

## Code6

October 23, 2023

```
[1]: # Function to calculate the mean
def calculate_mean(data):
    if not data:
        return 0
    return sum(data) / len(data)

# Function to calculate the slope and intercept
def calculate_regression_coefficients(data_points):
    x_values, y_values = zip(*data_points)
    x_mean = calculate_mean(x_values)
    y_mean = calculate_mean(y_values)
    numerator = sum((x - x_mean) * (y - y_mean) for x, y in data_points)
    denominator = sum((x - x_mean) ** 2 for x in x_values)
    slope = numerator / denominator
    intercept = y_mean - slope * x_mean
    return slope, intercept

# Function to predict
def make_predictions(data_points, slope, intercept):
    x_values = [x for x, _ in data_points]
    predicted_y_values = [slope * x + intercept for x in x_values]
    return predicted_y_values

# data points
data_points = [
    (2000, 72.5),
    (2001, 73.1),
    (2002, 73.8),
    # ... (other data points)
    (2018, 82.0),
    (2019, 82.5)
]

# Calculate regression coefficients
slope, intercept = calculate_regression_coefficients(data_points)

# Print the slope and intercept
```

```

print("Regression Coefficients:")
print(f"Slope (m): {slope}")
print(f"Intercept (b): {intercept}")

# Prompt the user to enter a new x value
new_x = float(input("\nEnter a new x value for prediction: "))

# Use the function to predict the corresponding y value
predicted_y = slope * new_x + intercept

# Display the predicted y
print(f"Predicted y value: {predicted_y}")

```

```

Regression Coefficients:
Slope (m): 0.5216216216216217
Intercept (b): -970.6362162162166

```

```

Enter a new x value for prediction: 6
Predicted y value: -967.5064864864868

```