Conditional Probability

Taking Input from the User

```
# To Find the Probability of A Given B: P[ A | B ]
x = float(input("Enter P [ A ∩ B ] : "))
y = float(input("Enter P[ B ] :"))
```

Calculating Probability

```
z = x/y
```

Proability

```
print(" P[ A | B ] = " + str(z))
```

```
 P[ A | B ] = 0.5
```

Example 1:

Susan took two tests. The probability of her passing both tests is 0.6. The probability of her passing the first test is 0.8. What is the probability of her passing the second test given that she has passed the first test?

OUTPUT:

```
Enter P [ A ∩ B ] : 0.6
Enter P[ B ] :0.8
 P[ A | B ] = 0.7499999999999999
```

Example 2:

A bag contains red and blue marbles. Two marbles are drawn without replacement. The probability of selecting a red marble and then a blue marble is 0.28. The probability of selecting a red marble on the first draw is 0.5. What is the probability of selecting a blue marble on the second draw, given that the first marble drawn was red?

OUTPUT:

```
Enter P [ A ∩ B ] : 0.28
Enter P[ B ] : 0.5
 P[ A | B ] = 0.56
```

Bayes

Taking input from the user

```
#P(B)
p_b = float(input("Enter P(B) : "))
# P(A|B)
p_a_given_b = float(input("Enter Probability of P(A|B): "))
# P(A)
p_a=  float(input("Enter P(A): "))
```

Caclulating the Probability

```
# calculate P(B|A)
result = float((p_a_given_b*p_b)/p_a)
```

Printing the Output

```
print('P( B | A ) = '+ str(result))
```

```
P( B | A ) = 0.4905405405405406
```

Example 1:

One of two boxes contains 4 red balls and 2 green balls and the second box contains 4 green and two red balls. By design, the probabilities of selecting box 1 or box 2 at random are 1/3 for box 1 and 2/3 for box 2. A box is selected at random and a ball is selected at random from it. Given that the ball selected is red, what is the probability it was selected from the first box?

OUTPUT:

```
Enter P(B) : 0.33
Enter Probability of P(A|B): 0.66
Enter P(A): 0.444
P( B | A ) = 0.4905405405405406
```

Example 2:

1% of a population have a certain disease and the remaining 99% are free from this disease. A test is used to detect this disease. This test is positive in 95% of the people with the disease and is also (falsely) positive in 2% of the people free from the disease. If a person, selected at random from this population, has tested positive, what is the probability that she/he has the disease?

OUTPUT:

```
Enter P(B) : 0.01
Enter Probability of P(A|B): 0.95
Enter P(A): 0.029
P( B | A ) = 0.3275862068965517
```

Bernoulli Distribution

Importing Libraries

```
from scipy.stats import bernoulli
import seaborn as sns
```

Taking Input from the user

```
p = float(input("Enter Probability of Success: "))
n = int(input("Enter NUmber of Ind. Bernoulli Trials: "))
```

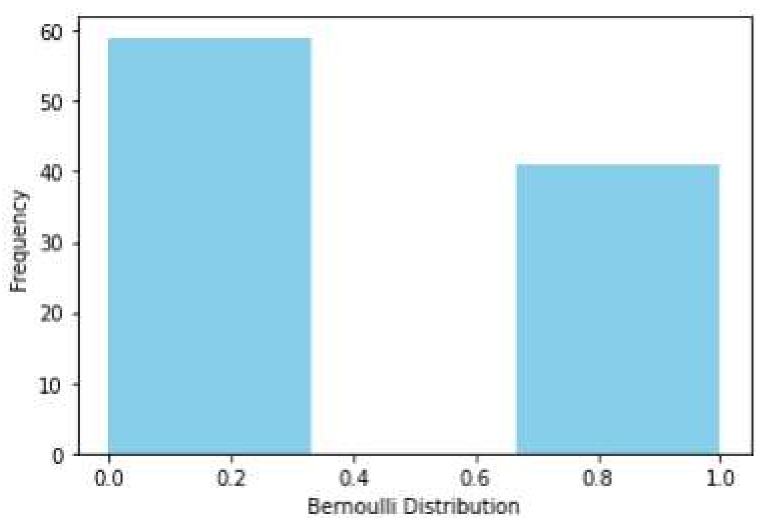
Calculating the Distribution

```
data_bern = bernoulli.rvs(size=n,p=p)
```

Plotting the output

```
ax= sns.distplot(data_bern,
                 kde=False,
                 color="skyblue",
                 hist_kws={"linewidth": 15,'alpha':1})
ax.set(xlabel='Bernoulli Distribution', ylabel='Frequency')
```

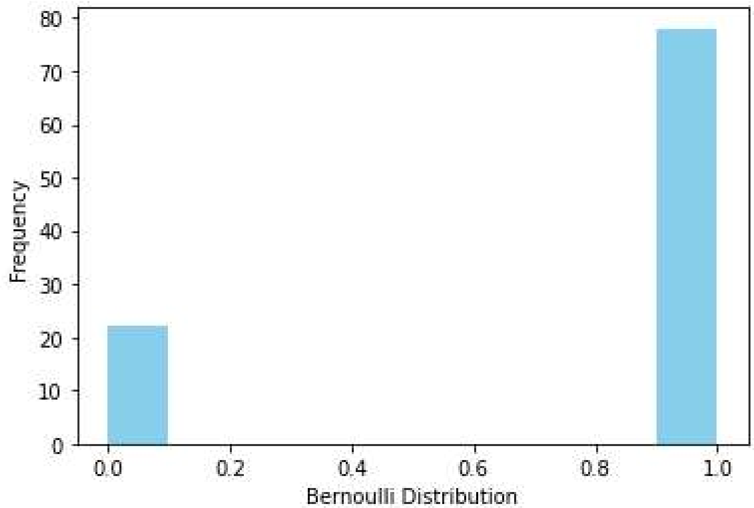
```
[Text(0.5, 0, 'Bernoulli Distribution'), Text(0, 0.5, 'Frequency')]
```



Example 1:

We have an unfair coin where the probability of success (p) or head is 0.8 and the probability of failure (q) or tail = 1-p = 1-0.8 = 0.2. If we denote head as 1 and tail as 0, we can plot this Bernoulli distribution as follows:

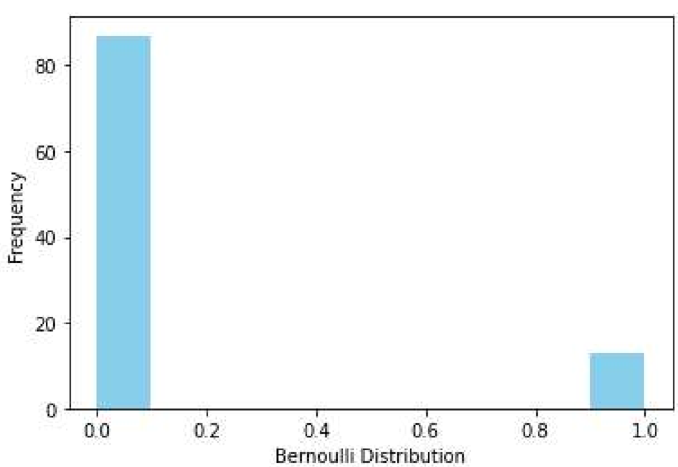
OUTPUT:



EXAMPLE 2:

The prevalence of a certain disease in the general population is 10%. If we randomly select a person from this population, we can have only two possible outcomes (diseased or healthy person). We call one of these outcomes (diseased person) success and the other (healthy person), a failure. The probability of success (p) or diseased person is 10% or 0.1. So, the probability of failure (q) or healthy person = 1-p = 1-0.1 = 0.9. If we denote diseased person as 1 and healthy person as 0, we can plot this Bernoulli distribution as follows:

OUTPUT:



Binomial Distribution

Imporintg Libraries

```
from scipy.stats import binom
import seaborn as sns
```

Taking input from the user

```
n = int(input("Enter NUmber of Ind. Bernoulli Trials: "))
p = float(input("Enter Probability of Success: "))
```

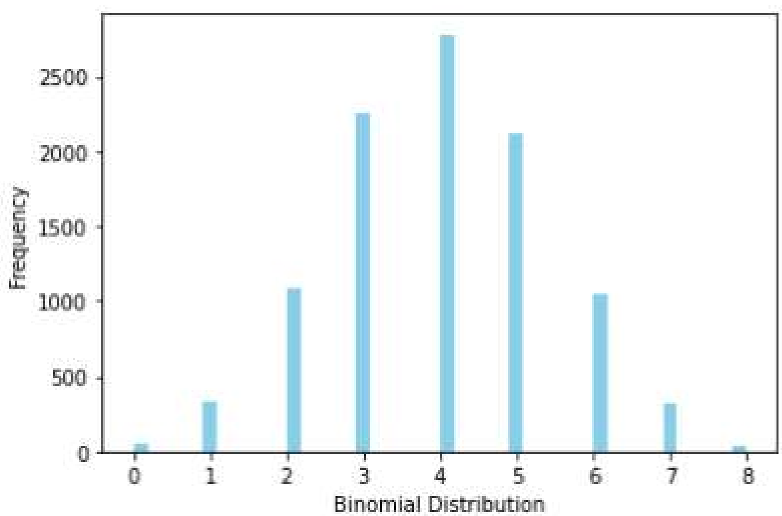
Calculating the Binomial Distribution

```
data_binom = binom.rvs(n=n,p=p,size=10000)
```

Plotting the Output

```
ax= sns.distplot(data_binom,
                 kde=False,
                 color="skyblue",
                 hist_kws={"linewidth": 15,'alpha':1})
ax.set(xlabel='Binomial Distribution', ylabel='Frequency')
```

```
[Text(0.5, 0, 'Binomial Distribution'), Text(0, 0.5, 'Frequency')]
```

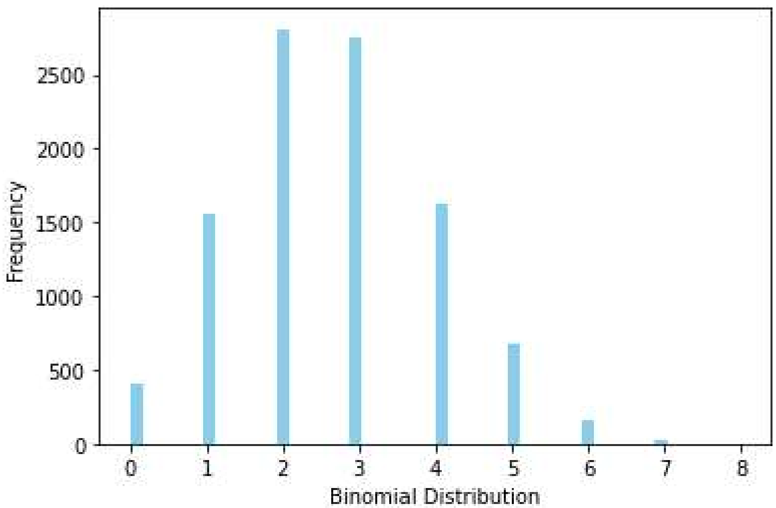


EXAMPLE:

Let k be the number of active speakers in a group of 8 speakers. Suppose that a speaker is active with probability 1/3. Find the probability that the number of active speakers is greater than 6.

OUTPUT:

```
Enter NUmber of Ind. Bernoulli Trials: 8
Enter Probability of Success: 0.33
```



Geometric Probality Law

Definition:

Geometric distribution -

```
                     A discrete random variable X is said to have
a geometric distribution if it has a probability density function
(p.d.f.) of the form:
                                          P(X = x) = q(x-1)p, where
q = 1 - p
```

Taking Values from the user

```
p = float(input("Enter Probability of Success: "))
q = float(input("Enter Probability of Failure: "))
m = int(input("Enter trial results in Success: "))
```

Calculating Probabily of fisrt success

```
prob_of_event = p*q ** (m-1)
prob_of_event = float(prob_of_event)
```

Displaying the Probability

```
print("Probability of Success at " + str(m) +"th trial : " + str(prob_of_event))
```

```
Probability of Success at 5th trial : 0.00567
```

EXAMPLE:

In an amusement fair, a competitor is entitled for a prize if he throws a ring on a peg from a certain distance. It is observed that only 30% of the competitors are able to do this. If someone is given 5 chances, what is the probability of his winning the prize when he has already missed 4 chances?

OUTPUT:

```
0.33
Enter Probability of Success: (Press 'Enter' to confirm or 'Escape' to cancel)

0.66
Enter Probability of Failure: (Press 'Enter' to confirm or 'Escape' to cancel)

5
Enter trial results in Success: (Press 'Enter' to confirm or 'Escape' to cancel)

Probability of Success at 5th trial : 0.06261662880000002
```

EXAMPLE:

Suppose a researcher is waiting outside of a library to ask people if they support a certain law. The probability that a given person supports the law is p = 0.2.

OUTPUT:

```
0.2
Enter Probability of Success: (Press 'Enter' to confirm or 'Escape' to cancel)

0.8
Enter Probability of Failure: (Press 'Enter' to confirm or 'Escape' to cancel)

3
Enter trial results in Success: (Press 'Enter' to confirm or 'Escape' to cancel)

Probability of Success at 3th trial : 0.12800000000000003
```

Probabilty Mass Function

Importing Libraries

```
from scipy.stats import geom
import matplotlib.pyplot as plt
import numpy as np
```

Taking Input from the user

```
p = float(input("Enter Probability of Success: "))
```

Calculating the PMF

```
# creating an array of values between
# 1 to 4 with a difference of 1
k = np.arange(1, 5, 1)
# k     Required. Specify float or array_like of floats representing number of Berı
# p     Required. Specify probability of success in each trial, must be in range ((
geom_pd = geom.pmf(k, p)
```

Plotting the Output

```
fig, ax = plt.subplots(1, 1, figsize=(10, 10))
ax.plot(k, geom_pd, 'bo', ms=8, label='geom pmf')
plt.ylabel("Probability", fontsize="12")
plt.xlabel("Number of Bernouli Trials", fontsize="12")
plt.title("Geometric Distribution - PMF ", fontsize="18")
ax.vlines(k, 0, geom_pd, colors='b', lw=5, alpha=0.5)
plt.show()
```