

강원지역혁신플랫폼

1기 학습

Machine Learning

이상값 처리 개념



▶ 학습목표

📁 이상값 처리 개념을 이해하고
구현할 수 있습니다.



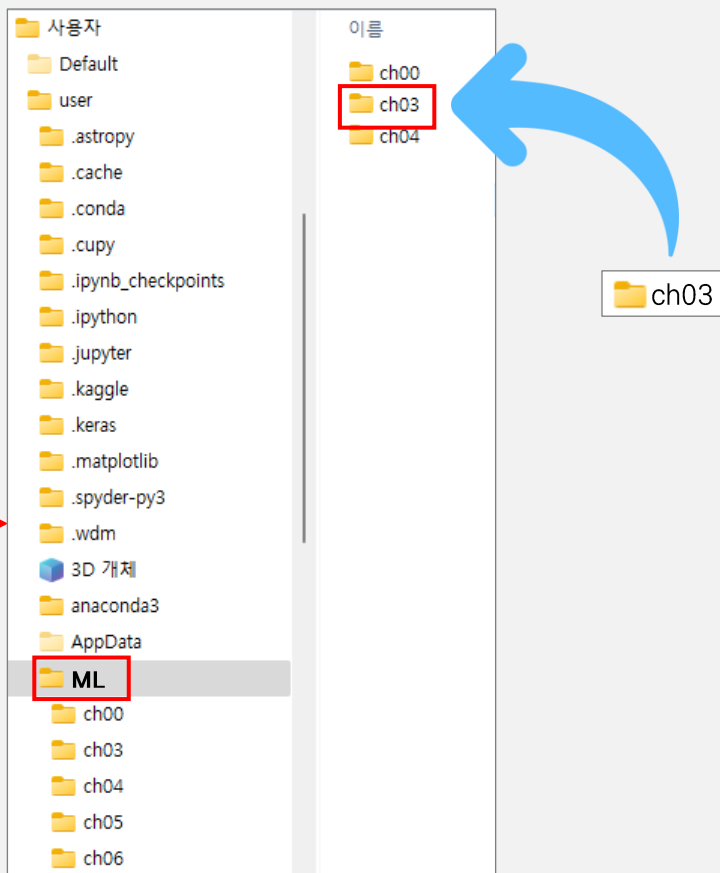


01 | 3주차 실습코드 복사하기

⚠ (권장) 아래와 같은 경로에 실행 소스가 존재하면 환경 구축 완료

◆ 3주차 실습코드 다운로드 → 압축해제 → ch03 폴더를 ML 하위 폴더로 복사

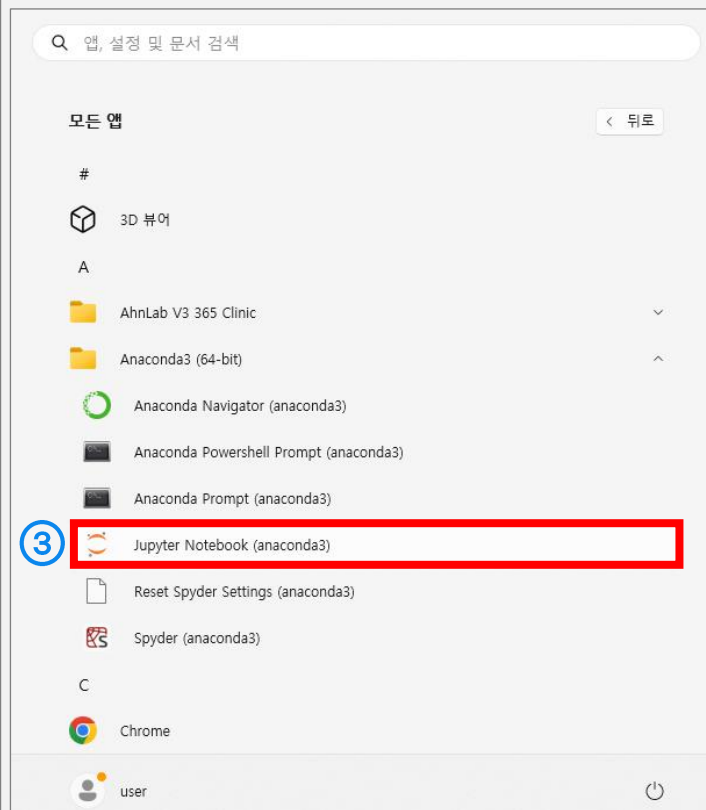
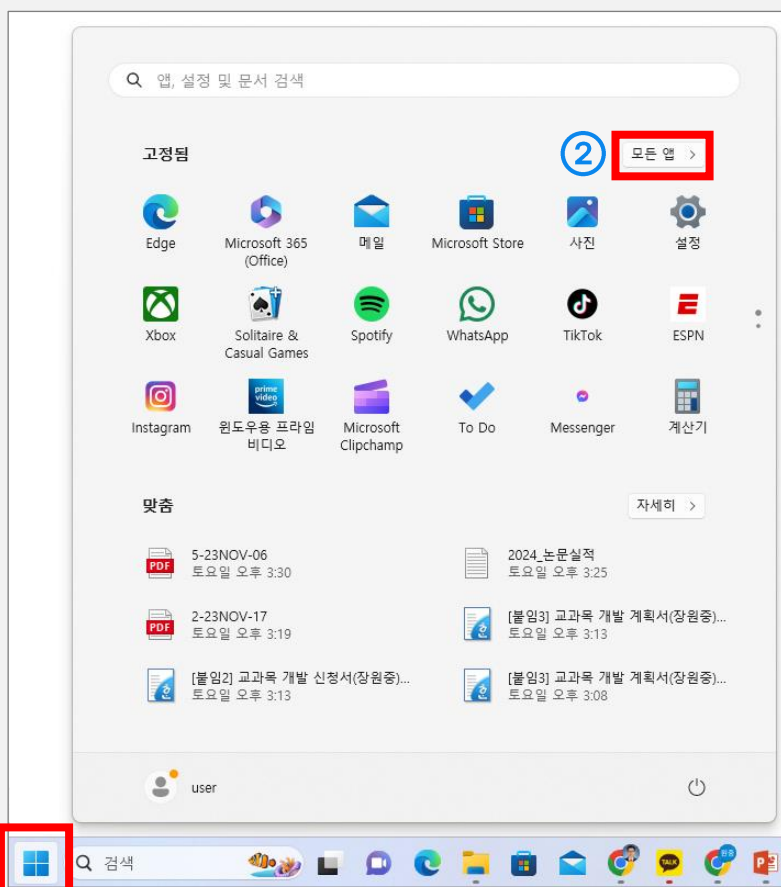
◆ c:\Users\user>ML> 컴퓨터이름 또는 사용자계정





02 | Jupyter Notebook 실행하기

- ◆ ① 시작 메뉴 클릭 > ② 모든 앱 버튼 클릭 > ③ Anaconda3(64-bit)
> “Jupyter Notebook (anaconda)” 메뉴 클릭하기





03 | ML 폴더

◆ ML 폴더를 클릭하기

jupyter

QuitLogout

FilesRunningClusters

Select items to perform actions on them.

UploadNew↺

0 ▾ /

Name ▾Last ModifiedFile size

<input type="checkbox"/>	3D Objects	일 년 전	
<input type="checkbox"/>	anaconda3	7달 전	
<input type="checkbox"/>	Contacts	9달 전	
<input type="checkbox"/>	Desktop	4달 전	
<input type="checkbox"/>	Documents	6분 전	
<input type="checkbox"/>	Downloads	2시간 전	
<input type="checkbox"/>	Favorites	9달 전	
<input type="checkbox"/>	ML	22분 전	
<input type="checkbox"/>	Links	9달 전	
<input type="checkbox"/>	Music	9달 전	
<input type="checkbox"/>	OneDrive	일 년 전	
<input type="checkbox"/>	Pictures	9달 전	
<input type="checkbox"/>	Saved Games	9달 전	
<input type="checkbox"/>	scikit_learn_data	8달 전	
<input type="checkbox"/>	seaborn-data	3달 전	
<input type="checkbox"/>	Searches	3달 전	
<input type="checkbox"/>	Videos	9달 전	
<input type="checkbox"/>	Untitled.ipynb	4달 전	1.64 kB



04 | ch03 폴더

◆ ch03 폴더 클릭하기

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕

<input type="checkbox"/> 0 ▾	📁 /	Name ▾	Last Modified	File size
<input type="checkbox"/>	📁 ch00		9일 전	
<input type="checkbox"/>	📁 ch03		5일 전	
<input type="checkbox"/>	📁 ch04		4일 전	
<input type="checkbox"/>	📁 ch05		2일 전	
<input type="checkbox"/>	📁 ch06		몇 초 전	
<input type="checkbox"/>	📁 ch07		몇 초 전	
			7일 전	
			7일 전	



05 | ch03_02_이상치처리.ipynb

✦ ch03_02_이상치처리.ipynb 파일 클릭하기

Files

Running

Open

Download

Rename

Duplicate




Delete

New

Upload

Refresh

/ ch03 /

Name	Last Modified	File Size
<input type="checkbox"/>  ch03_01_결측치처리.ipynb	12 minutes ago	244.1 KB
<input checked="" type="checkbox"/>  ch03_02_이상치처리.ipynb	12 minutes ago	301.3 KB
<input type="checkbox"/>  std_sample_data_filled.xlsx	12 days ago	18.7 KB
<input type="checkbox"/>  std_sample_data_outliers_filled.xlsx	11 days ago	18 KB
<input type="checkbox"/>  std_sample_data.xlsx	12 days ago	24.9 KB



06 | 이상치란?

△ 이상치 처리

- ◆ 이상치(Outlier)는 보통 관측된 데이터의 범위에서 많이 벗어난 아주 작은 값이나 큰 값을 의미함
 - 어떤 의사결정을 하는데 필요한 데이터 분석 혹은 모델링하는 경우에 이상치가 큰 영향을 미칠 수 있음
 - 데이터 전처리 과정에서 적절한 이상치 처리는 필수적임
- 이상치는 상대적인 개념임
 - ─ 어떤 데이터를 어떻게 분석하고, 어느 기준으로 이상치를 판단할 것이냐에 따라, 이상치 데이터들이 달라짐



07 | 이상치가 생기는 요인

◆ 이상치가 생기는 요인

- 데이터 수집 과정에서 오류가 발생하는 경우
- 데이터 자체가 이상치를 포함하고 있는 경우
- 변경점 발생으로 인한 데이터 분포가 변화하는 경우 등이 존재함



08 | 이상값 판단 방법

- ◆ 이상치 즉, “데이터의 범위에서 많이 벗어난 아주 작은 값이나 큰 값”이라는 것은 정확히 어떤 기준으로 판단할 수 있는지 생각해 보자.
 - 여기서는 아래와 같은 2가지 방법으로 생각해 봄
 - ▬ 표준 편차(Standard deviation)
 - ▬ 박스 플롯의 사분위 범위(IQR)

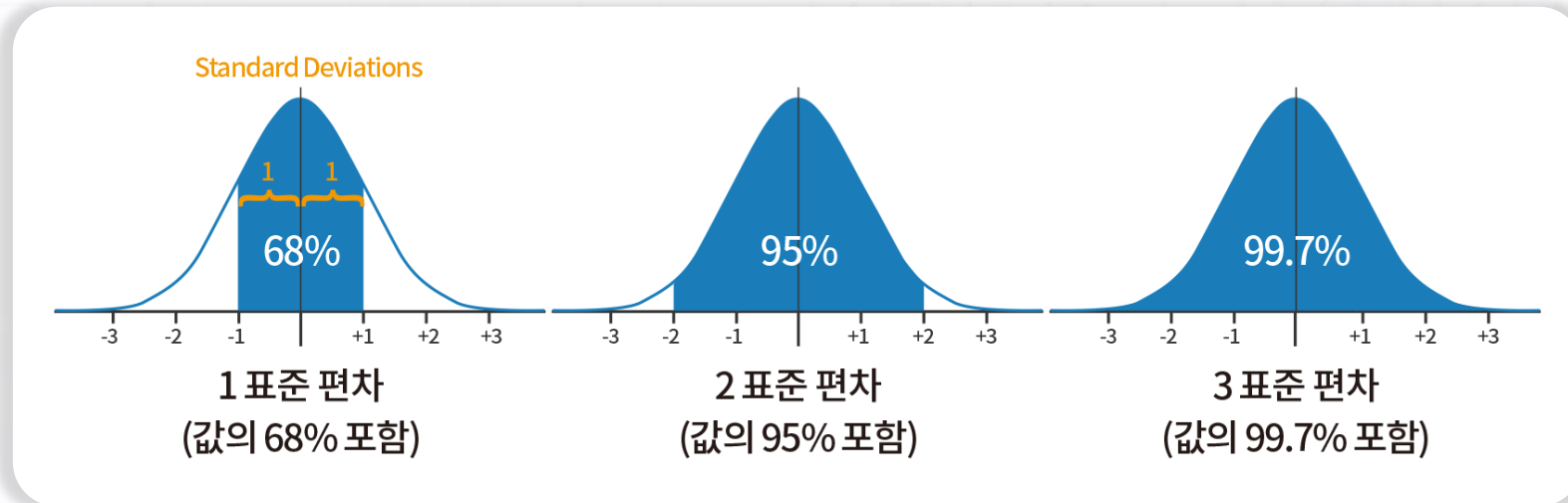


09 | 이상값 처리: 표준 편차

△ 표준 편차(standard deviation)

◆ 데이터의 분포가 정규 분포를 따르는 경우 데이터의 표준 편차를 이용해 이상치를 탐지하는 방법임

➢ 아래와 같이 표준 편차(파란색 범위)를 벗어나는 데이터는 이상치로 간주될 수 있음을 의미함





09 | 이상값 처리: 표준 편차

△ 표준 편차(standard deviation)

◆ 표준 편차는 아래 용어로 대체할 수 있음

➤ 표준 점수(Standard score)

➤ 시그마(Sigma)

➤ Z-점수(Z-score)

$$Z = \frac{X - \mu}{\sigma}$$

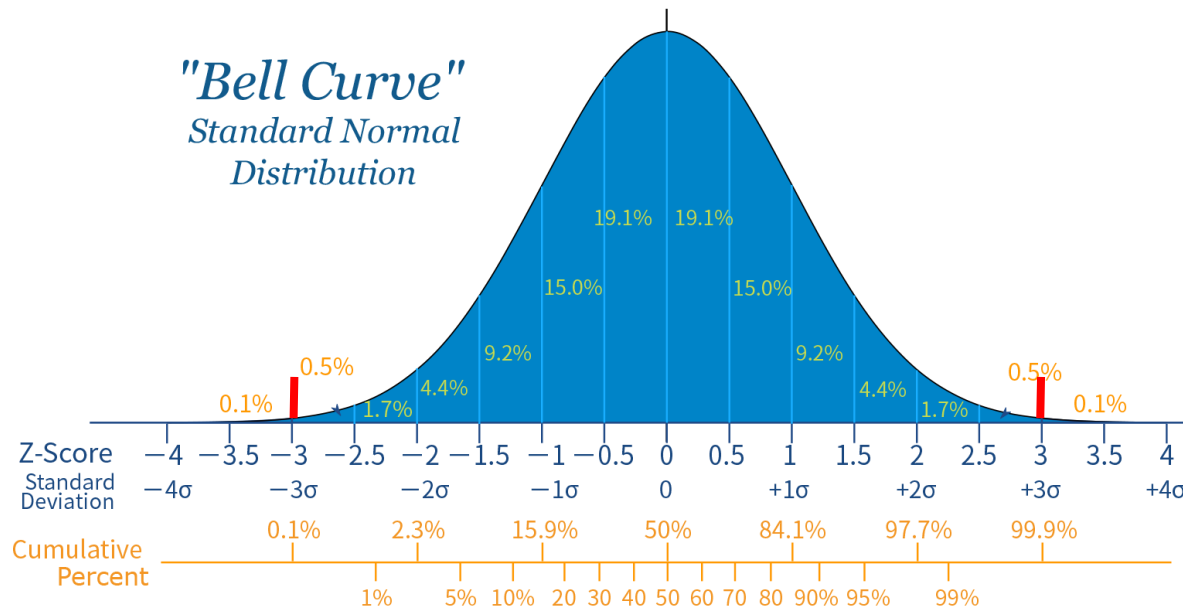
— 여기서 Z는 Z-점수, μ 는 평균, σ 는 표준편차임



09 | 이상값 처리: 표준 편차

△ 표준 편차(standard deviation)

- ◆ 데이터의 **Z-점수**는 해당 데이터가 **평균**으로부터 얼마의 **표준 편차만큼 벗어나 있는지를 의미함**
 - 예를 들어 ± 3 표준 편차 만큼을 벗어나는 데이터를 이상치로 처리하는 것임
 - ▬ 즉, Z-점수가 ± 3 표준 편차를 벗어나면 **이상치로 처리하는 작업**을 의미함





09 | 이상값 처리: 표준 편차

△ 표준 편차(standard deviation)

◆ 다음은 데이터에서 **평균**, **표준 편차**를 계산하는 코드이다.

➤ 아래의 경우 **평균=5.5**, **표준 편차=6.4013**인 것을 알 수 있음

```
data = [1, 3, 3, 2, 4, 1, 1, 12, 1, 2, 3, 2, 1, 2, 1, 11, 25, 4, 5, 9, 7, 21]
mean = np.mean(data) # 평균
std = np.std(data)    # 표준편차

print('데이터의 평균은', mean) # 데이터의 평균은 5.5
print('데이터의 표준 편차는', std) # 데이터의 표준 편차는 6.401349289585183
```



09 | 이상값 처리: 표준 편차

- ◆ 다음은 Z-점수를 이용해 이상값을 찾아내는 코드이다.
 - 데이터에서 평균, 표준 편차를 이용해 Z-점수를 계산함
 - Z-점수가 ± 3 표준 편차를 벗어나는 데이터를 찾아냄
 - 아래의 경우 “25”가 이상값인 것을 알 수 있음

```
threshold = 3
outlier = []

for i in data:
    z = (i-mean)/std
    print(z)
    if abs(z) > threshold:
        outlier.append(i)

print('데이터셋 내의 이상값은', outlier) # 데이터셋 내의 이상값은 [25]
```



09 | 이상값 처리: 표준 편차

◆ 다음은 Z-점수를 이용해 이상값을 찾아내는 코드이다.

> `scipy.stats.zscore()` 함수로 Z-점수를 계산함

- 데이터에서 Z-점수를 계산하고, Z-점수가 ± 3 표준 편차를 벗어나는 데이터를 찾아냄
- 아래의 경우 “25”가 이상값인 것을 알 수 있음

```
from scipy import stats

data = [1, 3, 3, 2, 4, 1, 1, 12, 1, 2, 3, 2, 1, 2, 1, 11, 25, 4, 5, 9, 7, 21]
threshold = 3

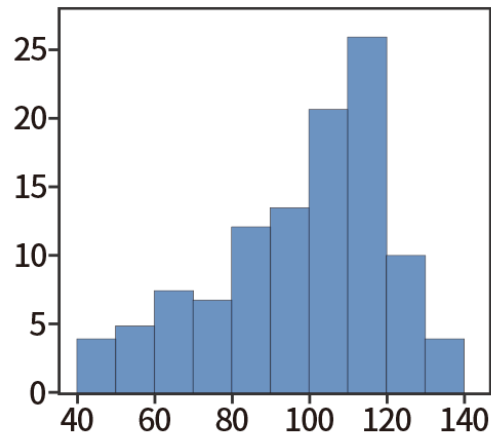
outlier = [i for i, z in zip(data, stats.zscore(data)) if abs(z) > threshold]
print('데이터셋 내의 이상값은', outlier)  # 데이터셋 내의 이상값은 [25]
```



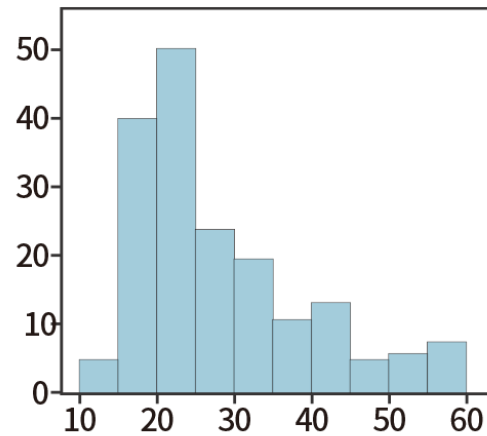
10 | 정규분포

◆ (참고) 정규 분포

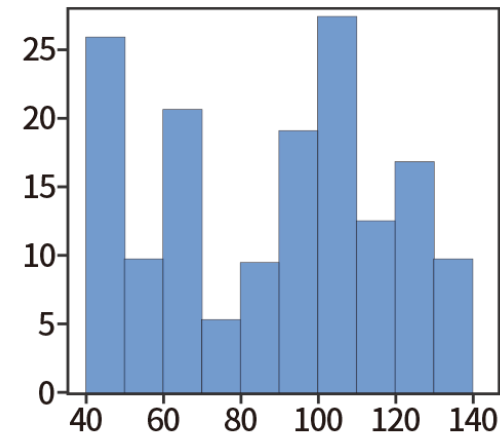
➤ 데이터는 아래 그림과 같이 다양한 방식으로 분산(확산) 될 수 있음



왼쪽으로 더 확산



오른쪽으로 더 확산



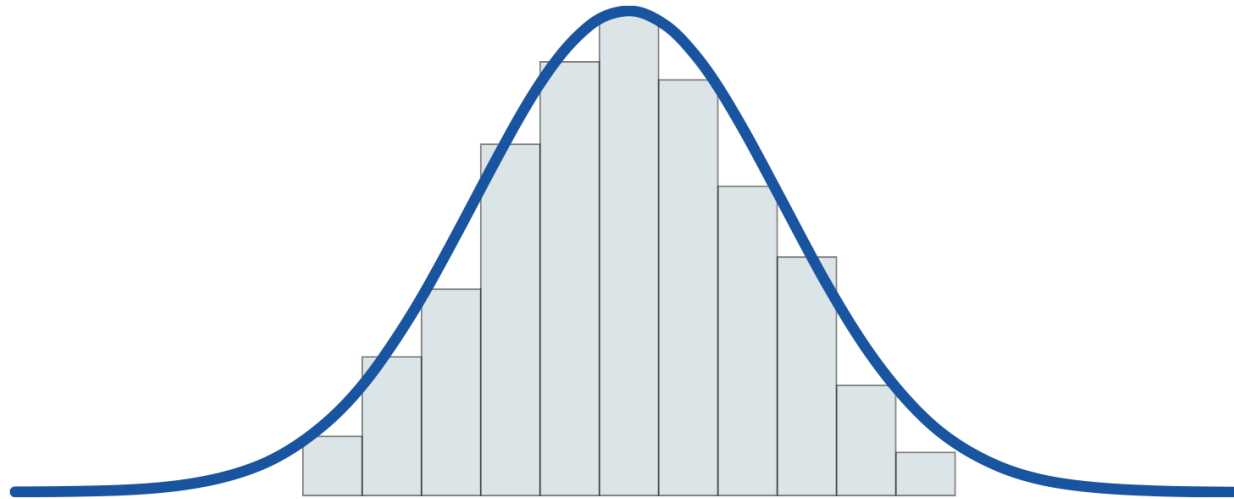
모두 뒤죽박죽 분산



10 | 정규분포

◆ (참고) 정규 분포

- ▶ 일반적으로 데이터가 좌우 편향 없이 중앙 값을 중심으로 하는 경향이 “정규 분포”에 가까워지는 경우가 많음

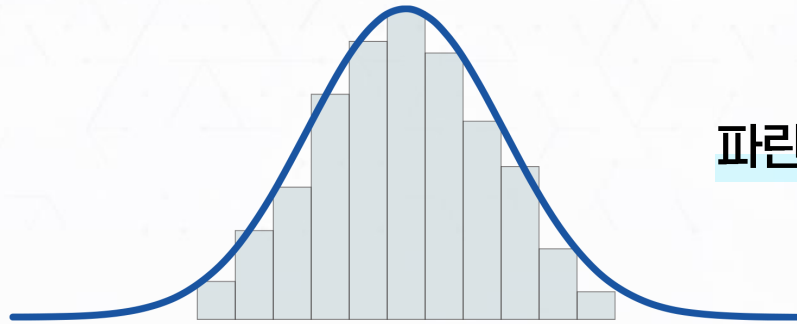


파란색 곡선이 정규분포



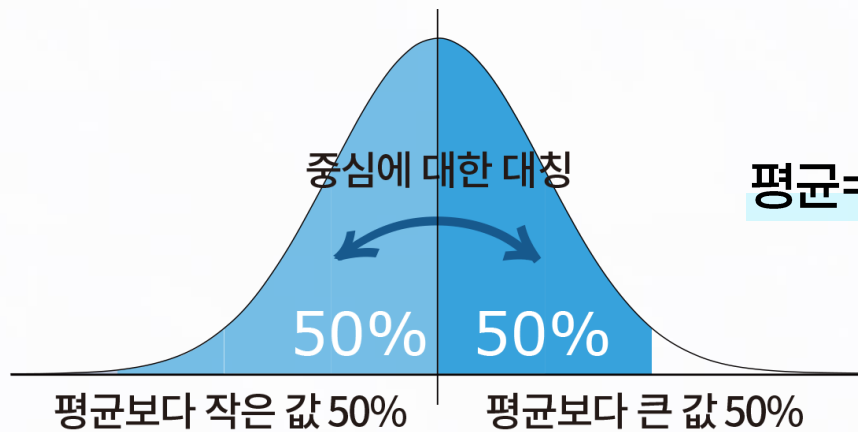
10 | 정규분포

> 아래 그림처럼 모양이 종처럼 보이기 때문에 종종 “벨 커브”라고 불림



파란색 곡선이 정규분포

— 위와 같은 데이터가 “정규 분포”임



평균=중앙값=최빈값



11 | 정규분포 → 표준정규분포

> 평균으로부터의 표준편차의 수는 “Z-점수”라고도 함

– Z-점수는 평균을 빼고, 표준 편차로 나누어 줌

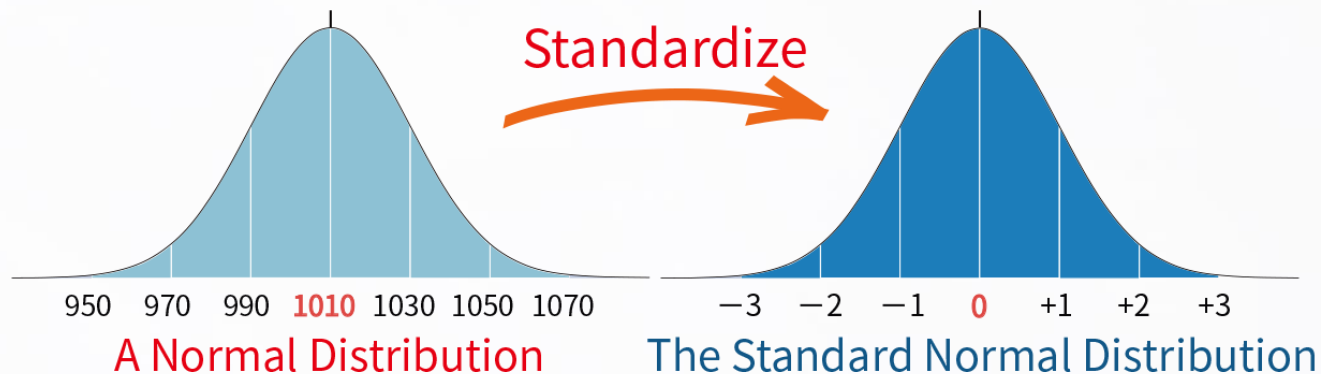
$$Z_i = \frac{x_i - \bar{x}}{s}$$

x_i : i 번째 x 값, \bar{x} : x의 평균,
 s : x의 표준편차, Z_i : x_i 번째 Z-점수

❖ 이와 같은 것을 “표준화”라고 부름

❖ 즉, 표준화는 어떤 정규분포를 표준정규분포로 변환할 수 있음

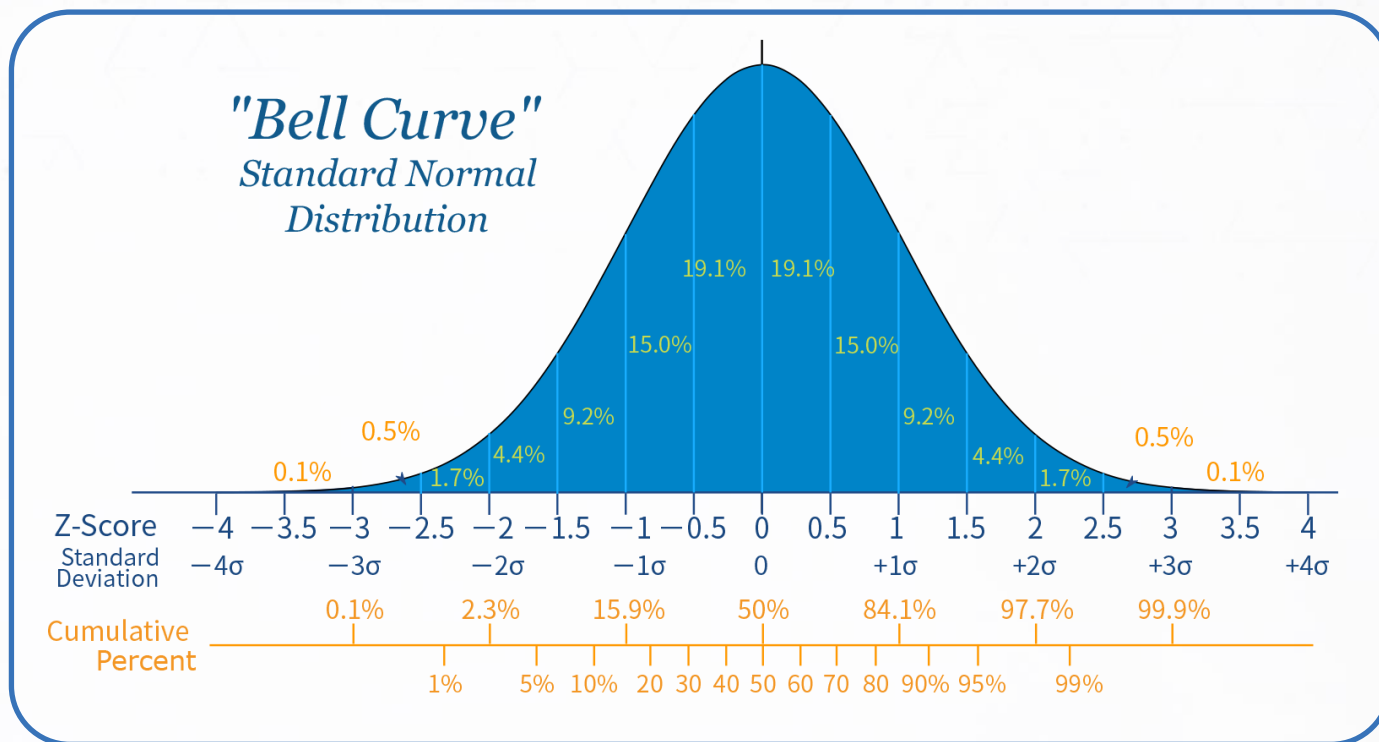
➔ 표준정규분포는 평균 0이고 표준 편차가 1인 정규분포임





12 | 표준정규분포

- ◆ 다음은 표준편차의 절반에 대한 백분율과 누적 백분율이 포함된 표준정규분포임



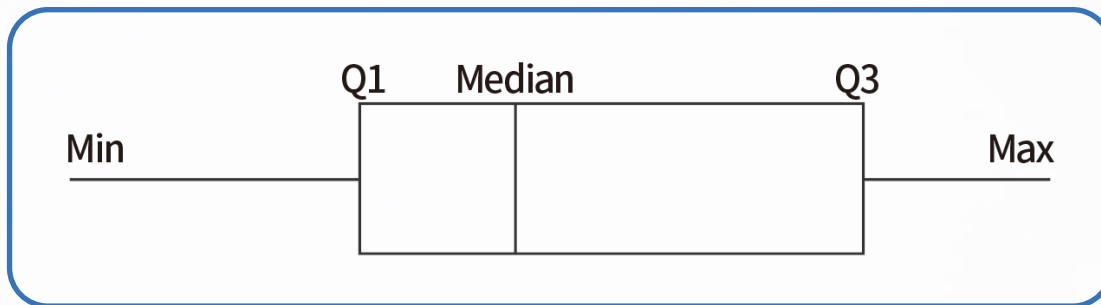


13 | 이상값 처리: 사분위 범위

△ 박스 플롯(box plot)의 사분위 범위(IQR)

◆ 박스 플롯은 다음을 포함하는 데이터 세트의 5개 숫자 요약을 표시하는 플롯 유형임

- ▶ 최소값(minimum)
- ▶ 첫 번째 사분위수(25% 백분위수)
- ▶ 중앙값(median)
- ▶ 세 번째 사분위수(75% 백분위수)
- ▶ 최대값(maximum)





13 | 이상값 처리: 사분위 범위

◆ 상자 그림은 다음과 같은 절차로 그릴 수 있음

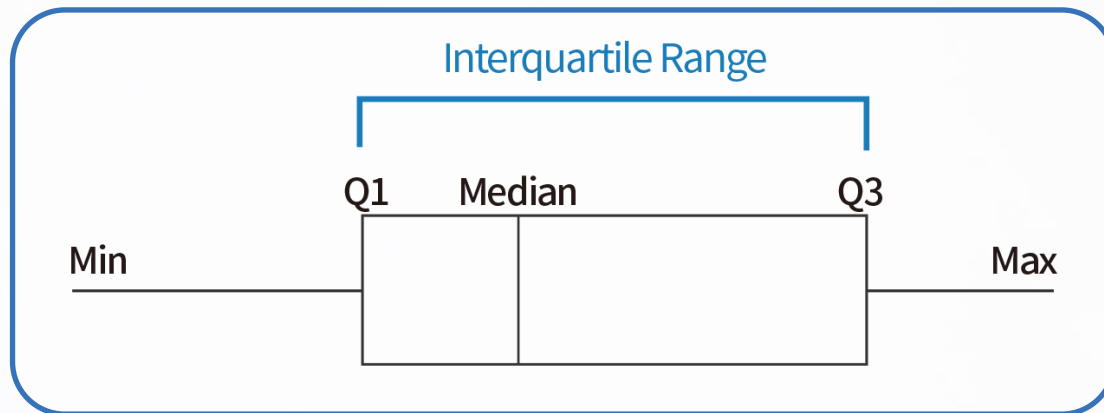
➤ 첫 번째 사분위수에서 세 번째 사분위수까지 상자를 그림

➤ 그런 다음 중앙값에 수직선을 그림

➤ 마지막으로 사분위수에서 최소값과 최대값까지 “수염”을 그림

➤ IQR(Interquartile Range)로 약칭하는 사분위수 범위는 제3사분위수와 제1사분위수 간의 차이임

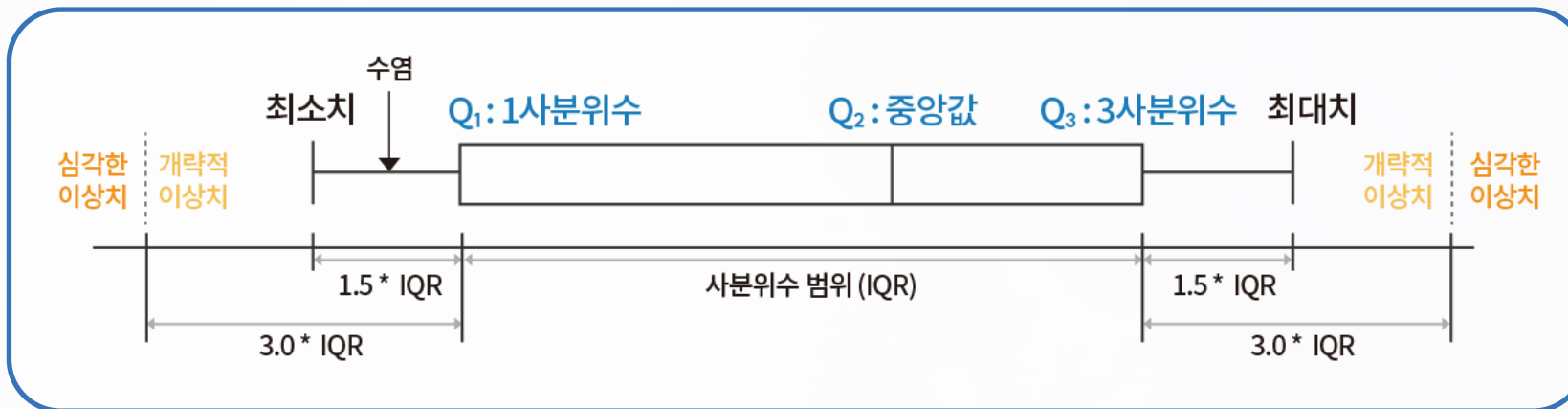
– 이것은 주어진 데이터 세트에서 값의 중간 50%가 얼마나 퍼져 있는지 알려줌





13 | 이상값 처리: 사분위 범위

- ◆ 박스 플롯의 **사분위 범위**(IQR with Box plots)는 **데이터의 분포**가 **정규 분포를 이루지 않거나 한 쪽으로 비뚤어진 경우**에 이용함
 - 즉, 데이터의 **IQR**(Interquartile Range) **값**을 이용해 **이상치**를 **탐지하는 방법**임
 - ─ 아래의 그림은 **IQR 값**을 이용해 **이상치**를 **처리하는 기준**임
 - ❖ $(Q_1 - 1.5 * IQR)$ 보다 작거나 $(Q_3 + 1.5 * IQR)$ 보다 큰 데이터는 **이상치**로 처리함
 - ❖ $(Q_1 - 3.0 * IQR)$ 보다 작거나 $(Q_3 + 3.0 * IQR)$ 보다 큰 데이터는 **심각한 이상치**로 처리함





14 | IQR을 이용한 이상값 탐지 예제

- ◆ 다음은 IQR을 이용해 이상값을 찾아내는 `get_outlier()` 함수를 정의하는 코드이다.
 - IQR으로 이상값을 찾는 `get_outlier()` 함수는 세 개의 인자 값을 넘겨 받음
 - `df`=데이터프레임 객체, `column`=열 이름, `weight`=이상치 가중치(기본값=1.5)

```
# 이상값을 찾아내는 함수 정의
def get_outlier(df=None, column=None, weight=1.5):
    quantile_25 = np.percentile(df[column].values, 25) # Q1 계산
    quantile_75 = np.percentile(df[column].values, 75) # Q3 계산

    IQR = quantile_75 - quantile_25          # IQR = Q3 - Q1
    IQR_weight = IQR * weight                # IQR * 1.5

    lowest = quantile_25 - IQR_weight
    highest = quantile_75 + IQR_weight

    outlier_idx = df[column][(df[column] < lowest) | (df[column] > highest)].index
    return outlier_idx
```



14 | IQR을 이용한 이상값 탐지 예제

- > num1, num2 두 개의 속성으로 구성된 **df_outlier** 데이터프레임 객체를 생성함
- > **num1** 속성의 **이상값**을 찾아냄
 - 실행결과 **16 인덱스**(=25)와 **21 인덱스**(=21)의 값이 **이상치 값인 것**을 알 수 있음

```
df_outlier = pd.DataFrame({'num1':[1, 3, 3, 2, 4, 1, 1, 12, 1, 2, 3, 2, 1, 2, 1, 11, 25, 4, 5, 9, 7, 21],  
                           'num2':[11, 13, 13, 12, 14, 11, 11, 22, 1, 12, 13, 12, 11, 12, 11, 21, 35, 14, 15, 19, 17, 31]  
                           })  
  
# num1 속성의 이상값 찾기  
outlier_idx = get_outlier(df=df_outlier, column='num1', weight=1.5)  
df_outlier['num1'][outlier_idx]
```

```
16    25  
21    21  
Name: num 1, dtype: int64
```



14 | IQR을 이용한 이상값 탐지 예제

> num2 속성의 이상값을 찾아냄

— 실행결과 8 인덱스(=1), 16 인덱스(=35)와 21 인덱스(=31)의 값이 이상치 값인 것을 알 수 있음

```
outlier_idx = get_outlier(df=df_outlier, column='num2', weight=1.5)
df_outlier['num2'][outlier_idx]
```

```
8      1
16     35
21     31
Name: num 2, dtype: int64
```



15 | IQR을 이용한 이상값 제거 예제

- ◆ 다음은 IQR을 이용해 이상값을 찾아내는 get_outlier() 함수로 이상값을 찾아내고, 이상값을 제거하는 코드이다.

➤ num1 속성의 이상값을 찾아내고 df_outlier 데이터프레임에서 제거함

— 실행결과 16, 21 인덱스가 df_outlier 데이터프레임 객체에서 제거된 것을 알 수 있음

```
df_outlier = pd.DataFrame({'num1':[1, 3, 3, 2, 4, 1, 1, 12, 1, 2, 3, 2, 1, 2, 1, 11, 25, 4, 5, 9, 7, 21],  
                           'num2':[11, 13, 13, 12, 14, 11, 11, 22, 1, 12, 13, 12, 11, 12, 11, 21, 35, 14, 15, 19, 17, 31]  
                           })
```

```
outlier_idx = get_outlier(df=df_outlier, column='num1', weight=1.5)
```

```
df_outlier.drop(outlier_idx, axis=0, inplace=True)
```

```
df_outlier
```

```
# 실행결과 16, 21 인덱스가 제거된 것을 볼 수 있다.
```

	Num1	num2
0	1	11
1	3	13
2	3	13
3	2	12
4	4	14
5	1	11
6	1	11
7	12	22
8	1	1
9	2	12
10	3	13
11	2	12
12	1	11
13	2	12
14	1	11
15	11	21
17	4	14
18	5	15
19	9	19
20	7	17



15 | IQR을 이용한 이상값 제거 예제

> num2 속성의 이상값을 찾아내고 df_outlier 데이터프레임에서 제거함

— 실행결과 8, 16, 21 인덱스가 df_outlier 데이터프레임 객체에서 제거된 것을 알 수 있음

```
df_outlier = pd.DataFrame({'num1':[1, 3, 3, 2, 4, 1, 1, 12, 1, 2, 3, 2, 1, 2, 1, 11, 25, 4, 5, 9, 7, 21],  
                           'num2':[11, 13, 13, 12, 14, 11, 11, 22, 1, 12, 13, 12, 11, 12, 11, 21, 35, 14, 15, 19, 17, 31]  
                           })
```

```
outlier_idx = get_outlier(df=df_outlier, column='num2', weight=1.5)
```

```
df_outlier.drop(outlier_idx, axis=0, inplace=True)
```

```
df_outlier
```

```
# 실행결과 8,16,21 인덱스가 제거된 것을 볼 수 있다.
```

	Num1	num2
0	1	11
1	3	13
2	3	13
3	2	12
4	4	14
5	1	11
6	1	11
7	12	22
9	2	12
10	3	13
11	2	12
12	1	11
13	2	12
14	1	11
15	11	21
17	4	14
18	5	15
19	9	19
20	7	17



16 | 샘플 데이터 집합으로 이상치 처리하기

△ 샘플 데이터 집합으로 이상치 처리하기

◆ 대학생 샘플 데이터 집합 읽어옴

➤ 아래의 표는 **대학생 샘플 데이터**의 **설명**임

NO	속성명	속성 설명
1	성명	학생 이름
2	학년	학년(1=1학년, 2=2학년, 3=3학년, 4=4학년)
3	키(cm)	키(cm)
4	몸무게(kg)	몸무게(kg)
5	취미	취미

➤ 대학생 샘플 데이터 집합은 위의 표와 같이 5개의 속성과 500개의 관측치로 구성됨



16 | 샘플 데이터 집합으로 이상치 처리하기

△ 샘플 데이터 집합으로 이상치 처리하기

◆ 다음은 대학생 **샘플 데이터 집합**을 읽어오는 코드이다.

➤ 대학생 샘플 데이터 집합의 **형상 (500, 5)인 것**을 알 수 있음

— “std_sample_data_filled.xlsx” 파일은 **결측치 처리가 완료된 데이터 집합**임

```
# 데이터 읽어오기
sample_data = pd.read_excel(os.getcwd()+'/std_sample_data_filled.xlsx')

# 데이터의 형상 # - shape 속성 : 데이터의 (행, 열) 크기를 확인
print(sample_data.shape) # (500, 5)
```



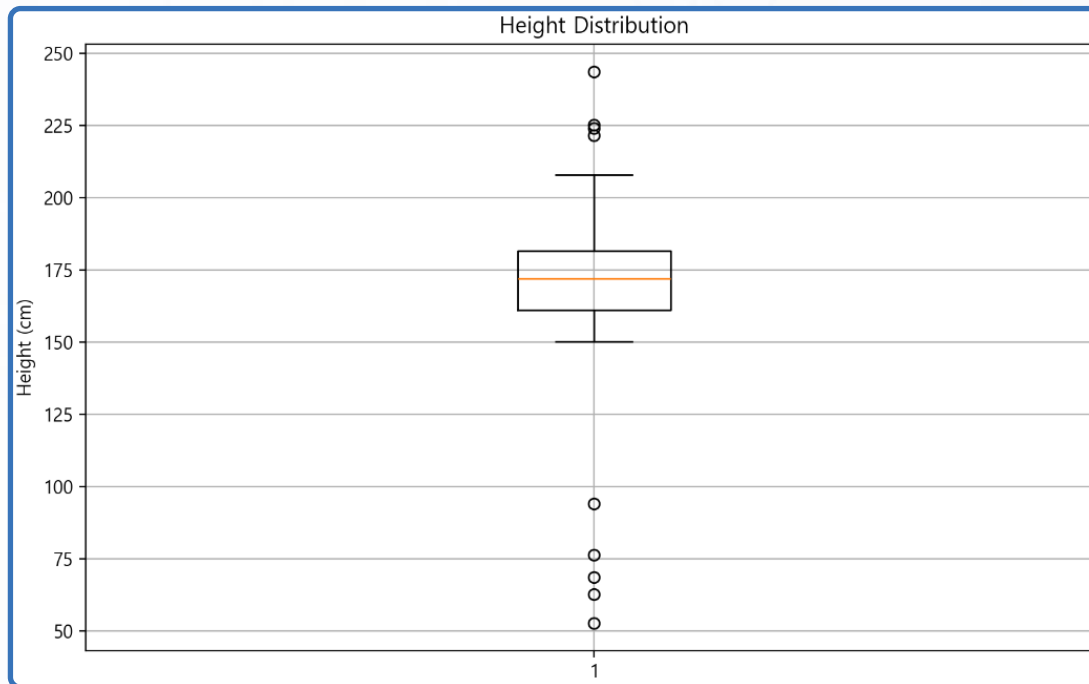
16 | 샘플 데이터 집합으로 이상치 처리하기

△ 샘플 데이터 집합으로 이상치 처리하기

◆ 다음은 대학생 샘플 데이터 집합에서 키(cm) 속성으로 상자그림을 그리는 코드이다.

➤ 아래 그림과 같이 이상치가 존재하는 것을 볼 수 있음

```
# 상자그림 생성
plt.figure(figsize=(10, 6))
plt.boxplot(sample_data["키(cm)"])
plt.title('Height Distribution')
plt.ylabel('Height (cm)')
plt.grid(True)plt.show()
```





16 | 샘플 데이터 집합으로 이상치 처리하기

△ 샘플 데이터 집합으로 이상치 처리하기

◆ 다음은 대학생 샘플 데이터 집합에서 키(cm) 속성의 이상치 데이터 행을 출력하는 코드이다.

➤ 아래와 같이 9개 데이터가 이상치인 것을 볼 수 있음

```
# 이상치 탐지: 상자그림을 이용한 IQR 방법 적용
Q1 = sample_data["키(cm)"].quantile(0.25)
Q3 = sample_data["키(cm)"].quantile(0.75)
IQR = Q3 - Q1
```

```
# 이상치 경계 설정
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

```
# 이상치 데이터 필터링
outliers = sample_data[(sample_data["키(cm)"] < lower_bound) | (sample_data["키(cm)"] > upper_bound)]
```

```
# 이상치 데이터 출력
print(outliers)
```

	성명	학년	키(cm)	몸무게(kg)	취미
75	최수현	3	68.521901	60.0	축구
96	박민수	1	94.095151	77.8	테니스
103	최민수	2	62.744453	59.4	달리기
174	정준호	4	243.731618	54.9	골프
199	이하윤	2	225.148292	65.2	축구
319	장지우	3	52.789536	76.4	등산
447	임지훈	3	76.362893	47.5	골프
458	조하윤	4	221.636048	51.3	수영
467	윤지우	1	224.002422	75.0	수영



16 | 샘플 데이터 집합으로 이상치 처리하기

△ 샘플 데이터 집합으로 이상치 처리하기

◆ 다음은 대학생 샘플 데이터 집합에서 키(cm) 속성의 이상치 데이터를 모두 제거하는 코드이다.

➤ 아래와 같이 이상치 데이터를 모두 제거한 후 그 결과를 "df_no_outliers_height" 객체에 할당함

```
# 이상치 탐지: 상자그림을 이용한 IQR 방법 적용
Q1 = sample_data["키(cm)"].quantile(0.25)
Q3 = sample_data["키(cm)"].quantile(0.75)
IQR = Q3 - Q1

# 이상치 경계 설정
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# 이상치 제거
df_no_outliers_height = sample_data[(sample_data["키(cm)"] >= lower_bound) & (sample_data["키(cm)"] <= upper_bound)]
```



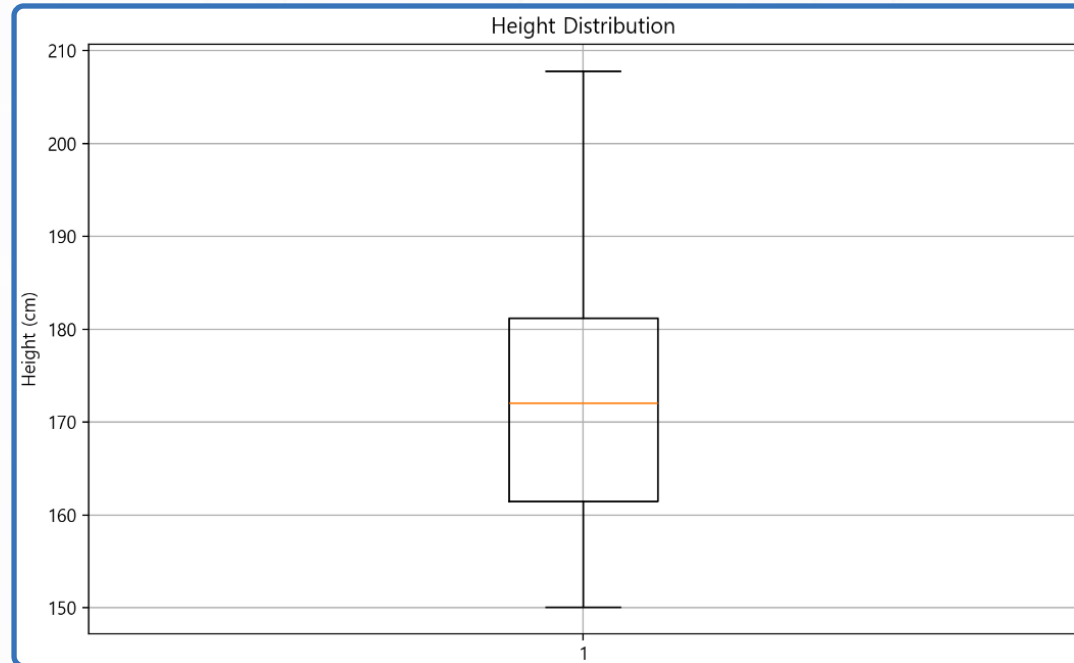
16 | 샘플 데이터 집합으로 이상치 처리하기

△ 샘플 데이터 집합으로 이상치 처리하기

◆ 다음은 대학생 샘플 데이터 집합에서 키(cm) 속성의 이상치 데이터가 제거된 데이터로 상자그림을 그리는 코드이다.

➤ 아래 그림과 같이 이상치 데이터가 모두 제거된 것을 볼 수 있음

```
# 상자그림 생성
plt.figure(figsize=(10, 6))
plt.boxplot(df_no_outliers_height["키(cm)"])
plt.title('Height Distribution')
plt.ylabel('Height (cm)')
plt.grid(True)
plt.show()
```





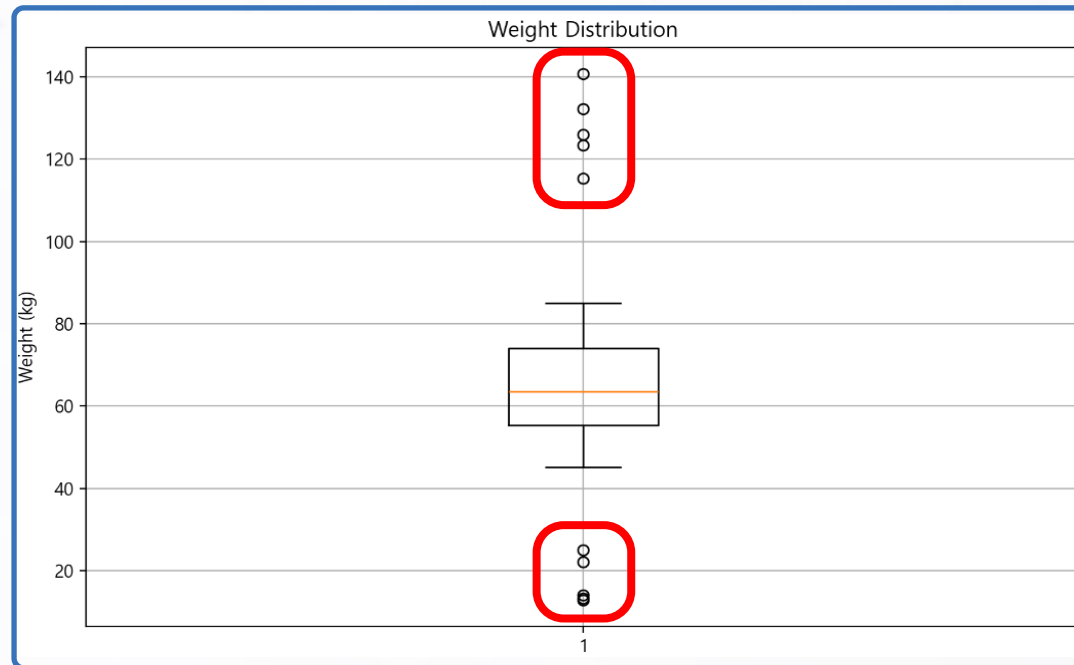
16 | 샘플 데이터 집합으로 이상치 처리하기

△ 샘플 데이터 집합으로 이상치 처리하기

◆ 다음은 대학생 샘플 데이터 집합에서 **몸무게(kg)** 속성으로 **상자그림**을 그리는 코드이다.

➤ 아래 그림과 같이 **이상치가 존재하는 것**을 볼 수 있음

```
# 상자그림 생성
plt.figure(figsize=(10, 6))
plt.boxplot(sample_data["몸무게(kg)"])
plt.title('Height Distribution')
plt.ylabel('Height (cm)')
plt.grid(True)plt.show()
```





16 | 샘플 데이터 집합으로 이상치 처리하기

△ 샘플 데이터 집합으로 이상치 처리하기

◆ 다음은 대학생 샘플 데이터 집합에서 **몸무게(kg)** 속성의 **이상치 데이터 행**을 **출력**하는 코드이다.

➤ 아래와 같이 **10개 데이터**가 **이상치인 것**을 볼 수 있음

```
# 이상치 탐지: 상자그림을 이용한 IQR 방법 적용
Q1 = sample_data["몸무게(kg)"].quantile(0.25)
Q3 = sample_data["몸무게(kg)"].quantile(0.75)
IQR = Q3 - Q1
```

```
# 이상치 경계 설정
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

```
# 이상치 데이터 필터링
outliers = sample_data[(sample_data["몸무게(kg)"] < lower_bound) | (sample_data["몸무게(kg)"] > upper_bound)]
```

```
# 이상치 데이터 출력
print(outliers)
```

	성명	학년	키 (cm)	몸무게(kg)	취미
34	윤서연	2	178.6	14.015741	달리기
42	정하윤	3	185.4	123.333757	골프
62	임지훈	1	182.3	12.935121	달리기
74	박예은	1	153.7	25.030951	테니스
131	박수현	1	174.8	22.216642	수영
157	정지우	2	186.1	13.353255	탁구
207	윤현우	1	155.7	140.792628	달리기
400	임현우	4	178.4	125.927393	축구
427	장수현	2	152.9	132.234416	축구
498	임지민	4	158.9	115.367520	탁구



16 | 샘플 데이터 집합으로 이상치 처리하기

△ 샘플 데이터 집합으로 이상치 처리하기

- ◆ 다음은 대학생 샘플 데이터 집합에서 **몸무게(kg)** 속성의 **이상치 데이터**를 **모두 제거**하는 코드이다.
 - 아래와 같이 이상치 데이터를 모두 제거한 후 그 결과를 “df_no_outliers_weight” 객체에 할당함

```
# 이상치 탐지: 상자그림을 이용한 IQR 방법 적용
Q1 = sample_data["몸무게(kg)"].quantile(0.25)
Q3 = sample_data["몸무게(kg)"].quantile(0.75)
IQR = Q3 - Q1

# 이상치 경계 설정
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# 이상치 제거
df_no_outliers_weight = sample_data[(sample_data["몸무게(kg)"] >= lower_bound) &
(sample_data["몸무게(kg)"] <= upper_bound)]
```



16 | 샘플 데이터 집합으로 이상치 처리하기

△ 샘플 데이터 집합으로 이상치 처리하기

◆ 다음은 대학생 샘플 데이터 집합에서 **몸무게(kg)** 속성의 **이상치 데이터**가 **제거된 데이터**로 **상자그림**을 그리는 코드이다.

➤ 아래 그림과 같이 **이상치 데이터**가 **모두 제거된 것**을 볼 수 있음

```
# 상자그림 생성
plt.figure(figsize=(10, 6))
plt.boxplot(df_no_outliers_weight["몸무게(kg)"])
plt.title('Height Distribution')
plt.ylabel('Height (cm)')
plt.grid(True)
plt.show()
```

