



PharmaSUG 2026 - Paper ET-283

An Innovative solution for Interactive Dashboards Using Python Flask Framework for Clinical Data Analytics

Manish Dineshbhai Bhagchandani, Epicacy Lifescience Analytics Pvt Ltd, Bengaluru, India

ABSTRACT

Traditional clinical reporting tools and built-in dashboard solutions often have fixed structures and limited flexibility, which can restrict deeper exploration of clinical trial data. This paper demonstrates how the Python Flask framework can be used to build interactive and customizable dashboards using SDTM and ADaM datasets. By moving away from rigid, pre-packaged tools to a lightweight web-based framework, Flask enables greater control over layout, logic, and user interaction while integrating smoothly with the Pharmaverse ecosystem. An end-to-end workflow is presented, including data preparation using Pandas, interactive visualizations created with Plotly, and enhanced user interactions using JavaScript. These dashboards support data validation, study review, safety monitoring, and exploratory data analysis by statistical programmers, clinical scientists, and study teams. They promote effective collaboration by providing timely and easy-to-interpret views of clinical data. Implemented within a secure environment, this framework offers a scalable and practical solution for enhancing clinical data review while aligning with regulatory expectations.

INTRODUCTION

In the modern clinical trial environment, the demand for real-time data exploration has outgrown the capabilities of traditional static tables, listings, and figures (TLFs). Clinical stakeholders often experience a latency gap between requesting specific data views and receiving the programmed outputs. This paper introduces an open-source, web-based framework utilizing Python Flask, *PyReadStat*, and *Plotly* to bridge this gap.

The developed application provides a self-service environment that securely reads raw [.sas7bdat](#) files directly from a directory and translates them into interactive, highly customizable dashboards. Users can seamlessly toggle SDTM standards for raw data exploration and ADaM datasets for derived statistical summaries.

SYSTEM ARCHITECTURE AND METHODOLOGY

The dashboard utilizes a decoupled architecture to ensure data integrity, high performance, and rapid deployment. The architecture comprises three primary components:

- **Data Ingestion Layer:** *PyReadStat* library is utilized to parse raw [.sas7bdat](#) files, maintaining variable attributes and formats without requiring a native SAS installation.
- **Backend Analytical Engine:** A Python Flask server routes asynchronous requests. It leverages the *pandas* library to perform real-time clinical aggregations, dynamically identifying treatment variables (e.g., ARM for SDTM, TRTP for ADaM) to generate clinical summaries.
- **Interactive Frontend:** An HTML/JavaScript interface captures user inputs, dynamically adjusts control visibility based on the selected plot type, and renders interactive graphs using the *Plotly JavaScript* library.

To handle SAS-specific date formats that standard JSON encoders cannot process, a custom JSON provider is embedded within the Flask application. Program 1 demonstrates how the user manages data ingestion and merges demographic details to assign treatment arms to findings domains.

```
dm_path = os.path.join(DATA_PATH, "dm.sas7bdat")
if file_name.upper() not in ["DM", "ADSL", "ADAE"] and
os.path.exists(dm_path):
    dm_df, _ = pyreadstat.read_sas7bdat(dm_path)
    dm_df["USUBJID"] = dm_df["USUBJID"].astype(str).str.strip()
    df = pd.merge(df, dm_df[["USUBJID", "ARM", "SEX"]], on="USUBJID",
how="left")
    return df
except: return None
```

Program 1. Caption for Sample Program

DYNAMIC CLINICAL DATA EXPLORATION

The framework is designed to provide specific logic paths depending on the clinical data standard and the statistical requirements of the selected domain.

DUAL-STANDARD COMPATIBILITY

The user interface features a "Data Standard" selector that dynamically updates the available analytical modules. When "SDTM" is selected, the tool provisions access to Demography (DM), Adverse Events (AE), Disposition (DS), Laboratory Measurements (LB), and Vital Signs (VS). Switching to "ADaM" populates the navigator with Subject Level Analysis (ADSL), Adverse Event Analysis (ADAE) etc.

INTELLIGENT USER INTERFACE CONTROLS

To minimize user error and maintain clinical relevance, the front-end JavaScript dynamically governs the available axes and grouping inputs. The interface adapts based on the selected analysis type:

- **Pie Chart:** The numeric Y-axis and grouping variables are programmatically hidden, enforcing a strict focus on proportional visualization of a single categorical variable (e.g., SEX or ACTARM).
- **Histogram / Bar Chart:** The numeric Y-axis is hidden as the system automatically computes categorical frequency counts, such as the incidence of specific Adverse Events (AEDECOD) or disposition reasons (DSDECOD).
- **Line Chart & Box Plot:** The system mandates the selection of a continuous numeric variable for the Y-axis (e.g., VSSTRESN). It populates the X-axis primarily with temporal variables like VISIT or ADY for longitudinal trends.

CLINICAL TLF GENERATION

A core innovation of this tool is the real-time translation of grouped data into industry-standard **Table, Listing, and Figure (TLF)** structures. Instead of relying on generic descriptive statistics, the framework uses a specialized aggregation engine to generate summary tables. When a user interacts with a chart, the system instantly calculates and displays:

- **N:** Subject or observation count.
- **Mean (SD):** Arithmetic mean and standard deviation.
- **Median:** The middle value for skewed distributions.
- **Range (Min - Max):** The full spread of the clinical data.

This summary is hidden by default to maintain a clean visual workspace but can be toggled by the stakeholder for deep-dive statistical review.



Figure 1. Dynamic Standard Selector and Module Navigation featuring Proportional Logic

ADAM ANALYSIS WORKFLOWS: PIE AND HISTOGRAM EXAMPLES

The framework natively identifies ADaM treatment variables, such as TRTP (Planned Treatment) or TRTA (Actual Treatment), to adapt summary algorithms for both subject-level and event-level analysis.

In the **ADSL module**, categorical variables like ACTARM allow stakeholders to visually confirm population balance through interactive **Pie Charts**. For example, by selecting ACTARM as the primary variable, a user can instantly visualize the proportional distribution of subjects across actual treatment groups. This facilitates rapid insights for reviewers, such as Data Safety Monitoring Boards (DSMBs), to ensure study randomization integrity.

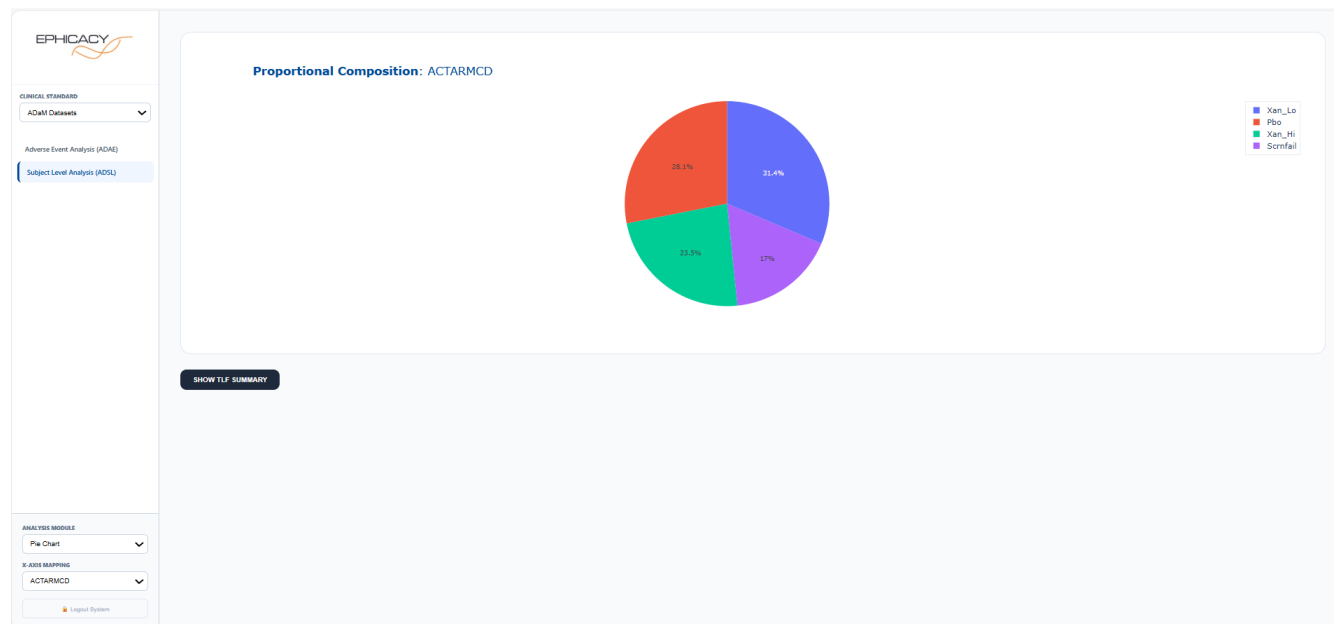


Figure 2. ADSL Pie Chart Proportion Output

FACETED LONGITUDINAL ANALYSIS: LINE CHART EXAMPLE

For findings domains such as **Laboratory Measurements (LB)** and **Vital Signs (VS)**, the tool utilizes **Line Charts** to visualize mean changes over time. By selecting VISIT as the X-axis and a numeric result like LBSTRESN as the Y-axis, the system generates longitudinal trend lines.

To handle the complexity of clinical data, the tool automatically facets these charts by categorical parameters like LBTESTCD or VSTESTCD. For instance, a stakeholder can view a grid of line charts showing the mean change in Blood Pressure, Heart Rate, and Respiratory Rate over multiple clinical visits, grouped by **ARM** or **SEX**. This allows monitors to identify treatment-emergent trends or systemic outliers across the entire study population in a single view.



Figure 3. Faceted Line Chart showing Trends in Vital Signs (VS) by Visit and Treatment Arm

STAKEHOLDER UTILIZATION AND COLLABORATION

This interactive solution extends utility beyond the internal study team, serving as a centralized portal for various oversight bodies:

- **Core Study Teams (Clinicians, Data Managers, and Statisticians):** Empowers the internal team to proactively monitor patient-level safety profiles, visualize data outliers for immediate cleaning, and perform "what-if" exploratory subgrouping before finalizing formal ADaM derivations.
- **Regulatory Bodies (FDA/EMA):** Provides an auditable and transparent environment where reviewers can verify study findings through direct interaction with submitted datasets, reducing the need for repeated information requests.
- **Data Safety Monitoring Boards (DSMB) & IRBs:** Enables rapid decision-making regarding patient safety through integrated adverse event (AE) incidence tracking and trend analysis during scheduled reviews.
- **Adjudication Committees:** Supports the review of clinical endpoints by providing consolidated, interactive views of subject history, laboratory results, and event severity in a single display.

SECURITY, DEPLOYMENT, AND SCALABILITY

To meet stringent clinical industry requirements, the framework is designed for execution within Secure GxP Environments. It can be deployed within an organization's virtual private cloud (VPC) or behind a security system using production-grade WSGI servers like Gunicorn or Nginx. Integration with LDAP or OAuth2 ensures that only authorized personnel have access to specific study data.

The solution is inherently Scalable. Since the backend logic is standard-driven rather than dataset-specific, adding new domains or studies simply requires placing the corresponding `.sas7bdat` files in the designated data directory. The Python-based backend can be containerized using Docker, allowing for rapid horizontal scaling across multiple studies or massive datasets without performance degradation.

RESULTS AND DISCUSSION

Deploying this Python Flask architecture eliminates the friction of traditional data review tasks.

The immediate rendering of dynamic *Plotly* charts allows data managers and clinical scientists to actively filter parameters, isolate outlier patients by hovering over data points, and independently scale axes across heavily faceted data (such as separating distinct laboratory tests). Furthermore, the "Group By" control allows users to instantly pivot their data viewing angle, breaking down trends by variables such as SEX, RACE, or AESEV without requiring a programmer to generate a new output.

CONCLUSION

This framework represents a significant paradigm shift in clinical trial reporting, successfully demonstrating that modern micro-frameworks like **Python Flask** provide a robust, highly scalable engine for the next generation of clinical analytics. By automating the ingestion of CDISC-compliant **SDTM and ADaM** `.sas7bdat` files and mapping them directly to faceted visualizations and on-demand TLF summaries, this solution bridges the gap between raw data and actionable intelligence.

The resulting self-service environment empowers cross-functional teams to move beyond the limitations of static PDF reports, facilitating agile study reviews and the rapid identification of critical safety signals. Ultimately, this approach not only improves operational efficiency but aligns with the broader industry evolution toward an **interactive and future-ready** clinical trial ecosystem, where transparency and real-time data exploration are the standard for regulatory success.

REFERENCES

- CDISC**. 2021. *SDTM Implementation Guide (v3.3)*. Standards for Human Clinical Trial Data. [Online]. Available: <https://www.cdisc.org/standards/foundational/sdtmig>
- CDISC**. 2021. *ADaM Implementation Guide (v1.3)*. Analysis Data Model Standards. [Online]. Available: <https://www.cdisc.org/standards/foundational/adam>
- Pharmaverse Community**. 2026. *Admiral: ADaM In R Asset Library (and Python equivalents)*. Open-Source Clinical Reporting Framework. [Online]. Available: <https://pharmaverse.org/>
- The Pandas Development Team**. 2026. *Pandas: Powerful Data Analysis and Manipulation Library for Python*. [Online]. Available: <https://pandas.pydata.org/>
- Pallets Projects**. 2026. *Flask Documentation (v3.0.x)*. Modern Web Development with Python. [Online]. Available: <https://flask.palletsprojects.com/>
- Plotly Technologies Inc**. 2026. *Plotly Python Graphing Library: Interactive Visualizations for Data Science*. [Online]. Available: <https://plotly.com/python/>
- Roche, A**. 2026. *PyReadStat Documentation*. Multi-platform library to read SAS (`sas7bdat`) and SPSS files into Python. [Online]. Available: <https://github.com/Roche/pyreadstat>
- Gunicorn Project**. 2026. *Gunicorn: Python WSGI HTTP Server for UNIX*. [Online]. Available: <https://gunicorn.org/>
- PythonAnywhere (Anaconda Inc.)**. 2026. *Deployment of Flask Applications in Cloud Environments*. [Online]. Available: <https://help.pythonanywhere.com/>

ACKNOWLEDGMENTS

The author thanks the leadership and colleagues at Efficacy Lifesciences Analytics for championing this exploration of open-source clinical technologies. Special appreciation goes to Prashant Kumar Srivastava, Mrityunjay Kumar, Akshata J Salián, the internal review committee, and the Conference Guides team for their invaluable technical insights and dedicated support. Finally, immense gratitude is extended to the PharmaSUG 2026 organizers and the global open-source community for continuously driving innovation in our industry.

CONTACT INFORMATION

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

Manish Dineshbhai Bhagchandani
Technical Lead – Statistical Programmer
Efficacy Lifescience Analytics Pvt Ltd, Bengaluru, India
Email: manish.bhagchandani@efficacy.com
LinkedIn: <https://www.linkedin.com/in/manish-bhagchandani-71088439a>

APPENDIX

Source Code Repository The complete Python Flask application, including the front-end HTML/JavaScript templates and back-end data processing logic, is open-source and available for review, replication, and collaborative development.

- **GitHub Repository:** https://github.com/bmanishd54/Python_Flask_Clinical_Dashboard

Live Application Demo To provide reviewers and conference attendees with hands-on experience of the interactive framework discussed in this manuscript, a live demonstration of the clinical dashboard has been temporarily deployed.

- **Application URL:** <https://dmanishd54.pythonanywhere.com>
- **Username:** Python_Efficacy
- **Password:** admin

Note: For security and hosting resource management, this live demonstration link will remain active for 30 days following the PharmaSUG 2026 conference. If you are reviewing this paper after the 30-days of expiration period and wish to access the tool or request a local deployment guide, please reach out to the corresponding author.