

프로그래밍을 이용한 주사위 던지기 모의실험

모의실험을 이용하여 "주사위 6번 던지기"를 반복적으로 실행할 때 숫자 1이 나오는 횟수의 평균의 분포를 확인하고자 한다. 즉, 주사위를 6번 던져서 숫자 1이 평균 1회 나와야 한다는 사실을 실험을 통해 확인한다. 또한, 주사위를 6번 던져서 숫자 1이 나오는 횟수의 평균분포가 정규분포를 따른다는 사실을 실험적으로 확인한다.

모의실험 1: 주사위 6번 던지기

주사위를 던지면 1에서 6까지의 정수가 무작위로 나온다. 이런 성질을 코딩에서는 "난수 생성"으로 구현할 수 있다.

모든 프로그래밍언어에는 난수를 생성하는 기능이 들어 있으며, 여기서는 파이썬(Python)이라는 언어를 사용한다.

파이썬에서 무작위로 생성하기

numpy라는 라이브러리에서 제공하는 randint 함수를 이용하여 지정된 구간에서 정수를 무작위로 생성할 수 있다.

```
In [1]: import numpy as np
```

예를 들어, 아래 코드는 0과 10 사이의 정수를 무작위적으로 5개 생성한다.

주의:

- 0은 포함됨
- 10은 포함되지 않음

```
In [2]: np.random.randint(0, 10, 5)
```

```
Out[2]: array([9, 7, 0, 3, 9])
```

따라서 주사위를 6번 던진 결과를 생성하는 코드는 다음과 같다.

```
In [3]: cube_6_times = np.random.randint(1,7,6)
```

```
In [4]: cube_6_times
```

```
Out[4]: array([6, 4, 6, 6, 5, 5])
```

모의실험 2: "주사위 6번 던지기"를 반복하기

모의실험 1을 원하는 횟수만큼 반복하기를 구현하도록 하자.

아래 코드에 정의된 `cube_experiment` 함수는 반복횟수를 입력 받으면 "주사위 6번 던지기"를 지정된 반복횟수만큼 반복하고, 반복할 때마다 1이 나온 횟수를 기록한다.

```
In [5]: def cube_experiment(num_to_repeat):  
  
    count_ones = np.empty([num_to_repeat,], dtype=float)  
  
    for times in np.arange(num_to_repeat):  
        experiment = np.random.randint(1,7,6)  
        count_ones[times] = experiment[experiment==1].shape[0]  
  
    return count_ones
```

```
In [6]: print(cube_experiment(30))  
  
[ 0.  1.  1.  3.  1.  0.  0.  1.  1.  1.  2.  0.  0.  0.  0.  3.  
 2.  0.  
 0.  2.  0.  2.  0.  1.  0.  0.  1.  0.  1.  2.]
```

표본평균의 분포

그래프 관련 라이브러리가 필요하다.

```
In [7]: import matplotlib.pyplot as plt  
        %matplotlib inline  
  
        import seaborn as sns  
        sns.set(color_codes = True)
```

'주사위 6번 던지기'를 30번 반복했을 때 1이 나온 평균들의 분포를 알아보자.

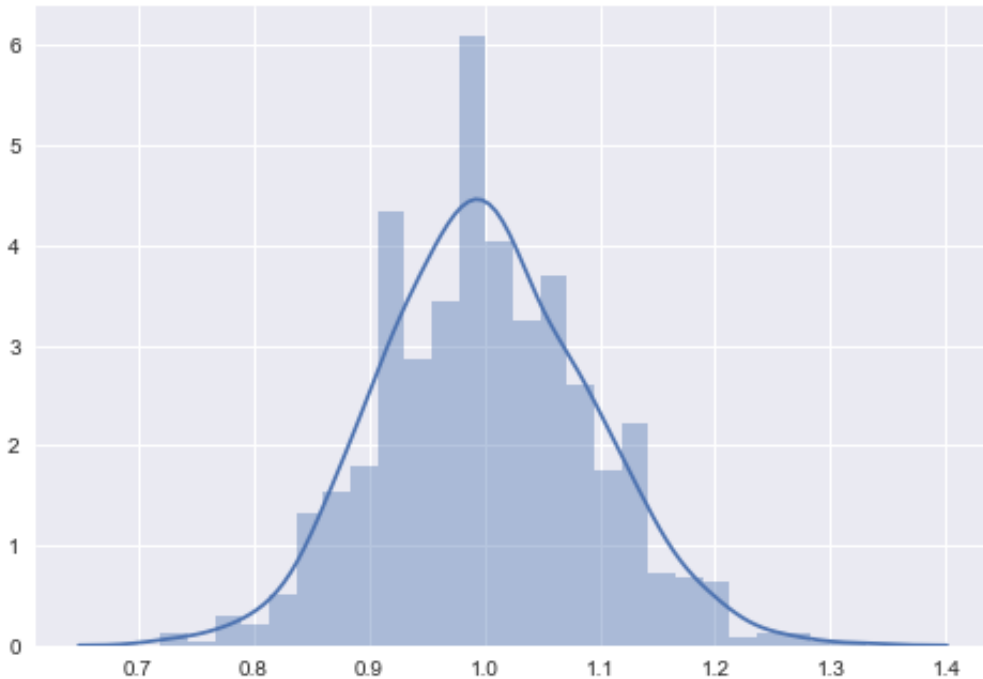
표본평균의 분포를 그래프로 확인하기 위해 "'주사위 6번 던지기'를 30번 반복하기"를 1000번 정도 해보도록 하자.

```
In [8]: count_ones_mean = np.empty([1000, ], dtype=float)

for times in np.arange(1000):
    count_ones_mean[times] = cube_experiment(100).mean()

sns.distplot(count_ones_mean, kde=True)
```

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x1112ff750>



신뢰구간

"동전 6번 던지기"를 30번 반복했을 때 숫자 1이 나오는 평균횟수의 신뢰구간을 다음과 같이 구할 수 있다. 신뢰수준은 95%를 사용한다.

```
In [9]: from scipy import stats
```

```
In [10]: cube_30 = cube_experiment(30)
stats.norm.interval(0.95, loc=cube_30.mean(), scale=cube_30.std()/n
p.sqrt(30))
```

Out[10]: (0.65357189179169484, 1.2797614415416385)