#_ important Pandas Operations [+100]

Basics & I/O:

- `pd.read_csv(filename)`: Read a comma-separated values file.
- `pd.read_excel(filename)`: Read an Excel file.
- `pd.read_sql(query, connection)`: Read from a SQL table/database.
- `df.to_csv(filename)`: Write to a CSV file.
- `df.to_excel(filename)`: Write to an Excel file.
- `df.head(n)`: Display the first `n` rows.
- `df.tail(n)`: Display the last `n` rows.
- `df.describe()`: Summary statistics.

Data Creation:

- `pd.DataFrame(data)`: Create a DataFrame.
- `pd.Series(data)`: Create a Series.

Selection:

- `df[col]`: Select column by column name.
- `df[[col1, col2]]`: Select multiple columns.
- `df.iloc[row, col]`: Select by row and column integer indices.
- `df.loc[row_label, col_label]`: Select by row and column labels.

Filtering:

- `df[df[col] > value]`: Rows where the column is greater than value.
- `df.query("col > value")`: Use query method to filter rows.
- `df[df[col].isin(values)]`: Rows where the column is in `values`.

Data Cleaning:

- `df.dropna()`: Drop missing values.
- `df.fillna(value)`: Fill missing values.
- `df.replace(old_val, new_val)`: Replace values.
- `df.drop_duplicates()`: Drop duplicate rows.

By: Waleed Mousa

Data Transformation:

- df.set_index(col): Set column as index.
- df.reset_index(): Reset index to default integer index.
- df.pivot_table(): Create a pivot table.
- df.melt(): Unpivot a DataFrame.

Combining Data:

- pd.concat([df1, df2]): Concatenate DataFrames.
- df1.append(df2): Append rows of DataFrames.
- pd.merge(df1, df2, on=col): Merge DataFrames using a column.

Aggregation:

- df.groupby(col): Group by column.
- df.agg(functions): Aggregate using one or more functions.

Sorting & Ranking:

- df.sort_values(by=col): Sort by column.
- df.rank(): Rank rows.

Apply Functions:

- df.apply(func): Apply a function.
- df[col].map(func): Apply a function to a column.

Time Series:

- pd.to_datetime(col): Convert a column to datetime.
- df.resample(): Resample time-series data.
- df.asfreq(): Convert time series frequency.

Text Data:

- df[col].str.split(): Split string values.
- df[col].str.contains(pattern): Check if string contains a pattern.
- df[col].str.replace(old, new): Replace text.

By: Waleed Mousa

Categorical Data:

- `df[col].astype("category")`: Convert column to categorical type.
- `df[col].cat.set_categories()`: Set categories.

Missing Data:

- `df.isna()`: Check for missing values.
- `df.notna()`: Check for non-missing values.

Plotting:

- `df.plot()`: Plot data.
- `df[col].hist()`: Plot a histogram.

Computations:

- `df[col].cumsum()`: Cumulative sum.
- `df[col].pct_change()`: Percentage change.

Statistical **Operations**:

- `df[col].mean()`: Mean of column.
- `df[col].median()`: Median of column.
- `df.corr()`: Correlation matrix.

Window Functions:

- `df[col].rolling(window)`: Create a rolling window.
- `df[col].expanding()`: Create an expanding window.

String Methods:

- `df[col].str.lower()`: Convert to lowercase.
- `df[col].str.upper()`: Convert to uppercase.
- `df[col].str.strip()`: Strip whitespaces.

Renaming & Reordering:

- `df.rename(columns=dict)`: Rename columns.
- `df.reorder_levels()`: Reorder levels on multi-level index.

Dummies & Factorize:

By: Waleed Mousa

- `pd.get_dummies(df[col])`: Convert categorical variable into dummy variables.
- `pd.factorize(df[col])`: Encode categorical values.

Working with Index:

- `df.set_index()`: Set column as index.
- `df.reset_index()`: Reset index.

Iteration:

- `df.iterrows()`: Iterate over DataFrame rows as index and Series.
- `df.itertuples()`: Iterate over DataFrame rows as namedtuples.

Advanced Indexing:

- `df.at[row, col]`: Access single value using row and column label.
- `df.iat[row, col]`: Access single value using row and column integer.

MultiIndex Operations:

- `df.stack()`: Stack columns to rows.
- `df.unstack()`: Unstack rows to columns.

Database-style Operations:

- `df.query(expr)`: Query DataFrame using a string expression.
- `df.eval(expr)`: Evaluate a string expression in the DataFrame context.

Performance:

- `pd.DataFrame.eval()`: Evaluate an expression using DataFrame columns.
- `pd.DataFrame.query()`: Query DataFrame using a compact string syntax.

Timezone Handling:

By: Waleed Mousa

- `df.tz_localize(tz)`: Localize tz-naive time series to a particular timezone.
- `df.tz_convert(tz)`: Convert tz-aware axis to target timezone.

Offset Aliases:

- `pd.Timedelta(days=1)`: Create a Timedelta of one day.
- `pd.DateOffset(months=3)`: Create a DateOffset of three months.

Sparse Data:

- `pd.SparseDataFrame(data)`: Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes.

Reshaping and Pivot Tables:

- `df.pivot(index, columns, values)`: Reshape data based on column values.
- `pd.melt(df)`: Unpivot a DataFrame.

Other **Operations**:

- `df.memory_usage()`: Memory usage of each column.
- `df.info()`: Concise summary of the DataFrame.
- `df.shape`: Return a tuple representing the dimensionality of the DataFrame.
- `df.dtypes`: Return the dtypes in the DataFrame.
- `df.columns`: Return the column labels of the DataFrame.
- `df.values`: Return a Numpy representation of the DataFrame.
- `df.T`: Transpose the DataFrame.
- `df.clip(lower, upper)`: Trim values at input threshold(s).
- `df.abs()`: Return a Series/DataFrame with absolute numeric value of each element.
- `df.all()`: Return whether all elements are True.
- `df.any()`: Return whether any element is True.
- `df.count()`: Count non-NA cells for each column or row.
- `df.empty`: Indicator whether DataFrame is empty.
- `df.bool()`: Return the bool of a single element Pandas object.
- `df.kurt()`: Return unbiased kurtosis.

By: Waleed Mousa

- `df.idxmax()`: Return index of first occurrence of maximum value.
- `df.idxmin()`: Return index of first occurrence of minimum value.
- `df.mode()`: Return the mode(s) of the dataset.
- `df.nunique()`: Count distinct observations.
- `df.quantile(q)`: Return value at the given quantile.
- `df.round()`: Round a DataFrame to a variable number of decimal places.
- `df.sem()`: Return unbiased standard error of the mean.
- `df.skew()`: Return unbiased skew.
- `df.to_dict()`: Convert the DataFrame to a dictionary.
- `df.to_string()`: Render a DataFrame to a console-friendly tabular output.