PROBLEM-SOLVING EXERCISE #4
SIMULATION

As we've previously seen, equations describing situations often contain uncertain parameters, that is, parameters that aren't necessarily a single value but instead are associated with a probability distribution function. When more than one of the variables is unknown, the outcome is difficult to visualize. A common way to overcome this difficulty is to simulate the scenario many times and count the number of times different ranges of outcomes occur. One such popular simulation is called a Monte Carlo Simulation. In this problem-solving exercise you will develop a program that will perform a Monte Carlo simulation on a simple profit function.

Consider the following total profit function:

$$P_T = nP_v$$

Where P_T is the total profit, n is the number of vehicles sold and P_v is the profit per vehicle.

PART A

Compute 5 iterations of a Monte Carlo simulation given the following information:

> n follows a uniform distribution with minimum of 1 and maximum 10
> P_v follows a normal distribution with a mean of $4200 and a standard deviation of $800
> Number of bins: 10

Recall that for all practical purposes we will use 3 std. deviations from the mean as the maximum value for parameters following a normal distribution. Obviously, 5 iterations are not very many. In fact, typically you would simulate 10,000 iterations or so to view meaningful results but we figured that we'd give you a break ☺.

i.) What are the ranges for the 10 bins?
ii.) Fill in the table below:

Parameter	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
n	7	5	2	8	4
P_v	$3450	$3725	$4500	$7700	$2700
P_T					
Bin #					
$ Range					

iii.) Fill in the frequency of occurrences of each bin:

1:	2:	3:	4:	5:
6:	7:	8:	9:	10:

PART B

Write the following three methods and include in the Part D code listing**:**

public int GetRandomUniform(int min, int max)
This method returns a random number from a uniform distribution between *min* and *max*.

public double GetRandomNormal(double mean, double stddev)
This method returns a random number from a normal distribution with a mean of *mean* and standard deviation of *stddev*

public int GetBinIndex(double mini, double maxi, int numbins, double valuetobin)
This method returns the Bin Index given an input minimum of *mini*, input maximum of *maxi*, *numbins* number of bins, and a value to bin of *valuetobin*

PART C

Include the methods created in part B to develop a Visual C# .NET program that will simulate the basic profit calculation, $P_T = nP_v$, where *n* follows a uniform distribution, P_v follows a normal distribution, and the user can input the number of bins and number of iterations. The user must also input the *min* and *max* for *n* and the *mean* and *standard deviation* for P_v. Finally, the user can click a button and the results will be graphed on a bar chart using the Microsoft Chart Control and the average total profit (P_T) will be displayed in a textbox. Turn in a screen shot of the resulting chart using:

1. Iterations: 10000 Bins: 5 *n-min*: 1 *n-max*: 10 P_v*-mean*: 6700 P_v*-stddev*: 875
2. Iterations: 10000 Bins: 10 *n-min*: 1 *n-max*: 10 P_v*-mean*: 6700 P_v*-stddev*: 875
3. Iterations: 10000 Bins: 10 *n-min*: 1 *n-max*: 10 P_v*-mean*: 8300 P_v*-stddev*: 1200

That's three screen shots.

PART D

Extend the Visual C# .NET program developed in part C to simulate the basic profit calculation, $P_T = nP_v$, where the user can select either a uniform or normal distribution for *n* using radio buttons and then must input the appropriate parameters (*min* and *max* if they select uniform, *mean* and *standard deviation* if they select normal) and they can similarly select either a uniform or normal distribution for P_v with appropriate parameters depending on the selection. Of course, the user will input the number of bins and number of iterations. Finally, the user can click a button and the results will be graphed on a bar chart using the Microsoft Chart Control and the average total profit (P_T) will be displayed in a textbox. Also include in the program any necessary input validation for all input values. Turn in a listing of the code and a screen shot of the resulting chart using:

1. Iterations: 10000 Bins: 5 *n-min*: 5 *n-max*: 11 P_v*-mean*: 7000 P_v*-stddev*: 2125
2. Iterations: 10000 Bins: 10 *n-mean*: 8 *n-stddev*: 2 P_v*-min*: 1550 P_v*-max*: 7000
3. Iterations: 10000 Bins: 10 *n- mean*: 10 *n- stddev*: 2 P_v*-min*: 1000 P_v*-max*: 5775

That's three screen shots and a listing of the code for Parts B and D.

PART E

You're going to go to a job interview for OU Car Co. Knowing that the field is highly competitive, you have run sales scenarios ahead of time experimenting with different numbers of customers and vehicle profits given that in one month OU Car Co. sells between 2 and 9 cars uniformly distributed with profits of $5,250 on the average with standard deviation of $650. Which has a higher payoff, focusing on selling to a couple more customers or by increasing the average sale (with the same std. dev. of $650) by retraining your sales force or do they have basically the same effect on total sales? Support your answer.

PROBLEM SOLVING DELIVERABLES

You should work in teams of either two or three. You may not work alone. Turn in your results for Parts A, B, C, D, and E and your team's signed cover page at the beginning of your team's lecture on Tuesday, November 15, 2022 for Professor Hanna's lecture or Wednesday, November 16, 2022 for Professor Siadat's lecture. One set of results should be submitted per team.