Paper AD16

Rise and Shine: Using R Shiny to explore insomnia clinical trial data

Matthew Brierley, Idorsia, Allschwil, Switzerland

ABSTRACT

In recent years many companies have embarked on projects seeking to introduce or extend R-based tools into the workflow around analysing clinical trial data. Particularly, R Shiny has been explored due to its ability to supplement static analysis with dynamic displays to speed-up data review and exploration. This year, in preparation for the completion of the first Phase III trials at Idorsia, the project team piloted the creation of Shiny applications to help the team understand and explore study results more quickly once the unblinded ADaM datasets became available. This paper summarizes the project detailing the background, content, and usage of the R Shiny apps. For groups considering or recently beginning upon similar projects the shared experience may give some reflection on the path ahead and for those with more mature projects a comparison of usage and workflows may be interesting.

INTRODUCTION

This year saw the completion of the first phase III clinical trials at Idorsia for a compound targeting insomnia. The phase III program comprised two pivotal trials with identical designs and differing only in the dose levels of the active treatment arms, plus an optional extension study which patients from both trials could elect to continue into.

CSR programming for all studies was performed by an external CRO partner. To help faster explore the unblinded data the Biometry team piloted using R Shiny applications developed in-house for each study. It was hoped that the apps would permit the group to accelerate exploration and interrogation of the data to generate faster insights and enable swifter progress with exploratory analyses within the team.

This was the first use of R Shiny within the Biometry group, and development of the app began approximately 3 months before the database lock of the first study. The intent was to make the apps available to the Biometry team shortly after the receipt of validated ADaM datasets. The target users were the project and trial statisticians, but the intent was to be able to share results of the app with cross-functional project team members during meetings.

DESIGNING CONTENT FOR MAXIMUM IMPACT

Separate apps were created for each of the pivotal trials, the extension study, and the Integrated Summary of Effectiveness (ISE). The apps for the pivotal trials were identical except for an additional tab included for the second study containing combined summaries of key results across both trials.

When discussing and prioritising content to include in the app we were always trying to anticipate what the most likely needs of the team would be following the release of the CSR results. We reviewed the Phase II results and the exploratory analyses that were performed following those trials. Since the safety profile of the compound was promising we focused more attention on developing content aimed at exploring the efficacy data.

We felt that simply trying to reproduce as many CSR outputs as possible within a shiny app was unlikely to result in the most effective overall result for exploratory purposes. Instead, we focused on analyses and particularly plots that sought to add context or alternative visual representations of central or complex reports, which could potentially be much more useful to the team.

EFFICACY DATA

Overall, numerous efficacy parameters were collected for each patient over the duration of the studies. There were 3 primary sources of efficacy data:

i. **Polysomnography (sleep lab) data**: During certain trial visits patients were required to spend nights at a sleep lab. This provided objective measurements of many sleep parameters including information on sleep duration and interruptions but also on the sleep architecture and time spent in different sleep stages.

ii. **Subjective sleep diary data:** Whilst on the trials, patients were required to complete a daily sleep diary. The diary captured subjective assessments of key sleep parameters such as time until sleep onset, and sleep duration and interruptions to sleep.

iii. **Questionnaire data:** Additionally, over the duration of study patients completed multiple questionnaires to assess daytime functional ability.

The combination of having many parameters to analyse, coupled with a selection of both protocol-defined and team-identified exploratory subgroups is often a great use-case for a Shiny app. The vast number of potential analysis combinations can never be covered within a CSR, and often the results across primary or key secondary endpoints may necessitate exploring particular lower level parameters in more detail. For example, a CSR may present multiple secondary analysis endpoints each comprised of composite individual parameters which are not summarised individually. Having a prepared Shiny app where any single collected parameter may be summarised dynamically can be extremely useful.

Every analysis report contained within the app could be broken down by subgroups. This was controlled by creating an additional ADSL-style data frame supplied to the app containing patient ID's along with protocol specified subgroups and further exploratory subgroups. This subgroup data frame was augmented further following the review of initial CSR results and based upon questions and requests from the team.

FURTHER CONTENT

Although efficacy analyses were viewed as a priority, we also included the following tabs containing other functionalities to balance out the app and cover the perceived most important secondary needs:

i. **Data review/filtering:** Occasionally, when reviewing certain reports within the app, it was helpful to be able to refer back to the source ADaM data and to be able to dynamically filter data and review information on patients that met certain criteria or to be able to quickly capture these patient IDs for further data review.

ii. **Demographics:** Customizable demographics table that could be split by subgroups to make quick checks of balance across subgroups.

iii. **Adverse Events:** Short cuts to select and review the key AE summaries presented in the CSR, and to be able to customise the reports either by splitting by subgroups, or filtering by selected terms or categories.

iv. **Custom subgroup generation:** In addition to including all the protocol-defined and team identified subgroups, we had some functionality to generate additional custom subgroups dynamically within the app. For example, it was possible to set cut-points for numeric baseline variables to determine a new subgroup which was then available for each report within the app.

DEVELOPMENT AND DEPLOYMENT

When the project was initiated, Idorsia did not have a platform for deploying Shiny applications or any existing R codebase for report generation to draw upon.

APP DEPLOYMENT

As R has been increasingly incorporated into clinical data workflows, most larger programming and biostats groups have already established processes for deploying Shiny applications. For other teams late to the party and especially for teams running similar pilots and looking to get something up-and-running quickly to be able to evaluate Shiny without first embarking on any complex and technical I.T. solutions we would highly recommend considering RStudio Connect.

RStudio Connect is an enterprise solution offered by RStudio that enables fast installation of an out-of-the-box hosting platform for Shiny and other R and python based products. For initial piloting, and in terms of being able to

move forward quickly and focus time on app development rather than I.T. discussions, our experience with RStudio Connect was very positive.

Of course, other hosting solutions are available, and depending on the user base, knowledge of R amongst I.T. partners and ambition of the project, may offer further benefits. Although RStudio Connect does go a long way in making things easy, it is likely nonetheless that some work will remain in terms of controlling access to clinical data repositories used within the apps.

DEVELOPMENT WORKFLOW AND TESTING

The apps were developed by a single developer completely dedicated to the task. Weekly review meetings were held with the project statistician to update on progress, review functionality, and discuss the priorities for the following week. After initially reproducing several of the critical outputs featuring the primary and key secondary endpoints, plans for subsequent material to include were based on trying to anticipate what questions may arise as the CSR results became available.

To ensure that we could have confidence in the results displayed within the app once the unblinded data was received, all initial development was performed using the ADaM datasets associated with the final dry-run outputs. Where possible we cross-checked results from the app directly against the dry-run data, and captured a few of these results as unit tests to make sure that subsequent updates didn't inadvertently introduce errors. Given the short timelines and rapid development we were unable to implement as robust testing process as we would have liked, and for future projects we are establishing some CI/CD processes around the app deployments to enable automated testing of certain results before deploying updated versions of applications.
We also worked with the programming group to follow relevant updates to ADaM datasets requested following the final dry-run and shortly before the database lock we switched to the latest blinded datasets to make sure that these updates to the ADaM datasets would not affect the app.

APP USAGE

As soon as the data transfer of the validated unblinded ADaM data was received, we were able to load the data into the apps and have immediate access to (unvalidated) key trial results and far beyond. As we began to receive initial outputs from the study lead tables, we were able to verify results in the app, and the project statistician was already able to begin preparing for the initial communication of these results and delving more deeply into the data. At this initial stage we did not share the apps when disclosing and discussing the first results, both because they were unvalidated and because we wanted the initial discussions to be completely focused on the contents of the lead tables.

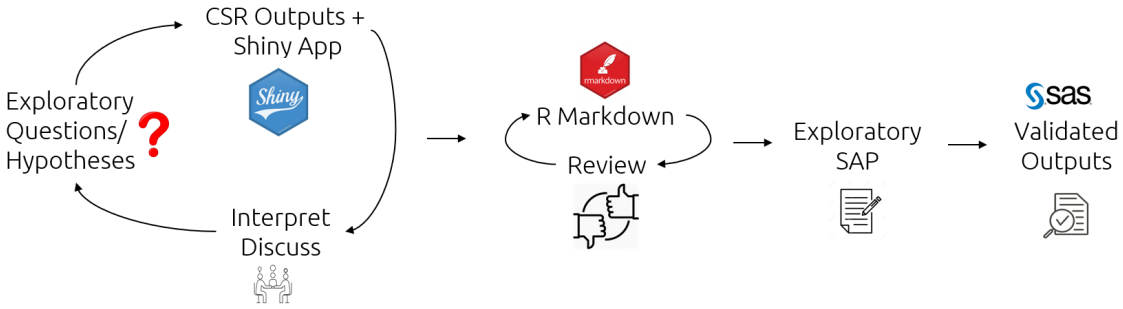
A notable success was the way that the apps were integrated into full-day cross-functional CSR results review meetings for each of the pivotal studies. The apps were used throughout the meeting as the team were able to break away from prepared slides and explore some of the questions raised immediately. The contrast between the app usage between the two meetings was striking: for the initial study, the project statistician responded to certain questions by suggesting that we could explore answers to particular questions within the app, whereas for the second study participants were more comfortable asking directly if we could review results in the app, and the app was featured much more prominently. This was a clear indication of both the acceptance of the apps into the meeting and a recognition of the value that they generated. By being able to answer questions that were not covered within the CSR results and that we had already anticipated, such as dynamically defining different response rates for certain endpoints, or plotting scatter plots and assessing relationships between certain parameters the overall value of the meeting was increased.

A further success was achieved by using the app heavily during a meeting with external insomnia experts to share selected results and help inform planning of external publications. Central to this meeting was a discussion around some of the questionnaire data that we had collected. For the CSR this was summarised at the domain level, but during the meeting the experts wanted to look at a more granular level. For example, to explore if certain questions were driving observed differences in domain scores, or the correlation between different questions within a single domain. These plots where not available in the CSR and may otherwise have taking significant programming time to produce in preparation for such a meeting. Given the burden of the programming team during that period, it is doubtful that without the existence of the Shiny app such a meeting could have been held at the same timepoint.

More generally, the apps were used frequently in smaller ad-hoc cross-functional meetings. As the team further digested the CSR results and more exploratory questions were raised, small targeted meetings were arranged to review certain reports within the app and solidify the requirements for report generation.

EXPLORATORY ANALYSIS WORKFLOW

Well-designed Shiny apps targeted at anticipating and supporting the exploratory analyses following the release of initial CSR results can open the door to efficiency gains and an overhaul of traditional processes and workflows for defining and executing exploratory analyses. At this critical time teams are seeking to understand the data and formulate and respond to questions as rapidly as possible. Traditionally, there have always been tensions as clinical teams draw up a large list of questions and requests in initial meetings which are then reviewed and vetted in subsequent meetings as negotiations follow about which questions need prioritising and translating into official requirements for programming groups. Programmers already occupied with completing remaining CSR outputs and further submission preparation are limited in terms of supportive exploratory outputs that can be produced, and do not want to spend time validating outputs that turn out not to be helpful to the team.



Using Shiny apps to directly address and explore many of these initial questions as they arise in meetings allows for more efficient meetings and faster understanding of the data. Study team members that do not have direct access to data are able to view more results whilst programmers are shielded from some of the initial requests. The diagram above shows the process that we followed for exploratory analyses.

Since access to the app was restricted to statisticians and programmers, there was also a need to be able to capture and distribute interesting results that were discussed and explored within initial meetings. For this purpose, we used RMarkdown to provide an intermediary review stage for documenting and furthering results explored into the app and discussing further before finally deciding on which exploratory analyses warranted inclusion in an ad-hoc exploratory SAP.

RMarkdown works extremely well in these exploratory circumstances because of its inherent advantages with reproducibility. Having exploratory documents that combine reports alongside the source code that created the analyses ensures it is always transparent what methods were used to create the analysis. Two pieces of functionality that we particularly embraced were the ability to use cold folding within the generated documents and the ability to use tabbed content to display results. Cold folding meant that throughout the document the associated code for the reports was not by default visible and distracting to clinical colleagues who we shared the documents with, whilst at the same time statisticians or programmers where able to toggle the display of sections of code to review or verify exactly how data had been filtered, or what covariates had been included in models.
Tabbed content allowed us to display results or plots of related parameters side-by-side across tabs, or CSR outputs alongside exploratory sensitivity analyses. In this way, information dense and easily navigable summary reference documents could be produced and shared quickly.

Once the exploratory SAP was defined, formal outputs were created in SAS by the programming group. Since the amount of outputs in the exploratory SAP was reduced to outputs known to be helpful there was less overall work for the programming group. Furthermore, the RMarkdown documents themselves (or R code within them) were sometimes reused for QC or acceptance checking of some of the SAS generated outputs.

The success and possibility of following this kind of workflow will depend upon how well the direction of exploratory results was anticipated and permitted from the initial content included into the app, the speed at which gaps in app coverage and exploratory demands could be met and the confidence of the team in the unvalidated results display with in the app. It's not realistic to expect that Shiny apps will cover every question generated by the team, but good up-front planning and spending time trying to anticipate certain questions can increase the chances of success. Having a clean code architecture behind the Shiny app, with reusable code modules or R functions, can also permit rapid inclusion of further analyses, or tweaks to extend existing functionality within the app as needed.

As with any disruptive change to process, it was important to stay flexible with the changes and evaluate what is working well and where new pain-points are introduced. A specific learning experience was trying to use some of the content from RMarkdown documents directly as shell substitutes for the programming group when certain outputs were required to be validated quickly. Anticipated as a short-cut, this led to certain frustrations as reports that had been initially rapidly developed for exploratory purposes, may have required small modifications when translating to formal outputs or not contained full or consistent titles and footnotes, and by bypassing the creation and agreement of further shells some outputs got stuck in a long cycle of final tweaks. In the future, we hope to be able to define in a more formal way the exploratory process outlined here.

TIPS FOR SUCCESS

This section provides some advice for app developers working on similar initiatives. Particularly this section is aimed at newer Shiny developers familiar with the fundamentals of Shiny but with less experience of building out larger applications.

APP CONTENT AND THE ROLE OF AN APP DEVELOPER

The design and content of the app should always reflect the overall purpose of the app and the nuances of a trial. For a large phase III trial or program of multiple trials the study team is much more likely to be focused on population level analyses and easy partitioning of results by subgroups and the ability to rapidly extent the subgroup selectors available, either through an extensible ADSL style subgroup dataset or dynamic subgroup creation within the app are often extremely useful. Conversely, for smaller early phase trials value is more likely to be found in the ability to drill down from summary results into patient level timelines or narratives and to help quickly understand how individual patients progressed through the trial. From experience, apps developed around the use-case summarised here (supportive exploration around the first released results) are often developed under relatively tight development timelines. If starting from scratch or a limited existing codebase it is not realistic to attempt to cover the whole CSR, but to try to understand where the study team expect that the majority of the post database lock explorations may be spent.

It's also important to recognise the role of the project statistician as the natural link between the app and cross functional members of the project team. Typically, access to such applications is restricted to programmer and statisticians and the wider project team is exposed to the app through the endorsement of the project statistician. If the project statistician does not trust the apps or is reluctant to promote their use in meetings, there will be a much lower ceiling of potential success. From experience at different companies, there has always been an extremely strong correlation between interest and support of the project statistician at project initiation and the overall success of the app.

Additionally, it is extremely advantageous for at least one app developer to be as embedded as deeply into the study team as possible. Attending programming meetings in the run-up to the database lock will highlight potential data issues and any last-minute updates to ADaM datasets which maybe be critical to certain analyses. Idorsia is a smaller pharma company, and I was fortunate to be invited to meetings post database lock where the global head of drug development and other senior managers where shown the results for the first time. Hearing the reaction and the initial questions and discussion that the results triggered was invaluable in being able to anticipate the direction that exploratory analyses may lead and already being able to think of how to extend certain functionality within the apps to prepare for this. Typically, programmers are often excluded from such meetings, and if we want to be seen as more than service providers and show how newer ways in which we can add value to project teams participation in such meetings is critical. Conversely, experience has shown the having a single developer being parachuted in from a central group and working alone and with limited interaction with a study team will make it much more difficult to identify and target the most useful analyses for the team or make small tweaks that could make large differences to the value of the apps.

CODE ORGANISATION AND DESIGN CONSIDERATIONS

When working on larger Shiny projects, it is important to recognise the distinction between 'business logic' parts of the code that handle activities such as data ingestion, data-processing, summarising and report generation, and the

remaining server and user interface code logic that stitches the business logic together in the app and manages the reactivity that Shiny is built upon and the appearance of the app. Wherever possible the business logic should be extracted away from the *server.R* file (or the server component of Shiny modules) and into a collection of functions that may be tested outside of the app. Debugging shiny applications can be unpleasant and as much as possible you want to be testing and debugging your code outside of the Shiny application itself. This separation will also make it easier to update and add functionality outside of the app initially and allows the same functions to be easily reused in exploratory RMarkdown documents or other reports. If the app is very large, it is worth considering create a separate R package to house these functions.

We used the {golem} package to organise the apps into individual packages. Golem aims at providing an opinionated way to structure applications inside an R package that can help to install some good practices for code management particularly through the use of Shiny modules. However, it is important to appreciate the whilst using Shiny modules is a great start, it is not alone a recipe for successful separation of code, and without attention it is easy to slip into bad practices within the modules themselves.

When thinking about the user interface across a larger app it is important to ensure that there is consistency across different pages and tabs. This leads to an improved overall experience for users, and can also help to build trust in the applications. There are a lot of introductory resources available with tips and advice that can be helpful especially to SAS programmers with no experience of web development. It is also an important step to recognise the role of the app developer is not to simply to make the app as flexible as possible, as this can often lead to an overcomplicated interface with too many input selections that overwhelm users. Instead, developers should be mindful of the balance between providing users sufficient flexibility to be able to answer questions that may arise, but also sufficient constraints to clearly direct them towards meaningful analyses. This is a skill developed with experience, and it can often be instructive to observe how users interact with the app and understand where confusion arises.

CONCLUSION

The success of this pilot project demonstrated that there is clearly a space for RShiny applications for improving efficiencies as teams seek to develop deeper understanding of study results and begin exploratory analyses. With a relatively short implementation time we were able to deploy apps that were useful immediately as unblinded ADaM data sets were received and throughout the busy period that followed. Within Biometry statisticians were able to investigate and answer a diverse range of questions and within cross functional team meetings we were able to use the apps to provide immediate feedback to many questions. The success of the project has already set expectations that such apps will become available for future trials.

Contributing factors to the success were the suitedness of the Phase III program with the many efficacy parameters collected from different sources, the accuracy that we were able to anticipate the focus of exploratory analyses, and the modular structure of the app that enabled continuous additions to extend the capabilities of the apps. The focus on not slavishly reproducing CSR results, but instead focusing on contextualising important results was also rewarded.

Going forwards, we would like to tighten the relationship between CSR programming activities and output content within the app. At some point when developing or demonstrating Shiny Apps thorny issues surrounding validation arise. Whilst it's possible to envisage one day that such apps may be considered validated, and a primary source of result dissemination to teams, there are many steps to navigate to make that a reality. Initially, the path forwards would include increasing the unit tests around the maturing codebase behind the apps and exploring ways to automate comparisons between results generated in the app and CSR outputs.

ACKNOWLEDGMENTS

Thanks to Scott Pain and Chris Ridley at Idorsia for the consistent support, many good ideas and feedback during the project. The work and ideas summarised here also draws upon previous ideas and discussions with Mark Baillie from my time at Novartis.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at: matthew.brierley@ext.idorsia.com