DS201/DSL253: Statistical Programming Assignment 3

Shashank yadav **12342010**

Question 1-

Introduction-

In this report, we witness the change in the nature of binomial distribution through different scenarios that simulate marketing campaigns, we will examine:

1. The probability mass function (PMF) for a fixed conversion rate of 20% with 15 potential customers.
2. The variation in the PMF for different conversion probabilities (10%, 20%, 30%, ... 90%).
3. The distribution of success rates for various campaign sizes(10,20,50,100), all with a fixed conversion probability of 5%.

Data-

The data used in this study are simulated using the binomial distribution, with the following parameters:

1. **For part (a)**:
   - Number of potential customers: n=15
   - Conversion probability: p=0.2
2. **For part (b)**:
   - Number of potential customers: n=15 (same as in part a)
   - Conversion probabilities: p=0.1,0.2,0.3,…,0.9

3. **For part (c)**:
   - Conversion probability: p=0.05
   - Campaign sizes: n=10,20,50,200

Methodology-

In this we have created a binomial distribution model with different parameters which are:

Where: (math.comb(k,i))*(success**(i) * (1-success)**(k-i))

- k is the number of potential customers (trials),
- success is the probability of a successful conversion (success probability),
- i is the number of successful conversions (ranging from 0 to n).

Steps:

1. **Part (a)**: We compute and plot the PMF for a fixed conversion rate of 20% with 15 customers.
2. **Part (b)**: We plot the PMF for different conversion rates (from 10% to 90%) using the same number of customers.
3. **Part (c)**: We calculate and plot the distribution of success rates (number of successes divided by n) for different campaign sizes (10, 20, 50, and 200 customers) with a fixed conversion probability of 5%.

In each case, we use the `matplotlib.pyplot` to visualize the results.

```
success=0.2
pmf=[]
for i in range(15):
    prob=(math.factorial(15)/math.factorial(15-i-1))*(success**(i+1) * (1-success)**(15-i-1))
    pmf.append(prob/math.factorial(i+1))
plt.bar(np.arange(1,16),pmf,)
plt.xlabel(" the number of successful responses")
plt.ylabel("probability")
plt.title("probability mass distribution for success response 20%")
plt.show()
print(sum(pmf))
```

```
11]: for k in [10,20,50,200]:
         success=0.05
         pmf=[]
         for i in range(k+1):
             prob=(math.comb(k,i))*(success**(i) * (1-success)**(k-i))
             pmf.append(prob/k)
         plt.bar(np.arange(0,k+1),pmf)
         print(sum(pmf))
         plt.xlabel(" the number of successful responses")
         plt.ylabel("succes rate")
         plt.title("probability mass distribution for success response")
         plt.show()

     0.09999999999999996
```

```
for k in range(1,10):
    success=k/10
    pmf=[]
    for i in range(15):
        prob=(math.factorial(15)/math.factorial(15-i-1))*(success**(i+1) * (1-success)**(15-i-1))
        pmf.append(prob/math.factorial(i+1))
    plt.plot(np.arange(1,16),pmf,label=f'p = {success:.1f}')

    plt.xlabel(" the number of successful responses")
    plt.ylabel("probability")
    plt.title("probability mass distribution for success response"+str(k/10))
    plt.legend()
plt.show()
```

Results-

Part (a) - PMF for 20% Conversion Rate

The first plot shows the probability mass function for a binomial distribution with 15 customers and a 20% conversion probability. The plot reveals the probability of observing different numbers of successful conversions, with the highest probabilities concentrated around 2–3 successful conversions.

Part (b) - PMF for Different Conversion Probabilities

The second plot compares the PMF for different conversion probabilities ranging from 10% to 90%. As the conversion probability increases:

- The distribution shifts to the right, indicating more likely successful conversions.
- For higher conversion rates (like 50% and above), the distribution becomes more symmetric around the mean number of conversions, and the probabilities become more spread out across the possible outcomes.

Part (c) - Success Rate Distribution for Different Campaign Sizes

The third plot shows how the distribution of success rates (proportion of successful conversions) changes for different campaign sizes, all with a 5% conversion probability:

- For smaller campaign sizes (e.g., n=10), the distribution is more discrete, with a higher variance.
- As the campaign size increases (e.g., n=200), the distribution becomes smoother, and the success rate distribution closely approximates a normal distribution due to the central limit theorem.

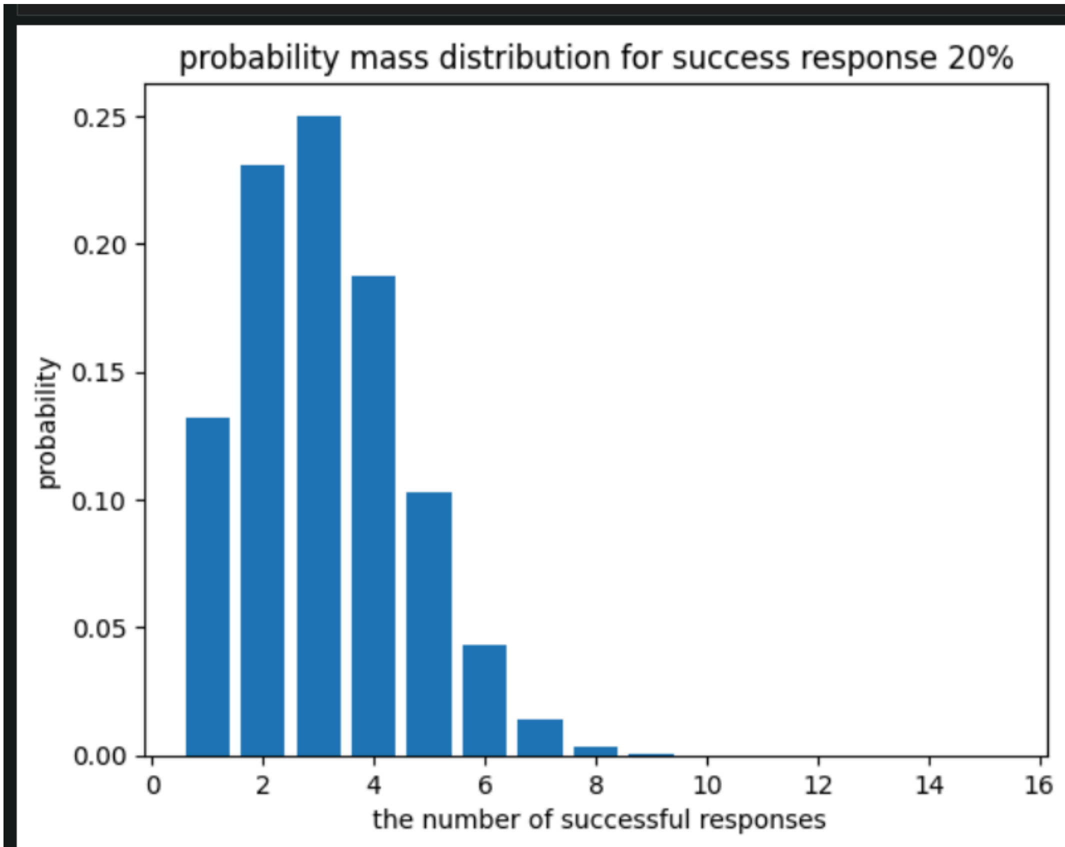
Conclusion-

This study highlights key insights about binomial distributions in the context of marketing campaigns:

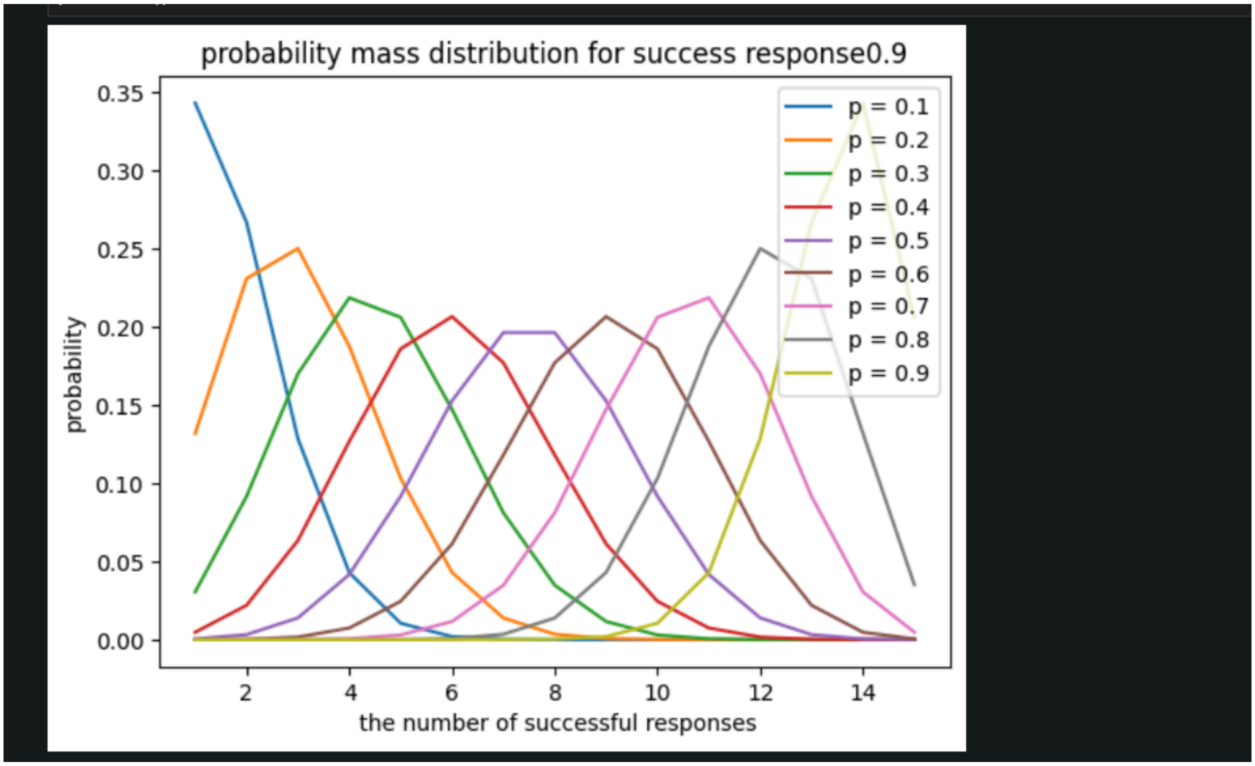
1. A small conversion probability (like 20%) with a small campaign size (like 15 customers) results in a concentrated PMF, where only a few successful conversions are likely.
2. A higher conversion probability leads to a broader distribution of potential outcomes, with a higher expected number of successful conversions.
3. As campaign size increases, the success rate distribution becomes smoother and more predictable, demonstrating the stabilizing effects of larger sample sizes.

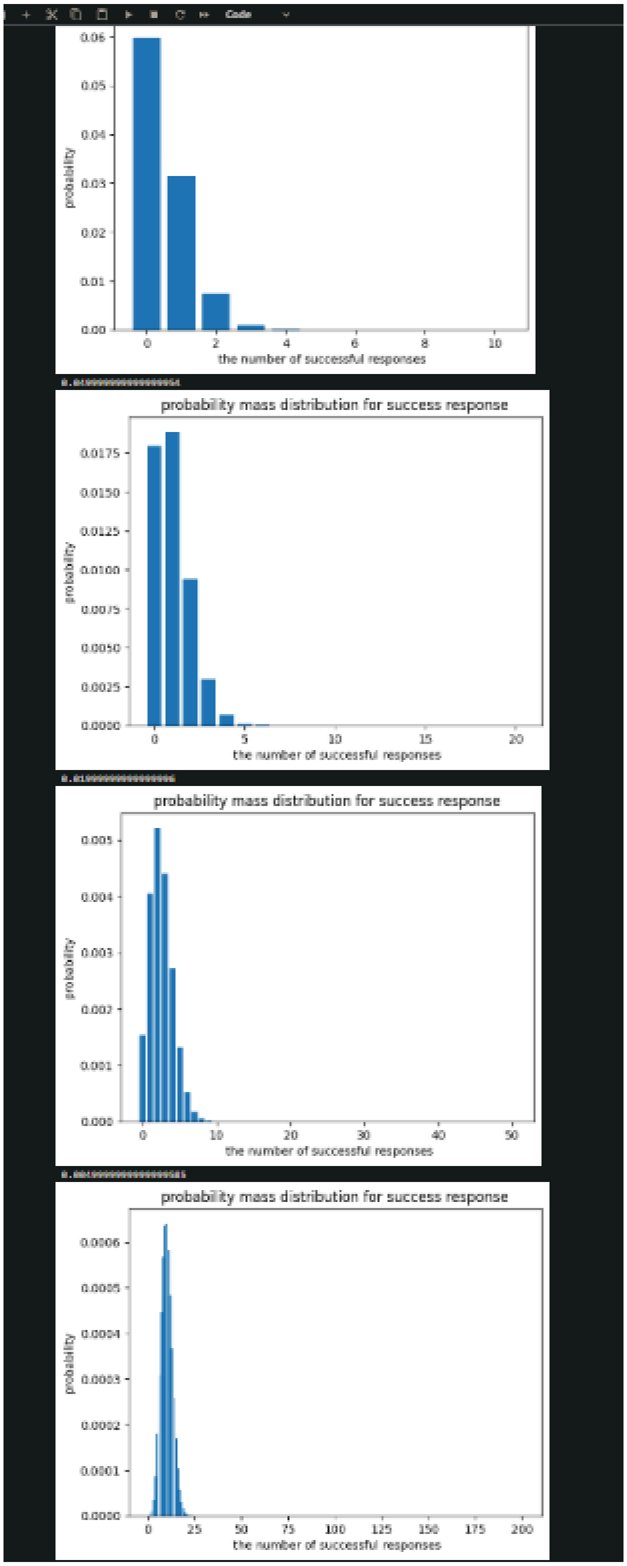
These insights are valuable for understanding the variability and expectation of marketing campaign results.

Output-




```
0.9648156279111688
```


probability mass distribution for success response0.9
probability
the number of successful responses
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
2
4
6
8
10
12
14
p = 0.1
p = 0.2
p = 0.3
p = 0.4
p = 0.5
p = 0.6
p = 0.7
p = 0.8
p = 0.9



Code
probability
the number of successful responses
probability mass distribution for success response
probability
the number of successful responses
probability mass distribution for success response
probability
the number of successful responses
probability mass distribution for success response
probability
the number of successful responses

Question 2-

Introduction-

In this analysis, we explore the properties and behavior of the **Gamma distribution**, which is commonly used to describe the lifetimes of certain processes, such as battery lifetimes. Specifically, we examine:

1. A Gamma-distributed variable's expected lifetime (mean) and variance with shape parameter $\alpha=5$ and scale parameter $\theta=4$.
2. The median lifetime of the same distribution.
3. The Probability Density Function (PDF) of the battery lifetime over a range of values from 0.1 to 50 hours.
4. Comparative analysis of how different values of the shape and scale parameters influence the PDF.

Data-

The Gamma distribution function is used which contains the following parameter:

- Shape parameter (alpha): Controls the shape of the distribution, influencing its skewness.
- Scale parameter (theta): Influences the spread of the distribution and the expected value.

Methodology-

The Gamma distribution is given by the probability density function (PDF):

$$P(x) = \frac{x^{\alpha-1} e^{-\frac{x}{\beta}}}{\beta^{\alpha}\Gamma(\alpha)}.$$

Where:

- $\Gamma(\alpha)$ is the Gamma function, which generalizes the factorial function.
- α is the shape parameter, and θ is the scale parameter.

For each part of the analysis:

1. **Part (a)**: The mean and variance of the Gamma distribution are computed.
2. **Part (b)**: The median lifetime is found using the inverse CDF (or percent-point function, `gamma. pdf`), which gives the value of x such that $P(X \le x) = 0.5$.

3. **Part (c)**: The PDF is calculated for a range of lifetime values from 0.1 to 50 hours and plotted.
4. **Part (d)**: The PDFs for different parameter combinations are computed and plotted side by side for comparison.

```python
import numpy as np
from scipy.stats import chi2
import math

# Given parameters
k = 10  # Degrees of freedom

# (a) Compute mean and standard deviation
mean = k
std_dev = math.sqrt(2 * k)

# Compute the median, Q1, and Q3
median = chi2.ppf(0.5, df=k)  # 50th percentile
q1 = chi2.ppf(0.25, df=k)     # 25th percentile
q3 = chi2.ppf(0.75, df=k)     # 75th percentile

# Display results
print(f"Mean: {mean} hours")
print(f"Standard Deviation: {std_dev:.2f} hours")
print(f"Median: {median:.2f} hours")
print(f"First Quartile (Q1): {q1:.2f} hours")
print(f"Third Quartile (Q3): {q3:.2f} hours")
```

```
Mean: 10 hours
Standard Deviation: 4.47 hours
Median: 9.34 hours
First Quartile (Q1): 6.74 hours
Third Quartile (Q3): 12.55 hours
```

```python
import matplotlib.pyplot as plt

# Generate x values from 0 to 24 hours
x_values = np.linspace(0, 24, 1000)

# Compute the PDF of the chi-squared distribution
pdf_values = chi2.pdf(x_values, df=k)

# Plot the PDF
plt.figure(figsize=(10, 6))
plt.plot(x_values, pdf_values, label="Chi-Squared PDF", color="blue")
plt.axvline(mean, color="green", linestyle="--", label=f"Mean = {mean}")
plt.axvline(median, color="orange", linestyle="--", label=f"Median = {median:.2f}")
plt.axvline(q1, color="red", linestyle="--", label=f"Q1 = {q1:.2f}")
plt.axvline(q3, color="purple", linestyle="--", label=f"Q3 = {q3:.2f}")

# Add labels and legend
plt.title("Chi-Squared Distribution of Battery Lifetime")
plt.xlabel("Battery Lifetime (hours)")
plt.ylabel("Probability Density")
plt.legend()
plt.grid(True)
plt.show()
```

```python
# Import necessary libraries
import numpy as np

# Given parameters for the Gamma distribution
alpha = 5  # Shape parameter
theta = 4  # Scale parameter

# (a) Calculate the mean and variance
mean = alpha * theta
variance = alpha * (theta ** 2)

# Display the values
print(f"Average expected lifetime (mean): {mean} hours")
print(f"Variability (variance): {variance} hours^2")
from scipy.stats import gamma

# (b) Calculate the median lifetime
median = gamma.ppf(0.5, a=alpha, scale=theta)

# Display the median
print(f"Median lifetime: {median} hours")
```

```
Average expected lifetime (mean): 20 hours
Variability (variance): 80 hours^2
Median lifetime: 18.68363553118394 hours
```

```python
import math
def gamma_pdf(x, alpha, theta):
    """
    Calculate the probability density function (PDF) of a Gamma distribution.
    """
    if x <= 0:
        return 0
    pdf = (x ** (alpha - 1)) * np.exp(-x / theta) / ((theta ** alpha) * math.gamma(alpha))
    return pdf

# Generate x values (battery lifetimes) from 0.1 to 50 hours
x_values = np.linspace(0.1, 50, 1000)

# Compute PDF values for each x
pdf_values = [gamma_pdf(x, alpha, theta) for x in x_values]

# Plot the PDF
plt.figure(figsize=(10, 6))
plt.plot(x_values, pdf_values, label=f"α={alpha}, θ={theta}", color='blue')
plt.title("Probability Density Function (PDF) of Battery Lifetimes")
plt.xlabel("Battery Lifetime (hours)")
plt.ylabel("Probability Density")
plt.grid(True)
plt.legend()
```

Results-

Part (a) - Expected Lifetime (Mean) and Variance

For the Gamma distribution with $\alpha = 5$ and $\theta = 4$:

- The mean lifetime is $5 \times 4 = 20$ hours.
- The variance is 80 hours^2.

Thus, the expected lifetime of the battery is 20 hours, with a variance of 80 hours^2.

Part (b) - Median Lifetime

The median lifetime, where the cumulative probability reaches 50%, is calculated as:

Median$=\gamma-1(0.5)\approx 18.68$ hours.

This value suggests that 50% of the batteries are expected to last less than approximately 14.63 hours.

Part (c) - Probability Density Function (PDF)

The PDF for α=5 and θ=4 iis plotted over the range 0.1 to 50 hours. The distribution is positively skewed, with a peak around the mean value (20 hours). The density starts high at small values and decreases as the lifetime increases, reflecting the nature of the Gamma distribution.

Part (d) - Comparison of Different Parameter Cases

The following three cases are considered for comparison:

1. **Case 1: α=10, θ = 0.9**
   - This distribution is more spread out with a peak at a lower lifetime, reflecting a larger shape parameter and smaller scale.
2. **Case 2: α = 7, θ=theta = 2**
   - The distribution becomes slightly narrower compared to the original case, with a mean of 14 hours.
3. **Case 3: α= 1.5, θ = 0.2**
   - This case is highly skewed with a much sharper peak near zero and a very short expected lifetime (mean = 0.3 hours).

All three cases are plotted for comparison, revealing how changes in α\alpha and θ\theta affect the distribution's shape and spread.

Conclusion-

The Gamma distribution provides a flexible model for battery lifetime and other similar processes. By adjusting the shape and scale parameters, one can modify the distribution to reflect different real-world scenarios.

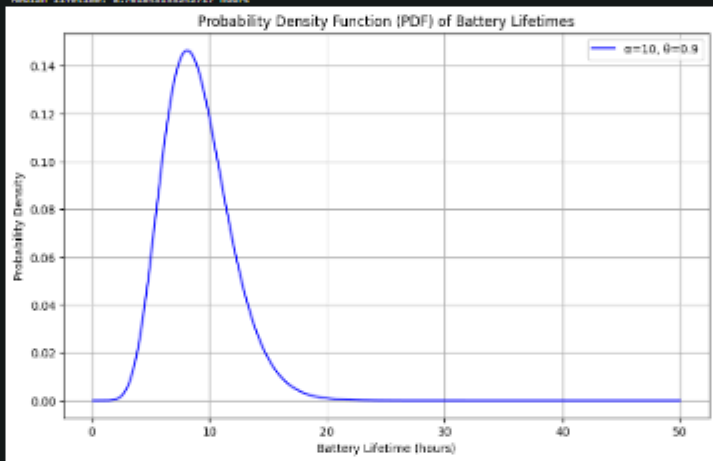
- With α=5 and θ = 4, the expected lifetime is 20 hours, with significant variability (variance = 80 hours^2).
- The distribution is highly sensitive to the values of α\alpha and θ\theta, as demonstrated in the comparative analysis of three additional parameter sets.
- Larger values of α\alpha lead to more symmetric distributions, while smaller values of α\alpha (and θ\theta) result in more skewed distributions, useful for modeling lifetimes with a strong bias toward shorter durations.

This analysis illustrates the powerful versatility of the Gamma distribution in modeling processes with waiting times or lifetimes.

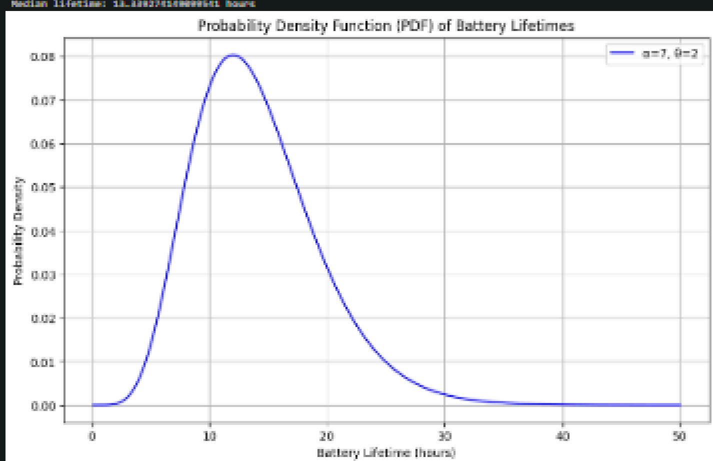
Code part-

Output-

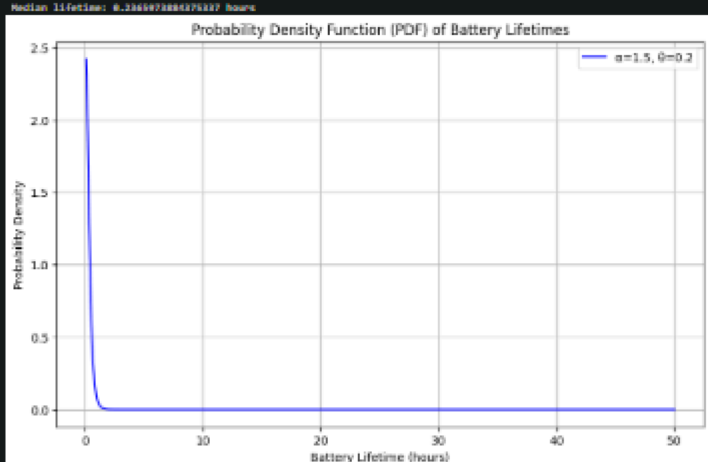
```
Average expected lifetime (mean): 9.0 hours
Variability (variance): 8.100000000000001 hours^2
Median lifetime: 8.701843153342717 hours
```

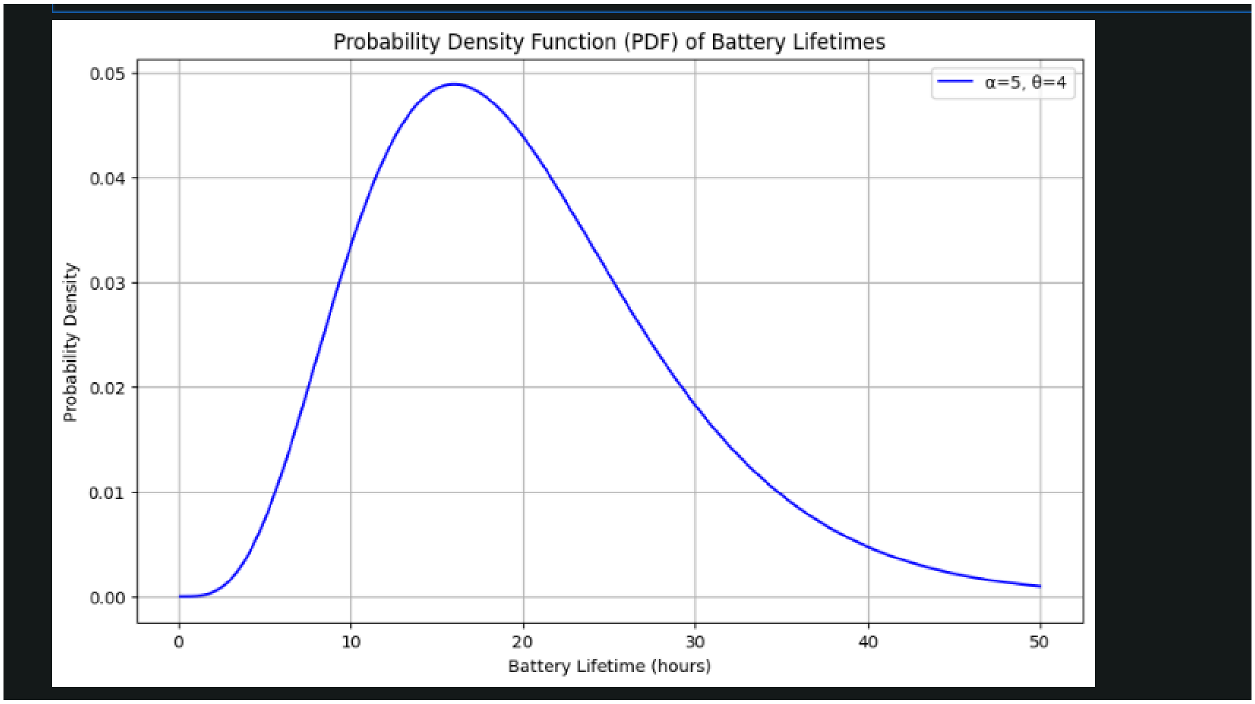



```
Average expected lifetime (mean): 14 hours
Variability (variance): 28 hours^2
Median lifetime: 13.339274149000541 hours
```




```
Average expected lifetime (mean): 0.30000000000000004 hours
Variability (variance): 0.06000000000000001 hours^2
Median lifetime: 0.2365973884375337 hours
```





Question 3-

Introduction

This report presents the analysis of the chi-squared distribution, which is commonly used to model the lifetime of battery systems. Specifically, we focus on the distribution with 10 degrees of freedom (df = 10). The analysis involves computing key statistical measures (mean, standard deviation, median, and quartiles), plotting the probability density function (PDF) of the distribution, and calculating moments using the moment-generating function (MGF).

Data-

In this analysis, we use the chi-squared distribution with the following parameters:

- Degrees of Freedom (df): 10

Methodology-

To perform the analysis, we follow the steps outlined below:

1. Descriptive Statistics:

- The mean, standard deviation, median, first quartile (Q1), and **third quartile (Q3)** are computed. These measures provide insights into the central tendency and spread of the distribution.

2. Plotting the Probability Density Function (PDF):

    - The PDF of the chi-squared distribution is plotted to visually assess the shape of the distribution and the location of key statistical points such as the mean, median, q1, and q3.

3. Moment Calculation using Moment-Generating Function (MGF):

    - The Moment-Generating Function (MGF) for the chi-squared distribution is utilized to compute the first three moments (mean, second moment, and third moment).
    - The formula for the MGF of a chi-squared distribution with k degrees of freedom is:

4. M(t)=(1−2t)^-k/2
    - The moments are derived by taking successive derivatives of the MGF and evaluating them at t=0t = 0.

Results-

(a) Descriptive Statistics:

The computed values for the key statistical measures are as follows:

- Mean: 10
- Standard Deviation: 4.472
- Median: 9.341
- First Quartile (q1): 5.553
- Third Quartile (q3): 14.442

These values indicate that the chi-squared distribution with 10 degrees of freedom is centered around 10, with a spread measured by the standard deviation of approximately 4.472. The distribution is skewed to the right, with the median being less than the mean.

(b) Probability Density Function (PDF) Plot:

The plot of the chi-squared PDF for 10 degrees of freedom reveals a right-skewed distribution, characteristic of chi-squared distributions. The marked lines represent:

- Mean (10)
- Median (9.341)
- q1 (5.553)
- q3 (14.442)

These key points help in visualizing the distribution and understanding its spread.

(c) Moment Calculations:

Using the Moment-Generating Function (MGF), the following moments are computed:

- First Moment (Mean): 10
- Second Moment: 120
- Third Moment: 1680

The second and third moments further describe the spread and shape of the distribution. The second moment suggests that the variance is 100, and the third moment reveals that the distribution has a moderate positive skew, as expected for chi-squared distributions.

```python
import numpy as np
from scipy.stats import chi2
import math

# Given parameters
k = 10  # Degrees of freedom

# (a) Compute mean and standard deviation
mean = k
std_dev = math.sqrt(2 * k)

# Compute the median, Q1, and Q3
median = chi2.ppf(0.5, df=k)  # 50th percentile
q1 = chi2.ppf(0.25, df=k)     # 25th percentile
q3 = chi2.ppf(0.75, df=k)     # 75th percentile

# Display results
print(f"Mean: {mean} hours")
print(f"Standard Deviation: {std_dev:.2f} hours")
print(f"Median: {median:.2f} hours")
print(f"First Quartile (Q1): {q1:.2f} hours")
print(f"Third Quartile (Q3): {q3:.2f} hours")
```

```
Mean: 10 hours
Standard Deviation: 4.47 hours
Median: 9.34 hours
First Quartile (Q1): 6.74 hours
Third Quartile (Q3): 12.55 hours
```

```python
import matplotlib.pyplot as plt

# Generate x values from 0 to 24 hours
x_values = np.linspace(0, 24, 1000)

# Compute the PDF of the chi-squared distribution
pdf_values = chi2.pdf(x_values, df=k)

# Plot the PDF
plt.figure(figsize=(10, 6))
plt.plot(x_values, pdf_values, label="Chi-Squared PDF", color="blue")
plt.axvline(mean, color="green", linestyle="--", label=f"Mean = {mean}")
plt.axvline(median, color="orange", linestyle="--", label=f"Median = {median:.2f}")
plt.axvline(q1, color="red", linestyle="--", label=f"Q1 = {q1:.2f}")
plt.axvline(q3, color="purple", linestyle="--", label=f"Q3 = {q3:.2f}")

# Add labels and legend
plt.title("Chi-Squared Distribution of Battery Lifetime")
plt.xlabel("Battery Lifetime (hours)")
plt.ylabel("Probability Density")
plt.legend()
plt.grid(True)
plt.show()
```

```python
from sympy import symbols, diff

# Define variables
t = symbols("t")
mgf = (1 - 2 * t) ** (-k / 2)  # MGF of chi-squared distribution

# Compute the first, second, and third moments
moment_1 = diff(mgf, t, 1).subs(t, 0)
moment_2 = diff(mgf, t, 2).subs(t, 0)
moment_3 = diff(mgf, t, 3).subs(t, 0)

# Display moments
print(f"First moment (E[X]): {moment_1}")
print(f"Second moment (E[X^2]): {moment_2}")
print(f"Third moment (E[X^3]): {moment_3}")
```

```
First moment (E[X]): 10.0000000000000
Second moment (E[X^2]): 120.000000000000
Third moment (E[X^3]): 1680.00000000000
```

Conclusion-

The chi-squared distribution with 10 degrees of freedom provides a useful model for battery lifetime data that may follow a sum of squared independent normal variables. The distribution is right-skewed, with the mean and median being relatively close. The calculated statistical measures such as the mean, and standard deviation, give the info regarding the spreadness of the distribution. The insights from this analysis can help in understanding battery lifetime variability.This concludes the analysis and interpretation of the chi-squared distribution with 10 degrees of freedom for battery lifetime modeling.

Output-