JOURNAL OF APPLIED ECONOMETRICS
J. Appl. Econ. **27**: 167–172 (2012)
Published online 26 October 2011 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/jae.1278

SOFTWARE REVIEWS

RSTUDIO: A PLATFORM-INDEPENDENT IDE FOR R AND SWEAVE

JEFFREY S. RACINE*
Department of Economics, McMaster University, Hamilton, Ontario, Canada

1. OVERVIEW

One frequently voiced complaint about the R programming environment (R Development Core Team, 2011) is its rather 'clunky' user interface and the fact that multiple interfaces exist and differ across platforms. Even seasoned R users acknowledge that there is room for improvement, and there have been a number of attempts made to create a more user-friendly interface. However, it is widely acknowledged that there remains room for improvement in this area.

Users of R also tend to be users of TeX/LaTeX, and a small but growing subset tend also to be users of Sweave. It is safe to say that interface-related complaints levied against R are also tossed at TeX/LaTeX and Sweave for similar (and often well-justified) reasons.

However, a recent entrant is poised to resolve these issues not only for R but also for this trifecta of programs that tend to be jointly used by econometricians and academics (i.e. R, Sweave, and TeX/LaTeX). The new entrant is titled 'RStudio' (see www.rstudio.org). It is currently in beta status (version 0.94 (RStudio Beta 3) as of this writing) and was publicly released on 28 February 2011. However, it is quite stable and currently suitable for general use. Like R, it is free and open. This review discusses the aspects of RStudio that are likely to be of most interest to econometricians. Naturally, this would also be of interest to academics using R in the classroom, as students would likely be more comfortable with this interface than existing ones, while the platform-independent nature would minimize cross-platform issues encountered by new users.

2. THE RSTUDIO INTEGRATED DEVELOPMENT ENVIRONMENT

RStudio is more than just a pretty (inter)face. It is in fact an integrated development environment (IDE) that has been carefully thought out and that anticipates the needs of R users who may grow beyond simply using R for their research and may wish to graduate to Sweave/TeX/LaTeX out of concerns for reproducibility and who wish to streamline their approach towards knowledge creation

* Correspondence to: Jeffrey S. Racine, Department of Economics, Kenneth Taylor Hall, McMaster University, 1280 Main Street West, Hamilton, Ontario, Canada L8S 4M4. E-mail: racinej@mcmaster.ca

and transfer. It works with the standard version of R available from the Comprehensive R Archive Network (CRAN) and is developed by J. J. Allaire, J. Cheng, J. Paulson, and P. DiCristina. RStudio is also a company, and its developers plan to sell services related to the software (including support, training, consulting, and hosting).

The RStudio IDE incorporates an editor (in this case adapted to the R language), a multiple document interface so that one can have a variety of documents open in different tabs, a window for interacting with R, and a window that serves as a file browser and language reference.

2.1. Installation

RStudio requires that a working installation of R be present on the system (see www.r-project.org), and if the user wishes to use Sweave then a working implementation of $\TeX/\LaTeX$ must also be present (see www.tug.org/texlive). Installation is straightforward and involves simply downloading the appropriate package for one's system and then conducting a standard install. RStudio requires that R version 2.11.0 or higher be installed (the current R version is 2.13.1 as of this writing), and binary distributions exist for Windows, Mac OS X, and Debian 6+ and Fedora 13+ versions of Linux (both 32- and 64-bit), while source tarballs exist for those who wish to compile from scratch.

2.2. The RStudio Interface

The IDE presents the user with four customizable panes: an 'editor' pane (upper left) where one edits/maintains source code, a 'console' pane where code is executed (lower left), a 'workspace/history' pane where objects in one's workspace are listed and summarized and where the command history can be viewed and manipulated (upper right), and a 'files/plots/packages/help' pane where one can browse files, view plots, maintain packages, and access help files (lower right). The interface can be extensively customized, as can the contents and location of each pane.

Figure 1 shows the IDE as it was used to write this article.[1]

2.3. RStudio Server: R in a Browser

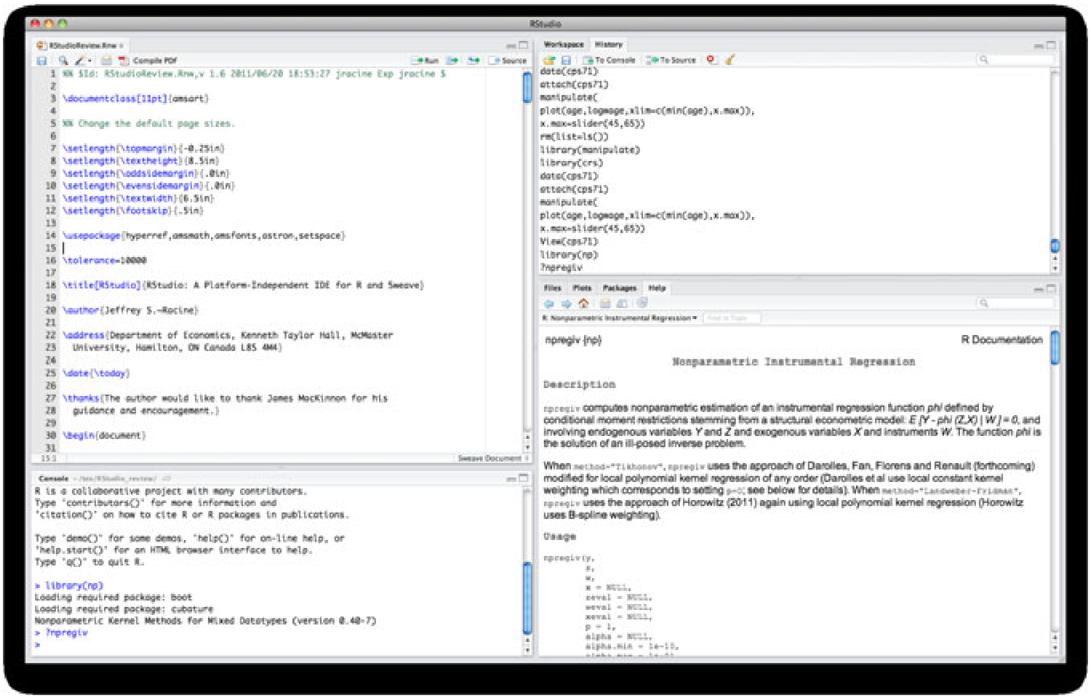
Not only is RStudio platform independent, but there is also a server-based version that can be run in a browser over the web. It may not be immediately apparent, but this presents an exciting opportunity for conducting research on hand-held devices including smart phones and other portable devices that may lack either processing power, easy-to-access file systems, or both. This feature is one that ought to be warmly welcomed by those among us who have yearned for statistical software running on portable devices.

2.4. Sweave/$\TeX/\LaTeX$ Support

Perhaps the most streamlined component of RStudio is the 'one-click PDF' feature for users of Sweave, so a few words on Sweave itself may be in order. The Sweave package for the R and S-plus statistical computing environments enables the user to construct a single file which includes both the code to be run in R/S-plus and the $\TeX/\LaTeX$ code comprising the text of the document. Files containing both types of code are referred to as .Rnw/.Snw files. The various sections

[1] I suspect that the authors added Sweave support for the creation of vignettes which are recommended parts of an R package, but those who use Sweave for their research independently of R will benefit from the 'one-click' generation of the final PDF version of their document.

DOI: 10.1002/jae



```
%% $Id: RStudioReview.Rnw,v 1.6 2011/06/20 18:53:27 jracine Exp jracine $

\documentclass[11pt]{amsart}

%% Change the default page sizes.

\setlength{\topmargin}{-0.25in}
\setlength{\textheight}{8.5in}
\setlength{\oddsidemargin}{.0in}
\setlength{\evensidemargin}{.0in}
\setlength{\textwidth}{6.5in}
\setlength{\footskip}{.5in}

\usepackage{hyperref,amsmath,amsfonts,astron,setspace}

\tolerance=10000

\title[RStudio]{RStudio: A Platform-Independent IDE for R and Sweave}

\author{Jeffrey S.~Racine}

\address{Department of Economics, Kenneth Taylor Hall, McMaster
  University, Hamilton, ON Canada L8S 4M4}

\date{\today}

\thanks{The author would like to thank James MacKinnon for his
  guidance and encouragement.}

\begin{document}
```

Console ~/tex/RStudio_review/

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(np)
Loading required package: boot
Loading required package: cubature
Nonparametric Kernel Methods for Mixed Datatypes (version 0.40-7)
> ?npregiv
>
```

```
data(cps71)
attach(cps71)
manipulate(
plot(age,logwage,xlim=c(min(age),x.max)),
x.max=slider(45,65))
rm(list=ls())
library(manipulate)
library(crs)
data(cps71)
attach(cps71)
manipulate(
plot(age,logwage,xlim=c(min(age),x.max)),
x.max=slider(45,65))
View(cps71)
library(np)
?npregiv
```

npregiv {np} R Documentation

Nonparametric Instrumental Regression

Description

npregiv computes nonparametric estimation of an instrumental regression function *phi* defined by conditional moment restrictions stemming from a structural econometric model: *E [Y - phi (Z,X) | W] = 0*, and involving endogenous variables *Y* and *Z* and exogenous variables *X* and instruments *W*. The function *phi* is the solution of an ill-posed inverse problem.

When method="Tikhonov", npregiv uses the approach of Darolles, Fan, Florens and Renault (forthcoming) modified for local polynomial kernel regression of any order (Darolles et al use local constant kernel weighting which corresponds to setting p=0; see below for details). When method="Landweber-Fridman", npregiv uses the approach of Horowitz (2011) again using local polynomial kernel regression (Horowitz uses B-spline weighting).

Usage

```
npregiv(y,
        z,
        w,
        x = NULL,
        zeval = NULL,
        weval = NULL,
        xeval = NULL,
        p = 1,
        alpha = NULL,
        alpha.min = 1e-10,
```

Figure 1. The RStudio IDE

(‘chunks’) of R/S-plus and TEX/LATEX code are included in the file in the order in which they are to be employed in the final document. Sweave then weaves together the code chunks to produce a .tex file that may be compiled using TEX/LATEX. Using Stangle, you can also strip off just the R code itself from the Sweave document. By using Sweave, an individual can create a dynamic document (Gentleman and Lang, 2004) which includes both the statistical analysis and the methods by which the output underlying the analysis are obtained. This process sidesteps a major source of research errors, namely the misreporting of computer output. For a detailed overview of Sweave, see Meredith and Racine (2009).

3. RSTUDIO HIGHLIGHTS

Much careful planning has gone into RStudio. Below we list some of the features that may be of interest to the reader.

3.1. Tabbed Source Editor

The use of tabs allows users to work with multiple documents in a streamlined setting (think ‘tabs’ as in tabbed web browsing). The source code editor in RStudio is by default tabbed, which allows one to efficiently work with multiple files. This feature is particularly powerful when using, say, Sweave as one can have one tab devoted to creating functioning R code and another to the Sweave document itself, into which the functioning code can be pasted.

The editor also has a number of useful features including syntax highlighting, code completion, multiple-file editing, and search/replace.

DOI: 10.1002/jae

One can directly execute code from within the source editor (the executed commands are inserted into the console where their output also appears) and can execute either a line, sequence of lines, or all code in the editor.

Note that clicking on a data object in the 'workspace/history' pane calls `View()`, which invokes a spreadsheet-style data viewer on a matrix-like `R` object, so that one can readily examine contents (editing of matrix and data.frame objects is not currently supported), while clicking on non-data objects pulls up `fix()`, which invokes `edit` on the object and then assigns the new (edited) version of the object in the user's workspace, which could be a very handy feature for novice users.

One deficiency, however, is that `RStudio` currently lacks support for external editors such as VIM or Emacs. For many users, this could be a drawback, as they would have to learn the idiosyncrasies of yet another editor. This being said, the editor is simple to use and does contain many features one would expect of a modern text editor.

3.2. Searchable History

Often one would like to repeat a set of commands that have been previously run (even perhaps in previous sessions). `RStudio` has a powerful environment for recalling such expressions which is accessible from the 'workplace/history' pane. One can quickly search for previous instances of code and then either send to the console for execution or insert directly into one's source code in the editor.

3.3. Code Transformations

Another powerful feature is the ability to select a series of `R` commands and then extract a function. Writing `R` functions can often be a bit tedious, so this feature definitely can streamline the process. When you find yourself running a set of commands repeatedly, this can help keep your code streamlined and uncluttered and also can help minimize mistakes by allowing you to maintain a function rather than multiple separate sequences of otherwise identical code.

3.4. Interactive Plotting with 'Manipulate'

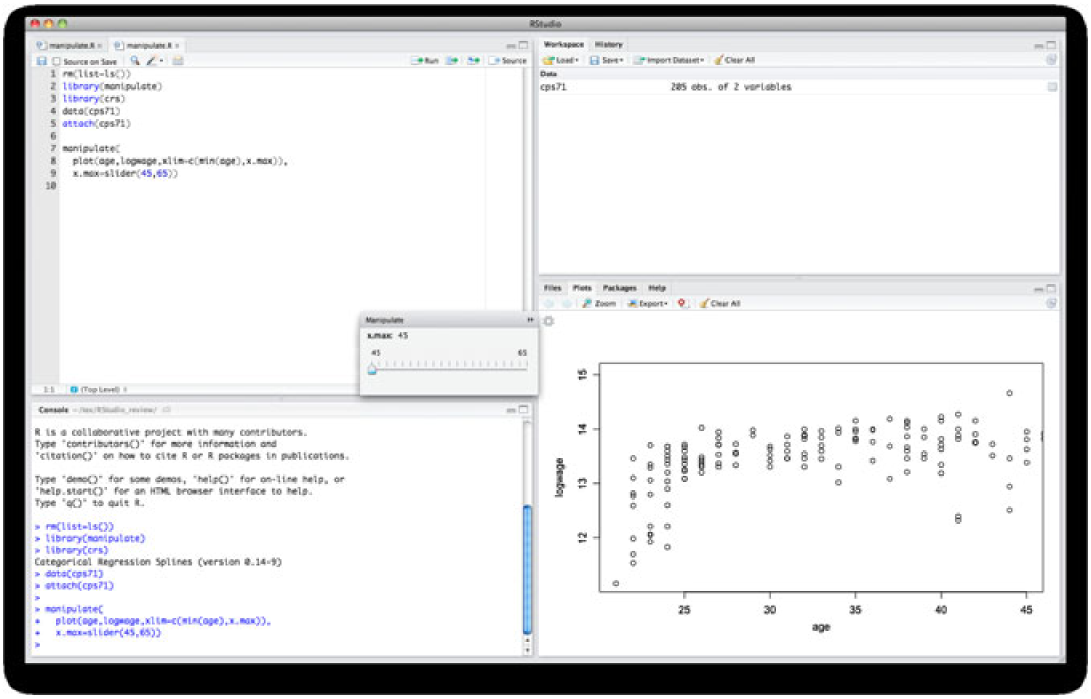
`RStudio` includes a package called 'manipulate' that provides for interactive capabilities when generating standard `R` plots. This is accomplished by binding plot inputs to custom controls rather than static hard-coded values and allows the user to dynamically alter a graph for enhanced understanding of particular features, as the example in Figure 2 demonstrates.

3.5. Other Features of Interest

`RStudio` can be customized in a number of ways in addition to those discussed above (i.e. customizing the panes). For instance, there are a number of themes that can be used to alter the appearance of the program to one's personal tastes.

One can also set a number of general `R` options from among the following:

- Initial working directory—Startup directory for `RStudio`. The initial `.RData` and .Rprofile files (if any) will be read from this directory. The current working directory and Files pane will also be set to this directory. Note that this setting can be overridden when launching `RStudio` using a file association or a terminal with a command line parameter indicating the initial working directory.
- Save workspace to `.RData` on exit—Ask whether to save `.RData` on exit, always save it, or never save it. Note that if the workspace is not dirty (no changes made) at the end of a session, then no prompt to save occurs even if Ask is specified.



```
rm(list=ls())
library(manipulate)
library(crs)
data(cps71)
attach(cps71)

manipulate(
  plot(age,logwage,xlim=c(min(age),x.max)),
  x.max=slider(45,65))
```

```
Data
cps71                    205 obs. of 2 variables
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> rm(list=ls())
> library(manipulate)
> library(crs)
Categorical Regression Splines (version 0.14-9)
> data(cps71)
> attach(cps71)
>
> manipulate(
+   plot(age,logwage,xlim=c(min(age),x.max)),
+   x.max=slider(45,65))
>
```

Figure 2. Enhancements to R - the manipulate package for creating interactive plots

- Restore .RData into workspace at startup—Load the .RData file (if any) found in the initial working directory into the R workspace (global environment) at startup. If you have a very large .RData file, then unchecking this option will improve startup time considerably.
- Always save history (even when not saving .RData)—Make sure that the .Rhistory file is always saved with the commands from your session even if you choose not to save the .RData file when exiting.
- Use single global history (rather than one per working directory)—Always save .Rhistory to the same location rather than into the current working directory of your R session when exiting RStudio.
- Remove duplicate entries in history—Prevent addition of commands to history if they are the exact same as the most recently added command.
- CRAN mirror—Set the CRAN mirror used for installing packages (can be overridden using the repos argument to install.packages).

One very nice and appreciated feature involves the storing of multiple plots that can be recalled simply by using the arrow buttons to go back/forward as one would when recalling previous pages in a web browser. Also, all plots appear in the plot pane so the plot device does not get overwritten.

Another welcome feature is the ability to export graphs in a variety of formats directly rather than having to write the R code to do so.

4. SUMMARY

RStudio provides an integrated development environment for users of R and Sweave that is user friendly and platform independent, and has nicely eliminated one of the sources of irritation associated

DOI: 10.1002/jae

with using R for research and for teaching, namely the lack of a powerful and platform-independent development environment for using R. The developers of RStudio are to be commended for their vision and contributions to the open source community.

ACKNOWLEDGEMENT

The author would like to thank James MacKinnon for his guidance and encouragement.

REFERENCES

Gentleman R, Lang DT. 2004. Statistical analyses and reproducible research. Bioconductor Project Working Papers paper 2, Bioconductor Project.

Meredith E, Racine J. 2009. Towards reproducible econometric research: the Sweave framework. *Journal of Applied Econometrics* **24**: 366–374.

R Development Core Team. 2011. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing: Vienna.

DOI: 10.1002/jae