Name: Guilherme dos Santos

Simple linear regression is a statistical method used to model the relationship between two variables: one independent variable (often denoted as X) and one dependent variable (often denoted as Y). The goal is to find a linear equation that best predicts the values of the dependent variable based on the values of the independent variable.

For simple linear regression to work effectively, certain assumptions and requirements need to be met. These assumptions help ensure the reliability and validity of the regression analysis. Here are some key requirements for simple linear regression: Linearity, Independence, Homoscedasticity (Constant Variance), Normality of Residuals, Random Sampling & No Outliers. While all the assumptions and requirements of simple linear regression are important, the assumption of linearity is often considered the most crucial.

consider a dataset related to business that involves predicting the monthly revenue of a store based on the advertising budget spent.

|  |  |
| --- | --- |
| Advertising Budget (X) | Monthly Revenue (Y) |
| 9500 | 44986 |
| 6500 | 35353 |
| 3000 | 18436 |
| 4000 | 26017 |
| 7500 | 36779 |
| 8000 | 40885 |
| 10500 | 50692 |
| 5000 | 30040 |
| 5500 | 33171 |
| 2000 | 14834 |
| 7000 | 35229 |
| 1000 | 12211 |
| 8500 | 41102 |
| 6000 | 32085 |
| 4500 | 25362 |
| 9000 | 43156 |
| 1500 | 12376 |
| 2500 | 17188 |
| 7000 | 20487 |
| 9000 | 46421 |

In this project, you are asked to apply Simple linear regression method using the given data set. Your goal is the is to create a model for the relationship between a dependent variable (Y) and an independent variable (X) to produce a linear equation. The general form is Y = mx + b, where m is the slope and b is the y-intercept.

**Step 1:** Create a scatter plot to visualize the relationship between X and Y. This can help you identify if a linear regression model is appropriate. Use excel to complete the this step and post the graph in the box below:

|  |
| --- |
| **Post the Graph here:** |

**Question:** Do you believe the data follows a linear distribution? Please substantiate your response:

**The distribution is very likely to be linear, analyzing the graph we can see that as the advertising budge gets bigger, the monthly revenue gets bigger too, and in what indicates in a linear manner.**

**Step 2:**

Using the following 2 java statements:

// Advertising Budget (X) array

int[] advertisingBudget = {9500, 6500, 3000, 4000, 7500, 8000, 10500, 5000, 5500, 2000, 7000, 1000, 8500, 6000, 4500, 9000, 1500, 2500, 7000, 9000};

// Modified Monthly Revenue (Y) array

int[] monthlyRevenue = {44986, 35353, 18436, 26017, 36779, 40885, 50692, 30040, 33171, 14834, 35229, 12211, 41102, 32085, 25362, 43156, 12376, 17188, 20487, 46421};

those 2 arrays are considered the training data set.

Write a java program to find the values of m and b. for that to happen, you should complete the following:

1. **Calculate the Mean:** Calculate the mean of X (mean\_x) and the mean of Y (mean\_y).
2. **Calculate the Slope (m):** Use the formula for the slope:

m=

mean\_x and mean\_y are the means of X and Y, respectively. xi & yi are the actual values of the training data.

You will need to use loop to calculate the numerator & denominator. If help is needed, contact the instructor.

1. **Calculate the Intercept (b):**

Use the formula for the y-intercept: b=mean\_y – m \* mean\_x

1. **Write the Regression Equation:**

Once you have the values for m and b, write (print on the screen) the equation:

Y=mx + b.

1. **Evaluate the Model:**

To check the model accuracy. We can use a statistical measure called R-Squared. R-squared is like a report card for a math model. It tells you how well the model predicts something. The score is between 0 (not explaining anything) and 1 (explaining everything). So, the higher the R-squared, the better the model at predicting. You will need to use loop to calculate the numerator & denominator. If help is needed, contact the instructor. the following formula shows how R-squared is calculated.

y\_predicted presents the results of applying the module using all the values of the advertisingBudget array.

**Question:** Print the value of R2, based on the produced value, can we conclude that the model indicating a strong fit?

1. **Make Predictions:**

Use the regression equation to make predictions for Y based on new values of X. print the value of Y (monthly revenue) for each of the following values of X (Advertising Budget):

X=13000 Y=?

X=18000 Y=?

X=2500 Y=?

X=6000 Y=?

X=10000 Y=?

**Post your project code below:**

|  |
| --- |
| /\*   \* Name: Guilherme dos Santos   \* 12/08/2024   \* Simple Linear Regression   \*   \* With this model we have two sets of values and we use them to   \* make a linear regression and predict future outcomes based on the previous ones.  \*/  //Main class for the project Simple Linear Regression  public class Project {      /\*\*       \* Method to calculate the mean of x and y.       \*       \* @param valueArray The array of integers which we want to calculate the mean.       \* @return The double mean.       \*/      public static double mean(int[] valueArray) {          // Initialize variables          double mean = 0; // Mean value          double sum = 0; // Sum of all values          // Loop through the array to calculate the sum of all values          for (int i = 0; i < valueArray.length; i++) {              sum += valueArray[i];          }          // Calculate the mean          mean = sum / valueArray.length;          // Return the mean so I can store it in a variable          return mean;      }      /\*\*       \* Method to calculate the numerator of the slope formula.       \*       \* @param budget       array of int representing the x values.       \* @param revenue      array of int representing the y values.       \* @param mean\_budget  double representing the mean of the x values.       \* @param mean\_revenue double representing the mean of the y values.       \* @return The double value of the numerator.       \*/      public static double numerator(int[] budget, int[] revenue, double mean\_budget, double mean\_revenue) {          double numerator = 0; // Initialize the numerator          // Loop through the arrays to calculate the numerator          for (int i = 0; i < budget.length; i++) {              numerator += (budget[i] - mean\_budget) \* (revenue[i] - mean\_revenue);          }          return numerator;      }      /\*\*       \* Method to calculate the denominator of the slope formula.       \*       \* @param x      array of int representing the x values.       \* @param mean\_x double representing the mean of the x values.       \* @return The double value of the denominator.       \*/      public static double denominator(int[] x, double mean\_x) {          double denominator = 0; // Initialize the denominator          // Loop through the array to calculate the denominator          for (int i = 0; i < x.length; i++) {              denominator += Math.pow(x[i] - mean\_x, 2);          }          return denominator;      }      /\*\*       \* Method to calculate the R^2 (R squared) value.       \*       \* @param budget  array of int representing the x values.       \* @param revenue array of int representing the y values.       \* @param m       double representing the slope.       \* @param b       double representing the intercept.       \* @param mean\_y  double representing the mean of the y values.       \* @return The double value of the denominator.       \*/      public static double r\_squared\_method(int[] budget, int[] revenue, double m, double b, double mean\_y) {          double r\_squared = 0;          double y\_predicted = 0;          double sse = 0; // Sum of Squared Errors          double sst = 0; // Total Sum of Squares          // Loop through the arrays to calculate the R^2 value          for (int i = 0; i < budget.length; i++) {              y\_predicted = (m \* budget[i]) + b;              sse += Math.pow(revenue[i] - y\_predicted, 2);              sst += Math.pow(revenue[i] - mean\_y, 2);          }          r\_squared = 1 - (sse / sst); // Calculate the R^2 value          return r\_squared;      }      public static void main(String[] args) {          // ------------------- Variables to use -------------------          int[] advertisingBudget = { 9500, 6500, 3000, 4000, 7500, 8000, 10500, 5000, 5500, 2000, 7000, 1000, 8500, 6000,                  4500, 9000, 1500, 2500, 7000, 9000 };          int[] monthlyRevenue = { 44986, 35353, 18436, 26017, 36779, 40885, 50692, 30040, 33171, 14834, 35229, 12211,                  41102, 32085, 25362, 43156, 12376, 17188, 20487, 46421 };          // Initialize the slope          double m = 0;          // New values to predict          int[] newValues = { 13000, 18000, 2500, 6000, 10000 };          // Initialize the intercept          double b = 0;          double r\_squared = 0;          // ------------------- Simple Linear Regression -------------------          // Calculate the mean of x and y          double mean\_x = mean(advertisingBudget);          double mean\_y = mean(monthlyRevenue);          // Calculate the numerator and denominator of the slope formula          double numerator = numerator(advertisingBudget, monthlyRevenue, mean\_x, mean\_y);          double denominator = denominator(advertisingBudget, mean\_x);          // Calculate the slope and intercept          m = numerator / denominator;          b = mean\_y - (m \* mean\_x);          // Print the slope and intercept          System.out.println("Slope (m): " + m);          System.out.println("Intercept (b): " + b);          // Print the regression equation          System.out.println("Regression Equation: Y = " + m + " \* x + " + b);          // Calculate the R^2 (R squared) value          r\_squared = r\_squared\_method(advertisingBudget, monthlyRevenue, m, b, mean\_y);          // Print the R^2 value          System.out.println("R squared");          System.out.println("R^2: " + r\_squared);          // ------------------- Predictions -------------------          System.out.println("------------------------------------------");          System.err.println("\nPredictions:\n");          System.out.println("------------------------------------------");          for (int i = 0; i < newValues.length; i++) {              double newY = m \* newValues[i] + b;              System.out.printf("X = %d\t\tY = %.2f\n", newValues[i], newY);              System.out.println("------------------------------------------");          }      }  } |

**Post the results of running your project blow:**

|  |
| --- |
| Slope (m): 3.97729245663574  Intercept (b): 7473.906817265026  Regression Equation: Y = 3.97729245663574 \* x + 7473.906817265026  R squared  R^2: 0.9003369132921898  ------------------------------------------  Predictions:  ------------------------------------------  X = 13000 Y = 59178.71  ------------------------------------------  X = 18000 Y = 79065.17  ------------------------------------------  X = 2500 Y = 17417.14  ------------------------------------------  X = 6000 Y = 31337.66  ------------------------------------------  X = 10000 Y = 47246.83  ------------------------------------------ |