Data Analysis Report - Outliers

Dataset 1: New York City Hotels (AB_NYC_2019.csv)

This dataset includes information on hotels in New York City along with their nightly rates and other relevant attributes. This analysis focuses on outlier detection and removal methods for the price column.

1. Data Overview

- **Dataset Name**: AB_NYC_2019.csv
- **Record Count**: 48,895
- **Column Count**: 16
- **Target Column**: price

2. Outlier Detection & Removal Techniques

2.1 Percentile-Based Technique

This technique identifies extreme values using specified percentiles as thresholds, allowing for the removal of outliers based on the distribution's tails.

- **Lower Percentile** (1%): $30.0
- **Upper Percentile** (99.9%): $3,000.0

Code Snippet:

python

Copy code

```
lower_percentile = Data.price.quantile(0.01)

upper_percentile = Data.price.quantile(0.999)

new_Data_without_outlier = Data[(Data['price'] > lower_percentile) & (Data['price'] < upper_percentile)]

outliers_Data = Data[(Data['price'] < lower_percentile) | (Data['price'] > upper_percentile)]
```

- **Data Reduction**: From 48,895 records to 48,183 records.
- **Outliers Removed**: 449 rows.

Observations:

- Hotels priced below $30 or above $3,000 were flagged as outliers.

2.2 Interquartile Range (IQR) Method

The IQR method, which measures the spread of the middle 50% of the data, was applied to further refine outlier detection.

- **Q1** (25th percentile): $69.0
- **Q3** (75th percentile): $175.0
- **IQR**: $106.0
- **Lower Limit**: Q1 - 1.5 * IQR = $-90.0
- **Upper Limit**: Q3 + 1.5 * IQR = $334.0

Code Snippet:

python

Copy code

```
Q1 = Data.price.quantile(0.25)

Q3 = Data.price.quantile(0.75)

IQR = Q3 - Q1

lower_limit = Q1 - 1.5 * IQR

upper_limit = Q3 + 1.5 * IQR

new_Data_without_outlierIQR = Data[(Data.price > lower_limit) & (Data.price < upper_limit)]
```

- **Data Reduction**: From 48,895 records to 45,918 records.

Observations:

- This method proved effective in capturing moderate outliers in the dataset.

2.3 Z-Score Technique

Z-Score measures the number of standard deviations a value is from the mean. This technique is useful for normally distributed data.

Threshold: |Z-score| > 3

Code Snippet:

python

Copy code

```
from scipy.stats import zscore

Data['Zscore'] = zscore(Data.price)

new_without_outlier = Data[(Data['Zscore'] > -3) & (Data['Zscore'] < 3)]
```

new_without_outlier2 = new_without_outlier.drop(columns=['Zscore'])

- **Data Reduction**: Not specified due to Z-score application on filtered normal data.

Observations:

- Z-score is effective in symmetric, normally distributed data but may require alternative thresholds for heavily skewed data.

Dataset 2: Weight-Height Analysis (weight-height.csv)

This dataset contains height and weight measurements for individuals. We focus on detecting and removing height outliers based on a normal distribution assumption.

1. Data Overview

- **Dataset Name**: weight-height.csv
- **Record Count**: 10,000
- **Column Count**: 3
- **Target Column**: Height

2. Outlier Detection - Standard Deviation Technique

2.1 Mean and Standard Deviation Calculation

The mean and standard deviation were calculated for Height.

- **Mean**: 66.37
- **Standard Deviation**: 3.85

2.2 Outlier Thresholds

Using three standard deviations from the mean to define outlier limits:

- **Lower Limit**: Mean - 3 * Standard Deviation = 54.82
- **Upper Limit**: Mean + 3 * Standard Deviation = 77.92

Code Snippet:

python

Copy code

mean = Data2.Height.mean()

std_dev = Data2.Height.std()

lower_limit = mean - 3 * std_dev

upper_limit = mean + 3 * std_dev

Data2_without_outlier = Data2[(Data2.Height > lower_limit) & (Data2.Height < upper_limit)]

- **Data Reduction**: From 10,000 records to 9,993 records.

Observations:

- This method is effective for normally distributed data and removed extreme height values likely due to measurement or data entry errors.

---

3. Summary of Findings

- **Percentile-Based Approach**: Effective for datasets with a high variance where extreme outliers could distort analysis.
- **IQR Method**: Useful for skewed data with mild to moderate outliers.
- **Z-Score**: Effective for normally distributed datasets but may need modification for skewed data.
- **Standard Deviation**: Works well on symmetric distributions and detected valid extreme measurements in height.

4. Visualizations

For better insights, consider visualizations such as box plots or histograms for price and height distributions before and after outlier removal. Here's a sample code for a histogram:

python

Copy code

import matplotlib.pyplot as plt

plt.hist(Data2_without_outlier.Height, bins=25, rwidth=0.8)

plt.title("Height Distribution after Outlier Removal")

plt.xlabel("Height")

plt.ylabel("Frequency")

plt.show()

5. Future Steps

- Further refine outlier detection thresholds based on business insights.
- Extend analysis to additional features for multidimensional outlier detection.