Topic 3 DQ 2

"Suppose you are given a biased coin with unknown probability of landing heads. Design an experiment to estimate the probability of heads using probability theory. How would you determine the number of coin flips to achieve a certain level of confidence in your estimate? Additionally, how can Python be used to simulate this experiment and visualize the results? Provide a detailed description of your experiment and your reasoning behind your choices."

Designing the Experiment

Objective: Estimate the probability of landing heads (p) for a biased coin.

Methodology:

Law of Large Numbers: The experiment relies on the law of large numbers, which states that as the number of trials increases, the sample mean will converge to the expected value. In this case, the proportion of heads observed will converge to the true probability of landing heads as the number of flips increases.

Confidence Interval: To determine the number of flips needed to achieve a certain level of confidence in the estimate, we can use a confidence interval for a proportion. The width of the confidence interval decreases as the number of flips increases. For a simple approximation, the margin of error (MoE) for a 95% confidence level can be calculated as:

where p is the estimated probability of heads, and n is the number of flips. We can solve for n to achieve a desired MoE.

Experiment Design: Decide on the desired level of confidence. Use the formula above to calculate the minimum number of flips required.

In [ ]:

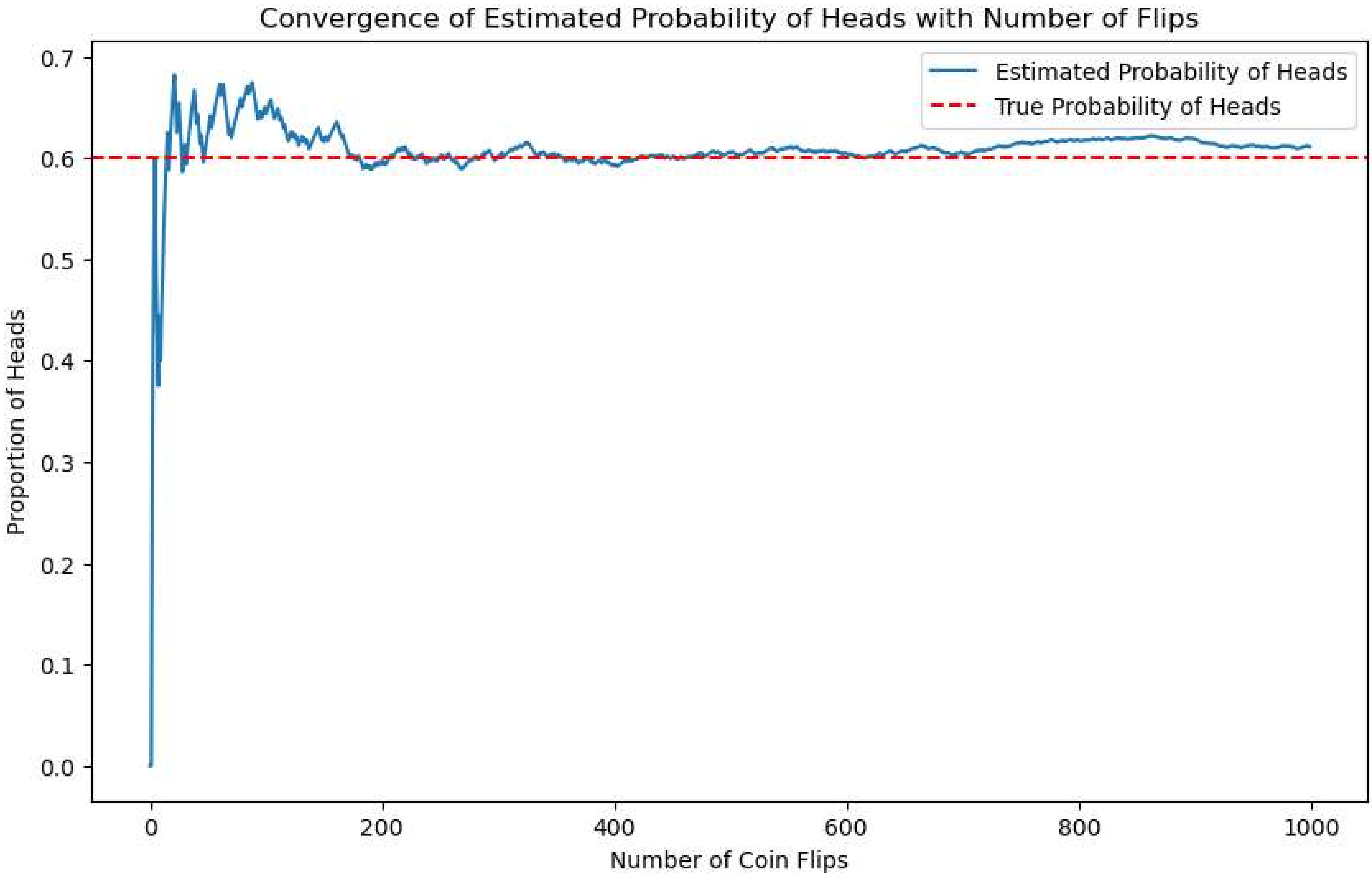
```python
import numpy as np
import matplotlib.pyplot as plt

# Simulate flipping a biased coin
def simulate_coin_flips(n, p_head):
    flips = np.random.choice(['Head', 'Tail'], size=n, p=[p_head, 1-p_head])
    heads_count = np.cumsum(flips == 'Head')
    proportion_heads = heads_count / np.arange(1, n+1)
    return proportion_heads

# Parameters
n_flips = 1000  # Number of flips
p_head_true = 0.6  # True probability of heads

# Simulation
proportion_heads = simulate_coin_flips(n_flips, p_head_true)

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(proportion_heads, label='Estimated Probability of Heads')
plt.axhline(y=p_head_true, color='r', linestyle='--', label='True Probability of Heads')
plt.xlabel('Number of Coin Flips')
plt.ylabel('Proportion of Heads')
plt.title('Convergence of Estimated Probability of Heads with Number of Flips')
plt.legend()
plt.show()
```



The plot shows the convergence of the estimated probability of heads towards the true probability as the number of coin flips increases, illustrating the law of large numbers. The horizontal dashed line represents the true probability of heads, while the solid line shows the estimated probability based on the cumulative proportion of heads observed.

This experiment design and simulation demonstrate how probability theory and statistical methods can be applied to estimate the probability of an event under uncertainty. Python serves as a powerful tool for simulating experiments, performing calculations, and visualizing results, enabling a deeper understanding of statistical concepts and their practical applications.

Sources:

Schwartz, J. (2020). The Law of Large Numbers. Retrieved from https://math.uchicago.edu/~may/VIGRE/VIGRE2008/REUPapers/Schwartz.pdf

ScienceDirect. (2004). Convergence in Probability. In ScienceDirect Topics. Retrieved February 10, 2024, from https://www.sciencedirect.com/topics/mathematics/convergence-in-probability

Rogel-Salazar, J. (2023). Definitely Maybe: Probability and Distributions. In Statistics and data visualisation with Python (1st ed.). CRC Press.

Routledge, R. (2024, February 3). law of large numbers. Encyclopedia Britannica. https://www.britannica.com/science/law-of-large-numbers