Experiment No.1
Getting introduced to data analytics libraries in Python and R.
Date of Performance:
Date of Submission:

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.

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.
- **22plot2:** 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 modem version, JupyterLab, as well as Spyder.

Attach Libraries you searched in Lab session-

Python Libaries:

1. **NumPy**
   High-performance numerical computing library with support for large, multi-dimensional arrays and matrices.
2. **Pandas**
   Data manipulation library providing data structures like DataFrame for efficient data analysis and manipulation.
3. **Matplotlib**
   2D plotting library for creating static, animated, and interactive visualizations in Python.
4. **Scikit-learn**
   Machine learning library that provides simple and efficient tools for data mining and data analysis.
5. **TensorFlow**
   Open-source machine learning framework with a focus on deep learning and high-performance numerical computations.
6. **Beautiful Soup**
   Library for pulling data out of HTML and XML files, with Pythonic idioms for web scraping.
7. **Requests**
   HTTP library for making web requests in Python, simplifying access to web resources.
8. **NLTK (Natural Language Toolkit)**
   Library for working with human language data, including tokenization, stemming, and part-of-speech tagging.
9. **Seaborn**
   Statistical data visualization library based on Matplotlib, providing a high-level interface for drawing attractive and informative statistical graphics.
10. **Flask**
    Lightweight web application framework for Python, suitable for building small to medium-sized web applications.

R Libraries:

1. **dplyr**
   Data manipulation package providing a grammar of data manipulation with a set of verbs for common tasks.
2. **ggplot2**
   Implementation of the Grammar of Graphics, offering a powerful and flexible system for creating static and interactive visualizations.

3. **tidyr**

   Package for cleaning and tidying data, with functions for reshaping and restructuring datasets.

4. **caret**

   Comprehensive package for classification and regression training, providing a consistent interface to various machine learning algorithms.

   Shiny

   Web application framework for R, allowing data scientists to create interactive and dynamic dashboards.

5. **Rmarkdown**

   Package for creating dynamic documents with R, integrating code, results, and narrative text in a single document.

6. **lubridate**

   Package for handling date-time data in R, making it easier to work with dates and times.

7. **ggvis**

   Interactive grammar of graphics for creating interactive plots and visualizations.

8. **rvest**

   Web scraping package for R, enabling the extraction of data from HTML web pages.

9. **stringr**

   Package for working with strings in R, providing simple and consistent functions for common string operations.

Conclusion-

In conclusion, both Python and R offer diverse and powerful libraries for various data science, machine learning, and web development tasks, catering to the preferences and requirements of data scientists and analysts. The choice between the two often depends on the specific needs of a project and individual preferences. Many practitioners find value in using both languages within their workflow.