Handling Outliers using Python:

Colab:

https://colab.research.google.com/drive/1_c5bDS0FLPO2-fN-meGhugeL93ULI76H#scrollTo=kfZvbhnjz-8C

Outliers are data points that significantly differ from other observations in a dataset. They can affect statistical analysis and machine learning models, so handling them properly is important.

Common methods to handle outliers in Python:

1. **Detection using IQR (Interquartile Range)**
   - Calculate Q1, Q3, and IQR.
   - Define lower and upper fences using Q1 - 1.5×IQR and Q3 + 1.5×IQR.
   - Values outside these limits are treated as outliers.
2. **Detection using Z-Score**
   - Measures how many standard deviations a value is from the mean.
   - Typically, values with a Z-score greater than ±3 are considered outliers.
3. **Removing Outliers**
   - Outliers can be removed if they are due to errors or noise.
   - This is done by filtering data within acceptable limits.
4. **Capping (Winsorization)**
   - Outliers are replaced with the nearest boundary values instead of removing them.
   - Useful when data loss is undesirable.
5. **Transformation**
   - Applying transformations like logarithmic or square root can reduce the impact of outliers.

Short Notes on IQR (Interquartile Range)

Definition:
The Interquartile Range (IQR) is the range of the middle 50% of a dataset. It measures statistical dispersion and is calculated as:

$$\text{IQR} = Q3 - Q1$$

Where:

- **Q1** = First Quartile (25th percentile)
- **Q3** = Third Quartile (75th percentile)

Purpose:

- Measures the spread of the central portion of the data.
- Less sensitive to extreme values compared to range or standard deviation.
- Used for **outlier detection** using the formula:
    - Lower Fence = Q1 − 1.5 × IQR
    - Upper Fence = Q3 + 1.5 × IQR