Using R on Debian: Past, Present, and Future

Douglas Bates
bates@stat.wisc.edu

Dirk Eddelbuettel
edd@debian.org

Albrecht Gebhardt
albrecht.gebhardt@uni-klu.ac.at

Submitted to useR! 2004

Abstract

In this paper, we discuss the R language and environment, and in particular its use on Debian Linux, as well as the Debian package management system.

1 Introduction

More and more research, both applied and theoretical, in physical, biological, and social sciences is being performed with computers. Researchers prefer to concentrate on their research, rather than on the provision of a suitable computing infrastructure. To this end, a 'nuts-to-bolts' distribution of software can help by providing high quality and up-to-date binary packages ready for immediate use. In this paper, we describe one such system: the R environment and language for 'programming with data', and in particular its availability for the Debian Linux distribution.

This paper is organized as follows. We first introduce the R environment and language, before examine its usage on a Debian system. The next section discusses creation of CRAN-based Debian packages in detail, before we conclude with a short review and an outlook.

2 About R

R (R Development Core Team, 2004) provides a powerful computing environment for doing statistical analysis, computation and graphics. It provides tools ranging from simple exploratory methods to complex modelling and visualisation functions. These tools can be used at the command-line, as well as via some graphical user interfaces. R is one implementation of the well established S language. One important feature of R is its extensibility. Users can write their own functions either in the S language, or (mostly for performance reasons, or to link with external libraries) code written in C, C++ or Fortran can be compiled and linked to R in a straightforward manner. Both R and compiled code can be made available in the form of so-called R packages. A large (over 320 as of February 2004) and fast-growing collection of contributed packages is available through the Comprehensive R Archive Network, or CRAN for short, which provides a series of mirrored file repositories around the globe.

R has been available for the Debian GNU/Linux distribution since the 0.61 release in 1997. Based on the original source code, several binary packages are provided which allows a Debian user to install all components of the R system, or just a subset. While that is similar to a Windows user opting to install only parts of the initial download, it is in fact based on separate binary packages.

Beyond the core R packages, additional utilties such as ESS (the 'Emacs Speaks Statistics' mode for a particularly powerful type of text editors) and the Xgobi (and now Ggobi) data visualization frontends have been made available, as have been other programs such as the postgresql-plr package which provides R as a procedural language embedded in the PostgreSQL relational database management system.

Given the large, and growing, extent of packages of CRAN, it was only a matter of time before individual packages would be integrated into Debian. This has happened over the course of the last year, see the Appendix for a current list of packages.

The contribution of this paper is to outline the future direction of an integration of CRAN packages into Debian – either directly within Debian, or via an archive suitable for `apt-get` hostes on the CRAN mirrors. The setup describe here should also be suitable for an use with other code repositories (built on top of R) such as the BioConductor project.

3 So even if we use R, why with Debian?

Different users of Debian would probably give different answers to this question. However, non-users of Debian are often converging on a single answer: the (real or perceived) difficulty of the Debian installation process. The next section discusses the interplay between an easier initial installation versus an easier long-term maintenance and upgrade path. Ideally, a computing platform such as a Linux installation should excel in both aspects.

In the context of R and Debian, it may also be worthwhile to point out that a significant portion of the infrastructure of the R Project, including the main CRAN host, is running n Debian systems.

3.1 Installation versus longer-term administration

It has been said that a large number of Linux users get their first experiences by installing a distribution with a strong focus of ease-of-use during installation such as SuSE, Mandrake, RedHat or others. Some of these users may experience, over the course of a few years and after some upgrade/reinstallation cycles, that maintaining a system can be tedious, and even prone to failures requiring a full reinstallations. It is in this area that Debian is very clearly recognised for its ease of maintenance and upgradeability of a Debian GNU/Linux system.

Other advantages of Debian are the large collection of available packages – as of February 2004, about 8150 source packages with 13900 binary packages are reported by `http://www.debian.gr.jp/~kitame/maint.cgi` based on packages in the development branch of Debian. A further advantage is the robust and powerful mechanism for determining package inter-dependencies, an aspect that can become troubling on other types of systems (see also the next section) yet which has worked extremely well for Debian leading to a reputation for reliability. Debian is also a distribution supporting a wider variety of different hardware architectures: currently, ten different platforms ranging from x86 to S/390 are supported.

3.2 Why provide Debian packages of R packages?

One reason for providing a Debian package of an R package is to use Debian package dependencies to ensure that any system libraries or include files required to compile the code in the R package are available. For example, the Debian postgresql-dev package must be installed if the R package Rpgsql is to be installed successfully. By providing the meta-information of required packages in a control file, the build process can be largely automated.

The second reason is for ease of maintenance, as we first mentioned above. Someone who already uses Debian tools such as `apt-get` to update the packages on a Debian system may find installing or updating a Debian package to be more convenient than installing the r-base Debian package plus learning to update R packages from within R or externally using R CMD INSTALL. Because R is beginning to be used more widely in fields such as in biology (e.g. Bioconductor) and social sciences, we should not count on the typical user being an R guru. Having R packages controlled by `apt-get` seems worth the small amount of overhead in creating the Debian packages. This also applies to systems maintained by (presumably non-R using) system administrators who may already be more familiar with Debian's package mechanism. By using this system to distribute CRAN packages, another learning curve is avoided for those who may not actually use R but simply provide it for others.

The third reason is quality control. The CRAN team already goes to great length, including un-supervised nightly builds, to ensure the individual quality and coherence of an R package. Embedding a binary R package in the Debian package management system provides additional control over dependencies between required components or libraries, as well as access to a fully automated system of 'build daemons' which can recompile a source package for up to ten other architectures – which provides a good portability and quality control test.

The fourth reason is scalability. More and more users are using several machines, or may need to share work with co-workers. Being able to create, distribute and install identical binary packages makes it easier to keep machines synchronised in order to provide similar environments.

The fifth reason plays on Debian's strength as a common platform for other 'derived' systems. Examples for Debian derivatives include entire distributions such as Lindows or Libranet, as well as Knoppix and its own derivatives such as Quantian. Providing Debian packages of R packages allows others to use these in entirely new environments.

4 Debianising CRAN: How ?

As noted above, R itself has been a part of Debian since 1997. Binary Debian packages of contributed R libraries have been created since March 2003, starting with RODBC and tseries. Currrently, several Debian maintainers working more-or-less individually provide a variety of CRAN packages totalling about thirty-five packages (see the Appendix for a list). A proposed Debian R Policy (Bates and Eddelbuettel, 2003) is aiding in keeping these package in a coherent and homogeneous form.

Also, most Windows users rely on the R packages (currently maintained by Uwe Ligges) for their operating system distributed at CRAN. SuSE Linux users can grab their packages from CRAN. The increasing work load with packaging especially these SuSE Linux packages lead one of us (A. Gebhardt) some years ago to switching to a more automated process. This resulted in a Perl script[1] which does the whole job of building the RPM files. This script is still in use by Detlef Steuer who is currently in charge of being the package builder for contributed package for the SuSE Linux distribution.

More recently, this script has been modified[2] to perform a similar task for the Debian community. It performs the following steps:

1. Retrieve an up-to-date package list (`ftp://cran.r-project.org/pub/R/contrib/main/PACKAGES`).
2. Retrieve the library description files (`ftp://cran.r-project.org/pub/R/contrib/main/Descriptions/*.DESCRIPTION`).
3. Parse these description files using the `R::Dcf` Perl module.
4. Determine a package dependency graph based on the "Depends:" statements found in the description files.
5. Generate a Debian package build structure (according to the Debian R package policy).
6. Finally build and install the Debian packages in correct order (bottom to top in the dependency graph).

This process relies much on the format of the DESCRIPTION files of the individual libraries. It can cope automatically with inter-library dependencies. However, it can not yet deal with dependencies on non-R software, e.g. external programs like xgobi or GRASS. In these cases, some help is required from a hard-coded dependency list that can be kept in a small 'knowledge base', either as simple flat-file or a small RDBMS. The build process involves an extra run of "`R CMD check`" and insures that way that only correctly working packages will be distributed.

[1] Available at `http://cran.r-project.org/bin/linux/suse/build-R-contrib-rpms.pl`.
[2] The new version is available at `http://www.math.uni-klu.ac.at/~agebhard/build-R-contrib-debs.pl`.

It is our intention to extend the meta-information in such a way that we should be able to provide unattended, automatic creation of Debian packages for CRAN. This could then be provided as a side-effect of the exiting quality control builds at CRAN where all packages are already (re-)build every night in order to ensure and control code and packaging quality.

5 Experiences and Outlook

The 'Debianisation' of R may be perceived as being overly confusing to new users due to the schere number of components – just as the whole of Debian may be with its 13,000 packages. While this may be true in the narrow sense, we feel that the added flexibility of being able to customize and adapt an installation is worth the marginal difficulty it may add. We would also argue to few truly novice users ever install an entire system from scratch. Just as it is said that 'nobody is installing Windows' (given the prevalence of purchasing computers with an operating system pre-installed), few new users will attempt to go from a blank hard disk to a working system. On the other hand, we feel that more experienced users do in fact value the added flexibility.

For more than six years, the core R packages have been a part of Debian. The experiences from that process are very positive and encouraging. As R grows, and more and more user-contributed packages are added to the CRAN repositories, it becomes desirable to provide a similar level of quality and robustness for the contributed packages as there is for the core parts of R.

The advantage of prebuilt base and contrib R debian packages is clearly the ease of installation, maintenance and upgradeability. It will not only reduce adminstrative efforts in central networked installations, but also simplify the process of tailoring specific Knoppix based CD images which e.g. can be handed out to students. Quantian[3] is one example of a Knoppix-derived 'live cdrom' that makes use of both R and the existing Debian R and CRAN packages. Another one is the Debian based campus wide GNU/Linux installation at Klagenfurt University. A subset of this installation is also available as CD image[4] containing first versions of the above mentioned `r-cran-*` Debian packages.

Providing R for Debian has been a rewarding experience. Anecdotal evidence suggests that Debian is used for R work in variety of educational, industry and government institutions all of which should benefit from having the very rich set of CRAN packages only one command away.

References

Douglas Bates and Dirk Eddelbuettel. Debian R Policy: Draft proposal, 2003. URL `http://lists.debian.org/debian-devel-0312/msg02332.html`.

BioConductor. BioConductor: Open Source Software for BioInformatics, 2004. URL `http://www.bioconductor.org`.

CRAN. CRAN: The Comprehensive R Archive Network, 2004. URL `http://CRAN.R-project.org`.

Debian. Debian Project, 2004. URL `http://www.debian.org`.

R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2004. URL `http://www.R-project.org`. ISBN 3-900051-00-3.

[3] `http://dirk.eddelbuettel.com/quantian.html`

[4] `ftp://ftp.uni-klu.ac.at/pub/unikluKNOPPIX.iso`

A Existing packages

```
edd@homebud:~> apt-cache search "^r-(.*cran|omegahat)-" | sort
r-cran-abind - GNU R abind multi-dimensional array combination function
r-cran-boot - GNU R package for bootstrapping functions from Davison and Hinkley
r-cran-car - GNU R Companion to Applied Regression by John Fox
r-cran-cluster - GNU R package for cluster analysis by Rousseeuw et al
r-cran-coda - Output analysis and diagnostics for MCMC simulations in R
r-cran-dbi - database interface for R
r-cran-design - GNU R regression modeling strategies tools by Frank Harrell
r-cran-effects - GNU R graphical and tabular effects display for glm models
r-cran-foreign - GNU R package to read / write data from other statistical systems
r-cran-gtkdevice - GNU R Gtk device driver package
r-cran-hmisc - GNU R miscellaneous functions by Frank Harrell
r-cran-its - GNU R package for handling irregular time series
r-cran-kernsmooth - GNU R package for kernel smoothing and density estimation
r-cran-lattice - GNU R package for 'Trellis' graphics
r-cran-lmtest - GNU R package for diagnostic checking in linear models
r-cran-mapdata - GNU R support for producing geographic maps (supplemental data)
r-cran-mapproj - GNU R support for cartographic projections of map data
r-cran-maps - GNU R support for producing geographic maps
r-cran-mcmcpack - routines for Markov Chain Monte Carlo model estimation in R
r-cran-mgcv - GNU R package for multiple parameter smoothing estimation
r-cran-nlme - GNU R package for (non-)linear mixed effects models
r-cran-qtl - [Biology] GNU R package for genetic marker linkage analysis
r-cran-rcmdr - GNU R platform-independent basic-statistics GUI
r-cran-rmysql - MySQL interface for R
r-cran-rodbc - GNU R package for ODBC database access
r-cran-rpart - GNU R package for recursive partitioning and regression trees
r-cran-rquantlib - GNU R package interfacing the QuantLib finance library
r-cran-statdataml - XML based data exchange format (R library)
r-cran-survival - GNU R package for survival analysis
r-cran-tkrplot - GNU R embedded Tk plotting device package
r-cran-tseries - GNU R package for time-series analysis and comp. finance
r-cran-vr - GNU R package accompanying the Venables and Ripley book on S
r-cran-xml - An XML package for the R language
r-noncran-lindsey - GNU R libraries contributed by Jim and Patrick Lindsey
r-omegahat-ggobi - GNU R package for the GGobi data visualization system
r-omegahat-rgtk - GNU R binding for Gtk
```