

Data Analysis & Visualization: Part II

📅 Date	@2022년 9월 1일 오전 9:30
🔗 Code: Lecture	C2-004_Lecture_05.ipynb
🔗 Code: Practice	C2-004_Practice_05.ipynb
🔗 Code: Practice-Sol	C2-004_Practice_05-Sol.ipynb
🔗 Lecture Note	
# No.	5

Matplotlib

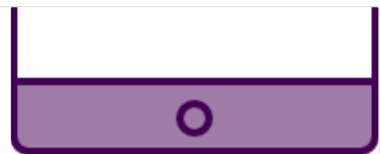
- Library for data visualization using various types of graphs in Python

```
import matplotlib
import matplotlib.pyplot as plt
```

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Try Matplotlib (on Binder) Be sure to check the Users guide and the API docs. The full text search is a good way to discover the docs including the many examples.

 <https://matplotlib.org/>



▼ 1. 기본 (꺾은선) 그래프: plot

- 기본 그래프: `plt.plot`

```
x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

plt.plot(x, y)
```

- 제목 설정: `plt.title`

```
plt.plot(x, y)
plt.title('Line Graph')
```

- 축 설정: `plt.xlabel`, `plt.ylabel`

```
# X축, Y축 제목 설정
plt.plot(x, y)
plt.title('Line Graph')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

plt.show()
```

```
# X축, Y축 제목 색상 설정
plt.plot(x, y)
plt.title('Line Graph')
plt.xlabel('X-axis', color='red')
plt.ylabel('Y-axis', color='#00aa00') # green

plt.show()
```

```
# X축, Y축 간격 표시 설정
plt.plot(x, y)
plt.title('Line Graph')
plt.xlabel('X-axis', color='red')
plt.ylabel('Y-axis', color='#00aa00')


plt.xticks([1, 3, 5, 7, 9])
plt.yticks([5, 10, 15, 20, 25, 30])
```

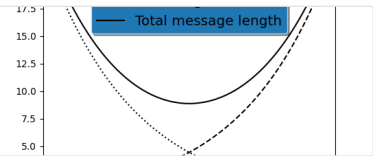
```
plt.show()
```

- 범례 설정: `plt.legend`

matplotlib.pyplot.legend - Matplotlib 3.5.3 documentation

matplotlib.pyplot.legend (* args , ** kwargs) [source] Place a legend on the Axes. Call signatures: The call signatures correspond to the following different ways to use this method: 1. Automatic detection of elements to be shown in the legend The elements to be added to the legend are automatically determined, when you do not pass in

 https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.legend.html



```
# Legend 표시 설정
plt.plot(x, y, label = 'Data1')
plt.legend()

plt.show()
```

```
# Legend 위치 설정1

plt.plot(x, y, label = 'Data1')
plt.legend(loc='lower right')
# plt.legend(loc='upper right')
# plt.legend( loc='best' )

plt.show()
```

```
# Legend 위치 설정2

plt.plot(x, y, label = 'Data1')
plt.legend(loc=(0.5, 0.7)) # x축, y축 좌표로 표시 [0, 1]

plt.show()
```

▼ 2. 그래프 스타일

- 그래프 선 굵기: `linewidth`

```
import matplotlib
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

# 그래프 선의 굵기
plt.plot(x, y, linewidth=5)
```

- 마커 설정: `marker`

matplotlib.markers - Matplotlib 3.5.3 documentation

Functions to handle markers; used by the marker functionality of plot , scatter , and errorbar . All possible markers are defined here: marker symbol description "None" , " " or "" nothing A list of (x, y) pairs used for Path vertices.

 https://matplotlib.org/stable/api/markers_api.html?highlight=marker#module-matplotlib.markers

```
# 마커 설정1

plt.plot(x, y, marker='o') # 마커 표시
plt.plot(x, y, marker='v') # 마커 표시

# 마커 표시 & 선 스타일은 대쉬
plt.plot(x, y, marker='o', linestyle='--')

# 마커 표시 & 마커 크기 설정
plt.plot(x, y, marker='X', markersize=10)

plt.show()
```

```
# 마커 설정2

# 마커 테두리 색 설정
```

```
plt.plot(x, y, marker='*', markersize=20, markeredgcolor='red')

# 마커 색 채우기 설정
plt.plot(x, y, marker='*', markersize=20, markeredgcolor='red', markerfacecolor='yellow')

plt.show()
```

- 선 스타일: linestyle

Linestyles - Matplotlib 3.5.3 documentation

 https://matplotlib.org/stable/gallery/lines_bars_and_markers/linestyles.html?highlight=linestyle

 https://matplotlib.org/stable/gallery/lines_bars_and_markers/linestyles.html?highlight=linestyle

```
# 선 스타일 설정1: 선 종류

plt.plot(x, y, linestyle=':')
plt.plot(x, y, linestyle='-.')
```



```
plt.show()
```

```
# 선 스타일 설정1: 선 종류

plt.plot(x, y, linestyle=':')
plt.plot(x, y, linestyle='-.')
```



```
plt.show()
```

List of named colors - Matplotlib 3.5.3 documentation

https://matplotlib.org/stable/gallery/color/named_colors.html

List of named colors - Matplotlib 3.5.3 documentation

https://matplotlib.org/stable/gallery/color/named_colors.html

```
# 선 스타일 설정2: 색상

plt.plot(x, y, color='k')
plt.plot(x, y, color='m')

plt.show()
```

```
# 선 스타일 설정2: 색상

plt.plot(x, y, color='k')
plt.plot(x, y, color='m')

plt.show()
```

- 다양한 스타일 예제

```
# 스타일 포맷: color, marker, linestyle

plt.plot(x, y, 'ro-') # color, marker, linestyle
plt.plot(x, y, 'bv--') # color, marker, linestyle
plt.plot(x, y, color='b', marker='v', linestyle='--') # color, marker, linestyle

plt.show()
```

```
# 스타일 포맷: color, marker, linestyle

plt.plot(x, y, 'ro-') # color, marker, linestyle
plt.plot(x, y, 'bv--') # color, marker, linestyle
plt.plot(x, y, color='b', marker='v', linestyle='--') # color, marker, linestyle

plt.show()
```

```
# 스타일 포맷2: 축약어
# https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html

plt.plot(x, y, marker='o', mfc='red', ms=10, mec='b', ls=':')
# plt.plot(x, y, marker='o', markerfacecolor='red', markersize=10, markeredgecolor='b', linestyle=':')

plt.show()
```

```
# 스타일 포맷2: 축약어
# https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html

plt.plot(x, y, marker='o', mfc='red', ms=10, mec='b', ls=':')
# plt.plot(x, y, marker='o', markerfacecolor='red', markersize=10, markeredgecolor='b', linestyle=':')

plt.show()
```

```
# 스타일 포맷2: 축약어
# https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html

plt.plot(x, y, marker='o', mfc='red', ms=10, mec='b', ls=':')
# plt.plot(x, y, marker='o', markerfacecolor='red', markersize=10, markeredgecolor='b', linestyle=':')

plt.show()
```

```
# 투영도
plt.plot(x, y, marker='o', mfc='red', ms=10, mec='b', ls=':', alpha = 0.3)

plt.show()
```

```
# 투영도
plt.plot(x, y, marker='o', mfc='red', ms=10, mec='b', ls=':', alpha = 0.3)

plt.show()
```

```
# 그래프 크기 설정1
plt.figure(figsize=(10, 5))
plt.plot(x, y)

plt.show()

# 그래프 크기 설정2: 해상도 조절
plt.figure(figsize=(10, 5), dpi=200) # dpi: dot per inch
plt.plot(x, y)

plt.show()
```

```
# 그래프 크기 설정1
plt.figure(figsize=(10, 5))
plt.plot(x, y)

plt.show()

# 그래프 크기 설정2: 해상도 조절
plt.figure(figsize=(10, 5), dpi=200) # dpi: dot per inch
plt.plot(x, y)

plt.show()
```

```
# 배경색
plt.figure(facecolor='yellow')
plt.plot(x, y)

plt.show()
```

▼ 3. 그래프 파일 저장

```
# 그래프 파일 저장 예제1

import matplotlib
import matplotlib.pyplot as plt

dir = '/content/drive/MyDrive/Colab Notebooks/C2-004_Python/Lecture_05/'

x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

plt.plot(x, y)
plt.savefig(dir + 'graph.png', dpi=100)
```

```
# 그래프 파일 저장 예제2

import matplotlib
import matplotlib.pyplot as plt

dir = '/content/drive/MyDrive/Colab Notebooks/C2-004_Python/Lecture_05/'

x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

plt.figure(dpi=200)
plt.plot(x, y)
plt.savefig(dir + 'graph2.png', dpi=100) # 파일 저장 시에만 해상도 따로 조정 가능
```

▼ 4. 데이터 표시 방법

- 텍스트 형태의 데이터 라벨링: `plt.text`

```
# 데이터 라벨링 표시: plt.text()

import matplotlib
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

plt.plot(x, y, marker='o')

for idx, txt in enumerate(y):
    plt.text(x[idx], y[idx], txt)

plt.show()
```

```
# 데이터 라벨링 위치조정

plt.plot(x, y, marker='o')

for idx, txt in enumerate(y):
    plt.text(x[idx], y[idx]+0.5, txt, ha='center', color='b') # horizontal alignment, color

plt.show()
```

- 여러 데이터를 표시하는 그래프

```
# 여러 데이터 표시 그래프
# COVID-19 백신 종류별 접종 인구 수

import matplotlib
import matplotlib.pyplot as plt

days = [1, 2, 3] # 1일, 2일, 3일
az = [2, 4, 8] # [단위: 만명] 아스트라제네카 접종 인구 수
pf = [3, 1, 6] # [단위: 만명] 화이자 접종 인구 수
md = [1, 3, 9] # [단위: 만명] 모더나 접종 인구 수
```

```
plt.plot(days, az)
plt.plot(days, pf)
plt.plot(days, md)

plt.show()
```

```
# 여러 데이터 표시 그래프 개선
# COVID-19 백신 종류별 접종 인구 수

days = [1, 2, 3] # 1일, 2일, 3일
az = [2, 4, 8] #[단위: 만명] 아스트라제네카 접종 인구 수
pf = [3, 1, 6] #[단위: 만명] 화이자 접종 인구 수
md = [1, 3, 9] #[단위: 만명] 모더나 접종 인구 수

plt.plot(days, az, label='az', marker='*',)
plt.plot(days, pf, label='pfizer', marker='o', linestyle='--')
plt.plot(days, md, label='modena', marker='s', linestyle='-.')
plt.legend()
# plt.legend(ncol=3)

plt.show()
```

▼ 5. 데이터프레임(DataFrame) 이용한 그래프

- 데이터프레임 파일 불러오기

```
import pandas as pd

dir = '/content/drive/MyDrive/Colab Notebooks/C2-004_Python/Lecture_04/'

df = pd.read_excel(dir + "onepiece.xlsx", index_col="역할")
# df
```

- 한글폰트 지정

```
import matplotlib
import matplotlib.pyplot as plt

# 한글폰트 지정
matplotlib.rcParams['font.family'] = 'NanumBarunGothic' # 나눔바른고딕 한글폰트 적용
matplotlib.rcParams['font.size'] = 12
matplotlib.rcParams['axes.unicode_minus'] = False # 한글폰트 사용 시, 마이너스 부호 깨짐현상 방지
```

- 데이터프레임(DataFrame) 데이터를 그래프로 표시하기

```
# DataFrame 데이터를 그래프로 표시

plt.plot(df['이름'], df['방어'])

plt.show()
```

```
# 여러 DataFrame 데이터 그래프로 표시

plt.plot(df['이름'], df['방어'])
plt.plot(df['이름'], df['지력'])

plt.grid(color='r', alpha=0.5, ls=':', linewidth=2)
plt.grid() # axis='x', axis='y'

plt.show()
```

▼ 6. 다양한 그래프: 막대 그래프

- 막대 그래프 기본

```
# 막대 그래프: bar chart

import matplotlib
import matplotlib.pyplot as plt

labels = ['Luffy', 'Zoro', 'Nami'] # 이름
values = [100, 95, 90] # 공격력
```

```
plt.bar(labels, values)

plt.show()
```

```
# 막대 그래프: 색상 설정 - color

labels = ['Luffy', 'Zoro', 'Nami'] # 이름
values = [100, 95, 90] # 공격력
colors = ['r', 'g', 'b'] # 색상 지정

plt.bar(labels, values, color=colors, alpha=0.5)

plt.show()
```

```
# 막대 그래프: 축 범위 설정 - xlim/ylim

labels = ['Luffy', 'Zoro', 'Nami'] # 이름
values = [100, 95, 90] # 공격력

plt.bar(labels, values)
plt.ylim(85, 105) # 축 범위 설정

plt.show()
```

```
# 막대 그래프: 막대 두께 설정 - width

plt.bar(labels, values, width=0.5)

plt.show()
```

```
# 막대 그래프: 축 제목 회전 - xticks/yticks(rotation)

plt.bar(labels, values, width=0.5)

# x축의 이름 데이터 각도를 45도 회전
plt.xticks(rotation=45)

# y축의 공격력 데이터 각도를 45도 회전
plt.yticks(rotation=45)

plt.show()
```

```
# 막대 그래프: 정의한 데이터 이름으로 출력 - xticks/yticks

labels = ['Luffy', 'Zoro', 'Nami'] # 이름
values = [100, 95, 90] # 공격력
ticks = ['Member1', 'Member2', 'Member3']

plt.bar(labels, values)
plt.xticks(labels, ticks)

plt.show()
```

• 막대 그래프 심화

```
# 막대 그래프 - 수평방향: horizontal bar chart

import matplotlib
import matplotlib.pyplot as plt

labels = ['Luffy', 'Zoro', 'Nami'] # 이름
values = [100, 95, 90] # 공격력

plt.barh(labels, values)
# plt.xlim(85, 105)

plt.show()
```

```
# 막대 그래프: 내부 패턴 채우기
# https://matplotlib.org/stable/gallery/shapes_and_collections/hatch_style_reference.htmlbar = plt.bar(labels, values)

bar[0].set_hatch('/') # ///
bar[1].set_hatch('x') # xxx
bar[2].set_hatch('..') # ...

plt.show()
```

```
# 막대 그래프: 데이터 레이블 추가

bar = plt.bar(labels, values)
plt.ylim(0, 120)

for idx, rect in enumerate(bar):
    plt.text(idx, rect.get_height(), values[idx])
    # plt.text(idx, rect.get_height()+0.8, values[idx], ha='center', color='b')

plt.show()
```

▼ 7. 다양한 그래프: 누적막대 그래프

- DataFrame 데이터 불러오기

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

# 한글폰트 지정
matplotlib.rcParams['font.family'] = 'NanumBarunGothic' # 나눔바른고딕 한글폰트 적용
matplotlib.rcParams['font.size'] = 12
matplotlib.rcParams['axes.unicode_minus'] = False # 한글폰트 사용 시, 마이너스 부호 깨짐현상 방지

dir = '/content/drive/MyDrive/Colab Notebooks/C2-004_Python/Lecture_04/'

df = pd.read_excel(dir + "onepice.xlsx", index_col="역할")
# df
```

- 누적막대 그래프의 필요성

```
# 누적 막대 그래프의 필요성

plt.bar(df['이름'], df['공격'])
plt.bar(df['이름'], df['지력'])

plt.show()
```

- 누적막대 그래프 표시 예제

```
# 누적 막대 그래프 표시1

plt.bar(df['이름'], df['공격'])
plt.bar(df['이름'], df['지력'], bottom=df['공격']) # 공격 데이터 위에 지력 데이터

plt.show()
```

```
# 누적 막대 그래프 표시2

plt.bar(df['이름'], df['공격'])
plt.bar(df['이름'], df['지력'], bottom=df['공격']) # 공격 데이터 위에 지력 데이터
plt.bar(df['이름'], df['방어'], bottom=df['공격'] + df['지력']) # "공격 + 지력" 데이터 위에 지력 데이터

plt.show()
```

```
# 누적 막대 그래프 표시3: legend 추가

plt.bar(df['이름'], df['공격'], label='공격')
plt.bar(df['이름'], df['지력'], bottom=df['공격'], label='지력') # 공격 데이터 위에 지력 데이터
plt.bar(df['이름'], df['방어'], bottom=df['공격'] + df['지력'], label='방어') # "공격 + 지력" 데이터 위에 지력 데이터
plt.legend(loc='best')

plt.show()
```

▼ 8. 다양한 그래프: 다중막대 그래프

- DataFrame 데이터 불러오기

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

```
# 한글폰트 지정
matplotlib.rcParams['font.family'] = 'NanumBarunGothic' # 나눔바른고딕 한글폰트 적용
matplotlib.rcParams['font.size'] = 12
matplotlib.rcParams['axes.unicode_minus'] = False # 한글폰트 사용 시, 마이너스 부호 깨짐현상 방지

dir = '/content/drive/MyDrive/Colab Notebooks/C2-004_Python/Lecture_04/'

df = pd.read_excel(dir + "onepice.xlsx", index_col="역할")
# df
```

- 다중 막대 그래프

```
# 다중 막대 그래프 그리기 (미완성)

import numpy as np

idx = np.arange(df.shape[0]) # DataFrame row 크기 확인
w = 0.3
plt.bar(idx - w, df['공격'])
plt.bar(idx, df['방어'])
plt.bar(idx + w, df['지력'])

plt.show()
```

```
# 다중 막대 그래프 그리기 (완성)

import numpy as np

idx = np.arange(df.shape[0]) # DataFrame row 크기 확인
w = 0.3
plt.bar(idx - w, df['공격'], width = w)
plt.bar(idx, df['방어'], width = w)
plt.bar(idx + w, df['지력'], width = w)

plt.show()
```

```
# 다중 막대 그래프 그리기 (개선)

import numpy as np

plt.figure(figsize=(10, 5))
plt.title('캐릭터별 능력치')

idx = np.arange(df.shape[0]) # DataFrame row 크기 확인
w = 0.3
plt.bar(idx - w, df['공격'], width = w, label='공격')
plt.bar(idx, df['방어'], width = w, label='방어')
plt.bar(idx + w, df['지력'], width = w, label='지력')

plt.legend(ncol=3, loc='best')
plt.xticks(idx, df['이름'], fontsize=15, rotation=60)
plt.ylim(0, 120)

plt.grid(axis='y', alpha=0.7, ls=':')

plt.show()
```

▼ 9. 다양한 그래프: 원 그래프

- DataFrame 데이터 불러오기

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

# 한글폰트 지정
matplotlib.rcParams['font.family'] = 'NanumBarunGothic' # 나눔바른고딕 한글폰트 적용
matplotlib.rcParams['font.size'] = 12
matplotlib.rcParams['axes.unicode_minus'] = False # 한글폰트 사용 시, 마이너스 부호 깨짐현상 방지

dir = '/content/drive/MyDrive/Colab Notebooks/C2-004_Python/Lecture_04/'

df = pd.read_excel(dir + "onepice.xlsx", index_col="역할")
# df
```

- 원 그래프 기초: plt.pie


```
# 원 그래프 기초

values = [30, 25, 20, 13, 10, 2]
labels = ['Python', 'Java', 'Javascript', 'C#', 'C/C++', 'Etc.']

plt.pie(values)
plt.pie(values, labels=labels)
plt.pie(values, labels=labels, autopct='%1f%%')
plt.pie(values, labels=labels, autopct='%1f%%', startangle=90)
plt.pie(values, labels=labels, autopct='%1f%%', startangle=90, counterclock=False)

plt.show()
```

```
# 원 그래프 강조

values = [30, 25, 20, 13, 10, 2]
labels = ['Python', 'Java', 'Javascript', 'C#', 'C/C++', 'Etc.']

# explode = [0.2, 0.1, 0, 0, 0, 0]
explode = [0.05] * 6
plt.pie(values, labels=labels, explode=explode)

plt.show()
```

```
# 원 그래프: Legend 추가

values = [30, 25, 20, 13, 10, 2]
labels = ['Python', 'Java', 'Javascript', 'C#', 'C/C++', 'Etc.']

explode = [0.05] * 6

plt.pie(values, labels=labels, explode=explode)
plt.legend(loc=(1.2, 0.4))

plt.show()
```

```
# 원 그래프: Legend 및 제목 추가

values = [30, 25, 20, 13, 10, 2]
labels = ['Python', 'Java', 'Javascript', 'C#', 'C/C++', 'Etc.']

explode = [0.05] * 6

plt.title('언어별 선호도')
plt.pie(values, labels=labels, explode=explode)
plt.legend(loc=(1.2, 0.4), title='프로그래밍 언어')

plt.show()
```

• 원 그래프 심화

```
# 원 그래프: 색상 설정

values = [30, 25, 20, 13, 10, 2]
labels = ['Python', 'Java', 'Javascript', 'C#', 'C/C++', 'Etc.']
colors = ['#ffadad', '#ffd6a5', '#fdffb6', '#cafff9', '#9bf6ff', '#a0c4ff']
explode = [0.05] * 6

plt.pie(values, labels=labels, autopct='%1f%%', startangle=90, counterclock=False, colors=colors, explode=explode)

plt.show()
```

```
# wedgeprops 기능

wedgeprops = {'width':0.2}
wedgeprops = {'width':0.6}
wedgeprops = {'width':0.6, 'edgecolor':'w', 'linewidth':3}

plt.pie(values, labels=labels, autopct='%1f%%', startangle=90, counterclock=False, colors=colors, wedgeprops=wedgeprops)

plt.show()
```

```
# 디스플레이 정보 설정

def custom_autopct(pct):
    return ('%.0f%%' % pct) if pct >= 10 else ''
    # return '{:.1f}%'.format(pct) if pct >= 10 else ''

wedgeprops = {'width':0.6, 'edgecolor':'w', 'linewidth':3}
plt.pie(values, labels=labels, autopct=custom_autopct, startangle=90, counterclock=False, colors=colors, wedgeprops=wedgeprops)
```

```
plt.pie(values, labels=labels, autopct=custom_autopct, startangle=90, counterclock=False, colors=colors, wedgeprops=wedgeprops, pctdistance=0.7)

plt.show()
```

```
# DataFrame 데이터 활용하여 원 그래프 그리기

grp = df.groupby('출신')
# grp.size()['위대한항로']

values = [grp.size()['이스트블루'], grp.size()['노스블루'], grp.size()['위대한항로'], grp.size()['웨스트블루'], grp.size()['사우스블루']]
labels = ['이스트블루', '노스블루', '위대한항로', '웨스트블루', '사우스블루']

plt.pie(values, labels=labels)
plt.title('출신 바다')

plt.show()
```

```
# 데이터에 함수 적용: apply

def add_joul(attack):
    return str(attack) + ' Joul'

df['공격'] = df['공격'].apply(add_joul) # 공격 데이터에 대해서 add_foul 함수를 호출한 결과 데이터 적용
df
```

▼ 10. 다양한 그래프: 산포도 그래프

- DataFrame 데이터 불러오기

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

# 한글폰트 지정
matplotlib.rcParams['font.family'] = 'NanumBarunGothic' # 나눔바른고딕 한글폰트 적용
matplotlib.rcParams['font.size'] = 12
matplotlib.rcParams['axes.unicode_minus'] = False # 한글폰트 사용 시, 마이너스 부호 깨짐현상 방지

dir = '/content/drive/MyDrive/Colab Notebooks/C2-004_Python/Lecture_04/'

df = pd.read_excel(dir + "onepice.xlsx", index_col="역할")
# df
```

- 산포도 그래프: plt.scatter

```
# 산포도 그래프 예시

plt.scatter(df['공격'], df['민첩'])
plt.xlabel('공격 점수')
plt.ylabel('민첩 점수')

plt.show()
```

- 산포도 그래프 스타일 변화

```
# 데이터 점의 크기 설정1

import numpy as np
sizes = np.random.rand(10) * 1000

plt.scatter(df['공격'], df['민첩'], s=sizes)
plt.xlabel('공격 점수')
plt.ylabel('민첩 점수')

plt.show()
```

```
# 데이터 점의 크기 설정2
import numpy as np
sizes = df['나이'] * 70

plt.scatter(df['공격'], df['민첩'], s=sizes)
plt.xlabel('공격 점수')
plt.ylabel('민첩 점수')

plt.show()
```

```
# 데이터 점의 색깔 설정1
# https://matplotlib.org/stable/tutorials/colors/colormaps.html

plt.scatter(df['공격'], df['민첩'], s=sizes, c=df['나이'], cmap='viridis', alpha=0.3)
plt.xlabel('공격 점수')
plt.ylabel('민첩 점수')

plt.show()
```

```
# 데이터 점의 색깔 설정2
# https://matplotlib.org/stable/tutorials/colors/colormaps.html

plt.figure(figsize=(7, 7))

plt.scatter(df['공격'], df['민첩'], s=sizes, c=df['나이'], cmap='viridis', alpha=0.3)
plt.xlabel('공격 점수')
plt.ylabel('민첩 점수')

# plt.colorbar()
plt.colorbar(ticks=[7, 14, 21], label='나이', shrink=0.7, orientation='horizontal')

plt.show()
```

▼ 11. 여러 그래프 표시

- DataFrame 데이터 불러오기

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

# 한글폰트 지정
matplotlib.rcParams['font.family'] = 'NanumBarunGothic' # 나눔바른고딕 한글폰트 적용
matplotlib.rcParams['font.size'] = 12
matplotlib.rcParams['axes.unicode_minus'] = False # 한글폰트 사용 시, 마이너스 부호 깨짐현상 방지

dir = '/content/drive/MyDrive/Colab Notebooks/C2-004_Python/Lecture_04/'

df = pd.read_excel(dir + "onepice.xlsx", index_col="역할")
# df
```

- 여러 그래프 표시: plt.subplots

```
# Subplots을 이용한 여러 그래프 그리기

fig, axs = plt.subplots(2, 2, figsize=(15, 10)) # 2 x 3 그래프
fig.suptitle('여러 그래프 표시')

# 첫 번째 그래프
axs[0, 0].bar(df['이름'], df['공격'], label='공격 점수', color = 'purple')
axs[0, 0].set_title('캐릭터별 공격력 그래프')
axs[0, 0].legend()
axs[0, 0].set_xlabel='이름', ylabel='공격'
axs[0, 0].set_facecolor('lightyellow')
axs[0, 0].grid(ls='--', lw=0.5)

# 두 번째 그래프
axs[0, 1].plot(df['이름'], df['방어'], label='방어 점수', color = 'b')
axs[0, 1].set_title('캐릭터별 방어력 그래프')
axs[0, 1].legend()
axs[0, 1].set_xlabel='이름', ylabel='방어'
axs[0, 1].set_facecolor('lightblue')

# 세 번째 그래프
axs[1, 0].barh(df['이름'], df['지력'], label='지력 점수', color = 'g')
axs[1, 0].set_title('캐릭터별 지력 그래프')
axs[1, 0].legend()
axs[1, 0].set_xlabel='이름', ylabel='지력'
axs[1, 0].grid(ls='--', lw=0.5)

# 네 번째 그래프
axs[1, 1].plot(df['이름'], df['민첩'], label='민첩 점수', color = 'r', alpha=0.3, lw = 3)
axs[1, 1].set_title('캐릭터별 민첩도 그래프')
axs[1, 1].legend()
axs[1, 1].set_xlabel='이름', ylabel='민첩')

plt.show()
```