

# Hw 1

P1

$$\cdot \{\Omega, \mathcal{F}, P\}$$

$$\cdot A \in \mathcal{F}, B \in \mathcal{F}, C \in \mathcal{F}$$

$$\cdot P(A) = 0.5$$

$$\cdot P(B) = 0.2$$

$$\cdot P(C) = 0.1$$

$$P(\overline{A \cup B}) = 0.45$$

$$\overline{P(A \cup B)} = 1 - 0.45$$

$$P(A \cup B) = 0.55 \Big] \downarrow$$

either A or B occur

$$P(A) + P(B) - P(A \cap B) = P(A \cup B)$$

$$0.55 = 0.5 + 0.2 - P(A \cap B)$$

$$P(A \cap B) = 0.15$$

1.  $P(A \cup B) = 0.55$

2.  $P(A \cap B) = 0.15$

3.  $P(A) - P(A \cap B) = 0.35$

4.  $A$  &  $B$  are not mutually exclusive, as  $P(A \cap B) > 0$

P2.

fair 4-sided die

1.  $P$  that on four rolls, at least 1 is a 4

$$P(\text{at least 1 } 4) = 1 - P(\text{no } 4)$$

$$1 - \left(\frac{3}{4}\right)^4 = 1 - \frac{81}{256} = \frac{175}{256}$$

0.683

2.  $P(\text{at least 1 a in 20}) =$

$$1 - P(\text{no a in 20})$$

$$1 - \left(\frac{3}{4}\right)^{20} = 0.947$$

3.  $0.90 = 1 - \left(\frac{3}{4}\right)^x$

$$0.10 = \left(\frac{3}{4}\right)^x$$

$$\log_{\frac{3}{4}} 0.10 = x = 8 \text{ rolls}$$

P3 . six-sided die - even  
faces twice as likely as

odd faces

• even -  $\frac{2}{9}$  each side • odd -  $\frac{1}{9}$  each side

$$P(\text{even}) = \frac{6}{9} = \frac{2}{3} \quad P(\text{odd}) = \frac{3}{9} = \frac{1}{3}$$

$P(\text{outcome} < u)$

$$P(3) + P(2) + P(1) = \frac{1}{9} + \frac{2}{9} + \frac{1}{9} = \frac{4}{9}$$

P4

•  $n$  power plants

• power plant  $i$  of  $n$  fails w/ probability  $P(i)$  independent of other power plants

• 2 power plants sufficient

• necessary to prevent black-out

• probability doesn't fail

$$= 1 - P_i$$

P5

a) probability of an ace =  $\frac{4}{52} = \frac{1}{13}$

b) jack of a spade =  $\frac{1}{52}$

c) jack of spades or  
six of diamonds =  $\frac{2}{52} = \frac{1}{26}$

d) any suite not spades or hearts  
=  $\frac{26}{52} = \frac{1}{2}$

P6)

$$P(H_1) = \frac{13}{52} = \frac{1}{4}$$

$$P(H_2) = \frac{12}{51} \cdot \frac{13}{52} + \frac{13}{51} \cdot \frac{39}{52}$$

$$\frac{156}{2652} + \frac{507}{2652} = \frac{663}{2652} = \frac{1}{4}$$

# Homework 1 Part 2

This is an individual assignment.

## Description

Create or edit this Jupyter Notebook to answer the questions below. Use simulations to answer these questions. An analytical solution can be useful to check if your simulation is correct but analytical solutions alone will not be accepted as a solution to a problem.

### Problem 9

Consider repeatedly rolling a fair 4-sided die.

1. Create a simulation to compute the probability that the top face will be 4 at least once on four rolls of the die?
2. Create a simulation to compute the probability that the top face will be 4 at least once on 20 rolls of the die?
3. Create a simulation to compute how many rolls of the die would you have to do to be 90% confident that you would see at least one 4?
4. Using the formula you have computed in problem 2 part 4, make a Python function that takes in the target value  $p$  and outputs the required number of rolls of an integer.
  - A. Find the values for  $p = 0.95$  and  $p = 0.99$ .
  - B. Use your simulation to verify that the number of rolls you specified is sufficient to achieve  $p \geq 0.95$ .

```
In [99]: import random
import numpy as np
import numpy.random as npr
import math

import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('ggplot')
```

```
In [100... def rollFourSided(numRolls = 4, numSims = 1000):
    faces = range(1,5)
    fourRolls = 0
    for sim in range(numSims):
        event = 0
        for rolls in range(numRolls):
            roll = random.choice(faces)
            if(roll == 4):
                event += 1
        if(event > 0):
            fourRolls += 1
    print('The probability of rolling a 4 at least once in', numRolls, 'rolls is',

In [101... rollFourSided()

The probability of rolling a 4 at least once in 4 rolls is 0.687

In [102... rollFourSided(20)

The probability of rolling a 4 at least once in 20 rolls is 0.998

In [103... def findRollCount(prob = 0.9, numSims = 1000):
    faces = range(1,5)
    rollCount = 0
    i = 1
    while(True):
        i += 1
        event = 0
        for sim in range(numSims):
            fourCheck = False ##Ensures that if more than one fours shows up, its n
            for j in range(i):
                roll=random.choice(faces)
                if(roll == 4):
                    fourCheck = True
            if(fourCheck):
                event += 1
        probCheck = event/numSims
        if(probCheck >= prob):
            break

    print('The experimental number of rolls to reach a probability of', prob, 'that

In [104... findRollCount()

The experimental number of rolls to reach a probability of 0.9 that at least one fo
ur will show up is 9

In [105... def calculatedRollCount(prob = 0.95):
    inverseProb = 1-prob
    x = math.log(inverseProb, 0.75)
    rolls = int (np.ceil(x))
    print('The calculated number of rolls to reach a probability', prob, 'is', roll

In [106... calculatedRollCount()

The calculated number of rolls to reach a probability 0.95 is 11
```

In [107...

calculatedRollCount(0.99)

The calculated number of rolls to reach a probability 0.99 is 17

## Problem 10

Create a simulation function where you will roll a fair 6-sided die twice. Use simulation to find out the probability of getting a 4,5, or 6 on the first toss and a 1,2,3 on the second toss.

In [108...

```
numSims = 100000
event = 0
for sim in range(numSims):
    faces = range(1,7)
    rollOne = random.choice(faces)
    rollTwo = random.choice(faces)
    if((rollOne > 3) and (rollTwo < 4)):
        event += 1
print('Probability equals', event/numSims)
```

Probability equals 0.24875

## Problem 11

Suppose that you have a bag with 3 coins. One of them is a fair coin, but the others are biased trick coins. When flipped, the three coins come up heads with probability  $\frac{1}{2}$ ,  $\frac{1}{3}$ , and  $\frac{1}{4}$ , respectively.

Consider the experiment where you pick one coin at random and flip it three times. Let  $H_i$  be the event that the coin comes up heads on flip  $i$ . What is the probability of the outcome  $H_1 \cap H_2 \cap \overline{H_3}$ ?

With small modification in your code, find out the probability of the outcome  $H_1 \cap \overline{H_2} \cap \overline{H_3}$ .

Use simulation to find out the probability.



In [109...

```
def threeFlips(numSims = 1000):
    event = 0
    for sim in range(numSims):
        coins = ['fair', 'biased', 'superBiased']
        coin = random.choice(coins)
        if(coin == 'fair'):
            thisCoin = ['H','T']
        if(coin == 'biased'):
            thisCoin = ['H','T','T']
        if(coin == 'superBiased'):
            thisCoin = ['H','T','T','T']
        flipOne = random.choice(thisCoin)
        flipTwo = random.choice(thisCoin)
        flipThree = random.choice(thisCoin)
        if((flipOne == 'H') and (flipTwo == 'H') and (flipThree == 'T')):
            event += 1
    print('The probability of Heads, Heads, Tails is', event/numSims)

def threeFlipsTwo(numSims = 1000):
    event = 0
    for sim in range(numSims):
        coins = ['fair', 'biased', 'superBiased']
        coin = random.choice(coins)
        if(coin == 'fair'):
            thisCoin = ['H','T']
        if(coin == 'biased'):
            thisCoin = ['H','T','T']
        if(coin == 'superBiased'):
            thisCoin = ['H','T','T','T']
        flipOne = random.choice(thisCoin)
        flipTwo = random.choice(thisCoin)
        flipThree = random.choice(thisCoin)
        if((flipOne == 'H') and (flipTwo == 'T') and (flipThree == 'T')):
            event += 1
    print('The probability of Heads, Tails, Tails is', event/numSims)
```

In [110...

```
threeFlips(10000)
threeFlipsTwo(10000)
```

The probability of Heads, Heads, Tails is 0.0805  
The probability of Heads, Tails, Tails is 0.136

## Submit Your Solutions

Confirm that you've successfully completed the assignment.

Along with the Notebook, include a PDF of the notebook with your solutions.

add and commit the final version of your work, and push your PDF file to your GitHub repository.

Submit the URL of your GitHub Repository as your assignment submission on Canvas.