



---

## **Practical 5. Develop a cost function of linear regression using sample dataset..**

### **Code:**

```
import numpy as np
import matplotlib.pyplot as plt
from typing_extensions import final
X = np.array([1, 2, 3, 4, 5])
Y = np.array([1, 2, 3, 4, 5])
plt.scatter(X, Y)
m = len(Y)
theta0 = 0.0
theta1 = 0.0
def hypothesis(X, theta0, theta1):
    return theta0 + theta1 * X
def cost_function(theta0, theta1, X, Y):
    total_cost = 0
    for i in range(m):
        total_cost += (hypothesis(X[i], theta0, theta1) - Y[i]) ** 2
    return total_cost / (2 * m)
theta1_values = np.linspace(-10, 10, 100)
cost_values = []
for theta1 in theta1_values:
    cost = cost_function(theta0, theta1, X, Y)
    cost_values.append(cost)
plt.plot(theta1_values, cost_values, label='Cost
Function')
plt.xlabel('Theta1')
plt.ylabel('Cost')
plt.title('Cost Function For Linear Regression ')
plt.legend()
```



---

```
plt.show()

initial_cost = cost_function(theta0, theta1, X, Y)
print(f"Initial Cost(MSE) with theta0 = {theta0} , and theta1 = {theta1} :
{initial_cost}")

theta0_exp = 0.2
theta1_exp = 0.8

final_cost = cost_function(theta0_exp, theta1_exp, X, Y)
print(f"Final Cost: {final_cost}")
print(f"Final Cost (MSE) with theta0 = {theta0} and theta1 = {theta1}: {final_cost}")
```

**Output:**

