Here is a comprehensive list of commonly used methods and functions available in pandas:

DataFrame Creation

- `pd.DataFrame()`: Creates a DataFrame from various data structures (lists, dictionaries, arrays, etc.).
- `pd.read_csv()`: Reads a CSV file into a DataFrame.
- `pd.read_excel()`: Reads an Excel file into a DataFrame.
- `pd.read_sql()`: Reads a SQL query or database table into a DataFrame.
- `pd.read_json()`: Reads a JSON string or file into a DataFrame.
- `pd.read_html()`: Reads HTML tables into a list of DataFrames.
- `pd.read_clipboard()`: Reads the clipboard into a DataFrame.
- `pd.read_pickle()`: Reads a pickle file into a DataFrame.

DataFrame Attributes

- `df.columns`: Returns the column labels of the DataFrame.
- `df.index`: Returns the row labels of the DataFrame.
- `df.dtypes`: Returns the data types of each column.
- `df.shape`: Returns a tuple representing the dimensionality of the DataFrame.
- `df.size`: Returns the number of elements in the DataFrame.
- `df.values`: Returns the DataFrame as a NumPy array.

Data Selection and Indexing

- `df.head()`: Returns the first n rows.
- `df.tail()`: Returns the last n rows.
- `df.sample()`: Returns a random sample of rows.
- `df.loc[]`: Access a group of rows and columns by labels or a boolean array.
- `df.iloc[]`: Access a group of rows and columns by integer position(s).
- `df.at[]`: Access a single value for a row/column label pair.
- `df.iat[]`: Access a single value for a row/column pair by integer position.

Data Cleaning and Preparation

- `df.drop()`: Drops specified labels from rows or columns.
- `df.dropna()`: Removes missing values.
- `df.fillna()`: Fills missing values with specified values.
- `df.replace()`: Replaces specified values with others.
- `df.rename()`: Renames labels.
- `df.duplicated()`: Returns boolean Series denoting duplicate rows.
- `df.drop_duplicates()`: Removes duplicate rows.
- `df.astype()`: Casts a pandas object to a specified dtype.
- `df.sort_values()`: Sorts by the values along either axis.
- `df.sort_index()`: Sorts by the index.
- `df.reset_index()`: Resets the index.

DataFrame Operations

- `df.append()`: Appends rows of other DataFrame to the end of the caller DataFrame.
- `df.merge()`: Merges DataFrame objects with a database-style join.
- `df.join()`: Joins columns of another DataFrame.
- `df.concat()`: Concatenates pandas objects along a particular axis.
- `df.pivot()`: Reshapes data based on column values.
- `df.pivot_table()`: Creates a pivot table.
- `df.melt()`: Unpivots a DataFrame from wide format to long format.
- `df.groupby()`: Groups data by specified columns.
- `df.agg()`: Aggregates using one or more operations over specified axis.
- `df.transform()`: Applies a function to each group independently.
- `df.apply()`: Applies a function along an axis of the DataFrame.
- `df.applymap()`: Applies a function elementwise.

Statistical Functions

- `df.mean()`: Returns the mean of the values.
- `df.median()`: Returns the median of the values.
- `df.mode()`: Returns the mode of the values.
- `df.min()`: Returns the minimum of the values.
- `df.max()`: Returns the maximum of the values.
- `df.sum()`: Returns the sum of the values.
- `df.count()`: Returns the number of non-null values.
- `df.std()`: Returns the standard deviation of the values.
- `df.var()`: Returns the variance of the values.
- `df.corr()`: Computes pairwise correlation of columns.
- `df.cov()`: Computes pairwise covariance of columns.
- `df.describe()`: Generates descriptive statistics.

Time Series

- `pd.to_datetime()`: Converts argument to datetime.
- `df.resample()`: Convenience method for frequency conversion and resampling of time series.
- `df.asfreq()`: Converts the DataFrame to another frequency.
- `df.shift()`: Shifts index by desired number of periods with an optional time frequency.
- `df.tz_localize()`: Localizes tz-naive DatetimeIndex to target time zone.
- `df.tz_convert()`: Converts tz-aware DatetimeIndex from one time zone to another.

Input and Output

- `df.to_csv()`: Writes the DataFrame to a CSV file.
- `df.to_excel()`: Writes the DataFrame to an Excel file.
- `df.to_sql()`: Writes the DataFrame to a SQL database.
- `df.to_json()`: Writes the DataFrame to a JSON file.
- `df.to_html()`: Writes the DataFrame to an HTML file.

- `df.to_clipboard()`: Copies the DataFrame to the clipboard.
- `df.to_pickle()`: Writes the DataFrame to a pickle file.

Visualization

- `df.plot()`: Plots the data using matplotlib.
- `df.hist()`: Draws histograms for each column.
- `df.boxplot()`: Draws box plots for each column.
- `df.scatter_matrix()`: Draws a matrix of scatter plots.