금융 통계학 기본(3) – 확률분포

이승준 fb.com/plusjune

히스토그램

- [요약통계] Summary statistics 는 데이터 집합의 특징을 간결하게 표현해 주지만, 데이터 집합을 정확하게 해석하기 힘들다.
- 데이터 집합 전체를 제대로 이해하기 위해서는 데이터가 어떤 모습으로 흩어져 있는지 살펴볼 필요가 있다. 즉 데이터가 어떻게 "분포"되어 있는지 살펴볼 필요가 있다.
- 가장 대표적인 것이 [도수분포표]이며, 이를 차트로 표현한 것이 [히스토그램]

데이터의 개수 세기

- 데이터의 개수를 세는 가장 간단한 방법은 딕셔너리를 사용하는 것
- 딕셔너리의 get(x, n) 메소드는 키(key) 값을 반환하며 키가 없으면 n을 반환

```
import numpy as np

data = np.random.randint(0, 10, 100)

h = {}
for x in data:
    h[x] = h.get(x, 0) + 1
```

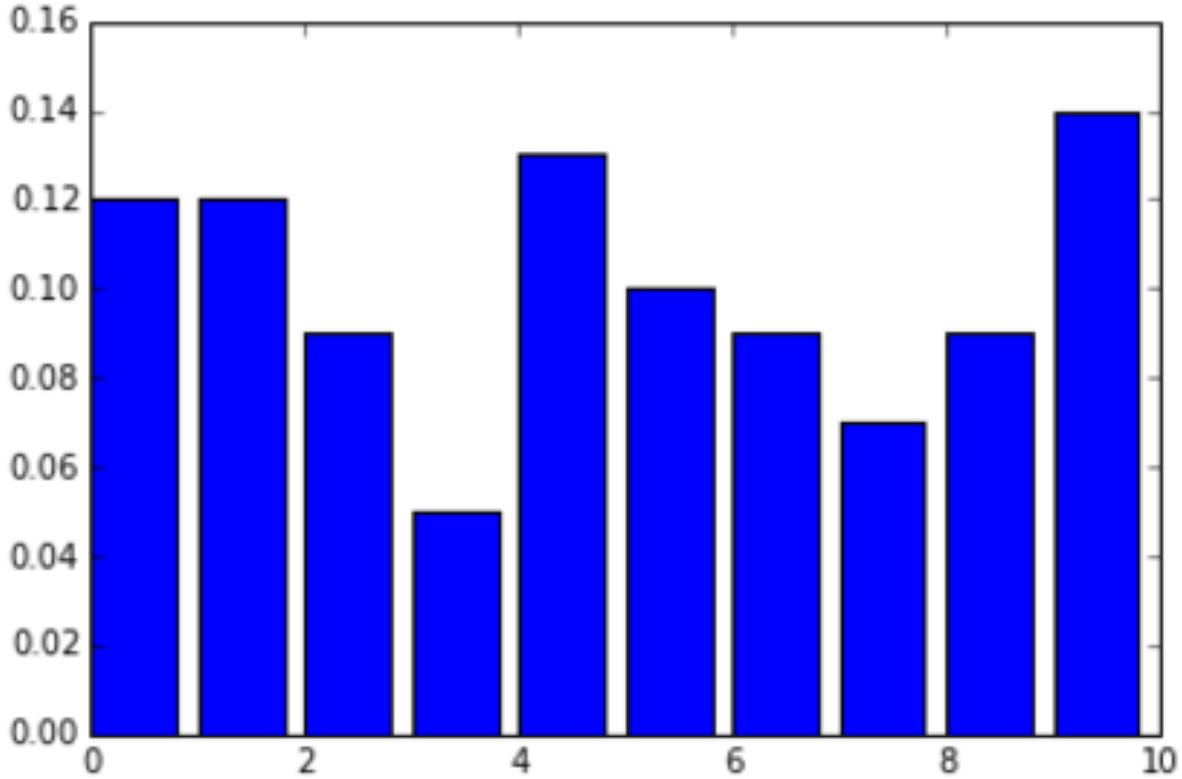
히스토그램 만들기

- 계산된 h 가 바로 히스토그램
- 0~9까지 각 값의 빈도(frequency)를 갖는다.
- 각 빈도값을 전체 데이터 개수로 나누어 주면 각 값이 발생하는 확률

```
n = len(data) # 데이터의 개수
p = {}
for i, freq in h.items():
    p[i] = freq / n
```

```
{0: 0.12,
 1: 0.12,
 2: 0.09,
 3: 0.05,
 4: 0.13,
 5: 0.1,
 6: 0.09,
 7: 0.07,
 8: 0.09,
 9: 0.14}
```

```
import matplotlib.pyplot as plt
plt.bar(p.keys(), p.values())
```



Numpy 히스토그램

- histogram() 함수: 시리즈 데이터의 히스토그램을 계산해 주는 함수.
- (hist, bins)을 반환, 히스토그램의 개수와 구간 배열 (각 구간을 cell 또는 bin 이라고 함)
- 반환되는 bins의 길이는 hist 개수 + 1

```
import numpy as np

count, bins = np.histogram(data, bins = [0, 5, 10])

print (count)
print (bins)
```

pandas_datareader

	Open	High	Low	Close	Volume	Adj Close	Ret
Date							
2013-01-01	1522000	1522000	1522000	1522000	0	1483035.32	NaN
2013-01-02	1533000	1576000	1527000	1576000	228900	1535652.87	0.034865
2013-01-03	1582000	1584000	1543000	1543000	284500	1503497.70	-0.021161
2013-01-04	1540000	1542000	1510000	1525000	259900	1485958.52	-0.011734
2013-01-07	1515000	1528000	1500000	1520000	252200	1481086.53	-0.003284

```
import pandas as pd

from datetime import datetime
from pandas_datareader import data, wb

start = datetime(2013, 1, 1)
end = datetime(2013, 12, 30)

df = data.get_data_yahoo("005930.KS", start, end)
df['Ret'] = np.log( df['Adj Close'] / df ['Adj Close'].shift(1) )
```

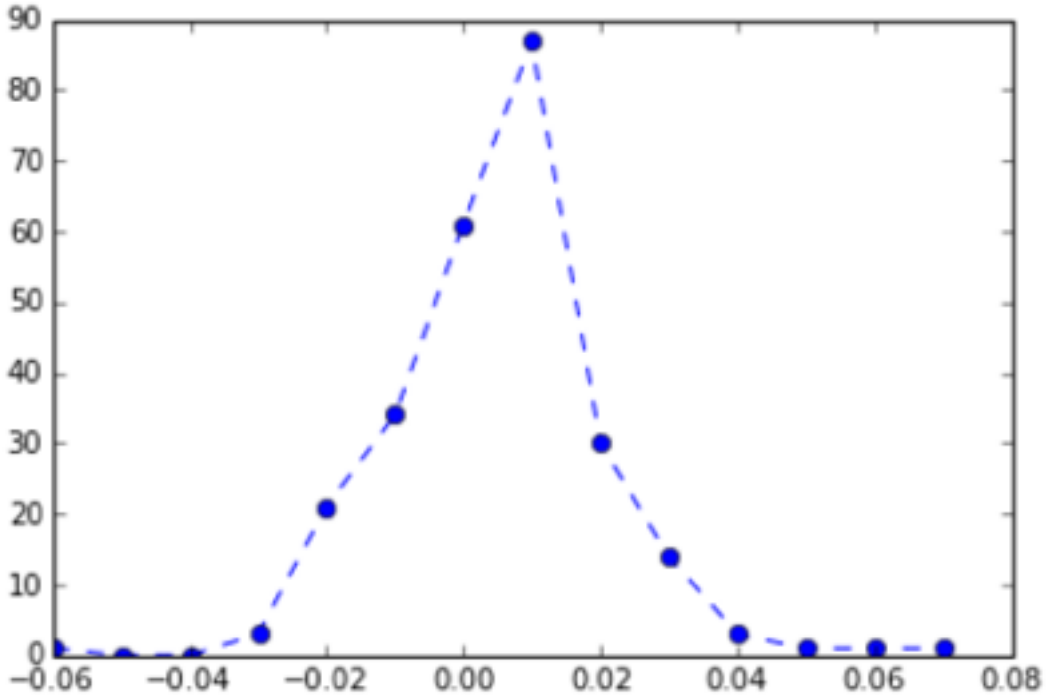
DataFrame.describe()

- 자주 사용하는 통계 요약 값들을 반환
- 개수, 평균, 표준편차, 최소, 최대, 사분위 값
- 값을 각각 얻을 수 도 있다.

```
df['Ret'].describe()

count     257.000000
mean       -0.000364
std         0.015108
min        -0.063794
25%        -0.009375
50%         0.000000
75%         0.007122
max         0.060415
Name: Ret, dtype: float64
```

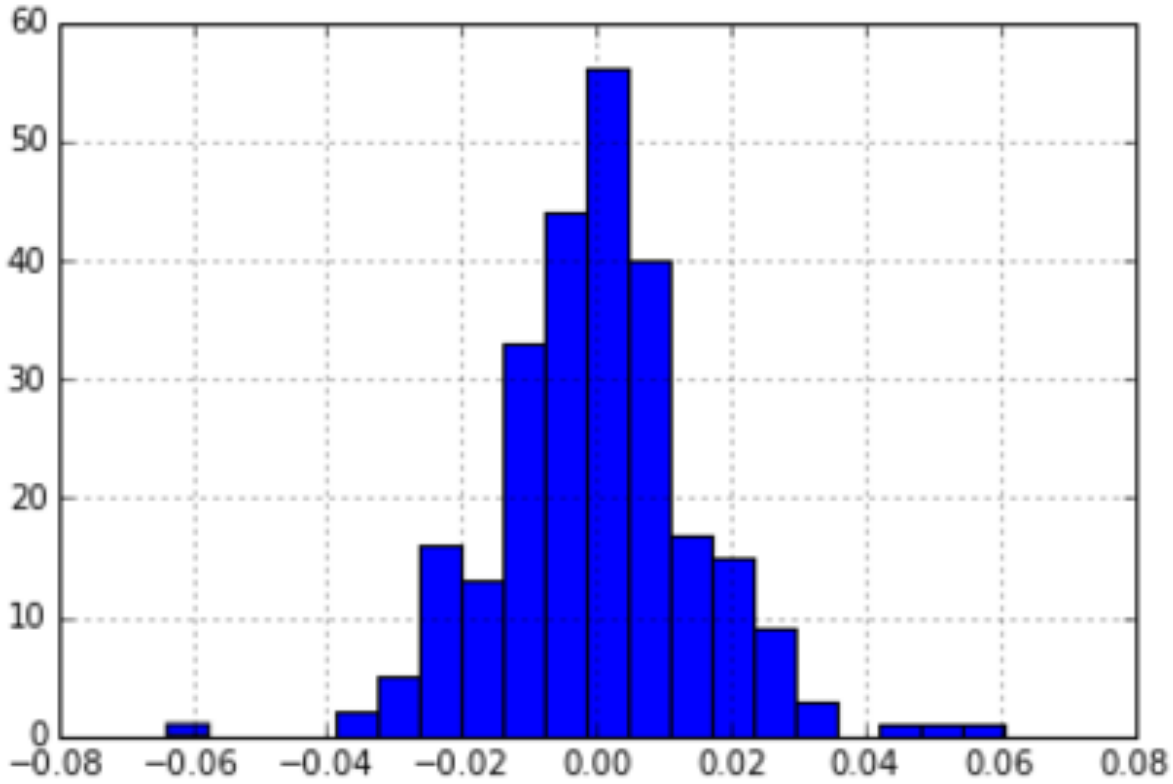
히스토그램 그리기 (1)

```
count, bins = np.histogram(df['Ret'], bins = np.arange(-0.07, +0.07, 0.01))
plt.plot(bins[1:], count, 'o--')
```



히스토그램 그리기 (2)

```
df['Ret'].hist(bins=20, normed=False)
```



정규분포

$$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

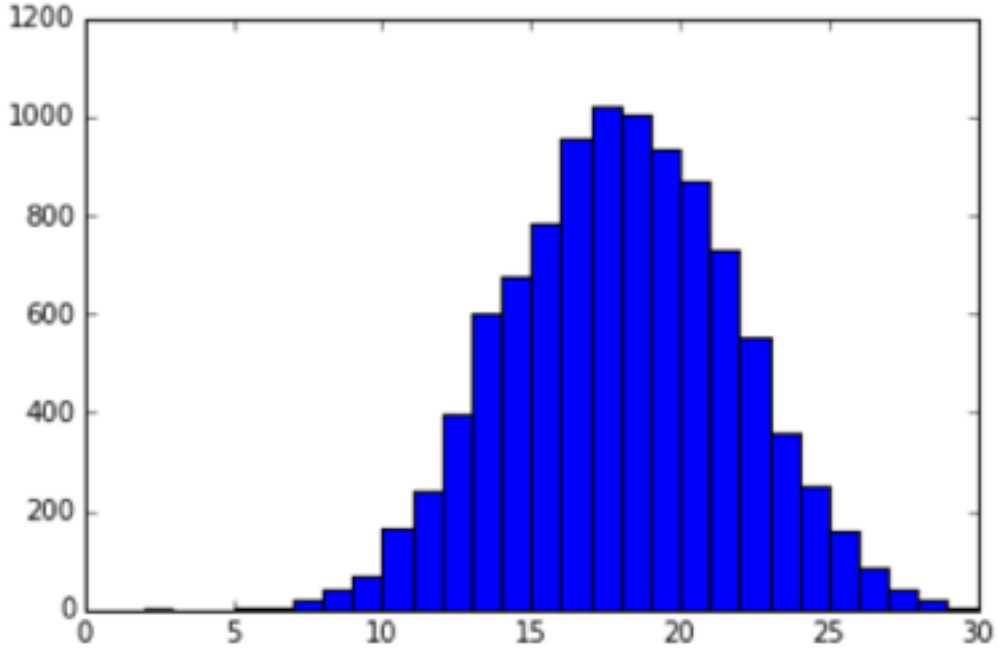
정규분포(Normal Distribution, 가우스 분포)는 연속확률분포의 하나

- 수집된 자료의 분포를 근사하는 데에 자주 사용
- 2개의 매개 변수 평균 μ과 표준편차σ 에 의해 모양이 결정, 분포를 $\mathrm{N}(\mu, \sigma^2)$로 표기
- 표준정규분포: 평균이 0이고 표준편차가 1인 정규분포 $\mathrm{N}(0,1)$

수익률은 셀수있는 이산확률분포, 하지만 범주가 많아지면 계산이 복잡, 연속확률분포에 근사하여 분석

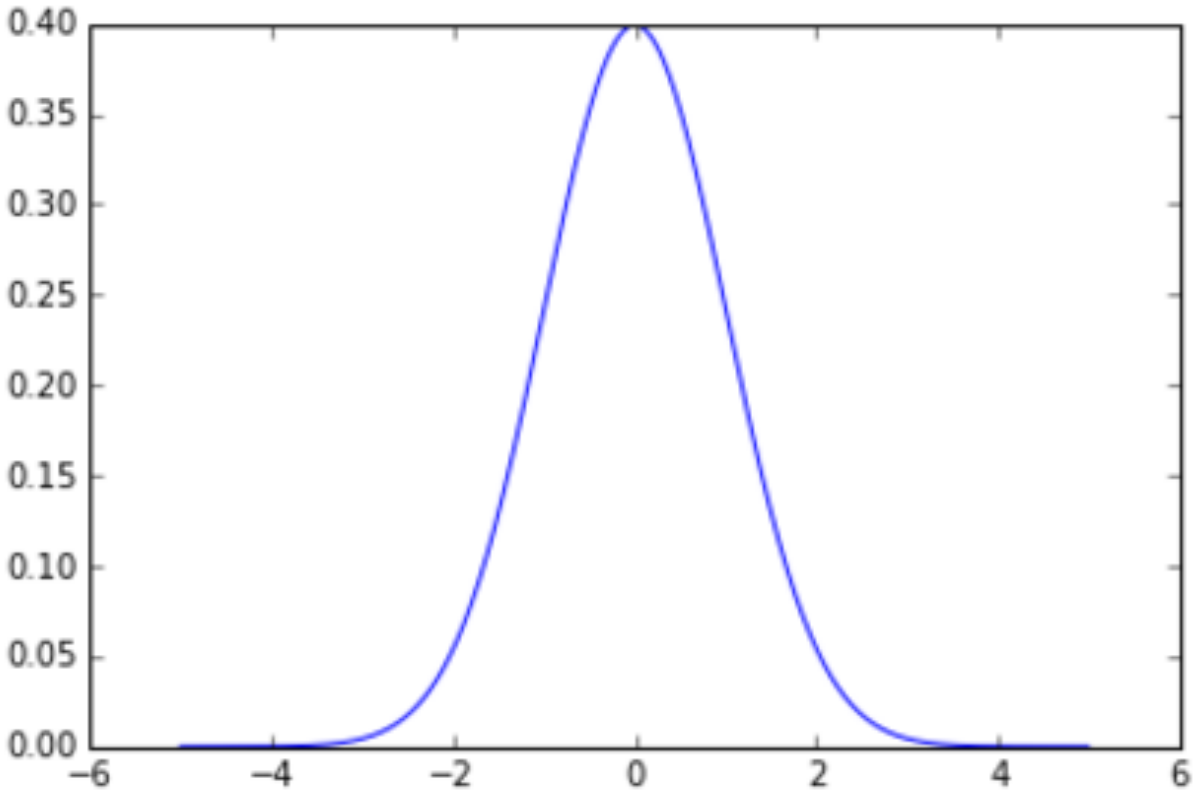
주사위 던지기 시뮬레이션

```
throw_count = 10000 # throw count
dices = 5 # number of dices

data = []
for i in range(throw_count):
    data.append( randint(1, 7, dices).sum() )

bins = range(2, 31) # 2~30

plt.hist(data, bins)
```

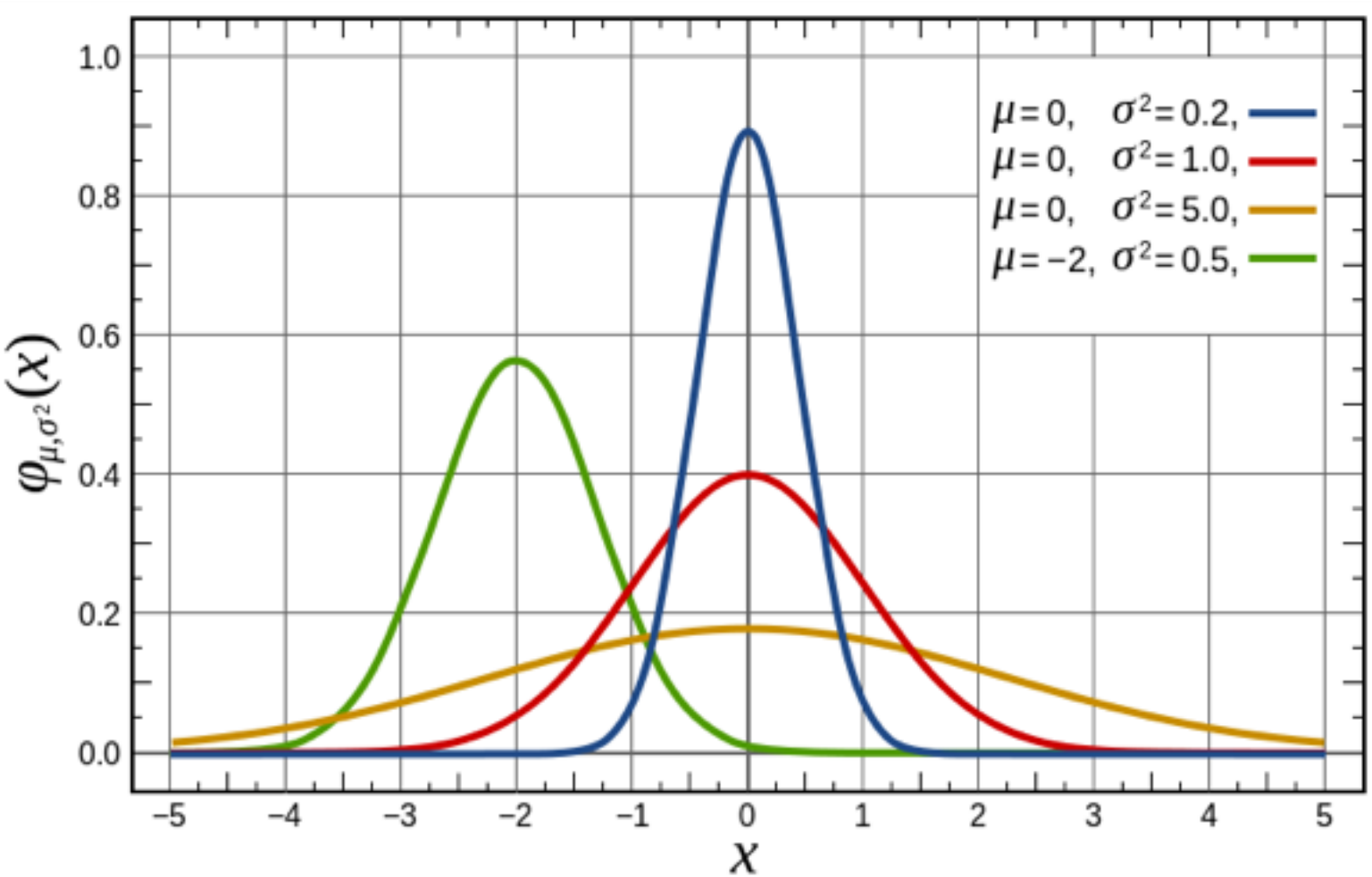


자연과학 현상, 사회과학 현상을 분석할 때 가장 빈번하게 활용되는 확률분포
실험이나 관찰을 통하여 수집된 자료의 확률분포는 대부분 정규분포를 따르기 때문.

```
r = np.arange(-5, 5, 0.01)
mu, sigma = 0, 1
plt.plot(r, stats.norm.pdf(r, mu, sigma))
```



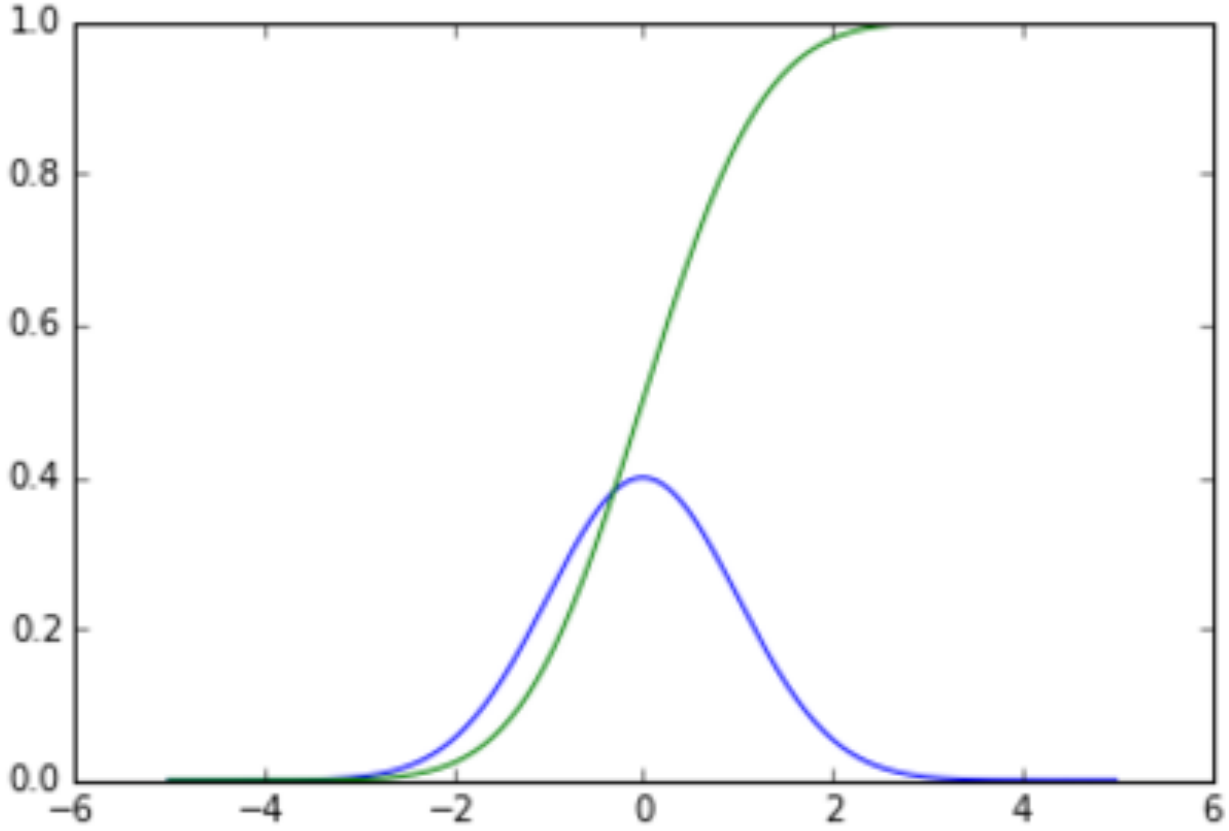
- 평균 μ과 표준편차σ 값에 따른 정규분포 $\mathrm{N}(\mu, \sigma^2)$의 모양

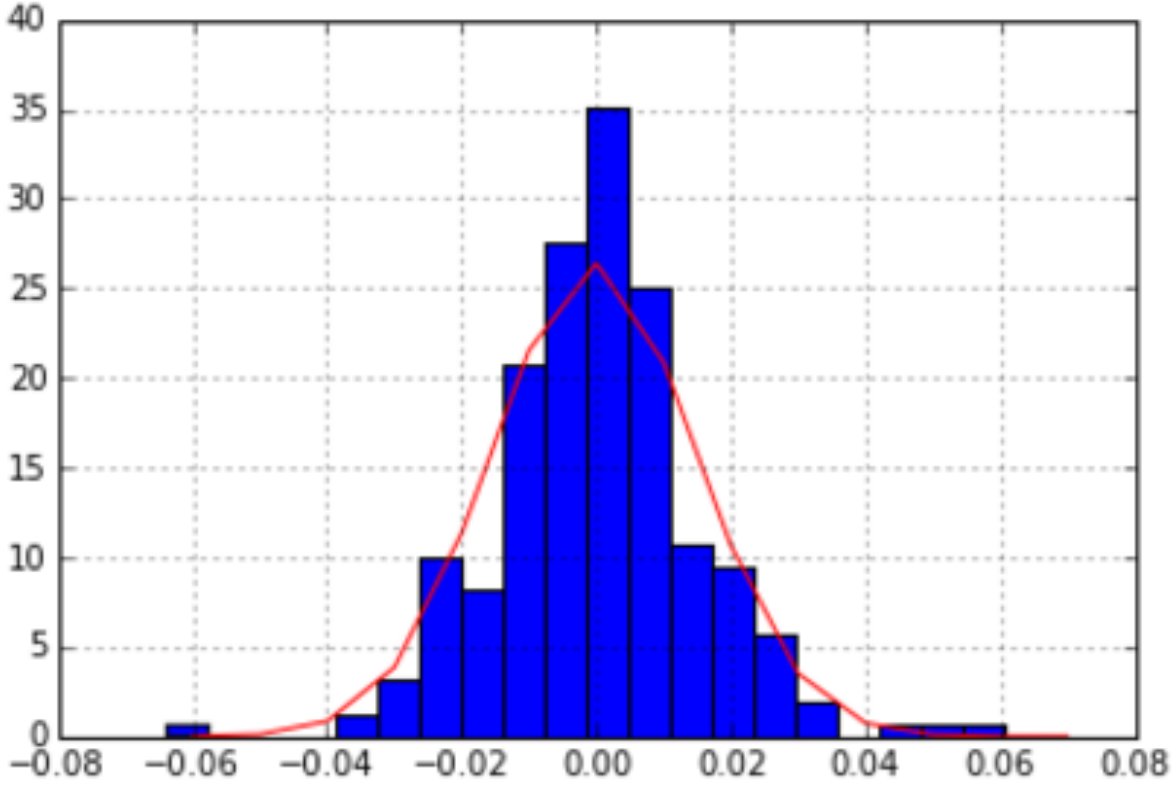


scipy.stats

- norm.pdf(x, mu, sigma)
- norm.cdf(x, mu, sigma)

```
from scipy.stats import norm

x = np.arange(-5, 5, 0.01)

Mu, sigma = 0, 1

plt.plot(x, norm.pdf(x, mu, sigma))
plt.plot(x, norm.cdf(x, mu, sigma))
```






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

df['Ret'].hist(bins=20, normed=True)

mu = df['Ret'].mean()
var = df['Ret'].var()
sigma = np.sqrt(var)
plt.plot(bins[1:], stats.norm.pdf(bins[1:], mu, sigma), color='r')
```