

Matplotlib Tutorial

https://github.com/dlcjfgmlnasa/Tongmyong_College_Tensorflow_Tutorial

Created by **Choelhui lee**

Matplotlib

- **Matplotlib** 는 다양한 데이터를 많은 방법으로 도식화 할 수 있도록 하는 파이썬 라이브러리로서, 우리는 matplotlib의 pyplot 을 이용하게 됩니다.
- **Matplotlib** 는 이용하면 우리가 이전에 알아본 numpy나 pandas에서 사용되는 자료구조를 쉽게 시각화 할 수 있습니다.



Matplotlib Install

```
pip install matplotlib
```

Matplotlib import

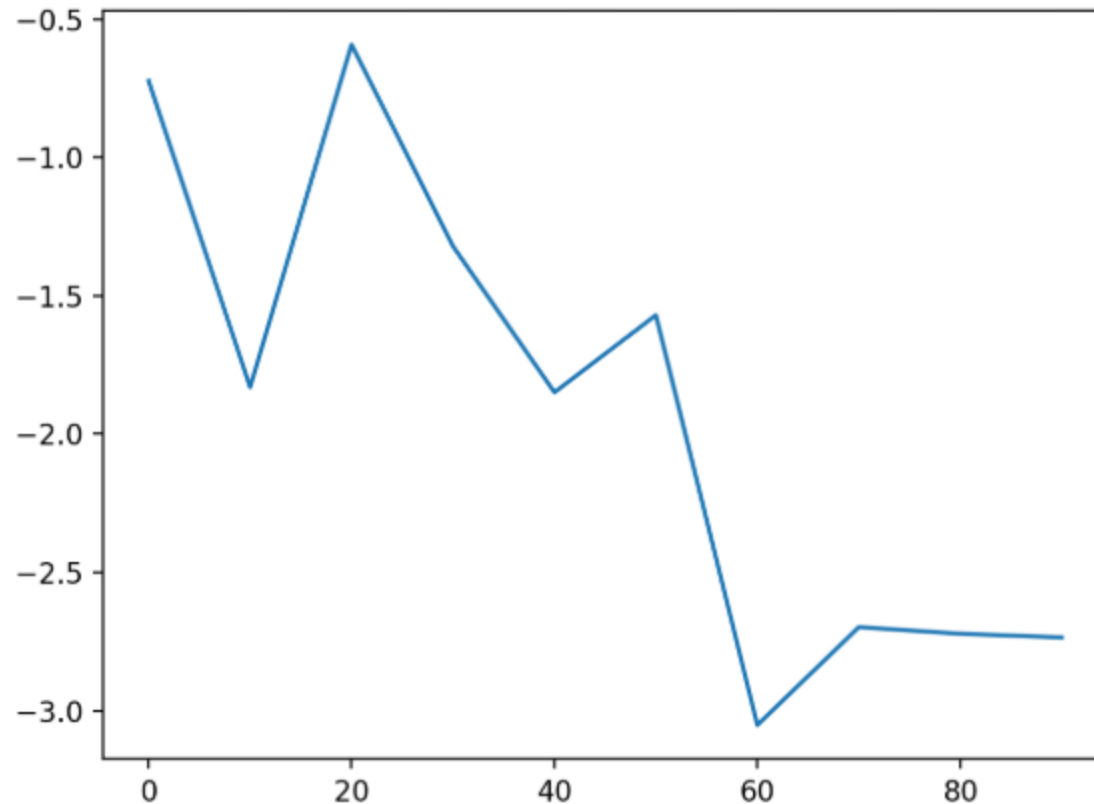
```
import matplotlib  
import matplotlib.pyplot as plt
```

01. Plot의 종류

01. Line plot 그리기

Series를 통한 line plot 그리기

```
s = pd.Series(np.random.randn(10).cumsum()),  
s.plot()
```



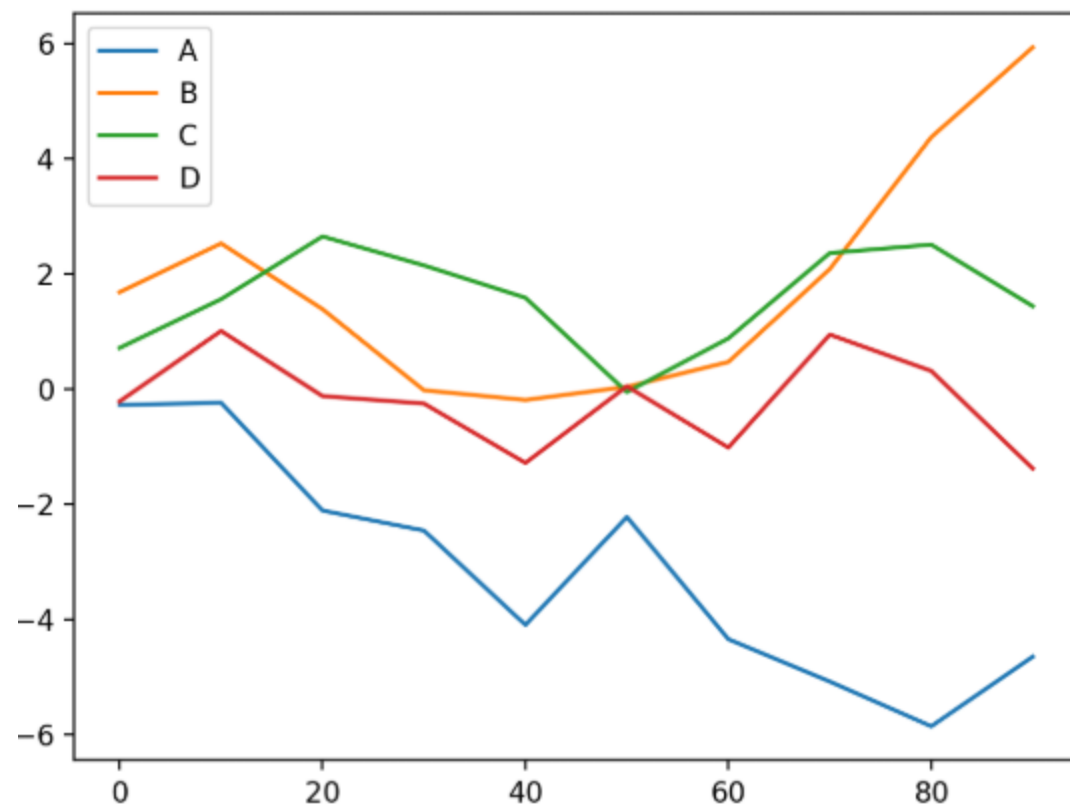
- `s` 라는 Series에서의 `index`와 `value`를 통해 그래프가 그려졌다
- 그래프 우측상단의 전원버튼을 누르기 전까지 우리는 해당 그래프를 `interactive`하게 조작할 수 있다.

DataFrame을 통한 line plot 그리기

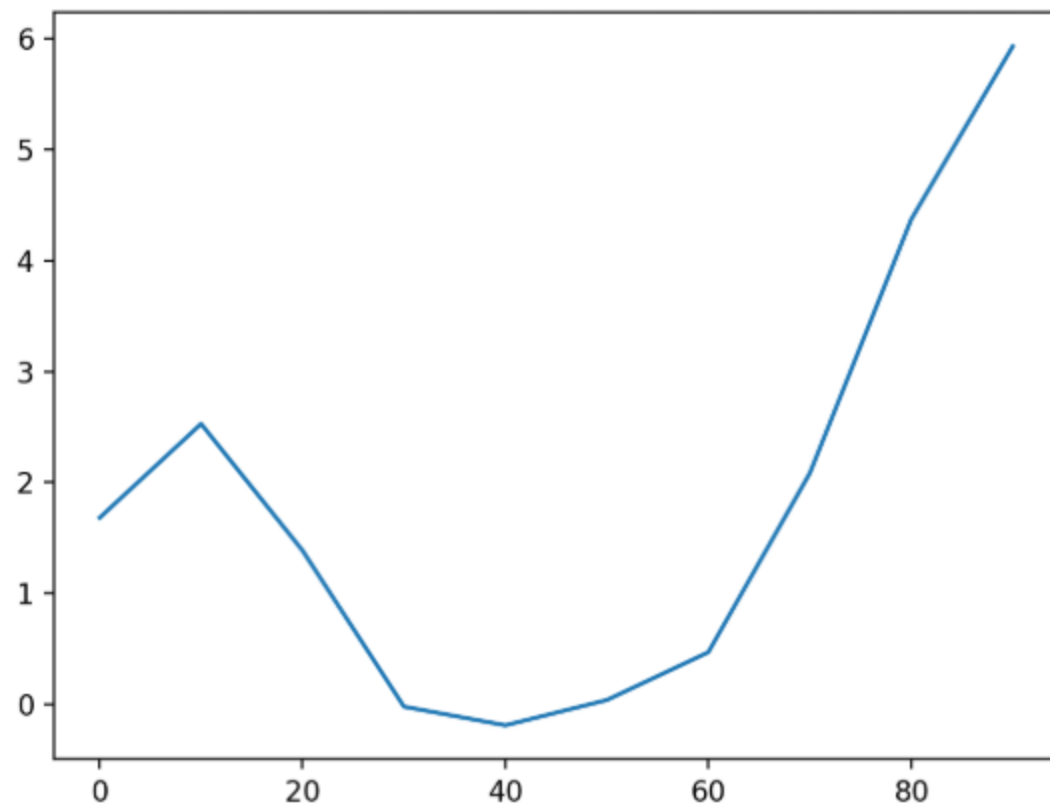
```
df = pd.DataFrame(np.random.randn(10, 4).cumsum(axis=0),  
                  columns=["A", "B", "C", "D"],  
                  index=np.arange(0, 100, 10))
```

	A	B	C	D
0	-0.278464	1.680385	0.711803	-0.216933
10	-0.239848	2.527778	1.558551	1.006354
20	-2.112968	1.384759	2.648977	-0.124528
30	-2.461009	-0.023573	2.145466	-0.253936
40	-4.098926	-0.191797	1.583091	-1.285248
50	-2.224330	0.036316	-0.053839	0.045480
60	-4.346708	0.467878	0.877064	-1.018642
70	-5.083230	2.082973	2.360633	0.942955
80	-5.860602	4.372568	2.506778	0.312459
90	-4.651125	5.932881	1.437739	-1.382153

```
df.plot()
```

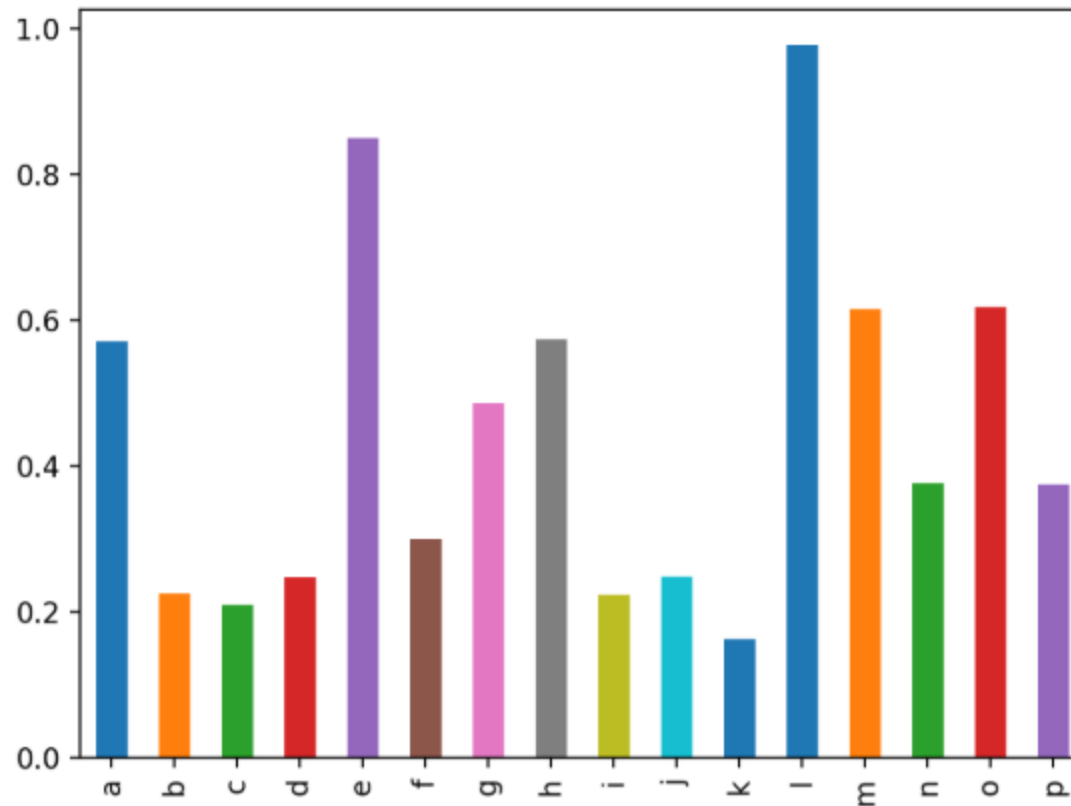


```
df['B'].plot()
```



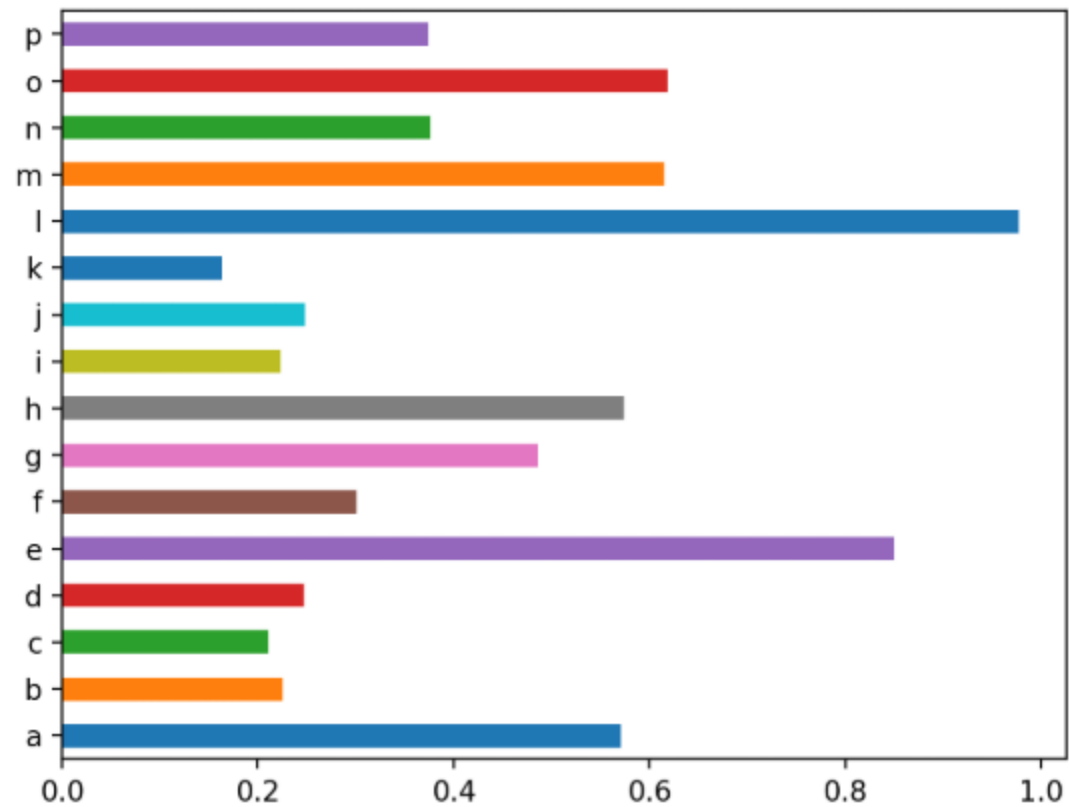
02. Bar plot 그리기

```
s2 = pd.Series(np.random.rand(16),  
               index=list("abcdefghijklmnop"))  
  
s2.plot(kind='bar')
```



가로방향의 bar plot그리기

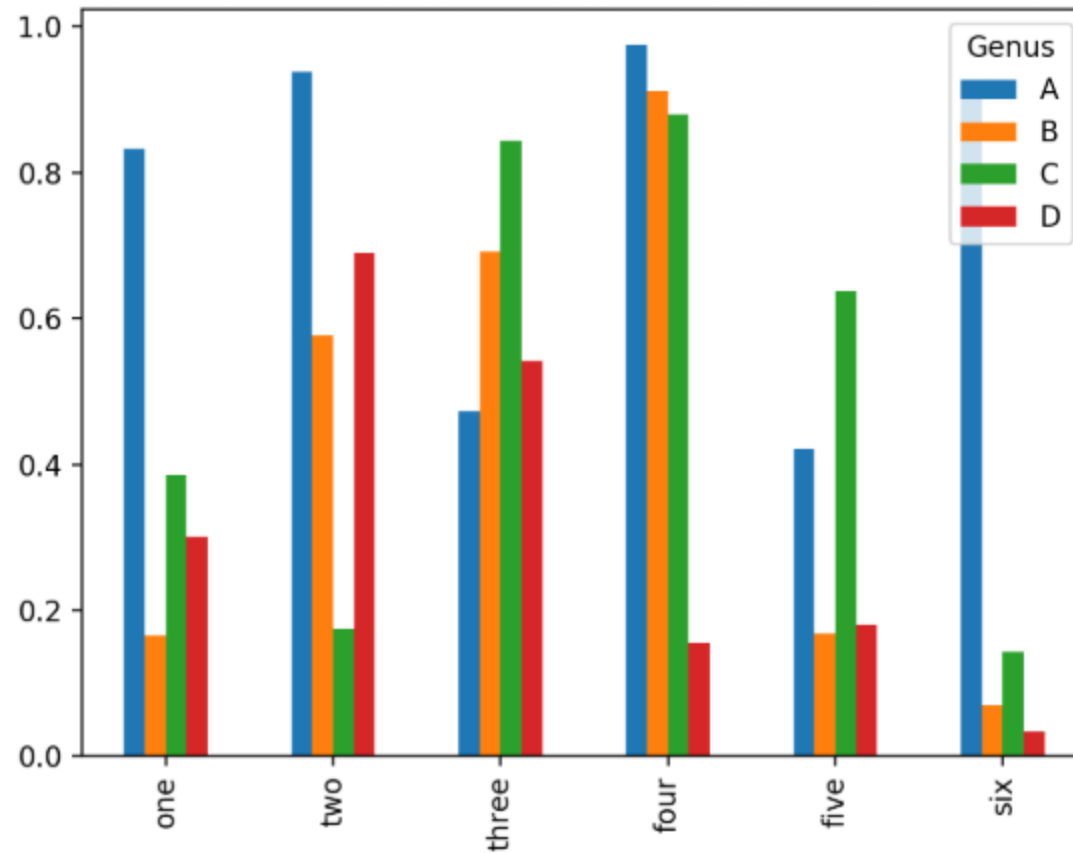
```
s2.plot(kind='barh')
```



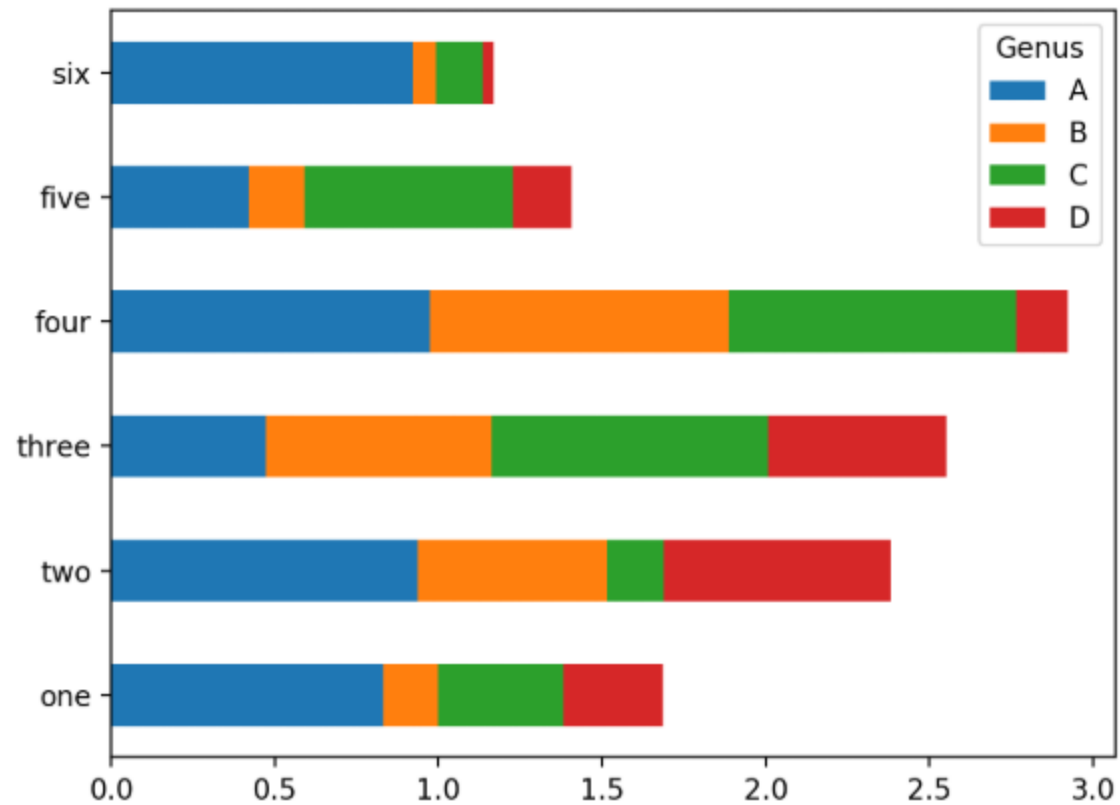
```
df2 = pd.DataFrame(np.random.rand(6, 4),  
                    index=["one", "two", "three", "four", "five", "six"],  
                    columns=pd.Index(["A", "B", "C", "D"], name="Genus"))
```

Genus	A	B	C	D
one	0.832213	0.165459	0.385868	0.300776
two	0.937578	0.576798	0.175512	0.690425
three	0.473119	0.690937	0.844016	0.542061
four	0.974779	0.911599	0.880104	0.155459
five	0.421689	0.168038	0.637749	0.181037
six	0.921647	0.069688	0.143649	0.033414

```
df2.plot(kind='bar')
```



```
df2.plot(kind='barh', stacked=True)
```



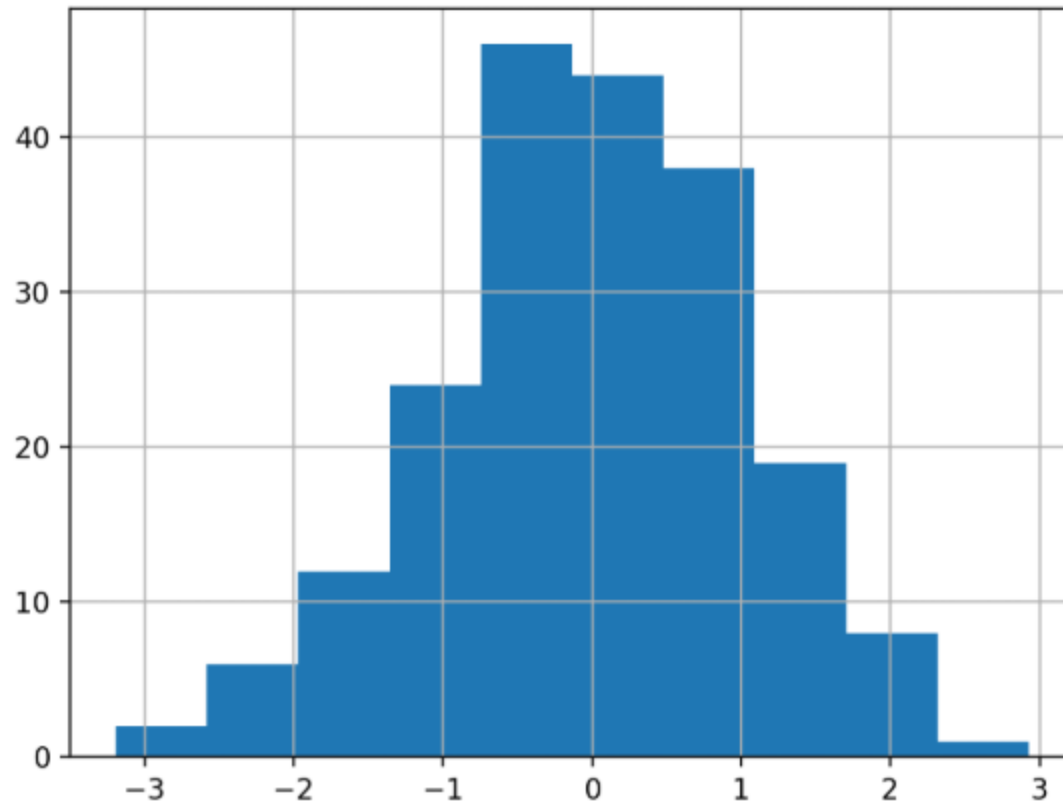
위와 같이 `Stacked` 속성을 `True`로 설정하면, 하나의 인덱스에 대한 각 열의 값을 한줄로 쌓아서 나타내준다.

03. Histogram 그리기

histogram은 index가 필요없다.

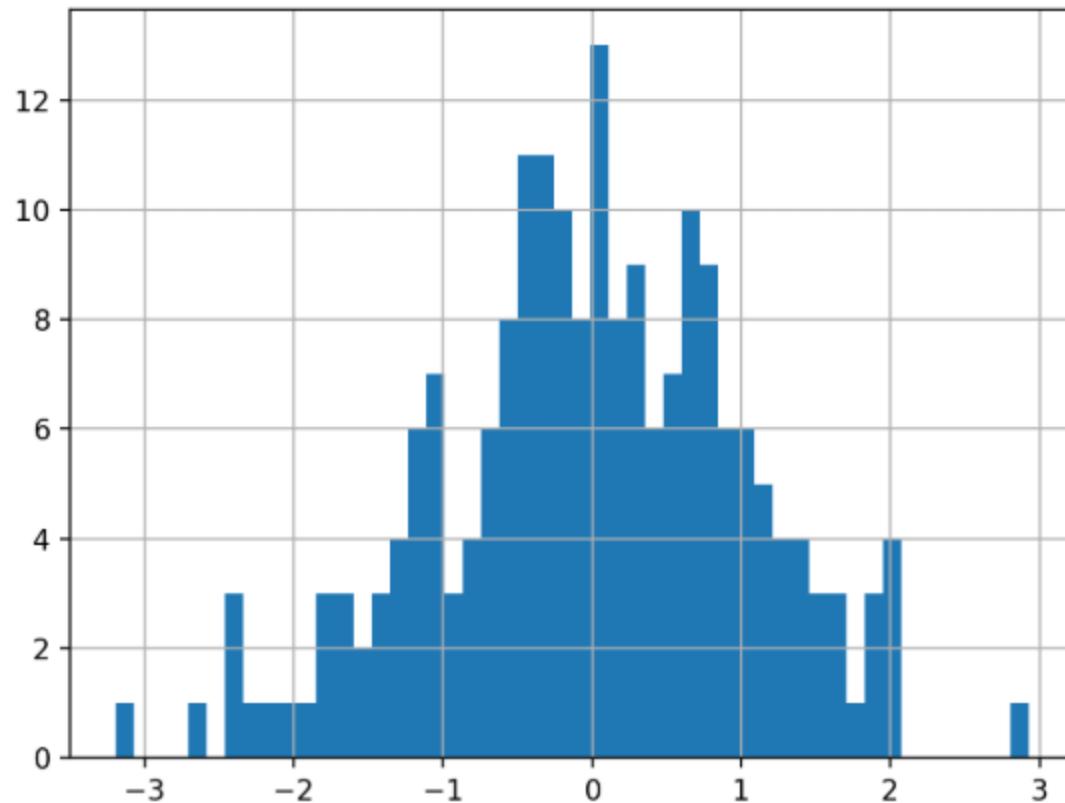
```
s3 = pd.Series(np.random.normal(0, 1, size=200))
```

```
s3.hist()
```



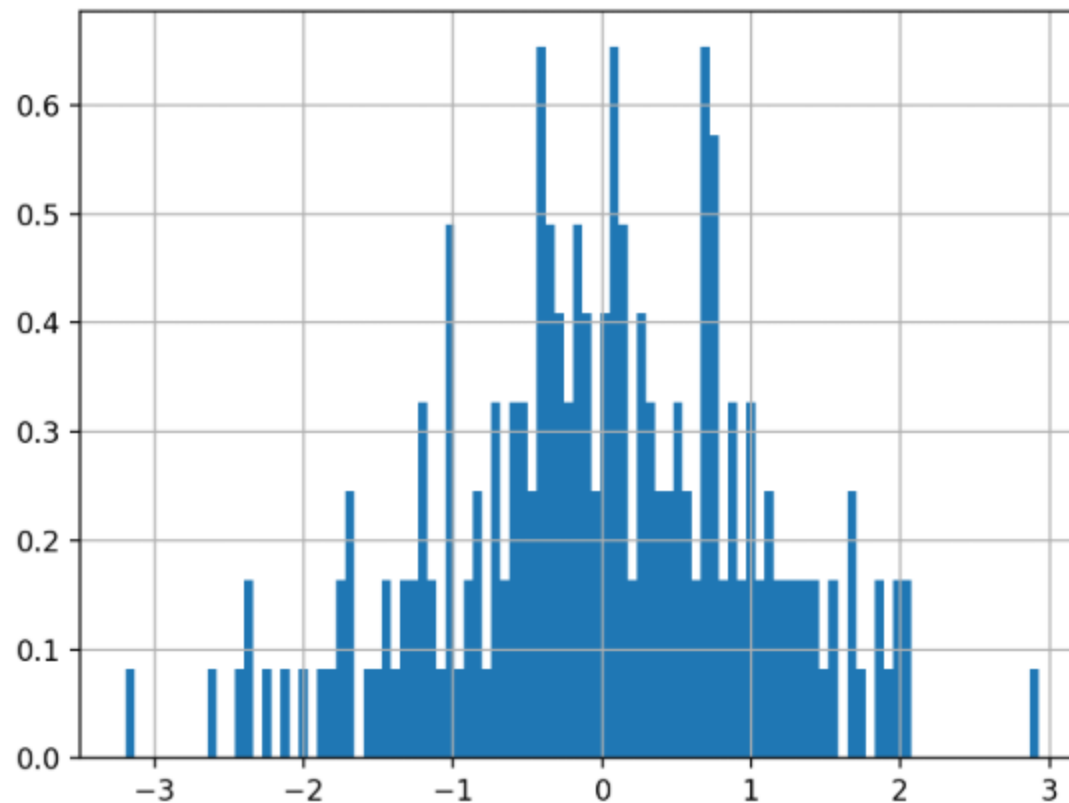
- x 축의 구간 개수를 bin이라고 한다.
- 이를 직접 설정

```
s3.hist(bins=50)
```



normed 속성을 *True*로 설정하면,
각 *bin*에 속하는 개수를 전체 개수로 나눈 비율,
즉 정규화 한 값을 *bar*의 높이로 사용하게 된다.

```
s3.hist(bins=100, normed=True)
```



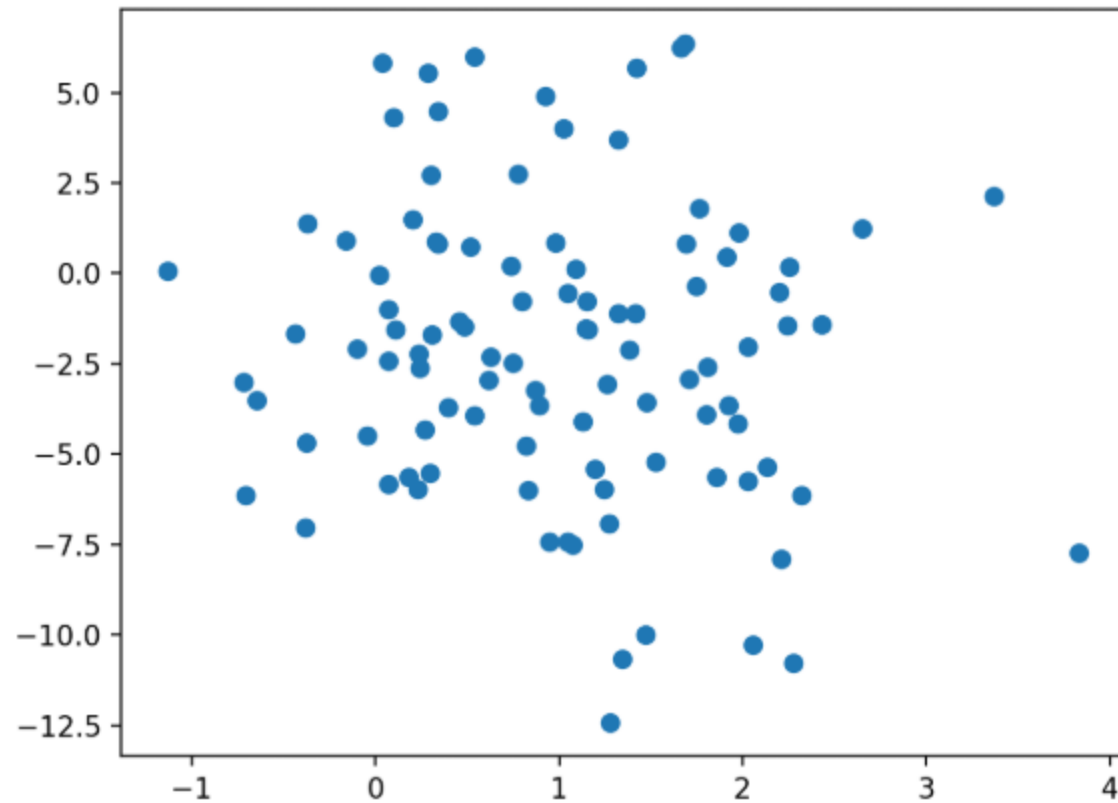
04. 산점도(Scatter plot) 그리기

산점도의 경우에는 서로 다른 두 개의 독립변수에 대해 두 변수가 어떤 관계가 있는지 살펴보기 위해 사용된다.

```
x1 = np.random.normal(1, 1, size=(100, 1))  
x2 = np.random.normal(-2, 4, size=(100, 1))  
X = np.concatenate((x1, x2), axis=1)  
X
```

```
df3 = pd.DataFrame(X, columns=["x1", "x2"])  
df3
```

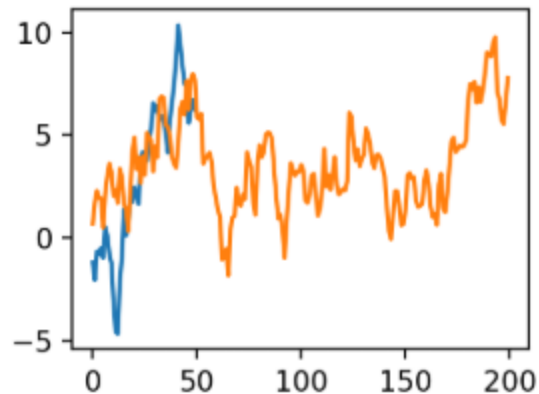
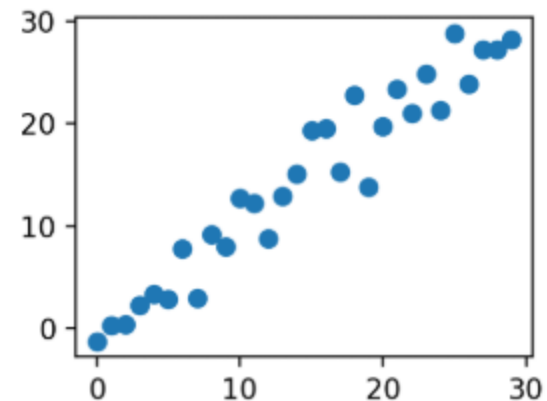
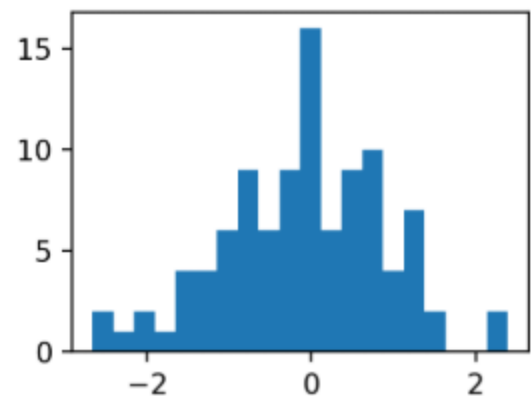
```
plt.scatter(df3['x1'], df3['x2']) # x1 이 x 축, x2 가 y 축
```



05. Plot 모양 변형하기

```
fig = plt.figure()
```

비어있는 figure가 생성된다.



subplot 추가하기, add_subplot 에는 총 3가지 인자가 들어간다.

```
ax1 = fig.add_subplot(2, 2, 1)
```

- **첫번째 숫자와 두번째 숫자** : 우리가 figure를 어떤 크기로 나눌지에 대한 값이다. 즉 위의 같은 경우는 2,2 이므로 우리의 figure를 2x2로 나눈다는 뜻
- **세번째 숫자** : 첫번째, 두번째 숫자로 나눈 figure에서 좌측상단으로 우측방향으로 숫자가 붙는다. 이때 우리가 add하고자 하는 subplot이 몇번째에 들어가는지를 나타낸다.
- 즉, 위와 같은 경우 figure는 다음과 같이 나누어진다.

```
1  2  
3  4
```

- 이때 우리는 1위치에 subplot을 추가하고 해당 subplot을 ax1이라는 변수로 반환받는다.

```
ax2 = fig.add_subplot(2,2,2)  
ax3 = fig.add_subplot(2,2,3)
```