



Grupo de Usuarios de R de Sevilla

Programación funcional en tidyverse: introducción a purrr

Juan Luis Pérez Carretero

5 Marzo 2019, 19 a 20:30 h

Sala TIC1, CRAI Reina Mercedes

<http://bit.ly/SevillaRmeetup>



R



- Podemos hablar de R como un ecosistema de desarrollo para ciencia de datos aplicada



Microsoft R Open

- **R Project** : R Base
- **Entornos de desarrollo** : Rstudio, Microsoft, Oracle, Tableau, Teradata.
- **Comunidad** : R consourtium, RWeekly, R-Help, #rstas, stackoverflow.



consortium



Programación Funcional

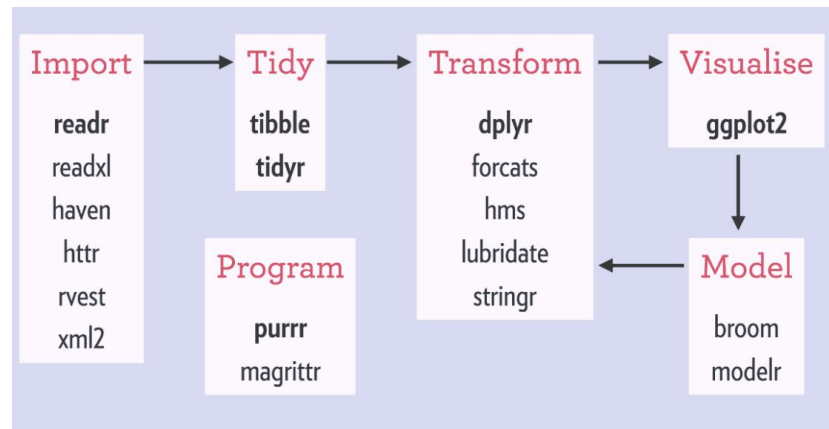
- ¿Podemos considerar R como un lenguaje de programación funcional?
 - **First Class Functions:** Funciones se pueden tratar como cualquier otro objeto del lenguaje.
 - **Pure Functions:**
 - Deterministas: Lo que devuelve sólo depende de la entrada que recibe
 - No tienen side-effects

“Strictly speaking, R isn’t a functional programming *language* because it doesn’t require that you write pure functions”

Hadley Wickham

Tidyverse

- Conjunto de librerías consistentes y unificadas, con un patrón de diseño común orientado a la ciencia de datos:
 - Permiten realizar el workflow completo de cualquier proyecto: Limpieza, exploración, modelado y comunicación.
 - Consistencia
 - Organización de los paquetes
 - Tidy data
 - Coding Style



Purrr

- Nos permite adoptar un estilo de programación funcional con tidyverse
 - Ofrece un conjunto de funciones para manipular vectores, listas o dataframes.
- El objetivo es reducir duplicidad en el código y la repetición mediante el concepto de iteración.
 - Favorece la lectura del código
 - Más sencillo y rápido realizar cambios
 - Menos propensión a errores o bugs

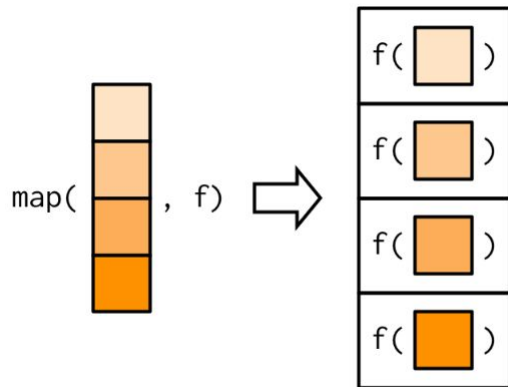


Purrr: Funciones base para iteración

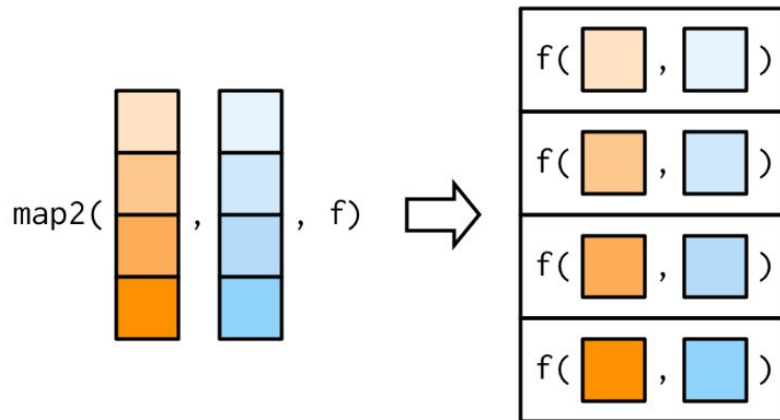
- Apply Functions
 - **map(.x,.f,...)**: Aplica .f a cada elemento del vector .x y devuelve una lista:
 - map_lgl(.x,.f,...)
 - map_int(.x,.f,...)
 - map_dbl(.x,.f,...)
 - map_chr(.x,.f,...)
 - map_df(.x,.f,...)
 - **map2(.x,.y,.f,...)** : Itera sobre dos vectores
 - **pmap(.l,.f,...)**: Itera sobre más de dos vectores
 - **walk(.x,.f,...)**: Se utiliza para llamadas a funciones por sus side-effects: print, llamadas a variables del sistema, I/O.No devuelve nada.

Purrr: Funciones base para iteración

`map(.x,.f,...)`



`map2(.x,.y,.f,...)`



Purrr: Ventajas map sobre programación imperativa

```
output <- vector("double", ncol(data))  
for (i in seq_along(data)) {  
  output[[i]] <- mean(data[[i]])  
}
```

```
output <- data %>%  
  map_dbl(.,mean)
```

“The chief benefits of using functions like `map()` is not speed, but clarity: they make your code easier to write and to read”

Hadley Wickham

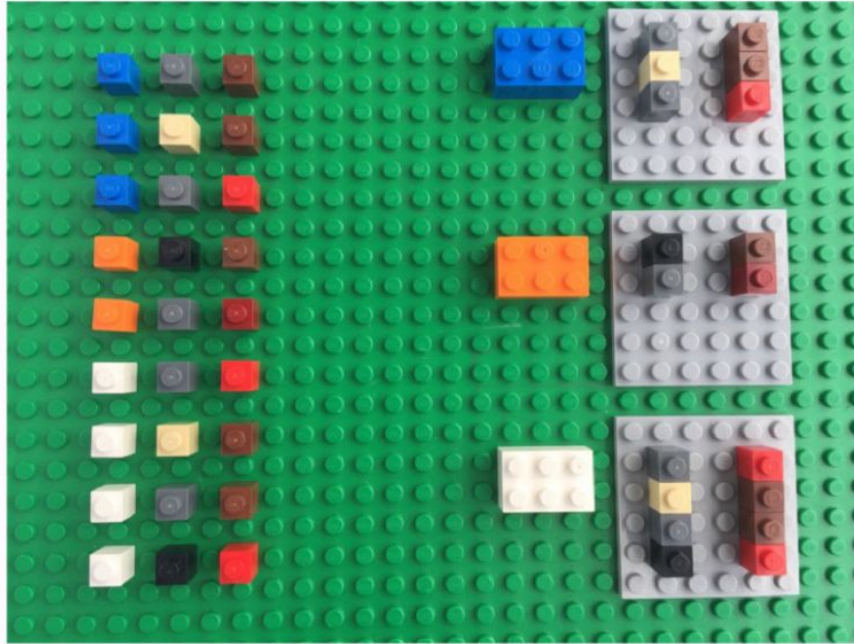
Purrr: Ventajas map_* sobre apply de R base

- Su principal ventaja es la consistencia de la sintaxis y tipos. Además de :
 - Consistencia en los tipos que devuelve
 - Shortcuts
 - Sintaxis consistente que facilita iteraciones complejas
 - Funciones adicionales que ofrece purrr para trabajar con listas
 - Compatible con list-columns dataframes

Lists-Columns

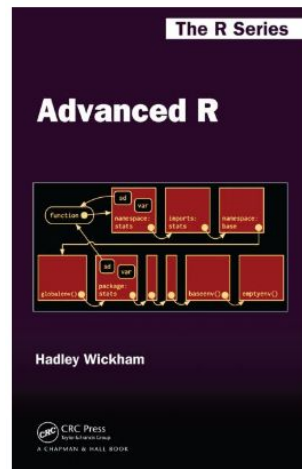
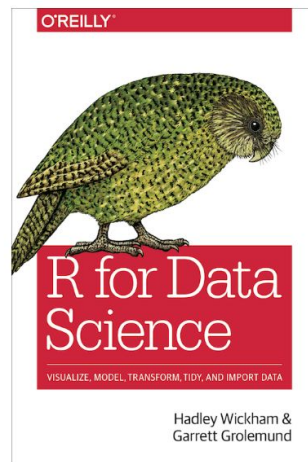
data frame

nested data frame



Bibliografía

- R for Data Science: <https://r4ds.had.co.nz/>
- AdvancedR: <http://adv-r.had.co.nz/>



Purrr

Ejemplos