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 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 [4]:

```
import random
import numpy as np
import numpy.random as npr

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

In [30]:

```
# Set the number of simulations
num_sims = 10000

# Set the number of rolls in each simulation
num_rolls = 4

# Initialize the counter for events where the top face is 4
event_counter = 0

# Perform simulations
for i in range(num_sims):
    # Roll the die for the specified number of times
    for j in range(num_rolls):
        # Simulate a roll of a fair 4-sided die
        roll_dice = random.randint(1, 4)

        # Check if the top face is 4, and if so, break out of the inner loc
        if roll_dice == 4:
            event_counter += 1
            break

# Calculate the probability of the event occurring at least once in the spe
probability = event_counter / num_sims

# Print the result
print("The probability that the top face will be 4 at least once on four rc
```

```
The probability that the top face will be 4 at least once on four rolls o
f the die is 0.681
```

In [27]:

```
# Set the number of simulations
num_sims = 10000

# Set the number of rolls in each simulation
num_rolls = 20

# Initialize the counter for events where the top face is 4
event_counter = 0

# Perform simulations
for i in range(num_sims):
    # Roll the die for the specified number of times
    for j in range(num_rolls):
        # Simulate a roll of a fair 4-sided die
        roll_dice = random.randint(1, 4)

        # Check if the top face is 4, and if so, break out of the inner loo
        if roll_dice == 4:
            event_counter += 1
            break

# Calculate the probability of the event occurring at least once in the spe
probability = event_counter / num_sims

# Print the result
print("The probability that the top face will be 4 at least once on four ro
```

```
The probability that the top face will be 4 at least once on four rolls o
f the die is 0.997
```

In [52]:

```
# Set the target confidence and the number of simulations
target_confidence = 0.90
num_sims = 1000

# Initialize the counter for the number of rolls
num_rolls = 1

# Perform simulations
while True:
    event_counter = 0
    for _ in range(num_sims):
        for _ in range(num_rolls):
            # Simulate a roll of a fair 4-sided die
            roll_dice = random.randint(1, 4)

            # Check if the top face is 4, and if so, break out of the inner
            if roll_dice == 4:
                event_counter += 1
                break

    # Calculate the confidence of the event occurring at least once in the
    confidence = event_counter / num_sims

    # If the confidence is greater than or equal to the target confidence,
    if confidence >= target_confidence:
        break

    num_rolls += 1

# Print the result
print("You would need", num_rolls, "rolls to be 90% confident that you woul
```

```
You would need 9 rolls to be 90% confident that you would see at least on
e 4
```

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 [53]:
```
# Set the number of simulations
num_sims = 10000

# Initialize the counter for events where the top face is 4 or 6 on the fir
event_counter = 0

# Perform simulations
for i in range(num_sims):
    # Simulate a roll of a fair 6-sided die
    roll_dice1 = random.randint(1, 6)
    roll_dice2 = random.randint(1, 6)

    # Check if the top face is 4 or 6 on the first toss and 1, 2, 3, or 5 c
    if (roll_dice1 in {4, 6}) and (roll_dice2 in {1, 2, 3, 5}):
        event_counter += 1

# Calculate the probability of the event
probability = event_counter / num_sims

# Print the result
print("The probability of getting a 4 or 6 on the first toss and a 1, 2, 3,
```

```
The probability of getting a 4 or 6 on the first toss and a 1, 2, 3, or 5
on the second toss is 0.2213
```

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 [54]:

```
# Set the number of simulations
num_sims = 10000

# Initialize counters for the two events
event_counter1 = 0
event_counter2 = 0

# Probabilities of getting heads for the three coins
probs = [1/2, 1/4, 1/6]

# Function to simulate flipping a coin with a given probability of getting
def flip_coin(p):
    return random.random() < p

# Perform simulations
for i in range(num_sims):
    p = random.choice(probs)
    flips = [flip_coin(p) for i in range(3)]

    # Check if the coin came up heads on all three flips
    if all(flips):
        event_counter1 += 1

    # Check if the coin came up heads on the first two flips
    if all(flips[:2]):
        event_counter2 += 1

# Calculate the probabilities of the two events
probability1 = event_counter1 / num_sims
probability2 = event_counter2 / num_sims

# Print the results
print("The probability of getting heads on all three flips is", probability
print("The probability of getting heads on the first two flips is", probabi
```

```
The probability of getting heads on all three flips is 0.0481
The probability of getting heads on the first two flips is 0.1129
```

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.