Monte-Carlo Simulation for a Call Center

By Sidhardh Burre

APMA 3100: Probability

October 22, 2021

# 1 Problem Statement

This project will introduce you to the fundamentals of *Monte-Carlo simulation* – a technique of artificially creating random variables which can be large, and which can be used to simulate the behavior of random processes, as well as to solve problems intractable analytically.

In a nutshell, it is a technique for numerical experimentation on a computer. While many software packages offer various simulation programs, we shall not use them here. For our objective is to train not as users, but as engineers—the designers and developers of new specialized systems. And to succeed in such creative tasks, we must understand the fundamentals of system simulation.

A representative of a high-speed Internet provider calls customers to assess their satisfaction with their service. It takes her 6 seconds to turn on a phone and dial a number; then 3 additional seconds to detect a busy signal, or 25 additional seconds to wait for 5 rings and conclude that no one will answer; and 1 second to end a call. After an unsuccessful call, she re-dials (over several days) until the customer answers or she has dialed four times. The *outcome* of each dialing is determined in an identical way: the customer being called is using the line with probability 0.2; or is unavailable to answer the call with probability 0.3; or is available and can answer the call within X seconds, which is a continuous random variable with mean of 12 seconds and the exponential distribution. The *calling process* ends when the customer answers the call, or when four unsuccessful calls have been completed.

Let W denote the time spend by the representative on calling one customer. Your objective is to estimate several statistics of W.

## 1.1 Defining the elements of the model

W[C] = The total time spent calling a single customer, a function of the number of times a customer is called and the results of each call.

t1 = The time to turn on the phone and dial a number (6 seconds in this case).

t2.1 = The time it takes for the call to be determined as no-answer, also the cutoff time

for the distribution (25 seconds).

t2.2 = The time it takes to identify that the call is busy (3 seconds).

t2.3 = The time taken for the customer to pick up, an exponential distribution with mean

μ. 0 ≤ t2.3 ≤ t2.1

μ = The mean time of an answered call (12 seconds).

t3 = The time taken to end a call (1 second).

C = the number of times a customer is called with data about how the call went (busy, no one will answer, answered)

p2.1 = The base probability of taking branch t2.1 (0.3)

P2.1 = The corrected probability of taking branch t2.1 (p2.1 + (1-p2.1-P2.2)e(-1/ μ)\*t2.1) = 0.362

P2.2 = The probability of taking branch t2.2 (0.2)

P2.3 = The probability of taking branch t2.3 (1-p2.1-P2.2)(1-e(-1/ μ)\*t2.1) = 0.438

The random variable X or t2.3 is an exponential random variable defining how long it takes for the customer to pick up. As we know that if the customer takes longer than 25 seconds to pick up, the call is ended and the callee must call back, the domain of X is only [0, 25). The probability of X>25 was simply added to another branch (essentially). As we have established that X is an exponential random variable, we know that the probability distribution function (PDF) and cumulative distribution function (CDF) will be of the following forms:

Text

Description automatically generatedBecause we were given the mean (or expected value) to be 12, we can solve for λ:

We find that λ is equal to 12. We can then plug this value in to get the PDF and CDF:

For λ > 0, x > 0:

## 1.2 Write the expression for the CDF of X and its inverse

We know the function for the CDF of an exponential variable:

Bounded over the range x є [0, INF). To invert the expression, we swap x and y:

And solve for y:

For our specific model where λ = 1/12, we gain Equation 1. The equation would be as such:

Assuming we maintain the bounds of the original equation, x є [0, 1].

## 1.3 Draw the tree diagram for the calling process

Diagram

Description automatically generated

Where black indicates non-decision-making guaranteed nodes, gray indicates decision making nodes, blue indicates probability-attached nodes, and green indicates terminating nodes. Note that only some time-consuming nodes have a probability associated with them indicating that some time-consuming nodes must be gone through every time.

# 2 Collect data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table 1: Real World Values for Times of Actions** | | | | | |
|  | **Time (sec) for** | | | | |
| **Trial** | **Dial Time** | **Busy Signal** | **No Answer** | **Phone Down** | **Callee pick up time** |
| 1 | 3 | 4 | 34 | 1 | 3 |
| 2 | 4 | 5 | 35 | 1 | 4 |
| 3 | 3 | 3 | 35 | 1 | 5 |
| 4 | 3 | 4 | 35 | 1 | 2 |
| 5 | 4 | 4 | 34 | 1 | 3 |
| **Avg** | 3.4 | 4 | 34.6 | 1 | 3.4 |
| **Problem** | 6 | 3 | 25 | 1 | 12 |

To generate values for the average time taken of each action, a set of 3 iPhones with carriers AT&T, T-Mobile, and Verizon were used. Of the values found the greatest differences were found in picking the phone up and prepping for the call, the time for no answer, and the time for picking up.

The time for picking the phone up and prepping the call may have been different due to how quickly the subjects in the conducted experiment could dial in phone numbers. Whereas the values provided may have been due to the subject being unable to type in phone numbers so quickly.

The time for no answer may also have been longer due to differences in phone type, the nearly ten second gap suggests that the experimental phones rang for longer across the board.

The time for picking up was determined by calling random users from the subject’s call log. This value may have been deflated due to the familiarity between the subjects and the callees. Unlike the unfamiliarity between the caller and callee in the experiment.

# 3 Design a Monte-Carlo simulation algorithm

The Monte-Carlo simulation has two primary components, the random number generator and the function that determines the output of each trial, or the total time taken to call a customer. In this particular simulation, a linear congruential random number generator was used. This function would output values between 0 and 1. Because of the specific implementation of the time-taken function, four times the number of experiments number of random numbers were needed.

The function that determined the time taken on each individual customer took in a random number and using that random number determined the added time value for that specific call. But irrespective of the random value, each call would automatically be incremented by 7 seconds to account for the set up/set down time. Where each call is only a component of the experiment. If the random number was such that it fell into the bucket that assumed the customer picked up (a 50% chance), the random variable was manipulated and scaled to encompass the complete [0, 1] range. Using this new scaled value, the time taken for the customer to pick up the phone was determined by inputting the scaled value into Equation 1 to get an output of time. If the customer was to take more than 25 seconds to pick up the phone, the time value for that trial would simply be incremented by 25 and the trial would continue. But if the customer were to pick up, the inverse X function detailed in section 1.2 would be applied to the scaled random variable to get the time taken to pick up. This value would subsequently be added to the total time taken for the trial and the trial would be ended.

This implementation was a Python-based function with many built-in Python math and statistics functions used. These functions came from the math, statistics, matplotlib, and numpy libraries. The values of random numbers 51, 52, and 53 are 0.062835693359375, 0.7091064453125, and 0.072784423828125 respectively.

To compute the values of P[W < X] where X is an arbitrary value, a sorted list of the times for each experiment was iterated through until the first value where the list value exceeded the value of X was found. The index of the preceding value divided by the total length of the list was returned to provide the probability of P[W < X].

# 5 Estimate and Analyze

For this experiment, 1000 realizations were generated. In the context of the problem, this was the equivalent of calling 1000 customers. In the box-and-whisker graph (Fig. 1), sets of times that consist of a lower number of realizations are displayed for comparison reasons. But all other Figures and statistics belong to the set of 1000 realizations.

|  |  |  |
| --- | --- | --- |
| **Table 2: Statistical Values** | | |
| **Property** | | **Value** |
| **mean** | | 41.712 |
| **Quartiles** | **1st** | 15.052 |
| **2nd** | 28.918 |
| **3rd** | 62.000 |
| **Probabilities** | **W < 15** | 0.247 |
| **W < 20** | 0.356 |
| **W < 30** | 0.516 |
| **W > 40** | 0.438 |
| **W > 70** | 0.212 |
| **W > 100** | 0.068 |
| **W > 120** | 0.015 |

As one can see, the distribution echoes that of a Geometric distribution due to an innate property of how the trials are conducted, the calls are repeated until success. Because we have a maximum of four calls, this induces pockets of distribution where the results clump into four pockets. This can been seen in the values above, namely by comparing the value of P[W > 40] and the value of P[W < 15]. By comparing those two values, we find that approximately 40% of all values land between 15 and 40 minutes. This statement is echoed in Figure 2, where we display a histogram of the frequencies of call times. The median is significantly lower than the mean implying that the data has a positive skew with a peak to the left of the mean. This suggests that there are high-value low-frequency values within the data-set. This is indicative of an exponential distribution for the PDF.

Below is the box-and-whisker plot for 10, 100, and 1000 realizations.

Chart, box and whisker chart

Description automatically generated

As one can see, as the median trends lower while the 1st and 3rd quartiles as well as the upper and lower maximum spread and tread away from one another. This suggests that for this experiment, the probability of low call times for a customer is extremely high, a reality that is exposed as larger numbers of realizations are generated for the model.

Chart, histogram

Description automatically generatedFurther, we find that the range of W, the call time per customer ranges from 7 to 128. This is in accordance with the project design. The first case, 7 cannot be achieved but a close enough value can be achieved if the customer picks up on the first trial and nearly immediately. The upper bound, 128, can be realized by the 25 second wait time to ensure that the call has failed 4 times combined with the 7 second prep time for each call (6 seconds to dial and 1 second to end the call).

As suggested originally, the frequencies do indeed clump into approximately four groups with the following approximate boundaries: [0, 32], [32, 64], [64, 96], [96, 128]. Analyzing each group, we find that the boundaries of the groups are approximately multiples of 32. This is in line with our experimental setup. The first most likely branch in our experiment is that the caller picks up. But this branch produces highly variable values. The second most likely branch, the branch where the caller waits 25 seconds, is still highly likely but produces very consistent values. This produces barriers at multiples of 32 due to the 7 second prep time required for each call.

Each individual group has what appears to be a distribution that is high in the beginning and tapers off towards the end, echoing an exponential distribution, coincidentally the same type of distribution as the call time. This is because the probability of proceeding along any number of 25 second branches prior to the customer picking up the phone (with a time defined by an exponential distribution), is very high. Although it may be simple to assume that W overall (the time spent to call each customer) is an exponential variable, this is clearly untrue. Although each group may express an exponential distribution, the overall distribution still does not echo this shape due to the separation of each exponential distribution-expressing group. This can be more clearly seen in a histogram of 100,000 realizations:

Chart, histogram

Description automatically generated

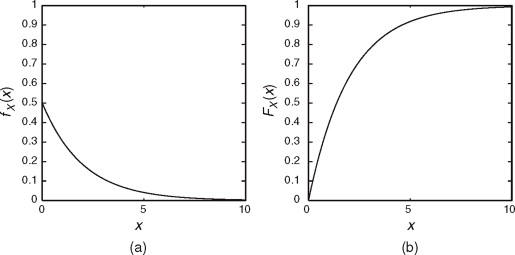
One can see a more definite exponential distribution as well as peaks at multiples of 32 + 12, echoing the mean of the call time. These peaks are most notably displayed at t = 12, 44, and 108. This is within expectations due to the original parameters of the experiment.

Chart, line chart

Description automatically generated

Looking at the graph here I would doubt that W can be an exponential random variable. This is because despite W having a shape that appears like an exponential random variable, if one analyzes the basis of the CDF, the PDF, one will find that the distribution is not that of an exponential random variable. This is glaringly obvious when viewing the lack of a smooth curve in both the CDF and PDF. Compared to Fig 5 which displays the CDF and PDF of an exponential variable, the curves generated from this experiment can be seen as non-similar.

Fig. 5: CDF and PDF of an exponential RV



# 7 Comment

The steps that were the most/least challenging where the initial modelling and report writing steps respectively. The reason the initial modelling was so difficult was due to the low specification in the original report, despite the relatively easy concept. The ambiguity in the original project specification produced much confusion. The programming and Monte-Carlo simulation were relatively simple in comparison.

The step that was the most time-consuming was coming up with the report and analyzing the data. Coming with useful and relevant analyses was a difficult, time-consuming process due to the need for perpetual reassessment.

The results of the simulation in this report should mostly be without errors. With frequent discussions with both the TA and Professor of this class, most of the conceptual errors was found and addressed.