Seaborn을 사용한 데이터 분포 시각화

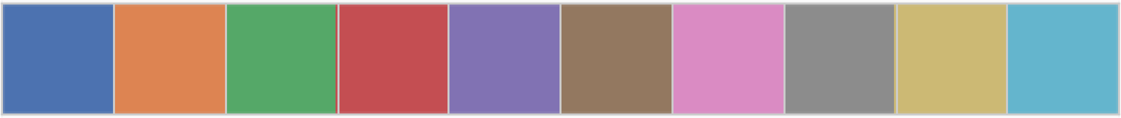
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

from matplotlib import font_manager, rc
```

```
font_name = font_manager.FontProperties(fname="C:/Windows/Fonts/MALGUN.TTF").get_name()
```

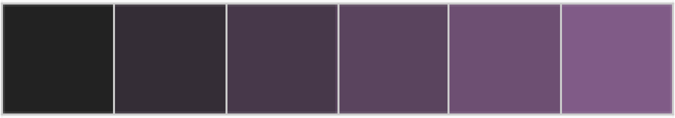
```
rc('font', family=font_name)

plt.rcParams['figure.figsize'] = (10,6)
%matplotlib inline
```

```
import seaborn as sns

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

```
# seaborn의 palette 기능

current_palette = sns.color_palette()
sns.palplot(current_palette)
```



```
sns.palplot(sns.dark_palette("muted purple", input='xkcd'))
```



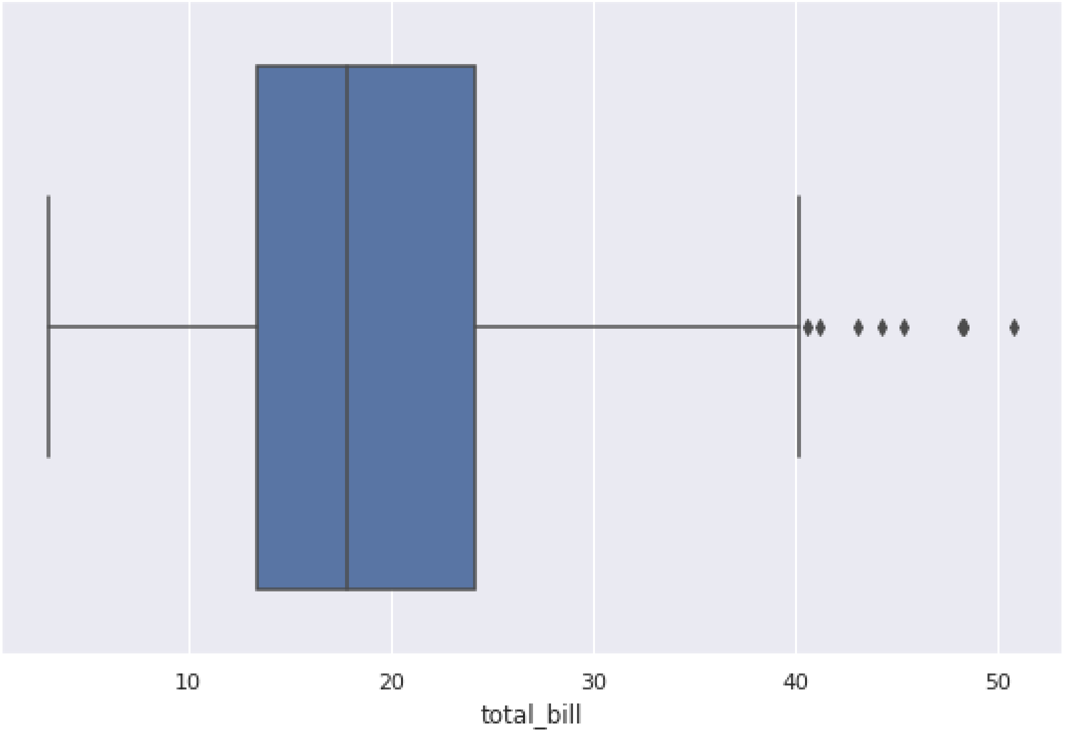
boxplot

```
plt.rcParams['figure.figsize'] = (9.5,6)

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

```
tips = sns.load_dataset('tips')
tips.head(5)
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

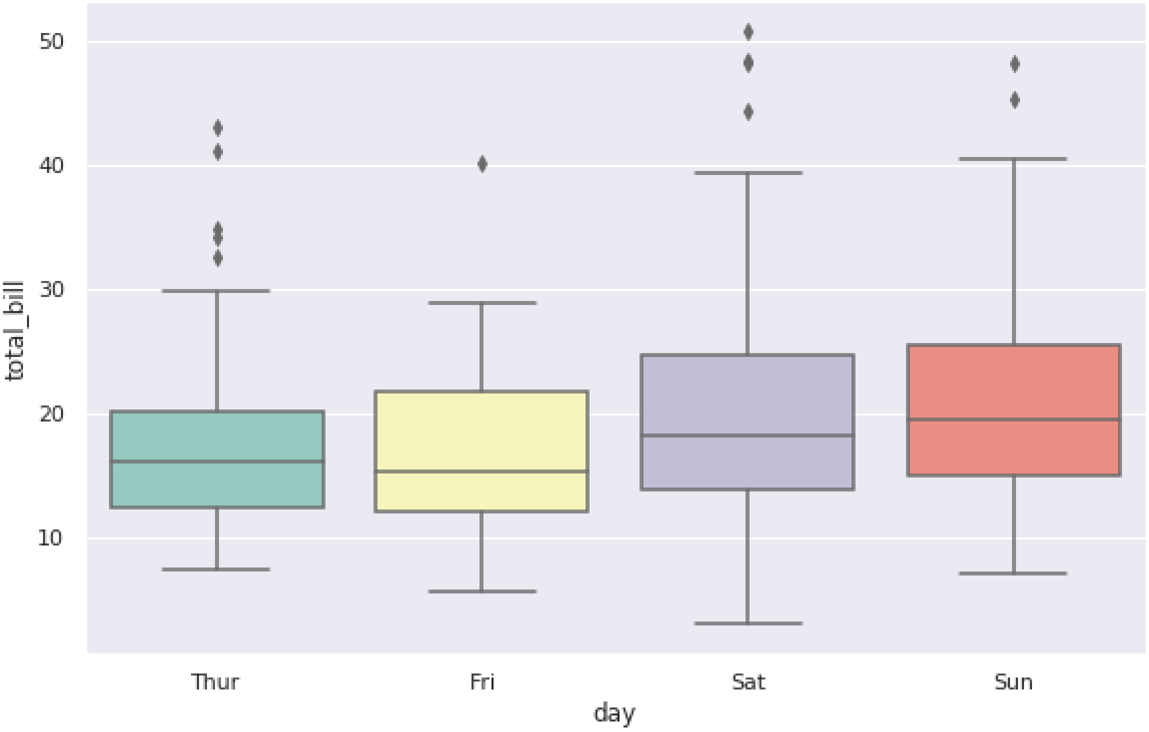
```
sns.boxplot(x=tips['total_bill'])
plt.show()
```



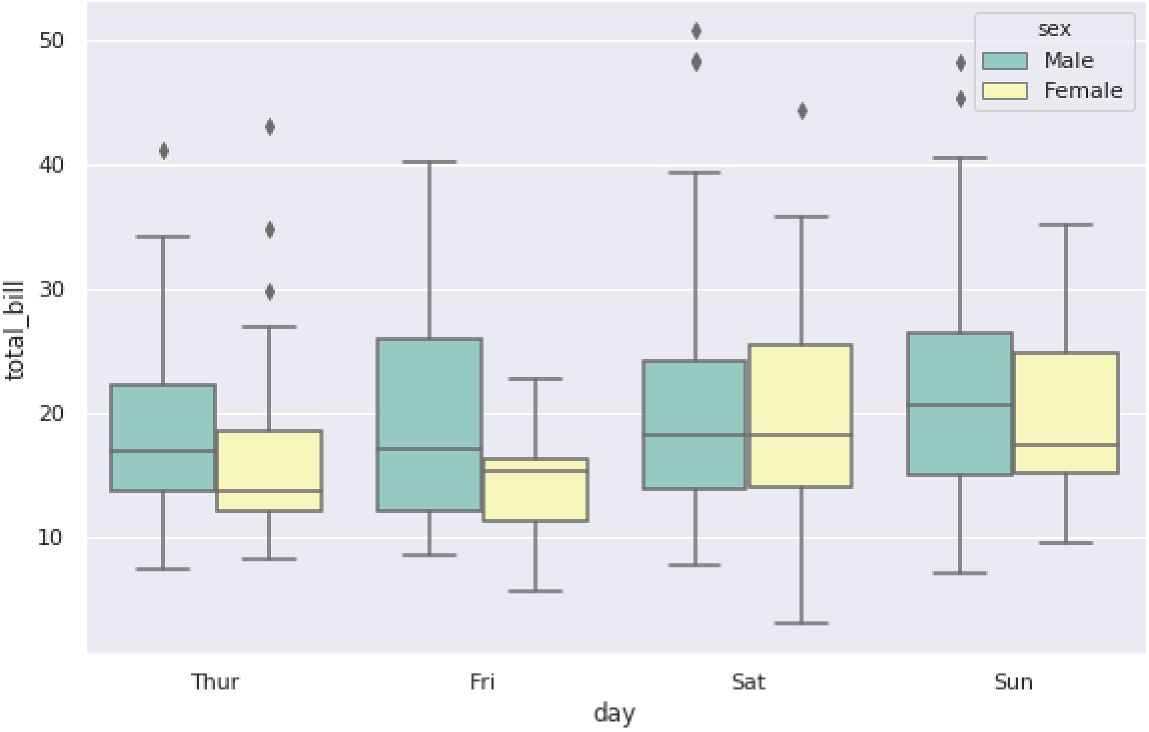
```
tips.day.unique()
```

```
[Sun, Sat, Thur, Fri]
Categories (4, object): [Sun, Sat, Thur, Fri]
```

```
sns.boxplot(x='day', y ='total_bill', data= tips, palette = 'Set3')
plt.show()
```



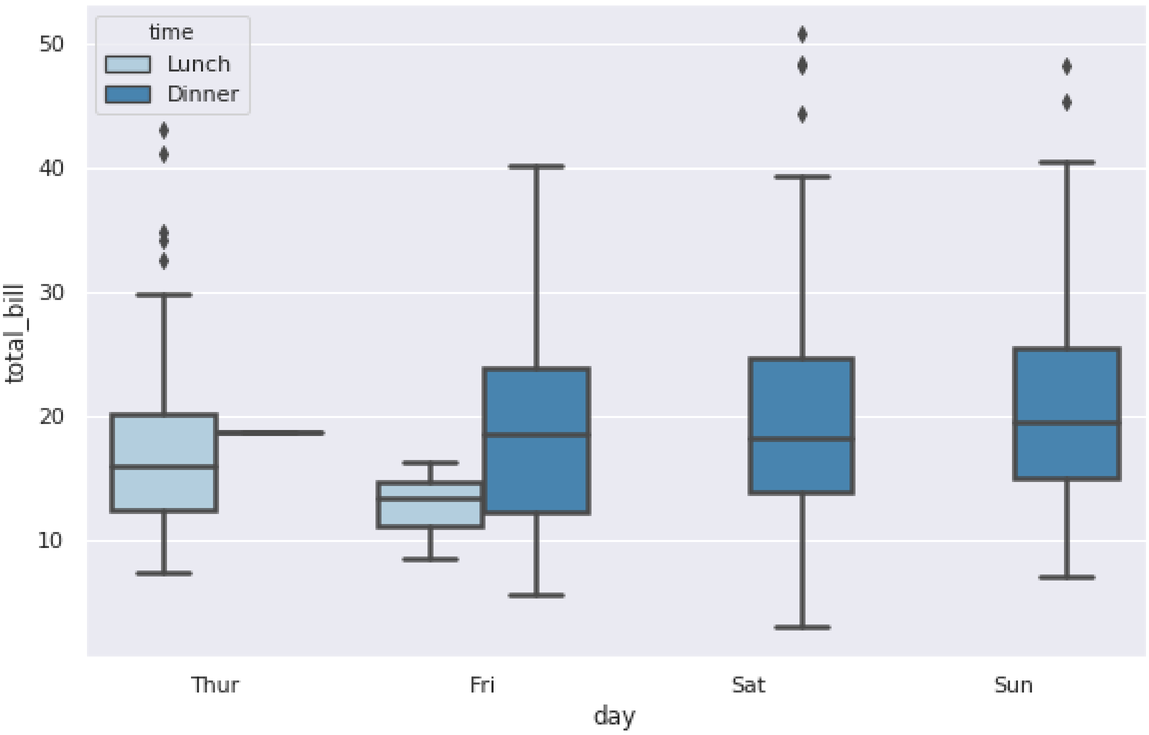
```
sns.boxplot(x='day',y='total_bill',hue='sex',data=tips,
           palette='Set3')    # hue = 카테고리 값을 가지는 변수의 이름을 지정
plt.show()
```



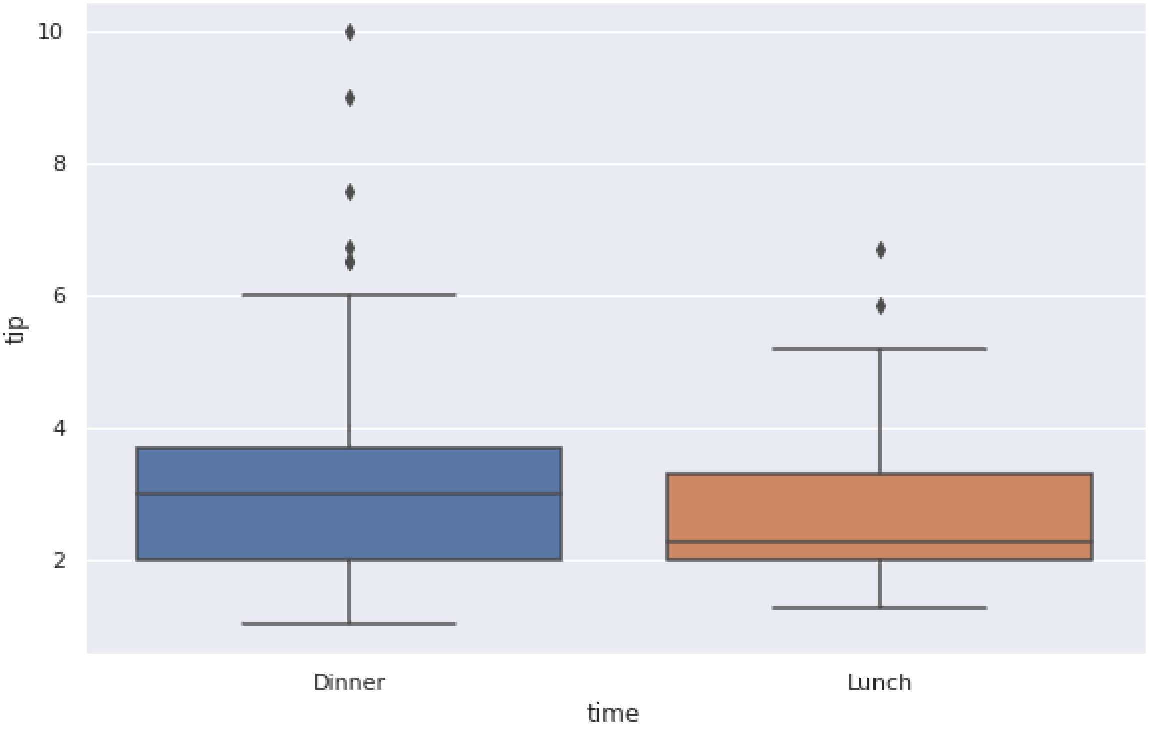
```
tips.head()
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

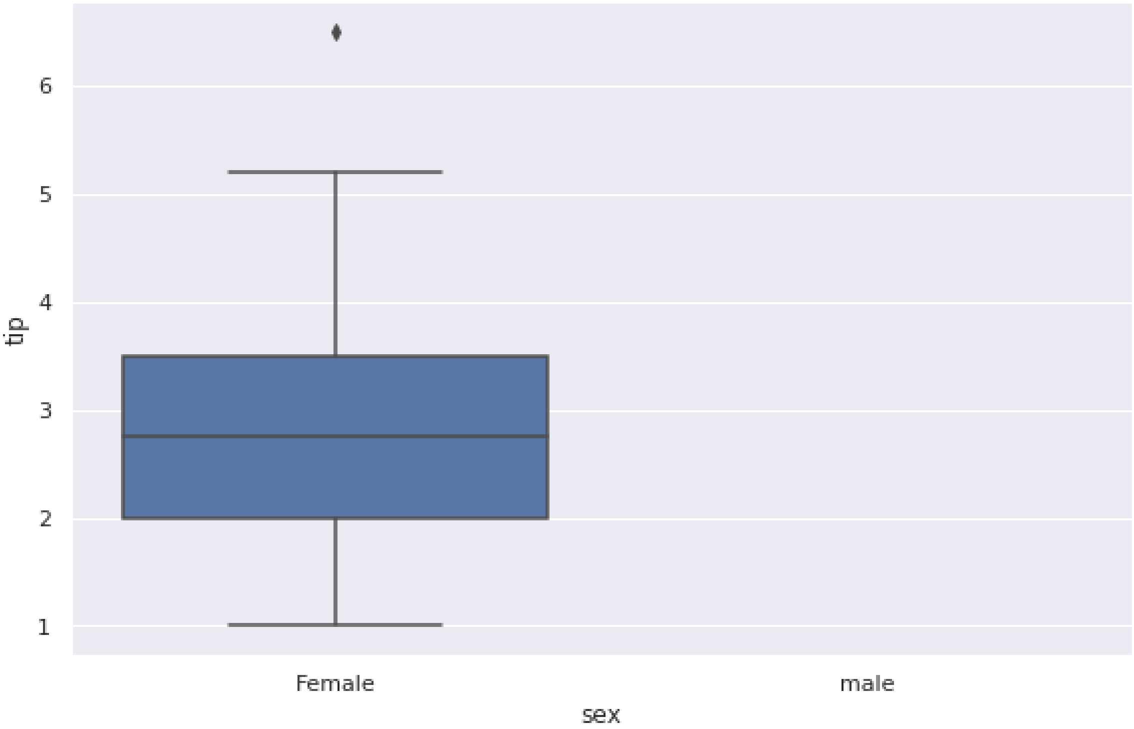
```
sns.boxplot(x='day',y='total_bill',hue='time',data=tips,
            palette='Blues', linewidth=2.5)
plt.show()
```



```
sns.boxplot(x="time", y="tip", data=tips, order=["Dinner", "Lunch"])
plt.show()
```



```
sns.boxplot(x="sex", y="tip", data=tips, order=["Female","male"])
plt.show()
```

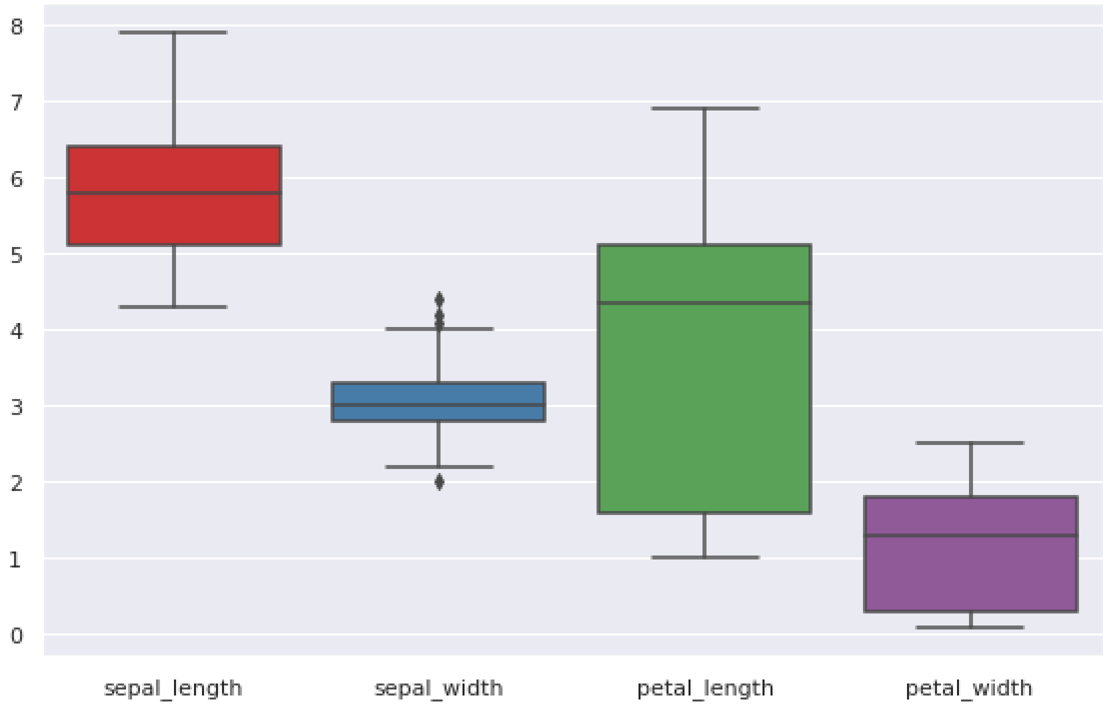


iris dataset

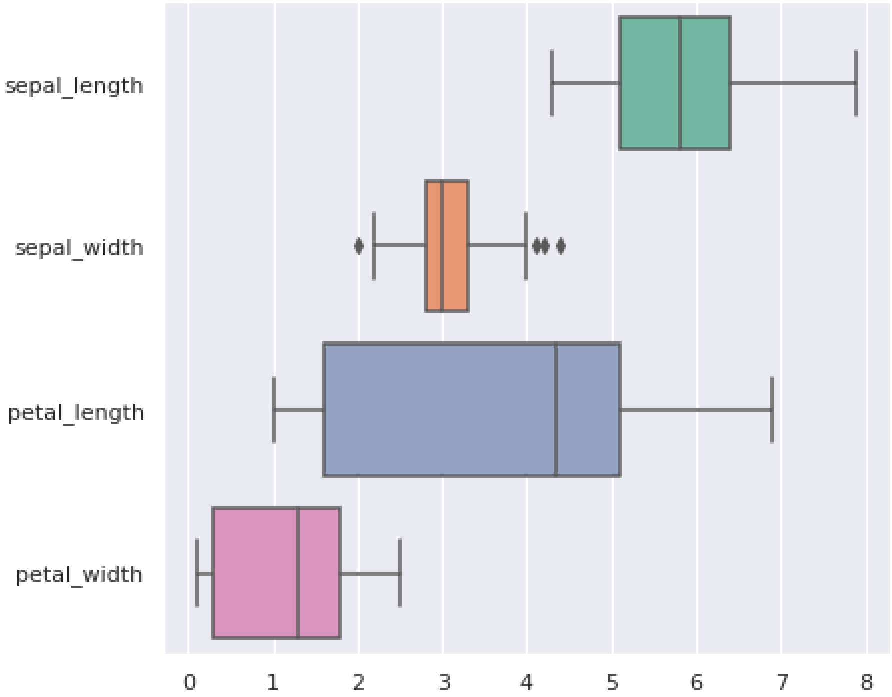
```
iris = sns.load_dataset("iris")
iris.head(5)
```

	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

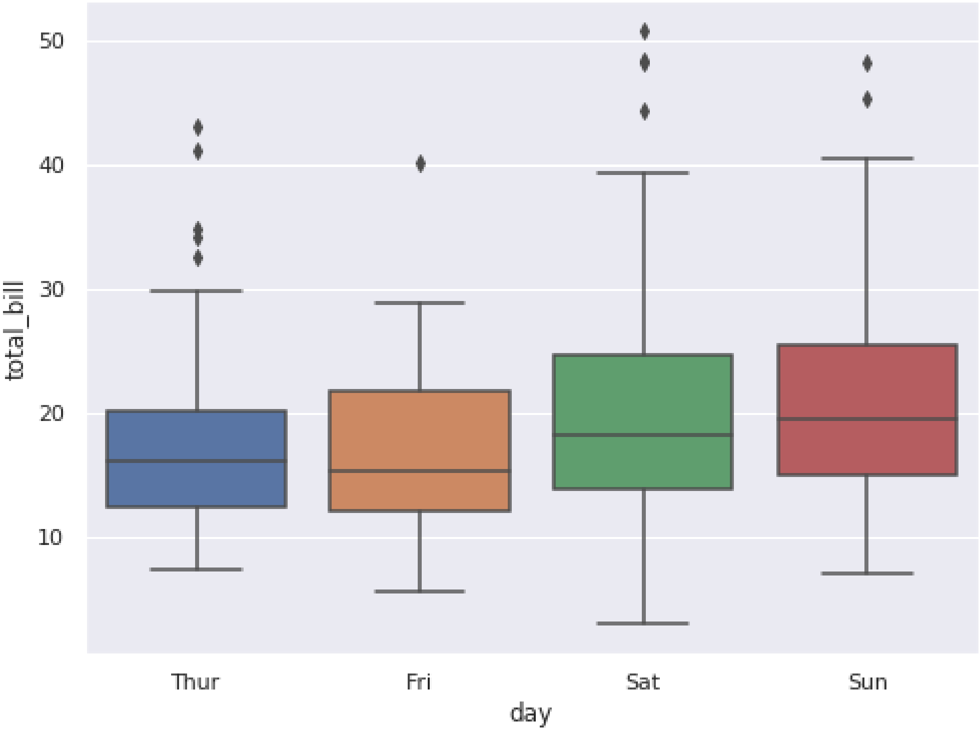
```
sns.boxplot(data=iris, palette="Set1")
plt.show()
```



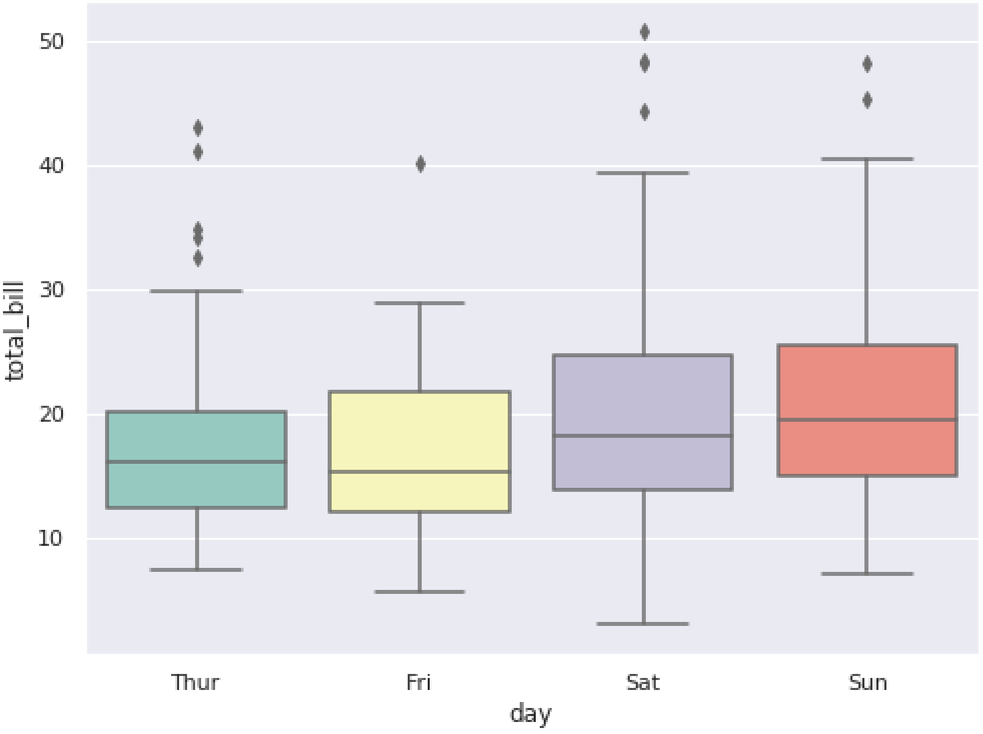
```
plt.figure(figsize=(6.5,6))
sns.boxplot(data=iris, orient='h', palette="Set2")
# orient = 'h'    :: 눕히기
plt.show()
```



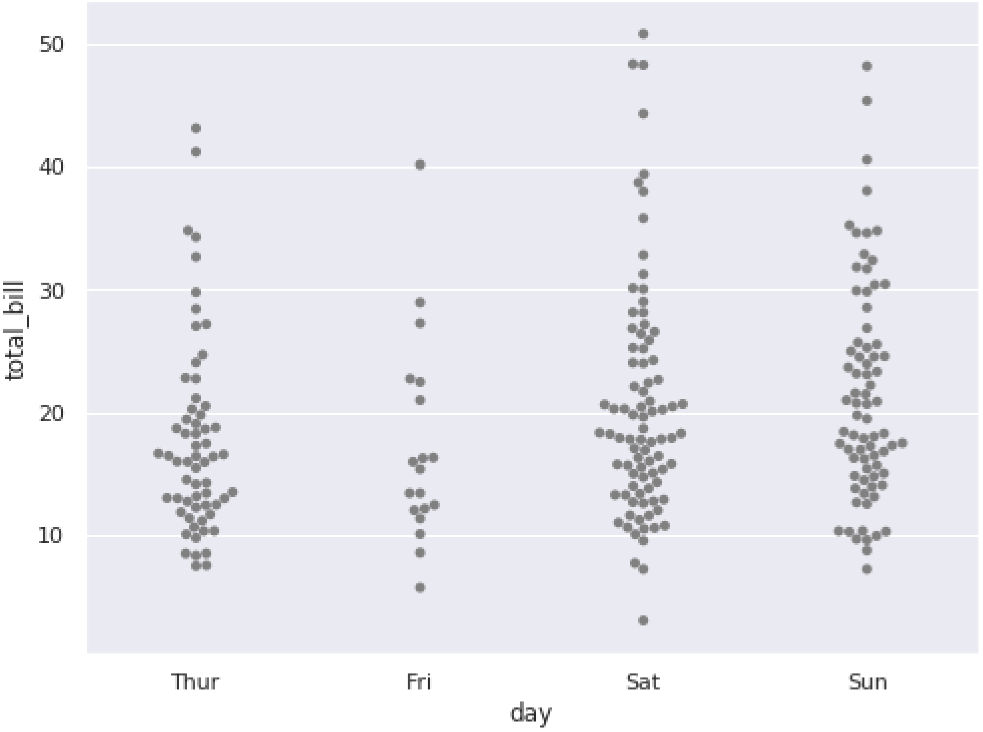
```
plt.figure(figsize=(8,6))
sns.boxplot(x="day", y="total_bill", data=tips)
plt.show()
```



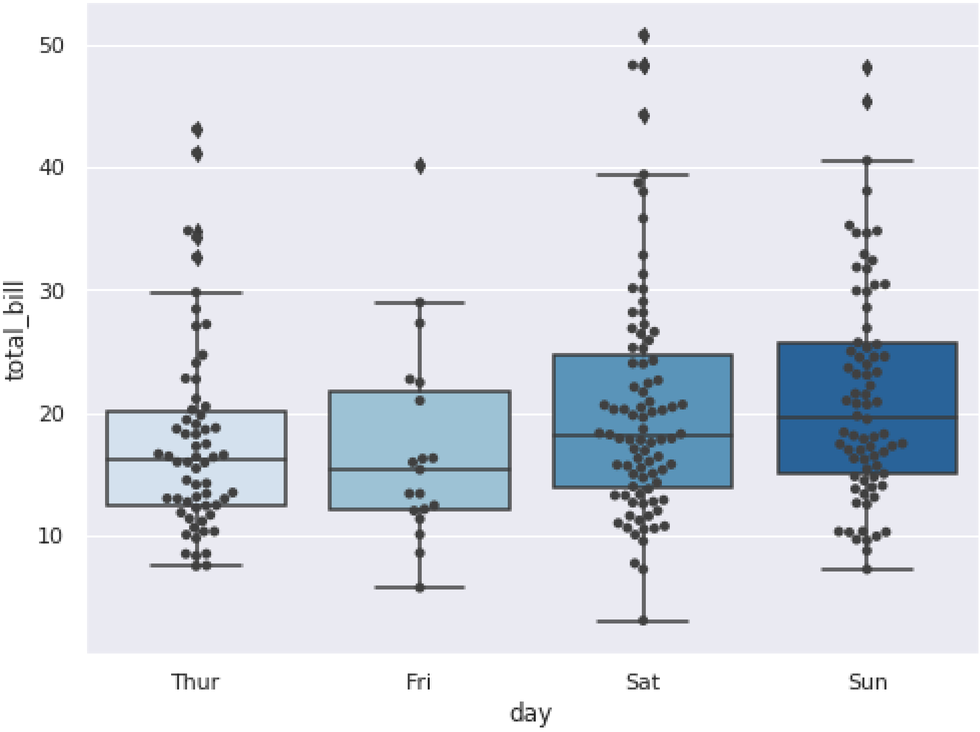
```
plt.figure(figsize=(8,6))
sns.boxplot(x="day", y="total_bill", data=tips,
           palette='Set3')
plt.show()
```



```
plt.figure(figsize=(8,6))
sns.swarmplot(x='day',y='total_bill',data=tips,color='0.5')
plt.show()
```

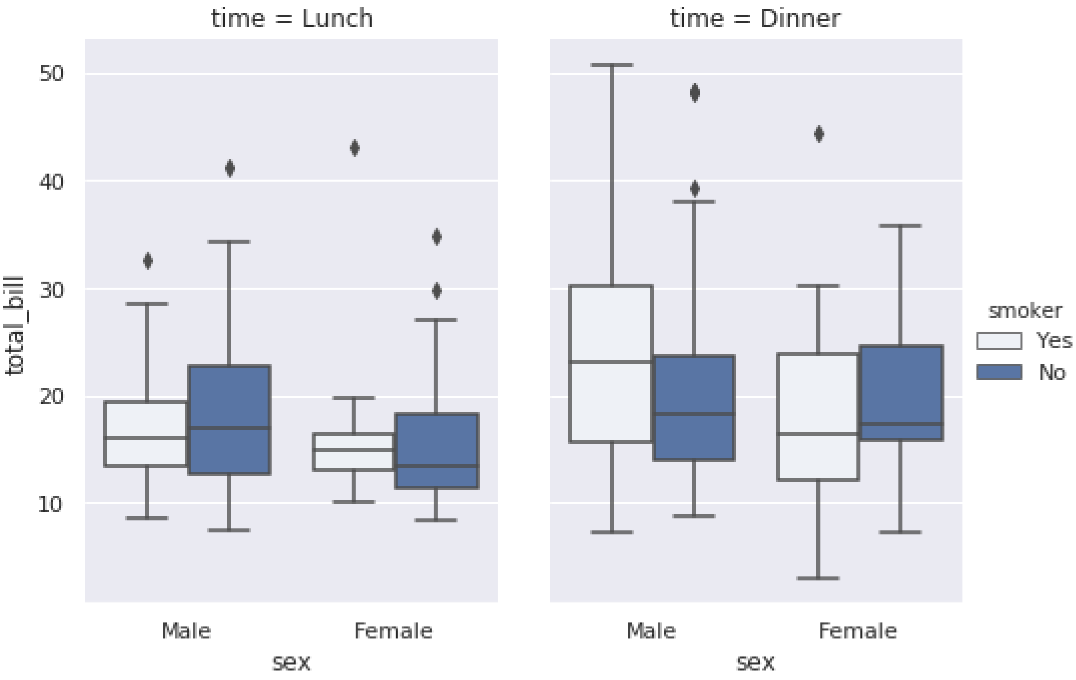


```
plt.figure(figsize=(8,6))
sns.swarmplot(x='day',y='total_bill',data=tips,color='0.25')
sns.boxplot(x='day',y='total_bill',data=tips,palette='Blues')
plt.show()
```



```
g = sns.FacetGrid(tips, col="time", size=5, aspect=.7)  #aspect : 그래프 폭
(g.map(sns.boxplot, 'sex','total_bill','smoker').despine(left=True).add_legend(
title='smoker'))
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/axisgrid.py:230: UserWarning: The `size` paramter has b
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.6/dist-packages/seaborn/axisgrid.py:715: UserWarning: Using the boxplot functio
  warnings.warn(warning)
/usr/local/lib/python3.6/dist-packages/seaborn/axisgrid.py:720: UserWarning: Using the boxplot functio
  warnings.warn(warning)
```



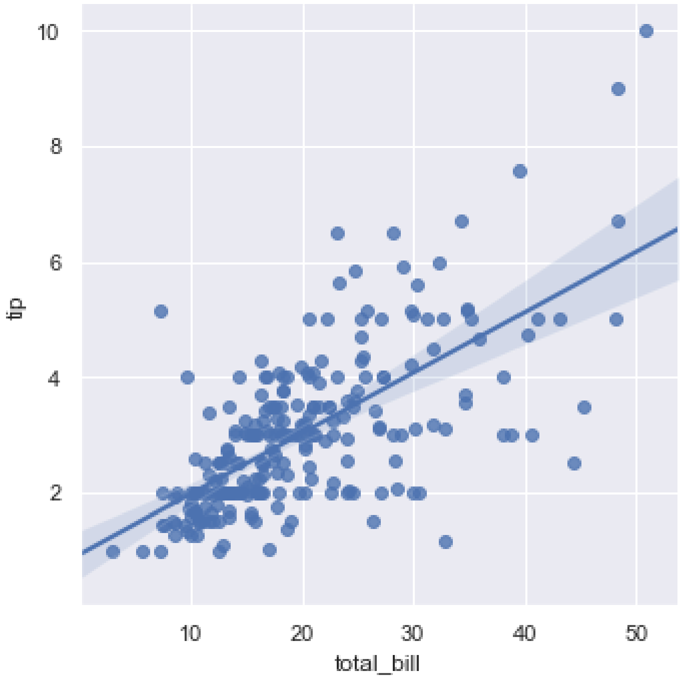
lmplot (대문자 i아님)

- 변수 간 상관관계를 표시할 때 자주 사용

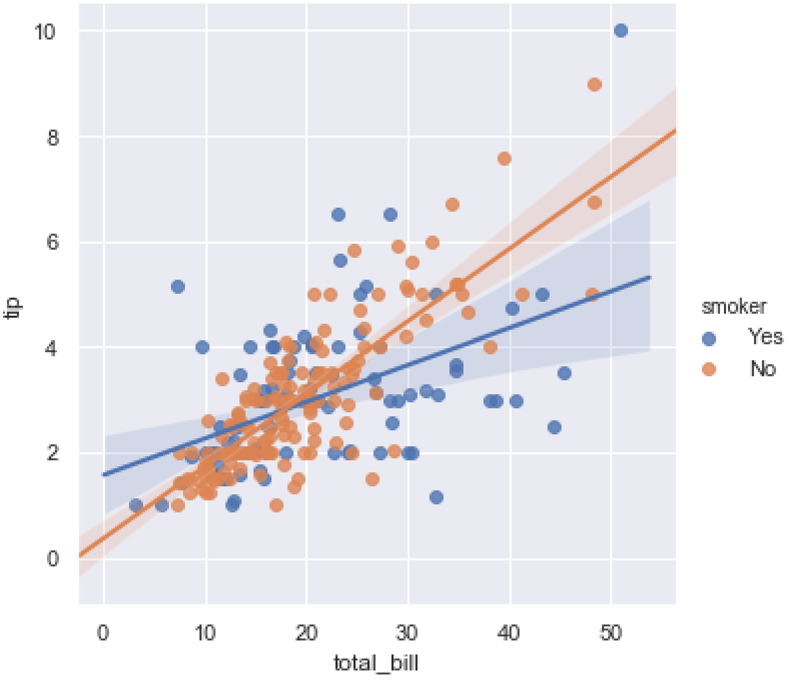
```
sns.set_style('whitegrid')
tips = sns.load_dataset('tips')
tips.head(5)
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

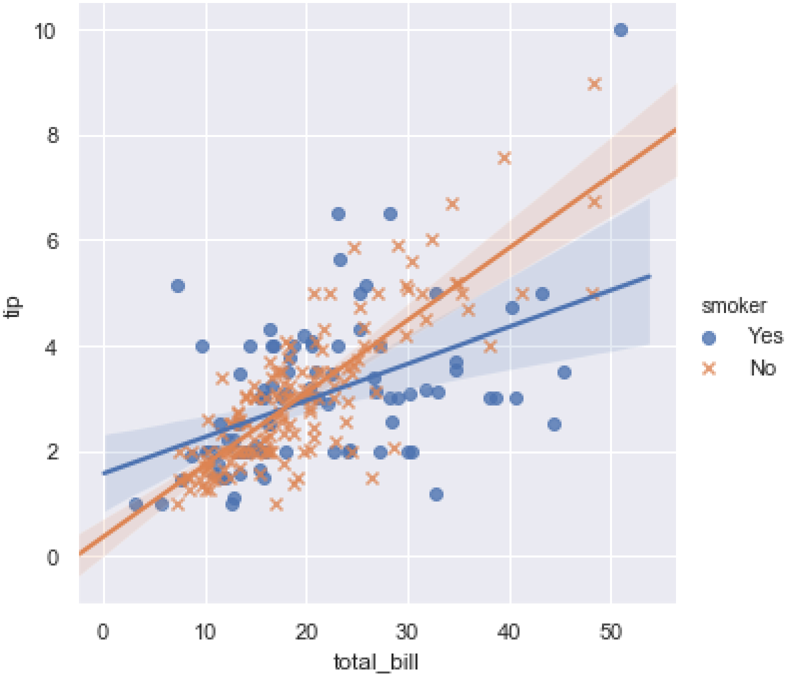
```
sns.set(color_codes=True)
sns.lmplot(x='total_bill',y='tip',data=tips)
plt.show()
```



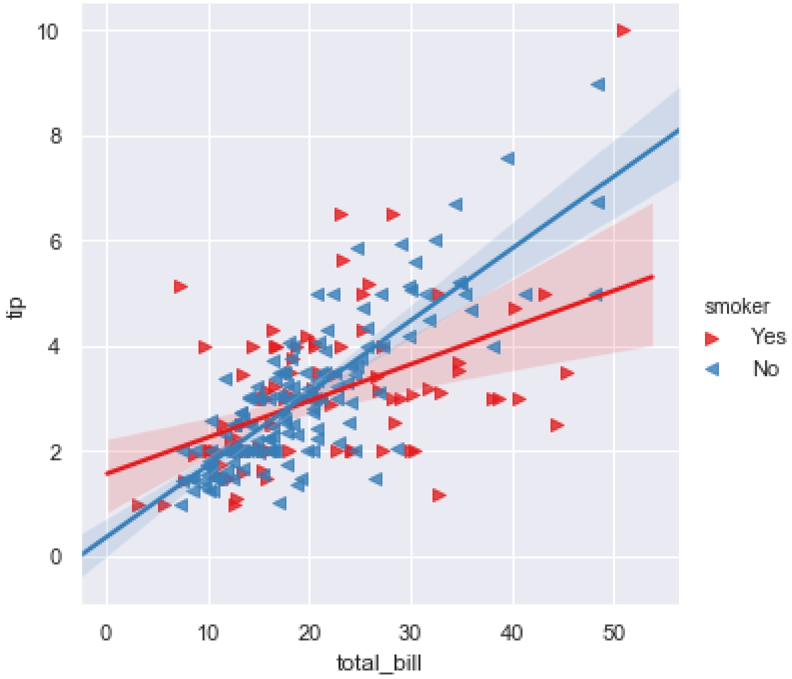
```
sns.lmplot(x="total_bill", y="tip", hue='smoker',data=tips)
plt.show()
```



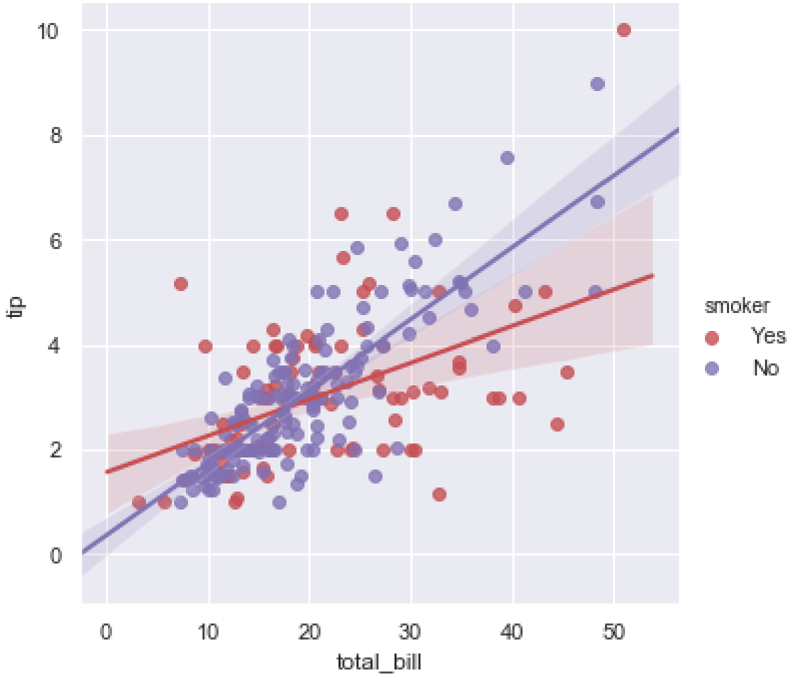
```
sns.lmplot(x="total_bill", y="tip", hue='smoker',data=tips,
          markers=["o","x"])
plt.show()
```



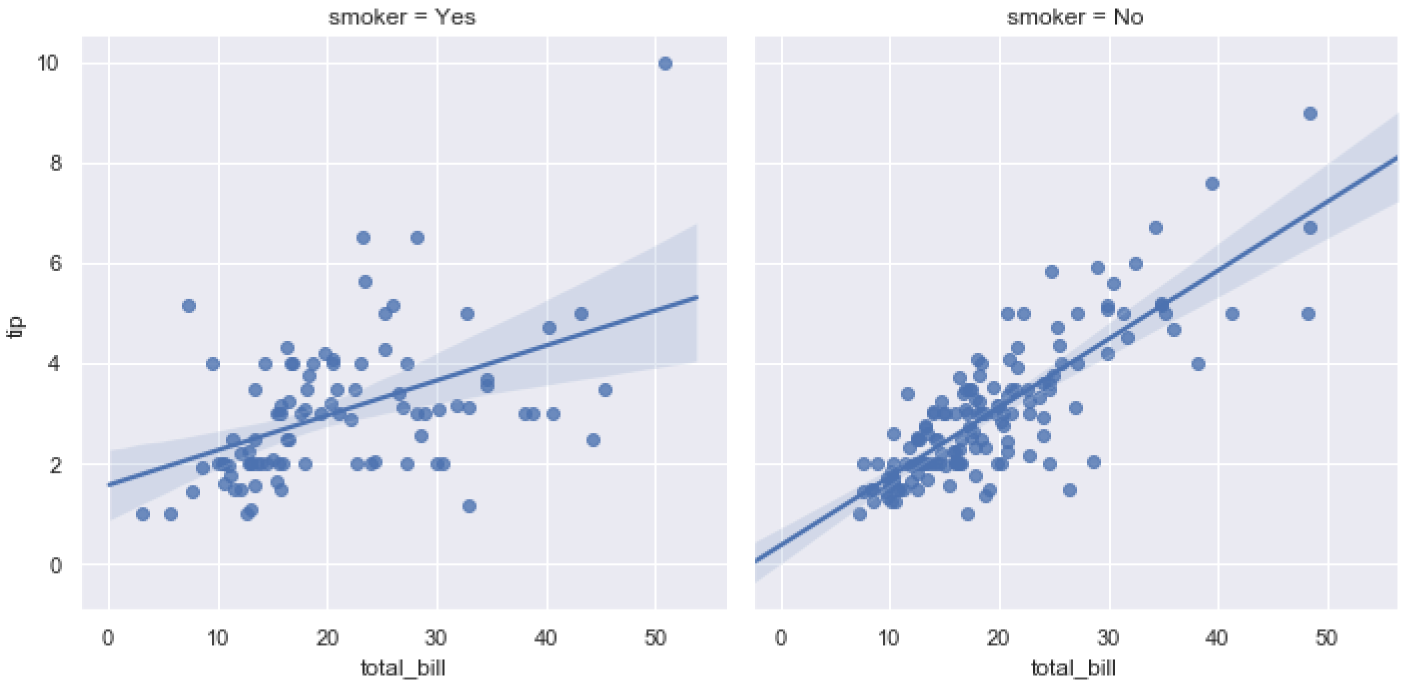
```
sns.lmplot(x="total_bill", y="tip", hue="smoker", data=tips, palette="Set1",
          markers=[">","<"])
plt.show()
```



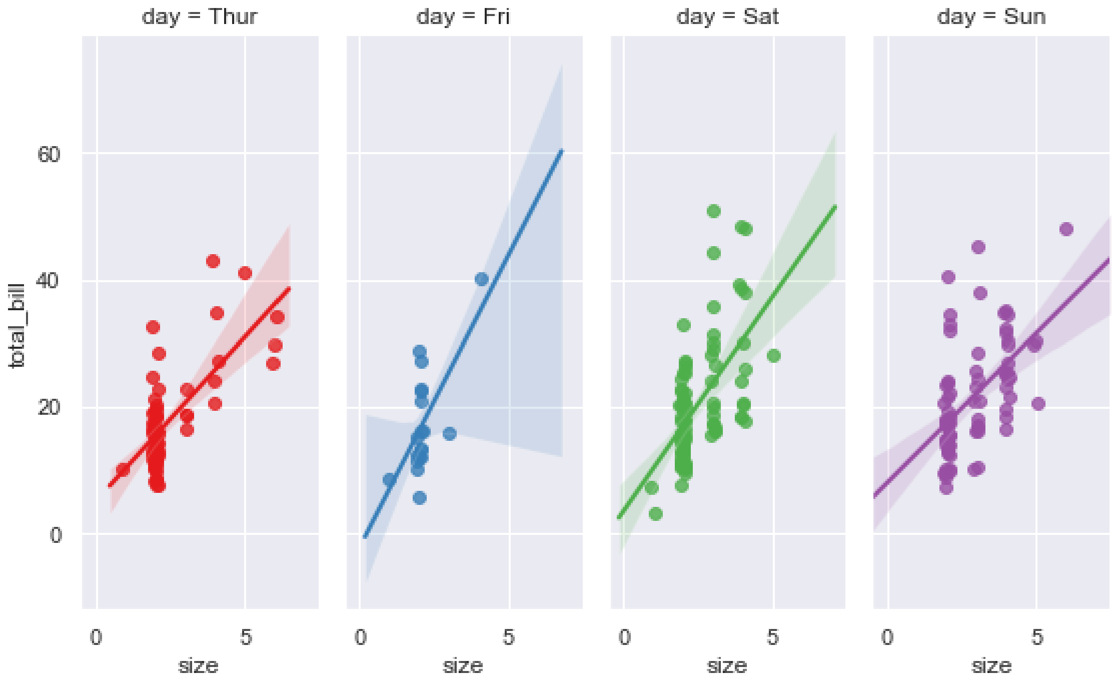
```
sns.lmplot(x="total_bill", y="tip", hue="smoker", data=tips,
           palette=dict(Yes="r", No="m"))
plt.show()
```



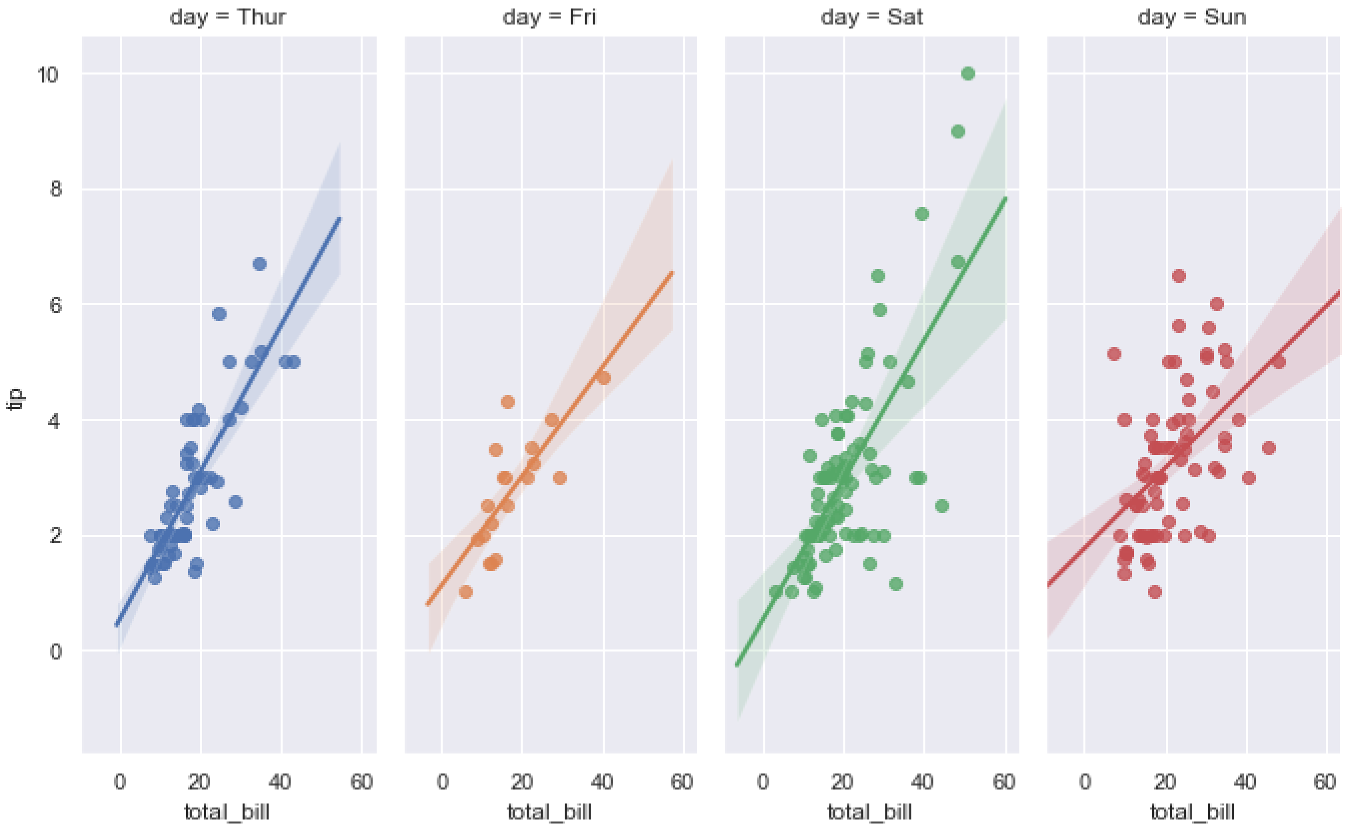
```
sns.lmplot(x="total_bill", y="tip", col="smoker", data=tips)
plt.show()
```



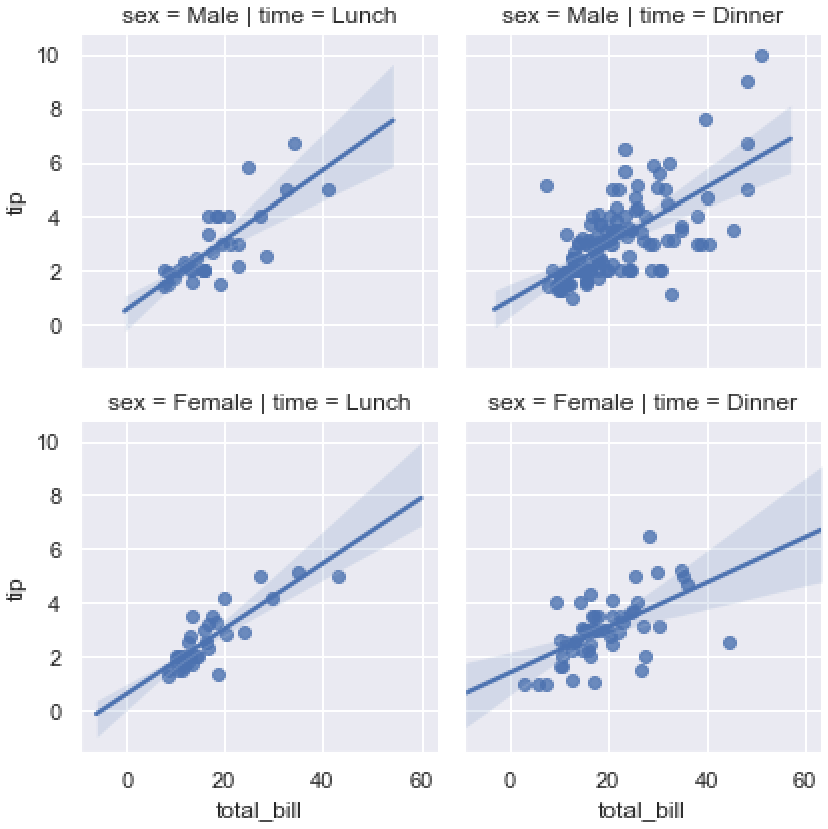
```
sns.lmplot(x='size',y='total_bill',hue='day',col='day',data=tips,aspect=0.4,
           x_jitter=.1,palette="Set1")
plt.show()
```



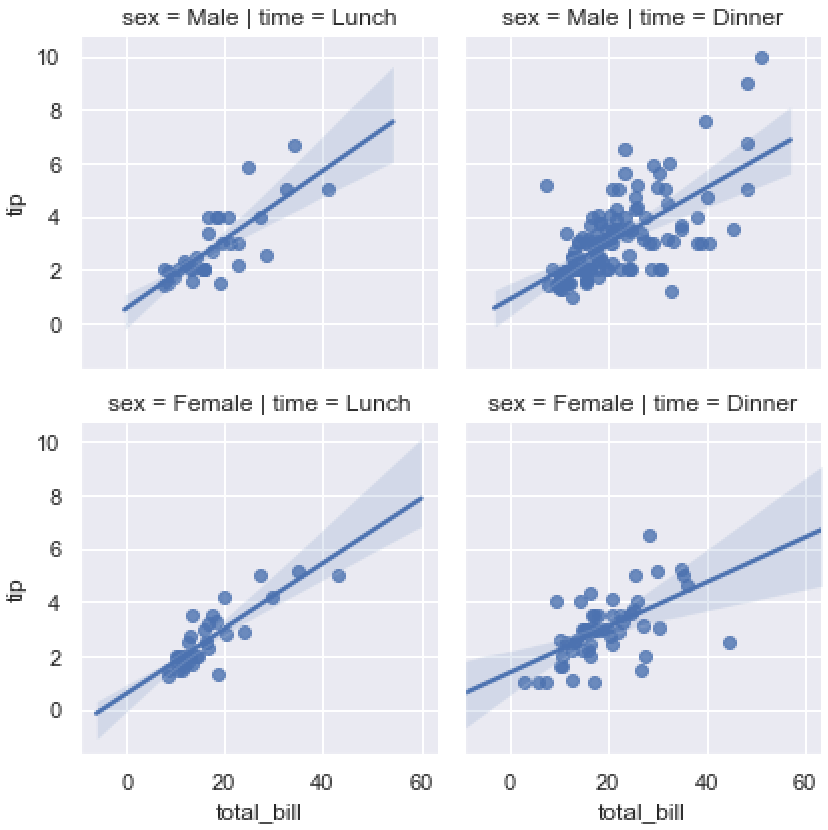
```
sns.lmplot(x="total_bill", y="tip", col="day", hue="day", data=tips,
           col_wrap=4, size=6,aspect=0.4)
plt.show()
```



```
sns.lmplot(x="total_bill", y="tip", row="sex", col="time", data=tips, size=3)
plt.show()
```



```
sns.lmplot(x='total_bill',y='tip',row='sex',col='time',data=tips,
          size=3)
g = (g.set_axis_labels('total_bill(US Dollars)',
                        'Tip')).set(xlim=(0,60),
                                    ylim=(0,12),xticks=[10,30,50],
                                    yticks=[2,6,10]).fig.subplots_adjust(wspace=.02)
plt.show()
```



kdeplot

```
np.random.seed(10)
```

```
mean, cov = [0,2],[(1, .5),(.5, 1)]
```

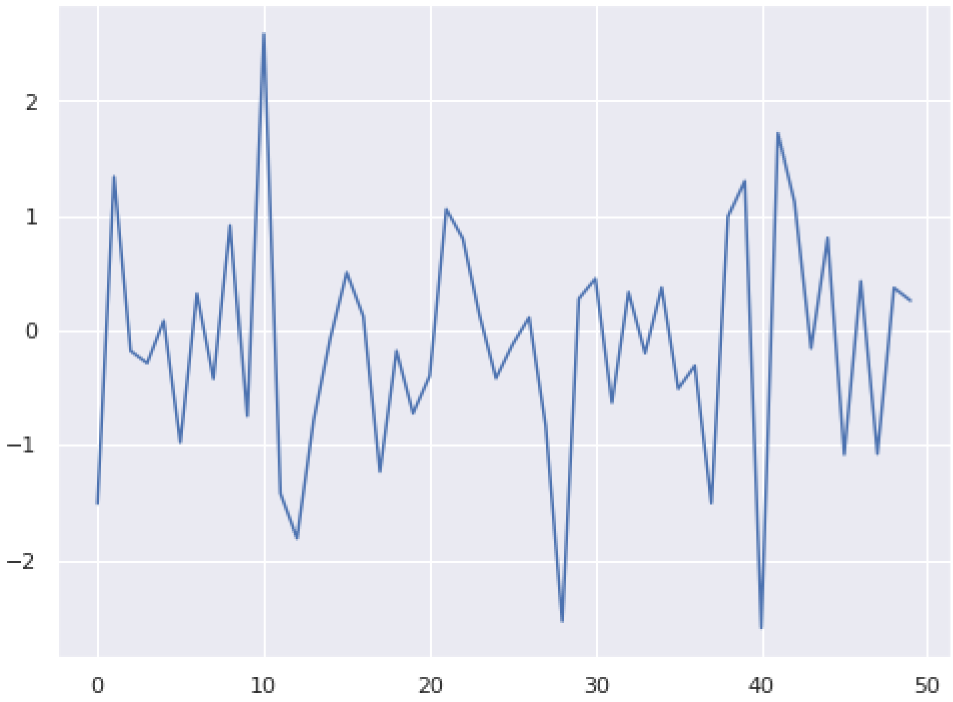
```
x,y = np.random.multivariate_normal(mean,cov,size=50).T
x
```

```
array([-1.51082723,  1.34254784, -0.17804996, -0.28421404,  0.08358362,
       -0.97653037,  0.32163435, -0.42056831,  0.91675795, -0.74574431,
        2.58444908, -1.42290719, -1.80945628, -0.78486393, -0.07169444,
        0.50615798,  0.12314225, -1.23067757, -0.17777585, -0.72215215,
       -0.39634135,  1.05824176,  0.80314519,  0.1376191 , -0.41572644,
       -0.12211242,  0.11346898, -0.83117842, -2.53260368,  0.27832104,
        0.45194018, -0.6286516 ,  0.33490191, -0.19674292,  0.37488635,
       -0.50702222, -0.3076567 , -1.5064103 ,  0.99594439,  1.30411574,
       -2.59154076,  1.71889063,  1.12288196, -0.15438215,  0.80903975,
       -1.08157546,  0.43056387, -1.07570917,  0.3727111 ,  0.26105424])
```

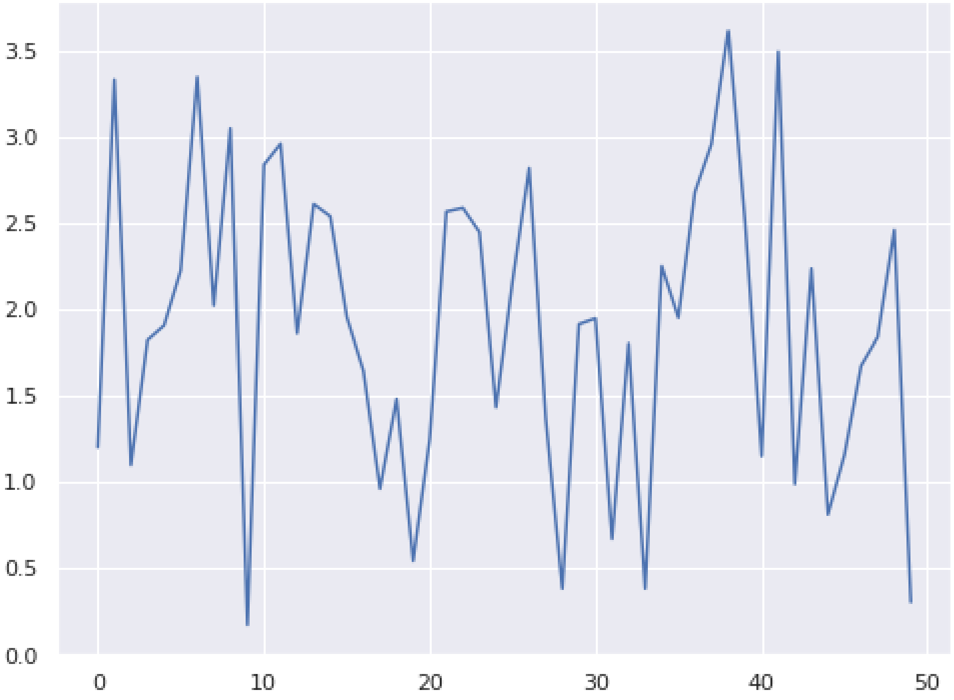
```
y
```

```
array([1.20445175, 3.33416399, 1.10186448, 1.82433448, 1.90898341,
       2.22650701, 3.34990843, 2.02456931, 3.05189483, 0.1744508 ,
       2.84107678, 2.96206014, 1.86316594, 2.61313245, 2.54150974,
       1.95684897, 1.64700024, 0.96433571, 1.48459181, 0.54587835,
       1.25278676, 2.56890455, 2.59044755, 2.44978904, 1.4368533 ,
       2.16698178, 2.821629  , 1.37240238, 0.38485526, 1.91614059,
       1.95021128, 0.67353837, 1.80560583, 0.38470105, 2.25298066,
       1.95388068, 2.68141576, 2.96124076, 3.61654506, 2.50610692,
       1.15327339, 3.49611693, 0.99116985, 2.23895907, 0.81464598,
       1.16296852, 1.67667091, 1.84255998, 2.46229871, 0.30654212])
```

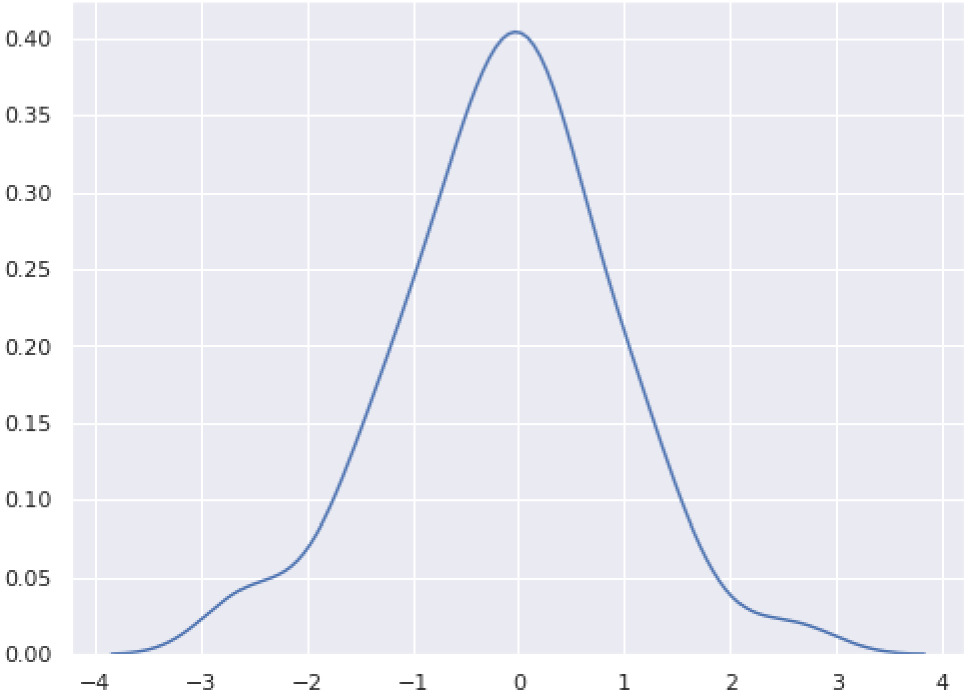
```
plt.figure(figsize=(8,6))
plt.plot(x)
plt.show()
```



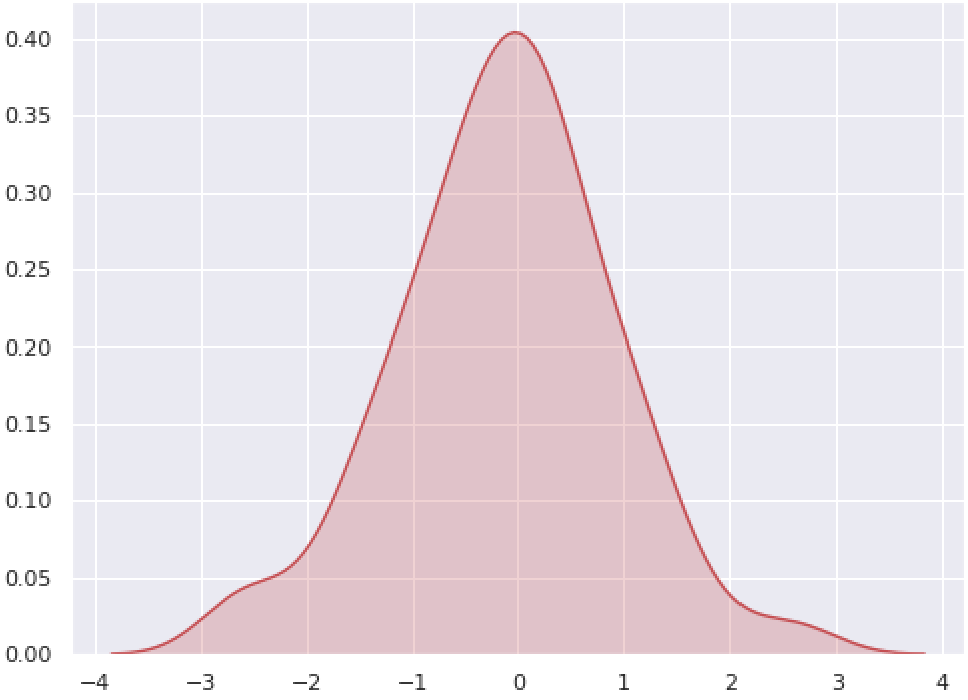
```
plt.figure(figsize=(8,6))
plt.plot(y)
plt.show()
```



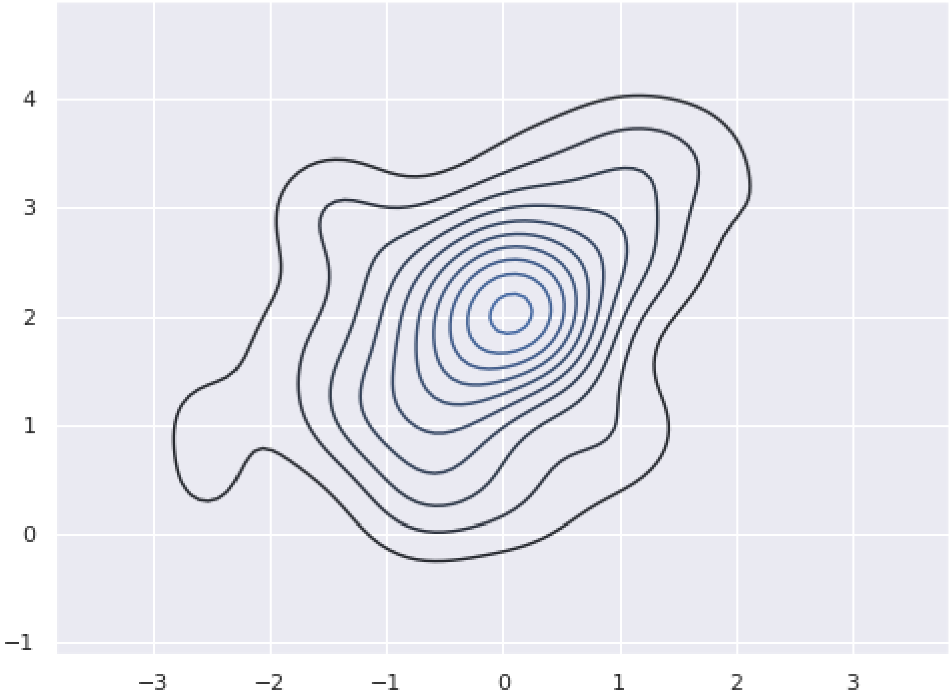
```
plt.figure(figsize=(8,6))
sns.kdeplot(x)
plt.show()
```



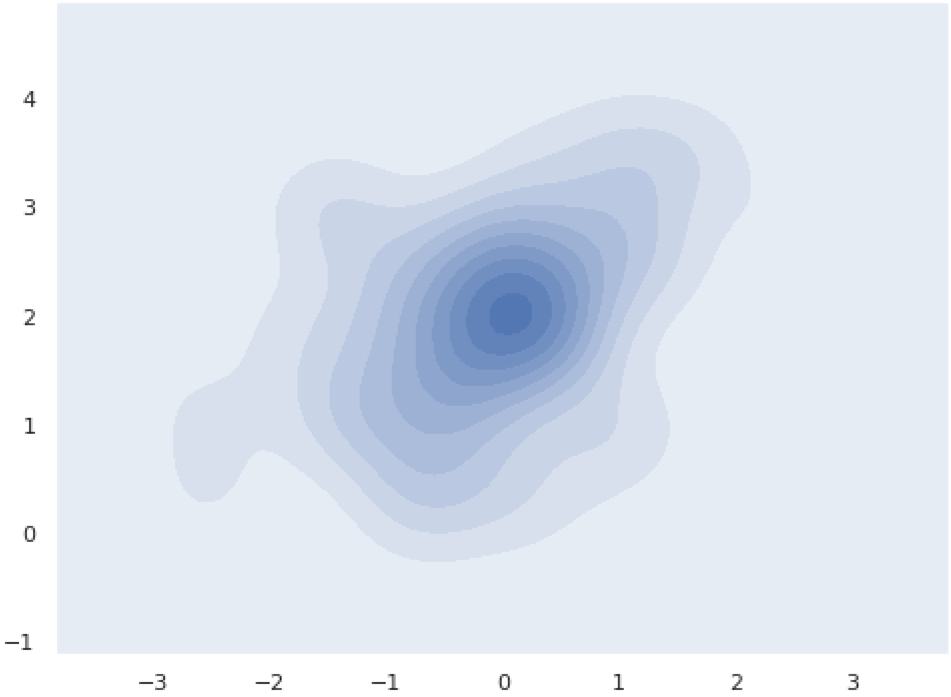
```
plt.figure(figsize=(8,6))
sns.kdeplot(x, shade=True, color='r')
plt.show()
```



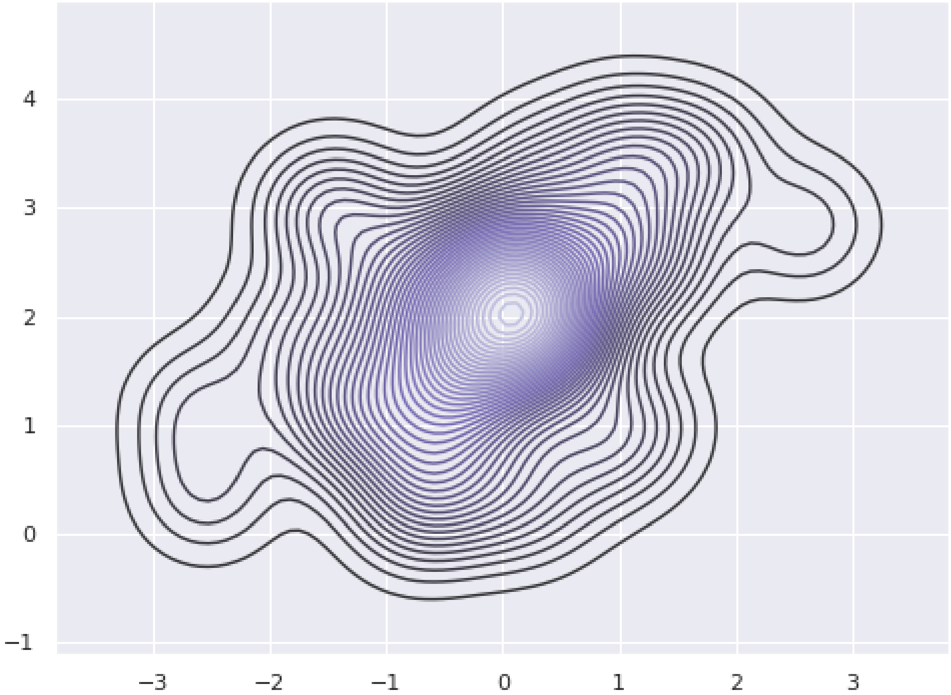
```
plt.figure(figsize=(8,6))
sns.kdeplot(x,y)
plt.show()
```



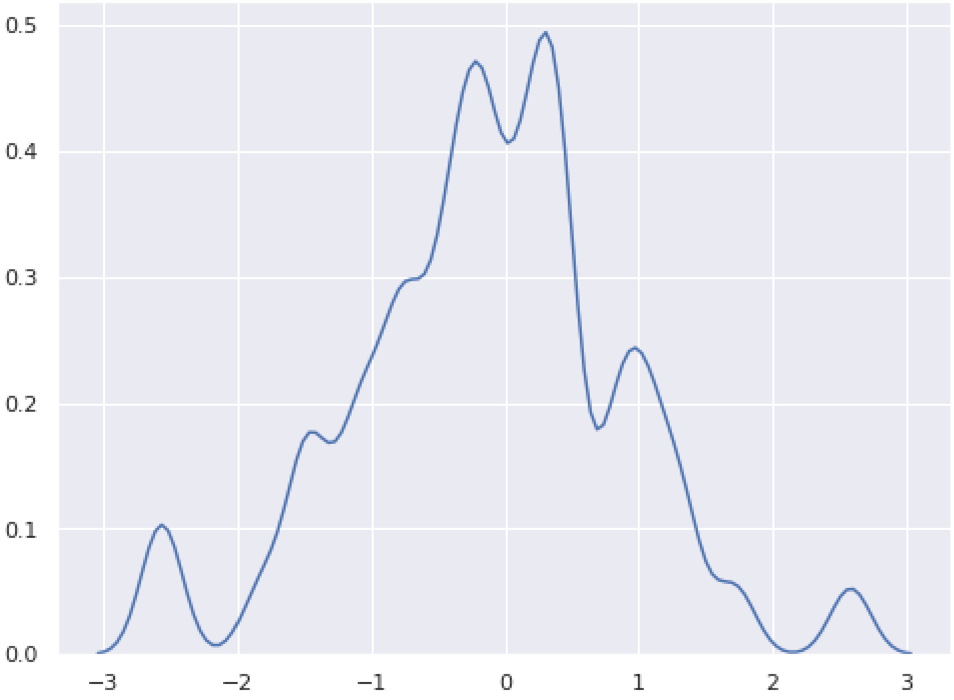
```
plt.figure(figsize=(8,6))
sns.kdeplot(x,y,shade=True)  # 그라데이션(shade)
plt.show()
```



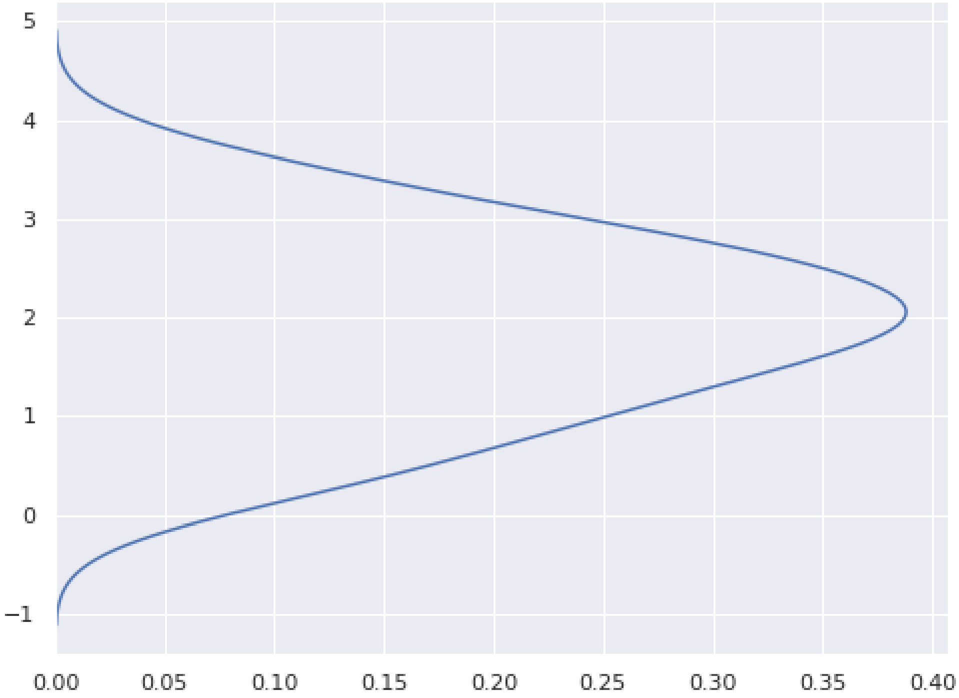
```
plt.figure(figsize=(8,6))
sns.kdeplot(x, y, n_levels=50, cmap='Purples_d')
plt.show()
```



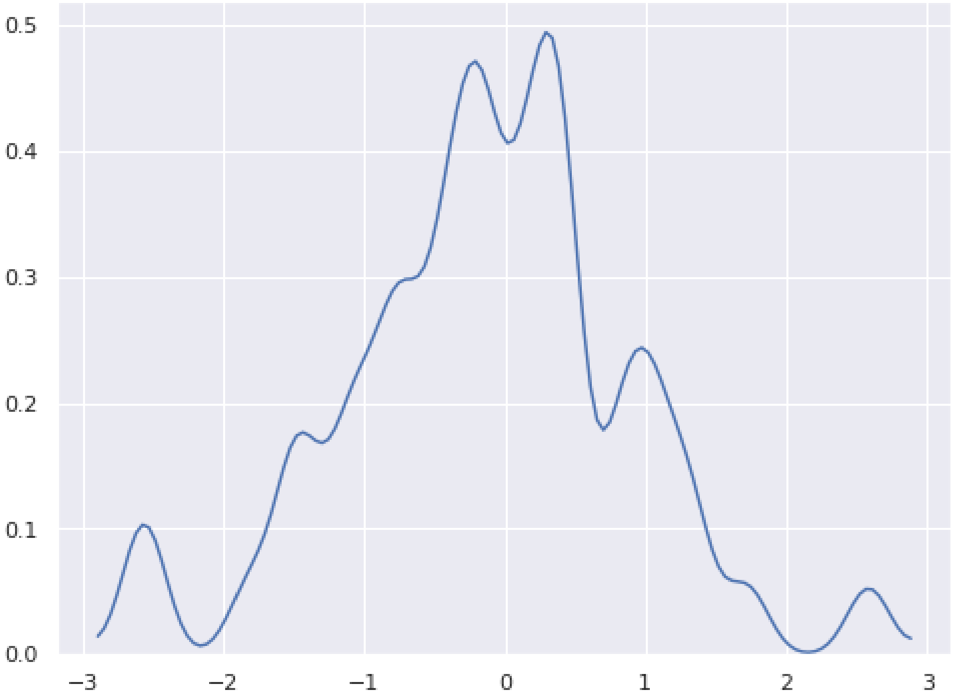
```
plt.figure(figsize=(8,6))
sns.kdeplot(x,bw=.15,)
plt.show()
```



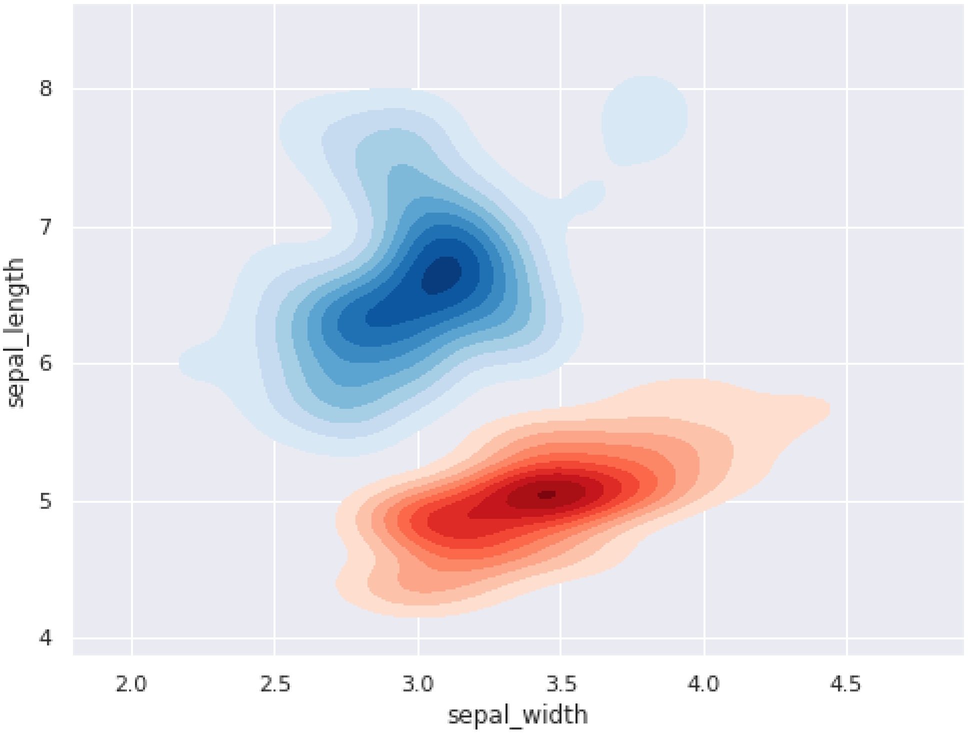
```
plt.figure(figsize=(8,6))
sns.kdeplot(y, vertical=True)   # vertical : 뒤집기
plt.show()
```



```
plt.figure(figsize=(8,6))
sns.kdeplot(x, bw=.15, cut=2)
plt.show()
```



```
plt.figure(figsize=(8,6))
iris = sns.load_dataset("iris")
setosa = iris.loc[iris.species == "setosa"]
virginica = iris.loc[iris.species == "virginica"]
sns.kdeplot(setosa.sepal_width, setosa.sepal_length, cmap="Reds",
            shade=True, shade_lowest=False)
sns.kdeplot(virginica.sepal_width, virginica.sepal_length,
            cmap="Blues", shade=True, shade_lowest=False)
plt.show()
```

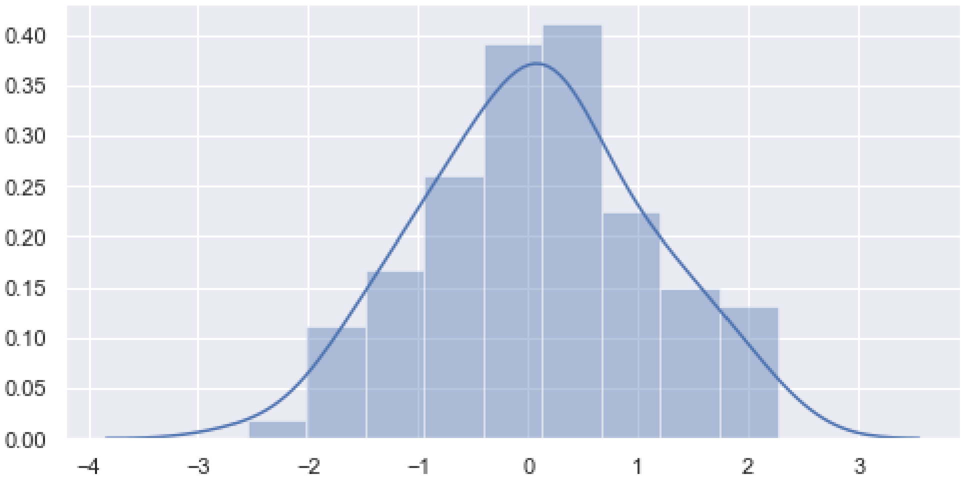


distplot

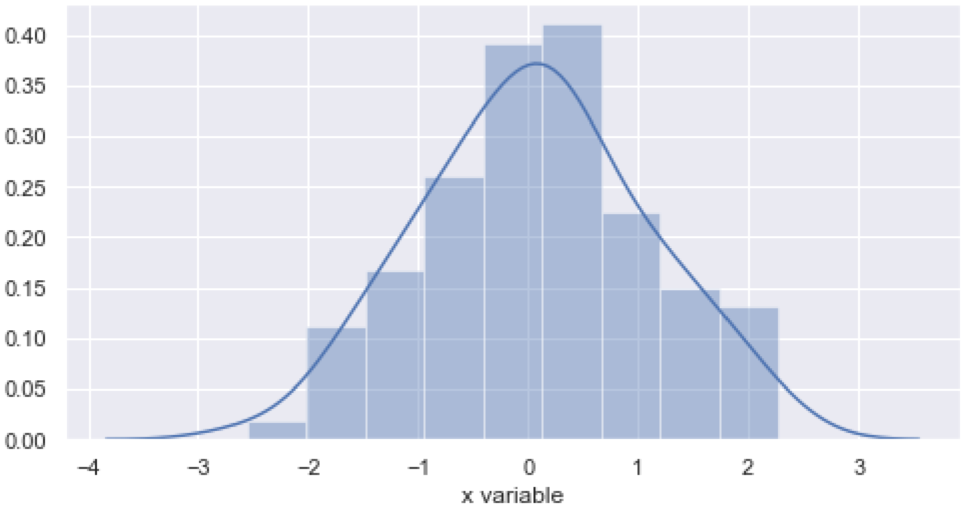
> http://seaborn.pydata.org/generated/seaborn.distplot.html?highlight=distplot#seaborn.distplot

```
sns.set(rc={"figure.figsize": (8, 4)}); np.random.seed(0)
```

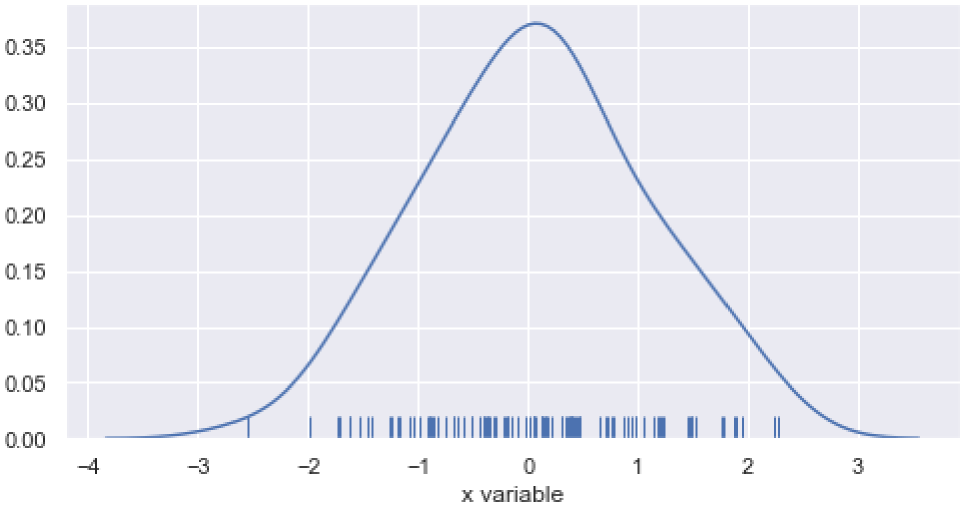
```
x = np.random.randn(100)
sns.distplot(x)
plt.show()
```



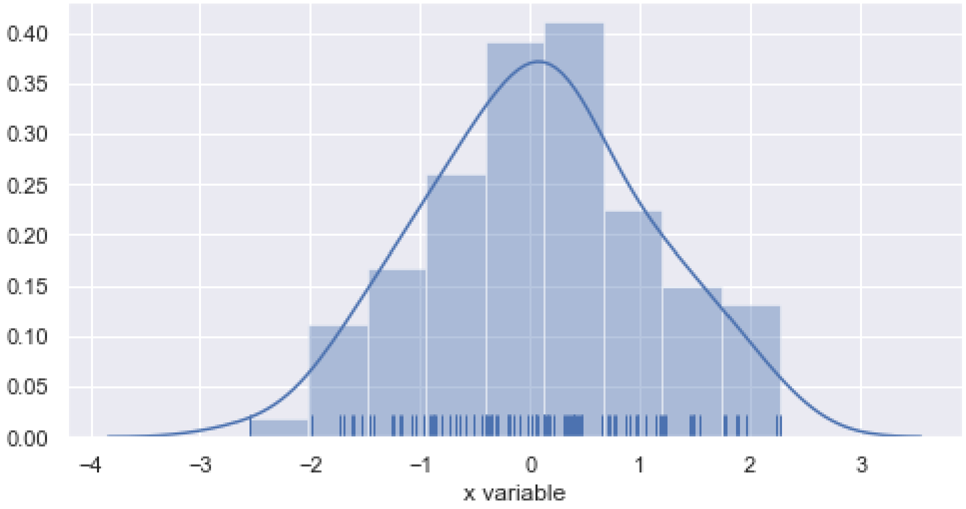
```
x = pd.Series(x, name="x variable")
sns.distplot(x)
plt.show()
```



```
sns.distplot(x, rug=True, hist=False)   # rug : 바코드 모양
plt.show()
```

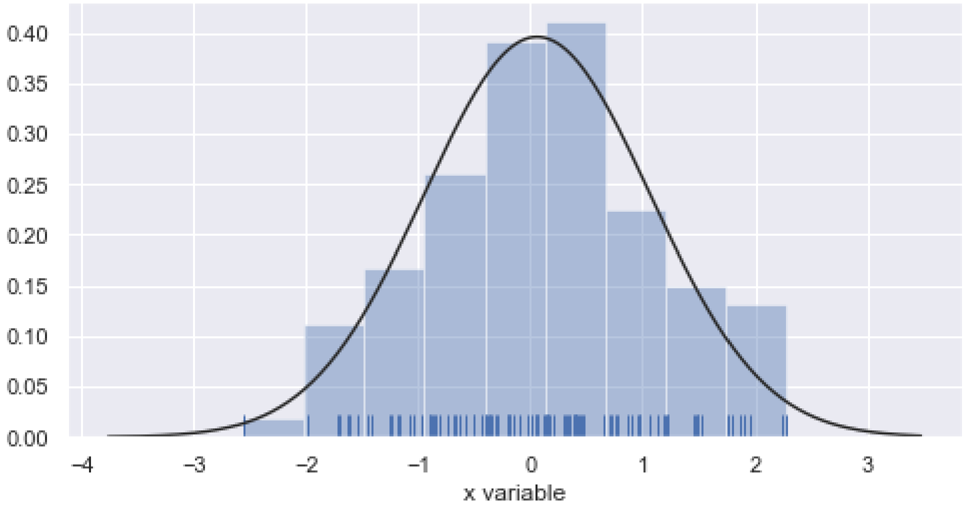


```
sns.distplot(x, rug=True, hist=True)
plt.show()
```

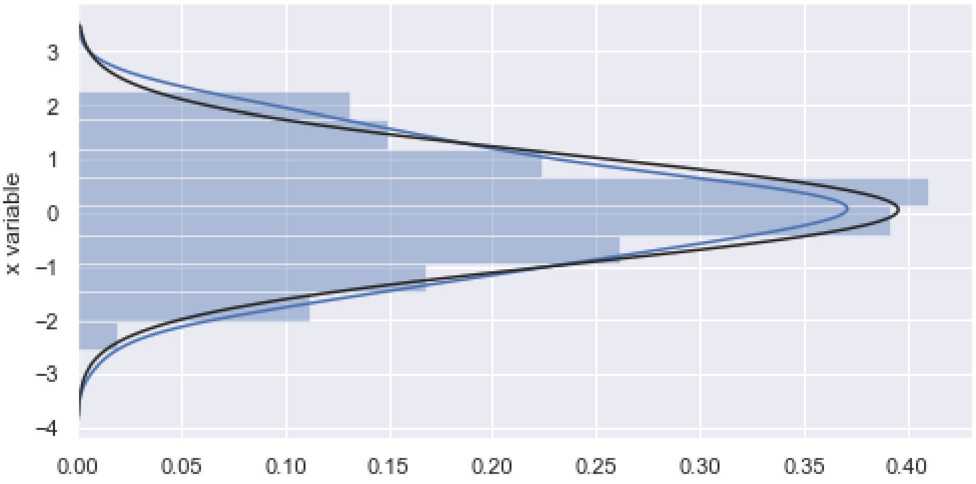


```
from scipy.stats import norm
```

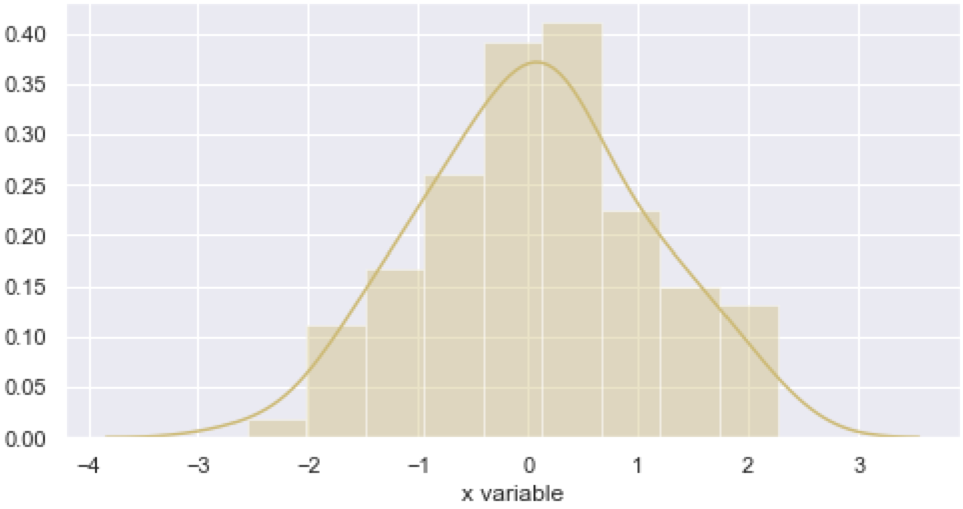
```
sns.distplot(x, rug = True, fit = norm, kde=False)
plt.show()
```



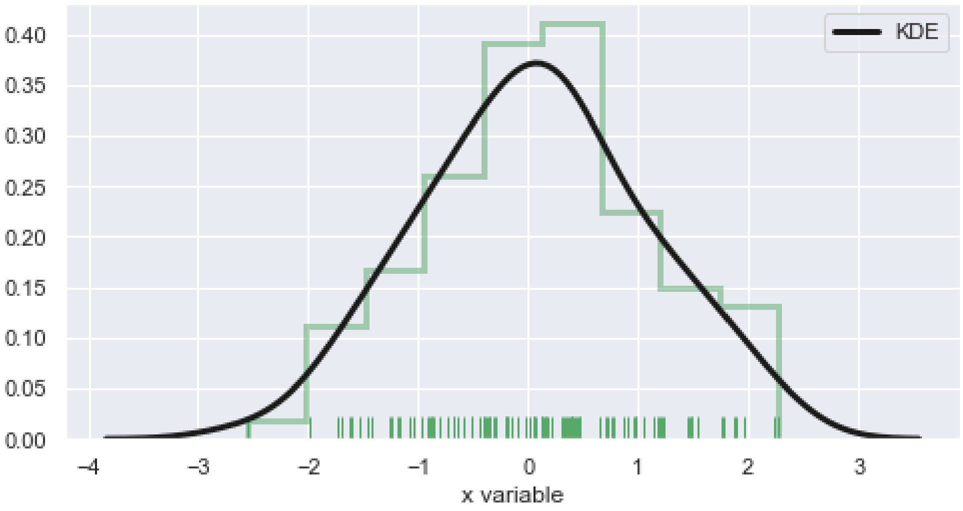
```
sns.distplot(x, rug = False, fit = norm, kde=True, vertical=True)
plt.show()
```



```
sns.set_color_codes()
sns.distplot(x, color="y")
plt.show()
```



```
sns.distplot(x, rug=True, rug_kws={"color": "g"},
             kde_kws={"color": "k", "lw": 3, "label": "KDE"},
             hist_kws={"histtype": "step", "linewidth": 3,
                       "alpha": .5, "color": "g"})
plt.show()
```

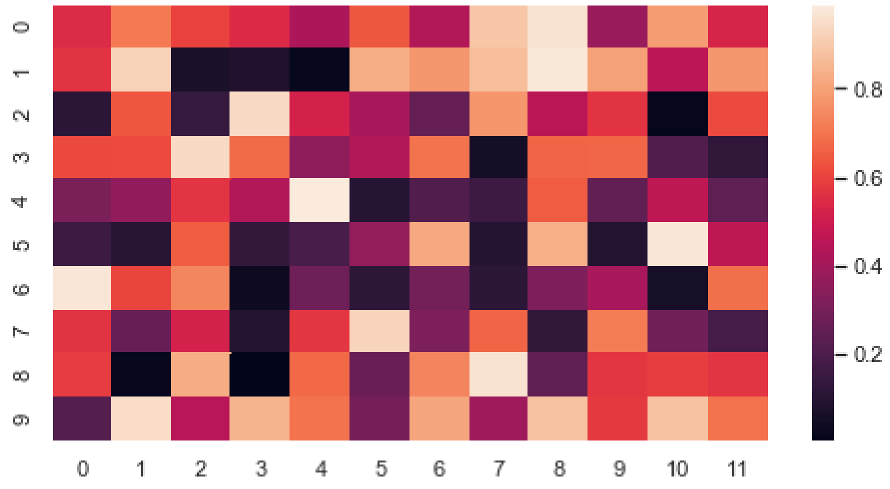


heatmap

```
import numpy as np; np.random.seed(0)
import seaborn as sns; sns.set()
uniform_data = np.random.rand(10, 12)
uniform_data
```

```
array([[0.5488135 , 0.71518937, 0.60276338, 0.54488318, 0.4236548 ,
        0.64589411, 0.43758721, 0.891773  , 0.96366276, 0.38344152,
        0.79172504, 0.52889492],
       [0.56804456, 0.92559664, 0.07103606, 0.0871293 , 0.0202184 ,
        0.83261985, 0.77815675, 0.87001215, 0.97861834, 0.79915856,
        0.46147936, 0.78052918],
       [0.11827443, 0.63992102, 0.14335329, 0.94466892, 0.52184832,
        0.41466194, 0.26455561, 0.77423369, 0.45615033, 0.56843395,
        0.0187898 , 0.6176355 ],
       [0.61209572, 0.616934  , 0.94374808, 0.6818203 , 0.3595079 ,
        0.43703195, 0.6976312 , 0.06022547, 0.66676672, 0.67063787,
        0.21038256, 0.1289263 ],
       [0.31542835, 0.36371077, 0.57019677, 0.43860151, 0.98837384,
        0.10204481, 0.20887676, 0.16130952, 0.65310833, 0.2532916 ,
        0.46631077, 0.24442559],
       [0.15896958, 0.11037514, 0.65632959, 0.13818295, 0.19658236,
        0.36872517, 0.82099323, 0.09710128, 0.83794491, 0.09609841,
        0.97645947, 0.4686512 ],
       [0.97676109, 0.60484552, 0.73926358, 0.03918779, 0.28280696,
        0.12019656, 0.2961402 , 0.11872772, 0.31798318, 0.41426299,
        0.0641475 , 0.69247212],
       [0.56660145, 0.26538949, 0.52324805, 0.09394051, 0.5759465 ,
        0.9292962 , 0.31856895, 0.66741038, 0.13179786, 0.7163272 ,
        0.28940609, 0.18319136],
       [0.58651293, 0.02010755, 0.82894003, 0.00469548, 0.67781654,
        0.27000797, 0.73519402, 0.96218855, 0.24875314, 0.57615733,
        0.59204193, 0.57225191],
       [0.22308163, 0.95274901, 0.44712538, 0.84640867, 0.69947928,
        0.29743695, 0.81379782, 0.39650574, 0.8811032 , 0.58127287,
        0.88173536, 0.69253159]])
```

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



```
cbar=False # cbar=False 컬러바 숨기기
cbar_kws={"orientation": "horizontal"} #컬러바 아래쪽 지평선에 두기
```