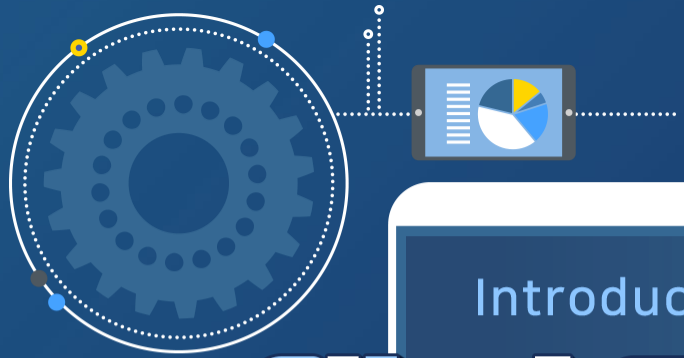
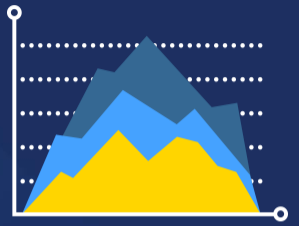


Introduction To Data Analysis

데이터분석입문

Lecture 14. 서울의 최저 최고 기온 분석

인공지능소프트웨어학과 강환수 교수



학습개요

- 막대 그래프로 기온 데이터 시각화
- 박스 그래프로 기온 데이터 시각화
  - 박스 그래프에서 박스와 수염의 의미

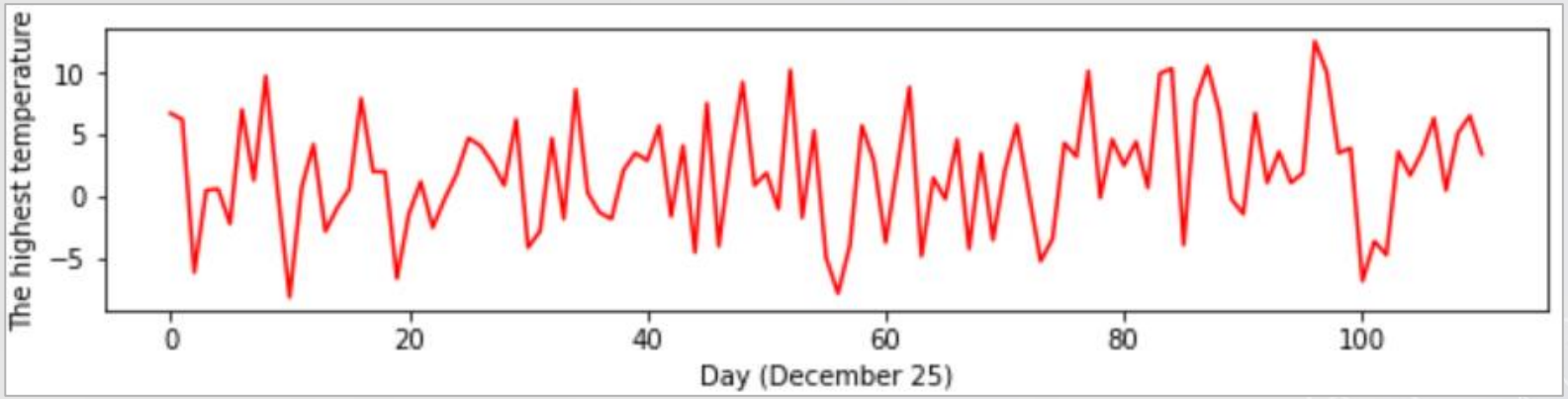
학습목표

- 기온 데이터 시각화를 위해 막대 그래프를 그릴 수 있다.
- 기온 데이터 시각화를 위해 박스 그래프를 그릴 수 있다.
- 컴프리헨션의 장점을 이해하고 활용할 수 있다.
- 박스 그래프에서 박스의 길이와 수염의 의미를 이해 할 수 있다.

LESSON 01

막대 그래프로 기온 데이터 시각화

매년 크리스마스의 최고 기온 데이터를 추출하여 그린 결과

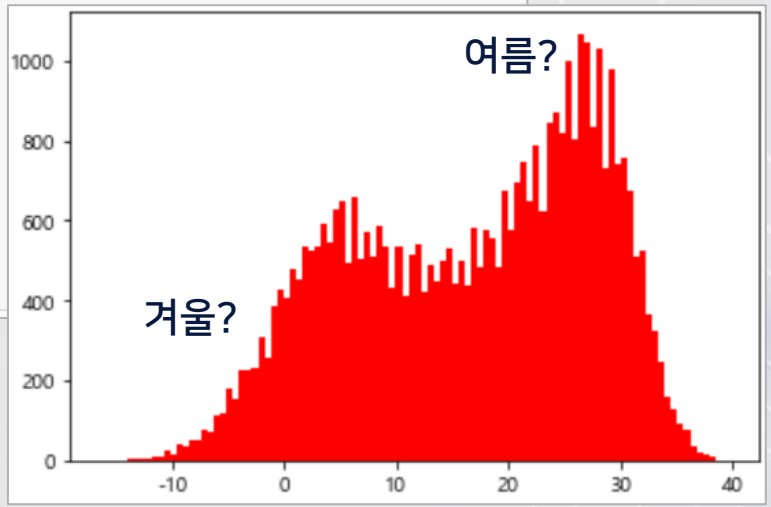


이 그래프만 봐서는 특별한 정보를 얻을 수가 없습니다.
꺾은선 그래프가 아닌 다른 형태로 시각화 하면 어떨까요?

기온 데이터를 히스토그램으로 시각화

최고 기온을 hist()로 그리기

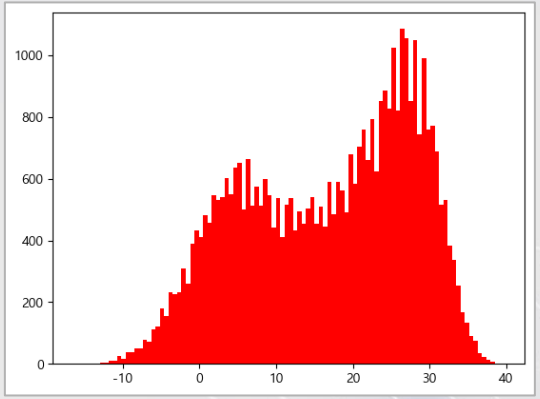
```
import csv
import matplotlib.pyplot as plt

f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
result = []

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

plt.hist(result, bins=100, color='r')
plt.show()
```



구간 내용 확인

반환 값, 2개로 확인

```
a, b, c = plt.hist(result, bins = 100, color = 'r')
print(f'구간 빈도수 앞 5개: {a[:5]}')
print(f'구간 빈도수 뒤 5개: {a[-5:]}')
print(f'구간 경계수 앞 5개: {b[:5]}')
print(f'구간 경계수 뒤 5개: {b[-5:]}')
print()
print(f'하나의 구간의 크기(직접계산): {(b[-1]-b[0])/100 :.3f}')
print(f'하나의 구간의 크기(자동계산): {b[1]-b[0] :.3f}')
```




```
구간 빈도수 앞 5개: [1. 0. 1. 0. 2.]
구간 빈도수 뒤 5개: [21. 14.  7.  0.  1.]
구간 경계수 앞 5개: [-16.3   -15.741 -15.182 -14.623 -14.064]
구간 경계수 뒤 5개: [37.364 37.923 38.482 39.041 39.6  ]

하나의 구간의 크기(직접계산): 0.559
하나의 구간의 크기(자동계산): 0.559
```

리스트 컴프리헨션

규칙적인 리스트를 쉽게 생성

```python
# 반복으로 리스트 생성
a = []
for i in range(5):
    a.append(i)
print(a)

# 컴프리헨션으로 리스트 생성
b = [i for i in range(5)]
print(b)
```

✓ 0.0s Python

```
[0, 1, 2, 3, 4]
[0, 1, 2, 3, 4]
```

```python
print([i ** 2 for i in range(5)])
print([pow(i, 3) for i in range(5)])
```

✓ 0.0s Python

```
[0, 1, 4, 9, 16]
[0, 1, 8, 27, 64]
```

다양한 컴프리헨션

리스트, 사전, 집합

```
print([[i+1] for i in range(5)])
print([(i+1, ) for i in range(5)])
print([{i: i**2} for i in range(5)])
print([{i+1} for i in range(5)])
```

✓ 0.0s Python

```
[[1], [2], [3], [4], [5]]
[(1,), (2,), (3,), (4,), (5,)]
[{0: 0}, {1: 1}, {2: 4}, {3: 9}, {4: 16}]
[{1}, {2}, {3}, {4}, {5}]
```

조건에 따른 항목을 구성하는 컴프리헨션

```python
# 반복 전체의 항목을 리스트에 삽입
print([i for i in range(6)])
# 반복 항목 중에서 짝수를 리스트에 삽입
print([i for i in range(6) if i%2 == 0])
# 반복 항목 중에서 홀수를 리스트에 삽입
print([i for i in range(6) if i%2 == 1])
```

✓ 0.0s Python

```
[0, 1, 2, 3, 4, 5]
[0, 2, 4]
[1, 3, 5]
```

조건 연산자와 컴프리헨션

```python
# 조건 연산자
age = 10
adult = '성인' if age >= 18 else '미성년'
print(adult)
```

✓ 0.0s Python

```
미성년
```

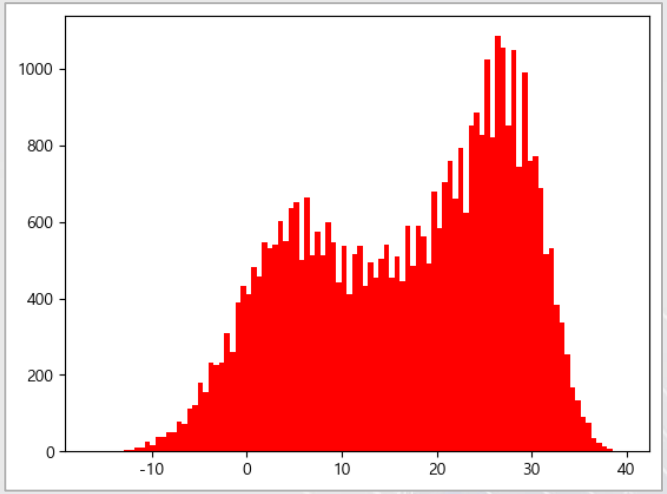
```python
# 리스트에 짝수면 0, 홀수면 1을 삽입
print([0 if i%2 == 0 else 1 for i in range(10)])

# 리스트에 짝수면 even, 홀수면 odd를 삽입
print(['even' if i%2 == 0 else 'odd' for i in range(10)])
```

✓ 0.0s Python

```
[0, 1, 0, 1, 0, 1, 0, 1, 0, 1]
['even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'ever
```

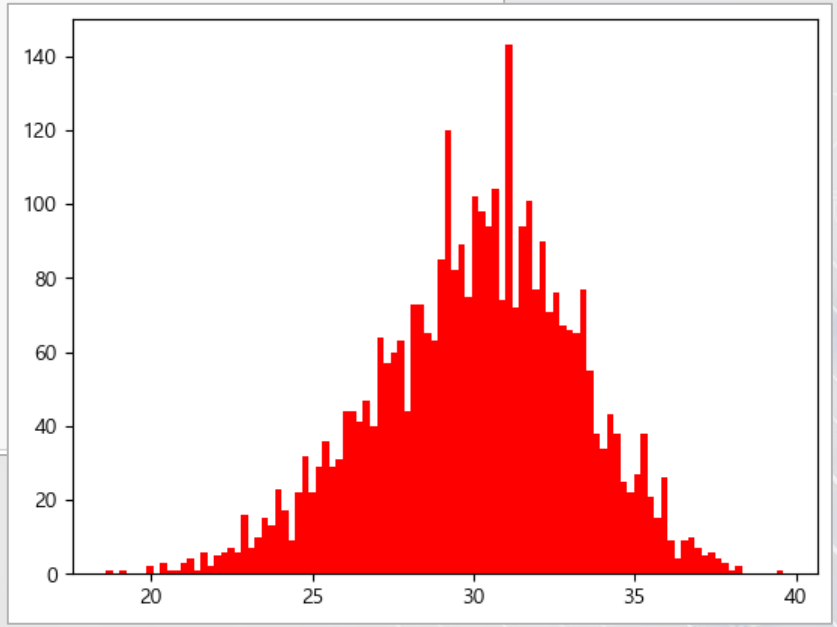
최고 기온 데이터를 컴프리헨션으로 생성해 히스토그램으로 시각화

```
import csv
import matplotlib.pyplot as plt

f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
next(data)

# 컴프리헨션으로 간단히 생성
# result = [float(row[-1]) for row in data if row[-1] != '']
result = [float(row[-1]) for row in data if row[-1]]

plt.hist(result, bins = 100, color = 'r')
plt.show()
```



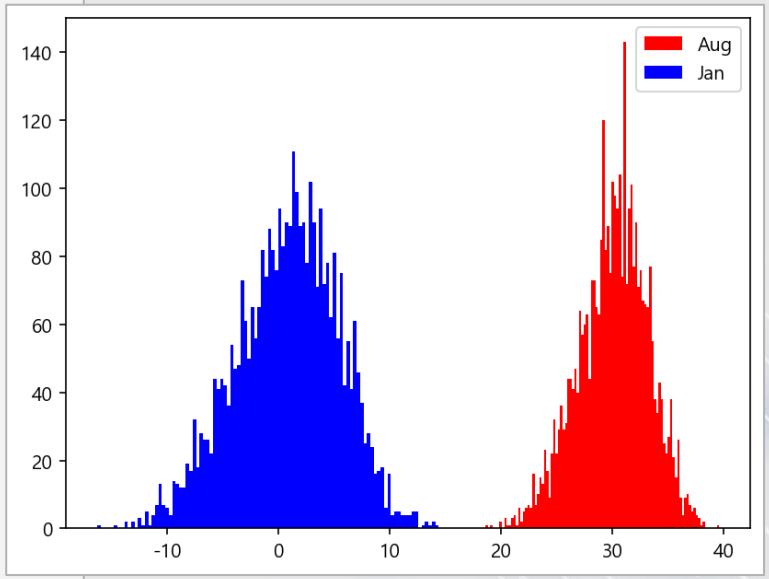
8월의 최고 기온 데이터를 히스토그램으로 시각화

```
import csv
import matplotlib.pyplot as plt

f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
aug = []

for row in data:
    month = row[0].split('-')[1]
    if row[4] != '':
        if month == '08':
            aug.append(float(row[4]))

plt.hist(aug, bins=100, color='r')
plt.show()
```



1월과 8월의 최고 기온 데이터를 히스토그램으로 시각화

```
import csv
import matplotlib.pyplot as plt

f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
next(data)
aug = []
jan = []

for row in data :
    month = row[0].split('-')[1]
    if row[-1] != '' :
        if month == '01':
            jan.append(float(row[-1]))
        if month == '08':
            aug.append(float(row[-1]))

plt.figure(dpi=150) # 해상도를 높이려면 기본 값인 100 이상으로
plt.hist(aug, bins = 100, color = 'r', label = 'Aug')
plt.hist(jan, bins = 100, color = 'b', label = 'Jan')
plt.legend()
plt.show()
```

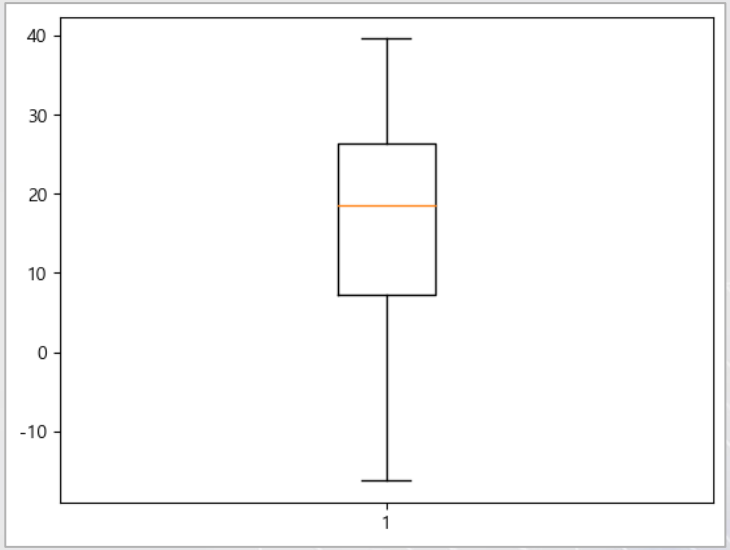


LESSON 02

박스 그래프로 기온 데이터 시각화

최고 기온 데이터를 상자 그림으로 시각화

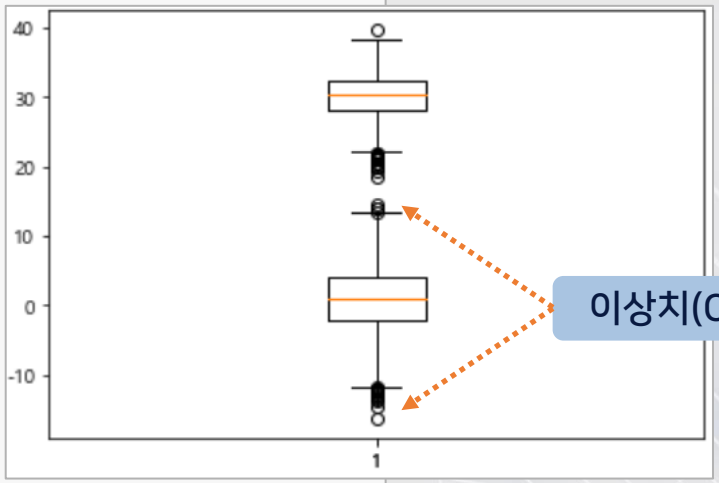
```
import csv
import matplotlib.pyplot as plt
f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
next(data)

result = [float(row[-1]) for row in data if row[-1]]
plt.boxplot(result)
plt.show()
```



1월과 8월의 최고 기온 데이터를 상자 그림으로 시각화

```
import csv
import matplotlib.pyplot as plt

f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
jan = []
aug = []

for row in data:
    month = row[0].split('-')[1]
    if row[4] != '':
        if month == '01':
            jan.append(float(row[4]))
        elif month == '08':
            aug.append(float(row[4]))

plt.boxplot(jan)
plt.boxplot(aug)
plt.show()
```

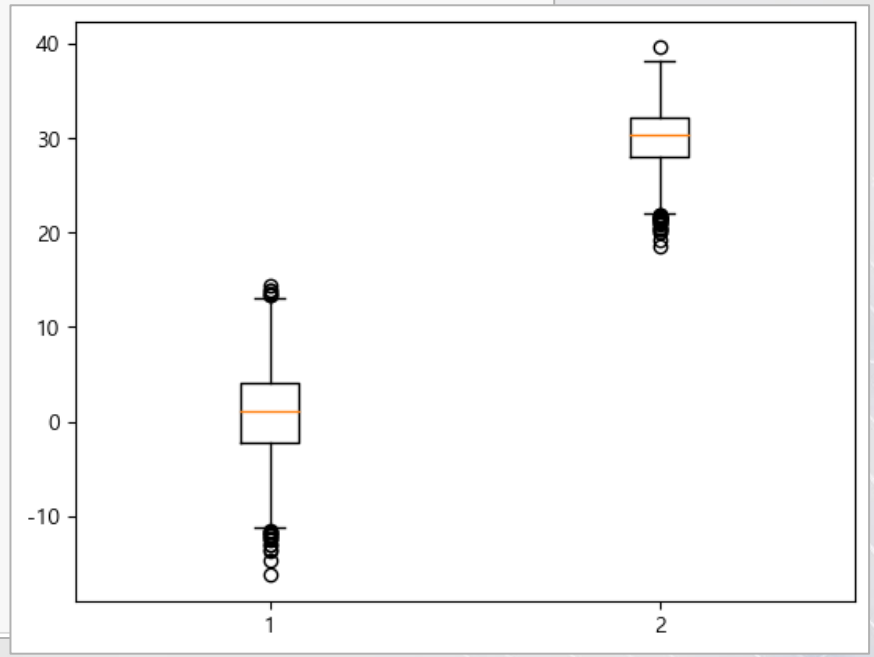
1월과 8월 데이터를
분리하여 표현할 순 없을까요?



1월과 8월의 최고 기온 데이터를 상자 그림으로 시각화

목록으로 순차적으로 그리기

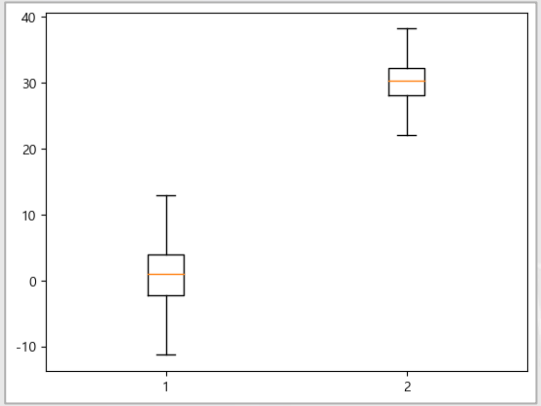
```
import csv
import matplotlib.pyplot as plt

f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
jan = []
aug = []

for row in data:
    month = row[0].split('-')[1]
    if row[4] != '':
        if month == '01':
            jan.append(float(row[4]))
        elif month == '08':
            aug.append(float(row[4]))

plt.boxplot([jan, aug])
plt.show()
```

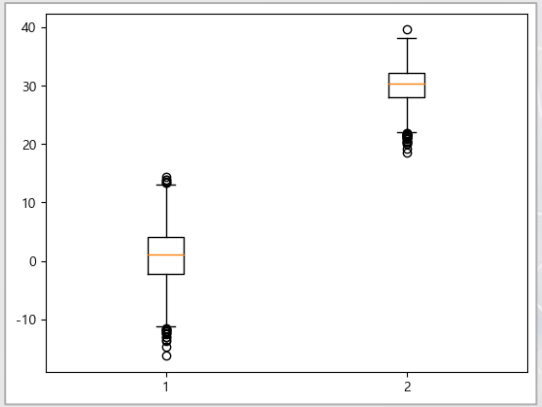


박스의 이상치 그리기

옵션 showfliers=True

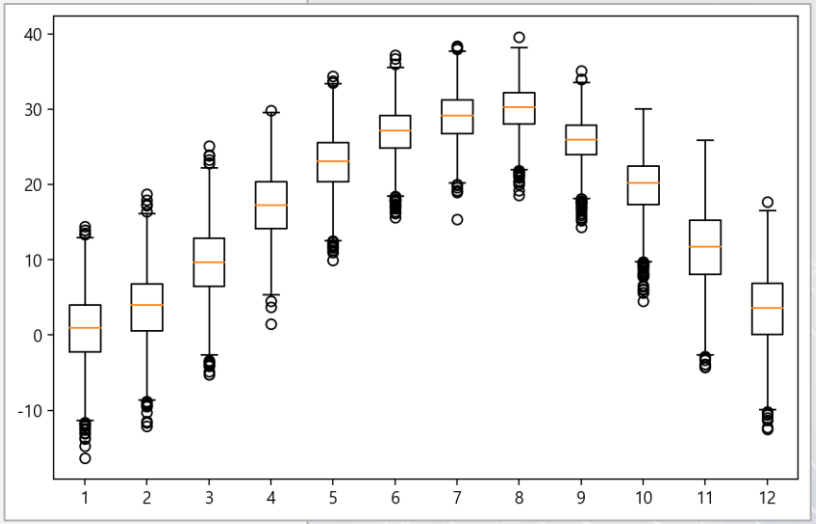
```
import csv
import matplotlib.pyplot as plt

f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
next(data)
aug = []
jan = []

for row in data :
    month = row[0].split('-')[1]
    if row[-1] != '' :
        if month == '08':
            aug.append(float(row[-1]))
        if month == '01':
            jan.append(float(row[-1]))

plt.boxplot([jan, aug], showfliers=False)
plt.show()
```





1월부터 12월까지의 최고 기온 데이터를 상자 그림으로 시각화

```
import csv
import matplotlib.pyplot as plt

f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
next(data)

# 월별 데이터를 저장할 리스트 month
month = [[] for i in range(12)]

for row in data :
    if row[-1] != '' :
        # 1월 => month[0]에 저장
        month[int(row[0].split('-')[1])-1].append(float(row[-1]))

plt.figure(figsize=(8,5), dpi=200)
plt.boxplot(month)
plt.show()
```

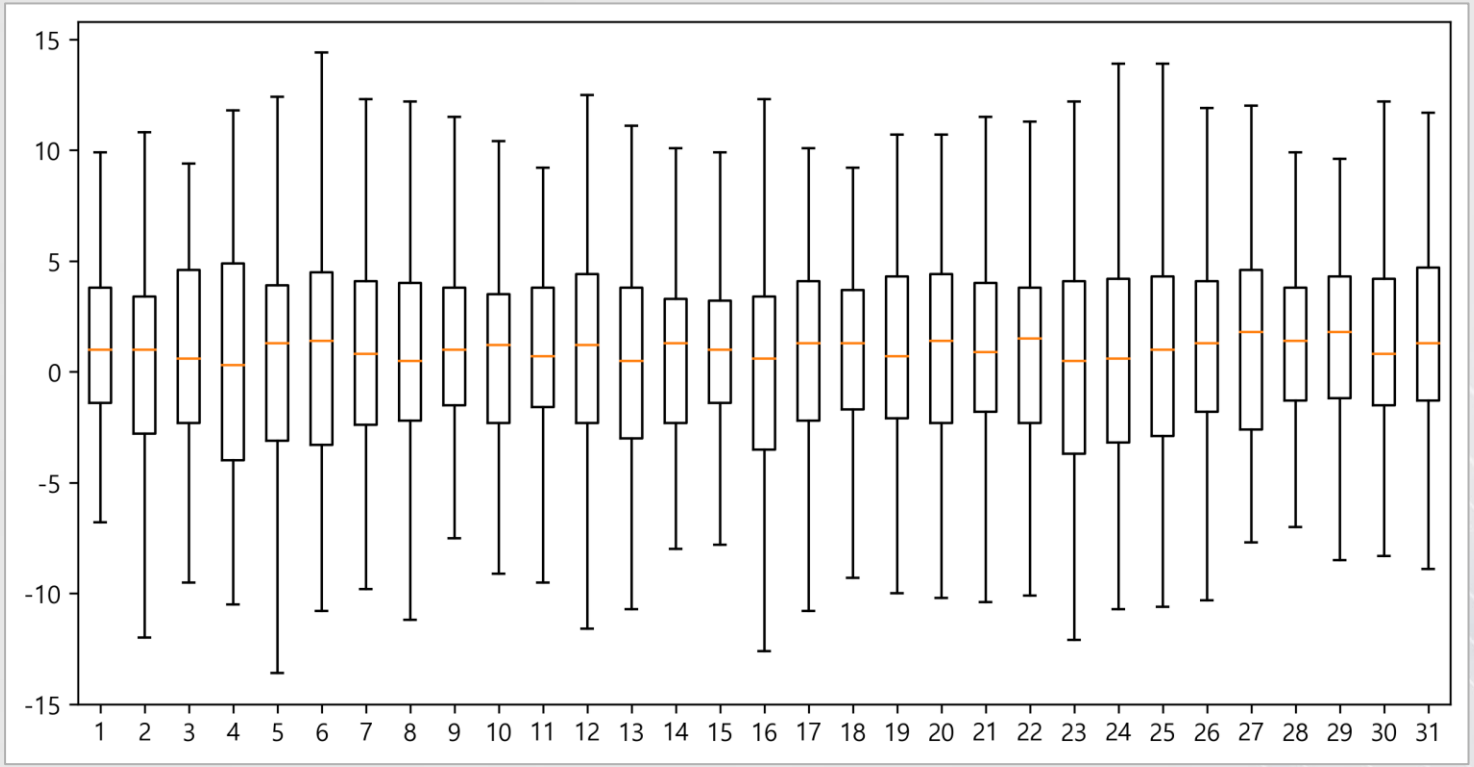


1월 일별 기온 데이터를 상자 그림으로 시각화

```
import csv
import matplotlib.pyplot as plt
f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
next(data)

day = [[] for i in range(31)]
for row in data :
    if row[-1] != '' :
        if row[0].split('-')[1] == '01':
            day[int(row[0].split('-')[2])-1].append(float(row[-1]))

plt.figure(figsize=(10, 5), dpi=300)
plt.boxplot(day, showfliers=False)
plt.show()
```

1월 일별 기온 데이터를 상자 그림으로 시각화

결과



boxplot() 내부 자료 의미

- **중앙값(median): 자료 수가 홀수면 정렬한 자료 의 가운데(중간) 값, 짝수면 가운데 2개의 평균 값**
- **Q1(제 1사분위수, 25th 백분위수)**
  - 데이터의 하위 순서로 25%에 해당하는 값(중간과 최하위 수의 중간 값)
- **Q3(제 3사분위수, 75th 백분위수)**
  - 데이터의 하위 순서로 75%에 해당하는 값(중간과 최상위 수의 중간 값)

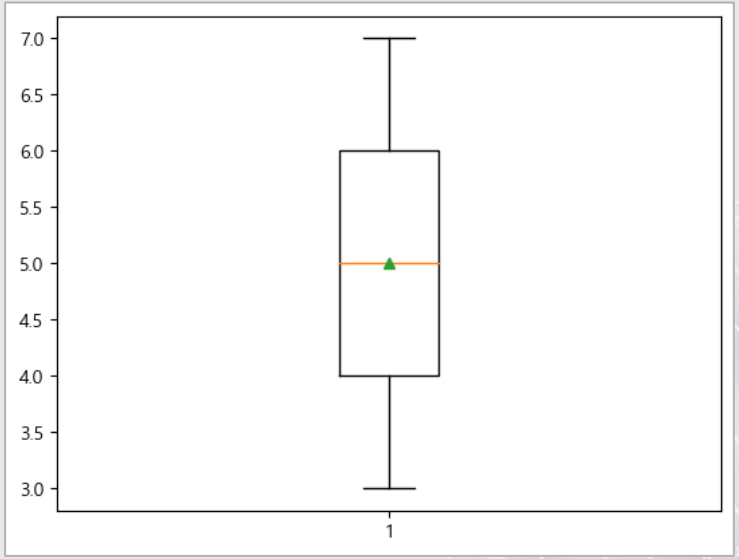
boxplot() 내부 자료 의미

IQR(Interquartile Range)는 Q3에서 Q1을 뺀 값으로, 데이터의 중간 50%를 나타냄, 박스 자체의 길이

```
import numpy as np
import matplotlib.pyplot as plt

# 가상의 데이터 생성
data = [3, 4, 5, 6, 7]
plt.boxplot(data, showmeans=True)
print(f'중앙값(median): {np.median(data)}')
print(f'평균값(mean): {np.mean(data)}')
plt.show()
```

✓ 0.1s

```
중앙값(median): 5.0
평균값(mean): 5.0
```



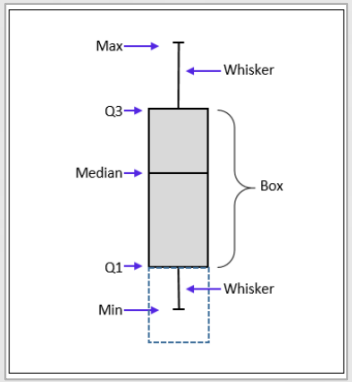
boxplot() 수염(whisker)

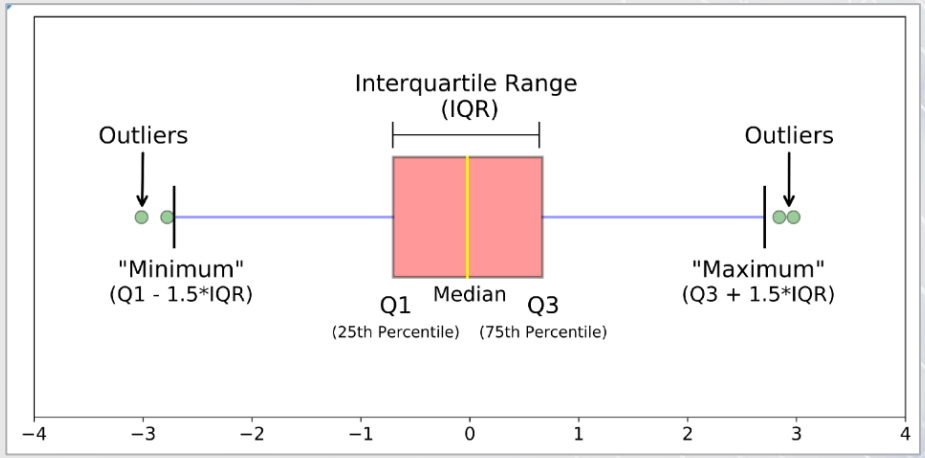
Interquirtile(Q3 – Q1)을 계산

- Q1과 Q3의 바깥쪽(각각 왼쪽, 오른쪽)으로 '1.5 (Q3 - Q1) 크기의 범위 내의 인접값'을 실선으로 연결하여 표시
- 없다면 수염 없음

이상치(outliers): 수염을 벗어나는 수

- Q1 – 1.5*IRQ 내를 벗어나는 수
- Q3 + 1.5*IRQ 내를 벗어나는 수

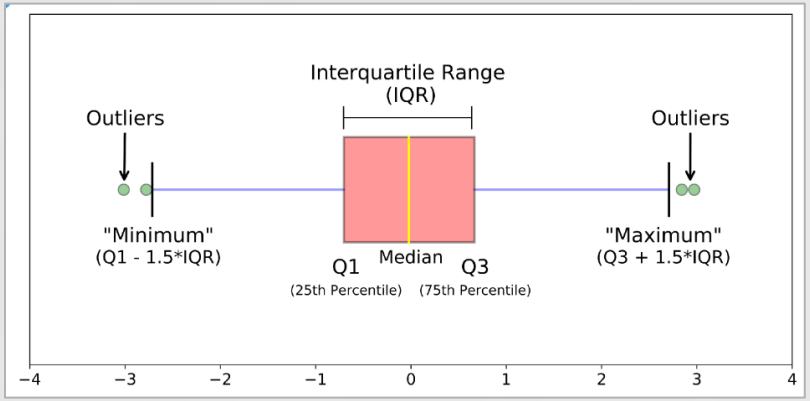




boxplot() 수염(whisker) 사례

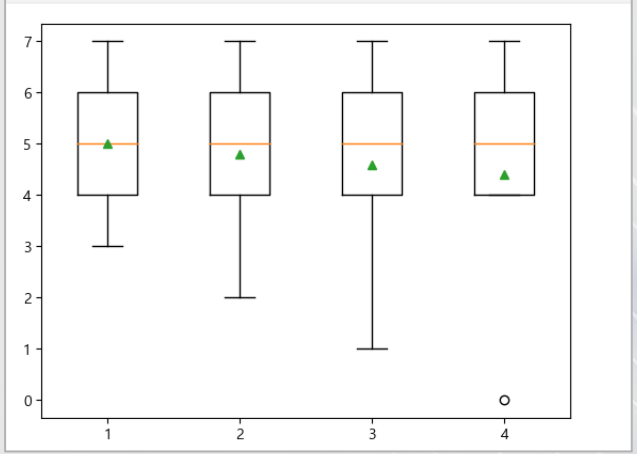
이상치(outliers): 수염을 벗어나는 수

- IRQ*1.5 == 3
- 4 – 3 = 1, 1 미만은 이상치
- 6 + 3 = 9, 9 초과는 이상치




```
import matplotlib.pyplot as plt

# 가상의 데이터 생성
d1 = [3, 4, 5, 6, 7]
d2 = [2, 4, 5, 6, 7]
d3 = [1, 4, 5, 6, 7]
d4 = [0, 4, 5, 6, 7]
plt.boxplot([d1, d2, d3, d4], showmeans=True)
plt.show()
```

0.1s



SUMMARY

학습정리

돗수분포표 hist()

- bins=100: 온도를 100개의 구간으로 나누어 빈도 수를 그리기
- a, b, c = plt.hist(result, bins=100)
  - a: 구간 빈도 수, b: 구간 경계 수

컴프리헨션

- 리스트, 사전, 집합을 간단히 만드는 방법

```
print([[i+1] for i in range(5)])
print([(i+1, ) for i in range(5)])
print([{i: i**2} for i in range(5)])
print([{i+1} for i in range(5)])
```

0.0s Python

```
[[1], [2], [3], [4], [5]]
[(1,), (2,), (3,), (4,), (5,)]
[{0: 0}, {1: 1}, {2: 4}, {3: 9}, {4: 16}]
[{1}, {2}, {3}, {4}, {5}]
```

박스 boxplot()

- 박스로 간략히 그리는 자료 분포