CSE - 411

Student ID: 1505112

September 23, 2020

1 Problem Definition

Geometric Distribution is the number failures before the first success where p(x) is the probability of x which is the number of failures before success. So, random variables of Geometric Distribution can be zero to infinity.

To simulate Geometric Distribution, we have to generate N random number. N is given as input. As Cumulative Distribution F(x) for Geometric Distribution is 1 if we take probability for all variables from zero to infinity. But Cumulative Distribution F(x) has been taken for only first 15 variables [0, 14]. Then a uniform random number [0, 1] can be generated for N times and compared each of them with cumulative distribution to find a number x for which cumulative distribution is immediate greater than the uniform random number. The number x is a random variable of Geometric distribution and frequency of x will be increased every time of that number occurrence. From this we can find a frequency distribution. Dividing frequency of each variable by N, we can get observed frequency in fraction (observed probability). Then two graphs of theoretical Geometric Distribution and observed frequency in fraction are plotted.

2 Curves of Observed probability and Theoretical probability

I have plotted two figures. In Figure-1, a curve is plotted showing Theoretical probability of Geometric Distribution. In Figure-2, a curve is plotted showing observed frequency in fraction (Observed probability).

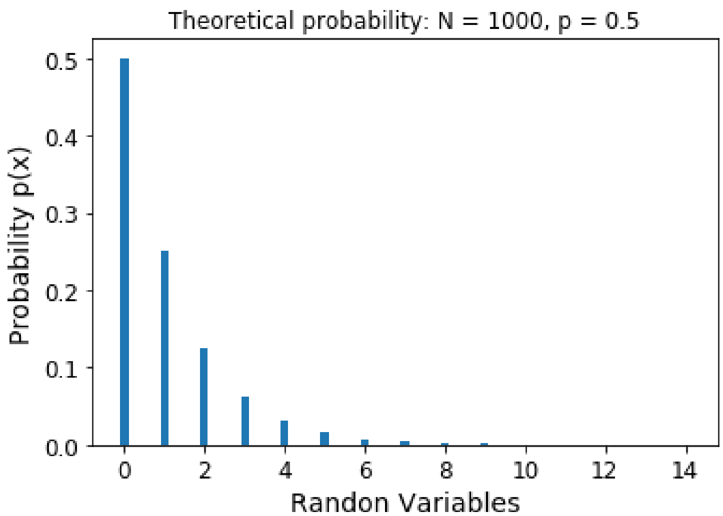

Figure 1:

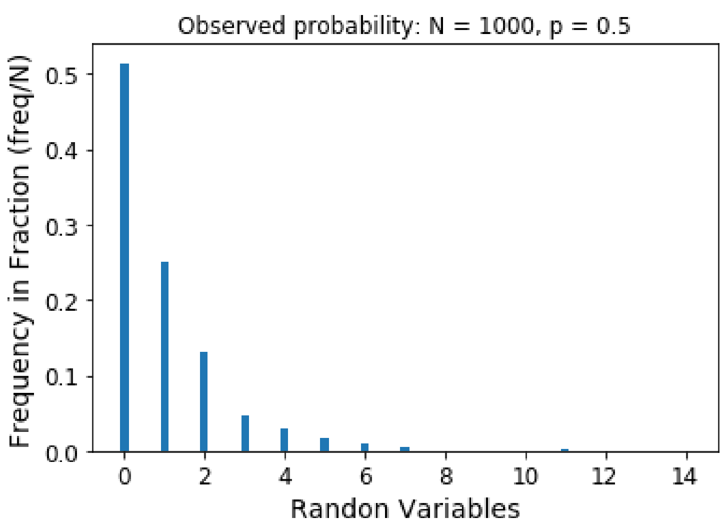

Figure 2:

3 Simulation Code

```

# imported libraries to plot graph
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.rc('axes', labelsize=14)
mpl.rc('xtick', labelsize=12)
mpl.rc('ytick', labelsize=12)

# import module for random number generation
from numpy import random
import math

# input N - #of random number
N = input("Enter number of random variables : ")
N = int(N)

cum_prob_range = 15 # cdf for [0, cum_prob_range-1]
p = 0.5 # probability of success (given)

# frequency list, each index is a random variable
freq_p = [0 for x in range(0,cum_prob_range)]
cum_prob = [0 for x in range(0,cum_prob_range)] # cdf values

# (frequency / N) for each x [0, cum_prob_range-1]
freq_frac = [0 for x in range(0,cum_prob_range)]
prob_t = [0 for x in range(0, cum_prob_range)] # theoretical
    probability

i = 0
# calculation of cumulative probability; F(x) = 1 - (1-p)^(x+1)
while i < cum_prob_range:
    cum_prob[i] = 1 - math.pow(1-p, i+1)
    i += 1

i = 0
# generation of random number and frequency distribution
while i < N:
    prob = random.uniform(0, 1) # random number between 0 to 1
    j = 0
    while j < cum_prob_range:
        # min x for which prob less than or equal cumulative
    probability of x
        if prob <= cum_prob[j]:
            freq_p[j] += 1
            break
        j += 1
    i += 1

i = 0
# calculation of observed and theoretical probability
while i < cum_prob_range:
    prob_t[i] = math.pow(1-p, i)*p # p(x) = p*(1-p)^x
    freq_frac[i] = freq_p[i] / N # p(x) = frequency(x) / N
    i += 1

```

```
# random variable from 0 to cum_prob_range-1
rand_var = [x for x in range(0, cum_prob_range)]
# plot graph
plt.bar(rand_var, prob_t, width = 0.2)
plt.xlabel('Randon Variables')
plt.ylabel('Probability p(x)')
plt.title("Theoretical probability: N = " + str(N) + ", p = " + str
    (p))
plt.show()

plt.bar(rand_var, freq_frac, width = 0.2)
plt.xlabel('Randon Variables')
plt.ylabel('Frequency in Fraction (freq/N)')
plt.title("Observed probability: N = " + str(N) + ", p = " + str(p)
    )
plt.show()
```

Listing 1: Simulation Code