**Exercise 7: Financial Forecasting**

using System;

using System.Linq;

namespace FinancialForecasting

{

    class Program

    {

        static void Main(string[] args)

        {

            Console.WriteLine("📊 Financial Forecasting Tool 📊");

            Console.WriteLine("--------------------------------");

            // Sample historical sales data (months vs revenue in $1000s)

            double[] months = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }; // 10 months

            double[] revenue = { 120, 150, 160, 180, 200, 210, 230, 250, 270, 300 }; // Revenue

            // Perform linear regression to get slope and intercept (y = mx + b)

            var (slope, intercept) = CalculateLinearRegression(months, revenue);

            Console.WriteLine($"\n📈 Trend Calculation:");

            Console.WriteLine($"Equation: Revenue = {slope:F2}x + {intercept:F2}\n");

            // Predict for next 3 months (month 11, 12, 13)

            Console.WriteLine("🔮 Forecast for Next 3 Months:");

            for (int i = 11; i <= 13; i++)

            {

                double predictedRevenue = slope \* i + intercept;

                Console.WriteLine($"Month {i,2}: ${predictedRevenue:F2}k (predicted)");

            }

            // Optionally: Show past vs predicted trend

            Console.WriteLine("\n📅 Past vs Predicted Trend:");

            for (int i = 0; i < months.Length; i++)

            {

                double predicted = slope \* months[i] + intercept;

                Console.WriteLine($"Month {months[i],2}: Actual = ${revenue[i]}k | Predicted = ${predicted:F2}k");

            }

        }

        static (double slope, double intercept) CalculateLinearRegression(double[] x, double[] y)

        {

            if (x.Length != y.Length)

                throw new ArgumentException("Data points mismatch!");

            double xSum = x.Sum();

            double ySum = y.Sum();

            double xySum = x.Zip(y, (a, b) => a \* b).Sum();

            double xSquaredSum = x.Select(a => a \* a).Sum();

            int n = x.Length;

            double slope = (n \* xySum - xSum \* ySum) / (n \* xSquaredSum - xSum \* xSum);

            double intercept = (ySum - slope \* xSum) / n;

            return (slope, intercept);

        }

    }

}

