Professor: *Dr. Mohammed Ayoub Alaoui Mhamdi*

**Full Name:**

**Number:**

**Signature:**

**Bishop’s University**

**CS 504 – Programming Languages for Data Analysis**

**Final exam**

**Winter 2021**

**April 26th, 2020**

*Be sure to read the whole exam before attempting any of it. This final exam is open book since it is online. The exam begins at 8:30 am and ends at 8:30 pm. Be sure to submit your answers in one pdf file. In addition, you have to submit your code in Julia, Python, R, and Octave. This final exam is individual not in group. Thus, any student who will get involved in plagiarism or relied on an online tutor will get attributed zero in total grade of the whole course. Use the provided white space to respond to each question.*

***Any late submission or update of the final exam after 8:30 pm will get penalized of 10% for each late 30 minutes.***

**Problem: Geometric Distribution**

Another distribution associated with Bernoulli trials is the *geometric distribution*. In this case, consider an infinite sequence of independent trials, each with success probability *p*, and let be the first trial that is successful. Using first principles it is easy to see that the PMF is,

An alternative version of the geometric distribution is the distribution of the random variable , counting the number of failures until success. Observe that for every sequence of trials, . From this it is easy to relate the PMFs of the random variables and see that,

In the Julia *Distributions* package, Geometric stands for the distribution of , not .

We now look at an example involving the popular casino game of roulette. Roulette is a game of chance, where a ball is spun on the inside edge of a horizontal wheel. As the ball loses momentum, it eventually falls vertically down, and lands on one of 37 spaces, numbered 0 to 36. There are 18 black spaces, 18 red, and a single space (‘zero‘) is green. Each spin of the wheel is independent, and each of the possible 37 outcomes is equally likely. Now let us assume that a gambler goes to the casino and plays a series of roulette spins. There are various ways to bet on the outcome of roulette, but in this case he always bets on black (if the ball lands on black he wins, otherwise he loses). Say that the gambler plays until his first win. In this case, the number of plays is a geometric random variable with support ….

To facilitate the task for you and to provide you an idea how you will perform the implementation in R, Python and Octave, the Julia is provided, and it is the following,

1 **using** StatsBase, Distributions, Plots; pyplot()

2

3 **function** rouletteSpins(p)

4 x = 0

5 **while true**

6 x += 1

7 **if** rand() < p

8 **return** x

9 **end**

10 **end**

11 **end**

12

13 p, xGrid, N = 18/37, 1:7, 10^6

14

15 mcEstimate = counts([rouletteSpins(p) **for** \_ **in** 1:N],xGrid)/N

16

17 gDist = Geometric(p)

18 gPmf = [pdf(gDist,x-1) **for** x **in** xGrid]

19

20 plot(xGrid, mcEstimate, line=:stem, marker=:circle,

21 c=:blue, ms=10, msw=0, lw=4, label="MC estimate")

22 plot!( xGrid, gPmf, line=:stem, marker=:xcross,

23 c=:red, ms=6, msw=0, lw=2, label="PMF",

24 ylims=(0,0.5), xlabel="x", ylabel="Probability")

The function rouletteSpins() defined in lines 3-11 is a straightforward way to generate a geometric random variable with support as above. Lines 5-10 loop until a value is returned from the function. In each iteration, we increment and check if we have a success (an event happening with probability ) via, rand()<p. Consider the second argument to pdf() in line 18. Here is used because the built-in geometric distribution is for the random variable above, which starts at 0, while we are interested in the geometric random variable starting at 1.

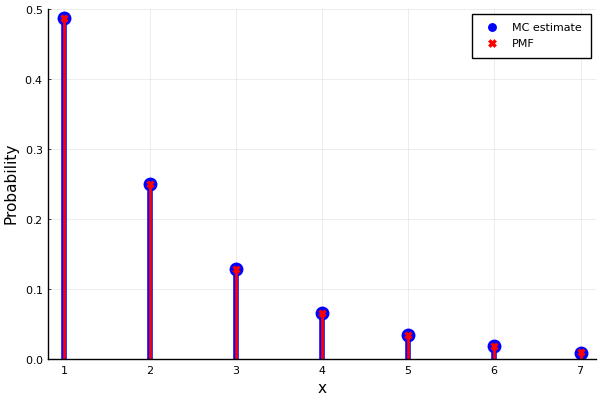


Figure 1: A geometric PMF

***Your task in this final exam is to write programs in Julia, Python, R, and Octave to plot Figure 1***. The Julia code is already provided. You are recommended to rely on the concept of this implementation on Julia to reuse the concepts in Python, Octave, and R.

Which programming language that provided relevant solution for this assignment? Justify your answer.

**Submission**

You have to submit a pdf file that contains four outputs that are related to Julia, Python, Octave, and R. In addition, you have to submit the sources files in Julia, Python, Octave, and R. ***Please, do not use .zip nor .rar file in your submission***.

Good luck :)