**Libraries for Data Science**

In this video, we will review several data science libraries. Libraries are a collection of functions and methods that enable you to perform a wide variety of actions without writing the code yourself. We will focus on Python libraries: **Scientific Computing Libraries in Python Visualization Libraries in Python High-Level Machine Learning and Deep Learning Libraries** **– “High-level” simply means you don’t have to worry about details**, although this makes it difficult to study or improve Deep Learning Libraries in Python Libraries used in other languages Libraries usually contain built-in modules providing different functionalities that you can use directly; these are sometimes called “frameworks.” There are also extensive libraries, offering a broad range of facilities.

**Pandas offers data structures and tools for effective data cleaning, manipulation, and analysis**. It provides tools to work with different types of data**. The primary instrument of Pandas is a two-dimensional table consisting of columns and rows. This table is called a “DataFrame” and is designed to provide easy indexing so you can work with your data.**

**NumPy libraries are based on arrays, enabling you to apply mathematical functions to these arrays. Pandas is actually built on top of NumPy Data visualization methods are a great way to communicate with others and show the meaningful results of analysis**. These libraries enable you to create graphs, charts and maps.

**The Matplotlib package is the most well-known library for data visualization, and it’s excellent for making graphs and plots.** The graphs are also highly customizable. Another high-level visualization library, Seaborn, is based on matplotlib.

**Seaborn makes it easy to generate plots like heat maps, time series, and violin plots**.

**For machine learning, the Scikit-learn library contains tools for statistical modeling, including regression, classification, clustering and others**. It is built on NumPy, SciPy, and matplotlib, and it’s relatively simple to get started. For this high-level approach, you define the model and specify the parameter types you would like to use.

**For deep learning, Keras enables you to build the standard deep learning model**. Like Scikit-learn, the high-level interface enables you to build models quickly and simply. It can function using graphics processing units (GPU), but for many deep learning cases a lower-level environment is required.

**TensorFlow** is a low-level framework used in large scale production of deep learning models. It’s designed for production but can be unwieldy for experimentation.

**Pytorch is used for experimentation, making it simple for researchers to test their ideas** .

**Apache Spark is a general-purpose cluster-computing framework that enables you to process data using compute clusters**. This means that you process data in parallel, using multiple computers simultaneously. The Spark library has similar functionality as Pandas Numpy Scikit-learn Apache Spark data processing jobs can use Python R Scala, or SQL There are many libraries for Scala, which is predominately used in data engineering but is also sometimes used in data science.

Let’s discuss some of the libraries that are complementary to **Spark Vegas** is a Scala library for statistical data visualizations. With Vegas, you can work with data files as well as Spark DataFrames. For deep learning, you can use BigDL.

**R has built-in functionality for machine learning and data visualization**, but there are also several complementary libraries: ggplot2 is a popular library for data visualization in R. You can also use libraries that enable you to interface with Keras and TensorFlow. R has been the de-facto standard for open-source data science but it is now being superseded by Python.

















