

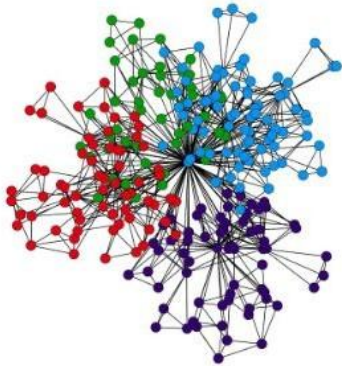
Taller de R


Mg. Ing. Layla Scheli

Introducción a R y RStudio

¿Qué es ?

Es un lenguaje para el procesamiento y análisis estadístico de los datos.



Descarguemos 
paso a paso

Descarguemos R:

1. Ingresar a la página www.r-project.org.
2. Hacer click en "download R" (descargar R) en la mitad de la página después de "Getting Started."
3. Seleccionar la localidad de CRAN correspondiente a tu país.
4. En la solapa "Download and Install R" clickéa según tu sistema operativo.

¿Qué es



?

Es uno de los entornos más cómodos para utilizar el lenguaje R

Tiene ciertas ventajas en su uso:

1. Autocompletado.
2. Resaltado de sintaxis.
3. Documentación y ayuda integrada.
4. Posibilidad de realizar informes con RMarkdown y de trabajo colaborativo

Descarguemos Rstudio:

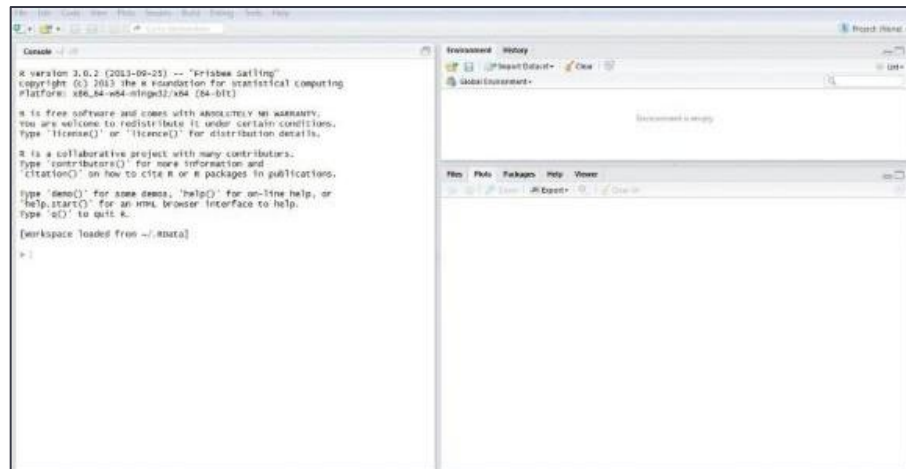
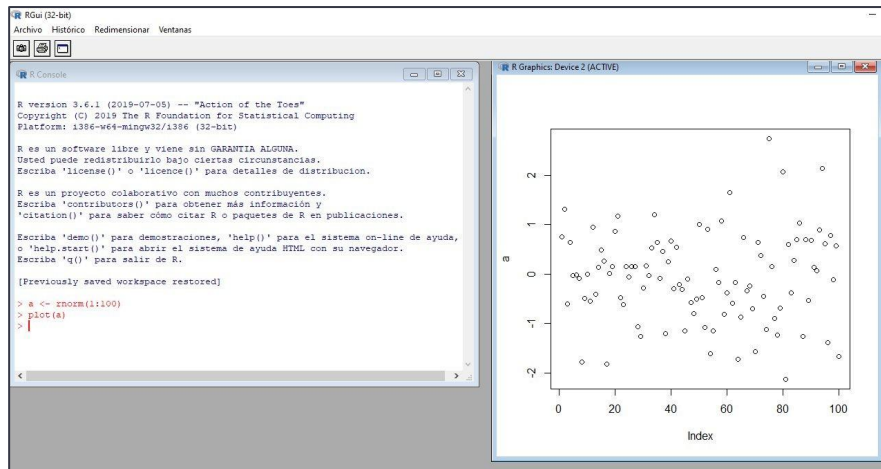
RStudio Desktop 2021.09.0+351 - [Release Notes](#)

1. Install R. RStudio requires R 3.0.1+.
2. Download RStudio Desktop. Recommended for your system:



Requires Windows 10 (64-bit)






```
1 a <- rnorm(1:100)
2 plot(a)
3
```

Editor de texto

Console Terminal Jobs

D:/Guada/Clases/EANT/Social Data Analytics/Clase 1/Clase_1/

R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (c) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

```
> a <- rnorm(1:100)
> plot(a)
```

Console

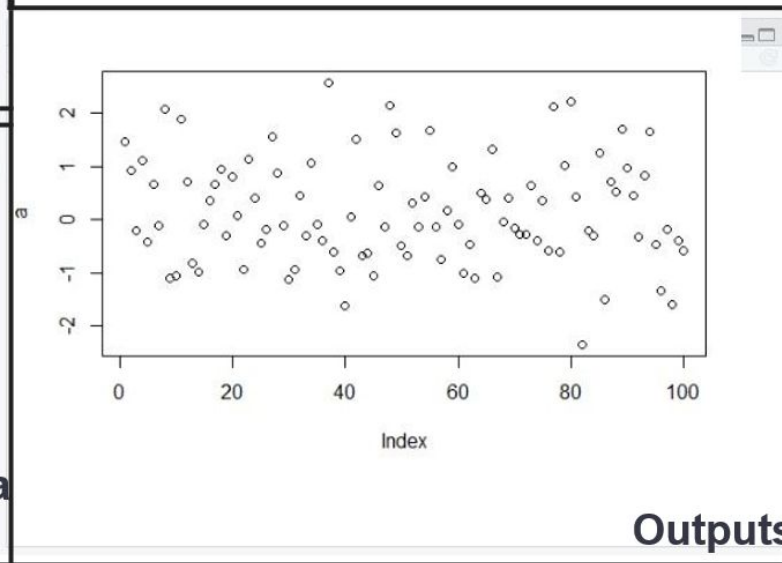
Environment History Connections

Global Environment

Values

a	num [1:100] 1.475 0.936 -0.196 1.105 -0.41 ...
---	--

Environment



Outputs

Error vs Warnings



Esto **ES** un error:

```
> install.packages("tidyverse")  
Error in (function (srcref) : unimplemented type (29) in 'eval'  
Error: no more error handlers available (recursive errors?); invoking 'abort' restart
```

Esto **NO** es un error:

```
> install.packages("leaflet")  
Installing package into 'C:/Users/Usuario/Documents/R/win-library/4.0'  
(as 'lib' is unspecified)  
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/leaflet_2.0.3.zip'  
Content type 'application/zip' length 2335048 bytes (2.2 MB)  
downloaded 2.2 MB
```

Variables



Son ***espacios reservados en la memoria*** que, como su nombre indica, pueden cambiar de contenido a lo largo de la ejecución de un programa

Pueden ser

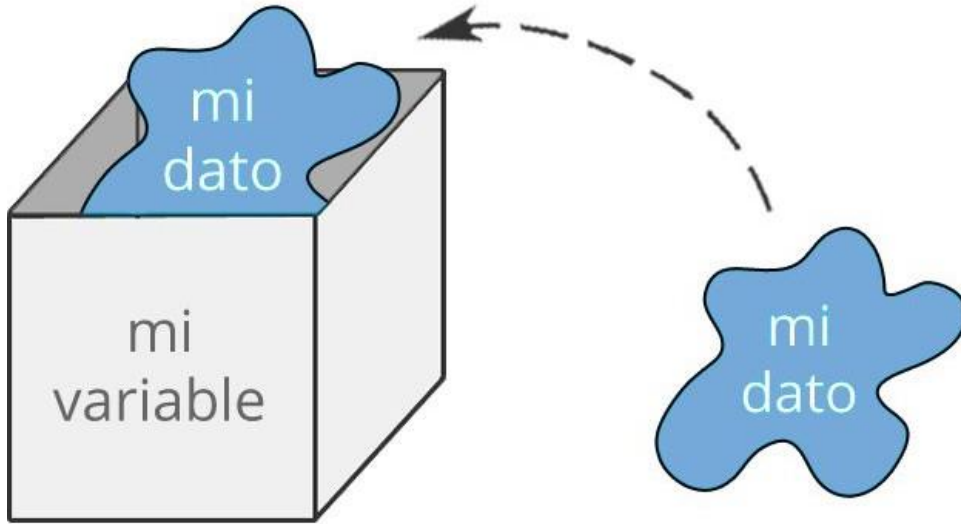
- Números
- Cadenas de texto
- Elementos complejos
- Funciones



Tipo de variables

Tipo	Ejemplo
numeric	3.14
character	"Hola, ¿cómo estás?"
factor	"Femenino", "Masculino", "Otr@"
logical	TRUE / FALSE
missing o no disponibles	NA

Asignación de variables



1

nombre <- objeto

2

nombre = objeto

3

objeto -> nombre

```
##{r}
#mi variable "x" tiene el valor 1
x <- 1
x
##
```

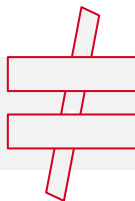
Atención!



1

R es case sensitive (distingue mayúsculas y minúsculas)

HOLA



hola

Directorio de trabajo

- `getwd()` → Preguntamos cuál es nuestro directorio de trabajo
- `setwd(<nuestra ruta>)` → Establecemos nuestro directorio de trabajo

*Tener en cuenta en Windows que \ es el carácter de escape, por lo tanto, al copiar una ruta se debe reemplazar \ por / o \\. Ejemplo C:/R o C:\\R

Vectores

R utiliza diferentes estructuras de datos. La estructura más simple es el vector, que es una colección ordenada de elementos. El vector, como todos los elementos en R, es un objeto. En un vector, todos sus elementos son del mismo tipo de dato.

La función `c()` nos permite construir un vector. Una vez que creamos un vector, es muy habitual querer guardar ese vector que acabamos de crear en un nuevo objeto. Para eso deberemos entonces, recurrir a lo que se conoce como “asignación” (`<-`).

La sintaxis para crear un objeto es:

ObjetoNombre <- valor

Clases de Objeto

- Vector.
- Matrices.
- Factores.
- DataFrame.



Funciones

Las funciones son secciones de código reutilizables que sirven para un propósito en particular. Las funciones pueden tomar entradas y salidas, y pueden ser reutilizadas a lo largo del programa. Organizar los programas en funciones ayuda a organizar y simplificar el código.

Es un ejemplo de abstracción: después de que has escrito una función, puede usar la función sin tener que preocuparte acerca de los detalles de cómo la función es implementada.

Gracias a la abstracción, otros pueden usarla (o “llamar”) a la función sin saber sus detalles en un nivel más profundo.

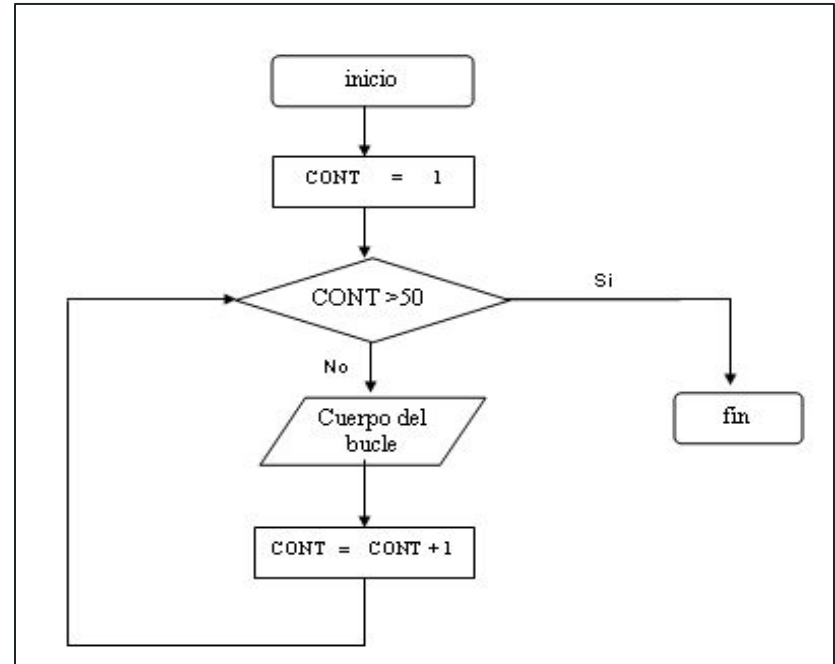
Funciones

Ejemplo:

```
1  mifuncion <- function(argumento1, argumento2)
2  {
3    #Cuerpo
4    #Resultado|
5  }
6
```

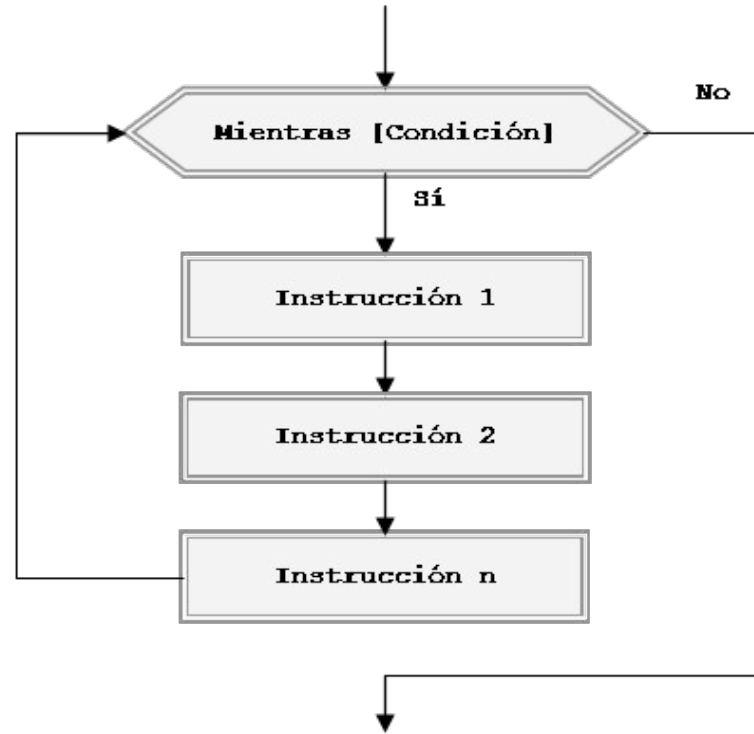
Estructuras de Repetición

for (i in expresión) expresión



Estructuras de Repetición

while (expresión condicional) expresión:



Estructuras de Repetición

Repeat expresión:

La estructura repeat ejecuta un bucle infinitamente. Por lo general, no es utilizada para realizar análisis, sino cuando se realiza programación. La única forma de terminar o romper el ciclo es con el break.

```
1 i = 0
2 repeat {
3   print(i)
4   i <- i+1
5   if (i==5) break
6 }
```

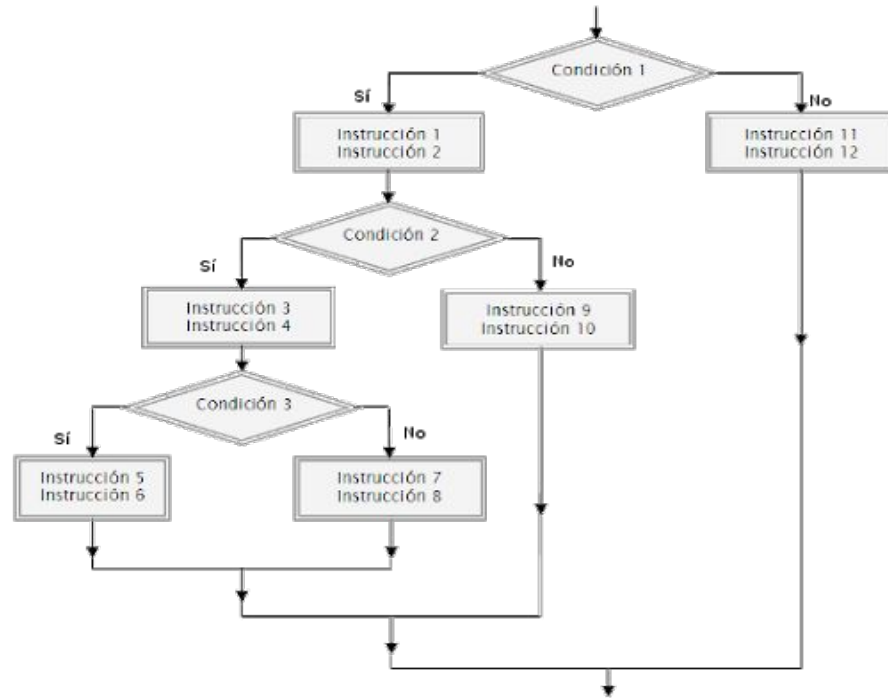
Estructuras Condicionales

Estructura básica:

If (expresión condicional)
 expresión

else
 expresión

Ifelse (expresión condicional,
expresión verdadera,
expresión falsa)



¡Gracias!

Contacto: layla.scheli@gmail.com

Linkedin: <https://www.linkedin.com/in/laylascheli/>