

- CSV 파일이란
- 공공 데이터를 이용하여 기온 분석하기
- 다양한 그래프 알아보기

공공 데이터를 이용하여 기온 분석하기

■ 공공데이터 가져오기

기상자료개방포털

데이터

기후통계분석

카탈로그

소통과 참여

기후통계
분석

기후분석

조건별통계

기온분석

강수량분석

순위값

장마

평년값

우리나라 기후평년값

세계기후평년값

바람장미

다중지점통계

현상일수

폭염일수

열대야일수

황사일수

강수일수

24절기

응용기상분석도구

데이터 개방
오픈 API

조건별통계

Home > 기후통계분석 > 기후분석 > 조건별통계 > 조건별통계

• 분류 지상 • 지역/지점 서울 선택 • 요소 기온

• 기간 월 2008 년 ~ 2018 년

• 조건

☐ 요소 평균기온 < 선택

☒ 월 01 ~ 12 월

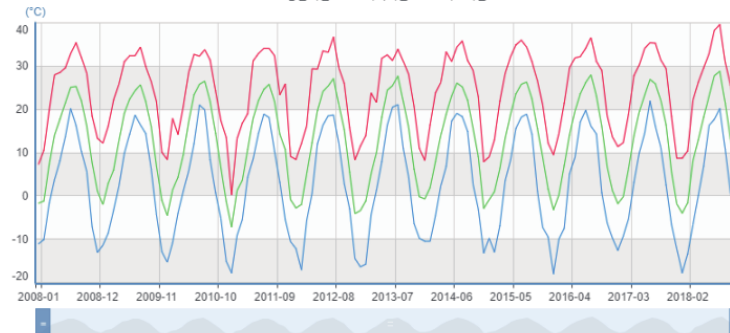
☐ 일 01 ~ 31 일

☐ 계절 봄

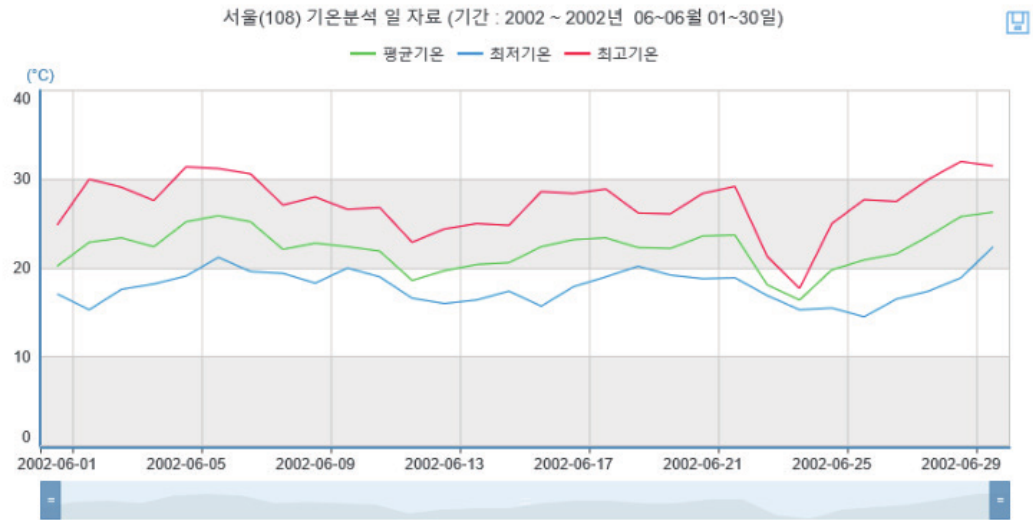
검색

서울(108) 기온분석 월 자료 (기간 : 2008 ~ 2018년 01~12월)

— 평균기온 — 최저기온 — 최고기온



공공 데이터를 이용하여 기온 분석하기



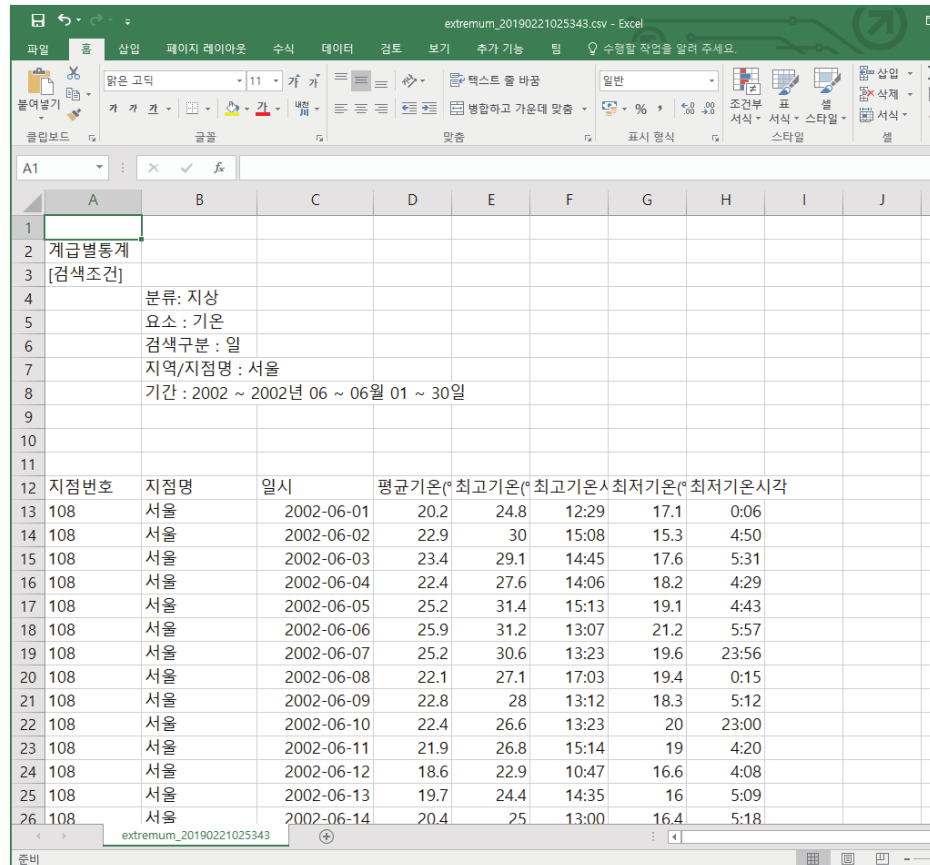
CSV 다운로드

지점	지점명	일시	평균기온(℃)	최고기온(℃)	최고기온시각	최저기온(℃)	최저기온시각
108	서울	2002-06-01	20.2	24.8	12:29	17.1	00:06
108	서울	2002-06-02	22.9	30.0	15:08	15.3	04:50
108	서울	2002-06-03	23.4	29.1	14:45	17.6	05:31
108	서울	2002-06-04	22.4	27.6	14:06	18.2	04:29
108	서울	2002-06-05	25.2	31.4	15:13	19.1	04:43
108	서울	2002-06-06	25.9	31.2	13:07	21.2	05:57

공공 데이터를 이용하여 기온 분석하기

■ CSV 파일

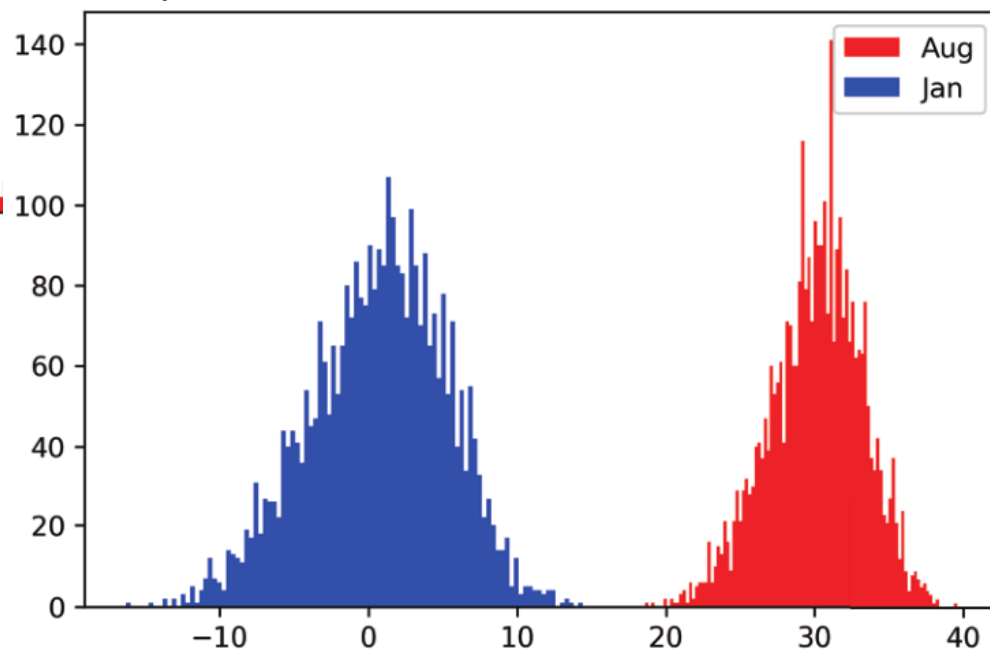
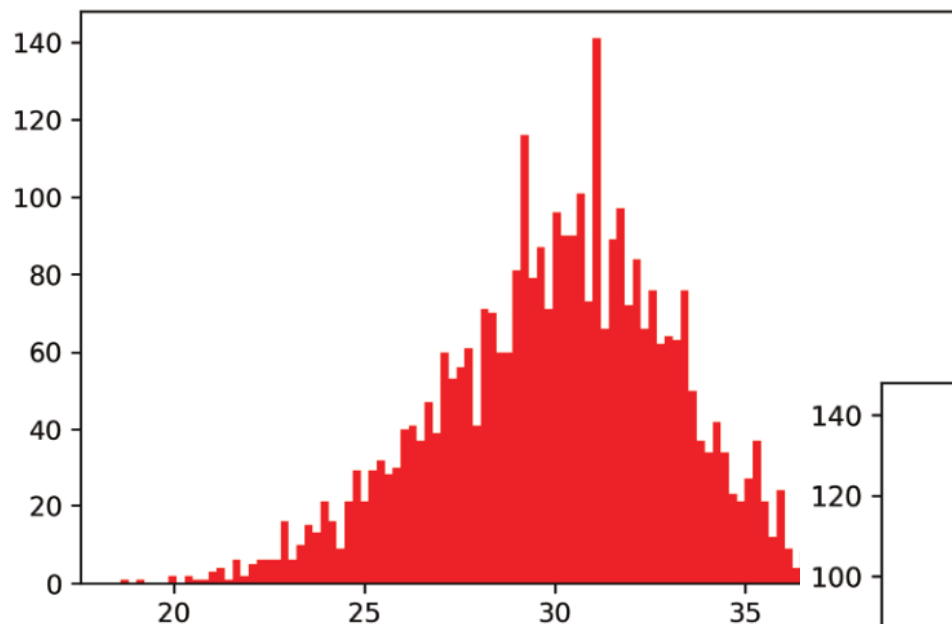
- `Comma - Separated Values`의 약자
- 각각의 데이터 값을 콤마(,)로 구분하는 파일 형식



지점번호	지점명	일시	평균기온(°C)	최고기온(°C)	최고기온시간	최저기온(°C)	최저기온시간
108	서울	2002-06-01	20.2	24.8	12:29	17.1	0:06
108	서울	2002-06-02	22.9	30	15:08	15.3	4:50
108	서울	2002-06-03	23.4	29.1	14:45	17.6	5:31
108	서울	2002-06-04	22.4	27.6	14:06	18.2	4:29
108	서울	2002-06-05	25.2	31.4	15:13	19.1	4:43
108	서울	2002-06-06	25.9	31.2	13:07	21.2	5:57
108	서울	2002-06-07	25.2	30.6	13:23	19.6	23:56
108	서울	2002-06-08	22.1	27.1	17:03	19.4	0:15
108	서울	2002-06-09	22.8	28	13:12	18.3	5:12
108	서울	2002-06-10	22.4	26.6	13:23	20	23:00
108	서울	2002-06-11	21.9	26.8	15:14	19	4:20
108	서울	2002-06-12	18.6	22.9	10:47	16.6	4:08
108	서울	2002-06-13	19.7	24.4	14:35	16	5:09
108	서울	2002-06-14	20.4	25	13:00	16.4	5:18

공공 데이터를 이용하여 기온 분석하기

- 8월의 최고 기온 데이터 히스토그램으로 표현하기



■ seoul.csv 파일 읽어오기

```
import csv
f = open('seoul.csv', 'r', encoding='cp949')
data = csv.reader(f, delimiter=',')
print(data)
f.close()
```

공공 데이터를 이용하여 기온 분석하기

■ seoul.csv 데이터 한 행(row)씩 읽어오기

```
import csv
f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
for row in data :
    print(row)
f.close()
```

→ 4칸 들여쓰기에 주의하세요!

실행
결과

```
['날짜', '지점', '평균기온(℃)', '최저기온(℃)', '최고기온(℃)']
['1907-10-01', '108', '13.5', '7.9', '20.7']
['1907-10-02', '108', '16.2', '7.9', '22']
['1907-10-03', '108', '16.2', '13.1', '21.3']
['1907-10-04', '108', '16.5', '11.2', '22']
(생략)
```

공공 데이터를 이용하여 기온 분석하기

■ next() 함수를 활용해 헤더 저장하기

```
import csv
f = open('seoul.csv')
data = csv.reader(f)
header = next(data)    # ❶
print(header)          # ❷
f.close()
```

실행
결과

```
['날짜', '지점', '평균기온(℃)', '최저기온(℃)', '최고기온(℃)']
```


공공 데이터를 이용하여 기온 분석하기

■ 헤더를 제외한 데이터 한 행씩 출력하기

```
import csv
f = open('seoul.csv')
data = csv.reader(f)
header = next(data)
for row in data :
    print(row)
f.close()
```

실행
결과

```
['1907-10-01', '108', '13.5', '7.9', '20.7']
['1907-10-02', '108', '16.2', '7.9', '22']
(생략)
```

공공 데이터를 이용하여 기온 분석하기

■ 서울이 가장 더웠던 날은 언제였을까? 얼마나 더웠을까?

- 질문 : 기상 관측 이래, 서울의 최고 기온이 가장 높았던 날은 언제였고, 몇 도였을까?
- 질문을 해결하는데 필요한 절차(알고리즘)는?

- 1 | 데이터를 읽어온다.
- 2 | 순차적으로 최고 기온을 확인한다.
- 3 | 최고 기온이 가장 높았던 날짜의 데이터를 저장한다.
- 4 | 최종 저장된 데이터를 출력한다.

공공 데이터를 이용하여 기온 분석하기

- 데이터 중 최고 기온을 실수로 변환하여 한 행씩 출력하기

```
import csv
f = open('seoul.csv')
data = csv.reader(f)
header = next(data)
for row in data :
    row[-1] = float(row[-1])    # 최고 기온을 실수로 변환
    print(row)
f.close()
```

- 최고 기온과 최고 기온이었던 날짜 찾아보기

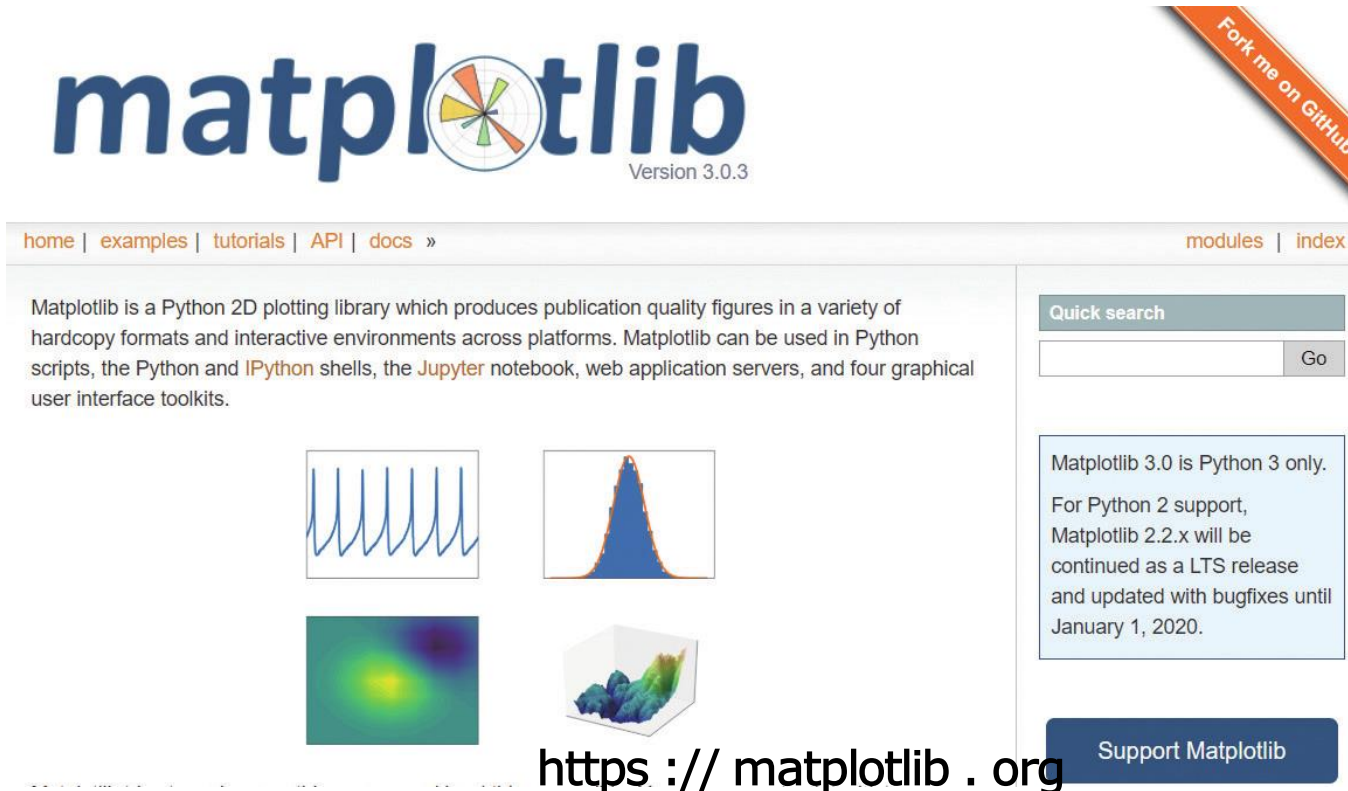
공공 데이터를 이용하여 기온 분석하기

```
import csv                                # CSV 모듈 불러오기
f = open('seoul.csv')                    # seoul.csv 파일 읽기 모드로 불러오기
data = csv.reader(f)
header = next(data)                      # 맨 윗줄을 header 변수에 저장하기
max_temp = -999                          # 최고 기온을 저장할 변수 초기화
max_date = ''                            # 최고 기온이었던 날짜를 저장할 변수 초기화
for row in data :
    if row[-1] == '' :                    # 만약 데이터가 누락되었다면 최고 기온을 -999로 저장
        row[-1] = -999
    row[-1] = float(row[-1])              # 문자열로 저장된 최고 기온 값을 실수로 변환
    if max_temp < row[-1] :                # 만약 지금까지 최고 기온보다 더 높다면 업데이트
        max_date = row[0]
        max_temp = row[-1]
f.close()                                # 파일 닫기
print('기상 관측 이래 서울의 최고 기온이 가장 높았던 날은', max_date +
      '로,', max_temp, '도 었습니다.')    # 출력
```

■ 그래프 그리기

‘matplotlib 라이브러리’란?

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



matplotlib Version 3.0.3

home | examples | tutorials | API | docs » modules | index

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Quick search

Go

Matplotlib 3.0 is Python 3 only.
For Python 2 support, Matplotlib 2.2.x will be continued as a LTS release and updated with bugfixes until January 1, 2020.

Support Matplotlib

<https://matplotlib.org>

■ 그래프 그리기

- matplotlib 라이브러리의 pyplot 모듈 불러오기

```
import matplotlib.pyplot
```

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

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

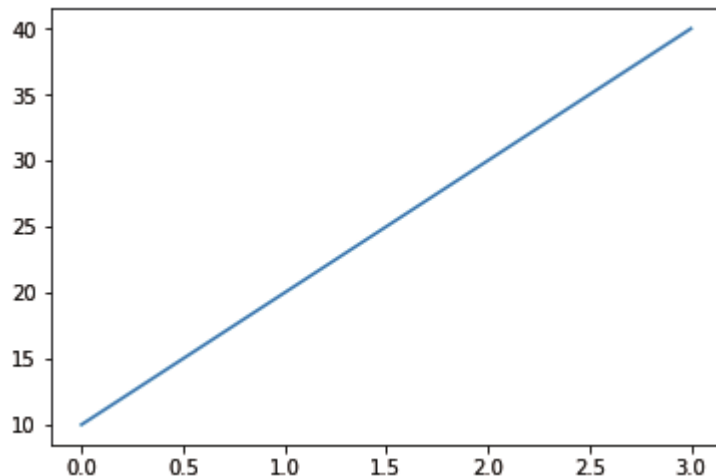
공공 데이터를 이용하여 기온 분석하기

■ 그래프 그리기

- plt.plot(데이터셋)

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

- 실행결과



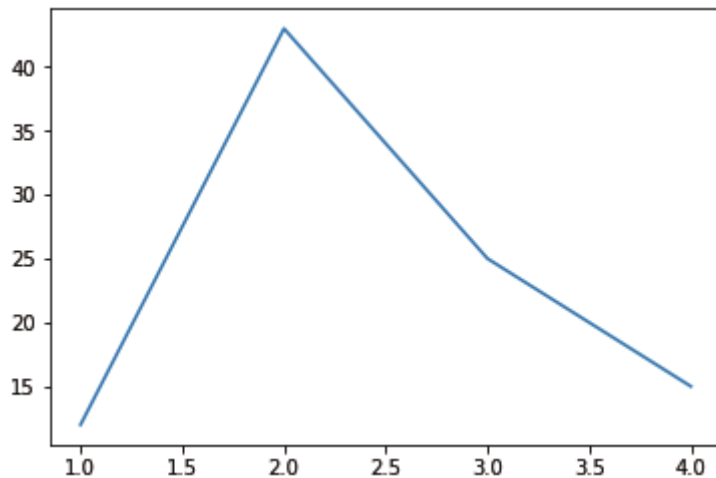
공공 데이터를 이용하여 기온 분석하기

■ 그래프 그리기

- plt.plot(x축 데이터셋, y축 데이터셋)

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

- 실행결과



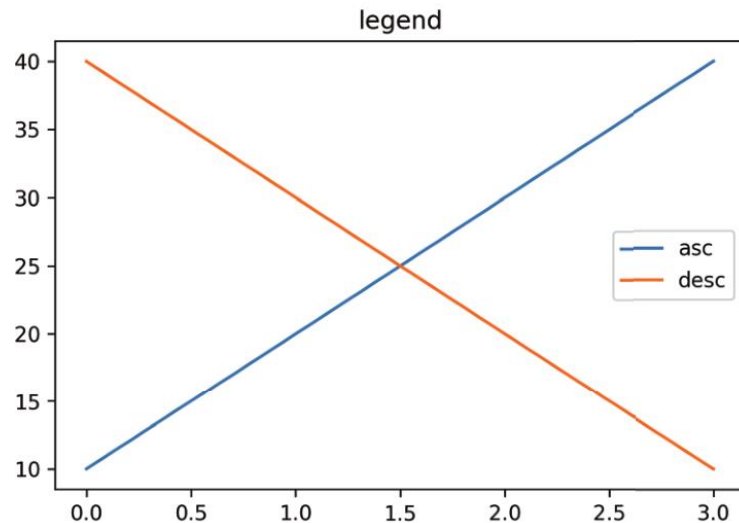
공공 데이터를 이용하여 기온 분석하기

■ 그래프 그리기

- 그래프에 제목, 범례 넣기

```
import matplotlib.pyplot as plt
plt.title('legend')
plt.plot([10, 20, 30, 40], label='asc')      # 증가를 의미하는 asc 범례
plt.plot([40, 30, 20, 10], label='desc')    # 감소를 의미하는 desc 범례
plt.legend()
plt.show()
```

- 실행결과



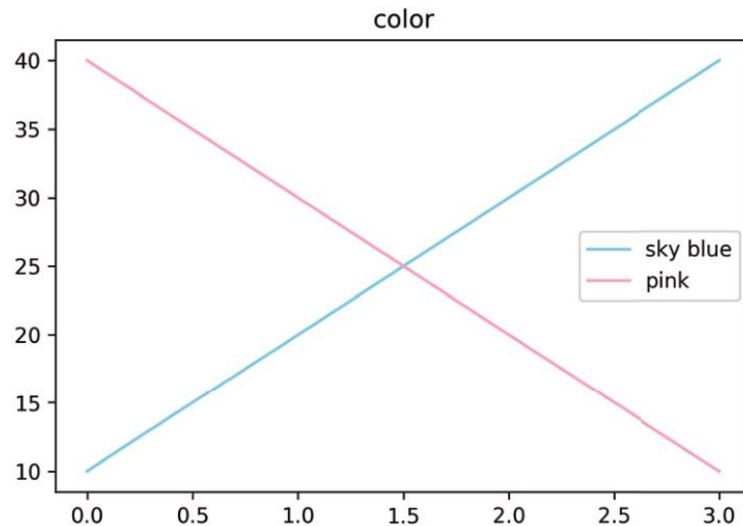
공공 데이터를 이용하여 기온 분석하기

■ 그래프 그리기

- 그래프 색상 바꾸기

```
import matplotlib.pyplot as plt
plt.title('color') # 제목 설정
# 그래프 그리기
plt.plot([10, 20, 30, 40], color='skyblue', label='skyblue')
plt.plot([40, 30, 20, 10], 'pink', label='pink')
plt.legend() # 범례 표시
plt.show()
```

- 실행결과



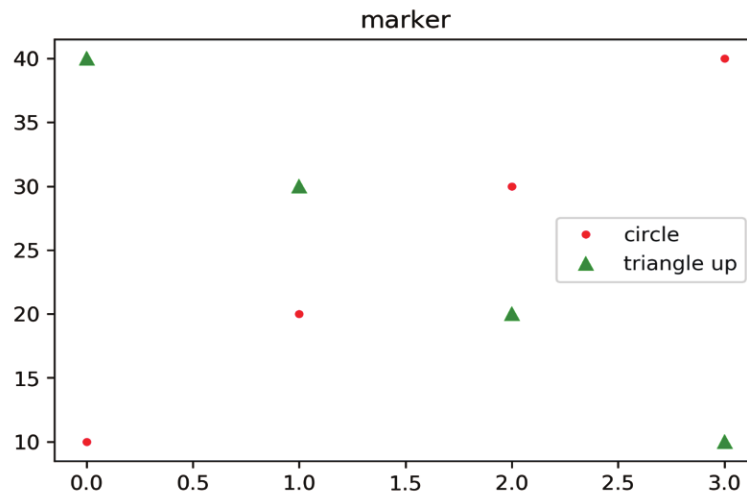
공공 데이터를 이용하여 기온 분석하기

■ 그래프 그리기

- 그래프 선 모양 바꾸기

```
import matplotlib.pyplot as plt
plt.title('marker') # 제목 설정
plt.plot([10, 20, 30, 40], 'r.', label='circle') # 빨간색 원형 마커 그래프
# 초록색 삼각형 마커 그래프
plt.plot([40, 30, 20, 10], 'g^', label='triangle up')
plt.legend() # 범례 표시
plt.show()
```

- 실행결과



공공 데이터를 이용하여 기온 분석하기

- 문자열을 분리하는 `split()` 함수

```
date = '1907-10-01'  
print(date.split('-'))
```

- 실행결과

```
['1907', '10', '01']
```

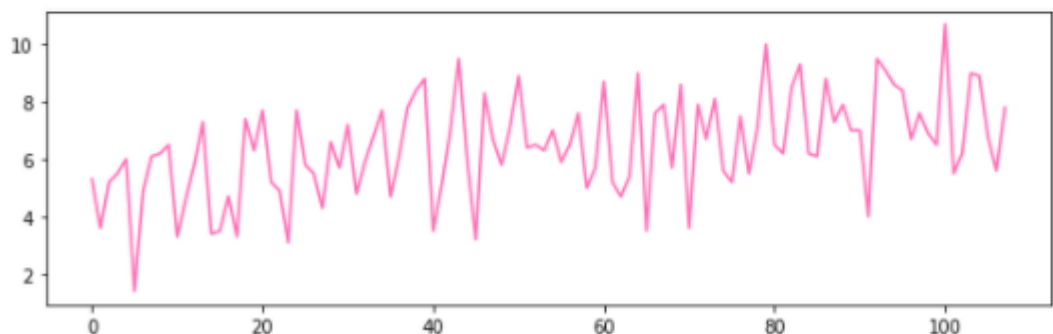
공공 데이터를 이용하여 기온 분석하기

- 11월의 평균기온 데이터 시각화하기(실습)

```
import matplotlib.pyplot as plt
import csv

f = open('data/seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)
next(data)
temp=[]
for i in data:
    if i[1]!='' and i[2]!='':
        if i[1].split('-')[1]=='11':
            temp.append(float(i[2]))

plt.figure(figsize=(10,3))
plt.plot(temp, 'hotpink')
plt.show()
```



공공 데이터를 이용하여 기온 분석하기

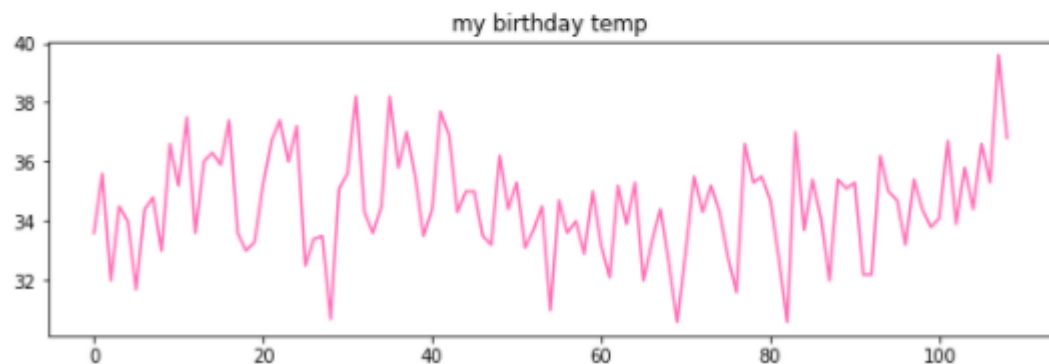
■ 8월1일의 최고 기온변화를 그래프로 나타내보기

```
import matplotlib.pyplot as plt
import csv

f = open('data/seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)
next(data)
max_temp=[]

for i in data:
    if i[4]!='':
        if i[1].split('-')[1] == '08' and i[1].split('-')[2] == '01':
            max_temp.append(float(i[4]))

plt.figure(figsize=(10,3))
plt.plot(max_temp, 'hotpink')
plt.title('my birthday temp')
plt.show()
```



공공 데이터를 이용하여 기온 분석하기

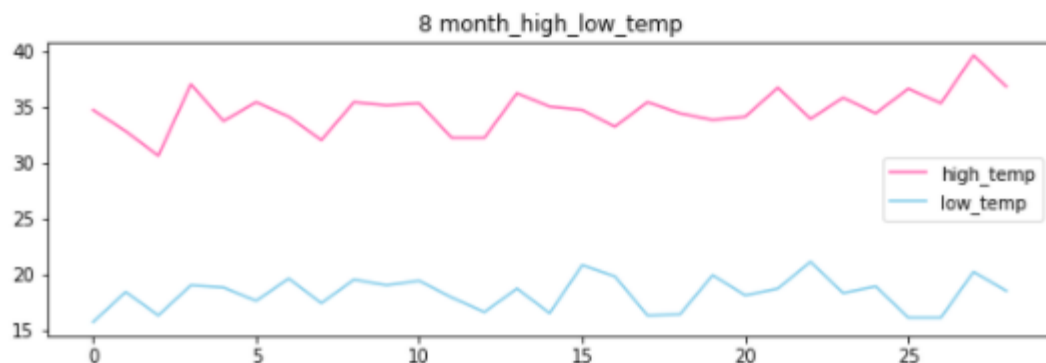
■ 1990이후의 최고 기온과 최저기온의 변화를 그래프로 나타내보기

```
import matplotlib.pyplot as plt
import csv

f = open('data/seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)
next(data)
high_temp=[]
low_temp=[]

for i in data:
    if i[4]!='' and i[7]!='':
        date = i[1].split('-')
        if 1990 < int(date[0]):
            if date[1]=='08' and date[2]=='01':
                high_temp.append(float(i[4]))
                low_temp.append(float(i[7]))

plt.rc('font', family='MalgunGothic')
plt.rcParams['axes.unicode_minus']=False
plt.figure(figsize=(10,3))
plt.plot(high_temp, 'hotpink', label='high_temp')
plt.plot(low_temp, 'skyblue', label='low_temp')
plt.title('8 month_high_low_temp')
plt.legend(loc=5)
plt.show()
```



공공 데이터를 이용하여 기온 분석하기

■ 한글 폰트 사용팁

- 한글 폰트 사용하기(Windows 기준)

```
plt.rc('font', family='Malgun Gothic')  
plt.title('내 생일의 기온 변화 그래프')
```

- 한글 폰트 사용시 마이너스 부호 표현하기

```
plt.rcParams['axes.unicode_minus'] = False
```

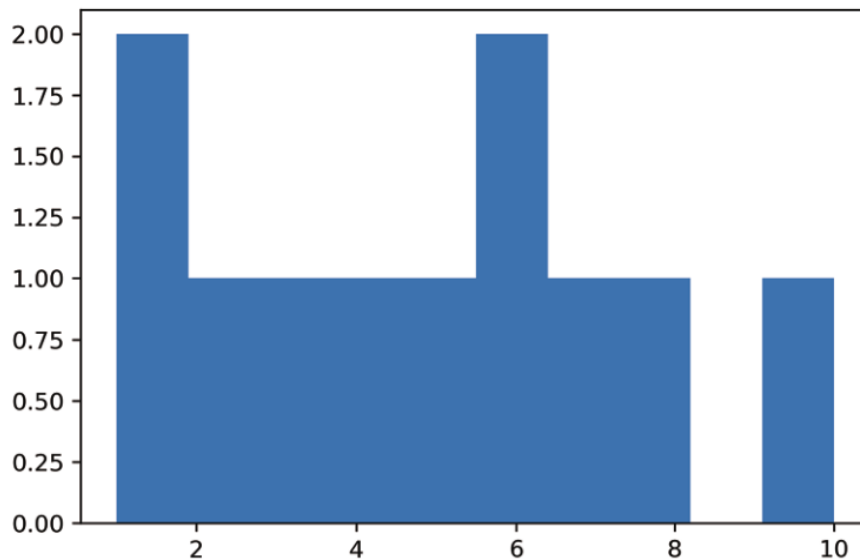

다양한 그래프 알아보기

■ 히스토그램

- hist() 함수 - 막대그래프로 표현

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

- 실행결과



다양한 그래프 알아보기

- 주사위 시뮬레이션의 절차

- 1 주사위를 굴린다.

- 2 나온 결과를 기록한다.

- 3 1~2 과정을 n번 반복한다.

- 4 주사위의 눈이 나온 횟수를 히스토그램으로 그린다.

- 주사위 시뮬레이션(5번 시행)

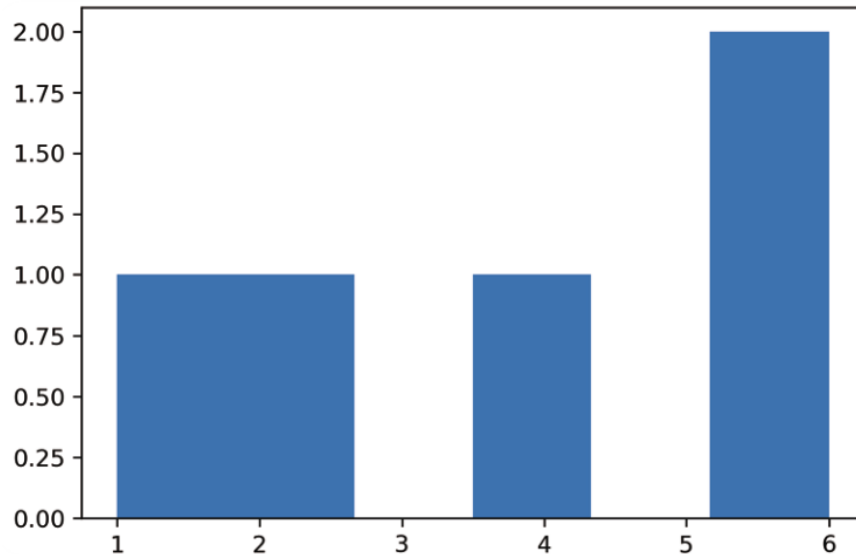
```
import random
dice = []
for i in range(5) :
    dice.append(random.randint(1,6))
print(dice)
```

다양한 그래프 알아보기

- 주사위 시뮬레이션 시각화(6개 구간의 히스토그램으로 표현)

```
import matplotlib.pyplot as plt  
(생략)  
plt.hist(dice, bins=6)  
plt.show()
```

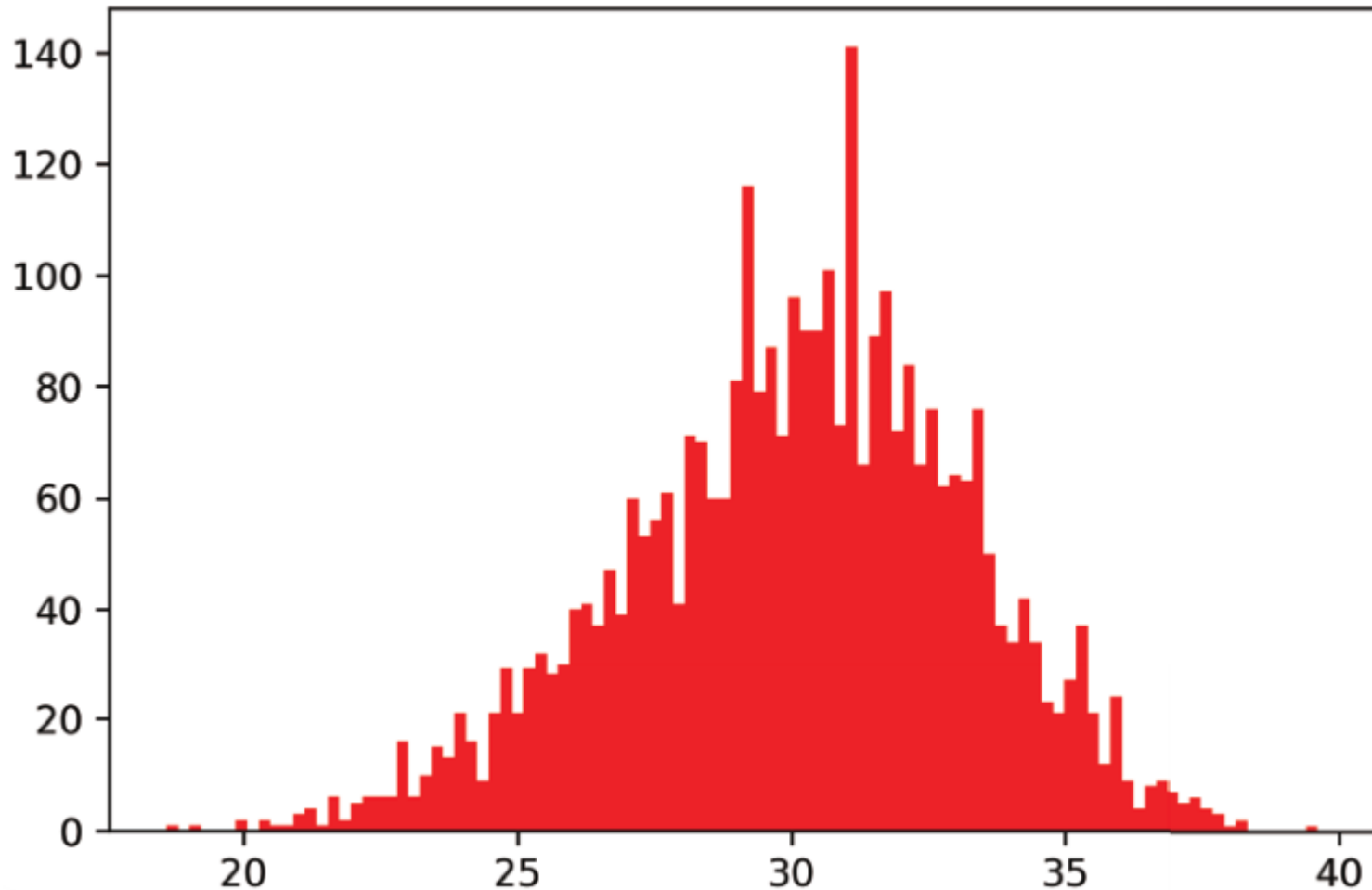
- 실행결과



- 100번 반복
- 1,000,000번 반복으로 변화 시켜 보자(균등한 비율로 나올것이다).

다양한 그래프 알아보기

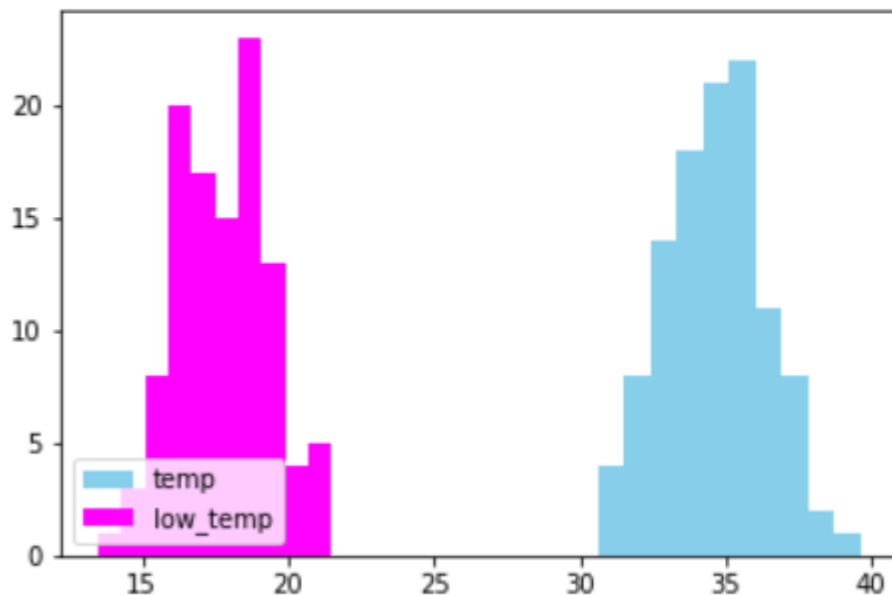
- 8월의 최고 기온 데이터 히스토그램으로 표현하기(스스로 해보기)



다양한 그래프 알아보기

- 8월의 최고 기온 최저기온 데이터 히스토그램으로 표현하기

```
plt.hist(high_temp, color='skyblue', label = 'temp')  
plt.hist(low_temp, color= 'magenta', label = 'low_temp')  
plt.legend(loc=3)  
plt.figure(figsize=(20,15))  
plt.show()
```

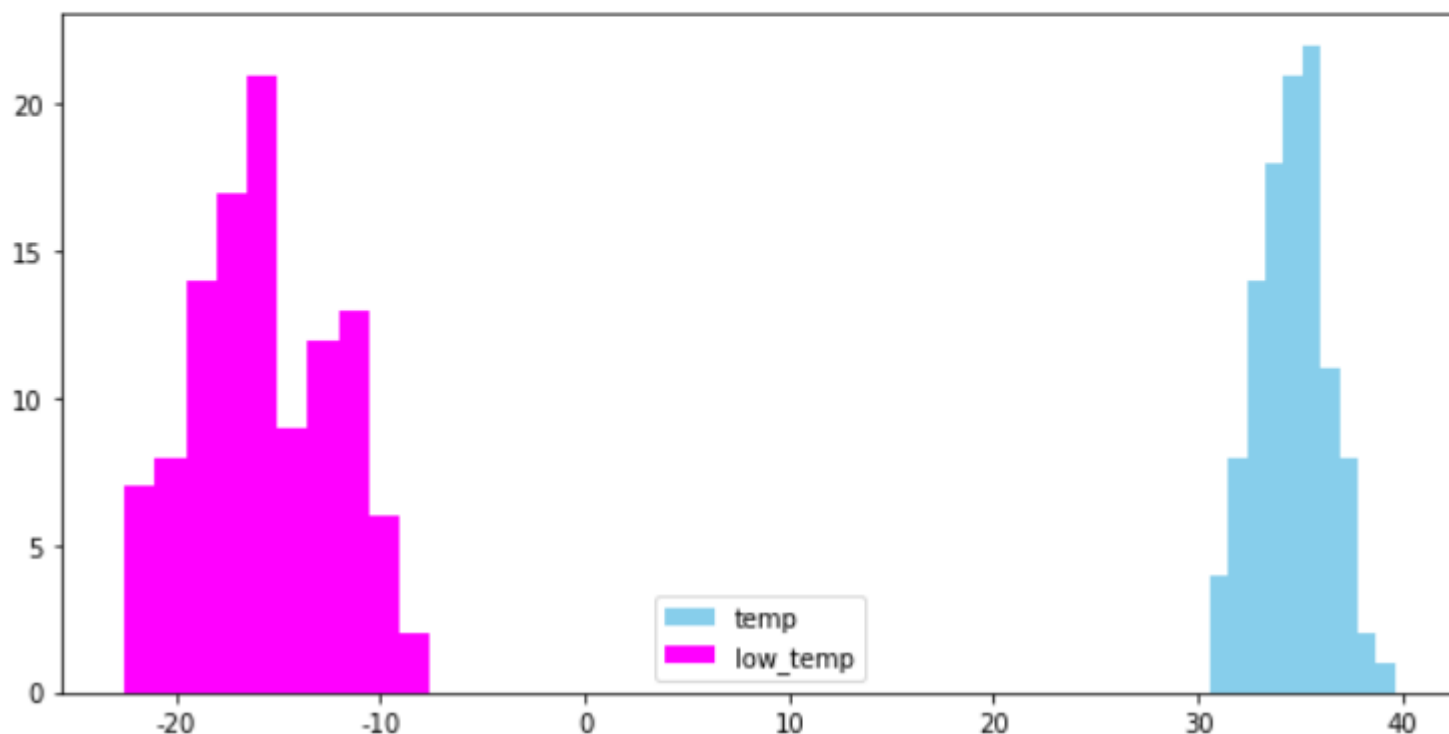


<matplotlib.figure.Figure at 0x7f273691ea58>

다양한 그래프 알아보기

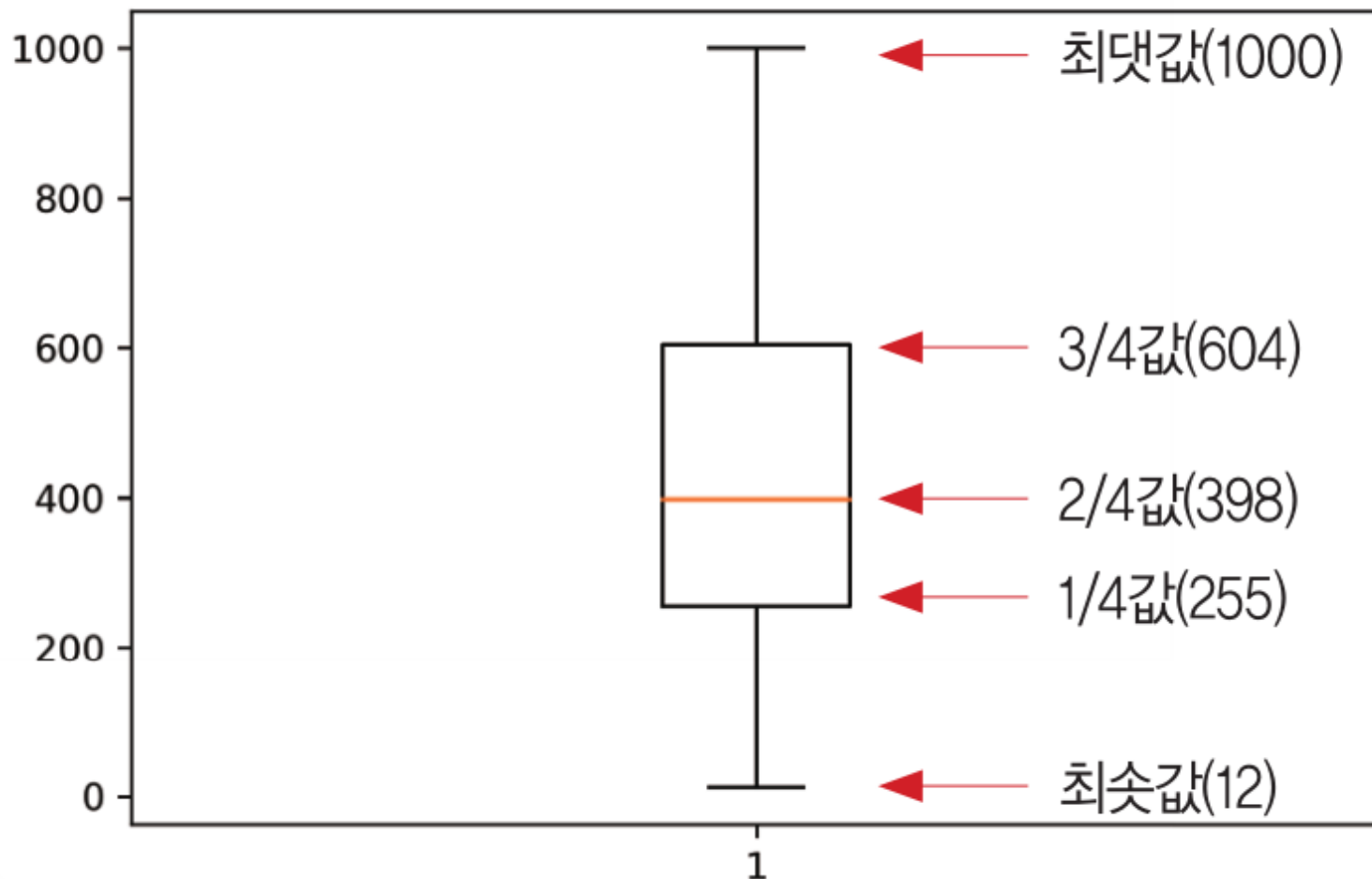
- 1월은 최저기온 8월은 최고기온을 히스토그램으로 표현하기

```
plt.figure(figsize=(10,5))  
plt.hist(high_temp, color='skyblue', label = 'temp')  
plt.hist(low_temp, color= 'magenta', label = 'low_temp')  
plt.legend(loc=8)  
  
plt.show()
```



■ BOXPLOT 그래프(상자그래프)

- 최댓값, 최솟값, 3/4, 2/4, 1/4 값을 표현할 수 있는 상자 그림(boxplot)



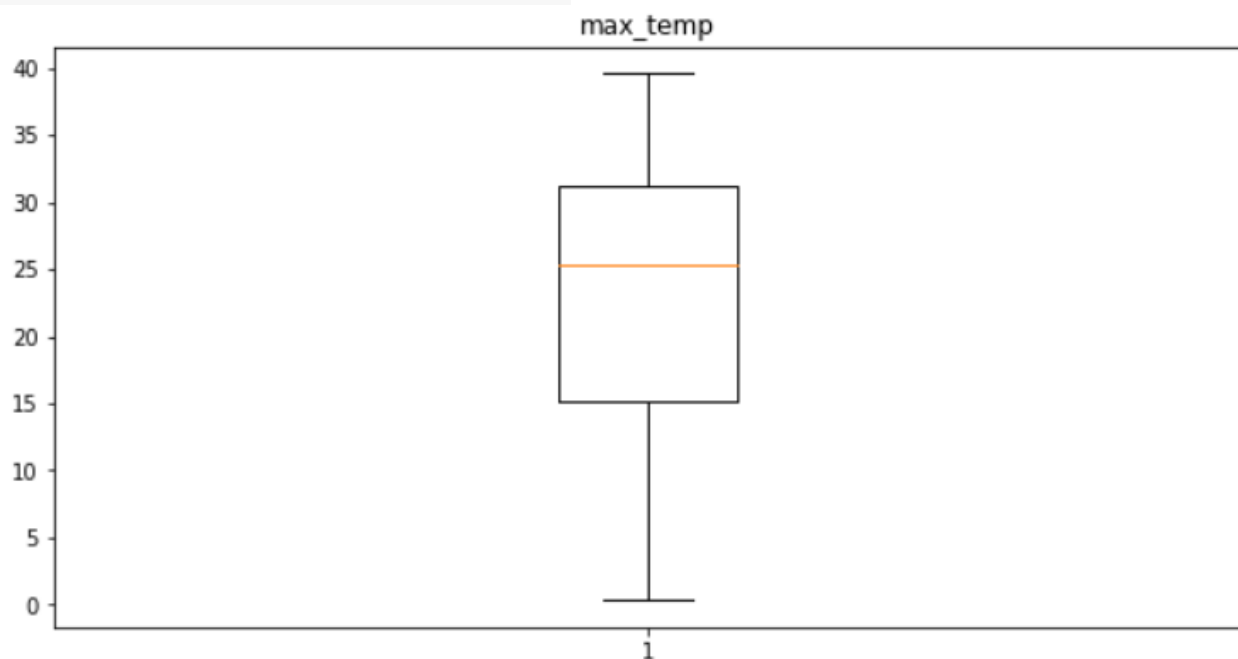
다양한 그래프 알아보기

■ 최고 기온 데이터 상자 그림으로 표현하기

```
f = open('data/seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)
next(data)
temp=[]

for i in data:
    if i[4]!='' :
        temp.append(float(i[4]))

plt.figure(figsize=(10,5))
plt.boxplot(temp)
plt.title('max_temp')
plt.show()
```



다양한 그래프 알아보기

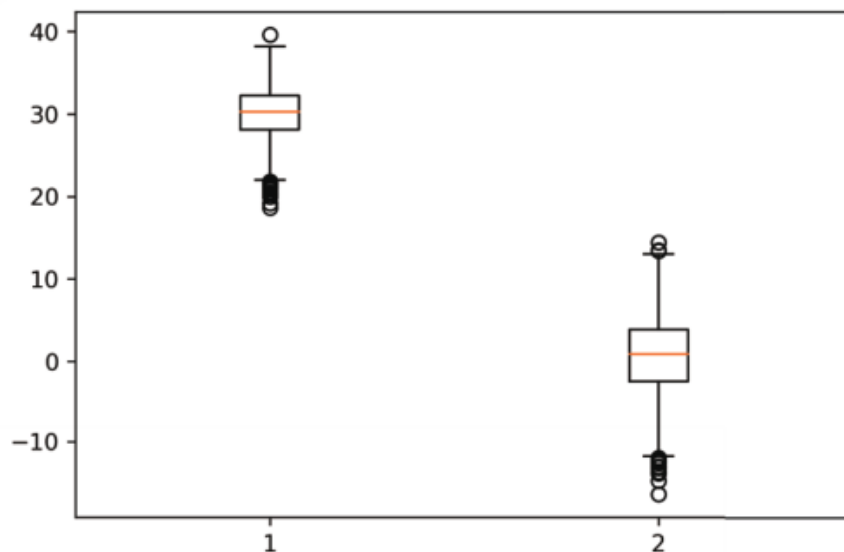
- 1월과 8월의 최고 기온 데이터 상자 그림으로 표현하기

```
f = open('data/seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)
next(data)
jan_temp=[]
aug_temp=[]
```

```
for i in data:
    if i[4]!='':
        date = i[1].split('-')
        if date[1]=='01':
            jan_temp.append(float(i[4]))
        if date[1] == '08':
            aug_temp.append(float(i[4]))
```

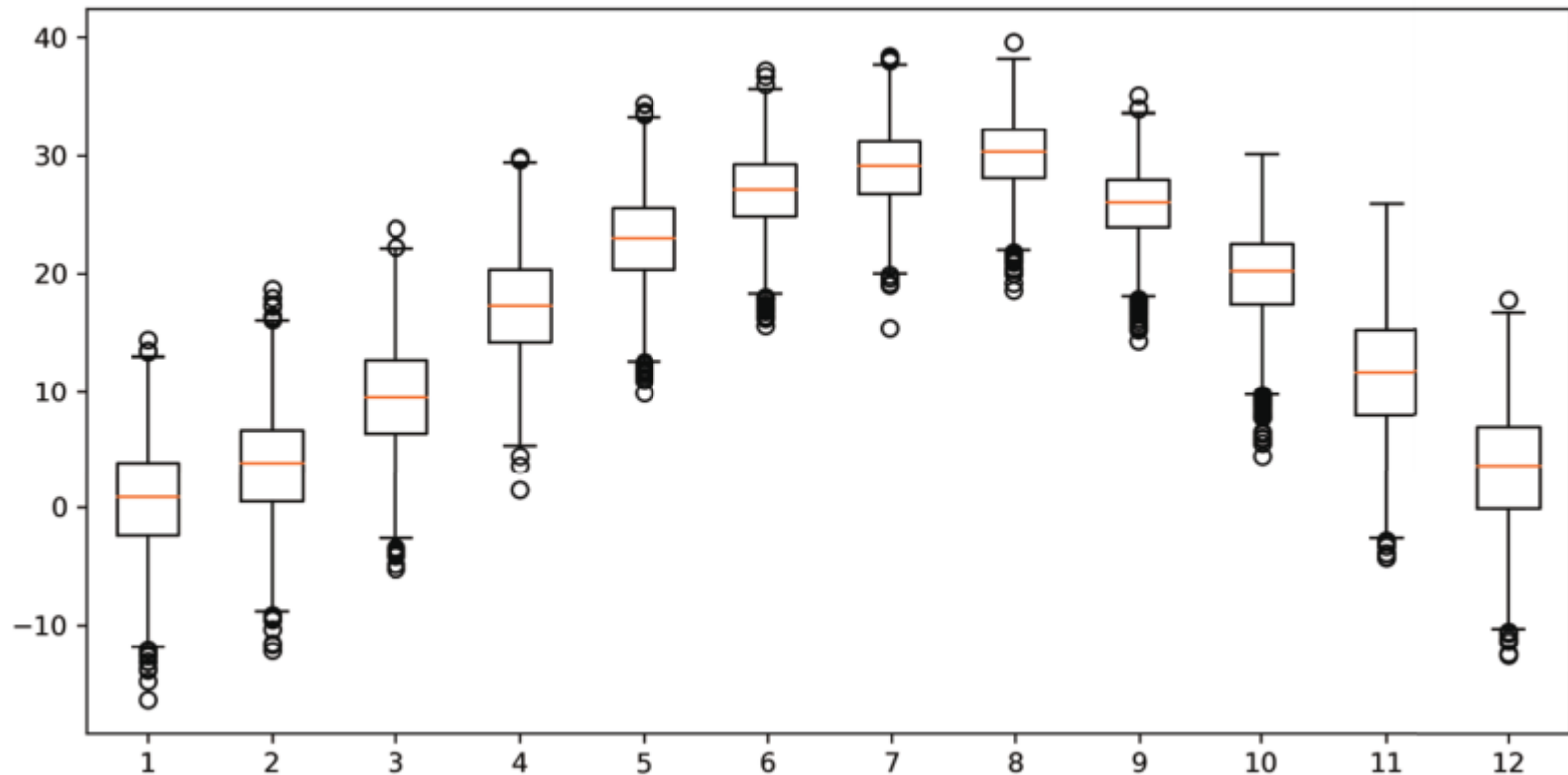
```
plt.figure(figsize=(10,5))
#plt.boxplot([jan_temp, aug_temp])
plt.boxplot(jan_temp)
plt.boxplot(aug_temp)
plt.title('jan_aug max_temp')
plt.show()
```

plt.boxplot([aug, jan])



다양한 그래프 알아보기

- 1월부터 12월까지 최고 기온 데이터 상자 그림으로 표현하기



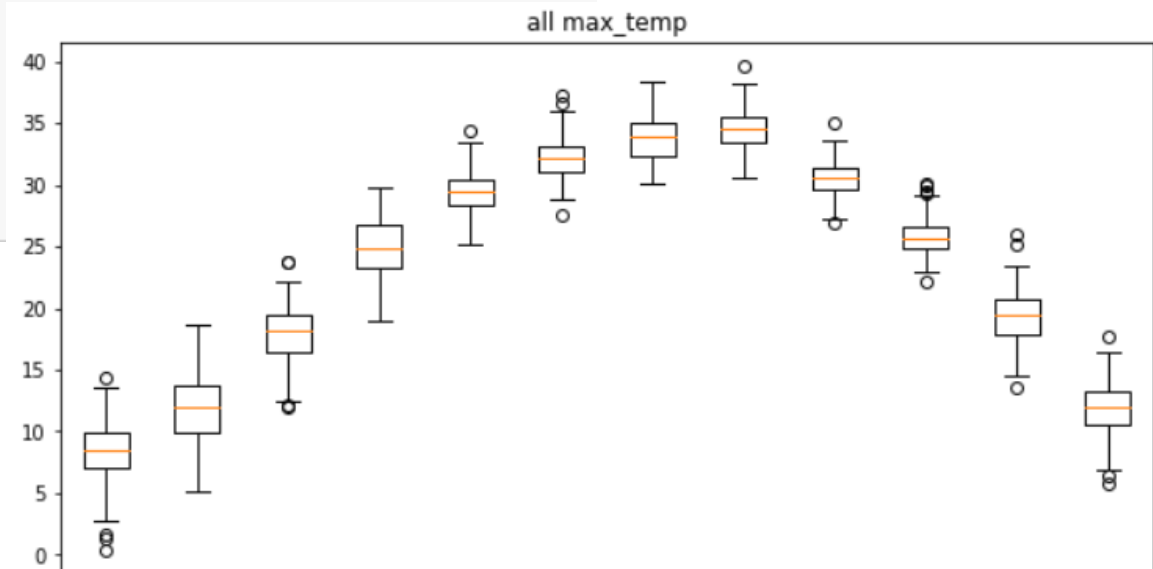
다양한 그래프 알아보기

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

```
f = open('data/seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)
next(data)
month=[]
for j in range(12):
    month.append([])

for i in data:
    if i[4]!='':
        date = i[1].split('-')
        month[int(date[1])-1].append(float(i[4]))
```

```
plt.figure(figsize=(10,5))
plt.boxplot(month)
plt.title('all max_temp')
plt.show()
```



다양한 그래프 알아보기

- 8월 1일부터 31일까지 최고 기온 데이터 상자 그림으로 표현하기

