Technique	Description	When to Use
Handling Missing Values	- `isna()`, `isnull()`: Detect missing values. - `fillna()`: Fill missing values with specified values. - `dropna()`: Drop rows or columns with missing values. - `interpolate()`: Interpolate missing values.	- When dealing with datasets containing missing values. - When preparing data for analysis or modeling. - When missing values can be handled through imputation, deletion, or interpolation without significantly affecting the analysis.
Handling Duplicates	- `duplicated()`: Identify duplicate rows. - `drop_duplicates()`: Remove duplicate rows.	- When dealing with datasets containing duplicate entries. - When preparing data for analysis or modeling. - When duplicate entries need to be removed to ensure data integrity.
Handling Outliers	- Statistical methods (e.g., Z-score, IQR) to detect and remove outliers. - `clip()`: Clip outliers to a specified range. - `winsorize()`: Winsorize outliers to a specified percentile range.	- When dealing with datasets containing outliers. - When preparing data for analysis or modeling. - When outliers need to be addressed to ensure accurate analysis or modeling results.
Data Transformation	- `map()`, `apply()`: Apply a function to each element. - `cut()`, `qcut()`: Bin values into discrete intervals. - `astype()`: Convert the data type of a column. - `transform()`: Apply a function to each group separately.	- When transforming data to a different format or scale. - When preparing data for analysis or modeling. - When specific transformations are required for analysis or modeling algorithms. - When applying group-wise transformations.
String Operations	- `str.strip()`, `str.lower()`, `str.upper()`: Modify string elements. - `str.contains()`, `str.replace()`: Perform searches and replacements. - `str.extract()`: Extract substrings using regular expressions. - `str.split()`: Split strings into substrings.	- When dealing with string data that requires cleaning or manipulation. - When preparing text data for analysis or modeling. - When specific string operations are needed for data processing. - When extracting information from text data.
Indexing and Selection	- `set_index()`, `reset_index()`: Set or reset the DataFrame index. - `loc[]`, `iloc[]`: Label-based and integer-based indexing. - `at[]`, `iat[]`: Fast scalar value access. - `query()`: Filter rows using a boolean expression.	- When restructuring DataFrame indexes for analysis or visualization. - When selecting specific rows or columns for analysis or visualization. - When accessing individual elements quickly. - When filtering rows based on a boolean condition.
Sorting Data	- `sort_values()`: Sort rows by column values. - `sort_index()`: Sort by index labels.	- When arranging data in a specific order for analysis or visualization. - When sorting data

		is necessary for further data processing.
Reshaping Data	- `melt()`: Unpivot DataFrame from wide to long format. - `pivot()`, `pivot_table()`: Reshape DataFrame from long to wide format. - `stack()`, `unstack()`: Pivot index and columns. - `merge()`, `join()`: Merge DataFrame objects by index or columns.	- When restructuring DataFrame layout for analysis or visualization. - When converting data between wide and long formats. - When dealing with hierarchical data structures. - When combining multiple datasets based on common columns or indexes.
Combining DataFrames	- `concat()`: Concatenate DataFrames along rows or columns. - `merge()`, `join()`: Merge DataFrame objects by index or columns. - `append()`: Append rows of one DataFrame to another. - `combine_first()`: Combine DataFrame objects, filling NaN values with values from another DataFrame.	- When combining multiple datasets into one for analysis or modeling. - When merging datasets based on common columns or indexes. - When appending rows of one DataFrame to another. - When filling missing values in one DataFrame with values from another DataFrame.
Handling Categorical Data	- `get_dummies()`: Convert categorical variable into dummy/indicator variables. - Label encoding and one-hot encoding techniques. - `replace()`: Replace values in a DataFrame.	- When preparing categorical data for analysis or modeling. - When converting categorical variables into a format suitable for machine learning algorithms. - When replacing specific values in a DataFrame.
Sampling Data	- `sample()`: Randomly select rows or columns from a DataFrame.	- When creating a subset of data for analysis or modeling. - When performing random sampling for statistical analysis.
Handling Time Series Data	- `pd.to_datetime()`: Convert column to datetime type. - Resampling methods (`resample()`) for changing frequency. - `rolling()`: Calculate rolling statistics. - `shift()`: Shift index by a specified number of periods.	- When working with time series data. - When analyzing or modeling temporal patterns. - When aggregating data over different time periods. - When calculating rolling statistics for time series analysis. - When creating lag features for time series modeling.