이상치 탐지 및 처리

이상치 탐지 방법

- **시각적 방법**: Box Plot, Scatter Plot 등을 통해 시각적으로 다른 데이터들과 동떨어진 값을 확인합니다.
- **통계적 방법**:
  - **Z-score**: 데이터가 정규분포를 따른다고 가정하고, 평균으로부터 표준편차의 몇 배만큼 떨어져 있는지를 측정합니다. 보통 Z-score가 ±2 또는 ±3을 벗어나면 이상치로 간주합니다.
  - **IQR (Interquartile Range)**: 데이터의 사분위 범위를 이용하여 정상 범위를 정의하고, 그 범위를 벗어나는 값을 이상치로 판단합니다. 데이터가 정규분포를 따르지 않을 때 더 효과적인 방법입니다.

이상치 처리 방법

- **제거(Deletion)**: 이상치로 판단된 데이터를 제거합니다.
  - 이상치가 명백한 측정 오류나 입력 오류이고, 그 수가 적을 때 가장 간단한 방법입니다.
  - 하지만 실제 의미 있는 희귀 데이터일 수 있으므로 신중해야 합니다.
- **대체(Replacement)**: 이상치를 상한/하한 값, 평균, 중앙값 등 다른 값으로 대체합니다.
  - 데이터 손실을 피하고 싶을 때 사용합니다.
  - 경계값으로 대체(Capping)하는 것이 일반적이며, 평균/중앙값 대체는 데이터의 분산을 줄일 수 있어 주의가 필요합니다.
- **변환(Transformation)**: 로그 변환 등을 통해 데이터 전체의 분포를 변경하여 이상치의 영향을 줄입니다.
- **분류 및 분석**: 이상치 자체가 중요한 정보일 경우(사기 탐지 등), 이를 별도로 분류하여 원인을 분석하거나, 이상치 탐지 모델링의 타겟으로 삼습니다.

적용 가능한 상황

- **데이터 품질 개선**: 데이터 입력 오류로 인해 발생한 비정상적인 값을 찾아내고 수정할 때.
- **모델 성능 향상**: 회귀 분석이나 평균 기반의 알고리즘에서 이상치가 모델의 예측을 크게 왜곡시키는 것을 방지하기 위해.
- **이상 현상 탐지**: 신용카드 사기 탐지, 시스템 침입 탐지 등 정상 범주에서 벗어나는 이례적인 패턴을 찾아내는 것이 분석의 주목표일 때. 이 경우 이상치는 제거 대상이 아닌 분석의 핵심 대상이 됩니다.

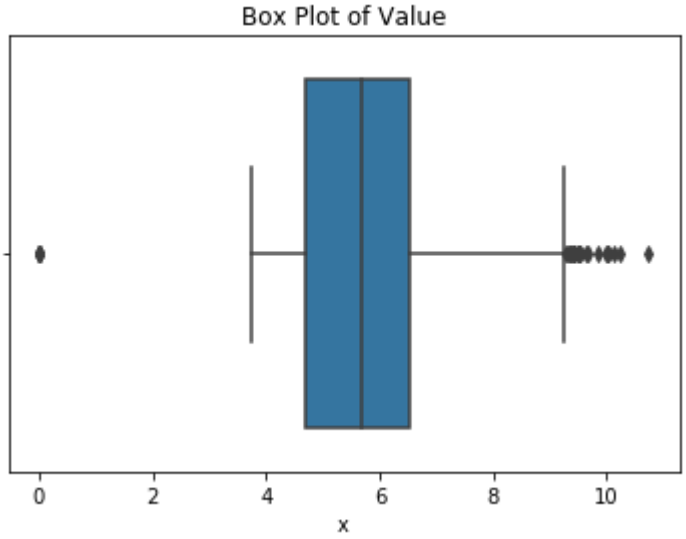
예제 데이터

```
import pandas as pd
import numpy as np

df = pd.read_csv("dataset/diamonds.csv")
```

1. 시각적 탐지 (Box Plot)

- **용도**: 데이터의 분포와 이상치를 시각적으로 빠르게 확인합니다.
- **주의사항**: `matplotlib` 또는 `seaborn` 라이브러리가 필요합니다.
- **코드 예시**

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 4))
sns.boxplot(x=df['x'])
plt.title('Box Plot of Value')
plt.show()
```



- **결과 해석**
  - 박스 플롯은 데이터의 사분위수를 시각화합니다.
  - 박스(box): IQR(Q1 ~ Q3) 범위
    - 박스 하단: 제1사분위수(Q1, 25%)
    - 박스의 선: 중앙값(Median, Q2, 50%)
    - 박스 상단: 제3사분위수(Q3, 75%)
  - 수염(whisker): Q1-1.5×IQR, Q3+1.5×IQR 범위 내의 최소·최대값
  - 이상치: 수염(whisker)을 벗어나는 점들

2. Z-score 기반 탐지 및 처리

- **용도**: 데이터가 정규분포에 가까울 때 이상치를 탐지합니다.
- **주의사항**: 평균과 표준편차 자체가 이상치에 의해 영향을 받기 때문에, 이상치가 매우 극단적일 경우 Z-score 방법의 탐지 성능이 저하될 수 있습니다(Masking effect).
- **코드 예시**

```
from scipy import stats

# Z-score 계산
df['z_score'] = stats.zscore(df['x'])

# Z-score의 절대값이 5 초과하는 경우를 이상치로 정의
threshold = 5
outliers_z = df[np.abs(df['z_score']) > threshold]
print("--- Outliers by Z-score ---")
```

```
print(outliers_z)

# 이상치 제거
df_no_outliers_z = df[np.abs(df['z_score']) <= threshold]
```

- **결과 해석**: `z_score` 열이 추가되고, 그 절대값이 5를 초과하는 행이 이상치로 식별됩니다. 이 조건에 해당하지 않는 데이터만 필터링하여 이상치를 제거할 수 있습니다.

3. IQR 기반 탐지 및 처리

- **용도**: 분포에 대한 가정이 필요 없어 Z-score보다 더 일반적으로 사용되는 이상치 탐지 방법입니다.
- **주의사항**: IQR의 배수(보통 1.5)는 분석가의 주관에 따라 조정될 수 있습니다. (e.g., 더 엄격하게 보려면 1.0, 더 관대하게 보려면 2.0)
- **코드 예시**

```
# 1사분위수(Q1)와 3사분위수(Q3) 계산
Q1 = df['x'].quantile(0.25)
Q3 = df['x'].quantile(0.75)
IQR = Q3 - Q1

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

print(f"Q1: {Q1}, Q3: {Q3}, IQR: {IQR}")
# Q1: 4.71, Q3: 6.54, IQR: 1.83
print(f"Lower Bound: {lower_bound}, Upper Bound: {upper_bound}")
# Lower Bound: 1.9649999999999999, Upper Bound: 9.285

# 이상치 탐지
outliers_iqr = df[(df['x'] < lower_bound) | (df['x'] > upper_bound)]
print("\n--- Outliers by IQR ---")
print(outliers_iqr)

# 이상치 대체 (Capping/Clipping)
df_capped = df.copy()
df_capped['x'] = np.clip(df['x'], lower_bound, upper_bound)
print("\n--- Capped DataFrame ---")
print(df_capped)
```

- **결과 해석**
  - Q1, Q3, IQR 값이 계산되고, 이를 바탕으로 정상 데이터의 범위(lower_bound ~ upper_bound)를 결정
  - 이 범위를 벗어나는 값들이 이상치로 식별
  - `np.clip` 함수를 사용하여 이상치를 경계값(lower_bound, upper_bound)으로 대체(Capping)
    - 데이터의 정보 손실을 최소화 / 이상치의 극단적인 영향을 줄이기

장단점 및 대안

방법	장점	단점	대안
Box Plot (시각화)	데이터 분포와 이상치를 직관적으로 빠르게 파악 가능.	다변량 데이터의 관계 속에서 나타나는 이상치는 탐지하기 어려움.	**Scatter Plot**: 두 변수 간의 관계에서 벗어나는 이상치를 탐지하는 데 유용.
Z-score	구현이 간단하고 통계적 의미가 명확함.	데이터가 정규분포를 따른다는 가정이 필요함. 평균과 표준편차가 이상치에 민감함(Robust하지 않음).	**Modified Z-score**: 평균 대신 중앙값(median), 표준편차 대신 MAD(Median Absolute Deviation)를 사용하여 이상치에 덜 민감하게 만듦.
IQR	분포에 대한 가정이 필요 없어 Z-score보다 강건함(Robust).	IQR의 배수(1.5)를 결정하는 데 주관이 개입될 수 있음.	**DBSCAN**: 밀도 기반 클러스터링 알고리즘으로, 어떤 군집에도 속하지 않는 노이즈 포인트를 이상치로 탐지할 수 있음. 복잡한 분포의 이상치 탐지에 효과적. **Isolation Forest, Local Outlier Factor (LOF)**: 머신러닝 기반의 이상치 탐지 알고리즘으로, 더 복잡하고 다차원적인 데이터에서 이상치를 효과적으로 찾아냄.