

PP20 A Framework for Interactive Ad-hoc Request Handling: Empowering Clinical Insights through Interactive Plots

Changhong Shi , Merck & Co., Inc., Rahway, NJ, USA (changhong_shi@merck.com)

Jaime Yan, Merck & Co., Inc., Rahway, NJ, USA (mingyu.yan1@merck.com) Chao Su, Merck & Co., Inc., Rahway, NJ, USA (chao.su@merck.com)

A framework for effectively handling ad-hoc requests in the clinical domain by leveraging interactive plots.

Background

Ad-hoc requests often require interactive and easily transferable outputs in a formal format. The primary purpose of such requests is to explore the data to support decision-making.

Target

Provide interactive plots with:

- Flexibility
- Low cost(learning and resource)
- Easy to handle and reproducible
- Quick with high quality

A Framework to standardize the process:

- Satisfy the most frequent ad-hoc requests
- Easy to maintain(modularize)
- User-friendly with diminishing-returns learning curve
- From start to end

Challenge

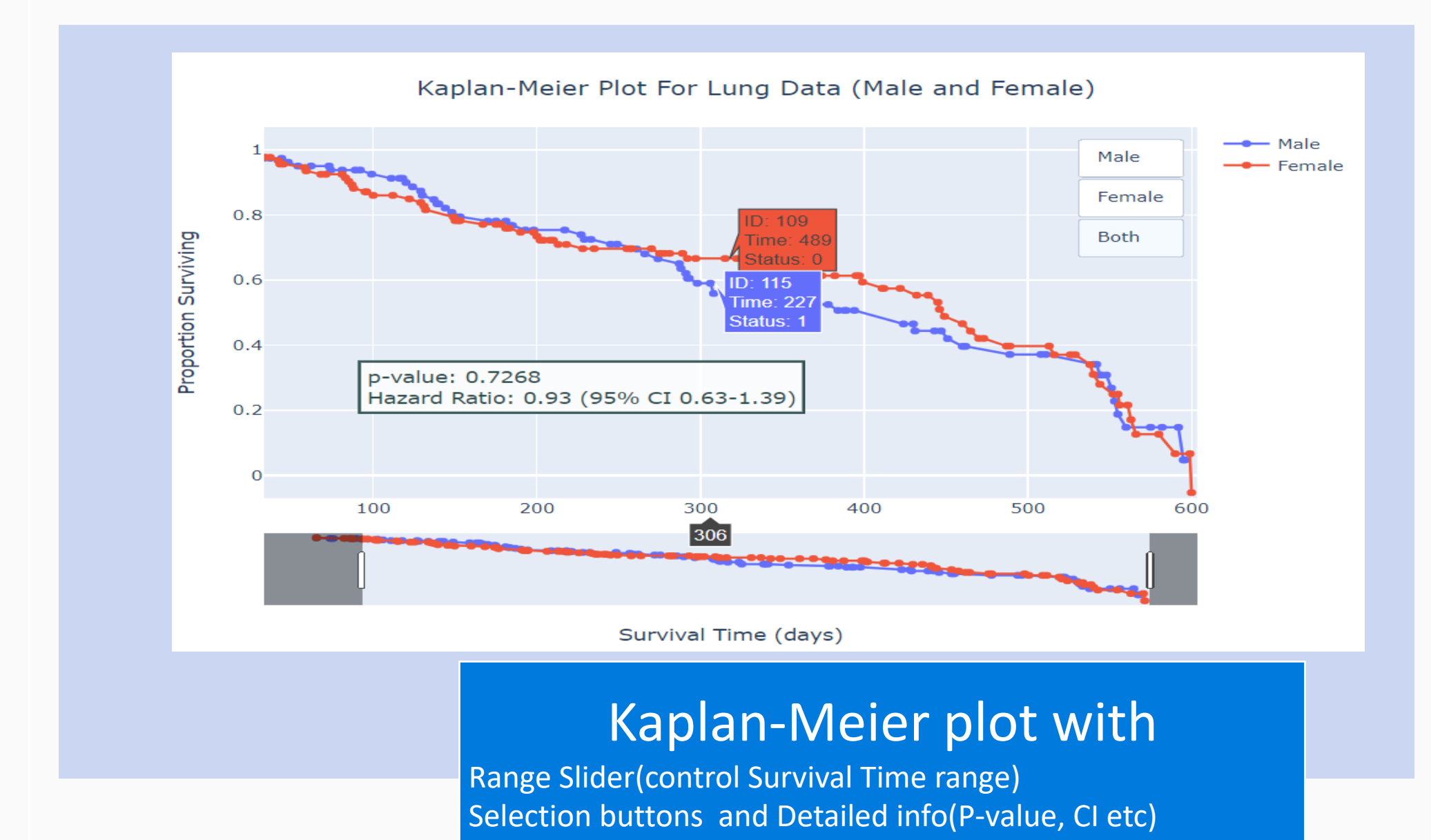
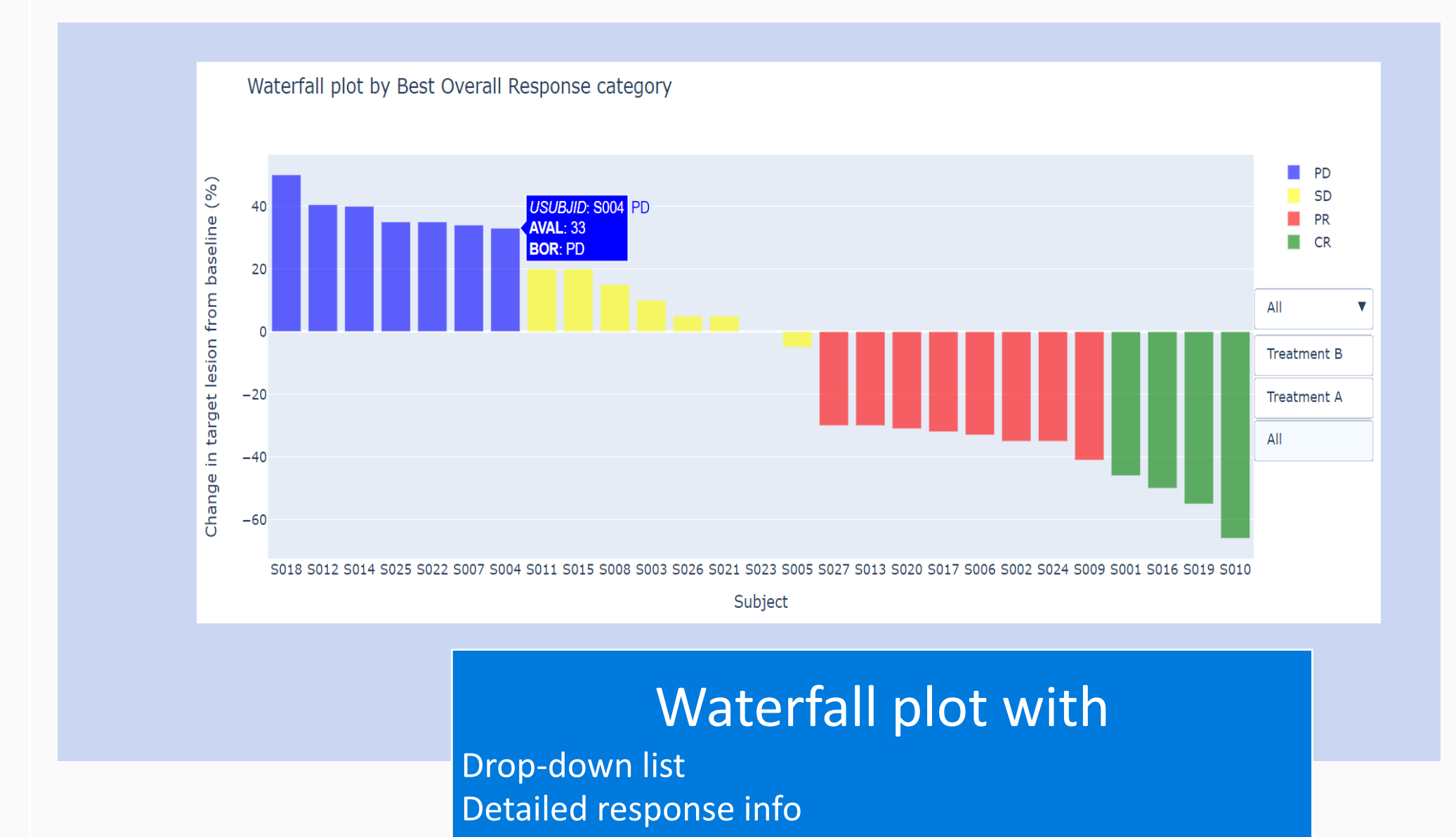
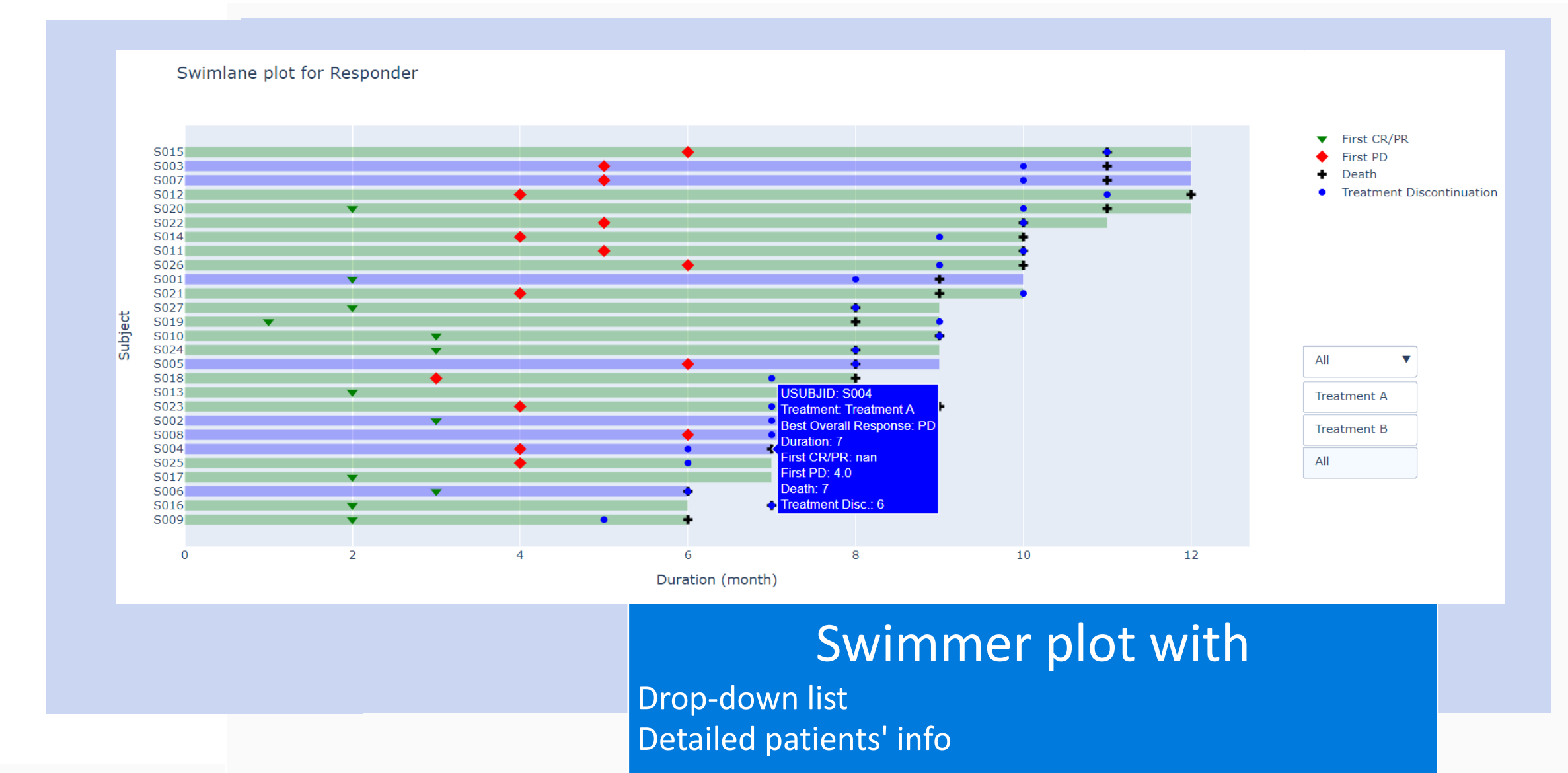
- Need open-source technology for interactive plots
- Decrease the training cost for SAS users to R/Python
- Set up the programming environment
- Select alternative packages for the same purpose
- Version control for the package
- Standardize the interactive plot-producing process
- Validate the R/Python result
- Make interactive results portable on any machine

Design/Structure

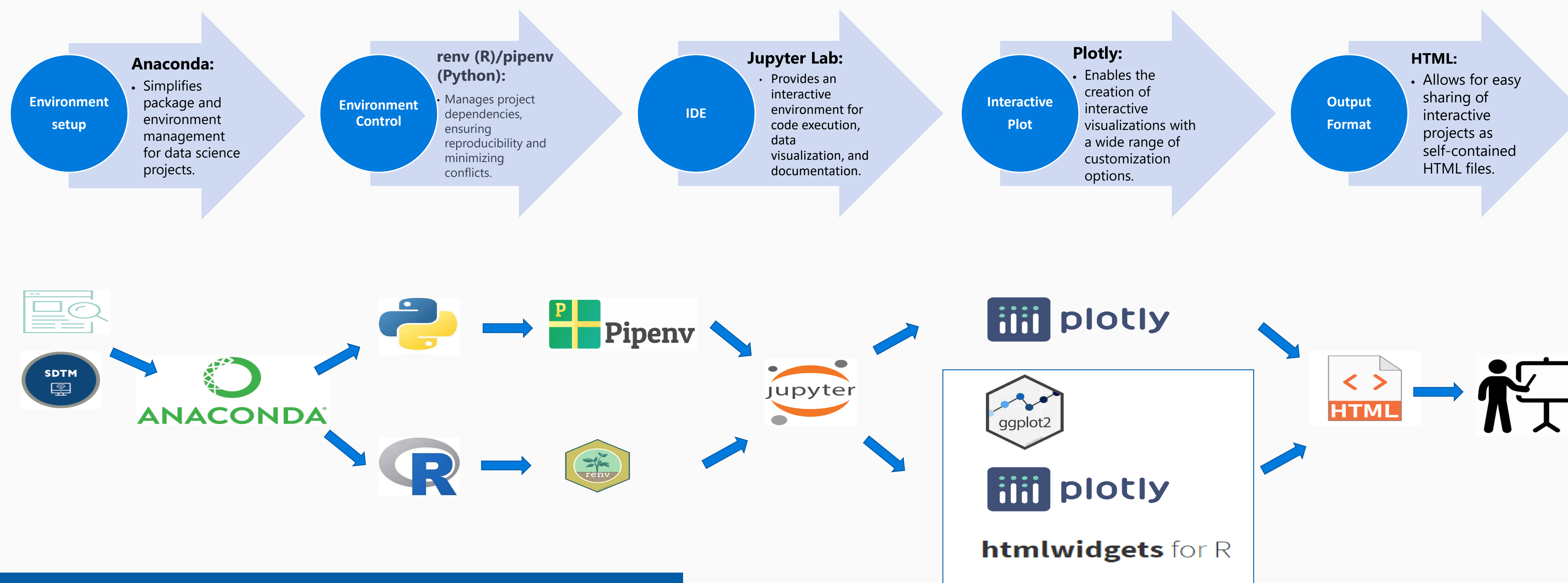
- Source Data: SDTM and ADaM
- Environment setting: ANACONDA
- Package version control: Pipenv(Python)/Renv(R) with lock file
- IDE: Jupyter Lab
- Interactive plot:
 - Plotly(Python)
 - Plotly+ggplot2+htmlwidgets(R)
- Output: HTML format
- Validation: Crosschecking between SAS, R, and Python

Solution

- Anaconda (distribution of the Python and R programming languages for scientific computing) for managing dependencies and environments
- Use renv in R and pipenv in Python to control the package version
- Use the same package solution for the entire project.
- Plotly for Python and ggplot2 plus Plotly for R
- Crosschecking result between SAS and R/Python
- Save in HTML format



The Full Workflow Process



Conclusion:

The presented poster introduces an interactive framework tailored to handle ad-hoc requests in the clinical domain. By leveraging environment configuration, version control, and interactive HTML plots, the framework enables clinical teams to efficiently analyze potential requirements, collaborate with experts, and deliver customized visualizations. This systematic approach improves productivity, facilitates informed decision-making, and enhances responsiveness in addressing ad-hoc requests. Implementing this framework has the potential to transform ad-hoc request handling, leading to improved efficiency and better insights into clinical research and practice.