Name - Omkar Phansopkar
UID - 2022701007

Title - Exp 10. Linear Regression

Using Excel

	Regression	
	Regression Model	Linear
	LINEST raw output	
	21.4436255106803	44.8312670905636
	0.138294968318454	3.27175680838505
	0.979707200117581	25.0119723401289
	24042.7239457108	498
	15041098.2957147	311548.182651
	Regression Statistics	
	R^2	0.979707200117581
	Standard Error	25.0119723401289
	Count of X variables	1
	Observations	500
	Adjusted R^2	0.97966645152344

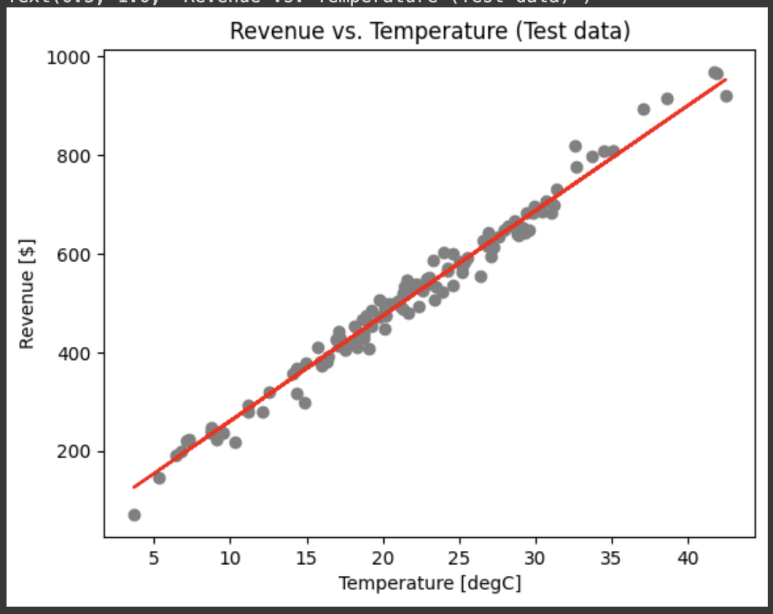
D	E	F	G	H	I	J	K
Confidence level	0.95						
	Coefficients	Standard Error	t-Statistic	P-value	Lower 95%	Upper 95%	
Intercept	44.8312670905636	3.27175680838505	13.70250594	1.72016E-36	38.40311895	Lower 0%	Upper 0%
Temperature	21.4436255106803	0.138294968318454	155.0571635	0	21.171912	Err:502	Err:502
						Err:502	Err:502
Temperature	Predicted Y	Revenue	Residual				
24.56688442	571.634336557209	534.7990284	-36.83530816				
26.00519115	602.47684744482	625.1901215	22.71327406				
27.79055388	640.761497227666	660.6322888	19.87079157				
20.59533505	486.469919169751	487.7069603	1.23704113				
11.50349764	291.507962545718	316.2401944	24.73223185				
14.35251388	352.601199870124	367.9407438	15.33954393				
13.70777988	338.775765420121	308.8945179	-29.88124752				
30.83398474	706.023688857153	696.7166402	-9.307048657				
0.976869989	65.7789013073019	55.39033824	-10.38856307				
31.66946458	723.939405667837	737.8008241	13.86141843				
11.45525338	290.473430701238	325.9684084	35.4949777				
3.664669577	123.415069120135	71.16015301	-52.25491611				
18.81182403	448.2249767627	467.4467066	19.22172984				
13.62450892	336.990134137966	289.5409341	-47.44920004				
39.53990899	892.710268198503	905.4776043	12.7673361				
18.48314099	441.176820741228	469.9090332	28.73221246				
25.93537514	600.97973907173	648.2099977	47.23025863				
42.51528041	956.513018684164	921.508275	-35.00474368				
29.58948056	679.337007274757	649.5611747	-29.77583257				
21.77594799	511.786540928174	534.6228653	22.83632437				
25.45783637	590.739576521019	612.1539491	21.41437258				

IceCreamData

Using sklearn

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

```
Text(0.5, 1.0, 'Revenue vs. Temperature (Test data)')
```




```
[49] print("SSE", np.sum((y_test - y_predict) ** 2))
     print("SST", np.sum((y_test - np.mean(y_test)) ** 2))
     print("MSE", mean_squared_error(y_test, y_predict))
     print("RMSE", np.sqrt(mean_squared_error(y_test, y_predict)))
     print("MAE", mean_absolute_error(y_test, y_predict))
     print("R^2", r2_score(y_test, y_predict))
```

0s

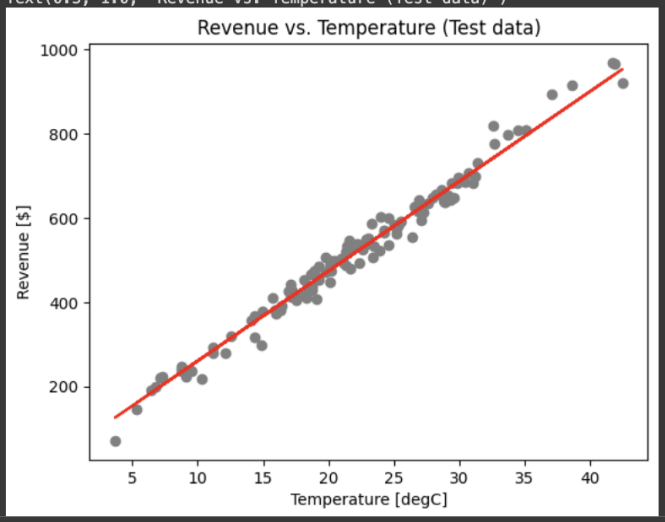
```
SSE 70396.14752312287
SST 3767647.8957343353
MSE 563.1691801849829
RMSE 23.731185814977366
MAE 18.748523562719573
R^2 0.981315624636043
```

Using own model

```python
class OwnLinearRegression:
    def __init__(self):
        self.coef_ = None
        self.intercept_ = None

    def fit(self, X, y):
        ones = np.ones((X.shape[0], 1))
        X = np.hstack((ones, X))

        X_transpose = np.transpose(X)
        X_transpose_dot_X = X_transpose.dot(X)
        X_transpose_dot_y = X_transpose.dot(y)
        coefficients = np.linalg.solve(X_transpose_dot_X, X_transpose_dot_y)

        self.intercept_ = coefficients[0]
        self.coef_ = coefficients[1:]

    def predict(self, X):
        if self.coef_ is None or self.intercept_ is None:
            raise Exception("Fit the model first !!!")
        return X.dot(self.coef_) + self.intercept_
```

```python
model_own = OwnLinearRegression()
model_own.fit(X_train, y_train)
y_predict_own = model.predict(X_test)
plt.scatter(X_test, y_test, color = 'gray')
plt.plot(X_test, y_predict_own, color = 'red')
plt.ylabel('Revenue [$]')
plt.xlabel('Temperature [degC]')
plt.title('Revenue vs. Temperature (Test data)')
```

```
Text(0.5, 1.0, 'Revenue vs. Temperature (Test data)')
```



0s completed at 1

```python
print("SSE", np.sum((y_test - y_predict_own) ** 2))
print("SST", np.sum((y_test - np.mean(y_test)) ** 2))
print("MSE", mean_squared_error(y_test, y_predict_own))
print("RMSE", np.sqrt(mean_squared_error(y_test, y_predict_own)))
print("MAE", mean_absolute_error(y_test, y_predict_own))
print("R^2", r2_score(y_test, y_predict_own))
```

```
SSE 70396.14752312208
SST 3767647.8957343353
MSE 563.1691801849767
RMSE 23.731185814977234
MAE 18.74852356271954
R^2 0.9813156246360433
```