Paper 348-2013

The SAS® Versus R Debate in Industry and Academia

Chelsea Loomis Lofland, University of California, Santa Cruz;
Rebecca Ottesen, California Polytechnic State University, San Luis Obispo

ABSTRACT

Despite industry being heavily dominated by SAS, R is used widely in academia due to being free and open source software that is structured around users being able to write and share their own functions. However, this disconnect leaves many students who are pursuing analytical degrees struggling to get a job with less SAS experience than desired by companies. Alternatively they could face the struggle of transitioning everything they learned in university from R to SAS. Ideally one would know every possible programming language and use the one that best suits the situation. This is rather unrealistic. Our goal is to show the benefits of these two very different software packages and how to leverage both of their strengths together. We also address many of the misconceptions we often see with R users who have left the SAS programming language years ago and have not kept up with current SAS trends and advances.

INTRODUCTION

We chose to focus on SAS and R as they currently dominate the programming language choices in the statistics field. We've noticed that there is a disconcerting trend in that R is being used heavily in Academia which seems to be at odds with what is going on within industry where SAS is primarily utilized. However the mastery of both packages is critical to a young person's success as each language plays an important role in analysis. Therefore taking a one sided preference for either software based on common misconceptions does a disservice to students and needs to be addressed.

There are several updated technologies in SAS that statistical programmers from other languages seem to have yet to discover because they are not keeping current with software. Graphics are a good example of an area SAS has been quickly excelling in, yet many people are unaware of these new advances so they stick to R alone just for its graphics. Another example of an area of SAS that is somewhat unknown is the possibility and ease of writing functions, which is a strength of R. The procedures in SAS are thoroughly vetted and have stellar documentation and tech support; however a new user might not understand the tools that are available or even know they exist. Additionally, SAS has excellent training courses, web based and user group based resources, and a plethora of books on a wide variety of subjects. Knowing about these technologies and tools, and how to use them appropriately can ease some of the fear of using SAS.

THE ISSUES ADDRESSED

The common misconceptions we come across at our academic institutions left us wanting to address the pros and cons of both languages. There could very well be even more issues at play, but the ones we chose to focus on seem to be the most common. We hope to dispel these and possibly provide new information to those who have not kept up with R or SAS.

NEW ADVANCES IN STATISTICAL METHODOLOGY

SAS

- PROs: Software and algorithms are thoroughly tested and SAS has tech support to address user needs quickly. SAS tries to implement new methodology into existing procedures when it makes sense to do so, as an option or additional statement, so that users do not need to learn yet another PROC. SAS also releases newsletters detailing new advances in the software.
- CONs: The delay in new advances until the next release of the software.

R

- PROs: Users can implement new methodology quickly or find a package from someone who already has. Easy to teach and understand the new methodology because students can see the functions right there in the code.

- CONs: The documentation of the new advances in R is user-based, so new methodology is not well organized and often has not been thoroughly tested. The developers are spread out rather than working together locally.

For this case the cons of one are really the pros for the other. In the case of R some would argue that they can see exactly what is going on in the code, and this is true if the user has the background to understand it. However for SAS while the PROCs are pre-packaged, the documentation has an enormous amount of detail about the mathematics involved for every single statement and option. If a user really wanted to get under the hood, the detail is all there and very easy to access in the technical documentation. A student or user blindly running code should be a concern in both languages. Someone who would run a PROC without understanding what they were doing would most likely run a pre-written R package without paying attention to the functions it was calling.

GRAPHICS

SAS

- PROs: The graphics in SAS are becoming increasingly flexible, polished and easy to use. In some PROCs, turning on ODS Graphics will automatically produce graphics without requiring additional code. This gives users a choice of using the defaults graphs or creating their own.
- CONs: The template language behind the graphics can be overwhelming and somewhat difficult to use, especially for beginners. New advancements, such as interactive graphs, can also still be difficult for beginners.

R

- PROs: Easy to create polished graphics. Graphs can also be run through loops to create animations.
- CONs: R graphics are not attached to statistical analyses, so graphs and analyses take place separately. The user must decide which graph is most appropriate, which can be beneficial or not depending on the user's background in statistics. This is more subjective to the user's preference, though modifying the graph to achieve specific dimensions or aspects is not necessarily an easy task.

Prior to version 9.2 of SAS the graphics were one of the main reasons that people gravitated toward R. One of the best features of R has been, and still is, the quality and ease of its graphics. However the current state of SAS/GRAPH with ODS graphics and SG procedures are have revived the graphics capabilities of the software. Using ODS graphics with the PROCs allows users to easily generate graphics that relate to the analysis that they are performing. New PROCs for creating more specific graphs, such as PROC SGPLOT, SGPANEL, and SGSCATTER are quickly growing; however they require a reasonable amount of coding. Additionally there are other great graphics options in SAS such as SGDESIGNER and SAS Enterprise Guide.

FUNCTIONS AND REUSABLE CODE

SAS

- PROs: SAS has an extensive function library as well as the ability to write user based functions that can be called in DATA and PROC steps. Also the macro programming language with massive scope and capability in which variables can be global or local.
- CONs: Writing user based functions and detailed macros requires a fairly extensive amount of programming knowledge to ensure accuracy.

R

- PROs: In R the ability to write any function is easy. Users can share functions with others by submitting to R-CRAN.
- CONs: Writing user based functions requires a detailed amount of knowledge to ensure accuracy. Variables are strictly local in scope.

In this case the two languages have virtually the same pro and con. In the past SAS users who wanted to run their own functions relied heavily on the macro programming language to get the job done. This was seen as ineffective and clunky by R users. However as of SAS version 9 PROC FCMP allows users to write their own functions, and as of version 9.2 these functions can be called in DATA steps as well as PROCs. This is useful for simple statistical functions, and also more complicated statistical functions which can be implemented using the IML language. Both languages face the issue of being able to carry functions out correctly and in an efficient manner. This requires

detailed knowledge of the process of which the user is trying to capture in the function. This is good from the view that a programmer needs to know what they are programming, yet dangerous in the sense that others can download a SAS macro or R package and use it while having no idea how it works or if it is working correctly for them. However with proper understanding the sharing of functions and macros makes extending them to particular needs very handy.

FREEWARE

SAS

- PROs: SAS has an OnDemand version of its software that is free to degree granting institutions.
- CONs: SAS proper and JMP are not free. OnDemand has limitations on which operating system it will run, and has been reported to be slow at times.

R

- PROs: R is free.
- CONs: Open source software can be a security concern for large companies.

The free alternative offered by SAS to universities is promising in terms of keeping it a viable option for professors to use in the classroom. The installation process and speed of OnDemand needs attention. That said, SAS and JMP are still not free and a company would still need to license the software. R can be installed for free. Many bloggers covering the debate have pointed out that to convert an existing company who uses SAS to R would be a waste of resources and money that far exceed annual licensing costs. The re-writing of code, transitioning the team, hiring new experts, etc…Also companies that have strict analytical vetting requirements would be better suited toward SAS. Small companies with no existing analytical infrastructure could take a hard look at whether to invest in licensed software with extensive history and resources, or go with free software that would require somewhat of an investment in advanced knowledge of the staff to create, manipulate and run the coding. In the end, time, money and these two options may very well break even.

USER SUPPORT

SAS

- PROs: SAS has extensive online documentation, expert technical support, professional training courses, many excellent books in press, and a tight knit user group and web based community. Problems can be addressed to SAS directly via tech support who replies very quickly and will work with the user to solve the problem.
- CONs: We really couldn't think of any.

R

- PROs: R has good books by example, online user documentation, R mailing list, and R meet-ups.
- CONs: Users rely on what others put out there about the software. There is a disconnect in the world-wide user group because the developers are so spread out. Packages are not written by the R Development Core-Team therefore they are not well polished and some could have questionable validity. It is also difficult to direct an issue to a particular person or support system.

The excellent support provided by SAS highlights its customer oriented design. The strength of SAS's support makes it ideal for beginners, and its extensive detail is useful for advanced users. R's disorganized documentation and lack of technical support make finding help a challenge. This is a big trade-off for the developer oriented design of R.

DATA MANIPULATION

SAS

- PROs: SAS can handle any data set or data type. DATA steps are designed purely for data management and therefore SAS excels at it. SAS handles big datasets well, with a variety of options for working with them; merging and PROC SQL have a quick run times.
- CONs: Coding can be a change of thinking in that SAS has an underlying loop for the DATA step.

R

- PROs: R is starting to be more capable of working with big data. It is designed well for matrix manipulation and has fast sorting times. R also excels at simulation based analysis.

- CONs: The design of R was focused around statistical computing and graphics, so data management tends to be time consuming and not as clean as SAS. One of the main reasons for this is the large number of different types of data structures which can make data manipulation in R a more difficult concept to grasp.

Somehow data management has been overlooked as an essential aspect of statistical programming, yet it is extremely important since real life data is very rarely clean and ready for analysis. Students who have used solely R have an unrealistic expectation of the state of the data they receive. Learning SAS is an excellent way to learn how to combat more realistic data sets. SAS can manage and analyze big and messy data, whereas R is mostly centered around analysis.

The object oriented data structure of R can prove problematic when dealing with messy data and it does not have an internal looping structure. Merging complicated datasets with large amounts of missing data then creating and modifying variables is a common use of SAS that utilizes standard tools. However, doing complex data manipulation in R is not standard and can turn into a nasty process.

Which software has better run time between SAS and R seems to depend on the task. Options such as MEMLIB can be set in SAS to use main memory as R does, rather than the hard drive to improve run time. R does not have this hard drive memory option and must use main memory.

INSTALLATION

SAS

- PROs: SAS comes as a packaged install with the components that are specified according to analytic needs and licensing. The updating of the license is very easy.
- CONs: The initial installation of SAS or updating to a new version of the software can be long and difficult. Magnify this by 1,000 in the context of explaining this process to a group of students new to the software who just want to run it on their laptops for class. Proper SAS also does not run on Macs, at least not easily, which are becoming an increasingly prominent laptop choice in the classroom.

R

- PROs: R is easy to install and runs on Windows, Mac and Unix. R installs very quickly. This is also true for RStudio, a common and useful user interface for R that many find preferable to R.
- CONs: You have to know about the packages that will meet your needs, seek them out, install them, and then research what they can do. As of this writing there are 4,379 packages available (and growing every day) which, while providing many options, can be difficult and time consuming when searching among all of them.

SAS can be difficult for users to obtain and the initial installation is sometimes tricky. However once it is installed problems with the software are rare and there are no additional packages or steps for specific analyses—those are already installed. R is the opposite, it is very easy to install, but packages are required for additional analyses and the time saved from installation can potentially be lost up in this area.

REPORTING

SAS

- PROs: SAS provides many useful procedures for creating detailed and polished reports.
- CONs: Some of the more detailed reporting procedures, such as TABULATE and REPORT, have a learning curve that takes place before being able to utilize them correctly.

R

- PROs: R includes Sweave which makes it possible to create a PDF containing text, tables, and charts by including LaTeX markup and R commands. Another new package knitr allows the quick creation of web content with less formatting issues
- CONs: R does not have a defined way of producing reports, and to do so would involve a serious programming investment. Reports seem to be a relatively new idea in R so are not yet as easy or quick to implement as in SAS. Sweave and knitr are leaders in this area for R, yet they can prove difficult to learn.

Users whose tasks revolve heavily around creating reports should consider this difference. While some time must be invested to learn SAS reporting procedures, once mastered they prove invaluable and highly flexible. The R

programming that would be required to create reports from the ground up may not outweigh the time investment in getting to know the SAS report procedures.

CONCLUSION

We see the solution to the R vs. SAS debate as three-fold. First, we need to understand as a statistical programming community that there is no clear winner. Both packages have their strengths and weaknesses. They need to co-exist and we in academia also need to teach them as they co-exist. Students will be better served in their degrees if they can explore their options and be able to use their skills as appropriate. By only teaching one methodology we are limiting them and at the very least making it more difficult for them to live up to their potential. Second, users need to keep their technology toolkit up to date. SAS and R have some great websites for learning about new technology advances. The SAS tech support webpage (http://support.sas.com/) has many resources for keeping current such as Focus Areas, E-newsletters, RSS feeds and blogs. The R-blogger website (http://www.r-bloggers.com/) contains news and tutorials contributed by a large number of users. Third, the ideal solution for this problem would be to learn both of these software packages and then to be able to integrate them both into the same analysis. There are various ways to do this such as using SAS IML and SAS IML//Studio (IML is an add-on to the basic SAS install). Or by executing the R code from SAS via the line command mode using the SAS X statement. For R users the system interface from R to SAS can be used to work with both languages. Using both technologies to leverage data manipulation and analysis seems like the winning solution for all.

REFERENCES

Bewerunge, Peter, Dr. "SAS and R - The Odd Couple.", <May 2011. <http://www.phuse.eu> <http://www.phuse.eu/download.aspx?type=cms&docID=2847>

CRAN Contributed Packages: http://cran.r-project.org/web/packages/

Gesmann, Markus. “Interactive reports in R with knitr and RStudio” < http://lamages.blogspot.com/2012/05/interactive-reports-in-r-with-knitr-and.html>

Lex Jansens’s Homepage: http://lexjansen.com

Revolutionary Analytics: http://blog.revolutionanalytics.com/2010/01/r-package-growth.html

Rossiter, D G “Introduction to the R Project for Statistical Computing for use at ITC”, 2012. <http://cran.r-project.org/doc/contrib/Rossiter-RIntro-ITC.pdf >

SAS Online Documentation: Using the SAS Editors under Windows: Using the Enhanced Editor. SAS(R) 9.2 Companion for Windows, Second Edition

SAS Support: http://support.sas.com/

Sascommunity.org: http://www.sascommunity.org/wiki/Main_Page

SAS-L (University of Georgia): http://www.listserv.uga.edu/archives/sas-l.html

Smith, David. “How to create PDF reports with R” < http://blog.revolutionanalytics.com/2010/12/how-to-create-pdf-reports-with-r.html>

RECOMMENDED READING

Statistical Programming with SAS/IML Software, R. Wicklin, SAS Press 2010.

CONTACT INFORMATION

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

Name: Chelsea Loomis Lofland
Address: 557 Sunnymount Ave

City, State ZIP: Sunnyvale, CA 94087
Work Phone: (408)242-2322
E-mail: Chelsea.Loomis.Lofland@gmail.com