R Tools for the Command Line

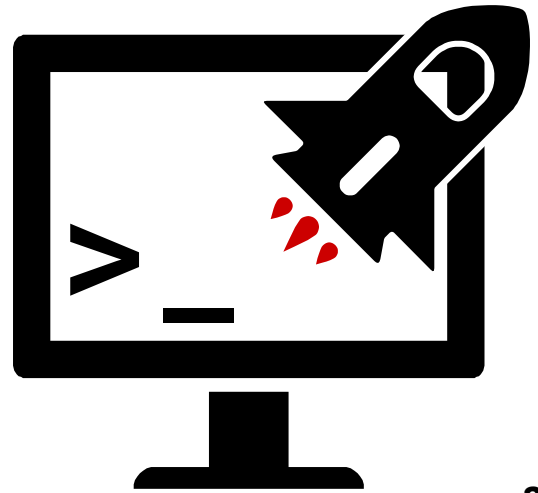> rt

R Tools for the Command Line

> rt

R Tools for the Command Line

```
> rt
```
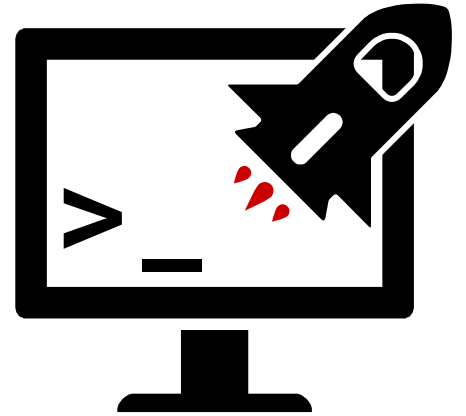
```
$ rt --help

Available Commands:
 - 'rbuild' to bundle a local package.
 - 'rcheck' to check a local package.
 - 'rclean' to remove '.[o|so]' files from a local package.
 - 'rcov' to test the coverage of a local package.
 - 'rdoc' to document a local package using roxygen.
 - 'rhub' to upload a local package to rhub service.
 - 'rinstall' to install a remote package (CRAN, GitHub etc.)
 - 'rknit' to knit a knitr document (.Rnw, .Rrst) to PDF.
 - 'rmake' to make a local package (document and install).
 - 'rpkgdown' to build static HTML documentation with pkgdown
 - 'rremove' to remove (uninstall) R packages.
 - 'rshine' to run a shiny app.
 - 'rspell' to check spelling in generated .Rd files.
 - 'rtest' to test a local package.
 - 'rupdate' to update all CRAN packages on your system
 - 'rusage' to check variable usage.
 - 'rwinbuild' to upload a local package to the winbuilder service.
```
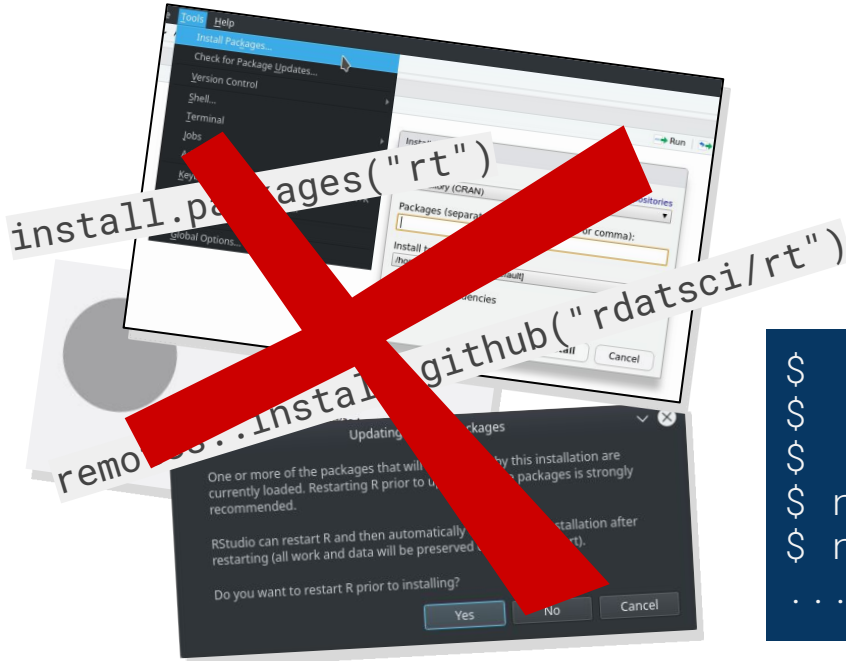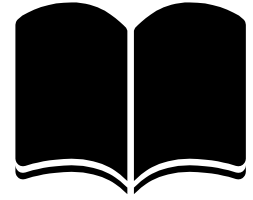
4

```
$ rt --help

Available Commands:
 - 'rbuild' to bundle a local package.
 - 'rcheck' to check a local package.
 - 'rclean' to remove '.[o|so]' files from a local package.
 - 'rcov' to test the coverage of a local package.
 - 'rdoc' to document a local package using roxygen.
 - 'rhub' to upload a local package to rhub service.
 - 'rinstall' to install a remote package (CRAN, GitHub etc.)
 - 'rknit' to knit a knitr document (.Rnw, .Rrst) to PDF.
 - 'rmake' to make a local package (document and install).
 - 'rpkgdown' to build static HTML documentation with pkgdown
 - 'rremove' to remove (uninstall) R packages.
 - 'rshine' to run a shiny app.
 - 'rspell' to check spelling in generated .Rd files.
 - 'rtest' to test a local package.
 - 'rupdate' to update all CRAN packages on your system
 - 'rusage' to check variable usage.
 - 'rwinbuild' to upload a local package to the winbuilder service.
```
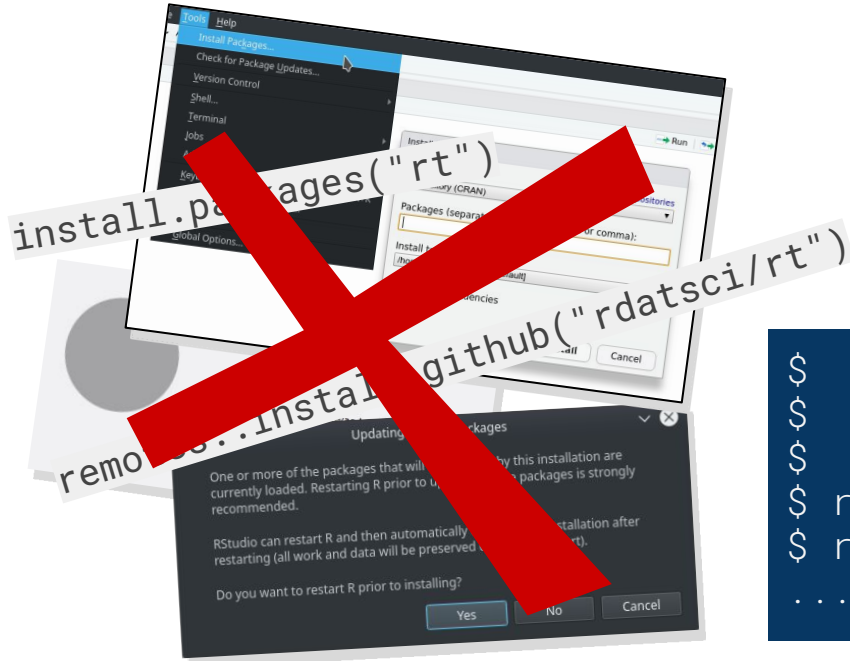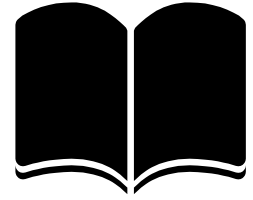

5

Why not in R (Studio)?

install.packages("rt")

remotes::install_github("rdatsci/rt")

```
$    rinstall rt
$    rinstall rdatsci/rt
$    rinstall tidyverse/dplyr@r-0.7.8
$ rinstall gitlab::(domain.com):usr/pkg
$ rinstall file.tar.gz
...
```

Why not in R (Studio)?

install.packages("rt")

remotes::install_github("rdatsci/rt")

```
$    rinstall rt
$    rinstall rdatsci/rt
$    rinstall tidyverse/dplyr@r-0.7.8
$ rinstall gitlab::(domain.com):usr/pkg
$ rinstall file.tar.gz
...
```
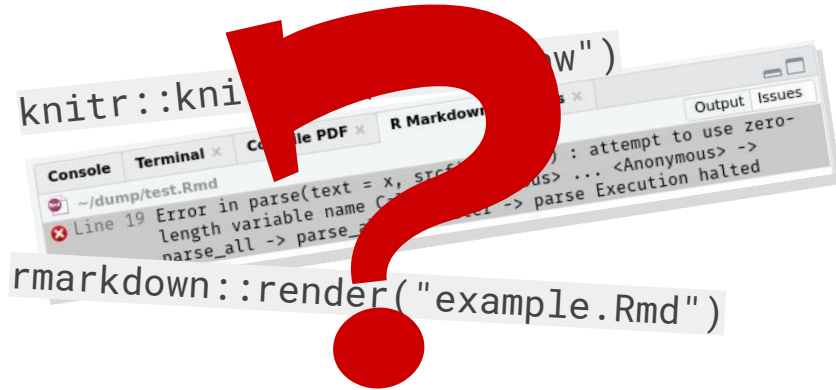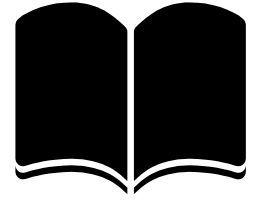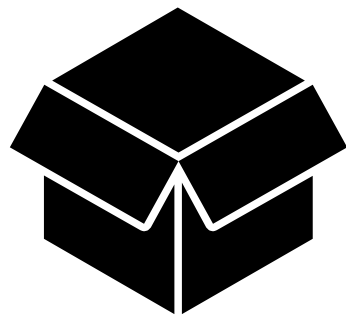
Why not in R (Studio)?

```
knitr::kni...          w")

rmarkdown::render("example.Rmd")
```

```
$    rknit main.Rnw
$    rknit example.Rmd
```

Manage R packages

```
~.config/rt/packages
```

```
## Uncomment to let rt update itself:
#rdatsci/rt
#basics
devtools
#work
mllg/batchtools
mlr-org/mlr
#git repo with subdirectories and branches/commits
https://foo.com/proj/repo.git/some_where/deep@v1.8.4
#local files
/home/user/srcs/pkg
#bioconductor
bioc::usr:pass@v1.8.4/pkg#commit123
```

edit package file via:
```
$ rt --packages
```
or using a text editor.

New packages can be put here and
installed via rupdate.

github.com/rdatsci/rt

Manage R packages

```
~.config/rt/packages
```

```
## Uncomment to let rt update itself:
#rdatsci/rt
#basics
devtools
#work
mllg/batchtools
mlr-org/mlr
#git repo with subdirectories and branches/commits
https://foo.com/proj/repo.git/some_where/deep@v1.8.4
#local files
/home/user/srcs/pkg
#bioconductor
bioc::usr:pass@v1.8.4/pkg#commit123
```
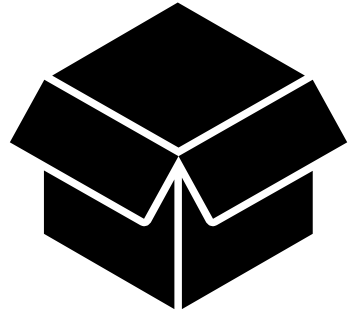
edit package file via:
```
$ rt --packages
```
or using a text editor.

New packages can be put here and installed via rupdate.

Manage R packages

```
~/.config/rt/packages

## Uncomment to let rt update itself:
#rdatsci/rt
#basics
devtools
#work
mllg/batchtools
mlr-org/mlr
#git repo with subdirectories and branches/commits
https://foo.com/proj/repo.git/some_where/deep@v1.8.4
#local files
/home/user/srcs/pkg
#bioconductor
bioc::usr:pass@v1.8.4/pkg#commit123
```
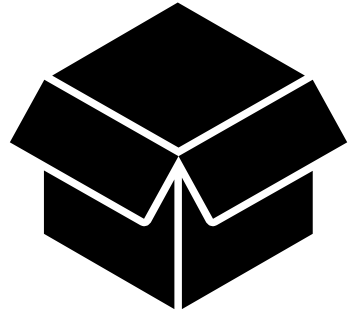
edit package file via:
```
$ rt --packages
```
or using a text editor.

New packages can be put here and installed via `rupdate`.

11

AUTO UPDATE

Automatic updates and rebuilds for new R versions!

development tools

rtest

run unit tests (supports filtering)

rcheck

run R CMD check

rhub

run check on rhub (supports various targets and changes maintainer automatically)

rwinbuild

run check on winbuilder (with changed maintainer)

rspell

spellcheck using aspell

rcov

reports codecoverage

rusage

reports unused and undeclared variables

rdoc

runs roxygenize

rmake

builds tarball

13

development tools

rtest

run unit tests (supports filtering)

rcheck

run R CMD check

rhub

run check on rhub (supports various targets and changes maintainer automatically)

rwinbuild

run check on winbuilder (with changed maintainer)

rspell

spellcheck using aspell

rcov

reports codecoverage

rusage

reports unused and undeclared variables

rdoc

runs roxygenize

rmake

builds tarball

14

Credits

https://github.com/rdatsci/rt
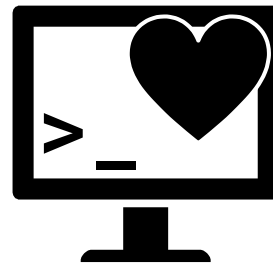
github.com/   **mllg**

Michel Lang

surmann

Dirk Surmann

jakob-r

Jakob Richter