학습 내용

3부. 데이터 분석 라이브러리 활용

11장. N차원 배열 다루기

12장. 데이터프레임과 시리즈

13장. 데이터 시각화

- 1. 시각화 개요
- 2. Matplotlib
- 3. Seaborn

14장. 웹 데이터 수집

1절. 시각화 개요

```
# 시작전 설정
import matplotlib.pyplot as plt
%matplotlib inline
# 그래프 해상도 높임
%config InlineBackend.figure_format='retina'
#한글설정
plt.rc('font', family='Malgun Gothic')
#plt.rc('font', family='AppleGothic') #mac
plt.rc('axes', unicode_minus=False)
# 경고 메세지 안보이게
import warnings
warnings.filterwarnings(action='ignore')
```

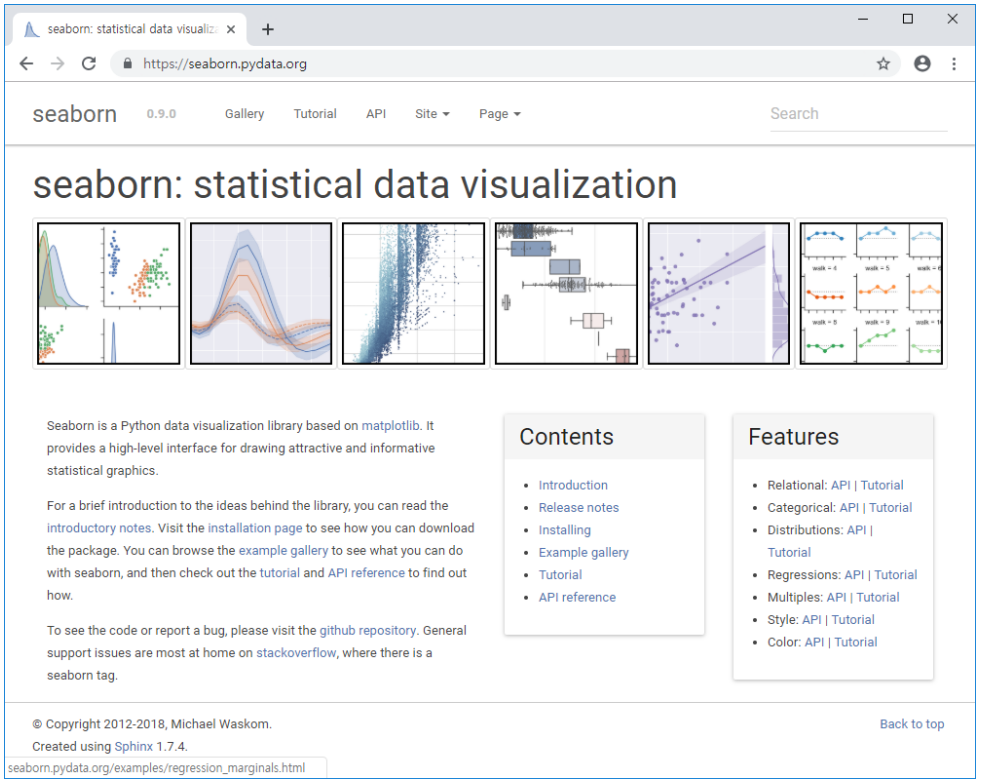
시각화 라이브러리

1절. 시각화 개요

- matplotlib, seaborn, plotnine, folium, poly.ly, pyecharts

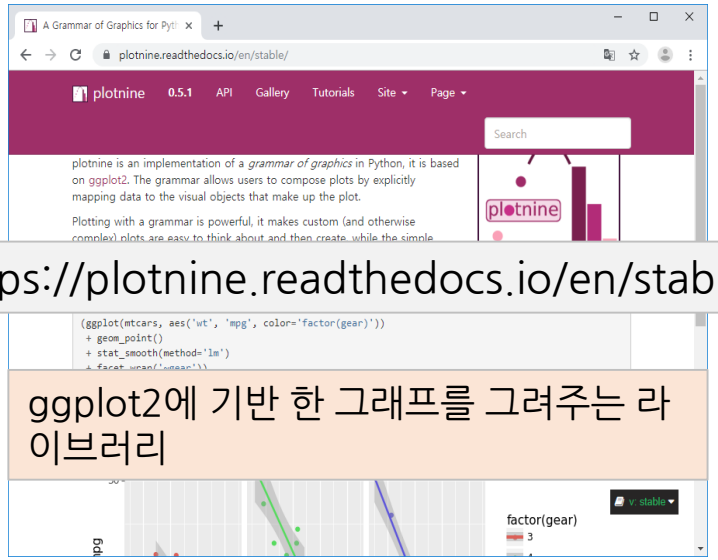
http://matplotlib.org

https://seaborn.pydata.org



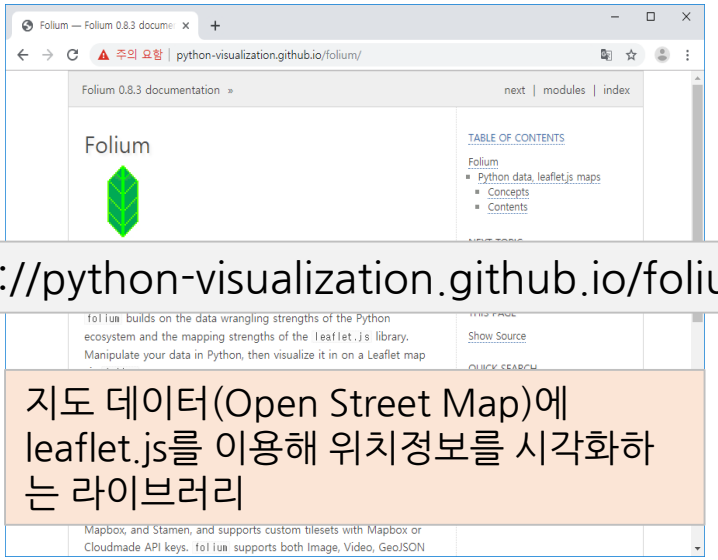
시각화 라이브러리

1절. 시각화 개요



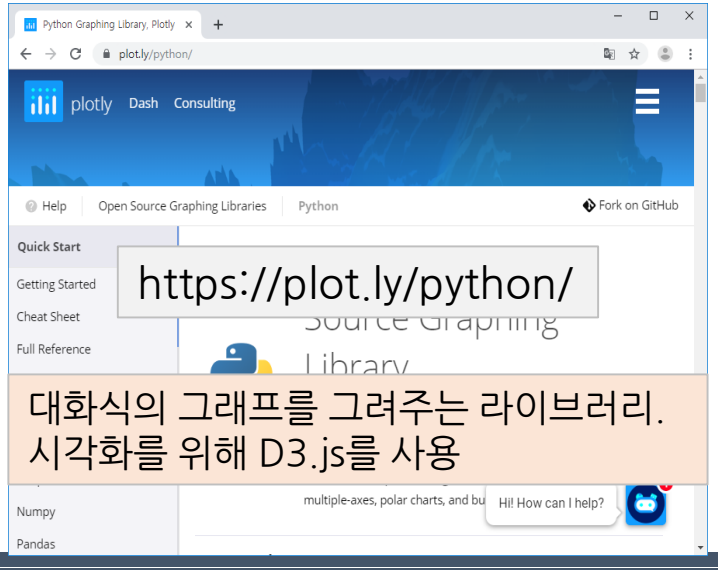
https://plotnine.readthedocs.io/en/stable/

ggplot2에 기반 한 그래프를 그려주는 라이브러리



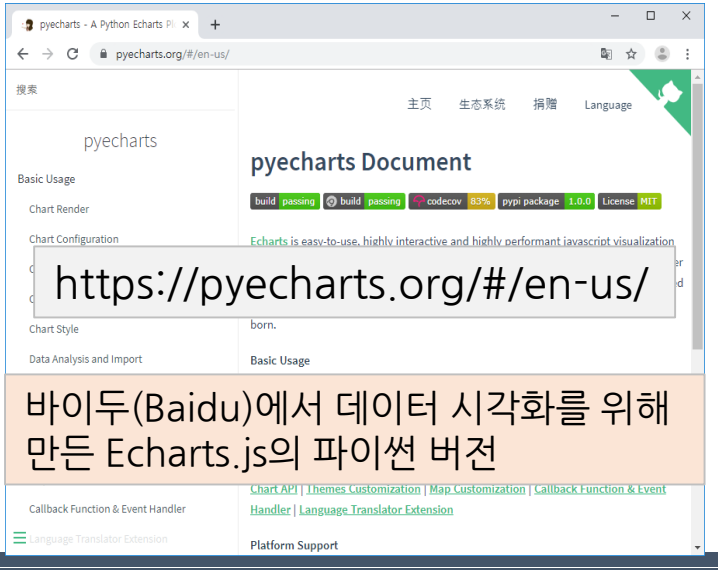
http://python-visualization.github.io/folium/

지도 데이터(Open Street Map)에 leaflet.js를 이용해 위치정보를 시각화하는 라이브러리



https://plot.ly/python/

대화식의 그래프를 그려주는 라이브러리. 시각화를 위해 D3.js를 사용



https://pyecharts.org/#/en-us/

바이두(Baidu)에서 데이터 시각화를 위해 만든 Echarts.js의 파이썬 버전

2절. Matplotlib

https://matplotlib.org/stable/api/pyplot_summary.html

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html

2.1. 패키지 임포트 및 기본 설정

2절. Matplotlib

- Matplotlib로 그래프를 그리기 위한 라이브러리 import
- `%matplotlib inline`
  - notebook을 실행한 브라우저에서 바로 그림을 볼 수 있게 해 줌
- `%config InlineBackend.figure_format = 'retina'`
  - 그래프를 더 높은 해상도로 그려줌

```
1  import matplotlib
2  import matplotlib.pyplot as plt
```

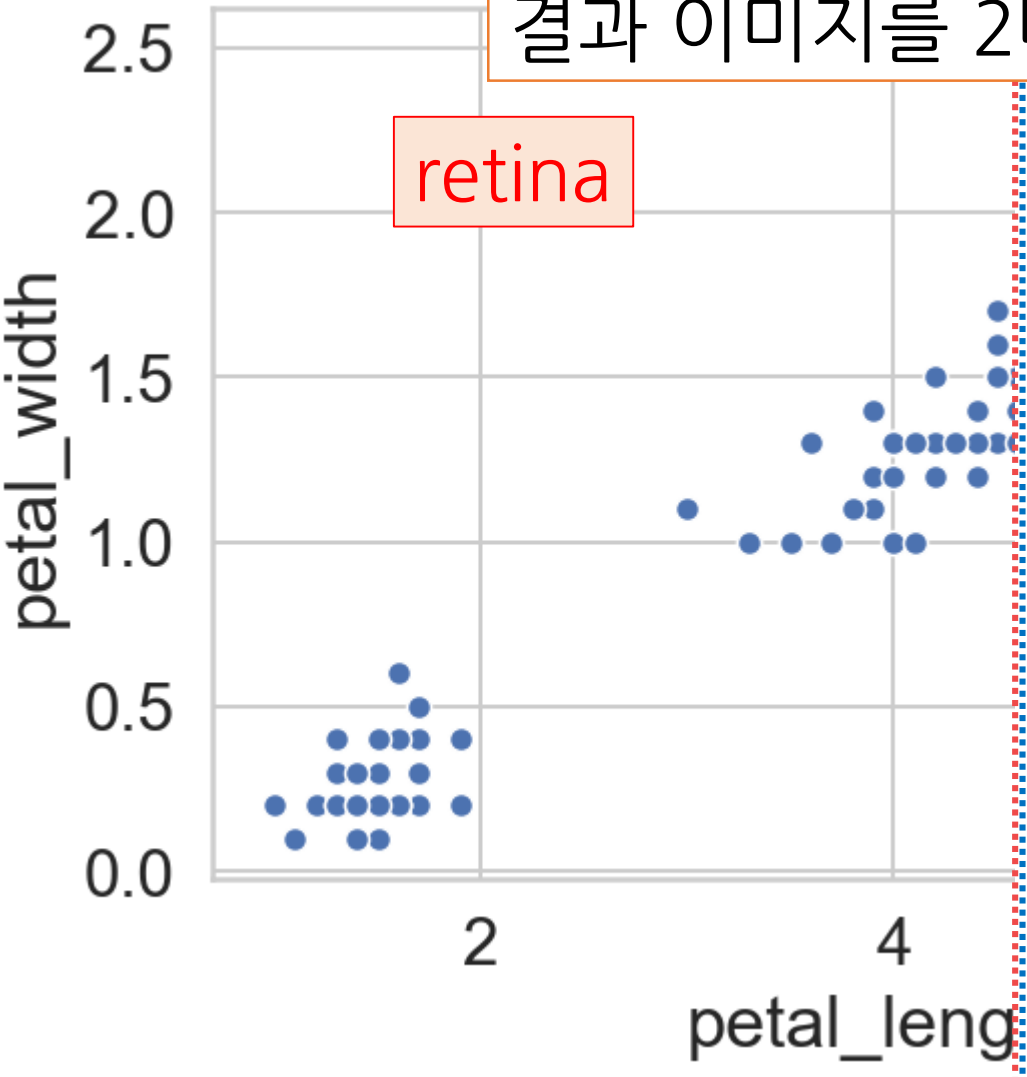
```
1  %matplotlib inline
2  %config InlineBackend.figure_format = 'retina'
3  print("Matplotlib 버전:", matplotlib.__version__)
```

```
Matplotlib 버전: 3.0.2
```

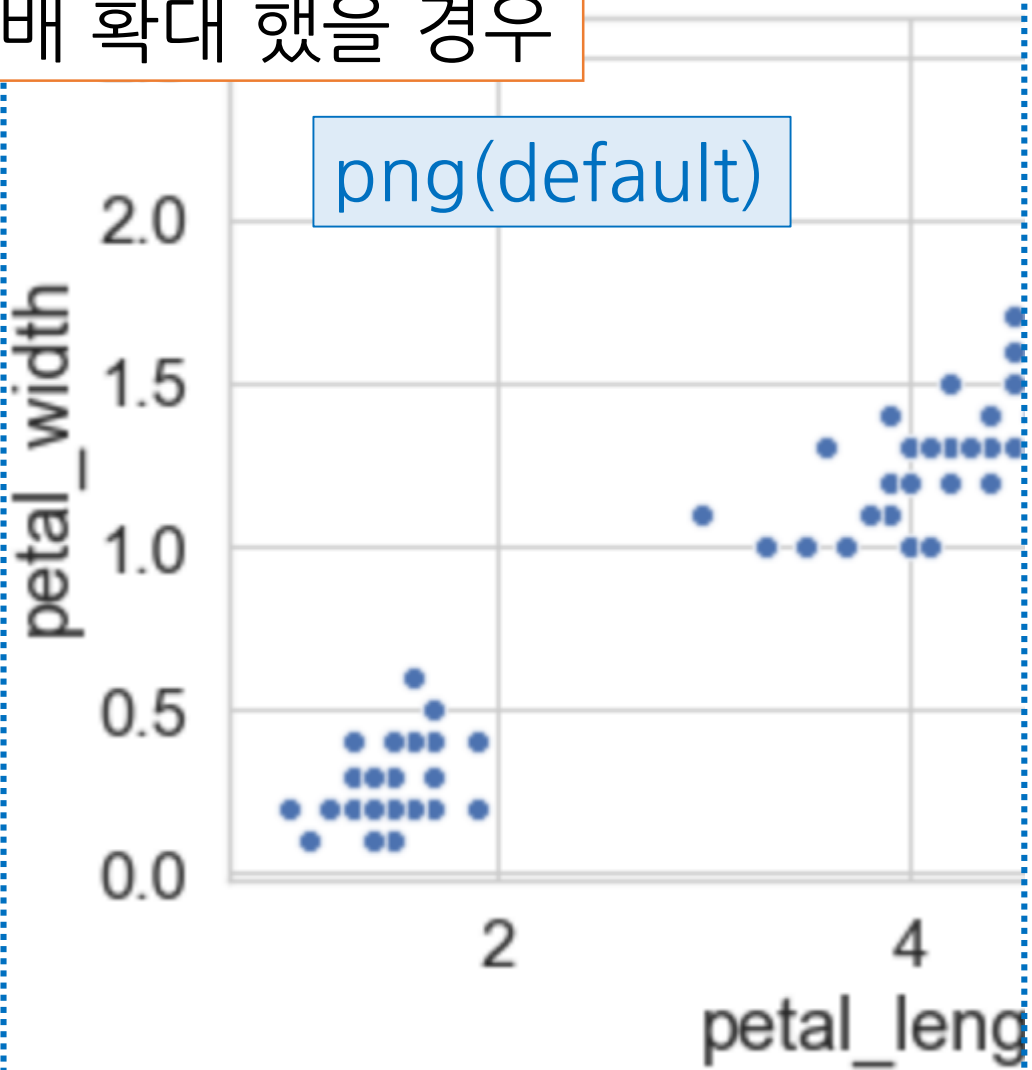
%config InlineBackend.figure_format = 'retina'

결과 이미지를 2배 확대 했을 경우

retina



png(default)



2.2. 그래프 객체

2절. Matplotlib

- Figure 객체 : Matplotlib에서 그래프를 그리기 위한 객체
- 도화지(Figure)를 plt.subplots()으로 분할해 각 부분에 그래프를 그리는 방식으로 시각화 함

```
matplotlib.pyplot.figure(num=None, figsize=None, dpi=None,
        facecolor=None, edgecolor=None, frameon=True,
        FigureClass=<class 'matplotlib.figure.Figure'>,
        clear=False, **kwargs)
```

- 구문에서…
  - num : 정수 또는 문자열, 선택 사항, 기본값 : None.이 값이 제공되지 않으면 새 그림이 만들어지고 그림 번호가 증가. figure 객체는 숫자 속성에 이 숫자를 저장. num이 제공되고 이 ID를 가진 그림이 이미 존재하면 활성화하고 참조를 반환함. 이 숫자가 없으면 새로 만들어 반환함. num이 문자열이면 윈도우 제목이 설정.
  - figsize : (float, float), 선택, 기본값 : None.너비와 높이를 인치 단위로 지정. 제공되지 않으면 기본값은 plt.rcParams["figure.figsize"]=[6.4, 4.8].

```
1  plt.figure()
```

plot()

- plot() 함수는 주어진 x, y 값을 선(lines)과 점(markers)으로 표시해 줌

```
matplotlib.pyplot.plot([x], y, [fmt], data=None, **kwargs)
```

- 구문에서...
  - *fmt* : 색, 점, 라인의스타일을 문자열로 지정. 예를 들면 'ro-'는 빨간색 동그란 점을 갖는 실선.
    - 점의 모양은 o(원), s(네모), v(역삼각형), ^(삼각형), x(x표시) 등
    - 선의 스타일은 '-'(실선), '--'(대시선), '-.'(대시닷선), ':'(점선), ''(선없음) 등

pandas.DataFrame.plot()

- 판다스의 데이터프레임 객체를 이용해서 그래프를 그릴 수 있음
- 데이터프레임의 plot() 함수는 데이터프레임의 데이터를 쉽게 시각화

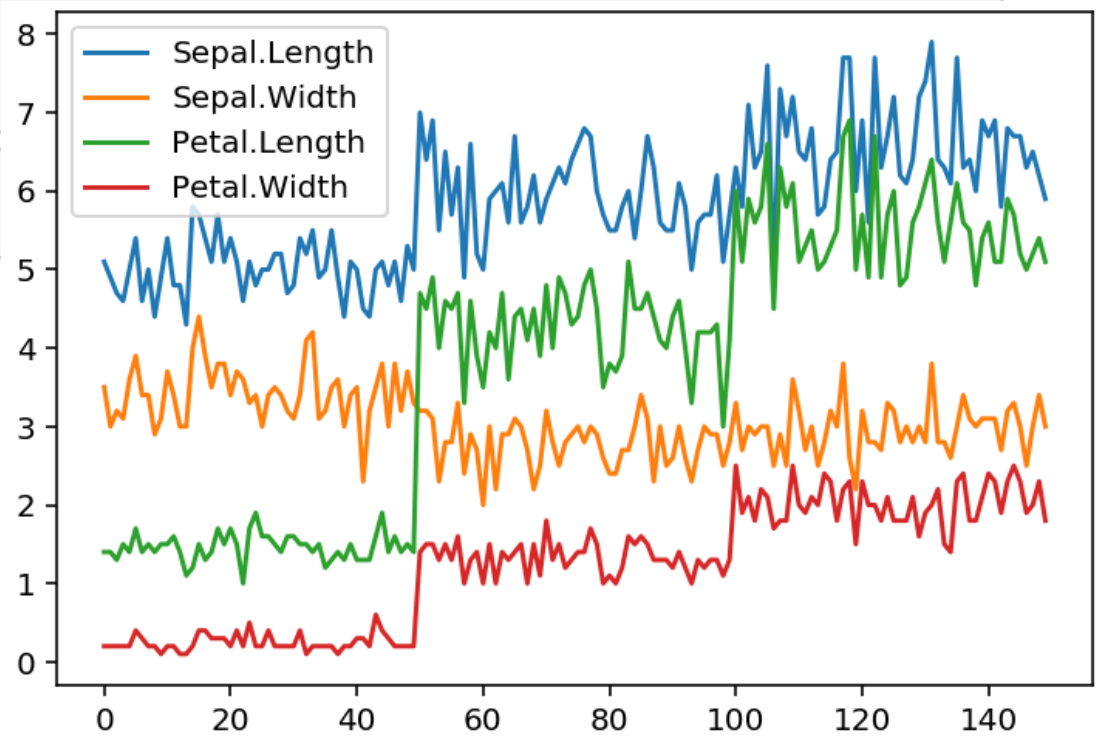
```
DataFrame.plot(x=None, y=None, kind='line', ax=None, subplots=False,
        sharex=None, sharey=False, layout=None, figsize=None,
        use_index=True, title=None, grid=None, legend=True, style=None,
        logx=False, logy=False, loglog=False, xticks=None, yticks=None,
        xlim=None, ylim=None, rot=None, fontsize=None, colormap=None,
        table=False, yerr=None, xerr=None, secondary_y=False,
        sort_columns=False, **kwds)
```

- 구문에서...
  - kind : 그래프의 종류('line' , 'bar', 'barh', 'hist', 'box', 'kde', 'density', 'area', 'pie', 'scatter', 'hexbin')

3) pandas.DataFrame.plot()


```
1  import numpy as np
2  import pandas as pd
3
4  import statsmodels.api as sm
5  iris = sm.datasets.get_rdataset("iris", package="datasets")
6  iris_df = iris.data
7  iris_df.head()
```

```
1  iris_df.plot()
```



https://stackoverflow.com/questions/30490740/move-legend-outside-figure-in-seaborn-tsplot : 범례사용

3) pandas.DataFrame.plot()

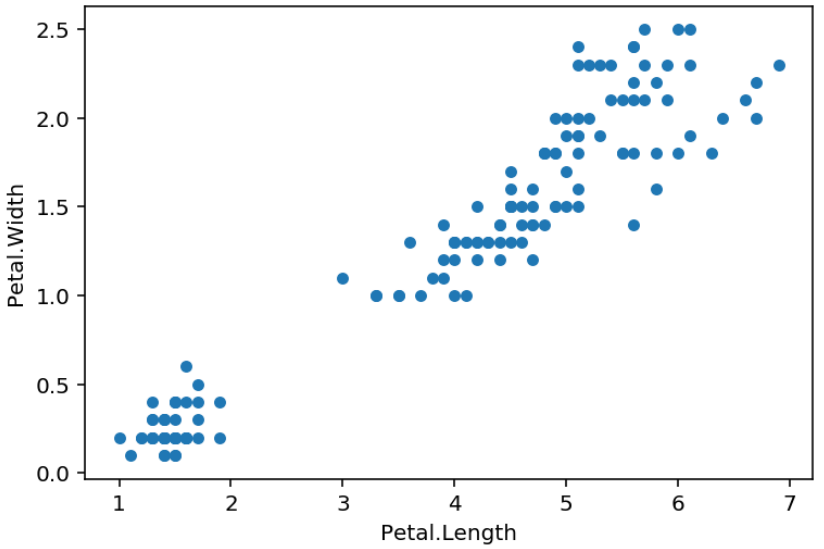

```
1 iris_df.corr()
```

	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

상관계수를 출력하고 상관관계가 높은 두 변수를 이용해서 산점도 그래프를 그려봄

```
1 iris_df.plot(x="Petal.Length",y='Petal.Width',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2ca92ce1710>
```



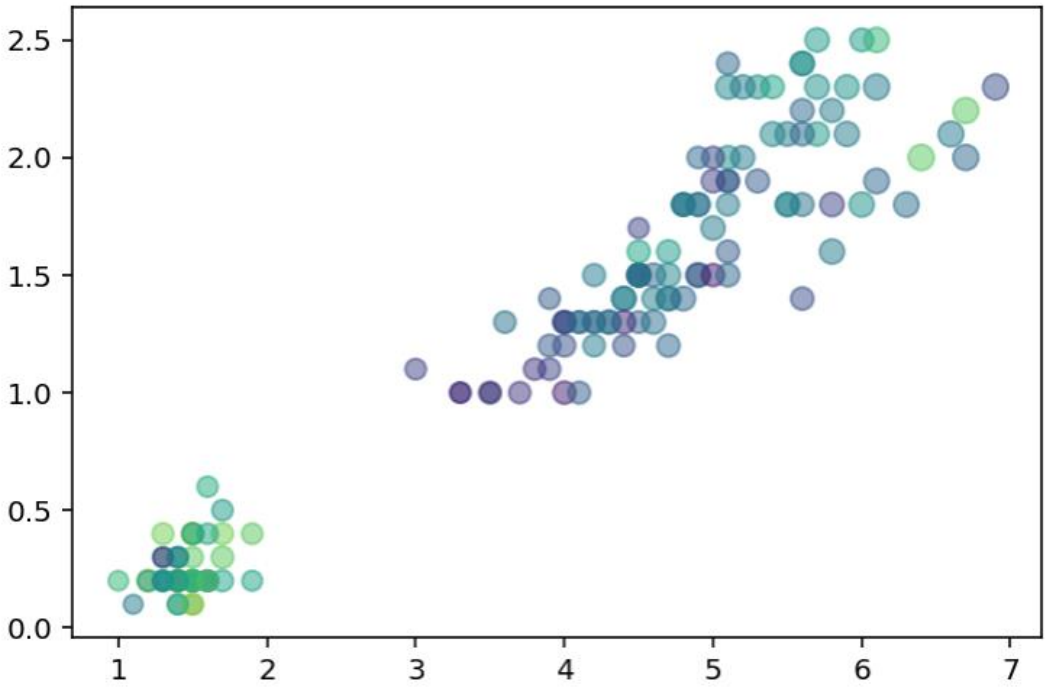
https://ko.wikipedia.org/wiki/상자_수염_그림

https://matplotlib.org/stable/tutorials/colors/colormaps.html

4) scatter()

- scatter() 함수는 산점도 그래프를 그려줌

```
1  plt.scatter(x=iris.petal_length, y=iris.petal_width,
2              s=iris.sepal_length*10, c=iris.sepal_width,
3              alpha=0.5)
4  plt.show()
```



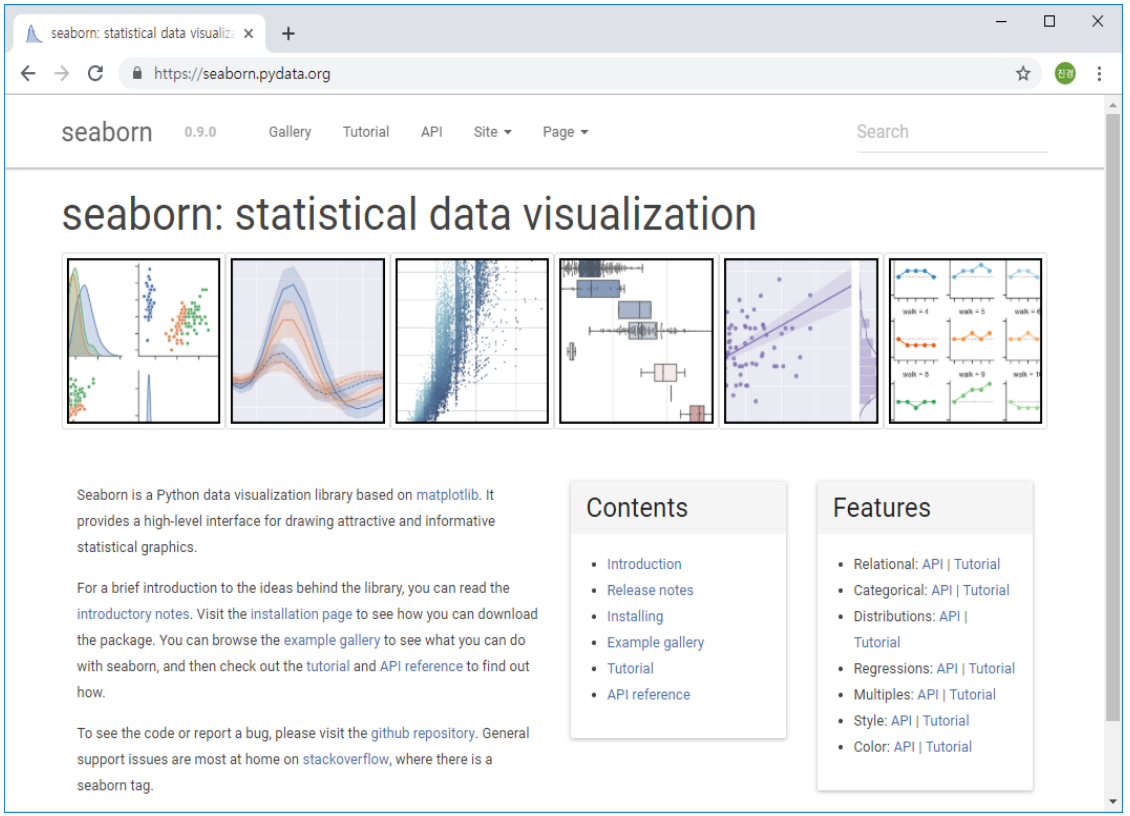
3절. Seaborn

- matplotlib을 기반으로 만든 고수준 그래픽 라이브러리
  - 공식사이트 : https://seaborn.pydata.org
  - seaborn API : https://seaborn.pydata.org/api.html
- Seaborn으로 그래프를 그리기 위해서 다음 단계를 따릅니다.
  1) 데이터 준비
  2) 미적속성 설정
  3) 함수를 이용하여 그래프 그리기
  4) 그래프 출력, 저장

3절. Seaborn

3절. Seaborn

- matplotlib를 기반으로 만들어진 고수준 그래프 라이브러리
  - 공식 사이트 : https://seaborn.pydata.org/
  - 그래프 API : https://seaborn.pydata.org/api.html



3.1. 데이터 준비하기

3절. Seaborn

- Seaborn의 내장 데이터셋
- iris = sns.load_dataset("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

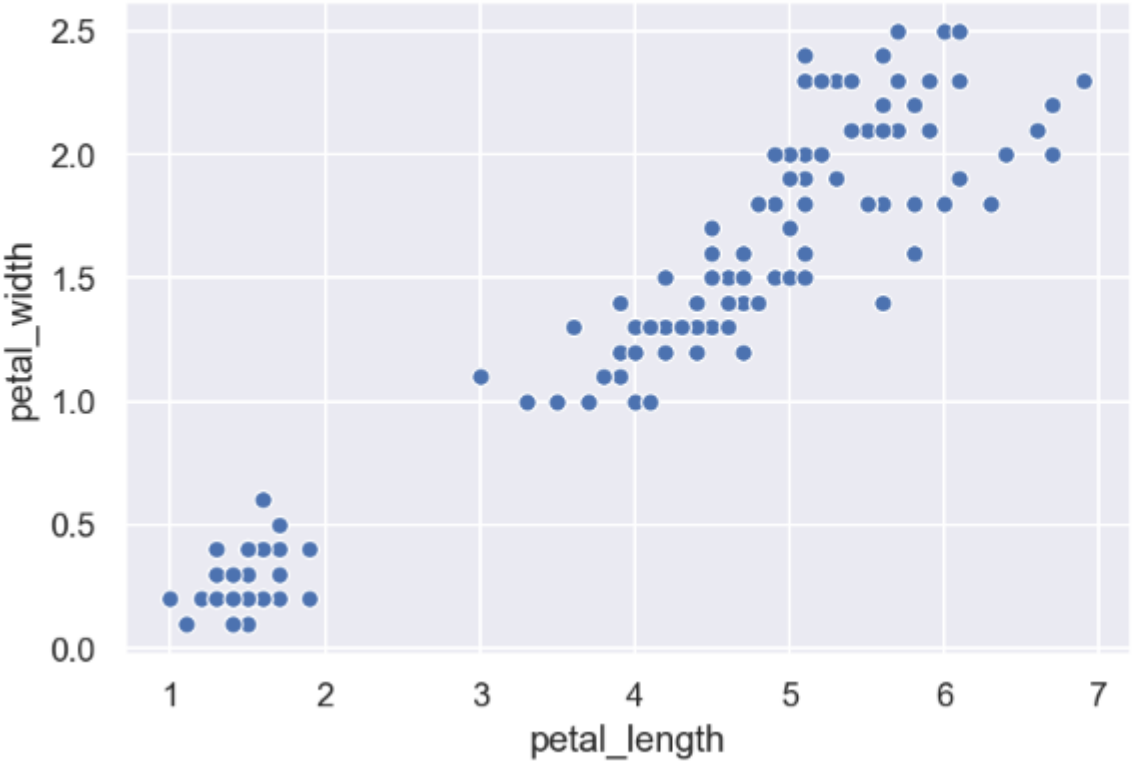
Seaborn으로 그래프 그리기

- 관계형(Relational), 범주형(Categorical), 분포(Distribution), 회귀(Regression), 행렬(Matrix)과 관련된 그래프 함수들 제공
- 이들 함수는 스스로 그래프를 그리는 기능이 있기 때문에 그래프 함수를 실행하면 바로 그래프를 그릴 수 있음
- 이들 그래프 함수들은 그래프가 그려지는 Axes(Matplotlib의 AxesSubplot) 객체를 반환하기 때문에 서브플롯(subplots())으로 나눈 축(axes) 영역에 그래프를 그릴 수 있음
- 씨본은 하나의 화면에 여러 개 그래프를 그릴 수 있도록 그리드 객체(FacetGrid, PairGird, JointGrid)를 지원하기 때문에 figure를 데이터에 따라 축의 수를 나눠서 그래프를 그릴 수 있음

scatterplot

- scatterplot()은 산점도 그래프를 그려줌

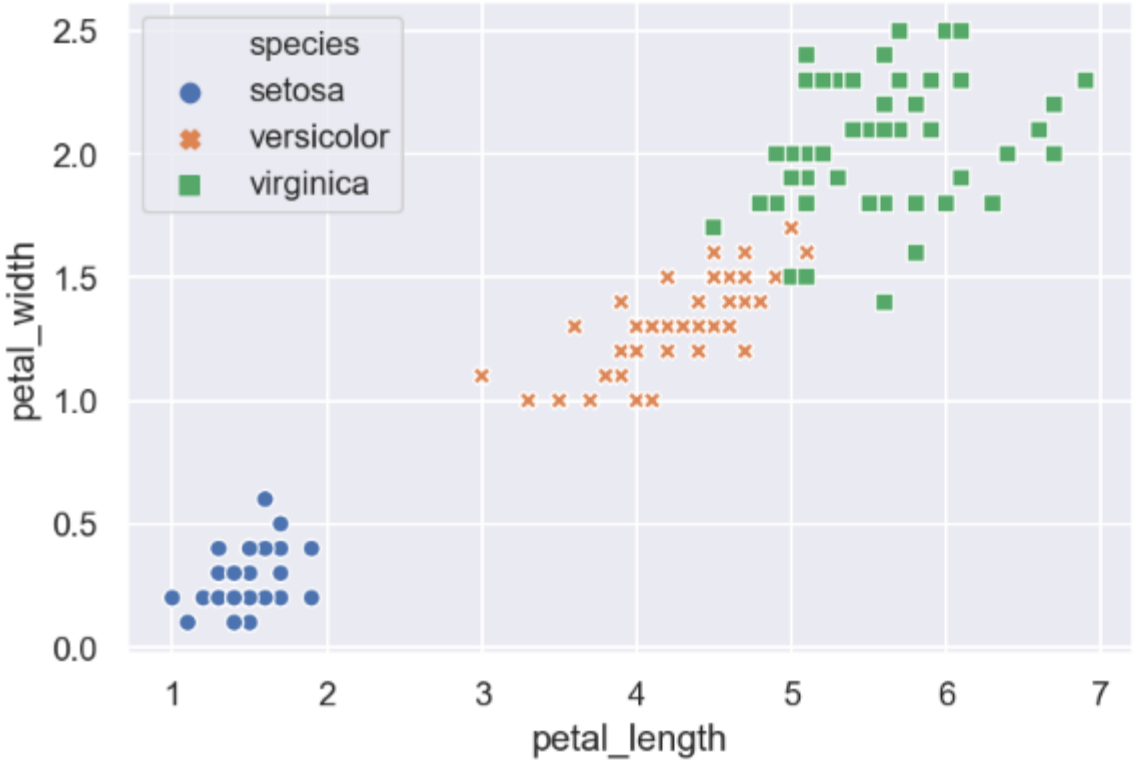
```
1  ax = sns.scatterplot(x="petal_length", y="petal_width", data=iris)
```



scatterplot

- 다른 변수로 그룹화 하고 다른 색상과 점의 스타일로 그룹을 표시

```
1  ax = sns.scatterplot(x="petal_length", y="petal_width",
2                       hue="species", style="species", data=iris)
```

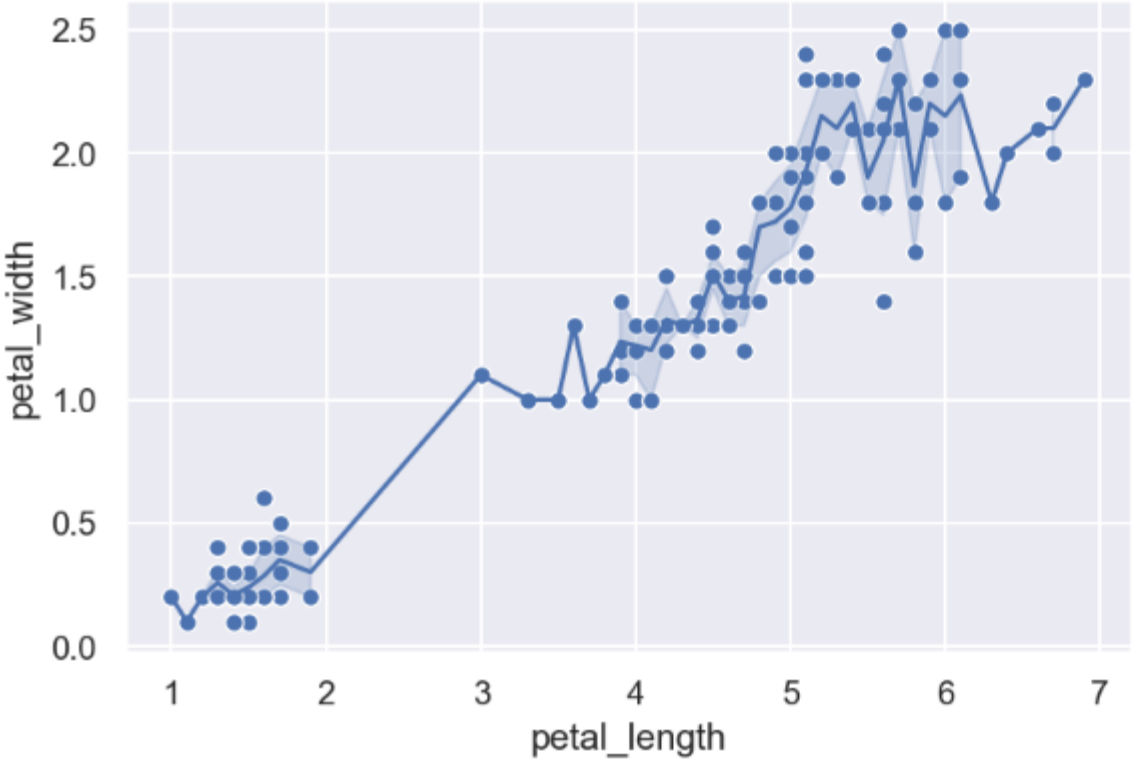


그래프 겹쳐 그리기

https://stackoverflow.com/questions/30490740/move-legend-outside-figure-in-seaborn-tsplot

- 축 하나에 그래프를 여러 개 그릴 수 있음

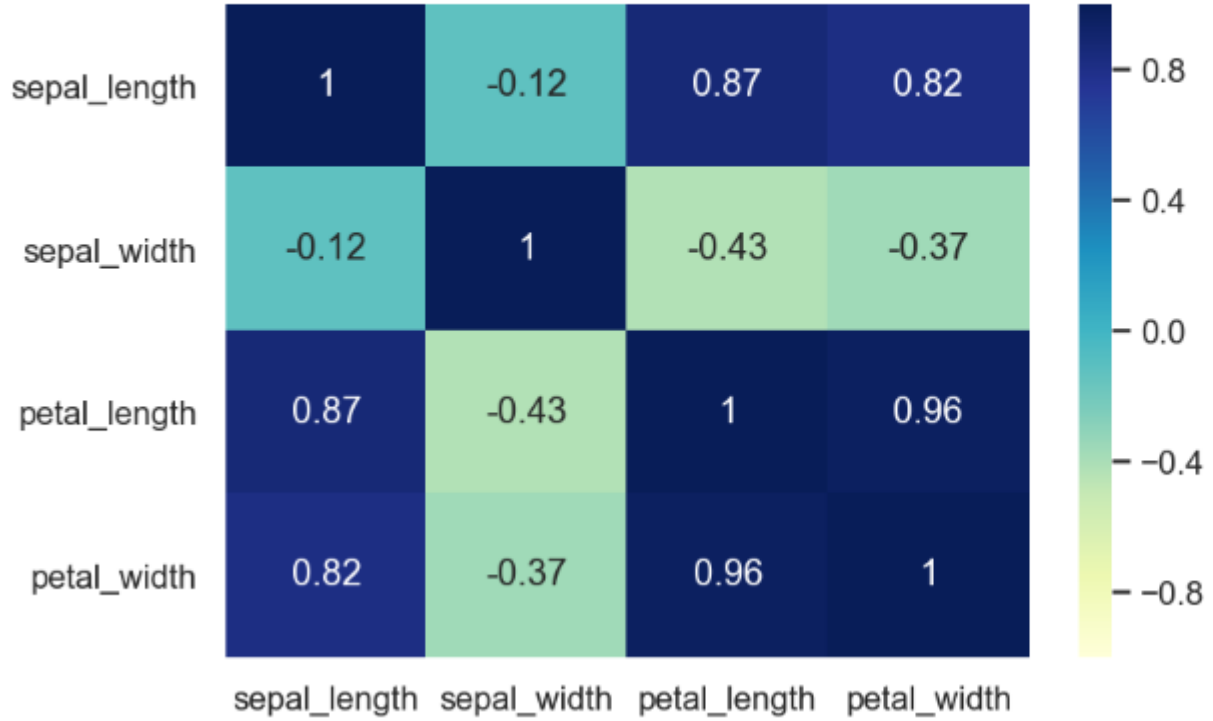
```
1  ax = sns.scatterplot(x="petal_length", y="petal_width", data=iris)
2  ax = sns.lineplot(x="petal_length", y="petal_width", data=iris)
```



heatmap

- 데이터를 색으로 인코딩 된 직사각형 행렬(히트맵)로 표시

```
1  ax = sns.heatmap(iris.corr(), vmin=-1, vmax=1, annot=True, cmap="YlGnBu")
```



annot=True이면 값을 표시