Instalación de paqueterías desde el código fuente

Joselyn Chávez
@josschavezf1

05 Agosto 2020

Material disponible en:

https://github.com/ComunidadBioInfo/cdsb2020

Basado en el RStudioConf2020 workshop *What They Forgot* disponible en:

rstd.io/wtf-2020-rsc

https://rstats.wtf/install-a-source-package.html

A qué nos referimos con instalación desde código fuente?

La mayoría de los paquetes que instalamos son paquetes binarios pre-compilados disponibles en CRAN o Bioconductor.

En ocasiones necesitaremos instalar versiones no compiladas, en desarrollo, anteriores o disponibles únicamente en repositorios de Github.

Existen funciones que nos ayudan a instalar paquetes desde su código fuente

devtools::install_dev()

devtools::install_github()

devtools::install_version()

devtools::install()

devtools::install_dev()

Instala la versión en desarrollo de paquetes de CRAN

Es importante tener en cuenta que solamente funciona para paquetes que cuentan con el link hacia la versión de desarrollo en su archivo DESCRIPTION de CRAN

dplyr: A Grammar of Data Manipulation

A fast, consistent tool for working with data frame like objects, both in memory and out of memory.

Version:	1.0.1
Depends:	R (≥ 3.2.0)
Imports:	ellipsis, generics, glue (≥ 1.3.2), lifecycle (≥ 0.2.0), magrittr (≥ 1.5), methods, R6
Suggests:	bench, broom, callr, covr, DBI, dbplyr (≥ 1.4.3), knitr, Lahman, lobstr, microben
Published:	2020-07-31
Author:	Hadley Wickham [aut, cre], Romain François [aut], Lionel Henry [aut],
Maintainer:	Hadley Wickham <hadley at rstudio.com>
BugReports:	https://github.com/tidyverse/dplyr/issues
License:	MIT + file LICENSE
URL:	https://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr
NeedsCompilation:	yes
Materials:	README NEWS
In views:	Databases, ModelDeployment
CRAN checks:	dplyr results

devtools::install_dev("dplyr")

	Package	LibPath	Version
1	dplyr	/Users/joselynchavez/Library/R/4.0/library	1.0.1.9000
2	dplyr	/Library/Frameworks/R.framework/Versions/4.0/Res...	1.0.1

devtools::install_github()

Instala un paquete directamente desde Github.

Para usar esta función se requiere conocer el nombre del propietario y del repositorio

Esto nos ayuda a obtener el código de paquetes que no están en CRAN o que no cuentan con el link hacia la versión en desarrollo

devtools::install_github("r-lib/pkgdown")

Instalará la última versión del paquete pkgdown disponible en Github

Revisemos qué versión se instaló?

installed.packages() %>% as_tibble() %>% filter(Package == "pkgdown")

Para instalar una versión anterior

devtools::install_github("r-lib/pkgdown@v1.1.0")

Para instalar la versión del paquete en un commit específico

devtools::install_github("r-lib/pkgdown@123abc")

Hagamos un ejercicio

Mi usuario de Github (josschavezf) contiene un paquete llamado ‘minipaquete’

Instala este paquete en Rstudio

Cuántas funciones tiene y cómo se usan? Puedes probar los ejemplos

Busca en Github la lista de commits y encuentra la versión del paquete sin ejemplos.

Instala esta versión sin ejemplos en Rstudio. (Necesitarás copiar el id completo del commit)

Prueba algunas entradas poco convencionales para tratar de romper la función.

devtools::install_version()

Instala versiones previas de un paquete que se encuentra en CRAN.

devtools::install_version("dplyr","1.0.0")

	Package	LibPath	Version
1	dplyr	/Users/joselynchavez/Library/R/4.0/library	1.0.0
2	dplyr	/Library/Frameworks/R.framework/Versions/4.0/Res...	1.0.1

devtools::install()

Clona e instala la versión en desarrollo de un paquete de CRAN y con la ayuda de la función devtools::load_all() permite trabajar en él como si fuera un proyecto creado localmente.

Clona el repositorio de Github

git clone https://github.com/tidyverse/glue.git

Obtén la ruta donde lo clonaste (pwd)

~/Documents/CDSB_2020/tempo/glue

Instala el paquete desde la ruta local

devtools::install("~/Documents/CDSB_2020/tempo/glue")

Para hacer modificaciones, abre el .Rproj

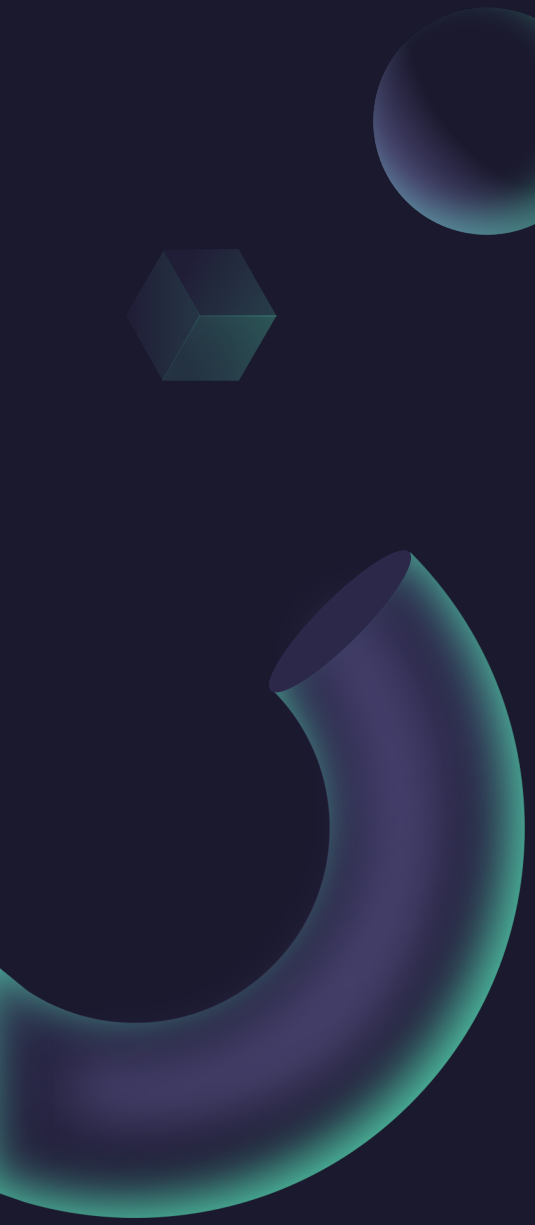
glue.Rproj

Hagamos un ejercicio

Clona el repositorio del minipaquete

Instala el paquete usando devtools::install()

Agrega una nueva función en la carpeta R e intégrala al paquete (No olvides hacer Build and Restart)



Instalación de paquetes en un directorio temporal

En ocasiones, necesitamos instalar paquetes de manera temporal, sin afectar los paquetes de nuestra sesión general.

Es necesario crear un directorio temporal

Utilizar el argumento lib de las funciones de instalación

Desinstala el paquete pkgdown

```
remove.packages("pkgdown")
```

Revisa que ya no esté instalado

```
installed.packages() %>% as_tibble() %>% filter(Package == "pkgdown") %>% View
```

Crea un directorio temporal

```
tmp_lib <- "~/Documents/CDSB_2020/tmp_library”
```

```
dir.create(tmp_lib)
```

Instala el paquete indicando la librería de instalación

devtools::install_github("r-lib/pkgdown", lib = tmp_lib)

Reinicia R

Carga el paquete indicando su localización. Prueba pkgdown::

library(pkgdown, lib = tmp_lib)

Cuando hayas terminado de usarlo puedes eliminar la conexión

unlink(tmp_lib, recursive = TRUE)

Prueba nuevamente pkgdown::

Hagamos un ejercicio

```
library(usethis)
```

```
usethis::use_course("comunidadbioinfo/paquetes-fuente")
```