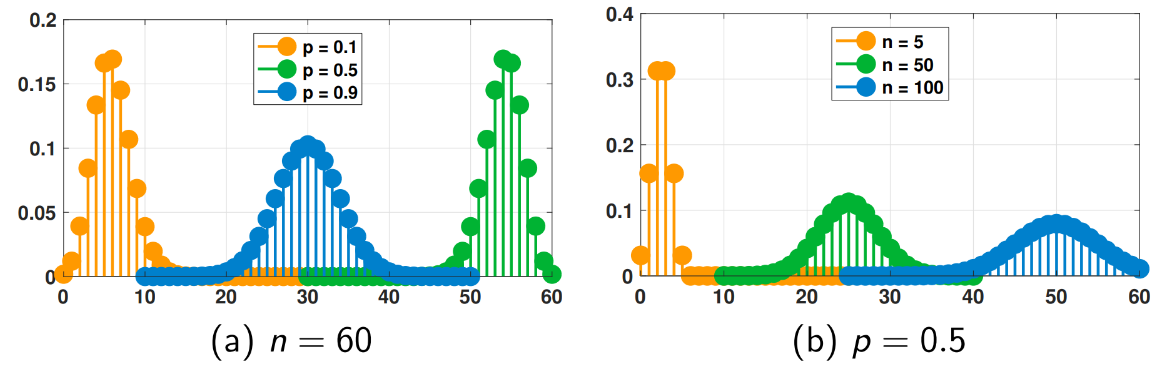
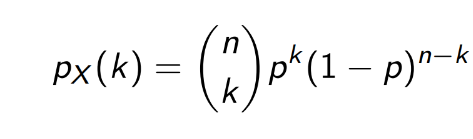
# **Binomial Discrete Distribution**

Definition:

A binomial distribution is a discrete probability distribution that models the number of successes in a fixed number of independent Bernoulli trials, where each trial has the same probability of success, denoted by "p." The distribution is characterized by two parameters: the number of trials, denoted by "n," and the probability of success, denoted by "p."

PMF Formula:



Where:

n : is number of trials

p : probability of success in one trial

k: number of successes

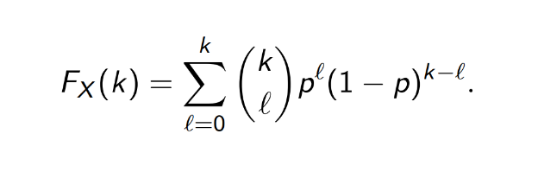
Properties of Uniform distribution:

Mean(E(x))= np

E(x²)= np(np+(1-p))

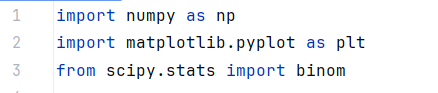
Var(x)= np(1-p)

*CDF:*



The Code:

We start by importing the necessary libraries:

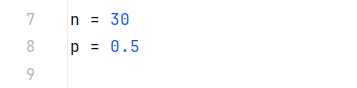


Line 1: **numpy** library helps in mathematical and logical operations on arrays

Line 2 : we use **matplotlib** library for used for plotting the pmf and cdf

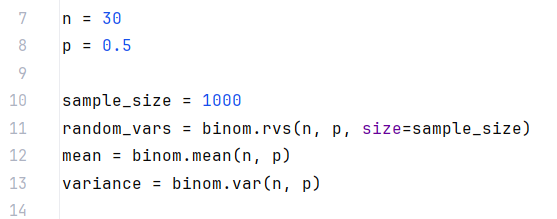
Line 3 : the **scipy.stats** library used for statistical and probabilistic operations

After that we set the parameters of the Binomial Distribution

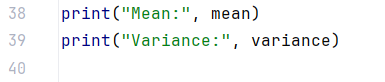


N is number of trials while p is probability of success note that p should be a value between 0 and 1

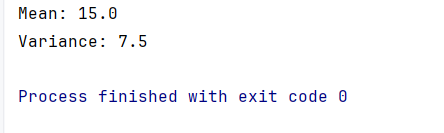
After that we generate the Mean, Variance and random variables



The binom.rvs() , binom.mean() and binom.var() functions are associated with the **scipy.stats** library and used for calcu;ating mean and variance and random variable using n and p ‘as in the context of the code’

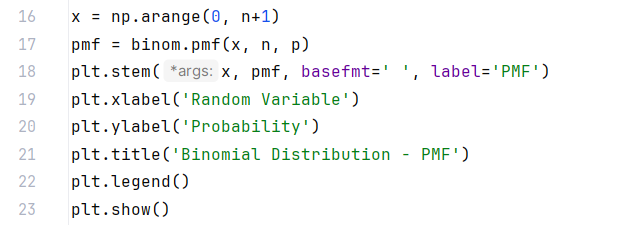
We use the following code to print mean and variance:

The output:



After all that we will plot the PMF and CDF:

**PDF:**

****

For the previous code:

-np.arange() This function creates an array x using **numpy's** `arange` function. The array contains values from 0 to n

-binom.pmf() this function calculates the PMF values using X array and n and p which are previously initiated

-plt.stem function used to create the stem plot pf PMF x array is the x axis

While pmf array is the y axis , we used **basefmt** to clear the base line

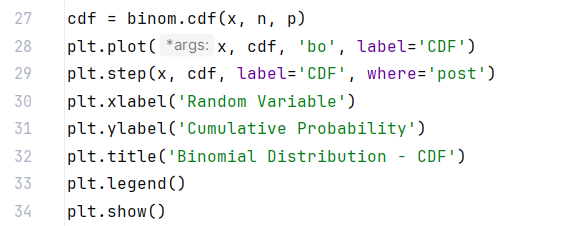
- plt.xlabel() and plt.ylabel() sets the labels of x-axis and y-axis

-plt.title() displays the title of the plot

-plt.legend() displays the legend of the plot

-plt.show() displays the plot on the screen

**CDF:**

****

for plotting CDF we will do the same as in PMFand just create a new array ‘cdf’

-binom.cdf() to create array ‘cdf’ and calculating CDF values by using x and n and p

-plt.plot() to plot CDF as dot by using (‘bo’) we may not use it and just use plt.step()

-plt.step() to plot CDF as a step

The last 5 lines are the same as the ones used in plotting PMF

**Plotting Output:**

