Here are some of the important functions in **Pandas** that you can use for data manipulation, cleaning, analysis, and transformation:

**1. Data Creation & Inspection**

1. pd.DataFrame() – Create a DataFrame from a dictionary or list.
2. pd.Series() – Create a Series from a list or dictionary.
3. df.head() – View the first 5 rows of a DataFrame.
4. df.tail() – View the last 5 rows of a DataFrame.
5. df.info() – Get information about a DataFrame (columns, data types, non-null values).
6. df.describe() – Summary statistics for numerical columns.
7. df.shape – Get the dimensions of a DataFrame (rows, columns).
8. df.columns – Get column names of a DataFrame.
9. df.dtypes – Check the data types of each column.
10. df.index – Get the index (row labels) of the DataFrame.

**2. Data Selection & Slicing**

1. df['column'] – Select a single column.
2. df[['col1', 'col2']] – Select multiple columns.
3. df.loc[] – Label-based indexing to select rows and columns.
4. df.iloc[] – Index-based selection (integer position).
5. df.at[] – Access a single scalar value by row/column labels.
6. df.iat[] – Access a single scalar value by row/column indices.
7. df.filter() – Subset the DataFrame using a string match for column names.
8. df.sample() – Randomly select a subset of rows.

**3. Data Cleaning**

1. df.isnull() – Check for missing values in the DataFrame.
2. df.notnull() – Check for non-missing values.
3. df.fillna() – Fill missing values with a specific value or method (e.g., forward fill).
4. df.dropna() – Drop rows or columns with missing values.
5. df.replace() – Replace values in the DataFrame (e.g., replace 0 with NaN).
6. df.duplicated() – Identify duplicate rows.
7. df.drop\_duplicates() – Remove duplicate rows.

**4. Data Transformation**

1. df.apply() – Apply a function along an axis (rows or columns).
2. df.map() – Map values of a Series according to an input correspondence.
3. df.groupby() – Group the DataFrame by one or more columns and apply aggregation.
4. df.pivot() – Reshape data by creating a new index and columns.
5. df.pivot\_table() – Create a pivot table with specified aggregation functions.
6. df.melt() – Unpivot a DataFrame from wide to long format.
7. df.stack() – Stack the DataFrame from columns to index.
8. df.unstack() – Unstack the DataFrame from index to columns.
9. df.rename() – Rename columns or index labels.
10. df.astype() – Change the data type of a column.

**5. Data Merging**

1. pd.merge() – Merge two DataFrames based on key columns.
2. pd.concat() – Concatenate DataFrames along rows or columns.
3. df.join() – Join DataFrames on indexes or key columns.
4. pd.merge\_asof() – Perform an as-of merge on ordered DataFrames.
5. pd.merge\_ordered() – Perform a merge with ordered data.

**6. Statistical Operations**

1. df.mean() – Calculate the mean of columns or rows.
2. df.median() – Calculate the median of columns or rows.
3. df.std() – Calculate the standard deviation.
4. df.var() – Calculate the variance.
5. df.corr() – Calculate the correlation between columns.
6. df.cumsum() – Calculate the cumulative sum.
7. df.cumprod() – Calculate the cumulative product.
8. df.count() – Count non-null values in each column.

**7. Input & Output**

1. pd.read\_csv() – Read a CSV file into a DataFrame.
2. df.to\_csv() – Write a DataFrame to a CSV file.

**8. Advanced Data Manipulation**

1. df.sort\_values() – Sort the DataFrame by one or more columns.
2. df.sort\_index() – Sort the DataFrame by the index.
3. df.set\_index() – Set the DataFrame index to one of its columns.
4. df.reset\_index() – Reset the index to default (convert index to a column).
5. df.diff() – Calculate the difference between rows (or columns).
6. df.rank() – Rank the elements in each column (or row).
7. df.corrwith() – Compute correlation with another Series or DataFrame.
8. df.idxmax() – Get the index of the first occurrence of the maximum value.
9. df.idxmin() – Get the index of the first occurrence of the minimum value.

**9. Window Functions**

1. df.rolling() – Create a rolling window for time-series data (e.g., moving averages).
2. df.expanding() – Perform expanding calculations (e.g., cumulative max).
3. df.ewm() – Exponentially weighted moving window functions.

**10. Data Aggregation**

1. df.agg() – Aggregate using one or more operations (e.g., sum, mean, etc.).
2. df.transform() – Apply a function element-wise to transform columns.
3. df.value\_counts() – Count unique values in a Series.
4. df.mode() – Get the mode(s) of a DataFrame or Series.
5. df.quantile() – Calculate the quantile of numerical columns.
6. df.mad() – Calculate the mean absolute deviation.
7. df.kurt() – Calculate kurtosis (measure of tailedness).
8. df.skew() – Calculate the skewness of the data.
9. df.nunique() – Count the number of unique values per column.

**11. Resampling and Time-Series**

1. df.resample() – Resample time-series data (e.g., downsample daily data to monthly).
2. df.asfreq() – Convert a time-series to a specified frequency (e.g., daily, monthly).
3. pd.to\_datetime() – Convert strings or numbers to datetime objects.
4. pd.date\_range() – Create a range of dates.
5. df.tz\_localize() – Localize a time-series to a specific timezone.
6. df.tz\_convert() – Convert time-series from one timezone to another.

**12. Input & Output (continued)**

1. pd.read\_excel() – Read an Excel file into a DataFrame.
2. df.to\_excel() – Write a DataFrame to an Excel file.
3. pd.read\_json() – Read a JSON file into a DataFrame.
4. df.to\_json() – Convert a DataFrame to JSON format.
5. pd.read\_sql() – Read from an SQL database into a DataFrame.
6. df.to\_sql() – Write a DataFrame to an SQL database.

**13. Miscellaneous Functions**

1. df.memory\_usage() – Get memory usage of a DataFrame.
2. df.eval() – Evaluate an expression within the context of the DataFrame (e.g., column-wise operations).
3. df.query() – Query the DataFrame using a SQL-like syntax.
4. pd.get\_dummies() – Convert categorical variables to dummy/indicator variables.
5. pd.cut() – Bin data into discrete intervals (useful for bucketing numerical data).
6. pd.qcut() – Bin data into equal-sized quantiles.
7. pd.factorize() – Encode categorical values as integer labels.
8. pd.pivot\_table() – Create a pivot table with multi-level indexing.
9. pd.crosstab() – Create a cross-tabulation (contingency table) of two factors.
10. df.explode() – Transform each element of a list-like column into a row.
11. pd.options.display.max\_columns – Change the maximum number of columns displayed.
12. pd.options.display.max\_rows – Change the maximum number of rows displayed.