Probability Distributions

In this lesson, probability distribution and its types are discussed.

We'll cover the following

Definition

The probability distribution function informs us what the probabilities of a range of outcomes defined in a random variable will be.

$$R_x = [1, 2, 3, 4]$$

$$P_x = [0.95, 0.02, 0.01, 0.02]$$

Here, R_x is the random variable containing some outcomes, and the values in P_x represent the probabilities of those outcomes, respectively.

Probabilities can be displayed of simple occurrences like tossing a coin to more complex occurrences, like the probability of a specific steroid treating a disease or not.

Data types

There are mainly two types of data that we encounter while figuring out the probability of events:

- **Discrete data:** A random variable **X** has discrete data if the data it contains is finite, countable, and has specific values. For example, the outcome of a number of students in a class can only be a specific number like **50** or **60** and

cannot be **50.5** or **57.7**. If a die is rolled, the only outcomes are **1, 2, 3, 4, 5, and 6**, which are finite, countable, and specific whole numbers.

- **Continuous data:** A random variable **X** has continuous data if all the data points it contains are in specific ranges. The range can either be finite or infinite. For example, if we have the data for heights of different buildings, we cannot find the probability that the height of a building is exactly **700** cm, but we can find the probability that the height of a building is between **650 - 750** cm. The height can be **700.075** or **699.874**, but not exactly **700**.

Types of the probability distribution

There are different types of probability distribution functions used to model various kinds of data. For this course, we will discuss the most commonly used probability distributions.

Uniform distribution

Events, where each and every outcome in a random variable has the same probability of occurrence will create a **uniform distribution** of probabilities. This means that if a random variable has **n** number of outcomes, each of those outcomes has a probability of occurrence equal to **1/n**.

Uniform probability distribution can be created for both discrete and continuous data.

- **Discrete uniform distribution**: It uses discrete data and contains a finite number of outcomes that all have the same probability of occurrence. For example, a rolling die and a coin toss are excellent examples of this. The probability of all the outcomes of both of the above cases is the same, and since they contain a finite number of outcomes, they form a discrete uniform distribution.

- **Continuous uniform distribution**: It uses continuous data, and the outcomes are either in a range or infinite and have the same probability of occurrence. For example, a random number generator is an excellent example of this. In this example, there is an endless number of outcomes that can exist, and every number has an equal possibility of occurrence. So, a continuous uniform distribution is formed for this.

Binomial distribution

This distribution gives probabilities of a discrete random variable having only two

possible outcomes in an experiment that is repeated multiple times. Therefore, in

this type of distribution, the probabilities for discrete values can be found. The function used to calculate binomial distribution is known as **probability mass function**.

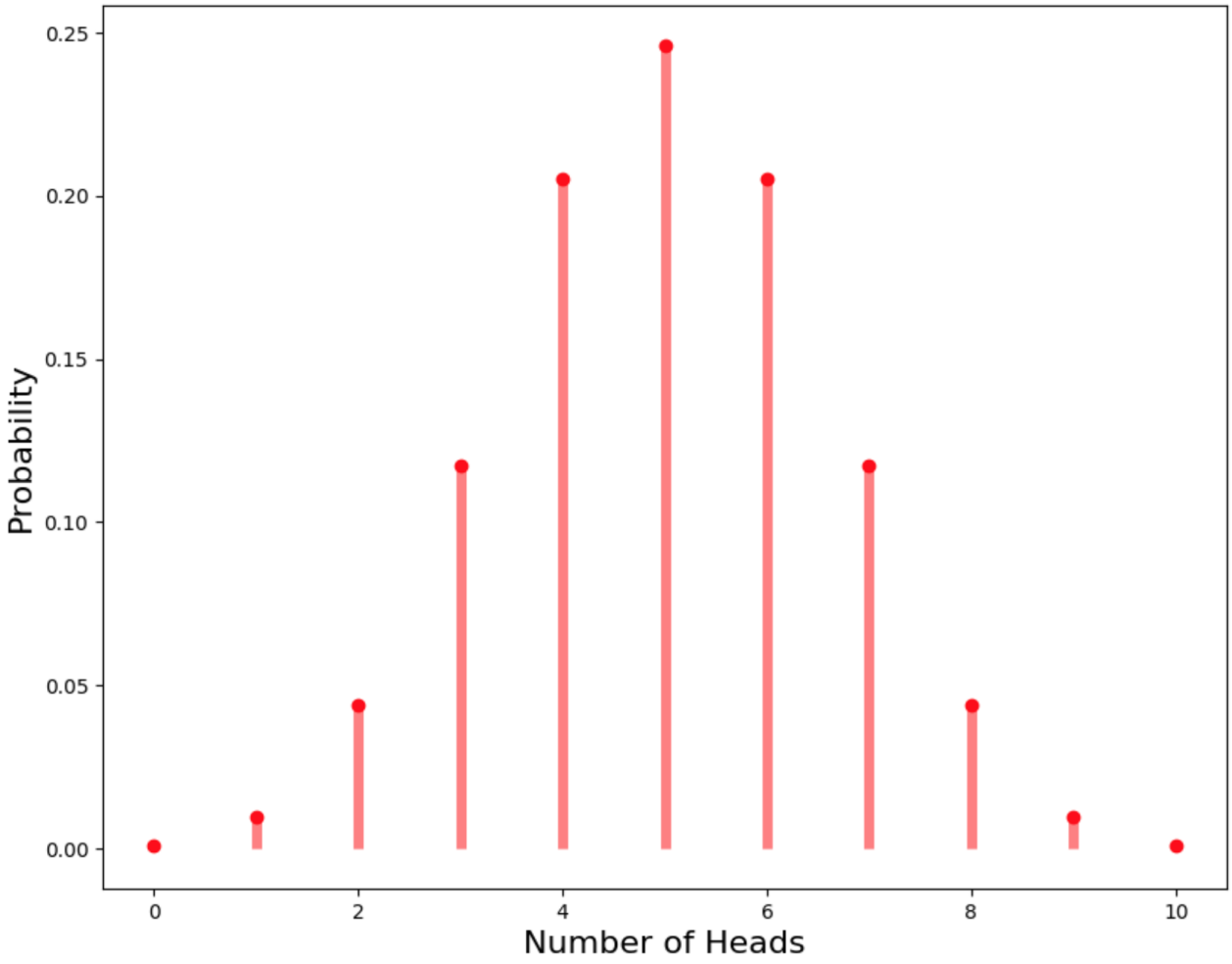
This method can be applied when only two outcomes are possible for each experiment performed, i.e., true or false, success or fail, win or lose, considering that each new outcome is independent of all other outcomes, meaning that results from one experiment should not have any impact on the result of another experiment.

For example, suppose that a coin is tossed *ten* times and the count of heads appearing each time is getting stored. Now, this cycle of tossing a coin *ten* times is repeated **n** number of times, and we want to know what is the probability of exactly *five* heads occurring in each cycle. As the probability of both heads and tails is 50-50 and the outcome of each toss can only be either **heads** or **tails**, it can be considered a binomial distribution problem.

Luckily, Python provides a built-in function for calculating probabilities using the binomial distribution, so we don't need to get into the formulas. The `pmf()` function, which translates to the **probability mass function**, is used to solve this problem.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import binom

# Number of experiments
n = 10
# Probability of success
p = 0.5

# Array of probable outcomes of number of heads
x = range(0,11)

# Get probabilities
prob = binom.pmf(x, n, p)

# Set properties of the plot
fig, binom_plot = plt.subplots(figsize=(10,8))
binom_plot.set_xlabel("Number of Heads",fontsize=16)
binom_plot.set_ylabel("Probability",fontsize=16)
binom_plot.vlines(x, 0, prob, colors='r', lw=5, alpha=0.5)

# Plot the graph
binom_plot.plot(x, prob, 'ro')
```

```
plt.show()
```



The above graph shows the probability of each number of heads appearing during the **ten** tossings of the coin. So, for our question, the probability that heads appear **five** times is the highest according to the result of the binomial distribution.

On **line** 7, `n` is declared, which is the number of times the coin is tossed.

On **line 9**, `p` is declared which is the probability of occurrence of heads in each toss, which in our case is **50%** so `p` is assigned `0.5`.

On **line 12**, `x` is defined. `x` contains the probable outcomes for *heads*, which in our case can be from `0 - 10`. **0** represents that no heads occurred, and **10** represents that in all *ten* tosses, only heads occurred.

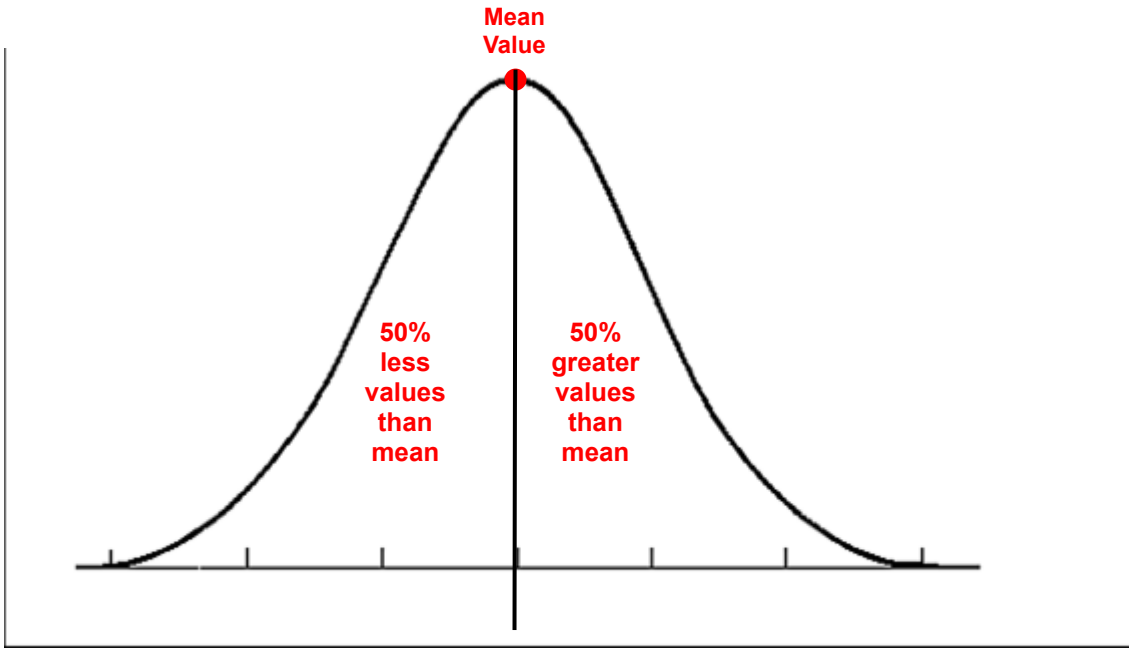
On **line 15**, the `binom.pmf()` function of the `scipy` Python package is used to calculate the binomial distribution, which returns the probabilities of all occurrences. This function takes the probable outcome array, the number of experiments and the expected probability of *heads* as parameters.

experiments and the expected probability of *heads* as parameters.

Normal distribution

This distribution uses a continuous random variable, so it is a certain type of continuous probability distribution. The results from a normal distribution are always centered around the *mean* or *average* value. This means that 50% of the values are below the mean, 50% are above the mean, and the majority of the values are close to the mean value. The function used to calculate probabilities in a normal distribution is known as a **probability density function**.

The following graph shows how normal distribution values can be displayed.



This graph plots the resultant values after computing normal distribution and is known as the ***bell curve*** as it looks like a bell in shape. As mentioned above, most of the values are close to the *average* value, so the following facts for normal distribution have been observed:

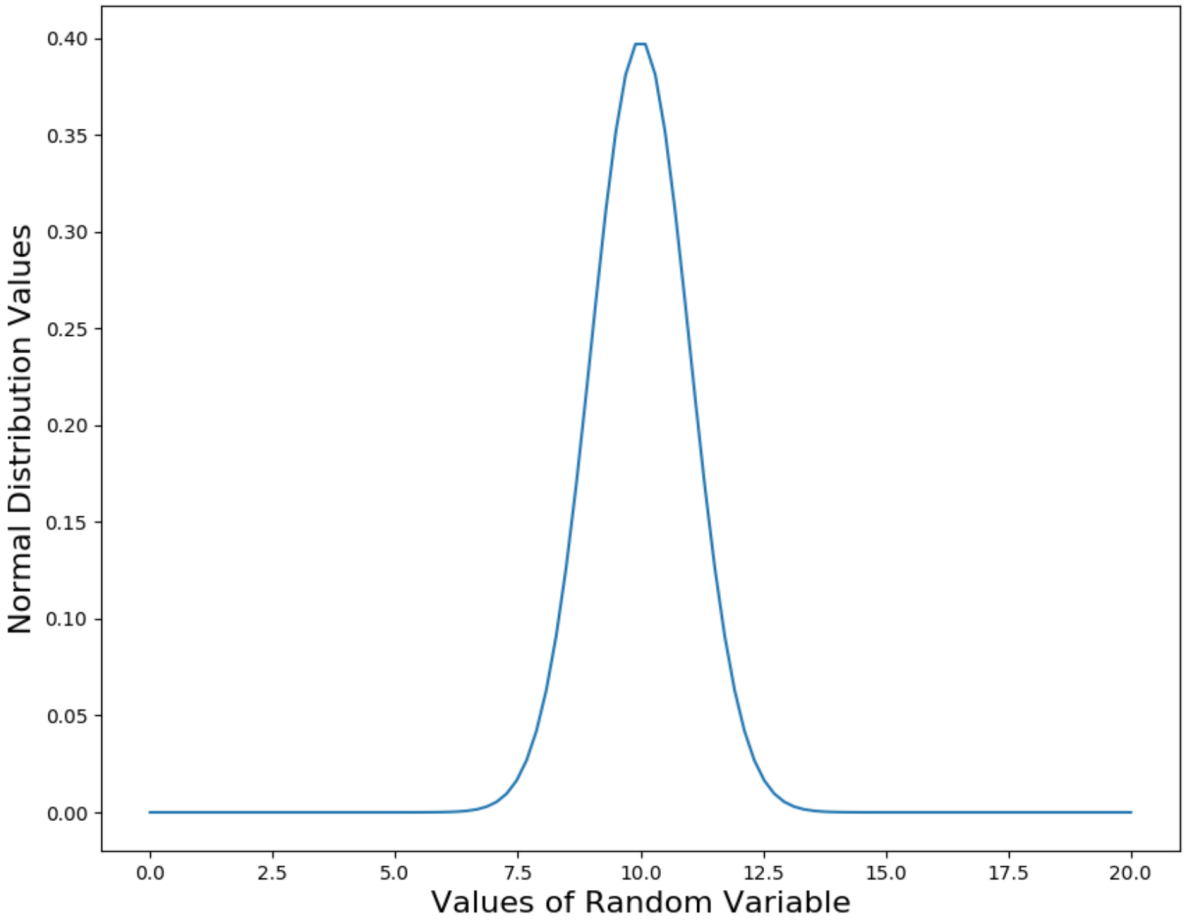
- **68%** of the values are within **one** standard deviation from the *mean*.
- **95%** of the values are within **two** standard deviations from the *mean*.
- **99.7%** of the values are within **three** standard deviations from the *mean*.

The normal distribution graph and the above information are used when we want to know if a given set of data follows a normal trend or not. The normal trend is when the normal distribution of a dataset produces a graph just like above, and the dataset can then be considered stable. If the resultant graph is left-skewed **(** the left curve is longer **)** or right-skewed **(** the right curve is longer **)**, then it indicates

that the dataset is not stable. For example, if a normal distribution graph is computed for a year's worth of data of some company's stock and the graph for it comes out to be perfect like above, it means that the company is stable, has steady growth, and is suitable for investing. If the graph comes out to be left or right-skewed, then it indicates some unstable elements depending on how much skewness is observed.

Luckily again, Python provides a built-in function for calculating normal distribution so we don't need to get into the formulas. The `pdf()` function which translates to the **probability density function**, is used to solve the following example.

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

# Generate 100 random numbers between 0 and 20
Rv = np.linspace(0.0, 20.0, 100)

#calculate the normal distribution
nd = scipy.stats.norm.pdf(Rv,Rv.mean())

# Set properties of the plot
fig, nd_plot = plt.subplots(figsize=(10,8))
nd_plot.set_xlabel("Values of Random Variable", fontsize=16)
nd_plot.set_ylabel("Normal Distribution Values", fontsize=16)

# Plot the graph
nd_plot.plot(Rv,nd)
plt.show()
```



A perfect normal distribution graph can be observed above. It is intentionally made like this using perfect data.

On **line 6**, the `linespace` function is used to generate *one-hundred* evenly spaced integers, meaning *one-hundred* numbers with the same difference in ascending order. Therefore, half of the numbers are below mean and half are above mean.

On **line 9**, the `norm.pdf()` function is used to calculate the normal distribution values. This function took the random variable and its mean as parameters.

In the next lesson, the central limit theorem is discussed.