLinearRegression = LinearRegression(fit_intercept = True)
SimpleLinearRegression.fit(X_train, y_train)
```

Out[41]:

```
LinearRegression()
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**
**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

In [45]:

```
print('Linear Model Coeff(m)', SimpleLinearRegression.coef_)
print('Linear Model Coeff(b)', SimpleLinearRegression.intercept_)
```

```
Linear Model Coeff(m) [[21.43896001]]
Linear Model Coeff(b) [44.7200979]
```

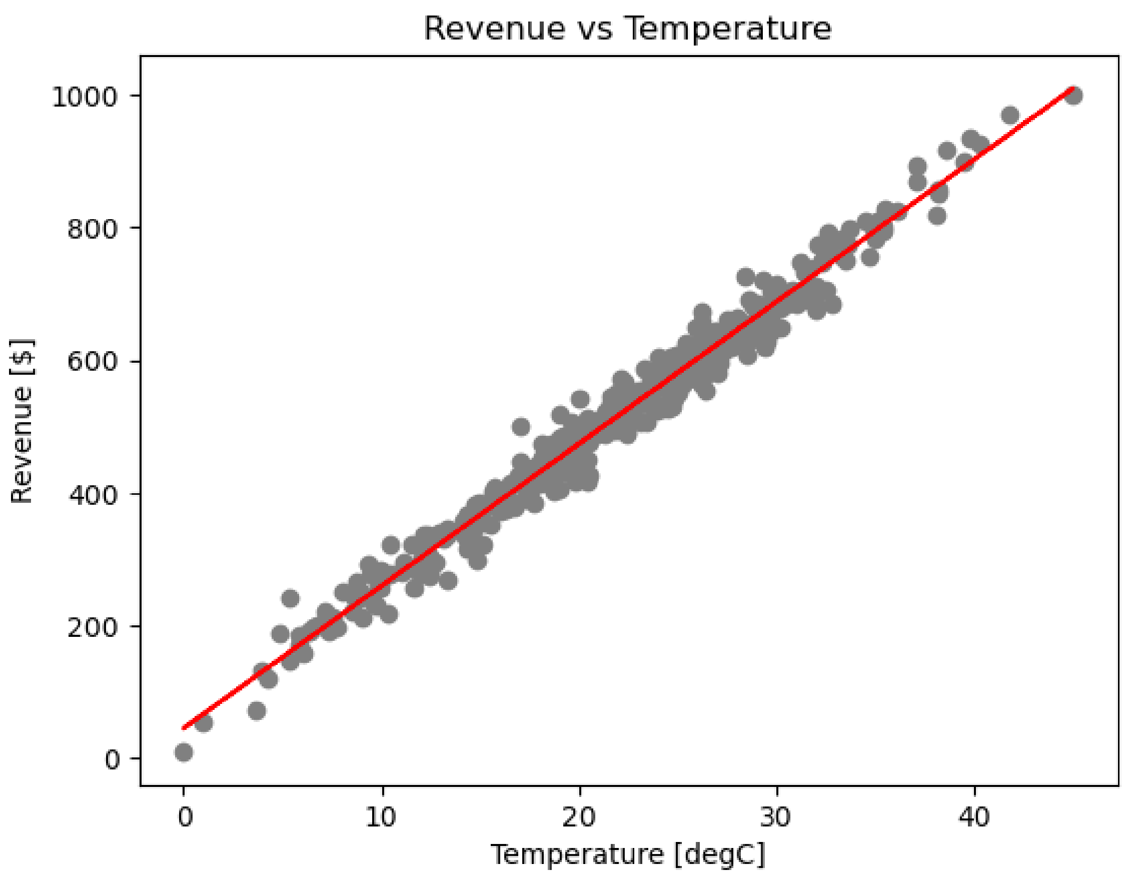
In [55]:

```
#Evaluate the model

plt.scatter(X_train, y_train, color = 'gray')
plt.plot(X_train, SimpleLinearRegression.predict(X_train), color = 'red')
plt.ylabel('Revenue [$]')
plt.xlabel('Temperature [degC]')
plt.title('Revenue vs Temperature')
```

Out[55]:

```
Text(0.5, 1.0, 'Revenue vs Temperature')
```



In [57]:

```
accuracy_LinearRegression = SimpleLinearRegression.score(X_test, y_test)
accuracy_LinearRegression
```

Out[57]:

```
0.9839719651993599
```

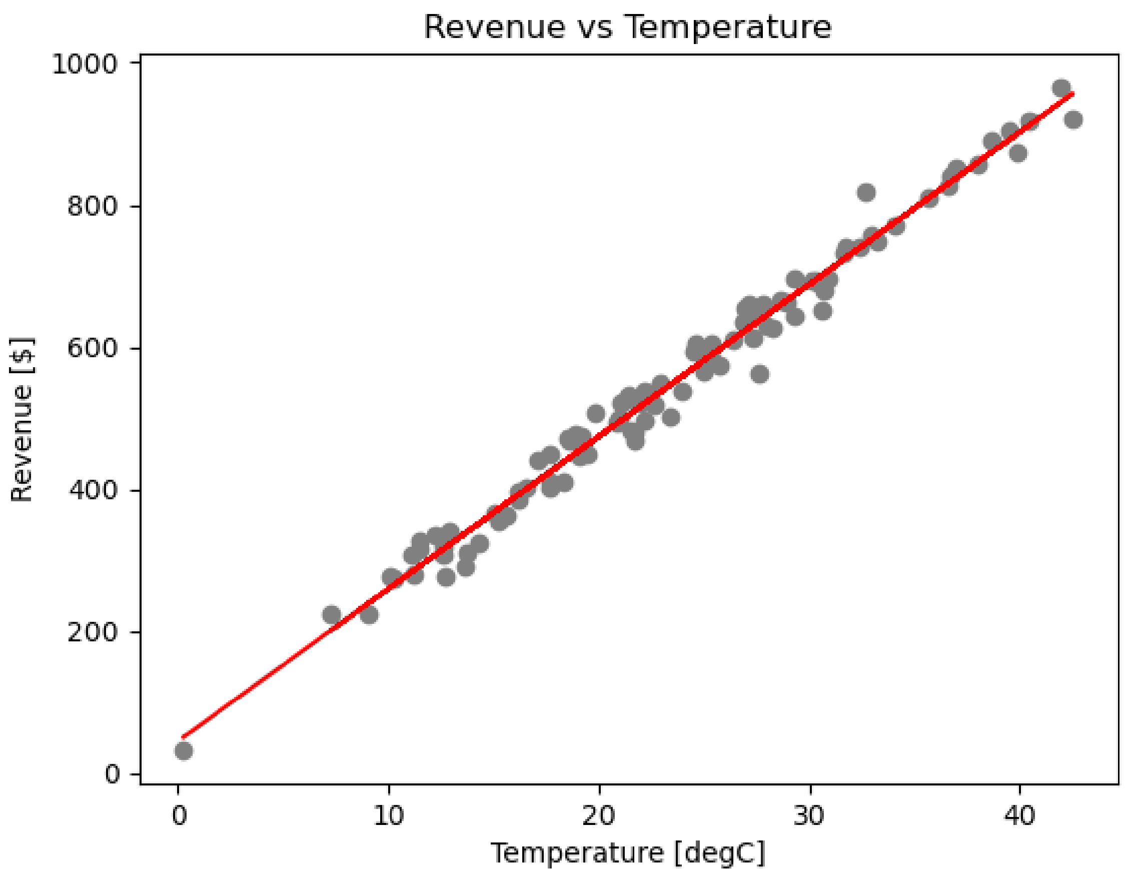
In [59]:

```
#Now apply it to test data

plt.scatter(X_test, y_test, color = 'gray')
plt.plot(X_test, SimpleLinearRegression.predict(X_test), color = 'red')
plt.ylabel('Revenue [$]')
plt.xlabel('Temperature [degC]')
plt.title('Revenue vs Temperature')
```

Out[59]:

```
Text(0.5, 1.0, 'Revenue vs Temperature')
```



In [63]:

```
#Using the trained model to generate predictions

Temp = np.array([20])
Temp = Temp.reshape(-1, 1)

Revenue = SimpleLinearRegression.predict(Temp)
print('Revenue Predictions =', Revenue)
```

```
Revenue Predictions = [[473.49929806]]
```