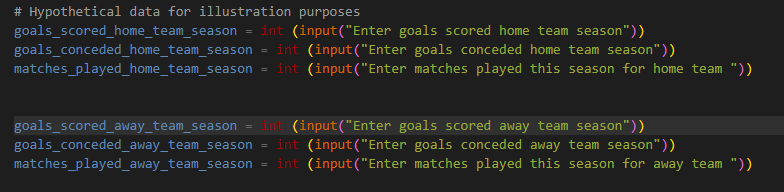
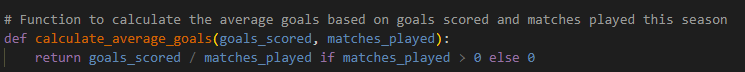
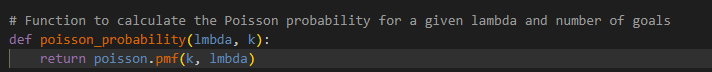
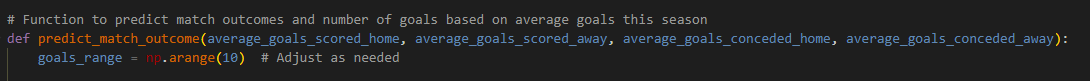
**Predict Football Match Outcomes**

1- Data Input

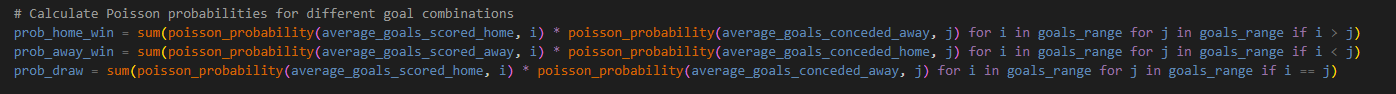
The user is prompted to input data for both home and away teams, including the goals scored, matches played, goals conceded, etc.

2- **Functions**

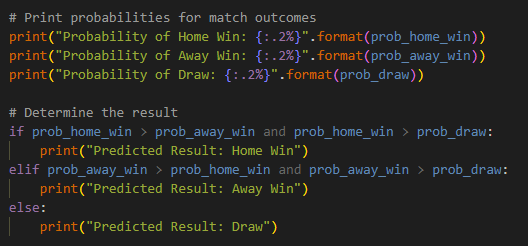
* **calculate\_average\_goals**: Calculates the average goals per match based on the total goals scored and matches played.
* **poisson\_probability**: Calculates the Poisson probability for a given lambda and number of goals using the **pmf** (probability mass function) from the **scipy.stats.poisson** module.
* **predict\_match\_outcome**: Predicts match outcomes and the number of goals based on the average goals scored and conceded by both home and away teams.

**3-Prediction:**

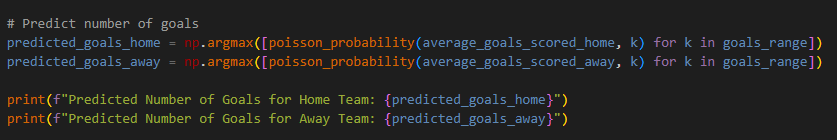
* The script calculates the average goals scored and conceded for both the home and away teams using the provided data.
* It then uses the Poisson distribution to calculate the probabilities of different goal combinations for home and away teams.



* The script prints the probabilities of a home win, away win, and draw, and predicts the match result based on the highest probability.



* Additionally, it predicts the number of goals for both the home and away teams.



Predict the time when an Earthquake might occur.

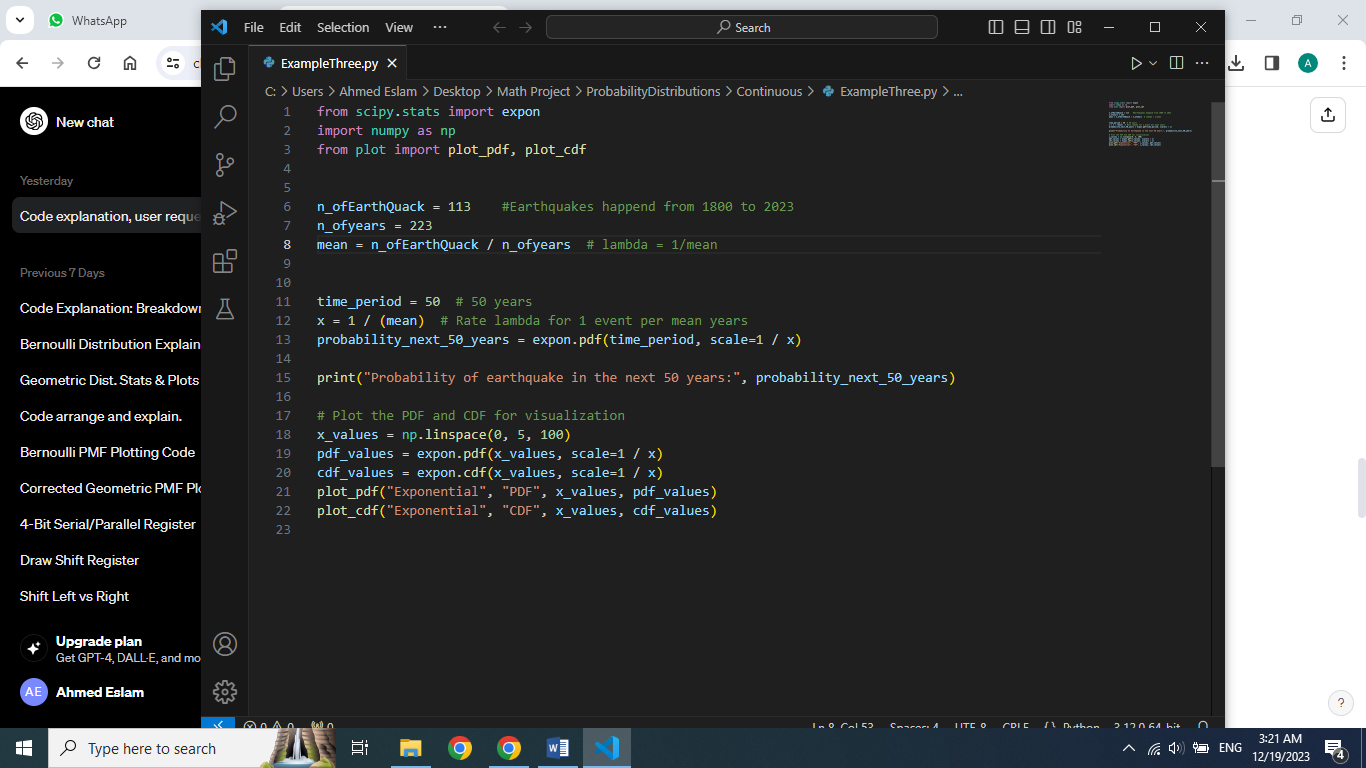
The exponential distribution is often concerned with the amount of time until some specific event occurs. For example, the amount of time until an earthquake occurs has an exponential distribution.

A group of people standing on a pile of rubble

Description automatically generated

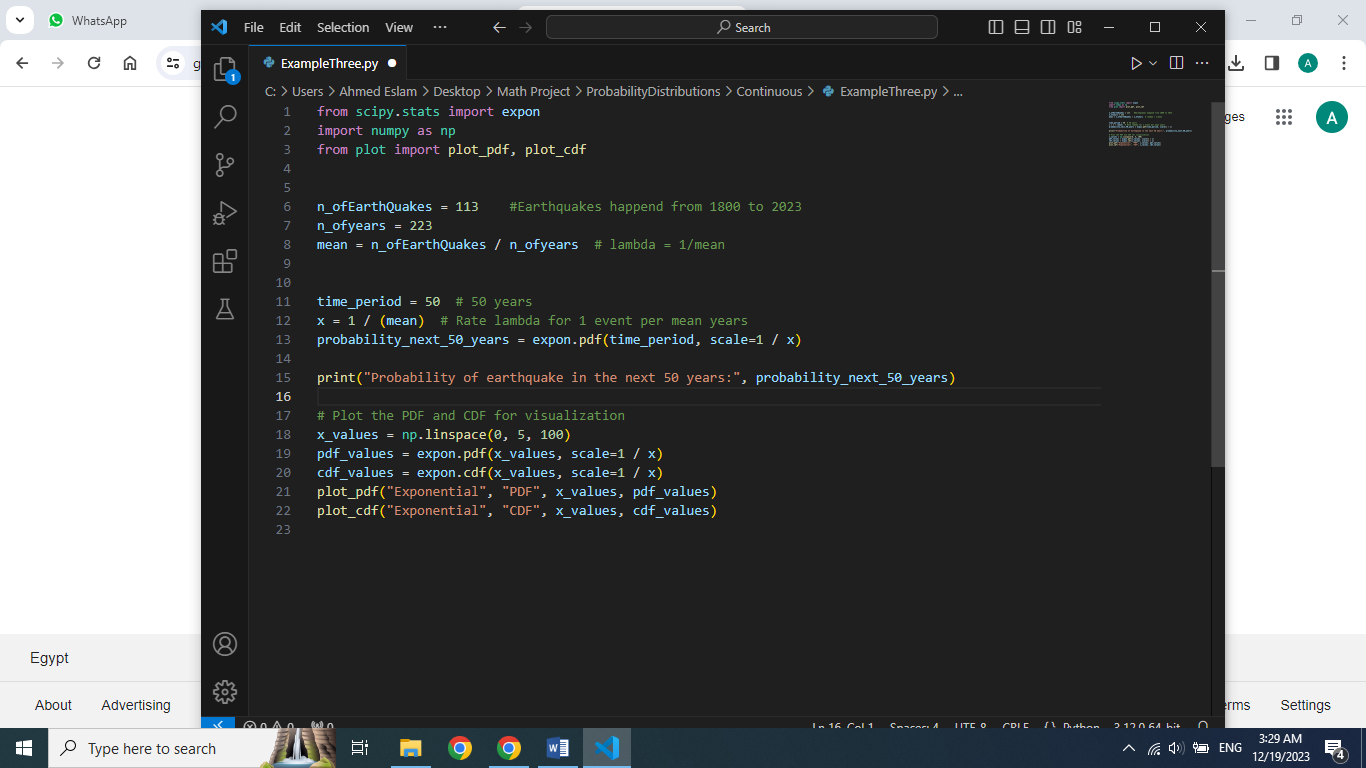
**Explaining Code in details:**

1. Importing needed libraries for exponential distribution



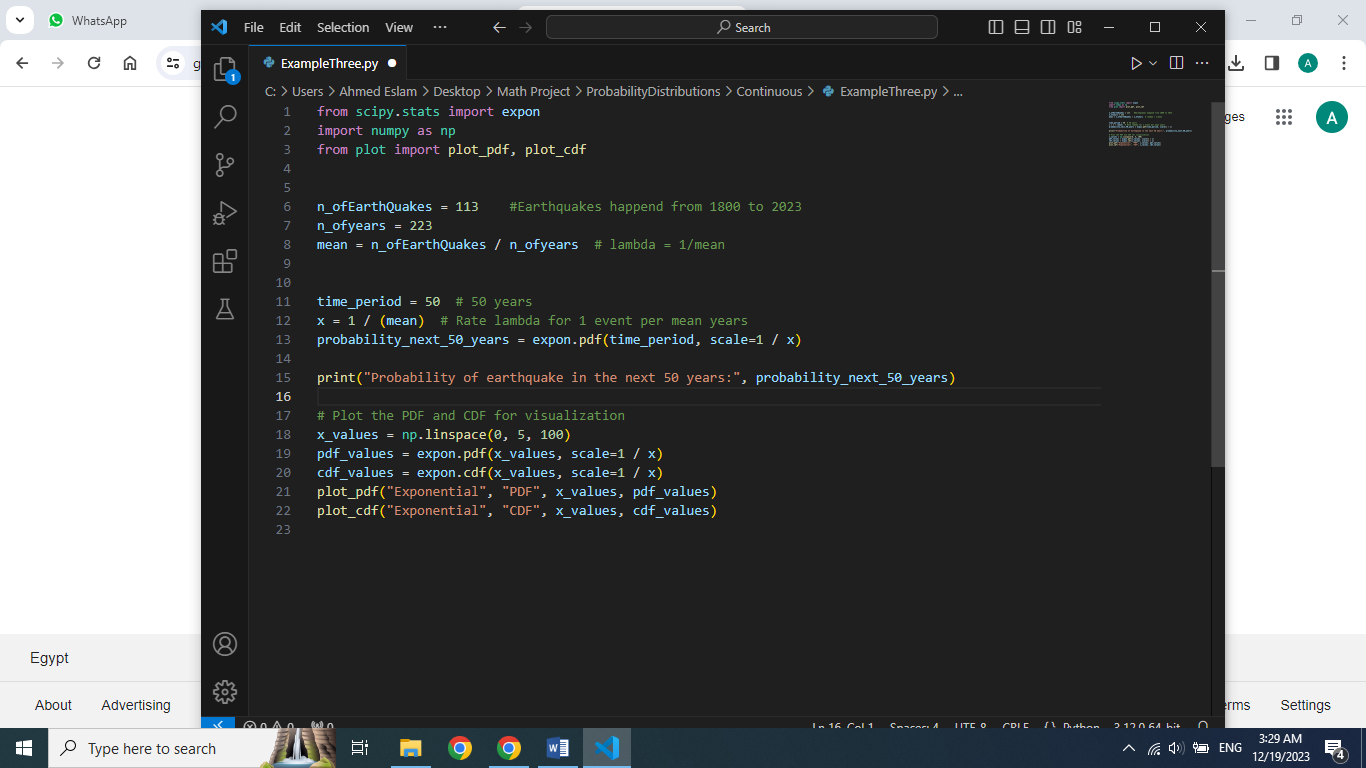
From "scipy.stats" we imported 'expon' for exponential distribution, "numpy" as 'np' for numerical operations and from "plot" we used 'plot\_pdf', 'plot\_cdf' for graphing functions.

2. Setting parameter for the Exponential distribution



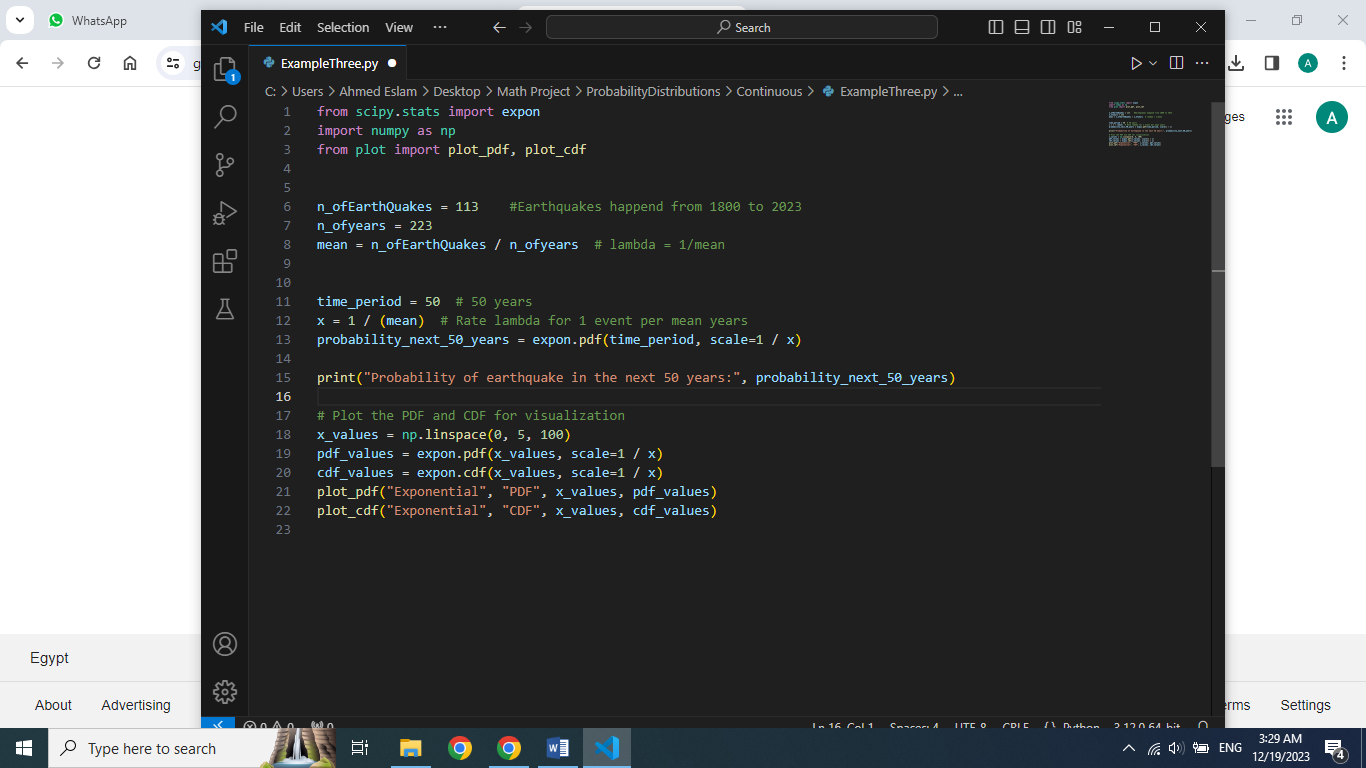
Mean is the rate of earthquakes per year, which is used as the rate parameter lambda for exponential distribution

3. Setting variables, calculating PDF and Printing Probability



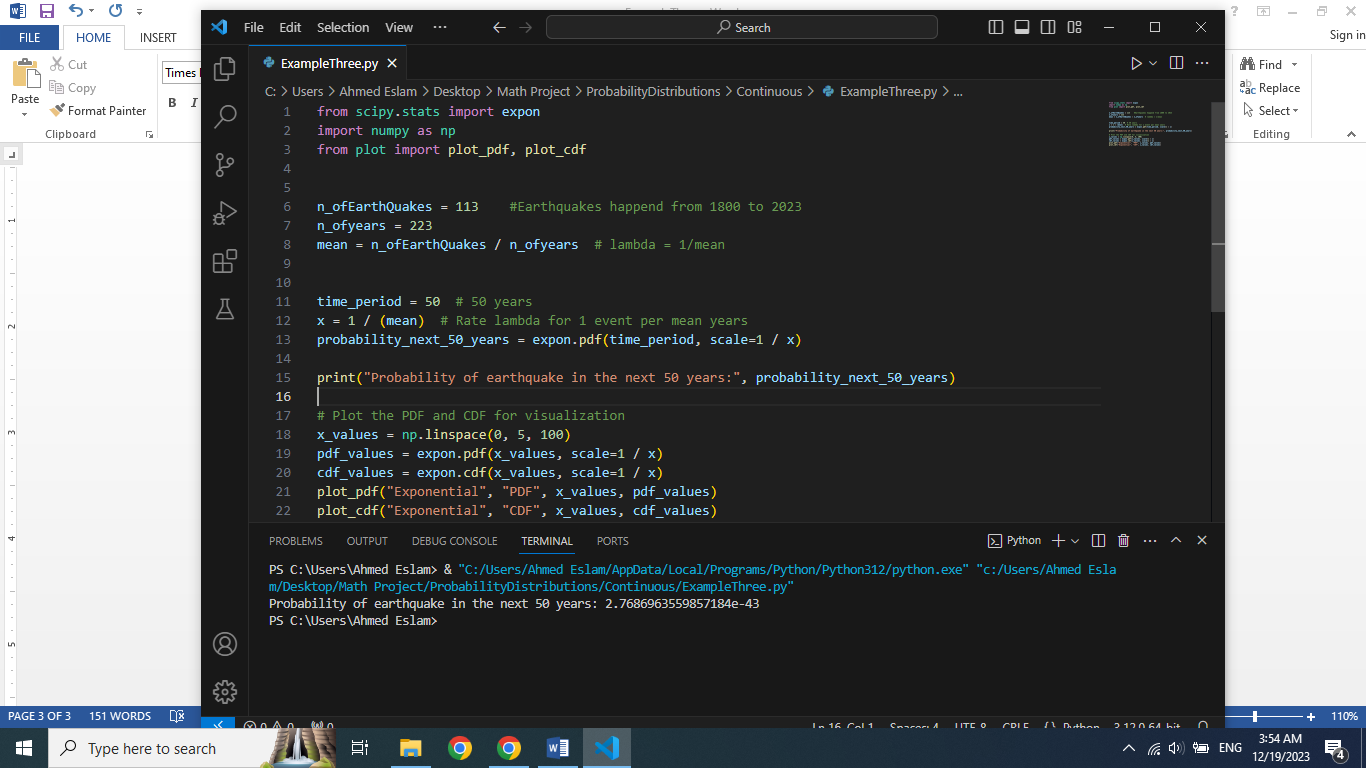
The variable "time\_period" is set to 50 years and then calculating x parameter and use "expon.pdf" for the Probabilty Density Function for the 50 years last printig it.

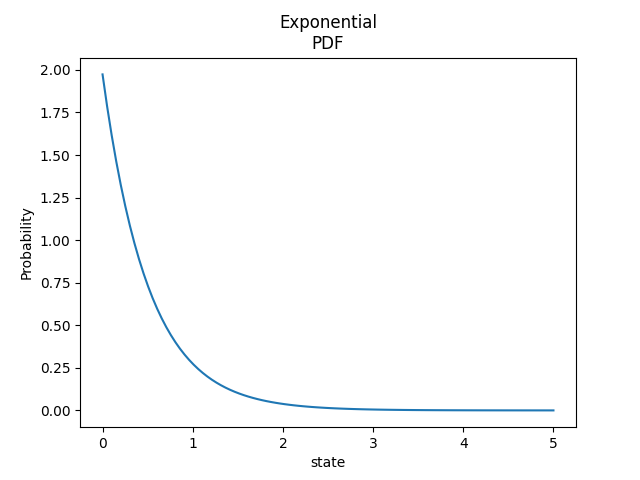
4. Graphing PDF and CDF

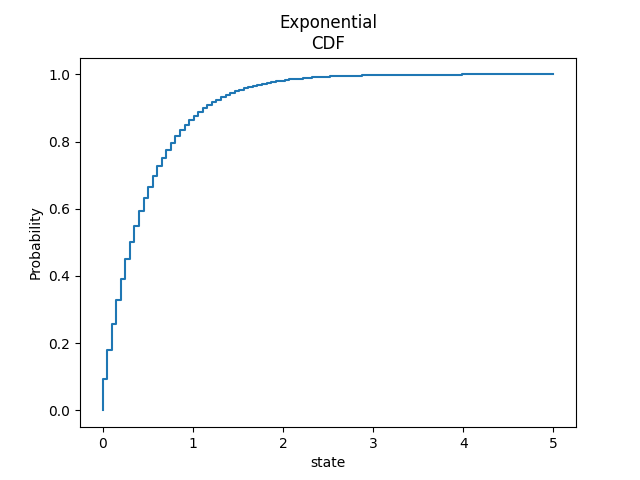


Output

1. Probability of occurrence Earthquake in the next 50 years



2. PDF and CDF Graphes



**Traffic density of street**

If the average number of cars that cross a particular street in a day is 25, then you can find the probability of 28 cars passing the street using the poisson formula.

* *X* is the random variable representing the number of events,
* *k* is the specific number of events we're interested in (in this case, 28 cars passing the street),
* *λ* is the average rate of events per interval.

In your example:

* *λ*=25 (average number of cars passing the street in a day),
* *k*=28 (the number of cars you want to find the probability for).

Plugging in these values, the probability of observing exactly 28 cars passing the street in a day is:

*P*(*X*=28)= *e*−25⋅2528​/28!

import numpy as np

import matplotlib.pyplot as plt

import time

from scipy.stats import poisson

def print\_pause(message):

print(message)

time.sleep(2)

def poisson\_drv():

# Set the parameter

rate\_poisson = 2 # Average rate (lambda) for the Poisson distribution

print("A service center receives an average of 2 customers per hour.")

print\_pause("What is the probability of receiving at most 3 customers in the next hour?")

print\_pause("We can use the Poisson random variable for this scenario.")

print\_pause(f"The average rate (lambda) is {rate\_poisson} customers per hour.")

# Generate Poisson random variable

poisson\_rvs = poisson.rvs(mu=rate\_poisson, size=1000)

# Calculate PMF and CDF

k\_values = np.arange(0, 10) # Adjust the range based on your scenario

pmf\_values = poisson.pmf(k\_values, mu=rate\_poisson)

cdf\_values = poisson.cdf(k\_values, mu=rate\_poisson)

# Calculate mean and variance

mean\_poisson = poisson.mean(mu=rate\_poisson)

variance\_poisson = poisson.var(mu=rate\_poisson)

# Display mean and variance

print\_pause(f"The MEAN value = {mean\_poisson}")

print\_pause(f"The VARIANCE value = {variance\_poisson}")

cdf = poisson.cdf(3, mu=rate\_poisson)

print\_pause("To calculate the probability of receiving at most 3 customers in the next hour,")

print\_pause(f"We must calculate the CDF of 3, which is approximately {cdf:.4f}")

time.sleep(4)

print\_pause("Now let's visualize the PMF and CDF:")

time.sleep(3)

show\_x = int(input("Show? (1 for Yes, 0 for No): "))

if show\_x == 1:

plt.figure(figsize=(12, 6))

plt.subplot(121)

plt.plot(k\_values, pmf\_values, "bo", ms=8, label="Poisson PMF")

plt.vlines(k\_values, 0, pmf\_values, colors="b", lw=5, alpha=0.5)

plt.title('Poisson Distribution PMF')

plt.xlabel('k')

plt.ylabel('Probability')

plt.subplot(122)

plt.step(k\_values, cdf\_values, where='post')

plt.title('Poisson Distribution CDF')

plt.xlabel('k')

plt.ylabel('Probability')

plt.tight\_layout()

plt.show()

print("Choose the number of Example that you want : ")

print("1- Binomial random variable ")

print("2- Poisson random variable ")

choose = int(input("Choose: "))

if choose == 1:

binomial\_drv()

elif choose == 2:

poisson\_drv()

else:

A graph of a graph of a graph

Description automatically generated with medium confidence print("Invalid choice. Please choose 1 or 2.")