1. A test is conducted which is consisting of 20 MCQs (multiple choices questions) with every MCQ having its four options out of which only one is correct. Determine the probability that a person undertaking that test has answered exactly 5 questions wrong.

**Code:**

import pandas as pd

import numpy as np

import scipy as sci

import matplotlib.pyplot as plt

import math

from scipy.stats import binom

*# Define the variable*

num\_of\_question = 20

*# probability of wrong answer of a single question*

prob\_wrong\_answer = 3/4 *# as one of the 4 question is right*

print('Probability of wrong answer of a single question:\t', prob\_wrong\_answer)

print('\nProbability of 5 wrong answers')

print('----------------------------')

print('\nSolution -1 - Mathemetical Model')

print('\t Using Formula: 20C5 \* power((1-prob\_wrong\_answer),15) \* power(prob\_wrong\_answer, 5)')

p\_five\_wrong\_answer = (math.factorial(20)/(math.factorial(15)\*math.factorial(5))) \\* math.pow((1-prob\_wrong\_answer),15) \* math.pow(prob\_wrong\_answer,5)

print('\nProbability of 5 wrong answers',p\_five\_wrong\_answer)

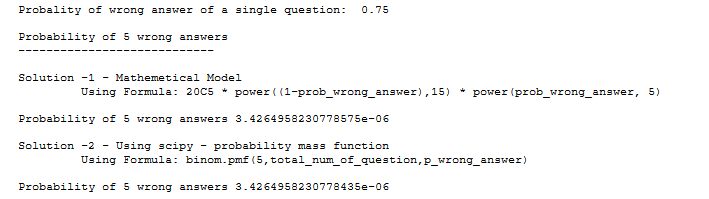
print('\nSolution -2 - Using scipy - probability mass function')

print('\t Using Formula: binom.pmf(5,total\_num\_of\_question,p\_wrong\_answer)')

p\_five\_wrong\_answer=binom.pmf(5,num\_of\_question,prob\_wrong\_answer)

print('\nProbability of 5 wrong answers',p\_five\_wrong\_answer)

**Output:**



1. A die marked A to E is rolled 50 times. Find the probability of getting a “D” exactly 5 times.

**Code:**

import pandas as pd

import numpy as np

import scipy as sci

import matplotlib.pyplot as plt

import math

from scipy.stats import binom

*# Define the variable*

num\_of\_roll = 50

*# probability of getting D in single roll*

prob\_D = 1/5 *# as in a roll, D has equal chance among A to E*

print('Probability of getting D in single roll:\t', prob\_D)

print('\nProbability of getting D exactly 5 times')

print('--------------------------------------')

print('\nSolution -1 - Mathemetical Model')

print('\t Using Formula: 50C5 \* power((1-prob\_D),45) \* power(prob\_D, 5)')

p\_five\_D = (math.factorial(50)/(math.factorial(45)\*math.factorial(5))) \\* math.pow((1-prob\_D),45) \* math.pow(prob\_D,5)

print('\nProbability of getting D exactly 5 times',p\_five\_D)

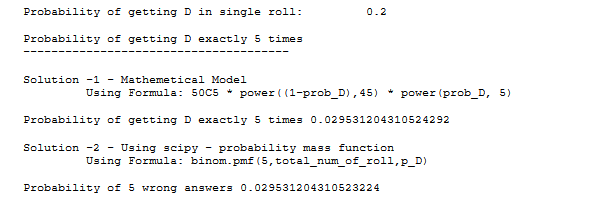
print('\nSolution -2 - Using scipy - probability mass function')

print('\t Using Formula: binom.pmf(5,total\_num\_of\_roll,p\_D)')

p\_five\_D=binom.pmf(5,num\_of\_roll,prob\_D)

print('\nProbability of 5 wrong answers',p\_five\_D)

**Output:**



1. Two balls are drawn at random in succession without replacement from an urn containing 4 red balls and 6 black balls. Find the probabilities of all the possible outcomes.

**Code:**

import pandas as pd

import numpy as np

import scipy as sci

import matplotlib.pyplot as plt

import math

from scipy.stats import binom

*# Define the variable*

num\_of\_balls = 10

num\_of\_red\_balls = 4

num\_of\_black\_balls = 6

*# Two balls are drawn at random in succession without replacement*

*# If Red ball is denoted by R and Balck ball is denoted by B then*

*# Possible outcome - RR, RB, BR, BB*

*# probability of 1st ball red = 4/10*

*# probability of 2nd ball red = 3/9 [when first ball is red]*

*# probability of 2nd ball red = 4/9 [when first ball is black]*

*# probability of 1st ball black = 6/10*

*# probability of 2nd ball black = 5/9 [when first ball is black]*

*# probability of 2nd ball black = 6/9 [when first ball is red]*

probablity\_RR = (4/10) \* (3/9)

probablity\_RB = (4/10) \* (6/9)

probablity\_BR = (6/10) \* (4/9)

probablity\_BB = (6/10) \* (5/9)

*# Create a DF with the Probability distribution and random variable*

lst\_color=['RR','RB','BR','BB']

df\_probability=pd.DataFrame({'Color':lst\_color, 'Probability':[probability\_RR,probability\_RB,probability\_BR,probability\_BB]})

print(df\_probability)

*# Plot the Probabalities distributions*

plt.bar(df\_probability.Color,df\_probability.Probability,width=.3)

plt.xlabel('Color of the Balls')

plt.xticks(lst\_color)

plt.ylabel('Probability')

plt.title('\nProbabalities distribution Plot\n')

plt.show()

**Output:**

