Automating R Package Citations in Reproducible Research Documents

Christopher Gandrud[1]

Abstract All of the R packages a reproducible computational research document relies on should be fully cited. R packages' syntax and capabilities change over time. If a piece of research does not fully cited the versions of the packages it used, it will be difficult to reproduce it. In this short note I introduce the `LoadandCite` command from the **repmis** R package. The command makes it much easier to document the R packages a piece of research depends on. If included in a dynamic reproducible research document, `LoadandCite` can (a) load all the R packages used to create the document, (b) create a BibTeX file that is updated every time the document is compiled with each loaded package's full citation information. It also allows the user to install the packages, including specific versions. Using this command makes it easier to create really reproducible research with R.

One of R's (R Core Team, 2013) main advantages is its very active community of developers who are rapidly improving and expanding its capabilities by creating add-on packages. Many pieces of computational research now depend on these packages. The rapid pace of package development does have an important drawback: it can quickly become difficult to reproduce research[2] that depends on particular package versions. So, for research to be really reproducible it is very important that researchers fully cite–including the version number–all of the R packages that their results depend on (see Jackson, 2012, for more details) as well as the version of R that they used.

However, we can do better than just create a static package citation list. There are two practical problems that are not solved by simply including citations in a static bibliography. First, though we may intend to fully cite the versions of the packages we used in our research, we may not actually do so. This is especially true if there are a number of packages that are updated over the course of the research process. If the package version cited differs from the one actually used and there have been syntax or other changes to the packages between the two versions, then it could be very difficult to reproduce the research. Second, to make our research really reproducible it should be reasonably *easy* for other researchers to not only understand how results were achieved in the abstract, but also replicate these results. Literate programming technologies, such as **knitr** (Xie, 2013a), have made it much easier to reproduce research results. Nontheless, would-be reproducers face an important obstacle when they compile these documents: they need to install any package dependencies that they don't already have. This may involve arduously hunting down one-by-one specific versions of the packages used in the original paper.

In this short note I propose and demonstrate a solution to these two problems. Researchers can use the `LoadandCite` command from the **repmis** package (Gandrud, 2013b) in their reproducible research documents to (a) load all of the packages that they used to create the document, (b) create a BibTeX file that is updated every time the document is compiled with each loaded package's full citation information, and (c) install specific package versions.

Concerns to Keep in Mind

Before discussing `LoadandCite`'s syntax it is important to address three issues that researchers should consider when using the command: (1) the need to manually verify automatically generated citations, (2) issues with installing old packages, and (3) difficulties future-proofing your research not addressed by `LoadandCite`.

Pitfalls of automatically generating package citations

Though `LoadandCite` makes it much easier to cite the packages used to create an analysis, it depends on each package developers/maintainers to properly document their packages. Most packages hosted

[1] I would like to thank Karthik Ram, who contributed code to the command, as well as two reviewers including Carl Boettiger. The full source code files used to create this paper are available at: `https://github.com/christophergandrud/LoadandCiteNote`

[2] For a discussion of the importance of reproducibility in computational research see Peng (2011).

on sites such as the Comprehensive R Archive Network (CRAN) are relatively well documented. However, their documentation may lack key information such as

Pitfalls of installing old packages

FILL IN

Difficulties future-proofing your research

FILL IN

Basic Syntax

Let's look at the basic `LoadandCite` syntax. Below, we'll see how to use it in reproducible LaTeX documents. The main argument that `LoadandCite` takes is a character vector of package names. For example, to load the **car** (Fox and Weisberg, 2013) and **knitr** packages create the following character vector:

```
Packages <- c(car, knitr)
```

We can now simply add this vector object to `LoadandCite`'s `pkgs` argument. We can also specify the name of the BibTeX file that we want to create and save the citations to with the `file` argument.[3] Let's use the file name 'Example.bib':

```
repmis::LoadandCite(pkgs = Packages, file = Example.bib)
```

The two packages will be loaded and a BibTeX file with citation information for the loaded versions created in the working directory. **Note** it is important to use a file name that is different from the one you use for non-R package citations. Otherwise you will accidentally overwrite your other citations.

If we want to have the packages installed as well as loaded and cited, we can set the argument `install = TRUE`. This will install the most recent version of the packages from the repository specified in the `repos` argument.[4] If we want to install specific package versions, we can include a character vector of package version numbers to have `LoadandCite` install from CRAN. The order of the package versions must match the order of the packages listed in the `pkgs` argument. For example, to install **car** version 2.0-17 and **knitr** version 1.1 use:

```
Vers <- c(2.0-17, 1.1)

repmis::LoadandCite(pkgs = Packages, install = TRUE,
                    versions = Vers, file = Example.bib)
```

You should avoid using old package versions in active research projects as they may contain bugs that were addressed in subsequent updates. It is better to specify the versions only in documents you are making available for replication.[5] Likewise, you probably only want to set `install = TRUE` in a file intended for replication. It will be unnecessarily and time consuming to install the packages every time you run your code.

If you are reproducing a piece of research and installing old package versions, it can be a good idea to install these into a separate library from the one(s) you normally use. You can specify what library to have `LoadandCite` install the packages into with the `lib` argument.[6]

R packages may depend on specific versions of R itself. The `Rversion` argument allows you to specify the version of R that you used to create your analysis. If you set, for example, the argument `Rversion = 3.0.2` then `LoadandCite` will check if the currently running version of R is 3.0.2. If it is not, a waring will be printed. Old versions of R can be downloaded from CRAN. Note that you may not be able to install an old version because it does not run on your system. This

[3] If no file is specified, then the packages are only loaded.

[4] `repos`' default is the user's default.

[5] If `install = FALSE` (the default) then specific package versions will not be installed, i.e. `LoadandCite` will ignore any values in the `versions` argument.

[6] If needed, remember to then add this library path to R with `.libPath`.

Using LoadandCite in a Knitted LaTeX Document

Possibly the best way to use **LoadandCite** with your research is to place it in a 'code chunk' at the beginning of a **knitr** created LaTeX document.[7] Doing this will load every R package used to create the document and update the bibliography file every time you knit it. For example:

```
\textless{\textless}include=FALSE\textgreater{\textgreater}=
  Packages <- c(car, knitr, repmis)
  repmis::LoadandCite(pkgs = Packages, file = Example.bib)
@
```

The code `<<include=FALSE>>=` and `@` delimit the beginning and end of the R code chunk. The `include=FALSE` option tells **knitr** to not include any messages, warnings, and so on created by running the code in the compiled presentation document. For more information on how to create reproducible research documents with **knitr** see Xie (2013b) and Gandrud (2013a).

You can cite the packages in your document using BibTeX cite keys as usual. The BibTeX keys created by `LoadandCite`[8] have the following pattern `R-PACKAGE_NAME`. For example, the **car** package's cite key will be `R-car`. To insert a citation for the **car** package like '(Fox and Weisberg, 2013)' in the text of your document simply type: `\citep{R-car}` in your LaTeX document where you would like the citation to appear. `LoadandCite` automatically generates a citation for the currently running version of R. The cite key that it assigns is `CiteR`.

Additionally add the name of the BibTeX file created by `LoadandCite` to your `bibliography` command near the end of your LaTeX document, i.e. `\bibliography{Example}`. Note that because **repmis** is itself an add-on R package, you should clearly inform anyone who would want to reproduce your research that they need to install it first before running `LoadandCite`. You could do this in a README file accompanying the replication files.

Summary

R's capabilities are developing very quickly. To be able to take advantage of these expanding capabilities while also making our research easily reproducible in the future, we need to make sure that the add-on packages we use are fully cited and that it is easy for future users to install the versions we used. Using `LoadandCite` in dynamically created research documents helps us do this.

Bibliography

J. Fox and S. Weisberg. *car: Companion to Applied Regression*, 2013. URL `http://CRAN.R-project.org/package=car`. R package version 2.0-19. [p2, 3]

C. Gandrud. *Reproducible Research with R and RStudio*. Chapman and Hall/CRC Press, Boca Raton, FL, 2013a. [p3]

C. Gandrud. *repmis: A collection of miscellaneous tools for reproducible research with R.*, 2013b. URL `http://cran.r-project.org/web/packages/repmis/`. R package version 0.2.7. [p1]

M. Jackson. How to cite and describe software. *Software Sustainability Institute*, 2012. `http://software.ac.uk/so-exactly-what-software-did-you-use`. [p1]

R. D. Peng. Reproducible Research in Computational Science. *Science*, 334:1226–1227, 2011. [p1]

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL `http://www.R-project.org/`. Version 3.0.2. [p1]

Y. Xie. *knitr: A general-purpose package for dynamic report generation in R*, 2013a. URL `http://CRAN.R-project.org/package=knitr`. R package version 1.5. [p1]

Y. Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC Press, Boca Raton, FL, 2013b. [p3]

[7] The command also works with a **knitr** Markdown document rendered with Pandoc. This can be useful for creating reproducible webpages. See the knitr documentation site for how to use Pandoc from R.

[8] The keys are generated with code modified from **knitr**'s `write_bib` command. `LoadandCite` does not formally depend on **knitr** to make it possible to install old versions of that package.

Christopher Gandrud
Hertie School of Governance
Friedrichstrasse 180
Berlin, 10117
Germany
christopher.gandrud@gmail.com