Why R; it's not a question, it's an answer.*

As published in Benchmarks RSS Matters, October 2013

`http://web3.unt.edu/benchmarks/issues/2013/10/rss-matters`

Jon Starkweather, PhD

*Warning: The contents of this document are solely the expressed thoughts and opinions of the document author. The contents of this document are not the expressed or implied thoughts or opinions of University of North Texas (UNT) administration, faculty, staff, or students. The contents of this document may contain which may offend or otherwise distress significant proportions of the reading audience by stating rather pointed and unpopular opinions and facts.

Jon Starkweather, PhD
`jonathan.starkweather@unt.edu`
Consultant
Research and **S**tatistical **S**upport

`http://www.unt.edu`

`http://www.unt.edu/rss`

RSS hosts a number of "Short Courses".
A list of them is available at:
`http://www.unt.edu/rss/Instructional.htm`

Those interested in learning more about R, or how to use it, can find information here:
`http://www.unt.edu/rss/class/Jon/R_SC`

Why R; it's not a question, it's an answer.

University of North Texas (UNT), like other large universities in the United States, has been facing a serious budget short fall for quite some time. The United States' (and perhaps the world's) economy has been stagnated for several years and fewer people can afford to attend college or university[1]. UNT is attempting to become a leading research institution. UNT has Four Bold Goals which are meant to guide the institution toward a brighter, more prestigious future (UNT Strategic Plan 2012-2017[2]). Please keep these thoughts in mind as you read below.

Research and Statistical Support (RSS) personnel have been advocating for years that it is well past the time when UNT should discontinue expensive licenses for commercial statistical software packages. This document will further advocate the replacement of commercial software (e.g. SAS, SPSS) with R[3] - and offer some reasons why that is a logical, reasonable, responsible, and indeed necessary choice.

First, of course; R is free – like the air you breathe is free. Anyone, anywhere can download, install, update, customize, and *use* R for absolutely no monetary cost...always, forever, and right now! R requires 2.5 things: (1.0) R requires effort, (1.0) R requires time, and (0.5) if you are using a Microsoft Windows operating system (OS); then you are required to have administrator privileges for that machine in order to install R. However, once installed, you do not need administrator privileges to use R. So, if you can spare a little effort, time, and you have administrator privileges on your Windows computer, then you too can use the most up-to-date, and technically advanced, data analysis software available. Wow! Who knew? Well, RSS knew; and has been screaming as much for years now. But, there was a lot contained in those first few sentences, so let's deconstruct them a bit.

Cost

If you can satisfy the 2.5 requirements (i.e. a little effort, time, & admin privileges if you are using a Windows OS)...then you can download and install R. The effort required to download R is minimal; you simply click on one of the links contained in this[4] web page (yes, it is implied that you are required to have web access, but; if you didn't, then you would not be reading this). Notice on that web page, there are three choices which correspond to the three major OS currently available. If, you are Macintosh (Mac) user, then you clearly are not a SAS user. If you doubt the confidence in that statement, then consider yourself challenged to use the latest version of SAS installed in a Mac OS. For those not inclined to take up the challenge, you might want to know SAS has not been officially available for, or supported on, Mac OS for several years now. Once R is downloaded, it takes only a few clicks to install it. First, click (or double click) on the installation application; then click your way through the installation... you don't even need to type anything. So there! You could, in just a couple of minutes, have a basic (but competent, stable, & reliable) statistical software package. However, if you are interested in doing analysis of complex data structures (and really, who isn't these days?), then you probably want to download and install all the additional packages endorsed by and available from CRAN[5]. As of today (Aug. 16, 2013); there are 4758 packages available on CRAN (keep in mind, there are many more packages available from other repositories). So, we come to the first real time requirement.

[1] If you need a citation for this information...then you should crawl out of the hole under the rock where you have been for the last 4+ years.

[2] `http://www.unt.edu/features/four-bold-goals/`

[3] `http://cran.r-project.org/`

[4] `http://cran.r-project.org/`

[5] `http://cran.r-project.org/web/packages/`

RSS personnel have always strongly advised people to install all the CRAN packages directly after installing the base R software. There are a variety of benefits to this strategy; such as having all the tools you are ever going to need easily on hand. At any rate, downloading and installing all the CRAN packages will consume approximately 4 hours; if done all in one R session. Essentially, once R is installed, you could let it run over night or during a morning while you are in meetings or classes. To accomplish this (downloading and installing all the packages), you simply open R, click on the 'Packages' button in the border task bar, then click on 'Set CRAN mirror...', then select the closest mirror site in terms of geographical location (e.g. "USA (TX 1)") from the list provided, then click on the 'Packages' button again, and select 'Install package(s)...' and select all the packages listed, and finally click the 'OK' button. R will subsequently download and install all the packages. You might be tempted to think approximately 4 hours is a long time.but a complete SAS install – with all the available modules – takes at least 4 hours and often longer. And of course, R costs you absolutely zero money; while SAS and SPSS costs are far from zero, especially if you require all their additional modules.

Take a moment and think about those monetary costs. UNT spends tens of thousands of dollars (if not more) each year to pay for enterprise level software licenses; including SAS and SPSS. If you advocate spending that money to support SAS and SPSS, then you are essentially advocating security for jobs at those corporations (SAS & IBM [SPSS]) over jobs and job security at UNT. Yes, that money could be spent on raises and new hires at UNT rather than *profit*, jobs, and raises at SAS and IBM, which now owns SPSS. Let's not forget that UNT's primary purpose and responsibility is to provide an education to students. If you advocate spending UNT's money to support SAS and SPSS, then you are advocating costs for UNT which are passed down to current and future students in terms of tuition and fee prices – thus decreasing the affordability of the education UNT provides.

Consider a hypothetical faculty member who has been and continues to be a loyal SPSS (or SAS) user...we'll call that faculty member Pat[6]. Pat forces students to use SPSS in the classes Pat instructs and Pat forces students to use SPSS for all the theses and dissertations which Pat supervises. Pat rationalizes the cost issue by pointing out that SPSS (and SAS) is available in its full version on campus (in computer labs) and SPSS (and SAS) provides a 'student version' of their software which costs much, much less than the full version. Some (a term used loosely here) tobacco executives have systematically targeted children with their harmful products because those executives believed getting kids addicted to their products would lead to lifelong, loyal consumption of their products. Pat's behavior demonstrates an extremely similar attitude. One might even be inclined to say, Pat and those like Pat are part of, and perpetuating, *the problem*. Once the students are forced to use SPSS (or SAS), it is extremely likely they will not use another statistical software package later – if you need evidence of this, talk to faculty, students, and researchers like Pat who have used SPSS or SAS for decades. Therefore, faculty like Pat are essentially forcing students to be lifelong SPSS (or SAS) users – and that means, the students will eventually need to pay for the full versions of SPSS (or SAS). Remember, each new *full* version of SPSS and SAS requires *full* payment (i.e. monetary cost) from the user. Every R version released has been absolutely free of monetary cost to the user...and any future versions will be free too.

Time

Many loyal (i.e. addicted) users of SAS and SPSS will readily admit that they do not want to switch from their *trusted* software to R because they claim it will *cost* too much in terms of their time and effort. It is true that most people who *switch* from SAS or SPSS to R find it a time and effort consuming task.

[6] Thank you Saturday Night Live (SNL)

However, when you consider the time and effort expended to learn SAS or SPSS and compare those amounts to the amount typically expended to learn R; you're likely to find those amounts to be nearly equal. In other words, yes; if you are *switching*, it will take some time and effort; but no more than what it took you to learn what you've been using. Dedicated SPSS users are most likely to complain in this respect, but keep in mind it took considerable time to learn the menu system and syntax of SPSS too. Also, in regards to SPSS; many of the sophisticated analyses used for complex data structures are not available in SPSS, let alone the menu system of SPSS (e.g. Structural Equation Modeling). And again, let's not forget the students who are in the unique position of learning a statistical software package for the first time (i.e. they are not *switching* from one software package to another); and therefore, students will be required to expend that same amount of time learning any of the three software packages mentioned.

Effort (i.e. Learning a programming language)

One of the most common (and loud) complaints from SPSS (and often SAS) users when considering a switch to R is that they do not want to learn an entire, or a new, programming language. R is a programming language environment. SAS is a programming language environment. SPSS; although it has a very user-friendly menu system...is a programming language environment. If fact, this complaint is often voiced precisely as: "Who wants to learn an entire (or a new) programming language?" One appropriate response to this question is: most of the developed and developing world! Some of you reading this have probably heard the term BRIC, which stands for or refers to Brazil, Russia, India, and China — the countries or economies which have, or will likely; surpass the United States in terms of world prominence. So it is equally appropriate to think of the answer to the above question as: millions of Brazilian students, millions of Russian Students, Millions of Indian students, and millions of Chinese students; all of whom, by the way, are very likely inclined to use a free software alternative rather than pay large sums of money for software offered from American companies (e.g. SAS & SPSS).

Current and future university level students are more technically computer savvy than ever before. In fact, it is likely that a larger percentage of incoming freshman are already familiar with some sort of programming language (e.g. web page function and design using HTML – after all, the ML stands for Markup Language). Many employers expect new graduates / potential employees to possess at least a rudimentary understanding of computer communications and the use of computer languages. Therefore, advocating the learning of a computer language is not nearly as outlandish or inappropriate for the majority of university students as it might have been 20 or more years ago. Let's face up to the fact that computers are now ubiquitous and virtually anyone able to enter university life has had some exposure to computers in one form or another (e.g. smart phones, lap tops, desktops, etc.).

Furthermore, there are a few things to keep in mind when thinking about the phrase *learning an entire computer language*. First, you might consider the word *entire* as a barrier or discouraging part of the question from above ("...learn an *entire* or a new computer language"). However, you should recognize, like anyone who has learned *any* language, that learning to speak, read, or write in a language does not necessarily mean you must learn the *entire* language. For instance; if you're reading this, then you can read American English...and yet, you probably do not know *all* the words in every (or any) English language dictionary. So, to use, let alone think in, those types of absolute terms (e.g. *entire*) is really quite ridiculous.

One of the variations of the complaint above regarding switching from SAS or SPSS to R is the following: "I don't want to become a computer programmer!" Well, an appropriate response to that complaint is..., you should consider the languages you have already been using; no, not the language of love or the English, Spanish, German...etc. language. Consider the computer language(s) you have been using if you have been conducting data analysis with a computer (i.e. not by hand calculation or an abacus). Regardless of which statistical software package you have used in the past (or continue to use), you are using a computer language to conduct analyses. Perhaps you prefer terms like SAS code, SAS script, MACRO(S), SAS syntax, or even SPSS syntax or SPSS Modules. All of those terms refer to one computer language or another. In SAS, one of the first things people learn is PROC MEANS (a *procedure* for calculating *means* of series of scores). In SPSS, one of the first things people learn is COMPUTE (a procedure for *computing* any number of different numeric expressions; such as total scores, sum together multiple variables, mean, standard deviation, etc.). These are examples of *functions* from the underlying computer languages which are used to perform some simple tasks in the languages' respective environments (i.e. SAS environment & SPSS environment). No doubt many of you conscientious readers are now thinking something like..."so what! I know the SAS code I need," or "so what! I use the menu system in SPSS to do what I need to do!"

Yes, well...oh my, this is a rather unfortunate and unpleasant or uncomfortable situation for us to have reached. You see; both the SAS language and the SPSS language originated way back in the age of the Mainframe computer. *The WHAT!?!?* It may be difficult, but try to imagine a computer which fills the space of a large classroom and is made up of vacuum tubes, beeping lights, a loud humming or ticking noise, a great deal of heat, and occasional sparks; which *requires* punch cards to do analysis...that's a basic description of a computer from 50 (or more) years ago – commonly called a mainframe computer. Both the SAS and SPSS languages originated on, and for, such computers. So, both of those languages are therefore very inflexible. Meaning, as new statistical techniques are devised, they are unlikely to be implemented into SAS or SPSS because of the core language. SAS has done a better job than SPSS at including new statistical analyses over the decades, but even SAS is lagging about 20 years behind R in that regard (and the lag is growing!).

Another aspect to consider when comparing the old mainframe languages of SAS and SPSS to R is the fact that the R language is object oriented. This means most things in the R language are much easier to learn and remember because they are intuitively named and / or constructed. For example, a simple independent samples *t* test involves the following:

```
t.test(Recall1 ~ Candy, alternative = 'two.sided', conf.level = 0.95,
       var.equal = FALSE, data = example1)
```

where t.test is the function, Recall1 is a continuous dependent (or outcome) variable predicted by ( ) the independent (or input) variable Candy (a grouping variable with Skittles & None as the two groups). The alternative = 'two.sided' specifies a two-tailed test of the *t* value, with a specified 95% confidence level, equal variances assumption not assumed (FALSE) and the data frame (data file read into R) called example1 which contains the variables specified in the formula (Recall1 equals or predicted by Candy). This *function* (t.test; like PROC MEANS or COMPUTE mentioned above) returns the output called (or specified) by the user defined *arguments* (i.e. alternative, conf.level, var.equal; similar to the options in SAS procedures or SPSS menu dialogue boxes). Below is the actual output from the function listed above.

```
        Welch Two Sample t-test

data:  Recall1 by Candy
t = -7.7566, df = 48.565, p-value = 4.774e-10
```

```
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -9.933199 -5.844579
sample estimates:
mean in group Skittles     mean in group None
              9.407407              17.296296
```

Software Development

Other important considerations when discussing SAS and SPSS versus R are the considerations of development and / or bug checking or reporting and implementation of bug corrections (e.g. software patches and implementation of new functions in new versions of the software). Large corporations, such as the SAS Institute and IBM (which owns SPSS now) tend to have very slow development tracks which involve many teams of software engineers (i.e. programmers) and many hierarchical management levels of approval (Raymond, 2010). Typically small segmented teams are responsible for development of specific functions within large software companies. These small teams are required to gain the approval of the many levels of organizational hierarchy prior to implementation of new products (i.e. analyses) and prior to implementation of changes (i.e. bug fixes). Large organizational structures such as SAS and IBM often also have user reporting systems for users to report problems with the software (i.e. bugs). These types of reporting systems often leave the user feeling as though their voice of complaint is unlikely to result in implementation of change; either through perceived loss of the complaint, ignoring the complaint, or an impersonal experience with the reporting system. Even if a user complaint gets through the reporting system and is directed to the appropriate team, the turn-around time for implementation is likely to be years due to the complex levels of hierarchical management approval and due to the necessary compatibility with other team's code (i.e. functions being changed by one team must be compatible with all the other teams' functions and compatible with all OS, etc.). As an example of a relatively recent bug; consider the categorical principal components analysis (with optimal scaling) function (of the CATEGORIES module) of SPSS which has a bug documented here[7].

The R development, bug reporting, and bug fixing scheme stands in stark contrast to the cumbersome schemes of SAS and SPSS mentioned above. The core R development team releases a new full version of the software approximately every 6 months (again, for free). SAS and SPSS release new full versions every 12 to 18 months or more (again; for a big monetary price). When speaking of the development of statistical software, it is readily apparent that R is much more receptive to and quicker to implement new methods than SAS or SPSS. Consider a fictional researcher, Dr. Smarty Pants at the University of Jupiter's Moon. Dr. Smarty Pants wants to do a new statistical technique, called the Wiz-Magic Decomposition analysis or WMD for short. Unfortunately, because WMD is so new, Dr. Smarty Pants cannot find WMD in any of the existing statistical software available (e.g. SAS, SPSS, & R). But, Dr. Smarty Pants is an R user. So, no matter who or where Dr. Smarty Pants happens to be, and no matter what analysis Dr. Smarty Pants wants to perform; any individual, like Dr. Smarty Pants, can write the code to perform the desired analysis and send it to CRAN as a new package/library. CRAN will then check it to make sure it works, has proper documentation, and post it so that everyone can then use the newest most advanced techniques, like WMD. You might think this process takes a great deal of time, but it does not.

[7] http://www.unt.edu/rss/class/Jon/SPSS_SC/Module9/M9_CATPCA/SPSS_M9_CATPCA.htm

Also, R provides users with virtually immediate confirmation of bug reporting by allowing R users to be members of the R development community – which means, any R user can post a potential bug report to the R bug tracking system[8], or to the R electronic mailing lists[9]. It is important to realize that because all R users are potentially developers and bug checkers that many, many more eyes are inspecting the actual code's functionality of the software than are inspecting the code of SAS or SPSS (prior to *new* users installing and using the software).

Furthermore, if a user finds a bug in a specific package's code or a specific function, then the user can simply look in the documentation (installed with the packages) to find the contact information of the author of that code or that function (or package) and contact *that person*. The author of this article has done just that and received a reply within two hours. This anecdote highlights one of the major motivating factors for why R bug fixes happen very, very quickly (and those of SAS & SPSS do not). The user has direct access to the contact information of the person or persons who wrote the exact code which the user believes is malfunctioning. There are two key components here: (1) the reporting speed and likelihood of the report getting to the person responsible for the code is near perfect and (2) the identification and reputation of the person who wrote the code is public and therefore, that person has a high degree of motivation to fix the code as quickly as possible and establish or maintain a good professional reputation as a result. It is easy to see that R's development, bug reporting, and bug fixing strategies are much better and quicker than the impersonal, often anonymous and nebulous, strategies of large (i.e. cumbersome) corporations such as SAS and IBM / SPSS.

Another benefit of R's transparency and open source perspective is the breadth and depth of user help or user community support available to new users (i.e. those attempting to learn R for the first time). RSS consultants very often tell clients that if they are reading or reviewing some R help media and they (the client) finds the author of that media distasteful (i.e. the client simply doesn't like the language or presentation style); then the client is instructed to toss aside that media and find another. The rationale for this instruction or advice is that there are so many avenues and media providing help to new users it is guaranteed that another author's language or presentation style is likely to be more easily digestible by that particular client. In fact, one of the strong points of R is the massive amount of free help available to new users (e.g. YouTube[10] videos, search engines[11], list serv(s)[12], new user guides[13], CRAN Task Views[14], websites[15], etc.).

Finally, when considering the choice of using any of the three statistical software packages mentioned in this article; it is important to acknowledge the functionality of each software package (i.e. the range of analyses and the breadth of compatibility with other software). Again, R clearly exceeds SAS and SPSS in terms of breadth of functioning and in terms of efficiency of use or code. As an example, consider (again) that SPSS does not have the ability to do structural equation modeling (SEM). Furthermore, SAS's and SPSS's ability to do many newer or complex analyses (e.g. Bayesian analyses, nonlinear hierarchical linear modeling [HLM]) are severely limited in the menu system, inefficient in syntax (i.e. clunky, or requiring hundreds of lines of code), or simply non-existent (e.g. moderated mediation in SEM or HLM). R can be made to do all of those things. R can also be used to produce publication quality

[8] `https://bugs.r-project.org/bugzilla3/`

[9] `http://www.r-project.org/posting-guide.html`

[10] `http://www.youtube.com/watch?v=mL27TAJGlWc`

[11] `http://www.rseek.org/`

[12] `http://www.r-project.org/mail.html`

[13] `http://cran.r-project.org/other-docs.html`

[14] `http://cran.r-project.org/web/views/`

[15] `http://www.unt.edu/rss/class/Jon/R_SC/`

reports, graphs / figures, web pages (including interactive[16] elements such as graphs), and it can easily interface or import / export information from a variety of databases or sources (e.g. SQL[17], HTML[18], LaTeX[19], etc.).

Conclusions

Do you really need a summarizing paragraph of all the information treated in detail above? Well, given the current world, country, state, and UNT economies; and the goals of UNT to be more efficient, less wasteful, and attain research prominence...why would anyone at UNT involved in a research oriented field advocate using SAS or SPSS when R is available (for free!)? If RSS personnel were open to betting (and they are not); a calculated wager might be offered such as this: we are willing to bet the monetary cost of the best statistical software package available, that most of the most respected researchers in any serious scientific discipline are R users. Feel free to ask those researchers (especially the ones at Ivy League universities ;-). We think you'll find that the researchers most respected are indeed R users. Really, don't take this author's word(s) for it; check for yourself. After all, part of the allure of R and the value of a university education is the ability to *research* a topic, amass evidence, analyze or weigh(t) that evidence's empirical value, and make an informed decision or choice.

Until next time, enjoy *California sunlight, sweet Calcutta rain, Honolulu starbright...*

References & Resources

Raymond, E. S. (2010). *The cathedral & the bazaar.* Sebastopol, CA: O'Reilly & Associates, Inc.

This article was last updated on October 18, 2013.

This document was created using LaTeX

[16] http://www.rstudio.com/shiny/
[17] http://www.r-bloggers.com/make-r-speak-sql-with-sqldf/
[18] http://cran.r-project.org/web/packages/R2HTML/index.html
[19] http://www.youtube.com/watch?v=j8Xk0brZwwk