Week 9 : Lab

Importing Library: Numpy

(From Last week's Lab, you can find more information about **"NumPy"** library).

NumPy is the fundamental Python library for numerical computing.

[ ]

import numpy as np

Importing Library: SciPy

(From Last week's Lab, you can find more information about **"SciPy"** library).

**SciPy** stands for Scientific python. For some statistical calculation we do need Scipy e.g, during last week, you have used stats from Scipy to calculate "Mode".

[ ]

from scipy import stats

Importing Library: Math

**Math** is a library you can apply for performing mathematical tasks by using different Math methods in this library.

[ ]

import math as mt

Importing Library: Random

Python has a built-in library **random** which can be used for generating random numbers/values

[ ]

import random as rnd

Generating Binomial Distribution:

We will now see how we can generate a Binomial Distribution and we will find the PMF, CMF and other statistical measures using **Stats** function from **SciPy** library.

Following is an example code given, for function binom(n=10,p=0.2), where, n is the number of trials and p holds the probability of success.

from scipy import stats

X = stats.binom(40, 0.5) # Declare X to be a binomial random variable

print(X.pmf(3)) # P(X = 3) which implies we are calculating the mass function for the third trial

print(X.cdf(4)) # P(X <= 4) which implies we are calculating cumulative distribution upto fourth trial

print(X.mean()) # Mean

print(X.var()) # Variance

print(X.std()) # Standard Deviation

Question:1

You are given a dice which you can roll five times to check how many times you find "6" and probability of the roll being a success is 80%. Write a code to find the probability mass function cumulative distribution function for the final trial.

[ ]

Generating Geometric Distribution:

Geometric probability distribution is a discrete probability distribution. It represents the probability that an event having probability p will happen (success) after X number of Bernoulli trials with X taking values of 1, 2, 3, …k Lets understand the concept in a more descriptive manner using basketball free throws shot example. In basketball, free throws or foul shots are unopposed attempts to score points by shooting from behind the free throw line. The goal is to find the probability that the shooter will have the first perfect throw in X number of shoots.Let’s calculate the probability of X = 1, 2, 3, 4, 5 number of throws for first successful throw. Given the probability of a perfect throw (success) is 0.6 and, thus, the probability of unsuccessful throw (failure) is 0.4 (1-0.6). The example code is given below

from scipy.stats import geom

import matplotlib.pyplot as plt

#

# X = Discrete random variable representing number of throws

# p = Probability of the perfect throw

#

X =

p = 0.04

#

# Calculate geometric probability distribution

#

geom\_pd = geom.pmf(X, p)

#

# Plot the probability distribution

#

fig, ax = plt.subplots(1, 1, figsize=(8, 6))

ax.plot(X, geom\_pd, 'bo', ms=8, label='geom pmf')

plt.ylabel("Probability", fontsize="18")

plt.xlabel("X - No. of Throws", fontsize="18")

plt.title("Geometric Distribution - No. of Throws Vs Probability", fontsize="18")

ax.vlines(X, 0, geom\_pd, colors='b', lw=5, alpha=0.5)

# Question 2

You ask people outside a polling station who they voted for until you find someone that voted for the independent candidate in a local election. If your probability of success is 0.2, what is the probability you meet an independent voter on your third try? Using the example codes given above, write a code to solve this question