PharmaSUG 2023 – Paper SI–202

Making Multilingual Programmers – A Targeted Approach to R for Clinical Trials Training

Jagan Mohan Achi, Jazz Pharmaceuticals;
Ashley Tarasiewicz, Atorus Research

ABSTRACT

Over the past few years many pharmaceutical organizations have encountered the same challenge – there is a wealth of training available on open–source languages like R, but very little training specific to the traditional clinical trial workflows we use on a daily basis. Companies are beginning to see and realize the benefits of having a multilingual programming team – they can incorporate the best parts of each programming language in their processes to maximize efficiency. In recent months, Jazz Pharmaceuticals has implemented several successful strategies to train their SAS® programmers in the use of R. This paper will examine the challenges and successes of finding the right training content, format, and candidates for teaching clinical programmers how to use R.

INTRODUCTION

The open–source for clinical programming discussions are everywhere these days, and at industry conferences, there are almost as many R presentations as there are SAS® presentations. It's a very exciting time, but also a little daunting as companies tackle the need to upskill teams of tenured SAS® programmers to R.

Over the past six months, Jazz Pharmaceuticals has worked with Atorus Research to train several cohorts of SAS® programmers in the use of R. The training program we began with is largely the same training program we currently use with a few tweaks based on feedback from learners who have gone through the program. The next few years are going to be full of learning as open–source finds a prominent place in clinical programming, but we believe we have found a successful approach to taking experts in SAS® and training them to use R for clinical trials.

TRAINING CONTENT

It's not difficult to find resources that teach how to program in R, but there is so much that R can do that those resources can be overwhelming. Plus, many resources are irrelevant to what a clinical programmer needs to know to complete their day–to–day job. Therefore, to effectively upskill learners who are already experts in SAS® and clinical programming, it's important to give them targeted material to focus on, and to leverage all the skills they already have.

WHICH PACKAGES TO USE

With over 10,000 user contributed R packages and so many that come with an R installation, it's a daunting task for a SAS® programmer to know where to start. When training, it's important to help learners understand everything that exists, but perhaps more important to give them a suggested place to start. In our training, we focus on base R packages, tidyverse packages, and pharmaverse packages.

The R community is full of passionate people, and there will always be a debate between the base R and tidyverse frameworks. Base R refers to all the functionality built into the R programming language, and the tidyverse is a framework of packages that add onto R. To be clear, we're not here to enter the debate, but to share our suggested initial approach for SAS® programmers.

In our training, we leverage base R to learn the fundamentals of R, but once the content moves from fundamentals to data manipulation, the training places an emphasis on tidyverse packages. Beginners tend to favor the tidyverse because it's simpler to read and use than base R. The design philosophy and syntax are consistent across packages and functions, and the key functions have descriptive names making reading code feel like reading straightforward instructions. Throughout our training, we cover each of the core tidyverse packages and how those packages can be used to complete clinical programming tasks.

In addition to base R and the tidyverse, we train on the rapidly evolving pharmaverse. The pharmaverse is made up of curated open–source R packages that enable clinical reporting from CRF to eSubmission. It expands on the tidyverse and provides targeted sets of functions to complete some of the most common clinical programming tasks such as creating ADaMs, display tables, adding metadata, and more.

Because the tidyverse builds on base R, and the pharmaverse builds on the tidyverse, we've found this selection of packages is the perfect starting tool kit for a SAS® programmer.

UTILIZING R DOCUMENTATION

Our training content provides the specifics for the traditional clinical trial workflows, but throughout our training, we've learned that to fully train SAS® programmers, we need to teach them how to use the documentation and resources they can find outside of our training as well. After all, part of being an excellent R programmer is being an independent learner, so our content also places a strong emphasis on showing how to use the R help pages and package websites to learn more about the packages we cover, as well as all the other packages we don't cover. Furthermore, the pharmaverse, in particular, is very much still evolving, so in the training we discuss how to monitor change logs and use package documentation so that learners can leverage their starting knowledge to self–train as the pharmaverse continues to progress.

APPLICABLE CONTENT

The training groups consist of people who are already experts in SAS® and clinical programming. That can be a challenge in that sometimes there's resistance to learning a new skill, but with the right content, we can leverage the skillset they have and translate those skills, not start from scratch.

The best way to get learners started is utilizing side by side comparisons of SAS® and R code. R makes much more sense to a SAS® programmer when you can show them that to replicate a SAS® MERGE statement, you can use the R dplyr mutating join functions. Or to replicate the results of a SAS® PROC TRANSPOSE you can use the R tidyr::pivot_wider() and tidyr::pivot_longer() functions. When the first few hours of training content cover transitioning SAS® knowledge directly to R knowledge, learners immediately see R material they can relate to things they do every day as a SAS® programmer.

Once SAS® skills are translated to R, it's important to guide learners to stop thinking like a SAS® programmer and immerse themselves in the R world – but to still leverage a language they know. If we're saying move away from thinking like a SAS® programmer, what language does that leave? Remember, you have SAS® clinical programming experts, so we utilize their clinical programming language expertise by using CDISC data sets throughout the training. So many R books show examples with random data, and learners don't want to practice with data sets that contain the make and model of a car, or the color and width of flowers. By using CDISC datasets when learning R, learners see examples that use a language they’re familiar with, which helps them constantly see where R can fit into their work.

PRACTICAL CONTENT

Any learner or company leader probably has one central question – what can someone do at the end of all this training? The content in training not only translates SAS® knowledge to R and uses CDISC data to introduce the packages, but it also covers task–specific things such as creating datasets and displays. By the end of the training, a learner has seen how to use base R, the tidyverse, and the pharmaverse to replicate some of the most common clinical programming tasks, such as programming SDTMs, ADaMs, tables, and figures. Beyond that, they know how to utilize documentation to learn even more about the rapidly evolving open–source R world, as well as the ability to see for themselves the strengths of SAS® and R and where each can give them efficiency in their everyday tasks.

TRAINING FORMAT

Let's face it, SAS® clinical programmers are busy. We're asking them to learn on top of everything they already have going on for projects, so how can this be interesting, engaging, and best fit into their workday?

LIVE GROUP SESSIONS VERSUS INDIVIDUAL SELF–PACED ELEARNING

With any training, there is often the discussion of live group sessions versus individual self–paced eLearning. Everyone learns in different ways, and both have pros and cons. In our experience, we found people do not make the time when they're asked to learn on their own – it's too easy for other obligations to take precedence, and learning takes a back burner. Because of this, we utilize live group sessions allowing us to join people together and keep training a priority. In modern times, live can mean in–person or virtually – it often depends on the learners' locations. If all learners are in one place, you can maximize focus with in–person sessions, but in the remote landscape many companies work in, live virtual sessions certainly work and are the best choice logistically.

As we learned through feedback, not everyone thrives in a live group environment. Therefore, to account for the people who may prefer individual self–paced learning, we also provide an online eLearning platform with content recordings, quizzes, and programming exercises, so learners can optionally use those materials to complement the live group sessions.

SESSION FORMAT

During a session, we found success by splitting it into two parts – a lecture and a guided practice. The lecture introduces the topics, shows examples, and provides a way for learners to ask questions. But most learners don't fully understand a concept until they can practice, which is where guided practice comes in. In guided practice, learners are given exercises to practice the concepts directly after learning them. By having a live coding session trainers can show demos of concepts in real–time, and more importantly, learners can practice programming while being able to ask questions to the trainers. The guided practices allow for additional clarification when learners put the concepts into practice, and the group setting allows everyone to gain from another learner's issues. Additionally, practicing in the session will enable learners to get enough out of the training during the live session. Any additional time learners put in outside of the live session only enhances their understanding.

INFRASTRUCTURE

Many companies don't yet have a shared R environment, so how can R training happen? During the guided practices, we utilize Posit Cloud. This ensures the training can begin regardless of where a company stands regarding a shared environment. Additionally, using Posit Cloud allows the trainers to install any necessary packages and load some starting files, such as data and helper code so that learners can focus on the material, not the environment itself.

FREQUENCY AND DURATION

Another big question, what is the time commitment? Ultimately, that depends on the group of learners and their project workload. If a team has a known lull in programming work or maybe an urgent need to start taking a multilingual approach, they can go through training fast and furious with half or full–day sessions one or more times per week. In our experience, that leads to learning burnout as well as challenges of making up any missed sessions. On the other hand, groups going through at a very slow pace with one or more weeks between sessions have found it difficult to keep the momentum going and typically see a lot of drop–off in attendance towards the end when people are tired of the learning dragging on. We've found the most success in 2–hour sessions every week. 2–hour sessions each week are frequent enough to keep the material fresh in a learner's mind, but sets a pace that if someone misses a session or wants to do some additional learning following a session, they have time before needing to attend the next session.

TRAINING CANDIDATES

Our team of candidates has an average of 10 years of experience using SAS® software. In addition, they have extensive submission experience and are skilled in SAS® programming, with some members also knowledgeable in other programming languages. The goal is to become an innovative group that caters to the needs of stakeholders by providing them with cutting–edge solutions. We are always eager to learn and grow, constantly seeking new challenges and opportunities to improve our skills.

Interactive applications are one of the key objectives, and we are driven to make this change a reality. We believe that using interactive applications can revolutionize how stakeholders interact with data, making it more accessible and user–friendly. With our team's expertise in SAS® programming and training in other programming languages, we can be well–equipped to create these applications and make them a reality.

CHOOSING THE RIGHT PEOPLE

Choosing the first cohorts among the team mentioned above was a significant decision since learning a new skill is a big change. We wanted to find the early adopters and give them an opportunity to select their cohort from the training schedule. By empowering the choice of cohort, we hoped to find the early change adopters for the R training and have them become the champions for the success of the training. Our aim was to create a supportive environment where the selected cohorts could learn together and provide feedback to improve the training program. The process of choosing the first cohorts was a collaborative effort that encouraged engagement and participation from the team. By involving the team in the selection process, we were able to identify those who were enthusiastic about learning R and who had the potential to become advocates for the training's success. We were confident that the early adopters of the R training would be the catalysts for driving change within the team and creating a culture of continuous learning.

HOLDING LEARNERS ACCOUNTABLE

Holding the team accountable for their learning is a crucial aspect of our approach to training. We recognize that attending classes is not enough and that the new skills acquired during training sessions can quickly be lost without continuous practice. To address this issue, we established some guardrails to ensure accountability for learning. We provide continuous handouts and practices that the team could use to reinforce their new skills and knowledge. We also set an expectation that our team will become a multilingual place with all the skills needed to function effectively within the department. We expect that our team will learn and use the new language in their everyday practice. By setting these expectations and establishing accountability measures, we are able to create a culture of continuous learning where everyone is responsible for their growth and development. Our team embraces this culture and takes ownership of their learning, which results in improved skills and a more dynamic and effective team.

WHAT WE'VE LEARNED

TRANSLATING SAS® SKILLS AND IMMERSING IN R WORLD

Many people think it's enough just to translate SAS® skills, but that's just the start. The goal should be to create multilingual programmers. We want companies with programmers that look at a problem and realize they have multiple programming languages in their toolkit to solve each problem. Learners mustn't program in R merely replicating what they do in SAS®. They need to know the R language well enough to identify where it can be used instead of SAS® for a particular task or perhaps to complete a task SAS® could never do.

REPETITION THROUGH VARIOUS TYPES OF MATERIAL

Few people learn the same way, and few learners start in the same spot. While the overwhelming feedback is that live sessions are an excellent jumpstart, learners need other resources to enhance those live sessions. Session handouts of key concepts are vital for many people in allowing them to preview the material before a session, and review it afterward. Session FAQs are helpful to several people as a way to look into answers to questions they either had or never even thought to ask. And access to an online eLearning platform in addition to the live sessions is critical for most in that it allows people to catch up if they miss a live session and expand on topics a learner wants to learn more about.

PRACTICAL EXAMPLES

Overwhelmingly, what learners have said they like the most is how practical the content is. Learners don't want to learn building blocks and use them to build a random pile – they want to learn how everything works to build the actual house they want. If what they're doing isn't connected to a payoff, they lose the stamina to wade through the problems. Examples must be applicable to both leverage what they know as a starting point and make the learning feel important and relevant.

FUTURE PLANS

While the training has certainly been an excellent starting point for making multilingual programmers, this is just the beginning. Once learners are introduced to R for clinical programming, subject matter experts or open–source enthusiasts can be identified for more advanced training to enhance their R skills further. After SAS® programmers have translated their skills to R and can replicate everyday clinical programming tasks, they can build on them and utilize R to do things outside the typical clinical pipeline. Furthermore, R is just one programming language. The whole concept of a multilingual programmer is multiple languages, with so many more to learn. Once learners have both SAS® and R under their belts, they'll be able to leverage their well–versed programming skills to learn additional languages in a fraction of the time.

CONCLUSION

There are many effective ways to train, however, when you need to train people who have done the same thing for years to do something totally different, there are ways to make the process more effective. When training SAS® clinical programmers in R, the best training programs will leverage their expertise. You have to provide a translation of skills rather than an introduction to something completely different. And to make the training click, it has to be relevant. You can't make busy learners waddle through a lot of generic stuff to get to the applicable things – training has to show them how they can use R to do the things they do daily, but also give them the tools to take R beyond that. R is not replacing SAS®, but R adds to a clinical programmer's tool kit to maximize efficiency and allow them to pick the best solution for the problem.

CONTACT INFORMATION

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

Jagan Mohan Achi
Jazz Pharmaceuticals
jaganmohan.achi@jazzpharma.com

Ashley Tarasiewicz
Atorus Research
Ashley.Tarasiewicz@atorusresearch.com
https://www.linkedin.com/in/ashley–tarasiewicz–6bb202132