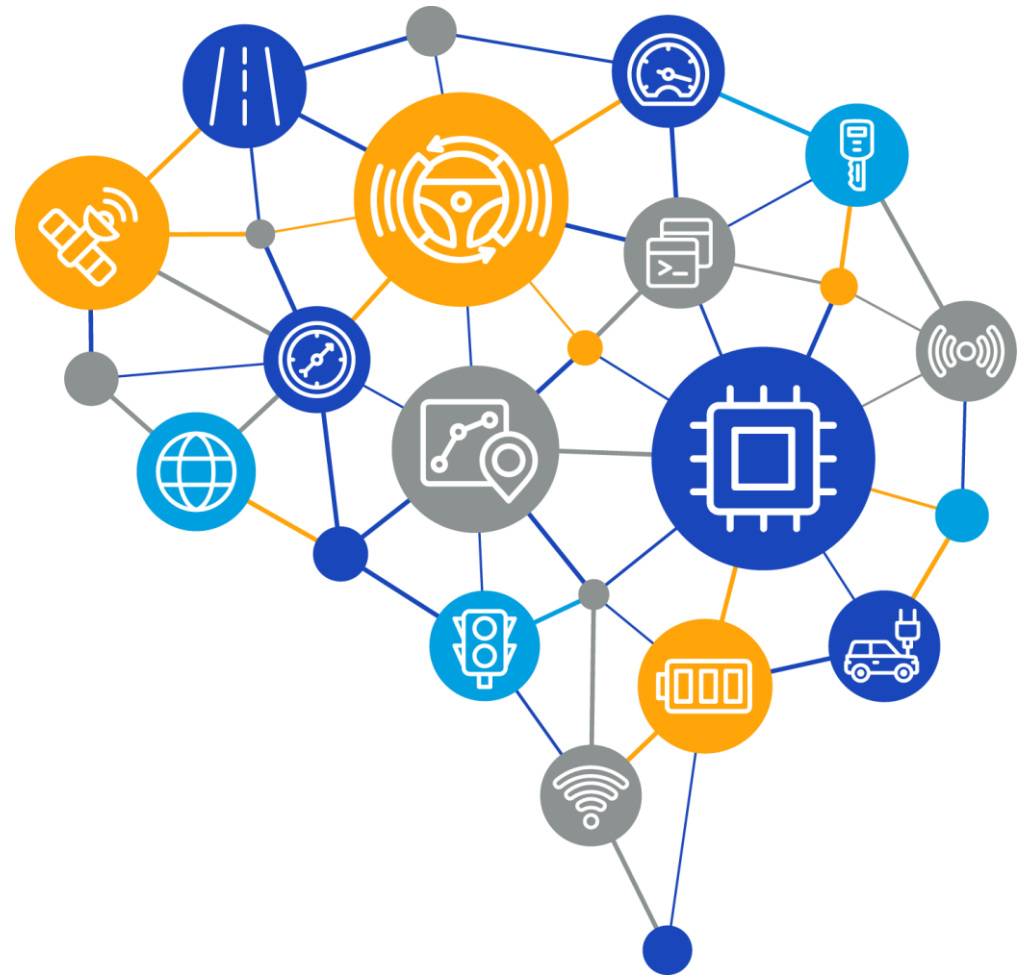


# Matplotlib 중급



강환수 교수

AI Experts  
Who Lead  
The Future

# 01

## Figure와 subplot

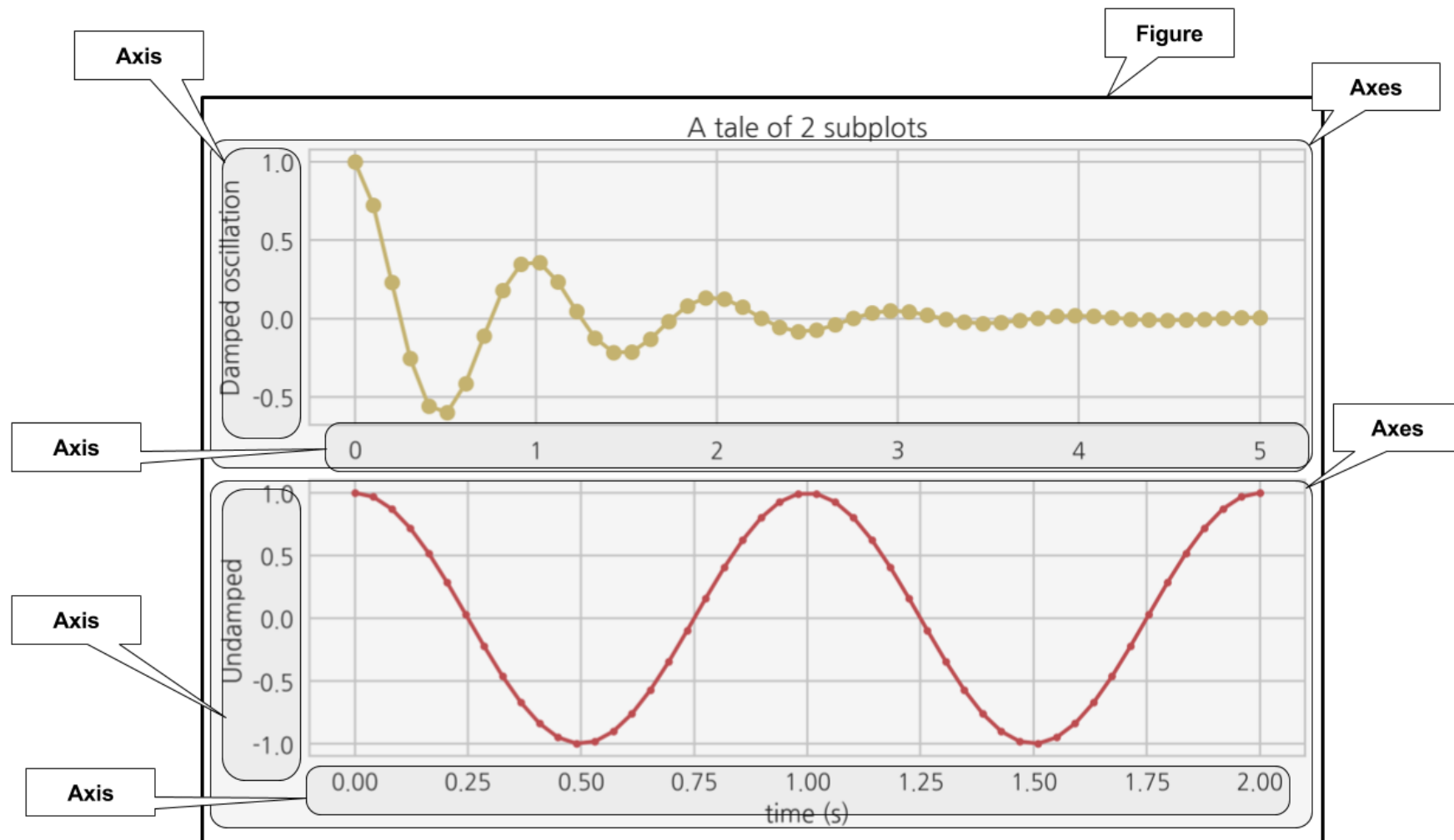
- **Figure 개요**

- 그려지는 그래프가 figure 객체 내에 존재, 그림을 그리는 종이라고 이해
- `figsize=(10, 6)`
  - 그림 크기: `figsize` 인수로 설정(크기와 비율을 지정)
- `fig = plt.figure(figsize=(10, 6))`
  - Figure 생성
  - figure 명령을 사용하여 그 반환 값으로 Figure 객체를 얻음
    - 일반적인 plot 명령 등을 실행하면 자동으로 Figure를 생성해 주기 때문에 일반적으로는 figure 명령을 잘 사용하지 않음
  - figure 명령을 명시적으로 사용하는 경우
    - 여러 개의 윈도우를 동시에 띄워야 하거나(line plot이 아닌 경우)
    - Jupyter 노트북 등에서(line plot의 경우) 그림의 크기를 설정할 경우

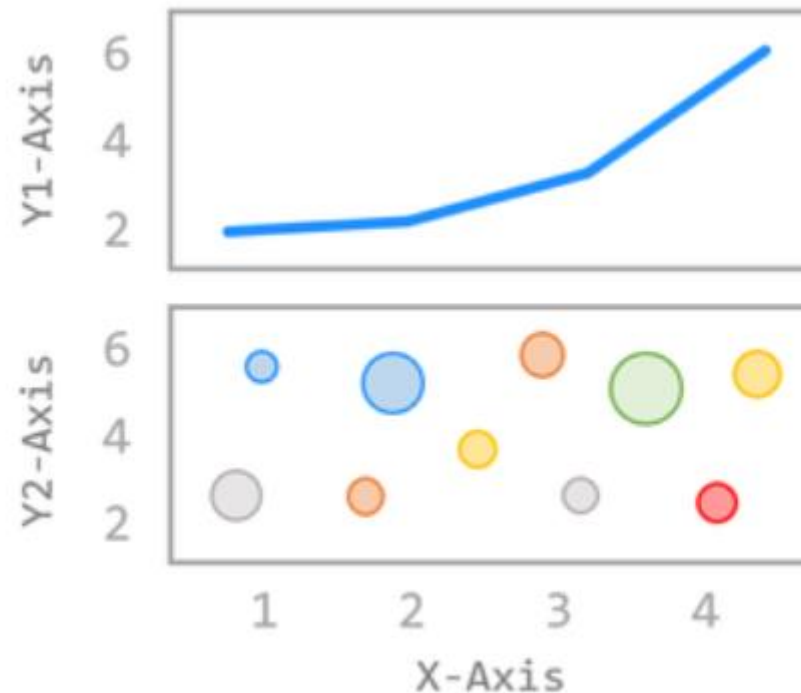
- **show()**

- 시각화 명령을 실제로 차트로 렌더링(rendering)하고 마우스 움직임 등의 이벤트를 기다리라는 지시
- 주피터 노트북에서는 셀 단위로 플롯 명령을 자동 렌더링 해주므로 show 명령이 필요 없음
- 일반 파이썬 인터프리터로 가동되는 경우를 대비하여 항상 마지막에 실행

- Figure는 그림이 그려지는 캔버스나 종이
  - Axes는 하나의 부분 그림
  - Axis는 가로축이나 세로축 등의 축



- `matplotlib.pyplot` 모듈의 `subplot()` 함수
  - 여러 개의 그래프를 하나의 그림에 나타내도록
- 그리드(grid) 형태의 Axes 객체들을 생성
  - Figure 안에 있는 각각의 플롯은 Axes 라고 불리는 객체
    - Figure 내부에 행렬(matrix) 형태의 여러 그림이 있으며 이를 Axes라 함



- - `plt.subplot()`
  - 가장 낮은 수준의 접근 방법이지만 가장 많이 사용되는 방법 중 하나
- - `plt.subplots()`
  - 가장 간단하며 가독성이 좋은 방법
- - `fig.add_subplot()`
  - `figure` 객체로 서브플롯의 접근에 사용

AI Experts  
Who Lead  
The Future

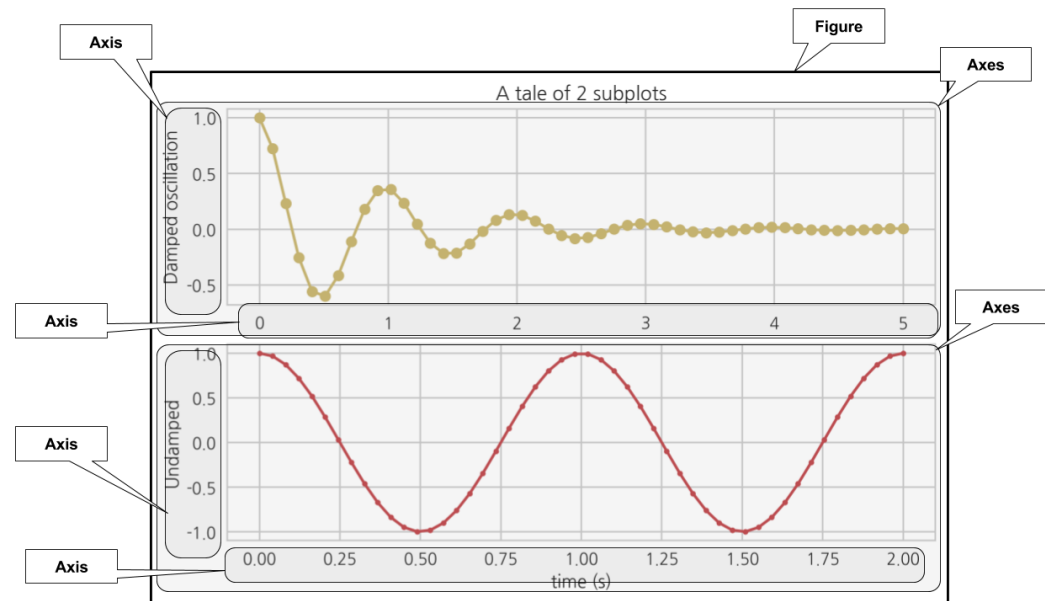
03

`plt.subplot()`

# plt.subplot

Python language

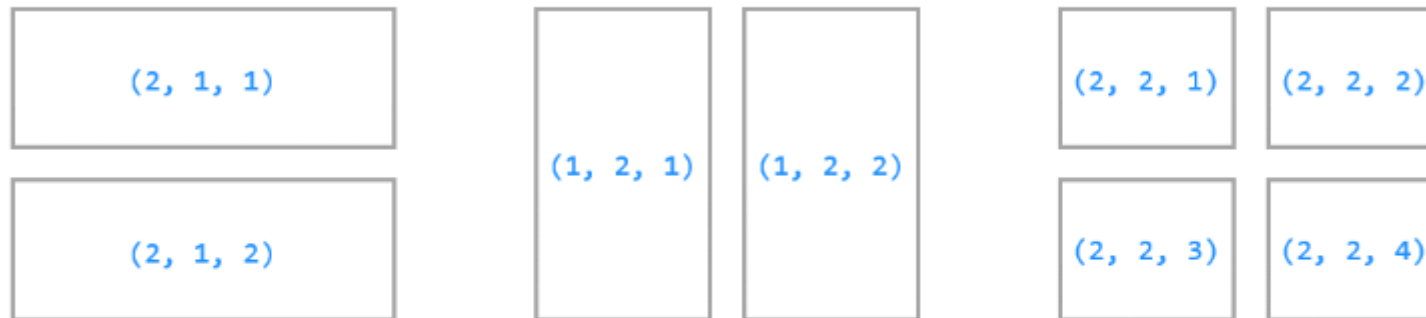
- 그리드(grid) 형태의 Axes 객체들을 생성
  - Figure 안에 있는 각각의 플롯은 Axes 라고 불리는 객체
    - Figure 내부에 행렬(matrix) 형태의 여러 그림이 있으며 이를 Axes라 함
- `ax1 = plt.subplot(2, 1, 1)`
  - `ax1 = plt.subplot('211')` 로도 가능
  - `subplot(m, n, number)`
  - 세 개의 인수
    - **m, n**
      - 전체 그리드 행렬의 모양을 지시하는 두 숫자
    - **number**
      - 인수가 네 개 중 어느 것인지를 의미하는 숫자
      - 첫번째 플롯을 가리키는 숫자가 0이 아니라 1임에 주의





# plt.subplot(row, column, index)

Python language



`plt.subplot(row, column, index)`

# 서브플롯 예

Python language

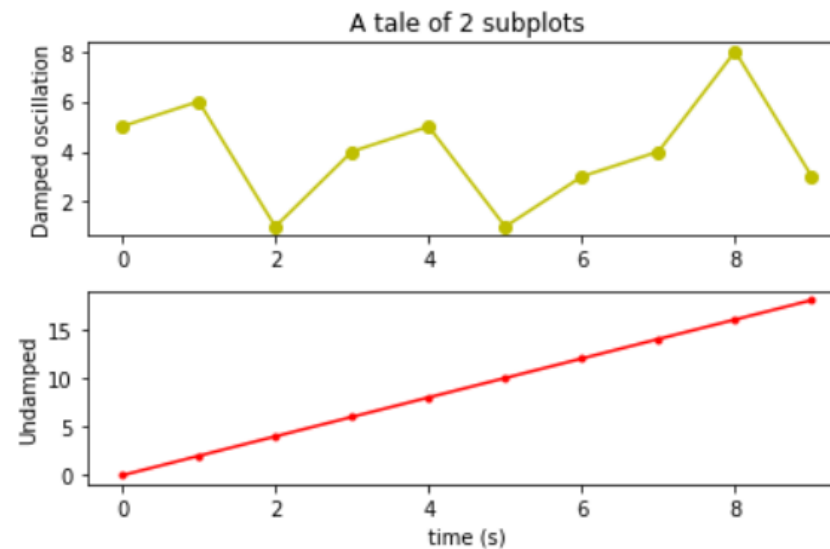
```
In [45]: ax1 = plt.subplot(2, 1, 1)
plt.plot(np.random.randint(1, 10, 10), 'yo-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')
print(ax1)

#ax2 = plt.subplot(2, 1, 2)
ax2 = plt.subplot(2, 1, 2)
plt.plot(np.arange(10), np.arange(10)*2, 'r.-')
plt.xlabel('time (s)')
plt.ylabel('Undamped')
print(ax2)

plt.tight_layout()
plt.show()
```

AxesSubplot(0.125,0.536818;0.775x0.343182)

AxesSubplot(0.125,0.125;0.775x0.343182)



AI Experts  
Who Lead  
The Future

04

`plt.subplots()`

- plt.subplots() 함수

- 여러 개의 서브플롯을 한 번에 생성하는 더 간단한 방법을 제공
- plt.subplots()는 행과 열의 개수를 지정하고 모든 서브플롯을 반환

```
import matplotlib.pyplot as plt

fig, axes = plt.subplots(2, 2) # 2행 2열의 서브플롯

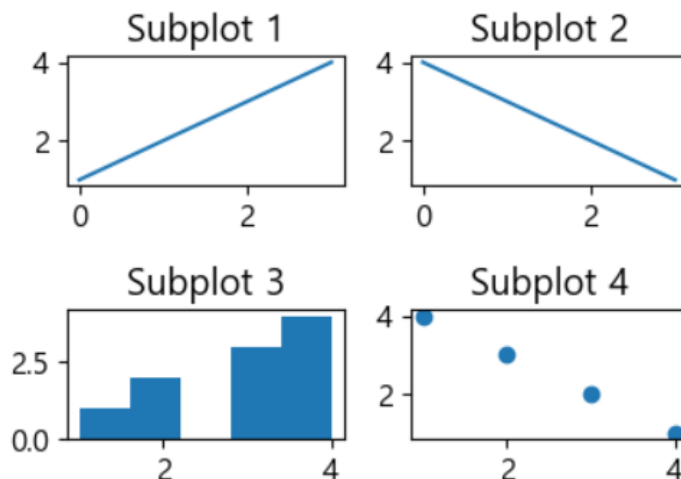
axes[0, 0].plot([1, 2, 3, 4])
axes[0, 0].set_title("Subplot 1")

axes[0, 1].plot([4, 3, 2, 1])
axes[0, 1].set_title("Subplot 2")

axes[1, 0].hist([1, 2, 2, 3, 3, 3, 4, 4, 4, 4], bins=5)
axes[1, 0].set_title("Subplot 3")

axes[1, 1].scatter([1, 2, 3, 4], [4, 3, 2, 1])
axes[1, 1].set_title("Subplot 4")

plt.tight_layout()
plt.show()
```



# plt.subplots(m, n)

Python language

## • 메소드

- matplotlib.pyplot.subplots(nrows=1, ncols=1, sharex=False, sharey=False, squeeze=True, subplot\_kw=None, gridspec\_kw=None, \*\*fig\_kw)
- 특정한 배치에 맞추어 여러 개의 서브플롯을 포함하는 figure를 생성
  - 인자 **sharex=True, sharey=True**
    - 각 축의 인자를 하나로 공유

Table 9-1. *pyplot.subplots options*

Argument	Description
nrows	Number of rows of subplots
ncols	Number of columns of subplots
sharex	All subplots should use the same x-axis ticks (adjusting the <code>xlim</code> will affect all subplots)
sharey	All subplots should use the same y-axis ticks (adjusting the <code>ylim</code> will affect all subplots)
subplot_kw	Dict of keywords passed to <code>add_subplot</code> call used to create each subplot
**fig_kw	Additional keywords to subplots are used when creating the figure, such as <code>plt.subplots(2, 2, figsize=(8, 6))</code>

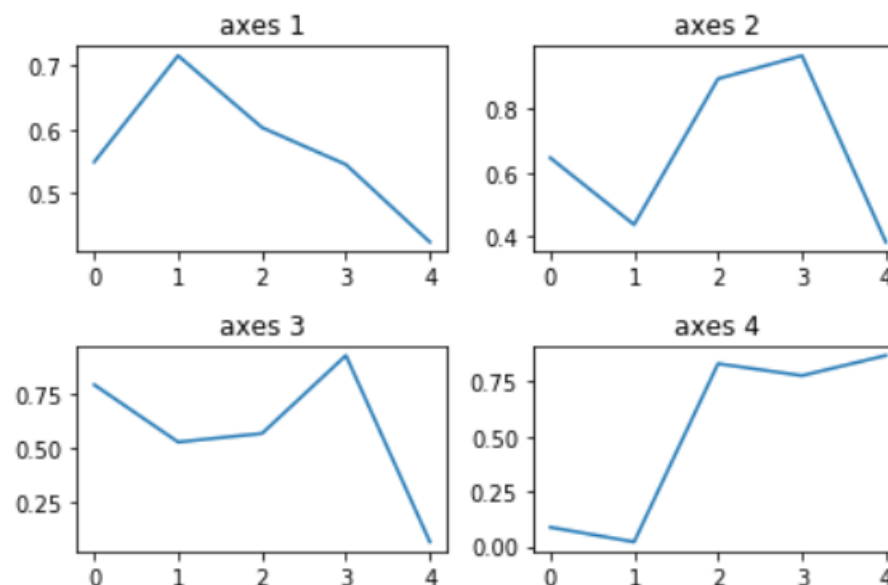
# plt.subplots() 활용

Python language

```
In [47]: fig, axes = plt.subplots(2, 2)

np.random.seed(0)
axes[0, 0].plot(np.random.rand(5))
axes[0, 0].set_title("axes 1")
axes[0, 1].plot(np.random.rand(5))
axes[0, 1].set_title("axes 2")
axes[1, 0].plot(np.random.rand(5))
axes[1, 0].set_title("axes 3")
axes[1, 1].plot(np.random.rand(5))
axes[1, 1].set_title("axes 4")

plt.tight_layout()
plt.show()
```



AI Experts  
Who Lead  
The Future

05

`fig.add_subplot()`

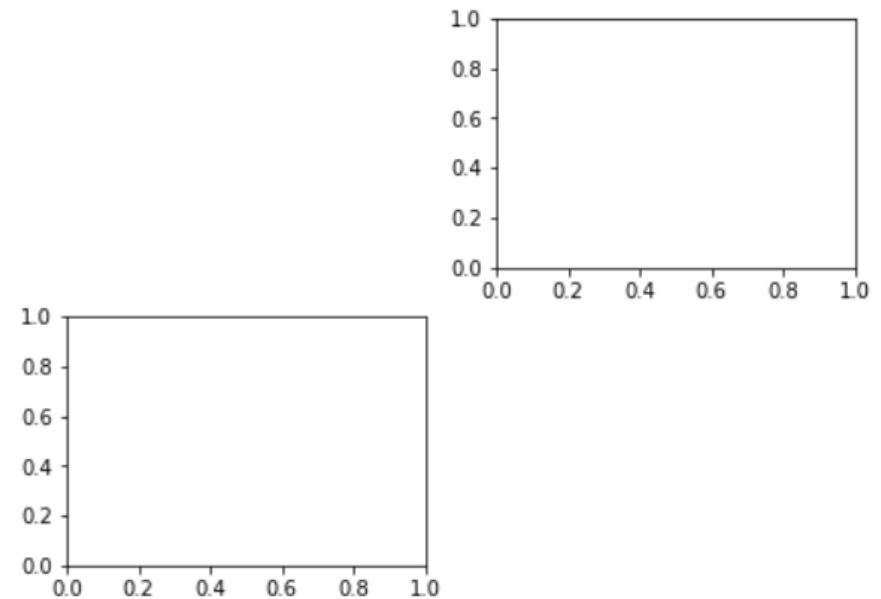
- **fig.add\_subplot() 메서드**
  - 서브플롯을 하나씩 추가할 수 있도록 figure 객체에 속한 메서드
  - 서브플롯을 생성하고 반환하며, 인덱스를 이용하여 위치를 지정



- **Axes: 내부의 subplot**

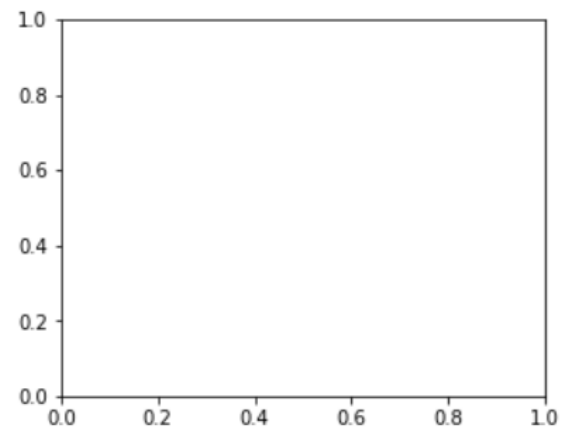
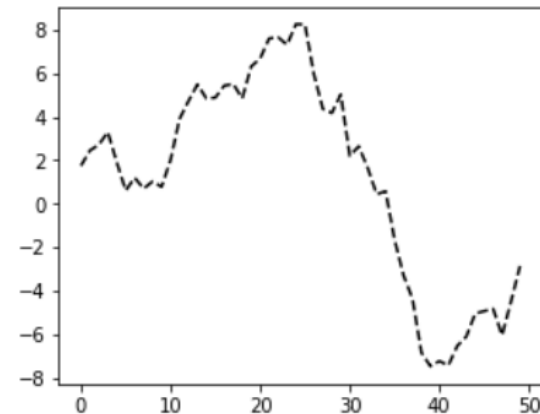
- ax1 = fig.add\_subplot(2, 2, 1)
  - 크기 2 x 2
  - 4개의 그림 중 1번 그림
- ax2 = fig.add\_subplot(2, 2, 2)
- ax3 = fig.add\_subplot(2, 2, 3)

```
In [15]: fig = plt.figure(figsize=(7, 5))  
#ax1 = fig.add_subplot(2, 2, 1)  
ax2 = fig.add_subplot(2, 2, 2)  
ax3 = fig.add_subplot(2, 2, 3)
```



- 가장 최근의 figure와 그 서브플롯에 그림

```
In [19]: fig = plt.figure(figsize=(10, 8))  
#ax1 = fig.add_subplot(2, 2, 1)  
ax2 = fig.add_subplot(2, 2, 2)  
plt.plot(np.random.randn(50).cumsum(), 'k--')  
ax3 = fig.add_subplot(2, 2, 3)
```

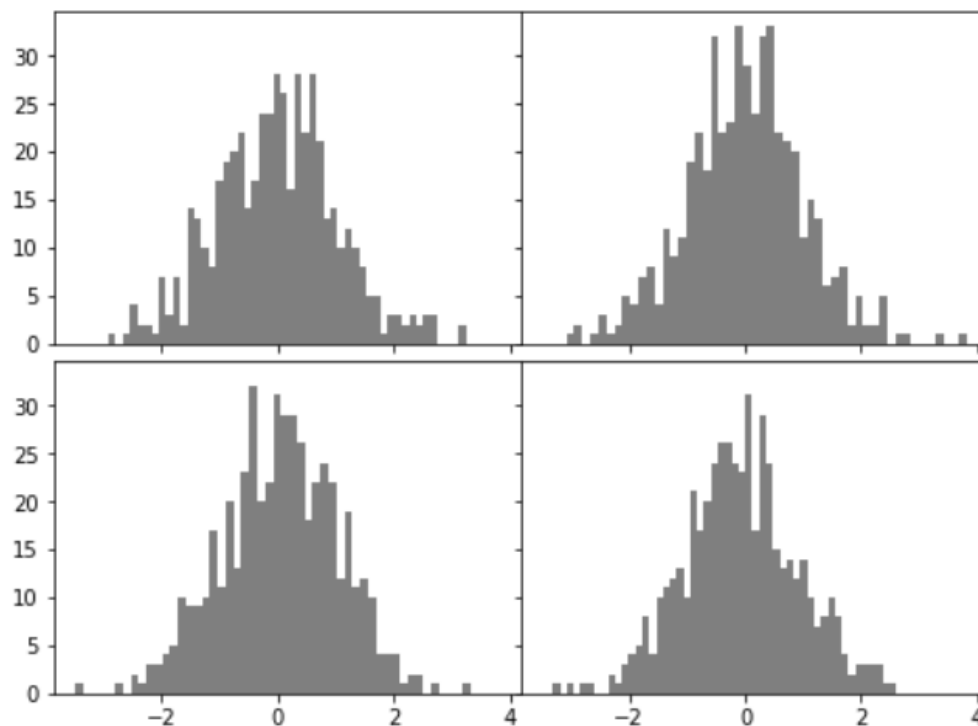


# 서브 플롯 간의 간격 조절하기

Python language

- `plt.subplots_adjust(wspace=0, hspace=.3)`
  - 서브플롯 간의 간격을 설정
    - 비율로 명시
      - .3은 30%의 공간 비우기

```
In [61]: fig, axes = plt.subplots(2, 2, sharex=True, sharey=True, figsize=(8, 6))
for i in range(2):
    for j in range(2):
        axes[i, j].hist(np.random.randn(500), bins=50, color='k', alpha=0.5)
plt.subplots_adjust(wspace=0, hspace=.05)
#plt.subplots_adjust(wspace=None, hspace=None) #적정한 공간 비우기
```



AI Experts  
Who Lead  
The Future

# 06

## Colab 한글지원

- 모듈 설치

-----  
-----  
# colab 한글지원 #1, #2, 이후에 바로 가능

# 1. 모듈 설치 koreanize-matplotlib  
# 참조 사이트 <https://github.com/ychoi-kr/koreanize-matplotlib>  
!pip install koreanize-matplotlib

# 2. 한글을 선명하게  
%config InlineBackend.figure\_format = 'retina'

```
[1] # 1. 모듈 설치 koreanize-matplotlib  
# 참조 사이트 https://github.com/ychoi-kr/koreanize-matplotlib  
!pip install koreanize-matplotlib
```

```
Requirement already satisfied: koreanize-matplotlib in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: matplotlib in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.8/dist-packages  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages
```

```
[2] %config InlineBackend.figure_format = 'retina'
```

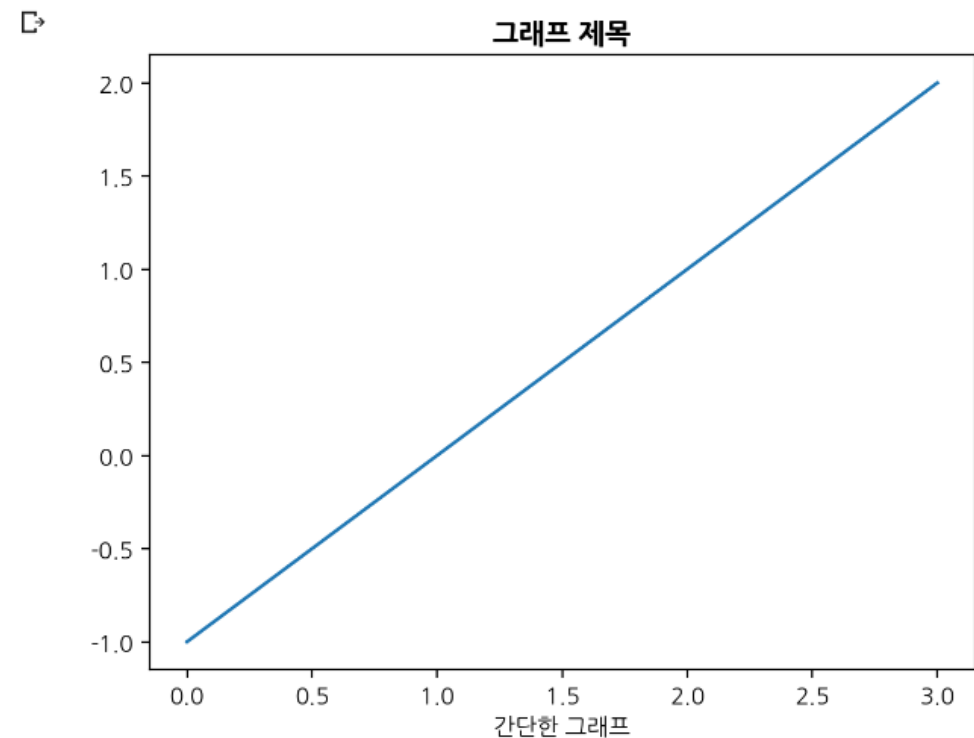
## # 3. 바로 실행 테스트

```
import koreanize_matplotlib
import matplotlib.pyplot as plt
```

```
plt.plot([-1, 0, 1, 2])
plt.title('그래프 제목',
fontweight="bold")
plt.xlabel('간단한 그래프')
plt.show()
```

```
import matplotlib.pyplot as plt
import koreanize_matplotlib

plt.plot([-1, 0, 1, 2])
plt.title('그래프 제목', fontweight="bold")
plt.xlabel('간단한 그래프')
plt.show()
```

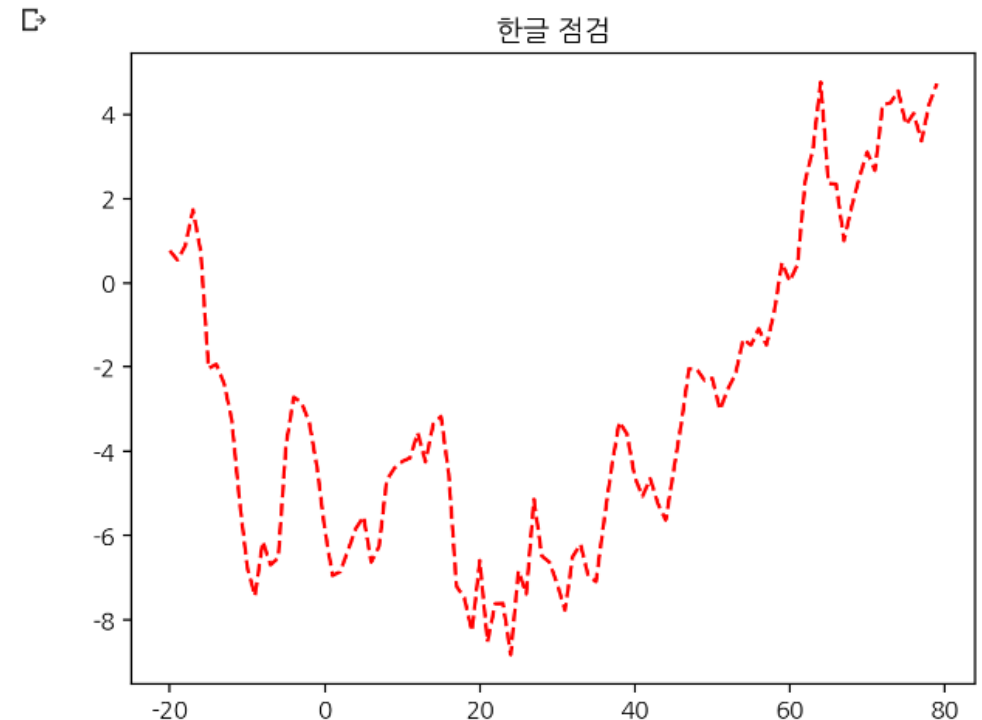


```
import numpy as np
import matplotlib.pyplot as plt

plt.plot(range(-20, 80),
np.random.randn(100).cumsum(), 'r--')
plt.title('한글 점검');
```

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

plt.plot(range(-20, 80), np.random.randn(100).cumsum(), 'r--')
plt.title('한글 점검');
```



```
fig = plt.figure(figsize=(15, 12))  
fig, ax = plt.subplots(subplot_kw=dict(projection='3d'),  
figsize=(15, 12))
```

```
graph_title = '시험'  
ax.set_title(f'{graph_title}')  
ax.set_xlabel('야호 (Hz)', labelpad=20)  
ax.set_ylabel('시간 (s)', labelpad=20)  
ax.set_zlabel('이건 뭐지 (dB/Hz)', labelpad=20)
```

```
ax.view_init(20, 20)  
plt.show()
```

