AUSTRIAN JOURNAL OF STATISTICS
Volume 41 (2012), Number 1, 59–66

Are There Too Many R Packages?

Kurt Hornik

Wirtschaftsuniversität Wien

Abstract: The number of R extension packages available from the CRAN repository has tremendously grown over the past 10 years. We look at this phenomenon in more detail, and discuss some of its consequences. In particular, we argue that the statistical computing community needs a more common understanding of software quality, and better domain-specific semantic resources.

Zusammenfassung: Die Anzahl der über das CRAN Repository verfügbaren R Erweiterungspakete ist in den letzten 10 Jahren enorm gewachsen. Wir untersuchen dieses Phänomen genauer, und diskutieren einige seiner Konsequenzen. Insbesondere argumentieren wir, dass die Statistical Computing Gemeinde ein gemeinsameres Verständnis von Softwarequalität, und bessere domänenspezifische semantische Ressourcen braucht.

Keywords: CRAN, Software Quality, Semantic Resources for Statistical Computing.

1 Introduction

Clearly, everyone will expect my contribution to this special issue to be about R and its role in computational statistics. Almost 10 years ago, in another special issue of this journal celebrating the 50th anniversary of the Austrian Statistical Society, Fritz Leisch and I contributed an article on the early history of the R project and Vienna's special role in it (Hornik and Leisch, 2002). Hence, a sequel covering what has happened since, or making some daring predictions on the future of R, to be compared with reality 10 years from now, might seem indicated. This paper will in fact discuss recent and future developments of the R project, but from a particular angle, looking at the amazing multitude of contributed R packages that have become available in the past 10 years in R package repositories, and specifically, on the Comprehensive R Archive Network (CRAN, `http://CRAN.R-project.org`). This focus is, of course, partially motivated by the fact that I have always played a very active role in maintaining the CRAN package repository, but mostly because I regard the development of R as a two-tier process where a core team provides the base system only, on which others can build by contributing extensions, typically in the form of packages. These packages are thus an integral part of the R computational environment, and clearly, the overwhelming majority of enhancements to this environment is being made available as contributed packages.

The R package system goes back to the mid-90s, and is based on two very simple ideas: a simple hierarchical layout for organizing the package contents (such as R code and documentation files or data sets), and a simple plain text database with the package metadata (including name, version and the license). As Fritz Leisch and I were early adopters of the Debian GNU/Linux system (`http://www.debian.org/`), the metadata

file ended up being named 'DESCRIPTION' and using the same simple tag-value format as Debian does for its control files. We wrote one (`build`) script for gathering the package contents and metadata into a single file (a suitably compressed tar format file representing a "source package"), and another (`INSTALL`) script for taking such a file and processing its contents so that R could subsequently use these. Initially, such source package files were distributed by simple file sharing mechanisms (email or manual retrieval from download areas). Again by looking at the Debian role model, the obvious next step was introducing package repositories: download areas containing the package files and a plain text database with all key package metadata, which the R package management system can read to find out about packages available for installation, or in need of updating (or removing). We soon realized that certain quality standards needed to be assured (e.g., that the key package metadata were available and usable, and that the packages were installable), and thus wrote another (`check`) script for doing so. And finally, Fritz and I (in 1997) started CRAN, and the first such package repository as its major component.

Clearly, the R package system has been a raving success. It is truly simple to create packages which others can use. In fact, packages can be employed as "containers" for all kinds of R extension material, such as data sets or documentation (manuals or books): because packages are versioned, keeping the extensions up-to-date can easily be achieved.

Whether the CRAN package repository has been a "success" is not so clear. Several years ago (certainly when the number of active packages on CRAN crossed the 1000 boundary) people started complaining that it would become increasingly hard to "find things" on CRAN or keep track of new packages on CRAN (in fact, *R News* and subsequently the *R Journal* thus far had a "News on CRAN" section with short descriptions of recently added packages, which has become increasingly useless [there were 384 such packages for the first issue in 2011], and hence will soon no longer be provided). Others have criticized the high variability of available packages in terms of perceived quality or usefulness. My standard reply is that I view package repositories as "warehouses" on which others can build "retail services", perhaps by providing portals ("myCRAN") maybe even allowing for personalizable recommendations, or specialized "views" to suitable subsets of the available contents or blogs making recommendations, or maybe deploying social bookmarking services. But quite interestingly and rather unfortunately, pretty much nothing along these lines has happened (CRAN itself added CRAN Task Views (Zeileis, 2005) and views based on ACM, JEL or MSC classifications). As of 2011-11-17, the number of active packages on CRAN is 3425: are these "too many" packages? In the following, I will investigate this question in some detail.

2 CRAN Packages

It helps to recall a few basic facts about CRAN. CRAN is "a collection of sites which carry identical material, consisting of the R distribution(s), the contributed extensions, documentation for R, and binaries" (Hornik, 2011). The CRAN master is at Wirtschaftsuniversität Wien; currently, there are 87 official "daily" mirrors, with perhaps many more unofficial ones. When source packages (obtained from the `build` script) are submitted to CRAN, they are checked by the gatekeeper (me, using the `check` script). If no problems

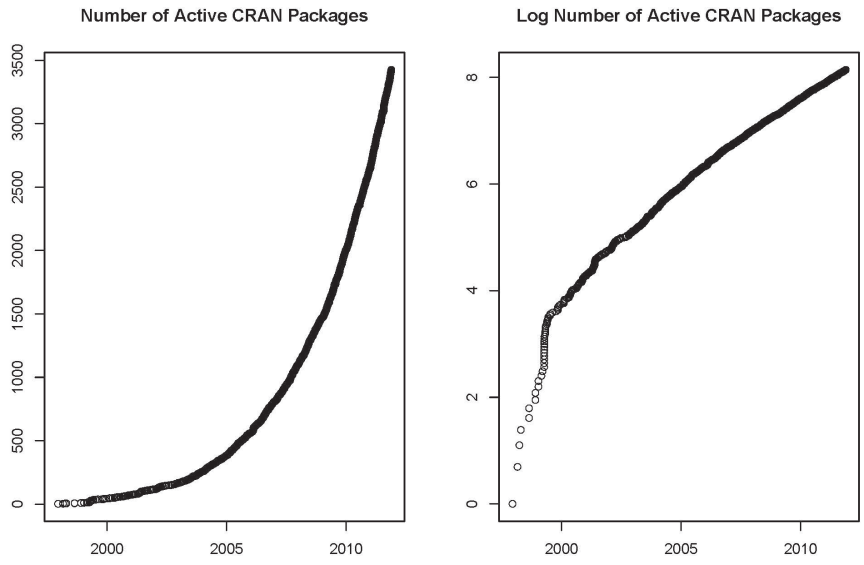

Figure 1: Development of the estimated number of active CRAN packages on a raw (left) and log (right) scale.

are diagnosed, the packages are *published* on CRAN by putting them into the main package area which contains all currently active packages, and updating the database for these. Older versions are *archived* (moved to the package archive area), but not removed and hence still available for download. Archivals can also happen when active packages (e.g., due to changes in R itself or other contributed packages) start having quality problems which are not addressed in due time. All packages ever published on CRAN are given a unique and *persistent* URL of the form `http://CRAN.R-project.org/package=`*`foo`*, which provides access to active and archived versions of the package.

2.1 Sizes

Figure 1 shows the development of the estimated number of active CRAN packages, both on a raw and on a log scale. I write "estimated" because it is unfortunately not possible to exactly reconstruct these numbers, as there is no explicit transaction logging of changes in the package repository. We can infer when a package was published by looking at its file timestamp (the "modification time"; since about 2008, the publication date-time is also recorded in the package metadata), but we cannot reliably tell when packages were possible archived, or resurrected from the archive. The plots actually show the cumulative publication counts of the currently active packages. We can clearly see the amazing, perhaps slightly "sub-exponential", growth of the number of CRAN packages.

Figure 2 shows the development of the overall size of the CRAN package repository, both in terms of the number of (active as well as archived) source package files and their aggregate file sizes in gigabytes (i.e., 10^9 bytes), both on a log-scale. The current number and total file size are 26152 and 17.97 GB, respectively. From a technology point of view,

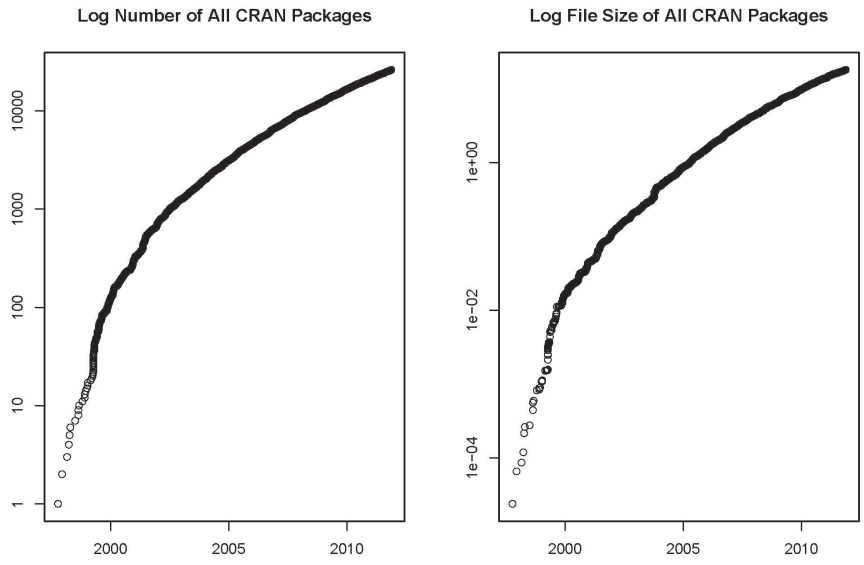

Figure 2: Development of the numbers and aggregate file sizes of all (active and archived) source packages on CRAN.

these figures are more relevant than the numbers and file sizes of the active packages, as they indicate the amount of data mirrors need to store and keep in sync with the master. In fact, the binary packages provided for the Mac OS X and Windows platform consume considerably more file space, and CRAN may move towards keeping archived source and binary packages in a central un-mirrored place eventually. In any case, the growth rates seem quite manageable, in particular when compared to similar figures in Linux distribution (see e.g. Wikipedia (2011a) for the development of the number of packages in Debian distributions).

Trying to predict the number of active CRAN packages in 5 or 10 years seems an obvious challenge: I will not accept it, both because I would first want to obtain a better understanding of the processes in the R user communities which drive the creation and dissemination of new packages, and because I can change the rules of the game through policy changes (e.g., by no longer accepting submissions with non-FOSS licenses, or violating resource constraints). However, prediction competitions seem very popular these days: I am planning to make a data set containing the CRAN package (estimated) publication dates and file sizes available as a CRAN web service in the near future.

2.2 Relations

As part of their metadata, R packages can declare on which other packages they depend. There are actually several dependency types; for what follows, let us say that package P *directly strongly depends* on package Q if P cannot successfully be loaded (and hence installed) without Q being loaded (and hence installed; technically, this corresponds to the `Depends`, `Imports` or `LinkingTo` predicates in the metadata), and that P *recursively*

Table 1: Summaries of the distributions of the numbers of direct and recursive reverse strong dependencies for the active CRAN packages as of 2011-11-17.

	Min	Q_1	Median	Mean	Q_3	Max
Direct dependencies	0.000	0.000	0.000	1.238	1.000	127.000
Recursive dependencies	0.000	0.000	0.000	1.937	1.000	236.000

strongly depends on Q if there is a path of direct strong dependencies from P to Q. Let $N_d(Q)$ and $N_r(Q)$ be the numbers of packages P which directly or recursively strongly depend on Q, respectively, i.e., which are *reverse* strong dependencies of Q.

Table 1 summarizes the distributions of these numbers for the currently active CRAN packages (excluding the so-called recommended ones which are shipped with the R base distribution). These summaries show that the package dependency relation network is extremely sparse. In fact, about 74 % of the packages have no strong dependencies, and about 12 % (or 10 %) have exactly one direct (or recursive) reverse strong dependency. These findings seem to indicate that most packages provide specialized solutions mostly built around the base R functionality, and hence can be used—if needed—with rather low incremental learning effort required. On the other hand, the packages with the highest numbers of reverse strong dependencies include those providing interfaces to C++, Java and XML, as well as **mvtnorm** (Genz et al., 2011), **sp** (Pebesma and Bivand, 2005), **coda** (Plummer et al., 2006) and **rgl** (Adler and Murdoch, 2011) providing, respectively, functionality for the multivariate normal (and t) distribution, basic classes and methods for spatial data, tools for MCMC output analysis and diagnostics, and 3-d visualization, which rather nicely seems to fit the picture of providing functionality quite commonly needed but not provided by base R.

Does a low number of reverse dependencies indicate low quality or usefulness? And hence, given the above, are there too many CRAN packages that "no one really needs"? I do not regard this a valid conclusion. All such judgments will eventually be subjective, or perhaps be based on the common understanding of a community of subjects, and the heterogeneity between R user communities is amazingly large in terms of application scopes, needs and preferences. Experience with the Journal of Statistical Software (`http://www.jstatsoft.org`), which has reviewed of a considerable number of R packages as an integral part of its submissions, has shown that leading scholars in statistical computing rather disagree when it comes to assessing the quality of statistical software.

2.3 Contents

Analyzing the package dependency structure only covers one aspect of package relatedness. Package authors often seem to prefer copying code from other packages than depending on these, or collaborate to develop a common infrastructure. If packages provide different solutions to similar tasks, they will evidently rather compete than cooperate through hierarchical dependencies. (They may share common dependencies in such cases, but as we have seen, dependencies are rather few in general.) Measuring similarity should thus be based on the *concepts* the packages relate to, and perhaps the software design pat-

terns they employ. Following modern information science, the set of relevant concepts would, along the relationships between the concepts, be formally represented by an ontology, which can then be employed to "reason about the entities within that domain" (Wikipedia, 2011c), in our case, about the CRAN packages in the knowledge domain of statistical computing.

Unfortunately, we are unaware of the existence of such semantic resources for statistical computing (or in fact, statistical science as a whole). To find packages with desired functionalities, or more generally, packages related to concepts of interest, such resources are urgently needed. Hornik and Murdoch (2011) show how a seed statistical (computing) dictionary can be obtained by running spell-checking tools on the R documentation files contained in the CRAN packages, and determining the most frequent terms flagged as possibly possibly mis-spelled (i.e., not contained in the basic dictionaries). Clearly, a considerable and concerted community effort will be needed to create the needed semantic resources. I think that CRAN can play a double role here, by "providing R packages to compute on R packages": certainly, with currently 75339 R documentation files in 3425 packages (and an even larger number of terms used), analyzing the texts in CRAN packages should constitute an attractive "large data set" challenge.

One might think that package authors should themselves apply good citizenship principles and indicate that their work was well researched by providing metadata indicating the relation of their package to other packages, or common concepts. I see at least two issues with this idea. One, such provisions are subjective and hence cannot be taken as authoritative (and therefore require establishing community consensus mechanisms and procedures). And two, there is a chicken and egg problem: it is in fact hard to find out which packages provide certain functionalities. But at least, they are conveniently available for searching them: the persistence of the CRAN package URLs implies that (as long as the CRAN package repository exists) packages are *reliable resources* which can and should be referenced (i.e., cited) by scientific works such as articles or other packages. I think this is a fundamental observation, and perhaps the most important service provided by CRAN: providing a reliable package warehouse.

The core aspect of publishing a package on CRAN is making it available for everyone else to access, i.e., making it "public" (which of course is what "publishing" actually means). Of course, one can argue how much control such be exercised when doing so. The current CRAN policies (packages must legally be redistributable by CRAN, pass `check` against the current release version of R, and use resources fairly) are quite permissive. It is instructive to compare this policy to the ones employed by Linux distributions and electronic preprint servers, which provide similar publication services: these are typically more meritocratic, or even peer reviewed. For example, Debian has quite strict processes for adding new Debian Maintainers or Debian Developers (e.g., `http://wiki.debian.org/DebianMaintainer`), which require one or more Debian Developers to advocate or sponsor an application. The arXiv.org e-print archive (`http://arxiv.org/`) has always employed a collection of moderators which review submissions and may re-categorize any that are deemed off-topic, and in 2004 added a system under which authors must first get endorsed, with endorsement coming "from either another arXiv author who is an endorser or is automatic, depending on various evolving criteria, which are not publicly spelled out" (Wikipedia, 2011b). Should CRAN become less permissive? I have my

doubts. In today's information society, it is basically always possible to publish articles or packages, as in principle everyone can start their own publication service, with varying degrees of reliability. I am convinced that scientific communities with an interest in R are better served by a few comprehensive package repositories which reliably provide package storage services, on which others can base additional services.

3 Conclusions

The number of R extension packages made available via CRAN has tremendously grown over the past 10 years, and most likely will continue to do so in the next 10 years. I take this as an indication of a renaissance in statistical computing, with tremendous increases in activity, productivity, and importance. Certainly, we would not honor Rudolf Dutter as one of the pioneers of modern statistical computing to whom this special issue is dedicated by not welcoming and embracing this renaissance—and hence the growth of CRAN. However, the growth also poses major challenges to the statistical computing community: it needs to work towards a common understanding of software quality, and to develop metadata resources for statistical software. This should start with journals appropriately providing information about reliable software resources employed in their publications, and making this information available to the public in machine readable form, so that it can be used in developing and enhancing networks of scholarly knowledge. In addition, statistical science needs domain-specific semantic resources (such as dictionaries and thesauri), which can be employed for mapping the knowledge space underlying modern statistical computing resources. (Clearly, these needs are not specific to CRAN or R.) Bluntly speaking, we need better data to do computational statistics on statistical computing solutions: certainly, the CRAN package repository should be an extremely valuable resource for the community, both for gathering and analyzing such data.

References

Daniel Adler and Duncan Murdoch. **rgl***: 3D visualization device system (OpenGL)*, 2011. URL `http://CRAN.R-project.org/package=rgl`. R package version 0.92.798.

Alan Genz, Frank Bretz, Tetsuhisa Miwa, Xuefei Mi, Friedrich Leisch, Fabian Scheipl, and Torsten Hothorn. **mvtnorm***: Multivariate Normal and t Distributions*, 2011. URL `http://CRAN.R-project.org/package=mvtnorm`. R package version 0.9-9991.

Kurt Hornik. The R FAQ, 2011. URL `http://CRAN.R-project.org/doc/FAQ/R-FAQ.html`. ISBN 3-900051-08-9.

Kurt Hornik and Friedrich Leisch. Vienna and R: Love, marriage and the future. In Rudolf Dutter, editor, *Festschrift 50 Jahre Österreichische Statistische Gesellschaft*, pages 61–70. Österreichische Statistische Gesellschaft, 2002. ISSN 1026-597X.

Kurt Hornik and Duncan Murdoch. Watch your spelling! *R Journal*, 2011. To appear.

Edzer J. Pebesma and Roger S. Bivand. Classes and methods for spatial data in R. *R News*, 5(2):9–13, November 2005. URL http://CRAN.R-project.org/doc/Rnews/.

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006. URL http://CRAN.R-project.org/doc/Rnews/.

Wikipedia. Debian — Wikipedia, the free encyclopedia, 2011a. URL http://en.wikipedia.org/w/index.php?title=Debian&oldid=461095814. [Online; accessed 17-November-2011].

Wikipedia. Arxiv — Wikipedia, the free encyclopedia, 2011b. URL http://en.wikipedia.org/w/index.php?title=ArXiv&oldid=460023627. [Online; accessed 18-November-2011].

Wikipedia. Ontology (information science) — Wikipedia, the free encyclopedia, 2011c. URL http://en.wikipedia.org/w/index.php?title=Ontology_(information_science)&oldid=459398883. [Online; accessed 18-November-2011].

Achim Zeileis. CRAN task views. *R News*, 5(1):39–40, 2005. URL http://CRAN.R-project.org/doc/Rnews/.

Author's address:

Kurt Hornik
Institute for Statistics and Mathematics
Vienna University of Economics and Business
Augasse 2–6
1090 Wien
Austria