Handling Missing Values (NA, Null) in Data

What Are Missing Values?

Missing values (represented as NA, NaN, null, or blank cells) occur when no data is stored for a variable in an observation. They're common in real-world datasets and can appear for various reasons:

Data wasn't collected

Example:
A researcher forgot to include a question about income level in a survey, so that data is missing for all participants.

Data entry errors

Example:
While typing in survey results, an assistant mistakenly enters “220” as a person’s age instead of “22”.

Information wasn't applicable

Example:
A survey question asks about pregnancy status. This question doesn’t apply to male respondents, so their answers are left blank.

Respondents refused to answer

Example:
In a census form, some participants chose not to disclose their monthly income due to privacy concerns.

Why Do We Handle Missing Values?

1) Some machine learning models (e.g., LinearRegression, KNN, etc.) can't handle NaNs. You must either drop or impute.

2) Visualization issues - Many plotting functions fail with missing values

3) Inaccuracy: Summary statistics (mean, median, etc.) become incorrect.

How Do We Handle Missing Values?

☑ Detection

1)df.isnull().sum()   # count missing values column-wise

Shows how many missing values (NaN) are in each column:

```
Name    0
Age     1
Salary  2
dtype: int64
```

2)df.isnull().any()   # check if any missing in each column

Shows whether each column contains any missing values (True or False):

```
Name    False
Age     True
Salary  True
dtype: bool
```

✅. Removal

1) Drop rows with missing values:

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

When You Should Drop Rows:

1) Very Few Missing Rows

If only a small percentage of rows have missing data (e.g., 1–5%), dropping them won't affect your analysis much.

✔Example:
You have 10,000 rows and only 20 rows have missing values → dropping is safe

2) Data is Completely Missing in a Row

If all or most columns in a row are NaN, it's better to drop it

When You Should NOT Drop Rows:

1. Too Much Data Will Be Lost

If many rows have missing values, dropping them could remove valuable information and introduce bias.

Example:
You have only 200 rows, and 100 of them have a missing value in one column — don't drop unless you must.

2. Missing Values Are in Important Columns

If a critical feature (like "Age" or "Income") is missing, it's often better to fill in (impute) the value instead of dropping

✅Drop columns with too many missing values

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

When You Should Drop Columns with Missing Values:

1. Too Much Missing Data in Column

If 50% or more values in a column are missing, it's often better to drop it (unless the feature is very important).

✅Example:
A column Height is missing in 80% of rows → drop it, not useful.

2. Column is Not Useful or Not Related to Target

If the column has missing values and isn’t helpful for analysis or prediction.

✅Example:
Profile_Picture_URL is missing in many rows, and you're predicting income → drop it.

✅Imputation (Filling Missing Values)

1)Fill with a fixed value:

```
df['column'].fillna(0, inplace=True)
```

When You **Should Use Fixed Value Imputation:

1. Missing Value Has Meaning

If a missing value actually means something, you can replace it with a fixed value.

✅Example:
Number of purchases missing → fill with 0 (means no purchases)

2.Categorical Variables with Missing as Unknown

Example:

City or Country missing → fill with 'Unknown' or 'Not Specified'

☑ Fill with column mean, median, or mode:

☑Numerical Data with Symmetrical Distribution — Use Mean

What is symmetrical distribution?

The data is evenly spread around the center.

The left and right sides of the distribution mirror each other.

Why use Mean?

The mean (average) is a good measure of central tendency here.

It considers all values and balances out perfectly in symmetrical data.

Filling missing values with the mean maintains the overall balance and average of the dataset.

Example: Age

Ages in a population may roughly follow a symmetrical pattern (depending on sample).

Filling missing ages with the mean age makes sense as it doesn’t bias towards very young or very old.

```
df['column'].fillna(df['column'].mean(), inplace=True)
```

✅ Numerical Data with Skewed Distribution — Use Median

What is skewed distribution?

The data is not symmetrical.

It’s biased towards one side, having a long tail on the other.

Examples: Income, house prices, sales amount (most people earn average, few earn very high).

Why use Median?

The median is the middle value when data is sorted.

It is less affected by outliers or extreme values.

Using median avoids the distortion caused by very high or very low values.

Example: Income or Salary

Income data is often right-skewed with some very high earners.

Filling missing salaries with the median gives a more representative central value

```
df['column'].fillna(df['column'].median(), inplace=True)
```

✅Categorical Data — Use Mode

What is categorical data?

Data that represents categories or groups rather than numbers.

Examples: Gender, City, Education level.

Why use Mode?

The mode is the most frequent category in the column.

Filling missing values with the mode assigns the most common category to missing places.

It's simple and effective when no other info is available.

Example: Gender or City

If most people in your dataset are male, missing gender can be filled with "Male".

If most respondents are from Dhaka, missing city can be filled with "Dhaka".

```
df['column'].fillna(df['column'].mode()[0], inplace=True)
```