Sumo: An Authenticating Web Application with an Embedded R Session

by Timothy T. Bergsma and Michael S. Smith

Abstract Sumo is a web application intended as a template for developers. It is distributed as a Java 'war' file that deploys automatically when placed in a Servlet container's 'webapps' directory. If a user supplies proper credentials, Sumo creates a session-specific Secure Shell connection to the host and a user-specific R session over that connection. Developers may write dynamic server pages that make use of the persistent R session and user-specific file space. The supplied example plots a data set conditional on preferences indicated by the user; it also displays some static text. A companion server page allows the user to interact directly with the R session. Sumo's novel feature set complements previous efforts to supply R functionality over the internet.

Despite longstanding interest in delivering R functionality over the internet (Hornik, 2011), embedding R in a web application can still be technically challenging. Anonymity of internet users is a compounding problem, complicating the assignment of persistent sessions and file space. Sumo uses Java servlet technology to minimize administrative burden and requires authentication to allow unambiguous assignment of resources. Sumo has few dependencies and may be readily adapted to create other applications.

Deployment

Use of Sumo requires an internet server with the following:

- users with passworded accounts (Sumo defers to the server for all authentication)
- a system-wide installation of R (http://cran.r-project.org)
- a running instance of a Java servlet container, e.g. Tomcat 6 (http://tomcat.apache.org)
- Secure Shell[1], e.g. OpenSSH (http://www.openssh.com) or freeSSHd (http://www.freesshd.com) for Windows.

Tomcat, the reference implementation for the Java Servlet and Java Server Pages (JSP) specifications, is widely available and surprisingly easy to install. While frequently run behind Apache (http://www.apache.org), it can function as a stand-alone web server. In our experience, it requires mere minutes to install under Linux (Ubuntu) or Mac OS X, and a little longer to install for Windows. Tomcat is open source.

Sumo is maintained at http://sumo.googlecode.com and is distributed under the GPL v3 open source license as a Java 'war' file. The current distribution, considered complete, is 'sumo-007.war' as of this writing. It can be downloaded, optionally renamed (e.g. 'sumo.war'), and saved in Tomcat's 'webapps' directory. Tomcat discovers, installs, and launches the web application. By default, Tomcat listens on port 8080. The application can be accessed with a web browser, e.g. at http://localhost:8080/sumo.

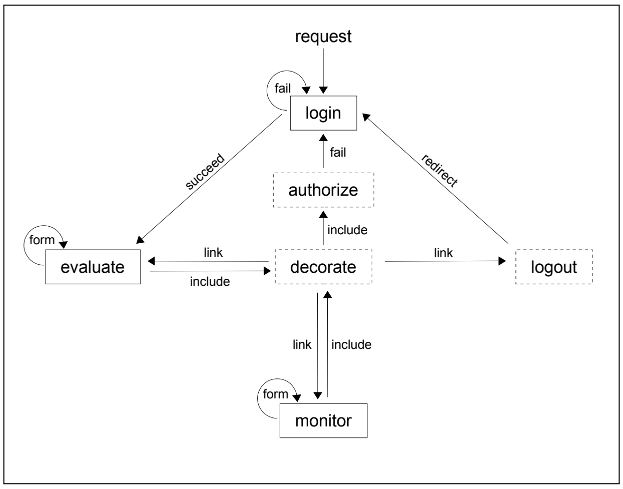


Figure 1: Sumo architecture: flow control among page requests. Solid boxes: visible JSPs. Dashed boxes: JSPs with no independent content.

Architecture

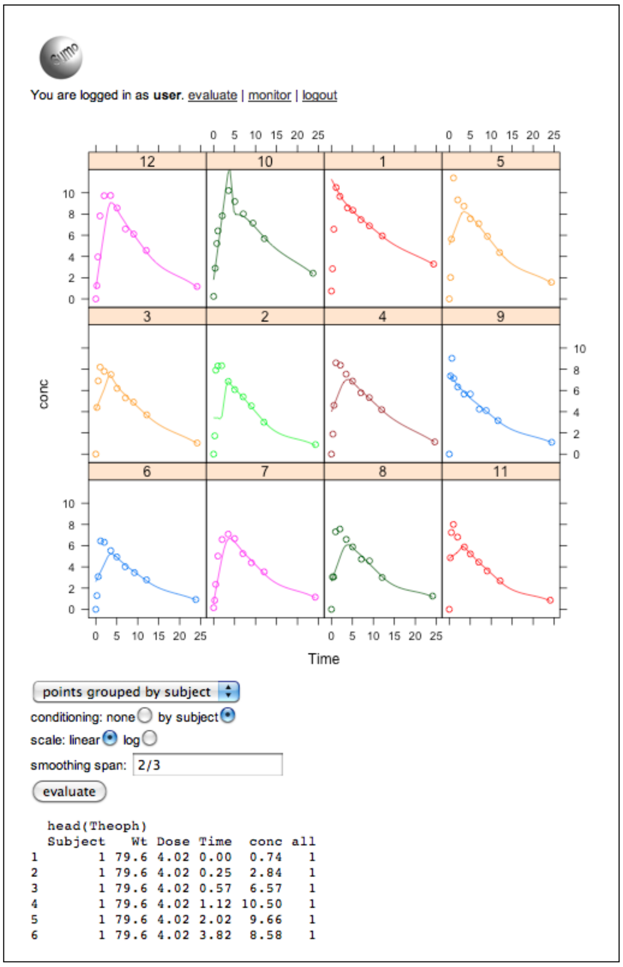
An http request for the web application Sumo defaults to the 'login.jsp' page (Figure 1). Tomcat creates a Java servlet session for the particular connection. 'login.jsp' forwards the supplied credentials to a servlet (not shown) that tries to connect to the local host[2] using Secure Shell. If successful, it immediately opens an R session using `R --vanilla`[3], stores pointers to the R session, and forwards the user to 'evaluate.jsp': the core of the application (see Figure 2). The only other viewable page is 'monitor.jsp',

[1] OS X includes OpenSSH. Check the box: System Preferences | Sharing | remote login. In '/etc/sshd_config', add "PasswordAuthentication yes". With freeSSHd for Windows, uncheck: settings | SSH | Use new console engine.

[2] Host is configurable in the file 'sumo.xml'.

[3] Command to start R is configurable in the file 'sumo.xml'.

which allows direct interaction with the R command line (Figure 3).



You are logged in as **user**. evaluate | monitor | logout

points grouped by subject

conditioning: none ○ by subject ●

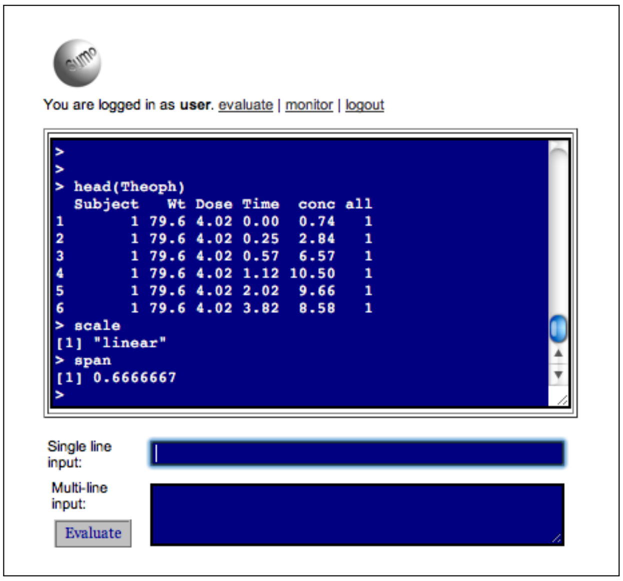
scale: linear ● log ○

smoothing span: 2/3

evaluate

```
  head(Theoph)
  Subject    Wt Dose Time  conc all
1       1 79.6 4.02 0.00  0.74   1
2       1 79.6 4.02 0.25  2.84   1
3       1 79.6 4.02 0.57  6.57   1
4       1 79.6 4.02 1.12 10.50   1
5       1 79.6 4.02 2.02  9.66   1
6       1 79.6 4.02 3.82  8.58   1
```

Figure 2: Sumo display: sample output of ‘evaluate.jsp’.



You are logged in as **user**. evaluate | monitor | logout

```
>
>
> head(Theoph)
  Subject    Wt Dose Time  conc all
1       1 79.6 4.02 0.00  0.74   1
2       1 79.6 4.02 0.25  2.84   1
3       1 79.6 4.02 0.57  6.57   1
4       1 79.6 4.02 1.12 10.50   1
5       1 79.6 4.02 2.02  9.66   1
6       1 79.6 4.02 3.82  8.58   1
> scale
[1] "linear"
> span
[1] 0.6666667
>
```

Figure 3: Sumo display: sample output of ‘monitor.jsp’.

Both ‘evaluate.jsp’ and ‘monitor.jsp’ use optional Sitemesh technology (http://www.sitemesh.org) to include ‘decorate.jsp’, which itself includes the header, the footer, and navigation links to ‘evaluate.jsp’, ‘monitor.jsp’, and ‘logout.jsp’. ‘decorate.jsp’ also includes ‘authorize.jsp’, which redirects to ‘login.jsp’ for non-authenticated requests. Thus decorated, ‘evaluate.jsp’ and ‘monitor.jsp’ always include links to each other and are only available to authenticated users.

The key to embedding R in Sumo is the exchange of information between the Java session and the R session. The exchange occurs primarily in ‘evaluate.jsp’. Similar to Sweave (Leisch, 2002), Java Server Pages support alternation between two languages: in this case, html and Java. The R session can be accessed using either language. With respect to Java, ‘evaluate.jsp’ defines a reference to an object called R that evaluates R code and returns a string result (with or without echo).

```
<%
 String directory = R.evaluate(
  "writeLines(getwd())",false);
%>
```

With respect to html, ‘evaluate.jsp’ includes Sumo’s RTags library (a JSP Tag Library), which allows blocks of R code to be inserted directly within html-like tags.

```
<%@
 taglib prefix="r" uri="/WEB-INF/RTags.tld"
%>
<r:R silent="true">
 library(lattice)
 Theoph$all <- 1
</r:R>

<pre>
 <r:R>
  head(Theoph)
 </r:R>
</pre>
```

Also included in ‘evaluate.jsp’ is ‘params2r.jsp’, which assures that parameters supplied by the user are automatically defined in the R session as either numeric (if possible) or character objects.

Note that ‘evaluate.jsp’ has a single web form (Figure 2) whose target is ‘evaluate.jsp’ itself. Interaction with a user consists of repeated calls to the page, with possibly modified parameters. The example given illustrates table and figure display, emphasizing a variety of input styles (pull-down, radio button, text box). The data frame Theoph is plottable eight different ways (grouped or ungrouped, conditioned or unconditioned, log or linear) with an adjustable smoothing span. Internally, ‘evaluate.jsp’ follows a well-defined sequence:

1. Capture any parameters specified by the user.

2. Supply defaults for parameters not user-specified.

3. Echo parameters to the R session as necessary.

4. Create and display a figure.

5. Present a form with current parameters as the defaults.

Additionally, static text is included to illustrate a very simple way of presenting tabular data.

Development

Sumo is intended as a template for web application developers. There are at least three ways to adapt Sumo. For experimental purposes, one can edit deployed JSPs in place (see 'webapps/sumo/'); Tomcat notices the changes and recompiles corresponding servlets. For better source control, one can unzip 'sumo.war'[4], edit files of interest, re-zip and redeploy. For a formal development environment, one can install Apache Ant (http://ant.apache.org) and check out Sumo's source:

```
svn checkout
http://sumo.googlecode.com/svn/trunk/ sumo
```

Then at the command prompt in the 'sumo' directory[5] use:

```
ant -f build.xml
```

Security

Sumo inherits most of its security attributes from the infrastructure on which it builds. Since the R session is created with a Secure Shell connection, it has exactly the file access configured at the machine level for the particular user. The JSPs and related servlets, which are not user-modifiable, verify the user before proceeding. 'SpecificImageServlet' (called from 'evaluate.jsp') can be used to request arbitrary files, but it defers to the R session's verdict on read permission. Over http, login credentials may be vulnerable to discovery. Where necessary, administrators may configure Tomcat or Apache to use the secure http protocol (i.e. https).

Persistence

While it is possible to evaluate R expressions in isolation, more functionality is available if results can persist in session memory or in hardware memory. Sumo uses both forms of persistence. A session is created for each user at login, and persists across page requests until logout. Files created on behalf of the user are stored in the server file space associated the user's account. The application developer can take advantage of both.

In some cases, it is possible to create R sessions that persist, not just *within* server sessions, but also *across* them. On systems supporting 'screen' (e.g. Linux, OS X), the command used to start R can be configured in 'sumo.xml' as

```
screen -d -r || screen R -- vanilla
```

An R session will be attached if one exists, else a new one will be created. On logout, the R session persists, and is reattached at next login.

Discussion

While Sumo is not the first attempt at an R-enabled web application, it offers simple deployment and standard architecture. Deployment consists of dropping the Sumo archive into the 'webapps' directory of a running instance of Tomcat. Development can be as simple as editing the resulting text files, even while the application is running.

Like **Rpad** (Short and Grosjean, 2007), Sumo accesses R functionality through the user's web browser. **Rpad** is arguably easier to deploy on a local machine, requiring only the **Rpad** package, but Sumo is easier to deploy on a server, as it requires no customization of Tomcat or Apache.

Relative to Sumo, rApache (Horner, 2012b) is a more sophisticated solution for the development of R-enabled web applications. It uses the Apache web server rather than Tomcat. Sumo is an example application, whereas rApache is a module. As such, rApache remains generally unprejudiced with respect to design choices like those for authentication and persistence. Like **Rpad**, rApache requires server configuration. rApache is currently only available for Linux and Mac OS X, whereas Tomcat (and therefore Sumo) is available for those platforms as well as Windows.

Of broader utility than just for web pages, the packages **brew** (Horner, 2011) and **R.rsp** (Bengtsson, 2012) independently provide functions for text preprocessing. Expressions within delimiters are evaluated in R, substituting the resulting values into the original text. The delimiters are similar or identical to those used in Java Server Pages. Sumo retains such delimiters to evaluate Java expressions and introduces alternative delimiters to evaluate R expressions.

Rook (Horner, 2012a) is a specification that defines an interface between a web server and R. Applications written against the specification can be run unmodified with any supporting server, such as R's

[4] 'war' files have 'zip' file architecture.

[5] The value of tomcat.dir in 'build.xml' may need modification.

internal web server or an rApache instance. As a convenience, **Rook** also supplies classes for writing Rook-compliant applications and for working with R's built-in web server. For comparison, Sumo is compliant with the Java Servlet specification and runs on supporting servers such as Tomcat.

Rserve (Urbanek, 2003) can be adapted for web applications, but is intended for applications where bandwidth is more important and security less so, relative to Sumo.

While Sumo does provide some access to the R command line, those interested primarily in running R remotely should favor the server version of RStudio (http://rstudio.org). RStudio provides some graphical interaction via the **manipulate** package, but the technique does not seem easy to formalize for an independent application.

Display of text is only briefly treated by Sumo; **R2HTML** (Lecoutre, 2003) seems a natural complement.

Summary

Sumo is a web application designed to be easily deployed and modified. It relies on industry-standard software, such as Tomcat and Secure Shell, to enlist the web browser as an interface to R functionality. It has few dependencies, few limitations, good persistence and good security. Enforced authentication of users allows unambiguous assignment of sessions and file space for full-featured use of R. The learning curve is shallow: while some Java is required, most interested parties will already know enough html and R to produce derivative applications quickly.

Acknowledgments

We thank Henrik Bengtsson, Jeffrey Horner, James Rogers, Joseph Hebert, and an anonymous reviewer for valuable comments.

Development and maintenance of Sumo is funded by Metrum Research Group LLC (http://www.metrumrg.com).

Bibliography

H. Bengtsson. *R.rsp: Dynamic generation of scientific reports*, 2012. URL http://CRAN.R-project.org/package=R.rsp. R package version 0.7.1.

J. Horner. *brew: Templating framework for report generation*, 2011. URL http://CRAN.R-project.org/package=brew. R package version 1.0-6.

J. Horner. *Rook: a web server interface for R*, 2012a. URL http://CRAN.R-project.org/package=Rook. R package version 1.0-3.

J. Horner. rApache: Web application development with R and Apache, 2012b. URL http://www.rapache.net.

K. Hornik. R FAQ: Frequently asked questions on R, 2011. URL http://cran.r-project.org/doc/FAQ/R-FAQ.html#R-Web-Interfaces. Version 2.13.2011-07-05, ISBN 3-900051-08-9.

E. Lecoutre. The R2HTML package. *R News*, 3(3):33–36, December 2003. URL http://cran.r-project.org/doc/Rnews/Rnews_2003-3.pdf.

F. Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle and B. Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg, 2002. URL http://www.stat.uni-muenchen.de/~leisch/Sweave. ISBN 3-7908-1517-9.

T. Short and P. Grosjean. *Rpad: Workbook-style, web-based interface to R*, 2007. URL http://www.rpad.org/Rpad. R package version 1.3.0.

S. Urbanek. Rserve: A fast way to provide R functionality to applications. In K. Hornik, F. Leisch, and A. Zeileis, editors, *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, Vienna, Austria, March 20-22 2003. URL http://www.ci.tuwien.ac.at/Conferences/DSC-2003/. ISSN 1609-395X.

Timothy T. Bergsma
Metrum Research Group LLC
2 Tunxis Road Suite 112
Tariffville CT 06081
USA
timb@metrumrg.com

Michael S. Smith
Consultant
26 Harris Fuller Road
Preston, CT 06365
USA
michael_s_smith@yahoo.com