R › R devel Login Register

# [RFC] A case for freezing CRAN

Classic List Threaded g+1 0 70 messages Options

1 2 3 4

**Jeroen Ooms.** 89 posts

Mar 18, 2014; 4:24pm [RFC] A case for freezing CRAN

Reply | Threaded | More

This came up again recently with an irreproducible paper. Below an attempt to make a case for extending the r-devel/r-release cycle to CRAN packages. These suggestions are not in any way intended as criticism on anyone or the status quo.

The proposal described in [1] is to freeze a snapshot of CRAN along with every release of R. In this design, updates for contributed packages treated the same as updates for base packages in the sense that they are only published to the r-devel branch of CRAN and do not affect users of "released" versions of R. Thereby all users, stacks and applications using a particular version of R will by default be using the identical version of each CRAN package. The bioconductor project uses similar policies.

This system has several important advantages:

## Reproducibility

Currently r/sweave/knitr scripts are unstable because of ambiguity introduced by constantly changing cran packages. This causes scripts to break or change behavior when upstream packages are updated, which makes reproducing old results extremely difficult.

A common counter-argument is that script authors should document package versions used in the script using sessionInfo(). However even if authors would manually do this, reconstructing the author's environment from this information is cumbersome and often nearly impossible, because binary packages might no longer be available, dependency conflicts, etc. See [1] for a worked example. In practice, the current system causes many results or documents generated with R no to be reproducible, sometimes already after a few months.

In a system where contributed packages inherit the r-base release cycle, scripts will behave the same across users/systems/time within a given version of R. This severely reduces ambiguity of R behavior, and has the potential of making reproducibility a natural part of the language, rather than a tedious exercise.

## Repository Management

Just like scripts suffer from upstream changes, so do packages depending on other packages. A particular package that has been developed and tested against the current version of a particular dependency is not guaranteed to work against *any future version* of that dependency. Therefore, packages inevitably break over time as their dependencies are updated.

One recent example is the Rcpp 0.11 release, which required all reverse dependencies to be rebuild/modified. This updated caused some serious disruption on our production servers. Initially we refrained from updating Rcpp on these servers to prevent currently installed packages depending on Rcpp to stop working. However soon after the Rcpp 0.11 release, many other cran packages started to require Rcpp >= 0.11, and our users started complaining about not being able to install those packages. This resulted in the impossible situation where currently installed packages would not work with the new Rcpp, but newly installed packages would not work with the old Rcpp.

Current CRAN policies blame this problem on package authors. However as is explained in [1], this policy does not solve anything, is unsustainable with growing repository size, and sets completely the wrong incentives for contributing code. Progress comes with breaking changes, and the system should be able to accommodate this. Much of the trouble could have been prevented by a system that does not push bleeding edge updates straight to end-users, but has a devel branch where conflicts are resolved before publishing them in the next r-release.

## Reliability

Another example, this time on a very small scale. We recently discovered that R code plotting medal counts from the Sochi Olympics generated different results for users on OSX than it did on Linux/Windows. After some debugging, we narrowed it down to the XML package. The application used the following code to scrape results from the Sochi website:

```
XML::readHTMLTable("http://www.sochi2014.com/en/speed-skating", which=2, skip=1)
```

This code was developed and tested on mac, but results in a different winner on windows/linux. This happens because the current version of the XML package on CRAN is 3.98, but the latest mac binary is 3.95. Apparently this new version of XML introduces a tiny change that causes html-table-headers to become colnames, rather than a row in the matrix, resulting in different medal counts.

This example illustrates that we should never assume package versions to be interchangeable. Any small bugfix release can have side effects altering results. It is impossible to protect code against such upstream changes using CMD check or unit testing. All R scripts and packages are really only developed and tested for a single version of their dependencies. Assuming anything else makes results untrustworthy, and code unreliable.

## Summary

Extending the r-release cycle to CRAN seems like a solution that would be easy to implement. Package updates simply only get pushed to the r-devel branches of cran, rather than r-release and r-release-old. This separates development from production/use in a way that is common sense in most open source communities. Benefits for R include:

- Regular R users (statisticians, researchers, students, teachers) can share their homemade scripts/documents/packages and rely on them to work and produce the same results within a given version of R, without manual efforts to manage package versions.

- Package authors can publish breaking changes to the devel branch without causing major disruption or affecting users and/or maintainers. Authors of depending packages have a timeframe to sync their package with upstream changes before the next release.

- CRAN maintainers can focus quality control and testing efforts on the devel branch around the time of the code freeze. No need for crisis management when a package update introduces some severe breaking changes. Users of released versions are unaffected.

[1] http://journal.r-project.org/archive/2013-1/ooms.pdf

---

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Frank Harrell**

1419 posts

Mar 19, 2014; 8:26am Re: [RFC] A case for freezing CRAN

Reply | Threaded | More

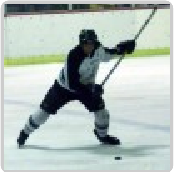
To me it boils down to one simple question: is an update to a package on CRAN more likely to (1) fix a bug, (2) introduce a bug or downward incompatibility, or (3) add a new feature or fix a compatibility problem without introducing a bug? I think the probability of (1) | (3) is much greater than the probability of (2), hence the current approach maximizes user benefit.

Frank
--
Frank E Harrell Jr Professor and Chairman School of Medicine
Department of Biostatistics Vanderbilt University

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Frank Harrell
Department of Biostatistics, Vanderbilt University

**Joshua Ulrich**



534 posts

Mar 19, 2014; 8:52am Re: [RFC] A case for freezing CRAN

Reply | Threaded | More

In reply to this post by Jeroen Ooms.

On Tue, Mar 18, 2014 at 3:24 PM, Jeroen Ooms <[hidden email]> wrote:
<snip>
> ## Summary
>
> Extending the r-release cycle to CRAN seems like a solution that would
> be easy to implement. Package updates simply only get pushed to the
> r-devel branches of cran, rather than r-release and r-release-old.
> This separates development from production/use in a way that is common
> sense in most open source communities. Benefits for R include:
>
Nothing is ever as simple as it seems (especially from the perspective of one who won't be doing the work).

There is nothing preventing you (or anyone else) from creating repositories that do what you suggest. Create a CRAN mirror (or more than one) that only include the package versions you think they should. Then have your production servers use it (them) instead of CRAN.

Better yet, make those repositories public. If many people like your idea, they will use your new repositories instead of CRAN. There is no reason to impose this change on all world-wide CRAN users.

Best,
--

Joshua Ulrich | about.me/joshuaulrich
FOSS Trading | www.fosstrading.com

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Duncan Murdoch-2** Mar 19, 2014; 8:52am **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

3037 posts

In reply to this post by Jeroen Ooms.

I don't see why CRAN needs to be involved in this effort at all. A third party could take snapshots of CRAN at R release dates, and make those available to package users in a separate repository. It is not hard to set a different repository than CRAN as the default location from which to obtain packages.

The only objection I can see to this is that it requires extra work by the third party, rather than extra work by the CRAN team. I don't think the total amount of work required is much different. I'm very unsympathetic to proposals to dump work on others.

Duncan Murdoch

On 18/03/2014 4:24 PM, Jeroen Ooms wrote:

> This came up again recently with an irreproducible paper. Below an
> attempt to make a case for extending the r-devel/r-release cycle to
> CRAN packages. These suggestions are not in any way intended as
> criticism on anyone or the status quo.
>
> The proposal described in [1] is to freeze a snapshot of CRAN along
> with every release of R. In this design, updates for contributed
> packages treated the same as updates for base packages in the sense
> that they are only published to the r-devel branch of CRAN and do not

... [show rest of quote]

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Kasper Daniel Hans** Mar 19, 2014; 10:00am **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

78 posts

In reply to this post by Joshua Ulrich

Our experience in Bioconductor is that this is a pretty hard problem.

What the OP presumably wants is some guarantee that all packages on CRAN work well together. A good example is when Rcpp was updated, it broke other packages (quick note: The Rcpp developers do a incredible amount of work to deal with this; it is almost impossible to not have a few days of chaos). Ensuring this is not a trivial task, and it requires some buy-in both from the "repository" and from the developers.

For Bioconductor it is even harder as the dependency graph of Bioconductor is much more involved than the one for CRAN, where most packages depends only on a few other packages. This is why we need to do this for Bioc.

Based on my experience with CRAN I am not sure I see a need for a coordinated release (or rather, I can sympathize with the need, but I don't think the effort is worth it).

What would be more useful in terms of reproducibility is the capability of installing a specific version of a package from a repository using install.packages(), which would require archiving older versions in a coordinated fashion. I know CRAN archives old versions, but I am not aware if we can programmatically query the repository about this.

Best,
Kasper

On Wed, Mar 19, 2014 at 8:52 AM, Joshua Ulrich <[hidden email]>wrote:

> On Tue, Mar 18, 2014 at 3:24 PM, Jeroen Ooms <[hidden email]>
> wrote:
> <snip>
> > ## Summary
> >
> > Extending the r-release cycle to CRAN seems like a solution that would
> > be easy to implement. Package updates simply only get pushed to the
> > r-devel branches of cran, rather than r-release and r-release-old.
> > This separates development from production/use in a way that is common
... [show rest of quote]

[[alternative HTML version deleted]]

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Dirk Eddelbuettel** Mar 19, 2014; 10:01am **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

1263 posts

In reply to this post by Joshua Ulrich

Piling on:

On 19 March 2014 at 07:52, Joshua Ulrich wrote:
| There is nothing preventing you (or anyone else) from creating
| repositories that do what you suggest.  Create a CRAN mirror (or more
| than one) that only include the package versions you think they
| should.  Then have your production servers use it (them) instead of
| CRAN.
|
| Better yet, make those repositories public.  If many people like your
| idea, they will use your new repositories instead of CRAN.  There is
| no reason to impose this change on all world-wide CRAN users.

On 19 March 2014 at 08:52, Duncan Murdoch wrote:
| I don't see why CRAN needs to be involved in this effort at all.  A
| third party could take snapshots of CRAN at R release dates, and make
| those available to package users in a separate repository.  It is not
| hard to set a different repository than CRAN as the default location
| from which to obtain packages.
|
| The only objection I can see to this is that it requires extra work by
| the third party, rather than extra work by the CRAN team. I don't think
| the total amount of work required is much different.  I'm very
| unsympathetic to proposals to dump work on others.

And to a first approximation some of those efforts already exist:

-- 200+ r-cran-* packages in Debian proper

-- 2000+ r-cran-* packages in Michael's c2d4u (via launchpad)

-- 5000+ r-cran-* packages in Don's debian-r.debian.net

The only difference here is that Jeroen wants to organize source packages.
But that is just a matter of stacking them in directory trees and calling

```
    setwd("/path/to/root/of/your/repo/version")
    tools::write_PACKAGES(".", type="source")'
```

to create PACKAGES and PACKAGES.gz.

Dirk

--
Dirk Eddelbuettel | [hidden email] | http://dirk.eddelbuettel.com

______________________________________________
[hidden email] mailing list

https://stat.ethz.ch/mailman/listinfo/r-devel

**hadley wickham** Mar 19, 2014; 10:17am Re: [RFC] A case for freezing CRAN Reply | Threaded | More

1799 posts

In reply to this post by Kasper Daniel Hansen-2

> What would be more useful in terms of reproducibility is the capability of
> installing a specific version of a package from a repository using
> install.packages(), which would require archiving older versions in a
> coordinated fashion. I know CRAN archives old versions, but I am not aware
> if we can programmatically query the repository about this.

See devtools::install_version().

The main caveat is that you also need to be able to build the package, and ensure you have dependencies that work with that version.

Hadley

--
http://had.co.nz/

---

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Geoff Jentry** Mar 19, 2014; 11:22am Re: [RFC] A case for freezing CRAN Reply | Threaded | More

17 posts

In reply to this post by Jeroen Ooms.

> using the identical version of each CRAN package. The bioconductor
> project uses similar policies.

While I agree that this can be an issue, I don't think it is fair to compare CRAN to BioC. Unless things have changed, the latter has a more rigorous barrier to entry which includes buy in of various ideals (e.g. interoperability w/ other BioC packages, making use of BioC constructs, the official release cycle). All of that requires extra management overhead (read: human effort) which considering that CRAN isn't exactly swimming in spare cycles seems unlikely to happen.

It seems like one could set up a curated CRAN-a-like quite easily, advertise the heck out of it and let the "market" decide. That is, IMO, the beauty of open source.

-J

---

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Jeroen Ooms.** Mar 19, 2014; 1:59pm Re: [RFC] A case for freezing CRAN Reply | Threaded | More

89 posts

In reply to this post by Duncan Murdoch-2

On Wed, Mar 19, 2014 at 5:52 AM, Duncan Murdoch <[hidden email]>wrote:

> I don't see why CRAN needs to be involved in this effort at all. A third
> party could take snapshots of CRAN at R release dates, and make those
> available to package users in a separate repository. It is not hard to set
> a different repository than CRAN as the default location from which to
> obtain packages.
>

I am happy to see many people giving this some thought and engage in the discussion.

Several have suggested that staging & freezing can be simply done by a third party. This solution and its limitations is also described in the paper [1] in the section titled "R: downstream staging and repackaging".

If this would solve the problem without affecting CRAN, we would have been done this obviously. In fact, as described in the paper and pointed out by some people, initiatives such as Debian or Revolution Enterprise already include a frozen library of R packages. Also companies like Google maintain their own internal repository with packages that are used throughout the company.

The problem with this approach is that when you using some 3rd party package snapshot, your r/sweave scripts will still only be reliable/reproducible for other users of that specific snapshot. E.g. for the examples above, a script that is written in R 3.0 by a Debian user is not guaranteed to work on R 3.0 in Google, or R 3.0 on some other 3rd party cran snapshot. Hence this solution merely redefines the problem from "this script depends on pkgA 1.1 and pkgB 0.2.3" to "this script depends on repository foo 2.0". And given that most users would still be pulling packages straight from CRAN, it would still be terribly difficult to reproduce a 5 year old sweave script from e.g. JSS.

For this reason I believe the only effective place to organize this staging is all the way upstream, on CRAN. Imagine a world where your r/sweave script would be reliable/reproducible, out of the box, on any system, any platform in any company using on R 3.0. No need to investigate which specific packages or cran snapshot the author was using at the time of writing the script, and trying to reconstruct such libraries for each script you want to reproduce. No ambiguity about which package versions are used by R 3.0. However for better or worse, I think this could only be accomplished with a cran release cycle (i.e. "universal snapshots") accompanying the already existing r releases.

> The only objection I can see to this is that it requires extra work by the
> third party, rather than extra work by the CRAN team. I don't think the
> total amount of work required is much different.  I'm very unsympathetic to
> proposals to dump work on others.

I am merely trying to discuss a technical issue in an attempt to improve reliability of our software and reproducibility of papers created with R.

[[alternative HTML version deleted]]

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Spencer Graves-2** Mar 19, 2014; 2:36pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

310 posts

What about having this purpose met with something like an expansion of R-Forge?  We could have packages submitted to R-Forge rather than CRAN, and people who wanted the latest could get it from R-Forge.  If changes I make on R-Forge break a reverse dependency, emails explaining the problem are sent to both me and the maintainer for the package I broke.

The budget for R-Forge would almost certainly need to be increased:  They currently disable many of the tests they once ran.

Regarding budget, the R Project would get more donations if they asked for them and made it easier to contribute.  I've tried multiple times without success to find a way to donate.  I didn't try hard, but it shouldn't be hard ;-)  (And donations should be accepted in US dollars and Euros -- and maybe other currencies.) There should be a procedure whereby anyone could receive a pro forma invoice, which they can pay or ignore as they choose.  I mention this, because many grants could cover a reasonable fee provided they have an invoice.

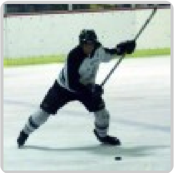
Spencer Graves

On 3/19/2014 10:59 AM, Jeroen Ooms wrote:

> On Wed, Mar 19, 2014 at 5:52 AM, Duncan Murdoch <[hidden email]>wrote:
>
>> I don't see why CRAN needs to be involved in this effort at all.  A third
>> party could take snapshots of CRAN at R release dates, and make those
>> available to package users in a separate repository.  It is not hard to set
>> a different repository than CRAN as the default location from which to
>> obtain packages.
>>
> I am happy to see many people giving this some thought and engage in the
... [show rest of quote]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Joshua Ulrich** Mar 19, 2014; 2:50pm Re: [RFC] A case for freezing CRAN Reply | Threaded | More



534 posts

In reply to this post by Jeroen Ooms.

On Wed, Mar 19, 2014 at 12:59 PM, Jeroen Ooms <[hidden email]> wrote:

> On Wed, Mar 19, 2014 at 5:52 AM, Duncan Murdoch <[hidden email]>wrote:
>
>> I don't see why CRAN needs to be involved in this effort at all.  A third
>> party could take snapshots of CRAN at R release dates, and make those
>> available to package users in a separate repository.  It is not hard to set
>> a different repository than CRAN as the default location from which to
>> obtain packages.
>>
>
... [show rest of quote]
The suggested solution is not described in the referenced article.  It
was not suggested that it be the operating system's responsibility to
distribute snapshots, nor was it suggested to create binary
repositories for specific operating systems, nor was it suggested to
freeze only a subset of CRAN packages.

> The problem with this approach is that when you using some 3rd party
> package snapshot, your r/sweave scripts will still only be
> reliable/reproducible for other users of that specific snapshot. E.g. for
> the examples above, a script that is written in R 3.0 by a Debian user is
> not guaranteed to work on R 3.0 in Google, or R 3.0 on some other 3rd party
> cran snapshot. Hence this solution merely redefines the problem from "this
> script depends on pkgA 1.1 and pkgB 0.2.3" to "this script depends on
> repository foo 2.0". And given that most users would still be pulling
> packages straight from CRAN, it would still be terribly difficult to
... [show rest of quote]
This can be solved by the third party making the repository public.

> For this reason I believe the only effective place to organize this staging
> is all the way upstream, on CRAN. Imagine a world where your r/sweave
> script would be reliable/reproducible, out of the box, on any system, any
> platform in any company using on R 3.0. No need to investigate which
> specific packages or cran snapshot the author was using at the time of
> writing the script, and trying to reconstruct such libraries for each
> script you want to reproduce. No ambiguity about which package versions are
> used by R 3.0. However for better or worse, I think this could only be
> accomplished with a cran release cycle (i.e. "universal snapshots")
... [show rest of quote]
This could be done by a public third-party repository, independent of
CRAN.  However, you would need to find a way to actively _prevent_
people from installing newer versions of packages with the stable R
releases.

--
Joshua Ulrich  |  about.me/joshuaulrich
FOSS Trading  |  www.fosstrading.com

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Carl Boettiger** Mar 19, 2014; 2:58pm Re: [RFC] A case for freezing CRAN Reply | Threaded | More

5 posts

In reply to this post by Spencer Graves-2

Dear list,

I'm curious what people would think of a more modest proposal at this time:

State the version of the dependencies used by the package authors when the
package was built.

Eventually CRAN could enforce such a statement be present in the
description. We encourage users to declare the version of the packages they
use in publications, so why not have the same expectation of developers?
This would help address the problem of archived packages that Jeroen
raises, as it is currently it is impossible to reliably install archived
packages because their dependencies have since been updated and are no
longer compatible. (Even if it passes checks and installs, we have no way
of knowing if the upstream changes have introduced a bug). This
information would be relatively straight forward to capture, shouldn't
change the way anyone currently uses CRAN, and should address a major pain
point anyone trying to install archived versions from CRAN has probably
encountered. What am I overlooking?

Carl

On Wed, Mar 19, 2014 at 11:36 AM, Spencer Graves <
[hidden email]> wrote:

> What about having this purpose met with something like an expansion
> of R-Forge? We could have packages submitted to R-Forge rather than CRAN,
> and people who wanted the latest could get it from R-Forge. If changes I
> make on R-Forge break a reverse dependency, emails explaining the problem
> are sent to both me and the maintainer for the package I broke.
>
>
> The budget for R-Forge would almost certainly need to be increased:
> They currently disable many of the tests they once ran.

... [show rest of quote]

--
Carl Boettiger
UC Santa Cruz
http://carlboettiger.info/

[[alternative HTML version deleted]]

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Jeroen Ooms.** Mar 19, 2014; 3:09pm Re: [RFC] A case for freezing CRAN Reply | Threaded | More

89 posts

In reply to this post by Kasper Daniel Hansen-2

On Wed, Mar 19, 2014 at 7:00 AM, Kasper Daniel Hansen
<[hidden email]> wrote:

> Our experience in Bioconductor is that this is a pretty hard problem.
>
> What the OP presumably wants is some guarantee that all packages on CRAN work well together.

Obviously we can not guarantee that all packages on CRAN work
together. But what we can do is prevent problems that are introduced
by version ambiguity. If author develops and tests a script/package
with dependency Rcpp 0.10.6, the best chance of making that script or
package work for other users is using Rcpp 0.10.6.

This especially holds if there is a big time difference between the author creating the pkg/script and someone using it. In practice most Sweave/knitr scripts used for generating papers and articles can not be reproduced after a while because the dependency packages have changed in the mean time. These problem can largely be mitigated with a release cycle.

I am not arguing that anyone should put manual effort into testing that packages work together. On the contrary: a system that separates development from released branches prevents you from having to continuously test all reverse dependencies for every package update.

My argument is simply that many problems introduced by version ambiguity can be prevented if we can unite the entire R community around using a single version of each CRAN package for every specific release of R. Similar to how linux distributions use a single version of each software package in a particular release of the distribution.

---

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Hervé Pagès** Mar 19, 2014; 5:00pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

452 posts

In reply to this post by Kasper Daniel Hansen-2

Hi,

On 03/19/2014 07:00 AM, Kasper Daniel Hansen wrote:
> Our experience in Bioconductor is that this is a pretty hard problem.

What's hard and requires a substantial amount of human resources is to run our build system (set up the build machines, keep up with changes in R, babysit the builds, assist developers with build issues, etc...)

But *freezing* the CRAN packages for each version of R is *very* easy to do. The CRAN maintainers already do it for the binary packages. What could be the reason for not doing it for source packages too? Maybe in prehistoric times there was this belief that a source package was aimed to remain compatible with all versions of R, present and future, but that dream is dead and gone...

Right now the layout of the CRAN package repo is:

```
├── src
│   └── contrib
└── bin
    ├── windows
    │   └── contrib
    │       ├ ...
    │       ├ 3.0
    │       ├ 3.1
    │       ├ ...
    └── macosx
        └── contrib
            ├ ...
            ├ 3.0
            ├ 3.1
            ├ ...
```

when it could be:

```
├── 3.0
│   ├── src
│   │   └── contrib
│   └── bin
│       ├── windows
│       │   └── contrib
│       └── macosx
│           └── contrib
├── 3.1
│   ├── src
```

```
   │    │    └── contrib
   │    └── bin
   │         ├── windows
   │         │    └── contrib
   │         └── macosx
   │              └── contrib
   ├── ...
```

That is: the split by version is done at the top, not at the bottom.

It doesn't use more disk space than the current layout (you can just throw the src/contrib/Archive/ folder away, there is no more need for it).

install.packages() and family would need to be modified a little bit to work with this new layout. And that's all!

The never ending changes in Mac OS X binary formats can be handled in a cleaner way i.e. no more symlinks under bin/macosx to keep backward compatibility with different binary formats and with old versions of install.packages().

Then in 10 years from now, you can reproduce an analysis that you did today with R-3.0. Because when you'll install R-3.0 and the packages required for this analysis, you'll end up with exactly the same packages as today.

Cheers,
H.

>
> What the OP presumably wants is some guarantee that all packages on CRAN
> work well together.  A good example is when Rcpp was updated, it broke
> other packages (quick note: The Rcpp developers do a incredible amount of
> work to deal with this; it is almost impossible to not have a few days of
> chaos).  Ensuring this is not a trivial task, and it requires some buy-in
> both from the "repository" and from the developers.
>
> For Bioconductor it is even harder as the dependency graph of Bioconductor

... [show rest of quote]

--
Hervé Pagès

Program in Computational Biology
Division of Public Health Sciences
Fred Hutchinson Cancer Research Center
1100 Fairview Ave. N, M1-B514
P.O. Box 19024
Seattle, WA 98109-1024

E-mail: [hidden email]
Phone:  (206) 667-5791
Fax:    (206) 667-1319

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Jeroen Ooms.** Mar 19, 2014; 5:28pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

89 posts

In reply to this post by Joshua Ulrich

On Wed, Mar 19, 2014 at 11:50 AM, Joshua Ulrich <[hidden email]> wrote:
>
> The suggested solution is not described in the referenced article.  It
> was not suggested that it be the operating system's responsibility to
> distribute snapshots, nor was it suggested to create binary
> repositories for specific operating systems, nor was it suggested to
> freeze only a subset of CRAN packages.

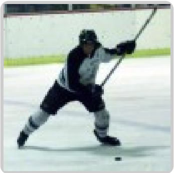
IMO this is an implementation detail. If we could all agree on a particular
set of cran packages to be used with a certain release of R, then it
doesn't matter how the 'snapshotting' gets implemented. It could be a
separate repository, or a directory on cran with symbolic links, or a page
somewhere with hyperlinks to the respective source packages. Or you can put
all packages in a big zip file, or include it in your OS distribution. You
can even distribute your entire repo on cdroms (debian style!) or do all of
the above.

The hard problem is not implementation. The hard part is that for
reproducibility to work, we need community wide conventions on which
versions of cran packages are used by a particular release of R. Local
downstream solutions are impractical, because this results in
scripts/packages that only work within your niche using this particular
snapshot. I expect that requiring every script be executed in the context
of dependencies from some particular third party repository will make
reproducibility even less common. Therefore I am trying to make a case for
a solution that would naturally improve reliability/reproducibility of R
code without any effort by the end-user.

[[alternative HTML version deleted]]

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Joshua Ulrich** Mar 19, 2014; 5:59pm Re: [RFC] A case for freezing CRAN Reply | Threaded | More



534 posts

On Wed, Mar 19, 2014 at 4:28 PM, Jeroen Ooms <[hidden email]> wrote:

> On Wed, Mar 19, 2014 at 11:50 AM, Joshua Ulrich <[hidden email]>
> wrote:
>>
>> The suggested solution is not described in the referenced article.  It
>> was not suggested that it be the operating system's responsibility to
>> distribute snapshots, nor was it suggested to create binary
>> repositories for specific operating systems, nor was it suggested to
>> freeze only a subset of CRAN packages.
>

... [show rest of quote]
So implementation isn't a problem.  The problem is that you need a way
to force people not to be able to use different package versions than
what existed at the time of each R release.  I said this in my
previous email, but you removed and did not address it: "However, you
would need to find a way to actively _prevent_ people from installing
newer versions of packages with the stable R releases."  Frankly, I
would stop using CRAN if this policy were adopted.

I suggest you go build this yourself.  You have all the code available
on CRAN, and the dates at which each package was published.  If others
who care about reproducible research find what you've built useful,
you will create the very community you want.  And you won't have to
force one single person to change their workflow.

Best,
--
Joshua Ulrich  |  about.me/joshuaulrich
FOSS Trading  |  www.fosstrading.com

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Dan Tenenbaum** Mar 19, 2014; 6:11pm Re: [RFC] A case for freezing CRAN Reply | Threaded | More

----- Original Message -----

> From: "Joshua Ulrich" <[hidden email]>
> To: "Jeroen Ooms" <[hidden email]>

96 posts

> Cc: "r-devel" <[hidden email]>
> Sent: Wednesday, March 19, 2014 2:59:53 PM
> Subject: Re: [Rd] [RFC] A case for freezing CRAN
>
> On Wed, Mar 19, 2014 at 4:28 PM, Jeroen Ooms
> <[hidden email]> wrote:
> > On Wed, Mar 19, 2014 at 11:50 AM, Joshua Ulrich

... [show rest of quote]

I don't see how the proposal forces anyone to do anything. If you have an old version of R and you still want to install newer versions of packages, you can download them from their CRAN landing page. As I understand it, the proposal only addresses what packages would be installed **by default** for a given version of R.

People would be free to override those default settings (by downloading newer packages as described above) but they should then not expect to be able to reproduce an earlier analysis since they'll have the wrong package versions. If they don't care, that's fine (provided that no other problems arise, such as the newer package depending on a feature of R that doesn't exist in the version you're running).

Dan

> I suggest you go build this yourself.  You have all the code
> available
> on CRAN, and the dates at which each package was published.  If
> others
> who care about reproducible research find what you've built useful,
> you will create the very community you want.  And you won't have to
> force one single person to change their workflow.
>
> Best,

... [show rest of quote]

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Jeroen Ooms.** Mar 19, 2014; 6:16pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

89 posts

In reply to this post by Joshua Ulrich

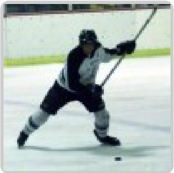
On Wed, Mar 19, 2014 at 2:59 PM, Joshua Ulrich <[hidden email]> wrote:
>
> So implementation isn't a problem.  The problem is that you need a way
> to force people not to be able to use different package versions than
> what existed at the time of each R release.  I said this in my
> previous email, but you removed and did not address it: "However, you
> would need to find a way to actively _prevent_ people from installing
> newer versions of packages with the stable R releases."  Frankly, I
> would stop using CRAN if this policy were adopted.

I am not proposing to "force" anything to anyone, those are your words. Please read the proposal more carefully before derailing the discussion. Below *verbatim* a section from the paper:

To fully make the transition to a staged CRAN, the default behavior of the package manager must be modified to download packages from the stable branch of the current version of R, rather than the latest development release. As such, all users on a given version of R will be using the same version of each CRAN package, regardless on when it was installed. The user could still be given an option to try and install the development version from the unstable branch, for example by adding an additional parameter to install.packages named devel=TRUE. However when installing an unstable package, it must be flagged, and the user must be warned that this version is not properly tested and might not be working as expected. Furthermore, when loading this package a warning could be shown with the version number so that it is also obvious from the output that results were produced using a non-standard version of the contributed package. Finally, users that would always like to use the very latest versions of all packages, e.g. developers, could install the r-devel release of R. This version contains the latest commits by R Core and downloads packages from the devel branch on CRAN, but should not be used or in production or reproducible research settings.

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Joshua Ulrich** Mar 19, 2014; 6:42pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More



534 posts

On Wed, Mar 19, 2014 at 5:16 PM, Jeroen Ooms <[hidden email]> wrote:

> On Wed, Mar 19, 2014 at 2:59 PM, Joshua Ulrich <[hidden email]> wrote:
>>
>> So implementation isn't a problem.  The problem is that you need a way
>> to force people not to be able to use different package versions than
>> what existed at the time of each R release.  I said this in my
>> previous email, but you removed and did not address it: "However, you
>> would need to find a way to actively _prevent_ people from installing
>> newer versions of packages with the stable R releases."  Frankly, I
>> would stop using CRAN if this policy were adopted.
... [show rest of quote]
<snip>

Yes "force" is too strong a word.  You want a barrier (however small)
to prevent people from installing newer (or older) versions of
packages than those that correspond to a given R release.

I still think you're going to have a very hard time convincing CRAN
maintainers to take up your cause, even if you were to build support
for it.  Especially because there's nothing stopping anyone else from
doing it.

--
Joshua Ulrich  |  about.me/joshuaulrich
FOSS Trading  |  www.fosstrading.com

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

**Hervé Pagès** Mar 19, 2014; 6:57pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

452 posts

In reply to this post by Joshua Ulrich

On 03/19/2014 02:59 PM, Joshua Ulrich wrote:

> On Wed, Mar 19, 2014 at 4:28 PM, Jeroen Ooms <[hidden email]> wrote:
>> On Wed, Mar 19, 2014 at 11:50 AM, Joshua Ulrich <[hidden email]>
>> wrote:
>>>
>>> The suggested solution is not described in the referenced article.  It
>>> was not suggested that it be the operating system's responsibility to
>>> distribute snapshots, nor was it suggested to create binary
>>> repositories for specific operating systems, nor was it suggested to
>>> freeze only a subset of CRAN packages.
... [show rest of quote]

Yeah we've already heard this "do it yourself" kind of answer. Not a
very productive one honestly.

Well actually that's what we've done for the Bioconductor repositories:
we freeze the BioC packages for each version of Bioconductor. But since
this freezing doesn't happen at the CRAN level, and many BioC packages
depend on CRAN packages, the freezing is only at the surface. Would be
much better if the freezing was all the way down to the bottom of the
sea. (Note that it is already if you install binary packages only.)

Yes it's technically possible to work around this by also hosting
frozen versions of CRAN, one per version of Bioconductor, and have
biocLite() (the tool BioC users use for installing packages) point to
these frozen versions of CRAN in order to get the correct dependencies

for any given version of BioC. However we don't do that because that
would mean extra costs for us in terms of storage space and bandwidth.
And also because we believe that it would be more effective and would
ultimately benefit the entire R community (and not just the BioC
community) if this problem was addressed upstream.

H.

>
> Best,
> --
> Joshua Ulrich | about.me/joshuaulrich
> FOSS Trading | www.fosstrading.com
>
> ______________________________________________
> [hidden email] mailing list
> https://stat.ethz.ch/mailman/listinfo/r-devel
... [show rest of quote]

--
Hervé Pagès

Program in Computational Biology
Division of Public Health Sciences
Fred Hutchinson Cancer Research Center
1100 Fairview Ave. N, M1-B514
P.O. Box 19024
Seattle, WA 98109-1024

E-mail: [hidden email]
Phone: (206) 667-5791
Fax: (206) 667-1319

______________________________________________
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

« Return to R devel | 228 views

1 2 3 4

Powered by Nabble

Edit this page