Pandas for Data Analysis Handbook

Pandas for Data Analysis Handbook

Data Loading and I/O:

- **read_csv(filepath):** Reads a CSV file into a DataFrame.
- **read_excel(filepath):** Reads an Excel file into a DataFrame.
- **to_csv(filepath):** Saves a DataFrame to a CSV file.
- **to_excel(filepath):** Saves a DataFrame to an Excel file.
- **read_sql(sql, con):** Reads data from a SQL database into a DataFrame (requires

Data Cleaning:

- **isnull():** Returns a DataFrame indicating missing values (NaNs).
- **notnull():** Returns the opposite of isnull().
- **fillna(value):** Replaces missing values with a specified value.
- **dropna(axis=0, inplace=False):** Drops rows or columns with missing values (axis=0 for rows, axis=1 for columns, inplace modifies the DataFrame).
- **duplicated():** Identifies duplicate rows.
- **drop_duplicates(inplace=False):** Removes duplicate rows (inplace modifies the DataFrame).

Data Inspection:

- **head(n):** Returns the first n rows of a DataFrame.
- **tail(n):** Returns the last n rows of a DataFrame.
- **info():** Provides information about data types and memory usage.
- **describe():** Generates summary statistics (mean, std, quartiles, etc.) for numerical columns.
- **dtypes:** Returns data types of all columns in a DataFrame.

Data Selection:

- **loc[indexer]:** Selects rows and/or columns by label-based indexing.
- **iloc[indexer]:** Selects rows and/or columns by integer-based indexing.
- **at[indexer]:** Selects a single value by label.
- **iat[indexer]:** Selects a single value by integer position.

Data Transformation:

- **astype(dtype):** Converts columns to a specific data type.
- **replace(to_replace, value):** Replaces specific values with new values.
- **rename(columns=mapper):** Renames columns with a dictionary or function.
- **apply(func, axis=0):** Applies a function to each row (axis=0) or column (axis=1).
- **map(func):** Applies a function to each element in a Series.

Pandas for Data Analysis Handbook

Aggregation and Calculation:

- **sum():** Calculates the sum of elements in a Series or DataFrame (axis=0 for rows, axis=1 for columns).
- **mean():** Calculates the mean (average) of elements in a Series or DataFrame.
- **std():** Calculates the standard deviation of elements in a Series or DataFrame.
- **min():** Returns the minimum value in a Series or DataFrame.
- **max():** Returns the maximum value in a Series or DataFrame.

String Manipulation:

- **str.strip():** Removes leading/trailing whitespace from string elements.
- **str.lower():** Converts strings to lowercase.
- **str.upper():** Converts strings to uppercase.
- **str.split(sep):** Splits strings into lists based on a separator.
- **str.replace(old, new):** Replaces occurrences of a substring with another.

Date and Time Manipulation:

- **to_datetime(errors='coerce'):** Converts strings to datetime format, handling potential errors.
- **dt.year:** Extracts the year from datetime columns.
- **dt.month:** Extracts the month from datetime columns.
- **dt.day:** Extracts the day from datetime columns.
- **dt.strftime(format):** Formats datetime columns according to a format string.

Missing Value Handling:

- **isna():** Returns a DataFrame indicating NaN values.
- **notna():** Returns the opposite of isna().
- **groupby(by):** Groups DataFrame rows based on one or more columns for aggregation or applying functions.
- **pivot_table(values, index, columns, aggfunc=None):** Creates a pivot table summarizing data by specified rows, columns, and aggregation functions.
- **concat(objs, ignore_index=True):** Concatenates DataFrames along a specified axis (0 for rows, 1 for columns), optionally ignoring original row indices.
- **merge(right, how='inner', on=None):** Merges two DataFrames based on a specified column or key, using different join types (inner, left, right, outer).
- **get_dummies(data, columns=None):** Creates one-hot encoded dummy columns for categorical variables in a DataFrame.