```
# Q1. What is the Probability density function?
```

The Probability Density Function (PDF) is a fundamental concept in probability theory and statistics. It is used to describe the probability distribution of a continuous random variable.

Probability Density Function is a mathematical function that characterizes the distribution of a continuous random variable and is essential for various statistical analyses and probability calculations.

In Simple Terms, The probability density function (PDF) is a statistical expression that defines the probability that some outcome will occur.

```
# Q2. What are the types of Probability distribution?
```

Types of Probability Distributions Statisticians divide probability distributions into the following types:

Discrete Probability Distributions Continuous Probability Distributions

Discrete Probability Distributions Discrete probability functions are the probability of mass functions. It assumes a discrete number of values.

Continuous Probability Distributions

A continuous probability distribution is one in which a continuous random variable X can take on any value within a given range of values – which can be infinite, and therefore uncountable. For example, time is infinite because you could count from 0 to a billion seconds, a trillion seconds, and so on, forever.

```
# Q3. Write a Python function to calculate the probability density function of a normal distribution with given mean and standard deviation at a given point.

# A normal distribution is a type of continuous probability distribution for a real-valued random variable.
# It is based on mean and standard deviation. The probability distribution function or PDF computes the likelihood of a single point in the distribution.

# PDF(x) = (1 / (std_dev * sqrt(2 * pi))) * exp(-0.5 * ((x - mean) / std_dev)^2) <----  formula to calculate PDF

# μ is the mean
# σ is the standard deviation of the distribution
# x is the number


import math

def normal_pdf(x, mean, std_dev):
```

```
    exponent = -0.5 * ((x - mean) / std_dev)**2
    coefficient = 1 / (std_dev * math.sqrt(2 * math.pi))
    pdf_value = coefficient * math.exp(exponent)
    return pdf_value



mean = 0
std_dev = 1
point = 1.5
pdf_at_point = normal_pdf(point, mean, std_dev)
print(f"PDF at point {point}: {pdf_at_point}")
```

```
PDF at point 1.5: 0.12951759566589174
```

```
# Using Scipy
<---------------------------------------------------------------------

from scipy.stats import norm
import numpy as np

data_start = -5
data_end = 5
data_points = 11
data = np.linspace(data_start, data_end, data_points)

mean = np.mean(data)
std = np.std(data)

probability_pdf = norm.pdf(3, loc=mean, scale=std)
print(probability_pdf)
```

```
0.0804410163156249
```

```
# Q4. What are the properties of Binomial distribution? Give two
examples of events where binomial distribution can be applied
```

Properties of Binomial Distribution The properties of the binomial distribution are:

There are two possible outcomes: true or false, success or failure, yes or no. There is 'n' number of independent trials or a fixed number of n times repeated trials. The probability of success or failure remains the same for each trial. Only the number of success is calculated out of n independent trials. Every trial is an independent trial, which means the outcome of one trial does not affect the outcome of another trial.

Examples of events where the binomial distribution can be applied:

Coin Flips: When flipping a biased coin multiple times, each flip can be considered a Bernoulli trial with two outcomes: heads (success) or tails (failure). The binomial distribution can be used to calculate the probability of getting a specific number of heads in a certain number of coin flips.

Quality Control: In manufacturing, when inspecting a batch of items for defects, each item can be considered a Bernoulli trial. The binomial distribution can be applied to determine the probability of finding a certain number of defective items in the batch based on the overall defect rate and the batch size.

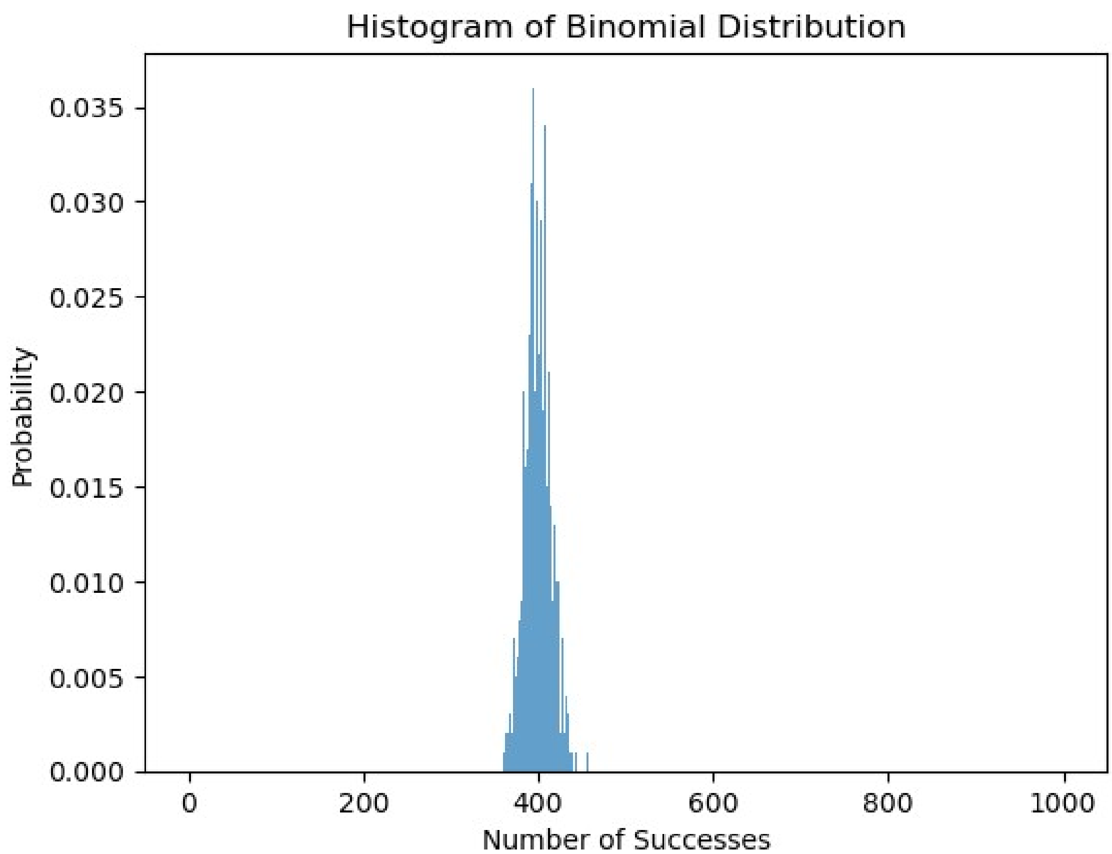
```
# Q5. Generate a random sample of size 1000 from a binomial
distribution with probability of success 0.4 and plot a histogram of
the results using matplotlib.


import numpy as np
import matplotlib.pyplot as plt

# Parameters for the binomial distribution
n = 1000  # Number of trials
p = 0.4   # Probability of success

# Generate random sample
random_sample = np.random.binomial(n, p, size=1000)

# Plot histogram
plt.hist(random_sample, bins=np.arange(0, n+1), density=True,
alpha=0.7)
plt.title('Histogram of Binomial Distribution')
plt.xlabel('Number of Successes')
plt.ylabel('Probability')
plt.show()
```




```python
# Q6. Write a Python function to calculate the cumulative distribution function of a Poisson distribution with given mean at a given point.



import scipy.stats as stats

def poisson_cdf(mean, x):
    cdf = stats.poisson.cdf(x, mu=mean)
    return cdf

# Example usage
mean = 3.5
x = 2
cdf_value = poisson_cdf(mean, x)
print(f"CDF at x={x}: {cdf_value}")
```

```
CDF at x=2: 0.32084719886213414
```

```python
# Q7. How Binomial distribution different from Poisson distribution?
```

In a Binomial distribution, there is a fixed number of trials (e.g. flip a coin 3 times) In a Poisson distribution, there could be any number of events that occur during a certain time interval (e.g. how many customers will arrive at a store in a given hour?)

Binomial Distribution: In the binomial distribution, the probability of success remains constant for each trial. It is denoted as "p."

Poisson Distribution: The Poisson distribution does not directly involve a probability of success. Instead, it is characterized by the average rate of occurrence of events, denoted as "λ" (lambda).

```
# Q8. Generate a random sample of size 1000 from a Poisson
distribution with mean 5 and calculate the sample mean and variance



import numpy as np

# Parameters for the Poisson distribution
mean = 5
sample_size = 1000

# Generate random sample
random_sample = np.random.poisson(mean, size=sample_size)

# Calculate sample mean and variance
sample_mean = np.mean(random_sample)
sample_variance = np.var(random_sample, ddof=1)  # ddof=1 for sample
variance

print(f"Sample Mean: {sample_mean}")
print(f"Sample Variance: {sample_variance}")
```

```
Sample Mean: 4.944
Sample Variance: 4.913777777777777
```

```
# Q9. How mean and variance are related in Binomial distribution and
Poisson distribution?
```

Both the mean and variance of probability distributions provide insights into the central tendency and the spread of the distribution. The relationship between mean and variance differs between the binomial distribution and the Poisson distribution.

Binomial Distribution: For a binomial distribution with parameters "n" (number of trials) and "p" (probability of success), the mean (μ) and variance (σ^2) are related as follows:

Mean: μ = n * p Variance: σ^2 = n * p * q, where q = 1 - p

Poisson Distribution: For a Poisson distribution with parameter "λ" (average rate of occurrence of events), the mean (μ) and variance (σ^2) are related as follows:

Mean: μ = λ Variance: σ^2 = λ

```
# Q10. In normal distribution with respect to mean position, where
does the least frequent data appear?
```

In a normal distribution (also known as a Gaussian distribution), the least frequent data points are found in the tails of the distribution, farthest from the mean.

The normal distribution is symmetric, and its shape resembles a bell curve, with the mean (μ) at the center.