In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import binom
from scipy.special import factorial
```

In [2]:
```
# A test is conducted which is consisting of 20 MCQs (multiple choices questions)
```

In [4]:
```
# Let us assume that 'n' is representing the number of trails attempted and 'k' is
# in those 'n' trails.
# This implies the number of failures clearly will be 'n-k'

n = 20
# n-k = 5
failures = 5
k = 15
# The probability of success = probability of giving a right answer = p_s
p_s = failures / n

# The probability of failure = probability of giving a wrong answer p_f = 1 - p_s
p_f = 1 - p_s

# When we substitute these values in the formula for Binomial Distribution we get,
# P (exactly 5 out of 20 answers incorrect) = c (20, 5) * (1/4)**15 * (3/4)**5

P = (factorial(n) / (factorial(k) * factorial(n - k))) * np.power(p_s,k) * np.powe

print("Probability of exactly 5 out of 20 answers incorrect is {:0.7f}".format(P))
```

```
Probability of exactly 5 out of 20 answers incorrect is 0.0000034
```

In [5]:
```
# A die marked A to E is rolled 50 times. Find the probability of getting a “D” ex
```

In [6]:
```
# Let us assume that in an experiment done, ‘n’ is representing the number of tria
#and that ‘k’ is the count of successes that is to be attained in those ‘n’ trials
#This implies that number of failures clearly will be ‘n - k’.

# Assuming 's' to be the probability of succeeding in a trial, we get that the pro

n = 50
k = 5
failures = n - k

# The probability of success = probaility of getting a "D"
p_s = 1/k

# Hence, the probability of failure = probability of not getting a "D"
p_f = 1 - p_s

print("Probability of getting exactly D in {} throw out of {} number of trails con
```

```
Probability of getting exactly D in 5 throw out of 50 number of trails conduct
ed is 0.2
```

In [7]:
```
sion without replacement from an urn containing 4 red balls and 6 black balls. Find
```

In [9]:
```
red_ball = 4
black_ball = 6
total_balls = red_ball + black_ball

# Sample of possible outcomes = {R_ballB_ball,B_ballR_ball,B_ballB_Ball,R_ballR_ba

# So the chances of picking a red ball first is 4 out of 10 balls. So the probabil
p_first_red_ball = red_ball / total_balls

# The chances of picking a black ball second is 6 out of 9 balls. So the probabili
p_second_black_ball = black_ball / (total_balls - 1)

# The chances of picking a black ball first is 6 out of 10. So the probability is

p_first_black_ball = black_ball / total_balls

# The chances of picking a red ball second is 4 out of 9 balls. So the probability

p_second_red_ball = red_ball / (total_balls - 1)

# Probability of both the balls are red is computed here as:
# Total number of ways to select 2 red balls from 4 red balls / Total number of wa

P_2_r  = (factorial(red_ball) / (factorial(2) * factorial(red_ball - 2))) / (facto

# Probability of both the balls are black is computed as :
# Total number of ways to select 2 black balls from 6 red balls / Total number of
# from 10 total balls.

P_2_b  = (factorial(black_ball) / (factorial(2) * factorial(black_ball -2))) / (fa
```

In [10]:
```
print("The probabilities of all possible outcomes of picking Red ball first, Black
```

```
The probabilities of all possible outcomes of picking Red ball first, Black ba
ll second is (0.40, 0.67)
```

In [11]:
```
print("The probabilities of all possible outcomes of picking Black ball first, Red
```

```
The probabilities of all possible outcomes of picking Black ball first, Red ba
ll second is (0.60, 0.44)
```

In [12]:
```
print("The probabilities of all possible outcomes of picking both the balls as bla
```

```
The probabilities of all possible outcomes of picking both the balls as black
is 0.0000000002
```

In [13]:
```
print("The probabilities of all possible outcomes of picking both the balls as red
```

```
The probabilities of all possible outcomes of picking both the balls as red is
0.0000000001
```

In [ ]: