파이썬 데이터분석
- Matplotlib

강사 : KAIST 김동훈

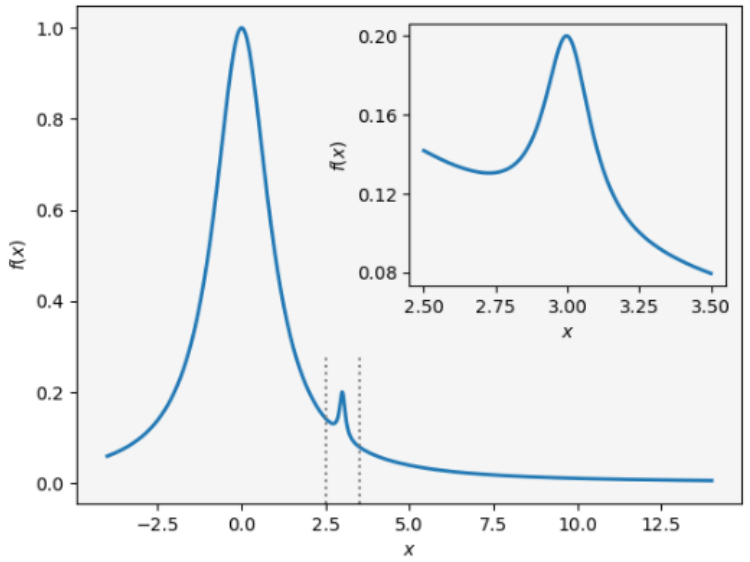
유투브 Link : https://www.youtube.com/watch?v=NZbP0kEX2y4&list=PL73qGQ0nG_q0mRvozqFA029sEErUVnada

Ⅰ. Matplotlib 개요

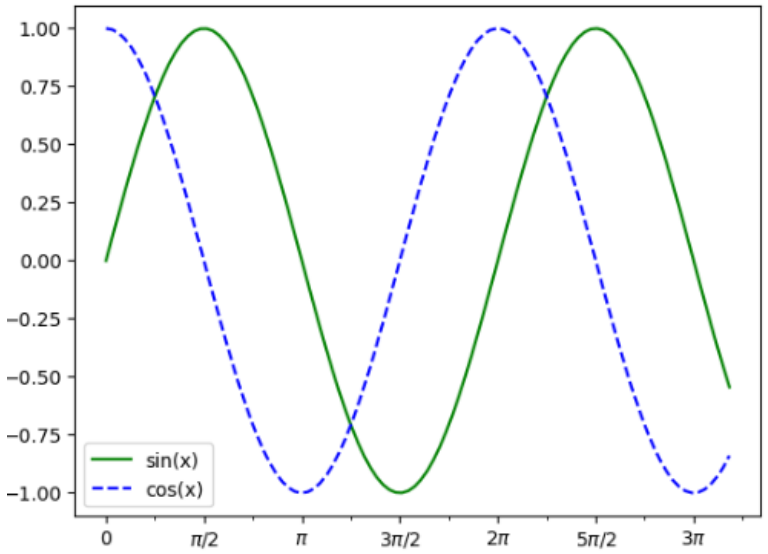
- What is Matplotlib?
- 이 과정에서 배우는 것들
- 무엇이 Matplotlib 을 어렵게 하는가?
- OO style 을 사용해야 하는 이유
- OO style vs MATLAB style

Ⅰ. Matplotlib 개요

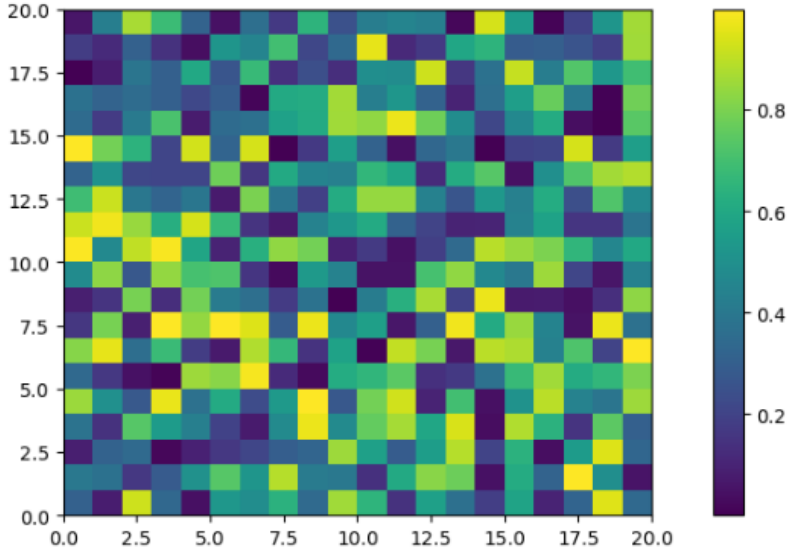
- 이 과정에서 배우게 될 것들



여러 그래프 동시에 그리기



범례상자와 눈금지정



Colorbar

Ⅰ. Matplotlib 개요

- 무엇이 Matplotlib 을 어렵게 하는가?

  OOStyle : Object-Oriented Style (객체지향) 스타일

  - 구글링하면 MATLAB style code 와 OOStyle code 가 혼재되어 있다.

    ex) “legend box location” @구글

```
plt.subplot(211)
plt.plot([1, 2, 3], label="test1")
plt.plot([3, 2, 1], label="test2")

# Place a legend above this subplot,
# fully use the given bounding box.
plt.legend(bbox_to_anchor=(0., 1.02,
           ncol=2, mode="expand", bor

plt.subplot(223)
plt.plot([1, 2, 3], label="test1")
plt.plot([3, 2, 1], label="test2")
# Place a legend to the right of this
plt.legend(bbox_to_anchor=(1.05, 1),

plt.show()
```

[1]

```
y_value = .55

fig, ax = plt.subplots(1, 2, sharex =
fig.set_size_inches(50,30)

ax[0].plot(x, y, label = "cos")
ax[0].set_ylim([0.8,3.2])
ax[0].legend(loc=2)

line1 ,= ax[1].plot(x,y)
ax[1].set_ylim([0.8,3.2])

axbox = ax[1].get_position()

fig.legend([line1], ["cos"], loc = (a

plt.show()
```

[2]

* 내용 출처 : https://en.wikipedia.org/wiki/Matplotlib

Ⅰ. Matplotlib 개요

- 무엇이 Matplotlib 을 어렵게 하는가?
  - axis 와 axes 혼동

axis = 그래프의 축 객체 // **axes** = subplot (단일 그래프)

* 내용 출처 : https://en.wikipedia.org/wiki/Matplotlib

Ⅰ. Matplotlib 개요

- OO style 을 사용해야 하는 이유
  - MATLAB style 과 OO style 사이에서 고통받지 말자
  - subplot 을 다루기 더 쉽다
  - 다른 Python library 등과 연동이 더 쉽다. ex) Pandas, Seaborn
  - Custom Plot Function 제작 가능
  - 모듈화 된 코드 작성 가능

※ OOStyle : Object-Oriented Style (객체지향) 스타일

Ⅰ. Matplotlib 개요

- OOStyle vs MATLAB Style

MATLAB Style

In [6]:
```
plt.plot(x, np.sin(x))
plt.title('A Sine Curve')
plt.xlabel('x')
plt.ylabel('sin(x)')
```

In [7]:
```
plt.subplots_adjust(hspace=0.4, wspace=0.4)

for i in range(1, 7):
    plt.subplot(2, 3, i)
    plt.text(0.5, 0.5, str((2, 3, i)),
    fontsize=18, ha='center')
```

OO Style

In [6]:
```
fig = plt.figure()
ax1 = fig.add_subplot(111)
ax1.plot(x, np.sin(x))
ax1.set_xlabel('x')
ax1.set_ylabel('sin(x)')
ax1.set_title('A Sine Curve')
```

In [6]:
```
fig = plt.figure()
fig.subplots_adjust(hspace=0.4, wspace=0.4)
for i in range(1, 7):
    ax = fig.add_subplot(2, 3, i)
    ax.text(0.5, 0.5, str((2, 3, i)),
    fontsize=18, ha='center')
```

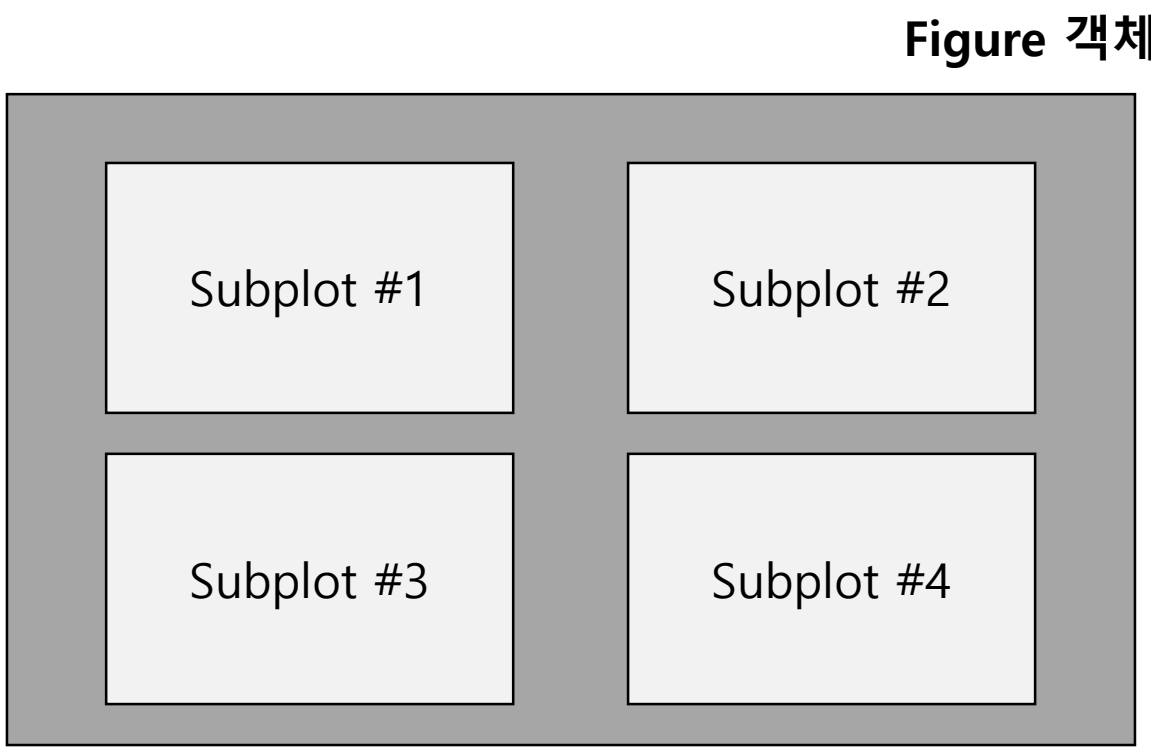
Ⅱ. Subplots

- What is subplot?
- Matplotlib 의 구성 요소
- figure, axes 객체
- Multiple subplots 생성방법

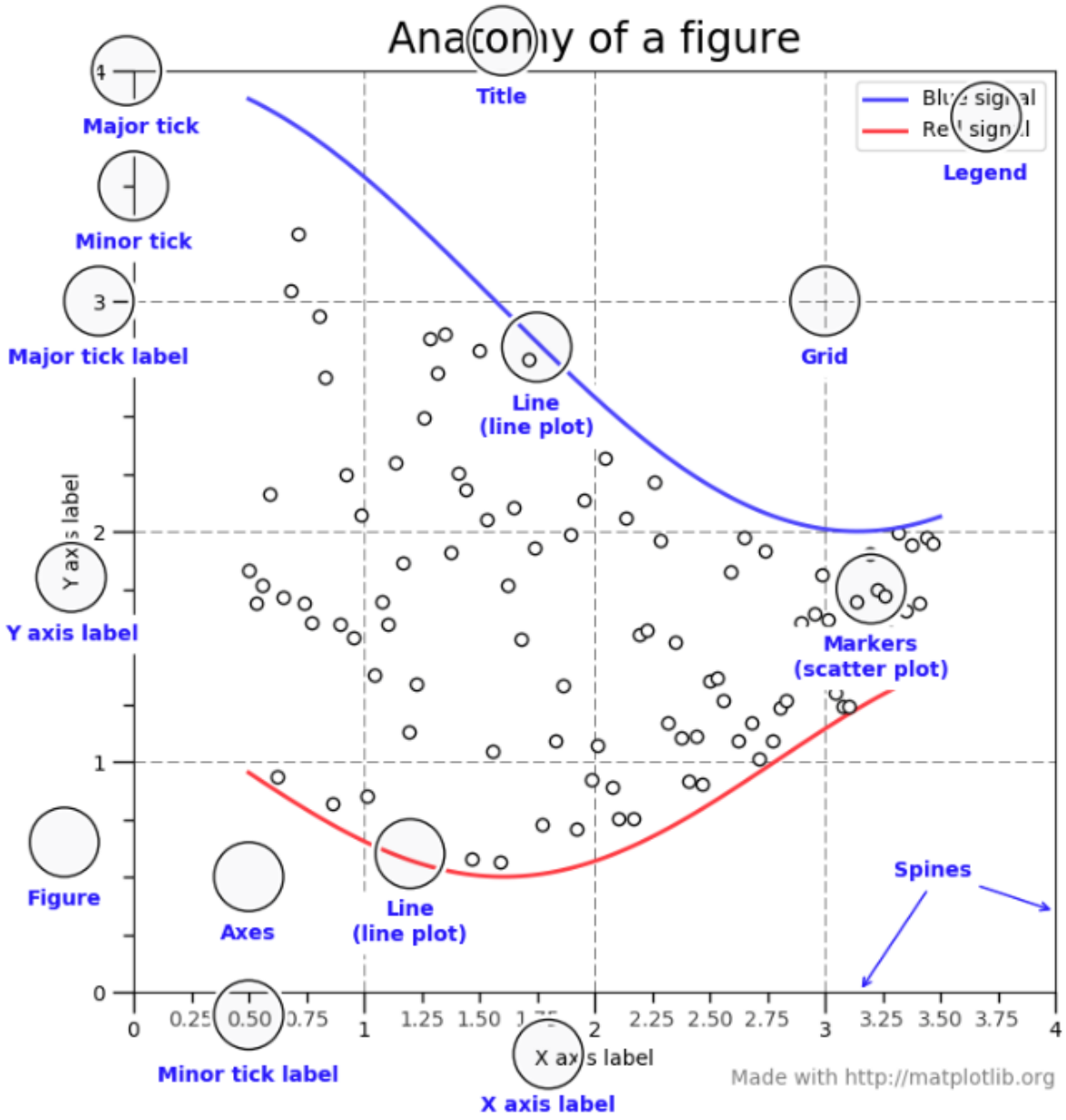
Ⅱ. Subplots

- **What is subplot ?**

  - Matplotlib 의 Figure 는 그래프가 그려지는 Canvas 를 의미한다.
  - Figure 객체 위의 단일 그래프를 의미한다.
  - Figure 객체는 여러 개의 subplot 을 가질 수 있다. (= multiple subplots)



Ⅱ. Subplots - Matplotlib의 구성 요소



① Figure : 전체 Drawing. 보통 하나 이상의 Axes 객체 소유

② Axes : 단일 그래프 객체 (A Plot) Figure 객체에 종속 됨.
Axis 와 혼동 금지
title, xlim, ylim 설정가능
OO interface 구현을 위한 진입점

③ Axis : 축. graph limit, tick (Locator), ticklabel (Formatter)

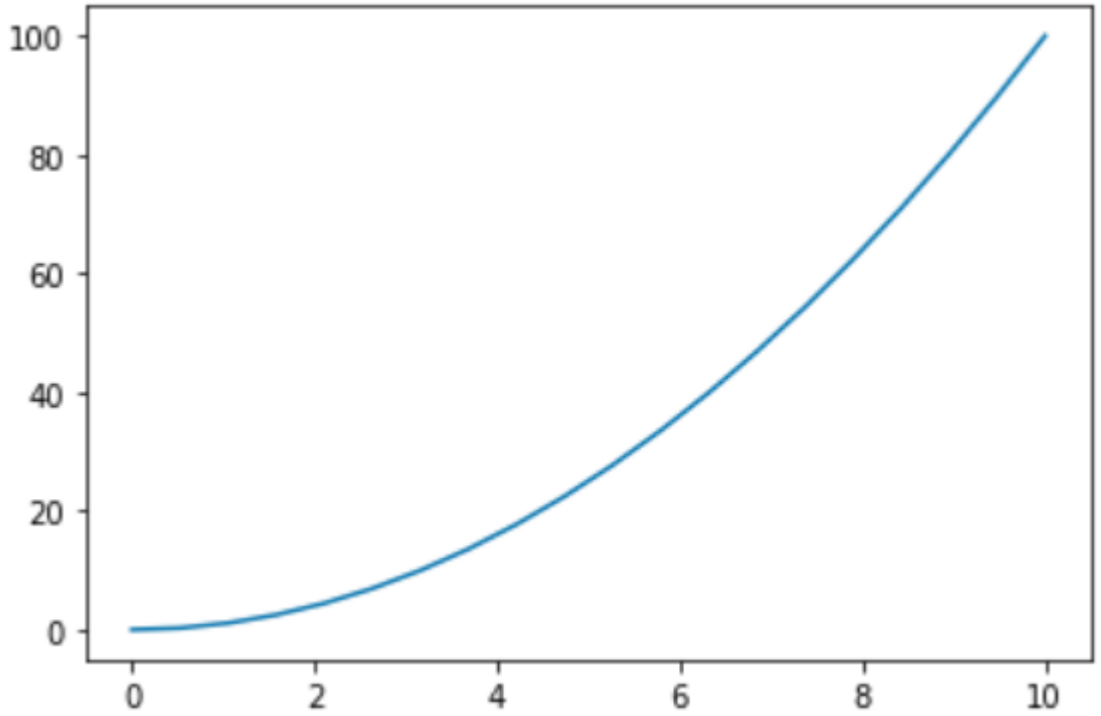
④ Artist : Figure 객체 위에 올라오는 모든 것.
(Text, Line2D, collection, Patch, …)
Figure 객체 rendering시, 모든 Artist 객체가 canvas에 그려 짐

그림출처 : https://matplotlib.org/stable/gallery/showcase/anatomy.html?highlight=anatomy

Ⅱ. Subplots

- Figure 와 Axes 예제

ex) Simple Plot, y = x^2



Ⅱ. Subplots

- Axes 객체(subplot) 를 추가하는 3 가지 방법

1. **plt.subplots**() : figure 객체와 Axes 객체를 한꺼번에 생성

```
fig, axes = plt.subplots(nrows, ncols)
```

2. **fig.add_subplot**() : figure 객체를 먼저 생성 후, Axes 객체를 하나씩 추가

```
fig = plt.figure()
ax = fig.add_subplot(nrows,ncols,index)
```

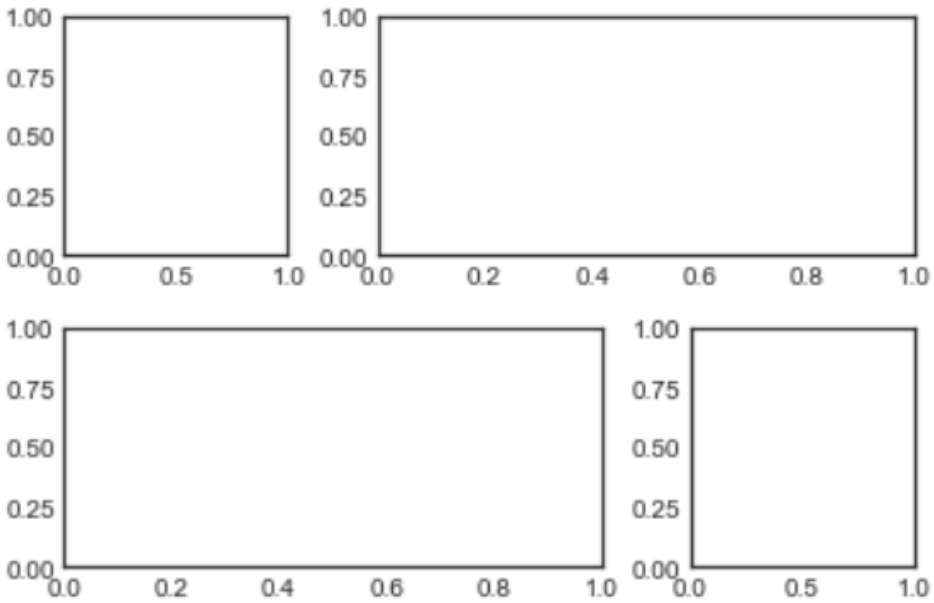
3. **fig.add_axes**() : figure 객체를 먼저 생성 후, Axes 객체를 절대좌표로 생성

```
fig = plt.figure()
ax = fig.add_axes([left, bottom, width, height])
```

Ⅱ. Subplots

- **fig.add_subplot() + GridSpec**

```
import matplotlib.gridspec as gridspec

fig = plt.figure()
spec = gridspec.GridSpec(2, 3, wspace=0.4, hspace=0.3)

ax1 = fig.add_subplot(spec[0, 0])
ax2 = fig.add_subplot(spec[0, 1:])
ax3 = fig.add_subplot(spec[1, :2])
ax4 = fig.add_subplot(spec[1, 2])
```

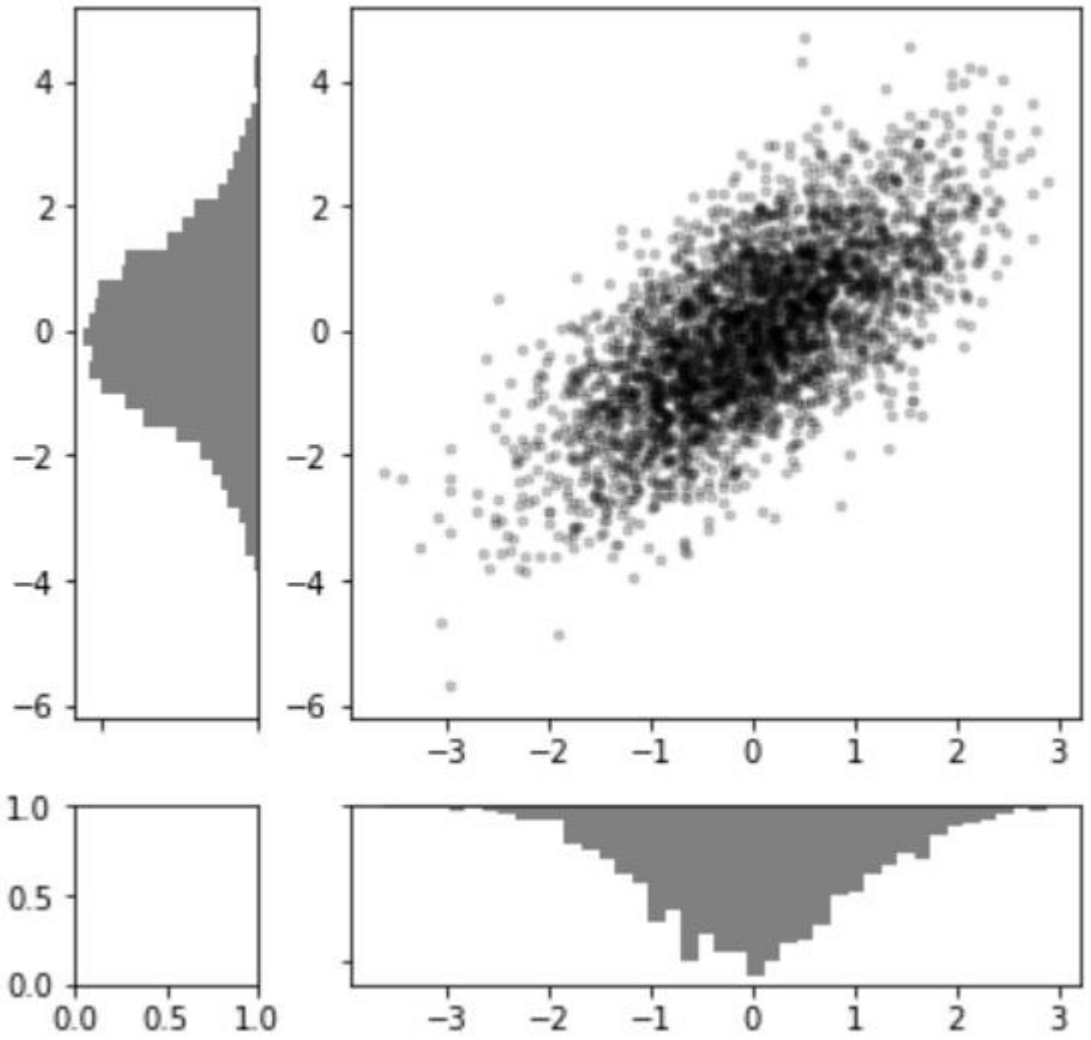


Ⅱ. Subplots

노트북 0203 Multi-axes histogram

- Axes 객체(subplot) 를 추가하는 3 가지 방법

gridspec 와 add_subplot 을 동시 사용 하여 2차원 정규분포의 히스토그램을 plot



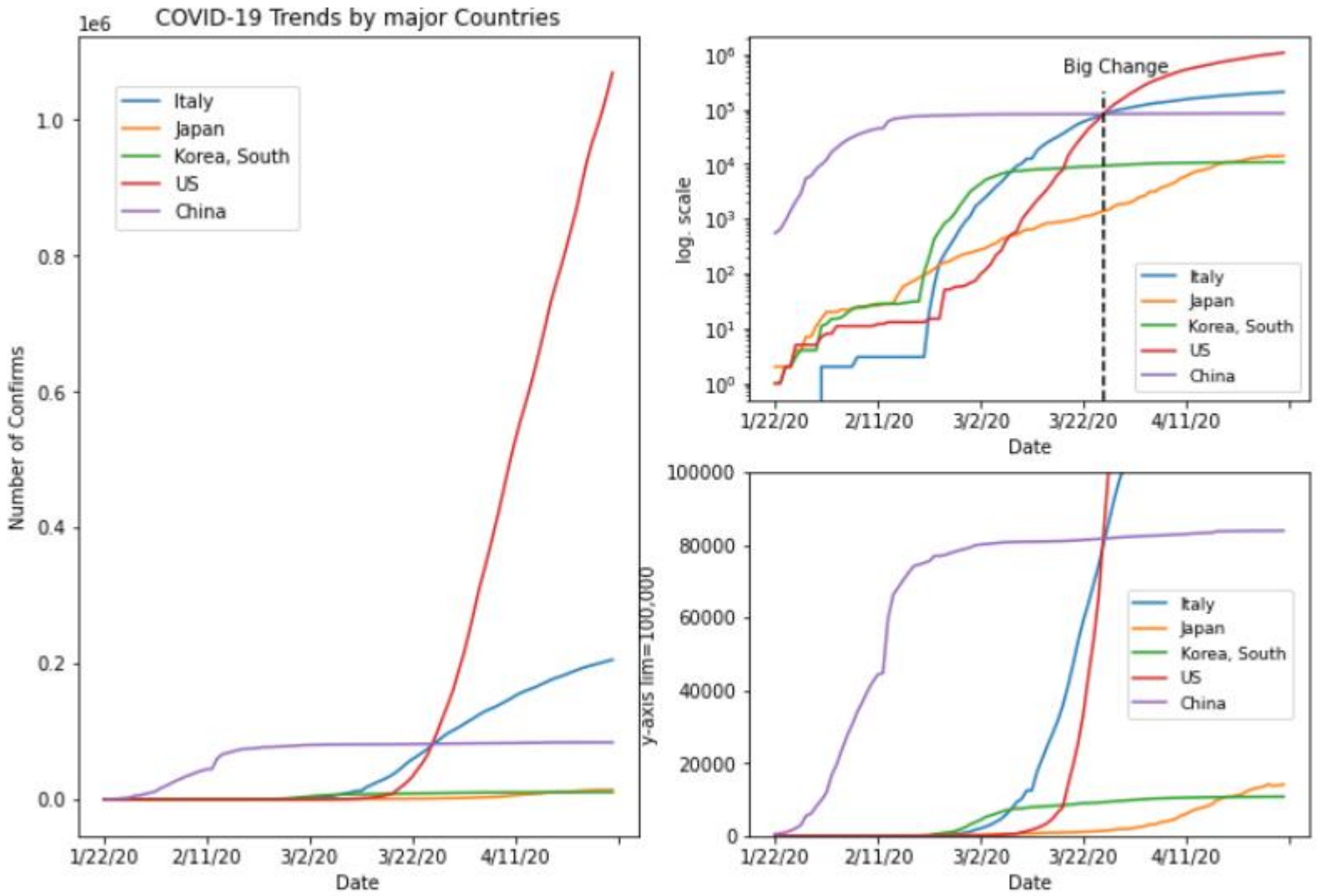
Ⅱ. Subplots

노트북 0204 Multiple Subplots COVID19

- Axes 객체(subplot) 를 추가하는 3 가지 방법

plt.subplots(),
fig.add_subplot(),
fig.add_axes()

3가지 방법으로 최대한 동일하게
Multiple Subplot 을 그려보았다.



Ⅲ. Graph Details

- Legend Box (범례상자)
- bbox to anchor
- Grid, Axis, Spine, ShareAxis
- Ticks (눈금)

Ⅲ. Graph Details

- LegendBox (범례상자) 를 생성하는 3가지 방법

1. **plot 함수 내에서 “label” 정의**
2. **line (handle) 을 정의하고 set_label 사용**
3. **line (handle) 을 정의하고 Axes.legend 안에서 label 명 부여**

Ⅲ. Graph Details

노트북 0302 LegendBox02

- LegendBox (범례상자) 의 위치, 형태, 모양, 색깔 등

  - **loc** : 위치 지정 가능 , ex) upper left, lower right
  - **frameon** : True 면 범례상자의 테두리를 표시
  - **ncol** : 범례를 하나의 컬럼으로 나타낼지 여러 개 컬럼으로 나타낼지 지정 가능
  - **framealpha** : 상자의 투명도 조정가능
  - 기타 : fancybox, shadow, borderpad

Ⅲ. Graph Details

- Grid 객체
  - 데이터의 위치를 더 명확하게 나타내기 위해 그래프에 그리드(격자)를 표현할 수 있다.

- Axis 객체
  - 그래프의 축을 뜻하며, 축의 범위를 지정가능하다.
  - axis(), set_xlim(), set_ylim()

- Spine 객체
  - 데이터 영역의 경계를 나타내는 선
  - set_position(), set_visible()

Ⅲ. Graph Details

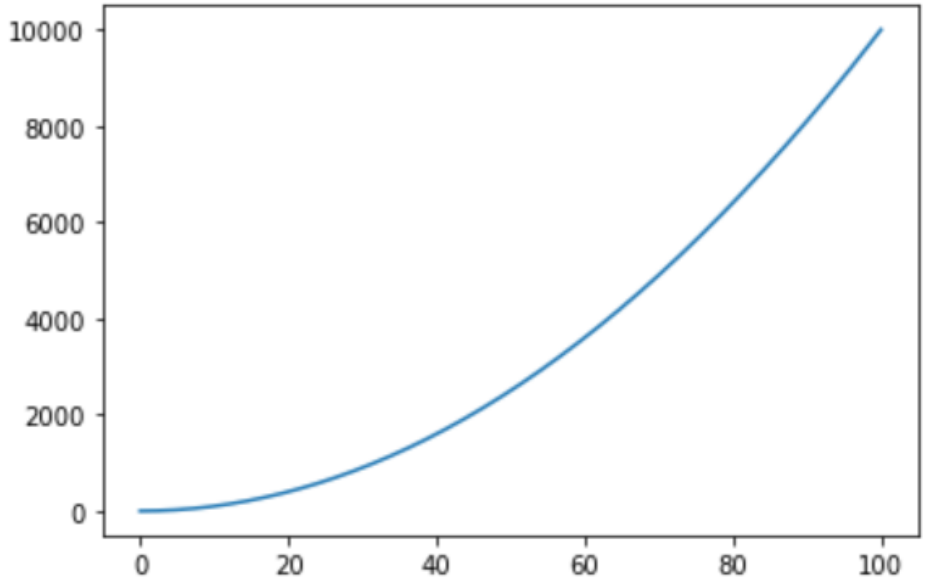
- Axis Scales
  - 그래프의 축 스케일을 다양하게 지정할 수 있다. ex) linear, log, logit
  - set_xscale(), set_ycale()

- Shared Axis (이중축)
  - 두 종류의 데이터를 동시에 하나의 그래프에 표시하기 위해 이중 축을 표시할 수 있다.
  - twinx()

Ⅲ. Graph Details

노트북 0306 Ticks_Locators_Formatters

- Ticks
  - 그래프 축 눈금
  - 주축 (major) 과 보조축 (minor) 로 나눠짐
  - Locators : 눈금의 위치를 제어
  - Formatters : 눈금의 라벨을 제어
  - Locators 와 Formatters 모두 주축, 보조축 눈금을 생성할 수 있다.
  - 주축은 default 켜져 있고, 보조축은 default 꺼져 있다. (NullLocator, NullFormatter)




```
In [4]: print(ax.xaxis.get_major_locator())
        print(ax.xaxis.get_minor_locator())
```

```
<matplotlib.ticker.AutoLocator object at 0x7f79a071ae80>
<matplotlib.ticker.NullLocator object at 0x7f7984905fd0>
```

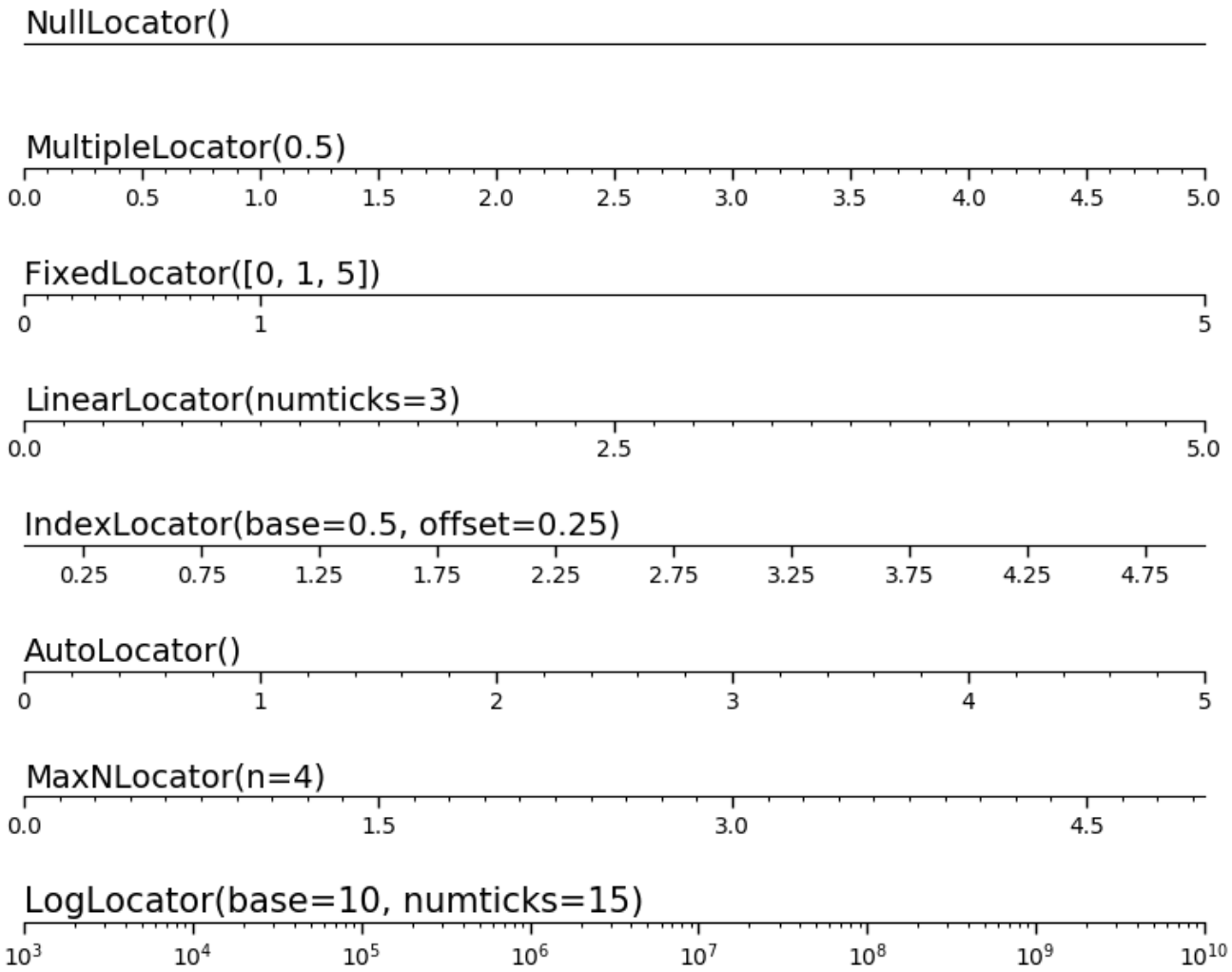
```
In [5]: print(ax.xaxis.get_major_formatter())
        print(ax.xaxis.get_minor_formatter())
```

```
<matplotlib.ticker.ScalarFormatter object at 0x7f79848833d0>
<matplotlib.ticker.NullFormatter object at 0x7f79a07411c0>
```

Ⅲ. Graph Details

노트북 0306 Ticks_Locators_Formatters

- Ticks – 다양한 Locators



눈금 없음

기준 값의 배수

고정된 값

최소~최대 값 사이에 균일하게

고정된 값

자동 눈금

최대 ticks 수(=n) 에 맞춰 자동 눈금

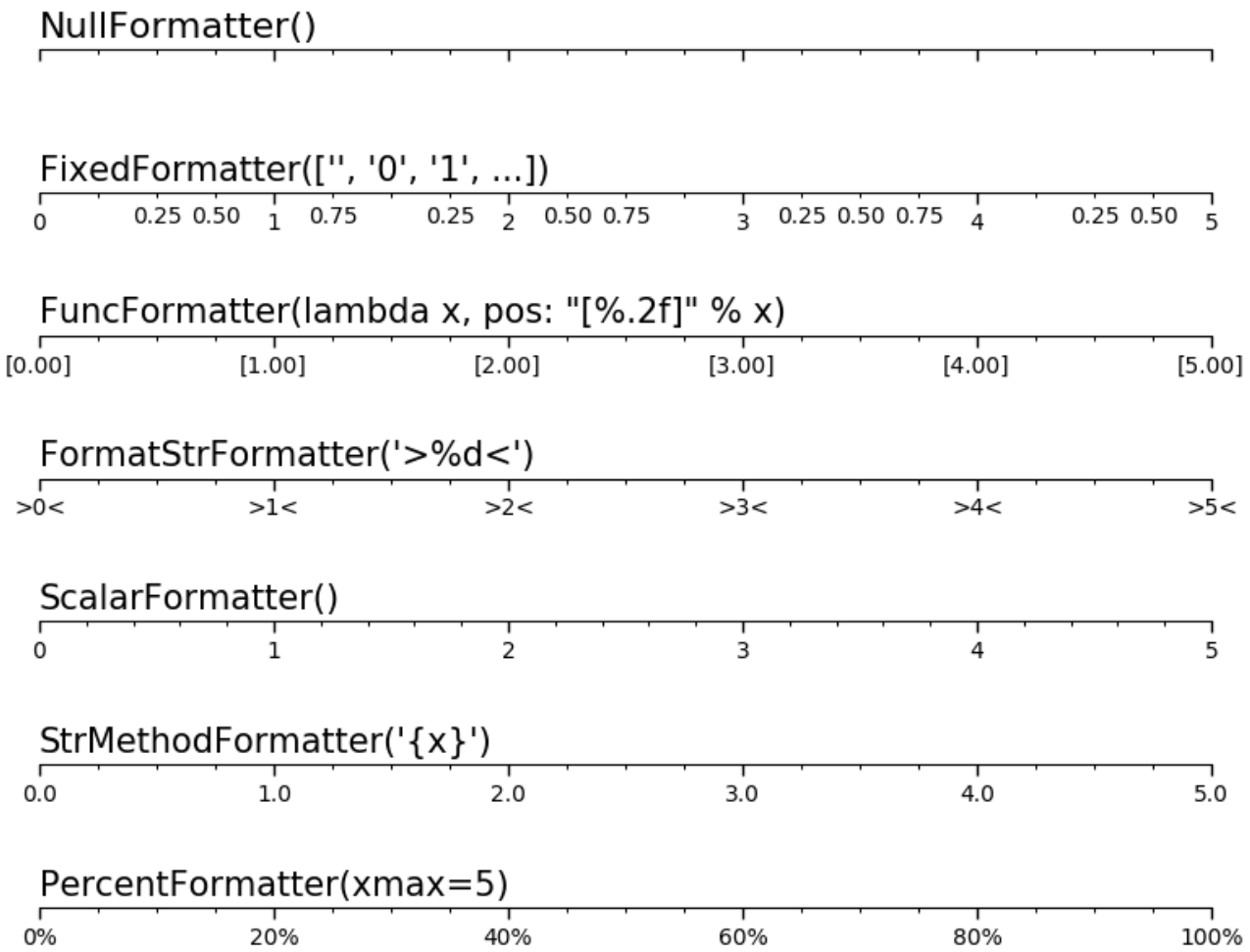
최소~최대 값 사이에 log scale로

참고 사이트 : https://matplotlib.org/3.1.1/gallery/ticks_and_spines/tick-locators.html

Ⅲ. Graph Details

노트북 0306 Ticks_Locators_Formatters

- Ticks – 다양한 Formatters



라벨 없음

수동 라벨

User defined function 으로 정의

old-style sprint format string

scalar 값으로 된 라벨

format 메서드 사용

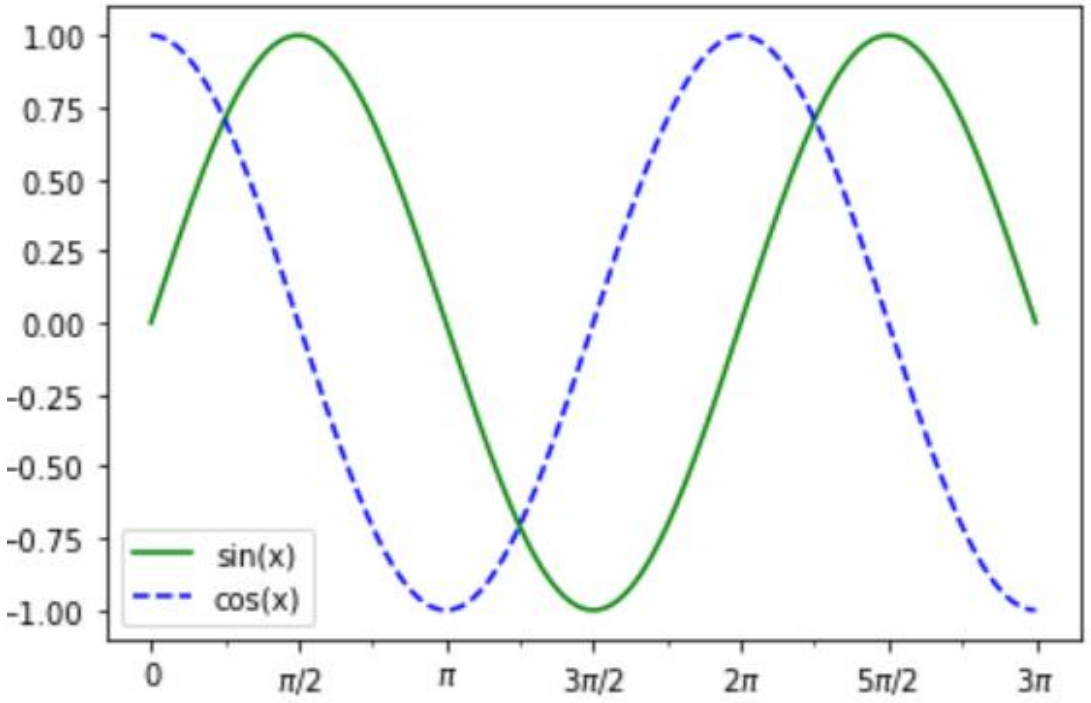
퍼센트로 라벨 표기

참고 사이트 : https://matplotlib.org/3.1.1/gallery/ticks_and_spines/tick-formatters.html

Ⅲ. Graph Details

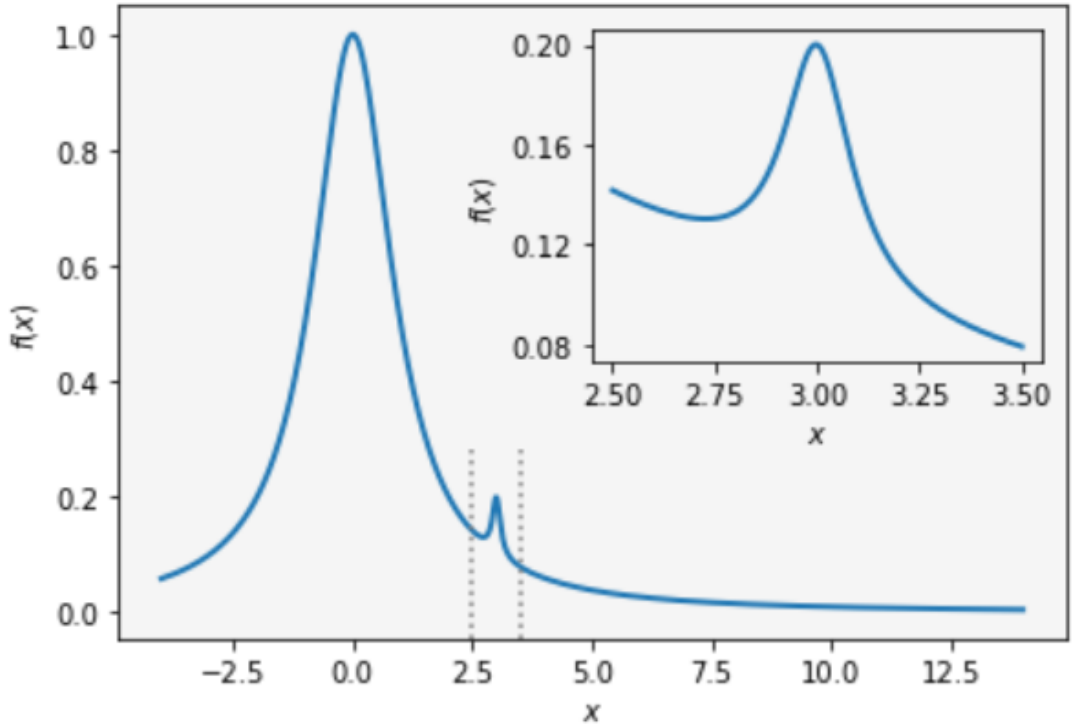
노트북 0307 Ticks_Examples

- Ticks – 다양한 Formatters



ax.xaxis.set_major_locator(MultipleLocator(np.pi/2))
ax.xaxis.set_minor_locator(MultipleLocator(np.pi/4))

ax.xaxis.set_major_formatter(FuncFormatter(format_func))



ax_insert.xaxis.set_major_locator(MaxNLocator(5))
ax_insert.yaxis.set_major_locator(MaxNLocator(4))

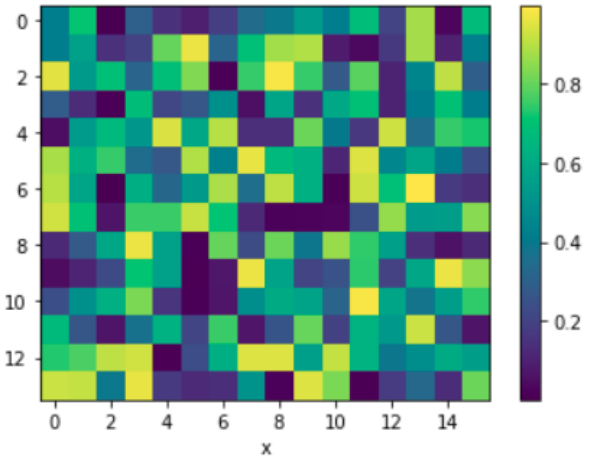
Ⅳ. Colorbar

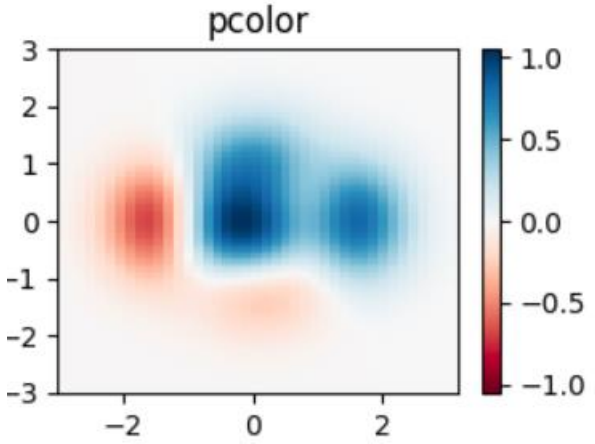
- Colorbar 개념
- Colorbar 위치 시키기
- Colorbar 와 ScalarMappable
- Colormap

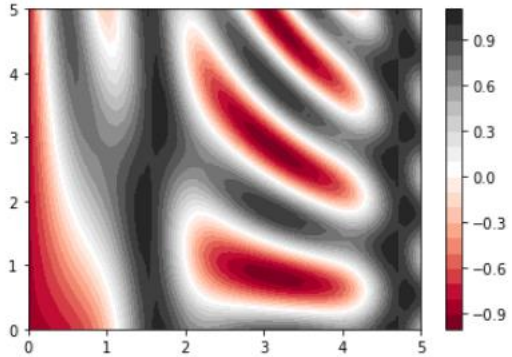
IV. Colorbar

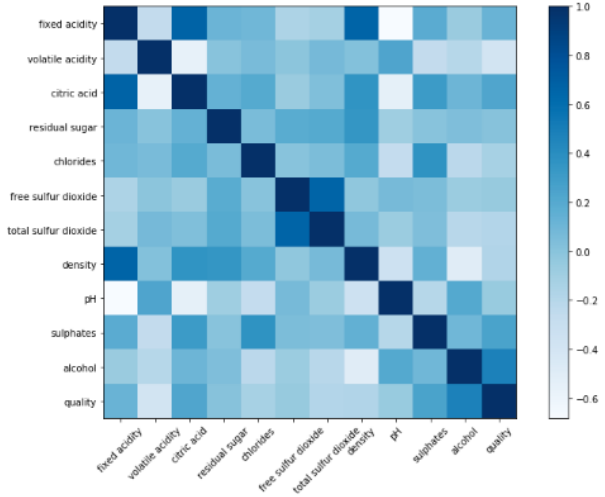
- Colorbar 를 사용하는 다양한 Plot 예시

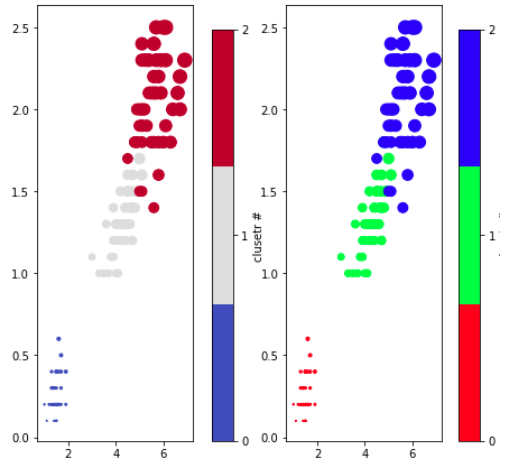
  - colorbar 와 같이 사용하는 subplot들 : **imshow, heatmap, pcolor(mesh), scatter, contour(f), *custom***

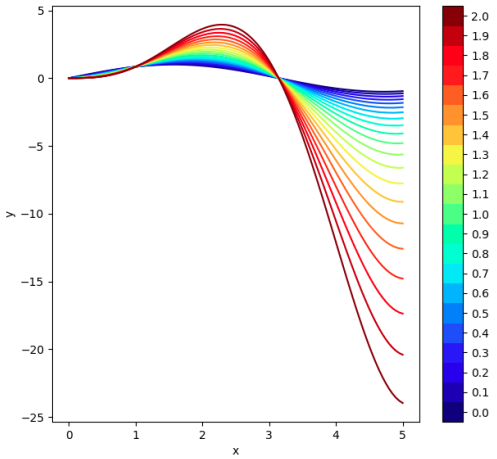










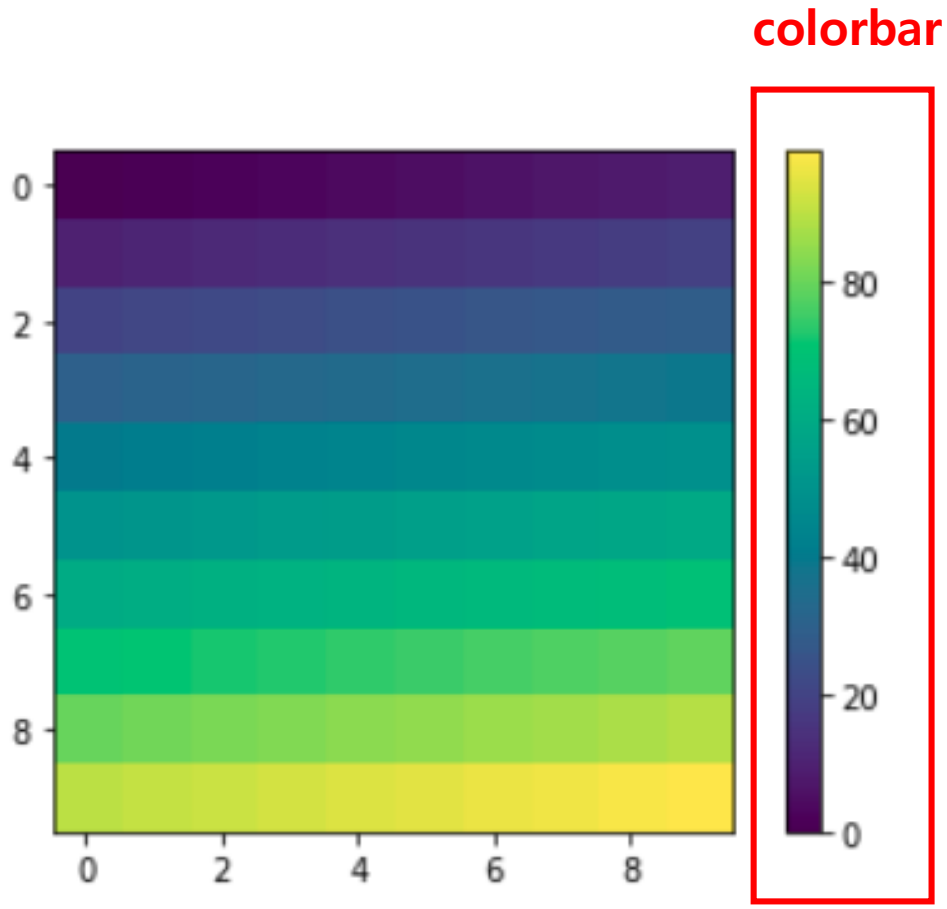


IV. Colorbar

노트북 0401 Simple Colorbar

- A very simple colorbar example

  - imshow & colorbar



IV. Colorbar

- What is a colorbar ?

  - matplotlib.pyplot.**colorbar**(mappable=None, cax=None, ax=None, **kw) 함수

mappable : matplotlib.**cm.ScalarMappable** (i.e., AxesImage, ContourSet, etc.)
※ *pyplot.colorbar function, which sets the default to the current image*

cax : Axes into which the colorbar will be drawn (colorbar 가 그려질 axes 객체)

ax : One or more **parent axes** from which space for a new colorbar axes will be **stolen**, if cax is None. **This has no effect if cax is set**.

use_gridspec : If cax is None, a new cax is created as an instance of Axes.

Colorbar 는 ScalarMappable 을 필요로 하는
또 하나의 Axes 객체이다. (= subplot 객체)

참고 : https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.colorbar.html

IV. Colorbar

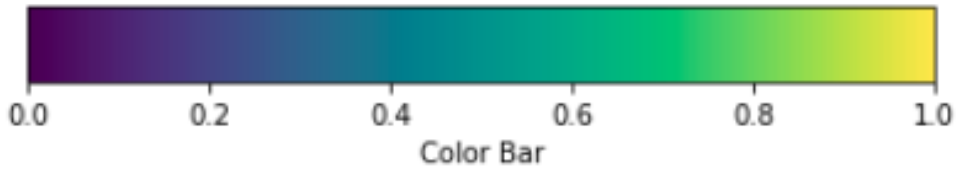
- What is a colorbar ?

  - matplotlib.pyplot.**colorbar**(mappable=None, cax=None, ax=None, **kw) 함수

Colorbar 는 ScalarMappable 을 필요로 하는 또 하나의 Axes 객체이다. (= subplot 객체)

- **ScalarMappable** : mixin class to map scalar data to RGBA

```
[[0.14, 0.72, 0.7 , 0.69, 0.25, 0.69],
 [0.23, 0.42, 0.37, 0.36, 0.06, 0.63],
 [0.71, 0.61, 0.65, 0.17, 0.15, 0.51],
 [0.88, 0.18, 0.46, 0.43, 0.5 , 0.16],
 [0.34, 0.26, 0.84, 0.8 , 0.43, 0.61],
 [0.15, 0.51, 0.3 , 0.86, 0.67, 0.63]]
```



IV. Colorbar

- What is a colorbar ?

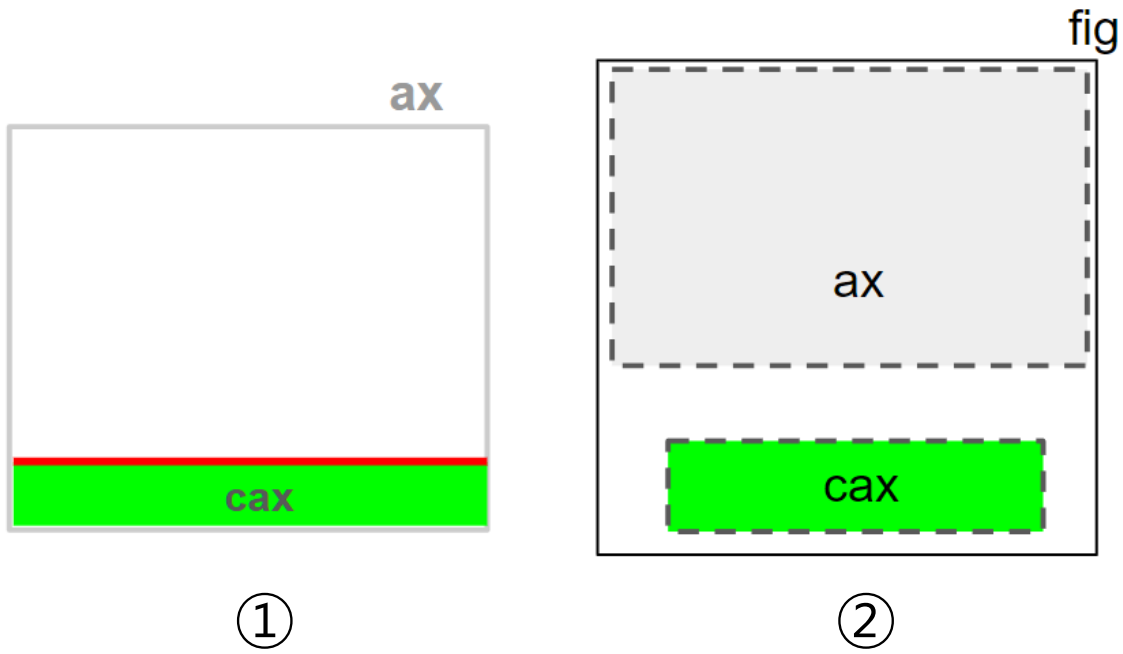
  - matplotlib.pyplot.**colorbar**(mappable=None, cax=None, ax=None, **kw) 함수

Colorbar 는 ScalarMappable 을 필요로 하는 또 하나의 Axes 객체이다. (= subplot 객체)

- **cax** : colorbar 가 그려질 axes 객체
- **ax** : 부모 Axes 객체

① 부모 Axes 객체의 영역을 같이 점유

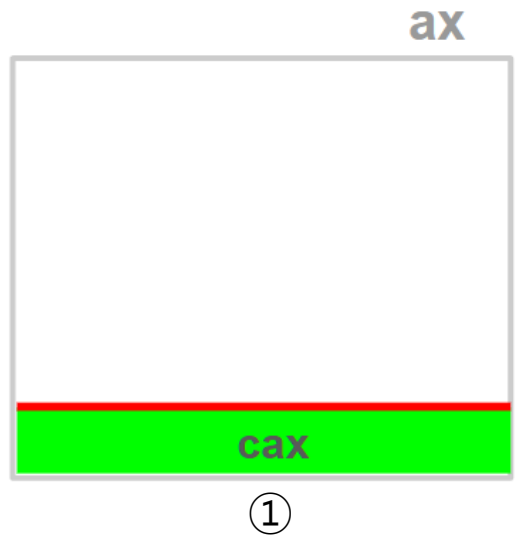
② 새로운 Axes 객체의 영역을 점유

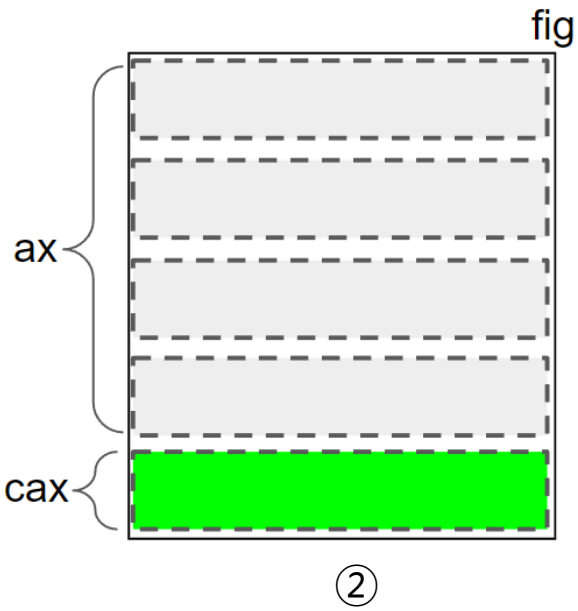


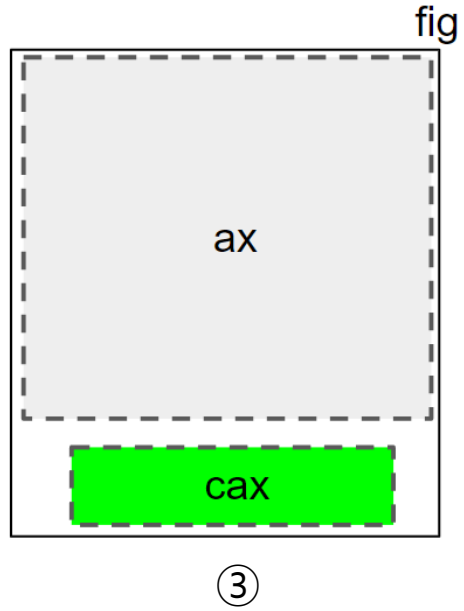
IV. Colorbar –Advanced

- Placing a colorbar

1. **using pad - ①**
2. **using axes divider - ①**
3. using subplots
   - **add_axes() - ③**
   - subplots() with gridspec_kw - ②
   - add_subplot() - ②
   - grid_spec & add_subplot() - ②



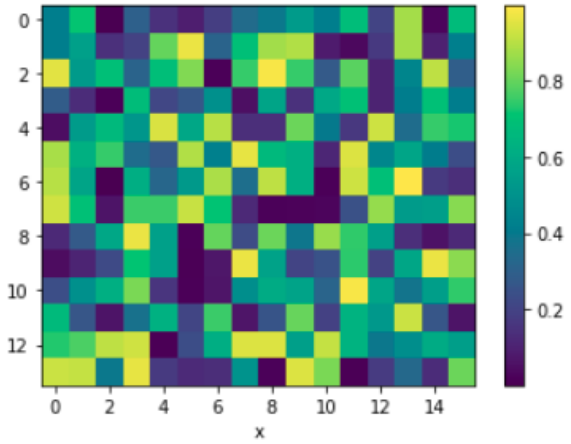




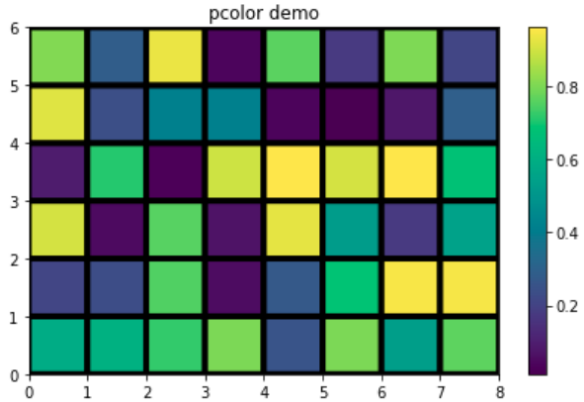
IV. Colorbar –Advanced

노트북 0404 ScalarMappable Colorbar

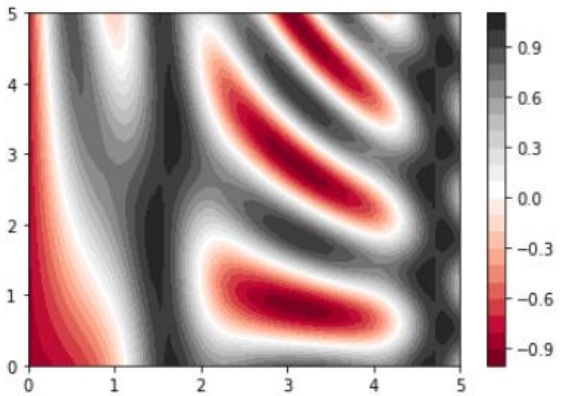
- ScalarMappable - various plots



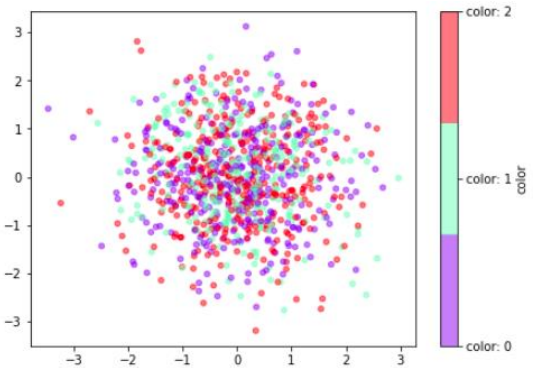
<class 'matplotlib.image.AxesImage'>



<class 'matplotlib.collections.PolyCollection'>



<class 'matplotlib.contour.QuadContourSet'>



<class 'matplotlib.collections.PathCollection'>

https://jakevdp.github.io/PythonDataScienceHandbook/04.04-density-and-contour-plots.html
https://frhyme.github.io/python-lib/matplotlib_extracting_color_from_cmap/

IV. Colorbar –Advanced

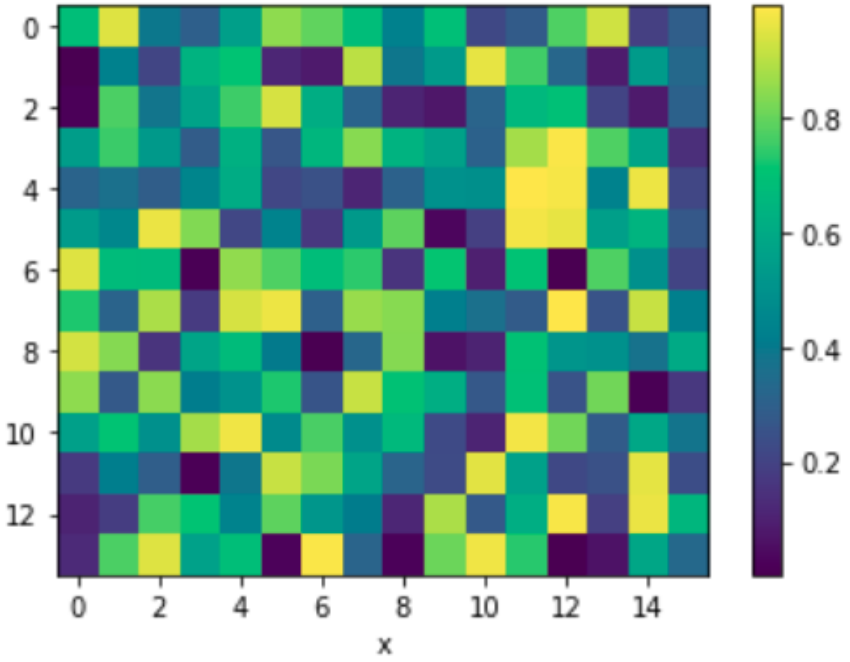
- ScalarMappable - **change vmin / vmax**

Colorbar 의 값의 범위는 (0,1) 로 고정하고 싶다.

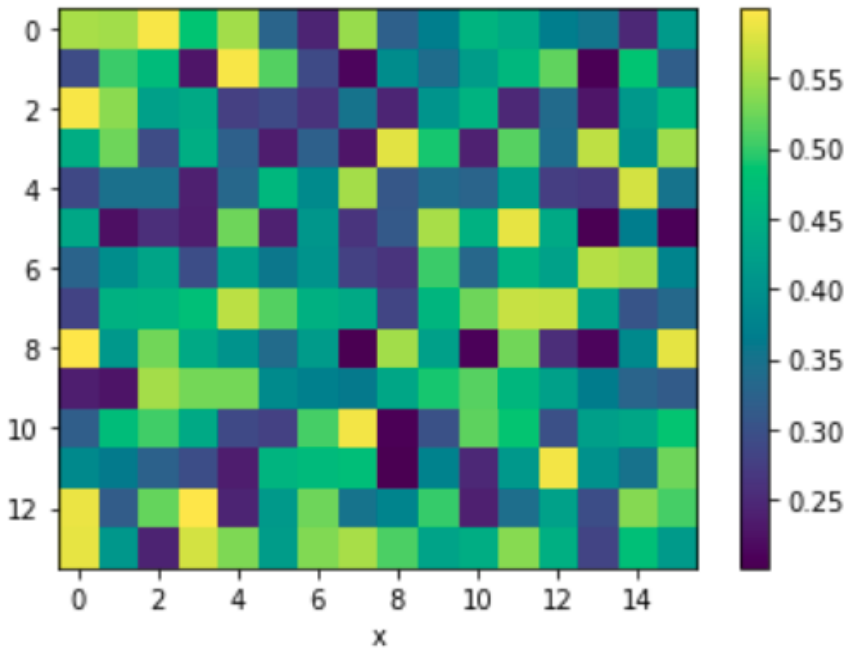
그런데 만약 Data 값이 변하면 어떻게 될까?

이게 원하는 결과?

(0.002,0.998)



(0.201,0.598)



IV. Colorbar –Advanced

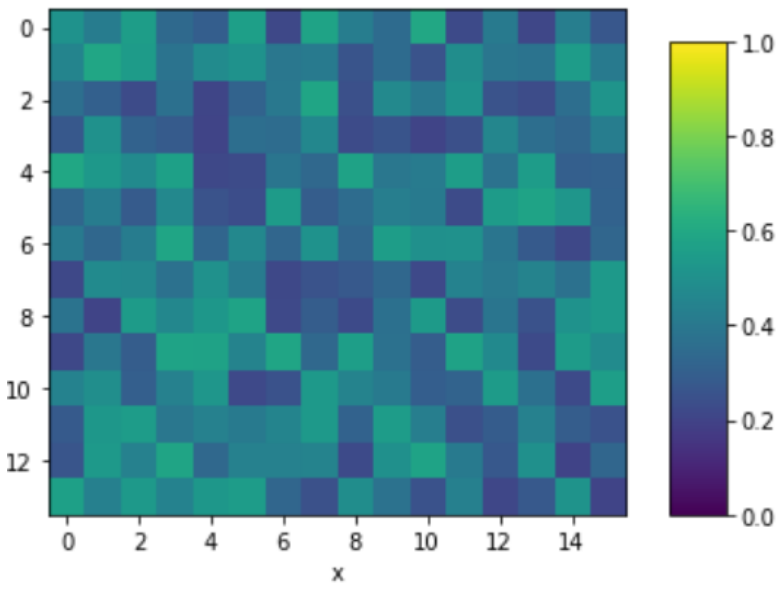
노트북 0405 ScalarMappable Change vmin, vmax

- ScalarMappable - **change vmin / vmax**

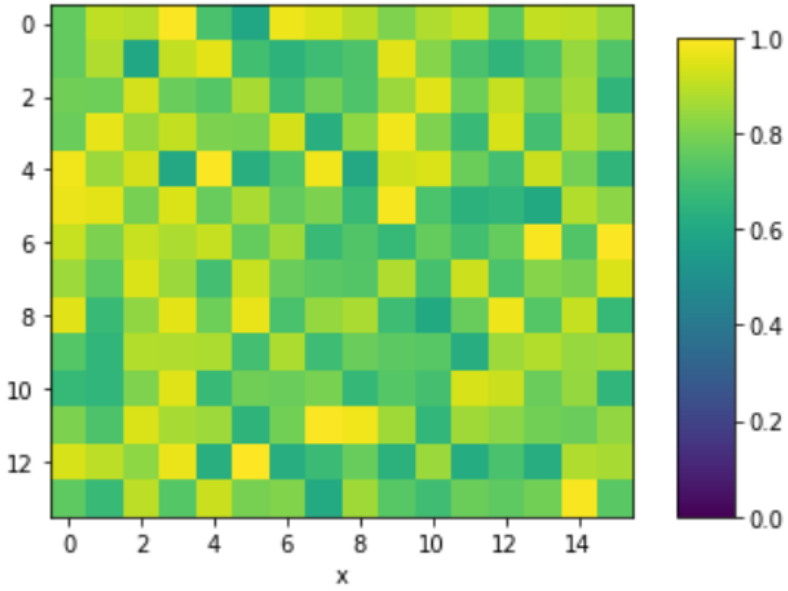
우리가 원하는 것은 데이터 값이 변하더라도 Colorbar 의 색 범위는 변하지 않는 것.

set_clim(vmin, vmax) 를 써서 Colorbar 의 색 범위를 동일하게 유지하자.

data range = [0.2, 0.6]



data range = [0.6, 1.0]



IV. Colorbar –Advanced

노트북 0406 Colormap

- Colormap - **제공되는 컬러 종류 확인**

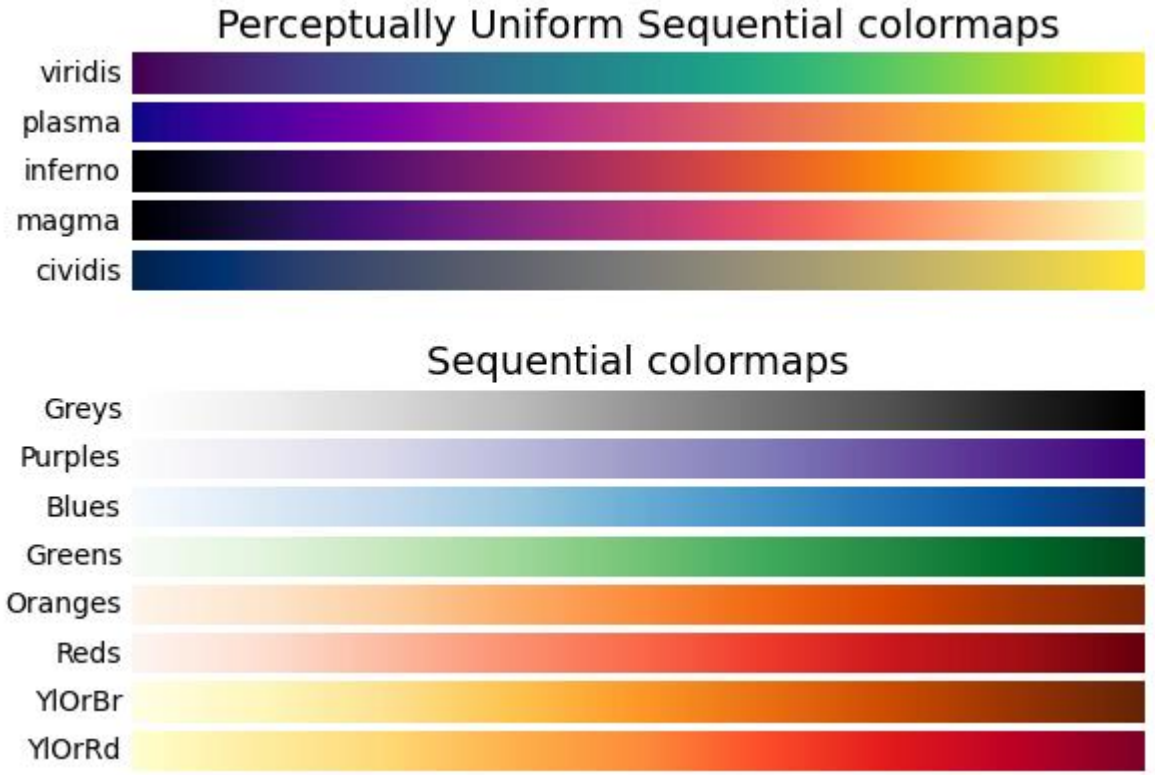
https://matplotlib.org/stable/gallery/color/named_colors.html

IV. Colorbar –Advanced

노트북 0406 Colormap

- Colormap - **제공하는 다양한 Colormap**

https://matplotlib.org/stable/gallery/color/colormap_reference.html

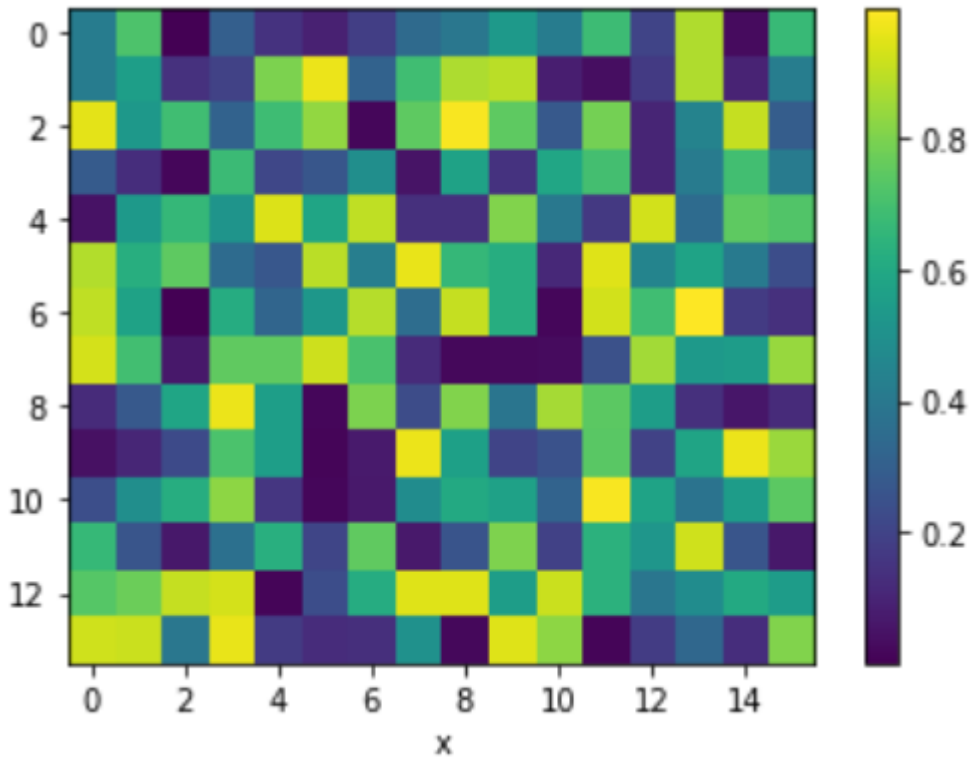


IV. Colorbar –Advanced

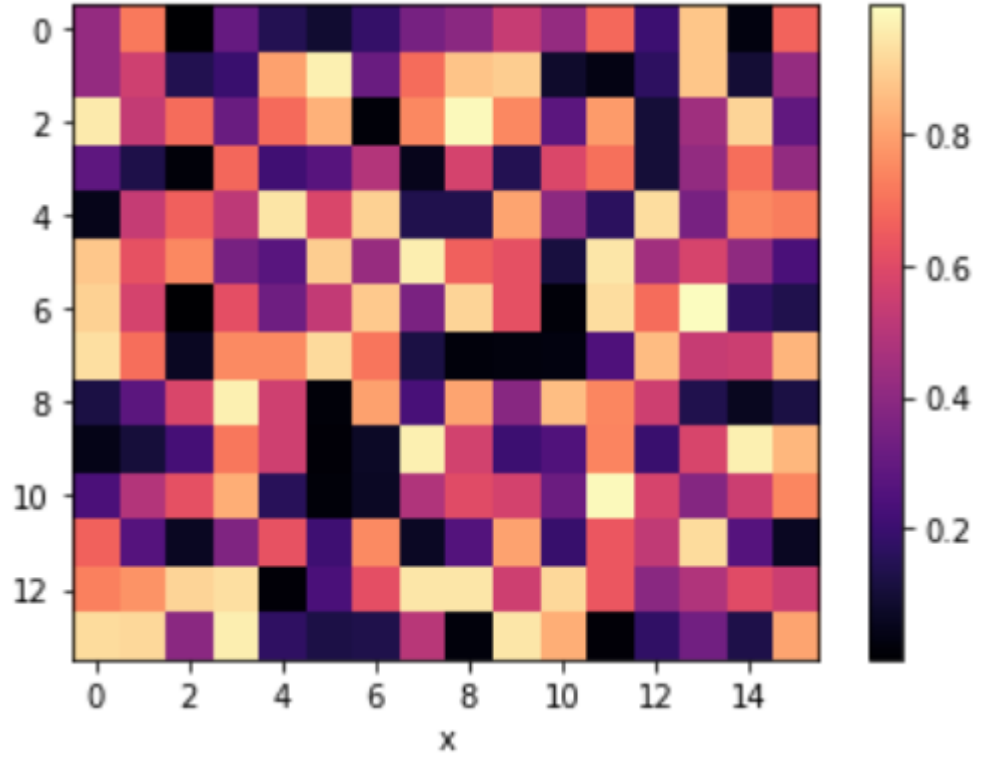
노트북 0406 Colormap

- Colormap - **컬러 맵을 사용해서 colorbar 를 그릴 수 있다.**

cmap = '**viridis**'



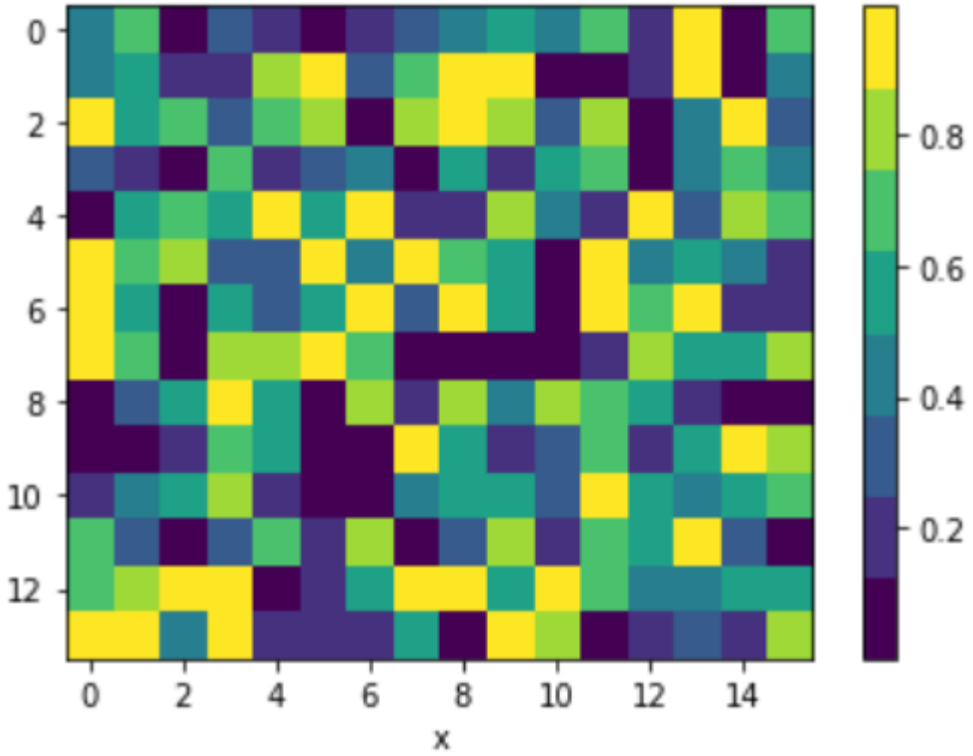
cmap = '**magma**'



IV. Colorbar –Advanced

노트북 0406 Colormap

- Colormap - **named Colormap 에 인자를 주어서 이산적 color 를 표시할 수 있다.**



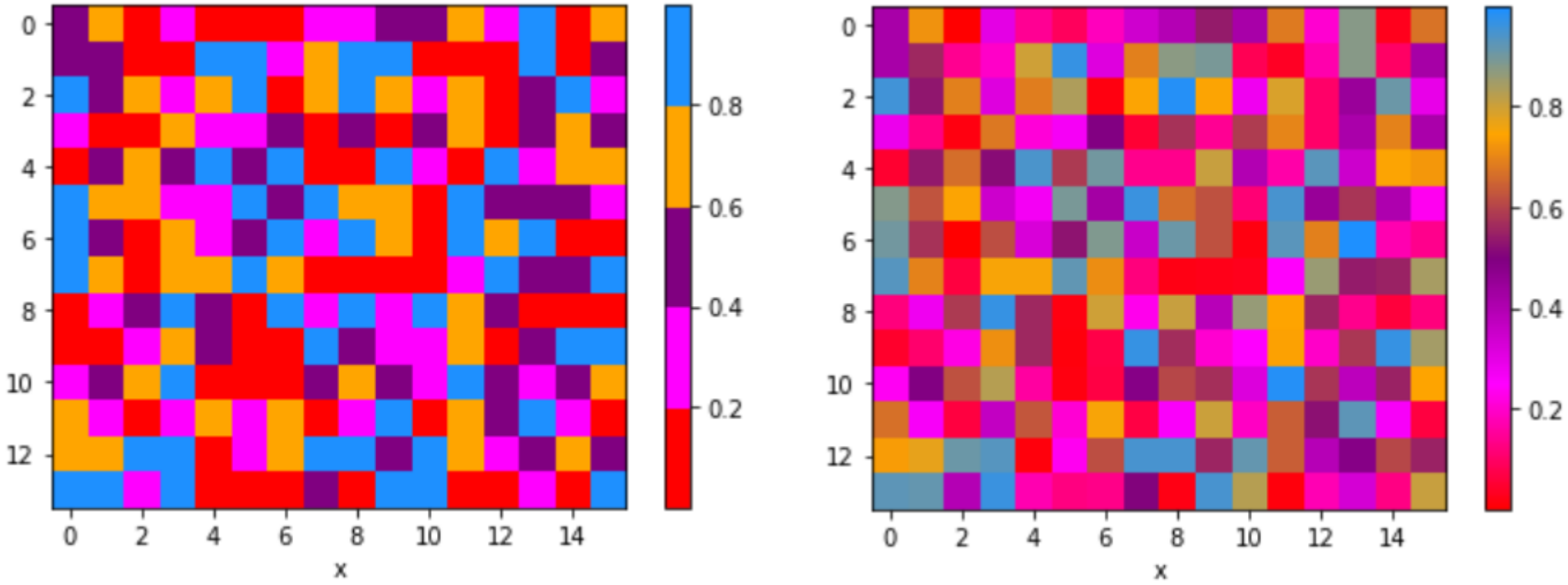

```
from matplotlib import cm

# get_cmap before create ScalarMappable
viridis_discrete = cm.get_cmap('viridis', 8)

im = ax.imshow(data, cmap=viridis_discrete)
```

IV. Colorbar –Advanced

노트북 0406 Colormap

- Colormap - **제공되는 colormap 이 사용하기 지겹다면?** ***Custom Colormap***

✓ ListedColormap, LinearSegmentedColormap 기능 활용



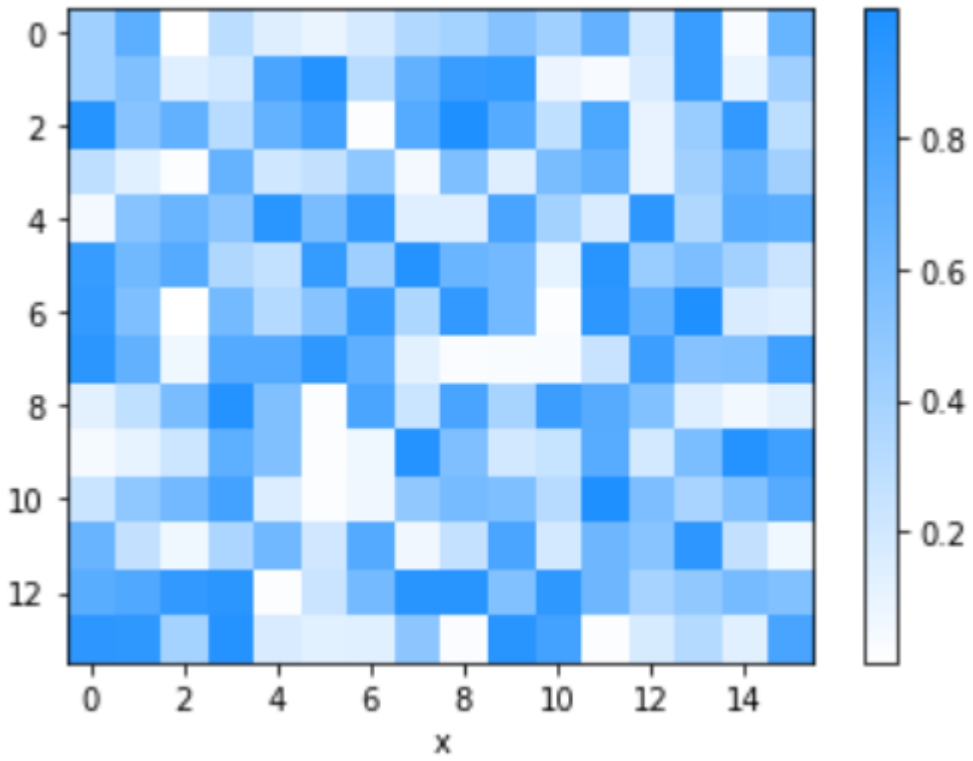
IV. Colorbar –Advanced

노트북 0406 Colormap

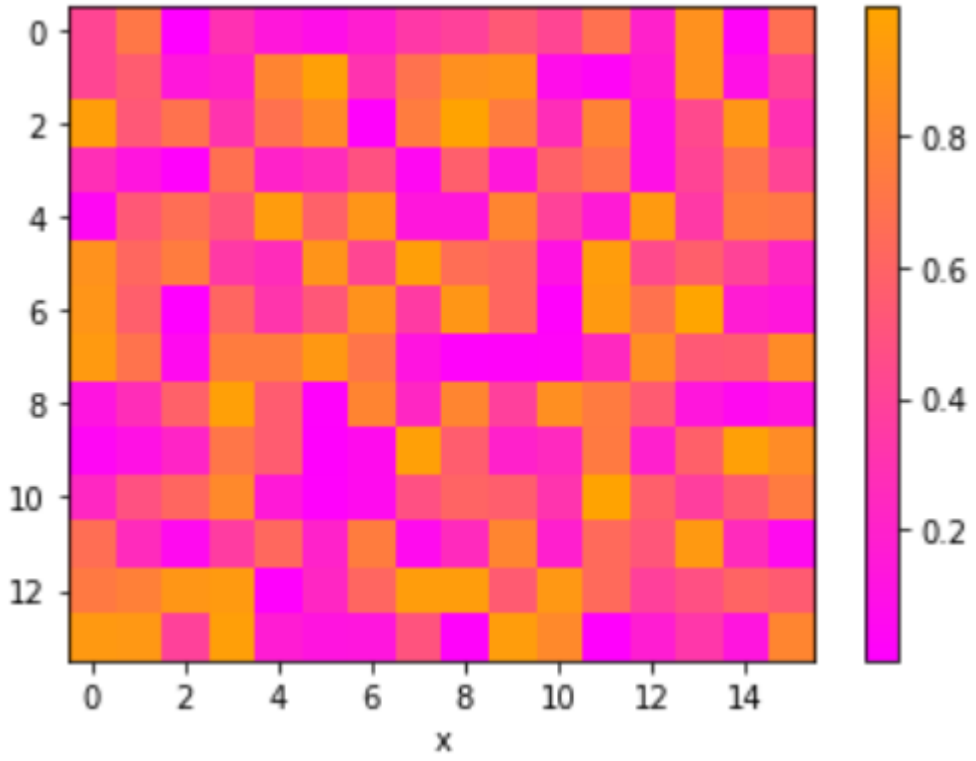
- Colormap - **제공되는 colormap 이 사용하기 지겹다면?** ***Custom Colormap***

  ✓ ListedColormap, LinearSegmentedColormap 기능 활용

["white", "orange"]



["magenta", "orange"]



Ⅴ. Matplotlib 종합

- Artist 객체
- 강수량 예제

V. Matplotlib 종합

- **Artist 객체**

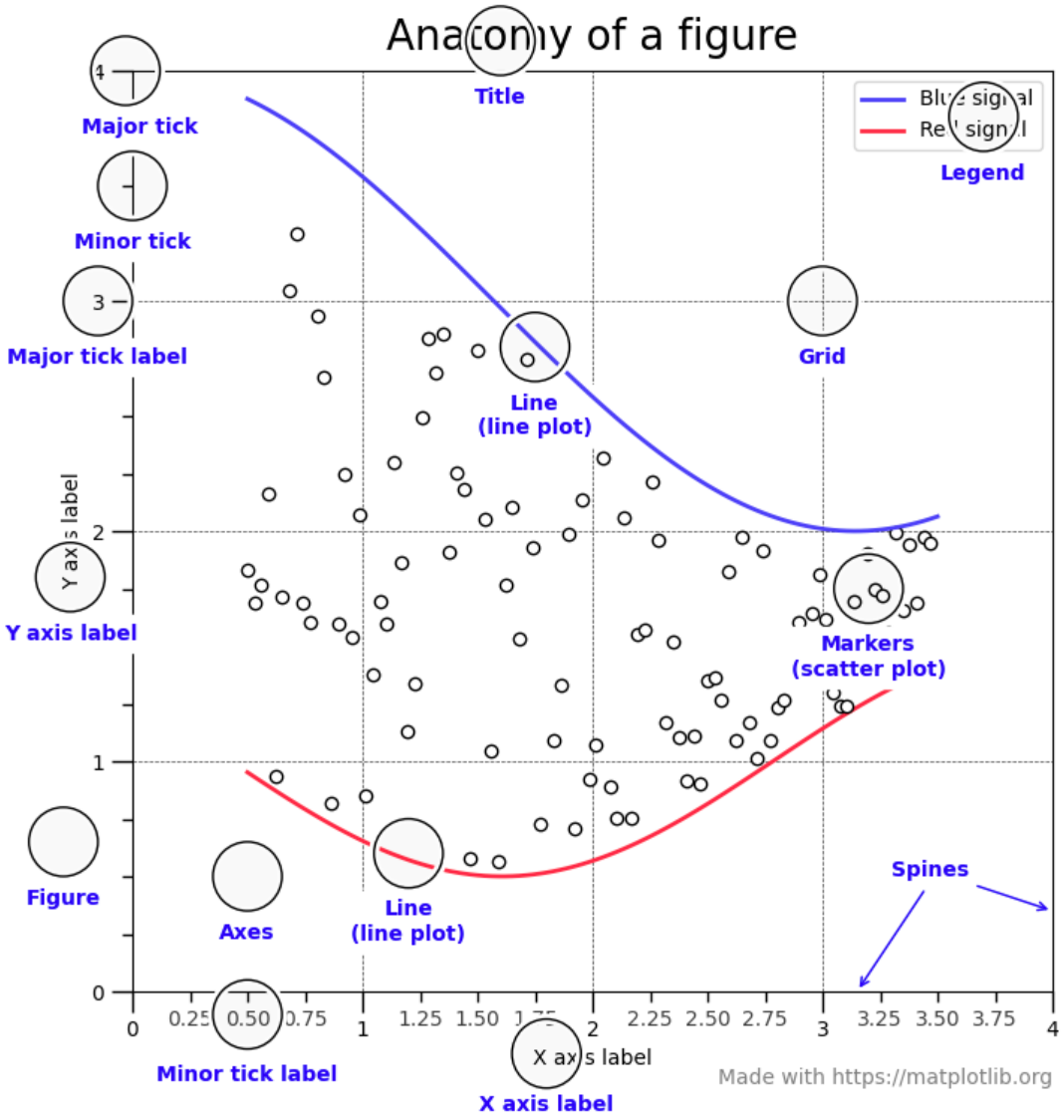


figure 위에 있는 모든 것은 기본적으로 **artist** 객체이다.
text, 2D line, collection object, path object 등

figure 가 rendering 될 때, 모든 artist 들이 canvas 위에 그려진다. 대부분의 artist 들을 Axes 객체에 묶여 있다.

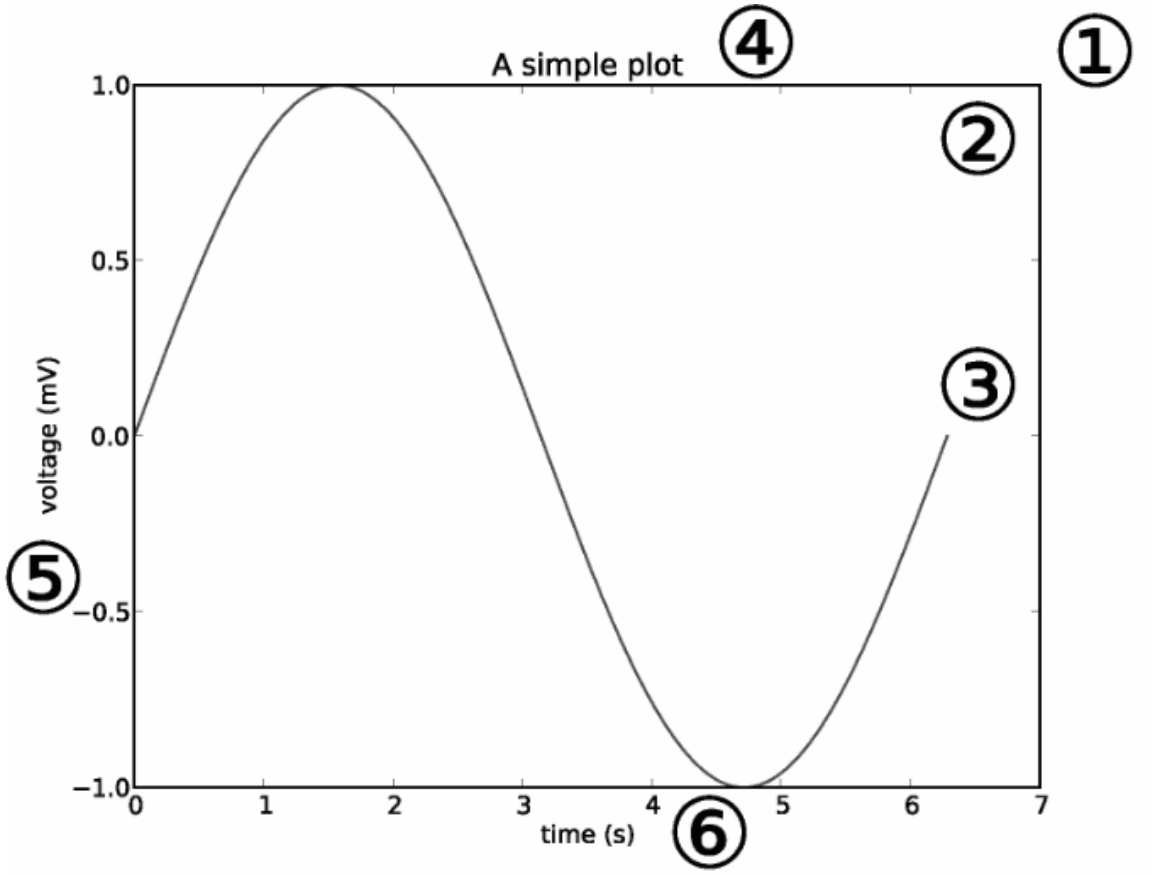
Multiple Axes 에 관여하거나 한 Axes 에서 다른 Axes 로 연결되어 있는 경우를 제외하고 그렇다.

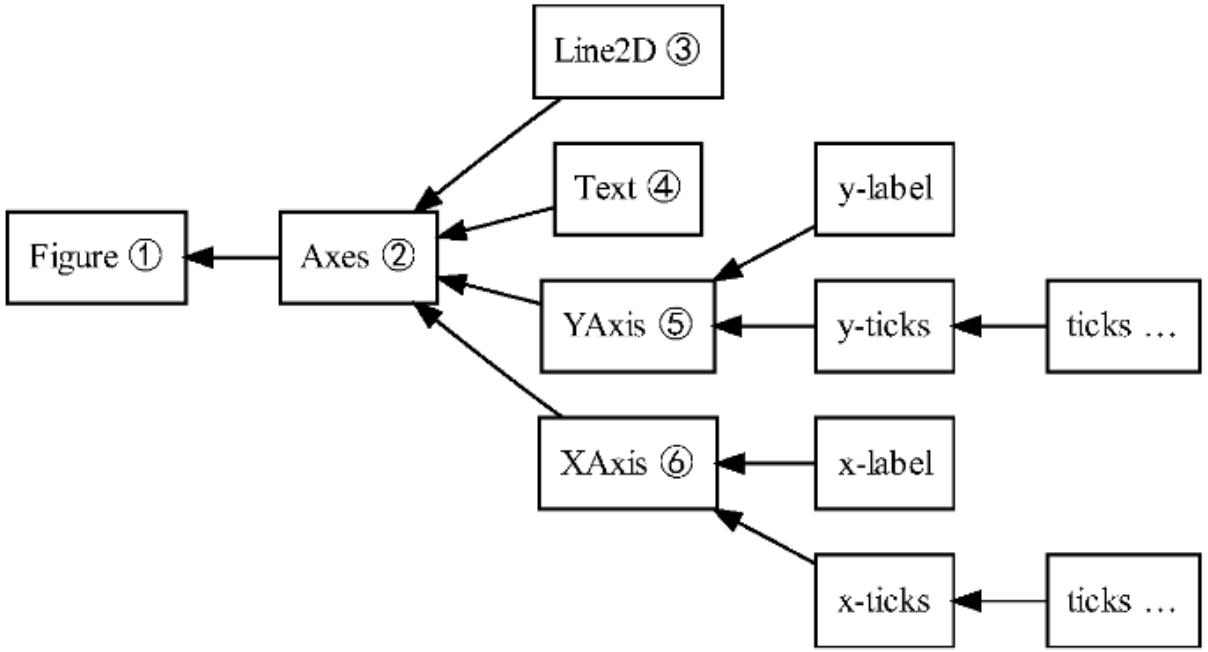
https://matplotlib.org/stable/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py

V. Matplotlib 종합

노트북 0501 Axes.add_artist

artist 객체들이 rendering 되는 순서



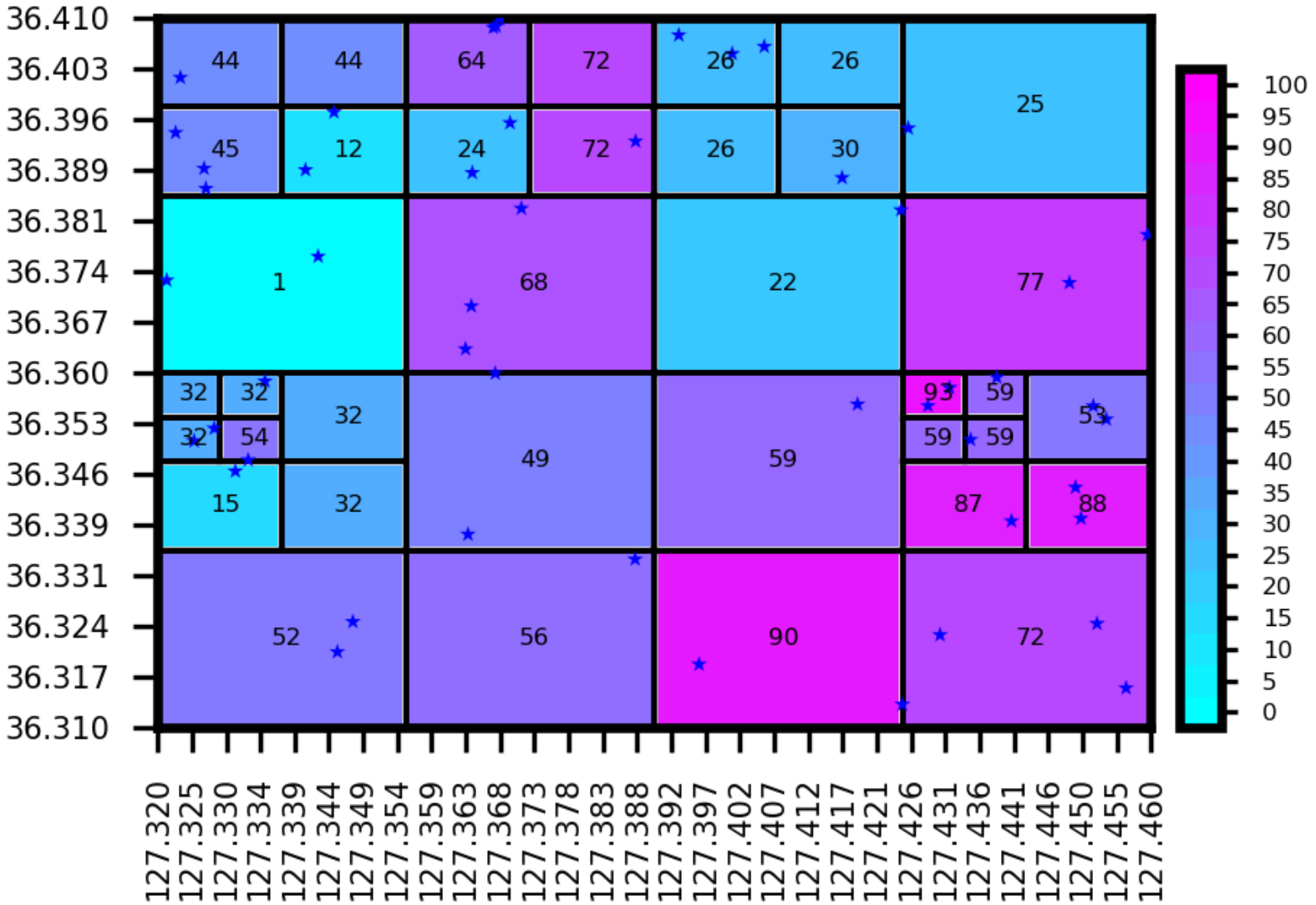


https://www.aosabook.org/en/matplotlib.html

V. Matplotlib 종합

노트북 0502 Precititation

강수량 분포를 QuadTree 로 나타내 보자.

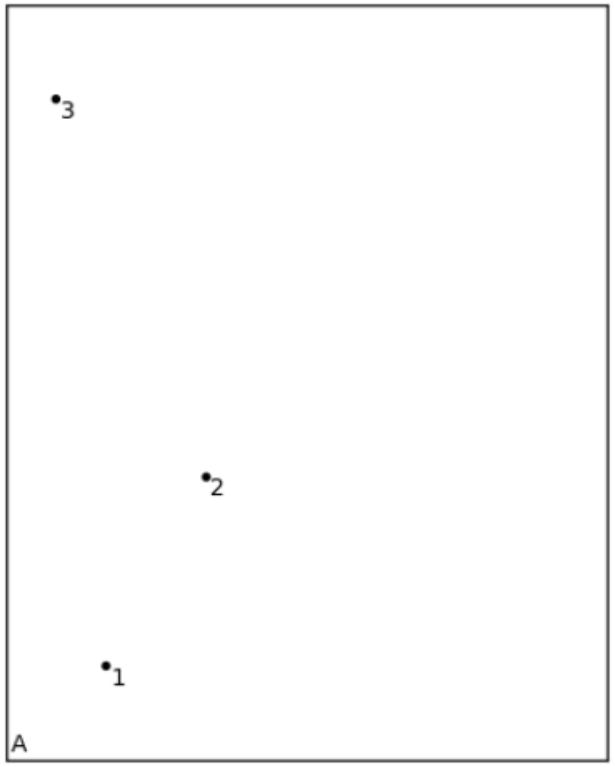


V. Matplotlib 종합

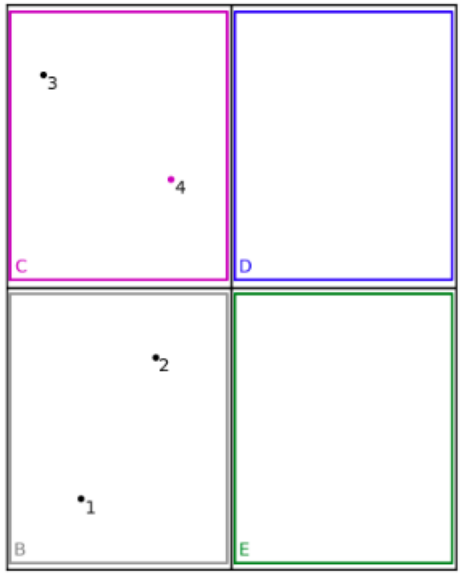
노트북 0502 Precititation

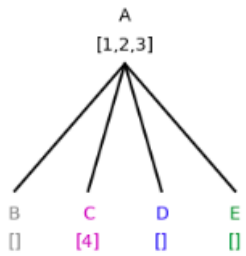
강수량 분포를 QuadTree 로 나타내 보자.

https://scipython.com/blog/quadtree-1-background/

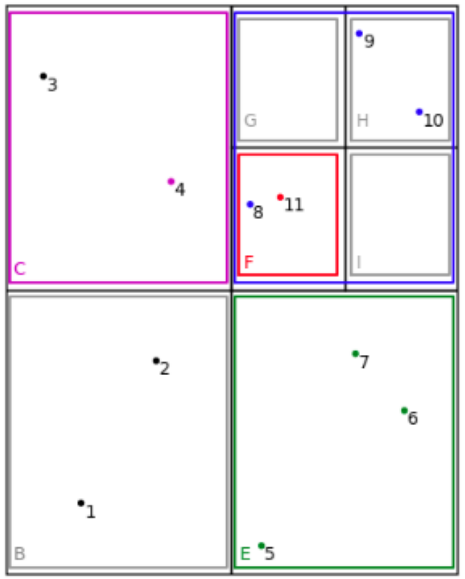


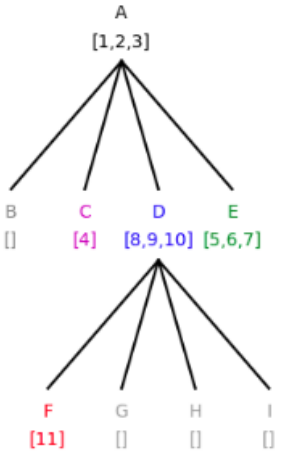
A
[1,2,3]





QuadTree :
2D 영역 4분할





V. Matplotlib 종합

노트북 0502 Precititation

강수량 분포를 QuadTree 로 나타내 보자.

1. Points Generation, ex) (127.367, 36.408, 62.29)
2. Insert Points into QuadTree
3. **Draw ScatterPlot** for each points
4. **Draw Main Contents** : draw each leaf nodes in the QuadTree
   - draw edges in each rectangles
   - draw rectangles with corresponding color of **(PREP-PREP_MIN)/(PREP_MAX - PREP_MIN)**
   - draw a text with the value of PREP
5. Set the locators and formatters for Main Subplot
6. **Draw Colorbar** (ScalarMappable "on the fly")