[RFC] A case for freezing CRAN

Philippe Grosjean-3 Mar 21, 2014; 8:51am Re: [RFC] A case for freezing CRAN Reply | Threaded | More

8 posts

On 21 Mar 2014, at 11:08, Rainer M Krug <[hidden email]> wrote:

> Jari Oksanen <[hidden email]> writes:
>
>> On 21/03/2014, at 10:40 AM, Rainer M Krug wrote:
>>
>>>
>>>
>>> This is a long and (mainly) interesting discussion, which is fanning out
>>> in many different directions, and I think many are not that relevant to
>>> the OP's suggestion.
... [show rest of quote]

We work on these too. So far, for latest CRAN version, we have successfully installed 4999 packages among the 5321 CRAN package on our platform. Regarding conflicts in term of function names, around 2000 packages are clean, but the rest produce more than 11,000 pairs of conflicts (i.e., same function name in different packages). For dependency errors, look at the cited references earlier. It is strange that a large portion of R CMD check errors on CRAN occur and disappear *without any version update* of a package or any of its direct or indirect dependencies! That is, a fraction of errors or warnings seem to appear and disappear without any code update. We have traced back some of these to interaction with the net (e.g., example or vignette downloading data from a server and the server may be sometimes unavailable). So, yes, a complex and difficult topic.

> Breakage of CRAN packages is a problem, to which I can not comment
> much. I have no idea how this could be saved unless one introduces more
> checks, which nobody wants. CRAN is a (more or less) open repository for
> packages written by engineers / programmers but also scientists of other
> fields - and that is the strength of CRAN - a central repository to find
> packages which conform to a minimal standard and format.
>
>>
>> Still a few words about reproducibility of scripts: this can be hardly
... [show rest of quote]

This would be a first step. Then, people would have to learn how to use, say, Sweave, in order to ensure reproducibility. This begins to be enforced by journal editors or publishers (JSS, or Elsevier comes to mind).

Best,

Philippe

> Absolutely - and I am also a sloppy scientists - I put my code online,
> but hope that not many people ask me later about it.
>
> Cheers,
>
> Rainer
>
>>
>> Cheers, Jari Oksanen
... [show rest of quote]

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

Karl Forner Mar 21, 2014; 12:08pm **Fwd: [RFC] A case for freezing CRAN** Reply | Threaded | More

47 posts

In reply to this post by Jeroen Ooms.

Interesting and strategic topic indeed.

One other point is that reproducibility (and backwards compatibility) is also very important in the industry. To get acceptance it can really help if you can easily reproduce results.

Concerning the arguments that I read in this discussion:

- "do it yourself"
The point is to discuss to find the best way for the community, and thinking collectively about this general problems can never hurt.
Once a consensus is reached we can think about the resources.

- "don't think the effort is worth it, instead install a specific version of package" + "new sessionInfoPlus()":
This could work, meaning achieving the same result, but not at the same price for users, because it would require each script writer to include its sessionInfo(), to store them along the scripts in repositories. And prior to running the scripts, you would have to install the snapshot of packages, not mentioning install problems and so on.

- "versions automatically at package build time (n DESCRIPTION)":
does not really solve the problems, because if package A is submitted with dependency B-1.0 and package C with dependency B-2 and do you do ?

- "exact deps versions":
will put a lot of burden of the developer.

- "I do not want to wait a year to get a new (or updated package)", "access to bug fixes":

Installed packages are already setup as libraries. By default you have the library inside the R installation, that contains base packages + those installed by install.packages() if you have the proper permissions, the personal library otherwise.
Why not organizing these libraries so that:
- normal CRAN versions associated with the R version gets installed along the base packages
- "critical updates", meaning important bugs found in normal CRAN versions installed in the critical/ library
- additional packages and updated package in another library.
This way, using the existing .libPaths() mechanism, or equivalently the lib.loc option of library, one could easily switch between the library that will ensure full compatibility and reproducibility with the R version, or add critical updates, or use the newer or updated packages.

- new use case.
Here in Quartz bio we have two architectures, so two R installations for

each R version. It is quite cumbersome to keep them consistent because the installed version depends on the moment you perform the install.packages().

So I second the Jeroen proposal to have a snapshot of packages versions tied to a given R version, well tested altogether. This implies as stated by Herve to keep all package source versions, and will solve the bioC reproducibility issue.

Best,
Karl Forner

On Tue, Mar 18, 2014 at 9:24 PM, Jeroen Ooms <[hidden email]>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]

[[alternative HTML version deleted]]

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

Tom Short-2 Mar 21, 2014; 12:16pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

38 posts

In reply to this post by Jeroen Ooms.

For me, the most important aspect is being able to reproduce my own work. Some other tools offer interesting approaches to managing packages:

* NPM -- The Node Package Manager for Node.js loads a local copy of all packages and dependencies. This helps ensure reproducibility and avoids dependency issues. Different projects in different directories can then use different package versions.

* Julia -- Julia's package manager is based on git, so users should have a local copy of all package versions they've used. Theoretically, you could use separate git repos for different projects, and merge as desired.

I've thought about putting my local R library into a git repository. Then, I could clone that into a project directory and use .libPaths(".Rlibrary") in a .Rprofile file to set the library directory to the clone. In addition to handling package versions, this might be nice for installing packages that are rarely used (my library directory tends to get cluttered if I start trying out packages). Another addition could be a local script that starts a specific version of R.

For now, I don't have much incentive to do this. For the packages that I use, R's been pretty good to me with backwards compatibility.

I do like the idea of a CRAN mirror that's under version control.

On Tue, Mar 18, 2014 at 4:24 PM, Jeroen Ooms <[hidden email]> 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

Gábor Csárdi Mar 21, 2014; 12:16pm **Re: Fwd: [RFC] A case for freezing CRAN** Reply | Threaded | More


97 posts

In reply to this post by Karl Forner

I agree with most of what you wrote, with one exception:

On Fri, Mar 21, 2014 at 12:08 PM, Karl Forner <[hidden email]> wrote:
[...]

> - "exact deps versions":
> will put a lot of burden of the developer.
>

Not really, in my opinion, if you have the proper tools. Most likely when you develop any given version of your package you'll use certain versions of other packages, probably the most recent at that time.

If there is a build tool that just puts these version numbers into the DESCRIPTION file, you don't need to do anything extra.

In fact, it is easier for the developer, because if you work on your release for a month, at the end you don't have to make sure that your package works with packages that were updated in the meanwhile.

Gabor

[...]

[[alternative HTML version deleted]]

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

Karl Forner Mar 21, 2014; 12:40pm **Re: Fwd: [RFC] A case for freezing CRAN** Reply | Threaded | More


47 posts

> On Fri, Mar 21, 2014 at 12:08 PM, Karl Forner <[hidden email]>wrote:
> [...]
>
> - "exact deps versions":
>> will put a lot of burden of the developer.
>>
>
> Not really, in my opinion, if you have the proper tools. Most likely when
> you develop any given version of your package you'll use certain versions
> of other packages, probably the most recent at that time.
... [show rest of quote]

I of course assumed that this part was automatic.

>
> In fact, it is easier for the developer, because if you work on your
> release for a month, at the end you don't have to make sure that your

> package works with packages that were updated in the meanwhile.
>

Hmm, what if your package depends on packages A and B, and that A depends on C v1.0 and B depends on C v1.1 ? This is just an example but I imagine that will lead to a lot of complexities.

>
> Gabor
>
> [...]
>

[[alternative HTML version deleted]]

Gábor Csárdi Mar 21, 2014; 1:27pm Re: Fwd: [RFC] A case for freezing CRAN Reply | Threaded | More

97 posts

On Fri, Mar 21, 2014 at 12:40 PM, Karl Forner <[hidden email]> wrote:
[...]

> Hmm, what if your package depends on packages A and B, and that A depends
> on C v1.0 and B depends on C v1.1 ? This is just an example but I imagine
> that will lead to a lot of complexities.
>

You'll have to be able to load (but not attach, of course!) multiple versions of the same package at the same time. The search paths are set up so that A imports v1.0 of C, B imports v1.1. This is possible to support with R's namespaces and imports mechanisms, I believe.

It requires quite some work, though, so I am obviously not saying to switch to it tomorrow. Having a CRAN-devel seems simpler.

Gabor

[[alternative HTML version deleted]]

Karl Forner Mar 21, 2014; 1:38pm Re: Fwd: [RFC] A case for freezing CRAN Reply | Threaded | More

47 posts

On Fri, Mar 21, 2014 at 6:27 PM, Gábor Csárdi <[hidden email]>wrote:

> On Fri, Mar 21, 2014 at 12:40 PM, Karl Forner <[hidden email]>wrote:
> [...]
>
>> Hmm, what if your package depends on packages A and B, and that A depends
>> on C v1.0 and B depends on C v1.1 ? This is just an example but I imagine
>> that will lead to a lot of complexities.
>>
>
> You'll have to be able to load (but not attach, of course!) multiple
... [show rest of quote]
not really: I think there are still cases (unfortunately) where you have to use depends, e.g. when defining S4 methods for classes implemented in other packages.
But my point is that you would need really really smart tools, AND to be able to install precise versions of packages.

> It requires quite some work, though, so I am obviously not saying to
> switch to it tomorrow. Having a CRAN-devel seems simpler.

>

Indeed.

[[alternative HTML version deleted]]

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

Gábor Csárdi Mar 21, 2014; 1:51pm **Re: Fwd: [RFC] A case for freezing CRAN** Reply | Threaded | More


97 posts

On Fri, Mar 21, 2014 at 1:38 PM, Karl Forner <[hidden email]> wrote:

> On Fri, Mar 21, 2014 at 6:27 PM, Gábor Csárdi <[hidden email]>wrote:
>
>> On Fri, Mar 21, 2014 at 12:40 PM, Karl Forner <[hidden email]>wrote:
>> [...]
>>
>>> Hmm, what if your package depends on packages A and B, and that A
>>> depends on C v1.0 and B depends on C v1.1 ? This is just an example but I
>>> imagine that will lead to a lot of complexities.
>>>
... [show rest of quote]
Yes, but these are some things that can be set as goals, and then we can work towards them slowly, keeping compatibility.

I would also emphasize that there is no need to (re)invent the wheel here, there are working models of software distributions, both for versioned dependencies (NPM), and having stable and devel repositories (almost all Linux distributions, BioC, etc.). Most of these are much bigger than CRAN, in terms of number of packages and volume.

Gabor

[[alternative HTML version deleted]]

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

Martin Maechler Mar 24, 2014; 6:28am **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More


1596 posts

In reply to this post by Hervé Pagès

>>>>> Hervé Pagès <[hidden email]>
>>>>> on Thu, 20 Mar 2014 15:23:57 -0700 writes:

> On 03/20/2014 01:28 PM, Ted Byers wrote:
>> On Thu, Mar 20, 2014 at 3:14 PM, Hervé Pagès
>> <[hidden email] <mailto:[hidden email]>> wrote:
>>
>> On 03/20/2014 03:52 AM, Duncan Murdoch wrote:
>>
>> On 14-03-20 2:15 AM, Dan Tenenbaum wrote:
>>
>>
>>
>>
>> ----- Original Message -----
>>
>> From: "David Winsemius" <[hidden email]
>> <mailto:[hidden email]>> To: "Jeroen Ooms"
>> <[hidden email]
>> <mailto:[hidden email]>> Cc: "r-devel"
>> <[hidden email] <mailto:[hidden email]>>
>> Sent: Wednesday, March 19, 2014 11:03:32 PM Subject: Re:
>> [Rd] [RFC] A case for freezing CRAN
>>
>>

>> On Mar 19, 2014, at 7:45 PM, Jeroen Ooms wrote:
>>
>> On Wed, Mar 19, 2014 at 6:55 PM, Michael Weylandt
>> <[hidden email]
>> <mailto:[hidden email]>> wrote:
>>
>> Reading this thread again, is it a fair summary of your
>> position to say "reproducibility by default is more
>> important than giving users access to the newest bug
>> fixes and features by default?" It's certainly arguable,
>> but I'm not sure I'm convinced: I'd imagine that the
>> ratio of new work being done vs reproductions is rather
>> high and the current setup optimizes for that already.
>>
>>
>> I think that separating development from released
>> branches can give us both reliability/reproducibility
>> (stable branch) as well as new features (unstable
>> branch). The user gets to pick (and you can pick
>> both!). The same is true for r-base: when using a
>> 'released' version you get 'stable' base packages that
>> are up to 12 months old. If you want to have the latest
>> stuff you download a nightly build of r-devel. For
>> regular users and reproducible research it is recommended
>> to use the stable branch. However if you are a developer
>> (e.g. package author) you might want to
>> develop/test/check your work with the latest r-devel.
>>
>> I think that extending the R release cycle to CRAN would
>> result both in more stable released versions of R, as
>> well as more freedom for package authors to implement
>> rigorous change in the unstable branch. When writing a
>> script that is part of a production pipeline, or sweave
>> paper that should be reproducible 10 years from now, or a
>> book on using R, you use stable version of R, which is
>> guaranteed to behave the same over time. However when
>> developing packages that should be compatible with the
>> upcoming release of R, you use r-devel which has the
>> latest versions of other CRAN and base packages.
>>
>>
>>
>> As I remember ... The example demonstrating the need for
>> this was an XML package that cause an extract from a
>> website where the headers were misinterpreted as data in
>> one version of pkg:XML and not in another. That seems
>> fairly unconvincing. Data cleaning and validation is a
>> basic task of data analysis. It also seems excessive to
>> assert that it is the responsibility of CRAN to maintain
>> a synced binary archive that will be available in ten
>> years.
>>
>>
>>
>> CRAN already does this, the bin/windows/contrib directory
>> has subdirectories going back to 1.7, with packages dated
>> October 2004. I don't see why it is burdensome to
>> continue to archive these. It would be nice if source
>> versions had a similar archive.
>>
>>
>> The bin/windows/contrib directories are updated every day
>> for active R versions. It's only when Uwe decides that a
>> version is no longer worth active support that he stops
>> doing updates, and it "freezes". A consequence of this
>> is that the snapshots preserved in those older
>> directories are unlikely to match what someone who keeps
>> up to date with R releases is using. Their purpose is to
>> make sure that those older versions aren't completely
>> useless, but they aren't what Jeroen was asking for.
>>
>>
>> But it is almost completely useless from a

>> reproducibility point of view to get random package
>> versions. For example if some people try to use R-2.13.2
>> today to reproduce an analysis that was published 2 years
>> ago, they'll get Matrix 1.0-4 on Windows, Matrix 1.0-3 on
>> Mac, and Matrix 1.1-2-2 on Unix. And none of them of
>> course is what was used by the authors of the paper (they
>> used Matrix 1.0-1, which is what was current when they
>> ran their analysis).
>>
>> Initially this discussion brought back nightmares of DLL
>> hell on Windows. Those as ancient as I will remember
>> that well. But now, the focus seems to be on
>> reproducibility, but with what strikes me as a seriously
>> flawed notion of what reproducibility means.
>>
>> Herve Pages mentions the risk of irreproducibility across
>> three minor revisions of version 1.0 of Matrix.

> If you use R-2.13.2, you get Matrix 1.1-2-2 on
> Linux.

No way! Matrix 1.1-2-2 has Depends: R (>= 2.15.2)

> AFAIK this is the most recent version of Matrix,
> aimed to be compatible with the most current version of R
> (i.e. R 3.0.3). However, it has never been tested with R-2.13.2.

Exactly. And for this reason, I have adopted to keep
Depends: R (>= ...)
in Matrix and partly, in other packages I maintain.

Doing so does prevent users of old versions of R to get new features, and even more importantly, get the latest (few, of course ! ;-) bug-fixes for Matrix.

But apart from this short note.
I'm very sympathetic with optionally providing easier (not "easy") ways of setting up old versions of R and packages, where users can pretty quickly use the printed (unfortunately, for now) output of sessionInfo(), to reinstall
1) the version of R
2) an install.packages() call which tries (!) to get the corresponding packages (in their correct version) from CRAN (including ./Archive/ !)..

similarly to what Duncan Murdoch has agreed to.

> I'm not saying that it should, that would be a
> big waste of resources of course. All I'm saying it that
> it doesn't make sense to serve by default a version that
> is known to be incompatible with the version of R being
> used. It's very likely to not even install properly.

[.....]

> Also note that back in October 2011, people using R-2.13.2
> would get e.g. ape 2.7-3 on Linux, Windows and
> Mac. Wouldn't it make sense that people using R-2.13.2
> today get the same? Why would anybody use R-2.13.2 today
> if it's not to run again some code that was written and
> used two years ago to obtain some important results?

I also tend to agree that it would be great if someone (Karl Millar -> Google ?) would setup a good time-stamping system for CRAN {and Bioconductor and Omegahat and ..?} packages.
Ideally that system would work by *using* the CRAN (and ..) infrastructure.

> Cheers, H.

I'm still unsure if I should agree with you (Hervé) that some freezing / "data base of package timestamps" should

happen on-CRAN in addition.

Martin

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

Gábor Csárdi Mar 24, 2014; 9:43am **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

97 posts

FWIW, I am mirroring CRAN at github now, here:
https://github.com/cran

One can install specific package versions using the devtools package:
library(devtools)
install_github("cran/<package>@<version>")

In addition, one can also install versions based on the R version, e.g.:
install_github("cran/<package>@R-2.15.3")
installs the version that was on CRAN when R-2.15.3 was released.

This is not very convenient yet, because the dependencies should be installed based on the R versions as well. This is in the works.

This is an experiment, and I am not yet committed to maintaining it in the long run. We'll see how it works and if it has the potential to be useful.

Plans for features:
- convenient install of packages from CRAN "snapshots", with all dependencies coming from the same snapshot.
- web page with package search, summaries, etc.
- binaries

Help is welcome, especially advice and feedback:
https://github.com/metacran/tools/issues

Best,
Gabor

[[alternative HTML version deleted]]

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

« Return to R devel | 231 views

1 2 3 4

Powered by Nabble

Edit this page