

SOFTWARE FOR STATISTICS FOR PHYSICS

JAMES T. LINNEMANN

*Michigan State University, Department of Physics & Astronomy,
E. Lansing, MI 48824, USA
E-mail: linnemann@pa.msu.edu*

I discuss two workshops held in 2004 and 2005 relevant to the software environment for statistical analysis in physics and astrophysics. The first largely explored the R environment used by statisticians and the Root environment widely used in particle physics and related fields. The second was a step towards starting a repository for software useful in statistical analyses for these fields. I also discuss some of the statistical software resources on the web of relevance to physicists.

1. Introduction

The work behind this talk grew out of the PHYSTAT2003 conference, where Louis Lyons invited proposals for focused PHYSTAT workshops. After my interactions with statisticians at PHYSTAT2003, the first thing I wanted was to improve the environment for doing particle and cosmic ray physics analysis using statistical tools. I particularly wanted to enhance what I already had available in Root¹, which is our everyday working environment. I clearly suffered from R envy. The R language and environment² is heavily used by statisticians. The second thing I missed was a web page for physicists of pointers to implementations of statistical methods. And the third thing I felt we lacked was a web site to collect new statistical software oriented to the needs of physics. The first and the third of these were subjects of workshops, and the second I started work on myself. I'll discuss each in turn.

I must apologize if I didn't invite you personally to these workshops. Each workshop had a limited goal: to try to do *work* in a day or two (or at least to start). To me, the right way to do that is to get at least some of the right people in a room. Thus, the workshops were designed with small attendance, to concentrate on discussion rather than hearing a parade of presentations. I'll leave you to judge how successful we were.

2. R and Root

In 2004 I organized a PHYSTAT workshop³ at MSU on statistical software, concentrating mainly, because of those able to attend, on Root and R. Two developers of major software systems attended: Luc Tierney of the R core development team (thanks to

valuable contacts by Jerry Friedman, Nancy Reid), and Rene Brun, lead developer of the Root system. Astronomers also attended: Eric Feigelson, who developed the StatCodes⁴ web site and who is working on the virtual observatory statistics project VOSTat⁵, and Tim Beers who developed the Rostat⁶ robust statistics package. Physicist/developers included Harrison Prosper, Scott Snyder, Sherry Towers (TerraFerMa⁷), and three physicist R users from Fermilab: Adam Lyon, Jim Kowalkowski, and Marc Paterno.

For those not familiar with Root, I would describe its key features as follows. It provides a GUI for publication-quality graphics and for making the cuts (data sub-region selections) we physicists are so fond of. It also provides I/O which scales to petabytes data sets consisting of collections of files containing event data (with each event individually tree-structured). Root uses a histogram as its base metaphor. Its primary interface is a command prompt, which accepts C++ as a language for interpreted and compiled macros. Root is extensible, though most might not say "easily." Root contains sophisticated nonlinear fitting and reporting of multidimensional parameter errors. Its collection of statistical algorithms is small, but growing. For example, robust (to outliers) curve fitting was recently added. Anna Kreshuk's talk at this conference gives more information on recent developments in Root.

For those not familiar with R, it is an elegant data manipulation language (R is a gnu implementation of the S language⁸), embedded in an environment rich in statistical functionality. The user sees a command prompt. Macros in R are interpreted, but heading toward byte-compilation. R is not GUI-

oriented, though hooks are being built: most users are satisfied with the command line. However, S+, a commercial⁹ implementation of the S language, does provide a rich GUI interface. Most S or S+ code runs happily in the R environment.

R is described by statisticians as a quick and easy interactive analysis tool, and is indeed the standard tool of professional research statisticians. So if a statistician suggests a method to you (for example bootstrapping, the lasso, bagging, boosting, cross-validation etc.), its probably implemented in R. The R environment has as built in functions a large range of sophisticated statistical tests and graphics, many of which are not in common physics usage.

R has links to further multidimensional graphics (Ggobi), and a broad package library¹⁰, with trivial download mechanism. R allows straightforward extensibility to new packages in R or C code. Functions and packages are often very fast if they are R-wrapped C code. R keeps data in virtual memory Data Frames, and uses vectors as its basic metaphor. R has interfaces to postgres, mysql, and other databases, and has parallel computation under development. While both Root and R are used outside their home communities, R and S documentation^{2, 8} is commercially published and available at Amazon.

Susan Holmes' talk at this conference discusses data visualization largely using R tools, and Marc Paterno's talk provides further detail on R use from a physicist's perspective. Also useful is Adam Lyon's talk¹¹ at the MSU workshop.

There were three main results of the workshop. Eric Feigelson was confirmed in his initial inclination to use R for the basis of the VOSTat project. Adam Lyon, encouraged by discussions with Luc Tierney, wrote a fairly general Root Tree reader for R. Rene Brun was perhaps further interested in R, encouraged on his existing path of adding statistical functionality to Root, and, I hope, inspired by R's elegant package mechanism¹⁰. Rene and I at this conference celebrated (?) a quarter century of my encouraging Rene to do even better than he has in providing an everyday environment for particle physicists.

3. Statistical Resources on the Web for Physicists

My second topic grew out of preparation for the software workshop just described. I wanted to survey what statistical resources were available on the web for physicists. Having a few lazy bones in my body, I wanted to know where I might find useful statistical software without having to write it all from scratch. In the process I developed a page of links at http://www.pa.msu.edu/people/linnemann/stat_resources.html.

I definitely don't want to claim there had been no effort in particle physics before mine. But to my shock, this is now the largest such page I know of. Others who had preceded me in HEP included Glen Cowan, and the CDF statistics committee. But the reason for the lack of pointer pages is, I believe, the lack of actual web statistics-oriented resources specific to physics.

Here again I suffer envy of other fields. In particular, astrophysicist Eric Feigelson has done an excellent job of surveying statistical resources at his StatCodes site⁴. Its point of view is quite general in fact – physicists should most certainly look there – though of course he is particularly interested in items relevant to astronomy and astrophysics, a few of which have found less application in particle physics. Tom Loredo¹² also has a very useful collection of links. Not surprisingly, there are many useful sites from statistics, particularly StatLib¹³. There are also quite a number of useful resources on multidimensional analysis which I included on my page. I'm sure many of you have your own favorite links to software, and I would be delighted for you to send them to me. I have avoided most references to commercial software, mainly because I have seldom seen my physicist colleagues use (i.e. pay for) commercial analysis software. Astrophysicists, however, find their productivity gains well worth the cost of the commercial IDL¹⁴ package for analysis, interpolation and manipulation of 2 and 3D image data; it contains substantial statistical functionality as well.

4. Towards a Repository for Statistical Software for Physicists

One conclusion I drew after searching for physics-oriented statistical resources was that I was also suffering from a serious case of repository envy. As-

tronomy has a number of user-contributed repositories under way for analysis and statistical codes, for example those maintained by the Astronomical Software Directory Service¹⁵ and NASA's HEASARC¹⁶. Even biology has *bioconductor*¹⁷, a large collection of R software for bioinformatics. There are a few HEP repositories¹⁸, but there is little physics-oriented analysis or statistical software on the web at present. In some ways this is surprising, as the web was invented for HEP. Assessing user interest in such a repository was the motivation for a 2005 workshop¹⁹ Mark Fishler and I organized at Fermilab.

Behind the archive is Mark Twain's notion that if you make it sufficiently attractive for someone to write statistical code, they might actually do it for you²⁰. Louis Lyons advised me that in giving this talk, I was coming to the right place to find software writers. And when I asked who in the audience had written statistical software of use to someone else, a goodly majority indeed raised their hands. I know I would find it useful to have access to the programs used to produce results for many of the talks at this conference.

The basic motivation for a software repository is sharing: don't reinvent the wheel; improve it. A repository requires some implied longevity which seems best met by having an organization rather than an individual as sponsor. Fermilab is potentially interested in such a role. Clearly a web interface is needed for upload, search, retrieval. One can envision a hierarchy of purposes, ranging from an archive for source code of software associated with physics or conference papers, through a downloadable package library (either of stand-alone packages, or packages adapted to particular frameworks), to a component library with various language or web interfaces, possibly with distribution of binaries for various platforms.

A *Statistical Software Archive.* The simplest repository function discussed at the workshop was an open archive (roughly analogous to arxiv.org). If you publish a statistical calculation in refereed physics papers or at statistical conferences such as this one, you could put the code in the archive, and reference it. With an archival repository available, one could hope that this becomes as much a part of the culture as submitting preprints to arxiv.org has become. Archiving offers the potential of substantial benefit for a modest effort.

The "guarantee" for users would be intentionally weak: once, the code compiled and ran on some machine and produced useful results. To allow reuse of code with credit to authors, the minimal information supplied would be the author, title, and a one-line explanation of purpose. Keywords and possibly the experiment to which it was relevant would make it easier to locate. Your grad student could start a project here, rather than from scratch, and possibly compare methods used by different experiments. Documentation would be encouraged (but not quite required). Version tracking would need to be supported by the system even at this basic level.

There are many candidates for software in such an archive: calculations of significance, limit setting programs, and goodness of fit tests come to mind, as does software for studying the behavior of statistical methods. In these areas, competing procedures exist: some are published, some not. Actual programs are very hard to find: you have to know of the method, and ask its author personally; at best, you might find some such code in your physics collaboration's CVS repository. Only a few such programs have public web interfaces (D0 or Babar have some).

A *Package Download Site.* A more sophisticated use of a repository would be software written explicitly for re-use (rather than archived for the historical record). Packages of this kind might be stand-alone programs, or packages for frameworks such as R or Root. Here there is a real need for well-designed conventions to support portability and simplify building and upload. Documentation now also becomes a vital issue, including of course any published references for the methods used. In this context, R's package mechanism is particularly admirable. Attaining the same level of simplicity for user and author for Root add-ons would be a real achievement. A repository sponsor can add real value by providing proper repository design to help authors reach users simply and effectively. Further value could be added by choosing packages (possibly even those originally submitted only for archiving) which are of sufficient interest to maintain for reuse at this level, and by providing assistance to authors on issues of portability, numerical techniques, base library choice, or other coding practices.

Candidates for packages of this type also spring readily to mind: multidimensional analysis packages such as Sherry Towers' TerraFerMa⁷ and Ilya

Narsky's StatPatternRecognition (described at this conference). Both are currently stand-alone programs rather than framework packages.

A *Linkable Toolkit*. An even higher level of functionality is also conceivable, by working from the basis of such a repository. One could imagine providing a toolkit library. One might aim to support writing "toy Monte Carlo" or ensemble test writing by providing a coherent library, perhaps building on the gsl (Gnu Scientific Library), or on the *mathcore* or *mathmore* libraries envisioned within Root. This would involve interacting with users to assess existing tools and designing and supplying missing ones. One might repackage existing programs as framework packages to enhance usability. Quality control (robustness, accuracy, etc) become critical at this level of ambition. Personnel for design, coding, and maintenance would be required. One might also provide a computation service for some of the simpler algorithms, with web data entry via forms or ASCII data files.

Plans. The main outcome of this workshop was a consensus statement (included here as an Appendix). Based on this consensus, we had further negotiations with the Fermilab computing division and carried forward discussion to this conference. Marc Paterno of Fermilab attended this conference in large part to assess the interest of a wider community in the repository. At this PHYSTAT2005 conference, Marc Paterno, Louis Lyons, and I have discussed these ideas with many people. These consultations uncovered potentially interesting synergies with Root and the Cedar physics archive in the UK. We also found that considerable interest was expressed for the archive, and were strongly encouraged to get started and see how things evolve once contributions begin arriving. Thus, we are in the process of preparing a proposal to the Fermilab Computing Division to support some version of an archive. Let us know if you think its worthwhile, and pass along any advice you might have. We are currently thinking about the level of manpower required to get started, and working through computer security, copyright, and license issues associated with such a venture.

In the end, it is hardly our vision of the repository and how it might be used that matter. What counts is how the community chooses to use it, and our main motivation is to provide a forum for unleashing the creativity of that community. Whether

it houses mathematica, mathcad, or matlab software for producing statistical figures for conference papers; C++ or Fortran routines used in Physical Review or Physics Letters or NIM articles; mass fitters, deconvolvers, goodness of fit or significance or limit calculators, or cunning ways of telling signal from background; whether submissions are programs or packages for R or Root; whether written in java, C, C++, perl, ruby, or python; or entirely other things, depends on what the community finds most useful.

5. The Reproducible Research Ideal

Reproducible Research²¹ is an interesting concept in some ways related to the repository. The ideal is that when you write a paper, you save (in a tar archive, say) the entire environment necessary for creating the paper through scripts, and the whole paper and its figures and tables are generated by executing a single high-level script. This tar archive would of course be an excellent submission to the software repository we have discussed.

We all know the kind of problems that led to these thoughts: you ask a graduate student to pick up a project and suggest one of your papers as a starting point, but the student finds it remarkably difficult to actually reproduce the plot you suggested. To do so requires having the same data set you used several years ago, and to use the programs with all the same settings. To achieve a reasonable approximation to this ideal requires as a minimum a powerful script-oriented method of producing figures and tables (such as R or Root) and all the data used in the paper. It also implicitly implies a data set of rather modest size, and a stable set of tools. Otherwise you'd have to save the entire contents of your computer each time.

More is required, however: directory conventions, makefiles, and many other details should be conventional and stable. Arxiv.org provides a subset of such an environment: you know that you will be able to rebuild a pdf file from the latex source and eps files if you meet arxiv's requirements. This ideal is achievable for most (not all) plots shown at this conference, and for most significance and limit calculations in our physics papers. It is problematic for large HEP data sets, which are not publicly available and not necessarily permanently archived with full version control. It is also problematic for analy-

ses which are long in duration (months to years, not hours to days). This is exacerbated when multiple analyses are combined into a single publication, as is often the case in large physics collaborations.

Still, the reproducible research ideal is well worth striving toward. Those who have created a research environment fully supporting the ideal describe it as a discipline with more benefits to authors than to readers wishing to build on the published research.

6. Conclusions

To summarize, I'd like you to take away three main points. First, R has many intrinsic attractions, and is a window to the statistics community. It should be better known in physics and astrophysics, and it is now possible to read Root trees in R. I would personally be delighted if everything in R appeared in Root, my everyday environment. Second, I started a page of web links to statistical software resources relevant to physicists and astrophysicists. If you find it useful, tell your colleagues, link to it, and more importantly, help me improve it. Third, we are trying to start a repository for statistics-oriented software of use to physicists and astrophysicists. I'd appreciate your discussing this repository within your collaboration, and encourage us (and the Fermilab Computing Division) if you think it should be pursued. And we hope you will also contribute software to the repository.

Appendix: Consensus Statement from the 2005 Fermilab Workshop

Following is a slightly abbreviated version of the consensus statement resulting from the workshop:

Currently, statistical tools are in use by individual physicists, and within collaborations. Their ultimate purpose is to make the best use of the data collected by collaborations. However, their effectiveness is limited by the lack of a straightforward mechanism for the community to share software on a wider basis, learn best practice from one another, and avoid unnecessary re-development of similar tools. Some tools are of general use (for example event classifiers, or limit calculation programs). These codes often embody standard practices within a collaboration, recent progress of understanding within our field, or implementation of important ideas developed by statisticians or within the machine learn-

ing communities. Other programs encode hard-won expertise in handling particular situations. Sharing such codes across research groups and collaborations contributes directly to the diffusion of such knowledge, and indirectly to improvement of our understanding of our data and the training of students by facilitating comparison of methods. A repository could provide, as objects of study and understanding, working codes which have been tested under realistic conditions. Such codes would also provide a point of departure for improvements, rather than having to first re-implement present ideas for lack of publicly-accessible code.

What sort of repository would support such efforts? We suggest a phased approach. The first and perhaps most important step would be a very open archival repository, where essentially anyone could upload code felt to be useful for statistical tasks in physics experiments. The repository should make it straightforward to store software used to perform calculations for a paper, and refer to those calculations in the publications: "we calculated the upper limit using a Bayesian technique assuming a flat prior in the cross section [17]", and reference [17] might read "C. Calvin & H. Hobbes, www.phystat.org/05/07/23/0013/, version 3". The repository would provide some basic expectations on what a submitting author should provide, but the absolute requirements would be purposely minimal, in order to encourage submission.

A submission should minimally include authors, an email contact address, a tar archive with code and a brief text description of what the submission does. There would be a possibility to provide keywords and an experiment of origin, but not a requirement. A read-me file would be encouraged to include documentation and the platform(s) on which the code had run. Overall, the effort required for submission should be less than or comparable to submitting a paper to arxiv.org.

Downloading code from the archive should be similarly straightforward. Search facilities from the web might start with a simple web listing of entries with a one line description, but could become more sophisticated as more entries became available. Attaching user feedback is another possible evolution path.

Fermilab would be a natural sponsor of such a repository, assuming that it could provide the desired

degree of openness. The lab hosts experiments which are currently producing much innovative statistical software, and the lab intends to be a center for ongoing research in particle and astro-particle physics. This is an important activity supporting data analysis, which does not require proximity to the physical location of the experiment. And there are members of the computing division with professional interests in this area.

A longer term vision of the repository goes beyond passively archiving code. One value-added activity would be to classify the submissions to distinguish archival entries from actively maintained packages. Capture of user assessment of such packages might be particularly useful. Packages could also benefit from expertise by improving the efficiency or portability of the submitted code. Design expertise might provide standards for packages which would make them more readily usable. A particular example of interest is the elegant R package mechanism: it would be a real achievement to have design standards which would allow a similar ease of package creation and import within the Root framework. Standards might include naming conventions, package directory structure, allowed base libraries, or build tools. Other activities might include mining the submissions for likely contributions to a linkable library (for example mathmore packages), identifying and writing code for missing functionality, integrating related packages, soliciting and supporting extensions of existing code (justifiable by a broader use base than a single experiment), or actively looking for interesting software produced by the statistical software community and providing web interfaces or language translation wrappers to support use by the physics community. Another possibility is maintenance of a list of such software, perhaps building on the software link web site developed by Jim Linne-
mann. Such value-added activities would best evolve over time as the use of the repository grows.

We intend to submit soon a more formal request to Fermilab management, and to approach large collaborations to solicit their support for such an endeavor.

Acknowledgments

Thanks to Tom Loredo for many astronomy links, and reminding me of Reproducible Research, and to

Bob Nichol for useful comments on this manuscript.

A word on references

I have omitted the initial `http://` in all the web references. Many more links are available at: http://www.pa.msu.edu/people/linnemann/stat_resources.html.

References

1. root.cern.ch. This site contains links to source code, online documentation and tutorials.
2. www.r-project.org; Venables and Smith, An Introduction to R, Network Theory Limited (2001); Dalgaard, Introductory Statistics with R, Springer (2002); Everitt, An R and S-Plus Companion to Multivariate Analysis, Springer (2005); zoonek2.free.fr/UNIX/48_R/all.html (R tutorial).
3. user.pa.msu.edu/linnemann/public/workshop
4. astrostatistics.psu.edu/statcodes
5. astrostatistics.psu.edu/vostat
6. Beers, T.C., Flynn, K., Gebhardt, K., "Measures of Location and Scale in Clusters of Galaxies. I. A Robust Approach," 1990, *Astronomical Journal*, 100, 32; see also Hoaglin, Mosteller, Tukey, *Understanding Robust and Exploratory Data Analysis*, Wiley(2000). There is no Rostat web site.
7. [www-d0.fnal.gov/\\$\sim\\$smjt/multiv.html](http://www-d0.fnal.gov/\simsmjt/multiv.html)
8. Becker, Chambers, and Wilks, *The New S Language*, Chapman and Hall (1988); Chambers and Hastie, *Statistical Models in S*, Chapman and Hall (1992); Venables and Ripley, *S Programming*, Spring (2000); Chambers, *Programming with Data: A Guide to the S Language*, Springer (2004).
9. www.insightful.com
10. cran.us.r-project.org
11. user.pa.msu.edu/linnemann/public/workshop/rInHep.ppt
12. www.astro.cornell.edu/staff/loredo/statpy
13. lib.stat.cmu.edu
14. www.rsinc.com/idl/
15. asds.stsci.edu/packages.html
16. heasarc.gsfc.nasa.gov/docs/software.html
17. www.bioconductor.org
18. ph-sft.web.cern.ch/ph-sft, www.freehep.org, cepa.fnal.gov/CPD, www.cedar.ac.uk, www2.slac.stanford.edu/computing/top_pages/software.htm
19. whcdf03.fnal.gov/PHYSTATworkshop, user.pa.msu.edu/linnemann/public/workshop/Fermi_Program.htm
20. Mark Twain, *Adventures of Tom Sawyer* (1876); see how Tom handles the chore of painting the fence around his house.
21. www.stat.washington.edu/jaw/jaw.research.reproducible.html