## 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.<br>· Has many functionalities for data analysis.<br>· Great for statistical analysis. | · General-purpose programming languages are useful beyond just data analysis.<br>· Has gained popularity for its code readability, speed, and many functionalities. .<br>· Has high ease of deployment and reproducibility. |

| | | |
|---|---|---|
| Disadvantages | · More difficult to learn for people with no software development background.<br>· Limited user community compared to Python<br>· R is considered a computationally slower language compared to Python, especially if the code is written poorly.<br>· 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<br>· Poor memory efficiency<br>· Python does not have as many libraries for data science as R.<br>· Python requires rigorous testing as errors show up in runtime.<br>· 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 Libraries:

1. Dplyr:

Use: Data manipulation. Renowned for its role in transforming and manipulating data, dplyr simplifies tasks like filtering, grouping, summarizing, and joining datasets. Its functions enhance the efficiency and ease of data exploration, making it a cornerstone library for analysts and data scientists engaged in R programming.

2. Ggplot2:

Use: Data visualization. ggplot2 is a powerful library for crafting compelling and informative data visualizations. Employing a grammar of graphics approach, it allows users to create a diverse range of static and interactive plots, aiding in the effective communication of data patterns and insights.

3. Tidyr:

Use: Data tidying. tidyr plays a crucial role in preparing and cleaning datasets for analysis. Its functions facilitate the reshaping and organizing of messy data, ensuring that information is structured in a way that is conducive to accurate and insightful data exploration.

4. Caret:

Use: Machine learning. caret is a versatile package that simplifies the process of developing and comparing machine learning models. Providing a unified interface for various algorithms, it streamlines the training and testing phases, enabling data scientists to efficiently implement and assess predictive models for different tasks.

5. Stringr:

Use: String manipulation. String manipulation is made seamless with stringr, a library designed for handling and transforming character strings. Its functions simplify text processing tasks, enabling users to efficiently work with textual data and extract valuable insights.

6. Data.table:

Use: Data manipulation. data.table extends the capabilities of the traditional data.frame, offering fast and memory-efficient tools for handling large datasets. This makes it an invaluable library for analysts dealing with substantial amounts of data, enhancing both speed and efficiency in data processing tasks.

7. Shiny:

Use: Web application development. Shiny revolutionizes the creation of interactive web applications directly from R scripts. This library enables data scientists and analysts to build user-friendly interfaces for sharing data insights, fostering collaboration, and expanding the accessibility of data-driven findings to a broader audience.

8. tidyverse:

Use: Data analysis. tidyverse is a comprehensive collection of R packages, including dplyr, ggplot2, and tidyr. It promotes a consistent and efficient workflow for data analysis and visualization, providing a unified and coherent set of tools for seamless exploration and interpretation of datasets.

9. Rvest:

Use: Web scraping. rvest is a valuable library for extracting data from websites. By providing functions to navigate HTML structures, it simplifies the process of web scraping, allowing users to retrieve and organize information from web pages efficiently.

10. ROCR:

Use: Model evaluation. ROCR is an essential library for evaluating the performance of classification models. Its tools enable the creation of receiver operating characteristic (ROC) curves, aiding in the visual and quantitative assessment of a model's accuracy and effectiveness in making predictions.

Python Libraries:

1. NumPy:

Numerical computing. NumPy provides arrays and mathematical functions for efficient scientific computing, facilitating tasks like linear algebra and statistical analysis. It's essential for data manipulation and numerical operations in Python.

2. Pandas:

Data manipulation. Pandas offers DataFrames for seamless handling and analysis of structured data. It streamlines data cleaning, exploration, and transformation, playing a key role in data science workflows.

3. Matplotlib:

Data visualization. Matplotlib is a versatile plotting library enabling the creation of static, animated, and interactive visualizations. It supports the effective communication of data insights through a wide range of graphs and charts.

4. Seaborn:

Statistical data visualization. Seaborn, built on Matplotlib, simplifies the creation of aesthetically pleasing statistical graphics. It enhances the representation of data patterns and relationships in a visually appealing manner.

5. Scikit-learn:

Machine learning. Scikit-learn provides a comprehensive set of tools for implementing machine learning algorithms. It offers functionalities for data preprocessing, model training, and evaluation, contributing to efficient model development.

6. TensorFlow:

Deep learning. TensorFlow is an open-source deep learning library widely employed for building and training neural network models. Its applications span image and speech recognition, natural language processing, and more.

7. PyTorch:

Deep learning. PyTorch is a dynamic deep learning library favored for its flexibility. It's intuitive for researchers and developers, making it easy to build and experiment with neural network models efficiently.

8. NLTK (Natural Language Toolkit):

Natural language processing (NLP). NLTK provides tools for working with human language data, including tokenization, stemming, and part-of-speech tagging. It facilitates the development of NLP applications in Python.

9. Django:

Web development. Django, a high-level web framework, simplifies web application development with features like an object-relational mapper (ORM) and a built-in admin interface. It fosters the creation of scalable and maintainable web applications.

10. Flask:

Web development. Flask, a lightweight web framework, offers simplicity and flexibility for building small to medium-sized web applications. It makes it easy to get started with web development in Python and provides a solid foundation for web projects.

**Conclusion-**

R Libraries:

Understanding and leveraging key R libraries is essential for proficient data analysis, visualization, machine learning, and specialized tasks like web scraping. dplyr, ggplot2, and others streamline processes, enhancing efficiency and empowering users to extract meaningful insights from diverse datasets, fostering advanced analytics and informed decision-making.

Python Libraries:

After exploring Python libraries, it's evident that they play a pivotal role in enhancing the language's capabilities. Libraries like NumPy, Pandas, Matplotlib, and Scikit-learn empower Python for scientific computing, data analysis, visualization, and machine learning. The extensive library ecosystem underscores Python's versatility and popularity in diverse domains.