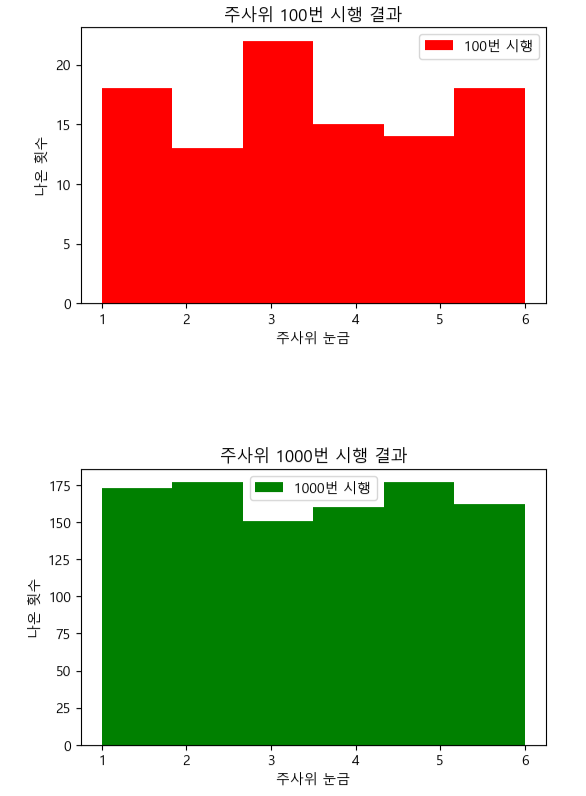
**<주사위 시뮬레이션 결과 분석>**

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* 주사위의 시행 횟수를 100번에서 100,000번까지 늘릴수록 각각의 눈금이 나오는 확률이 1/6에 가까워진다. (균등한 분포를 보여줌)

텍스트, 도표, 스크린샷, 직사각형이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 도표, 직사각형이(가) 표시된 사진

자동 생성된 설명

* 각 눈금이 몇 번 나왔는지를 보다 정밀하게 측정해보기 위해 y범위를 줄이고 히스토그램을 나타내 보았다. (해당 그래프를 나타내기 위한 코드는 아래 결과 밑에 첨부함)

1. 100번 시행 결과
   * 각 주사위 눈금은 수학적 확률에 근거하면 약 16.6번이 나와야 한다.
   * 1, 3, 6 눈금이 2, 4, 5 눈금에 비해 많이 나왔다. (16.6번 보다 많이 나왔다.)
2. 1,000번 시행 결과
   * 각 주사위 눈금은 수학적 확률에 근거하면 약 166.6번 나와야 한다.
   * 1, 2, 5 눈금이 3, 4, 6 눈금에 비해 많이 나왔다. (166.6번 보다 많이 나왔다.)
3. 10,000번 시행 결과
   * 각 주사위 눈금은 수학적 확률에 근거하면 약 1666.6번 나와야 한다.
   * 1, 2, 5, 6 눈금이 3, 4 눈금에 비해 많이 나왔다. (1666.6번 보다 많이 나왔다.)
4. 100,000번 시행 결과
   * 각 주사위 눈금은 수학적 확률에 근거하면 약 16666.6번 나와야 한다.
   * 3, 5, 6 눈금이 1, 2, 4 눈금에 비해 많이 나왔다. (16666.6번 보다 많이 나왔다.)

|  |
| --- |
| import csv  import matplotlib.pyplot as plt  def main():  f = open('random.csv','r')  data = csv.reader(f, delimiter=',')  next(data)  result = [[],[],[],[]]  for row in data:  if(row[0] != ''):  result[0].append(int(row[0]))  if(row[1] != ''):  result[1].append(int(row[1]))  if(row[2] != ''):  result[2].append(int(row[2]))  if(row[3] != ''):  result[3].append(int(row[3]))  f.close()    plt.rcParams['font.family']='Malgun Gothic'  plt.rcParams['axes.unicode\_minus']=False    plt.figure(figsize=(6,20))    plt.subplot(4,1,1)  plt.title('주사위 100번 시행 결과')  plt.xlabel('주사위 눈금')  plt.ylabel('나온 횟수')  plt.hist(result[0], bins=6, color='r', label='100번 시행', histtype='bar')  plt.ylim([  min(result[0].count(1),  result[0].count(2),  result[0].count(3),  result[0].count(4),  result[0].count(5),  result[0].count(6)) - 1,  max(result[0].count(1),  result[0].count(2),  result[0].count(3),  result[0].count(4),  result[0].count(5),  result[0].count(6)) + 1])  plt.legend()    plt.subplot(4,1,2)  plt.title('주사위 1000번 시행 결과')  plt.xlabel('주사위 눈금')  plt.ylabel('나온 횟수')  plt.hist(result[1], bins=6, color='g', label='1000번 시행')  plt.ylim([  min(result[1].count(1),  result[1].count(2),  result[1].count(3),  result[1].count(4),  result[1].count(5),  result[1].count(6)) - 1,  max(result[1].count(1),  result[1].count(2),  result[1].count(3),  result[1].count(4),  result[1].count(5),  result[1].count(6)) + 1])  plt.legend()    plt.subplot(4,1,3)  plt.title('주사위 10000번 시행 결과')  plt.xlabel('주사위 눈금')  plt.ylabel('나온 횟수')  plt.hist(result[2], bins=6, color='b', label='10000번 시행')  plt.ylim([  min(result[2].count(1),  result[2].count(2),  result[2].count(3),  result[2].count(4),  result[2].count(5),  result[2].count(6)) - 10,  max(result[2].count(1),  result[2].count(2),  result[2].count(3),  result[2].count(4),  result[2].count(5),  result[2].count(6)) + 10])  plt.legend()    plt.subplot(4,1,4)  plt.title('주사위 100000번 시행 결과')  plt.xlabel('주사위 눈금')  plt.ylabel('나온 횟수')  plt.hist(result[3], bins=6, color='y', label='100000번 시행')  plt.ylim([  min(result[3].count(1),  result[3].count(2),  result[3].count(3),  result[3].count(4),  result[3].count(5),  result[3].count(6)) - 10,  max(result[3].count(1),  result[3].count(2),  result[3].count(3),  result[3].count(4),  result[3].count(5),  result[3].count(6)) + 10])  plt.legend()    plt.subplots\_adjust(left=0.125,  bottom=0.1,  right=0.9,  top=0.9,  wspace=0.2,  hspace=0.6)    plt.show()    if \_\_name\_\_=="\_\_main\_\_":  main() |