Problem Statement: Consider a random vector, a collection of random values of n-dimension [min ‘n’ = 100]. Fit Poisson's distribution and Gaussian distribution for these collected values.

```
# import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats


# Generate random values for an n-dimensional vector (n >= 100)
n = 150     #n>=100
random_vector = np.random.randint(1, 100, n)   # Generate an n-dimensional array of random integers between 1 and 100
np.random.seed(42)   # For reproducibility


# Fit Poisson distribution
lambda_poisson = np.mean(random_vector)   # Poisson distribution is parameterized by its mean (λ)


# Fit Gaussian (Normal) distribution
mu, std = np.mean(random_vector), np.std(random_vector)   # Gaussian is parameterized by mean and std dev(standard deviation)


# Visualization
# Plot histogram of the random vector
plt.hist(random_vector, bins=20, density=True, alpha=0.6, color='g', label='Data')
```

#creates a histogram of the random_vector data with 20 bins, normalizes the height to form a probability density(Probability density is a way to show how likely it is for a value to occur, with higher values meaning more likely, and the total area under the curve always adding up to 1.),

#adds 60% transparency, sets the color to green, and labels it as 'Data'.

Generate points for plotting fitted Poisson and Gaussian curves

x = np.arange(0, 150) #creates an array of values from 0 to 149 to use as the x-axis for plotting

poisson_pmf = stats.poisson.pmf(x, lambda_poisson) # Poisson PMF,calculates the Poisson probability mass function (PMF) at each value in x using lambda_poisson

#The Poisson probability mass function (PMF) gives the probability of a given number of events happening in a fixed interval of time or space,

#assuming the events occur independently and at a constant average rate (λ). It's used for counting events, like the number of phone calls a call center gets in an hour

gaussian_pdf = stats.norm.pdf(x, mu, std) # Gaussian PDF,calculates the Gaussian (Normal) probability density function (PDF) at each value in x using the mean mu and standard deviation std

#A probability density function (PDF) shows how likely different values of a continuous random variable are,

#with the area under the curve representing the probability of the variable falling within a specific range. The total area under the curve equals 1

Plot Poisson fit

plt.plot(x, poisson_pmf, 'bo', ms=9, label='Poisson Fit (λ={:.2f})'.format(lambda_poisson), alpha=0.6)

#plots the Poisson probability mass function as blue circles ('bo') at the x-values, sets the marker size to 8,

#labels the plot with the calculated λ value, and adds 60% transparency to the points

Plot Gaussian fit

plt.plot(x, gaussian_pdf, 'r-', lw=3, label='Gaussian Fit (μ={:.2f}, σ={:.2f})'.format(mu, std), alpha=0.6)

#plots the Gaussian probability density function as a red line ('r-') at the x-values, sets the line width to 2, labels the plot with the calculated mean (μ) and standard deviation (σ),

#and adds 60% transparency to the line

Add labels and legend

plt.xlabel('Value')

plt.ylabel('Probability Density')

plt.legend(loc='best')   #adds a legend to the plot at the best location, automatically determining the most suitable position to avoid overlapping with the plotted data

plt.title('Fitting Poisson and Gaussian Distributions')

Show the plot

plt.show()