Paper DV08

Approachable and Scalable Web Application Solutions using Shiny for Python and R Shiny in Pharmaceutical Industry

Vikash Jain, Genpro Research, Hillsborough, New Jersey, USA
Denny Sebastian, Genpro Research, Thiruvananthapuram, India

ABSTRACT

Shiny is an R package that helps to make interactive web applications and dashboards. Now the creators of Shiny have introduced it for Python also. Since the library is still in alpha, it has several issues, and further enhancements can be expected. In this context, we explore the possibilities of Py shiny library, figure out the unknown bugs and contribute to the enhancement of python shiny framework.

In fact, python already has an option to build web applications since 2017 with the help of Dash framework. So, a comparison of the capabilities of python Dash framework and the R shiny framework can help to improve the Py shiny library. Since R Shiny was launched a decade ago in July 2012, it has a long development history that may not be incorporated in Py shiny. Thus, a comparison of Py shiny v/s R shiny will be significant. Data visualization is always an inevitable factor in dashboards, so visualization possibilities in R and python will also be explained in the paper.

INTRODUCTION

Web dashboards becomes an increasingly popular tool in the pharmaceutical industry for analyzing and presenting complex data. These dashboards can be used to provide a high-level view of critical information, such as drug development timelines, clinical trial data, regulatory compliance, and market trends. Also, web dashboards play a critical role in the pharmaceutical and biotechnology industries by providing real-time data, visualizations, and insights to support informed decision-making and drive operational efficiency. Here are some of the common ways in which web dashboards are used in the pharmaceutical industry:

- Clinical trial management: Web dashboards can be used to track the progress of clinical trials, monitor patient enrollment and assess the effectiveness of treatments.
- Sales and marketing: Sales and marketing teams can use dashboards to monitor sales trends, track the performance of marketing campaigns, and analyze the impact of promotions on product sales.
- Supply chain management: Dashboards can be used to monitor the supply chain of pharmaceutical products, track inventory levels, and ensure that the right products are being shipped to the right places at the right time.
- Regulatory compliance: Dashboards can help organizations track and manage compliance with regulations such as FDA requirements, ensuring that they are meeting all the necessary standards.
- Business intelligence: Dashboards can be used to analyze large amounts of data from various sources to gain insights into key business metrics, such as customer behavior and financial performance.

There are several tools used to develop dashboards in the pharmaceutical industry, some of the most popular are:

- Tableau: It is a data visualization tool that helps to create interactive dashboards, reports, and charts.
- Power BI: It is a cloud-based business analytics service provided by Microsoft, used to create interactive data visualizations and dashboards.
- R Shiny: R Shiny is an open-source package in the R programming language that provides a framework for building web-based interactive applications. It allows users to create dynamic dashboards and web apps with interactive graphics, tables, and other user-friendly features

Posit launched Python Shiny to provide a simple and powerful platform for creating interactive web-based applications and data visualizations. Shiny is a popular open-source web framework that allows developers to build web-based applications using R and Python. By launching Python Shiny, Posit aimed to make it easier for developers who are familiar with Python to create interactive data-driven applications without having to learn a new language or

framework. This allowed users to take advantage of the power and simplicity of Python, combined with the interactivity and visualization capabilities of Shiny, to create engaging and effective applications for data analysis and exploration.

PYTHON DASHBOARD FRAMEWORKS

Python dashboard frameworks is a collection of tools and libraries used to develop and design interactive dashboards using the Python programming language. These frameworks allow for the creation of data visualization, data exploration, and data analysis in a web-based interface. The dashboards produced with these frameworks can be embedded in websites, shared with others, and interacted with by end-users. Examples of such frameworks include Plotly Dash, Dash by Plotly, and Shiny.

- Plotly-Dash: A high-level framework for building dashboards and web applications.
- Dash: A web-based framework for building data dashboards and applications.
- Streamlit: A framework for building data dashboards and applications in Python, based on the Flask web framework.
- Flask Framework: A framework for building responsive, modern dashboards and web applications in Flask.
- Shiny Framework: A framework for building data dashboards in R, but can also be used with Python.

PYTHON DASH FRAMEWORKS

The Plotly-Dash and Dash are similar in that they are both open-source Python frameworks for building web applications. The main difference between them is the underlying data visualization library. Plotly-Dash is built on top of the Plotly library, which provides interactive and animated charts and graphs, while Dash is built on top of the Flask web framework and uses a more basic set of plotting libraries.

One of the main advantages of Python Dash is its simplicity. With just a few lines of code, you can create interactive plots, graphs, and dashboards. It also has a rich set of components that can be easily customized, such as dropdown menus, sliders, and buttons. This makes it easy for developers to build interactive applications without having to spend a lot of time on the front-end development. Dash apps are built and published in the Web, so the full power of CSS is available. But we don't have to write any Javascript or HTML. Python Dash is a powerful and easy-to-use framework, but the framework is not well-suited for building large-scale applications due to its limited scalability. While it may not be suitable for all applications, it's a valuable tool for developers looking to quickly create interactive data visualizations.

DASH AND R-SHINY FRAMEWORKS

Python Dash and R Shiny, both frameworks offer a wide range of features and capabilities, including data visualization, data manipulation, and interactive user interface elements. R Shiny is designed to be easy to use and offers a wide range of interactive elements, including sliders, checkboxes, and dropdown menus. Shiny also offers a number of pre-built visualizations and plotting libraries, such as ggplot2, plotly, which can be easily integrated into a dashboard or application. R Shiny may be a better fit for projects that require heavy statistical analysis or complex data visualizations, while Python Dash may be more suitable for projects that require machine learning or data manipulation.

Python Dash offers a wide range of customization options, including the ability to build custom components and integrate with other Python libraries. R Shiny also offers a number of customization options, but may not be as flexible as Python Dash in terms of integration with other libraries or custom component development. In terms of performance, both Python Dash and R Shiny offer fast and efficient web application development. However, Python Dash may have an edge in terms of performance due to its integration with the Plotly.js library, which is optimized for data visualization and interaction.

FEATURES OF PYTHON SHINY FRAMEWORKS

Python shiny framework will look familiar If we are familiar with R Shiny framework. Mainly, Python doesn't use $ to access methods and properties, and also uses function decorators for rendering. The syntax is bit different from R shiny, but it shouldn't feel like a whole new framework. Py Shiny also offers a range of built-in widgets and layouts for creating user interfaces. These include buttons, checkboxes, dropdown menus, and more. Developers can customize the appearance and functionality of these widgets to fit the needs of their application. The backend of Shiny for Python is the uvicorn web server. FastAPI, one of the most well-liked backend frameworks for Python, is powered by it. Py Shiny apps will be quick and scalable if implemented correctly because it is incredibly reliable and performant.

Py Shiny also has strong integration with other Python libraries, such as NumPy and Pandas, which can be used to perform data analysis and manipulation. This makes it easy to create data-driven web applications that can visualize and analyze data in real-time. Py shiny applications can be deployed to a server and accessed by users through a web browser, making it easy to share and collaborate on projects.

MIGRATION OF R PROGRAMMING

As the healthcare industry continues to evolve, the importance of data analytics has become increasingly evident. This has led to a shift in the use of statistical programming languages in the clinical setting, with a growing trend towards the use of R programming rather than SAS. The migration of R programming in the clinical setting is a growing trend that is driven because R is an open-source language, is freely available to anyone who wants to use it. This has made it a popular choice for clinical researchers and analysts who are looking for a cost-effective way to analyze their data. Additionally, R has a large community of users and contributors, making it easier to find help and support when needed.

Another factor contributing to the migration of R in clinical programming is its versatility. Unlike SAS, which is designed primarily for data analysis and management, R has a wide range of applications, including data visualization, machine learning, and predictive modeling. This makes it a more versatile tool for clinical researchers and analysts who are looking to explore different aspects of their data. In addition, R is a more flexible language than SAS, allowing users to create custom programs and workflows to meet their specific needs. This can be particularly useful in the clinical setting, where researchers may need to manipulate or analyze data in unique ways.

Despite these advantages, there are also some challenges associated with the use of R in the clinical setting. For example, some users may find it difficult to learn and use, especially if they are not familiar with programming. Additionally, R can be slower than SAS when working with large datasets, which can be a challenge for clinical researchers who need to process and analyze large amounts of data quickly. Another example, the validation of the clinical results generated from R, validation of clinical data analysis results is critical for ensuring that healthcare providers have accurate and trustworthy information to support decision-making and improve patient outcomes.

Why we required validation/cross check on clinical results from R

- Ensuring accuracy: Validation helps to confirm the accuracy of the results produced by the analysis, which is crucial for making informed decisions about patient care.
- Maintaining credibility: By validating the results, healthcare providers can demonstrate the rigor of their methods and build trust with patients and other stakeholders.
- Preventing errors: Validation helps to identify errors or biases in the data analysis, reducing the risk of incorrect conclusions being drawn.
- Improving quality: Regular validation of results can help to improve the quality of data analysis, ensuring that it remains relevant and reliable over time.

DATA COMPARISON TOOL

Data Comparison Tool is a dashboard developed using shiny framework which allows users to compare two datasets (SDTM datasets generated from R and SAS) to identify differences and similarities between them. This tool is commonly used to verify the accuracy and consistency of data, identify errors or discrepancies, and reconcile data from different sources. This data comparison tools can help users to save time and ensure the accuracy and consistency of their data.

Data Comparison Tool has an option to upload R and SAS SDTM datasets from user location and tool will perform the data comparison of the two uploaded datasets and gives comparison summary results. Also, Dashboard has the feature to draw comparison summaries at different levels and provides graphs of the comparison results to visualize the deviations. These comparison tools are developed using R shiny and python shiny framework in order to demonstrate and test the python shiny framework. The dashboards have similar user interface and, tool input options accept only '.RData' and '.sas7bdat' files. This tools is relatively simple but shows you how to work with python shiny UI elements and render tables and charts. Mainly the dashboards have two tabs as follows,

- Comparison Summary: Provides comparison summary results of the uploaded files
- Graphs: Provides the grouped summary graphs of the uploaded files.

Once the R data comparison tool is launched, dashboard looks like:

Figure 1: R – Data Comparison Tool

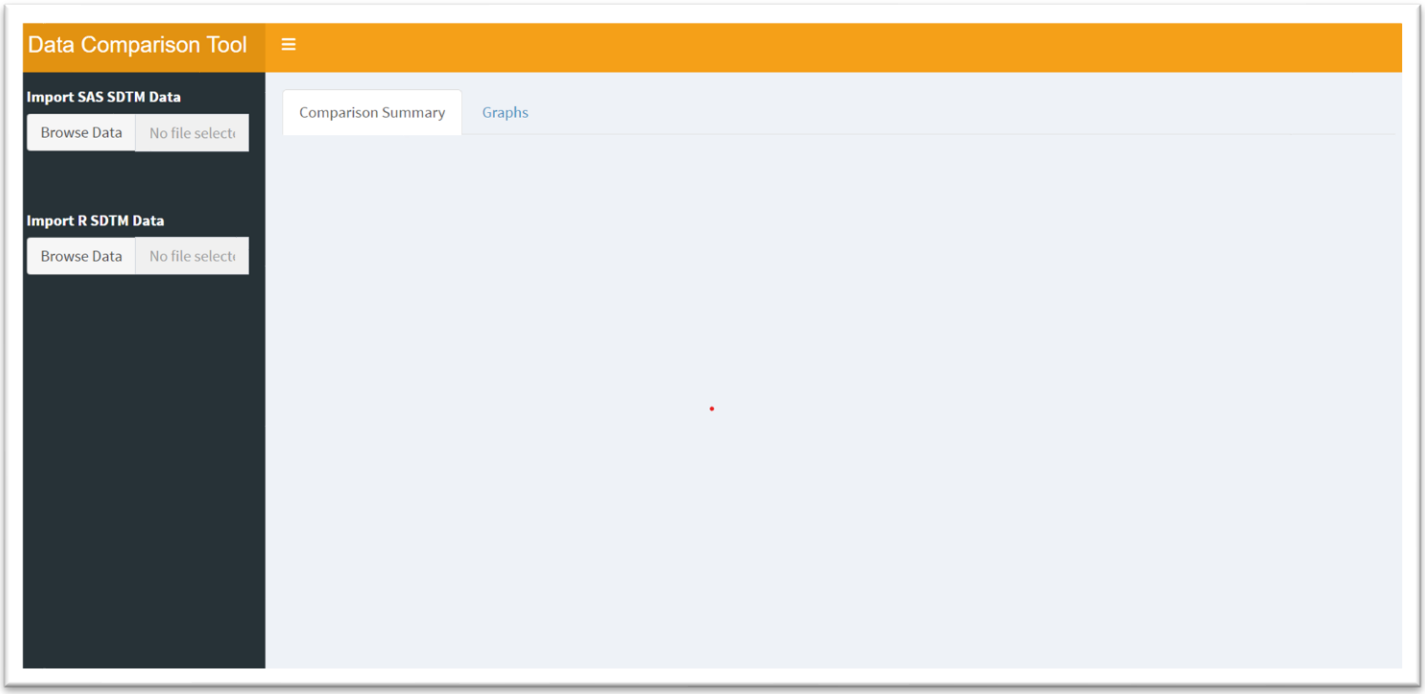


Figure 2: R – Data Comparison Tool

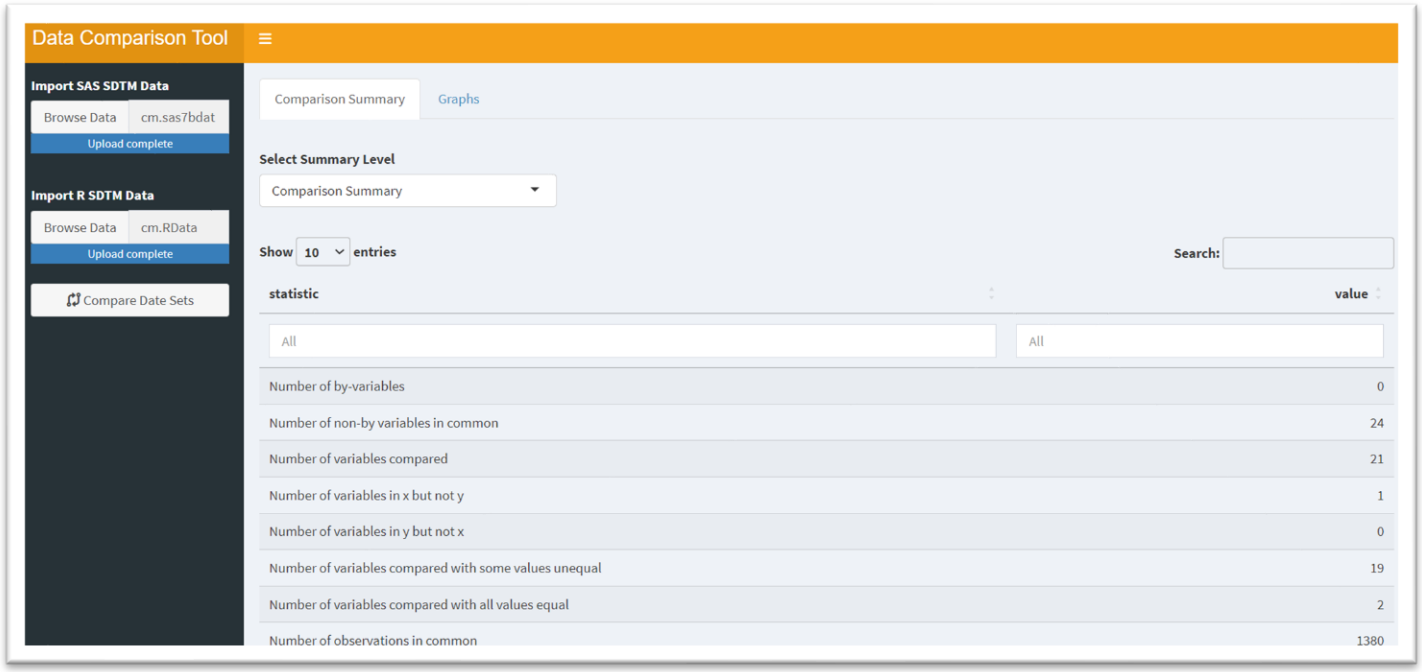
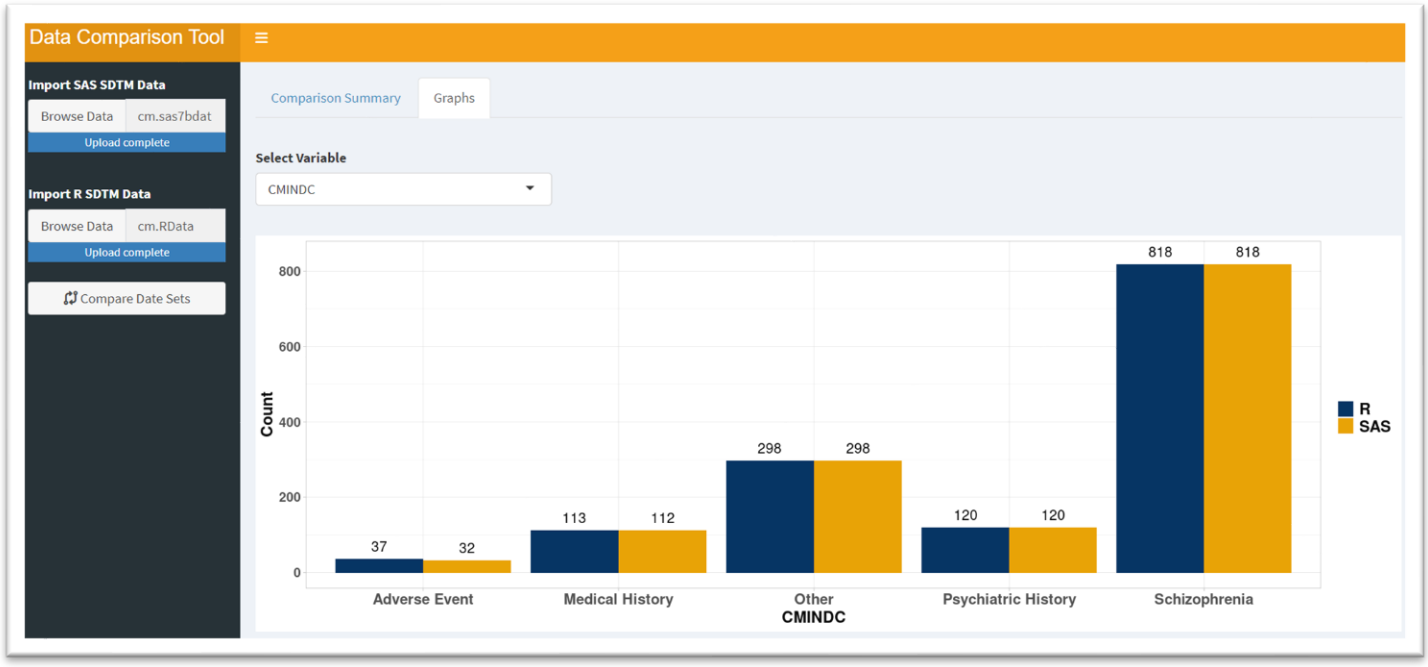


Figure 3: R – Data Comparison Tool



Once the python data comparison tool is launched, the dashboard looks like:

Figure 4: Python – Data Comparison Tool

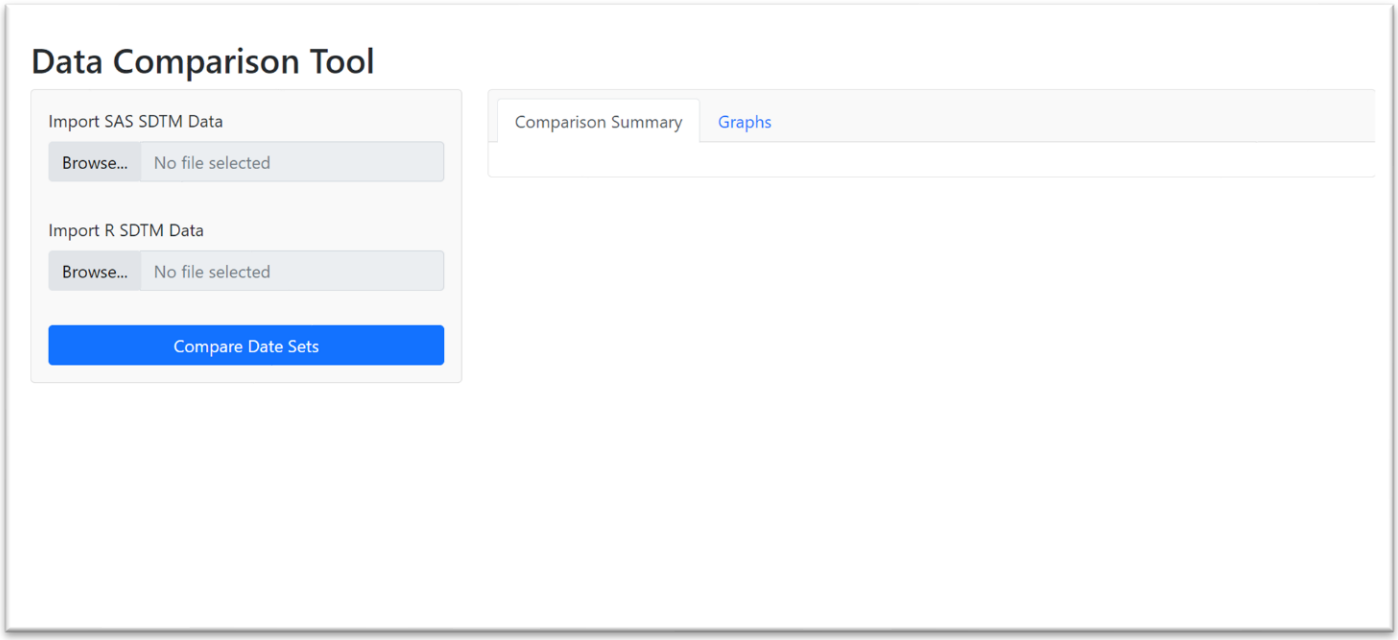
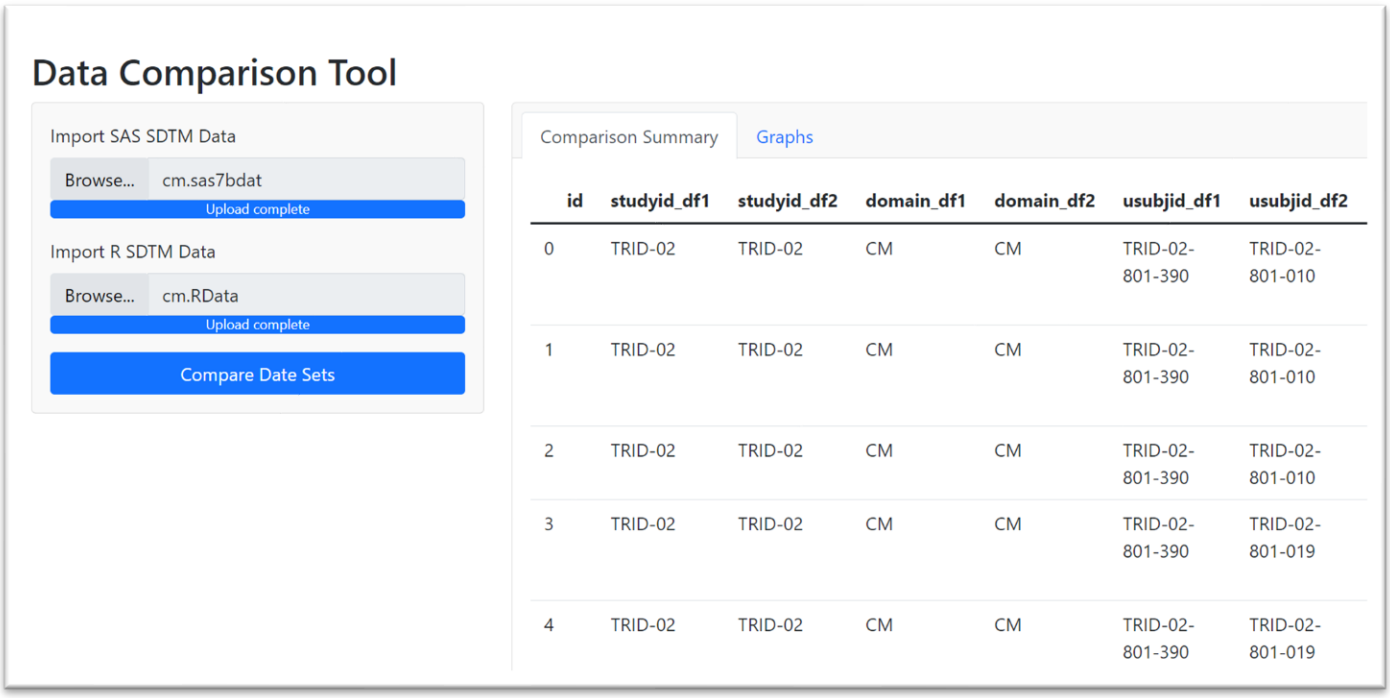
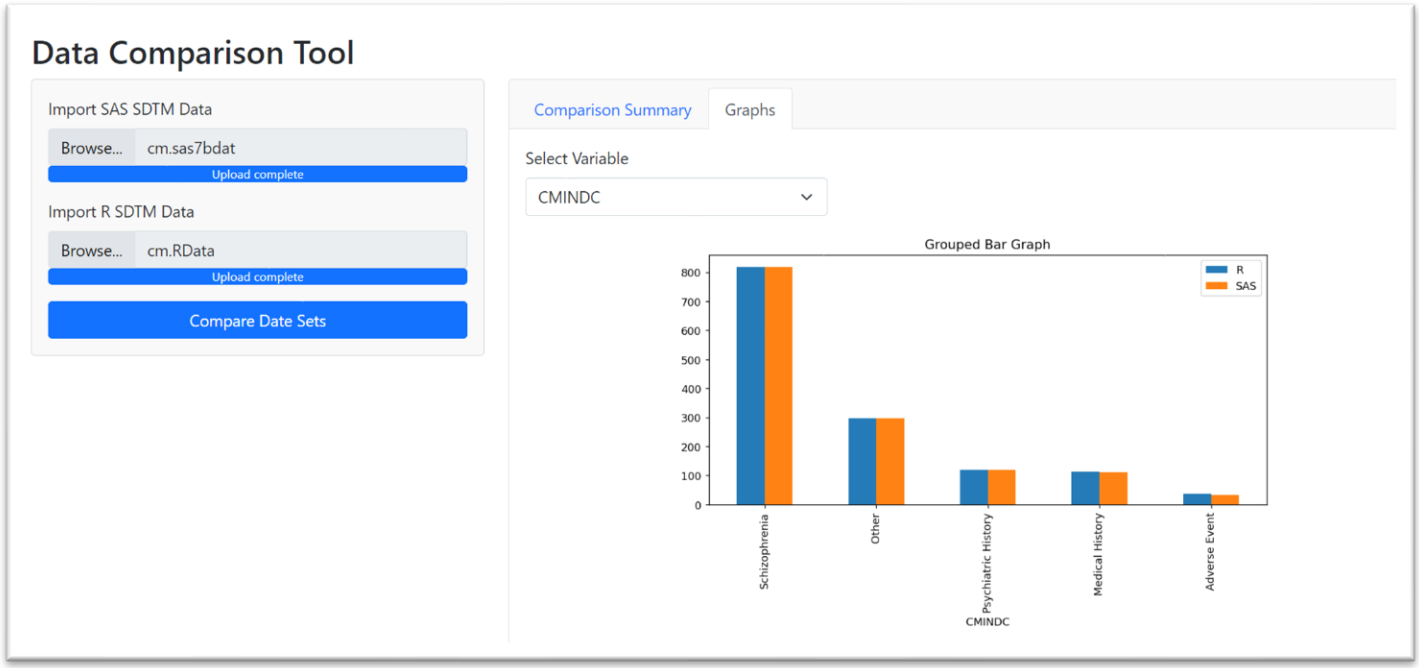


Figure 5: Python – Data Comparison Tool



id	studyid_df1	studyid_df2	domain_df1	domain_df2	usubjid_df1	usubjid_df2
0	TRID-02	TRID-02	CM	CM	TRID-02-801-390	TRID-02-801-010
1	TRID-02	TRID-02	CM	CM	TRID-02-801-390	TRID-02-801-010
2	TRID-02	TRID-02	CM	CM	TRID-02-801-390	TRID-02-801-010
3	TRID-02	TRID-02	CM	CM	TRID-02-801-390	TRID-02-801-019
4	TRID-02	TRID-02	CM	CM	TRID-02-801-390	TRID-02-801-019

Figure 6: Python – Data Comparison Tool



NOTES OF PYTHON FRAMEWORKS

As mentioned in the abstract, R Shiny was launched a decade ago in July 2012, it has a long development history that is not be incorporated in Py shiny.

- Lack of package/modules: Shiny is a great framework for developing web applications in R, because of the ecosystem of related packages: development tools, charts, tables, UI components, etc. At this stage, Shiny for Python lacks such an ecosystem. While Shiny developers have access to such great component libraries like shiny.semantic or shinyWidgets, in Python everything has to be written from scratch.
- Limited customization: While these frameworks provide a lot of built-in functionality and pre-built components, customizing them can be challenging.
- Deployment challenges: Deploying an application built with python frameworks might be challenging, particularly if we want to deploy the python dashboard in the Posit, we have use R studio to publish the application.

VISUALIZATION TECHNIQUES IN R AND PYTHON

Visualization techniques in R and Python refer to various methods used to create graphical representations of data. R and Python are popular programming languages for data analysis and visualization, and they offer several libraries and tools for creating visualizations.

Some of the popular R packages available for creating graphs are:

- ggplot2 - This is a powerful package for creating graphics and visualizations. It provides a high-level interface for creating aesthetically pleasing graphs.
- plotly - This package allows the creation of interactive graphs with a variety of options for customization.
- lattice - This package provides a system for creating trellis plots and other complex graphics.
- base - This is a basic package that is included with R and provides functions for creating basic graphs.
- ggraph - This package is used for creating network and graph visualizations.
- igraph - This package is used for creating and manipulating graphs and networks.

- visNetwork - This package is used for creating interactive network visualizations.
- networkD3 - This package is used for creating interactive network visualizations using D3.
- rCharts - This package is used for creating interactive charts and graphs.

Some of the popular python modules/packages available for creating graphs are:

- Matplotlib - It is a popular data visualization library used to create static, animated, and interactive visualizations in Python.
- Seaborn - It is a Python data visualization library based on Matplotlib. Seaborn provides beautiful default styles and color palettes to make statistical plots more attractive and informative.
- Plotly - It is an interactive, browser-based plotting library for Python. It allows users to create interactive, web-based dashboards, scientific charts, and graphs.
- NetworkX - It is a Python package for the creation, manipulation, and study of complex networks.
- Bokeh - It is a Python library for creating interactive visualizations for modern web browsers. Bokeh provides a flexible and powerful way to create interactive plots, dashboards, and data applications.
- Pygal - It is a Python library for creating scalable vector graphics (SVG) charts and graphs. It supports a wide range of chart types, including line, bar, pie, and radar charts.
- Altair - It is a declarative visualization library for Python, based on the Vega-Lite visualization grammar. Altair makes it easy to create complex visualizations with just a few lines of code.
- Plotnine - It is a Python library for creating beautiful and effective visualizations based on the ggplot2 package in R. Plotnine provides a high-level interface for creating attractive and informative visualizations with minimal coding.

CONCLUSION

The pharmaceutical industry is no exception to the increasing demand for scalable and approachable web application solutions. Shiny provides a flexible and cost-effective way to develop such applications for data analysis, modeling, and visualization purposes. Shiny can help pharmaceutical companies make data-driven decisions, improve patient outcomes, and ensure regulatory compliance. This article explores the benefits and applications of using Shiny in the pharmaceutical industry.

1. User-friendly interface: Shiny's user interface is intuitive and easy to use, making it an ideal choice for non-technical users who want to explore data, visualize trends, and perform simple analyses. This feature makes Shiny applications accessible to a wider audience, including decision-makers, researchers, and clinicians.
2. Scalability: Shiny applications are scalable, meaning they can handle large datasets and multiple users simultaneously without compromising performance. This is critical in the pharmaceutical industry, where large datasets are commonplace, and multiple users need access to data for various purposes.
3. Customizable: Shiny allows developers to customize applications to meet specific user needs. This means that stakeholders can request specific features and functionalities, and developers can easily integrate them into the application.
4. Integration: Shiny can integrate with other tools and technologies such as databases, APIs, and cloud services, making it a powerful and versatile platform for data analysis and visualization.
5. Hosting and Deployment: Shiny web applications can be easy deployed using RStudio Connect, shinyapps.io and shiny server

Python and R are valuable tools and widely used in pharmaceutical trials for data analysis and statistical modeling. Python is particularly useful for data manipulation, machine learning, and visualization, while R is particularly useful for statistical analysis and graphics. R also becomes one of the most effective tools for interactive data visualization and storytelling when used in combination with Shiny. Many clinical trial teams use both languages together to maximize their analytical capabilities.

Dashboards are important in the pharmaceutical area because they provide a clear and concise overview of the critical information that is required to make informed decisions. Also, dashboard helps to improve decision-making, streamline processes, and increase efficiency. By providing real-time access to critical information, pharmaceutical companies can make informed decisions that ultimately improve patient outcomes. Shiny is a powerful and flexible web application framework that can be used in various applications in the pharmaceutical industry. Integrating Shiny to Python is undoubtedly a positive development for the web framework.

REFERENCES

https://shinylive.io/
https://shiny.rstudio.com/py/api/reference/shiny.ui.output_table.html
https://shiny.rstudio.com/deploy/

ACKNOWLEDGMENTS

We would like to express our gratitude to Genpro Research Team for their invaluable support and encouraging in the development of the research work. Our heartfelt thanks Limna Salim for providing us with guidance and feedback throughout the research work. Thanks to Senju Mani for supporting the development side of python dashboard. And finally, Our families and friends for their unwavering support and encouragement throughout the research work.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name : Vikash Jain

Company : Genpro Research Inc

Work Phone : 703-577-0496

Email : vikash.jain@genproresearch.com

Author Name : Denny Sebastian

Company : Genpro Research Inc

Work Phone : +91.9539693707

Email : denny.sebastian@genproresearch.com