Paper TT09

Heading to base camp on our way up the open-source mountain...

Andy Nicholls, GSK, London, UK

Abstract

Many companies are embarking on a multilingual programming journey that embraces a growing trend towards the use of open-source languages such as R and python for clinical reporting. In 2023, following a review of internal progress and external market trends, GSK Biostatistics decided to double down in its open-source commitments with two new goals to increase the prominence of open-source within the department:

- To transition tool development to use open-source languages.
- To achieve open-source code parity by end 2025.

These new commitments were set against a backdrop of successful internal initiatives and several ongoing industry collaborations led by the R Consortium and Phuse. These collaborations have been phenomenally productive. Whilst some unknowns remain, the journey for those starting in 2023 is certainly a lot easier than it was. It's only a matter of time until open-source languages become the dominant source in industry.

Intro

Open-source languages like R and python are not new. Most major pharmaceutical companies have been using them in some capacity for many years. But it is only within say, the past 4-5 years, that companies have really started to explore the full potential of using open-source languages for clinical trial analysis and reporting.

Why are companies leaning more and more towards open-source? And what does it take to get there? This paper seeks to briefly describe the benefits adopting open-source languages. And through a GSK case study, it looks at the systems, tools, and 'wow factor' case studies that need to align to make the journey.

Why Open-source?

It might not be the first reason that people think of, but one of the main benefits of adopting languages such as R and python for analytics is the extensibility that comes from their open package system. R and python are built from packages. Packages (aka libraries or modules) are self-contained bundles of code that help extend a language by offering new functionality: statistics / machine learning methods; database connectivity; graphics, etc. The languages grow and evolve through the contribution of new packages to popular repositories.

Both R and python have thriving packages ecosystems. First and foremost, packages are relatively quick and easy to develop. Second, the underlying open-source ethos means that those developing packages *usually* contribute their work back into the ecosystem for others to use. It's hard to say why so many contribute back. But there is a level of kudos / coolness associated with the more successful package developers. It's also a competitive market with individuals and organisations challenging each other to be the first to solve a particular problem. Whether it's a statistical method or a programming API, it's never long before someone has wrapped it up into a package for others to consume. Of course, this competitive landscape can lead to duplication. But competition can be healthy. The best thing about open-source is that everything is in the open! The better-written packages quickly rise to become the de facto standard.

From a pharmaceutical perspective, it has been both surprising and impressive to see the way in which the open-source culture has been able to thrive. Rightly or wrongly, the industry has a reputation for being cautious. Companies have spent years privately developing custom reporting ecosystems to try to stay ahead of their competitors. Yet in the last 1-2 years there has been an explosion of support for the Pharmaverse concept. Here, pharmaceutical companies and CRO's alike have joined forces to collaborate and share thinking around the next generation of clinical reporting tools in R. Open-source is infectious!

In addition to the collaboration and sharing culture, one of the benefits packages bring to the pharmaceutical setting is their structure. Packages enforce many good-practice behaviours and encourage others. The package bundle includes its own built-in help documentation, which can be further extended via additional 'vignettes' (eg a quick start guide). And they support unit testing frameworks that can be run at installation time to ensure that the package functions as the author intended. To release an R package onto CRAN (Comprehensive R Archive Network), the package must pass something called an "R CMD check". The *R CMD check* combines many smaller checks to ensure consistency in the package build. Is every function argument documented within a help file? Do the examples run successfully? Do the unit tests all pass (running on a variety of operating systems)? And so on. And the results are out there in the open for all to see.

The package structure isn't the only reason why open-source languages are growing in popularity within the industry. It cannot be ignored that academia is has been embracing open-source languages for several years. New methods tend to be developed in open-source languages first. And those leaving academia for industry are typically both well-versed in the use of open-source languages and keen to use them.

There are also language-specific benefits. The R (an also now python) ecosystem includes Shiny and Quarto, which have provided Statisticians, Programmers and Data Scientists with the means to create interactive, dynamic applications and reports. What's more, these are both reproduceable and traceable. And the languages themselves excel in their respective areas. R is a statistical language, written in the functional way that makes sense to Statisticians. Python strikes just the right balance between a general-purpose programming language and analytics, which makes it the go-to tool for machine learning.

In theory, open-source languages are also free. But to claim open-source is cost-free is misleading. The languages themselves may be free, but building new systems to accommodate them certainly costs money. As does retraining hundreds of members of staff. In the short-term, a move to open-source could even be considered expensive. However, the long-term benefits in terms of the competitive advantages they bring (and of course the lack of license fees) is a clearly factor in the decision to adopt them.

GSK Commitments

GSK Biostatistics consists of around 900 Statisticians, Programmers and Data Scientists working across Clinical Development, Research and Manufacturing. The majority of these employees are focussed on Clinical Development. Up until 2017, the use of R within Clinical Development was generally confined to graphics from a reporting perspective. Although the picture was more balanced when it came to study design, with teams taking advantage of R to implement novel Bayesian designs. Python use was negligible.

Since 2017, GSK has embarked on various initiatives aimed at creating a 'multilingual' Statistics, Programming, and Data Science community. In 2023, following a review of internal progress and external market trends, GSK Biostatistics decided to double down in its open-source commitments with two new goals. These were:

- To transition tool development to use open-source languages. Building the next generation of clinical reporting tools in open-source languages.
- To achieve open-source code parity by end 2025. By the end of 2025 >=50% of new code should be written in open-source languages.

To understand these goals requires some context about the GSK journey up to that point.

The GSK Journey

The GSK open-source journey has been ongoing for many years. But for Biostatistics, the first real commitments were made around late 2017 when it was decided to build a new in-house R training course, focused on the Tidyverse ecosystem. Eventually, three primary courses were written using R's bookdown format. Topics were centred on: basic data manipulation; graphics; simulation. Over time, further mini-courses have followed covering more advanced programming topics such as string manipulation, function-writing and Shiny.

Shortly after embarking on the training program, work began on GSK's first dedicated open-source language platform, WARP (Working Area for R and Python). Identifying an opportunity to promote the use of open-source beyond the current requirements, the team repurposed funding that was intended to build an HPC platform for study design work in R. Instead, they built a more general-purpose platform that included the Professional RStudio IDE and associated RStudio (now Posit) products: Package Manager; and Connect.

Perhaps the most important component of GSK's WARP platform has been Connect, which enables individuals to publish Shiny apps, rmarkdown documents and now Quarto artefacts at the click of a button. The WARP team made an early decision not to restrict publishing to GxP complaint output, which helped contribute to a wave of new innovative ideas. Amongst many impactful publications on the platform, was an application that proved to be a crucial component in GSK's internal Covid-19 response planning. Statisticians and Data Scientists worked with colleagues in Clinical Operations to construct an app that enabled study teams to re-model their recruitment projections based on ever-evolving Covid restrictions. The team were able to pull together various data sources, fit, and deploy predictive models via a working prototype within a matter of weeks. This simply would not have been possible using existing closed-source solutions. Such 'wow factor' examples have played a key role in addressing the 'why R?' question at GSK.

The adoption of open-source languages cannot *only* rely on the wow factor. To really embed open-source languages, it is important to show that they can also fulfil the needs of the 'day job'. The 'day job' in this case

means clinical reporting. Can they be used safely, effectively and efficiently within the niche world of clinical trial reporting?

One can easily find countless publications relating to the use of R or python for data manipulation, statistics and machine learning. But for many years, it was claimed that regulators would not accept the use of open-source. For their part, the FDA released their Statistical Software Clarifying Statement (U.S. Food & Drug Administration, 2015) in 2015 to effectively debunk that rumour. However, published examples of teams using open-source languages have been limited. At GSK, a small Biostatistics-led team was formed to address potential quality concerns via a project named "R4GxP". The *R4GxP* team built a new process for the assessment of internal and external R packages, adapted from the thinking outlined in the R Validation Hub's white paper (Nicholls, Bargo, & Sims, 2020) and the Modernization of Statistical Analytics (MSA) Framework (TransCelerate Biopharma Inc., 2021). To briefly summarise, the GSK package assessment is essentially a virtual audit. Measurable data about the software development lifecycle of a package is combined with a subjective review of package documentation and tests to evaluate the level of risk that a package poses. With the approval of our internal QA team, the process was formally released earlier in 2023 along with corresponding frozen (locked down) R installations to enable study teams to use R for their production programming.

Commented [AN1]: https://www.fda.gov/media/109552/download

For many companies, the main reason for sticking with closed-source solutions is that they are deeply embedded within their reporting ecosystems. These companies have built large suites of internal tools to support their workflows, with the corresponding Standard Operating Procedures and training to match. Such customisation for open-source simply did not exist at the start of GSK's journey. This is where package development played a key role. R package developers at GSK were able to build several new tools to support a revised clinical reporting process, including new tools based around the concept of Analysis Results Data (ARD). In the spirit of open-source, this development has largely taken place in the open, and has include collaborations with other pharmaceutical companies to build packages like: {admiral}, {metacore} and {metatools} for ADaM creation; and {tfrmt} as a tool to format Analysis Results Data.

The final piece of the GSK puzzle has been study team support. Along the journey, there have been various support initiatives such as "R4QC" where study teams were supported in their use of R as a QC tool. At the centre of the support model was a resource hub; an internal website built using blogdown and deployed on Connect. There is also an R 'Coders Corner' where individuals and/or teams can regularly compete to solve R-based challenges and win a prize. And latterly there has been "AccelerateR". With 80% of the department trained in R, focus shifted towards generic training towards hands-on support. The AccelerateR squad, works with study teams on a short term basis to provide additional bespoke training, template code and general consultancy and advice in order to get them up and running with R more quickly.

Beginning the Journey in 2023

For a company starting out today the journey to open-source is much easier. Stories are being told through organisations such as Phuse, PSI, and the R Consortium, and through vendors like Posit who have worked with their customers to release stories of open-source adoption such as the Novo Nordisk story from September 2023. The R Consortium in particular, has several working groups (R Consortium, 2023) dedicated to the use of R in the pharmaceutical industry and healthcare generally. The R Validation Hub's white paper (Nicholls, Bargo, & Sims, 2020) has been adopted and adapted by several leading pharmaceutical companies (2022 All-hands: Case Studies), demonstrating that the theory of the white paper can successfully be implemented. The Submissions working group are embarking on phase 3 of their journey with R-based submissions, having recently completed a submission with Shiny.

The Pharmaverse continues to grow and evolve under the guidance of Phuse, with 26 packages listed on the Pharmaverse website as of 24th October 2023. The Pharmaverse packages cover a broad range of topics such as the creation of tables, listings and figures; ADaM dataset generation; logging; formatting; and risk assessments. As new challenges arise, the Pharmaverse shows every sign of adapting (quickly) to these new challenges and providing useable solutions.

One challenge that all companies have encountered, and will continue to encounter, is numerical discrepancies between software. This occurs when two pieces of software produce different results when asked, apparently, the same question. Both may be correct in the sense that they are performing the calculations outlined in their help documentation. This is where projects like CAMIS come in. CAMIS started life as a Phuse project but has now evolved into a collaboration between Phuse, PSI and ASA. CAMIS seeks to explain differences between software. At present, the focus is on explaining differences between R and SAS, but the intention is to expand to include languages like python and Julia. These differences may be due to rounding methods or different default settings within the chosen statistical methods. Once the cause of a difference is understood, it can be explained.

In future, it is likely that protocols and analysis plans will need to be more agnostic to software, and hence more specific about the chosen method. The CAMIs project is the perfect place to start.

What about python?

Much of the focus of this paper is on R. Given its focus on statistical modelling, R is a natural first step for an industry focused on the implementation of statistical models. This is where the majority of open-source innovation has been to date. But python is also built on a package ecosystem. The aforementioned Transcelerate paper covers any open- or closed-source analytic software. And the general ethos of the R Validation Hub's white paper can similarly be applied to python.

R and python tend to be very closely linked when it comes to platforms. Both are built on similar package ecosystems with strong interoperability between the two. If you're building a platform for one, why not build a platform for both?! GSK started with a set of RStudio products. But as **R**Studio became Posit, so their products have evolved to enable both languages. Whether that's python functionality within the RStudio IDE or the inclusion of VS Code within Workbench. Other vendors are following a similar path, starting with python and throwing in R.

Conclusion: Into the Unknown

There has been a huge amount of progress in climbing the open-source mountain. Arguably, the industry has reached base camp already. But there's still a way to go. Companies are openly sharing their stories of successful submissions generated using in R. But these submissions are not without their pain points. It is not yet standard to generate a submission package using R (or python). As such, companies have reported a lack of familiarity with more niche, cutting edge packages, or R in general. And companies that have attempted to submit using internal packages have had to implement various workarounds to meet reviewer requests. It's also fair to say that these submissions have generally been limited to particular regulators. A round-the-world picture of R-based submissions is yet to emerge.

Until open-source submissions become more regular, doubts will no doubt remain for many. But the remaining obstacles are becoming fewer and further between. It is the opinion of this author that it's more a question of *when* and not *if* open-source languages become the dominant force in industry. Climbing mountains is not easy. Maybe generative AI will make us all redundant in a few years! But in the absence of such a force majeure, it seems inevitable that the trend towards open-source will continue and, eventually, dominate industry.

References

2022 All-hands: Case Studies. (n.d.). Retrieved from R Validation Hub: https://www.pharmar.org/casestudies/

Nicholls, A., Bargo, P. R., & Sims, J. (2020, January 23). Retrieved from https://www.pharmar.org/white-paper/

R Consortium. (2023). Retrieved from https://www.r-consortium.org/all-projects/isc-working-groups

TransCelerate Biopharma Inc. (2021). Retrieved from https://www.transceleratebiopharmainc.com/wp-content/uploads/2021/04/MoA-Initiative_MSAFramework_April-2021.pdf

U.S. Food & Drug Administration. (2015, May 06). *Statistical Software Clarifying Statement*. Retrieved from https://www.fda.gov/media/161196/download

Contact Information

Andy Nicholls
GSK
GSK House, 980 Great West Road
Brentford
Middlesex
United Kingdom
TW8 9GS
Email: andy.p.nicholls@gsk.com