Reproducible Research: the Bottom Line

Jan de Leeuw
UCLA Statistics Program *

March 11, 2001

From the interesting and provocative paper by Buckheit and Donoho [2] we take the following quotation.

> When we publish articles containing figures which were generated by computer, we also publish the complete software environment which generated the figures.

This principle is quite forcefully and recognizably motivated with problems in current research practice. Buckheit and Donoho have taken their inspiration from the "Green" Stanford geophysicist Jon Claerbout (his views are expounded in more detail in [3]). They formulate what I shall call *Claerbout's Principle*.

> An article about computational science in a scientific publication is *not* the scholarship itself, it is merely *advertising* of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.

These are very commendable quotations, and I agree completely with them, but they do not go far enough.

First, there is no reason to single out figures. The same "Principle" obviously applies to tables, standard errors, and so on. The fact that figures often happen to be easier to reproduce, does not preclude that we should apply the same rule to any form of computer-generated output.

*Written while visiting Universitat Pompeu Fabra, Barcelona, Spain. Comments by David Donoho, Frederic Udina, Michael Greenacre, and Ricard Torres are gratefully acknowledged.

Second, there is no reason to limit the Claerbout's Principle to published articles. We can make exactly the same statement about our lectures and teaching, certainly in the context of graduate teaching. We must be able to give our students our code and our graphics files, so that they can display and study them on their own computers (and not only on our workstations, or in crowded university labs).

And third, and perhaps most importantly, it is not clearly defined what a "software environment" is. Buckheit and Donoho apply the principle in such a way that everybody who wants to check their results is forced to buy MatLab®. Not Mathematica®, Macsyma®, or S-plus®. Those you may need to buy for other articles. This violates the *Freeware Principle*, advocated most vocally by Richard Stallman in the GNU Manifesto [4].

> Arrangements to make people pay for using a program, including licensing of copies, always incur a tremendous cost to society through the cumbersome mechanisms necessary to figure out how much (that is, which programs) a person must pay for. And only a police state can force everyone to obey them. Consider a space station where air must be manufactured at great cost: charging each breather per liter of air may be fair, but wearing the metered gas mask all day and all night is intolerable even if everyone can afford to pay the air bill. And the TV cameras everywhere to see if you ever take the mask off are outrageous. It's better to support the air plant with a head tax and chuck the masks.

Buckheit and Donoho realize this, and admit that using commercial software such as MatLab® does violate the idea of an "ideal" environment.

Commercial software is not only cumbersome, but also as a rule more expensive for students than for faculty, and consequently inherently unfair. It is seemingly true that students pay less, but of course in most cases faculty pay nothing, their departextregisteredent or grant pays for it. Also, it is less expensive for Americans to do science, and it very difficult for people in developing or undeveloping countries. Moreover, commercial software is closed, which means that its properties have to be taken more or less on faith (which can sometimes be quite a leap of faith, compare Richard Fateman's famous review of Mathematica® [1]).

One reason for the enormous success of TEX, except for the fact that the output looks good, is that it is almost infinitely portable, and that TEX files can be send by email all over the world, and come out looking precisely the same in Kazhakstan. Now that journals are taken electronic submissions routinely, the nightextregisteredare of the commercial word processor once again rears its ugly

head. Thousands of files, written in hundreds of commercial wordprocessors, arrive as unreadable ASCII at the editorial offices.

Thus it is unfortunate, although quite understandable, that Buckheit and Donoho emphasize MatLab® and not a freely available clone such as Octave [5], or perhaps even one of the freely available statistical environments such as Xlisp-Stat [6],[7]. Octave now runs on OS-2 and DOS, as well as on most Unix systems, and it can run most MatLab® code. As the UCLA Statistics WWW server [8] shows, Xlisp-Stat can nicely be combined, at least in principle, with HTML to provide dynamic and interactive research papers and textbooks.

For serious research projects in the sciences, we now at least some free and open tools. True, commercial software is often more glossy, and better supported. But the glossiness is irrelevant, and the better support is, to a large extent, a self-fulfilling prophecy. Emacs and TEX are the prime examples showing that non-commercial software can indeed be top-of-the-line.

References

[1] `ftp://peoplesparc.berkeley.edu/pub/papers/mma.review.ps.Z`

[2] `ftp://playfair.stanford.edu/pub/buckheit/wavelab.ps.Z`

[3] `http://sepwww.stanford.edu/`

[4] `ftp://prep.ai.mit.edu/pub/gnu/GNUinfo/GNU`

[5] `ftp://ftp.che.utexas.edu/pub/octave`

[6] `ftp://umnstat.stat.umn.edu/pub/xlispstat`

[7] `http://www.stat.ucla.edu/develop/lisp/xlisp/xlisp-stat/`

[8] `http://www.stat.ucla.edu/practice/`