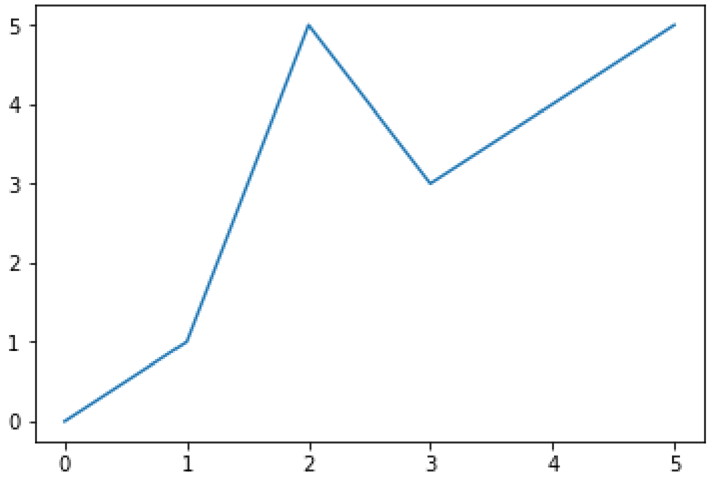
%matplotlib inline 모드로 그래프 작성

```
import matplotlib.pyplot as plt
```

```
%matplotlib inline
```

```
data_list = [0,1,5,3,4,5]

plt.figure
plt.plot(data_list)
plt.show()
```



%matplotlib notebook 모드로 그래프 작성

- jupyter의 기능

```
%matplotlib notebook
```

```
data_list = [0,1,5,3,4,5]

plt.figure
plt.plot(data_list)
plt.show()
```

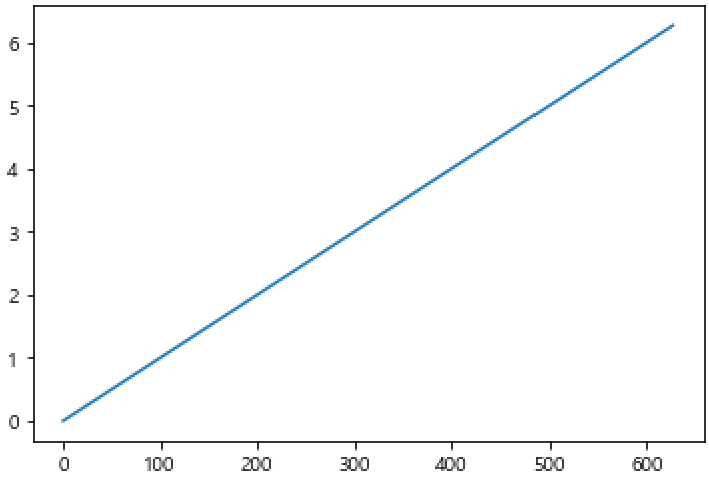
Numpy 배열 그리기

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

```
from matplotlib import font_manager, rc
```

```
font_name = font_manager.FontProperties(fname="C:/Windows/Fonts/MALGUN.TTF").get_name()
rc('font', family = font_name)

plt.rcParams['figure.figsize'] = (10,6)

%matplotlib inline
```

```
t = np.arange(0, 2*3.14, 0.01)

plt.figure()
plt.plot(t)
plt.show()
```

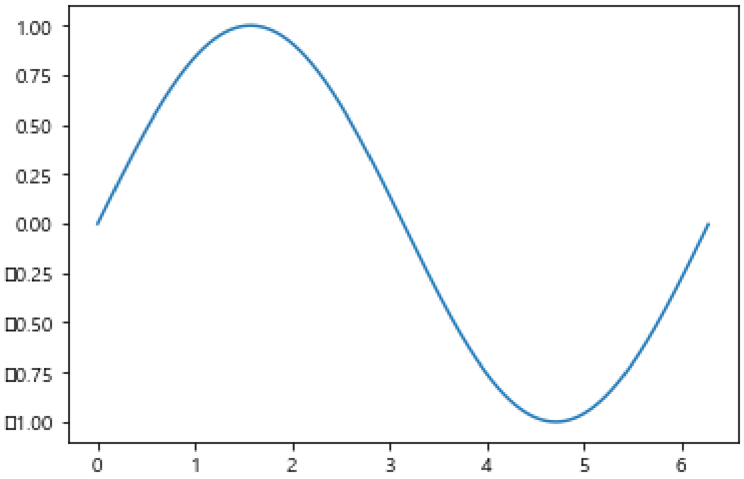


```
## numpy를 이용하여 시간축과 함수 그래프
```

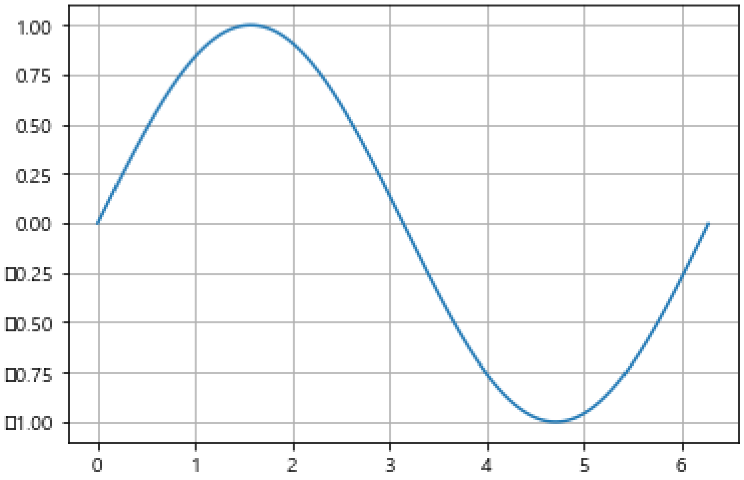
```
import math



P1 = math.pi
P1
```

```
3.141592653589793
```

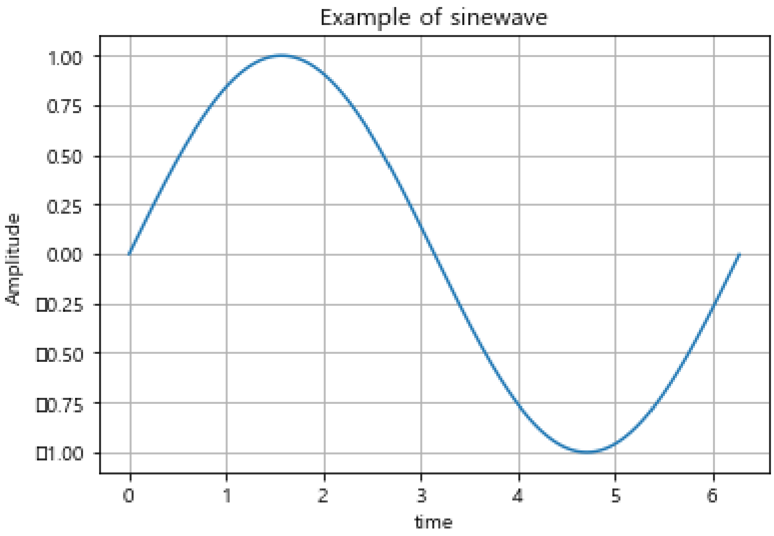
```
t = np.arange(0, 2*P1, 0.01)
y = np.sin(t)

plt.figure(figsize = (6,4))
plt.plot(t,y)
plt.show()
```



```
plt.figure(figsize = (6,4))
plt.plot(t,y)
plt.grid()
plt.show()
```



```
plt.figure(figsize = (6,4))
plt.plot(t,y)
plt.grid()
plt.xlabel('time')
plt.ylabel('Amplitude')
plt.title('Example of sinewave')
plt.show()
```



```
dy = np.diff(y)    #numpy의 차분 함수 사용
dy[:10]
```

```
array([0.00999983, 0.00999883, 0.00999683, 0.00999383, 0.00998984,
       0.00998484, 0.00997884, 0.00997185, 0.00996386, 0.00995487])
```

```
dy = np.insert(dy, 0, 0)/0.01     #차분의 특성 : 맨 앞에 의미없는 값 두기
dy[:10]
```

```
array([0.        , 0.99998333, 0.99988334, 0.99968335, 0.9993834 ,
       0.99898351, 0.99848372, 0.99788409, 0.99718466, 0.99638552])
```

```
a = np.array([[1,1], [2,2], [3,3]])
a
```

```
array([[1, 1],
       [2, 2],
       [3, 3]])
```

```
np.insert(a, 2, 5)
```

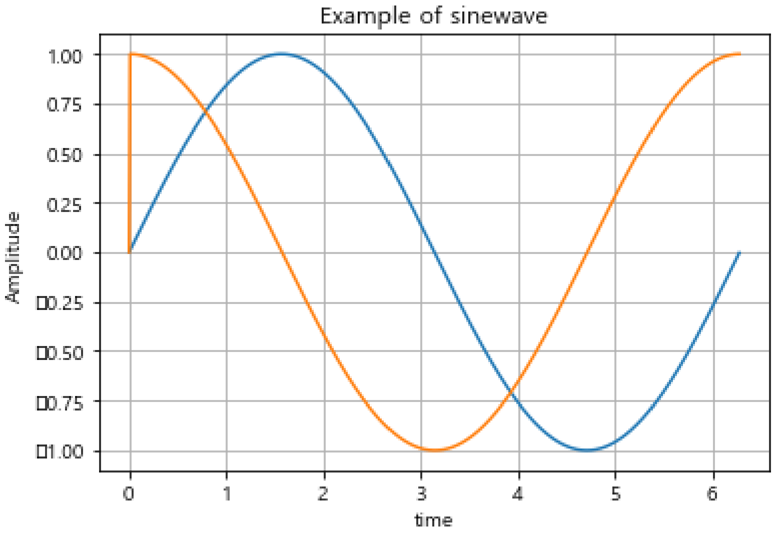
```
array([1, 1, 5, 2, 2, 3, 3])
```

```
np.insert(a, 2, 5, axis=1)
```

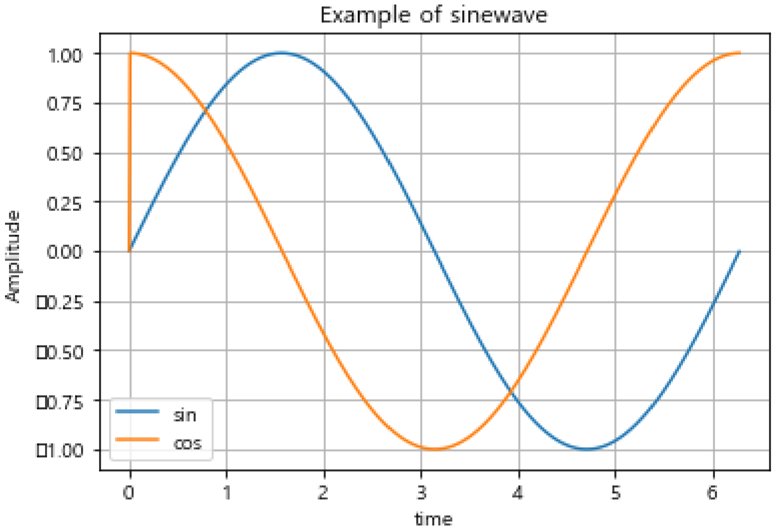
```
array([[1, 1, 5],
       [2, 2, 5],
       [3, 3, 5]])
```

두 개의 그래프 작성

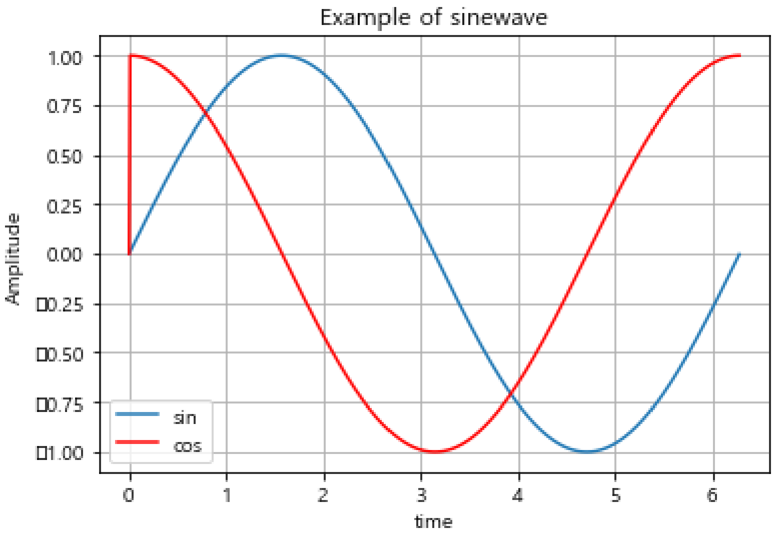
```
plt.figure(figsize=(6,4))
plt.plot(t,  y)
plt.plot(t, dy)
plt.grid()
plt.xlabel('time')
plt.ylabel('Amplitude')
plt.title('Example of sinewave')
plt.show()
```



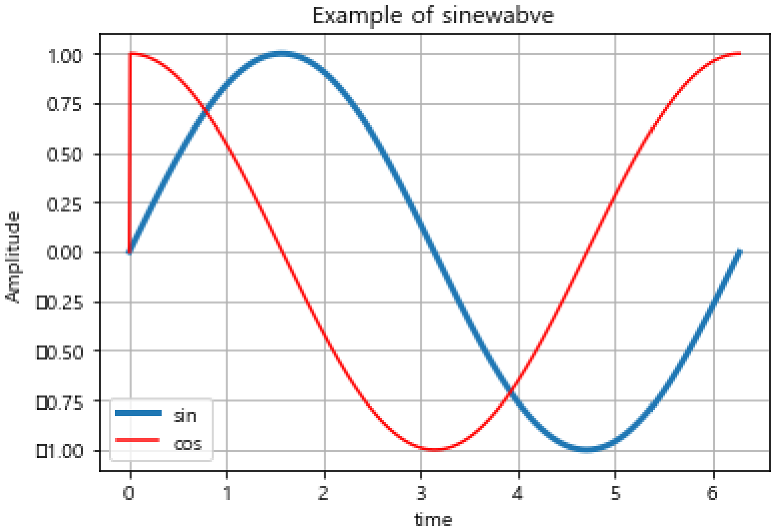
```
plt.figure(figsize=(6,4))
plt.plot(t,  y, label='sin')
plt.plot(t, dy, label='cos')
plt.grid()
plt.legend()                # legend
plt.xlabel('time')
plt.ylabel('Amplitude')
plt.title('Example of sinewave')
plt.show()
```



```
plt.figure(figsize=(6,4))
plt.plot(t, y, label='sin')
plt.plot(t, dy, 'r', label='cos')   # color : ['b', 'g', 'r', 'c', 'm', 'y', 'k', 'w' ]
plt.grid()
plt.legend()
plt.xlabel('time')
plt.ylabel('Amplitude')
plt.title('Example of sinewave')
plt.show()
```

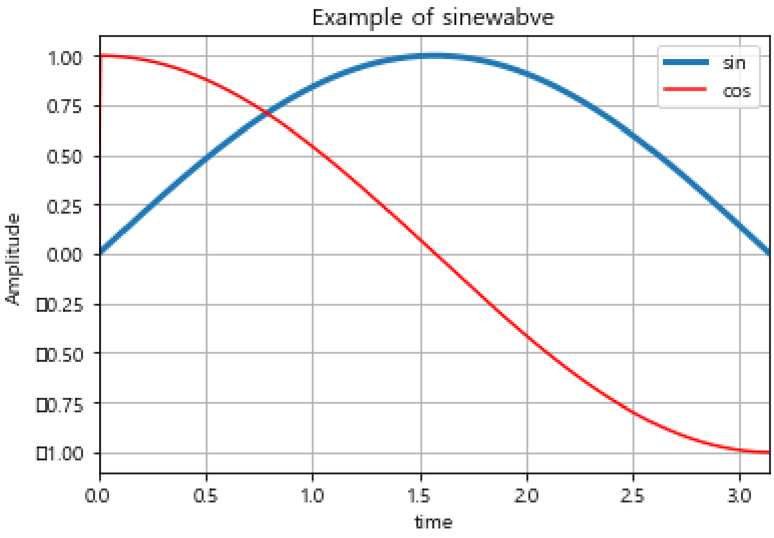


```
plt.figure(figsize=(6,4))
plt.plot(t, y, lw=3, label = 'sin')
plt.plot(t, dy, 'r', label = 'cos')
plt.grid()
plt.legend()
plt.xlabel('time')
plt.ylabel('Amplitude')
plt.title('Example of sinewabve')
plt.show()
```

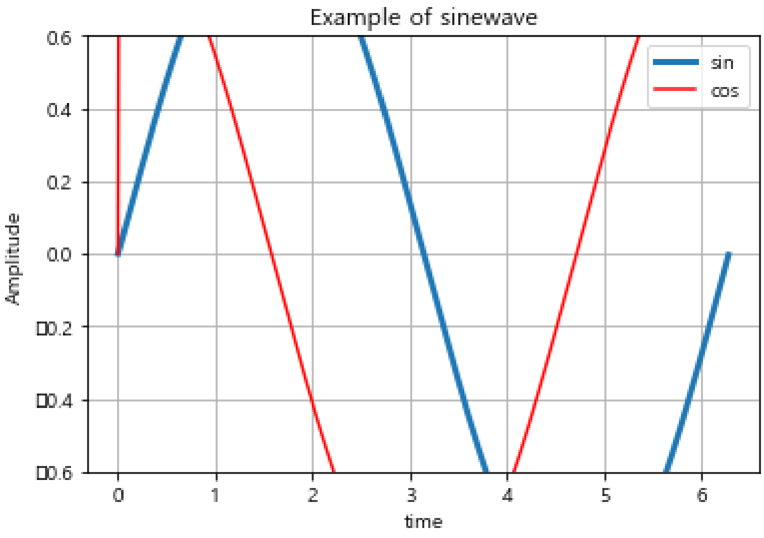


축별로 그리는 범위 조절

```
plt.figure(figsize=(6,4))
plt.plot(t, y, lw=3, label = 'sin')
plt.plot(t, dy, 'r', label = 'cos')
plt.grid()
plt.legend()
plt.xlabel('time')
plt.ylabel('Amplitude')
plt.title('Example of sinewabve')
plt.xlim(0, 3.14)
plt.show()
```



```
plt.figure(figsize=(6,4))
plt.plot(t, y, lw=3, label='sin')
plt.plot(t, dy, 'r', label='cos')
plt.grid()
plt.legend()
plt.xlabel('time')
plt.ylabel('Amplitude')
plt.title('Example of sinewave')
plt.ylim(-1.2/2, 1.2/2)     # set the ylim to ymin, ymax
plt.show()
```



```
#변수초기화
%reset
```

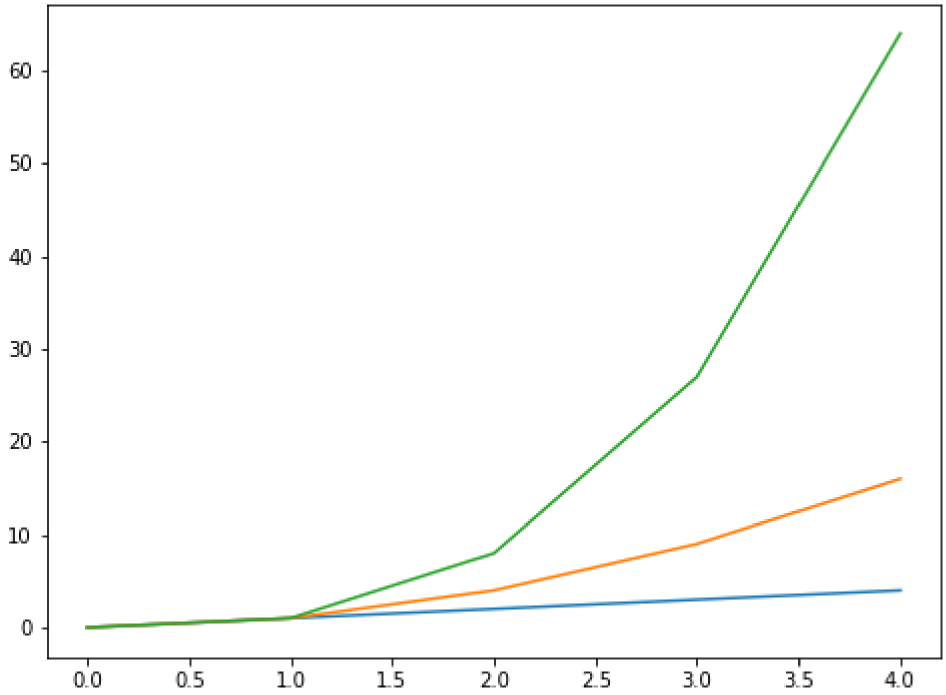
두 개 이상의 그래프 그리기 다른 방법

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

from matplotlib import font_manager, rc

# font_name = font_manager.FontProperties(fname="/Library/Fonts/AppleGothic.ttf").get_name()
```

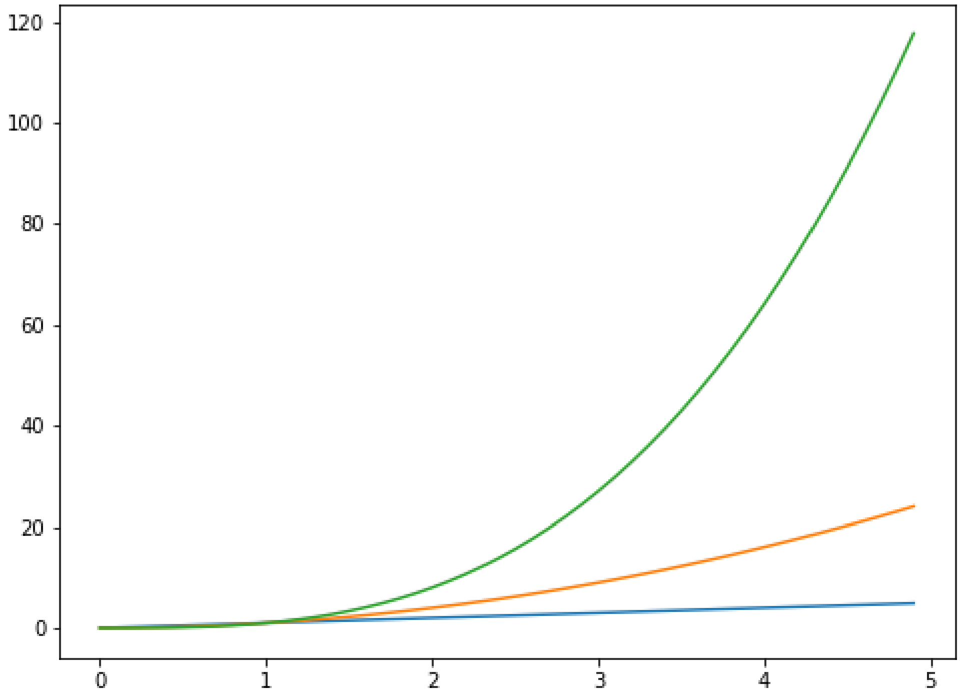
```
font_name = font_manager.FontProperties(fname="C:/Windows/Fonts/MALGUN.TTF").get_name()
rc('font', family=font_name)

%matplotlib inline
```

```
plt.rcParams['figure.figsize'] = (8,6)
```

```
t = np.arange(0, 5, 1.0)

plt.figure
plt.plot(t, t, t, t**2, t**3)
plt.show()
```

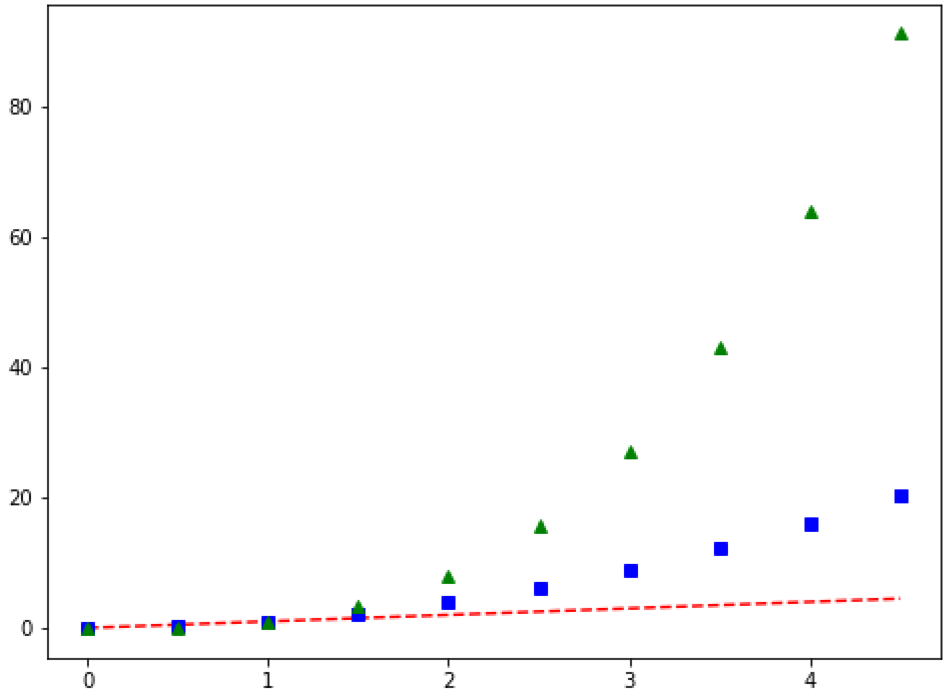


```
t = np.arange(0, 5, 0.1)     # 0.1, 0.5, 1.0

plt.figure
plt.plot(t, t, t, t**2, t, t**3)
plt.show()
```

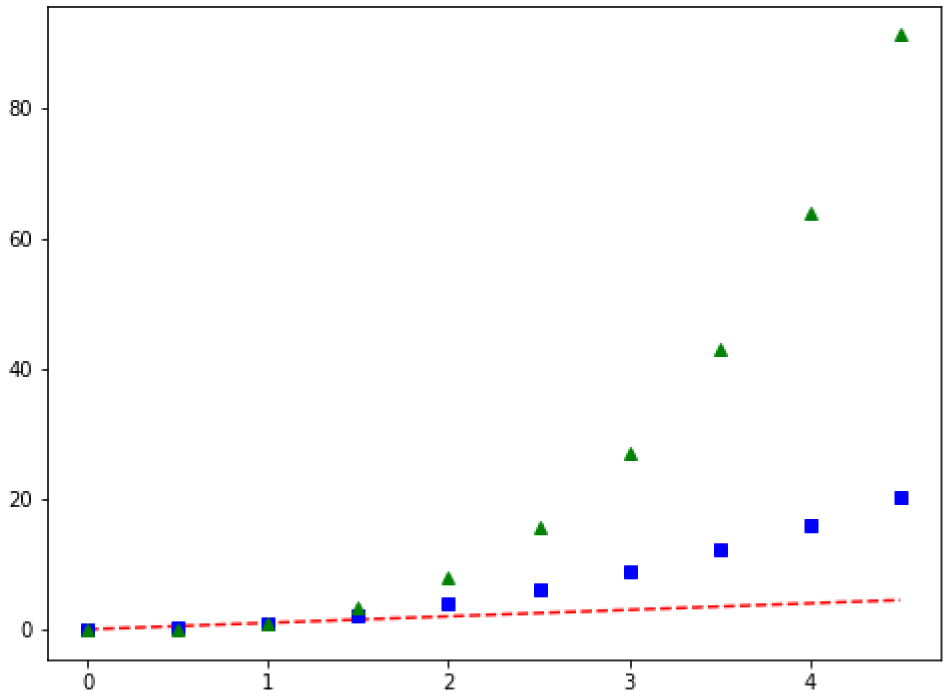


마커 적용

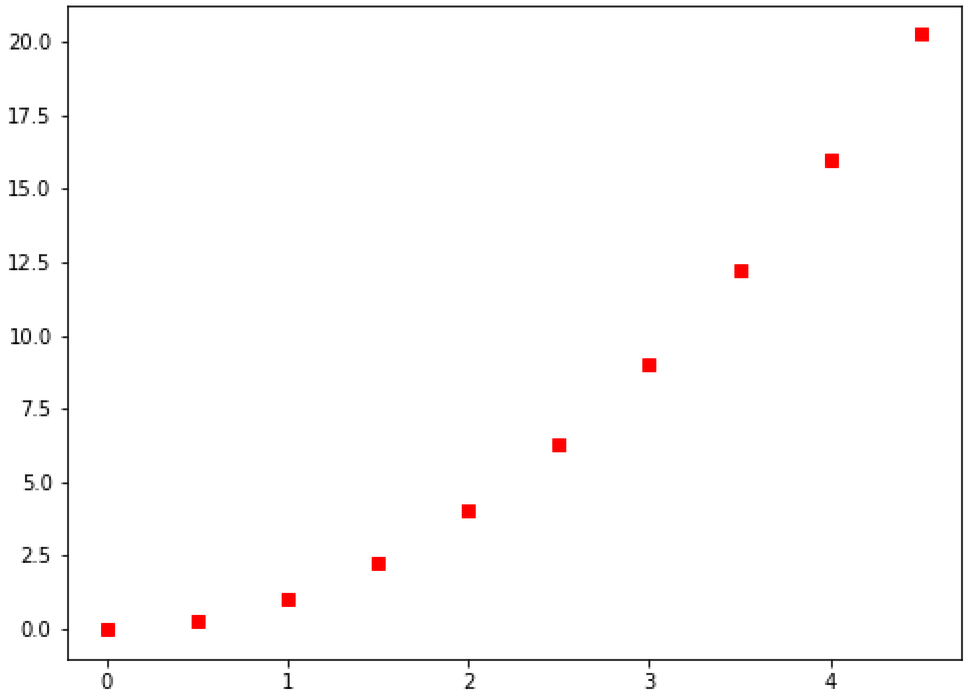
```
t = np.arange(0, 5, 0.5)

plt.figure
plt.plot(t, t, 'r--')
plt.plot(t, t**2, 'bs')
plt.plot(t, t**3, 'g^')
plt.show()
```

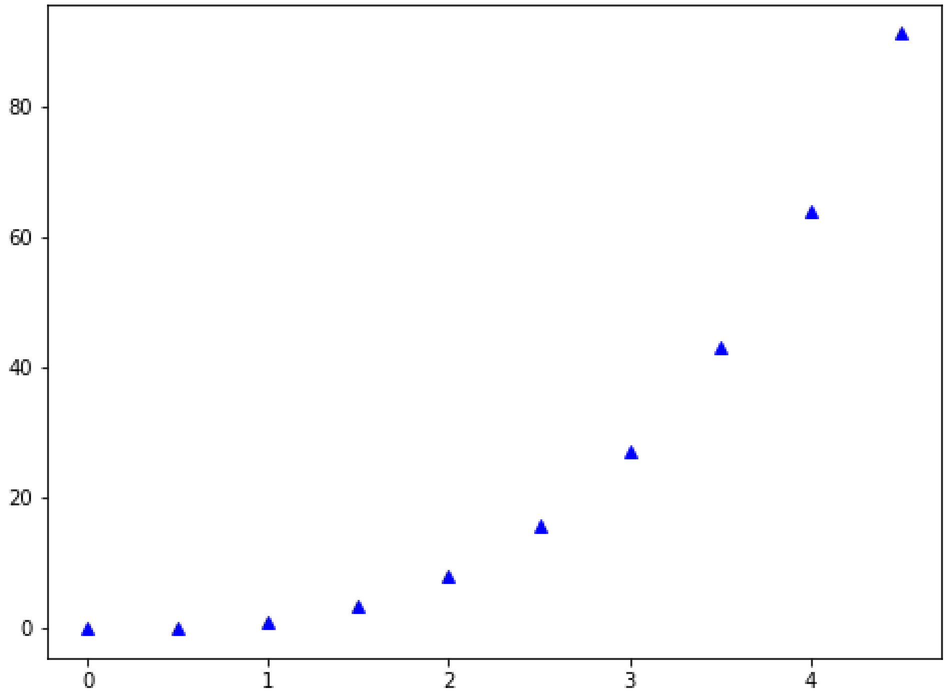


```
t = np.arange(0, 5, 0.5)

plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```



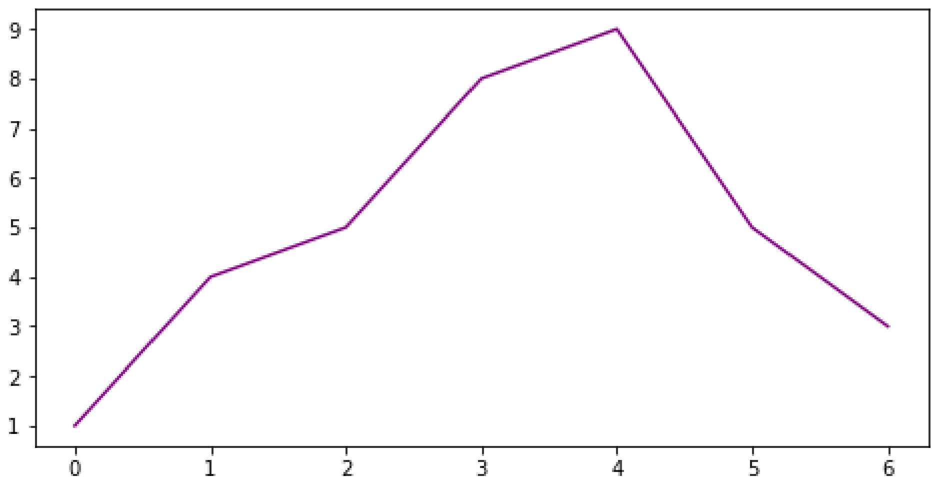
```
t = np.arange(0, 5, 0.5)

fig1 = plt.figure(1)
plt1 = plt.plot(t, t**2, 'rs')

fig2 = plt.figure(2)
plt2 = plt.plot(t, t**3, 'b^')

plt.show()
```



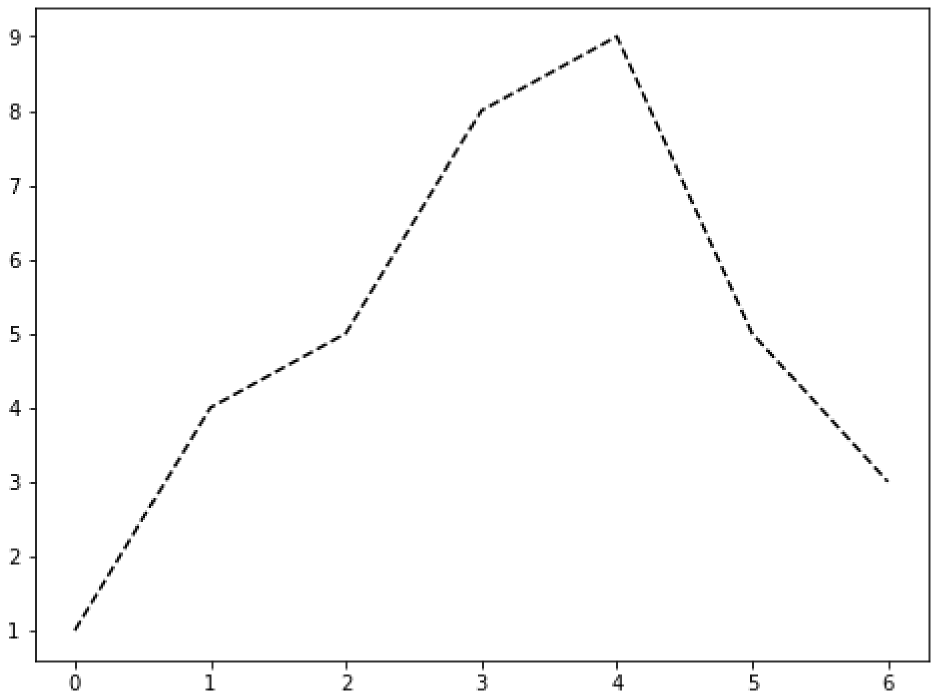


```
import matplotlib.pyplot as plt

t = [0, 1, 2, 3, 4, 5, 6]
y = [1, 4, 5, 8, 9, 5, 3]
```

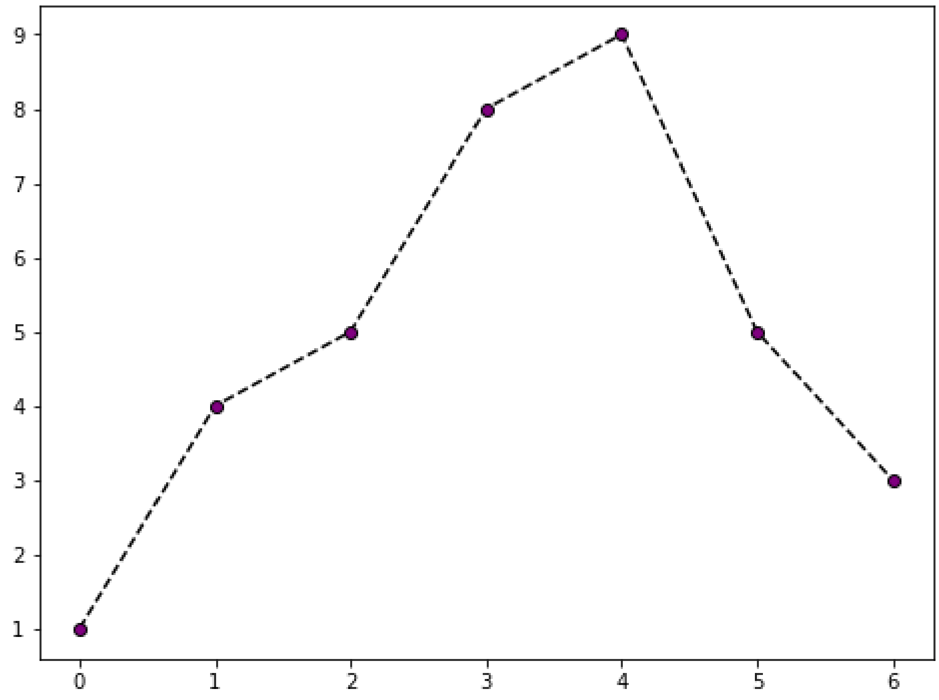
```
plt.figure(figsize=(8,4))
plt.plot(t,y,color='purple')
plt.show()
```



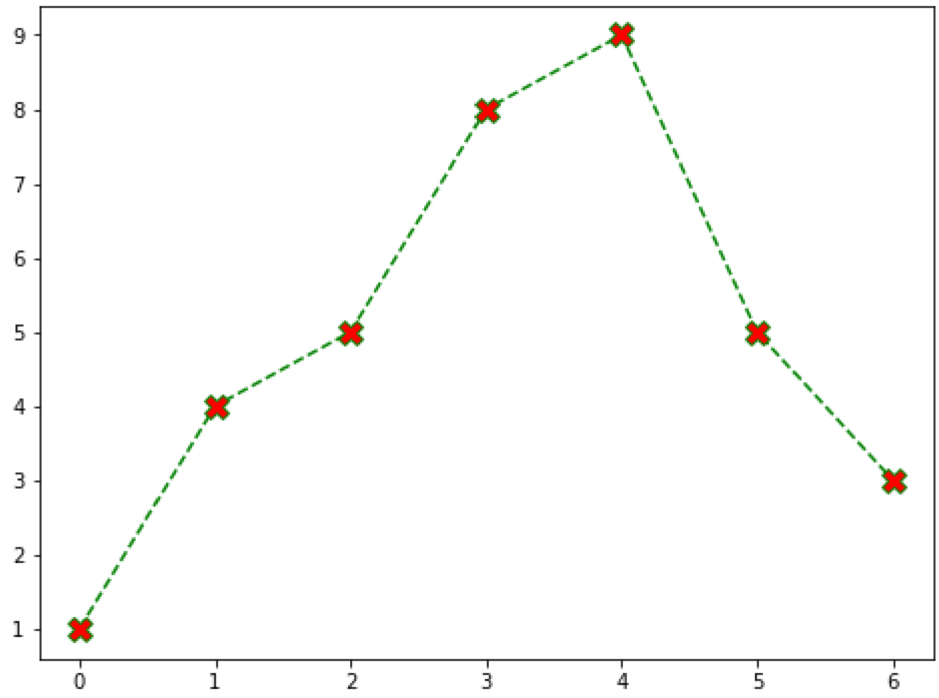
```
plt.figure(figsize=(8,6))
plt.plot(t, y, color='black', linestyle='dashed')
plt.show()
```



```
plt.figure(figsize=(8,6))
plt.plot(t, y, color='black', linestyle='dashed', marker='o',
        markerfacecolor='purple')
plt.show()
```



```
plt.figure(figsize=(8,6))
plt.plot(t, y, color='green', linestyle='dashed', marker='X',
        markerfacecolor='red', markersize=12)
plt.show()
```



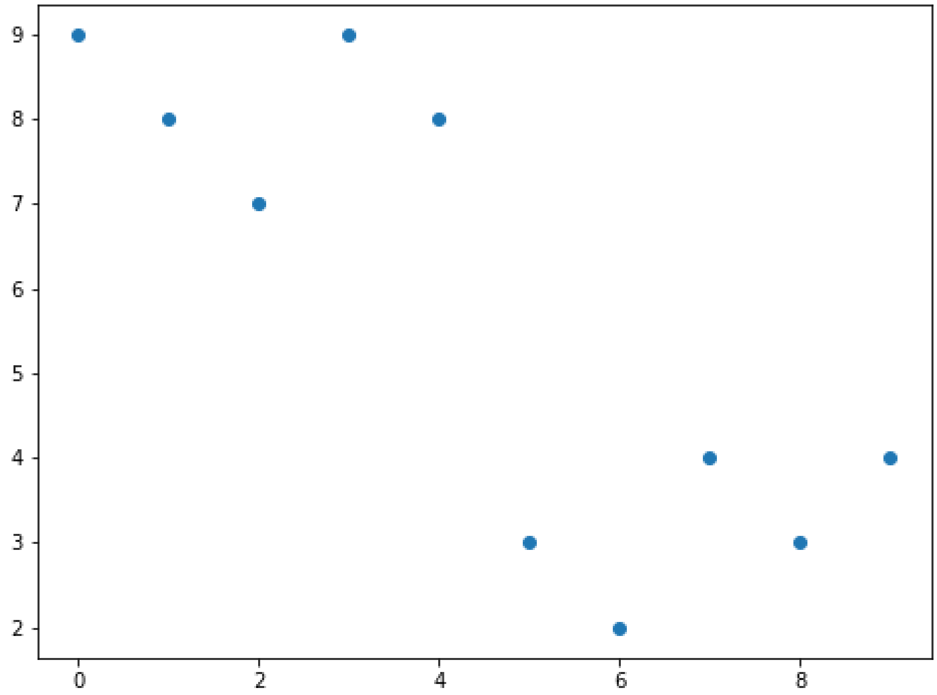
scatter 함수 사용

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

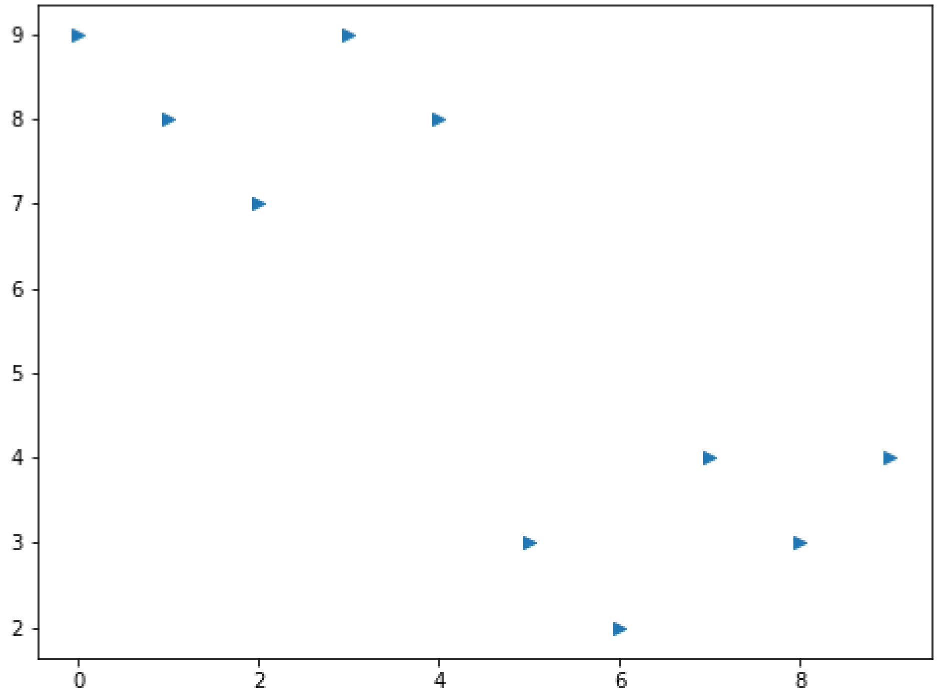
```
t = np.array([0,1,2,3,4,5,6,7,8,9])
```

```
y = np.array([9,8,7,9,8,3,2,4,3,4])
```

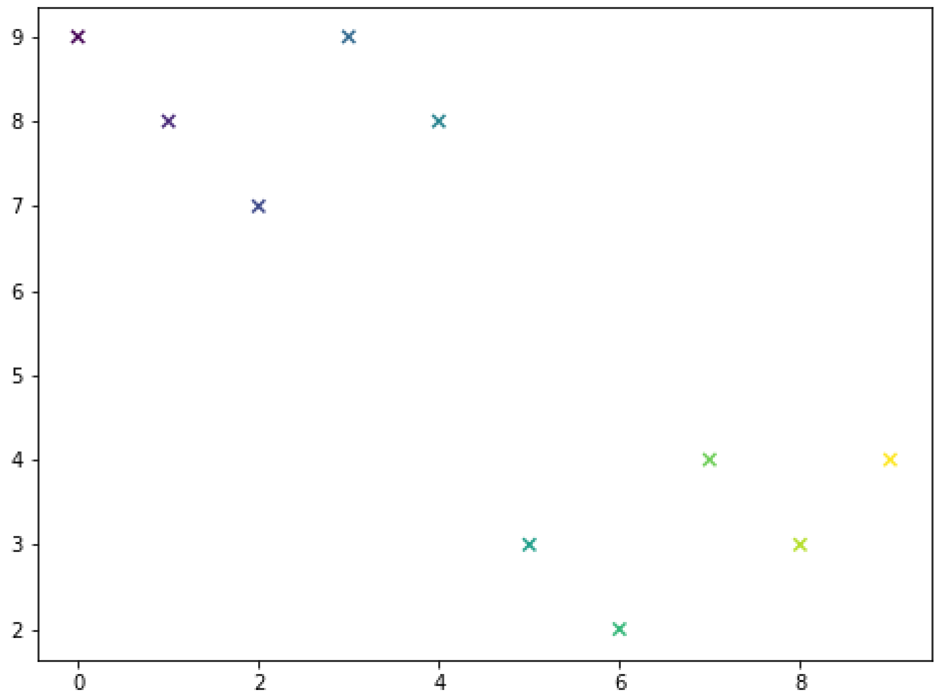
```
plt.figure(figsize=(8,6))
plt.scatter(t,y)
plt.show()
```



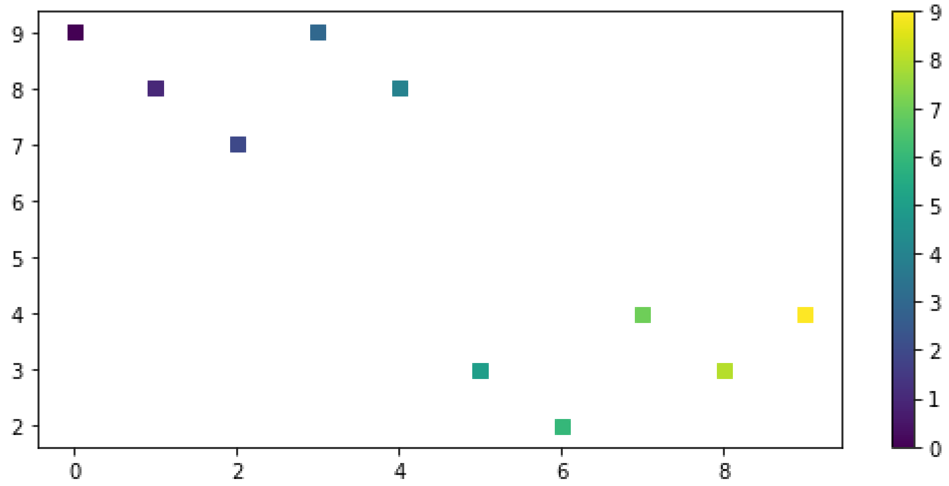
```
plt.figure(figsize=(8,6))
plt.scatter(t,y,marker='>')
plt.show()
```



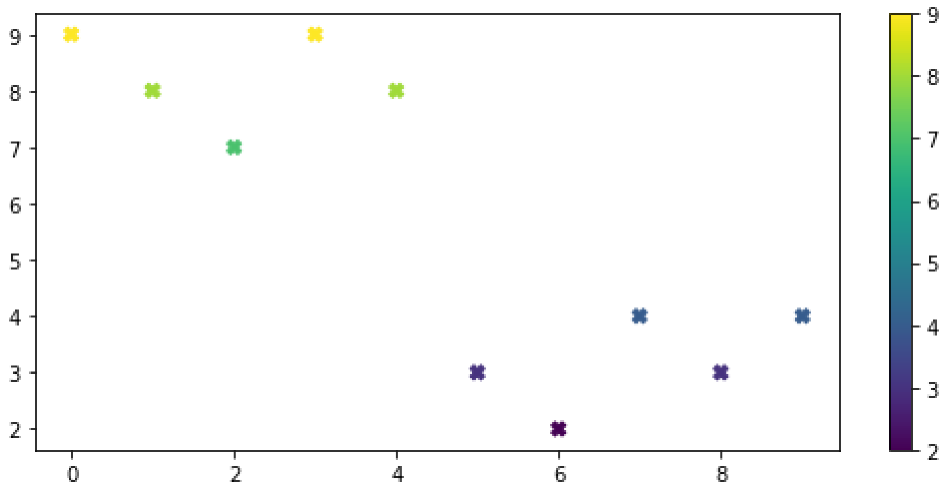
```
colormap = t     # colormap의 기준 t

plt.figure(figsize=(8,6))
plt.scatter(t,y, s=40, c=colormap,marker='x')
plt.show()
```



```
colormap = t     # colormap의 기준 t

plt.figure(figsize=(9,4))
plt.scatter(t,y, s=50, c=colormap, marker=',')
plt.colorbar()
plt.show()
```

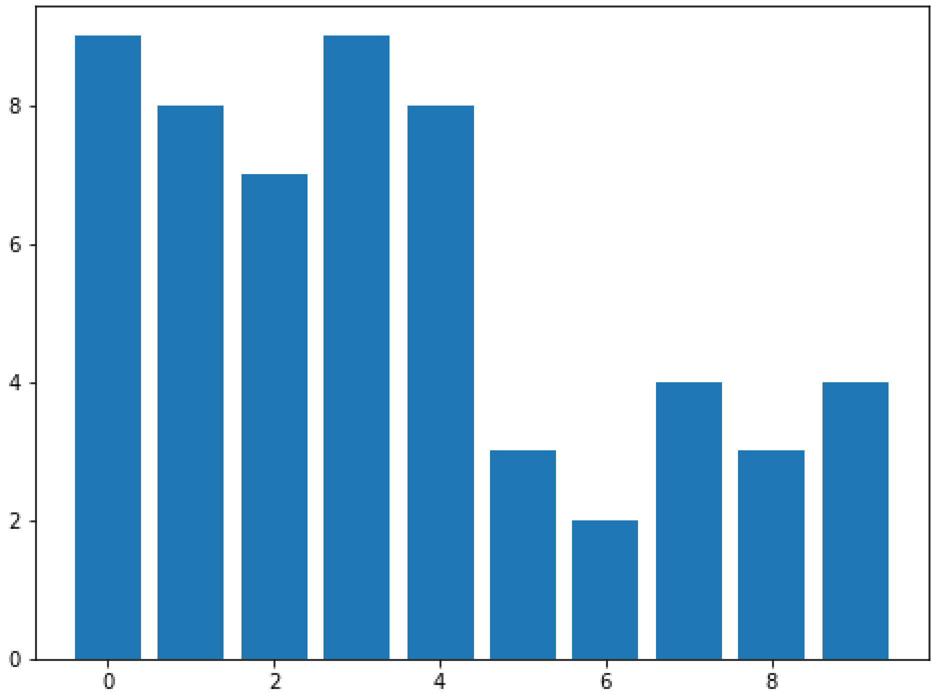


```
colormap = y

plt.figure(figsize=(9,4))
plt.scatter(t,y, s=50, c=colormap, marker='X')
plt.colorbar()
plt.show()

# y가 작아질수록 짙어짐
```



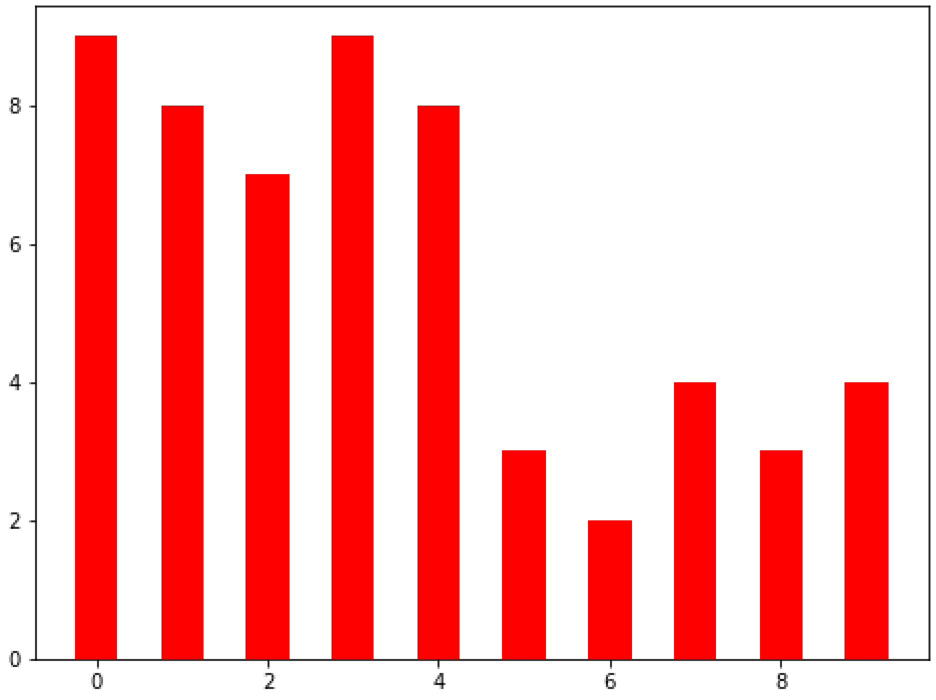
Bar 그래프 작성

```
plt.figure(figsize=(8,6))
plt.bar(t,y)
plt.show()
```



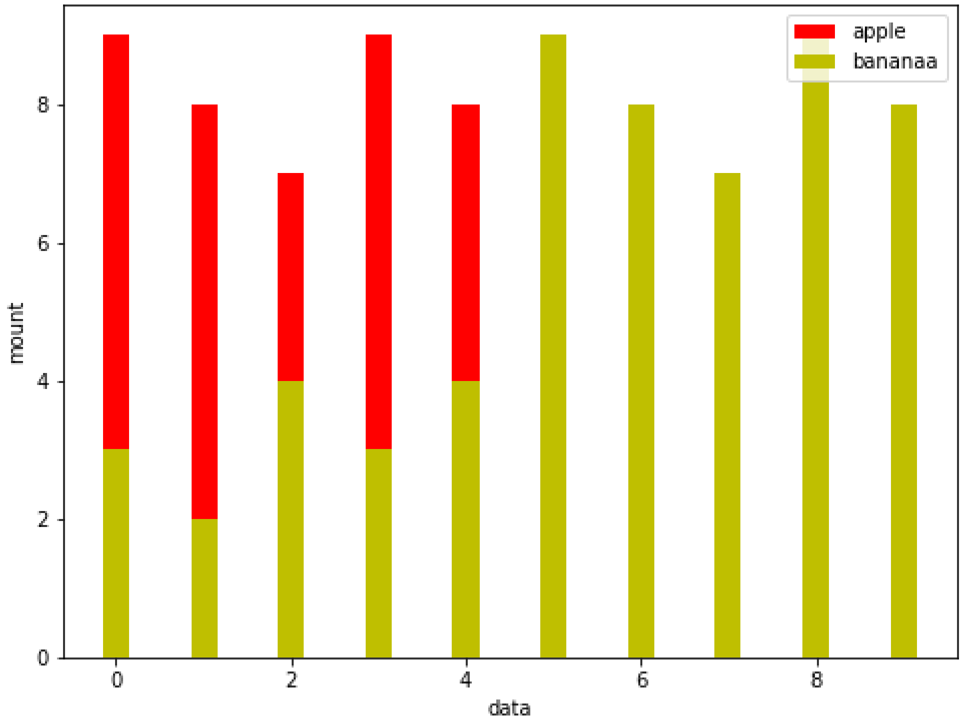
```
?plt.bar
```

```
plt.figure(figsize=(8,6))
plt.bar(t,y, width = 0.5, color='red')
# plt.bar(left, height, width=0.8,bottom=None, hold=None, data=None, **kwargs)
# width = 너비
plt.show()
```

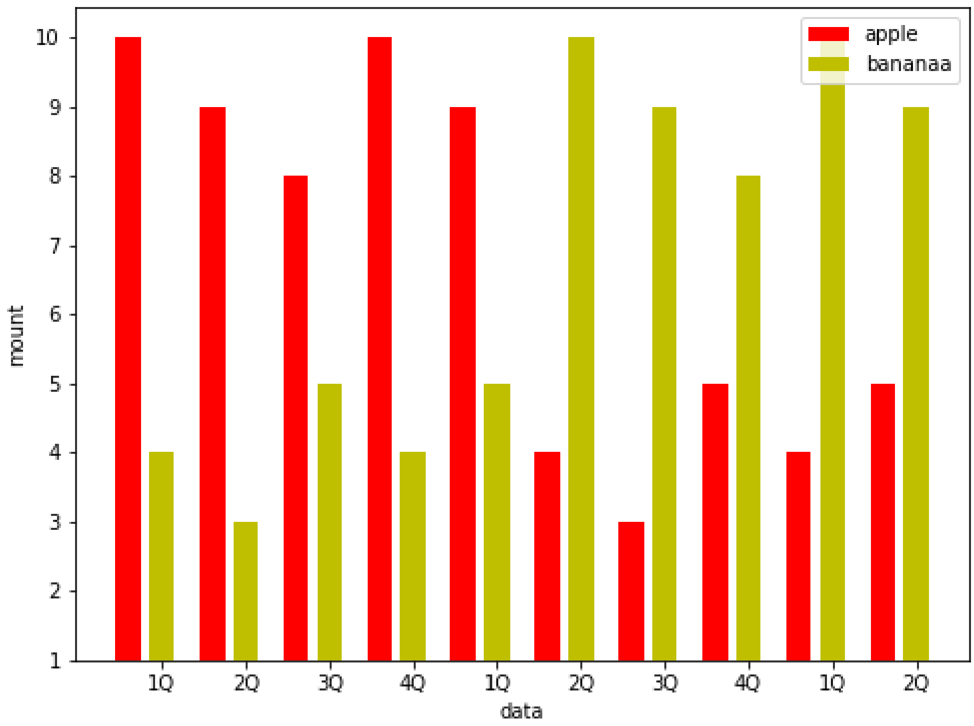


```
y1 = np.array([3,2,4,3,4, 9,8,7,9,8])
```

```
plt.figure(figsize=(8,6))
plt.bar(t, y, color='r', width=0.3, label = 'apple')
plt.bar(t, y1, color='y', width=0.3, label = 'bananaa')
plt.xlabel('data')
plt.ylabel('mount')
plt.legend()
plt.show()
```

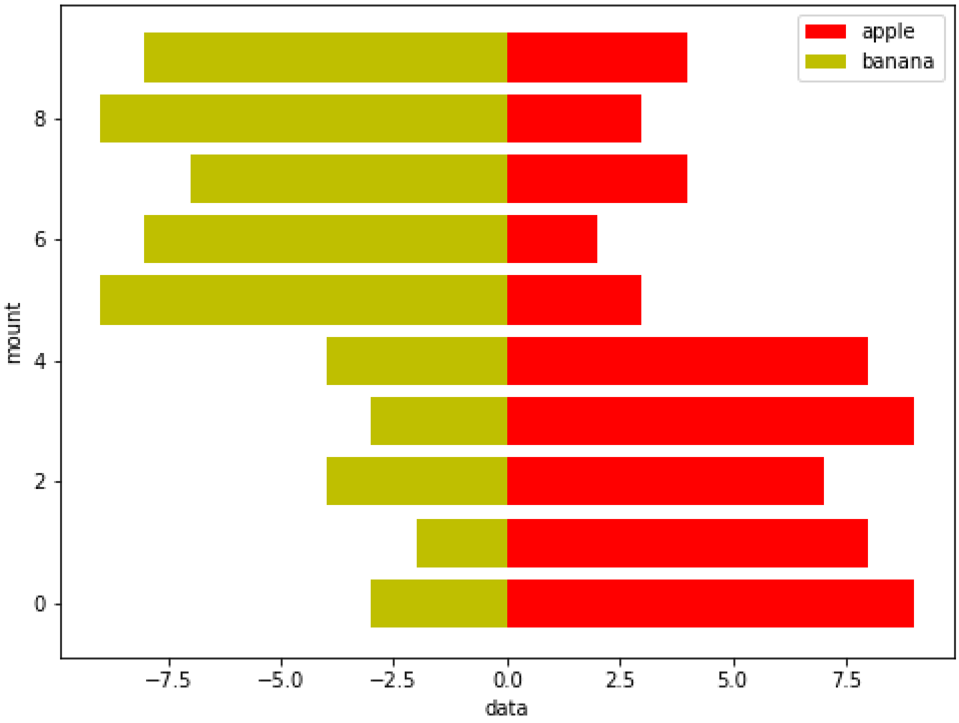


```
plt.figure(figsize=(8,6))
plt.bar(t, y, color='r', width=0.3, label = 'apple')
plt.bar(t+ 0.4, y1, color='y', width=0.3, label = 'bananaa')
plt.xlabel('data')
plt.ylabel('mount')
plt.legend()

# 막대 X축 명칭 변경
# t+ N 으로 막대그룹 위치 변경

plt.yticks(t, (range(1,11)))     # y축도 1단위로 보고 싶을 때 맞춰주기
plt.xticks(t+0.4, ('1Q','2Q','3Q','4Q','1Q','2Q','3Q','4Q','1Q','2Q'))
plt.show()
```

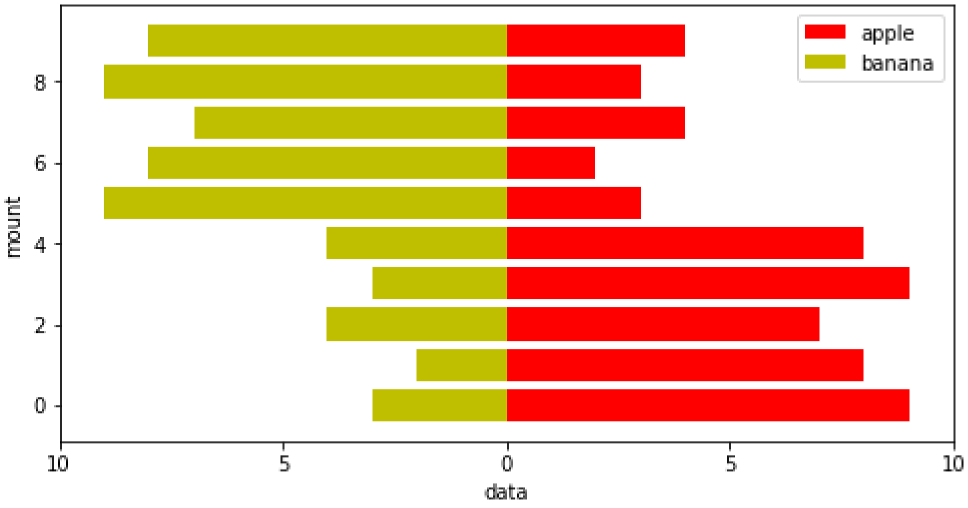


좌우 bar 그래프

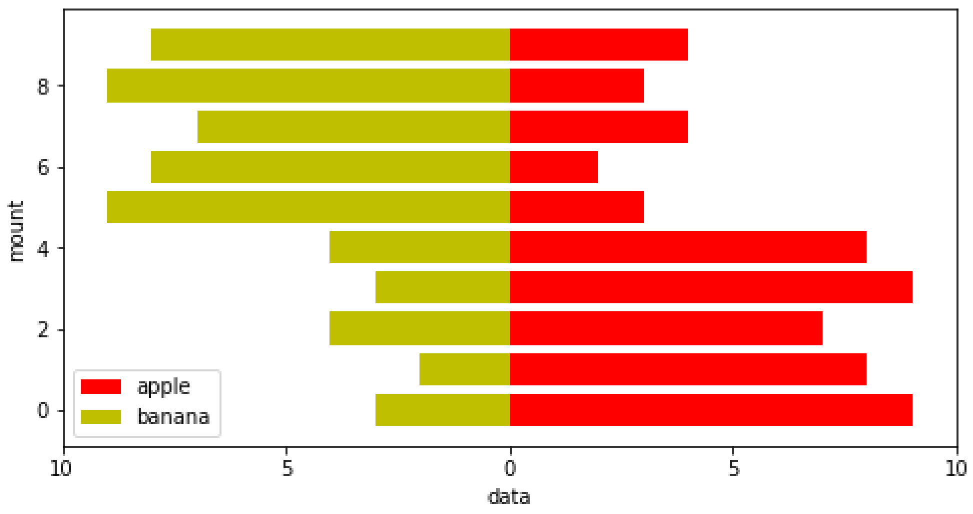
```
plt.figure(figsize=(8,6))
plt.barh(t,y, color='r', label='apple')
plt.barh(t,-y1,color='y',label='banana')  # -를 취해 반대쪽으로 뻗어가게 해줌
plt.xlabel('data')
plt.ylabel('mount')
plt.legend()
plt.show()
```



```
plt.figure(figsize=(8,4))
plt.barh(t,y, color='r', label='apple')
plt.barh(t,-y1,color='y',label='banana')  # -를 취해 반대쪽으로 뻗어가게 해줌
plt.xlabel('data')
plt.ylabel('mount')
plt.legend()
plt.xticks([-10,-5,0,5,10],('10','5','0','5','10'))
plt.show()
```



```
plt.figure(figsize=(8,4))
plt.barh(t,y, color='r', label='apple')
plt.barh(t,-y1,color='y',label='banana')  # -를 취해 반대쪽으로 뻗어가게 해줌
plt.xlabel('data')
plt.ylabel('mount')
plt.legend(loc=3)
plt.xticks([-10,-5,0,5,10],('10','5','0','5','10'))
plt.show()
```

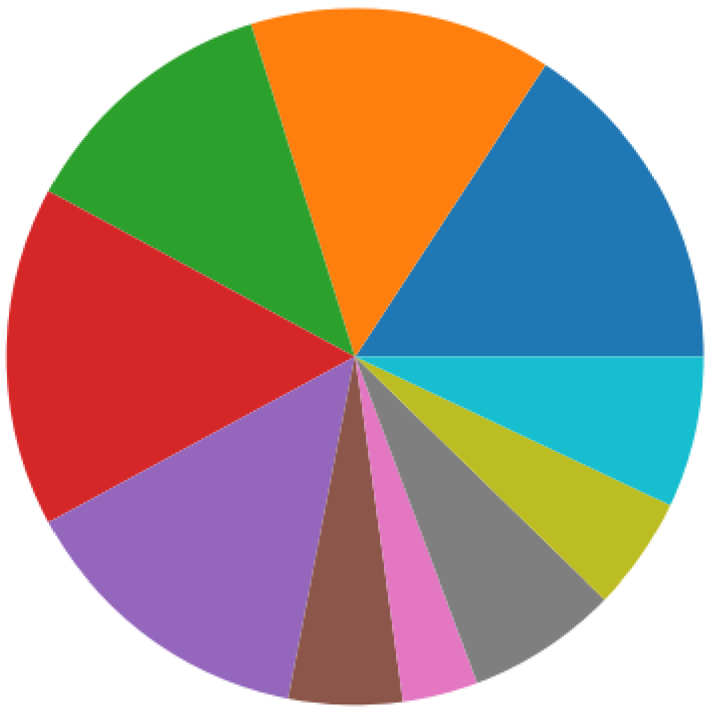


pie 그래프 작성

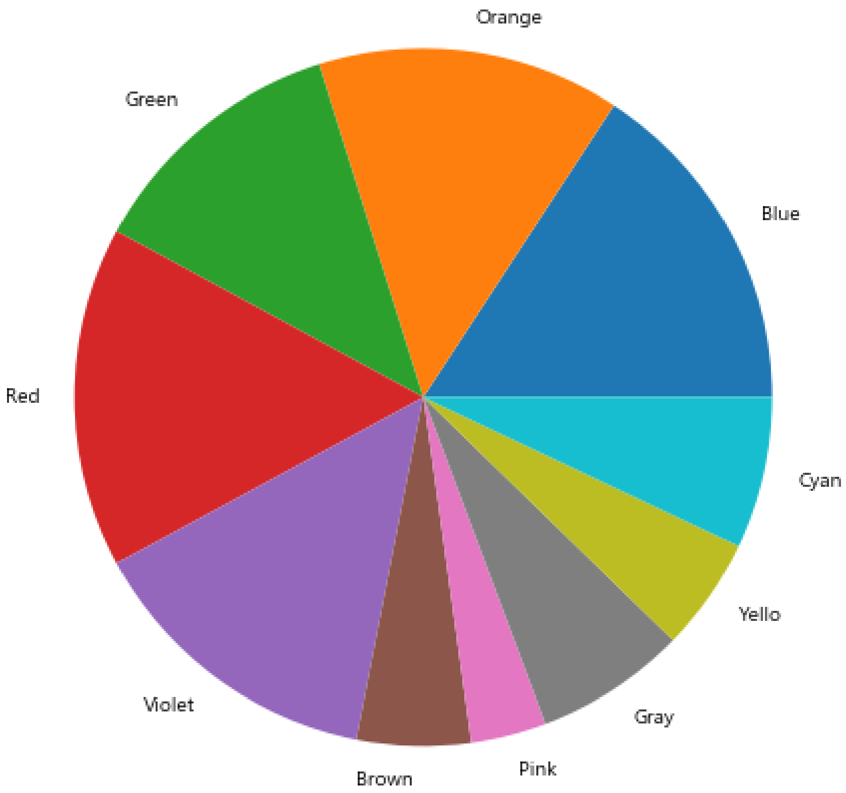
```
y
```

```
array([9, 8, 7, 9, 8, 3, 2, 4, 3, 4])
```

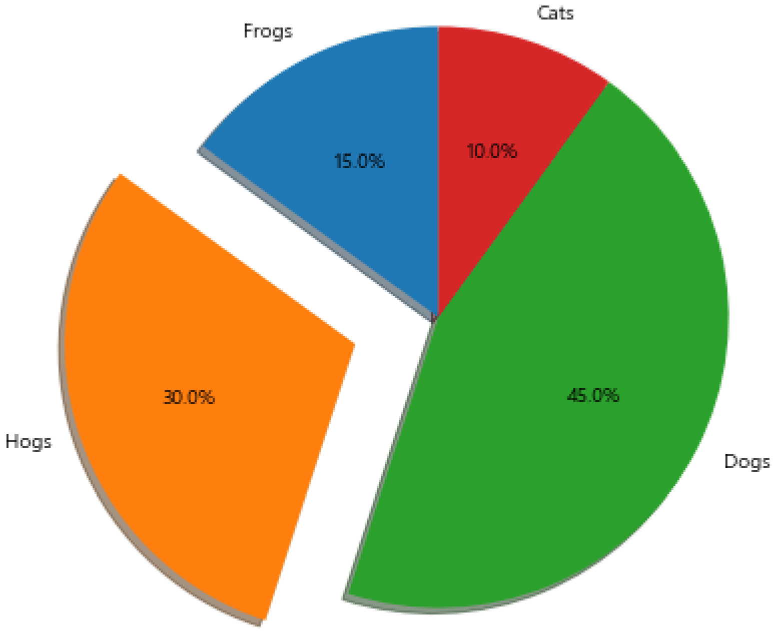
```
plt.figure(figsize=(8,8))
plt.pie(y)
plt.show()
```



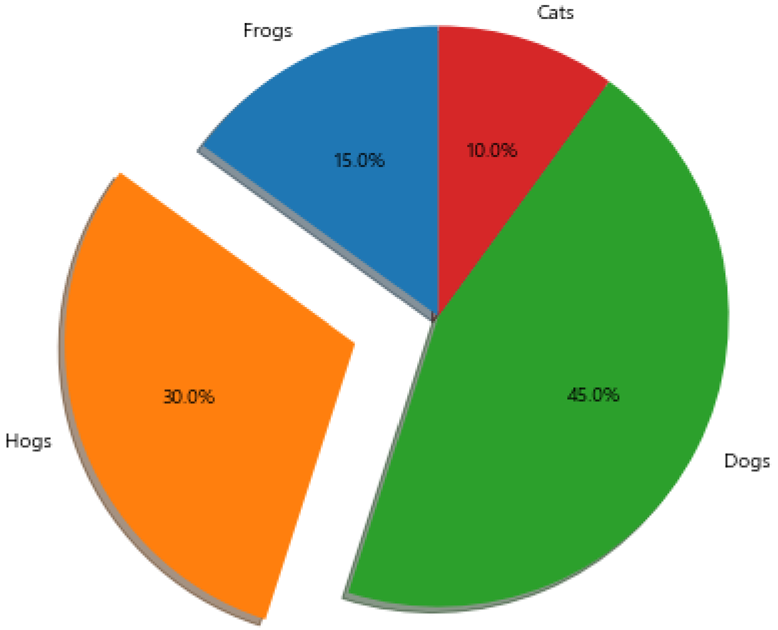
```
label = ['Blue', 'Orange', 'Green','Red','Violet',
         'Brown','Pink', 'Gray','Yello', 'Cyan']

plt.figure(figsize=(8,8))
plt.pie(y, labels=label)
plt.show()
```



```
labels = 'Frogs', 'Hogs','Dogs', 'Cats'
sizes=[15,30,45,10]
explode = (0,0.3,0,0)    #몇 번째를 얼마나 띄워 놓을지.

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
       shadow=True,startangle=90)
ax1.axis('equal')       #원이 타원이 되지 않도록 가로세로 비율을 똑같이

plt.show()
```

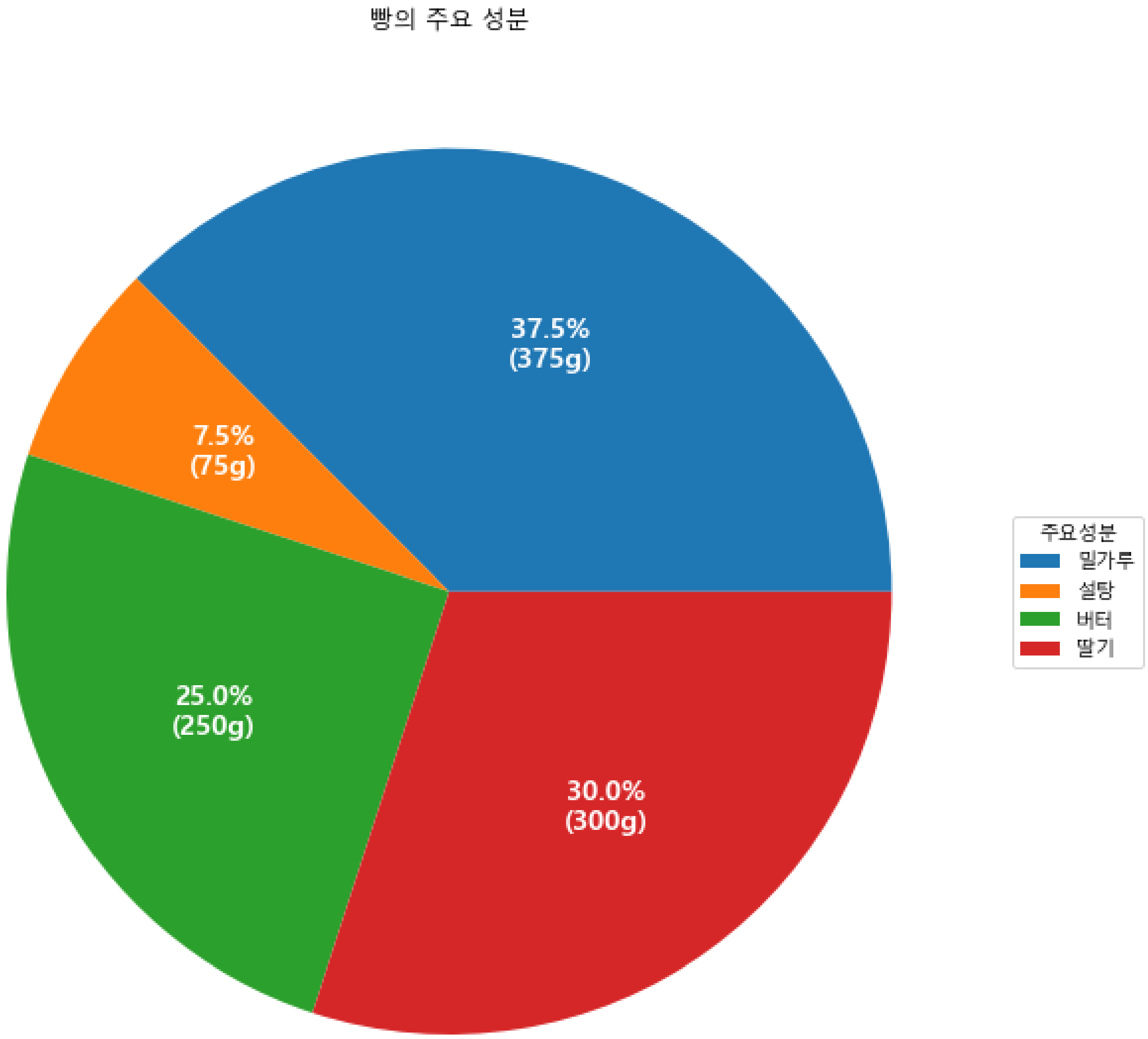


```
sizes=[15,30,45,10]

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=(0,0.3,0,0) , labels=('Frogs', 'Hogs','Dogs', 'Cats'), autopct='%1.1f%%',
       shadow=True,startangle=90)
ax1.axis('equal')       #원이 타원이 되지 않도록 가로세로 비율을 똑같이

plt.show()
```



```
fig, ax = plt.subplots(figsize=(10,12), subplot_kw=dict(aspect="equal"))

recipe = ["375 g 밀가루",
         "75 g 설탕",
         "250 g 버터",
         "300 g 딸기"]

data = [float(x.split()[0]) for x in recipe]   # recipe의 맨 앞글자
ingredients = [x.split()[-1] for x in recipe]  # recipe의 뒷글자
```

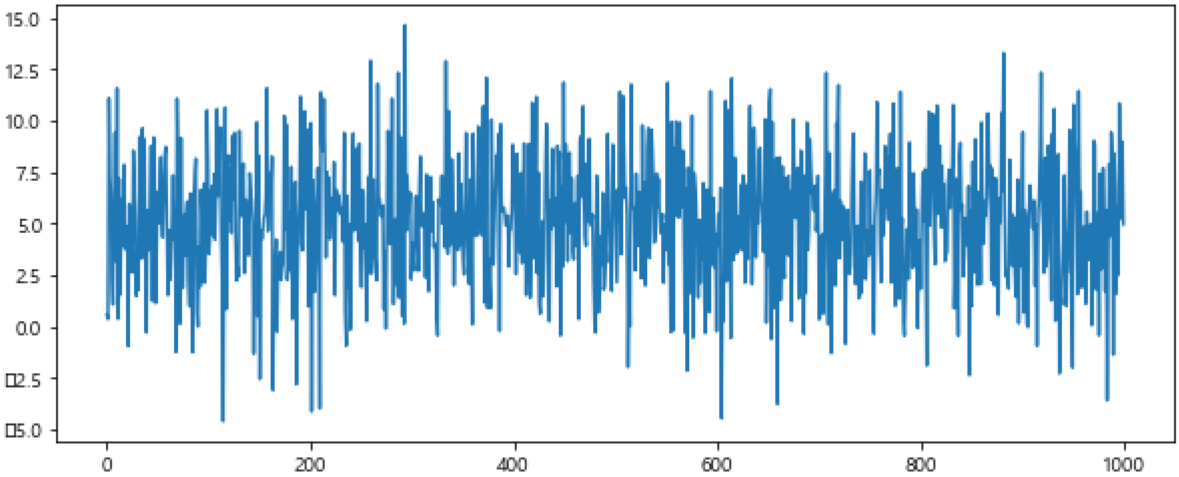
```
def func(pct, allvals) :
    absolute = int(pct/100.*np.sum(allvals))
    return "{:.1f}%₩n({:d}g)".format(pct, absolute)

# {:.1f} = 소숫점 한자리수까지 -> {:.2f} = 소숫점 두자리수까지
#"{:.1f}%   = 소수점 한자리수까지%
#₩n 한 칸 띄우기
# ({:d}g)   = 수치 g

wedges, texts, autotexts = ax.pie(data, autopct=lambda pct : func(pct,data),
                                 textprops = dict(color='w'))

ax.legend(wedges, ingredients,
         title="주요성분",
         loc="center left",
         bbox_to_anchor=(1,0,0.5,1))

plt.setp(autotexts, size=14, weight="bold")

ax.set_title("빵의 주요 성분")

plt.show()
```



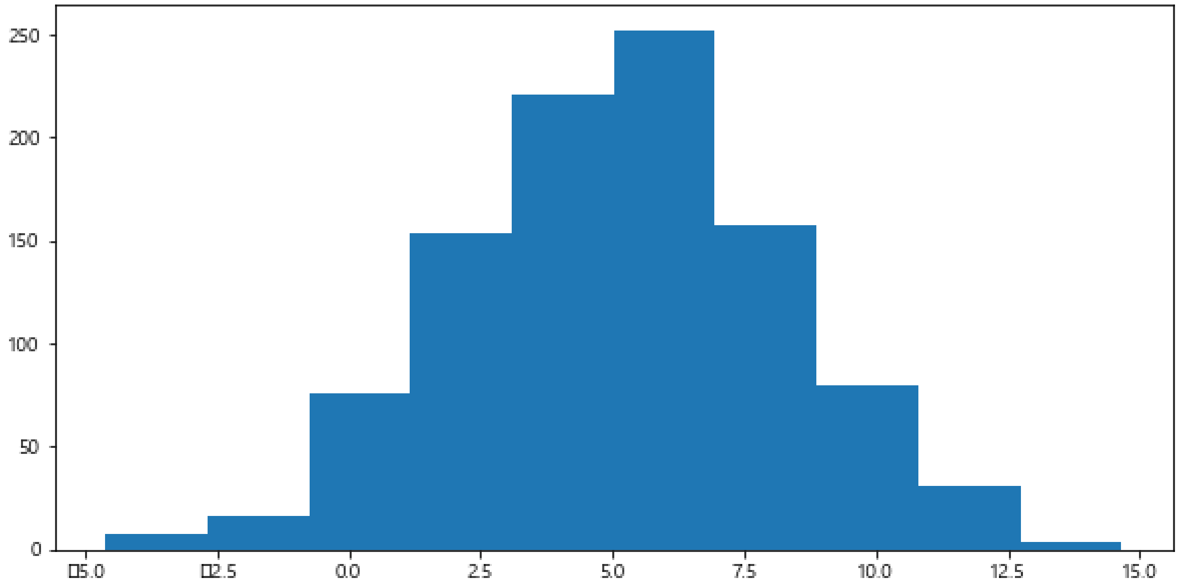
히스토그램 작성

```
data = np.random.normal(5, 3, 1000)
```

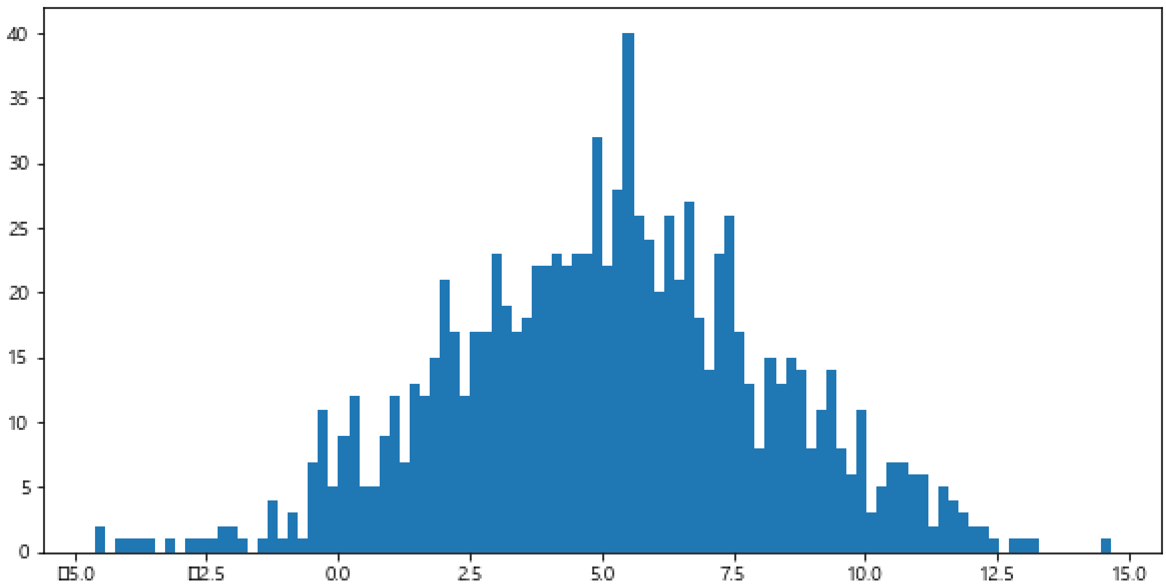
```
plt.figure(figsize=(10,4))
plt.plot(data)
plt.show()
```



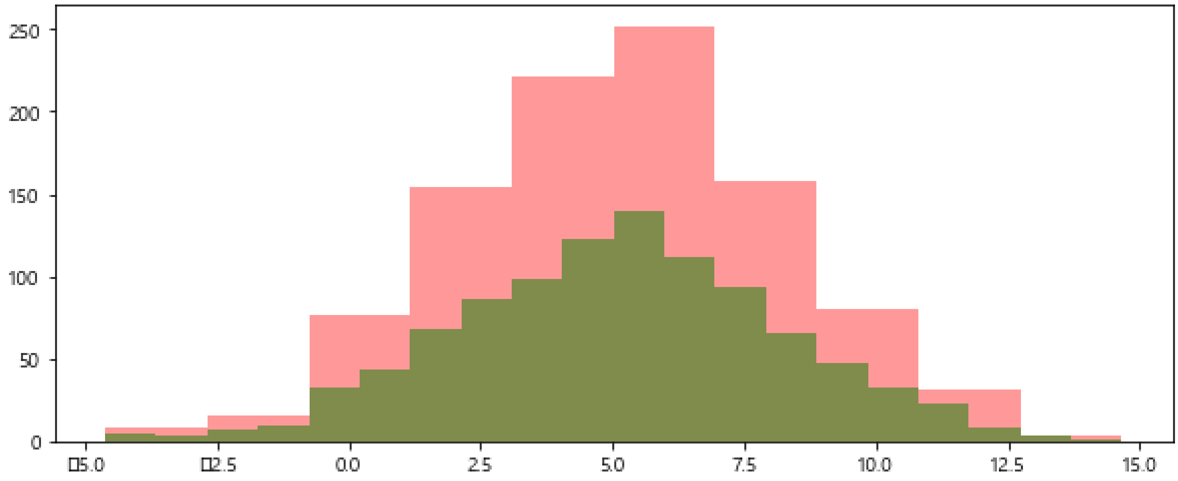
```
plt.figure(figsize=(10,5))
plt.hist(data)
plt.show()
```



```
plt.figure(figsize=(10,5))
plt.hist(data, bins=100)
plt.show()
```

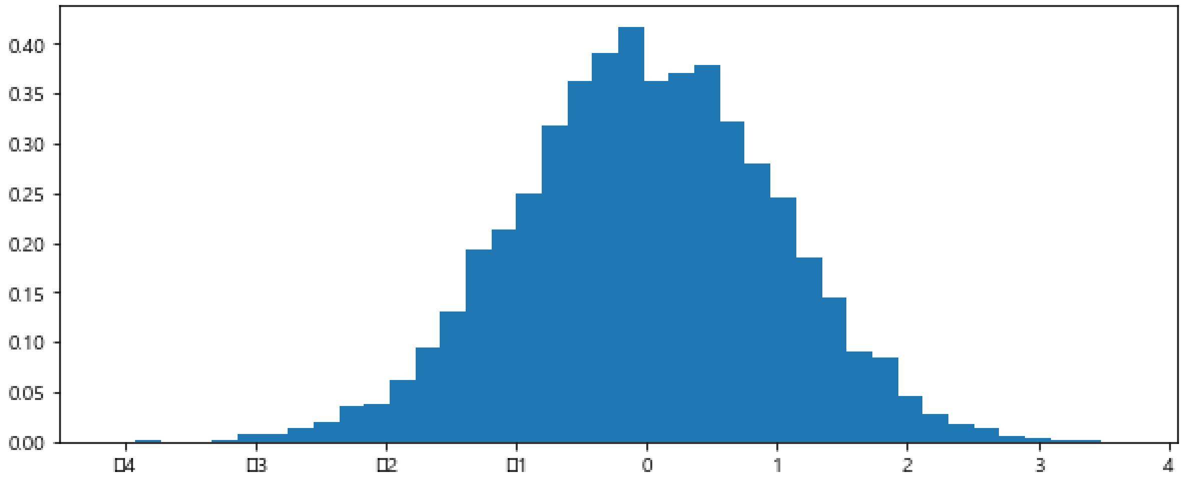


```
plt.figure(figsize=(10,4))
plt.hist(data, bins=10, facecolor='red',
        alpha=0.4, histtype='stepfilled')
plt.hist(data, bins=20, facecolor='green',
        alpha=0.5, histtype='stepfilled')

#alpha = 투명도
plt.show()
```



```
x= np.random.randn(10000)
```

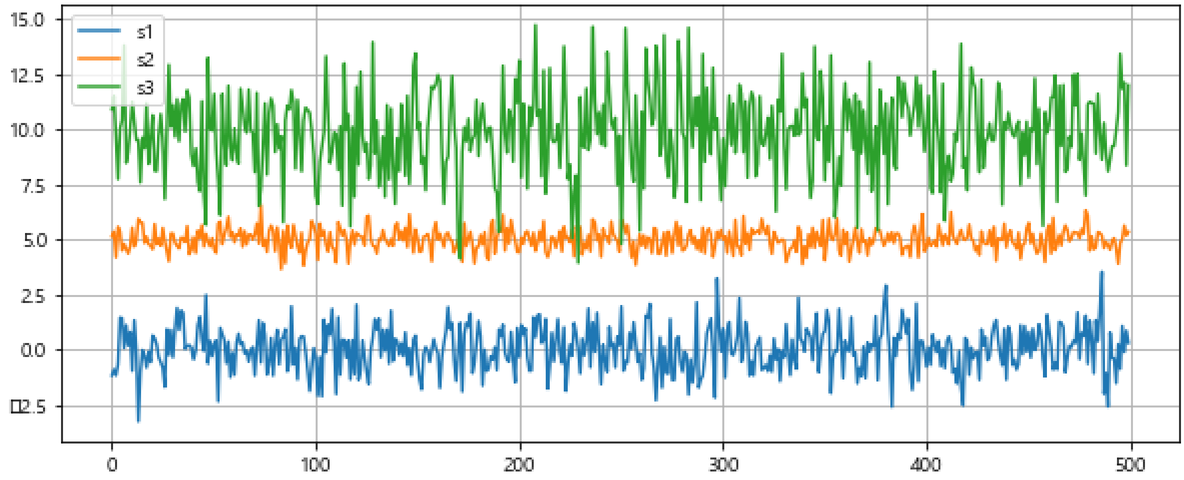
```
plt.figure(figsize=(10,4))
plt.hist(x, density=100, bins=40)
plt.show()
```



box 그래프 작성

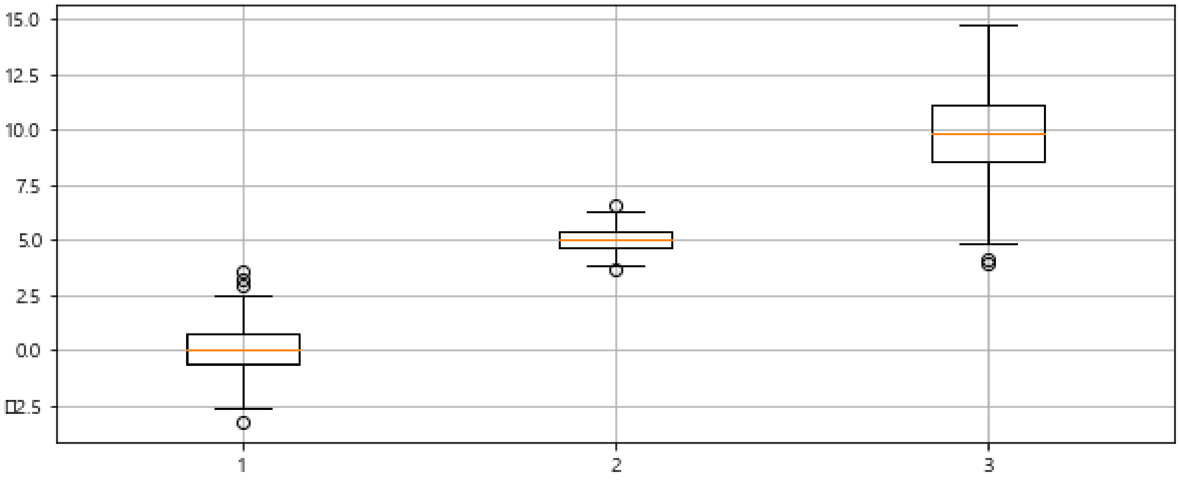
```
# scale : 분산
# loc : 평균값

s1 = np.random.normal(loc=0,  scale=1.0, size=500)
# loc : Mean ("centre") of the distribution.
s2 = np.random.normal(loc=5,  scale=0.5, size=500)
# scale : Standard deviation (spread or "width") of the distribution.
s3 = np.random.normal(loc=10, scale=2.0, size=500)
# size : Output shape
```

```
plt.figure(figsize=(10,4))
plt.plot(s1, label='s1')
plt.plot(s2, label='s2')
plt.plot(s3, label='s3')
plt.grid()
plt.legend()
plt.show()
```



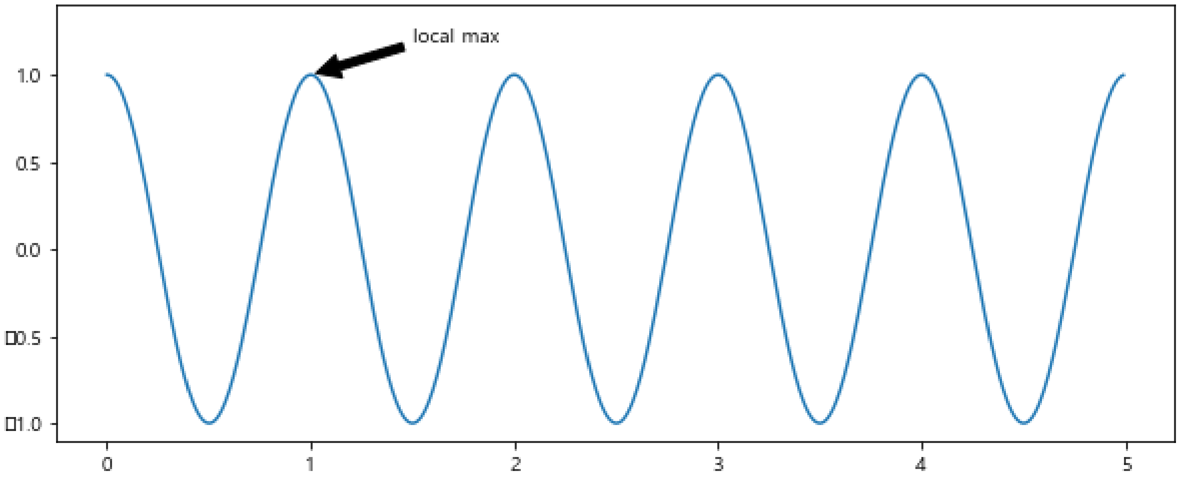
```
plt.figure(figsize=(10,4))
plt.boxplot((s1, s2, s3))
plt.grid()
plt.show()
```

```
C:₩Python₩Anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:211: RuntimeWarning:
  font.set_text(s, 0.0, flags=flags)
C:₩Python₩Anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:180: RuntimeWarning:
  font.set_text(s, 0, flags=flags)
```



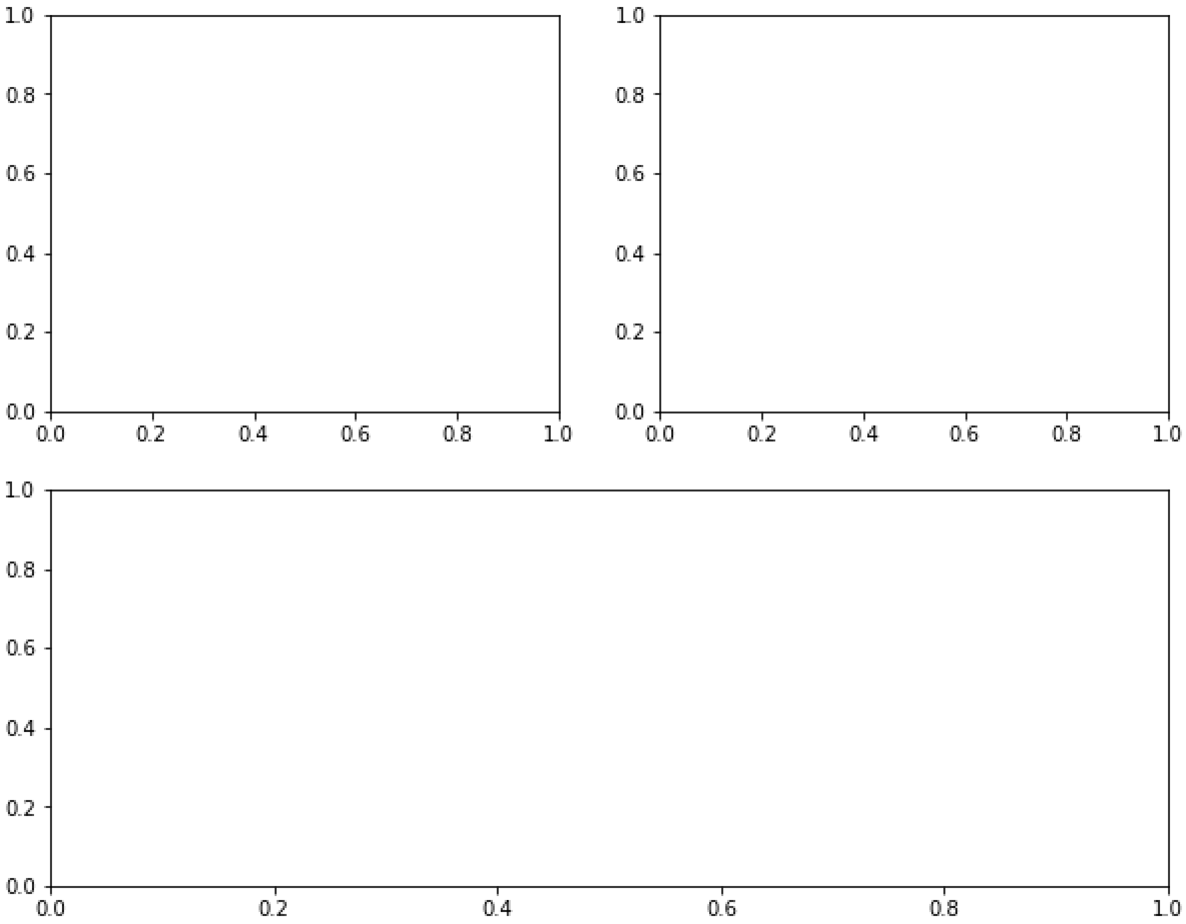
annotation

```
t = np.arange(0,5,0.01)
y = np.cos(2*np.pi*t)

plt.figure(figsize=(10,4))
plt.plot(t,y)
plt.annotate('local max', xy=(1,1), xytext=(1.5,1.2),
            arrowprops=dict(facecolor='black',shrink=0.05))
plt.ylim(-1.1,1.4)
plt.show()
```



subplot 적용

```
plt.figure(figsize=(10,8))
plt.subplot(221)
plt.subplot(222)
plt.subplot(212)
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0