Visualization

Data handling section

SAMSUNG
Data Science Academy

최성철 교수
Director of TEAMLAB

SAMSUNG

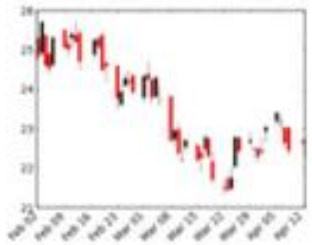
matplotlib

우리의 데이터는 어떻게 생겼을까?

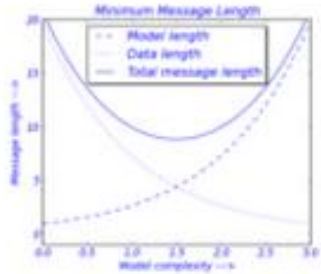
Visualization
데이터 시각화

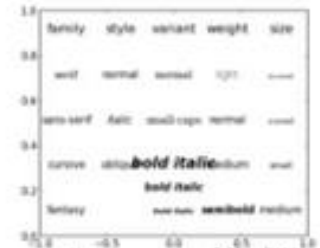
파이썬의 대표적인 시각화 도구

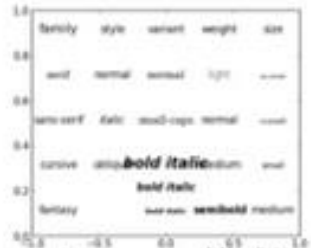

matplotlib

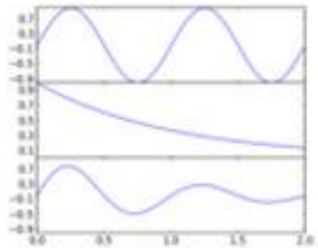


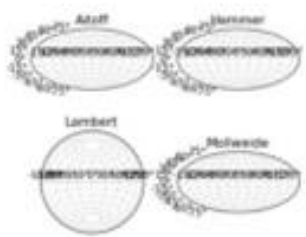

SPY daily

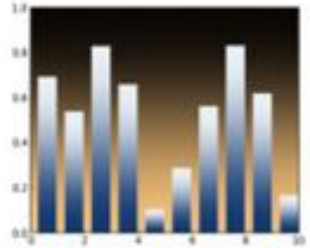

Minimum Message Length
Model length
Data length
Total message length
Message length ⟶ n
Model complexity ⟶ n

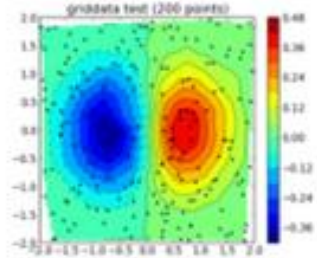

family
style
variant
weight
size
normal
bold
bold italic
italic
semibold
monospace
large

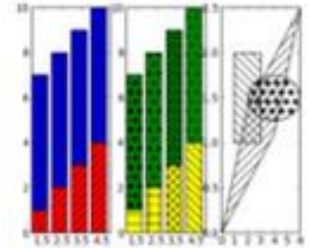

family
style
variant
weight
size
normal
bold
bold italic
italic
semibold
monospace
large

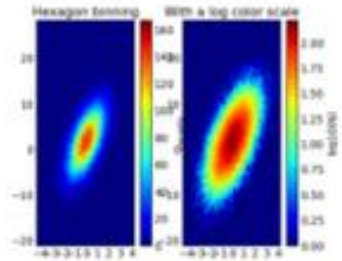


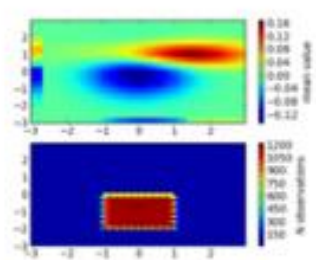

Aitoff
Hammer
Lambert
Mollweide

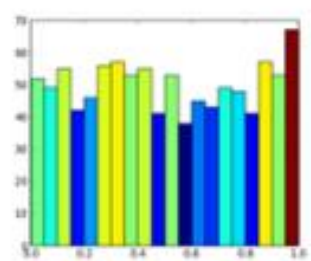


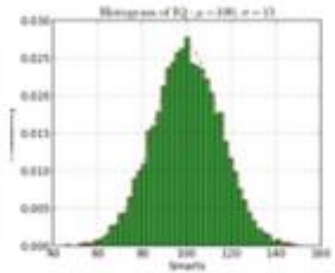

griddata test (200 points)

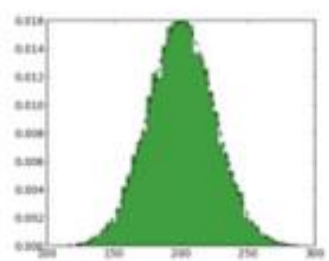


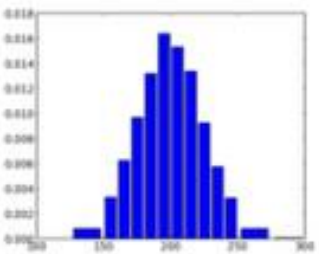

Hexagon binning
With a log color scale

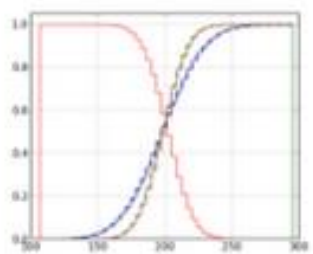

mean value
N observations

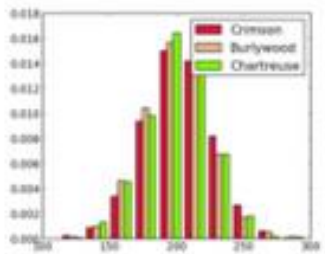


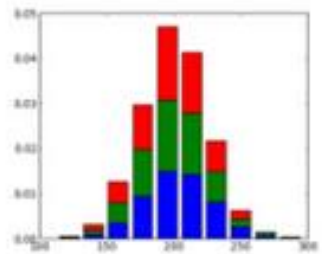


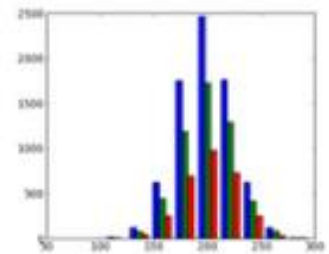







Crimson
Burlywood
Chartreuse





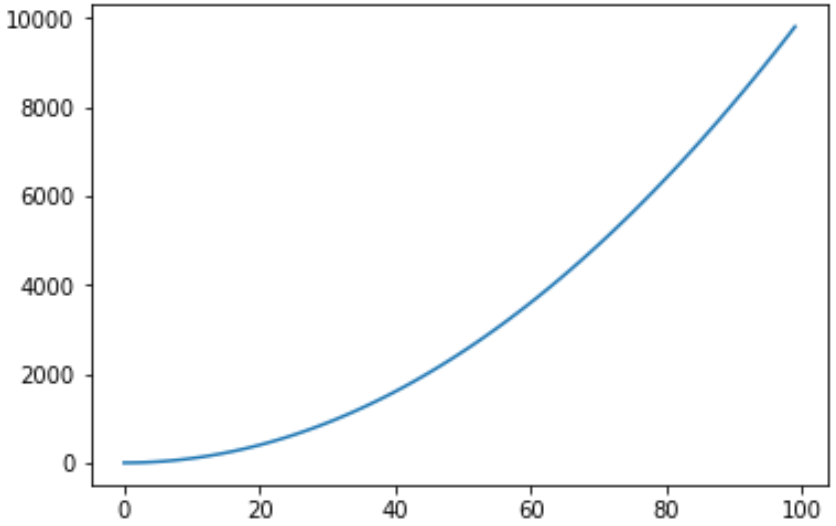
다양한 graph 지원
Pandas 연동!

기본 사용법

matplotlib

- pyplot 객체를 사용하여 데이터를 표시
- pyplot 객체에 그래프들을 쌓은 다음 flush

```
import matplotlib.pyplot as plt

X = range(100)
Y = [value**2 for value in X]
plt.plot(X, Y)
plt.show()
```



matplotlib

- 최대 단점 argument를 kwargs 받음,
- 고정된 argument가 없어서
  alt+tab으로 확인이 어려움

```
Signature: plt.plot(*args, **kwargs)
Docstring:
Plot lines and/or markers to the
:class:`~matplotlib.axes.Axes`.  *args* is a variable length
argument, allowing for multiple *x*, *y* pairs with an
optional format string.  For example, each of the following is
legal::

    plot(x, y)        # plot x and y using default line style and color
    plot(x, y, 'bo')  # plot x and y using blue circle markers
    plot(y)           # plot y using x as index array 0..N-1
    plot(y, 'r+')     # ditto, but with red plusses
```

matplotlib

- Graph는 원래 figure 객체에 생성됨
- pyplot 객체 사용시, 기본 figure에 그래프가 그려짐

```
X_1 = range(100)
Y_1 = [np.cos(value) for value in X]

X_2 = range(100)
Y_2 = [np.sin(value) for value in X]

plt.plot(X_1, Y_1)
plt.plot(X_2, Y_2)
plt.show()
```

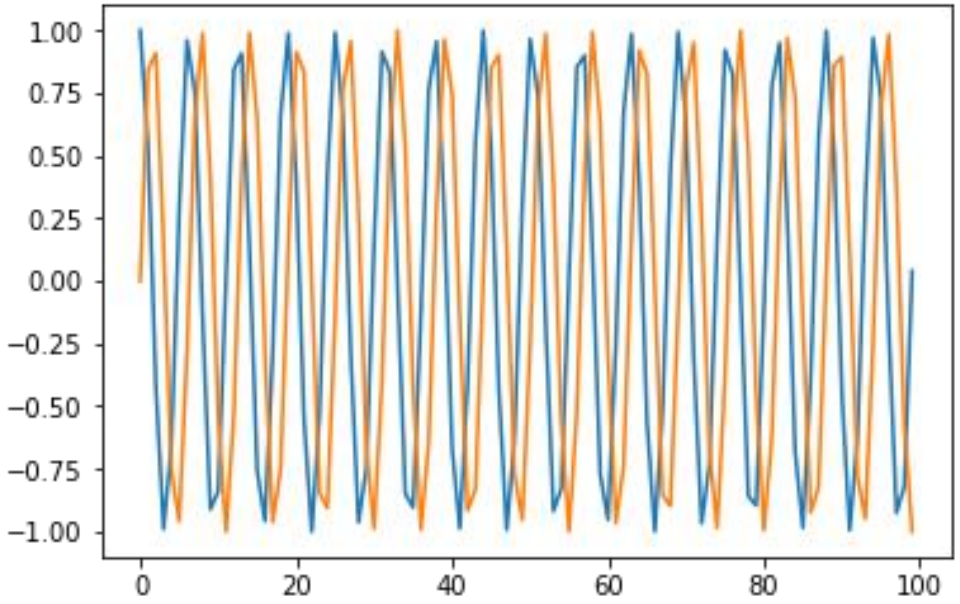
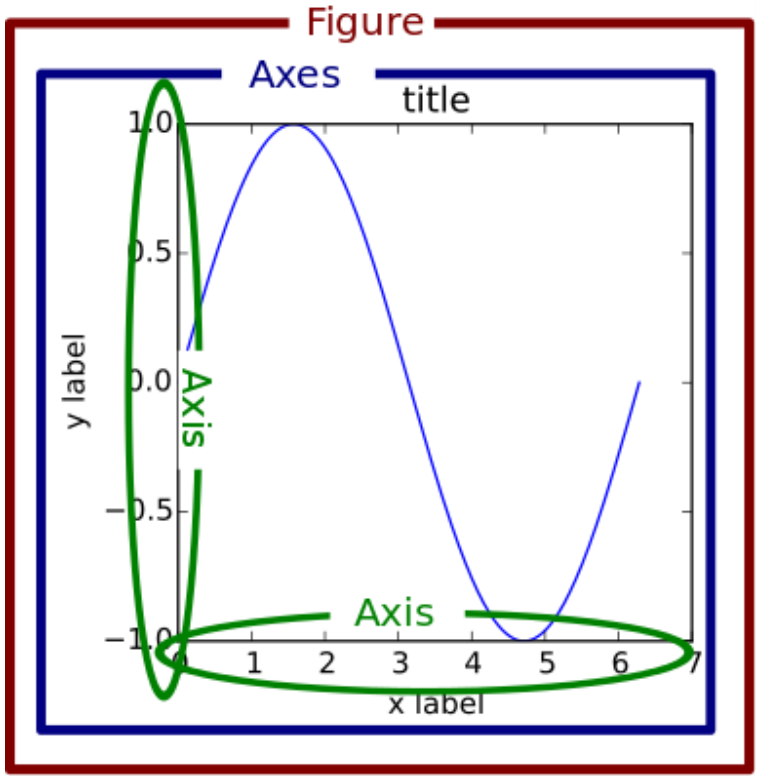


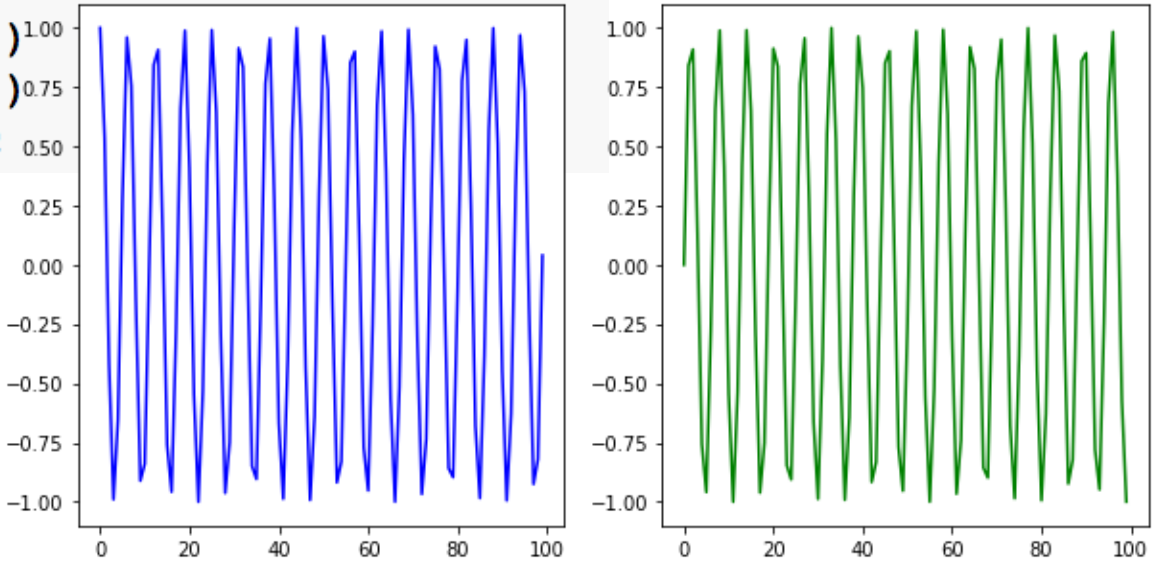
Figure & Axes

- Matplotlib는 Figure 안에 Axes로 구성
- Figure 위에 여러 개의 Axes를 생성



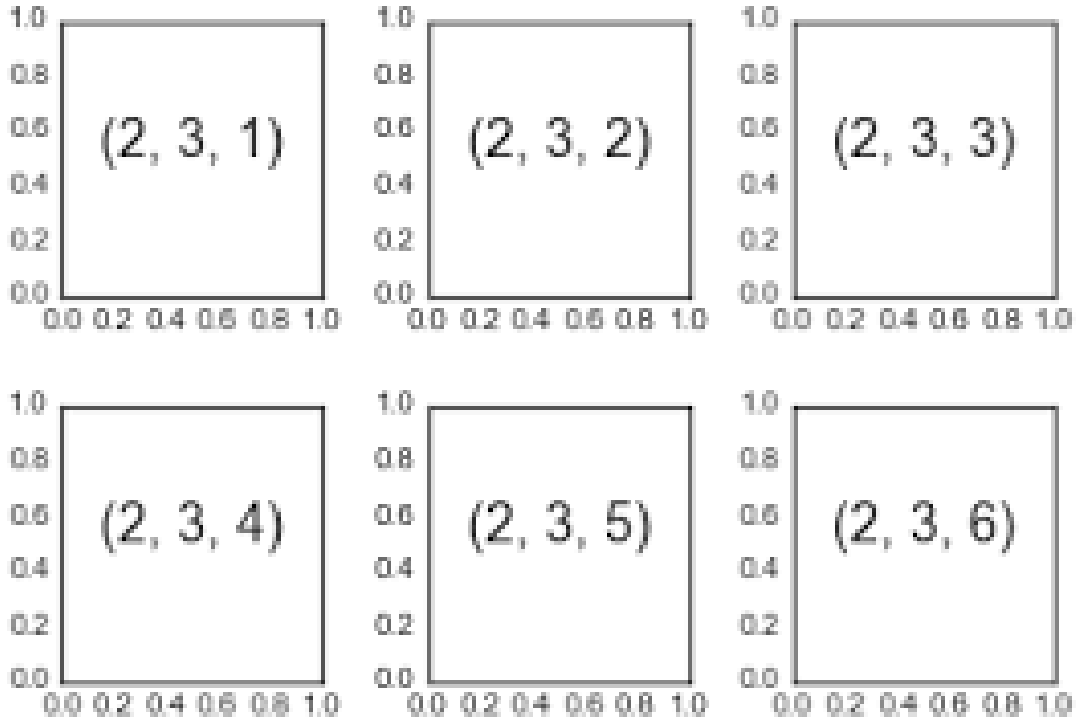
matplotlib

```
fig = plt.figure() # figure 반환
fig.set_size_inches(10,5) # 크기지정
ax_1 = fig.add_subplot(1,2,1) # 두개의 plot 생성
ax_2 = fig.add_subplot(1,2,2)  # 두개의 plot 생성

ax_1.plot(X_1, Y_1, c="b")
ax_2.plot(X_2, Y_2, c="g")
plt.show() # show & flush
```



subplots

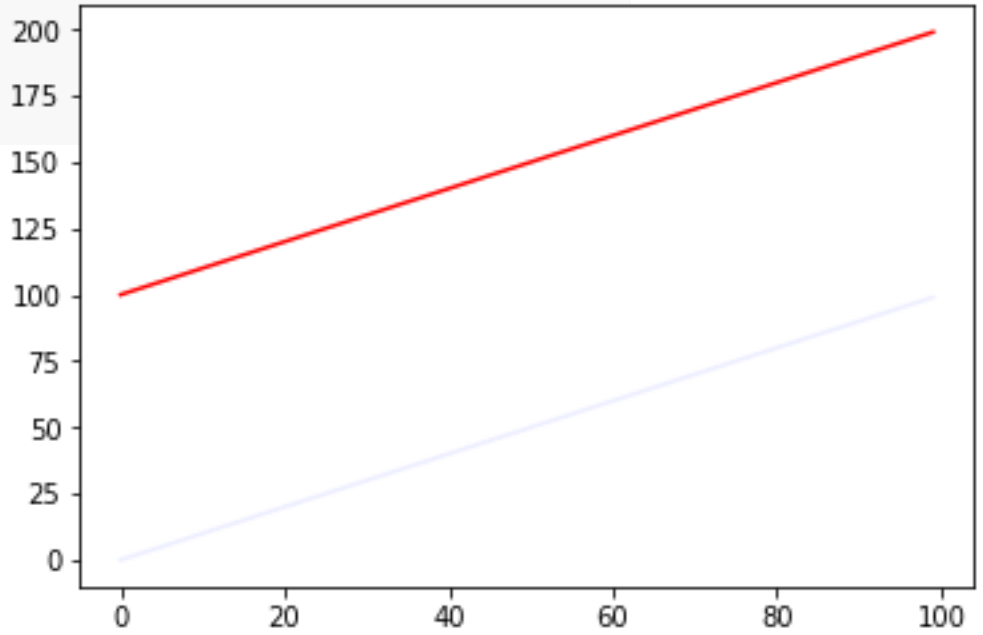
- Subplot의 순서를 grid로 작성



Set color

- Color 속성을 사용
- Float → 흑백, rgb color, predefined color 사용

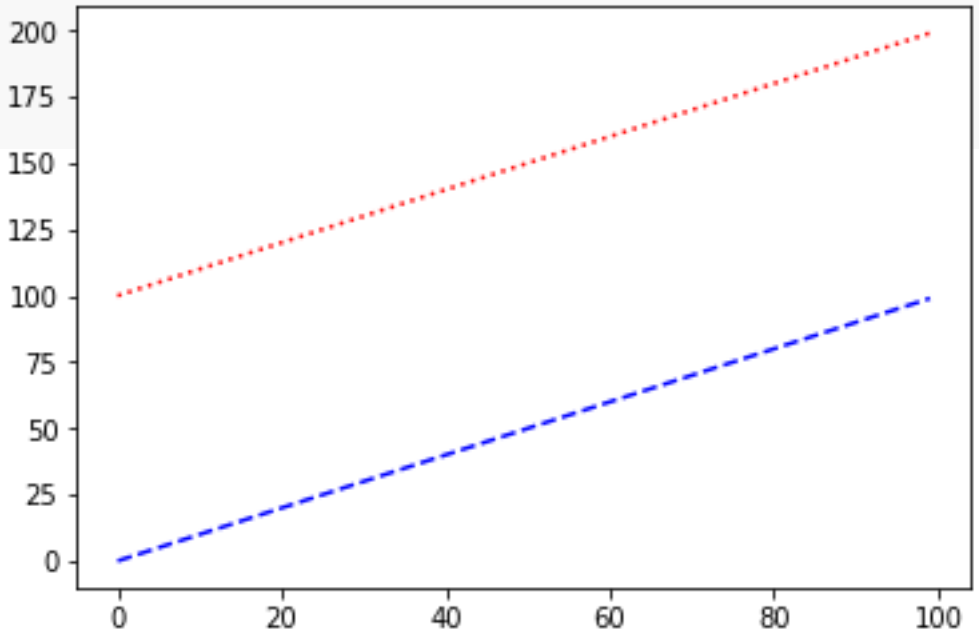
```
plt.plot(X_1, Y_1, color="#eeefff")
plt.plot(X_2, Y_2, color="r")

plt.show()
```



Set linstyle

- ls 또는 linestyle 속성 사용

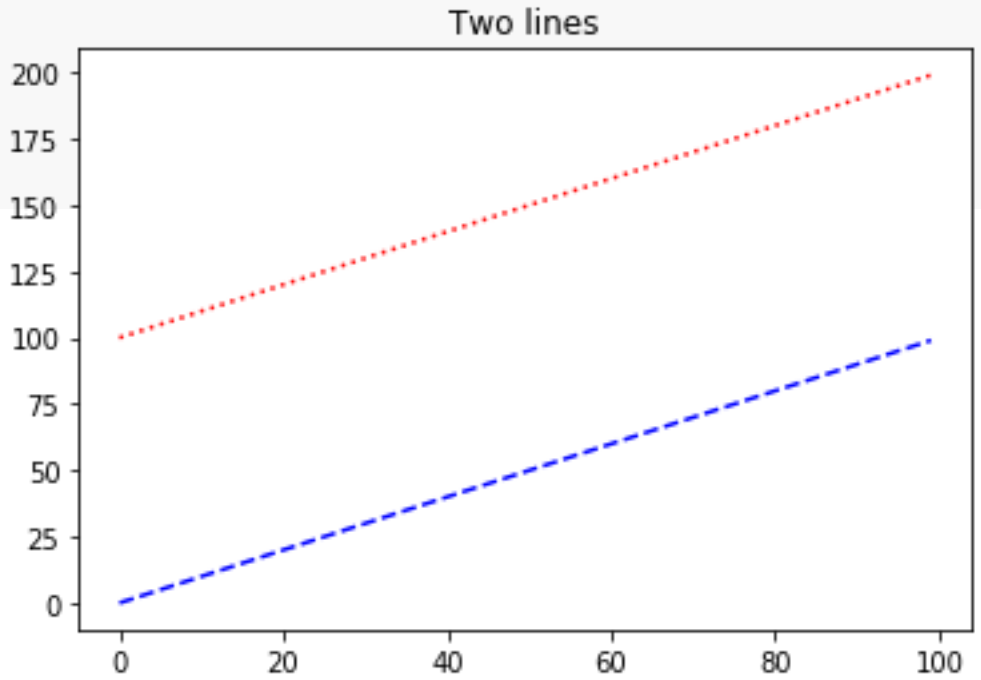
```
plt.plot(X_1, Y_1, c="b", linestyle="dashed")
plt.plot(X_2, Y_2, c="r", ls="dotted")

plt.show()
```



Set title

- Pyplot에 title함수 사용, figure의 subplot별 입력 가능

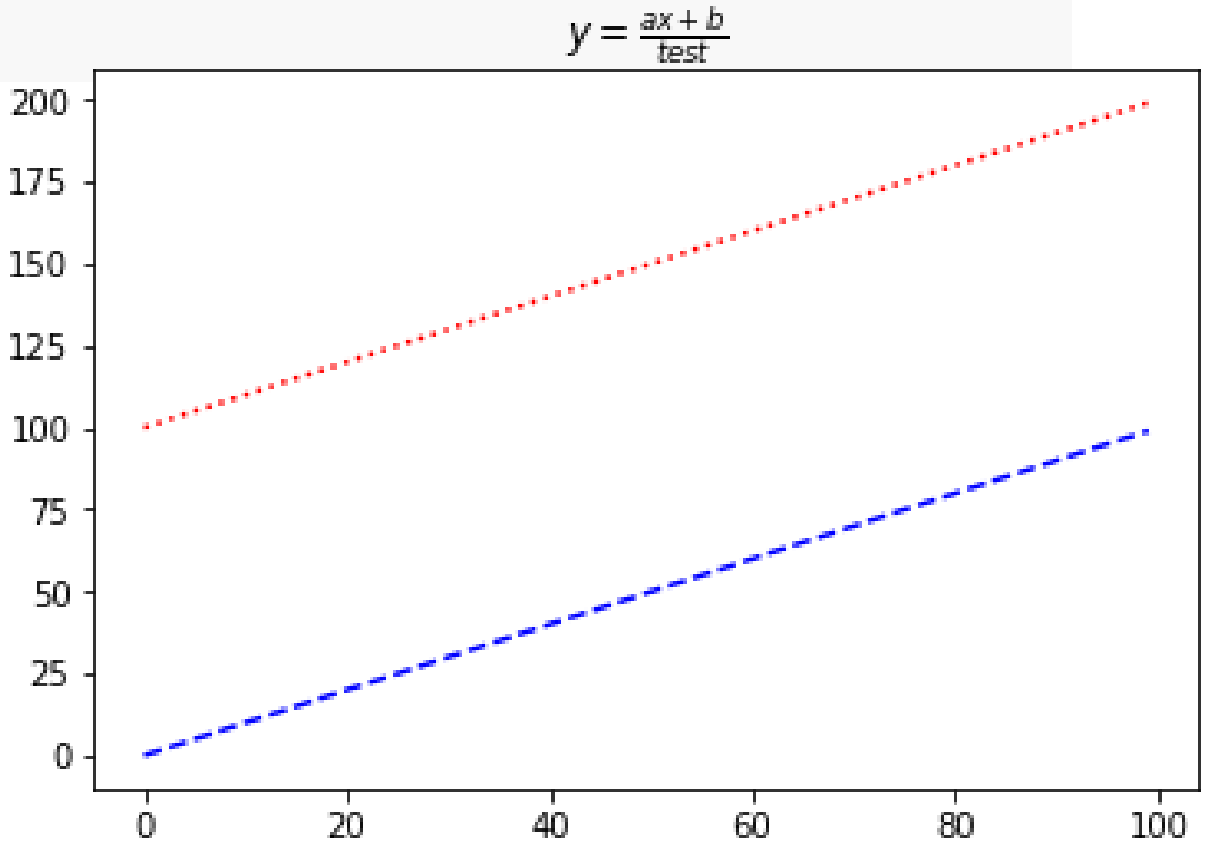
```
plt.plot(X_1, Y_1, color="b", linestyle="dashed")
plt.plot(X_2, Y_2, color="r", linestyle="dotted")

plt.title("Two lines")
plt.show()
```



Set title

- Latex 타입의 표현도 가능 (수식 표현 가능)

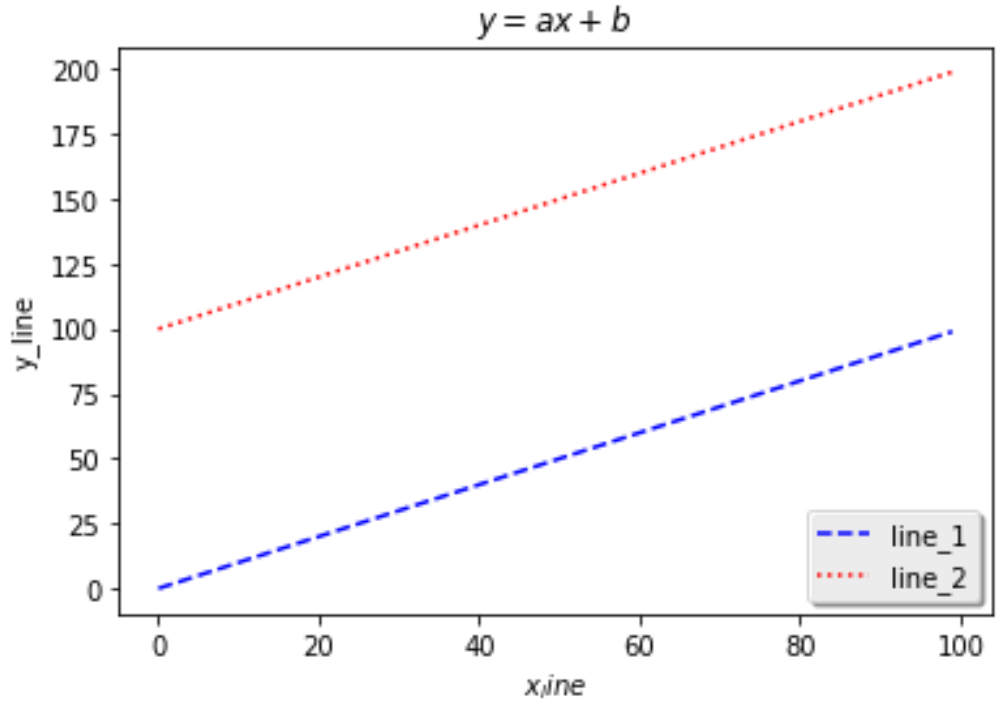
```
plt.title('$y = \\frac{ax + b}{test}$')
plt.show()
```



Set legend

- Legend 함수로 범례를 표시함, loc 위치등 속성 지정

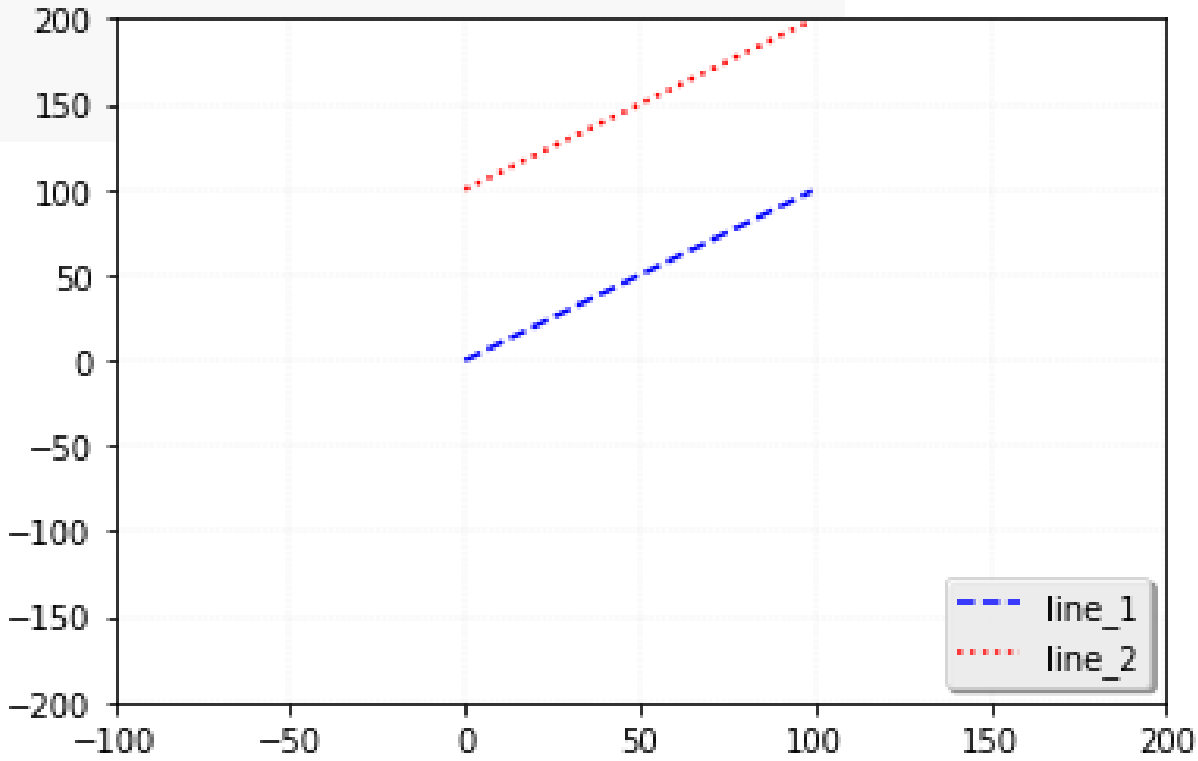
```
plt.plot(X_1, Y_1, color="b", linestyle="dashed", label='line_1')
plt.plot(X_2, Y_2, color="r", linestyle="dotted", label='line_2')
plt.legend(shadow=True, fancybox=True, loc="lower right")
```



Set grid & xylim

- Graph 보조선을 긋는 grid와 xy축 범위 한계를 지정

```
plt.grid(True, lw=0.4, ls="--", c=".90")
plt.xlim(-100, 200)
plt.ylim(-200, 200)
```

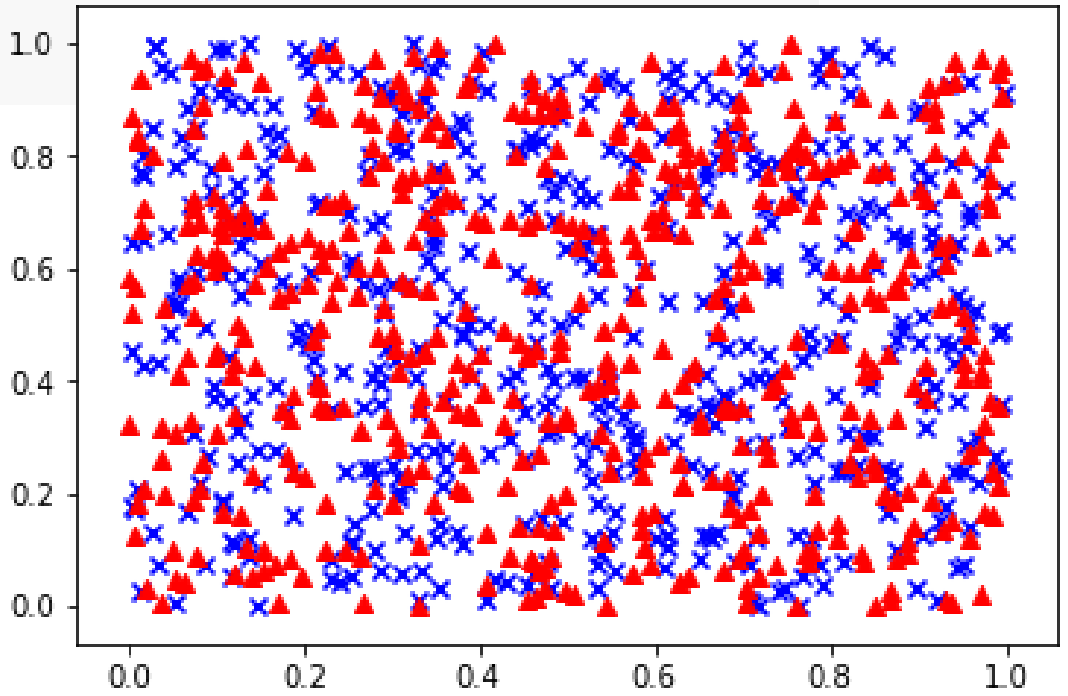


Matplotlib Graph

Scatter

- scatter 함수 사용, marker: scatter 모양지정

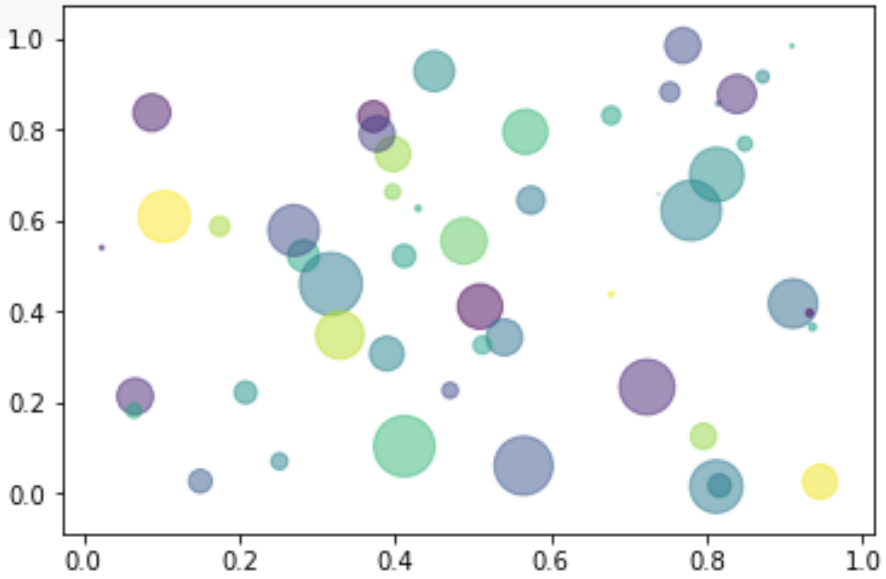
```
data_1 = np.random.rand(512, 2)
data_2 = np.random.rand(512, 2)

plt.scatter(data_1[:,0], data_1[:,1], c="b", marker="x")
plt.scatter(data_2[:,0], data_2[:,1], c="r", marker="^")

plt.show()
```



Scatter

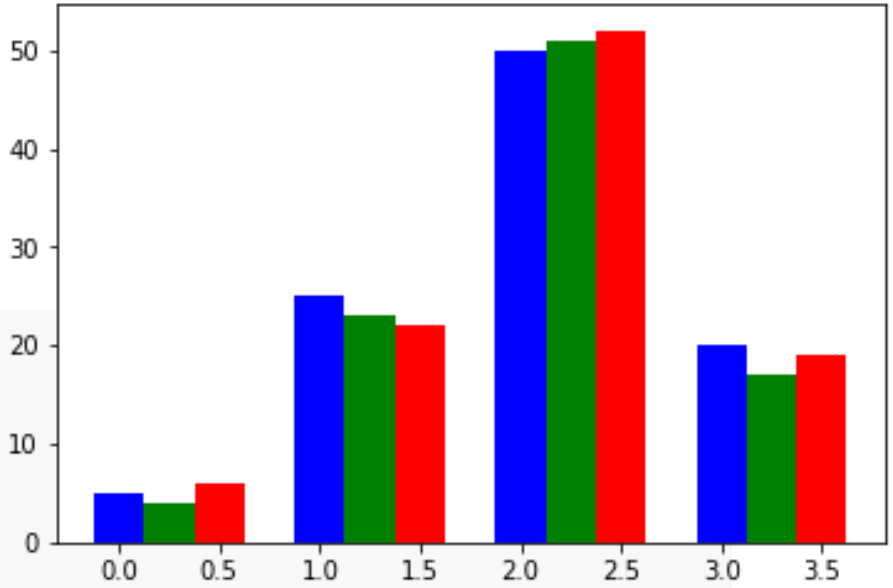
- s: 데이터의 크기를 지정, 데이터의 크기비교가능

```
N = 50
x = np.random.rand(N)
y = np.random.rand(N)
colors = np.random.rand(N)
area = np.pi * (15 * np.random.rand(N))**2
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.show()
```



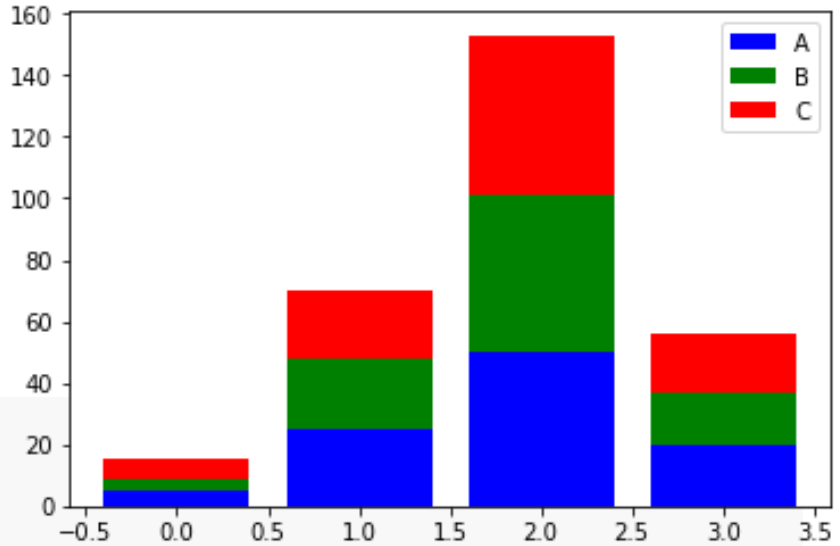
Bar chart

- Bar 함수 사용




```
data = [[5., 25., 50., 20.],
        [4., 23., 51., 17],
        [6., 22., 52., 19]]

X = np.arange(4)
plt.bar(X + 0.00, data[0], color = 'b', width = 0.25)
plt.bar(X + 0.25, data[1], color = 'g', width = 0.25)
plt.bar(X + 0.50, data[2], color = 'r', width = 0.25)
plt.xticks(X+0.25, ("A","B","C", "D"))
plt.show()
```

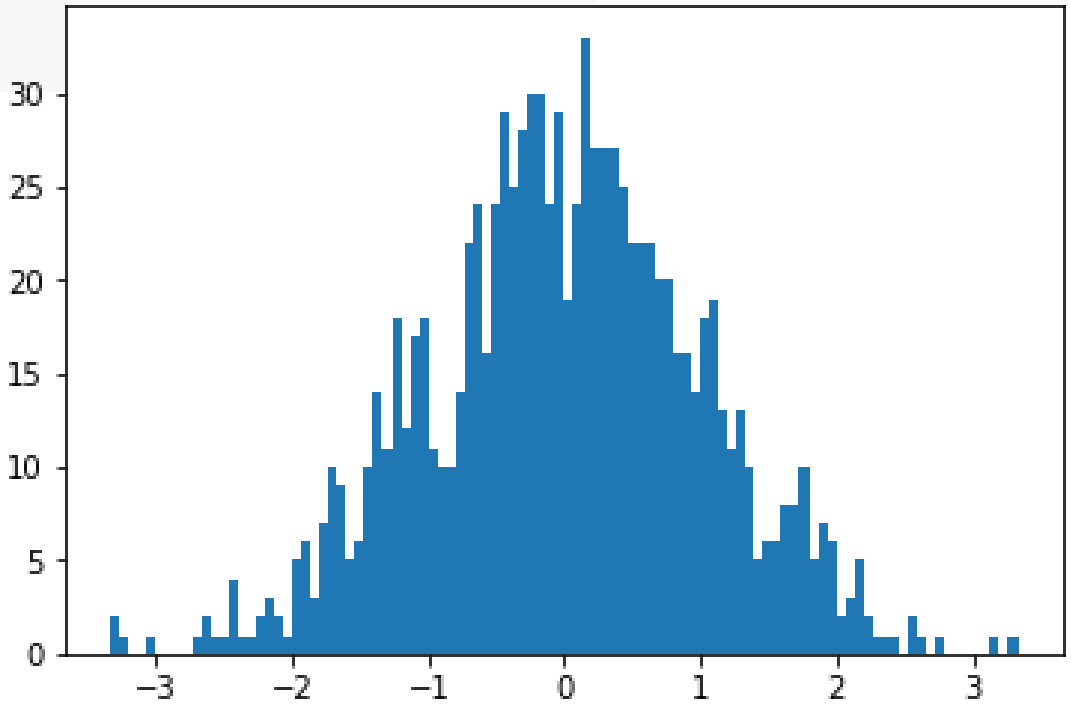
Bar chart




```
color_list = ['b', 'g', 'r']
data_label = ["A","B","C"]
X = np.arange(data.shape[1])
for i in range(data.shape[0]):
    plt.bar(X, data[i], bottom = np.sum(data[:i], axis=0),
            color = color_list[i], label=data_label[i])
plt.legend()
plt.show()
```

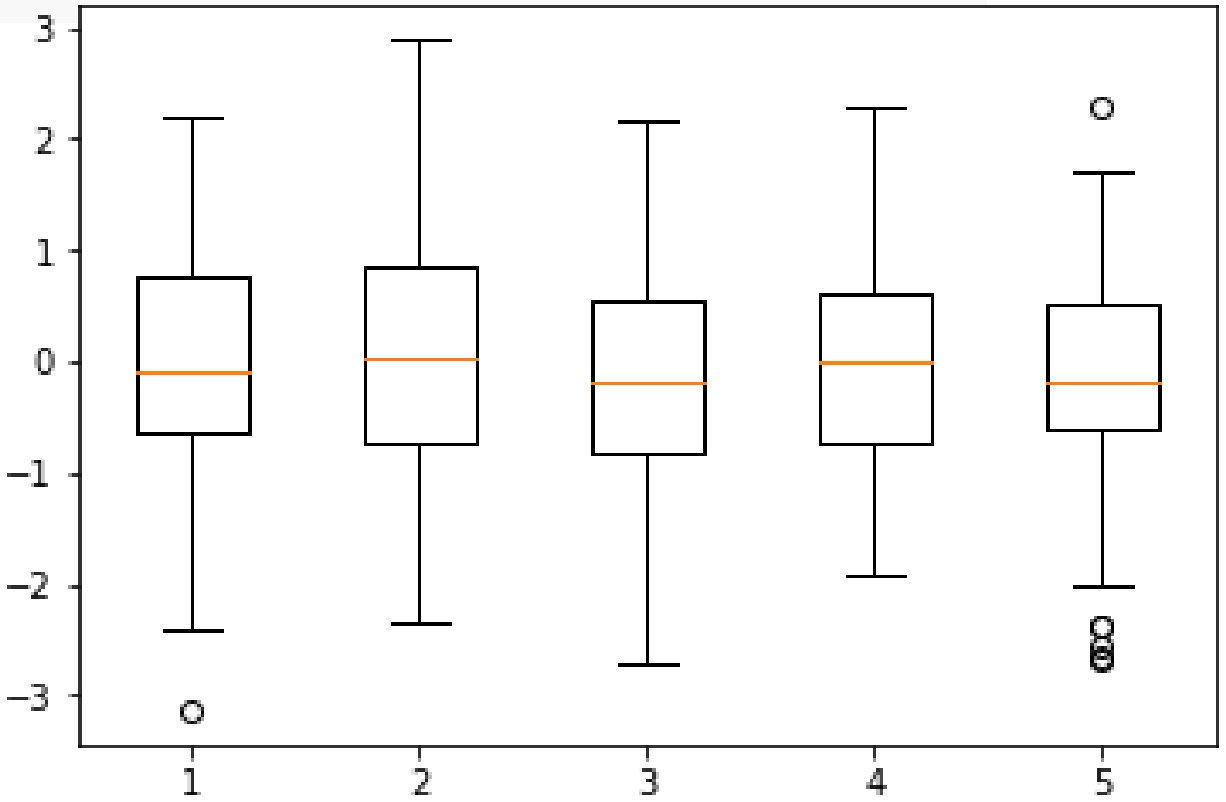
histogram

```
X = np.random.randn(1000)
plt.hist(X,bins=100)
plt.show()
```



boxplot

```
data = np.random.randn(100,5)
plt.boxplot(data)
plt.show()
```



matplotlib with pandas

pandas matplotlib

- matplotlib를 사용한 그래프 지원
- Dataframe, series별로 그래프 작성 가능

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2

Boston House Price Dataset

X 변수
13개

Y 변수

[01] CRIM	자치시(town) 별 1인당 범죄율
[02] ZN	25,000 평방피트를 초과하는 거주지역의 비율
[03] INDUS	비소매상업지역이 점유하고 있는 토지의 비율
[04] CHAS	찰스강에 대한 더미변수(강의 경계에 위치한 경우는 1, 아니면 0)
[05] NOX	10ppm 당 농축 일산화질소
[06] RM	주택 1가구당 평균 방의 개수
[07] AGE	1940년 이전에 건축된 소유주택의 비율
[08] DIS	5개의 보스턴 직업센터까지의 접근성 지수
[09] RAD	방사형 도로까지의 접근성 지수
[10] TAX	10,000 달러 당 재산세율
[11] PTRATIO	자치시(town)별 학생/교사 비율
[12] B	1000(Bk-0.63)^2, 여기서 Bk는 자치시별 흑인의 비율을 말함.
[13] LSTAT	모집단의 하위계층의 비율(%)
[14] MEDV	본인 소유의 주택가격(중앙값) (단위: $1,000)

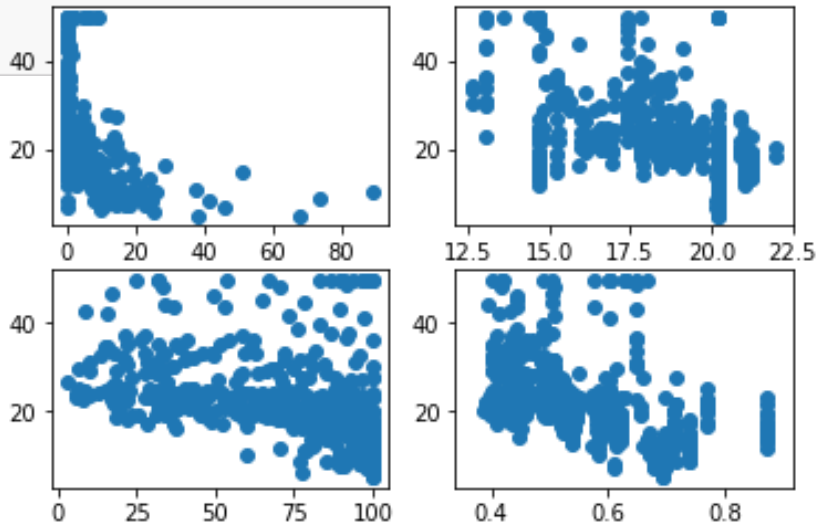
http://www.dator.co.kr/ctg258/textyle/1721307

http://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html

pandas matplotlib

- 데이터간의 상관관계를 볼 때 scatter graph 사용 가능

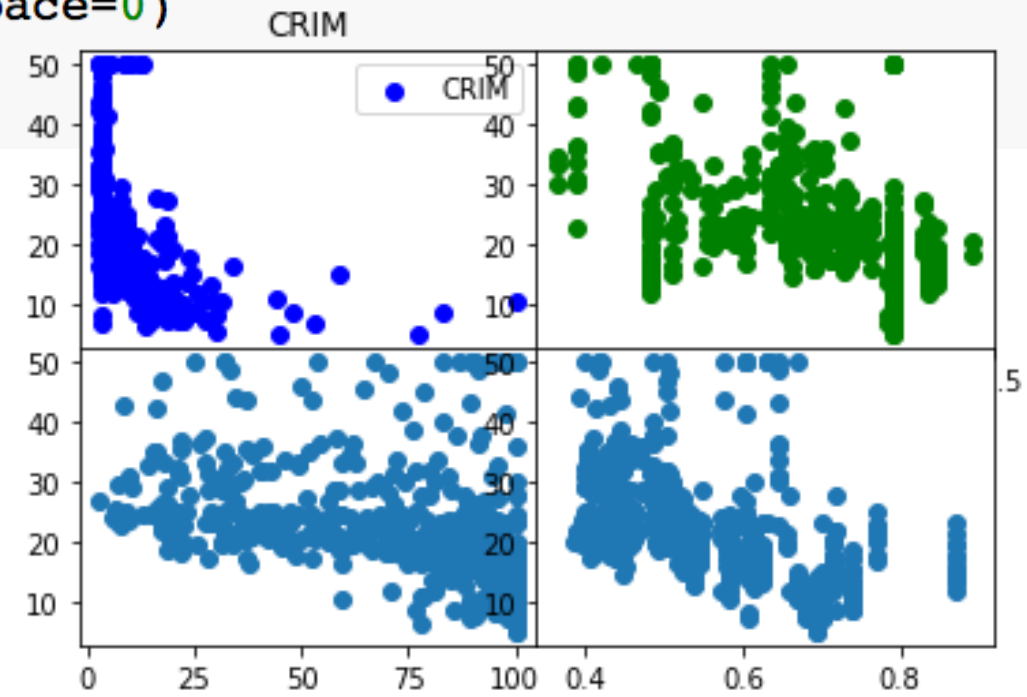
```
fig = plt.figure()
ax = []
for i in range(1,5):
    ax.append(fig.add_subplot(2,2,i))
ax[0].scatter(df_data["CRIM"], df_data["MEDV"])
ax[1].scatter(df_data["PTRATIO"], df_data["MEDV"])
ax[2].scatter(df_data["AGE"], df_data["MEDV"])
ax[3].scatter(df_data["NOX"], df_data["MEDV"])
plt.show()
```



pandas matplotlib

- matplotlib의 꾸미기 함수 그대로 사용함

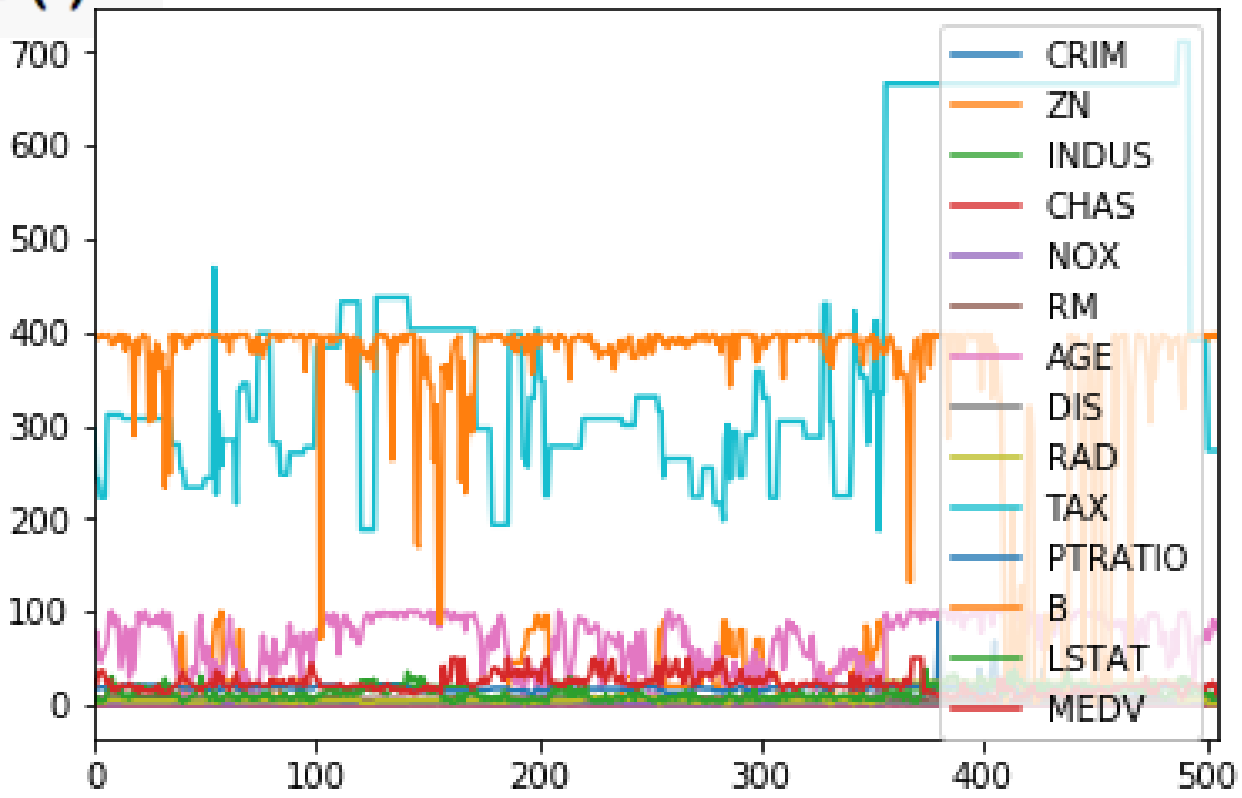
```
ax[0].scatter(df_data["CRIM"], df_data["MEDV"], color="b", label="CRIM")
ax[1].scatter(df_data["PTRATIO"], df_data["MEDV"], color="g"  )
ax[2].scatter(df_data["AGE"], df_data["MEDV"] )
ax[3].scatter(df_data["NOX"], df_data["MEDV"])
plt.subplots_adjust(wspace=0, hspace=0)
ax[0].legend()
ax[0].set_title("CRIM")
```



pandas matplotlib - plot function

- plot 함수를 사용하면 전체 데이터의 graph를 그림

```
df_data.plot()
```



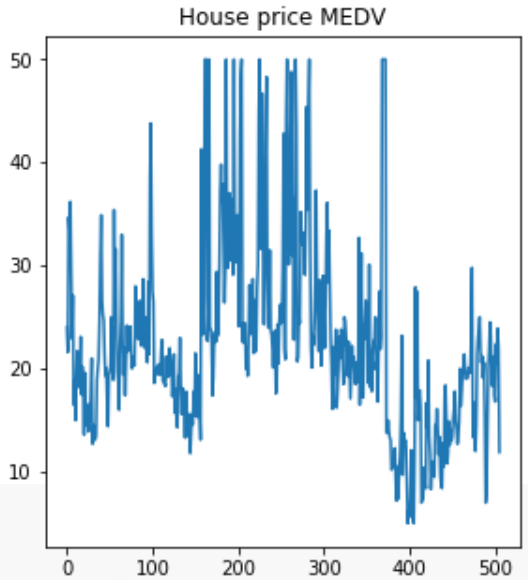
pandas matplotlib - plot function

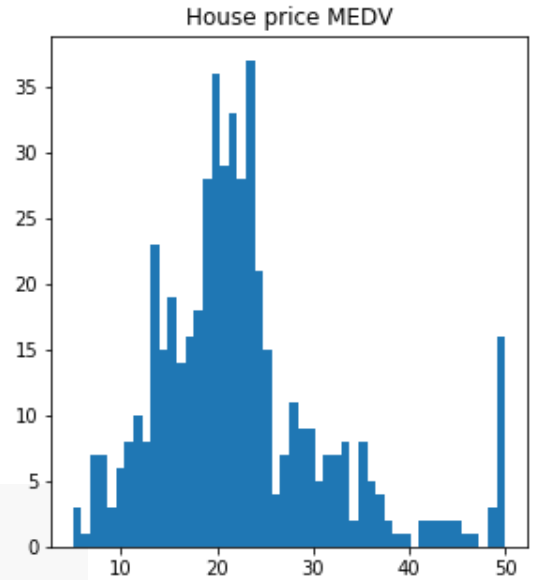
- kind : plot 종류
- x : x축 컬럼
- y : y축 컬럼

```
import matplotlib.pyplot as plt
import pandas as pd

df.plot(kind='scatter',x='num_children',y='num_pets',color='red')
plt.show()
```

http://queirozf.com/entries/pandas-dataframe-plot-examples-with-matplotlib-pyplot

pandas matplotlib

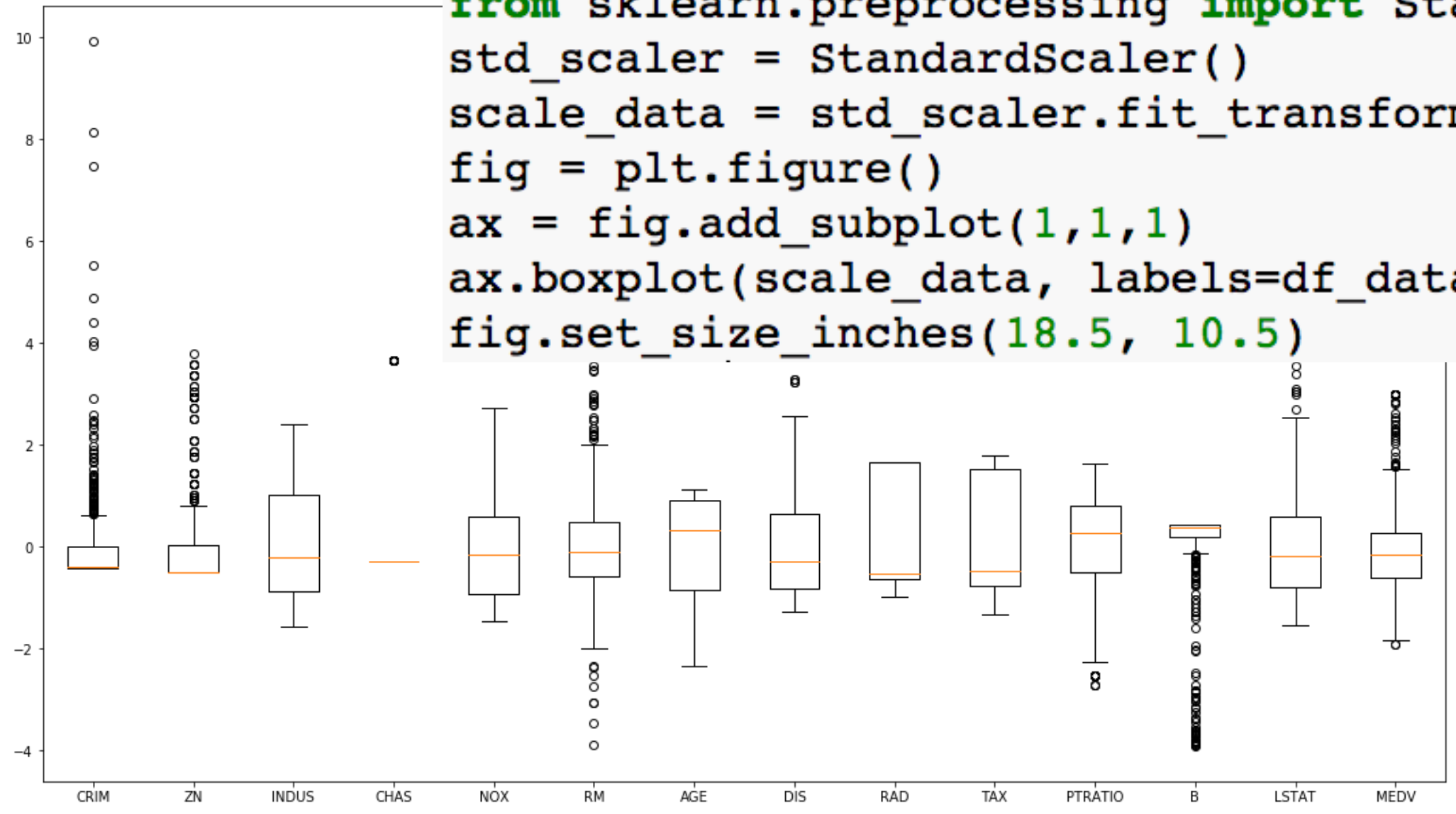





```
fig = plt.figure()
fig.set_size_inches(10,5)
ax_1 = fig.add_subplot(1,2,1)
ax_2 = fig.add_subplot(1,2,2)
ax_1.plot(df_data["MEDV"])
ax_2.hist(df_data["MEDV"], bins=50)
ax_1.set_title("House price MEDV")
ax_2.set_title("House price MEDV")
```

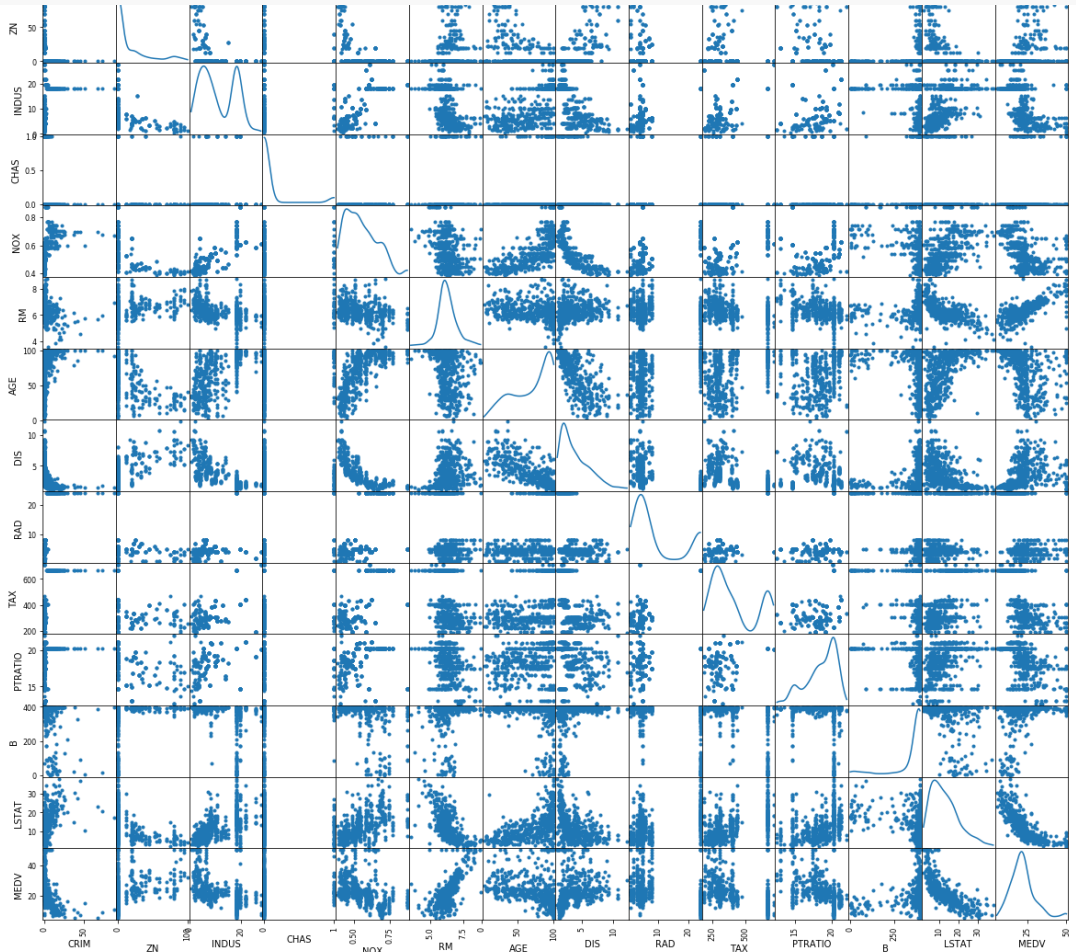
Scaled boxplot

```
from sklearn.preprocessing import StandardScaler
std_scaler = StandardScaler()
scale_data = std_scaler.fit_transform(df_data)
fig = plt.figure()
ax = fig.add_subplot(1,1,1)
ax.boxplot(scale_data, labels=df_data.columns)
fig.set_size_inches(18.5, 10.5)
```



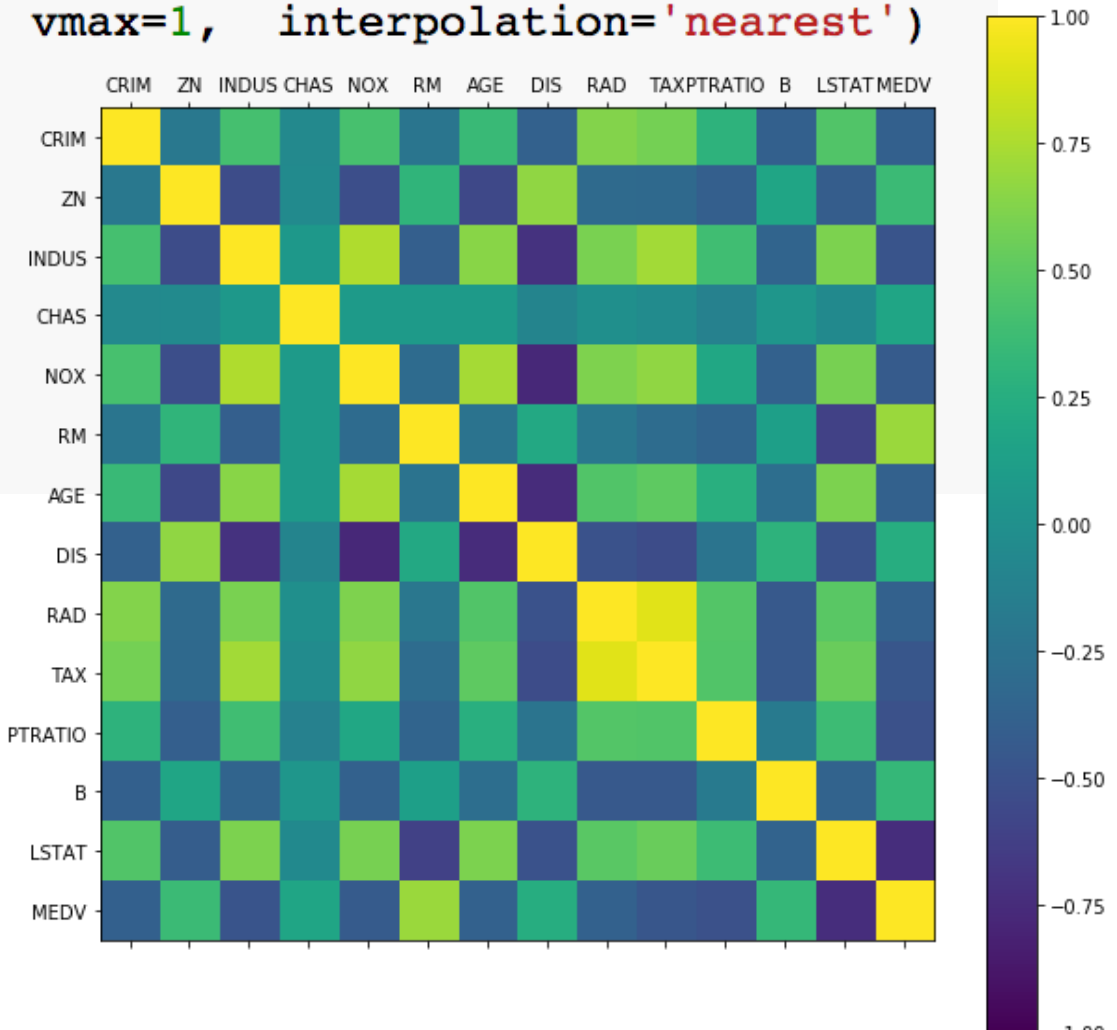
Scatter matrix

```
pd.scatter_matrix(df_data, diagonal="kde", alpha=1,  figsize=(20, 20))
plt.show()
```



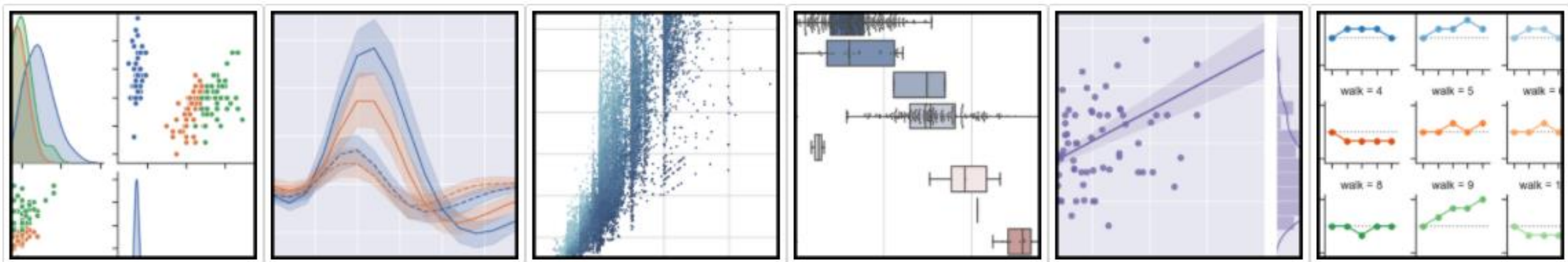
Scatter matrix

```
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(corr_data, vmin=-1, vmax=1,  interpolation='nearest')
fig.colorbar(cax)
fig.set_size_inches(10,10)
ticks= np.arange(0,14,1)
ax.set_xticks(ticks)
ax.set_yticks(ticks)

ax.set_xticklabels(df_data.columns)
ax.set_yticklabels(df_data.columns)
```



seaborn

seaborn: statistical data visualization



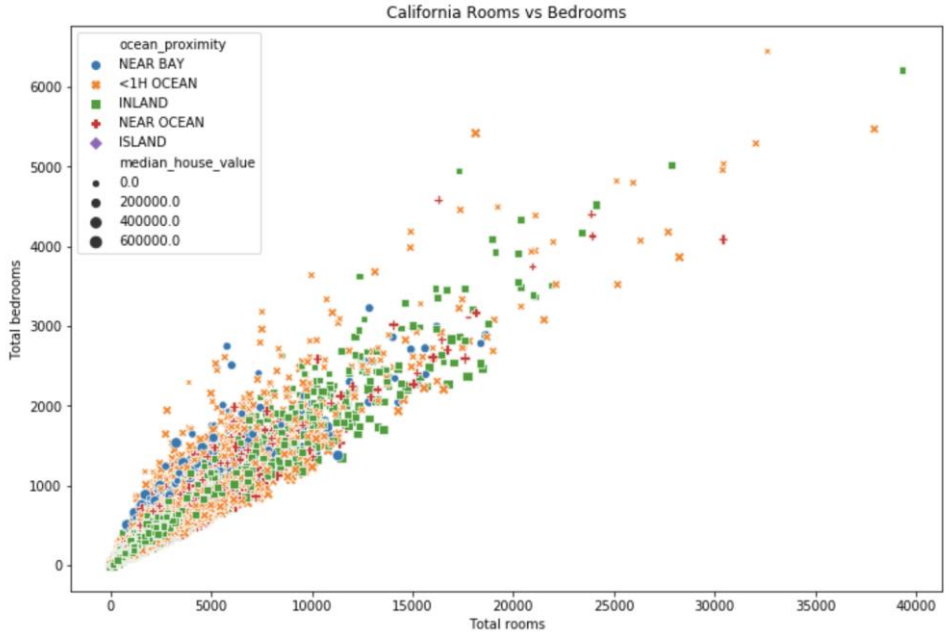
Matplotlib를 더 쉽게!

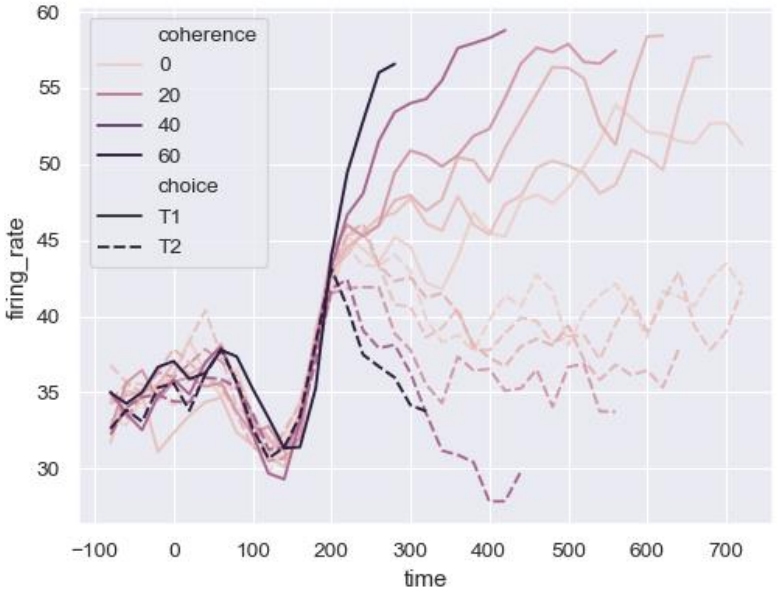
seaborn

- 기존 matplotlib에 기본 설정을 추가
- 복잡한 그래프를 간단하게 만들 수 있는 wrapper
- 간단한 코드 + 예쁜 결과

seaborn

- 기존 matplotlib에 기본 설정을 추가
- 복잡한 그래프를 간단하게 만들 수 있는 wrapper
- 간단한 코드 + 예쁜 결과





seaborn - basic plots

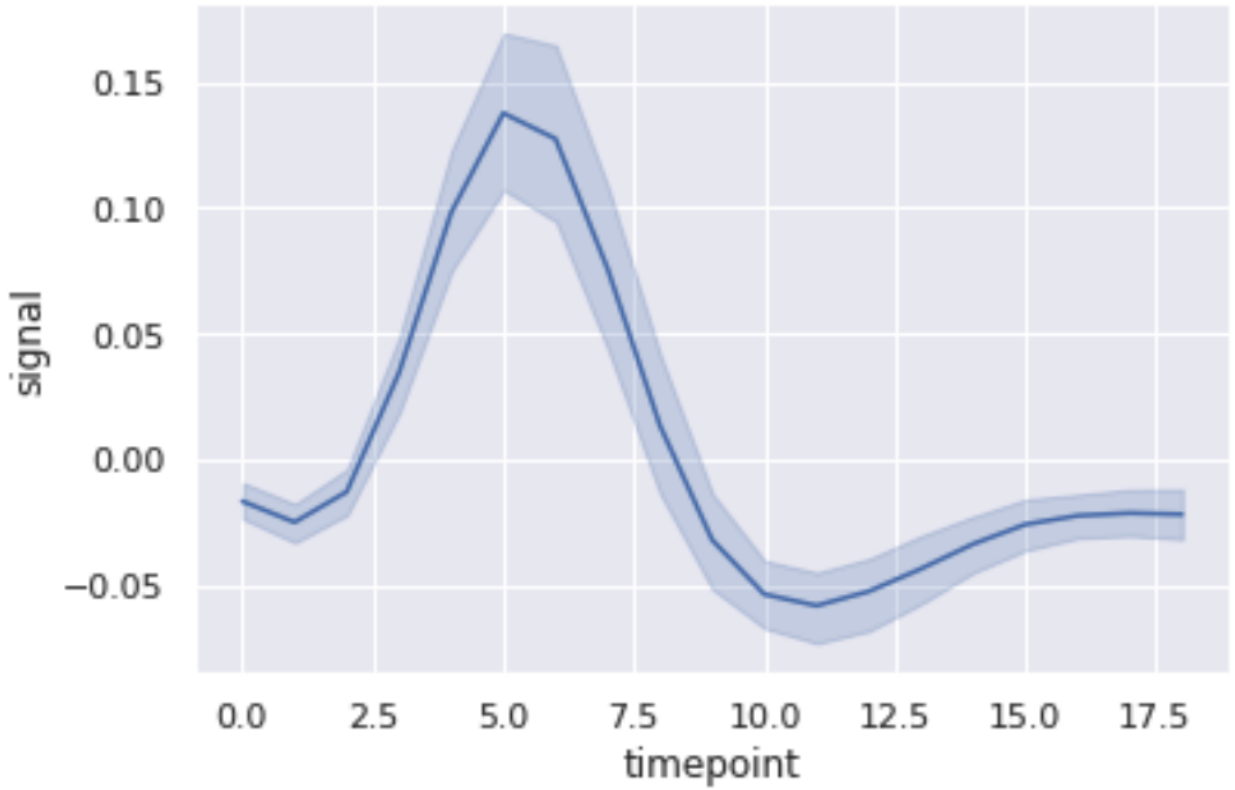
- matplotlib와 같은 기본적인 plot
- 손쉬운 설정으로 데이터 산출
- lineplot, scatterplot, countplot 등

```
sns.lineplot(x="total_bill", y="tip", data=tips)
```

seaborn - basic plots

```
1  # Plot the responses for different events and regions
2  sns.lineplot(x="timepoint", y="signal", data=fmri)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a23af3278>
```



seaborn - basic plots

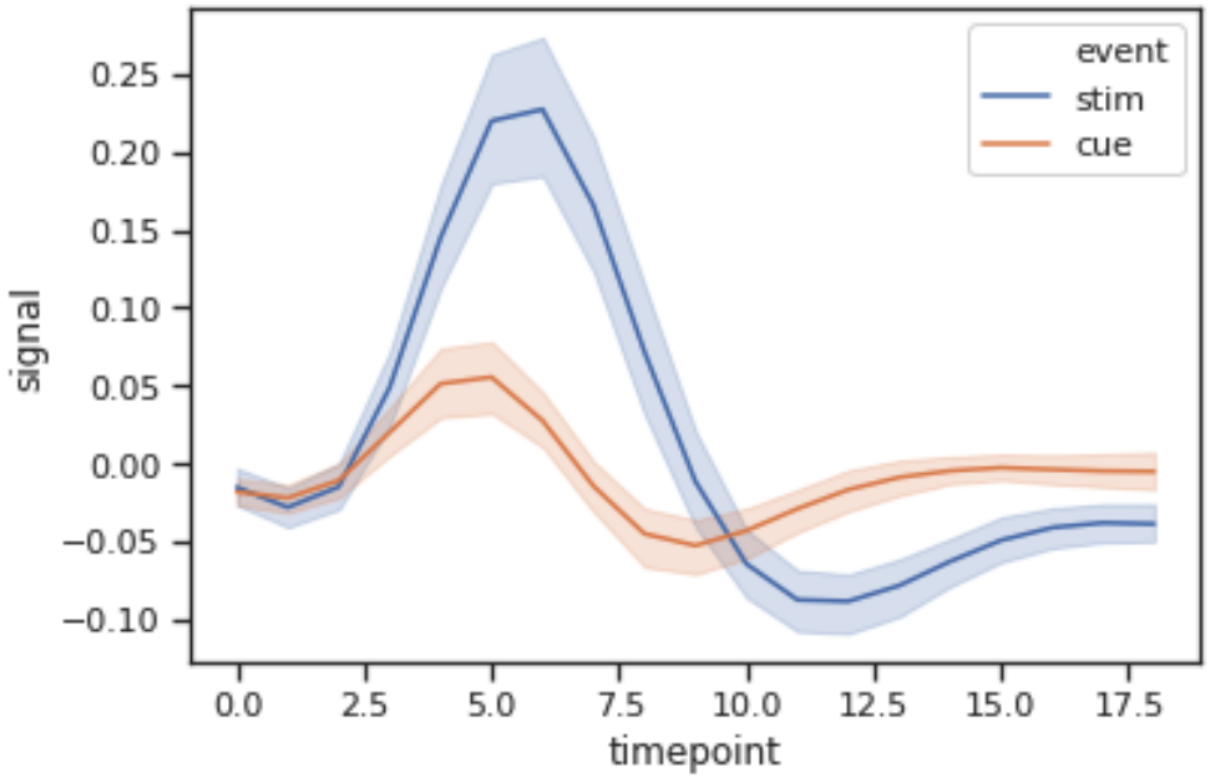
```
1 fmri.sample(n=10, random_state=1)
```

	subject	timepoint	event	region	signal
806	s6	18	cue	parietal	0.019532
691	s5	15	cue	frontal	-0.019507
148	s5	8	stim	parietal	0.006805
676	s13	0	cue	parietal	-0.018394
156	s11	7	stim	parietal	0.254042
27	s1	17	stim	parietal	-0.038021
200	s11	4	stim	parietal	0.087175
262	s3	0	stim	parietal	-0.008576
94	s4	12	stim	parietal	-0.090036
339	s4	5	stim	frontal	0.455575

seaborn - basic plots

```
1  sns.lineplot(x="timepoint", y="signal", hue="event", data=fmri)
```

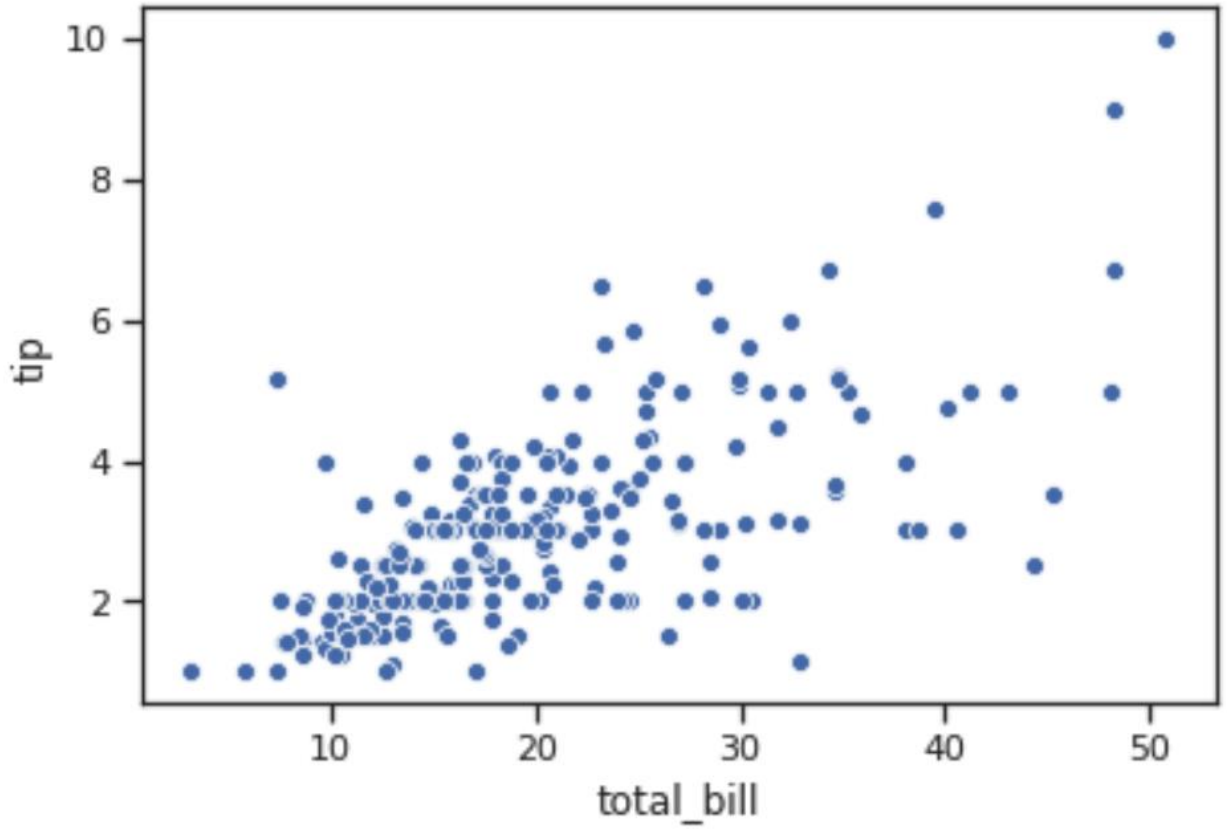
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a24a1bac8>
```



seaborn - basic plots

```
1  sns.scatterplot(x="total_bill", y="tip", data=tips)
```

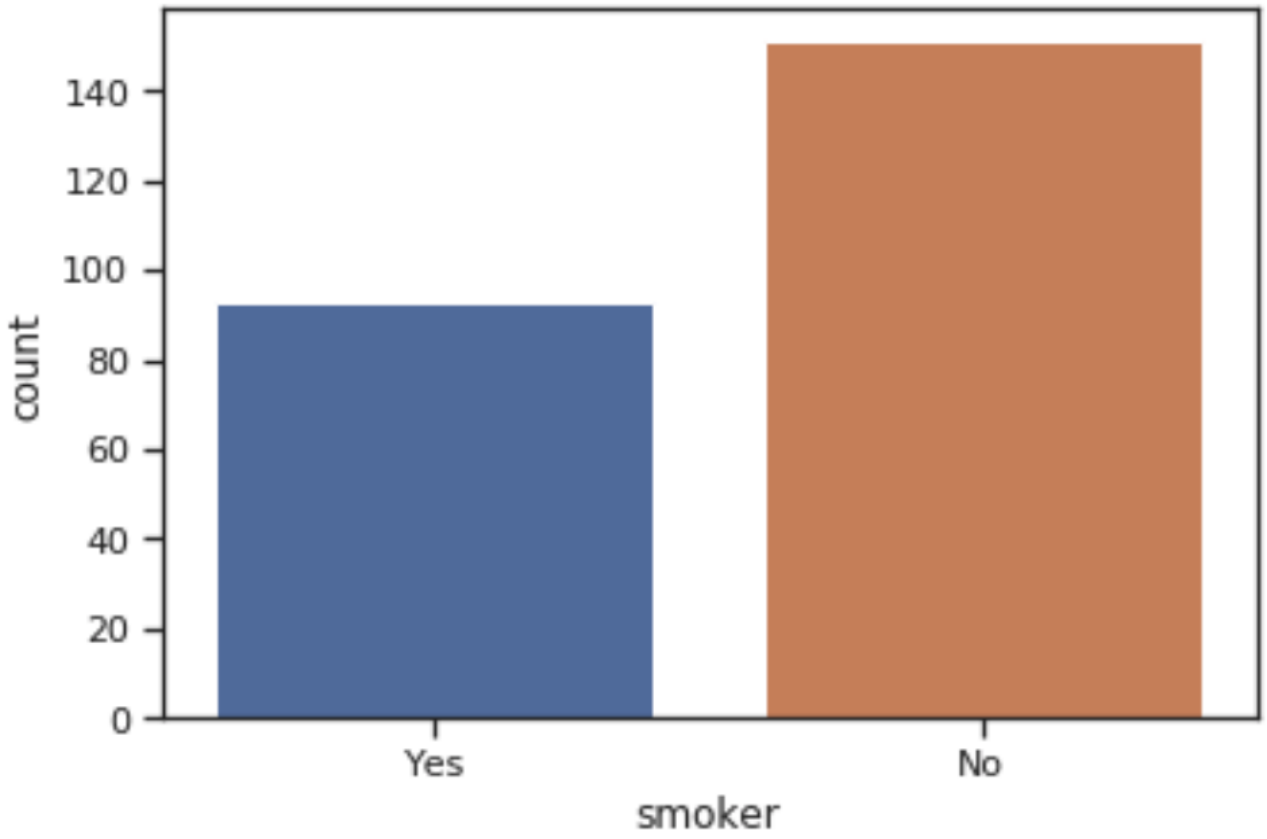
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a25d37e10>
```



seaborn - basic plots

```
1  sns.countplot(x="smoker",data=tips)
```

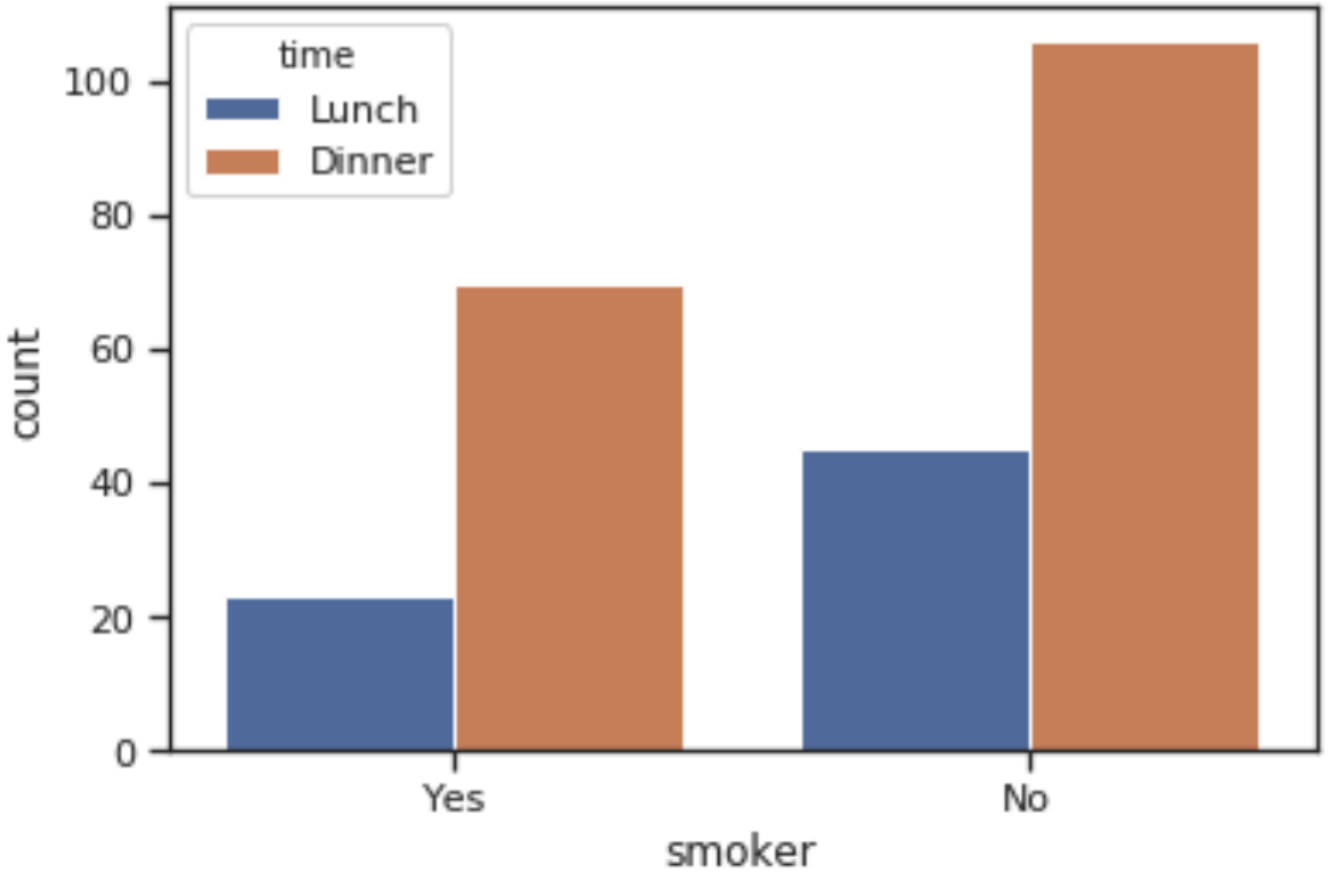
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a257d4eb8>
```



seaborn - basic plots

```
1  sns.countplot(x="smoker", hue="time", data=tips)
```

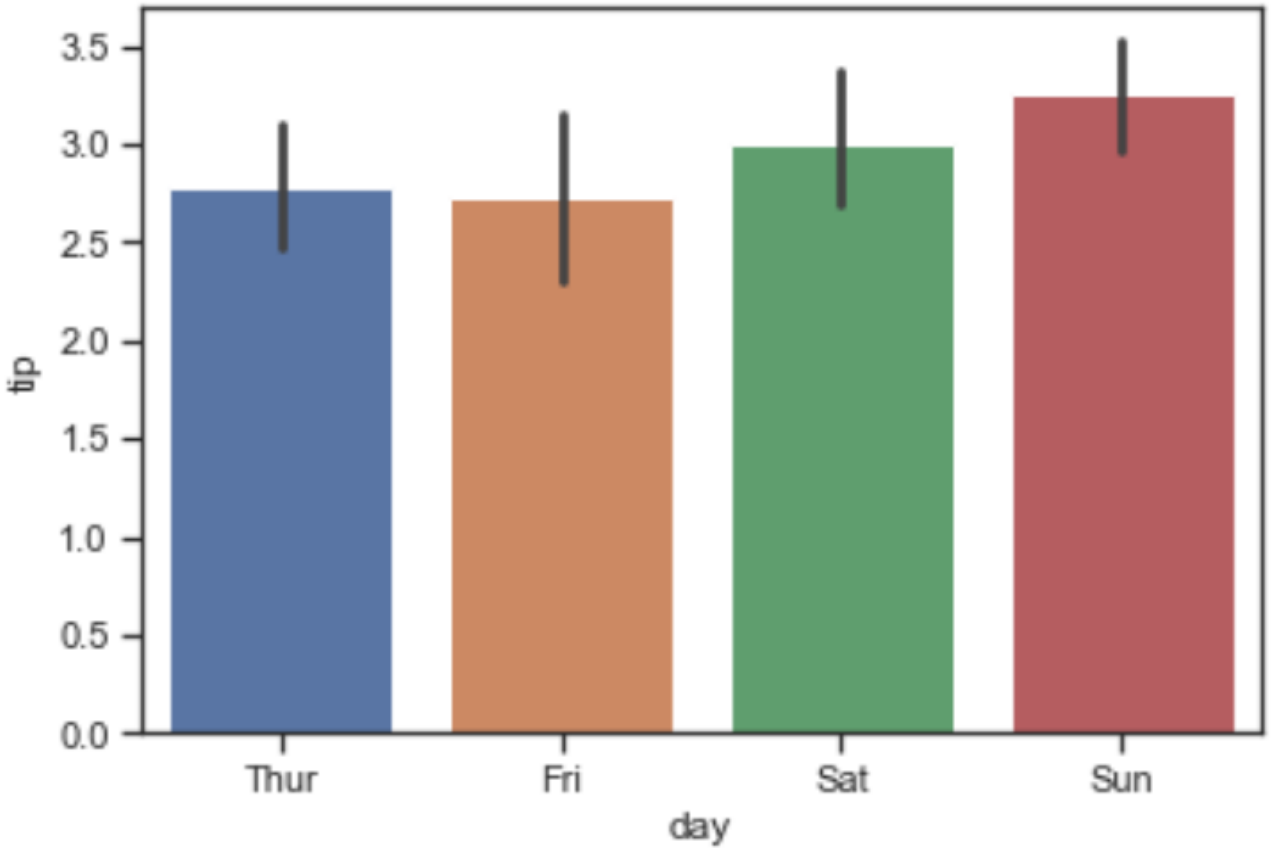
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a24ef6d30>
```



seaborn - basic plots

```
sns.barplot(x="day", y="tip", data=tips)
```

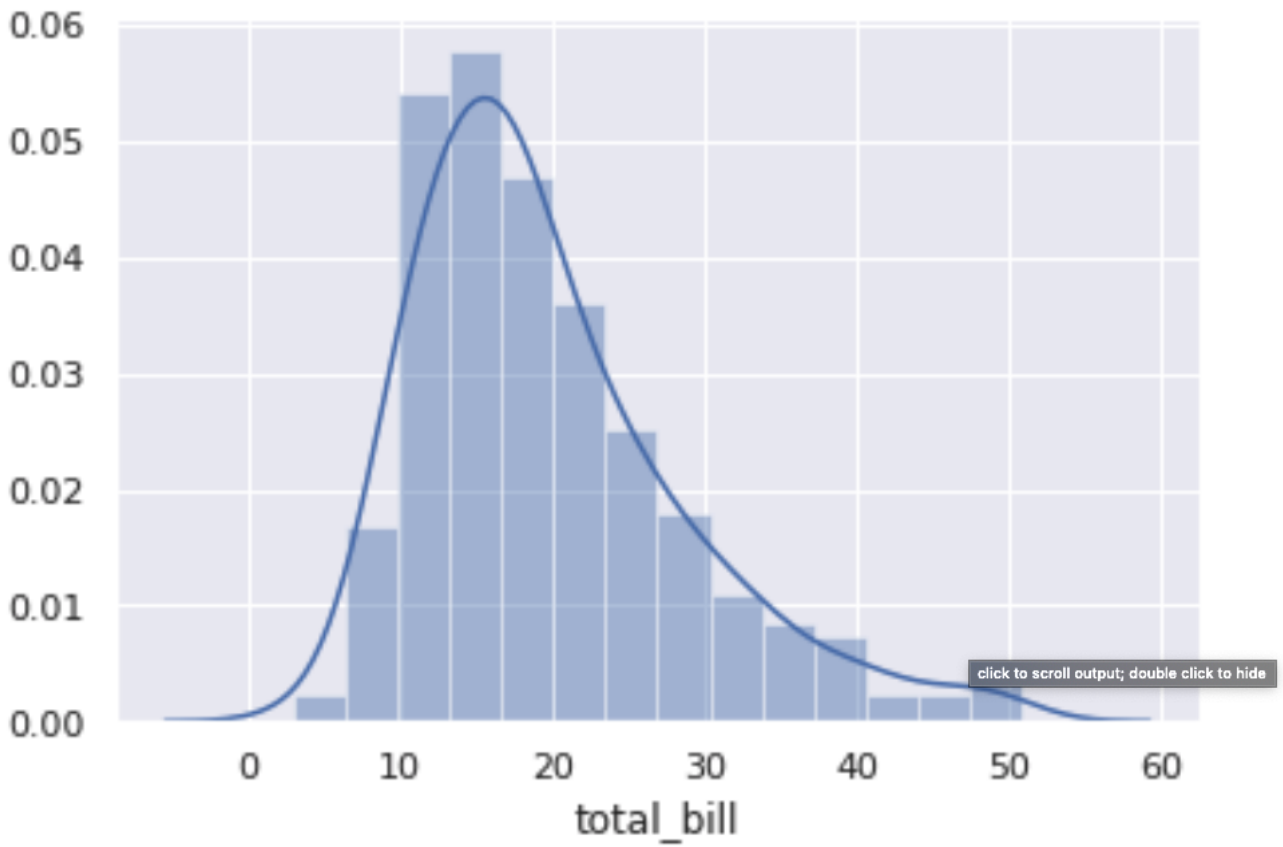
```
<matplotlib.axes._subplots.AxesSubplot at 0x171688f9c08>
```



seaborn - basic plots

```
1  sns.set(style="darkgrid")
2  sns.distplot(tips["total_bill"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a26db24a8>
```



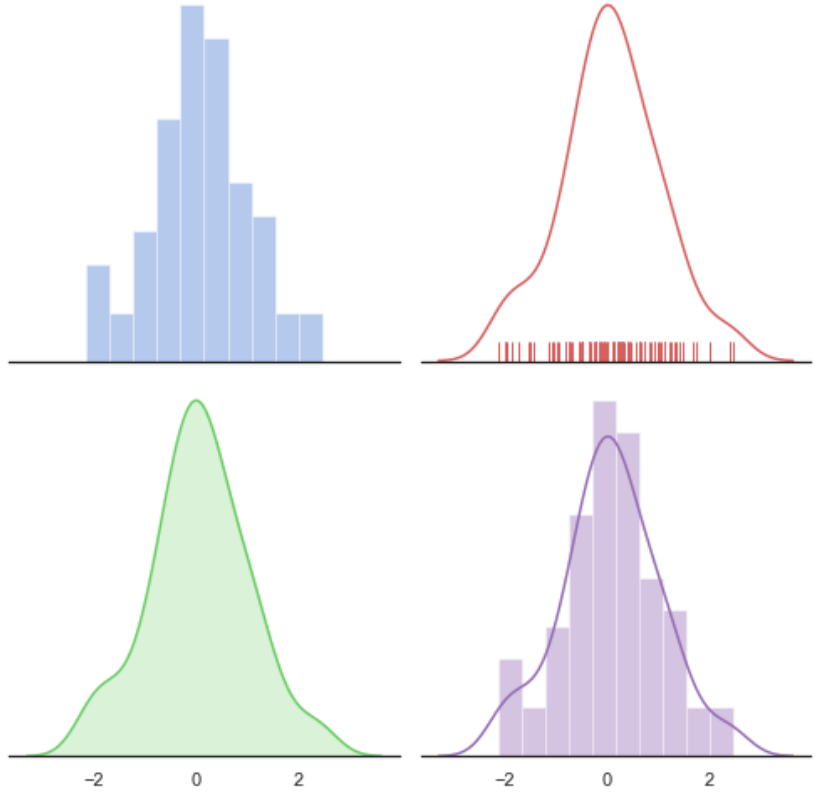
seaborn – predefined plots

- Viloinplot – boxplot에 distribution을 함께표현
- Stripplot – scatter와 category 정보를 함께 표현
- Swarmplot – 분포와 함께 scatter를 함께 표현
- Pointplot – categor별로 numeri의 평균, 신뢰구간 표시
- regplot – scatter + 선형함수를 함께 표시

seaborn - mutliple plots

- 한 개이상의 도표를 하나의 plot에 작성
- Axes를 사용해서 grid를 나누는 방법

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

sns.set(style="white", palette="muted", color_codes=True)
rs = np.random.RandomState(10)

f, axes = plt.subplots(2, 2, figsize=(7, 7), sharex=True)
sns.despine(left=True)

d = rs.normal(size=100)

sns.distplot(d, kde=False, color="b", ax=axes[0, 0])
sns.distplot(d, hist=False, rug=True, color="r", ax=axes[0, 1])
sns.distplot(d, hist=False, color="g", kde_kws={"shade": True}, ax=axes[1, 0])
sns.distplot(d, color="m", ax=axes[1, 1])

plt.setp(axes, yticks=[])
plt.tight_layout()
```



seaborn – predefined mutliple plots

- replot – Numeric 데이터 중심의 분포 / 선형 표시
- catplot – category 데이터 중심의 표시
- FacetGrid – 특정 조건에 따른 다양한 plot을 grid로 표시
- pairplot – 데이터 간의 상관관계 표시
- lmplot – regression 모델과 category 데이터를 함께 표시

SAMSUNG

Data Science Academy