

**Introducción al uso de R y R Commander para el  
análisis estadístico de datos en ciencias sociales**

**Rosario Collatón Chicana**

(2014)

## ÍNDICE

INTRODUCCIÓN .....	4
1. INSTALAR R.....	5
1.1. Descargar R.....	5
1.2. Instalar R.....	6
2. EL AMBIENTE DE TRABAJO EN R.....	11
2.1. Iniciar una sesión de trabajo en R.....	11
2.2. El ambiente de trabajo en R.....	11
2.3. Organizar ventanas.....	14
2.4. Ubicación de la sesión de trabajo.....	15
3. ELEMENTOS DE PROGRAMCIÓN EN LENGUAJE R.....	16
3.1. Los objetos de R.....	16
3.2. Atributos intrínsecos de los objetos.....	18
3.3. Creación de objetos.....	18
3.4. Algunas recomendaciones para escribir en lenguaje R.....	24
3.5. Solicitar ayuda.....	24
4. TRATAMIENTO Y EXPLORACIÓN DE ARCHIVOS.....	26
4.1. Preparar archivos externos que puedan ser leídos por el paquete básico de R.....	26
4.2. Leer archivos desde R.....	27
4.3. Explorar el contenido de un archivo.....	31
4.4. Segmentar archivos.....	32
4.5. Guardar y recargar marcos de datos.....	33
5. TRATAMIENTO DE VARIABLES.....	35
5.1. Convertir vectores en factores o crear variables cualitativas.....	35
5.2. Eliminar variables de un marco de datos.....	36
5.3. Renombrar variables.....	37
5.4. Crear nuevas variables a partir de otras existentes haciendo cálculos.....	37
5.5. Recodificar una variable numérica usando la función <code>recode ( )</code> o <code>Recode ( )</code> ...	38
5.6. Modificar datos desde el Editor.....	40
6. ANÁLISIS UNIVARIADO.....	42
6.1. Distribución de frecuencias.....	42

6.2. Medidas de tendencia central .....	43
6.3. Medidas de dispersión .....	43
6.4. Medidas de posición .....	44
6.5. Resumir medidas estadísticas de todas las variables de una base de datos .....	44
6.6. Resumir medidas estadísticas para una variable de la base de datos.....	45
7. ANÁLISIS BIVARIADO .....	46
7.1. Tablas cruzadas .....	46
7.2. Test de independencia con Chi cuadrado.....	50
7.3. Pruebas $t$ .....	52
7.4. Correlación bivariada.....	57
8. GRÁFICOS .....	62
8.1. Gráficos de barras .....	62
8.2. Gráficos de sectores en porcentajes.....	63
8.3. Histogramas.....	64
8.4. Diagrama de caja .....	65
8.5. Diagramas de dispersión .....	67
8.6. Gráfico de comparación de cuantiles (QQ-plot) .....	71
8.7. Pirámide de edades .....	73
9. CARACTERÍSTICAS Y FUNCIONES BÁSICA DE R COMMANDER .....	79
9.1. Instalar R Commander .....	79
9.2. Descripción del ambiente de trabajo de R Commander .....	81
9.3. Tratamiento de archivos con R Commander.....	83
9.4. Tratamiento de variables con R Commander.....	88
9.5. Análisis univariado con R Commander .....	93
9.6. Análisis bivariado con R Commander .....	97
9.7. Gráficos con R Commander.....	110
BIBLIOGRAFÍA .....	124
ANEXOS .....	125

## INTRODUCCIÓN

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).

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.

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.

Este manual fue escrito como resultado de la adquisición de competencias en mi formación doctoral, en el Centre de recherche en démographie et sociétés de la Université catholique de Louvain, la cual fue posible gracias a una beca del Programa ALBAN y tiene como objetivo, introducir a estudiantes e investigadores de las ciencias sociales, especialmente a los no especialistas en programación, en el tratamiento y el análisis estadístico de datos usando R y R Commander.

Los datos procesados como ejemplos en este manual provienen de tres fuentes: la Encuesta Demográfica y de Salud Familiar Perú, Endes-2012; Eurostat, y el Censo Nacional de Población y Vivienda de Perú de 2007, los cuales son de libre acceso.

El manual está dividido en nueve capítulos. En el primero se describe cómo instalar R; en el segundo se describe el ambiente de trabajo en R; en el tercero se da a conocer algunos elementos básicos de programación en lenguaje R; en el cuarto se aborda el tratamiento de archivos y algunos procedimientos para la exploración de los mismos; en el quinto capítulo se detallan los procedimientos para el tratamiento de variables; en el sexto, se da a conocer procedimientos para el análisis estadístico descriptivo univariado; en el séptimo capítulo, se aborda los procedimientos de análisis bivariado; en el octavo capítulo se describen los procedimientos para la construcción de gráficos y en el último capítulo se describen las características y funciones básicas de R Commander.

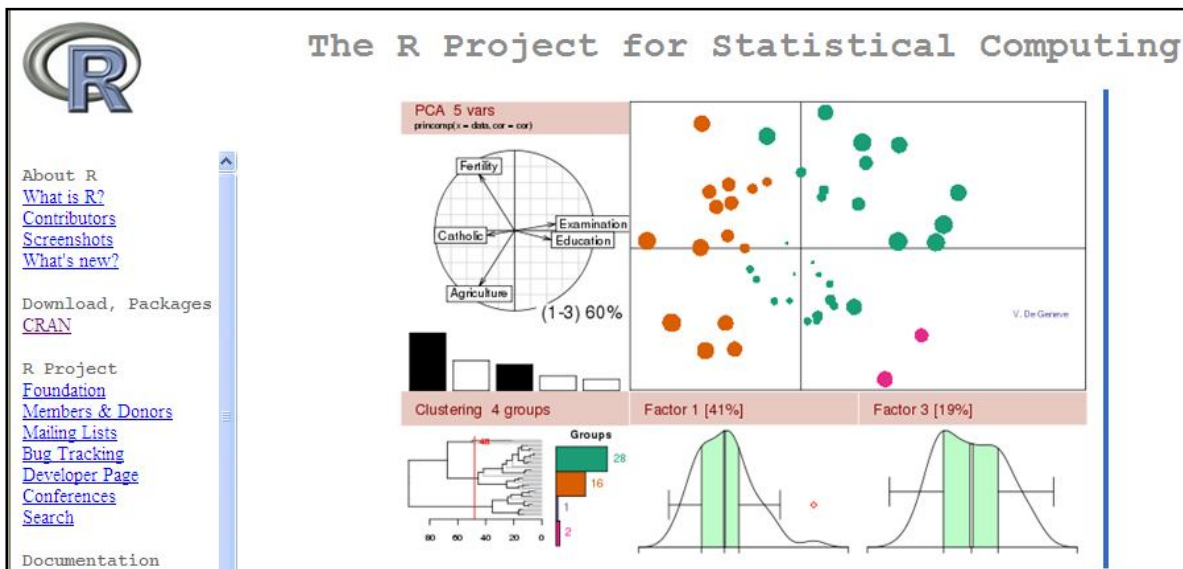
# 1. INSTALAR R

Para instalar R seguimos los siguientes pasos:

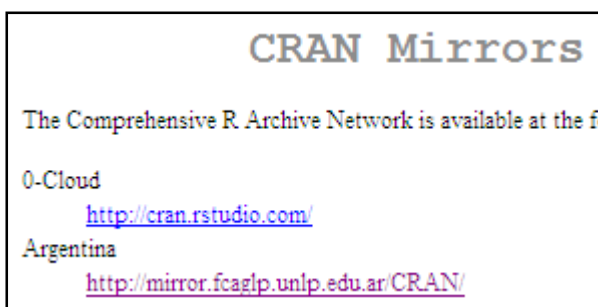
## 1.1. Descargar R

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

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



- Seleccionamos una CRAN Mirror



- Escogemos el sistema operativo con el que vamos a trabajar.

Por ejemplo: Download R for Windows.

**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](#).

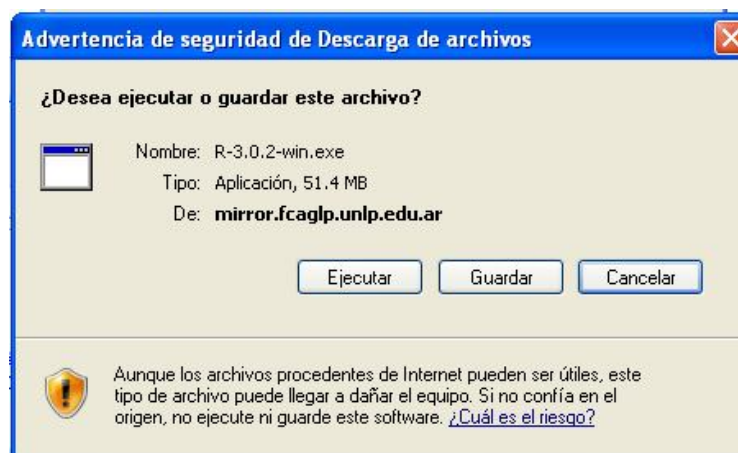
- Descargamos el programa

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)

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



## 1.2. Instalar R

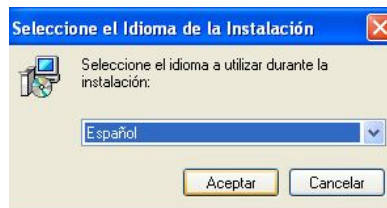
Para ello hacemos doble clic sobre el ícono del archivo compilado de R



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



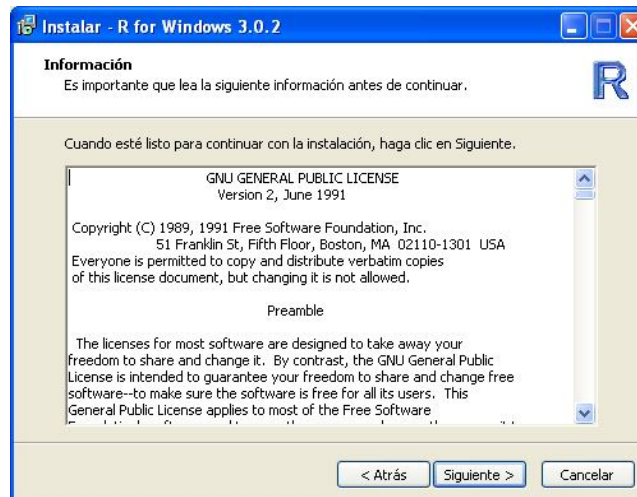
- *Seleccionamos el idioma de instalación*



- *Seguimos las instrucciones del Asistente de Instalación de R*



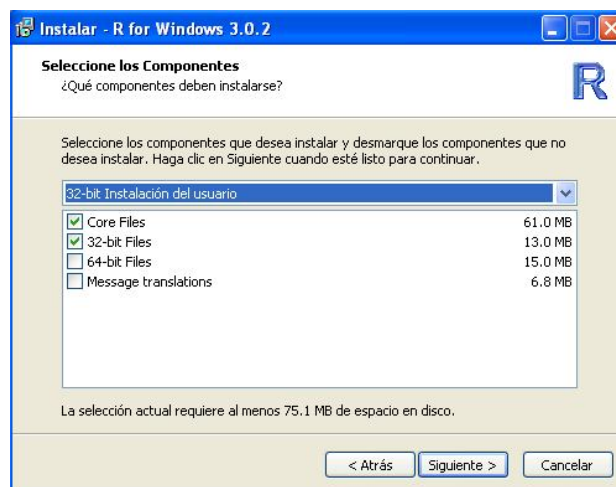
Leemos las condiciones de licencia de R



- *Seleccionamos la carpeta donde instalaremos R*



- *Seleccionamos los componentes a instalar*

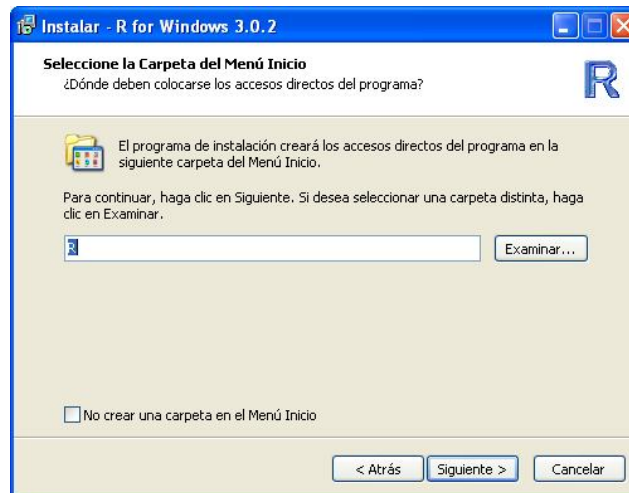




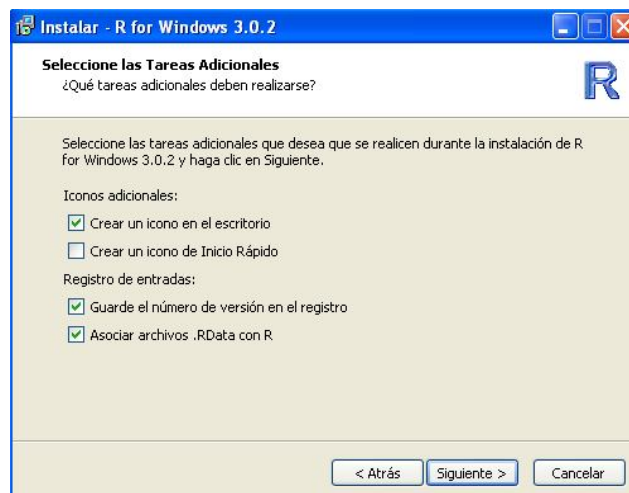
- *Especificamos si utilizaremos opciones de configuración*



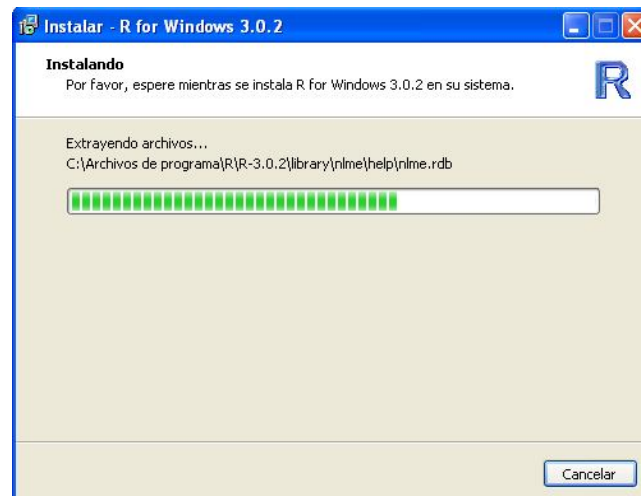
- *Seleccionamos dónde se crearán accesos directos al programa*



- *Seleccionamos tareas adicionales como la de "Crear un ícono en el escritorio"*



Una vez ejecutadas las acciones anteriores, R se instalará automáticamente.



- *Para terminar el proceso hacemos clic sobre el botón "Finalizar".*



## 2. EL AMBIENTE DE TRABAJO EN R

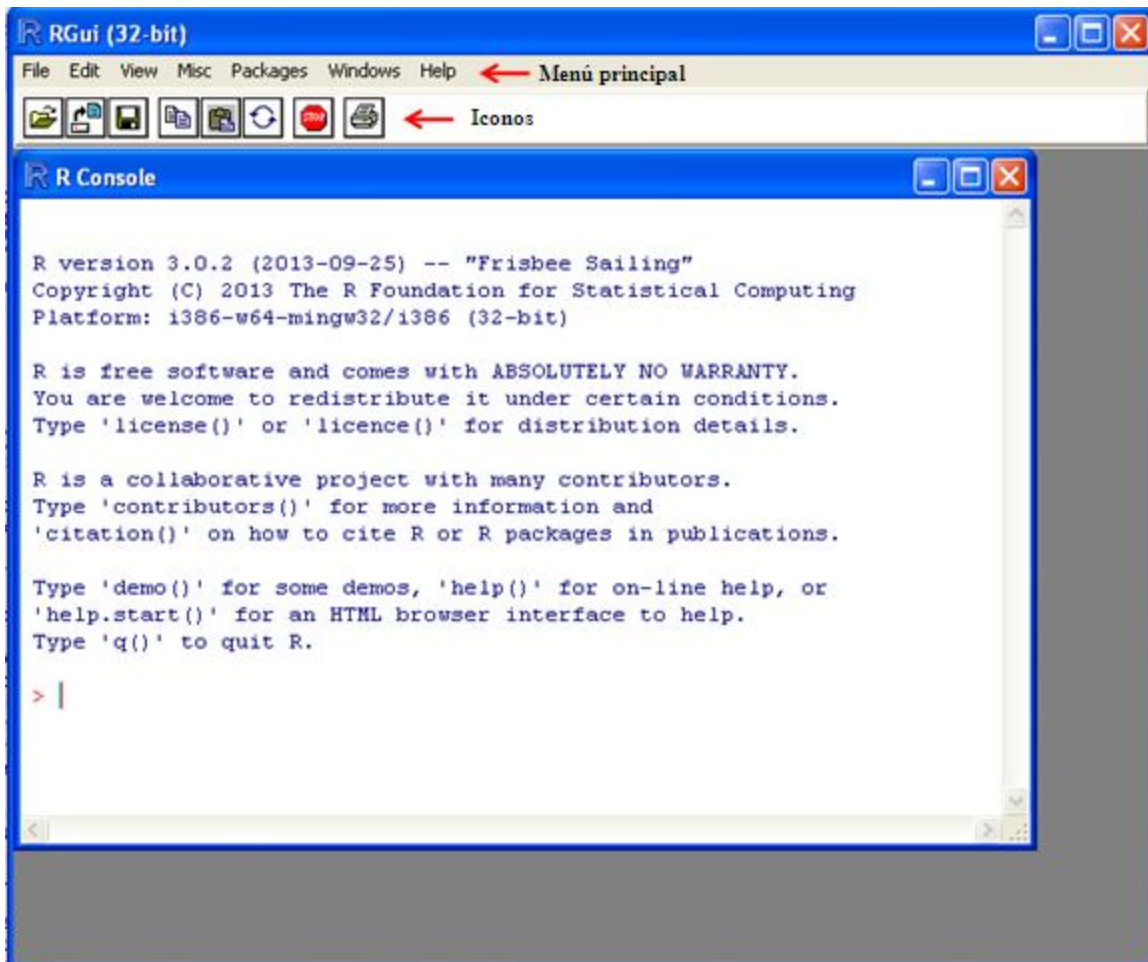
### 2.1. Iniciar una sesión de trabajo en R.

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



### 2.2. El ambiente de trabajo en R

Al abrir R se mostrará la siguiente imagen:



En la imagen podemos identificar los siguientes elementos:

- *El menú principal*

Compuesto por los menús: File, Edit, View, Misc, Package, Windows, y Help. 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 manual.

- *Los íconos de funciones*

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.

- *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.

Otros elementos de R, son visibles al realizar procedimientos previos:

- *La ventana Script*

Para obtener esta ventana, desplegamos el menú File → New Script.

Como podemos observar, al activar la ventana Script, los menús File y Edit 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).



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



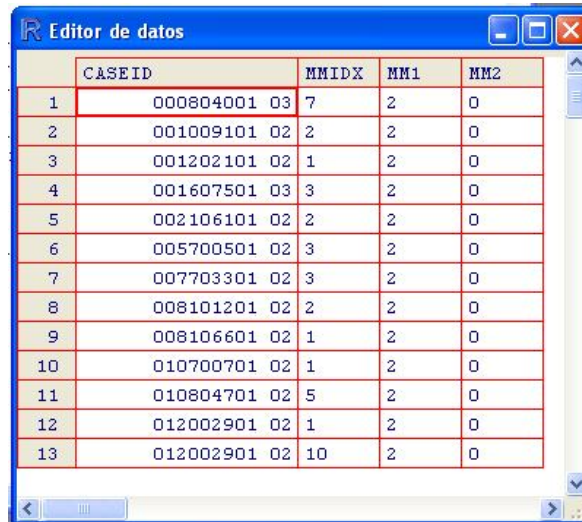
Aunque las instrucciones para R, también pueden escribirse en la Consola, una de las ventajas de escribirlas en la ventana Script es que podemos introducir modificaciones, fácilmente y que podemos guardarlas en un archivo, para uso futuro.

Para grabar un Script desplegamos el menú File → Save o File → Save as...

Para abrir una Script existente desplegamos el menú File→ Open Script...

- *El editor de datos*

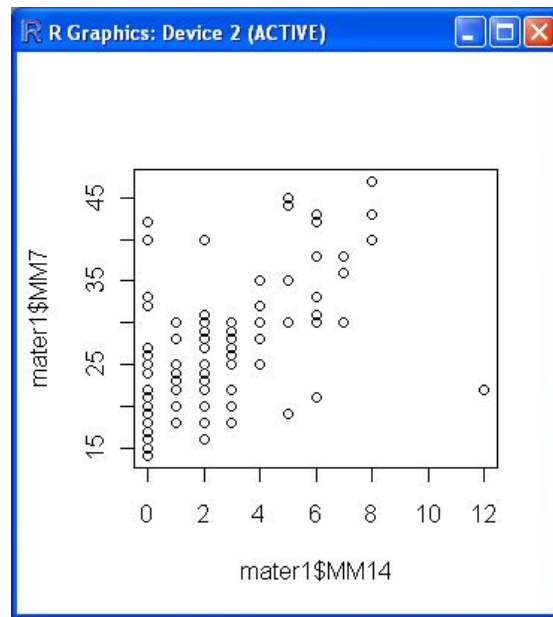
Para acceder a esta ventana, previamente debemos haber pedido a R que haga la lectura de un archivo de datos. Luego desplegamos el menú Edit→ Data editor, y escribimos el nombre del conjunto de datos que deseamos editar, por ejemplo: mater1.



	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

- *La ventana de gráficos*

Se activa automáticamente al dar instrucciones a R, para realizar un gráfico. (Ver detalles en el capítulo 8).

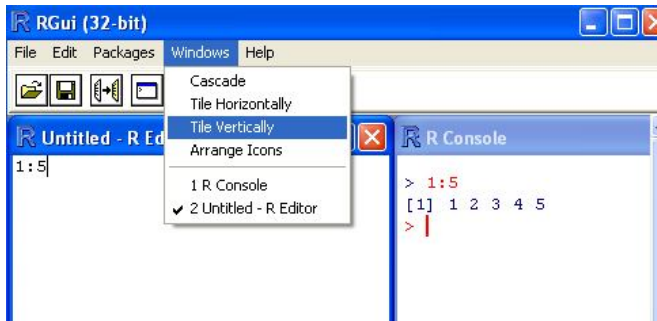


### 2.3. Organizar ventanas

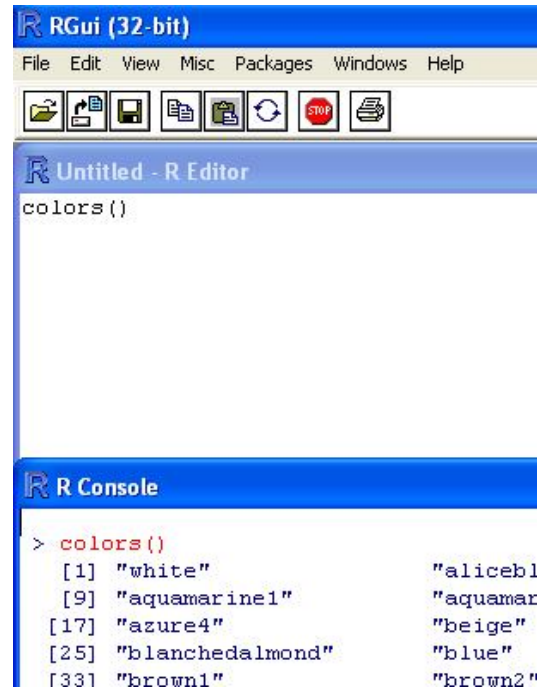
Para trabajar con mayor comodidad, se puede organizar la presentación de las ventanas de Script, Consola y/o gráficos, en forma paralela, ya sea de manera vertical u horizontal. Para ello desplegamos el menú:

Windows → Tile Vertically o Windows → Tile Horizontally

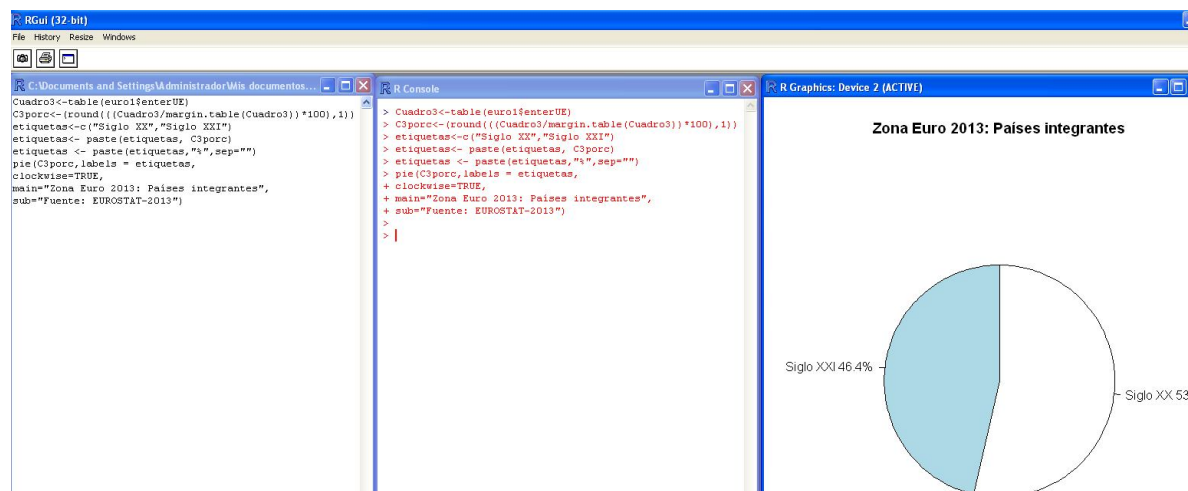
Ventanas organizadas verticalmente



Ventanas organizadas horizontalmente



R organizará todas las ventanas que, se encuentren abiertas. En el siguiente gráfico se observa cómo se organizan tres ventanas abiertas en una sesión, desplegando el menú Windows → Tile Vertically



## 2.4. 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.

```
> getwd()
[1] "C:/Documents and Settings/Administrador/Mis documentos"
```

- *Cambiar de directorio*

Para cambiar de directorio de trabajo, podemos desplegar el menú File → Change dir...



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.

```
> setwd("D:/")
> getwd()
[1] "D:/"
```

### 3. ELEMENTOS DE PROGRAMACIÓN EN LENGUAJE R

“R trabaja con objetos” (Paradis, 2003: 9). Estos objetos pueden ser: estructuras de datos como los vectores, los factores, las matrices, los marcos de datos, entre otros; o de funciones como las funciones matemáticas, las funciones estadísticas, las funciones para realizar gráficos, o inclusive funciones para realizar otras funciones.

#### 3.1. 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.

##### 3.1.1. 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
```

##### 3.1.2. 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
```

##### 3.1.3. 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).

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
```

### 3.1.4. 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`.

```
> 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
```

### 3.1.5. 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

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`

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

```

### 3.2. Atributos intrínsecos de los objetos

Entre los atributos intrínsecos de los objetos tenemos: el **modo** (numérico, caracter, lógico...) y la **longitud** (número de elementos que contiene un objeto). Es importante tomar en consideración estos atributos de los objetos, ya que depende de ellos la aplicabilidad de una función. (Venables & Smith, 2011).

### 3.3. 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.

### 3.3.1. Creación de vectores

#### 3.3.1.1. Creación de vectores numéricos

- *Crear vectores con el operador dos puntos*

El operador dos puntos ( : ) indica una secuencia de números consecutivos.

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

Para ello escribiremos en la Consola<sup>1</sup>, 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

Nótese que al crear el objeto `x`, este no apareció automáticamente en pantalla. Como dijimos en el punto anterior, este fue guardado en memoria, y sólo apareció cuando lo invocamos (en este caso, al digitar `x` en la Consola y presionar el botón “Enter” del teclado).

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.

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
```

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,

---

<sup>1</sup> Recordemos que podemos escribir las instrucciones directamente en la Consola o a partir de la ventana Script.

R solo tomará en cuenta la última asignación (1:5) de valores al vector `x`, para posteriores procedimientos.

- 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
```

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
```

- 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
```

- Datos perdidos (NA) en un vector numérico<sup>2</sup>

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
```

### 3.3.1.2. 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.

<sup>2</sup> También puede introducirse un dato perdido a un vector carácter.

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
```

### 3.3.1.3. 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"
```

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"
```

### 3.3.2. 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

```

Ejemplo 2: Para una mejor ilustración veamos la creación del factor (`yf`) a partir del vector de tipo carácter (`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

```

### 3.3.3. 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

```

### 3.3.4. 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" "urbana" "rural"
[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
```

- 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
```

Una vez creado el marco de datos, podemos llamar variables, a los vectores y factores incluidos en él.

Bases de datos externas, pueden ser leídas por R como marcos de datos usando la función `read.table` (ver procedimientos en el capítulo 4).

### 3.4. Algunas recomendaciones para escribir en lenguaje R

- Al escribir instrucciones podemos hacer comentarios, utilizando el símbolo # al comienzo de la oración.
- El nombre de un objeto debe empezar con una letra y puede incluir letras y números.
- R discrimina entre mayúsculas y minúsculas.
- Las funciones se escriben seguidas de paréntesis.
- Los corchetes se usan para referir posiciones.
- Para programar en R, necesitamos conocer el uso de sus funciones, el tipo objeto al que puede aplicárseles, sus argumentos, así como los valores que requieren ser definidos. Podemos acceder a toda esta información a través de diversos procedimientos de solicitud de ayuda en R (ver 3.5).

### 3.5. Solicitar ayuda

Hay diversas maneras de solicitar ayuda para escribir en lenguaje R, aquí describiremos tres:

- *Si conocemos el nombre de la función*

Escribimos en la consola el nombre de esta, precedido por el signo de cierre de interrogación (?). Como resultado R nos mostrará una descripción de la función, la forma de su uso, los argumentos, detalles, valores, funciones asociadas y ejemplos.

```
> ?colors
starting httpd help server ... done
```

```
colors {grDevices} R Documentation

                Color Names

Description
Returns the built-in color names which R knows about.

Usage
colors (distinct = FALSE)
colours(distinct = FALSE)

Arguments
distinct logical indicating if the colors returned should all be distinct; e.g., "snow" and "snow1"
are effectively the same point in the (0:255)^3 RGB space.


Details
These color names can be used with a col= specification in graphics functions.
```



- *Si no conocemos el nombre de la función*

Escribimos doble signo de interrogación antes de la palabra (en inglés) asociada a la función. Luego de ello R nos mostrará una página con diversos recursos asociados a la palabra sobre la cual buscamos información.

```
> ??colour
```

Search Results 

---

⤴

The search string was "colour"

**Vignettes:**

[colorspace:hcl-colors](#) HCL-Based Color Palettes in R [PDF](#) [source](#) [R code](#)

**Code demonstrations:**

[grDevices:colors](#) A show of R's predefined colors() [\(Run demo in console\)](#)  
[grDevices:hclColors](#) Exploration of hcl() space [\(Run demo in console\)](#)

**Help pages:**

[colorspace:HLS](#) Create HLS Colors  
[colorspace:HSV](#) Create HSV Colors  
[colorspace:LAB](#) Create LAB Colors  
[colorspace:LUV](#) Create LUV Colors  
[colorspace:RGB](#) Create RGB Colors

- *Solicitar ejemplos de la manera en que se emplea una función*

```
> example(mean)
```

```
mean> x <- c(0:10, 50)
```

```
mean> xm <- mean(x)
```

```
mean> c(xm, mean(x, trim = 0.10))  
[1] 8.75 5.50
```

## 4. TRATAMIENTO Y EXPLORACIÓN DE ARCHIVOS

Como señaláramos en el punto 3.3.4 del capítulo anterior, R puede leer archivos externos como marcos de datos (data frame). En esta parte leeremos archivos con extensión `*.txt`, `*.csv`, y `*.dat`, usando el paquete básico instalado en R.

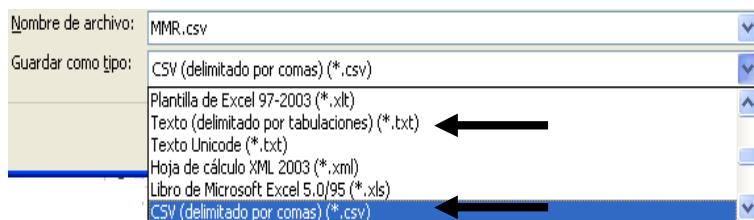
Dado que la mayoría de veces las bases de datos se encuentra archivados en formato SPSS o Excel, lo primero que haremos será preparar estos archivos para que puedan ser leídos por el paquete básico de R. Aunque debemos mencionar que R cuenta con paquetes que se pueden descargar para leer directamente los archivos con extensión `*.sav` y `*.xls` o `*.xlsx`.

De manera específica, en esta parte usaremos el archivo `MMR1.sav`,<sup>3</sup> construido en base a la fusión de dos archivos de la Encuesta Demográfica y de Salud Familiar-Endes, Perú, 2012: `REC0111.sav` y `REC83.sav`, integrantes de los módulos: 323-Modulo66 y 323-Modulo73, respectivamente, y disponibles en: <http://iinei.inei.gob.pe/microdatos/>. Se trata de explorar diversas maneras de leer su contenido desde R.

Este archivo contiene información sobre 126 mujeres de 15 a 49 años, fallecidas en circunstancias relacionadas con el embarazo, parto o aborto. Esta información fue dada por sus hermanas sobrevivientes, las que fueron entrevistadas en la Endes-2012<sup>4</sup>. El archivo contiene 28 variables.

### 4.1. Preparar archivos externos que puedan ser leídos por el paquete básico de R

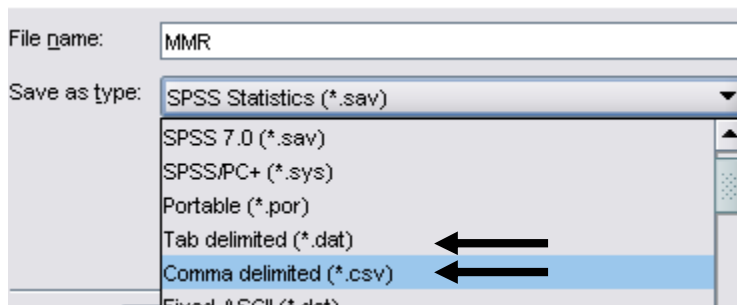
Para transportar archivos creados en Excel o en SPSS que puedan ser leídos por el paquete básico de R, primero seleccionamos la opción **Guardar como/Save as**; luego abrimos la pestaña **Guardar como tipo/Save as type**; y finalmente seleccionamos un formato. Los formatos disponibles desde Excel son: `*.csv` o `*.txt`:



<sup>3</sup> A partir del archivo `MMR1.sav`, se generaron archivos con otras extensiones como: `MMR.csv`. Ver imagen completa de `MMR.csv` en Anexo 1.

<sup>4</sup> Por ello en la variable de identificación de la base de datos `CASEID`, podemos observar aparentes duplicaciones de datos para los casos (filas) 13, 80 y 82. Sin embargo debemos recordar que el objeto de análisis no son las hermanas sobrevivientes (123) sino las hermanas fallecidas (126).

Y los formatos disponibles desde SPSS son: \*.csv o \*.dat.



Los archivos resultantes serán:



## 4.2. Leer archivos desde R

### 4.2.1. Leer archivos con la función `read.table()`

#### 4.2.1.1. Leer archivos especificando el nombre del archivo

- *Leer archivos delimitados por comas (\*.csv)*

```
> mater1 = read.table(file="MMR.csv", header=TRUE, sep=",")
> mater1
```

	CASEID	MMIDX	MM1	MM2	MM3	MM4	MM5	MM6	MM7	MM8	MM9	MM10	MM11	MM12	
1	000804001	O3	7	2	0	NA	NA	NA	14	25	NA	2	NA	1	NA
2	001009101	O2	2	2	0	NA	NA	NA	6	30	NA	2	NA	3	NA
3	001202101	O2	1	2	0	NA	NA	NA	NA	38	NA	2	NA	3	NA
4	001607501	O3	3	2	0	NA	NA	NA	11	14	NA	2	NA	1	NA
5	002106101	O2	2	2	0	NA	NA	NA	36	18	NA	2	NA	1	NA

Donde:

`mater1` es un objeto de tipo marco de datos para R.

`read.table` es la función que nos permite leer el archivo `MMR.csv` desde R

`header` es un argumento de la función `read.table` que lee la primera fila del archivo `MMR.csv`, como una fila que contiene los nombres de las variables de la base de datos

`sep` indica el elemento que actúa como separador de los datos en este caso, la coma

- *Leer archivos delimitados por tabulaciones (\*.txt)*

```
> mater2 = read.table(file="MMRE.txt", header=TRUE, sep="\t")
> mater2
```

	CASEID	MMIDX	MM1	MM2	MM3	MM4	MM5	MM6	MM7	MM8	MM9	MM10	
1	000804001	03	7	2	0	NA	NA	NA	14	25	NA	2	NA
2	001009101	02	2	2	0	NA	NA	NA	6	30	NA	2	NA
3	001202101	02	1	2	0	NA	NA	NA	NA	38	NA	2	NA

En este caso `sep="\t"`, indica que las tabulaciones son los separadores de datos.

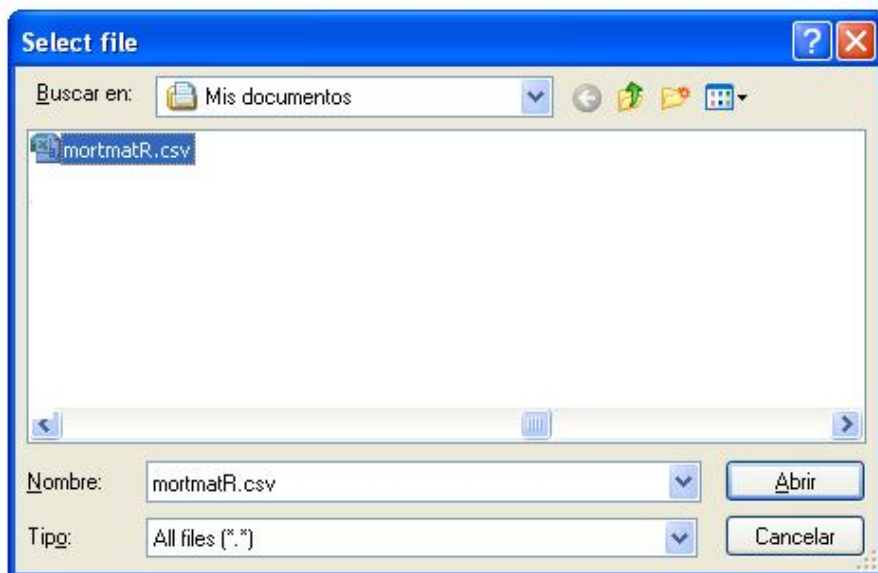
- Leer archivos delimitados por tabulaciones (\*.dat)

```
> mater3 = read.table(file="MM.dat", header=TRUE, sep="\t")
> mater3
```

	CASEID	MMIDX	MM1	MM2	MM3	MM4	MM5	MM6	MM7	MM8	MM9	MM10	
1	000804001	03	7	2	0	NA	NA	NA	14	25	NA	2	NA
2	001009101	02	2	2	0	NA	NA	NA	6	30	NA	2	NA
3	001202101	02	1	2	0	NA	NA	NA	NA	38	NA	2	NA

#### 4.2.1.2. Leer archivo seleccionándolos desde una carpeta

```
> mater4 = read.table(file.choose(), header=TRUE, sep=",")
```



```
> mater4
```

	CASEID	MMIDX	MM1	MM2	MM3	MM4	MM5	MM6	MM7	MM8	MM9	MM10	
1	000804001	03	7	2	0	NA	NA	NA	14	25	NA	2	NA
2	001009101	02	2	2	0	NA	NA	NA	6	30	NA	2	NA
3	001202101	02	1	2	0	NA	NA	NA	NA	38	NA	2	NA

#### 4.2.1.3. Leer un segmento de archivo

- Leer un segmento de archivo de Excel

Primero seleccionamos en Excel, la hoja o el segmento de hoja que queremos leer, y activamos la opción copiar.

	A	B	C	D	E
1	CASEID	MMIDX	MM1	MM2	MM3
2	000804001 03	7	2	0	
3	001009101 02	2	2	0	
4	001202101 02	1	2		
5	001607501 03	3	2		
6	002106101 02	2	2		
7	005700501 02	3	2		
8	007703301 02	3	2		
9	008101201 02	2	2		

Luego escribimos en la Consola lo siguiente:

```
> mater5 = read.table("clipboard")
> mater5
      CASEID MMIDX MM1 MM2
000804001    3    7  2  0
001009101    2    2  2  0
001202101    2    1  2  0
001607501    3    3  2  0
002106101    2    2  2  0
005700501    2    3  2  0
```

En caso que seleccionemos un segmento que no incluya la primera fila, la cual por lo general, contiene el nombre de las variables, R les asignará los siguientes nombres: V1, V2, V3...Vn.

	A	B	C	D	E
1	CASEID	MMIDX	MM1	MM2	MM3
2	000804001 03	7	2	0	
3	001009101 02	2	2	0	
4	001202101 02	1	2		
5	001607501 03	3	2		
6	002106101 02	2	2		
7	005700501 02	3	2		
8	007703301 02	3	2		
9	008101201 02	2	2		
10	008106601 02	1	2		

```
> mater6 = read.table("clipboard")
> mater6
      V1 V2 V3 V4 V5
1 1202101 2 1 2 0
2 1607501 3 3 2 0
3 2106101 2 2 2 0
4 5700501 2 3 2 0
```

Sin embargo, atención con este procedimiento. Como se puede observar, R ha dividido la variable CASEID en V1 y V2, y ha anulado los ceros a la izquierda que identifican a la variable CASEID como una variable de tipo carácter, y no de tipo numérico.

- Leer un segmento de archivo de SPSS

Los procedimientos son los mismos que cuando se trabaja con un segmento de Excel.

	CASEID	MMIDX	MM1	MM2	MM3
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		
8	008101201 02	2	2		
9	008106601 02	1	2		
10	010700701 02	1	2		
11	010804701 02	5	2		
12	012002901 02	1	2		
13	012002901 02	10	2	0	
14	012602401 02	9	2	0	

```
> mater7 = read.table("clipboard")
```

```
> mater7
```

```
      V1 V2 V3 V4 V5
1      804001 3 7 2 0
2     1009101 2 2 2 0
3     1202101 2 1 2 0
4     1607501 3 3 2 0
5     2106101 2 2 2 0
6     5700501 2 3 2 0
7     7703301 2 3 2 0
8     8101201 2 2 2 0
9     8106601 2 1 2 0
10    10700701 2 1 2 0
11    10804701 5 2 2 0
```

La diferencia con Excel es que, al copiar un segmento de archivo SPSS no se trasladan los nombres originales de las variables, incluso si las incluimos en la selección. SPSS les asigna nuevos nombres. Y al igual que en Excel cuando copiábamos un segmento de archivo que no incluía las filas con los nombres de las variables, se debe tener cuidado con la variable CASEID, ya que R la descompone en dos variables (V1 y V2).

### 4.2.2. Leer archivos con la función `read.csv()`

```
> mater8 = read.csv(file="MATERMAT.csv")
> mater8
```

	CASEID	MMIDX	MM1	MM2	MM3	MM4	MM5	MM6	MM7	MM8	MM9	MM10	
1	000804001	03	7	2	0	NA	NA	NA	14	25	NA	2	NA
2	001009101	02	2	2	0	NA	NA	NA	6	30	NA	2	NA
3	001202101	02	1	2	0	NA	NA	NA	NA	38	NA	2	NA

### 4.3. Explorar el contenido de un archivo

- Ver el nombre de todas las variables de un archivo usando la función `names()`

```
> names(mater1)
```

[1]	"CASEID"	"MMIDX"	"MM1"	"MM2"	"MM3"	"MM4"	"MM5"
[8]	"MM6"	"MM7"	"MM8"	"MM9"	"MM10"	"MM11"	"MM12"
[15]	"MM13"	"MM14"	"MM15"	"MM16"	"MMC1"	"MMC2"	"filter_."
[22]	"V001"	"V002"	"V003"	"V101"	"V102"	"V103"	"V190"

- Listar todos los casos de una variable

```
> mater1$V190
```

[1]	2	3	1	1	1	2	1	1	1	3	2	3	3	2	1	1	1	2	2	1	1	1	1	1	2	4	1	3	3	2	3	2	2	5	1	2	3
[38]	5	4	2	3	5	3	2	2	2	3	3	4	1	1	2	1	1	2	4	2	4	4	3	1	1	1	1	1	2	1	1	1	1	1	4	2	1
[75]	2	1	1	3	1	1	1	2	3	1	1	2	3	5	4	3	1	1	1	1	1	1	2	2	1	1	1	5	2	5	1	1	1	1	3	2	
[112]	2	2	1	3	3	3	2	2	2	1	2	5	4	3	3																						

- Listar el nombre de las filas [1] (casos) y el de las columnas [2] (variables) usando la función `labels`

```
> labels(mater1)
```

```
[[1]]
```

[1]	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"	"10"	"11"	"12"
[13]	"13"	"14"	"15"	"16"	"17"	"18"	"19"	"20"	"21"	"22"	"23"	"24"
[25]	"25"	"26"	"27"	"28"	"29"	"30"	"31"	"32"	"33"	"34"	"35"	"36"
[37]	"37"	"38"	"39"	"40"	"41"	"42"	"43"	"44"	"45"	"46"	"47"	"48"
[49]	"49"	"50"	"51"	"52"	"53"	"54"	"55"	"56"	"57"	"58"	"59"	"60"
[61]	"61"	"62"	"63"	"64"	"65"	"66"	"67"	"68"	"69"	"70"	"71"	"72"
[73]	"73"	"74"	"75"	"76"	"77"	"78"	"79"	"80"	"81"	"82"	"83"	"84"
[85]	"85"	"86"	"87"	"88"	"89"	"90"	"91"	"92"	"93"	"94"	"95"	"96"
[97]	"97"	"98"	"99"	"100"	"101"	"102"	"103"	"104"	"105"	"106"	"107"	"108"
[109]	"109"	"110"	"111"	"112"	"113"	"114"	"115"	"116"	"117"	"118"	"119"	"120"
[121]	"121"	"122"	"123"	"124"	"125"	"126"						

```
[[2]]
```

[1]	"CASEID"	"MMIDX"	"MM1"	"MM2"	"MM3"	"MM4"
[7]	"MM5"	"MM6"	"MM7"	"MM8"	"MM9"	"MM10"
[13]	"MM11"	"MM12"	"MM13"	"MM14"	"MM15"	"MM16"
[19]	"MMC1"	"MMC2"	"filter_."	"V001"	"V002"	"V003"
[25]	"V101"	"V102"	"V103"	"V190"		

```
> |
```

- Conocer el número de variables que contiene un marco de datos usando la función `length`

```
> length(mater1)
[1] 28
```

- Listar un intervalo de casos de una variable del marco de datos haciendo referencia a su nombre

Ejemplo: listar los 10 primeros casos de la variable `V190`. Es decir, las diez primeras filas de la variable `V190`.

```
> mater1$V190[1:10]
[1] 2 3 1 1 1 2 1 1 1 3
```

- Listar un intervalo de casos de una variable del marco de datos haciendo referencia a su posición en la base de datos

Ejemplo: listar los 10 primeros casos de la variable `V190`. Es decir, las diez primeras filas de la columna 28.

```
> mater1[1:10,28]
[1] 2 3 1 1 1 2 1 1 1 3
```

#### 4.4. Segmentar archivos

La variable `V190`,<sup>5</sup> del marco de datos `mater1`, contiene información para 126 casos, sobre los niveles socioeconómicos de las hermanas que brindaron información sobre sus hermanas fallecidas:

```
> mater1 = read.table(file="MMR.csv", header=TRUE, sep=",")
> mater1$V190
 [1] 2 3 1 1 1 2 1 1 1 3 2 3 3 2 1 1 1 2 2 1 1 1 1 1 2 4 1 3 3 2 3 2 2 5 1 2 3
[38] 5 4 2 3 5 3 2 2 2 3 3 4 1 1 2 1 1 2 4 2 4 4 3 1 1 1 1 1 2 1 1 1 1 1 4 2 1
[75] 2 1 1 3 1 1 1 1 2 3 1 1 2 3 5 4 3 1 1 1 1 1 1 2 2 1 1 1 5 2 5 1 1 1 1 3 2
[112] 2 2 1 3 3 3 2 2 2 1 2 5 4 3 3
```

Donde: 1= Muy pobre, 2= Pobre, 3=Medio, 4=Rico, 5=Muy rico.

Por ejemplo si quisiéramos trabajar solo con las hermanas, que brindaron información, que viven en situaciones de pobreza, entonces debemos segmentar el marco de datos.

Para ello usaremos la función `subset()` de la siguiente manera:

```
> pobres <- subset(mater1, subset=V190<3)
```

Donde:

---

<sup>5</sup> Etiquetada como "índice de riqueza" en la base de datos original.



- `pobres` Nombre del nuevo marco de datos que contendrá los elementos seleccionados.
- `subset` Función que selecciona todos los elementos del objeto (en este caso del marco de datos `mater1`) que deben conservarse, en este caso, todos los elementos de `V190` menores al código 3.

La opción `labels` nos permite conocer los casos (filas) que han sido seleccionadas para integrar el nuevo marco de datos, y además nos lista los nombres de las variables consideradas en él.

```
> labels(pobres)
[[1]]
 [1] "1"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "11" "14" "15" "16"
[13] "17" "18" "19" "20" "21" "22" "23" "24" "25" "27" "30" "32"
[25] "33" "35" "36" "40" "44" "45" "46" "50" "51" "52" "53" "54"
[37] "55" "57" "61" "62" "63" "64" "65" "66" "67" "68" "69" "70"
[49] "71" "73" "74" "75" "76" "77" "79" "80" "81" "82" "83" "85"
[61] "86" "87" "92" "93" "94" "95" "96" "97" "98" "99" "100" "101"
[73] "102" "104" "106" "107" "108" "109" "111" "112" "113" "114" "118" "119"
[85] "120" "121" "122"

[[2]]
 [1] "CASEID"  "MMIDX"   "MM1"     "MM2"     "MM3"     "MM4"
 [7] "MM5"     "MM6"     "MM7"     "MM8"     "MM9"     "MM10"
[13] "MM11"    "MM12"    "MM13"    "MM14"    "MM15"    "MM16"
[19] "MMC1"    "MMC2"    "filter_." "V001"    "V002"    "V003"
[25] "V101"    "V102"    "V103"    "V190"
```

El nuevo marco de datos al que hemos llamado: “`pobres`”, contiene información sobre 87 casos y 28 variables.

A continuación podemos verificar, que ahora, sólo los casos que registraron códigos 1 o 2 forman parte de variable `V190`.

```
> pobres$V190
 [1] 2 1 1 1 2 1 1 1 2 2 1 1 1 2 2 1 1 1 1 2 1 2 2 2 1 2 2 2 2 1 1 2 1 1 2 2
[39] 1 1 1 1 1 2 1 1 1 1 1 2 1 2 1 1 1 1 1 2 1 1 2 1 1 1 1 1 2 2 1 1 1 2 1 1
[77] 1 1 2 2 2 1 2 2 2 1 2
```

#### 4.5. Guardar y recargar marcos de datos

- *Guardar usando `save()`*

En el punto anterior se creó el marco de datos “`pobres`”, si queremos usarlo en posteriores sesiones de trabajo, necesitamos guardarlo en un archivo. Para ello utilizamos la función `save()` de la siguiente manera:

```
> save(pobres, file="C:/Documents and Settings/Administrador/Mis documentos/POOR1.RData")
```



POOR1.RData

El archivo "POOR1.RData" guarda el objeto: marco de datos "pobres".

- *Recargar un archivo de extensión \*.RData en una nueva sesión de trabajo usando load()*

```
> load("C:/Documents and Settings/Administrador/Mis documentos/POOR1.RData")
> pobres
```

	CASEID	MMIDX	MM1	MM2	MM3	MM4	MM5	MM6	MM7	MM8	MM9	MM10	MM11	MM12	
1	000804001	03	7	2	0	NA	NA	NA	14	25	NA	2	NA	1	NA
3	001202101	02	1	2	0	NA	NA	NA	NA	38	NA	2	NA	3	NA
4	001607501	03	3	2	0	NA	NA	NA	11	14	NA	2	NA	1	NA

También podemos cargar los archivos con extensión \*.RData, usando R Commander (ver 9.3.1.1).

- *Guardar marcos de datos usando write.table()*

En el ejemplo siguiente usamos una versión recortada del marco de datos mater1, en donde hemos eliminado las variables MM1, MM2, MM3, MM4, MM5, MM8, MM10, MM12 y MM14, para crear el archivo delimitado por comas: "ahora.csv".

```
> write.table(mater1,
+ file = "D:/ahora.csv",
+ sep = ",",
+ col.names = NA)
```

- *Recargar usando read.table*

En una sesión posterior en R, podemos leer el archivo "ahora.csv" en R.

```
> mater11 = read.table(file="D:/ahora.csv", header=TRUE, sep=",")
> mater11
```

	X	CASEID	MMIDX	MM6	MM7	MM9	MM11	MM13	MM14	MM15	MMC1	MMC2	filter_.	V001	V002	V003	V101	V102	V103	V190	
1	1	000804001	03	7	14	25	2	1	2	3	9998	7	7	1	8	40	3	1	2	2	2
2	2	001009101	02	2	6	30	2	3	1	5	9998	8	4	1	10	91	2	1	1	2	3
3	3	001202101	02	1	NA	38	2	3	3	7	2012	5	3	1	12	21	2	1	2	3	1
4	4	001607501	03	3	11	14	2	1	2	3	9998	7	7	1	8	40	3	1	2	2	2

Podemos observar que R ha impreso la primera columna de mater1, como la variable x, que en realidad corresponde al número de línea del marco de datos mater1.

## 5. TRATAMIENTO DE VARIABLES

En este capítulo usaremos el marco de datos `mater1`, construido en el capítulo anterior.

### 5.1. Convertir vectores en factores o crear variables cualitativas

Para crear variables cualitativas en R, debemos convertir los vectores numéricos (variables cuantitativas) de un marco de datos, en factores (variables cualitativas).

Como ya vimos en 4.4, la variable `V190` da cuenta de los niveles socioeconómicos de las mujeres entrevistadas. Donde: 1=Muy pobre; 2=Pobre, 3=Medio, 4=Rico y 5=Muy Rico.

```
> mater1 = read.table(file="MMR.csv", header=TRUE, sep=",")
> mater1[28]
      V190
1         2
2         3
3         1
4         1
5         1
6         2
7         1
```

Como vemos, la variable `V190` es un vector numérico. Para convertirlo en un factor (variable cualitativa) con el mismo nombre (`V190`), escribimos las siguientes instrucciones:

```
mater1<-transform(mater1,
V190=factor(V190,
labels=c("Muy pobre","Pobre","Medio","Rico","Muy rico")))
```

Podemos observar los cambios de vector a factor de la variable `V190`, escribiendo la siguiente instrucción antes y después de la transformación:

```
summary(mater1[28])
```

V190 antes	V190 después
<pre> V190 Min.   :1.000 1st Qu.:1.000 Median :2.000 Mean   :2.063 3rd Qu.:3.000 Max.   :5.000</pre>	<pre> V190 Muy pobre:54 Pobre    :33 Medio    :23 Rico     : 9 Muy rico : 7</pre>

También se puede observar estos cambios desde la ventana del Editor de datos:

R Editor de datos							
	V001	V002	V003	V101	V102	V103	V190
1	8	40	3	1	2	2	2
2	10	91	2	1	1	2	3
3	12	21	2	1	2	3	1
4	16	75	3	1	2	3	1

R Data Editor							
	V001	V002	V003	V101	V102	V103	V190
1	8	40	3	1	2	2	Pobre
2	10	91	2	1	1	2	Medio
3	12	21	2	1	2	3	Muy pobre
4	16	75	3	1	2	3	Muy pobre

Sin embargo es preferible crear una variable cualitativa con un nombre asociado a la variable de origen, por ejemplo V190REC

```
> mater1<-transform(mater1,
+ V190REC=factor(V190,
+ labels=c("Muy pobre","Pobre","Medio","Rico","Muy rico")))
```

R Data Editor				
	V102	V103	V190	V190REC
1	2	2	2	Pobre
2	1	2	3	Medio
3	2	3	1	Muy pobre
4	2	3	1	Muy pobre
5	2	3	1	Muy pobre
6	2	2	2	Pobre

## 5.2. Eliminar variables de un marco de datos

Eliminamos de `mater1`, 7 variables que no contienen información: `MM3`, `MM4`, `MM5`, `MM8`, `MM10`, `MM12` y `MM16` y 2 constantes: `MM1` y `MM2`. Para ello, escribimos las siguientes instrucciones:

```
> mater1 = read.table(file="MMR.csv", header=TRUE, sep=",")
> mater1$MM1 <- NULL
> mater1$MM2 <- NULL
> mater1$MM3 <- NULL
> mater1$MM4 <- NULL
> mater1$MM5 <- NULL
> mater1$MM8 <- NULL
> mater1$MM10 <- NULL
> mater1$MM12 <- NULL
> mater1$MM16 <- NULL
```

El resultado es una un marco de datos con 19 variables.

```
> length(mater1)
[1] 19
```

	CASEID	MMIDX	MM6	MM7	MM9	MM11	MM13	MM14	MM15	MMC1	MMC2	filter_	V001	V002	V003	V101	V102	V103	V190
1	000804001 03	7	14	25	2	1	2	3	9998	7	7	1	8	40	3	1	2	2	2
2	001009101 02	2	6	30	2	3	1	5	9998	8	4	1	10	91	2	1	1	2	3
3	001202101 02	1	NA	38	2	3	3	7	2012	5	3	1	12	21	2	1	2	3	1
4	001607501 03	3	11	14	2	1	3	0	9998	6	6	1	16	75	3	1	2	3	1

### 5.3. Renombrar variables

Para renombrar variables utilizamos la función `names()`.

Ejemplo: renombrar la variable V101 (región de residencia) como DPTO.

```
> names(mater1)[c(16)] <- c("DPTO")
```

Donde:

`[c(16)]` es la posición de la variable V101, en el marco de datos `mater1`, donde realizaremos la modificación. En este caso, la columna 16 (donde se ubica la variable V101).

`c("DPTO")` Nuevo nombre a asignar a la posición 16 del marco de datos `mater1`.

Antiguo nombre de la variable: V101

V003	V101	V102
3	1	2
2	1	1
2	1	2

Nuevo nombre de la variable: DPTO

V003	DPTO	V102
3	1	2
2	1	1
2	1	2

### 5.4. Crear nuevas variables a partir de otras existentes haciendo cálculos

Crearemos nuevas variables, a partir de cálculos matemáticos que efectuaremos sobre la variable MMC1 (número de hermanos).

- *Logaritmo natural de una variable*

```
> mater1$LGMMC1 <- with(mater1, log(MMC1))
```

Donde:

LGMMC1 es el nombre de la variable a crear en el marco de datos `mater1`

`with()` Es la función que permite la creación de la nueva variable luego de realizar la función `log()` en la base de datos `mater1`.

`log()` función que permite calcular el logaritmo natural de una variable cuantitativa.

- *Valores tipificados de una variable*

El valor tipificado mide el número de desviaciones estándar que hay entre un valor dado y la media.

Para crear una variable que represente valores tipificados de otra seguiremos los siguientes pasos: primero crearemos un **vector numérico** (VZ) asignándole como elementos, los valores tipificados de la variable a tipificar (MMC1), calculadas con la función `scale()`. Luego creamos la **variable con valores tipificados** (VZMMC1) en el marco de datos (`mater1`), asignándole para ello, el vector numérico (VZ) creado en el paso anterior.

```
> VZ <- scale(mater1[,c("MMC1")])
> mater1$VZMMC1 <- VZ[,1]
```

- *Raíz cuadrada de una variable*

```
> mater1$RCSMMC1 <- with(mater1, sqrt(MMC1))
```

- *Raíz cuadrada de una variable, redondeada a un dígito*

```
> mater1$RCRMMC1 <- with(mater1, round(sqrt(MMC1),1))
```

A continuación vemos los resultados de estos cuatro procedimientos en `mater1`.

MMC1	LGMMC1	VZMMC1	RCSMMC1	RCRMMC1
7	1.9459101	-0.2130127	2.645751	2.6
8	2.0794415	0.1546530	2.828427	2.8
5	1.6094379	-0.9483440	2.236068	2.2
6	1.7917595	-0.5806783	2.449490	2.4

Dado que la variable `RCSMMC1` es la raíz cuadrada de la variable `MMC1`, y la variable `RCRMMC1` es la raíz cuadrada redondeada a un dígito, de la variable `MMC1`, sus medidas de tendencia central son diferentes.

RCSMMC1		RCRMMC1	
Min.	:1.000	Min.	:1.000
1st Qu.	:2.289	1st Qu.	:2.250
Median	:2.828	Median	:2.800
Mean	:2.703	Mean	:2.687
3rd Qu.	:3.162	3rd Qu.	:3.200
Max.	:3.873	Max.	:3.900

## 5.5. Recodificar una variable numérica usando la función `recode()` o `Recode()`

`recode()` es una función que sirve para recodificar vectores numéricos, de carácter o factores. En este caso `Recode` es un alias de la función `recode`, y se utiliza para evitar

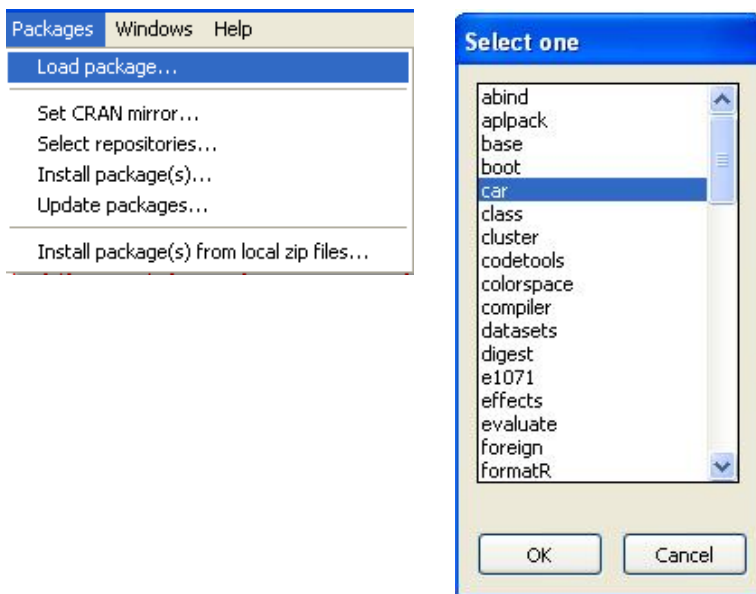
conflictos con funciones similares entre ciertos paquetes de R. (Fox, 2011 en R Documentation).

Para activar la función `recode()` necesitamos cargar el paquete `car`, de la siguiente manera:

Desplegamos el menú Packages → Load package...

En la ventana emergente “Select one”, seleccionamos `car`,

Finalmente hacemos clic sobre el botón OK



Ejemplo: Reducir a tres categorías la variable `V190` que tiene cinco (ver 5.1), creando la variable `NSE` (Nivel socioeconómico).

Para ello crearemos un factor numérico usando la función `Recode()`:

```
> mater1$NSE<-Recode(mater1$V190,
+ '1:2=1; 3=2; 4:5=3',
+ as.factor.result=TRUE)
```

También podemos crear un factor carácter usando la misma función:

```
> mater1$NSECTG<-Recode(mater1$V190,
+ '1:2="Pobre"; 3="Medio"; 4:5="Rico"',
+ as.factor.result=TRUE)
```

A continuación usando `table()`, podemos observar los resultados de ambos procedimientos:

```
> table(mater1$NSE)
 1  2  3
87 23 16
> table(mater1$NSECTG)
Medio Pobre Rico
 23   87  16
```

En la tabla de frecuencia de la variable NSE, los resultados se presentan en orden numérico, mientras que en la de la variable NSECTG el orden es alfabético. Para forzar el orden “Pobre-Medio-Rico” en la variable NSECTG, debemos agregar el argumento `levels` a la función `Recode`, de la siguiente manera:

```
> mater1$NSECTG<-Recode(mater1$V190,
+ '1:2="Pobre"; 3="Medio"; 4:5="Rico"',
+ as.factor.result=TRUE, levels=c("Pobre", "Medio", "Rico"))
```

Donde:

`levels` indica en qué orden deben presentarse las categorías .

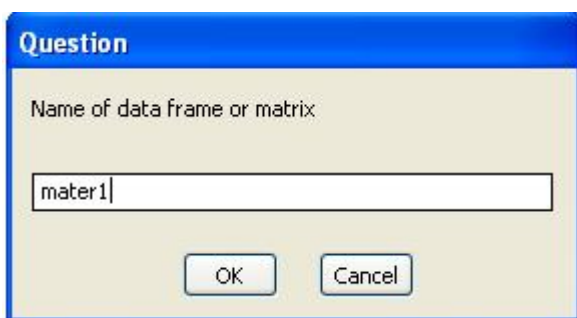
Verificamos los resultados

```
> table(mater1$NSECTG)
Pobre Medio Rico
 87   23  16
```

## 5.6. Modificar datos desde el Editor

En 2.2. describimos la ventana de edición de datos. Aquí ilustramos cómo acceder concretamente a ella para editar datos. Para ello desplegamos el menú:

Edit → Data editor...



Al abrirse el editor de datos, en la Consola se escribirá automáticamente la siguiente instrucción:

```
R Console
> fix(mater1)
```



Una vez abierto el editor, podemos hacer modificaciones directamente sobre la base de datos, como en el siguiente ejemplo: fila 3 de la variable MM6.

	CASEID	MMIDX	MM6
1	000804001 03	7	14
2	001009101 02	2	6
3	001202101 02	1	NA
4	001607501 03	3	11

	CASEID	MMIDX	MM6
1	000804001 03	7	14
2	001009101 02	2	6
3	001202101 02	1	1
4	001607501 03	3	11

## 6. ANÁLISIS UNIVARIADO

Para comenzar a trabajar en este capítulo, primero vinculamos el marco de datos `mater1`<sup>6</sup> con la siguiente instrucción:

```
> attach(mater1)
```

Ver detalles de las razones de este procedimiento en 3.1.5.

### 6.1. Distribución de frecuencias

En esta parte elaboraremos tablas de distribución de frecuencias en valores absolutos y en valores relativos.

#### 6.1.1. Frecuencia absoluta

Para crear una tabla de distribución de frecuencias, usamos la función `table()`.

Ejemplo: crear una tabla de distribución de frecuencias de la variable “nivel socioeconómico” (NSECTG) de las mujeres que fueron entrevistadas.

```
> table(NSECTG)
NSECTG
Pobre Medio Rico
  87    23   16
```

- *Convertir una tabla de frecuencia absoluta en un objeto*

El objetivo de esta conversión, es poder tratar a la tabla resultante, como objeto base para hacer operaciones con ella y así generar otras tablas.

Ejemplo: Convertir en el objeto `Cuadro1`, la tabla de frecuencia de la variable `NSECTG`.

Creamos el objeto `Cuadro1`, usando el operador asignar (`<-`) de la siguiente manera:

```
> Cuadro1 <- table(NSECTG)
> Cuadro1
NSECTG
Pobre Medio Rico
  87    23   16
```

En adelante nos referiremos al objeto `Cuadro1` cada vez que queramos utilizar la tabla de distribución de frecuencia en valores absolutos de la variable `NSECTG`.

- *Frecuencia absoluta acumulada usando la función `cumsum()`*

---

<sup>6</sup> Ver información sobre el contenido de esta base de datos en el Capítulo 4.

```
> cumsum(Cuadro1)
Pobre Medio Rico
  87   110   126
```

### 6.1.2. Frecuencia relativa

- *Tabla de frecuencia relativa (proporciones)*

```
> Cuadro1/margin.table(Cuadro1)
NSECTG
  Pobre   Medio   Rico
0.6904762 0.1825397 0.1269841
```

- *Tabla de frecuencia relativa en porcentajes y redondeada a dos dígitos*

```
> round((Cuadro1/margin.table(Cuadro1))*100,2)
NSECTG
Pobre Medio Rico
69.05 18.25 12.70
```

- *Frecuencia relativa acumulada*

```
> cumsum(round((Cuadro1/margin.table(Cuadro1))*100,2))
Pobre Medio Rico
69.05 87.30 100.00
```

## 6.2. Medidas de tendencia central

En este punto trabajaremos con la variable cuantitativa MMC1 (Número de hermanos) del marco de datos mater1.

- *Media*

```
> mean(MMC1)
[1] 7.579365
```

- *Mediana*

```
> median(MMC1)
[1] 8
```

- *Cuartiles*

```
> quantile(MMC1)
  0%   25%   50%   75%  100%
1.00  5.25  8.00 10.00 15.00
```

## 6.3. Medidas de dispersión

- *Rango*

```
> range(MMC1)
[1] 1 15
```

- *Varianza*

```
> var(MMC1)
[1] 7.397651
```

- *Desviación estándar*

```
> sd(MMC1)
[1] 2.719862
```

- *Valores estandarizados o tipificación de una variable*

Ver 5.4.

## 6.4. Medidas de posición

Para ejecutar instrucciones sobre medidas de posición se requiere cargar el paquete `e1071` (Ver como cargar un paquete en 5.5)

- *Sesgo*

```
> skewness(MMC1)
[1] 0.0005004876
```

- *Curtosis*

```
> kurtosis(MMC1)
[1] -0.4139902
```

## 6.5. Resumir medidas estadísticas de todas las variables de una base de datos

La función `summary()` muestra la media, mediana, cuartiles, valor mínimo y valor máximo, para variables cuantitativas y la frecuencia absoluta para variables cualitativas. Ejemplo: resumir medidas estadísticas del marco de datos `mater1`.

```
> summary(mater1)
      CASEID      MMIDX      MM6      MM7      MM9
012002901 02: 2   Min.   : 1.000   Min.   : 0.0   Min.   :14.00   Min.   :1.000
086704101 03: 2   1st Qu.: 1.000   1st Qu.:11.0   1st Qu.:20.00   1st Qu.:2.000
086704301 02: 2   Median : 3.000   Median :19.0   Median :23.00   Median :2.000
000804001 03: 1   Mean   : 3.016   Mean   :19.7   Mean   :25.25   Mean   :1.754
001009101 02: 1   3rd Qu.: 4.000   3rd Qu.:26.0   3rd Qu.:30.00   3rd Qu.:2.000
001202101 02: 1   Max.   :10.000   Max.   :98.0   Max.   :47.00   Max.   :2.000
(Other)      :117      NA's    :33

      MM11      MM13      MM14      MM15      MMC1      MMC2
Min.   :1.000   Min.   :1.000   Min.   : 0.000   Min.   :1983   Min.   : 1.000   Min.   : 0.00
1st Qu.:2.000   1st Qu.:1.000   1st Qu.: 0.000   1st Qu.:2012   1st Qu.: 5.250   1st Qu.: 3.00
Median :3.000   Median :2.000   Median : 1.500   Median :9998   Median : 8.000   Median : 4.00
Mean   :2.294   Mean   :2.143   Mean   : 2.167   Mean   :7904   Mean   : 7.579   Mean   : 4.69
3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.: 3.000   3rd Qu.:9998   3rd Qu.:10.000   3rd Qu.: 7.00
Max.   :3.000   Max.   :3.000   Max.   :12.000   Max.   :9998   Max.   :15.000   Max.   :11.00

      filter_      V001      V002      V003      DPTO      V102
Min.   :0.0000   Min.   : 8.0   Min.   : 1.0   Min.   : 1.000   Min.   : 1.00   Min.   :1.00
1st Qu.:0.0000   1st Qu.:356.5   1st Qu.:25.0   1st Qu.: 2.000   1st Qu.: 6.00   1st Qu.:1.00
Median :0.0000   Median :724.5   Median :49.5   Median : 2.000   Median :11.00   Median :2.00
Mean   :0.1587   Mean   :684.8   Mean   :54.6   Mean   : 2.468   Mean   :11.52   Mean   :1.54
3rd Qu.:0.0000   3rd Qu.:952.0   3rd Qu.:75.0   3rd Qu.: 2.000   3rd Qu.:17.00   3rd Qu.:2.00
Max.   :1.0000   Max.   :1378.0   Max.   :211.0   Max.   :10.000   Max.   :25.00   Max.   :2.00

      V103      V190      LGMMC1      Z.MMC1      RCSMMC1      RCRMMC1
Min.   :0.000   Min.   :1.000   Min.   :0.000   Min.   :-2.4190   Min.   :1.000   Min.   :1.000
1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.655   1st Qu.:-0.8564   1st Qu.:2.289   1st Qu.:2.250
Median :3.000   Median :2.000   Median :2.079   Median : 0.1547   Median :2.828   Median :2.800
Mean   :2.246   Mean   :2.063   Mean   :1.945   Mean   : 0.0000   Mean   :2.703   Mean   :2.687
3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:2.303   3rd Qu.: 0.8900   3rd Qu.:3.162   3rd Qu.:3.200
Max.   :3.000   Max.   :5.000   Max.   :2.708   Max.   : 2.7283   Max.   :3.873   Max.   :3.900

NSE      NSECTG
1:87     Medio:23
2:23     Pobre:87
3:16     Rico :16
```

## 6.6. Resumir medidas estadísticas para una variable de la base de datos

```
> summary(mater1[,"MMC1"])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.000  5.250   8.000   7.579 10.000  15.000
> summary(mater1[,"NSECTG"])
Pobre Medio Rico
  87   23   16
```

Donde: [,"MMC1"], nos indica la columna de mater1 que debe resumirse.

## 7. ANÁLISIS BIVARIADO

En este punto realizaremos cuatro tipos de análisis bivariado: análisis descriptivo con tablas cruzadas, test de asociación o de independencia con Chi cuadrado, pruebas  $t$  para la diferencia de medias y correlación bivariada.

### 7.1. Tablas cruzadas

También llamadas tablas de contingencia. Son tablas construidas en base a dos variables: una variable fila y una variable columna. A partir de esta disposición de los datos se puede establecer relaciones entre estas dos variables.

#### 7.1.1. Tablas cruzadas con valores absolutos

Para elaborar tablas cruzadas usamos la función `table()`.

Ejemplo: elaborar una tabla cruzada de las variables: nivel socioeconómico (`V190REC`)<sup>7</sup> y lugar de residencia en la niñez (`V103REC`).

Para ello convertiremos el vector `V103` (correspondiente a la variable ENDES `V103`, lugar de residencia en la niñez, con 5 alternativas de respuesta: 0=Capital, ciudad grande, 1=Ciudad, 2=Pueblo, 3=Campo, 4=Exterior), en el factor `V103REC` con 2 alternativas de respuesta: “Urbano” y “Rural”. Para ello cargaremos el paquete `car` y escribiremos las siguientes instrucciones:

```
> mater1$V103REC<-recode(mater1$V103,
+ '0:2="Urbano"; 3="Rural"',
+ as.factor.result=TRUE)
```

Activamos la función `attach()`:

```
> attach(mater1)
```

Verificamos la recodificación con:

```
> table(V103REC)
V103REC
Rural Urbano
  71    55
```

Y elaboramos la tabla cruzada en valores absolutos usando la función `table()`:

---

<sup>7</sup> Ver procedimientos de su creación en 5.1.

```
> table(V190REC, V103REC)
      V103REC
V190REC Rural Urbano
Muy pobre  43    11
Pobre      16    17
Medio      10    13
Rico        2     7
Muy rico   0     7
```

Un beneficio que tenemos al usar la función `attach()` es que se añaden los nombres de las variables en las tablas de resultados.

*- Tablas cruzadas en valores absolutos usando capas*

Ejemplo: elaborar una tabla cruzada de la variable V190REC (índice de riqueza actual)<sup>8</sup> y la variable V103REC (lugar de residencia en la niñez), según la variable V102REC (lugar de residencia actual).

Para transformar el vector V102, que corresponde a la variable ENDES V102, lugar de residencia actual, con dos categorías de respuesta: 1=Urbano, 2=Rural, en el factor V102REC con dos niveles, escribiremos lo siguiente:

```
> mater1<-transform(mater1,
+ V102REC=factor(V102, labels=c("Urbano", "Rural")))
> table(mater1$V102REC)
Urbano  Rural
   58    68
```

Debido a que hemos creado una nueva variable, debemos activar nuevamente la función `attach()`

Luego crearemos las tablas cruzadas por capas.

---

<sup>8</sup> Ver procedimientos para su creación en 5.1.

```
> table(V19OREC, V103REC, V102REC)
, , V102REC = Urbano
```

V19OREC	V103REC	
	Rural	Urbano
Muy pobre	1	3
Pobre	11	11
Medio	6	12
Rico	1	6
Muy rico	0	7

```
, , V102REC = Rural
```

V19OREC	V103REC	
	Rural	Urbano
Muy pobre	42	8
Pobre	5	6
Medio	4	1
Rico	1	1
Muy rico	0	0

## 7.1.2. Tablas cruzadas en valores relativos

Para activar las funciones de tablas cruzadas relativas, necesitamos cargar los paquetes<sup>9</sup>: `abind` y `Rcmdr`.

### 7.1.2.1. Tablas cruzadas con porcentajes en columnas

Ejemplo: elaborar una tabla cruzada de la variable nivel socioeconómico con 5 categorías (`V19OREC`)<sup>10</sup> y la variable lugar de residencia en la niñez (`V103REC`), con porcentajes por columnas

```
> colPercents(table(V19OREC, V103REC))
```

V19OREC	V103REC	
	Rural	Urbano
Muy pobre	60.6	20.0
Pobre	22.5	30.9
Medio	14.1	23.6
Rico	2.8	12.7
Muy rico	0.0	12.7
Total	100.0	99.9
Count	71.0	55.0

<sup>9</sup> Ver un ejemplo de cómo cargar el paquete `abind` en 5.5 y del paquete `Rcmdr` en 9.1.

<sup>10</sup> Ver procedimientos para su creación en 5.1.



### 7.1.2.2. Tablas cruzadas con porcentajes en filas

```
> rowPercents(table(V190REC, V103REC))
      V103REC
V190REC Rural Urbano Total Count
Muy pobre 79.6  20.4  100   54
Pobre     48.5  51.5  100   33
Medio     43.5  56.5  100   23
Rico      22.2  77.8  100    9
Muy rico   0.0 100.0  100    7
```

### 7.1.2.3. Tablas cruzadas con porcentajes respecto al total

```
> totPercents(table(V190REC, V103REC))
      Rural Urbano Total
Muy pobre 34.1   8.7 42.9
Pobre     12.7  13.5 26.2
Medio      7.9  10.3 18.3
Rico       1.6   5.6  7.1
Muy rico   0.0   5.6  5.6
Total     56.3  43.7 100.0
```

### 7.1.2.4. Tablas cruzadas con porcentajes en columnas usando capas

```
> colPercents(table(V190REC, V103REC, V102REC))
, , V102REC = Urbano
```

```
      V103REC
V190REC Rural Urbano
Muy pobre  5.3   7.7
Pobre     57.9  28.2
Medio     31.6  30.8
Rico       5.3  15.4
Muy rico   0.0  17.9
Total    100.1 100.0
Count     19.0  39.0
```

```
, , V102REC = Rural
```

```
      V103REC
V190REC Rural Urbano
Muy pobre 80.8  50.0
Pobre     9.6  37.5
Medio      7.7   6.2
Rico       1.9   6.2
Muy rico   0.0   0.0
Total    100.0  99.9
Count     52.0  16.0
```

## 7.2. Test de independencia con Chi cuadrado

Realizamos el test de Chi cuadrado para probar la hipótesis de que el comportamiento de una variable es independiente del comportamiento de la otra. Este test también es llamado de asociación.

Para realizar el test con R, utilizamos la función `chisq.test()`, del paquete básico.

Ejemplo: establecer si el nivel socioeconómico actual de las mujeres entrevistadas<sup>11</sup> (V190REC), varía en función al lugar en donde residieron durante su niñez (V103REC).

```
> chisq.test(table(V190REC, V103REC))

      Pearson's Chi-squared test

data:  table(V190REC, V103REC)
X-squared = 27.5753, df = 4, p-value = 1.521e-05

Warning message:
In chisq.test(table(V190REC, V103REC)) :
  Chi-squared approximation may be incorrect
```

Para realizar el test de Chi cuadrado con valores esperados en la tabla de contingencia, lo convertimos en el objeto `Prueba`.

```
> Prueba<-chisq.test(table(V190REC, V103REC))
Warning message:
In chisq.test(table(V190REC, V103REC)) :
  Chi-squared approximation may be incorrect
> Prueba

      Pearson's Chi-squared test

data:  table(V190REC, V103REC)
X-squared = 27.5753, df = 4, p-value = 1.521e-05

> round(Prueba$expected, 0)
      V103REC
V190REC  Rural Urbano
Muy pobre    30     24
Pobre        19     14
Medio        13     10
Rico         5      4
Muy rico     4      3
```

---

<sup>11</sup> Ver características de las mujeres en la introducción al capítulo 5.

Como podemos observar, R nos advierte del posible error de aproximación a la estimación de Chi cuadrado, lo que está relacionado con la presencia de celdas con valores esperados, inferiores a 5 (ver distribución de la frecuencia esperada para la categoría muy rico).

Por esta razón recodificaremos la variable nivel socioeconómico con cinco categorías de respuesta (V190REC) en la variable NSECTG (nivel socioeconómico con tres categorías de respuesta);<sup>12</sup> y luego realizaremos nuevamente las tablas de contingencia en valores absolutos y relativos, y el test de Chi cuadrado.

```
> table(NSECTG, V103REC)
      V103REC
NSECTG Rural Urbano
Pobre   59     28
Medio   10     13
Rico     2     14
> colPercents(table(NSECTG, V103REC))
      V103REC
NSECTG Rural Urbano
Pobre  83.1   50.9
Medio  14.1   23.6
Rico    2.8   25.5
Total 100.0 100.0
Count  71.0   55.0
> Prueba2<-chisq.test(table(NSECTG, V103REC))
> Prueba2

      Pearson's Chi-squared test

data:  table(NSECTG, V103REC)
X-squared = 18.7072, df = 2, p-value = 8.665e-05
> round(Prueba2$expected, 0)
      V103REC
NSECTG Rural Urbano
Pobre   49     38
Medio   13     10
Rico     9      7
> detach(mater1)
```

En base a los resultados del test, rechazamos la hipótesis nula de independencia entre el nivel socioeconómico actual de las mujeres que tuvieron alguna hermana que falleció en circunstancias relacionadas al embarazo, parto o aborto y el lugar en donde residieron durante su niñez,  $\chi^2=18.71$ ,  $df=2$ ,  $n=126$ ;  $p < .05$ . Es decir, existe una asociación significativa entre las variables.

---

<sup>12</sup> Ver el procedimiento de recategorización en 5.5 y activar nuevamente la función `attach()`.

### 7.3. Pruebas $t$

Las pruebas  $t$  constituyen un conjunto de pruebas de hipótesis sobre las medias de variables cuantitativas. Así tenemos:

- Prueba  $t$  para muestras independientes
- Prueba  $t$  para muestras relacionadas
- Prueba  $t$  para una muestra

Para realizar las pruebas  $t$  usaremos el archivo externo `EUROC.csv`, una versión del archivo `euro.sav`<sup>13</sup> con el cual construiremos el marco de datos `euro1`. Este archivo contiene información sobre indicadores sociodemográficos en 2006 y 2011, para los 28 países considerados parte de la Unión Europea en 2013. La fuente de estos datos es Eurostat, actualizada a julio de 2013 y está disponible en:

<http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home/>

#### 7.3.1. Prueba $t$ para muestras independientes

La prueba  $t$  para muestras independientes mide la diferencia de medias entre dos grupos de una variable.

Ejemplo: queremos saber si la media de la tasa de ocupación masculina de 2006 en la Zona Euro 28 (v4), fue significativamente diferente para los países que ingresaron a dicha zona durante el siglo XX, a la de aquellos que lo hicieron durante el siglo XXI.

```
> euro1 = read.table(file="EUROC.csv", header=TRUE, sep=",")
> euro1
```

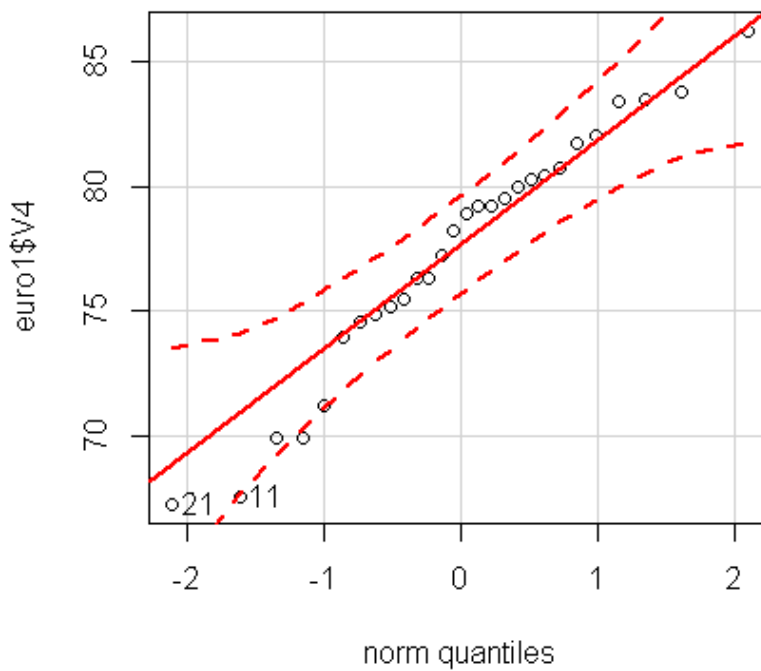
Para realizar esta prueba primero examinaremos si las variables de análisis siguen una distribución normal y presentan igualdad de varianzas.

- *Evaluación de la distribución normal*

Vamos a evaluar si la forma de la distribución de la variable v4 se acerca a la de una distribución normal, primero a través del gráfico de comparación de cuantiles (ver definición y detalles de su construcción en 8.6) y luego a través del test de Shapiro-Wilk, mediante la función `shapiro.test()`.

---

<sup>13</sup> Ver imagen de archivo `euro.sav` en Anexo 2.



La cercanía de los puntos a la recta nos permite observar una distribución de los datos de la variable, cercana a la normal, en el gráfico de comparación de cuantiles.

```
> shapiro.test(euro1$V4)

      Shapiro-Wilk normality test

data:  euro1$V4
W = 0.9617, p-value = 0.3815
```

Los resultados del test de Shapiro-Wilk, confirman la distribución normal de la variable V4, con un nivel de significación de 0,382.

- *Evaluación de la igualdad de varianzas*

Usamos el test de Levene mediante la función `leveneTest()` para evaluar si los grupos de análisis presentan la misma dispersión. Para ello, primero creamos la variable `enterUE`, luego mostramos su distribución y finalmente realizamos la prueba de Levene:

```

> euro1<-transform(euro1,
+ enterUE=factor(V11_recod,labels=c("Siglo XX","Siglo XXI")))
> table(euro1$enterUE)

  Siglo XX Siglo XXI
      15      13
> leveneTest(euro1$V4, euro1$enterUE, center=mean)
Levene's Test for Homogeneity of Variance (center = mean)
  Df F value Pr(>F)
group 1  4.3121 0.04786 *
      26
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

De acuerdo a los resultados, rechazamos la hipótesis nula de igualdad de varianza ( $F = 4.312, p = 0.048$ ).

#### - Prueba $t$

El dato de que la hipótesis sobre igualdad de varianzas ha sido rechazada, será tomado en consideración al momento de escribir las instrucciones para realizar la prueba  $t$  (`var.equal=FALSE`)

```

> t.test(V4~enterUE, alternative='two.sided',
+ conf.level=.95, var.equal=FALSE, data=euro1)

Welch Two Sample t-test

data:  V4 by enterUE
t = 2.4662, df = 18.473, p-value = 0.02364
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.6568645 8.1195458
sample estimates:
mean in group Siglo XX mean in group Siglo XXI
      79.42667          75.03846

```

De acuerdo a los resultados podemos decir que en 2006, existieron diferencias significativas entre las medias de la tasa de ocupación masculina de los países de la Zona Euro 28, según siglo de incorporación a dicha Zona. La media de la tasa de ocupación masculina de los países que ingresaron a la Zona Euro durante el siglo XX fue de 79.43% mientras que la de los países que lo hicieron durante el siglo XXI fue de 75.04% ,  $t=2, df=18.47, n=28, p < .05$ .

### 7.3.2. Prueba $t$ para muestras relacionadas

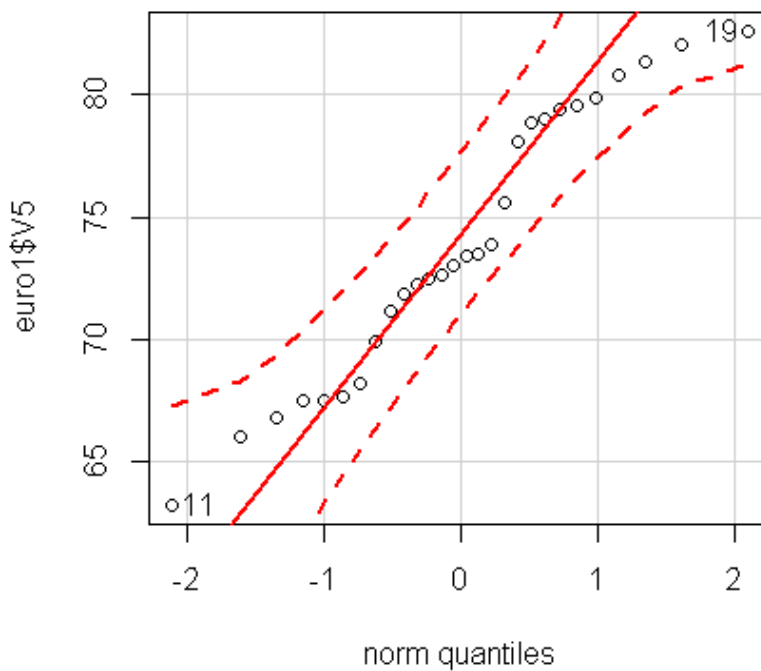
Mide si la diferencia de las medias de dos variables (caracterizada por representar los mismos casos, medidos en tiempos diferentes (antes-después), o por casos emparejados) es diferente de 0.

La hipótesis nula ( $H_0$ ) establece que no hay diferencia de medias, es decir, que la diferencia entre ellas es igual a 0.

Ejemplo: establecer si la diferencia de medias entre la tasa de ocupación masculina en 2011 ( $v_5$ ) y en 2006 ( $v_4$ ) es significativamente diferente de 0 ( $H_1$ ).

- *Evaluación de la distribución normal*

En el punto 7.3.1, usando el gráfico de comparación de cuantiles y el test de Shapiro-Wilk evaluamos que la variable  $v_4$  tenía una distribución normal. Por lo que ahora, solo evaluaremos la forma de la distribución de la variable  $v_5$ .



```
> shapiro.test(euro1$V5)
```

```
Shapiro-Wilk normality test
```

```
data: euro1$V5
```

```
W = 0.9483, p-value = 0.1793
```

Ambos instrumentos confirman la distribución normal de la variable  $v_5$ .

- *Prueba t*

```

> t.test(euro1$V4, euro1$V5, alternative='two.sided',
+ conf.level=.95, paired=TRUE)

      Paired t-test

data:  euro1$V4 and euro1$V5
t = 4.0246, df = 27, p-value = 0.0004146
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.729649 5.327494
sample estimates:
mean of the differences
      3.528571

```

En base a los resultados de la prueba, rechazamos la hipótesis nula de que la diferencia de medias entre la tasa de ocupación masculina de 2011 y la de 2006, en la Zona Euro 28, haya sido igual a 0,  $t = 4,02$ ,  $df = 27$ ,  $p < ,000$ . La diferencia promedio entre ambas medias fue calculada en 3,5, dentro de un intervalo de confianza que va de 1,73 a 5,33.

### 7.3.3. Prueba $t$ para una muestra

Si bien la prueba  $t$  para una muestra no se comprende dentro del análisis estadístico bivariado, sino más bien, dentro del univariado, reseñaremos aquí su uso como parte del conjunto de las pruebas  $t$ .

La prueba  $t$  para una muestra, mide la diferencia entre la media de una variable y un valor hipotético dado.

La hipótesis nula ( $H_0$ ) es que la media de una variable cuantitativa es igual a un valor hipotético dado.

Ejemplo: queremos saber si la media de la tasa de ocupación masculina en 2011(v5) para la Zona Euro 28 fue significativamente **diferente** de 76,16<sup>14</sup> ( $H_1$ ).

De acuerdo con el punto 7.3.2, podemos decir que la variable v5, presenta una distribución normal.

- *Prueba  $t$*

---

<sup>14</sup> Media calculada de la tasa de ocupación masculina durante el período 2000-2007 para la Zona Euro (27)



```
> t.test(euro1$V5, alternative='two.sided', mu=76.16, conf.level=.95)
```

```
One Sample t-test
```

```
data: euro1$V5
t = -2.1885, df = 27, p-value = 0.03746
alternative hypothesis: true mean is not equal to 76.16
95 percent confidence interval:
 71.70506 76.01637
sample estimates:
mean of x
 73.86071
```

En base a los resultados, rechazamos la hipótesis nula. Por lo que podemos decir que la media de la tasa de ocupación masculina para la Zona Euro 28 en 2011 fue significativamente diferente de 76,16,  $t=-2,18$ ,  $df=27$ ,  $p<.05$ . La media se calculó en 73,86 dentro de un intervalo de confianza que va de 71,71 a 76,02.

#### 7.4. Correlación bivariada

Hay varios tipos de coeficiente de correlación. El coeficiente de correlación Pearson describe y mide la fuerza de la relación lineal entre dos variables cuantitativas continuas. Este coeficiente puede asumir valores entre -1 y +1, donde: -1 indica una correlación negativa perfecta y +1 una correlación positiva perfecta; además se considera que el 0 indica la ausencia de una relación lineal entre las variables de análisis.

En R se puede obtener una medida de la correlación entre dos variables con la función `cor()`.

Ejemplo: estimar el coeficiente de correlación entre el porcentaje de nacimientos fuera del matrimonio (V2)<sup>15</sup> y la tasa de ocupación femenina en 2006 (V6).

```
> cor(euro1$V6, euro1$V2)
[1] 0.627377
```

Sin embargo esta simple operación no nos permite conocer ni el nivel de significación de esta estimación ni el intervalo de confianza en el que está comprendida. Para ello debemos utilizar la función `cor.test`, del paquete básico, verificando previamente, que las variables de análisis presenten una distribución normal.

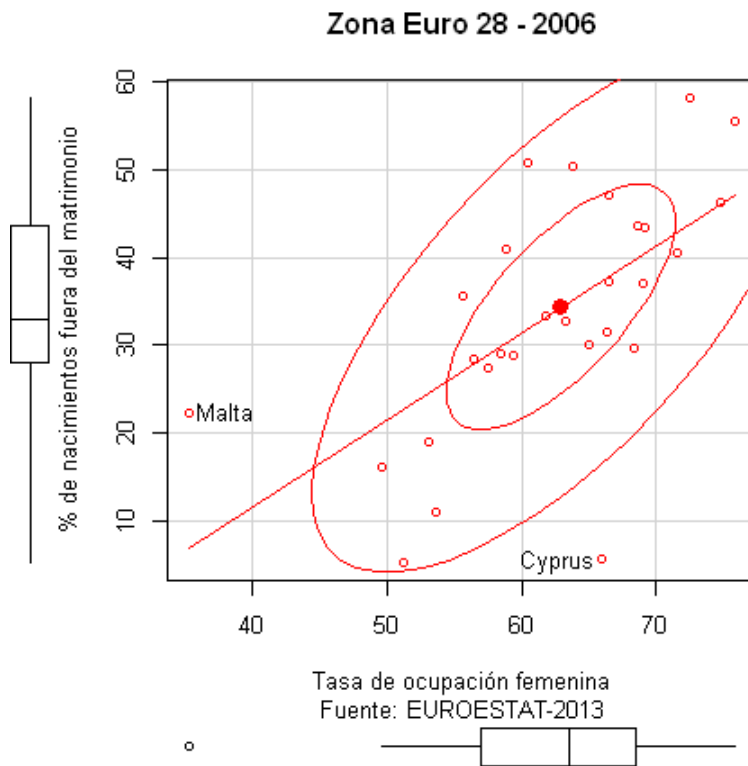
Además es recomendable observar la forma de la relación entre las variables de análisis así como la presencia de valores extremos, utilizando un diagrama de dispersión.

- *Evaluación del sentido de la relación y detección de valores extremos utilizando un diagrama de dispersión*

Ver detalles de su construcción en el punto 8.5.

---

<sup>15</sup> Eurostat define esta variable como: nacimientos, donde el estado civil de la madre al momento del nacimiento fue distinto al de casada.

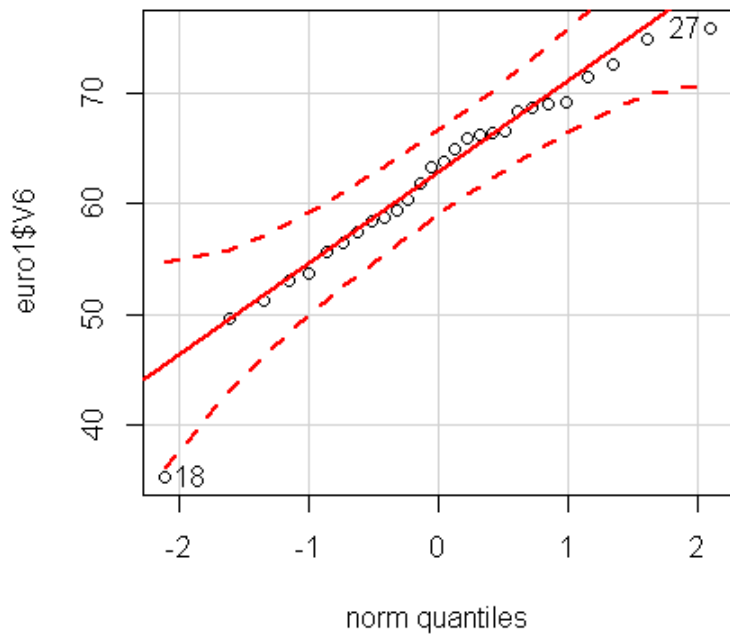
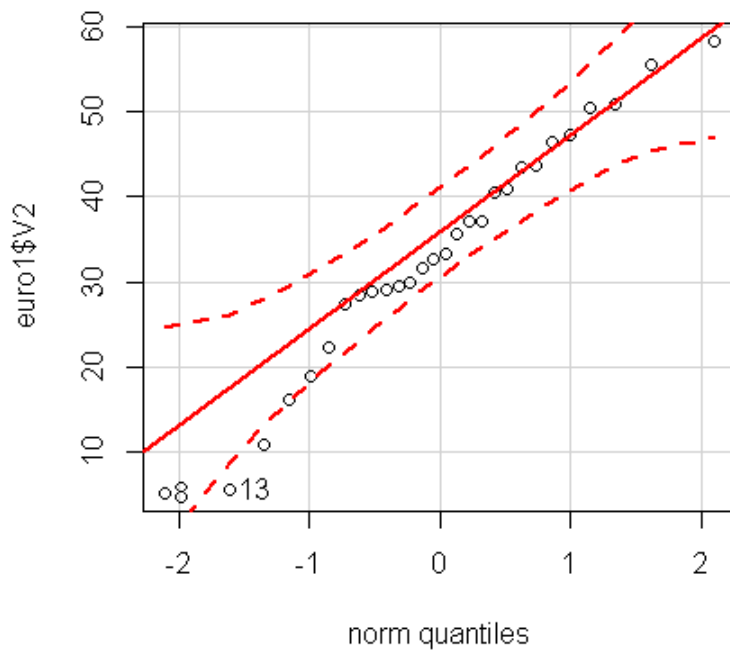


El gráfico nos permite identificar una relación lineal de tipo positiva entre las variables de análisis. Asimismo, podemos observar la presencia de un valor extremo (Malta) y de otro que se aleja de la relación lineal, pero que no constituye un valor extremo (Chipre).

- *Evaluación de la distribución normal*

Realizaremos la evaluación de la distribución normal de las variables de análisis con los gráficos de cuantiles y el Test de Shapiro-Wilk.

A continuación se muestran los gráficos de cuantiles (Ver detalles de su construcción en 8.6):



Como podemos observar, ambos gráficos nos describen una distribución normal de las variable V2 y V6.

A continuación se muestran los resultados del test de Shapiro-Wilk:

```
> shapiro.test(euro1$V2)

      Shapiro-Wilk normality test

data:  euro1$V2
W = 0.9724, p-value = 0.6468
```

```
> shapiro.test(euro1$V6)

      Shapiro-Wilk normality test

data:  euro1$V6
W = 0.947, p-value = 0.1665
```

El Test de normalidad Shapiro-Wilk nos confirma la distribución normal de ambas variables.

- *Prueba de correlación*

```
> cor.test(euro1$V2, euro1$V6, alternative="two.sided", method="pearson")

      Pearson's product-moment correlation

data:  euro1$V2 and euro1$V6
t = 4.1081, df = 26, p-value = 0.0003525
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3320103 0.8107013
sample estimates:
      cor
0.627377
```

Existe una correlación positiva entre las variables analizadas. Lo que indica que cuando la tasa de ocupación femenina aumentó, el porcentaje de nacimientos fuera del matrimonio, también lo hizo, el coeficiente de correlación ( $r$ ) fue igual 0,627,  $df= 26$ ,  $n= 28$ ,  $p < .000$ , y se calculó en un intervalo de confianza de 0,332 y 0,811, lo que indica que la correlación difiere significativamente de 0.

Como hemos podido observar en el gráfico de comparación de cuantiles, Chipre, representa un país cuyos valores de las variables de análisis, se alejan de la relación.

Podríamos retirar del conjunto de países de análisis a Chipre, para observar los cambios en la fuerza de la relación de relación entre las variables. Para ello, escribimos la siguiente instrucción:

```
> euro3 <- euro1[-c(13),]
```

Donde:

`euro3` es el nuevo marco de datos que contendrá todos los casos de `euro1` menos el caso correspondiente a la fila 13 que contiene los datos de Chipre.

`euro1[-c(13),]` indica que conservamos todos los datos de `euro1` menos la fila 13.

Los resultados de la prueba de correlación serían entonces, los siguientes:

```
> cor.test(euro3$V2, euro3$V6, alternative="two.sided", method="pearson")

Pearson's product-moment correlation

data: euro3$V2 and euro3$V6
t = 5.2057, df = 25, p-value = 2.191e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4700197 0.8643380
sample estimates:
      cor
0.7212146
```

## 8. GRÁFICOS

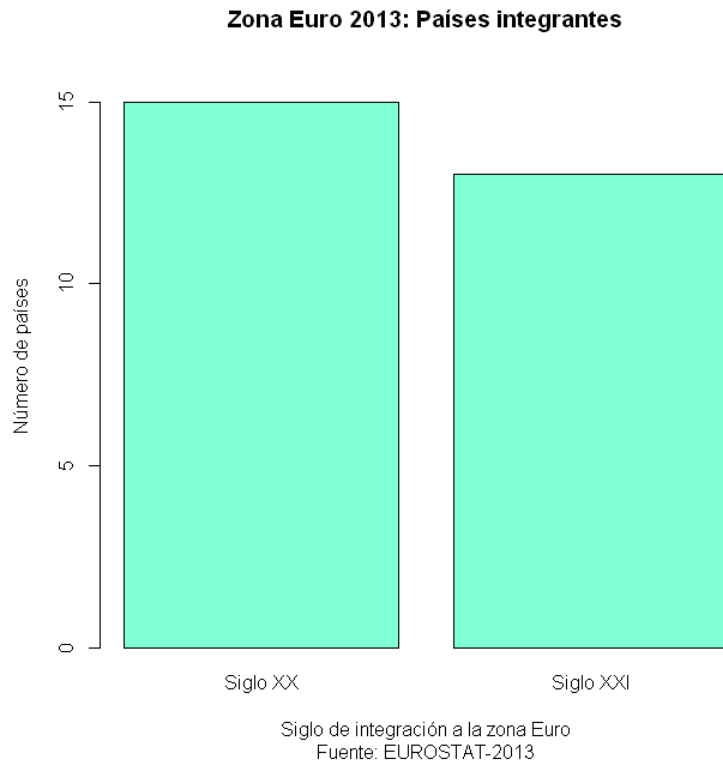
En este capítulo daremos ejemplos de cómo se construyen los gráficos estadísticos de mayor uso en el análisis estadístico básico.

### 8.1. Gráficos de barras

El gráfico de barras nos permite observar la forma en que se distribuyen los datos de las variables cuantitativas. Esta distribución se puede mostrar en valores absolutos o relativos.

Ejemplo: construir un gráfico de barras en valores absolutos para la variable Siglo de entrada a la Zona Euro (enterUE).

```
> barplot(table(euro1$enterUE),
+ main="Zona Euro 2013: Países integrantes",
+ sub="Fuente: EUROSTAT-2013",
+ xlab="Siglo de integración a la zona Euro",
+ ylab="Número de países",
+ ylim=c(0,16),
+ col="aquamarine")
```



## Colores

R maneja 657 colores, a cuyos nombres podemos tener acceso a través de la función `colors()`.

```
> colors()
 [1] "white"           "aliceblue"       "antiquewhite"    "antiquewhite1"
 [5] "antiquewhite2"  "antiquewhite3"  "antiquewhite4"  "aquamarine"
 [9] "aquamarine1"   "aquamarine2"    "aquamarine3"    "aquamarine4"
[13] "azure"          "azure1"         "azure2"         "azure3"
[17] "azure4"        "beige"          "bisque"         "bisque1"
[21] "bisque2"       "bisque3"        "bisque4"        "black"
[25] "blanchedalmond" "blue"           "blue1"          "blue2"
[29] "blue3"         "blue4"          "blueviolet"     "brown"
[33] "brown1"        "brown2"         "brown3"         "brown4"
[37] "burlywood"     "burlywood1"    "burlywood2"    "burlywood3"
[41] "burlywood4"    "cadetblue"     "cadetblue1"    "cadetblue2"
[45] "cadetblue3"    "cadetblue4"    "chartreuse"     "chartreuse1"
[49] "chartreuse2"   "chartreuse3"   "chartreuse4"    "chocolate"

...//

[649] "wheat3"           "wheat4"           "whitesmoke"      "yellow"
[653] "yellow1"          "yellow2"          "yellow3"         "yellow4"
[657] "yellowgreen"
```

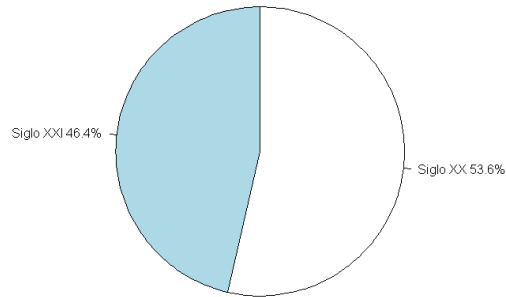
## 8.2. Gráficos de sectores en porcentajes

Al igual que los gráficos de barras, los gráficos de sectores nos permiten observar la forma en que se distribuyen los datos de las variables cualitativas y también se puede mostrar en valores absolutos o relativos.

Ejemplo: elaborar un gráfico de sectores en porcentajes de la variable “Siglo de entrada a la Zona Euro” (`enterUE`).

```
> Cuadro3<-table(euro1$enterUE)
> C3porc<- (round( (Cuadro3/margin.table(Cuadro3)) *100), 1)
> etiquetas<-c("Siglo XX", "Siglo XXI")
> etiquetas<- paste(etiquetas, C3porc)
> etiquetas <- paste(etiquetas, "%", sep="")
> pie(C3porc, labels = etiquetas,
+ clockwise=TRUE,
+ main="Zona Euro 2013: Países integrantes",
+ sub="Fuente: EUROSTAT-2013")
```

Zona Euro 2013: Países integrantes



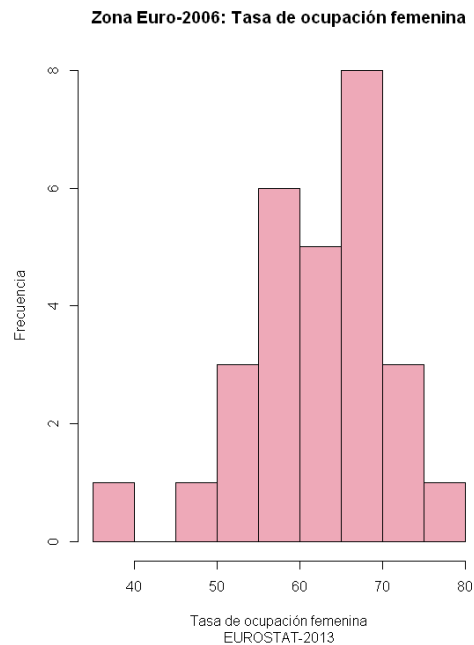
Fuente: EUROSTAT-2013

### 8.3. Histogramas

Estos gráficos nos permiten describir la forma de la distribución de variables cuantitativas.

Ejemplo: elaborar un histograma con frecuencias absolutas, de la tasa de ocupación femenina en la Zona Euro en 2006 (v6), incluidos los países incorporados hasta 2013.

```
> hist(euro1$V6,
+ main="Zona Euro-2006: Tasa de ocupación femenina",
+ sub="EUROSTAT-2013",
+ xlab="Tasa de ocupación femenina",
+ ylab="Frecuencia",
+ col="pink2")
```





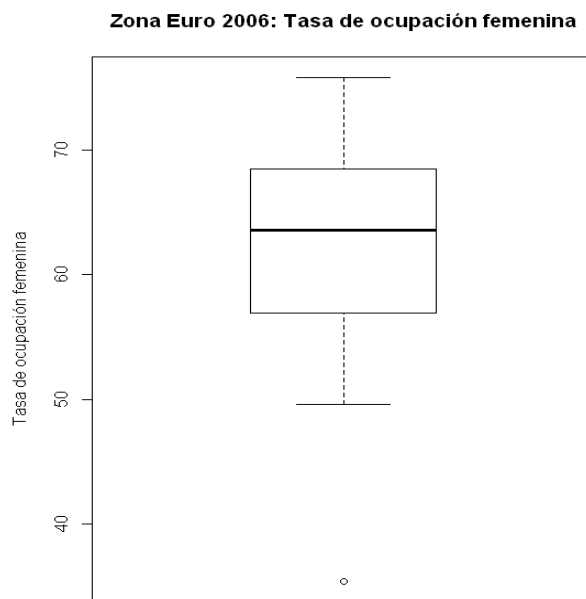
## 8.4. Diagrama de caja

El diagrama de caja nos permite observar la dispersión de los valores de una variable.

- *Diagrama de caja de una variable cuantitativa:*

Ejemplo: elaborar un diagrama de caja de la variable tasa de ocupación femenina en 2006 (V6), incluidos los países incorporados hasta 2013

```
> boxplot(euro1$V6,
+ main="Zona Euro 2006: Tasa de ocupación femenina",
+ sub="Fuente: EuroStat-2013",
+ ylab="Tasa de ocupación femenina")
```



- *Diagrama de caja de una variable cuantitativa en función a una cualitativa*

Ejemplo: elaborar un diagrama de caja de la tasa de ocupación femenina (V6) según siglo de entrada a la Unión Europea (enterUE), diferenciado por colores y con identificación del nombre del país para los casos de valores extremos.

Se puede obtener los mismos resultados con cualquiera de las secuencias de instrucciones siguientes:

```

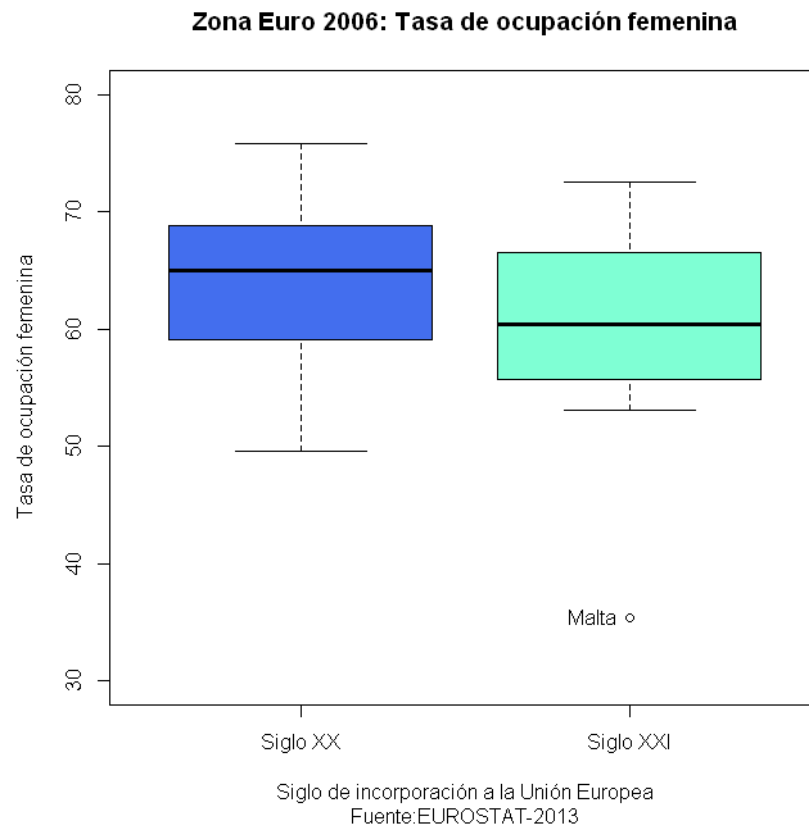
> with(euro1, Boxplot(V6, enterUE,
+ labels=c("Belgium", "Bulgaria",
+ "Czech Republic", "Denmark", "Germany", "Estonia", "Ireland", "Greece", "Spain",
+ "France", "Croatia", "Italy", "Cyprus", "Latvia", "Lithuania", "Luxembourg",
+ "Hungary", "Malta", "Netherlands", "Austria", "Poland", "Portugal",
+ "Romania", "Slovenia", "Slovakia", "Finland", "Sweden", "United Kingdom"),
+ main="Zona Euro 2006: Tasa de ocupación femenina",
+ sub="Fuente:EUROSTAT-2013",
+ xlab="Siglo de incorporación a la Unión Europea",
+ ylab="Tasa de ocupación femenina",
+ ylim=c(30,80),
+ col=c("royalblue2", "aquamarine")))
[1] "Malta"

> Boxplot(euro1$V6, euro1$enterUE,
+ labels=c("Belgium", "Bulgaria",
+ "Czech Republic", "Denmark", "Germany", "Estonia", "Ireland", "Greece", "Spain",
+ "France", "Croatia", "Italy", "Cyprus", "Latvia", "Lithuania", "Luxembourg",
+ "Hungary", "Malta", "Netherlands", "Austria", "Poland", "Portugal",
+ "Romania", "Slovenia", "Slovakia", "Finland", "Sweden", "United Kingdom"),
+ main="Zona Euro 2006: Tasa de ocupación femenina",
+ sub="Fuente:EUROSTAT-2013",
+ xlab="Siglo de incorporación a la Unión Europea",
+ ylab="Tasa de ocupación femenina",
+ ylim=c(30,80),
+ col=c("royalblue2", "aquamarine")))
[1] "Malta"

> attach(euro1)
> Boxplot(V6, enterUE,
+ labels=c("Belgium", "Bulgaria",
+ "Czech Republic", "Denmark", "Germany", "Estonia", "Ireland", "Greece", "Spain",
+ "France", "Croatia", "Italy", "Cyprus", "Latvia", "Lithuania", "Luxembourg",
+ "Hungary", "Malta", "Netherlands", "Austria", "Poland", "Portugal",
+ "Romania", "Slovenia", "Slovakia", "Finland", "Sweden", "United Kingdom"),
+ main="Zona Euro 2006: Tasa de ocupación femenina",
+ sub="Fuente:EUROSTAT-2013",
+ xlab="Siglo de incorporación a la Unión Europea",
+ ylab="Tasa de ocupación femenina",
+ ylim=c(30,80),
+ col=c("royalblue2", "aquamarine")))
[1] "Malta"

```

Observamos que al ejecutar las instrucciones, además del gráfico, aparece una lista con los nombre de los países que presentan valores extremos en la variable cuantitativa que estamos analizando, en este caso: Malta.



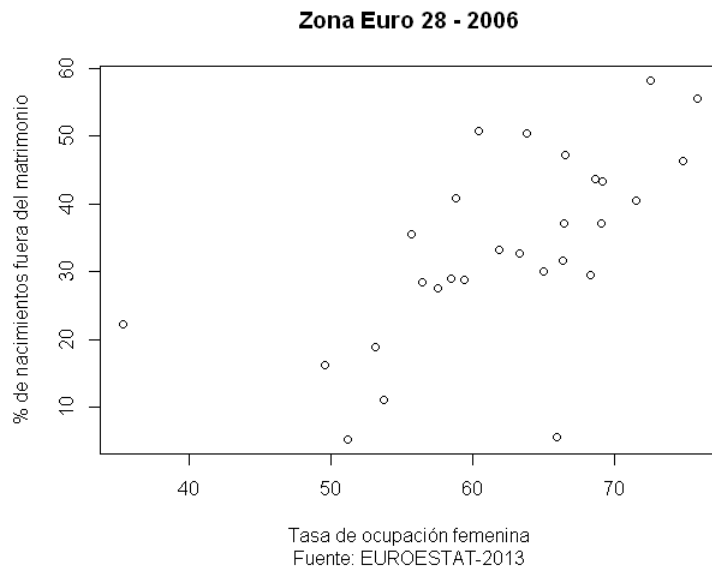
## 8.5. Diagramas de dispersión

Usamos estos gráficos para ilustrar la forma de la relación entre dos variables cuantitativas, así como, para identificar posibles valores extremos.

*- Diagrama de dispersión simple*

Ejemplo: elaborar un diagrama de dispersión entre la tasa de ocupación femenina y el porcentaje de niños nacidos fuera del matrimonio, usando el paquete básico.

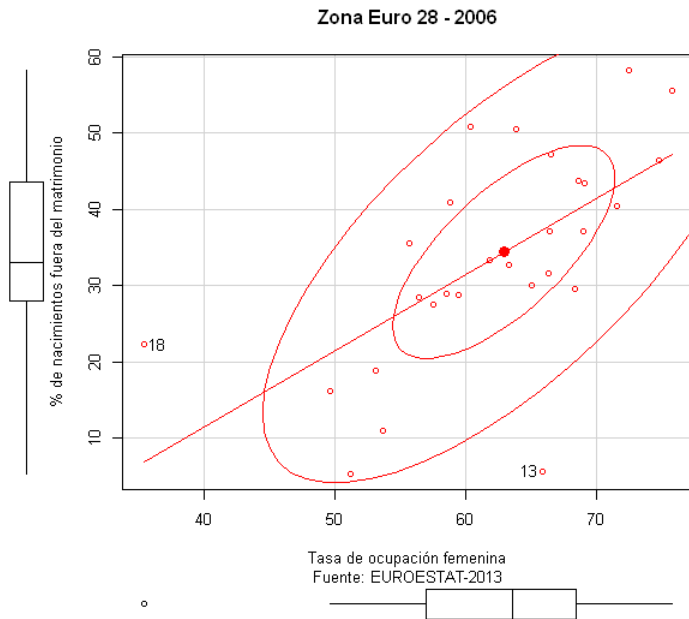
```
> plot(euro1$V6,euro1$V2,
+ main="Zona Euro 28 - 2006",
+ sub="Fuente: EUROESTAT-2013",
+ xlab="Tasa de ocupación femenina",
+ ylab="% de nacimientos fuera del matrimonio")
```



- *Diagrama de dispersión con línea de tendencia, elipse e identificación de algunos puntos con números*

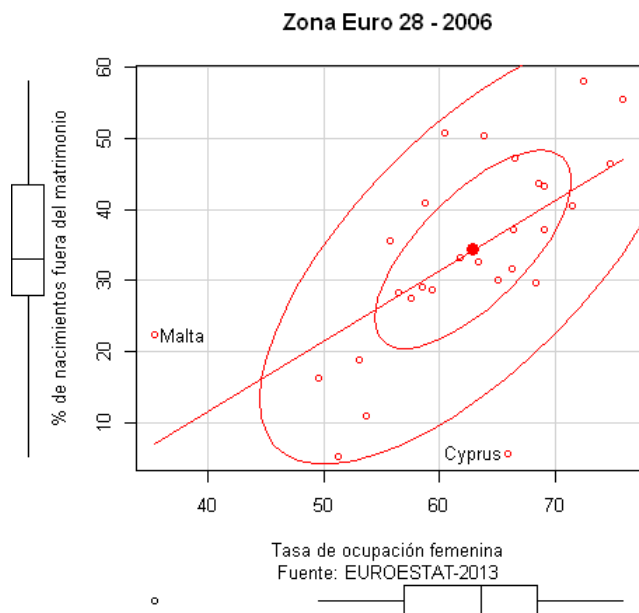
Para elaborar este diagrama y los siguientes es necesario que el paquete Rcmdr esté cargado.

```
> scatterplot(euro1$V6,euro1$V2,
+ ellipse=TRUE,
+ labels,
+ id.n=2,
+ main="Zona Euro 28 - 2006",
+ sub="Fuente: EUROESTAT-2013",
+ xlab="Tasa de ocupación femenina",
+ ylab="% de nacimientos fuera del matrimonio",
+ lwd=1.5,
+ col="red")
13 18
13 18
```



- *Diagrama de dispersión con línea de tendencia, elipse y algunos puntos etiquetados*

```
> scatterplot (euro1$V6,euro1$V2,
+ ellipse=TRUE,
+ labels,
+ id.n=2,
+ labels=c("Belgium","Bulgaria",
+ "Czech Republic","Denmark","Germany","Estonia","Ireland","Greece","Spain",
+ "France","Croatia","Italy","Cyprus","Latvia","Lithuania","Luxembourg",
+ "Hungary","Malta","Netherlands","Austria","Poland","Portugal",
+ "Romania","Slovenia","Slovakia","Finland","Sweden","United Kingdom"),
+ main="Zona Euro 28 - 2006",
+ sub="Fuente: EUROESTAT-2013",
+ xlab="Tasa de ocupación femenina",
+ ylab="% de nacimientos fuera del matrimonio",
+ lwd=1.5,
+ col="red")
Cyprus Malta
  13     18
```

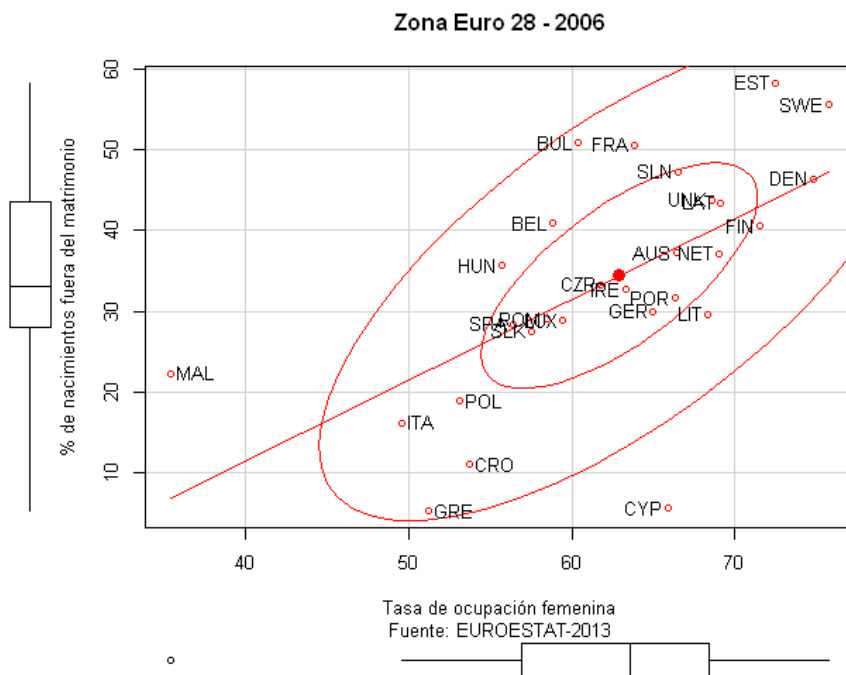


Cabe precisar, que el único valor considerado extremo en el diagrama de dispersión es Malta.

- *Diagrama de dispersión con línea de tendencia, elipse y todos los puntos etiquetados*

```
> scatterplot(euro1$V6,euro1$V2,
+ ellipse=TRUE,
+ labels,
+ id.n=28,
+ labels=c("BEL","BUL",
+ "CZR","DEN","GER","EST","IRE","GRE","SPA",
+ "FRA","CRO","ITA","CYP","LAT","LIT","LUX",
+ "HUN","MAL","NET","AUS","POL","POR",
+ "ROM","SLN","SLK","FIN","SWE","UNK"),
+ main="Zona Euro 28 - 2006",
+ sub="Fuente: EUROESTAT-2013",
+ xlab="Tasa de ocupación femenina",
+ ylab="% de nacimientos fuera del matrimonio",
+ lwd=1.5,
+ col="red")
```

```
BEL BUL CZR DEN GER EST IRE GRE SPA FRA CRO ITA CYP LAT LIT LUX HUN MAL NET AUS
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
POL POR ROM SLN SLK FIN SWE UNK
 21 22 23 24 25 26 27 28
```



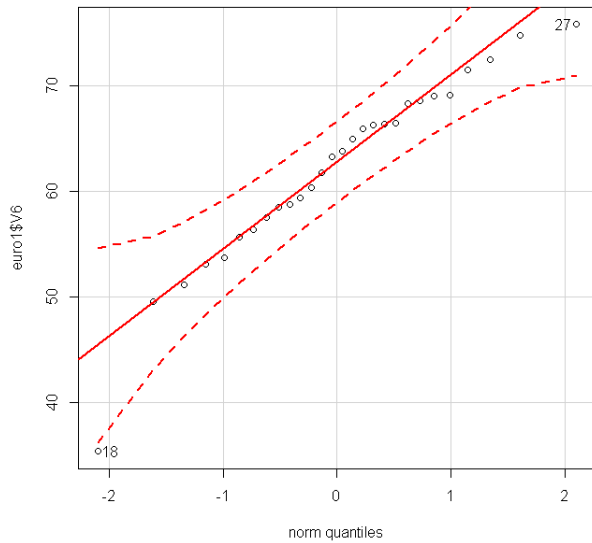
## 8.6. Gráfico de comparación de cuantiles (QQ-plot)

“Sirve para comparar los datos observados a los datos que se debería tener si estos siguieran perfectamente una cierta distribución, a menudo una distribución normal. Los valores observados y los ideales (cuantiles) son comparados sobre un gráfico x-y que muestra una tendencia lineal en caso de normalidad.” (SMCS, 2008)<sup>16</sup>

Ejemplo: elaborar un gráfico de comparación de cuantiles con identificación de algunos valores.

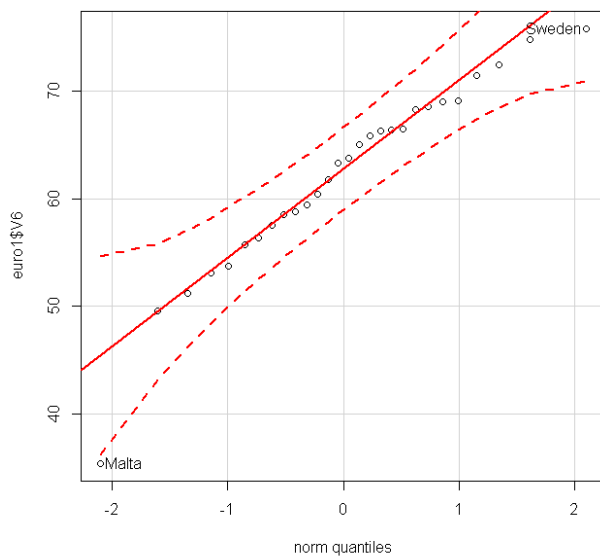
```
> qqPlot(euro1$V6, dist="norm", id.method="y", id.n=2, labels=rownames(euro1))
18 27
 1 28
```

<sup>16</sup> Traducción libre.



- Gráfico de comparación de cuantiles con etiquetado

```
> qqPlot(euro1$V6,
+ dist="norm",
+ id.method="y",
+ id.n=2,
+ labels=c("Belgium", "Bulgaria",
+ "Czech Republic", "Denmark", "Germany", "Estonia", "Ireland", "Greece", "Spain",
+ "France", "Croatia", "Italy", "Cyprus", "Latvia", "Lithuania", "Luxembourg",
+ "Hungary", "Malta", "Netherlands", "Austria", "Poland", "Portugal",
+ "Romania", "Slovenia", "Slovakia", "Finland", "Sweden", "United Kingdom"))
Malta Sweden
1 28
```





## 8.7. Pirámide de edades

Las pirámides de edades son gráficos que nos permiten observar la forma de la estructura de una población según sexo y edad. A continuación mostraremos dos maneras de construir pirámides de edades en R.

- Usando el paquete básico.

Aquí construiremos una pirámide de edades usando una adaptación del programa presentado por Correa y Gonzáles (2002: 74-76), que funciona con el paquete básico de R.

Ejemplo: construir una pirámide de edades, usando el archivo RPIRMD.csv, que contiene información de la población censada en la provincia de Puno-Perú, durante el Censo de Población y Vivienda de 2007, el cual leeremos en R como `puno1`<sup>17</sup>.

```
> puno1 = read.table(file="RPIRMD.csv", header=TRUE, sep=",")
> puno1
```

		edad	Hombre	Mujer
1	De 0 a 4	años	10147	9568
2	De 5 a 9	años	11149	10590
3	De 10 a 14	años	12577	11417
4	De 15 a 19	años	11487	11196
5	De 20 a 24	años	10468	10686
6	De 25 a 29	años	9281	9949
7	De 30 a 34	años	7908	8658
8	De 35 a 39	años	6950	8123
9	De 40 a 44	años	6434	6952
10	De 45 a 49	años	5444	6080
11	De 50 a 54	años	4968	5294
12	De 55 a 59	años	4246	4499
13	De 60 a 64	años	3375	3642
14	De 65 a 69	años	2558	2763
15	De 70 a 74	años	2277	2458
16	De 75 a 79	años	1814	1906
17	De 80 a 84	años	1015	1150
18	De 85 a 89	años	647	694
19	De 90 a 94	años	188	218
20	De 95 a 99	años	188	272

---

<sup>17</sup> Ver contenido en Anexo 2.

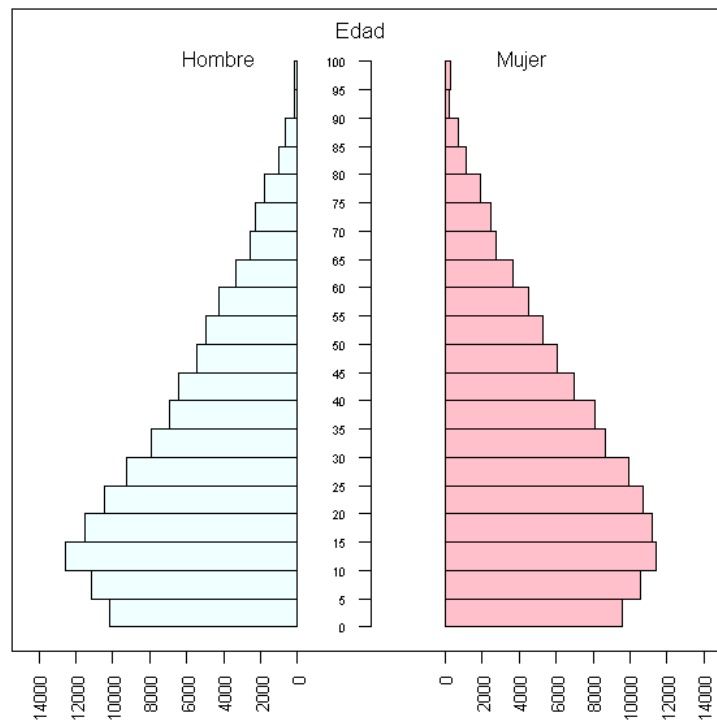
```

> attach(puno1)
> piramide<-function(Hombre,Mujer,amplitud,escalax,
+ edadmax) {
+ max1<-max(c(Hombre,Mujer))
+ min.x<--(max1%/%escalax+1)*escalax
+ max.x<-(max1%/%escalax+1)*escalax
+ n<-length(Mujer)
+ plot(0,0,type="n",xaxt='n',yaxt='n',ylim=c(0,edadmax+5),
+ xlim=c(min.x-2*escalax,max.x+2*escalax),xlab="",ylab="")
+ ejex1<-seq(2*escalax,max.x+2*escalax,,by=escalax)
+ ejex2<--ejex1[order(-ejex1)]
+ ejex<-c(ejex2,ejex1)
+ ejexe1<-seq(0,max.x,by=escalax)
+ ejexe2<--ejexe1[order(-ejexe1)]
+ ejexe<-c(ejexe2,ejexe1)
+ axis(1,at=ejex,labels=as.character(abs(ejexe)),
+ cex.axis=0.8,las=2)
+ eje1<-c(seq(0,edadmax,by=amplitud))
+ axis(2,at=eje1,labels=as.character(eje1),
+ cex.axis=0.6,las=2,pos=0)
+ for(i in 1:n){
+ x1<-2*escalax
+ x2<-Mujer[i]+2*escalax
+ x3<--Hombre[i]-2*escalax
+ y1<-(i-1)*amplitud
+ y2<-y1+amplitud
+ rect(x1,y1,x2,y2,col='pink')
+ rect(-x1,y1,x3,y2,col='azure')
+ }
+ x.l1<--max1/16-2.5*escalax
+ x.l2<-max1/16+2*escalax
+ title(main=paste("Puno 2007: Pirámide de edades",sep="\n"))
+ legend(x.l1,edadmax+5,"Hombre",bty="n",xjust=1)
+ legend(x.l2,edadmax+5,"Mujer",bty="n")
+ legend(0,edadmax+10,"Edad",bty='n',xjust=0.5,adj=c(0.5,0.5)) }
> amplitud<-5
> escalax<-2000
> edadmax<-100
> fig<-piramide(Hombre,Mujer,amplitud,escalax,
+ edadmax)

> detach(puno1)

```

### Puno 2007: Pirámide de edades



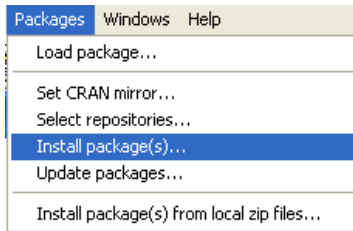
- Usando la función `pyramid()` de *Minato Nakazawa*

Para construir la pirámide de edades de la provincia de Puno-Perú en 2007, esta vez usaremos la función `pyramid()` escrita por Nakazawa (2013: 2):

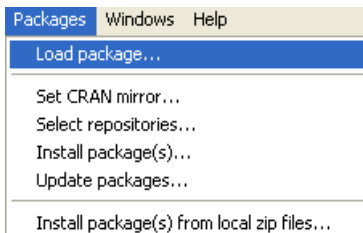
```
pyramid(data, Laxis=NULL, Raxis=NULL,
AxisFM="g", AxisBM="", AxisBI=3, Cgap=0.3, Cstep=1,
Csize=1,
Llab="Males", Rlab="Females", Clab="Ages", GL=TRUE,
Cadj=-0.03,
Lcol="Cyan", Rcol="Pink", Ldens=-1, Rdens=-1, main="",
...)
```

Para usar esta función, primero debemos instalar el paquete `pyramid`.

Para ello seguimos la secuencia que se muestra en el gráfico:



Luego cargamos el paquete.



Además las variables del archivo donde se encuentran los datos deben estar dispuestas organizadas en el siguiente orden: variable Hombres, variable Mujeres, variable Edades, como se muestra cuando leemos en R como `puno2`, el archivo `pyramideNKZW.csv`.

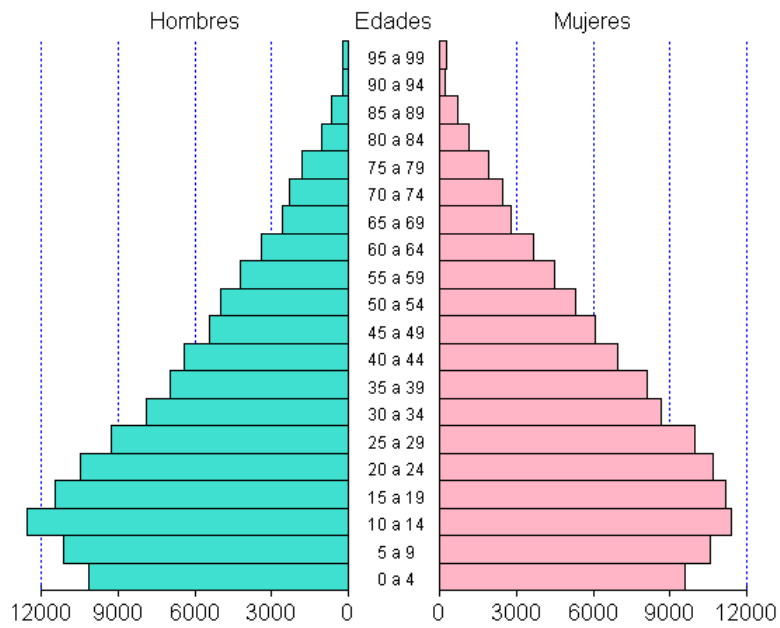
```
> puno2 = read.table(file="piramideNKZW.csv", header=TRUE, sep=",")
> puno2
```

	Hombres	Mujeres	Edad
1	10147	9568	0 a 4
2	11149	10590	5 a 9
3	12577	11417	10 a 14
4	11487	11196	15 a 19
5	10468	10686	20 a 24
6	9281	9949	25 a 29
7	7908	8658	30 a 34
8	6950	8123	35 a 39
9	6434	6952	40 a 44
10	5444	6080	45 a 49
11	4968	5294	50 a 54
12	4246	4499	55 a 59
13	3375	3642	60 a 64
14	2558	2763	65 a 69
15	2277	2458	70 a 74
16	1814	1906	75 a 79
17	1015	1150	80 a 84
18	647	694	85 a 89
19	188	218	90 a 94
20	188	272	95 a 99

Luego aplicamos la función `pyramid()`

```
> pyramid(puno2,
+ Laxis=seq(0,13000,by=3000), Raxis=NULL,
+ AxisFM="d", AxisBM="", AxisBI=3,
+ Cgap=0.3, Cstep=1, Csize=0.75,
+ Llab="Hombres", Rlab="Mujeres", Clab="Edades",
+ GL=TRUE, Cadj=-0.015,
+ Lcol="Turquoise", Rcol="Pink1",
+ Ldens=-1, Rdens=-1,
+ main="Puno 2007: Pirámide de edades",
+ sub="Fuente: INEI-CPV-2007")
```

### Puno 2007: Pirámide de edades

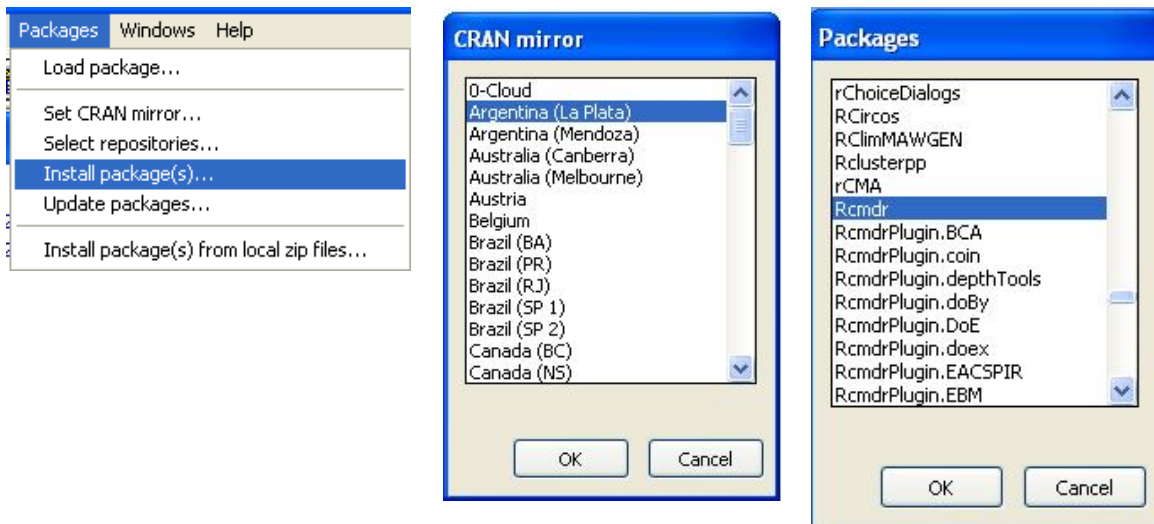


Fuente: INEI-CPV-2007

## 9. CARACTERÍSTICAS Y FUNCIONES BÁSICA DE R COMMANDER

### 9.1. Instalar R Commander

Para instalar R Commander, desde el menú Packages de R, primero seleccionamos Install package(s), luego seleccionamos un CRAN mirror y finalmente el paquete Rcmdr.



Entonces se mostrará el siguiente mensaje en la Consola y el cursor quedará en espera.

```
R R Console
> utils:::menuInstallPkgs()
trying URL 'http://mirror.fcaglp.unlp.edu.ar/CRAN/bin/windows/contrib/3.0/Rcmdr$
Content type 'application/zip' length 3956226 bytes (3.8 Mb)
opened URL
downloaded 3.8 Mb

package 'Rcmdr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Documents and Settings\Administrador\Configuración local\Temp\Rtmpie$
> |
```

Ahí escribimos `library(Rcmdr)`. Al digitar esta instrucción por primera vez, aparecerá un mensaje que nos advierte que **Rcmdr** necesita instalar otros paquetes. Para ello, hacemos clic sobre el botón “Sí”.

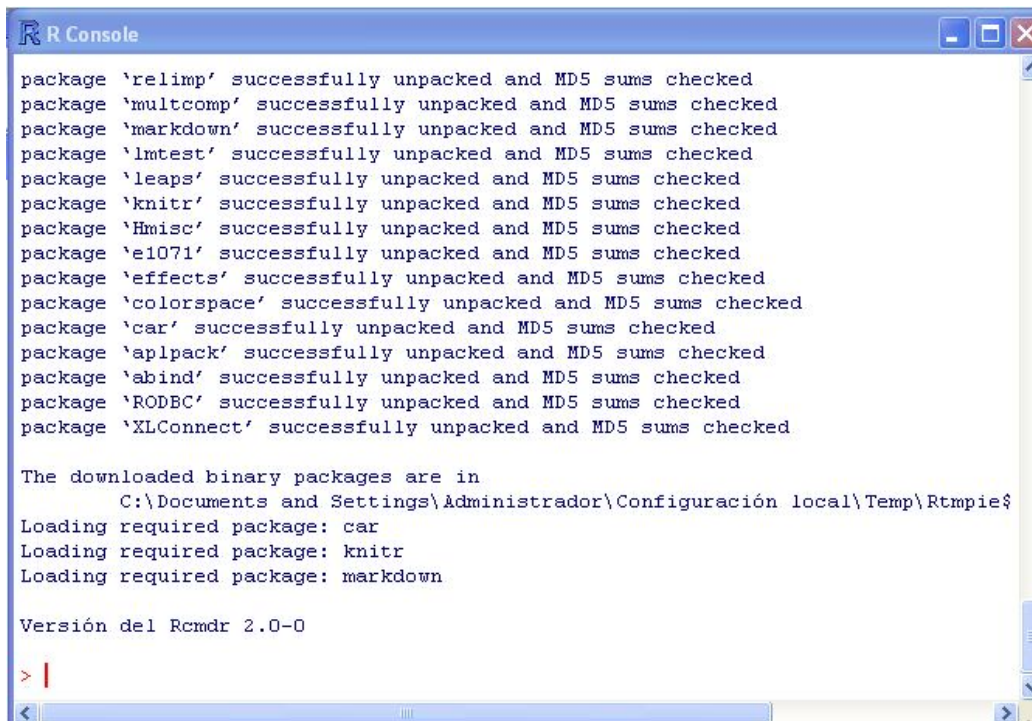
```
> library(Rcmdr)
```



Entonces aparecerá la ventana de instalación de los paquetes restantes. Hacemos clic sobre OK.



R Commander nos notificará el éxito de la instalación a través de la Consola.



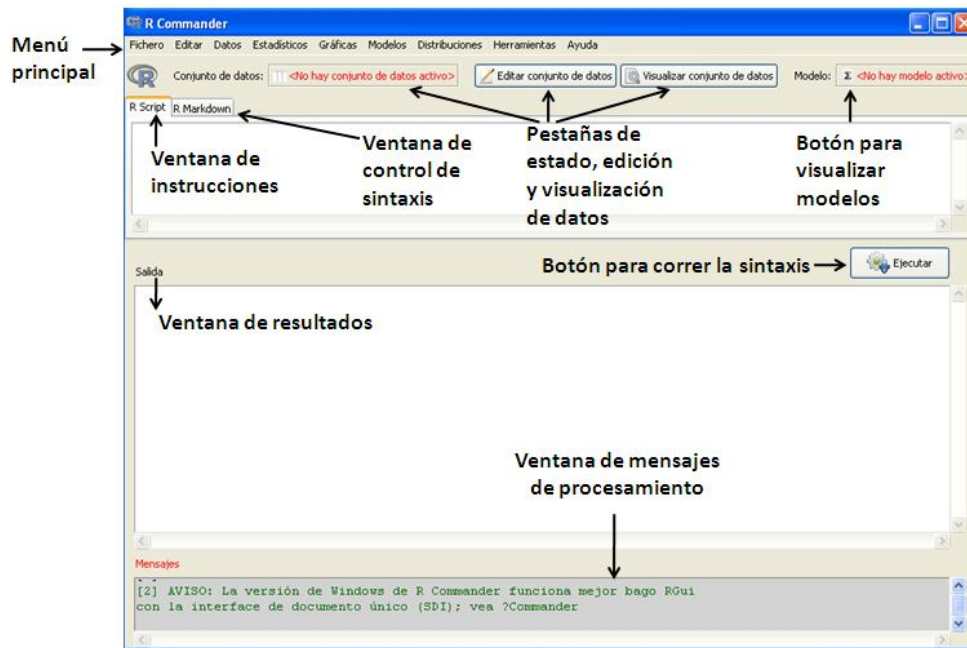
Y se abrirá la ventana de R Commander.

En las siguientes sesiones podemos entrar a R Commander digitando `library(Rcmdr)` en la Consola de R, o podemos cargarlo desde el menú Packages → Load package...



## 9.2. Descripción del ambiente de trabajo de R Commander

En el siguiente gráfico se muestran las ventanas de trabajo principales en R Commander.



### - *El menú principal*

Contiene los menús: Fichero, Editar, Datos, Estadísticos, Gráficas, Modelos, Distribuciones, Herramientas y Ayuda. Desde estos menús, enviaremos instrucciones a R para activar un conjunto de datos, realizar transformaciones de los mismos o de sus variables, realizar análisis estadísticos y gráficos con ellos, guardar los resultados o solicitar ayuda para algún procedimiento de análisis de datos.

### - *Barra de herramientas*

Contiene las siguientes pestañas: *pestañas de estado*, que nos permite identificar la base de datos activa; *pestaña de edición* que nos permite editar las bases de datos que hayan sido cargadas a R Commander; *pestaña de visualización*, que nos permite obtener una imagen de la base de datos activa; y *pestaña de modelos*, que cuando trabajamos con modelos estadísticos, nos muestra cuál es el que está activo.

### - *La ventana R Script*

Es aquella en donde se escriben automáticamente las instrucciones que enviamos desde los menús. También podemos escribir manualmente, instrucciones en esta ventana.

### - *El botón Ejecutar*

Nos permite ejecutar manualmente una o más instrucciones seleccionadas, en la ventana Script.

- *La ventana de resultados*

En ella también se escriben automáticamente las instrucciones enviadas desde los menús, además de los resultados de dichas instrucciones.

- *La ventana R Markdown*

Aquí se convierten las instrucciones dadas a R Commander en documentos R Markdown<sup>18</sup>. Cuando esta ventana está activa, al hacer clic sobre el botón “Generar informe HTML”, se compila el documento R Markdown en formato HTML permitiendo su publicación en una página web.

- *La ventana de mensajes*

Aquí se muestran mensajes en relación a las instrucciones para el análisis de datos que hemos dado a R Commander. Los mensajes en azul indican que las instrucciones se procesaron sin inconvenientes. Los mensajes que aparecen en rojo, indican que hay error en las instrucciones dadas, y que por lo tanto no se ha podido reportar los resultados; si esto ocurre, debemos corregir la instrucción. Los mensajes en verde<sup>19</sup> indican que el resultado se obtuvo, pero que hay detalles que deben tenerse en cuenta para una correcta interpretación de los resultados.

- *Ubicación de la sesión de trabajo*

Para poder ubicarnos en el directorio en el que vamos a trabajar seleccionamos:

Fichero → Cambiar directorio de trabajo →

Y en la siguiente ventana escogemos el directorio y la carpeta en los que depositaremos todos los objetos que desarrollemos en R:

---

<sup>18</sup> Markdown es una herramienta para convertir textos planos en formato HTML para su publicación en un sitio web. (John Gruber, 2004)

<sup>19</sup> Por ejemplo al instalar Rcmdr, vemos que en la ventana de mensajes aparece uno en letras de color verde, que nos sugiere que: “La versión de R Commander funciona mejor ba[j]o RGui con la interface de documento único (SDI); vea ?Commander”. Esto se puede hacer desplegando el menú Edit → GUI preferences... y activando SDI, en la opción Single or multiple windows de la ventana RGui Configuration Editor. Para realizar los ejemplos de este manual, no se ha realizado este cambio.



### 9.3. Tratamiento de archivos con R Commander

#### 9.3.1. Trabajar con archivos externos

En R Commander podemos trabajar con archivos externos de dos formas: cargando archivos o importándolos.

##### 9.3.1.1. Cargando archivos

Esta modalidad de trabajo nos permite hacer modificaciones directas en el conjunto de datos inserto en el archivo, a través de la ventana de edición.

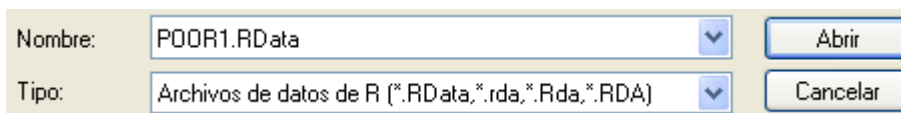
- *Archivos que pueden ser cargados*

Archivos con extensión \*.RData, \*.rda, \*.Rda, \*.RDA.

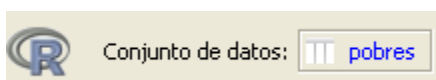
- *Procedimiento para cargar archivos*

Datos → Cargar conjunto de datos

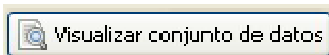
Ejemplo: cargar el conjunto de datos `pobres` que se encuentra en el archivo `POOR1.RData`, creado en 4.5.



Al finalizar, podemos observar que el conjunto de datos “`pobres`”, inserto en el archivo `POOR1.RData`, está activo.



Y podemos obtener una imagen de su contenido, haciendo clic sobre el botón “Visualizar conjunto de datos”.



	CASEID	MMIDX	MM1	MM2	MM3	MM4	MM5	MM6	MM7	MM8	MM9	MM10	MM11	MM12	MM13
1	000804001 03	7	2	0	NA	NA	NA	14	25	NA	2	NA	1	NA	2
3	001202101 02	1	2	0	NA	NA	NA	NA	38	NA	2	NA	3	NA	3
4	001607501 03	3	2	0	NA	NA	NA	11	14	NA	2	NA	1	NA	3
5	002106101 02	2	2	0	NA	NA	NA	36	18	NA	2	NA	1	NA	3

#### - Edición de archivos cargados

Como ya señalamos, al cargar un archivo externo, es posible editar el conjunto de datos inserto en él. Para ello hacemos clic sobre el botón “Editar el conjunto de datos” e introducimos las modificaciones consideradas pertinentes.



	row.names	CASEID	MMIDX	MM1	MM2	MM3	MM4	MM5	MM6	MM7	MM8	MM9
1	1	000804001 03	7	2	0				14	25		2
2	3	001202101 02	1	2	0				NA	38		2
3	4	001607501 03	3	2	0				11	14		2
4	5	002106101 02	2	2	0				36	18		2
5	6	005700501 02	3	2	0				32	17		2

### 9.3.1.2. Importando archivos

Bajo esta modalidad de trabajo, **no** podemos entrar a la ventana de edición para modificar los datos.

#### - Archivos que pueden ser importados

Podemos importar a R, archivos con formato de texto, desde un directorio o desde una dirección URL, así como archivos de los siguientes programas: SPSS, STATA, SAS, Minitab, Excel, Access, DBase.

#### - Procedimiento de importación de archivos

En esta parte trabajaremos con el archivo `RPUNOE1.sav`, construido en base a la fusión de dos archivos de la Encuesta Demográfica y de Salud Familiar-Endes, Perú, 2012: `RECH0.sav` y `RECH1.sav`, integrantes del módulo: 323-Modulo64, y disponibles en: <http://inei.inei.gob.pe/microdatos/>.

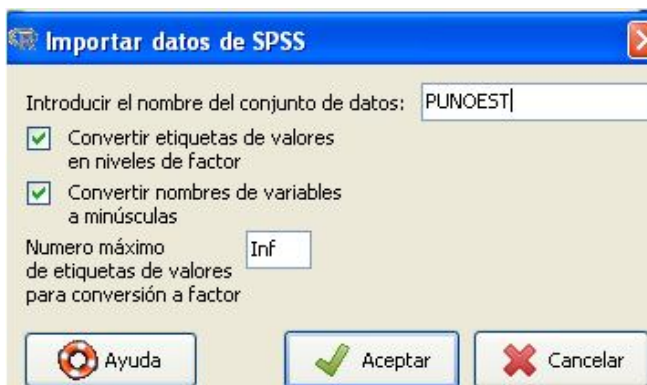
Este archivo contiene información sobre las características educativas de 550 niños de 12 a 16 años de edad del departamento de Puno-Perú, 2012.

Para importar el archivo externo SPSS `RPUNOE1.sav` a R, donde lo leeremos como el conjunto de datos `PUNOEST` seguiremos los siguientes pasos:

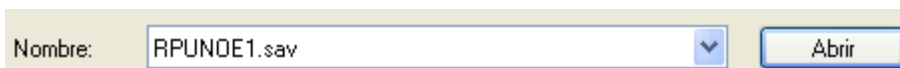
Primero desplegamos el menú:

Datos → Importar datos → desde datos SPSS

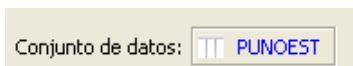
En la ventana “Importar datos de SPSS” escribimos como nombre del conjunto de datos: `PUNOEST`, y dejamos activadas las opciones “Convertir etiquetas de valores en niveles de factor” y “Convertir nombres de variables a minúsculas”, luego hacemos clic sobre el botón “Aceptar”.



Luego se abrirá una ventana emergente en la que seleccionaremos y abriremos el archivo `RPUNOE1.sav`.



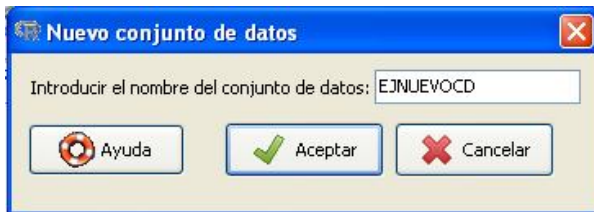
Al finalizar, la pestaña “Conjunto de datos”, nos indicará que `PUNOEST` está activo:



### 9.3.2. Crear un conjunto de datos

Para crear un conjunto de datos desde el editor de datos de R Commander, desplegamos el menú:

Datos → Nuevo conjunto de datos... e introducimos el nombre del nuevo conjunto de datos:



Al aceptar, se abrirá la ventana del Editor de Datos con un encabezado que contiene los nombres de las variables autogeneradas en columnas; en el margen izquierdo los números de las filas; y con casilleros en blanco, al centro, listos para escribir sobre ellos nuestros propios datos.

	var1	var2	var3	var4	var5	var6
1						
2						
3						
4						
5						
6						

Allí podemos escribir por ejemplo:

	var1	var2	var3	var4	var5	var6
1	S001	15	16	Urbana	hombre	
2	S002	16	16	Rural	mujer	
3	S003	13	17	Rural	mujer	
4	S004	14	12	Urbana	hombre	
5	S005	15	14	Urbana	hombre	
6						
7						

Al finalizar, en la ventana aparecerá el siguiente mensaje:

```
[12] NOTA: El conjunto de datos EJNIUEVOCD tiene 5 filas y 5 columnas.
```

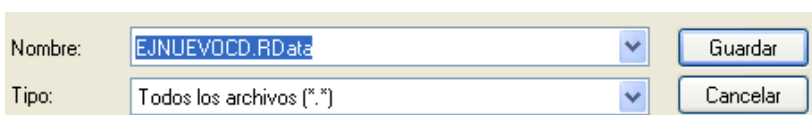
Y el conjunto de datos se mostrará como activo

Conjunto de datos: EJNIUEVOCD

Para guardarlo, desplegamos el menú:

Datos → Conjunto de datos activos → Guardar el conjunto de datos activos...

Y guardamos este conjunto de datos como EJNIUEVOCD.RData



### 9.3.3. Cambiar de conjunto de datos activo

Al igual que en R, en R Commander podemos trabajar con varios conjuntos de datos a la vez. Como ya dijimos, en 9.2, la base de datos activa estará visible en la *pestaña de estado* “Conjunto de datos:”. Si queremos cambiar de conjunto de datos, haremos clic sobre el nombre del activo (en este caso PUNOEST), para abrir la ventana “Seleccionar conjunto de d...” que contiene la lista de los conjuntos de datos disponibles en la sesión de trabajo.



### 9.3.4. Trabajar con solo un conjunto de variables de los archivos

Para crear un nuevo conjunto de datos, que solo contenga un conjunto de variables de otro conjunto de datos activo, desplegamos el menú:

Datos → Conjunto de datos activo → Filtrar el conjunto de datos activo

En la ventana “Filtrar el conjunto de datos” desactivamos el recuadro “Incluir todas las variables”, y luego seleccionamos las variables con las que deseamos trabajar en un nuevo conjunto de datos (presionamos la tecla CTRL, para facilitar la selección si estas no se ubican de manera consecutiva). Escribimos el nombre del nuevo conjunto de datos (en el ejemplo PUNOEST3) y hacemos clic sobre el botón “Aceptar”.



## 9.4. Tratamiento de variables con R Commander

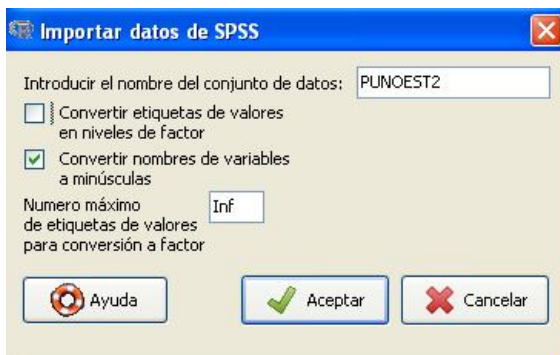
### 9.4.1. Convertir vectores en factores o crear variables cualitativas

En los archivos de bases de datos como los de de SPSS (\*.sav), las variables cualitativas tienen por lo general códigos numéricos y etiquetas alfanuméricas

Ejemplo: La variable hv025 (lugar de residencia) del archivo RPUNO01 tiene los códigos de respuestas 1 y 2, y sus etiquetas correspondientes son 1 = “Urbano” y 2 = “Rural”.

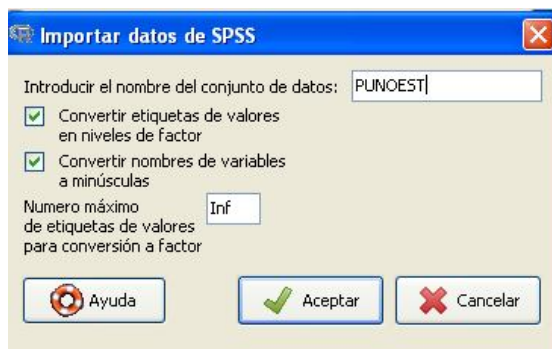
Al importar este tipo de archivos, R Commander puede leerlos de dos maneras.

Si al momento de importar el archivo no activamos la opción “Convertir etiquetas de valores en niveles de factor”, R Commander leerá los códigos numéricos de estas variables como vectores y los tratará como variables numéricas, como se puede observar a continuación:



hv025	hv025
1	Min. :1.000
1	1st Qu.:1.000
1	Median :2.000
1	Mean :1.655
2	3rd Qu.:2.000
2	Max. :2.000
2	

Por el contrario, si activamos la opción “Convertir etiquetas de valores en niveles de factor”, R Commander convertirá, de manera automática, todas las variables etiquetadas que contenga el archivo SPSS en factores. Veamos el resultado de esta elección con la variable hv025.



hv025	hv025
Urbana	Urbana:190
Urbana	Rural :360
Urbana	
Rural	
Rural	
Rural	

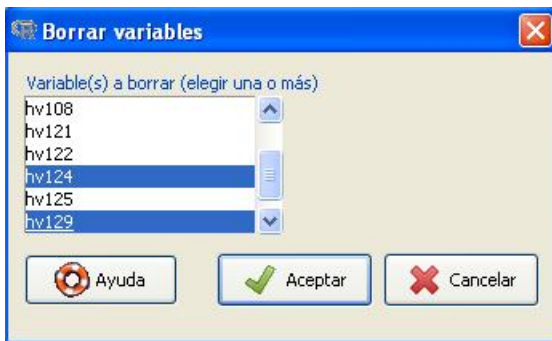


### 9.4.2. Eliminar variables

Una vez activado el conjunto de datos, podemos eliminar variables del mismo, desplegando el menú:

Datos → Modificar variables del conjunto de datos activo → Eliminar variables del conjunto de datos

Luego seleccionamos una o más variables, y hacemos clic sobre el botón Aceptar.

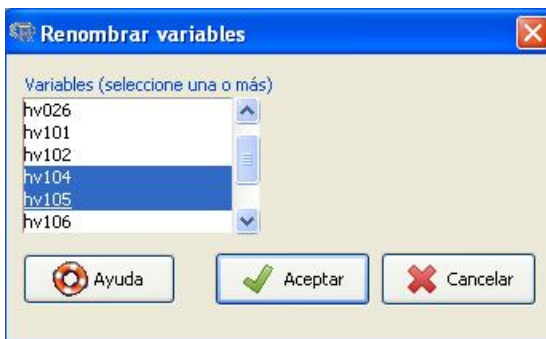


### 9.4.3. Renombrar variables

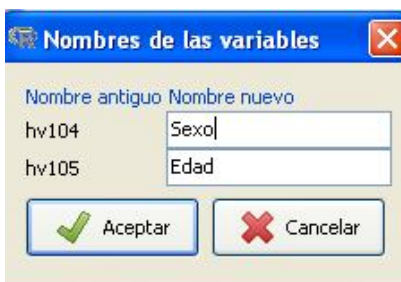
Para renombrar variables desplegamos el menú:

Datos → Modificar variables del conjunto de datos activo → Renombrar variables.

En la ventana emergente seleccionamos una o más variables.



Y escribimos los nuevos nombres de la o las variable(s) seleccionada(s).



A continuación podemos observar que las variables hv104 y hv105 han sido renombradas como Sexo y Edad.

Antes		Después	
hv104	hv105	Sexo	Edad
Hombre	16	Hombre	16
Mujer	15	Mujer	15
Hombre	14	Hombre	14
Hombre	15	Hombre	15
Mujer	12	Mujer	12

#### 9.4.4. Crear nuevas variables

Para crear nuevas variables desplegamos el menú:

Datos → Modificar variables del conjunto de datos activo → Calcular una nueva variable y luego seleccionamos una variable cuantitativa.

Ejemplo: crear una variable que exprese a la variable número de años de instrucción (hv108) en logaritmos.

hv108	hv108LN
11	2.3978953
10	2.3025851
8	2.0794415
9	2.1972246
7	1.9459101
7	1.9459101

En el conjunto de datos observaremos a la variable hv108, y además, a la variable hv108LN, que expresa los valores de la variables hv108 en logaritmos.

Así podemos generar otras variables con otras expresiones de cálculo como elevar al cuadrado, calcular la raíz cuadrada...

#### - Tipificar variables

Para crear valores estandarizados o tipificados de una variable desplegamos el menú:

Datos → Modificar variables del conjunto de datos activo → Tipificar variables...



```
> .Z <- scale(euro[,c("v2")])
> euro$Z.v2 <- .Z[,1]
```



v2	Z.v2
40.9	0.53389016
50.8	1.24540146
33.3	-0.01232054
46.4	0.92917422
30.0	-0.24949098

#### 9.4.5. Recodificar

Para recodificar variables primero verificaremos los códigos y categorías correspondientes a la variable a recodificar.

Ejemplo recodificar la variable “lugar de residencia-4 categorías” (hv026) del conjunto de datos PUNOEST, en la variable “lugar de residencia-3 categorías”, en el orden “campo-pueblo-ciudad mediana (hv026for).

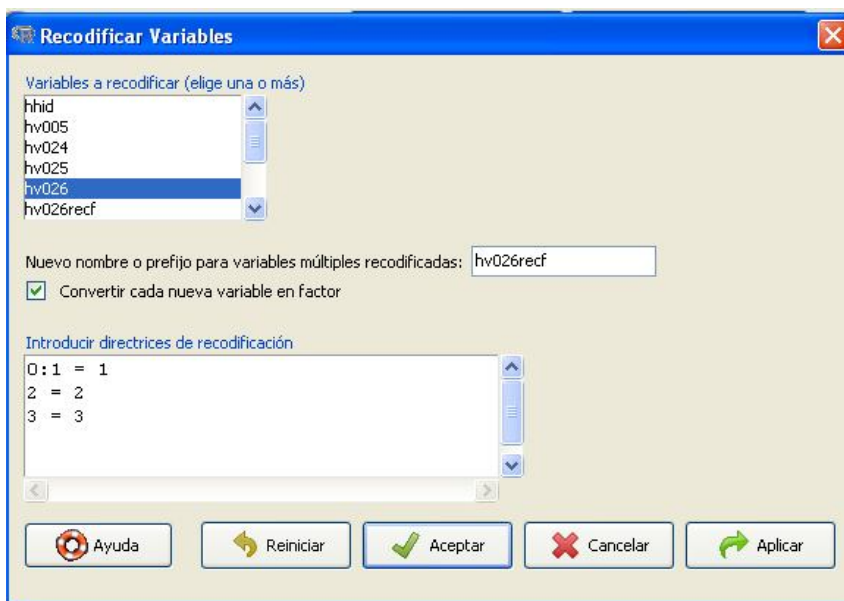
En el archivo original RPUNOE1.sav, esta variable hv026 tiene cuatro niveles: 0 = “Capital, ciudad grande”; 1 = Ciudad mediana; 2 = “Pueblo” y 3=“Campo”.

Capital, ciudad grande	Ciudad mediana	Pueblo
0	47	143
Campo		
360		

Como podemos observar, en la tabla de frecuencia solicitada a R, la categoría “Capital, ciudad grande”, no tiene elementos, por lo que vamos a crear otra variable a partir de la recodificación de hv026, a la que llamaremos hv026recf, que tendrá sólo tres niveles: 1 = Ciudad mediana; 2 = “Pueblo” y 3=“Campo”. Para ello desplegamos el menú:

Datos → Modificar variables del conjunto de datos activo → Recodificar variables

Luego elegimos hv026; escribimos su nuevo nombre (hv026recf) dejamos seleccionada la opción “Convertir cada nueva variable en factor” e introducimos las “directrices de recodificación”.



Al elaborar tablas de distribución de frecuencias, el resultado será el siguiente.

Campo	Ciudad mediana	Pueblo
360	47	143

Como vemos las categorías de las variables han sido ordenadas alfabéticamente. Para cambiar el orden de presentación de las categorías de la variable, podemos, ya sea especificarlo con el argumento `levels`, vía la ventana R Script:

```
PUNOEST$hv026recf <- Recode(PUNOEST$hv026, '0:1 =1; 2 = 2; 3 = 3',
  as.factor.result=TRUE, levels=c("Ciudad mediana", "Pueblo", "Campo"))
```

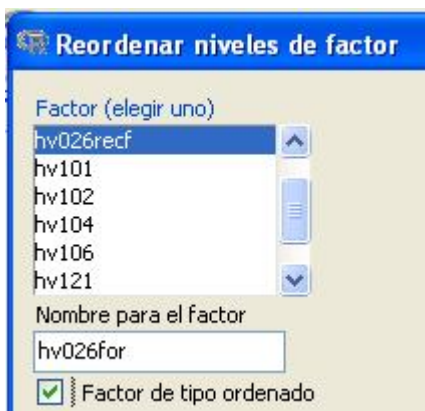
Con el resultado siguiente:

Ciudad mediana	Pueblo	Campo
47	143	360

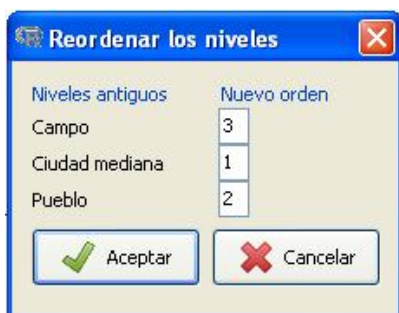
O podemos desplegar el menú:

Datos → Modificar variables del conjunto de datos activo → Reordenar niveles de factor

En la ventana emergente escogemos: “Reordenar niveles de factor”, seleccionamos la variable a reordenar, le damos un nuevo nombre, en este caso `hv026for`, activamos la opción factor ordenado, y hacemos clic sobre el botón “Aceptar”.



Luego reordenamos los niveles como se muestra en el siguiente gráfico y hacemos clic sobre el botón “Aceptar”.



#### 9.4.6. Modificar datos desde el Editor

Esta opción solo es posible cuando **cargamos** archivos a R. Ver punto 9.3.1.1., para más detalles.

### 9.5. Análisis univariado con R Commander

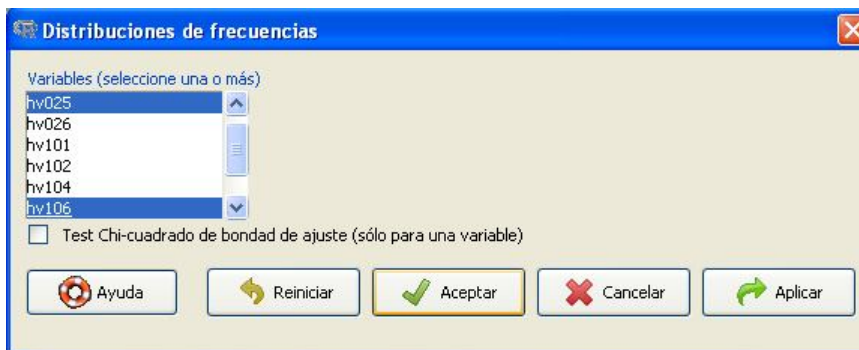
#### 9.5.1. Tablas de distribución de frecuencias absolutas y relativas

Al solicitar tablas de distribución de frecuencias en R, automáticamente se elaboran tanto las absolutas como las relativas.

Ejemplo: elaborar las tablas de distribución de frecuencias de las variables “lugar de residencia-2 categorías” (hv025) y “mayor nivel de educación alcanzado” (hv106).

Estadísticos → Resúmenes → Distribución de frecuencias...

En la ventana “Distribuciones de frecuencias”, seleccionamos las variables.



El resultado para la variable “lugar de residencia-2 categorías” (hv025) es el siguiente:

```
Salida
> .Table <- table(PUNOEST$hv025)
> .Table # counts for hv025
Urbana Rural
  190   360
> round(100*.Table/sum(.Table), 2) # percentages for hv025
Urbana Rural
 34.55  65.45
```

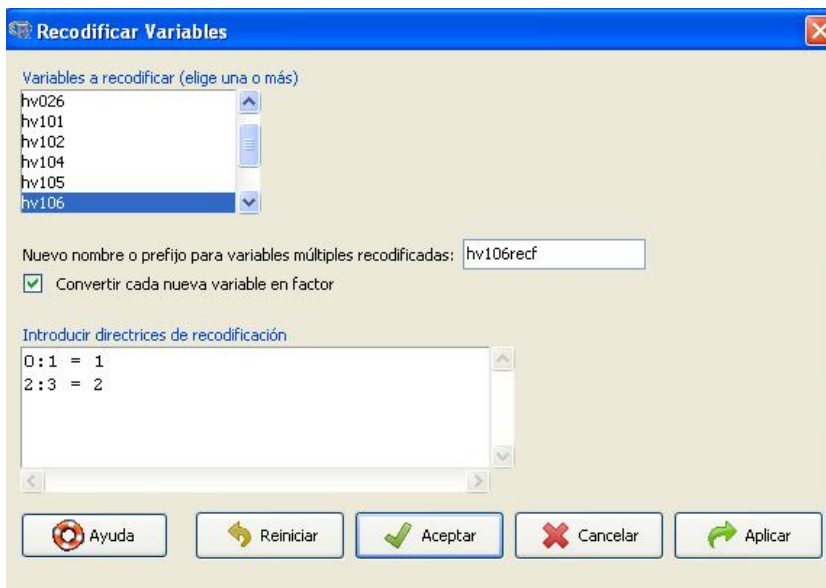
Los resultados para la variable “mayor nivel de educación alcanzado” (hv106) presenta niveles sin datos.

```
> .Table <- table(PUNOEST$hv106)
> .Table # counts for hv106
Sin nivel, inicial      Primaria      Secundaria      Superior      NS
                   0             165             385             0             0
> round(100*.Table/sum(.Table), 2) # percentages for hv106
Sin nivel, inicial      Primaria      Secundaria      Superior      NS
                   0             30             70             0             0
```

Para evitar que se presenten estos niveles, recodificaremos hv106 siguiendo la secuencia siguiente:

Datos → Modificar variables del conjunto de datos activo → Recodificar variables

En la ventana “Recodificar Variables”, seleccionamos la variable a recodificar (hv106), y le asignamos un nuevo nombre (hv106recf), aceptamos la opción predefinida por R Commander: “Convertir cada nueva variable en factor” e introducimos las directrices de recodificación como se ilustra a continuación



Al solicitar la distribución de frecuencia de la nueva variable `hv106recf`, el resultado será el siguiente:

```
> PUNOEST$hv106recf <- Recode(PUNOEST$hv106, '0:1 = 1; 2:3 = 2', as.factor.result=TRUE)
> .Table <- table(PUNOEST$hv106recf)
> .Table # counts for hv106recf

Primaria Secundaria
    165     385
> round(100*.Table/sum(.Table), 2) # percentages for hv106recf

Primaria Secundaria
    30     70
```

### 9.5.2. Medidas de resumen

- *Medidas de resumen para **todas** las variables del conjunto de datos*

Estadísticos → Resúmenes → Conjunto de datos activo

```

> summary(PUNOEST)
      hhid      hvidx      hv101      hv102      hv104      hv105
129604201: 4   Min.   :1.000   Hijo/hija   :468   No: 0   Hombre:293   Min.   :12.00
058006901: 3   1st Qu.:3.000   Nieto(a)   : 32   Sí:550   Mujer :257   1st Qu.:13.00
059107301: 3   Median :3.000   Hermano(a) : 14                                     Median :14.00
059606301: 3   Mean   :3.478   Jefe       : 12                                     Mean   :13.93
059608601: 3   3rd Qu.:4.000   Otro pariente: 12                               3rd Qu.:15.00
059802301: 3   Max.   :7.000   Hijo adoptivo: 11                             Max.   :16.00
(Other)      :531                                     (Other)   : 1

      hv106      hv108      hv121      hv122
Sin nivel, inicial: 0   Min.   : 2.000   No           : 40   Sin nivel, inicial: 40
Primaria           :165   1st Qu.: 6.000   Asiste actualmente :510   Primaria           : 50
Secundaria         :385   Median  : 8.000   Asiste algunas veces: 0   Secundaria         :455
Superior           : 0    Mean   : 7.585                                     Superior           : 5
NS                  : 0    3rd Qu.: 9.000                                     NS                  : 0
Max.               :11.000

      hv124      hv125      hv129      hv005      hv024
Min.   : 0.000   No: 11   Nunca asistió           : 0   Min.   : 514075   Puno           :550
1st Qu.: 7.000   Sí:539   Ingresado a la escuela   : 2   1st Qu.: 876555   Amazonas: 0
Median : 8.000   Avanzado           :491   Median :1043893   Ancash : 0
Mean   : 7.913   Repitente          : 17   Mean   :1053728   Apurimac: 0

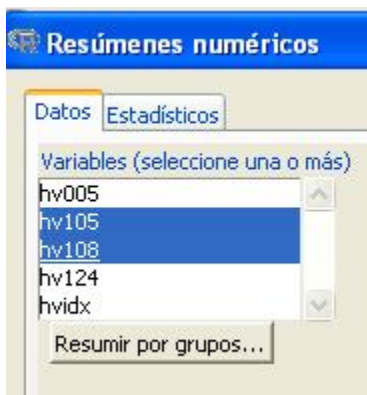
```

...

- *Resúmenes de variables cuantitativas.*

Estadísticos → Resúmenes → Resúmenes numéricos

En la ventana “Datos” seleccionamos las variables a resumir, en el ejemplo, las variables hv105 (edad) y hv108 (número de años de estudio):



Y en la ventana “Estadísticos”, podemos seleccionar diferentes medidas de resumen: tendencia central, dispersión y posición.





Para las variables analizadas los resultados son los siguientes:

```

      mean      sd IQR      cv  skewness  kurtosis 0% 25% 50% 75% 100%  n
hv105 13.925455 1.430824  2 0.1027488  0.05278762 -1.3087540 12 13 14 15 16 550
hv108  7.585455 1.775918  3 0.2341216 -0.14605517 -0.3012333  2  6  8  9 11 550

```

También podemos solicitar estas medidas por grupos; por ejemplo, según el área de residencia (hv025).



Con el resultado siguiente:

```

Variable: hv105
      mean      sd IQR      cv  skewness  kurtosis 0% 25% 50% 75% 100%  n
Urbana 13.88947 1.459649  2 0.1050903  0.08042944 -1.373219 12 13 14 15 16 190
Rural  13.94444 1.417056  2 0.1016215  0.03958193 -1.273613 12 13 14 15 16 360

Variable: hv108
      mean      sd IQR      cv  skewness  kurtosis 0% 25% 50% 75% 100%  n
Urbana  7.915789 1.662937  2 0.2100784  0.1280853 -0.9235558  4  7  8  9 11 190
Rural  7.411111 1.810797  3 0.2443354 -0.2192898 -0.2163096  2  6  7  9 11 360

```

## 9.6. Análisis bivariado con R Commander

### 9.6.1. Tablas cruzadas absolutas y relativas

Ejemplo: elaborar una tabla cruzada de la variable “lugar de residencia-3 categorías” (hv026for)<sup>20</sup> y la variable “mayor nivel educativo alcanzado-2 categorías” (hv106recf).

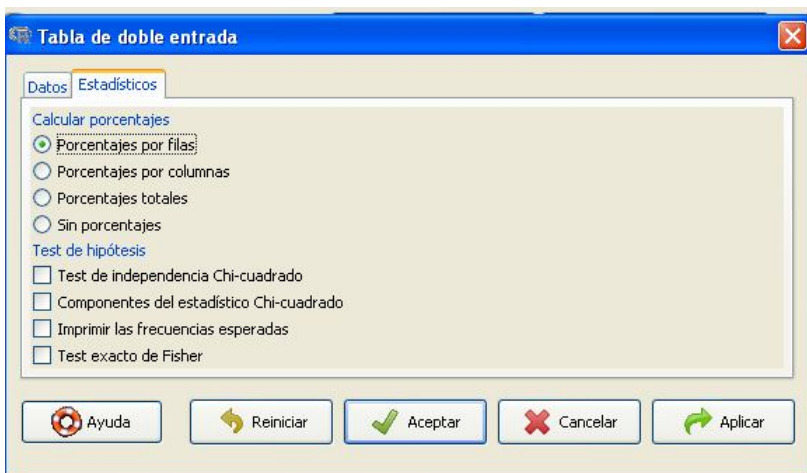
Para realizar tablas cruzadas o de contingencia desplegamos el menú:

Estadísticos → Tablas de contingencia → Tabla de doble entrada

En la ventana “Datos”, elegimos una variable fila (hv026for) y una variable columna (hv106recf).



En la ventana “Estadísticos”, elegimos “Porcentajes por filas”.



Y en la ventana de resultados aparecerá lo siguiente

<sup>20</sup> Creada en 9.4.5.

```
> .Table <- xtabs(~hv026for+hv106recf, data=PUNOEST)

> .Table
      hv106recf
hv026for  Primaria Secundaria
Ciudad mediana      7      40
Pueblo              38     105
Campo              120     240

> rowPercents(.Table) # Row Percentages
      hv106recf
hv026for  Primaria Secundaria Total Count
Ciudad mediana  14.9     85.1  100    47
Pueblo         26.6     73.4  100   143
Campo         33.3     66.7  100   360
```

También se puede pedir los porcentajes en *columns*.

Porcentajes por columnas

```
> colPercents(.Table) # Column Percentages
      hv106recf
hv026for  Primaria Secundaria
Ciudad mediana  4.2     10.4
Pueblo         23.0    27.3
Campo         72.7    62.3
Total         99.9   100.0
Count        165.0   385.0
```

Y porcentajes respecto al *total*.

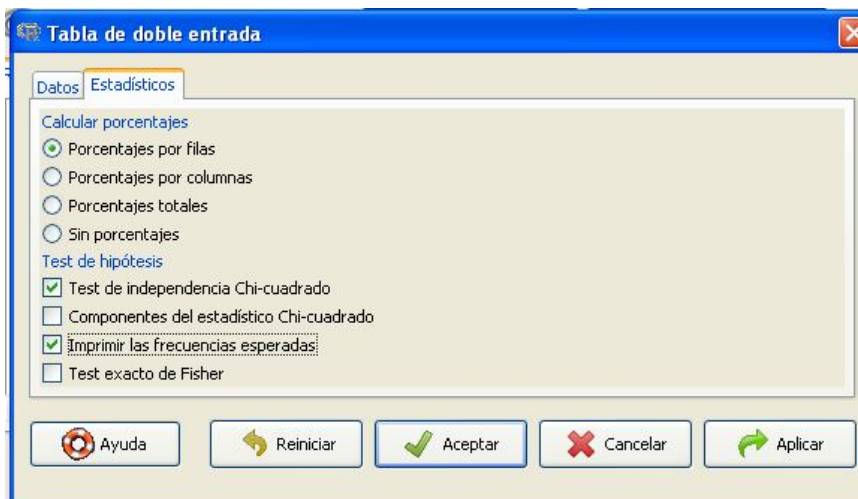
Porcentajes totales

```
> totPercents(.Table) # Percentage of Total
      Primaria Secundaria Total
Ciudad mediana  1.3     7.3   8.5
Pueblo         6.9    19.1  26.0
Campo        21.8    43.6  65.5
Total        30.0    70.0 100.0
```

### 9.6.2. Test de independencia con Chi cuadrado

Ejemplo: establecer si el máximo nivel educativo alcanzado por los niños de de 12 a 16 años en el departamento de Puno-Perú en 2007 (hv106recf), varía en función del lugar donde residen (hv026for).

Para realizar el test de independencia con Chi cuadrado, realizamos los procedimientos anteriores, pero además en la ventana “Estadísticos”, activamos las opciones: porcentaje en filas, Test de independencia Chi-cuadrado e imprimir frecuencias esperadas.



El resultado será el siguiente:

```
> .Table <- xtabs(~hv026for+hv106recf, data=PUNOEST)

> .Table
          hv106recf
hv026for  Primaria Secundaria
Ciudad mediana      7      40
Pueblo              38     105
Campo              120     240

> rowPercents(.Table) # Row Percentages
          hv106recf
hv026for  Primaria Secundaria Total Count
Ciudad mediana  14.9    85.1    100     47
Pueblo          26.6    73.4    100    143
Campo          33.3    66.7    100    360

> .Test <- chisq.test(.Table, correct=FALSE)

> .Test

      Pearson's Chi-squared test

data:  .Table
X-squared = 7.8117, df = 2, p-value = 0.02012

> .Test$expected # Expected Counts
          hv106recf
hv026for  Primaria Secundaria
Ciudad mediana  14.1    32.9
Pueblo          42.9   100.1
Campo          108.0   252.0
```

En base a los resultados del test, rechazamos la hipótesis nula de independencia entre el lugar de residencia y el máximo nivel educativo alcanzado por los niños de 12 a 16 años en el departamento de Puno  $\chi^2=7,81$ ,  $df=2$ ,  $n=550$ ;  $p < ,05$ . Es decir, existe una asociación significativa entre las variables.

### 9.6.3. Pruebas $t$

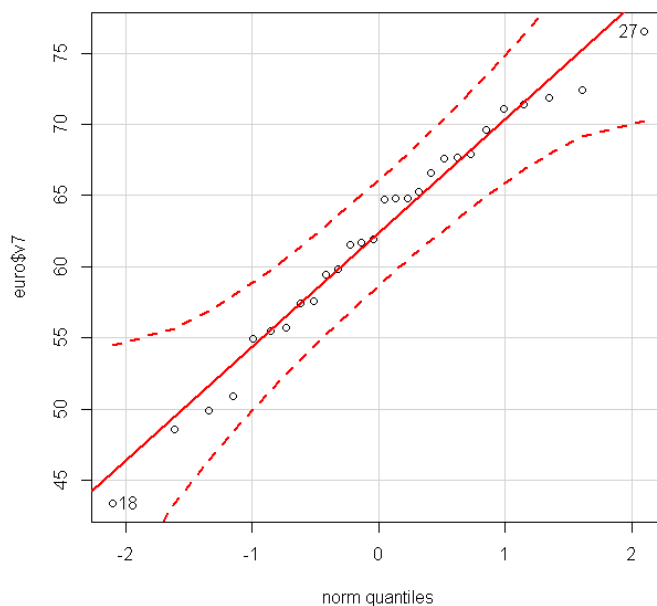
#### 9.6.3.1. Prueba $t$ para muestras independientes

En esta parte usaremos la base de datos `euro.sav`,<sup>21</sup> que leeremos en R como el conjunto de datos `euro`.

Ejemplo: queremos saber si la media de la tasa de ocupación femenina en 2011 (`v7`) en los países que conforman la Zona Euro 28, fue significativamente diferente para los países que ingresaron a dicha zona durante el siglo XX, a la de aquellos que lo hicieron durante el siglo XXI (`enterUE`).

Antes de realizar la prueba  $t$ , evaluamos la distribución normal de la variable `v7` así como la igualdad de varianzas de los grupos de análisis:

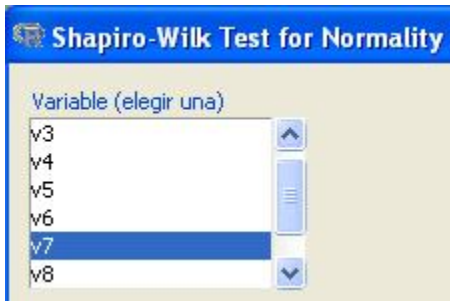
Para la prueba de normalidad usamos: el gráfico de comparación de cuantiles (Ver procedimientos sobre la construcción del gráfico en 9.7.6).



<sup>21</sup> Ver imagen del archivo `euro.sav` en Anexo 2, y una reseña sobre su contenido en 7.3.

Y para el test de Shapiro-Wilk, desplegamos el menú:

Estadísticos → Resúmenes → Test de Normalidad de Shapiro-Wilk...



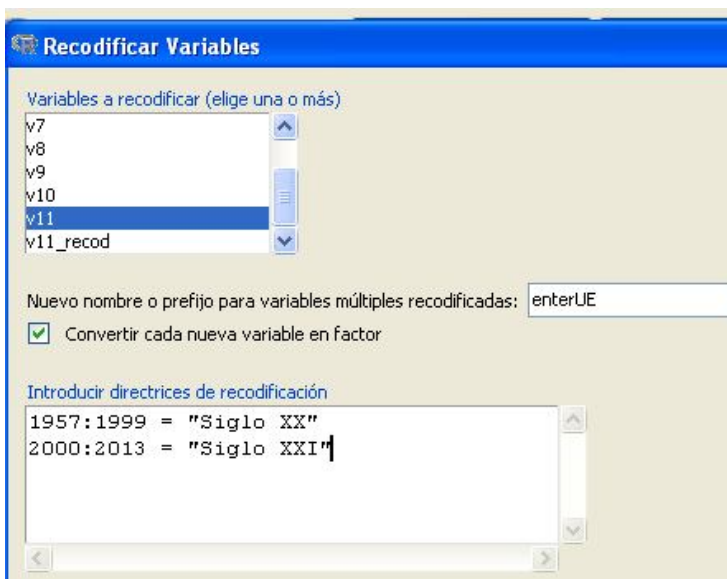
Shapiro-Wilk normality test

```
data: euro$v7
W = 0.9746, p-value = 0.7076
```

Probada la normalidad de la distribución de la variable, realizamos la Prueba de Levene sobre la igualdad de varianza de los grupos de análisis.

Pero antes de realizar la prueba de Levene, crearemos la variable “enterUE”, a partir de la variable v11, de tal manera que queden definidas dos categorías: países que entraron a la Unión Europea durante el siglo XX, y aquellos que lo hicieron a partir del siglo XXI. Para ello desplegamos el menú:

Datos → Modificar variables del conjunto de datos activo → Recodificar variables



- Prueba de Levene sobre igualdad de varianzas

Estadísticos → Varianzas → Test de Levene

Elegimos el factor (enterUE), la variable explicada (v7) y definimos como centro, la media.



Los resultados serán los siguientes:

```
> tapply(euro$v7, euro$enterUE, var, na.rm=TRUE)
Siglo XX Siglo XXI
69.91210  52.22577

> leveneTest(euro$v7, euro$enterUE, center=mean)
Levene's Test for Homogeneity of Variance (center = mean)
      Df F value Pr(>F)
group  1  0.3316 0.5697
      26
```

El test de Levene confirma la igualdad de las varianzas de los grupos de análisis (69,91 para los países que ingresaron a la Zona Euro en el siglo XX y 52,23 para los países que lo hicieron en el siglo XXI),  $p = 0,570$ .

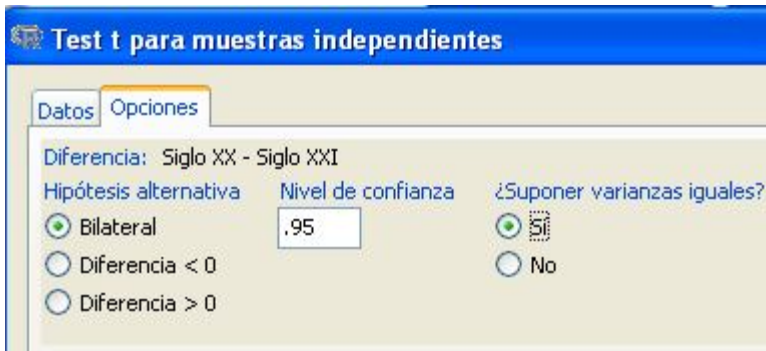
En base a esta información realizamos la prueba  $t$  para muestras independientes, desplegando el menú:

Estadísticos → Medias → Test para muestras independientes

En la ventana “Datos”, seleccionamos las variables para la prueba.



Y en la ventana “Opciones”, aceptamos las predefinidas por R Commander, salvo en la opción ¿Suponer varianzas iguales?, en donde seleccionaremos “Sí”.



Los resultados serán los siguientes:

```
Two Sample t-test

data:  v7 by enterUE
t = 1.6728, df = 26, p-value = 0.1064
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.139673  11.101724
sample estimates:
 mean in group Siglo XX mean in group Siglo XXI
                64.47333                59.49231
```

De acuerdo a los resultados podemos decir que en 2011, no existieron diferencias significativas entre las medias de la tasa de ocupación femenina de los países de la Zona Euro, según siglo de incorporación a dicha Zona. La media de la tasa de ocupación femenina de los países que ingresaron a la Zona Euro durante el siglo XX fue de 64,48 mientras que la de los países que lo hicieron durante el siglo XXI fue de 59,49,  $t=1,68$ ,  $df=26$ ,  $n=28$ ,  $p = 0,106$ .

### 9.6.3.2. Prueba $t$ para muestras relacionadas

Ejemplo: establecer si la diferencia de medias entre la tasa de ocupación femenina en 2011 (v7) y en 2006 (v6) es significativamente diferente de 0.

- *Evaluación de la distribución normal*

En los puntos 9.6.3.1., y 7.4., usando el gráfico de comparación de cuantiles y el test de Shapiro-Wilk evaluamos que las variables v7 y v6 presentaban una distribución normal.

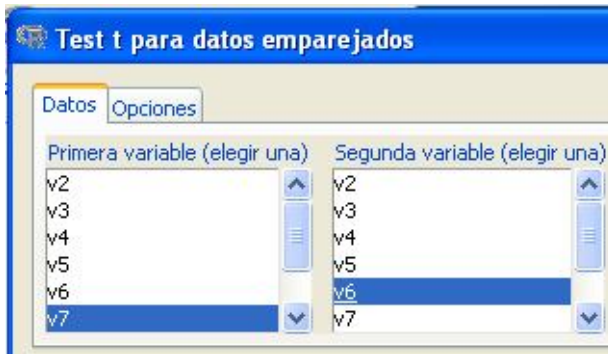
- *Prueba  $t$*



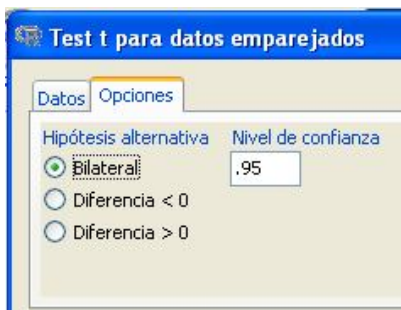
Para realizar la prueba  $t$  para muestras relacionadas, desplegamos el menú:

Estadísticos → Medias → Test  $t$  para datos relacionados...

En la ventana “Datos”, seleccionamos las variables a emparejar, en este caso  $v7$  (tasa de ocupación femenina en 2011) y  $v6$  (tasa de ocupación femenina en 2006).



En la ventana “Opciones” solicitamos un test bilateral a un nivel de confianza del 95%.



Los resultados serán los siguientes:

```

Paired t-test

data:  euro$v7 and euro$v6
t = 0.1372, df = 27, p-value = 0.8919
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.096204  1.253347
sample estimates:
mean of the differences
      0.07857143

```

En base a los resultados de la prueba, aceptamos la hipótesis nula de que la diferencia de medias entre la tasa de ocupación femenina de 2011 y la de 2006, en la Zona Euro, haya sido igual a 0,  $t = 0.137$ ,  $df = 27$ ,  $p < 0,892$ .

### 9.6.3.3. Prueba t para una muestra

Ejemplo: establecer si la media de la tasa de ocupación femenina 2011(v7) para la Zona Euro 28 fue significativamente **diferente** a 61,9 (valor que representa la tasa de ocupación femenina en la Zona Euro 17, en el mismo período) ( $H_1$ ).

De acuerdo con el punto 9.6.3.1., podemos decir que la variable v7, presenta una distribución normal.

Para realizar la prueba t para una muestra, desplegamos el menú:

Estadísticos → Medias → Test t para una muestra

Seleccionamos una variable, la hipótesis alternativa  $\neq \mu_0$ ; establecemos un valor hipotético para la hipótesis nula, en este caso 68,5 y un nivel de confianza del 95%.

Los resultados son los siguientes:

```
One Sample t-test
```

```
data: euro$v7
t = 0.17, df = 27, p-value = 0.8663
alternative hypothesis: true mean is not equal to 61.9
95 percent confidence interval:
 59.01384 65.30759
sample estimates:
mean of x
 62.16071
```

En base a los resultados, aceptamos la hipótesis nula. Por lo que podemos decir que la media de la tasa de ocupación femenina para la Zona Euro 28 en 2011 no fue significativamente diferente de 61,9,  $t=0,17$ ,  $df=27$ ,  $p<.866$ . La media para la Zona Euro 28 se calculó en 62,16 dentro de un intervalo de confianza que va de 59,01 a 65,31.

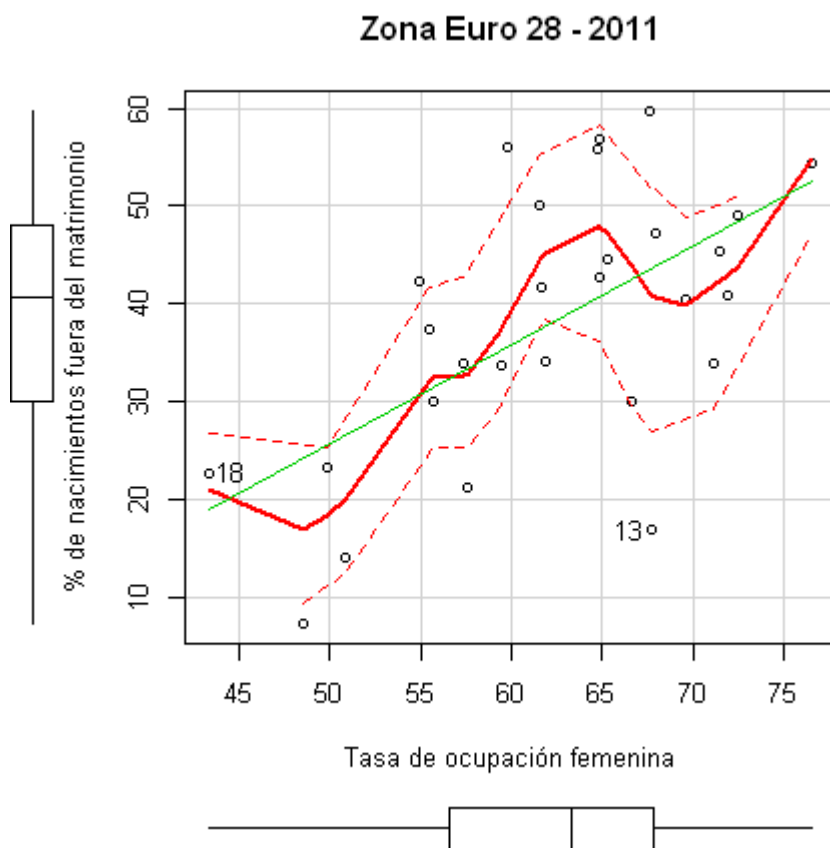
### 9.6.4. Correlación bivariada

Ejemplo: establecer la correlación entre el la tasa de ocupación femenina en 2011 en la Zona Euro 28 (v7) y el porcentaje de niños nacidos fuera del matrimonio en 2011 (v3), el nivel de significación de esta estimación y el intervalo de confianza en el que está comprendida.

Antes de establecer la correlación entre las variables v7 y v3, vamos a observar la forma de la relación entre las variables así como la presencia de valores extremos, utilizando un diagrama de dispersión, además realizaremos la prueba de normalidad para la variable v3.

- *Evaluación del sentido de la relación y detección de valores extremos utilizando un diagrama de dispersión*

Ver detalles de su construcción en el punto 9.7.5.

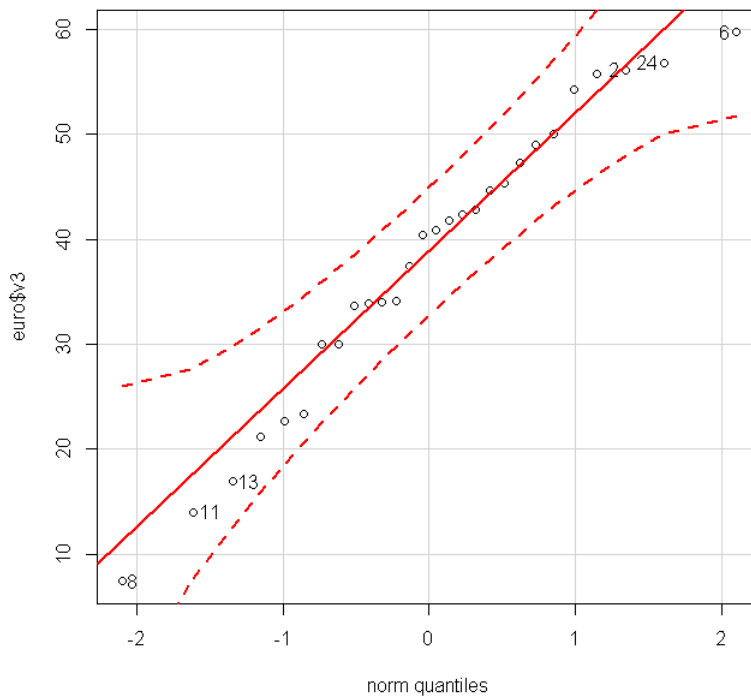


El gráfico nos permite identificar una relación lineal positiva entre las variables, sin embargo, existe un valor que se aleja de esta forma de relación, sin constituir un valor extremo: el 13 (correspondiente a Chipre).

- *Evaluación de la distribución normal*

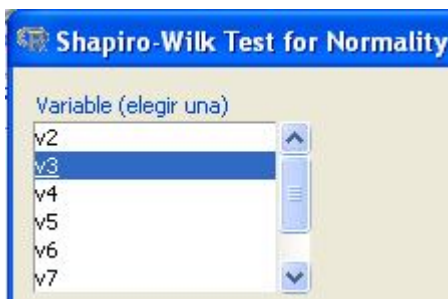
En 9.6.3.1., utilizando el gráfico de comparación cuantiles y el Test de Shapiro-Wilk, realizamos la evaluación de la distribución normal de la variable  $v_7$ . Aquí realizaremos lo mismo para la variable  $v_3$ .

A continuación se muestra el gráfico de comparación de cuantiles (Ver detalles de su construcción en 9.7.6):



Para realizar el test de Shapiro-Wilk desplegamos el menú:

Estadísticos → Resúmenes → Test de normalidad de Shapiro-Wilk...



Shapiro-Wilk normality test

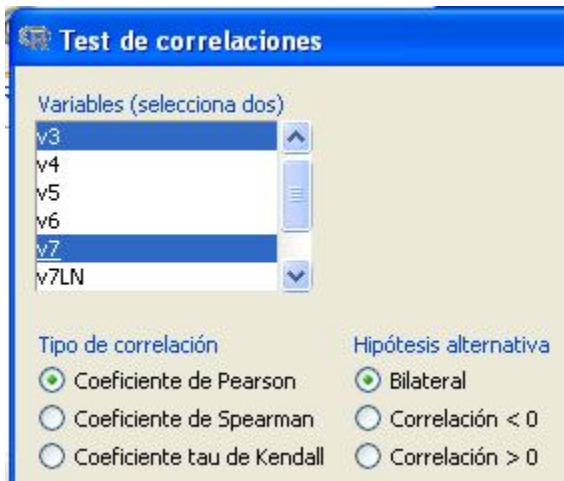
```
data: euro$v3
W = 0.9675, p-value = 0.516
```

Comprobada la distribución normal de ambas variables, realizamos el test de correlación.

- *Test de correlación*

Estadísticos → Resúmenes → Test de correlación

Luego seleccionamos las variables, y dejamos marcadas las opciones por defecto: Tipo de correlación: Coeficiente de Pearson e Hipótesis alternativa: Bilateral



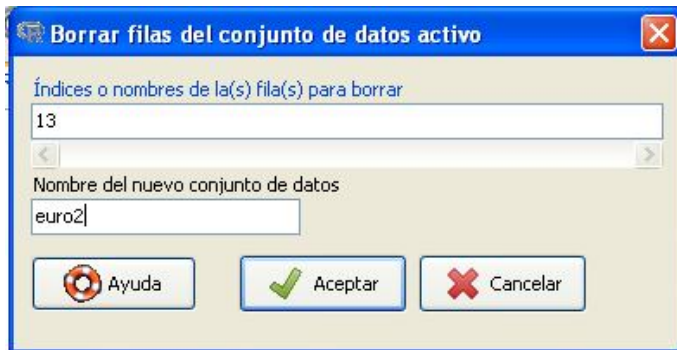
Pearson's product-moment correlation

```
data: euro$v3 and euro$v7
t = 3.7891, df = 26, p-value = 0.0008083
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2873127 0.7930582
sample estimates:
      cor
0.5964557
```

Existe una correlación positiva entre las variables analizadas. Lo que indica que cuando la tasa de ocupación femenina aumentó, el porcentaje de niños nacidos fuera del matrimonio, también lo hizo, el coeficiente de correlación ( $r$ ) fue igual 0,596,  $p < ,000$ , y se calculó en un intervalo de confianza de 0.287 y 0.793, lo que indica que la correlación difiere significativamente de 0.

Si retiramos del conjunto de datos a Chipre, país que representa el valor que más se aleja de la relación lineal, siguiendo la secuencia:

Conjunto de datos activo → Borrar fila(s) del conjunto de datos activo...



Los resultados de la prueba de correlación serían los siguientes:

```

Pearson's product-moment correlation

data: euro2$v3 and euro2$v7
t = 4.5576, df = 25, p-value = 0.0001173
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3946812 0.8389107
sample estimates:
      cor
0.6736571

```

## 9.7. Gráficos con R Commander

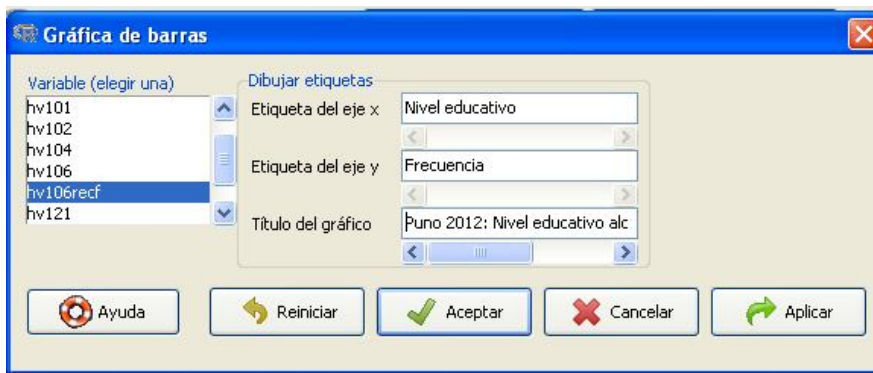
En R Commander, usando la opción **Gráficos** es posible realizar una gran variedad de gráficos de variables cualitativas y cuantitativas, así como de sus relaciones; en este punto realizaremos seis de los gráficos más utilizados en ciencias sociales.

### 9.7.1. Gráfico de barras

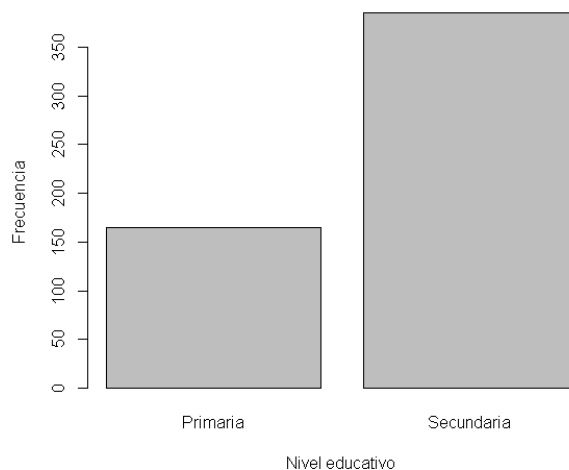
Para elaborar un gráfico de barras desplegamos el menú:

Gráficos → Gráfica de barras

Seleccionamos la variable a graficar y escribimos las etiquetas respectivas.



**Puno 2012: Nivel educativo entre niños de 12 a 16 años**



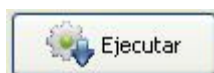
Podemos mejorar la presentación del gráfico de barras añadiendo las siguientes instrucciones en la ventana Script:

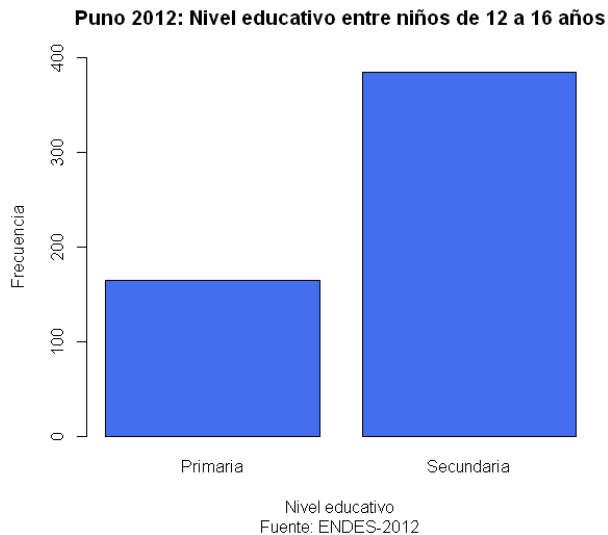
- `sub="Fuente: ENDES-2012"`, para indicar la fuente de datos
- `ylim=c(0,400)`, para indicar que la escala de valores del eje y va de 0 hasta 400
- `col="royalblue2"`, para cambiar el color de las barras.

La secuencia quedará así:

```
barplot(table(PUNOEST$hv106recf), xlab="Nivel educativo", ylab="Frecuencia",
  main="Puno 2012: Nivel educativo entre niños de 12 a 16 años", sub="Fuente: ENDES-2012",
  ylim=c(0,400), col="royalblue2")
```

Luego marcamos las instrucciones y hacemos clic sobre el botón "Ejecutar".





### - Colores

Podemos acceder al nombre de los colores disponibles, desplegando el menú:

Gráfica → Gama de colores...

Luego aparecerá el siguiente gráfico.



Al hacer clic sobre uno de los recuadros de colores, se abrirá una gama más grande de colores, desde la cual podemos seleccionarlos e inclusive personalizarlos.





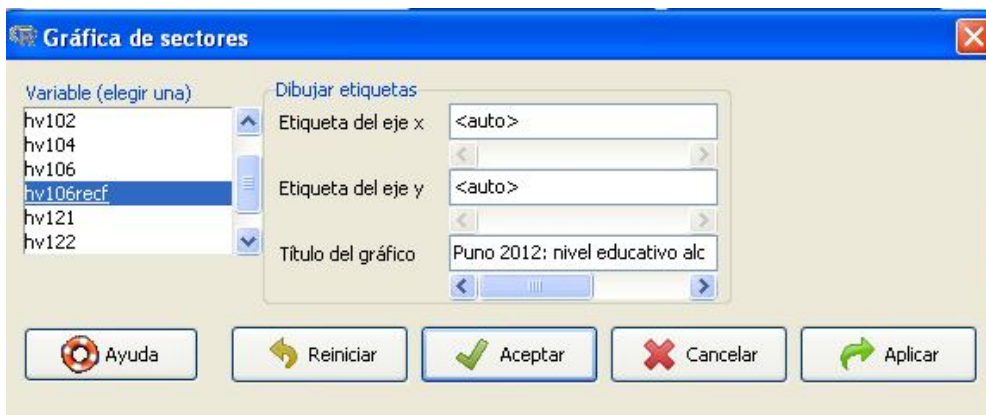
Luego de seleccionar el color, al hacer clic en Aceptar, aparecerá nuevamente la ventana “Elegir la gama de colores”, pero esta vez se escribirá el nombre del color seleccionado.

### 9.7.2. Gráfico de sectores

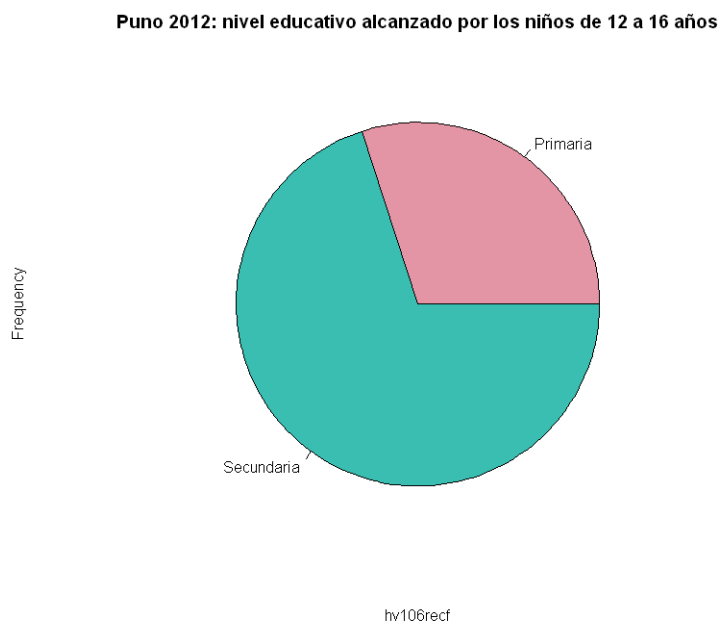
Para realizar este gráfico, desplegamos el menú:

Gráficas → Gráfica de sectores...

Luego seleccionamos una variable, en este caso: nivel educativo alcanzado (hv106recf).



El resultado será el siguiente:



Para eliminar las etiquetas: “Frecuency”, “hv106recf” y agregar la fuente al gráfico, en la ventana R Script borraremos las siguientes instrucciones:

```
xlab="hv106recf",
```

ylab="Frequency" ,

y agregaremos la instrucción:

sub="Fuente: ENDES-2012" ,

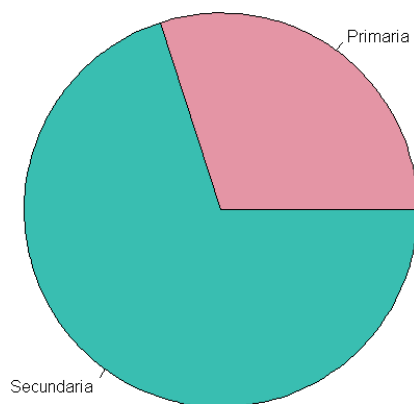
La modificación quedará como sigue:

```
pie(table(PUNOEST$hv106recf), labels=levels(PUNOEST$hv106recf),
     main="Puno 2012: nivel educativo alcanzado por los niños de 12 a 16 años",
     sub="Fuente: ENDES-2012",
     col=rainbow_hcl(length(levels(PUNOEST$hv106recf))))
```

Luego sombrearemos la instrucción y haremos clic sobre el botón “Ejecutar”

Entonces aparecerá el siguiente gráfico en valores absolutos:

**Puno 2012: nivel educativo alcanzado por los niños de 12 a 16 años**



Fuente: ENDES-2012

### 9.7.3. Histograma

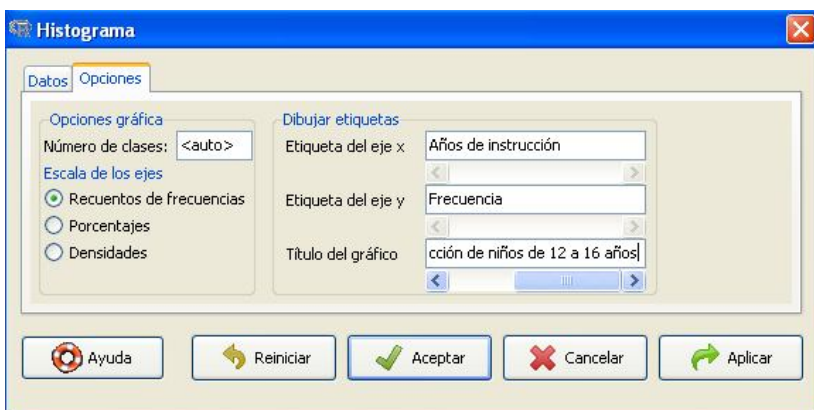
Para realizar este gráfico, desplegamos el menú:

Gráficas → Histograma...

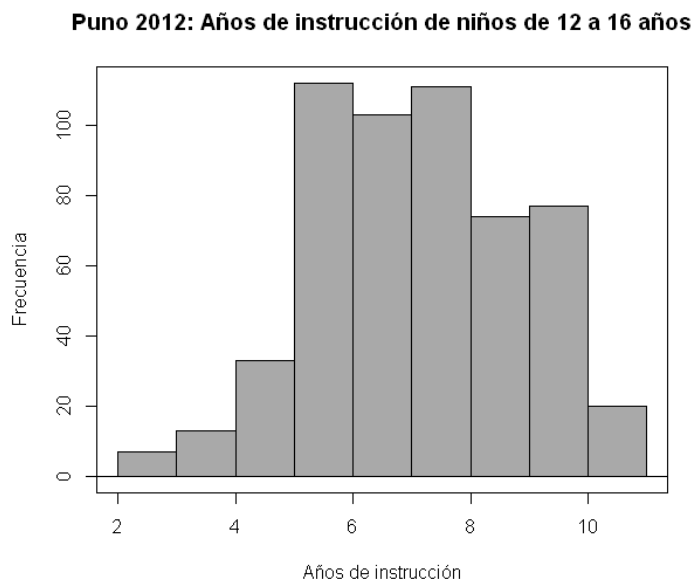
En la ventana de “Datos”, seleccionamos la variable a graficar.



En la ventana “Opciones” aceptamos las “Opción gráfica”: Número de clases: auto, y en la “Escala de ejes”: Recuentos de frecuencias. Luego etiquetamos los ejes y le damos un título al gráfico.



El resultado será el siguiente

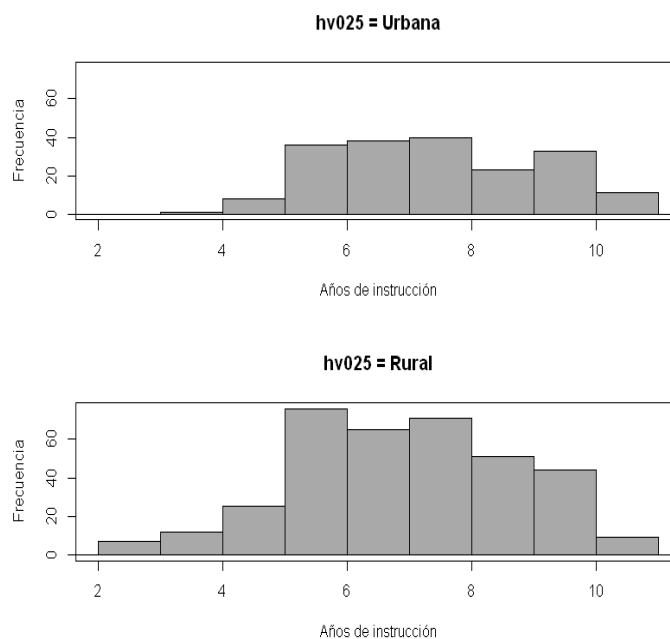


Para realizar histogramas por grupos, activamos la ventana “Grupos” en la ventana “Datos”, y seleccionamos la variable de agrupación, en este caso hv025 (lugar de residencia).



El resultado será el siguiente

Puno 2012: años de instrucción de niños de 12 a 16 años según lugar de residencia



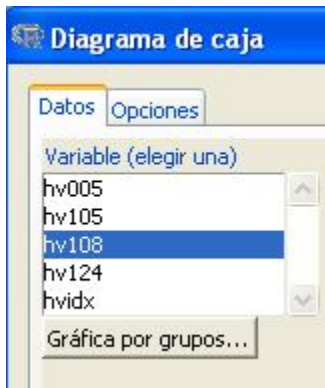
#### 9.7.4. Diagrama de caja

En este punto mostraremos primero, cómo elaborar diagramas de caja para una variable cuantitativa, y luego, cómo hacerlo para una variable cuantitativa en función a una cualitativa

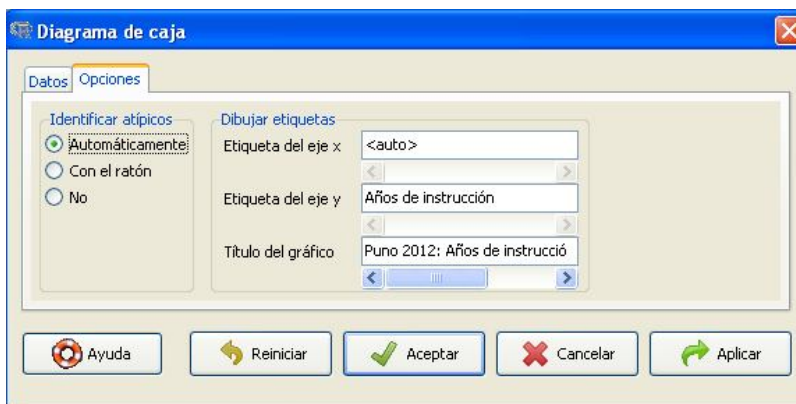
- *Diagrama de caja de una variable cuantitativa*

Gráficas → Diagrama de caja...

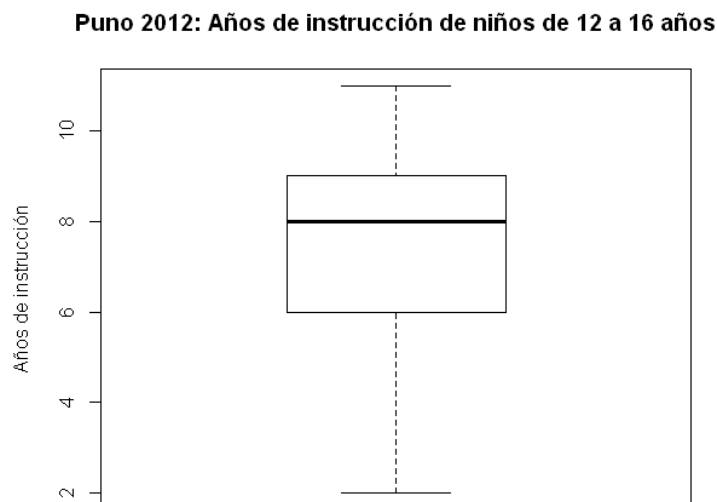
En la ventana “Datos”, seleccionamos la variable a graficar. En este caso la variable hv108 (Número de años de estudio).



En la ventana “Opciones”, aceptamos la opción predefinida “Identificar atípicos”: Automáticamente, luego escribimos las etiquetas.



El resultado será el siguiente:



- *Diagrama de caja de una variable cuantitativa en función a una cualitativa*

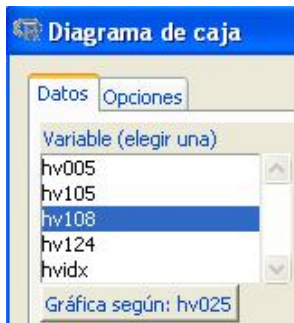
Para presentar el diagrama de caja de una variable cuantitativa en función de grupos de una variable cualitativa, en la ventana de “Datos”, además de seleccionar la variable cuantitativa, haremos clic sobre el botón:

Gráfica por grupos...

Luego en la ventana “Grupos”, elegiremos la variable cualitativa que definirá los grupos según los cuales necesitamos obtener los diagramas de caja para la variable hv108 (número de años de instrucción), en este caso, la variable hv025 (lugar de residencia).



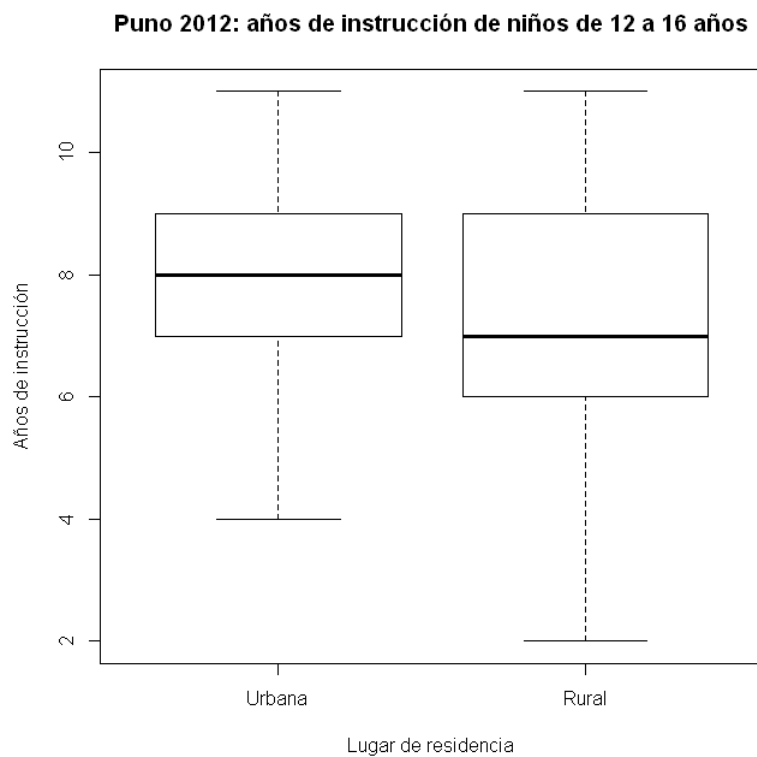
Después de realizar esta operación veremos que se registra en la ventana “Datos” el mensaje: “Gráfica según: hv025”.



Luego ingresaremos a la ventana “Opciones”, aceptaremos la opción predefinida “Identificar atípicos”: Automáticamente y finalmente escribiremos las etiquetas correspondientes.

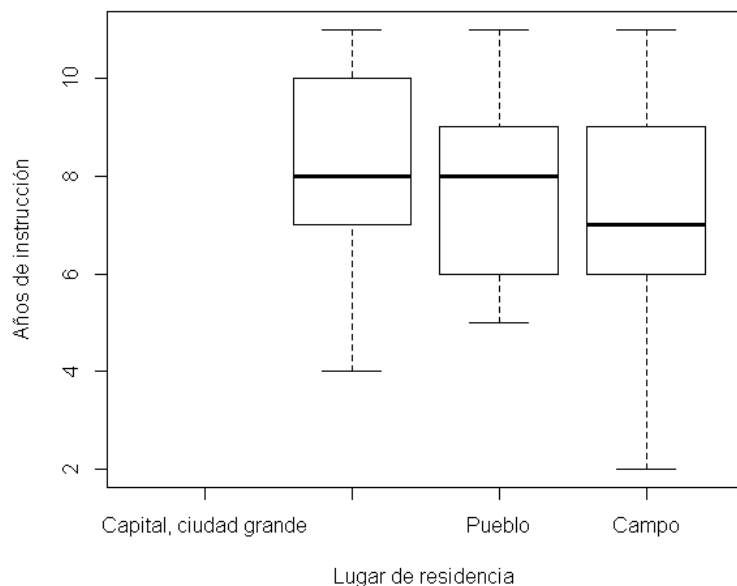


El resultado será el siguiente:



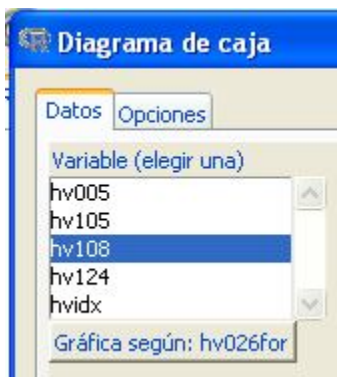
Cuando ciertos niveles de la variable de agrupación carecen de información, esto se refleja en los gráficos. Como ejemplo veremos la variable hv026.

### Puno 2012: Años de instrucción de niños de 12 a 16 años



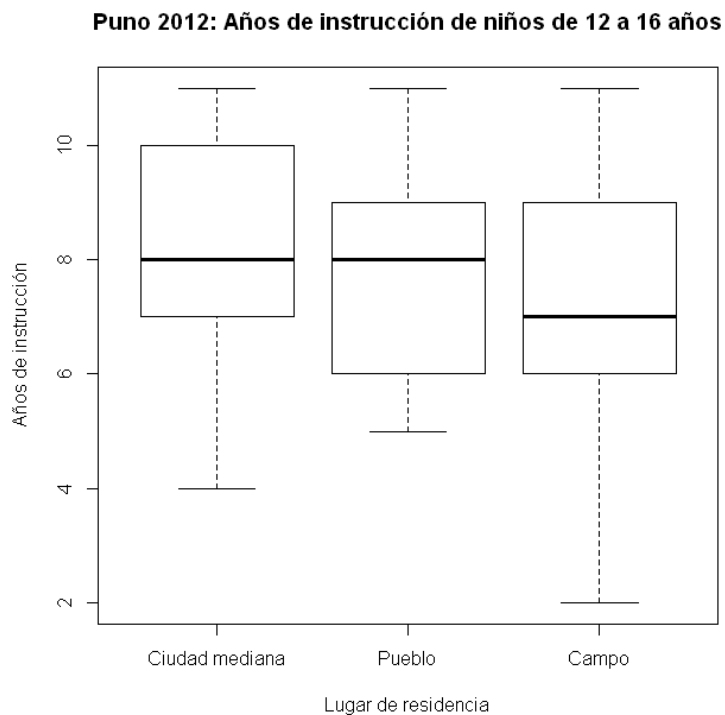
En el ejemplo, la variable `hv026`, tiene cuatro niveles, donde el nivel 0 = “Capital, ciudad grande”, carece de información. Por ello, antes de realizar el gráfico debemos recodificar la variable `hv026`, creando la nueva variable `hv026for` (ver procedimiento en 9.4.5).

Luego instruiremos a R para que realice el gráfico con la variable `hv026for`



El resultado será el siguiente:



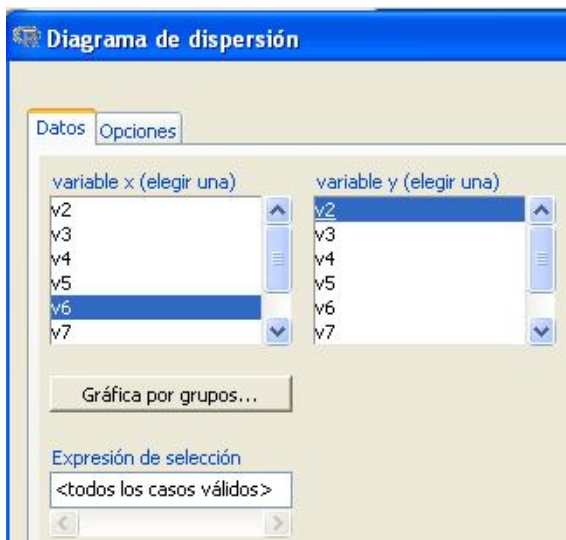


### 9.7.5. Diagrama de dispersión

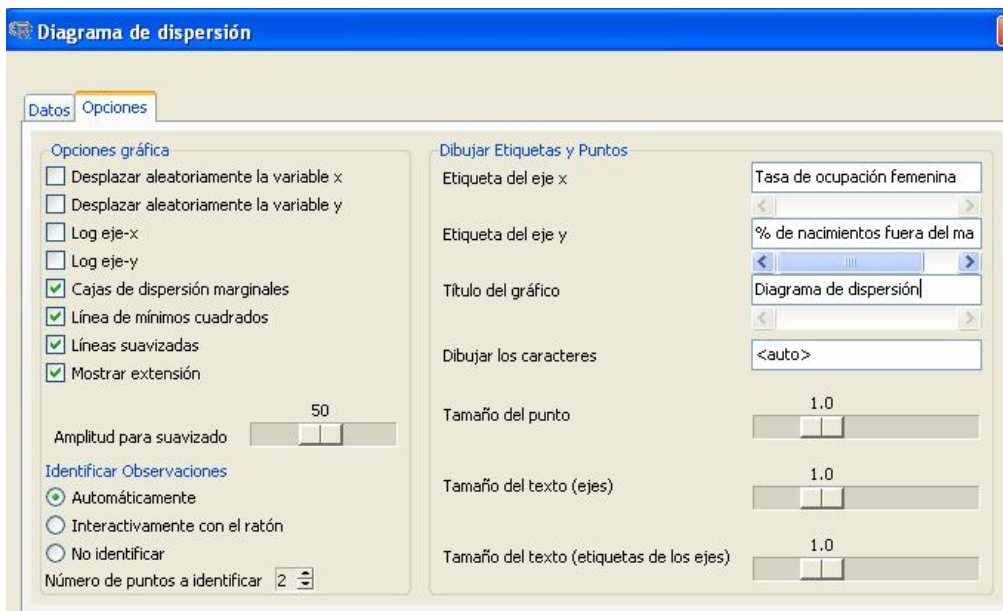
Para realizar este gráfico, desplegamos el menú:

Gráficas → Diagrama de dispersión

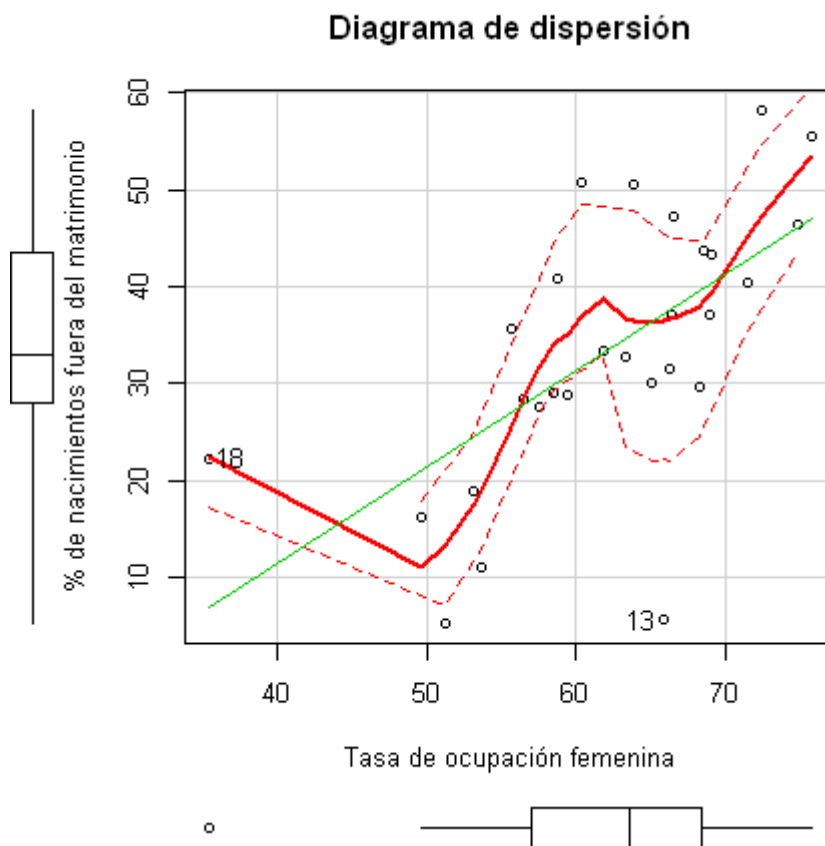
Luego seleccionamos las variables  $x$  e  $y$ .



En la ventana “Opciones”, aceptamos las opciones de gráfica predefinidas por R y escribimos las etiquetas.



El resultado será el siguiente:

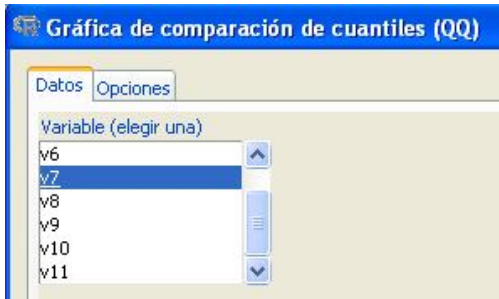


### 9.7.6. Gráfico de comparación de cuantiles

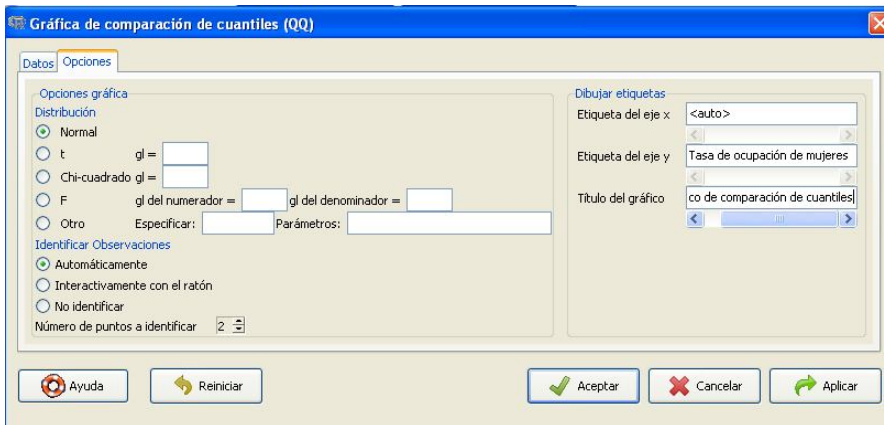
Para realizar este gráfico, desplegamos el menú:

Gráficas → Gráfica de comparación de cuantiles

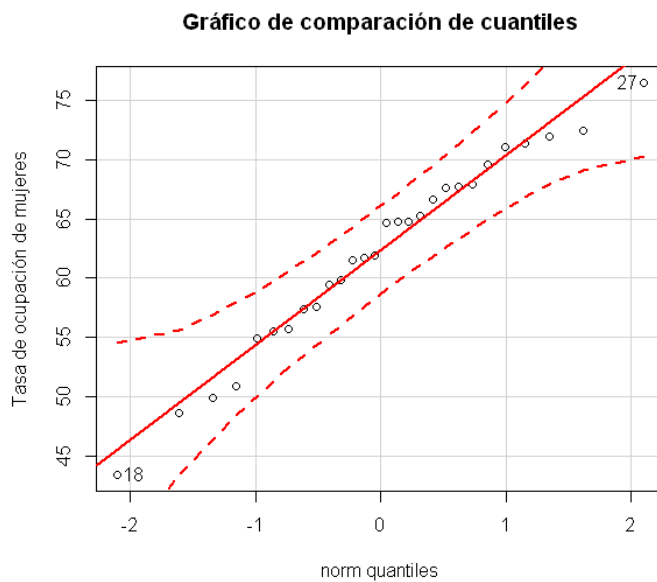
Luego seleccionamos la variable a graficar.



Aceptamos las opciones de gráfica predefinidas: Distribución normal e Identificar Observaciones Automáticamente. Y finalmente escribimos las etiquetas correspondientes.



El resultado será el siguiente:



## BIBLIOGRAFÍA

Chihara, L. (2010). *R Guide*. Consultado el 17/01/2014 en <http://people.carleton.edu/~lchihara/Splus/RVectors.pdf>

Correa, J. C., y González, N. (2002). *Gráficos Estadísticos con R*. Consultado el 14/01/2014 en <http://cran.r-project.org/doc/contrib/grafi3.pdf>

Fox, J. (2011). *Recode a variable*. En R Documentation: <http://127.0.0.1:25813/library/car/html/recode.html>

Gruber, J. (2004). Markdown. *Daring Fireball*. Consultado el 26/02/2014 en: <http://daringfireball.net/projects/markdown/>

Ihaka, R. (1998). A Free Software Project. A Brief History. *R: Past and Future History*. Consultado el 14/01/2014 en [http://cran.r-project.org/doc/html/interface98-paper/paper\\_2.html](http://cran.r-project.org/doc/html/interface98-paper/paper_2.html)

Nakazawa, M. (2013). *Package 'pyramid'*. Consultado el 18/01/2011 en <http://cran.r-project.org/web/packages/pyramid/pyramid.pdf>

Paradis, E. (2003). *R para principiantes*. Consultado el 02/08/2011 en [http://cran.r-project.org/doc/contrib/rdebuts\\_es.pdf](http://cran.r-project.org/doc/contrib/rdebuts_es.pdf)

R Project <http://cran.r-project.org>

R Tutorial #8 – Reading data from files. (05/03/2011). Consultado el 02/08/2011 en <http://www.youtube.com/watch?v=9kImnwZHQyc&feature=related>

SMCS-Institut de statistique-UCL. (2008). *Modules de formation. Pratique de la statistique avec SAS Entreprise Guide*. Institut de statistique, Université catholique de Louvain.

Universidad de Puerto Rico- Capítulo 3. Estadística descriptiva. Consultado el 25/01/2014 en <http://math.uprag.edu/tablas-frecuencias-R.pdf>

Venable, W. N. y Smith, D. M. (2011). *An Introduction to R. Notes on R: A Programming Environment for Data Analysis and Graphics. Version 2.13.1*. R Development Core Team. Consultado el 02/08/2011 en <http://cran.r-project.org>.

Venable, W. N. y Smith, D. M. (2013). *An Introduction to R. Notes on R: A Programming Environment for Data Analysis and Graphics. Version 3.02*. R Development Core Team. Consultado el 20/02/2014 en <http://cran.r-project.org/doc/manuals/R-intro.pdf>

# ANEXOS

## 1. Marco de datos mater1

	caseid	mmidx	mm6	mm7		mm9	mm11	mm13	mm14	mm15	mmc1	mmc2	v001	v002	v003	v101	v102	v103	v190
1	000804001	03	7	14	25	Murió durante el embarazo	Embarazo	2	3	9998	7	7	8	40	3	Amazonas	Rural	Pueblo	Pobre
2	001009101	02	2	6	30	Murió durante el embarazo	Parto	1	5	9998	8	4	10	91	2	Amazonas	Urbano	Pueblo	Medio
3	001202101	02	1	NA	38	Murió durante el embarazo	Parto	3	7	2012	5	3	12	21	2	Amazonas	Rural	Campo	Muy pobre
4	001607501	03	3	11	14	Murió durante el embarazo	Embarazo	3	0	9998	6	6	16	75	3	Amazonas	Rural	Campo	Muy pobre
5	002106101	02	2	36	18	Murió durante el embarazo	Embarazo	3	1	9998	10	5	21	61	2	Amazonas	Rural	Campo	Muy pobre
6	005700501	02	3	32	17	Murió durante el embarazo	Embarazo	2	0	9998	9	7	57	5	2	Ancash	Rural	Pueblo	Pobre
7	007703301	02	3	NA	33	Muerte no contada	Embarazo	2	0	1999	4	4	77	33	2	Apurímac	Rural	Capital, ciudad grande	Muy pobre
8	008101201	02	2	30	21	Murió durante el embarazo	Parto	3	0	9998	6	5	81	12	2	Apurímac	Rural	Campo	Muy pobre
9	008106601	02	1	22	23	Murió durante el embarazo	Embarazo	3	1	9998	6	4	81	66	2	Apurímac	Rural	Campo	Muy pobre
10	010700701	02	1	2	20	Murió durante el embarazo	Embarazo	1	1	9998	3	0	107	7	2	Arequipa	Urbano	Capital, ciudad grande	Medio
11	010804701	02	5	8	23	Muerte no contada	Embarazo	3	2	9998	7	4	108	47	2	Arequipa	Urbano	Campo	Pobre
12	012002901	02	1	NA	30	Murió durante el embarazo	Parto	3	1	1983	10	4	120	29	2	Ayacucho	Rural	Campo	Medio
13	012002901	02	10	19	18	Murió durante el embarazo	Parto	3	1	9998	10	4	120	29	2	Ayacucho	Rural	Campo	Medio
14	012602401	02	9	10	38	Murió durante el embarazo	Aborto	1	6	9998	9	9	126	24	2	Ayacucho	Rural	Campo	Pobre
15	016105601	01	5	NA	43	Muerte no contada	Parto	3	6	2005	7	6	161	56	1	Cajamarca	Rural	Campo	Muy pobre
16	016300401	01	1	23	36	Muerte no contada	Aborto	3	7	9998	4	4	163	4	1	Cajamarca	Rural	Campo	Muy pobre
17	018301701	01	3	37	14	Murió durante el embarazo	Embarazo	3	0	9998	8	7	183	17	1	Cusco	Rural	Campo	Muy pobre
18	018607701	02	5	8	27	Murió durante el embarazo	Aborto	1	0	9998	8	0	186	77	2	Cusco	Rural	Campo	Pobre
19	020112401	01	4	23	20	Muerte no contada	Parto	3	0	9998	11	9	201	124	1	Huancavelica	Urbano	Pueblo	Pobre
20	020601301	02	1	18	25	Murió durante el embarazo	Embarazo	3	2	9998	10	9	206	13	2	Huancavelica	Rural	Campo	Muy pobre
21	020604301	01	1	NA	44	Muerte no contada	Parto	2	5	2003	3	2	206	43	1	Huancavelica	Rural	Campo	Muy pobre
22	021105101	02	6	10	27	Murió durante el embarazo	Embarazo	3	2	9998	7	6	211	51	2	Huancavelica	Rural	Campo	Muy pobre
23	021704901	02	2	30	22	Murió durante el embarazo	Aborto	2	12	9998	4	4	217	49	2	Huancavelica	Rural	Campo	Muy pobre
24	022705101	02	3	3	32	Murió durante el embarazo	Parto	2	4	9998	6	3	227	51	2	Huánuco	Rural	Campo	Muy pobre
25	026706001	02	3	20	20	Murió durante el embarazo	Embarazo	3	1	9998	7	1	267	60	2	Ica	Rural	Pueblo	Pobre
26	027509401	01	1	21	17	Murió durante el embarazo	Embarazo	1	0	9998	7	3	275	94	1	Junin	Urbano	Ciudad	Rico
27	028905701	04	1	NA	32	Murió durante el embarazo	Aborto	1	0	2012	5	2	289	57	4	Junin	Urbano	Pueblo	Muy pobre
28	029311901	02	2	NA	25	Muerte no contada	Aborto	1	4	1985	8	2	293	119	2	Junin	Rural	Campo	Medio
29	031507501	02	3	32	30	Murió durante el embarazo	Parto	3	1	9998	11	10	315	75	2	La Libertad	Urbano	Campo	Medio
30	033703601	02	5	13	26	Muerte no contada	Aborto	1	0	9998	6	4	337	36	2	Lambayeque	Urbano	Campo	Pobre
31	034706301	02	2	NA	27	Murió durante el embarazo	Embarazo	1	2	2008	6	1	347	63	2	Lambayeque	Urbano	Capital, ciudad grande	Medio
32	035411501	02	1	10	45	Murió durante el embarazo	Parto	1	5	9998	4	3	354	115	2	Lima	Urbano	Capital, ciudad grande	Pobre
33	036412301	02	8	NA	14	Muerte no contada	Aborto	2	0	2009	8	4	364	123	2	Lima	Urbano	Pueblo	Pobre
34	042405701	08	6	10	42	Muerte no contada	Aborto	1	6	9998	14	11	424	57	8	Lima	Urbano	Capital, ciudad grande	Muy rico
35	044503301	02	1	19	22	Murió durante el embarazo	Aborto	3	1	9998	2	0	445	33	2	Lima	Rural	Campo	Muy pobre
36	044610801	02	1	NA	47	Murió durante el embarazo	Parto	1	8	2007	10	5	446	108	2	Loreto	Urbano	Pueblo	Pobre
37	044902801	02	4	NA	20	Muerte no contada	Aborto	2	1	2005	10	1	449	28	2	Loreto	Urbano	Ciudad	Medio
38	045404001	02	5	26	15	Muerte no contada	Aborto	2	0	9998	11	5	454	40	2	Loreto	Urbano	Pueblo	Muy rico
39	045604201	02	4	33	20	Murió durante el embarazo	Parto	2	2	9998	10	10	456	42	2	Loreto	Urbano	Pueblo	Rico
40	046801401	02	6	10	15	Murió durante el embarazo	Aborto	1	0	9998	9	5	468	14	2	Loreto	Urbano	Ciudad	Pobre
41	047507701	01	3	NA	19	Murió durante el embarazo	Aborto	1	0	1995	3	3	475	77	1	Madre de Dios	Urbano	Capital, ciudad grande	Medio
42	047706701	06	1	34	17	Muerte no contada	Aborto	1	0	9998	9	7	477	67	6	Madre de Dios	Urbano	Capital, ciudad grande	Muy rico

mater1																			
	caseid	mmidx	mm6	mm7		mm9	mm11	mm13	mm14	mm15	mmc1	mmc2	v001	v002	v003	v101	v102	v103	v190
43	047905001	02	1	22	20	Murió durante el embarazo	Embarazo	1	0	9998	8	8	479	50	2	Madre de Dios	Urbano	Capital, ciudad grande	Medio
44	048403901	05	3	7	23	Muerte no contada	Aborto	3	2	9998	7	3	484	39	5	Madre de Dios	Urbano	Campo	Pobre
45	048403901	08	3	7	23	Muerte no contada	Aborto	3	2	9998	7	6	484	39	8	Madre de Dios	Urbano	Campo	Pobre
46	050606301	01	3	30	16	Murió durante el embarazo	Aborto	2	2	9998	7	7	506	63	1	Moquegua	Urbano	Pueblo	Pobre
47	053706801	04	1	NA	42	Muerte no contada	Aborto	1	0	2008	6	3	537	68	4	Pasco	Urbano	Capital, ciudad grande	Medio
48	053706802	02	1	NA	40	Muerte no contada	Aborto	1	0	2008	6	4	537	68	2	Pasco	Urbano	Capital, ciudad grande	Medio
49	054404801	02	2	36	16	Muerte no contada	Aborto	3	0	9998	12	9	544	48	2	Pasco	Rural	Campo	Rico
50	054707101	02	4	8	26	Murió durante el embarazo	Embarazo	3	3	9998	8	4	547	71	2	Pasco	Rural	Campo	Muy pobre
51	054806001	02	2	20	30	Murió durante el embarazo	Aborto	3	7	9998	7	7	548	60	2	Pasco	Rural	Campo	Muy pobre
52	055301801	02	1	25	23	Murió durante el embarazo	Parto	3	1	9998	8	2	553	18	2	Piura	Urbano	Pueblo	Pobre
53	056400201	06	1	NA	23	Murió durante el embarazo	Parto	3	1	2004	8	4	564	2	6	Piura	Rural	Campo	Muy pobre
54	056400201	10	1	NA	23	Murió durante el embarazo	Parto	3	1	2004	8	6	564	2	10	Piura	Rural	Campo	Muy pobre
55	059706401	02	2	20	18	Murió durante el embarazo	Parto	3	2	9998	2	0	597	64	2	Puno	Rural	Pueblo	Pobre
56	062021101	02	1	NA	43	Murió durante el embarazo	Parto	2	6	2004	5	3	620	211	2	San Martin	Urbano	Pueblo	Rico
57	065208401	01	2	37	20	Murió durante el embarazo	Parto	2	2	9998	11	7	652	84	1	Tacna	Rural	Pueblo	Pobre
58	066712001	03	2	NA	30	Murió durante el embarazo	Embarazo	1	2	2011	3	3	667	120	3	Tumbes	Urbano	Campo	Rico
59	070909801	02	3	33	21	Murió durante el embarazo	Parto	1	0	9998	10	8	709	98	2	Ucayali	Urbano	Capital, ciudad grande	Rico
60	071605901	02	5	NA	17	Murió durante el embarazo	Parto	1	0	2008	5	4	716	59	2	Amazonas	Urbano	Pueblo	Medio
61	072100101	02	2	27	20	Muerte no contada	Parto	2	0	9998	6	3	721	1	2	Amazonas	Urbano	Pueblo	Muy pobre
62	072401301	07	2	NA	22	Murió durante el embarazo	Parto	3	3	2010	7	2	724	13	7	Amazonas	Rural	Pueblo	Muy pobre
63	072408301	02	1	12	30	Murió durante el embarazo	Parto	3	7	9998	9	7	724	83	2	Amazonas	Rural	Campo	Muy pobre
64	072506301	02	1	9	43	Muerte no contada	Parto	2	8	9998	9	4	725	63	2	Amazonas	Rural	Campo	Muy pobre
65	072601101	04	1	13	15	Murió durante el embarazo	Parto	3	0	9998	6	5	726	11	4	Amazonas	Rural	Campo	Muy pobre
66	072601401	02	4	20	30	Muerte no contada	Embarazo	2	3	9998	8	8	726	14	2	Amazonas	Rural	Pueblo	Pobre
67	072701501	02	5	27	18	Muerte no contada	Parto	3	1	9998	13	6	727	15	2	Amazonas	Rural	Campo	Muy pobre
68	072702801	02	3	10	29	Murió durante el embarazo	Parto	3	3	9998	12	9	727	28	2	Amazonas	Rural	Campo	Muy pobre
69	072702901	02	2	24	20	Muerte no contada	Parto	3	3	9998	12	5	727	29	2	Amazonas	Rural	Campo	Muy pobre
70	073400101	02	4	NA	22	Murió durante el embarazo	Parto	2	1	2002	7	1	734	1	2	Amazonas	Rural	Campo	Muy pobre
71	073402201	02	3	NA	24	Murió durante el embarazo	Parto	2	2	2011	10	3	734	22	2	Amazonas	Rural	Campo	Muy pobre
72	075604501	04	1	NA	27	Murió durante el embarazo	Parto	2	3	2006	5	1	756	45	4	Ancash	Rural	Pueblo	Rico
73	077607901	02	1	27	28	Murió durante el embarazo	Embarazo	2	3	9998	1	1	776	79	2	Apurimac	Urbano	Ciudad	Pobre
74	078610301	02	9	22	18	Murió durante el embarazo	Parto	3	1	9998	9	7	786	103	2	Apurimac	Rural	Campo	Muy pobre
75	086102101	02	3	NA	24	Murió durante el embarazo	Parto	3	1	2002	10	3	861	21	2	Cajamarca	Rural	Campo	Pobre
76	086107401	05	3	8	28	Murió durante el embarazo	Parto	3	1	9998	10	8	861	74	5	Cajamarca	Rural	Campo	Muy pobre
77	086107501	02	3	NA	25	Murió durante el embarazo	Parto	3	1	2002	10	4	861	75	2	Cajamarca	Rural	Campo	Muy pobre
78	086209701	02	4	14	35	Murió durante el embarazo	Parto	2	4	9998	9	5	862	97	2	Cajamarca	Rural	Ciudad	Medio
79	086704101	03	1	31	21	Murió durante el embarazo	Parto	3	6	9998	9	7	867	41	3	Cajamarca	Rural	Campo	Muy pobre
80	086704101	03	5	10	33	Murió durante el embarazo	Parto	3	6	9998	9	7	867	41	3	Cajamarca	Rural	Campo	Muy pobre
81	086704301	02	1	23	30	Murió durante el embarazo	Parto	3	6	9998	5	2	867	43	2	Cajamarca	Rural	Campo	Muy pobre
82	086704301	02	3	18	28	Murió durante el embarazo	Parto	3	4	9998	5	2	867	43	2	Cajamarca	Rural	Campo	Muy pobre
83	089608301	02	6	29	25	Murió durante el embarazo	Parto	3	0	9998	8	8	896	83	2	Cusco	Urbano	Campo	Pobre
84	089803601	01	5	15	23	Murió durante el embarazo	Parto	1	2	9998	5	5	898	36	1	Cusco	Urbano	Campo	Medio

mater1																			
	caseid	mmidx	mm6	mm7	mm9	mm11	mm13	mm14	mm15	mmc1	mmc2	v001	v002	v003	v101	v102	v103	v190	
85	090300601	02	1	20	20	Murió durante el embarazo	Aborto	3	1	9998	10	3	903	6	2	Cusco Rural		Campo	Muy pobre
86	092807001	02	1	12	30	Murió durante el embarazo	Parto	3	4	9998	8	2	928	70	2	Huancavelica Rural		Campo	Muy pobre
87	093107001	02	3	11	35	Murió durante el embarazo	Parto	3	5	9998	10	8	931	70	2	Huancavelica Rural		Campo	Pobre
88	093604401	02	5	25	25	Murió durante el embarazo	Embarazo	2	2	9998	9	5	936	44	2	Huánuco Urbano		Campo	Medio
89	093800801	02	6	21	19	Murió durante el embarazo	Parto	1	0	9998	7	7	938	8	2	Huánuco Urbano	Capital, ciudad grande		Muy rico
90	093900801	02	4	10	26	Muerte no contada	Embarazo	1	0	9998	6	0	939	8	2	Huánuco Urbano		Pueblo	Rico
91	094213001	02	1	NA	23	Murió durante el embarazo	Parto	3	1	2010	5	1	942	130	2	Huánuco Rural		Campo	Medio
92	094305701	02	3	4	40	Murió durante el embarazo	Parto	2	8	9998	15	8	943	57	2	Huánuco Rural		Pueblo	Muy pobre
93	094506501	01	5	20	14	Murió durante el embarazo	Parto	3	0	9998	8	2	945	65	1	Huánuco Rural		Campo	Muy pobre
94	095203201	05	2	NA	24	Murió durante el embarazo	Aborto	1	0	2011	5	4	952	32	5	Huánuco Rural		Campo	Muy pobre
95	095203201	06	2	NA	24	Murió durante el embarazo	Aborto	1	0	2011	5	5	952	32	6	Huánuco Rural		Campo	Muy pobre
96	095203301	02	6	98	18	Murió durante el embarazo	Parto	3	3	9998	10	9	952	33	2	Huánuco Rural		Campo	Muy pobre
97	095300401	04	2	11	40	Muerte no contada	Embarazo	3	2	9998	3	3	953	4	4	Huánuco Rural		Campo	Muy pobre
98	095609701	02	1	NA	20	Murió durante el embarazo	Parto	3	0	2007	5	0	956	97	2	Ica Urbano		Campo	Pobre
99	096702801	02	4	20	23	Murió durante el embarazo	Aborto	3	1	9998	11	11	967	28	2	Ica Urbano		Campo	Pobre
100	103111201	02	4	0	22	Murió durante el embarazo	Embarazo	3	3	9998	11	2	1031	112	2	La Libertad Urbano		Campo	Muy pobre
101	103901101	02	4	1	31	Murió durante el embarazo	Parto	3	2	9998	6	6	1039	11	2	La Libertad Rural		Pueblo	Muy pobre
102	103901101	06	4	1	31	Murió durante el embarazo	Parto	3	2	9998	6	5	1039	11	6	La Libertad Rural		Pueblo	Muy pobre
103	109308601	02	5	19	25	Murió durante el embarazo	Embarazo	1	2	9998	8	5	1093	86	2	Lima Urbano		Ciudad	Muy rico
104	110007101	02	4	20	18	Murió durante el embarazo	Embarazo	1	0	9998	5	2	1100	71	2	Lima Urbano		Campo	Pobre
105	110409801	02	2	NA	22	Murió durante el embarazo	Embarazo	3	2	1991	3	3	1104	98	2	Lima Urbano		Pueblo	Muy rico
106	116506101	02	3	NA	30	Muerte no contada	Aborto	1	2	2004	9	2	1165	61	2	Loreto Urbano	Capital, ciudad grande		Muy pobre
107	117708201	02	5	10	15	Murió durante el embarazo	Embarazo	3	0	9998	5	3	1177	82	2	Loreto Rural		Campo	Muy pobre
108	117906701	04	2	22	20	Murió durante el embarazo	Embarazo	2	3	9998	9	9	1179	67	4	Loreto Rural		Pueblo	Muy pobre
109	118001401	02	3	17	22	Muerte no contada	Aborto	1	3	9998	5	2	1180	14	2	Loreto Rural		Ciudad	Muy pobre
110	119715301	02	2	24	16	Murió durante el embarazo	Aborto	1	0	9998	4	2	1197	153	2	Madre de Dios Urbano	Capital, ciudad grande		Medio
111	120407201	02	2	15	31	Murió durante el embarazo	Parto	3	6	9998	8	4	1204	72	2	Madre de Dios Rural		Campo	Pobre
112	124804601	02	1	14	25	Murió durante el embarazo	Parto	1	2	9998	9	1	1248	46	2	Pasco Urbano		Pueblo	Pobre
113	124808301	03	2	NA	29	Murió durante el embarazo	Parto	2	2	2008	9	8	1248	83	3	Pasco Urbano		Pueblo	Pobre
114	125903601	01	1	13	30	Muerte no contada	Parto	3	5	9998	3	1	1259	36	1	Pasco Rural		Pueblo	Muy pobre
115	126501701	02	6	17	28	Muerte no contada	Aborto	1	2	9998	11	10	1265	17	2	Piura Urbano		Campo	Medio
116	127203301	02	2	39	20	Murió durante el embarazo	Parto	2	0	9998	12	6	1272	33	2	Piura Urbano		Pueblo	Medio
117	127212401	02	2	39	20	Murió durante el embarazo	Parto	2	0	9998	12	10	1272	124	2	Piura Urbano		Campo	Medio
118	127602101	02	6	13	22	Murió durante el embarazo	Parto	1	0	9998	10	4	1276	21	2	Piura Urbano		Campo	Pobre
119	127602101	03	7	13	22	Murió durante el embarazo	Parto	1	0	9998	10	9	1276	21	3	Piura Urbano		Ciudad	Pobre
120	127602101	05	7	13	22	Murió durante el embarazo	Parto	1	0	9998	10	8	1276	21	5	Piura Urbano		Campo	Pobre
121	130700401	01	1	18	18	Murió durante el embarazo	Parto	3	1	9998	5	1	1307	4	1	Puno Rural		Campo	Muy pobre
122	131710001	01	2	28	20	Murió durante el embarazo	Parto	2	0	9998	8	4	1317	100	1	San Martin Urbano		Campo	Pobre
123	135005401	02	2	38	19	Muerte no contada	Parto	1	5	9998	5	5	1350	54	2	Tacna Urbano		Ciudad	Muy rico
124	135209201	02	3	38	21	Muerte no contada	Parto	1	0	9998	6	6	1352	92	2	Tacna Urbano		Ciudad	Rico
125	137612801	02	1	14	28	Murió durante el embarazo	Embarazo	1	1	9998	8	4	1376	128	2	Tumbes Urbano	Capital, ciudad grande		Medio
126	137811901	03	2	NA	30	Murió durante el embarazo	Aborto	1	6	1989	11	7	1378	119	3	Tumbes Urbano		Campo	Medio

## 2. Base de datos euro.sav

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V11_recod
1	Belgium	40,90	50,00	74,00	73,00	58,80	61,50	1,80	1,81	1,85	1957	Sí
2	Bulgaria	50,80	56,10	69,90	66,00	60,40	59,80	1,38	1,51	1,48	2007	No
3	Czech Republic	33,30	41,80	80,40	79,90	61,80	61,70	1,33	1,43	1,50	2004	No
4	Denmark	46,40	49,00	83,80	79,00	74,80	72,40	1,85	1,75	1,89	1973	Sí
5	Germany	30,00	33,90	77,20	81,40	65,00	71,10	1,33	1,36	1,38	1957	Sí
6	Estonia	58,20	59,70	79,50	73,50	72,50	67,80	1,55	1,52	1,65	2004	No
7	Ireland	32,70	33,70	83,40	68,20	63,30	59,40	1,92	2,05	2,10	1973	Sí
8	Greece	5,30	7,40	80,30	71,10	51,20	48,60	1,40	1,42	1,51	1981	Sí
9	Spain	28,40	37,40	80,70	67,60	56,40	55,50	1,37	1,36	1,46	1986	Sí
10	France	50,50	55,80	74,90	73,90	63,80	64,70	2,00	2,01	2,01	1957	Sí
11	Croatia	11,00	14,00	67,60	63,20	53,70	50,90	1,38	1,40	1,46	2013	No
12	Italy	16,20	23,40	75,50	72,60	49,60	49,90	1,35	1,40	1,42	1957	Sí
13	Cyprus	5,60	16,90	86,20	79,60	65,90	67,70	1,45	1,35	1,46	2004	No
14	Latvia	43,40	44,60	78,20	67,50	69,10	65,30	1,35	1,34	1,44	2004	No
15	Lithuania	29,60	30,00	75,20	67,50	68,30	66,60	1,31	1,76	1,47	2004	No
16	Luxembourg	28,80	34,10	78,90	78,10	59,40	61,90	1,65	1,52	1,61	1957	Sí
17	Hungary	35,60	42,30	69,90	66,80	55,70	54,90	1,34	1,23	1,35	2004	No
18	Malta	22,30	22,70	79,20	78,90	35,40	43,40	1,39	1,49	1,44	2004	No
19	Netherlands	37,10	45,30	83,50	82,60	69,00	71,40	1,72	1,76	1,77	1957	Sí
20	Austria	37,20	40,40	80,00	80,80	66,40	69,60	1,41	1,42	1,41	1995	Sí
21	Poland	18,90	21,20	67,30	72,20	53,10	57,60	1,27	1,30	1,39	2004	No
22	Portugal	31,60	42,80	79,20	73,40	66,30	64,80	1,36	1,35	1,37	1986	Sí
23	Romania	29,00	30,00	71,20	69,90	58,50	55,70	1,32	1,25	1,35	2007	No
24	Slovenia	47,20	56,80	76,30	71,80	66,50	64,80	1,31	1,56	1,53	2004	No
25	Slovakia	27,50	34,00	74,60	72,50	57,50	57,40	1,24	1,45	1,32	2004	No
26	Finland	40,50	40,90	76,30	75,60	71,50	71,90	1,84	1,83	1,85	1995	Sí
27	Sweden	55,50	54,30	81,70	82,10	75,80	76,50	1,85	1,90	1,91	1995	Sí
28	United Kingdom	43,70	47,30	82,00	79,40	68,60	67,90	1,84	1,96	1,96	1973	Sí

Donde:

Nombre	Etiqueta	Nombre	Etiqueta
V1	País	V7	Tasa de ocupación femenina 2011
V2	Hijos fuera del matrimonio 2006	V8	Tasa de fecundidad 2006
V3	Hijos fuera del matrimonio 2011	V9	Tasa de fecundidad 2011
V4	Tasa de ocupación masculina 2006	V10	Tasa de fecundidad 2008
V5	Tasa de ocupación masculina 2011	V11	Año de incorporación a la UE
V6	Tasa de ocupación femenina 2006	V12	Siglo de incorporación