# Data Cleaning and Preprocessing Guide

## Part 1: Cleaning Steps Using Python

1. Load the dataset using pandas.

   Example: df = pd.read_csv('filename.csv')

2. Handle missing values:

   - Detect using df.isnull().sum()

   - Drop missing rows using df.dropna()

   - Fill missing values using df.fillna(value), e.g., df['column'].fillna(df['column'].median())

3. Remove duplicate records using df.drop_duplicates()

4. Standardize text values:

   - Convert to lowercase: df['column'] = df['column'].str.lower()

   - Remove whitespace: df['column'] = df['column'].str.strip()

5. Convert date formats using:

   - df['date_column'] = pd.to_datetime(df['date_column'], format='%d-%m-%Y')

6. Rename columns for uniformity:

   - df.columns = df.columns.str.lower().str.replace(' ', '_')

7. Check and fix data types:

   - Use df.dtypes to check types

- Convert using df['column'] = df['column'].astype(desired_type)

## Part 2: Summary of Changes (Example)

Data Cleaning Summary for 'Customer Personality Analysis' Dataset:

1. Missing Values:

- Detected missing values using df.isnull().sum()
- Filled 'Income' missing values with median using df['Income'].fillna(df['Income'].median())

2. Duplicate Records:

- Removed 42 duplicates using df.drop_duplicates()

3. Standardization:

- Converted 'Education' and 'Marital_Status' to lowercase and stripped whitespace
- Renamed columns to snake_case format using df.columns.str.lower().str.replace(' ', '_')

4. Date Formats:

- Converted 'Dt_Customer' column to datetime format using pd.to_datetime()

5. Data Types:

- Converted 'Income' to float and 'Kidhome' to integer

## Part 3: Interview Questions and Detailed Answers

# Data Cleaning and Preprocessing Guide

1. What are missing values and how do you handle them?

   - Missing values represent absence of data. Handle by:

     - Dropping rows/columns with too many nulls (df.dropna())

     - Imputing with mean, median, or mode (df.fillna())

     - Using interpolation or predictive models

2. How do you treat duplicate records?

   - Identify using df.duplicated()

   - Remove using df.drop_duplicates()

   - Always verify that duplicates are true duplicates before removing

3. Difference between dropna() and fillna()?

   - dropna(): Removes rows/columns with missing values

   - fillna(): Replaces missing values with specified values (mean, median, etc.)

4. What is outlier treatment and why is it important?

   - Outliers are extreme data values that distort analysis

   - Detect using box plots, Z-score, or IQR methods

   - Treat by removing, capping, or transforming data (log or square root)

5. Explain the process of standardizing data.

   - Transforming data into a consistent format:

     - Column names to lowercase and underscores

     - Dates in a uniform format (e.g., dd-mm-yyyy)

  - Text values made consistent (e.g., 'Male', 'male' -> 'male')
  - Numeric types corrected (e.g., int, float)

6. How do you handle inconsistent data formats (e.g., date/time)?

- Use pd.to_datetime() to parse and format dates
- Convert timezones if needed
- Use string operations to clean and format textual inconsistencies

7. What are common data cleaning challenges?

- Incomplete or missing values
- Inconsistent formatting and data entry errors
- Mixed data types in a single column
- Outliers and noisy data
- Duplicates and data redundancy

8. How can you check data quality?

- Use df.info(), df.describe(), df.isnull().sum(), df.duplicated().sum()
- Visual checks: histograms, box plots
- Value counts and unique() for categorical data
- Consistency across related columns (e.g., age vs date of birth)