

# 파이썬 시각화

with  python™

# 01. 데이터 시각화의 중요성



Florence Nightingale (1820 ~ 1910)

# 01. 데이터 시각화의 중요성



Florence Nightingale (1820 ~ 1910)

간호사  
(크림전쟁)

저술활동  
(간호를 위하여)

통계학자

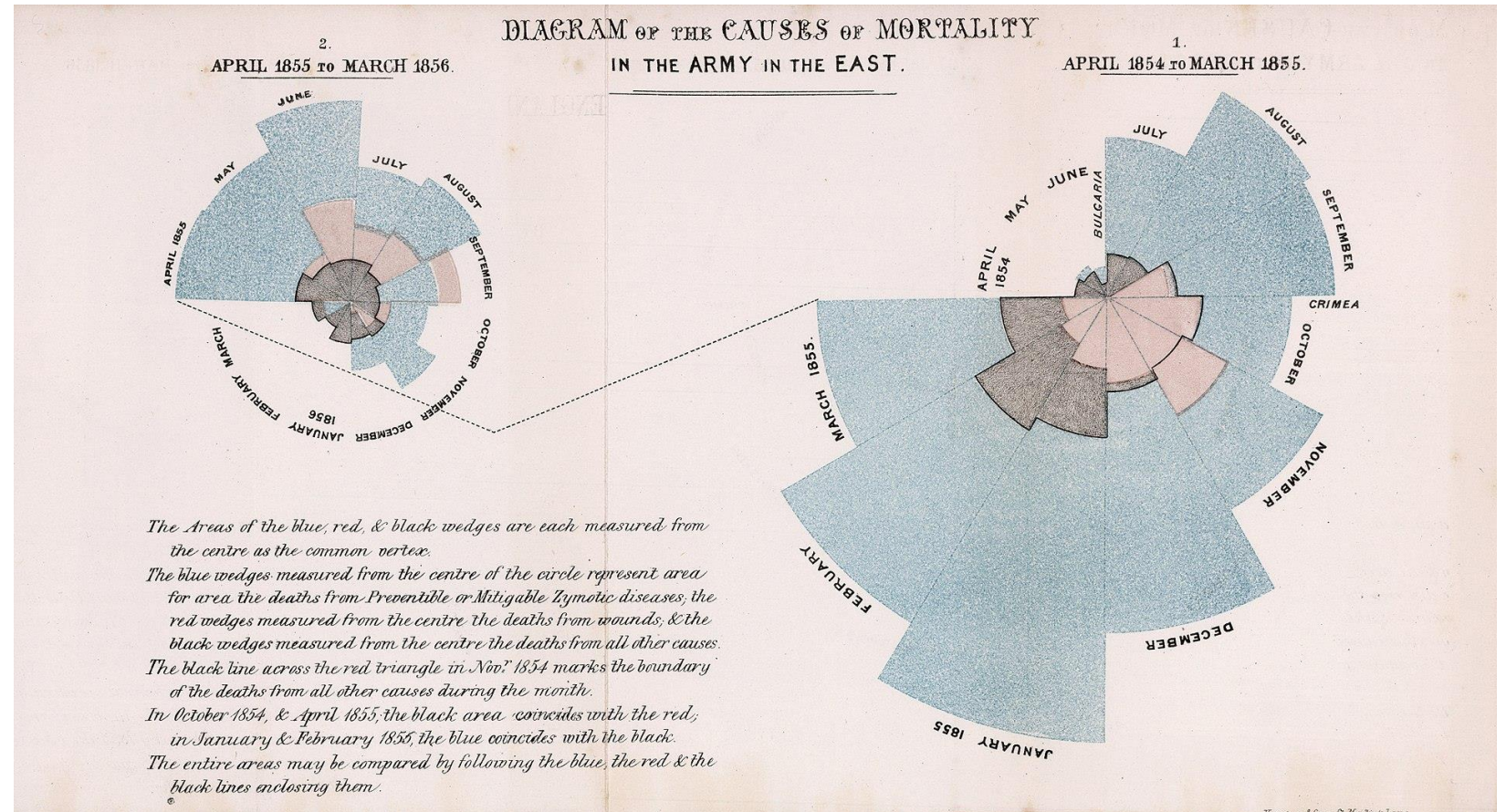


# 01. 데이터 시각화의 중요성

## “로즈 다이어그램 Rose Diagram”



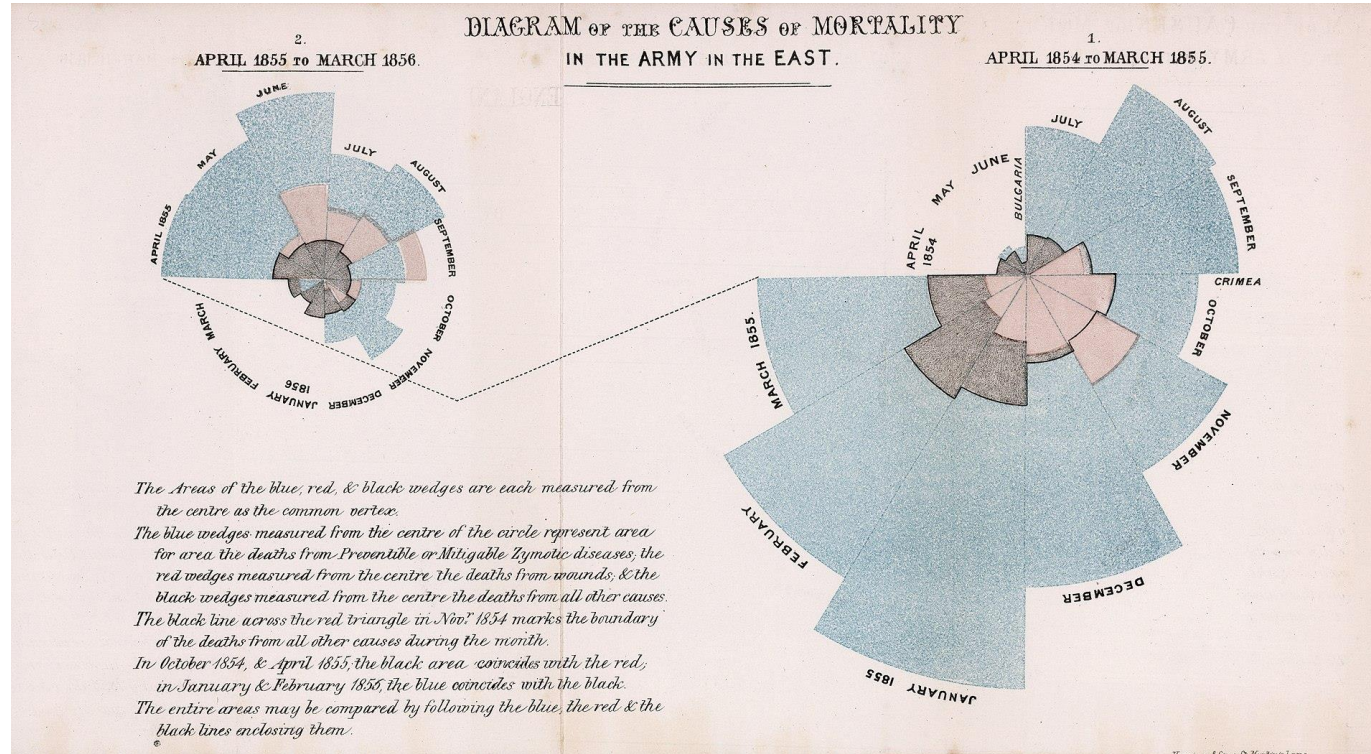
Florence Nightingale (1820 ~ 1910)



출처: <https://commons.wikimedia.org/wiki/File:Nightingale-mortality.jpg>



# 01. 데이터 시각화의 중요성



“ 전쟁보다 전염병에 의한 사망이 더 많으니 야전병영 위생을 개선해야 합니다! ”

# 02. 수치 데이터 분석과 시각화 분석의 조합

## 앤스컴 콰르텟(Anscombe's Quartet)

I		II		III		IV	
<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.5
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

출처: [https://en.wikipedia.org/wiki/Anscombe%27s\\_quartet](https://en.wikipedia.org/wiki/Anscombe%27s_quartet)

”

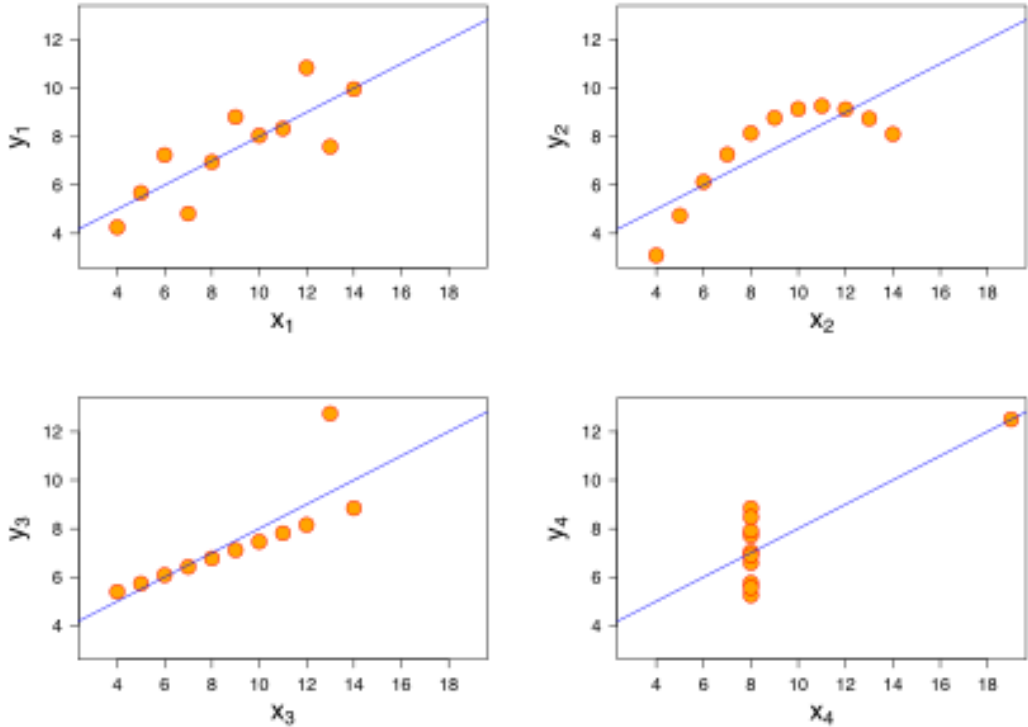
# 02. 수치 데이터 분석과 시각화 분석의 조합

## 앤스컴 콰르텟(Anscombe's Quartet)

동일한 기술 통계량

항목	값
x 평균	9
x 표본분산	11
y 평균	7.50
y 표본분산	4.125
x와 y의 상관	0.816
선형회귀선	$y = 3.00 + 0.500x$
선형회귀 결정계수	0.67

4개의 다른 시각화



”

### 03. 데이터 시각화의 원리

삭제  
(Delete)

분리  
(Divide)

강조  
(Highlight)

배열  
(Arrange)



# 03. 데이터 시각화의 원리

◆ 시각화의 기본 원리 : 삭제 > 분리 > 강조 > 배열



시각화 원리	설명
삭제 (cut, Delete)	필수적인 데이터 남기고, 의미 없는 Chart는 삭제
분리 (Divide)	데이터 분리한 뒤, 별도 Chart로 만든 후, 논리 순서로 배열
강조 (Highlight)	필수 데이터는 강조, 부가적 데이터는 약하게 또는 숨김 처리 (대조)
배열 (Arrange)	데이터를 영역 별로 그룹핑 한 후, 영역 간 및 영역 내 데이터 간 논리 순서로 구조화

Gene Zelazny, <Say it with Charts>

# 04. 데이터 차트의 종류

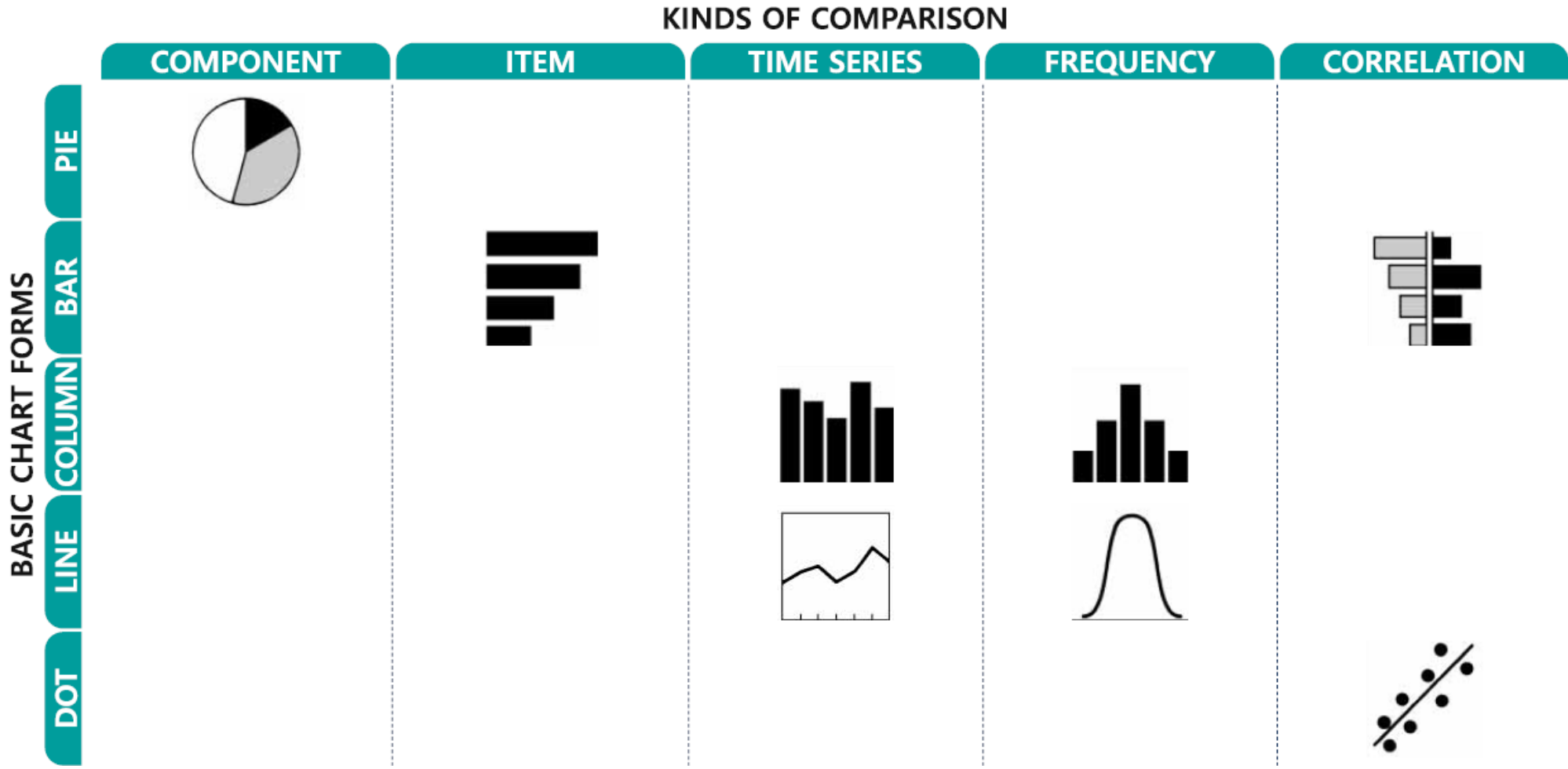
◆ Chart 종류 선택

메시지	데이터의 유형	차트 형식
'비율', '퍼센트', '비중'	구성 요소(Component) 비교 : 백분율	Pie, Stacked Column
'~보다 많음', '~보다 적음'	항목(Item) 비교 : 항목의 순위	Bar, Waterfall
'변화', '성장', '변동'	시간적 추이(Time Series) 비교	Column or Line 차트
'분포는~'	빈도분포(Frequency) 비교	Column or Line 차트
'~에 관련된다', '~에 따라 변화'	상관관계 (Correlation) 비교	Scatter or Paired Bar Chart

Gene Zelazny, <Say it with Charts>

# 04. 데이터 차트의 종류

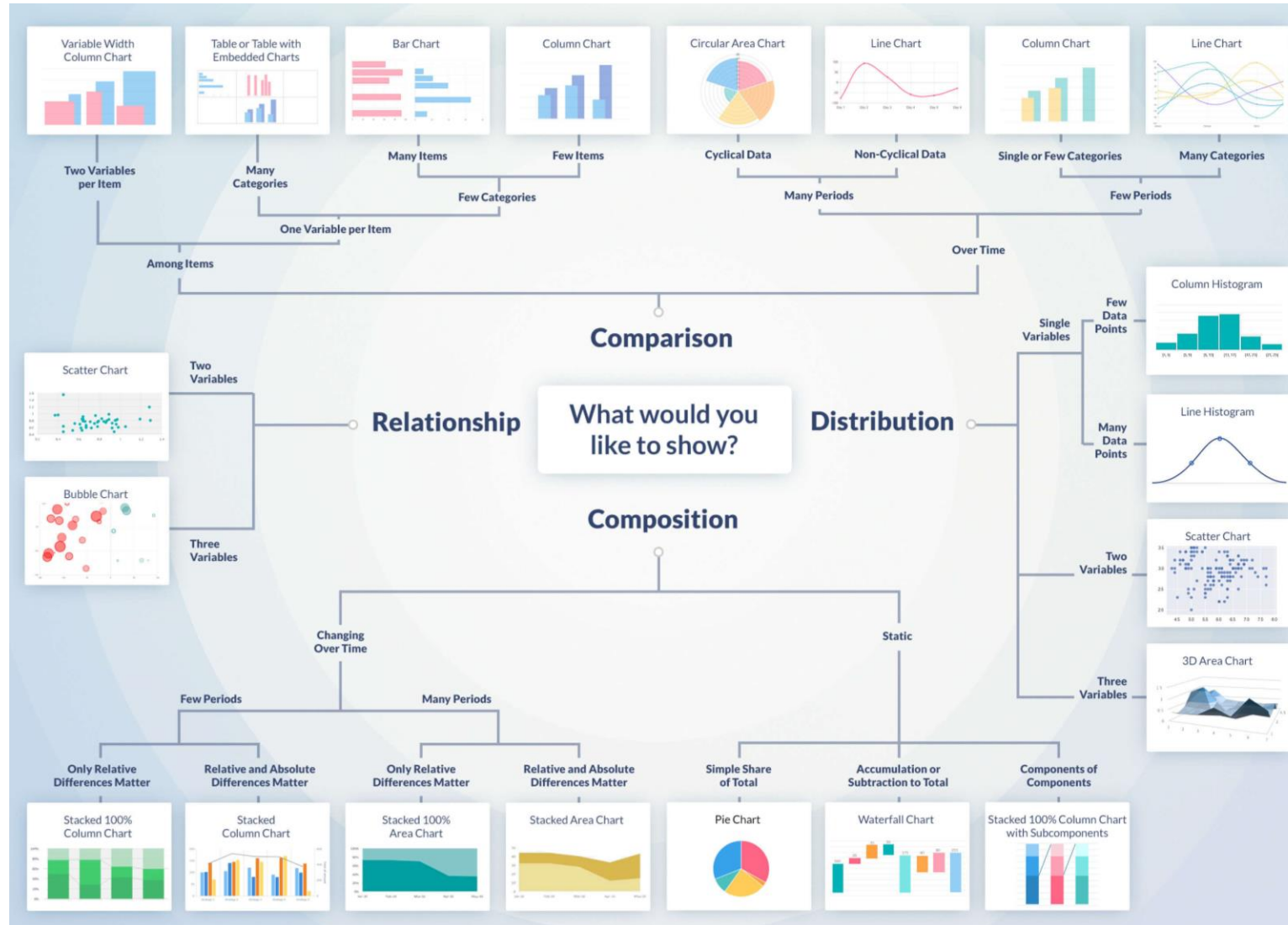
◆ Chart 종류 선택



Gene Zelazny, <Say it with Charts>

# 04. 데이터 차트의 종류

## ◆ Chart 종류 선택

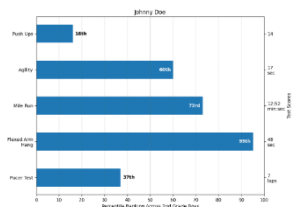
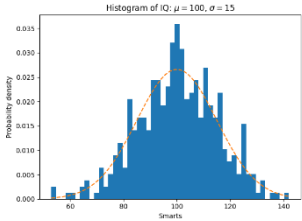
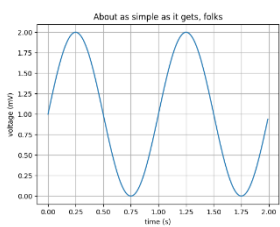


출처 : <https://www.tapclicks.com/wp-content/uploads/How-to-Visualize-your-Data-with-Charts-and-Graphs.jpg>

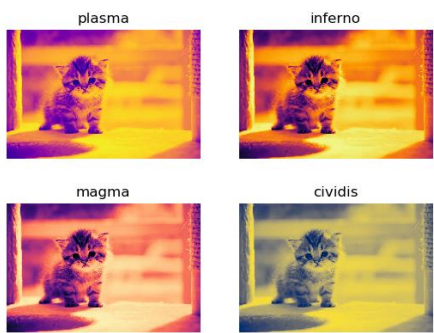
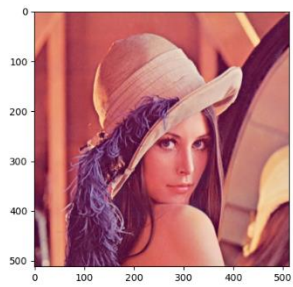
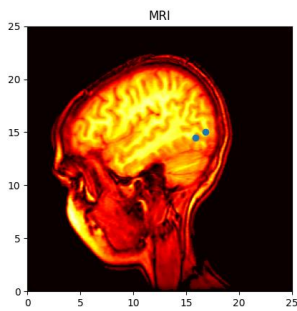
# 05. matplotlib + seaborn



정형 데이터 시각화



이미지 데이터 시각화

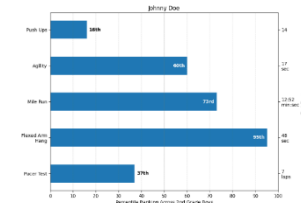
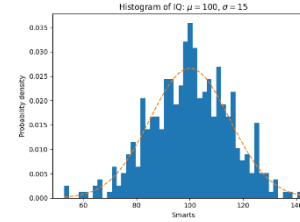
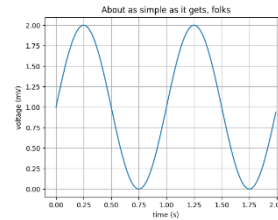




# 05. matplotlib + seaborn

matplotlib

정형 데이터 시각화



## - 보완 및 장점

- ① 비 전공자들에게 matplotlib 시각화 문법은 조금 어렵다.
- ② pandas 데이터 프레임에서 쉽게 시각화 구현 가능 하도록 한다.
- ③ 통계 (회귀선) 그래프 등을 쉽게 구현할 수 있도록 한다.

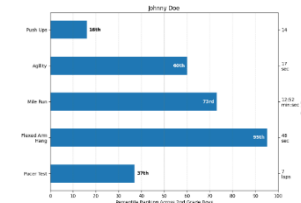
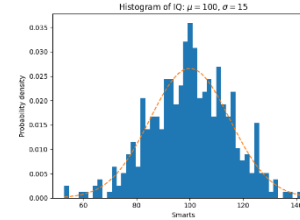
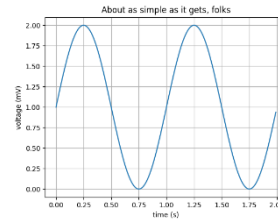


seaborn

# 05. matplotlib + seaborn

정형 데이터 시각화

matplotlib



## – 보완 및 장점

- ① 비 전공자들에게 matplotlib 시각화 문법은 조금 어렵다.
- ② pandas 데이터 프레임에서 쉽게 시각화 구현 가능 하도록 한다.
- ③ 통계 (회귀선) 그래프 등을 쉽게 구현할 수 있도록 한다.



seaborn

## – 단점

- ① 세부 옵션을 수정 하려면 Matplotlib을 알아야 한다.

# 05. matplotlib + seaborn

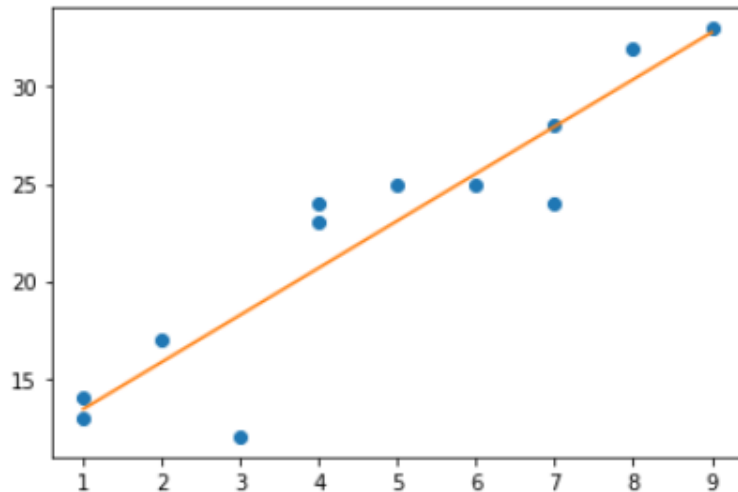
matplotlib

```
[1]: import numpy as np
import matplotlib.pyplot as plt

# 데이터 생성
x = np.array([1, 1, 2, 3, 4, 4, 5, 6, 7, 7, 8, 9])
y = np.array([13, 14, 17, 12, 23, 24, 25, 25, 24, 28, 32, 33])

m, b = np.polyfit(x, y, 1)
plt.plot(x, y, 'o')
plt.plot(x, m*x+b)
```

```
[1]: [<matplotlib.lines.Line2D at 0x226212833d0>]
```



seaborn

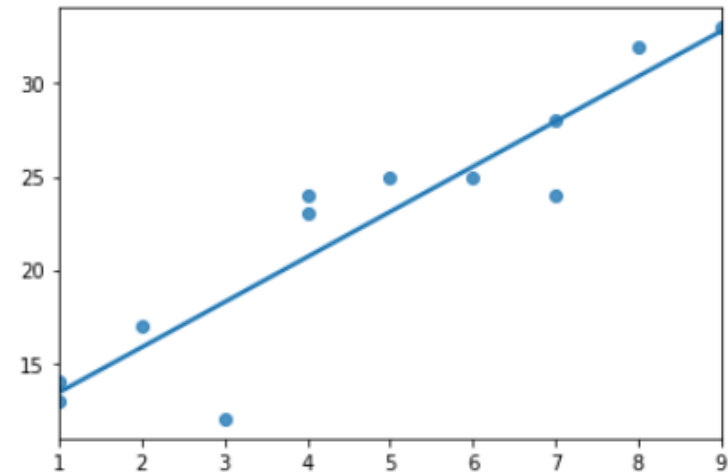
```
[1]: import numpy as np
import seaborn as sns

# 데이터 생성
x = np.array([1, 1, 2, 3, 4, 4, 5, 6, 7, 7, 8, 9])
y = np.array([13, 14, 17, 12, 23, 24, 25, 25, 24, 28, 32, 33])

sns.regplot(x, y, ci=None)
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\\_decorators.py  
nly valid positional argument will be `data`, and passing other a  
warnings.warn(

```
[1]: <AxesSubplot:>
```



## 05. matplotlib + seaborn

### ◆ 두 라이브러리 장점 위주 혼용

- ✓ 시각화 라이브러리 **seaborn**으로 기본 색상 + 글꼴 설정
- ✓ 시각화 라이브러리 **matplotlib**으로 시각화 틀 잡고 시각화 수행
- ✓ 각 라이브러리 특성에 맞추어서 선택적으로 활용하는 것 추천

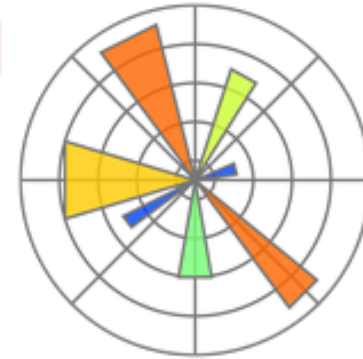


★

★ ★ ★ ★

간단하고 깔끔하게 vs 구석구석 섬세하게  
통계전문 시각화 vs 아무거나 시각화  
밀도 함수 etc. vs 영상 해석 etc.

일단 그릴 때 명령어 난이도  
손을 많이 댈 때 명령어 난이도

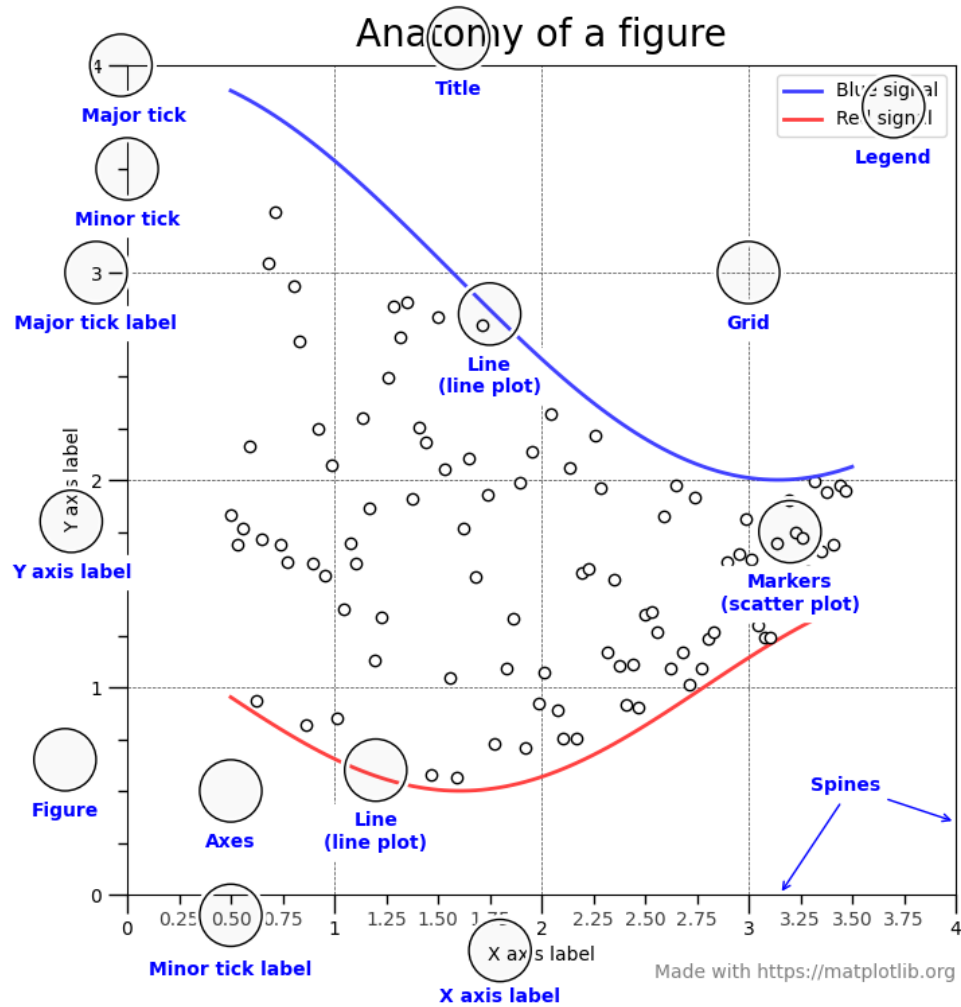


★ ★

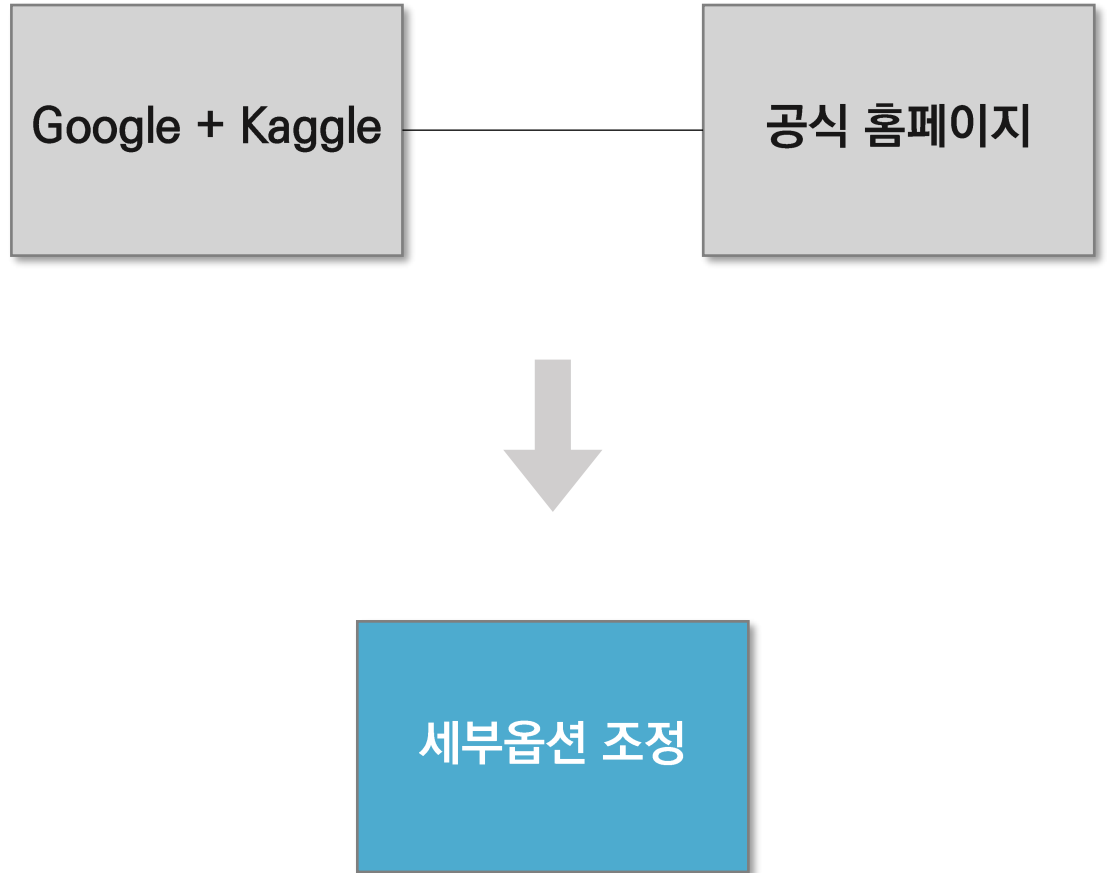
★ ★ ★

추천 블로그 : [https://jehyunlee.github.io/2020/10/10/Python-DS-37-seaborn\\_matplotlib4/](https://jehyunlee.github.io/2020/10/10/Python-DS-37-seaborn_matplotlib4/)

## 06. 세부 옵션



출처: <https://matplotlib.org/stable/gallery/showcase/anatomy.html>





# 07. 기본 코드 작성

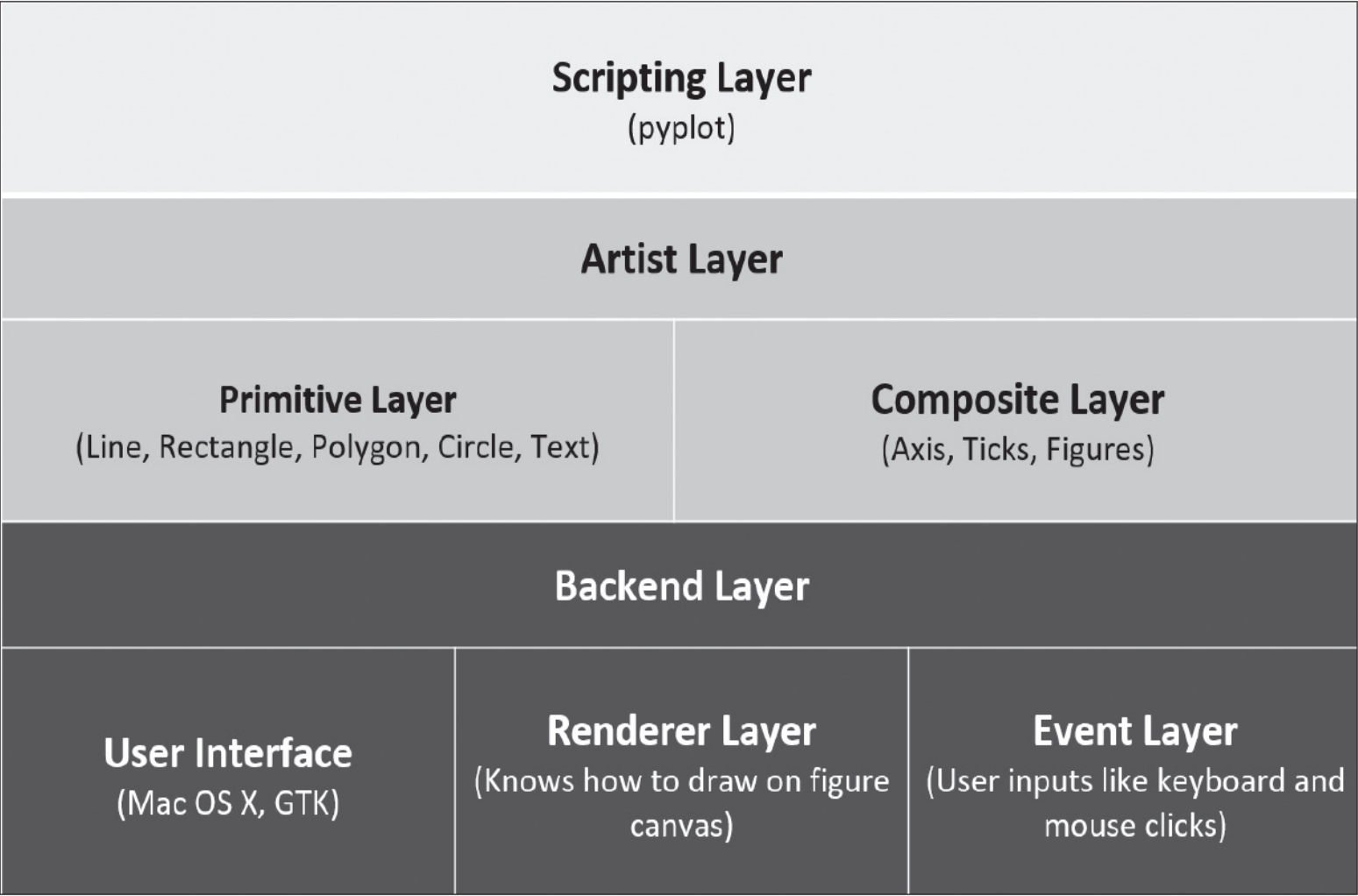
◆ 객체 지향 방식으로 코드 작성 시작 추천

```
1 : import matplotlib.pyplot as plt
2 : fig, ax = plt.subplots(figsize=(10, 6))
3 : plt.show()
    (Graph)
```

주요 용어	설명
fig(figure의 약어)	<ul style="list-style-type: none"><li>matplotlib에서 최상위 컨테이너 또는 창이며, plot을 만드는 일종의 캔버스</li><li>하나의 figure에 하나 이상의 plot()을 포함할 수 있고, 다중 패널 또는 하위 plot 레이어아웃을 그릴 수 있음</li></ul>
ax(axes의 약어)	<ul style="list-style-type: none"><li>데이터를 그릴 수 있는 figure내의 subplot 또는 특정 영역</li><li>제목(title), 레이블(labels), 격자선(gridlines) 등 같은 속성을 독립적으로 사용자 지정할 수 있음</li></ul>

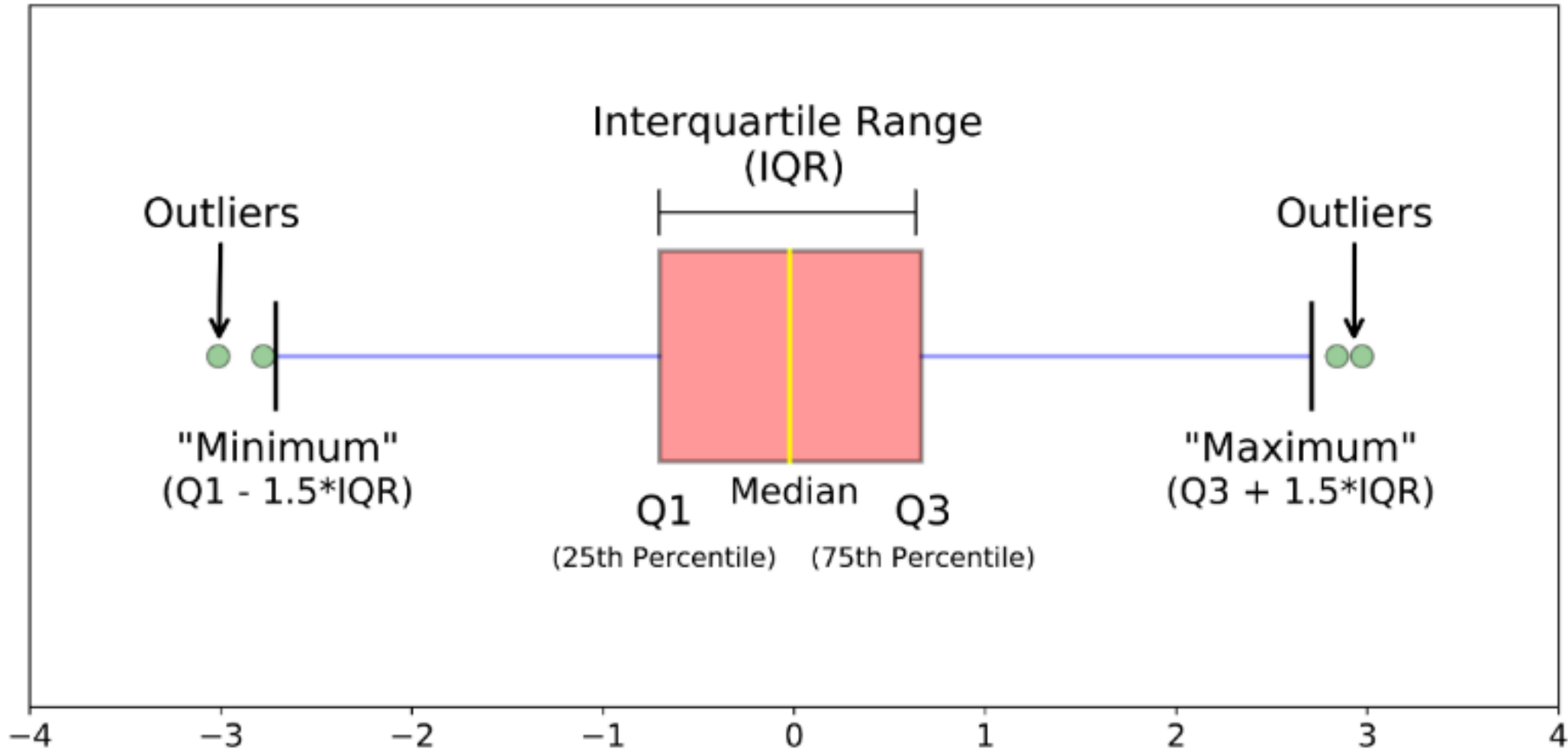
# 08. matplotlib architecture

◆ matplotlib 라이브러리의 아키텍처



## 09. boxplot

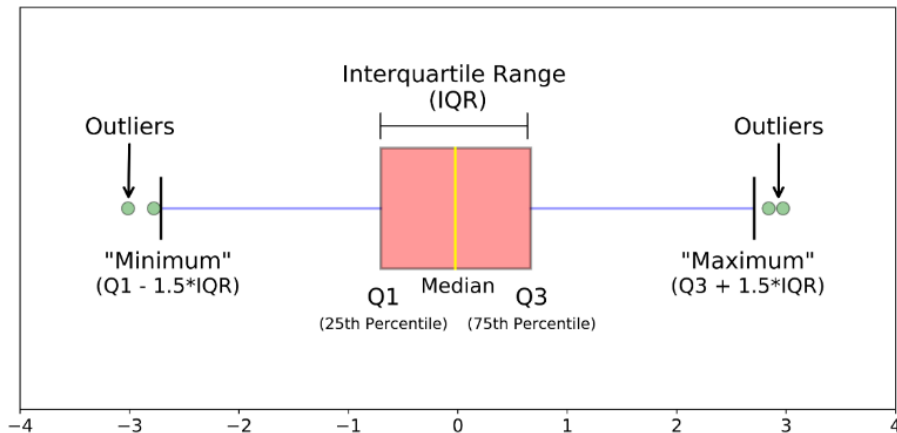
◆ 박스플롯(Box Plot) : 데이터의 분포와 이상치(outlier)를 동시에 보여주는 시각화



# 09. boxplot

◆ 박스플롯(Box Plot) : 데이터의 분포와 이상치(outlier)를 동시에 보여주는 시각화

구분	내용
Q1(제1사분위수)	데이터의 하위 25%에 해당하는 값
Q3(제3사분위수)	데이터의 상위 25%에 해당하는 값
IQR(사분위 범위)	상자의 길이는 사분위 범위를 나타내며 IQR은 데이터의 중간 50% 범위를 의미
중앙값(Median)	데이터의 중간 값을 나타냄
수염(Whiskers)	데이터의 변동 범위를 나타내며, $1.5 * IQR$ 규칙을 사용하여 그려짐
이상치(Outliers)	수염의 바깥에 위치하는 데이터 포인트, 일반적인 분포에서 벗어난 값들을 나타냄



## Chapter 03. Matplotlib & Seaborn

















**강의 실습 영상 참고**



# 10. plotly

## ◆ plotly 소개

- ✓ 대화형 차트, 그래프 및 대시보드를 생성하기 위한 다양한 도구 제공
- ✓ 2013년, 캐나다에 회사 설립
- ✓ Dash Enterprise 제공, 다양한 언어에서 활용 가능

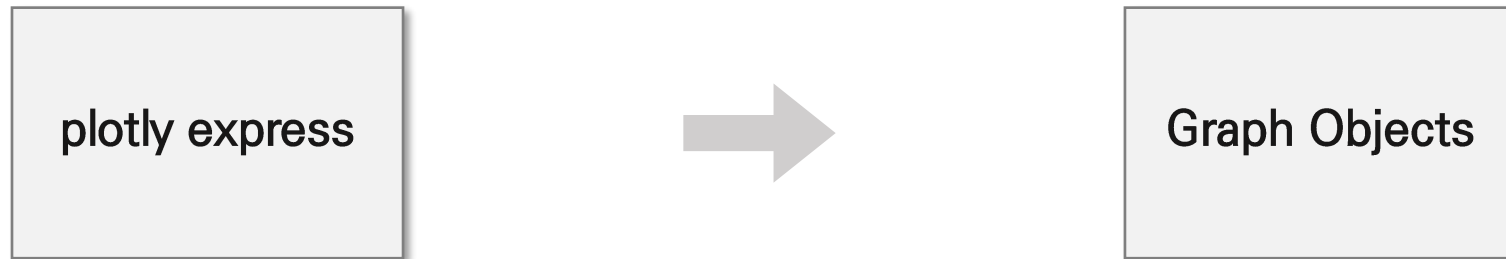
 Plotly Python Open Source Graphing Library  Star 13,238	 Plotly R Open Source Graphing Library  Star 2,358	 Plotly Julia Open Source Graphing Library  Star 369	 Plotly Javascript Open Source Graphing Library  Star 15,571
 Plotly ggplot2 Open Source Graphing Library  Star 2,358	 Plotly F# Open Source Graphing Library  Star 428	 Plotly MATLAB® Open Source Graphing Library  Star 334	 Plotly Dash Open Source Analytical App Framework  Star 18,468

출처 : <https://plotly.com/graphing-libraries/>

# 10. plotly

## ◆ Graph Objects vs Plotly Express

Graph Objects	Plotly Express
Low Level Interface	High Level Interface
<ul style="list-style-type: none"><li>▪ 세부적인 커스터마이징이 가능함</li><li>▪ 복잡한 상호작용과 레이아웃 만들기 가능</li><li>▪ 배울 것이 많음</li><li>▪ 고급 사용자</li></ul>	<ul style="list-style-type: none"><li>• 간단하고 직관적인 문법으로 쉽게 시각화 생성 가능</li><li>• 다양한 표준 차트</li><li>• 세밀한 커스터마이징은 제한됨</li><li>• 입문자</li></ul>



출처 : <https://plotly.com/graphing-libraries/>

# 10. plotly

## ◆ plotly 코드 기본 예제

```
•[6]: import plotly.graph_objects as go
import numpy as np

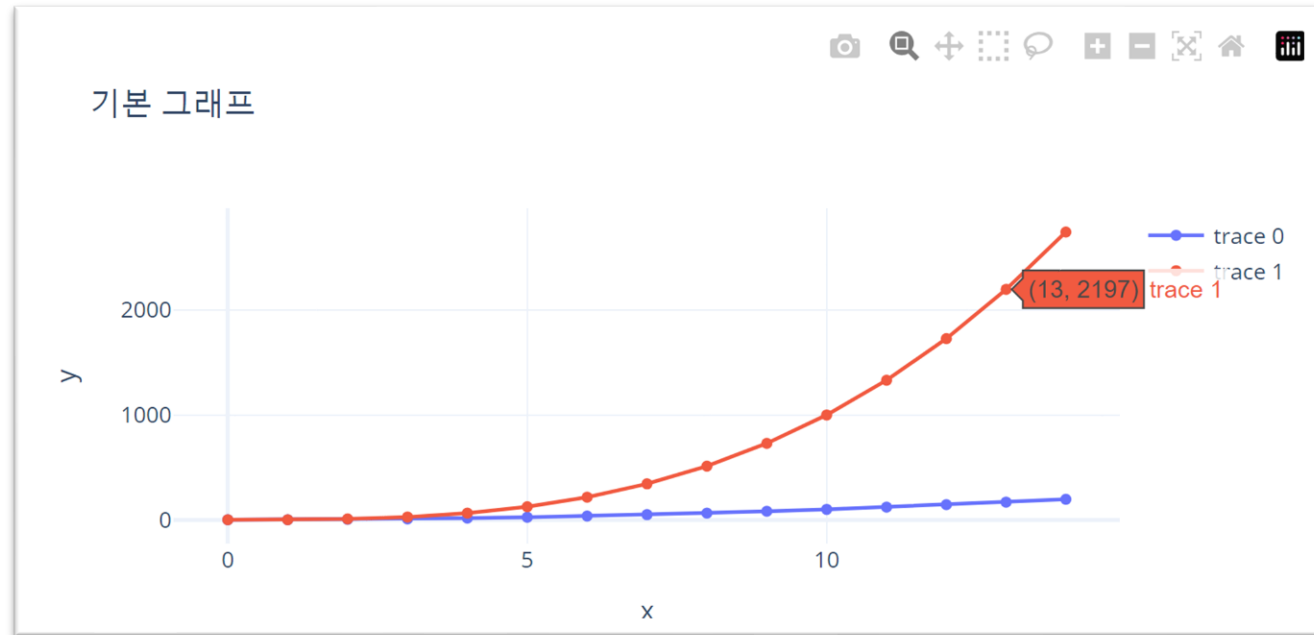
# 데이터 불러오기
x = np.arange(0,15,1)
y1, y2 = x**2, x**3

# figure 생성
fig = go.Figure()

# 시각화 코드
for y in [y1, y2]:
    fig.add_trace(go.Scatter(x=x, y=y))

# 시각화 코드 추가 옵션
fig.update_layout(title='기본 그래프',
                    xaxis_title='x',
                    yaxis_title='y',
                    template='plotly_white')

# 시각화 보여주기
fig.show()
```



# 10. plotly

## ◆ plotly 코드 기본 예제

```
•[6]: import plotly.graph_objects as go
import numpy as np

# 데이터 불러오기
x = np.arange(0,15,1)
y1, y2 = x**2, x**3

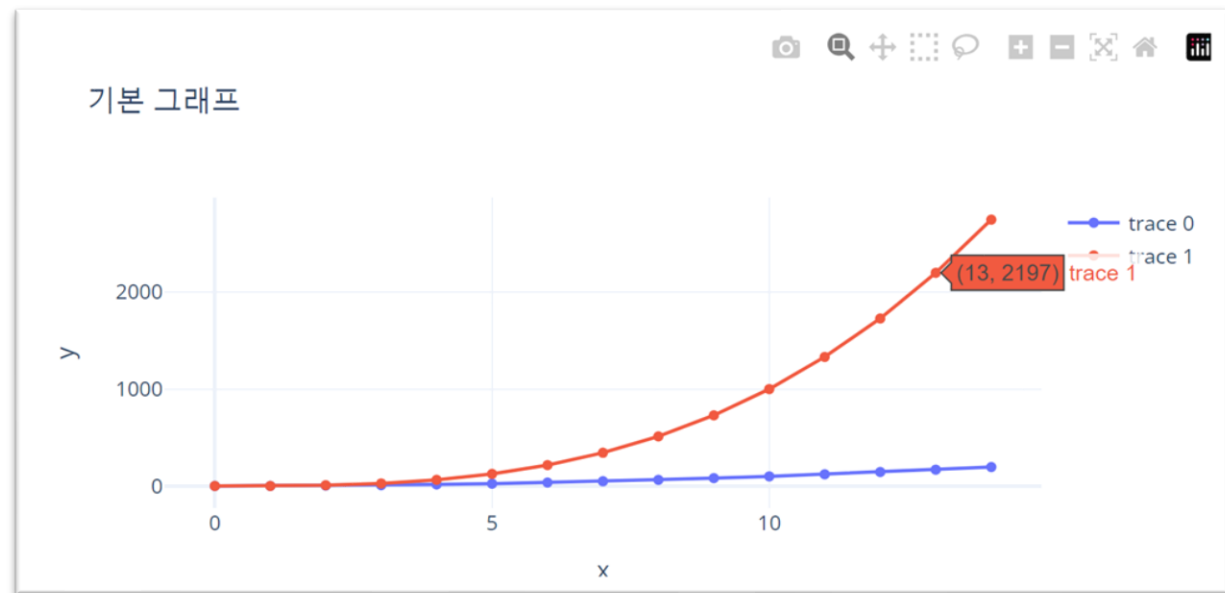
# figure 생성
fig = go.Figure()

# 시각화 코드
for y in [y1, y2]:
    fig.add_trace(go.Scatter(x=x, y=y))

# 시각화 코드 추가 옵션
fig.update_layout(title='기본 그래프',
                    xaxis_title='x',
                    yaxis_title='y',
                    template='plotly_white')

# 시각화 보여주기
fig.show()
```

코드 순서	설명
데이터 불러오기	가상의 데이터를 불러온다.
객체 생성	Figure Object 생성
Add traces	시각화 기본 차트 생성 메서드
Update Layout	레이아웃과 스타일을 수정하는 데 사용



# 10. plotly

## ◆ plotly 주요 클래스

구분	설명
subplots	<ul style="list-style-type: none"><li>• 하나의 Figure에 다양한 유형의 Plot을 결합할 수 있음</li><li>• 유연한 Grid 레이아웃을 제공하여 여러 행 또는 열에 걸쳐 Plot을 만들 수 있음</li><li>• 서로 다른 데이터 집합 또는 동일한 데이터의 서로 다르게 비교 시 유용</li></ul>
Figure Factory <i>권고안해도</i>	<ul style="list-style-type: none"><li>• Plotly Express에서 없는 특수한 Plot을 쉽게 표현 가능</li><li>• <del>But, 버전 업그레이드가 될 때마다 plotly.express에 통합되는 중, deprecated</del></li><li>• Figure Factory 클래스에 맞게 데이터 별도 정의 필요</li></ul>
io	<ul style="list-style-type: none"><li>• Plotly 시각화 객체 input/output과 관련되어 있는 클래스</li><li>• JSON, HTML, static images를 불러오고 내보낼 수 있음</li></ul>



# 10. plotly

## ◆ plotly graph update

### Chart 생성

- `plotly.express`
- `graph_objects` & `add_trace()`



### Chart 업데이트

- `update_traces()`
- `update_layout()`

구분	설명
<code>update_traces</code>	<ul style="list-style-type: none"><li>• 이미 생성된 각 chart의 속성을 수정하는 데 사용</li><li>• trace의 type, 색, 스타일, 템플릿 등 추가 편집 가능</li><li>• Mouse Hover 정보 설정 및 trace의 동작 수정 가능 (예: line → marker)</li></ul>
<code>update_layout</code>	<ul style="list-style-type: none"><li>• 전체 figure의 레이아웃을 수정</li><li>• title, legend, axes labels, annotations 수정</li><li>• Background color, font styles</li></ul>

## Chapter 03. plotly

**강의 실습 영상 참고**