

기초 인공지능 프로그래밍

10장 Matplotlib 라이브러리 (데이터 시각화)

matplotlib

- 점이나 선, 막대 그래프 등의 시각적 이미지를 사용하여 데이터를 시각화하는 라이브러리
 - 데이터 분석 및 추론에 도움
 - matplotlib 라이브러리의 pyplot 모듈을 사용

```
from matplotlib import pyplot as plt
```

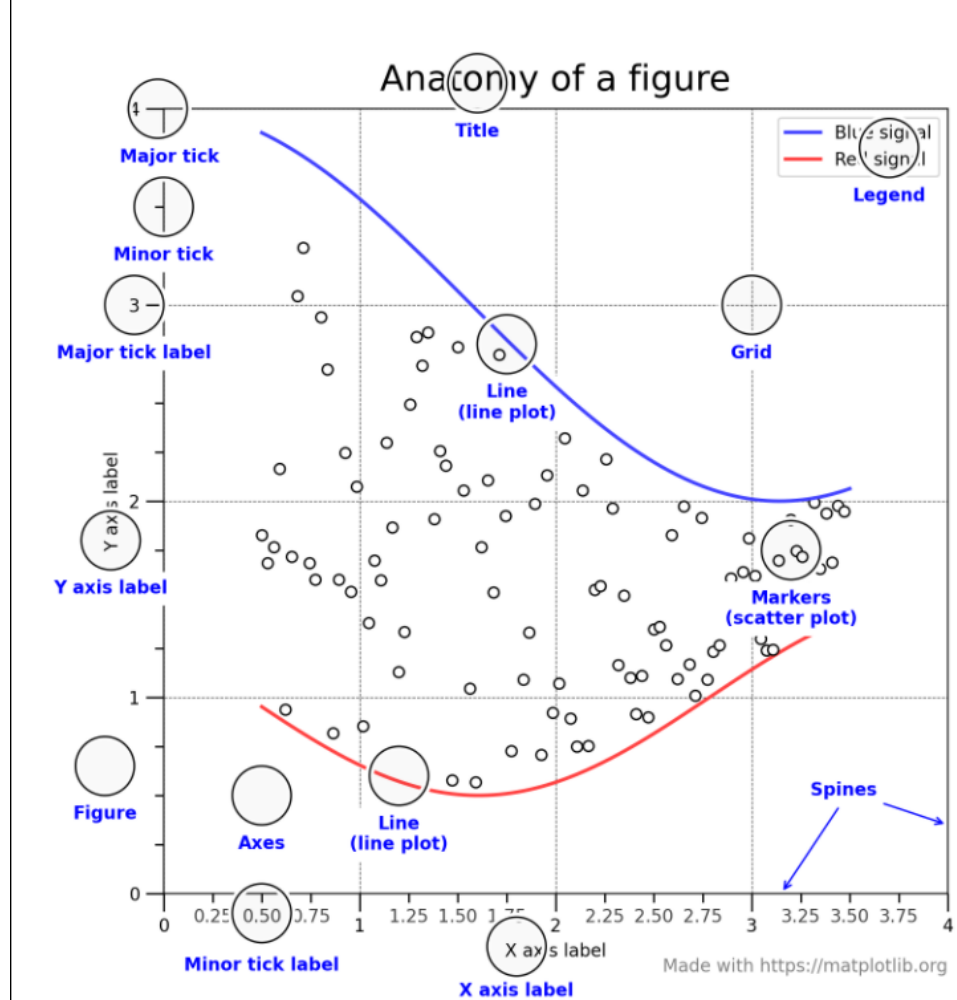
또는

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

matplotlib

Parts of a Figure

Here are the components of a Matplotlib Figure.



matplotlib

- matplotlib.pyplot 모듈 함수
 - plt.plot(x축값, y축값, 'option')
: 라인(실선, 점선....) 플롯을 그리는 함수
 - plt.bar(x축값, y축값)
: 수직 막대 그래프를 그리는 함수
 - plt.scatter(x축값, y축값)
: 산점도를 그리는 함수
 - plt.pie()
: 파이 차트를 그리는 함수
 - plt.title(제목)
: 그래프에 제목 붙이는 함수
 - plt.xlabel(제목), plt.ylabel(제목)
: x축, y축에 제목 붙이는 함수
 - plt.legend([라인1범례, 라인2범례,])
: 각 라인에 대한 범례 추가, 범례를 순서대로 지정하면 됨
 - plt.xlim(n, m), plt.ylim(n, m)
: x축, y축의 범위 설정(n에서 m 까지)
 - plt.show()
: 그래프를 화면에 표시

matplotlib

- 선의 색상, 스타일, 마커 등의 옵션(option)을 지정하여 그래프를 그릴 수 있음

color

Character	Color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black

line style

Character	description
':'	dotted line style
'--'	dashed line style

marker

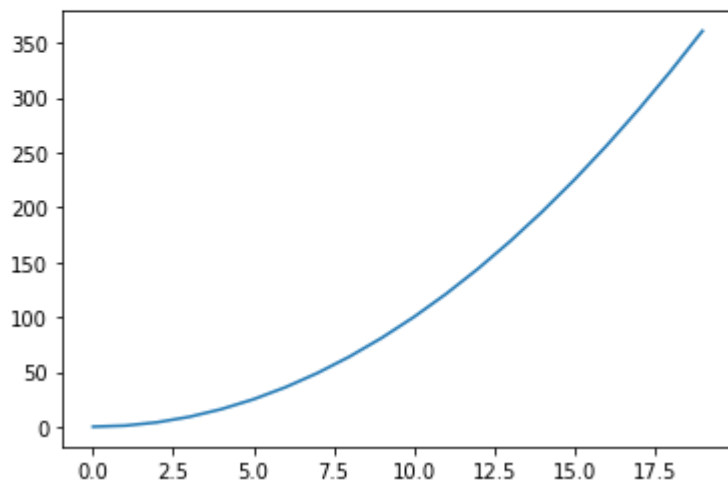
Character	description
.'	point marker
'o'	circle marker
's'	square marker
'*'	star marker
'^'	triangle_up marker

```
import matplotlib.pyplot as plt
plt.plot(x, y)
plt.plot(x, y, 'g--o') # green, dashed line, circle marker 지정
```

matplotlib

- 2차 함수 ($y = x^2$) 그래프 그리기

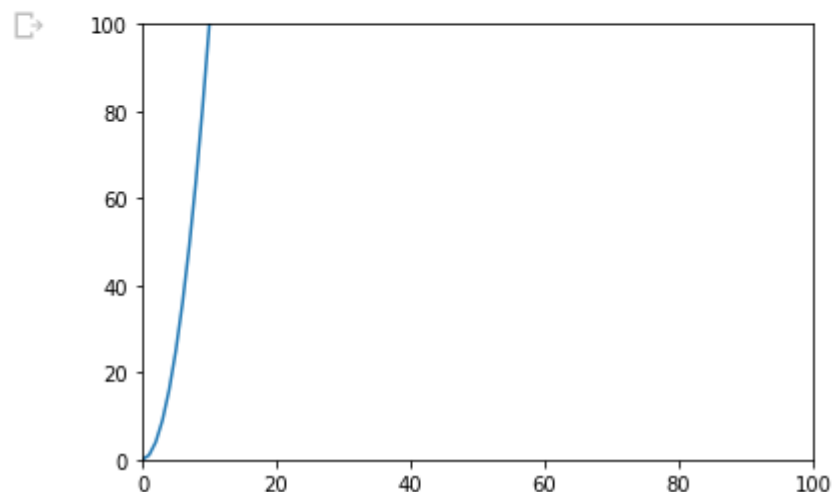
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.arange(20)
4 plt.plot(x**2)
5 plt.show()
```



x 값 증가에 따른 y 값 증가 속도를
좀 더 확실하게 파악할 수 있음

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.arange(20)
4 plt.plot(x**2)
5 plt.axis([0,100, 0,100])
6 plt.show()
```

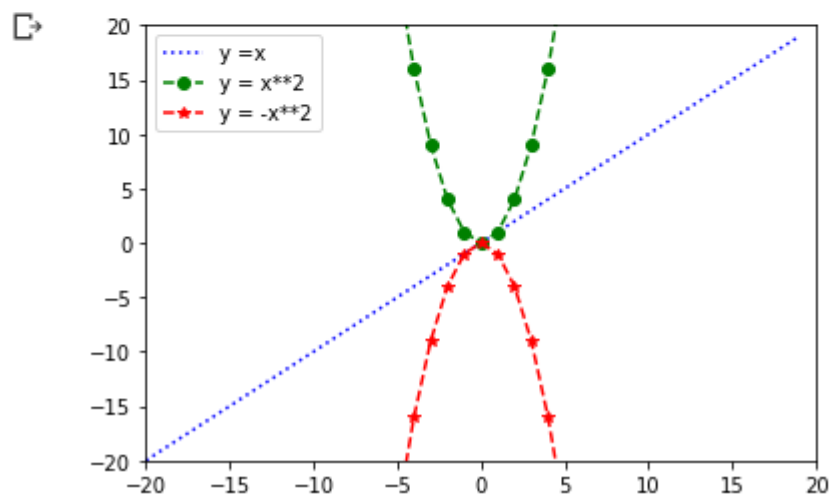
x, y 축 범위 설정



matplotlib

- 3개 함수 그래프 그리기

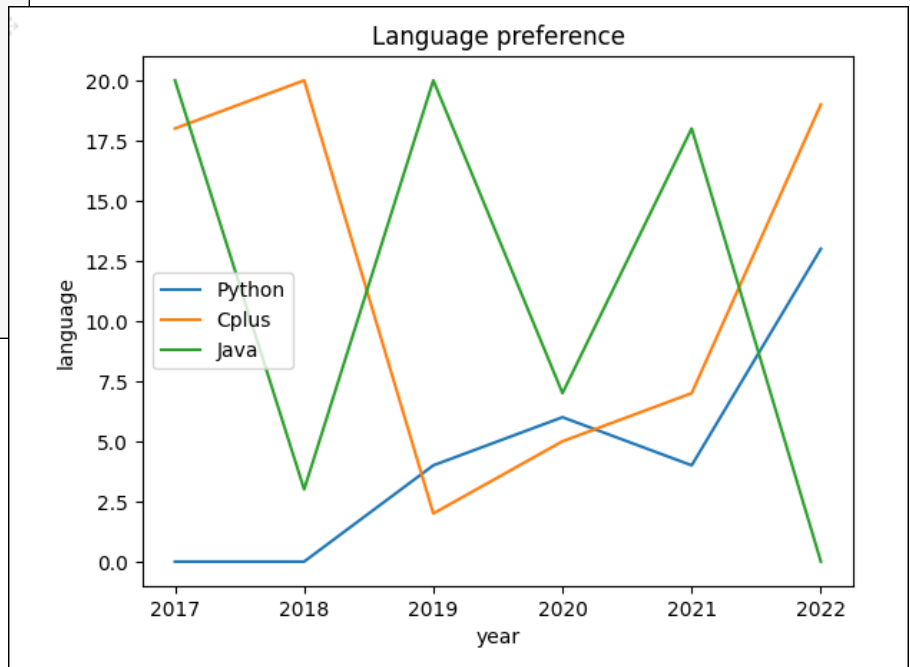
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.arange(-20, 20)
4 y1 = x
5 y2 = x**2
6 y3 = -x**2
7 plt.plot(x, y1, 'b:')
8 plt.plot(x, y2, 'g--o')
9 plt.plot(x, y3, 'r--*')
10 plt.legend(["y =x", "y = x**2", "y = -x**2"])
11 plt.axis([-20, 20, -20, 20])
12 plt.show()
```



matplotlib

- 3개 programming language에 대한 년도 별 20명 학생의 선호도 그래프 그리기(난수로 임의의 데이터 생성)

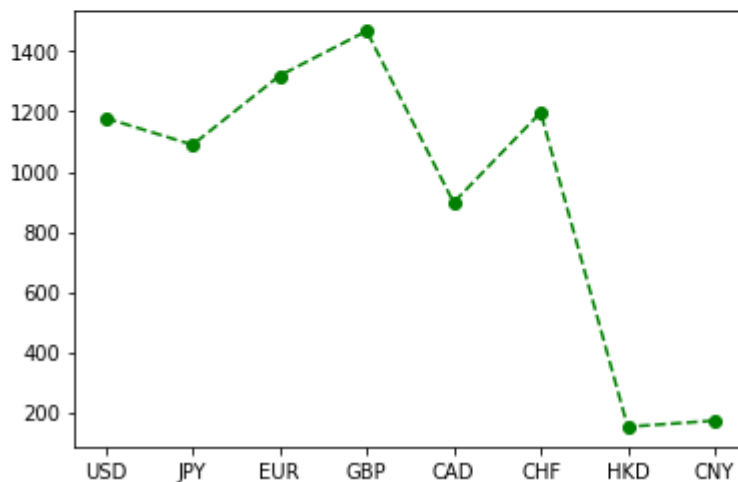
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 year = range(2017,2023)
4 Python = np.random.randint(0,21,6)
5 cPlus = np.random.randint(0,21,6)
6 Java = np.random.randint(0,21,6)
7 plt.plot(year, Python)
8 plt.plot(year, cPlus)
9 plt.plot(year, Java)
10 plt.title("Language preference")
11 plt.xlabel("year")
12 plt.ylabel("language")
13 plt.legend(["Python", "Cplus", "Java"])
14 plt.show()
```



matplotlib

- 특정 날짜의 통화별 매매기준율 그래프 그리기

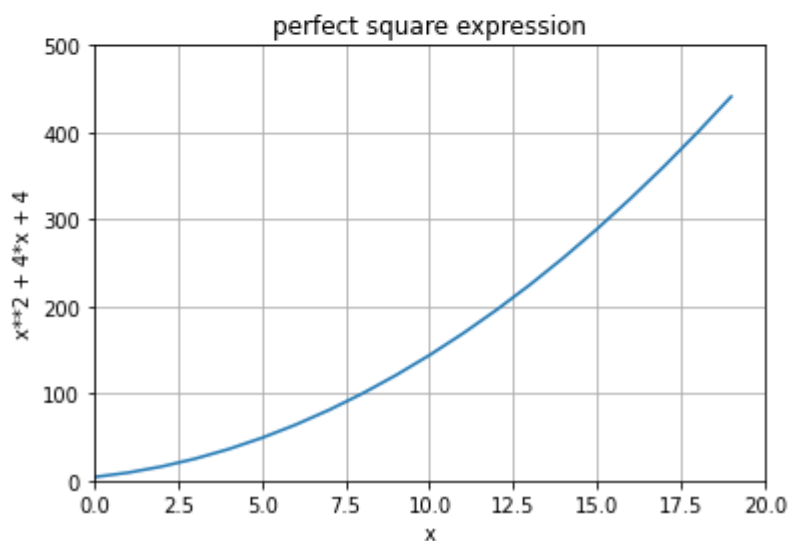
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 y= np.array([1178.5,1089.39,1319.37,1468.28,
4             897.3,1197.05,150.88,171.17]) # 매매기준율
5 x= np.array(['USD', 'JPY', 'EUR', 'GBP',
6             'CAD', 'CHF', 'HKD', 'CNY']) # 통화표시
7
8 plt.plot( x, y, 'go--') # 그래프 생성
9 plt.show() # 화면에 표시
```



matplotlib

- 2차 방정식 그래프 그리기

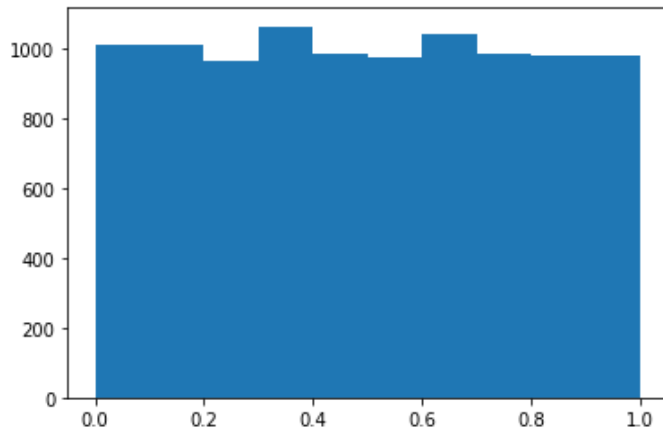
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.arange(20)
5 y = x**2 + 4*x + 4
6 plt.plot(x, y)
7 plt.grid("True")
8 plt.xlabel("x")
9 plt.ylabel("x**2 + 4*x + 4")
10 plt.xlim(0, 20)
11 plt.ylim(0, 500)
12 plt.title("perfect square expression")
13 plt.show()
```



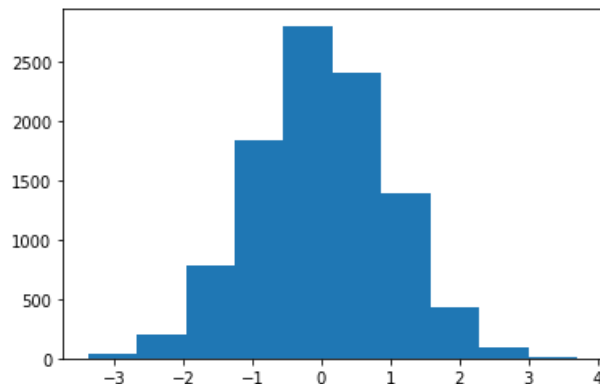
matplotlib

- 히스토그램은 주어진 데이터를 몇 개의 구간으로 나누고, 각 구간의 빈도를 조사하여 나타내는 막대 그래프이며 데이터의 분포 상태 확인에 유용함

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 data = np.random.rand(10000) # rand() 함수는 0.0에서 1.0 사이의
4                               # 균등 분포 난수 생성. 10000개를 생성
5 plt.hist(data, bins = 10)    # 10개 구간으로 나누어 빈도 조사
6 plt.show()
```



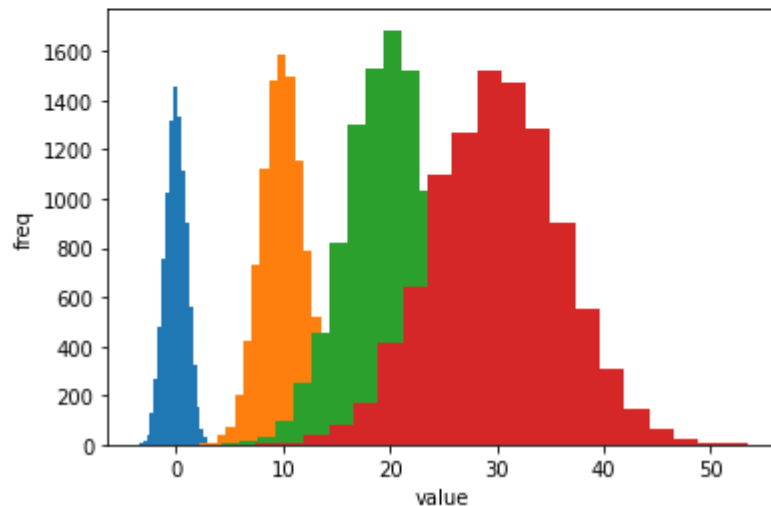
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 data = np.random.randn(10000) # randn() 함수는 평균 0,
4                               # 표준편차 1의 표준 정규 분포 난수 생성
5 plt.hist(data, bins = 10)    # 10개 구간으로 나누어 빈도 조사
6 plt.show()
```



matplotlib

- 평균과 표준편차가 다른 정규 분포 히스토그램

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 %matplotlib inline
4 m = [0, 10, 20, 30]; std = [1, 2, 4, 6]
5 data = np.zeros((len(m), 10000))
6 for i in range(len(m)) :
7     data[i] = m[i] + std[i]*np.random.randn(10000)
8     plt.hist(data[i], bins = 20)
9 plt.xlabel("value")
10 plt.ylabel("freq")
11 plt.show()
```

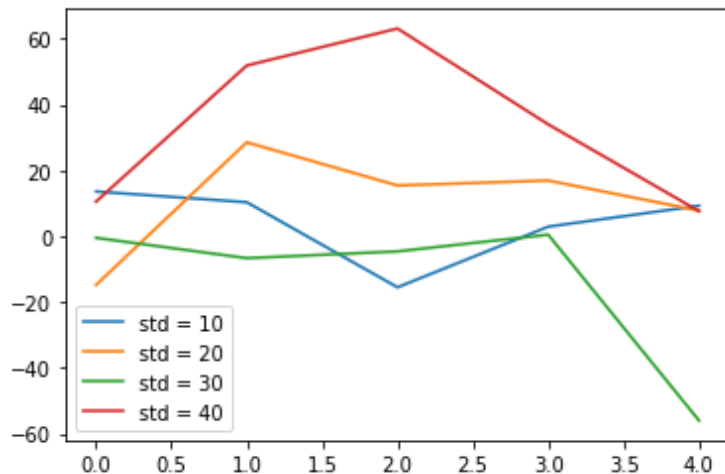


matplotlib

- 평균 0과 여러 표준 편차 값으로 정규 분포 데이터를 10개씩 얻어 데이터의 분포 상태를 그래프 그리기

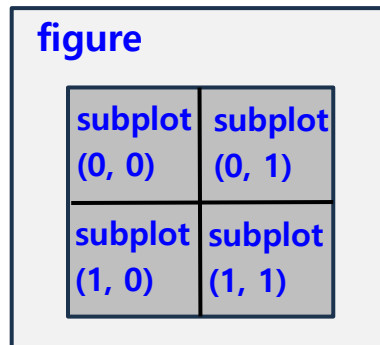


```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 std = [10, 20, 30, 40]
4 data = np.zeros((len(std), 5))
5 for i in range(len(std)) :
6     data[i] = np.random.normal(0, std[i], 5) #평균 0, 표준 편차 std[i]인 데이터 5개
7     plt.plot(data[i])
8 plt.legend(["std = "+str(std[0]), "std = "+str(std[1]), "std = "+str(std[2]), "std = "+str(std[3])]) # 범례
9 plt.show()
```



matplotlib

- 여러 개의 그래프 그리기
 - 도화지 한 장에 그래프 한 개 그리기 개념이면 앞 강의 예제와 같은 방식으로 그리면 됨
 - 표현할 각 그래프가 서로 연관성이 없는 그래프인 경우는 그래프를 각자의 도화지에 그리는 것이 좋음
 - 같은 주제이거나 연관성이 있는 그래프인 경우는 하나의 도화지를 영역 분할하여 같이 그리는 것이 좋음
 - figure : 도화지에 해당
 - subplot : figure 에서 그래프가 그려지는 영역

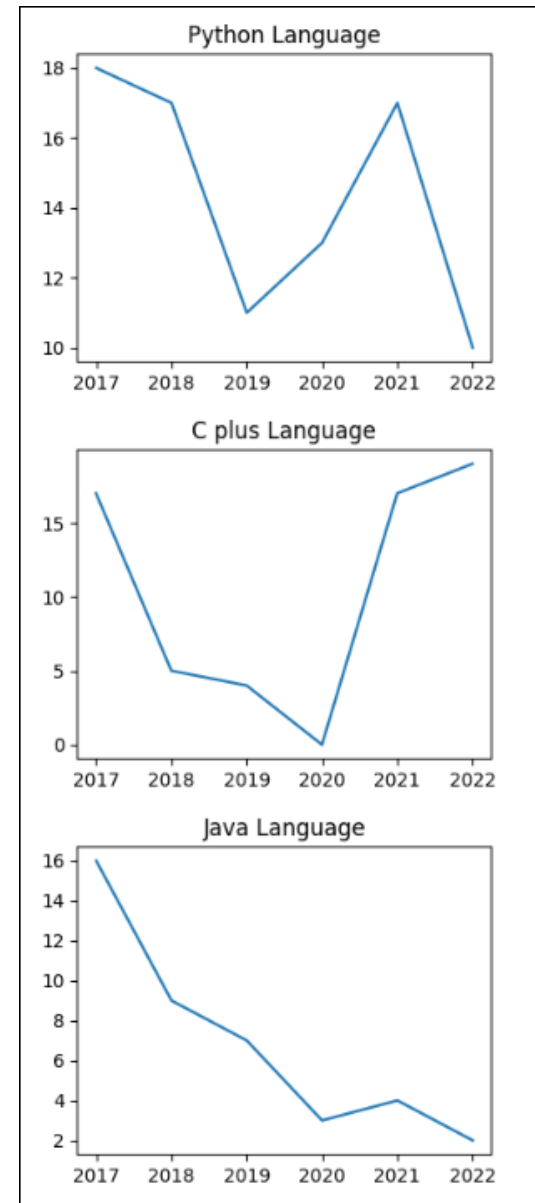


matplotlib

- 여러 개의 figure에 각 그래프 그리기
 - 지정한 ID를 가지는 figure를 생성
 - 선택된 figure에 그래프 그리기

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 year = range(2017,2023)
4 Python = np.random.randint(0,21,6)
5 cPlus = np.random.randint(0,21,6)
6 Java = np.random.randint(0,21,6)
7 fig1 = plt.figure(1, figsize = (4,3))
8 plt.plot(year, Python)
9 plt.title("Python Language")
10 fig2 = plt.figure(2, figsize = (4,3))
11 plt.plot(year, cPlus)
12 plt.title("C plus Language")
13 fig3 = plt.figure(3, figsize = (4,3))
14 plt.plot(year, Java)
15 plt.title("Java Language")
16 plt.show()
```

↓
ID, size 지정은 해주지 않아도 됨



matplotlib

- 하나의 figure에 여러 개의 그래프 그리기
 - figure를 원하는 만큼의 subplot으로 구분하여 각 subplot에 그래프 그림

```
import matplotlib.pyplot as plt
import numpy as np
year = range(2017,2023)
Python = np.random.randint(0,21,6)
cPlus = np.random.randint(0,21,6)
Java = np.random.randint(0,21,6)
fig,(sub1, sub2, sub3) = plt.subplots(1, 3, constrained_layout = True)
sub1.plot(year, Python)
sub1.set_title("Python")
sub2.plot(year, cPlus)
sub2.set_title("C plus")
sub3.plot(year, Java)
sub3.set_title("Java")
plt.show()
```

그래프 간의 위치를 고려해서 배치

