Woka

시각화

Woka Inc.
Song Chioh

chioh@woka.kr

Woka

목차

1. 시각화 개요

데이터 시각화

데이터를 그래프나 차트와 같은 시각적 요소로 표현 → 데이터의 **패턴, 트렌드, 이상치 등을 쉽게 파악**

데이터 분석 결과를 **명확하고 직관적으로** 전달 = 원활한 **커뮤니케이션**

Python 제공 시각화 라이브러리: matplotlib, seaborn

https://velog.io/@dosigner/Data-visualization
https://github.com/matplotlib/matplotlib

Woka

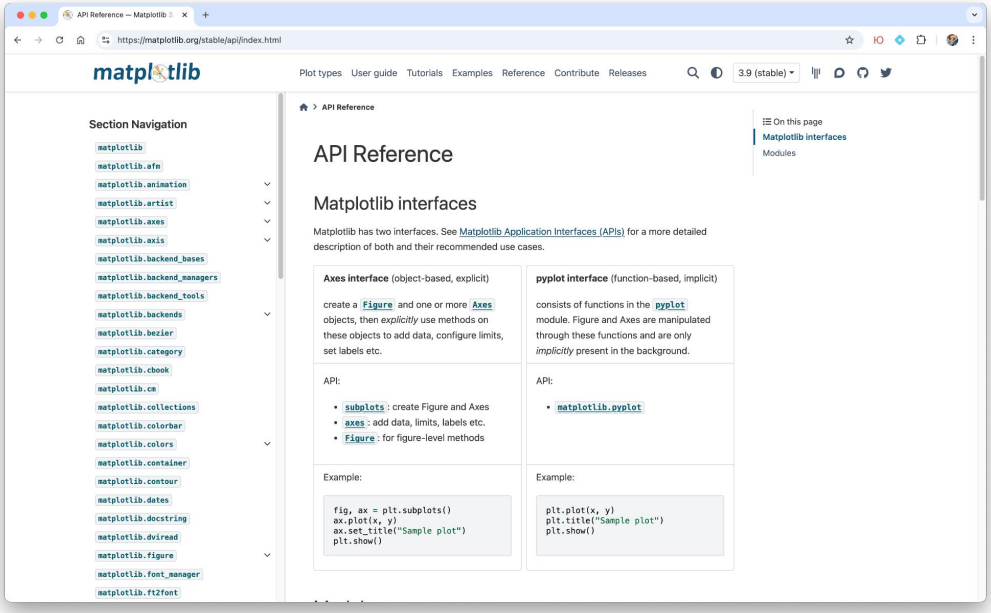
2. matplotlib

matplotlib

Python에서 가장 널리 사용되는 데이터 시각화 라이브러리

다양한 종류의 플롯과 차트

https://matplotlib.org/stable/api/index.html



matplotlib

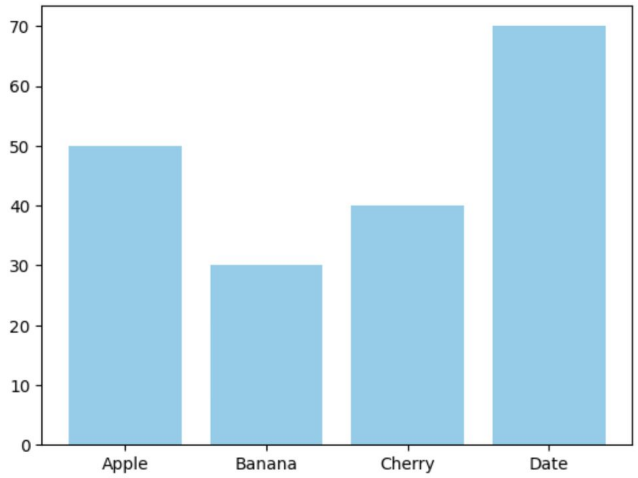
패키지 불러오기

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

Bar Plot: `plt.bar()`

범주형 데이터를 시각화하는 데 유용

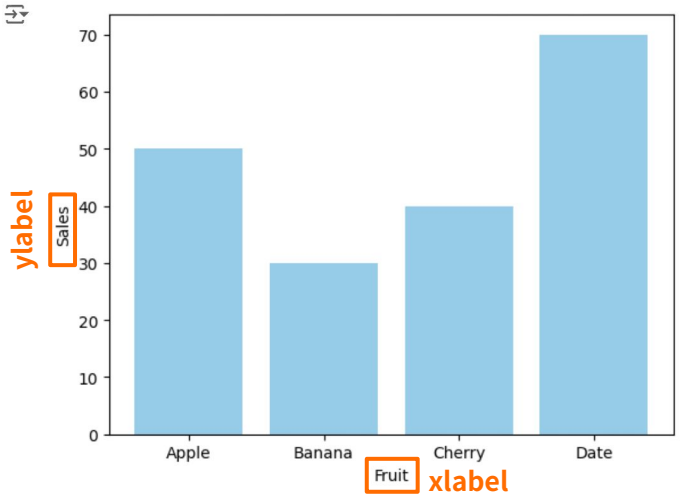
```
fruits = ['Apple', 'Banana', 'Cherry', 'Date']
sales = [50, 30, 40, 70]

plt.bar(fruits, sales, color='skyblue')
plt.show()
```



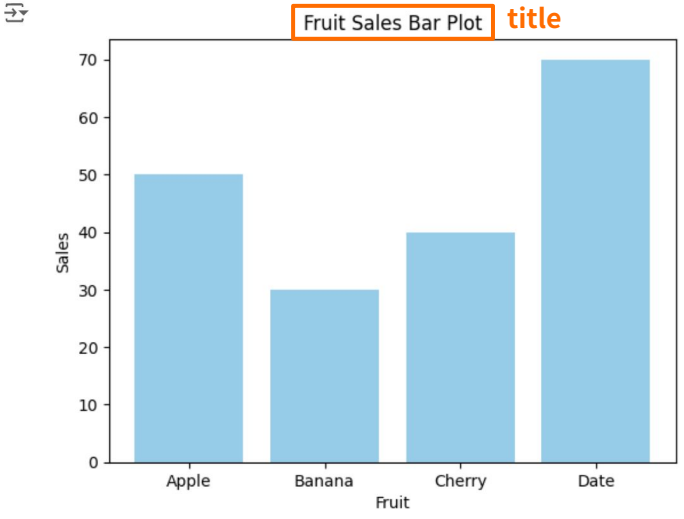
- xlabel, ylabel, title 추가하기

데이터의 의미를 명확하게

```
fruits = ['Apple', 'Banana', 'Cherry', 'Date']
sales = [50, 30, 40, 70]

plt.bar(fruits, sales, color='skyblue')
plt.xlabel('Fruit')
plt.ylabel('Sales')
plt.show()
```



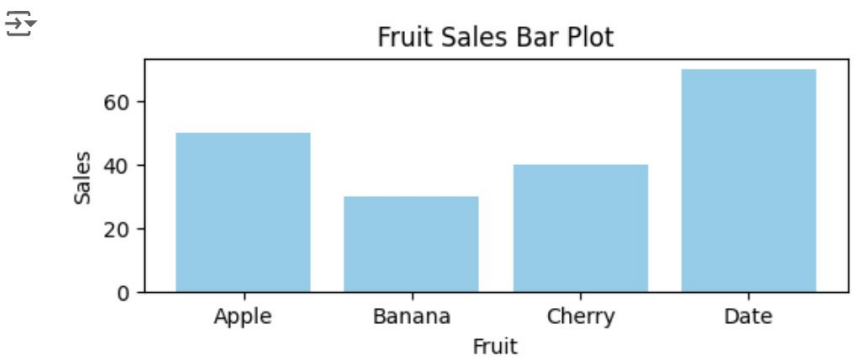

```
fruits = ['Apple', 'Banana', 'Cherry', 'Date']
sales = [50, 30, 40, 70]

plt.bar(fruits, sales, color='skyblue')
plt.xlabel('Fruit')
plt.ylabel('Sales')
plt.title('Fruit Sales Bar Plot')
plt.show()
```



- figsize

그래프의 크기 조절

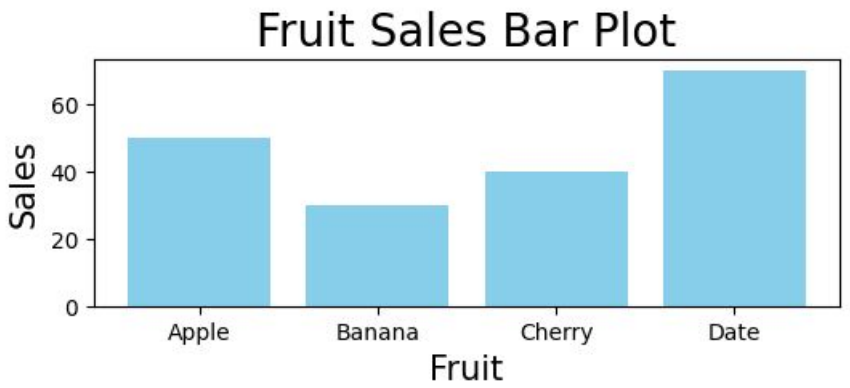
```
[20] fruits = ['Apple', 'Banana', 'Cherry', 'Date']
     sales = [50, 30, 40, 70]

     plt.figure(figsize=(6, 2))
     plt.bar(fruits, sales, color='skyblue')
     plt.xlabel('Fruit')
     plt.ylabel('Sales')
     plt.title('Fruit Sales Bar Plot')
     plt.show()
```



- fontsize

텍스트 크기 변경

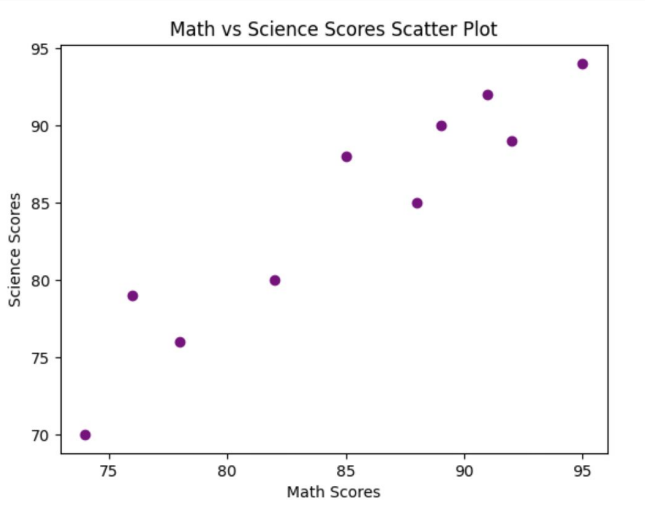
```
fruits = ['Apple', 'Banana', 'Cherry', 'Date']
sales = [50, 30, 40, 70]

plt.figure(figsize=(6, 2))
plt.bar(fruits, sales, color='skyblue')
plt.title('Fruit Sales Bar Plot', fontsize=20)
plt.xlabel('Fruit', fontsize=15)
plt.ylabel('Sales', fontsize=15)
plt.show()
```



Scatter Plot: `plt.scatter()`

두 변수 간의 관계를 시각화하는 데 사용

```
math_scores = [85, 78, 92, 88, 76, 95, 89, 74, 91, 82]
science_scores = [88, 76, 89, 85, 79, 94, 90, 70, 92, 80]

plt.scatter(math_scores, science_scores, color='purple')
plt.xlabel('Math Scores')
plt.ylabel('Science Scores')
plt.title('Math vs Science Scores Scatter Plot')
plt.show()
```

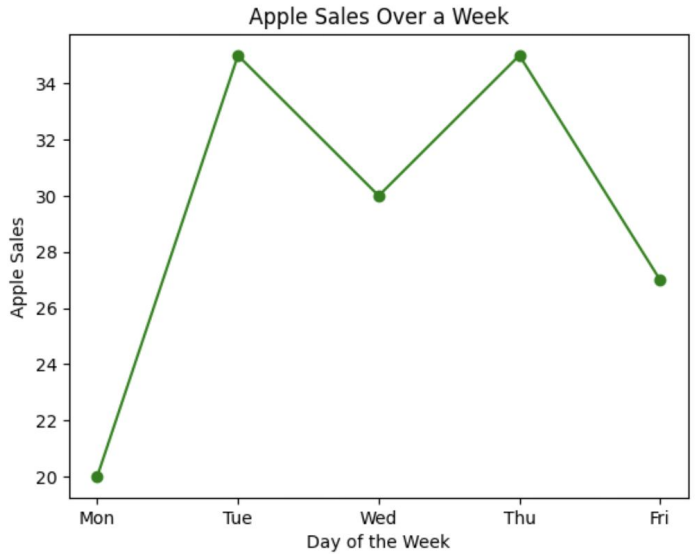


Line Plot: `plt.plot()`

시간에 따른 데이터의 변화 시각화

변수 간의 관계 시각화

```
days = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri']
apple_sales = [20, 35, 30, 35, 27]

plt.plot(days, apple_sales, marker='o', color='green')
plt.xlabel('Day of the Week')
plt.ylabel('Apple Sales')
plt.title('Apple Sales Over a Week')
plt.show()
```

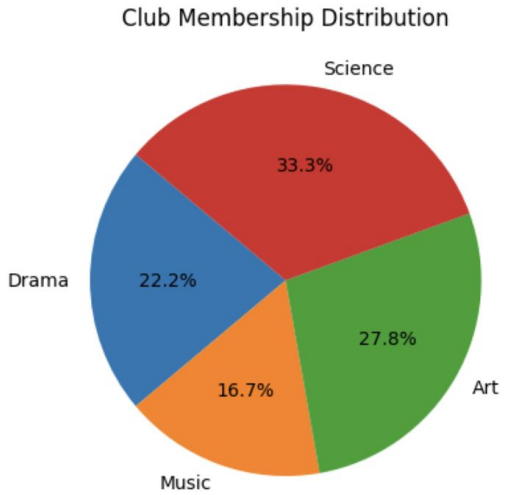


Pie Chart: `plt.pie()`

데이터의 구성 비율 시각화

```
clubs = ['Drama', 'Music', 'Art', 'Science']
members = [20, 15, 25, 30]

plt.pie(members, labels=clubs, autopct='%1.1f%%', startangle=140)
plt.title('Club Membership Distribution')
plt.show()
```

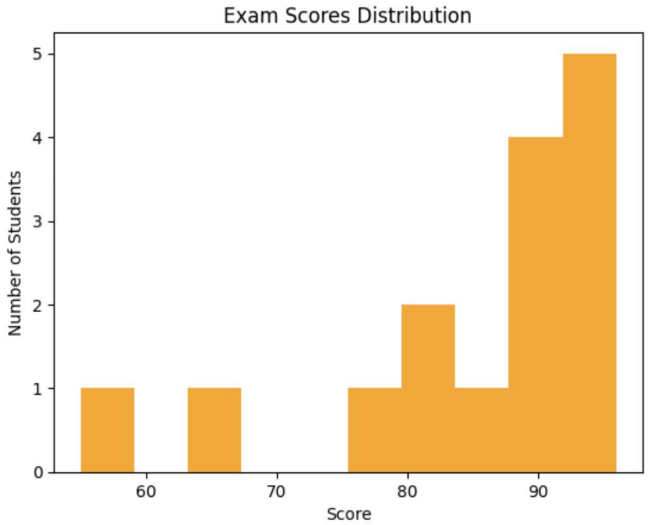
- autopct: 숫자값 표현식
- startangle: 파이의 시작각도(시계반대방향)



Histogram

데이터의 분포 시각화

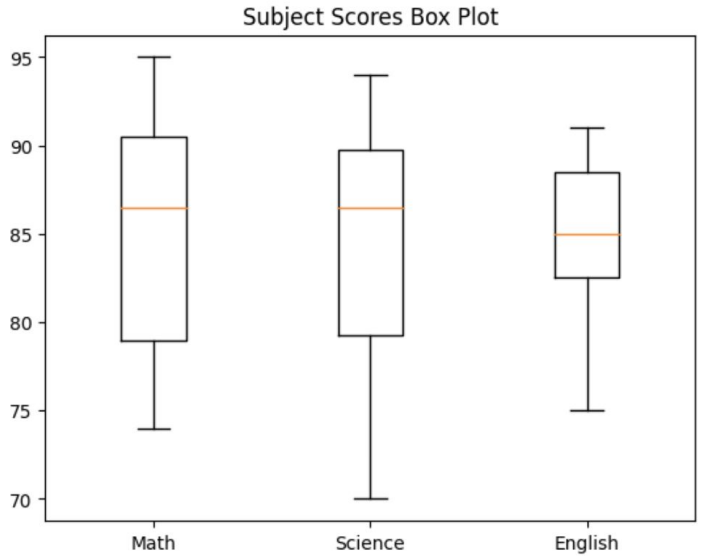
```
scores = [55, 67, 78, 81, 82, 85, 88, 89, 90, 91, 92, 93, 94, 95, 96]

plt.hist(scores, bins=10, color='orange')
plt.xlabel('Score')
plt.ylabel('Number of Students')
plt.title('Exam Scores Distribution')
plt.show()
```



Boxplot

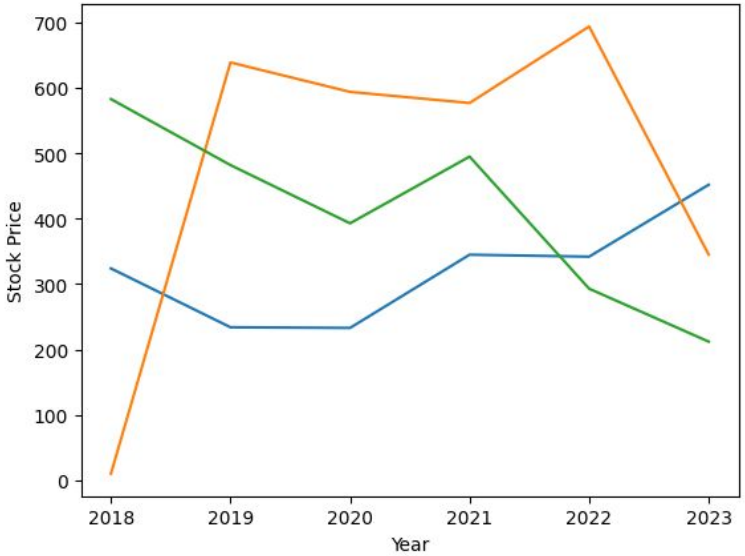
데이터의 분포(중앙값, 사분위수, 최소값, 최대값, 이상치) 시각화

```
data = {'Math': [85, 78, 92, 88, 76, 95, 89, 74, 91, 82],
        'Science': [88, 76, 89, 85, 79, 94, 90, 70, 92, 80],
        'English': [75, 85, 89, 91, 82, 87, 78, 85, 89, 84]}

plt.boxplot(data.values())
plt.xticks([1, 2, 3], data.keys())
plt.title('Subject Scores Box Plot')
plt.show()
```



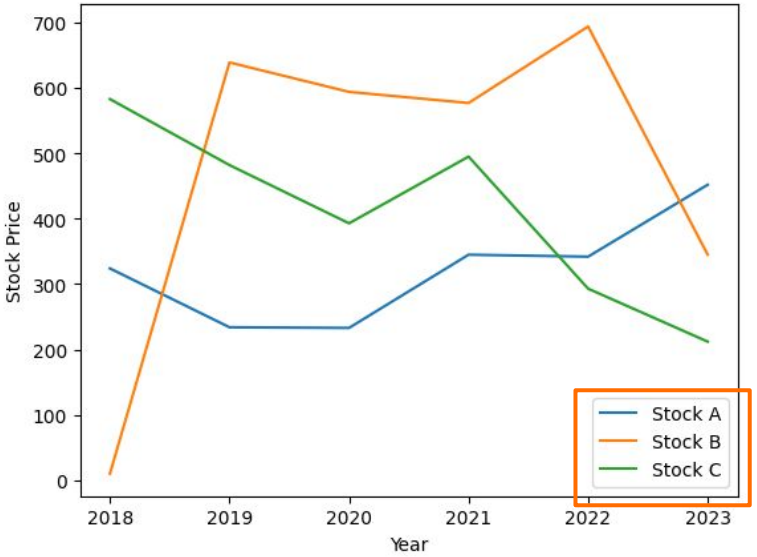
하나의 차트에 여러 plot 그리기

```
year = [2018, 2019, 2020, 2021, 2022, 2023]
stockA = [324, 234, 233, 345, 342, 452]
stockB = [10, 639, 594, 577, 694, 345]
stockC = [583, 482, 393, 495, 293, 212]

plt.plot(year, stockA)
plt.plot(year, stockB)
plt.plot(year, stockC)
plt.xlabel('Year')
plt.ylabel('Stock Price')
plt.show()
```



범례 표시하기

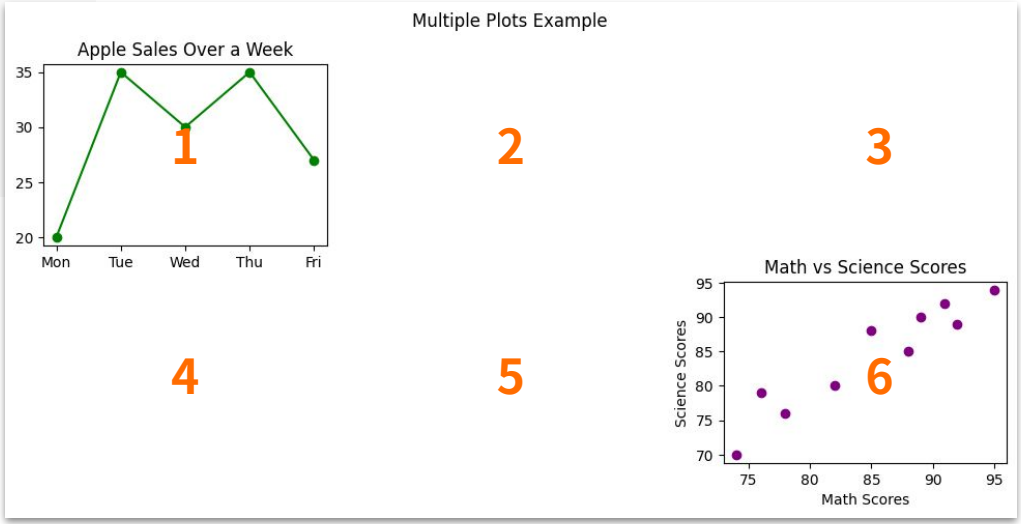
```
year = [2018, 2019, 2020, 2021, 2022, 2023]
stockA = [324, 234, 233, 345, 342, 452]
stockB = [10, 639, 594, 577, 694, 345]
stockC = [583, 482, 393, 495, 293, 212]

plt.plot(year, stockA)
plt.plot(year, stockB)
plt.plot(year, stockC)
plt.legend(['Stock A', 'Stock B', 'Stock C'])
plt.xlabel('Year')
plt.ylabel('Stock Price')
plt.show()
```



subplots로 하나의 fig에 여러 plot 그리기

```
plt.figure(figsize=(12, 5))

plt.subplot(2, 3, 1)
plt.plot(days, apple_sales, marker='o', color='green')
plt.title('Apple Sales Over a Week')

plt.subplot(2, 3, 6)
plt.scatter(math_scores, science_scores, color='purple')
plt.xlabel('Math Scores')
plt.ylabel('Science Scores')
plt.title('Math vs Science Scores')

plt.suptitle('Multiple Plots Example')
plt.show()
```

fig 내 행 개수, fig 내 열 개수, subplot의 인덱스



Woka

3. seaborn

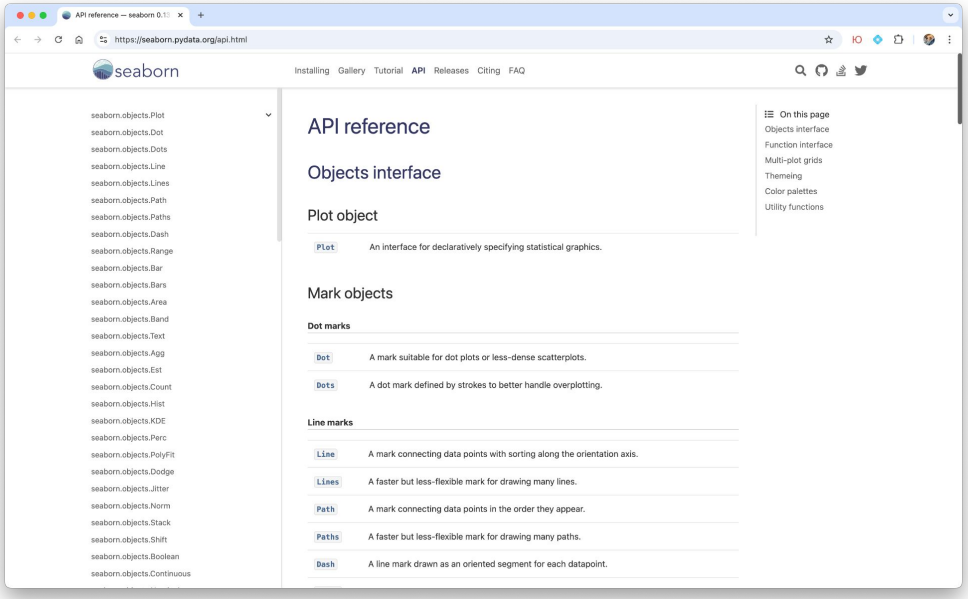
seaborn

matplotlib을 기반으로 한 고급 시각화 라이브러리

- 데이터프레임을 직접 사용해서 그래프 시각화
- 그래프와 색상 테마의 세분화
- 통계적 데이터 시각화에 강점

https://seaborn.pydata.org/api.html

```
[22] import seaborn as sns
```



Bar Plot

DataFrame의 데이터를 직접 사용해서 범주형 데이터 시각화

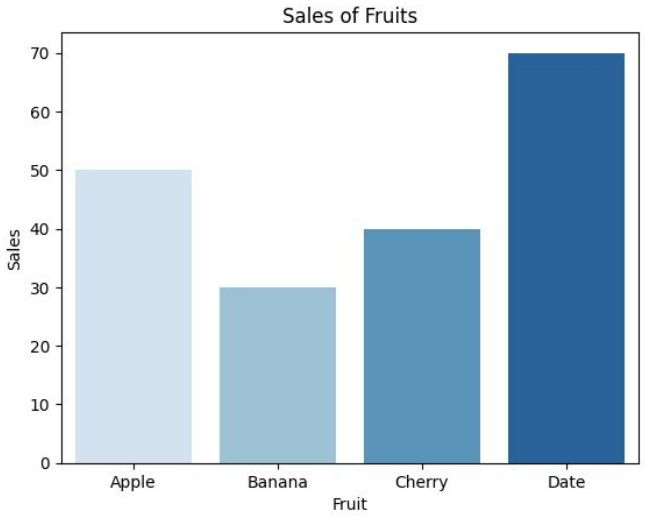
```
seaborn.barplot(data=None, *, x=None, y=None, hue=None, order=None,
hue_order=None, estimator='mean', errorbar=('ci', 95), n_boot=1000, seed=None,
units=None, weights=None, orient=None, color=None, palette=None, saturation=0.75,
fill=True, hue_norm=None, width=0.8, dodge='auto', gap=0, log_scale=None,
native_scale=False, formatter=None, legend='auto', capsize=0, err_kws=None,
ci=<deprecated>, errcolor=<deprecated>, errwidth=<deprecated>, ax=None, **kwargs) #
```

- ***data***: DataFrame, Series, dict, array, list of arrays
- ***x***, ***y***, ***hue***: names of variables in ***data***

Bar Plot

과일별 판매량 분포

```
df = pd.DataFrame({
    'Fruit': ['Apple', 'Banana', 'Cherry', 'Date'],
    'Sales': [50, 30, 40, 70]
})

sns.barplot(x='Fruit', y='Sales', data=df, hue='Fruit', palette='Blues')
plt.title('Sales of Fruits')
plt.show()
```



Scatter Plot

두 연속형 변수 간의 관계 시각화

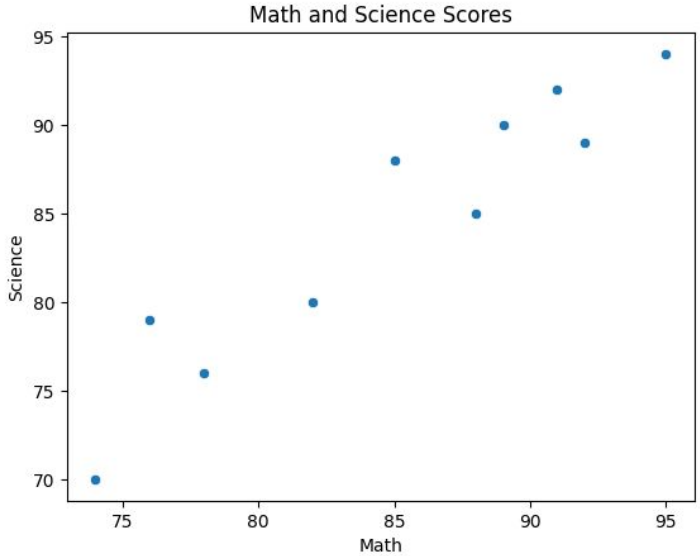
```
seaborn.scatterplot(data=None, *, x=None, y=None, hue=None, size=None, style=None,
palette=None, hue_order=None, hue_norm=None, sizes=None, size_order=None,
size_norm=None, markers=True, style_order=None, legend='auto', ax=None, **kwargs)
```

- ***data***: pandas.DataFrame, numpy.ndarray, mapping, sequence
- ***x***, ***y***, ***hue***: vectors or keys in ***data***

Scatter Plot

학생들의 수학/과학 점수의 분포

```python
math_scores = [85, 78, 92, 88, 76, 95, 89, 74, 91, 82]
science_scores = [88, 76, 89, 85, 79, 94, 90, 70, 92, 80]

df = pd.DataFrame({
    'Math': math_scores,
    'Science': science_scores
})

sns.scatterplot(x='Math', y='Science', data=df)
plt.title('Math and Science Scores')
plt.show()
```



Line Plot

데이터의 시간에 따른 변화

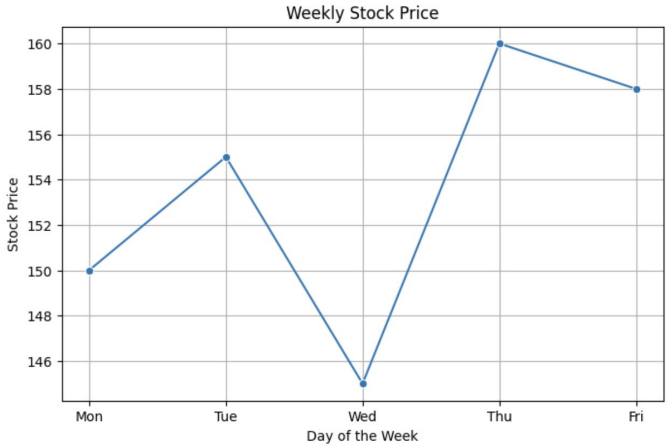
```
seaborn.lineplot(data=None, *, x=None, y=None, hue=None, size=None, style=None,
units=None, weights=None, palette=None, hue_order=None, hue_norm=None, sizes=None,
size_order=None, size_norm=None, dashes=True, markers=None, style_order=None,
estimator='mean', errorbar=('ci', 95), n_boot=1000, seed=None, orient='x',
sort=True, err_style='band', err_kws=None, legend='auto', ci='deprecated', ax=None,
**kwargs) #
```

- ***data***: pandas.DataFrame, numpy.ndarray, mapping, sequence
- ***x***, ***y***, ***hue***: vectors or keys in ***data***
- ***markers***: boolean, list, dictionary

Line Plot

일주일 동안 주식 가격의 변화

```
[33] data = {
         'Day': ['Mon', 'Tue', 'Wed', 'Thu', 'Fri'],
         'Stock Price': [150, 155, 145, 160, 158]
     }

     df = pd.DataFrame(data)

     plt.figure(figsize=(8, 5))
     sns.lineplot(x='Day', y='Stock Price', data=df, marker='o')
     plt.title('Weekly Stock Price')
     plt.xlabel('Day of the Week')
     plt.ylabel('Stock Price')
     plt.grid(True)
     plt.show()
```



Histplot

Woka

데이터의 분포

```
seaborn.histplot(data=None, *, x=None, y=None, hue=None, weights=None,
stat='count', bins='auto', binwidth=None, binrange=None, discrete=None,
cumulative=False, common_bins=True, common_norm=True, multiple='layer',
element='bars', fill=True, shrink=1, kde=False, kde_kws=None, line_kws=None,
thresh=0, pthresh=None, pmax=None, cbar=False, cbar_ax=None, cbar_kws=None,
palette=None, hue_order=None, hue_norm=None, color=None, log_scale=None,
legend=True, ax=None, **kwargs)
```

- ***data***: pandas.DataFrame, numpy.ndarray, mapping, or sequence
- ***x***, ***y***: vectors of keys in ***data***
- ***bins***: str, number

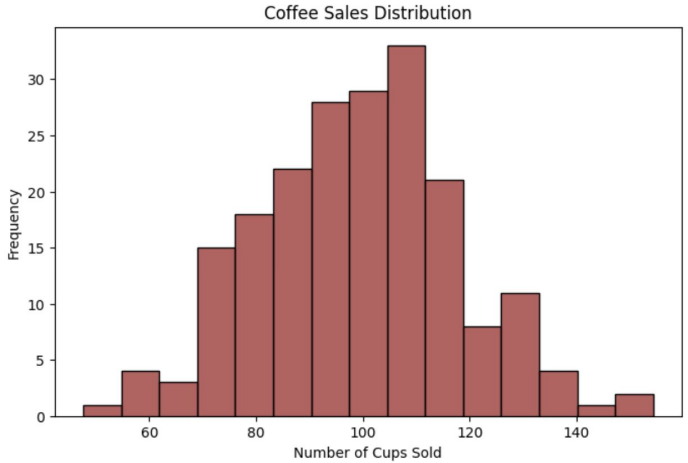
https://seaborn.pydata.org/generated/seaborn.histplot.html

Histplot

Woka

카페의 일간 커피 판매량 분포

```
[36] # 카페에서 하루 동안 판매된 커피 컵 수
     np.random.seed(42)
     coffee_sales = np.random.normal(100, 20, 200)  # 평균 100, 표준편차 20, 200개 데이터

     # Seaborn의 histplot 사용
     plt.figure(figsize=(8, 5))
     sns.histplot(coffee_sales, bins=15, color='brown')
     plt.title('Coffee Sales Distribution')
     plt.xlabel('Number of Cups Sold')
     plt.ylabel('Frequency')
     plt.show()
```



Boxplot

데이터의 분포(중앙값, 사분위수, 최소값, 최대값, 이상치) 시각화

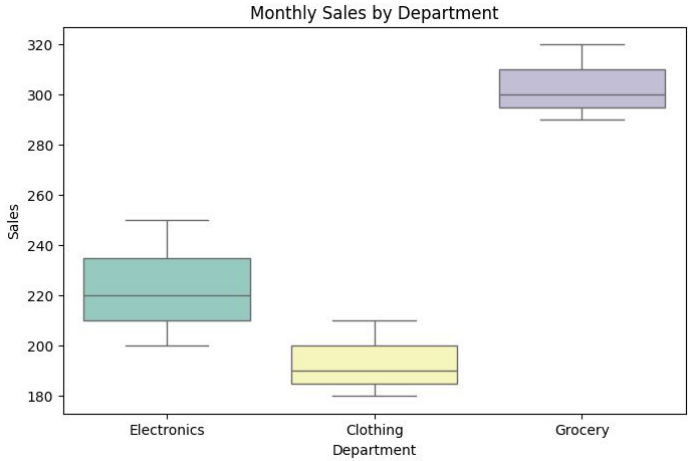
```
seaborn.boxplot(data=None, *, x=None, y=None, hue=None, order=None,
hue_order=None, orient=None, color=None, palette=None, saturation=0.75, fill=True,
dodge='auto', width=0.8, gap=0, whis=1.5, linecolor='auto', linewidth=None,
fliersize=None, hue_norm=None, native_scale=False, log_scale=None, formatter=None,
legend='auto', ax=None, **kwargs) #
```

- ***data***: pandas.DataFrame, numpy.ndarray, mapping, or sequence
- ***x***, ***y***: vectors of keys in ***data***

Boxplot

각 부서별 월간 판매량 비교

```
[38] # 부서별 월간 판매량
     data = {
         'Department': ['Electronics', 'Electronics', 'Electronics',
                        'Clothing', 'Clothing', 'Clothing',
                        'Grocery', 'Grocery', 'Grocery'],
         'Monthly Sales': [200, 220, 250,
                           180, 190, 210,
                           300, 320, 290]
     }

     # DataFrame 생성
     df = pd.DataFrame(data)

     # Seaborn의 boxplot 사용
     plt.figure(figsize=(8, 5))
     sns.boxplot(x='Department', y='Monthly Sales', data=df, hue='Department', palette='Set3')
     plt.title('Monthly Sales by Department')
     plt.xlabel('Department')
     plt.ylabel('Sales')
     plt.show()
```



Heatmap

Woka

변수 간의 상관관계(상관계수) 시각화

- 상관계수: 두 변수 간의 선형적 관계의 강도와 방향을 나타내는 통계적 지표 (값: -1~1)

```
seaborn.heatmap(data, *, vmin=None, vmax=None, cmap=None, center=None,
robust=False, annot=None, fmt='.2g', annot_kws=None, linewidths=0,
linecolor='white', cbar=True, cbar_kws=None, cbar_ax=None, square=False,
xticklabels='auto', yticklabels='auto', mask=None, ax=None, **kwargs) #
```

- ***data***: rectangular(2차원) dataset
- ***annot***: bool

Heatmap

주택 가격 데이터에서 변수들 간의 상관관계

```python
# 주택 가격 데이터
np.random.seed(42)
data = {
    'Size (sqft)': np.random.randint(500, 3500, 100),
    'Bedrooms': np.random.randint(1, 6, 100),
    'Age (years)': np.random.randint(0, 50, 100),
    'Price ($)': np.random.randint(100000, 500000, 100)
}

# DataFrame 생성
df = pd.DataFrame(data)

# 변수 간의 상관계수 계산
corr_matrix = df.corr()

# Seaborn의 heatmap 사용
plt.figure(figsize=(8, 5))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap of House Features')
plt.show()
```