

[Python] 시각화 사용법 - seaborn을 통한 그래프 만들기 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-11-10 오후 11:57

URL: <https://continuous-development.tistory.com/153?category=736681>

Python

[Python] 시각화 사용법 - seaborn을 통한 그래프 만들기

2020. 10. 22. 17:48 수정 삭제 공개

seaborn 이란

Seaborn은 Matplotlib에 기반하여 제작된 파이썬 데이터 시각화 모듈이다. 고수준의 인터페이스를 통해 직관적이고 아름다운 그래프를 그릴 수 있다

seaborn 사용법

```
import seaborn as sns
```

일반적으로 as를 sns로 한다.

간단한 예제

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import matplotlib as mpl

import datetime as dt
```

```
import seaborn as sns
```

```
import json
```

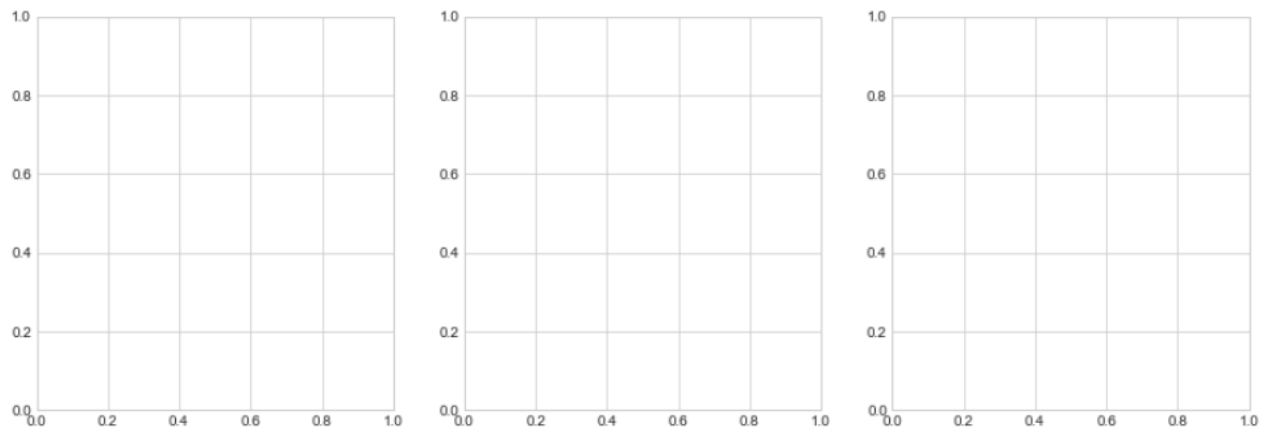
```
iris = sns.load_dataset('iris')  
  
titanic = sns.load_dataset('titanic')  
  
tips = sns.load_dataset('tips')  
  
flights = sns.load_dataset('flights')
```

seaborn 에서 제공해주는 데이터 들이다. 이걸 통해 여러 그래프를 그려보자.

barplot

스타일 테마를 설정 5가지 (darkgrid, whitegrid, dark, white, ticks)

```
sns.set_style('whitegrid')  
  
# 그래프 객체를 생성하기 위한 피겨 객체 얻어오기  
fig = plt.figure(figsize=(15,5))  
  
# 하나의 fig(도화지라고 생각하면 된다) 에 3 가지 그래프를 그리는 방법이다.  
# subplot(행 열 인덱스) 라고 생각하면 된다.  
area01 = fig.add_subplot(1,3,1)  
area01 = fig.add_subplot(1,3,2)  
area01 = fig.add_subplot(1,3,3)  
  
plt.show()
```



스타일에 따라 다르게 나오는 fig

그래프 객체를 생성하기 위한 피겨 객체 얻어오기

```
fig = plt.figure(figsize=(15,5))
```

```
sns.set_style('ticks')
```

```
area01 = fig.add_subplot(1,3,1)
```

```
area02 = fig.add_subplot(1,3,2)
```

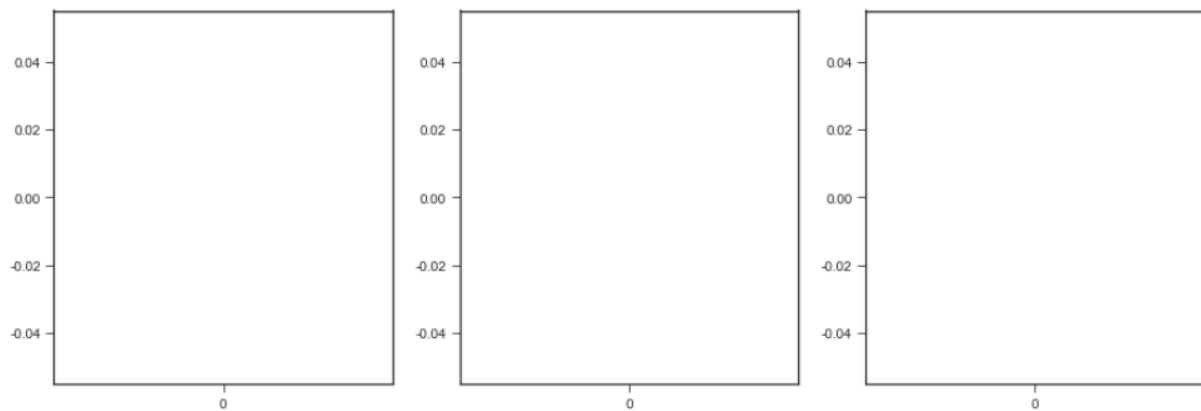
```
area03 = fig.add_subplot(1,3,3)
```

```
sns.barplot(ax=area01)
```

```
sns.barplot(ax=area02)
```

```
sns.barplot(ax=area03)
```

```
plt.show()
```



```
# 그래프 객체를 생성하기 위한 피겨 객체 얻어오기
fig = plt.figure(figsize=(15,5))

sns.set_style('dark')

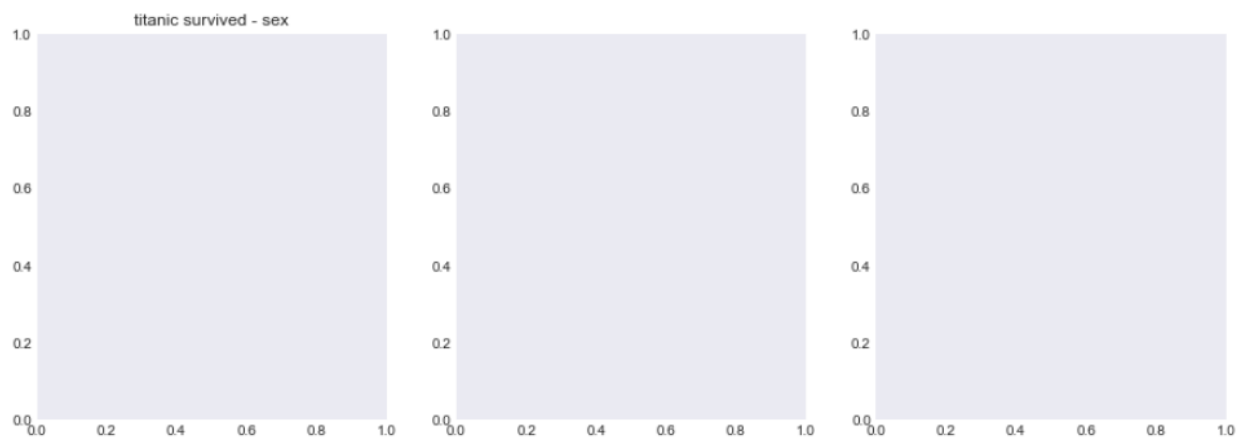
area01 = fig.add_subplot(1,3,1)
area01.set_title('titanic survived - sex')

area02 = fig.add_subplot(1,3,2)
area03 = fig.add_subplot(1,3,3)
```

```
# 그래프 객체를 생성하기 위한 피겨 객체 얻어오기
fig = plt.figure(figsize=(15,5))
sns.set_style('dark')

area01 = fig.add_subplot(1,3,1)
area01.set_title('titanic survived - sex')

area02 = fig.add_subplot(1,3,2)
area03 = fig.add_subplot(1,3,3)
```



```
# 그래프 객체를 생성하기 위한 피겨 객체 얻어오기
fig = plt.figure(figsize=(15,5))
sns.set_style('dark')

area01 = fig.add_subplot(1,3,1)
area01.set_title('titanic survived - sex')

area02 = fig.add_subplot(1,3,2)
area03 = fig.add_subplot(1,3,3)

# 성별에 따른 생존률
sns.barplot(x='sex', y='survived', data=titanic, ax=area01)
sns.barplot(x='sex', y='survived', hue='class', data=titanic, ax=area02) # hue로 그룹화가 된다.
```

```
sns.barplot(x='sex', y='survived', hue='class', dodge=False, data=titanic, ax=area03) # dodge 는 누적 bar 그래프를 만
plt.show()
```

그래프 객체를 생성하기 위한 피겨 객체 얻어오기

```
fig = plt.figure(figsize=(15,5))
sns.set_style('dark')
```

```
area01 = fig.add_subplot(1,3,1)
area01.set_title('titanic survived - sex')
```

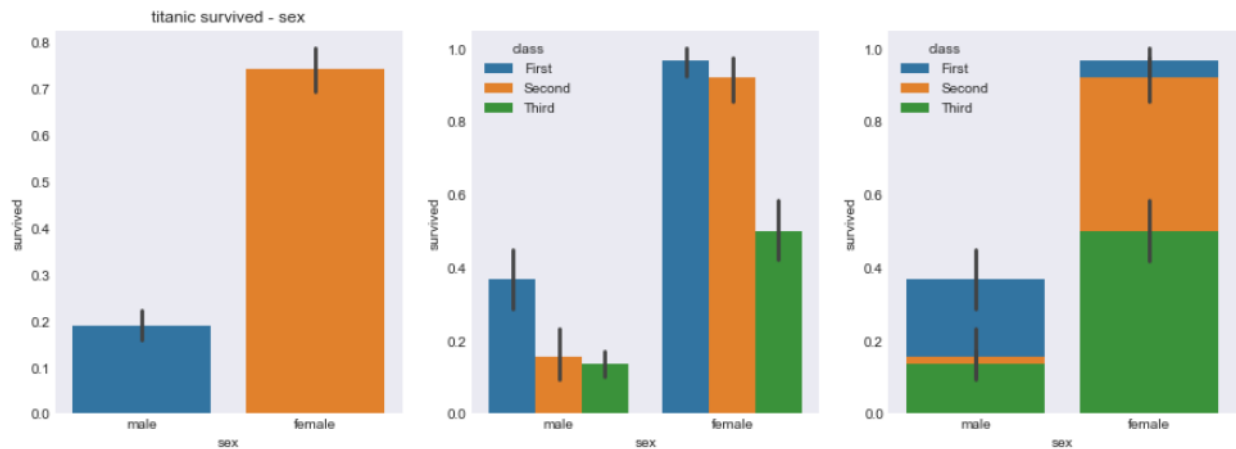
```
area02 = fig.add_subplot(1,3,2)
area03 = fig.add_subplot(1,3,3)
```

#성별에 따른 생존율

```
sns.barplot(x='sex', y='survived', data=titanic, ax=area01)
```

```
sns.barplot(x='sex', y='survived', hue='class', data=titanic, ax=area02) # hue로 그룹화가 된다.
```

```
sns.barplot(x='sex', y='survived', hue='class', dodge=False, data=titanic, ax=area03) # dodge 는 누적 bar 그래프를 만든다.
plt.show()
```



countplot

카테고리 값별로 데이터 빈도를 보는 차트다.

y축을 그리지 않으며 해당 변수의 개수를 y축으로 사용한다.

그래프 객체를 생성하기 위한 피겨 객체 얻어오기

```
fig = plt.figure(figsize=(15,5))
```

```
sns.set_style('whitegrid')
```

```
area01 = fig.add_subplot(1,3,1)
```

```
area01.set_title('titanic class')
```

```
area02 = fig.add_subplot(1,3,2)
```

```
area02.set_title('titanic class - who')
```

```
area03 = fig.add_subplot(1,3,3)
```

```
area03.set_title('titanic class - who(stacked)')
```

#countplot은 y 축이었다 count가 y로 되기 때문이다.

```
sns.countplot(x='class',data=titanic, ax=area01)
```

```
sns.countplot(x='class',hue='who',palette='Set3',data=titanic, ax=area02) # hue로 그룹화가 된다. palette로 그래프의
```

```
sns.countplot(x='class',hue='who',dodge=False,palette='Set3',data=titanic, ax=area03) # dodge는 누적 bar 그래프를 만든다.  
plt.show()
```

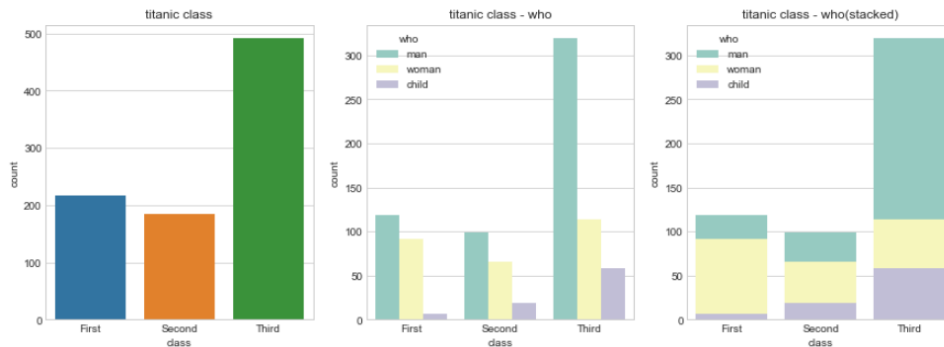
In [21]: # 그래프 객체를 생성하기 위한 피겨 객체 열어오기

```
fig = plt.figure(figsize=(15,5))  
sns.set_style('whitegrid')
```

```
area01 = fig.add_subplot(1,3,1)  
area01.set_title('titanic class')  
area02 = fig.add_subplot(1,3,2)  
area02.set_title('titanic class - who')  
area03 = fig.add_subplot(1,3,3)  
area03.set_title('titanic class - who(stacked)')
```

#countplot은 y 축이었다 count가 y로 되기 때문이다.

```
sns.countplot(x='class',data=titanic, ax=area01)  
sns.countplot(x='class',hue='who',palette='Set3',data=titanic, ax=area02) # hue로 그룹화가 된다. palette로 그래프의 타이틀도 고른다.  
sns.countplot(x='class',hue='who',dodge=False,palette='Set3',data=titanic, ax=area03) # dodge는 누적 bar 그래프를 만든다.  
plt.show()
```



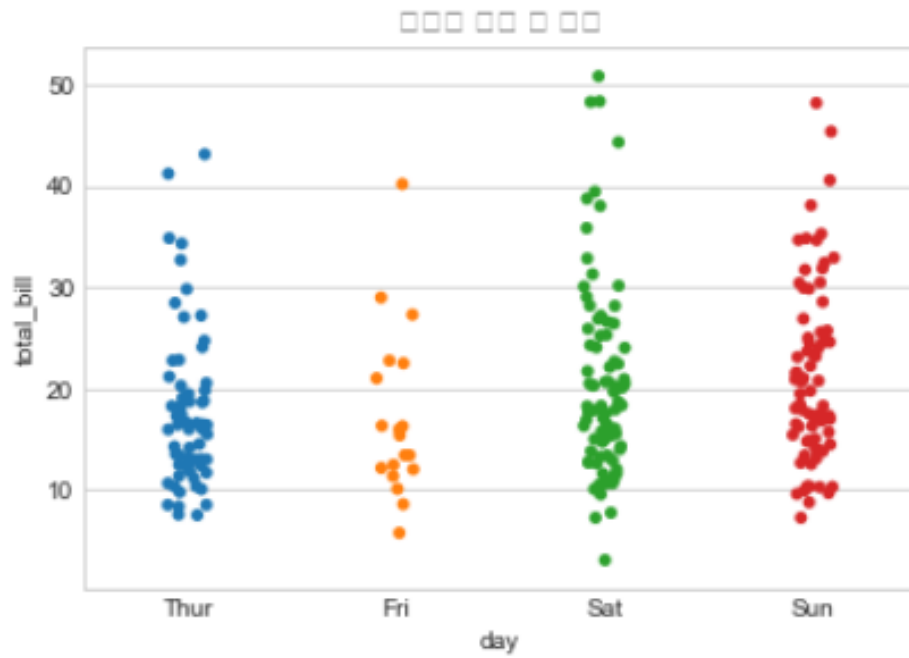
#stripplot()

스카터 플롯처럼 모든 데이터를 점으로 시각화

jitter=True 설정하면 위치를 무작위로 바꿔서 겹치지 않게 해 준다.

```
sns.stripplot(x='day',y='total_bill',data=tips,jitter=True)  
plt.title('요일별 팁을 준 횟수')  
plt.show()
```

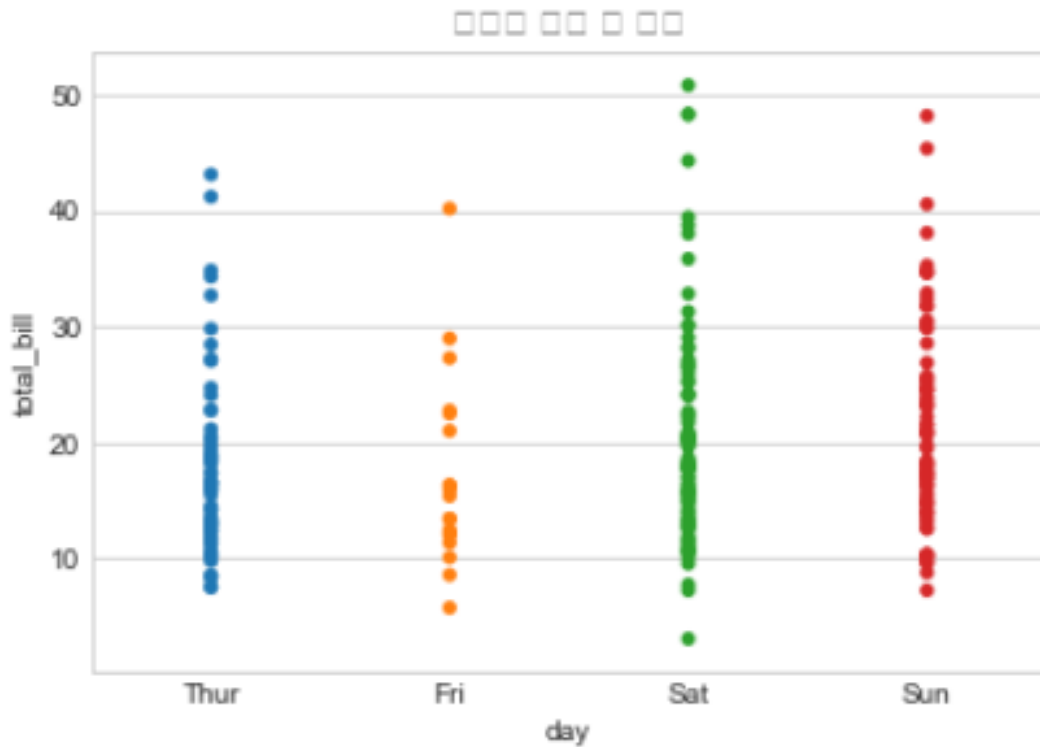
```
sns.stripplot(x='day',y='total_bill',data=tips,jitter=True)
plt.title('요일별 팁을 준 횟수')
plt.show()
```



jitter의 유무

```
sns.stripplot(x='day',y='total_bill',data=tips)
plt.title('요일별 팁을 준 횟수')
plt.show()
```

```
sns.stripplot(x='day',y='total_bill',data=tips)
plt.title('요일별 팁을 준 횟수')
plt.show()
```



그래프 객체를 생성하기 위한 피겨 객체 얻어오기

```
fig = plt.figure(figsize=(15,5))
```

```
sns.set_style('whitegrid')
```

```
area01 = fig.add_subplot(1,2,1)
```

```
area01.set_title('titanic class')
```

```
area02 = fig.add_subplot(1,2,2)
```

```
area02.set_title('titanic class - who')
```

#countplot은 y 축이없다 conut가 y로 되기 때문이다.

```
sns.stripplot(x='class',y='age',palette='Set3',data=titanic, ax=area01)
```

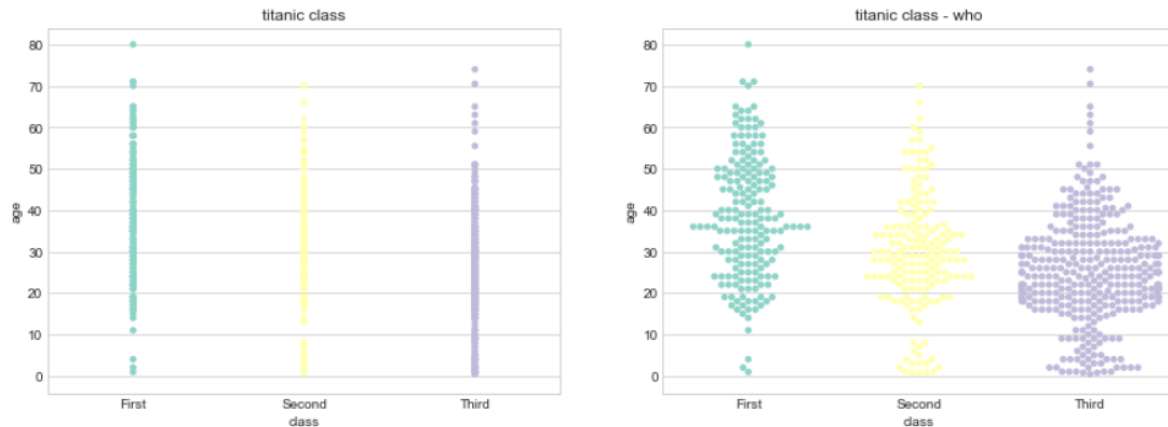
```
sns.swarmplot(x='class',y='age',palette='Set3',data=titanic, ax=area02) # hue로 그룹화가 된다. palette로 그래프 타,
plt.show()
```



```
# 그래프 객체를 생성하기 위한 피겨 객체 얻어오기
fig = plt.figure(figsize=(15,5))
sns.set_style('whitegrid')
```

```
area01 = fig.add_subplot(1,2,1)
area01.set_title('titanic class')
area02 = fig.add_subplot(1,2,2)
area02.set_title('titanic class - who')
```

```
#countplot은 y 축이없다 count가 y로 되기 때문이다.
sns.stripplot(x='class',y='age',palette='Set3',data=titanic, ax=area01)
sns.swarmplot(x='class',y='age',palette='Set3',data=titanic, ax=area02) # hue로 그룹화가 된다. palette로 그래프 타일을 고른다.
plt.show()
```



boxplot(), violinplot()

전체적인 퍼짐 정도를 한눈에 보기 좋게 보여준다.

```
sns.set_style('whitegrid')

fig = plt.figure(figsize=(15,5))

ax01 = fig.add_subplot(2,2,1)
ax01.set_title('box')
ax02 = fig.add_subplot(2,2,2)
ax02.set_title('box')
ax03 = fig.add_subplot(2,2,3)
ax03.set_title('violin')
ax04 = fig.add_subplot(2,2,4)
ax04.set_title('violin')

sns.boxplot(x='alive',y='age',data=titanic,ax=ax01)
sns.boxplot(x='alive',y='age',hue='sex',data=titanic,ax=ax02)
sns.violinplot(x='alive',y='age',data=titanic,ax=ax03)
sns.violinplot(x='alive',y='age',hue='sex',data=titanic,ax=ax04)

plt.show()
```

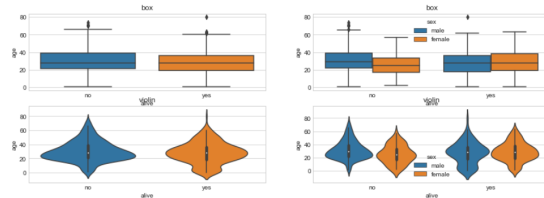
• boxplot(), violinplot()

```
sns.set_style('whitegrid')
fig = plt.figure(figsize=(15,5))

ax01 = fig.add_subplot(2,2,1)
ax01.set_title('box')
ax02 = fig.add_subplot(2,2,2)
ax02.set_title('box')
ax03 = fig.add_subplot(2,2,3)
ax03.set_title('violin')
ax04 = fig.add_subplot(2,2,4)
ax04.set_title('violin')

sns.boxplot(x='alive', y='age', data=titanic, ax=ax01)
sns.boxplot(x='alive', y='age', hue='sex', data=titanic, ax=ax02)
sns.violinplot(x='alive', y='age', data=titanic, ax=ax03)
sns.violinplot(x='alive', y='age', hue='sex', data=titanic, ax=ax04)

plt.show()
```



heatmap()

하나이상의 카테고리 값의 변화를 색깔 변화로 시각화하는 차트

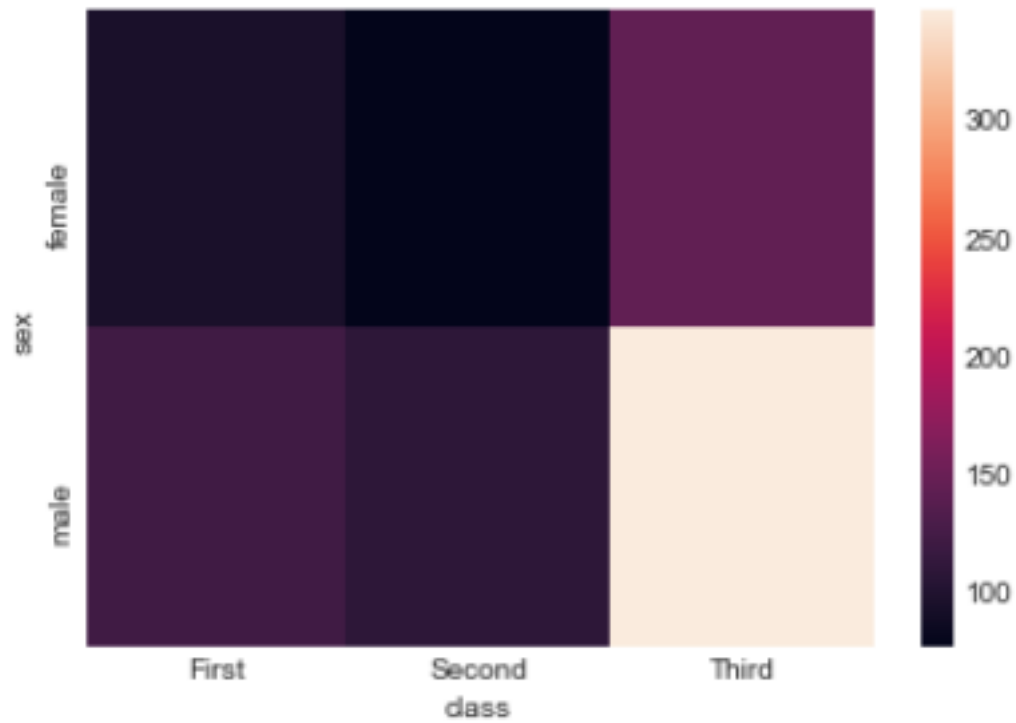
```
# titanic
# 피벗테이블로 범주형 변수를 각각 행, 열로 재구분하여 정리
# 인덱스 성별, 열 클래스

table = titanic.pivot_table(index=['sex'], columns=['class'], aggfunc='size')
table.head()
```

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

```
sns.heatmap(table)
plt.show()
```

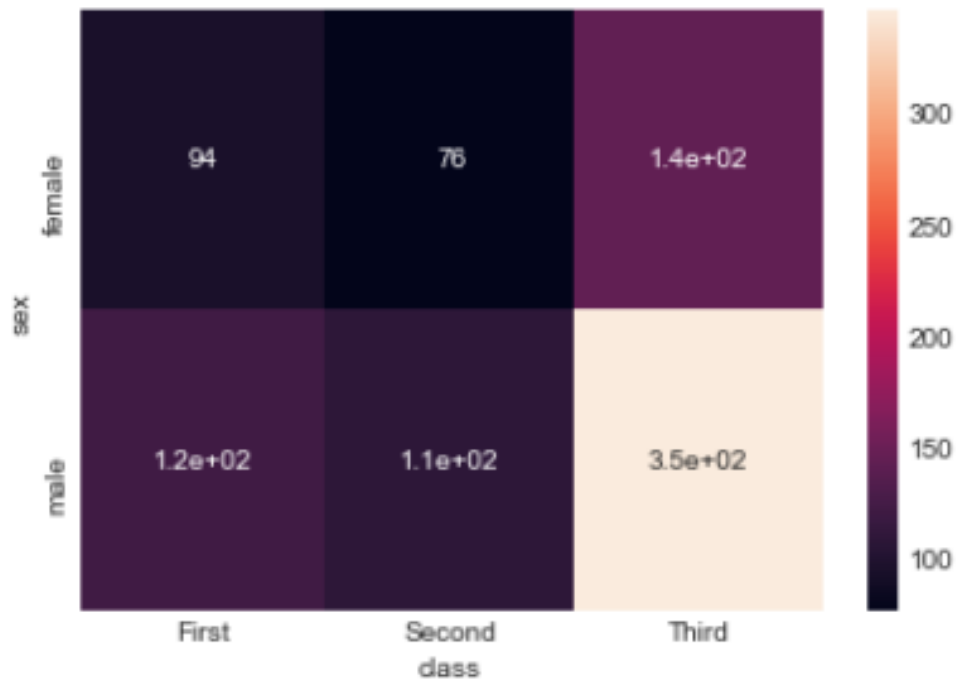
```
sns.heatmap(table)  
plt.show()
```



annot

해당 수의 값을 넣어준다

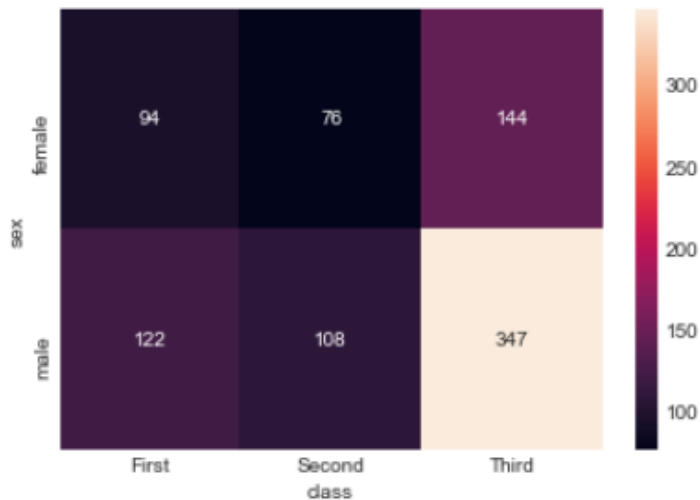
```
sns.heatmap(table,annot=True) # annot값을 넣는다.
plt.show()
```



#fmt

format을 나타낸다. 그래서 정수 값으로 값이 나오게 한다.

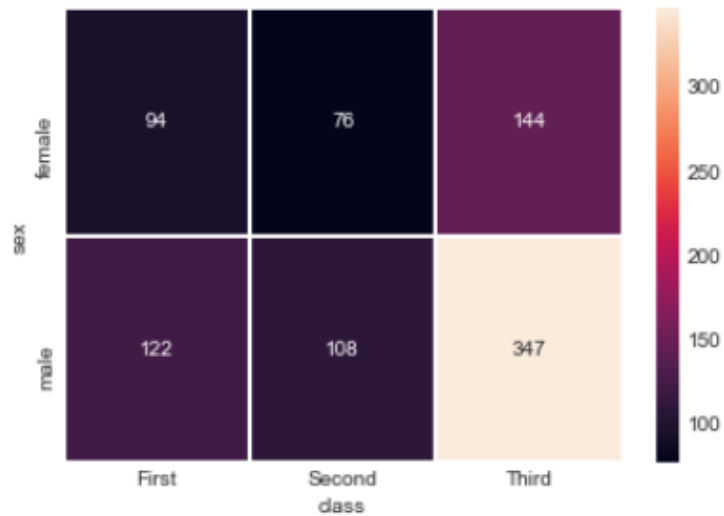
```
sns.heatmap(table,annot=True, fmt='d') # fmt='d' 는 정수값으로 나오게 한다.
plt.show()
```



linewidth

각 칸의 간격을 만든다.

```
sns.heatmap(table,annot=True, fmt='d',linewidth=1) # linewidth 간격을 둔다  
plt.show()
```



예제)

```
: # flights 연도, 월 별 승객수에 대한 heatmap
: # x 연도, y축 월
```

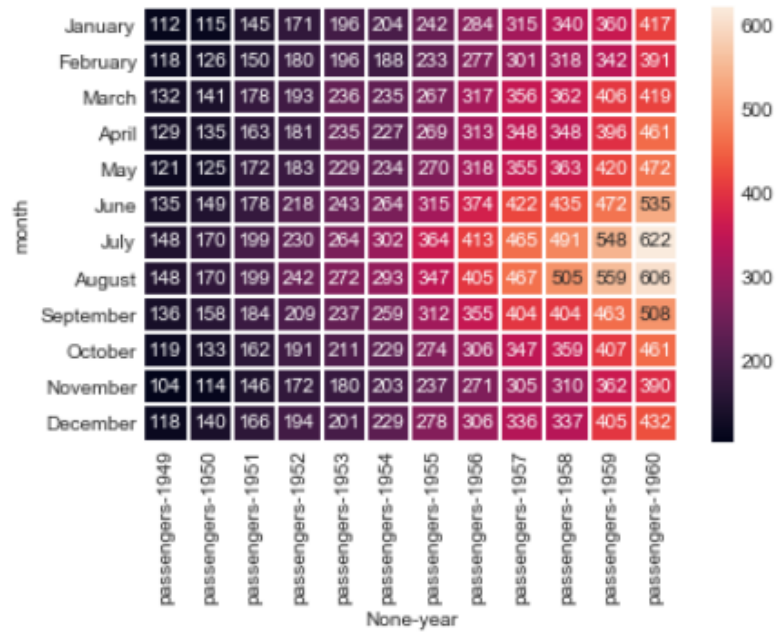
```
: flights.tail()
```

	year	month	passengers
139	1960	August	606
140	1960	September	508
141	1960	October	461
142	1960	November	390
143	1960	December	432

```
table = flights.pivot_table(index=['month'], columns=['year'])
table
```

	passengers											
year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
January	112	115	145	171	196	204	242	284	315	340	360	417
February	118	126	150	180	196	188	233	277	301	318	342	391
March	132	141	178	193	236	235	267	317	356	362	406	419
April	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
June	135	149	178	218	243	264	315	374	422	435	472	535
July	148	170	199	230	264	302	364	413	465	491	548	622
August	148	170	199	242	272	293	347	405	467	505	559	606
September	136	158	184	209	237	259	312	355	404	404	463	508
October	119	133	162	191	211	229	274	306	347	359	407	461
November	104	114	146	172	180	203	237	271	305	310	362	390
December	118	140	166	194	201	229	278	306	336	337	405	432

```
sns.heatmap(table,annot=True, fmt='d',linewidth=1) # linewidth 간격을 둔다
plt.show()
```



'Python' 카테고리의 다른 글

[Python] 시각화 사용법 - seaborn을 통한 그래프 만들기

[Python] 시각화 사용법 - folium 을 통한 지도 시각화 및 마커(marker) 찍기

[Python] 시각화 사용법 - matplotlib을 통한 box plot 그리기

[Python] matplotlib을 통한 bar plot 그리기

[Python] matplotlib 한글 폰트 깨짐 현상 , 마이너스 기호(폰트)가 깨지는 현상 해결방법

[Python] 시각화 사용법 - matplotlib을 통한 line plot 그리기(lim,ticks 등등)

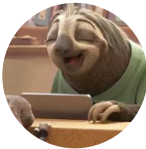
python seaborn

python seaborn 사용법

파이썬 seaborn

파이썬 seaborn 사용법

파이썬 시각화 그래프



나아무늘보

혼자 끄적끄적하는 블로그 입니다.