

matplotlib

matplotlib

- ▶ 파이썬 2D plot 시각화 라이브러리
- ▶ 다양한 저작물hardcopy에 사용할 출판물 수준의 각종 도해figure 제작
- ▶ numpy 배열을 기반으로 구축
- ▶ scipy 스택과 함께 작동되도록 고안
- ▶ 대량의 데이터를 쉽게 이해할 수 있도록 시각화해 줌

matplotlib

- ▶ 선 그래프(line plot)
- ▶ 산점도(scatter plot)
- ▶ 막대 그래프(bar chart)
- ▶ 히스토그램(histogram)
- ▶ 상자수염 그래프(box plot)
- ▶ 파이그래프(pie plot)

matplotlib.pyplot vs pylab

- ▶ matplotlib.pyplot 패키지는 MATLAB의 시각화 명령을 거의 그대로 사용할 수 있도록 Matplotlib의 하위 API를 포장(wrapping)한 명령어 집합 제공
- ▶ pyplot은 matplotlib에서 지원하는 모듈 중 하나
- ▶ pyplot은 state-machine interface 제공
- ▶ pyplot 단순한 코드만 몇 줄로만으로 그래프가 생성

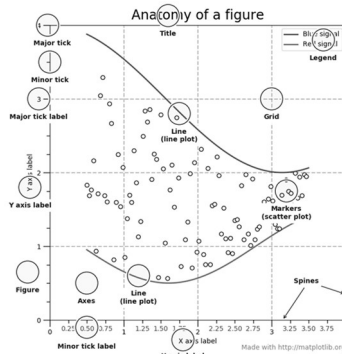
matplotlib.pyplot vs pylab

- ▶ pylab 패키지는 pyplot과 NumPy를 하나의 네임스페이스로 정의해 둔 것
- ▶ MATLAB과 유사하게 각종 함수들을 ‘모듈명.함수명’ 아닌 ‘함수명’ 형태로 바로 호출할 수 있음
- ▶ 편리하긴 하지만 네임스페이스를 오염시킬 우려 존재

matplotlib.pyplot vs pylab

- ▶ import pylab
- 또는
- ▶ import matplotlib.pyplot as plt
- ▶ import numpy as np
- 또는
- ▶ from pylab import *

matplotlib figure



matplotlib interactive mode

- ▶ matplotlib를 파이썬 shell이나 주피터 노트북에서 실행하는 것을 의미
- ▶ 즉, 명령을 라인단위나 블록 단위로 입력해서 실행하고 그 결과를 바로 확인하는 것을 말함

matplotlib interactive mode

- ▶ 대화형 모드는 `pyplot.ion()`에 의해 켜지고 `pyplot.ioff()`에 의해 꺼짐
- ▶ 작성된 그래프를 refresh하려면 `draw()`를 사용
- ▶ 작성된 그래프를 화면에 나타내려면 `show()`를 사용

Line plot

- ▶ 선 그래프는 데이터가 시간, 순서 등에 따라 어떻게 변화하는지 보여주기 위해 사용
- ▶ `pylab` 서브패키지의 `plot` 명령 사용
- ▶ `matplotlib.pyplot.plot(x, y, fmt, **kwargs)`
- ▶ `fmt`는 색깔, 마커/선 유형을 의미하는 문자열

fmt – color

- ▶ 색깔을 지정하는 방법은 색 이름 혹은 약자를 사용하거나 `#` 문자로 시작되는 RGB코드를 사용
- ▶ 자주 사용되는 색깔은 한글자 약자를 사용할 수 있음
- ▶ http://matplotlib.org/examples/color/named_colors.html

fmt – marker

- ▶ 데이터 위치를 나타내는 기호를 마커(marker)라 함
- ▶ 마커를 지정하려면 마커 문자열을 입력해야 함
- ▶ https://matplotlib.org/api/markers_api.html#module-matplotlib.markers

fmt – line

- ▶ 선 모양에는 실선(solid), 대시선(dashed), 점선(dotted), 대시-점선(dash-dot)등이 있음
- ▶ 선모양을 지정하려면 선모양 문자열을 입력해야 함
- ▶ https://matplotlib.org/gallery/lines_bars_and_markers/line_styles_reference.html

**kwargs

https://matplotlib.org/api/_as_gen/matplotlib.lines.Line2D.html#matplotlib.lines.Line2D

속성	입력자료형	설명
label	object	라벨로 입력받는 변수를 문자열형(string, str)으로 지칭하여 라벨을 세팅한다.
linestyle	str	선스타일을 설정하며, fmt에서 line자리에 입력되는 문자열과 동일한 문자열을 입력받는다. 'solid', 'dashed', 'dashdot', 'dotted'로도 입력이 가능하다.
linewidth	float	선의 두께를 설정한다.
color	str	선의 색깔을 설정하며, fmt에서 color자리에 입력되는 문자열과 동일한 문자열을 입력받는다. 색을 지칭하는 영어단어(red, blue 등)으로도 입력이 가능하다.
dashes	sequence(floats)	사용자가 직접 점선(dash line)을 조정할 수 있게 한다. 실수형의 시퀀스(리스트, 튜플 등)를 입력받으며 요소의 수는 반드시 짝수여야 한다.
marker	str	마커스타일을 설정하며, fmt에서 marker자리에 입력되는 문자열과 동일한 문자열을 입력받는다.
markersize	float	마커의 크기를 설정한다.
markevery	None, int, (int, int), list(int), slice, float, (float, float)	데이터에서 마커를 만들 샘플을 추출하도록한다. 예를들어, markevery=20이면 20번째 데이터마다 마커를 생성한다. int로 입력될 경우 입력되는 수에 대응하여 스킵을 하여 마커를 생성한다. float으로 입력될 경우 상대적인 거리에 따라 마커를 생성한다.
visible	bool	입력되는 선 혹은 artist들을 보이게 할지 안할지 결정해주는 변수이다. True로 입력시 보이며, False로 입력시 보이지 않게된다.
fillstyle	str	마커를 채우는 방식을 설정한다. 'full', 'left', 'right', 'bottom', 'top', 'none'을 입력으로 받는다.

Axis

- ▶ plot의 축 범위와 스케일을 지정
- ▶ `xlim(*args, **kwargs)`
- ▶ `ylim(*args, **kwargs)`
- ▶ `xscale(value, **kwargs)`
- ▶ `yscale(value, **kwargs)`

ticks

- ▶ plot의 축상의 위치 표시 지점을 틱(tick)이라고 함
- ▶ 틱에 써진 숫자 혹은 글자를 틱 라벨(tick label)이라 함
- ▶ `xticks(ticks=None, labels=None, **kwargs)`
- ▶ `yticks(ticks=None, labels=None, **kwargs)`
- ▶ `xlabel(xlabel, fontdict=None, labelpad=None, **kwargs)`
- ▶ `ylabel(ylabel, fontdict=None, labelpad=None, **kwargs)`

grid

- ▶ 틱 위치를 잘 보여주기 위해 plot 중간에 그리드 선(grid line)을 그릴 수 있음
- ▶ `gird(b=None, which='major', axis='both', **kwargs)`

title

- ▶ 그래프의 제목을 설정
- ▶ `title(label, fontdict=None, loc='center', pad=None, **kwargs)`
- ▶ `suptitle(t, **kwargs)`

legend

- ▶ 여러 개의 plot을 동시에 그리는 경우에 각 plot이 어떤 자료인지 나타내기 위해 사용
- ▶ 자동기능을 이용한 작성
- ▶ 이미 입력된 플롯 요소에 대한 라벨링을 통한 작성
- ▶ 데이터와 라벨의 직접 매칭
- ▶ `legend(*args, **kwargs)`

legend

위치 문자열	위치 코드 번호
'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

Scatter

- ▶ 다양한 마커와 컬러를 사용하여 만든 그래프
- ▶ `scatter(x, y, s=None, c=None, marker=None, cmap=None, norm=None, vmin=None, vmax=None, alpha=None, linewidths=None, verts=None, edgecolors=None, *, data=None, **kwargs)`

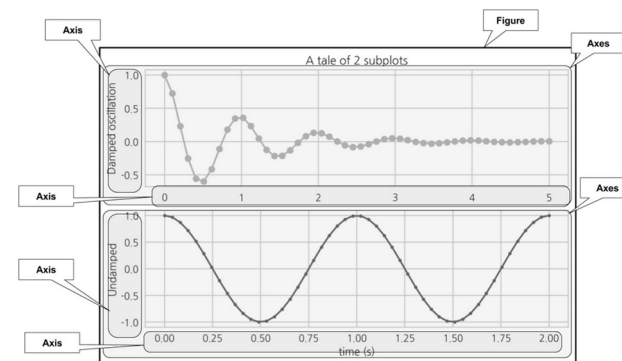
Histogram

- ▶ 도수 분포의 상태를 막대 모양으로 표현한 그래프
- ▶ `hist(x, bins=None, range=None, density=None, weights=None, cumulative=False, bottom=None, histtype='bar', align='mid', orientation='vertical', rwidth=None, log=False, color=None, label=None, stacked=False, normed=None, *, data=None, **kwargs)`

Bar

- ▶ 값의 크기별로 막대 모양으로 표현한 그래프
- ▶ 가로 막대 그래프는 `barh()` 함수를 사용
- ▶ `bar(x, height, width=0.8, bottom=None, *, align='center', data=None, **kwargs)`

Figure, Axes, Axis



Figure, Axes, Axis

- ▶ matplotlib의 plot은 Figure, Axes, Axis 객체 등으로 구성
- ▶ Figure > Axes > Axis 순으로 포함

Figure, Axes, Axis

- ▶ 모든 plot은 Figure 객체의 하위요소
- ▶ 하나의 Figure는 하나의 아이디 숫자와 윈도우(Window)를 가짐
- ▶ figure 명령을 명시적으로 사용하는 경우 여러 개의 윈도우를 동시에 띄울 수 있음
- ▶ `figure(num=None, figsize=None, dpi=None, facecolor=None, edgecolor=None, frameon=True, FigureClass=<class 'matplotlib.figure.Figure'>, clear=False, **kwargs)`

subplot

- ▶ 하나의 윈도우(figure)안에 여러 개의 plot을 나타내야 할 때가 있음
- ▶ figure 속 각각의 plot은 Axes 객체에 속함
- ▶ figure에 Axes를 생성하려면 subplot 명령 사용
- ▶ plot명령을 사용하면 자동으로 Axes를 생성해 줌
- ▶ subplots 명령은 2차원 ndarray 형태로 Axes 객체가 반환됨

subplot

- ▶ `subplot(nrows, ncols, index, **kwargs)`
- ▶ `subplot(pos, **kwargs)`
- ▶ `subplot(axes)`
- ▶ plot간의 간격을 자동으로 맞춰 줌
- ▶ `tight_layout(pad=1.08, h_pad=None, w_pad=None, rect=None)`