



Universidad del
Rosario

Facultad de
Economía

Laboratorio de
Finanzas

Escuela de Ingeniería,
Ciencia y Tecnología

TALLER DE DATA SCIENCE EN **R**



Sueño**SER**

Potencializa tu perfil
académico y profesional.

6:00 p.m a 8:00 p.m

10 al **12** de octubre

Trasciende,
construye
y Lidera

Rosarios
con
Propósito



Miguel Angel Orjuela Rocha
Ing. de Sistemas y Computación

Skills:

- Desarrollo Fullstack
- Analítica de datos
- Transferencia de conocimiento

Contacto:

✉ miguela.orjuela@urosario.edu.co

[in](https://www.linkedin.com/in/miguel-orjuela/) <https://www.linkedin.com/in/miguel-orjuela/>

[gh](https://github.com/maorjuela73) <https://github.com/maorjuela73>

Contenido

- Revisión de instalación
- Los programas R y RStudio
- Interfaz de RStudio
- Programación básica
 - Tipos de datos
 - Estructuras de control
 - Funciones



Revisión de instalación



Los programas R y RStudio

¿Qué es R?

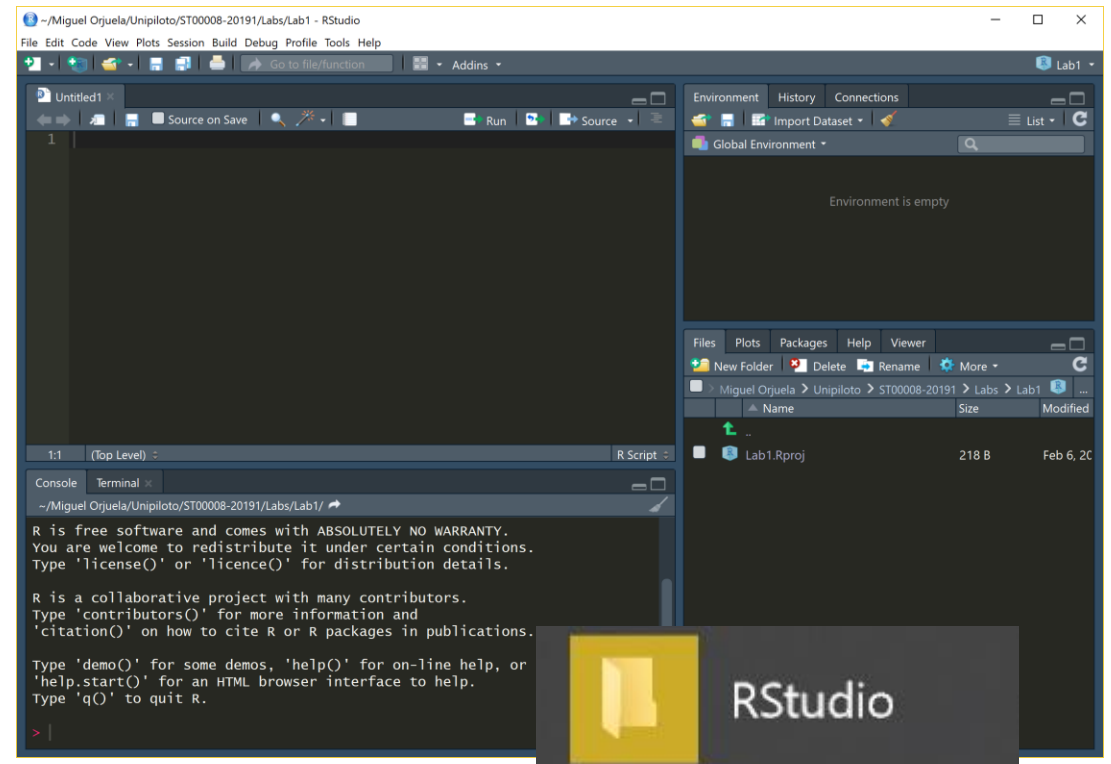
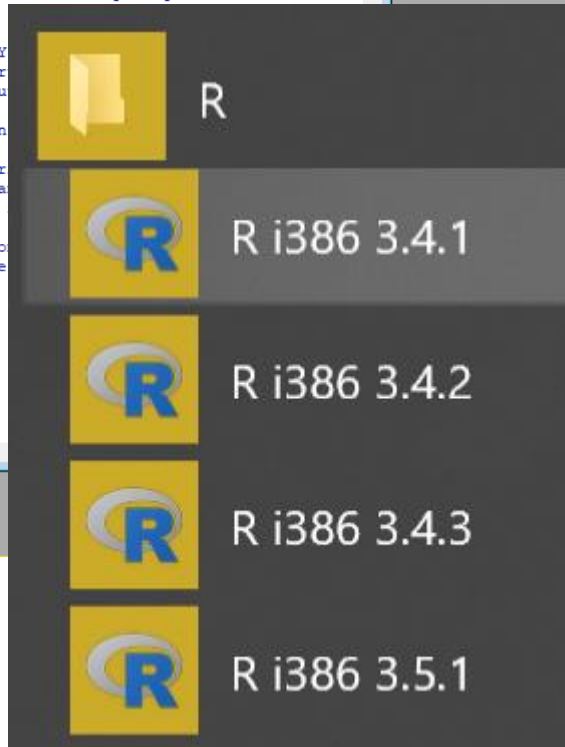
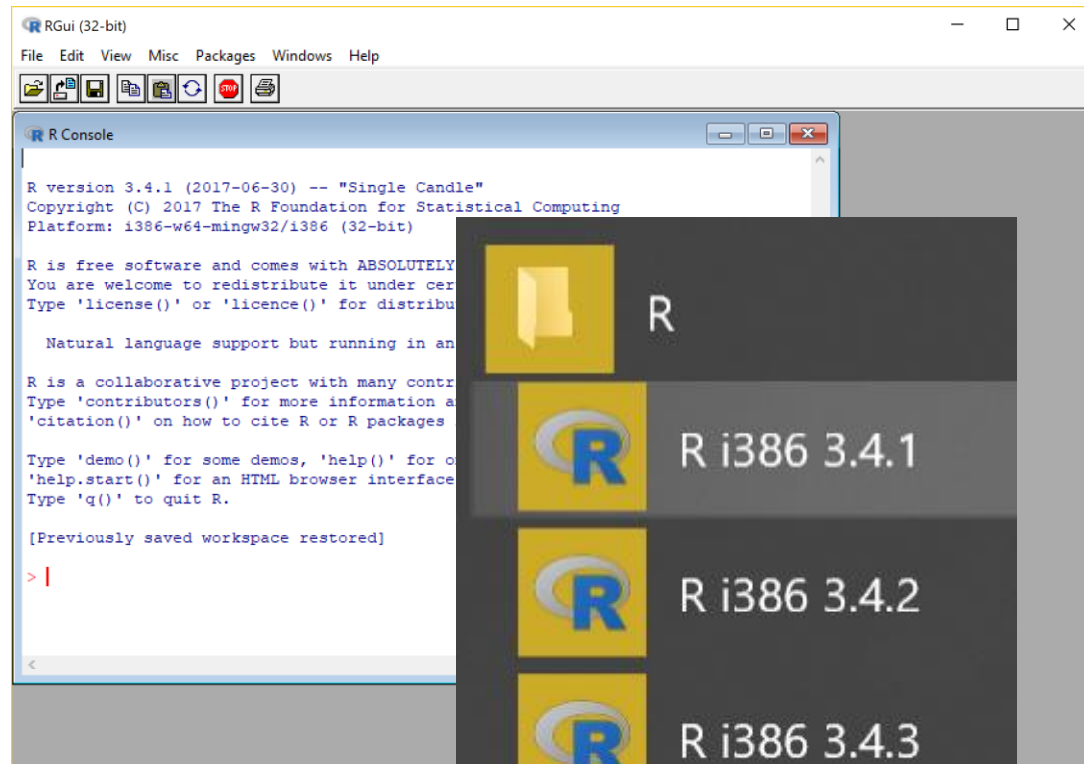
- Es un software libre
- Diseñado para hacer análisis estadísticos y gráficas
- Tiene una curva de aprendizaje relativamente sencilla
- Permite hacer prototipos de modelos muy rápido y con resultados muy buenos
- Permite compartir código de forma sencilla

¿Qué es RStudio?

- Es una empresa que produce software especial para programar en R
 - Entorno Integrado de Desarrollo
 - Soluciones de escritorio y web
 - Paquetes especiales para R
- Tiene unas soluciones gratuitas (community) y otras pagas (profesionales)

R y RStudio

Los programas R y RStudio



RStudio Cloud

Los programas R y RStudio

- Servicio web que permite ejecutar R en la nube (servidores de RStudio)

<https://rstudio.cloud> ▼ Traducir esta página

RStudio Cloud

RStudio Cloud is a lightweight, cloud-based solution that allows anyone to do, share, teach and learn data science online. ... There is nothing to configure and ...

- Se inicia sesión y se crea un nuevo proyecto

New Project ▼



New RStudio Project



New Jupyter Project



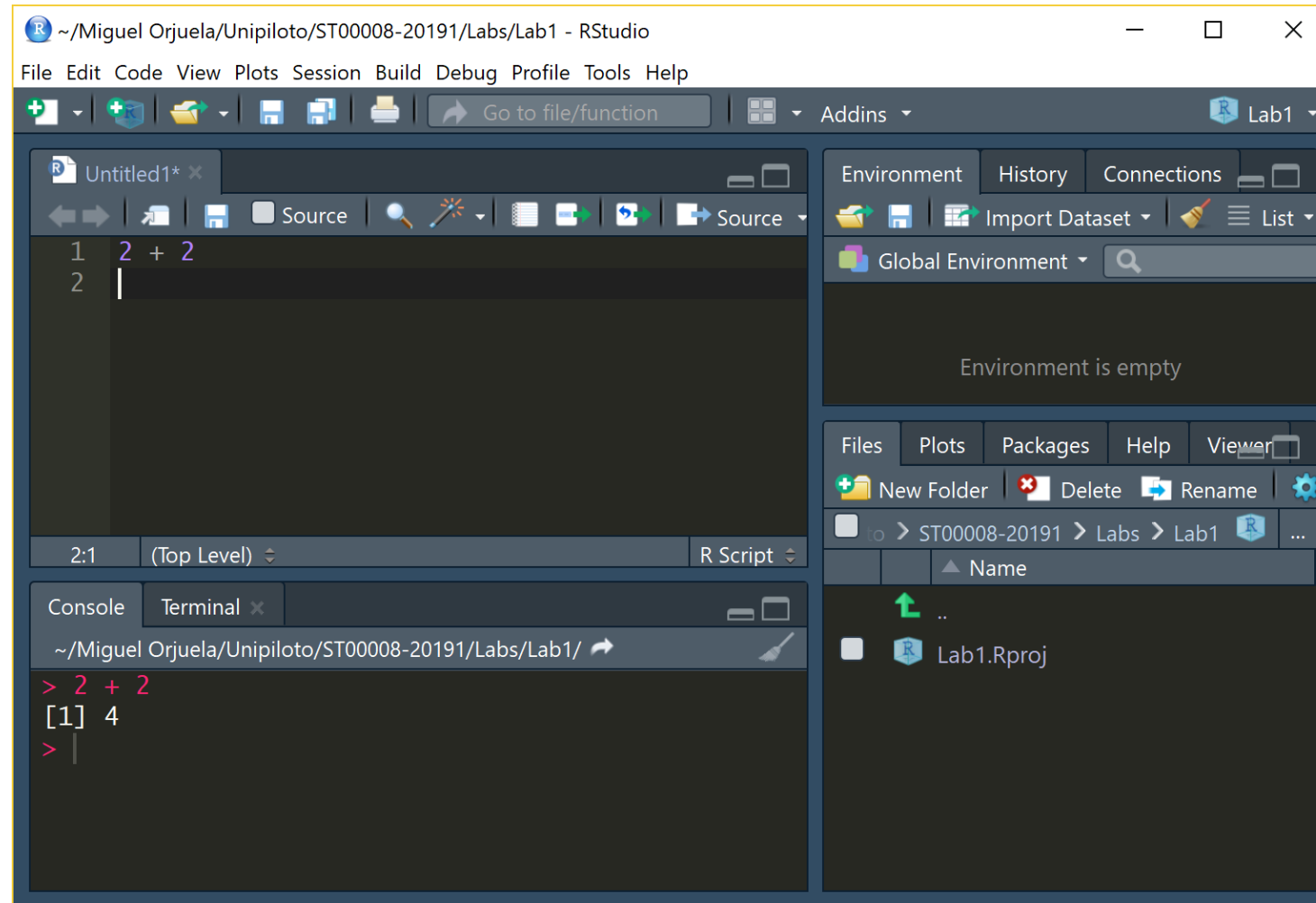
New Project from Git Repository



Interfaz de RStudio

Interfaz de RStudio

Interfaz de RStudio



RStudio IDE :: CHEAT SHEET

Documents and Apps

Annotations for Documents and Apps pane:

- Open Shiny, R Markdown, knitr, Sweave, LaTeX, Rd files and more in Source Pane
- Check spelling
- Render output
- Choose output format
- Choose insert location
- Insert code chunk
- Jump to previous chunk
- Jump to next chunk
- Run selected chunk
- Publish to server
- Show file outline
- Access markdown guide at **Help > Markdown Quick Reference**
- Jump to chunk
- Set knitr chunk options
- Run this and all previous code chunks
- Run this code chunk
- RStudio recognizes that files named **app.R**, **server.R**, **ui.R**, and **global.R** belong to a shiny app
- Run app
- Choose location to view app
- Publish to shinyapps.io or server
- Manage publish accounts

Write Code

Annotations for Write Code pane:

- Navigate tabs
- Open in new window
- Save
- Find and replace
- Compile as notebook
- Run selected code
- Re-run previous code
- Source with or without Echo
- Show file outline
- Multiple cursors/column selection with **Alt + mouse drag**
- Code diagnostics that appear in the margin. Hover over diagnostic symbols for details.
- Syntax highlighting based on your file's extension
- Tab completion to finish function names, file paths, arguments, and more.
- Multi-language code snippets to quickly use common blocks of code.
- Jump to function in file
- Change file type
- Working Directory
- Maximize, minimize panes
- Press **↑** to see command history
- Drag pane boundaries

R Support

Annotations for R Support pane:

- Import data with wizard
- History of past commands to run/copy
- Display .RPres slideshows **File > New File > R Presentation**
- Load workspace
- Save workspace
- Delete all saved objects
- Search inside environment
- Choose environment to display from list of parent environments
- Display objects as list or grid
- Displays saved objects by type with short description
- View in data viewer
- View function source code
- Create folder
- Upload file
- Delete file
- Rename file
- Change directory
- Path to displayed directory
- A file browser keyed to your working directory. Click on file or directory name to open.

Pro Features

Annotations for Pro Features pane:

- Share Project
- Active shared with Collaborators...
- Start new R Session in current project
- Close R Session in project
- Select R Version
- PROJECT SYSTEM**
- File > New Project**
- RStudio saves the call history, workspace, and working directory associated with a project. It reloads each when you re-open a project.
- Name of current project
- RStudio opens plots in a dedicated Plots pane
- Navigate recent plots
- Open in recent window
- Export plot
- Delete plot
- Delete all plots
- GUI Package manager lists every installed package
- Install Packages
- Update Packages
- Create reproducible package library for your project
- Click to load package with **library()**. Unlick to detach package with **detach()**
- Package version installed
- Delete from library
- RStudio opens documentation in a dedicated Help pane
- Home page of helpful links
- Search within help file
- Search for help file
- Viewer Pane displays HTML content, such as Shiny apps, RMarkdown reports, and interactive visualizations
- Stop Shiny app
- Publish to shinyapps.io, rpubs, RSConnect, ...
- Refresh
- View(<data>)** opens spreadsheet like view of data set

Debug Mode

Annotations for Debug Mode pane:

- Open with **debug()**, **browser()**, or a breakpoint. RStudio will open the debugger mode when it encounters a breakpoint while executing code.
- Launch debugger mode from origin of error
- Open traceback to examine the functions that R called before the error occurred
- Click next to line number to add/remove a breakpoint.
- Highlighted line shows where execution has paused
- Run commands in environment where execution has paused
- Examine variables in executing environment
- Select function in traceback to debug
- Step through code one line at a time
- Step into and out of functions to run
- Resume execution mode
- Quit debug mode

Version Control with Git or SVN

Annotations for Version Control pane:

- Turn on at **Tools > Project Options > Git/SVN**
- Stage files
- Show file diff
- Commit staged files
- Push/Pull to remote
- View History
- Added
- Deleted
- Modified
- Renamed
- Untracked
- Open shell to type commands
- current branch

Package Writing

Annotations for Package Writing pane:

- File > New Project > New Directory > R Package**
- Turn project into package. Enable roxygen documentation with **Tools > Project Options > Build Tools**
- Roxygen guide at **Help > Roxygen Quick Reference**



Interfaz de RStudio

1 LAYOUT

Move focus to Source Editor
Move focus to Console
Move focus to Help
Show History
Show Files
Show Plots
Show Packages
Show Environment
Show Git/SVN
Show Build

Windows/Linux	Mac
Ctrl+1	Ctrl+1
Ctrl+2	Ctrl+2
Ctrl+3	Ctrl+3
Ctrl+4	Ctrl+4
Ctrl+5	Ctrl+5
Ctrl+6	Ctrl+6
Ctrl+7	Ctrl+7
Ctrl+8	Ctrl+8
Ctrl+9	Ctrl+9
Ctrl+0	Ctrl+0

2 RUN CODE

Search command history
Navigate command history
Move cursor to start of line
Move cursor to end of line
Change working directory
Interrupt current command
Clear console
Quit Session (desktop only)
Restart R Session
Run current line/selection
Run current (retain cursor)
Run from current to end
Run the current function
Source a file
Source the current file
Source with echo

Windows/Linux	Mac
Ctrl+↕	Cmd+↕
↕/↕	↕/↕
Home	Cmd+←
End	Cmd+→
Ctrl+Shift+H	Ctrl+Shift+H
Esc	Esc
Ctrl+L	Ctrl+L
Ctrl+Q	Cmd+Q
Ctrl+Shift+F10	Cmd+Shift+F10
Ctrl+Enter	Cmd+Enter
Alt+Enter	Option+Enter
Ctrl+Alt+E	Cmd+Option+E
Ctrl+Alt+F	Cmd+Option+F
Ctrl+Alt+G	Cmd+Option+G
Ctrl+Shift+S	Cmd+Shift+S
Ctrl+Shift+Enter	Cmd+Shift+Enter

3 NAVIGATE CODE

Goto File/Function
Fold Selected
Unfold Selected
Fold All
Unfold All
Go to line
Jump to
Switch to tab
Previous tab
Next tab
First tab
Last tab
Navigate back
Navigate forward
Jump to Brace
Select within Braces
Use Selection for Find
Find in Files
Find Next
Find Previous
Jump to Word
Jump to Start/End
Toggle Outline

Windows/Linux	Mac
Ctrl+.	Ctrl+.
Alt+L	Cmd+Option+L
Shift+Alt+L	Cmd+Shift+Option+L
Alt+O	Cmd+Option+O
Shift+Alt+O	Cmd+Shift+Option+O
Shift+Alt+G	Cmd+Shift+Option+G
Shift+Alt+J	Cmd+Shift+Option+J
Ctrl+Shift+.	Ctrl+Shift+.
Ctrl+F11	Ctrl+F11
Ctrl+F12	Ctrl+F12
Ctrl+Shift+F11	Ctrl+Shift+F11
Ctrl+Shift+F12	Ctrl+Shift+F12
Ctrl+F9	Cmd+F9
Ctrl+F10	Cmd+F10
Ctrl+P	Ctrl+P
Ctrl+Shift+Alt+E	Ctrl+Shift+Option+E
Ctrl+F3	Cmd+E
Ctrl+Shift+F	Cmd+Shift+F
Win: F3, Linux: Ctrl+G	Cmd+G
W: Shift+F3, L:	Cmd+Shift+G
Ctrl+↔	Option+↔
Ctrl+↕/↕	Cmd+↕/↕
Ctrl+Shift+O	Cmd+Shift+O

4 WRITE CODE

Attempt completion
Navigate candidates
Accept candidate
Dismiss candidates
Undo
Redo
Cut
Copy
Paste
Select All
Delete Line
Select
Select Word
Select to Line Start
Select to Line End
Select Page Up/Down
Select to Start/End
Delete Word Left
Delete Word Right
Delete to Line End
Delete to Line Start
Indent
Outdent
Yank line up to cursor
Yank line after cursor
Insert yanked text
Insert <-
Insert %>%
Show help for function
Show source code
New document
New document (Chrome)
Open document
Save document
Close document
Close document (Chrome)
Close all documents
Extract function
Extract variable
Reindent lines
(Un)Comment lines
Reflow Comment
Reformat Selection
Select within braces
Show Diagnostics
Transpose Letters
Move Lines Up/Down
Copy Lines Up/Down
Add New Cursor Above
Add New Cursor Below
Move Active Cursor Up
Move Active Cursor Down
Find and Replace
Use Selection for Find
Replace and Find

Windows/Linux
Tab or Ctrl+Space
↕/↕
Enter, Tab, or ↵
Esc
Ctrl+Z
Ctrl+Shift+Z
Ctrl+X
Ctrl+C
Ctrl+V
Ctrl+A
Ctrl+D
Shift+[Arrow]
Ctrl+Shift+↔
Alt+Shift+↔
Alt+Shift+↔
Shift+PageUp/Down
Shift+Alt+↕/↕
Ctrl+Backspace

Tab (at start of line)
Shift+Tab
Ctrl+U
Ctrl+K
Ctrl+Y
Alt+↔
Ctrl+Shift+M
F1
F2
Ctrl+Shift+N
Ctrl+Alt+Shift+N
Ctrl+O
Ctrl+S
Ctrl+W
Ctrl+Alt+W
Ctrl+Shift+W
Ctrl+Alt+X
Ctrl+Alt+V
Ctrl+I
Ctrl+Shift+C
Ctrl+Shift+/
Ctrl+Shift+A
Ctrl+Shift+E
Ctrl+Shift+Alt+P

Mac
Tab or Cmd+Space
↕/↕
Enter, Tab, or ↵
Esc
Cmd+Z
Cmd+Shift+Z
Cmd+X
Cmd+C
Cmd+V
Cmd+A
Cmd+D
Shift+[Arrow]
Option+Shift+↔
Cmd+Shift+↔
Cmd+Shift+↔
Shift+PageUp/Down
Cmd+Shift+↕/↕
Ctrl+Opt+Backspace
Option+Delete
Ctrl+K
Option+Backspace
Tab (at start of line)
Shift+Tab
Ctrl+U
Ctrl+K
Ctrl+Y
Option+↔
Cmd+Shift+M
F1
F2
Cmd+Shift+N
Cmd+Shift+Opt+N
Cmd+O
Cmd+S
Cmd+W
Cmd+Option+W
Cmd+Shift+W
Cmd+Option+X
Cmd+Option+V
Cmd+I
Cmd+Shift+C
Cmd+Shift+/
Cmd+Shift+A
Ctrl+Shift+E
Cmd+Shift+Opt+P
Ctrl+T
Option+↕/↕
Cmd+Option+↕/↕
Ctrl+Option+Up
Ctrl+Option+Down
Ctrl+Option+Shift+Up
Ctrl+Opt+Shift+Down
Cmd+F
Cmd+E
Cmd+Shift+J

WHY RSTUDIO SERVER PRO?

RSP extends the the open source server with a commercial license, support, and more:

- open and run multiple R sessions at once
- tune your resources to improve performance
- edit the same project at the same time as others
- see what you and others are doing on your server
- switch easily from one version of R to a different version
- integrate with your authentication, authorization, and audit practices

Download a free 45 day evaluation at

www.rstudio.com/products/rstudio-server-pro/

5 DEBUG CODE

Toggle Breakpoint
Execute Next Line
Step Into Function
Finish Function/Loop
Continue
Stop Debugging

Windows/Linux	Mac
---------------	-----

Shift+F9	Shift+F9
F10	F10
Shift+F4	Shift+F4
Shift+F6	Shift+F6
Shift+F5	Shift+F5
Shift+F8	Shift+F8

6 VERSION CONTROL

Show diff
Commit changes
Scroll diff view
Stage/Unstage (Git)
Stage/Unstage and move to next

Windows/Linux	Mac
---------------	-----

Ctrl+Alt+D	Ctrl+Option+D
Ctrl+Alt+M	Ctrl+Option+M
Ctrl+↕/↕	Ctrl+↕/↕
Spacebar	Spacebar
Enter	Enter

7 MAKE PACKAGES

Build and Reload
Load All (devtools)
Test Package (Desktop)
Test Package (Web)
Check Package
Document Package

Windows/Linux	Mac
---------------	-----

Ctrl+Shift+B	Cmd+Shift+B
Ctrl+Shift+L	Cmd+Shift+L
Ctrl+Shift+T	Cmd+Shift+T
Ctrl+Alt+F7	Cmd+Opt+F7
Ctrl+Shift+E	Cmd+Shift+E
Ctrl+Shift+D	Cmd+Shift+D

8 DOCUMENTS AND APPS

Preview HTML (Markdown, etc.)
Knit Document (knitr)
Compile Notebook
Compile PDF (TeX and Sweave)
Insert chunk (Sweave and Knitr)
Insert code section
Re-run previous region
Run current document
Run from start to current line
Run the current code section
Run previous Sweave/Rmd code
Run the current chunk
Run the next chunk
Sync Editor & PDF Preview
Previous plot
Next plot
Show Keyboard Shortcuts

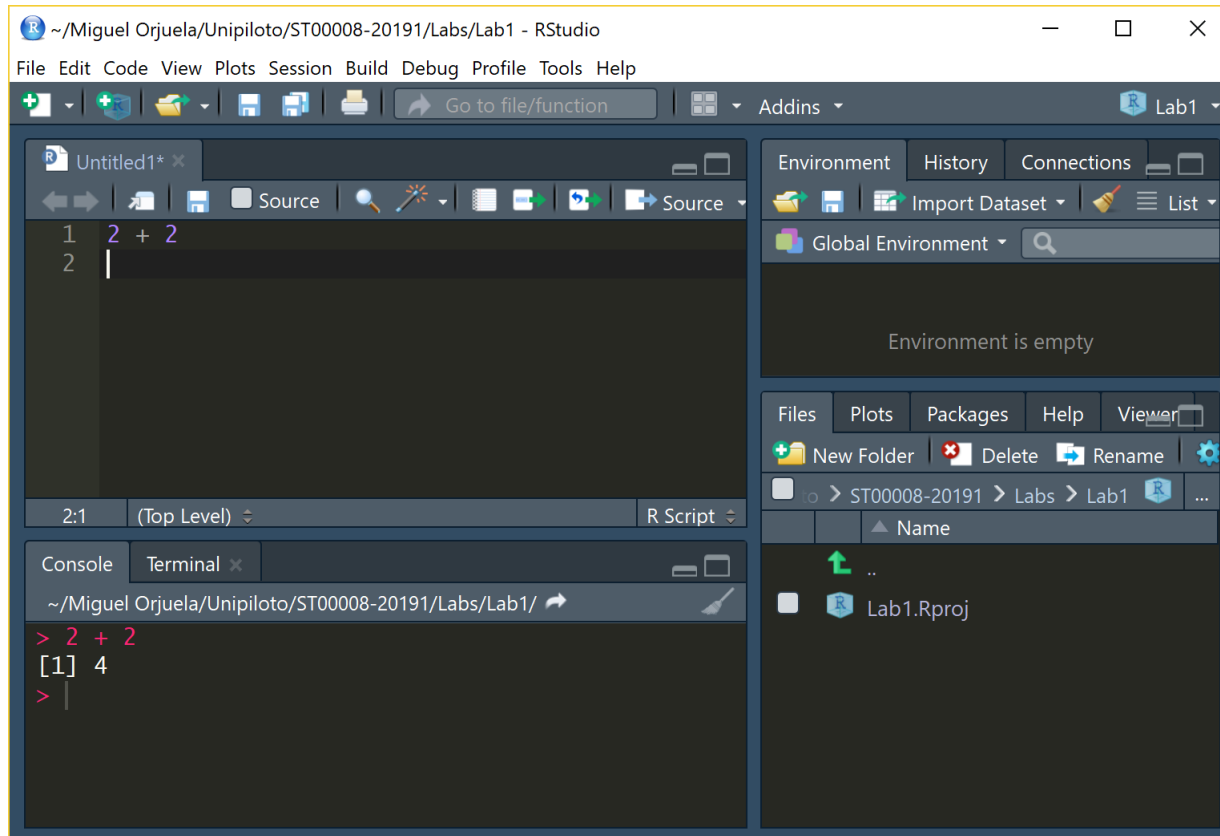
Windows/Linux	Mac
---------------	-----

Ctrl+Shift+K	Cmd+Shift+K
Ctrl+Shift+K	Cmd+Shift+K
Ctrl+Shift+K	Cmd+Shift+K
Ctrl+Shift+K	Cmd+Shift+K
Ctrl+Alt+I	Cmd+Option+I
Ctrl+Shift+R	Cmd+Shift+R
Ctrl+Shift+P	Cmd+Shift+P
Ctrl+Alt+R	Cmd+Option+R
Ctrl+Alt+B	Cmd+Option+B
Ctrl+Alt+T	Cmd+Option+T
Ctrl+Alt+P	Cmd+Option+P
Ctrl+Alt+C	Cmd+Option+C
Ctrl+Alt+N	Cmd+Option+N
Ctrl+F8	Cmd+F8
Ctrl+Alt+F11	Cmd+Option+F11
Ctrl+Alt+F12	Cmd+Option+F12
Alt+Shift+K	Option+Shift+K



Comandos rápidos

Interfaz de RStudio



Paso 1: Escriba $2 + 2$

Paso 2: Oprima Ctrl+Enter

Paso 3: Oprima Ctrl+L



Programación básica

Calculadora

Programación básica

```
2 + 2  
sqrt(9)  
log(10)
```


Ingreso de expresiones

■ Operador de asignación <-

Shortcut: Alt + - (tecla menos) inserta asignaciones

```
x <- 1  
print(x)  
x  
msg <- "hola"
```

■ La gramática del lenguaje determina si una expresión está completa o no

■ Comentarios con

```
# este es un comentario
```

Evaluación de expresiones

- Ingreso, evaluación, resultado impreso (autoprint)

```
x <- 5 # nada se imprime  
x # auto-printing  
print(x) # print explícito
```

- Secuencia de números

```
y <- 11:40  
y
```

Tipos de datos

Clases básicas o **atómicas** de objetos

- character
- numeric (números reales)
- integer
- complex
- logical (true/false)

Vectores y listas

- vector – objetos de la misma clase
- list – objetos de diferente clase

Tipos de datos

Programación básica

Tipo de dato	Descripción	Definición
Numeric	Números decimales	<code>numero <- 1.0</code>
Integer	Números enteros	<code>int <- 1</code>
Character	Cadenas de texto	<code>str <- "un texto"</code>
Complex	Números complejos	<code>comp <- 3+2i</code>
Logical	Verdadero (TRUE) o falso (FALSE). Es a menudo el resultado de operaciones lógicas.	<code>a <- 1; b <- 2; a < b</code>
Factor	Este no es estrictamente un tipo de dato, pero vale la pena describirlo aquí. Una variable factor es una variable categórica. Los vectores de caracteres a menudo se almacenan como factores para explotar funciones para tratar datos categóricos. Por ejemplo, en análisis de regresión.	Aplique <code>as.factor()</code> a un vector de caracteres.

Conocer el tipo de dato

```
class(x)  
typeof(x)  
str(x)
```

Comparaciones lógicas

Programación básica

Operador	Comparación	Ejemplo	Resultado
<	Menor que	5 < 3	FALSE
<=	Menor o igual que	5 <= 3	FALSE
>	Mayor que	5 > 3	TRUE
>=	Mayor o igual que	5 >= 3	TRUE
==	Exactamente igual que	5 == 3	FALSE
!=	No es igual que	5 != 3	TRUE

Conversión entre tipos

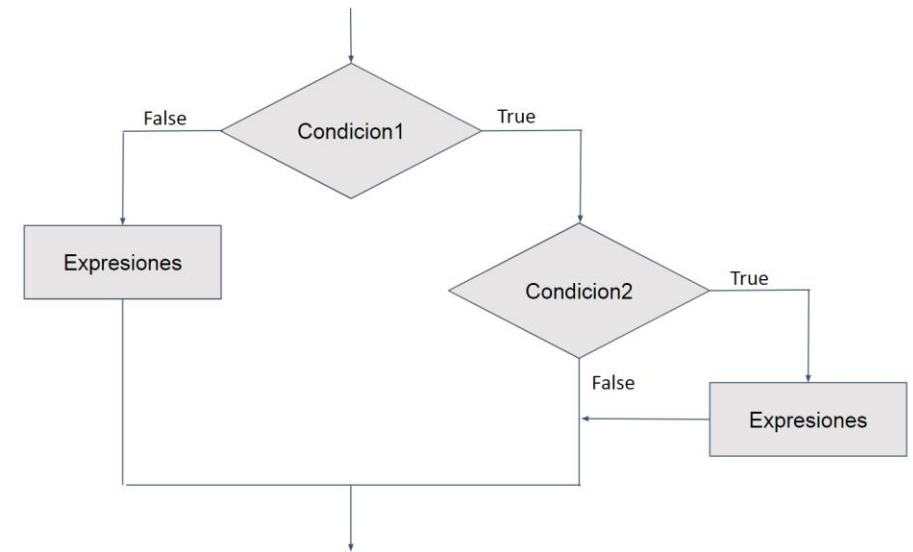
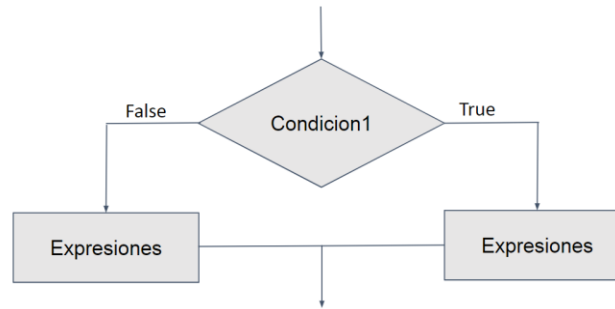
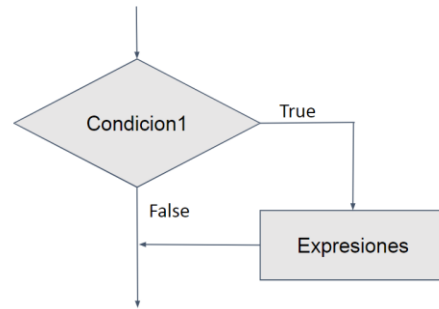
- Funciones para hacer casting entre tipos de variables

```
x <- 6  
class(x)  
as.numeric(x)  
as.logical(x)  
as.character(x)
```

- ¿Qué pasa cuando no hay forma razonable de transformar los datos?

```
x <- "a"  
as.numeric(x)  
as.logical(x)  
as.complex(x)
```

Instrucciones condicionales



Instrucciones condicionales

```
x <- runif(1,0,10)
```

```
if(<condition>) {  
  ## hacer algo  
} else {  
  ## hacer algo diferente  
}
```

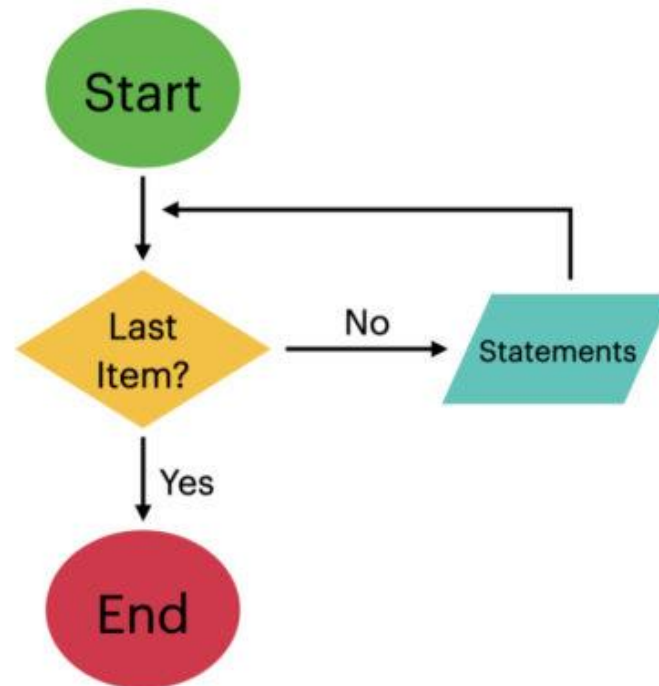
```
if(<condition>) {  
  ## hacer algo  
} else {  
  ## hacer algo diferente  
}
```

```
if(<condition1>) {  
  ## hacer algo  
} else if(<condition2>) {  
  ## hacer algo diferente  
} else {  
  ## hacer algo diferente  
}
```

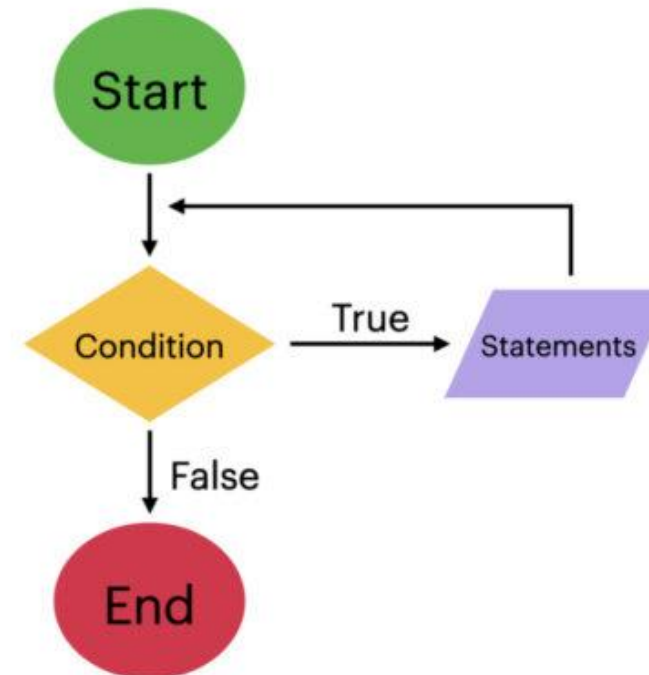
Ciclos



For Loop



While Loop



Ciclos



En R, los ciclos `for` toman una variable iteradora y le asignan valores sucesivos de una secuencia o vector.

Los ciclos `for` se usan más comúnmente para iterar sobre los elementos de un objeto (lista, vector, etc.)

```
for(i in 1:10) {  
  print(i)  
}
```

Ciclos

Los ciclos `while` inician verificando una condición, si se cumple, su bloque de código se ejecuta. Una vez ejecutado, se vuelve a verificar, y así sucesivamente hasta que la condición evalúa a falso.

```
count <- 0

while(count < 10) {
  print(count)
  count <- count + 1
}
```

Funciones

- Forma de dar nombre a un bloque de código para agruparlo y reutilizarlo

```
saludar <- function()  
{  
  print("hola")  
}  
saludar()
```

Sin retorno

```
sumar_dos <-  
function(numero) {  
  return(numero + 2)  
}  
sumar_dos(4)
```

Con retorno

Gracias por tu asistencia y participación 😊

Contacto

✉ miguela.orjuela@urosario.edu.co

🌐 <https://www.linkedin.com/in/miguel-orjuela/>

🐙 <https://github.com/maorjuela73>

Links de interés

- <https://r4ds.had.co.nz/>
- https://bookdown.org/chescosalgado/intro_r/