

Release management and governance structure of the R project

Peter Dalgaard

Center for Statistics
Copenhagen Business School

UseR!, Warsaw, July 8, 2026

Outline

Intro

Brief history of R

Release management

Governance

Challenges for the Future

The R Core keynote

- ▶ We like to remind R users that R didn't and doesn't just appear out of nothing
- ▶ There is a dedicated group of people who has been working for decades on improving R and keeping it stable and with minimal flaws
- ▶ (Recent events may have made this somewhat less necessary to say...)

Governance structures

- ▶ The Core Team
- ▶ CRAN ecosystem and the CRAN team
- ▶ R Foundation
- ▶ R Consortium

Origins of R

- ▶ Local development at Auckland (Ross/Robert) c. 1992–1993
- ▶ Martin Mächler convinces R&R to release under GPL 1995
- ▶ CRAN 1997
- ▶ Core Team extension August 1997
- ▶ First in-person meeting at DSC in March 1999, Vienna
- ▶ Decision to have 1.0.0 on the nerdy date 2000-02-29

Date: Sat, 16 Aug 1997 09:15:45 +1200 (NZST)
From: Ross Ihaka <ihaka@stat.auckland.ac.nz>
To: Kurt.Hornik@ci.tuwien.ac.at, p.dalgaard@kubism.ku.dk,
thomas@biostat.washington.edu
Subject: Invitation ...
Cc: maechler@stat.math.ethz.ch, rgentlem@stat1.stat.auckland.ac.nz

We have had a bit of a discussion on enlarging the R "core team" At present this seems to consist of Robert, Martin and myself although the following people also have commit privileges in the CVS tree:

Luke Tierney	no introduction needed
Heiner Schwarte	developer of dyn.load etc
Paul Murrell	my PhD student

[...]

As major contributors (and apparently sane people) we would like to invite you to be part of the R "core team".

[...]

We can't promise you anything much in return, except a free copy of R :-) and perhaps a publication on "distributed development of statistical software". Since you are clearly hopeless software junkies, perhaps you don't need any more incentive.

[...]

DSC 1999



Statistical computing landscape when R began

- ▶ PC becoming capable and available
- ▶ Linux, Free Software, GPL, “contribute a little, get a lot back”
- ▶ Statistical software tied to mainframe computers: SAS dominant, with fringe markets for Genstat, S-PLUS, expensive
- ▶ Code development for new methods was . . . idiosyncratic (StatLib)
- ▶ New methods not easily integrated in statistical platforms

S and R brought:

- ▶ Full programming language (as opposed to SAS)
- ▶ Integration of data handling and analysis
- ▶ Flexible graphics subsystem
- ▶ Model formulas
- ▶ Seamless programming of extensions
- ▶ Dynamic loading of C/Fortran code (much easier with R)

The CRAN ecosystem

- ▶ Created a marketplace for decentralized development of packages
- ▶ Ease of installation
- ▶ Ensure at least a minimum level of documentation
- ▶ Ensure package *maintenance* – keeping up with changes in computing platforms

Growth of R

- ▶ Improvements to the core language – Garbage collection, large vectors, byte compiler. . .
- ▶ Increasing automation of package checking and release procedures
- ▶ R Foundation formed 2002
- ▶ Explosive interest, both users and developers
- ▶ CRAN reaching 10000+ packages, becoming unwieldy
- ▶ Conferences going from classroom size to main lecture hall and beyond

useR! 2008 (Dortmund)



useR! 2009 (Rennes)



Consolidation, cultural change

- ▶ Cutesy names 2011
- ▶ R becomes mainstream, Microsoft involved
- ▶ R Consortium 2014
- ▶ RStudio and surrounding prescriptive culture easing entry for new users
- ▶ Tidyverse, ggplot
- ▶ Data Science becomes a thing
- ▶ Mailing lists traffic moving to StackExchange, GitHub, etc.

Intro

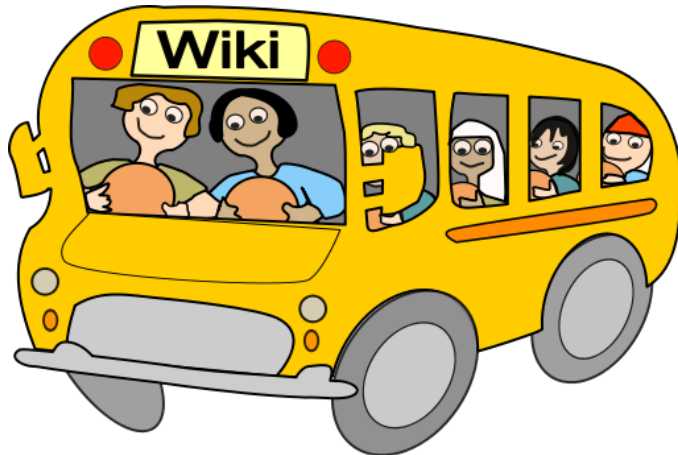
Brief history of R

Release management

Governance

Challenges for the Future

Steering mechanism



Basic principles

- ▶ Keep the base code maintainable
- ▶ New functionality mostly via packages, with separate maintainers
- ▶ *“Move slowly and do not break things”*
- ▶ **No** master plan — core developers can do what they feel is desirable (and face the consequences). Usually after discussions, though

Computing platforms

- ▶ R works on Windows, MacOS X and multiple Unix/Linux variants
- ▶ ... *and* their variants over time
- ▶ Graphical User Interfaces (GUIs)
- ▶ Operating system interfaces
- ▶ Compilers
- ▶ Binary packaging formats
- ▶ Build configuration (flags, libraries, etc.)

Automating things

- ▶ Version control: “Subversion” repository in Zurich
- ▶ (Yes, we have heard about GitHub, but . . .)
- ▶ Makefiles
- ▶ Configure script to find system dependencies, autoconf to generate it
- ▶ Common help file format (`.Rd`), tools to generate HTML, latex, an plain text help files
- ▶ Package format specification and build tools

Testing and checking

- ▶ Formal checks — documentation consistent with function definitions, all fields present in package descriptions, all examples runnable, etc
- ▶ Run all examples and check for errors
- ▶ Specific testing, regression tests, targeted consistency checks
- ▶ Automated checks of all packages (CRAN)
- ▶ Room for improvement: Code coverage, etc.

Development principles

- ▶ Statistical software is somewhat special
 - ▶ High emphasis on correctness
 - ▶ Large amounts of legacy code (FORTRAN!)
- ▶ —→ Conservative, carefully tested releases
- ▶ NOT stable/unstable releases like Linux kernel

Release process

- ▶ Need feedback from users, especially those on obscure platforms
- ▶ Need *stability*
- ▶ Phased releases (alpha/beta/RC)

One-year cycles

- ▶ We have gravitated toward a main release (x.y.0) in April and 3-4 patch releases per year.
- ▶ Wrap-up version shortly before next main release.
- ▶ Typically x.y.1 will be installed at academic institutions in sync with the academic calendar
- ▶ No new features before next release

The `pbkrtest` messup

- ▶ We did try to move towards more flexible releases around R version 3.x
- ▶ Then package `pbkrtest` — tests in unbalanced mixed models — introduced a dependency on $R \geq 3.2.3$ (the `sigma` function)
- ▶ Package `Rcmdr` used `pbkrtest`...
- ▶ ...and wouldn't install at educational institutions (of which very few actually needed `pbkrtest`)
- ▶ So we became more careful again

R governance structures

Major players

- ▶ R Core
- ▶ CRAN Team
- ▶ R Foundation
- ▶ R Consortium

Notice that these are separate entities, with no explicit hierarchy.

R Core

- ▶ Currently 19 members, varying level of activity (within/between members)
- ▶ Core maintenance
- ▶ Bug fixing
- ▶ New fundamental features, (byte compiler, garbage collector, | >, etc)
- ▶ (Tomáš Kalibera had a major hand in lots of this)
- ▶ Orchestrating releases

CRAN team

- ▶ Predates (!) R Core by half a year or so
- ▶ Original idea from Vienna Group, parallel to CPAN, CTAN
- ▶ Subset of R Core, but autonomous
- ▶ Maintains CRAN repository
- ▶ Processing submissions
- ▶ Checking packages vs R versions, compilers, etc.
- ▶ QC/QA checking: seeking out bad coding practices
- ▶ Ensures that maintainers actually maintain

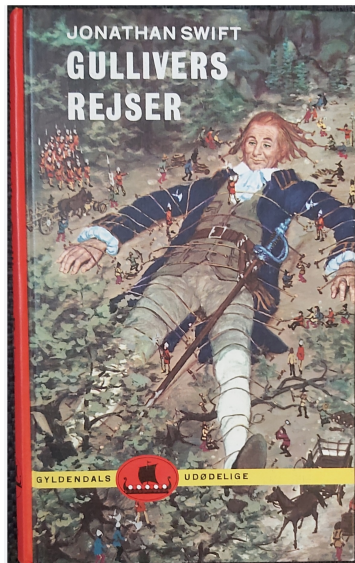
What R Core may look like



However...

- ▶ Changing things in base R easily has unexpected consequences when not just plain bug-fixes
- ▶ Apparently simple changes can break multiple CRAN packages
- ▶ E.g. Ravi Varadan pointed out that regression diagnostics misbehaves for `glm` objects, because it should use working, not deviance residuals
- ▶ Turns out that a number of packages reused the code for non-`lm/glm` objects, so broke on `residuals(type="working")`
- ▶ Plus further complications trying to work around that

What R Core feels like



R Foundation

- ▶ Originally, extension of R Core
- ▶ Planned to hold the R Copyright, but that proved impossible
- ▶ Formal institution with bylaws, board, elections etc.
- ▶ Open board meetings (spanning 18 time zones!)
- ▶ Conferences, meetings
- ▶ Grant applications
- ▶ Financed by donations, conference profits (if any. . .)

R Consortium

- ▶ Industry initiative, c. 2014
- ▶ Members: RF + Platinum/Silver members (paying)
- ▶ (Posit, Microsoft, etc.)
- ▶ Infrastructure Steering Committee
 - ▶ ISC Projects (smallish grants)
 - ▶ Financed 50% of Tomáš Kalibera

Longer-term issues

- ▶ Rejuvenation
 - ▶ We're getting older
 - ▶ Core team was 40-ish, 25 years ago.
- ▶ Single points of failure
- ▶ Governance structures are somewhat fragile. Strategic decisions are hard to make.
- ▶ Diversity, recruitment
- ▶ Generic issues of getting academic merit

Strategic issues

- ▶ Perhaps we are a bit too much Data Science and too little Statistics?
- ▶ The CRAN “Wild West”: can check packages formally, but what packages do is out of control.
- ▶ Do we want to have tighter control? How and by whom?
- ▶ R Core has to some extent shied away from taking on more than we could maintain

Possible paths forward

- ▶ May need to recruit in the same way as in 1997: Among established scientists who just cannot stay away...
- ▶ Grants/investments — R is not the only project with an age issue
 - ▶ Sovereign Tech Fund
 - ▶ Software Sustainability Institute: Research Software Maintenance Fund (Heather Turner, R Dev Day/Sprint, etc.)

Summary

- ▶ R evolved as a classical “scientific revolution” (a.m. Kuhn)
- ▶ It has been maintained carefully and conservatively
- ▶ By and large its governance structures are in pretty good shape, with some fragile points (as often seen in scientific institutions)
- ▶ Rejuvenation is becoming a pressing issue as professors turn to emeritus status