

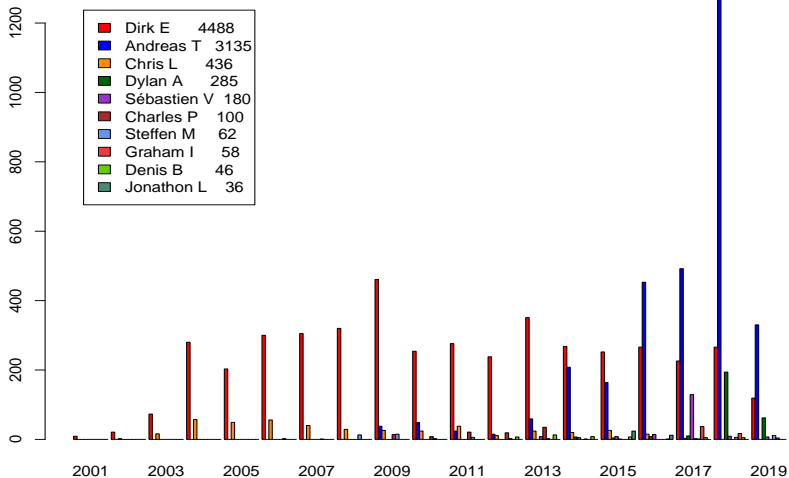
R packaging BoF

Andreas Tille

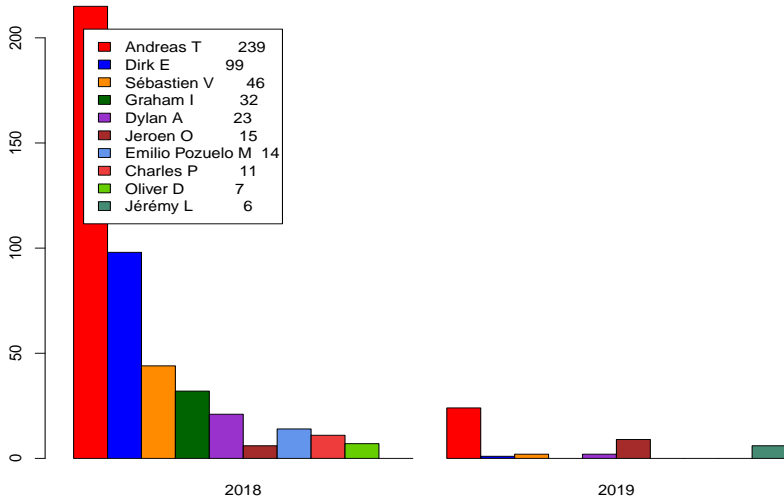
DebConf 19

Curitiba, 21. July 2019

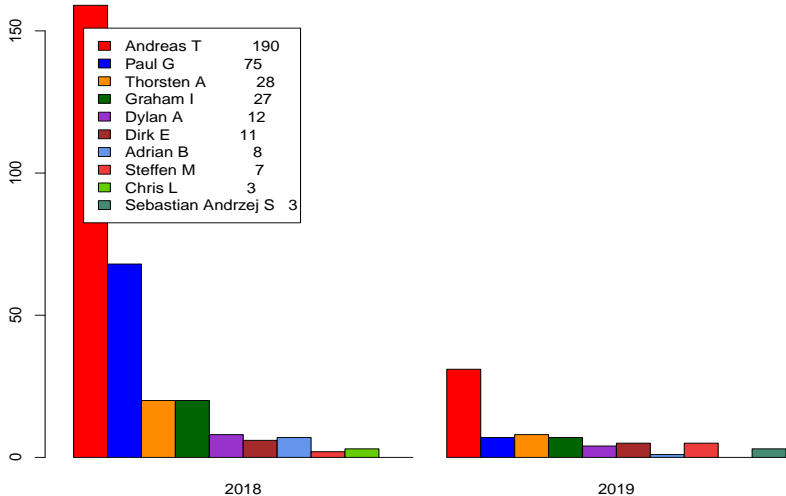
Top 10 Uploaders of R packages



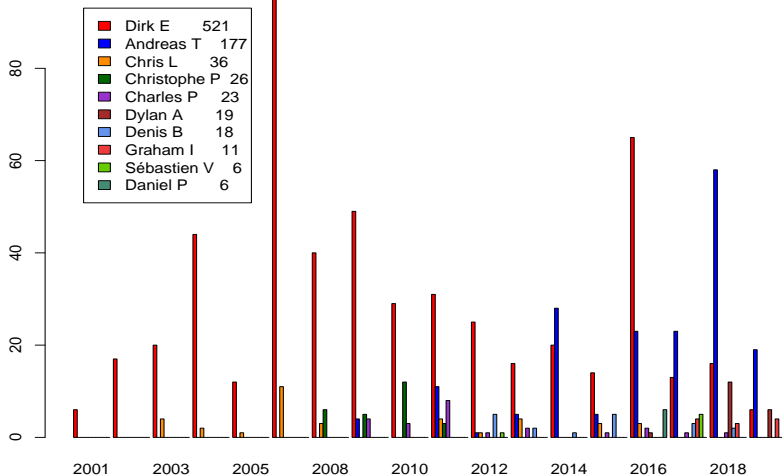
Top 10 discussion on Debian R user mailing list



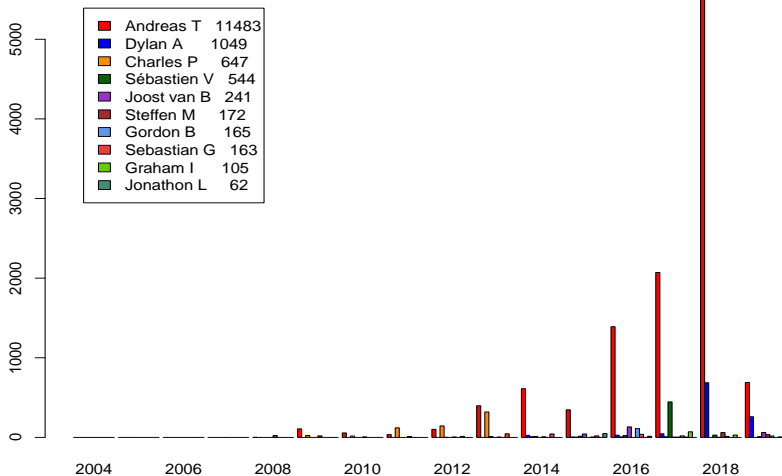
Top 10 R pkg developer mailing list



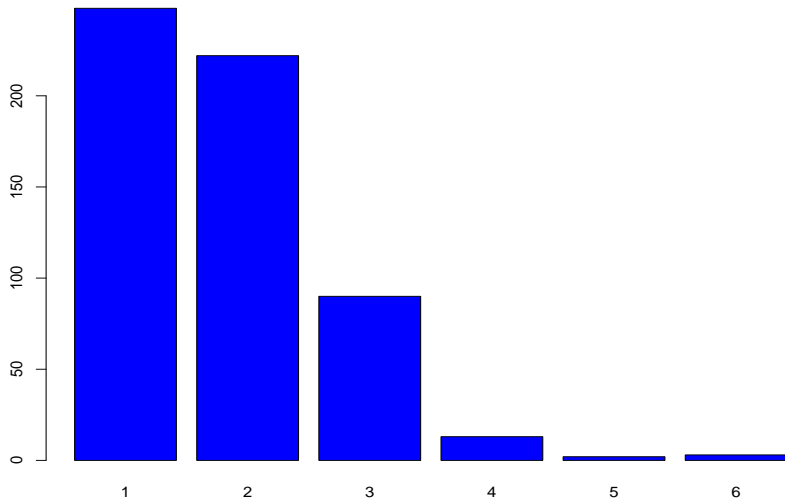
Top 10 bug hunters of R packages



Top 10 committers to r-pkg Git



Maintainer per package relation



*Waking up in the morning and realising
that somebody else has solved your
problem from yesterday*

dh-make-R / dh-update-R

- **Creating / updating R packages nearly 100% automatically**
- We can try right now if you know some R package which is not yet packaged
- `dh-make-R` creates a `debian/` dir which is nearly ready to upload
- Also not yet packaged dependencies are prepared the same way if not yet ITPed

dh-make-R / dh-update-R

- Creating / updating R packages nearly 100% automatically
- We can try right now if you know some R package which is not yet packaged
- `dh-make-R` creates a `debian/` dir which is nearly ready to upload
- Also not yet packaged dependencies are prepared the same way if not yet ITPed

dh-make-R / dh-update-R

- Creating / updating R packages nearly 100% automatically
- We can try right now if you know some R package which is not yet packaged
- `dh-make-R` creates a `debian/` dir which is nearly ready to upload
- Also not yet packaged dependencies are prepared the same way if not yet ITPed

dh-make-R / dh-update-R

- Creating / updating R packages nearly 100% automatically
- We can try right now if you know some R package which is not yet packaged
- `dh-make-R` creates a `debian/` dir which is nearly ready to upload
- Also not yet packaged dependencies are prepared the same way if not yet ITPed

routine-update

- Script to run when there is a new upstream release
- Calls `uscan` to fetch new version
- Injects new version, updates packaging (Vcs fields, Standards-Version, debhelper)
- Calls `dh-update-R` which even prepares new dependencies if needed
- Hint: In case you want to fix something inside the package without a new upstream version existing call
`routine-update -force`
(otherwise `routine-update` stops when there is no new upstream version)

routine-update

- Script to run when there is a new upstream release
- Calls `uscan` to fetch new version
- Injects new version, updates packaging (Vcs fields, Standards-Version, debhelper)
- Calls `dh-update-R` which even prepares new dependencies if needed
- Hint: In case you want to fix something inside the package without a new upstream version existing call
`routine-update -force`
(otherwise `routine-update` stops when there is no new upstream version)

routine-update

- Script to run when there is a new upstream release
- Calls `uscan` to fetch new version
- Injects new version, updates packaging (Vcs fields, Standards-Version, debhelper)
- Calls `dh-update-R` which even prepares new dependencies if needed
- Hint: In case you want to fix something inside the package without a new upstream version existing call
`routine-update -force`
(otherwise `routine-update` stops when there is no new upstream version)

routine-update

- Script to run when there is a new upstream release
- Calls `uscan` to fetch new version
- Injects new version, updates packaging (Vcs fields, Standards-Version, debhelper)
- Calls `dh-update-R` which even prepares new dependencies if needed
- Hint: In case you want to fix something inside the package without a new upstream version existing call
`routine-update -force`
(otherwise `routine-update` stops when there is no new upstream version)

routine-update

- Script to run when there is a new upstream release
- Calls `uscan` to fetch new version
- Injects new version, updates packaging (Vcs fields, Standards-Version, debhelper)
- Calls `dh-update-R` which even prepares new dependencies if needed
- Hint: In case you want to fix something inside the package without a new upstream version existing call
`routine-update -force`
(otherwise `routine-update` stops when there is no new upstream version)

prepare_missing_cran_package

- You can manually call this script that is used by *dh-r*

inject-into-salsa-git

- Stolen from Debian Med team
- Injects the repository formerly created by *prepare_missing_cran_package*

inject-into-salsa-git

- Stolen from Debian Med team
- Injects the repository formerly created by *prepare_missing_cran_package*

itp_from_debian_dir

- If you want to ITP a package this calls reportbug for you
- Option to add additional receivers of the mail

itp_from_debian_dir

- If you want to ITP a package this calls reportbug for you
- Option to add additional receivers of the mail

Information about new packages and open bugs

- https://salsa.debian.org/r-pkg-team/maintenance-utilities/blob/master/outdated_r-packages.txt
- Updated twice a day showing recent new packages and open bugs
- Drained from public UDD clone

Information about new packages and open bugs

- https://salsa.debian.org/r-pkg-team/maintenance-utilities/blob/master/outdated_r-packages.txt
- Updated twice a day showing recent new packages and open bugs
- Drained from public UDD clone

Information about new packages and open bugs

- https://salsa.debian.org/r-pkg-team/maintenance-utilities/blob/master/outdated_r-packages.txt
- Updated twice a day showing recent new packages and open bugs
- Drained from public UDD clone

CI via autopkgtests

- Testsuite: autopkgtest-pkg-r
- <https://people.debian.org/~elbrus/ci/regressions.html>

CI via autopkgtests

- Testsuite: autopkgtest-pkg-r
- <https://people.debian.org/~elbrus/ci/regressions.html>

Open issue

- `r-exclude-doc`
- Removal of docs including compressed JS
- Does anybody know how to recreate those docs??

Open issue

- `r-exclude-doc`
- **Removal of docs including compressed JS**
- Does anybody know how to recreate those docs??

Open issue

- `r-exclude-doc`
- Removal of docs including compressed JS
- Does anybody know how to recreate those docs??

gobby

`infinote://gobby.debian.org/debconf19/bof/r-pkg_bof`