

21강

matplotlib

21-1 matplotlib 기본

- 데이터 시각화 작업은 데이터 분석할 때 매우 중요하다.
 - ✓ 특이 값을 찾아낼 수 있다.
 - ✓ 데이터 변형이 필요한 지점을 확인할 수 있다.
- matplotlib는 데이터를 시각화하기 쉽게 구성된 파이썬의 대표적인 라이브러리이다.
- matplotlib는 다양한 그래프를 제공하며 pandas의 자료구조와 연동할 수 있는 기능을 가지고 있다.
- matplotlib는 다양한 GUI 백 엔드를 지원하고 있으며 pdf, svg, jpg, png, gmp, gif 등 일반적으로 널리 사용되는 포맷으로 도표를 저장할 수 있다.

21-1 matplotlib 기본

● 기본 그래프

- ✓ matplotlib는 그래프의 설정하기 위해 pyplot의 다양한 함수를 이용할 수 있다.

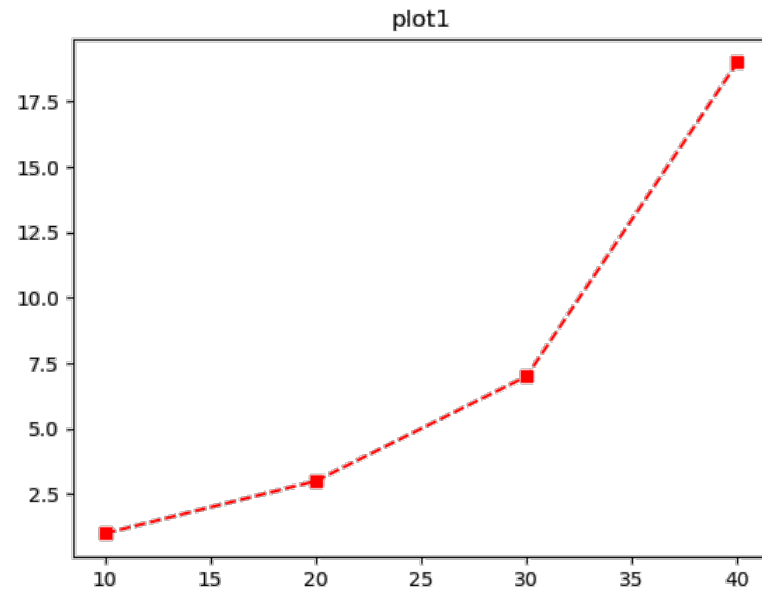
pyplot.title(): 그래프 제목 설정

pyplot.plot(): 선 그래프를 그리기 위한 데이터 정보와 선 그래프의 설정을 할 수 있다.

pyplot.show(): 앞 서 설정한 그래프를 도식화 한다.

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

plt.title('plot1')
plt.plot([10,20,30,40],[1,3,7,19], 'rs--')
plt.show()
```



21-1 matplotlib 기본

● 기본 그래프

<색상표>

문자열	약자
Blue	b
Green	g
Red	r
Cyan	c
Magenta	m
Yellow	y
Black	k
white	w

<선 스타일>

문자열	의미
-	실선 스타일
--	대시선(dashed) 스타일
-.	대시 점선 스타일
:	점선 스타일

<마커>

문자열	의미
.	Point marker
,	Pixel marker
o	Circle marker
v	Triangle down marker
^	Triangle up marker
<	Triangle left marker
>	Triangle right marker
s	Square marker
p	Pentagon marker
*	Star marker
h	Hexagon marker
+	Plus marker
x	X marker

21-1 matplotlib 기본

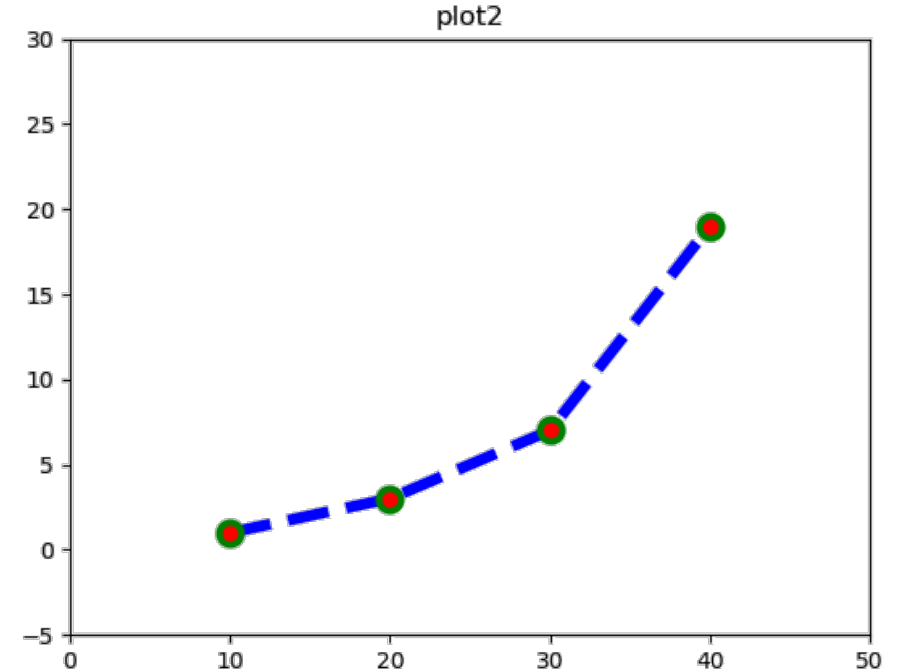
● 기본 그래프

- ✓ 축약 방법 외에도 명시적인 형태로 선그래프를 지정하여 사용할 수 있다.

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

plt.title('plot2')
plt.plot([10,20,30,40],[1,3,7,19], c='b', lw=5, ls='--', marker='o', ms=10,
mec='g', mew='3', mfc='r')
plt.xlim(0, 50)
plt.ylim(-5, 30)
plt.show()
```

문자열	약자	의미
color	c	선 색깔
linewidth	lw	선 굵기
linestyle	ls	선 스타일
marker	m	마커 종류
markersize	ms	마커 크기
markeredgecolor	mec	마커 선 색깔
markeredgewidth	mew	마커 선 굵기
markerfacecolor	mfc	마커 내부 색깔



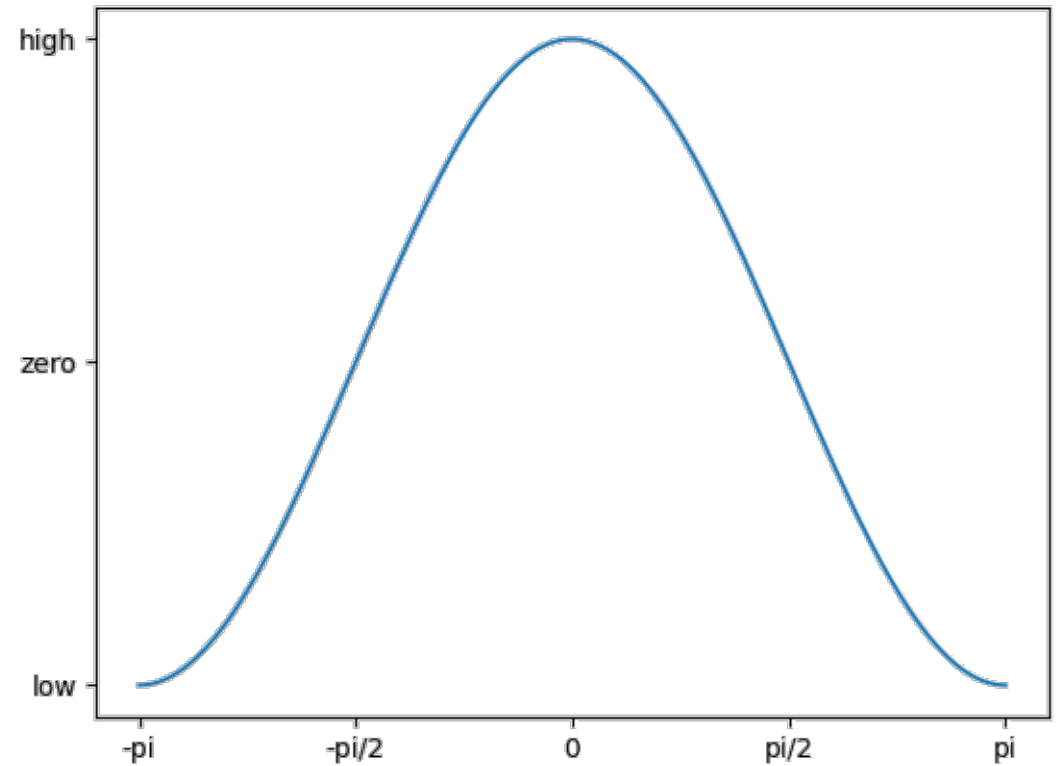
21-1 matplotlib 기본

● 기본 그래프

- ✓ xticks, yticks를 이용하여 tick을 설정할 수 있다.

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

x = np.linspace(-np.pi, np.pi, 256)
y = np.cos(x)
plt.plot(x, y)
plt.xticks([-np.pi, -np.pi / 2, 0, np.pi / 2, np.pi],
           ['-pi', '-pi/2', '0', 'pi/2', 'pi'])
plt.yticks([-1, 0, 1], ['low', 'zero', 'high'])
plt.show()
```



21-2 multi graph

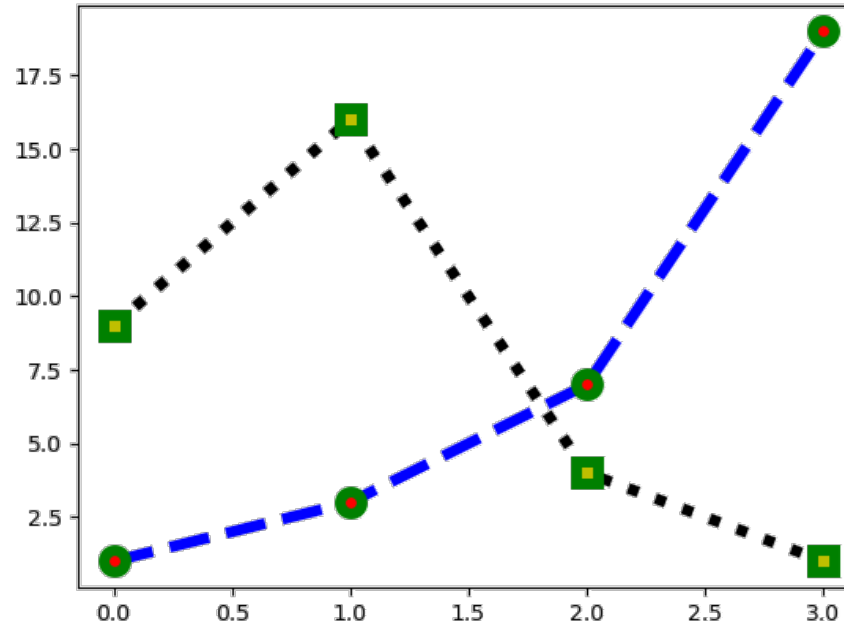
- 한 공간에 그래프 여러 개 그리기

- ✓ 한 공간에 여러 개의 그래프를 그리기 위해서는 그래프 수 만큼 plot 함수를 설정하여 그리면 된다.

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

plt.plot([1,3,7,19], c='b', lw=5, ls='--', marker='o', ms=10,
mec='g', mew=5, mfc='r')

plt.plot([9, 16, 4, 1], c='k', lw=5, ls=':', marker='s', ms=10,
mec='g', mew=5, mfc='y')
plt.show()
```



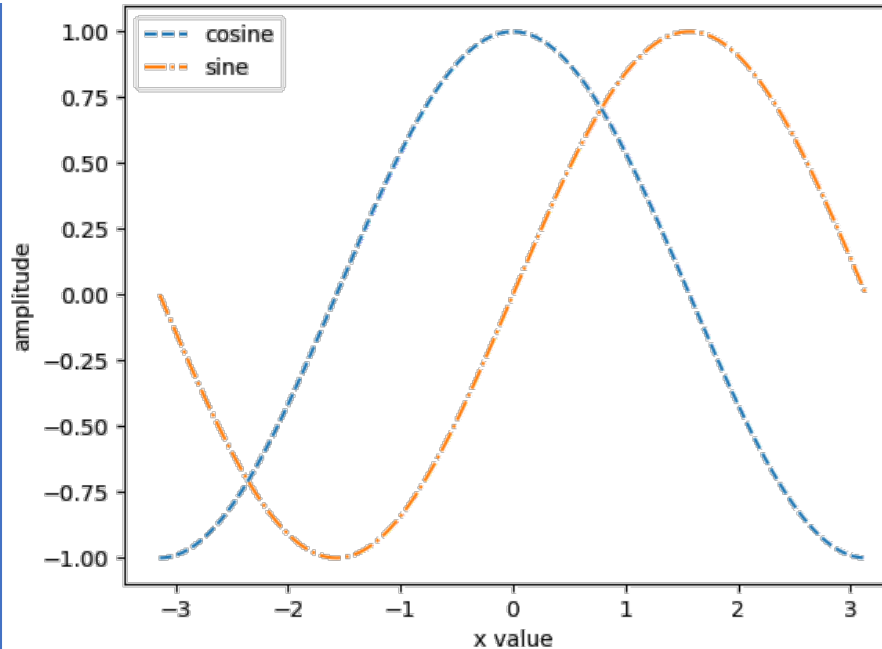
21-2 multi graph

● 범례

- ✓ 여러 개의 그래프를 동시에 그리는 경우에는 각 선이 무슨 자료를 표시하는지를 보여주기 위해 legend 함수를 이용하여 범례를 추가할 수 있다.

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

x = np.linspace(-np.pi, np.pi, 256)
c,s = np.cos(x), np.sin(x)
plt.plot(x, c, ls='--', label='cosine')
plt.plot(x, s, ls='-.', label='sine')
plt.legend(loc=2)
plt.xlabel('x value')
plt.ylabel('amplitude')
plt.show()
```



문자열	숫자
best	0
upper right	1
upper left	2
lower left	3
lower right	4
right	5
center left	6
center right	7
lower center	8
upper center	9
center	10

21-2 multi graph

● 여러 개의 공간에 그래프 그리기

- ✓ 여러 개의 공간에 그래프를 그리기 위해서는 공간을 나누어 설정하여 한다.
- ✓ subplot(nrow, ncol, index)을 이용하여 여러 개의 공간을 나눌 수 있다.

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

x1 = np.linspace(0.0, 3.0)
x2 = np.linspace(0.0, 3.0)
y1 = np.cos(3 * np.pi * x1) * np.exp(-2*x1)
y2 = np.cos(3 * np.pi * x2)

plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'go-')
plt.title('multi graph')
plt.ylabel('result1')

plt.subplot(2, 1, 2)
plt.plot(x2, y2, 'b.-')
plt.xlabel('time')
plt.ylabel('result2')
plt.tight_layout()
plt.show()
```

