

The R logo is a large, stylized graphic in the background, composed of several blue and yellow rectangular blocks arranged in a 3D-like structure. It is positioned on the left side of the slide, behind the main title.

ANÁLISIS DE DATO CON R



contáctenos: enei@inei.gob.pe / 433-3127



Pasaje Hernán Velarde 285 Lima.

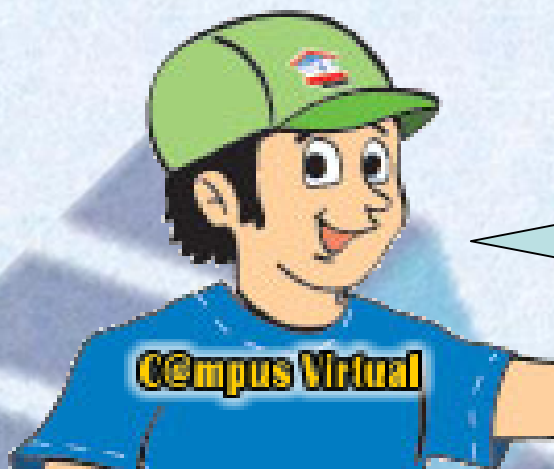
Entre la cuadra 01 y 02 de la Av. Arequipa.

Correo: enei@inei.gob.pe / campusvirtual@inei.gob.pe

Teléfonos: 433-3127 - 332-4650

Centro Andino de Formación y Capacitación en Estadística

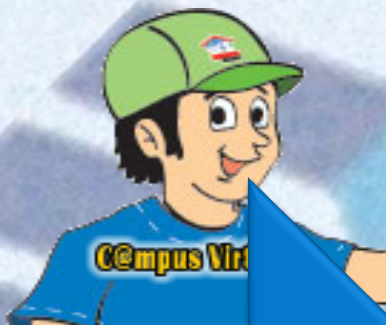
Cursos Especializados en Estadística e Informática



**Estimado alumno,
buen día. Cualquier
consulta no dudes en
comentarme o
avisarme.**

**Este curso es
netamente práctico y
se que lograremos
objetivos importantes.**

**Recuerda siempre nuestro correo
para cualquier consulta**



campusvirtual@inei.gob.pe

CURSO BASICO DE “R”



¿Por qué comprar un programa estadístico si existe R?

La estadística es una ciencia aliada a la investigación científica. Los científicos que trabajan en ecología, recursos naturales e ingeniería comúnmente emplean programas de computación para realizar análisis estadísticos. En este trabajo se revisan brevemente dos de los programas estadísticos más usados en estudios ecológicos, SPSS y SAS, y se comparan con el software estadístico R. Sobre la base de este análisis, se propone el uso de R en ciencias ecológicas e ingeniería en Latinoamérica y en países del tercer mundo en general, porque ofrece el uso gratuito de un software de primer nivel, así como también un mayor control de los análisis conducidos, extensa documentación, y un ambiente de programación desarrollado para aplicaciones estadísticas y con capacidad para ser empleado en otras áreas cuantitativas de diversas disciplinas.

Christian Salas

School of Forestry and Environmental Studies, Yale University, USA.

Departamento de Ciencias Forestales, Universidad de La Frontera, Temuco, Chile.

Ecología Austral 18:223-231. Agosto 2008

Asociación Argentina de Ecología

The R User Conference 2014

June 30 - July 3 2014 UCLA, Los Angeles, California

[John Chambers](#) Interfaces , Eficiencia y Big Data

El uso de R continúa creciendo , en particular en el número y la diversidad de paquetes que se aplican R a una amplia gama de fuentes de datos y técnicas analíticas. Al mismo tiempo , la estadística es actualmente "caliente " , sobre todo por implicación en la "ciencia de datos" y "big data". En conjunto , estos fenómenos han estimulado el interés en mejorar el uso de R para aplicaciones con grandes exigencias para la computación y / o tamaño de los datos . En respuesta sensata es necesario comprender el modelo esencial en I para el cálculo y datos; de hecho, los conceptos claves ir todo el camino de vuelta a principios del S. Los retos han crecido enormemente , pero también lo han hecho las opciones y las herramientas posibles . Esta charla se analizan diversos enfoques , utilizando como ejemplos algunos proyectos actuales prometedoras .

The R User Conference 2014

June 30 - July 3 2014 UCLA, Los Angeles, California

[Martin Mächler](#) - Buenas Prácticas en la R Programación

En la primera reunión de usuario de R celebrada en Viena en 2004 , había presentado siete directrices de buenas prácticas de programación R, los llamé "reglas" . En esta nueva visita , voy a preguntar cuánto ha cambiado - o no. Los espacios de nombres han traído aún más justificación para enfatizar funciones como los ingredientes principales de gran parte de un buen código R . Tenemos más y mejores herramientas para la investigación reproducible y análisis de datos hoy en día, y voy a tocar en algunas de las consecuencias que veo para la organización del código de usuario. Como parte de ella ha sido mi especialidad dentro de R Core , también veremos en algunos aspectos de un FAQ generalizado 7.31 y lo que todo programador debe saber acerca de la aritmética computacional .

INTRODUCCION AL SISTEMA R

R es un software de libre uso y distribución bajo Licencia Pública General de GNU, para programar análisis estadístico y gráfico. R fue creado en 1993 por Robert Gentleman y Ross Ihaka del Departamento de Estadística de la Universidad de Auckland-Nueva Zelanda y desde 1997 se desarrolla con aportes de diversas partes del mundo, bajo la coordinación del equipo principal de desarrollo de R (R Core Team Development) (R Project).

INTRODUCCION AL SISTEMA R

R funciona con paquetes de programación, los cuales están disponibles en una Red Comprehensiva de Archivos R (Comprehensive R Archive Network, CRAN) en sitios web llamados MIRROR- sitios que contienen réplicas exactas de R- desde los cuales los usuarios finales pueden descargarlos. Actualmente están disponibles 92 CRAN-MIRROR, en 45 países de los cinco continentes, 14 de las cuales se encuentran en instituciones - principalmente universidades - de siete países de América Latina: Argentina, Brasil, Chile, Colombia, Ecuador, México y Venezuela.

INTRODUCCION AL SISTEMA R

El paquete de instalación de R, nos permite realizar análisis estadísticos y gráficos básicos; para realizar otros más complejos es necesario instalar paquetes adicionales. Esencialmente R funciona como un lenguaje de programación, es decir, para realizar una acción, hay que escribir una secuencia de instrucciones que luego serán ejecutadas, sin embargo, en una sesión de R, podemos instalar y cargar una Interfaz Gráfica de Usuario (GUI), creada por John Fox: el paquete R Commander, con el cual es posible programar usando ventanas.

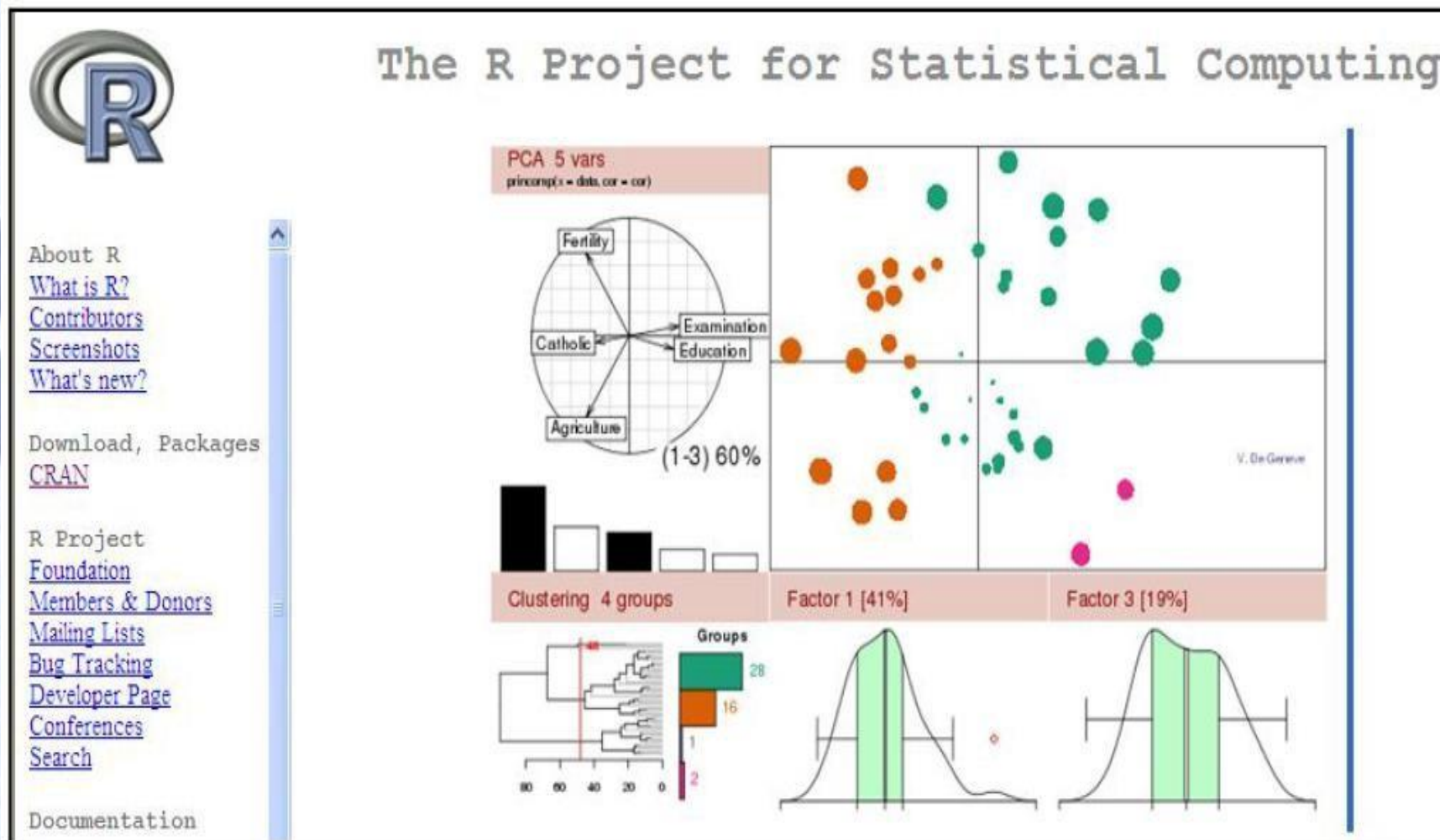
I. INSTALAR R

Descargar R

- Ingresamos a la página Web del proyecto R en la siguiente dirección:

<http://www.r-project.org>

The R Project for Statistical Computing



I. INSTALAR R

Seleccionamos una CRAN Mirror



I. INSTALAR R

Escogemos el sistema operativo con el que vamos a trabajar.

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, Windows and Mac users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#) ←
- [Download R for Windows](#)

Cuando instalamos R por primera vez, seleccionamos el subdirectorio “base”

R for Windows

Subdirectories:

[base](#) Binaries for base distribution (managed by Duncan Murdoch). This is what you want to [install R for the first time](#).

I. INSTALAR R

Descargamos el programa

Para ello: hacemos clic sobre “Download R.3.0.2 for Windows (32/64 bit)”

R-3.0.2 for Windows (32/64 bit)

[Download R 3.0.2 for Windows](#) (52 megabytes, 32/64 bit)

I. INSTALAR R

Luego seleccionamos la opción “Guardar” este archivo. Después de esta acción se creará un ícono de archivo compilado de R.



I. INSTALAR R

Para ello hacemos doble clic sobre el ícono del archivo compilado de R



Al abrirse la ventana “Abrir archivo-Advertencia de seguridad”, hacemos clic sobre el botón “Ejecutar”



I. INSTALAR R

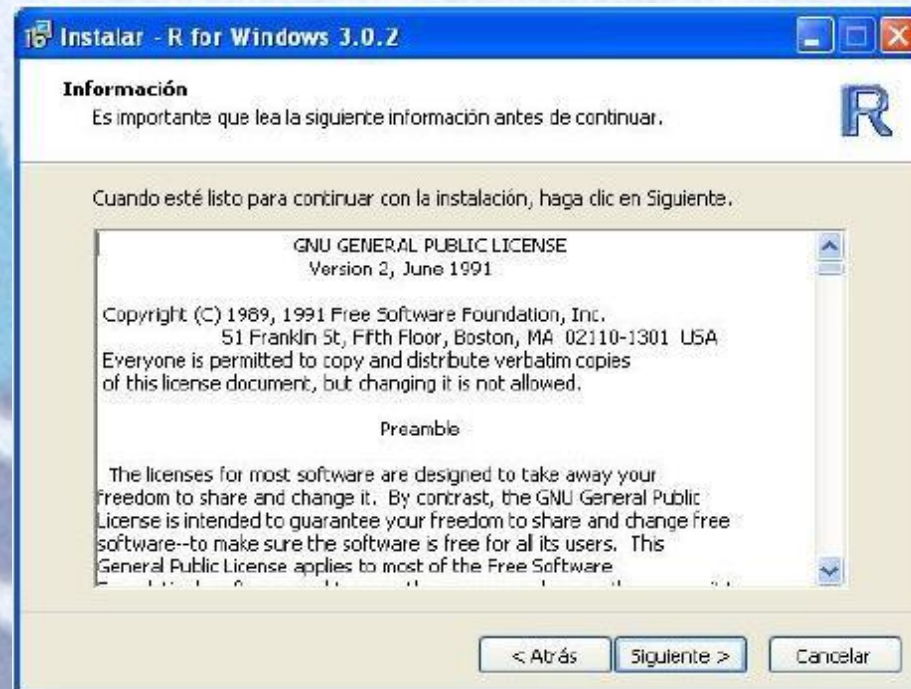
Seleccionamos el idioma de instalación

Seguimos las instrucciones del
Asistente de Instalación de R



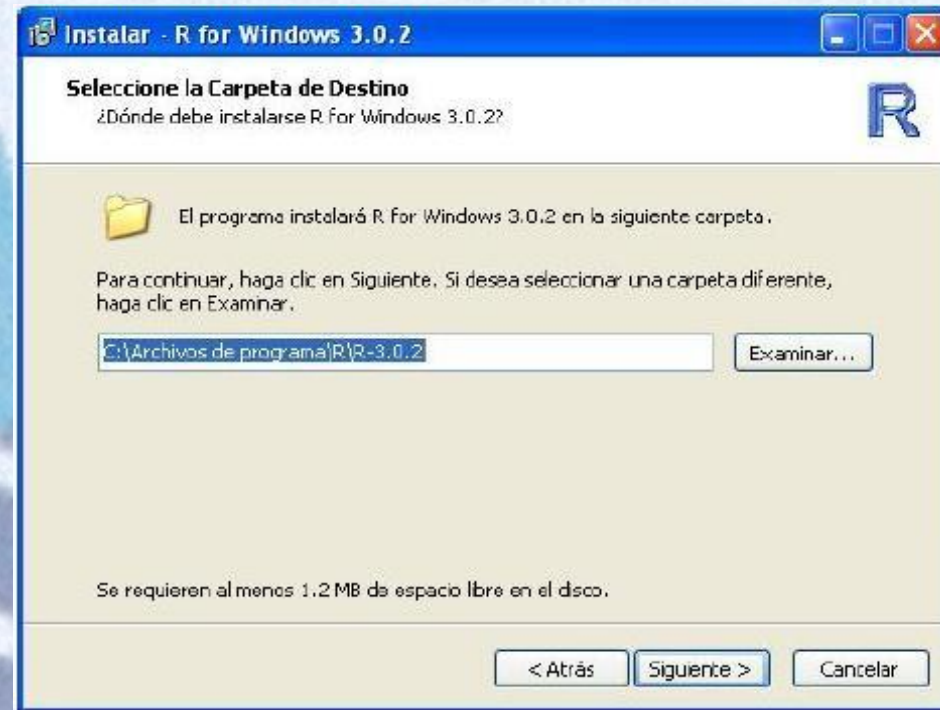
I. INSTALAR R

Leemos las condiciones de licencia de R



I. INSTALAR R

Seleccionamos la carpeta donde instalaremos R



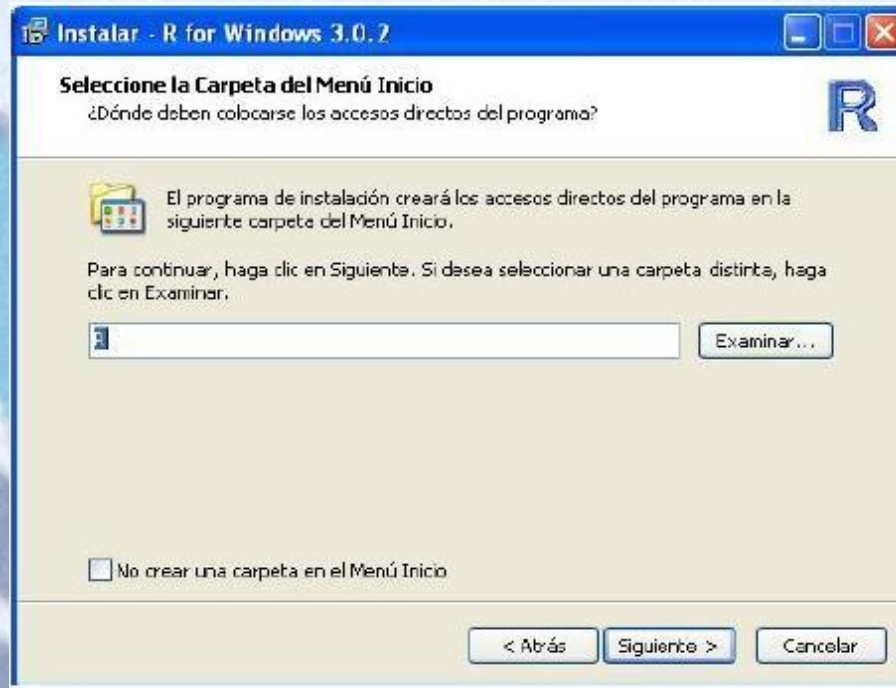
I. INSTALAR R

Seleccionamos los componentes a instalar, modifique de acuerdo a su PC



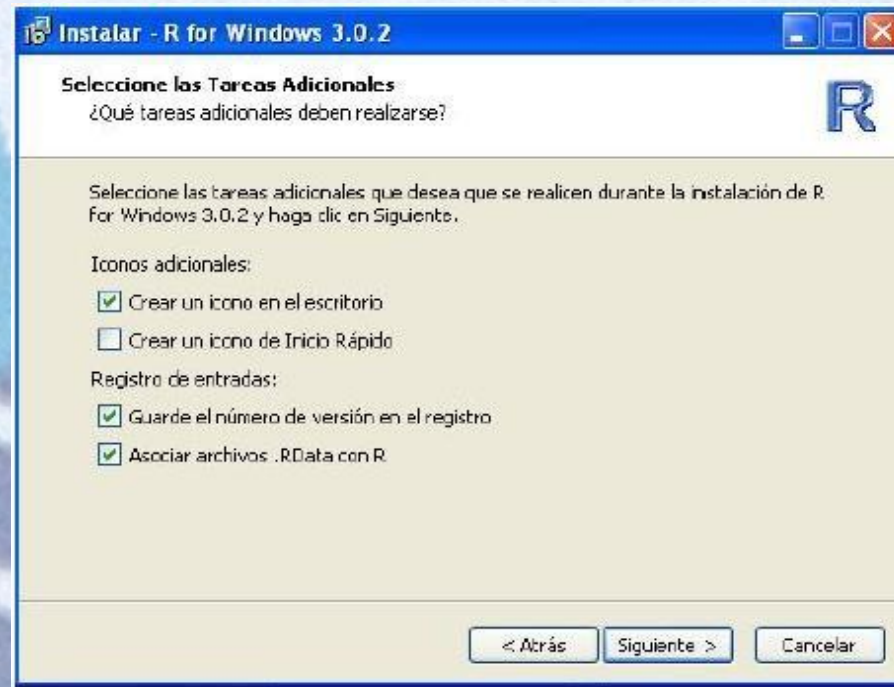
I. INSTALAR R

Seleccionamos dónde se crearán accesos directos al programa



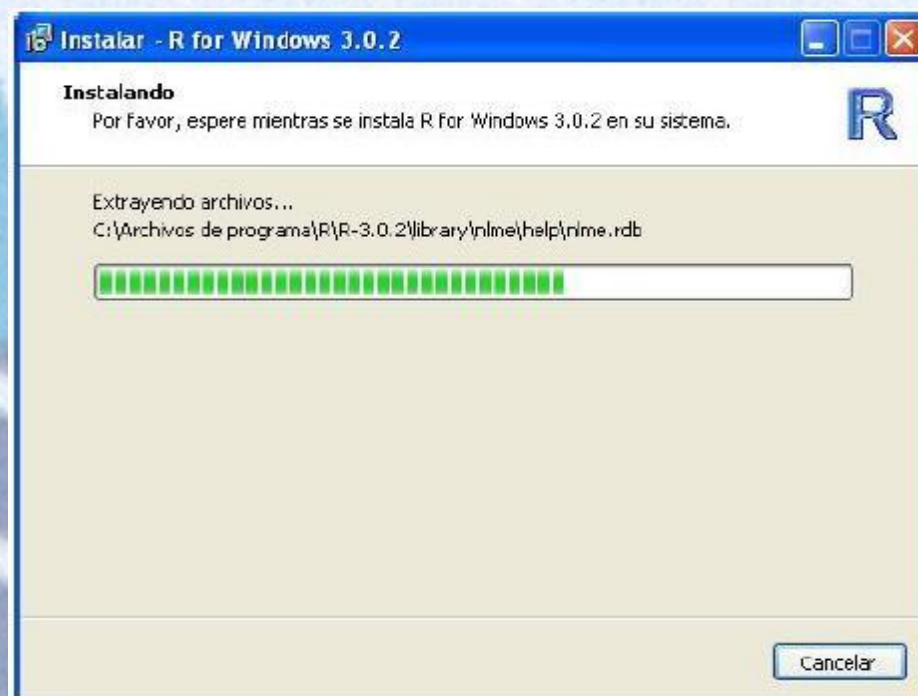
I. INSTALAR R

Seleccionamos tareas adicionales como la de “Crear un ícono en el escritorio”



I. INSTALAR R

Una vez ejecutadas las acciones anteriores, R se instalará automáticamente



I. INSTALAR R

Para terminar el proceso hacemos clic sobre el botón “Finalizar”.



II. AMBIENTE DE TRABAJO DE R

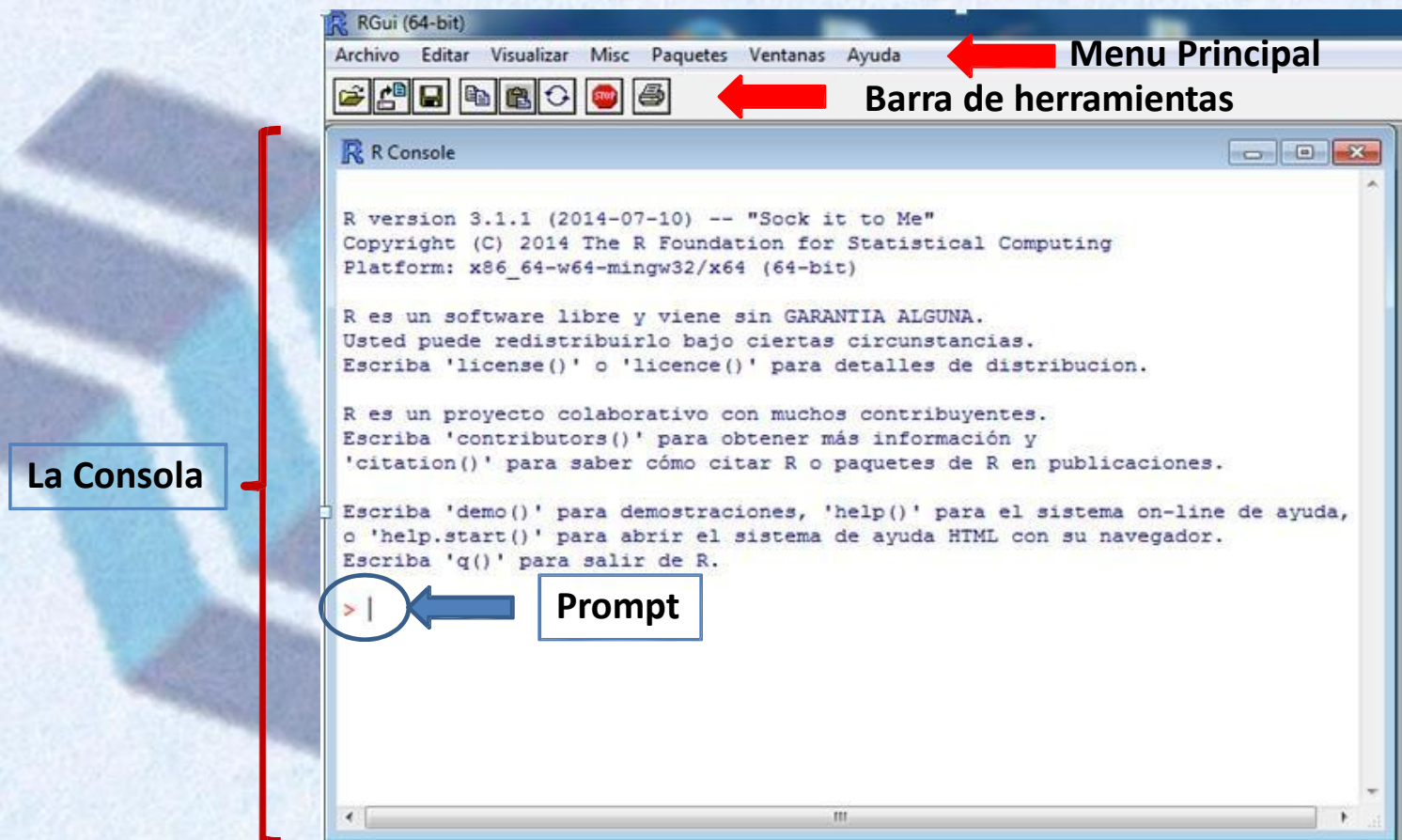
Iniciar una sesión de trabajo en R.

Hacemos doble clic sobre el ícono de R que aparece en el escritorio



II. AMBIENTE DE TRABAJO DE R

La primera pantalla



II. AMBIENTE DE TRABAJO DE R

EL MENÚ Y LA BARRA DE HERRAMIENTAS EN R



- El menú principal

Compuesto por los menús: Archivo, Editar, Visualizar, Misc, Paquetes, Ventanas, y Ayuda. Al desplegar estos menús, podemos realizar procedimientos complementarios a la escritura de programas en R. Las funciones específicas a las que podemos acceder a través de cada menú, las daremos a conocer a lo largo del curso.

- La barra de herramientas

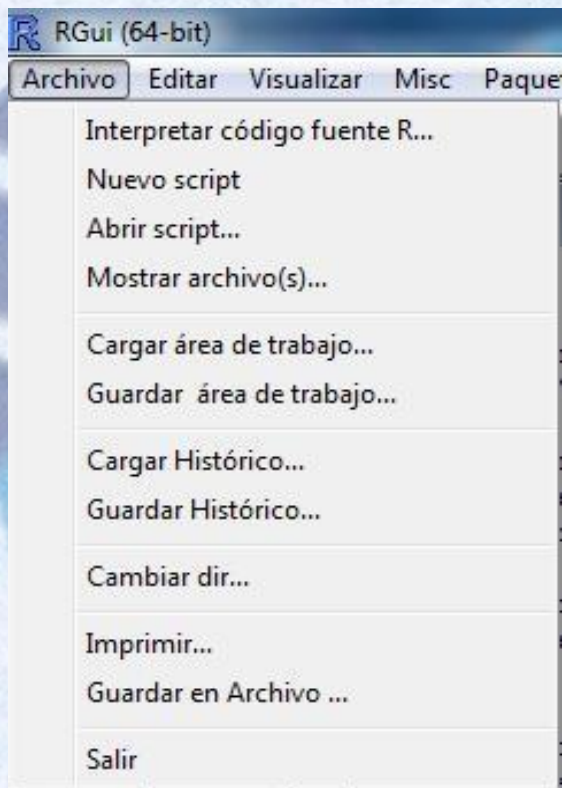
Constituyen accesos abreviados o rápidos a las funciones más usadas de R, como: abrir archivos de programas (documentos con extensión *.txt, *.R); cargar espacios de trabajo (archivos con extensión *.RData); copiar; pegar; copiar y pegar consecutivamente en la Consola, interrumpir la ejecución de instrucciones, e imprimir

II. AMBIENTE DE TRABAJO DE R

- La consola

La consola es el espacio en donde: en letras rojas, aparecen las instrucciones dadas a R y en letras azules, sus resultados. Las instrucciones pudieron ser escritas en la ventana Script y luego ejecutadas, apareciendo automáticamente en letras rojas en la Consola; o pudieron escribirse directamente en ella; en este último caso, si las instrucciones están completas, la ejecución se realizará al presionar la tecla ENTER, de no ser el caso, aparecerá el signo +, indicando que nos falta terminar de escribirlas.

II. AMBIENTE DE TRABAJO DE R



- La ventana Script

Para obtener esta ventana, desplegamos el menú **Archivo** → **Nuevo Script**.

Como podemos observar, al activar la ventana Script, los menús Archivo y Editar muestran opciones sólo pertinentes a esta ventana.

También se reduce el número de íconos disponibles y se presentan los íconos: Ejecutar (Run line or selection), y cambiar a Consola (Return focus to Console).

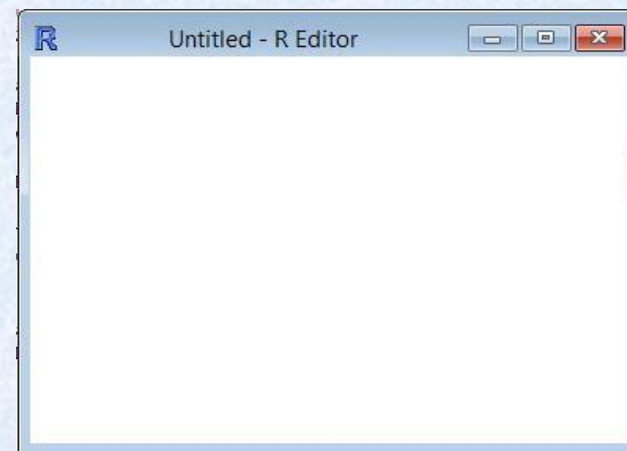
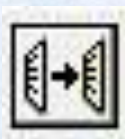
II. AMBIENTE DE TRABAJO DE R

El menu Script



La ventana Script

La ventana Script es un espacio en donde podemos escribir instrucciones. Para que R las ejecute primero debemos seleccionarlas y luego realizar cualquiera de las acciones siguientes: desplegar el menú: Editar → Run line o selection; presionar la tecla F5; presionar simultáneamente las teclas CTRL+R; o presionar el siguiente ícono.



II. AMBIENTE DE TRABAJO DE R

La ventana Script

Aunque las instrucciones para R, también pueden escribirse en la Consola, una de las ventajas de escribirlas en la ventana Script es que podemos introducir modificaciones, fácilmente y que podemos guardarlas en un archivo, para uso futuro.

Para **grabar** un Script:

Archivo → Guardar

Archivo → Guardar como...

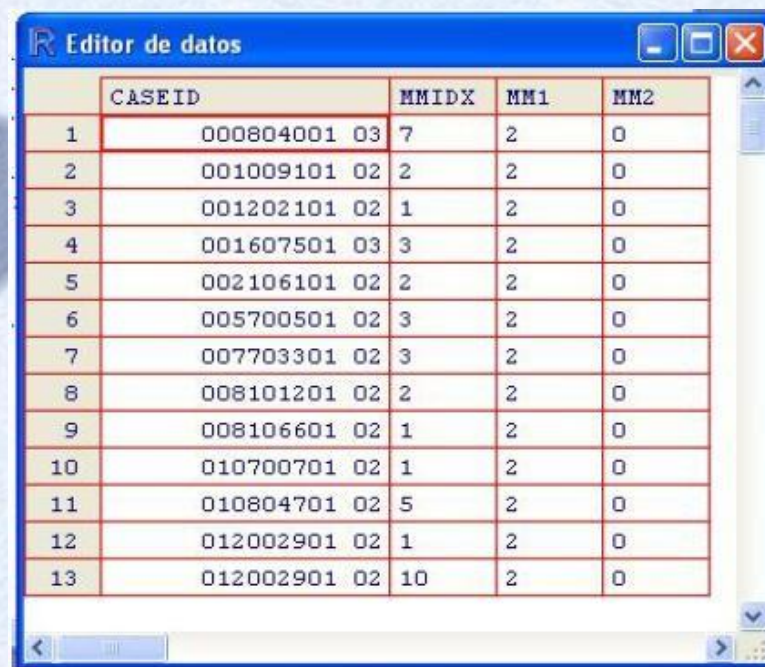
Para **abrir** una Script existente::

Archivo → Abrir Script

II. AMBIENTE DE TRABAJO DE R

El editor de datos

Para acceder a esta ventana, previamente debemos haber pedido a R que haga la lectura de un archivo de datos. Luego desplegamos el menú **Editar** → **Editor de datos**, y escribimos el nombre del conjunto de datos que deseamos editar, por ejemplo: data1

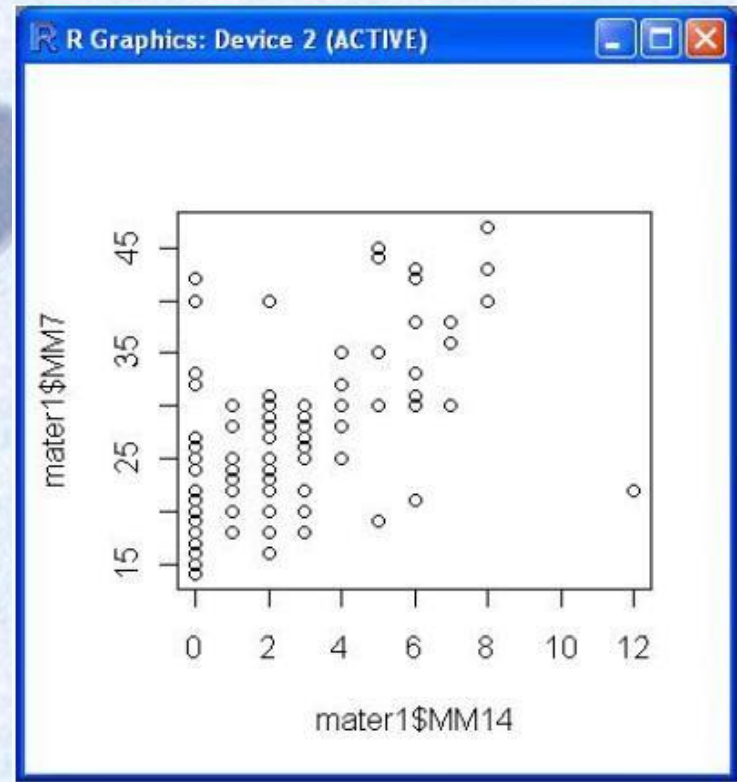


	CASEID	MMIDX	MM1	MM2
1	000804001 03	7	2	0
2	001009101 02	2	2	0
3	001202101 02	1	2	0
4	001607501 03	3	2	0
5	002106101 02	2	2	0
6	005700501 02	3	2	0
7	007703301 02	3	2	0
8	008101201 02	2	2	0
9	008106601 02	1	2	0
10	010700701 02	1	2	0
11	010804701 02	5	2	0
12	012002901 02	1	2	0
13	012002901 02	10	2	0

II. AMBIENTE DE TRABAJO DE R

La ventana de gráficos

Se activa automáticamente al dar instrucciones a R, para realizar un gráfico.



II. AMBIENTE DE TRABAJO DE R

Ubicación de la sesión de trabajo

Saber en qué directorio estamos trabajando

Usamos la función:

getwd()

para identificar el directorio, en el cual, estamos trabajando.

II. AMBIENTE DE TRABAJO DE R

Cambiar de directorio

Para cambiar de directorio de trabajo, podemos desplegar el menú

Archivo → Cambiar dir...



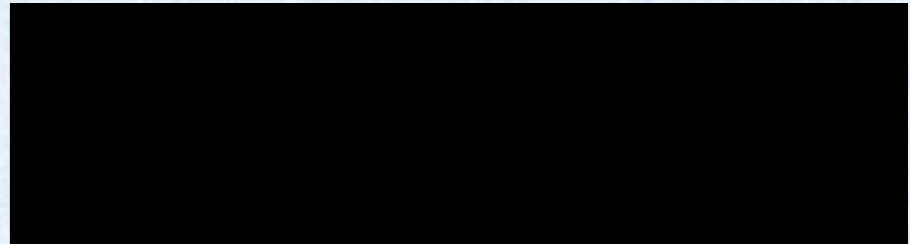
II. AMBIENTE DE TRABAJO DE R

Configurar directorio de trabajo

También podemos usar la función

setwd()

indicando la dirección completa en la que queremos que se depositen los elementos que vayamos creando durante una sesión de trabajo.



III. LOS OBJETOS EN R

Los objetos pueden ser: estructuras de datos como los vectores, los factores, las matrices, los marcos de datos, entre otros; o de funciones como las matemáticas, las estadísticas, para realizar gráficos, o inclusive funciones para realizar otras funciones.

Los objetos de R

Los objetos que más usaremos en este manual son los siguientes: vectores, factores, matrices, marcos de datos y funciones. Aquí describiremos sus principales características.

Vectores

El vector es la estructura de datos básica y puede asumir diversos *modos*, entre ellos: numéricos, carácter y lógicos.

Ejemplo: el vector numérico `x` es una secuencia de números consecutivos del 1 al 4.

```
> x  
[1] 1 2 3 4
```

III. LOS OBJETOS EN R

Factores

Un factor es un objeto que tiene como base un vector, al cual se le ha identificado sus niveles. Estos niveles describen grupos en el vector.

Ejemplo: el factor yf tiene 8 elementos y dos grupos o niveles, el grupo de hombres y el grupo de mujeres.

```
> yf
[1] hombre mujer hombre mujer mujer mujer hombre hombre
Levels: hombre mujer
```

Matriz

Es una tabla o arreglo de dos dimensiones (filas y columnas). Una matriz tiene todos sus elementos de un mismo modo, factores o vectores, pero no ambos (Paradis, 2003; Maindonald, 2008).

III. LOS OBJETOS EN R

Matriz

Ejemplo: la matriz m1 tiene 20 elementos, 5 filas y cuatro 4 columnas

```
> m1
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	6	11	16
[2,]	2	7	12	17
[3,]	3	8	13	18
[4,]	4	9	14	19
[5,]	5	10	15	20

III. LOS OBJETOS EN R

Marco de datos (data frame)

Es una estructura de datos compleja, de vectores y factores de la misma longitud.

Ejemplo: el marco de datos grupo 1, está compuesto por los factores resi y sexo y los vectores edad y nota.

```
> grupol
  resi sexo edad nota
1 urbana  h   15   15
2 rural   m   16   15
3 rural   h   17   14
4 rural   m   17   13
5 urbana  h   18   17
6 urbana  m   19   18
7 urbana  h   18   10
8 urbana  h   16   17
9 rural   m   15   12
```


III. LOS OBJETOS EN R

Funciones

Son objetos que nos permiten realizar diversas tareas con otros objetos. Estas tareas pueden ser tratamiento de archivos, tratamientos de variables, operaciones matemáticas simples y complejas; análisis estadísticos; y gráficos.

Ejemplo 1: calcular la media de la variable edad del marco de datos grupo1

```
> mean(grupo1$edad)
[1] 16.77778
```

Donde:

mean	es la función que calcula la media de la variable cuantitativa edad
grupo1	es el marco de datos al que pertenece la variable edad
\$	es la notación usada para vincular una variable a su marco de datos correspondiente

III. LOS OBJETOS EN R

Ejemplo 2: usar la función **attach()** para vincular las variables con su marco de datos, luego calcular la media de las variables edad y nota, y presentar la distribución de frecuencias de las variables **resi** y **sexo** que se encuentran en el marco de datos **grupo1**.

```
> attach(grupo1)
> mean(edad)
[1] 16.77778
> mean(nota)
[1] 14.55556
> table(resi)
resi
rural urbana
      4      5
> table(sexo)
sexo
h m
5 4
```

Donde:

attach es la función para vincular variables con su marco de datos.

mean es la función usada para estimar la media de los vectores edad y nota.

table es la función usada para elaborar las tablas de distribución de frecuencia de los factores resi y sexo

III. LOS OBJETOS EN R

Como podemos observar, al usar la función **attach()**, evitamos escribir el nombre del marco de datos seguido de la notación **\$**, cada vez que invocamos el nombre de la variable sobre la cual deseamos se aplique una función dada. Para desvincular las variables de su marco de datos usamos la función **detach()**.

En esta situación, para ejecutar una función sobre las variables necesitaremos escribirlas precedidas del nombre del marco de datos y la notación **\$**, como en el siguiente ejemplo:

```
> detach(grupo1)
> mean(edad)
Error in mean(edad) : object 'edad' not found
> mean(grupo1$edad)
[1] 16.77778
```

III. LOS OBJETOS EN R

Creación de objetos

En R podemos crear un objeto usando el operador **asignar** (`<-` o `->`). Al crearse el objeto, R lo guarda en memoria, y solo podremos visualizarlo cuando lo “invoquemos”, tal como observaremos a continuación.

Creación de vectores numéricos

Ejemplo1: Crear el vector `x` como una secuencia de números consecutivos entre 1 y 4.

Para ello escribiremos en la Consola, lo siguiente:

```
> x <- 1:4
> x
[1] 1 2 3 4
```

Donde:

`x` es el nombre asignado al vector que describe la secuencia 1:4

`<-` es el operador asignar

`1:4` es la instrucción que escribimos para obtener como salida la secuencia de números consecutivos del 1 al 4.

`[1]` es el contador de elementos del objeto al iniciar cada fila de resultados

`1 2 3 4` es el vector resultante

III. LOS OBJETOS EN R

Creación de vectores numéricos

Asimismo, para crear el vector x también podemos escribir:

```
> 1:4 -> x  
> x  
[1] 1 2 3 4
```

Es decir, el operador asignar (->) puede ser colocado en cualquiera de las dos direcciones, solo debemos tener cuidado de que la flecha apunte hacia el nombre del objeto.

III. LOS OBJETOS EN R

Creación de vectores numéricos

Ejemplo 2: Crear el vector **x** como una secuencia de números consecutivos entre 1 y 5.

```
> x <- 1:5  
> x  
[1] 1 2 3 4 5
```

Nota: En el Ejemplo 2 observamos que es posible asignar el mismo nombre a dos vectores diferentes (el nombre **x** había sido previamente asignado al vector 1 2 3 4). Sin embargo, **R solo tomará en cuenta la última asignación (1:5) de valores al vector x**, para posteriores procedimientos.

III. LOS OBJETOS EN R

Creación de vectores numéricos

Crear vectores usando la función `seq()`

Con la **función** `seq()` también podemos crear una secuencia de números consecutivos, pero añadiéndole el argumento **by** podemos introducir saltos en dicha secuencia e indicar a R de qué tamaño debe ser cada salto.

```
> x<-seq(0,10,by=2)
> x
[1] 0 2 4 6 8 10
```

III. LOS OBJETOS EN R

Creación de vectores numéricos

Usando el argumento **length** podemos indicar a R, cuál es el número total de saltos que debe contener el intervalo de números.

```
> y<-seq(0,10,length=6)
> y
[1] 0 2 4 6 8 10
```

```
> y<-seq(1,15,length=6)
> y
[1] 1.0 3.8 6.6 9.4 12.2 15.0
> y<-seq(0,15,length=6)
> y
[1] 0 3 6 9 12 15
> |
```


III. LOS OBJETOS EN R

Crear vectores usando la función **rep()**

La función **rep()** repite una secuencia de números.

En el ejemplo, **rep()** repite tres veces la secuencia de números consecutivos 1:2.

```
> x<-rep(1:2,3)
> x
[1] 1 2 1 2 1 2
```

Crear vectores sin un patrón particular usando la función combinar **c()**

```
> x<-c(5,7,1,0)
> x
[1] 5 7 1 0
```

III. LOS OBJETOS EN R

Creación de vectores numéricos

Datos perdidos (NA) en un vector numérico

Un vector puede contener un dato perdido, en este caso se le asignará el valor especial **NA**.

```
> z <- c(7,9,5,8,9,3,NA,4)
> z
[1] 7 9 5 8 9 3 NA 4
```


III. LOS OBJETOS EN R

Creación de vectores lógicos

“Los vectores lógicos se generan por *condiciones*” (Venables & Smith, 2011: 9).

Para crear este tipo de vectores, usamos operadores lógicos como:

< (menor);

<= (menor o igual);

> (mayor);

>= (mayor o igual);

== (igual);

!= (diferente);

& intersección (Y) de dos expresiones lógicas,

| unión (O) de dos expresiones lógicas.

III. LOS OBJETOS EN R

Creación de vectores lógicos

Ejemplo: dado el vector numérico `z`, crear el vector lógico `w`, que nos permita identificar: qué elementos de `z` son menores que 8

```
> z <- c(7,9,5,8,9,3,4)
> z
[1] 7 9 5 8 9 3 4
> w <- z < 8
> w
[1] TRUE FALSE TRUE FALSE FALSE TRUE TRUE
```

Como vemos en el ejemplo, los elementos de un vector lógico pueden ser: TRUE o FALSE, sin embargo, si el vector lógico se construye en base a un vector numérico que contiene un dato perdido, en el vector lógico se le asignará a este, el valor especial NA.

```
> z <- c(7,9,5,8,9,3,NA,4)
> z
[1] 7 9 5 8 9 3 NA 4
> w <- z < 8
> w
[1] TRUE FALSE TRUE FALSE FALSE TRUE NA TRUE
```


III. LOS OBJETOS EN R

Creación de vectores caracter

Crear vectores caracter usando `rep()`

Los elementos de un vector caracter deben escribirse usando comillas dobles (") o simples (').

Ejemplo: crear el vector caracter x que contenga el caracter "x", repetido 5 veces.

```
> x<-rep("x",5)
> x
[1] "x" "x" "x" "x" "x"
```

Crear vectores caracter usando la función combinar `c()`

```
> y<-c("costa","sierra","selva")
> y
[1] "costa" "sierra" "selva"
```

III. LOS OBJETOS EN R

Creación de vectores caracter

En el siguiente ejemplo combinamos la función `rep()` y `c()` para crear el vector carácter `z`.

```
> z<-rep(c("x","y"),4)
> z
[1] "x" "y" "x" "y" "x" "y" "x" "y"
```


III. LOS OBJETOS EN R

Creación de factores

Para crear factores, se usa la función **factor()**.

Ejemplo 1: A partir del vector $x = 1, 2, 1, 2, 2, 2, 1, 1$, crear el factor xf .

```
> x<-c(1,2,1,2,2,2,1,1)
> x
[1] 1 2 1 2 2 2 1 1
> xf<-factor(x)
> xf
[1] 1 2 1 2 2 2 1 1
Levels: 1 2
```

Un ejemplo de funciones que se pueden ejecutar con un factor es `table()`, cuyo resultado es la distribución de frecuencia de los niveles del factor

```
> table(xf)
xf
1 2
4 4
```

III. LOS OBJETOS EN R

Creación de factores

Ejemplo 2: Para una mejor ilustración veamos la creación del factor (yf) a partir del vector de tipo caracter (y), así como los resultados de la función table() aplicada a este factor.

```
> y
[1] "hombre" "mujer"  "hombre" "mujer"  "mujer"  "mujer"  "hombre" "hombre"
> yf<-factor(y)
> yf
[1] hombre mujer  hombre mujer  mujer  mujer  hombre hombre
Levels: hombre mujer
> table(yf)
yf
hombre  mujer
      4      4
```


III. LOS OBJETOS EN R

Creación de una matriz

Para crear una matriz usamos la **función matrix()**, detallamos sus componentes, e indicamos el número de **filas (nrow)** y/o **columnas (ncol)** de las que estará compuesta.

Ejemplo: crear la matriz m1 de 5 filas y 4 columnas con la secuencia de números consecutivos del 1 al 20.

```
> m1 <- matrix(1:20,nrow=5)
> m1
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	6	11	16
[2,]	2	7	12	17
[3,]	3	8	13	18
[4,]	4	9	14	19
[5,]	5	10	15	20

Además con la función `dim()`, podemos reportar el número de filas y columnas que tiene la matriz.

```
> dim(m1)
[1] 5 4
```

III. LOS OBJETOS EN R

Creación de un marco de datos (data frame)

Para crear un marco de datos, utilizamos la **función data.frame()**.

Ejemplo 1. Crear el marco de datos grupo1, con los factores **resi** y **sexo** y los vectores **edad** y **nota**.

Creación de los vectores de tipo carácter **x** y **w**, y de los vectores numéricos **y** y **z**:

```
> x<-c("urbana","rural","rural","rural","urbana","urbana","urbana","urbana","rural")
> x
[1] "urbana" "rural"  "rural"  "rural"  "urbana" "urbana" "urbana"
[8] "urbana" "rural"
> w<-c("h","m","h","m","h","m","h","h","m")
> w
[1] "h" "m" "h" "m" "h" "m" "h" "h" "m"
> y<-c(15,16,17,17,18,19,18,16,15)
> y
[1] 15 16 17 17 18 19 18 16 15
> z<-c(15,15,14,13,17,18,10,17,12)
> z
[1] 15 15 14 13 17 18 10 17 12
```


III. LOS OBJETOS EN R

Conversión de los vectores x y w en los factores xf y wf:

```
> xf<-factor(x)
> xf
[1] urbana rural  rural  rural  urbana urbana urbana urbana rural
Levels: rural urbana
> wf<-factor(w)
> wf
[1] h m h m h m h h m
Levels: h m
```

Creación del marco de datos grupo1 con los factores resi y sexo y los vectores edad y nota.

```
> grupo1<-data.frame(resi=xf, sexo=wf, edad=y, nota=z)
> grupo1
  resi sexo edad nota
1 urbana  h   15   15
2 rural   m   16   15
3 rural   h   17   14
4 rural   m   17   13
5 urbana  h   18   17
6 urbana  m   19   18
7 urbana  h   18   10
8 urbana  h   16   17
9 rural   m   15   12
```

Comunicación constante con la Escuela del INEI

Correo de la Escuela del INEI
enei@inei.gob.pe

Área de Educación Virtual
campusvirtual@inei.gob.pe

