Moiz Zulfiqar (08229) and Mohit Rai (08229) || Group 1

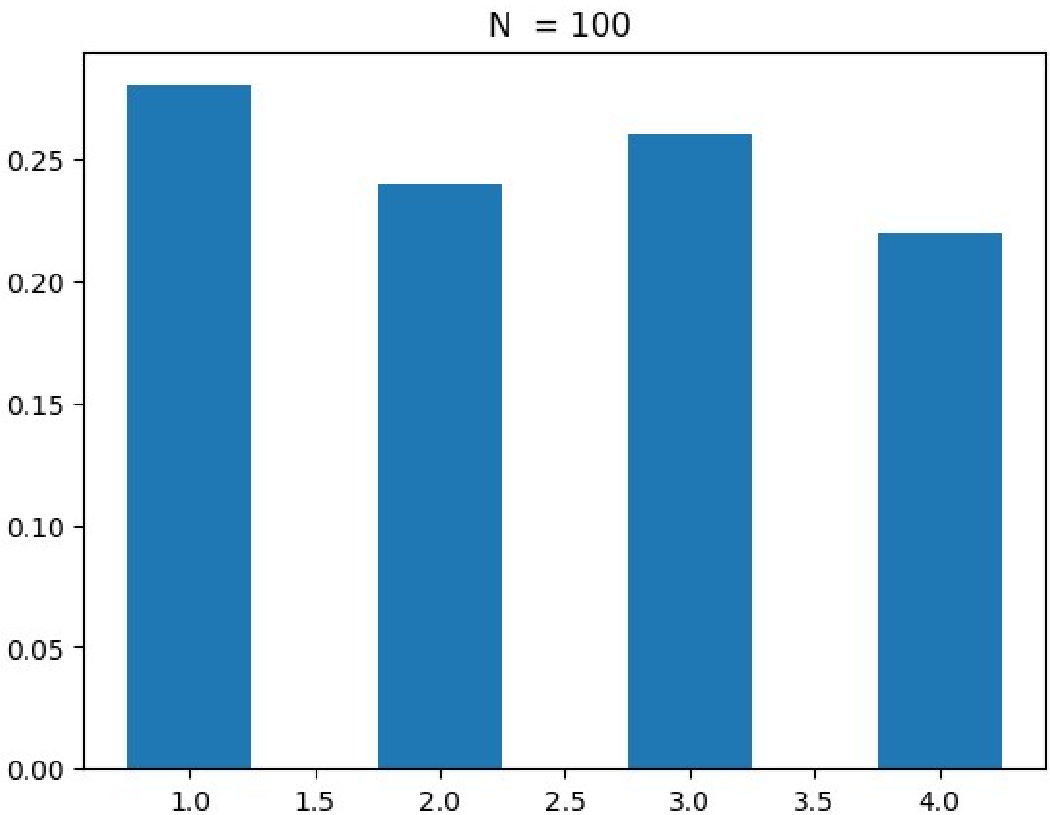
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[10 Marks] Task 1 : Plot Histogram

The following code simulates the rolling of a fair 4-sided die 100 times and plots the histogram of the outputs.

Setup

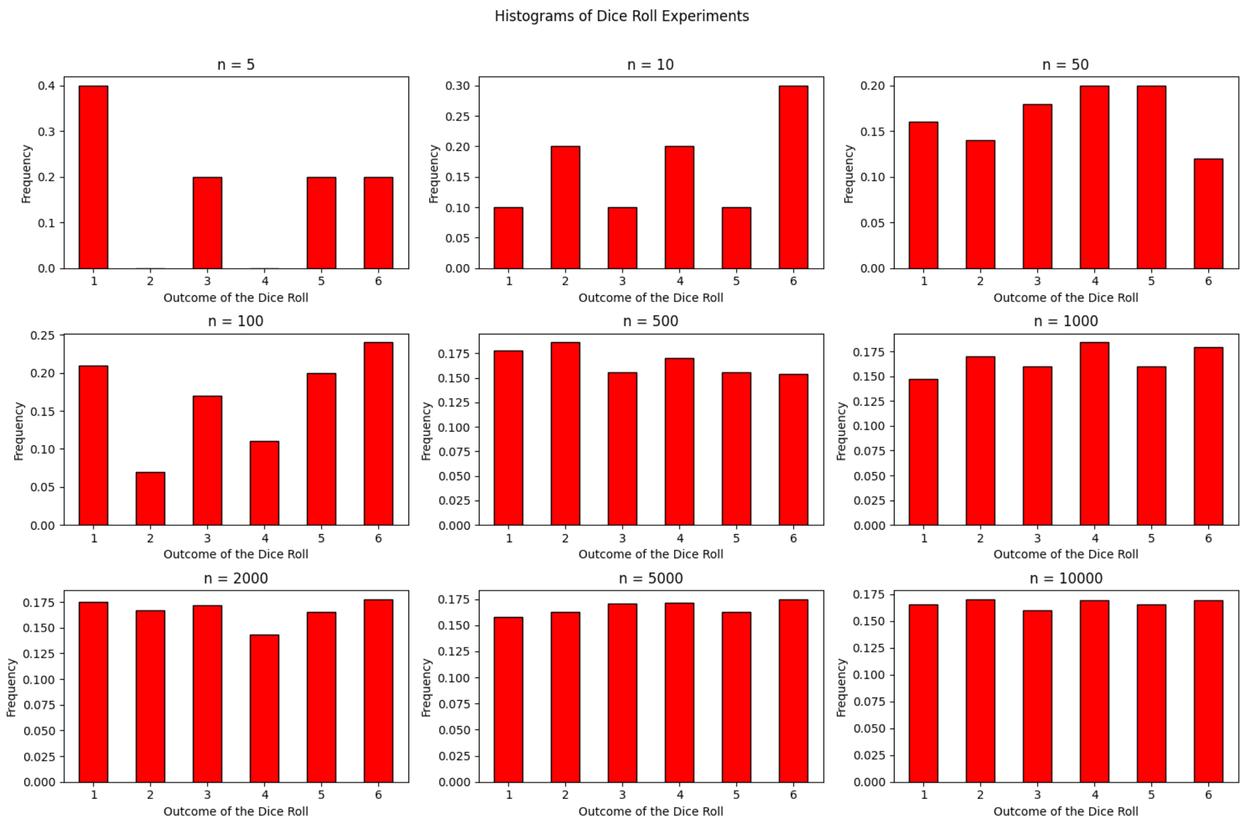
```
# Simulates the roll of a a fair 4-sided die. Size determines the
number of times the die is rolled.
fair  = np.random.randint(low=1,high=5, size=100)
# Plot Histogram.
plt.hist(fair,bins=[1,2,3,4,5],rwidth = 0.5, density=True,
align='left')
plt.title("N  = %i "%100)
plt.show()
```



In this task, you need to simulate the throwing of a fair 6-sided die (Octahedron) N times and plot the corresponding histogram of outputs. You are required to experiment with different values of N and comment how the shape of the histogram changes with increasing values of N.

```
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
# Define the array of number of experiments
num_of_experiments = [5, 10, 50, 100, 500, 1000, 2000, 5000, 10000]

# Create subplots
num_rows = 3
num_cols = 3
fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 10))
fig.suptitle('Histograms of Dice Roll Experiments')

# Iterate over number of experiments and plot histograms
for i, num_exp in enumerate(num_of_experiments):
    row = i // num_cols
    col = i % num_cols

    # Generate random dice rolls
    dice_rolls = np.random.randint(1, 7, size=num_exp)

    # Plot histogram
    axes[row, col].hist(dice_rolls,bins=[1,2,3,4,5,6,7], rwidth=0.5,
```

```
density=True, align='left', edgecolor='black',color='red')
    axes[row, col].set_title("n = {}".format(num_exp))
    axes[row, col].set_xlabel("Outcome of the Dice Roll")
    axes[row, col].set_ylabel("Frequency")

plt.tight_layout()
plt.subplots_adjust(top=0.9)
plt.show()
```



Comments

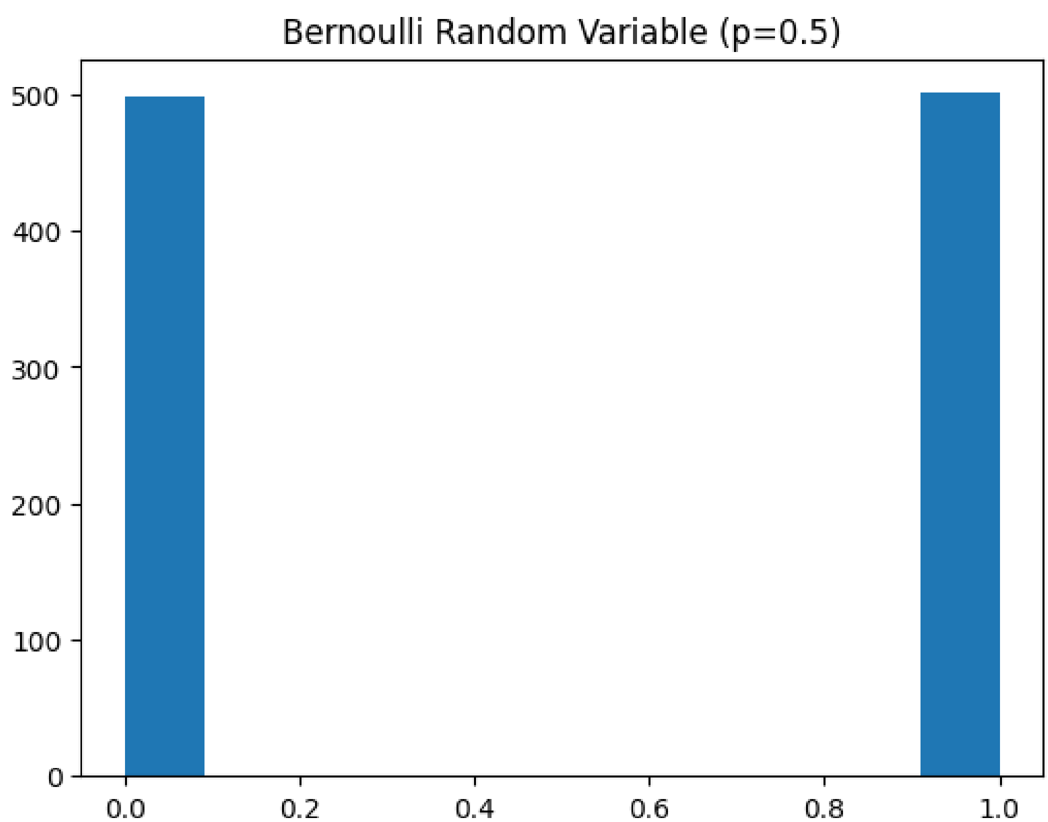
As we experiment with the value of N, which denotes the number of trials and we increase it, we notice reduced variability which shows that with larger sample sizes, the variability in the histogram from one simulation to another will decrease. This is because larger sample sizes provide more stable estimates of the underlying probabilities, resulting in histograms that are more consistent across multiple runs. This basically shows that the height of the separate bins is more evenly spread out, unlike in the case of a small N, where there was a vast difference in the heights of the bins. We see this change because as the number of trials increase the bins are able to more accurately represent the probability of each value due to the excessive amonunt of trails.As the value of N increases, thee is a sort of Uniformity, which appears for with the shape of the Histogram. We can observe that the shape moves towards the a expected theoritical Uniform distribution for a fair six-sided dice, which in its case would be a uniform distribution. As the N increases, the variability of the Histogram decreases, it tends to acheive smootheness. It approaches uniformity, with in larger experiment the observed outcome of each outcome approaches equal probability for 1 to 6, as it is expected for an ideal scenario. Finally, the peaks of the distribution become narrower and histogram becomes stabel, less susceptible to changes.

[20 Marks] Task 2: Common Discrete Random Variables

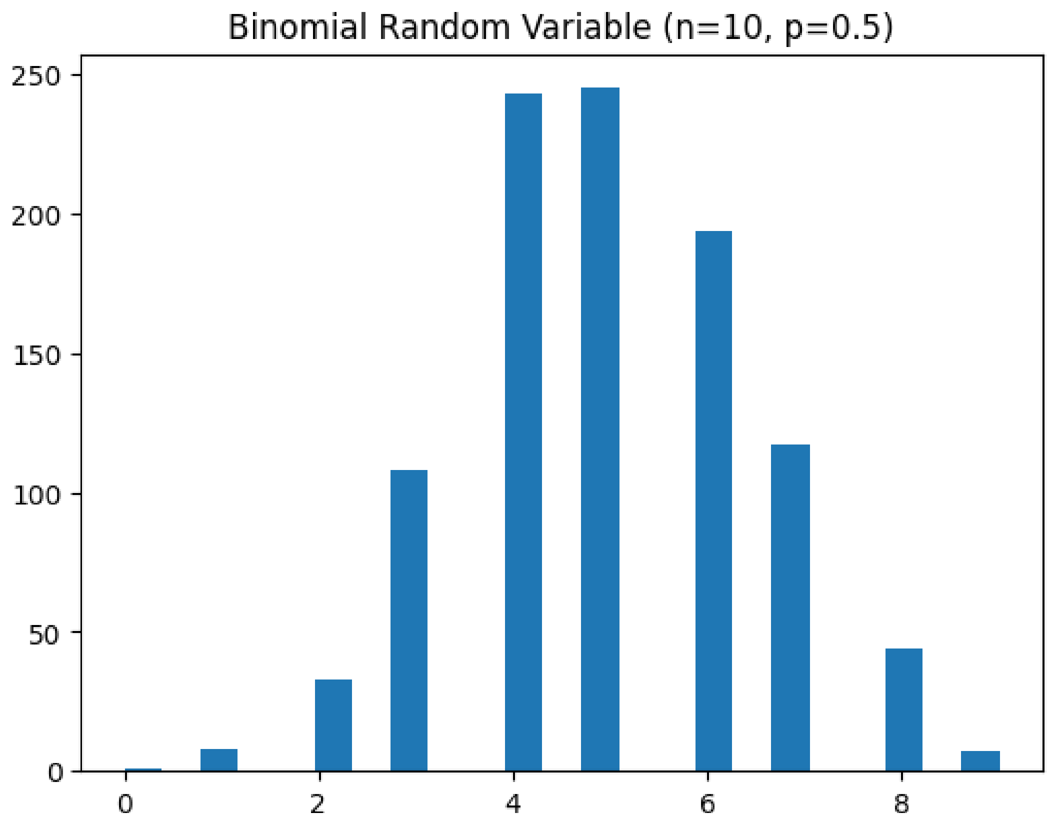
In Task 1, you were introducted to the relationship between histogram and PMF of Random Variables. In Task 2, you will be introduced to common Discrete Random Variables (DRVs) that you have already discussed in class. The aim of Task 2 is to introduce you to the simulation of these DRVs in Python. In this task, you are required to vary parameters for each DRV while keeping others constant and comment on how does this affect the shape of the their histogram.

The following code snippets show you how to simulate the generation of 1000 samples of the following types of random variables: Bernoulli, Binomial, Geometric, Poisson. The code also plots the histogram for the 1000 samples of each type of random variable.

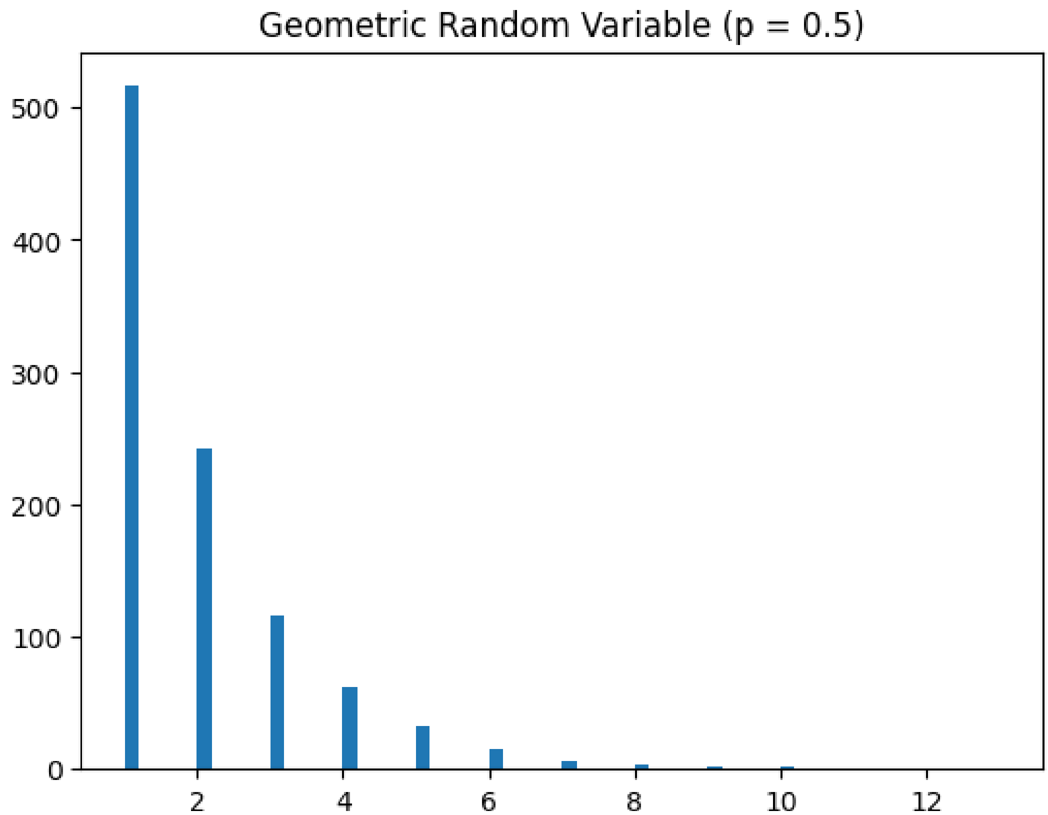
```
# Generate 1000 samples of a Bernoulli Random Variable
p = 0.5 # Probability of Success
n = 1   # Number of states (When n = 1, Binomial Random Variable
becomes a Bernoulli Random Variable)
size  = 1000 # Number of trials
X = np.random.binomial(n,p,size=size)
plt.title("Bernoulli Random Variable (p=0.5)")
plt.hist(X,bins='auto')
plt.show()
```



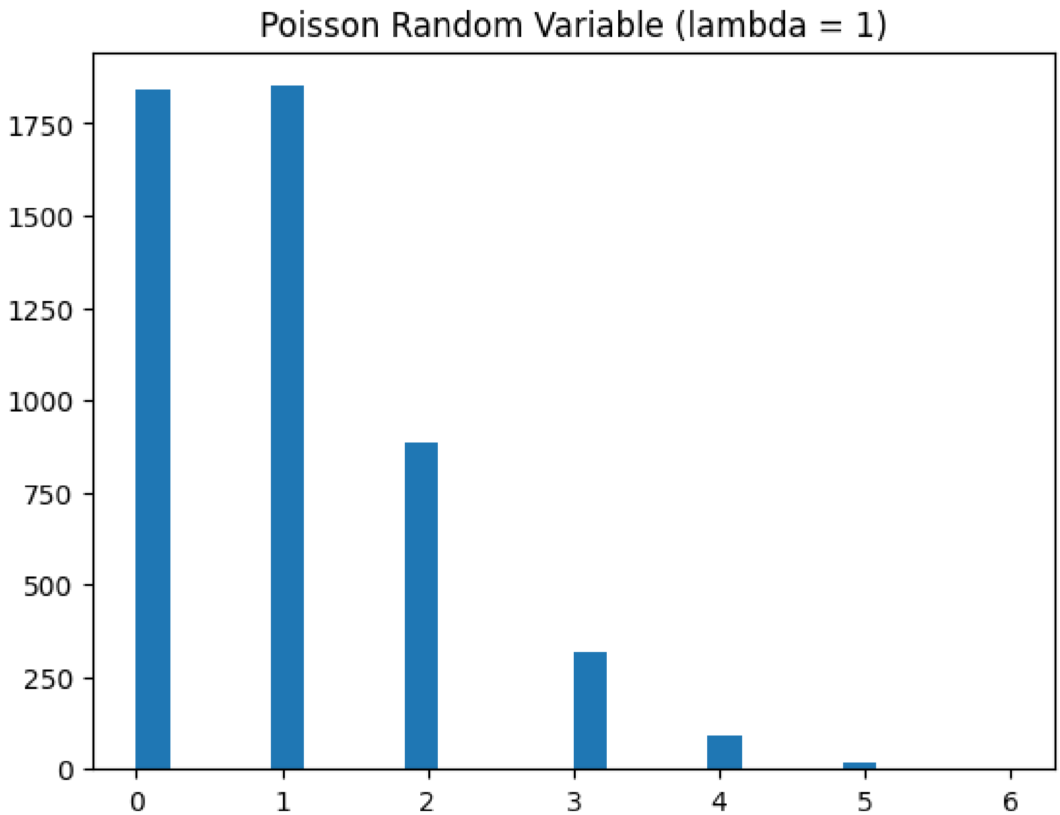
```python
# Generate 1000 samples of a Binomial Random Variable
p = 0.5 # Probability of a single success
n = 10  # Number of states
size  = 1000 # Number of trials
X = np.random.binomial(n,p,size=size)
plt.title("Binomial Random Variable (n=10, p=0.5)")
plt.hist(X,bins="auto");
plt.show()
```




```python
# Generate 1000 samples of a Geometric Random variables
p = 0.5
size = 1000
plt.title("Geometric Random Variable (p = 0.5)")
X = np.random.geometric(p,size=size)
plt.hist(X,bins="auto")
plt.show()
```




```
# Generate 5000 samples of a Poisson Random Variable
lambd = 1
X = np.random.poisson(lambd,size=5000)
plt.title("Poisson Random Variable (lambda = 1)")
plt.hist(X,bins="auto")
plt.show()
```

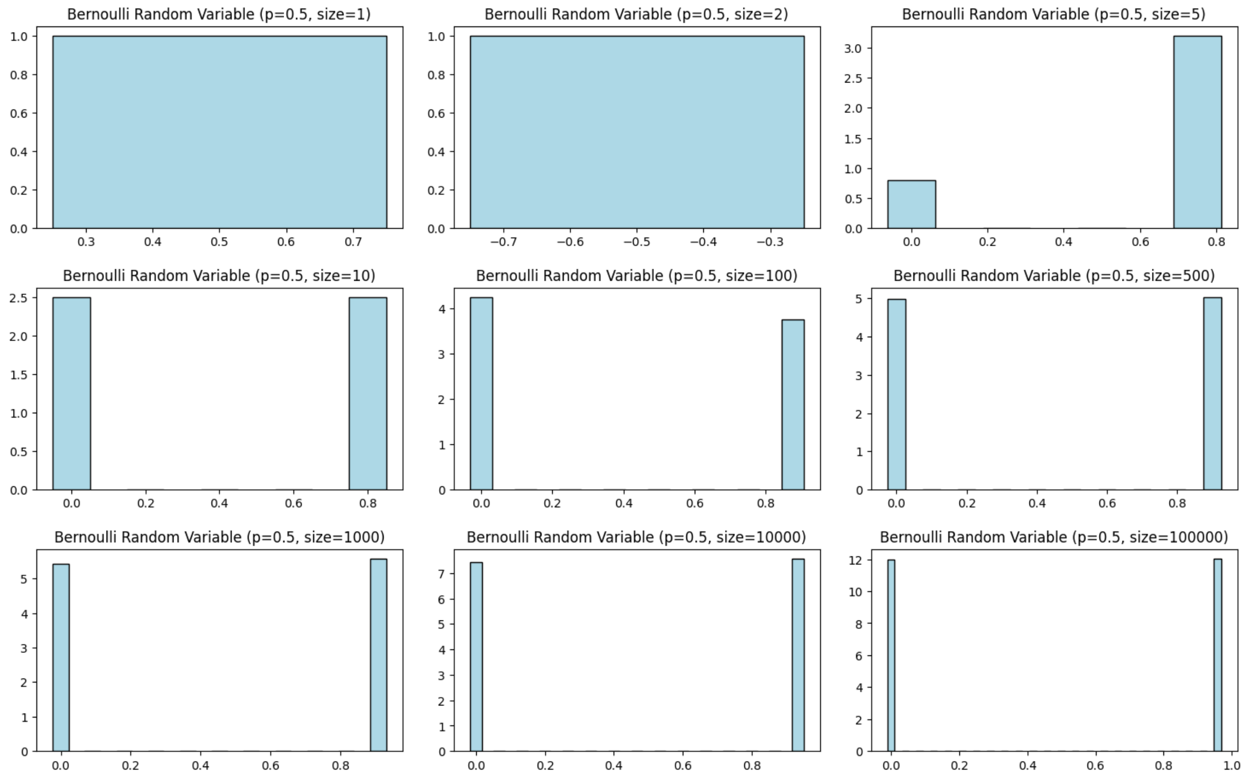


In this task, you are required to vary parameters for each Discrete RV (Bernoulli, Binormail, Geometric, Poission) while keeping others constant and display how does this affect the shape of the their histogram.

Bernoulli Random Variable:

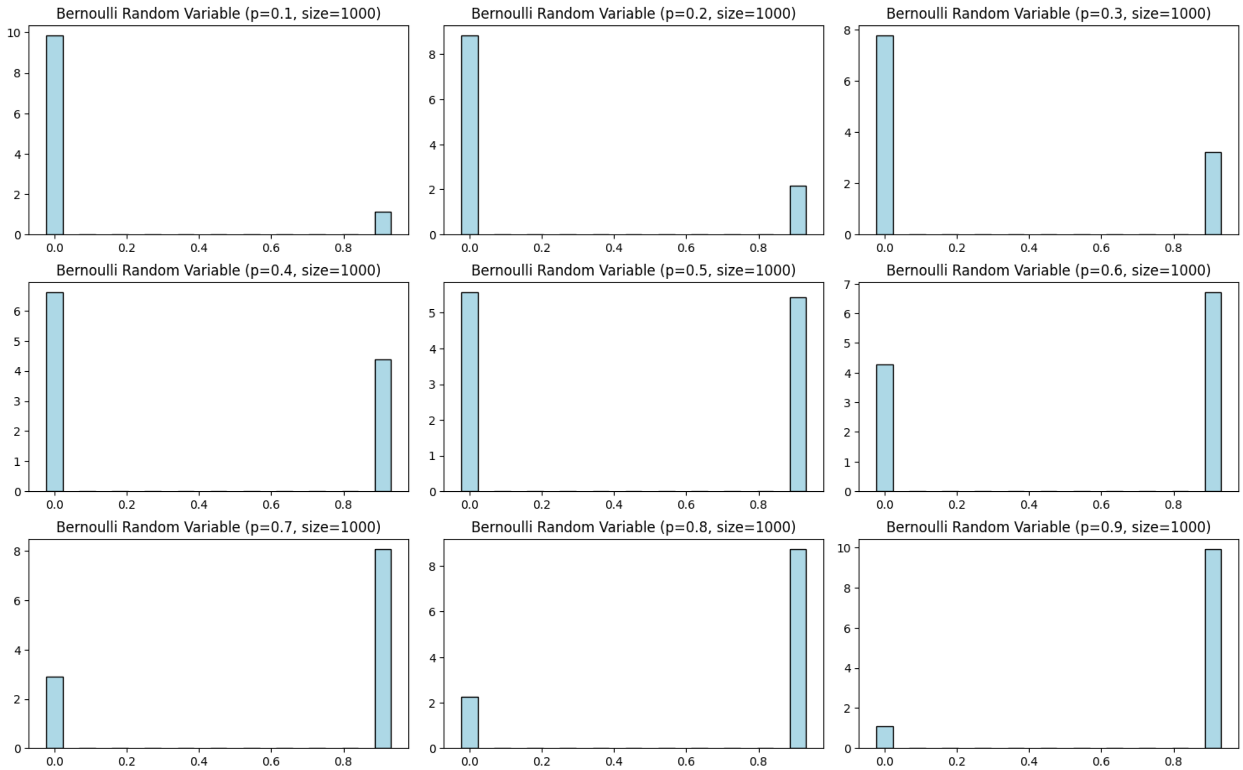
```
# Varying the Bernoulli Random Variable
# Keeping the P constant and varying the Sizes
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
p = 0.5  # Probability of Success
sizes = [1, 2, 5, 10, 100, 500, 1000, 10000, 100000]  # Different
sizes
num_rows = 3
num_cols = 3

fig, axs = plt.subplots(num_rows, num_cols, figsize=(15, 10))
axs = axs.flatten()

for i, size in enumerate(sizes):
    X = np.random.binomial(1, p, size=size)
    axs[i].hist(X, bins='auto',rwidth=0.5, density=True, align='left',
edgecolor='black',color='lightblue')
    axs[i].set_title(f"Bernoulli Random Variable (p={p},
size={size})")
```

```
plt.tight_layout(pad=2.0)  # Adjust the padding as needed
plt.subplots_adjust(top=0.9)
plt.show()
```




```
# Varying the Bernoulli Random Variable
# Keeping the size constant and varying P
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
size = 1000  # Constant sample size
ps = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]  # Different
probabilities
num_rows = 3
num_cols = 3

fig, axs = plt.subplots(num_rows, num_cols, figsize=(15, 10))
axs = axs.flatten()

for i, p in enumerate(ps):
    X = np.random.binomial(1, p, size=size)
    axs[i].hist(X, bins='auto',rwidth=0.5, density=True, align='left',
edgecolor='black',color='lightblue')
    axs[i].set_title(f"Bernoulli Random Variable (p={p},
size={size})")

plt.tight_layout()
```

```
plt.subplots_adjust(top=0.9)
plt.show()
```



Comments:

Since this is a Bernoulli Random Variable there is only the probability of success which is 1 and the probability of failure is 0, hence we see only 2 bins.

When P is changed: which denotes the value of the the probability of success, we can observe that the height of the bin which is at the value 1, will change according its magnitude i.e if it is greater than the probability of failure, then the height of the bin at 1 will be greater than the height of the bin at 0, but if the value of the probability of success is lesser than the value of the probability of failure, then the height of the bin at 0 will be greater.

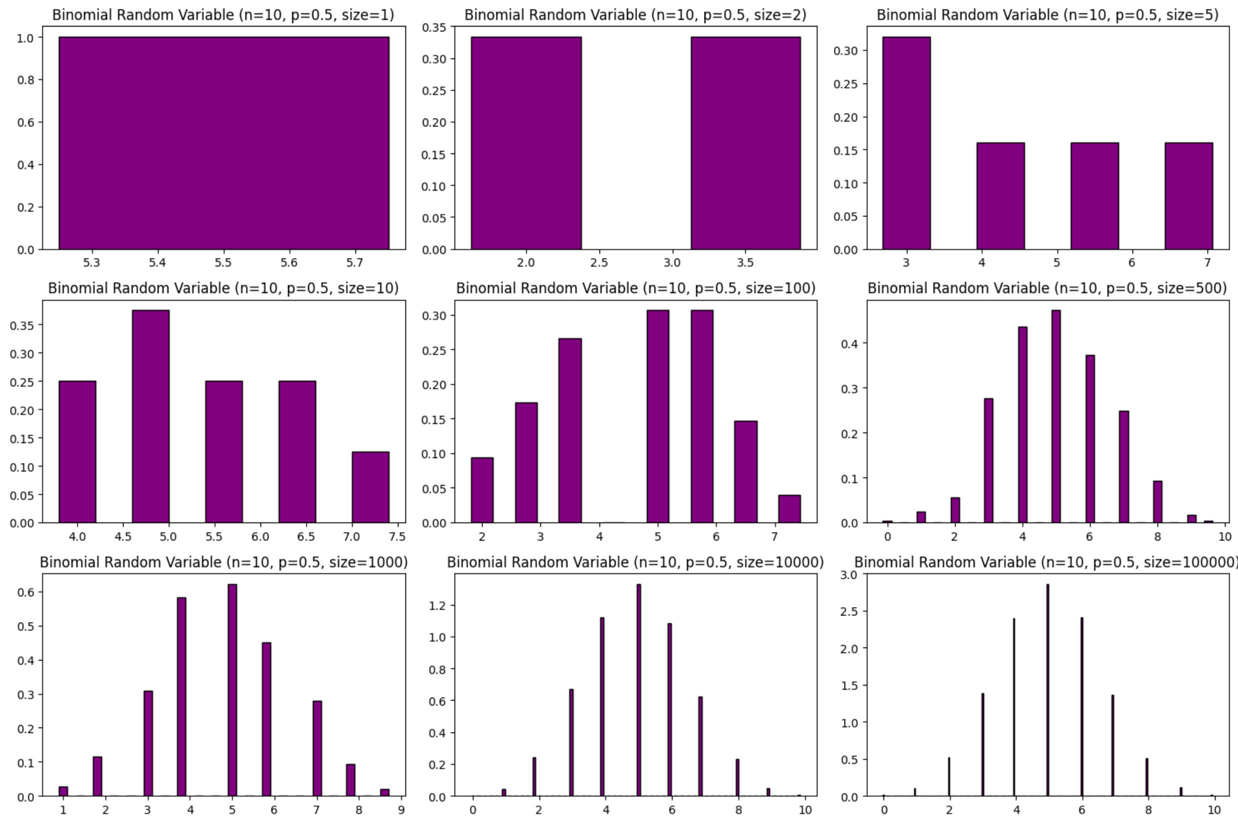
When N is changed: Since this is a Bernoulli Random Variable, which only has 2 states; success and failure, proceeding to change the state would prevent this from remaining a Bernoulli Random Variable because it is supposed to have 2 probabilites only.

When size is changed: When we change size to 1, which denotes the number of trials, we can observe that for a Bernoulli random variable, it means that only one trial is conducted. In this case, the resulting histogram would not provide a meaningful representation of the distribution, as there would be only one outcome (either success or failure) for the single trial. If size is changed to greater than 1 and other consecutive numbers, we can see that with such a small sample size, the resulting histogram does not accurately represent the true distribution of the Bernoulli random variable, but can still give us some insights into the behavior of the variable. However if the size is changed to values greater than 10, we obtain a histogram that provides a more accurate representation of the distribution of the Bernoulli random variable for each probability of success (p). With larger sample sizes, we have more data to estimate the probabilities of success and failure, resulting in histograms that are , stable and more accurate.

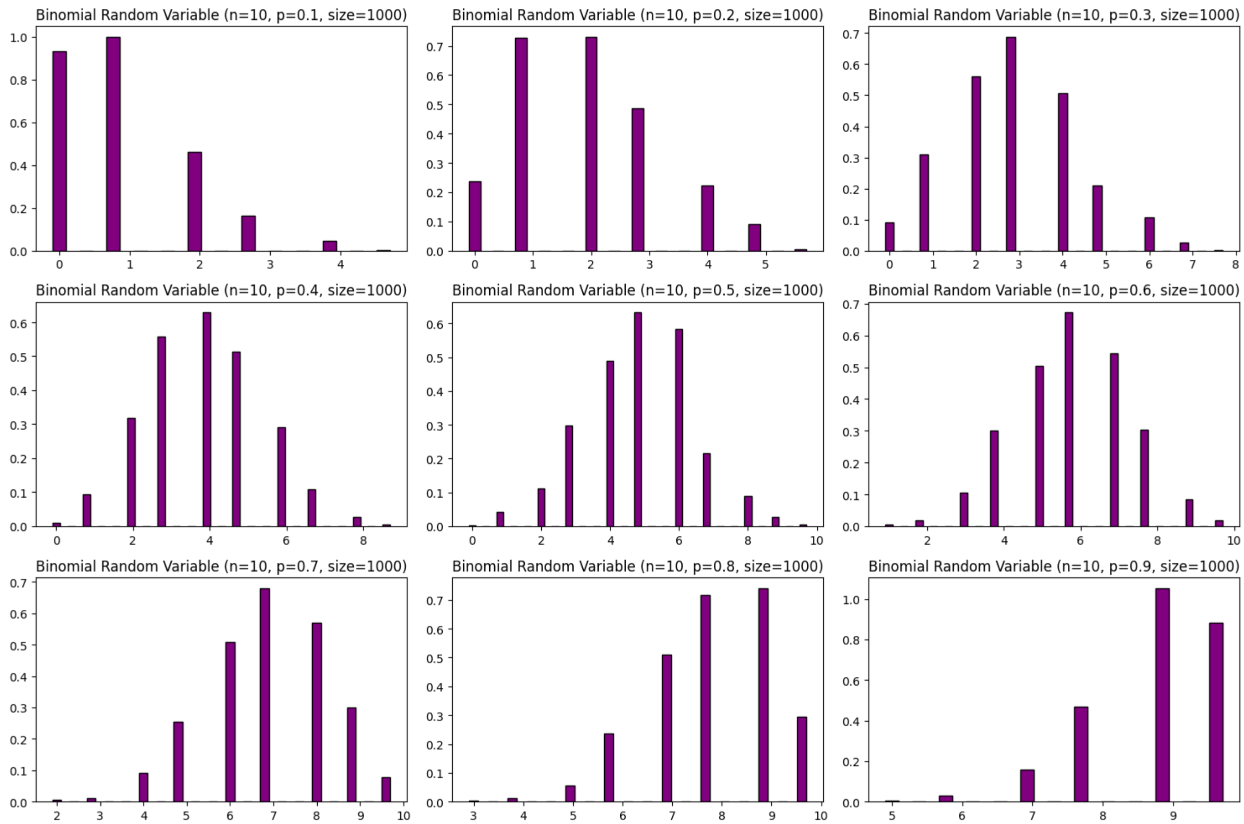
Final Conclusion: The Central Limit Theorem states that when sample size (size) is changed while the probability of success (p) remains constant at 0.5, histograms tend to show a more symmetric shape around the midpoint (0.5). This indicates that the sample size increases reflect the theoretical distribution of a Bernoulli random variable with p = 0.5. While larger sample sizes produce smoother distributions, smaller sizes may exhibit more variability. On the other hand, notice that greater probabilities cause histograms to move towards 1, while lower probabilities cause them to shift towards 0, when p is changed while keeping the sample size constant at 1000. Histograms should be symmetric at the midpoint at p = 0.5, which is the most balanced situation for a Bernoulli distribution and indicates an equal chance of success and failure.

Binomial Random Variable

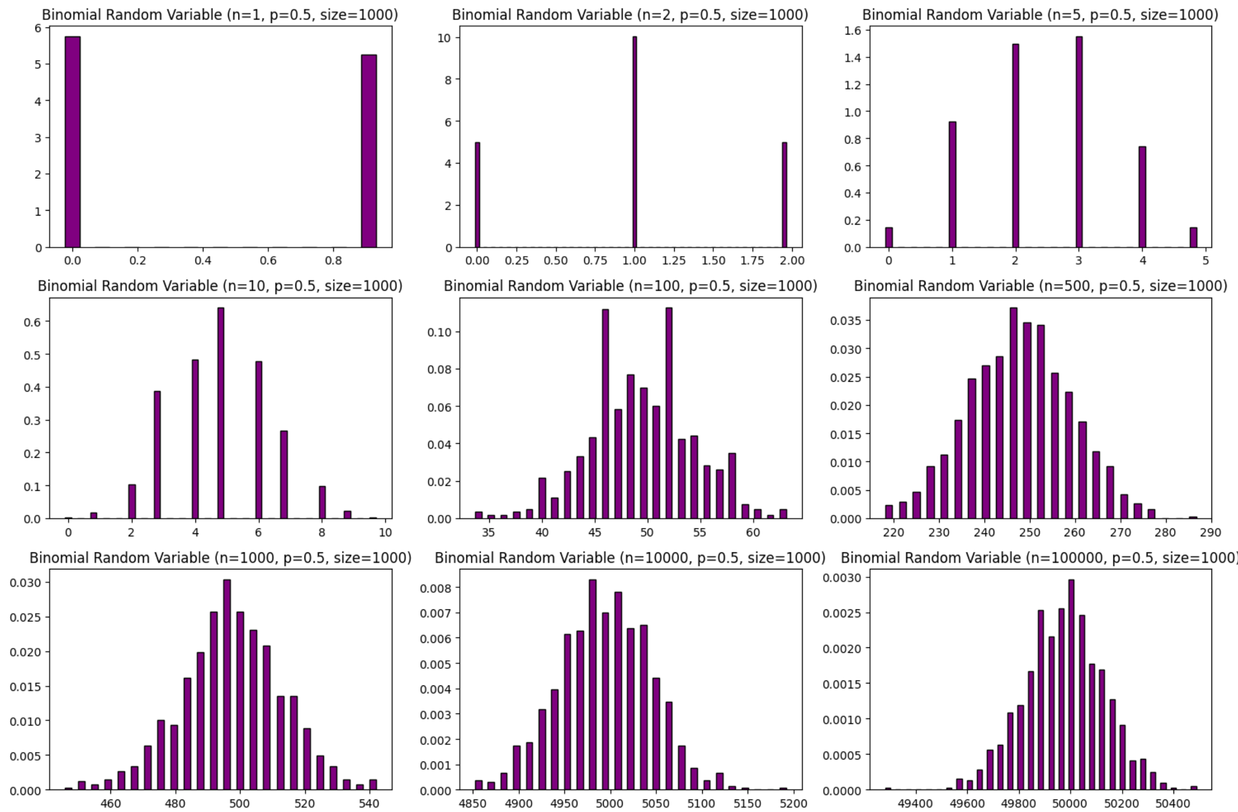
```
# keeping P, n constant and varying sizez
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
p = 0.5  # Probability of a single success
n = 10   # Number of trials
sizes = [1, 2, 5, 10, 100, 500, 1000, 10000, 100000]  # Different
sizes
num_cols = 3
num_rows = 3

fig, axs = plt.subplots(num_rows, num_cols, figsize=(15,10))
axs = axs.flatten()

for i, size in enumerate(sizes):
    X = np.random.binomial(n, p, size=size)
    axs[i].hist(X, bins="auto",rwidth=0.5, density=True, align='left',
edgecolor='black',color='purple')
    axs[i].set_title(f"Binomial Random Variable (n={n}, p={p},
size={size})")

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




```python
# keeping n , sizez constant and varying P
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
n = 10   # Number of trials
p_vary = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
size = 1000
num_cols = 3
num_rows = 3

fig, axs = plt.subplots(num_rows, num_cols, figsize=(15,10))
axs = axs.flatten()

for i, p in enumerate(p_vary):
    X = np.random.binomial(n, p, size=size)
    axs[i].hist(X, bins="auto",rwidth=0.5, density=True, align='left',
edgecolor='black',color='purple')
    axs[i].set_title(f"Binomial Random Variable (n={n}, p={p},
size={size})")

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




```python
#keeping p and size constant and varying n
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
p = 0.5  # Probability of a single success
n_vary = [1, 2, 5, 10, 100, 500, 1000, 10000, 100000]
size = 1000
num_cols = 3
num_rows = 3

fig, axs = plt.subplots(num_rows, num_cols, figsize=(15,10))
axs = axs.flatten()

for i, n in enumerate(n_vary):
    X = np.random.binomial(n, p, size=size)
    axs[i].hist(X, bins="auto",rwidth=0.5, density=True, align='left',
edgecolor='black',color='purple')
    axs[i].set_title(f"Binomial Random Variable (n={n}, p={p},
size={size})")

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



Comments:

Increasing p: would increase the probability of success in each trial. This results in higher counts for larger values in the generated samples. In a histogram, we observe a shift towards the right (higher values) and a higher peak around the expected value (mean). Higher values of p would also mean a greater concentration of probability mass around the expected value of np.

Decreasing p: Decreasing p would decrease the probability of success in each trial. This would likely result in higher counts for smaller values in the generated samples. In a histogram, we might observe a shift towards the left (lower values) and a higher peak around the lower end of the distribution. Lower values of p would also mean a greater concentration of probability mass around the expected value of np.

Increasing n: Increasing n would increase the number of trials in each sample. This results in a distribution that is more spread out and closer to a normal distribution (bell curve) due to the Central Limit Theorem. In a histogram, we observe a narrower and taller distribution with a more pronounced peak around the expected value of np. However this resemblence to uniform distribution is only for certain range of value of n, anything outside this range and the distribution does not look like a uniform distribution.

Decreasing n: would decrease the number of trials in each sample. This would likely result in a distribution that is less spread out and more variable. In a histogram, we might observe a broader and flatter distribution with less pronounced peaks.

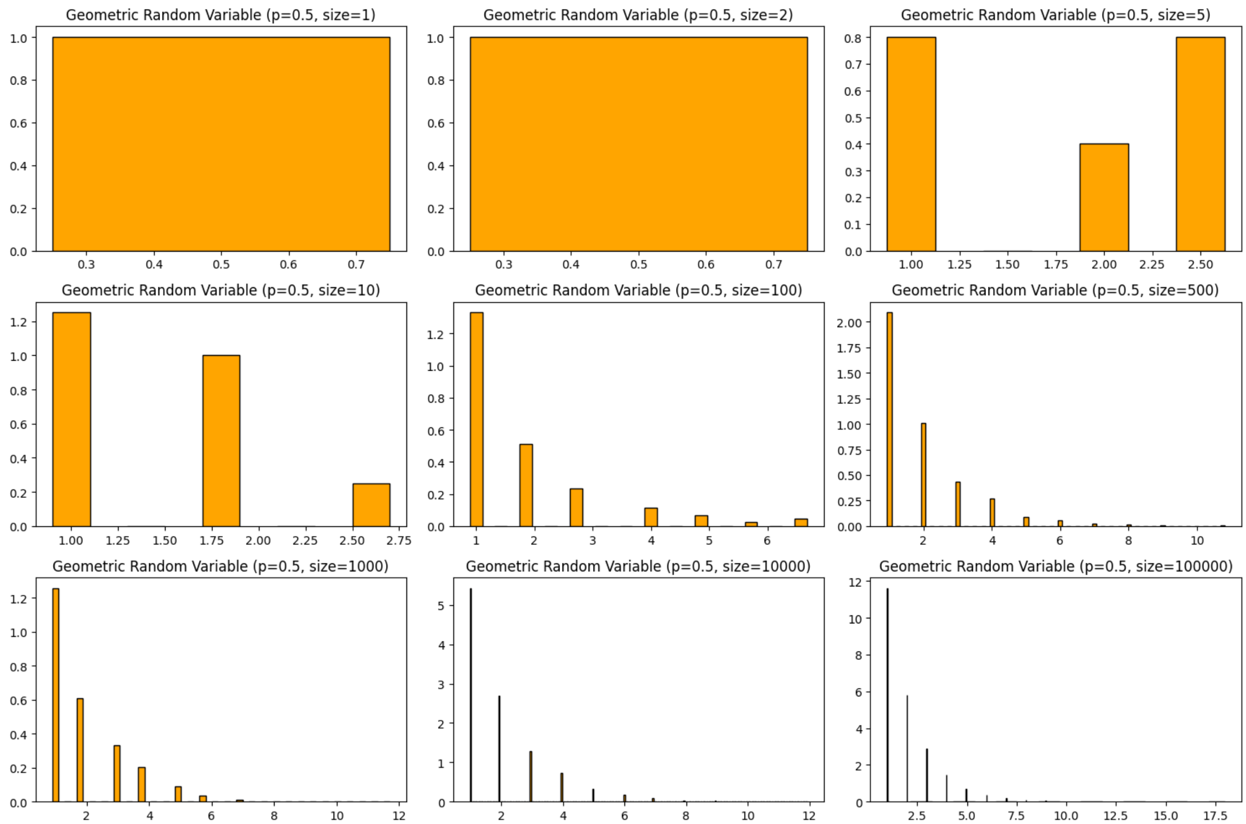
Increasing size: Increasing the number of samples (size) would provide a more accurate estimate of the underlying distribution. The histogram becomes smoother and more stable as the sample size increases. With a larger sample size, the histogram would provide a more accurate representation of the true distribution.

Decreasing size: Decreasing the number of samples (size) would result in a less accurate estimate of the underlying distribution. The histogram would have more variations and as a result be less reliable. With a smaller sample size, there would be more uncertainty in the representation of the true distribution.

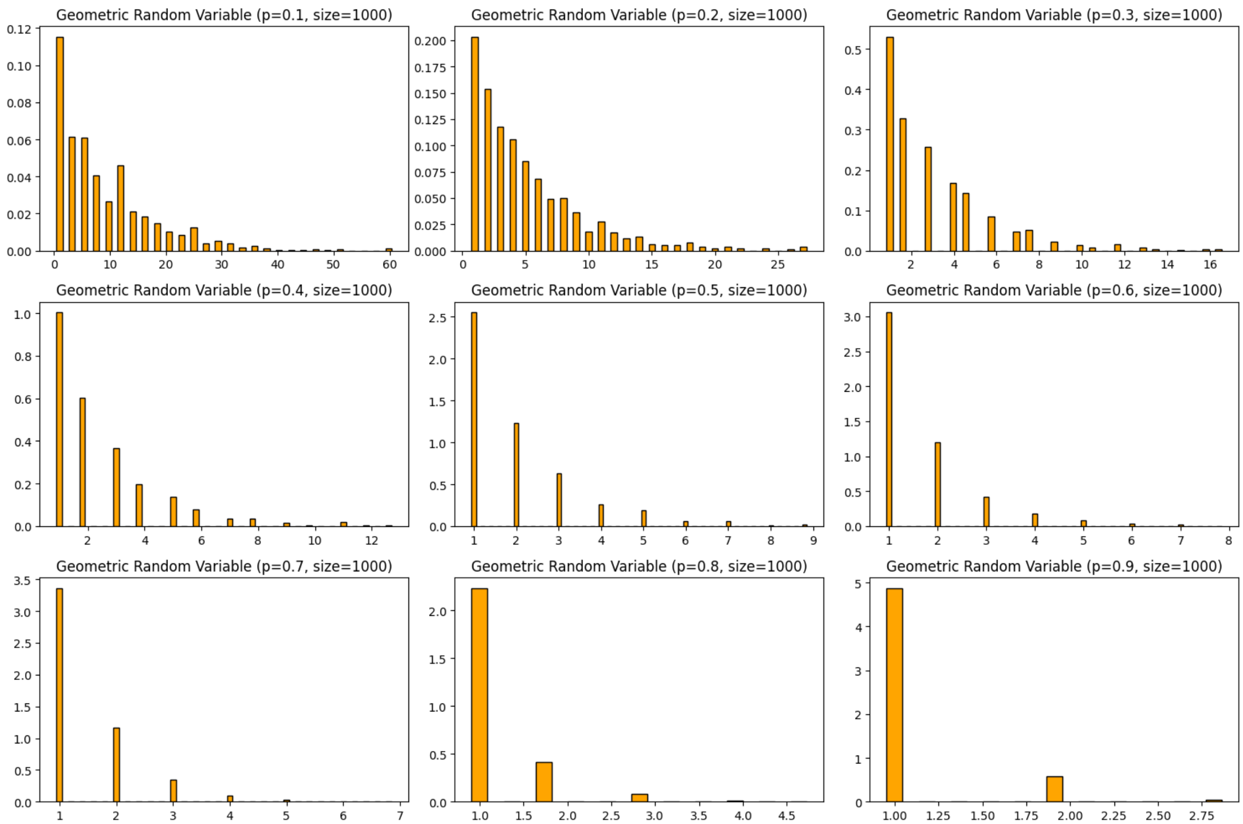
Final Conclusion: Numerous observations can be made by adjusting the binomial distribution's parameters. The Central Limit Theorem states that when the sample size (size) is changed while the number of trials (n) and the probability of success (p) remain constant, the histograms tend to become more symmetrical around the mean as the size increases. Larger samples produce smoother distributions. Histograms will change towards higher values as p increases and towards lower values as p lowers when p is varied while keeping size and n constant. This indicates how the likelihood of success is changing. Lastly, changing n while maintaining p and size results in histograms that are taller and narrower, with the peak moving towards the mean (n * p) as n increases. This suggests that the distribution becomes more precise and less variable as trial counts rise. On the other hand, larger histograms with flatter peaks, indicating greater variability, may result from smaller values of n. P affects the skewness of the graph, size affects the symmetric property of the hostogram and n affects the variability and precision of the histogram.

Geometric Random Variable

```
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
p = 0.5  # Probability of success
sizes = [1, 2, 5, 10, 100, 500, 1000, 10000, 100000]  # Different
sizes

num_cols = 3
num_rows = 3

fig, axs = plt.subplots(num_rows, num_cols, figsize=(15, 10))
axs = axs.flatten()

for i, size in enumerate(sizes):
    X = np.random.geometric(p, size=size)
    axs[i].hist(X, bins="auto",rwidth=0.5, density=True, align='left',
edgecolor='black',color='orange')
    axs[i].set_title(f"Geometric Random Variable (p={p},
size={size})")

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




```python
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
ps = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
size = 1000  # Different sizes

num_cols = 3
num_rows = 3

fig, axs = plt.subplots(num_rows, num_cols, figsize=(15, 10))
axs = axs.flatten()

for i, p in enumerate(ps):
    X = np.random.geometric(p, size=size)
    axs[i].hist(X, bins="auto",rwidth=0.5, density=True, align='left', 
edgecolor='black',color='orange')
    axs[i].set_title(f"Geometric Random Variable (p={p}, 
size={size})")

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



Comments:

Increasing p: would increase the probability of success in each trial of the geometric random variable. This means that on average, fewer trials would be required until the first success occurs. As a result, the histogram shifts towards the left, with higher counts for smaller values (fewer trials required until success). The shape of the distribution would have a higher concentration of values towards the lower end.

Decreasing p: would decrease the probability of success in each trial, leading to a lower likelihood of success in each trial. Consequently, more trials would be required on average until the first success occurs, resulting in a histogram that shifts towards the right. The distribution becomes more spread out, with a higher concentration of values towards the higher end.

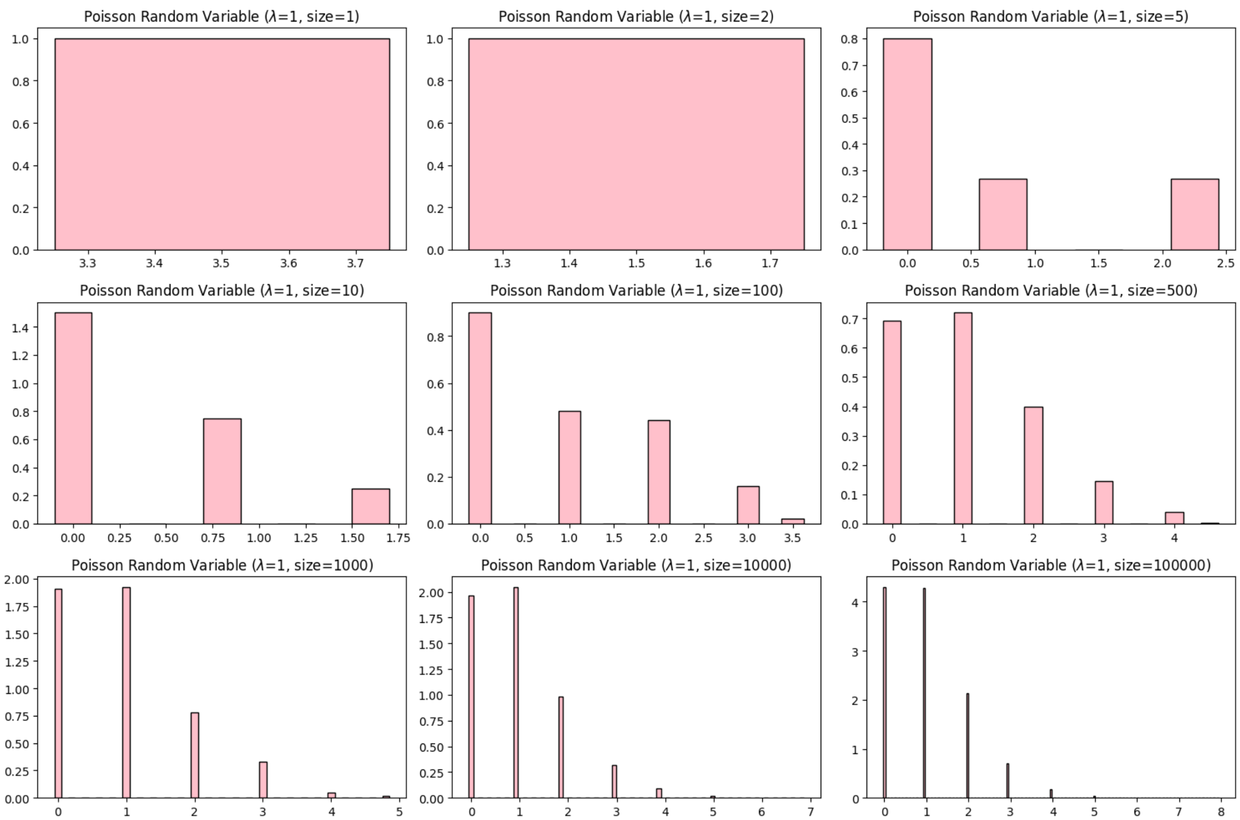
Increasing the number of samples (size): would provide more data resulting in a histogram that is more representative of the underlying distribution. With a larger sample size, the histogram becomes smoother and more stable, with less variability between different runs of the simulation. The shape of the histogram may remain consistent, but the accuracy and reliability of the representation would improve with a larger sample size.

Decreasing size: decreasing the number of samples (size) would result in a smaller dataset for analysis and visualization. This leads to a histogram that is less representative of the underlying distribution. With a smaller sample size, the histogram shows more variability between different runs of the simulation, and the representation of the distribution may be less reliable.

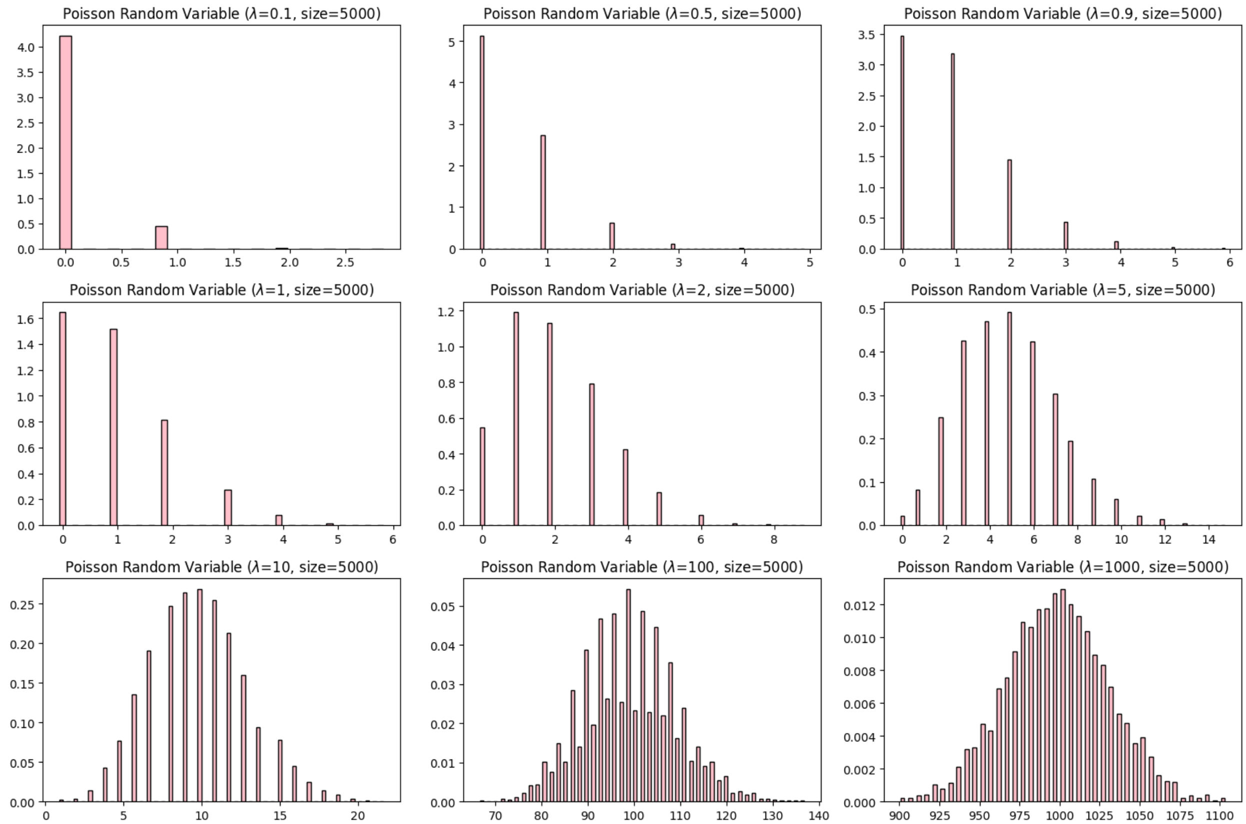
Final Conclusion: You should anticipate to see variations in the distribution's shape while experimenting with the Geometric random variable by changing the size and p (at p=0.5). A decrease in the average number of trials needed to succeed is reflected in shorter tails and a higher concentration of data points near the beginning when p is increased. On the other hand, changes in the probability of success (p) have an impact on the distribution's peak and tail lengths when size is held constant (at size=1000) and p is varied. Lower p values, which indicate a higher average number of attempts until success, move the peak away from 1 and prolong the tails, whereas higher p values shift the peak towards 1 and shorten the tails. These observations show how changes in p and size affect the properties of the distribution and the probability of success in a sequence of independent Bernoulli trials.

Poisson Random Variable

```
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
lambd = 1  # Lambda parameter
sizes = [1, 2, 5, 10, 100, 500, 1000, 10000, 100000]  # Different sizes

num_plots = len(sizes)
num_cols = 3
num_rows = 3

fig, axs = plt.subplots(num_rows, num_cols, figsize=(15, 10))
axs = axs.flatten()

for i, size in enumerate(sizes):
    X = np.random.poisson(lambd, size=size)
    axs[i].hist(X, bins="auto" ,rwidth=0.5, density=True, align='left', edgecolor='black',color='pink')
    axs[i].set_title(f"Poisson Random Variable ($\lambda$={lambd}, size={size})")

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




```
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
lambdas = [0.1, 0.5, 0.9, 1, 2, 5, 10, 100, 1000]  # Different lambda
values
size = 5000


num_cols = 3
num_rows = 3

fig, axs = plt.subplots(num_rows, num_cols, figsize=(15, 10))
axs = axs.flatten()


for i, lambd in enumerate(lambdas):
    X = np.random.poisson(lambd, size=size)
    axs[i].hist(X, bins="auto" ,rwidth=0.5, density=True,
align='left', edgecolor='black',color='pink')
    axs[i].set_title(f"Poisson Random Variable ($\lambda$={lambd},
size={size})")

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



Comments:

Increasing λ: would increase the average rate at which events occur in a fixed interval of time or space. This means that, on average, more events would occur within the same interval. As a result, the histogram of the generated samples shifts towards the right, with higher counts for larger values (more events occurring). The distribution now has a higher concentration of values towards the higher end.

Decreasing λ: would decrease the average rate at which events occur. This means that, on average, fewer events would occur within the same interval. Consequently, the histogram of the generated samples shifts towards the left, with higher counts for smaller values (fewer events occurring). The distribution becomes more spread out, with a higher concentration of values towards the lower end.

Increasing size: Increasing the number of samples (size) would result in a histogram that is more representative of the distribution. With a larger sample size, the histogram becomes smoother and more stable, with less variability between different runs of the simulation. The shape of the histogram may remain consistent, but the accuracy and reliability of the representation would improve with a larger sample size.

Decreasing size: decreasing the number of samples (size) would result in a smaller dataset for analysis and visualization. This leads to a histogram that is less representative of the underlying distribution. With a smaller sample size, the histogram shows more variability between different runs of the simulation, and the representation of the distribution becomes less reliable. The shape of the histogram may still reflect the characteristics of the Poisson distribution, but there may be more uncertainty in the estimates of probabilities.

Final Conclusion: Changes in the distribution's shape and spread can be seen while experimenting with the Poisson random variable by changing the size and lambda (at lambda=1). A higher average rate of occurrence is shown in histograms with larger peaks and a greater concentration of data points around the mean that arise from increasing lambda. On the other hand, histograms with fewer peaks and a wider distribution of data points result from reducing lambda. Histograms with less variability and a greater accuracy in representing the underlying distribution are typically produced by larger sample numbers. Higher lambda values produce histograms with higher peaks and shorter tails, and vice versa. Keeping size constant (at size=5000) while adjusting lambda illustrates how changes in the average rate of occurrence affect the central tendency and variability of the distribution. With Higher values of lambda the distribution does towards Normal Distribution. Changing the size with keeping the lambda constant keeps varies the tails and lengths of probabilites at different values.

[20 Marks] Task 3 : Mathematical Model for the Number of Maiden Overs in an ODI Cricket Match

For Task 3, you have been provided a CSV file (ODIData.csv) that contains some information about all the One Day International (ODI) cricket matches played in the 21st century. An ODI cricket match consists of 100 overs (divided into two innings of 50 overs each). A maiden over is an over in which no run is scored. In this task, you need to develop a mathematical model for number of maiden overs in an ODI cricket match using the probability terminology discussed in class.

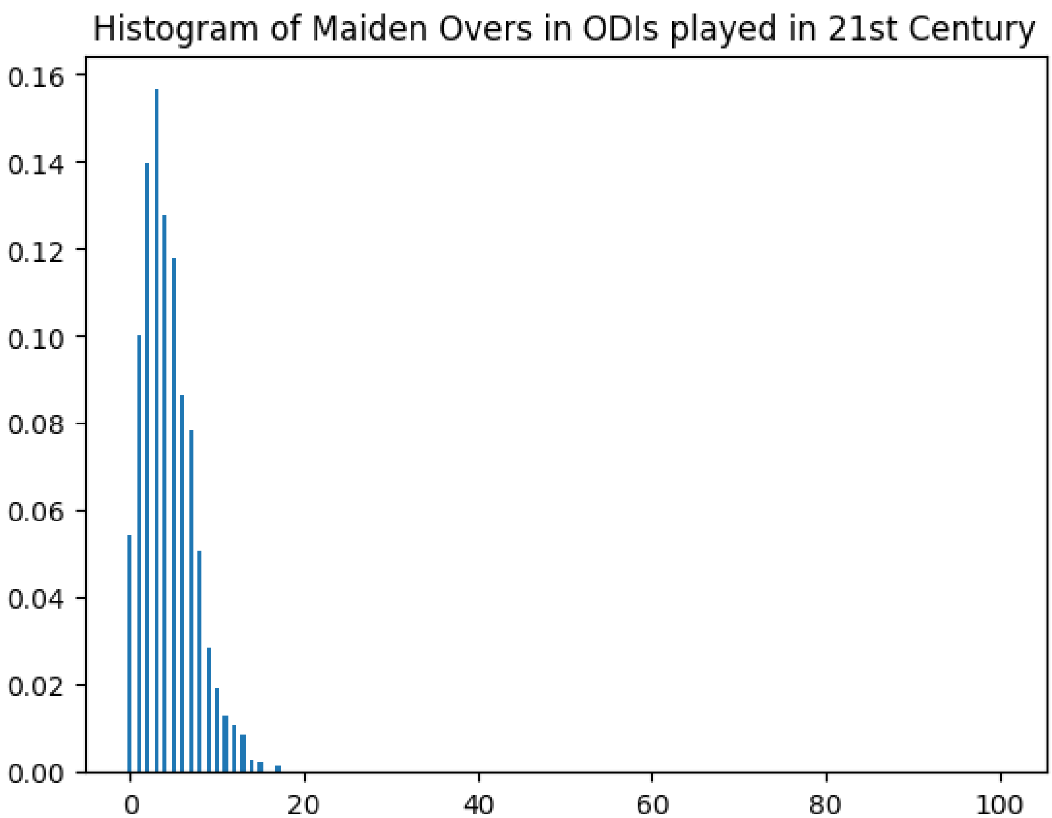
Let's model the number of maiden overs in an ODI cricket match next week as a random variable X. As part of this modeling exercise, what will you choose as the PMF of X? You can choose from various possible PMFs of commmon DRVs in the last section.

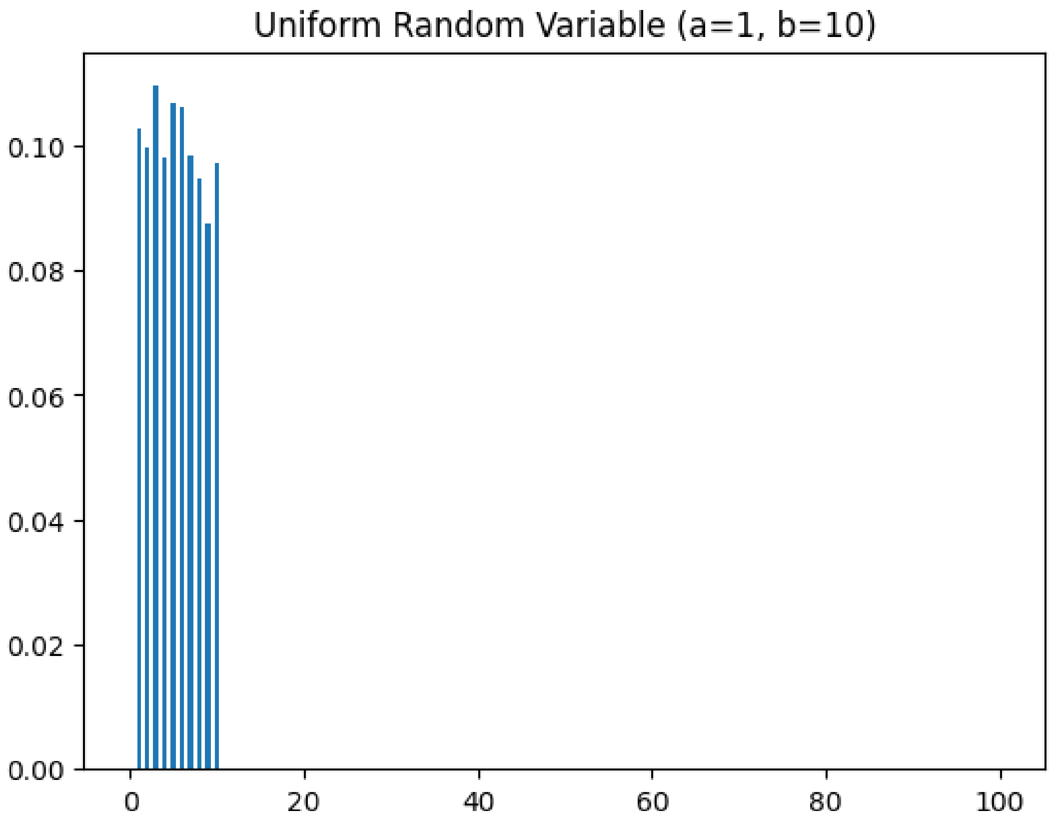
However, you have to justify your answer based on the historical data available in ODIData.csv file. In particular, justify your answer by calculating the "error" between the histogram of maiden overs in the past matches with the histogram of the data generated by your proposed PMF of X. The "error" is defined as the sum of the absolute difference between the corresponding values in the histogram.

For example, the code snippet given below does the following: 1) plots the histogram of maiden overs in ODIs based on the given data 2) plots the histogram of the data generated by a uniform RV with parameter a=1 and b=10. 3) calculates the "error" between the historical data and the proposed PMF

```
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
odi_data_df = pd.read_csv('ODIData.csv')
number_of_maiden_overs = np.array((odi_data_df['Mdns']))
over_bins = np.arange(102)

(bin_values_historical_data, bins, patches) =
plt.hist(number_of_maiden_overs, bins=over_bins, rwidth = 0.5,
density=True, align='left')
plt.title('Histogram of Maiden Overs in ODIs played in 21st Century')
plt.show()
```

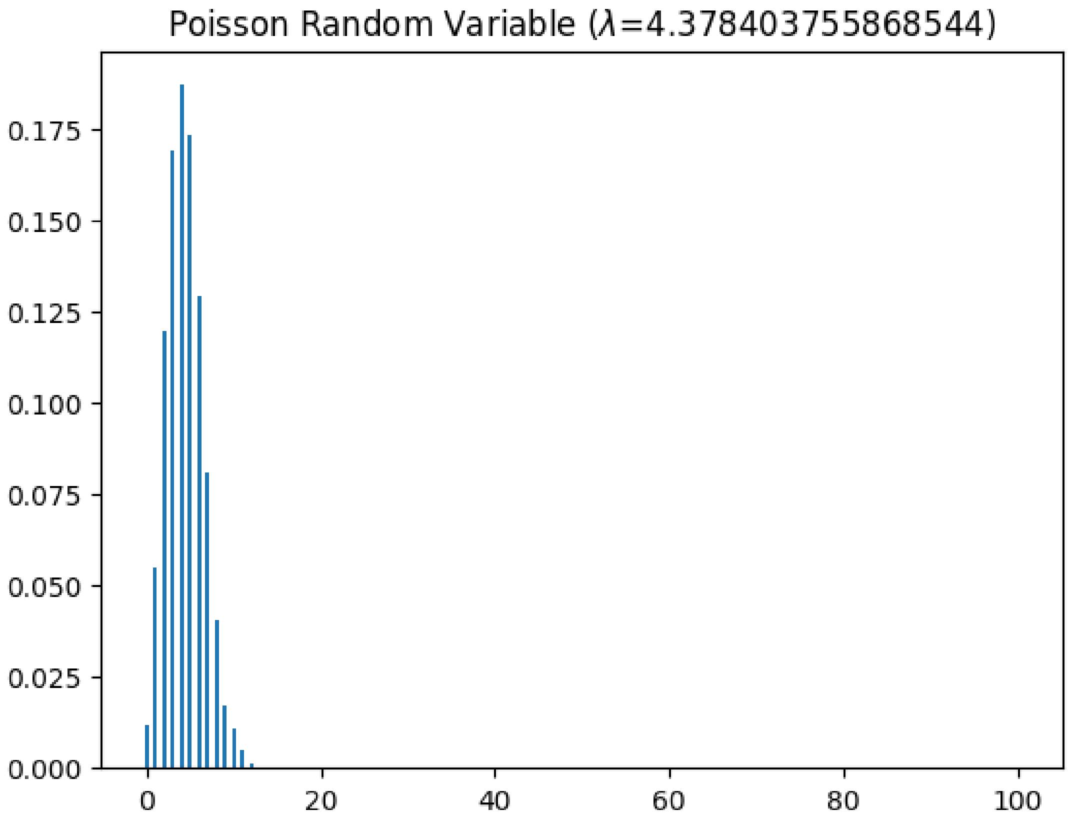
```
# Generate 4000 samples of a Uniform Random Variable with a = 1, b =
10.
a = 1 # parameter a of uniform RV
b = 10  # parameter b of uniform RV
size  = 4000 # Number of trials
X = np.random.randint(low=a,high=b+1, size=size)
plt.title("Uniform Random Variable (a=1, b=10)")
(bin_values_pmf_data, bins, patches) = plt.hist(X,bins=over_bins,
rwidth = 0.5, density=True, align='left' );
plt.show()

difference_in_bin_values = bin_values_historical_data -
bin_values_pmf_data
error = np.sum(abs(difference_in_bin_values))
print("Error when the Proposed PMF is Unifrom with parameter a=1 and
b=10: " + str(error))
```





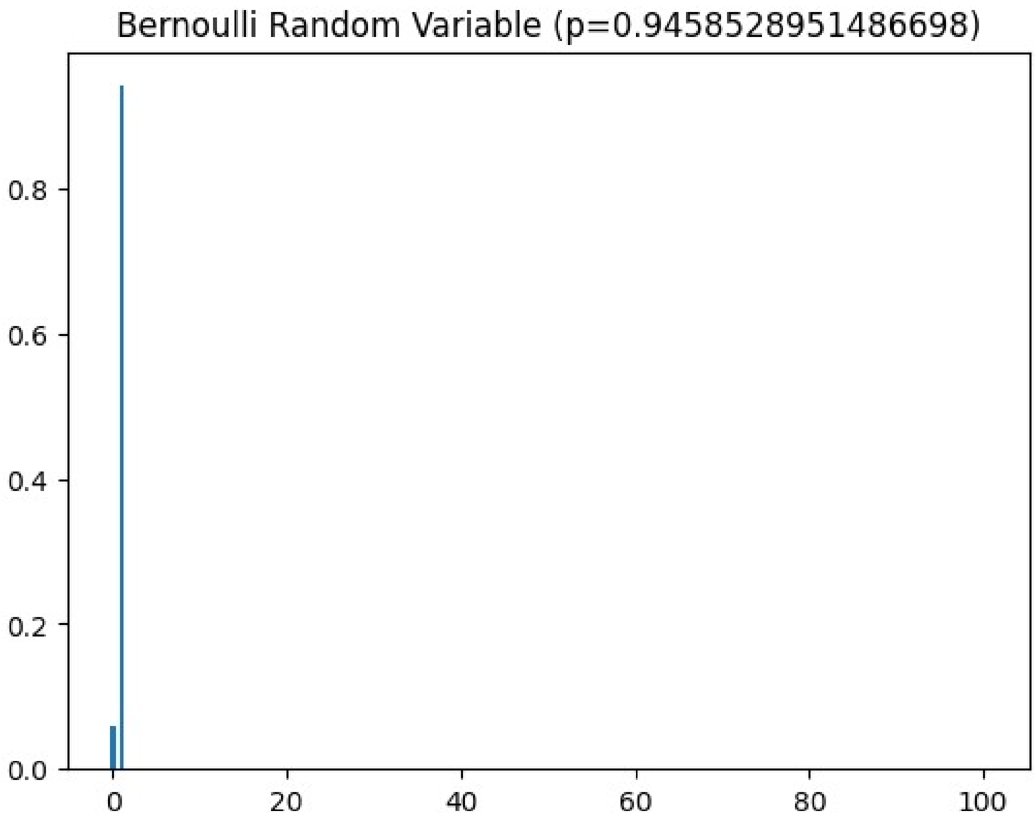

```
Error when the Proposed PMF is Unifrom with parameter a=1 and b=10:
0.44511345852895146
```

```
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
# Poisson distribution
lambda_value = np.mean(number_of_maiden_overs)
X_poisson = np.random.poisson(lambda_value,
size=len(number_of_maiden_overs))
(bin_values_pmf_poisson, bins, patches) = plt.hist(X_poisson,
bins=over_bins, rwidth=0.5, density=True, align='left')
plt.title(f"Poisson Random Variable ($\lambda$={lambda_value})")
plt.show()
error_poisson = np.sum(np.abs(bin_values_historical_data -
bin_values_pmf_poisson))
print("Error when the Proposed PMF is Poisson with parameter lambda="
+ str(lambda_value) + ": " + str(error_poisson))
```



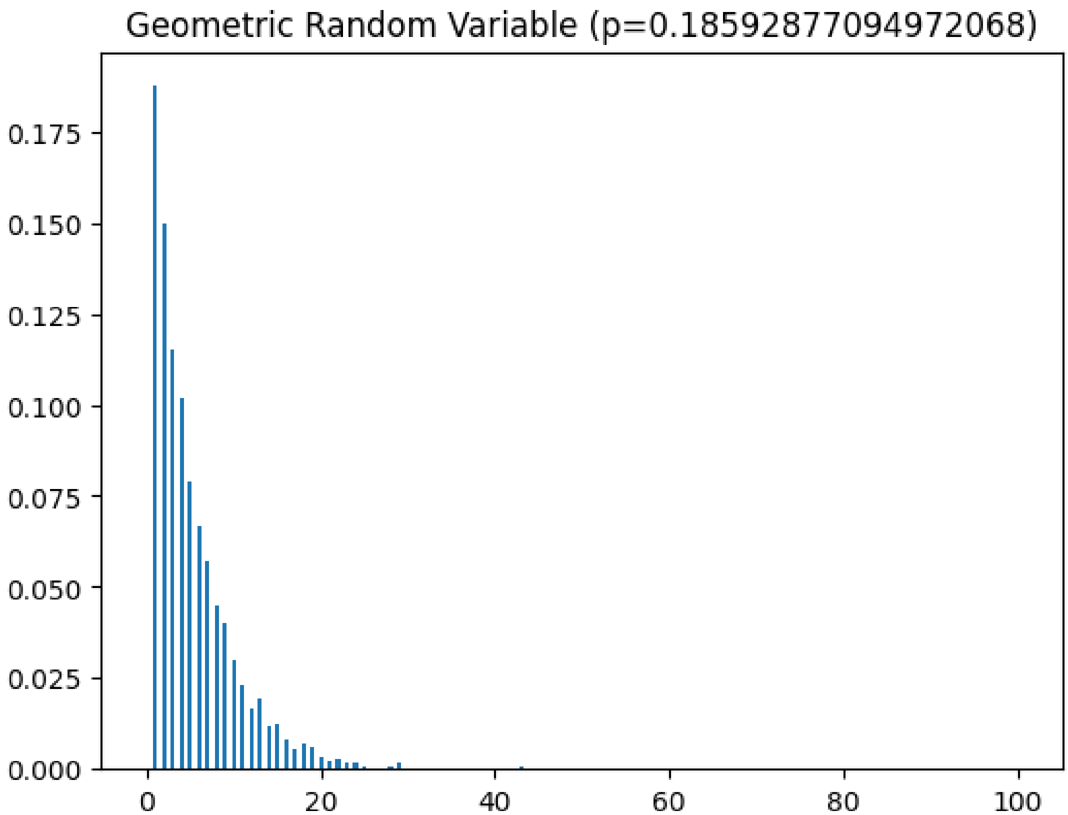

```
Error when the Proposed PMF is Poisson with parameter
lambda=4.378403755868544: 0.3436619718309859
```

```python
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
# Bernoulli distribution
p_bernoulli = np.mean(number_of_maiden_overs > 0)
X_bernoulli = np.random.binomial(1, p_bernoulli,
size=len(number_of_maiden_overs))
(bin_values_pmf_bernoulli, bins, patches) = plt.hist(X_bernoulli,
bins=over_bins, rwidth=0.5, density=True, align='left')
plt.title(f"Bernoulli Random Variable (p={p_bernoulli})")
plt.show()
error_bernoulli = np.sum(np.abs(bin_values_historical_data -
bin_values_pmf_bernoulli))
print("Error when the Proposed PMF is Bernoulli with parameter p=" +
str(p_bernoulli) + ": " + str(error_bernoulli))
```




```
Error when the Proposed PMF is Bernoulli with parameter
p=0.9458528951486698: 1.6913928012519561
```

```
# Name and ID: Mohit Rai (mr06638) and Moiz Zulfiqar (mz08229)
# Geometric distribution
p_geometric = 1 / (np.mean(number_of_maiden_overs) + 1)
X_geometric = np.random.geometric(p_geometric,
size=len(number_of_maiden_overs))
(bin_values_pmf_geometric, bins, patches) = plt.hist(X_geometric,
bins=over_bins, rwidth=0.5, density=True, align='left')
plt.title(f"Geometric Random Variable (p={p_geometric})")
plt.show()
error_geometric = np.sum(np.abs(bin_values_historical_data -
bin_values_pmf_geometric))
print("Error when the Proposed PMF is Geometric with parameter p=" +
str(p_geometric) + ": " + str(error_geometric))
```




```
Error when the Proposed PMF is Geometric with parameter
p=0.18592877094972068: 0.41377151799687006
```

Comments

Please provide your answer here. Also, please explain how your code in the previous section justifies your answer.

By studying the already given data set, we concluded that a Poisson Random Variable with lambda 4 would be best suited as the mean maiden overs across all ODI matches from the 21st Century is 4.38. Therefore, we expect it near a value of 4 maiden overs next week. This expectation is also justified by a lower error value as compared to the one given above.This suggests that the diversity and distribution pattern of maiden overs in one-day international cricket matches are well captured by the Poisson distribution. Consequently, the description might be changed to highlight that, among the PMFs taken into consideration, the Poisson distribution provides the most accurate depiction of the number of maiden overs in ODI cricket matches, making it the most appropriate model for this situation. The absolute error obtained by the other Random variables is much greater, hence they will not be considered.