10 CONSEJOS PARA EMPEZAR CON R SIN MORIR EN EL INTENTO

Utiliza una user interface!

R es un programa fantástico de cálculo estadístico y mucho más… pero R por sí solo es muy difícil que lo manejes. La user interface es muy densa. Necesitas una ayuda. ¡Necesitas una user interface! Principalmente existen dos: RCommander y RStudio.

Y la pregunta que me harás es, ¿y cuál elijo? Te lo pongo fácil:

Crees que vas a necesitar hacer análisis estadísticos sencillos. Sin complicaciones. ¿No quieres programar? Utiliza RCommander. Podrás hacer análisis estadísticos del estilo SPSS, MINITAB

¿Quieres hacer análisis personalizados y automatizados? Utiliza RStudio. Necesitas aprender a programar pero con esta user interface tendrás más que de sobras para hacer análisis muy muy interesantes. Análisis automatizados y gráficos espectaculares.

Como R+Rstudio juntos pueden hacer muchas más cosas que con RCommander he elegido RStudio.
En esta guía encontrarás recomendaciones del dúo R+RStudio.

Sigue un camino de aprendizaje marcado

Uno de los problemas que he detectado y que a mí me ha pasado es perder tiempo en buscar formacion por internet como un loco y no dar con el blanco. Por este motivo te he preparado un listado de recursos adaptados a cada nivel de aprendizaje.

Antes de empezar con R llévate contigo un post excepcional que te servirá de referencia para que vayas aprendiendo poco a poco.
No hace falta que busques en internet los mejores recursos. Aquí te he resumido la mejor ruta de 7 pasos para conseguir que aprendas en R y convertirte en un data scientist.

Hazme caso y échale un vistazo. ¡Mejor aún! Si vas en serio con R. Deja el post en favoritos y sácale el mejor partido.

>> ACCESO AL POST – 7 PASOS PARA CONVERTIRTE EN UN DATA SCIENTIST EN R

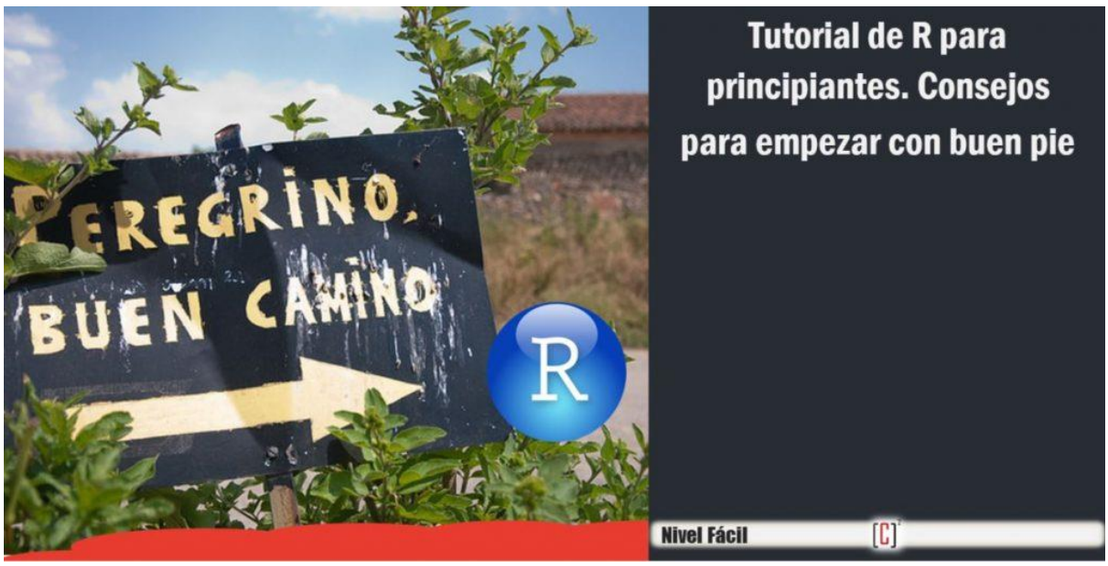
Aprende a abrir un código y ejecútalo

Antes de empezar mejor tener una visión panorámica de R+RStudio. ¿Cómo puedes conseguirlo? Te lo pongo fácil.

Te propongo otro artículo que te va ayudar a entender que es R y RStudio. Te voy a dar las bases fundamentales para que aprendas esta interfaz.

- Cómo está estructurada
- Cómo cargar un script o código de R
- Dónde están todos los aspectos visuales de RStudio. (datos, código, gráficos, el help…)

Una de los aspectos más importantes es saber abrir un archivo.R. En este artículo te voy a enseñar a abrir tu primer Script de R y ejecutarlo. Y algún regalo más.



>> ACCESO AL POST – TUTORIAL DE R PARA PRINCIPIANTES

Un código para gobernarlos a todos

Siempre es mejor trabajar desde la base. Si eres novato total, este código, te va a venir genial para aprender la sintaxis más básica del lenguaje R. Es un código todoterreno que te permite tener las estructuras básicas más utilizadas cuando empieces a programar.

```
#*********************************************************************
#*********************************************************************
#           MATERIAL DESCARGABLE DE CONCEPTOS CLAROS
#*********************************************************************
#*********************************************************************
# Autor: Jordi Ollé Sánchez
# Fecha: 08/02/2016
# E-mail: jordi@conceptosclaros.com
# Explicación: Este códgio permite ver las secuencias de código más utilizadas
# en la práctica
#
# ESTRUCTURA DEL CÓDIGO
#*********************************************************************
# 1.VECTORES, MATRICES Y COMANDOS BÁSICOS
# 2. ITERACIÓN Y CONDICIONAL
# 3. ITERACIÓN Y GRÁFICO

# ¿QUÉ HACE ESTE CÓDIGO?
#*********************************************************************
# Define variables tipo valor, tipo vectores, tipo matriz. Vas a ver
# cómo puedes calcular la longitud del vector, el número de filas
# el número de columnas de una matriz. VAs a aprender a acceder a un
# vector y una matriz. Do ejemplos de aplicación con iteración y condicional.

# El código empieza aqui...

#*********************************************************************
# 1. VECTORES, MATRICES Y COMANDOS BÁSICOS
#    Calcula los valores divisibles por 3 del 1 al 130
#*********************************************************************

# Asignar un valor a una variables llamada "a"
a <- 10
# Definir una secuencia de 0 a 2 con pasos de 0.01
z <- seq(0,5,0.02)
# Acceder a la segunda posición del vector
z[2]
# La suma de todos los componentes del vector
sum(z)
# La media del vector
mean(z)
# Dibujar los puntos del vector
plot(z)
# Dibujar la linea de un vector
plot(z,type='l')
# Calcular la longitud del vector
length(z)
# Definir un vector a partir de los valores dados
m <- c(0.3,9,4.5,9)
# Definir una matriz a partir de los valores de la matrix
# nrow indica el número de filas y ncol el número de columnas
A <- matrix(data=c(3,-2,4,9),nrow=2,ncol=2)
# Calcular el número de filas
nFilas <- nrow(A)
# Calcular el número de columnas
nCol <- ncol(A)
# Acceder al valor de la matriz de la fila 2 y la columna 1
A[2,1]

#*********************************************************************
# 2. ITERACIÓN Y CONDICIONAL
#    Calcula los valores divisibles por 3 del 1 al 130
#*********************************************************************

# Definir un vector vacio
s <- vector()
# Definir un contador que estará dentro del condicional
j <- 0
# Crear una iteración for con un contador "i" que va de 1 a 130
for(i in 1:130) {
 # Condicional que te dice si la i es divisible por 2 y lo guarda en un vector s
 if (i%%3 == 0) {
   print(i)
   j = j+1
   s[j] = i
   }
 else{
```

```
 }
}
# Calcular la longitud del vector de los valores divisibles por 2
N = length(s);

#*****************************************************************
# 3. ITERACIÓN Y GRÁFICO
#    Calcula el área de un círculo de radio  0 a 25 y lo dibuja
#*****************************************************************
# Calcula el área de un ciruclo de radio r
# El radio r va desde 0 a 25
r <- seq(0,5,0.02)
A <- vector()
for (i in 1:length(r)) {
 A[i] <- pi*r[i]^2
}
# Dibuja el area del círculo en función del radio
plot(r,A,type='l')
```

El ABC de la sintaxis en R

Te he preparado otro código genial. Esta vez con la sintaxis de las estructuras de R más típicas para que las tengas siempre mano. Te servirá de plantilla cuando quieres programar con condicionales, iteraciones etc…

El uso de las estructuras, las sintaxis son básicas para crear un gran código.

Siempre pierdo mucho tiempo en acordarme cómo eran las estructuras y supongo que tu también. ¡Este código nos va a venir genial y no ahorrará mucho tiempo!

Condicional if…

```
if (test_expression) {
  statement
}
```

Ejemplo condicional if…

```
x <- 5
if(x > 0){
  print("Positive number")
}
```

Condicional if…else

```
if (test_expression) {
  statement1
} else {
  statement2
}
```

Ejemplo condicional if…else

```
x <- -5
if(x > 0){
  print("Non-negative number")
} else {
  print("Negative number")
}
```

Condicional mutiples else if…

```
if ( test_expression1) {
  statement1
} else if ( test_expression2) {
  statement2
} else if ( test_expression3) {
  statement3
} else
  statement4
```

Ejemplos condicional multiples else if…

```
x <- 0
if (x < 0) {
  print("Negative number")
} else if (x > 0) {
  print("Positive number")
} else
  print("Zero")
```

Iteración for…

```
for (val in sequence)
{
   statement
}
```

Ejemplo iteración for…

```
x <- c(2,5,3,9,8,11,6)
count <- 0
for (val in x) {
   if(val %% 2 == 0)  count = count+1
}
print(count)
```

Iteración con condición while…

```
while (test_expression)
{
  statement
}
```

Ejemplo while….

```
i <- 1
while (i < 6) {
  print(i)
  i = i+1
}
```

Ejemplo break…

```
x <- 1:5
for (val in x) {
   if (val == 3){
       break
   }
   print(val)
}
```

Repetición repeat…

```
repeat {
  statement
}
```

Ejemplo repetición repeat…

```
x <- 1
repeat {
  print(x)
  x = x+1
  if (x == 6){
      break
  }
}
```

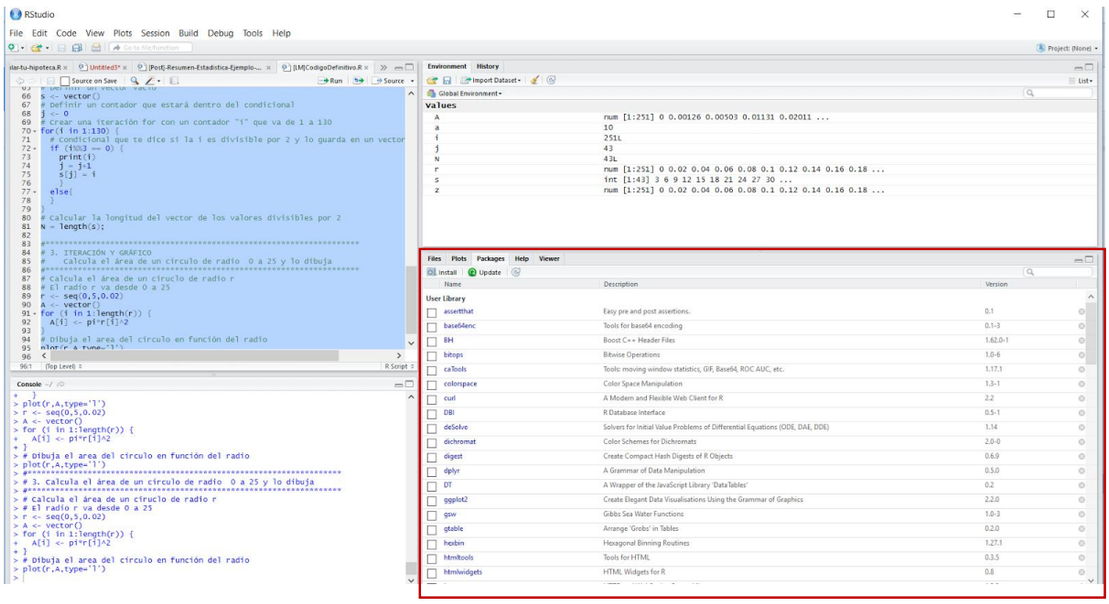
Cargar e instalar funciones es crucial

Una de las funcionalidades de R más interesantes es la opción de cargar paquetes de funciones o packages. Esto te permite ampliar las posibilidades de R. La mayoría de paquetes han sido validados y creados por expertos de la comunidad científica.

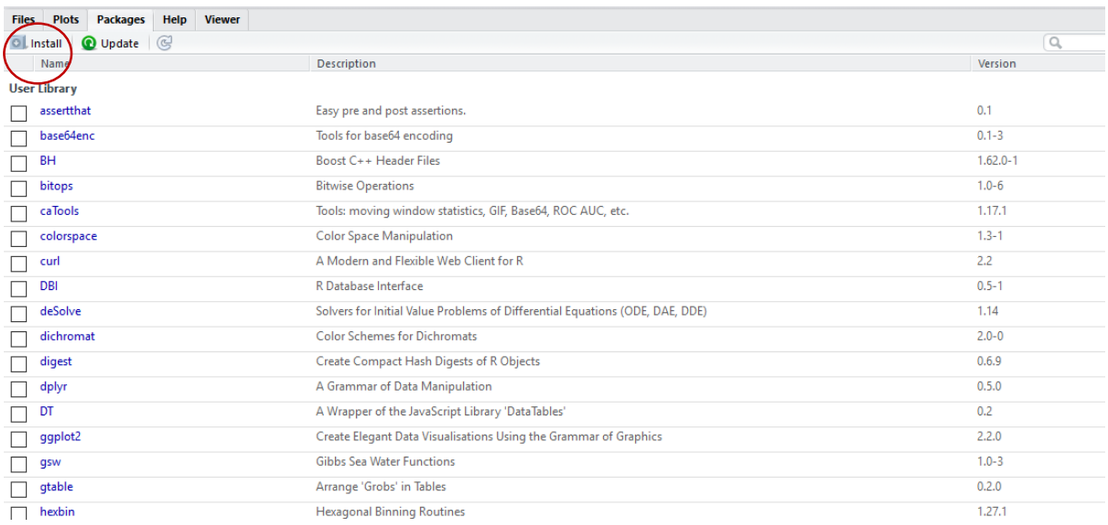
Puedes encontrar el listado de paquetes disponibles aquí.

Pero para utilizarlos es muy importante que sepas cargarlos y cómo hacerlo. Aquí unos pasos.

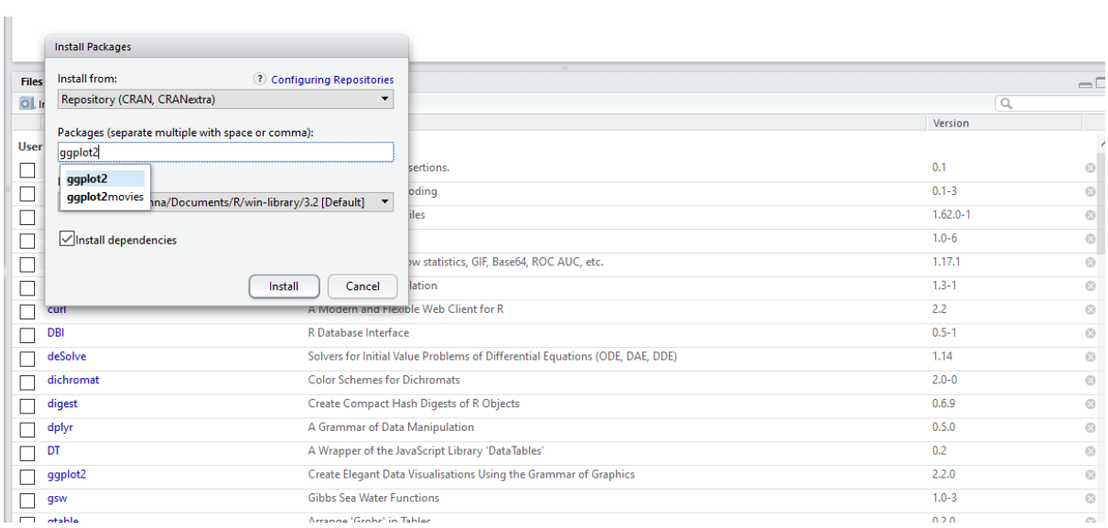
PASO1- Ves a la ventana de visualización y dale a Pakages:



PASO2- Instalar el paquete si no lo tienes instalado

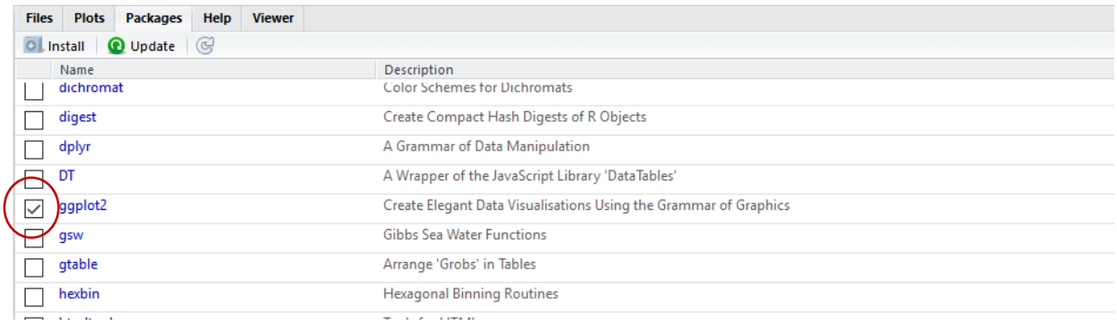


PASO3- Escribe el nombre del paquete, lo seleccionas y lo instalas



PASO4- Cargar el paquete una vez instalado

Haces un click al paquete que quieres cargar.



Automatiza la instalación y carga de paquetes de funciones y ahorrarás mucho tiempo

Si me has caso. Y has acabado el punto 2 ya habrás visto que te he dejado un código para cargar e instalar paquetes de forma automática. De esta forma te evitas cargarlos manualmente un y otra vez cuando arranques R.
Este código te va ahorrar tiempo y trabajo inútil.

Mi recomendación es que lo pongas al principio de cada código que generes en R. De esta forma vas a poder manejar muy cómodamente los paquetes en R. Nunca más tendrás problemas con la carga de paquetes.

Igualmente te dejo el código aquí abajo:

```
#*********************************************************************
# 1.1 INSTALAR PAQUETES DE FUNCIONES
#*********************************************************************
# Lista de paquetes de funciones a instalar
.packages = c("ggplot2", "plotly", "xlsx","scales")
# Instala los paquetes sinó los tienes instalados
.inst <- .packages %in% installed.packages()
if(length(.packages[!.inst]) > 0) install.packages(.packages[!.inst])
#*********************************************************************
# 1.2 CARGAR PAQUETES O CREAR FUNCIONES
#*********************************************************************
# Carga los paquetes sinó los tienes cargados
lapply(.packages, require, character.only=TRUE)
```

No pierdas de vista dónde se guardan los paquetes de funciones instaladas

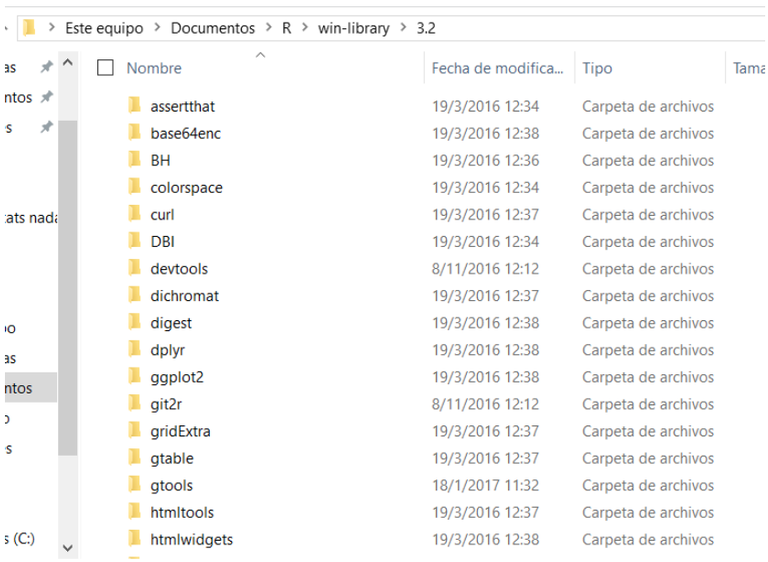
Unos de los aspectos que no era muy consciente en mis inicios es dónde se alojan los paquetes que voy instalando y descargando.

Es muy interesante que sepas dónde se guardan los paquetes de funciones. O lo que es lo mismo, dónde se instalan todos las paquetes que descargues.

De esta manera podrás tener controladas las librerías que vayas instalando y las versiones de las mismas.

La librería de funciones por defecto se instalan en Documentos >> R

En esta ubicación se guardaran todas las versiones de R que te descargues.
Por cada versión nueva tendrás los paquetes de funciones que has descargado.



Dentro de cada carpeta podrás encontrar la funciones que utiliza R, ejemplos, el manual.
Todos paquetes que aparecen en RStudio los puedes ver en esta ruta.

Acude al foro stack overflow y tendrás respuestas de dudas concretas de programación

El mundo de R tiende a infinito. Igual que internet. Por eso tenemos que ser eficaces en nuestras búsquedas. El mejor sitio para responder dudas concretas de programación y utilización de paquetes es stack overflow.

Stack overflow es de lejos el mejor foro y el más completo para resolver dudas concreta de R. Te va a venir genial hacer búsquedas en inglés tipo:

Tu duda en inglés + stack overflow

Seguro que tu duda ha sido solucionada antes por unos cuantos. Ahora no tienes excusa de ponerte a programar.

Completa la vídeo guía de R que te he preparado

Si no lo has hecho aún no sé a qué esperas.

Te he preparado una vídeo guía genial y te va ayudar a dar el salto con R a base de solucionar pequeños ejemplos. Te va a venir fantástico. Hazme caso. Acaba la video guía y tendrás las bases de R bien asentadas.

¿Me quieres ayudar?

[C]² Conceptos Claros

COMPARTE EL BLOG EN TUS REDES
ME AYUDARÁS A CRECER

¡Hola! Soy Jordi, y ayudo a **investigadores en busca de la excelencia** a aplicar herramientas matemáticas para aprovechar sus datos y tomar mejores decisiones. Su confianza aumentará y sus investigaciones tendrán el rigor técnico que tanto anhelan. Más sobre mi"