makeR: An R Package for Managing Document Templates and Versions

by Jason M. Bryer

Abstract The idea of build automation is not new. GNU Make and Java Ant are well established and robust build automation systems but require the use and installation of additional software. The **makeR** package provides a simplified framework written entirely in R to manage Sweave projects where multiple versions are created based upon a single source repository. For example, a monthly report where each version is identical, with perhaps the exception of easily extracted properties (e.g. date ranges for data extraction, title, etc.). An example project summarizing recent posts from R-Bloggers is provided.

R (R Development Core Team, 2011), LaTeX (Mittelbach and Rowley, 1999), and Sweave (Leisch, 2002) have proven to be incredibly useful for conducting reproducible research. However, managing document versions within R is limited. The **makeR** package attempts to provide the same ease-of-use for document versioning that the **devtools** (Wickham, 2011) and **ProjectTemplate** (White, 2011) packages have provided for package development and data analysis, respectively. This package attempts to solve the problem where multiple versions of a document are required but the underlying analysis and typesetting code remains static or can be abstracted through the use of variables or properties. For example, many researchers conduct monthly, quarterly, or annual reports where the only difference from version-to-version, from an analysis and typesetting perspective, is the data input. Clearly R and LaTeX are an ideal solution to this problem. The **makeR** package provides a framework to automate the process of generating new documents from a single source repository.

Project framework

There are three attributes to a particular document build, major (which can be numeric or character), minor, and build. Each major version is explicitly defined by the user. For example, if the goal of the project is to generate monthly reports then it would be appropriate to name each version using the month and year. Within each major version are minor versions. The minor version is always numeric and is incremented automatically upon each release. Lastly, build is a global numeric that is atomically incremented upon each document build. This index provides a unique identifier for each document that is created separate from the major and minor version identifiers.

From the perspective of the file system, the **makeR** package maintains a project file, 'PROJECT.xml' (this will be discussed in further detail below), and three directories, 'source', 'build', and 'release'. The 'source' directory should contain an '.Rnw' file along with any support files. The entire contents of this directory, including any subdirectories, will be copied for each build. **makeR** will create a new subdirectory in 'builds' for each unique major and minor combination. Lastly, the 'release' directory will contain "released" documents (i.e. final PDFs). The `releaseVersion` function will rename build files to include the major and minor versions.

Consider, for example, a new project that contains a file 'Example.Rnw' in the 'source' directory. Each call of the `buildVersion` function will copy all files from the 'source' directory to the 'builds/1.0' directory, the call `Stangle`, `Sweave`, and `texi2pdf` to create 'builds/1.0/Example.pdf'. Calling `releaseVersion` will copy 'bulids/1.0/Example.pdf' to 'release/Example-1.0.pdf' and then increment the minor version so that subsequent calls to `buildVersion` will create the 'builds/1.1' directory.

Properties

Properties are what differentiate one version from another and can be defined a the project level or version level. When **makeR** builds a document project level properties will be assigned before version level properties thereby giving version level properties priority over project level properties. The generic functions `addProperty` and `removeProperty`, whereby the first parameter is either objects of class type `Project` or `Version`, provide an interface for manipulating properties.

Package framework

Figure 3 depicts the internal representation of a **makeR** project. All interaction with a project will occur through the `Project` class object. By convention, all properties of classes begin with a capital letter and are represented in the boxes with dashed lined borders. Functions (or methods) begin with a lowercase letter and are represented in shaded boxes. Class objects and lists are represented by rounded squares. Project, Version, and Build are special classes for **makeR** whereas Versions and Builds are simply built

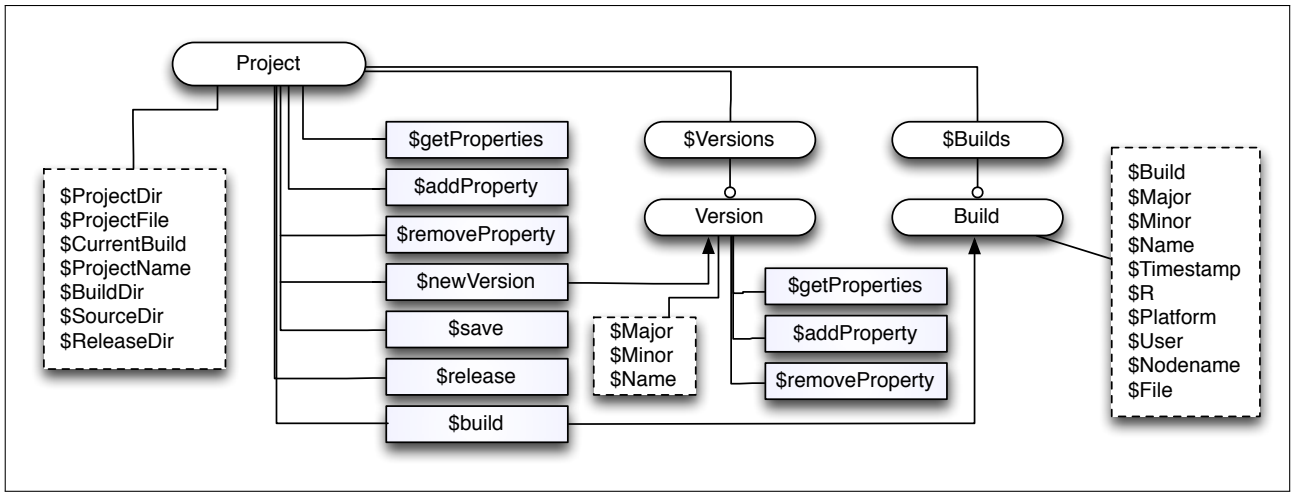


Figure 1: **makeR** structure

in R lists. All manipulation of a project should be done through the appropriate functions. Although the attributes (i.e. elements of a list or class) are accessible given R's programming framework, doing so could result in unexpected results.

Example

The R-Bloggers site provides a wealth of information about R aggregated from many different R bloggers. Like many continuously changing data sources, we wish to periodically report on recent activity on the R-Bloggers site. The base source file, 'rbloggers.Rnw', is included in the package (in the 'inst/doc/robloggers' directory) as well as being hosted on Github. This example can also be run using `demo(makeR)`.

Create new project

```
myProject = Project(name="RBloggers",
  projectDir="~/rbloggers",
  properties=list(email=email, passwd=passwd))
```

Create initial version

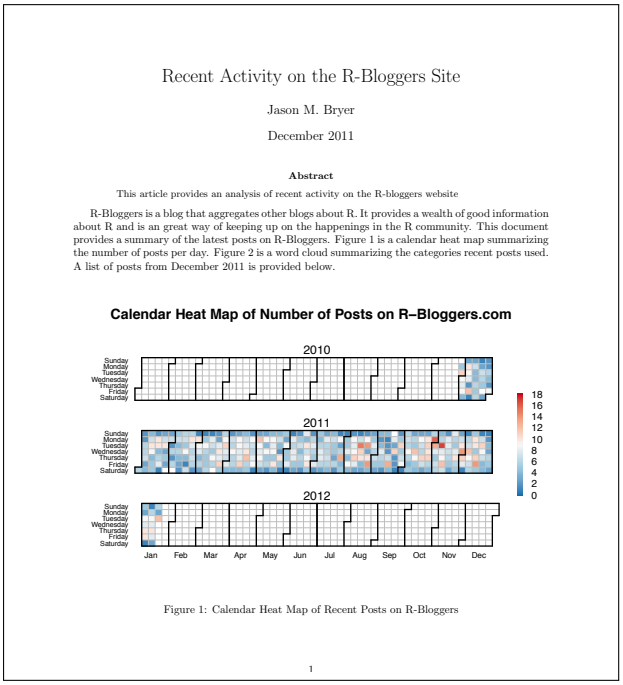
```
myProject$newVersion(myProject,
  name='2011-12',
  properties=list(startDate='2011-12-01',
  endDate='2011-12-31'))
```

Building the document

```
myProject$build()
```

Releasing the document

```
myProject$release()
```



Recent Activity on the R-Bloggers Site

Jason M. Bryer

December 2011

Abstract

This article provides an analysis of recent activity on the R-bloggers website

R-Bloggers is a blog that aggregates other blogs about R. It provides a wealth of good information about R and is an great way of keeping up on the happenings in the R community. This document provides a summary of the latest posts on R-Bloggers. Figure 1 is a calendar heat map summarizing the number of posts per day. Figure 2 is a word cloud summarizing the categories recent posts used. A list of posts from December 2011 is provided below.

Calendar Heat Map of Number of Posts on R–Bloggers.com

Figure 1: Calendar Heat Map of Recent Posts on R-Bloggers

1

Figure 2: Output from the R-Bloggers example.

Project attributes

- `BuildDir` - the directory where builds will occurr.
- `Builds` - list of completed builds.
- `CurrentBuild` - an integer of the last build.
- `ProjectDir` - the base directory where the project is located.
- `ProjectFile` - the name of the source file to be built.
- `ProjectName` - the name of the project.

- `Properties` - a list of the project properties. See also `getProperties()`.
- `ReleaseDir` - the directory where released files will be located.
- `SourceDir` - the directory containing the source files.
- `Versions` - a list of the project versions.
- `file.info` - the file info from the last time the 'PROJECT.xml' file was read.

Project methods

- `build` Builds the project.
  - `version` - (optional) the version to build.
  - `saveEnv` - if TRUE, the build environment (.rda) will be saved in the build directory.
  - `builder` - the builder function.
- `rebuild` Rebuilds the project without first copying the files.
  - `version` - (optional) the version to rebuild.
  - `saveEnv` - if TRUE, the build environment (.rda) will be saved in the build directory.
  - `builder` - the bulder function.
- `save` Saves the PROJECT.xml file.
- `newVersion` Creates a new versions of the project.
  - `name` - (optional) the version name.
  - `properties` - version specific properties.
- `release` Releases a version (i.e. copies the built file to the releases directory)
  - `version` - (optional) the version to release. If omitted the latest version will be released.
- `getProperties` Returns the project properties.
- `addProperty` Adds a project property.
  - `name` - The property name.
  - `value` - The property value.
- `removeProperty` Removes the given project property.
  - `name` - The property name.
- `getReleases` Returns a list of released files.
- `openRelease` Opens the given released file with the system's default application.
  - `file` - The released file to open.

The project file format

The 'PROJECT.xml' file defines the properties, versions, and builds of a particular project. The file created and edited using the **XML** (Lang, 2011) package. Although the **makeR** package provides functions to manage projects, using XML allows for other R programmers to interact with the **makeR** project framework. Figure 3 represents the contents of 'PROJECT.xml' after running `demo(makeR)`.

Custom builders

```
builder.rnw <- function(file, theenv, ...) {
  Stangle(file)
  envstr = env2string(theenv)
  thecall = paste('Rscript -e "', envstr,
      ' Sweave(', file, ')"', sep='')
  system(thecall)
  texi2pdf(paste(substr(file, 1,
      (nchar(file)-4)), '.tex', sep=''))
  return(paste(substr(file, 1,
      (nchar(file)-4)), '.pdf', sep=''))
}
```

Package development

The latest stable version of **makeR** can be installed from your local CRAN server. Development versions are hosted on Github. The latest development version can be installed using the **devtools** (Wickham, 2011) package:

```
install_github('makeR', 'jbryer')
```

Bibliography

D. T. Lang. *XML: Tools for parsing and generating XML within R and S-Plus.*, 2011. URL http://CRAN.R-project.org/package=XML. R package version 3.4-3.

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. Physika Verlag, Heidelberg, Germany, 2002. ISBN 3-7908-1517-9.

F. Mittelbach and C. Rowley. *The LaTeX3 project*, 1999. URL http://www.latex-project.org/guides/ltx3info.pdf.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL http://www.R-project.org/. ISBN 3-900051-07-0.

```
<?xml version="1.0"?>
<project name="RBloggers" buildDir="build" releaseDir="release" sourceDir="source">
  <property name="email" value="GOOGLE READER USERNAME" type="character"/>
  <property name="passwd" value="GOOGLE READER PASSWORD" type="character"/>
  <versions>
    <version name="2011-12" major="1" minor="1">
      <property name="startDate" value="2011-12-01" type="character"/>
      <property name="endDate" value="2011-12-31" type="character"/>
    </version>
  </versions>
  <builds>
    <build major="1" minor="0" build="1" name="2011-12" timestamp="Wed Jan 18 12:29:51 2012"
        R="R version 2.14.0 (2011-10-31)" platform="x86_64-apple-darwin9.8.0"
        nodename="Jason-Bryers-MacBook-Air-2.local"
        user="jbryer" file="rbloggers.pdf"/>
  </builds>
</project>
```

Figure 3: 'PROJECT.xml' for the R-Bloggers demo project.

J. M. White. *ProjectTemplate: Automates the creation of new statistical analysis projects.*, 2011. URL http://CRAN.R-project.org/package=ProjectTemplate. R package version 0.3-5.

H. Wickham. *devtools: Tools to make developing R code easier*, 2011. URL http://CRAN.R-project.org/package=devtools. R package version 0.4.

Jason M. Bryer
Excelsior College
7 Columbia Circle
Albany, NY 12203
USA
jason@bryer.org