Assignment 1 Report

Srimant Mohanty
2021207
srimant21207@iiitd.ac.in

Q1. **Coin Toss**

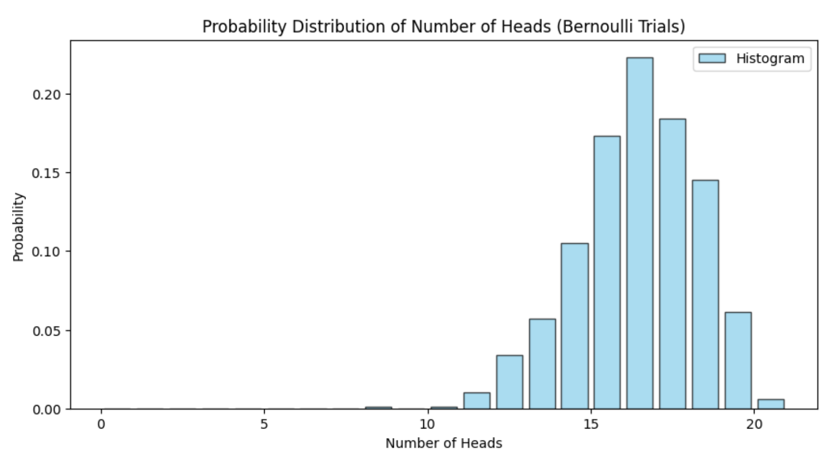
Tossing a coin 20 times and observing the possibility of obtaining a Head(Success) or Tail(Failure) can be modeled as Bernoulli Process where let p be the probability of obtaining a Head and (1-p) or q be the probability of obtaining a Tail in a particular trial.

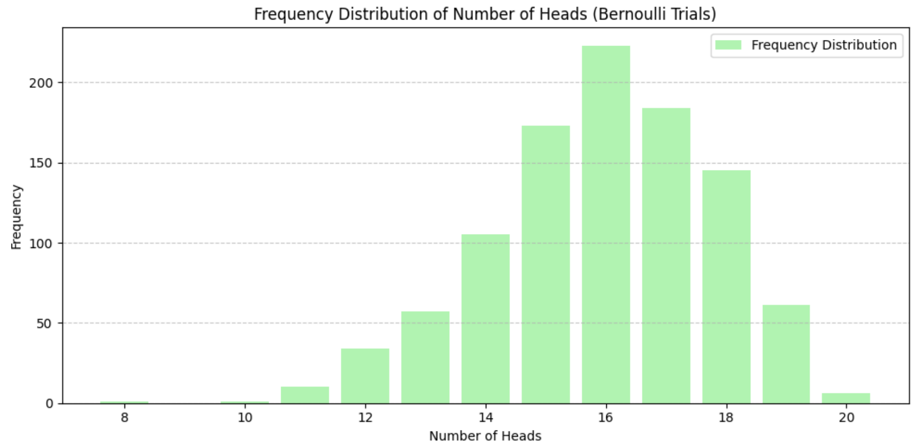
The model is following all the assumptions of the Bernoulli Process since:-

1) Tossing a coin is a discrete time and discrete state process; $Z_t = 0$ or $Z_t = 1$ (heads or tails).
2) Z_t is an independent and identically distributed random variable. Tossing a coin is a series of Bernoulli trials each of which is independent of the previous/other trial belonging to the same Bernoulli distribution.
3) There are only 2 possible outcomes of each trial; (Heads/Success) or (Tails/Failure). Probability of success is the same in every trial.
4) Process starts at t=0 but arrival happens at t>=1 only. Heads/Tails can be obtained after t=1 only.
5) Only one arrival can happen at any given point of time.

a) We take the probability of success(heads) as p=0.8 in this particular case and simulate the process of flipping the coin 20 times each for 1000 times. Then, we count the number of heads obtained out of the 20 coin tosses for every simulation.

We have made the following 2 plots for this simulation:-





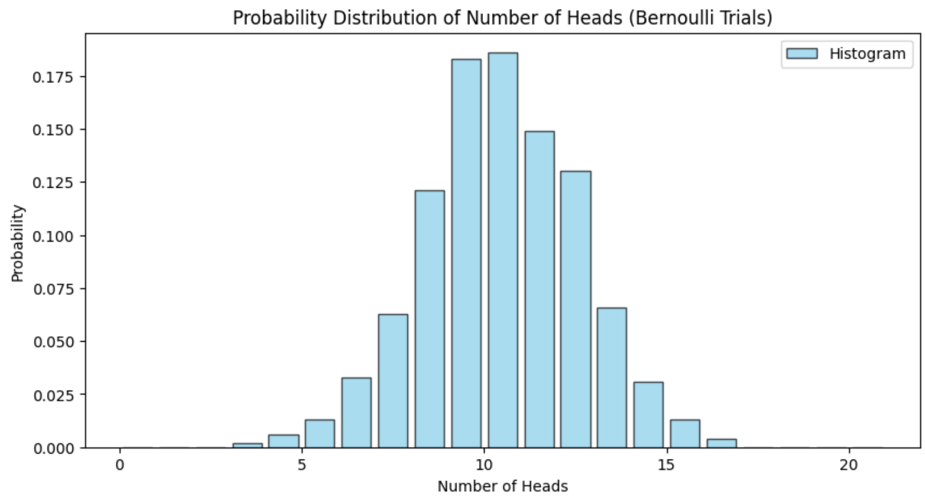
Observations:

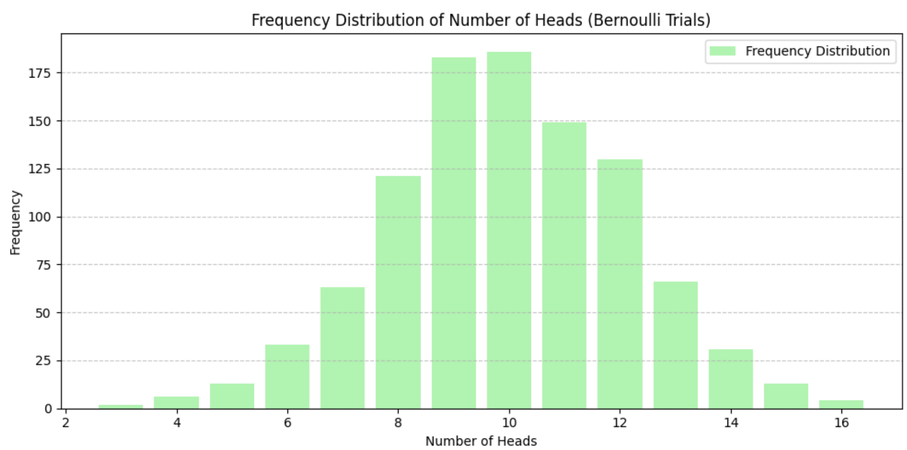
- The first plot is that of the probability distribution of the number of heads obtained i.e the distribution of the count of the number of heads obtained during the process(20 coin tosses)
- The most likely value of the number of heads obtained in a process is 16, which is similar to the expected value in the binomial distribution i.e $E[X] = np = 20 \times 0.8 = 16$.
- The second plot is that of frequency distribution/count of number of heads obtained in each process which again shows the expected value as 16.

Code:

```
[10] import numpy as np
     import matplotlib.pyplot as plt
     import math
     import random

     def bernoulli(p):
         if random.random() < p:
             return 1
         else:
             return 0

     p=0.8
     n=1000
     heads=[]
     for i in range(n):
         head=0
         for j in range(20):
             head+=bernoulli(p)
         heads.append(head)

     plt.figure(figsize=(10, 5))
     plt.hist(heads, bins=range(0, 20+2), density=True, rwidth=0.8, color='skyblue', edgecolor='black', alpha=0.7, label='Histogram')
     plt.xlabel('Number of Heads')
     plt.ylabel('Probability')
     plt.title('Probability Distribution of Number of Heads (Bernoulli Trials)')
     plt.legend()
     plt.figure(figsize=(10, 5))
```

b) We now change the probability of success to p = 0.5 and get the following plots:-





Observations:

- After decreasing the probability of success(p) to p = 0.5 we observe that the plot has shifted leftwards i.e fewer number of heads obtained in each process.
- However, this value is in agreement with the expected value i.e E[X] = np = 10. The highest probability of the number of heads that can be obtained in one process is that of 10 heads.
- Similarly, the next plot shows the frequency distribution, which is maximum for 10 heads.

Code:

```
p=0.5
n=1000
heads=[]
for i in range(n):
    head=0
    for j in range(20):
        head+=bernoulli(p)
    heads.append(head)

plt.figure(figsize=(10, 5))
plt.hist(heads, bins=range(0, 20+2), density=True, rwidth=0.8, color='skyblue', edgecolor='black', alpha=0.7, label='Histogram')
plt.xlabel('Number of Heads')
plt.ylabel('Probability')
plt.title('Probability Distribution of Number of Heads (Bernoulli Trials)')
plt.legend()

head_total = {}
for head in heads:
    if head in head_total:
        head_total[head] += 1
    else:
        head_total[head] = 1
counts = list(head_total.keys())
frequencies = [head_total[count] for count in counts]

plt.figure(figsize=(10, 5))
plt.bar(counts, frequencies, width=0.8, align='center', color='lightgreen', alpha=0.7, label='Frequency Distribution')
plt.xlabel('Number of Heads')
plt.ylabel('Frequency')
plt.title('Frequency Distribution of Number of Heads (Bernoulli Trials)')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend()

plt.tight_layout()
plt.show()
```

Q2.

The number of patients arriving in the hospital's emergency room in a time interval (0, t] at the rate 5 patients per hour can be modeled as a poisson process. It satisfies all the following assumptions necessary for a process to be modeled as a poisson process:-

1. The rate (λ) at which arrivals are happening is constant and not evolving over time.

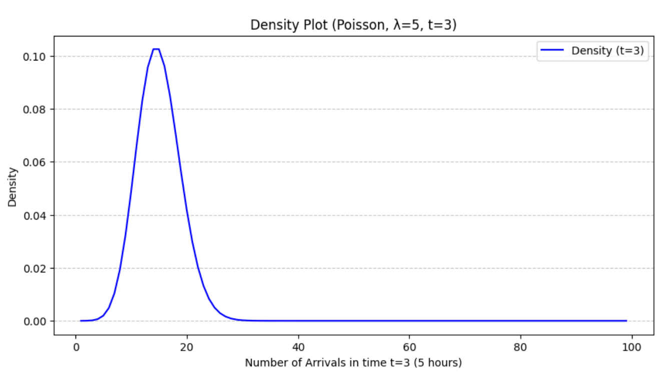
2. The arrivals are independent of each other i.e the arrivals taking place in one particular interval is independent of the arrivals occurring in another non-overlapping interval.

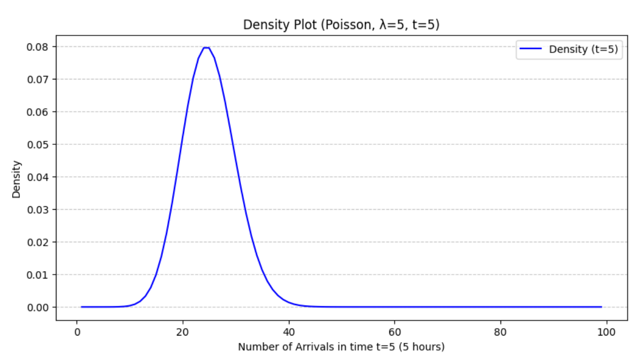
3. More than 1 arrival cannot occur simultaneously i.e at a given point of time t only one particular arrival can take place.

Hence, this process can be modeled as a poisson process.

a) This part asks us to simulate the density of the number of arrivals in a given time t or in an interval (0, t] and $\lambda = 5$. Here we choose to simulate the process for two values of t, i.e $t = 3$ & $t = 5$ (in hours).

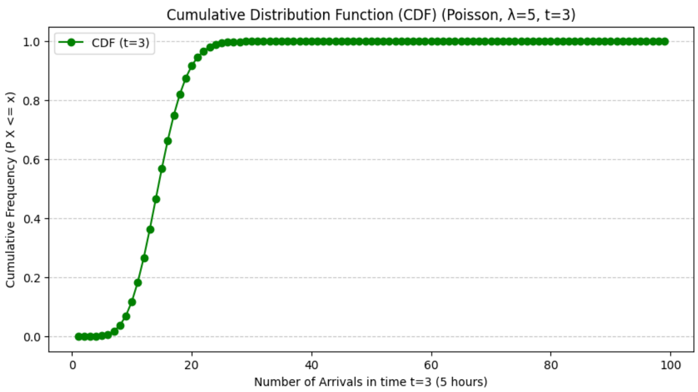
The following are the plots for the density of the number of arrivals in time t=3 & t=5 :-



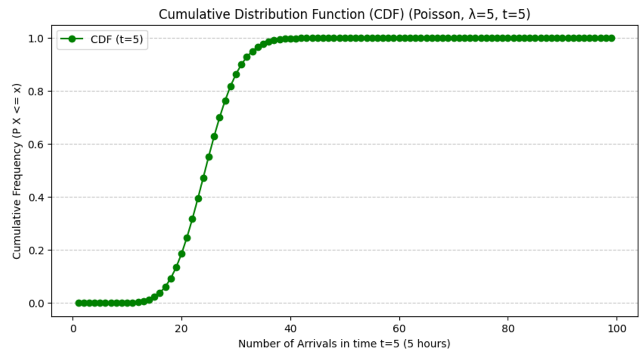


Observations:

- For t=3 the expected value of the number of arrivals in 5 hours is equal to 15, the density is maximum at around 15 arrivals.
- For t=5 the expected value of the number of arrivals in 5 hours is equal to 25, the density is maximum at around 25 arrivals.
- This is in agreement with the expectation of poisson process in time t i.e $E[X] = \lambda t$.



f the

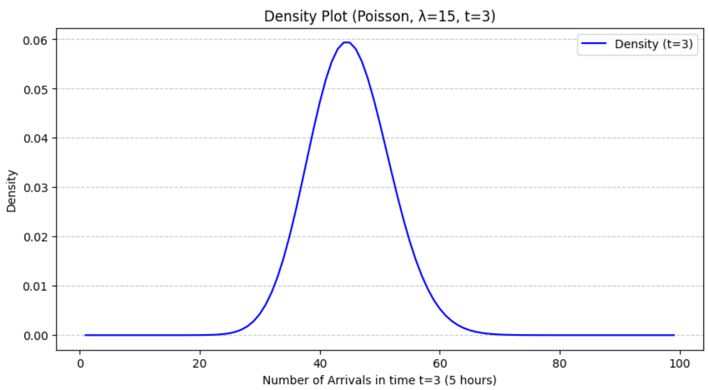


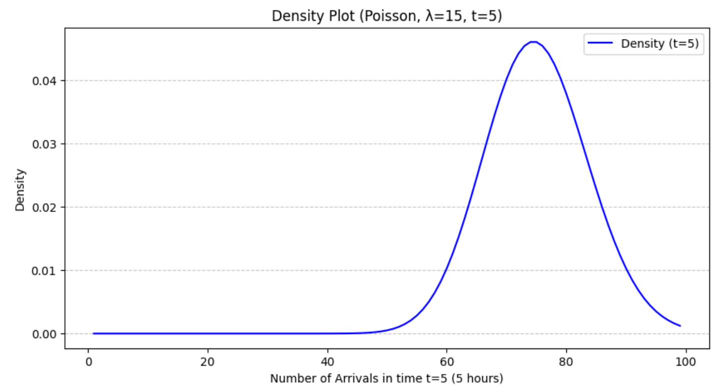
Code:

```
import numpy as np
import matplotlib.pyplot as plt
import math
import random

def poiss(lam,x,t):
    a=lam*t
    b=a**x
    c=math.exp(-a)
    d=math.factorial(x)
    return (b*c)/d

lam=5
time=[3,5]
for t in time:
  x=[i for i in range(1,100)]
  y=[poiss(lam,i,t) for i in x]
  plt.figure(figsize=(10, 5))
  plt.plot(x, y, color='blue', label=f'Density (t={t})')
  plt.xlabel(f'Number of Arrivals in time t={t} (5 hours)')
  plt.ylabel('Density')
  plt.title(f'Density Plot (Poisson, λ=5, t={t})')
  plt.grid(axis='y', linestyle='--', alpha=0.7)
  plt.legend()
  plt.show()

  cumulative_frequencies = np.cumsum(y)

  plt.figure(figsize=(10, 5))
  plt.plot(x, cumulative_frequencies, marker='o', linestyle='-', color='green', label=f'CDF (t={t})')
  plt.xlabel(f'Number of Arrivals in time t={t} (5 hours)')
  plt.ylabel('Cumulative Frequency (P X <= x)')
  plt.title(f'Cumulative Distribution Function (CDF) (Poisson, λ=5, t={t})')
  plt.grid(axis='y', linestyle='--', alpha=0.7)
  plt.legend()
  plt.show()
```

b) In this part we increase the rate of arrivals or λ to λ = 15 and then simulate the density of the number of arrivals until a time t i.e t = 3 & t = 5 in our case.

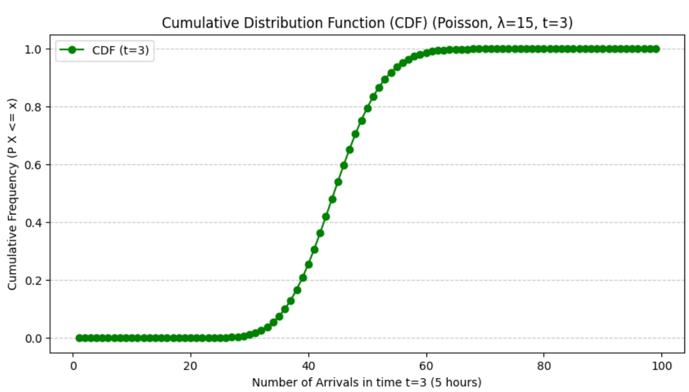
The following are the plots for the density of the number of arrivals in time t=3 & t=5 :-

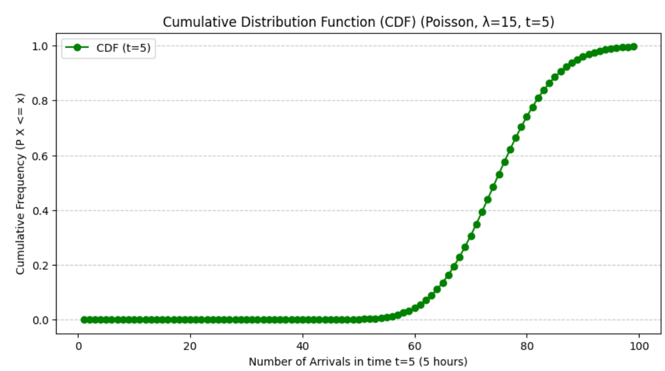




Observations:

- For t=3 the expected value of the number of arrivals in 5 hours is equal to 45, the density is maximum at around 45 arrivals.
- For t=5 the expected value of the number of arrivals in 5 hours is equal to 75, the density is maximum at around 75 arrivals.
- This is because on increasing the value of λ or increasing the arrival rate leads to an increase in the total number of arrivals in a particular time interval.
- This is in agreement with the expectation of poisson process in time t i.e $E[X] = \lambda t$.





The above plots show the cumulative distribution of the number of arrivals i.e the number of arrivals until a given time t=x such that x is less than equal to t.

Code:

```
import numpy as np
import matplotlib.pyplot as plt
import math
import random

def poiss(lam,x,t):
    a=lam*t
    b=a**x
    c=math.exp(-a)
    d=math.factorial(x)
    return (b*c)/d

lam=15
time=[3,5]
for t in time:
  x=[i for i in range(1,100)]
  y=[poiss(lam,i,t) for i in x]
  plt.figure(figsize=(10, 5))
  plt.plot(x, y, color='blue', label=f'Density (t={t})')
  plt.xlabel(f'Number of Arrivals in time t={t} (5 hours)')
  plt.ylabel('Density')
  plt.title(f'Density Plot (Poisson, λ=15, t={t})')
  plt.grid(axis='y', linestyle='--', alpha=0.7)
  plt.legend()
  plt.show()

  cumulative_frequencies = np.cumsum(y)

  plt.figure(figsize=(10, 5))
  plt.plot(x, cumulative_frequencies, marker='o', linestyle='-', color='green', label=f'CDF (t={t})')
  plt.xlabel(f'Number of Arrivals in time t={t} (5 hours)')
  plt.ylabel('Cumulative Frequency (P X <= x)')
  plt.title(f'Cumulative Distribution Function (CDF) (Poisson, λ=15, t={t})')
  plt.grid(axis='y', linestyle='--', alpha=0.7)
  plt.legend()
  plt.show()
```

c)

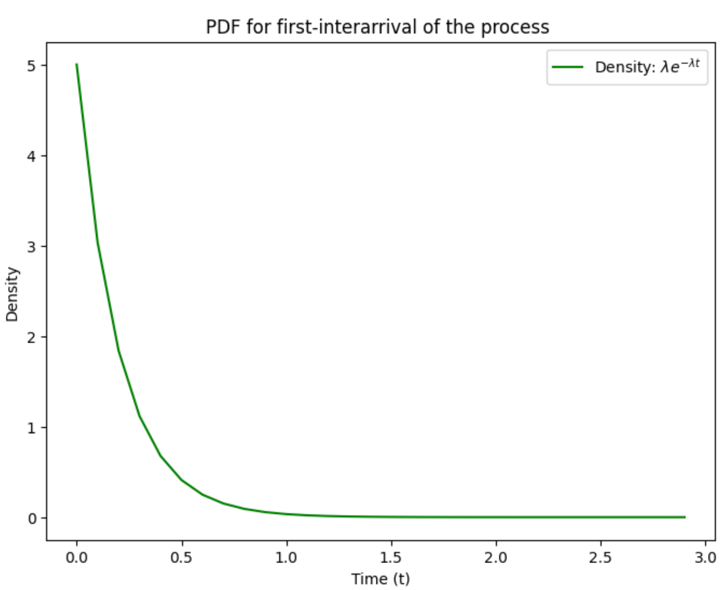
In this part we need to simulate the first inter-arrival process, say X1 be the first inter arrival time. We know that X1 ~ exponential(λ).

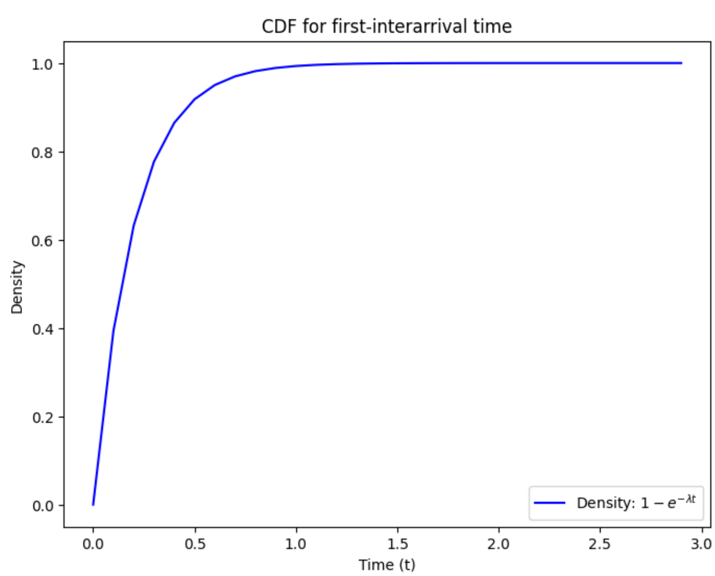
Thus PDF for inter-arrival time is given by $\lambda e^{(-\lambda t)}$ where t is greater than equal to 0.

Similarly its CDF is is given by 1 - $e^{(-\lambda t)}$ where t is greater than equal to 0.

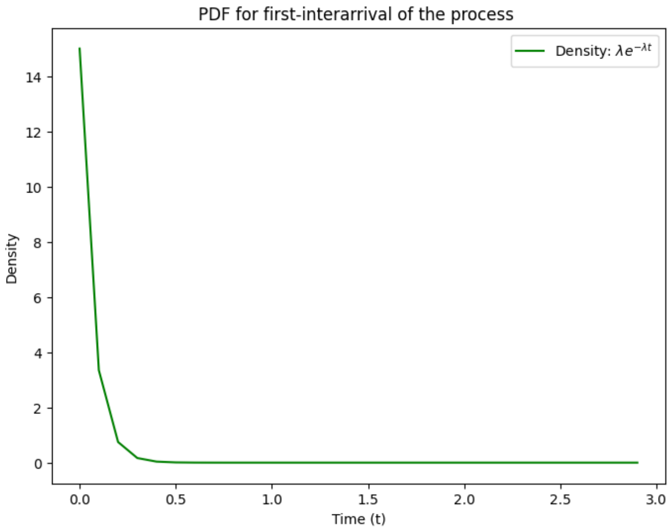
Thus we get the following plots;

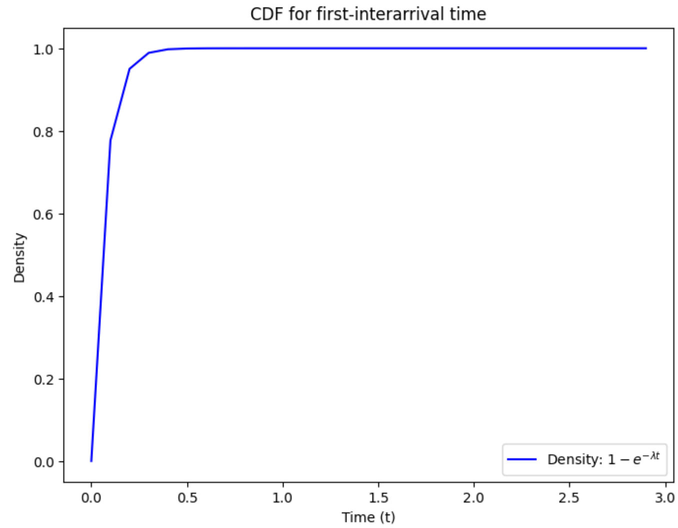
For λ = 5,





For λ = 15,





Observation:

- The first interarrival time follows an exponential distribution with parameter λ.
- As λ increase i.e arrival rate increases, the first inter-arrival time decreases.
- This is in agreement with the known result that expected value of exponential distribution is E[X] = 1/λ.

Code:

```
import numpy as np
import matplotlib.pyplot as plt
import math
import random

def expo1(lam,t):
    a=lam*t
    b=math.exp(-a)
    c=1-b
    return c

def expo2(lam,t):
    a=math.exp(-lam*t)
    b=lam*a
    return b

l=[5,15]

for lam in l:

  t=[i for i in np.arange(0,3,0.1)]
  y=[expo2(lam,i) for i in t]

  plt.figure(figsize=(8, 6))
  plt.plot(t, y, label='Density: $\lambda e^{-\lambda t}$', color='green')
  plt.xlabel('Time (t)')
  plt.ylabel('Density')
  plt.title('PDF for first-interarrival of the process ')
  plt.legend()
  plt.show()

  #increment t by 0.1
  t=[i for i in np.arange(0,3,0.1)]
  y=[expo1(lam,i) for i in t]

  plt.figure(figsize=(8, 6))
  plt.plot(t, y, label='Density: $1 - e^{-\lambda t}$', color='blue')
  plt.xlabel('Time (t)')
  plt.ylabel('Density')
  plt.title('CDF for first-interarrival time')
  plt.legend()
  plt.show()
```