

# 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.

```
In [1]: import random
import numpy as np
import numpy.random as npr
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('bmh')
```

## Problem 1

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 [17]: #Part 1
simulations = 10000
dice = [1,2,3,4,5,6]
rolls = 4
fours = 0

for i in range (simulations):
    roll = random.choices(dice, k = rolls)
    for j in (roll):
        if j == 4:
            fours += 1
            break
print(f"In {simulations} experiments, at least one four was rolled {fours} times.")
print(f"Probability of rolling at least one four in {rolls} rolls: {fours/simulations}")

#Part 2
rolls = 20
fours = 0

for i in range (simulations):
    roll = random.choices(dice, k = rolls)
    for j in (roll):
        if j == 4:
            fours += 1
            break
print(f"In {simulations} experiments, at least one four was rolled {fours} times.")
print(f"Probability of rolling at least one four in {rolls} rolls: {fours/simulations}")

#Part 3
rolls = 20
fours = 0
looping = True

while looping == True:
    #Resets the four counter
    fours = 0
    #Runs the previous code from parts 1 and 2
    for i in range (simulations):
        roll = random.choices(dice, k = rolls)
        for j in (roll):
            if j == 4:
                fours += 1
                break
    #Removes a dice from the pool to test a lower number
    rolls -= 1
    #Checks whether the probability is close to 90
    if (fours/simulations*100 <= 90):
        looping = False
        break

print(f"Probability of rolling at least one four in {rolls} rolls: {fours/simulations}")
print(f"With {rolls} to {rolls + 1} dice, there is about a 90% probability of rolling at least one four")

#Part 4
rolls = 20
fours = 0
looping = True

```

```

while looping == True:
    #Resets the four counter
    fours = 0
    #Runs the previous code from parts 1 and 2
    for i in range (simulations):
        roll = random.choices(dice, k = rolls)
        for j in (roll):
            if j == 4:
                fours += 1
                break
    #Removes a dice from the pool to test a lower number
    rolls -= 1
    #Checks whether the probability is close to 90
    if (fours/simulations*100 <= 95):
        looping = False
        break

print(f"Probability of rolling at least one four in {rolls} rolls: {fours/simulations*100}")
print(f"With {rolls} to {rolls + 1} dice, there is about a 90% probability of rolling at least one four.")

#Added 10 more dice for simulating the .99 probability
rolls = 30
fours = 0
looping = True

while looping == True:
    #Resets the four counter
    fours = 0
    #Runs the previous code from parts 1 and 2
    for i in range (simulations):
        roll = random.choices(dice, k = rolls)
        for j in (roll):
            if j == 4:
                fours += 1
                break
    #Removes a dice from the pool to test a lower number
    rolls -= 1
    #Checks whether the probability is close to 90
    if (fours/simulations*100 <= 99):
        looping = False
        break

print(f"Probability of rolling at least one four in {rolls} rolls: {fours/simulations*100}")
print(f"With {rolls} to {rolls + 1} dice, there is about a 90% probability of rolling at least one four.")

```

In 10000 experiments, at least one four was rolled 5177 times.  
 Probability of rolling at least one four in 4 rolls: 51.77

In 10000 experiments, at least one four was rolled 9734 times.  
 Probability of rolling at least one four in 20 rolls: 97.34

Probability of rolling at least one four in 11 rolls: 88.63  
 With 11 to 12 dice, there is about a 90% probability of rolling at least one four.

Probability of rolling at least one four in 15 rolls: 94.73  
 With 15 to 16 dice, there is about a 90% probability of rolling at least one four.

Probability of rolling at least one four in 23 rolls: 98.8  
 With 23 to 24 dice, there is about a 90% probability of rolling at least one four.

## Problem 2

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 or 6 on the first toss and a 1,2,3, or 5 on the second toss.

```
In [22]: simulations = 10000
dice = [1,2,3,4,5,6]
rolls = 2
win = 0

for i in range(simulations):
    roll = random.choices(dice, k = rolls)
    if roll[0] == 4 or roll[0] == 6:
        if roll[1] == 1 or roll[1] == 2 or roll[1] == 3 or roll[1] == 5:
            win += 1
print(f"Probability of rolling a 4 or 6 and then a 1, 2, 3, or 5 is: {win/simulations}")
```

Probability of rolling a 4 or 6 and then a 1, 2, 3, or 5 is: 21.98 in 10000 experiments!

## Problem 3

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  $\frac{1}{2}$ ,  $\frac{1}{4}$ , and  $\frac{1}{6}$ , 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 [31]: #Set variables
coin = 0
simulations = 10000
outcome = 0

#Select a coin
bag = [1,2,3]
bag_choice = random.choice(bag)
#Define the space of the coin
if bag_choice == 1:
    coin = ["Heads", "Tails"]
if bag_choice == 2:
    coin = ["Heads", "Tails", "Tails", "Tails"]
if bag_choice == 3:
    coin = ["Heads", "Tails", "Tails", "Tails", "Tails", "Tails"]
#I'm 101% sure there is a better way of doing this

#Simulate the experiment
for i in range (simulations):
    flip = random.choices(coin, k = 3)
    if flip[0] == "Heads" and flip[1] == "Heads" and flip[2] == "Tails":
        outcome += 1

print(f"The probability of getting two heads and one tail is {outcome/simulations*100}")

#Modification for heads, tails, tails

bag_choice = random.choice(bag)
#Define the space of the coin
if bag_choice == 1:
    coin = ["Heads", "Tails"]
if bag_choice == 2:
    coin = ["Heads", "Tails", "Tails", "Tails"]
if bag_choice == 3:
    coin = ["Heads", "Tails", "Tails", "Tails", "Tails", "Tails"]
#I'm 101% sure there is a better way of doing this

#Simulate the experiment
for i in range (simulations):
    flip = random.choices(coin, k = 3)
    if flip[0] == "Heads" and flip[1] == "Tails" and flip[2] == "Tails":
        outcome += 1
print(f"The probability of getting one head and two tails is {outcome/simulations*100}")

```

The probability of getting two heads and one tail is 5.050000000000001

The probability of getting one head and two tails is 17.49

# 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.

In [ ]: