



데이터 시각화 이해와 실습

Lecture 04. 데이터 시각화 기초

동덕여자대학교
데이터사이언스 전공
권 범

목차

- ❖ 01. 기본 그래프 그리기 <-- 꺾은 선 그래프 <-- matplotlib
- ❖ 02. 크리스마스의 기온 변화를 그래프로 그리기
- ❖ 03. 기온 데이터를 다양하게 시각화하기 <-- 히스토그램

01. 기본 그래프 그리기

- 02. 크리스마스의 기온 변화를 그래프로 그리기
- 03. 기온 데이터를 다양하게 시각화하기

01. 기본 그래프 그리기

❖ 시각화 라이브러리

- 데이터 분석 목적에 따라 선택적으로 사용

Basic

시각화 라이브러리	특징
맷플롯립(Matplotlib)	파이썬에서 가장 많이 사용하며, 데이터를 차트나 플롯으로 시각화하는 라이브러리이다. 판다스의 데이터프레임을 바로 시각화할 때도 내부적으로 맷플롯립을 사용한다. 맷플롯립은 데이터 분석 이전의 데이터 이해를 위한 시각화 또는 데이터 분석 이후의 결과를 시각화하기 위해 사용된다.
시본(Seaborn)	맷플롯립을 기반으로 색 테마, 차트 기능 등을 추가해 주는 라이브러리이다. 맷플롯립과 함께 사용하며 히트맵, 카운트 플롯 등을 제공한다.
폴리움(Folium)	지도 데이터를 이용하여 위치 정보를 시각화하는 라이브러리이다. 자바스크립트 기반으로 상호작용의 그래프를 그릴 수 있다.
파이차트(Pyecharts)	바이두(Baidu)에서 데이터 시각화를 위해 만든 파이썬 버전의 라이브러리이다. 다양한 그래프들이 내장되어 있고 자바스크립트 기반으로 상호작용의 그래프를 그릴 수 있다.
플로트나인(Plotnine)	R의 ggplot2에 기반을 두어 그래프를 그려 주는 라이브러리이다. R 시각화 경험이 있다면 편리하게 사용할 수 있다.
플로틀리(Plotly)	상호작용 그래프를 그려 주는 라이브러리이다. R, 스칼라, 파이썬, 자바스크립트, 매트랩 등에서 사용할 수 있다. 사용이 쉽고 세련된 도구이며 온라인과 오프라인 버전이 따로 존재한다.
보케(Bokeh)	맷플롯립의 대화형 버전으로, 다양한 기능을 제공하고 디자인이 훌륭하며 플롯(축)들 간의 링크가 가능하다. 반면 시본에 비교하여 문법이 복잡하다는 단점이 있다.

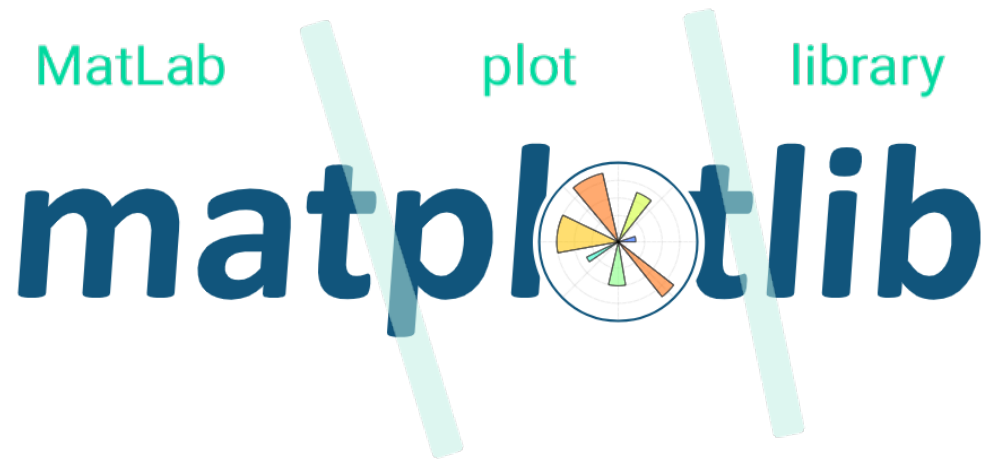
시각화 라이브러리

[사진출처] 데이터 분석을 위한 전처리와 시각화 with 파이썬 (출판사: 길벗)

01. 기본 그래프 그리기

❖ matplotlib 라이브러리란? (1/7)

- 파이썬으로 데이터를 시각화할 때 가장 많이 사용하는 라이브러리 중 하나
- 2차원 형태의 그래프, 이미지 등을 그릴 때 사용
- 실제 과학 컴퓨팅 연구 분야나 인공지능 연구 분야에서도 많이 활용

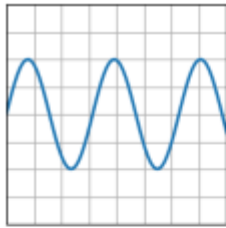


01. 기본 그래프 그리기

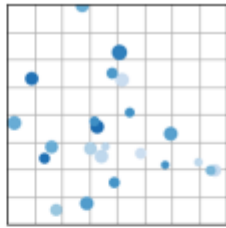
❖ matplotlib 라이브러리란? (2/7)

- 어떤 그래프들을 그릴 수 있나요?

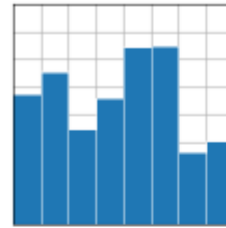
Basic Plot Types



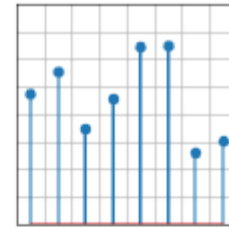
`plot(x, y)`



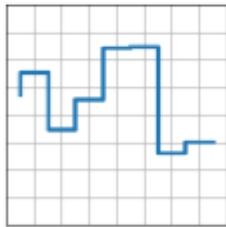
`scatter(x, y)`



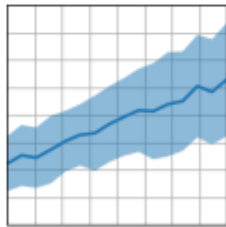
`bar(x, height) / barh(y, width)`



`stem(x, y)`



`step(x, y)`



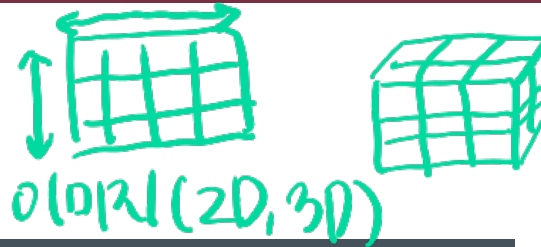
`fill_between(x, y1, y2)`

막대 그래프

01. 기본 그래프 그리기

❖ matplotlib 라이브러리란? (3/7)

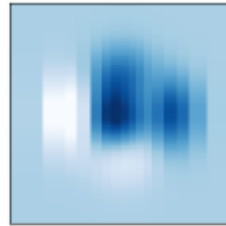
- 어떤 그래프들을 그릴 수 있나요?



Plots of Arrays and Fields



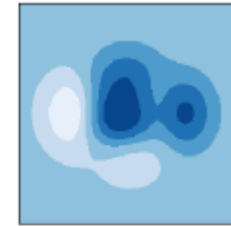
`imshow(Z)`



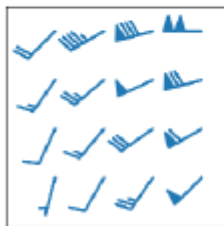
`pcolormesh(X, Y, Z)`



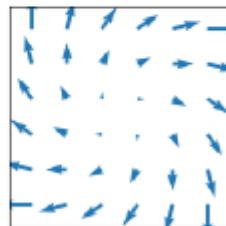
`contour(X, Y, Z)`



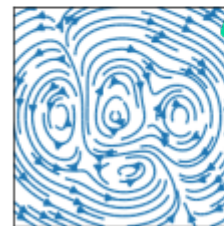
`contourf(X, Y, Z)`



`barbs(X, Y, U, V)`



`quiver(X, Y, U, V)`



`streamplot(X, Y, U, V)`

← 2차원

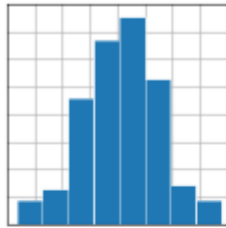
01. 기본 그래프 그리기

❖ matplotlib 라이브러리란? (4/7)

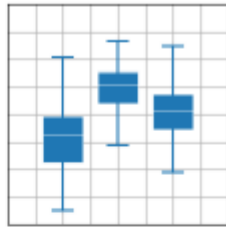
- 어떤 그래프들을 그릴 수 있나요?

Statistics Plots

ch.03

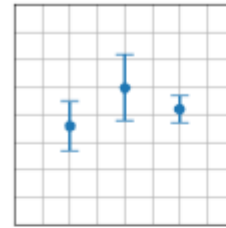


hist(x)

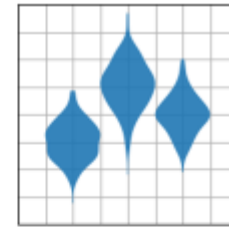


boxplot(X)

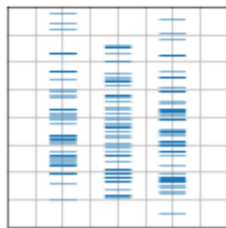
상자 그림



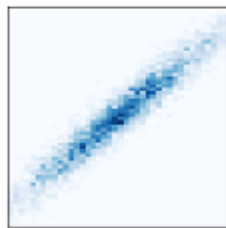
errorbar(x, y, yerr, xerr)



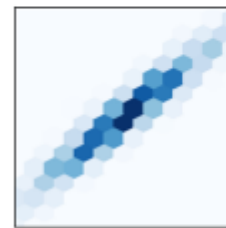
violinplot(D)



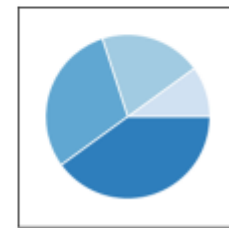
eventplot(D)



hist2d(x, y)



hexbin(x, y, C)



pie(x)

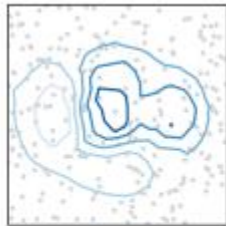
원 그래프, 파이 차트

01. 기본 그래프 그리기

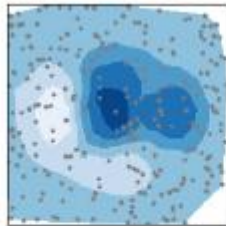
❖ matplotlib 라이브러리란? (5/7)

- 어떤 그래프들을 그릴 수 있나요?

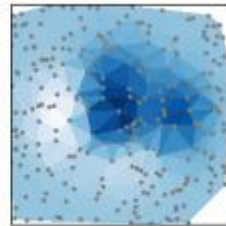
Unstructured Coordinates



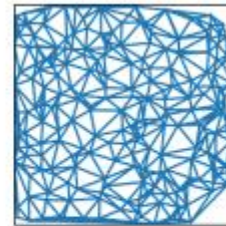
`tricontour(x, y, z)`



`tricontourf(x, y, z)`



`tripcolor(x, y, z)`

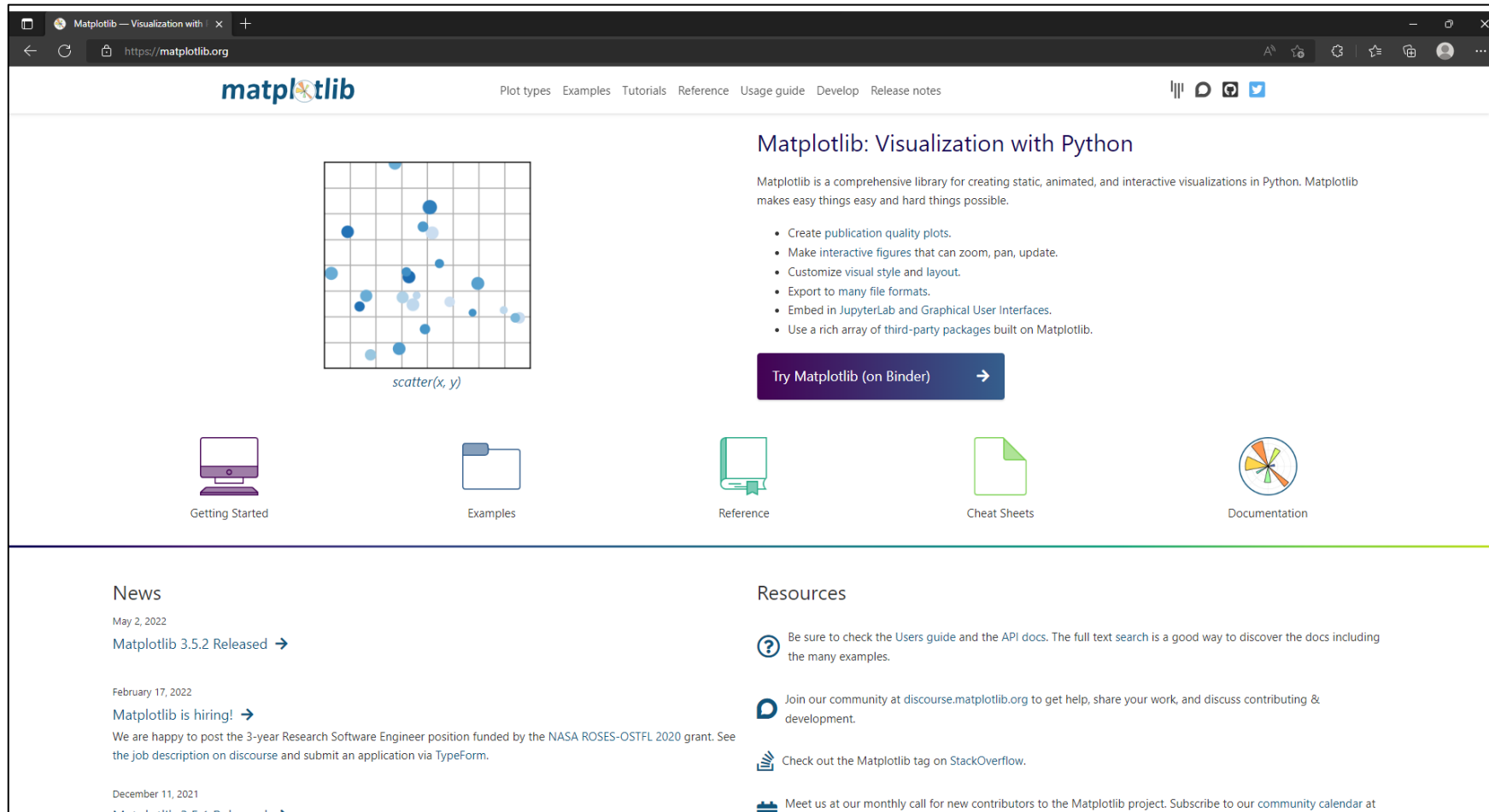


`triplot(x, y)`

01. 기본 그래프 그리기

❖ matplotlib 라이브러리란? (6/7)

- matplotlib 홈페이지: <https://matplotlib.org/>



01. 기본 그래프 그리기

❖ matplotlib 라이브러리란? (7/7) `pip install` 라이브러리 이름(아나콘다는 기본 설치되어 있음)

- matplotlib의 pyplot 모듈 불러오기

```
1 import matplotlib.pyplot
```

```
1 from matplotlib import pyplot
```

- matplotlib 라이브러리의 pyplot 모듈을 'plt'라는 별명으로 부르기(alias)

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

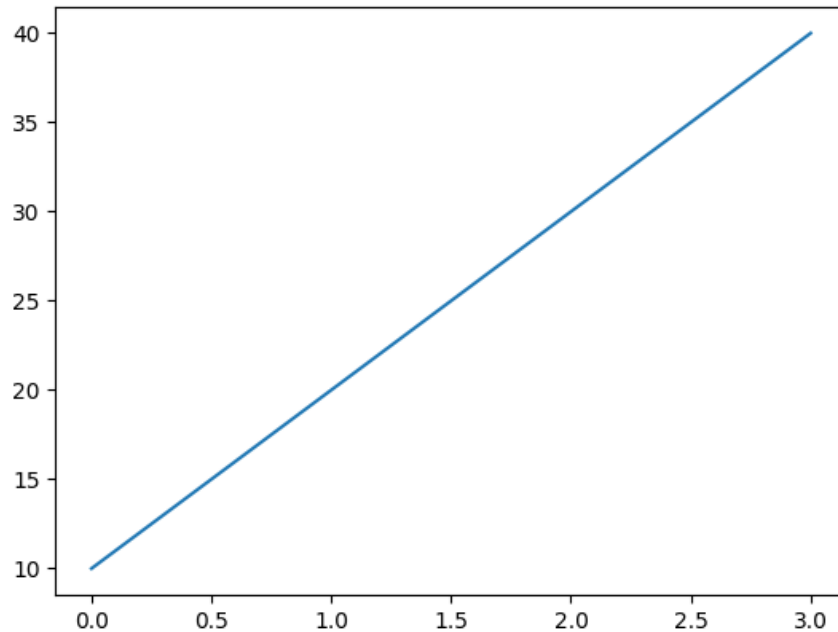
```
1 from matplotlib import pyplot as plt
```

01. 기본 그래프 그리기

❖ 선 그래프 그리기: ① plt.plot([y축 데이터])

```
1 import matplotlib.pyplot as plt
2 plt.plot([10, 20, 30, 40])
3 plt.show()
```

실행결과



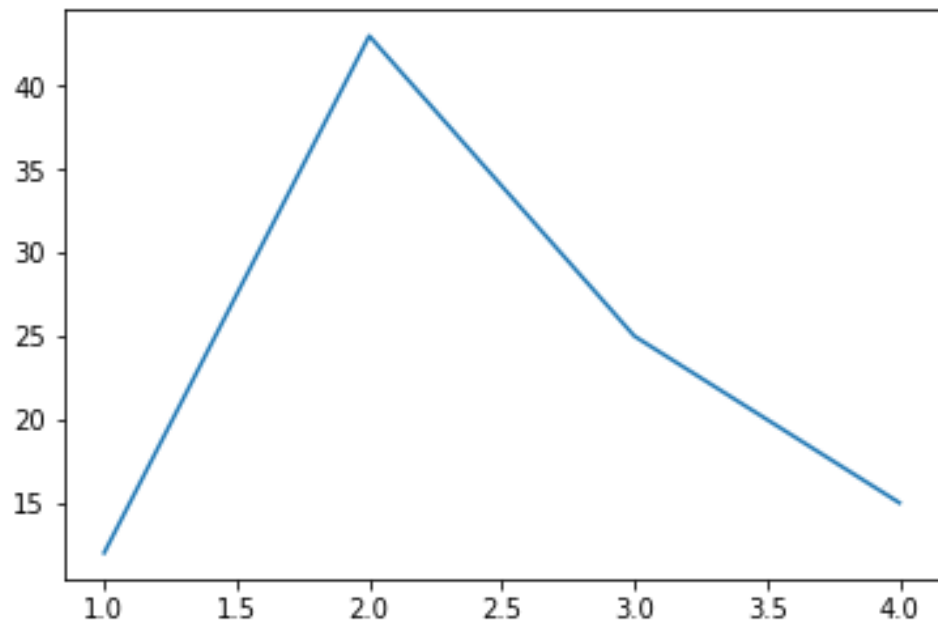
- ✓ 현재 plt.plot()에 입력된 리스트 값은 y축 값임
- ✓ x축 값도 입력하고 싶다면 어떻게 해야 할까?

01. 기본 그래프 그리기

❖ 선 그래프 그리기: ② `plt.plot([x축 데이터], [y축 데이터])`

```
1 import matplotlib.pyplot as plt
2 plt.plot([1, 2, 3, 4], [12, 43, 25, 15])
3 plt.show()
```

실행결과



- ✓ 첫 번째 리스트 값은 x축을 나타냄
- ✓ 두 번째 리스트 값은 y축을 나타냄

01. 기본 그래프 그리기

❖ 선 그래프 그리기: ② plt.plot([x축 데이터], [y축 데이터])

```
1 import matplotlib.pyplot as plt
2 plt.plot([1, 2, 3, 4], [12, 43, 25, 15, 10])
3 plt.show()
```

실행결과

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-5-64a4c9281e2e> in <cell line: 2>()
      1 import matplotlib.pyplot as plt
----> 2 plt.plot([1, 2, 3, 4], [12, 43, 25, 15, 10])
      3 plt.show()
... (중략) ...
ValueError: x and y must have same first dimension, but have shapes (4,) and (5,)
```

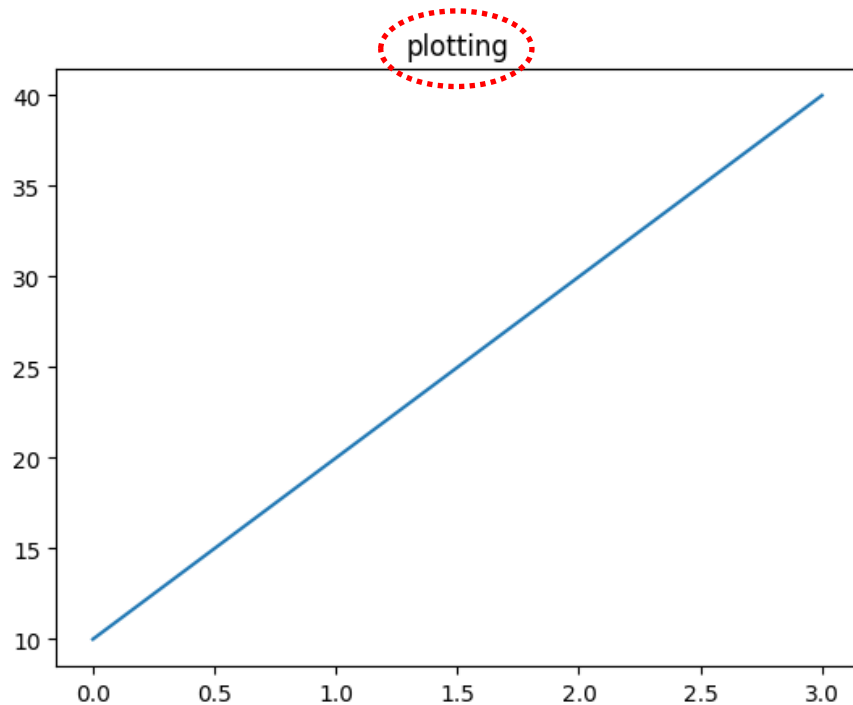
x축 데이터와 y축 데이터의 개수가
같지 않으면, 오류가 발생함

01. 기본 그래프 그리기

❖ 그래프에 제목(Title) 추가하기

```
1 import matplotlib.pyplot as plt
2 plt.title("plotting")
3 plt.plot([10, 20, 30, 40])
4 plt.show()
```

실행결과

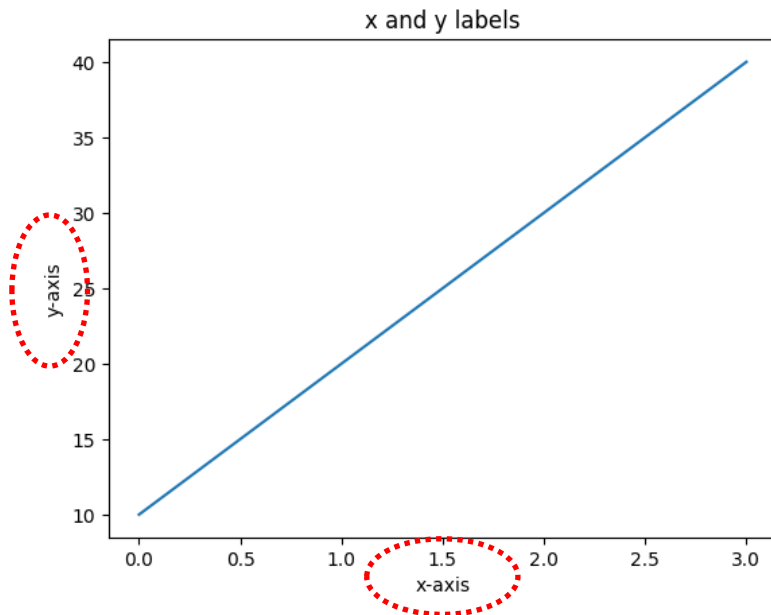


01. 기본 그래프 그리기

❖ 그래프에 x축 및 y축 레이블(Label) 정보 추가하기

```
1 import matplotlib.pyplot as plt
2 plt.title("x and y labels")
3 plt.plot([10, 20, 30, 40])
4 plt.xlabel("x-axis")
5 plt.ylabel("y-axis")
6 plt.show()
```

실행결과



다른 사람이 내가 만든 그래프를 봤을 때, 쉽게 이해할 수 있게 x, y축 레이블 정보를 추가하도록 하자

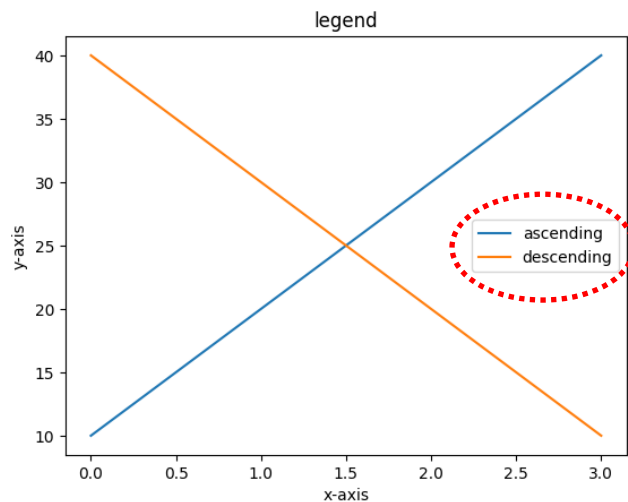
01. 기본 그래프 그리기

❖ 그래프에 범례(legend) 추가하기

```
1 import matplotlib.pyplot as plt
2 plt.title("legend")
3 plt.plot([10, 20, 30, 40], label="ascending")
4 plt.plot([40, 30, 20, 10], label="descending")
5 plt.xlabel("x-axis")
6 plt.ylabel("y-axis")
7 plt.legend()
8 plt.show()
```

- legend
(명사) 전설
(명사) 범례

실행결과



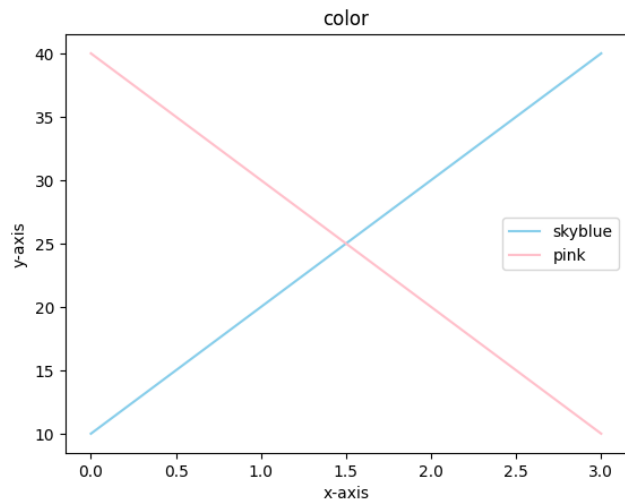
- ✓ 경우에 따라서 그래프를 출력할 때, 흑백으로 출력하는 경우가 있음
- ✓ 이 경우, 색(Color)으로 범례를 구분하는 것이 어려울 수 있음

01. 기본 그래프 그리기

❖ 그래프 색상(Color) 바꾸기

```
1 import matplotlib.pyplot as plt
2 plt.title("color")
3 plt.plot([10, 20, 30, 40], label="skyblue", color="skyblue")
4 plt.plot([40, 30, 20, 10], label="pink", c="pink")
5 plt.xlabel("x-axis")
6 plt.ylabel("y-axis")
7 plt.legend()
8 plt.show()
```

실행결과



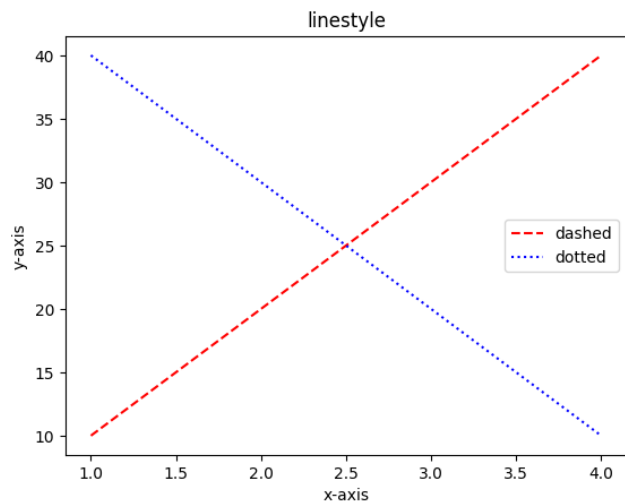
- ✓ color 대신에 축약어인 c로도 특성 지정이 가능함
- ✓ 아래 색상에 대해서는 약자로 표기할 수 있음
- ✓ red → r
- ✓ green → g
- ✓ blue → b
- ✓ black → k
- ✓ yellow → y

01. 기본 그래프 그리기

❖ 그래프 선 모양(Line Style) 바꾸기

```
1 import matplotlib.pyplot as plt
2 plt.title("linestyle")
3 plt.plot([10, 20, 30, 40], label="dashed", c='r', linestyle="--")
4 plt.plot([40, 30, 20, 10], label="dotted", c='b', ls=':')
5 plt.xlabel("x-axis")
6 plt.ylabel("y-axis")
7 plt.legend()
8 plt.show()
```

실행결과



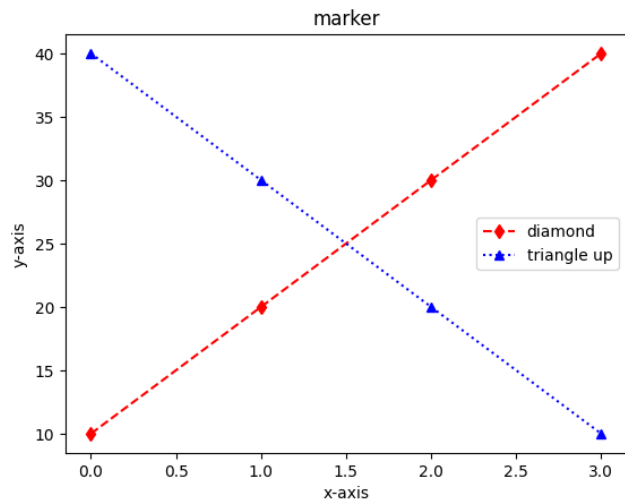
- ✓ linestyle 대신에 축약어인 ls로도 특성 지정이 가능함
- ✓ '-', '--', '-.', ':', " 등의 선 모양을 지정할 수 있음

01. 기본 그래프 그리기

❖ 그래프 마커(Marker) 모양 바꾸기

```
1 import matplotlib.pyplot as plt
2 plt.title("marker")
3 plt.plot([10, 20, 30, 40], label="diamond", c='r', ls="--", marker='d')
4 plt.plot([40, 30, 20, 10], label="triangle up", c='b', ls=':', marker='^')
5 plt.xlabel("x-axis")
6 plt.ylabel("y-axis")
7 plt.legend()
8 plt.show()
```

실행결과

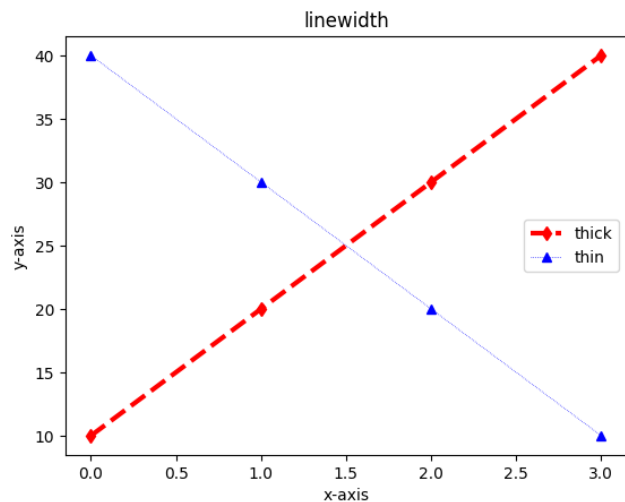


01. 기본 그래프 그리기

❖ 그래프 선 굵기(Line Width) 바꾸기

```
1 import matplotlib.pyplot as plt
2 plt.title("linewidth")
3 plt.plot([10, 20, 30, 40], label="thick", c='r', ls="--", marker='d', linewidth=3)
4 plt.plot([40, 30, 20, 10], label="thin", c='b', ls=':', marker='^', lw=0.5)
5 plt.xlabel("x-axis")
6 plt.ylabel("y-axis")
7 plt.legend()
8 plt.show()
```

실행결과



- ✓ linewidth 대신에 축약어인 lw로도 특성 지정이 가능함
- ✓ 원하는 실수 값으로 지정하면 됨

02. 크리스마스의 기온 변화를 그래프로 그리기

01. 기본 그래프 그리기

03. 기온 데이터를 다양하게 시각화하기

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ① 데이터에 질문하기 (1/2)

- 매년 크리스마스의 최고 기온을 그래프로 그린다면 어떤 모양일까?
- 데이터를 읽어와서 최고 기온 데이터를 출력해 보자

```
1 import csv
2
3 f = open("seoul.csv", encoding="cp949")
4 data = csv.reader(f)
5
6 header = next(data)
7
8 for row in data:
9     print(row[4])
10
11 f.close()
```

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ① 데이터에 질문하기 (2/2)

- 최고 기온 데이터를 리스트에 저장하자

```
1 import csv
2
3 f = open("seoul.csv", encoding="cp949")
4 data = csv.reader(f)
5 header = next(data)
6
7 result = [] # 최고 기온 데이터를 저장할 리스트 생성
8 for row in data:
9     if row[4] != ',': # 최고 기온 데이터가 존재한다면
10         result.append(float(row[4])) # result 리스트에 최고 기온 값을 추가
11
12 f.close()
13
14 print(len(result))
15 print(result)
```


02. 크리스마스의 기온 변화를 그래프로 그리기

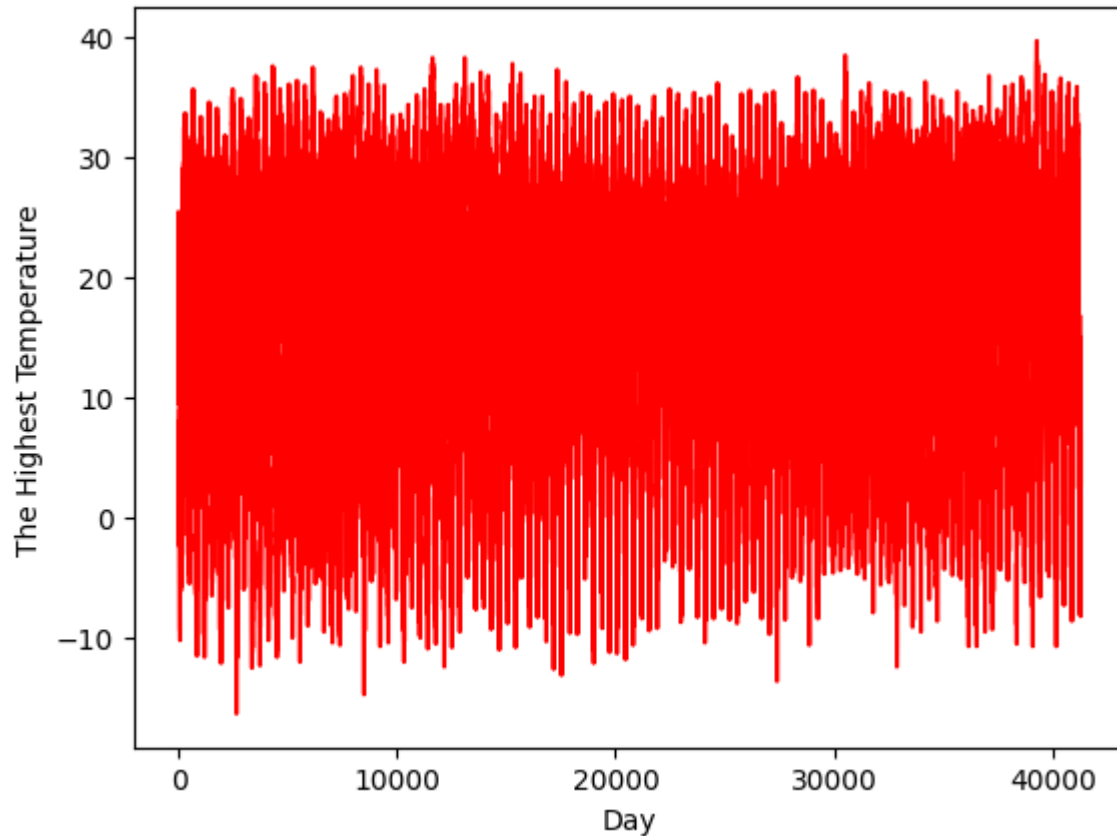
❖ ② 데이터 시각화하기 (1/4)

```
1 import csv
2 import matplotlib.pyplot as plt
3
4 f = open("seoul.csv", encoding="cp949")
5 data = csv.reader(f)
6 header = next(data)
7
8 result = []
9 for row in data:
10     if row[4] != '':
11         result.append(float(row[4]))
12
13 f.close()
14
15 plt.plot(result, 'r')
16 plt.xlabel("Day")
17 plt.ylabel("The Highest Temperature")
18 plt.show()
```

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ② 데이터 시각화하기 (2/4)

실행결과



그래프가 뭉쳐 있어서
가시성이 떨어짐

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ② 데이터 시각화하기 (3/4)

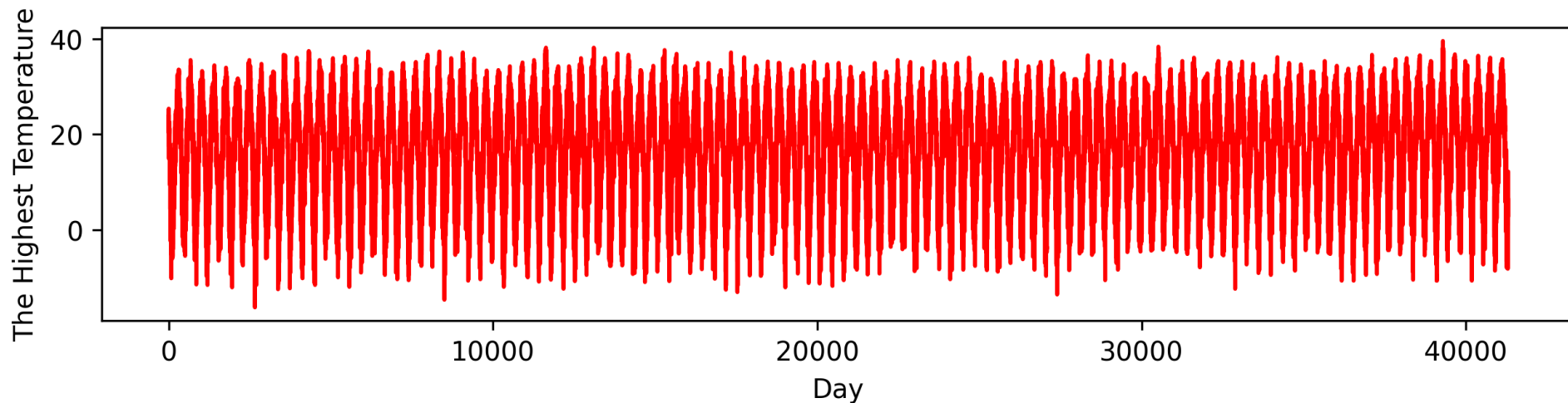
```
1 import csv
2 import matplotlib.pyplot as plt
3
4 f = open("seoul.csv", encoding="cp949")
5 data = csv.reader(f)
6 header = next(data)
7
8 result = []
9 for row in data:
10     if row[4] != '':
11         result.append(float(row[4]))
12
13 f.close()
14 plt.figure(figsize=(10, 2))
15 plt.plot(result, 'r')
16 plt.xlabel("Day")
17 plt.ylabel("The Highest Temperature")
18 plt.show()
```

- ✓ figsize=(가로 길이, 세로 길이)
- ✓ 길이의 단위: 인치(Inch)
- ✓ 1 inch = 2.54 cm

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ② 데이터 시각화하기 (4/4)

실행결과



그래프를 크게 해봐도
의미를 파악하는 것이 어려움

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (1/16)

- 매년 크리스마스(12월 25일)의 최고 기온 데이터를 추출하기 위해서
우선 '2020-12-25' 같은 형태의 날짜 데이터를 '-'를 기준으로 년, 월, 일로 분리해야 함
- 파이썬이 제공하는 split() 함수를 이용해, 분리해 보자

```
1 s = "Happy New Year!"  
2 print(s.split())
```

실행결과

```
['Happy', 'New', 'Year!']
```

**split() 함수는 사용자가 설정하는 특정 문자가 없다면,
기본적으로 공백 문자를 기준으로 문자열을 분리함**

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (2/16)

- `split()` 함수를 이용하여 '2020-12-25' 같은 형태의 날짜 데이터를 '-'를 기준으로 년, 월, 일로 분리해 보자

```
1 date = "2000-12-25"  
2 print(date.split('-'))
```

실행결과

```
['2000', '12', '25']
```

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (3/16)

- 리스트의 인덱싱(Indexing) 기능을 활용하여, 날짜의 년, 월, 일 정보를 각각 추출해 보자

```
1 date = "2000-12-25"  
2 print(date.split('-')[0])  
3 print(date.split('-')[1])  
4 print(date.split('-')[2])
```

실행결과

```
2000  
12  
25
```

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (4/16)

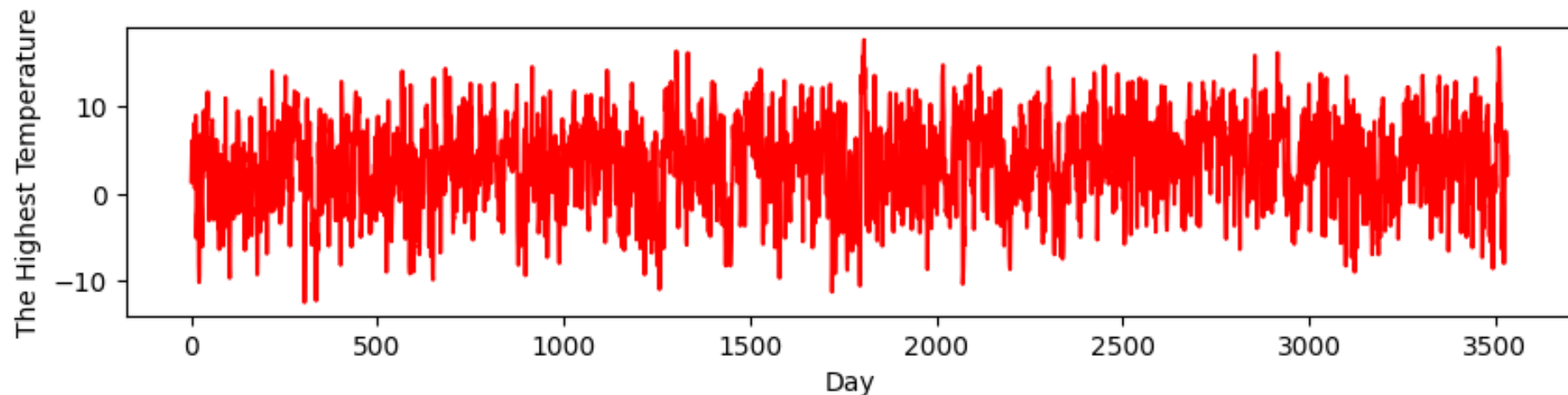
- split() 함수를 이용하여, 매년 12월의 최고 기온 데이터만 추출하여 그래프로 그려 보자

```
1 import csv
2 import matplotlib.pyplot as plt
3
4 f = open("seoul.csv", encoding="cp949")
5 data = csv.reader(f)
6 header = next(data)
7
8 result = []
9 for row in data:
10     if row[4] != '':
11         if row[0].split('-')[1] == "12":
12             result.append(float(row[4]))
13
14 f.close()
15 plt.figure(figsize=(10, 2))
16 plt.plot(result, 'r')
17 plt.xlabel("Day")
18 plt.ylabel("The Highest Temperature")
19 plt.show()
```


02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (5/16)

실행결과



02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (6/16)

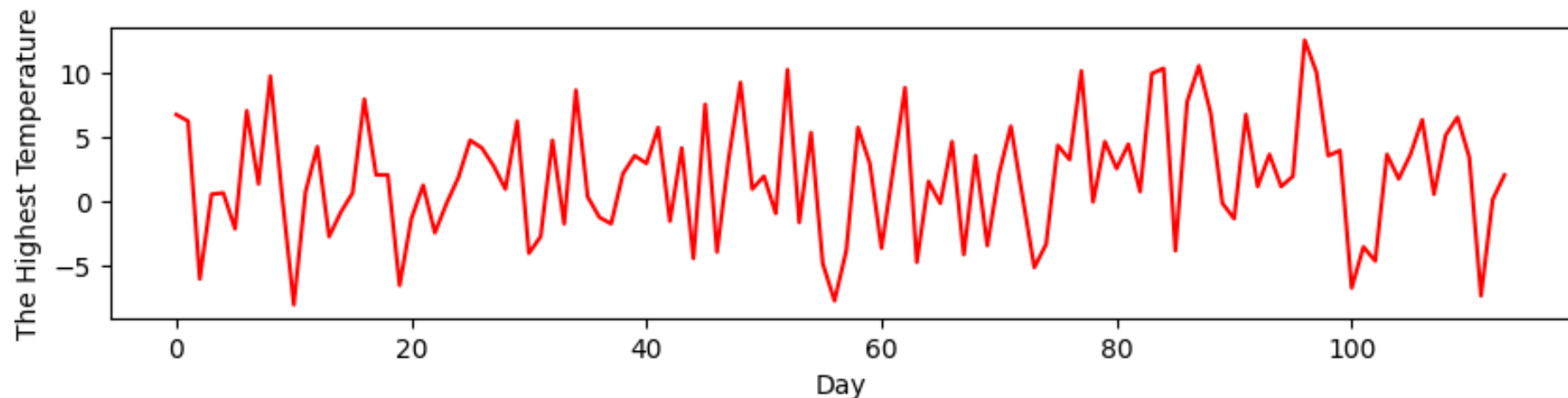
- split() 함수를 이용하여, 매년 12월 25일의 최고 기온 데이터만 추출하여 그래프로 그려 보자

```
1 import csv
2 import matplotlib.pyplot as plt
3
4 f = open("seoul.csv", encoding="cp949")
5 data = csv.reader(f)
6 header = next(data)
7
8 result = []
9 for row in data:
10     if row[4] != '':
11         if row[0].split('-')[1] == "12" and row[0].split('-')[2] == "25":
12             result.append(float(row[4]))
13
14 f.close()
15 plt.figure(figsize=(10, 2))
16 plt.plot(result, 'r')
17 plt.xlabel("Day")
18 plt.ylabel("The Highest Temperature")
19 plt.show()
```

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (7/16)

실행결과



02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (8/16)

- 매년 12월 25일의 평균, 최저, 최고 기온 데이터로 그래프를 그려 보자

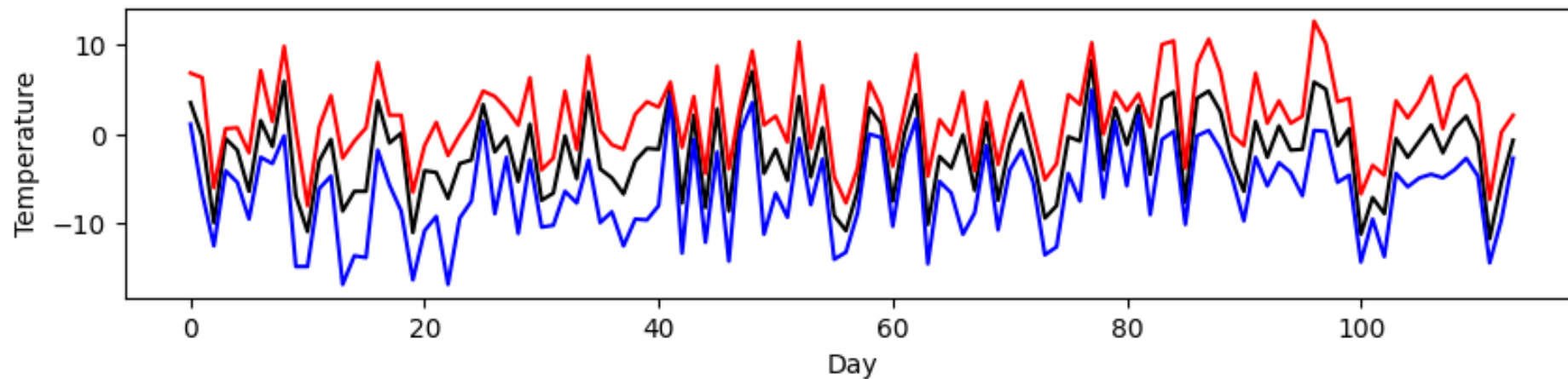
```
1 import csv
2 import matplotlib.pyplot as plt
3 f = open("seoul.csv", encoding="cp949")
4 data = csv.reader(f)
5 header = next(data)
6
7 mean = []          # 평균 기온 데이터를 저장할 리스트 생성
8 low = []           # 최저 기온 데이터를 저장할 리스트 생성
9 high = []          # 최고 기온 데이터를 저장할 리스트 생성
10 for row in data:
11     if row[4] != ',':
12         if row[0].split('-')[1] == "12" and row[0].split('-')[2] == "25":
13             mean.append(float(row[2]))
14             low.append(float(row[3]))
15             high.append(float(row[4]))
16
17 f.close()
18
19
```

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (9/16)

```
20 plt.figure(figsize=(10, 2))
21 plt.plot(mean, 'k')
22 plt.plot(low, 'b')
23 plt.plot(high, 'r')
24 plt.xlabel("Day")
25 plt.ylabel("Temperature")
26 plt.show()
```

실행결과



02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (10/16)

- 그래프를 다듬어 보자
- 한글 폰트 사용하기(Windows 기준)

◆ Malgun Gothic은 “맑은 고딕”임

◆ 만약 macOS 운영체제를 사용하고 있다면, “AppleGothic”이라고 쓰면 됨

```
1 plt.rc("font", family="Malgun Gothic")
2 plt.title("크리스마스의 기온 변화 그래프")
```

◆ 한글 폰트 사용시 마이너스 부호 표현하기

```
1 plt.rcParams["axes.unicode_minus"] = False
```

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (11/16)

- 그래프를 다듬어 보자

```
20 plt.figure(figsize=(10, 4))
21 plt.rc("font", family="Malgun Gothic")
22 plt.rcParams["axes.unicode_minus"] = False
23 plt.title("크리스마스의 기온 변화 그래프")
24 plt.plot(mean, 'k', label="mean")
25 plt.plot(low, 'b', label="low")
26 plt.plot(high, 'r', label="high")
27 plt.xlabel("Day(December 25)")
28 plt.ylabel("Temperature")
29 plt.legend()
30 plt.grid()
31 plt.show()
```

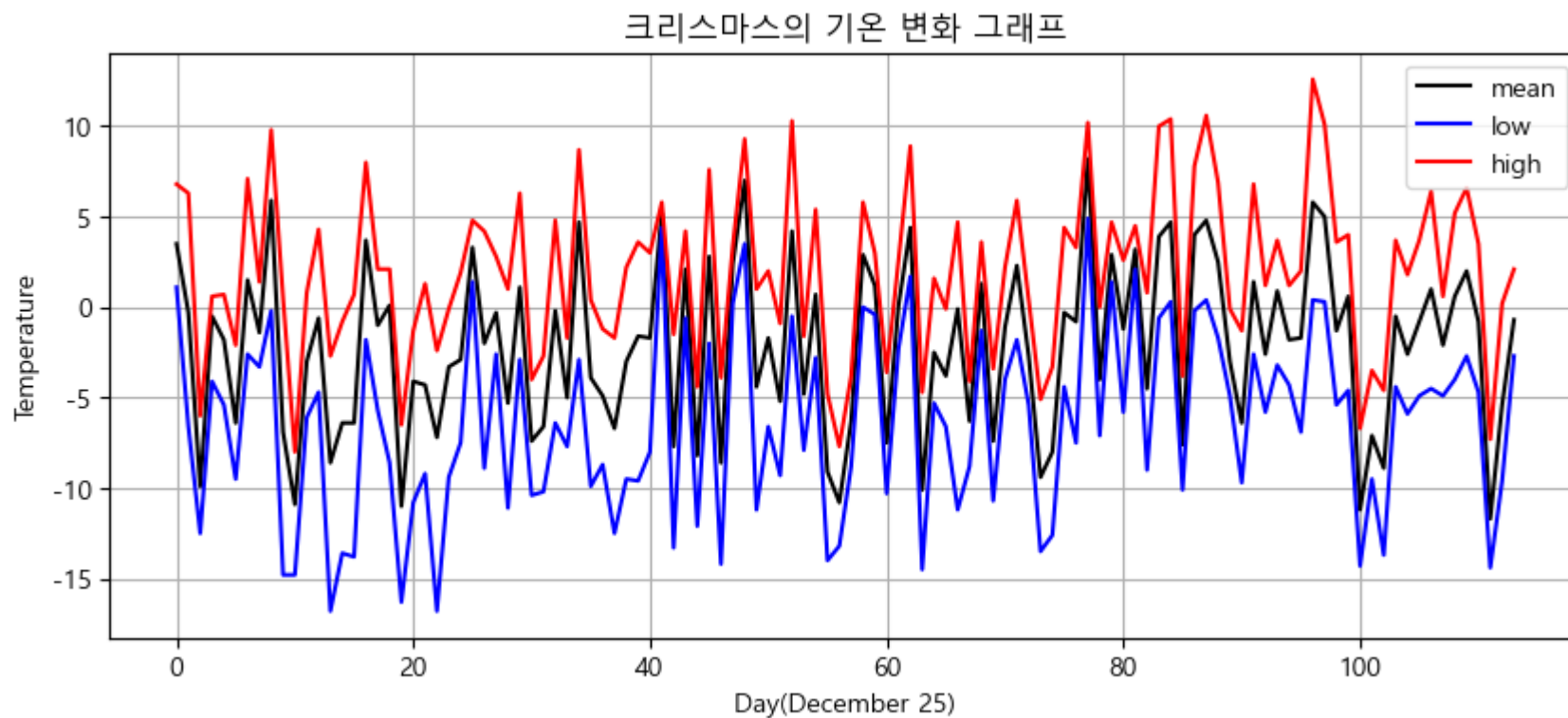
코랩(Colab)은 Linux 운영체제를 사용하고 있어,
나눔 폰트를 설치해야 에러 없이 실행됨

WARNING:matplotlib.font_manager.findfont: Font family 'Malgun Gothic' not found.

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (12/16)

실행결과



02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (13/16)

- 연평균 기온 변화 그래프를 그려 보자

```
1 import csv
2 import matplotlib.pyplot as plt
3
4 f = open("seoul.csv", encoding="cp949")
5 data = csv.reader(f)
6 header = next(data)
7
8 year = []          # 연도 데이터를 저장할 리스트 생성
9 day = []           # 1년 단위로 일별 평균기온 데이터를 저장할 리스트 생성
10 mean = []          # 연평균 기온을 저장할 리스트 생성
11
12
13
14
15
16
17
18
19
```

다음 슬라이드로 넘어가기 전에, 연평균 기온 변화 그래프를 그리려면
어떻게 코드를 작성해야 하는지 고민해 보자

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (14/16)

```
20 for row in data:
21     y = int(row[0].split('-')[0]) # 연도 데이터 추출
22     if y not in [1907, 1950, 1951, 1952, 1953, 2024]:
23         if row[2] != '': # 평균기온 데이터가 결측치가 아니라면
24             day.append(float(row[2]))
25             if row[0].split('-')[1] == "12" and row[0].split('-')[2] == "31":
26                 # 12월 31일이 되면, 지금까지 day 리스트에 저장했던 평균기온 데이터의
27                 # 평균 값을 계산해 mean 리스트에 추가
28                 mean.append(sum(day)/len(day))
29                 year.append(y) # 연도 데이터 y를 year 리스트에 추가
30                 day = [] # 다음 연도 계산을 위해, 리스트를 비움
31     elif y == 2024:
32         break
33
34 f.close()
35
36
37
38
39
```

02. 크리스마스의 기온 변화를 그래프로 그리기

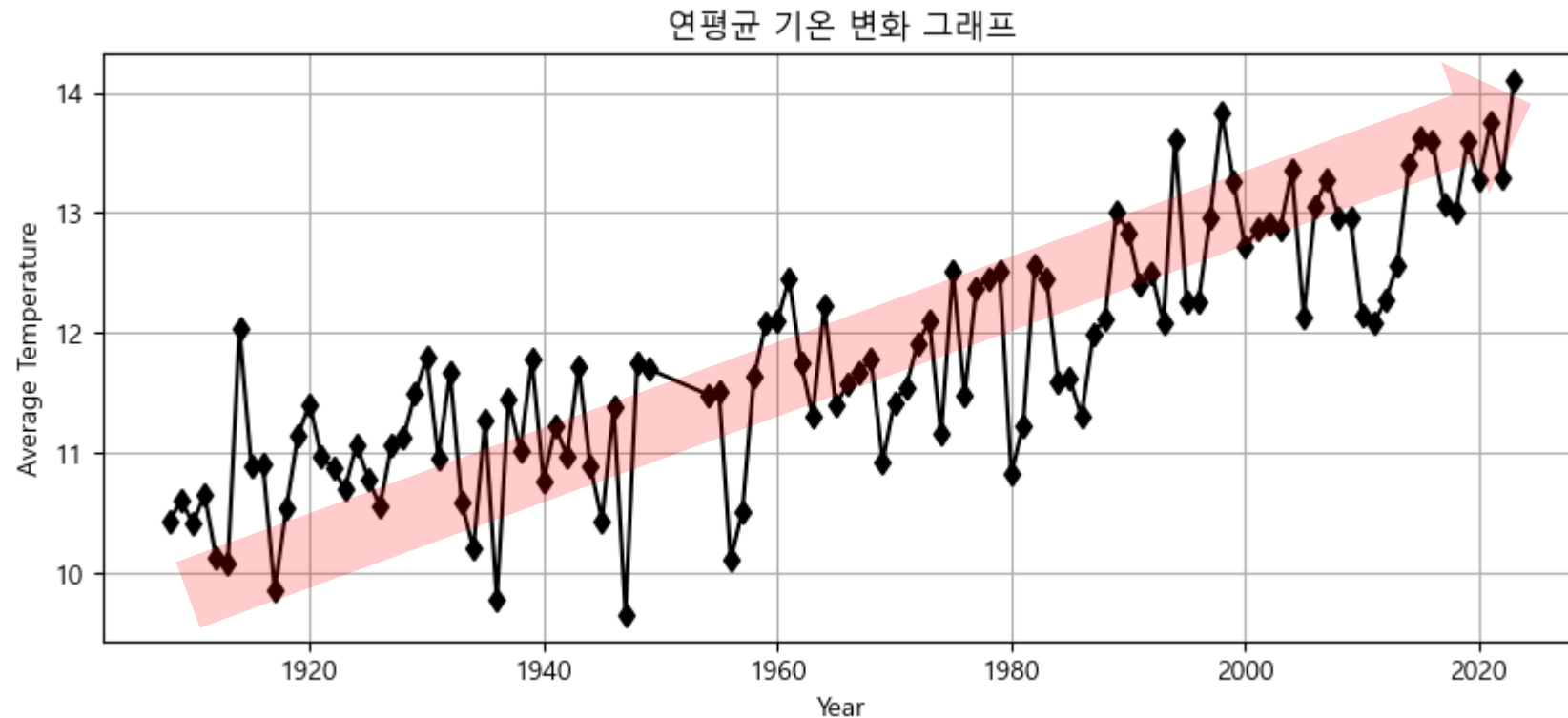
❖ ③ 날짜 데이터 추출하기 (15/16)

```
40 plt.figure(figsize=(10, 4))
41 plt.rc("font", family="Malgun Gothic")
42 plt.rcParams["axes.unicode_minus"] = False
43 plt.title("연평균 기온 변화 그래프")
44 plt.plot(year, mean, 'k', marker='d')
45 plt.xlabel("Year")
46 plt.ylabel("Average Temperature")
47 plt.grid()
48 plt.show()
```

02. 크리스마스의 기온 변화를 그래프로 그리기

❖ ③ 날짜 데이터 추출하기 (16/16)

실행결과



서울의 연평균 기온이 상승하고 있음을 알 수 있음

1. 히스토그램

2. 상자 그림 Box plot

03. 기온 데이터를 다양하게 시각화하기

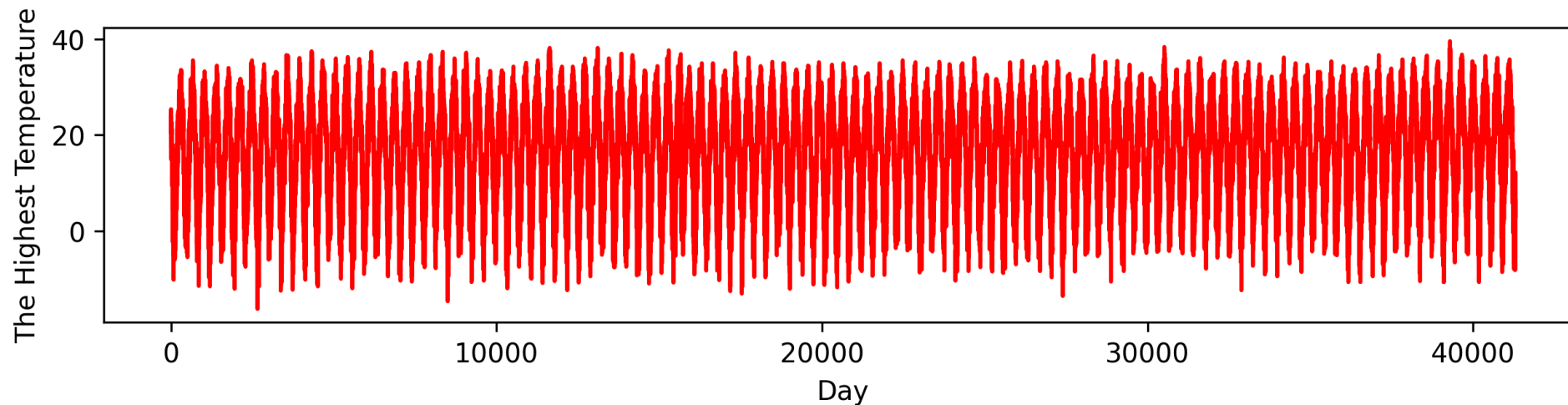
01. 기본 그래프 그리기

02. 크리스마스의 기온 변화를 그래프로 그리기

03. 기온 데이터를 다양하게 시각화하기

❖ ① 데이터에 질문하기

- 아래 그래프는 최고 기온 데이터를 추출하여 그린 결과임



이 그래프만 보서는 특별한 정보를 얻을 수가 없음.
꺾은선 그래프가 아닌 다른 형태로 시각화 하면 어떨까?

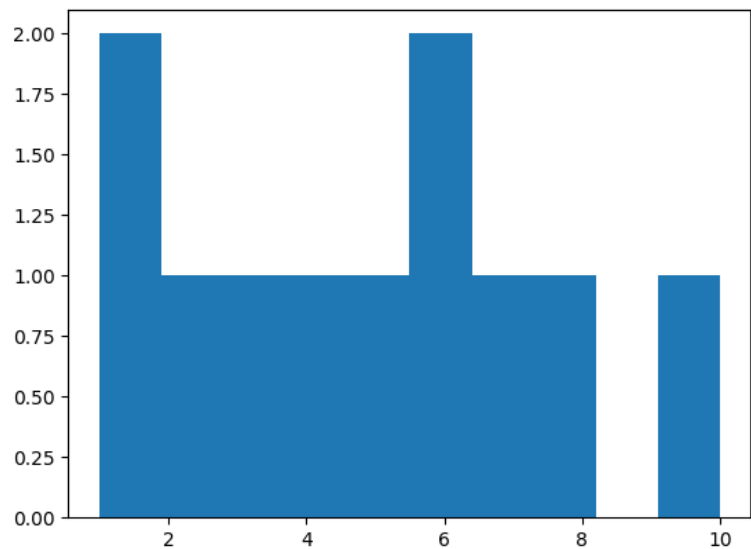
03. 기온 데이터를 다양하게 시각화하기

❖ ② 히스토그램 (1/6)

- 히스토그램(Histogram)은 데이터의 분포 상태를 직사각형 모양의 막대 그래프로 나타냄
- 데이터의 빈도에 따라 직사각형의 높이가 결정됨 → hist() 함수

```
1 import matplotlib.pyplot as plt
2 plt.figure()
3 plt.hist([1, 1, 2, 3, 4, 5, 6, 6, 7, 8, 10])
4 plt.show()
```

실행결과



03. 기온 데이터를 다양하게 시각화하기

❖ ② 히스토그램 (2/6)

- 주사위 시뮬레이션

- ◆ Step 1) 주사위를 굴림 1과 6사이의 랜덤 숫자를 만듦

- ◆ Step 2) 나온 결과를 기록함 리스트에 저장함

- ◆ Step 3) Steps 1-2의 과정을 n번 반복함 for 반복문

- ◆ Step 4) 주사위의 눈이 나온 횟수를 히스토그램으로 그림

03. 기온 데이터를 다양하게 시각화하기

❖ ② 히스토그램 (3/6)

- 1과 6사이의 랜덤 숫자 만들기

```
1 import random
2 print(random.randint(1, 6))
```

실행결과

2

매 실행마다 값이 달라짐

03. 기존 데이터를 다양하게 시각화하기

❖ ② 히스토그램 (4/6)

- 주사위 시뮬레이션 5회 수행하기

```
1 import random
2 dice = []
3 for j in range(5):
5     dice.append(random.randint(1, 6))
6
7 print(dice)
```

실행결과

[4, 1, 2, 1, 6]



매 실행마다 값이 달라짐

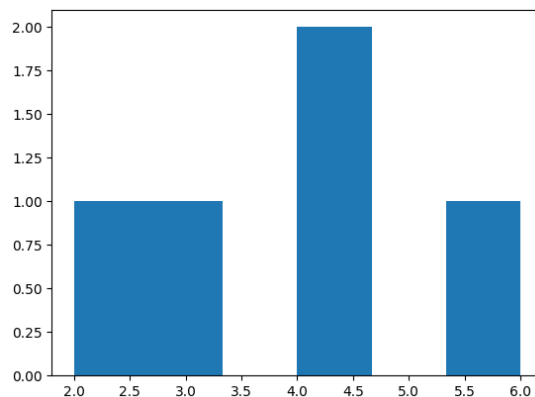
03. 기온 데이터를 다양하게 시각화하기

❖ ② 히스토그램 (5/6)

- 주사위 시뮬레이션 5회 수행 결과를 히스토그램으로 시각화하기

```
1 import random
2 import matplotlib.pyplot as plt
3 dice = []
5 for j in range(5):
6     dice.append(random.randint(1, 6))
7
8 plt.figure()
9 plt.hist(dice, bins=6)
10 plt.show()
```

실행결과

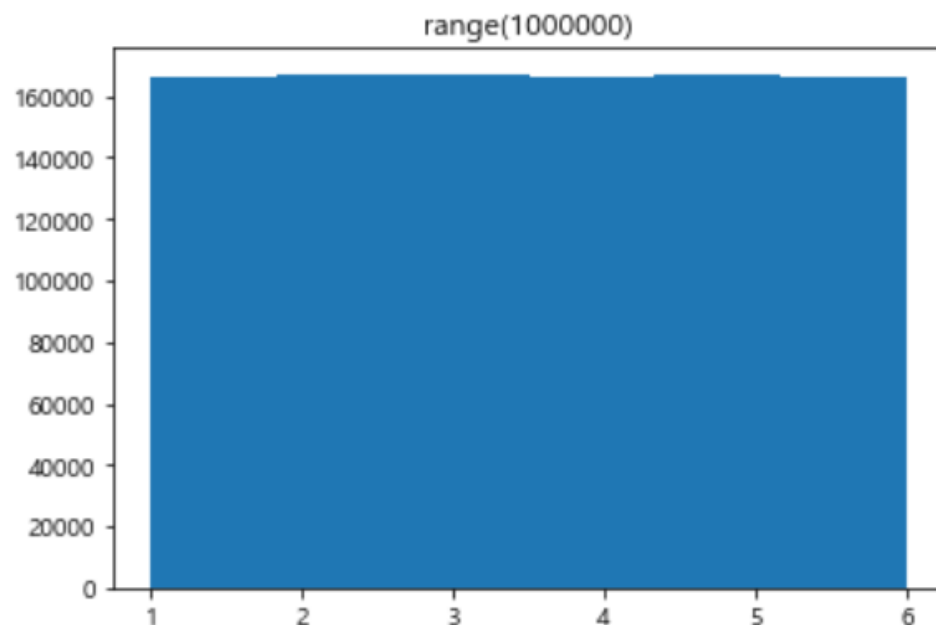
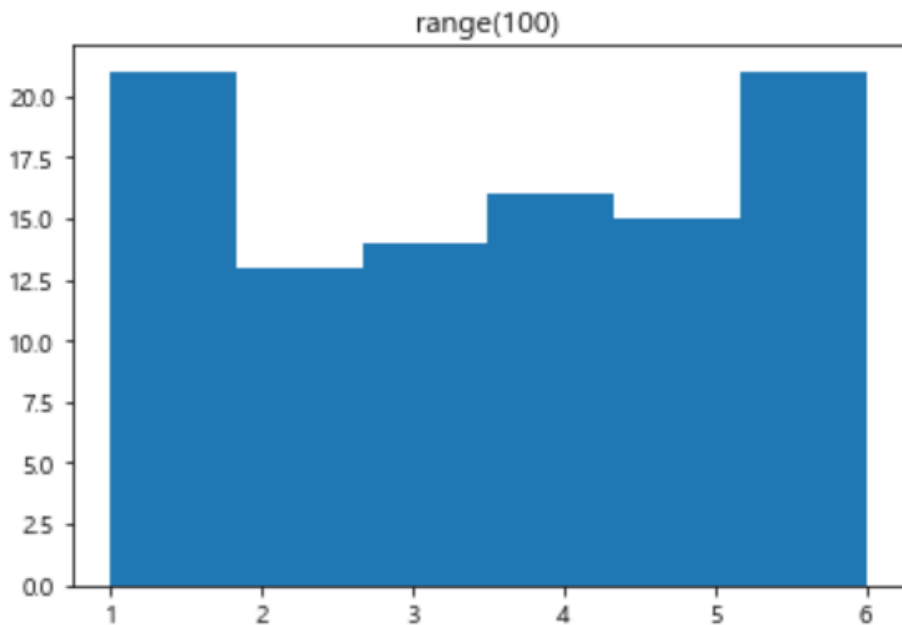


매 실행마다 달라짐

03. 기온 데이터를 다양하게 시각화하기

❖ ② 히스토그램 (6/6)

- `range(5)`를 `range(100)`, `range(1000000)`으로 수정하여, 히스토그램 결과를 확인해 보자



주사위를 던지는 횟수가 늘어날 수록 주사위의 특정 숫자가 나오는 횟수가 전체의 1/6에 가까워 짐을 알 수 있음

큰 수의 법칙(Law of Large Numbers, LLN)

03. 기온 데이터를 다양하게 시각화하기

❖ ③ 기온 데이터를 히스토그램으로 표현하기 (1/6)

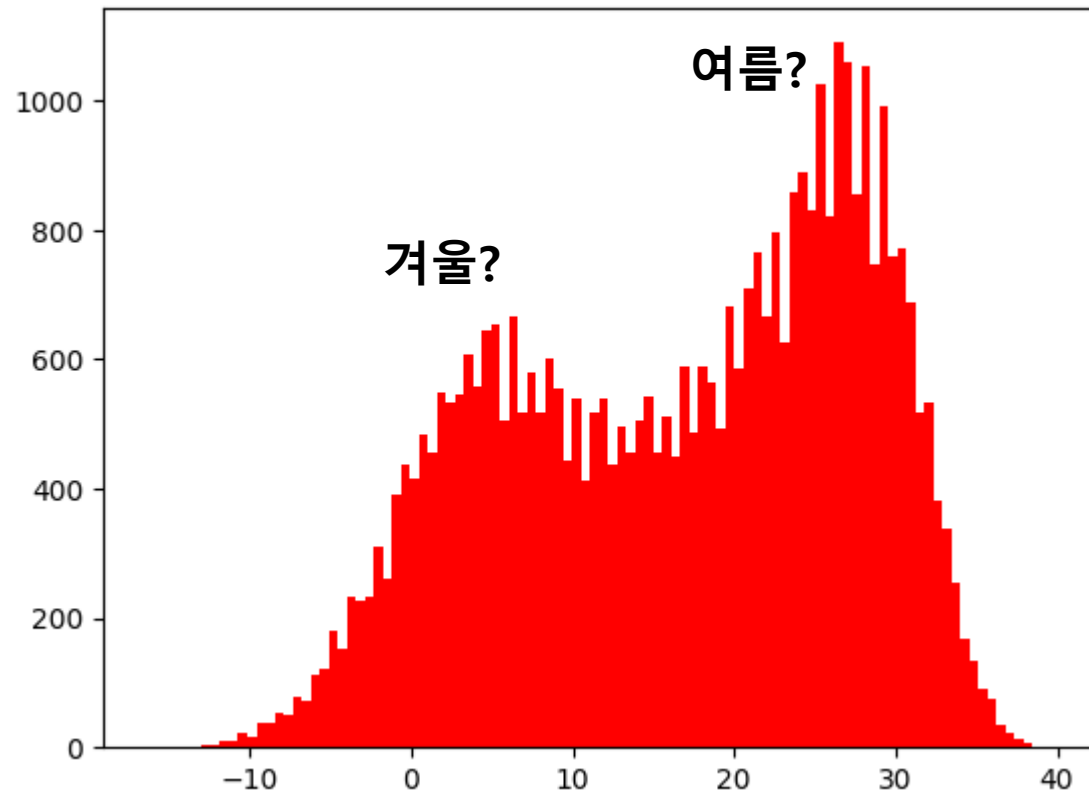
- 최고 기온 데이터를 히스토그램으로 시각화하기

```
1 import csv
2 import matplotlib.pyplot as plt
3
4 f = open("seoul.csv", encoding="cp949")
5 data = csv.reader(f)
6 next(data)
7
8 result = []
9 for row in data:
10     if row[4] != '':
11         result.append(float(row[4]))
12
13 f.close()
14
15 plt.figure()
16 plt.hist(result, bins=100, color='r')
17 plt.show()
```

03. 기온 데이터를 다양하게 시각화하기

❖ ③ 기온 데이터를 히스토그램으로 표현하기 (2/6)

실행결과



03. 기온 데이터를 다양하게 시각화하기

❖ ③ 기온 데이터를 히스토그램으로 표현하기 (3/6)

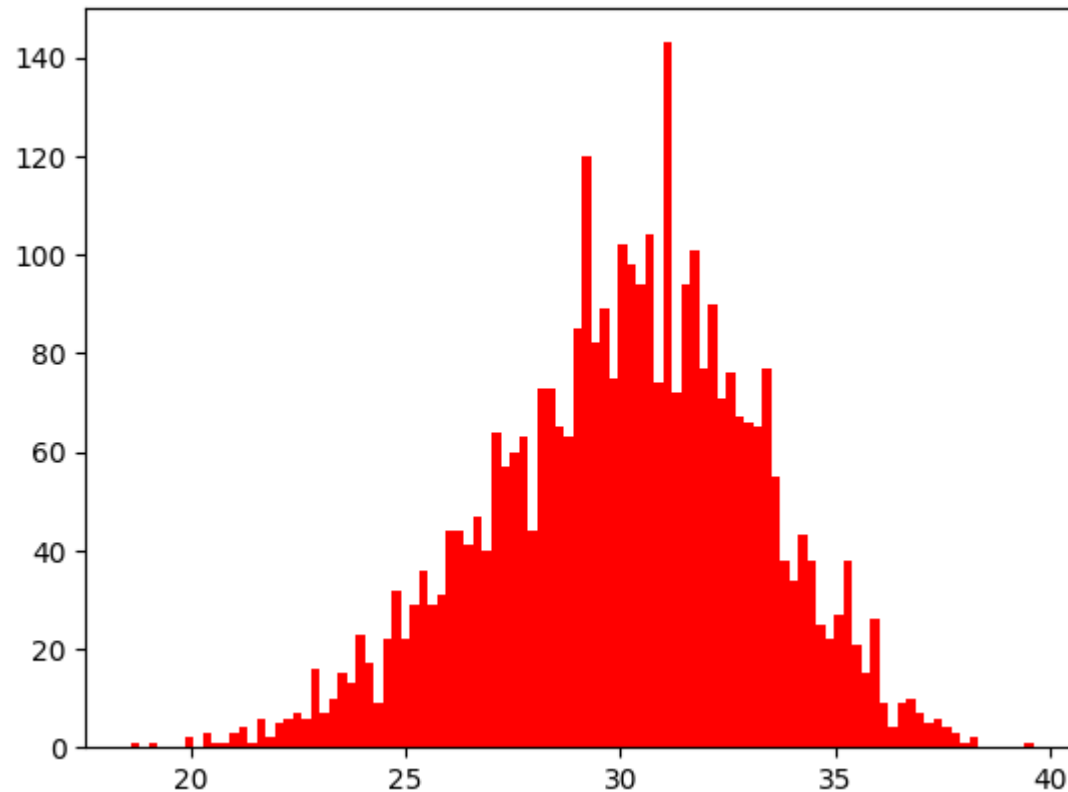
- 8월의 최고 기온 데이터를 히스토그램으로 시각화하기

```
1 import csv
2 import matplotlib.pyplot as plt
3
4 f = open("seoul.csv", encoding="cp949")
5 data = csv.reader(f)
6 next(data)
7
8 aug = []
9 for row in data:
10     month = row[0].split('-')[1]
11     if row[4] != '':
12         if month == "08":
13             aug.append(float(row[4]))
14
15 f.close()
16
17 plt.figure()
18 plt.hist(aug, bins=100, color='r')
19 plt.show()
```

03. 기온 데이터를 다양하게 시각화하기

❖ ③ 기온 데이터를 히스토그램으로 표현하기 (4/6)

실행결과



03. 기온 데이터를 다양하게 시각화하기

❖ ③ 기온 데이터를 히스토그램으로 표현하기 (5/6)

- 1월과 8월의 최고 기온 데이터를 히스토그램으로 시각화하기

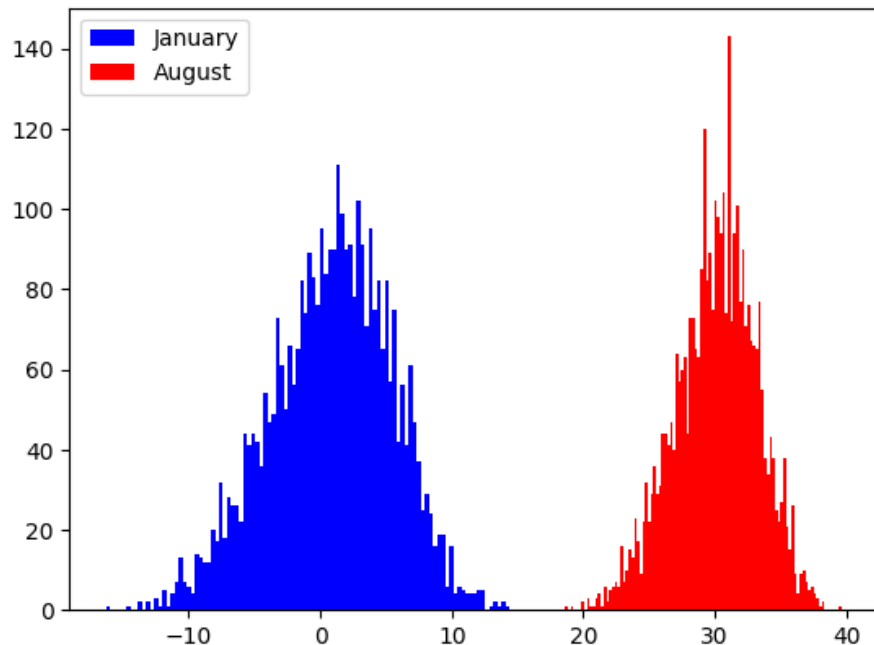
```
1 import csv
2 import matplotlib.pyplot as plt
3
4 f = open("seoul.csv", encoding="cp949")
5 data = csv.reader(f)
6 next(data)
7
8 jan, aug = [], []
9 for row in data:
10     month = row[0].split('-')[1]
11     if row[4] != '':
12         if month == "01":
13             jan.append(float(row[4]))
14         elif month == "08":
15             aug.append(float(row[4]))
16
17 f.close()
18
19
```

03. 기온 데이터를 다양하게 시각화하기

❖ ③ 기온 데이터를 히스토그램으로 표현하기 (6/6)

```
20 plt.figure()  
21 plt.hist(jan, bins=100, color='b', label="January")  
22 plt.hist(aug, bins=100, color='r', label="August")  
23 plt.legend()  
24 plt.show()
```

실행결과



03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (1/11)

- 상자 그림(Boxplot)은 데이터에서 얻어낸 최대값, 최소값, 상위 1/4, 2/4(중앙), 3/4에 위치한 값을 보여주는 그래프임

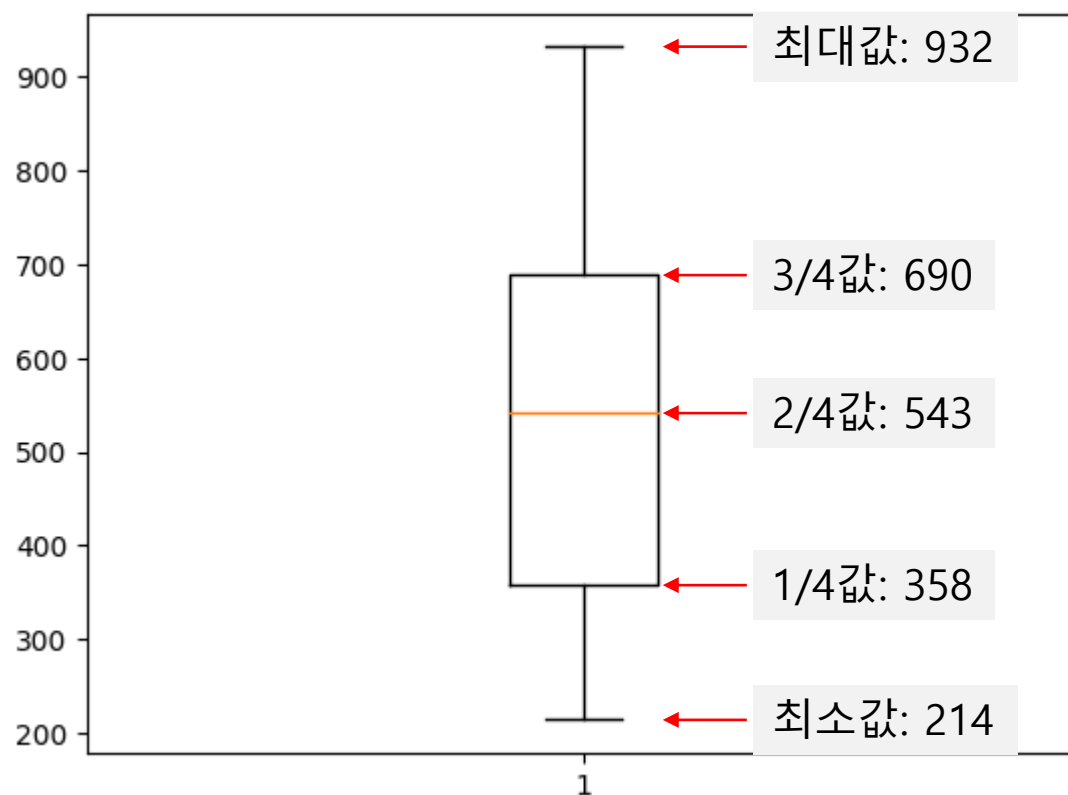
```
1 import matplotlib.pyplot as plt
2 import random
3
4 result = []
5 for j in range(13):
6     result.append(random.randint(1,1000))
7
8 print(sorted(result))
9
10 plt.figure()
11 plt.boxplot(result)
12 plt.show()
```

03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (2/11)

실행결과

[214, 234, 260, 358, 421, 542, 543, 594, 646, 690, 792, 813, 932]



03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (3/11)

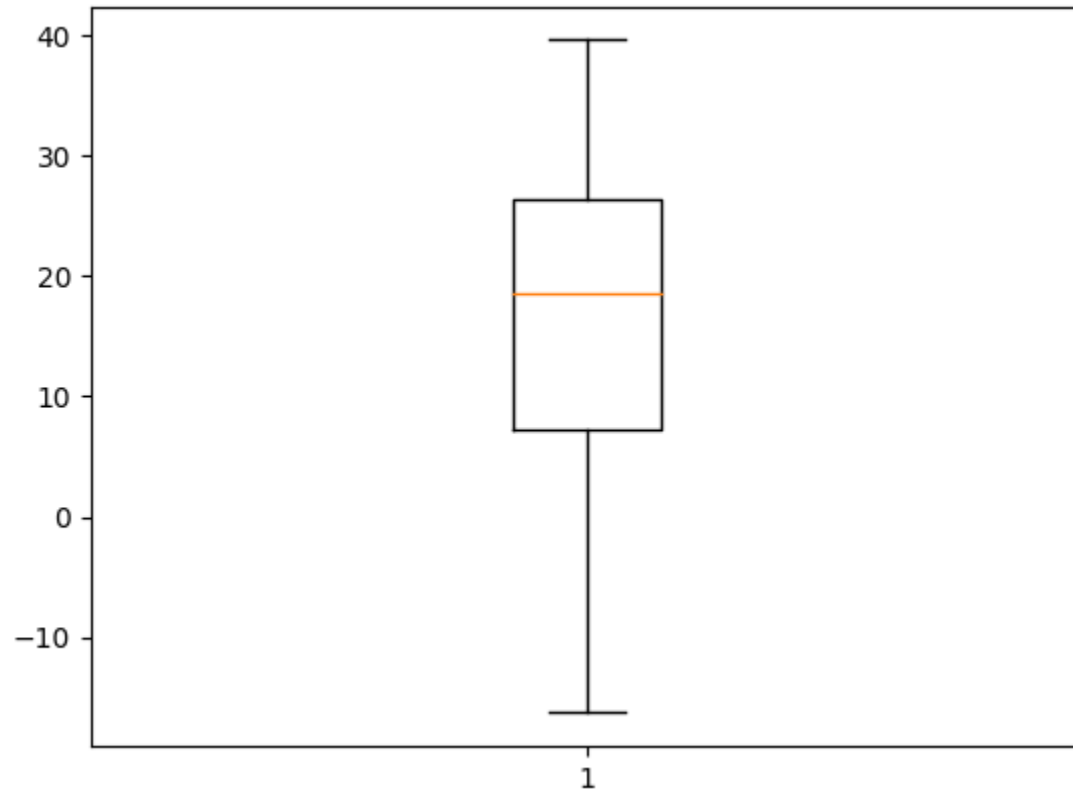
- 최고 기온 데이터를 상자 그림으로 시각화하기

```
1 import csv
2 import matplotlib.pyplot as plt
3
4 f = open("seoul.csv", encoding="cp949")
5 data = csv.reader(f)
6 next(data)
7
8 result = []
9 for row in data:
10     if row[4] != '':
11         result.append(float(row[4]))
12
13 f.close()
14
15 plt.figure()
16 plt.boxplot(result)
17 plt.show()
```

03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (4/11)

실행결과



03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (5/11)

- 1월과 8월의 최고 기온 데이터를 상자 그림으로 시각화하기

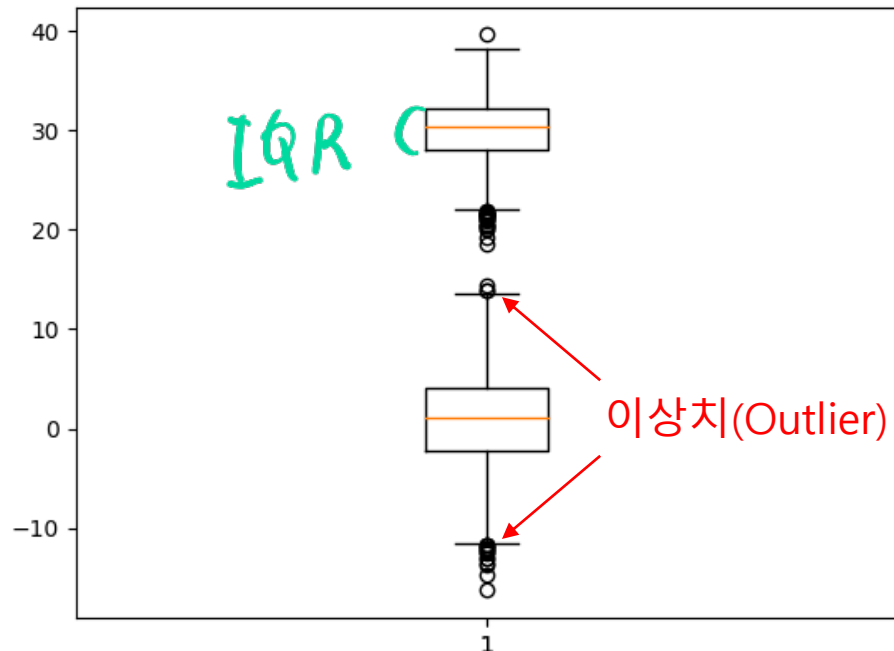
```
1 import csv
2 import matplotlib.pyplot as plt
3
4 f = open("seoul.csv", encoding="cp949")
5 data = csv.reader(f)
6 next(data)
7
8 jan, aug = [], []
9 for row in data:
10     month = row[0].split('-')[1]
11     if row[4] != '':
12         if month == "01":
13             jan.append(float(row[4]))
14         elif month == "08":
15             aug.append(float(row[4]))
16
17 f.close()
18
19
```

03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (6/11)

```
20 plt.figure()  
21 plt.boxplot(jan)  
22 plt.boxplot(aug)  
23 plt.show()
```

실행결과



사분범위(IQR) = 3사분위 - 1사분위
1사분위 - IQR * 1.5
3사분위 + IQR * 1.5

1월과 8월 데이터를
분리하여 표현할 순 없을까?

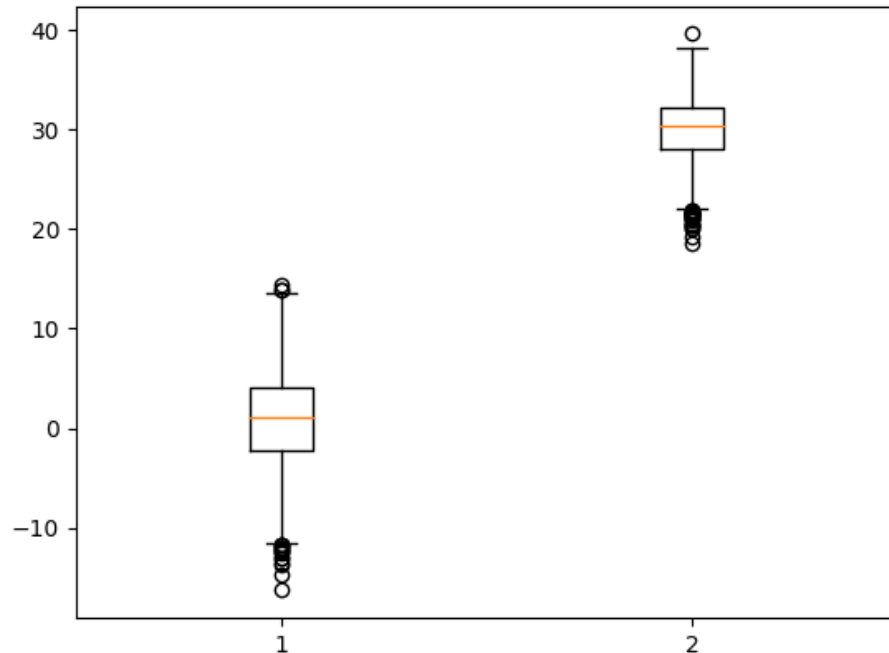
03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (7/11)

- 1월과 8월의 최고 기온 데이터를 분리하여 상자 그림으로 시각화하기

```
20 plt.figure()  
21 plt.boxplot([jan, aug])  
22 plt.show()
```

실행결과



03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (8/11)

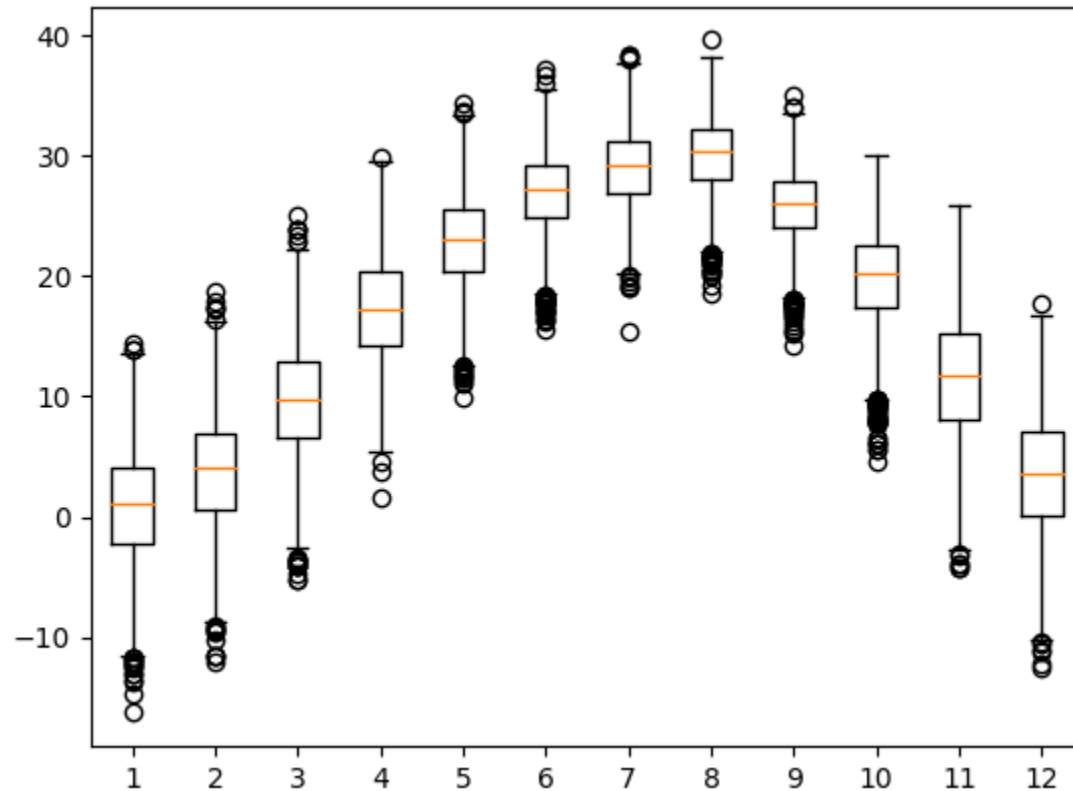
- 1월부터 12월까지의 최고 기온 데이터를 상자 그림으로 시각화하기

```
1 import csv
2 import matplotlib.pyplot as plt
3 f = open("seoul.csv", encoding="cp949")
4 data = csv.reader(f)
5 next(data)
6
7 # 월별 데이터를 저장할 리스트 month (12개 생성)
8 month = [[], [], [], [], [], [], [], [], [], [], [], []]
9 for row in data:
10     if row[4] != '':
11         # 월과 같은 번호의 인덱스에 월별 데이터를 추가(예: 1월 → month[0])
12         month[int(row[0].split('-')[1])-1].append(float(row[4]))
13
14 f.close()
15
16 plt.figure()
17 plt.boxplot(month)
18 plt.show()
```

03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (9/11)

실행결과



03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (10/11)

- 1월 일별 기온 데이터를 상자 그림으로 시각화하기

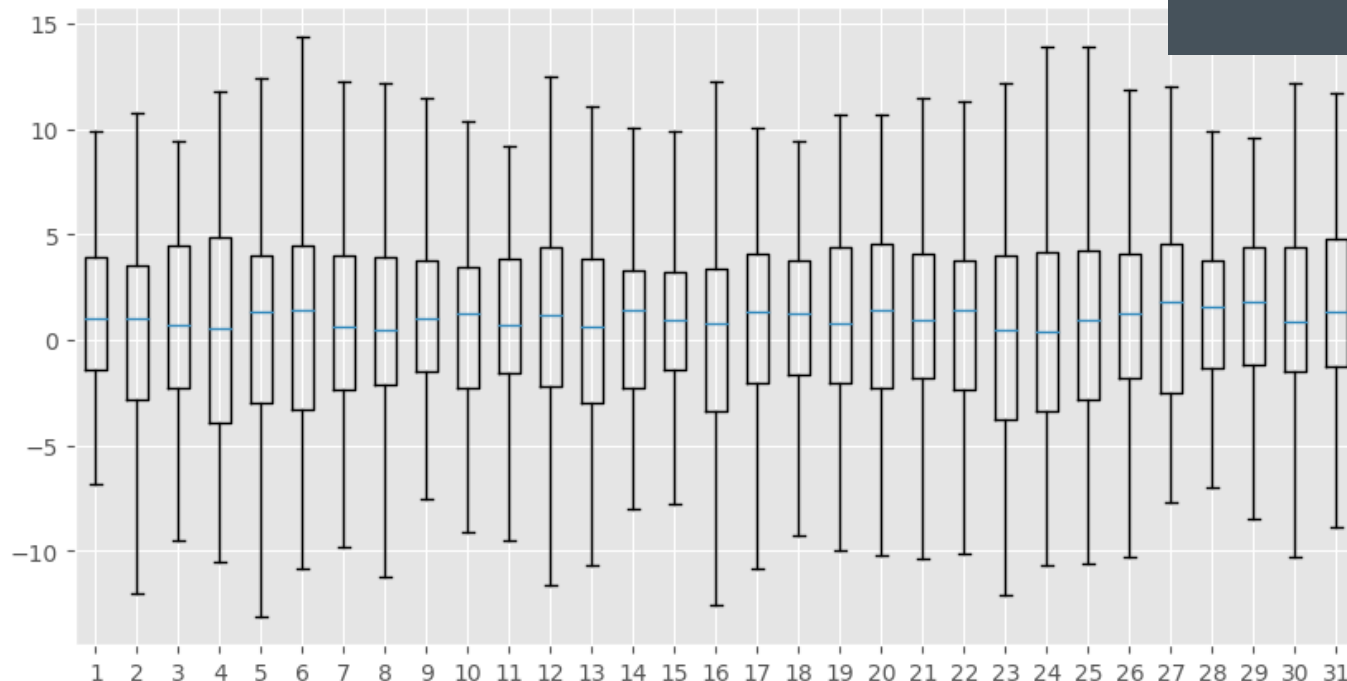
```
1 import csv
2 import matplotlib.pyplot as plt
3 f = open("seoul.csv", encoding="cp949")
4 data = csv.reader(f)
5 next(data)
6
7 day = []
8 for _ in range(31):
9     day.append([])
10
11 for row in data:
12     if row[4] != '':
13         if row[0].split('-')[1] == "01":
14             # 일과 같은 번호의 인덱스에 일별 데이터 추가(예: 1일 → day[0])
15             day[int(row[0].split('-')[2])-1].append(float(row[4]))
16
17 f.close()
18
19
```

03. 기온 데이터를 다양하게 시각화하기

❖ ④ 기온 데이터를 상자 그림으로 표현하기 (11/11)

```
20 plt.style.use("ggplot")           # 그래프 스타일 지정
21 plt.figure(figsize=(10, 5))      # 그래프 크기 수정
22 plt.boxplot(day, showfliers=False) # 이상치(Outlier) 표시 생략
23 plt.show()
```

실행결과



그래프를 기본값 스타일로 되돌리는 방법:
`plt.style.use("default")`

- ❖ 01. 기본 그래프 그리기
- ❖ 02. 크리스마스의 기온 변화를 그래프로 그리기
- ❖ 03. 기온 데이터를 다양하게 시각화하기

THANK YOU!

Q & A

- Name: 권범
- Office: 동덕여자대학교 인문관 B821호
- Phone: 02-940-4752
- E-mail: bkwon@dongduk.ac.kr