모두를 위한 파이썬 프로그래밍

15주 빅데이터의 시각화

빅데이터의 시각화 이해

1

파이썬을 활용한 데이터 분석

목적

- 연관규칙 찾기 (예 : 유튜브, 넷플릭스, 기저기와 맥주)
  - 제품, 친구 추천서비스 (아마존, 페이스북 친구추천)
- SNS 감성 분류 및 맞춤 서비스 제공
- 번역 및 음성인식(시리, 왓슨, 구글번역 등)
- 통계적 방법을 통한 분석 및 예측
  - 범죄지역 예측(미국의 PredPol), 예측배송(아마존)
  - 댓글 봇(bot) 적용(네이버, 다음)
  - 소셜빅데이터분석, 송길영, <여기에 당신의 욕망이 보인다>

  https://www.youtube.com/watch?v=RapehyA5UWQ
- 퀀트투자

데이터 통계 시각화

왜? 데이터 시각화가 필요한가

■ **시각화는 인간의 뇌에 가장 높은 인상을 전달하는 수단이다**

- 시각 피질은 인간 두뇌에서 가장 큰 자극을 주며 수치로 사용하는 것보다 전달이 좋다
- 천개의 글자보다 한 개의 그림이 뛰어나다

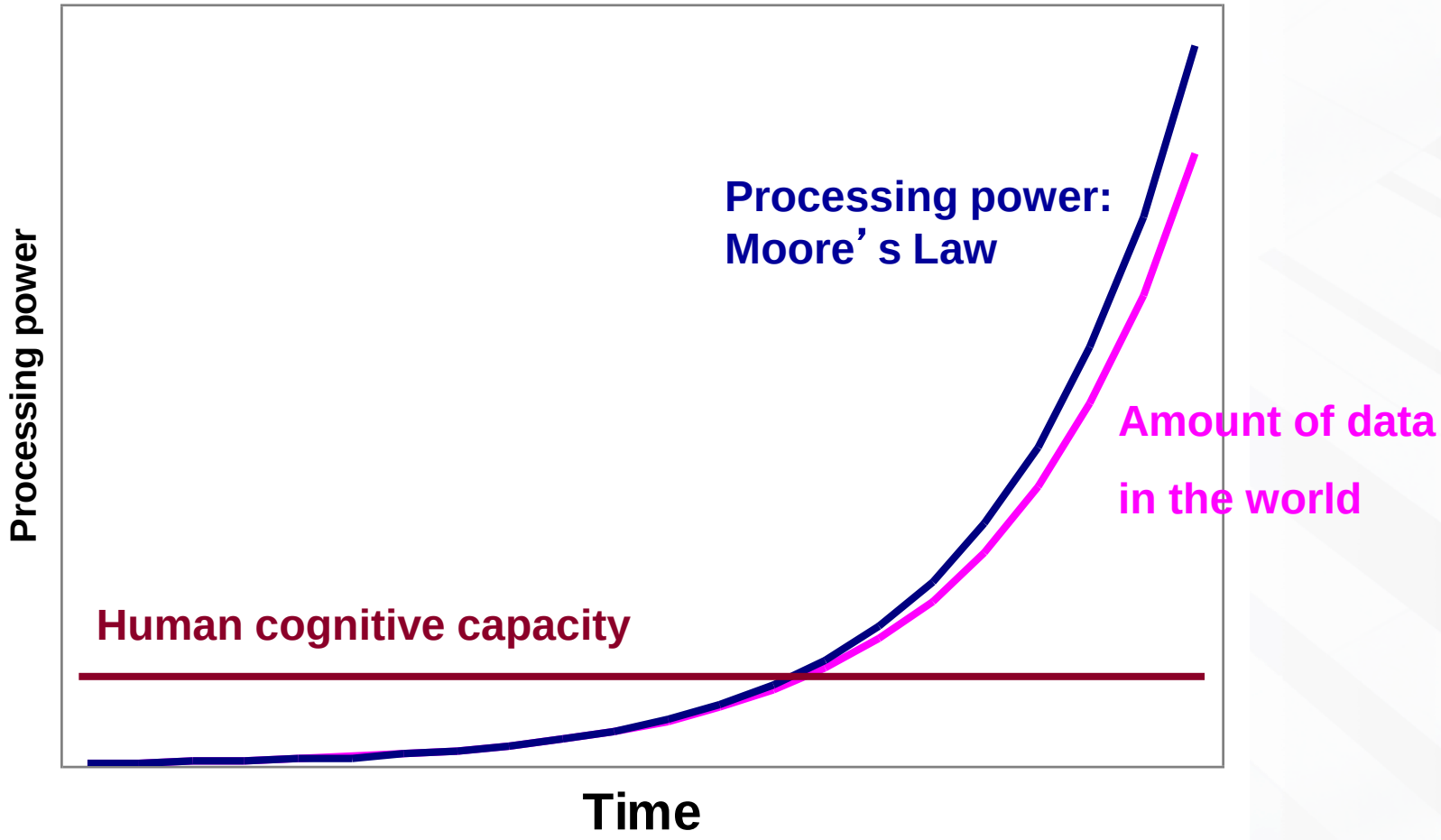
■ **차트로 다루기 어려운 빅데이터들은 특별한 시각화가 필요하다**

- 빅데이터 결과를 엑셀의 차트로 표현하기에는 한계가 있다

■ **동일한 통계적 수치라고 하더라도 시각화를 통해 다르게 그려질 수 있다. 즉, 다른 의미를 찾을 수 있다**

왜? 데이터 시각화가 필요한가

- **빅데이터의 증가에 따른 차트의 표현에 한계**



Idea adapted from "Less is More" by Bill Buxton (2001)

왜? 데이터 시각화가 필요한가

Anscombe's Quartet

- 4개의 데이터 집합 모두에 대해 동일한 통계값을 가짐

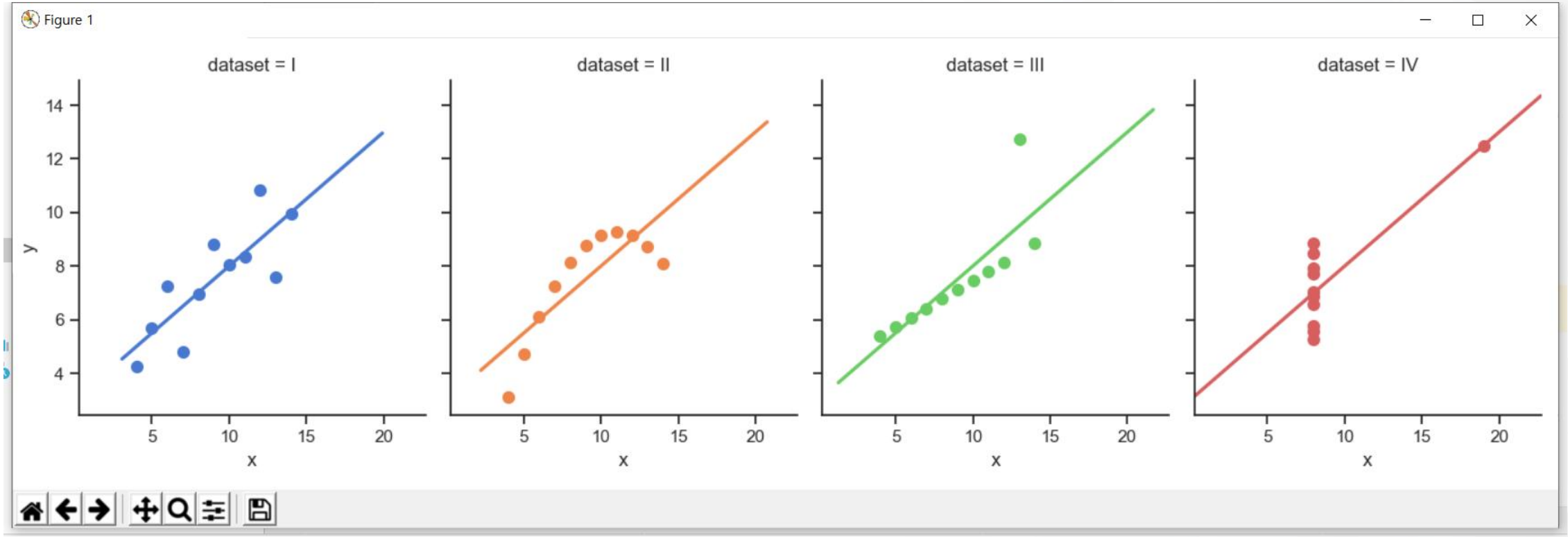
I		II		III		IV	
x	y	x	y	x	y	x	y
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.1	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.1	4	5.39	19	12.5
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

속성	값
평균 x	9
x의 샘플 분산 : s_x^2	11
평균 y	7.50
y의 샘플 분산 : s_y^2	4.125
x와 y 사이의 상관 관계	0.816
선형 회귀 선	$y = 3.00 + 0.500x$
선형 회귀의 결정 계수 : R^2	0.67

왜? 데이터 시각화가 필요한가

- **Anscombe's Quartet 예제**

  https://seaborn.pydata.org/examples/anscombes_quartet.html

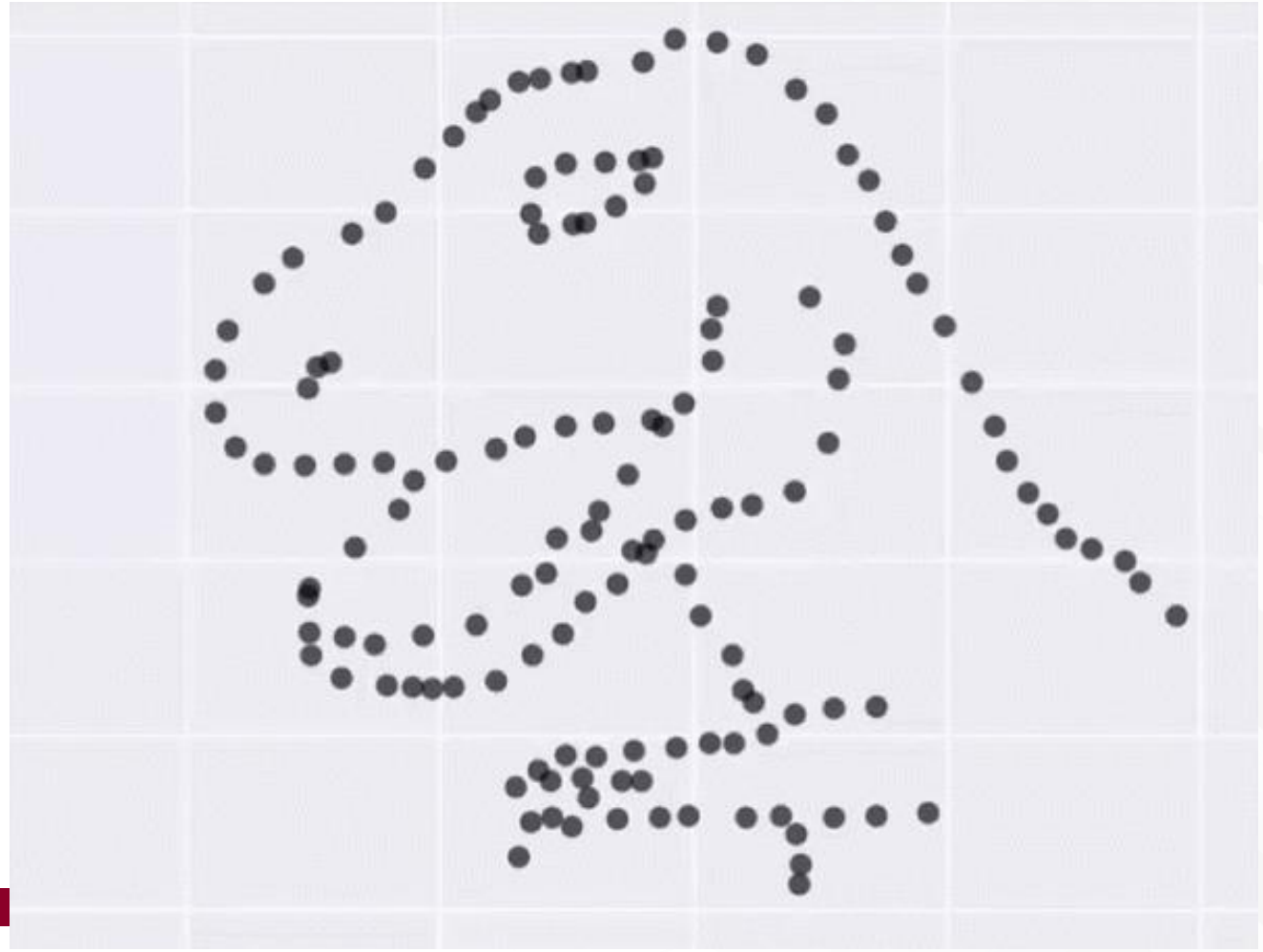


Same Stats, Different Graphs

https://www.autodeskresearch.com/publications/samestats

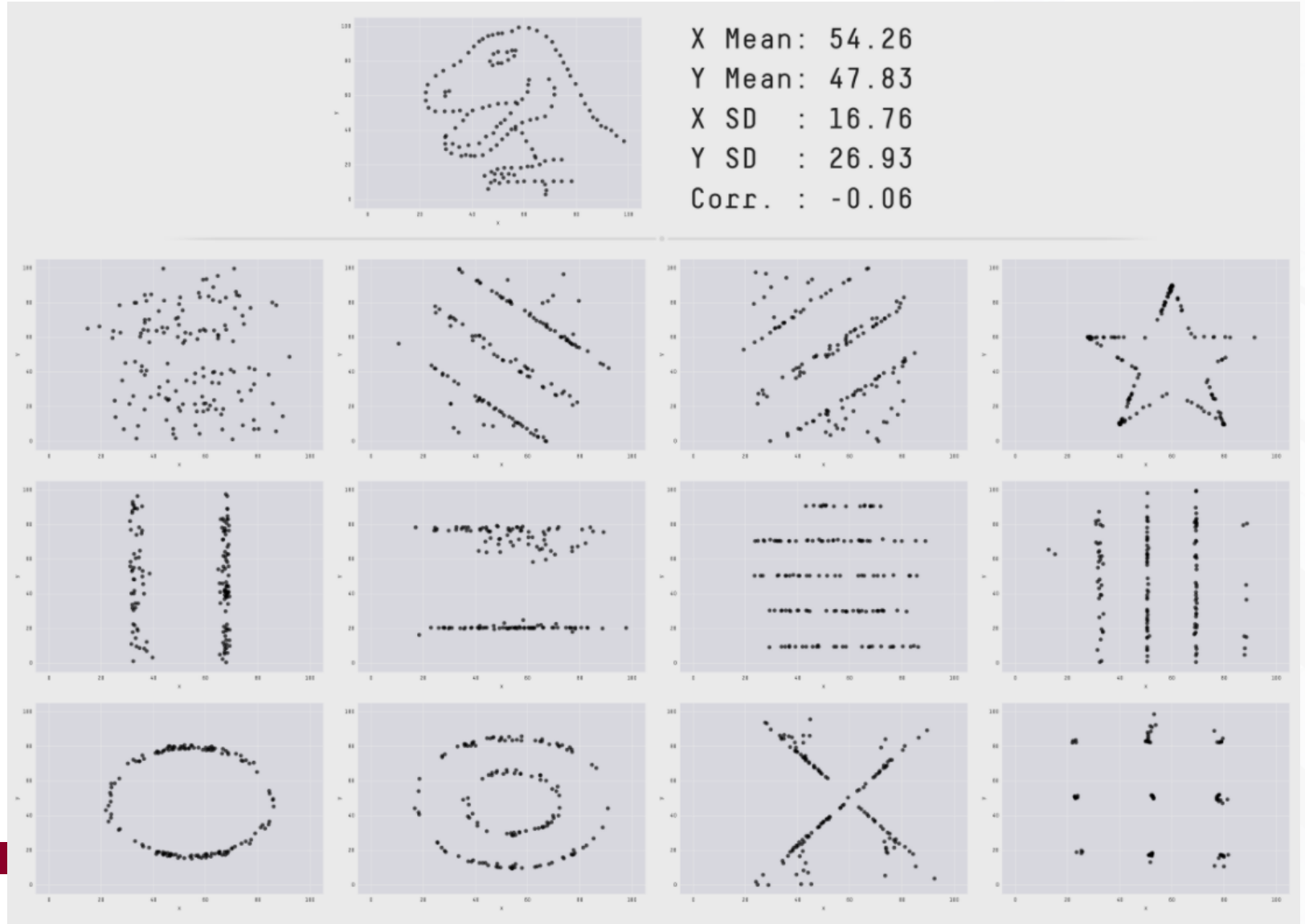
- **평균, 표준편차, 상관계수가 모두 동일한 그림**

```
X Mean: 54.26
Y Mean: 47.83
X SD   : 16.76
Y SD   : 26.93
Corr.  : -0.06
```



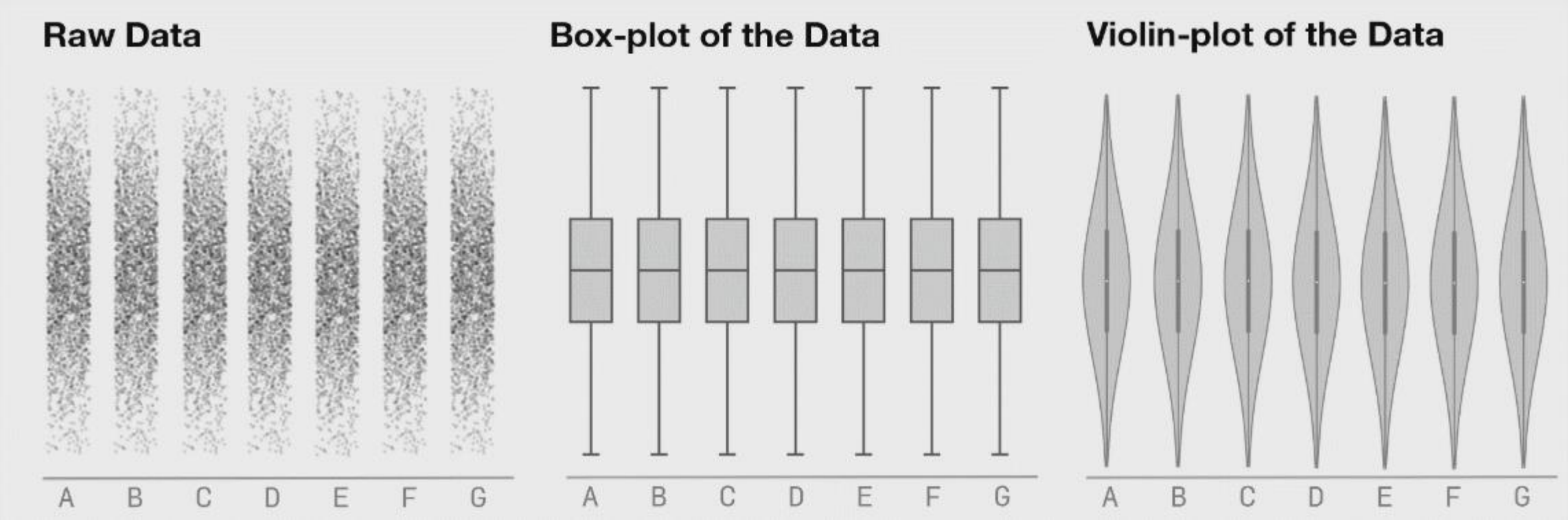
Same Stats, Different Graphs

https://www.autodeskresearch.com/publications/samestats



Same Stats, Different Graphs

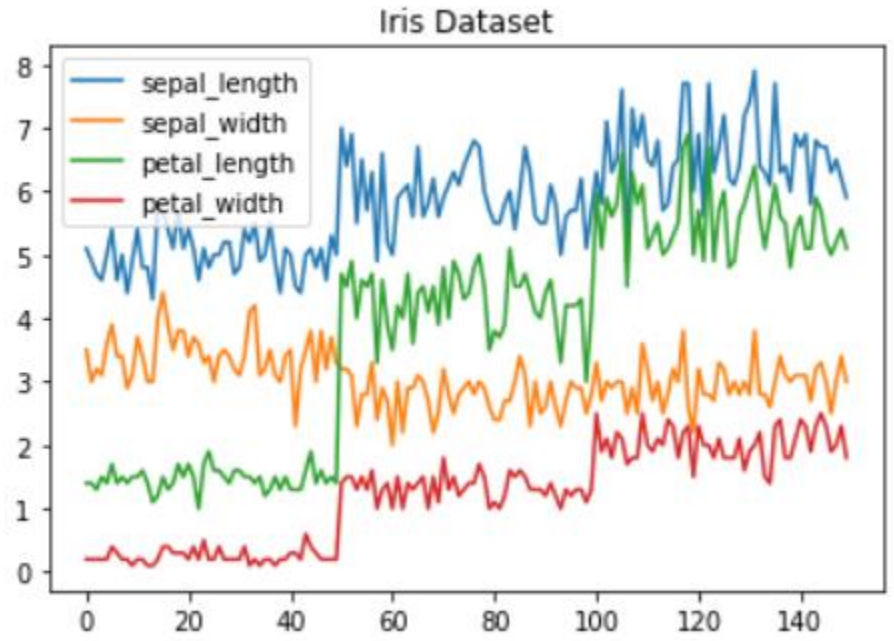
- 데이터 집합을 잘 표현할 수 있는 시각화 플롯을 사용해야 함



2 파이썬 시각화 도구

popular plotting libraries

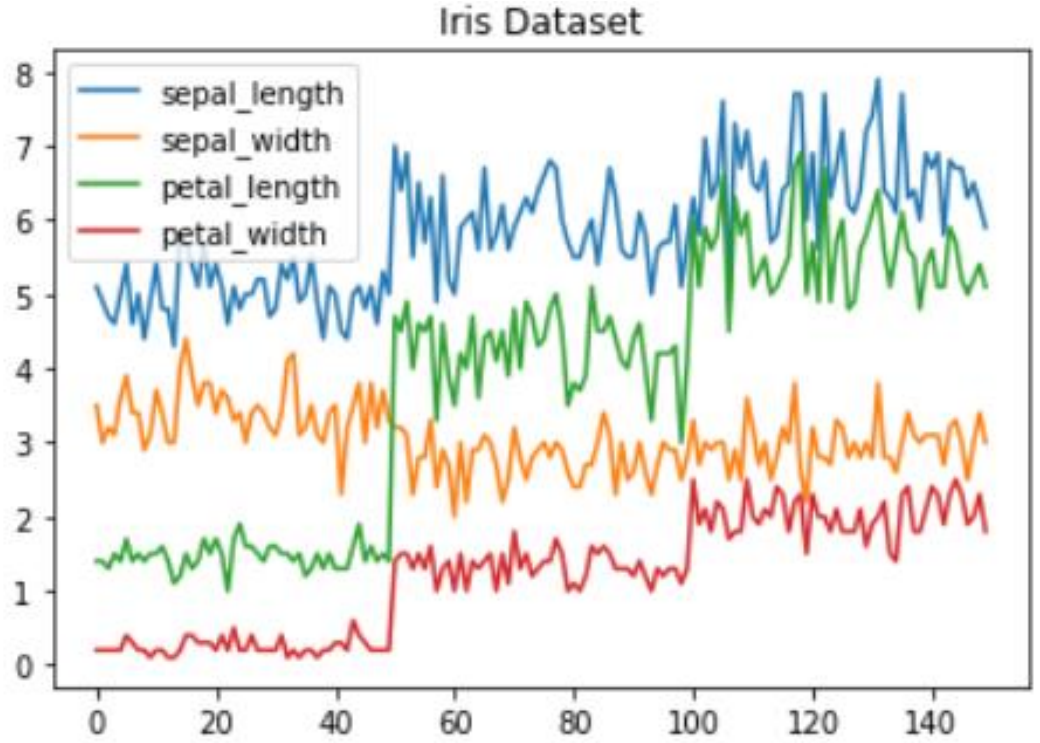
- **Matplotlib**
  - 기본 수준의 그리기 제공




```
# get columns to plot
columns = iris.columns.drop(['class'])
# create x data
x_data = range(0, iris.shape[0])
# create figure and axis
fig, ax = plt.subplots()
# plot each column
for column in columns:
    ax.plot(x_data, iris[column], label=column)
# set title and legend
ax.set_title('Iris Dataset')
ax.legend()
```

popular plotting libraries

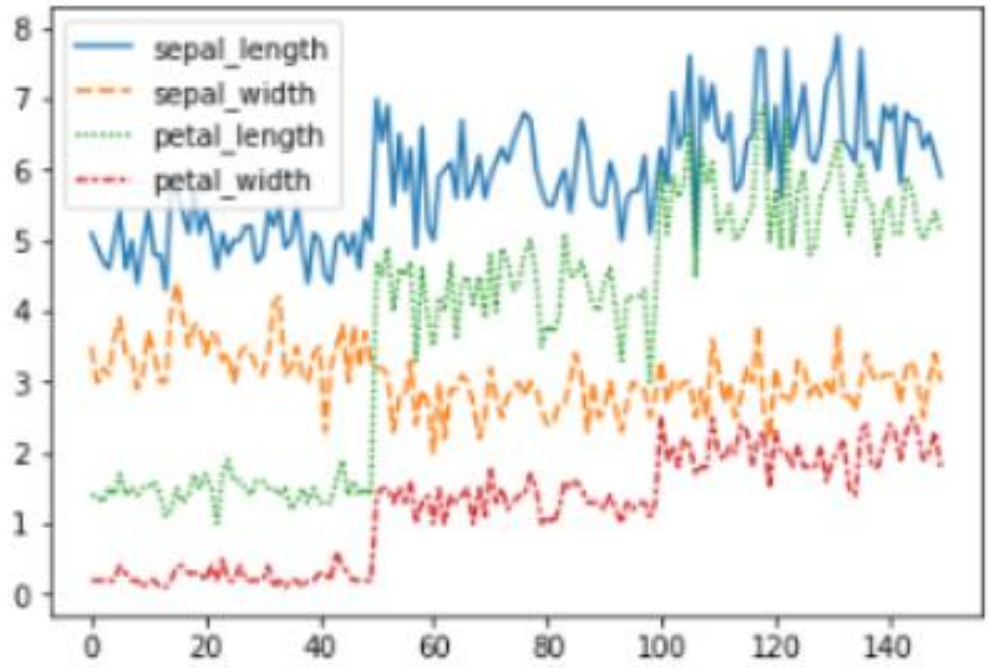
- **Pandas**
  - Matplotlib에 기반하여 쉬운 인터페이스 제공




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

iris = pd.read_csv("iris.csv", index_col=0)
df = pd.DataFrame(iris)

df.plot.line(title='Iris Dataset')

plt.show()
```

popular plotting libraries

- **Seaborn**
  - Matplotlib을 기반으로 다양한 색상 테마와 통계용 차트 등의 기능을 추가한 시각화 패키지

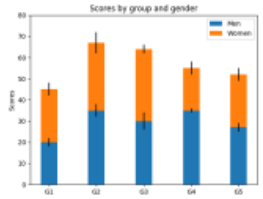



```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

iris = pd.read_csv("iris.csv", index_col=0)
df = pd.DataFrame(iris)

sns.lineplot(data=iris)

plt.show()
```

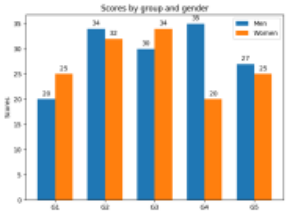
Matplotlib

https://matplotlib.org/

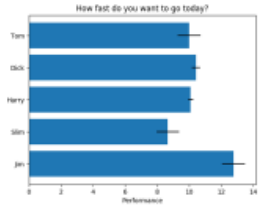
- 기본 시각화 도구



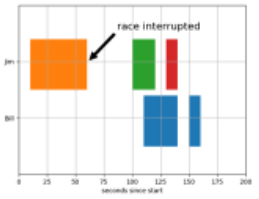
Stacked Bar Graph



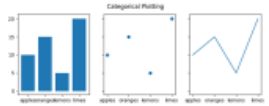
Grouped bar chart with labels



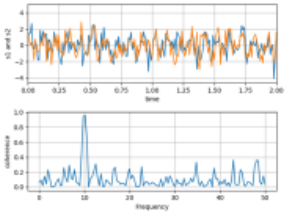
Horizontal bar chart



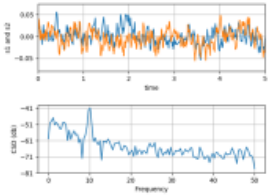
Broken Barh



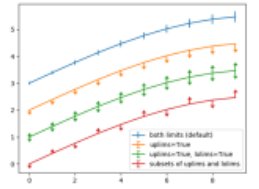
Plotting categorical variables



Plotting the coherence of two signals



CSD Demo

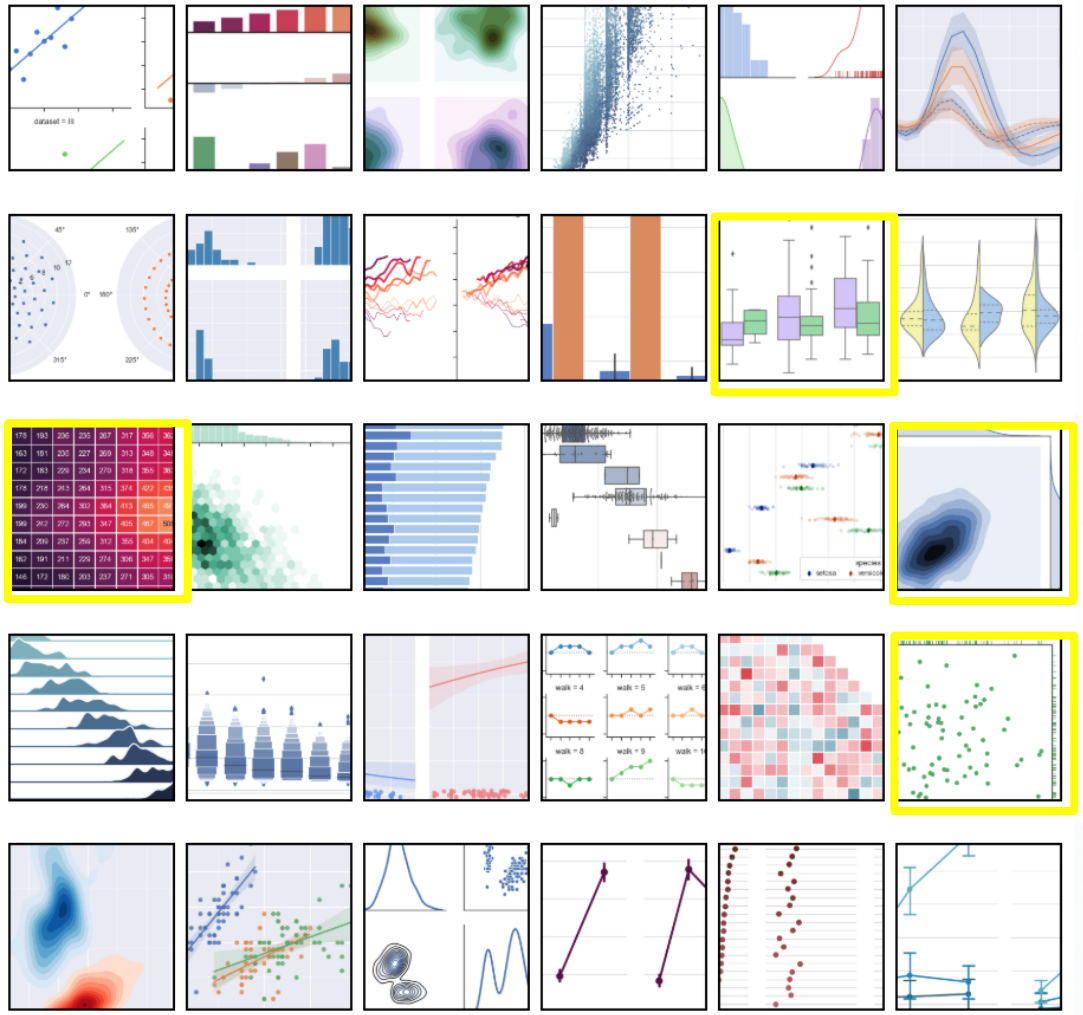


Errorbar limit selection

Seaborn

https://seaborn.pydata.org/

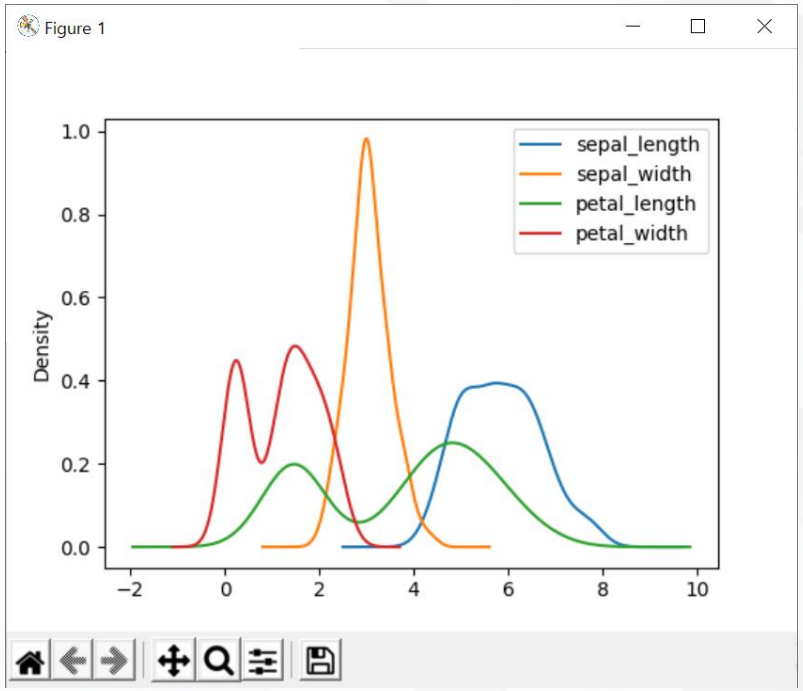
- Matplotlib보다 심플한 코딩과 향상된 데이터 시각화 패키지



Pandas 시각화 실습

- Matplotlib보다 심플한 코딩과 향상된 데이터 시각화 패키지

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

iris = pd.read_csv("iris.csv", index_col=0)
df = pd.DataFrame(iris)
df.plot(kind='kde') # kind = { bar, box, hist, kde...}

print(df)
plt.show()
```



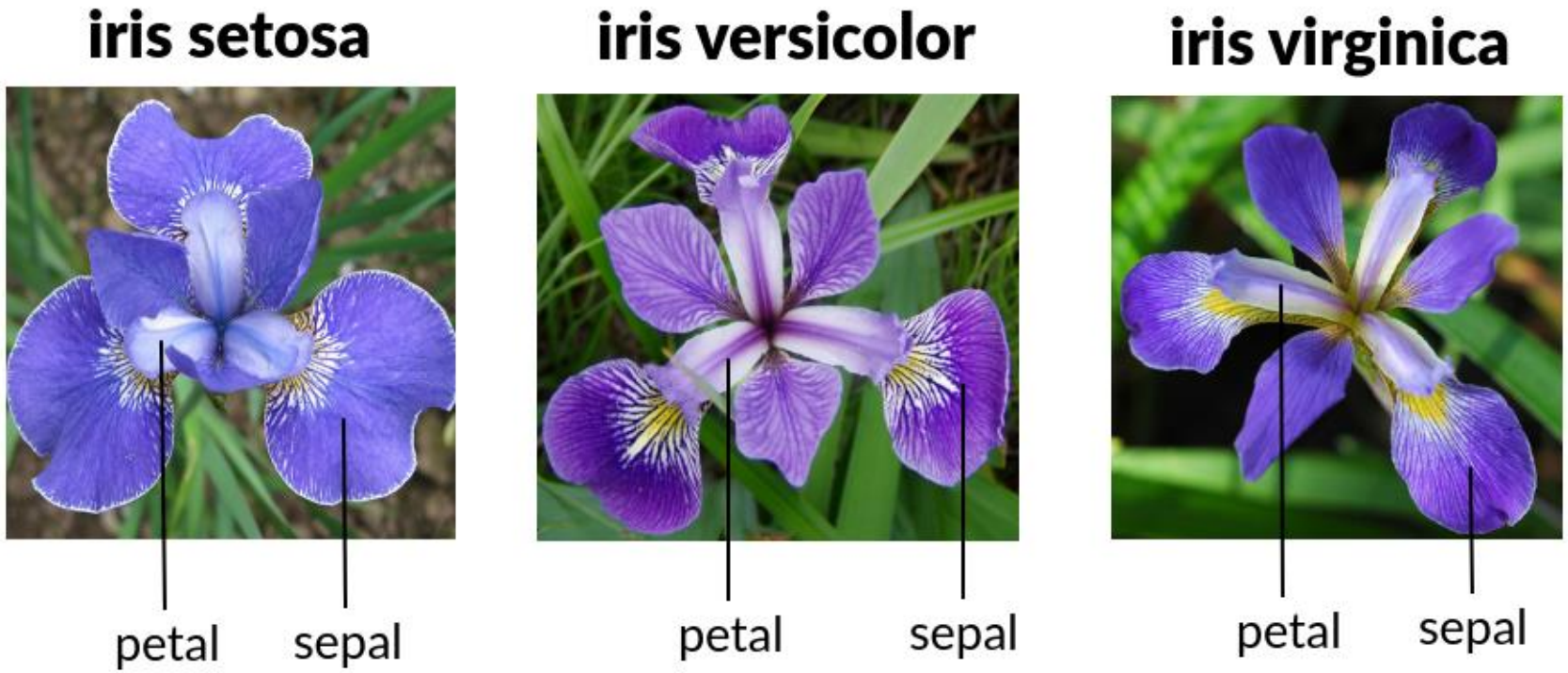
Seabone을 이용한 시각화

3

Seaborn 활용

- 3가지 붓꽃 품종을 시각화로 구분하기

```
iris = sns.load_dataset("iris")          # 붓꽃 데이터
titanic = sns.load_dataset("titanic")    # 타이타닉호 데이터
tips = sns.load_dataset("tips")          # 팁 데이터
flights = sns.load_dataset("flights")    # 여객운송 데이터
```



Seaborn 활용

- iris 데이터 확인

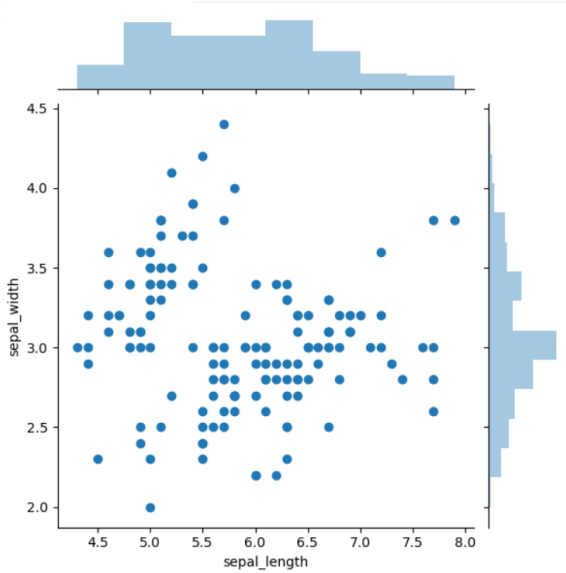
```
import seaborn as sns
import matplotlib.pyplot as plt
iris = sns.load_dataset('iris')
print(iris)
```

```
     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

[150 rows x 5 columns]
```

Seaborn - jointplot

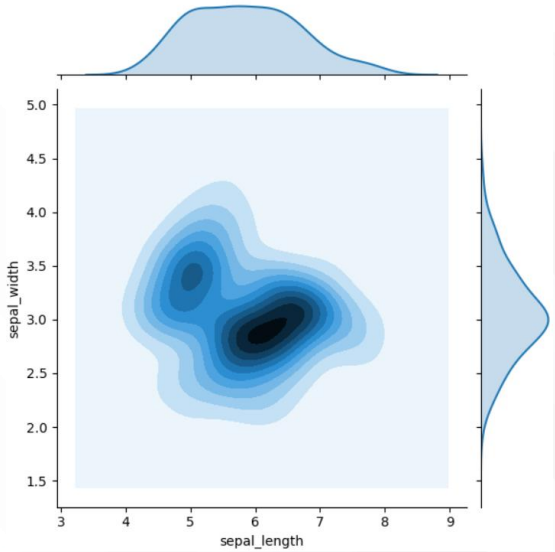
- 두 변수간의 산점도(Scatterplots)

```
import seaborn as sns
import matplotlib.pyplot as plt

iris = sns.load_dataset('iris')

sns.jointplot(x='sepal_length', y='sepal_width', data=iris)
plt.show()
```



- Kernel density estimation

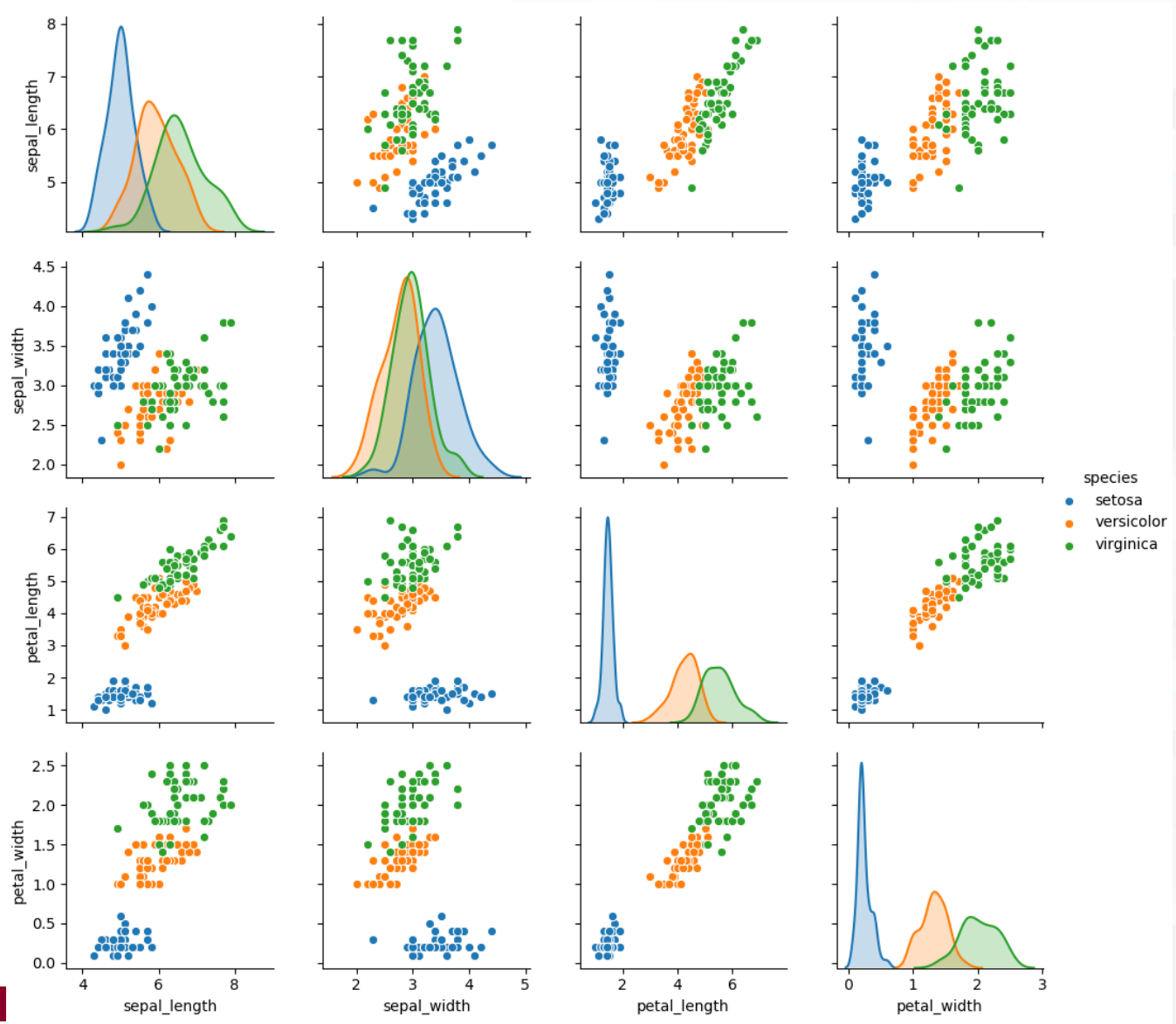
```
sns.jointplot(x='sepal_length',
y='sepal_width', data=iris, kind='kde')
plt.show()
# kind='hex'
```



Seaborn - pairplot

- **품종별 시각화로 구분하기**

```
sns.pairplot(iris, hue='species')
plt.show()
```



Seaborn - heatmap

■ **타이타닉 탑승자 명단으로 선실등급별 탑승자 현황을 히트맵으로 그리기**

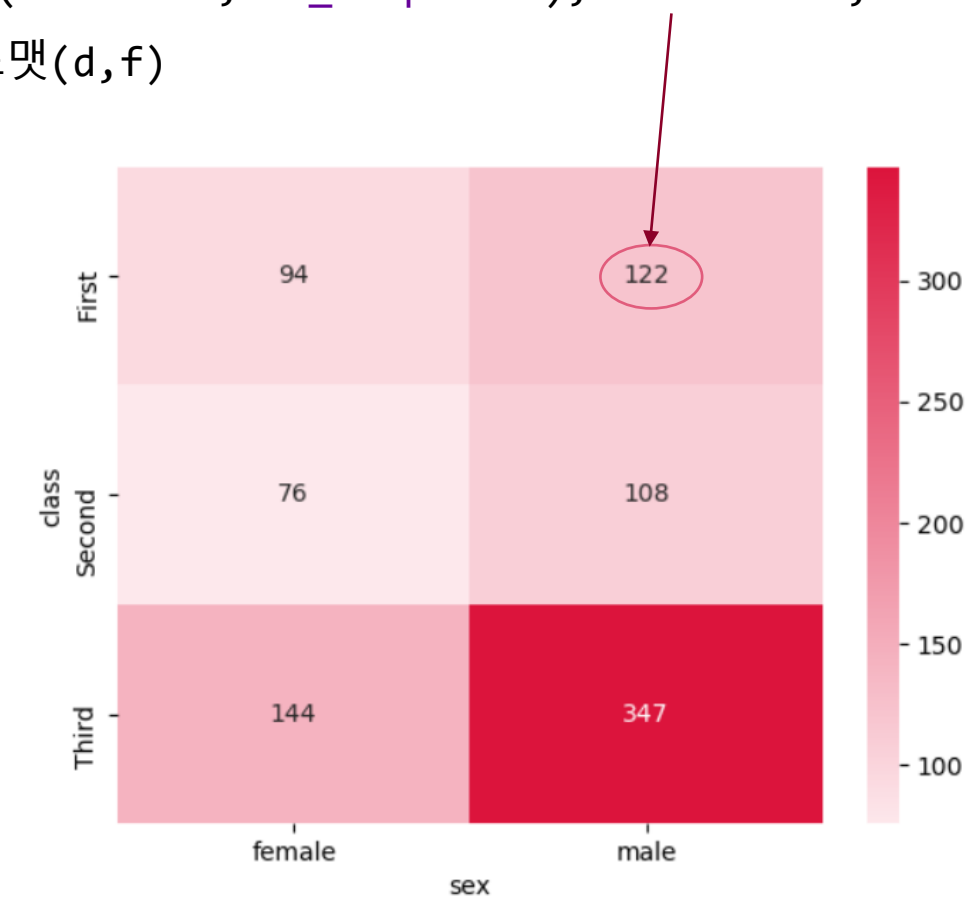
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
5	0	3	male	NaN	0	0	8.4583	Q	Third	man	True	NaN	Queenstown	no	True
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	E	Southampton	no	True
7	0	3	male	2.0	3	1	21.0750	S	Third	child	False	NaN	Southampton	no	False
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	NaN	Southampton	yes	False
9	1	2	female	14.0	1	0	30.0708	C	Second	child	False	NaN	Cherbourg	yes	False
10	1	3	female	4.0	1	1	16.7000	S	Third	child	False	G	Southampton	yes	False
11	1	1	female	58.0	0	0	26.5500	S	First	woman	False	C	Southampton	yes	True
12	0	3	male	20.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southampton	no	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southampton	yes	True
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southampton	no	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbourg	yes	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenstown	no	True

891 rows × 15 columns

Seaborn - heatmap

```
titanic = sns.load_dataset('titanic')
titanic_size = titanic.pivot_table(index='class', columns='sex', aggfunc='size')
# aggfunc(Aggregate Function)
sns.heatmap(titanic_size, cmap=sns.light_palette('crimson', as_cmap=True), annot=True, fmt='d')
# 크림슨색, 전체 맵색상 밴런싱 적용, 주석, 주석의 포맷(d,f)

print(titanic_size)
plt.show()
```

sex	female	male
class		
First	94	122
Second	76	108
Third	144	347



Seaborn - heatmap

■ **연도별 비행기 탑승자 수**

```
flights = sns.load_dataset('flights')
print(flights)
```

```
     year      month  passengers
0    1949    January         112
1    1949   February         118
2    1949      March         132
3    1949      April         129
4    1949        May         121
..    ...        ...         ...
139  1960     August         606
140  1960  September         508
141  1960    October         461
142  1960   November         390
143  1960   December         432

[144 rows x 3 columns]
```

Seaborn - heatmap

- **연도별 비행기 탑승자 수**

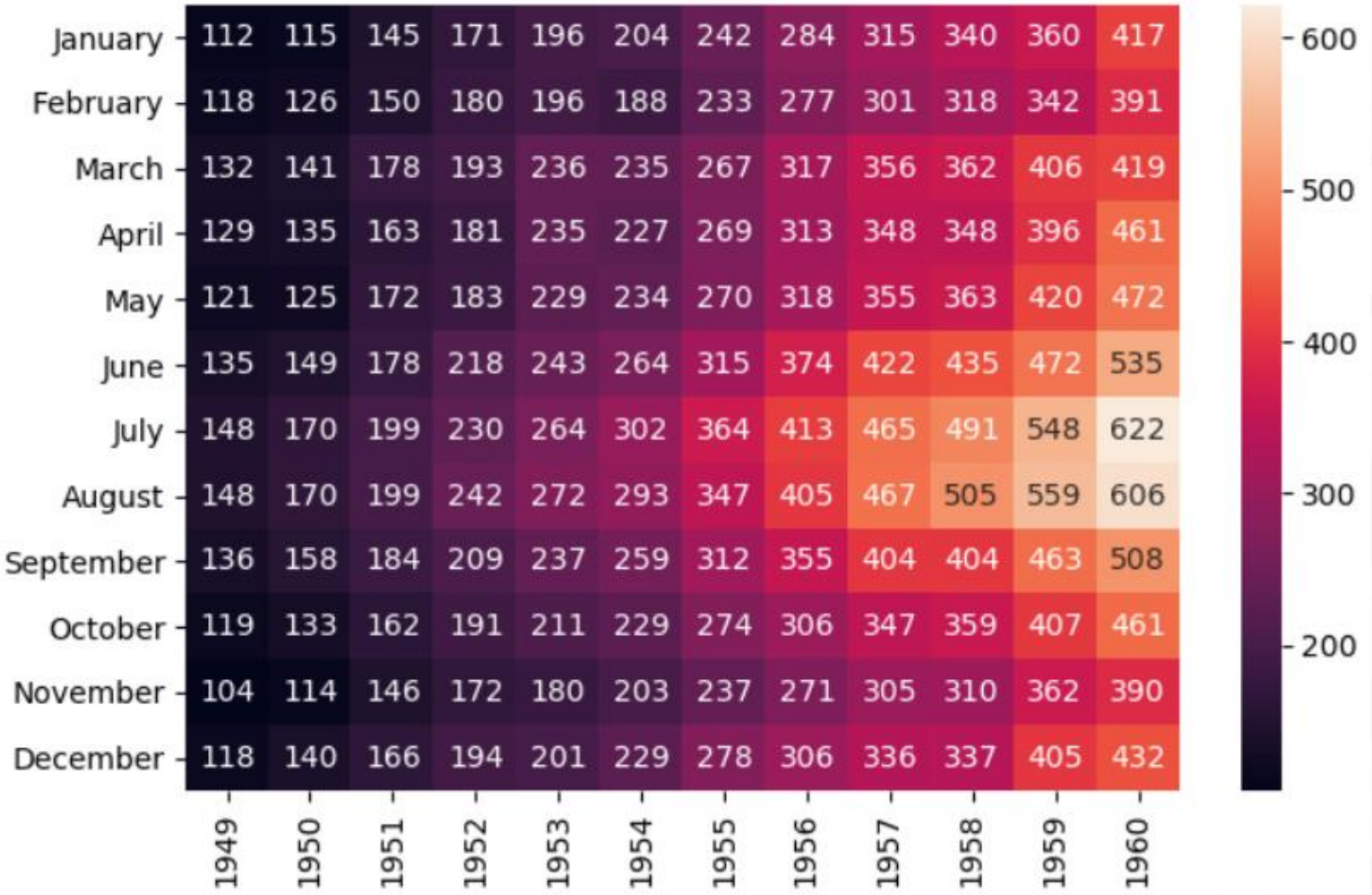
```
import seaborn as sns
import matplotlib.pyplot as plt
flights = sns.load_dataset('flights')
print(flights)
```

year	1949	1950	1951	1952	1953	...	1956	1957	1958	1959	1960
month						...					
January	112	115	145	171	196	...	284	315	340	360	417
February	118	126	150	180	196	...	277	301	318	342	391
March	132	141	178	193	236	...	317	356	362	406	419
April	129	135	163	181	235	...	313	348	348	396	461
May	121	125	172	183	229	...	318	355	363	420	472
June	135	149	178	218	243	...	374	422	435	472	535
July	148	170	199	230	264	...	413	465	491	548	622
August	148	170	199	242	272	...	405	467	505	559	606
September	136	158	184	209	237	...	355	404	404	463	508
October	119	133	162	191	211	...	306	347	359	407	461
November	104	114	146	172	180	...	271	305	310	362	390
December	118	140	166	194	201	...	306	336	337	405	432

```
[12 rows x 12 columns]
```

Seaborn - heatmap

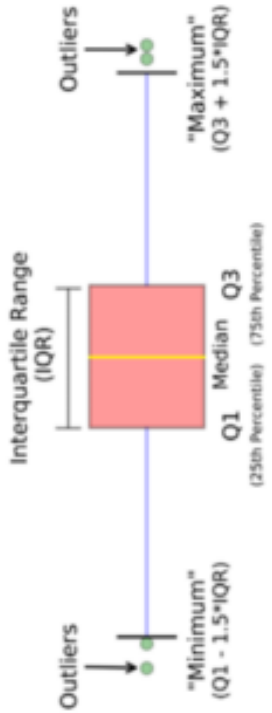
```
flights_passengers = flights.pivot('month', 'year', 'passengers')
sns.heatmap(flights_passengers, annot=True, fmt='d')
plt.show()
```



Seaborn - boxplot

- 측정값들의 최댓값, 최솟값, 중앙값, 사분편차를 사용하여 모양으로 분포 확인

```
tips = sns.load_dataset('tips')
print(tips)
```



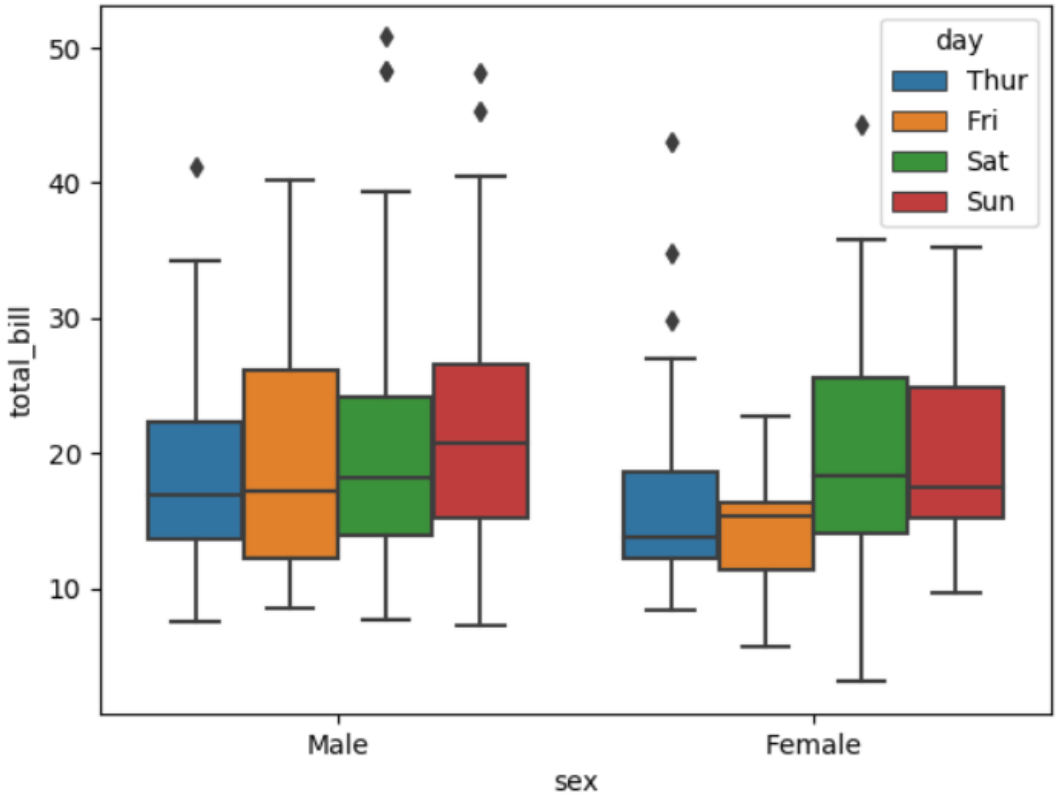
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
..	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

[244 rows x 7 columns]

Seaborn - boxplot

- 성별을 기준으로 돈을 많이 지출하는 요일은?

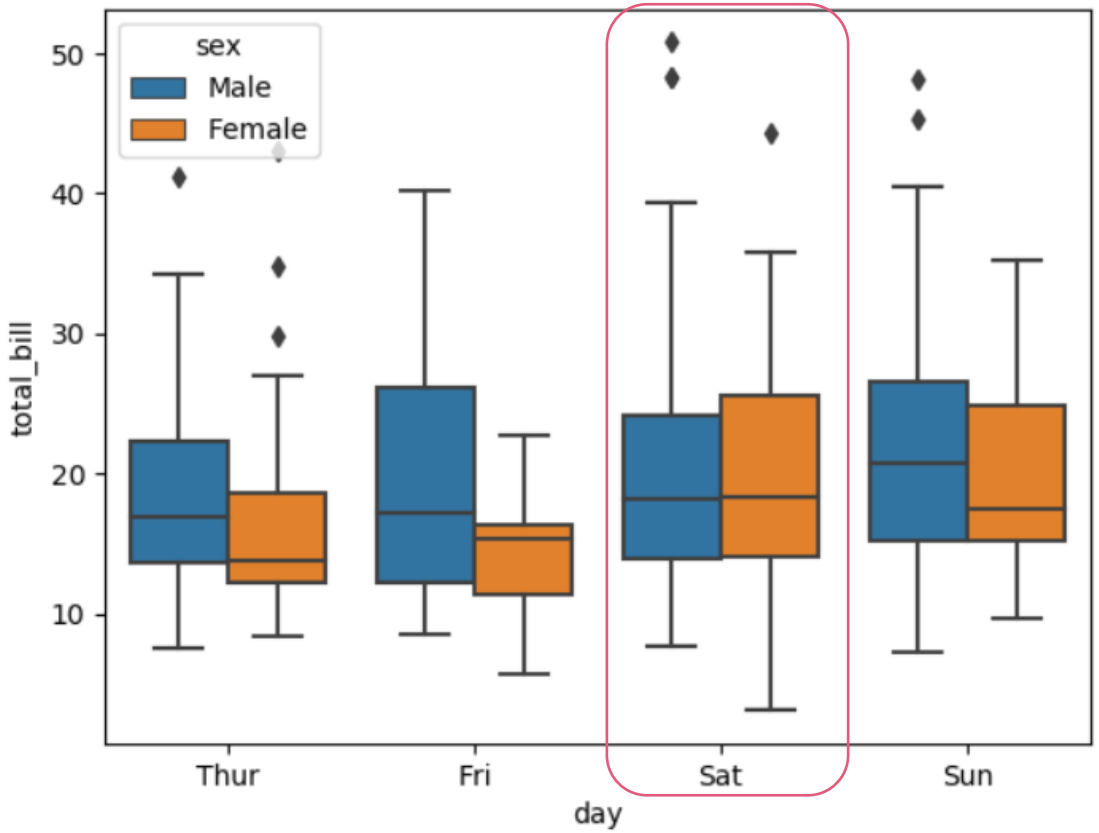
```
tips = sns.load_dataset('tips')
sns.boxplot(x='sex', y='total_bill', hue='day', data=tips)
plt.show()
```



Seaborn - boxplot

- 요일을 기준으로 돈을 많이 지출하는 사람(남성, 여성)은?

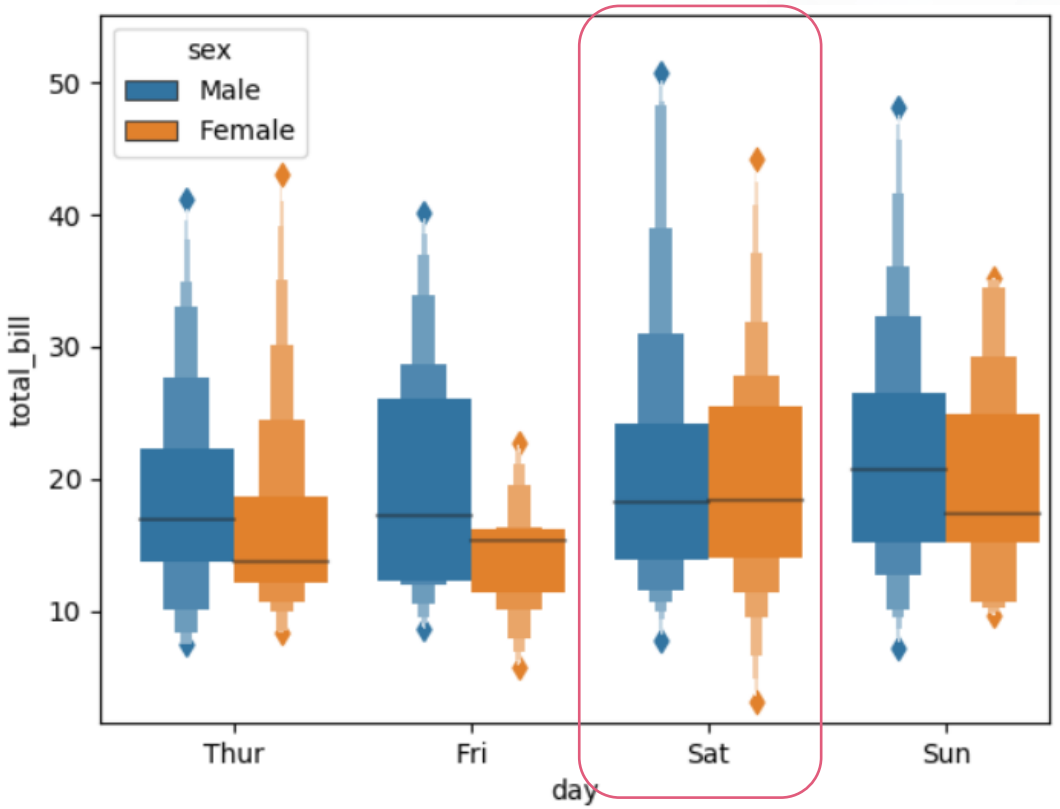
```
tips = sns.load_dataset('tips')
sns.boxplot(x='day', y='total_bill', hue='sex', data=tips)
plt.show()
```



Seaborn - boxenplot

- 요일을 기준으로 돈을 많이 지출하는 사람(남성, 여성)은?

```
tips = sns.load_dataset('tips')
sns.boxenplot(x='day', y='total_bill', hue='sex', data=tips)
plt.show()
```



Thank you