

# 파이썬 라이브러리를 활용한 데이터 분석

## 9장 그래프와 시각화

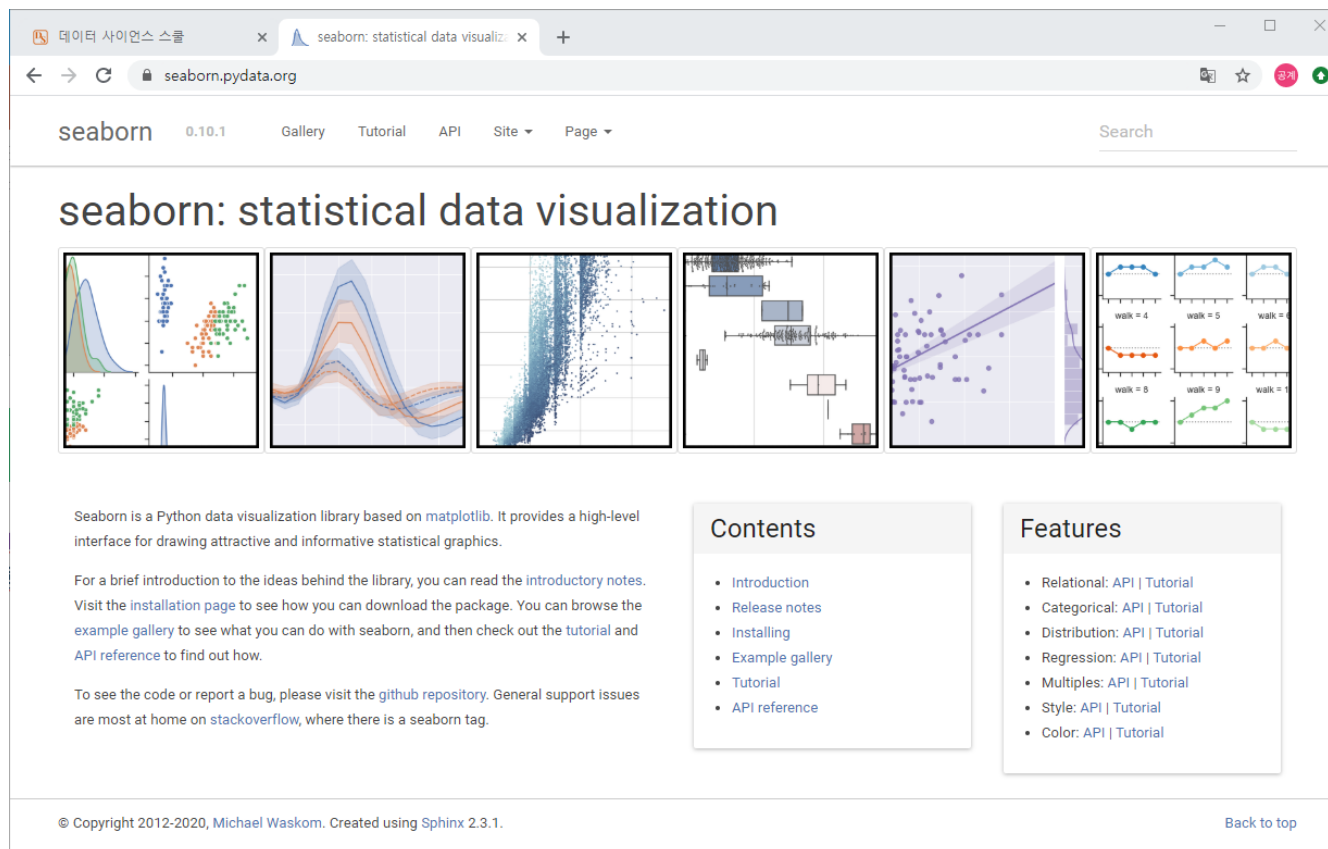
2020.07.03 1h

# 9장 그래프와 시각화

seaborn

# seaborn

- Series와 DataFrame 객체를 시각화하는 통계 그래픽 라이브러리
  - 마이클 와스콤이 개발
    - <https://seaborn.pydata.org/>



# Seaborn 패키지

- Matplotlib을 기반으로 다양한 색상 테마와 통계용 차트 등의 기능을 추가한 시각화 패키지
  - 기본적인 시각화 기능은 Matplotlib 패키지 기반
  - 통계 기능은 Statsmodels 패키지에 의존
    - <http://seaborn.pydata.org/>

# Seaborn 그래프

## • 필요

- total\_bill
  - 팁이 포함된 총액
- tip:
  - 팁 액
- smoker
- day
- time
- size
  - 식사 인원
- tip\_pct
  - 식비에서 팁의 비율

## • 요일 별 팁 비율

- x: 팁 비율
  - 팁 비율의 평균
  - 가운데 직선
  - 95% 신뢰구간
- y: 요일

## • 해석

- 일요일과 금요일에  
팁 비율이 높음

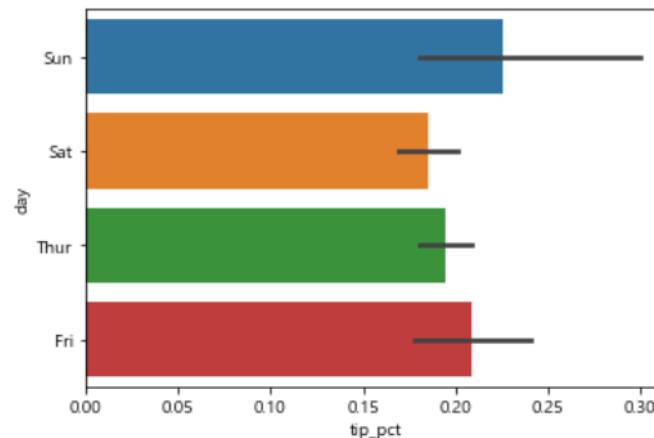
```
In [173]: import seaborn as sns
tips['tip_pct'] = tips['tip'] / (tips['total_bill'] - tips['tip'])
tips.head()
```

```
Out[173]:
```

	total_bill	tip	smoker	day	time	size	tip_pct
0	16.99	1.01	No	Sun	Dinner	2	0.063204
1	10.34	1.66	No	Sun	Dinner	3	0.191244
2	21.01	3.50	No	Sun	Dinner	3	0.199886
3	23.68	3.31	No	Sun	Dinner	2	0.162494
4	24.59	3.61	No	Sun	Dinner	4	0.172069

```
In [174]: sns.barplot(x='tip_pct', y='day', data=tips, orient='h')
```

```
Out[174]: <matplotlib.axes._subplots.AxesSubplot at 0x23c198dbd88>
```



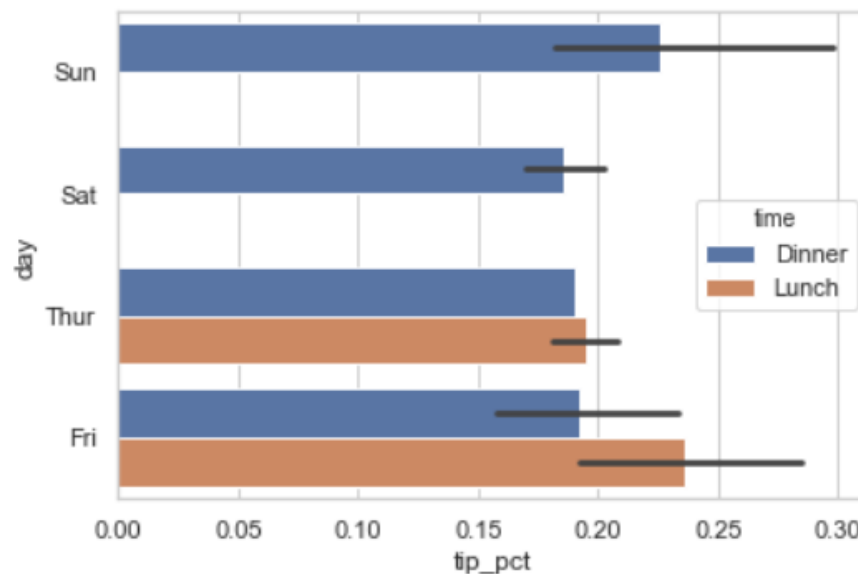
# 옵션 hue=

- 내부 구분
  - hue='time'

	total_bill	tip	smoker	day	time	size	tip_pct
0	16.99	1.01	No	Sun	Dinner	2	0.063204
1	10.34	1.66	No	Sun	Dinner	3	0.191244
2	21.01	3.50	No	Sun	Dinner	3	0.199886
3	23.68	3.31	No	Sun	Dinner	2	0.162494
4	24.59	3.61	No	Sun	Dinner	4	0.172069

```
In [181]: sns.set(style="whitegrid")
          sns.barplot(x='tip_pct', y='day', hue='time', data=tips, orient='h')
```

```
Out[181]: <matplotlib.axes._subplots.AxesSubplot at 0x23c19b06788>
```



# Seaborn의 배경 설정

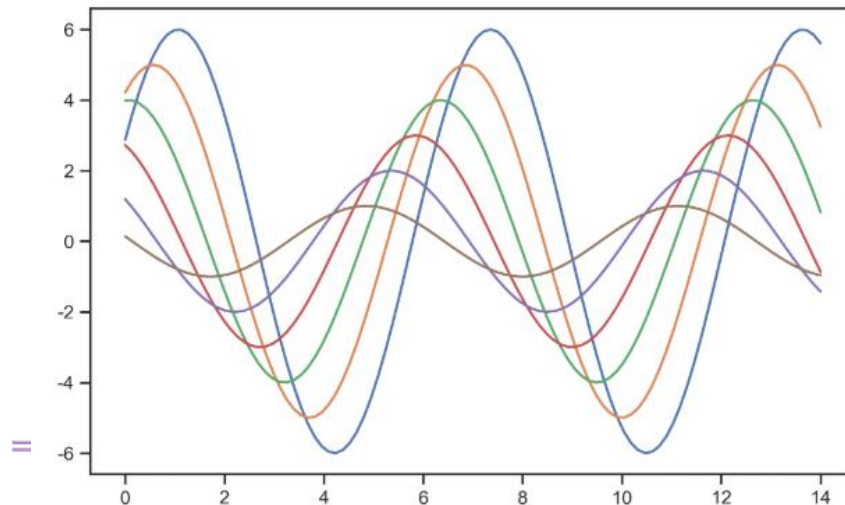
- **set 명령**
  - 색상, 틱 스타일 등 전반적인 플롯 스타일을 Seaborn 기본 스타일로 수정
- **set\_style 명령**
  - 틱 스타일만 수정
    - **darkgrid, whitegrid, dark, white, 그리고 ticks** 스타일을 제공

In [30]:

```
def sinplot(flip=1):
    x = np.linspace(0, 14, 100)
    for i in range(1, 7):
        plt.plot(x, np.sin(x + i * .5) * (7 - i) * flip)
```

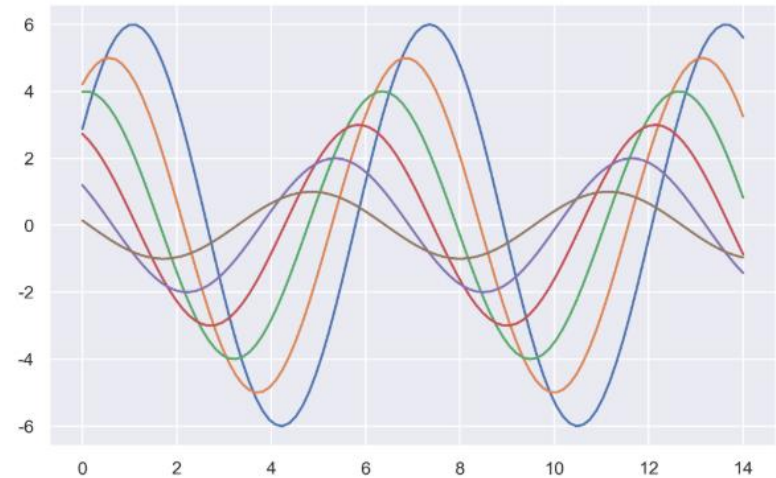
In [31]:

```
sns.set_style("ticks")
sinplot()
```



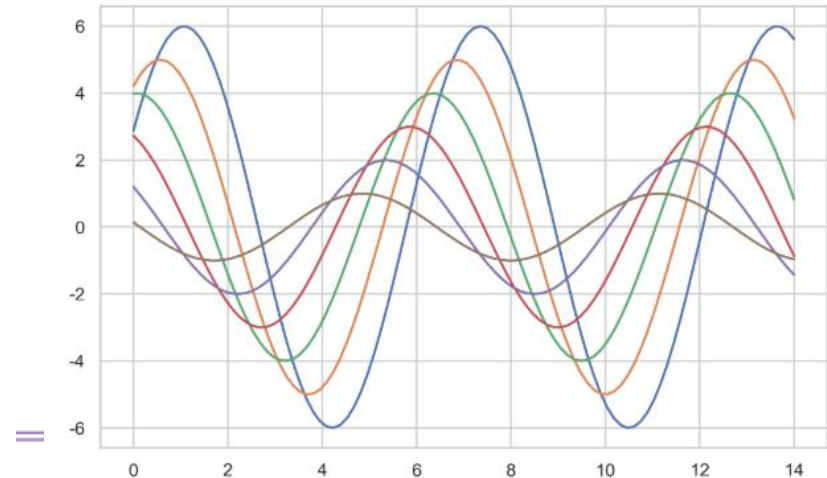
In [32]:

```
sns.set_style("darkgrid")
sinplot()
```



In [33]:

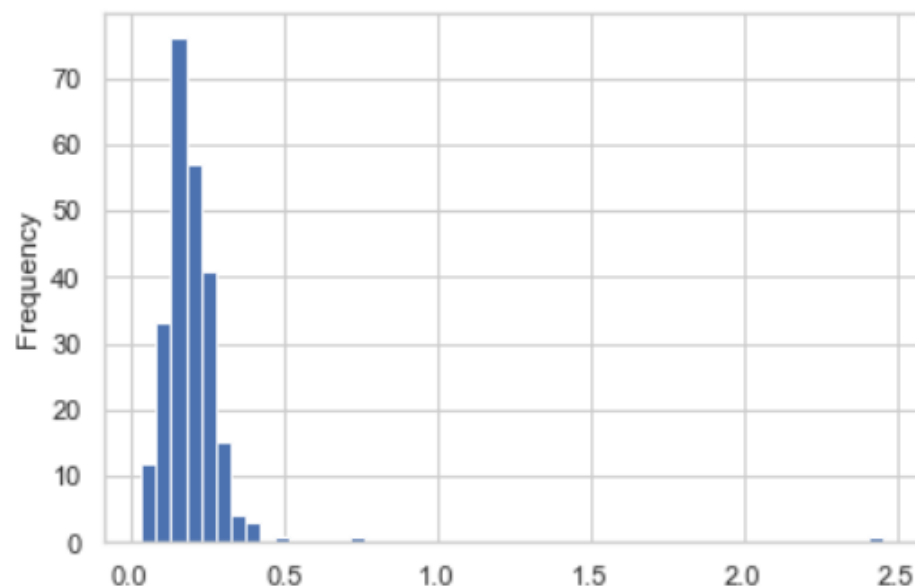
```
sns.set_style("whitegrid")
sinplot()
```



# 팁 비율 히스토그램

```
In [188]: tips['tip_pct'].plot.hist(bins=50)
```

```
Out[188]: <matplotlib.axes._subplots.AxesSubplot at 0x23c1ad14148>
```



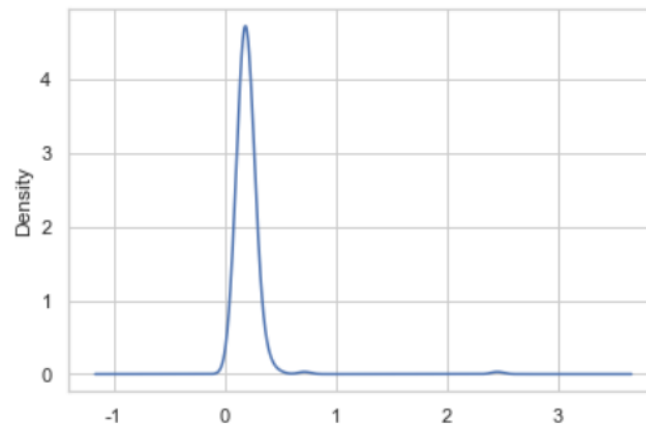


# 밀도 그래프

- KDE(kernel density estimate) 그래프
  - 데이터를 사용해 추정되는 연속된 확률 분포를 그림
- Seaborn의 distplot
  - 히스토그램과 밀도 그래프를 한번에 그려줌

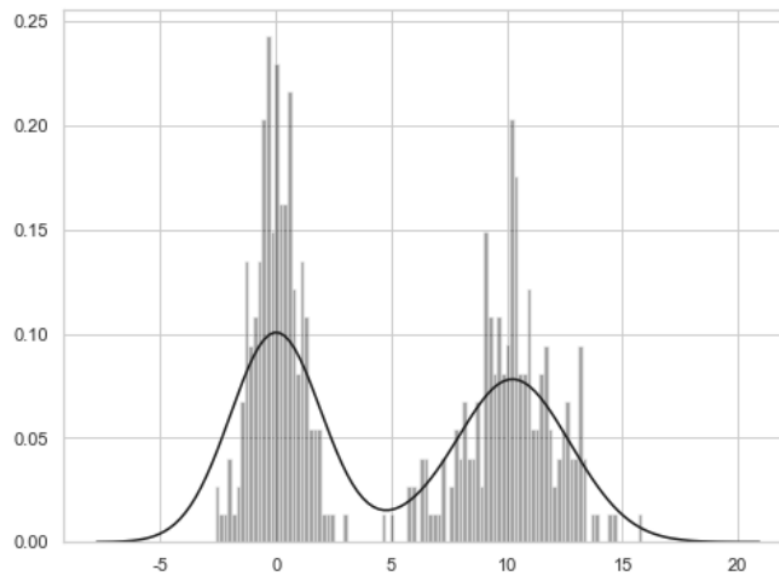
```
In [213]: tips['tip_pct'].plot.density()
```

```
Out[213]: <matplotlib.axes._subplots.AxesSubplot at 0x23c1b7e4208>
```



```
In [218]: comp1 = np.random.normal(0, 1, size=200)
comp2 = np.random.normal(10, 2, size=200)
values = pd.Series(np.concatenate([comp1, comp2]))
sns.distplot(values, bins=100, color='k')
```

```
Out[218]: <matplotlib.axes._subplots.AxesSubplot at 0x23c1cb0b9c8>
```

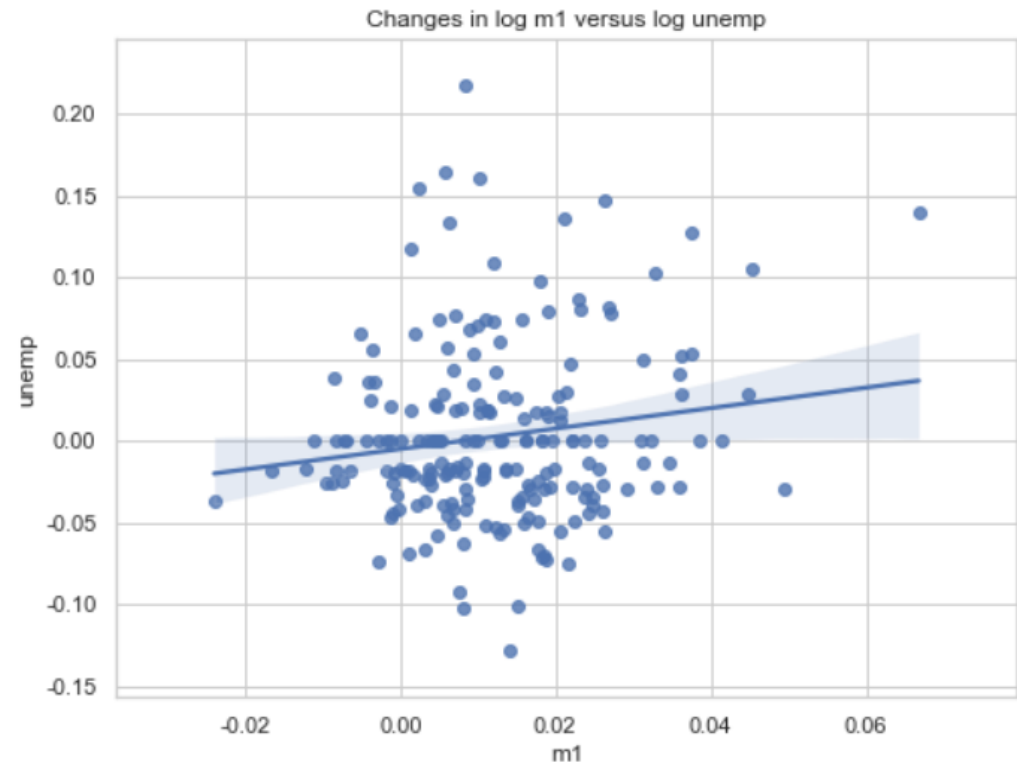


# 산포도

- 두 개의 1차원 묶음 간의 관계를 표시
- `sns.regplot()`
  - 산포도와 선형회귀선
    - X로 추정되는  
y 값을 이은 선

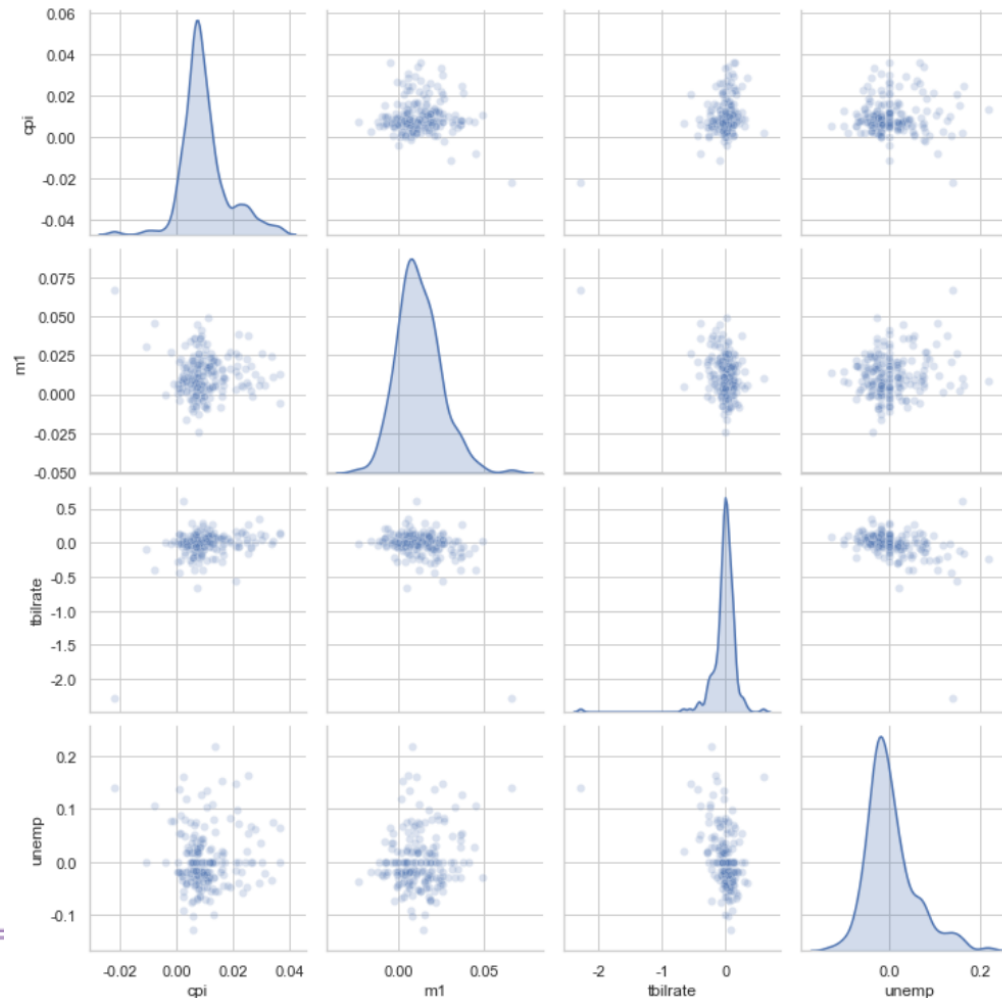
```
In [221]: sns.regplot('m1', 'unemp', data=trans_data)
           plt.title('Changes in log %s versus log %s' % ('m1', 'unemp'))

Out[221]: Text(0.5, 1.0, 'Changes in log m1 versus log unemp')
```



# 산포도 행렬

- `sns.pairplot(trans_data, diag_kind='kde', plot_kws={'alpha': 0.2})`
  - 변수 그룹 간의 모든 산포도를 그림

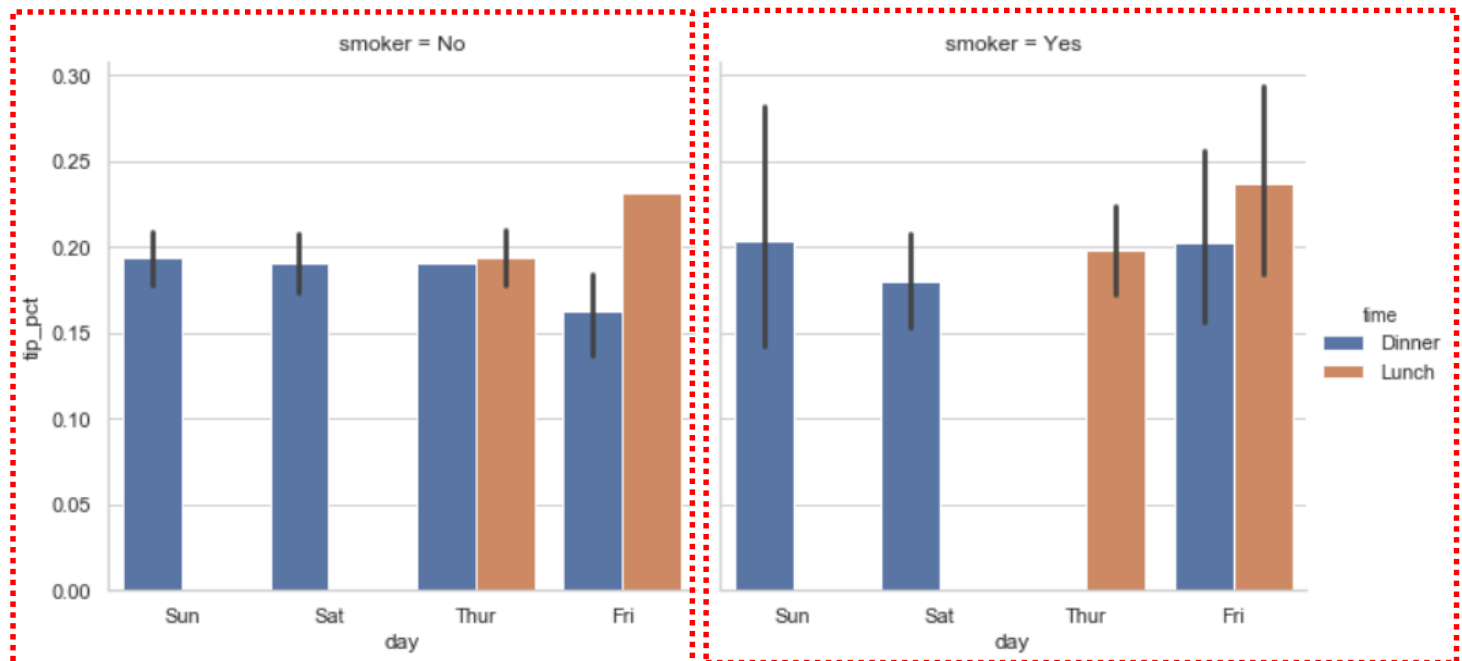


# 여러 그림을 표현하는 Facet grid

- 다양한 범주형 값을 가지는 데이터 시각화 방법
  - Seaborn의 catplot()
    - 색상(hue)과 행(row) 등을 동시에 사용하여 3 개 이상의 카테고리 값에 의한 분포 변화를 보여 줌

```
In [225]: #factorplot
sns.catplot(x='day', y='tip_pct', hue='time', col='smoker',
            kind='bar', data=tips[tips.tip_pct < 1])
```

```
Out[225]: <seaborn.axisgrid.FacetGrid at 0x23c1daa3308>
```



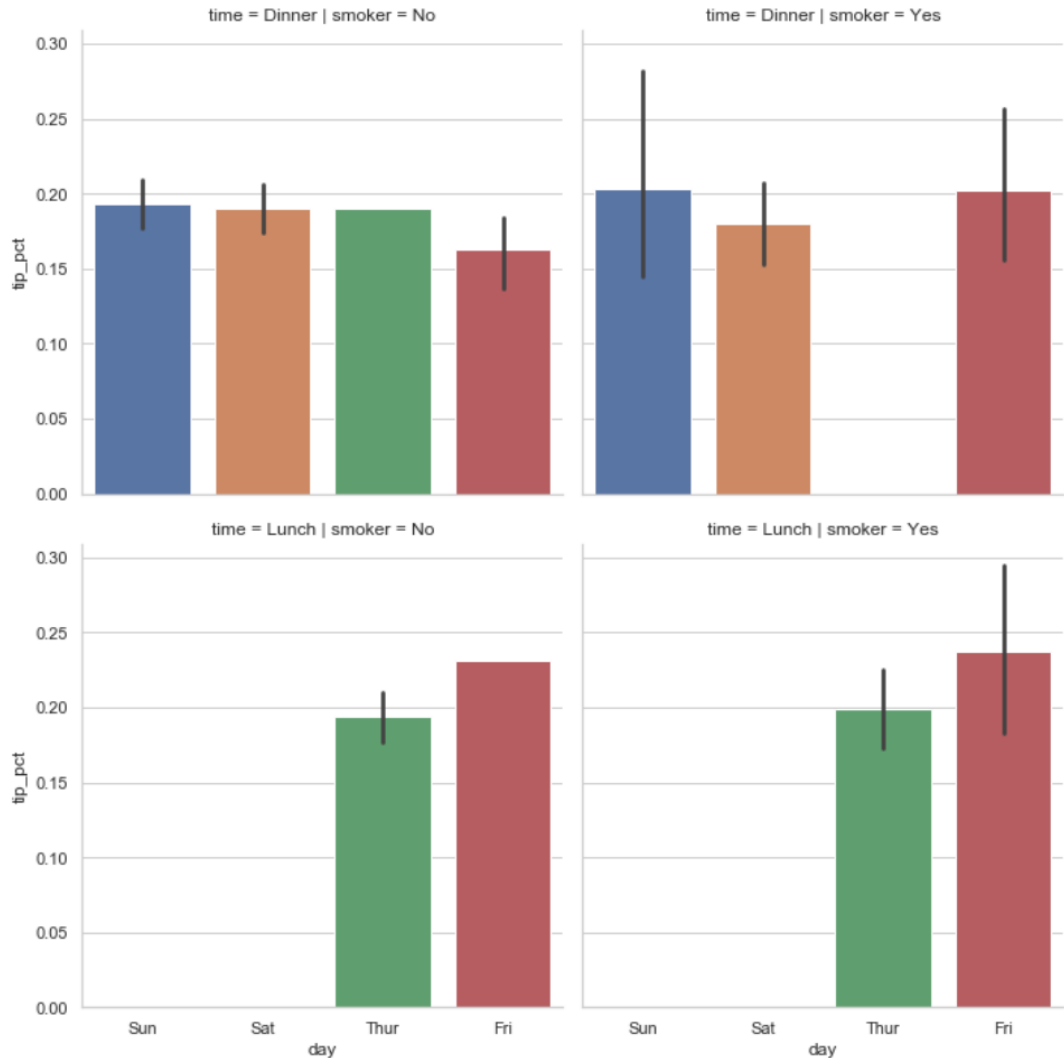
## 칼럼을 더 추가

```
In [232]: sns.catplot(x='day', y='tip_pct', row='time', col='smoker',
                      kind='bar', data=tips[tips.tip_pct < 1])
```

```
Out[232]: <seaborn.axisgrid.FacetGrid at 0x23c1f605d48>
```

- 옵션 row='time'

	total_bill	tip	smoker	day	time	size	tip_pct
0	16.99	1.01	No	Sun	Dinner	2	0.063204
1	10.34	1.66	No	Sun	Dinner	3	0.191244
2	21.01	3.50	No	Sun	Dinner	3	0.199886
3	23.68	3.31	No	Sun	Dinner	2	0.162494
4	24.59	3.61	No	Sun	Dinner	4	0.172069

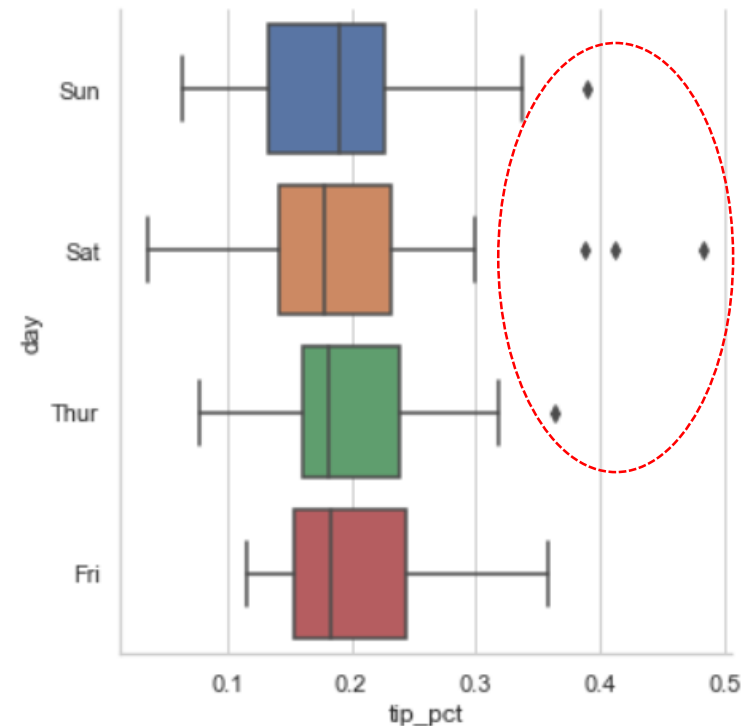


# kind=box

- 중간값, 사분위, 특잇값 표시
  - 박스 내부의 가로선은 중앙값
    - 박스는 실수 값 분포에서 1사분위수(Q1)와 3사분위수(Q3)를 뜻하고
    - 3사분위수와 1사분위의 차이 (Q3 - Q1)를 IQR(interquartile range)
  - 특잇값
    - 아웃라이어(outlier), 점으로 표시

```
In [233]: sns.catplot(x='tip_pct', y='day', kind='box',
                    data=tips[tips.tip_pct < 0.5])
```

```
Out[233]: <seaborn.axisgrid.FacetGrid at 0x23c1fb90588>
```



# 웹을 위한 대화형 그래픽 도구

- 2010년부터 개발
- Bokeh
  - <https://bokeh.org/>
- Plotly
  - <https://plotly.com/>

# 보케 예제

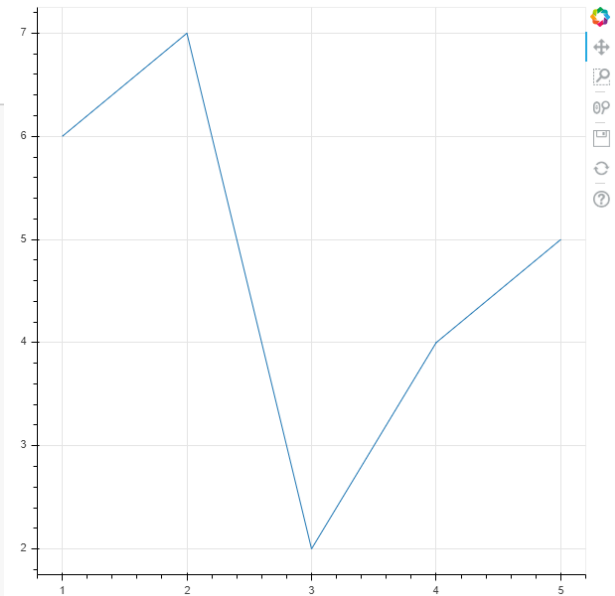
```
In [234]: from bokeh.plotting import figure, show

# from bokeh.plotting import output_file
# 출력파일 지정, 없으면 소스파일과 동일한 html 파일로 생성
# output_file("line.html")

p = figure()
# p = figure(plot_width=400, plot_height=400)

# add a line renderer
p.line([1, 2, 3, 4, 5], [6, 7, 2, 4, 5])
# p.line([1, 2, 3, 4, 5], [6, 7, 2, 4, 5], line_width=2)

show(p)
```



```
In [235]: from bokeh.plotting import figure, show, output_file

p = figure(plot_width=400, plot_height=400)
p.quad(top=[2, 3, 4], bottom=[1, 2, 3], left=[1, 2, 3],
        right=[1.2, 2.5, 3.7], color="#B3DE69")

show(p)
output_file('rectangles.html')
```

