Debian Packages of R Software

Last updated by Johannes Ranke on 2017-12-07

General information

Packages for the base R system (https://packages.debian.org/search?searchon=sourcenames&keywords=r-base&exact=true) have been part of the Debian distribution since 1997 (http://metadata.ftp-master.debian.org/changelogs//main/r/r-base/r-base_3.4.3-1_changelog), thanks to Douglas Bates, and are diligently maintained by Dirk Eddelbuettel since 2001. R, as well as many add-on packages (from CRAN and others repositories) are available via the regular Debian distribution mechanisms. Hence, running

```
apt-cache search "^r-.*" | sort
```

in a shell should get you started with a list of available packages.

Debian sid (unstable)

The latest stable version of R (https://packages.debian.org/sid/r-base) and many R packages from CRAN (https://packages.debian.org/search?keywords=r-cran&searchon=names&suite=unstable) and the Bioconductor project (https://packages.debian.org/search?keywords=r-bioc&searchon=names&suite=unstable) are readily available in official Debian sid repositories.

Installation

With an appropriate entry in `/etc/apt/sources.list` (see below for Debian branches other than sid/unstable), the newest R release including recommended packages can be installed using a command sequence like

```
apt-get update
apt-get install r-base r-base-dev
```

While updating your package lists you might get a warning about a missing key, telling you that the integrity of packages can not be verified. You can ignore this if you trust the CRAN servers and continue with the installation. Otherwise, please refer to the section on secure apt below.

You only need `r-base-dev` if you want to compile R packages yourself or other software depending on R (see section below on administration and maintenance). Be aware that you may also have to install build dependencies (typically -dev packages containing headers). The list r-sig-debian (https://stat.ethz.ch/mailman/listinfo/r-sig-debian) is a good place to ask if you run into problems.

You may want to install the automatically tuned Atlas or the multi-threaded OpenBlas library in order to get higher performance for linear algebra operations

```
apt-get install libatlas3-base
```

or

```
apt-get install libopenblas-base
```

Administration and Maintenance

The R packages part of `r-base` and `r-recommended` are installed into the directory `/usr/lib/R/library`. These can be updated using usual package maintenance tools like apt-get or aptitude.

The other R packages available as precompiled Debian packages `r-cran-*` and `r-bioc-*` are installed into `/usr/lib/R/site-library`.

The command

```
apt-cache rdepends r-base-core
```

shows all packages that depend on `r-base-core`. This comprises a large number of contributed packages from CRAN and other repositories.

If you want to install R packages not provided as Debian packages, or if you want to use newer versions, you need to build them from source which requires the development package `r-base-dev` that can be installed by

```
apt-get install r-base-dev
```

This pulls in the basic requirements for compiling R packages. R packages may then be installed by the local user/admin from the CRAN source packages, typically from inside R using the

```
R> install.packages()
```

function or using

```
R CMD INSTALL
```

from a shell. If you have proper write permissions in `/usr/local/lib/R/site-library/`, and you have not set `R_LIBS_USER` manually, they will be installed there. Otherwise, you will be asked if a directory in your home directory should be created for these packages. A routine update of such locally compiled packages can be done using

```
R> update.packages(.libPaths()[1])
```

which will update the packages in the first part of your library path. You can have a look at the components of this path by

```
R> .libPaths()
```

If you would like to update R packages that have been installed via the Debian package management system which are installed somewhere under `/usr/lib/`, I would recommend to do this the Debian way using the source packages from Debian unstable.

Pathways to R Packages

In order to find packages, R looks at the variables `R_LIBS_USER` and `R_LIBS_SITE`. On Debian and Ubuntu, `R_LIBS_USER` is set in `/etc/R/Renviron` to

```
R_LIBS_USER=${R_LIBS_USER-'~/R/$platform-library/3.4'}
```

where `$platform` is something like `'x86_64-pc-linux-gnu'` and depending on the version of R installed. You can override this in `~/.Renviron`. `R_LIBS_SITE` is set in `/etc/R/Renviron` to

```
R_LIBS_SITE=${R_LIBS_SITE-'/usr/local/lib/R/site-library:/usr/lib/R/site-library:/usr/lib/R/library'}
```

This means that packages installed from within R take precedence over the ones installed via the Debian package management system if you happen to have two versions installed at the same time.

Backports on CRAN

As R and related packages are currently not part of the official Debian backports repositories like `jessie-backports` (please contact r-sig-debian (https://stat.ethz.ch/mailman/listinfo/r-sig-debian) if you would like to change this), the R versions in released or frozen Debian releases get out of date. Therefore, updates of a subset of these R related packages are provided here for such Debian releases.

Supported packages

The packages recommended by the R core team (`r-recommended`) are updated on CRAN upon each new release of R. Currently, these are:

```
r-cran-boot
r-cran-class
r-cran-cluster
r-cran-codetools
r-cran-foreign
r-cran-kernsmooth
r-cran-lattice
r-cran-mass
r-cran-matrix
r-cran-mgcv
r-cran-nlme
r-cran-nnet
r-cran-rpart
r-cran-spatial
r-cran-survival
```

Backports of the following packages are also supplied:

```
littler
r-cran-rodbc
python-rpy
python-rpy2
rkward
r-cran-jags (entailing backports of jags and r-cran-coda)
```

Supported branches

Debian stretch (stable)

For a backport of R 3.4.3 to stretch, please add something like

```
deb http://<favourite-cran-mirror>/bin/linux/debian stretch-cran34/
```

to the file `/etc/apt/sources.list` on your computer. You need to substitute `<favourite-cran-mirror>` by one of the mirror URLs listed in the mirror list (http://cran.r-project.org/mirrors.html).

Please note that R packages from the Debian stretch distribution are not compatible with R 3.4.2, as it provides r-api-3.4, while the stretch packages depend on r-api-3. See

https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=861333 (https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=861333)

for details. Also, local packages installed before the update to R 3.4.0 may not work correctly if they use calls to .C or .Fortran internally.

R packages you have compiled locally can be updated using

```
update.packages(lib.loc="/usr/local/lib/R/site-library", ask=FALSE, checkBuilt=TRUE)
```

Finally, you may have to force the installation of the backported version of `r-cran-coda` or `rkward` by using `apt-get install -t stretch-cran34` , as the version from Debian stretch has a higher version number than the backport, but is not compatible with R >3.4.2 from this repository.

Debian jessie (oldstable)

For a backport of R 3.4.3 to jessie, please add something like

```
deb http://<favourite-cran-mirror>/bin/linux/debian jessie-cran34/
```

to the file `/etc/apt/sources.list` on your computer. You need to substitute `<favourite-cran-mirror>` by one of the mirror URLs listed in the mirror list (http://cran.r-project.org/mirrors.html).

For jessie, the same notes given in the section for stretch apply, *i.e.* packages from the Debian archive are not compatible with this backport, and locally compiled packages may have to be rebuilt.

Debian wheezy (oldoldstable)

The last R release backported to wheezy and made available on CRAN is currently R 3.2.5 with the codename “Very, Very Secure Dishes”.

In order to address the security vulnerability https://security-tracker.debian.org/tracker/CVE-2016-8714 (https://security-tracker.debian.org/tracker/CVE-2016-8714), I applied the commits r71664 and r71667 to src/library/grDevices/src/devPS.c.

Backporting R 3.3.0 and later would entail backporting curl 7.28 to wheezy, which in turn would require libgnutla28-dev (which is available in wheezy-backports), linghttp2-dev (only available in jessie) and librtmp-dev >= 2.4+20131018.git (not available in wheezy).

The milestone release R 3.0.0 published on April 3, 2013 is not compatible with any R packages built on earlier versions. This means that R packages from the Debian wheezy archive do not work under R 3.x.y. This is the source line needed for R 3.x.y on Debian wheezy

```
deb <favourite-cran-mirror>/bin/linux/debian wheezy-cran3/
```

Again, you need to substitute `<favourite-cran-mirror>` by one of the mirror URLs listed in the mirror list (http://cran.r-project.org/mirrors.html).

Of course the R packages `r-cran-*` provided here are compiled using the current R version. Before installing R 3.x.y be sure you do not rely on loading R packages from Debian wheezy. If unsure, look at the debs you have installed using

```
dpkg --get-selections | grep r-cran
```

and compare with the list of packages supported here (see above, also for a procedure to upgrade packages install locally from source R packages).

This repository also contains backports of tcl8.6, tk8.6 and texinfo for wheezy, as the upstream Debian package of R >= 3.2.0 need these versions.

Please be aware that the R packages available as binary .deb packages in wheezy (with names starting with `r-cran-`) are not compatible with R versions 3.0.0 and later. Only a confined number of R packages is provided here.

Updating packages installed locally from sources, which you need to do after an upgrade from R 2.a.b to R 3.x.y, is explained below under the section administration and maintenance. What you probably need to do is

```
R> update.packages(.libPaths()[1], checkBuilt=TRUE)
```

in R in order to rebuild packages that are installed in addition to the Debian `r-cran-*` packages. For other changes in R please refer to the NEWS (http://cran.r-project.org/src/base/NEWS) file.

Secure apt

Since the release of R 3.4.0 in April 2017, the backports to stretch and jessie are signed with the current signing key of `Johannes Ranke <jranke@uni-bremen.de>` (alternative uid is `Johannes Ranke (wissenschaftlicher Berater) <johannes.ranke@jrwb.de>`) with key fingerprint `E19F 5F87 1288 99B1 92B1 A2C2 AD5F 960A 256A 04AF` . This key is signed with my old 1024 bit key with fingerprint `6212 B7B7 931C 4BB1 6280 BA13 06F9 0DE5 381B A480` that I used to sign older releases like the backport of R 3.2.5 to wheezy (oldoldstable) that is still on CRAN.

You can fetch and import the current key using

```
apt-key adv --keyserver keys.gnupg.net --recv-key 'E19F5F87128899B192B1A2C2AD5F960A256A04AF'
```

If this doesn't work, it might be due to a firewall blocking port 11371. Alternatively, you can search for the fingerprint at http://keyserver.ubuntu.com (http://keyserver.ubuntu.com), check the fingerprint and copy the key block shown when klicking on the link in the line starting with **pub** into a plain text file, named, for instance, `jranke.asc` wich you add to apt with `apt-key add jranke.asc` .

Supported Platforms

There are i386 and amd64 binaries for wheezy and later releases. Since R 3.1.0, R is fully functional on arm and armel binaries for jessie and wheezy were provided here up to R 3.4.2. Since R 3.4.2, binaries for armhf and arm64 are provided for Debian stretch, however R 3.4.3 did not build on arm64 due to this bug (https://bugs.r-project.org/bugzilla/show_bug.cgi?id=17363).

For other architectures, you can use the source packages from one of the repositories

```
deb-src <favourite-cran-mirror>/bin/linux/debian jessie-cran34/
deb-src <favourite-cran-mirror>/bin/linux/debian stretch-cran34/
```

to compile binary Debian packages.

Reporting Problems

The best place to report problems with these packages or ask R questions specific to Debian is the r-sig-debian (https://stat.ethz.ch/mailman/listinfo/r-sig-debian) mailing list. See

https://stat.ethz.ch/mailman/listinfo/r-sig-debian (https://stat.ethz.ch/mailman/listinfo/r-sig-debian)

for more information.

Backporting for Debian

Anyone interested in building Debian packages (e.g. for an unsupported release, another architecture or an old R version) can have a look at the build scripts used by the current maintainer. These can be inspected at

http://cgit.jrwb.de/r-backports (http://cgit.jrwb.de/r-backports)

or cloned from the git repository

```
git clone http://cgit.jrwb.de/r-backports
```

The scripts contain some small changes needed to compile the original Debian packages on the supported release.

Installing R-devel or a release branch from svn

The following notes are an attempt to merge the hints given in a thread on r-sig-debian started end of March 2015, and the tips given at https://developer.r-project.org/SVNtips.html (https://developer.r-project.org/SVNtips.html). I have not used this a lot and I assume you have some experience in building software on unix systems. No warranties, your mileage may vary.

First, make sure you have a source repository in your `/etc/apt/sources/list`, like

```
deb-src <favourite-cran-mirror>/bin/linux/debian jessie-cran3/
```

in order to make it easier to get the build dependencies. Then update your list of available packages and run as root

```
apt-get build-dep r-base
```

To install the build dependencies. The following commands do not need root privileges, and are safer to run as a normal user.

I keep all sources managed by subversion under `~/svn`, so I do

```
export RTOP=~/svn/R
```

Please adapt to your needs. The location of the source code for the base R system is conveniently defined as en environment variable by

```
export REPOS=https://svn.r-project.org/R
```

R-devel

We change to our RTOP directory and check out the latest revision of R-devel

```
cd $RTOP
svn co $REPOS/trunk r-devel/source
mkdir $RTOP/r-devel/build
```

The second time around we only need to

```
cd $RTOP/r-devel/source
svn up
```

Then we need to get the sources of the recommended packages

```
cd $RTOP/r-devel/source/tools
./rsync-recommended
```

We build in the separate directory created above, in order not to pollute the source code

```
cd $RTOP/r-devel/build
../source/configure
```

There are a lot of configure options you might want to use. However, we are not using the install target here, so we do not set `--prefix`. This is to avoid conflicts with the Debian packages of released versions of R. The next step builds the sources.

```
make
make check
make pdf
make info
```

Of course doing checks and making pdf and info documentation is optional. Then we use links to make R-devel and Rscript-devel available (obviously supposes write permission on /usr/local/bin).

```
cd /usr/local/bin
ln -s /$RTOP/r-devel/build/bin/R R-devel
ln -s /$RTOP/r-devel/build/bin/Rscript Rscript-devel
```

This will make the commands `R-devel` and `Rscript-devel` available. Note that the only entry on `.libPaths()` is `$RTOP/r-devel/build/library`, so this is where packages will be installed.

R-patched

After the release of a version R-x.y.0, the so-called release branch is used to introduce changes that meet the development guidelines (http://developer.r-project.org/devel-guidelines.txt).

We can build such branches (aka R-patched) using the following scheme, which is adapted from the procedure given for R-devel, but without duplicating all comments.

At the time of this writing, the release branch can be defined by

```
export RPATCHED=branches/R-3-4-branch
```

R-patched can than be built and made available by

```
cd $RTOP
svn co $REPOS/$RPATCHED r-patched/source
mkdir $RTOP/r-patched/build
cd $RTOP/r-patched/source/tools
./rsync-recommended
```

Again, we build in the separate directory created above, in order not to pollute the source code

```
cd $RTOP/r-patched/build
../source/configure
make
make check
make pdf
make info
```

Then, as privileged user:

```
cd /usr/local/bin
ln -s /$RTOP/r-patched/build/bin/R R-patched
ln -s /$RTOP/r-patched/build/bin/Rscript Rscript-patched
```

The builds can be updated by updating the sources from the repository, syncing the latest versions of the recommended packages and rebuilding.

Debian R policy

An RFC on a first public draft of 'Debian R Policy' has been posted to debian-devel in 2003 (https://lists.debian.org/debian-devel/2003/12/msg02367.html).

Acknowledgements

The Debian R packages are maintained by Dirk Eddelbuettel. The packages present on CRAN for Debian stable and oldstable are provided by Johannes Ranke. Thanks to Mathieu Basille for restructuring the README in March 2015.