Computer Assignment 1

Deliverable: A Python computer program that uses the simulation approach outlined below as the solution to the given problem. To earn credit for the assignment the program must satisfy the rubric below.

Topic

Using simulation to solve a probability problem.

Introduction

There may be multiple approaches to solving a math problem. Historically abstract symbolic arguments were the accepted way of addressing math problems. As computational aids have grown numerical approaches have become more common; though, purist may view them unfavorably. With the modern computers numerical approaches have become ubiquitous. When we focus our interest on that branch of mathematics called probability here too computers can be used to solve problems. The application of computers to solving a probability problem is sometimes referred to as *simulation*. The idea is to code the probability experiment intrinsic to the problem. If this code satisfactorily models the probability experiment, then it may be repeated for as many trials of the experiment as desired. This provides a way of simulating the probability experiment for the problem being addressed. Thus if feasible, a computational solution yielding an answer that approximates the true answer may be obtained.

Example 'Balls in Cans' Problem

> Three balls are tossed randomly into 5 cans. What is the probability that they all land in different cans? (Note each of the cans are able to hold from 0 to 3 balls.) Probability and Stochastic Processes by F. Solomon (1987), page 190.

The problem can be solved using different approaches. The problem is first solved in the traditional abstract way. In this context there are at least two ways to solve this problem.

First Solution

One way to respond to the problem is with classical probability: the ratio of the number of outcomes in the event to the number of outcomes in the sample space. We ask ourselves, what are the number of outcomes in the sample space? Because each ball has five choices and these choices are independent (lexicographical definition of independent not the probabilistic definition of independent) the answer is $5 \cdot 5 \cdot 5 = 125$. Similarly, we ask ourselves, what are the number of outcomes in the event? For the event we want them all in separate cans; so, while one ball has five choices, the other two will have only four or three choices respectively and the answer is $5 \cdot 4 \cdot 3 = 60$. The ratio is the answer; thus, $60/125 = 0.48$.

Second Solution

Another way to respond to the problems is with conditional probability: The product of the three conditional probabilities. Then we need to both ask ourselves three questions and answer those questions.

First question, what is the conditional probability of the first ball? Answer, choose any of the five available cans from the five cans or $5/5$.

Second question, what is the conditional probability of the second ball? Answer, choose any of the remaining four available cans from the five cans or $4/5$.

Third question, what is the conditional probability of the third ball? Answer, choose any of the remaining three available cans from the five cans or $3/5$.

The product is the answer; thus, $1 \cdot \frac{4}{5} \cdot \frac{3}{5} = \frac{12}{25} = 0.48$.

A computational solution of the problem.

Consider that each ball has a choice of five cans and that the cans are chosen at random. We can label the three balls (starting at zero for reasons that will become apparent) as, $\{0, 1, 2\}$. We can label the five cans with the numbers one through five, $\{1, 2, 3, 4, 5\}$. In the context of computer programming what is a possible way of doing the assignment of balls into cans?

As the reader is no doubt familiar computer languages typically have arrays (Python calls these lists). The storage locations in an array (list) are known as elements and each element in an array is assigned a unique number known as a subscript. So, we can have three elements in the array with indexes 0, 1, and 2. We can have a random number generator provide randomly the numbers 1, 2, 3, 4, and 5. Any one of these five numbers can be assigned to each of the three elements. What are some possible instances of these arrays?

[3, 3, 3] three the same

[2, 3, 2] two the same

[4, 1, 1] two the same

[2, 1, 5] all different

[5, 2, 4] all different

When we have written the code to produce all the possibilities then we have modeled the probability experiment. The goal is to solve the probability problem by using both a computer and a computer language to create a *simulation* of the probability problem. The model of the probability experiment above is part of this simulation. Additionally, we need to separate from all the results those that satisfy the requirements of the event: all elements (cans) are different. The experiment would be repeated many times (as big as needed) and the ratio of the successful results to all the results will approach the correct answer with increasing accuracy (with more repetitions). It is not reasonable for the answer to be exactly the true answer. The Python program of this solution is provided with this assignment.

Conclusion

Arguably a way of understanding probability theory is to explore both a traditional 'pencil and paper' solution and a modern 'computational' solution using a fundamentally different approach from the 'pencil and paper' solution. In this way the reader should be able to look back over what has been done and see how and why the different approaches successfully address the same problem. In closing to evaluate whether we have been successful the reader is now invited to solve the problem below. A rubric is provided as a guide to efficiently prepare and submit the necessary work.

Problem 'Auditor'

Three firms using the same auditor independently and randomly select a month in which to conduct their annual audits. What is the probability that all 3 months are different? Elementary Statistics by M. Triola (1997), Page 152.

Rubric

Name and I.D. # Name of Assignment Due Date Submission Date	Not Satisfactory: Data requested absent.	Satisfactory: Has all data requested.
A block comment at beginning of program summarizing it. Line comments throughout the program explaining its operation.	Not Satisfactory: Conveying incomplete thoughts or unnecessarily excessive.	Satisfactory: A brief two or three sentence explanation.
Identify lines in program that model the experiment intrinsic to the probability problem.	Not Satisfactory: Experiment not identified or blocked off.	Satisfactory: Experiment clearly identified.
List references.	Not Satisfactory: No references listed.	Satisfactory: References – assignment handout, internet, students, and etc. listed
Run the program multiple times with magnitude shifts in the number of overall repetitions of the experiment.	Not Satisfactory: Not run for a sufficient number of values to show change in output.	Satisfactory: Output shows increasing accuracy with increasing repetitions of experiment.
Upload the .py file of the program to and a PDF of the program with the output.	Not Satisfactory: Absent files. Absent output.	Satisfactory: The two files and the output are submitted.