

---

# 데이터 시각화

---

# Table of contents

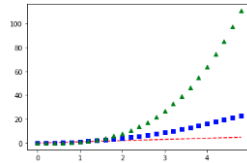
---

## 데이터시각화

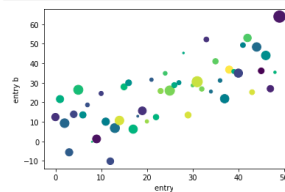
- Matplotlib을 이용한 시각화
- Pandas를 이용한 시각화
- Seaborn을 이용한 시각화
- Folium을 이용한 지도 시각화

# Matplotlib을 이용한 시각화

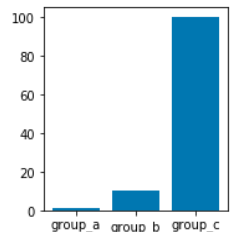
- 직선그래프(plot)



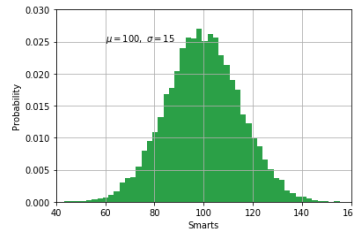
- 산포도 (scatter)



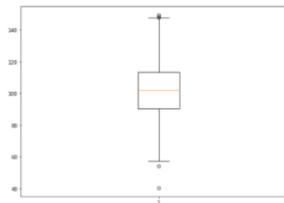
- 막대그래프(bar, barh)



- 히스토그램(hist)



- 상자그림(boxplot)



# Matplotlib을 이용한 시각화

## ➤ Figure와 Subplot

- matplotlib에서 그래프는 Figure 객체 내에 존재
- 기본으로 존재하는 Figure가 아닌 새로운 Figure을 생성하기 위해 pyplot에 있는 figure 함수를 이용하여 생성

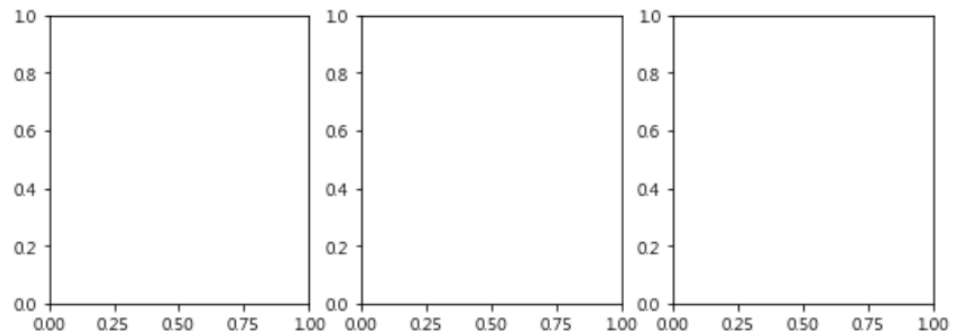
```
>>> fig1 = plt.figure(1, figsize=(9,3))
```

- figsize 함수를 이용하여 그래프의 크기(가로, 세로)를 정함
- Figure 객체 내에 서브 플롯을 생성하기 위해서 subplot 함수를 이용

```
>>> plt.subplot(131)
```

```
>>> plt.subplot(132)
```

```
>>> plt.subplot(133)
```



# Pandas를 이용한 시각화

- Pandas를 이용한 시각화
  - 데이터프레임.plot(kind='')

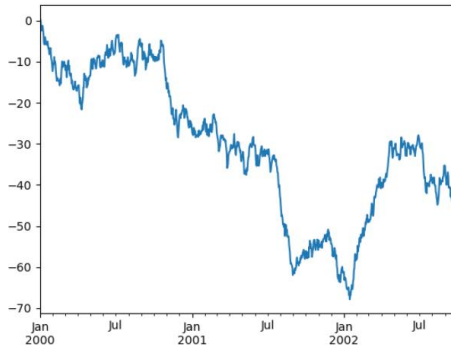
```
import matplotlib.pyplot as plt

plt.close('all')

ts = pd.Series(np.random.randn(1000),
               index=pd.date_range('1/1/2000', periods=1000))

ts = ts.cumsum()

ts.plot()
```



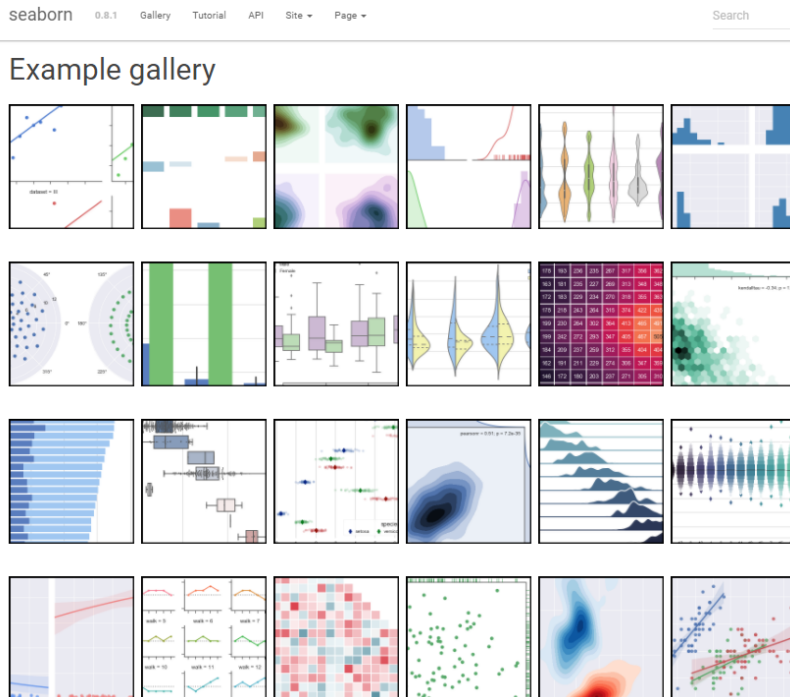
## Kind 옵션

- ✓ 'bar' or 'barh' for bar plots
- ✓ 'hist' for histogram
- ✓ 'box' for boxplot
- ✓ 'kde' or 'density' for density plots
- ✓ 'area' for area plots
- ✓ 'scatter' for scatter plots
- ✓ 'hexbin' for hexagonal bin plots
- ✓ 'pie' for pie plots

# Seaborn을 이용한 시각화

## ➤ Seaborn이란?

- matplotlib를 기반으로 하는 Python visualization 라이브러리로, attractive한 통계적 그래프를 제공하는 라이브러리
- 공식사이트 <http://seaborn.pydata.org/>



boxplot, Implot, pairplot, heatmap

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

# Seaborn을 이용한 시각화

## ➤ 스타일

- Seaborn을 import 하면 색상 등을 Matplotlib에서 제공하는 기본 스타일이 아닌 Seaborn에서 지정한 기본 스타일로 바꿈
- set 명령으로 색상, 틱 스타일 등 전반적인 플롯 스타일을 Seaborn 기본 스타일로 바꿈
- set\_style 명령은 tick 스타일만 변경 가능
  - darkgrid, whitegrid, dark, white, ticks 스타일을 제공
- set\_color\_codes 명령으로는 기본 색상을 가리키는 문자열을 변경

```
>>> import seaborn as sns
>>> sns.set()
>>> sns.set_style('whitegrid')
>>> sns.set_color_codes()
```

# Folium을 이용한 지도 시각화

➤ <https://python-visualization.github.io/folium/>

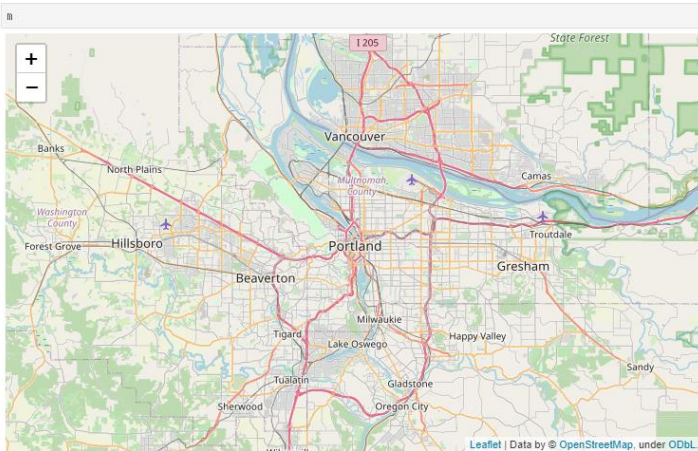
Folium 11



Python data, leaflet.js maps

```
import folium
```

```
m = folium.Map(location=[45.5236, -122.6750])
```



```
m = folium.Map(
    location=[45.372, -121.6972],
    zoom_start=12,
    tiles='Stamen Terrain'
)

folium.Marker(
    location=[45.3288, -121.6625],
    popup='Mt. Hood Meadows',
    icon=folium.Icon(icon='cloud')
).add_to(m)

folium.Marker(
    location=[45.3311, -121.7113],
    popup='Timberline Lodge',
    icon=folium.Icon(color='green')
).add_to(m)

folium.Marker(
    location=[45.3300, -121.6823],
    popup='Some Other Location',
    icon=folium.Icon(color='red', icon='info-sign')
).add_to(m)

m
```

