Introduction to Pandas for Data Analysis

Installation, Data Structures, and Data Preparation

Moussa Doumbia

Howard University

February 5, 2024

Overview of Pandas

- An open-source data analysis and manipulation tool.
- Offers data structures and operations for manipulating numerical tables and time series.
- Ideal for:
  - Data cleaning: Removing duplicates using `df.drop_duplicates()`.
  - Data filling: Replacing NaN with the mean: `df.fillna(df.mean())`.
  - Data normalization: Scaling data: `(df - df.min()) / (df.max() - df.min())`.
  - Statistical analysis: Summary statistics: `df.describe()`.

Installing Pandas

- Prerequisites: Python and pip.
- Installation Command: `pip install pandas`.
- Verifying Installation: `import pandas as pd; print(pd.__version__)`.

Core Components of Pandas

- **Series**: One-dimensional array. Example: `pd.Series([1, 2, 3])`.
- **DataFrame**: Two-dimensional tabular data. Example: `pd.DataFrame('A': [1, 2], 'B': [3, 4])`.

Understanding Series in Pandas

- Creation: From a list or dict: `pd.Series('a': 1, 'b': 2)`.
- Basic Operations:
  - Indexing: `series[0]`.
  - Slicing: `series[:2]`.
  - Appending: `series.append(other_series)`.
  - Deleting: `series.drop(['a', 'b'])`.

Understanding DataFrames in Pandas

- Creation: From a list of dicts: `pd.DataFrame(['A': 1, 'B': 2])`.
- Basic Operations:
  - Selecting: `df['A']`.
  - Adding: `df['D'] = df['A'] + df['B']`.
  - Deleting: `df.drop(columns=['B'])`.

Data Understanding with Pandas

- Descriptive Statistics: `df.describe()` for summary statistics.
- Data Inspection:
  - Viewing top rows: `df.head()`.
  - Viewing bottom rows: `df.tail()`.
  - Dataset info: `df.info()`.
  - Shape of DataFrame: `df.shape`.

Data Preparation with Pandas

- Handling Missing Data:
  - Checking: `df.isnull()`.
  - Filling missing: `df.fillna(method='ffill')`.
  - Dropping missing: `df.dropna()`.
- Data Transformation:
  - Merging: `pd.merge(df1, df2, on='key')`.
  - Joining: `df1.join(df2)`.
  - Concatenating: `pd.concat([df1, df2])`.
  - Reshaping: `df.pivot(index='date', columns='column')`.

Practical Example with Pandas

- Data Analysis Task:
  - Loading data: `df = pd.read_csv('file.csv')`.
  - Inspecting data: `df.head()`, `df.describe()`.
  - Visualizing data: `df['column'].hist()`.
- Insights and actions based on the analysis.

Conclusion Resources

- Recap of the power and flexibility of Pandas.
- Encouragement to apply these tools to real datasets.
- Resources:
  - Official documentation: `https://pandas.pydata.org/pandas-docs/stable/`
  - Tutorials: Kaggle, DataCamp, etc.
  - Community forums: Stack Overflow, GitHub, etc.