

## Overview

Are you starting your “R” journey? Are you struggling with where to begin?

If so, then this article is for you. It will explain, at a high level, the basic steps for introducing R into a life science organization.

Clearly, every organization is different, and will ultimately need a roadmap tailored to its own unique needs. Yet, it is possible to outline some general steps that will apply to any organization, and to give you a sense of what the journey to R will look like.

### STEP 1

#### Assemble R Core Team

The first step any organization will need in their journey to R is to assemble an R core team to drive the adoption of R. This team will make the necessary decisions to set up the technical infrastructure, outline new roles, identify needed skills, and make recommendations on hardware and software. The team should include representatives from business, IT, and at least one person that has some familiarity with R.

The team should be small enough so that it can make decisions quickly, but large enough to cover the skills needed to accomplish the project. If there is literally no one in your organization who has any familiarity with R, you might consider hiring someone, or reaching out to a consulting organization for guidance.

### STEP 2

#### Develop a Training Approach

It is important to develop a training approach early in your R journey. Your core R team will benefit from a consistent learning approach and early R learning will better inform your objectives and activities. Your larger team will need to start upskilling as soon as possible and having a consistent vehicle for continuing R learning is key.

Training in R can be a significant challenge since R is not yet widely used in the clinical programming space, and there are no established traditions on how to conduct it. In addition, people who know clinical programming are typically fluent in SAS®, and R is a very different language.

The best R training programs will also leverage your team’s existing SAS® expertise. It is easier and more effective to take the learner from what they already know to what they don’t know. This method will ease the transition to new knowledge and improve retention.

### STEP 3

#### Set Up Pilot Environment

The next step is to set up a pilot R environment. The pilot R environment will be used to evaluate packages, products and architectures. It will also be used to begin growing more experience with R.

The R pilot environment will not be validated. It should not be used for any production work. The purpose of the pilot environment is to give the R core team a sandbox for figuring out what the production environment will look like.

There are many, many decisions to make. The R pilot environment will give you a space make those decisions.

### STEP 3a

#### Install R

The first decision you are going to have to make is which distribution of R to use. There are several. Are you going to use CRAN, MRAN, or Bioconductor? There are also third-party, validated versions of R that are popular with life science organizations.

This decision is significant, and will have far-reaching implications on the capabilities of your R environment. Ultimately, it will determine how your R programming team will work, and which tools they have available to them. So, this decision should not be made lightly. You may want to play around with different distributions, and consider the advantages and disadvantages of each.

# Roadmap to R

## STEP 3b

### Decide on an Initial Package Set

Help narrow your decision on the R distribution by considering which R package to use in your R environment. There are thousands of packages to choose from. Which will you use and what is the initial package set?

The consensus among life science organizations is to go with popular, well-established, and well-documented packages. Many organizations rely on packages supported by foundations, commercial enterprises, or other life science companies.

For instance, the Tidyverse family of packages is largely supported by developers who work at RStudio. These types of packages tend to have better design, better testing, and better documentation.

However, these packages may not cover all your needs. To fill in the gaps, you may need to go outside the package set provided by your R distribution and Tidyverse. You may even need to develop your own packages.

In either case, the pilot R environment will come to the rescue. The pilot environment will give you a space to evaluate or develop any additional packages, and determine which set is right for your organization. The general strategy here is to define a set of immediate goals for your R environment, pilot those goals, and let that determine the initial package set.

## STEP 3c

### Decide on a Package Management System

R is an open source software built by thousands of programmers working independently. The packages are constantly changing, and they are all on different release schedules. Functions are being added, removed, and changed all the time.

A life science organization, however, requires that the statistical software remain stable for the life of a study, and even long afterward. **So how can you ensure a stable software environment when R itself is moving?**

**The answer is with a package management system.** A package management system allows you to “lock in” a set of packages of a particular version for a particular study. Two popular package management systems are renv and pacman. Yet there are others.

Evaluating and deciding on a package management system is something every organization moving to R has to consider. It is the only way to create stability with open source software that is in constant flux.

## STEP 3d

### Decide on a Source Code Repository/Versioning System

Most organizations already have a source code repository or Apache® Subversion® with stakeholder agreement. Historically, life science organizations rely on Subversion or CVS. These systems will also work with R.

However, the open source movement, and **R in particular, is increasingly consolidating on Git and Github. Git has many advantages for team-driven development.** It allows branching, merging, rollbacks, comparisons, and control over which changes are committed. Git is also integrated into RStudio.

While it is not necessary to use Git and Github with R for versioning, you may want to consider their advantages and disadvantages relative to your current version control system. Again, the pilot environment can help you evaluate these tools and decide whether or not you want to make a change.

## STEP 4

### Begin R Development

Once you have a validated R environment, **you will very likely want to begin development on some R tools** and utilities.

Tool development is typically done by a development team. Ideally, the development team will be made up of one or more people from the R core team that helped design the R environment. Depending on the number of tools you want to develop, and the speed at which you want to develop them, you may need to enhance the development team with additional resources.

You must also decide where you want to begin. With a new, unproven language, most organizations are hesitant to use R for production programming work. Many companies decide to begin with QC tasks. Once they build confidence and expertise in QC, then they move into production activities.

Very often, organizations spend considerable time with a small-scale R operation, running mostly behind the scenes. R usage may not spread beyond this small team for a year or more. Yet with increasing confidence comes the desire to broaden the base of R users.

# Roadmap to R

STEP  
5

## Set Up Production Environment

By the time all of the above steps are complete, you will have a very good picture of what you want your production R environment to look like. The next step is to create it.

Many organizations choose to install R on their existing servers. **It is reasonable to install R on your existing servers** because, generally, R does not interfere with other software. So, the likelihood of breaking your existing environment is low. Other organizations, however, decide to create a completely separate set of servers to host their R environment. In the end, the decision will be based on the particulars of your organization and the people involved.

In either case, the end result is the same: to create DEV, TEST and PROD R environments that will support the R goals of your organization. The best installation strategy is to set up the development environment first, perform compatibility testing, and then move up to TEST and PROD. If you do not have the expertise to validate the system, there are contractors that can help. There are also nonprofit initiatives like R Validation Hub that can provide some guidance.

## CONCLUSION

Introducing open source technology into an organization holds many promises. It promises to lower licensing costs, increase performance, and tap into a rapidly evolving technology sector.

In a life science organization, however, any introduction of new technology must be carefully executed. The approach outlined above attempts to lay out a roadmap that minimizes risk.

The roadmap recommends first establishing an R core team, then setting up a pilot environment from which you can make decisions on your distribution, packages and support systems. Once these decisions have been made, you are in a position to proceed with setting up your production environment and beginning some tool development. As comfort with the technology starts to settle in, you are ready to expand R to a broader population via a high-quality training program.

Of the R training programs available, only the Experis Accel2R program takes this approach. This program was designed specifically for training existing clinical programmers in the use of R. It is currently the gold standard for upskilling clinical programmers and is being adopted by life sciences organizations of all sizes.

Hopefully this roadmap will give you some sense of what the journey to R looks like in a life science organization. If you would like to discuss this topic further, **please contact us**.



Experis®  
ManpowerGroup

### About Accel2R by Experis® Solutions

Experis Solutions, part of ManpowerGroup, has been providing timely, high-quality, and cost-effective services to clinical organizations since 1982. Accel2R is an accelerated learning program developed by Experis Solutions that's designed for clinical organizations (i.e. pharma, medical devices companies, and customer research organizations) to upskill their teams to

R. Accel2R uses advanced learning techniques, a clinical-specific curriculum, learner AI, a 24x7 learning lab and a unique SAS® to R learning approach to transition from SAS® to R in 4 to 6 months.

Learn more at <https://accel2r.com>