



History and Ecology of R

Martyn Plummer

International Agency for
Research on Cancer

ANF R avancé et performances
Aussois 6 Oct 2015

Pre-history

Before there was R , there was S .

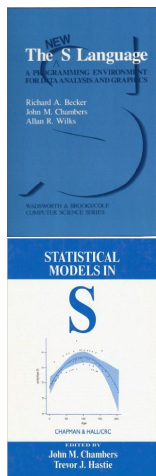
The S language

Developed at AT&T Bell laboratories by Rick Becker, John Chambers, Doug Dunn, Paul Tukey, Graham Wilkinson.

Version 1	1976–1980	Honeywell GCOS, Fortran-based
Version 2	1980–1988	Unix; Macros, Interface Language
	1981–1986	QPE (Quantitative Programming Environment)
	1984–	General outside licensing; books
Version 3	1988–1998	C-based; S functions and objects
	1991–	Statistical models; informal classes and methods
Version 4	1998	Formal class-method model; connections; large objects
	1991–	Interfaces to Java, Corba?

Source: Stages in the Evolution of S <http://ect.bell-labs.com/sl/S/history.html>

The “Blue Book” and the “White Book”

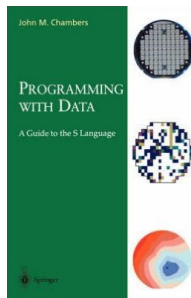


Key features of S version 3 outlined in two books:

- Becker, Chambers and Wilks, *The New S Language: A Programming Environment for Statistical Analysis and Graphics* (1988)
 - Functions and objects
- Chambers and Hastie (Eds), *Statistical Models in S* (1992)
 - Data frames, formulae

These books were later used as a prototype for R.

The “Green Book”



Key features of S version 4 were outlined in Chambers, *Programming with Data* (1998).

- S as a programming language
- Introduced formal classes and methods, which were later introduced into R by John Chambers himself.

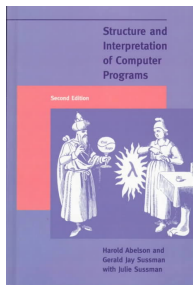
The Rise and Fall of S-PLUS

- 1988. Statistical Science releases first version of S-PLUS.
- 1993. Acquires exclusive license to distribute S. Merges with Mathsoft.
- 2001. Changes name to Insightful.
- 2004. Purchases S language for \$2 million.
- 2008. Insightful sold to TIBCO. S-PLUS incorporated into TIBCO Spotfire.

History

How R started, and how it turned into an S clone

The Dawn of R



- Ross Ihaka and Robert Gentleman at the University of Auckland
- An experimental statistical environment
- Scheme interpreter with S-like syntax
 - Replaced scalar type with vector-based types of S
 - Added lazy evaluation of function arguments
- Announced to *s-news* mailing list in August 1993.

A free software project

- June 1995. Martin Maechler (ETH, Zurich) persuades Ross and Robert to release R under GNU Public License (GPL)
- March 1996. Mailing list *r-testers* mailing list
 - Later split into three *r-announce*, *r-help*, and *r-devel*.
- Mid 1997. Creation of *core team* with access to central repository (CVS)
 - Doug Bates, Peter Dalgaard, Robert Gentleman, Kurt Hornik, Ross Ihaka, Friedrich Leisch, Thomas Lumley, Martin Maechler, Paul Murrell, Heiner Schwarte, Luke Tierney
- 1997. Adopted by the GNU Project as “GNU S”.

The draw of S

“Early on, the decision was made to use S-like syntax. Once that decision was made, the move toward being more and more like S has been irresistible”

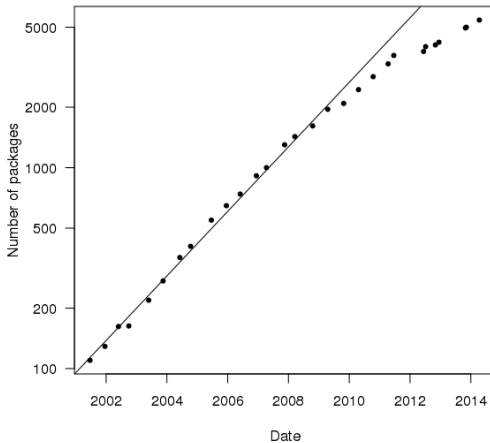
– Ross Ihaka, *R: Past and Future History* (Interface '98)

R 1.0.0, a complete and stable implementation of S version 3, was released in 2000.

Packages

- Comprehensive R Archive Network (CRAN) started in 1997
 - Quality assurance tools built into R
 - Increasingly demanding with each new R release
- Recommended packages distributed with R
 - Third-party packages included with R distribution
 - Provide more complete functionality for the R environment
 - Starting with release 1.3.0 (completely integrated in 1.6.0)

Growth of CRAN



The present

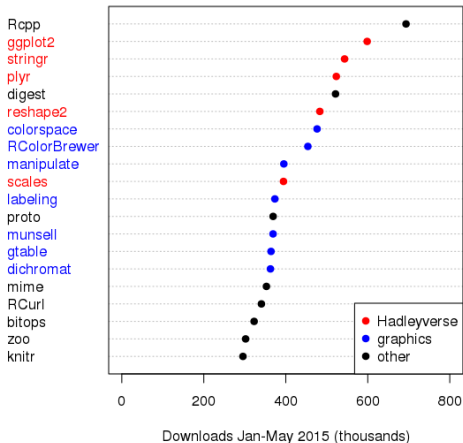
The current era is characterized by

- A mature R community
- Large penetration of R in the commercial world (“data science”, “analytics”, “big data”)
- Increasing interest in the R language from computer scientists.

Community

- UseR! Annual conference
 - Alternating between Europe and N. America
- R Journal.
 - Journal of record, peer-reviewed articles, indexed
 - Also Journal of Statistical Software (JSS) has many articles dedicated to R packages.
- Migration to social media
 - Stack Exchange/Overflow, Github, Twitter (#rstats)

Much important R infrastructure is now in package space



Commercial R

Several commercial organizations provide commercial versions of R including support, consulting, ...

- Revolution Computing, later Revolution Analytics (2007–2014), purchased by Microsoft.
- RStudio (2010–)
- Mango Solutions

Validation and Reliability

- *R: Regulatory Compliance and Validation Issues* guidance document by The R Foundation
- ValidR by Mango Solutions
- MRAN, a time-stamped version of CRAN
 - Allows analysis to be re-run with exactly the same package versions at a later date.
 - Used by Revolution R Open

Attack of the Clones (and forks)

Name	Implementation	Commercial sponsor	Open source
pqR	C fork		Yes
CXXR	C++ fork	Google	Yes
ORBIT	C fork	Huawei	Yes
Renjin	Java	BeDataDriven	Yes
FastR	Java (Truffle/Graal)	Oracle	Yes
Riposte	C++	Tableau Research	Yes
TERR	C++	TIBCO	No

A number of projects have looked improving the efficiency of R, either by forking the original codebase or by re-implementing R.

The R Foundation for Statistical Computing

A non-profit organization working in the public interest, founded in 2002 in order to:

- Provide support for the R project and other innovations in statistical computing.
- Provide a reference point for individuals, institutions or commercial enterprises that want to support or interact with the R development community.
- Hold and administer the copyright of R software and documentation (This never happened)

The R Consortium

In 2015, a group of organizations created a consortium to support the R ecosystem:

R Foundation A statutory member of The R Consortium

Gold members Microsoft, RStudio

Silver members TIBCO

Bronze members Alteryx, Google, Hewlett Packard, Ketchum
Trading LLC, Mango Solutions, Oracle

The Future

*“Prediction is very difficult, especially about the future”
– variously attributed to Niels Bohr, Piet Hein, Yogi Bera*

Trends

We cannot make predictions, but some long-term trends are very visible:

- Average age of R Core Team?
- Younger R developers more closely associated with industry than academia
- R Consortium provides mechanism for substantial investment in R infrastructure

R language versus R implementation

- R has no formal specification
- R language is defined by its implementation (“GNU R”)
- Long-term future of R may depend on formal specification of the language, rather than current implementation.

Simply start over and build something better

The x in this function is randomly local or global

```
f = function() {  
  if (runif(1) > .5)  
    x = 10  
  x  
}
```

“In the light of this, I’ve come to the conclusion that rather than “fixing” R, it would be better and much more productive to simply start over and build something better” – Ross Ihaka, Christian Robert’s blog, September 13, 2010

Back to the Future

Ross Ihaka and Duncan Temple Lang propose a new language built on top of common lisp with:

- Scalar types
- Type hinting
- Call-by-reference semantics
- Use of multi-cores and parallelism
- More strict license to protect work donated to the commons

