Python – 시각화

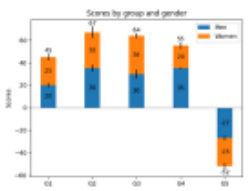
Index

MatplotHib

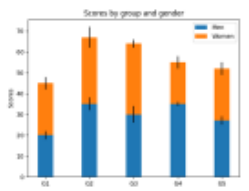
파이썬 시각화 라이브러리

Matplotlib

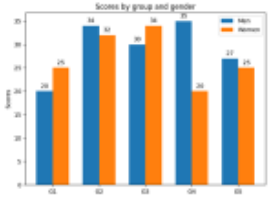
- 파이썬의 가장 기본적인 데이터 시각화 패키지
- https://matplotlib.org
- pip install matplotlib



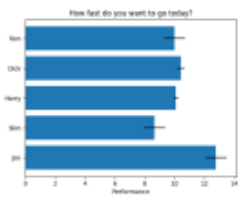
Bar Label Demo



Stacked bar chart



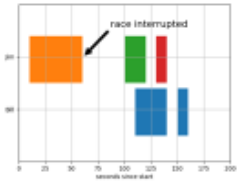
Grouped bar chart with labels



Horizontal bar chart

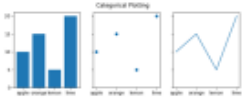
기타 시각화 라이브러리

seaborn

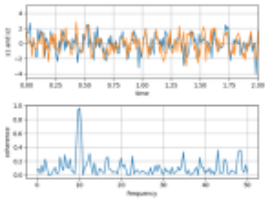


Broken Barh

CapStyle



Plotting categorical variables



Plotting the coherence of two signals

차트 종류 – https://matplotlib.org/stable/gallery/index.html

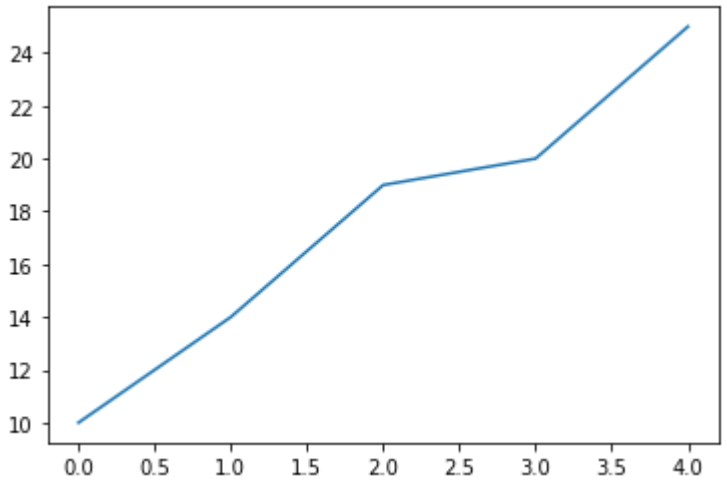
Line Graph

라인그래프

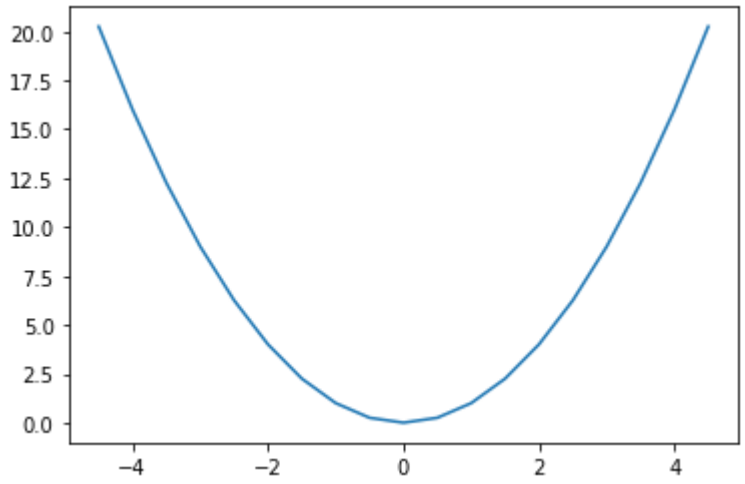
라인그래프 그리기

- 선형 그래프

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

x = [10, 14, 19, 20, 25]
plt.plot(x)
plt.show()
```

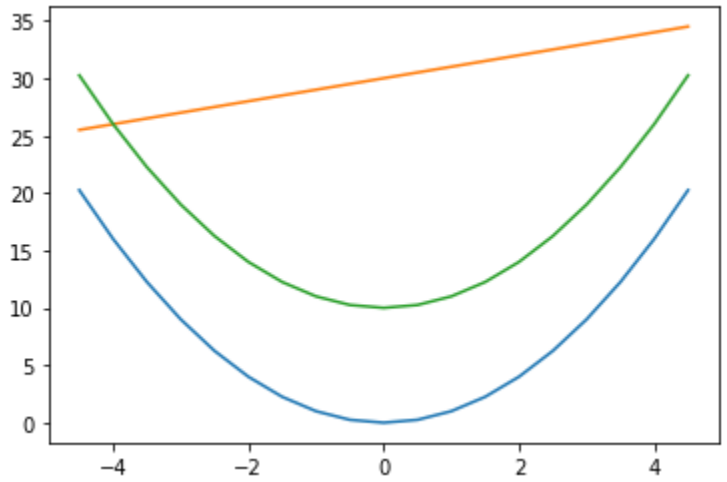



```
x = np.arange(-4.5, 5, 0.5)
y = x**2
plt.plot(x, y)
plt.show()
```



한 번에 여러 그래프 그리기

```
x=np.arange(-4.5,5,0.5)

y1=x**2
y2=x+30
y3=x**2+10

plt.plot(x,y1)
plt.plot(x,y2)
plt.plot(x,y3)

plt.show()
```



Graph Options

그래프 옵션

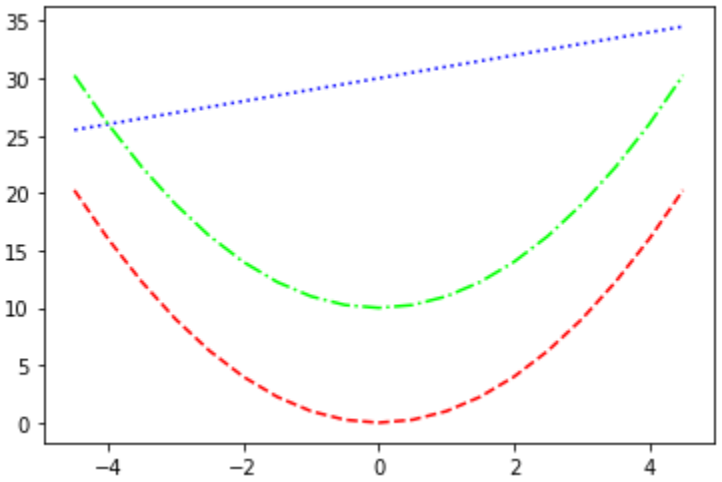
plot(color="", linestyle="")

- 선의 색 조정, color
- color="red" # red, green, blue 등 색 이름
- color="r" # r, g, b, y, m , k 등 색 이름 약자
- color='#ff0000' # rgb 색상코드

- 선 종류, linestyle
- linestyle='sold' # 실선('-')
- linestyle='dashed' # 파선 ('--')
- linestyle='dashdot' # 1점 쇄선 ('-.')
- linestyle='dotted' # 점선 (':')

- 동시에 적용하는 방법
- plt.plot(x, y, ':r') # 붉은 색 점선

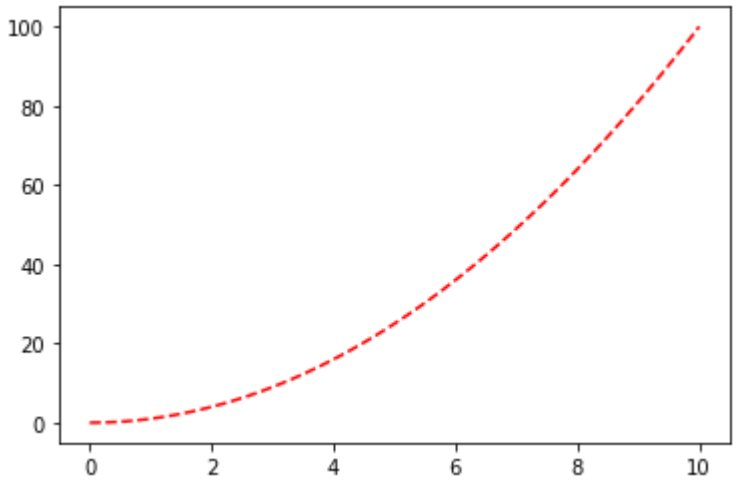
```
x = np.arange(-4.5, 5, 0.5)

y1 = x**2
y2 = x+30
y3 = x**2+10

plt.plot(x, y1, color="red", linestyle="dashed")
plt.plot(x, y2, "b:")
plt.plot(x, y3, color='#00FF00', linestyle="-.")

plt.show()
```

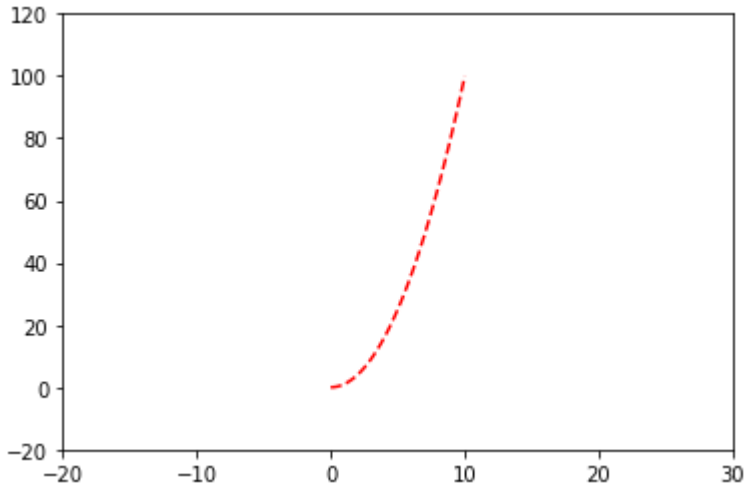


축의 범위 조정

- x축 범위 조정 xlim(시작, 끝)
- y축 범위 조정 ylim(시작, 끝)
- axis([x시작, x끝, y시작, y끝])

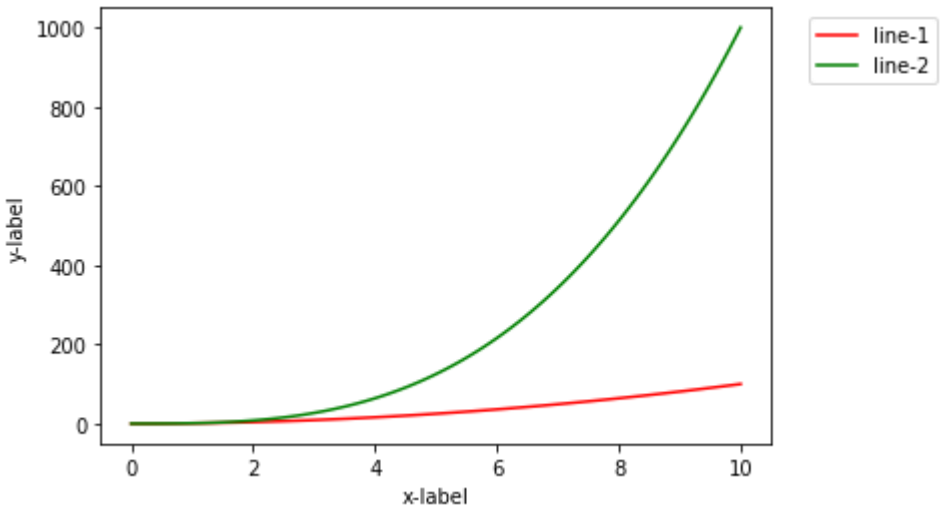
```
x = np.linspace(0,10,100)
y = x**2

plt.plot(x, y, '--r')
plt.xlim(-20, 30)
# plt.ylim(-20, 120)
# plt.axis([-20, 30, -20, 120])

plt.show()
```





Label & legend

- x축 범위 조정 xlim(시작, 끝)
- y축 범위 조정 ylim(시작, 끝)
- axis([x시작, x끝, y시작, y끝])



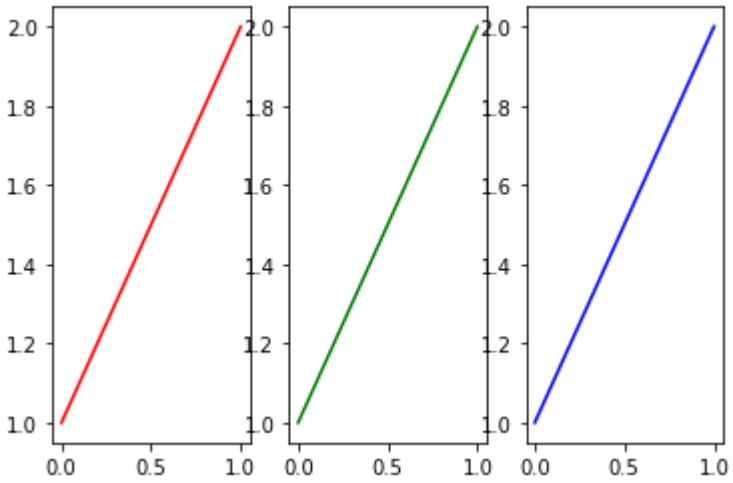

```
x = np.linspace(0,10,100)

y1 = x**2
y2 = x**3

# 선라벨
plt.plot(x, y1, 'r', label='line-1')
plt.plot(x, y2, 'g', label='line-2')

# x축,y축라벨
plt.xlabel('x-label')
plt.ylabel('y-label')

# plt.legend(loc=2) # legend 위치 (그래프 내)
plt.legend(bbox_to_anchor=(1.04,1)) # legend 위치(그래프 밖)
plt.show()
```

legend 위치 별 코드 – 사이트 내 표 참고
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.legend.html

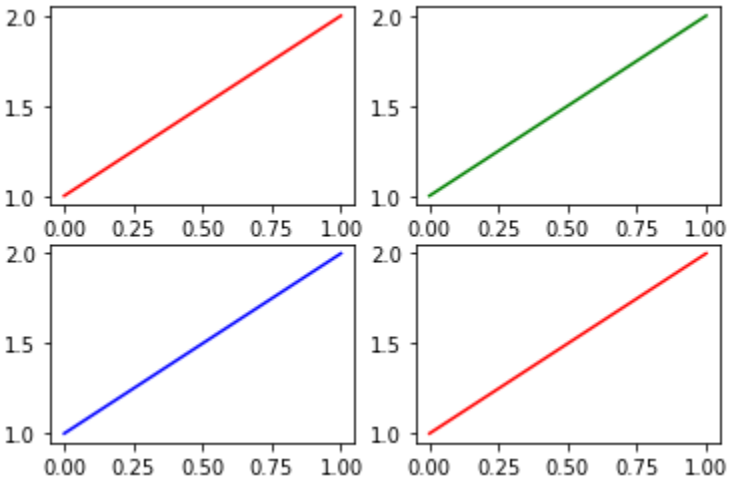
Subplot

- 한 번에 여러 개의 차트를 표시
- plt.subplot(행열순번)

```
plt.subplot(131)
plt.plot([1,2],color='r')

plt.subplot(132)
plt.plot([1,2],color='g')

plt.subplot(133)
plt.plot([1,2],color='b')

plt.show()
```



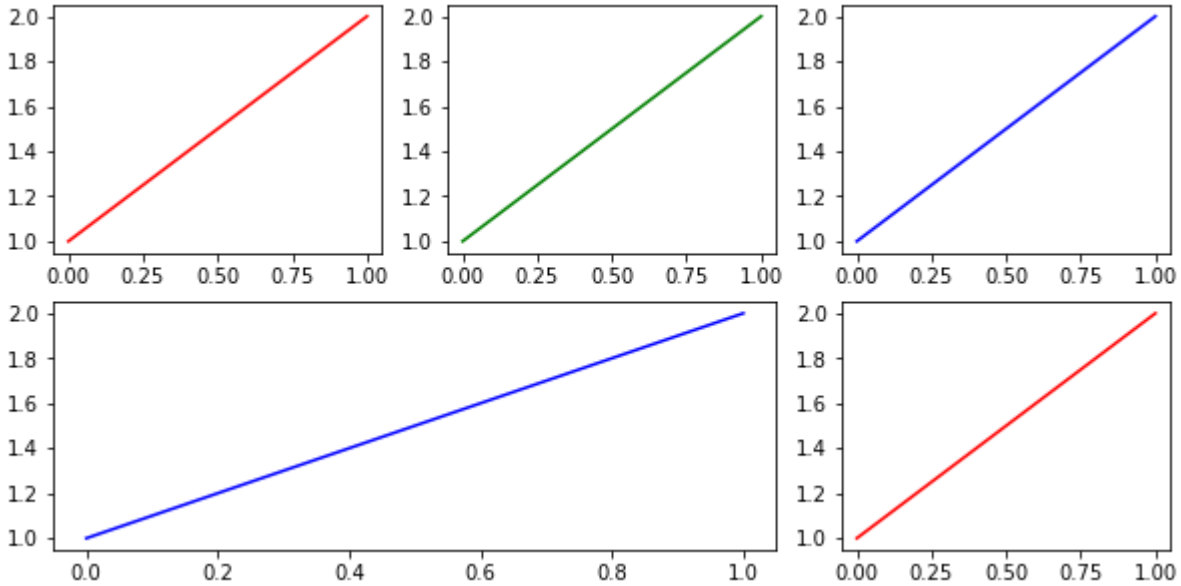

```
plt.subplot(221)
plt.plot([1,2],color='r')

plt.subplot(222)
plt.plot([1,2],color='g')

plt.subplot(223)
plt.plot([1,2],color='b')

plt.subplot(224)
plt.plot([1,2],color='r')

plt.show()
```



Subplot & grid

- plt.figure(figsize=(10,5)) # 그래프 크기 조정
- grid = plt.GridSpec(2, 3) # 행, 열

```
plt.subplot(grid[0,0])
plt.plot([1,2],color='r')

plt.subplot(grid[0,1])
plt.plot([1,2],color='g')

plt.subplot(grid[0,2])
plt.plot([1,2],color='b')

plt.subplot(grid[1,:2])
plt.plot([1,2],color='b')

plt.subplot(grid[1,2])
plt.plot([1,2],color='r')

plt.show()
```

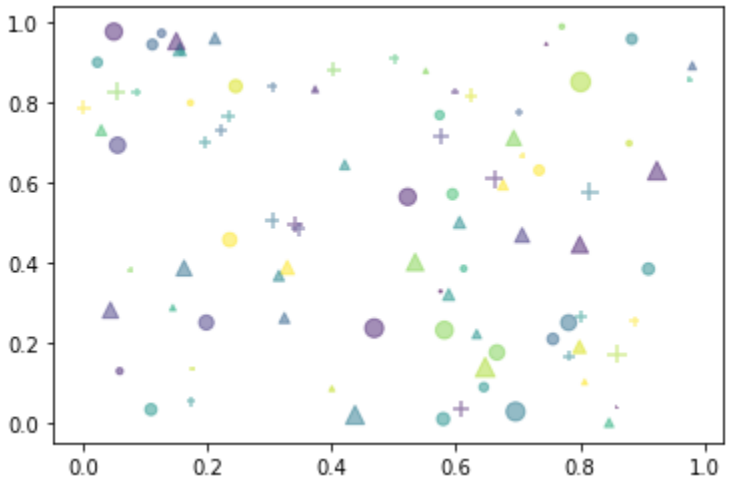


Other Graphs

기타 그래프

Scatter (산점도)

```
x=np.random.rand(30)
y=np.random.rand(30)

a=np.random.rand(30)
b=np.random.rand(30)

e=np.random.rand(30)
f=np.random.rand(30)

color=np.random.rand(30)

size=100*np.random.rand(30)

plt.scatter(x,y,s=size,c=color,alpha=0.5,marker='o')
plt.scatter(a,b,s=size,c=color,alpha=0.5,marker='^')
plt.scatter(e,f,s=size,c=color,alpha=0.5,marker='+')

plt.show()
```

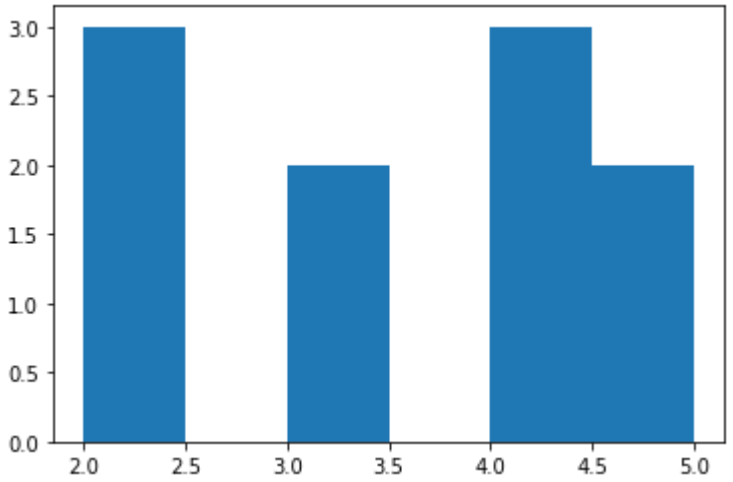


c = 색상
s = 사이즈
alpha = 투명도
marker = 모양

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.scatter.html

Histogram (히스토그램)

```
data=[np.random.randint(1,7) for i in range(10)]

plt.hist(data,bins=6,alpha=1)

plt.show()
```



https://matplotlib.org/stable/gallery/statistics/hist.html

Bar (바차트)

```
labels = ['10대','20대','30대','40대','50대']
plt.bar(labels, [np.random.randint(1,50) for i in range(5)])

# 세로그래프
plt.barh(labels,[np.random.randint(1,50) for i in range(5)], color='g')

plt.title('title')
plt.xlabel('age')
plt.ylabel('value')
plt.show()
```