



Skills
Network

Data Visualization with Python

Cheat Sheet : Data Preprocessing Tasks in Pandas

| Task | Syntax | Description | Example |
|-------------------------------|---|---|---|
| Load CSV data | <code>pd.read_csv('filename.csv')</code> | Read data from a CSV file into a Pandas DataFrame | <code>df_can=pd.read_csv('data.csv')</code> |
| Handling Missing Values | <code>df.dropna()</code> | Drop rows with missing values | <code>df_can.dropna()</code> |
| | <code>df.fillna(value)</code> | Fill missing values with a specified value | <code>df_can.fillna(0)</code> |
| Removing Duplicates | <code>df.drop_duplicates()</code> | Remove duplicate rows | <code>df_can.drop_duplicates()</code> |
| Renaming Columns | <code>df.rename(columns={'old_name': 'new_name'})</code> | Rename one or more columns | <code>df_can.rename(columns={'Age': 'Years'})</code> |
| Selecting Columns | <code>df['column_name']</code> OR <code>df.column_name</code> | Select a single column | <code>df_can.Age</code> OR <code>df_can['Age']</code> |
| | <code>df[['col1', 'col2']]</code> | Select multiple columns | <code>df_can[['Name', 'Age']]</code> |
| Filtering Rows | <code>df[df['column'] > value]</code> | Filter rows based on a condition | <code>df_can[df_can['Age'] > 30]</code> |
| Applying Functions to Columns | <code>df['column'].apply(function_name)</code> | Apply a function to transform values in a column | <code>df_can['Age'].apply(lambda x: x + 1)</code> |

| Task | Syntax | Description | Example |
|--------------------------|--|--|---|
| Creating New Columns | <code>df['new_column'] = expression</code> | Create a new column with values derived from existing ones | <code>df_can['Total'] = df_can['Quantity'] * df_can['Price']</code> |
| Grouping and Aggregating | <code>df.groupby('column').agg({'col1': 'sum', 'col2': 'mean'})</code> | Group rows by a column and apply aggregate functions | <code>df_can.groupby('Category').agg({'Total': 'mean'})</code> |
| Sorting Rows | <code>df.sort_values('column', ascending=True/False)</code> | Sort rows based on a column | <code>df_can.sort_values('Date', ascending=True)</code> |
| Displaying First n Rows | <code>df.head(n)</code> | Show the first n rows of the DataFrame | <code>df_can.head(3)</code> |
| Displaying Last n Rows | <code>df.tail(n)</code> | Show the last n rows of the DataFrame | <code>df_can.tail(3)</code> |
| Checking for Null Values | <code>df.isnull()</code> | Check for null values in the DataFrame | <code>df_can.isnull()</code> |
| Selecting Rows by Index | <code>df.iloc[index]</code> | Select rows based on integer index | <code>df_can.iloc[3]</code> |
| | <code>df.iloc[start:end]</code> | Select rows in a specified range | <code>df_can.iloc[2:5]</code> |
| Selecting Rows by Label | <code>df.loc[label]</code> | Select rows based on label/index name | <code>df_can.loc['Label']</code> |
| | <code>df.loc[start:end]</code> | Select rows in a specified label/index range | <code>df_can.loc['Age':'Quantity']</code> |
| Summary Statistics | <code>df.describe()</code> | Generates descriptive statistics for numerical columns | <code>df_can.describe()</code> |

Cheat Sheet : Plot Libraries

| Library | Main Purpose | Key Features | Programming Language | Level of Customization | Dashboard Capabilities | Types of Plots Possible |
|-------------------|--------------------------|---|----------------------|------------------------|--|--|
| Matplotlib | General-purpose plotting | Comprehensive plot types and variety of customization | Python | High | Requires additional components and customization | Line plots, scatter plots, bar charts, histograms, pie |

| Library | Main Purpose | Key Features | Programming Language | Level of Customization | Dashboard Capabilities | Types of Plots Possible |
|-----------------|--|---|-----------------------|------------------------|--|---|
| Pandas | Fundamentally used for data manipulation but also has plotting functionality | options Easy to plot directly on Panda data structures | Python | Medium | Can be combined with web frameworks for creating dashboards | charts, box plots, heatmaps, etc. Line plots, scatter plots, bar charts, histograms, pie charts, box plots, etc. |
| Seaborn | Statistical data visualization | Stylish, specialized statistical plot types | Python | Medium | Can be combined with other libraries to display plots on dashboards | Heatmaps, violin plots, scatter plots, bar plots, count plots, etc. |
| Plotly | Interactive data visualization | interactive web-based visualizations | Python, R, JavaScript | High | Dash framework is dedicated for building interactive dashboards | Line plots, scatter plots, bar charts, pie charts, 3D plots, choropleth maps, etc. |
| Folium | Geospatial data visualization | Interactive, customizable maps | Python | Medium | For incorporating maps into dashboards, it can be integrated with other frameworks/libraries | Choropleth maps, point maps, heatmaps, etc. |
| PyWaffle | Plotting Waffle charts | Waffle charts | Python | Low | Can be combined with other libraries to display waffle chart on dashboards | Waffle charts, square pie charts, donut charts, etc. |