Probability Distribution in Python

If you are performing any random experiment then it may be possible that some outcome has more chances to occur than others, this is called a probability distribution. To calculate this probability distribution, the main functions are the probability mass function and probability density function based on the data type.

There are two types of data:

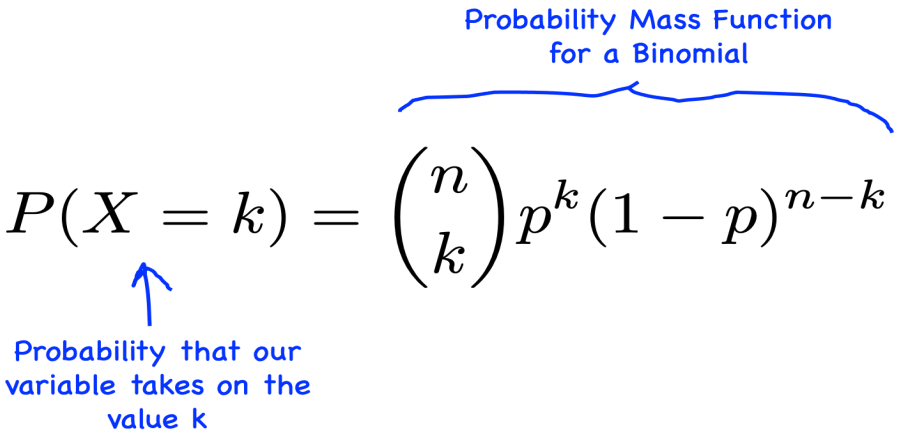
1. **Discrete**, which can take only specified values. The probability distribution for this is calculated using the Probability Mass Function.
2. **Continuous**, which can take any value within a given range. The probability distribution for this is calculated using the Probability Density Function.

Here, we are going to learn how we can calculate the probability distribution for various types of data in python.

Probability Mass Function

Discrete Probability Distribution is also known as Probability Mass Function. Examples of this are tossing a coin or rolling a die. Each possible value has a non-zero probability and the probabilities for all possible outcomes must sum to one.

The formula for PMF for a binomial distribution is:



The probability that our variable takes on the value k is calculated as the product of nCr and the probability of success(p) raises to the number of successes(k) and the probability of failure(1-p) to the number of failures(n-k).

In python binom.pmf() method is used to calculate the probability mass function. The binom class is present in the scipy library's stats module.

To use the binom.pmf function, you must import scipy at the very start of the program:

```
from scipy.stats import binom
```

The syntax for binom.pmf method

```
scipy.stats.binom.pmf(r,n, p)
```

Parameters
The binom.pmf function takes in three parameters:

- **n**: the total number of trials or times the experiment will be carried out.
- **r**: a list of integers from 0 to n, inclusive.
- **p**: the probability that the outcome of a single experiment will be a success. The value of p must be between 0 and 1, inclusive.

Note: Binomial distribution is used to model experiments that have only one of two outcomes, success or failure.

Return value
The binom.pmf method returns a list with the same number of values and sequences as r. The return value is the probability mass function for the values in r.

Example:
Let's say we flip a coin 5 times and calculate the probability of getting heads 3 times.

```
from scipy.stats import binom

print(binom.pmf(k=3, n=5, p=0.5))
```

In the above code:
K = 3 means getting 3 heads
N = 5 means repeating the experiment 5 times.
P = 0.5 means the probability of a head is 0.5 for each experiment.

The output of this is "0.3125" which means there is a 31.25% chance of getting 3 heads after flipping the coin 5 times.

Here we learned about the pmf. In future videos, we are going to learn about the pdf function.