

데이터 시각화 (2024)

데이터과학부 정진명

(jmjung@suwon.ac.kr, 글로벌경상관 918호)

3 주차

Contents

- plot 함수
 - sin곡선 그리기
 - 기울기 함수 그리기
 - 파일 읽어서 그리기
- scatter 함수 소개

sin 곡선 그리기

```
import math

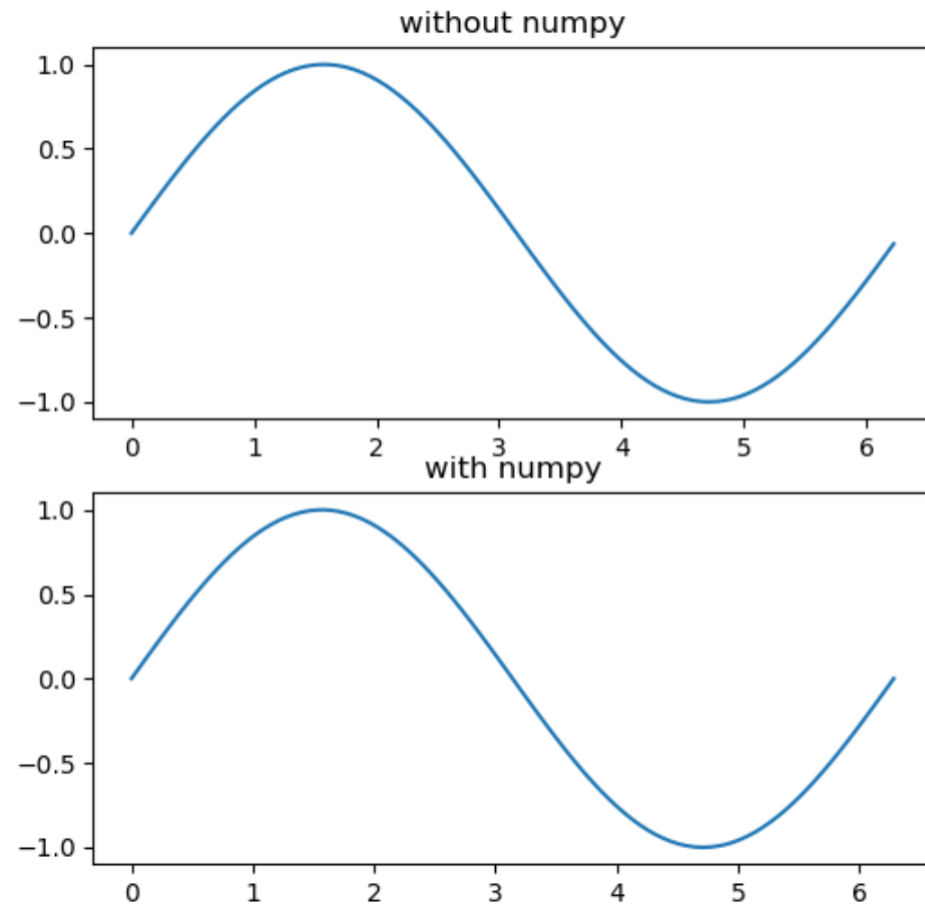
fig=plt.figure(figsize=(6,6), dpi=100)
(ax1, ax2)=fig.subplots(2,1)

## without numpy
T=range(100)
X=[(2*math.pi*t)/len(T) for t in T]
Y=[math.sin(value) for value in X]

ax1.plot(X,Y)
ax1.set_title("without numpy")

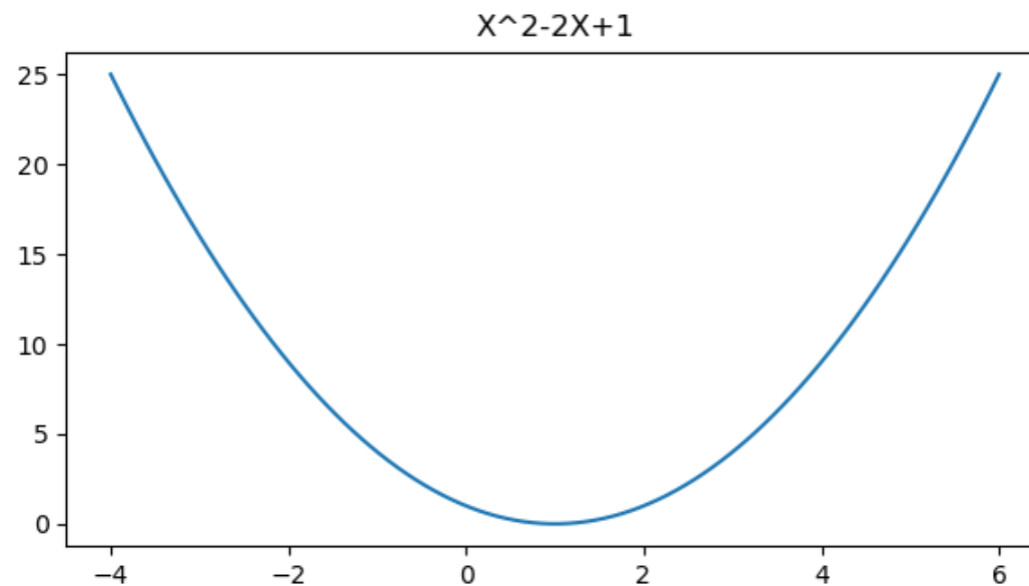
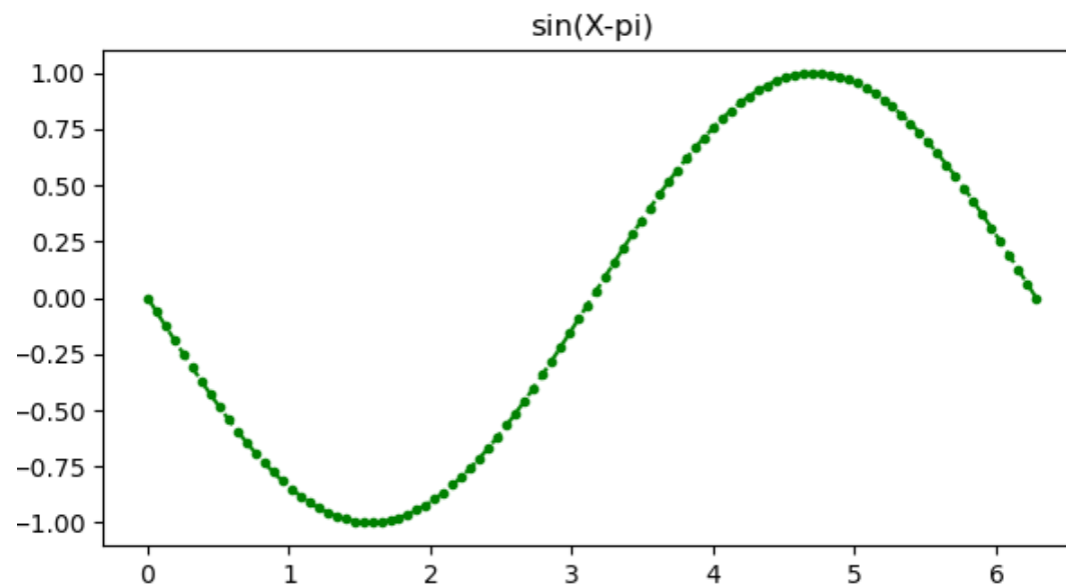
## with numpy
X=np.linspace(0,2*np.pi, 100)
Y=np.sin(X)

ax2.plot(X,Y)
ax2.set_title("with numpy")
```



실습 1

numpy의 linspace와 sin함수를 사용하여
오른쪽과 같은 그림을 그리시오

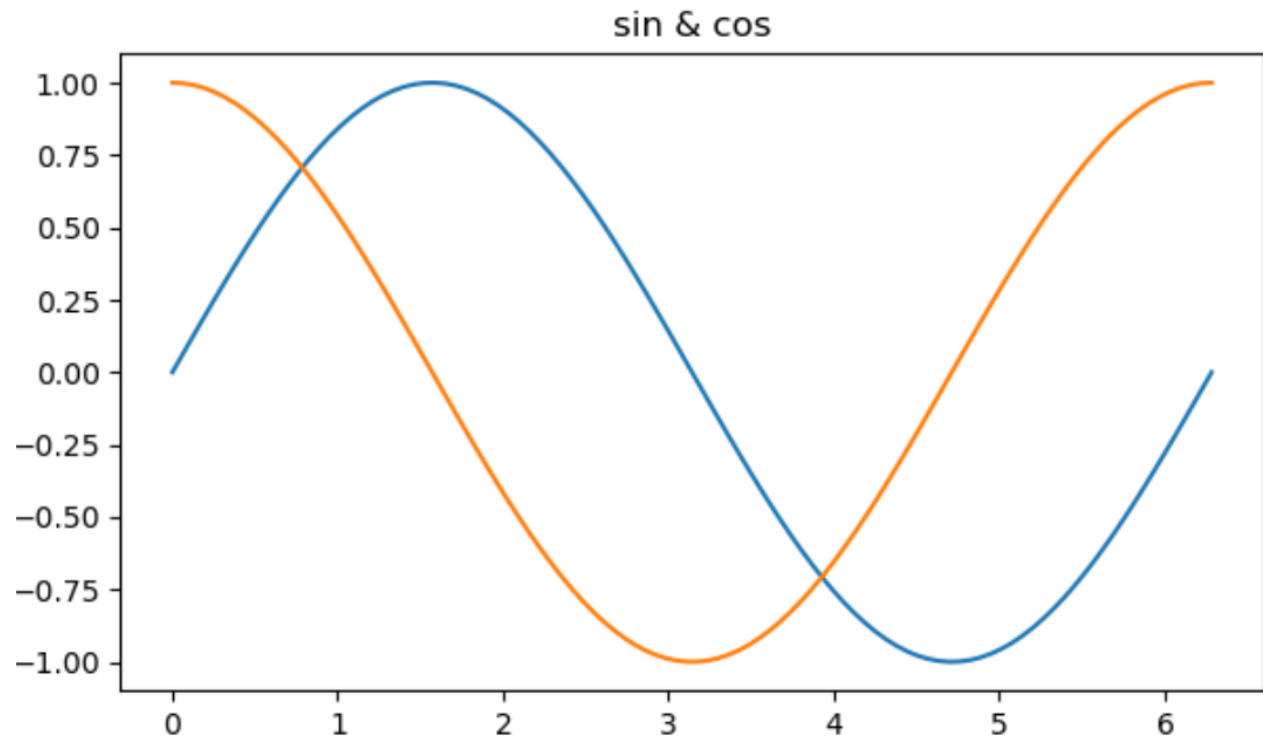


다중 곡선 그리기

```
fig=plt.figure(figsize=(7,4), dpi=100)  
ax=fig.subplots() (← axes는 하나만 생성)
```

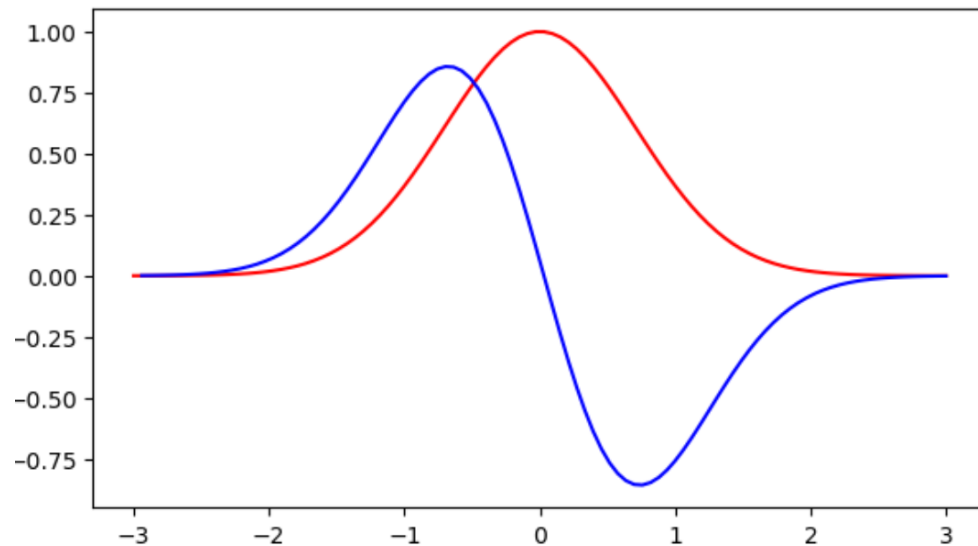
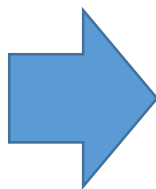
```
X=np.linspace(0,2*np.pi,100)  
Ya=np.sin(X)  
Yb=np.cos(X)
```

```
ax.plot(X,Ya) (← sin을 그린다)  
ax.plot(X,Yb) (← 그 위에 cos을 그린다)  
ax.set_title("sin & cos")
```

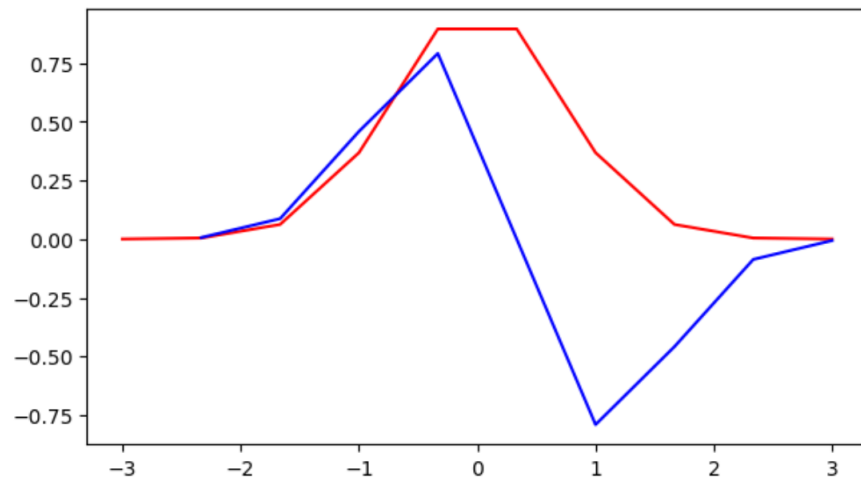


다중 곡선 그리기 2

```
def plot_slope(X1,Y1,ax1):  
    Xs=X1[1:]-X1[:-1]  
    Ys=Y1[1:]-Y1[:-1]  
    ax1.plot(X1[1:], Ys/Xs, 'b')  
    element wise operation in numpy  
fig=plt.figure(figsize=(7,4), dpi=100)  
ax=fig.subplots()  
  
X=np.linspace(-3,3,100)  
Y=np.exp(-X**2)  
  
ax.plot(X,Y,'r')  
plot_slope(X,Y,ax)
```

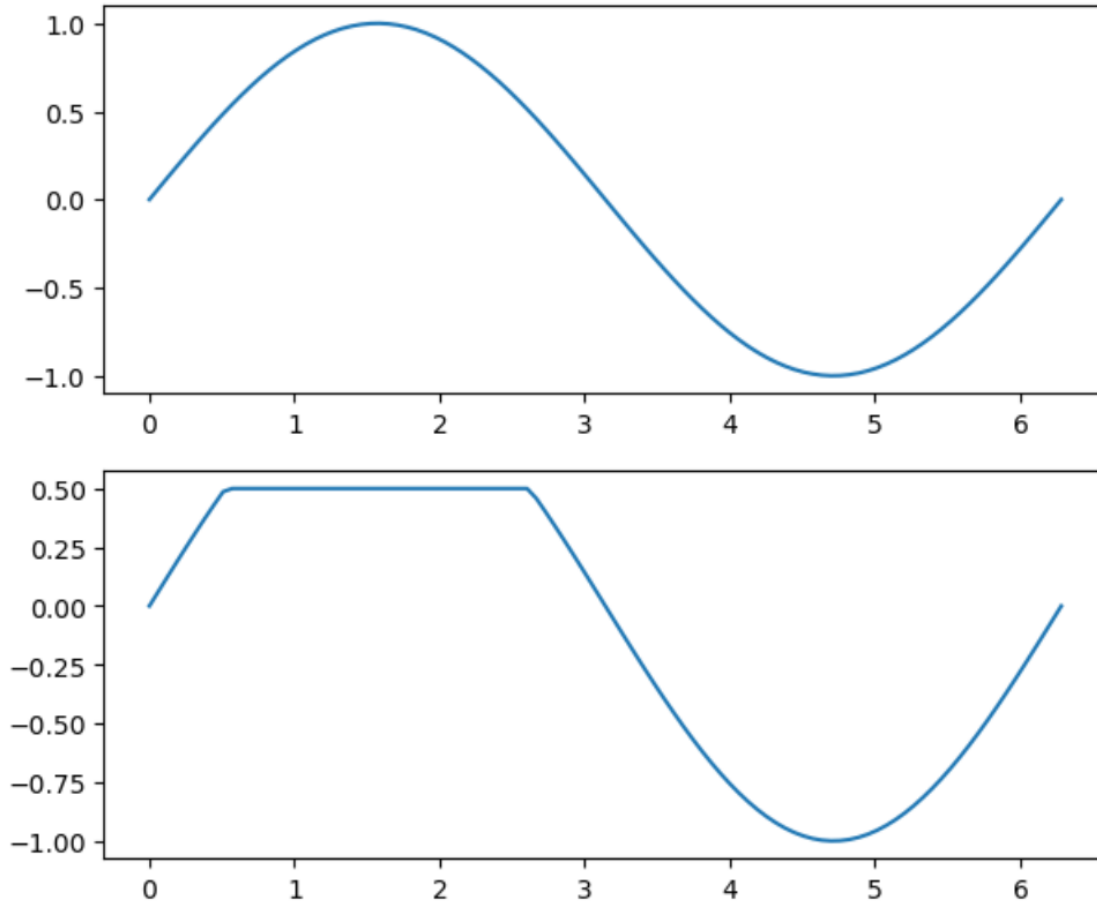


`X=np.linspace(-3,3,100)` → `X=np.linspace(-3,3,10)`



실습 2

- 아래와 같이 두 개의 axes에 그림을 그리시오
- 1) ax1: $y=\sin(x)$
- 2) ax2: $y=\sin(x)$ 인데 y 가 0.5보다 큰 경우 $y=0.5$



← 구현방법

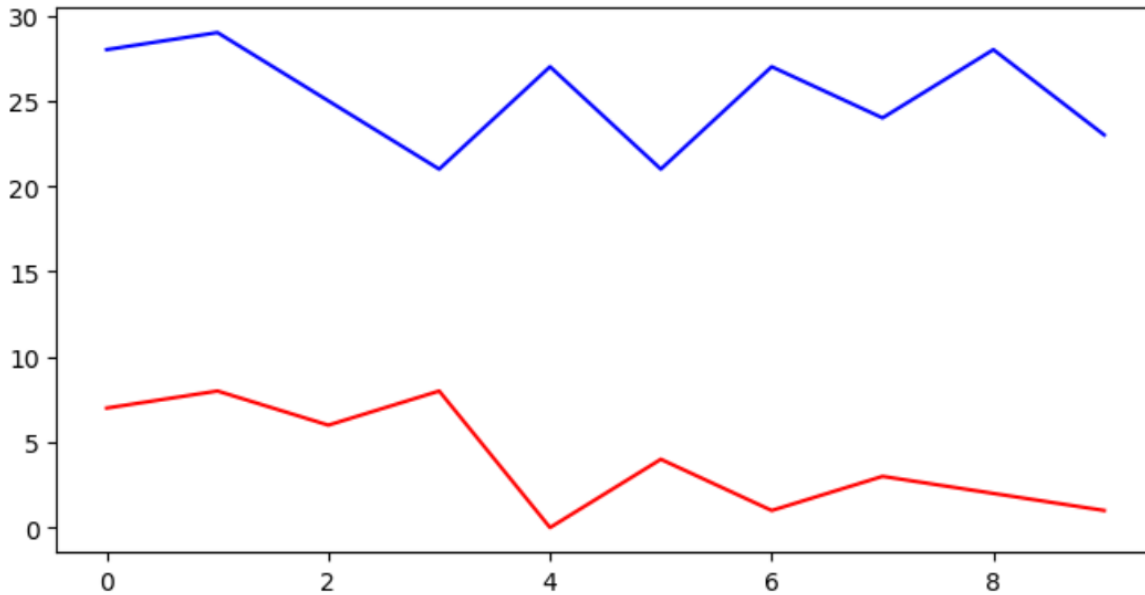
- 1) list comprehension (for if else)
- 2) np.where

실습 3

- 주어진 X, Y1, Y2에 대하여 각 (X,Y1), (X,Y2)에 해당하는 라인을 plot함수를 활용하여 그리시오
- 단, Y의 값들의 합이 짝수일 경우는 빨간색, 홀수일 경우는 파란색으로 그리시오
- Y1, Y2는 랜덤하게 생성됨

```
('Y1_sum: ', 40, 'Y1_left: ', 0)
```

```
('Y2_sum: ', 253, 'Y2_left: ', 1)
```



plot from data 1

- 주어진 3w_d1.txt를 읽어서 X,Y에 대하여 plot함수로 그림을 그리시오
(단 X는 0부터 시작하는 정수)

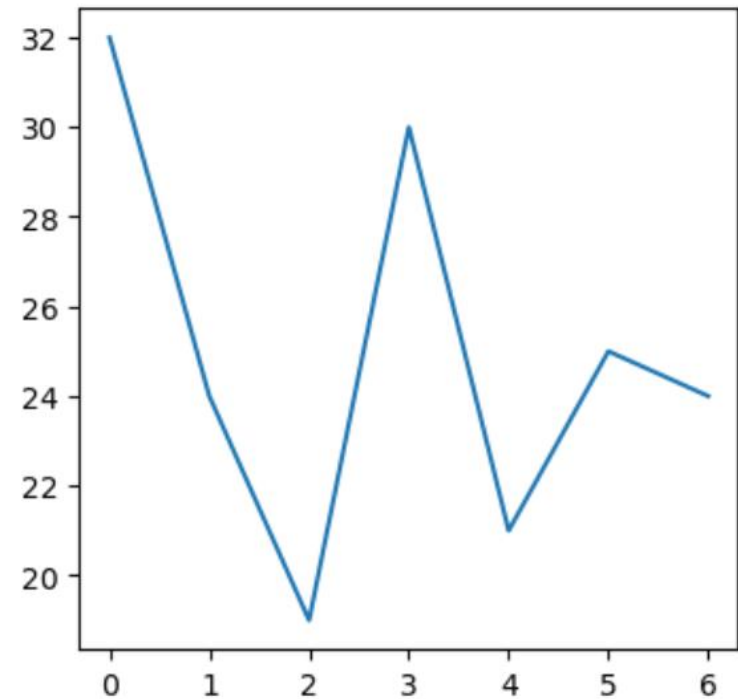
```
fig=plt.figure(figsize=(4,4), dpi=100)
ax=fig.subplots()

data=pd.read_table('data/3w_d1.txt', sep='\t')
X=data.index
Y=data['Y']

ax.plot(X,Y)
```

[<matplotlib.lines.Line2D at 0x2338d836f70>]

| | Y |
|---|----|
| 0 | 32 |
| 1 | 24 |
| 2 | 19 |
| 3 | 30 |
| 4 | 21 |
| 5 | 25 |
| 6 | 24 |



plot from data 2

- 주어진 data_p41.txt를 읽어서 Y1,Y2,Y3에 대하여 plot 하시오
(단 X는 0부터 시작하는 정수)

```
fig=plt.figure(figsize=(12,4), dpi=100)
(ax1, ax2, ax3) =fig.subplots(1,3).flatten()

data=pd.read_table('data/3w_d2.txt', sep='\t')
data

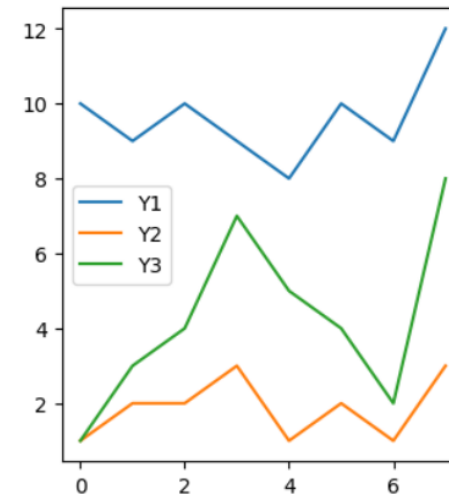
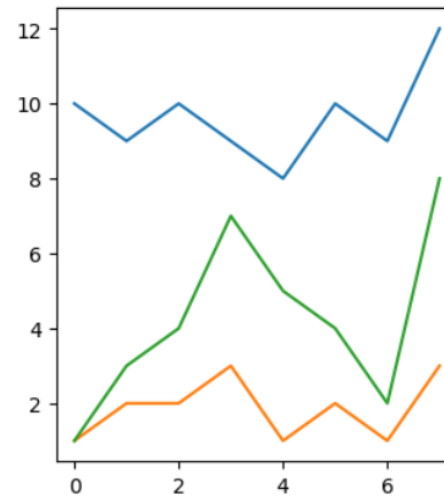
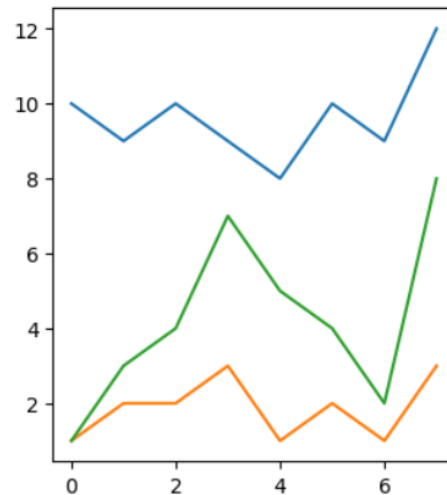
X=data.index

## ax1
Y1=data['Y1']
Y2=data['Y2']
Y3=data['Y3']

ax1.plot(X, Y1)
ax1.plot(X, Y2)
ax1.plot(X, Y3)

## ax2
ax2.plot(X, data)

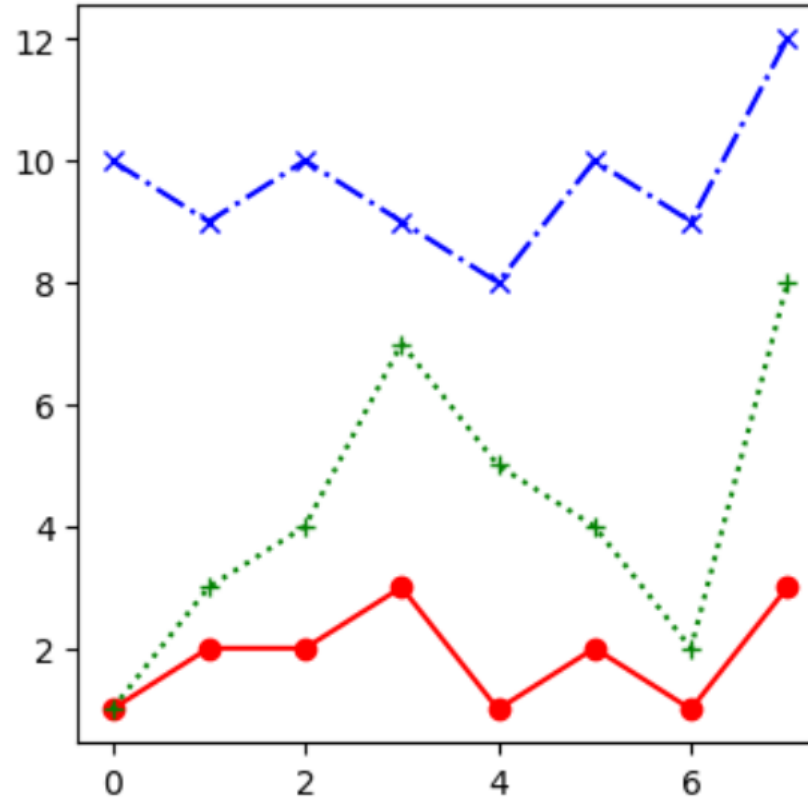
## ax3 (pandas plot)
data.plot.line(ax=ax3)
```



실습 4

- 3w_2d.txt 파일을 읽어서 다음과 같이 marker, line, color를 지정하여 plot하시오
(axes plot을 사용하여 개별로 그리시오)

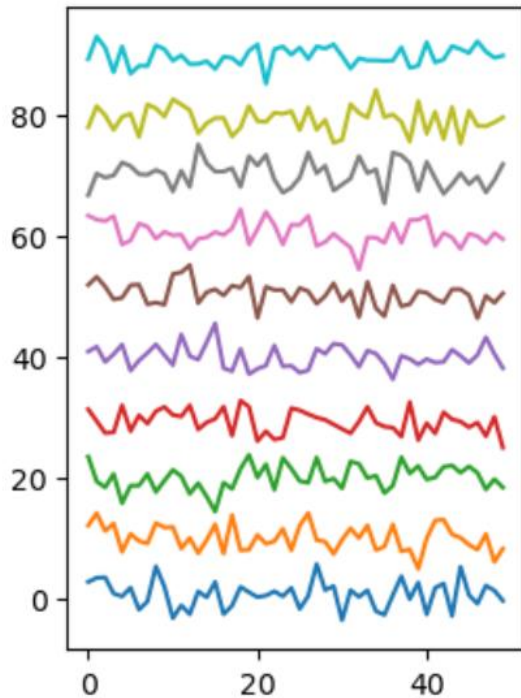
| | Y1 | Y2 | Y3 |
|---|----|----|----|
| 0 | 1 | 1 | 10 |
| 1 | 2 | 3 | 9 |
| 2 | 2 | 4 | 10 |
| 3 | 3 | 7 | 9 |
| 4 | 1 | 5 | 8 |
| 5 | 2 | 4 | 10 |
| 6 | 1 | 2 | 9 |
| 7 | 3 | 8 | 12 |



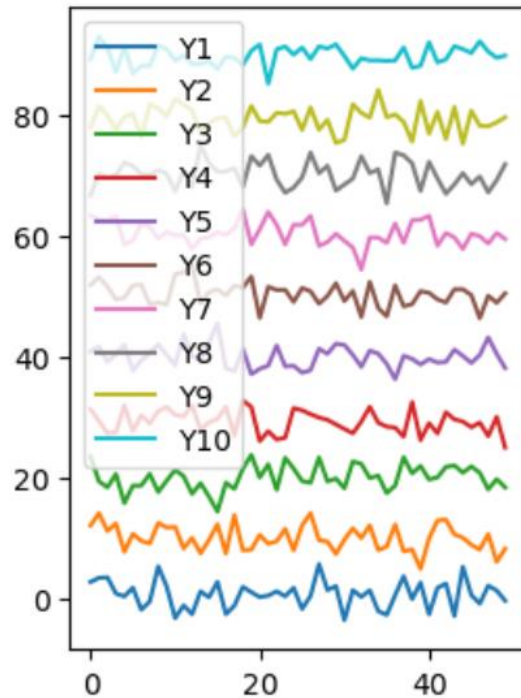
실습 5

- 3w_d3.txt를 읽어서 각 열 (Y1,Y2,Y3...Y10)에 대하여 plot 하시오
(X는 0,1,...,n-1. n: 행의 개수)

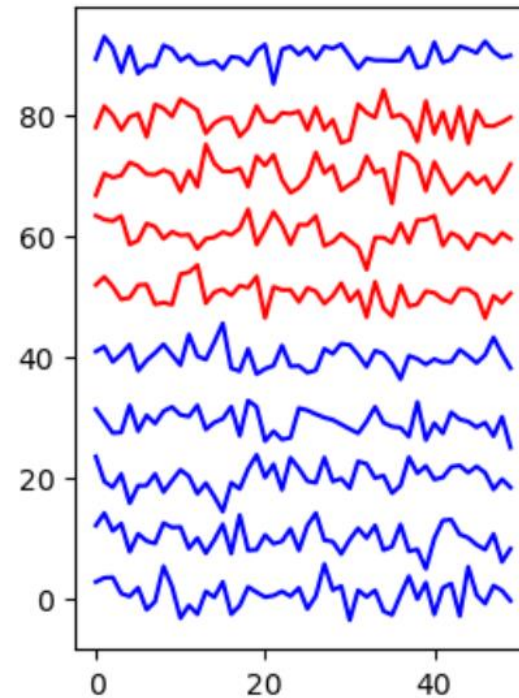
(axe plot, for loop)



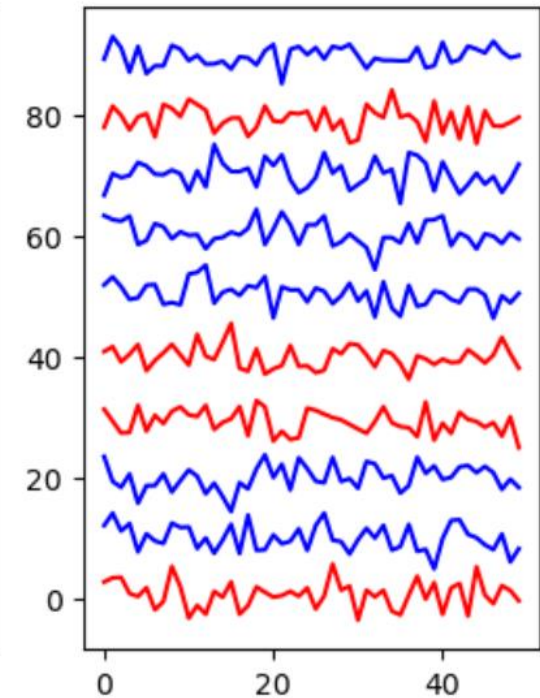
(pandas plot)



(column 평균이
50 이상 80이하인 컬
럼은 빨간색
나머지 컬럼은 파란색)



(Y1, Y4, Y5, Y9
column은 빨간색
나머지 컬럼은 파란색)



scatter

scatter

- scatter
 - X,Y를 점으로 나타내는 함수
 - plot함수에서도 구현이 가능하지만, 점들만 나타낼 때에는 scatter 함수가 적합함
 - cf) plot 함수는 한점이 다른 점을 따라가는 형태로, 데이터의 순서가 의미를 갖는 데이터에 사용 (예: 시계열 데이터)

scatter

matplotlib.pyplot.scatter

```
matplotlib.pyplot.scatter(x, y, s=None, c=None, marker=None, cmap=None, norm=None, vmin=None, vmax=None, alpha=None, linewidths=None,
verts=<deprecated parameter>, edgecolors=None, *, plotnonfinite=False, data=None, **kwargs)
```

[\[source\]](#)

A scatter plot of y vs. x with varying marker size and/or color.

Parameters:

x, y : float or array-like, shape (n,)

The data positions.

s : float or array-like, shape (n,), optional

The marker size in points**2. Default is `rcParams['lines.markersize'] ** 2`.

⋮

Returns:

PathCollection

marker: MarkerStyle

The marker style is a marker. See [marker](#).

cmap: str or Colormap

A Colormap ins

norm: Normalize, de

If `c` is an array of the colormap of the image.

vmin, vmax : float, de

vmin and *vmax*
cmap. If None,
when *norm* is g

alpha : float, default: 0.05

The alpha blend

linewidths : float or array

The linewidth of the π transition is also affected by the laser intensity as well.

edgecolors : {'face', 'r'
(default: 'face')

The edge color

- 'face': The
- 'none': N
- A color o

For non-filled m

plotnonfinite : bool, d

Set to plot point

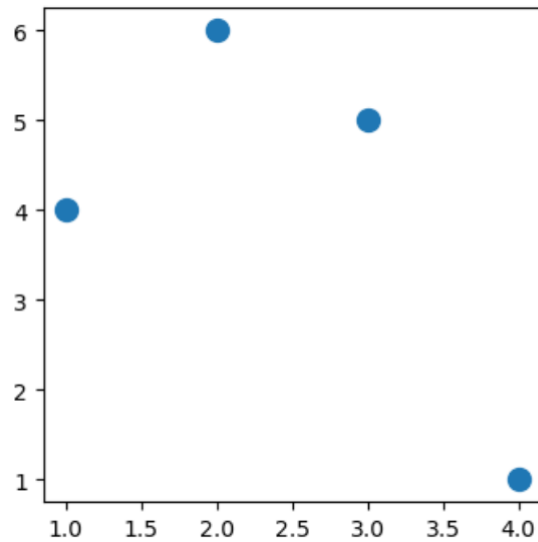
scatter

```
fig=plt.figure(figsize=(4,4), dpi=100)  
ax=fig.subplots()
```

```
X=[1,2,3,4]
```

```
Y=[4,6,5,1]
```

```
ax.scatter(X,Y, s=100)
```



scatter

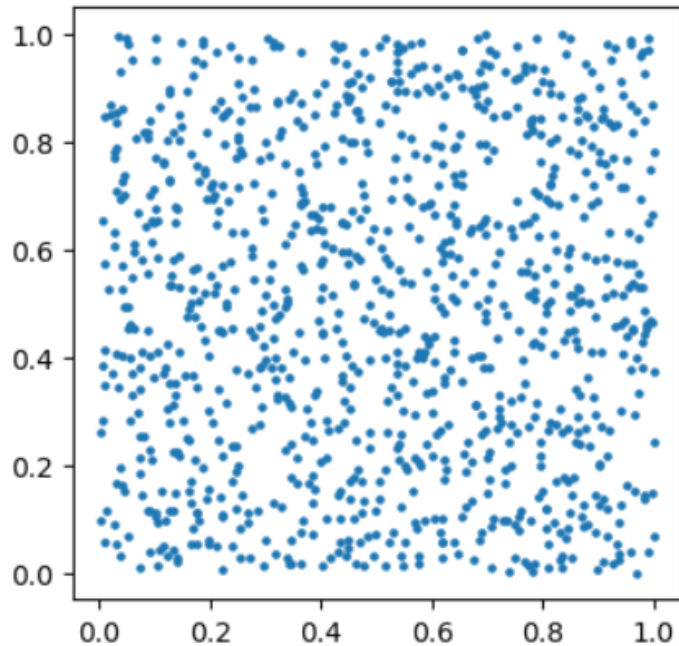
```
fig=plt.figure(figsize=(4,4), dpi=100)
ax=fig.subplots()

data=np.random.uniform(size=(1000,2))
type(data)      numpy package
data[:10]

ax.scatter(data[:,0], data[:,1], s=5)|
```

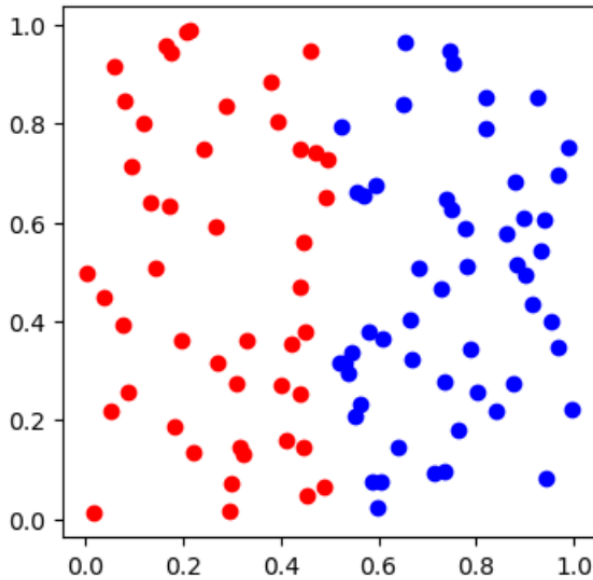
x **y**

→ (0~1) 범위의 uniform distribution 에서 추출한 값으로
1000 by 2 matrix 생성



실습 6

- 아래 그림과 같이 100개의 점을 두 가지 색깔로 scatter plot 하시오
 - 각 점의 x, y 좌표는 0~1에서 uniform하게 sampling한 값
 - 빨강 그룹의 x 좌표는 0~0.5, 파랑 그룹의 x 좌표는 0.5~1



실습 7

```
fig=plt.figure(figsize=(10,4), dpi=100)
ax1, ax2 =fig.subplots(1,2)

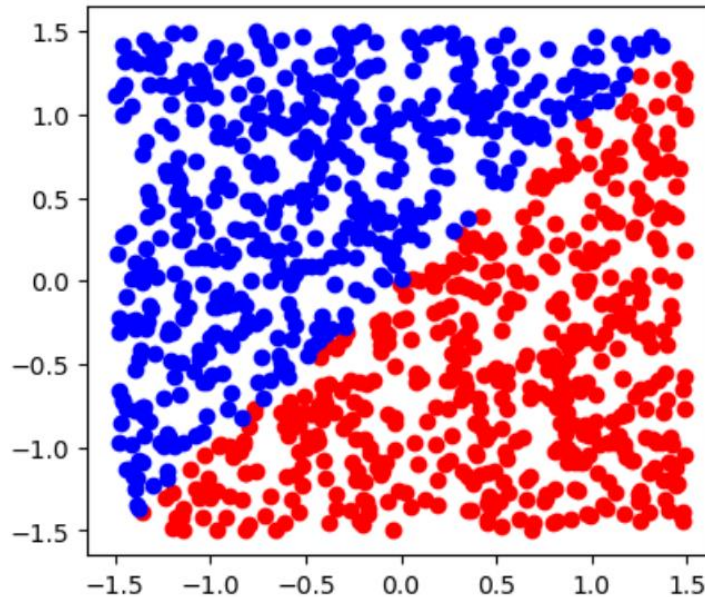
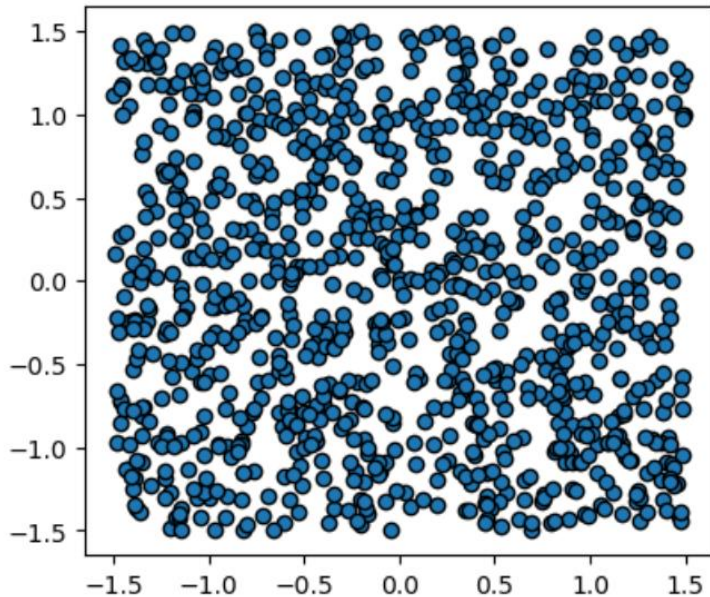
data=np.random.uniform(-1.5,1.5,size=(1000,2))

X=data[:,0]
Y=data[:,1]

_=ax1.scatter(X, Y, edgecolor='k')
```

코드작성

- 아래 그림과 같이 1000개의 점을 두 가지 색깔로 scatter plot 하시오
 - X의 값이 Y값보다 큰 경우: 빨강
 - X의 값이 Y값보다 작은 경우: 파랑



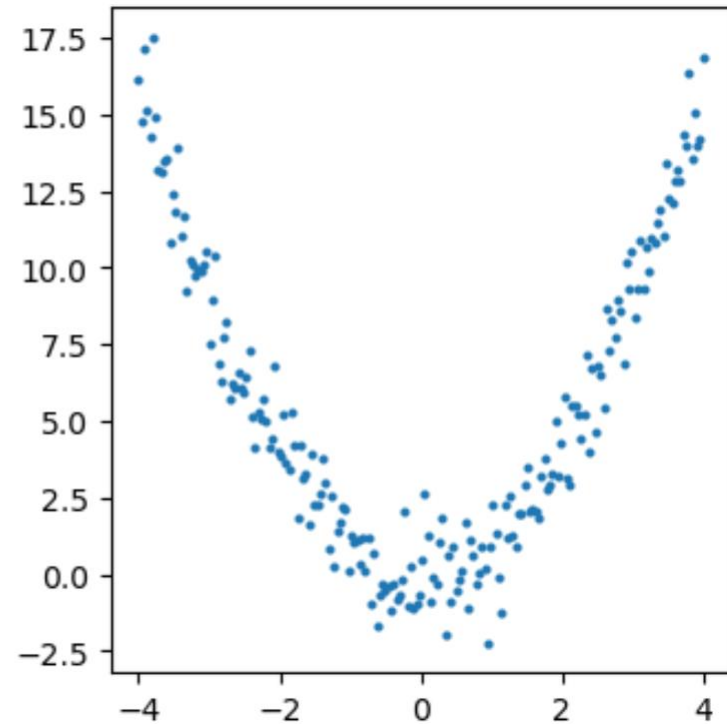
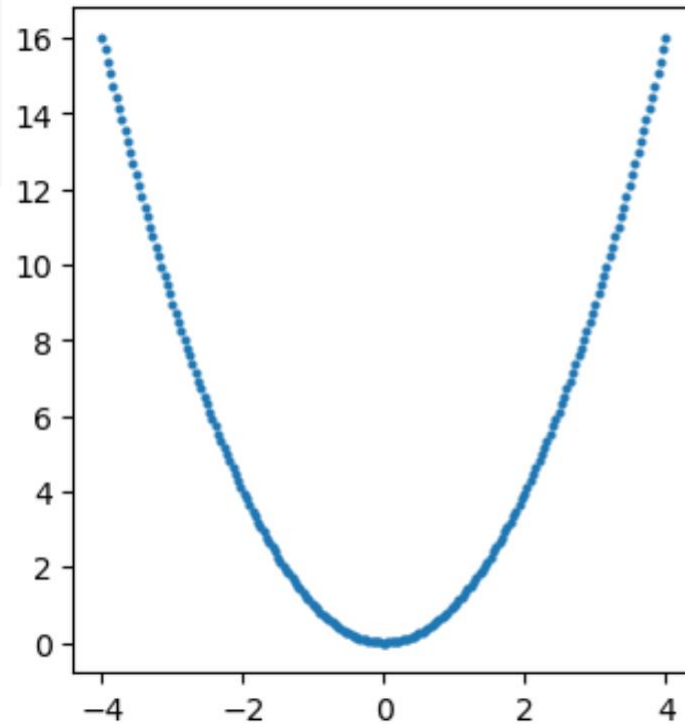
특정 함수에 noise 추가해서 scatter 해보기

```
fig=plt.figure(figsize=(12,6), dpi=100)  
(ax1,ax2)=fig.subplots(1,2)
```

```
X=np.linspace(-4,4,200)  
n=np.random.normal(0,1,size=200)
```

```
Y0=X**2  
Y1=X**2+n
```

```
ax1.scatter(X,Y0,s=4)  
ax2.scatter(X,Y1,s=4)
```



Q & A

Thank you