

# Práctica 4: Paquetes de R

Desarrollo de paquetes en R

8 de junio de 2017

## Tarea

En el archivo de datos `datos_practicas.RData` encontrarás el banco de datos `privacion_valencia` y el objeto espacial `carto_valencia` con la cartografía de la ciudad de Valencia. El banco de datos `privacion_valencia` contiene una serie de indicadores sociales para las 598 secciones censales de la ciudad de Valencia. Estos indicadores son en este orden: desempleo, instrucción insuficiente, instrucción insuficiente en jóvenes (16-29 años), trabajadores manuales, envejecimiento, trabajadores eventuales, residentes extranjeros, hogares mono-parentales, problemas de ruido, problemas de contaminación, problemas de limpieza, problemas de comunicación, problemas de zonas verdes, problemas de delincuencia y problemas de aseo. Todas las variables han sido medidas como el porcentaje de población (en cada sección) que pertenece a estos colectivos o que dice observar dichos problemas. Todos estos indicadores podrían ser reflejo de la privación económica de cada una de las secciones censales de la ciudad de Valencia.

Empleando los paquetes de R `devtools`, `roxygen2` y `testthat`, y partiendo de la función `pinta_mapas()`,

```
pinta_mapas <- function(cartografia, variable, n_grupos){
  if (!require(RColorBrewer)){
    install.packages("RColorBrewer")
    library(RColorBrewer)
  }
  if (!require(sp)){
    install.packages("sp")
    library(sp)
  }
  paleta <- brewer.pal(n_grupos, "BrBG")
  grupos <- quantile(variable, probs = seq(0, 1, 1 / n_grupos))
  pcorte <- c(grupos[1] - 0.5, grupos[2:n_grupos], grupos[n_grupos + 1] + 0.5)
  colores <- paleta[
    findInterval(variable[match(cartografia@data$CUSEC, privacion_valencia[, 1])], pcorte)
  ]
  leyenda <- c()
  for (j in 2:length(pcorte)){
    leyenda[j] <- paste0(round(pcorte[j - 1], 2), " - ", round(pcorte[j], 2))
  }
  plot(cartografia, col = colores)
  legend("bottomright", leyenda[-1], title = "Índice de privación",
        border = NULL, fill = paleta, bty = "n")
}
```

crea un paquete de R denominado `pintamapas` que contenga:

- la función `pinta_mapas()`,
- los datos `datos_practicas.RData` utilizados en anteriores prácticas,
- documentación de la función principal y de los datos,
- pruebas de código de la función,
- un archivo `README.md` compilado desde un documento R Markdown.

## Resolución

En primer lugar hay que crear la estructura del paquete dándole nombre a este:

```
devtools::create("pintamapas")
```

Se abre el proyecto de RStudio y el siguiente paso es editar el archivo DESCRIPTION, de forma que quede algo parecido a esto:

```
Package: pintamapas
Title: Mapas con colorines
Version: 0.0.0.9000
Authors@R: person("Aitor", "Tilla-Fresca", email = "aitor.tilla@fresca.com",
  role = c("aut", "cre"))
Description: Este paquete utiliza la función pinta_mapas() para representar
  elementos en un mapa.
Depends: R (>= 3.4.0)
License: What license is it under?
Encoding: UTF-8
LazyData: true
```

A continuación se escoge una licencia para el proyecto. En este caso nos hemos decantado por una licencia MIT:

```
devtools::use_mit_license(copyright_holder = "Aitor Tilla-Fresca")
```

En relación con la función, se observan dos dependencias (RColorBrewer y sp), de forma que se introducen en el campo apropiado del archivo DESCRIPTION:

```
devtools::use_package("RColorBrewer", type = "Imports")
devtools::use_package("sp", type = "Imports")
```

Se edita la función para adaptarla al formato de paquete y se documenta con comentarios roxygen2:

```
## @title Función para pintar mapas
## @description Dado un objeto de clase SpatialPolygonDataFrame, la función
##   representa una variables dividiendo su valor en un número de grupos.
## @param cartografia objeto de clase SpatialPolygonDataFrame.
## @param variable Vector numérico a representar.
## @param n_grupos Valor numérico identificando el número de grupos a representar.
## @details Esa función... cómo mola... se merece... una ola... ¡uuuuuuuuuuuuuuuu!
## @return Devuelve un gráfico con el mapa coloreado.
## @examples
## \dontrun{
##   library(pintamapas)
##   data(privacion_valencia)
##   data(carto_valencia)
##   ACP <- princomp(privacion_valencia[, 2:16])
##   ind_privacion <- list()
##   for (i in 1:15) {
##     ind_privacion[[i]] <- as.matrix(privacion_valencia[, 2:16]) %*% ACP$loadings[, i]
##   }
##   # Se representa el primer componente
##   pinta_mapas(carto_valencia, ind_privacion[[1]], 5)
## }
## @export
pinta_mapas <- function(cartografia, variable, n_grupos, title = NULL) {
```

```

paleta <- RColorBrewer::brewer.pal(n_grupos, "BrBG")[1:n_grupos]
grupos <- quantile(variable, probs = seq(0, 1, 1 / n_grupos))
pcorte <- c(grupos[1] - 0.5, grupos[2:n_grupos], grupos[n_grupos + 1] + 0.5)
colores <- paleta[
  findInterval(variable[match(cartografia@data$CUSEC, privacion_valencia[, 1])], pcorte)
]
leyenda <- c()
for (j in 2:length(pcorte)) {
  leyenda[j] <- paste0(round(pcorte[j - 1], 2), " - ", round(pcorte[j], 2))
}
sp::plot(cartografia, col = colores)
legend("bottomright", leyenda[-1], title = title,
      border = NULL, fill = paleta, bty = "n")
}

```

Se incluyen los datos a utilizar:

```

load("datos/datos_practicas.RData")
devtools::use_data(carto_valencia, privacion_valencia)

```

Se documentan los datos en el script `./R/datos.R`:

```

#' @title Cartografía de la ciudad de Valencia
#' @description Cartografía de la ciudad de Valencia con división en secciones
#' censales según INE (2011).
#' @format Un Spatial Polygons Data Frame (sp) con 598 filas y siete variables:
#' \describe{
#'   \item{id}{id de la sección}
#'   \item{cusec}{código completo hasta nivel de sección censal (provincia,
#'     municipio, distrito, sección)}
#'   \item{csec}{código de sección censal}
#'   \item{cdis}{código de distrito}
#'   \item{cudis}{código completo hasta nivel de distrito (provincia, municipio,
#'     distrito)}
#'   \item{area}{área de la sección}
#'   \item{perimeter}{perímetro de la sección}
#' }
#'
"carto_valencia"

#' @title Indicadores de privación para 598 secciones censales de Valencia
#' @description Indicadores de privación para 598 secciones censales de Valencia
#' en base al proyecto MEDEA, utilizando datos del censo de 2001 del INE. Los
#' indicadores son desempleo, instrucción insuficiente, instrucción insuficiente
#' en jóvenes (16-29 años), trabajadores manuales, envejecimiento, trabajadores
#' eventuales, residentes extranjeros, hogares mono-parentales, problemas de
#' ruido, problemas de contaminación, problemas de limpieza, problemas de
#' comunicación, problemas de zonas verdes, problemas de delincuencia y
#' problemas de aseo.
#' @format Un banco de datos (clase data.frame) con 598 filas y 16 variables:
#' \describe{
#'   \item{seccion_censal}{código de sección censal}
#'   \item{desempt}{porcentaje de población activa en situación de desempleo}
#'   \item{insufict16}{porcentaje de población con instrucción insuficiente}
#'   \item{inst1629}{porcentaje de población entre 19-29 años con instrucción

```

```
#'      insuficiente}
#'      \item{manuales}{porcentaje de población que son trabajadores manuales}
#'      \item{envejet}{porcentaje de población mayor de 65 años}
#'      \item{eventual}{porcentaje de empleados con trabajo eventual}
#'      \item{resexj}{porcentaje de población residente de origen extranjero}
#'      \item{monomart}{porcentaje de hogares mono-parentales}
#'      \item{ruido}{porcentaje de población que declara tener problemas de
#'        ruido}
#'      \item{contamina}{porcentaje de población que declara tener problemas de
#'        contaminación}
#'      \item{limpieza}{porcentaje de población que declara tener problemas de
#'        limpieza}
#'      \item{comunica}{porcentaje de población que declara tener problemas de
#'        comunicación}
#'      \item{zverdes}{porcentaje de población que declara tener problemas de
#'        zonas verdes}
#'      \item{delincuencia}{porcentaje de población que declara tener problemas
#'        de delincuencia}
#'      \item{aseo}{porcentaje de población que declara tener problemas de aseo}
#'    }
#'
"privacion_valencia"
```

Es el turno de construir la documentación:

```
devtools::document()
```

Se genera un archivo README.md y se edita, incorporando información acerca del paquete:

```
devtools::use_readme_Rmd()
```

El último paso en la etapa de desarrollo es construir las pruebas de código, para lo cual se usa `devtools`:

```
devtools::use_testthat()
```

Creando el directorio `./test/` en el que se introduce el archivo `testthat.R`, y el directorio `./test/thestthat/`, que será donde se almacene el script con las pruebas de la función. Dicho script podría comprobar aspectos vistos durante la sesión, como:

1. se devuelve un error si el argumento `cartografia` no es de clase `SpatialPolygonsDataFrame`,
2. se devuelve un error si el argumento `variable` no es un vector numérico de longitud igual a la longitud de la cartografía,
3. se devuelve un error si el argumento `n_grupos` no es un vector numérico de longitud 1,

y podría tener el siguiente aspecto:

```
context("Comprobaciones de clase y dimensiones correctas")
library(pintamapas)
data(privacion_valencia)
data(carto_valencia)
test_that("pinta_mapas solo funciona con SpatialPolygonsDataFrame", {
  expect_error(pinta_mapas("a", 1:10, 1))
  expect_error(pinta_mapas(TRUE, 1:10, 1))
  expect_error(pinta_mapas(1:10, 1:10, 1))
})
test_that("pinta_mapas solo funciona con variables numéricas", {
  expect_error(pinta_mapas(carto_valencia, 1, 1))
})
```

```
    expect_error(pinta_mapas(carto_valencia, 1:40000, 1))
  })
  test_that("pinta_mapas solo funciona con n_grupos numérico y longitud 1", {
    expect_error(pinta_mapas(carto_valencia, privacion_valencia, "a"))
    expect_error(pinta_mapas(carto_valencia, privacion_valencia, 1:10))
  })
```