

# Matplotlib

2019.10



# 1. Matplotlib 란?

## ■ 개요

- 파이썬에서 자료를 시각화(Visualization)해주는 패키지
- Pandas 시리즈와 데이터프레임의 plot 메소드 활용

## ■ 그릴 수 있는 Chart / Plot

- 라인 플롯(line plot)
- 산점도(scatter plot)
- 막대 그래프(bar chart), 히스토그램(histogram)
- 박스 플롯(box plot)
- 파이 차트(pie chart)
- 컨투어 플롯(contour plot), 서피스 플롯(surface plot)

## ■ 참고 사이트

- 갤러리(<http://matplotlib.org/gallery.html>)

## 2. 한글 설정

```
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm
import numpy as np
import pandas as pd

mpl.rcParams['axes.unicode_minus'] = False    # minus 표시

[(f.name, f.fname) for f in fm.fontManager.ttflist if 'Malgun' in f.name]
[('Malgun Gothic', 'C:\\Windows\\Fonts\\malgunbd.ttf'),
 ('Malgun Gothic', 'C:\\Windows\\Fonts\\malgun.ttf')]

plt.rcParams["font.family"] = 'Malgun Gothic'
plt.rcParams["font.size"] = 10

#####
mpl.rc('axes', unicode_minus=False)
mpl.rc('font', family='Malgun Gothic')
```

### 3. Line Plot

- 데이터가 시간, 순서 등에 따라 어떻게 변화하는지 보여주기 위해 사용
- Series 타입에서 value가 y축, index가 x축

```
x = [0, 1, 2, 3]
y = [0, 1, 4, 9]
plt.plot(x, y)
plt.show()
```

```
s = pd.Series(np.random.randn(10).cumsum(), index=np.arange(10))
plt.plot(s)          # s.plot()
```

```
df = pd.DataFrame(np.random.randn(10, 4).cumsum(axis=0),
                  columns=(['A', 'B', 'C', 'D']), index=(np.arange(0,100,10)))
```

```
plt.plot(s, color='g', marker='o', linestyle='-')
plt.plot(s, 'bs--')
```

color: blue, green, red, cyan, magenta, yellow, black(k), white  
marker: circle(o), 삼각형(^), 역삼각형(v), 사각형(s), 별(\*), 플러스(+)  
linestyle: 직선(-), dashed(--), dash-dot(-.), dotted(:)

## 4. 산점도(Scatter Plot)

- Anscombe's quartet: 기술 통계치가 거의 같은 4가지 데이터 셋
- x와 y의 관계를 눈으로 확인하고자 할 때

```
x = np.random.randn(100)
y = np.random.randn(100)
plt.scatter(x, y)
plt.show()
```

```
x = np.random.rand(100)
y = np.random.rand(100)
colors = np.random.rand(100)
area = np.pi * (15 * np.random.rand(100))**2
plt.scatter(x, y, label='Samples', s=area, c=colors, alpha=0.5)
s: 도형의 크기(size)
c: 도형의 색상(color)
alpha: 투명도 (0-완전 투명, 1-완전 불투명)
```

```
x1 = np.random.normal(1, 1, size=(100, 1))
x2 = np.random.normal(-2, 4, size=(100, 1))
plt.scatter(x1, x2, color='r', marker='s')
```

## 5. 막대 그래프(Bar chart), 히스토그램(histogram)

- 특정 그룹의 데이터의 추세와 정량적인 분포

```
s2 = pd.Series(np.random.rand(10), index=list('abcdefghij'))
s2.plot(kind='bar')          # plt.plot(s2, kind='bar')
# plt.show()
```

```
df2 = pd.DataFrame(np.random.rand(6, 4),
                    index = ['one', 'two', 'three', 'four', 'five', 'six'],
                    columns = pd.Index(['A', 'B', 'C', 'D'], name='Genus'))
df2.plot(kind='barh')
df2.plot(kind='barh', stacked=True)
```

```
s3 = pd.Series(np.random.randn(200))
plt.hist(s3)                # s3.hist()
```

```
plt.hist(s3, bins=50)       # default는 10
```

```
plt.hist(s3, bins=100, normed=True)
```

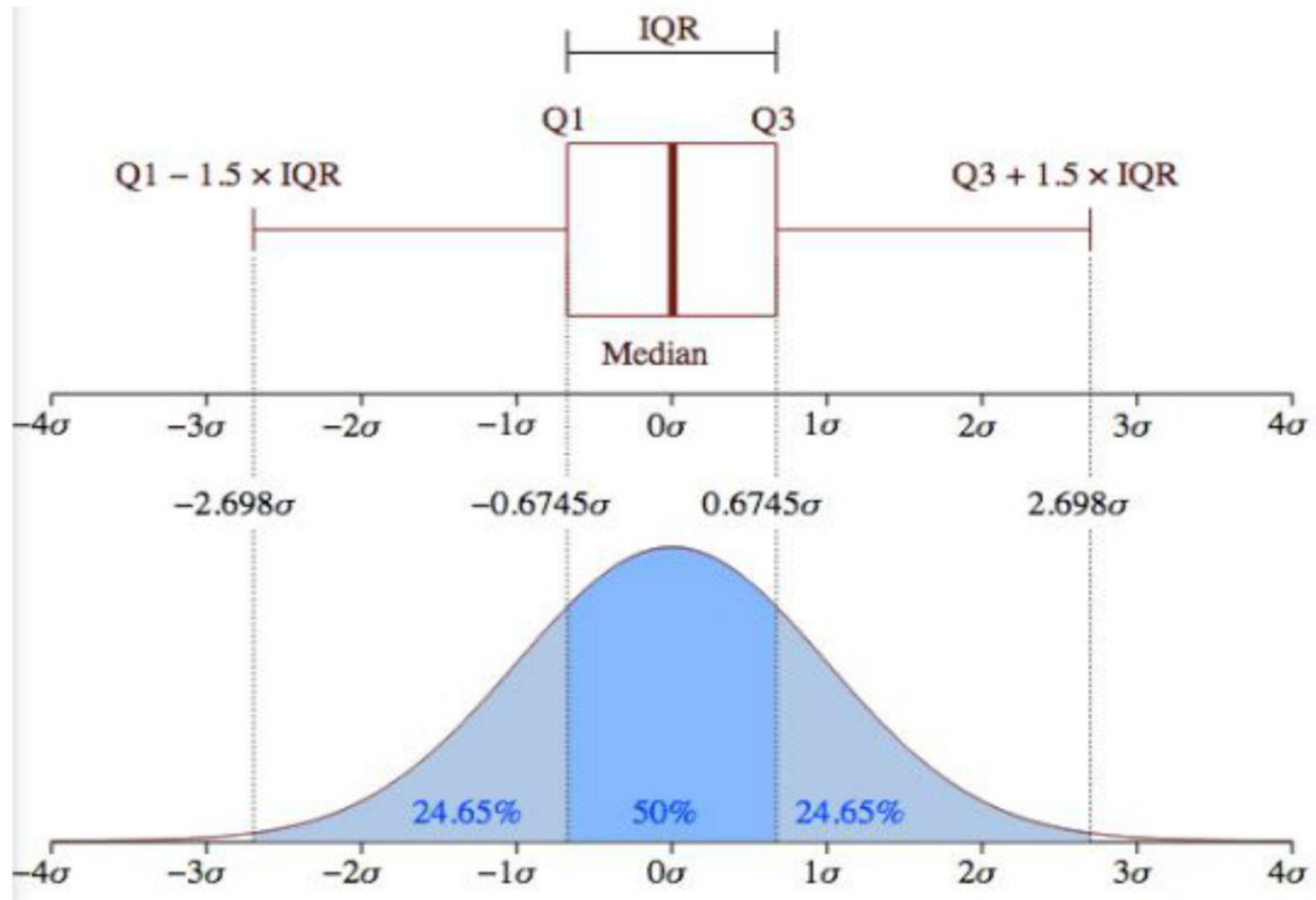
## 6. 박스 플롯(Box plot)

- 기초 통계량 확인하고자 할 때

```
s4 = np.random.randn(1000)
```

```
s5 = np.random.normal(loc=10, scale=2, size=1000)
```

```
plt.boxplot([s4, s5])
```



## 7. 제목, 레이블, 범례 작성

```
import math
xs, ysin, ycos = [], [], []
for x in range(360):
    xs.append(x)
    ysin.append(math.sin(math.pi * x / 180))
    ycos.append(math.cos(math.pi * x / 180))

plt.plot(xs, ysin, label='Sine curve')
plt.plot(xs, ycos, label='Cosine curve')
plt.xlim([-20, 380]);      plt.ylim([-1.2, 1.2])
plt.xlabel('X-Value');      plt.ylabel('Y-Value')
plt.title('삼각함수')
plt.legend(loc='center')    # label을 범례로 표시

plt.axhline(y=0, color='k', linewidth=0.5) # 수평선(x축)
plt.axvline(x=0, color='k', linewidth=0.5) # 수직선(y축)
plt.axvline(x=360, color='k', linewidth=0.5)

xs = np.arange(0, 2*np.pi, 0.01)
ysin = np.sin(xs); ycos = np.cos(xs)
```



## 8. 여러 개의 그래프를 그리는 방법(Subplot)

```
def f(t):  
    return np.exp(-t) * np.cos(2*np.pi*t)  
  
def g(t):  
    return np.sin(np.pi*t)  
  
t1 = np.arange(0.0, 5.0, 0.01)  
t2 = np.arange(0.0, 5.0, 0.01)  
  
plt.subplot(221)                # 2 x 2, 그래프중 첫번째  
plt.plot(t1, f(t1))  
plt.subplot(2, 2, 2)  
plt.plot(t2, g(t2))  
  
plt.subplot(223)  
plt.plot(t1, f(t1), 'r-')  
plt.subplot(224)  
plt.plot(t2, g(t2), 'r-')  
  
plt.show()
```

## 8. 여러 개의 그래프를 그리는 방법(Subplot)

```
import seaborn as sns
ans = sns.load_dataset('anscombe')

ans.shape; ans.head(); ans.tail()
ans.groupby('dataset').describe()

ds1 = ans[ans['dataset'] == 'I'];    ds2 = ans[ans['dataset'] == 'II']
ds3 = ans[ans['dataset'] == 'III'];  ds4 = ans[ans['dataset'] == 'IV']

fig = plt.figure()
axes1 = fig.add_subplot(2, 2, 1);    axes2 = fig.add_subplot(2, 2, 2)
axes3 = fig.add_subplot(2, 2, 3);    axes4 = fig.add_subplot(2, 2, 4)

axes1.plot(ds1['x'], ds1['y'], 'o'); axes2.plot(ds2['x'], ds2['y'], 'o')
axes3.plot(ds3['x'], ds3['y'], 'o'); axes4.plot(ds4['x'], ds4['y'], 'o')

axes1.set_title('Dataset 1');         axes2.set_title('Dataset 2')
axes3.set_title('Dataset 3');         axes4.set_title('Dataset 4')

fig.suptitle("Anscombe's Quartet")
fig.tight_layout()
```

## 9. 다른 시각화 패키지

---

### ■ Seaborn

- 세련된 그래프
- <https://seaborn.pydata.org/examples/index.html>

### ■ Bokeh

- 웹 기반 인터랙티브 플롯
- <https://bokeh.pydata.org/en/latest/>

### ■ Folium

- 지도데이터에 'Leaflet.js'를 이용하여 위치정보를 시각화
- 웹에서 사용 가능