Project 3 - Bernoulli trials; Binomial; Poisson

California State University Long Beach

EE 381

Probability and Statistics

Christopher Masferrer

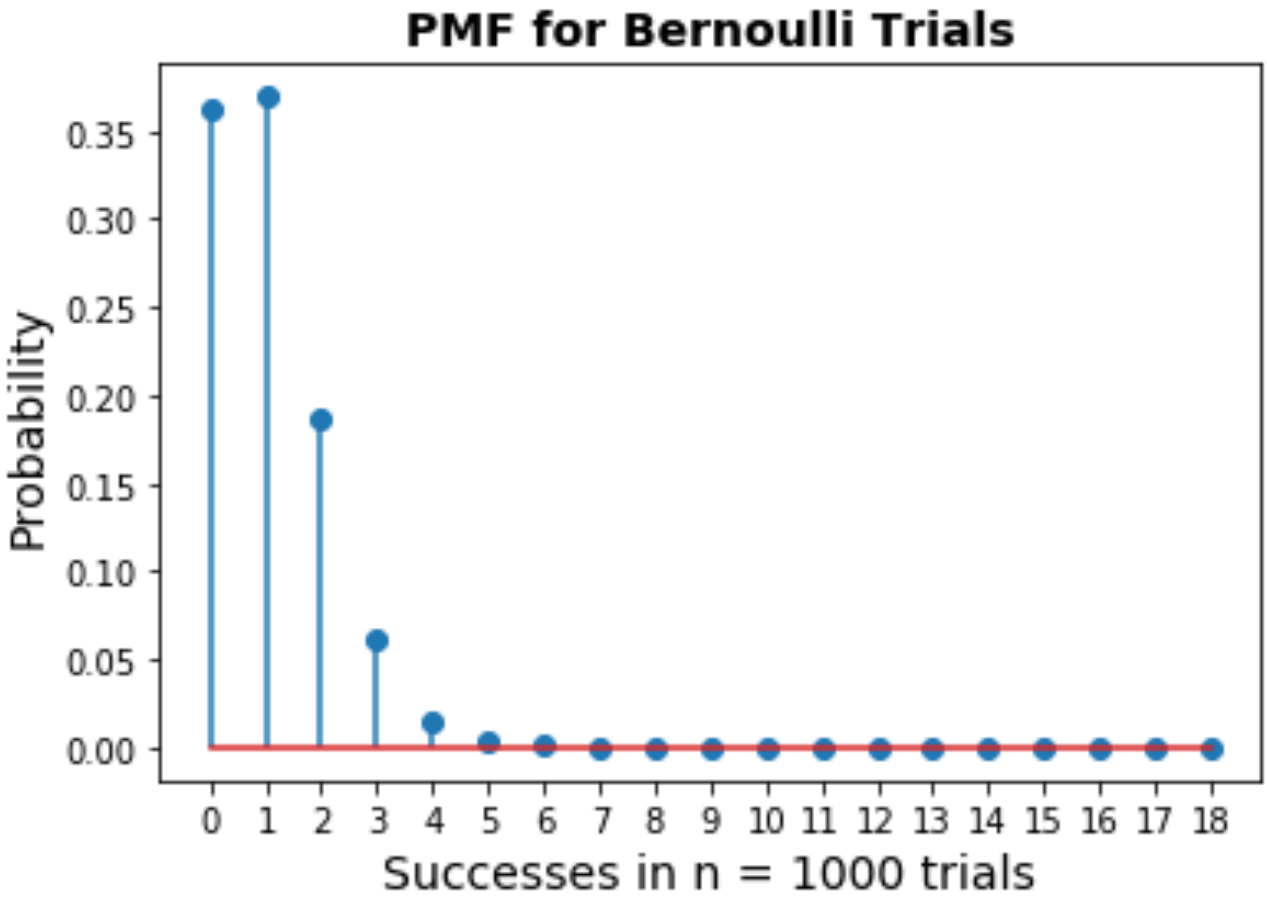
T/TH 5:30 - 6:20

Experiment 1: Bernoulli Trials

Intro: For this experiment I will be rolling 3 n sided dice and counting how many successful runs I have, if die 1 = 1, die 2 = 2, and die 3 = 3. This experiment will be repeated 10,000 times and the results on the successes within 1000 trials will be recorded in X and then displayed in a PMF chart.

Methodology: I used the n sided dice code from the first lab to generate values for the 3 dice. Once the dice were rolled, the valid success were placed in X. Following that, the graph code also from the first lab was used for the PMF plot.

Results:



Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Oct  6 14:57:23 2019

@author: christophermasferrer
"""

#Christopher Masferrer
#EE 381
#Lab 3
import numpy as np
import matplotlib.pyplot as plt

def trials(c, p):
    N = 10000
    n = 1000

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

    for j in range(0, N):
        count = 0
        x1 = np.random.choice(c,n,p=p)
        x2 = np.random.choice(c,n,p=p)
        x3 = np.random.choice(c,n,p=p)
        for k in range (0, n):
            if x1[k] == 1 and x2[k] == 2 and x3[k] == 3:
                count += 1
        X[j] = count

    #Plotting
    b= range(0, 20)
    sb = np.size(b)
    h1, bin_edges = np.histogram(X, bins = b)
    b1 = bin_edges[0 : sb - 1]
    plt.close('all')
    prob = h1/N
    #Plots and labels
    plt.stem(b1, prob, use_line_collection= True)
    plt.title('Bernoulli Trials: PMF - Experimental Results', fontsize = 14, fontweight = 'bold')
    plt.xlabel('Number of successes in n = 1000 trials', fontsize = 14)
    plt.ylabel('Probability', fontsize = 14)
    plt.xticks(b1)
    plt.show()
```

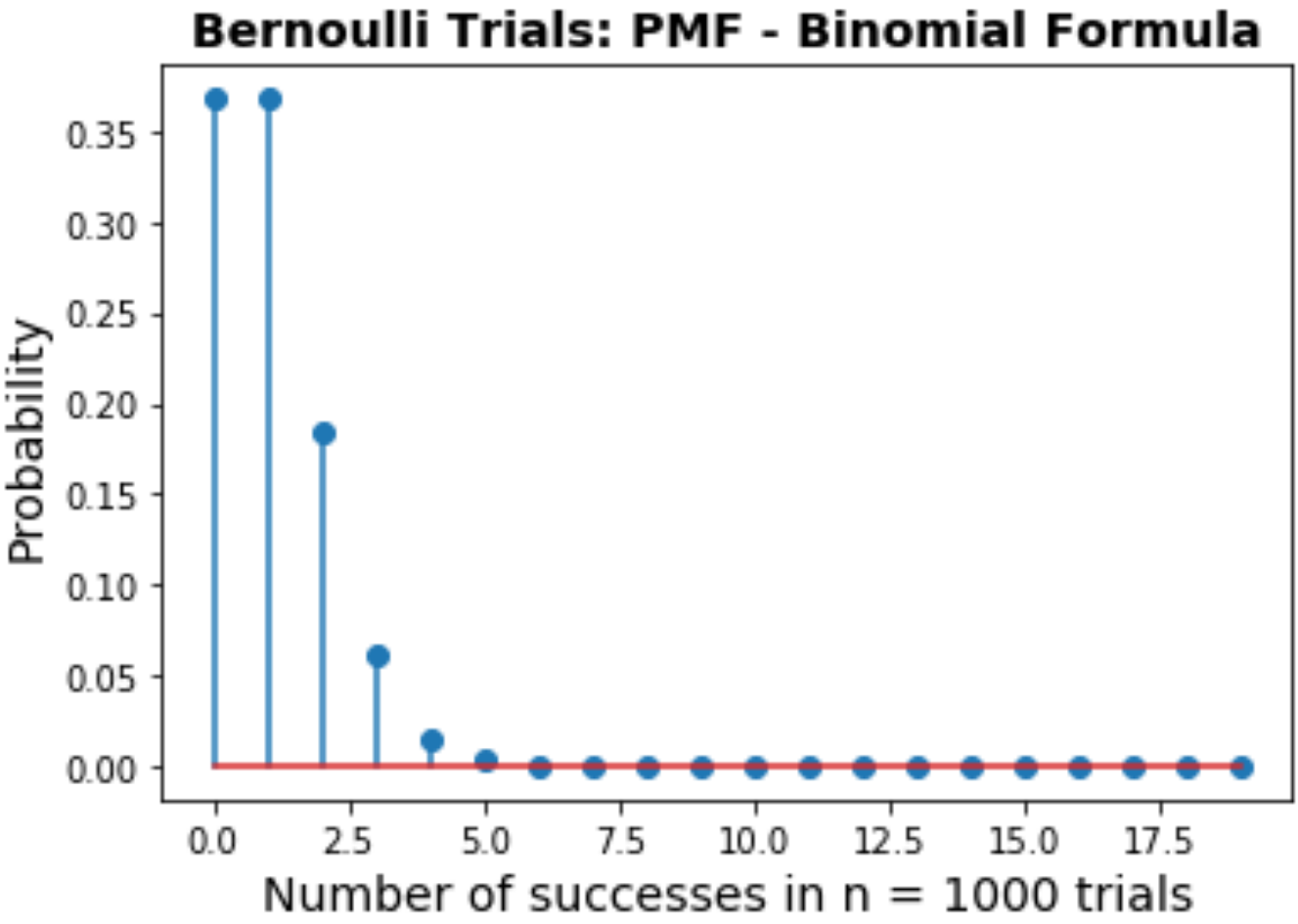
```
c = [1, 2, 3, 4, 5, 6]
p=[0.1, 0.1, 0.1, 0.3, 0.2, 0.2]
trials(c, p)
```

Experiment 2: Binomial distribution

Intro: For this experiment, I will be calculating the binomial distribution of the previous experiment. This experiment will be repeated once and the results on the successes within 1000 trials will be recorded in y and then displayed in a PMF chart.

Methodology: I calculated the values for p and q, as well as assigning the value of 1000 to n. I used the binomial formula and stored it in y, and then displayed the graph below. The Graph is similar to that of the first experiment.

Results:



Code:

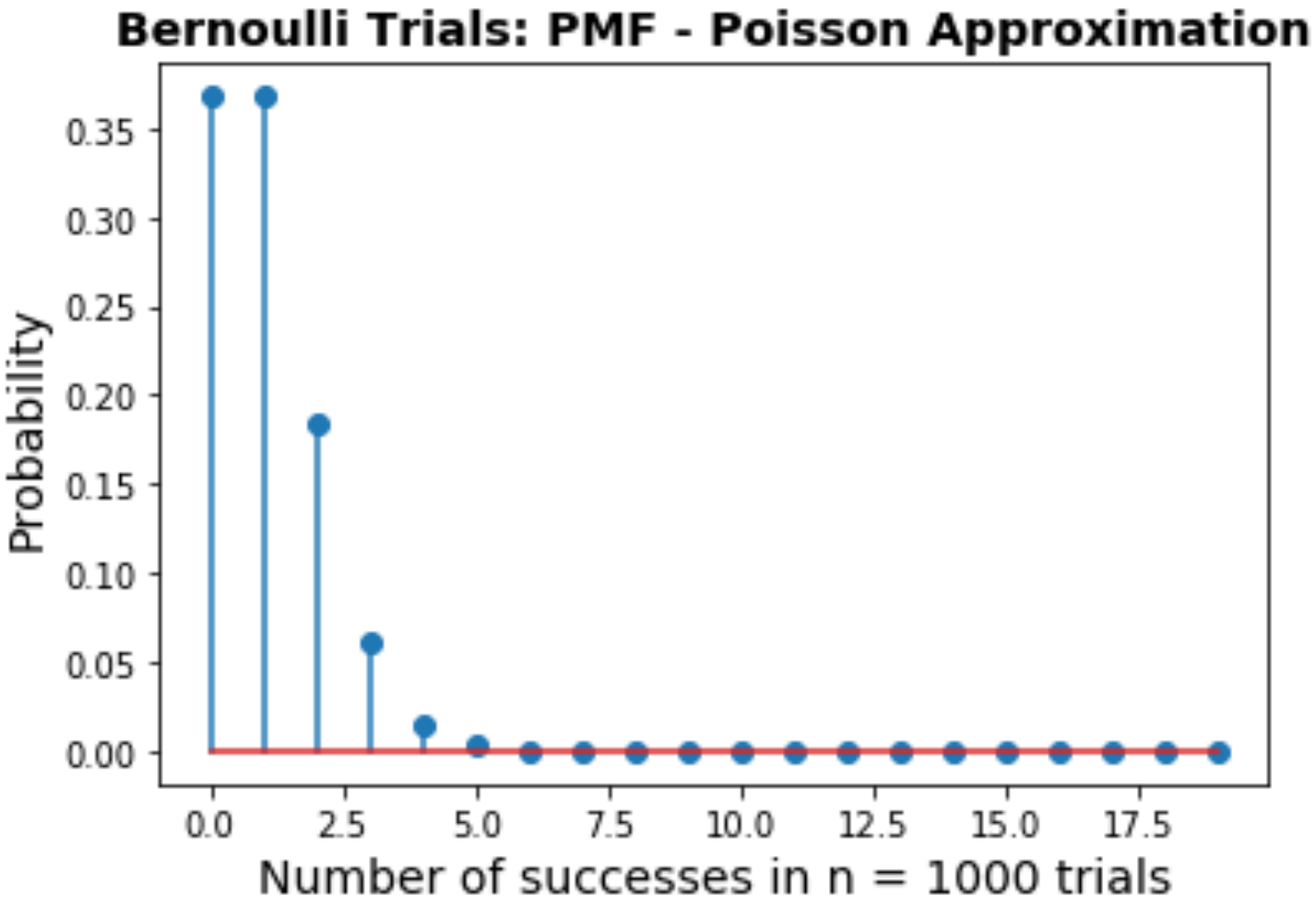
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Oct  7 09:02:08 2019

@author: christophermasferrer
"""

#Christopher Masferrer
#EE 381
#Lab 3
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import binom

def binomial():
    p = 0.1 * 0.1 * 0.1
    q = 1 - p
    n = 1000
    k = np.arange(0,20)
    y = binom.pmf(k, n, p)

    #Plots and labels
    plt.stem(k, y)
    plt.title('Bernoulli Trials: PMF - Binomial Formula', fontsize = 14, fontweight = 'bold')
    plt.xlabel('Number of successes in n = 1000 trials', fontsize = 14)
    plt.ylabel('Probability', fontsize = 14)
    plt.show()


binomial()
```

Experiment 3: Poisson distribution

Intro: For this experiment, I will be calculating the poisson distribution of the first part of the lab. This experiment will be repeated once and the results on the successes within 1000 trials will be recorded in y and then displayed in a PMF chart.

Methodology: I used most of the code from the binomial distribution portion of the lab. Rather than adding in q, I calculated for lamda and then used the poisson formula and recorded the results in y. After, I generated the graph which was once again similar to the graph from the first part of the lab.

Results:



Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct  9 08:51:29 2019

@author: christophermasferrer
"""

#Christopher Masferrer
#EE 381
#Lab 3
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import poisson

def poisson_app():
    n = 1000
    p = 0.1*0.1*0.1
    lamda = n * p
    k = np.arange(0, 20)
    y = poisson.pmf(k, lamda)

    #Plots and labels
    plt.stem(k, y)
    plt.title('Bernoulli Trials: PMF - Poisson Approximation', fontsize = 14, fontweight = 'bold')
    plt.xlabel('Number of successes in n = 1000 trials', fontsize = 14)
    plt.ylabel('Probability', fontsize = 14)
    plt.show()

poisson_app()
```