**ACTIVIDAD #1**

**Lesson 2: Delving into probabilities and various types of distributions in the context of data analysis.**

**Tipo actividad: Reading: "Types of Probability Distributions Every Data Science Expert Should know", and matching activity**

**Discuss some keywords of the reading "Types of Probability Distributions Every Data Science Expert Should know"**

1. **Probability Distribution:** A statistical method determining all probable values and possibilities that a random variable can deliver from a particular range.

2. **Random Variable:** A variable whose possible values are outcomes of a random phenomenon.

3. **Discrete Distributions:** Probability distributions for variables with a finite or countably infinite set of possible values.

4. **Continuous Distributions:** Probability distributions for variables with an uncountably infinite set of possible values.

5. **Normal Distribution:** A symmetric probability distribution that is frequently occurring around its mean value, representing various natural events.

**9) Reading: "Types of Probability Distributions Every Data Science Expert Should know"**

**Types of Probability Distributions Every Data Science Expert Should know**

Data Science has become one of the most popular interdisciplinary fields. It uses scientific approaches, methods, algorithms, and operations to obtain facts and insights from unstructured, semi-structured, and structured datasets. Organizations use these collected facts and insights for efficient production, business growth, and to predict user requirements. Probability distribution plays a significant role in performing data analysis equipping a dataset for training a model. In this article, you will learn about the types of Probability Distribution, random variables, types of discrete distributions, and continuous distribution. To know more about linear discriminant analysis in machine learning, click here.

**What is Probability Distribution?**

A Probability Distribution is a statistical method that determines all the probable values and possibilities that a random variable can deliver from a particular range. This range of values will have a lower bound and an upper bound, which we call the minimum and the maximum possible values.

Various factors on which plotting of a value depends are standard deviation, mean (or average), skewness, and kurtosis. All of these play a significant role in Data science as well. We can use probability distribution in physics, engineering, finance, data analysis, machine learning, etc.

Get to know more about measures of dispersion.

**Significance of Probability distributions in Data Science**

In a way, most of the data science and machine learning operations are dependent on several assumptions about the probability of your data. Probability distribution allows a skilled data analyst to recognize and comprehend patterns from large data sets; that is, otherwise, entirely random variables and values. Thus, it makes probability distribution a toolkit based on which we can summarize a large data set. The density function and distribution techniques can also help in plotting data, thus supporting data analysts to visualize data and extract meaning.

**General Properties of Probability Distributions**

Probability distribution determines the likelihood of any outcome. The mathematical expression takes a specific value of x and shows the possibility of a random variable with p(x). Some general properties of the probability distribution are –

1. The total of all probabilities for any possible value becomes equal to 1.
2. In a probability distribution, the possibility of finding any specific value or a range of values must lie between 0 and 1.
3. Probability distributions tell us the dispersal of the values from the random variable. Consequently, the type of variable also helps determine the type of probability distribution.

**Common Data Types**

Before jumping directly into explaining the different probability distributions, let us first understand the different types of probability distributions or the main categories of the probability distribution. Data analysts and data engineers have to deal with a broad spectrum of data, such as text, numerical, image, audio, voice, and many more. Each of these have a specific means to be represented and analyzed. Data in a probability distribution can either be discrete or continuous. Numerical data especially takes one of the two forms.

**Discrete data:** They take specific values where the outcome of the data remains fixed. Like, for example, the consequence of rolling two dice or the number of overs in a T-20 match. In the first case, the result lies between 2 and 12. In the second case, the event will be less than 20. Different types of discrete distributions that use discrete data are:

* Binomial Distribution
* Hypergeometric Distribution
* Geometric Distribution
* Poisson Distribution
* Negative Binomial Distribution
* Multinomial Distribution

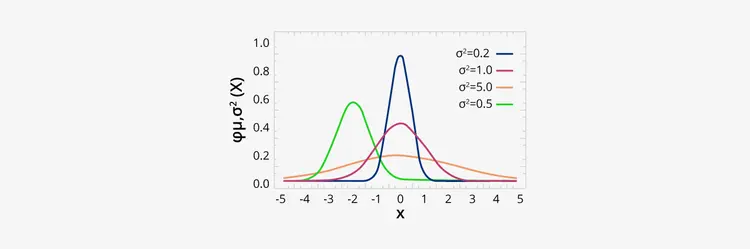
**Continuous data:** It can obtain any value irrespective of bound or limit. Example: weight, height, any trigonometric value, age, etc. Different types of continuous distributions that use continuous data are:

* Beta distribution
* Cauchy distribution
* Exponential distribution
* Gamma distribution
* Logistic distribution
* Weibull distribution

**Types of Probability Distribution explained**

Here are some of the popular types of Probability distributions used by data science professionals. (Try all the code using Jupyter Notebook)

**Normal Distribution:** It is also known as Gaussian distribution. It is one of the simplest types of continuous distribution. This probability distribution is symmetrical around its mean value. It also shows that data at close proximity of the mean is frequently occurring, compared to data that is away from it. Here, mean = 0, variance = finite value



Here, you can see 0 at the center is the Normal Distribution for different mean and variance values.

Here is a code example showing the use of Normal Distribution:

from scipy.stats import norm

import matplotlib.pyplot as mpl

import numpy as np

def normalDist() -> None:

fig, ax = mpl.subplots(1, 1)

mean, var, skew, kurt = norm.stats(moments = 'mvsk')

x = np.linspace(norm.ppf(0.01), norm.ppf(0.99), 100)

ax.plot(x, norm.pdf(x),

'r-', lw = 5, alpha = 0.6, label = 'norm pdf')

ax.plot(x, norm.cdf(x),

'b-', lw = 5, alpha = 0.6, label = 'norm cdf')

vals = norm.ppf([0.001, 0.5, 0.999])

np.allclose([0.001, 0.5, 0.999], norm.cdf(vals))

r = norm.rvs(size = 1000)

ax.hist(r, normed = True, histtype = 'stepfilled', alpha = 0.2)

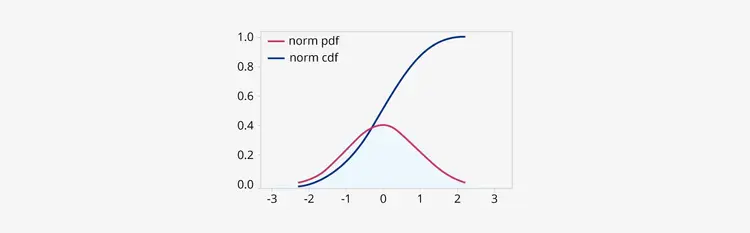
ax.legend(loc = 'best', frameon = False)

mpl.show()

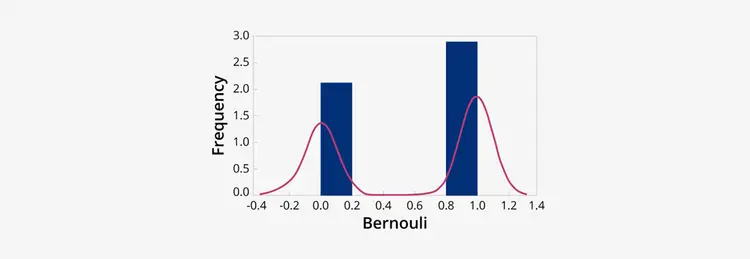
normalDist()

Copy Code

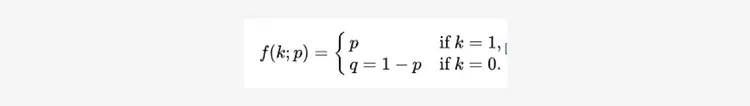
Output:



**Bernoulli Distribution:** It is the simplest type of probability distribution. It is a particular case of Binomial distribution, where n=1. It means a binomial distribution takes 'n' number of trials, where n > 1 whereas, the Bernoulli distribution takes only a single trial.



Probability Mass Function of a Bernoulli’s Distribution is:



where p = probability of success and q = probability of failure

Here is a code example showing the use of Bernoulli Distribution:

from scipy.stats import bernoulli

import seaborn as sb

def bernoulliDist():

data\_bern = bernoulli.rvs(size=1200, p = 0.7)

ax = sb.distplot(

data\_bern,

kde = True,

color = 'g',

hist\_kws = {'alpha' : 1},

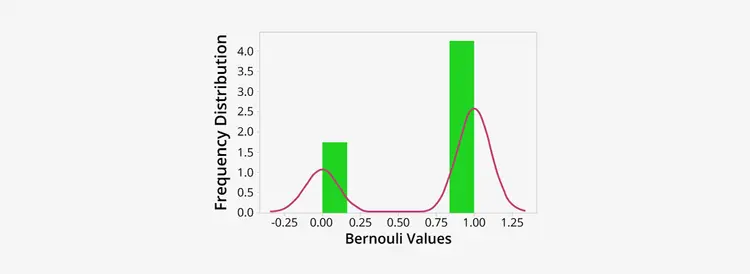
kde\_kws = {'color': 'y', 'lw': 3, 'label': 'KDE'})

ax.set(xlabel = 'Bernouli Values', ylabel = 'Frequency Distribution')

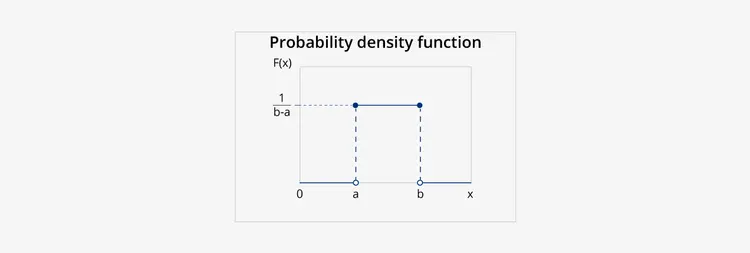
bernoulliDist()

Copy Code

Output:



**Continuous Uniform Distribution:** In this type of continuous distribution, all outcomes are equally possible; each variable gets the same probability of hit as a consequence. This symmetric probabilistic distribution has random variables at an equal interval, with the probability of 1/(b-a).



Here is a code example showing the use of Uniform Distribution:

from numpy import random

import matplotlib.pyplot as mpl

import seaborn as sb

def uniformDist():

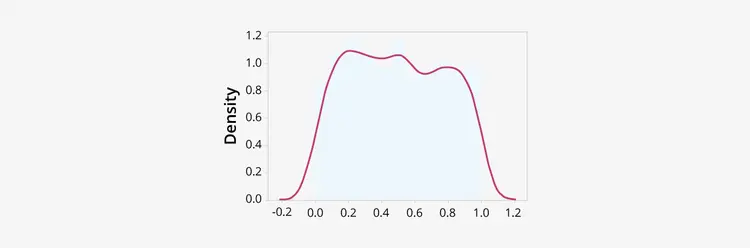
sb.distplot(random.uniform(size = 1200), hist = True)

mpl.show()

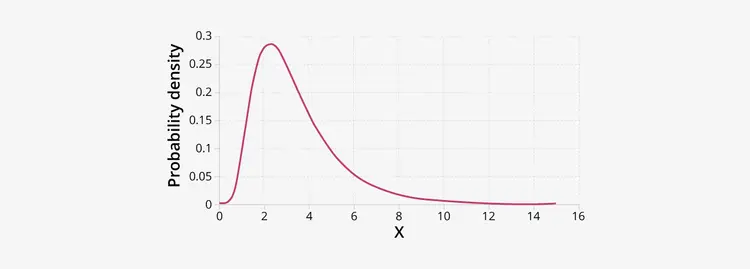
uniformDist()

Copy Code

Output:



**Log-Normal Distribution:** A Log-Normal distribution is another type of continuous distribution of logarithmic values that form a normal distribution. We can transform a log-normal distribution into a normal distribution.



Here is a code example showing the use of Log-Normal Distribution

import matplotlib.pyplot as mpl

def lognormalDist():

muu, sig = 3, 1

s = np.random.lognormal(muu, sig, 1000)

cnt, bins, ignored = mpl.hist(s, 80, normed = True, align ='mid', color = 'y')

x = np.linspace(min(bins), max(bins), 10000)

calc = (np.exp( -(np.log(x) - muu) \*\*2 / (2 \* sig\*\*2))

/ (x \* sig \* np.sqrt(2 \* np.pi)))

mpl.plot(x, calc, linewidth = 2.5, color = 'g')

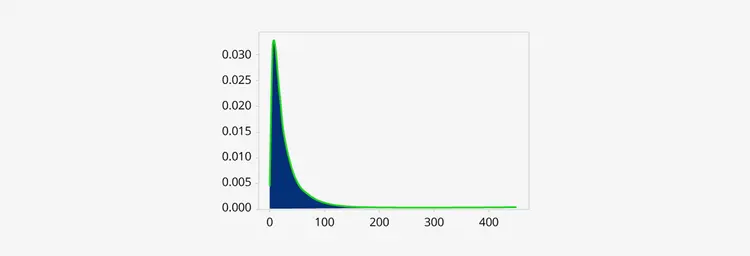
mpl.axis('tight')

mpl.show()

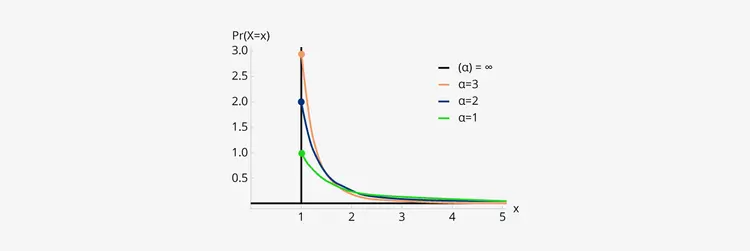
lognormalDist()

Copy Code

Output:



**Pareto Distribution:** It is one of the most critical types of continuous distribution. The Pareto Distribution is a skewed statistical distribution that uses power-law to describe quality control, scientific, social, geophysical, actuarial, and many other types of observable phenomena. The distribution shows slow or heavy-decaying tails in the plot, where much of the data reside at its extreme end.



Here is a code example showing the use of Pareto Distribution –

import numpy as np

from matplotlib import pyplot as plt

from scipy.stats import pareto

def paretoDist():

xm = 1.5

alp = [2, 4, 6]

x = np.linspace(0, 4, 800)

output = np.array([pareto.pdf(x, scale = xm, b = a) for a in alp])

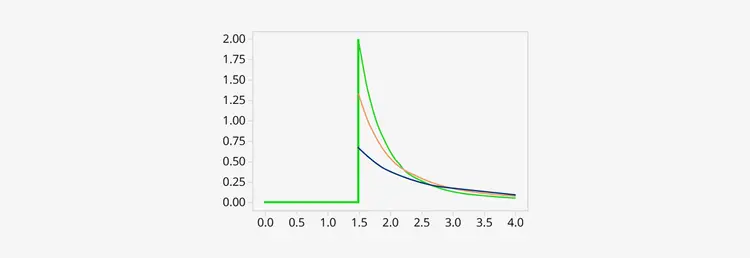
plt.plot(x, output.T)

plt.show()

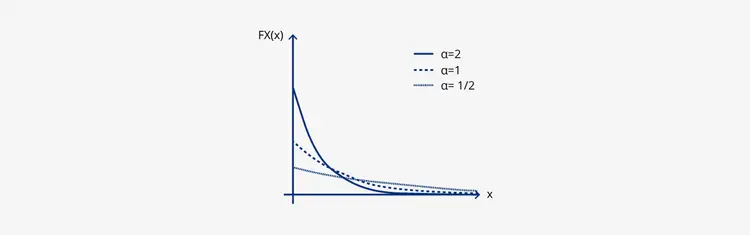
paretoDist()

Copy Code

Output:



**Exponential Distribution:** It is a type of continuous distribution that determines the time elapsed between events (in a Poisson process). Let’s suppose, that you have the Poisson distribution model that holds the number of events happening in a given period. We can model the time between each birth using an exponential distribution.



Here is a code example showing the use of Pareto Distribution –

from numpy import random

import matplotlib.pyplot as mpl

import seaborn as sb

def expDist():

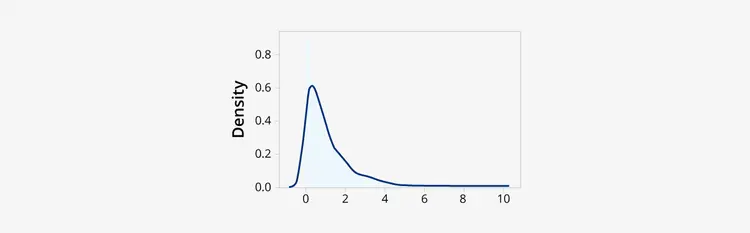
sb.distplot(random.exponential(size = 1200), hist = True)

mpl.show()

expDist()

Copy Code

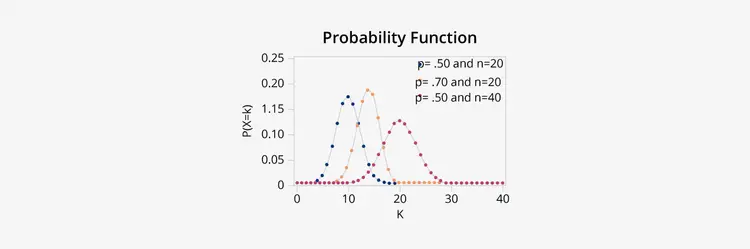
Output:



**Types of the Discrete probability distribution –**

There are various types of Discrete Probability Distribution a Data science aspirant should know about. Some of them are –

**Binomial Distribution:** It is one of the popular discrete distributions that determine the probability of x success in the 'n' trial. We can use Binomial distribution in situations where we want to extract the probability of SUCCESS or FAILURE from an experiment or survey which went through multiple repetitions. A Binomial distribution holds a fixed number of trials. Also, a binomial event should be independent, and the probability of obtaining failure or success should remain the same.



Here is a code example showing the use of Binomial Distribution –

from numpy import random

import matplotlib.pyplot as mpl

import seaborn as sb

def binomialDist():

sb.distplot(random.normal(loc = 50, scale = 6, size = 1200), hist = False, label = 'normal')

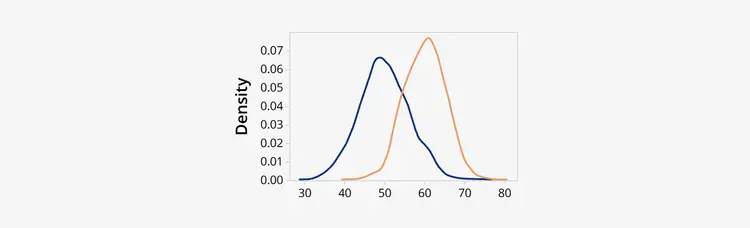
sb.distplot(random.binomial(n = 100, p = 0.6, size = 1200), hist = False, label = 'binomial')

plt.show()

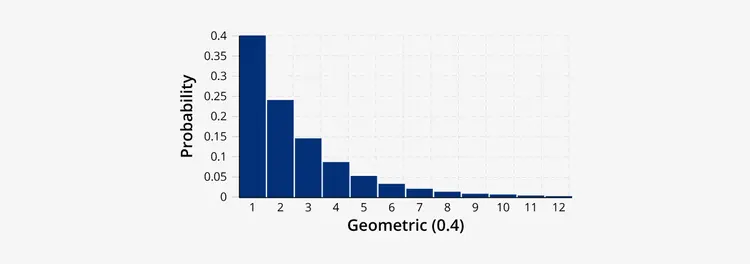
binomialDist()

Copy Code

Output:



Geometric Distribution: The geometric probability distribution is one of the crucial types of continuous distributions that determine the probability of any event having likelihood ‘p’ and will happen (occur) after 'n' number of Bernoulli trials. Here 'n' is a discrete random variable. In this distribution, the experiment goes on until we encounter either a success or a failure. The experiment does not depend on the number of trials.



Here is a code example showing the use of Geometric Distribution –

import matplotlib.pyplot as mpl

def probability\_to\_occur\_at(attempt, probability):

return (1-p)\*\*(attempt - 1) \* probability

p = 0.3

attempt = 4

attempts\_to\_show = range(21)[1:]

print('Possibility that this event will occur on the 7th try: ', probability\_to\_occur\_at(attempt, p))

mpl.xlabel('Number of Trials')

mpl.ylabel('Probability of the Event')

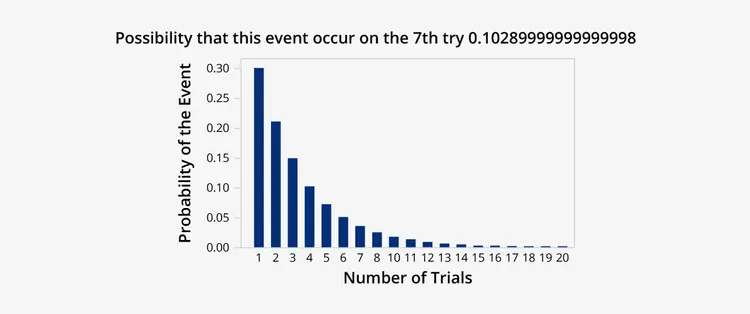
barlist = mpl.bar(attempts\_to\_show, height=[probability\_to\_occur\_at(x, p) for x in attempts\_to\_show], tick\_label=attempts\_to\_show)

barlist[attempt].set\_color('g')

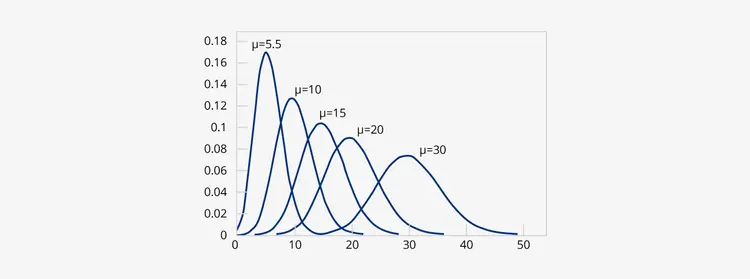
mpl.show()

Copy Code

Output:



Poisson Distribution: Poisson distribution is one of the popular types of discrete distribution that shows how many times an event has the possibility of occurrence in a specific set of time. We can obtain this by limiting the Bernoulli distribution from 0 to infinity. Data analysts often use the Poisson distributions to comprehend independent events occurring at a steady rate in a given time interval.



Here is a code example showing the use of Poisson Distribution

from scipy.stats import poisson

import seaborn as sb

import numpy as np

import matplotlib.pyplot as mpl

def poissonDist():

mpl.figure(figsize = (10, 10))

data\_binom = poisson.rvs(mu = 3, size = 5000)

ax = sb.distplot(data\_binom, kde=True, color = 'g',

bins=np.arange(data\_binom.min(), data\_binom.max() + 1),

kde\_kws={'color': 'y', 'lw': 4, 'label': 'KDE'})

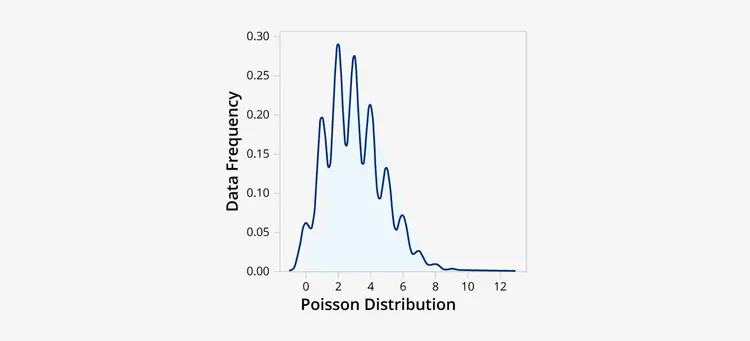
ax.set(xlabel = 'Poisson Distribution', ylabel='Data Frequency')

mpl.show()

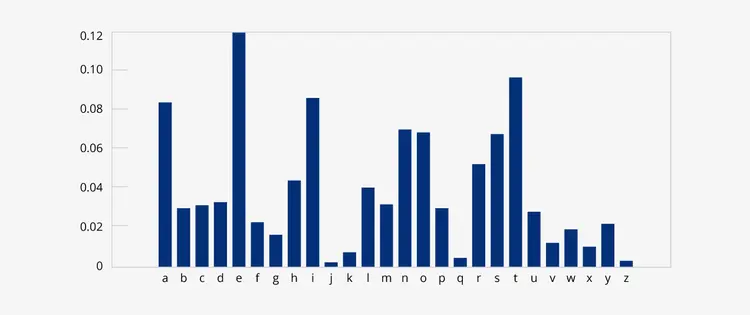
poissonDist()

Copy Code

Output:



Multinomial Distribution: A multinomial distribution is another popular type of discrete probability distribution that calculates the outcome of an event having two or more variables. The term multi means more than one. The Binomial distribution is a particular type of multinomial distribution with two possible outcomes - true/false or heads/tails.



Here is a code example showing the use of Multinomial Distribution –

import numpy as np

import matplotlib.pyplot as mpl

np.random.seed(99)

n = 12

pvalue = [0.3, 0.46, 0.22]

s = []

p = []

for size in np.logspace(2, 3):

outcomes = np.random.multinomial(n, pvalue, size=int(size))

prob = sum((outcomes[:,0] == 7) & (outcomes[:,1] == 2) & (outcomes[:,2] == 3))/len(outcomes)

p.append(prob)

s.append(int(size))

fig1 = mpl.figure()

mpl.plot(s, p, 'o-')

mpl.plot(s, [0.0248]\*len(s), '--r')

mpl.grid()

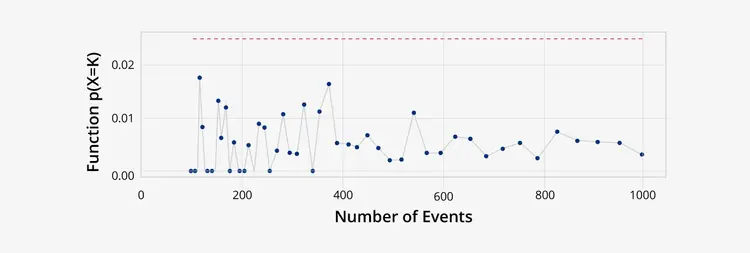
mpl.xlim(xmin = 0)

mpl.xlabel('Number of Events')

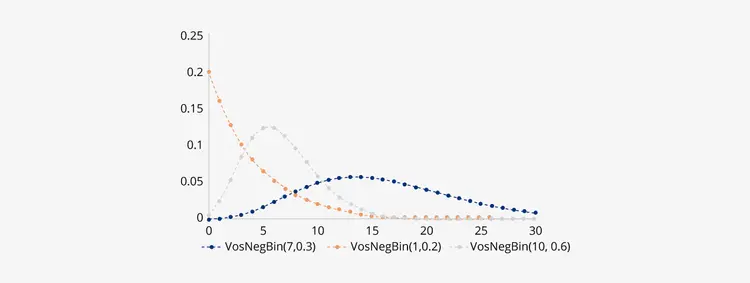
mpl.ylabel('Function p(X = K)')

Copy Code

Output:



**Negative Binomial Distribution:** It is also a type of discrete probability distribution for random variables having negative binomial events. It is also known as the Pascal distribution, where the random variable tells us the number of repeated trials produced during a specific number of experiments.



Here is a code example showing the use of Negative Binomial Distribution –

import matplotlib.pyplot as mpl

import numpy as np

from scipy.stats import nbinom

x = np.linspace(0, 6, 70)

gr, kr = 0.3, 0.7

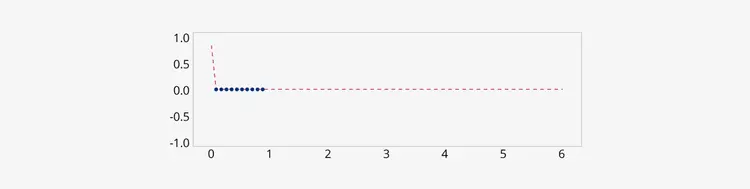
g = nbinom.ppf(x, gr, kr)

s = nbinom.pmf(x, gr, kr)

mpl.plot(x, g, "\*", x, s, "r--")

Copy Code

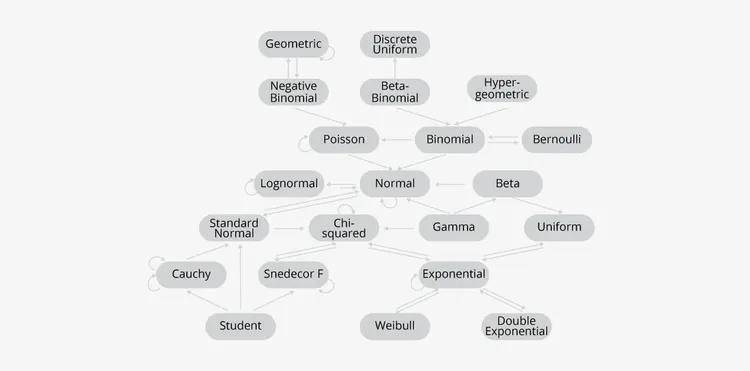
Output:



Apart from these mentioned distribution types, various other types of probability distributions exist that data science professionals can use to extract reliable datasets. In the next topic, we will understand some interconnections & relationships between various types of probability distributions.

**Relationship between various Probability distributions –**

It is surprising to see that different types of probability distributions are interconnected. In the chart shown below, the dashed line is for limited connections between two families of distribution, whereas the solid lines show the exact relationship between them in terms of transformation, variable, type, etc.



**Conclusion**

Probability distributions are prevalent among data analysts and data science professionals because of their wide usage. Today, companies and enterprises hire data science professionals in many sectors, namely, computer science, health, insurance, engineering, and even social science, where probability distributions appear as fundamental tools for application. It is essential for Data analysts and data scientists. to know the core of statistics. Probability Distributions perform a requisite role in analyzing data and cooking a dataset to train the algorithms efficiently. If you want to learn more about data science - particularly probability distributions and their uses, check out KnowledgeHut's comprehensive Data science course.

**10) Matching heading activity about the previous reading.**

Matching Heading Activity:

Column A: Headings

1. Common Data Types in Probability Distributions

2. Understanding Probability Distributions

3. Significance of Probability Distributions

4. Exploring Specific Probability Distributions

5. Types of Data Representations

Column B:

a. Overview of different types of probability distributions used in data science

b. Recognizing the importance of probability distributions in data science

c. Introduction to the general properties of probability distributions

d. Exploring various types of discrete and continuous data representations

e. In-depth exploration of popular probability distributions and their applications