R Core Team: Development Principles and War Stories

Peter Dalgaard

Center for Statistics
Copenhagen Business School

R Users Conference in Korea, May 30th 2014

Outline

Outline

R's history in brief

- S (c.1985), written by John Chambers et al. (ACM software systems award 1998)
- Started approx.1994, Ross Ihaka & Robert Gentleman in Auckland
- Available via Internet 1996
- Version 1.0.0 2000-02-29
- Version 2.0.0 2004-10-04

Formation of the Core Team

1996 R mailing lists, testing, debugging, discussing
1997 Invitation to join Core Team
1999 First physical meeting in Vienna

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.
[...]

Outline

People on the Core Team

Ross Ihaka, Robert Gentleman Founding fathers

Kurt Hornik, Fritz Leisch CRAN maintenance, QC issues

Uwe Ligges Windows builder

Simon Urbanek Mac builds and support, Bugzilla

Brian Ripley Translation coordinator. Testing + a lot of stuff all over the place

Martin Mächler Mailing lists, Matrix, lme4 package

Luke Tierney Computer language guru, compiler, garbage collector, parallel processing

Peter Dalgaard Release management

Duncan Murdoch Parser improvements, debugging

Paul Murrell Graphics

People on the Core Team, cont.

John Chambers S4 methods

Duncan Temple Lang Inter-language interfaces

Thomas Lumley Survey package, general biostats powerhouse

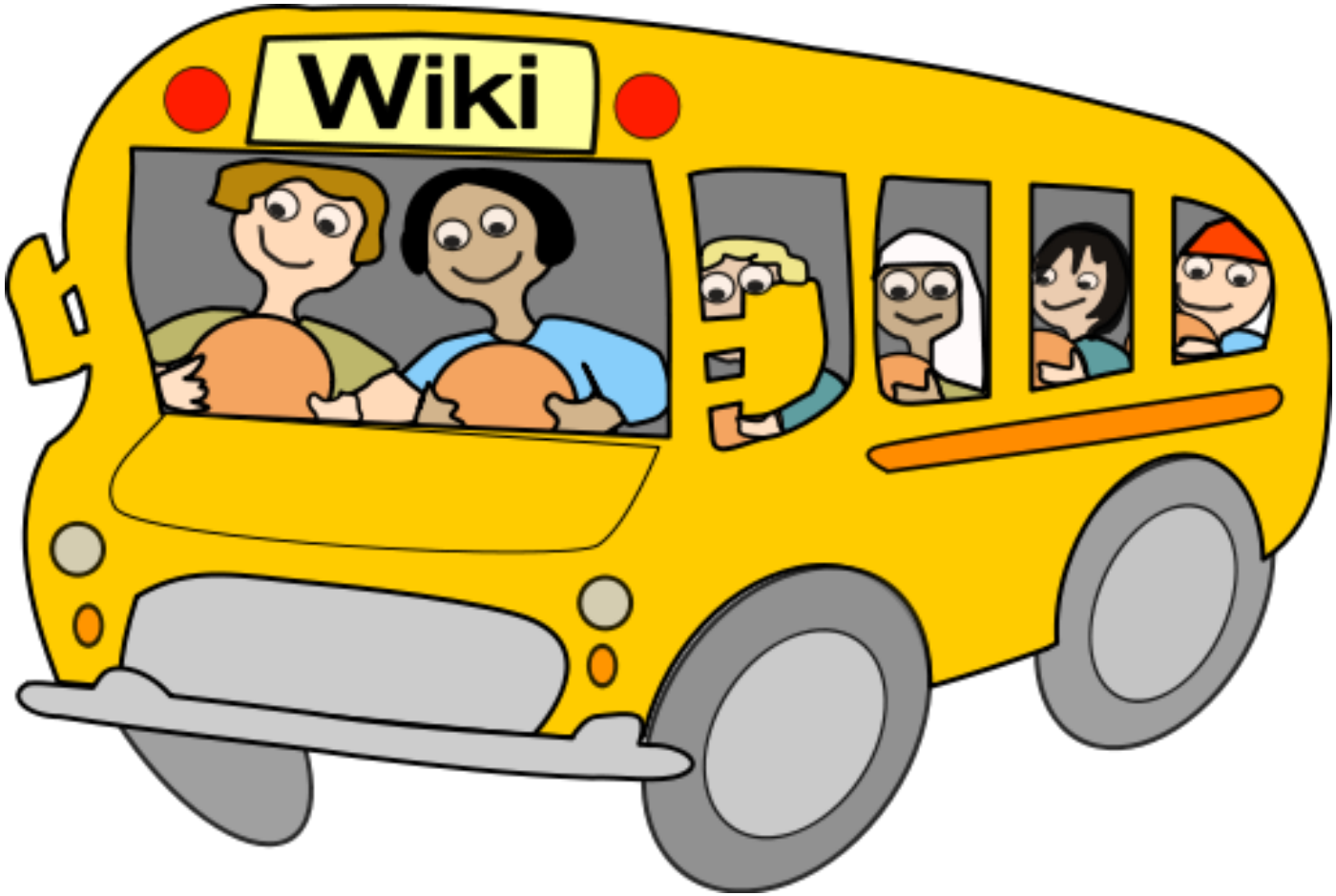
Deepayan Sarkar Lattice graphics

Martyn Plummer Fedora/Red Hat connection

Currently inactive Seth Falcon, Stefano Iacus, Doug Bates

Outline

Steering



Themes

- Issues we need to tackle
- Tools to make the task manageable
- Release mechanisms

Basic principles

- Keep the base code maintainable
- New functionality via packages, with separate maintainers
- Careful and conservative development – avoid breaking CRAN packages unnecessarily
- **No** master plan — core developers can do what they feel is desirable (and face the consequences)

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
- Binary packaging formats
- Build configuration (flags, libraries, etc.)

Automating Things

- Version control (“Subversion” repository in Zurich)
- The `make` program describes how to build targets from subtargets using *makefiles*
- The GNU Autotools create the `configure` script that sorts out all the little system dependencies.
- 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 check — 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

Development Principles

- Statistical software development is slightly special compared to other software
  - High emphasis on correctness
  - Large amounts of legacy code (FORTRAN!)
  - Dependency on many tools
- Conservative, carefully tested releases

Release Process

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

Outline

1. Development process sensitivity

The following story is a little technical, but it shows just how easily apparently minor modifications can destabilize things (notice the dates!)

Exotic languages

Back in 2008 when R 2.7.1 was current, we received two reports about R crashing under Windows XP. Both from machines in Korea. The second one came in on August 7. At that point Brian Ripley set out to investigate and traced the issue to a faulty translation file, which he removed from the sources on August 11.

Rarely checked build steps

Meanwhile, on August 10 we had scheduled R 2.7.2 for August 25. When the release process started on the 15th, it turned out that the source package wouldn't build. The `make dist` command listed the now absent translation file as a dependency.
So I commented out the offending line in the Makefile — somewhat carelessly, as it turned out.

Rare operating system anomalies

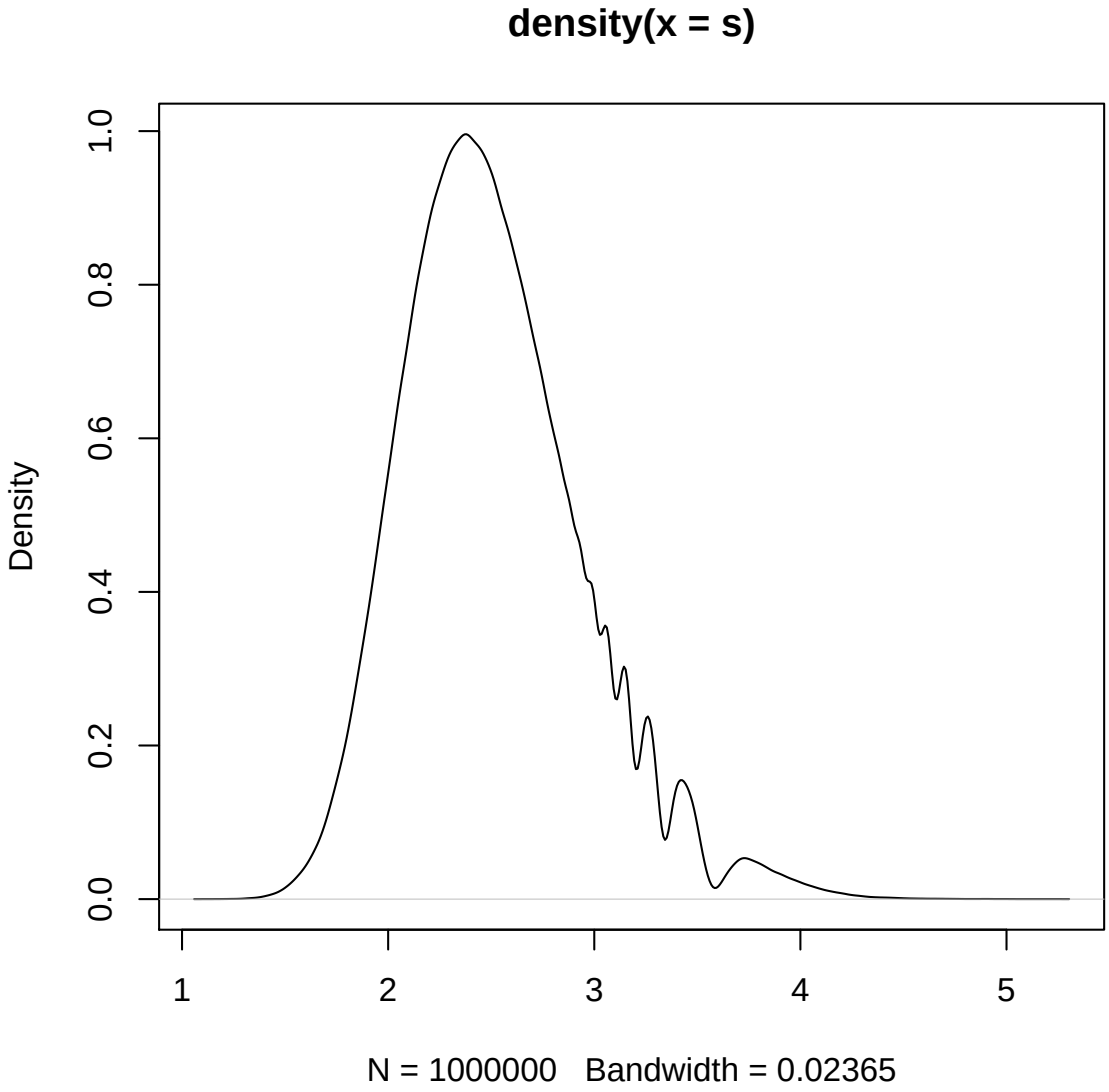
Not all versions of the `make` program allow you to use comments. In particular, R would now not install on FreeBSD, a platform that noone in R Core was using at the time. Fortunately, someone reported the problem on August 17, just before the code freeze on August 18.

2. Random Normals

In several cases, R has been instrumental in finding obscure bugs in algorithms

- Around version 1.6, R's random number mechanism used
  - the *Marsaglia-Multicarry* method as the base uniform generator
  - the *Kinderman-Ramage* method for normal variate generation
- Someone came up with the following example which shows a set of strange ripples

```
RNGversion("1.6.0")
s <- replicate(1e6,max(rnorm(100)))
plot(density(s))
```


density(x = s)
Density
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
N = 1000000 Bandwidth = 0.02365

Random Normals, cont'd

- Replacing the uniform generator removed the problem
- Replacing the normal variate method *also* removed the problem
- The Kinderman-Ramage method is quite complicated, but part of it is that
  - one uniform number is used to decide whether the random value is out in the tail
  - a second independent uniform number is used to decide where in the tail
- This particular usage pattern revealed an unfortunate aspect of the Marsaglia-Multicarry method (therefore no longer the default).

Summary

- R Core has a conservative and careful approach to the maintenance of R
- This can be a tricky job at times
- However, it contributes to
  - The credibility of R as a viable platform for statistical analysis
  - The long-term validity and reliability of statistical algorithms