Manuel Beltran

Oct. 16, 2019

EE 381

Project 3

Problem 1

Introduction

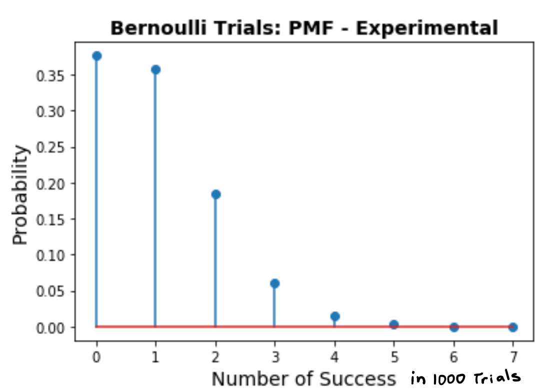
There is 3 unfair dice. A trial is conducted at an attempt to see a successful roll of dice1 being 1, dice2 being 2, and dice3 being 3. 1000 trials are run to see the amount of successes one can achieve. This process is repeated 10,000 times to be able to figure out the probability of the certain amount of successes.

Methodology

All 3 dice use a function to simulate 1000 trials. The resulted 3 arrays are then run through to check for the success condition and record the total amount of successes. This process is redone 10,000 times with each total amount of successes saved to an array. The probability of having a certain amount of successes is calculated and graphed.

Results and Conclusion

The results below show the probability of having x amount of successes within 1000 trials.



An Appendix

import numpy as np

import matplotlib.pyplot as plt

p0 = (.1, .1, .1, .3, .2, .2)

```
c = [1,2,3,4,5,6]
trials = 1000;
N = 10000
s = np.zeros((N, 1))
largest = 0
for i in range(N):
    success = 0
    d1 = np.random.choice(c,trials,p=p0)
    d2 = np.random.choice(c,trials,p=p0)
    d3 = np.random.choice(c,trials,p=p0)
    for x in range(len(d1)):
        if (d1[x] == 1 and d2[x] == 1 and d3[x] == 3):
            success = success + 1
    if success > largest:
        largest = success
    s[i] = success
# Plotting
b = range (0, largest + 2)
sb = np.size(b)
h1, bin_edges = np.histogram(s, bins = b)
b1 = bin_edges[0:sb-1]
plt.close('all')
prob = h1/N

# Plots and labels
plt.stem(b1,prob)
plt.title('Bernoulli Trials: PMF - Experimental', fontsize = 14, fontweight = 'bold')
plt.xlabel('Number of Success', fontsize = 14)
plt.ylabel('Probability', fontsize = 14)
plt.xticks(b1)
```

Problem 2

Introduction

Like problem 1, the same experiment is done but this time the Binomial Formula is used to calculate the probabilities.

Methodology

For the formula:

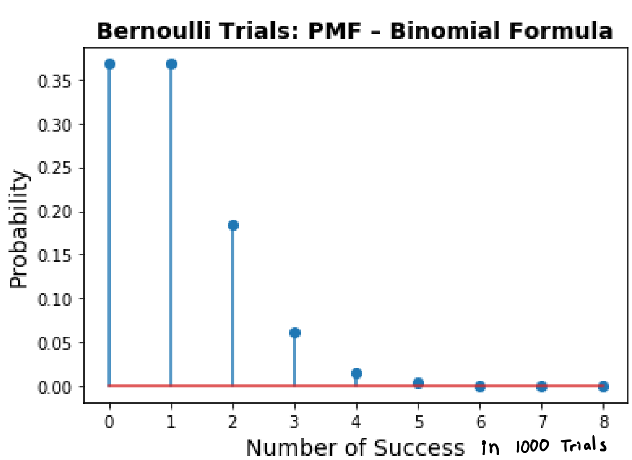
$$p(X = x) = \binom{n}{x} p^x q^{n-x}$$

Since the Binomial Function uses combinations, a simple function is created to calculate combinations. P is .001 and Q is 1-P. x is the amount of success. This is all used to calculate the probability of x successes in n=1000 trials.

A loop is run to use the formula to calculate the probability of 0 successes to x amount of successes. The probability of having a certain amount of successes is graphed on PMF.

Results and Conclusion

The results below show the probability of having x amount of successes within 1000 trials.



An Appendix

```
import numpy as np

import matplotlib.pyplot as plt

import math


def nCr(n,r):

    f = math.factorial

    return f(n) // f(r) // f(n-r)


successes = [0,1,2,3,4,5,6]

trials = 1000;

N = 9

s = np.zeros((N, 1))

p= .001

for x in range(N):

    s[x] = nCr(trials,x) * p**x * (1-p)**(trials-x)


# Plotting

b = range (0, N + 1)

sb = np.size(b)

h1, bin_edges = np.histogram(successes, bins = b)

b1 = bin_edges[0:sb-1]

plt.close('all')

prob = s


# Plots and labels

plt.stem(b1,prob)

plt.title('Bernoulli Trials: PMF – Binomial Formula', fontsize = 14, fontweight = 'bold')

plt.xlabel('Number of Success', fontsize = 14)

plt.ylabel('Probability', fontsize = 14)
```

plt.xticks(b1)

Problem 3

Introduction

Like problem 1, the same experiment is done but this time Poisson Approximation is used to calculate the probabilities.

Methodology

For the formula:

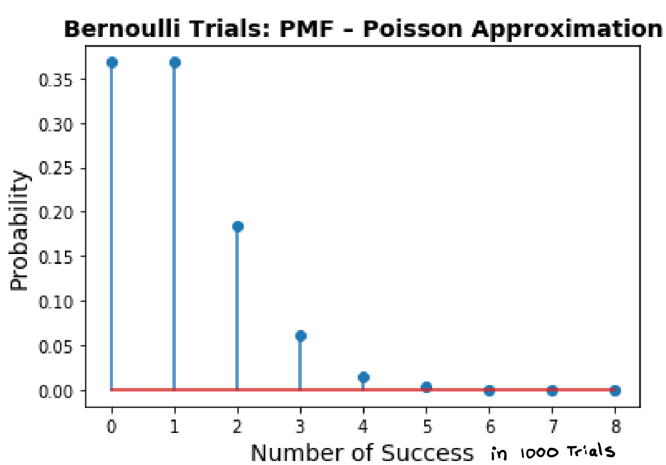
$$p(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

Lambda is P*n. P is .001. x is the amount of success. This is all used to calculate the probability of x successes in 1000 trials.

A loop is run to use the formula to calculate the probability of 0 successes to x amount of successes. The probability of having a certain amount of successes is graphed.

Results and Conclusion

The results below show the probability of having x amount of successes within 1000 trials.



An Appendix

```
import numpy as np
import matplotlib.pyplot as plt
import math

f = math.factorial
successes = [0,1,2,3,4,5,6]
trials = 1000;
N = 9
s = np.zeros((N, 1))
p= .001
Lambda = p * trials
for x in range(N):
    s[x] = Lambda**x * math.exp(-Lambda) / f(x)

# Plotting
b = range (0, N + 1)
sb = np.size(b)
h1, bin_edges = np.histogram(successes, bins = b)
b1 = bin_edges[0:sb-1]
plt.close('all')
prob = s

# Plots and labels
plt.stem(b1,prob)
plt.title('Bernoulli Trials: PMF – Poisson Approximation', fontsize = 14, fontweight = 'bold')
plt.xlabel('Number of Success', fontsize = 14)
plt.ylabel('Probability', fontsize = 14)
plt.xticks(b1)
```