



R2U

CRAN AS UBUNTU BINARIES

Dirk Eddebuettel

Chicago R User Group

15 Sep 2022

https://dirk.eddebuettel.com/papers/chirug_sep2022_r2u.pdf

WHO AM I ?

[tile]DB

Products ▾

Blog

Documentation

Community ▾

Log in

Sign up

Data management made universal

Manage any data as multi-dimensional arrays and
access with any tool at global scale.

Get started

Request a demo



(Adjunct) Clinical Professor, University of Illinois

- teaching [STAT447 'Data Science Programming Methods'](#)

Open Source Work

- [Debian](#) developer since 1995, currently maintaining about 185 [packages](#)
- R package author since 2003, author or maintainer of well over 60 [CRAN packages](#)
- R Foundation Board Member; JSS Associate Editor
- Rocker Project co-founder: Docker for R, including official 'r-base' image

STATUS QUO: CRAN AND BINARIES



Windows: Child's Play!

macOS: Elegant if you
do as you're told



Linux: Ah, yes, ...,
packages would be
nice to have



CRAN BINARIES ACROSS OPERATING SYSTEMS

- Windows** works pretty well for **binaries** (but building from *source* can be a pain having to first install **Rtools**, getting other source dependencies, lack of package manager so some flee for Conda ...)
- macOS** works pretty well for **binaries** from what one hears (modulo edge uses with OpenMP, or Fortran, or at times building from **source**, some use **brew** or Conda)
- Linux** Tower of Babel: some distros have some **binary** packages of some versions, but users do not always know, installation from sources works “for those who know”, can be time confusing, various tricks (**ccache**, **c2d4u**, ...), some flee for Conda

Today's talk is about getting the Linux use case to the ease of the prior two

R2U

PRIOR WORK ON CRAN INTO DEBIAN / UBUNTU

- cran2deb v1** First fully automated conversion of CRAN packages into Debian binaries (via Perl) in early 2000s by Albrecht Gebhard; building on this David Vernazobres, Albrecht Gebhard, Stefan Moeller, Dirk Eddelbuettel had working system with a few thousand binaries (see useR! 2007 talk)
- cran2deb v2** Excellent GSoCode work by Charles Blundell with a full R rewrite, ~ 6k or 7k packages, I ran it for a ~ year til machine died (see useR! 2009 talk)
- debian-r** During his PhD studies, Don Armstrong did full CRAN and BioC builds til his machine died (2015 personal reference)
- c2d4u** Extending the cran2deb work, Michael Rutter with new approach using Launchpad, still active after 10+ years, now ~ 5k packages (useR! 2011 talk)
- r-builders** R Consortium Grant for Michael Rutter, Don Armstrong, Gabor Csardi, Dirk Eddelbuettel: Grant received, later returned as no liftoff

Conclusions

- Building all of CRAN from source is *hard* and *a lot of work*
- It can be done (c.f. Don Armstrong), and *it is being done*
 - Iñaki Ucar for Fedora (leaning on Fedora build infra)
 - Detlef Steuer for OpenSUSE (leaning on OBS)
 - Michael Rutter for Ubuntu, at least partially (leaning on Launchpad)
- It seems to work better if we can rely on systems that build packages
- Question: How do we fill the gap of a *complete* Ubuntu (and/or Debian) solution?

SOME BACKGROUND ASIDES

ASIDE: WHAT IS A .DEB PACKAGE?

From the *outside* via

`--status | -s:`

Shows there is metadata.

Package name, version,
dependencies, ...

Command above queries an *installed* package.
For a given `.deb` file use `dpkg -f *.deb`

```
edd@rob:~$ dpkg -s hello
Package: hello
Status: install ok installed
Priority: optional
Section: devel
Installed-Size: 108
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: amd64
Version: 2.10-2ubuntu4
Replaces: hello-debhelper (< 2.9), hello-traditional
Depends: libc6 (>= 2.34)
Breaks: hello-debhelper (< 2.9)
Conflicts: hello-traditional
Description: example package based on GNU hello
 The GNU hello program produces a familiar, friendly greeting. It
 allows non-programmers to use a classic computer science tool which
 would otherwise be unavailable to them.
.
 Seriously, though: this is an example of how to do a Debian package.
 It is the Debian version of the GNU Project's 'hello world' program
 (which is itself an example for the GNU Project).
Homepage: http://www.gnu.org/software/hello/
Original-Maintainer: Santiago Vila <sanvila@debian.org>
edd@rob:~$
```

ASIDE: WHAT IS A .DEB PACKAGE?

From the *inside* via

`--listfile | -L:`

Binary, documentation,
help

Organized in a tree
relative to a common root

Command above queries an *installed* package.

For a given `.deb` file use `dpkg -c *.deb`

```
edd@rob:~$ dpkg -L hello
/.
/usr
/usr/bin
/usr/bin/hello
/usr/share
/usr/share/doc
/usr/share/doc/hello
/usr/share/doc/hello/NEWS.gz
/usr/share/doc/hello/changelog.Debian.gz
/usr/share/doc/hello/copyright
/usr/share/info
/usr/share/info/hello.info.gz
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/hello.1.gz
edd@rob:~$
```


ASIDE: WHAT DOES R CMD INSTALL –BINARY CREATE

Command used is

R CMD INSTALL
--binary .

```
edd@rob:~/git/rcppnloptexample(master)$ R CMD INSTALL --build .
* installing to library '/usr/local/lib/R/site-library'
* installing *source* package 'RcppNloptExample' ...
** using staged installation
** libs
ccache g++-11 -I"/usr/share/R/include" -DNDEBUG -I'/usr/local/lib/R/site-library/Rcpp/include' -I'/usr/local/lib/R/site-library/nloptr/include' -fpic -g -O3 -Wall -pipe -pedantic -Wno-ignored-attributes -c RcppExports.cpp -o RcppExports.o
ccache g++-11 -I"/usr/share/R/include" -DNDEBUG -I'/usr/local/lib/R/site-library/Rcpp/include' -I'/usr/local/lib/R/site-library/nloptr/include' -fpic -g -O3 -Wall -pipe -pedantic -Wno-ignored-attributes -c nlopt.cpp -o nlopt.o
ccache g++-11 -WL,-S -shared -L/usr/lib/R/lib -WL,-Bsymbolic-functions -flto=auto -ffat-lto-objects -flto=auto -WL,-z,relro -o RcppNloptExample.so RcppExports.o nlopt.o -L/usr/lib/R/lib -lR
installing to /usr/local/lib/R/site-library/00LOCK-rcppnloptexample/00new/RcppNloptExample/libs
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary location
** checking absolute paths in shared objects and dynamic libraries
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* creating tarball
packaged installation of 'RcppNloptExample' as 'RcppNloptExample_0.0.1_R_x86_64-pc-linux-gnu.tar.gz'
* DONE (RcppNloptExample)
edd@rob:~/git/rcppnloptexample(master)$
```

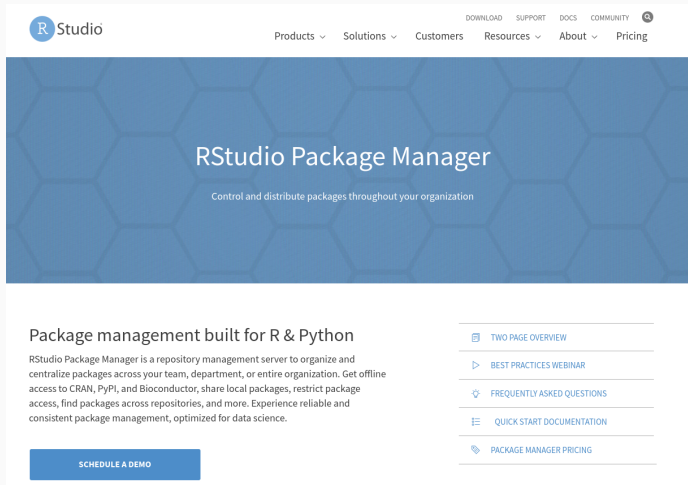
ASIDE: WHAT DOES R CMD INSTALL –BINARY CREATE

Inside is another ‘tree’ of files, including a shared library, in a tarball.

But no metadata whatsoever!

```
edd@rob:~/git/rcppnloptexample(master)$ tar tvfz RcppNloptExample_0.0.1_R_x86_64-pc-linux-gnu.tar.gz
-rw-rw-r-- edd/staff      1034 2022-08-14 16:53 RcppNloptExample/DESCRIPTION
-rw-rw-r-- edd/staff       89 2022-08-14 16:53 RcppNloptExample/INDEX
drwxrwxr-x edd/staff       0 2022-08-14 16:53 RcppNloptExample/Meta/
-rw-rw-r-- edd/staff      280 2022-08-14 16:53 RcppNloptExample/Meta/Rd.rds
-rw-rw-r-- edd/staff      122 2022-08-14 16:53 RcppNloptExample/Meta/features.rds
-rw-rw-r-- edd/staff      289 2022-08-14 16:53 RcppNloptExample/Meta/hsearch.rds
-rw-rw-r-- edd/staff      143 2022-08-14 16:53 RcppNloptExample/Meta/links.rds
-rw-rw-r-- edd/staff      333 2022-08-14 16:53 RcppNloptExample/Meta/nsInfo.rds
-rw-rw-r-- edd/staff      994 2022-08-14 16:53 RcppNloptExample/Meta/package.rds
-rw-rw-r-- edd/staff      134 2022-08-14 16:53 RcppNloptExample/NAMESPACE
-rw-rw-r-- edd/staff      385 2022-08-14 16:53 RcppNloptExample/NEWS.Rd
drwxrwxr-x edd/staff       0 2022-08-14 16:53 RcppNloptExample/R/
-rw-rw-r-- edd/staff     1058 2022-08-14 16:53 RcppNloptExample/R/RcppNloptExample
-rw-rw-r-- edd/staff     1372 2022-08-14 16:53 RcppNloptExample/R/RcppNloptExample.rdb
-rw-rw-r-- edd/staff      264 2022-08-14 16:53 RcppNloptExample/R/RcppNloptExample.rdx
drwxrwxr-x edd/staff       0 2022-08-14 16:53 RcppNloptExample/help/
-rw-rw-r-- edd/staff       46 2022-08-14 16:53 RcppNloptExample/help/AnIndex
-rw-rw-r-- edd/staff     1442 2022-08-14 16:53 RcppNloptExample/help/RcppNloptExample.rdb
-rw-rw-r-- edd/staff      171 2022-08-14 16:53 RcppNloptExample/help/RcppNloptExample.rdx
-rw-rw-r-- edd/staff      108 2022-08-14 16:53 RcppNloptExample/help/aliases.rds
-rw-rw-r-- edd/staff      135 2022-08-14 16:53 RcppNloptExample/help/paths.rds
drwxrwxr-x edd/staff       0 2022-08-14 16:53 RcppNloptExample/html/
-rw-rw-r-- edd/staff     1182 2022-08-14 16:53 RcppNloptExample/html/00Index.html
-rw-rw-r-- edd/staff     1735 2022-08-14 16:53 RcppNloptExample/html/R.css
drwxrwxr-x edd/staff       0 2022-08-14 16:53 RcppNloptExample/libs/
-rwxrwxr-x edd/staff    93784 2022-08-14 16:53 RcppNloptExample/libs/RcppNloptExample.so
edd@rob:~/git/rcppnloptexample(master)$
```

RSPM (RSTUDIO PACKAGE MANAGER)



The screenshot shows the RStudio Package Manager website. At the top, there is a navigation bar with the RStudio logo on the left and links for Products, Solutions, Customers, Resources, About, and Pricing in the center. On the right side of the navigation bar are links for Download, Support, Docs, and Community, along with a search icon. Below the navigation bar is a large blue banner with a hexagonal pattern. The banner contains the text "RStudio Package Manager" in a large, white, sans-serif font, and below it, in a smaller white font, "Control and distribute packages throughout your organization". Below the banner, the page is divided into two columns. The left column has a section titled "Package management built for R & Python" followed by a paragraph describing the tool's capabilities: "RStudio Package Manager is a repository management server to organize and centralize packages across your team, department, or entire organization. Get offline access to CRAN, PyPI, and Bioconductor, share local packages, restrict package access, find packages across repositories, and more. Experience reliable and consistent package management, optimized for data science." Below this paragraph is a blue button with the text "SCHEDULE A DEMO". The right column contains a vertical list of links, each preceded by a small icon: "TWO PAGE OVERVIEW" (document icon), "BEST PRACTICES WEBINAR" (play button icon), "FREQUENTLY ASKED QUESTIONS" (lightbulb icon), "QUICK START DOCUMENTATION" (list icon), and "PACKAGE MANAGER PRICING" (tag icon).

RStudio

Products Solutions Customers Resources About Pricing

DOWNLOAD SUPPORT DOCS COMMUNITY

RStudio Package Manager

Control and distribute packages throughout your organization

Package management built for R & Python

RStudio Package Manager is a repository management server to organize and centralize packages across your team, department, or entire organization. Get offline access to CRAN, PyPI, and Bioconductor, share local packages, restrict package access, find packages across repositories, and more. Experience reliable and consistent package management, optimized for data science.

[SCHEDULE A DEMO](#)

- [TWO PAGE OVERVIEW](#)
- [BEST PRACTICES WEBINAR](#)
- [FREQUENTLY ASKED QUESTIONS](#)
- [QUICK START DOCUMENTATION](#)
- [PACKAGE MANAGER PRICING](#)

RSPM EXAMPLE (ON MY OS + R VERSION)

```
edd@rob:~$ wget -q https://packagemanager.rstudio.com/cran/_linux__jammy/latest/src/contrib/RcppNLOptExample_0.0.1.tar.gz?r_version=4.2 -O RcppNLOptExample_0.0.1.tar.gz
edd@rob:~$ tar tvzf RcppNLOptExample_0.0.1.tar.gz
-rw-rw-r-- builder/builder 973 2022-07-12 12:57 RcppNLOptExample/DESCRIPTION
-rw-rw-r-- builder/builder 76 2022-07-12 12:57 RcppNLOptExample/INDEX
drwxrwxr-x builder/builder 0 2022-07-12 12:57 RcppNLOptExample/Meta/
-rw-rw-r-- builder/builder 278 2022-07-12 12:57 RcppNLOptExample/Meta/Rd.rds
-rw-rw-r-- builder/builder 121 2022-07-12 12:57 RcppNLOptExample/Meta/features.rds
-rw-rw-r-- builder/builder 288 2022-07-12 12:57 RcppNLOptExample/Meta/hsearch.rds
-rw-rw-r-- builder/builder 142 2022-07-12 12:57 RcppNLOptExample/Meta/links.rds
-rw-rw-r-- builder/builder 333 2022-07-12 12:57 RcppNLOptExample/Meta/nsInfo.rds
-rw-rw-r-- builder/builder 1012 2022-07-12 12:57 RcppNLOptExample/Meta/package.rds
-rw-rw-r-- builder/builder 134 2022-07-12 12:57 RcppNLOptExample/NAMESPACE
-rw-rw-r-- builder/builder 385 2022-07-12 12:57 RcppNLOptExample/NEWS.Rd
drwxrwxr-x builder/builder 0 2022-07-12 12:57 RcppNLOptExample/R/
-rw-rw-r-- builder/builder 1058 2022-07-12 12:57 RcppNLOptExample/R/RcppNLOptExample
-rw-rw-r-- builder/builder 1366 2022-07-12 12:57 RcppNLOptExample/R/RcppNLOptExample.rdb
-rw-rw-r-- builder/builder 262 2022-07-12 12:57 RcppNLOptExample/R/RcppNLOptExample.rdx
drwxrwxr-x builder/builder 0 2022-07-12 12:57 RcppNLOptExample/help/
-rw-rw-r-- builder/builder 46 2022-07-12 12:57 RcppNLOptExample/help/AnIndex
-rw-rw-r-- builder/builder 1484 2022-07-12 12:57 RcppNLOptExample/help/RcppNLOptExample.rdb
-rw-rw-r-- builder/builder 171 2022-07-12 12:57 RcppNLOptExample/help/RcppNLOptExample.rdx
-rw-rw-r-- builder/builder 107 2022-07-12 12:57 RcppNLOptExample/help/aliases.rds
-rw-rw-r-- builder/builder 156 2022-07-12 12:57 RcppNLOptExample/help/paths.rds
drwxrwxr-x builder/builder 0 2022-07-12 12:57 RcppNLOptExample/html/
-rw-rw-r-- builder/builder 1174 2022-07-12 12:57 RcppNLOptExample/html/00Index.html
-rw-rw-r-- builder/builder 1735 2022-07-12 12:57 RcppNLOptExample/html/R.css
drwxrwxr-x builder/builder 0 2022-07-12 12:57 RcppNLOptExample/libs/
-rwxrwxr-x builder/builder 1006520 2022-07-12 12:57 RcppNLOptExample/libs/RcppNLOptExample.so
edd@rob:~$
```

We can download a tar.gz

No metadata whatsoever!

So this like R CMD

INSTALL --binary !!

Positive Aspects

- Windows + macOS + various Linux distros, also multiple R versions, even Python
- ‘just the binary’ – what `R CMD INSTALL --binary` creates

Less Positive Aspects

- no (real, automatic, full) system dependencies
- no system management integration (on OSs that have it)
- some packages are source and *still* need building
- no Debian, no arm64

Overall

- not bad at all, some warts (for Linux users) notwithstanding

What makes it sing

- use build artifacts – which RSPM gives us – as input in Debian build step
- use Debian package tools for *proper* integration (given parameterization)
- “cheaply, quickly, reliably” create packages *with full system dependencies*
- including from source where needed using the standard full-build way

Enter r2u

- operational and up since May
- currently serving about over 5k binaries/day

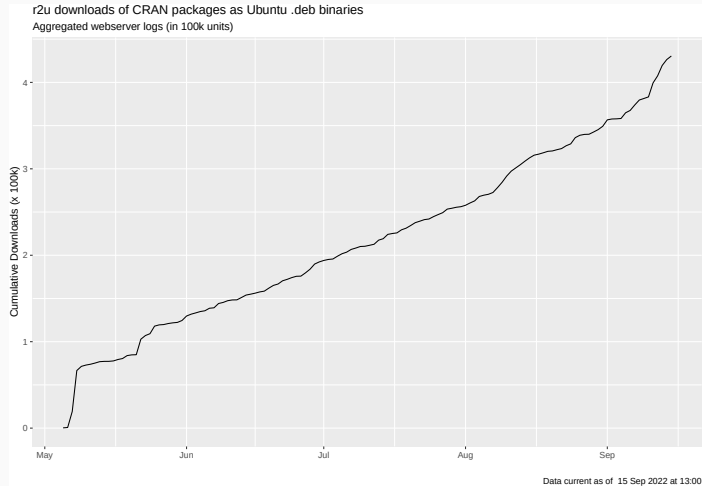
SUMMARY

Binary CRAN Packages Rock

- Now you can rock them wherever Ubuntu runs
- Your server, and if so inclined, your desktop / laptop
- Your cloud instance at AWS, GCS, Azure, ...
- Your continuous integration integration action runner
- You name it: Linux and Ubuntu are fairly universal
- r2u gives you all of CRAN at an instant with full dependency resolution

An Optional Extra Cherry on Top

- The **bspm** package by Iñaki is an extra cherry on top
- It **trace()**s the **install.packages()** function
- And connects it to **apt** with proper ‘translation’
- So **install.packages("ggplot2")** becomes ...
- ... **sudo apt install r-cran-ggplot2**
- Very convenient here as we do not have to explain **apt** naming
- But **r2u** (as a repository) works with or without it



BUT WAIT

Announcing a Mirror

- Thrilled to share that r2u.stat.illinois.edu will now serve you!
- Use <https://r2u.stat.illinois.edu/ubuntu> in the apt entry
- At Internet2 speeds and ‘big campus’ network reliability
- Big thanks to the Statistics department for hosting
- and to Rami Dass, Principal Infrastructure Engineer, ATLAS, for the setup

Thanks to

- R (package) authors for creating something wonderful in the commons
- The CRAN team for all they do making it *reliably* accessible
- Albrecht, David, Stefan, Charles, Don, Michael, ... for all the earlier work
- RSudio (soon posit) for RSPM, and Iñaki for BSPM
- Statistics at the U of Illinois Urbana-Champaign for hosting r2u
- Rami Dass for invaluable and patient help in making that happen
- Pixabay for providing the 'free use / no attribution needed' images
- My GitHub sponsors for all the coffee money