

# Programación en R para el análisis de datos

## Flujos de trabajo y proyectos

---

Nicolás Schmidt

`mail::nschmidt@cienciassociales.edu.uy`

`GitHub::@Nicolas-Schmidt`

Departamento de Ciencia Política

Facultad de Ciencias Sociales

1. Scripts
2. Memoria
3. Encoding
4. Intertemporalidad
5. Rutas de búsqueda
6. Proyectos
7. Paquetes
8. Objetos y Funciones
9. Exploración de datos

# Scripts

---

Un script es un archivo que tiene extensión .R pero no es mas que texto plano. Se puede escribir en un .txt o en cualquier editor de texto simple.

Un script contiene una rutina, un programa o una o varias funciones que se desean guardar.

Dos recomendaciones:

- Nombre del archivo claro
- No sean económicos con los comentarios ('#')

# Memoria

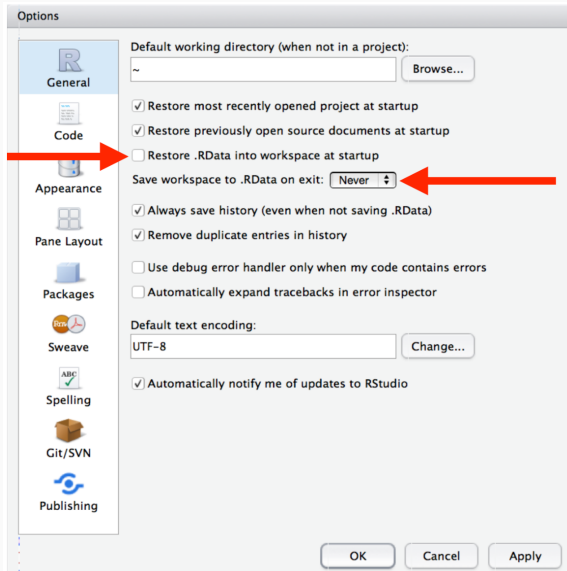
---

En R todo lo que permanece durante una sesión de trabajo está asociado a un nombre.

Una práctica deseable es iniciar una nueva sesión de trabajo con todo el ambiente de trabajo (básicamente el `.GlobalEnv`) limpio.

```
ls()  
object()  
rm(list = ls())
```

# Configurar: → Tools → Global Options... → General

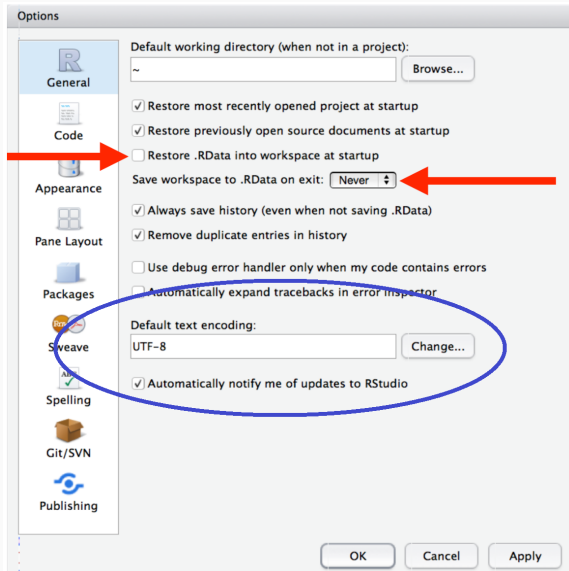




# Encoding

---

# Configurar: UTF-8



# Intertemporalidad

---



Windows File Explorer window titled "cursoR". The address bar shows the path "cursoR". The left sidebar has "Acceso rápido" (Quick access) highlighted. The main area displays a table of files and folders:

Nombre	Fecha de modificación	Tipo	Tamaño
a	21/3/2016 11:51	Archivo RPRES	1 KB
Funciones	16/4/2017 21:27	Archivo R	9 KB
Graficos	27/10/2016 17:15	Archivo R	6 KB
Notas 1	29/4/2016 12:30	Archivo R	6 KB
Funciones_finales	16/4/2017 21:27	Archivo R	9 KB
Graficos_posta	27/10/2016 17:15	Archivo R	6 KB
Notas 2	29/4/2016 12:30	Archivo R	6 KB

7 elementos



## Causal de divorcio del análisis de datos!

cursoR

Archivo Inicio Compartir Vista

cursoR

ES UN CHISTE NO??

Buscar en cursoR

Nombre	Fecha de modificación	Tipo	Tamaño
a	21/3/2016 11:51	Archivo RPRES	1 KB
Funciones	16/4/2017 21:27	Archivo R	9 KB
Graficos	27/10/2016 17:15	Archivo R	6 KB
Notas 1	29/4/2016 12:30	Archivo R	6 KB
Funciones_finales	16/4/2017 21:27	Archivo R	9 KB
Graficos_posta	27/10/2016 17:15	Archivo R	6 KB
Notas 2	29/4/2016 12:30	Archivo R	6 KB

QUE!??

FINALES DE QUE??

HAY GRAFICOS QUE NO SON POSTA??

7 elementos

## Rutas de búsqueda

---

**setwd()**

# Proyectos

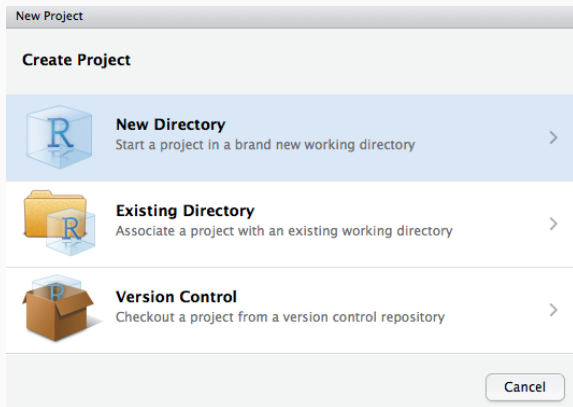
---

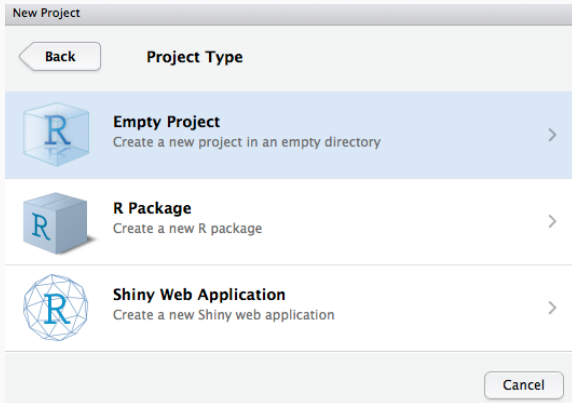


Trabajar en proyectos permite controlar las rutas de búsqueda y los directorios.

Esto es realmente importante si se trabaja en conjunto con otras personas, o si vamos a enviar el trabajo que estamos haciendo a otra persona.


Creación: → File → New Project...





New Project

[Back](#) **Create New Project**



Directory name:

Create project as subdirectory of:  
 [Browse...](#)

☐ Create a git repository

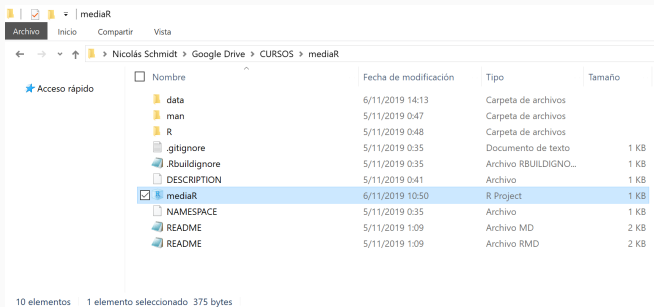
☐ Use packrat with this project

☐ Open in new session

[Create Project](#) [Cancel](#)

**here::here()**

# Funcionamiento de `here::here()`



## Ejemplo:

```
datos <- rio::import(here::here("data", "mis_datos.xlsx"))
datos$new <- apply(datos[, 4:8], 1, sum, na.rm = TRUE)
rio::export(datos, here::here("data", "datos.xlsx"))
```

# Paquetes

---

Un paquete de R conceptualmente es un conjunto de funciones relativas a un asunto, método o área.

Computacionalmente es un espacio de nombres ( `'NAMESPACE'` ) que están asociados al nombre del paquete.

Los nombres que viven en ese ambiente básicamente son los nombres de las funciones. Cada vez que se quiere usar alguna funcionalidad de un paquete se debe llamar a la función por su nombre.



# Uso de paquetes

Para poder usar un paquete es necesario cargar ese espacio de nombres a la ruta de búsqueda en la sesión de trabajo:

```
search()
```

```
## [1] ".GlobalEnv"      "package:knitr"    "package:stats"  
## [4] "package:graphics" "package:grDevices" "package:utils"  
## [7] "package:datasets" "Autoloads"        "package:base"
```

```
library(MASS)
```

```
search()
```

```
## [1] ".GlobalEnv"      "package:MASS"     "package:knitr"  
## [4] "package:stats"   "package:graphics" "package:grDevices"  
## [7] "package:utils"   "package:datasets" "Autoloads"  
## [10] "package:base"
```

# Uso de paquetes

```
library(dplyr, pos = 3)
```

```
##  
## Attaching package: 'dplyr'  
## The following object is masked _by_ 'package:MASS':  
##  
##   select  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
search()
```

```
## [1] ".GlobalEnv"          "package:MASS"          "package:dplyr"  
## [4] "package:knitr"        "package:stats"         "package:graphics"  
## [7] "package:grDevices"    "package:utils"         "package:datasets"  
## [10] "Autoloads"            "package:base"
```

# Uso de paquetes

```
select
```

```
## function (obj)
## UseMethod("select")
## <bytecode: 0x556b4759c2a8>
## <environment: namespace:MASS>
```

```
dplyr::select
```

```
## function (.data, ...)
## {
##   UseMethod("select")
## }
## <environment: namespace:dplyr>
```

Una manera de evitar conflicto de nombres es llamar a la función y su respectivo ambiente:

```
nombre_del_paquete::nombre_de_la_función()
```

El *match* **nombre del paquete** con **nombre de función** de los paquetes que están en CRAN es único, por lo que nunca va a dar problema esa búsqueda.

# Objetos y Funciones

---

“To understand computations in R, two slogans are helpful:

Everything that exists is an object.

Everything that happens is a function call.”

John Chambers

# Objetos y funciones

```
c(1:5)
```

```
## [1] 1 2 3 4 5
```

```
letras <- letters
```

```
vec <- c(2, 3, 5, 8)
```

```
promedio <- function(x){sum(x)/length(x)}
```

```
promedio(vec)
```

```
## [1] 4.5
```

```
library(pryr)
```

```
fun_calls(promedio)
```

```
## [1] "{" "/" "sum" "length"
```

```
ls()
```

```
## [1] "letras" "promedio" "vec"
```

# Objetos y funciones

```
1 + 3
```

```
## [1] 4
```

```
`+`(1, 3)
```

```
## [1] 4
```

```
do.call(`+`, list(1,3))
```

```
## [1] 4
```

```
numeros <- c(1, 3)  
suma <- do.call(`+`, as.list(numeros))  
suma
```

```
## [1] 4
```

```
do.call(sum, as.list(numeros))
```

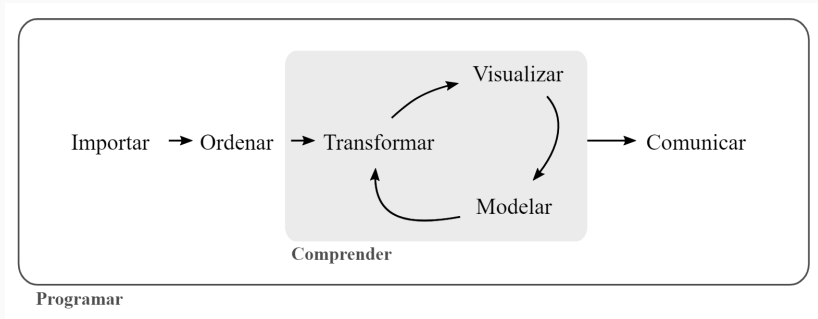
```
## [1] 4
```



# Exploración de datos

---

# Flujo del análisis de datos



Fuente: r4ds