Pandas

Modules

Python Modules: Modules are simply python files (.py) which contain python code. This code can define functions, classes, variables etc.

Modules allow us to organize our code by grouping related functionalities, which makes it easier to use and understand.

Writing code into smaller, more manageable pieces will help you
1) debug easier
2) create reusable code
3) make the code more understandable to the end user.

Packages

Python Packages: Python packages are an organized collection of related python modules in a single directory.

- Some of the most widely used modules in finance and data science for data analysis:
  - Pandas
  - Numpy
  - Scipy
  - Scikit-Learn

Numpy

NumPy is a powerful Python library used for numerical and mathematical operations. It stands for "Numerical Python." One of its main features is the array, which is a multidimensional container for holding homogeneous data.

Why use NumPy?

Efficient Data Storage:
- NumPy arrays are more memory-efficient than Python lists, especially for large datasets. They provide a way to store and manipulate numerical data effectively.

Fast Operations:
- NumPy operations are implemented in C, making them much faster than equivalent Python operations. This is crucial for numerical computations.

Mathematical Functions:
- NumPy provides a wide range of mathematical functions that work seamlessly with arrays. This includes operations like mean, median, standard deviation, and more.

Broadcasting:
- NumPy allows operations between arrays of different shapes and sizes through a feature called broadcasting. This makes it easy to perform element-wise operations on arrays.

NumPy is like a supercharged toolbox for numerical operations in Python. It makes working with numerical data more efficient, provides powerful mathematical functions, and simplifies complex operations on arrays. As you delve deeper into data science, machine learning, or any field involving numerical computations, you'll find NumPy to be an invaluable companion.

Official Documentation: https://numpy.org/doc/stable/user/absolute_beginners.html

Key numpy applications...

1. Arrays:

- The fundamental building block of NumPy is the array. It can be 1D (like a list), 2D (like a table), or even higher dimensional.

2. Indexing and Slicing:

- You can access elements in an array using indexing, just like in Python lists. Slicing allows you to extract portions of the array.

3. Mathematical Operations:

- NumPy provides a wide range of mathematical operations that can be performed element-wise on arrays

4. Broadcasting:

- Broadcasting allows you to perform operations on arrays of different shapes and sizes.
- Broadcasting in NumPy is a powerful feature that allows operations between arrays of different shapes and sizes without the need for explicit expansion or duplication of data. It simplifies operations and makes code more concise

Pandas is a fast, powerful, flexible and easy to use data analysis and manipulation tool.

Building block for doing practical, real world data analysis in Python.

What can we use Pandas for?

- **DataFrame:** Pandas provides a fast and efficient way to work with tables of data, called DataFrames. Think of it like a spreadsheet or a table in a database.
- **Data Input/Output:** You can easily read and write data in different formats like CSV, Excel, SQL databases, and more.
- **Handling Missing Data:** Pandas makes it easy to work with data that might have missing values. It helps you clean up and organize messy data.
- **Reshaping and Pivoting:**You can rearrange and reshape your data easily, making it more suitable for analysis.
- **Slicing and Indexing:** Pandas allows you to select and work with specific parts of your data easily.
- **Size Mutability:** You can change the size of your data structures by adding or removing columns.
- **Grouping and Aggregation:** You can group your data based on certain criteria and perform operations on each group. This is helpful for summarizing or transforming your data.
- **Merging and Joining**: Pandas allows you to combine data from different sources easily.
- **Time Series Operations**: If you're working with time-related data, Pandas has tools for handling time series, such as generating date ranges, shifting dates, and more.
- **Performance:** Pandas is optimized for speed, making it efficient for working with large datasets.

Exercises

Exercise 1: Array Creation

- Create a 1D NumPy array with integers from 1 to 10.
- Create a 2D NumPy array (matrix) with shape (3, 3) and fill it with random values.

Exercise 2: Array Operations

- arr = np.array([1, 2, 3, 4, 5])

  Square each element of the array

- Multiply the corresponding elements of these two arrays:

  arr1 = np.array([1, 2, 3])

  arr2 = np.array([4, 5, 6])

Exercises

Exercise 3: Indexing and Slicing

- Create a 2D NumPy array with shape (4, 4) and fill it with sequential numbers (1 to 16). Extract the last column.
- Extract diagonal elements from:
  - matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

Exercise 2: Mathematical functions

- Create a 2D array with random values and find the sum, mean, and standard deviation along each axis.
- Calculate the mean, median, and standard deviation.
  - arr = np.array([1, 2, 3, 4, 5])

Installation of IDE

- Anaconda
  - https://docs.anaconda.com/free/anaconda/install/index.html
- VSCode:
  - https://dev.to/sourcegraph/installing-and-customizing-visual-studio-code-vs-code-setup-from-scratch-3c62
- Jupyter in VScode
  - https://code.visualstudio.com/docs/datascience/jupyter-notebooks

The End