OPSO79-1-UCSH2021

Diseños de investigación e introducción a dplyr en R. Bloque práctico.

27/08/2021

Nuestro mundo en datos (provación)

Our World in data

`OurWorldInData` es una publicación en-línea que presenta datos y resultados empíricos que muestran el cambio en las condiciones de vida en todo mundo.

Un estudiante de magíster en políticas públicas está desarrollando un paquete en R para descargar y visualizar directamente estos datos.

Como está en desarrollo aún no se encuentra en la CRAN. Si se quiere ocupar tiene que descargarse así:

```
install.packages("devtools")

devtools::install_github("piersyork/owidR")

library(owidR)
```

Más adelante entenderemos bien. Por ahora veamos lo que nos permite hacer en R Cloud.


Lenovo
Dra: Paula Daza Narbona

Un poco más de R base

Insertar lenguaje R en .rmd

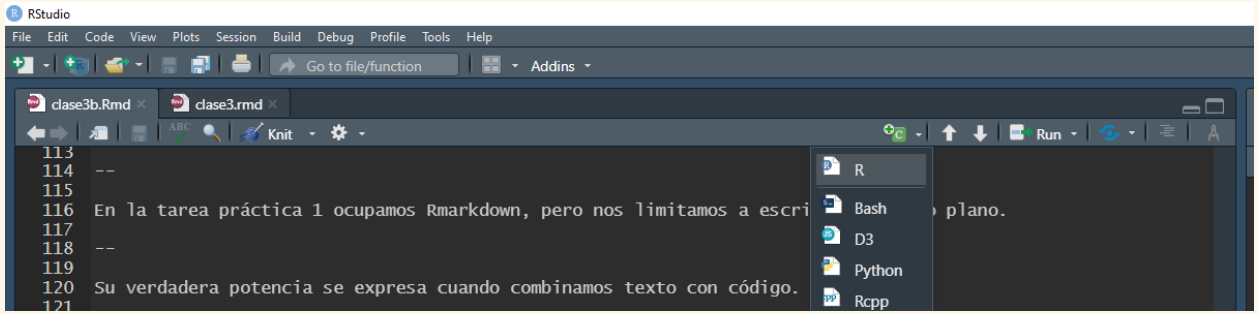
Hemos visto tres formas de interactuar con R:

- Consola.
- Script `.R`.
- Script `.rmd` o `Rmarkdown`.

En la tarea práctica 1 ocupamos Rmarkdown, pero nos limitamos a escribir en texto plano.

Su verdadera potencia se expresa cuando combinamos texto con código.

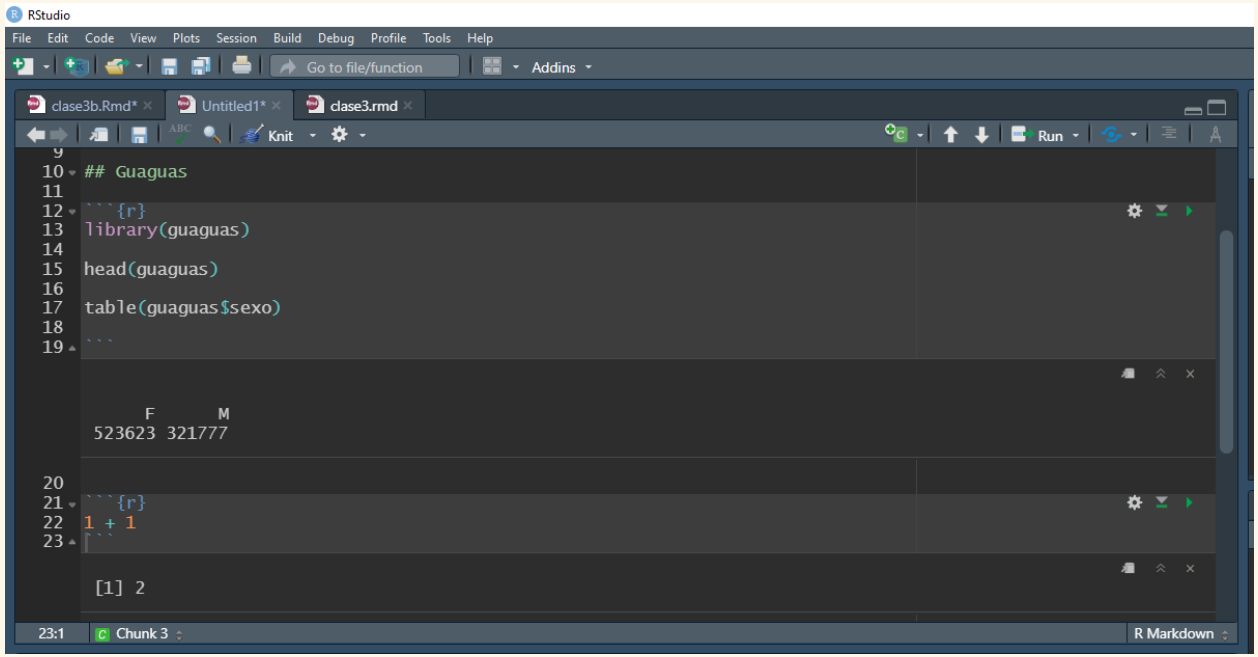
Para agregar un `chunk` o trozo de código de lenguaje R en RMarkdown:



Insertar lenguaje R en .rmd

Aparecerá lo siguiente:

En el espacio interno podemos escribir código R:



Algunos ejemplos

Operadores lógicos

Los **operadores lógicos** son muy importantes para la programación.

R cuenta con operadores de comparación binaria.

```
x < y    # menor que
x > y    # mayor que
x <= y   # menor o igual que
x >= y   # mayor o igual que
x == y   # igual a
x != y   # distinto a
```

👀: Nota que == permite comparar si dos valores son iguales. Ten cuidado de **NO** usar = que es interpretado como un operador de asignación (es como usar **<-**).

Operadores lógicos

Algunos ejemplos con números:

```
x <- c(1,2,5)
y <- c(4,4,3)
x == y
#> [1] FALSE FALSE FALSE
x != y
#> [1] TRUE TRUE TRUE
x < y
#> [1]  TRUE  TRUE FALSE
```

Otros operadores muy importantes son | (o) e & (y)

```
guaguas[guaguas$nombre=="Salvador" &
          (guaguas$anio==1972| guaguas$anio==1979),]
```

```
## # A tibble: 2 x 5
##    anio nombre   sexo      n proporcion
##   <dbl> <chr>    <chr> <dbl>      <dbl>
## 1  1972 Salvador M       183   0.000576
## 2  1979 Salvador M        59   0.000218
```

Introducción a paquete dplyr

El paquete dplyr

Instalar y cargar como cualquier otro paquete:

```
install.packages("dplyr")
```

```
library(dplyr)
```

El paquete dplyr

Nos proporciona una "gramática" particular para manipular y aplicar funciones sobre bases de datos.

El paquete fue desarrollado por Hadley Wickham de RStudio. Todo lo que haremos con sus funciones se puede hacer con r base.

Ocuparemos muchos de sus paquetes (ggplot, tidyr, haven, readxl, httr, lubridate, etc.).

Manipulación básica

Revisaremos 5 verbos básicos de `dplyr`:

- `select()`
- `filter()`
- `arrange()`
- `rename()`
- `mutate()`

Select()

Nos permite seleccionar de forma intuitiva las columnas que indiquemos.

```
select(data, columna)
```

Selecciona 1 columnas de la base `guaguas`:

```
select(guaguas,nombre)
```

```
## # A tibble: 845,400 x 1
##    nombre
##    <chr>
##  1 María
##  2 José
##  3 Juan
##  4 Luis
##  5 Rosa
##  6 Ana
##  7 Manuel
##  8 Olga
##  9 Carlos
## 10 Pedro
```

Select()

Podemos seleccionar más de una columna agregandolas como argumento

```
select(guaguas,nombre,n)
```

```
## # A tibble: 845,400 x 2
##    nombre     n
##    <chr>  <dbl>
##  1 María   2130
##  2 José     984
##  3 Juan     636
##  4 Luis     631
##  5 Rosa     426
##  6 Ana      340
##  7 Manuel   326
##  8 Olga     289
##  9 Carlos   277
## 10 Pedro    269
## # ... with 845,390 more rows
```

select()

E incluso reordenar las columnas, ocupando sus nombres o sus posiciones

```
select(guaguas,5:1)
```

```
## # A tibble: 845,400 x 5
##    proporcion     n sexo  nombre  anio
##         <dbl> <dbl> <chr> <chr>  <dbl>
##  1     0.104   2130 F     María   1920
##  2     0.0483   984 M     José    1920
##  3     0.0312   636 M     Juan    1920
##  4     0.0310   631 M     Luis    1920
##  5     0.0209   426 F     Rosa    1920
##  6     0.0167   340 F     Ana     1920
##  7     0.0160   326 M     Manuel  1920
##  8     0.0142   289 F     Olga    1920
##  9     0.0136   277 M     Carlos  1920
## 10     0.0132   269 M     Pedro   1920
## # ... with 845,390 more rows
```

filter()

Nos permite seleccionar las filas que indiquemos.

Para eso utilizamos operadores lógicos (> , < , >=, <=, ==, !=, &)

```
filter(guaguas, anio==1920 & nombre=="Pedro")
```

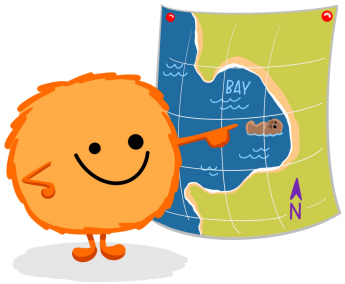
```
## # A tibble: 1 x 5
##    anio nombre sexo      n proporcion
##   <dbl> <chr>  <chr> <dbl>      <dbl>
## 1  1920 Pedro  M       269     0.0132
```

dplyr::filter()

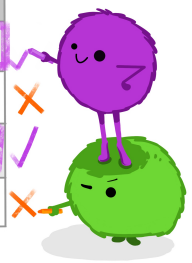
KEEP ROWS THAT satisfy your CONDITIONS

keep rows from... | this data... | ONLY IF... | type is "otter" | AND | site is "bay"

```
filter(df, type == "otter" & site == "bay")
```



type	food	site
otter	urchin	bay
shark	seal	channel
otter	abalone	bay
otter	crab	wharf



@allison_horst

arrange()

rename()

mutate()

Operador pipe (tubo)

Solo si queda tiempo en la clase.

Es un operador de `magrittr` que se combina con los verbos de dplyr.

Se escribe %>% (*pipe* o tubo).

El operador %>% nos permite concatenar funciones, haciendo más sencilla la lectura del código. Se lee de izquierda a derecha.

¿Como saber los nombres de las variables de la base de datos `guaguas`?

```
names(guaguas)   ## la manera "tradicional" o R base
```

```
## [1] "anio"       "nombre"     "sexo"       "n"          "proporcion"
```

Con pipes sería así:

```
guaguas %>% names()  ## el objeto primero, luego la función
```

```
## [1] "anio"       "nombre"     "sexo"       "n"          "proporcion"
```

Pipes

Con pipes podemos concatenar funciones.

Tarea práctica N°2 🏠

Ocupa las herramientas de `dplyr` vistas en clase sobre la base `guaguas` para responder a las preguntas

- Señalar cuantas personas con tu nombre nacieron el mismo año que tú.
- Señalar cual fue el nombre más usado el año que tú naciste.
- Guardar en un objeto nuevo los 10 nombres de mujer más usados el año que tú naciste.
- Crea una nueva variable en el nuevo objeto que se llame "biblico".
- Determinar cuál de los siguientes tres nombres tiene mayores inscripciones a lo largo del tiempo (1920-2019): ¿María, Juan o José?
- Crea dos nuevas bases de datos llamadas Salvador y Augusto. Cada una solo debe tener casos entre 1960 y 1990. Identifica en que año estuvo más de moda cada nombre.

Recursos web utilizados

Xaringan: Presentation Ninja, de Yihui Xie. Para generar esta presentación.

Ilustraciones de Allison Horst

Para reforzar y seguir aprendiendo

Administración de datos en R. ¿Qué es dplyr?.

Capítulo 5 libro "Ciencia de Datos" de Hadley Wickham

Operadores lógicos en R

Otra explicación de las pipes.

Bibliografía utilizada

Wickham, H. (2021). *R Para Ciencia de Datos*. URL: https://es.r4ds.hadley.nz/.