Packages 101

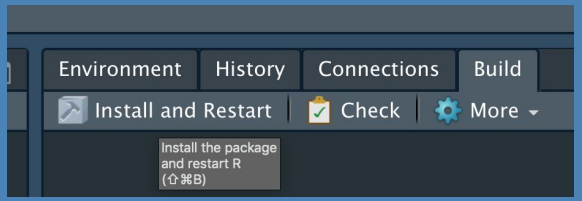
Exercise

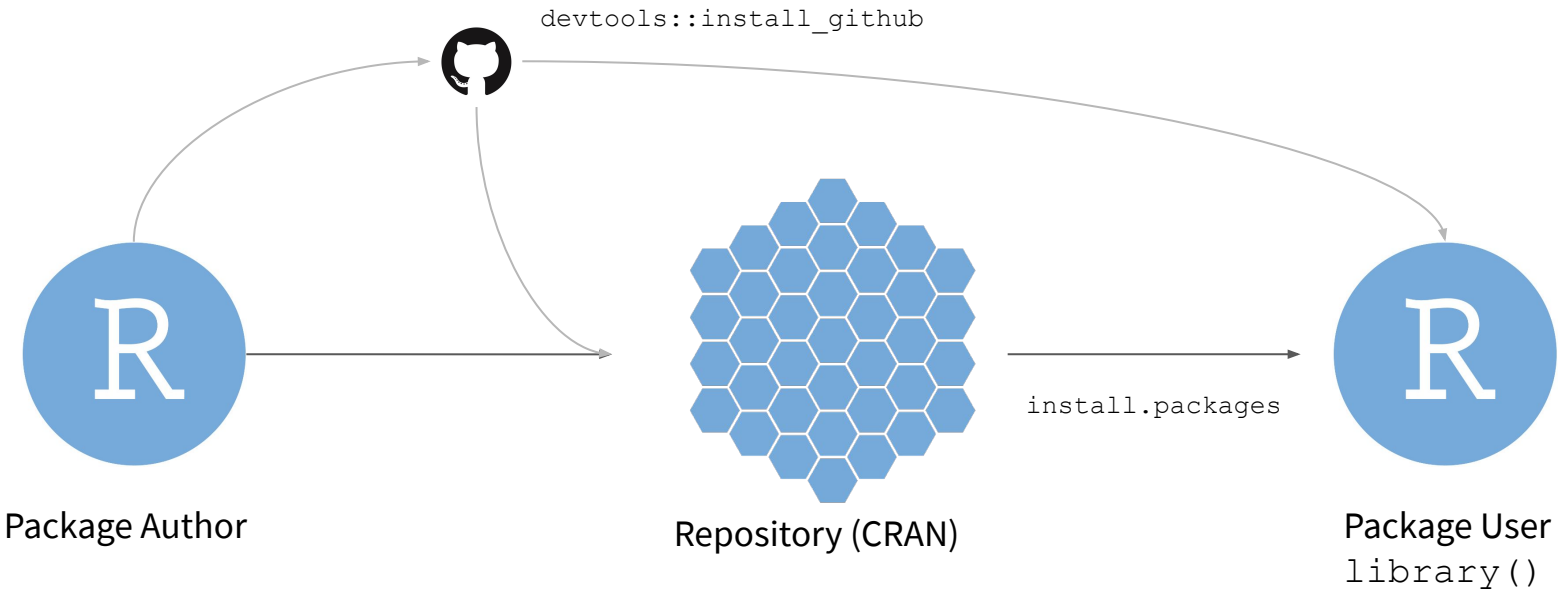
1. Run `usethis::create_package("~/helloworld")`. What does the skeleton of the package look like?

2. Run `usethis::use_r("hello")` and then copy this:
   ```
   hello <- function(name){ sprintf("Hello %s!", name)}
   ```

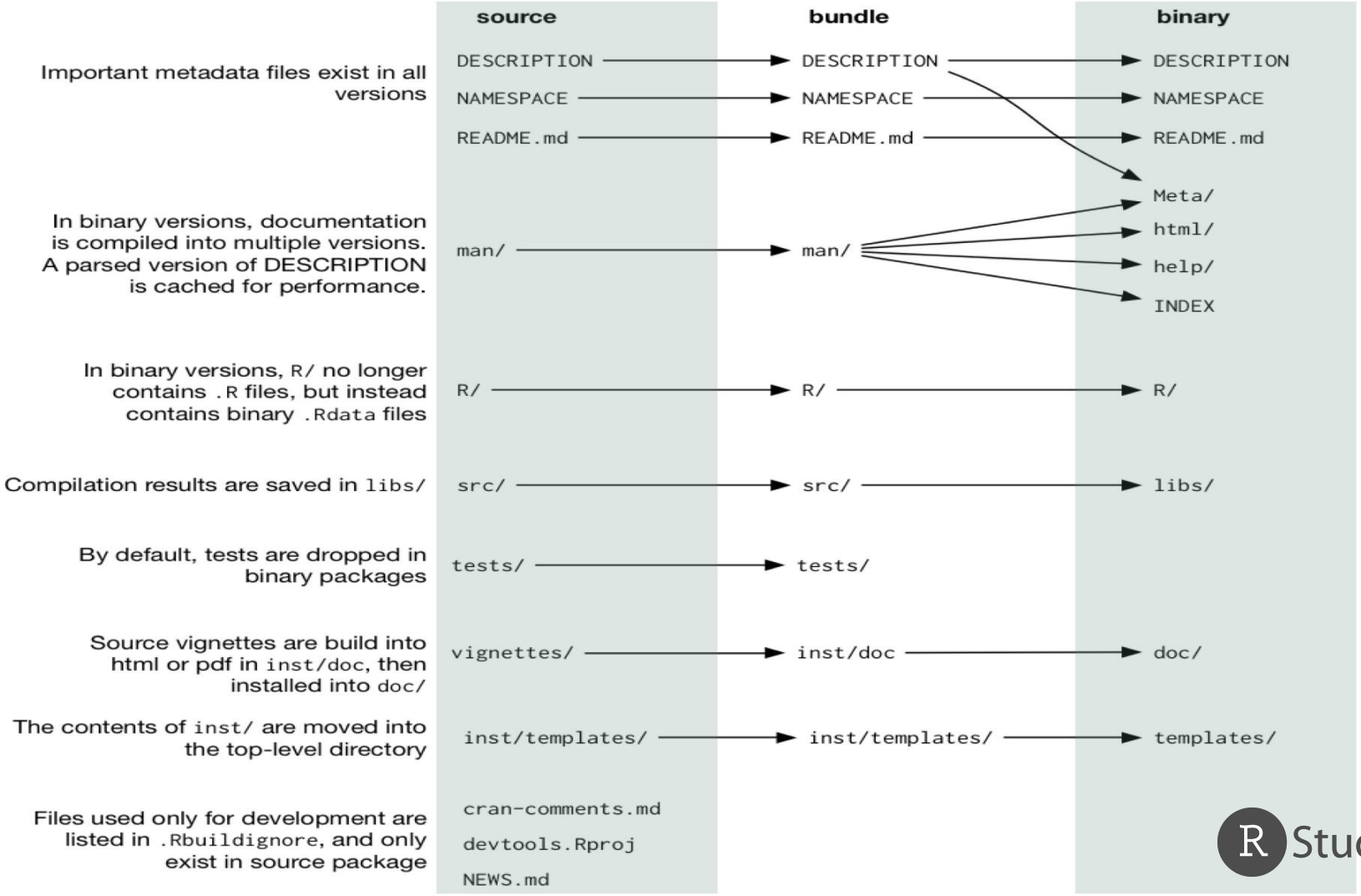
3. ```
   library(helloworld)
   hello("Sean")
   ```

4. What happened when we ran?


   ```


devtools::install_github
install.packages
Package Author
Repository (CRAN)
Package User
library()

RStudio


source
bundle
binary
Important metadata files exist in all versions
DESCRIPTION
NAMESPACE
README.md
DESCRIPTION
NAMESPACE
README.md
DESCRIPTION
NAMESPACE
README.md
In binary versions, documentation is compiled into multiple versions. A parsed version of DESCRIPTION is cached for performance.
man/
man/
Meta/
html/
help/
INDEX
In binary versions, R/ no longer contains .R files, but instead contains binary .Rdata files
R/
R/
R/
Compilation results are saved in libs/
src/
src/
libs/
By default, tests are dropped in binary packages
tests/
tests/
Source vignettes are build into html or pdf in inst/doc, then installed into doc/
vignettes/
inst/doc
doc/
The contents of inst/ are moved into the top-level directory
inst/templates/
inst/templates/
templates/
Files used only for development are listed in .Rbuildignore, and only exist in source package
cran-comments.md
devtools.Rproj
NEWS.md

Important metadata files exist in all versions

In binary versions, documentation is compiled into multiple versions. A parsed version of DESCRIPTION is cached for performance.

In binary versions, `R/` no longer contains `.R` files, but instead contains binary `.Rdata` files

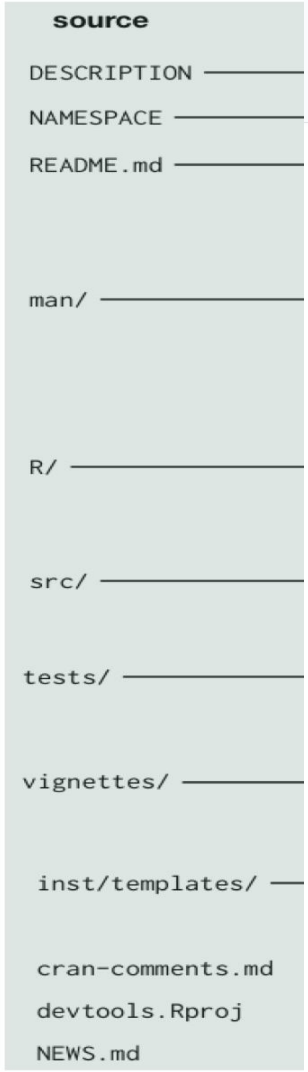
Compilation results are saved in `libs/`

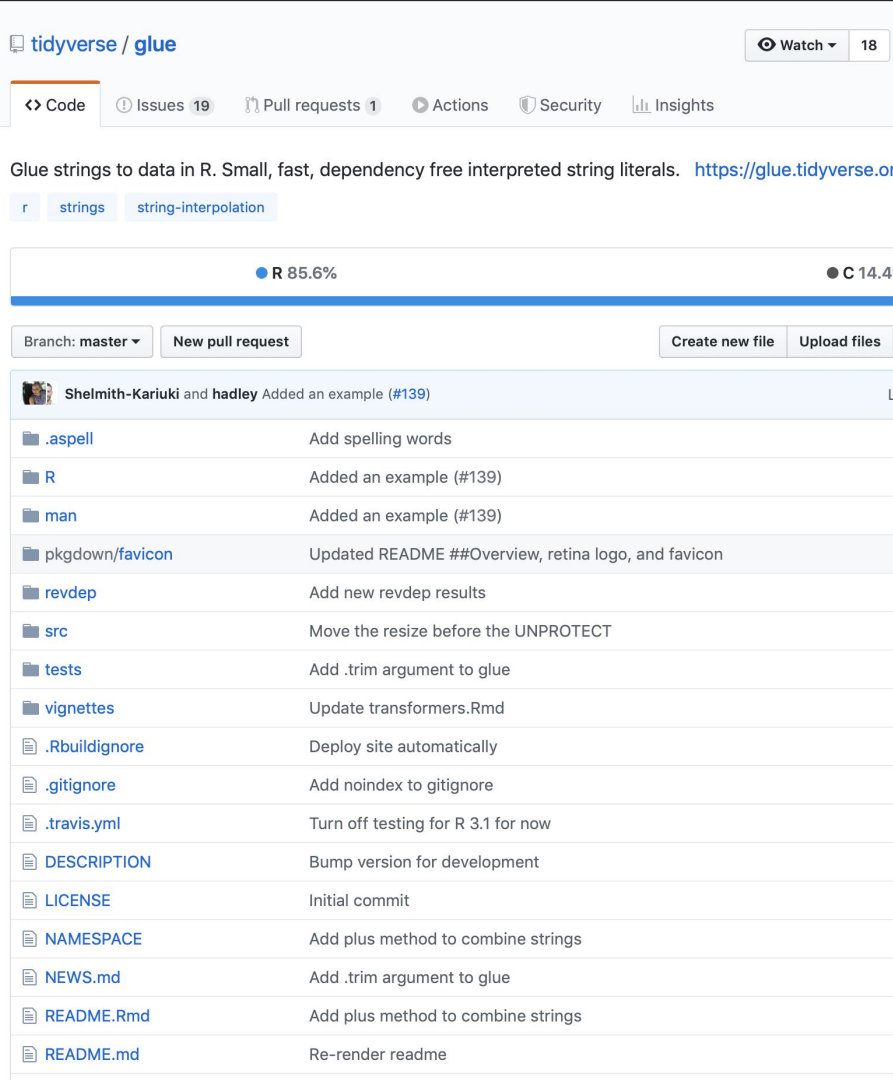
By default, tests are dropped in binary packages

Source vignettes are build into html or pdf in `inst/doc`, then installed into `doc/`

The contents of `inst/` are moved into the top-level directory

Files used only for development are listed in `.Rbuildignore`, and only exist in source package

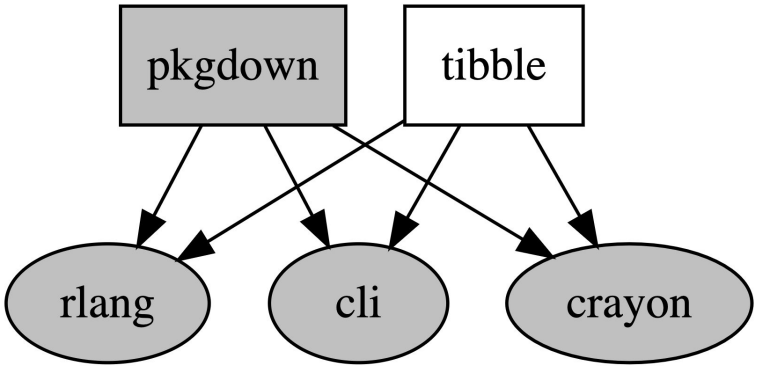




Exercise

1. Run `.libPaths()`
2. Can you find the installed version of the `helloworld` package?
3. How does it compare to the source version?
4. What else is in your library?

Dependencies



RStudio®

Exercise

1. In our package project, run `usethis::use_package("glue")`
2. Change our function to:
   ```
   hello <- function(name){glue::glue("Hello, {name}!")}
   ```
3. Re-build our package and try out our new function.
4. What version of glue is used? Why?

More Resources

https://r-pkgs.org/whole-game.html

https://github.com/jennybc/pkg-dev-tutorial

RStudio