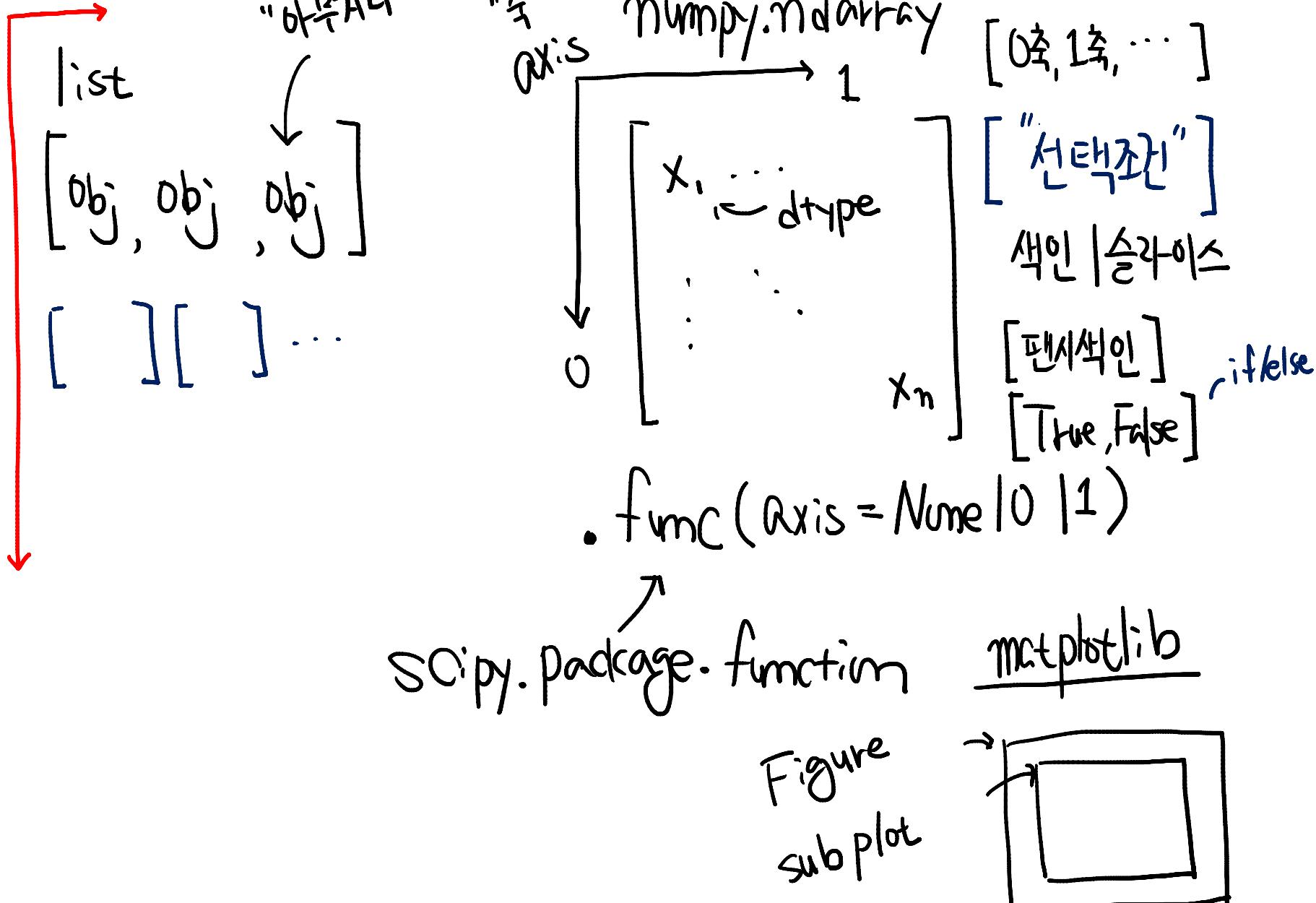


파이썬 딥러닝

이성주

seongjoo@codebasic.io

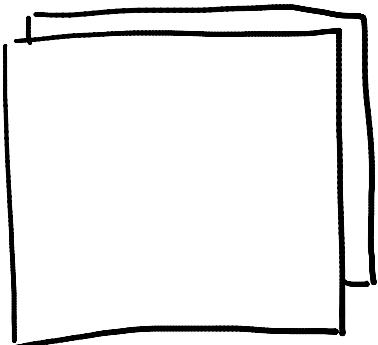


- read_csv()
- read_excel()

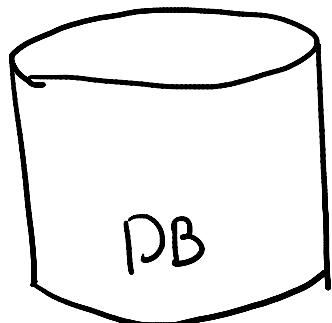
:

" I/O "

"수치연산"



File

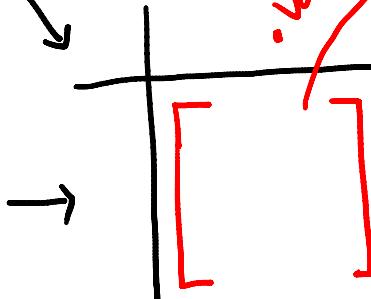


www

pandas

numpy / scipy

values



→ DataFrame

• to_csv()

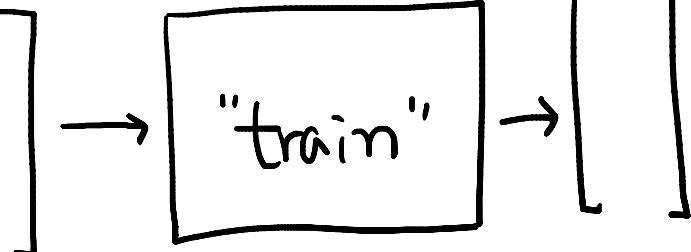
• to_excel()

• to_sql()

Scikit-Learn
TensorFlow / Keras

Model 1

ndarray

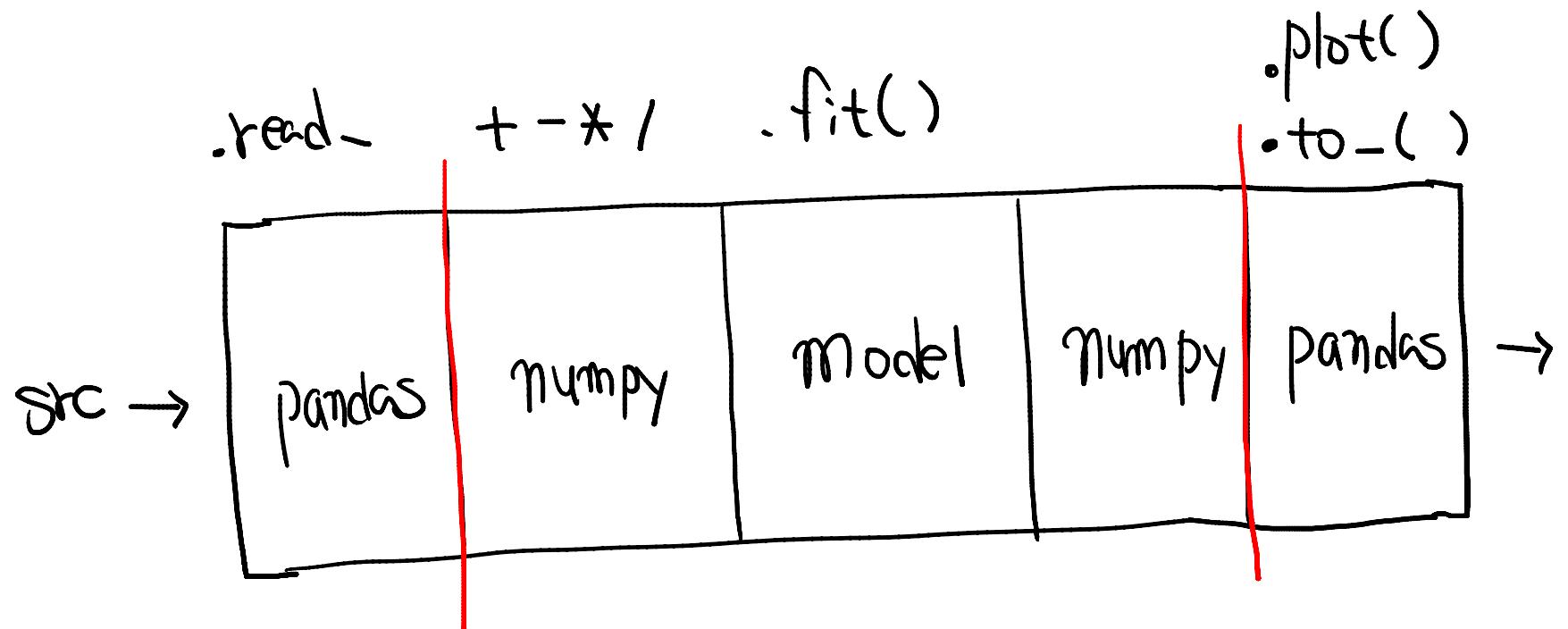


"평가"

pandas

matplotlib

• plot()



File Edit View Insert Cell Kernel Widgets Help

In [1]: `import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline`

In [2]: `plt.plot(np.arange(10))`

Out[2]: [`<matplotlib.lines.Line2D at 0x81bc860>`]

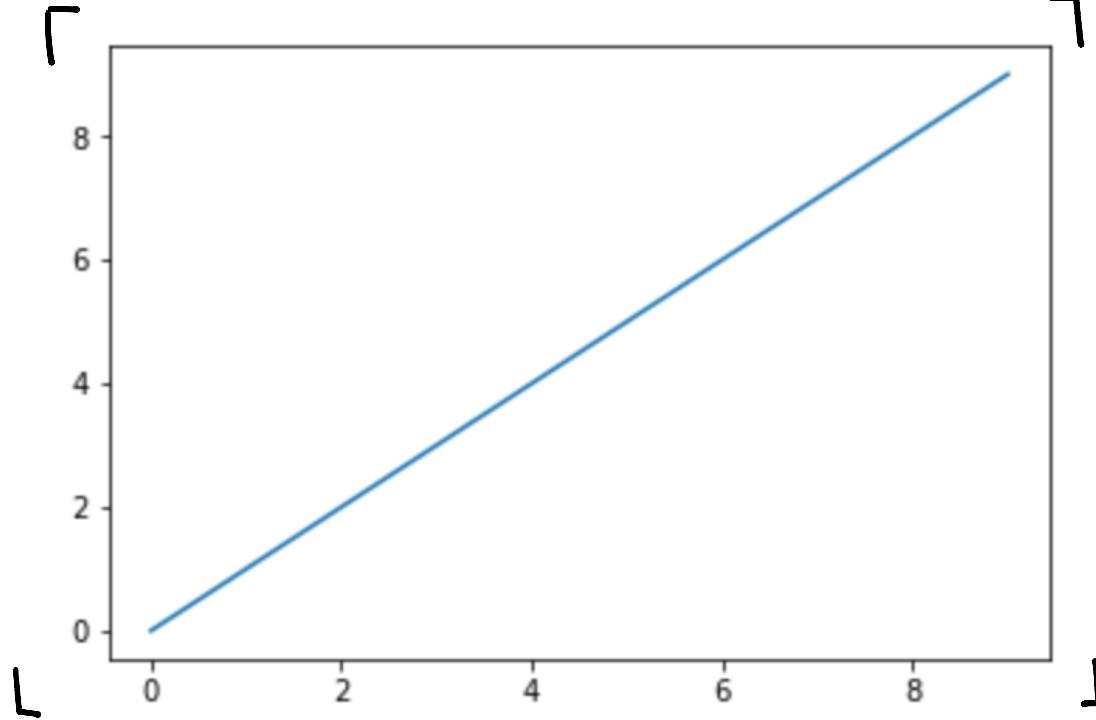


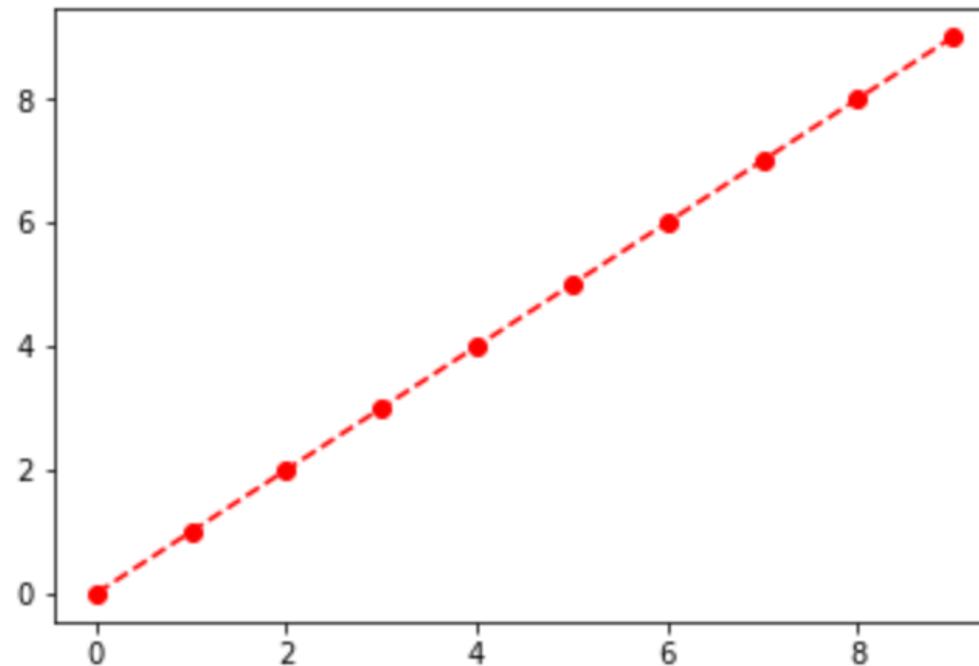
Figure "E"

In [2]: plt.plot(np.arange(10))

영문자 "q"
↓

In [3]: plt.plot(np.arange(10), linestyle='--', color='red', marker='o')

Out[3]: [<matplotlib.lines.Line2D at 0x6580c88>]



In []:

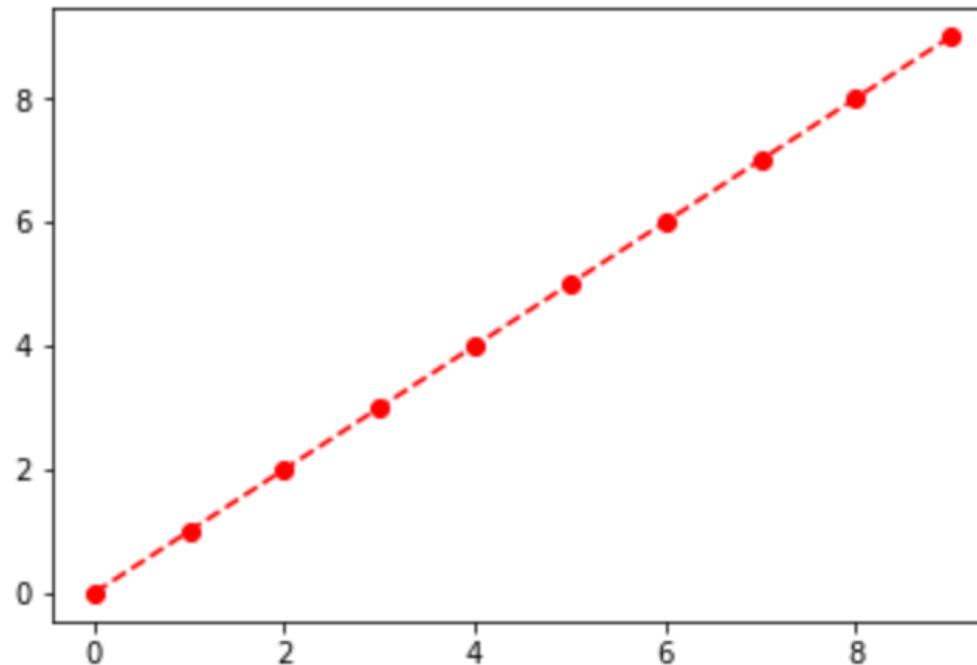
File Edit View Insert Cell Kernel Widgets Help

In [3]: `plt.plot(np.arange(10), linestyle='--', color='red', marker='o')`

My
...
marker
linestyle
'o--'

In [5]: `plt.plot(np.arange(10), 'ro--')`

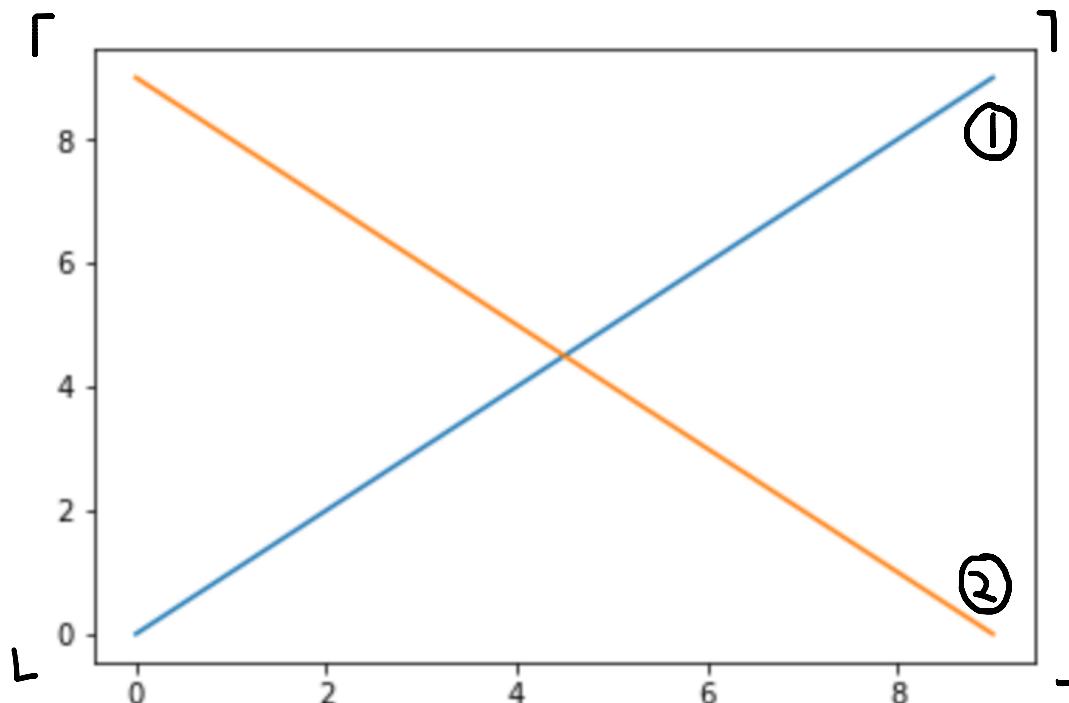
Out[5]: [`<matplotlib.lines.Line2D at 0x69b9908>`] 'O'



In []:

```
In [6]: plt.plot(np.arange(10))
plt.plot(np.arange(10)[::-1])
```

```
Out[6]: [
```



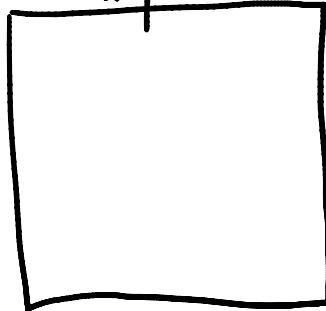
```
In [ ]:
```

Subplots

plt.plot()

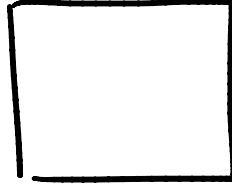
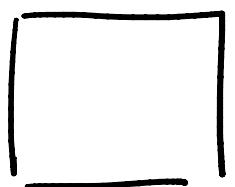
↳ plt.figure()

 └ subplot ┌

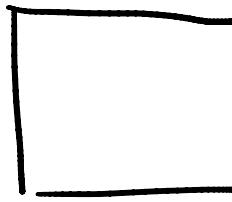
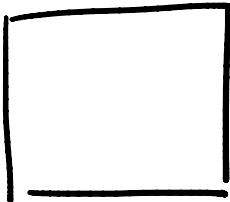


└ ┌

 └ plt.figure()
 └ fig.add_subplots() ┌



 └ ┌



└

File Edit View Insert Cell Kernel Widgets Help

In [7]: `fig = plt.figure()`

<matplotlib.figure.Figure at 0x69d16d8>

Figure | X |
F

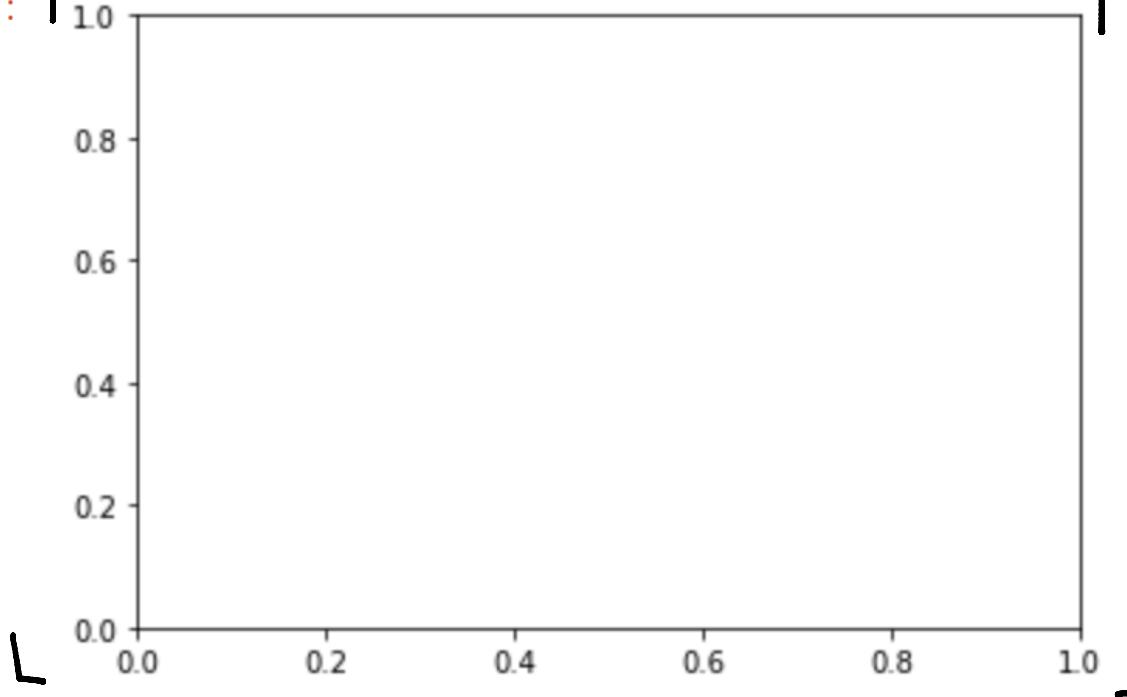
In [8]: `subplot = fig.add_subplot(1, 1, 1)`

| X |

In [9]: `fig`

Subplot

Out[9]:



File Edit View Insert Cell Kernel Widgets Help

In [8]: subplot = fig.add_subplot(1, 1, 1)

Before

In [9]: fig

plt.plot()

After

fig=plt.figure()

Subplot=fig.add_subplot()

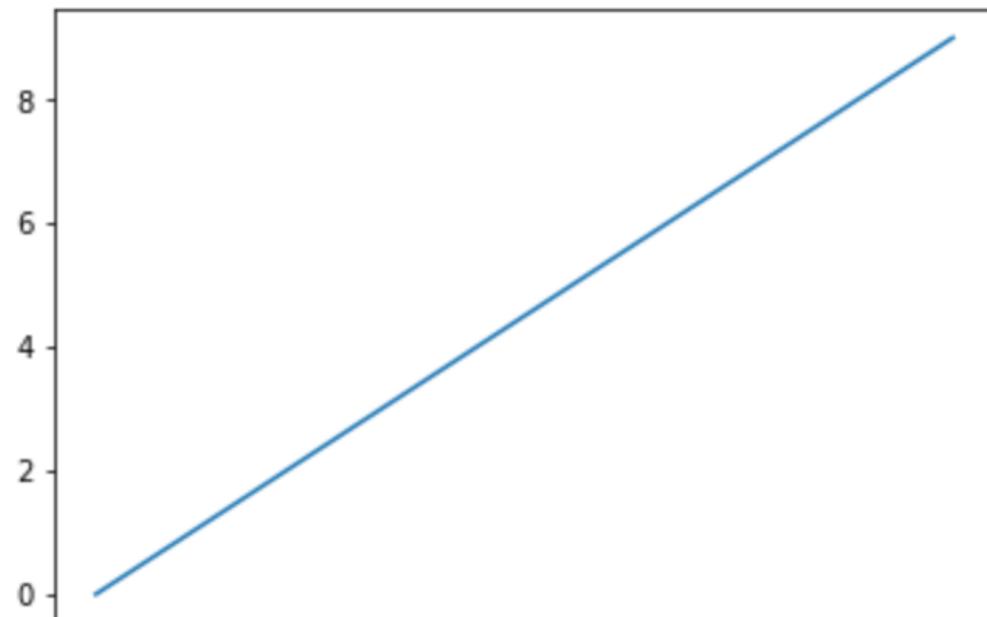
Subplot.plot()

In [10]: subplot.plot(np.arange(10))

Out[10]: [`<matplotlib.lines.Line2D at 0x69df940>`]

In [11]: fig

Out[11]:



File Edit View Insert Cell Kernel Widgets Help

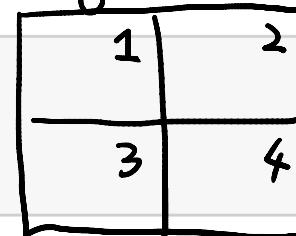
In [12]: `fig = plt.figure()`

<matplotlib.figure.Figure at 0x82d4f98>

"subplot"

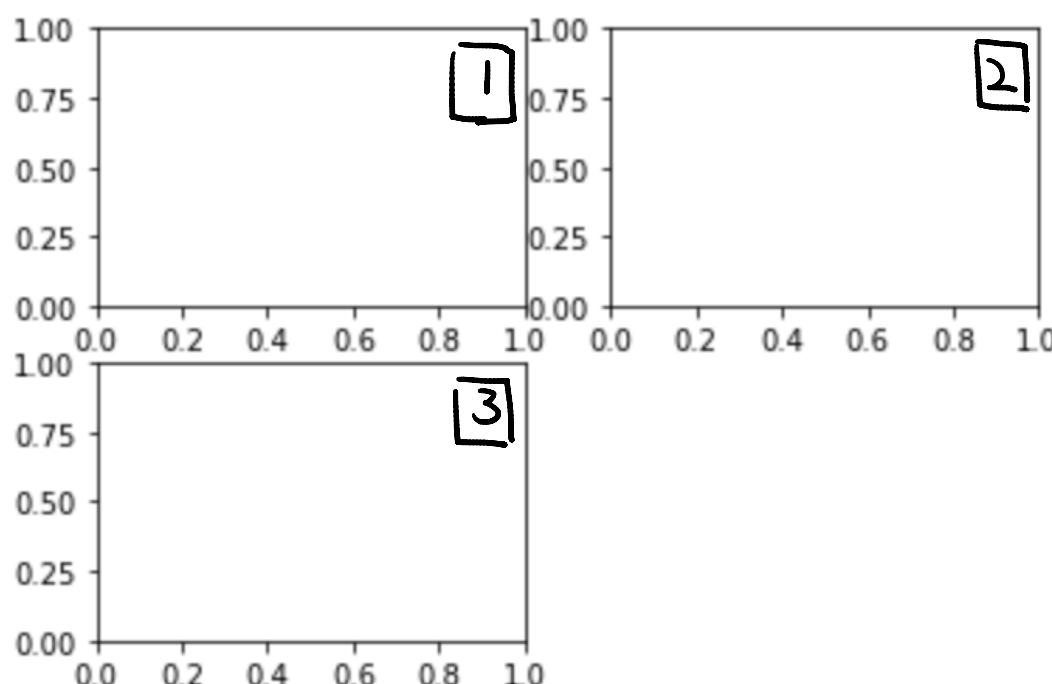
In [13]: `ax1 = fig.add_subplot(2, 2, 1)`
`ax2 = fig.add_subplot(2, 2, 2)`
`ax3 = fig.add_subplot(2, 2, 3)`

Figure



In [14]: `fig`

Out [14]:

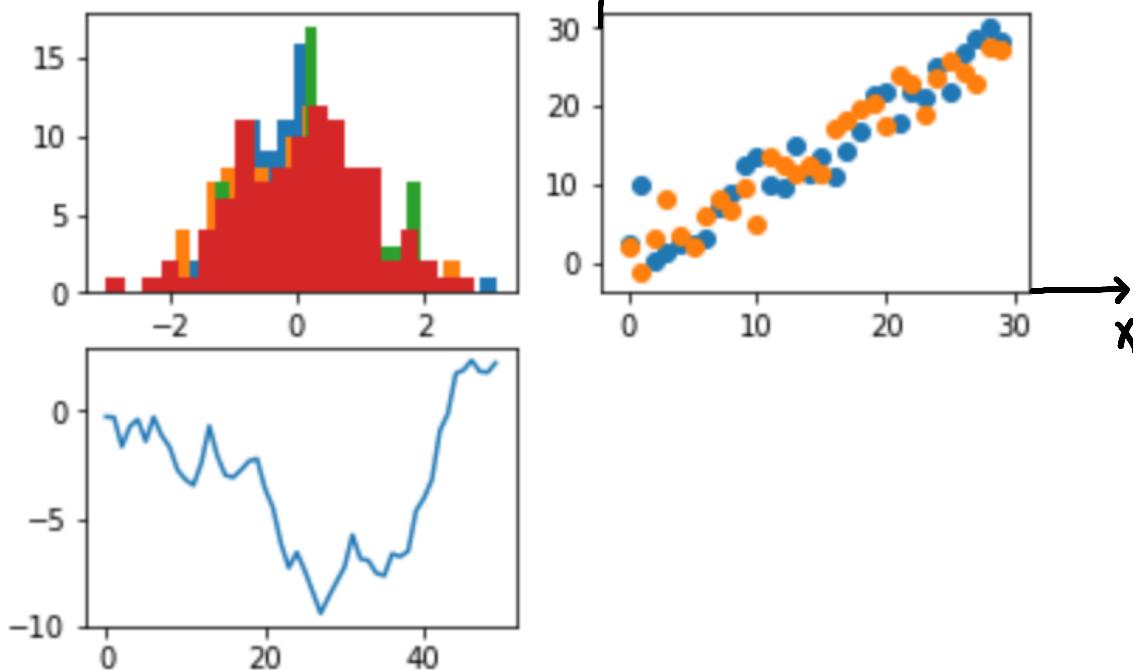


ax.scatter(x, y)

In [18]:
ax1.hist(np.random.randn(100), bins=20)
ax2.scatter(np.arange(30), np.arange(30)+3*np.random.randn(30))
ax3.plot(np.random.randn(50).cumsum())

fig

Out[18]:



In []:

File Edit View Insert Cell Kernel Widgets Help

fig

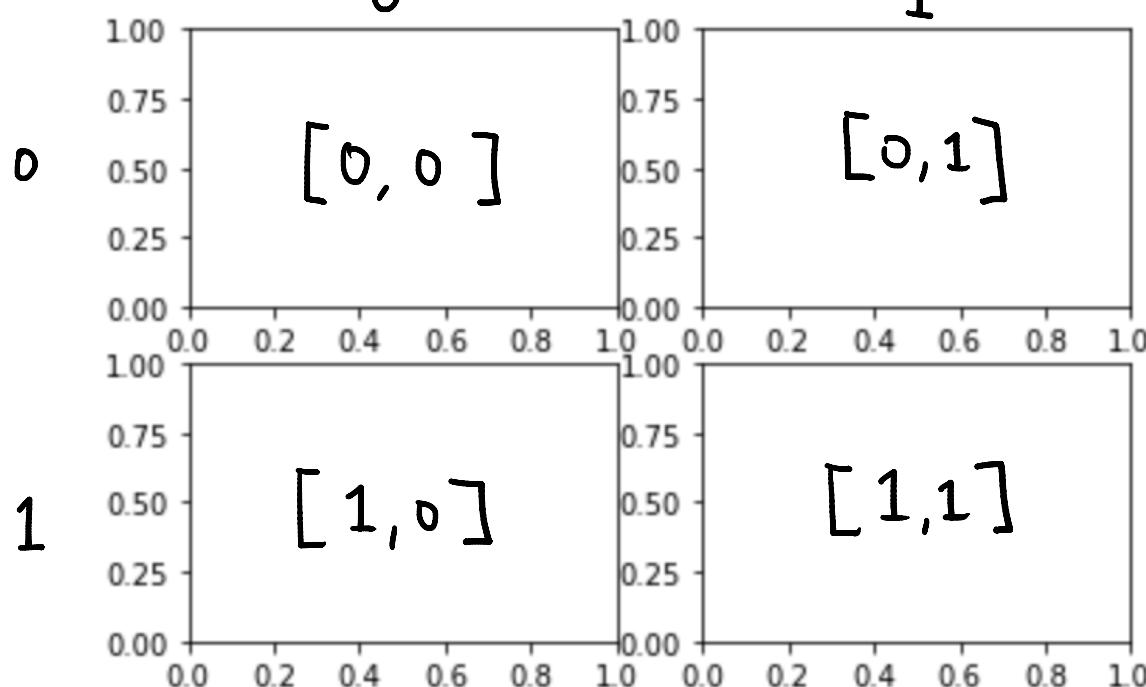
fig=plt.figure()

...

In [19]: fig, axes = plt.subplots(2, 2)



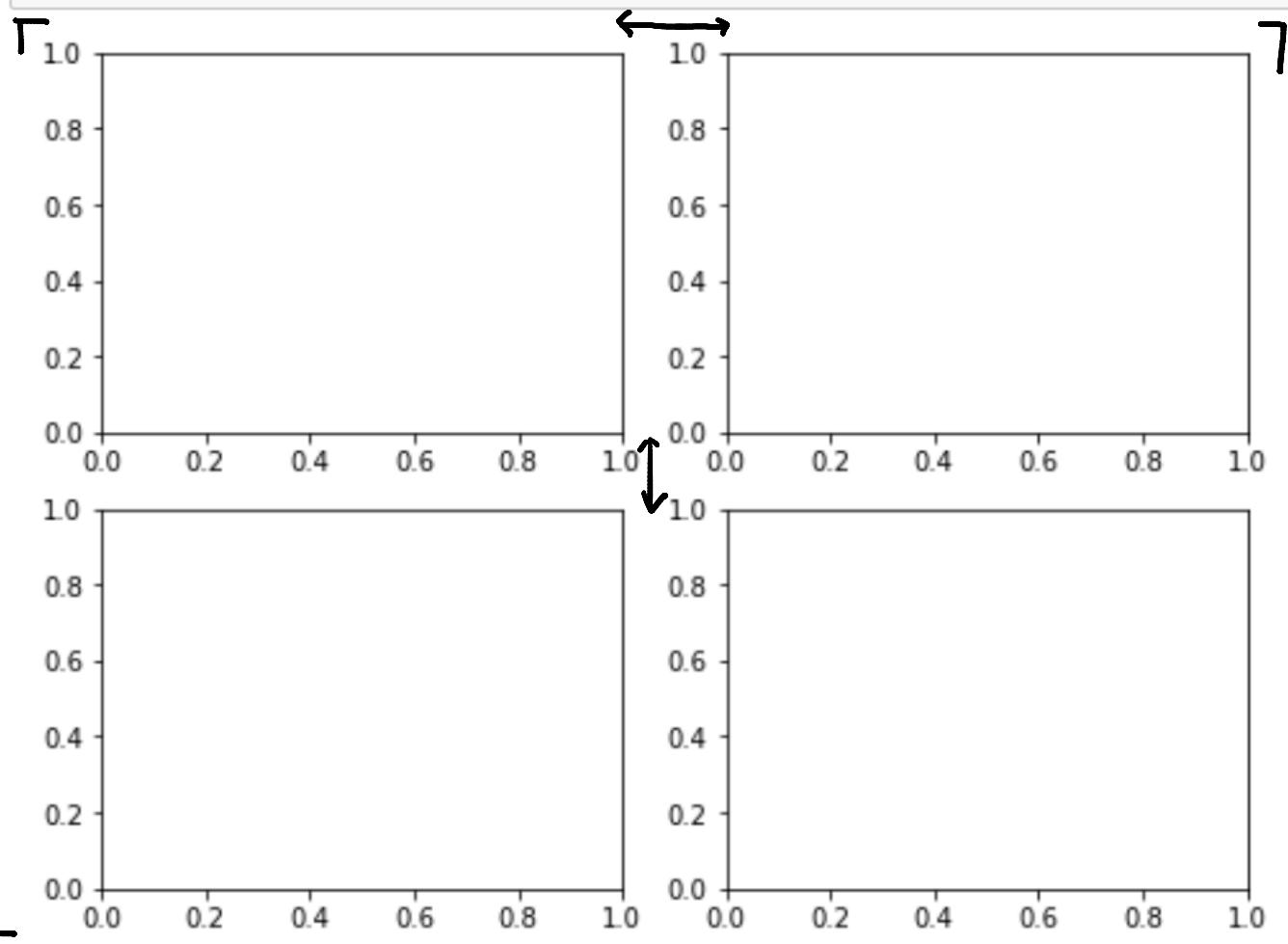
:
1



In []:

```
In [20]: fig, axes = plt.subplots(2, 2, figsize=(8,6))
```

6inch



```
In [ ]:
```

8 inch

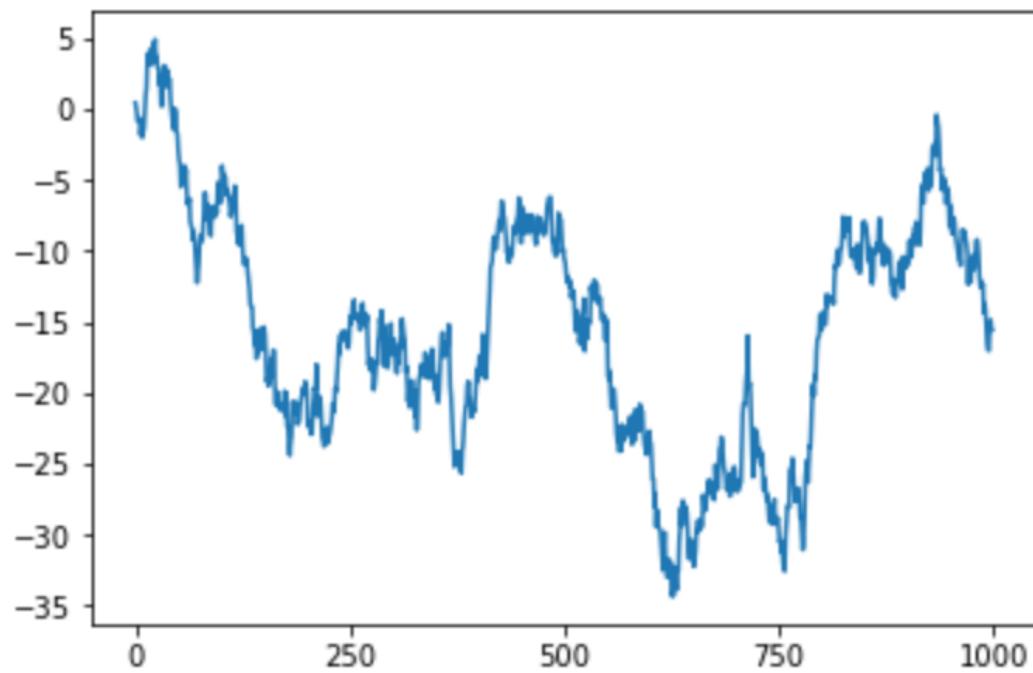
File Edit View Insert Cell Kernel Widgets Help

```
ax.plot(np.random.randn(1000).cumsum())
```

...

In [23]: `ax.set_xticks([0, 250, 500, 750, 1000])
fig`

Out[23]:



set_xticks([



In []:

In [24]: `import pandas as pd`

CSV "A, B , C"
DataFrame

In [26]: `%pwd`

Out[26]: 'C:\Users\student\Documents\pydeep'

In [27]: `pd.read_csv('data/iris.data', header=None)`

Out[27]:

	0	1	2	3	4
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.6	3.1	1.5	0.4	Iris-setosa
10	5.0	3.6	1.4	0.2	Iris-setosa
11	5.5	4.2	1.4	0.2	Iris-setosa
12	4.9	3.5	1.4	0.2	Iris-setosa
13	4.8	3.4	1.5	0.2	Iris-setosa
14	4.8	3.0	1.4	0.2	Iris-setosa
15	5.1	3.8	1.5	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.1	3.8	1.5	0.3	Iris-setosa
18	5.4	3.9	1.5	0.4	Iris-setosa
19	5.1	3.5	1.4	0.2	Iris-setosa
20	4.5	3.2	1.4	0.2	Iris-setosa
21	4.5	3.2	1.4	0.2	Iris-setosa
22	4.6	3.4	1.4	0.2	Iris-setosa
23	5.0	3.4	1.5	0.2	Iris-setosa
24	5.4	3.9	1.5	0.4	Iris-setosa
25	5.0	3.4	1.4	0.2	Iris-setosa
26	4.9	3.1	1.4	0.2	Iris-setosa
27	4.7	3.2	1.3	0.2	Iris-setosa
28	4.6	3.1	1.5	0.2	Iris-setosa
29	5.0	3.6	1.4	0.2	Iris-setosa
30	5.5	4.2	1.4	0.2	Iris-setosa
31	4.9	3.5	1.4	0.2	Iris-setosa
32	4.8	3.4	1.5	0.2	Iris-setosa
33	4.8	3.0	1.4	0.2	Iris-setosa
34	5.1	3.8	1.5	0.2	Iris-setosa
35	5.7	4.4	1.5	0.4	Iris-setosa
36	5.1	3.8	1.5	0.3	Iris-setosa
37	5.4	3.9	1.5	0.4	Iris-setosa
38	5.1	3.5	1.4	0.2	Iris-setosa
39	4.5	3.2	1.4	0.2	Iris-setosa
40	4.5	3.2	1.4	0.2	Iris-setosa
41	4.6	3.4	1.4	0.2	Iris-setosa
42	5.0	3.4	1.5	0.2	Iris-setosa
43	5.4	3.9	1.5	0.4	Iris-setosa
44	5.0	3.4	1.4	0.2	Iris-setosa
45	4.9	3.1	1.4	0.2	Iris-setosa
46	4.7	3.2	1.3	0.2	Iris-setosa
47	4.6	3.1	1.5	0.2	Iris-setosa
48	5.0	3.6	1.4	0.2	Iris-setosa
49	5.5	4.2	1.4	0.2	Iris-setosa
50	4.9	3.5	1.4	0.2	Iris-setosa
51	4.8	3.4	1.5	0.2	Iris-setosa
52	4.8	3.0	1.4	0.2	Iris-setosa
53	5.1	3.8	1.5	0.2	Iris-setosa
54	5.7	4.4	1.5	0.4	Iris-setosa
55	5.1	3.8	1.5	0.3	Iris-setosa
56	5.4	3.9	1.5	0.4	Iris-setosa
57	5.1	3.5	1.4	0.2	Iris-setosa
58	4.5	3.2	1.4	0.2	Iris-setosa
59	4.5	3.2	1.4	0.2	Iris-setosa
60	4.6	3.4	1.4	0.2	Iris-setosa
61	5.0	3.4	1.5	0.2	Iris-setosa
62	5.4	3.9	1.5	0.4	Iris-setosa
63	5.0	3.4	1.4	0.2	Iris-setosa
64	4.9	3.1	1.4	0.2	Iris-setosa
65	4.7	3.2	1.3	0.2	Iris-setosa
66	4.6	3.1	1.5	0.2	Iris-setosa
67	5.0	3.6	1.4	0.2	Iris-setosa
68	5.5	4.2	1.4	0.2	Iris-setosa
69	4.9	3.5	1.4	0.2	Iris-setosa
70	4.8	3.4	1.5	0.2	Iris-setosa
71	4.8	3.0	1.4	0.2	Iris-setosa
72	5.1	3.8	1.5	0.2	Iris-setosa
73	5.7	4.4	1.5	0.4	Iris-setosa
74	5.1	3.8	1.5	0.3	Iris-setosa
75	5.4	3.9	1.5	0.4	Iris-setosa
76	5.1	3.5	1.4	0.2	Iris-setosa
77	4.5	3.2	1.4	0.2	Iris-setosa
78	4.5	3.2	1.4	0.2	Iris-setosa
79	4.6	3.4	1.4	0.2	Iris-setosa
80	5.0	3.4	1.5	0.2	Iris-setosa
81	5.4	3.9	1.5	0.4	Iris-setosa
82	5.0	3.4	1.4	0.2	Iris-setosa
83	4.9	3.1	1.4	0.2	Iris-setosa
84	4.7	3.2	1.3	0.2	Iris-setosa
85	4.6	3.1	1.5	0.2	Iris-setosa
86	5.0	3.6	1.4	0.2	Iris-setosa
87	5.5	4.2	1.4	0.2	Iris-setosa
88	4.9	3.5	1.4	0.2	Iris-setosa
89	4.8	3.4	1.5	0.2	Iris-setosa
90	4.8	3.0	1.4	0.2	Iris-setosa
91	5.1	3.8	1.5	0.2	Iris-setosa
92	5.7	4.4	1.5	0.4	Iris-setosa
93	5.1	3.8	1.5	0.3	Iris-setosa
94	5.4	3.9	1.5	0.4	Iris-setosa
95	5.1	3.5	1.4	0.2	Iris-setosa
96	4.5	3.2	1.4	0.2	Iris-setosa
97	4.5	3.2	1.4	0.2	Iris-setosa
98	4.6	3.4	1.4	0.2	Iris-setosa
99	5.0	3.4	1.5	0.2	Iris-setosa
100	5.4	3.9	1.5	0.4	Iris-setosa
101	5.0	3.4	1.4	0.2	Iris-setosa
102	4.9	3.1	1.4	0.2	Iris-setosa
103	4.7	3.2	1.3	0.2	Iris-setosa
104	4.6	3.1	1.5	0.2	Iris-setosa
105	5.0	3.6	1.4	0.2	Iris-setosa
106	5.5	4.2	1.4	0.2	Iris-setosa
107	4.9	3.5	1.4	0.2	Iris-setosa
108	4.8	3.4	1.5	0.2	Iris-setosa
109	4.8	3.0	1.4	0.2	Iris-setosa
110	5.1	3.8	1.5	0.2	Iris-setosa
111	5.7	4.4	1.5	0.4	Iris-setosa
112	5.1	3.8	1.5	0.3	Iris-setosa
113	5.4	3.9	1.5	0.4	Iris-setosa
114	5.1	3.5	1.4	0.2	Iris-setosa
115	4.5	3.2	1.4	0.2	Iris-setosa
116	4.5	3.2	1.4	0.2	Iris-setosa
117	4.6	3.4	1.4	0.2	Iris-setosa
118	5.0	3.4	1.5	0.2	Iris-setosa
119	5.4	3.9	1.5	0.4	Iris-setosa
120	5.0	3.4	1.4	0.2	Iris-setosa
121	4.9	3.1	1.4	0.2	Iris-setosa
122	4.7	3.2	1.3	0.2	Iris-setosa
123	4.6	3.1	1.5	0.2	Iris-setosa
124	5.0	3.6	1.4	0.2	Iris-setosa
125	5.5	4.2	1.4	0.2	Iris-setosa
126	4.9	3.5	1.4	0.2	Iris-setosa
127	4.8	3.4	1.5	0.2	Iris-setosa
128	4.8	3.0	1.4	0.2	Iris-setosa
129	5.1	3.8	1.5	0.2	Iris-setosa
130	5.7	4.4	1.5	0.4	Iris-setosa
131	5.1	3.8	1.5	0.3	Iris-setosa
132	5.4	3.9	1.5	0.4	Iris-setosa
133	5.1	3.5	1.4	0.2	Iris-setosa
134	4.5	3.2	1.4	0.2	Iris-setosa
135	4.5	3.2	1.4	0.2	Iris-setosa
136	4.6	3.4	1.4	0.2	Iris-setosa
137	5.0	3.4	1.5	0.2	Iris-setosa
138	5.4	3.9	1.5	0.4	Iris-setosa
139	5.0	3.4	1.4	0.2	Iris-setosa
140	4.9	3.1	1.4	0.2	Iris-setosa
141	4.7	3.2	1.3	0.2	Iris-setosa
142	4.6	3.1	1.5	0.2	Iris-setosa
143	5.0	3.6	1.4	0.2	Iris-setosa
144	5.5	4.2	1.4	0.2	Iris-setosa
145	4.9	3.5	1.4	0.2	Iris-setosa
146	4.8	3.4	1.5	0.2	Iris-setosa
147	4.8	3.0	1.4	0.2	Iris-setosa
148	5.1	3.8	1.5	0.2	Iris-setosa
149	5.7	4.4	1.5	0.4	Iris-setosa
150	5.1	3.8	1.5	0.3	Iris-setosa
151	5.4	3.9	1.5	0.4	Iris-setosa
152	5.1	3.5	1.4	0.2	Iris-setosa
153	4.5	3.2	1.4	0.2	Iris-setosa
154	4.5	3.2	1.4	0.2	Iris-setosa
155	4.6	3.4	1.4	0.2	Iris-setosa
156	5.0	3.4	1.5	0.2	Iris-setosa
157	5.4	3.9	1.5	0.4	Iris-setosa
158	5.0	3.4	1.4	0.2	Iris-setosa
159	4.9	3.1	1.4	0.2	Iris-setosa
160	4.7	3.2	1.3	0.2	Iris-setosa
161	4.6	3.1	1.5	0.2	Iris-setosa
162	5.0	3.6	1.4	0.2	Iris-setosa
163	5.5	4.2	1.4	0.2	Iris-setosa
164	4.9	3.5	1.4	0.2	Iris-setosa
165	4.8	3.4	1.5	0.2	Iris-setosa
166	4.8	3.0	1.4	0.2	Iris-setosa
167	5.1	3.8	1.5	0.2	Iris-setosa
168	5.7	4.4	1.5	0.4	Iris-setosa
169	5.1	3.8	1.5	0.3	Iris-setosa
170	5.4	3.9	1.5	0.4	Iris-setosa
171	5.1	3.5	1.4	0.2	Iris-setosa
172	4.5	3.2	1.4	0.2	Iris-setosa
173	4.5	3.2	1.4	0.2	Iris-setosa
174	4.6	3.4	1.4	0.2	Iris-setosa
175	5.0	3.4	1.5	0.2	Iris-setosa
176	5.4	3.9	1.5	0.4	Iris-setosa
177	5.0	3.4	1.4	0.2	Iris-setosa
178	4.9	3.1	1.4	0.2	Iris-setosa
179	4.7	3.2	1.3	0.2	Iris-setosa
180	4.6	3.1	1.5	0.2	Iris-setosa
181	5.0	3.6	1.4	0.2	Iris-setosa
182	5.5	4.2	1.4	0.2	Iris-setosa
183	4.9	3.5	1.4	0.2	Iris-setosa
184	4.8	3.4	1.5	0.2	Iris-setosa
185	4.8	3.0	1.4	0.2	Iris-setosa
186	5.1	3.8	1.5	0.2	Iris-setosa
187	5.7	4.4	1.5	0.4	Iris-setosa
188	5.1	3.8	1.5	0.3	Iris-setosa
189	5.4	3.9	1.5	0.4	Iris-setosa
190	5.1	3.5	1.4	0.2	Iris-setosa
191	4.5	3.2	1.4	0.2	Iris-setosa
192	4.5	3.2	1.4	0.2	Iris-setosa
193	4.6	3.4	1.4	0.2	Iris-setosa
194	5.0	3.4	1.5	0.2	Iris-setosa
195	5.4	3.9	1.5	0.4	Iris-setosa
196	5.0	3.4	1.4	0.2	Iris-setosa
197	4.9	3.1	1.4	0.2	Iris-setosa
198	4.7	3.2	1.3	0.2	Iris-setosa
199	4.6	3.1	1.5	0.2	Iris-setosa
200	5.0	3.6	1.4	0.2	Iris-setosa
201	5.5	4.2	1.4	0.2	Iris-setosa
202	4.9	3.5	1.4	0.2	Iris-setosa
203	4.8	3.4	1.5	0.2	Iris-setosa
204	4.8	3.0	1.4	0.2	Iris-setosa
205	5.1	3.8	1.5	0.2	Iris-setosa
206	5.7	4.4	1.5	0.4	Iris-setosa
207	5.1	3.8	1.5	0.3	Iris-setosa
208	5.4	3.9	1.5	0.4	Iris-setosa
209	5.1	3.5	1.4	0.2	Iris-setosa
210	4.5	3.2	1.4	0.2	Iris-setosa
211	4.5	3.2	1.4	0.2	Iris-setosa
212	4.6	3.4	1.4	0.2	Iris-setosa
213	5.0	3.4	1.5	0.2	Iris-setosa
214	5.4	3.9	1.5	0.4	Iris-setosa
215	5.0	3.4	1.4	0.2	Iris-setosa
216	4.9	3.1	1.4	0.2	Iris-setosa
217	4.7	3.2	1.3	0.2	Iris-setosa
218	4.6	3.1	1.5	0.2	Iris-setosa
219	5.0	3.6	1.4	0.2	Iris-setosa
220	5.5	4.2	1.4	0.2	Iris-setosa
221	4.9	3.5	1.4	0.2	Iris-setosa
222	4.8	3.4	1.5	0.2	Iris-setosa
223	4.8	3.0	1.4	0.2	Iris-setosa
224	5.1	3.			

File Edit View Insert Cell Kernel Widgets Help

In [28]: `frame = pd.read_csv('data/iris.data', header=None)`

In [29]: `type(frame)`

Out[29]: `pandas.core.frame.DataFrame`

In [30]: `frame[:5]`

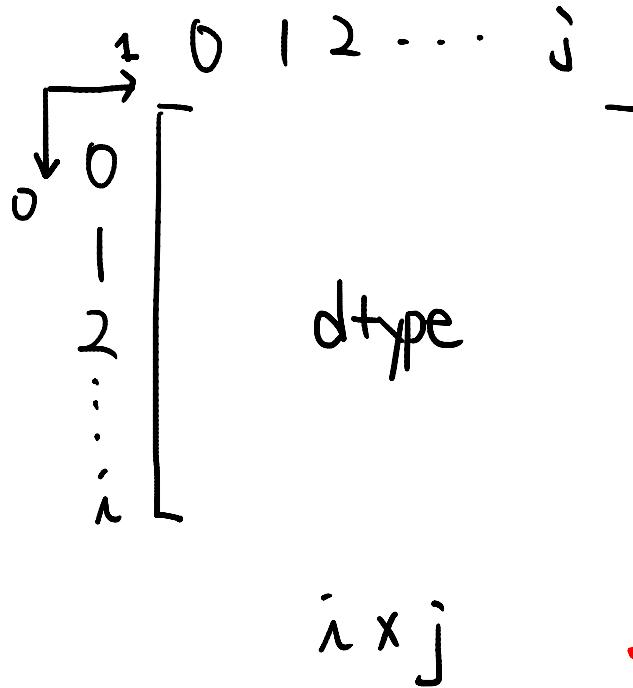
Out[30]:



	0	1	2	3	4
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In []:

ndarray



+
2010
+
~~tx~~ pandas

Diagram illustrating a Pandas DataFrame:

The DataFrame has columns labeled A, B, and C. The rows are labeled 0, 1, 2, 3.

Annotations:

- Red boxes highlight specific elements:
 - Row 0: Column A (labeled 'A')
 - Row 1: Column B (labeled 'B')
 - Row 2: Column C (labeled 'C')
 - Row 3: Column A (labeled 'A')
 - Column 0: Row 0 (labeled '0')
 - Column 1: Row 1 (labeled '1')
 - Column 2: Row 2 (labeled '2')
 - Column 3: Row 3 (labeled '3')
- Red arrows point to the labels: "int", "float", and "Object".
- A red bracket labeled "columns" spans across the top of the columns.
- A red bracket labeled "ndarray" spans across the top of the first three rows.
- Red arrows point from the labels "index" and "ndarray" to their respective components.

0	1	4	C
A	1	4.5	A
B	2	6.7	B
C	3	8.9	C

색인/열 선택

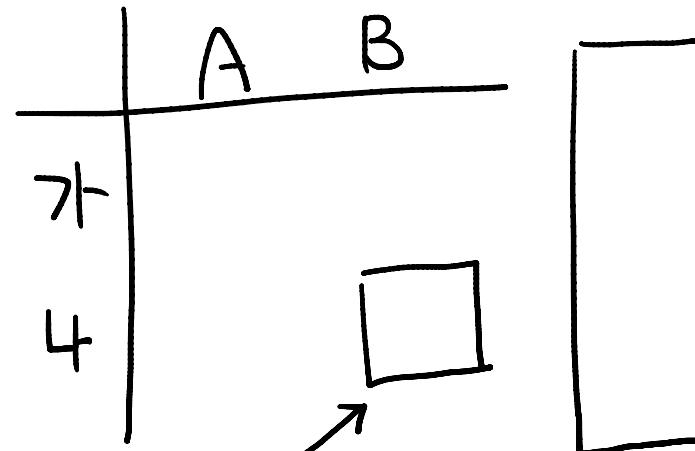
$\alpha \in \{0, 1\}$ ($n > 0$)

	0	1
0	0,0	0,1
1	1,0	1,1

$[0^{\text{축}}, 1^{\text{축}}, \dots]$

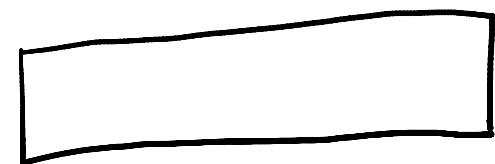
"열선택"

DataFrame ($n=2$) ['A']

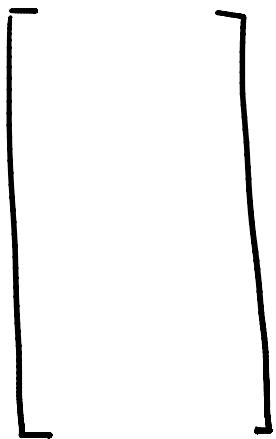


.loc['4', 'B'] .loc['가']

색인 열



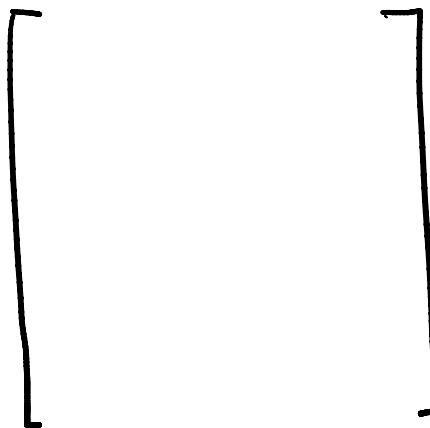
벡터



ndarray

Series

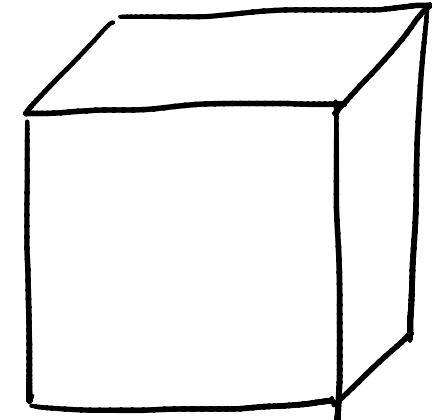
행렬



ndarray

DataFrame

텐서 ($n > 2$)



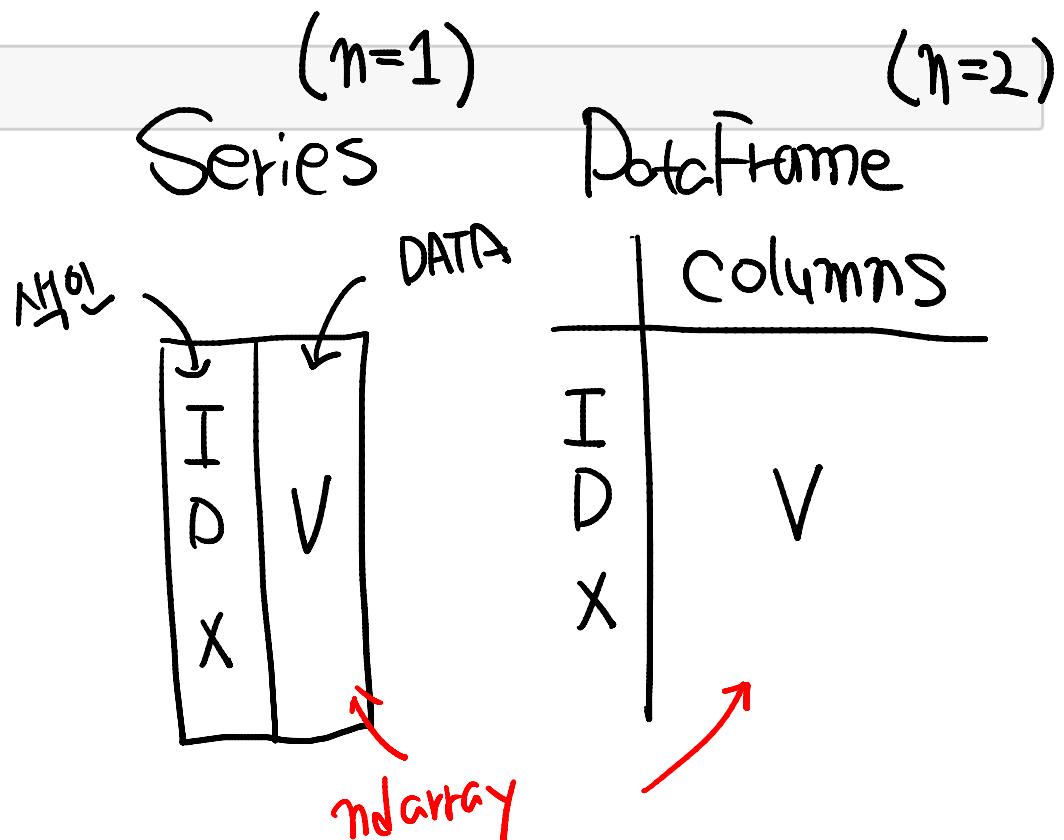
ndarray

```
In [34]: type(frame[0])
```

```
Out[34]: pandas.core.series.Series
```

```
In [35]: frame[0]
```

```
Out[35]: 0    5.1  
1    4.9  
2    4.7  
3    4.6  
4    5.0  
5    5.4  
6    4.6  
7    5.0  
8    4.4  
9    4.9  
10   5.4  
11   4.8  
12   4.8  
13   4.3  
14   5.8  
15   5.7  
16   5.4  
17   5.1  
18   5.7
```



File Edit View Insert Cell Kernel Widgets Help

Out [29]: pandas.core.frame.DataFrame

In [30]: frame[:5] "편의상" 범위선택은 기본적으로 색인

Out [30]:

	0	1	2	3	4
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

frame.iloc[:, :5]

In [34]: type(frame[0])

Out [34]: pandas.core.series.Series

In [35]: frame[0]

...

In [36]: frame[0].index

File Edit View Insert Cell Kernel Widgets Help

In [36]: frame[0].index

Out[36]: RangeIndex(start=0, stop=150, step=1)

In [38]: frame[0].values[:5]

Out[38]: array([5.1, 4.9, 4.7, 4.6, 5.])

In [39]: frame.index

Out[39]: RangeIndex(start=0, stop=150, step=1)

In [41]: frame.values[:5]

Out[41]: array([[5.1, 3.5, 1.4, 0.2, 'Iris-setosa'],
 [4.9, 3.0, 1.4, 0.2, 'Iris-setosa'],
 [4.7, 3.2, 1.3, 0.2, 'Iris-setosa'],
 [4.6, 3.1, 1.5, 0.2, 'Iris-setosa'],
 [5.0, 3.6, 1.4, 0.2, 'Iris-setosa']], dtype=object)

ndarray →

↓

In []:

File Edit View Insert Cell Kernel Widgets Help

In [46]: `frame['x1']`

...

In [47]: `frame[['x4', 'x1']]`

...

In [48]: `frame.loc[0]`

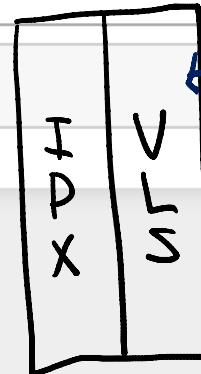
Out [48]:

```
x1      5.1  
x2      3.5  
x3      1.4  
x4      0.2  
label   Iris-setosa  
Name: 0, dtype: object
```

	X1	X2	X3	X4	label
0	5.1	3.5	1.4	0.2	ST

Series

In []:



In [48]: frame.loc[0]

DfFrame.loc[색인조건, 열조건]

In [49]: frame.loc[[1,0]]

ndarray [0축, 1축]

In [50]: frame.loc[[0,1], ['x1', 'x2', 'label']]

정수
색인
라벨

Out [50]: 열(1)

	x1	x2	label
0	5.1	3.5	Iris-setosa
1	4.9	3.0	Iris-setosa

In []:

ndarray + & 라벨 슬라이스 ... [0:3] 0 ≤ 색인 < 3

In [64]: NC['이호준': '나성범']

['A' : 'G'] 'A' ≤ 'G'

Out [64]:

팀명	경기	타석	타수	안타	홈런	득점	타점	볼넷	삼진	도루	BABIP	타율	출루율
----	----	----	----	----	----	----	----	----	----	----	-------	----	-----

선수명

이호준	NC	126	508	442	123	20	46	87	60	109	2	0.324	0.278	0.362
-----	----	-----	-----	-----	-----	----	----	----	----	-----	---	-------	-------	-------

김종호

김종호	NC	128	546	465	129	0	72	22	57	100	50	0.352	0.277	0.376
-----	----	-----	-----	-----	-----	---	----	----	----	-----	----	-------	-------	-------

나성범

나성범	NC	104	458	404	98	14	55	64	33	95	12	0.279	0.243	0.319
-----	----	-----	-----	-----	----	----	----	----	----	----	----	-------	-------	-------

In []:

In [64]: NC['이호준':'나성범']

.loc[색인, 열]

In [65]: NC.loc[:, '경기':'도루']

Out [65]:

[경기 타석 타수 안타 홈런 득점 타점 볼넷 삼진 도루]

선수명

선수명	경기	타석	타수	안타	홈런	득점	타점	볼넷	삼진	도루
모창민	108	436	395	109	12	57	51	37	68	16
이호준	126	508	442	123	20	46	87	60	109	2
김종호	128	546	465	129	0	72	22	57	100	50
나성범	104	458	404	98	14	55	64	33	95	12
조영훈	120	426	380	107	6	38	39	39	56	4
이현곤	91	161	139	38	0	10	9	16	14	2
이상호	102	138	125	31	0	26	13	9	21	24
강진성	3	3	2	1	0	1	0	1	0	0
조평호	26	86	79	21	2	12	7	6	24	1
반민우	32	48	42	11	0	10	6	5	7	9

File Edit View Insert Cell Kernel Widgets Help

In [67]: 안타홈런 = NC[['안타', '홈런']]

In [68]: 안타홈런.mean()

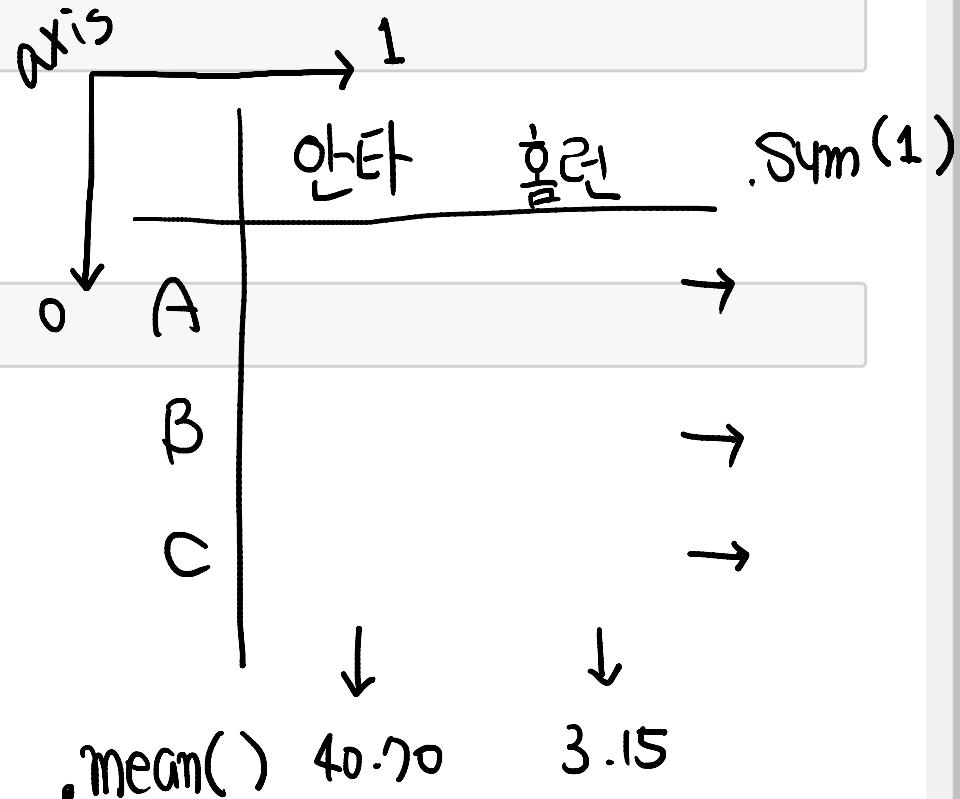
Out[68]:

안타	40.70
홈런	3.15
dtype:	float64

In [69]: 안타홈런.sum(1)

Out[69]:

선수명	
모창민	121
이호준	143
김종호	129
나성범	112
조영훈	113
이현곤	38
이상호	31
강진성	1
조평호	23
박민우	11
박정준	45
이승호	0
마낙길	9
김성욱	1
강구성	0



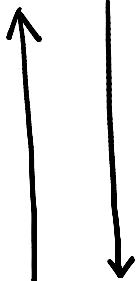
```
In [71]: 합계 = 안타홈런.sum(1).sort_values(ascending=False)
```

Out [71]: 선수명

이호준	143
김종호	129
모창민	121
조영훈	113
나성범	112
지석훈	71
박정준	45
이현곤	38
이상호	31
차화준	25
조평호	23
박민우	11
마낙길	9
김동건	3
강진성	1
김성욱	1
김종찬	1
이승호	0
강구성	0
이창섭	0

dtype: int64

ascending

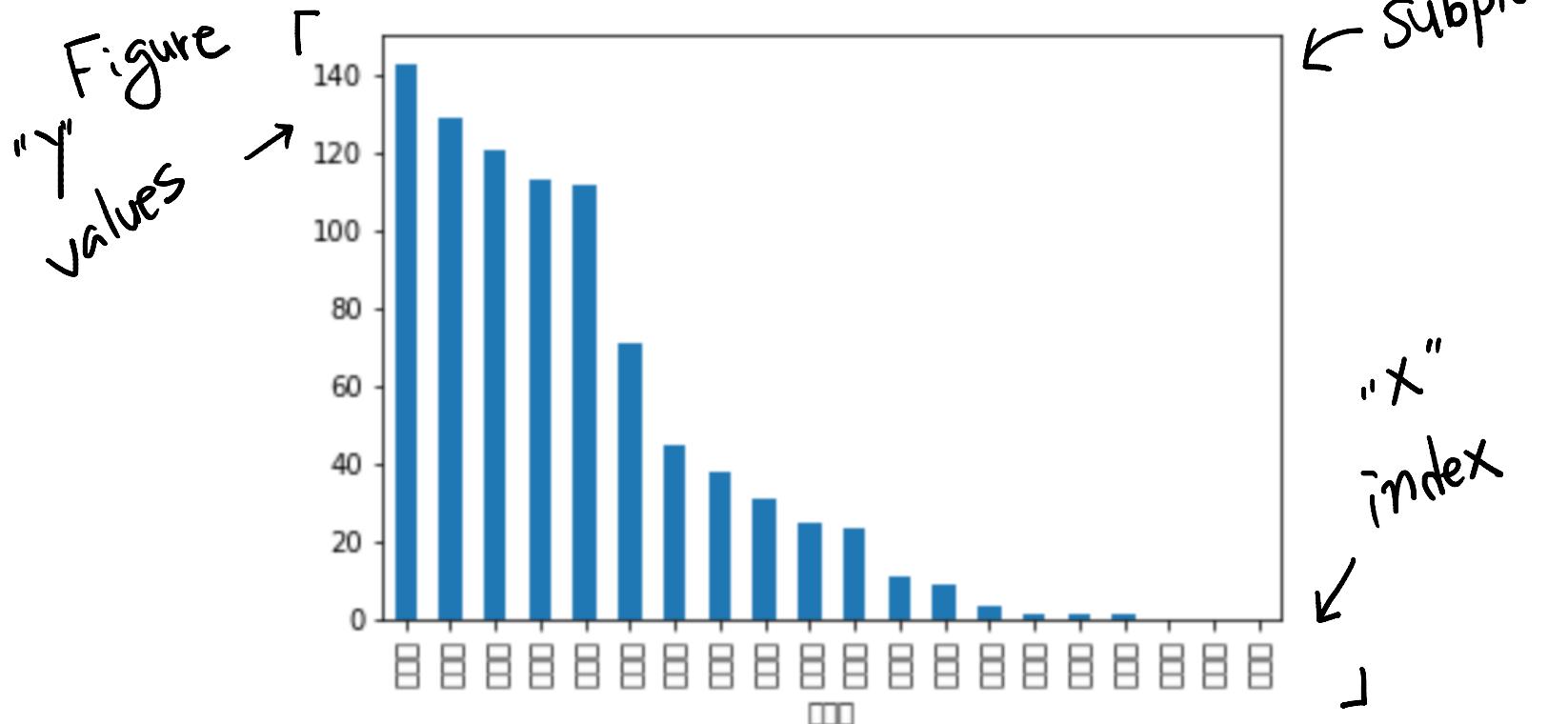


True False

```
In [72]: 합계 = 안타홈런.sum(1).sort_values(ascending=False)
```

```
In [73]: 합계.plot(kind='bar')
```

```
Out[73]: <matplotlib.axes._subplots.AxesSubplot at 0xb59fc50>
```



```
In [ ]:
```

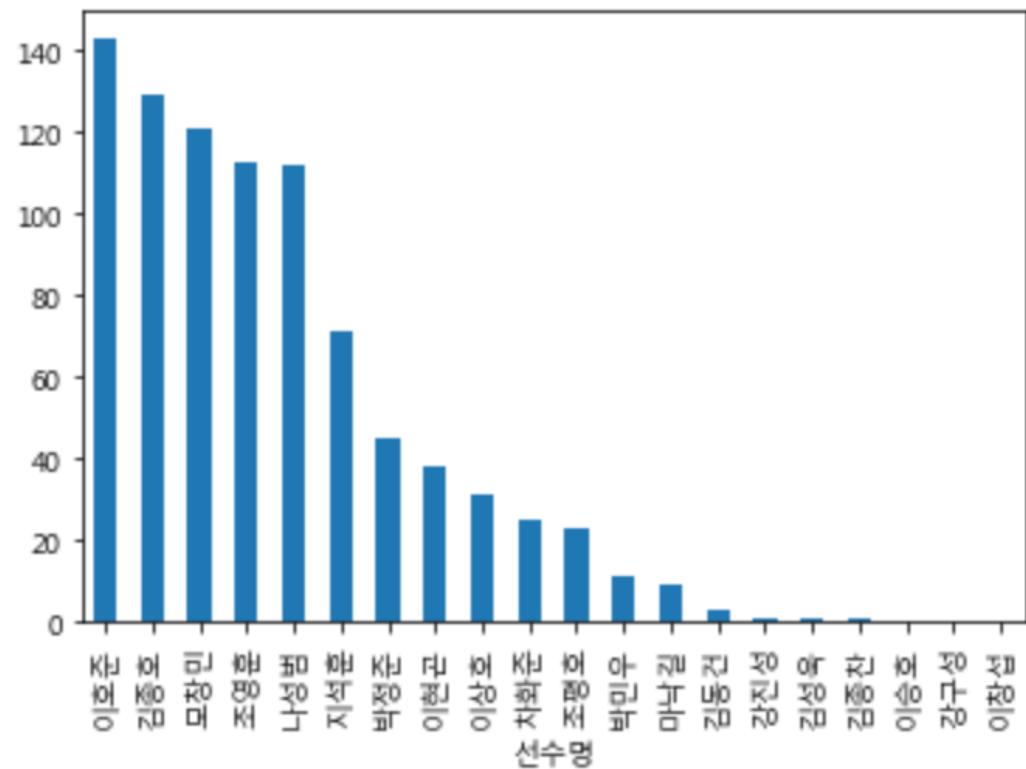
File Edit View Insert Cell Kernel Widgets Help

In [72]: 합계 = 안타율단.sum(1).sort_values(ascending=False)

In [77]: plt.rc('font', family='Malgun Gothic')

In [78]: 합계.plot(kind='bar')

Out[78]: <matplotlib.axes._subplots.AxesSubplot at 0xbb463c8>



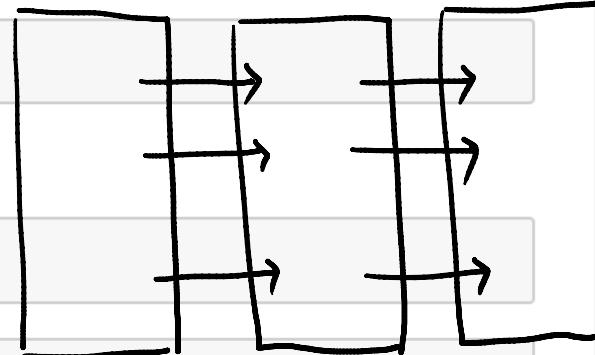
```
In [83]: frame = pd.read_clipboard()
```

Series .Str .astype()

```
In [84]: type(frame)
```

```
Out[84]: pandas.core.frame.DataFrame
```

```
In [87]: frame = frame.set_index('종목명')
```



```
In [90]: frame = frame.drop('토론실', axis=1)
```

.mean()

```
In [99]: frame['현재가'].str.replace(',', '').astype('float').mean()
```

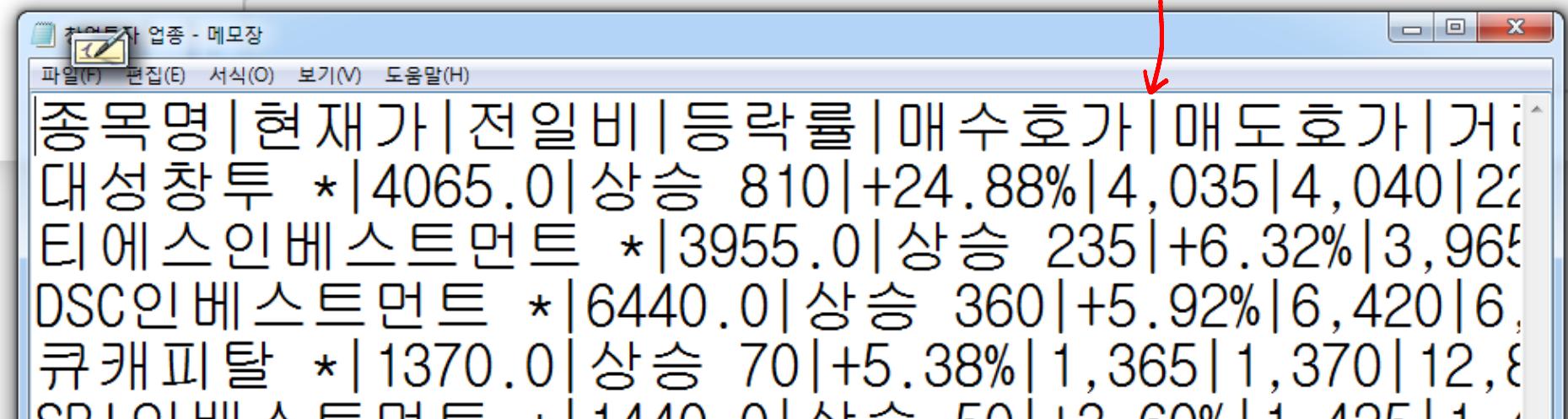
```
Out[99]: 3408.777777777778
```

“각 문자열에 대해”

```
In [ ]:
```

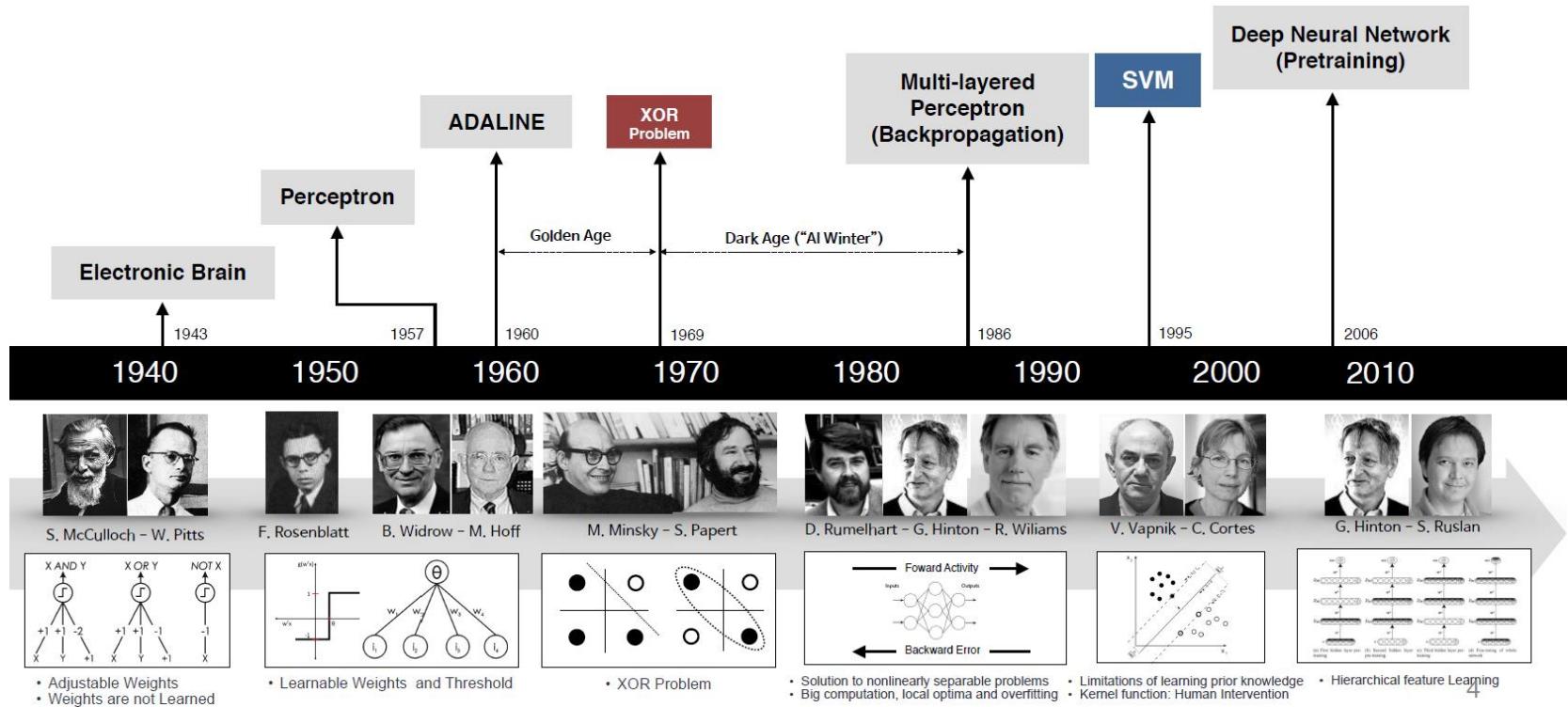
벡터 단위 문자열 처리

```
In [105]: frame.to_csv('창업투자 업종.txt', sep='|')
```

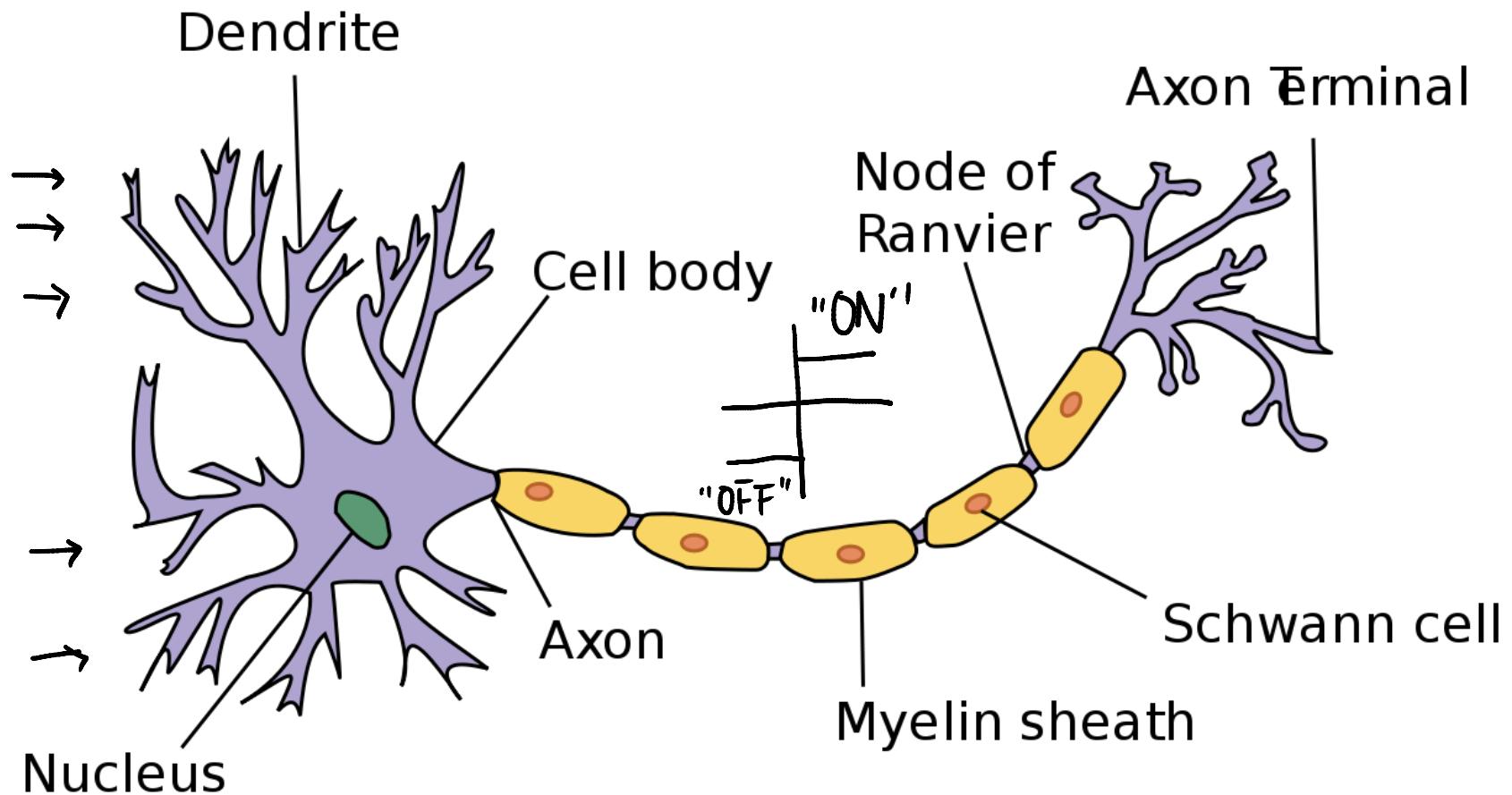


종목명	현재가	전일비	등락률	매수호가	매도호가	거래량
대성창투 *	4065.0	상승 810	+24.88%	4,035	4,040	22,500
티에스인베스트먼트 *	3955.0	상승 235	+6.32%	3,965		
DSC인베스트먼트 *	6440.0	상승 360	+5.92%	6,420	6,450	16,500
큐캐피탈 *	1370.0	상승 70	+5.38%	1,365	1,370	12,800

인공 신경망의 발전사



뉴런



$$\hat{y} = f(x; w)$$

1943 McCulloch-Pitts 누런

입력

가중치

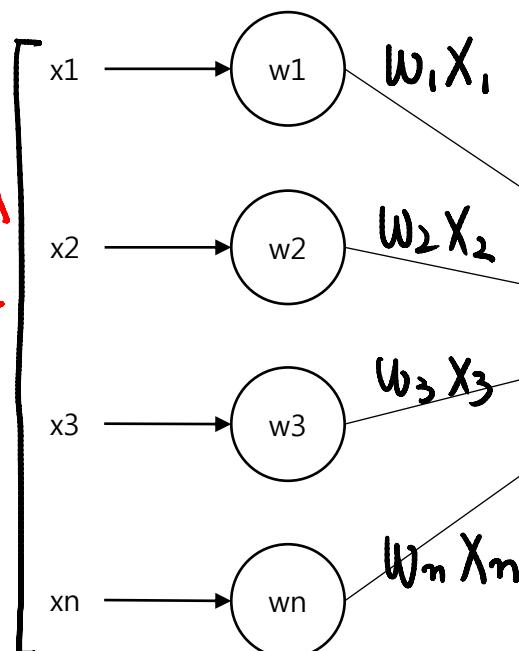
$$w_i x_i$$

$w_i = 0 \rightarrow$ "중요 X"

$w_i > w_j \rightarrow$



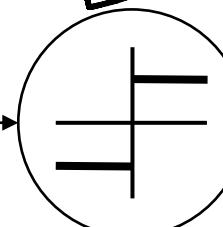
도장
~자원



$$z = w_1 x_1 + w_2 x_2 + \dots + w_n x_n$$

"활성화"

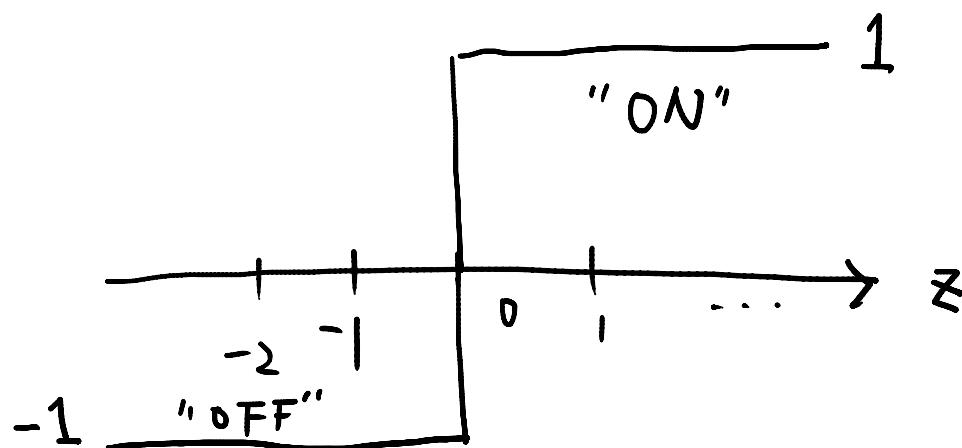
Activation



"ON"

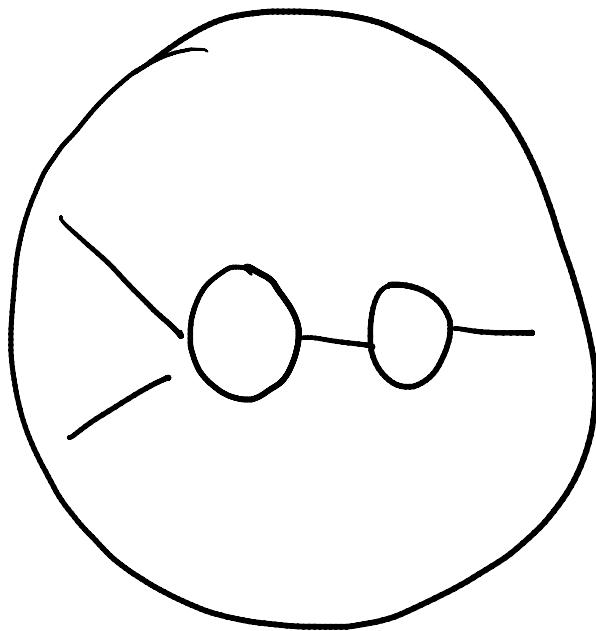
1

"OFF"
-1



20C

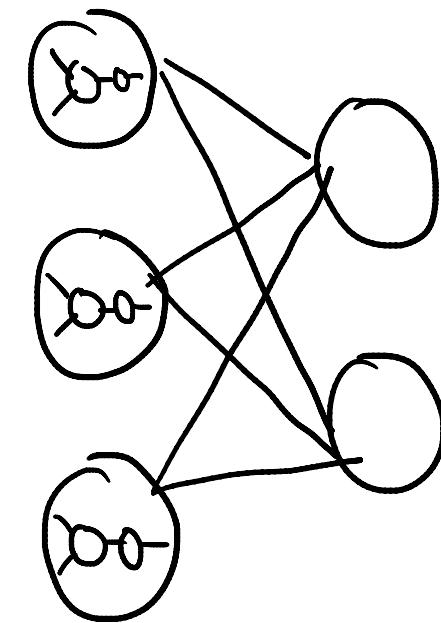
인공뉴런



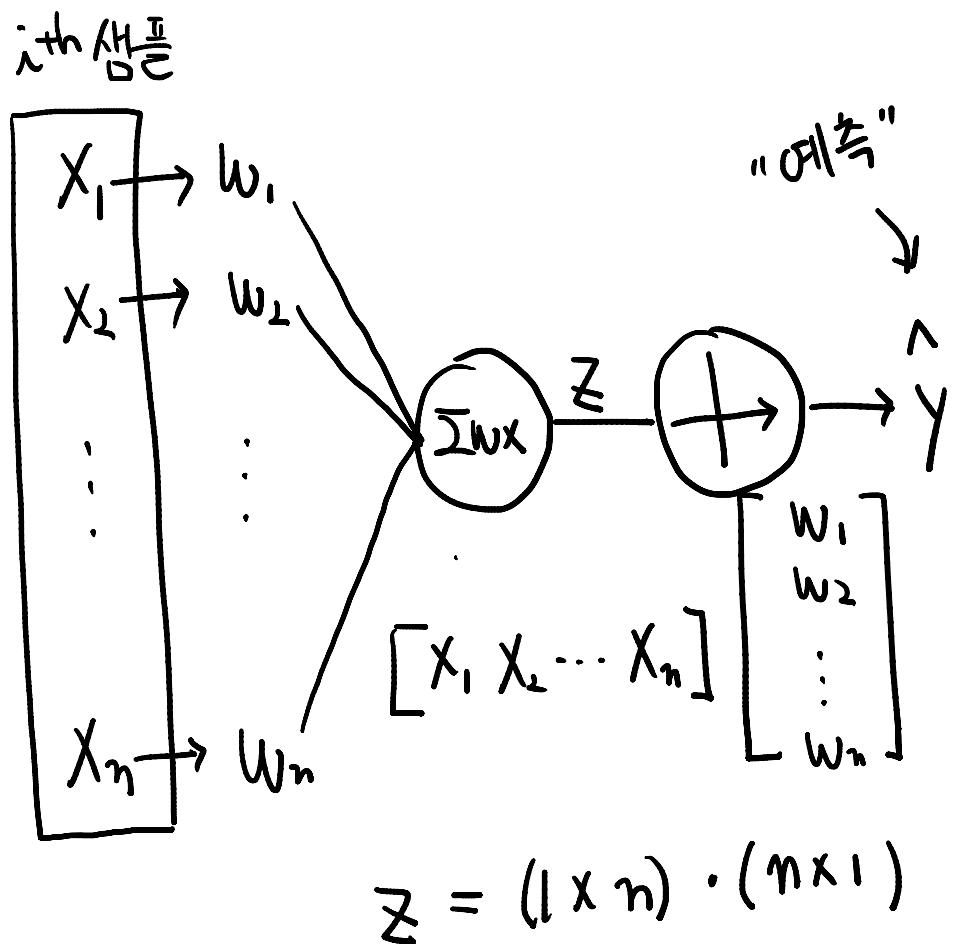
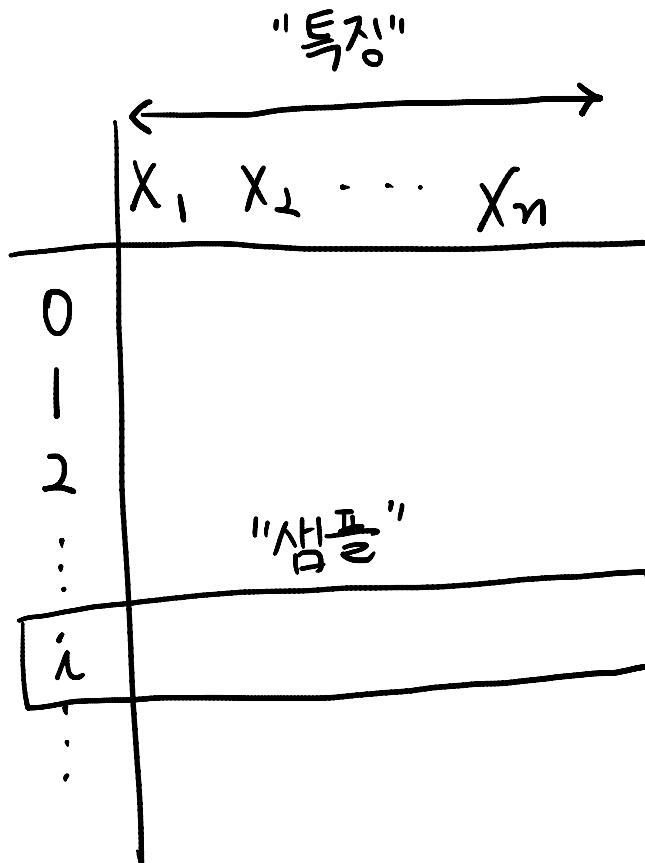
21C

인공신경망

Deep



Data



1943 MCP 뉴런

```
In [106]: x = np.array([0, 1])
```

```
In [107]: w = np.array([0.5, 0.5])  $0 \rightarrow X_1 - w_1$ 
```

```
In [ ]: z = np.dot(x, w)
```

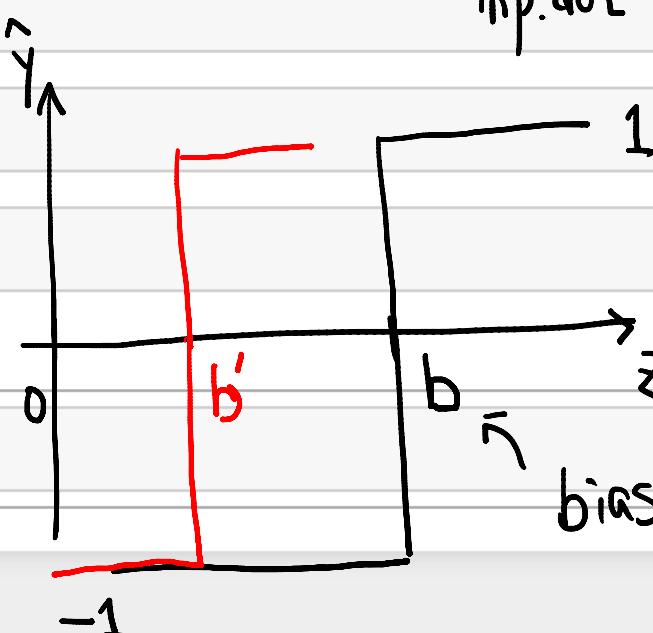
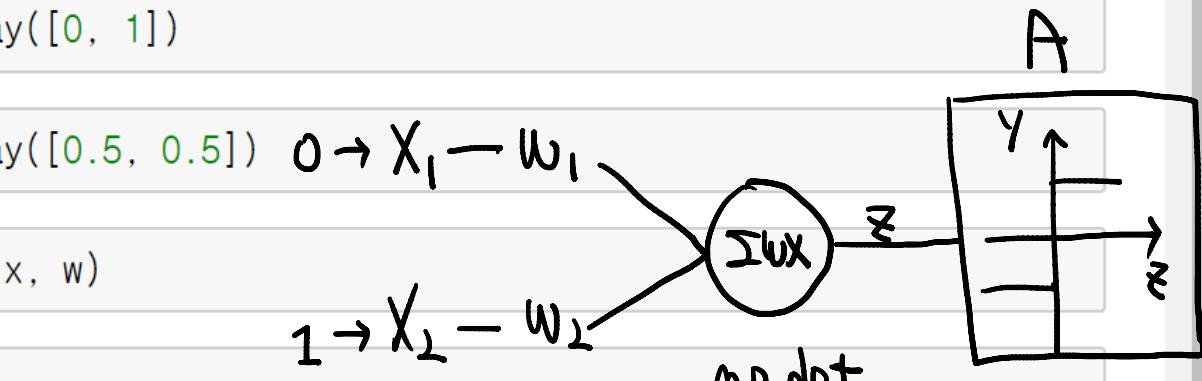
```
In [108]: z
```

```
In [111]: y = 1 if z > 0 else -1
```

```
In [112]: y
```

```
Out[112]: 1
```

```
In [ ]:
```

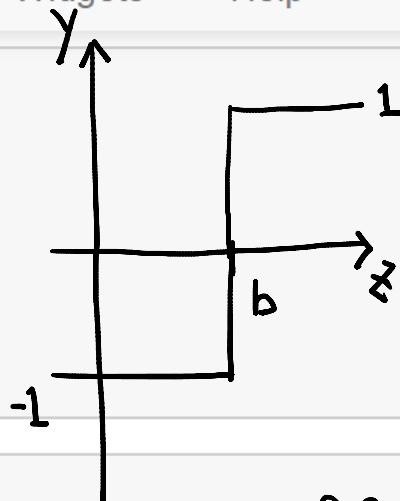


bias "편향"

In [119]:

```
def AND(x1, x2):
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = 0.7
    z = np.dot(x, w)
    y = 1 if z > b else -1
    return 1 if y > 0 else 0
```

MCP



In [120]: AND(0, 0)

Out[120]: 0

In [121]: AND(0, 1)

Out[121]: 0

In [122]: AND(1, 0)

Out[122]: 0

In [123]: AND(1, 1)

Out[123]: 1

e.g.

$$y = \begin{cases} 1 & \rightarrow 1 \text{ 'RED' } 'A' \\ -1 & \rightarrow 0 \text{ 'Blue' } 'B' \end{cases}$$

File Edit View Insert Cell Kernel Widgets Help

Out[116]: -1

```
In [125]: def make_logic_gate(w, b):
    def logic_gate(x1, x2):
        x = np.array([x1, x2])
        z = np.dot(x, w)
        y = 1 if z > b else -1
        return 1 if y > 0 else 0
    return logic_gate
```



```
In [132]: def test_logic(logic_gate):
    for x1, x2 in [(0, 0), (0, 1), (1, 0), (1, 1)]:
        y = logic_gate(x1, x2)
        print(x1, x2, '|', y)
```

```
In [127]: AND = make_logic_gate(w=np.array([0.5, 0.5]), b=0.7)
```

```
In [133]: test_logic(AND)
```

0	0		0
0	1		0
1	0		0
1	1		1

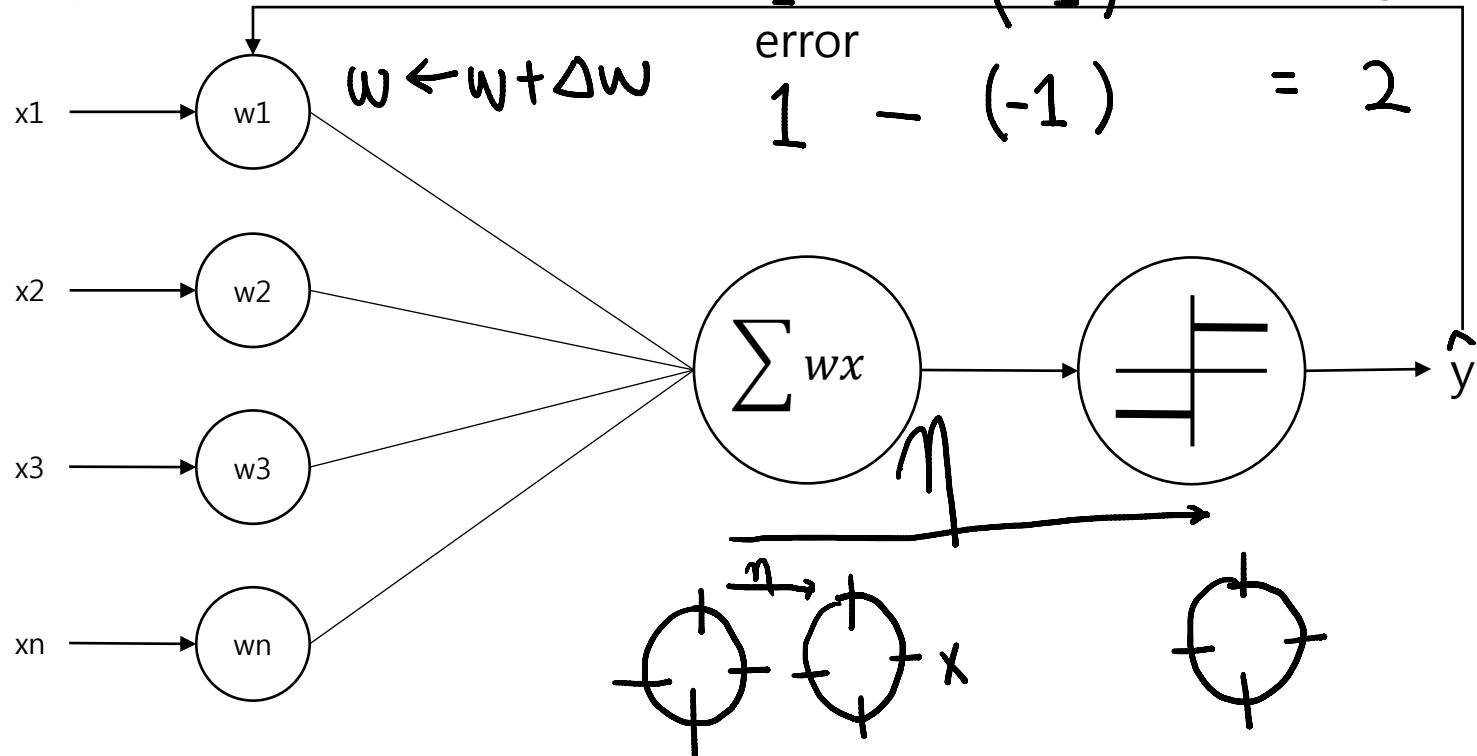
In []:

$$\Delta w_i := \eta \overbrace{ERR}^{\text{하나의}} \times \overbrace{x_i}^{\sim \text{외부환경}}$$

1958 퍼셉트론

"하나"

$$\Delta w \propto ERR$$



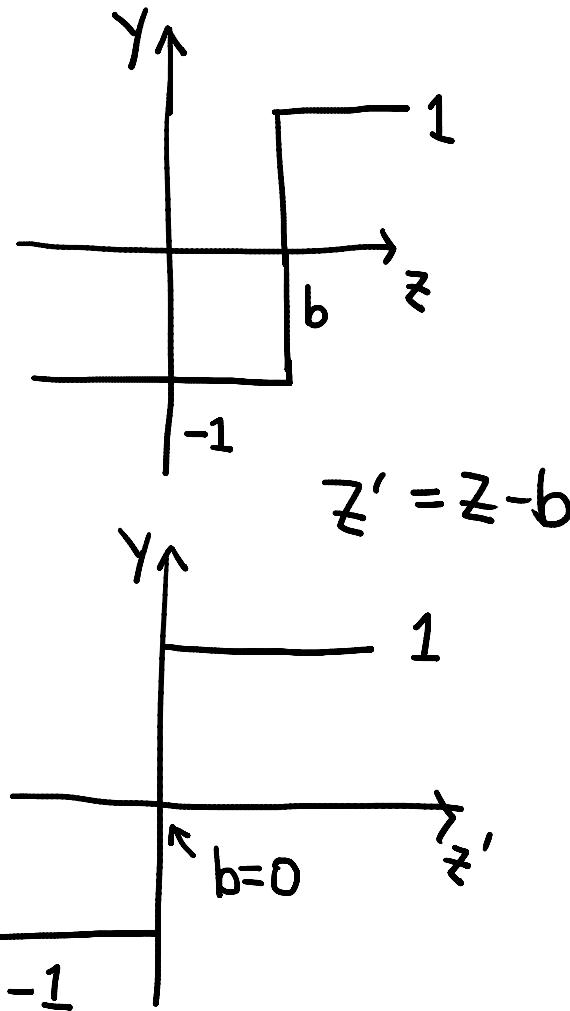
y : 정답 \hat{y} : 예측 ERR

$$\begin{array}{rcl} 1 & - & 1 \\ \hline = & 0 \end{array}$$

$$\begin{array}{rcl} -1 & - & (-1) \\ \hline = & 0 \end{array}$$

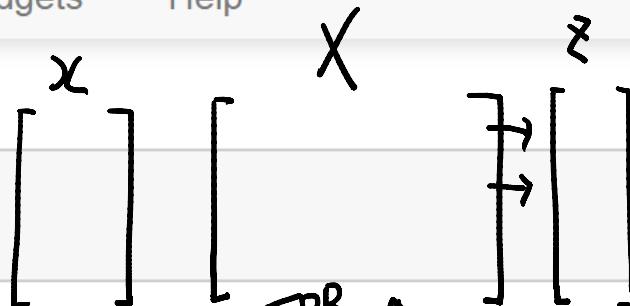
$$\begin{array}{rcl} 1 & - & (-1) \\ \hline = & 2 \end{array}$$

$$\begin{aligned}
 & \text{Diagram:} \\
 & \text{Inputs } x_1, x_2, \dots, x_n \text{ are weighted by } w_1, w_2, \dots, w_n \text{ and summed with a bias } b_0 = w_0 \text{ to produce } z. \\
 & z = \sum w_i x_i + b_0 \\
 & z = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b_0 \\
 & z' = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b \quad \downarrow w_0 x_0 \\
 & z' = \sum_{i=0}^n w_i x_i \quad x_0 = 1
 \end{aligned}$$



1 1 | 0

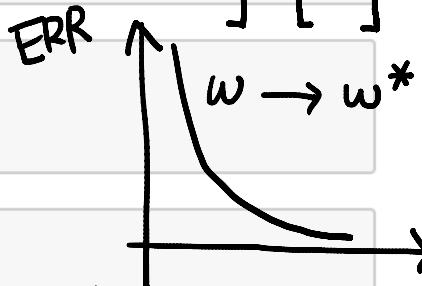
In [138]: `def net_input(X, w, b):
 return np.dot(X, w) + b`



In [139]: `def activation(z):
 return np.where(z > 0, 1, -1)`
1 if $z > b$ else -1

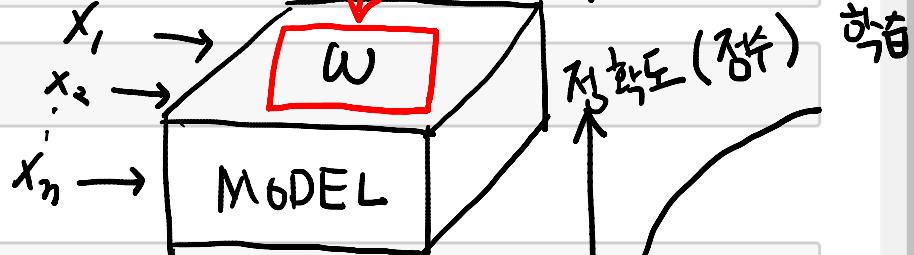
학습 모드

In [141]: `z = net_input(x, w, b)`



In [142]: `z`

Out[142]: 1.1000000000000001



In [143]: `y = activation(z)`

$$1) \text{최적화} \\ w^* = \underset{w}{\operatorname{argmin}} E(w)$$

In [144]: `y`

Out[144]: array(1)

$$E(w^*) \rightarrow 0$$

In []:

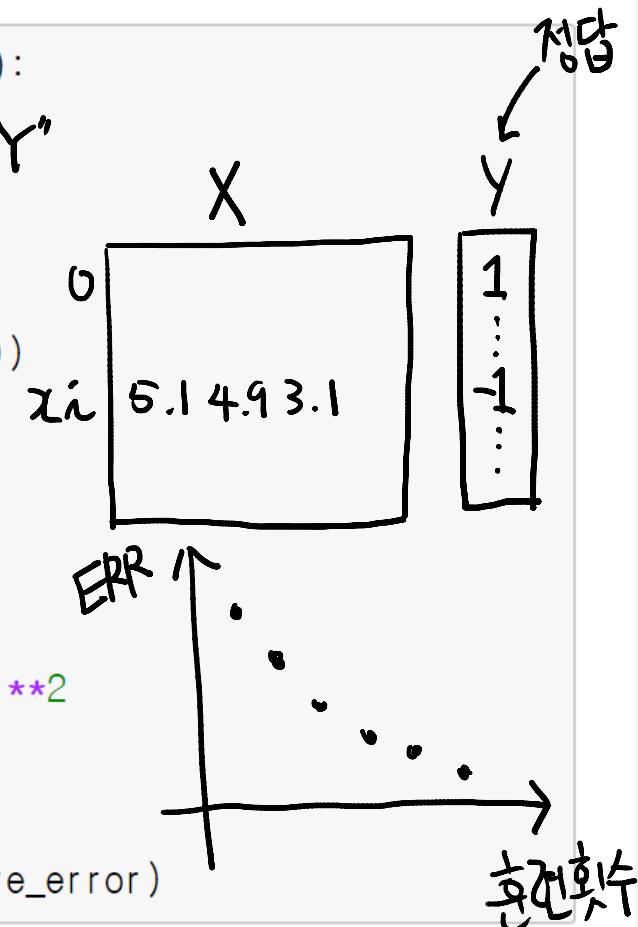
학습 모드

```
In [145]: def fit(X, y, 훈련횟수=10, 학습률=0.1):
    w = np.array([0., 0.])      "BABY"
    b = 0.0

    error_history = []
    for i in range(훈련횟수):
        print('훈련횟수: {}'.format(i))
        sum_square_error = 0
        for xi, yi in zip(X, y):
            zi = net_input(xi, w, b)
            yi_pred = activation(zi)
            # 가중치 갱신
            error = yi - yi_pred
            sum_square_error += error **2
            update = 학습률 * error
            w += update * xi
            b += update
        error_history.append(sum_square_error)
```

zip(x,y) "각 샘플에 대해" MCP "Perceptron"

$\begin{matrix} \text{zip}(x,y) \\ (x_1, y_1) \\ (x_2, y_2) \\ \vdots \end{matrix}$



In []:

```
return w, b, error_history
```



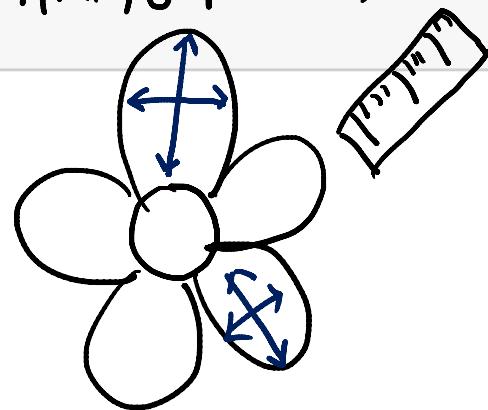
In [147]: `iris = pd.read_csv('data/iris.data', header=None)`

In [148]: `iris[:5]`

Out [148]:

	0	1	2	3	X	y
0	5.1	3.5	1.4	0.2	Iris-setosa	
1	4.9	3.0	1.4	0.2	Iris-setosa	
2	4.7	3.2	1.3	0.2	Iris-setosa	
3	4.6	3.1	1.5	0.2	Iris-setosa	
4	5.0	3.6	1.4	0.2	Iris-setosa	

기계학습 Hello, World!



In []:

File

Edit

View

Insert

Cell

Kernel

Widgets

Help

1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

$$\hat{y} = \begin{cases} 1 \\ -1 \end{cases}$$

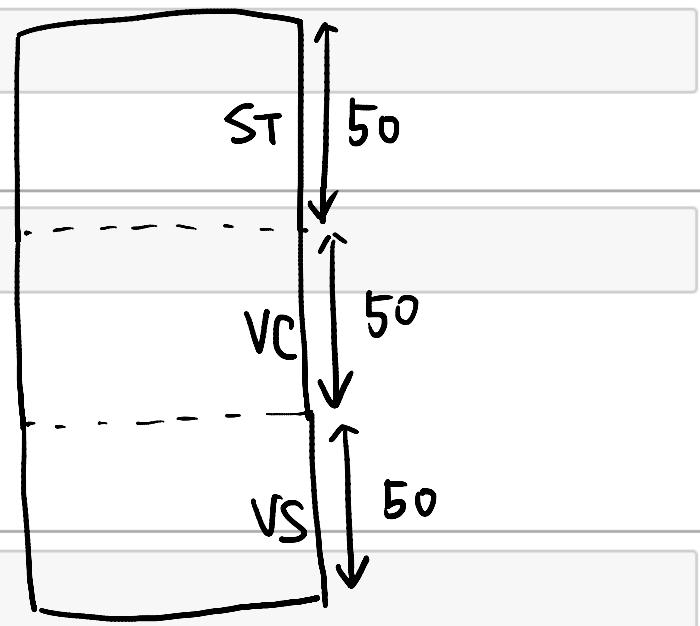
```
In [150]: len(iris)
```

```
Out[150]: 150
```

```
In [152]: iris[4].value_counts()
```

```
Out[152]: Iris-virginica    50  
Iris-versicolor      50  
Iris-setosa          50  
Name: 4, dtype: int64
```

```
In [ ]:
```



In [180]: `w, b, error_history = fit(X, y)`

훈련횟수: 1

w: [-0.38 0.06 -0.66 -0.24] b: 0.0 Error: 8

훈련횟수: 2

w: [-0.76 0.12 -1.32 -0.48] b: 0.0 Error: 8

훈련횟수: 3

w: [0.26 0.82 -1.04 -0.44] b: 0.2 Error: 4

훈련횟수: 4

w: [0.26 0.82 -1.04 -0.44] b: 0.2 Error: 0

훈련횟수: 5

w: [0.26 0.82 -1.04 -0.44] b: 0.2 Error: 0

훈련횟수: 6

w: [0.26 0.82 -1.04 -0.44] b: 0.2 Error: 0

훈련횟수: 7

w: [0.26 0.82 -1.04 -0.44] b: 0.2 Error: 0

훈련횟수: 8

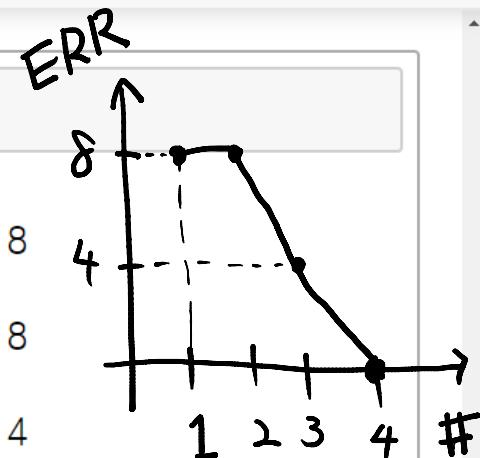
w: [0.26 0.82 -1.04 -0.44] b: 0.2 Error: 0

훈련횟수: 9

w: [0.26 0.82 -1.04 -0.44] b: 0.2 Error: 0

훈련횟수: 10

w: [0.26 0.82 -1.04 -0.44] b: 0.2 Error: 0



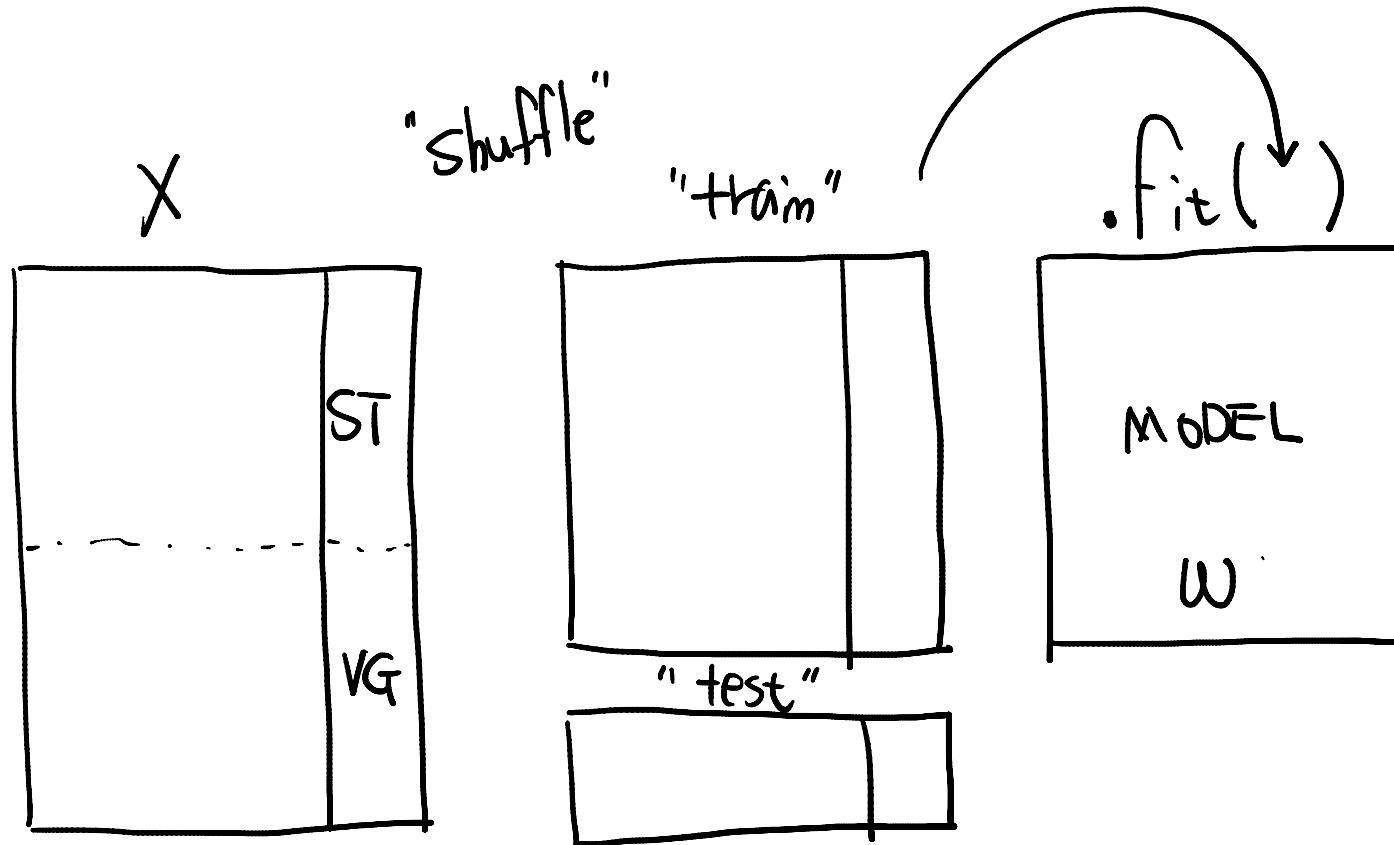
① w^*

② 일반화

$ERR(w_{train}^*) \rightarrow 0$

$ERR(\downarrow) \rightarrow 0$
test

In []:



File Edit View Insert Cell Kernel Widgets Help

In [181]: `shuffled_index = np.random.permutation(range(len(data)))`

In [186]: `shuffled_index[:10]`

Out[186]: `array([71, 6, 80, 98, 28, 73, 96, 15, 85, 17])`

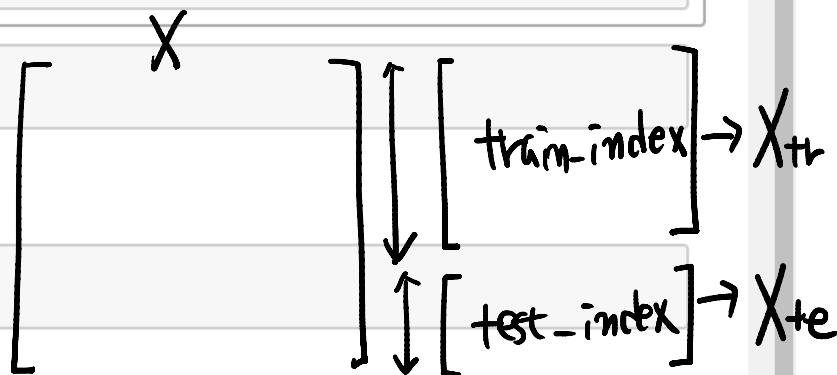
In [183]: `train_index, test_index = shuffled_index[:70], shuffled_index[70:]`

In [184]: `len(train_index)`

Out[184]: 70

In [185]: `len(test_index)`

Out[185]: 30



In [162]: `X = data.loc[:, 0:3].values`

In [166]: `y = data[4]`

In [160]: `y.value_counts()`

...

File Edit View Insert Cell Kernel Widgets Help

```
y = activation(z)  
return y
```

"정우"



In [191]: y_pred = predict(X[test_index], w, b)

In [192]: y_pred

...

In [193]: y[test_index]

...

In [197]: score = (y_pred == y[test_index]).sum() / len(test_index)

In [198]: score

Out[198]: 1.0

In []:

plot_util.py

Trusted

Python 3

File

Edit

View

Insert

Cell

Kernel

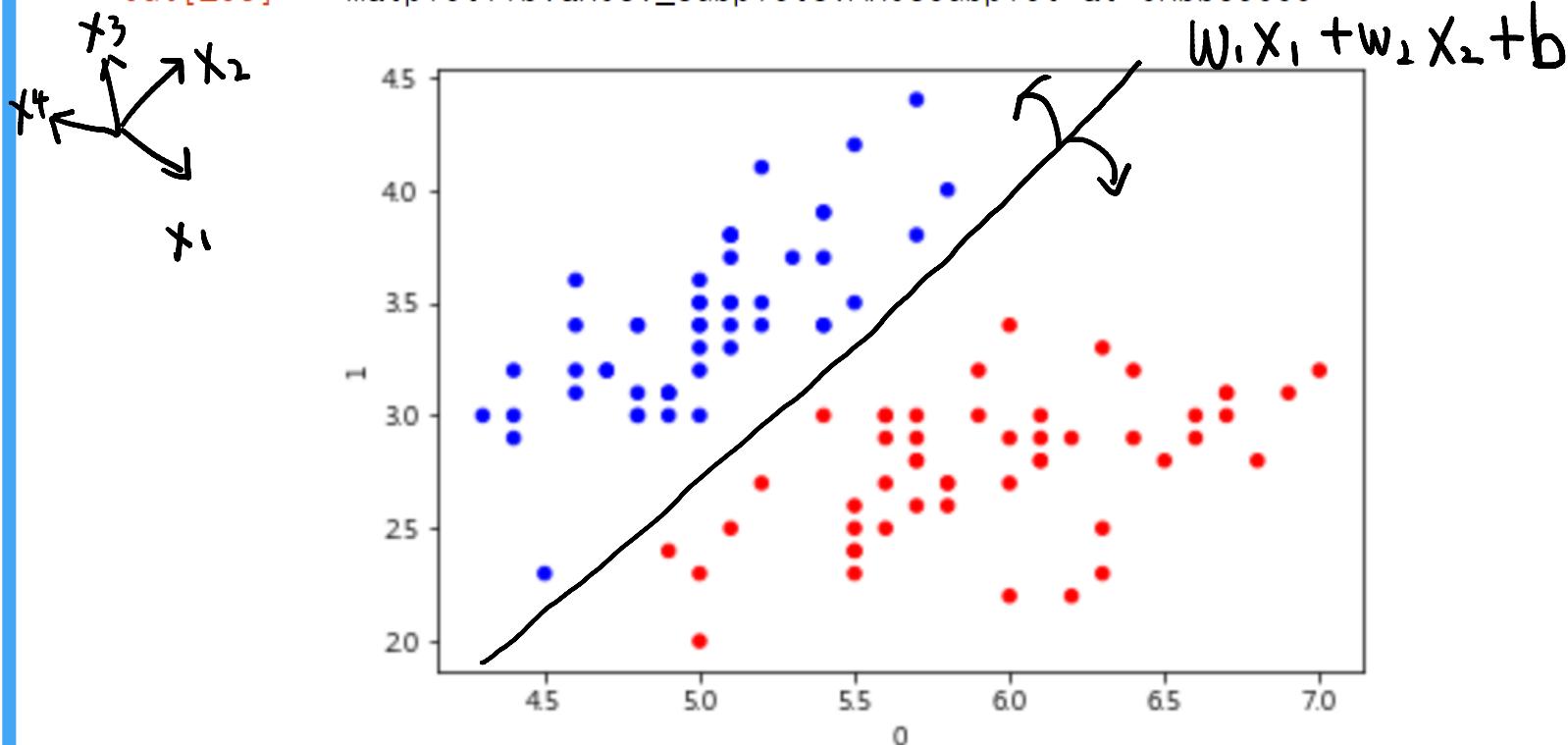
Widgets

Help

```
In [201]: from plot_util import get_colormap
```

```
In [203]: c = get_colormap(y, colors='rb')
data.plot(kind='scatter', x=0, y=1, c=c)
```

Out[203]: <matplotlib.axes._subplots.AxesSubplot at 0xbb69cc0>



```
In [ ]:
```