

⋮ 문제해결을 위한 데이터 분석 및 시각화

데이터시각화 기초

한성대학교 노은희 교수

“미래로 향하는 새로운 이정표”



오늘의 학습

학습내용

- Matplotlib 라이브러리를 이용하여 그래프 그리기



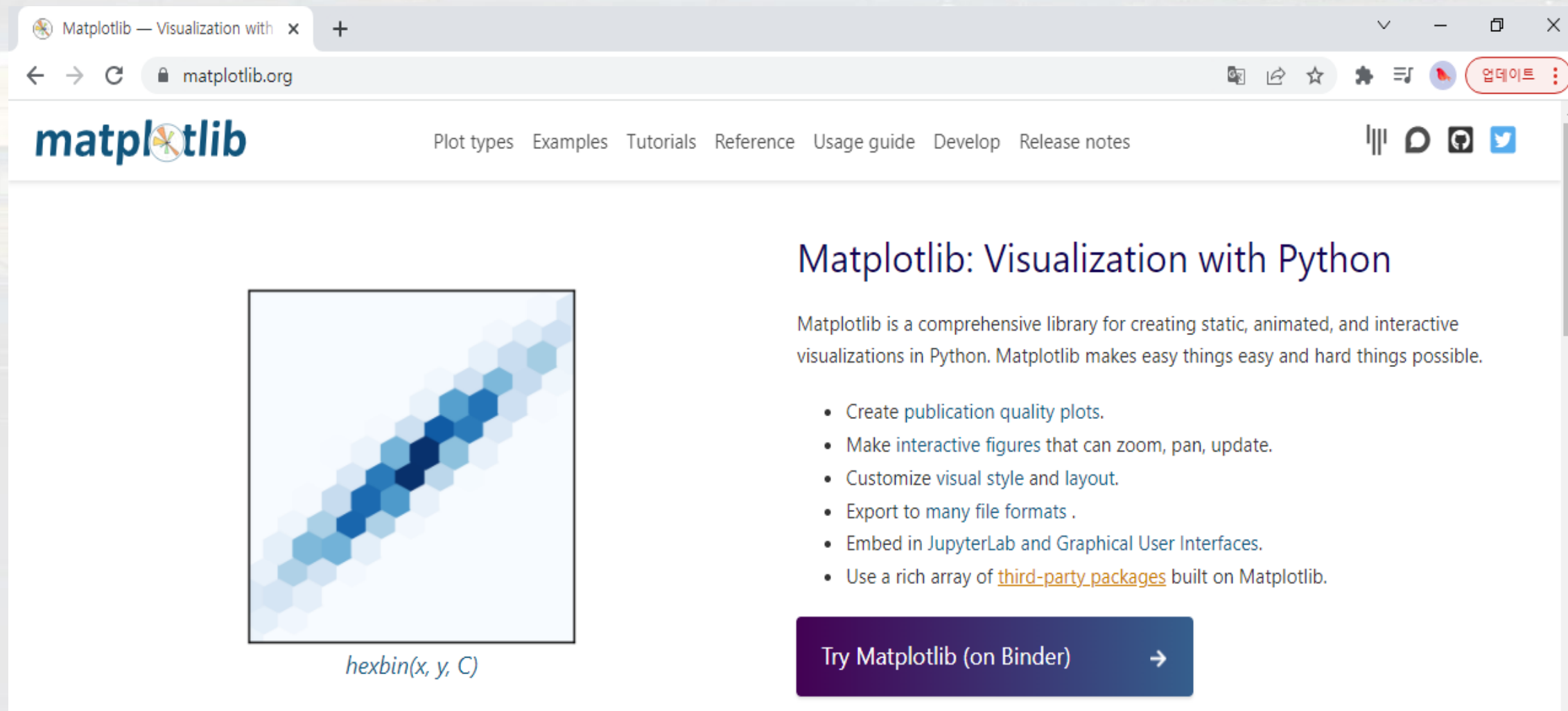
Plot

플롯(Plot) : 그래픽으로 랜더링하여 화면으로 출력하기 전에 데이터를 가상의 공간에 미리 그려놓는 것 `plt.show()`해야 차트가 시각화 되어 보이게 됨.


matplotlib라는 라이브러리

<https://matplotlib.org/>

데이터를 시각화해주는 가장 기본적인 라이브러리



The screenshot shows the Matplotlib website homepage. At the top, there's a navigation bar with the Matplotlib logo and links to Plot types, Examples, Tutorials, Reference, Usage guide, Develop, and Release notes. Below the navigation bar, the main heading reads "Matplotlib: Visualization with Python". To the left of the text is a hexbin plot showing a diagonal distribution of data points. To the right of the plot is a list of bullet points describing the library's capabilities. At the bottom right, there is a button labeled "Try Matplotlib (on Binder)" with a right-pointing arrow.

Matplotlib — Visualization with  x +

← → ↻ matplotlib.org 업데이트

matplotlib Plot types Examples Tutorials Reference Usage guide Develop Release notes

Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

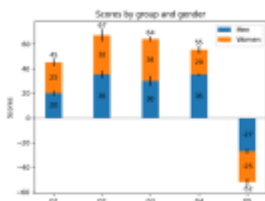
- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats .
- Embed in JupyterLab and Graphical User Interfaces.
- Use a rich array of [third-party packages](#) built on Matplotlib.

Try Matplotlib (on Binder) →

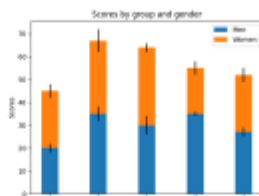
hexbin(x, y, C)

다양한 그래프

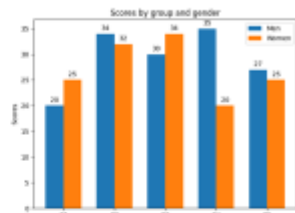
Lines, bars and markers



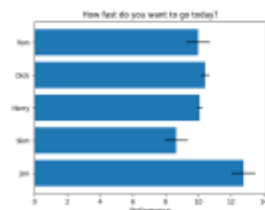
Bar Label Demo



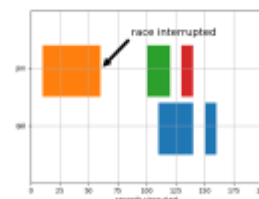
Stacked bar chart



Grouped bar chart with labels



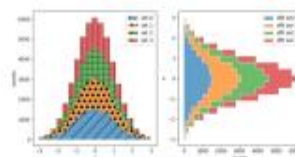
Horizontal bar chart



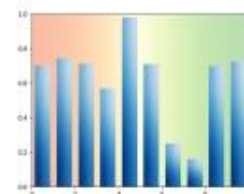
Broken Barh



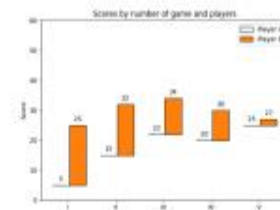
CapStyle



Hatch-filled histograms



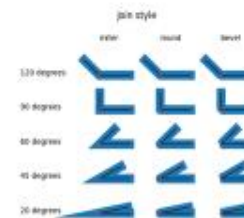
Bar chart with gradients



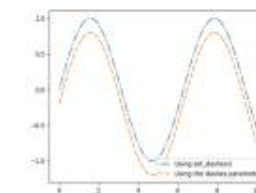
Hat graph



Discrete distribution as horizontal bar chart



JoinStyle



Customizing dashed line styles



Pyplot 라이브러리 불러오기

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

2D 데이터(X와 Y 값)를 기준으로 선이 이어지게 시각화

`plt.plot()`을 이용하여 기본적으로 `plot(x, y)` 순서로 2개의 리스트를 넣어 주지만,
만약 하나의 리스트만 넣을 경우 `x`는 자동적으로 0부터 할당

기본 그래프 그리기

Plot()함수는 직선이나 꺾은선 형태의 그래프를 그릴때 사용

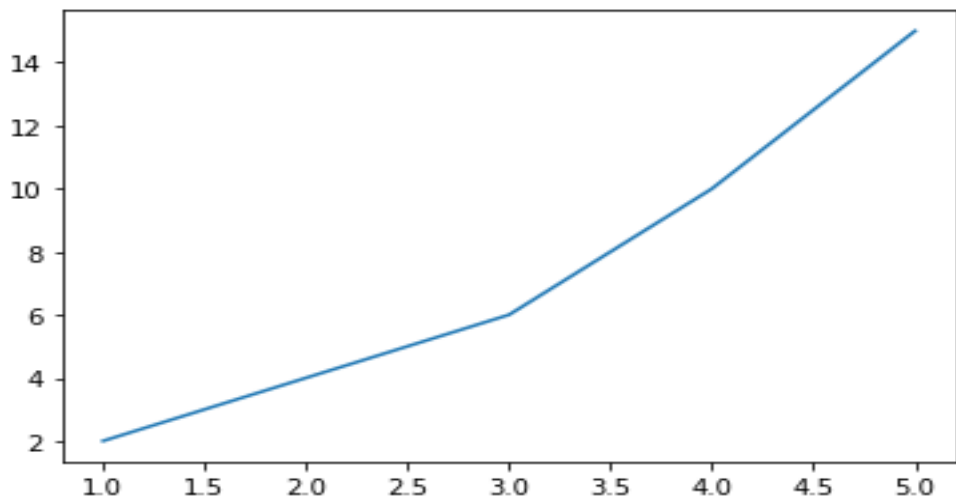
x축 값들

y축 값들

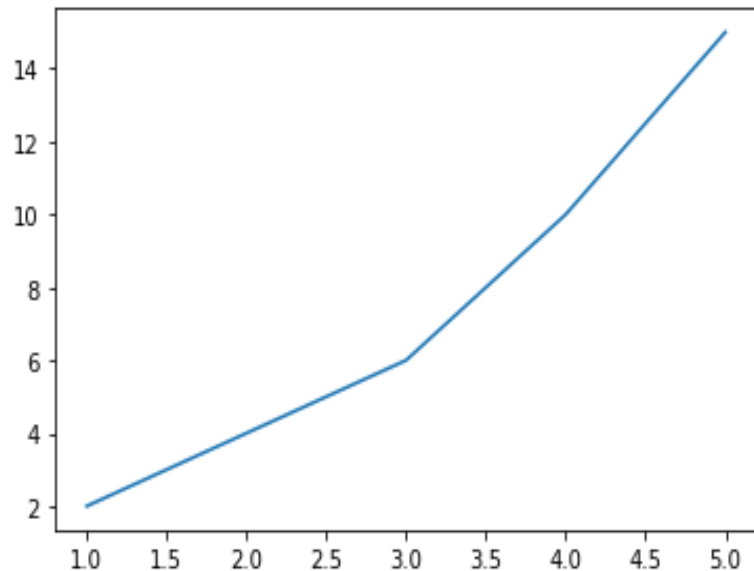
`plt.plot([1,2,3,4,5],[2,4,6,10,15])` 점 (1,2),(2,4),(3,6),(4,10),(5,15)을 잇는 꺾은선 그래프

```
1 import matplotlib.pyplot as plt
2 x = [1,2,3,4,5]
3 y = [2,4,6,10,15]
4 plt.plot(x,y)
```

[<matplotlib.lines.Line2D at 0x2501afbdf30>]



```
1 import matplotlib.pyplot as plt # 라이브러리 불러오기
2 plt.plot([1,2,3,4,5],[2,4,6,10,15]) # plot()함수에 x축, y축 데이터 입력하기
3 plt.show() # 그래프 보여주기
```



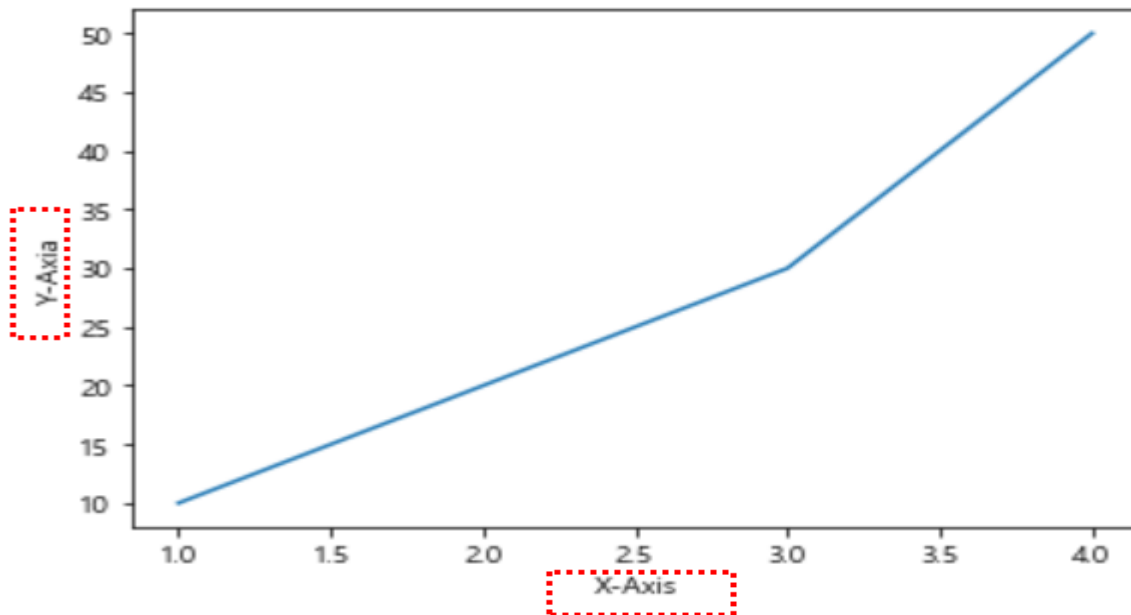
그래프 축 레이블 추가하기

축 레이블 추가하기

X,Y 축 레이블을 붙이기 위해서는 `plt.xlabel(축이름)`, `plt.ylabel(축이름)`

그래프와 제목과 x축, y축 레이블 넣기

```
1 import matplotlib.pyplot as plt
2 plt.xlabel('X-Axis')
3 plt.ylabel('Y-Axia')
4 plt.plot([1,2,3,4],[10,20,30,50])
5 plt.show()
```



그래프에 제목 넣기

그래프에 제목넣기

```
1 import matplotlib.pyplot as plt
2 plt.title('제목') #한글을 넣으면 글자가 깨짐
3 plt.xlabel('X-Axis')
4 plt.ylabel('Y-Axis')
5 plt.plot([1,2,3,4],[10,20,30,50])
6 plt.show()
```

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: RuntimeWarning: Glyph 51228 missing from current font.

font.set_text(s, 0.0, flags=flags)

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: RuntimeWarning: Glyph 47785 missing from current font.

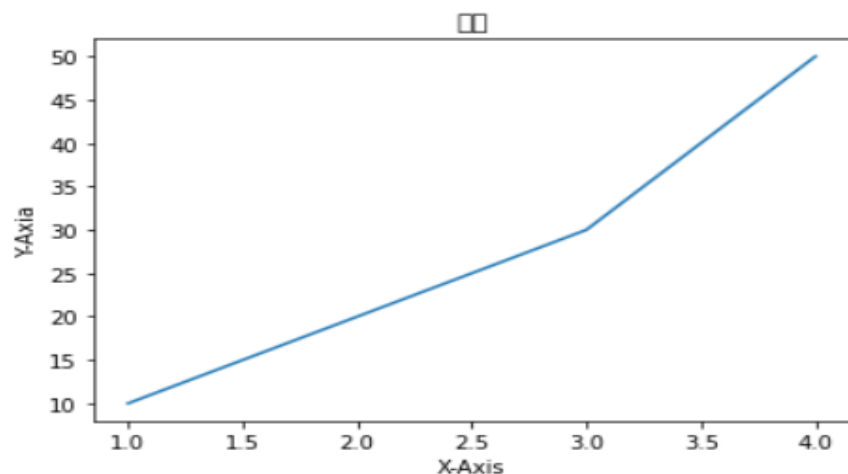
font.set_text(s, 0.0, flags=flags)

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: RuntimeWarning: Glyph 47785 missing from current font.

font.set_text(s, 0.0, flags=flags)

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: RuntimeWarning: Glyph 47785 missing from current font.

font.set_text(s, 0.0, flags=flags)



plt.title(제목) 등의 함수를 사용

한글을 넣고자 할 때는 한글 폰트를 설정하는 코드도 함께 작성

Windows 사용자

```
plt.rc('font', family = 'Malgun Gothic')
```

```
plt.title('제목' )
```

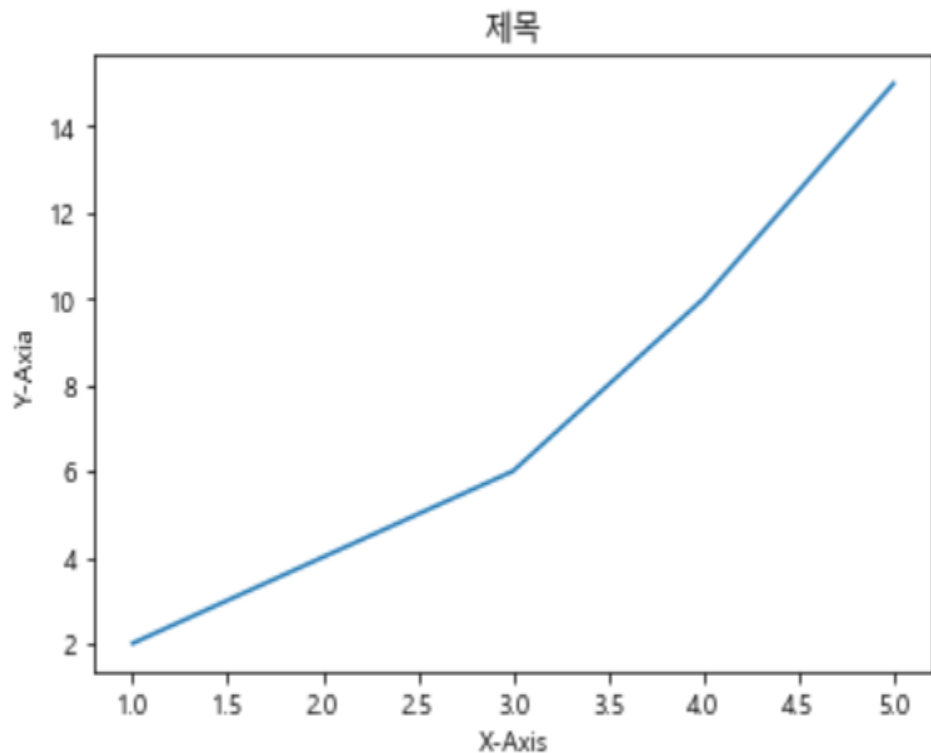
MacOS 사용자

```
plt.rc('font', family='AppleGothic')
```

```
plt.title('제목' )
```

제목에 한글깨짐 현상 해결방법

```
1 import matplotlib.pyplot as plt
2 plt.rc('font', family = 'Malgun Gothic') # Malgun Gothic은 '맑은 고딕'글꼴/ macOS 운영체제는 AppleGothic라고 쓰기
3 plt.title('제목')
4 plt.xlabel('X-Axis')
5 plt.ylabel('Y-Axis')
6 plt.plot([1,2,3,4,5], [2,4,6,10,15])
7 plt.show()
```

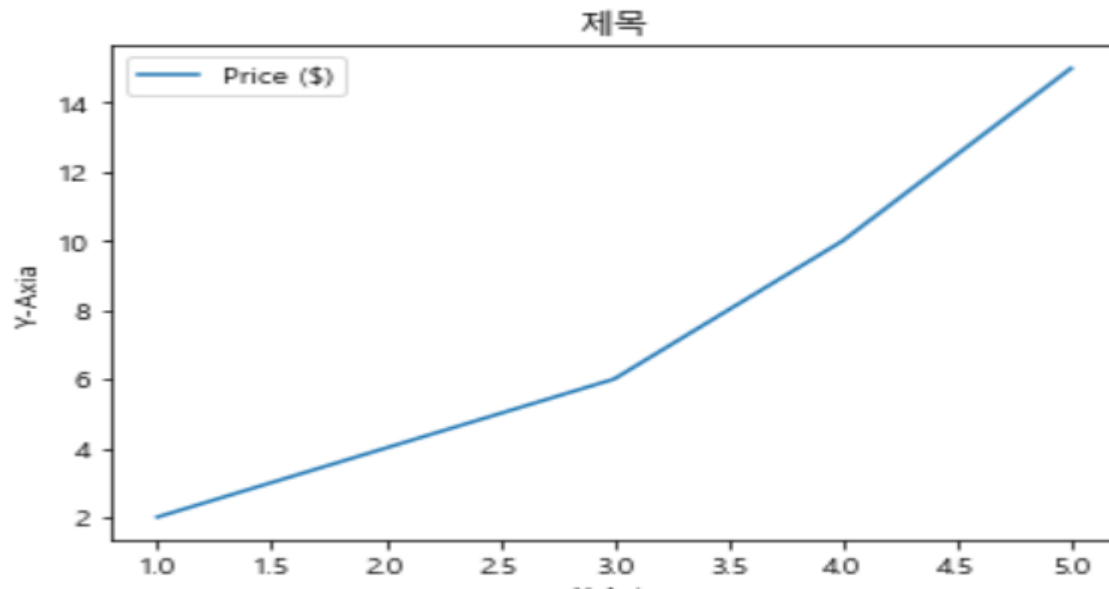


그래프에 범례 추가하기

범례 (Legend)는 그래프에 데이터의 종류를 표시하기 위한 텍스트

matplotlib.pyplot 모듈의 legend() 함수를 사용

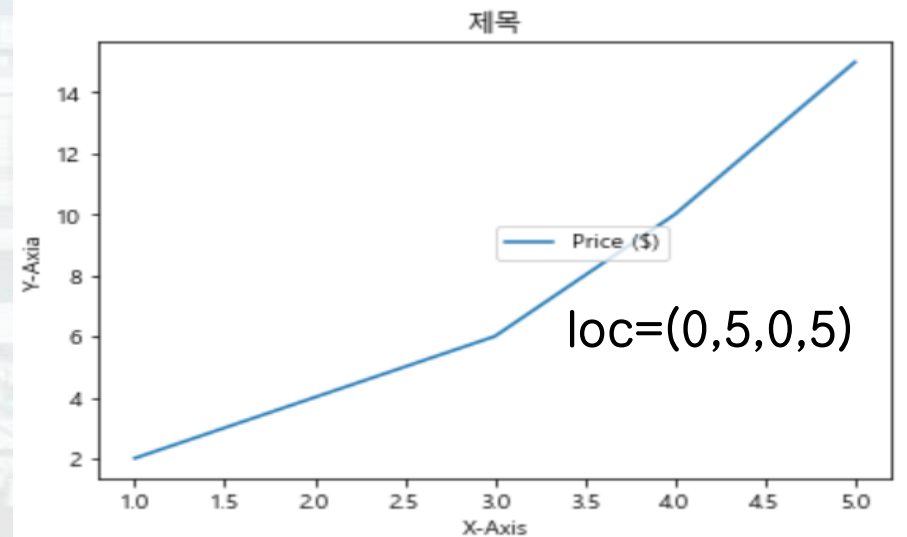
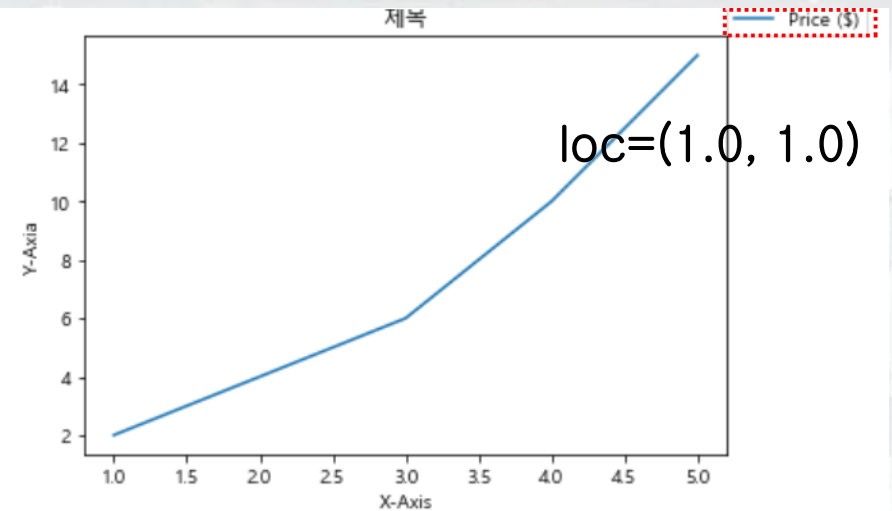
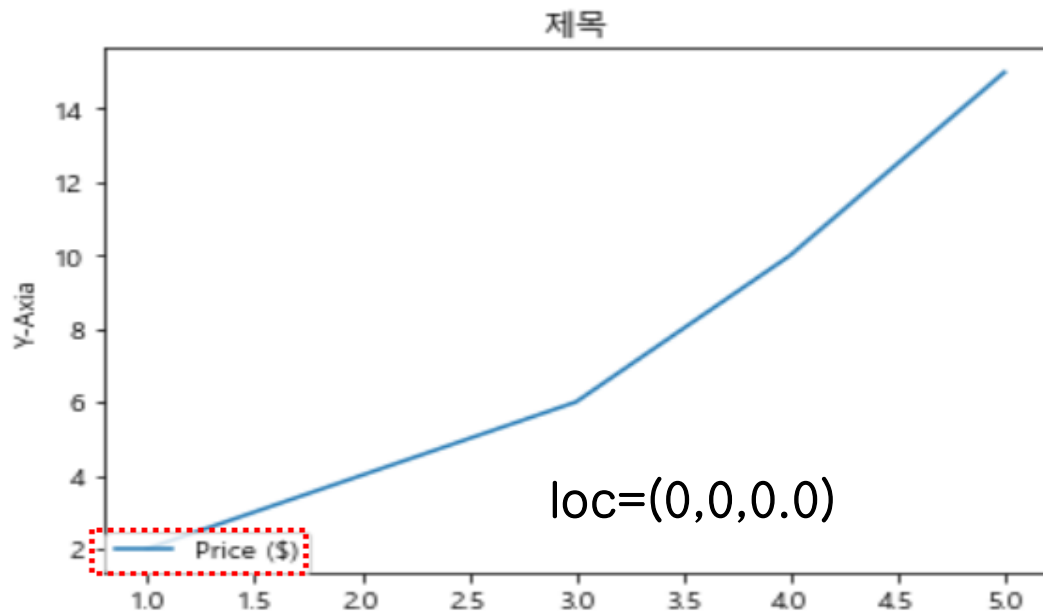
```
1 import matplotlib.pyplot as plt
2 plt.rc('font', family = 'Malgun Gothic')
3 plt.title('제목')
4 plt.xlabel('X-Axis')
5 plt.ylabel('Y-Axis')
6 plt.plot([1,2,3,4,5], [2,4,6,10,15], label='Price ($)')
7 plt.legend()
8 plt.show()
```



범례 위치 지정하기

$loc=(0.0, 0.0)$ 은 데이터 영역의 왼쪽 아래, $loc=(1.0, 1.0)$ 은 데이터 영역의 오른쪽 위 위치

```
1 import matplotlib.pyplot as plt
2 plt.rc('font', family = 'Malgun Gothic')
3 plt.title('제목')
4 plt.xlabel('X-Axis')
5 plt.ylabel('Y-Axis')
6 plt.plot([1,2,3,4,5], [2,4,6,10,15], label='Price ($)')
7 plt.legend(loc=(0,0))
8 plt.show()
```



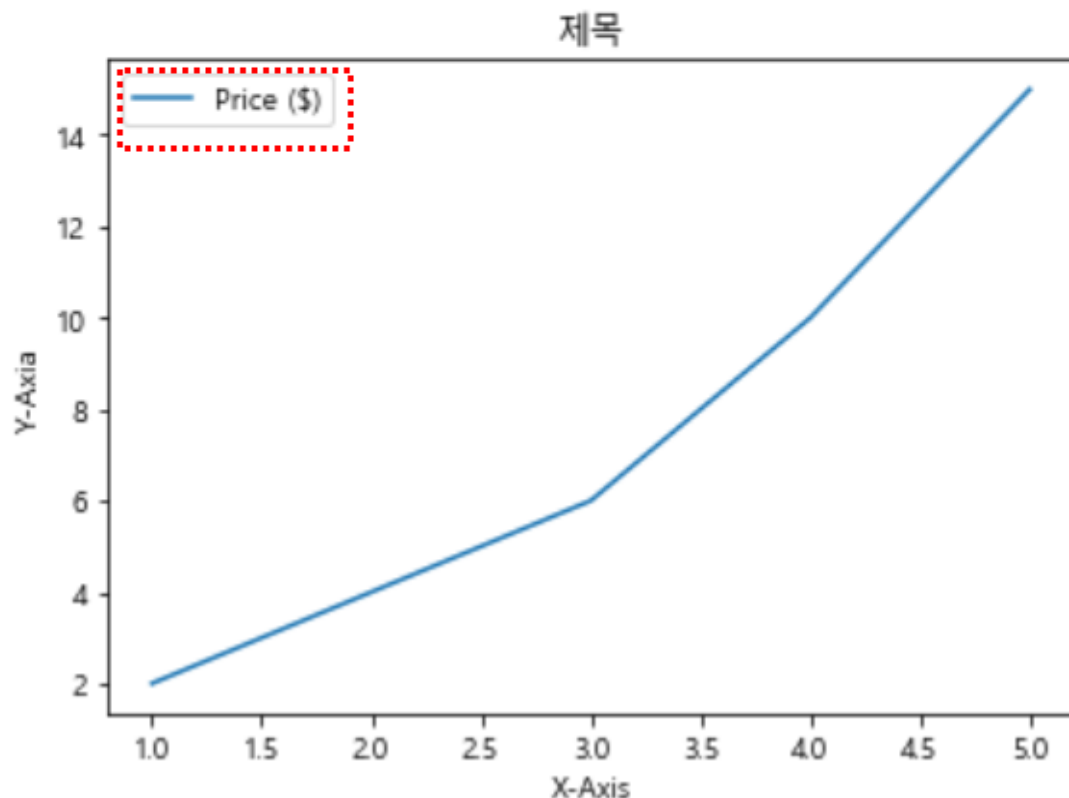
범례 위치 지정하기

범례위치 지정하기

범례의 위치는 loc에 들어가는 숫자에 따라 0 ~ 10 까지 넣을 수 있습니다.

0	그래프에 따라 자동 위치	6	왼쪽 중간
1	오른쪽 상단	7	오른쪽 중간
2	왼쪽 상단	8	중앙 하단
3	왼쪽 하단	9	중앙 상단
4	오른쪽 하단	10	정중앙
5	오른쪽		

```
1 import matplotlib.pyplot as plt
2 plt.rc('font', family = 'Malgun Gothic')
3 plt.title('제목')
4 plt.xlabel('X-Axis')
5 plt.ylabel('Y-Axis')
6 plt.plot([1,2,3,4,5], [2,4,6,10,15], label='Price ($)')
7 plt.legend(loc=2)
8 plt.show()
```



그래프에 색상 넣기

color 속성을 추가

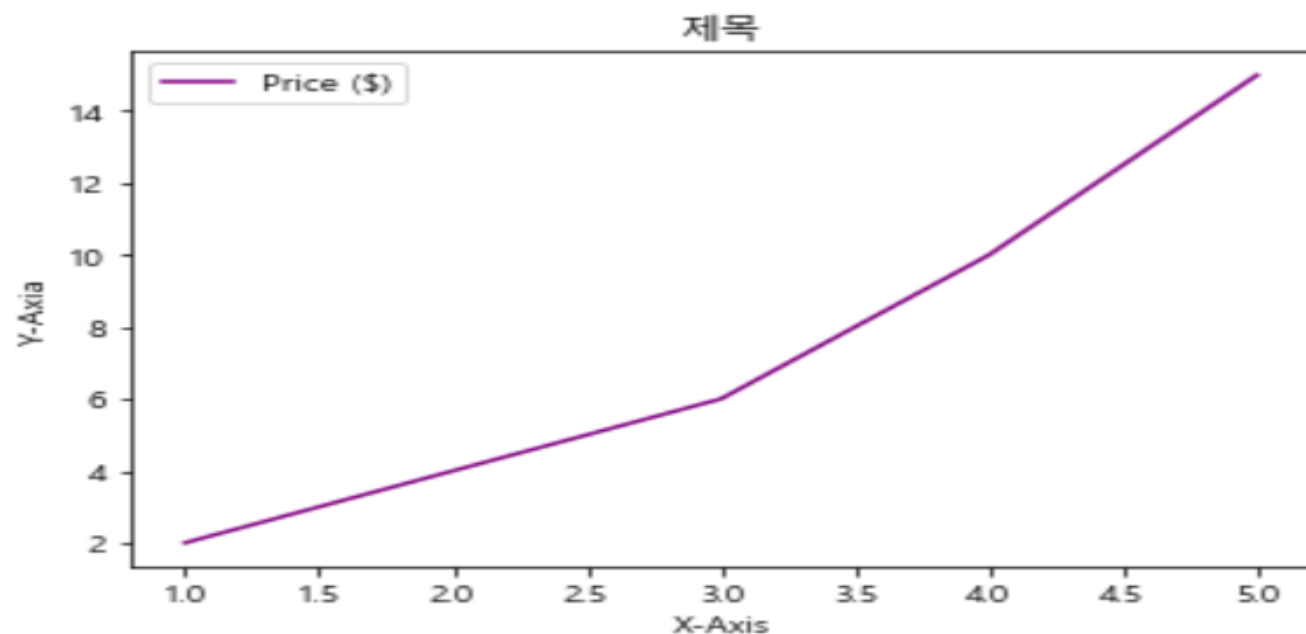
문자열	색
'b'	파란색(blue)
'g'	초록색(green)
'r'	빨간색(red)
'c'	청록색(cyan)
'm'	자주색(magenta)
'y'	노란색(yellow)
'w'	흰색(white)
'k'	검정색(black)

사용할 수 있는 다양한 색상 : https://matplotlib.org/gallery/color/named_colors.html

그래프에 색상 넣기

그래프에 선 색 넣기

```
1 import matplotlib.pyplot as plt
2 plt.rc('font', family = 'Malgun Gothic')
3 plt.title('제목')
4 plt.xlabel('X-Axis')
5 plt.ylabel('Y-Axis')
6 plt.plot([1,2,3,4,5], [2,4,6,10,15], label='Price ($)', color='purple')
7 plt.legend(loc=2)
8 plt.show()
```



그래프에 색상 넣기

color 속성을 추가

사용할 수 있는 다양한 색상 : https://matplotlib.org/gallery/color/named_colors.html

The screenshot shows the Matplotlib website's 'List of named colors' gallery. The page is organized into several sections, each displaying a set of color swatches with their corresponding names and abbreviations.

- Base Colors:** Includes primary colors (b, g, r), secondary colors (c, m, y), and black/white (k, w).
- Tableau Palette:** A set of 11 colors used in Tableau, including tab:blue, tab:orange, tab:green, tab:red, tab:purple, tab:brown, tab:pink, tab:gray, tab:olive, and tab:cyan.
- CSS Colors:** A large collection of standard CSS color names, such as black, white, red, green, blue, yellow, orange, and many others, arranged in four columns.

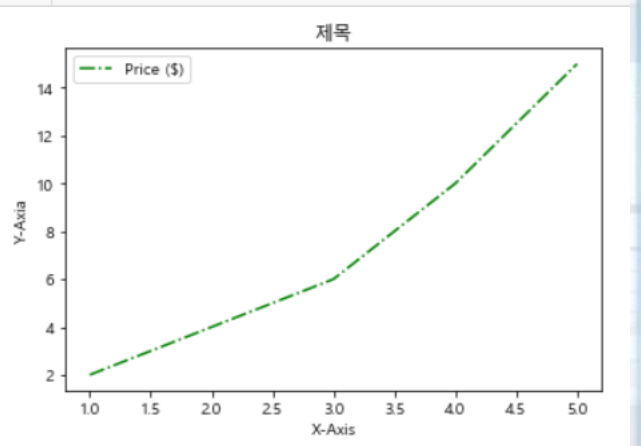
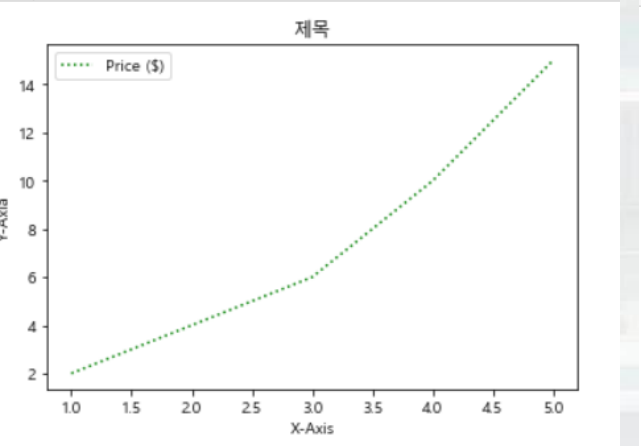
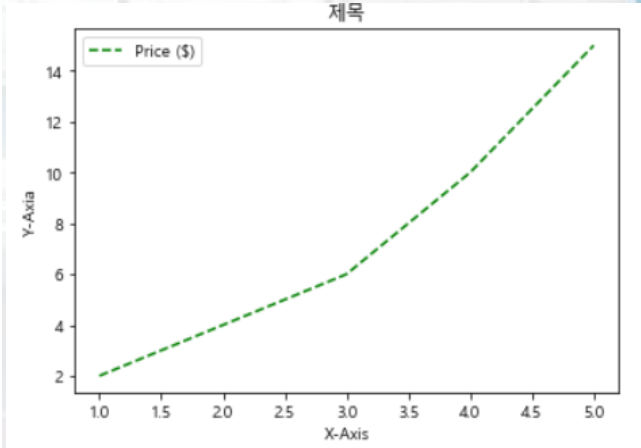
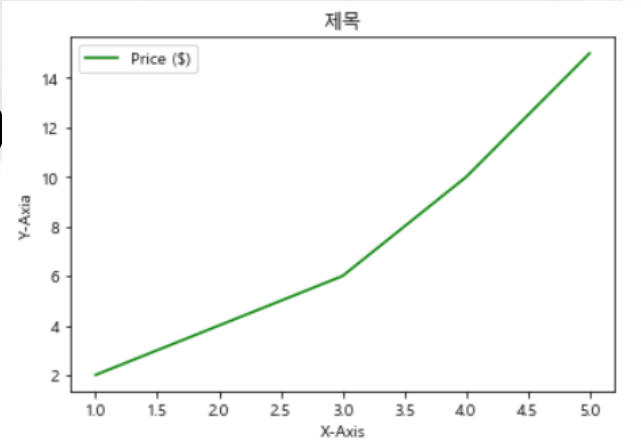
The left sidebar contains a search bar and a list of links to other documentation pages, including 'Colors in the default property cycle', 'Colorbar', 'Colormap reference', and 'List of named colors'.



그래프 선 모양 변경하기

plot()함수는 기본적으로 직선으로 그래프
linestyle 속성에 원하는 선 모양을 지정(linestyle->ls가능)

문자열	선
'-'	실선(Solid line)
'--'	대시선(Dashed line)
'.'	점선(Dotted line)
'-.'	대시-점(Dashed-dot)



마커모양 바꾸기

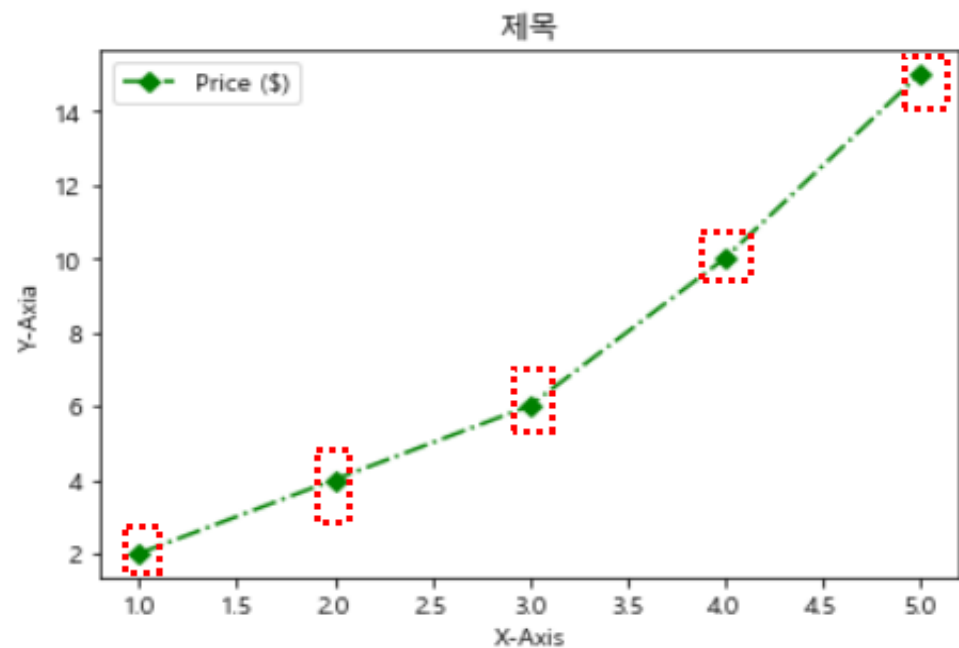
plot() 함수에 marker 속성을 설정하면 선, 점, 동그라미, 별 모양 등으로 모양을 바꿀수 있습니다.

문자열	설명
's'	사각형(squre)
'p'	오각형(pentagon)
'*'	별모양(star)
'D'	다이아몬드(diamond)
'd'	얇은 다이아몬드(thin diamond)
'o'	원(circle)

마커모양 바꾸기

마커 모양 바꾸기

```
1 plt.rc('font', family = 'Malgun Gothic')
2 plt.title('제목')
3 plt.xlabel('X-Axis')
4 plt.ylabel('Y-Axis')
5 plt.plot([1,2,3,4,5],[2,4,6,10,15], 'D', label='Price ($)', color='g', linestyle = '-.')
6 plt.legend(loc=2)
7 plt.show()
```



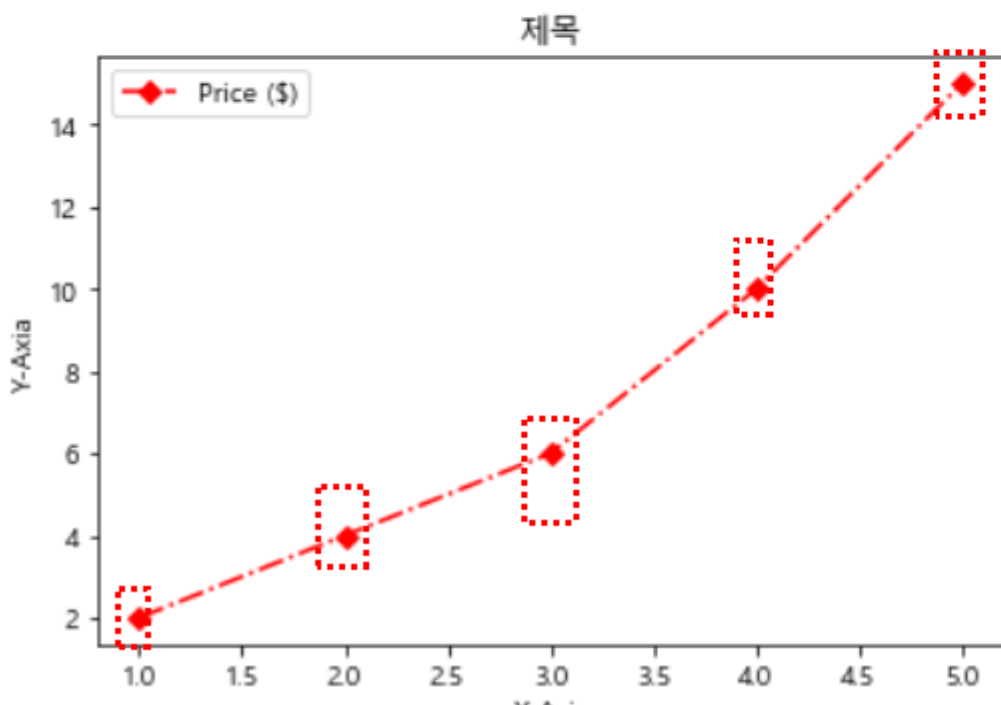
마커모양 바꾸기

```
1 plt.rc('font', family = 'Malgun Gothic')
2 plt.title('제목')
3 plt.xlabel('X-Axis')
4 plt.ylabel('Y-Axis')
5 plt.plot([1,2,3,4,5],[2,4,6,10,15], 'rD', label='Price ($)', linestyle = '-.')
6 plt.legend(loc=2)
7 plt.show()
```

read 색상

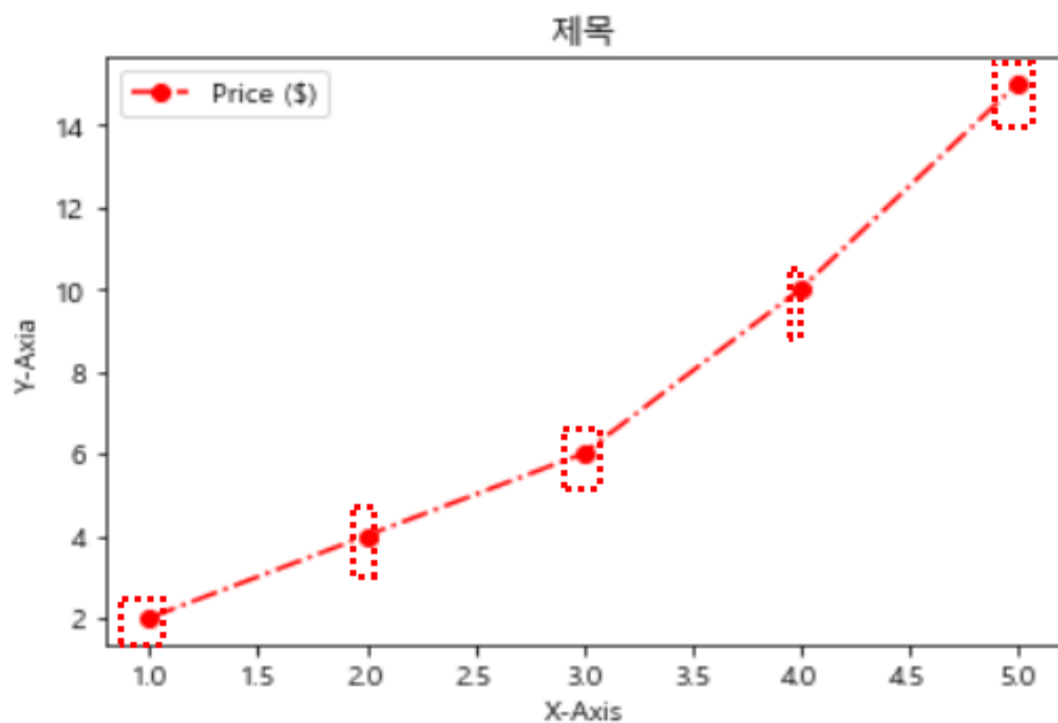
마커모양

`plt.plot(x,y,'rD')`



마커모양 바꾸기

```
1 plt.rc('font', family = 'Malgun Gothic')
2 plt.title('제목')
3 plt.xlabel('X-Axis')
4 plt.ylabel('Y-Axia')
5 plt.plot([1,2,3,4,5],[2,4,6,10,15], 'ro' label='Price ($)', linestyle = '-.')
6 plt.legend(loc=2)
7 plt.show()
```



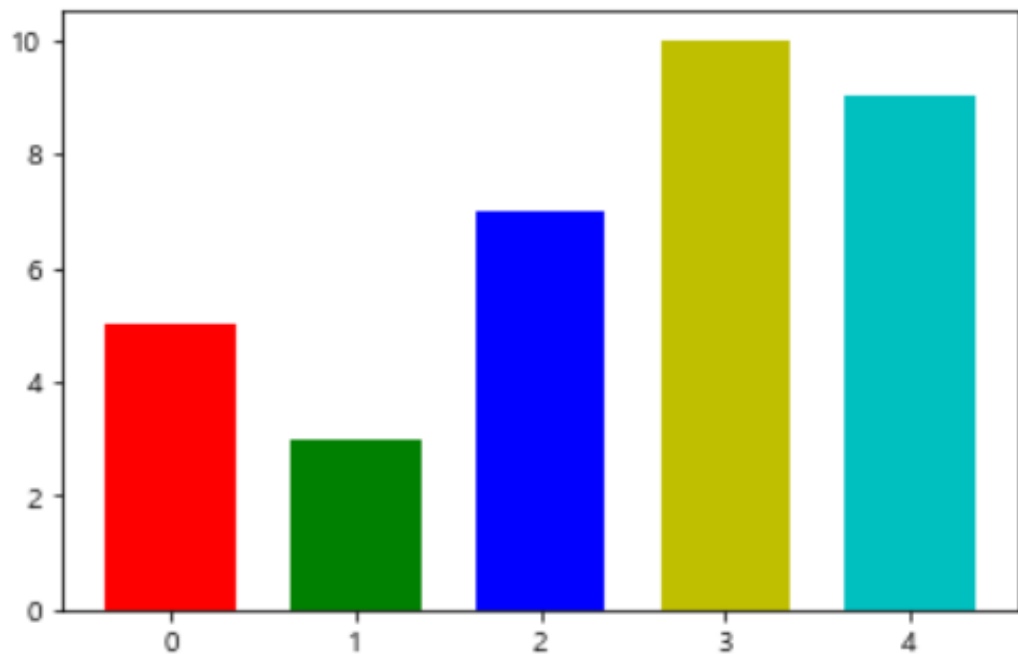
다양한 차트 및 플롯

Bar 차트 plt.bar() 함수를 사용

Bar 차트
조사한 수를 막대로
나타낸 그래프, 수량
의 많고 적음을 한눈
에 쉽게 비교

Bar 차트 그리기

```
1 import matplotlib.pyplot as plt
2 y = [5, 3, 7, 10, 9]
3 x = range(len(y))
4 plt.bar(x, y, width=0.7, color=['r','g','b','y','c'])
5 plt.show()
```



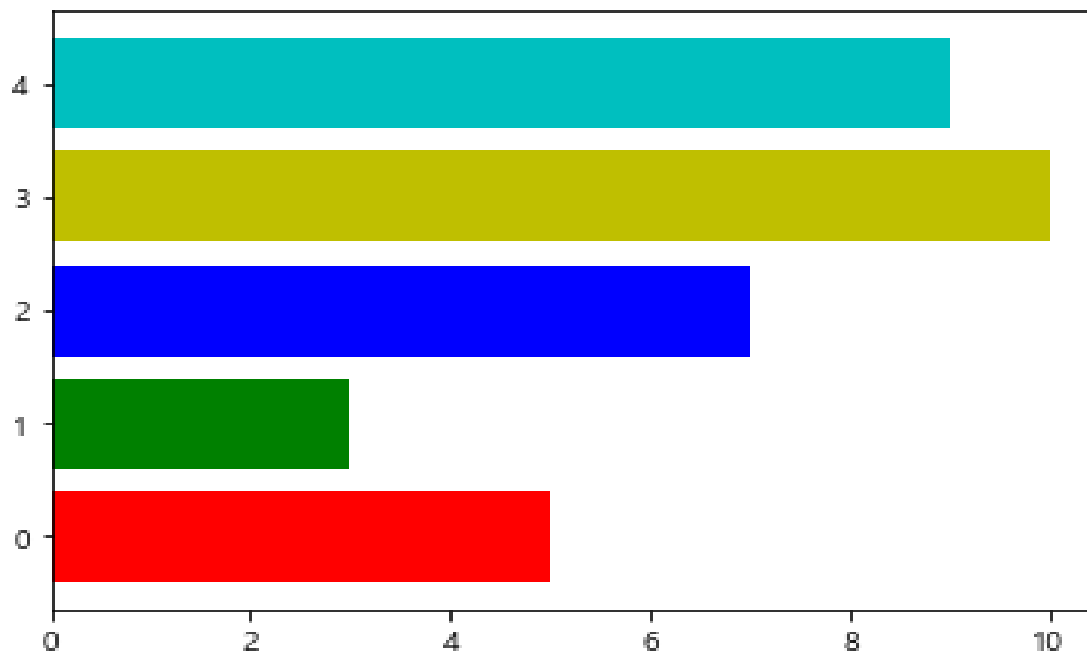
다양한 차트 및 플롯

Bar 차트

plt.barh() 함수를 사용

가로 방향으로 그리기

```
1 import matplotlib.pyplot as plt
2 y = [5, 3, 7, 10, 9]
3 x = range(len(y))
4 plt.barh(x, y, color=['r', 'g', 'b', 'y', 'c'])
5 plt.show()
```



다양한 차트 및 플롯

PI 차트

plt.pie() 함수를 사용

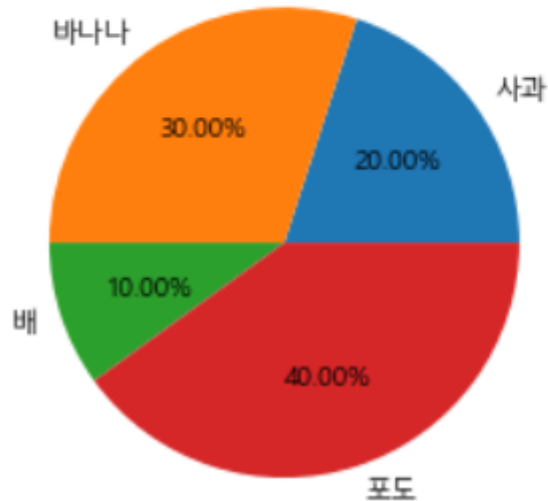
파이 차트 (Pie chart, 원 그래프)는 범주별 구성 비율을 원형으로 표현한 그래프

위의 그림과 같이 부채꼴의 중심각을 구성 비율에 비례하도록 표현

PI 차트 그리기

```
1 import matplotlib.pyplot as plt
2
3 ratio = [20, 30, 10, 40]
4 labels = ['사과', '바나나', '배', '포도']
5
6 plt.pie(ratio, labels=labels, autopct='%.2f%%')
7 plt.show()
```

autopct는 부채꼴 안에 표시될 숫자의 형식을 지정



다양한 차트 및 플롯

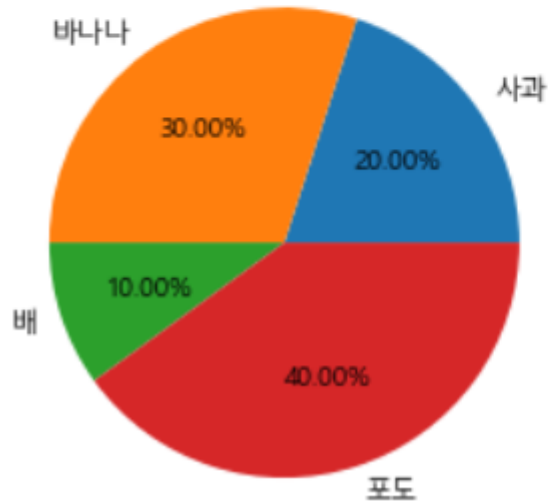
PI 차트 plt.pie() 함수를 사용

파이 차트 (Pie chart, 원 그래프)는
범주별 구성 비율을 원형으로 표현
한 그래프
위의 그림과 같이 **부채꼴의 중심각**
을 구성 비율에 비례하도록 표현

autopct : 각 파이 문자열 출
력 형식 설정
.%.2f는 소수 둘째 자리까지
출력하라는 의미
%%는 문자 '%'를 표현

PI 차트 그리기

```
1 import matplotlib.pyplot as plt
2
3 ratio = [20, 30, 10, 40]
4 labels = ['사과', '바나나', '배', '포도']
5
6 plt.pie(ratio, labels=labels, autopct='%.2f%%')
7 plt.show()
```



다양한 차트 및 플롯

PI 차트

plt.pie() 함수를 사용

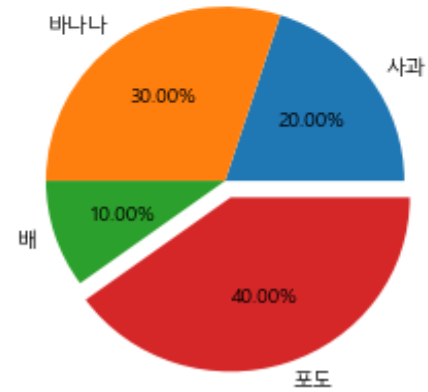
autopct : 각 파이 문자열 출력
형식 설정

%.2f는 소수 둘째 자리까지 출력하라는 의미

%%는 문자 '%'를 표현

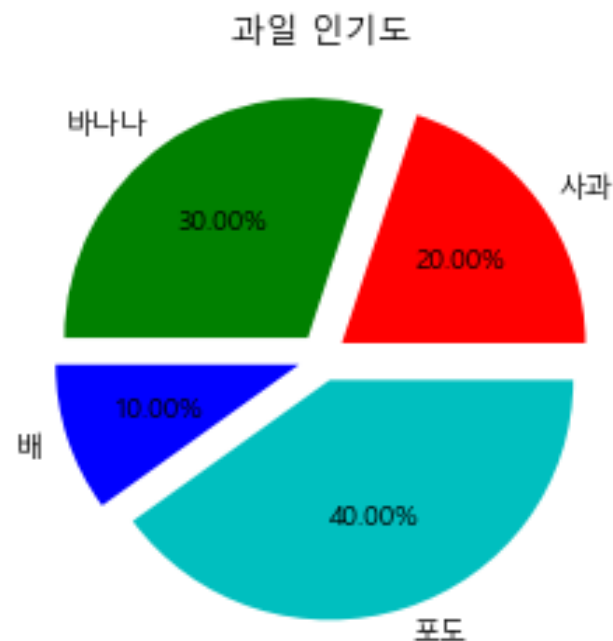
explode : 각 파이를 분리하여 표현

```
1 import matplotlib.pyplot as plt
2
3 ratio = [20, 30, 10, 40]
4 labels = ['사과', '바나나', '배', '포도']
5
6 plt.pie(ratio, labels=labels, autopct='%.2f%%' , explode = (0,0,0,0.1)) # explode : 각 파이를 분리하여 표현
7
8 plt.show()
```



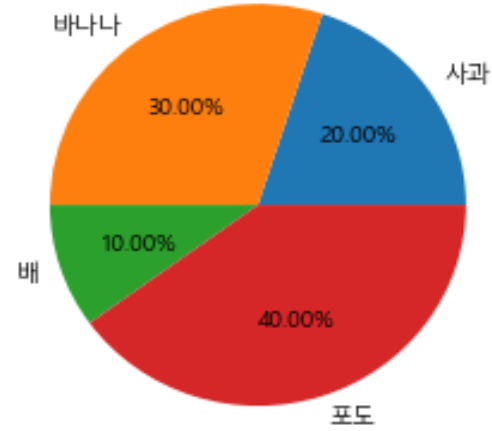
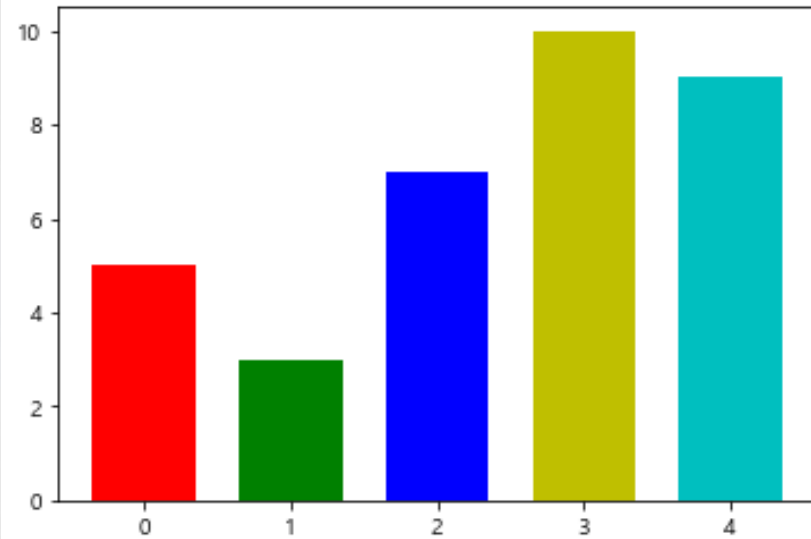
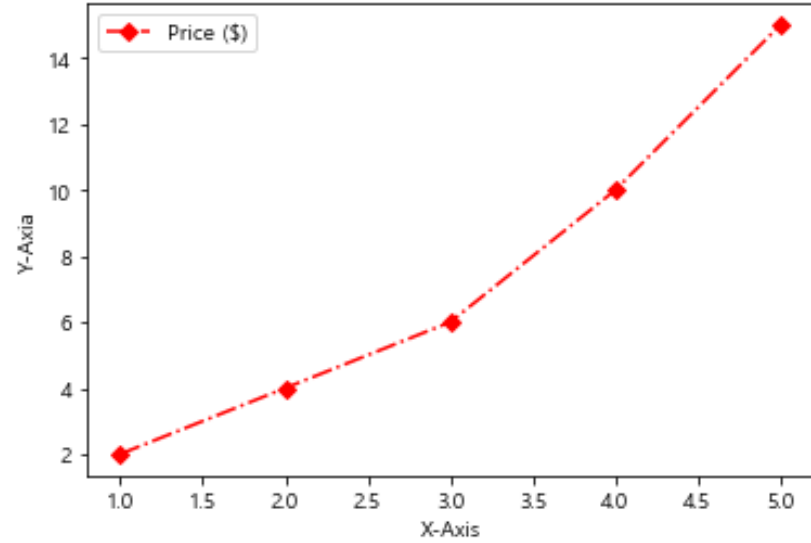
다양한 차트 및 플롯

```
1 ratio = [20,30,10,40]
2 labels = ['사과','바나나','배','포도']
3 plt.title('과일 인기도') # 파이 차트 제목 넣기
4 plt.pie(ratio, labels = labels, autopct = '%.2f%%' , explode = (0.1,0.1,0.1,0.1), colors =['r','g','b','c'])
5 plt.show()
```



마무리

제목



과일 인기도

