빅데이터 실습

9주차 2차시

데이터 시각화 - matplotlib 실습 [1]

데이터 시각화
matplotlib 실습 [1]

학습개요

1/ 단일 Axes 그리기 〈 라인 차트, 바 차트 〉

2/ 2개 이상의 Axes를 하나의 Figure에 그리기

01

단일 Axes 그리기
< 라인 차트, 바 차트 >

실습 준비하기

◎ import

- ✔ pandas, numpy

▶ Run | Markdown

In [1]:
```
import pandas as pd
import numpy as np
from pandas import Series, DataFrame
```

데이터 시각화 (Matplotlib) 활용와 왕좌의 게임 데이터 분석

1. matplotlib 간단 실습

1.1 하나의 그래프 그리기

1.1.1 line graph 그리기

In [ ]:

실습 준비하기

◎ import

✔ matplotlib은 별로의 라이브러리이므로 import 수행

데이터 시각화 (Matplotlib) 활용와 왕좌의 게임 데이터 분석

1. matplotlib 간단 실습

In [2]:
```
import matplotlib
import matplotlib.pyplot as plt
```

In [ ]:

주의하기

✔ matplotlib만 import할 경우 하위 모듈들은 import 되지 않음

In [ ]:

In [ ]:

matplotlib 하나의 그래프 그리기

◎ line graph 그리기

Run Code

1. matplotlib 간단 실습

In [3]:
```
import matplotlib
import matplotlib.pyplot as plt
```

1.1 하나의 그래프 그리기

1.1.1 line graph 그리기

In [4]:
```
sr = Series([10,20,30], index = ['A','B','C'])
sr
```

Out[4]:
```
A    10
B    20
C    30
dtype: int64
```

In [ ]:

matplotlib 하나의 그래프 그리기

◎ line graph 그리기

✔ plt.plot( )

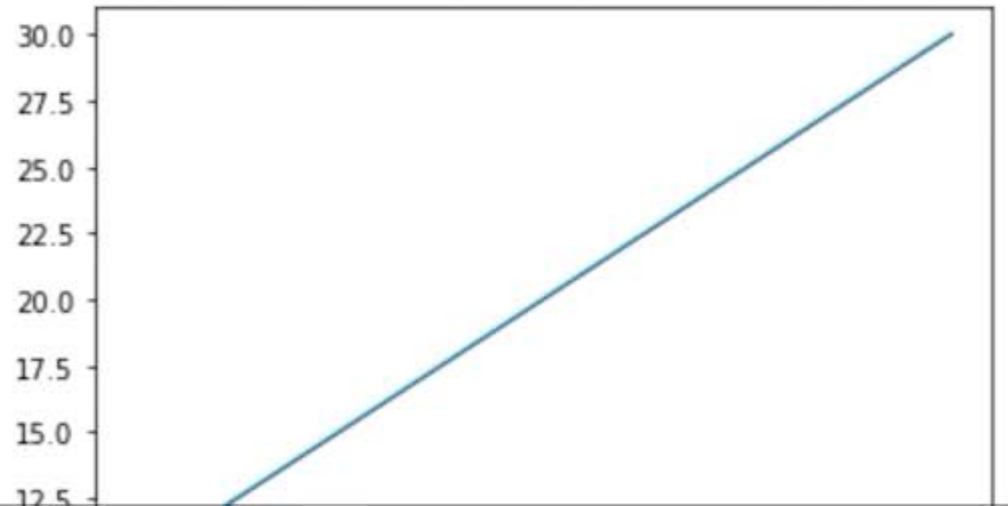
Run Markdown

```
In [ ]: sr = Series([10,20,30], index = ['A','B','C'])
        sr
```

```
Out[4]: A    10
        B    20
        C    30
        dtype: int64
```

```
In [6]: plt.plot(sr)
```

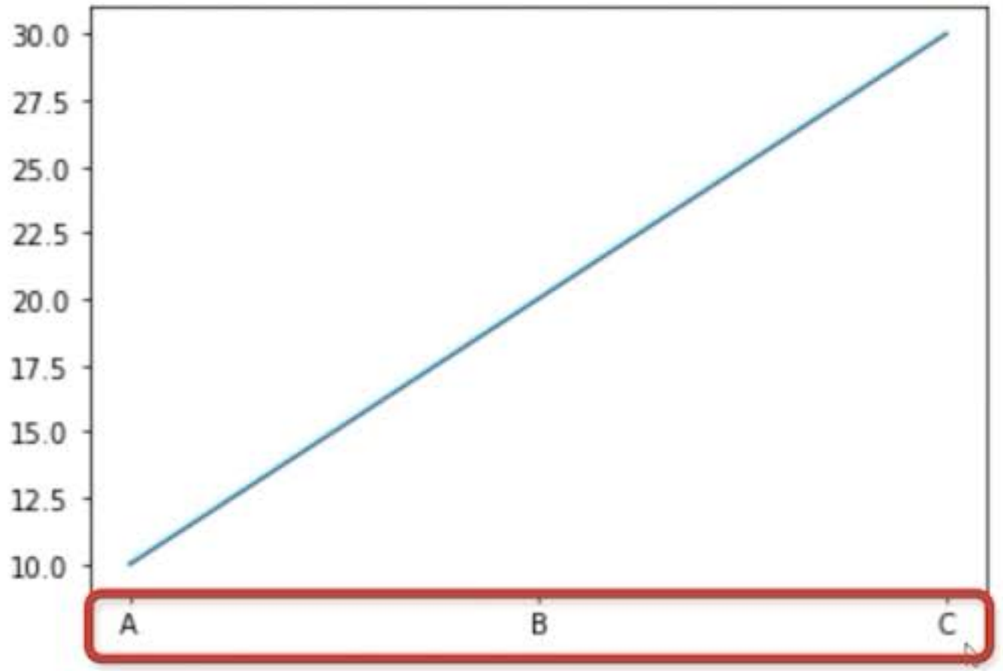
```
Out[6]: [<matplotlib.lines.Line2D at 0x20157ba00d0>]
```



matplotlib 하나의 그래프 그리기

◎ line graph 그리기

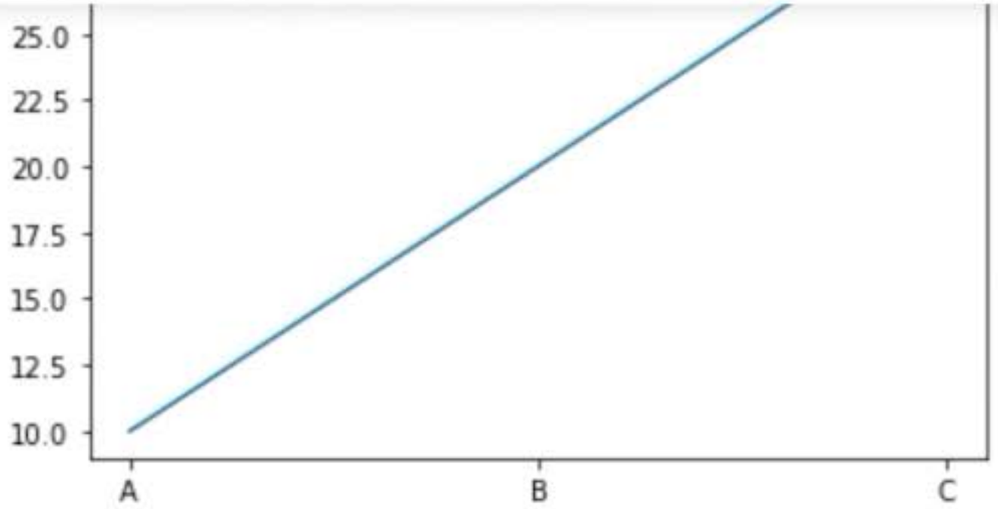
- ✔ plt.plot( )
- ✔ 새로운 figure와 axes 객체를 생성한 후, 생성된 객체에 그래프를 그림



1.1.2 bar graph 그리기

matplotlib 하나의 그래프 그리기

◎ bar graph 그리기



1.1.2 bar graph 그리기

In [ ]:
```
sr = Series([1,2,4,8,16], index = ['A','B','C','D','E'])
```

In [ ]:

In [ ]:

In [ ]:

matplotlib 하나의 그래프 그리기

◎ bar graph 그리기

✔ plt.bar( )

Run Code

```
A     1
B     2
C     4
D     8
E    16
dtype: int64
```

In [10]:
```
plt.bar(sr)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-10-d28f2f743f80> in <module>
----> 1 plt.bar(sr)

TypeError: bar() missing 1 required positional argument: 'height'
```

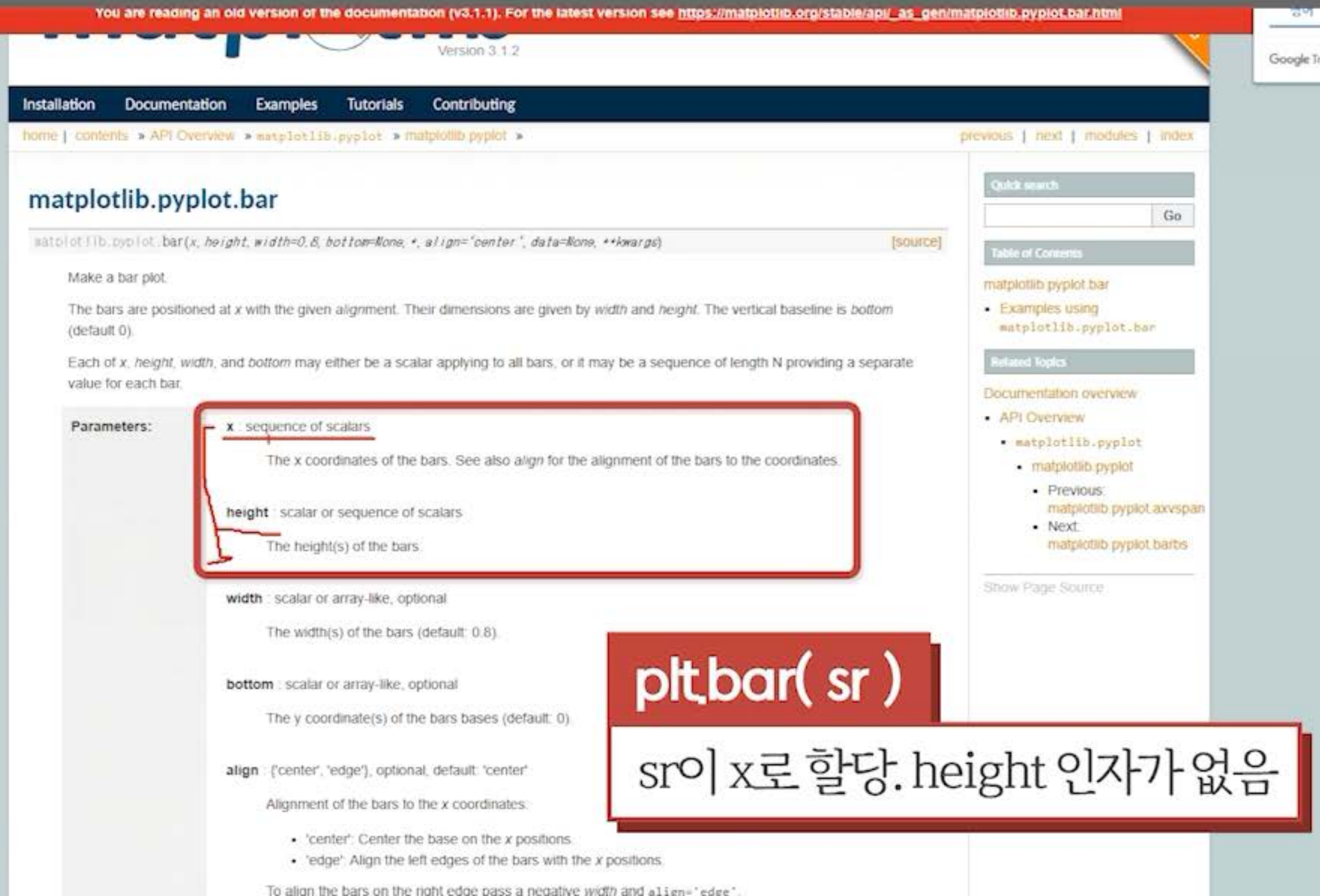
In [ ]:

In [ ]:

1 1 3 두 개의 그래프를 하나로 그리기

matplotlib 하나의 그래프 그리기

◎ bar graph 그리기



plt.bar(sr)

sr이 x로 할당. height 인자가 없음

matplotlib 하나의 그래프 그리기

◎ bar graph 그리기

✔ plt.bar( )

Run Code

```
A     1
B     2
C     4
D     8
E    16
dtype: int64
```

In [10]:
```
plt.bar(sr.index, sr.values)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-10-d28f2f743f80> in <module>
----> 1 plt.bar(sr)

TypeError: bar() missing 1 required positional argument: 'height'
```

In [ ]:

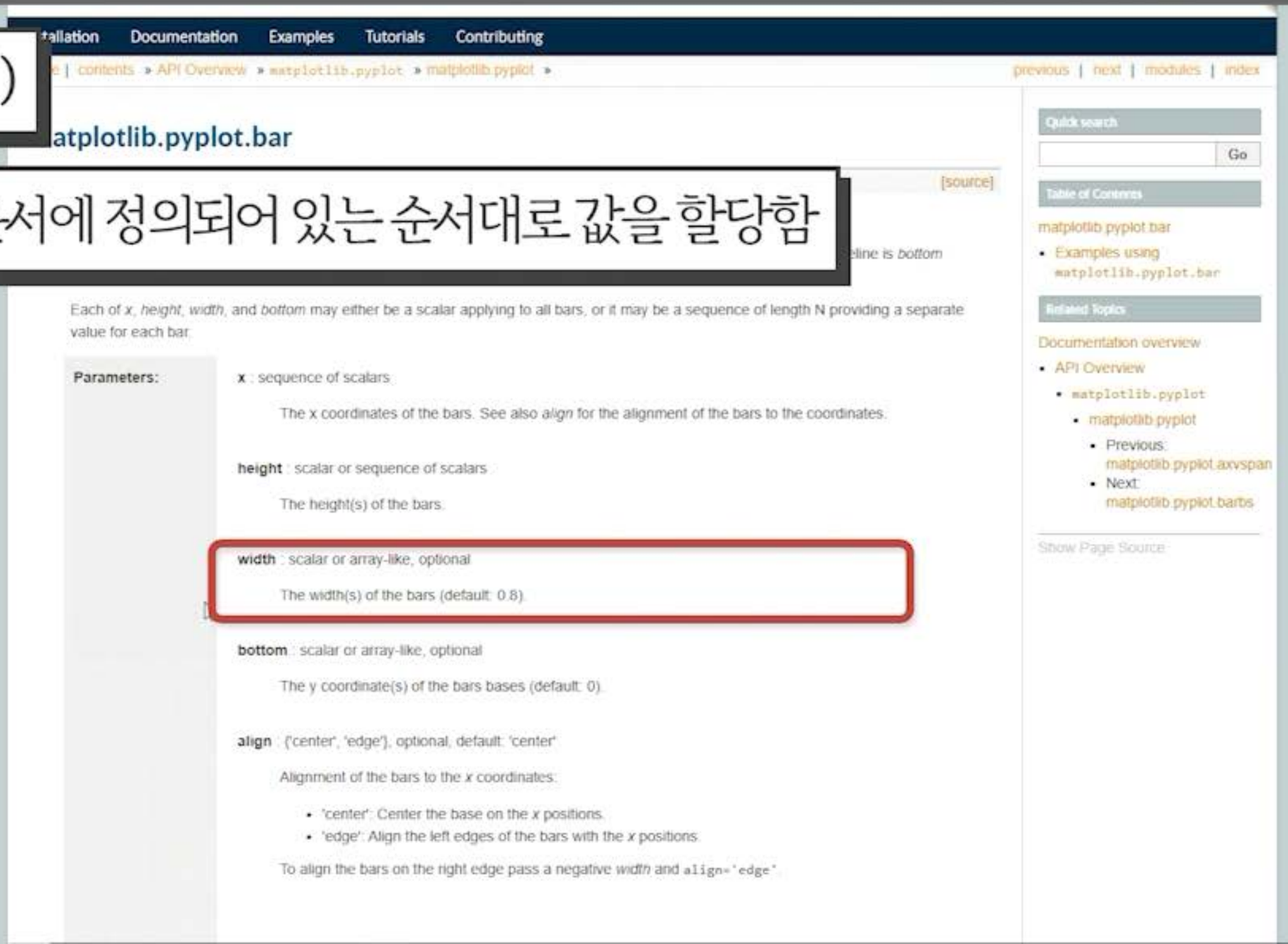
In [ ]:

1.1.3 두 개의 그래프를 하나로 그리기

matplotlib 하나의 그래프 그리기

◎ bar graph 그리기

- ✔ plt.bar( )
- ✔ api 문서에 정의되어 있는 순서대로 값을 할당함



tallation Documentation Examples Tutorials Contributing

e | contents » API Overview » matplotlib.pyplot » matplotlib.pyplot » previous | next | modules | index

atplotlib.pyplot.bar

[source]

eline is *bottom*

Each of *x*, *height*, *width*, and *bottom* may either be a scalar applying to all bars, or it may be a sequence of length N providing a separate value for each bar.

Parameters:

x : sequence of scalars

The x coordinates of the bars. See also *align* for the alignment of the bars to the coordinates.

height : scalar or sequence of scalars

The height(s) of the bars.

width : scalar or array-like, optional

The width(s) of the bars (default: 0.8).

bottom : scalar or array-like, optional

The y coordinate(s) of the bars bases (default: 0).

align : {'center', 'edge'}, optional, default: 'center'

Alignment of the bars to the *x* coordinates:

- 'center': Center the base on the *x* positions.
- 'edge': Align the left edges of the bars with the *x* positions.

To align the bars on the right edge pass a negative *width* and `align='edge'`.

Quick search

Go

Table of Contents

matplotlib.pyplot.bar

- Examples using `matplotlib.pyplot.bar`

Related Topics

Documentation overview

- API Overview
  - `matplotlib.pyplot`
    - matplotlib.pyplot
      - Previous: matplotlib.pyplot.axvspan
      - Next: matplotlib.pyplot.barbs

Show Page Source

matplotlib 하나의 그래프 그리기

◎ bar graph 그리기

✔ plt.bar( )

```
D     8
E    16
dtype: int64
```

```
In [12]: plt.bar(x = sr.index, height = sr.values)
```

```
Out[12]: <BarContainer object of 5 artists>
```

Tip.

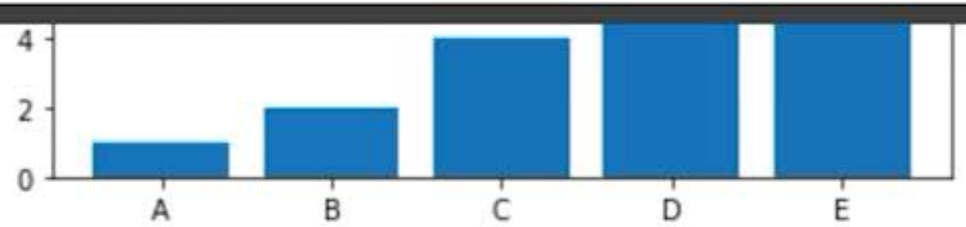
✔ pandas의 plot( )은 그래프 종류 상관 없이
일관된 인터페이스를 제공하여 사용하기 더 편함

인자명을 명시하면, 순서 상관 없이 해당 인자의 값으로 할당할 수 있습니다.

matplotlib 하나의 그래프 그리기

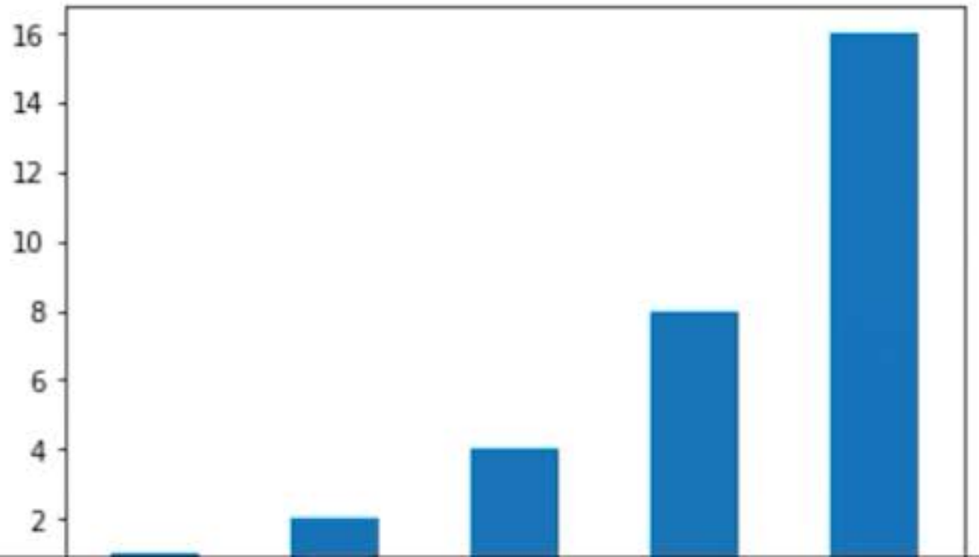
◎ bar graph 그리기

✔ plot(kind = 'bar')를 사용하면 bar graph를 동일하게 그릴 수 있음



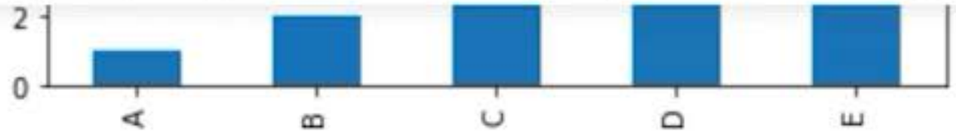

```
In [14]: sr.plot(kind = 'bar')
```

```
Out[14]: <AxesSubplot:>
```



matplotlib 두 개의 그래프를 하나로 그리기

◎ subplots( ) : 새로운 figure, axes를 생성하여 리턴



1.1.3 두 개의 그래프를 하나로 그리기

- plt.subplots() 함수는 새로운 figure와 axes(subplots)를 생성해 준다.

In [15]:
```
fig, axes = plt.subplots()
```

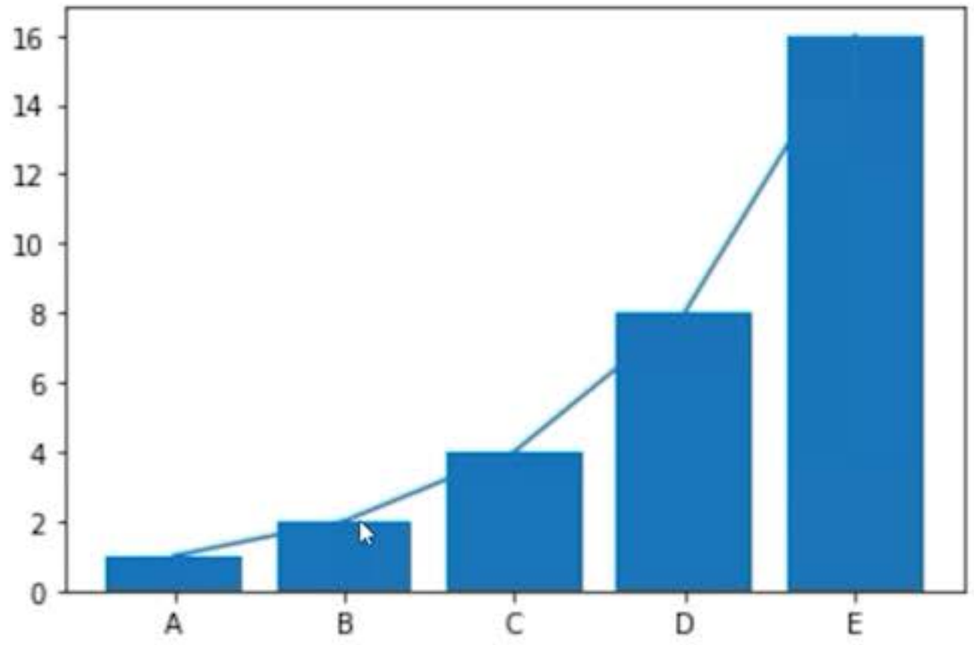
matplotlib 두 개의 그래프를 하나로 그리기

◎ 동일한 axes에 bar graph 그리기

Run Code

In [19]:
```
fig, axes = plt.subplots()
axes.plot(sr)
axes.bar(sr.index, sr.values)
```

Out[19]: <BarContainer object of 5 artists>



In [17]:

Out[17]: [<matplotlib.lines.Line2D at 0x20158fcef70>]

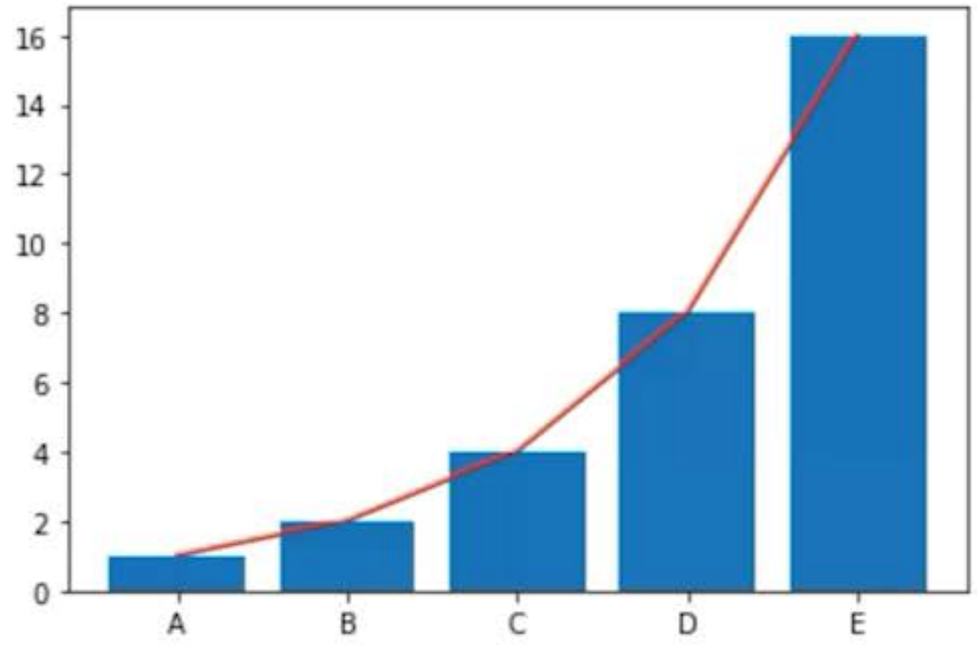
matplotlib 두 개의 그래프를 하나로 그리기

◎ line graph의 color를 red로 변경

Run Code

In [20]:
```
fig, axes = plt.subplots()
axes.plot(sr, color = 'r')
axes.bar(sr.index, sr.values)
```

Out[20]: <BarContainer object of 5 artists>



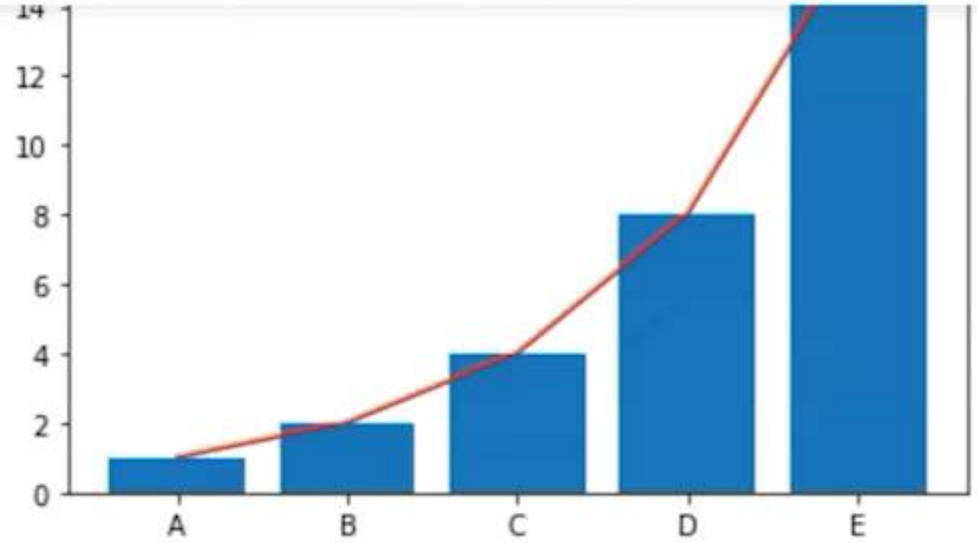
In [17]:

Out[17]: [<matplotlib.lines.Line2D at 0x20158fcef70>]

matplotlib 두 개의 그래프를 서로 다른 y축 적용하기

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code



1.1.4 두 개 그래프를 서로 다른 y축 적용하기

```
In [ ]: sr2 = Series([500, 300, 250, 180, 700], index = ['A','B','C','D',''])
```

```
In [ ]:
```

1.2. 여러 개 그래프 그리기

하나의 figure 안에 여러 개의 axes를 그리기

matplotlib 두 개의 그래프를 서로 다른 y축 적용하기

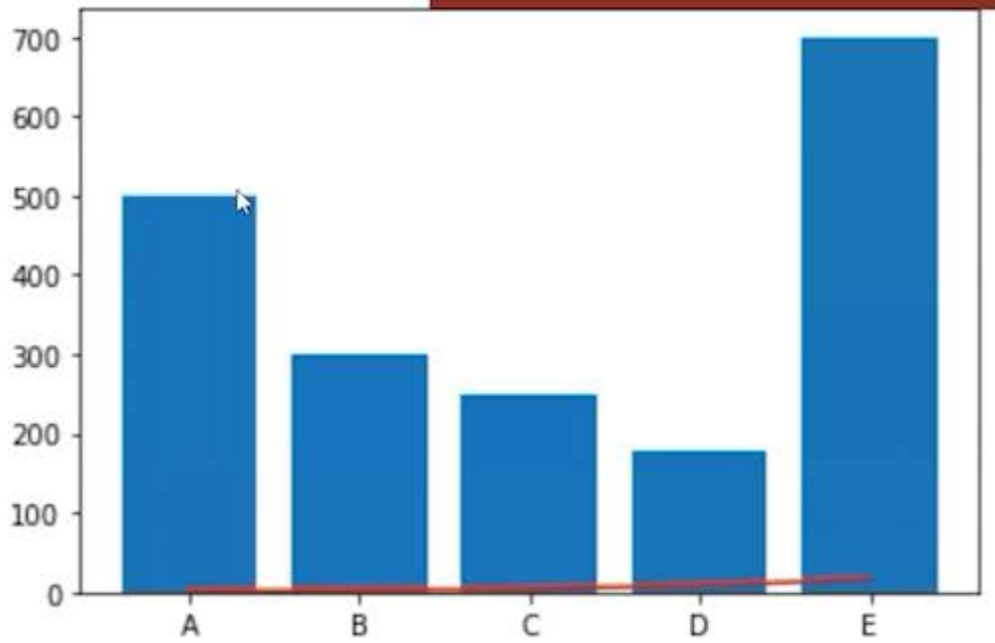
◎ sr : line graph를 빨간색으로 그림

◎ sr2 : bar graph 그림

```
In [22]: fig, ax = plt.subplots()
         ax.plot(sr, color = 'r'
         ax.bar(sr2.index, sr2.v
```

```
Out[22]: <BarContainer object of
```

✔ sr과 sr2의 y값의 범위가 차이가 많이 나서 제대로 보이지 않음



1.2. 여러 개 그래프 그리기

matplotlib 두 개의 그래프를 서로 다른 y축 적용하기

◎ 시리즈 값이 상대적으로 작아 값의 변화를 이해하기 어려움

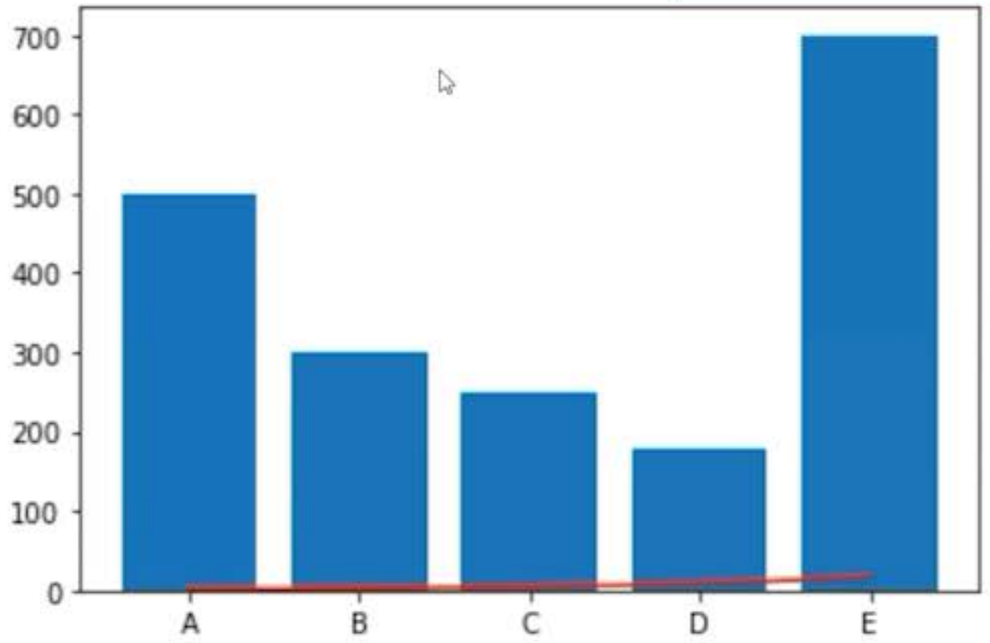
Run Markdown

```
In [21]: sr2 = Series([500, 300, 250, 180, 700], index = ['A','B','C','D','E'])

In [23]: fig, ax = plt.subplots()
         ax.plot(sr, color = 'r')
         ax.bar(sr2.index, sr2.values)

Out[23]: <BarContainer object of 5 artists>
```

- ✔ 2개의 y축을 각각 적용
- ✔ 왼쪽 : bar graph, 오른쪽 : line graph



matplotlib 두 개의 그래프를 서로 다른 y축 적용하기

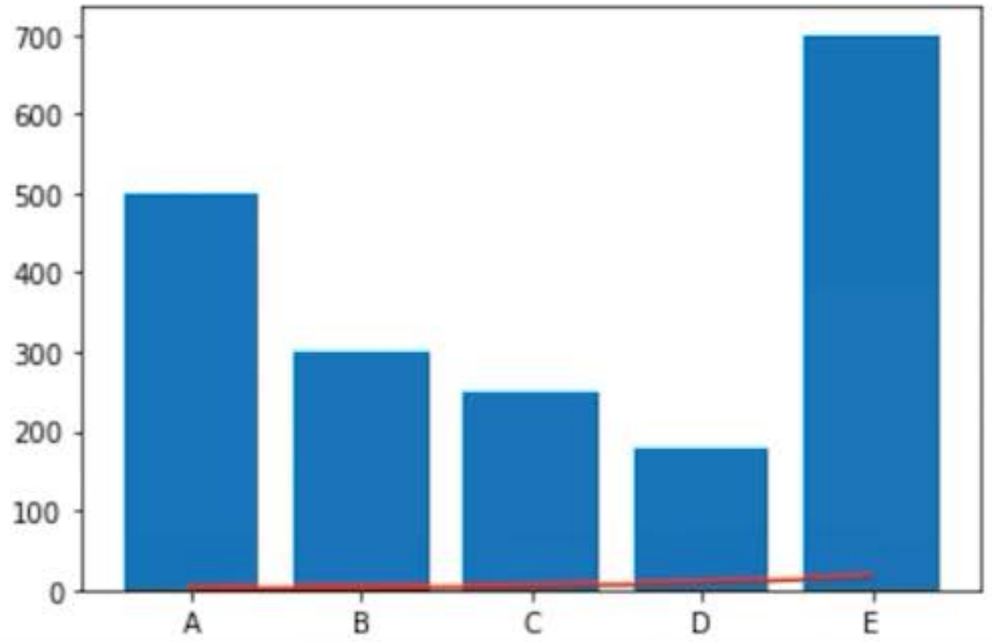
◎ y축이 다르기 때문에 서로 다른 axes에 각각 그려주어야 함

✔ ax2=ax.twinx( ) : x축은 공유

```
, 250, 180, 700], index = ['A','B','C','D','E'])
```

In [23]:
```
fig, ax = plt.subplots()
ax.plot(sr, color = 'r')
ax2 = ax.twinx()
ax2.bar(sr2.index, sr2.values)
```

Out[23]: <BarContainer object of 5 artists>

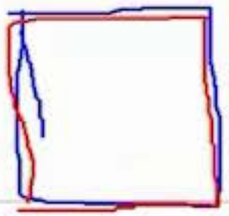


matplotlib 두 개의 그래프를 서로 다른 y축 적용하기

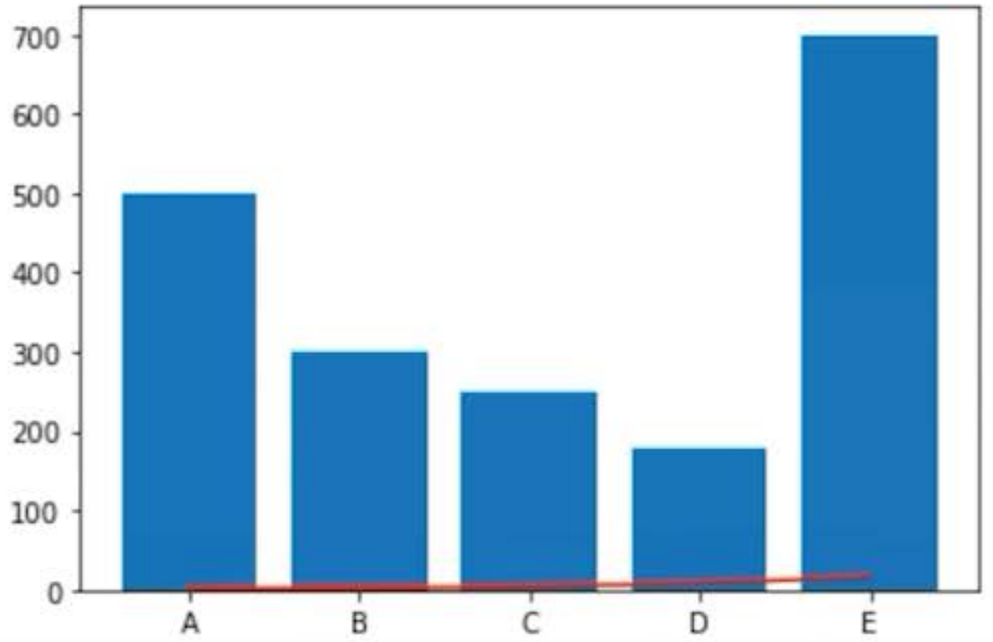
◎ y축이 다르기 때문에 서로 다른 axes에 각각 그려주어야 함

✔ 첫 번째 axes와 두 번째 axes는 x축을 공유해서 중첩하겠다는 의미

```
In [23]: fig, ax = plt.subplots()
         ax.plot(sr, color = 'r')
         ax2 = ax.twinx()
         ax2.bar(sr2.index, sr2.values)
```



```
Out[23]: <BarContainer object of 5 artists>
```



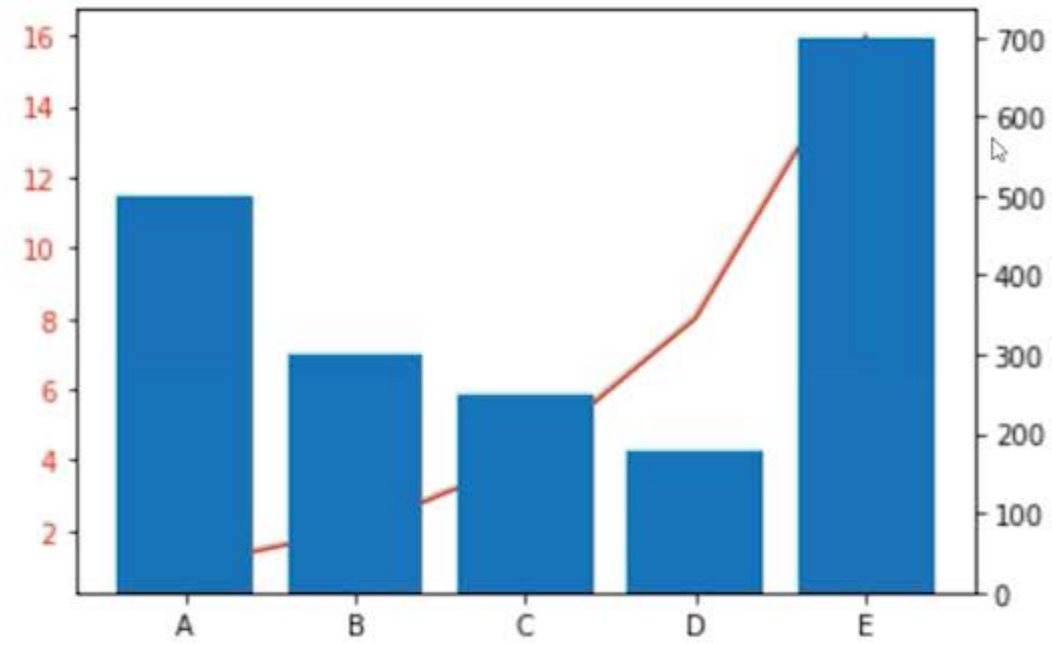
matplotlib 두 개의 그래프를 서로 다른 y축 적용하기

◎ tick_params를 사용하여 line graph y축의 라벨컬러를 레드로 설정

Run Code

In [26]:
```
fig, ax = plt.subplots()
ax.plot(sr, color = 'r')
ax.tick_params(axis = 'y', labelcolor = 'r')
ax2 = ax.twinx()
ax2.bar(sr2.index, sr2.values)
```

Out[26]: <BarContainer object of 5 artists>



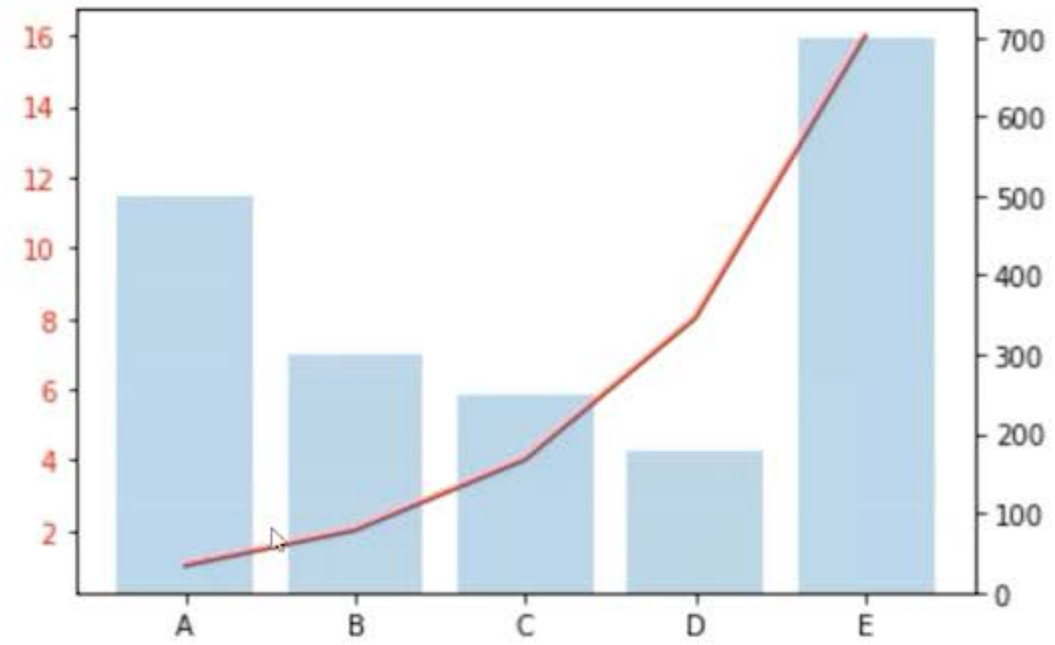
matplotlib 두 개의 그래프를 서로 다른 y축 적용하기

◎ alpha값으로 투명도 조절(0~1 사이의 값)

Run Code

In [28]:
```
fig, ax = plt.subplots()
ax.plot(sr, color = 'r')
ax.tick_params(axis = 'y', labelcolor = 'r')
ax2 = ax.twinx()
ax2.bar(sr2.index, sr2.values, alpha = 0.3)
```

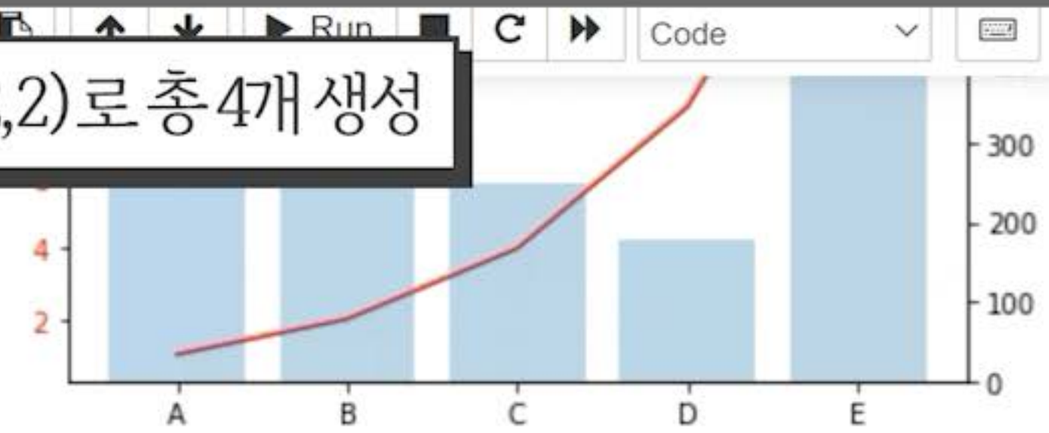
Out[28]: <BarContainer object of 5 artists>



matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

✔ plt.subplots(2,2)로 총 4개 생성



1.2. 여러 개 그래프 그리기

하나의 figure 안에 여러 개의 axes를 그리기

In [ ]:
```
fig, ax = plt.subplots(2,2)
```

In [ ]:

In [ ]:

In [ ]:

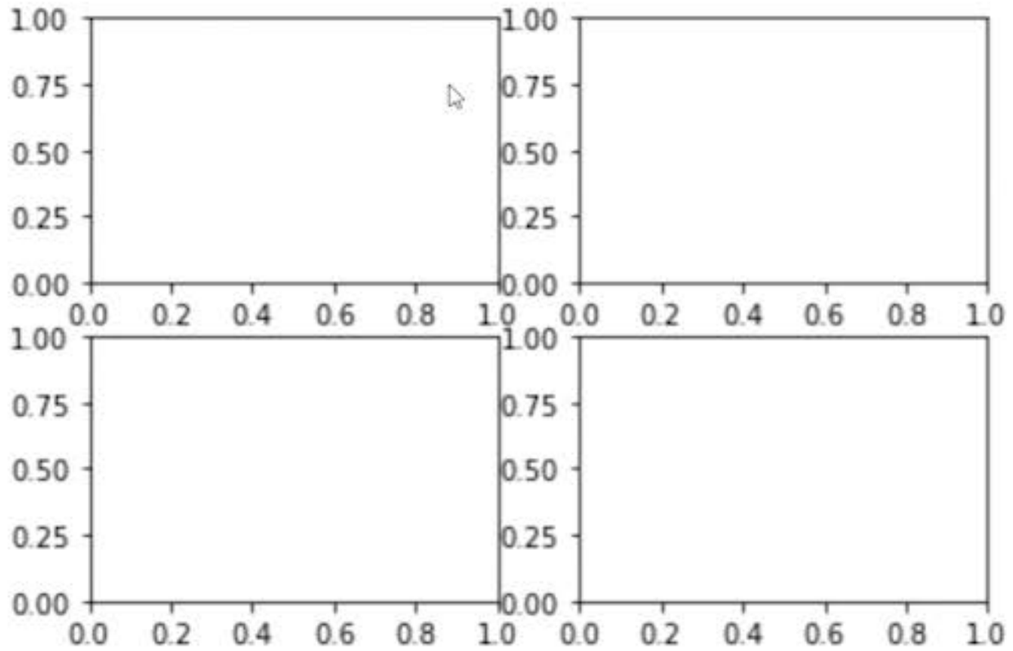
In [ ]:

matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

- ✔ ax ➡ ax_list 로 변경
- ✔ ax_list에는 4개의 각각의 axes가 2x2 배열 형태로 저장되어 있음

In [29]:
```
fig, ax_list = plt.subplots(2,2)
```



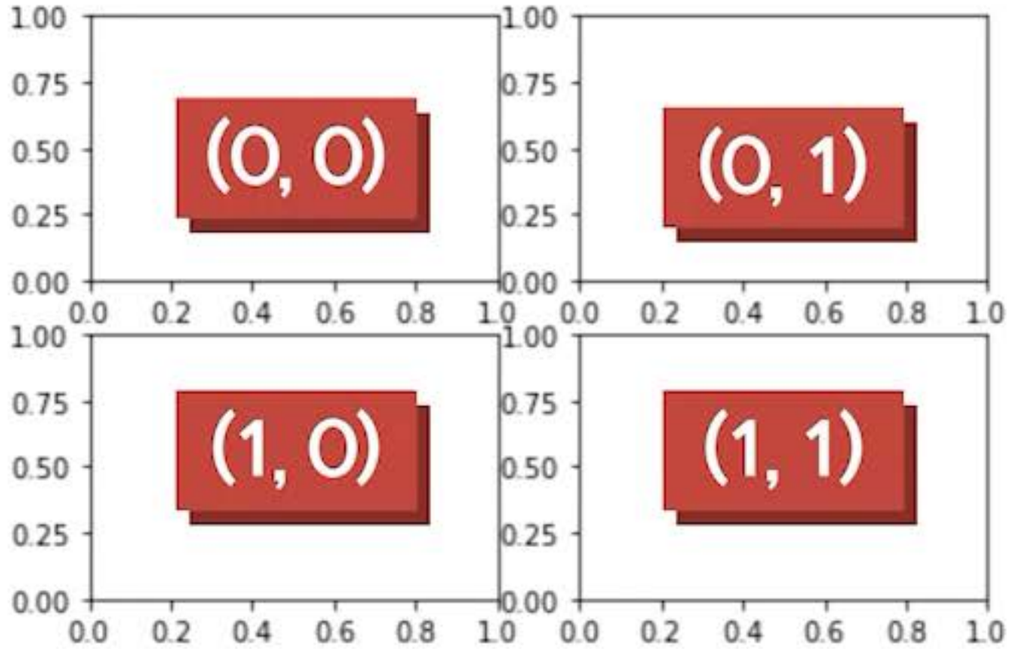
matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

Run Code

하나의 figure 안에 여러 개의 axes를 그리기

In [29]:
```
fig, ax_list = plt.subplots(2,2)
ax
```



In [ ]:

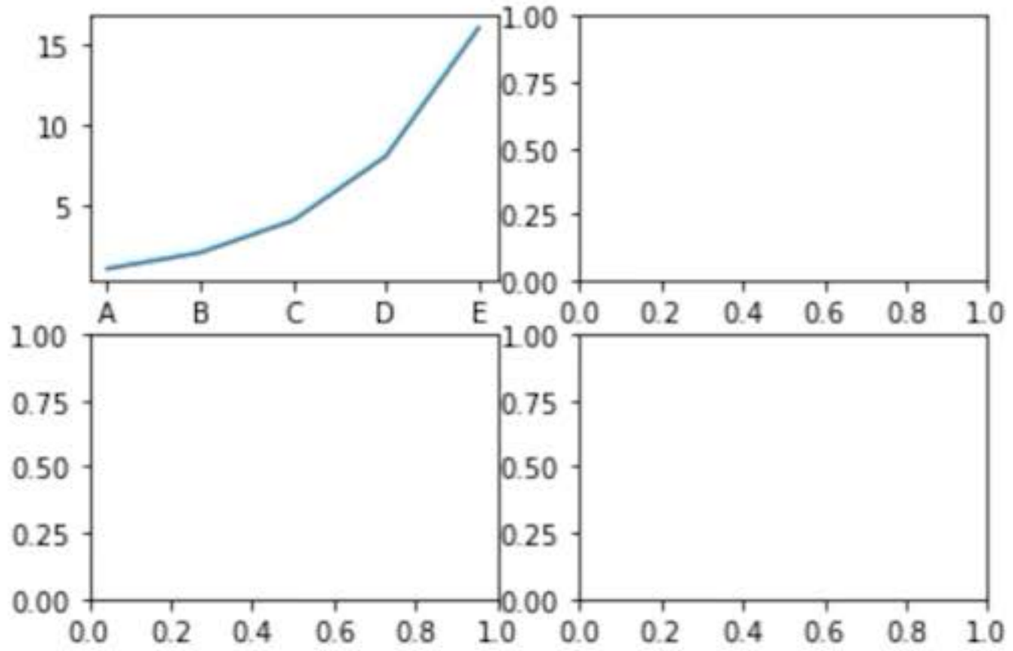
matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

✔ ax_list[0,1]에는 히스토그램(histogram) 그림

```
In [30]: fig, ax_list = plt.subplots(2,2)
         ax_list[0,0].plot(sr)
         ax_list[0,1].hist(ran)

Out[30]: [<matplotlib.lines.Line2D at 0x20159711dc0>]
```



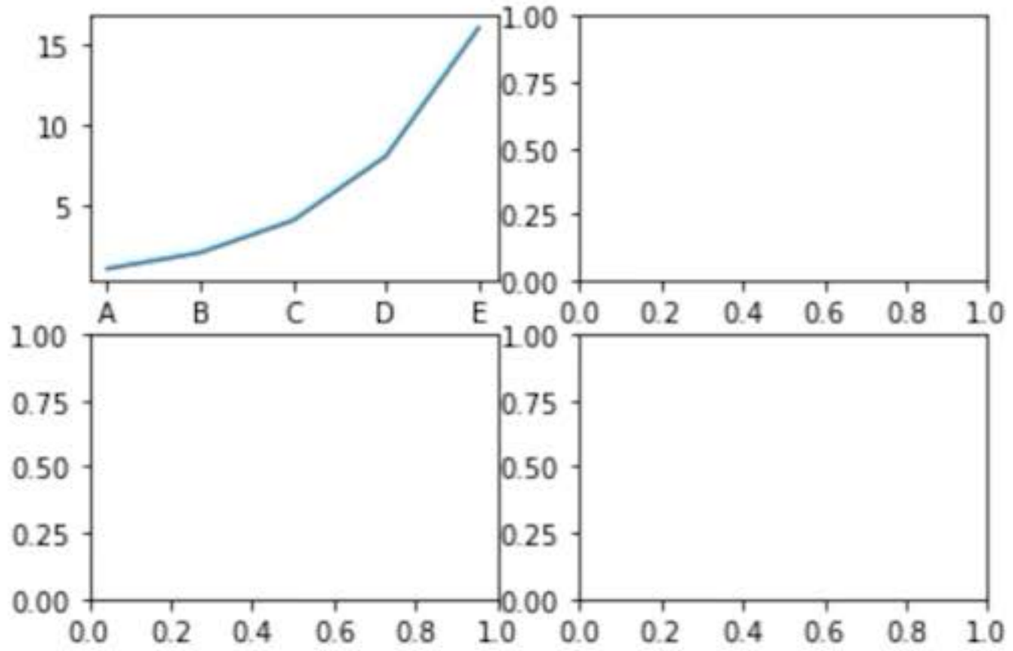
matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

✔ 랜덤하게 생성할 값의 범위 : 1~100사이 값으로 50개 생성

```
In [30]: fig, ax_list = plt.subplots(2,2)
         ax_list[0,0].plot(sr)
         ax_list[0,1].hist(random.sample(range(1,100), 50))
```

```
Out[30]: [<matplotlib.lines.Line2D at 0x20159711dc0>]
```



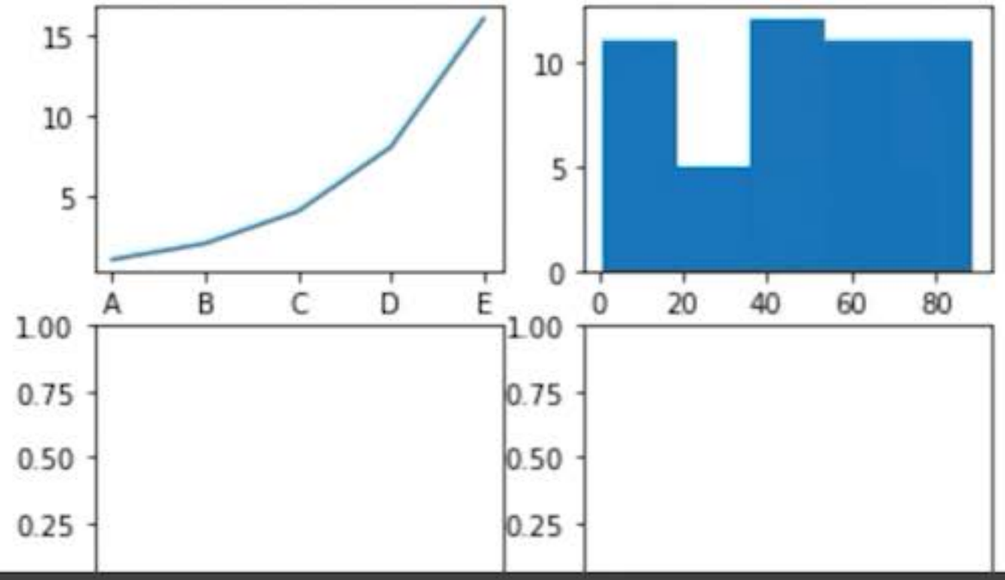
matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

Run Code

```
In [33]: fig, ax_list = plt.subplots(2,2)
         ax_list[0,0].plot(sr)
         ax_list[0,1].hist(random.sample(range(1,100), 50), bins = 5)
```

```
Out[33]: (array([11.,  5., 12., 11., 11.]),
          array([ 1. , 18.6, 36.2, 53.8, 71.4, 89. ]),
          <BarContainer object of 5 artists>)
```



5개의 버켓으로 데이터의 분포도를 확인 가능합니다.

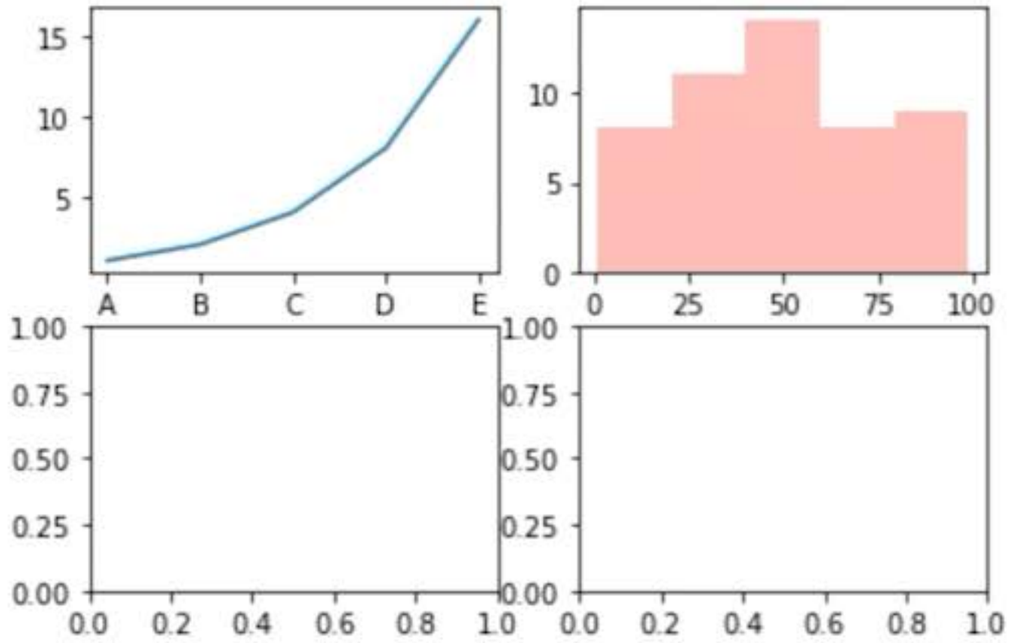
matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

Run Code

In [34]:
```
fig, ax_list = plt.subplots(2,2)
ax_list[0,0].plot(sr)
ax_list[0,1].hist(random.sample(range(1,100), 50), bins = 5, color = 'r', alpha = 0.3)
```

Out[34]:
```
(array([ 8., 11., 14.,  8.,  9.]),
 array([  1. ,  20.6,  40.2,  59.8,  79.4,  99. ]),
 <BarContainer object of 5 artists>)
```



matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

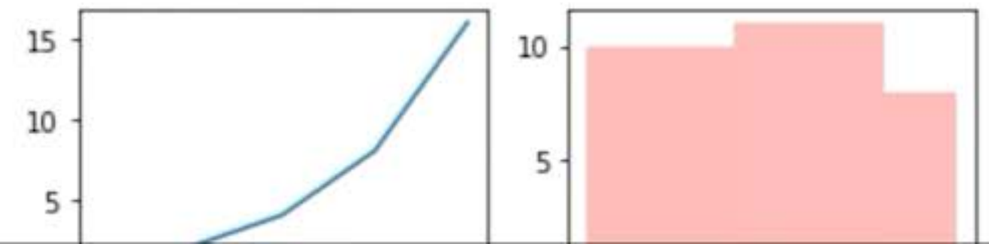
✔ ax_list[1,0]에는 Bar graph 그림

Run Code

In [35]:
```
fig, ax_list = plt.subplots(2,2)
ax_list[0,0].plot(sr)
ax_list[0,1].hist(random.sample(range(1,100), 50), bins = 5, color = 'r', alpha = 0.3)
ax_list[1,0].bar(sr2.index, sr2_values)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-35-d2920c951150> in <module>
      2 ax_list[0,0].plot(sr)
      3 ax_list[0,1].hist(random.sample(range(1,100), 50), bins = 5, color = 'r', al
pha = 0.3)
----> 4 ax_list[1,0].bar(sr2.index, sr2_values)

NameError: name 'sr2_values' is not defined
```



matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

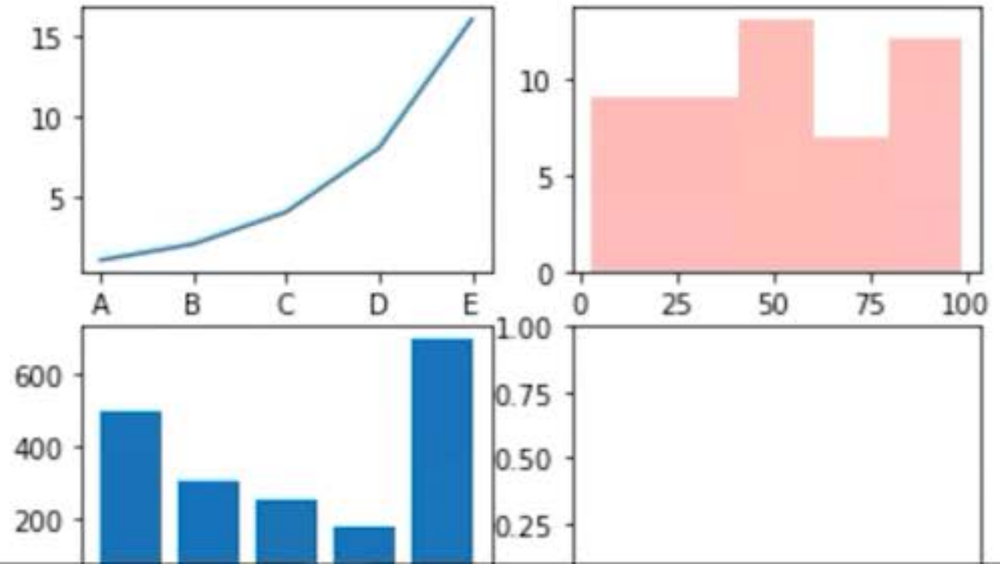
✔ ax_list[1,1]에는 scatter graph 그림

Run Code

In [32]:
```
import random
```

In [36]:
```
fig, ax_list = plt.subplots(2,2)
ax_list[0,0].plot(sr)
ax_list[0,1].hist(random.sample(range(1,100), 50), bins = 5, color = 'r', alpha = 0.3)
ax_list[1,0].bar(sr2.index, sr2.values)
ax_list[1,1].sca
```

Out[36]: <BarContainer object of 5 artists>



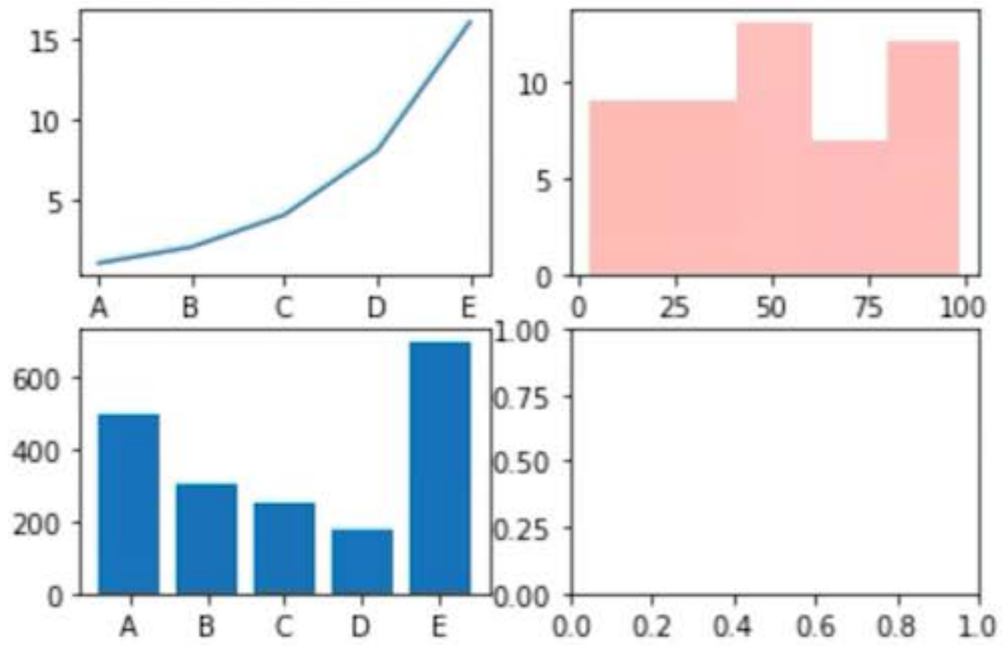
matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

✔ random.sample을 x값과 y값에 적용

```
ax_list[0,0].plot(sr)
ax_list[0,1].hist(random.sample(range(1,100), 50), bins = 5, color = 'r', alpha = 0.3)
ax_list[1,0].bar(sr2.index, sr2.values)
ax_list[1,1].scatter(random.sample(range(1,100), 50), random.sample(range(1,100), 50))
```

Out[36]: <BarContainer object of 5 artists>



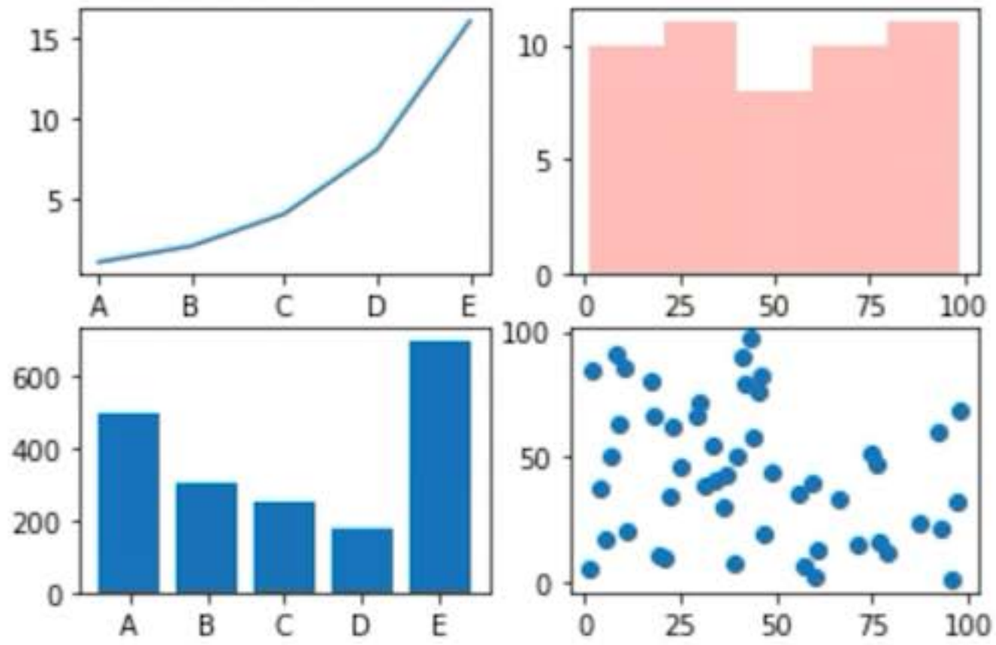
matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

✔ 알파값을 적용하여 데이터의 밀집도를 표시함

```
ax_list[0,0].plot(sr)
ax_list[0,1].hist(random.sample(range(1,100), 50), bins = 5, color = 'r', alpha = 0.3)
ax_list[1,0].bar(sr2.index, sr2.values)
ax_list[1,1].scatter(random.sample(range(1,100), 50), random.sample(range(1,100), 50))
```

Out[37]: <matplotlib.collections.PathCollection at 0x20159cf62e0>



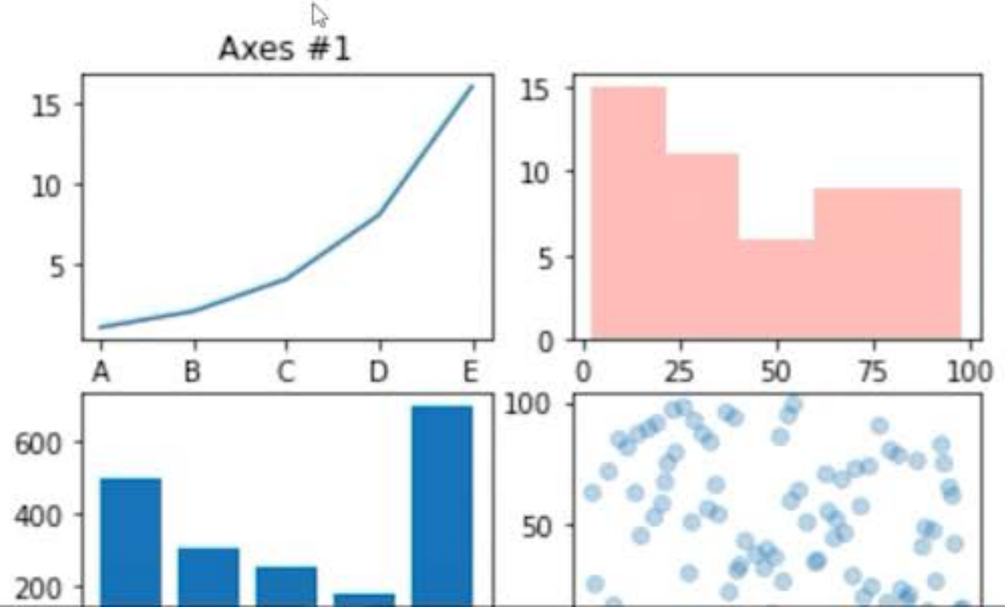
matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

✔ ax_list[0,0]에 set_title로 axes의 이름 부여

```
ax_list[0,0].plot(sr)
ax_list[0,0].set_title('Axes #1')
ax_list[0,1].hist(random.sample(range(1,100), 50), bins = 5, color = 'r', alpha = 0.3)
ax_list[1,0].bar(sr2.index, sr2.values)
ax_list[1,1].scatter(random.sample(range(1,100), 90), random.sample(range(1,100), 90),
                     alpha = 0.3)
```

Out[40]: <matplotlib.collections.PathCollection at 0x201597bfd00>



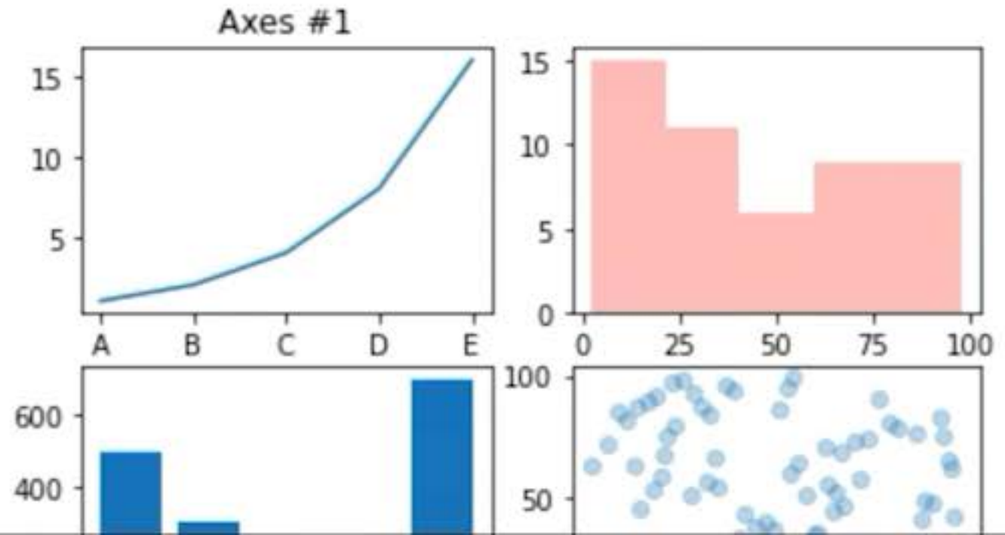
matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

✔ ax_list[0,0]에 x, y label 붙이기

```
s(2,2)
ax_list[0,0].plot(sr)
ax_list[0,0].set_title('Axes #1')
ax_list[0,0].x_label()
ax_list[0,1].hist(random.sample(range(1,100), 50), bins = 5, color = 'r', alpha = 0.3)
ax_list[1,0].bar(sr2.index, sr2.values)
ax_list[1,1].scatter(random.sample(range(1,100), 90), random.sample(range(1,100), 90),
                     alpha = 0.3)
```

Out[40]: <matplotlib.collections.PathCollection at 0x201597bfd00>



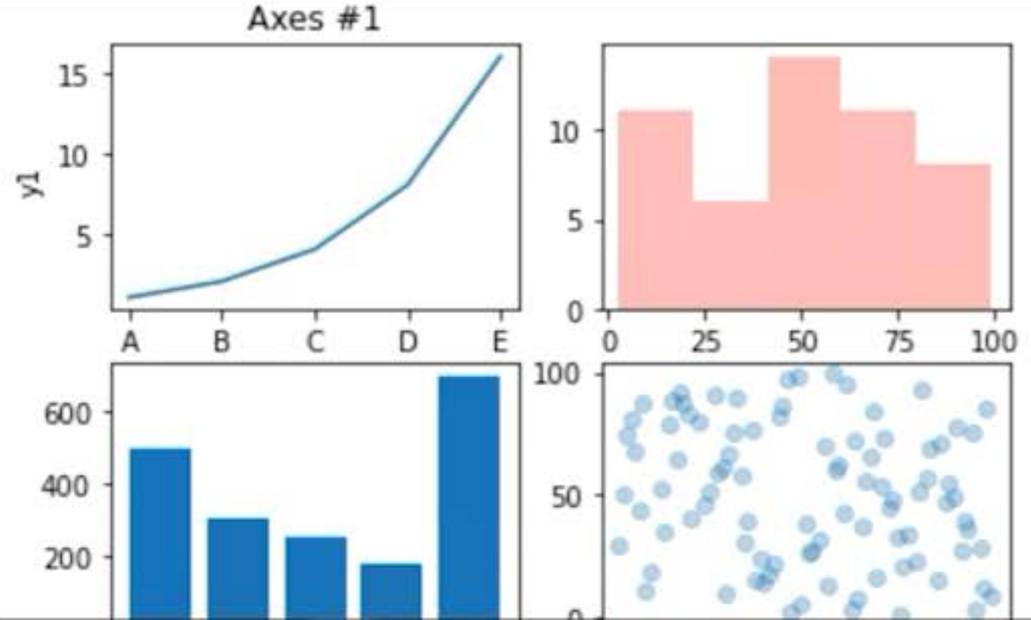
matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기


```
ax_list[0,0].plot(sr)
ax_list[0,0].set_title('Axes #1')
ax_list[0,0].set_xlabel('x1')
ax_list[0,0].set_ylabel('y1')
ax_list[0,1].hist(random.sample(range(1,100), 50), bins = 5, color = 'r', alpha = 0.3)
ax_list[1,0].bar(sr2.index, sr2.values)
ax_list[1,1].scatter(random.sample(range(1,100), 90), random.sample(range(1,100), 90),
                     alpha = 0.3)
```

주의하기

✔ x축 제목은 아래쪽 그래프(axes)에 의해 가려져서 안보이는 것임

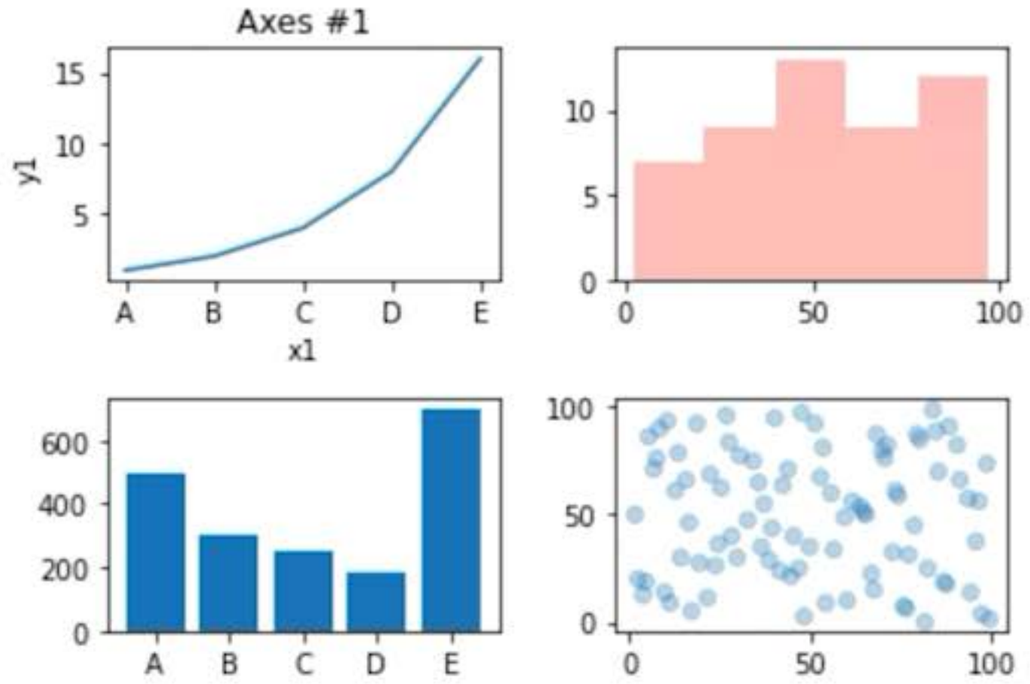


matplotlib 여러 개 그래프 그리기

◎ 하나의 figure 안에 여러 개의 axes를 그리기

✔ 각 axes들의 여백 조정

```
set_xlabel('x1')
set_ylabel('y1')
hist(random.sample(range(1,100), 50), bins = 5, color = 'r', alpha = 0.3)
ax_list[1,0].bar(sr2.index, sr2.values)
ax_list[1,1].scatter(random.sample(range(1,100), 90), random.sample(range(1,100), 90),
                     alpha = 0.3)
plt.subplots_adjust(hspace = 0.5, wspace = 0.3)
```



학습완료

1/ 단일 Axes 그리기 〈 라인 차트, 바 차트 〉

2/ 2개 이상의 Axes를 하나의 Figure에 그리기