



Round 7

**PRESS
START**



《 Round 7 》

- 데이터 EDA 개요
- 데이터 시각화
- 데이터 상관 분석
- 이상치



New
Assignment



《 Round 7 》

- 데이터 EDA 개요 《
- 데이터 시각화
- 데이터 상관 분석
- 이상치



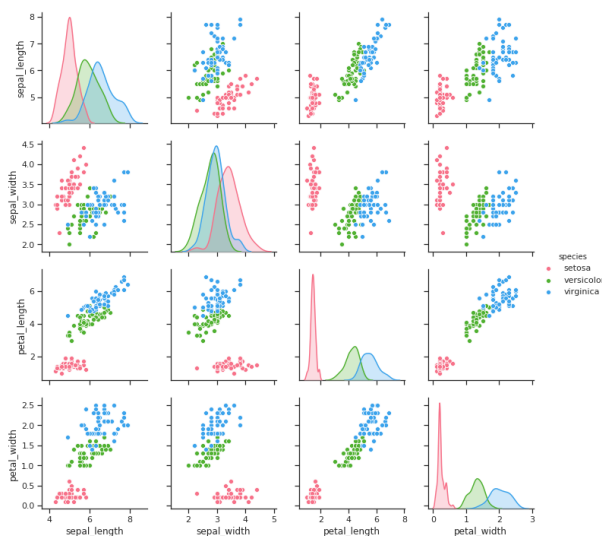
Let's
Go



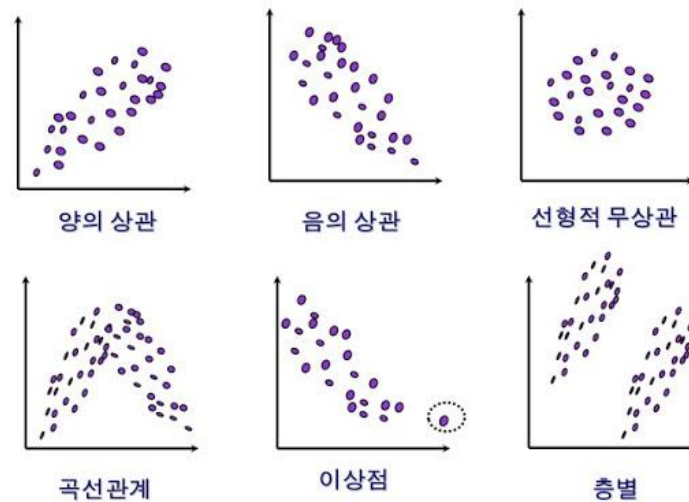
EDA(Exploratory Data Analysis)

- 주어진 데이터 셋을 통해 탐색적으로 충분한 정보를 찾는 분석 방법
- EDA를 통한 인사이트 도출과 가설설정은 분석의 큰 토대가 됨

HOW?



시각화



상관성 분석

EDA에서 시각화를 하는 이유?

I		II		III		IV	
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

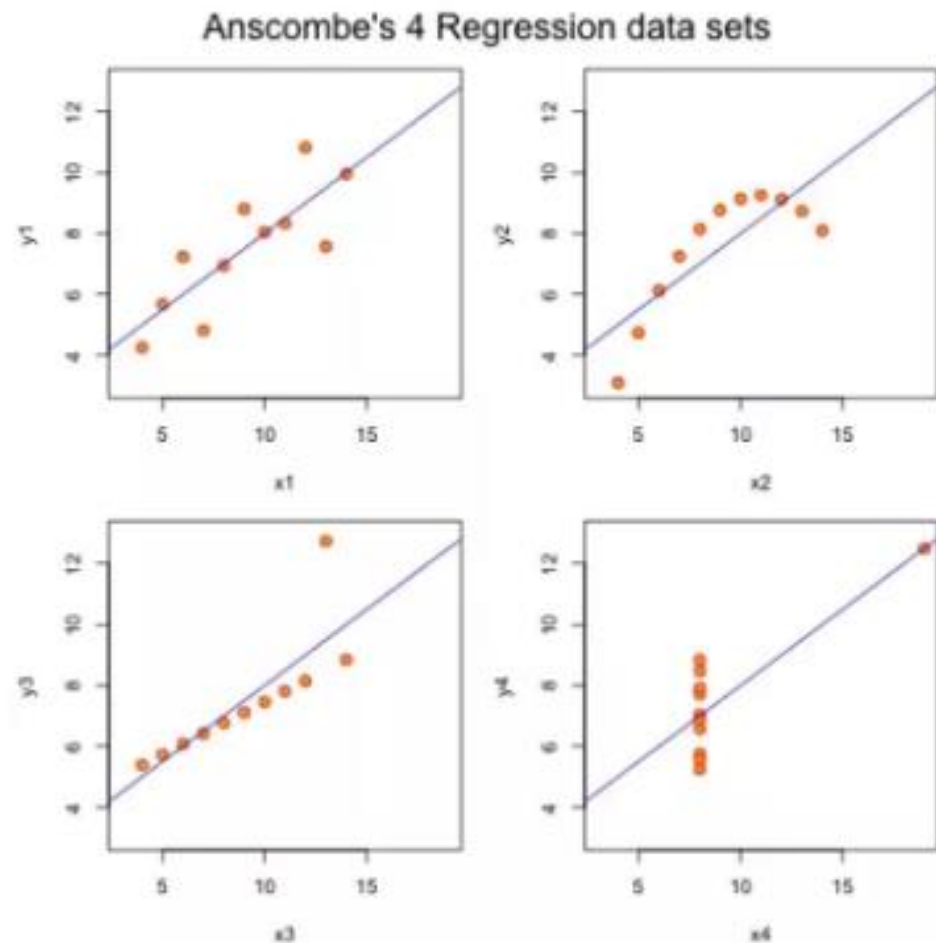
Mean of X	11.0	Correlation between X and Y	0.875
Variance of X	10.0	Linear regression	$y=3.0+0.5x$
Mean of Y	7.5		
Variance of Y	3.75		

각 데이터셋(I ~ IV)은 얼마나 비슷할까?

EDA에서 시각화를 하는 이유?

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

Mean of X	11.0	Correlation between X and Y	0.875
Variance of X	10.0	Linear regression	$y=3.0+0.5x$
Mean of Y	7.5		
Variance of Y	3.75		



요약 통계 정보만으로는 데이터를 정확하게 볼 수 없다.

Python 시각화 라이브러리

matplotlib
Version 3.1.0



seaborn

《 Round 7 》

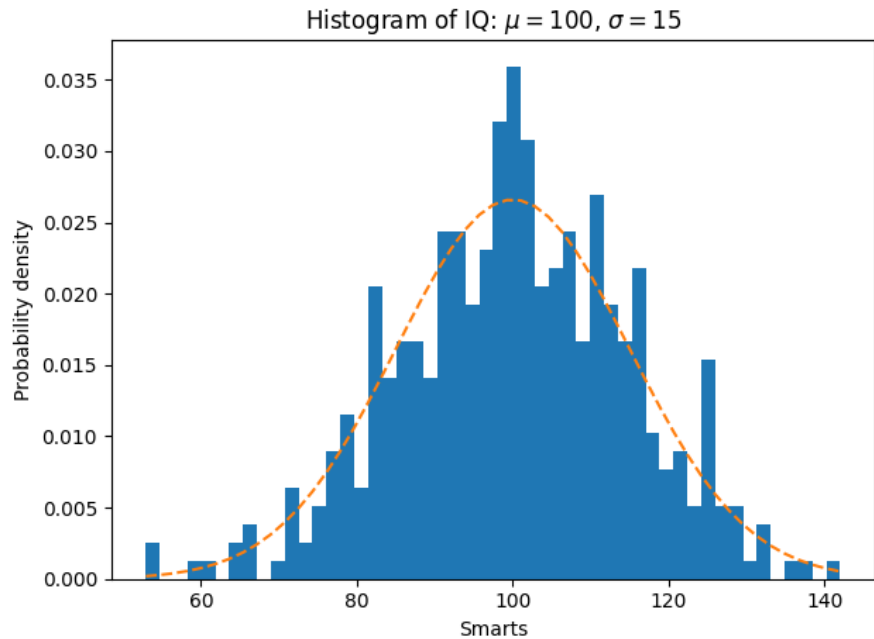
- 데이터 EDA 개요
- 데이터 시각화 《
- 데이터 상관 분석
- 이상치



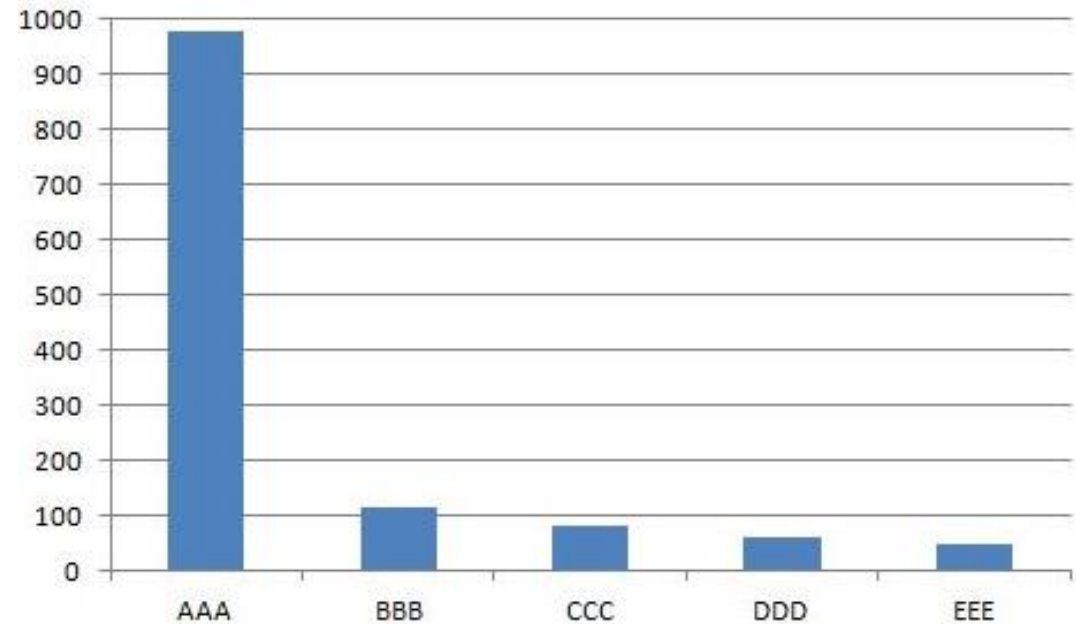
Let's
Go



그래프의 종류 - Bar

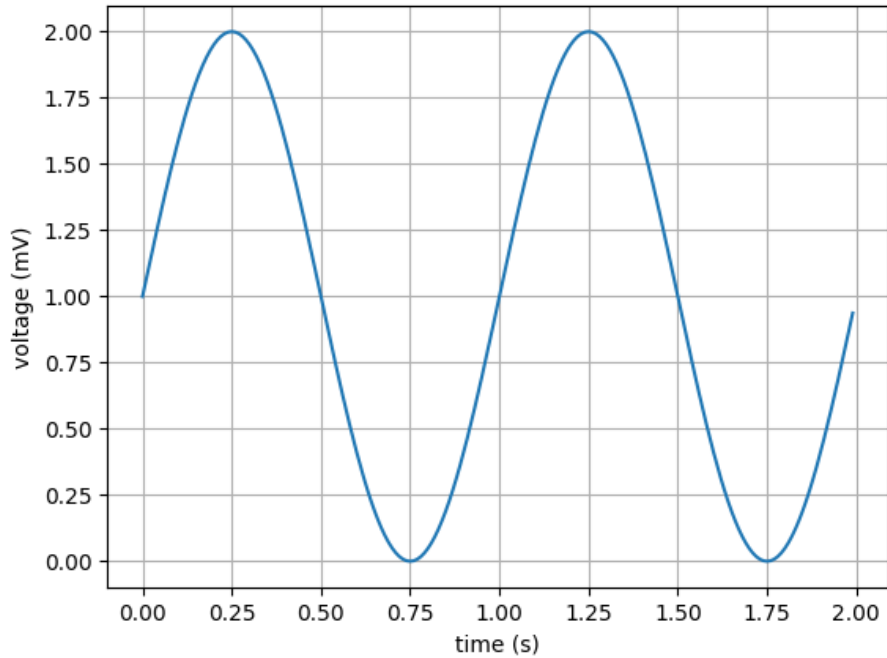


- 히스토그램(Histogram)
도수의 분포를 나타낸 막대모양의 그래프

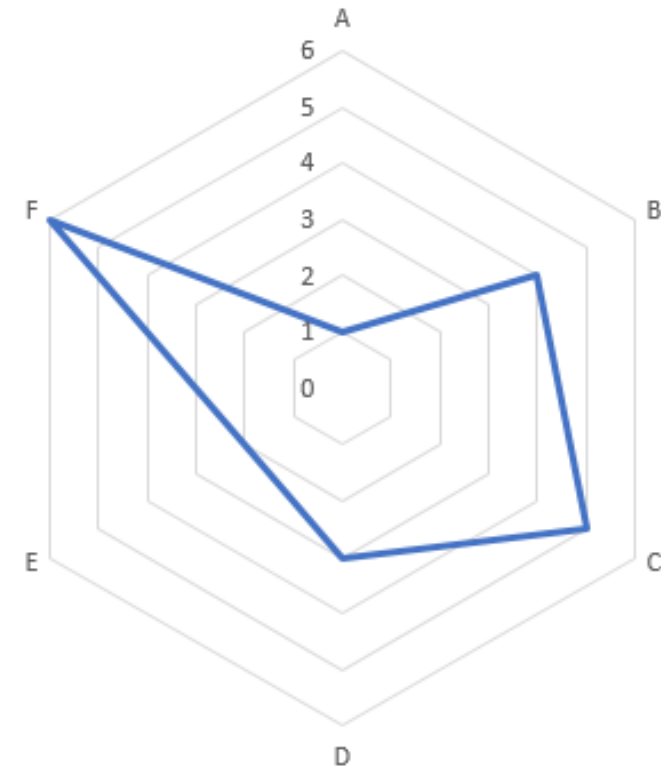


- 막대그래프(Bar plot)
이산적 자료의 양을 막대모양의 길이로 나타낸 그래프

그래프의 종류 - Line

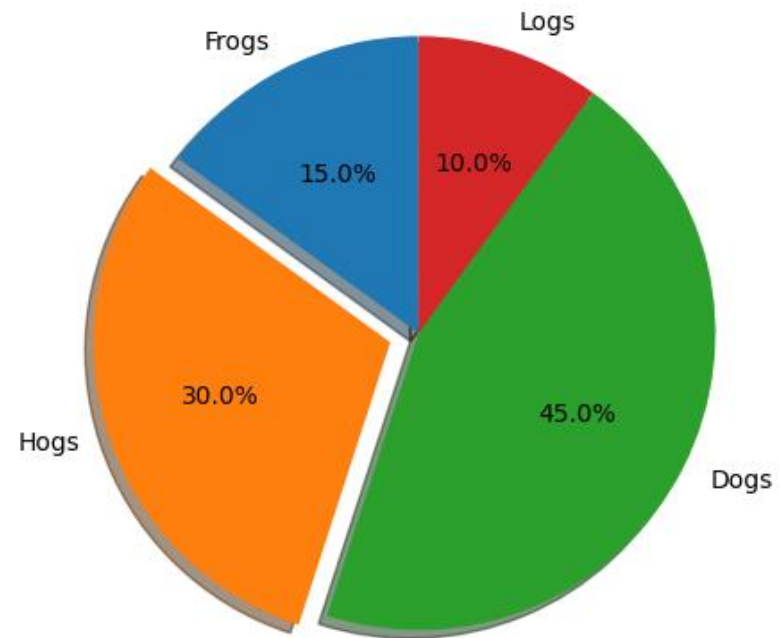


- 꺾은선 그래프(Line plot)
단위 흐름에 따른 자료의 양을 선으로 이은 그래프



- 방사형 그래프(Radar chart)
검사 결과 등에 사용되는 다각형 그래프

그래프의 종류 - Pie



- 원 그래프(Pie chart)
전체에 대한 각 항목의 비율을 원모양으로 나타낸 그래프

간단한 시각화 실습

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

# 데이터 읽어온 후 합계 데이터 프레임만 추출
elc_df = pd.read_excel('남북한발전전력량.xlsx')
elc_sum_df = elc_df.iloc[[0, 5], 2:]

# index에 이름 지정 및 int 형변환
elc_sum_df.index = ['south', 'north']
elc_sum_df.astype(int)
```

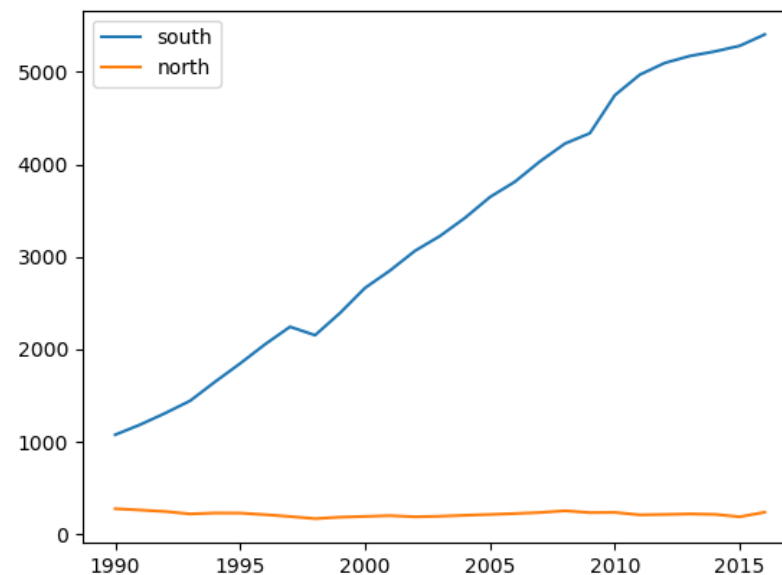
	1990	1991	1992	1993	2016
'south'	1077	1186	1310	1444	5404
'north'	277	266	247	221	239

간단한 시각화 실습

```
# 행 인덱스를 x축 데이터로 쓰기 때문에 년도의 x와 y의 국가를 바꿔줄
elc_sum_df = elc_sum_df.T
elc_sum_df.plot()

# 찍은 plot 출력
plt.show()
```

'년도'	'south'	'north'
1990	277	1077
1991	266	1086
1992	247	1310
1993	221	1444
1993	221	1444

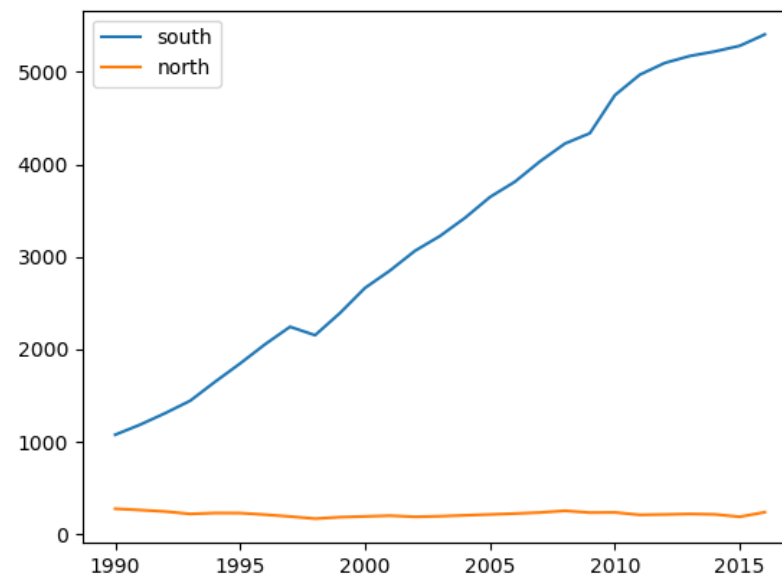


간단한 시각화 실습

```
# 행 인덱스를 x축 데이터로 쓰기 때문에 년도의 x와 y의 국가를 바꿔줄
elc_sum_df = elc_sum_df.T
elc_sum_df.plot()

# 찍은 plot 출력
plt.show()
```

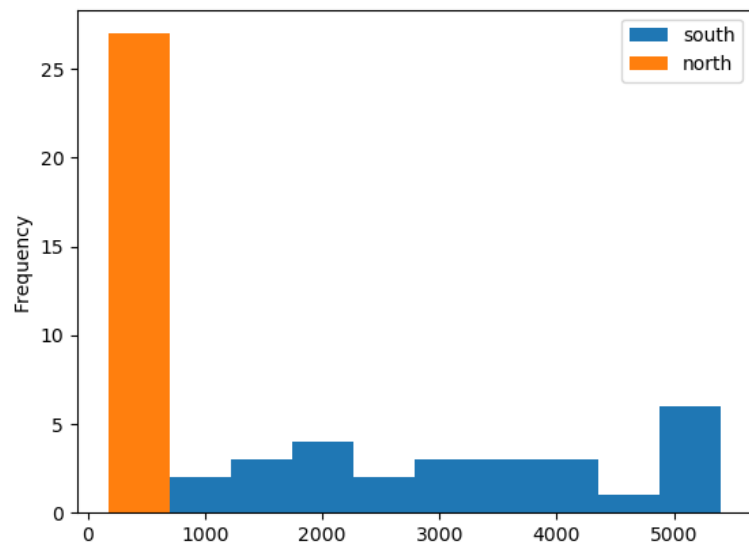
'년도'	'south'	'north'
1990	277	1077
1991	266	1086
1992	247	1310
1993	221	1444
1993	221	1444



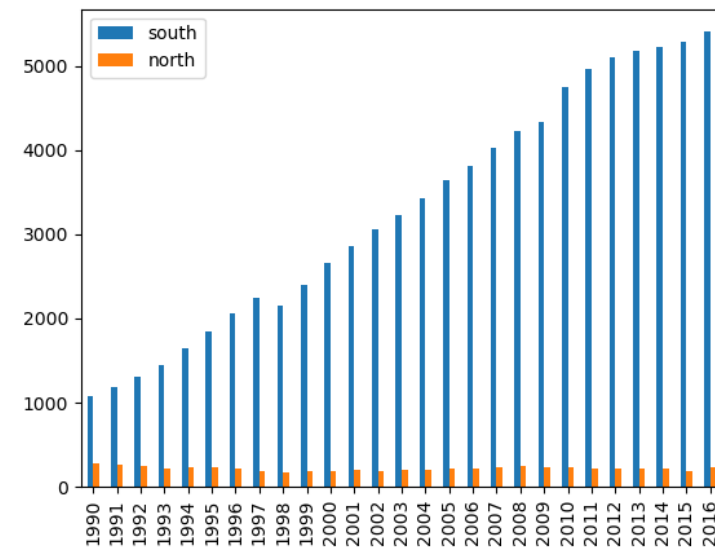
Q. 다음 데이터셋.T로 그리기 부적합한 그래프는?

	1990	1991	1992	1993	2016
'south'	1077	1186	1310	1444	5404
'north'	277	266	247	221	239

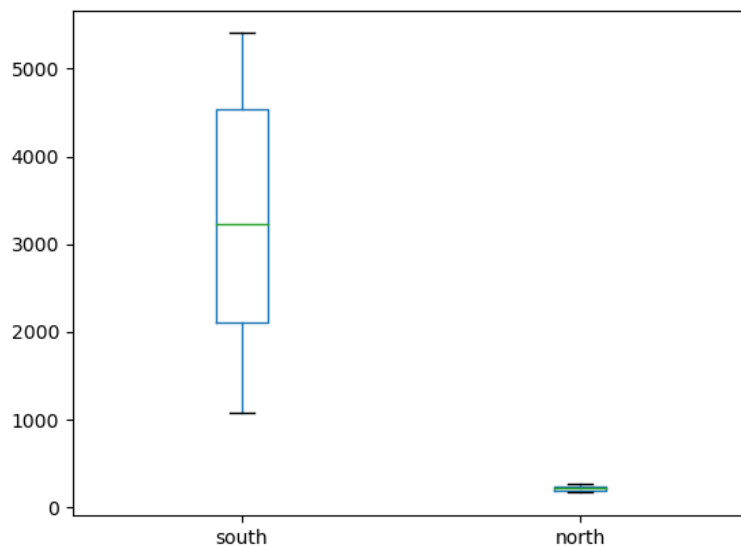
- 1) `DataFrame.plot(kind = "hist")` # 히스토그램
- 2) `DataFrame.plot(kind = "bar")` # 막대 그래프
- 3) `DataFrame.plot(kind = "box")` # 박스플롯
- 4) `DataFrame.plot(kind = "scatter", x='south, y='north')` # 산점도
- 5) `DataFrame.plot(kind = "pie")` # 원 그래프



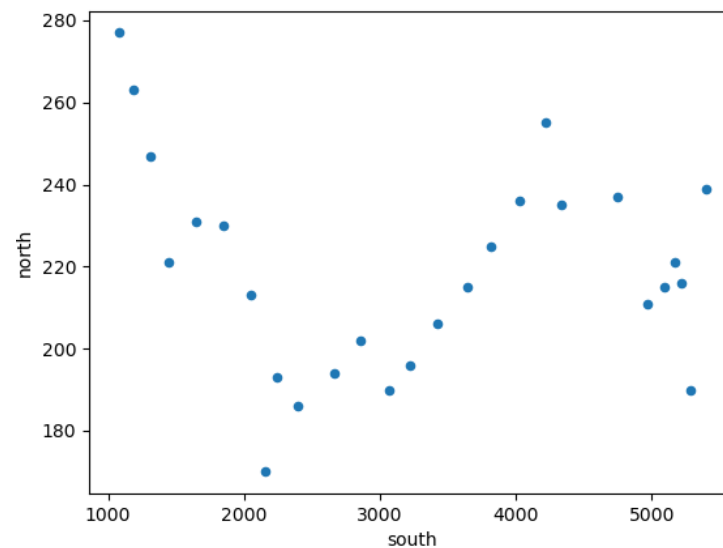
DataFrame.plot(kind = "hist")



DataFrame.plot(kind = "bar")



DataFrame.plot(kind = "box")



DataFrame.plot(kind = "scatter", x = "south", y = "north")



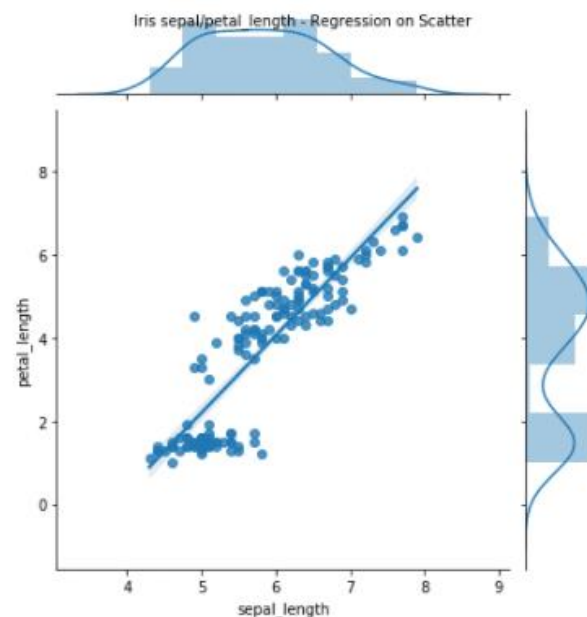
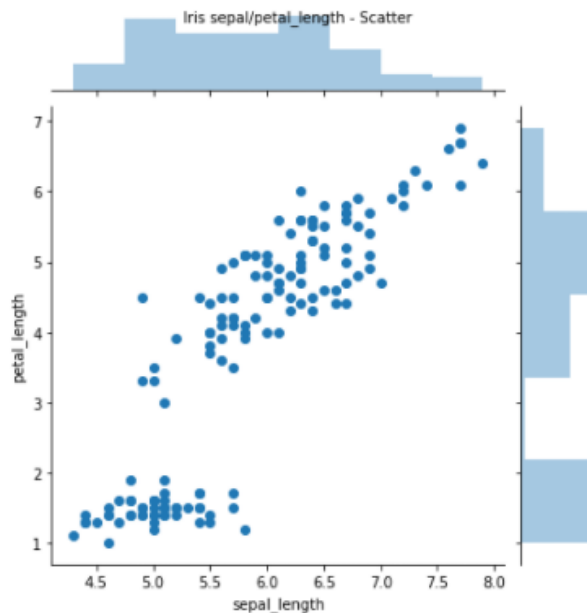
- matplotlib을 기반으로 하는 고급 시각화 도구
 - 다양한 실습용 데이터 내장
 - pip install seaborn
 - import seaborn as sns

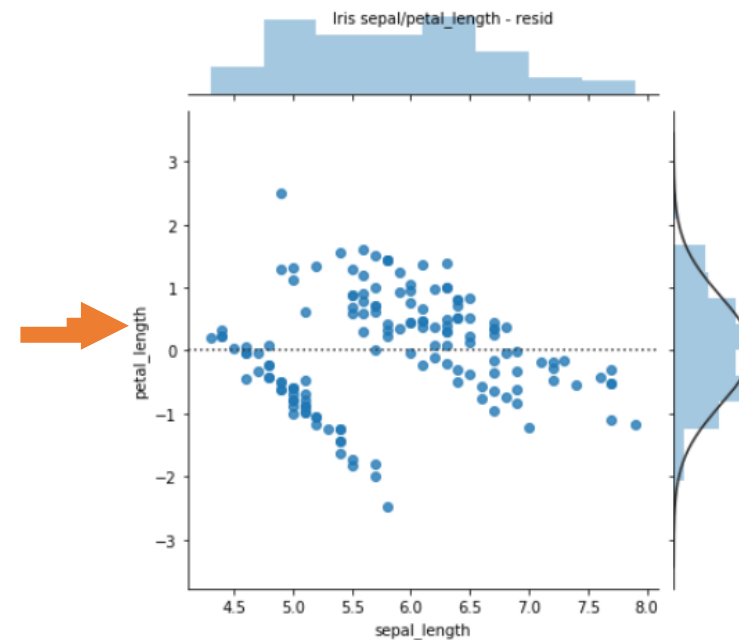
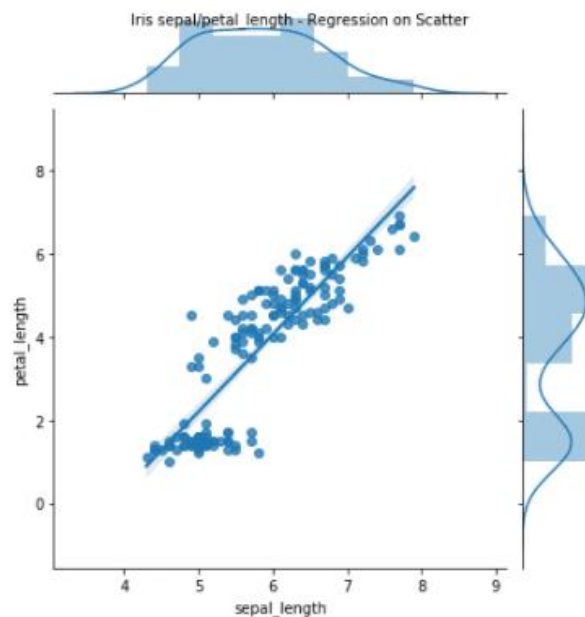
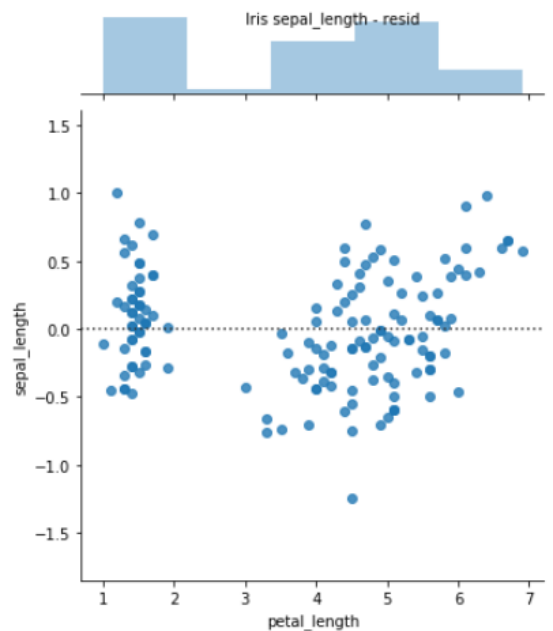
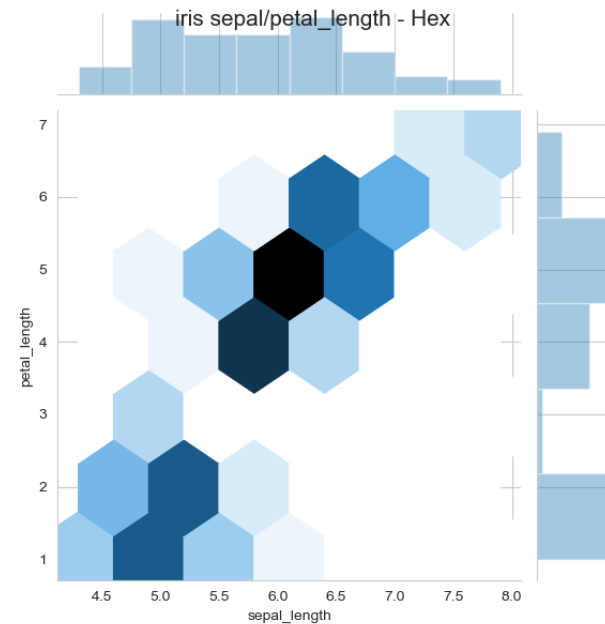
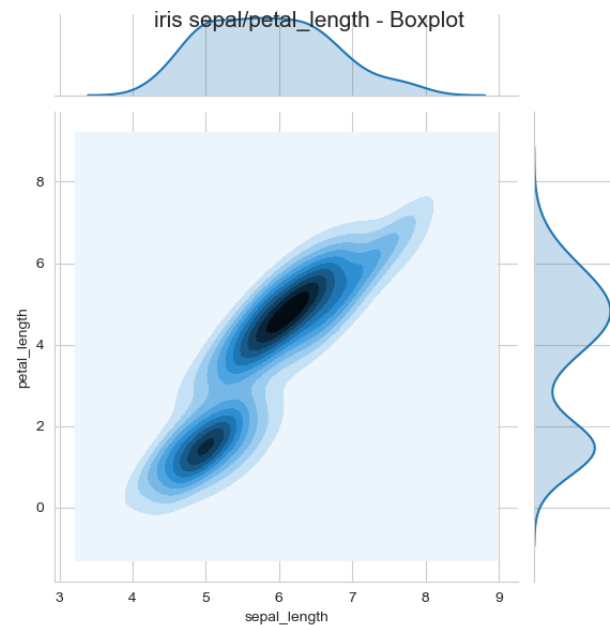
Seaborn 다중 시각화 실습

```
jop1 = sns.jointplot(x='sepal_length', y='petal_length', kind = 'scatter', data=iris_df)
jop2 = sns.jointplot(x='sepal_length', y='petal_length', kind = 'reg', data=iris_df)
jop3 = sns.jointplot(x='sepal_length', y='petal_length', kind = 'kde', data=iris_df)
jop4 = sns.jointplot(x='sepal_length', y='petal_length', kind = 'hex', data=iris_df)
jop5 = sns.jointplot(x='sepal_length', y='petal_length', kind = 'resid', data=iris_df)
jop6 = sns.jointplot(x='petal_length', y='sepal_length', kind = 'resid', data=iris_df)

jop1.fig.suptitle('Iris sepal/petal_length - Scatter', size = 10)
jop2.fig.suptitle('Iris sepal/petal_length - Regression on Scatter', size = 10)
jop3.fig.suptitle('Iris sepal/petal_length - K dense graph', size = 10)
jop4.fig.suptitle('Iris sepal/petal_length - Hex graph', size = 10)
jop5.fig.suptitle('Iris petal_length - residual', size = 10)
jop6.fig.suptitle('Iris sepal_length - residual', size = 10)

plt.show()
```



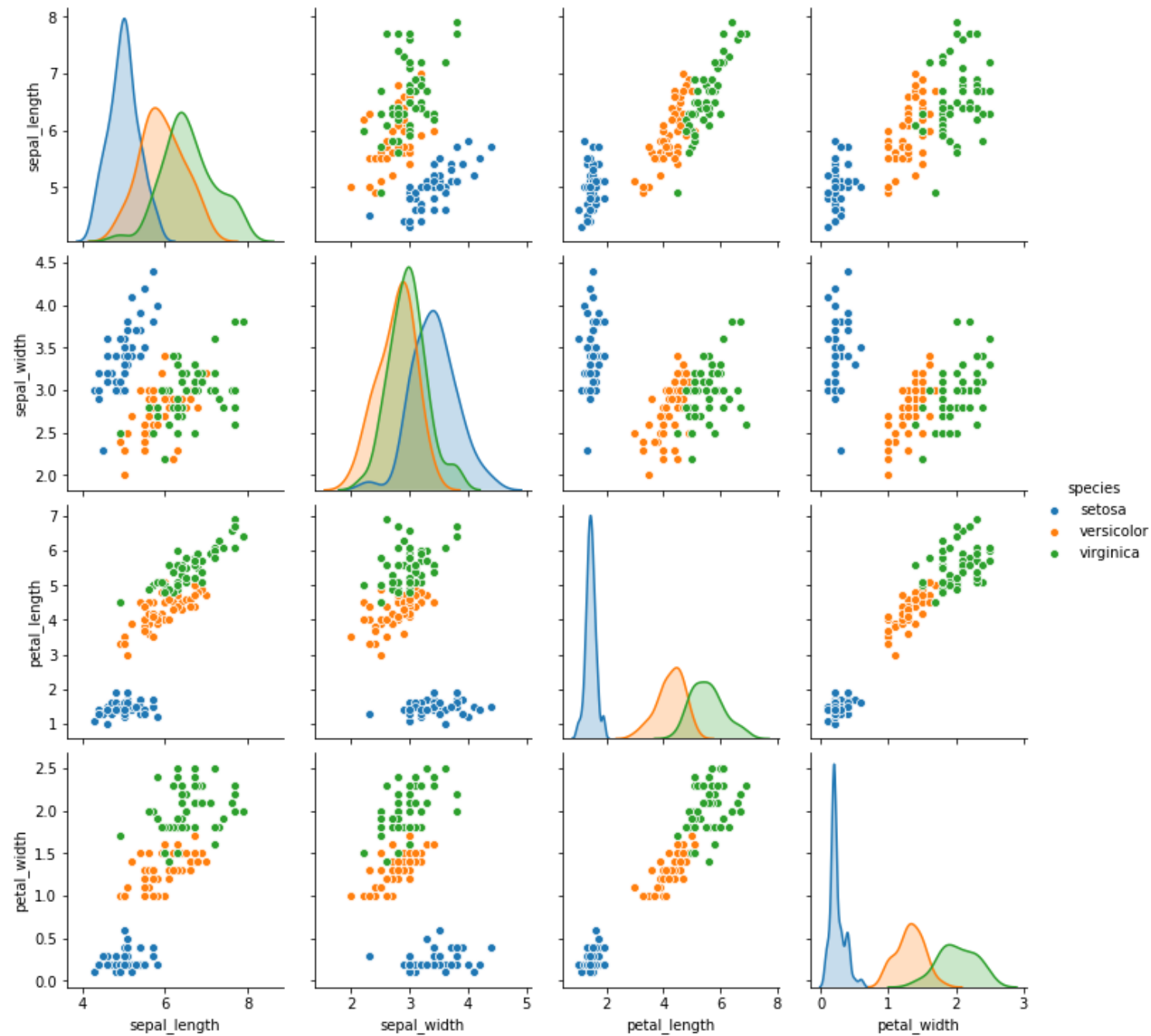


Seaborn 이변수 데이터 분포(pairplot) 실습



“모든 변수 쌍들에 대해서 plot을 쉽게 찍어 볼 수는 없을까?”

```
ppl = sns.pairplot(iris_df, hue = "species")  
plt.show()
```



《 Round 7 》

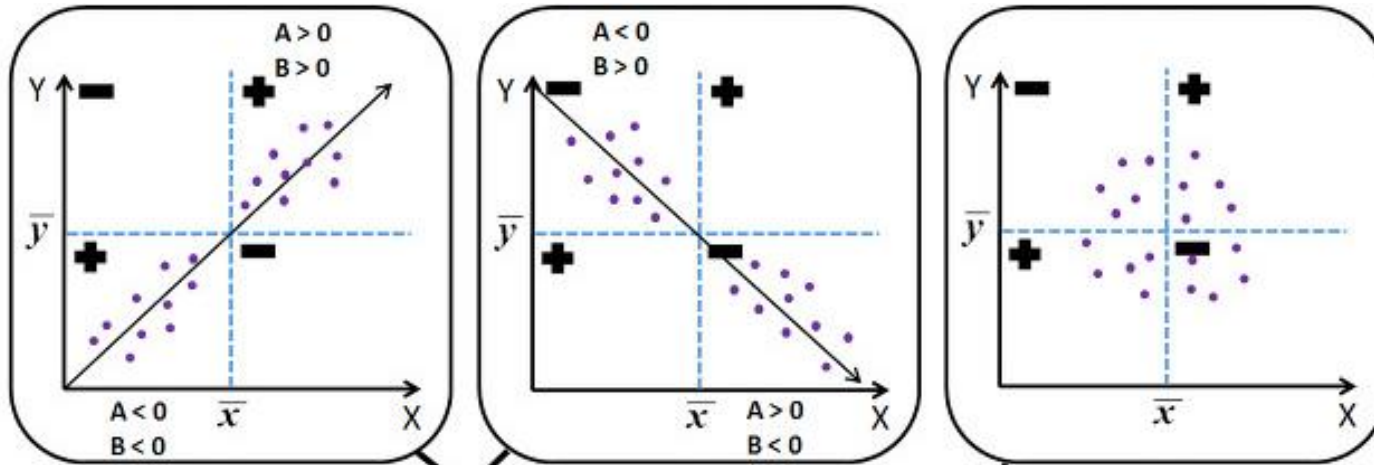
- 데이터 EDA 개요
- 데이터 시각화
- 데이터 상관 분석 《
- 이상치



Let's
Go



상관(성) 분석(Correlation Analysis)



- 양적인(Quantitative) 두 변수 간의 관계가 얼마나 유의한지 분석하는 것
- 상관 분석을 통해서 변수들의 관계를 규명하고 가설 설정에 도움을 줄 수 있음
- 이렇게 구한 상관분석의 양적 결과를 피어슨 상관계수(Pearson's correlation coefficient)라고 함

상관(성) 분석(Correlation Analysis)

```
iris_df.corr()
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

- petal_length와 petal_width는 1에 가깝기 때문에 강한 양의 선형관계를 지님
- sepal_length와 sepal_width의 상관계수는 0에 가깝기 때문에 선형관계가 거의 없음
 - 그러나 피어슨 상관계수가 낮다고 하여 관계 자체가 없다고는 할 수는 없음.

《 Round 7 》

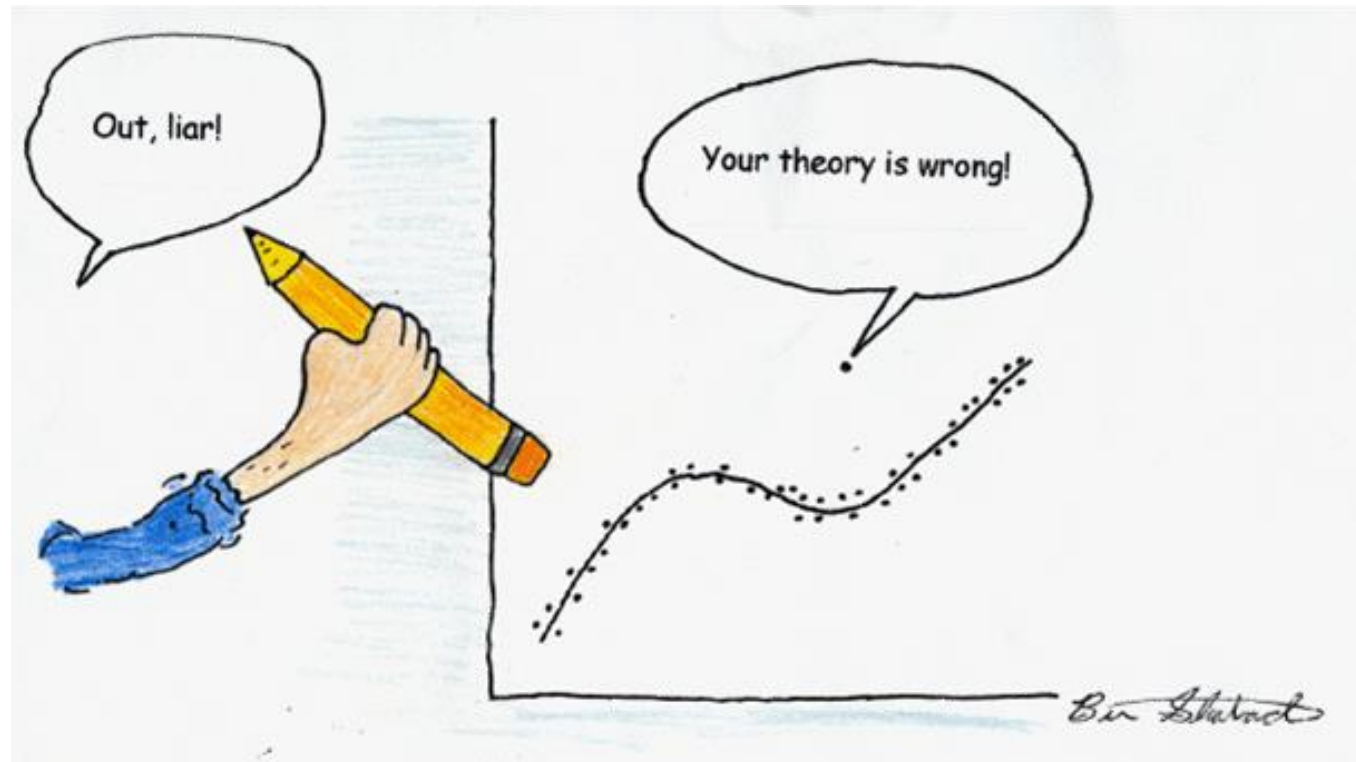
- 데이터 EDA 개요
- 데이터 시각화
- 데이터 상관 분석
- 이상치 《



Let's
Go

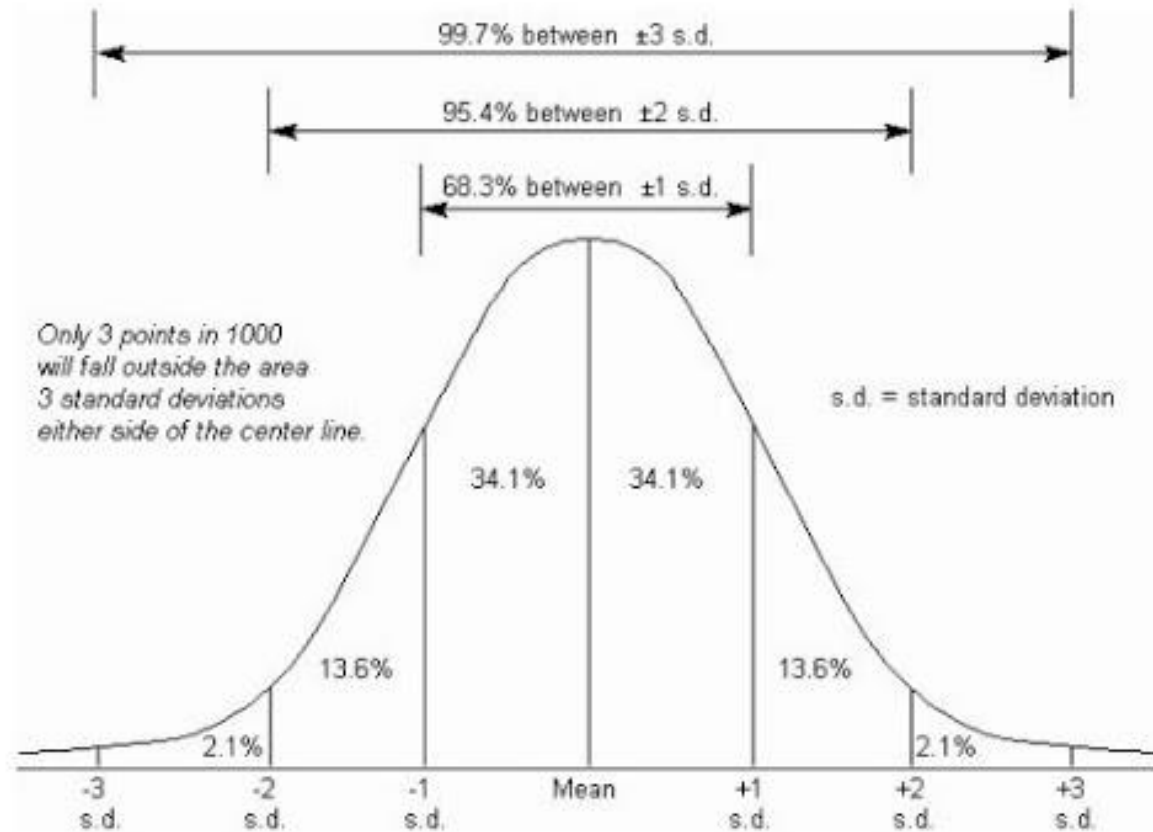


이상치(Outlier)



**전체적인 데이터/샘플 범위에서 동떨어진 관측값으로,
모델을 크게 왜곡시킬 가능성이 있음.**

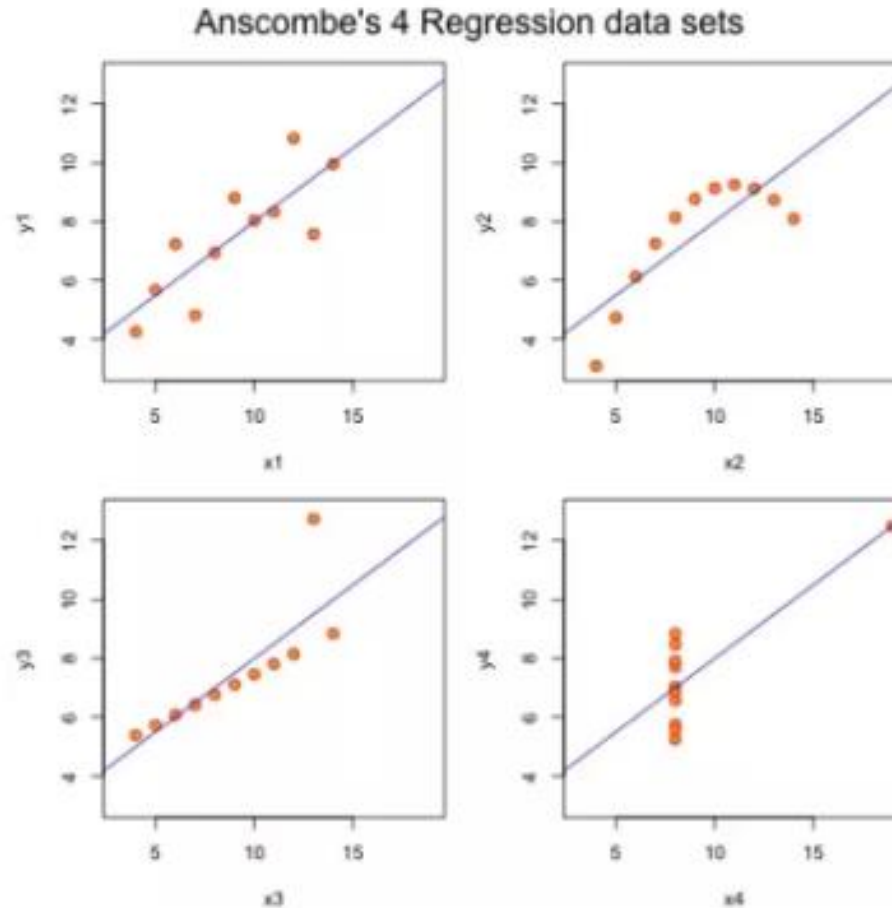
이상치를 결정하는 방법?



일반적으로 6σ , 즉 ± 3 표준편차에 해당하는 값을 이상치라고 봄
목적과 자료에 따라 3σ , 4σ , 5σ 로도 설정

IQR방식, 앤드류스 그림, 마하라노비스 거리로도 이상치 결정가능

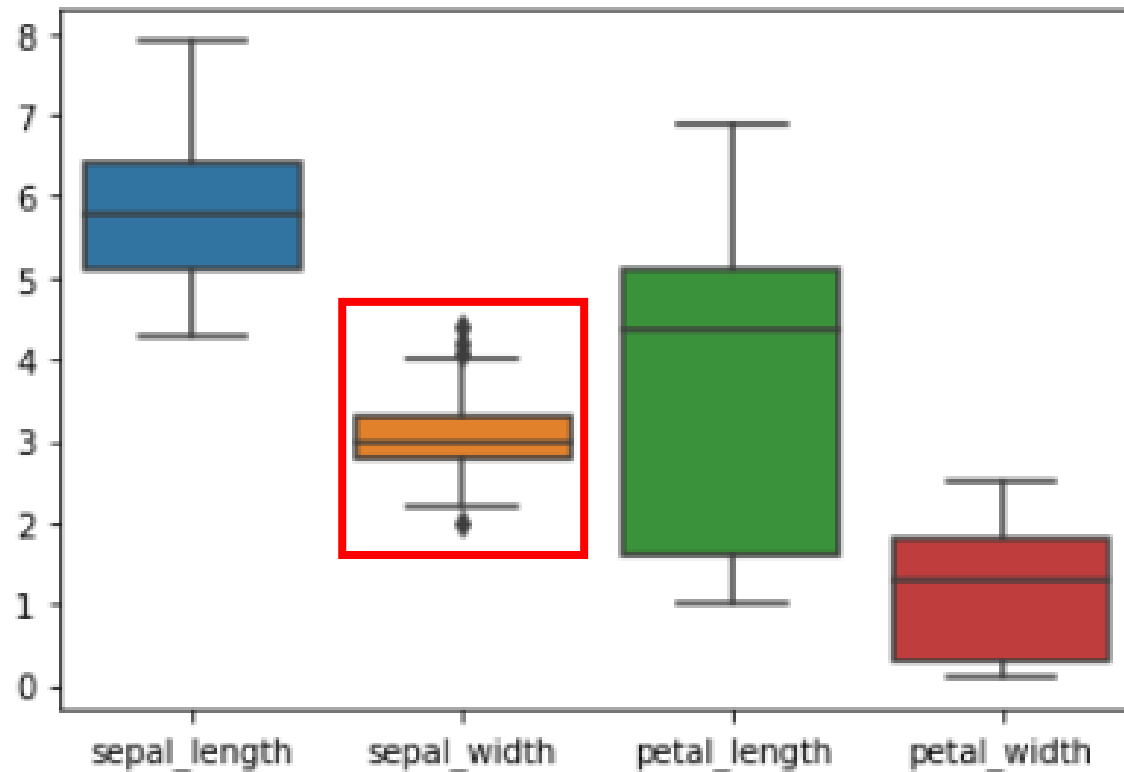
이상치를 결정하는 방법?



가장 직관적인 방법은 시각화
시각화를 통해서 이상치를 주관적으로 확인할 수 있음.

Boxplot을 통한 이상치 탐색

```
irisBp = sns.boxplot(data=iris_df)  
  
plt.show()
```



이상값 처리(Outlier treatment)

이상 값을 찾았다면...

a. 단순 삭제

- Human error에 의한 경우 해당 관측치를 삭제하면 됨.
- ex) 단순 오타, 주관식 설문 등의 비현실적 응답, 처리과정에서의 오류 등

이상값 처리(Outlier treatment)

이상 값을 찾았다면...

b. 대체

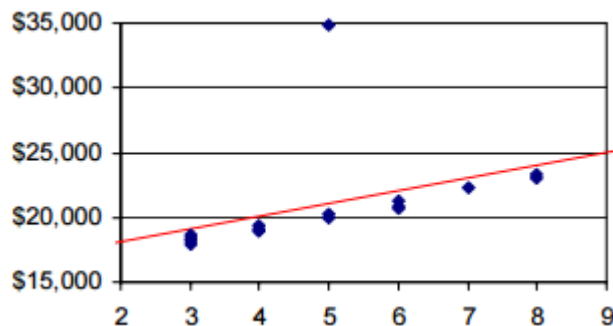
- 평균, 중간값, 중앙값 등으로 대체
- 결측값과 유사하게 다른 변수들을 사용해서 예측모델을 만들고,
 - 이상값을 예측한 후 해당 값으로 대체
- 이상값이 자연발생한 경우 삭제/대체를 통해 모델을 만들면
현상/예측을 잘 설명할 수 없을 수도 있음.

이상값 처리(Outlier treatment)

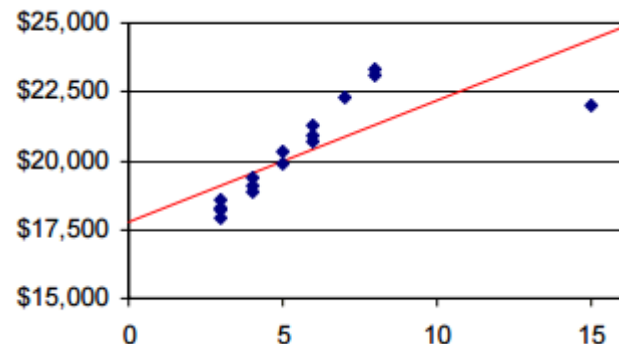
이상 값을 찾았다면...

c. 이상치가 자연발생 했을 경우의 방법

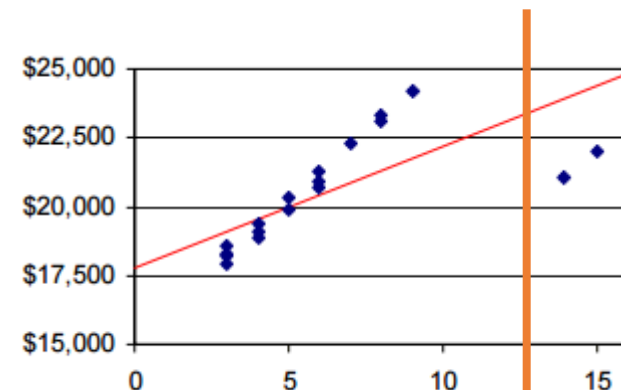
ex) 년차별 소득 수준



전문직종 종사 여부를 변수화
(종속변수가 outlier일 경우)



년차의 범위를 10년까지로 리샘플링
(종속변수, 독립변수가 outlier)



케이스 분리 해석
(특정 경향의 outlier가 여러 개일 경우)

NEXT STAGE

