지능형통신시스템 - 프로그래밍 과제

i) 설계

배경 : 주사위를 던졌을 때 6이 나올 확률을 몬테카를로 시뮬레이션을 실행하여 알아보고자 하였으며 강의자료에서 예시로 보여주신 동전을 던졌을 때의 확률, Pi 값의 확률에 대한 코드를 반영하여 코드를 작성하고자 하였다.

컴퓨터 환경 : MAC 14.0, Colab 에서 구현

예상 결과 : 주사위 6개의 면 중, 한 면이 나오는 것이기 때문에 1/6 의 확률로 근삿값인 0.16으로 수렴하는 그래프를 보일 것으로 예상된다.

ii) 코드 작성

```
import matplotlib.pyplot as plt
import random

# 시뮬레이션 설정
num = 10000
count_six = 0
res = []

# 시뮬레이션 실행
for i in range(1, num):
    # 주사위 던지기 (1에서 6 사이의 정수 무작위 생성)
    roll = random.randint(1, 6)
    if roll == 6:
        count_six += 1

    # 현재까지의 6 나올 확률 계산 및 기록
    res.append(count_six / i)

# 결과 시각화
plt.plot(range(1, num), res, 'bo:', label=f"Final Probability
plt.legend(loc='lower right')
```

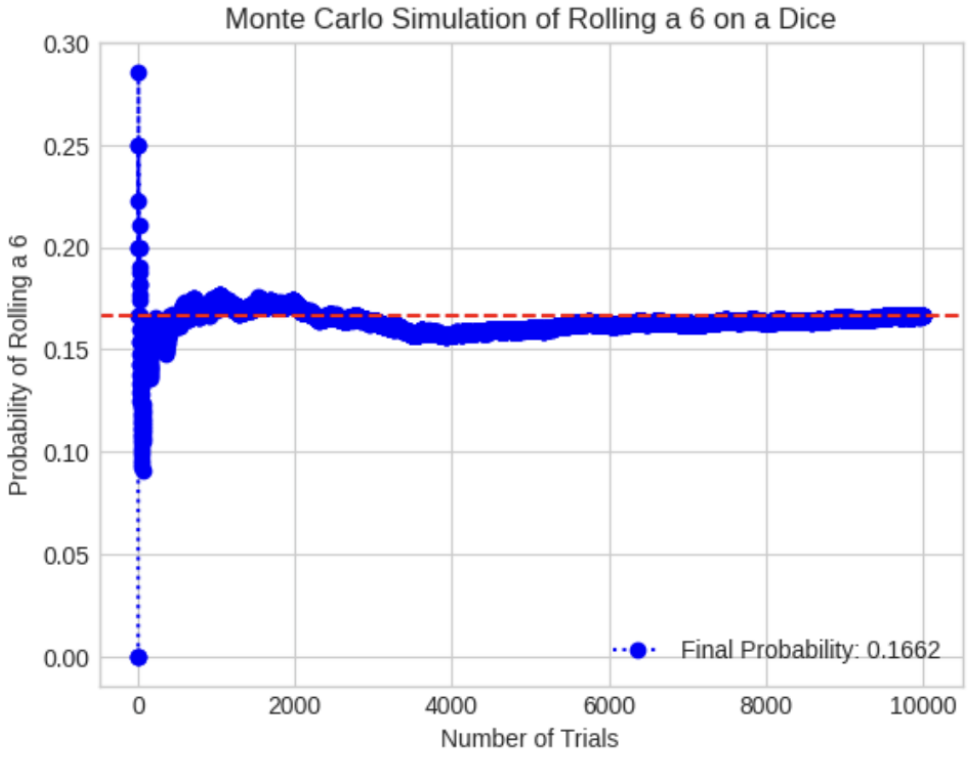
```
plt.axhline(y=res[-1], color="r", linestyle="--")

plt.xlabel('Number of Trials')
plt.ylabel('Probability of Rolling a 6')
plt.title('Monte Carlo Simulation of Rolling a 6 on a Dice')
plt.show()
```

주석 이외의 추가적인 설명을 덧붙이자면, 예시 코드에서는 random.uniform 을 사용하여 랜덤한 실수를 생성하였다면 위 코드에서는 random.randint 를 사용하여 랜덤한 정수를 생성하였다. uniform 은 [a, b)에서 랜덤한 실수를 반환하고, randint 는 [a, b]에서 랜덤한 정수를 반환한다. 따라서 주사위 면을 선택하는 위 상황에서는 randint 가 더 적절하다고 판단하였다.

또한 xlabel, ylabel, title 을 추가해주어 가독성을 높였다.

iii) 결과 분석



앞선 코드를 실행하였을 때 위와 같은 그래프가 결과로 나왔다. 이는 예상하였던 1/6 ≒ 0.1666 에 수렴하는 형태를 보인다. Final Probability Ehgks 0.1662 로 예상범위와 0.241% 의 차이를 보이기 때문에 결과값이 옳게 출력되었음을 알 수 있다.

이는 특정한 수에 수렴하는 형태를 가지기 때문에 지수분포나 누적 분포 함수로 설명하기 어렵다. 수업 시간에 학습한 지수분포나 CDF 는 특정 확률 분포이지만, 위의 결과는 단순한 빈도 기반 확률 계산이다. 주사위에서 각 숫자가 나올 확률은 균등하며, 위 코드에서는 '6'이 나올 확률이 시간(시도 횟수)에 따라 어떻게 변하는지를 보여주고 있다. 이는 균등 분포를 따르며, 위에서 언급한 분포들과는 다른 특성을 가지고 있다.