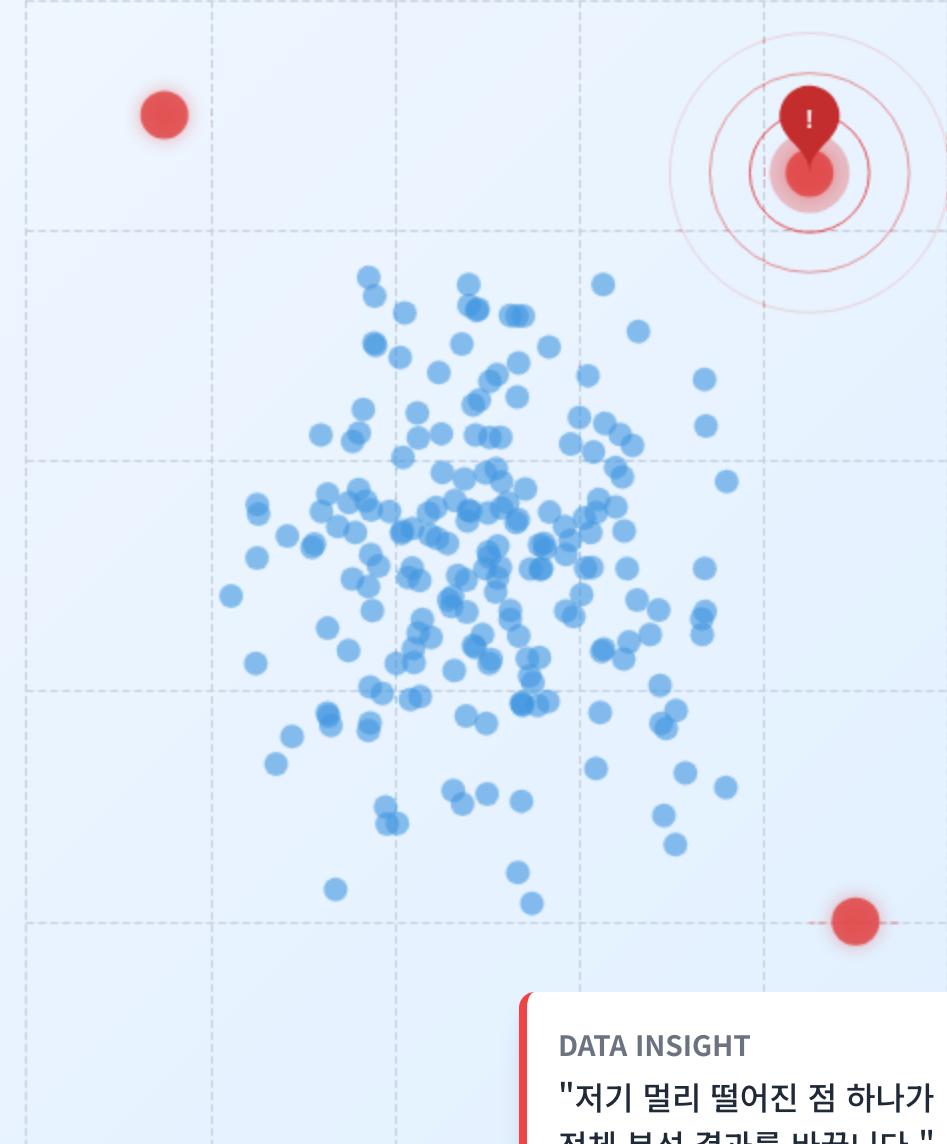


데이터 전처리의 핵심, 이상치(Outlier) 탐지

평균의 함정에서 벗어나
데이터의 진짜 모습을 보는 법

LECTURE 05

▣ 머신러닝 데이터 전처리 과정



이상치(Outlier)란 무엇인가?

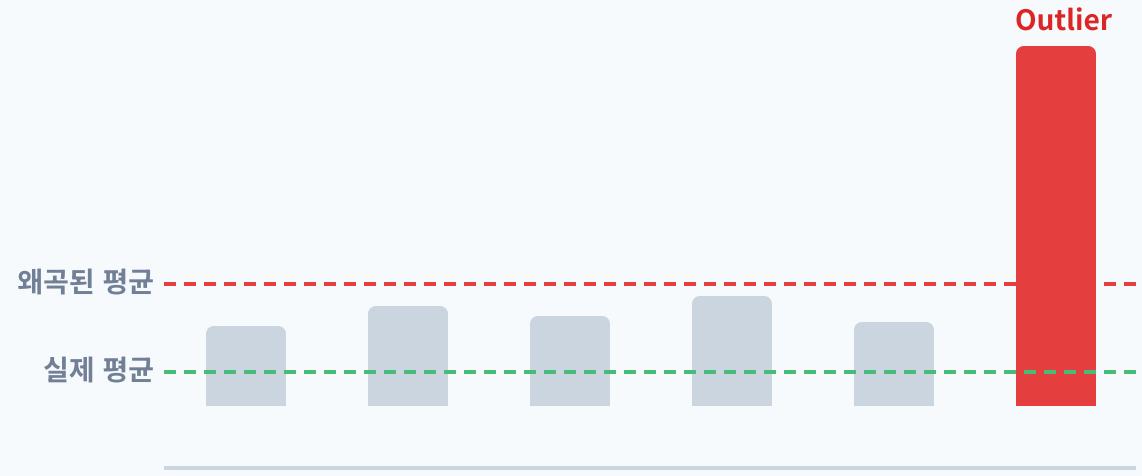
- 정의: 전체 데이터의 패턴이나 분포에서 현저하게 벗어난 관측값
- Noise(단순 오류)와 Anomaly(특이 현상)를 구분하는 것이 중요

💡 초보자를 위한 비유

"우리 반 친구들의 한 달 용돈 평균을 계산하는데,
갑자기 **이재용 회장**이 전학을 온다면?"

→ 평균이 수십억 원으로 왜곡되어
실제 반 친구들의 용돈 수준을 대변하지 못함

데이터 분포와 평균의 왜곡



Noise (잡음)

입력 오류, 센서 고장 등
→ 제거해야 할 대상



Anomaly (이상 징후)

사기 탐지, 새로운 트렌드 발견
→ 분석해야 할 핵심 기회

기초 통계량으로 힌트 얻기

🔍 describe() 해석:

데이터의 전반적인 요약을 한 줄로 확인

↔ Mean(평균) vs Median(중앙값 50%):

두 값의 차이가 크다면? → 이상치 의심

⚠ Min/Max 논리 확인:

도메인 상식에 어긋나는 값(Impossible Value) 점검

✖ 논리적 오류 예시

타이타닉 승객 데이터에서...

"나이(Age) 최댓값이 200세?"

"요금(Fare) 최솟값이 -50달러?"

→ 시각화 전에 수치만으로도 1차 필터링 가능

● ● ● df.describe()

Pandas Output Example

	Age	Fare
count	714.0	891.0
mean	29.7	32.2
std	14.5	49.7
min	0.4	0.0
25%	20.1	7.9
50%	28.0	14.4
75%	38.0	31.0
max	200.0	512.3

✖ 불가능한 값

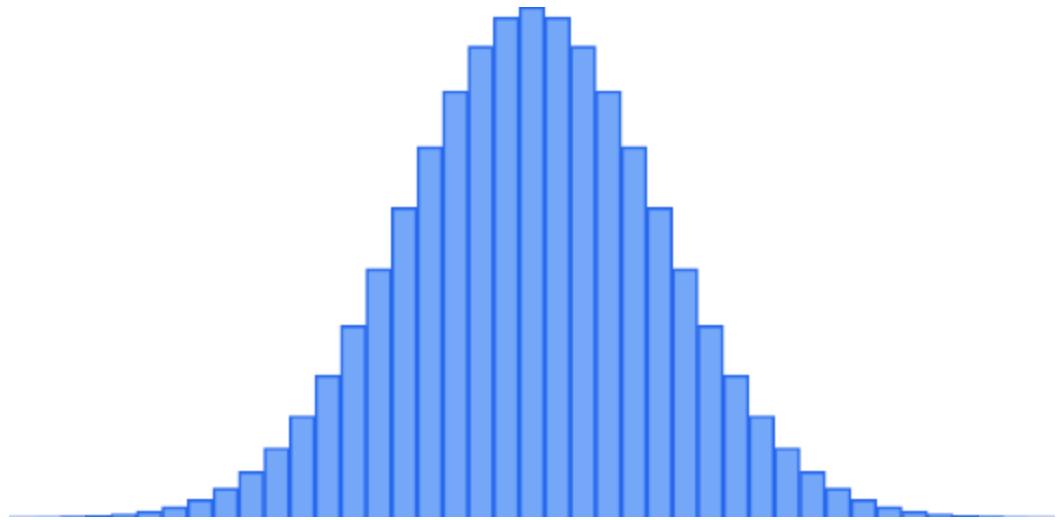
나이가 200세? → 명

백한 데이터 오류

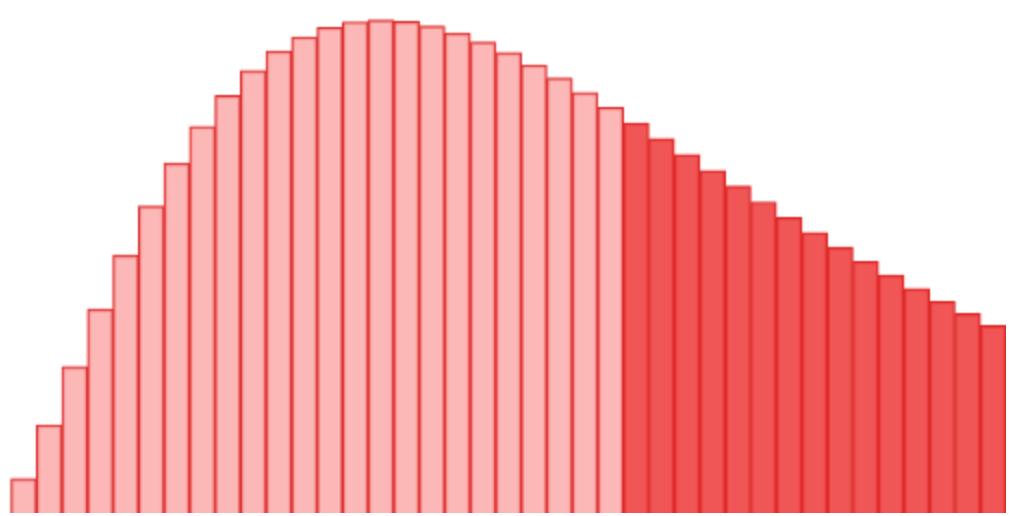
히스토그램으로 분포 확인

데이터의 모양(Shape)을 보면 이상치의 힌트가 보입니다.

✓ 정규 분포 (Normal)



⚠ 치우친 분포 (Skewed)



Bell Shape (종 모양)



Mean \approx Median인 이상적인 형태.

Z-Score 등 통계적 기법이 잘 작동합니다.

Right Skewed (우측 꼬리)

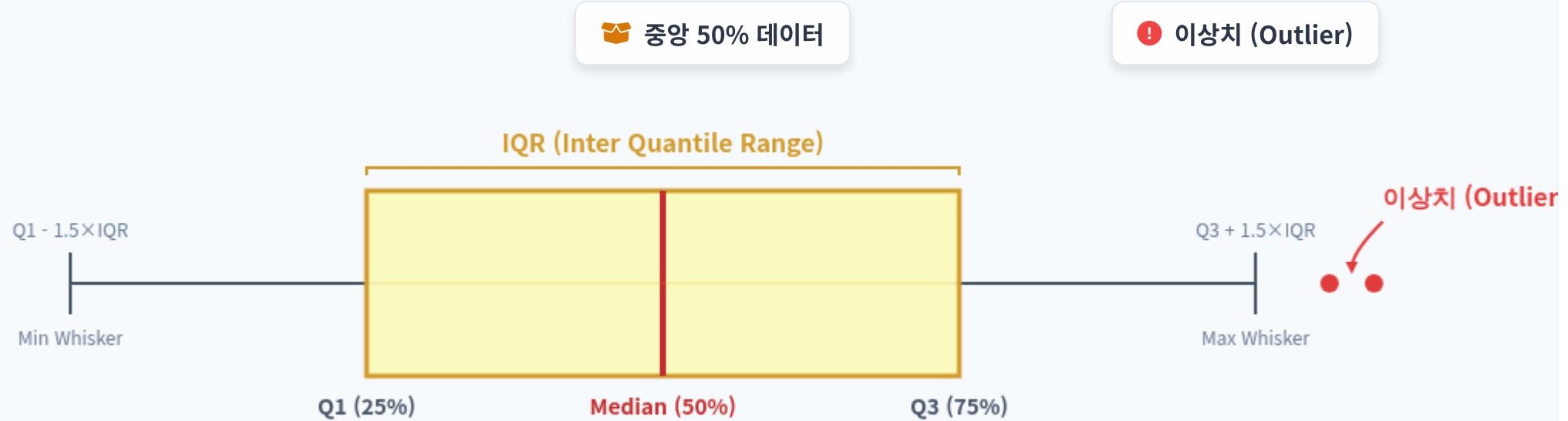


오른쪽 꼬리에 이상치(Outlier)가 숨어있습니다.

Log 변환이나 IQR 방식이 필수적입니다.

박스플롯(Boxplot) 구조 이해

데이터의 분포를 상자와 수염으로 요약해 보여주는 '택배 상자' 모델입니다.



실습: Titanic 요금(Fare) 이상치 확인



데이터 로드부터 시각화까지, 한 눈에 패턴 찾기

titanic_outlier_check.py

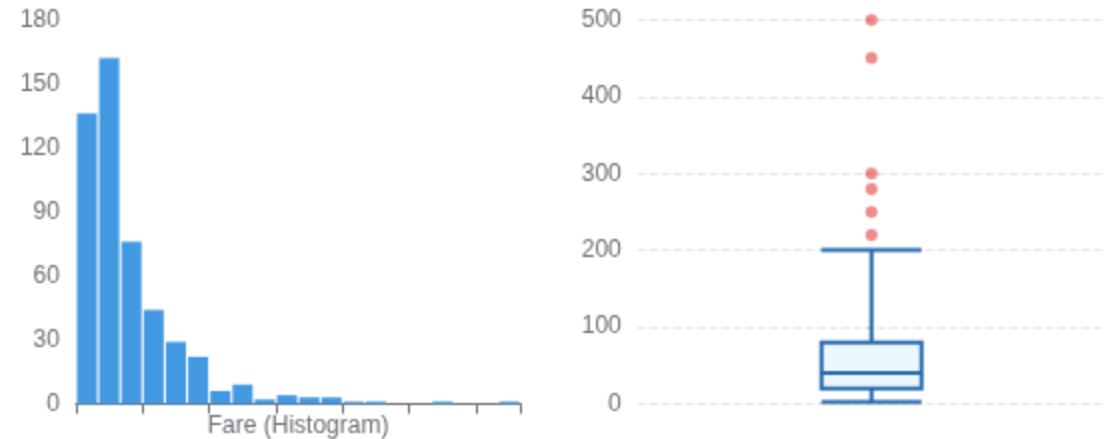
```
import seaborn as sns
# 1. 데이터 로드 (Seaborn 내장 데이터)
df = sns.load_dataset('titanic')

# 2. 'Fare' 컬럼 시각화 (Histogram & Boxplot)
fig, ax = plt.subplots(1, 2, figsize=(12, 5))

# 히스토그램: 분포의 치우침 확인
sns.histplot(df['fare'], bins=30, kde=True, ax=ax[0])
ax[0].set_title('Fare Distribution (Skewed)')

# 박스플롯: 이상치 점 확인
sns.boxplot(y=df['fare'], ax=ax[1])
ax[1].set_title('Boxplot of Fare')
plt.show()
```

EXECUTION RESULT



분석 포인트

- 오른쪽으로 긴 꼬리(Right Skewed):** 대부분의 승객은 저렴한 요금을 냈지만, 소수의 고액 요금자가 존재함.
- 박스플롯의 수많은 점들:** 상자(Box) 밖으로 벗어난 값들이 많음. 이들이 모두 '오류'일까? 아니면 '특실 승객'일까?

통계적 기준: Z-Score vs IQR

데이터의 분포 형태에 따라 적절한 이상치 탐지 도구를 선택하세요.



Z-Score

Standard Score

정규분포용

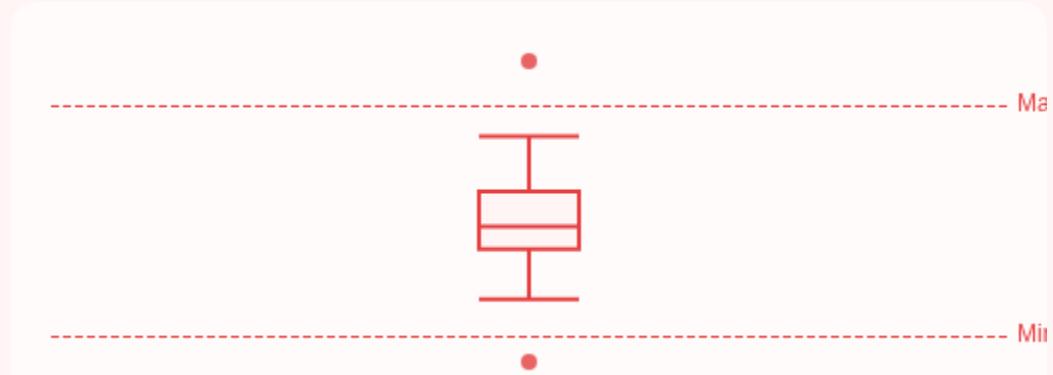


IQR 방식

Tukey Fences

치우친 분포용

VS



- ✓ **가정:** 데이터가 정규분포(Bell Shape)를 따른다고 가정
- ✓ **기준:** 평균(μ)에서 표준편차(σ)의 3배 이상 떨어진 값
- ❗ **단점:** 평균 자체가 이상치에 민감하여 기준이 흔들릴 수 있음

- ✓ **특징:** 데이터 분포 모양에 상관없이 사용 가능 (Non-parametric)
- ✓ **기준:** Q1, Q3 사분위수를 기준으로 울타리(Fence) 설정
- ★ **장점:** 이상치에 영향을 덜 받는 **강건한(Robust)** 방법

💡 선택 가이드: 데이터를 먼저 그려보세요!

종 모양 (Symmetric) → Z-Score

한쪽 쓸림 (Skewed) → IQR

방법 1: Z-Score (표준화 점수)

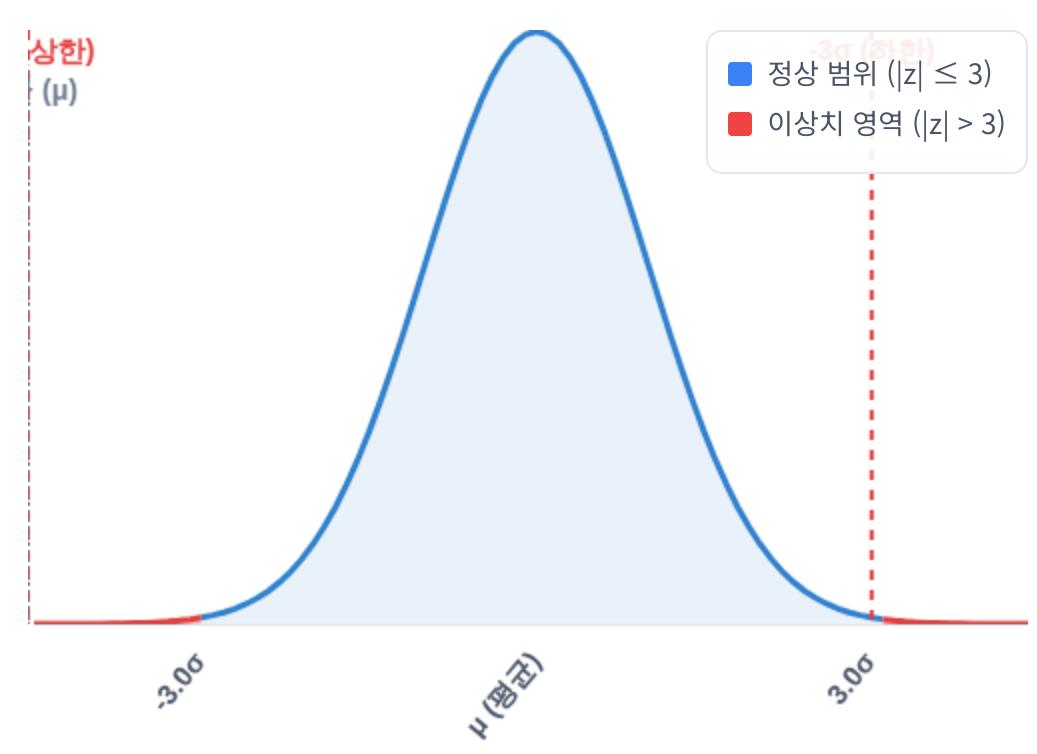
$$\text{공식 } Z = \frac{x - \mu}{\sigma}$$

- **판단 기준:** 통상적으로 $|z| > 3$ 이면 이상치
(평균에서 표준편차의 3배 이상 떨어진 값)
- ⚡ **특징:** 계산이 빠르고 단순하지만,
평균(μ)과 표준편차(σ) 자체가 이상치에 민감함

🏃‍♂️ 초보자를 위한 비유: 운동장 조회

"조회대(평균)를 기준으로 서 있는데,
친구들 뭉치(분포)에서 30미터(3표준편차) 이상
혼자 멀리 떨어져 있는 학생은 누구?"

정규분포와 임계치(Threshold)



* 3σ 범위 내에 전체 데이터의 약 99.7%가 포함됨

방법 2: IQR과 Tukey Fences

✓ IQR (Interquartile Range):

데이터의 중앙 50%를 나타내는 범위

공식: $IQR = Q3(75\%) - Q1(25\%)$

✓ Tukey Fences (이상치 경계):

Lower Limit: $Q1 - 1.5 \times IQR$

Upper Limit: $Q3 + 1.5 \times IQR$

✓ 특징: 평균/표준편차와 달리

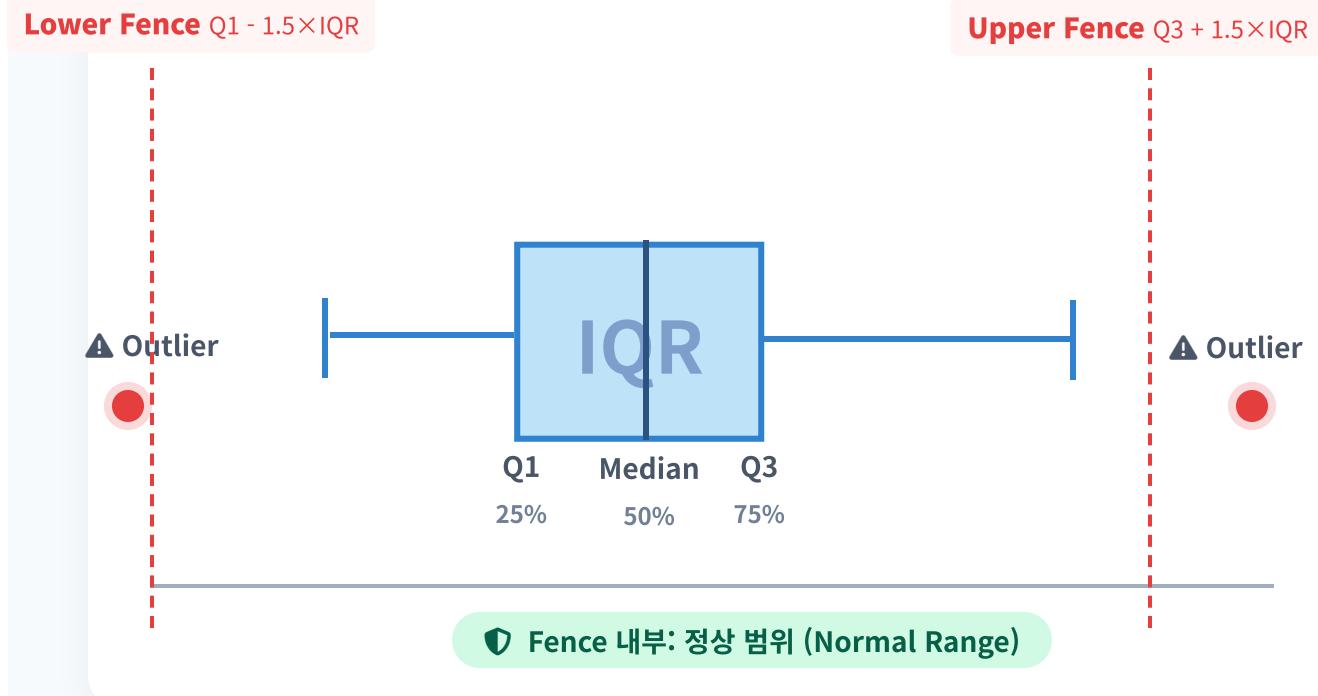
이상치 자체에 영향을 덜 받음(Robust)

💡 초보자를 위한 비유

"울타리(Fence)가 쳐진 마당 안에서 데이터들이 뛰어납니다."

이 울타리를 넘어간 공은 주워와야 할(검토 대상) 이상치입니다."

IQR 기준 이상치 탐지 구조



* 데이터 분포가 한쪽으로 치우쳐(Skewed) 있을 때 특히 유용합니다.



실습: Z-Score vs IQR 필터링

기준이 다르면 결과도 다르다? 교집합과 차집합 확인하기

outlier_filter_compare.py

```
from scipy.stats import zscore
import numpy as np

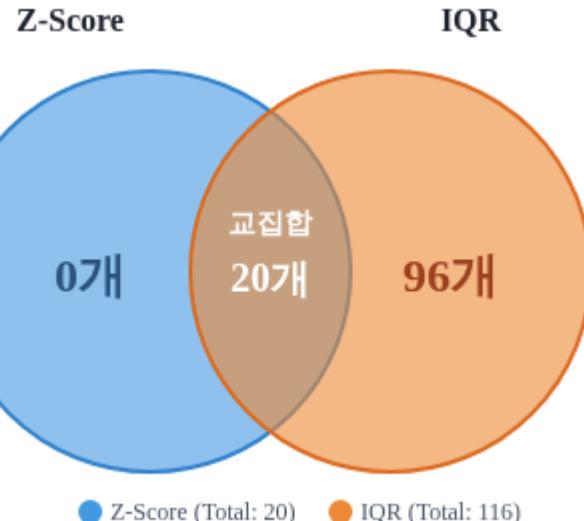
# == Method 1: Z-Score ==
# 정규분포 가정 하에  $|z| > 3$  필터링
df['z_score'] = zscore(df['fare'])
outlier_z = df[np.abs(df['z_score']) > 3]
print(f"Z-Score detected: {len(outlier_z)}")
```

```
# == Method 2: IQR (Robust) ==
# 4분위수를 이용한 Tukey Fences
Q1 = df['fare'].quantile(0.25)
Q3 = df['fare'].quantile(0.75)
IQR = Q3 - Q1

upper_fence = Q3 + (1.5 * IQR)
outlier_iqr = df[df['fare'] > upper_fence]
print(f"IQR detected: {len(outlier_iqr)}")
```

```
# 비교: 교집합 확인
common = len(set(outlier_z.index) & set(outlier_iqr.index))
print(f"Common Outliers: {common}")
```

VISUAL COMPARISON (VENN DIAGRAM)



결과 해석

- IQR이 더 많은 이상치를 탐지함 (116개):** Titanic Fare 데이터처럼 오른쪽으로 길게 늘어진(Skewed) 분포에서는 Z-Score보다 IQR이 더 민감하게 반응합니다.
- 완전한 포함 관계 (Subset):** Z-Score로 탐지된 20개는 모두 IQR 탐지 결과에 포함됩니다. 이 20개의 교집합은 '확실한' 이상치 후보입니다.

처리 전략 1: 삭제(Dropping)

■ 적용 조건: 명백한 오기입(Error)이거나,
전체 데이터 대비 비중이 매우 적을 때

▲ 주요 위험: 무조건적인 삭제는
정보 손실과 모델의 편향(Bias)을 초래함

❑ 실무 원칙: 삭제 버튼을 누르기 전에
반드시 도메인 지식으로 이유를 검증할 것

👉 잠깐! 지우기 전에 확인하세요

"단순히 튀는 값이라고 해서 지우면 안 됩니다."

✓ 나이 200세 → **오류 (삭제 OK)**

✗ 연봉 100억 → **특이값 (삭제 보류)**

⚠ DATA LOSS WARNING



데이터는 한 번 지우면
돌아오지 않습니다.

Dropping 판단 체크리스트



명백한 허먼 에러인가?

예: 키 -180cm, 나이 999세



데이터가 충분한가?

삭제해도 통계적 유의성 유지 가능



패턴이 있는 이상치인가?

특정 시간대/그룹에서만 발생한다면 보존

처리 전략 2: 원저라이징 (Winsorizing)

✓ 개념 (Capping): 이상치를 제거하지 않고,
상한(Upper) 또는 하한(Lower) 값으로 대체

- ✓ 핵심 장점:
- 데이터 개수(Sample Size) 유지
 - 극단값에 의한 평균/분산 왜곡 완화

● 초보자를 위한 비유: 과자 봉지

"과자 봉지가 너무 부풀어 박스에 안 들어갈 때,
내용물(과자)은 버리지 않고 공기만 빼서
부피를 줄여 꼭 눌러 담는 것과 같습니다."
→ 극단적으로 튀어나온 값을 경계선 안쪽으로 꼭 누름

전후 분포 변화 비교

Winsorizing 적용 시 꼬리(Tail)의 변화



처리 전략 3 로그 변환 (Log Transform)

※ 목적 및 개념

값의 스케일(Scale)을 대폭 줄여 **우측 꼬리가 긴 분포**를 정규분포에 가깝게 변환

▣ 주요 활용 데이터

연봉 집값 매출액 등

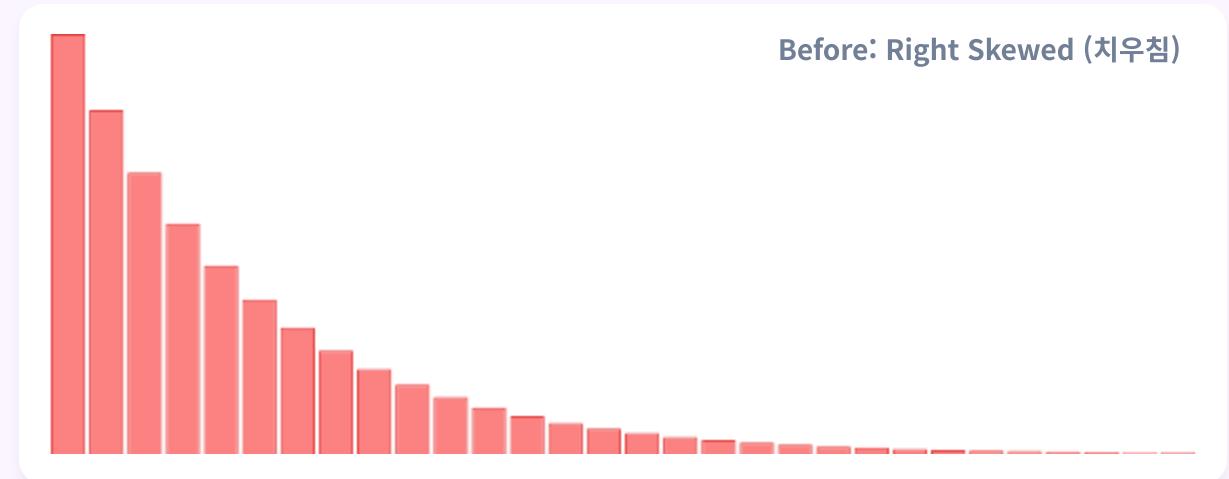
단위가 크고 양의 값을 가지는 금융/경제 데이터

⚠ 수학적 주의사항

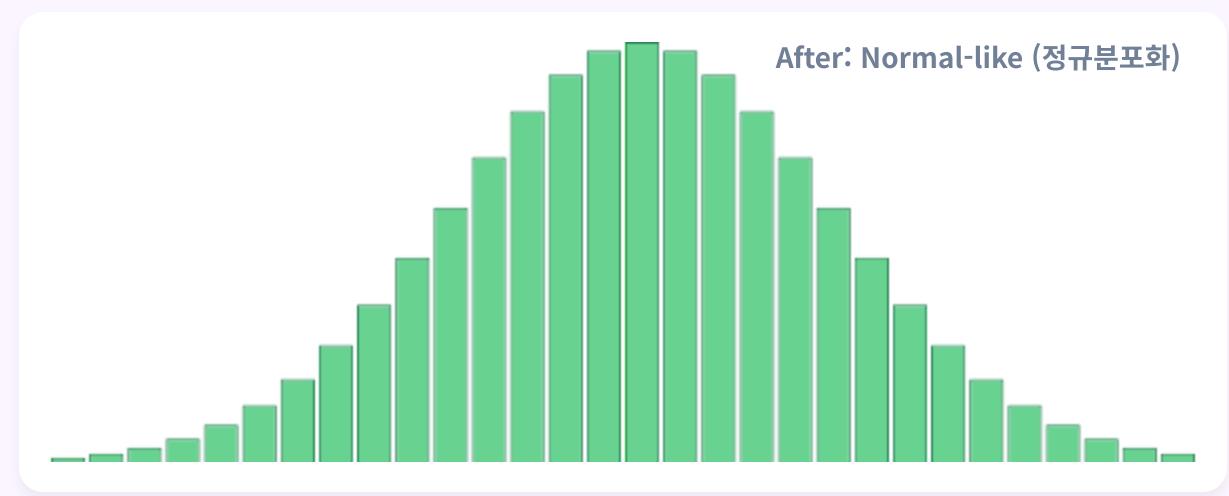
0 또는 음수가 포함된 데이터라면?

$\text{np.log}(0) = -\text{Inf}$ (에러 발생)

✓ 해결책: `np.log1p(x)` 사용 ($\log(x+1)$ 효과)



↓ Log Transformation (`np.log1p`)



금융 데이터 실습: FDS 이상 거래 탐지

신용카드 사기(Fraud) 의심 거래를 찾아내기 위한 데이터 처리 파이프라인



데이터 탐색

히스토그램과 박스플롯으로
데이터 분포 확인.
"얼마나 꼬리가 긴가?"



기준 선정

비대칭 분포에 강한 IQR
방식을 사용하여 이상치
경계선(Fence) 설정.



전처리 수행

극단값을 상한선으로 누르는
원저라이징 혹은 로그 변환
적용.



해석 및 검증

처리 전후의 평균/분산 변화를
비교하고, 실제 사기 데이터와
대조.

Why Financial Data?

- 금융 데이터(금액)는 전형적인 양의 왜도(Positive Skewness)를 가집니다.
- 대부분 소액 결제지만, 소수의 고액 결제가 평균을 크게 흔들기 때문에 이상치 처리가 필수적입니다.

FDS 실습 Step 1-2: 분포 확인과 IQR 추출



금융 데이터의 극단적 치우침 확인 및 통계적 이상치 필터링

fds_outlier_detection.py

```
import pandas as pd
import seaborn as sns

# 1. 금융 거래 데이터 로드 (가상 데이터)
df = pd.read_csv('credit_card_transactions.csv')

# 2. 분포 시각화 (Boxplot)
sns.boxplot(x=df['Amount'])
plt.title('Transaction Amount Distribution')

# 3. IQR 방식 이상치 인덱스 추출
Q1 = df['Amount'].quantile(0.25)
Q3 = df['Amount'].quantile(0.75)
IQR = Q3 - Q1

# Tukey Fences 설정
lower_fence = Q1 - 1.5 * IQR
upper_fence = Q3 + 1.5 * IQR

# 이상치(Outlier) 조건 필터링
outliers = df[(df['Amount'] < lower_fence) | (df['Amount'] > upper_fence)]

print(f"이상치 개수: {len(outliers)}건")
print(f"Upper Fence 기준: {upper_fence:.2f}")
```

EXECUTION RESULT: Amount Distribution



분석 인사이트

- 극단적 치우침(Extreme Skewness):** 대부분의 거래는 100달러 미만이지만, 일부 거래는 수천 달러에 육박함.
- IQR의 강건함(Robustness):** 평균(Mean)은 극단값에 의해 왜곡되지만, IQR은 중앙 50%를 기준으로 하므로 안정적인 기준선(Upper Fence)을 제공함.

FDS 실습: 처리 전략 적용 (Step 3-4)



원저라이징(Winsorizing)과 로그 변환(Log Transform)으로 데이터 다듬기

fds_outlier_treatment.py

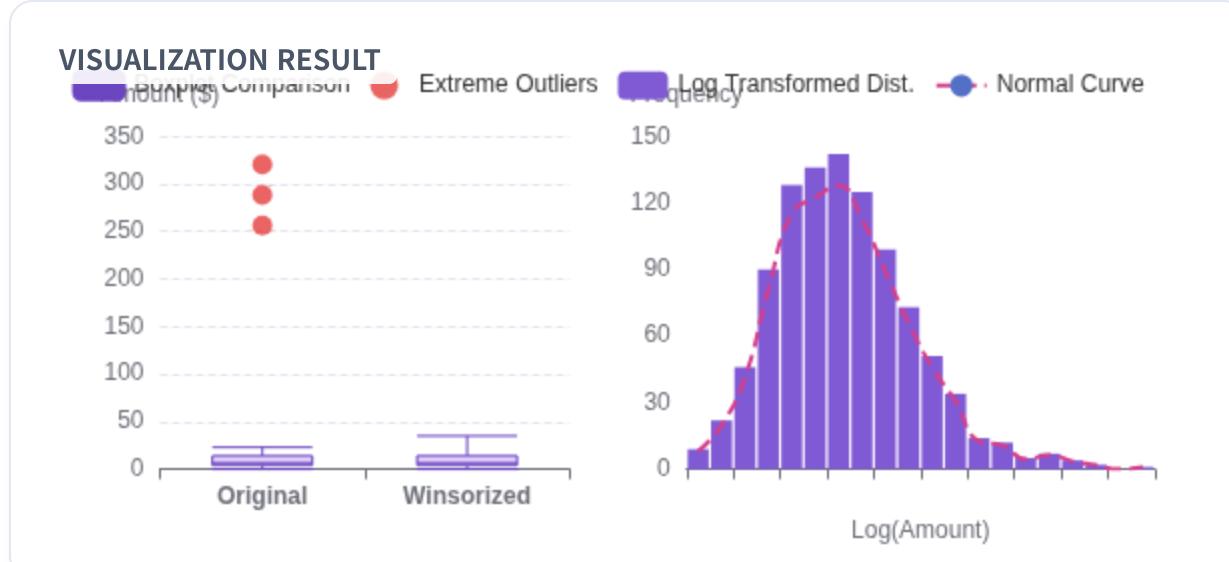
```
import numpy as np
from scipy.stats.mstats import winsorize

# [Step 3] 원저라이징: 상위 5% 값 캡핑(Capping)
# 데이터를 삭제하지 않고 경계값으로 눌러 담음
df['amt_winsor'] = winsorize(df['Amount'], limits=[0, 0.05])

# 처리 전후 통계량 비교 (평균과 표준편차 변화)
print("Original Mean:", df['Amount'].mean())
print("Winsor Mean :", df['amt_winsor'].mean())
# -> 극단적 이상치 영향이 줄어 평균이 안정됨

# [Step 4] 로그 변환: 분포 펴기
# 0을 처리하기 위해 log(x+1) 사용
df['amt_log'] = np.log1p(df['Amount'])

# 시각화: Right Skewed -> Bell Shape 근접
sns.histplot(df['amt_log'], kde=True)
plt.title("Log Transformed Distribution")
```



✓ 처리 효과 해석

- 원저라이징(좌측):** 극단적으로 튀던 이상치들이 상한선(Upper Fence) 위치로 '압축'되었습니다. 데이터 손실 없이 분포의 길이를 줄였습니다.
- 로그 변환(우측):** 한쪽으로 쏠려있던 금액 데이터가 '종 모양(Bell Shape)'에 가깝게 변했습니다. 이제 머신러닝 모델이 패턴을 더 잘 학습할 수 있습니다.

퀴즈와 선택 가이드

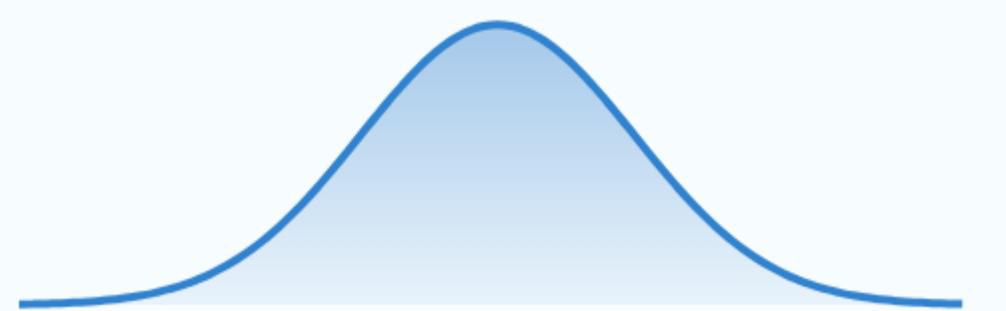
"어떤 상황에서 어떤 기준을 적용해야 할까요? 데이터의 형태가 정답을 알려줍니다."



Z-Score 선택

정규분포 가정

빠른 판단



- ✓ 데이터가 좌우 대칭인 종 모양(Bell Shape)일 때
- ✓ 평균(Mean)이 데이터의 중심을 잘 대표할 때
- ✓ 빠르고 표준화된 기준(예: 3σ)이 필요할 때

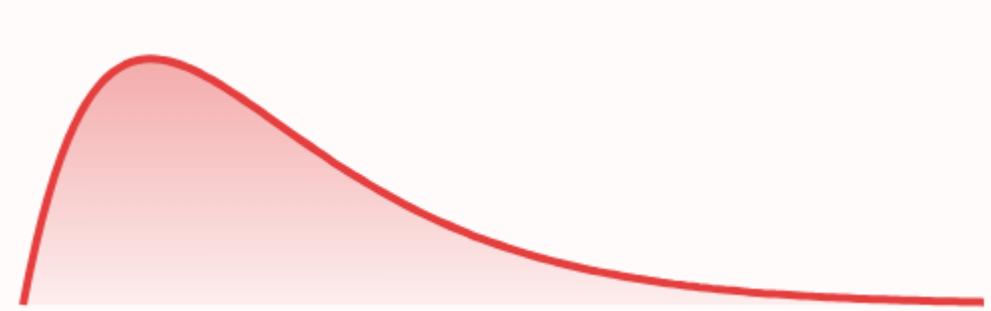


IQR 방식 선택

비모수적 방법

강건성(Robust)

OR



- ✓ 데이터가 한쪽으로 치우친(Skewed) 분포일 때
- ✓ 극단적인 값 때문에 평균이 신뢰할 수 없을 때
- ✓ 이상치에 영향을 덜 받는 강건한 기준이 필요할 때



실무 Tip: 지우지 말고 활용하세요!

이상치라고 무조건 삭제하면 정보 손실이 발생합니다. `is_outlier` 컬럼을 만들어 태깅(Tagging)한 후, 모델의 입력 변수로 활용하면 예측 성능이 오히려 향상될 수 있습니다.

핵심 요약 및 정리

오늘 배운 이상치 탐지 프로세스를 한 눈에 복습합니다.



1. 시각적 탐지

- ✓ **Histogram:** 분포의 꼬리 확인
- ✓ **Boxplot:** 벗어난 점 식별
- ✓ 의심 구간 파악이 최우선



2. 통계적 기준

- ✓ **Z-Score:** 정규분포일 때 ($|z| > 3$)
- ✓ **IQR:** 치우친 분포일 때
- ✓ 데이터 모양에 맞춰 선택



3. 처리 전략

- ✓ **삭제:** 확실한 오류일 때만
- ✓ **원저라이징:** 상/하한으로 대체
- ✓ **로그변환:** 스케일 압축



4. 해석 및 검증

- ✓ 처리 전후 통계량 비교
- ✓ 도메인 지식 기반 해석
- ✓ 비즈니스 임팩트 고려



Action Item

가지고 있는 데이터셋으로 **describe()**를 실행하고,
Boxplot을 그려서 나만의 이상치를 찾아보세요!