<u>MATH 371 : Assignment -1</u>

-Chaitanya Garg
2021248

<u>Question 1: Modeling a Coin Toss By Bernoulli Process</u>

A coin toss is a Bernoulli(as it has only 2 possible outcomes a head/tail) trial with its outcome Z, Z~Bernoulli(p) , where p is the probability of arrival, say in this case, arrival of head.

So a coin toss can be modeled as

{ p ; arrival of head
Z = {
{ 1-p ; otherwise/arrival of tails

Now, suppose we make a procedure wherein we toss the coin with above Z~Bernoulli(p), 20 times.
Zi, denoting the outcome of each toss (i belongs to {1,2,..19,20})
Let N = sigma(i=1 to 20): Zi
Each trial Zi is independent of each other
The probability of arrival, i.e., receiving heads is same(=p) for each trial.
Hence, the assumptions lead us to conclude that Zi are IID RV's
We are studying the process for discrete states and discrete instance
Where N denote the number of heads achieved in 20 tosses so,
N~Binomial(n=20,p)

For each value of N, the corresponding value on the histogram is incremented by 1.

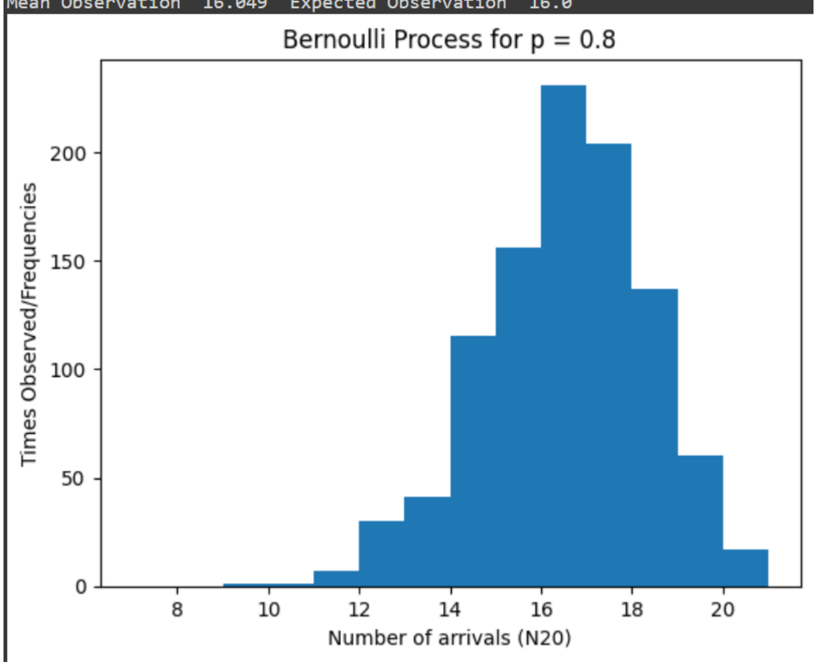
Code for simulating the procedure while keeping in mind the above idea

```python
def Bernoulli(p):
    arrivals = [] #Monitoring Arrivals
    for i in range(1000):
        n=0
        iter = 0
        while(iter < 20): #Simulating 20 iterations/trials for a coin toss
            iter+=1
            if(random.random()<=p):
                n+=1
        arrivals.append(n)
    #Plotting the histogram
    plt.hist(arrivals, bins=range(min(arrivals)-2,max(arrivals) + 2), density=False)
    plt.xlabel('Number of arrivals (N20)')
    plt.ylabel('Times Observed/Frequencies')
    plt.title('Bernoulli Process for p = {}'.format(p))
    print("Mean Observation ",(sum(arrivals)/1000), " Expected Observation ",20*p)
    plt.show()
```

Idea: For each coin toss, a random number is generated in range(0,1) using random.random() if the number generated is more than the given probability then its a no-arrival else its an arrival.

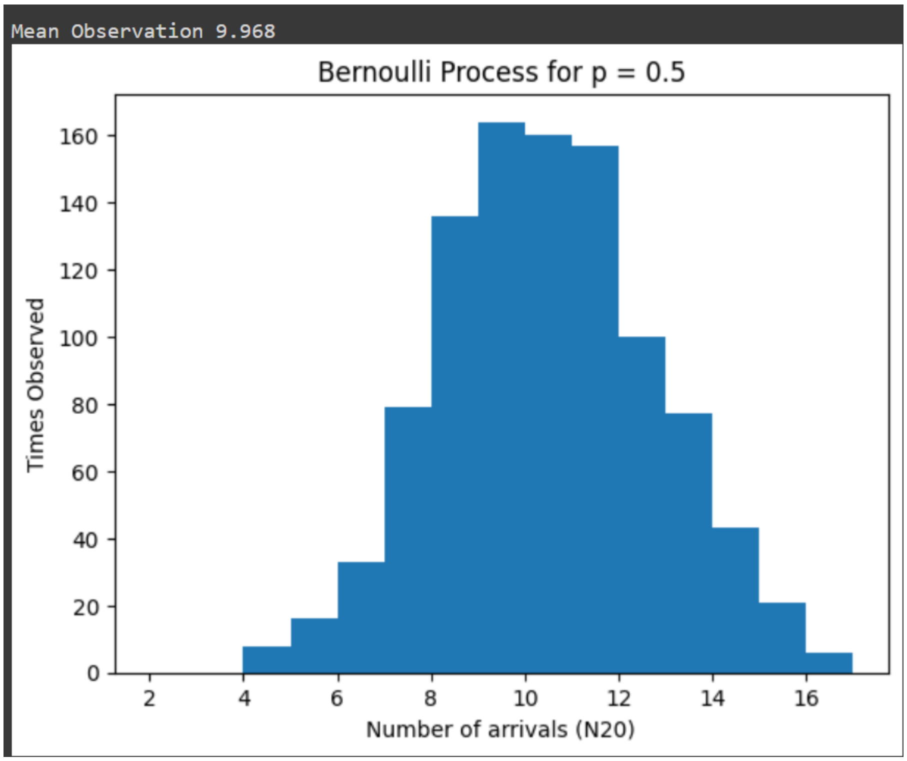
1.a)

```
Mean Observation  16.049  Expected Observation  16.0
```



Note: The corresponding bar for each number of arrivals appears after that number

1.b)



In the former(1.a), the bars seem to be concentrated around 16, which is expected since, expected number of outcomes is 16 and mean is 16.05. While in the latter(1.b), the bars seem to be concentrated around 10, which is expected since, expected number of arrivals is 10 and mean is 9.97.

Question 2: Modeling Patient Coming to hospital -Poisson

Given: Patients arrive in a hospital's emergency room at rate lambda per hour. To model this process as a Poisson Process

To Do: To Study the number of visitors in (0,t]

Solution:

The pmf of a Possion(lambda,t) is given by,
P(X=x)=e^(-lambda*t)((lambda*t)^x)/x!

Let arr be the interarrival time, implying time taken for one arrival

Let N(ti, ti+arr) = #of arrivals in (ti,ti+arr)
= #of arrivals in (0, arr]
P(arr>t)=P(X=0) = e^(-lambda*t)
P(arr<t)=1 - e^(-lambda*t)
Let P(arr<t) be p.
p= 1 - e^(-lambda*delt)
e^(-lambda*delt) = 1-p
-lambda*delt = log(1-p)

delt= -log(1-p)/lambda

So, idea is to keep incrementing the number of arrivals until Delt+totaltime < t, then delt is added to total time and if not, the simulation is terminated and number of arrivals noted. Here p is taken out by using random.random() which gives number between 0 and 1. Hence, fulfilling and carrying out the process.

The attached code captures my idea:

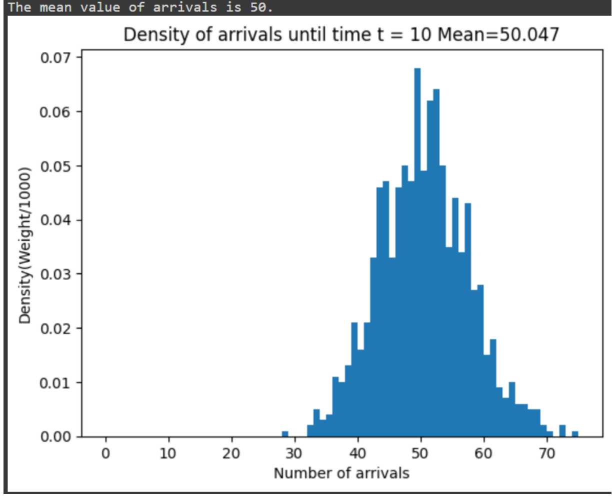
```
def Poisson(lam, t):
    arrivals = []
    lamt = lam*t
    sim=1000
    for i in range(sim):
        n = 0
        start = 0
        while start < t:
            start += -math.log(1 - random.random()) / lam
            if start < t:
                n += 1
        arrivals.append(n)

    # Plot the simulated density
    plt.hist(arrivals, bins=range(max(arrivals) + 2), density=True)
    plt.xlabel('Number of arrivals')
    plt.ylabel('Density(Weight/{})'.format(sim))
    plt.title('Density of arrivals until time t = {} Mean={}'.format(t,sum(arrivals)/sim))

    # Calculate the mean value of arrivals
    mean_arrivals = lam * t

    print('The mean value of arrivals is {}.'.format(mean_arrivals))
```

```
start += -math.log(1 - random.random()) / lam
```

The following is the main crux, wherein delt(interarrival time) is added to start as shown above

Q2.a)

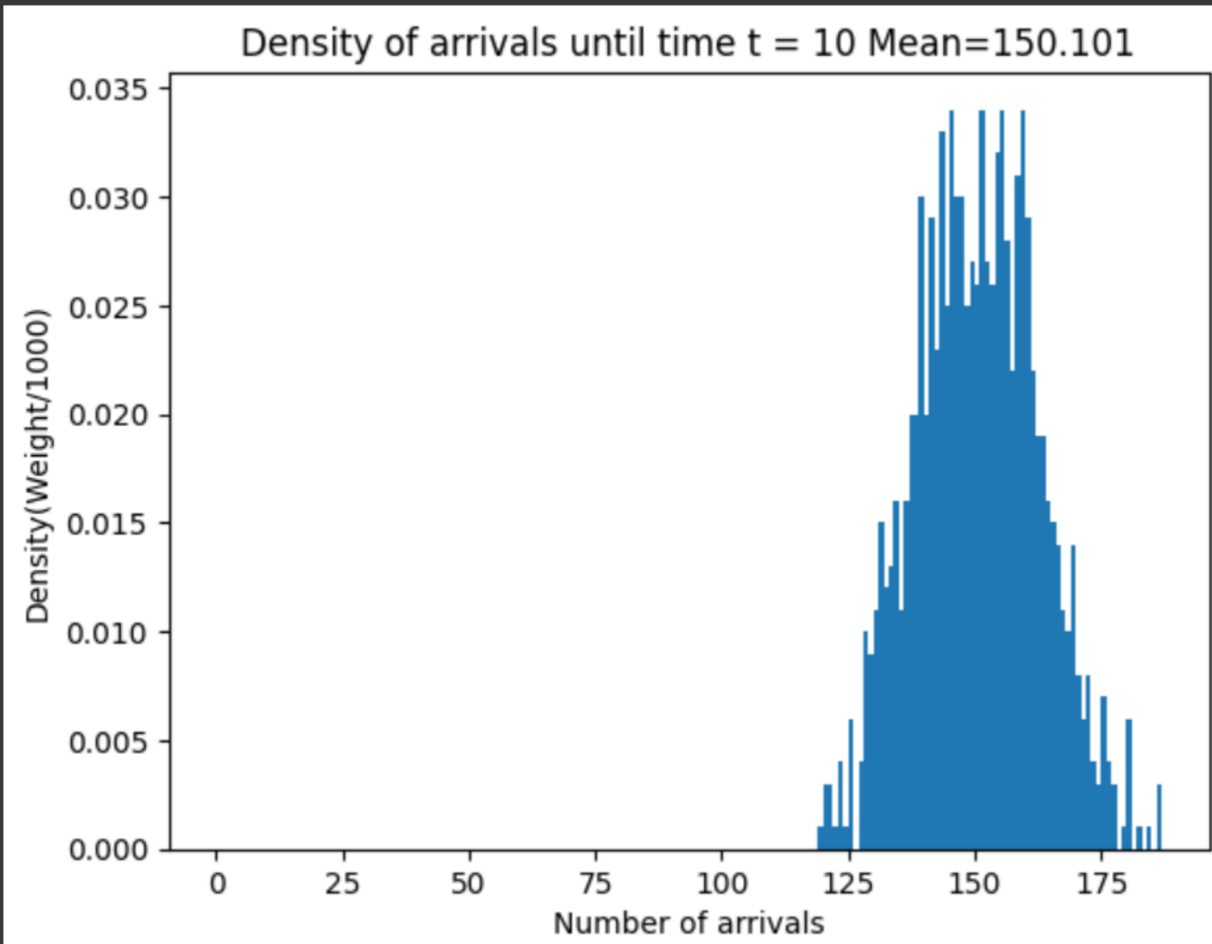
```
1   Poisson(5,10)
The mean value of arrivals is 50.
```



2.b)

```
Poisson(15,10)
```

```
The mean value of arrivals is 150.
```



A stark difference between the 2 graphs is observed, the first with lambda=5 had a major concentration of the number of arrivals near its expected value of 50. While the second, exhibits a similar nature by having concentration of the number of arrivals near its expected value of 150. So, it can conclude that graph depends on the distribution and parameters.

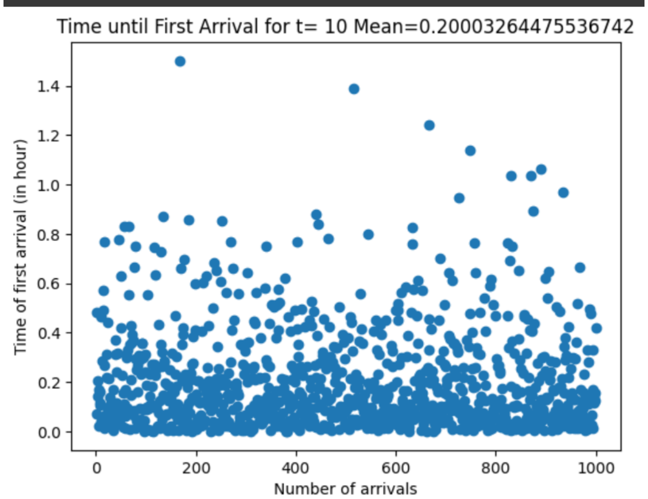
2.c)

In this part the first occurrence of each patient is noted and has been plotted .

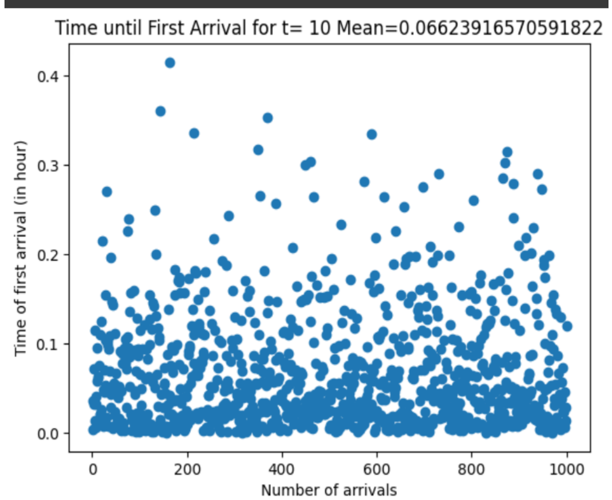
```python
def PoissonFirstArrival(lam, t):
    arrivals = []
    lamt = lam*t #Mean
    index=[]
    for i in range(1000):
        index.append(i+1)
        n = 0
        start = 0
        while start < t:
            start += -math.log(1 - random.random()) / lam #Calculating delt
            if start < t:
                n += 1
                break
        arrivals.append(start) #Appending it to array/list
    # plt.hist(arrivals, bins=range(max(arrivals) + 2), density=True)
    return arrivals,index
```

Corresponding Scatter Plots are plotted for lambda=5,15 and t-10 in both

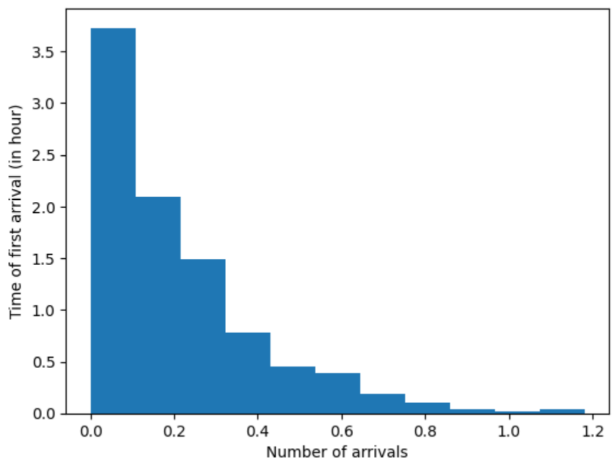
```python
1  PoissonFirstArrival(5, 10)
```

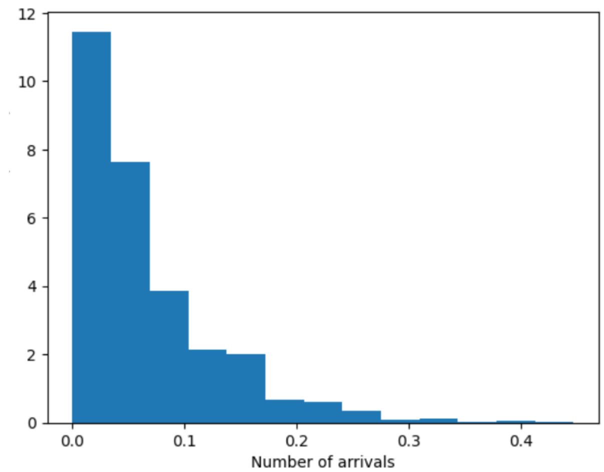



```python
1  PoissonFirstArrival(15, 10)
```



The scatter plots seem similar, however, there is a huge difference in the time of first arrival and mean of time of first arrival.
Corresponding Histograms are plotted for lambda=5,15 and t-10
Since, it is difficult to interpret the scatter plot, we view them as histograms





Now, we infer that both are decreasing curves(of the form of exponentially decreasing) like they should be.

The Assignment file consists of the report, code files for questions 1 and 2, both of which are in python and my jupyter notebook in which I worked. The graphs are the same as attached in the report and jupyter notebook.

This is the Original Work of Chaitanya Garg(2021248).
Discussed with: Tanishq Jain, Raghav Sakhuja