



2024 D&A

시각화

Basic Session 6차시



EDA Competition 일정

EDA Competition 일정

EDA Competition 시작 : 2024.04.09 (화)

EDA Competition 결과물 제출 : 2024.05.19 (일) 23:59

결과물 제출은 ppt 보고서와 소스 코드가 담긴 ipynb 파일을 '.zip' 파일로 압축하여 tjfud0216@kookmin.ac.kr로 제출

ppt 보고서

1. 배경
2. 사용한 데이터
3. 전처리 과정
4. 분석 & 시각화 과정
5. 결론 및 활용 방안

소스 코드

1. 전처리
2. 분석
3. 시각화 과정

EDA Competition 결과물 발표 : 2024.05.21 (화)

한 팀당 발표 시간 7 분 부여 (시간 엄수)

발표 후 쉬는 시간 동안 순위 결정

바로 당일 시상식 진행

CONTENTS.

01. 시각화

- 시각화

02. pandas

- pandas
- 유형

03. matplotlib

- matplotlib
- 유형
- 그래프 꾸미기
- 여러 그래프 그리기

04. seaborn

- seaborn
- 유형
- 그래프 꾸미기

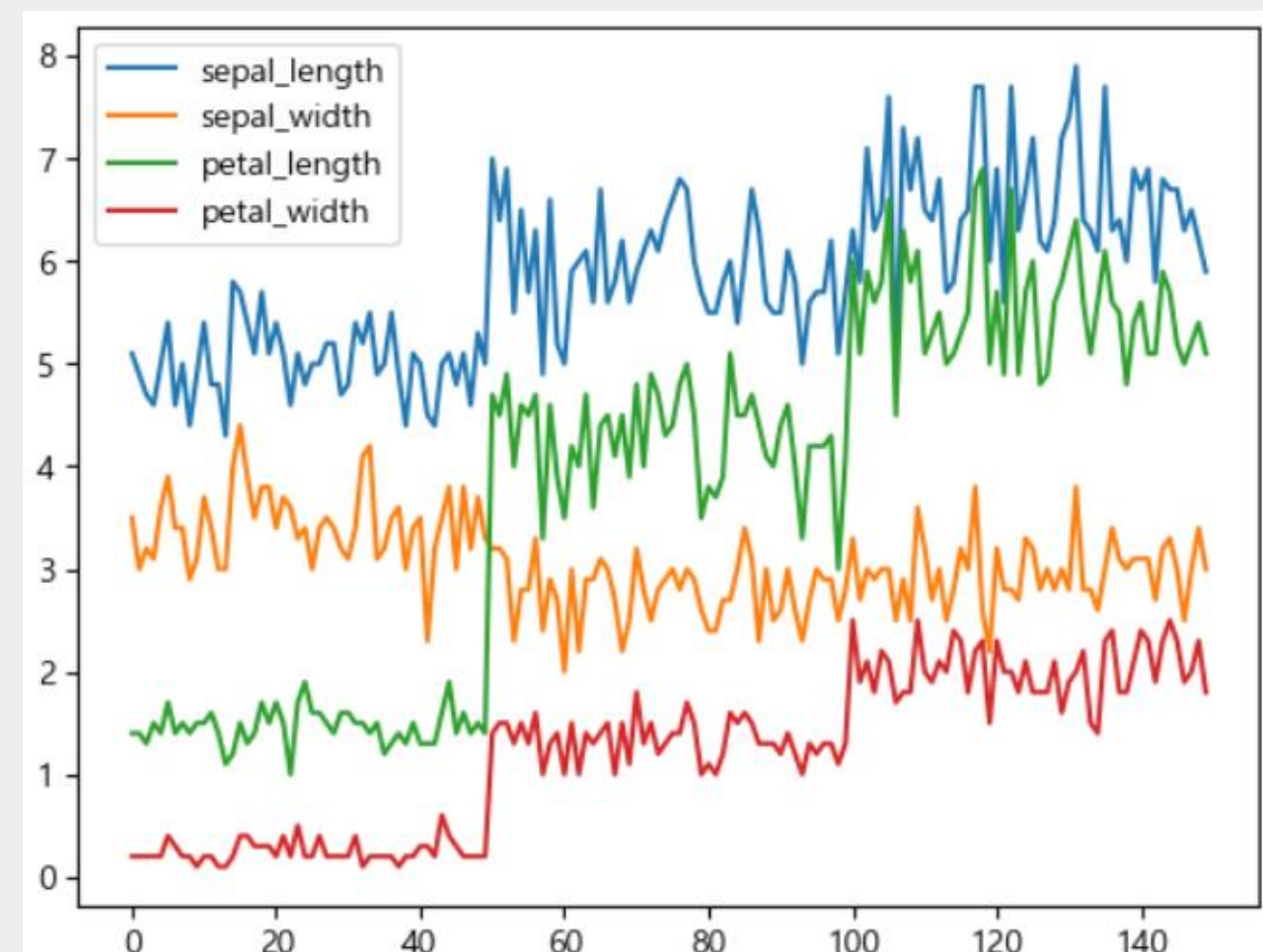
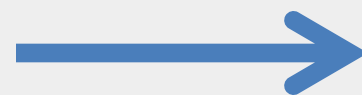
시각화

시각화

시각화

데이터를 시각적으로 표시하는 것
데이터 자체나 수치로만 볼 때는 알 수 없었던 데이터의 패턴, 다른 요소들 간의 연관성 등의 인사이트를 발견하여
더 나은 의사결정 도출 가능

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica



pandas

pandas

.plot()

pandas의 DataFrame, Series의 시각화를 위한 메서드
해당 메서드에서 **내부적으로** matplotlib을 불러와서 사용

*matplotlib을 import하지 않고 matplotlib 기능 활용

사용 방법

```
# plot.종류()  
DF.plot.line()
```

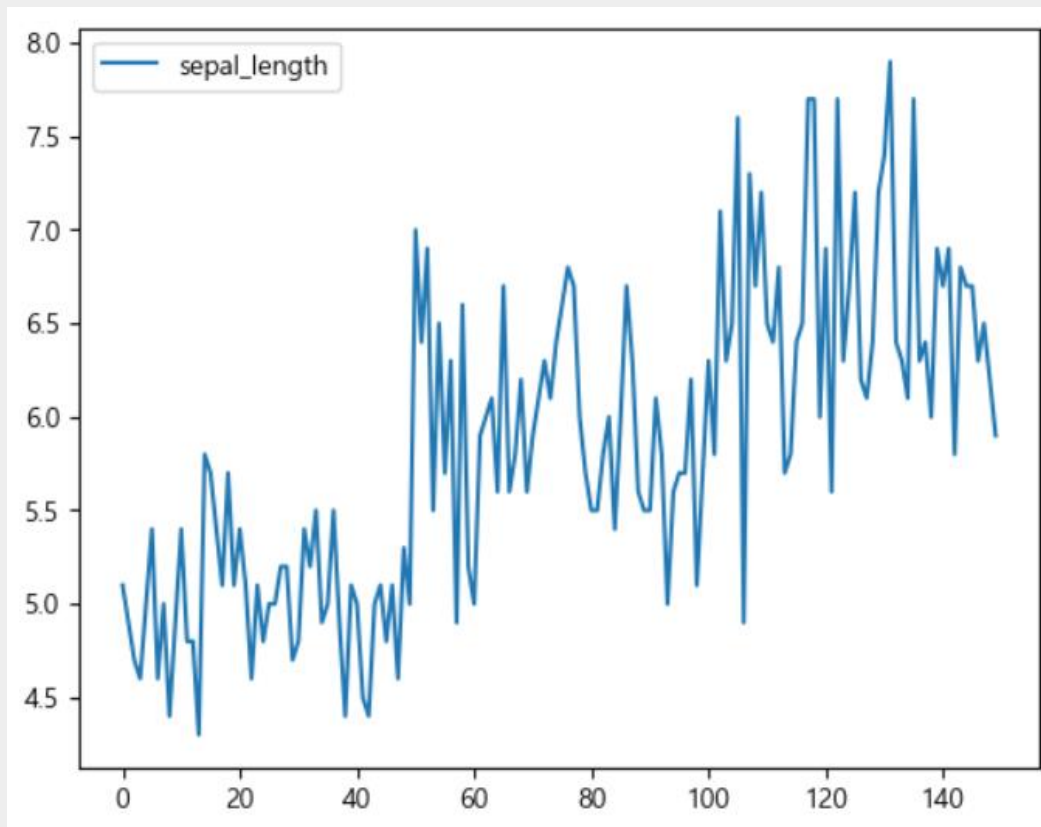
```
# plot(kind = '종류')  
DF.plot(kind = 'line')
```

pandas

유형

line

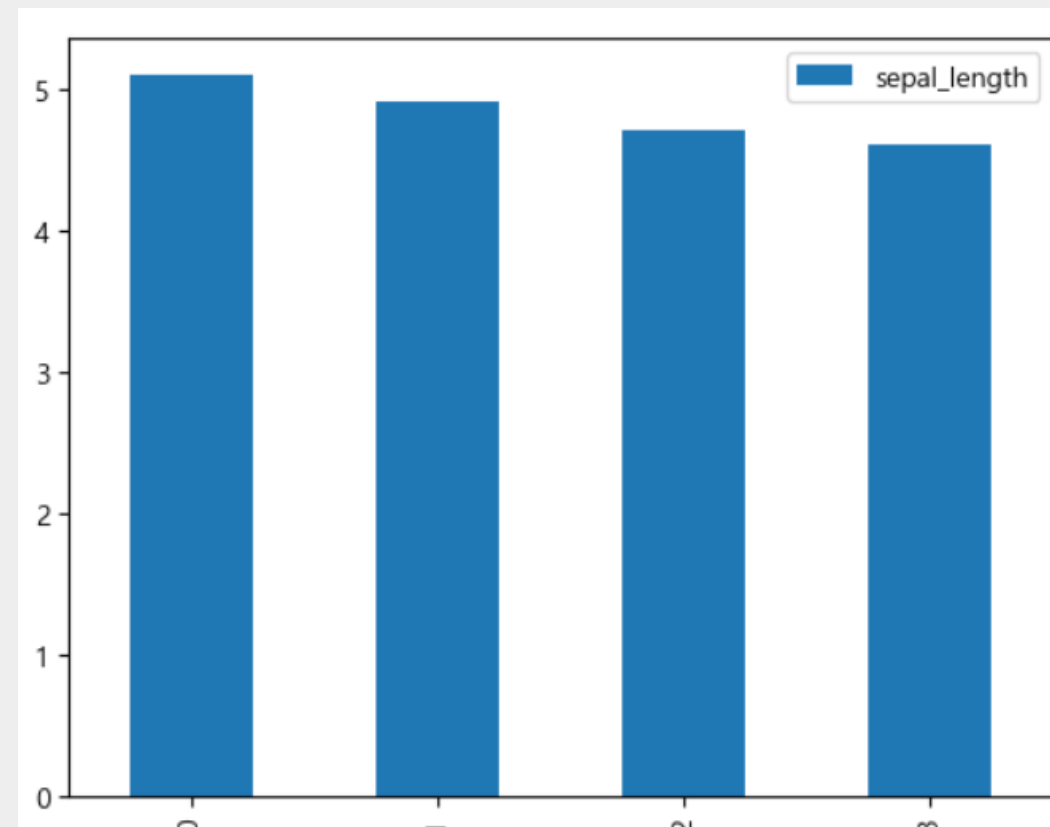
수치형 데이터의 변화(순서, 추세)



```
.plot()  
.plot.line()  
.plot(kind = 'line')
```

bar

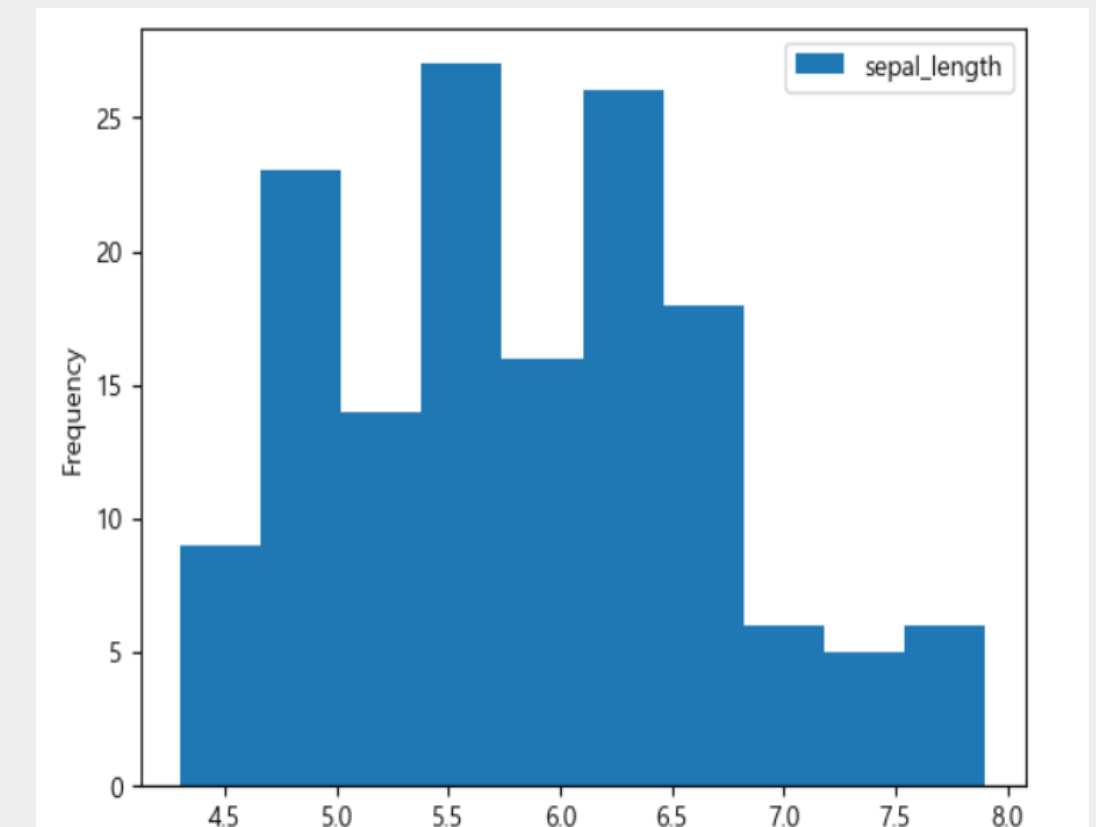
수치형 데이터의 값 비교



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

hist

데이터의 분포

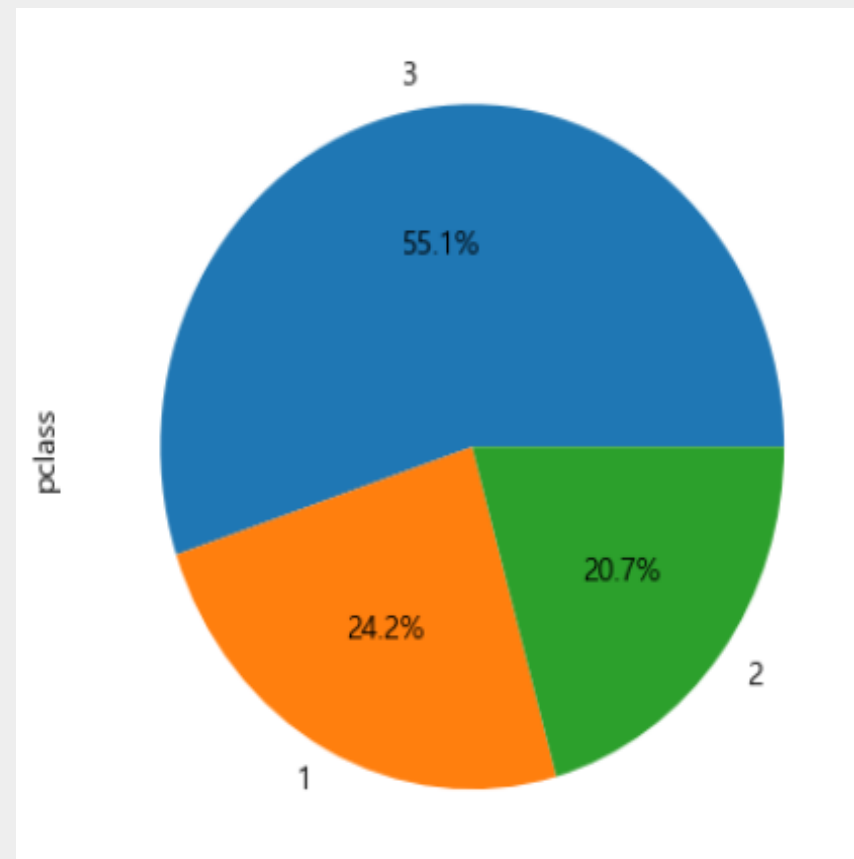


```
.plot.hist()  
.plot(kind = 'hist')
```

pandas 유형

pie

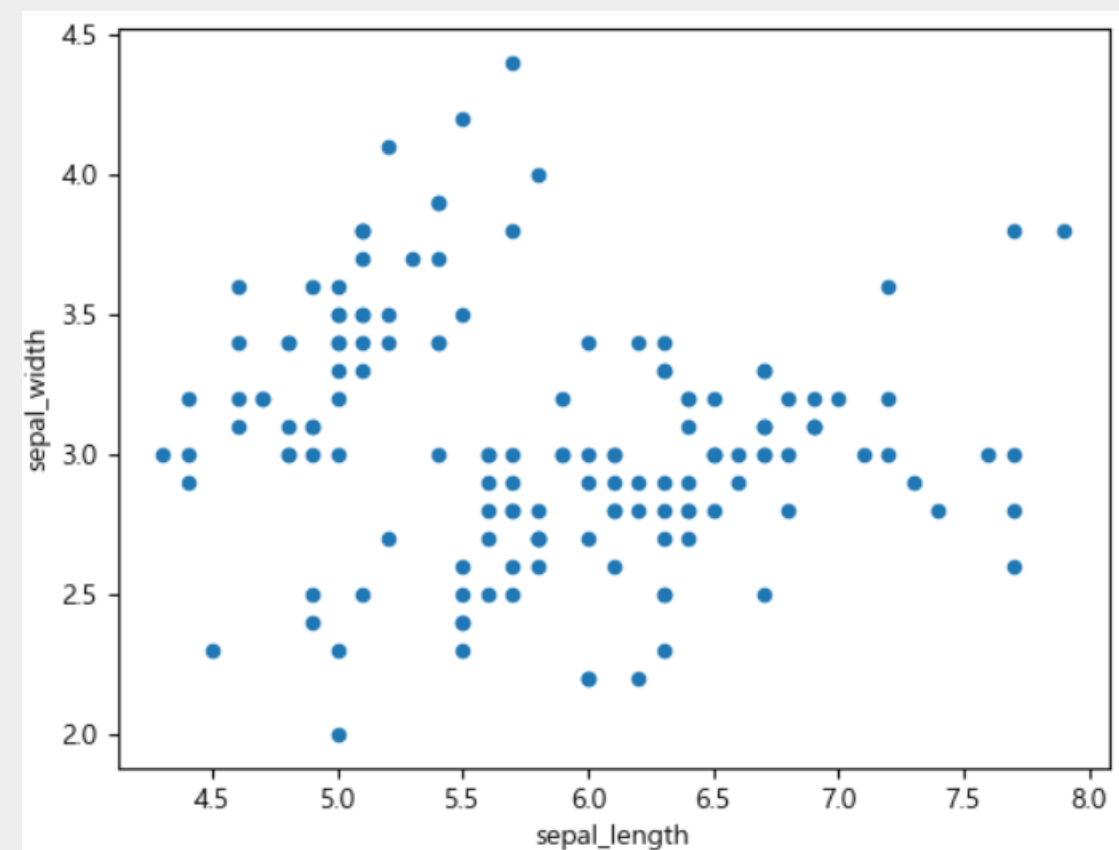
범주형 데이터의 범주 비교



```
.plot.pie()  
.plot(kind = 'pie')
```

scatter

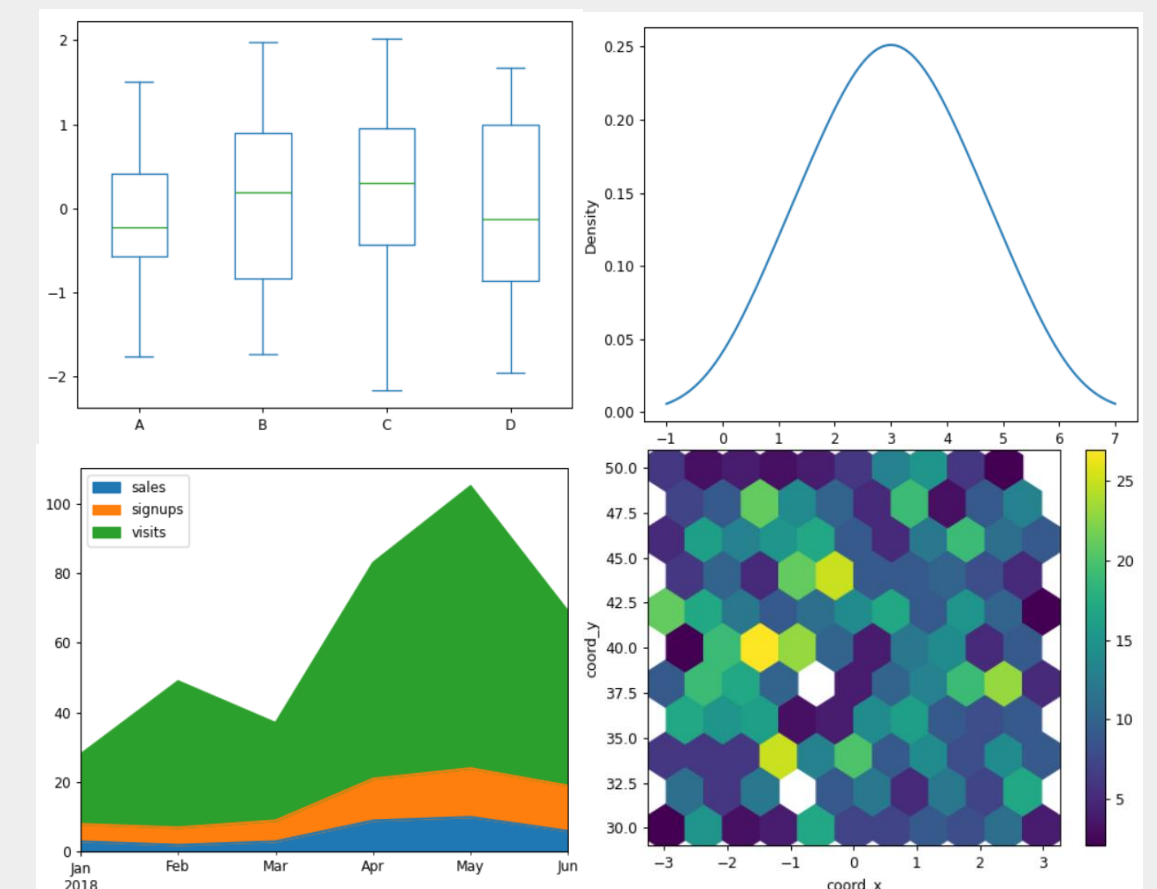
두 수치형 데이터 간의 상관관계



```
.plot.scatter(x, y)  
.plot(x, y, kind = 'scatter')
```

기타

barh, box, kde, density,
area, hexbin(DataFrame only)



<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html#pandas.DataFrame.plot>

matplotlib

matplotlib

matplotlib

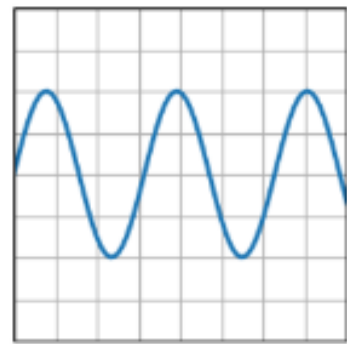
가장 대표적인 python의 시각화 라이브러리
matplotlib.pyplot의 함수를 사용하여 시각화



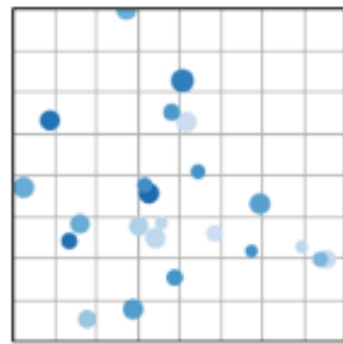
사용 방법

```
import matplotlib.pyplot as plt  
plt.plot()
```

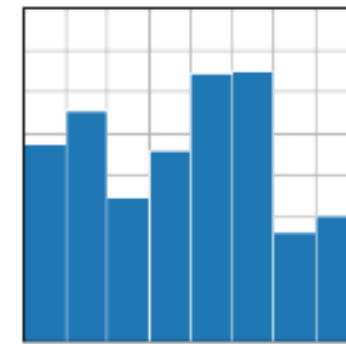

Basic plots



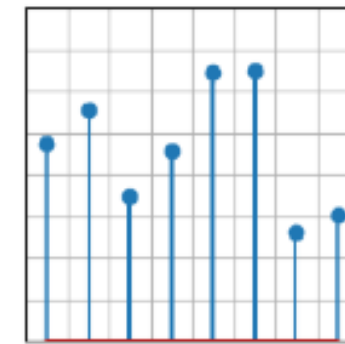
`plot(x, y)`



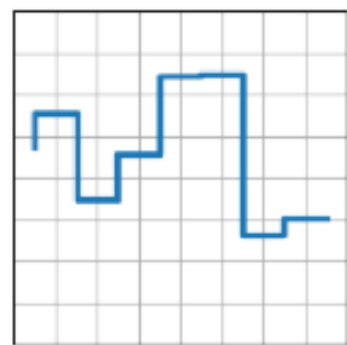
`scatter(x, y)`



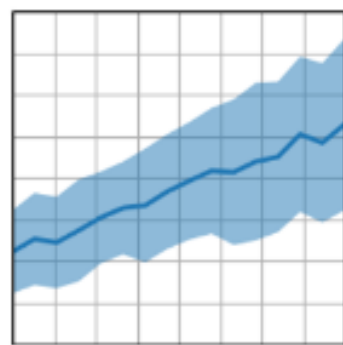
`bar(x, height) / barh(y, width)`



`stem(x, y)`

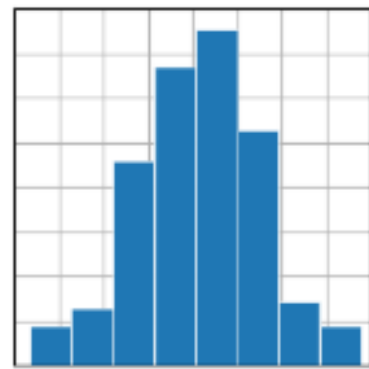


`step(x, y)`

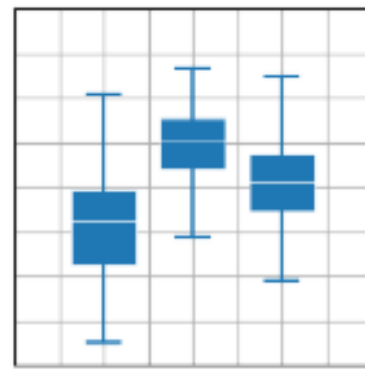


`fill_between(x, y1, y2)`

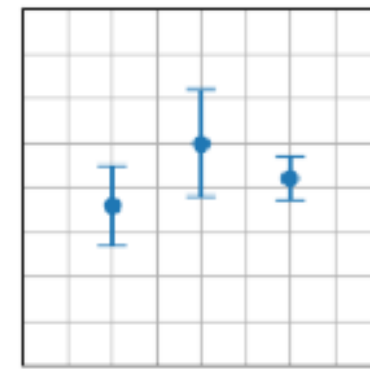
Statistics plots



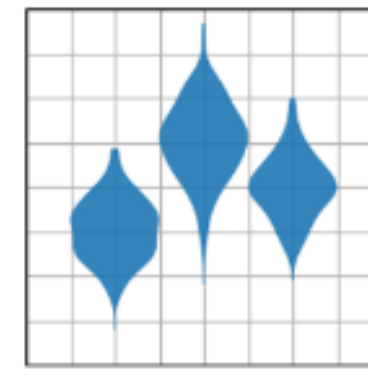
hist(x)



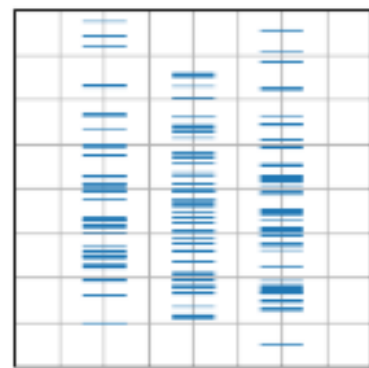
boxplot(X)



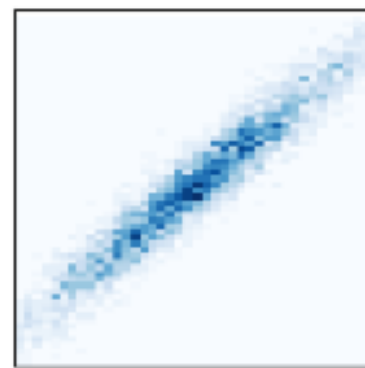
errorbar(x, y, yerr, xerr)



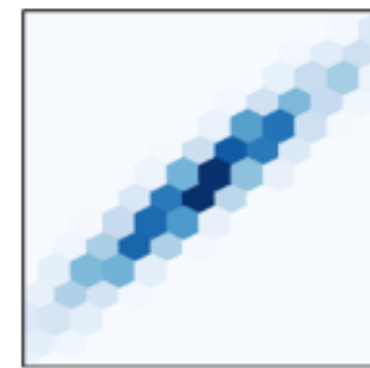
violinplot(D)



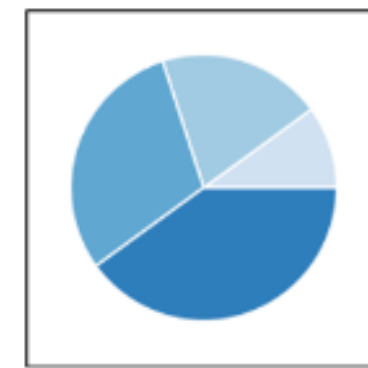
eventplot(D)



hist2d(x, y)



hexbin(x, y, C)



pie(x)

그래프 꾸미기

그래프 요소

크기 (Figure Size)

```
plt.figure(figsize = (width, height))
```

그래프 제목

```
plt.title('그래프 제목')
```

축 Label

```
plt.xlabel('X축 Label')
```

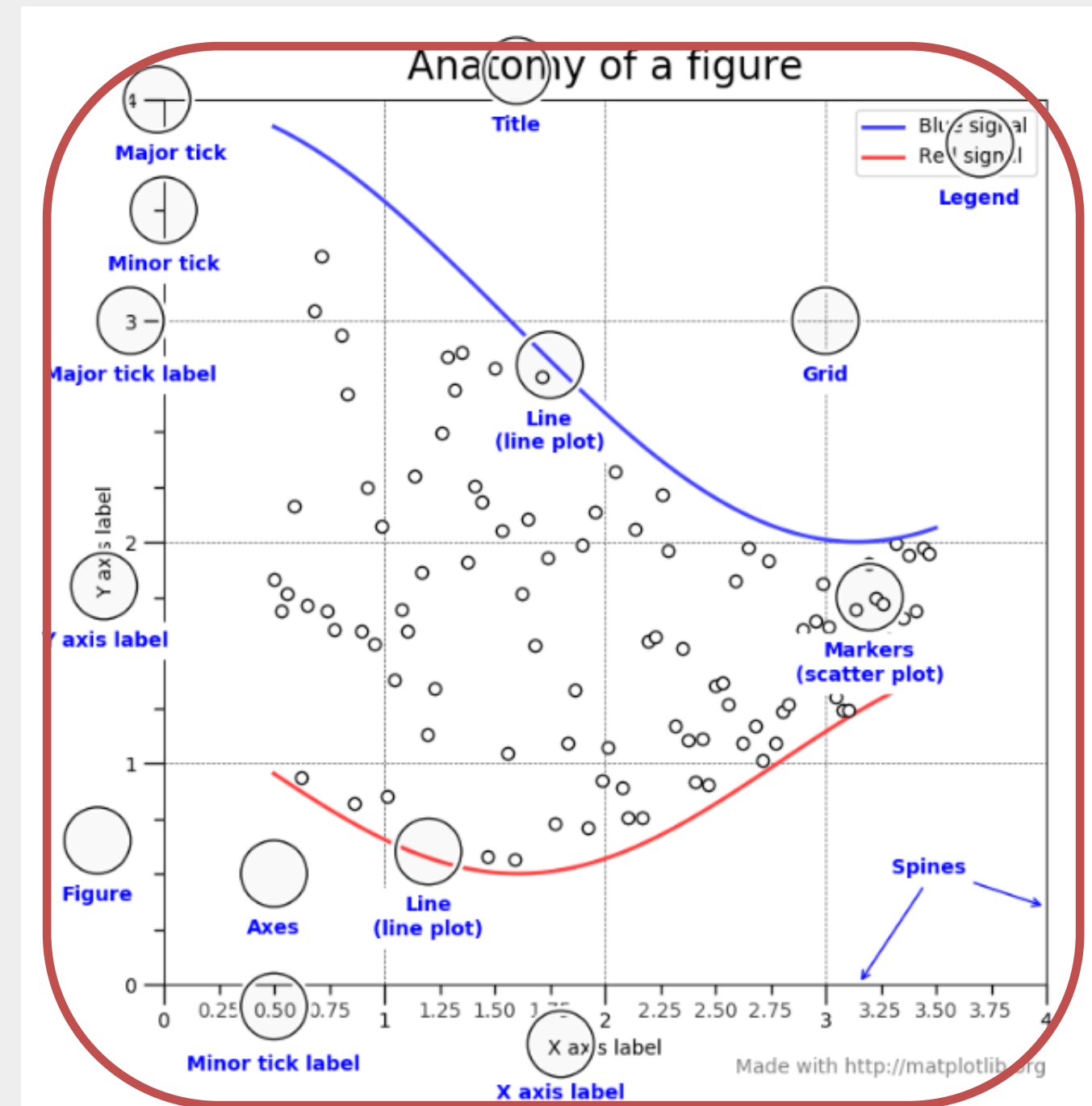
```
plt.ylabel('Y축 Label')
```

축 범위 지정

```
plt.xlim([xmin, xmax])
```

```
plt.ylim([ymin, ymax])
```

```
# plt.axis([xmin, xmax, ymin, ymax])
```



그래프 꾸미기

그래프 요소

크기 (Figure Size)

```
plt.figure(figsize = (width, height))
```

그래프 제목

```
plt.title('그래프 제목')
```

축 Label

```
plt.xlabel('X축 Label')
```

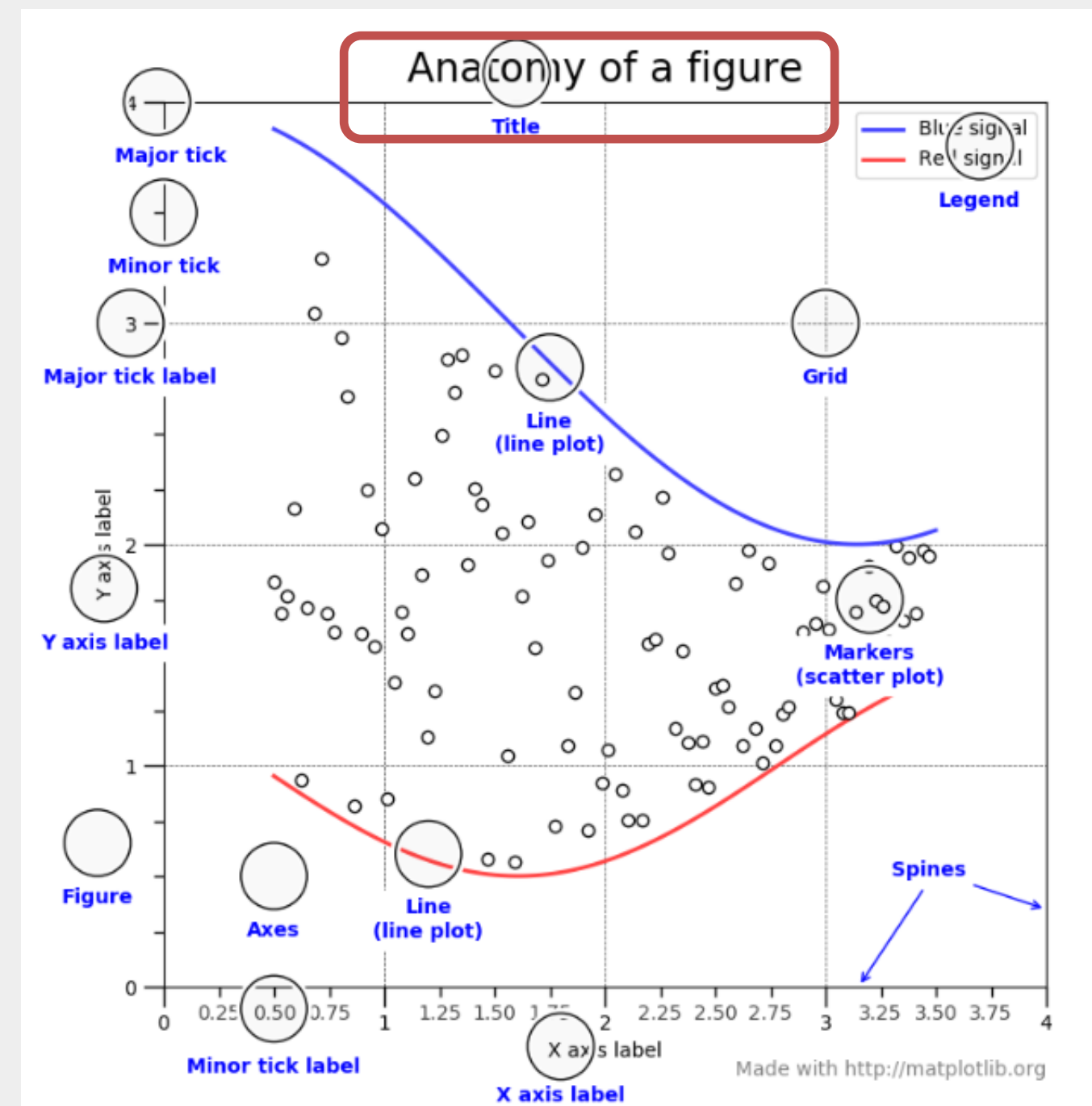
```
plt.ylabel('Y축 Label')
```

축 범위 지정

```
plt.xlim([xmin, xmax])
```

```
plt.ylim([ymin, ymax])
```

```
# plt.axis([xmin, xmax, ymin, ymax])
```



그래프 꾸미기

그래프 요소

크기 (Figure Size)

```
plt.figure(figsize = (width, height))
```

그래프 제목

```
plt.title('그래프 제목')
```

축 Label

```
plt.xlabel('X축 Label')
```

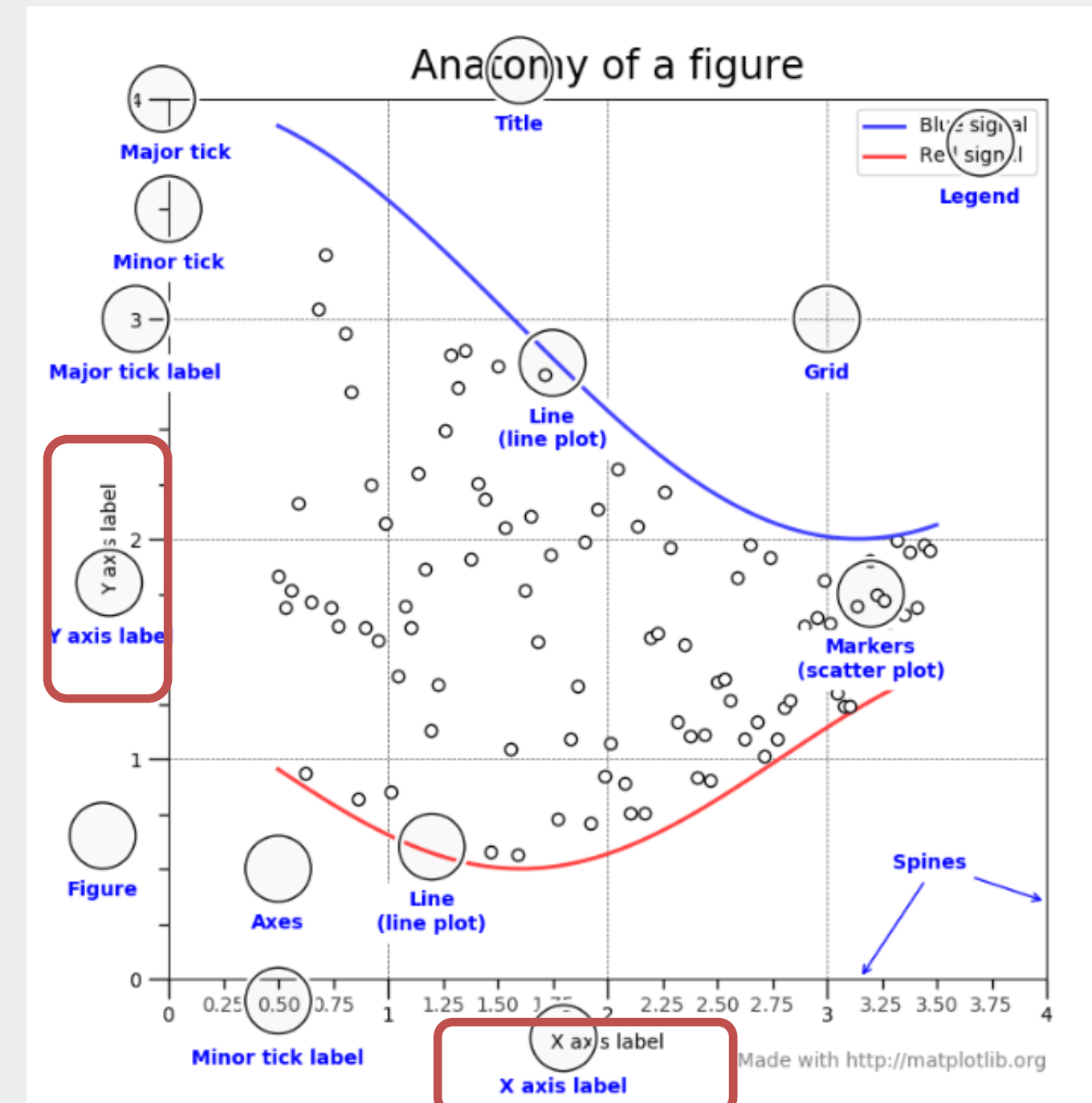
```
plt.ylabel('Y축 Label')
```

축 범위 지정

```
plt.xlim([xmin, xmax])
```

```
plt.ylim([ymin, ymax])
```

```
# plt.axis([xmin, xmax, ymin, ymax])
```



그래프 꾸미기

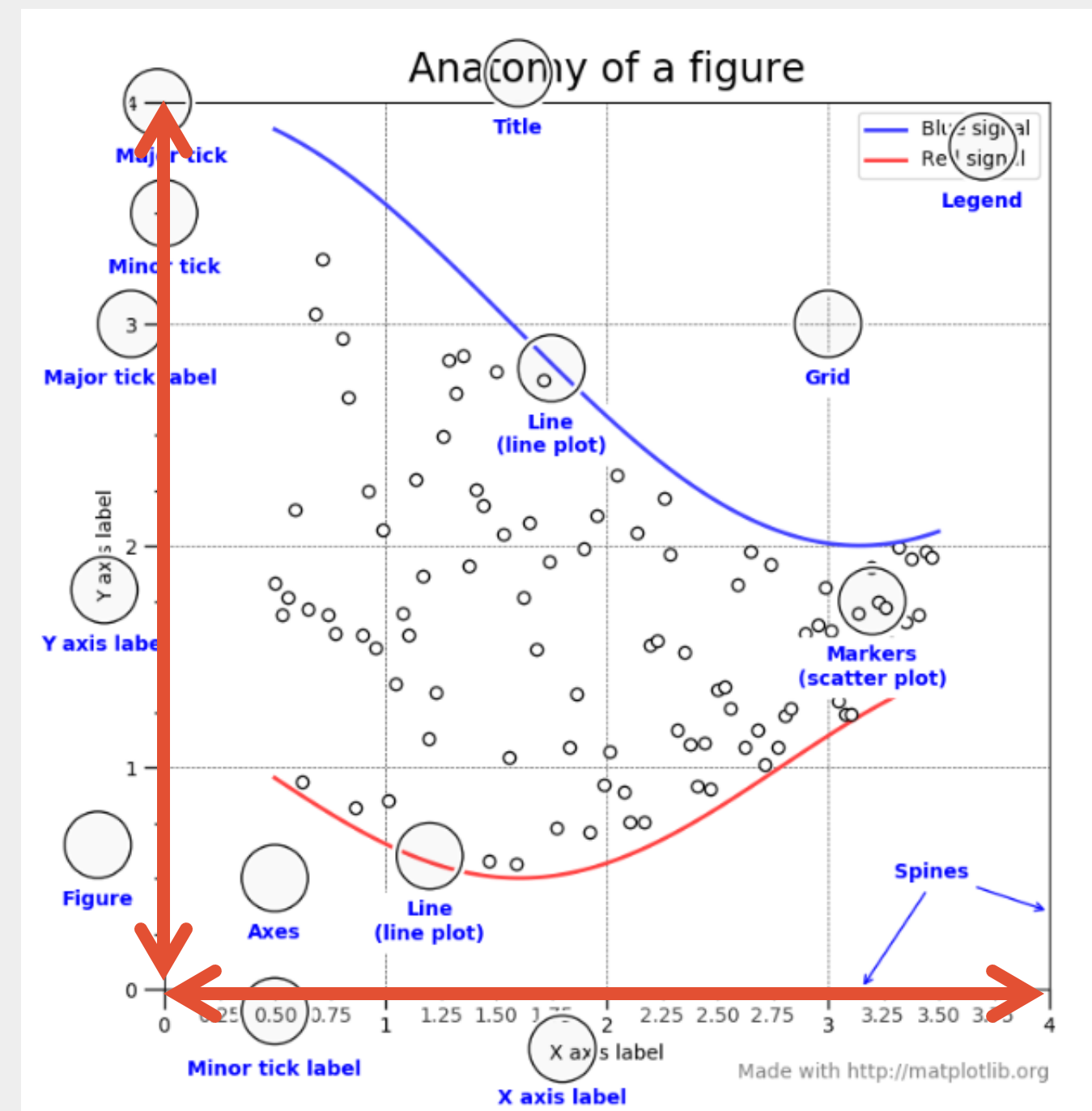
그래프 요소

```
# 크기 (Figure Size)
plt.figure(figsize = (width, height))
```

```
# 그래프 제목
plt.title('그래프 제목')
```

```
# 축 Label
plt.xlabel('X축 Label')
plt.ylabel('Y축 Label')
```

```
# 축 범위 지정
plt.xlim([xmin, xmax])
plt.ylim([ymin, ymax])
# plt.axis([xmin, xmax, ymin, ymax])
```



그래프 꾸미기

그래프 요소

축 눈금

```
plt.xticks(np.arange(xmin, xmax+1))
```

```
plt.yticks(np.arange(ymin, ymax+1))
```

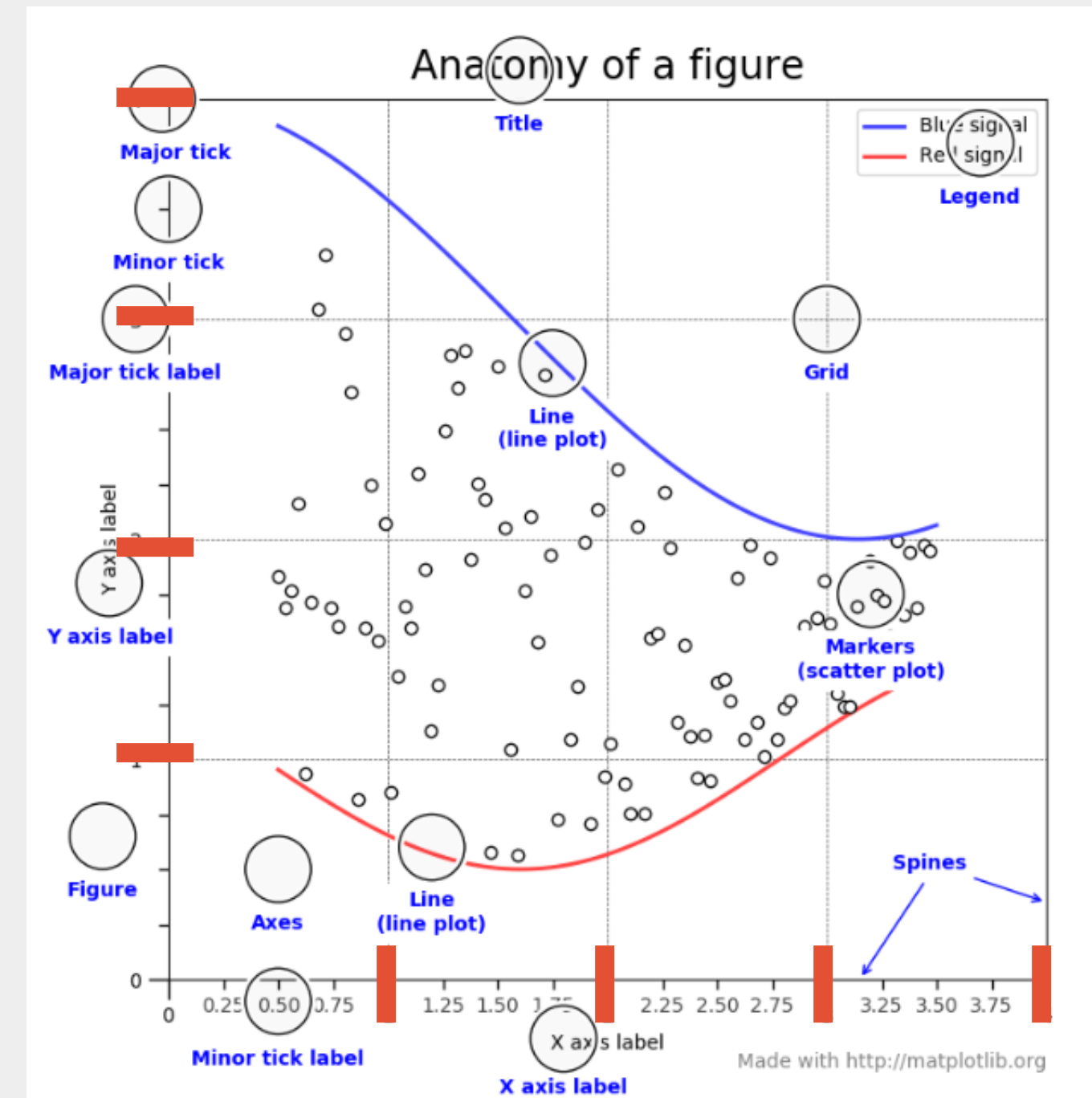
```
# plt.yticks([0, 2, 4])
```

범례

```
plt.legend()
```

그리드

```
plt.grid(True)
```



그래프 꾸미기

그래프 요소

축 눈금

```
plt.xticks(np.arange(xmin, xmax+1))
```

```
plt.yticks(np.arange(ymin, ymax+1))
```

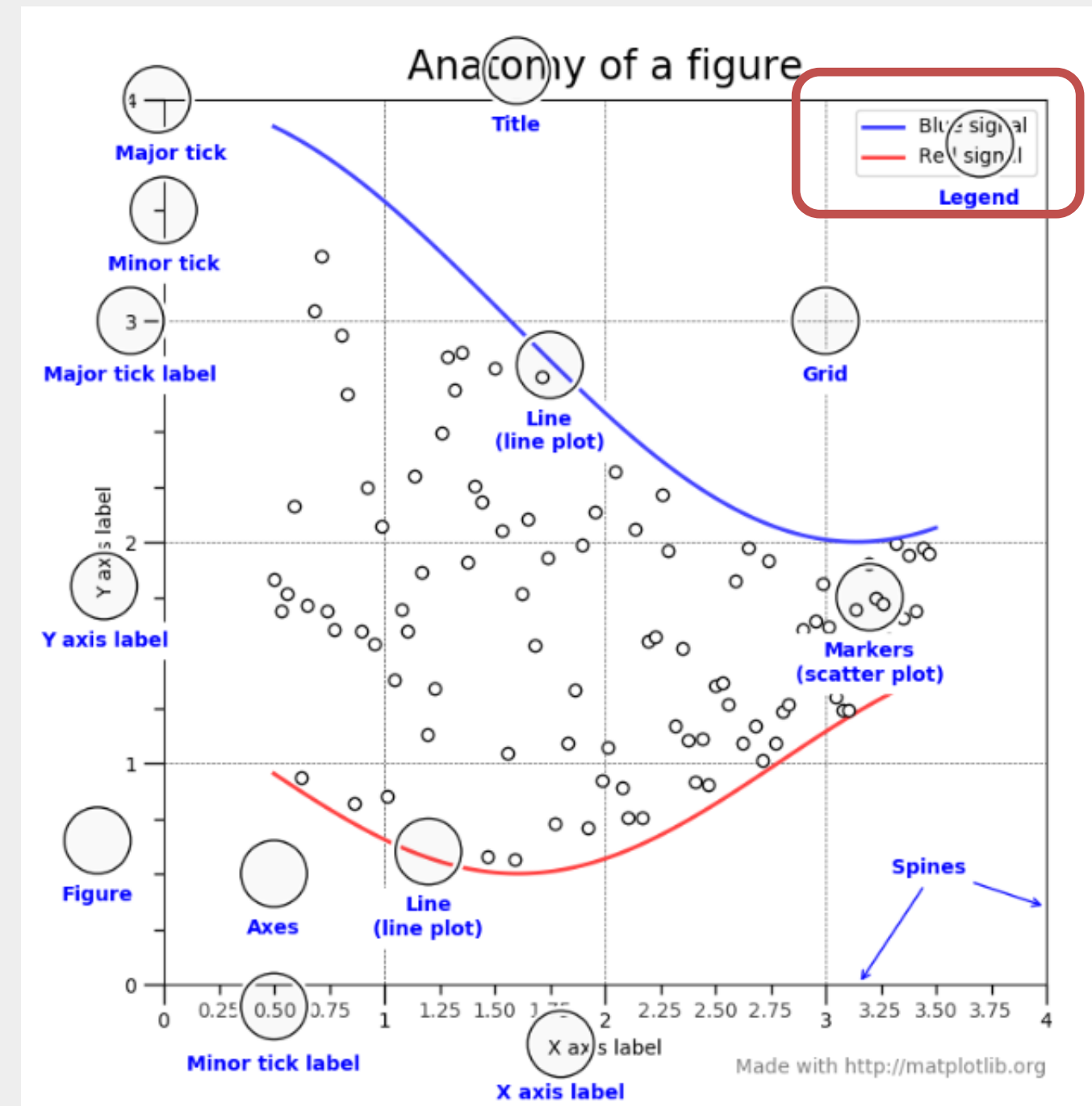
```
# plt.yticks([0, 2, 4])
```

범례

```
plt.legend()
```

그리드

```
plt.grid(True)
```



그래프 꾸미기

그래프 요소

축 눈금

```
plt.xticks(np.arange(xmin, xmax+1))
```

```
plt.yticks(np.arange(ymin, ymax+1))
```

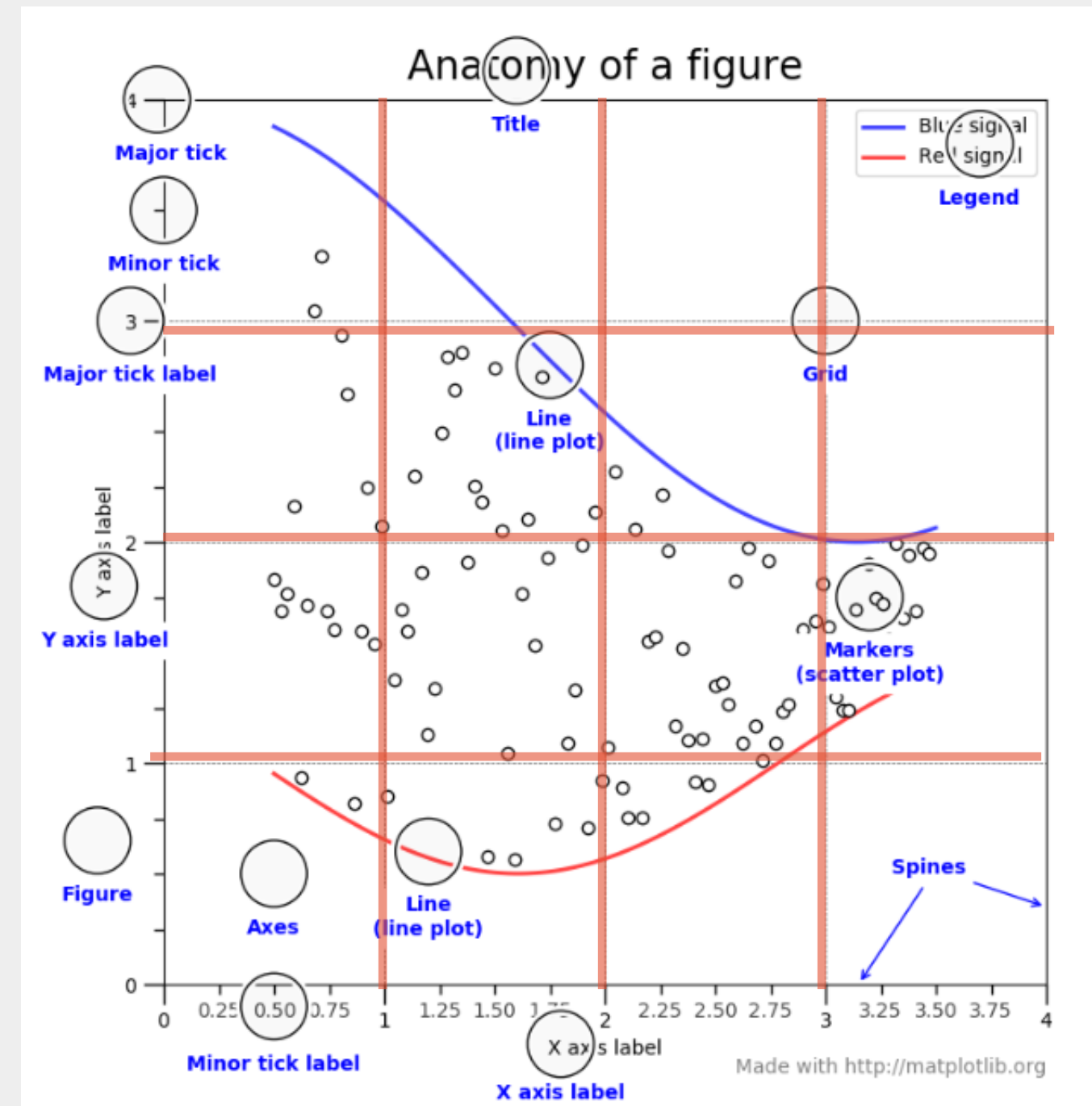
```
# plt.yticks([0, 2, 4])
```

범례

```
plt.legend()
```

그리드

```
plt.grid(True)
```




matplotlib


그래프 꾸미기


color


black	bisque	forestgreen	slategrey
dimgray	darkorange	limegreen	lightsteelblue
dimgray	burlywood	darkgreen	cornflowerblue
gray	antiquewhite	green	royalblue
grey	tan	lime	ghostwhite
darkgray	navajowhite	seagreen	lavender
darkgrey	blanchedalmond	mediumseagreen	midnightblue
silver	papayawhip	springgreen	navy
lightgray	moccasin	mintcream	darkblue
lightgrey	orange	mediumspringgreen	mediumblue
gainsboro	wheat	mediumaquamarine	blue
whitesmoke	oldlace	aquamarine	slateblue
white	floralwhite	turquoise	darkslateblue
snow	darkgoldenrod	lightseagreen	mediumslateblue
rosybrown	goldenrod	mediumturquoise	mediumpurple
lightcoral	cornsilk	azure	rebeccapurple
indianred	gold	lightcyan	blueviolet
brown	lemonchiffon	paleturquoise	indigo
firebrick	khaki	darkslategray	darkorchid
maroon	palegoldenrod	darkslategrey	darkviolet
darkred	darkkhaki	teal	mediumorchid
red	ivory	darkcyan	thistle
mistyrose	beige	aqua	plum
salmon	lightyellow	cyan	violet
tomato	lightgoldenrodyellow	darkturquoise	purple
darksalmon	olive	cadetblue	darkmagenta
coral	yellow	powderblue	fuchsia
orangered	olivedrab	lightblue	magenta
lightsalmon	yellowgreen	deepskyblue	orchid
sienna	darkolivegreen	skyblue	mediumvioletred
seashell	greenyellow	lightskyblue	deeppink
chocolate	chartreuse	steelblue	hotpink
saddlebrown	lawngreen	aliceblue	lavenderblush
sandybrown	honeydew	dodgerblue	palevioletred
peachpuff	darkseagreen	lightslategray	crimson
peru	palegreen	lightslategrey	pink
linen	lightgreen	slategray	lightpink

linestyle

 Solid
`plt.plot(x, y, '-')`

 Dashed
`plt.plot(x, y, '--')`

 Dotted
`plt.plot(x, y, ':')`

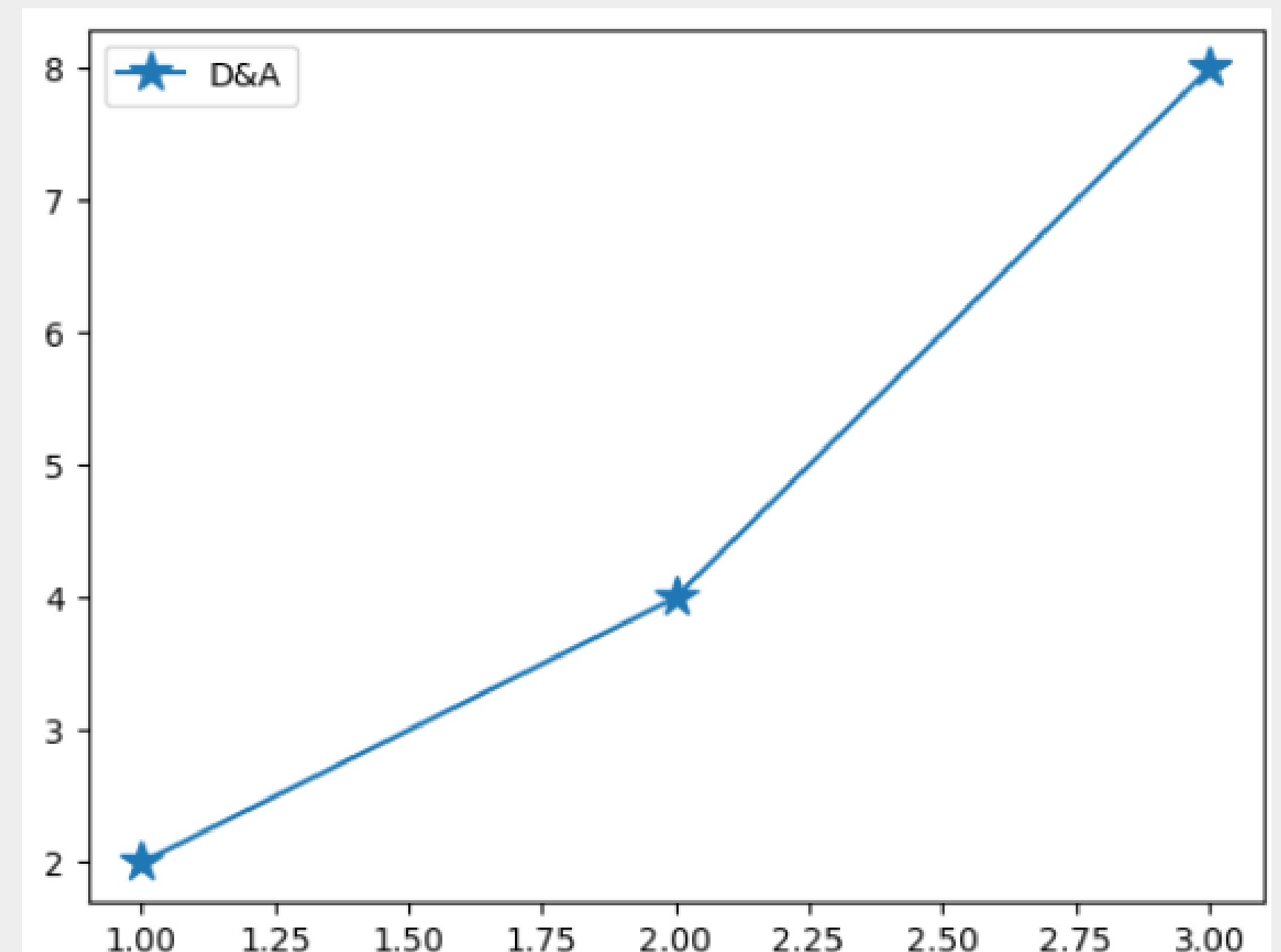
 Dash-dot
`plt.plot(x, y, '-.')`

matplotlib

그래프 꾸미기

marker

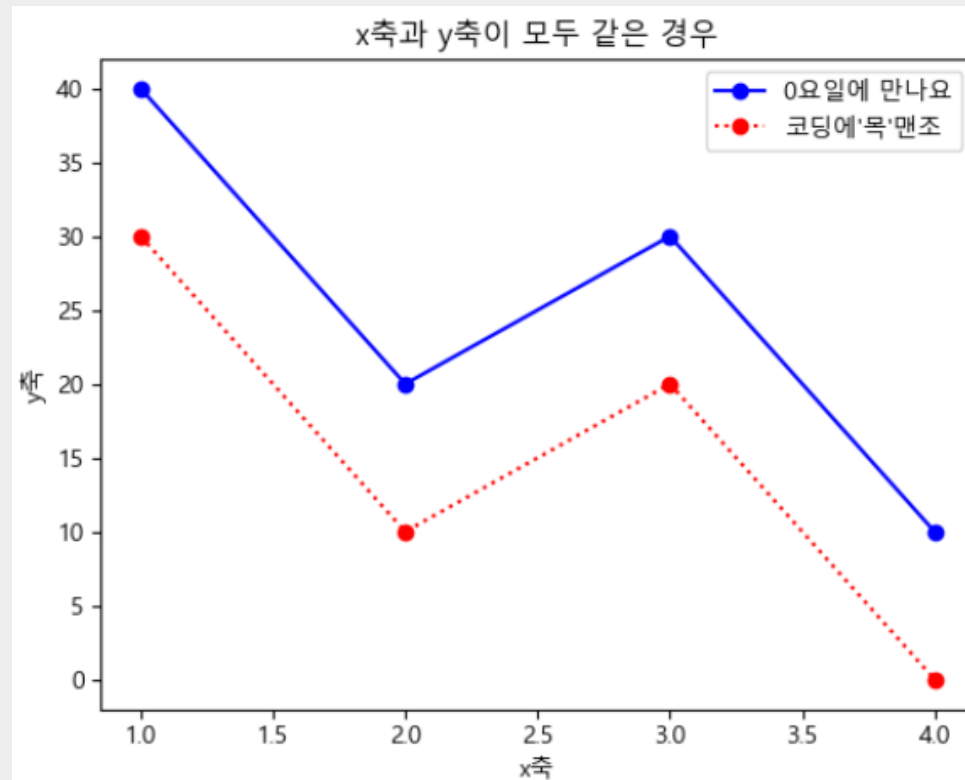
marker	symbol	description	"h"	hexagon1
"."	•	point	"H"	hexagon2
"."	•	pixel	"+"	plus
"o"	●	circle	"x"	x
"v"	▼	triangle_down	"x"	x (filled)
"^"	▲	triangle_up	"D"	diamond
"<"	◀	triangle_left	"d"	thin_diamond
">"	▶	triangle_right	" "	vline
"1"	⋿	tri_down	"_"	hline
"2"	⋿	tri_up	0 (TICKLEFT)	tickleft
"3"	⋿	tri_left	1 (TICKRIGHT)	tickright
"4"	⋿	tri_right	2 (TICKUP)	tickup
"8"	●	octagon	3 (TICKDOWN)	tickdown
"s"	■	square	4 (CARETLEFT)	caretleft
"p"	◆	pentagon	5 (CARETRIGHT)	caretright
"p"	⊕	plus (filled)	6 (CARETUP)	caretup
"*"	★	star	7 (CARETDOWN)	caretdown



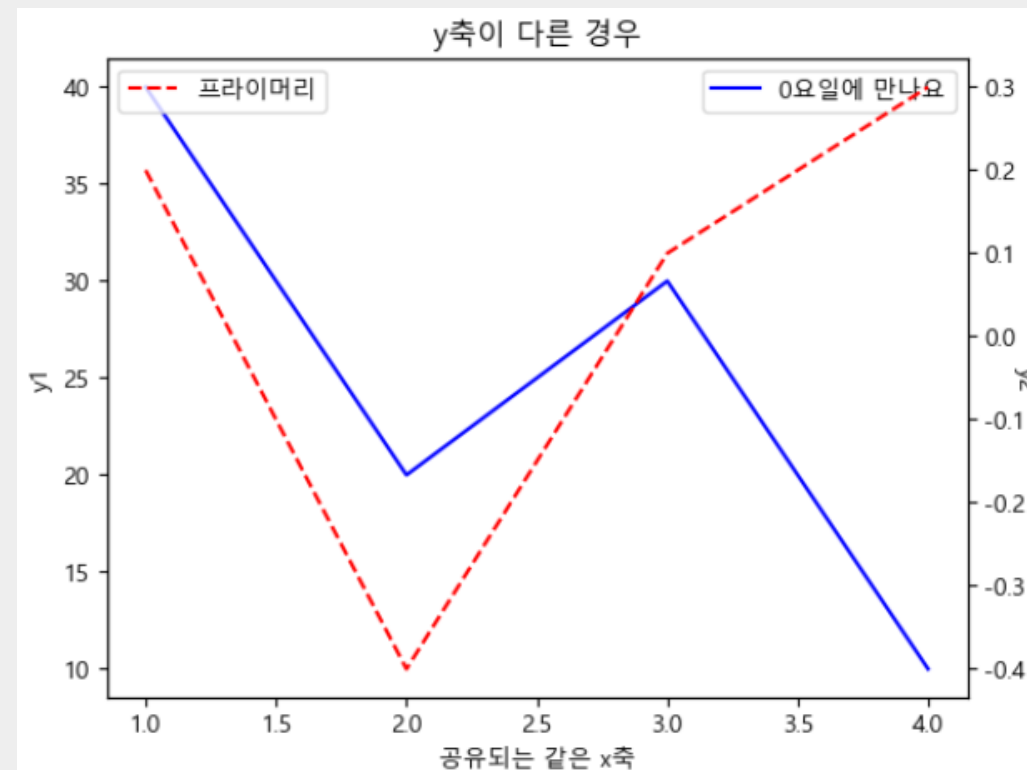
여러 그래프 그리기

겹쳐 그리기

x축과 y축이 모두 같은 경우

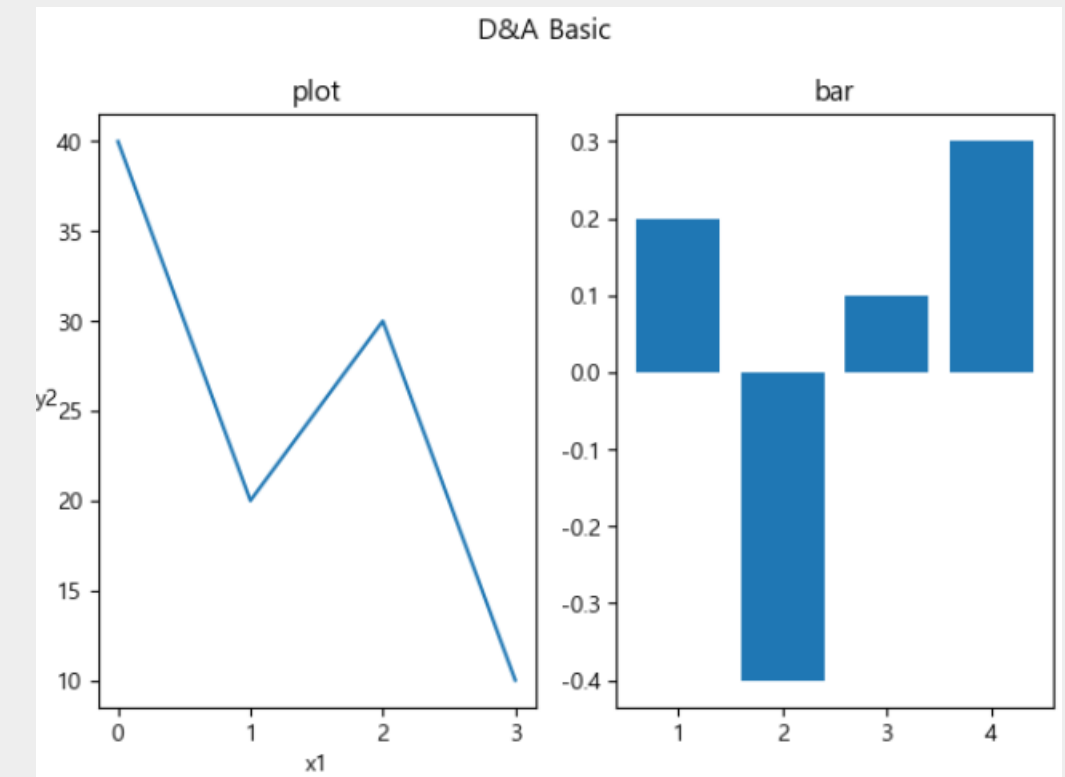


y축이 다른 경우 (subplots)



나란히 그리기

subplot, subplots



seaborn

seaborn

matplotlib 기반 시각화 라이브러리
통계 그래프를 그리기 위한 고급 인터페이스 제공



사용 방법

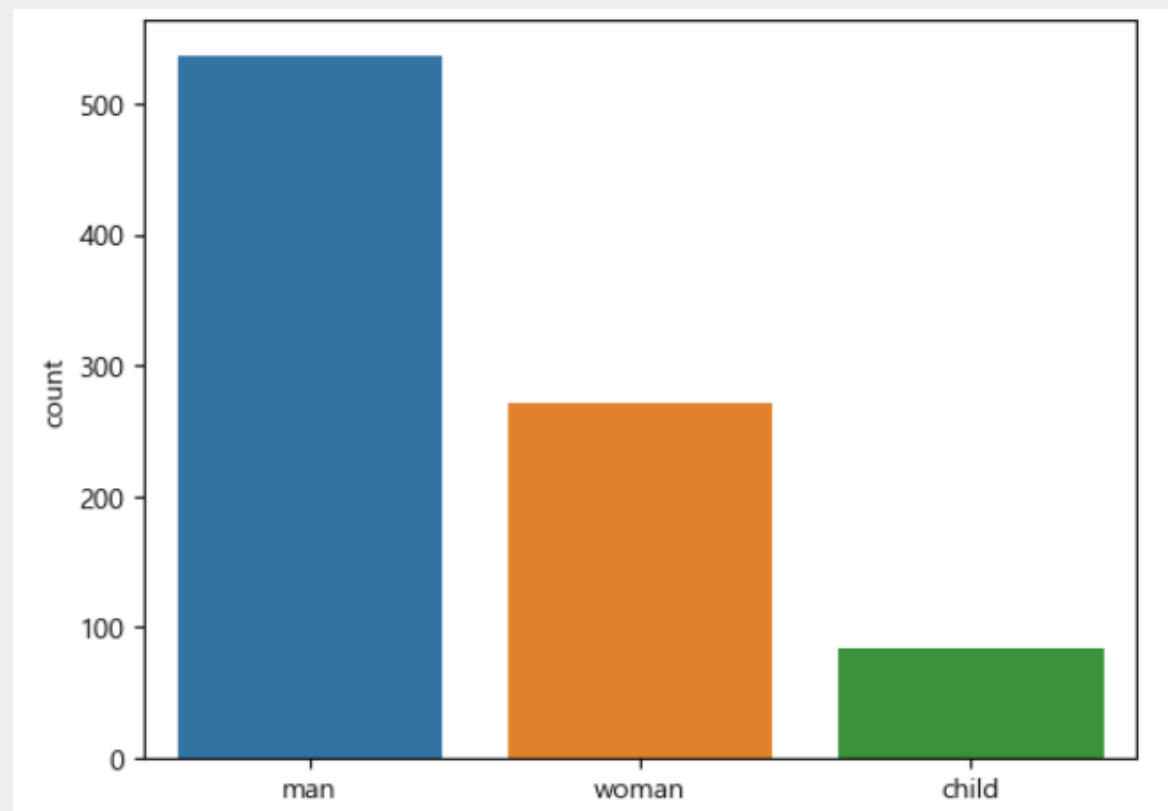
```
import seaborn as sns

# sns.유형()
sns.countplot()
```

seaborn 유형

countplot

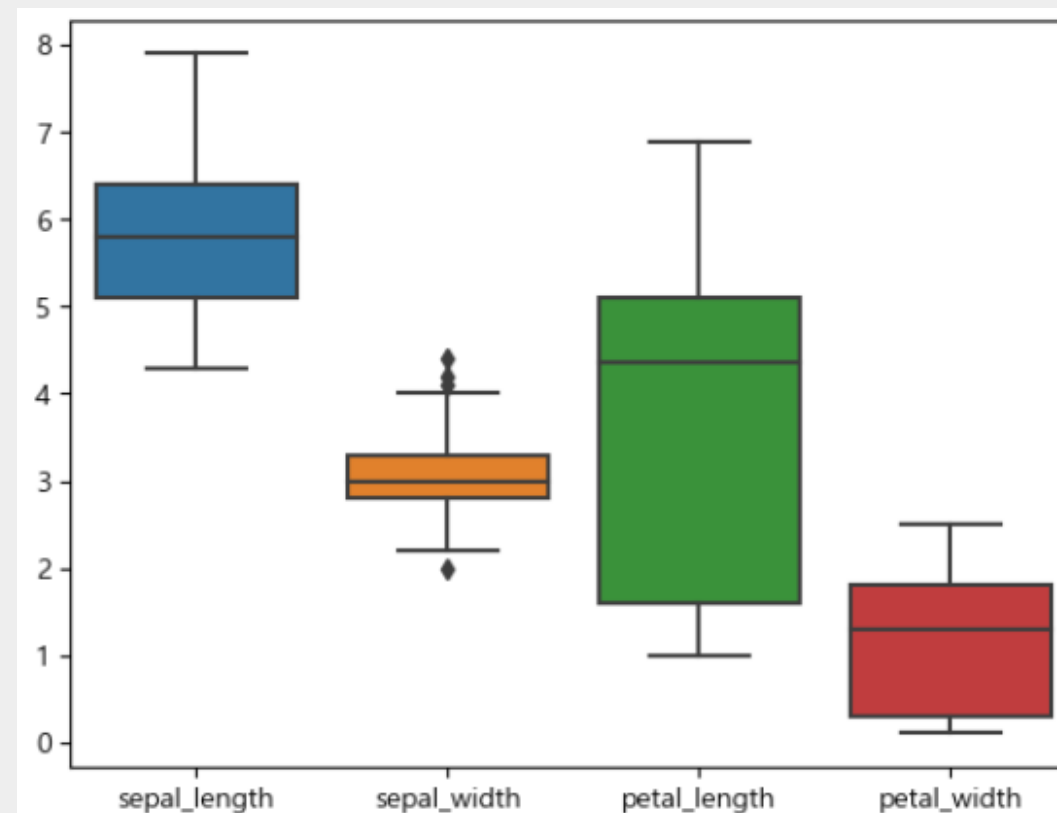
데이터 항목별 개수



```
sns.countplot(x, data)  
sns.countplot(data.x)
```

boxplot

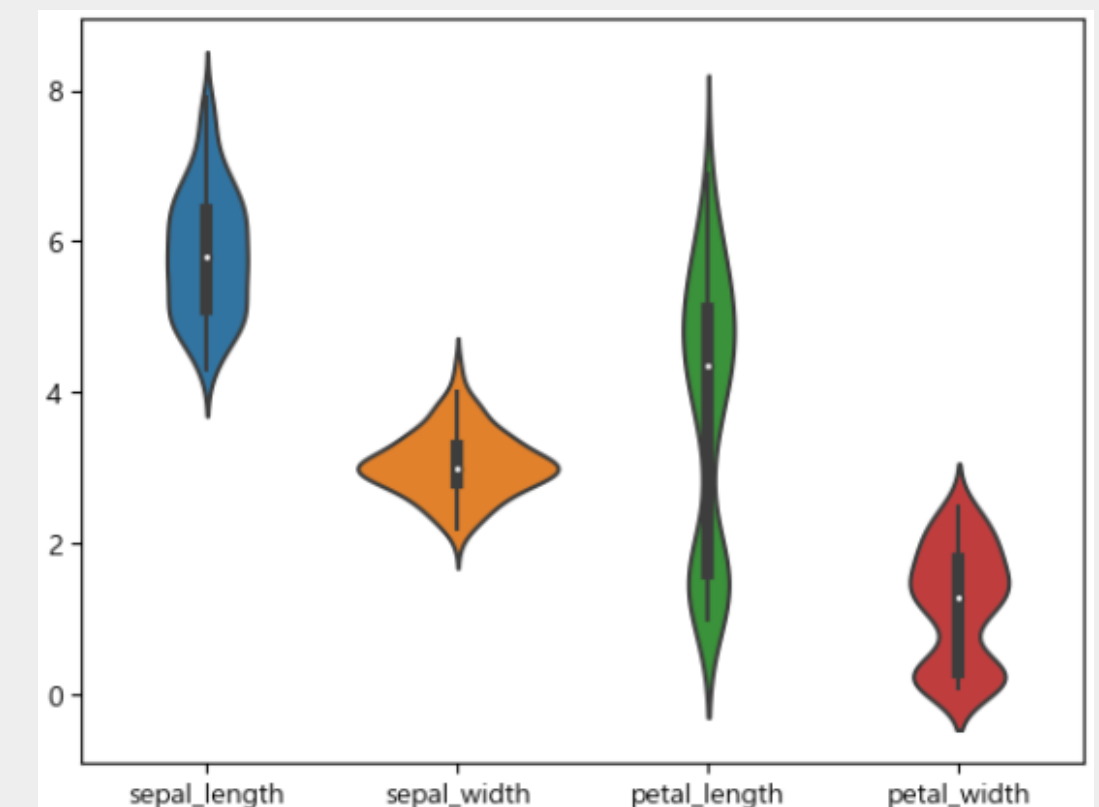
데이터의 분포나 전체 형상



```
sns.boxplot(x)  
sns.boxplot(data)
```

violinplot

boxplot + kde
데이터의 분포나 전체 형상

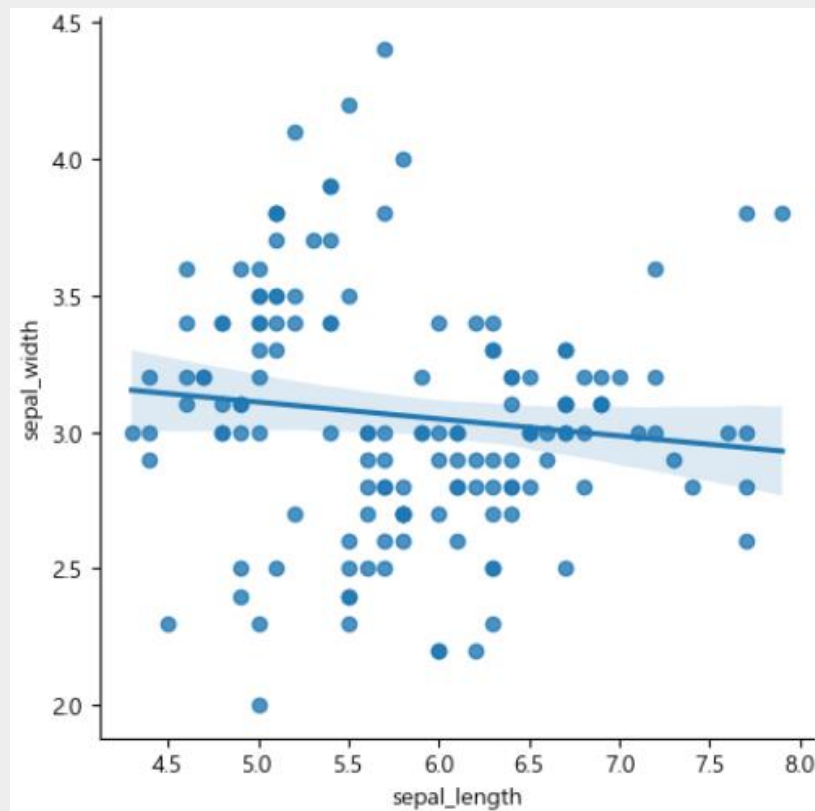


```
sns.violinplot(x)  
sns.violinplot(data)
```

seaborn 유형

Implot

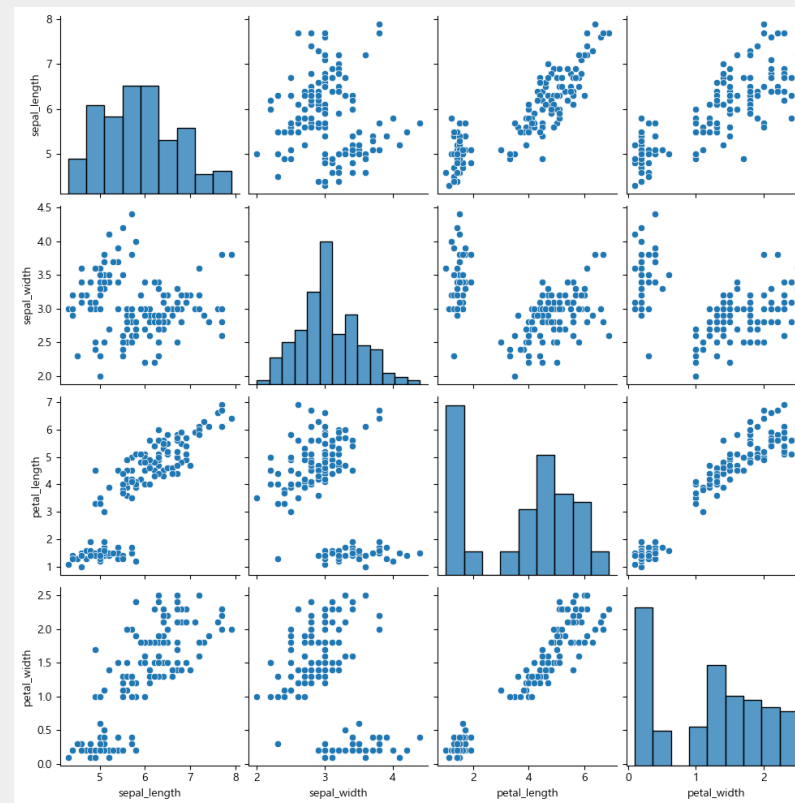
데이터 간의 선형관계
scatter plot + 추세선



```
sns.lmplot(x, y)
```

pairplot

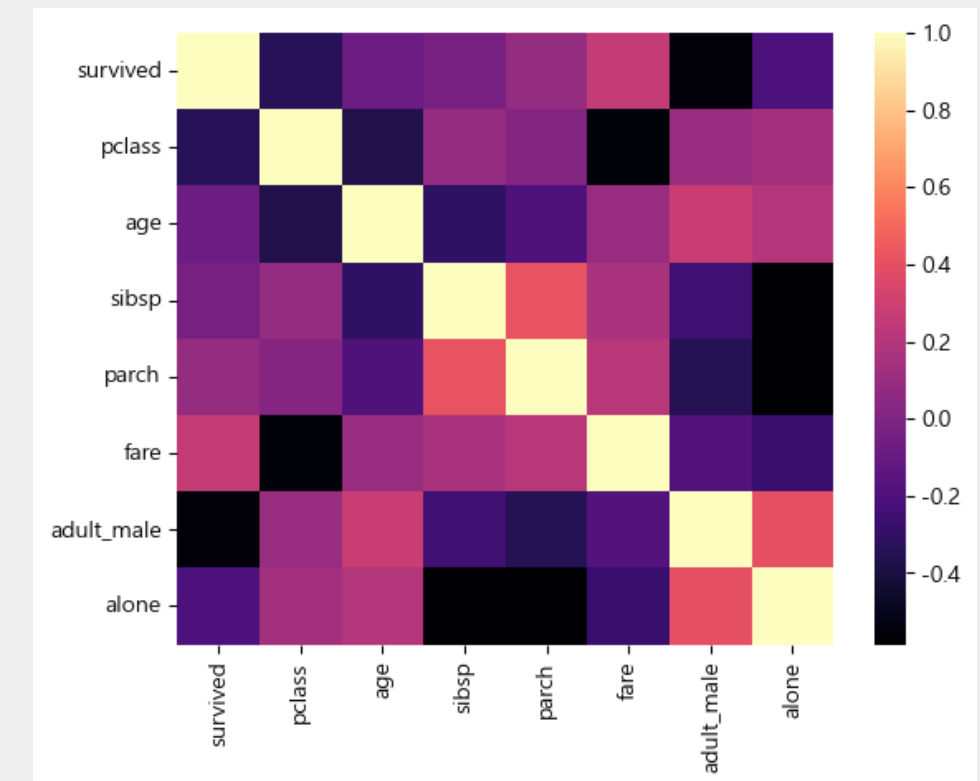
데이터 간의 상관관계
column이 같으면 hist, 다른 scatter



```
sns.pairplot(data)
```

heatmap

데이터 간의 상관관계
색으로 표현



```
sns.heatmap(x)
```

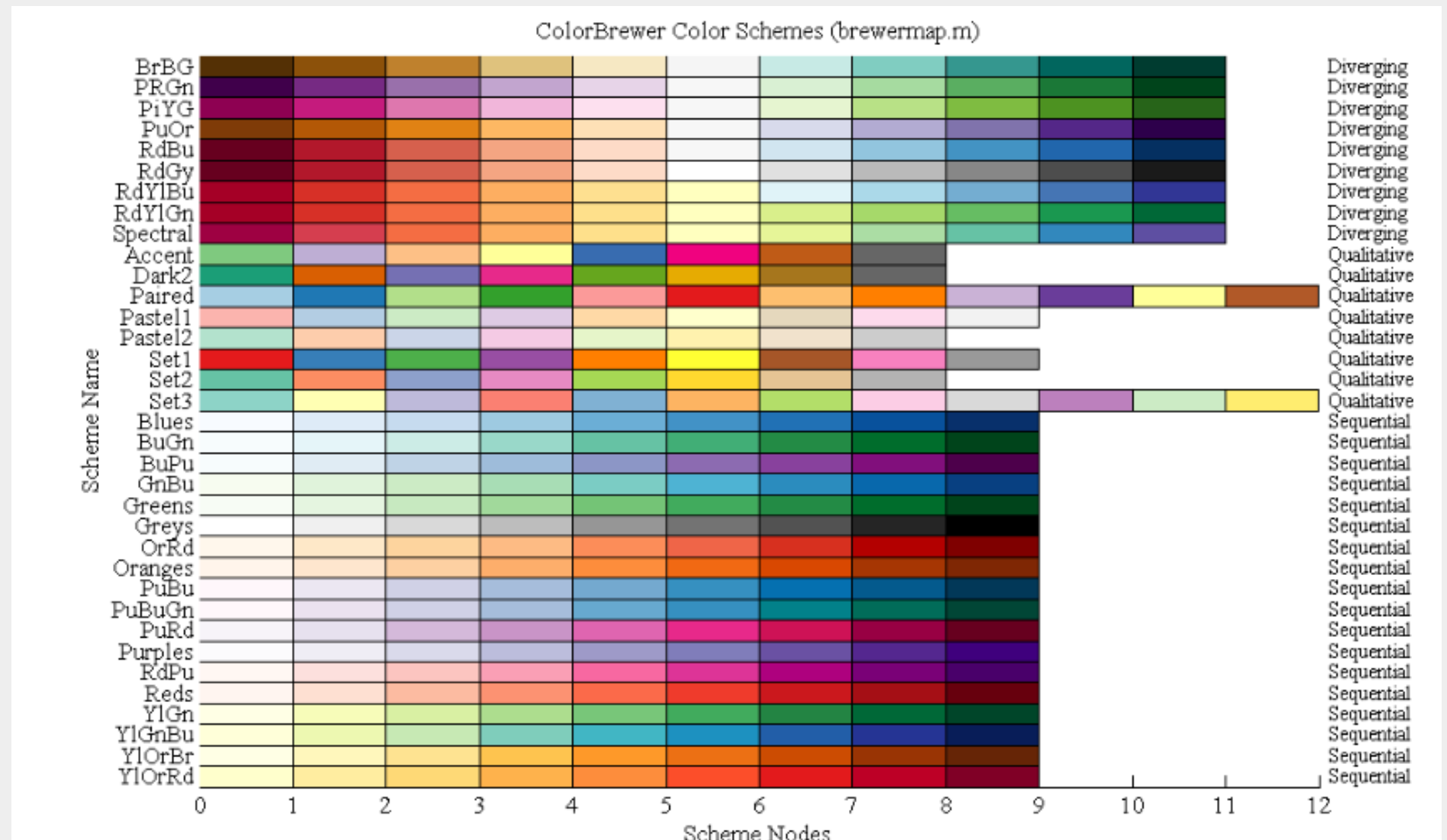
seaborn

그래프 꾸미기

color palette

seaborn은 스타일 지정을 위한
color palette 지원

```
sns.color_palette()
```



https://seaborn.pydata.org/tutorial/color_palettes.html

seaborn

그래프 꾸미기

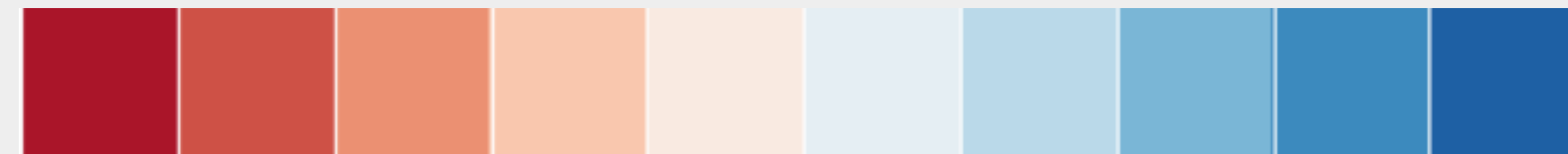
color palette

Qualitative



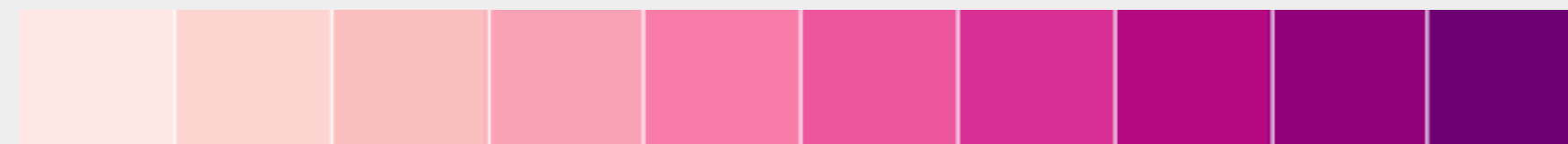
- 고유한 색상들로 이루어짐
- 범주형 데이터

Diverging



- 양 끝 색이 강조되도록 이루어짐
- 높고 낮음을 표시해야 하는 데이터

Sequential



- 밝은 색부터 어두운 색까지 차례대로 이루어짐
- 순서가 있는 데이터

6주차

팀 과제

1. EDA Competiton 데이터를 이용해 5개 이상 시각화 해보기 (라이브러리, 종류 무관)

- ipynb 파일에 자유 양식으로 작성

2. EDA Competiton 기획서 작성하기

REFERENCE

<https://pandas.pydata.org/>

<https://matplotlib.org/>

<https://seaborn.pydata.org/>

https://codetorial.net/matplotlib/set_linestyle.html

<https://www.codecademy.com/article/seaborn-design-ii>



2024 D&A

THANK YOU

2024.04.30

Basic Session 6차시

