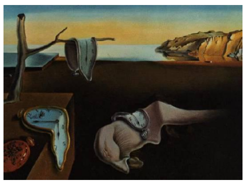


Journal of Statistical Software

*September 2016, Volume 73, Issue 2.* *doi: 10.18637/jss.v073.i02*

# R and the *Journal of Statistical Software*

**John Fox**
McMaster University

**Allison Leanage**
McMaster University

**Abstract**

The *Journal of Statistical Software* was founded by Jan de Leeuw in 1996, the year before the Comprehensive R Archive Network (CRAN) first made R and contributed R packages widely available on the Internet. Within a few years, R came increasingly to dominate contributions to JSS. We trace the continuing development of R and CRAN, and the representation of R and other statistical software in the pages of JSS.

*Keywords*: R statistical computing environment, JSS, statistical software.

## 1. Introduction

Jan de Leeuw founded the *Journal of Statistical Software* in 1996 with the general objectives of providing an outlet for work on free statistical software, promoting free statistical software and open-access publishing, and increasing the visibility of the then-new UCLA Department of Statistics (see De Leeuw and Mullen 2014 for a brief history). In the same year, the Comprehensive R Archive Network (CRAN) was born, and began to distribute the R statistical computing environment (R Core Team 2016) and contributed R packages on the Internet. As we will show, JSS and R grew up together (as De Leeuw and Mullen 2014 acknowledge), and R, now the preeminent platform for developing free statistical software, came increasingly to dominate the pages of JSS.

Section 2 of this paper briefly reviews the history and development of R. Section 3 traces the representation of various software, including R, in JSS. Section 4 is devoted to speculative concluding remarks.

## 2. The continuing development of R

As described by Ihaka and Gentleman (1996) and Ihaka (1998), R began in the early 1990s as the personal project of Ross Ihaka and Robert Gentleman, both then at the University

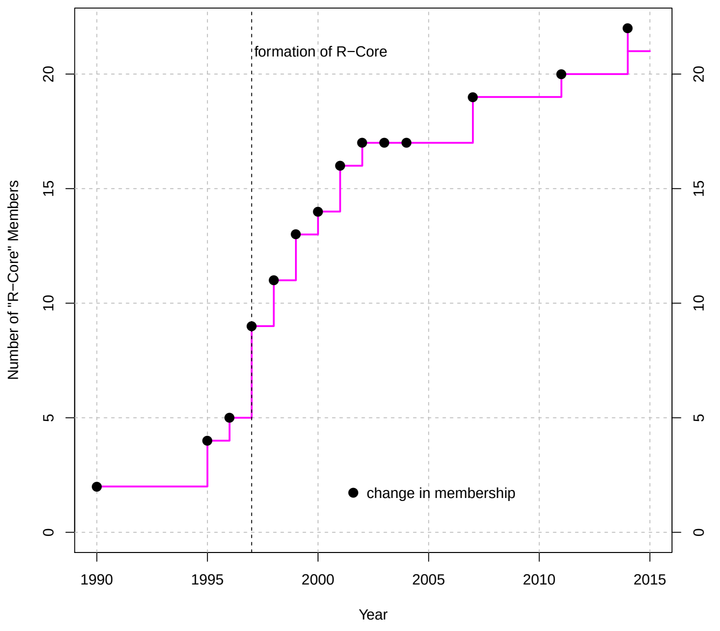


Figure 1: Growth in the R Core Team, founded in 1997, and its informal predecessor since the inception of the R Project in 1990. The dots represent changes in membership: For example, in 2014, one member left and two joined. *Source of data:* Interviews with members of R Core and `https://CRAN.R-project.org/src/base/`.

of Auckland in New Zealand. Ihaka and Gentleman were interested in exploring the design of statistical programming languages and in providing a tool with which they could teach statistics. They settled on a strategy of combining the syntax of S, familiar to them and then in wide use among statisticians, with the semantics of Scheme, a dialect of Lisp.

By the mid-1990s, Ihaka and Gentleman started to attract other developers to their project, eventually formalized in 1997 as the R Core Team. Figure 1, which extends a similar graph reported by Fox (2009), shows the growth in the R Core Team and its informal predecessor. It is apparent that the rapid initial growth in the R Core Team slowed down, and that there has been modest turnover.

Figure 2, tracing the history of "commits" by members of the R Core Team to the **Subversion** (SVN, Pilato, Collins-Sussman, and Fitzpatrick 2004) version-control archive containing the code for R, also extends – and augments – results reported by Fox (2009). The gray line with diamonds and the right-hand axis display the *number* of commits, which initially grew rapidly and then gradually declined. The left-hand axis is scaled in *percent*: The magenta line with triangles shows the percentage of members of R Core who made at least one commit to the SVN archive in a given year; the cyan line with circles shows the Gini coefficient of inequality in commits among members of R Core, expressed as a percentage; and the blue line with squares shows the percentage of commits made by the most active member of R Core in a given year (who is Brian Ripley since 2000, the year after he joined R Core).

Commits are an imperfect measure of the activity of R Core members because different

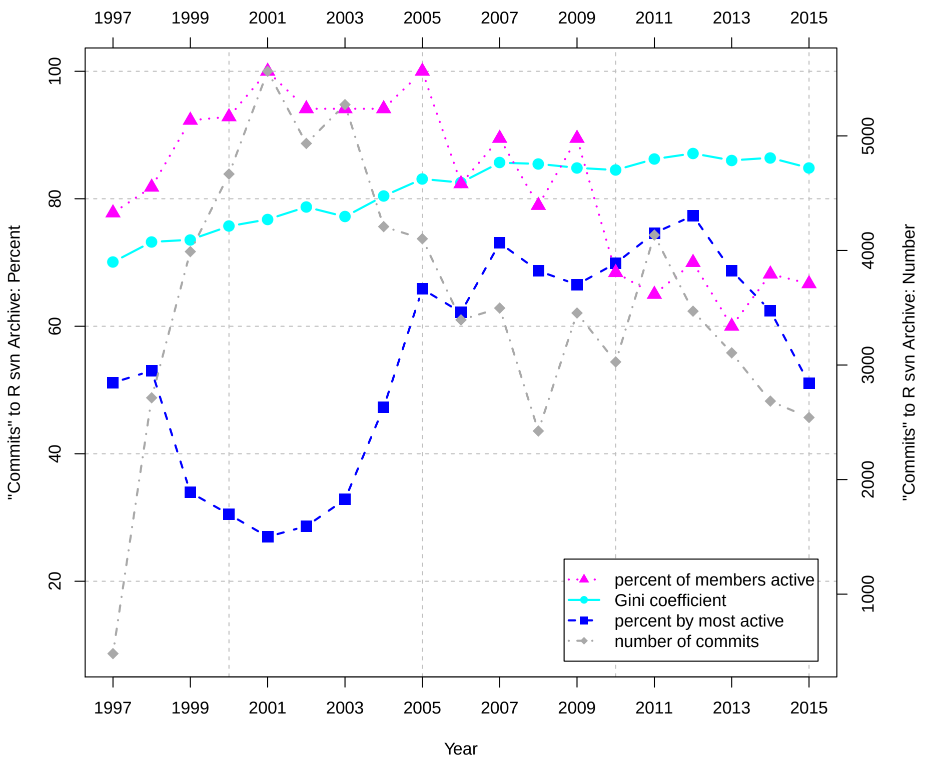
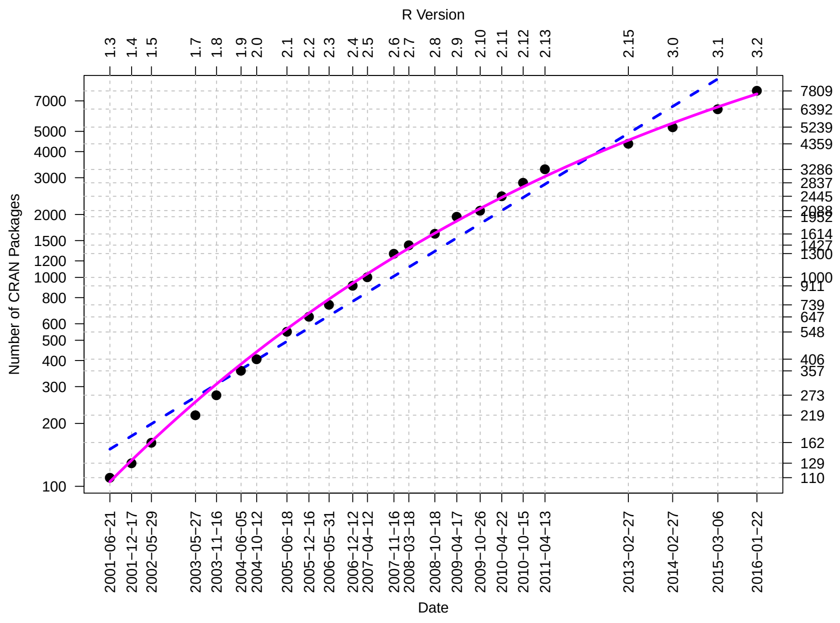


Figure 2: “Commits” to the R SVN code archive by members of the R Core Team, 1997–2015. The gray line with diamonds and the right-hand axis trace the number of commits. The left-hand axis is scaled in percent: The magenta line with triangles traces the percentage of members of R Core who made at least one commit to the SVN archive in a given year; the cyan line with circles traces the Gini coefficient of inequality in commits among members of R Core, expressed as a percentage; the blue line with squares traces the percentage of commits made by the most active member of R Core in a given year. *Source of data:* `https://developer.R-project.org/`

commits can represent very different contributions to the R code base, both in size and in importance, and because members of R Core make contributions to the R Project that are not reflected in the SVN archive, such as managing the periodic release of R software and overseeing the operation of CRAN. Notwithstanding this caveat, it is clear that contributions to the SVN archive remain highly unequal, despite the relative decline in the percentage by its most active member, and a substantial proportion of the membership of R Core is now inactive in the code archive.

As mentioned, in the same year as the advent of the R Core Team, 1997, CRAN was formed to distribute R and contributed R packages on the Internet. The first non-beta version of R, version 1.0.0, appeared on CRAN in February of 2000. As we write this paper 16 years later, the current version of R is 3.2.3, and there are nearly 8000 contributed packages on CRAN. Additional aspects of the history and organization of the R Project for Statistical Computing are discussed by Fox (2009), and a time-line of R milestones appears in the Wikipedia article on R (Wikipedia 2016).

(a)



(b)

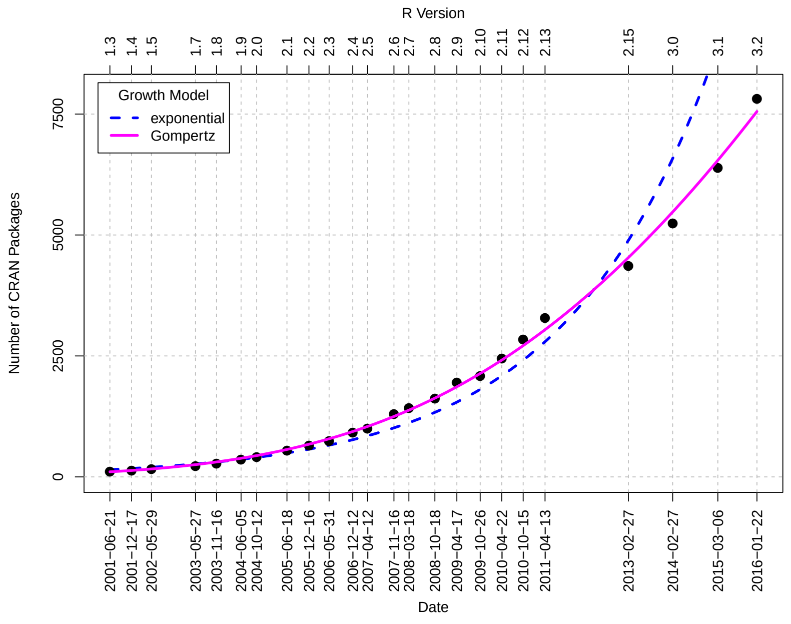


Figure 3: Growth in the number of CRAN packages, 2001–2016. In the upper panel (a), the number of packages is plotted on the log-count scale, so that a linear trend represents exponential growth. In the bottom panel (b), the number of packages is plotted on the count scale. Both panels show the fit of the exponential growth model (blue broken line) and three-parameter Gompertz model (magenta solid line). *Source of data:* `https://svn.R-project.org/R/branches/`.

| *Model* | *AIC* | *BIC* |
|---|---|---|
| 3-Parameter Logistic | 314.07 | 318.79 |
| 4-Parameter Logistic | 302.62 | 308.51 |
| 3-Parameter Gompertz | 300.53 | 305.24 |
| 4-Parameter Gompertz | 301.81 | 307.71 |
| 3-Parameter Weibull | 332.61 | 337.33 |
| 3-Parameter Weibull | 319.64 | 325.53 |

Table 1: AIC and BIC for each of several growth models fit to the CRAN packages data.

Fox (2008, 2009) reported data concerning the growth of the CRAN package archive, which included nearly 2000 packages in 2009. Graphing the number of packages on a log-count scale against time revealed nearly exponential growth in the number of CRAN packages. An updated version of this graph, which appears in the upper panel of Figure 3, shows that the growth rate of CRAN has since slowed down. The lower panel of Figure 3 shows the same data, but with the vertical axis on the count scale.

In addition to the exponential growth model, fit by linear least squares to the log-count data, which, as Figure 3 reveals, fits the data poorly, with the help of the **drc** package (Ritz and Streibig 2005), we fit several other standard simple growth models to the data, including three and four-parameter versions of the logistic, Gompertz, and Weibull models. All of these models were fit by nonlinear least squares to the count data. Table 1 shows the AIC and BIC for each of these models. Both the AIC and BIC favor the three-parameter Gompertz model, but the four-parameter logistic and four-parameter Gompertz models do nearly as well. Moreover, all three of these models provide very similar fits to the data, with the fitted values for the models having inter-correlations exceeding 0.999.

Both panels of Figure 3 graph the fit of the exponential and three-parameter Gompertz growth models. The Gompertz model is

$$\mathit{packages} = \theta_1 \exp\{-\exp\left[\theta_2\left(\mathit{years} - \theta_3\right)\right]\} + \varepsilon \tag{1}$$

where *packages* is the package count; *years* starts at 0 on 2001-01-21, the date of the first available data point; and the errors $\varepsilon$ are assumed to be NID$(0, \sigma^2)$.[1] In this parametrization of the Gompertz model, $\theta_1$ is the asymptotic number of packages.

Estimated parameters (with standard errors in parentheses) are $\widehat{\theta}_1 = 56,299$ (10,321), $\widehat{\theta}_2 = -0.07817$ (0.00511), $\widehat{\theta}_3 = 23.51$ (1.72), and $\widehat{\sigma} = 114.7$. This result therefore implies that CRAN will grow to approximately seven times its current size, eventually reaching 56,000 packages.

Although the Gompertz growth model fits the existing data reasonably well, the asymptotic number of packages $\theta_1$ is imprecisely estimated, and the asymptote is not approached until about 75 years from now. It is of course ridiculous to project software use so far into the future. Even the 10-year projection to 22,452 packages (with a standard error of 1288 packages) is seriously suspect. As the baseball player Yogi Berra reputedly said, "It's tough to make

[1]The assumptions of independent errors with constant variance are potentially problematic for time-series data on counts. An examination of the residuals from the Gompertz growth model suggests, however, that they are only moderately autocorrelated: The first-order residual autocorrelation is $\widehat{\rho}_1 = 0.107$, and the largest absolute residual autocorrelation, at lag 3, is modest, $\widehat{\rho}_3 = -0.406$. The magnitude of the residuals does, however, appear to grow with the magnitude of the fitted values.

predictions, especially about the future." (He also said, "I never said most of the things I said.") Nevertheless, although the growth rate of CRAN is now less than exponential, there appears still to be considerable room for future growth, a possibility that we expect will cause some distress to the CRAN maintainers and to R users who find it increasingly difficult to know what's available on CRAN (see our remarks in Section 4).

## 3. R in the pages of JSS

We examined the 978 contributions to JSS that have appeared between the inaugural volume in 1996 (identified as 1997 on the JSS website, but actually published in 1996) and volume 69 in 2016, including 768 articles, 63 "code snippets", 140 book reviews, and 7 software reviews. In each case, we read the abstract of the contribution to determine what software was used, reading the article itself if this was not clear from the abstract, and recording multiple instances of software if more than one was employed. This strategy runs the risk of missing secondary software not mentioned in an otherwise clear abstract, but we should at least have been able to record accurately the primary software used in each contribution.

Tables 2 and 3 summarize our general findings. Table 3 distinguishes software mentioned 10 or more times, combining less frequently mentioned software in the *other* category. It is clear from Table 2 that the large majority of JSS contributions of each kind – for example, 712/768 = 93% of articles – concern a single kind of software; and it is also apparent from Table 3 that, with the exception of software reviews, most contributions – for example, 577/768 = 75% of articles – in JSS discuss R. Indeed, of the 69 volumes of JSS published to date, 11 (volumes 18, 20, 22, 24, 27, 38, 44, 48, 49, 60, and 63) are special volumes describing statistical software written in R. The only other software to get similar treatment is Lisp-Stat (Tierney 1990), to which volume 13 (Valero-Mora and Udina 2005) was devoted.

The evolution of software coverage in articles published in JSS is depicted in Figure 4. Because only four articles appeared in volume 1, published in 1996, we combined the data for volumes 1 and 2 in the initial bar of each graph; and because at the time we wrote this paper only one volume of JSS, with two articles, was published in 2016, the graphs in Figure 4 are for the period 1996–2015. As it turns out, however, both 2016 articles describe R packages.

The top panel of Figure 4 shows the article counts by year for each of the seven most frequently

| | *Type of contribution* | | | | |
|---|---|---|---|---|---|
| *Multiplicity* | Article | Book review | Code snippet | Software review | *Total* |
| 1 | 712 | 59 | 124 | 5 | 900 |
| 2 | 40 | 3 | 12 | 1 | 56 |
| 3 | 11 | 0 | 2 | 0 | 13 |
| 4 | 3 | 1 | 0 | 1 | 5 |
| 5 | 1 | 0 | 0 | 0 | 1 |
| 9 | 0 | 0 | 2 | 0 | 2 |
| 10 | 1 | 0 | 0 | 0 | 1 |
| *Total* | 768 | 63 | 140 | 7 | 978 |

Table 2: Number of software items covered ("multiplicity") by type of contribution to JSS, 1996–2016: Counts. *Source of data:* `https://www.jstatsoft.org/issue/archive`.

| | *Type of contribution* | | | | |
|---|---|---|---|---|---|
| *Software* | Article | Code snippet | Book review | Software review | *Total* |
| R | 577 | 37 | 96 | 1 | 711 |
| Other | 58 | 3 | 21 | 7 | 89 |
| SAS | 34 | 15 | 14 | 1 | 64 |
| MATLAB | 36 | 3 | 11 | 0 | 50 |
| S/S-PLUS | 23 | 0 | 9 | 0 | 32 |
| Fortran | 26 | 0 | 3 | 0 | 29 |
| C/C++ | 24 | 1 | 2 | 0 | 27 |
| Lisp-Stat | 21 | 0 | 0 | 0 | 21 |
| Stata | 11 | 2 | 4 | 1 | 18 |
| BUGS | 9 | 3 | 4 | 0 | 16 |
| Java | 12 | 2 | 1 | 0 | 15 |
| Spreadsheets | 11 | 0 | 1 | 0 | 12 |
| SPSS | 2 | 1 | 6 | 1 | 10 |
| Python | 8 | 2 | 0 | 0 | 10 |
| *Mentions* | 852 | 69 | 172 | 11 | 1104 |
| *Total contributions* | 768 | 63 | 140 | 7 | 978 |

Table 3: Software mentions (counts) by type of contribution in JSS, 1996–2016.

mentioned kinds of software – software appearing in at least 20 articles – in declining order of overall mention. The number of articles published annually in JSS has increased substantially from the inaugural year of 1996. Since the first article mentioning R appeared in 1999, the number of articles referencing R has also grown dramatically.

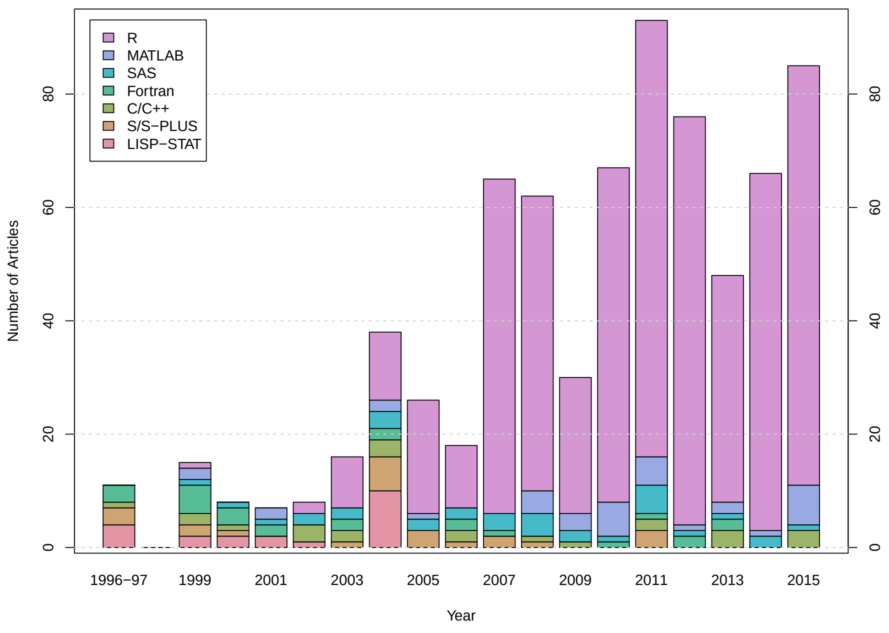
The lower panel of Figure 4 shows the percentage share of mentions by year for each kind of software. It's obvious that R came rapidly to dominate the content of the journal. In contrast, Lisp-Stat, which had a substantial representation in JSS during the first few years of publication disappeared entirely from its pages after 2004.

Although it is of interest to know how the coverage of statistical software in JSS compares to the use of statistical software by statisticians, other "data scientists," and researchers in substantive fields that employ statistical methods, it is very difficult to obtain good estimates of software use. To our knowledge, existing studies of statistical software use are of two kinds: those using objective but indirect indicators of software use (e.g., Muenchen 2014); and those based on surveys conducted with web-based voluntary-response samples (e.g., King and Magoulas 2015; KDnuggets 2015). In our opinion, the latter in particular are nearly worthless for estimating rates of software use in clearly defined populations of programmers and data analysts.

As well, because the mandate of JSS is to support the development of free software (De Leeuw and Mullen 2014), it is natural that largely preprogrammed commercial statistical packages, such as SAS and SPSS, have lower representation in the pages of the journal than free software developed in a free statistical programming language like R. Of course, book and software reviews could be a natural exception.

In light of these considerations, what is perhaps most striking is that only 10 contributions to JSS involved Python (Van Rossum and others 2016), which is apparently approximately as

(a)



(b)

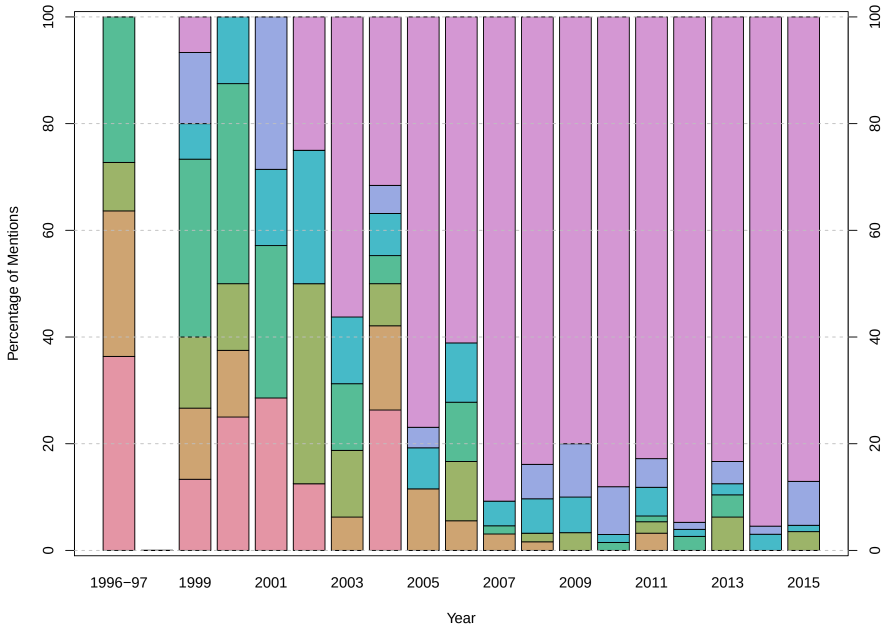


Figure 4: Software used in articles in JSS, by year, 1996–2015. The upper panel (a) shows the *number* of articles published in JSS for each of seven kinds of software; the lower panel (b) shows the *percentage* share for each kind of software. An individual article may employ more than one kind of software (see Table 2).

popular as (or perhaps even more popular than) R among so-called "data scientists" (see the sources cited above). Like R, Python is a free, open-source programming language, and a great deal of software developed in Python is also free. Moreover, while its statistical capabilities – both built-in and provided by contributed packages – are less extensive than R's, Python does have substantial statistical functionality. Perhaps the virtual absence of Python from the pages of JSS reflects cultural differences between statisticians and (other classes of) "data scientists."

## 4. Concluding remarks

In this concluding section, we speculate about the future of R and its relationship to the *Journal of Statistical Software*. The JSS provides a significant outlet for developers of statistical software, currently software primarily written in R. In addition to providing validation for their work – important especially to academics who engage in the development of statistical software – we believe that JSS performs additional useful functions for users as well as developers of R software, not least by furnishing some guidance to the contents of the maze of CRAN packages.

At the same time, the development of software in R, reflected both in the content of JSS and in the CRAN package archive, is continuing at a rapid, if decelerating, rate. Writing nearly seven years ago, Fox (2009) identified several challenges to the continuing development of R. We'll revisit each of these issues in turn:

- Fox (2009) suggested that the highly unequal division of labor, flat non-hierarchial organization, and modified-consensus decision-making of the R Core Team are potential obstacles to moving the R Project forward. The R Core Team has continued to advance the quality and capabilities of standard R software, but at least part of the development of basic functionality in R – for example, advanced tools for interacting with the R interpreter, for package management, for accommodating "big data," and for multi-threading – has moved outside of R Core, some of it to commercial organizations such as RStudio, Google, and Microsoft (through the latter's acquisition of Revolution Analytics, which distributes a proprietary version of R). These and some other corporations, along with the R Foundation, which owns the free software copyright for R and whose members include the R Core Team, have formed the R Consortium, with the goals of "developing, maintaining, and distributing R software" (see `https://www.R-consortium.org/`).

  The implications of these developments for the continued provision of R as free, open-source, cooperatively developed software are not entirely clear, and of course are not necessarily negative: Commercial organizations such as those mentioned have resources not directly available to the R Core Team and the R Foundation, and have the capacity to expand further and to support the user base for R.

- Fox (2009) also identified a potential for conflict between innovation in statistical software and the necessity of maintaining backwards compatibility. As the number of R users grows – especially users who are not also R programmers – and as the number of CRAN packages grows, this challenge intensifies.

  Although there are some exceptions, such as SAS (SAS Institute Inc. 2013) and SPSS (IBM Corporation 2016), most statistical software has a limited lifespan. Few people currently use BMDP or OSIRIS, for example (or, we would wager, know what they are), and the brief ascendancy of Lisp-Stat in the pages of JSS is long past. We anticipate that JSS will continue to track the development of (particularly free) statistical software, whether or not that software is written in R.

To illustrate potential sources of competition, R is widely recognized to be inefficient for certain kinds of tasks, an observation that has motivated improved facilities for incorporating compiled code in R programs, most notably the **Rcpp** package (Eddelbuettel and Francois 2011; Eddelbuettel 2013). The failure adequately to address this and related issues (such as memory management and multi-threading) in a generalized manner opens the door to competing statistical computing environments. One member of the R Core Team, for example, is currently focusing his efforts on the development of Julia (Bezanson, Edelman, Karpinski, and Shah 2015), an R- and MATLAB-like programming environment for scientific – including statistical – computation, built from scratch. Julia has yet to penetrate the pages of JSS, and whether it will be able to overcome R's currently vast advantage in statistical functionality remains, of course, to be seen.

Moreover, to see other programming languages simply as competitors to R is too narrow a perspective. As John Chambers (2014) has explained, both R and its predecessor S were *designed* to interface with other software. For example, R can be used with open-source tools like **Hadoop** (Apache Software Foundation 2016a) and **Spark** (Apache Software Foundation 2016b) for dealing with very large data sets. Indeed, its open design is one of R's strengths.

- Finally, Fox (2009, page 12) suggested that the size and flat organization of the CRAN package archive poses challenges to users, concluding that, "It is hard to imagine that, without further development, the current structure of CRAN and the tools that surround it could usefully survive, say, five more years of exponential growth."

  That prediction has clearly proven incorrect: Although, as we have shown, subsequent growth has been less than exponential, the CRAN package archive is now four times larger than it was in 2009, and it has not yet collapsed under its own weight.

  There are, however, signs of strain, including in the burden on the CRAN maintainers of keeping the archive up-to-date. If CRAN really does grow to several times its current size it is not obvious to us that it can continue to be maintained without organizational change.

There are also competing R package archives, most notably the **Bioconductor** archive (Huber *et al.* 2015), founded in 2001 and focused on R software for bioinformatics, which now includes more than 1000 packages. As well, sophisticated tools have been developed for maintaining and distributing R packages on GitHub (Wickham 2015), offering another possible source of R software, potentially in competition with CRAN. GitHub is also potentially problematic because it permits the convenient distribution of R packages without the quality control and modest curation provided by the more traditional CRAN and **Bioconductor** package archives. Although one *could* regard GitHub primarily as a development platform for R packages eventually destined for CRAN, comparable, for example, to R-Forge (Theußl and Zeileis 2009), we believe (though we have no systematic data) that there is a trend towards releasing packages on GitHub as an alternative to CRAN.

These challenges to its ascendancy notwithstanding, we expect that in the near future R will continue to dominate both the development of statistical software and the representation of this software in the pages of JSS.

## Acknowledgments

The work reported in this paper was supported by a grant to John Fox from the Social Sciences and Humanities Research Council of Canada (SSHRC). We are grateful to the guest editors and to an anonymous reviewer for comments on a draft of the paper.

We would like to join the other contributors to this special volume in thanking Jan de Leeuw for his vital contributions to founding and nurturing the *Journal of Statistical Software*, and to wish him well in his retirement.

## References

Apache Software Foundation (2016a). *Apache **Hadoop***. URL http://hadoop.apache.org/.

Apache Software Foundation (2016b). *Apache **Spark***. URL http://spark.apache.org/.

Bezanson J, Edelman A, Karpinski S, Shah VB (2015). "Julia: A Fresh Approach to Numerical Computing." arXiv:1411.1607 [cs.MS], URL http://arxiv.org/abs/1411.1607.

Chambers JM (2014). "Interfaces, Efficiency, and Big Data." The R User Conference, useR! 2014, URL http://user2014.stat.ucla.edu/files/chambers.pdf.

De Leeuw J, Mullen K (2014). "The Journal of Statistical Software: Past, Present, and Future." The R User Conference, useR! 2014, URL http://gifi.stat.ucla.edu/janspubs/2014/notes/deleeuw_mullen_U_14.pdf.

Eddelbuettel D (2013). *Seamless R and C++ Integration with **Rcpp***. Springer-Verlag, New York.

Eddelbuettel D, Francois R (2011). "**Rcpp**: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(1), 1–18. doi:10.18637/jss.v040.i08.

Fox J (2008). "Editorial." *R News*, **8**(2), 1–2.

Fox J (2009). "Aspects of the Social Organization and Trajectory of the R Project." *The R Journal*, **1**(2), 5–13.

Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, Carvalho BS, Bravo HC, Davis S, Gatto L, Girke T, Gottardo R, Hahne F, Hansen KD, Irizarry RA, Lawrence M, Love MI, MacDonald J, Obenchain V, Olés AK, Pagès H, Reyes A, Shannon P, Smyth GK, Tenenbaum D, Waldron L, Morgan M (2015). "Orchestrating High-Throughput Genomic Analysis with **Bioconductor**." *Nature Methods*, **12**(2), 115–121.

IBM Corporation (2016). *IBM SPSS Statistics 24*. IBM Corporation, Armonk. URL http://www.ibm.com/analytics/us/en/technology/spss/.

Ihaka R (1998). "R: Past and Future History." Unpublished paper, downloaded 2016-02-08 from `https://www.stat.auckland.ac.nz/~ihaka/downloads/Interface98.pdf`.

Ihaka R, Gentleman R (1996). "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics*, **5**(3), 299–314.

KDnuggets (2015). "Primary Programming Language for Analytics, Data Mining, Data Science Tasks: R, Python, or Other." Web page, downloaded 2016-05-09, `http://www.kdnuggets.com/polls/2015/r-vs-python.html`.

King J, Magoulas R (2015). *2015 Data Science Salary Survey*. O'Reilly, Sebastopol. URL `http://www.oreilly.com/data/free/files/2015-data-science-salary-survey.pdf`.

Muenchen RA (2014). "The Popularity of Data Analysis Software." Unpublished paper, downloaded 2016-05-09 from `http://r4stats.com/articles/popularity/`.

Pilato CM, Collins-Sussman B, Fitzpatrick BW (2004). *Version Control with* ***Subversion***. O'Reilly. URL `http://svnbook.red-bean.com/`.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Ritz C, Streibig JC (2005). "Bioassay Analysis using R." *Journal of Statistical Software*, **12**(5), 1–22. `doi:10.18637/jss.v012.i05`.

SAS Institute Inc (2013). *The SAS System, Version 9.4*. SAS Institute Inc., Cary. URL `http://www.sas.com/`.

Theußl S, Zeileis A (2009). "Collaborative Software Development Using R-Forge." *The R Journal*, **1**(1), 9–14.

Tierney L (1990). *Lisp-Stat: An Object-Oriented Environment for Statistical Computing and Dynamic Graphics*. John Wiley & Sons, New York.

Valero-Mora P, Udina F (2005). "The Health of Lisp-Stat." *Journal of Statistical Software*, **13**(10), 1–5. `doi:10.18637/jss.v013.i10`.

Van Rossum G, others (2016). *Python Programming Language*. URL `http://www.python.org/`.

Wickham H (2015). *R Packages: Organize, Test, Document, and Share Your Code*. O'Reilly, Sebastopol.

Wikipedia (2016). "R (Programming Language)." Downloaded 2016-02-08 from `https://en.wikipedia.org/wiki/R_(programming_language)`.

**Affiliation:**

John Fox
Department of Sociology
McMaster University
Hamilton, Ontario, Canada L8S 4M4
E-mail: jfox@mcmaster.ca
URL: http://socserv.socsci.mcmaster.ca/jfox/

Allison Leanage
Department of Sociology
McMaster University
Hamilton, Ontario, Canada L8S 4M4

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
September 2016, Volume 73, Issue 2 *Submitted:* 2016-03-24
doi:10.18637/jss.v073.i02 *Accepted:* 2016-06-01