# Comprehensive guide on Data Cleaning

Data cleaning is essential to ensure the accuracy and reliability of your data analysis or machine learning models. Here's a comprehensive guide to different data cleaning strategies, including when to use each method, examples, and code snippets.

## 1. Handling Missing Values

### Types of Missing Data

**Missing completely at Random (MCAR):** The missingness of data points is entirely random and not related to any other data points.

**Missing at Random (MAR):** The missingness is related to some observed data but not to the missing data itself.

**Missing Not at Random (MNAR):** The missingness is related to the missing data itself (e.g., people with higher incomes might be less likely to report their income).

### Strategies for Handling Missing Data

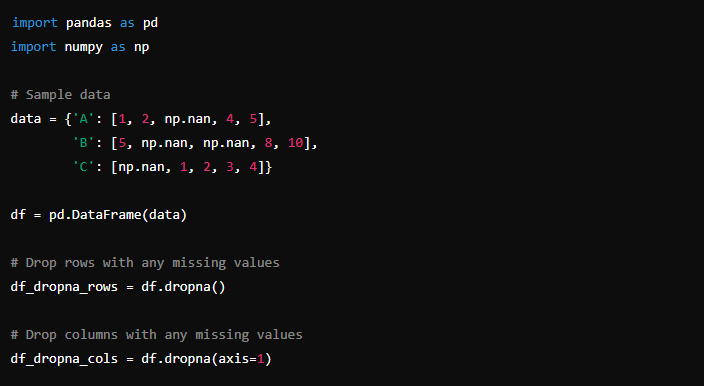
**a. Removing Missing Values**

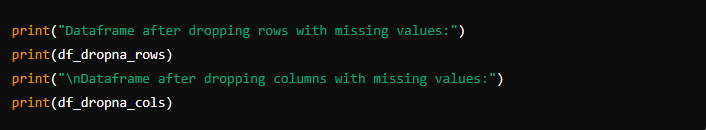
**When to Use:** This is appropriate when you have a large dataset and the amount of missing data is small (e.g., <5% of the total data). Removing rows or columns with missing values will not significantly impact the analysis.

**Pros:** Simple and quick; no need for complex algorithms.

**Cons:** Can lead to loss of valuable information, especially if missing data is substantial.

Example:





### b. Imputing Missing Values

Imputation involves replacing missing data with substituted values. There are various imputation strategies:

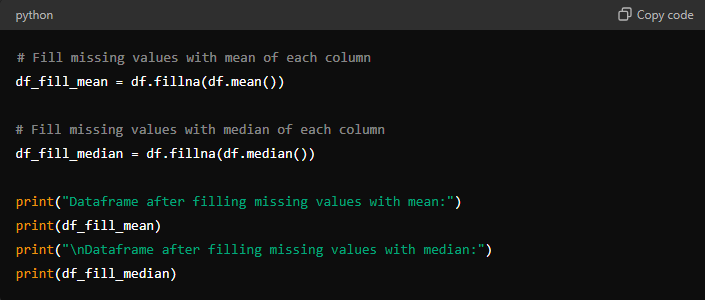
**Mean/Median Imputation:**

**When to Use:** Suitable for numerical data with no significant outliers. If the data distribution is skewed, use the median; otherwise, use the mean.

**Pros:** Easy to implement; maintains the sample size.

**Cons**: Can distort the original distribution; reduces variance.

**Example Code:**



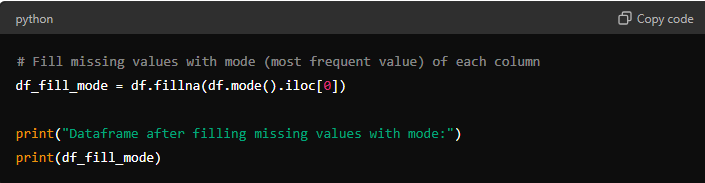
**Mode Imputation:**

**When to Use:** Suitable for categorical data.

**Pros:** Retains the most frequent category.

**Cons:** Can introduce bias if one category is overrepresented.

**Example Code:**



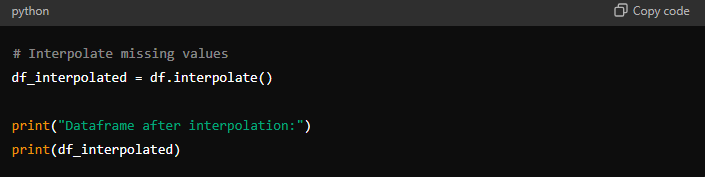
**Interpolation:**

**When to Use:** Best for time series or sequential data where you can reasonably assume that the missing values fall along a trend.

**Pros:** Preserves trends and continuity in time series data.

**Cons:** Can introduce errors if the assumptions about trends are incorrect.

**Example Code:**



### c. Using Advanced Imputation Techniques

**K-Nearest Neighbors (KNN) Imputation:**

**When to Use:** When the dataset is not too large, and the missing values are not random. This method considers the similarity between observations.

**Pros:** Can produce more accurate estimates than simple mean/median imputation.

**Cons:** Computationally expensive; not suitable for very large datasets.

**Multiple Imputation:**

**When to Use:** When missing data is significant and you want to account for the uncertainty of missing data.

**Pros:** Provides a more accurate reflection of the uncertainty due to missing data.

**Cons:** More complex and computationally intensive.

## 2. Removing Duplicates

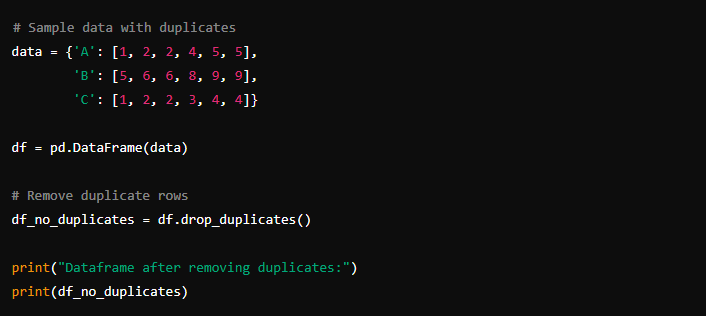
### a. Removing Duplicate Rows or Columns

**When to Use:** When you have multiple instances of the same data point. This often occurs when merging datasets.

**Pros:** Prevents bias and errors in analysis due to repeated data.

**Cons:** If duplicates contain unique information, it may lead to data loss.

**Example Code:**



## 3. Fixing Structural Errors

Structural errors occur when data is mis-formatted or incorrectly labeled, often due to human error.

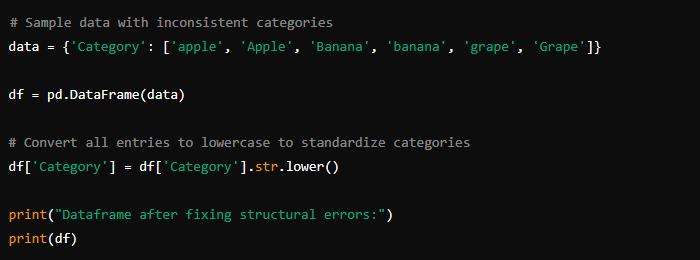
### a. Correcting Typos and Inconsistent Capitalization

**When to Use:** Common in categorical variables where categories should be standardized.

**Pros**: Ensures consistency and correctness in categorical variables.

**Cons:** Can be time-consuming if done manually.

E**xample Code:**



## 4. Handling Outliers

Outliers can skew data and affect the results of your analysis or machine learning models.

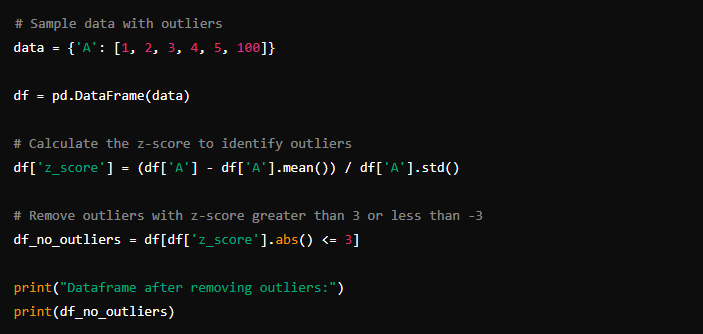
### a. Removing Outliers

**When to Use:** When you are confident that the outliers are errors or not representative of the data you are analyzing.

**Pros:** Prevents skewing of analysis and model results.

**Cons:** May lead to loss of potentially valuable data.

**Example Code:**



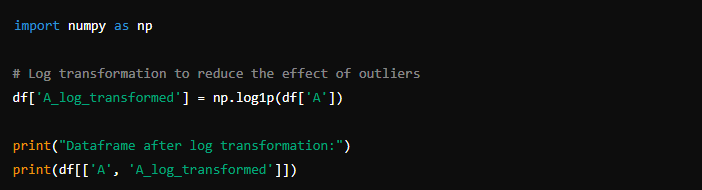
### b. Transforming Data

**When to Use:** When you cannot remove outliers, but their influence needs to be reduced (e.g., using log transformation).

**Pros:** Reduces the effect of outliers while retaining them in the dataset.

**Cons:** Can make interpretation more complex.

**Example Code:**



## 5. Encoding Categorical Variables

Machine learning models typically require numerical input. Categorical data must be converted into a numerical format.

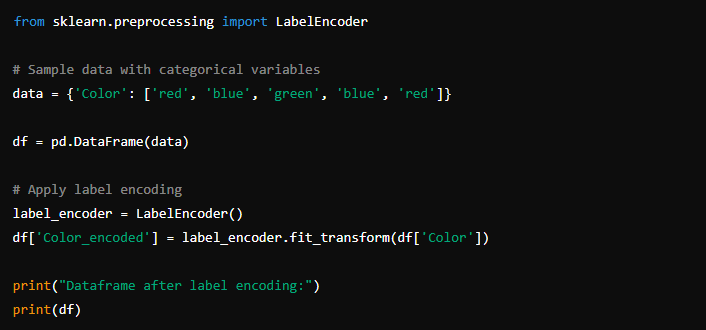
### a. Label Encoding

**When to Use:** For ordinal data where the categories have an intrinsic order.

**Pros:** Simple to implement; preserves ordinal relationship.

**Cons:** Imposes an arbitrary ordering on nominal data, which can introduce errors.

**Example Code:**



### b. One-Hot Encoding

**When to Use:** For nominal data where categories do not have an intrinsic order.

**Pros:** Avoids imposing ordinal relationships; suitable for nominal data.

**Cons:** Can lead to high dimensionality with many categories.

**Example Code:**

