

purrr

Fundamentos lenguajes: R

Alberto Torres Barrán y Irene Rodríguez Luján

2020-01-10

Iteración

- Se utiliza para aplicar el mismo código a varias entradas
- La forma más conocida son los bucles:
 - for

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

- while

```
i <- 1  
while (i <= length(df)) {  
  i <- i + 1  
}
```

- Hemos visto otra forma "oculta" de iteración, las funciones `summarize_all()` y `summarize_if()`

purrr

- La librería proporciona funciones que sustituyen a los bucles en la mayoría de los casos más comunes:
 - `map()` , crea una lista
 - `map_lgl()` , crea un vector lógico
 - `map_int()` , crea un vector de enteros
 - `map_dbl()` , crea un vector de dobles
 - `map_chr()` , crea un vector de cadenas de caracteres
 - `map_df()` , crea un data frame

```
df <- select_if(mpg, is.numeric)
map_dbl(df, mean, na.rm = TRUE)
##      displ      year      cyl      cty      hwy
##  3.471795 2003.500000  5.888889 16.858974 23.440171
```

`map_*()` acepta funciones definidas por el usuario y tiene una sintáxis especial para declarar funciones anónimas

```
map_df(df, ~(. - min(.) / (max(.) - min(.))))  
## # A tibble: 234 x 5  
##   displ  year  cyl  cty  hwy  
##   <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1  1.50 1777.    3  17.7 28.6  
## 2  1.50 1777.    3  20.7 28.6  
## 3  1.70 1786.    3  19.7 30.6  
## 4  1.70 1786.    3  20.7 29.6  
## 5  2.50 1777.    5  15.7 25.6  
## 6  2.50 1777.    5  17.7 25.6  
## 7  2.80 1786.    5  17.7 26.6  
## 8  1.50 1777.    3  17.7 25.6  
## 9  1.50 1777.    3  15.7 24.6  
## 10 1.70 1786.    3  19.7 27.6  
## # ... with 224 more rows
```