Data Preprocessing

It prepares raw data into a clean and organized format suitable for further analysis or model building.

Importance of Data Preprocessing

- Ensures **data quality** and **consistency :** Removes errors, duplicates, inconsistencies, outliers, and handles missing values, making data clean, reliable, and consistent
- Improves **model performance** and **reliability :** Well-preprocessed data leads to more accurate and robust models by preventing issues like overfitting and underfitting. It also allows algorithms to better identify patterns and relationships
- **Makes data usable for algorithms:** Many machine learning algorithms require data to be in specific formats (such as normalized numbers, encoded categories) to function correctly and efficiently
- **Reduces computational complexity:** By removing irrelevant or redundant features (dimensionality reduction), preprocessing speeds up analysis and model training
- **Supports data integration and privacy:** Preprocessing helps merge data from different sources and can include anonymization or redaction steps to ensure compliance and confidentiality
- Prevents **biased** or **misleading** insights : Clean data results in more trustworthy analytics and decision-making, uncovering genuine business trends rather than artifacts of poor data

- What is data preprocessing and why is it crucial in data analysis or machine learning projects?
- Explain the phrase "garbage in, garbage out" in the context of data preprocessing.

Tasks in data preprocessing

- **Data Cleaning**: Detecting and correcting errors or inconsistencies, handling missing values, and removing duplicate records to ensure the quality and reliability of the data.
- **Data Transformation**: Converting data into suitable formats required for analysis.
- **Data Reduction**: Reducing data volume or dimensionality by removing irrelevant, redundant, or highly correlated features.
- **Data Integration**: Combining data from multiple sources into a coherent dataset, resolving discrepancies and ensuring consistency across different data sources.

Data Cleaning

1. Assess Data Quality

Before cleaning, evaluate:

1. **Completeness** – **Definition**: All required data is present and available.

   Are there missing values? Yes /No

Example Scenario:
A hospital maintains a database of patients. If many records are **missing information about blood pressure or medical history**, the dataset is **incomplete**, making it unreliable for diagnosing patterns or building a predictive model for heart diseases.

Impact: Incomplete data can lead to incorrect conclusions or biased models.

2. **Consistency** –Data is uniform and does not contradict across different sources or parts of the dataset.

   Are data formats and units uniform?

   Example Scenario:
   In a retail sales database, if one column says a product was sold on **“2024-06-01”**, but another related system logs the sale as **“2024-05-31”**, this is a **consistency issue**.
   ***Impact**: Inconsistent data may cause incorrect aggregation, reporting errors, or unreliable dashboards.*

3. **Accuracy** – Data correctly represents the real-world values or events.

   Are values valid and reasonable?

Example Scenario:
An e-commerce website records customer delivery addresses. If a customer's city is wrongly entered as **“Delihi” instead of “Delhi”**, the data is inaccurate. This could result in **failed deliveries** or **logistical issues**.

***Impact**: Inaccurate data leads to errors in analysis, misinformed decisions, and customer dissatisfaction.*

4. **Uniqueness** – No repeated records or duplicate entries exist in the dataset.

   Are there duplicate records?

Example Scenario:
In a student registration system, the same student is accidentally entered **twice** with slightly different spellings (“Aditi S.” and “Adithi S.”). The database will count them as **two separate students**, inflating enrollment numbers.

***Impact**: Duplicates cause inflated counts, biased statistics, and poor user experience.*

5. **Timeliness** – Data is up-to-date and relevant to the current time or task.

   Is the data up to date?

Example Scenario:
A bank uses customer financial data to offer credit cards. If it relies on **last year's income details**, it may **reject eligible customers** whose income has recently increased.

***Impact**: Outdated data results in lost business opportunities and poor decision-making.*

6. **Validity** - Data follows the correct format, type, and rules.

   Example Scenario:
   In a contact form, if a user enters "abcd" in the **Phone Number** field, the data is invalid. It should only accept **digits with a specific length**.

   ***Impact**: Invalid data leads to system errors, failed transactions, or unusable records.*

7. **Relevance** : Data is appropriate and useful for the analysis goal.

Example Scenario:
For predicting customer churn, collecting data on **weather conditions** is irrelevant. However, **user activity**, **subscription history**, and **support interactions** are highly relevant.

Impact: Irrelevant data increases processing time and can **reduce model performance** by introducing noise.

Dimension	Meaning	Example Scenario	Impact

Completeness	All necessary data is present	Missing patient history in medical data	Incomplete analysis
Accuracy	Data reflects the real-world truth	Wrong address in logistics	Failed deliveries
Consistency	No contradictions across data	Mismatched dates in two systems	Report errors
Uniqueness	No duplicates in data	Same customer listed twice	Inflated numbers
Timeliness	Data is up-to-date	Using last year's income	Wrong decisions
Validity	Data meets format/rules	Invalid phone number format	System issues
Relevance	Data fits the analysis goal	Collecting weather data for churn prediction	Wasted resources, low accuracy

Explain how different data quality dimensions affect the outcomes of data analysis.

2. Data Anomalies

Data anomalies are unusual, incorrect, inconsistent, or unexpected values in the dataset that can significantly affect the outcome of data analysis or model performance.

Types of data anomalies include:

1. **Missing values :** Values that are **not recorded** or are **blank/null** in one or more fields.

📓 **Scenario:**

In a student database, the "email" or "mobile number" fields are blank for 10% of the students. This is a **missing value anomaly**.

🔍 **Common Causes:**

- Manual entry errors
- Sensors or systems failing to capture data
- Optional fields not filled

⚠ **Impact:**

- Some machine learning models will **fail or throw errors** if missing values aren't handled.
- Analysis may be **biased** or **incomplete**.

2. **Outliers :** Data points that lie far outside the normal range of values.

📒 **Scenario:**

In a salary dataset of employees where the average salary is ₹50,000/month, one entry shows ₹5,000,000. This is likely an outlier (possibly a CEO, or a data error).

🔍 **Common Causes:**

- Entry errors (e.g., extra zero)
- Legitimate extreme cases (e.g., VIP customer)
- Different units used (e.g., dollars vs rupees)

⚠ **Impact:**

- Can distort means, variances, and correlations
- Mislead clustering or regression models

3. **Duplicated entries :** The same data record appears more than once in the dataset.

📕 **Scenario:**

In a COVID vaccination dataset, the same person is listed twice under slightly different spellings or ID numbers. Both records are treated as two different individuals.

🔍 **Common Causes:**

- Data entry from multiple sources
- No primary key or unique identifier enforcement
- Copy-paste errors

⚠ **Impact:**

- Inflated counts and incorrect analysis
- May lead to double billing or wrong targeting

4. Inconsistent formatting **:** Different formats or values used for the same entity or meaning.

📕 **Scenario:**

A column representing gender contains values like `Male`, `M`, `male`, and `m`. All refer to the same thing, but are inconsistently entered.

🔍 **Common Causes:**

- Lack of standardized data entry
- Data merged from multiple systems

⚠ **Impact:**

- Prevents correct grouping or filtering
- Data cleaning becomes harder
- May cause misclassification

5. Invalid Data (Violation of Rules/Formats) - Data that doesn't conform to expected types, ranges, or formats.

📓 **Scenario:**

- A "Date of Birth" column contains a value like "31-02-2024"
- A "Phone Number" has alphabetic characters

🔍 **Common Causes:**

- Lack of data validation
- User input errors

⚠ **Impact:**

- Software or pipelines **crash** or produce errors
- Leads to **inaccurate reports**

6. Noisy Data - Data with **random or meaningless variations**, making it difficult to detect patterns.

📓 **Scenario:**

Sensor readings from a temperature monitor show values like 32, 31.9, 200, 32.1. That value "200" is **noise**.

🔍 **Common Causes:**

- Sensor malfunction
- Background disturbances

- Random fluctuations

⚠ Impact:

- Hinders trend detection
- Reduces model accuracy
- Requires smoothing or filtering

7. Conflicting Data : Contradictory data points that cannot be true at the same time.

📰 Scenario:

In a bank database, one table shows a customer account as “closed,” but another still shows active transactions.

🔍 Common Causes:

- Poorly synchronized systems
- Partial updates or system failures

⚠ Impact:

- Confusion in decision-making
- Legal or compliance risks

Anomaly Type	Description	Example Scenario	Likely Impact
Missing Values	Null or empty fields	Missing email in student records	Incomplete or failed models
Outliers	Extreme, unusual values	₹5,000,000 salary in ₹50,000 avg group	Distorted statistics
Duplicates	Same record repeated	Two identical vaccination entries	Overcounting
Inconsistent Data	Same info in multiple formats	"Male", "M", "m" in gender field	Inaccurate grouping
Invalid Data	Doesn't meet format/type rules	"abcd" in phone number	System errors
Noisy Data	Random errors in data	Temperature sensor shows 200°C briefly	Misleading trends
Conflicting Data	Contradictory information	Account marked closed but active transactions	Operational errors

How to Deal with Data Anomalies

1 Missing Values

There are three types:

1. **MCAR** (Missing Completely At Random)
2. **MAR** (Missing At Random)
3. **MNAR** (Missing Not At Random)

🔧 **Handling Techniques:**

1. **Remove rows/columns** with too many missing values

```
df.dropna()  # drop rows
df.dropna(axis=1)  # drop columns
```

2. **Impute missing values** using:

- Mean/Median/Mode
- K-Nearest Neighbors (KNN)

```
df [‘column_name].fillna(df [column_name].mean(), inplace=True)
```

Detecting Missing Values with Pandas

❖ Use .info(): Shows non-null counts and data types.

❖ Use .isna() or .isnull(): Detects missing values (NaNs) in the DataFrame.

❖ Use .describe(): May reveal anomalies when counts differ across variables.

Diagnosing Types of Missing Values

❖ **Statistical and Visual Techniques:**

- **Heatmap of missing data** using seaborn:

```
import seaborn as sns

sns.heatmap(df.isna(), cbar=False)
```

Approaches to Deal with Missing Values

Handling missing data is a critical part of data preprocessing. The method chosen depends on the nature, volume, and impact of the missing data.

1 Keep the Missing Values As Is

- **When to Use**:
  - The model or algorithm can **handle missing values internally** (e.g., Decision Trees, XGBoost).
  - The **missingness itself is meaningful** (e.g., unanswered survey question = user behavior).
- **Advantages**:
  - Preserves all original data.
  - Useful for **unsupervised learning or clustering**, where missingness may form a pattern.
- **Disadvantages**:
  - Many ML algorithms (e.g., logistic regression, KNN, SVM) **do not support nulls**.
  - Can **bias analysis** if not properly accounted for.

---

2 Remove Data Objects (Rows) with Missing Values

- **How**:

```
df.dropna(inplace=True)
```

- **When to Use**:
  - When the number of missing rows is **small**.
  - When data is **missing at random (MCAR)** and won't introduce bias.
- **Advantages**:

- Simple and quick.
  - Avoids making incorrect assumptions about the data.
- **Disadvantages**:
  - Risk of **data loss**.
  - Can **bias** the dataset if missingness is not random.

---

3 Remove Attributes (Columns) with Missing Values

- **How**:

```
df.dropna(axis=1, inplace=True)
```

- **When to Use**:
  - When **a column has a high percentage of missing values** (e.g., >50%).
  - When the column is **not critical** for the model or analysis.
- **Advantages**:
  - Simplifies dataset.
  - Useful when features are **irrelevant or redundant**.
- **Disadvantages**:
  - Loss of **potentially useful information**.
  - May affect model performance if important features are dropped.

---

4 Estimate and Impute Missing Values

Replace missing values using statistical or machine learning techniques.

Common Imputation Techniques:

Method	Description	When to Use

Mean/Median/Mode Imputation	Replace missing values with mean/median/mode of the column	For numerical or categorical data with low missing rate
KNN Imputation	Use similarity with nearest neighbors to fill values	When data is dense and relationships are nonlinear
Regression Imputation	Predict missing values using other columns	When there are strong correlations
Interpolation	Fill values based on trends (linear, time-series)	For time-series data
MICE (Multivariate Imputation by Chained Equations)	Iterative model-based imputation	When multiple variables are missing

- **Pandas Example**:

```
df['age'].fillna (df['age'].mean(), inplace=True)  #  mean imputation
```

- **Advanced Example with sklearn**:

```
from sklearn.impute import KNNImputer
imputer = KNNImputer (n_neighbors=3)
df_imputed = pd.DataFrame(imputer.fit_transform(df), columns=df.columns)
```

- **Advantages**:
  - Preserves data size.
  - Can maintain statistical integrity if done correctly.
- **Disadvantages**:
  - Risk of introducing **bias** or incorrect patterns.
  - Requires **assumptions** about the data.

✅ Summary Table

Approach	Use When	Pros	Cons
Keep as is	Missingness is informative or model supports it	Preserves all data	May lead to model issues
Remove rows	Missingness is random and small	Simple and clean	Data loss
Remove columns	Feature not important and many values missing	Simplifies model	Potential loss of info
Imputation	Data is missing at random (MAR) or limited	Keeps structure	May introduce bias

---

✅ 1. Get All Rows with Any Missing Values

python

```python
missing_rows = df[df.isna().any(axis=1)]
print(missing_rows.index)
```

🔍 Explanation:

- `df.isna()` returns a boolean DataFrame.
- `.any(axis=1)` checks if **any column in a row is NaN**.
- The result is a filtered DataFrame containing only rows with at least one missing value.
- `.index` gives the row indices.

✅ 2. Get Rows with Missing Values in a Specific Column

```python
missing_in_col = df[df['column_name'].isna()]
print(missing_in_col.index)
```

Replace `'column_name'` with the name of the column you're checking.

✅ 3. Get the Row Numbers as a List

```python
missing_indices = df[df.isna().any(axis=1)].index.tolist()
print(missing_indices)
```

This returns the row indices **as a list**, which is useful for iteration or reporting.

✅ 4. Get Boolean Mask for Missing Rows

```python
mask = df.isna().any(axis=1)
```

This returns a **boolean Series** that can be used to filter or analyze further.

Outliers

Outliers are extreme values that differ significantly from the rest of the data. Detecting them is crucial to avoid skewed analysis, inaccurate models, and misleading conclusions.

1 Univariate Outlier Detection

Analyzing **one variable** at a time to detect extreme values.

🔍 Common Techniques:

- **Boxplot (IQR method)**
- **Z-score or Modified Z-score**
- **Histogram or Density plot**

📓 Example:

Detect unusually high temperatures recorded by a sensor.

📌 IQR Method (Interquartile Range):

```python
Q1 = df['temperature'].quantile(0.25)
Q3 = df['temperature'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = df[(df['temperature'] < lower_bound) | (df['temperature'] > upper_bound)]
```

📌 Z-score Method:

```python
from scipy.stats import zscore
df['z_score'] = zscore(df['temperature'])
outliers = df[df['z_score'].abs() > 3]
```

🟢 Use When:

- You want to check **distribution-based** outliers for a **single feature** (e.g., height, income, marks)

2 Bivariate Outlier Detection

Analyzing outliers based on the **relationship between two variables**.

🔍 Common Techniques:

- **Scatter plots**
- **Mahalanobis distance**
- **Correlation + outlier isolation**

📘 Example:

A student has **high study hours but very low marks** — indicating something unusual.

📌 Scatter Plot Approach:

```python
import seaborn as sns
sns.scatterplot(x='study_hours', y='marks', data=df)
```

3 Time Series Outlier Detection

Detecting points in a time series that deviate from the **temporal pattern**.

🔍 Common Techniques:

- **Rolling statistics + z-score**

📘 Example:

Sudden spike/drop in daily website traffic, temperature, or electricity usage.

📌 **Rolling Z-Score Method:**

```python
df['rolling_mean'] = df['value'].rolling(window=7).mean()
df['rolling_std'] = df['value'].rolling(window=7).std()
df['z_score'] = (df['value'] - df['rolling_mean']) / df['rolling_std']
outliers = df[df['z_score'].abs() > 2]
```

📌 What is a Z-Score?

A **Z-score** tells us **how far** a data point is from the **mean** (average) of a dataset, measured in **standard deviations**.

🔢 Z-Score Formula:

$$\text{Z-score} = \frac{(x - \mu)}{\sigma}$$

Where:

- x = value of the data point
- μ = mean of the dataset
- σ = standard deviation of the dataset

Z-Score	Interpretation
0	The data point is exactly at the mean
+1	1 standard deviation above the mean
-1	1 standard deviation below the mean
+2/+3	Much higher than the average (possible outlier)
-2/-3	Much lower than the average (possible outlier)

Dealing with Outliers

1 Do Nothing

✅ **When to Use:**

- Outliers are **valid values** that carry important information (e.g., VIP customers, rare diseases).
- You're using **models robust to outliers** (e.g., decision trees, random forests).
- Outlier is **not affecting the distribution or performance** significantly.

📓 **Example:**

A bank dataset includes one billionaire customer. This extreme income value is real and important for segmentation.

Pros:

- Preserves full dataset
- Captures rare but real cases

● **Cons:**

- May skew mean, variance, or model predictions (e.g., linear regression)

---

2 Replace with the Upper Cap or Lower Cap (Winsorization)

✅ **When to Use:**

- Outliers are **extreme but not errors**
- You want to reduce their **impact without removing data**

📘 **Example:**

In a dataset of student scores, a few scores above 100 (possibly bonus points) are capped at 100.

```
import numpy as np

Q1 = df['score'].quantile(0.25)
Q3 = df['score'].quantile(0.75)
IQR = Q3 - Q1

lower_limit = Q1 - 1.5 * IQR
upper_limit = Q3 + 1.5 * IQR

df['score'] = np.where(df['score'] > upper_limit, upper_limit,
              np.where(df['score'] < lower_limit, lower_limit, df['score']))
```

Pros:

- Keeps all data points
- Limits influence of outliers

● **Cons:**

- May distort values slightly
- Not suitable for categorical outliers

3 Perform a Log Transformation

✓ **When to Use:**

- Data is **right-skewed** due to positive outliers (e.g., income, sales)
- You want to **normalize distribution** for linear models or visualization

📘 **Example:**

Sales data with a few large orders ($10,000+) is log-transformed to reduce skew.

df['log_sales'] = np.log1p(df['sales']) # log(1 + x) handles 0s

□ **Pros:**

- Reduces skewness and compresses outliers
- Works well for **positive, continuous data**

● **Cons:**

- Cannot handle zero or negative values without adjustment
- Transformed data is less interpretable

4 Remove Data Objects with Outliers

✅ **When to Use:**

- Outliers are **clearly due to error** (e.g., typing mistake: 9999 kg weight)
- Outliers are **statistically extreme** and not useful for analysis
- You're confident removal won't **bias** the dataset

📓 **Example:**

Temperature sensors show -273°C — clearly an error (absolute zero).

```
from scipy.stats import zscore

df['z_score'] = zscore(df['temperature'])
df = df[df['z_score'].abs() < 3]
```

Pros:

- Clean, interpretable dataset
- Avoids skewing of statistical models

- **Cons:**

- Potential **loss of information**
- Can introduce **bias** if not random

Strategy	When to Use	Pros	Cons
Do Nothing	Outliers are valid or important	Retains true variation	May affect model accuracy
Cap with Upper/Lower Bound	Outliers are real but extreme	Reduces impact without deletion	Slight distortion of data
Log Transformation	Skewed data, positive numeric values	Normalizes data, reduces skew	Harder to interpret, needs > 0
Remove Outliers	Obvious error or extreme invalid case	Clean data, robust stats	Loss of data, risk of bias

Suppose you are given a dataset with a lot of missing values and duplicate rows. What preprocessing steps would you apply before using this dataset for modeling?

1. Which of the following is not a data quality dimension?

A. Accuracy
B. Timeliness
C. Noise
D. Validity
✓ **Answer:** C. Noise

---

2. Which method is suitable when missing data is Missing Completely At Random (MCAR) and the proportion is small?

A. Keep missing as is
B. Remove rows with missing values
C. Remove columns
D. Regression imputation
✓ **Answer:** B. Remove rows with missing values

---

3. What does a Z-score of +3 indicate?

A. Value is below average
B. Value is missing
C. Value is 3 standard deviations above the mean
D. Value is invalid
✓ **Answer:** C. Value is 3 standard deviations above the mean

---

4. Which of the following methods is best for imputing missing time-series data?

A. KNN Imputation
B. Regression Imputation
C. Mode Imputation
D. Interpolation
✓ **Answer:** D. Interpolation

---

5. Which outlier handling technique transforms the scale to reduce skewness?

A. Drop rows
B. Winsorization
C. Log transformation

D. Mean imputation
✅ **Answer:** C. Log transformation

? Scenario-Based Questions

Scenario 1: Student Database

You are analyzing a student registration database. The `email` field is blank for 20% of the students, and you discover a few students are entered twice with spelling differences.

Q1. What types of data anomalies are present here?
✅ **Expected Answer:** Missing values and Duplicates

Q2. What preprocessing steps should you consider before analysis?
✅ **Expected Answer:** Impute or drop missing emails depending on model; deduplicate records using fuzzy matching or unique IDs.

Scenario 2: Retail Dataset

You find that the `gender` column contains values like `'Male'`, `'M'`, `'m'`, and `'female'`.

Q1. Which data quality dimension is violated?
✅ **Answer:** Consistency

Q2. What cleaning step is recommended?
✅ **Answer:** Standardize the gender column using `.replace()` or mapping to a uniform format.

Scenario 3: Sensor Data

A temperature sensor logs values like `32.1°C`, `31.9°C`, `200°C`, `32.0°C`.

Q1. Identify the anomaly.
✅ **Answer:** Outlier or Noisy data

Q2. Suggest a method to detect and deal with it.
✅ **Answer:** Use IQR or Z-score to detect outliers, optionally smooth or filter the noisy reading.

Scenario 4: Salary Data

The salary column in your dataset includes values such as ₹50,000, ₹52,000, and ₹5,000,000.

Q1. What type of anomaly does this represent?
✅ **Answer:** Outlier

Q2. Give two methods to handle this.
✅ **Answer:** Cap the extreme values (Winsorization) or log-transform the column.

Scenario 5: Machine Learning Dataset

You are preparing data for logistic regression. The dataset contains missing values in a critical feature, and another feature is skewed with outliers.

Q1. What should you do with missing values?
✅ **Answer:** Impute using mean/median or regression-based methods.

Q2. How would you deal with skewed outliers?
✅ **Answer:** Apply log transformation or remove outliers depending on distribution.

---

6. What type of missing data occurs when the probability of missingness is related to the observed data but not the missing data itself?

A. MCAR
B. MAR
C. MNAR
D. Random
✅ **Answer:** B. MAR

---

7. In the Shapiro-Wilk test, a p-value less than 0.05 implies:

A. Data is normally distributed
B. There are no outliers
C. Data is not normally distributed
D. The variance is too high
✅ **Answer:** C. Data is not normally distributed

---

8. Which of the following techniques helps reduce the effect of extreme values without removing them?

A. Deletion
B. Log transformation
C. Winsorization
D. Mode imputation
✅ **Answer:** C. Winsorization

---

9. Which pandas function helps detect missing values?

A. `pd.detect_na()`
B. `df.has_null()`
C. `df.isna()`
D. `df.find_missing()`
✅ **Answer:** C. `df.isna()`

---

10. What statistical test would you use to check the independence of two categorical variables?

A. t-test
B. ANOVA
C. Chi-square test
D. Z-test
✅ **Answer:** C. Chi-square test

▯ More Scenario-Based Questions

Scenario 6: Healthcare Dataset

You are analyzing a hospital's patient dataset. The `smoking_status` column has many missing entries, but analysis shows that younger patients are more likely to leave it blank.

Q1. What type of missing data is this?
✅ **Answer:** MAR (Missing At Random)

Q2. Suggest a suitable imputation strategy.
✅ **Answer:** Impute using mode separately for age groups or apply regression imputation.

Scenario 7: Online Survey

In a user feedback survey, users who rated the product poorly were less likely to answer questions about satisfaction.

Q1. What kind of missingness does this indicate?
✅ **Answer:** MNAR (Missing Not At Random)

Q2. Should you delete these records? Why or why not?
✅ **Answer:** Not advisable. Bias may increase; consider model-based or multiple imputation techniques.

Scenario 8: Vehicle Dataset

A dataset on cars has an attribute `price`, with values ranging from ₹4 lakhs to ₹4 crores. A histogram shows a strong right skew.

Q1. Suggest a preprocessing technique to make the distribution more normal.
✅ **Answer:** Apply log transformation on the price column.

Q2. What kind of model would benefit from this transformation?
✅ **Answer:** Linear models that assume normally distributed residuals.

Scenario 9: Product Quality Inspection

In an industrial dataset, sensor data values sometimes spike due to electrical noise. These values don't represent the true measurement.

Q1. Are these values outliers or noise?
✅ **Answer:** Noise

Q2. Suggest a method to handle it.
✅ **Answer:** Apply smoothing techniques (e.g., moving average), or domain-based thresholding.

Scenario 10: Student Marks Dataset

The average mark in a class is 65, but three students scored 0 due to absence. You need to analyze performance trends.

Q1. How would you treat the absent students' scores?
✅ **Answer:** Replace 0 with NaN, and use imputation if needed or remove them from performance analysis.

Q2. If you use mean imputation, what is the risk?
✅ **Answer:** It could distort the true average and reduce variability.