Tajamul Khan

Pandas Cheat Sheet

@Tajamulkhann

Import Export Data

- **pd.read_csv(filename):** Read data from a CSV file.
- **pd.read_table(filename):** Read data from a delimited text file.
- **pd.read_excel(filename):** Read data from an Excel file.
- **pd.read_sql(query, connection_object):** Read data from a SQL table/database.
- **pd.read_json(json_string):** Read data from a JSON formatted string, URL, or file.
- **pd.read_html(url):** Parse an HTML URL, string, or file to extract tables to a list of DataFrames.
- **pd.DataFrame(dict):** Create a DataFrame from a dictionary (keys as column names, values as lists).
- **df.to_csv(filename):** Write to a CSV file.
- **df.to_excel(filename):** Write to an Excel file.
- **df.to_sql(table_nm, connection_object):** Write to a SQL table.
- **df.to_json(filename):** Write to a file in JSON format.

@Tajamulkhann

Inspect Data

- **df.head():** View the first 5 rows of the DataFrame.
- **df.tail():** View the last 5 rows of the DataFrame.
- **df.sample():** View the random 5 rows of the DataFrame.
- **df.shape:** Get the dimensions of the DataFrame.
- **df.info():** Get a concise summary of the DataFrame.
- **df.describe():** Summary statistics for numerical columns.
- **df.dtypes:** Check data types of columns.
- **df.columns:** List column names.
- **df.index:** Display the index range.

@Tajamulkhann

Select Index Data

- **`df['column']`**: Select a single column.
- **`df[['col1', 'col2']]`**: Select multiple columns.
- **`df.iloc[0]`**: Select the first row by position.
- **`df.loc[0]`**: Select the first row by index label.
- **`df.iloc[0, 0]`**: Select a specific element by position.
- **`df.loc[0, 'column']`**: Select a specific element by label.
- **`df[df['col'] > 5]`**: Filter rows where column > 5.
- **`df.iloc[0:5, 0:2]`**: Slice rows and columns.
- **`df.set_index('column')`**: Set a column as the index.

@Tajamulkhann

Cleaning Data

- **df.isnull():** Check for null values.
- **df.notnull():** Check for non-null values.
- **df.dropna():** Drop rows with null values.
- **df.fillna(value):** Replace null values with a specific value.
- **df.replace(1, 'one'):** Replace specific values.
- **df.rename(columns={'old': 'new'}):** Rename columns.
- **df.astype('int'):** Change data type of a column.
- **df.drop_duplicates():** Remove duplicate rows.
- **df.reset_index():** Reset the index.

@Tajamulkhann

Sort Filter Data

- **df.sort_values('col'):** Sort by column in ascending order.
- **df.sort_values('col', ascending=False):** Sort by column in descending order.
- **df.sort_values(['col1', 'col2'], ascending=[True, False]):** Sort by multiple columns.
- **df[df['col'] > 5]:** Filter rows based on condition.
- **df.query('col > 5'):** Filter using a query string.
- **df.sample(5):** Randomly select 5 rows.
- **df.nlargest(3, 'col'):** Get top 3 rows by column.
- **df.nsmallest(3, 'col'):** Get bottom 3 rows by column.
- **df.filter(like='part'):** Filter columns by substring.

@Tajamulkhann

Group Data

- **df.groupby('col'):** Group by a column.
- **df.groupby('col').mean():** Mean of groups.
- **df.groupby('col').sum():** Sum of groups.
- **df.groupby('col').count():** Count non-null values in groups.
- **df.groupby('col')['other_col'].max():** Max value in another column for groups.
- **df.pivot_table(values='col', index='group', aggfunc='mean'):** Create a pivot table.
- **df.agg({'col1': 'mean', 'col2': 'sum'}):** Aggregate multiple columns.
- **df.apply(np.mean):** Apply a function to columns.
- **df.transform(lambda x: x + 10):** Transform data column-wise.

@Tajamulkhann

Merge Join Data

- **pd.concat([df1, df2]):** Concatenate DataFrames vertically.
- **pd.concat([df1, df2], axis=1):** Concatenate DataFrames horizontally.
- **df1.merge(df2, on='key'):** Merge two DataFrames on a key.
- **df1.join(df2):** SQL-style join.
- **df1.append(df2):** Append rows of one DataFrame to another.
- **pd.merge(df1, df2, how='outer', on='key'):** Outer join.
- **pd.merge(df1, df2, how='inner', on='key'):** Inner join.
- **pd.merge(df1, df2, how='left', on='key'):** Left join.
- **pd.merge(df1, df2, how='right', on='key'):** Right join.

@Tajamulkhann

Statistical Operations

- **df.mean()**: Column-wise mean.
- **df.median()**: Column-wise median.
- **df.std()**: Column-wise standard deviation.
- **df.var()**: Column-wise variance.
- **df.sum()**: Column-wise sum.
- **df.min()**: Column-wise minimum.
- **df.max()**: Column-wise maximum.
- **df.count()**: Count of non-null values per column.
- **df.corr()**: Correlation matrix.

@Tajamulkhann

Data Visualization

- **`df.plot(kind='line')`**: Line plot.
- **`df.plot(kind='bar')`**: Vertical bar plot.
- **`df.plot(kind='barh')`**: Horizontal bar plot.
- **`df.plot(kind='hist')`**: Histogram.
- **`df.plot(kind='box')`**: Box plot.
- **`df.plot(kind='kde')`**: Kernel density estimation plot.
- **`df.plot(kind='pie', y='col')`**: Pie chart.
- **`df.plot.scatter(x='c1', y='c2')`**: Scatter plot.
- **`df.plot(kind='area')`**: Area plot.

@Tajamulkhann

Found Helpful ?

Repost

Follow for more!