10. 데이터 시각화 (Visualization) #2

데이터 시각화

- 일반적으로 사람은 필요한 정보의 80% 가량을 시각을 통해서 받아들임
- 수치 데이터보다 시각적으로 보이는 그림이 더 직관적으로 이해할 수 있음
- 그러므로 효과적인 시각화는 데이터를 분석하고 추론하는 데 중요함

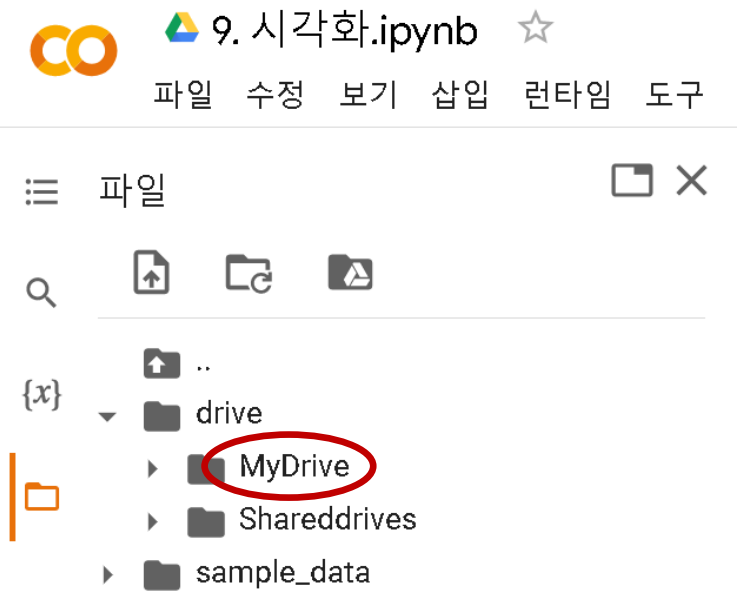
파이썬 데이터 시각화 도구

- matplotlib
  - 파이썬에서 널리 사용되는 시각화 도구로 간단한 막대, 선, 산점도 그래프를 생성
- 판다스 시각화 도구
  - plot( ) 시각화 메소드를 내장 (matplotlib 를 사용)
- seaborn
  - Matplotlib을 기반으로 고급 기능 및 통계용 차트 등이 추가됨
  - pip install seaborn

Colab 에 파일 업로드 및 읽기

```
[3]  from google.colab import drive
     drive.mount('/content/drive')

     Mounted at /content/drive
```

1) 구글 드라이브 마운트 (연결)



2) 구글 드라이브로 파일 업로드

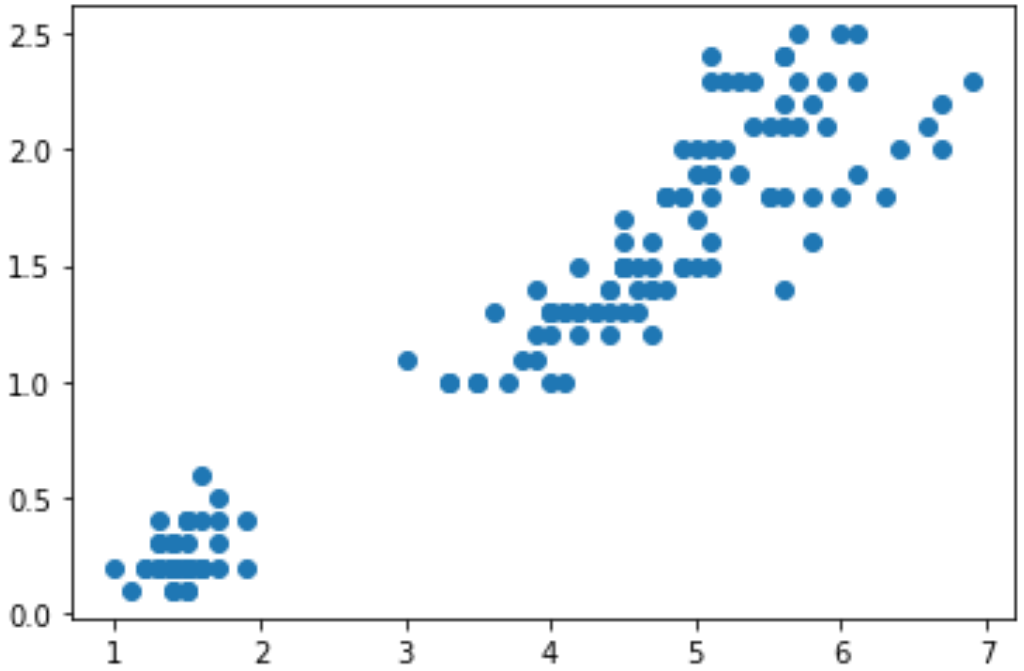
```
[4]  import pandas as pd

     filename = '/content/drive/My Drive/etc/iris.csv'
     df = pd.read_csv(filename)
     print(df)
```

3) 파일 위치 경로 작성 및 읽기

matplotlib 시각화 예시

- PetalLength 와 PetalWidth 두 값에 대한 산점도 그래프

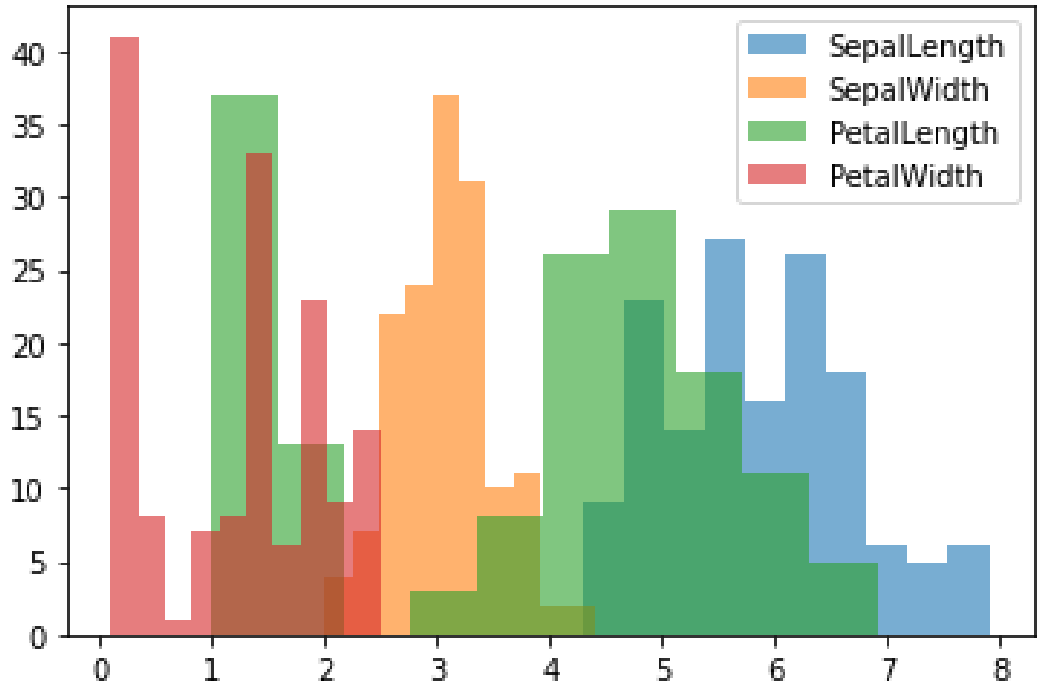
```
X = df["PetalLength"]
Y = df["PetalWidth"]
plt.scatter(X, Y)
plt.show()
```



matplotlib 시각화 예시

- 4개의 특성에 대한 히스토그램을 한 그래프로 표현

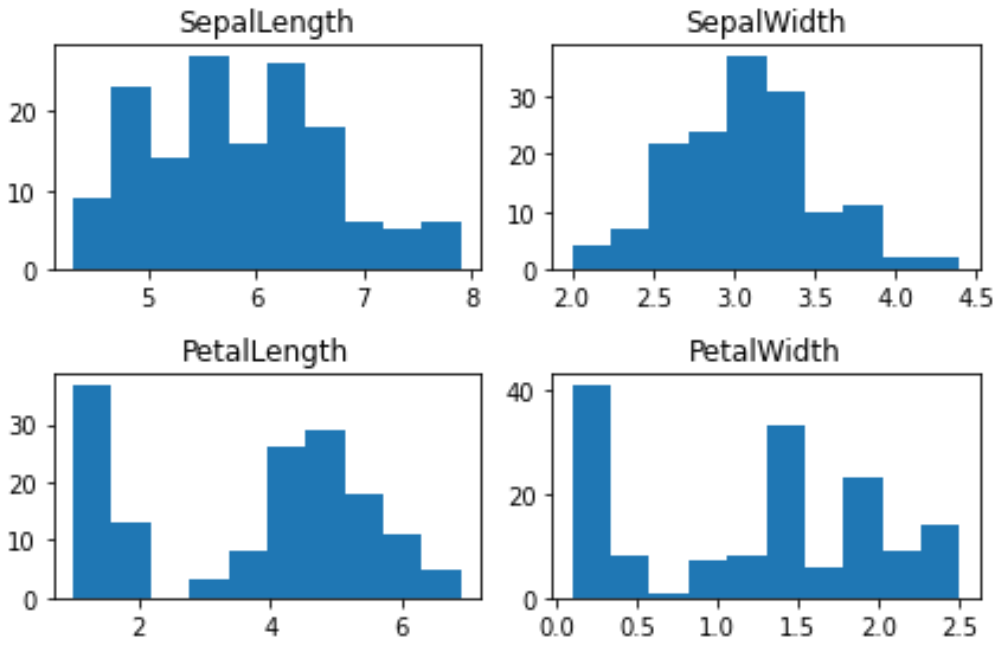
```
Y1 = df["SepalLength"]
Y2 = df["SepalWidth"]
Y3 = df["PetalLength"]
Y4 = df["PetalWidth"]

plt.hist(Y1, alpha=0.6, label="SepalLength")
plt.hist(Y2, alpha=0.6, label="SepalWidth")
plt.hist(Y3, alpha=0.6, label="PetalLength")
plt.hist(Y4, alpha=0.6, label="PetalWidth")
plt.legend()
plt.show()
```



matplotlib 시각화 예시

```
plt.subplot(2,2,1)
plt.hist(Y1)
plt.title("SepalLength")
plt.subplot(2,2,2)
plt.hist(Y2)
plt.title("SepalWidth")
plt.subplot(2,2,3)
plt.hist(Y3)
plt.title("PetalLength")
plt.subplot(2,2,4)
plt.hist(Y4)
plt.title("PetalWidth")
plt.tight_layout()
plt.show()
```

- 4개 히스토그램을 subplot() 으로 재구성



판다스 시각화

- 판다스는 데이터를 시각화 하는 라이브러리 matplotlib의 기능을 일부분 내장
- 별도로 import 하지 않아도 간단하게 데이터를 그래프로 표현할 수 있음

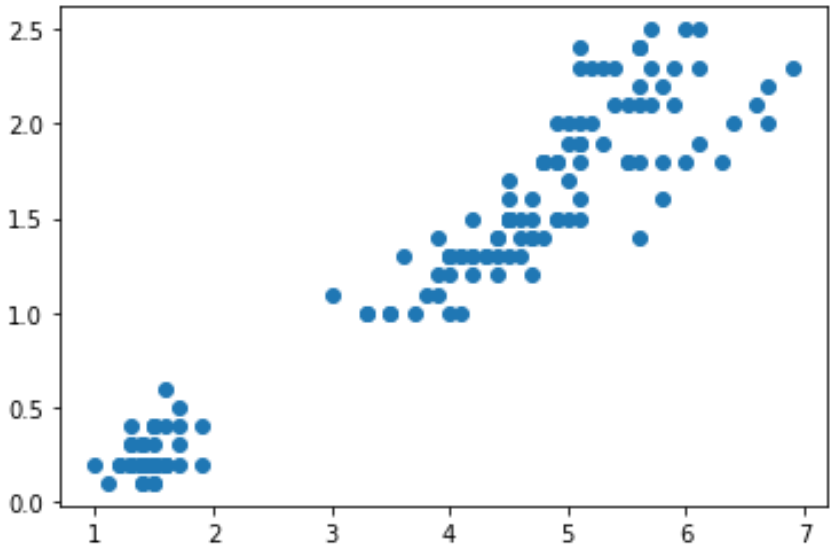
https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html#plot-formatting

Option	종류	Option	종류
'line'	선 그래프	'kde'	커널 밀도 그래프
'bar'	막대 그래프 - 수직	'area'	면적 그래프
'barh'	막대 그래프 - 수평	'pie'	원형 그래프
'his'	히스토그램 그래프	'scatter'	산점도 그래프
'box'	박스 그래프(사분위수)	'hexbin'	고밀도 산점도 그래프

판다스 시각화 예시

- 데이터프레임 메소드 plot.scatter( ) 를 호출

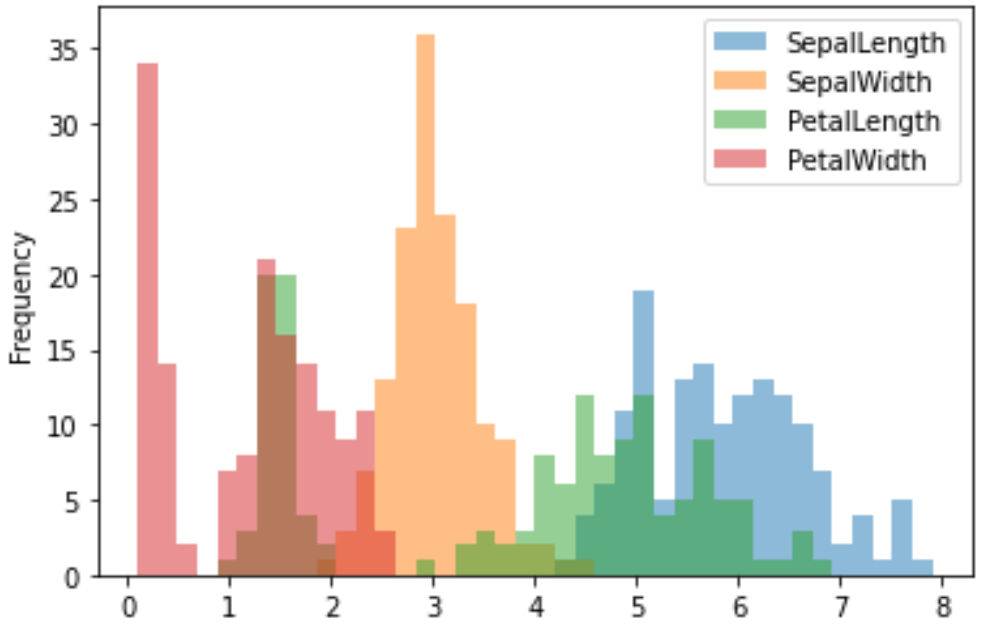
```
df.plot.scatter("PetalLength", "PetalWidth")
plt.show()
```



판다스 시각화 예시

- 필요한 열에 대해 데이터프레임을 구성한 후 plot.hist( ) 메소드를 사용

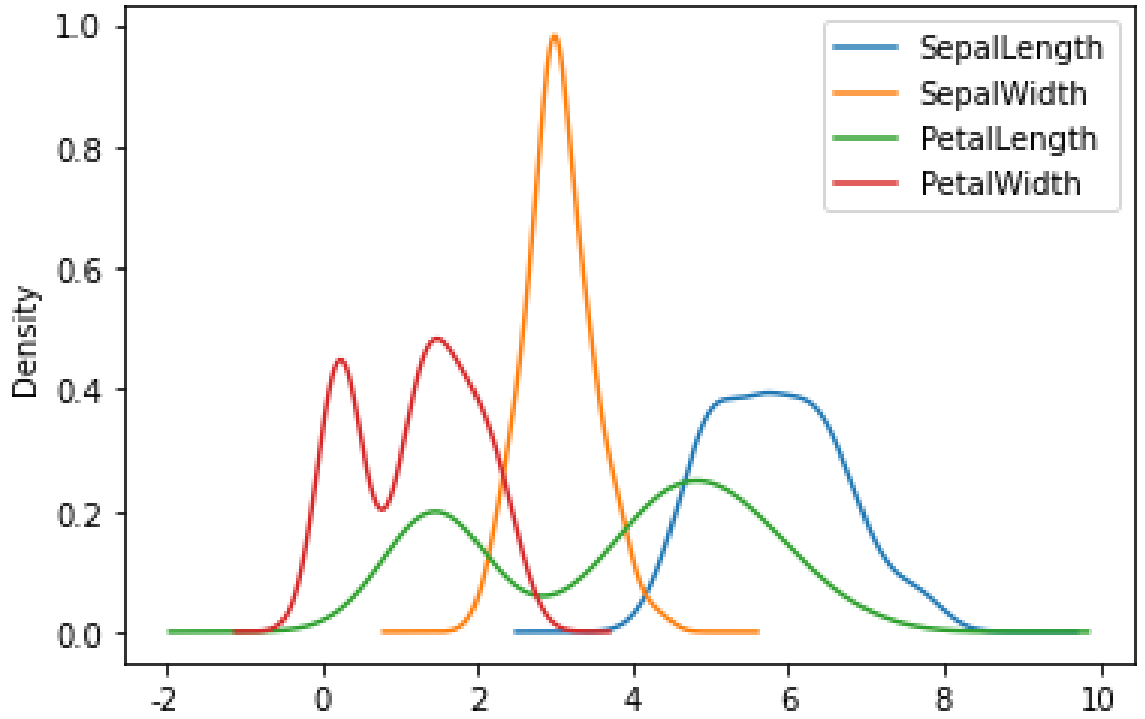
```
df2=df.drop("caseno", axis=1)
df2.plot.hist(alpha=0.5, bins=40)
plt.show()
```



판다스 시각화 예시

- 커널밀도추정 (Kerner Density Estimation, KDE) 그래프 지원
  - 히스토그램 분포를 부드럽게 선으로 표현

```
df2=df.drop("caseno", axis=1)
df2.plot.kde()
plt.show()
```



판다스 시각화 예시

- iris 데이터셋에서 꽃의 종류에 따른 특징 값 (feature) 평균

```
df2 = df.drop( "caseno", axis=1 )
gb = df2.groupby( "Species" ).mean()
print(gb)
print()
print(gb.T)
```

Species	SepalLength	SepalWidth	PetalLength	PetalWidth
setosa	5.006	3.428	1.462	0.246
versicolor	5.936	2.770	4.260	1.326
virginica	6.588	2.974	5.552	2.026

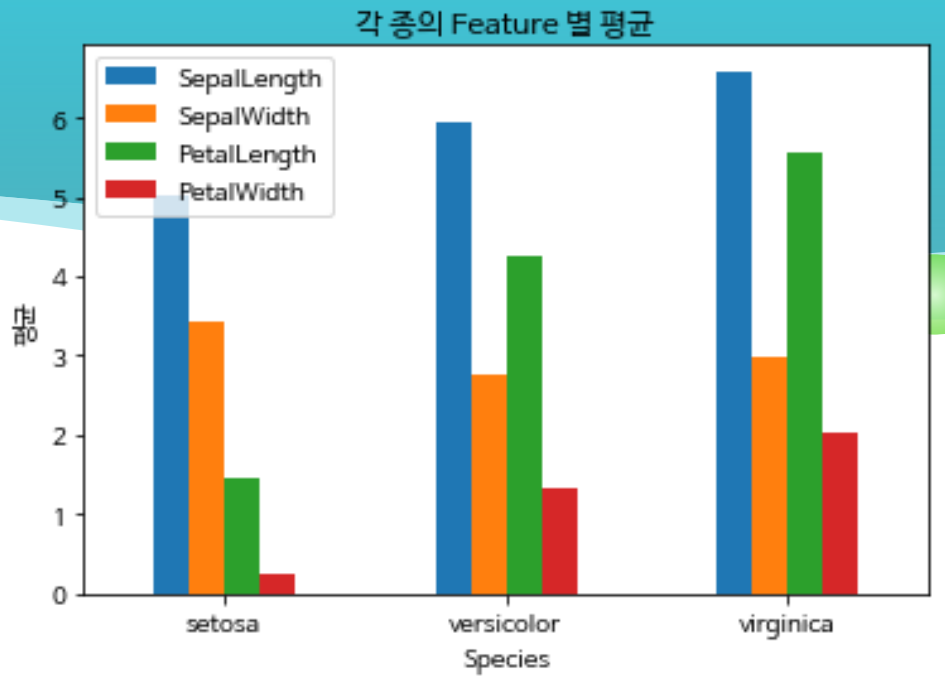
Species	setosa	versicolor	virginica
SepalLength	5.006	5.936	6.588
SepalWidth	3.428	2.770	2.974
PetalLength	1.462	4.260	5.552
PetalWidth	0.246	1.326	2.026

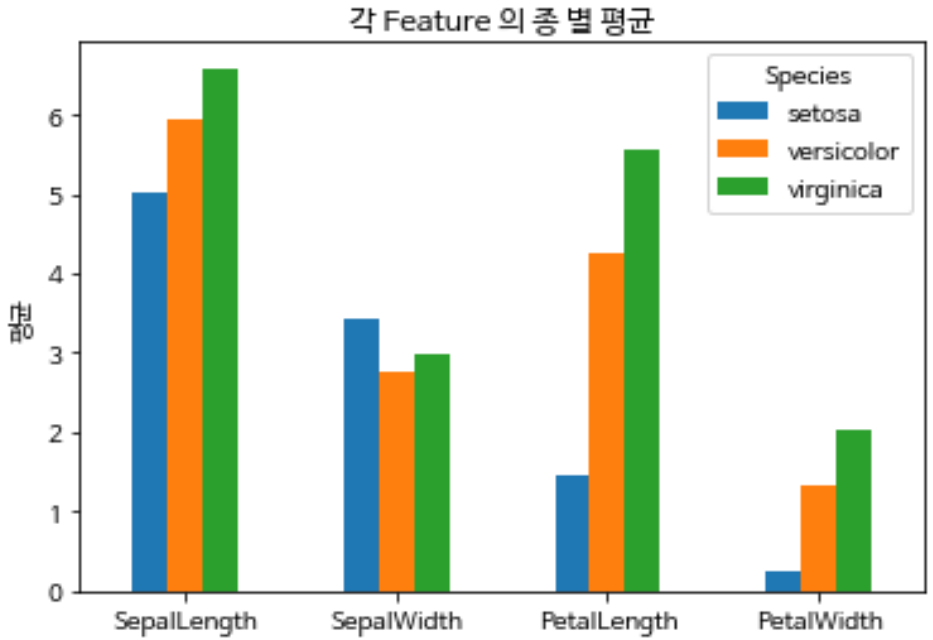
DataFrame.T : **행렬을 주 대각선을 기준으로 하여 뒤집음**

판다스 시각화 예시

- plot.bar( ) 메소드 사용

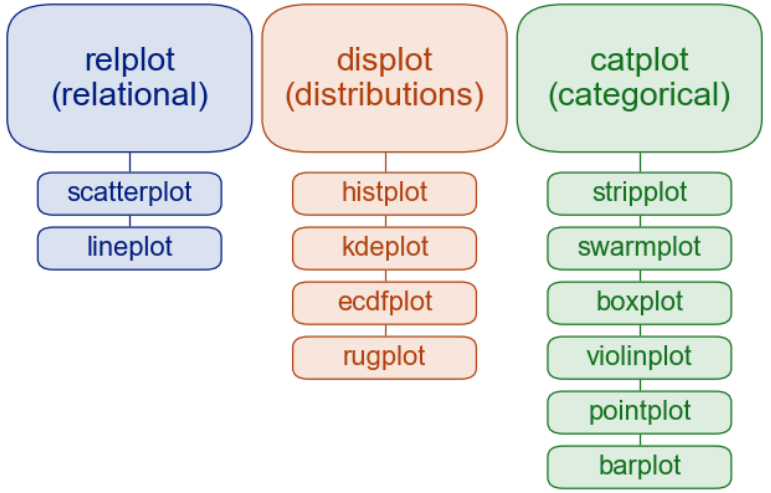
```
# 앞에 이어서
gb.plot.bar(rot=0)
plt.title("각 종의 Feature별 평균")
plt.ylabel("평균")
plt.show()
gb.T.plot.bar(rot=0)
plt.title("각 종의 Feature별 평균")
plt.ylabel("평균")
plt.show()
```





seaborn 라이브러리

- Matplotlib을 기반으로 다양한 색상 테마와 통계용 차트 등의 기능을 추가한 시각화 패키지
  - matplotlib 경험자는 쉽게 익힐 수 있음
  - 고수준 인터페이스를 제공하여 보다 쉽고 편리하게 시각화 가능
  - 보다 세련된 그래프로 시각화 효율성 높임
  - 연습용 데이터셋을 함께 제공



- **설치**: pip install seaborn

http://seaborn.pydata.org/

seaborn 라이브러리

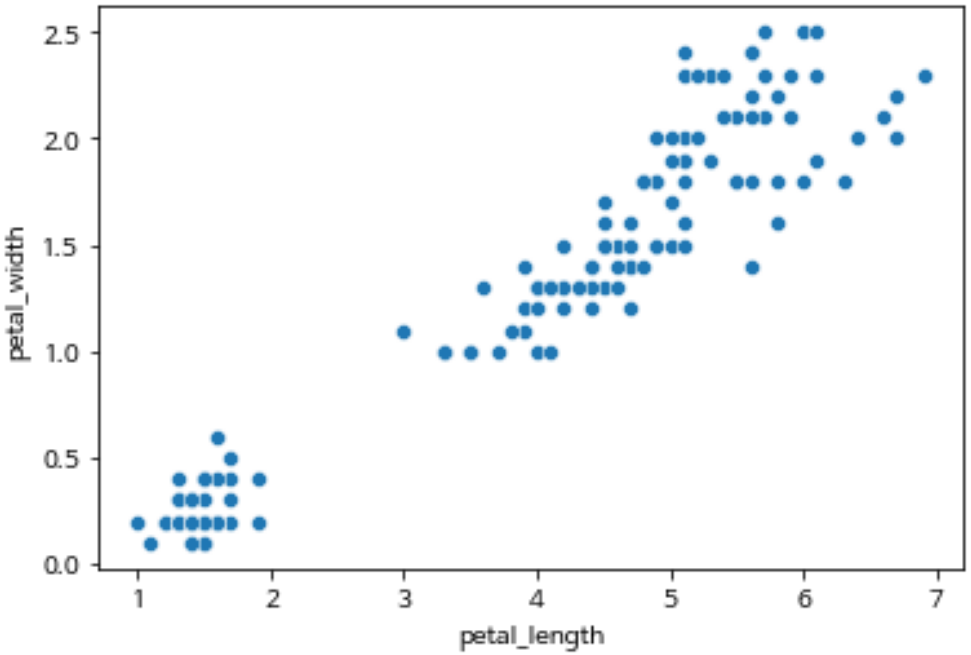
내장된 데이터셋 불러오기

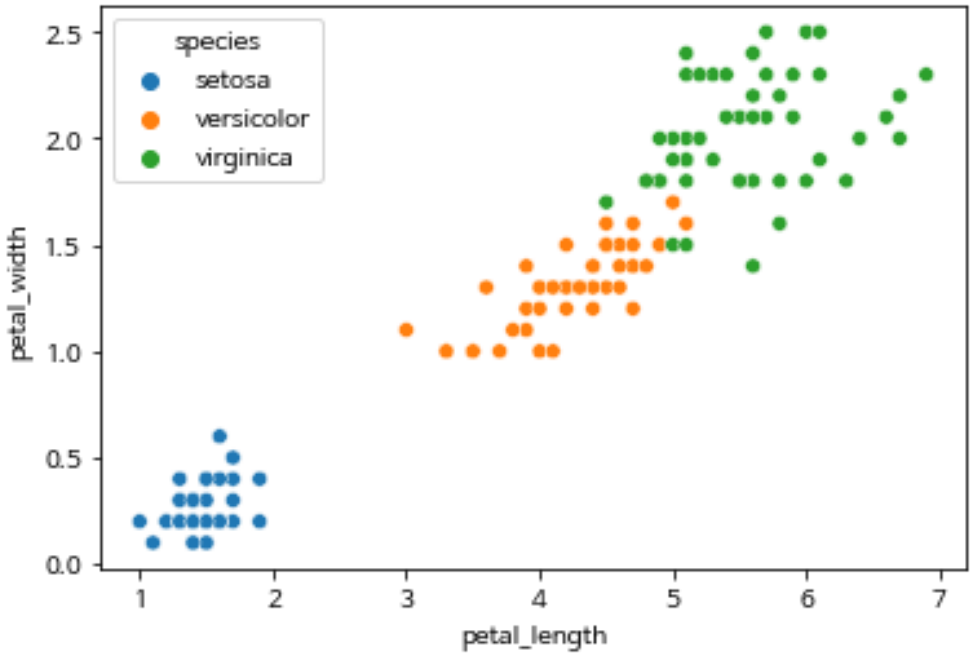
```
import seaborn as sns
iris = sns.load_dataset("iris")
print(iris)
# "caseno" 컬럼이 없는 것을 확인
```

```
     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 : scatterplot( )

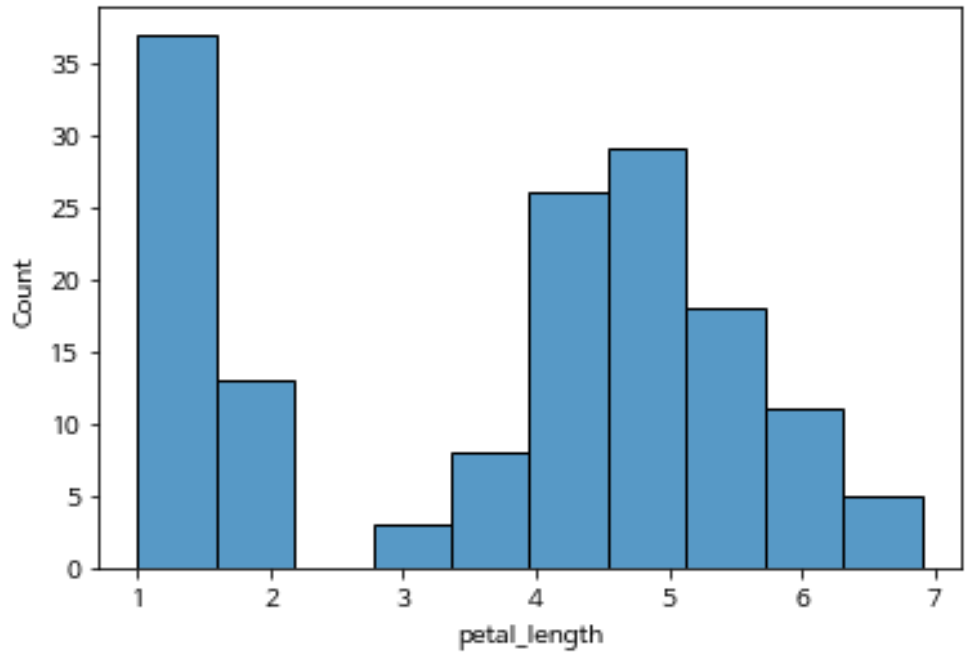
```
sns.scatterplot(data=iris, x="petal_length", y="petal_width")
plt.show()
sns.scatterplot(data=iris, x="petal_length", y="petal_width", hue="species")
plt.show()
```

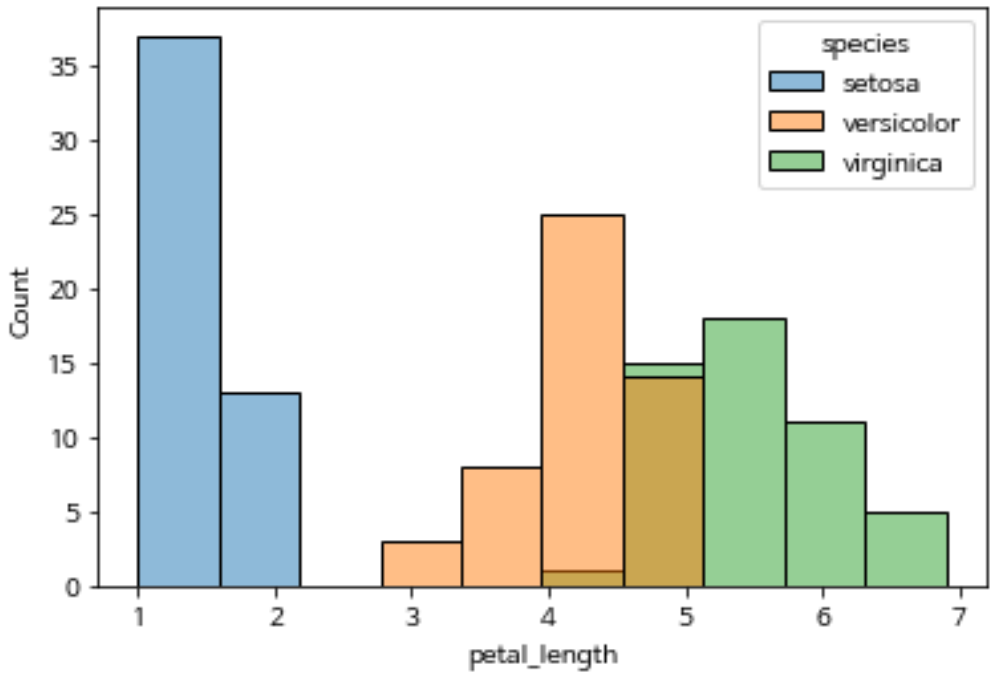




seaborn : hisplot( )

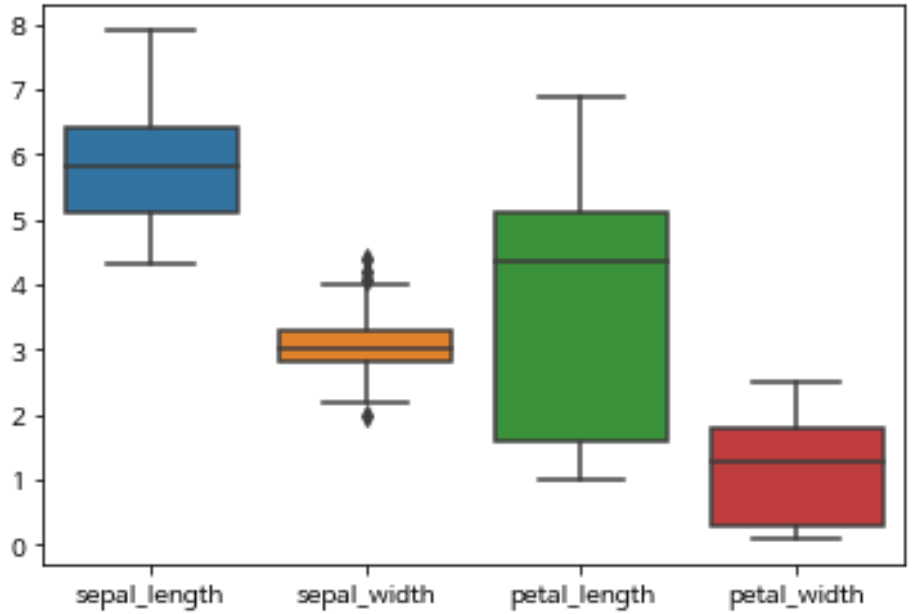
```
sns.histplot(data=iris, x="petal_length", bins=10)
plt.show()
sns.histplot(data=iris, x="petal_length", bins=10, hue="species")
plt.show()
```



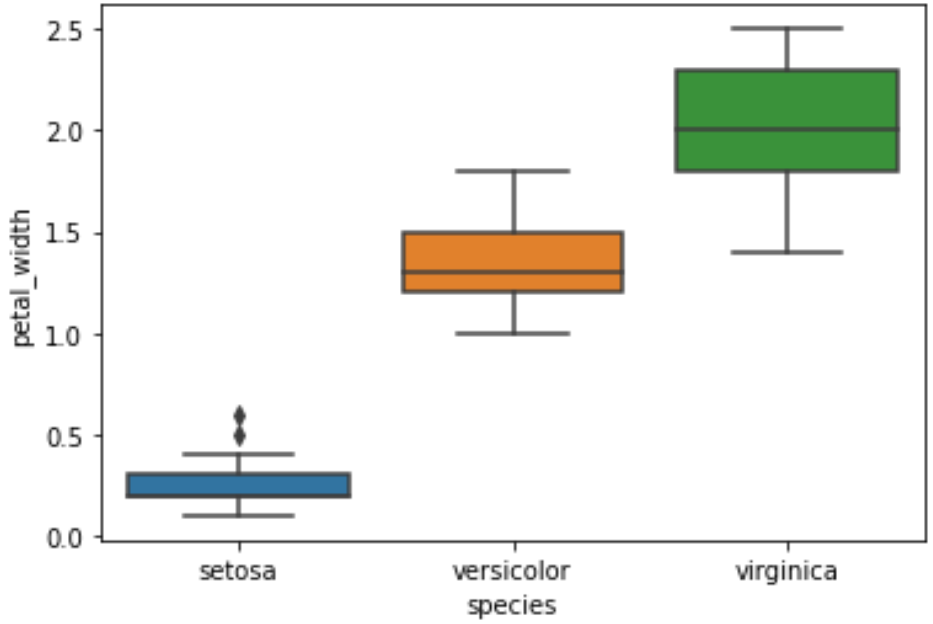


seaborn : boxplot( )

```
sns.boxplot(data=iris)
plt.show()
```

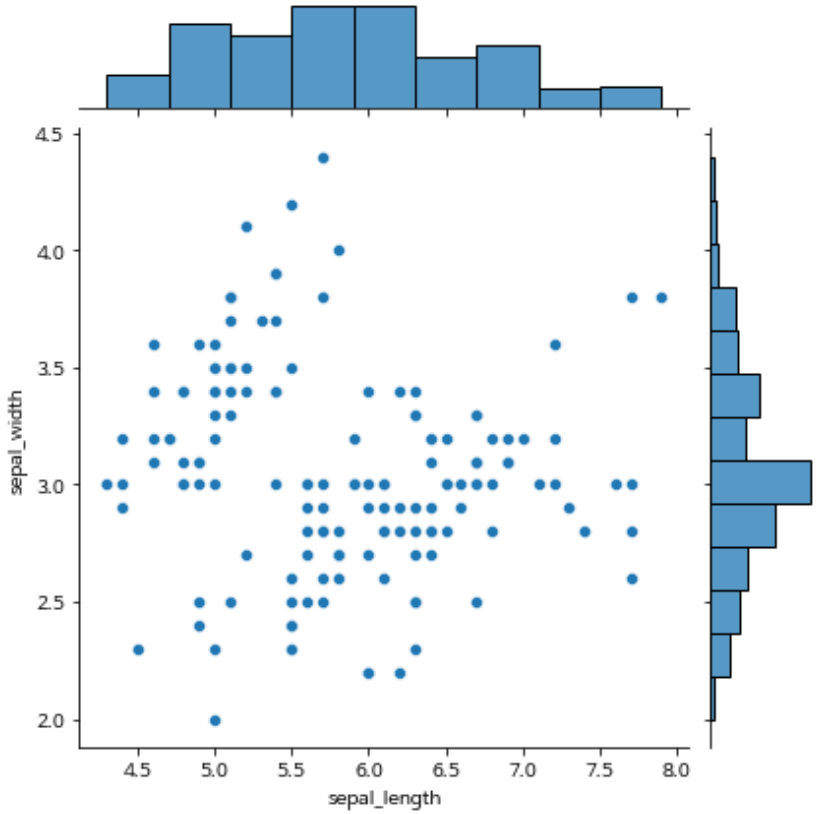


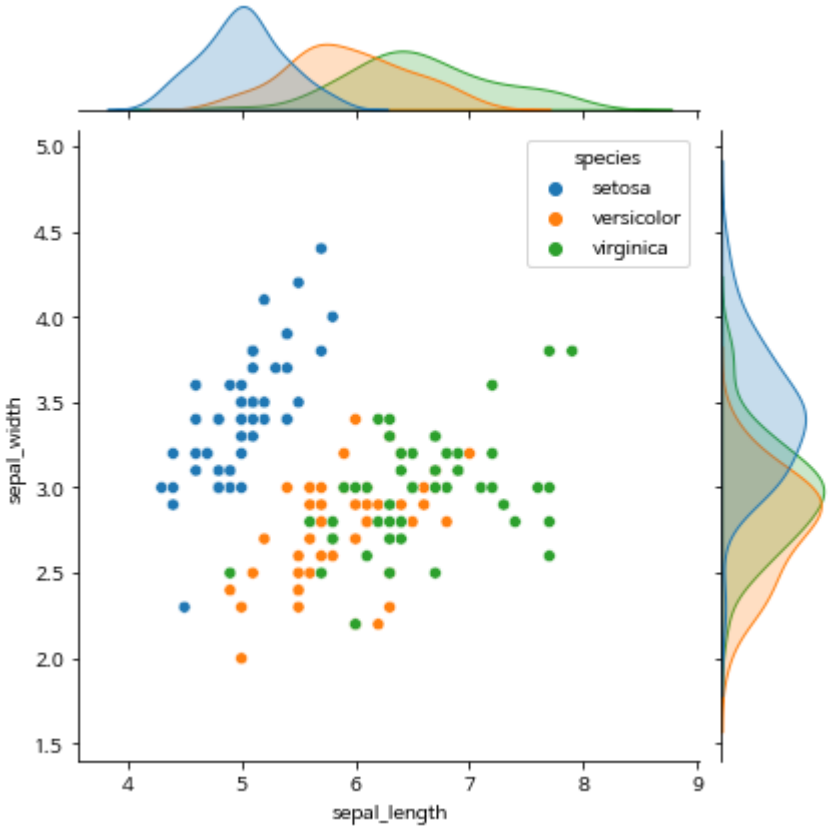

```
sns.boxplot(data=iris, x="species",
            y="petal_width")
plt.show()
```



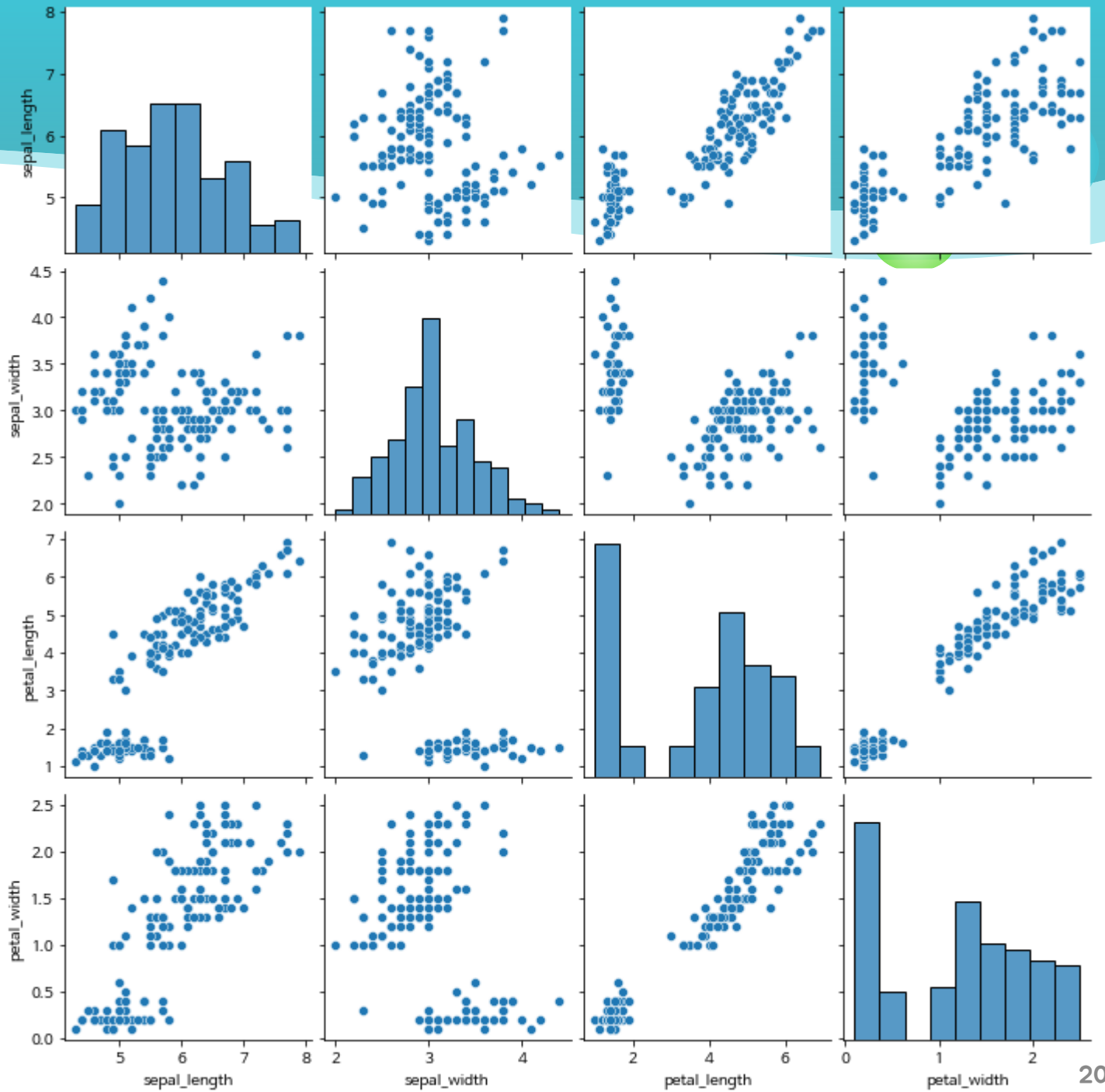
seaborn : jointplot( )

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

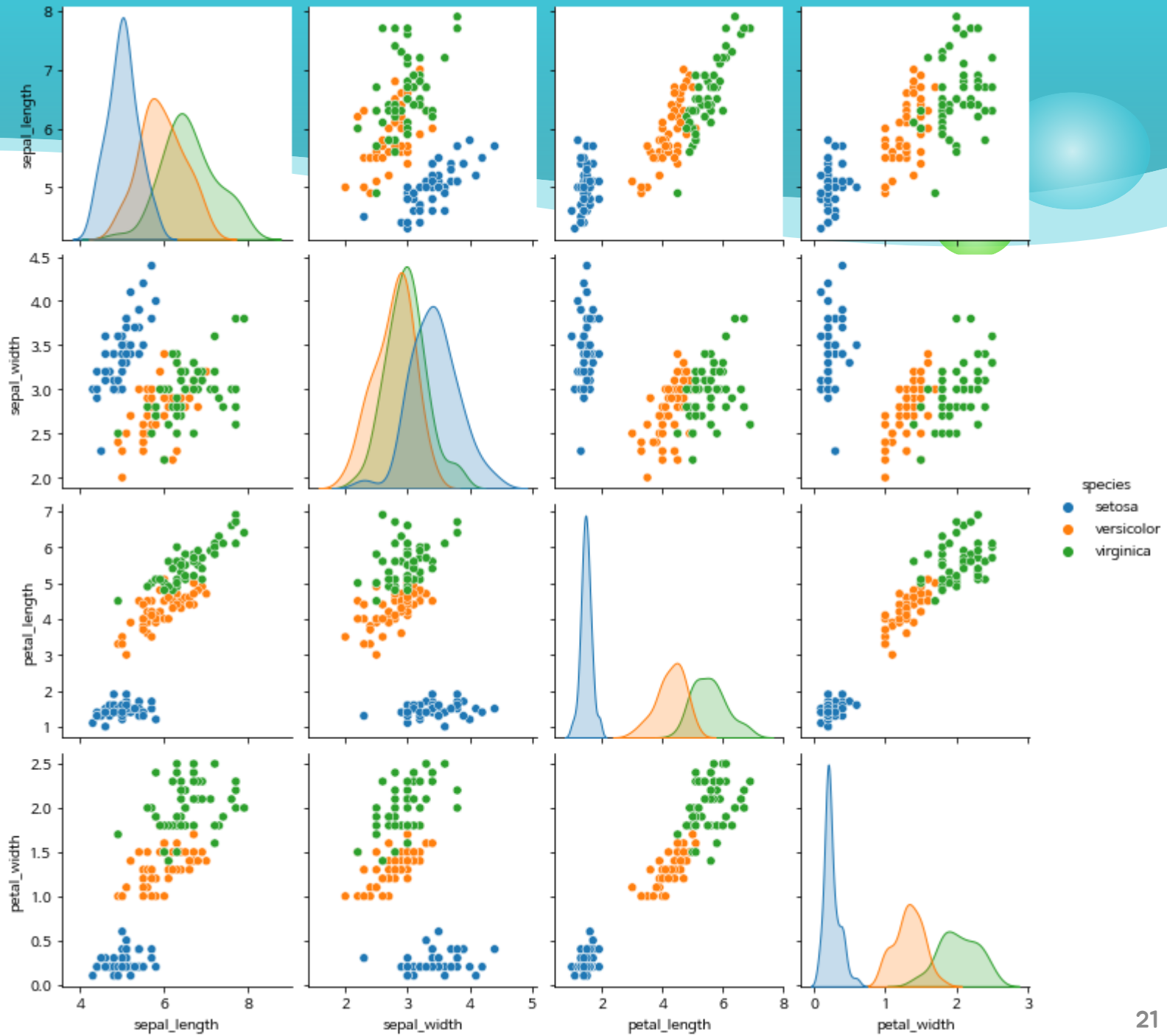




- seaborn: pairplot()
  - 그리드(grid) 형태로 각 데이터 열의 조합에 대해 scatter plot
  - 같은 데이터가 만나는 대각선 영역에는 해당 데이터의 histogram



- seaborn: pairplot()
  - hue 인자를 사용하여 특정 값에 따른 색조 구분



시각화 라이브러리 요약

- matplotlib
  - 가장 기본적인 시각화 라이브러리
  - 리스트, 판다스 등 자료구조 모두 사용 가능
- pandas plot
  - 판다스 내장된 plot( ) 메소드 사용
  - 데이터프레임 시각화에 용이
- seaborn
  - 보다 세련된 그래프로 시각화
  - 고수준 시각화 인터페이스 제공
  - 연습 데이터셋 함께 제공

데이터 시각화 연습 : flights

- seaborn 에서 제공하는 "flights" 데이터셋 불러오기

```
flights = sns.load_dataset("flights")
print(flights)
print(flights['year'].value_counts())
```

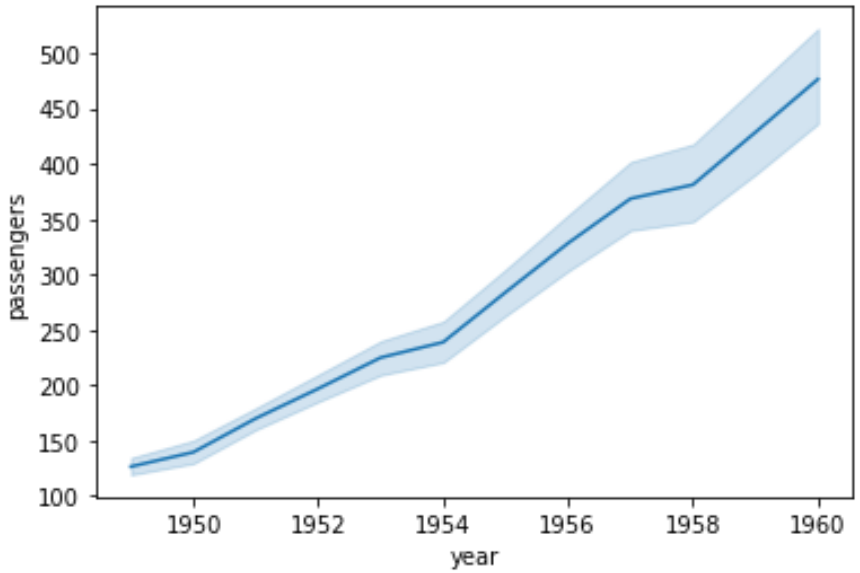
```
     year month  passengers
0    1949   Jan         112
1    1949   Feb         118
2    1949   Mar         132
3    1949   Apr         129
4    1949   May         121
..    ...   ...         ...
139  1960   Aug         606
140  1960   Sep         508
141  1960   Oct         461
142  1960   Nov         390
143  1960   Dec         432

[144 rows x 3 columns]
```

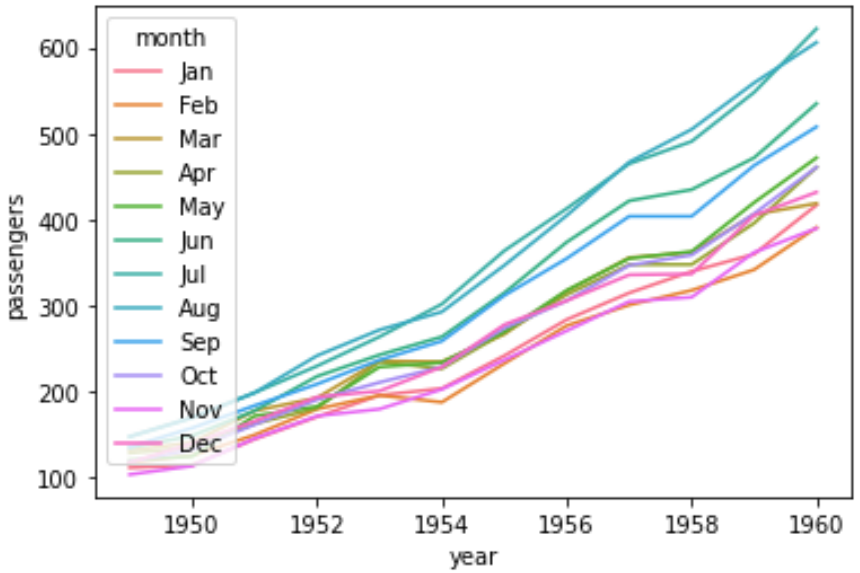
```
1949    12
1950    12
1951    12
1952    12
1953    12
1954    12
1955    12
1956    12
1957    12
1958    12
1959    12
1960    12
Name: year, dtype: int64
```

데이터 시각화 연습 : flights

- lineplot( ) 을 사용하여 연도별 승객 수를 시각화

```
sns.lineplot(data=flights, x="year", y="passengers")
plt.show()
sns.lineplot(data=flights, x="year", y="passengers", hue="month")
plt.show()
```

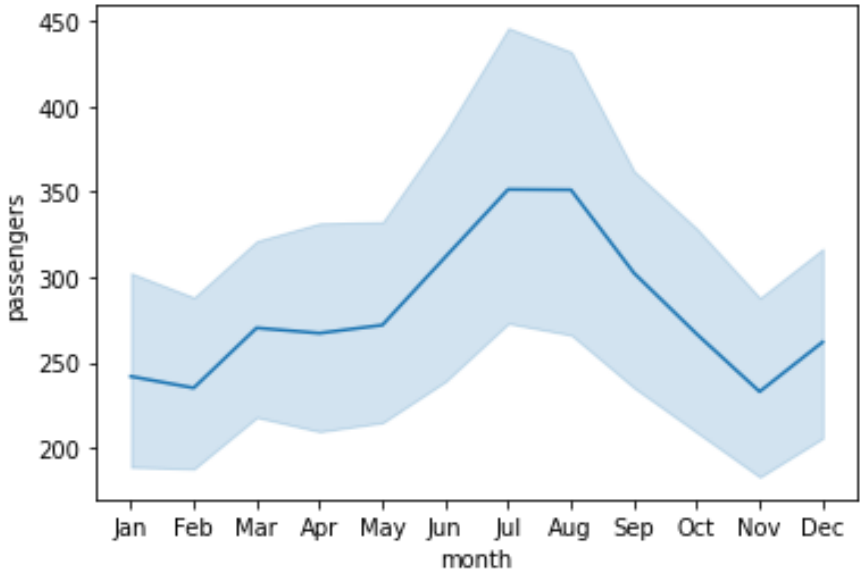


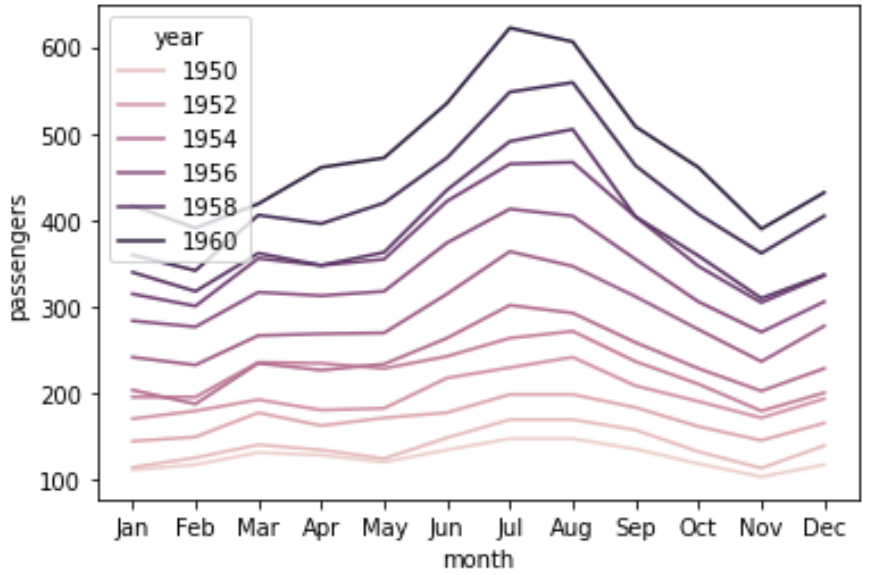


데이터 시각화 연습 : flights

- lineplot( ) 을 사용하여 월별 승객 수 를 시각화

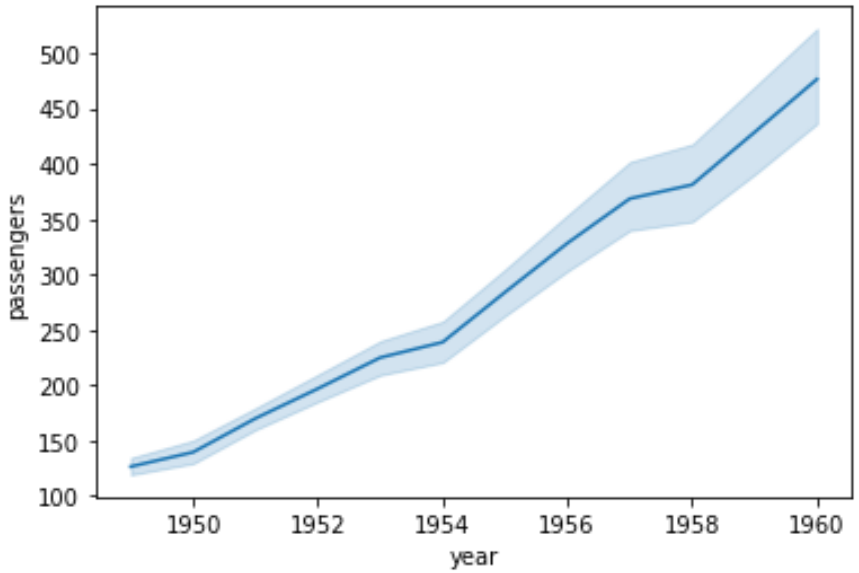
```
sns.lineplot(data=flights, x="month ", y="passengers")
plt.show()
sns.lineplot(data=flights, x="month", y="passengers", hue="year")
plt.show()
```

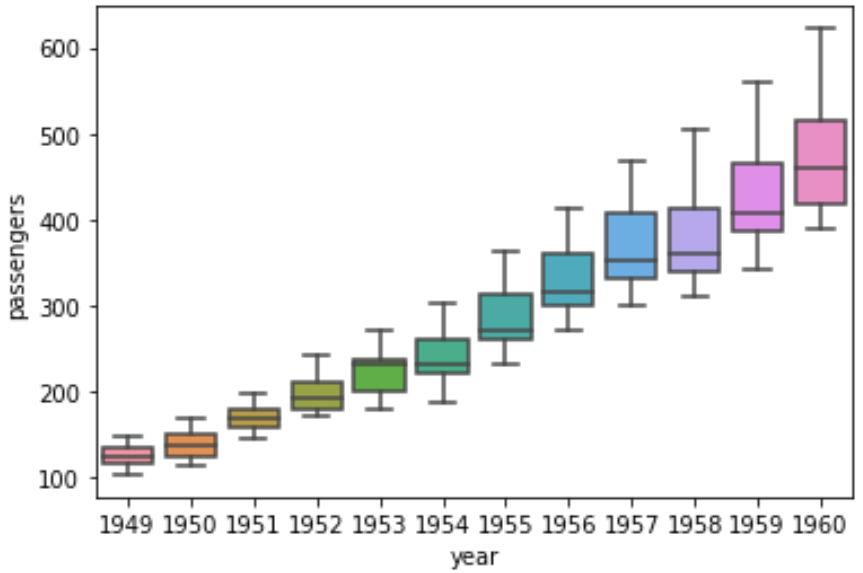




데이터 시각화 연습 : flights

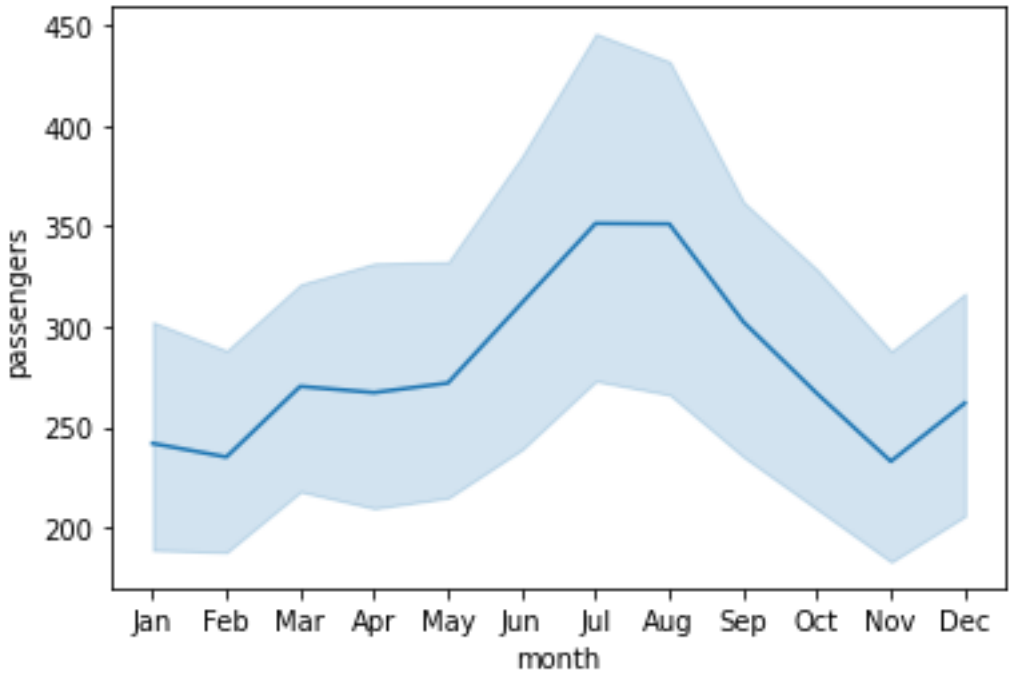
```
sns.lineplot(data=flights, x="year", y="passengers")
plt.show()
sns.boxplot(data=flights, x="year", y="passengers")
plt.show()
```

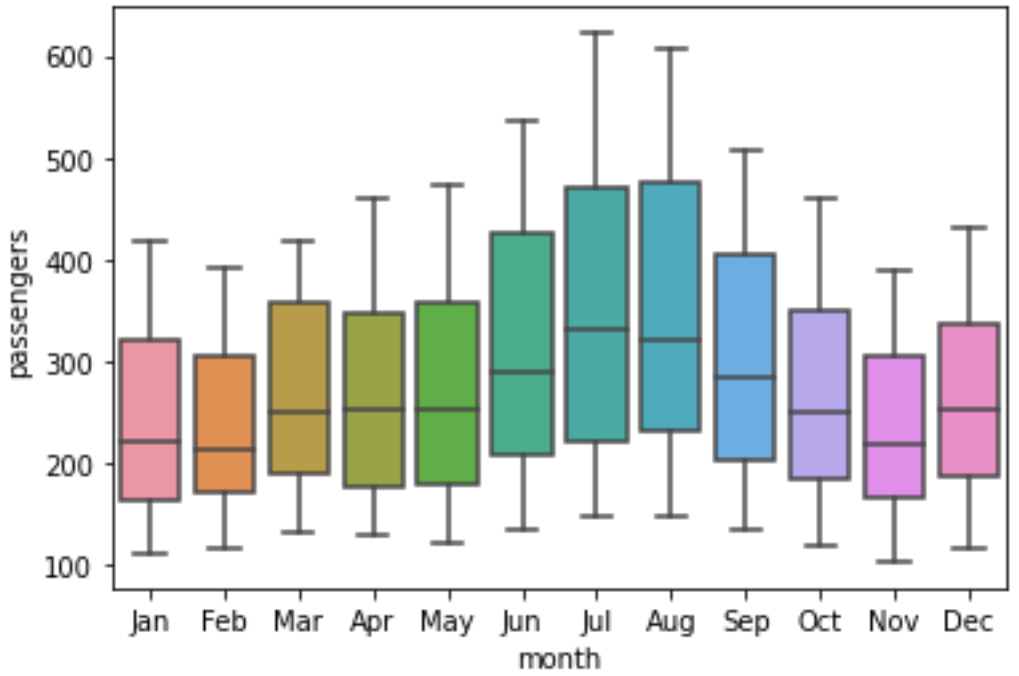




데이터 시각화 연습 : flights

```
sns.lineplot(data=flights, x="month ", y="passengers")
plt.show()
sns.boxplot(data=flights, x="month", y="passengers")
plt.show()
```





데이터 시각화 연습 : flights

pivot( ) 함수로 테이블 재구성

```
print(flights)
pvt = flights.pivot("month", "year", "passengers")
print(pvt)
```

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121
..	...	...	...
139	1960	Aug	606
140	1960	Sep	508
141	1960	Oct	461
142	1960	Nov	390
143	1960	Dec	432

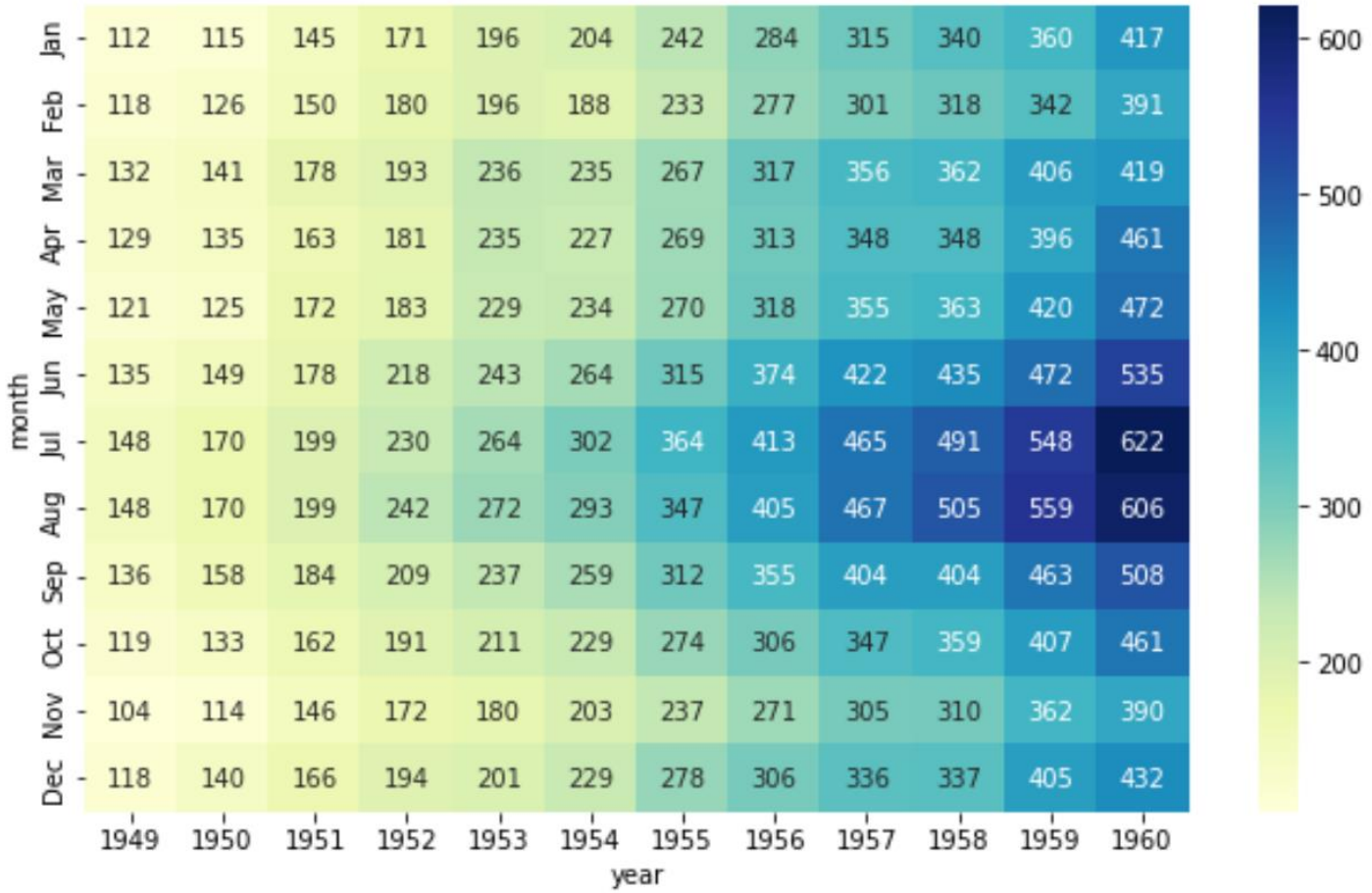
[144 rows x 3 columns]

year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
Jan	112	115	145	171	196	204	242	284	315	340	360	417
Feb	118	126	150	180	196	188	233	277	301	318	342	391
Mar	132	141	178	193	236	235	267	317	356	362	406	419
Apr	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472
Jun	135	149	178	218	243	264	315	374	422	435	472	535
Jul	148	170	199	230	264	302	364	413	465	491	548	622
Aug	148	170	199	242	272	293	347	405	467	505	559	606
Sep	136	158	184	209	237	259	312	355	404	404	463	508
Oct	119	133	162	191	211	229	274	306	347	359	407	461
Nov	104	114	146	172	180	203	237	271	305	310	362	390
Dec	118	140	166	194	201	229	278	306	336	337	405	432

데이터 시각화 연습 : flights

- 히트맵 (heatmap) 그래프

```
plt.figure( figsize=(10,6))
sns.heatmap(pvt, annot=True,
            fmt="d", cmap="YlGnBu")
plt.show()
```



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