Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 1

Aim: Introduction to Data analytics libraries in Python and R.

Objective- Understand the use of Python and R, To effectively use libraries for data science.

Description:

Why Choose Python?

Python is a general-purpose, open-source programming language used in various software domains, including data science, web development, and gaming.

Launched in 1991, Python is one of the most popular programming languages in the world, occupying the top position in several programming language popularity indices, such as the TIOBE Index and the PYPL Index.

One of the reasons for the worldwide popularity of Python is its community of users. Python is backed by a vast community of users and developers who ensure the smooth growth and improvement of the language, as well as the continuous release of new libraries designed for all kinds of purposes.

Python is an easy language to read and write due to its high similarity with human language. In fact, high readability and interpretability are at the heart of the design of Python. For these reasons, Python is often cited as a go-to programming language for newcomers with no coding experience.

Over time, Python has been gaining popularity in the field of data science thanks to its simplicity and the endless possibilities provided by the hundreds of specialized libraries and packages that support any kind of data science task, such as data visualization, machine learning, and deep learning.

Why Choose R?

R is an open-source programming language specifically created for statistical computing and graphics.

Since its first launch in 1992, R has been widely adopted in scientific research and academia. Today, it remains one of the most popular analytics tools used in both traditional data analytics and the rapidly-evolving field of business analytics. It ranks 11th and 7th position in the **TIOBE** Index and the **PYPL** Index, respectively.

Designed with statisticians in mind, with R, you can use complex functions within a few lines of code. All kinds of statistical tests and models are readily available and easily used, such as linear modeling, non-linear modeling, classifications, and clustering.

The extensive possibilities R offers are mostly due to its huge community. It has developed one of the richest collections of data-science-related packages. All of them are available via the Comprehensive R Archive Network (**CRAN**).

Another feature that makes R particularly remarkable is the power to generate quality reports with support for data visualization and its available frameworks to create interactive web applications. In this sense, R is widely considered the best tool for making beautiful graphs and visualizations

R vs Python: Key Differences

Purpose

While Python and R were created with different purposes –Python as a general-purpose programming language and R for statistical analysis–nowadays, both are suitable for any data science task. However, Python is considered a more versatile programming language than R, as it's also extremely popular in other software domains, such as software development, web development, and gaming.

Type of Users

As a general-purpose programming language, Python is the standard go-to choice for software developers breaking into data science. Plus, Python's focus on productivity makes it a more suitable tool to build complex applications.

By contrast, R is widely used in academia and certain sectors, such as finance and pharmaceuticals. It is the perfect language for statisticians and researchers with limited programming skills.

Learning curve

Python's intuitive syntax is considered one of the closest programming languages to English. This makes it a very good language for new programmers, with a smooth and linear learning curve. Although R is designed to run basic data analysis easily and within minutes, things get harder with complex tasks, and it takes more time for R users to master the language.

Overall, Python is considered a good language for beginner programmers. R is easier to learn when you start out, but the intricacies of advanced functionalities make it more difficult to develop expertise.

Popularity

Although new programming languages, like **Julia**, are recently gaining momentum in data science, Python and R remain the absolute kings in the discipline.

However, in terms of popularity –always a very slippery concept– the differences are striking. Python has consistently outranked R, especially in recent years. Python ranks first in several programming language popularity indexes. This is due to the widespread use of Python in multiple software domains, including data science. By contrast, R is mostly employed in data science, academia, and certain sectors.

Common Libraries

Both Python and R have robust and extensive ecosystems of packages and libraries specifically designed for data science. Most packages in Python are hosted in the Python Package Index (**PyPi**), whereas **R packages** are normally stored in the Comprehensive R Archive Network (**CRAN**).

Below you can find a list of some of the most popular data science libraries in R and Python.

R packages:

- **dplyr**: It is a data manipulation library for R.
- **tidyr**: a great package that will help you get your data clean and tidy.
- **ggplot2**: the perfect library for visualizing data.
- **Shiny**: It is the ideal tool for creating interactive web apps directly from R.
- **Caret**: one of the most important libraries for machine learning in R.

Python packages:

- **NumPy**: provides a large collection of functions for scientific computing.
- **Pandas**: perfect for data manipulation.
- **Matplotlib**: the standard library for data visualization.
- **Scikit-learn**: is a library in Python that provides many machine learning algorithms.
- **TensorFlow**: a widely used framework for deep learning.

Common IDEs

An IDE, or Integrated Development Environment, enables programmers to consolidate the different aspects of writing a computer program. They are powerful interfaces with integrated capabilities that allow developers to write code more efficiently.

In Python, the most popular IDEs in data science are Jupyter Notebooks and its modern version, JupyterLab, as well as Spyder.

As for R, the most commonly used IDE is RStudio. Its interface is organized so that the user can view graphs, data tables, R code, and output all at the same time.

Python vs R: A Comparison

	R	**Python**
Purpose	Very popular in academia and research, finance and data science	Well-suited for many programming domains, including data science, web development, software development, and gaming
First Release	1993	1991
Type of Language	General-purpose programming language	General-purpose programming language
Open Source?	Yes	Yes

Ecosystem	Nearly 19,000 packages available in the Comprehensive R Archive Network (**CRAN**)	+300,000 available packages in the Python Package Index (**PyPi**)
Ease of Learning	R is easier to learn when you start out, but gets more difficult when using advanced functionalities.	Python is a beginner-friendly language with English-like syntax.
IDE	RStudio. Its interface is organized so that the user can view graphs, data tables, R code, and output all at the same time.	Jupyter Notebooks and its modern version, JupyterLab, and Spyder.
Advantages	· Widely considered the best tool for making beautiful graphs and visualizations. · Has many functionalities for data analysis. · Great for statistical analysis.	· General-purpose programming languages are useful beyond just data analysis. · Has gained popularity for its code readability, speed, and many functionalities. . · Has high ease of deployment and reproducibility.

Disadvantages	· More difficult to learn for people with no software development background. · Limited user community compared to Python · R is considered a computationally slower language compared to Python, especially if the code is written poorly. · Finding the right library for your task can be tricky, given the high number of packages available in CRAN	· Weak performance with huge amounts of data · Poor memory efficiency · Python does not have as many libraries for data science as R. · Python requires rigorous testing as errors show up in runtime. · Visualizations are more convoluted in Python than in R, and results are not as eye pleasing or informative.
Trends	11th in TIOBE and 7th in PYPL (December 2022)	1th in TIOBE and 1th in PYPL (December 2022)

Attach Libraries you searched in Lab session-

R programming Libraries

1. Dplyr

Dplyr is mainly used for data manipulation in R. Dplyr is actually built around these 5 functions. These functions make up the majority of the data manipulation you tend to do. You can work with local data frames as well as with remote database tables.

2. Ggplot2

Ggplot2 is the one of the best library for data visualization in R. The ggplot2 library implements a "grammar of graphics" (Wilkinson, 2005). This approach gives us a coherent way to produce visualizations by expressing relationships between the attributes of data and their graphical representation.

3. Esquisse

This package has brought the most important feature of Tableau to R. Just drag and drop, and get your visualization done in minutes. This is actually an enhancement to ggplot2. This addin allows you to interactively explore your data by visualizing it with the ggplot2 package. It allows you to draw bar graphs, curves, scatter plots, histograms, then export the graph or retrieve the code generating the graph.

4. Shiny

This is a very well known package in R. When you want to share your stuff with people around you and make it easier for them to understand and explore it visually, you can use shiny. It's a Data Scientist's best friend. Shiny makes it easier to build interactive web apps. You can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards. You can also extend your Shiny apps with CSS themes, htmlwidgets, and JavaScript actions.

5. Lubridate

This library serves its purpose really well. It's mainly used for data wrangling. It makes the dealing of date-time easier in R. You can do everything you ever wanted to do with date arithmetic using this library, although understanding & using available functionality can be somewhat complex here. When you are analyzing time series data and want to aggregate the data by month then you can use floor_date from lubridate package, it gets your work done quite easily. It has wide range of functions.

6. Knitr

This package is used for dynamic report generation in R. The purpose of knitr is to allow reproducible research in R through the means of Literate Programming. This package also enables integration of R code into LaTeX, Markdown, LyX, HTML, AsciiDoc, and reStructuredText documents. You can add R to a markdown document and easily generate reports in HTML, Word and other formats. A must-have if you're interested in reproducible research and automating the journey from data analysis to report creation.

7. Mlr

This package is absolutely incredible in performing machine learning tasks. It almost has all the important and useful algorithms for performing machine learning tasks. It can also be termed as the extensible framework for classification, regression, clustering, multi-classification and survival analysis. It also has filter and wrapper methods for feature selection. Another thing is most operations performed here can be parallelized.

8. DT

It is a wrapper of javascript library DataTables. It is used for data display, you can display R matrices and data frames as interactive HTML tables. You can create a sortable table with minimum amount of code using this library, actually you can create a sortable, searchable table in just one line of code. You can also style your table. DataTables also provides filtering, pagination, sorting, and many other features in the tables.

9. RCrawler

RCrawler is a contributed R package for domain-based web crawling and content scraping. It adds the functionality of crawling that Rvest package lacks. RCrawler can crawl, parse, store pages, extract contents, and produce data that can be directly employed for web content mining applications. The process of a crawling operation is performed by several concurrent processes or nodes in parallel, so it's recommended to use 64bit version of R.

10. Caret

Caret stands for classification and regression training. One of the primary tools in the package is the train function which can be used to. evaluate, using re-sampling, the effect of model tuning parameters on performance. Caret has several functions that attempt to streamline the model building and evaluation process, as well as feature selection and other techniques. This package alone is all you need to know for solve almost any supervised machine learning problem. It provides a uniform interface to several machine learning algorithms and standardizes various other tasks such as Data splitting, pre-processing, feature selection, variable importance estimation etc.

Python Libraries

1. TensorFlow
The first in the list of python libraries for data science is TensorFlow. TensorFlow is a library for high-performance numerical computations with around 35,000 comments and a vibrant community of around 1,500 contributors. It's used across various scientific fields. TensorFlow is basically a framework for defining and running computations that involve tensors, which are partially defined computational objects that eventually produce a value.

2. SciPy
SciPy (Scientific Python) is another free and open-source Python library for data science that is extensively used for high-level computations. SciPy has around 19,000 comments on GitHub and an active community of about 600 contributors. It's extensively used for scientific and technical computations, because it extends NumPy and provides many user-friendly and efficient routines for scientific calculations.

3. NumPy
NumPy (Numerical Python) is the fundamental package for numerical computation in Python; it contains a powerful N-dimensional array object. It has around 18,000 comments on GitHub and an active community of 700 contributors. It's a general-purpose array-processing package that provides high-performance multidimensional objects called arrays and tools for working with them. NumPy also addresses the slowness problem partly by providing these multidimensional arrays as well as providing functions and operators that operate efficiently on these arrays.

4. Pandas
Next in the list of python librabries is Pandads. Pandas (Python data analysis) is a must in the data science life cycle. It is the most popular and widely used Python library for data science, along with NumPy in matplotlib. With around 17,00 comments on GitHub and an active community of 1,200 contributors, it is heavily used for data analysis and cleaning. Pandas provides fast, flexible data structures, such as data frame CDs, which are designed to work with structured data very easily and intuitively.

5. Matplotlib
Matplotlib has powerful yet beautiful visualizations. It's a plotting library for Python with around 26,000 comments on GitHub and a very vibrant community of about 700 contributors. Because of the graphs and plots that it produces, it's extensively used for data visualization. It also provides an object-oriented API, which can be used to embed those plots into applications.

6. Keras
Similar to TensorFlow, Keras is another popular library that is used extensively for deep learning and neural network modules. Keras supports both the TensorFlow and Theano backends, so it is a good option if you don't want to dive into the details of TensorFlow.

7. Scikit-learn
Next in the list of the top python libraries for data science comes Scikit-learn, a machine learning library that provides almost all the machine learning algorithms you might need. Scikit-learn is designed to be interpolated into NumPy and SciPy.

8. PyTorch

Next in the list of top python libraries for data science is PyTorch, which is a Python-based scientific computing package that uses the power of graphics processing units. PyTorch is one of the most commonly preferred deep learning research platforms built to provide maximum flexibility and speed.

9. Scrapy
The next known python libraries for data science is Scrapy. Scrapy isone of the most popular, fast, open-source web crawling frameworks written in Python. It is commonly used to extract the data from the web page with the help of selectors based on XPath.

10. LightGBM
The LightGBM Python library is a popular tool for implementing gradient-boosting algorithms in data science projects. It provides a high-performance implementation of gradient boosting that can handle large datasets and high-dimensional feature spaces.

Conclusion-

R Programming Libraries:

R offers a rich ecosystem of libraries tailored for statistical computing and data analysis. Packages like dplyr and ggplot2 streamline data manipulation and visualization, while caret and randomForest facilitate machine learning tasks. The language's strength lies in its specialized packages for statistical modeling and exploration.

Python Libraries:

Python, a versatile programming language, boasts a vast array of libraries for diverse domains. NumPy and Pandas excel in data manipulation, Matplotlib and Seaborn in visualization, and scikit-learn in machine learning. Python's versatility extends beyond data science, making it a go-to language for various applications.