APMA 3100 Project 2 (Final Report)

By Matthew Beck and Spencer Hernandez

Honor Pledge: On my honor as a student I have neither given nor received aid on this assignment

X: Matthew Beck & Spencer Hernandez

Introduction

A worker at an Internet Service Provider is calling customers to receive feedback on their service. She will call the customer until they answer, or until she has called 4 times, whichever is sooner. There is a 0.5 probability that the customer is available and will answer their phone in X seconds (where X is an exponential random variable), a 0.3 probability that the customer is unavailable, and a 0.2 probability that the customer is currently calling someone else. The total time spent by the worker calling a customer is denoted as W, and can be calculated by taking into account the following. It takes the worker 6 seconds to dial the customer's phone number, and 1 second for the call to end. The time in between the dialing will either take the worker 3 seconds to wait for a busy signal, or it could take 25 seconds to wait for 5 rings and then hang up the call (note that the customer can answer the call anywhere between 0 and 25 seconds).

1.) Formulate a Model

1.1

$t_i = 6$: Initial time to dial and call

$t =$ Duration of ringing {Time for busy Signal (3 s), otherwise ($0 \leq t \leq 25$)}

$t_e = 1$: Time to end call

$i = 1, 2, 3, 4$: Number of dials for customer to answer

C = State of the Customer { Customer is using the line (0.2), Customer isn't available (0.3), Customer is available (0.5)}

W: Time spent calling by representative

X: Given the customer can answer, the time it takes for the customer to answer

1.2

E[x] = 12

$\lambda = \frac{1}{E[x\}} = \frac{1}{12}$

PDF:

$$f_X(x) = \begin{cases} \frac{1}{12}e^{-x/12}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

CDF:

$$F_x(x) = \int_0^x \frac{1}{12} e^{-x/12}\, dx = 1 - e^{-x/12}$$

$$F_x(x) = \begin{cases} 1 - e^{-x/12}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Inverse CDF:

$$y = 1 - e^{-\frac{x}{12}}$$

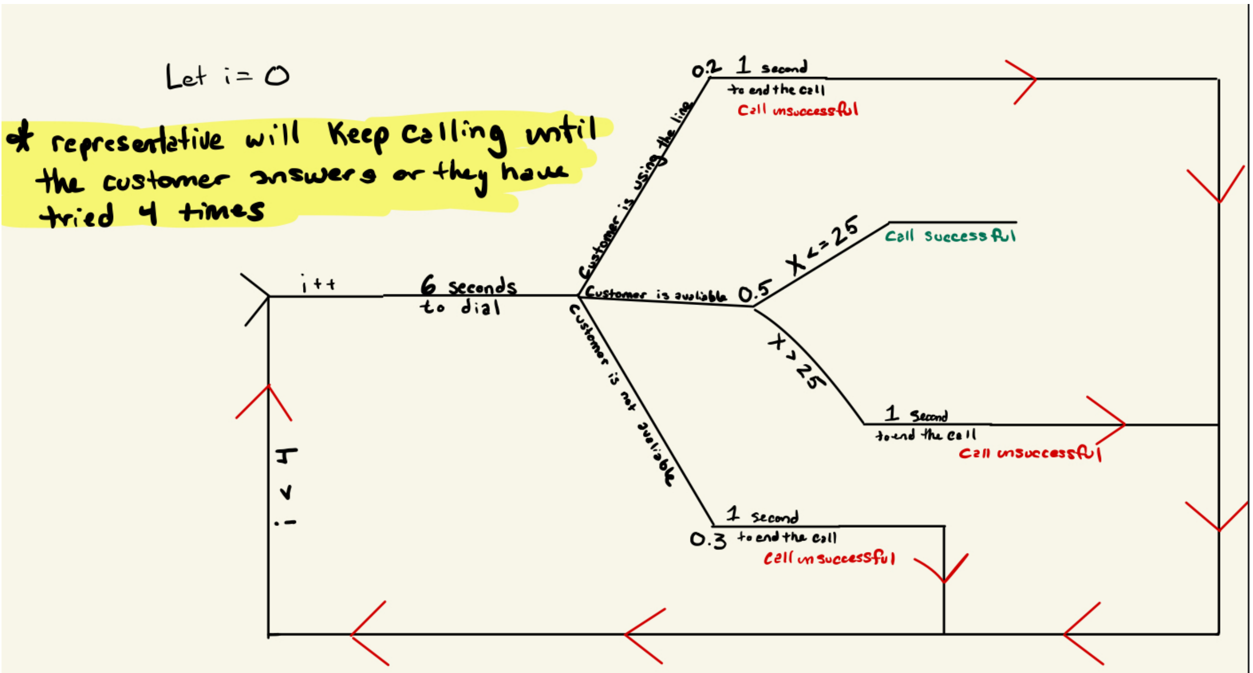
$$x = 1 - e^{-y/12}$$

$$1 - x = e^{-y/12}$$

$$-\frac{y}{12} = \ln(1-x)$$

$$y = F^{-1}(x) = -12\ln(1-x)$$

1.3



2.) Collect Data

	Time to Turn on Phone and Dial # (sec)	**Time to Detect Busy Signal (sec)**	**Time to Wait for 5 Rings (sec)**	**Time to End Call (sec)**	**Time to Pick Up (sec)**
Trial 1	7.01	2.29	28.85	1.17	24.96
Trial 2	5.51	3.68	27.94	0.85	4.53
Trial 3	6.99	4.47	28.33	1.52	7.89
Average Time	6.50	3.48	28.37	1.18	12.46

3&4.) Design Algorithm & Simulate Calling Process

1. **Description of Generation of Waiting Time (W)**
   Our algorithm defines the parameters for the Linear Congruential Generator (LCG) including x_0 (the seed), a (the multiplier), c (increment), and K (modulus). It then runs a for loop from 0 to n (the algorithm's parameter to calculate the n^{th} realization of the

random variable X), which will calculate the current iteration of x according to the $x_i = (ax_{i-1} + c) \% K$. The algorithm will then set x to the next iteration of x. Now that we have the i^{th} realization, x_i, we can divide it by K to get the i^{th} realization of u, which is nothing more than a pseudo-random probability of that x value. Given this probability, the algorithm then plugs in u_i into the inverse formula of the Exponential Random Variable ($F^{-1}(u_i) = -12ln(1 - \frac{x_i}{K})$) for some realization of how long it takes for the customer to answer the call (given that they are able to), which is our X variable.

We coded this in Python and used a Monte Carlo algorithm by running a for loop n times, where n is the number of independent realizations, which generates the realization of W in accordance with the algorithm above, and then uses a random number generator to determine whether the customer is busy or not. The random number is a number between 1 and 100 where if the number is less than or equal to 20 then it will add 10 to a variable called totalTime and increase a call counter by one. Otherwise, we check if the random number is less than or equal to 50 or the realization of W is greater than 25, if so we increase totalWait by 32 seconds and increase the call counter by one. Then, we have an else statement which represents the customer answering the phone which adds (6 + the realization of W) to the totalWait and sets a boolean variable, pickup, to true. This is all in a while loop which ensures that the customer has not picked up and that they haven't been called four times. When either of these cases are true, the totalWait is appended to a list that we calculated statistics for after all the iterations of the for loop.

2. **Generation of u_{51}, u_{52}, and u_{53}**

Based on our LCG algorithm, we generated the following results for u_{51}, u_{52}, and u_{53}:

$u_{51} = 0.4273$

$u_{52} = 0.7682$

$u_{53} = 0.0933$

```
51
 LCG:  56004
 U:  0.427276611328125
52
 LCG:  100689
 U:  0.7681961059570312
53
 LCG:  12226
 U:  0.0932769775390625
```

5.) Estimations

The estimations that were generated by our code can be seen below:

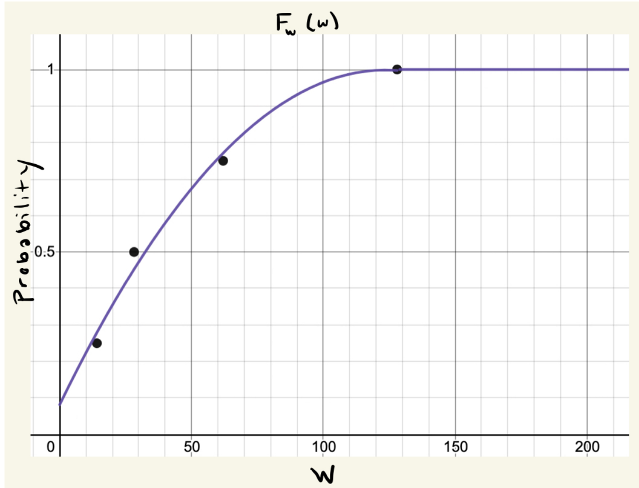
```
The mean is: 43.70671221054583
The first quartile is: 14.11373561233988
The median is: 28.22412384688014
The third quartile is: 62.0
The probability W is less than or equal to 15 is: 0.2682682682682683
The probability W is less than or equal to 20 is: 0.3813813813813814
The probability W is less than or equal to 30 is: 0.5145145145145145
The probability W is greater than 40 is: 0.42242242242242245
The probability W is greater than 65 is:  0.24124124124124124
The probability W is greater than 85 is:  0.15315315315315314
The probability W is greater than 105 is:  0.12212212212212212
```

6.) Analyze

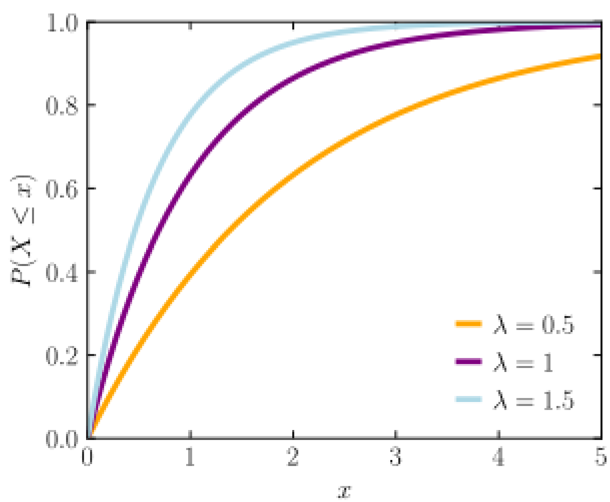
6.1) The mean is greater than the median by approximately 15 indicating that the distribution is right-skewed with a large right tail.

6.2) The sample space of W is [6,128]. This is where $F_x(x)$ is greater than or equal to 0. So in the minimum of the sample space is where it takes 6 seconds to dial and they pick up the phone instantly. The maximum of 128 comes from the client not picking up for the 4 tries to call them where each phone call takes a total of 32 seconds.

6.3) We generated the graph below to represent the CDF of W as closely as possible using interpolation. We believe that W could be represented by an exponential random variable because the CDF that we generated for W is a similar shape to the CDF of an exponential random variable. This can be seen through comparison of the graph for the CDF of random variable W and the graph of the CDFs of multiple exponential random variables as shown below.



Graph of the CDF for random variable W



Graph of multiple exponential random variable CDFs (with arbitrary λ values to demonstrate the shape is correct)

7.) Comments

i.) The most challenging step was figuring out how to generate pseudo-random numbers using the LCG because of how x has a base case and must increment on subsequent calls. The least challenging step was performing the ad-hoc experiment because all it involved was starting and stopping a timer while recording data.

ii.) The step that is most time consuming was plotting the CDF. Depending on how we set our minimum and maximum, the probability at w=0 would be incorrect or the probability would equal 1 at some small w value (around 100). The least time consuming part was also the ad-hoc experiment. Again, this is because it required little thought and was straight to the point.

iii.) We are fairly confident that our answers are accurate. The CDF we graphed is a very close regression of the data points generated from the simulation. We know that the LCG generates the correct random numbers based off of the first 3 that were provided in the project description, and there is very little room for error when generating the realizations of x, leaving the uncertainty to calculating w from our x value. After running through the code and following the logic from the tree diagram, we feel confident that we do not have any error in this area either.