

Introducción a R

Ángelo Gutiérrez Daza

Banco de Guatemala

2020

Introducción

- ▶ Qué es R?
 - ▶ Un lenguaje y un entorno para computación estadística y elaboración de gráficas
 - ▶ Ampliamente utilizado para minería y análisis de datos
 - ▶ Potentes librerías para análisis estadístico y numérico

¿Por qué R?

▶ **Bueno:**

- ▶ Un lenguaje de programación completo
- ▶ Sencillo, flexible y poderoso
- ▶ Un sin fin de librerías disponibles para todas las necesidades
 - ▶ Ver <https://cran.r-project.org/web/packages/index.html>

▶ **Bonito:**

- ▶ Uno de los paquetes estadísticos más usados en el mundo
- ▶ Gran comunidad de usuarios
- ▶ Gráficas con calidad de publicación

▶ **Barato:**

- ▶ ***Costo monetario:*** Cero
 - ▶ *Open Sorce*
- ▶ ***Costo de oportunidad:*** Bajo
 - ▶ *Relativamente fácil de aprender*

Open Source = Gratis?

- ▶ Si: Cualquiera puede ir a la página web y bajarse R gratis
- ▶ Pero también implica más cosas:
 - ▶ Acceso completo a el código original
 - ▶ Capacidad para extender el software y corregir errores
 - ▶ Una herramienta gratuita y accesible a todos, lo que promueve resultados **replicables**
 - ▶ Participación de estadísticos y científicos de primera línea implementando los métodos más recientes
 - ▶ Especialmente en campos de alta demanda como el ***data mining***

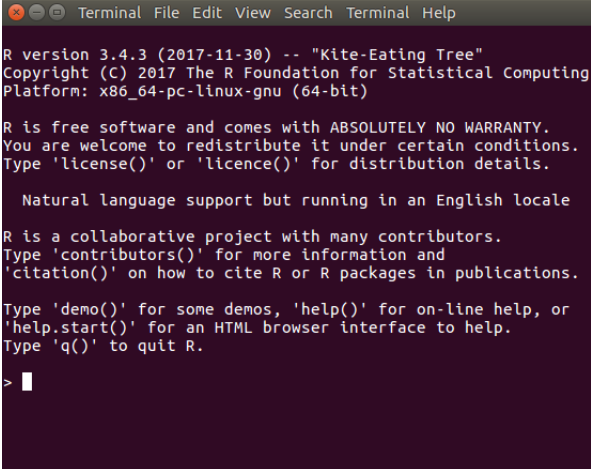
Obtener R

- ▶ Se puede descargar la instalación base desde cualquier servidor miembro del “Comprehensive R Archive Network”, o CRAN:

<https://cran.r-project.org/mirrors.html>

- ▶ Versión actual: **3.6.2 (2019-12-12) -- "Dark and Stormy Night"**

R Console (Linux)



```
Terminal File Edit View Search Terminal Help

R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.

  Natural language support but running in an English locale

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.

> █
```

Mis primeros pasos en R

- ▶ **Nuestro primer comando:** Imprimir en la pantalla "Hello World"
 - ▶ `print("Hello World")`
- ▶ **Nuestro segundo comando:** Crear la variable "a" cuyo valor es "1"
 - ▶ `a <- 1`
- ▶ **Nuestro tercer comando:** Crear el vector fila $A = [1, 2]$
 - ▶ `A <- cbind(1, 2)`

R Console (Linux)

```
Terminal File Edit View Search Terminal Help

R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.

  Natural language support but running in an English locale

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.

> print("Hello World")
[1] "Hello World"
> a <- 1
> A <- cbind(1,2)
> 
```


Dar instrucciones a R

- ▶ Hemos dado **instrucciones** a R
- ▶ En particular, hemos dado dos tipos de comandos básicos:
 - ▶ **Expresiones:** se evalúa e imprime una instrucción
 - ▶ Ej. `print("Hello World")`
 - ▶ **Asignaciones:** el resultado no se imprime automáticamente
 - ▶ Ej. `a <- 1`
- ▶ **Múltiples comandos:** Los comandos se pueden separar por ; o por una nueva línea
- ▶ Si un comando queda incompleto, la consola de R mostrará un +
- ▶ **Añadir comentarios:** Todo lo que se escriba después del # será tomado por R como un comentario y lo ignorará a la hora de ejecutar una instrucción

Asignación de Variables

- ▶ El nombre de las variables debe empezar por una letra y no debe contener espacios
- ▶ R es sensible a mayúsculas y minúsculas: `PARCE`, `Parce` y `parce` son diferentes variables!
- ▶ Después de asignar una variable, podemos ingresar el nombre de la variable a la consola para imprimir su contenido
 - ▶ Si la variable no ha sido asignada, R nos lo indicara con un error

El *Workspace*

- ▶ Es el conjunto de objetos definidos por el usuario en una sesión:
 - ▶ Vectores, matrices, data frames, listas, funciones, etc
- ▶ Podemos usar el comando `ls()` para ver las variables en el *workspace*
- ▶ También podemos guardar el *workspace* actual para utilizarlo más adelante en otra sesión usando `save.image("AlgunNombre")`
 - ▶ El archivo se guarda con la extensión **.RData** por defecto
- ▶ Podemos cargar dicho workspace usando `load("AlgunNombre.RData")`

R Console (Linux)

```
Terminal File Edit View Search Terminal Help

R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.

  Natural language support but running in an English locale

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.

> print("Hello World")
[1] "Hello World"
> a <- 1
> A <- cbind(1,2)
> a
[1] 1
> A
      [,1] [,2]
[1,]    1    2
> B
Error: object 'B' not found
> ls()
[1] "a" "A"
> 
```

Historial de Comandos

- ▶ En la consola, la tecla ▲ permite ver comandos ingresados anteriormente
- ▶ También podemos utilizar la función `history()` para ver los últimos 25 comandos ingresados
- ▶ Podemos guardar nuestra historia de comandos usando
 - ▶ `savehistory(file="AlgunNombre")`
- ▶ La extensión del archivo, por defecto, es **.Rhistory**
- ▶ Podemos recuperar la historia de comandos usando
 - ▶ `loadhistory(file="AlgunNombre")`

El Directorio de Trabajo

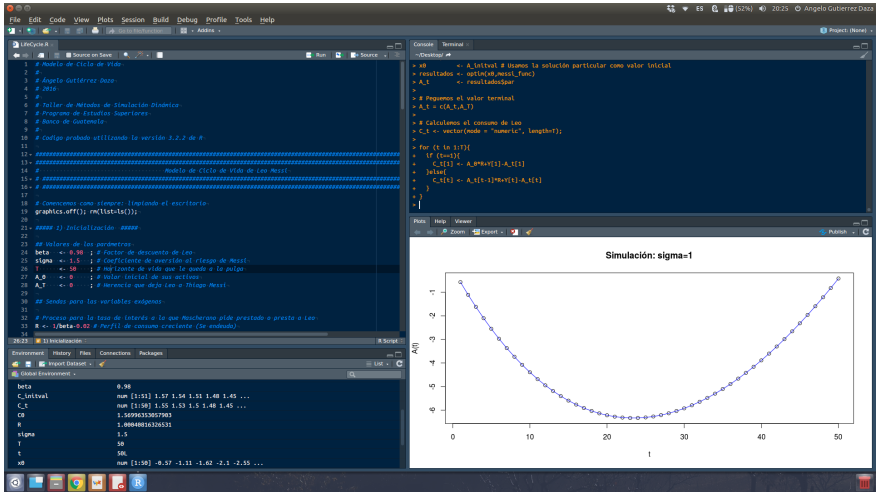
- ▶ Si se quiere leer o escribir archivos en una carpeta en particular, se debe especificar el *directorio de trabajo* de R
- ▶ El comando `getwd()` nos indica cual es el directorio de trabajo actual
- ▶ El comando `setwd()` nos permite cambiar de directorio:
 - ▶ Ejemplo: `setwd("c:/Usuario/Angelo")`

Editor de Texto

- ▶ Trabajar directamente desde la consola no es eficiente
- ▶ Es mejor crear archivos de texto que permitan la reproducción y documentación de actividades: *scripts*
- ▶ Para ello, necesitamos tener a la mano un buen editor o un entorno de desarrollo integrado:
 - ▶ **Mala idea:** Word, Bloc de Notas
 - ▶ **Buena idea:** Vim o Emacs (si ya los conocen o quieren sufrir)
 - ▶ **Excelente idea:** Sublime Text, Atom, **Visual Studio Code**

Entorno de Desarrollo Integrado (IDE)

- ▶ Muchos programadores prefieren emplear un **entorno de desarrollo integrado** (IDE)
- ▶ Son programas que agrupan:
 - ▶ Un editor sofisticado para escribir código
 - ▶ Un compilador y un depurador
 - ▶ Muchas cosas más: Visualización de figuras y el workspace, listado de comentarios...
- ▶ Estos permiten tener todo al alcance de un comando y programar de forma interactiva
- ▶ El IDE que recomiendo para R es **RStudio**
- ▶ Se puede descargar gratis en
 - ▶ <https://www.rstudio.com/products/rstudio/download/>



Lo que sigue

1. **Introducción a RStudio**
2. **Manipulación de Datos en R**
3. **Funciones**
4. **Gráficas en R**
5. **Estructuras de Control**
6. **Ánalysis Numérico en R**
7. **RMarkdown**