Rweb: Web-based Statistical Analysis

Jeff Banfield

Department of Mathematical Science

Montana State University
Bozeman, MT 59717

Abstract

Rweb is a freely accessible statistical analysis environment that is delivered through the World Wide Web (WWW). It is based on R, a well known statistical analysis package. The only requirement to run the basic Rweb interface is a WWW browser that supports forms. If you want graphical output you must, of course, have a browser that supports graphics. The interface provides access to WWW accessible data sets, so you may run Rweb on your own data. Rweb can provide a four window statistical computing environment (code input, text output, graphical output, and error information) through browsers that support Javascript. There is also a set of point and click modules under development for use in introductory statistics courses.

1   Introduction

The World Wide Web is changing what students expect from us and the way we teach. Many courses, or parts of courses, are now available on line. Even students in traditional lecture courses are asking for assignments and course information to be made available through the Web. Rweb, like other Web-based statistical packages such as WebStat (West and Ogden, 1997), was developed to provide easy to use statistical analysis over the Web. Rweb can be used as a complete computing environment for advanced courses, a simple point and click interface for introductory courses, or it can be incorporated as an interactive component in online assignments for any course.

Ihaka and Gentleman (1996) have given the world, R, an excellent, public domain, statistical analysis package that is source code compatible with S (and Splus). R is available as free software under the Free Software Foundation's GNU general public license. Rweb is a set of Web pages and Perl scripts that provide a Web based interface to R and is freely available from the author.

2   Using Rweb

Rweb is available at `http://www.math.montana.edu/Rweb/`. This URL takes you to a Web page (Figure 1) that explains a little about R and Rweb, provides links to additional information about R, an online set of notes about programming in R, and allows you to choose which version of Rweb you would like to use:
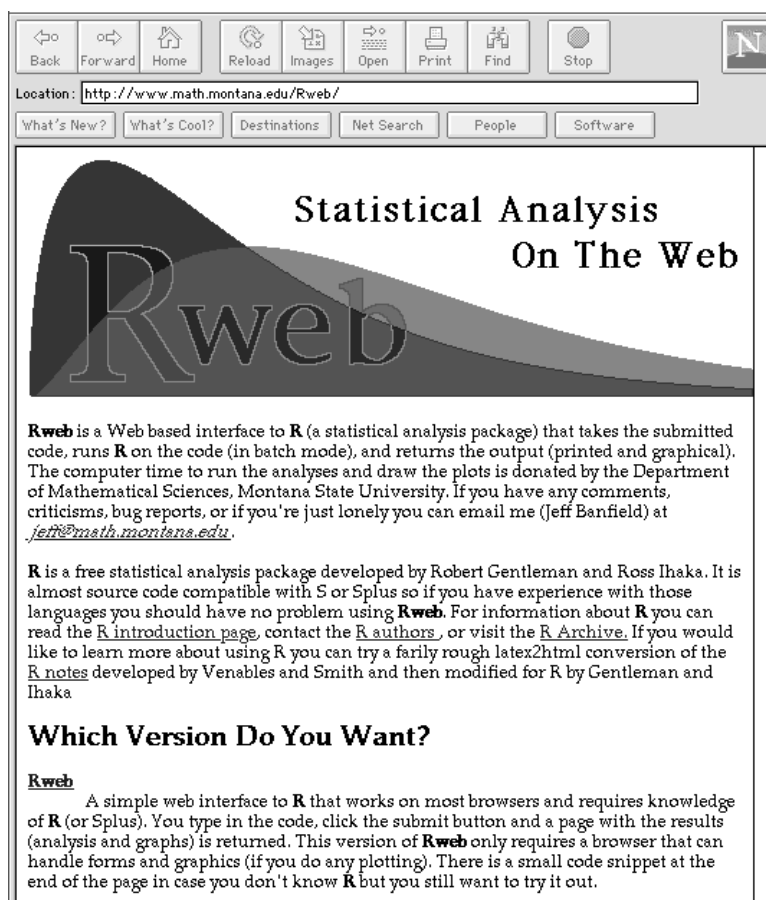

Figure 1: The main Rweb page with links to the three versions of Rweb.

- The basic Rweb that will run on most browsers. This version requires that you know how to program in R (although a small snippet of code

is provided at the end of the web page for you to try out if you don't know R).

- The Javascript version of Rweb. This version requires a Javascript capable browser and provides a more sophisticated interface.

- Rweb modules. These are point and click interfaces that are under development for use in introductory statistics courses.

All of the Rweb versions allow you to supply a URL for a Web accessible data set. The dataset will be read into a Rweb data frame using the `read.table` function and the option `headers = T`. This means the data must be in text format where each row is an observation and each column is a variable. The `header = T` option assumes the first line of the file contains the variable names.

Here is a small example data set:

```
Age   BloodPressure   Height   Gender
32    191             150      Female
54    205             176      Male
```

Note that white space is used to separate entries so the variable name Blood Pressure must be written as one word, BloodPressure. More information on the `read.table` function (and the R language in general) is available at one of the R archives listed in `http://lib.stat.cmu.edu/R/CRAN/`. There is an excellent set of notes on using R available in the R archives. The main Rweb page provides links to the R archive and an online version of the notes.

2.1   Basic Rweb

The basic Rweb link provides access to a web page (Figure 2) that is compatible with any browser that can handle forms and graphics. The page provides links to an online S tutorial and an online version of the R help pages. There is also a small snippet of R code at the end of the Web page so someone not familiar with R can at least try it out. The main features of the basic Rweb page are:

Figure 2: The basic Rweb page. The user types code into the text window then clicks on the submit button to have R execute the code.

- The code window. A text window for typing in R code. You can cut and paste code into this window. You can also edit the code in the window. The exact features, and how they are implemented depend on the browser and platform you are using.

- Submit button. Once you have the code in the text window click on the submit button to execute it.

- Erase Everything button. This clears everything in the text window.

- Dataset URL text field. This is the text field where you can type in a URL for a Web accessible data set. If you type in a URL, the dataset it references will be used in the evaluation of the code in the code window.

After you click on the submit button your code is sent to a Perl script that runs R, in batch mode, on your code and then formats and returns the

R output (both text and graphics) as a new Web page (Figure 3). Section 3 of this paper provides more details about the inner workings of Rweb.
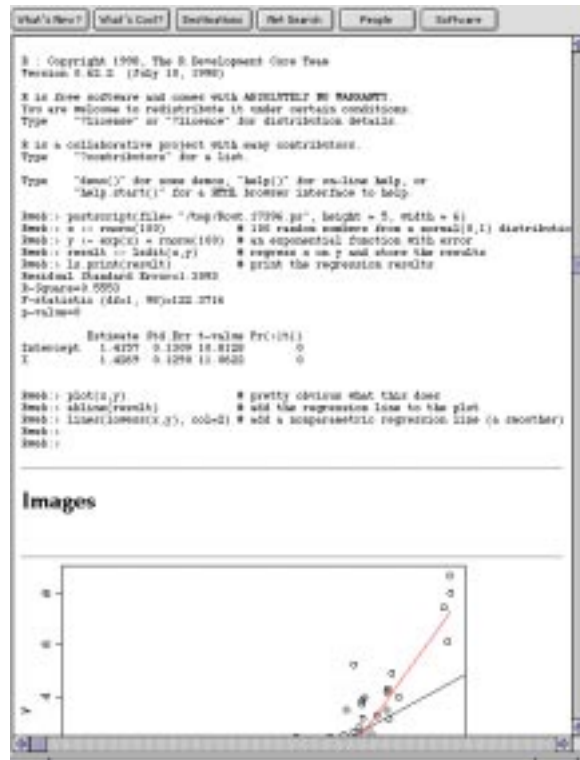

Figure 3: A typical output page created by running basic Rweb.

The entire R output, including error messages, is returned so you can see any printed output as well as where things went wrong if you have an error. To correct any errors, or to modify your code, you can use the back button on your Web browser, edit the code in the code window, and resubmit. You will get a new output page.

If you use the Dataset URL option the Perl script will check to make sure the URL references a valid file. If the URL does not reference a valid file the returned page will contain an error message. This check is only for the validity of the URL, it does not check to see if the data are appropriate for entry into R. If something goes wrong reading the data into R you will get an error message in the R output that may be helpful in identifying the

problem. Rweb will automatically insert the code necessary to read in the data.

2.2 Javascript Version of Rweb

The Javascript version of Rweb provides a more sophisticated interface to R but requires a Web browser that understands Javascript. The initial Javascript Web page (Figure 4) contains a brief description of the Javascript version of Rweb, a link to a help page on R functions, several code snippets, and a button to start up the Javascript version of Rweb. By clicking on this button you will create a new browser window where you can type, edit, or paste R code. This code window will remain open until you close it.

Figure 4: The page you use to start the Javascript version of Rweb.

The code window has a text area for entering and editing R code, a submit button, a button to clear the text area, and a text field for entering

the URL of an external data set. Once you have entered the code you want
to run and clicked on the submit button, several new windows will pop up
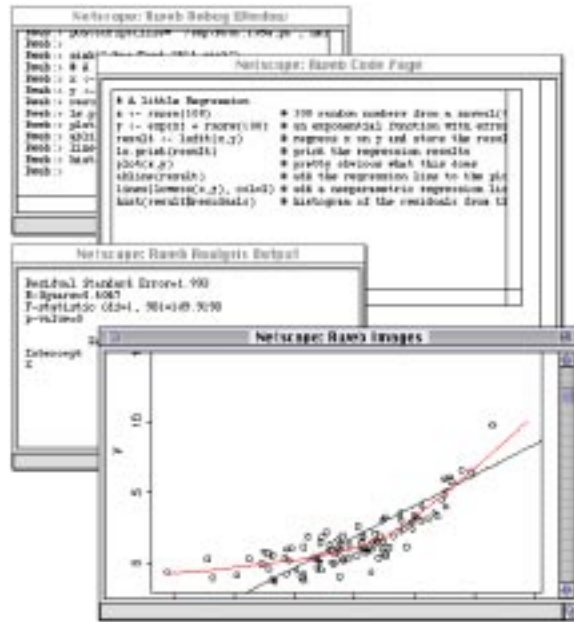on your screen (Figure 5).

Figure 5: The four windows of the Javascript version of Rweb. From front
to back: Graphs, Text output, Code, Debug.

- Text output. The text output window will contain anything printed
  from Rweb. For example the text output from a regression analysis
  or an ANOVA table will be displayed in the text output window. The
  contents of this window come from the R `sink` function.

- Debugging information. The debugging window displays the usual
  R output that includes the code you submitted, the code the Perl
  script inserts for graphics output and external dataset entry, and error
  messages. If something goes wrong with the R execution the debugging
  window will show you the place in the code where the error occurred.
  Normally, depending on your programming skills and typing ability,
  you won't use the debugging window on a regular basis so you can
  keep it iconified until you need it.

- Graphs. The graphics window will display all of the graphics output. This window will only appear if you have created graphical output. Once the window is created it remains on the screen until you close it.

One way to use the four window display is to have the code, text, and graphics windows open and the debugging window iconified (to save screen space). You can enter and edit the R code in the code window, submit it, and watch the updates in the text and graphics windows. All three windows will remain on the screen and they can be arranged so all three are visible at the same time (unless you have a really small monitor).

2.3 Rweb Modules

The Rweb modules (Figures 6 and 7) are a series of point and click interfaces for standard statistical analyses. Currently there are modules for:

Figure 6: The Rweb modules page. The user may choose the type of analysis using the drop down Analysis Menu and a data set (or external data entry option) with the Data Set Menu.
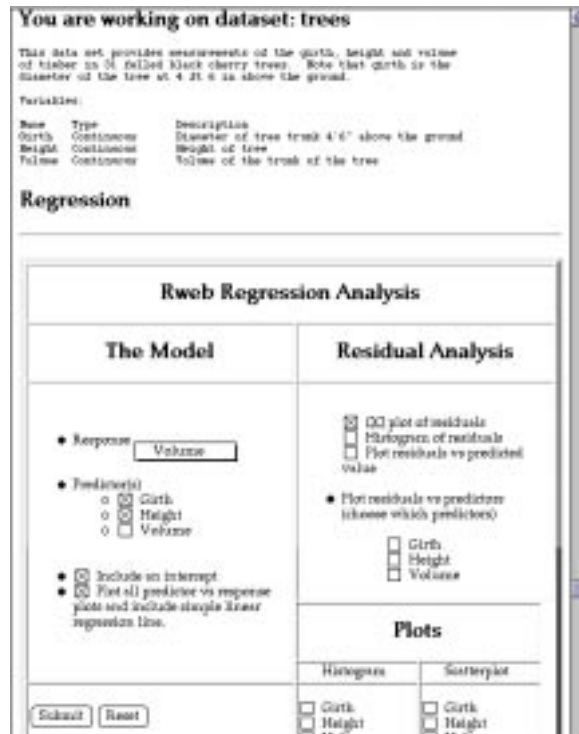
Figure 7: The point an click interface for the Rweb regression module.

- Summary statistics and graphics

- Linear Regression (simple and multiple)

- ANOVA (one and two-way)

- A probability calculator (P-values for Normal, T, Binomial, and Chi Square)

- Two Way Tables

The Rweb modules are a work in progress, both in adding new modules and improving the ones that are currently implemented. They are designed for use in introductory statistics courses. There are several built in datasets that students may select. They also have the option of entering a URL for

9

an external dataset. The advantage of the Rweb modules over classic statistical packages is that the students only have to learn (and many already know) how to operate a Web browser. The author's experience using the Rweb modules in introductory statistics courses at Montana State University is that they have virtually eliminated student problems and complaints about the computer aspect of assignments. Students still complain about assignments, just not about using the computer.

2.4   Embedding Rweb in Online Documents

The previous discussion centered around using the main Rweb page available from Montana State University. The Rweb shell scripts, however, can be incorporated into any online document, not just the ones from MSU. As an example visit the demonstration regression assignment page at `http://www.math.montana.edu/Rweb/Demo/Regression.html` This page illustrates how to call the Rweb Perl scripts at various places within a Web page. You can copy the source for this page to your computer, modify it for your purposes, and then make it available to your students.

3   Behind the Scenes

The Rweb interface to R is a collection of cgi scripts written in Perl (Herman, E., 1997; Wall, L., et. al., 1996) The main Rweb page (where you choose the version of Rweb that you want to use), the Basic Rweb page, and the introduction to Javascript Rweb page are the only pages that physically exist as HTML documents. All other Rweb pages are created "on the fly" by the cgi scripts. The Readme file that comes with the Rweb distribution contains detailed descriptions of how the scripts call each other, call other software, pass the code to R, and return the output to the user. This section will provide a general overview of how Rweb operates. If you would like to setup your own Rweb server the Rweb distribtution is available at:

   `http://www.math.montana.edu/Rweb/Resources.html`

3.1   Background

Web browsers are what most people use to surf the Web. Netscape and Internet Explorer are the two most common browsers. Web servers are the software that actually makes the Web work by sending the Web browsers what they ask for. Rweb is currently delivered by an Apache Web server

running on a Sun workstation. Rweb does not use any of the special features of the Apache server or the Sun system, however it does make heavy use of the UNIX operating system. With minor modifications, Rweb should work on any UNIX based Web server.

3.2 The Transient Nature of Rweb

All of the R program files, output files, and conversion files are stored in the `/tmp` directory and removed when the Rweb script ends (except for image files which cannot be removed until they have been sent to the browser). Since each call to Rweb starts a new process, the process ID number can be used to track and remove all of the temporary files.

Because of the transient nature of these files once the user has received the R output and graphs there is nothing left on the server from their call. All files have been removed so each submission to Rweb begins from scratch, even if the user only makes a slight modification to the code.

3.3 Running R

Some of the Rweb scripts (such as the call for the Rweb Modules or the Javascript version of Rweb) just build new Web pages that are presented to the user. Most of the scripts, however, build R programs based on the user input.

There are three basic files created by each call to Rweb:

- R code ... this is the file that will be used to run R in batch mode. Depending on what the user has requested there are several lines of header code that are inserted before the user code. For example a call to the `postscript` function defining a graphics output file or a `read.table` call to bring in a user specified data set.

- R text output ... this file is created by redirecting the output from the batch call to R into a temporary file. The R text output file has the code, errors, and printed output.

- Postscript file containing all of the graphical output from R.

There are also a few other files created depending on what the user has requested. If a Web accessible data set is requested it is copied into a temporary file and loaded by the R header code. The Javascript version of

Rweb uses the `sink` function to create a file for the printed output. This file is later directed to the Rweb text output window.

3.4 Dealing with Graphs

Returning the graphical output from R to the user is the most complex part of Rweb. Every batch call to R has the graphics output stored in a Postscript file. Since most Web browsers don't display Postscript, this file is converted to a collection of temporary GIF files using the `pstoimg` function that comes with the `latex2html` conversion program.

As with all of the temporary files, the GIF files created by `pstoimg` are stored in the `/tmp` directory. Unfortunately, these files cannot be removed when the script exits since the only information that is returned to the user's browser is a link to the image and it is not known when the images are actually transmitted to the user's browser. To get around this problem Rweb returns all images links as calls to a non-parsed header cgi script that formats and returns the image then removes the image file.

4   Related Work

There are several other Web based programs for statistical analysis based on Java, server side includes, and other forms based access to statistical programs. A partial listing includes:

- Server-side Statistics Scripting in PHP
  `http://www.stat.ucla.edu/journals/jss/v02/i01/`

- Statistical Analysis with Webstat, a Java applet for the World Wide Web `http://www.stat.ucla.edu/journals/jss/v02/i03/`

- The UCLA archive
  `http://www.stat.ucla.edu/calculators/`

- The Globally Accessible Statistical Procedures initiative
  `http://www.stat.sc.edu/rsrch/gasp/`

There are several excellent programs in the above list, but none of them provide a complete statistical package. Since Rweb is based on R, a fully developed statistical package that has been tested for years throughout the world, it provides complete statistical analysis capabilities including:

- A programming environment: conditional operators, control statements, looping, assignment, etc.

- Data manipulation.

- Linear algebra operations.

- Statistical operators: summaries, linear model functions, probability distributions, etc.

- A complete suite of graphics capabilities.

- A library of built in datasets.

- A large colletion of specialized statistical functions.

There are disadvantages to Rweb as well:

- Developing new point and click modules requires knowledge of HTML and Perl.

- All of the analysis is run on a single machine. This has the potential of being slow when a lot of people are trying to use Rweb at the same time. So far, after numerous Rweb assignments involving several hundred students, this has not been a problem.

- Rweb is designed to run as a transient program. That is, each time Rweb is asked to execute R code it starts up a new R process, runs the code, returns the results, removes all files that were created, and then exits. It is not possible to store files or data on the server. However, using cut and paste operations all code and output can be stored on the users computer.

5 Summary

Rweb was developed in response to student and instructor frustration with student lab support for specialized statistical packages (such as Minitab or JMP). Experience has shown that all software has problems; they can range from not being able to find built in data sets to not being able to print graphics. These difficulties cause frustration and anger, especially for novice users. These problems can, in part, be overcome by help in

the student labs but, unfortunately, lab assistants are rarely knowledgeable about specialized packages such as statistical analysis software. When course material is delivered through the Web students only need to use a Web browser. If a student does not know how to use a Web browser the assistants (and in fact most of the other students in the lab) will be able to help. Rweb has been used by several hundred students in numerous introductory statistics courses with, so far, no reported problems in the student labs.

Rweb provides complete statistical analysis capability delivered through an easy to use Web interface that is compatible with almost all Web browsers. The interface can be as sophisticated as a complete computing environment or or as simple as a point and click program. The Rweb scripts can easily be incorporated into online documents with no special programming skills other than the ability to write a Web page and knowledge of Splus or R. Because Rweb is delivered through a Web browser it eliminates many of the problems that students face using computers for course assignments. Rweb output can be printed or easily included in word processor documents using cut and paste operations.

The author is currently working on developing new point and click interfaces for introductory statistics classes and online tutorials and lectures that allow the students to use Rweb for exploring statistical concepts. Anyone interested in setting up an Rweb server for their own use (thus reducing the number of people using the server at Montana State University) may download the entire Rweb distribution (including Web pages, cgi scripts, modules, datasets, and installation instructions) from:

`http://www.math.montana.edu/Rweb/Resources.html`

6   References

6.1   Publications

Herrmann, E.., (1997) *Teach Yourself CGI Programming with Perl 5 in a Week,* 2nd *Edition,* Sams.net Publishing, Indianapolis, Indiana.

Ihaka, R. and Gentleman, R., (1996) "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics,* Vol. 5, No. 3, pp. 299-314.

Wall, L., Christiansen, T. and Schwartz, R., (1996) *Programming Perl,* 2nd *Edition,* O'Reilly & Associates, Sebastopol, California.

West, R.W.and Ogden , R.T. (1997) "Statistical Analysis with Webstat, a

Java applet for the World Wide Web," *Journal of Statistical Software,* 2:3.

6.2 Web Addresses

Archive for R:
>   http://lib.stat.cmu.edu/R/CRAN/.

The Globally Accessible Statistical Procedures initiative:
>   http://www.stat.sc.edu/rsrch/gasp/

Rweb:
>   http://www.math.montana.edu/Rweb/

Rweb Regression Demo:
>   http://www.math.montana.edu/Rweb/Demo/Regression.html

Rweb Source Code:
>   http://www.math.montana.edu/Rweb/Resources.html

Server-side Statistics Scripting in PHP:
>   http://www.stat.ucla.edu/journals/jss/v02/i01/

The UCLA archive:
>   http://www.stat.ucla.edu/calculators/

Webstat:
>   http://www.stat.ucla.edu/journals/jss/v02/i03/