

# Homework 04: Bayes and DRV's

---

Name: Julia Troni

---

This assignment is due on Canvas by **6:00PM on Friday September 23**. Your solutions to theoretical questions should be done in Markdown directly below the associated question. Your solutions to computational questions should include any specified Python code and results as well as written commentary on your conclusions. Remember that you are encouraged to discuss the problems with your classmates, but **you must write all code and solutions on your own**.

## NOTES:

- Any relevant data sets should be available in the Homework 01 assignment write-up on Canvas. To make life easier on the grader if they need to run your code, do not change the relative path names here. Instead, move the files around on your computer.
- If you're not familiar with typesetting math directly into Markdown then by all means, do your work on paper first and then typeset it later. Remember that there is a [reference guide](#) linked on Canvas on writing math in Markdown. **All** of your written commentary, justifications and mathematical work should be in Markdown.
- Because you can technically evaluate notebook cells in a non-linear order, it's a good idea to do **Kernel → Restart & Run All** as a check before submitting your solutions. That way if we need to run your code you will know that it will work as expected.
- It is **bad form** to make your reader interpret numerical output from your code. If a question asks you to compute some value from the data you should show your code output **AND write a summary of the results** in Markdown directly below your code.
- This probably goes without saying, but... For any question that asks you to calculate something, **you must show all work and justify your answers to receive credit**. Sparse or nonexistent work will receive sparse or nonexistent credit.

---

Import Pandas and NumPy.

In [1]:

```
# Per the standard import pandas as 'pd' and numpy as 'np'
import pandas as pd
import numpy as np

# You may need binom to calculate a binomial coefficient
from scipy.special import binom
```

In case you create a graph, load Matplotlib's Pylab library to set up Jupyter so that it will plot directly in the notebook.

In [2]:

```
import matplotlib.pyplot as plt
%matplotlib inline
# 'inline' puts your graph in the cell versus a new popup window
```

# Problem 1

---

You have become concerned about your diet and decide that at each of your regular three meals per day that you eat, you are going to add a serving with a particular important nutrient. Sometimes you add vitamin C, sometimes you add protein, and sometimes you add iron.

For instance, on a given day, you may eat an additional vitamin C serving for breakfast, an additional protein serving for lunch, and another additional protein serving for dinner.

On a different day you decide to add iron to breakfast, vitamin C to lunch, and protein to dinner. What you choose to add to each meal is a random choice.

Let  $X$  be a random variable that indicates how many servings of protein you added on a given day.

## Part A

---

**(2 points)** What is the probability mass function of  $X$ ?

**solution:** Put your solution to part A here:

$$X = \{0, 1, 2, 3\}$$

$$P(X = a) = p(a)$$

a=	0	1	2	3
P(X=a)=p(a)	8/27	12/27	6/27	1/27

## Part B

---

**(2 points)** What is the cumulative distribution function for  $X$ ?

**solution:**

a=	0	1	2	3
F(a)	8/27	20/27	26/27	27/27 =1

## Part C

---

**(2 points)** What is the probability that you added 2 or fewer servings of protein on a day?

**solution:**

$$P(X \leq 2) = P(X = 0) + P(X = 1) + P(X = 2)$$

$$P(X \leq 2) = \frac{8}{27} + \frac{12}{27} + \frac{6}{27}$$

$$P(X \leq 2) = \frac{26}{27}$$

Equivalently

$$P(X \leq 2) = 1 - P(X = 3) = 1 - \frac{1}{27} = \frac{26}{27}$$

Equivalently, from the CDF we have

$$F(2) = \frac{26}{27}$$

## Part D

---

**(2 points)** What is the probability that you added 2 or more servings of protein?

**solution:**

$$P(X \geq 2) = P(X = 2) + P(X = 3)$$

$$P(X \geq 2) = \frac{6}{27} + \frac{1}{27} = \frac{7}{27}$$

## Problem 2

---

You drink 3 cups of coffee each day, and in a continuation of your plans to change your diet you decide to stop using sugar in your coffee.

However, you find it too difficult to completely stop using sugar. Therefore, 20% of the time you end up using sugar in your coffee anyway.

Let  $X$  be the random variable that indicates the number of 'sugared' coffees out of three that you drink per day.

**(2 points)** What is the probability mass function? Consider 'no sugar' to be success.

**solution:**

a=	0	1	2	3
$P(X=a)=p(a)$	64/125	48/125	12/125	1/125

## Problem 3

---

Consider the probability mass function  $P(X = x) = 2nx^2 - 2nx$  with the discrete random variable taking on values 1, 2, 3, 4.

## Part A

---

**(3 points)** What is the value of  $n$ ?

**solution:**

$$P(X = x) = 2nx^2 - 2nx$$

$$1 = P(X = 1) + P(X = 2) + P(X = 3) + P(X = 4)$$

$$1 = 2n(1^2 - 1) + 2n(2^2 - 2) + 2n(3^2 - 3) + 2n(4^2 - 4)$$

$$1 = 0 + 4n + 12n + 24n$$

$$1 = 40n$$

$$n = \frac{1}{40}$$

## Part B

---

**(2 points)** What is  $P(X = 4)$ ?

**solution:**

$$P(X = 4) = 2 \cdot \frac{1}{40} \cdot (4^2 - 4)$$

$$P(X = 4) = \frac{1}{20} \cdot 12$$

$$P(X = 4) = \frac{3}{5}$$

## Part C

---

**(2 points)** What is  $P(X \leq 3)$ ?

**solution:**

$$P(X \leq 3) = 1 - P(X = 4) = 1 - \frac{3}{5} = \frac{2}{5}$$

# Problem 4

---

One of the following graphs (either graph 1 or graph 2) is a PMF and one is not.

**(2 points)** Draw the CDF for the one valid PMF.

GRAPH 1 

GRAPH 2 

**solution:** Put your solution to #4 here:

- Graph 2 is the correct PMF because the sum of the masses, or the area under the curve equals 1

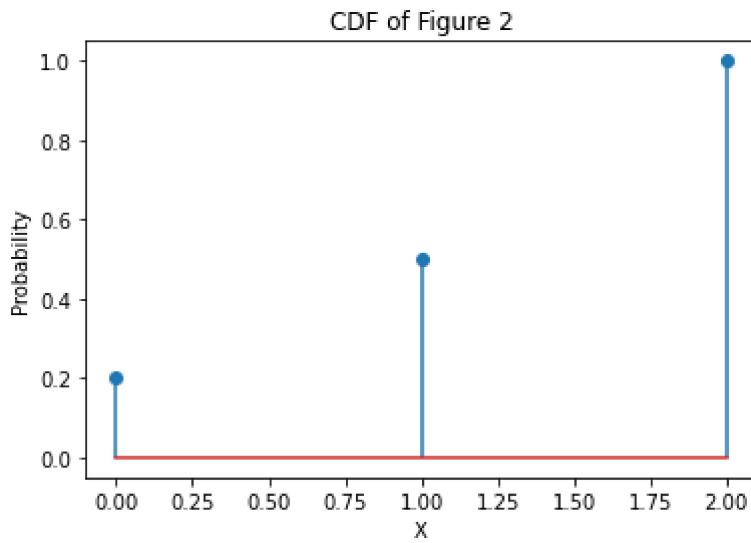
In [3]:

```
##code to display the CDF

pmf=np.array([0.2,0.3,0.5])
cdf=np.cumsum(pmf) #cumulative sum

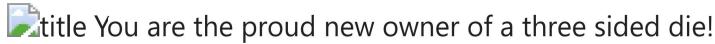
x = np.array([0,1,2])
y = np.array(cdf)

plt.stem(x, y)
plt.xlabel("X")
plt.ylabel("Probability")
plt.title("CDF of Figure 2")
plt.show()
```



## Problem 5

---



The sides are marked '1', '2', and '3'. Let  $X$  be the amount of 3's obtained after rolling the die twice.

If  $S$  is the sample space, then  $X : S \rightarrow \mathbb{R}$ , and  $p(x_i) = P(X = x_i) = P(\{s \in S | X(s) = x_i\})$

### Part A

---

1] (**2 points**) What is  $X((3,3))$ ?

2] (**2 points**) What is  $X((2,3))$ ?

**solution:**

$X((3,3)) = 2$  because there are 2 3's

$X((2,3)) = 1$  because there is just 1 3

## Part B

---

1] **(2 points)** What is  $p(2)$ ?

2] **(2 points)** What is  $p(1)$ ?

**solution:**

I am interpreting this as the probability of rolling 2 3's:

$p(2) = \frac{1}{9}$  since there is only one way to roll 2 3's out of 9 possible rolls

Similarly, I am interpreting this as the probability of rolling exactly 1 3:

$p(1) = \frac{4}{9}$  since there are only 4 ways to roll 1 3's out of 9 possible rolls

## Part C

---

1] **(2 points)** Create a table of values for the PMF.

2] **(4 points)** Create (code) a graph for the PMF.

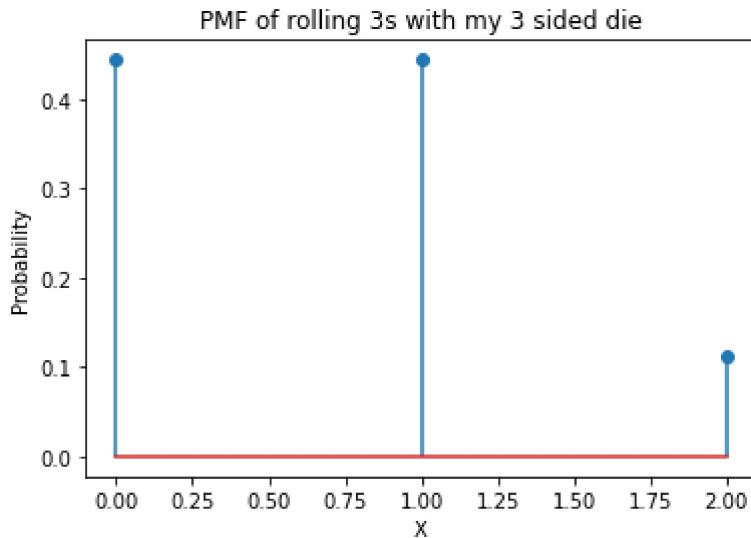
PMF

a=	0	1	2
$P(X=a)=p(a)$	$4/9$	$4/9$	$1/9$

**solution:** Put your solution to part C, #1 here:

In [4]:

```
# code for the graph of the PMF:  
  
pmf=np.array([4/9,4/9,1/9])  
  
x = np.array([0,1,2])  
y = np.array([4/9,4/9,1/9])  
  
plt.stem(x, y)  
plt.xlabel("X")  
plt.ylabel("Probability")  
plt.title("PMF of rolling 3s with my 3 sided die")  
plt.show()
```



## Part D

---

- 1] **(2 points)** Create a table of values for the CDF.
- 2] **(2 points)** Create a piecewise function for the CDF.
- 3] **(4 points)** Create (code) a graph for the CDF.

CDF

a =	0	1	2
F(a)	4/9	8/9	9/9 = 1

**solution:** Put your solution to Part D, #1, here:

Piecewise cumulative distribution function: (1)

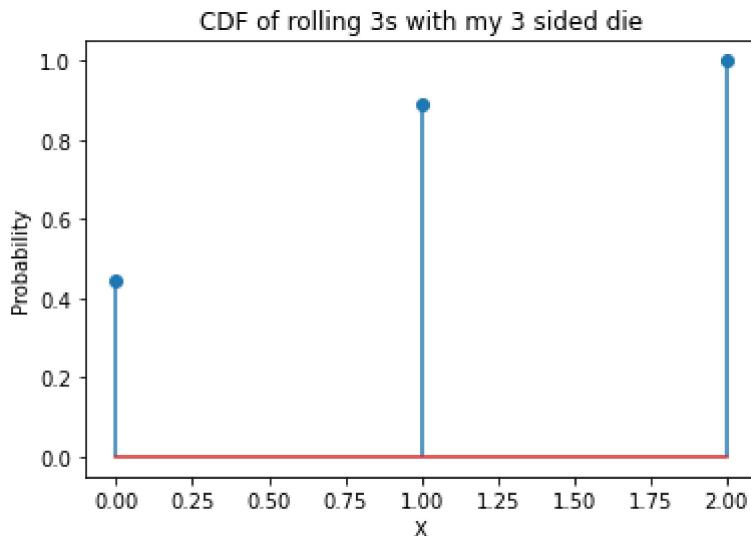
$$F(a) = \begin{cases} 0 & \text{for } x < 0 \\ \frac{4}{9} & \text{for } 0 \leq x < 1 \\ \frac{8}{9} & \text{for } 1 \leq x < 2 \\ \frac{9}{9} = 1 & \text{for } x \geq 3 \end{cases} \quad (2)$$

**solution:** Put your solution to Part D, #2, here:

```
In [5]: # Code for graph for the CDF

pmf=np.array([4/9,4/9,1/9])
cdf=np.cumsum(pmf)
x = np.array([0,1,2])
y = np.array(cdf)

plt.stem(x, y)
plt.xlabel("X")
plt.ylabel("Probability")
plt.title("CDF of rolling 3s with my 3 sided die")
plt.show()
```



## Problem 6

---

### Part A

We are still playing with our new three sided die and we are still considering rolling a '3' a success. Only now we are rolling the die 10 times!

Suppose you actually rolled the 3-sided die ten times and counted how many times you rolled a '3'.

You could get zero amount of 3's.

You could roll a '3' only once.

You could roll a '3' two out of ten times.

You might even roll a '3' ten out of ten times!

Write a **function** that takes in the parameters  $n = 10$  (for ten rolls of the 3-sided die) and  $p = \frac{1}{3}$  (for the probability of rolling a '3'). The function should **return the PMF** as a Numpy array.

**(4 points)** Use the function to print out the PMF as a table of values after rolling the 3-sided die 10 times.

i.e. the table should show the probability of rolling zero 3's, one 3, two 3's,..., ten 3's.

In [9]:

```
def binomial_pmf(n, p): ##function to calculate the PMF for a binomial distribution
    pmf = np.array([binom(n,k)*(p**k)*(1-p)**(n-k) for k in range(0,n+1)])
    return pmf

pmf = binomial_pmf(10,1/3) #using the above function to get the number of 3's after rol
print("The probability of rolling...")
```

```
for k, prob in enumerate(pmf):
    print(" {} threes is {:.4f}".format(k,prob))
```

The probability of rolling...

```
0 threes is 0.0173
1 threes is 0.0867
2 threes is 0.1951
3 threes is 0.2601
4 threes is 0.2276
5 threes is 0.1366
6 threes is 0.0569
7 threes is 0.0163
8 threes is 0.0030
9 threes is 0.0003
10 threes is 0.0000
```

## Part B

Suppose you rolled the die ten times and wrote down how many 3's resulted.

Then, you again rolled the die ten times and again wrote down how many 3's resulted.

And again you roll ten times and record. And again. And again.

In totality, lets say you recorded results 20 times. That is, twenty times in a row you rolled the 3-sided die 10 times and recorded the amount of 3's that appeared out of the 10 rolls.

You might get 20 results like [2 2 4 4 2 4 5 2 5 2 2 4 2 1 3 1 3 2 3 3] representing 2 out of 10, 2 out of ten, 4 out of ten, etc.

In order to determine how many successes (amount of 3's) TYPICALLY result when you roll this die ten times, you could look at a histogram (a distribution) of your 20 recordings. Better yet, a more accurate picture results from looking at a distribution of 100000 recordings.

**(4 points)** Create (code) a density histogram of 100000 results to get an estimation of the distribution (aka PMF).

In [10]:

```
n = 10 #roll 3 side die 10 times
p = 1/3 #prob of rolling a 3

#running 100000 experiments
sample = np.random.binomial(n=n, p=p, size=100000)
bins = np.arange(-0.5, 11.5, 1)

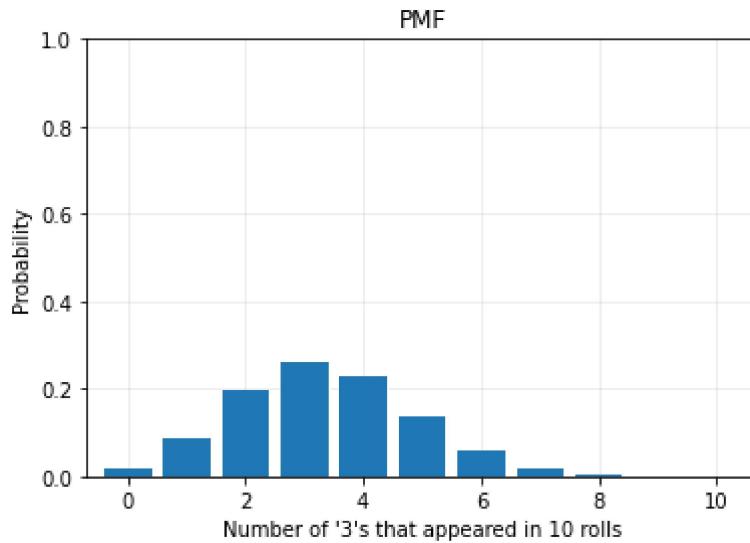
X = np.array(range(n+1))
diepmf = binomial_pmf(n=n, p=p) #using function to get pmf distribution

fig, ax = plt.subplots(nrows=1, ncols=1) #initialize graph

#greating a histogram of the pmf of the sample
ax.hist(sample, edgecolor='white', density=True, bins=bins, rwidth=0.8)
ax.set_xlabel("Number of '3's that appeared in 10 rolls")
ax.set_ylabel("Probability")
ax.set_title("PMF")
ax.grid(alpha=0.25)
```

```
ax.set_xlim([-0.7, 10.7]);
ax.set_ylim([0, 1])
ax.bar(X, diepmf)
ax.set_axisbelow(True)
ax.grid(alpha=0.25);
```

*#showing that most experiments you will roll 2-4 3's  
#but hardly ever/never get 9 or 10 threes which is what we expect*



## Part C

**(1 point)** From the PMF just created, what appears to be the most common result? In other words, how many times will '3' most commonly appear after rolling a 3-sided die ten times?

**solution:**

- The PMF above shows that most experiments you will roll 2-4 3's out of 10 rolls and you will hardly ever/never get 8-10 threes. This is what we expect since the probability of rolling a 3 is  $1/3$ .
- It is most common to roll a three 3 times during the ten times