The binomial distribution model deals with finding the probability of success of an event which has only two possible outcomes in a series of experiments. For example, tossing of a coin always gives a head or a tail. The probability of finding exactly 3 heads in tossing a coin repeatedly for 10 times is estimated during the binomial distribution.

dbinom(x, size, prob)

pbinom(x, size, prob)

qbinom(p, size, prob)

rbinom(n, size, prob)

Following is the description of the parameters used −

* **x** is a vector of numbers.
* **p** is a vector of probabilities.
* **n** is number of observations.
* **size** is the number of trials.
* **prob** is the probability of success of each trial.
* dbinom()
* This function gives the probability density distribution at each point.
* [Live Demo](http://tpcg.io/wZ05YE)
* # Create a sample of 50 numbers which are incremented by 1.
* x <- seq(0,50,by = 1)
* # Create the binomial distribution.
* y <- dbinom(x,50,0.5)
* # Give the chart file a name.
* png(file = "dbinom.png")
* # Plot the graph for this sample.
* plot(x,y)
* # Save the file.
* dev.off()
* pbinom()
* This function gives the cumulative probability of an event. It is a single value representing the probability.
* [Live Demo](http://tpcg.io/SKWAGl)
* # Probability of getting 26 or less heads from a 51 tosses of a coin.
* x <- pbinom(26,51,0.5)
* print(x)
* qbinom()
* This function takes the probability value and gives a number whose cumulative value matches the probability value.
* [Live Demo](http://tpcg.io/IkEstq)
* # How many heads will have a probability of 0.25 will come out when a coin
* # is tossed 51 times.
* x <- qbinom(0.25,51,1/2)
* print(x)
* rbinom()
* This function generates required number of random values of given probability from a given sample.
* [Live Demo](http://tpcg.io/5KLmSt)
* # Find 8 random values from a sample of 150 with probability of 0.4.
* x <- rbinom(8,150,.4)
* print(x)

We use the seaborn python library which has in-built functions to create such probability distribution graphs. Also, the scipy package helps is creating the binomial distribution.

from scipy.stats import binom

import seaborn as sb

binom.rvs(size=10,n=20,p=0.8)

data\_binom = binom.rvs(n=20,p=0.8,loc=0,size=1000)

ax = sb.distplot(data\_binom,

kde=True,

color='blue',

hist\_kws={"linewidth": 25,'alpha':1})

ax.set(xlabel='Binomial', ylabel='Frequency')

ax.plot()