Python libraries and packages

Overview

Python is a powerful programming language, and the existence of packages and libraries makes it so that it can be used to solve nearly any problem. Packages are essentially collections of Python code that has been written by other developers, allowing you to reuse that code whenever you need to. Libraries, in turn, are simply a collection of packages. By using the "import" keyword in a Python file, you can bring in all the functionality from those packages or libraries that you might need.

For the purposes of this reading, there are three types of libraries. The first are visualization packages, which allow you to create plots and graphics of data. Next, there are operational packages. These allow you to clean and manipulate your data along with performing some more advanced mathematical functions. Finally, there are machine learning packages. Machine learning packages give many functions to help build models from a dataset, along with functionality to examine the model once it has been built.

Again, these categories are not set in stone, and some packages contain functionality in multiple of these categories. Additionally, these are not the only packages you will come across in your learning journey! Python has thousands and thousands of packages publicly available, allowing you to accomplish nearly anything.

Operational Packages

NumPy

NumPy, an abbreviation for Numerical Python, is one of the most important Python packages for scientific computing. At its core, it allows developers to use multidimensional array objects along with various other objects that stem from the multidimensional array, such as matrices. Additionally, it contains a variety of functions that allow for quick operations on said array objects.

```python
import numpy as np

matrix = np.array([[3, 4], [6, 8], [9, 12]])
print(matrix)
```

Run
Reset

Here, the NumPy package is imported and aliased as `np`. A simple two-dimensional array is created with some random values and then printed out.

pandas

Pandas is another widely used package when performing data analysis or science in Python. While not as fundamental as NumPy, it provides high-performance data manipulation and analytical tools that baseline Python does not have. Before the release of pandas, Python was mainly used to refine raw data and prepare it for further analysis. But with pandas, you are able to perform all of the stages of the data analytical process.

```python
import pandas as pd

data = {
    "top_speed": [420, 400, 390, 380],
    "duration": [50, 47, 45, 40]
}

data_frame = pd.DataFrame(data)

data_frame.head()
```

Run
Reset

Here, pandas is imported and aliased as `pd`. A dictionary is defined containing data on various hula-hoopers, specifically their top speed and how long they hooped for. That information is then used to create a pandas dataframe.

Visualization Packages

MatPlotlib

Matplotlib is another very popular library used widely in the data analytics industry. It provides various tools and functions to visualize data in Python. It is cross platform, and makes it very easy to create two dimensional plots from arrays in Python. With only a few lines of code, you can create various types of plots, such as scatter plots, histograms, bar charts and box plots.

```python
import pandas as pd

# Import Matplotlib
import matplotlib.pyplot as plt

data = {
    "top_speed": [420, 400, 390, 380],
    "duration": [50, 47, 45, 40]
}

data_frame = pd.DataFrame(data)

# Create plot using Matplotlib
plt.scatter(data_frame["top_speed"], data_frame["duration"])

# Display scatter plot
```

Run

```
# Display scatter plot
plt.show()
```

Reset

Here, the pyplot module of the matplotlib library is imported and aliased as `plt`. To create a scatter plot of the hula-hooper data, only a single line of code is needed!

Seaborn

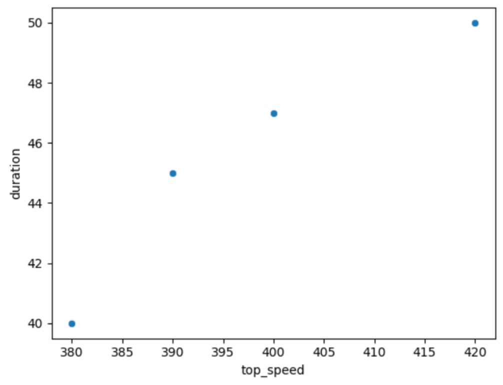
Seaborn is another visualization package for Python, developed specifically with the intent to make it easier to create beautiful visualizations. The package is actually an extension of Matplotlib, using it as a foundation. Because of this, it is more meant to compliment Matplotlib rather than completely replace it. Whereas Matplotlib can get very complex very quickly as you customize your plots, Seaborn is focused on keeping things simple.

Note: The following code block is not interactive.

```
import seaborn as sns

data = {
    "top_speed": [420, 400, 390, 380],
    "duration": [50, 47, 45, 40]
}

sns.scatterplot(data=data_frame, x="top_speed", y="duration")
```



Notice how this looks almost exactly the same as the matplotlib code snippet. Remember, seaborn is entirely built on top of matplotlib. Seaborn is just there to give some extra functionality.

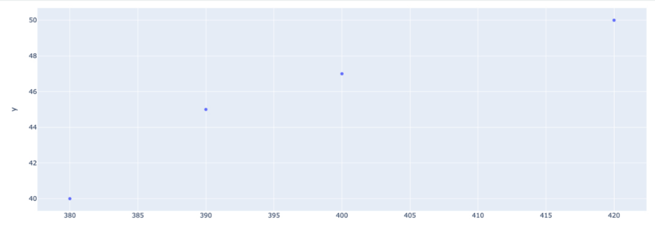
Plotly

Plotly is yet another open-source library that is used to create data visualizations in Python. With it, you are able to make very beautiful visualizations very easily that are accessible to a wide range of audiences. It supports nearly 50 unique types of charts that are commonly used in many different industries such as finance, geography, and statistics.

Note: The following code block is not interactive.

```
import plotly.express as px

px.scatter(x=data_frame["top_speed"], y=data_frame["duration"])
```



This is another plot of the same data, however this plot arguably looks a bit more visually appealing out of the gate. This same graph is achievable with the other packages, but with more work.

Machine Learning Packages

scikit-learn

Scikit-learn, sometimes referred to as Sklearn, is one of the most robust libraries available for machine learning in Python. It contains a host of tools that can be used for classification and regression problems, statistical modeling, clustering, and dimensionality reduction. It is actually built on top of some of the packages that you've already learned about and used, such as NumPy, pandas, and Matplotlib.

Key takeaways

Knowing the packages that are available to you is essential to successfully creating models or analyzing data. By continuing to develop knowledge about the tools that are available to you, you are able to know "what to reach for" no matter what situation you find yourself in.

Python Packages and Libraries

Operational Packages

- NumPy
  - Allows for more mathematical operations in Python, provides functions for array-like objects, etc
- pandas
  - Creation of data frames, analyzing data, cleaning data, manipulating data, performing efficient operations on large data sets

Visualization Packages

- MatPlotLib
  - Easy-to-learn difficult-to-master graphing library for Python. Great for quick, exploratory graphs
- Seaborn
  - Built on top of matplotlib, allows for easier customization of plots compare to matplotlib
- Plotly
  - Easy to create beautiful, presentation quality plots and graphs. Lots of built in functionality and can have interactive elements.

Machine Learning Packages

- scikit-learn
  - Provides functionality for a host of machine learning models and analytical tools.

Go to next item ✓ Completed

Like Dislike Report an issue