

Changing the Language of Problem Solving:

5 Learning Challenges Moving from SAS® to R

Accel2R

Powered by Experis® Solutions

Overview

One thing we've learned in 2020 is that the tools you use to get your work done affect the way your brain thinks. This includes how you communicate, approach and solve problems. If you change a tool, you will likely need to reprogram your brain to effectively get your work done.

If you've spent any time programming in SAS®, you might say your mind has been “wired” to solve data and analytical problems by thinking in SAS® code. Any time you need help, you know who to ask and where to look for documentation and answers to your questions.

SAS® software has been the gold standard for large companies that need to take data from essentially any source, twist and turn it, and produce analytical results. There are SAS® customers all over the world that pay an annual fee to use the software. SAS® has their headquarters in Cary, North Carolina where they have hundreds of computer scientists, statisticians, marketers and support staff.

Whereas R software has traditionally been popular in academic settings, R has recently been growing in popularity and acceptance in other industries, particularly in the pharmaceutical industry.

For those moving from SAS® to R, there are similarities and differences.

Similarities

- Both are scripting languages that are interpreted from the top to the bottom of a program.
- The programs are stored in ASCII files that can be read by a text editor.
- There is an IDE which allows you to develop code, execute it step by step, and view intermediate variable and data set values:
 - SAS® has SAS Display Manager®, SAS Enterprise Guide®, or SAS Studio®.
 - R has an IDE called RStudio.
- Both have a proprietary data table format.
- Both can leverage an SQL language:
 - SAS® has PROC SQL and R has the sqldf package.
- Both have advanced analytics capabilities.
- Both can be run locally on your desktop or leveraged from a server.
- Both can be used in Windows, UNIX, Linux, or Mainframe operating systems.

Differences

SAS® has modules that you must license and pay to use on an annual basis.

SAS® has a proprietary data table format called SAS Data Sets®.

SAS® data sets can have variable types of numeric and/or character.

SAS® is primarily procedure-based.

SAS® has the SAS Data Step®, which allows you to step through the data record by record with complete control.

SAS® has global program variables called SAS Macro Variables® which are character-based.

SAS® is developed, maintained, documented, and supported by a central group in Cary, North Carolina.

SAS® has the Output Delivery System which can output SAS® results to HTML, PDF, and Word very easily.

SAS has SAS Macro® for development of reusable code which has weak scoping for data and variables. There are methods for creating actual functions in SAS®, but this is less popular.

R has packages that you can download, install, and use for free.

R has a proprietary data table format called Data Frames. They are similar, but R has more variable types and the variable attributes are handled differently.

R can have variable types of logical, numeric, date, time, and character.

R is primarily function-based.

R does not have a similar functionality.

R has global variables, vectors, and lists which support different variable types such as logical, numeric, date, time, and character.

R is developed, maintained, documented, and supported by a disparate group of developers that submit their packages to CRAN for review.

R has R Markdown to output results to HTML—but getting results into PDF and Word files is traditionally very challenging. However, there is development currently underway to address this.

R does not have the “code substitution” functionality like SAS® but leans more on methods for users creating their own functions.



5 Learning Challenges Moving from SAS® to R

Learning Challenges

Navigating how to use a new IDE to develop code, view variable values, data and results; and interpret error messages comes with learning challenges.

1. Debugging Code:

Base R does not generate a log that is as nice as SAS®. You will need to adjust to the types of errors and warnings that the console and script editor point out. As you get used to the syntax of R, debugging code will get easier, but it can be frustrating at first.

2. Saying Goodbye to the Data Step Mindset

The SAS® Data Step does many things behind the scenes. If you have been programming in SAS®, you may lean on the SAS Data Step® to handle data management and reporting.

3. Letting Go of SAS® Macro Behavior

You may currently find value in leveraging global variables and developing reusable code. In SAS®, you can lean on SAS Macro®. Although SAS Macro® definitions and parameters are stored and passed in a structure that appears like a sub-routine or function, SAS Macro® is just text substitution before compile time. R functions behave like a more traditional sub-routine.

4. Getting Your Results Into Publishable Reports

SAS® procedures such as TABULATE and REPORT make it very easy to create complex reports with functionality including spanning headers, automatic paging, column wrapping, etc. The Output Delivery System (ODS) in SAS® makes it easy to reroute program results to HTML, PDF, and Word. Most of this functionality is relatively new in R but maturity is expected to increase quickly.

5. Getting Statistical Results to Match Current Industry Standards

It may seem pretentious to call SAS® the industry standard, but in many industries, such as pharmaceutical, this is traditionally the case. These organizations have depended on SAS® for analytic results for years and are expecting those results to be consistent moving forward. For organizations that are implementing R and needing analytics results to match results in SAS®, there will be challenges.

Such as:

- a. There are similar packages and functions in R like the modules and procedures in SAS® that perform analytics. There is extensive documentation provided by SAS Institute® that compares across analytical procedures that you may find lacking in R. Unlike R, SAS® also has technical support that connects you to an expert to help troubleshoot the procedure and analysis you are working on.
- b. Behind the scenes in SAS® and R, there are settings for numeric precision, convergence criteria, and rounding. These differences need to be uncovered and rectified for SAS® and R results to match exactly. These differences are not always easy to uncover as there is not a central organization in R that keeps track of these settings to ensure consistency across packages and functions.

CONCLUSION

Learning to program in R is not trivial after being immersed in SAS® for a significant amount of time. It is like moving from a 4th generation language to a lower-level language. Writing code to accomplish equivalent processes in R that you did in SAS® is not always an easy mapping. Having access to step-by-step training with familiar examples along with hands-on exercises is essential for an efficient transition.



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 24/7 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>