

**Statistical Consulting Group
UCLA Academic Technology Services
Technical Report Series**

2006 January 30, revised 2007 February 27

Report Number 1, Comment Number 1

R Relative to Statistical Packages:
Comment 1 on Technical Report Number 1 (Version 1.0)
Strategically using General Purpose Statistics Packages:
A Look at Stata, SAS and SPSS

Patrick Burns, Burns Statistics, patrick@burns-stat.com

Stata is a registered trademark of StataCorp.
SAS is a registered trademark of SAS Institute.
SPSS is a registered trademark of SPSS Corporation.
S-PLUS is a registered trademark of Insightful Corporation.

You can contact the author of this comment by sending email to patrick@burns-stat.com .

The recommended citation for this report is.

Burns, P. (2006). *R Relative to Statistical Packages: Comment 1 on Technical Report Number 1 (Version 1.0) Strategically using General Purpose Statistics Packages: A Look at Stata, SAS and SPSS* (Technical Report Series, Report Number 1, Comment 1). Statistical Consulting Group: UCLA Academic Technology Services. Available at <http://www.ats.ucla.edu/stat/technicalreports/>

1 Introduction

The technical report *Strategically using General Purpose Statistics Packages: A Look at Stata, SAS and SPSS* focuses on comparing strengths and weaknesses of SAS, SPSS and Stata. There is a section on R, which some have suspected damns R with faint praise. In particular, R is characterized as hard to learn. Finally there are sections on a number of very specialized pieces of statistical software.

The primary purpose of this comment is to provide an alternative view of the role that R has in the realm of statistical software. I am acting partly as author and partly as editor. A thread on the R-help email list called “A Comment on R” <http://thread.gmane.org/gmane.comp.lang.r.general/53014> is a group response to the technical report—I summarize some of what is said in that thread. Predictably, the R community has a higher opinion of R than that expressed in the technical report. I owe thanks to those who participated in the thread and to those who sent comments privately.

A statement in the technical report that I definitely agree with is that it is good to have a full complement of tools readily available. I find the low priority given to R in the report at odds with this statement. However, there are understandable reasons—given later—for this.

The technical report fails to mention accuracy, which should be a consideration when selecting statistical software. See the *American Statistician* articles by B. D. McCullough “Assessing the Reliability of Statistical Software” Part I and Part II. Some models, such as non-linear regression, are much more likely to have accuracy problems of practical significance than others.

A complement to this document is a Rosetta stone of sorts called “R for SAS and SPSS Users” by Bob Muenchen. This displays code in all three of R, SAS and SPSS for a number of tasks. A related piece called “Comparison of SAS and SPSS Products with R Packages and Functions” compares SAS and SPSS products to functionality that may be found in R.

2 The Big Picture

2.1 Statistical Depth

Though statistics is vast, I’ll simplify it to two extremes. There is statistics in the lesser sense: “I need to find a plausible sounding hypothesis test that gives me a p-value less than 5% so I can publish my work.” If this is as far as you are going, then R is not for you. Your search will be much more efficient in a traditional statistics package.

Alternatively, there is statistics in the large sense: “I want to know what my data have to say.” This, almost by definition, is hard. My sister-in-law says that if she needs statistics for an experiment, then she needs to redo the experiment—her results are unambiguous. (Ernest Rutherford was there before she was, by the way.) If data are of statistical interest, then it means there is some sort of ambiguity (perhaps only from too much data). If your goal is to find what is in your data, then sooner or later R is likely to provide you functionality which can’t be found elsewhere.

2.2 Shaping Thought

Statistical packages tend to have a restrictive point of view. For instance that only one dataset at a time is of interest, and data are always rectangular. Putting data into this Procrustean bed can be dangerous—you may adapt your thinking to your computing, rather than adapt your computing to your situation.

Because of this statistical package straitjacket, spreadsheets are often used to supplement computations. (Even worse, statistical analyses are done in spreadsheets.) While spreadsheets are a wonderful tool, it is not hard to make a spreadsheet that is too complicated for its own good. See “Spreadsheet Addiction” on the Burns Statistics website.

R has an extremely rich range of data structures, and only available computer memory limits the number of datasets that can be involved in a computation. R is a good alternative to both statistical packages and spreadsheets. While R was born in the statistical community, it is useful far beyond statistics.

2.3 Learning Steepness

Since R has a rich set of objects and is a programming language, it is naturally harder to learn than a statistical package. It is especially hard to learn for people who are versed in statistical packages because they have expectations that are wrong.

This “unlearning” problem is probably part of the reason that R received such poor marks in the technical report. But I think there is a more fundamental reason. The report was written by a statistical consultant in a university setting. This is a job where problems come from virtually any academic field, and even within a single field the problems are likely to be diverse. To give good advice for using R to all comers in this environment is a Herculean task. There are hundreds of R packages available, any one of which might be suitable to some client. Only a few people in the world have the breadth of knowledge to really excel at the task. On the other hand, R has been quite successfully used in just such an environment.

Individual users have a much easier task. They need some fundamental knowledge of R, and then they need to learn about a few particular techniques relating to their field. Once competency is achieved, the user can merely keep a lookout for new packages that might help do things better.

In the thread on R-help John Fox writes:

... how hard one finds it to learn something is a function of the intrinsic difficulty of the thing, the person’s previous experience, preferred modes of thinking, etc., and how learning is approached.

2.4 Who Uses and Likes R?

A substantial fraction of statisticians use R, but the vast majority of R users are not statisticians. The list of fields in which R is applied seems endless, and many people report that use of R is substantially increasing in their field.

Below are some, in effect, testimonials for R. Many of these come from the “fortunes” package. I have omitted some of the most extremely pro-R comments.

Felix Grant in *Scientific Computing World* (November 2004):

It’s a huge, awe-inspiring package—easier to perceive as such because the power is not hidden beneath a cosmetic veneer.

Roger Peng (on R-help June 2004):

I don’t think that anyone actually believes that R is designed to make *everyone* happy. For me, R does about 99% of the things that I need to do, but sadly, when I need to order a pizza, I still have to pick up the telephone.

Bill Venables (on R-help January 2004):

The R engine [...] is pretty well uniformly excellent code but you have to take my word for that. Actually, you don’t. The whole engine is open source so, if you wish, you can check every line of it. If people were out to push dodgy software, this is not how they’d go about it.

David Brahm (on R-help January 2006):

Any doubts about R’s big-league status should be put to rest, now that we have a Sudoku Puzzle Solver.

John Maindonald (in the aforementioned R-help thread):

In my view R is such a versatile tool for scientific computing that anyone contemplating a career in science, and who expects to [do] their own computations that have a substantial data analysis component, should learn R.

3 R

R is a dialect of the S language. So the very brief introduction to using it “A Guide for the Unwilling S User” works for both R and S-PLUS. The S language was developed at Bell Labs to create a computing environment for data analysis and graphics. S-PLUS is a commercial version of the language while R is a free, open-source version.

3.1 Use

There are several features of R worth mentioning here, and An Introduction to the S Language contains more of my views on the subject.

3.1.1 R is a Language

R is a language. This is quite important. It means that you don’t have to do repetitive tasks—you can make the computer do it. With the openness of the S language, it means that it is easy to make changes to functions that don’t do quite what you want.

3.1.2 R is Interactive

R is interactive (though batch jobs can be performed). One consequence of this is that you don’t get the “everything but the kitchen sink” output that is common in statistical packages.

3.1.3 Good Graphics

Graphics are very powerful tools for exploring data, and such plots are easy in R. Publication quality graphics are usually not much more work.

Graphics and interactivity are a big part of the power of R. Remember that our goal is to make the data talk. A session might look like: make a couple of standard plots, compute a few statistics, this suggests a new way to plot the data, which might suggest relationships you’d not thought of before. This is an entirely different approach than is (or could be) used with statistical packages. Several people have stated that R forces you to think more. As long as the thinking is about the data rather than the computing (which becomes true with practice), this is a good thing—it’s a more in-depth form of statistics.

3.1.4 R Tames Resampling

Resampling methods (bootstrapping, random permutation tests, and so on) are extremely useful, and they are typically not very imposing with current computing power. R has packages devoted to bootstrapping, but I generally find them unnecessary. Naive bootstraps are almost always good enough for the exploratory work that I tend to do. Resampling is easy in R (just use the `sample` function inside a `for` loop). I know what the sampling scheme is—there is no need to look up documentation. It’s just normal operating procedure. More on this can be found in “The Statistical Bootstrap and Other Resampling Methods”.

The resampling is done using *subscripting*. That is, by selecting some set of observations where the set is described by some other object that you have created. Flexible and natural subscripting conventions are another key feature of R.

Simulation is similar to resampling, and is equally easy in R.

3.1.5 R Plays Well with Others

You can download new R packages over the internet directly in R.

One of the R packages is “fortunes” which has a number of statements, generally made on one of the R mailing lists. (I, for instance, am honored for having a bug in my code.) In R type:

```
install.packages('fortunes')
library(fortunes)
```

to get and use the “fortunes” package (assuming the machine on which you are running R is connected to the internet).

There are convenient connections to many languages and programs. This includes the operating system.

3.1.6 Widely Available

R works on Windows, Macintosh, Linux and Unix platforms. The same R program that you run on your office Unix server will run on your home PC and your Mac laptop. The binary data files generated by one can be read on the others.

3.2 Problems

While being a language is one of R’s greatest strengths, it can make it harder to learn for those with no programming experience. (But those who have worked with data in other programming languages often think they have gone to heaven when switching to R.)

A feature of R that can be problematic is that all objects need to be in memory. Obviously that limits the size of datasets that can be handled. This is the price to be paid for non-rectangular data. Many R users never run into memory problems, while others need to be continuously conscious of the sizes of their objects. R has functions to access databases. The typical solution when working with large datasets is to store the data in a database and bring pieces of the data into R as needed. This is generally effective, but does add complexity.

The advent of 64-bit machines means that memory can be quite large now. In many fields datasets will probably grow slower than the affordable sizes of machines. In a few fields datasets may well grow faster than machines. But these latter datasets may not to be rectangular either.

Another weakness is the lack of coherent documentation that covers the whole of R. There are efforts to improve on the current state, but this is never going to be perfect—managing an avalanche isn’t going to happen.

3.3 Organization

There is what might be termed an “official” R of a few packages that are created by the R Core Team. In addition there are hundreds of packages that have been contributed by many people. Some of these packages represent cutting-edge statistical research. A lot of statistical research is first implemented in R.

All software has bugs. Core R has amazingly few bugs, and bugs that are found often have fixes available for them within a week.

R is not supported in the same fashion as commercial software, but many people have commented that they find the support via the R-help mailing list to be better support than from commercial companies. On the other hand, others find the R-help list quite hostile and think that some commercial support, SAS’s for instance, is much preferable.

The last several releases of R have had internationalization features. For example, there is now a mechanism to easily translate error messages. In addition to its price, this makes it much easier for R to become a global standard.

R has a tremendous amount of momentum, both in terms of user base and functionality. If a decision to go with R looks good at present, it is likely to look much better in the future.

4 Stata and R

Stata has many similarities to R. The main difference is the one-rectangular-dataset restriction in Stata. Even some strongly pro-R people see Stata as a viable choice for use in teaching some classes. The consensus of these people seems to be that while Stata is programmable, programming of any complexity is much better done in R.

The consistency (and speed) of Stata across model fitting functions has been mentioned by several as an asset.

Stata stores data in single precision. However, from what little I've seen it appears that Stata does reasonably well on accuracy tests, so single precision storage doesn't seem to be limiting in that sense. Single precision storage can be an advantage over R with large datasets. Some datasets may fit well on a particular computer if stored in single precision but not fit well with double precision.

5 SAS and R

While SAS is perceived to be very good at data handling, Frank Harrell disagrees. He writes,

I find that R is far ahead of SAS in this respect although most people are shocked to hear me say that. We are doing all our data manipulation (merging, recoding, etc.) in R for pharmaceutical research. The ability to deal with lists of data frames also helps us a great deal when someone sends us a clinical trial database made of 50 SAS datasets.

Big data is where SAS beats R in the eyes of some. Some people continue to use SAS for large estimation problems while using R for everything else.

The experience of some teachers has been that SAS is very hard for students to learn relative to R. This is directly contrary to the impression given in the technical report.

R users seem to view SAS graphics as decidedly inferior.

At least one user has encountered many bugs in SAS, but few in R. This same user found programming much harder in SAS than in R.

R is very suitable for creating reports at scheduled times—daily, weekly or whatever. While this is also possible in SAS, it is not a project to be taken on lightly.

Here is a synopsis of one person's story. He used SAS and, being a fan of open-source, attempted to learn R. He became frustrated with R and gave up. When he had a simple problem that he couldn't do in SAS, he quickly solved it with R. Then over about a month he became comfortable with R from consistent study of it. In hindsight he thinks that the initial problem was that he hadn't changed his way of thinking to match R's approach, and he wanted to master R immediately.

6 SPSS and R

There have been very few comments about SPSS versus R. This probably means that SPSS is perceived to be the most distant from R. SPSS has been characterized as the prototypical “statistical package”—it is inflexible, produces voluminous output from individual rectangular datasets, and has poor programming tools.

Another fortune is:

The documentation level of R is already much higher than average for open source software and even than some commercial packages (esp. SPSS is notorious for its attitude of “You want to do one of these things. If you don't understand what the output means, click help and we'll pop up five lines of mumbo-jumbo that you're not going to understand either.”)

7 Conclusion

R is much more of a complement to one of Stata, SAS or SPSS than any of these three is to another.

Jonathan Baron (personal communication, January 2006):

Another point, which I repeatedly make to students, is that R is free and will continue to exist. Nothing can make it go away. Once you learn it, you are no longer subject to price increases (e.g., from zero, when, as a grad student, you use your advisor's copy of SAS, to several hundred dollars or more after you leave). You can take it with you wherever you go. The investment in learning thus has a long-term payoff.