

The R logo is a large, stylized graphic in the background of the slide. It consists of a blue and yellow geometric shape that resembles a stylized 'R' or a series of stacked blocks. The logo is positioned on the left side of the slide, behind the main title.

ANÁLISIS DE DATO CON R



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



Pasaje Hernán Velarde 285 Lima.

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

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

Teléfonos: 433-3127 - 332-4650

Centro Andino de Formación y Capacitación en Estadística

Cursos Especializados en Estadística e Informática

**Recuerda siempre nuestro correo
para cualquier consulta**



campusvirtual@inei.gob.pe

CURSO BASICO DE “R”



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).

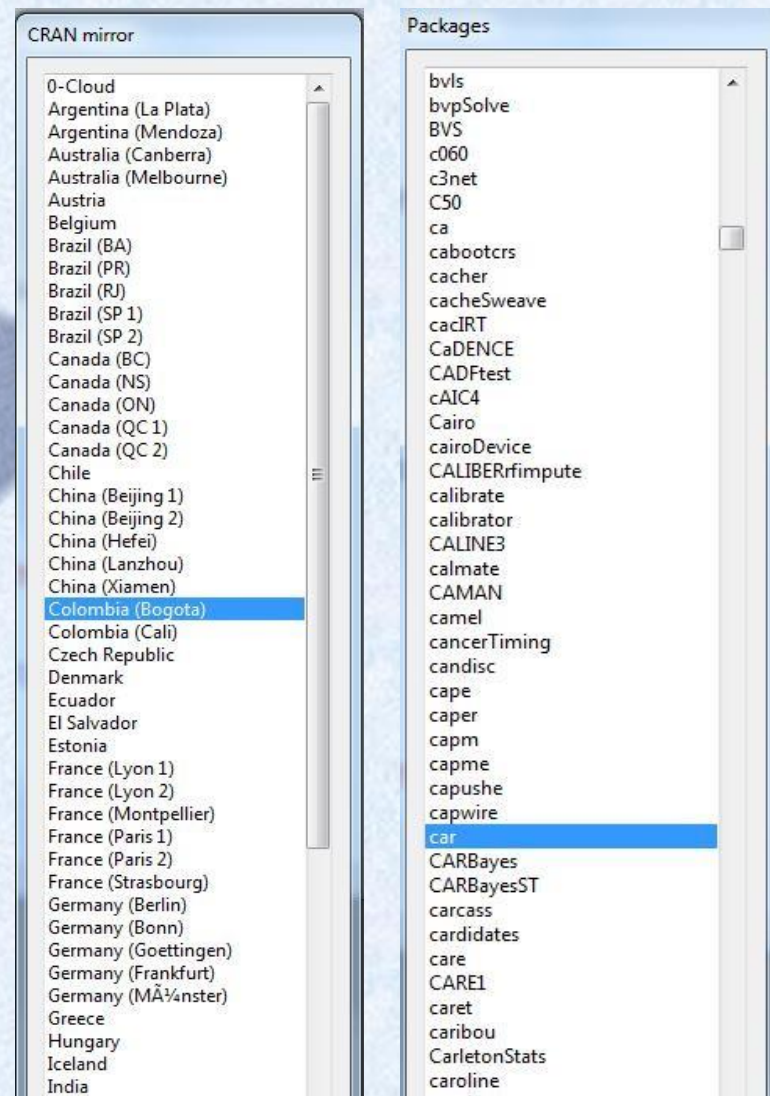
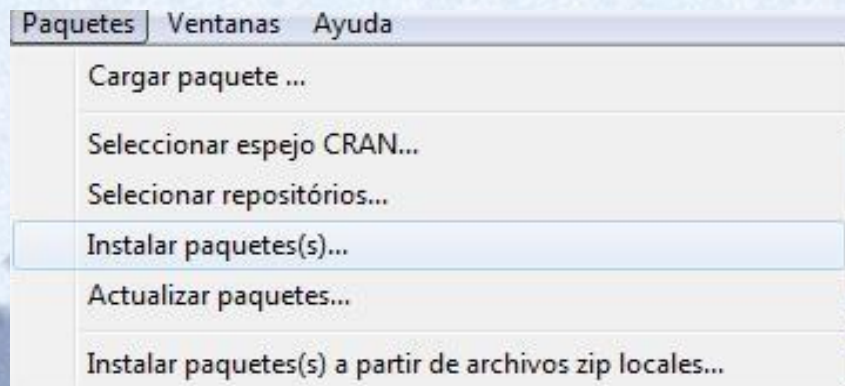
Para activar la función **`recode()`** necesitamos seguir los siguientes pasos:

- Instalar el paquete de algún MIRROR cercano a nuestro país, por ejemplo Colombia(Bogota). (No viene en el paquete básico)
- Una vez instalado, hay que cargar el **paquete car**, de la siguiente manera:

Desplegamos el menú **Paquetes** → **Cargar paquete...**

En la ventana emergente “Select one”, seleccionamos **car**, Finalmente hacemos clic sobre el botón OK

Recodificar una variable numérica usando la función recode() o Recode()



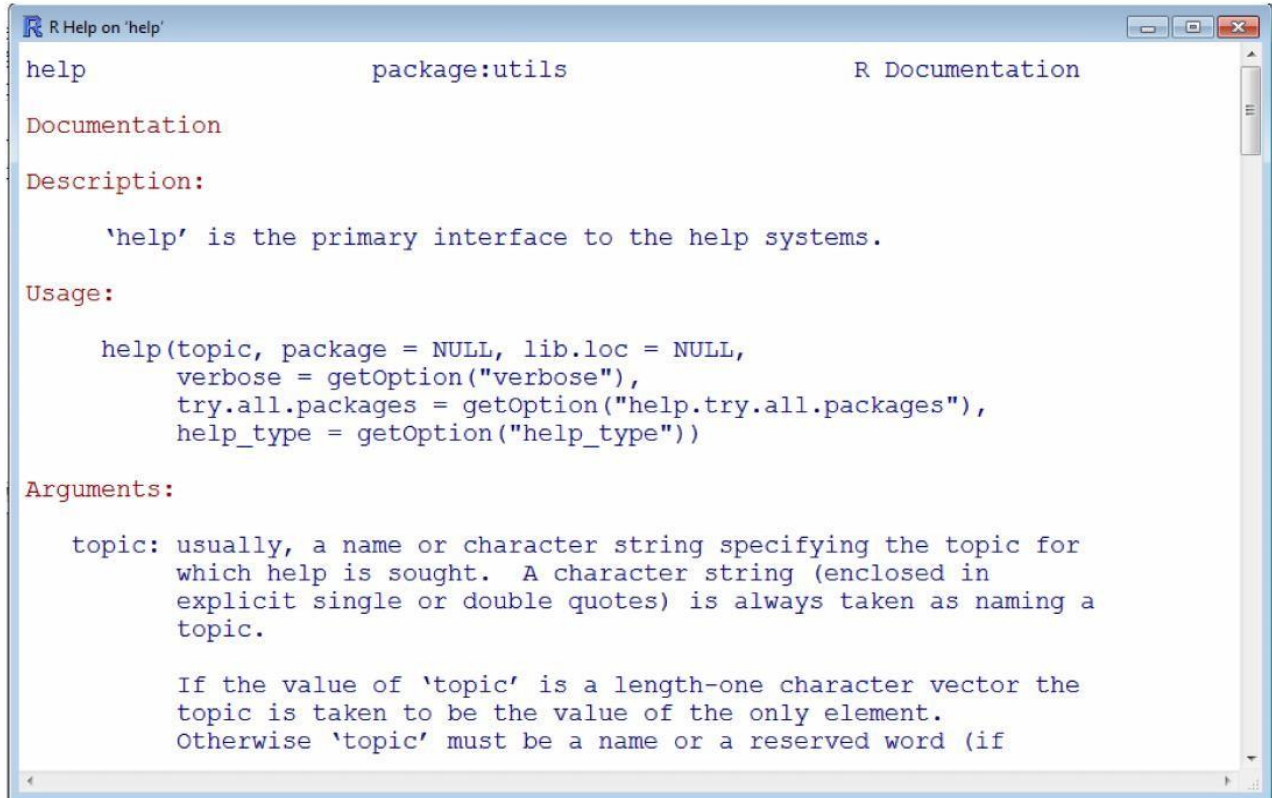
La Ayuda en R

¿Cómo se pide ayuda en R?

- `help()`,?
- `help.search("")`
- `help.start()`

La Ayuda en R

```
> help()
```



The screenshot shows the R Help window titled "R Help on 'help'". The window displays the following information:

```
help                                package:utils                        R Documentation

Documentation

Description:

  'help' is the primary interface to the help systems.

Usage:

  help(topic, package = NULL, lib.loc = NULL,
        verbose = getOption("verbose"),
        try.all.packages = getOption("help.try.all.packages"),
        help_type = getOption("help_type"))

Arguments:

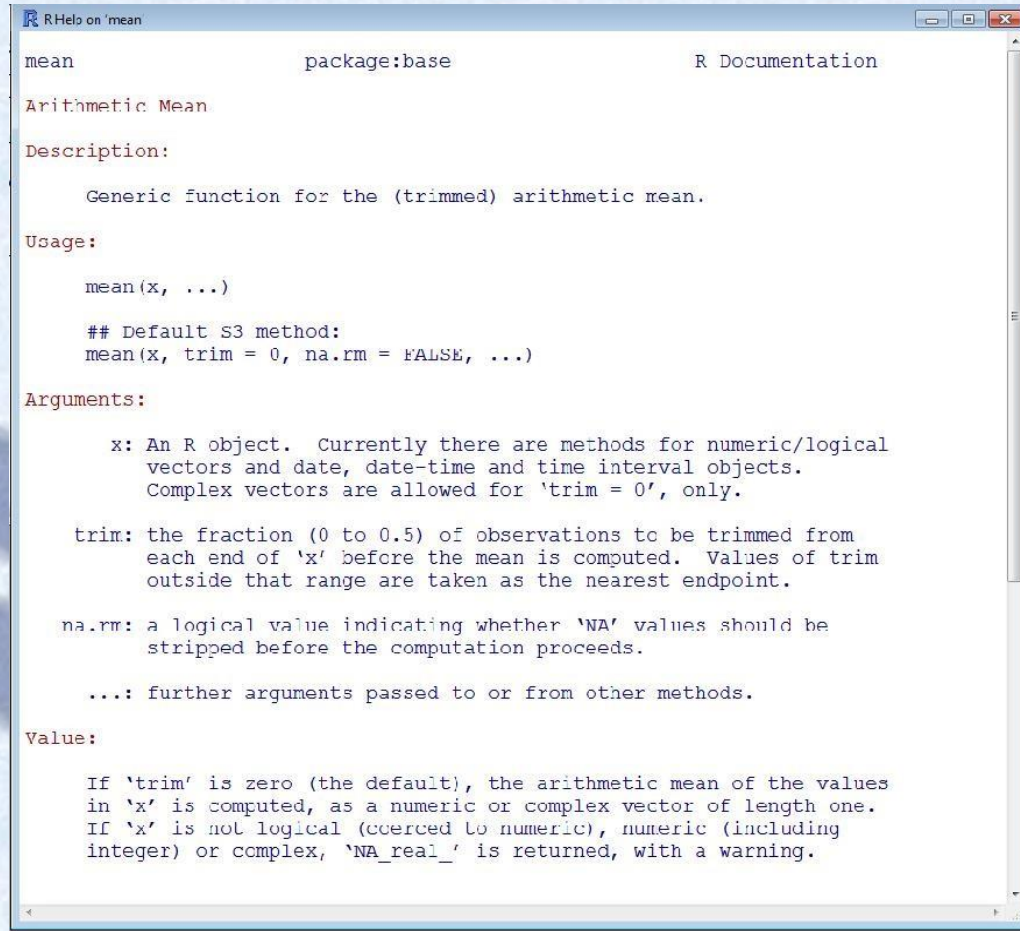
  topic: usually, a name or character string specifying the topic for
         which help is sought.  A character string (enclosed in
         explicit single or double quotes) is always taken as naming a
         topic.

         If the value of 'topic' is a length-one character vector the
         topic is taken to be the value of the only element.
         Otherwise 'topic' must be a name or a reserved word (if
```


La Ayuda en R

Buscar algo específico

```
> help(mean)
```



```
R Help on 'mean'
```

mean package:base R Documentation

Arithmetic Mean

Description:

Generic function for the (trimmed) arithmetic mean.

Usage:

```
mean(x, ...)
```

Default S3 method:

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments:

x: An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex vectors are allowed for 'trim = 0', only.

trim: the fraction (0 to 0.5) of observations to be trimmed from each end of 'x' before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.

na.rm: a logical value indicating whether 'NA' values should be stripped before the computation proceeds.

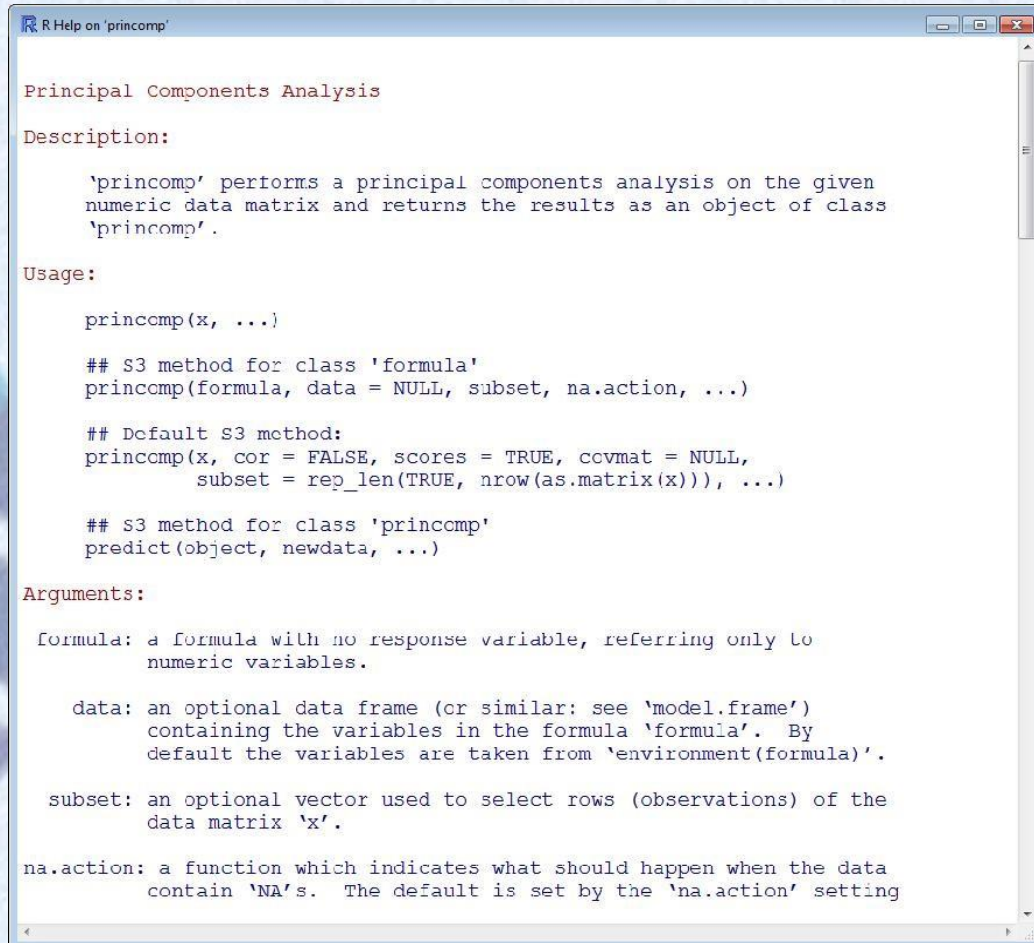
...: further arguments passed to or from other methods.

Value:

If 'trim' is zero (the default), the arithmetic mean of the values in 'x' is computed, as a numeric or complex vector of length one. If 'x' is not logical (coerced to numeric), numeric (including integer) or complex, 'NA_real_' is returned, with a warning.

La Ayuda en R

```
> help(princomp)
```



```
R Help on 'princomp'
```

Principal Components Analysis

Description:

'princomp' performs a principal components analysis on the given numeric data matrix and returns the results as an object of class 'princomp'.

Usage:

```
princomp(x, ...)
```

```
## S3 method for class 'formula'
```

```
princomp(formula, data = NULL, subset, na.action, ...)
```

```
## Default S3 method:
```

```
princomp(x, cor = FALSE, scores = TRUE, covmat = NULL,
```

```
        subset = rep_len(TRUE, nrow(as.matrix(x))), ...)
```

```
## S3 method for class 'princomp'
```

```
predict(object, newdata, ...)
```

Arguments:

formula: a formula with no response variable, referring only to numeric variables.

data: an optional data frame (or similar: see 'model.frame') containing the variables in the formula 'formula'. By default the variables are taken from 'environment(formula)'.

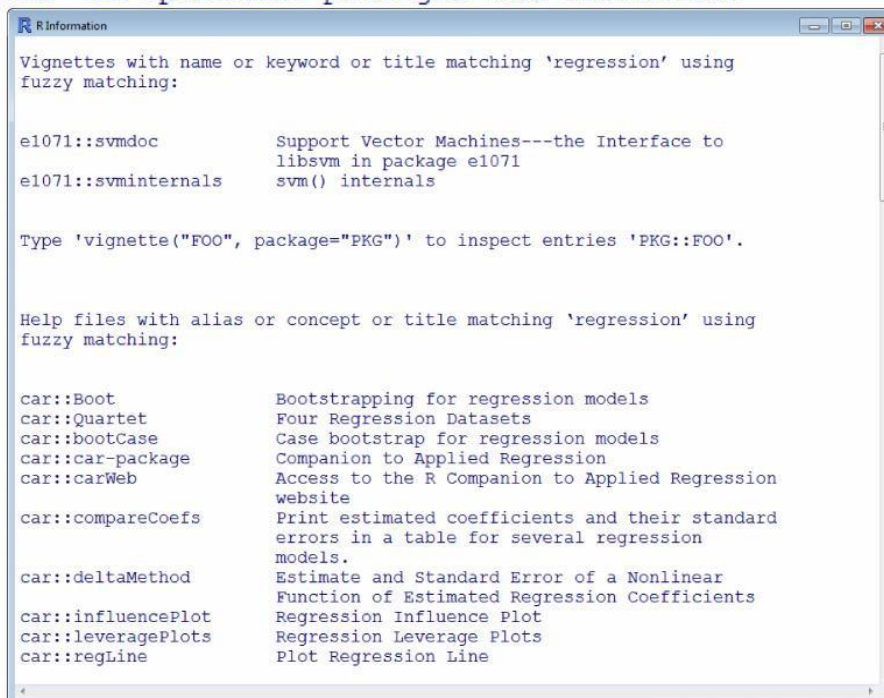
subset: an optional vector used to select rows (observations) of the data matrix 'x'.

na.action: a function which indicates what should happen when the data contain 'NA's. The default is set by the 'na.action' setting

La Ayuda en R

Buscar ayuda sobre un concepto estadístico

```
> help(regression)
No documentation for 'regression' in specified packages and libraries:
you could try '??regression'
> help.search("regression")
```



R Information

Vignettes with name or keyword or title matching 'regression' using fuzzy matching:

e1071::svmdoc	Support Vector Machines---the Interface to libsvm in package e1071
e1071::svminternals	svm() internals

Type 'vignette("FOO", package="PKG")' to inspect entries 'PKG::FOO'.

Help files with alias or concept or title matching 'regression' using fuzzy matching:

car::Boot	Bootstrapping for regression models
car::Quartet	Four Regression Datasets
car::bootCase	Case bootstrap for regression models
car::car-package	Companion to Applied Regression
car::carWeb	Access to the R Companion to Applied Regression website
car::compareCoefs	Print estimated coefficients and their standard errors in a table for several regression models.
car::deltaMethod	Estimate and Standard Error of a Nonlinear Function of Estimated Regression Coefficients
car::influencePlot	Regression Influence Plot
car::leveragePlots	Regression Leverage Plots
car::regLine	Plot Regression Line

La Ayuda en R

Equivalencias

```
> ?mean
```

```
> help(mean)
```

```
> ??regression
```

```
> help.search("regression")
```


SCRIPTS

¿Qué son los *scripts*?

- Los *scripts* son la forma más recomendable de interactuar con R
 - Guardamos registro de todo lo que hacemos
 - Podemos reutilizarlo más adelante
- Los *scripts* facilitan la colaboración entre investigadores: basta con enviar una base de datos y un *script*
- Para escribir un *script* podemos utilizar cualquier editor de texto

SCRIPTS

¿Cómo trabajar con scripts?

- Mediante el comando `source()`
- Mediante “copia-pegar”

Atención

- Para que los *scripts* sean útiles es vital “comentarlos” (#)

ÁREA DE TRABAJO (WORKSPACE)

Todos los objetos que se crean en R se almacenan en el área de trabajo (*workspace*)

- Podemos guardar el *workspace*: `save.image()`
- Podemos recuperar un *workspace* previamente guardado:
`load()`
- Podemos ver qué objetos se encuentran en el *workspace*: `ls()`
- Podemos eliminar objetos del *workspace*: `rm()`

ÁREA DE TRABAJO (WORKSPACE)

```
[Previously saved workspace restored]
```

```
> x<-c(1,2,3,4,5)
```

```
> #Vemos los objetos creados en el área de trabajo
```

```
> ls()
```

```
[1] "test"          "test.csv"      "test.csv1"     "test.missing" "x"
```

```
[6] "y"
```

```
> x
```

```
[1] 1 2 3 4 5
```

```
> y
```

```
[1] 0 3 6 9 12 15
```

```
> test
```

	prgtype	gender	id	ses	schtyp	level
1	general	0	70	4	1	1
2	vocati	1	121	4	2	1
3	general	0	86	4	3	1
4	vocati	0	141	4	3	1
5	academic	0	172	4	2	1
6	academic	0	113	4	2	1
7	general	0	50	3	2	1
8	academic	0	11	1	2	1

```
> |
```


ÁREA DE TRABAJO (WORPSPACE)

```
> #Si quiero borrar algún objeto
> rm("test.csv")
> rm("test.csv1")
> ls()
[1] "test"          "test.missing" "x"          "y"
> #Si quiero guardar mi área de trabajo
> save.image("Area1")
> #Si quiero visualizar el directorio de trabajo
> dir()
[1] "Area1"          "auto.dta"      "Clase2.R"
[4] "CostaNorte.RData" "Enaho01-2013-200.dta" "Enaho2013400d.dat"
[7] "Enaho2013400p.csv" "Enaho2013400p.sav" "Enaho2013400p.xlsx"
[10] "Enaho2013400p0.csv" "hogares.csv"     "hogares.dat"
[13] "hogares.sav"      "hogares.xls"     "hogares.xlsx"
[16] "Importar.R"       "ImportarSTATA.R" "mujeres.dta"
[19] "scan.txt"         "SumariaM.csv"    "SumariaM.sav"
[22] "test.csv"         "test.txt"        "test_missing.txt"
[25] "test_missing_comma.txt"
> rm("test")
> rm("test.missing")
> rm("x")
> ls()
[1] "y"
> rm("y")
> ls()
character(0)
> #Llamo a area de trabajo guardada
> load("Area1")
> ls()
[1] "test"          "test.missing" "x"          "y"
> x
[1] 1 2 3 4 5
```

ARCHIVO HISTÓRICO

- Todos los comandos que vamos ejecutando en R (también los que producen errores) se van guardando en un *archivo histórico*
 - Podemos desplazarnos entre los comandos anteriores con *flecha_arriba* y *flecha_abajo*
 - Podemos guardar el archivo histórico: `savehistory(file="")`
 - Podemos recuperar archivos históricos:
`loadhistory(file="")`
- El archivo histórico es muy útil como punto de partida para la elaboración de un *script*

ARCHIVO HISTÓRICO

```
> x
[1] 1 2 3 4 5
> #Hallo el promedio
> mean(x)
[1] 3
> #Hallo la medianan
> median(x)
[1] 3
> #Hallo la desviación estandar
> std(x)
Error: no se pudo encontrar la función "std"
> sd(x)
[1] 1.581139
> #Verifico el directorio de trabajo
> getwd()
[1] "D:/School/R/Escuela/Datos"
> #Guardo mi archivo histórico
> savehistory("Historia4")
> dir()
[1] "Area1" "auto.dta" "Clase2.R"
[4] "CostaNorte.RData" "Enaho01-2013-200.dta" "Enaho2013400d.dat"
[7] "Enaho2013400p.csv" "Enaho2013400p.sav" "Enaho2013400p.xlsx"
[10] "Enaho2013400p0.csv" "Historia4" "hogares.csv"
[13] "hogares.dat" "hogares.sav" "hogares.xls"
[16] "hogares.xlsx" "Importar.R" "ImportarSTATA.R"
[19] "mujeres.dta" "scan.txt" "SumariaM.csv"
[22] "SumariaM.sav" "test.csv" "test.txt"
[25] "test_missing.txt" "test_missing_comma.txt"
```

ARCHIVO HISTÓRICO

Veamos el archivo histórico con el explorador de archivos

```
Historia4: Doc de notas
Archivo Edición Formato Ver Ayuda
help()
help(mean)
help(princomp)
help(regression)
help.search("regression")
?mean
??regression
x<-c(1,2,3,4,5)
#Vemos los objetos creados en el área de trabajo
ls()
x
y
test
#Si quiero borrar algún objeto
rm("test.csv")
rm("test.csv1")
ls()
#Si quiero guardar mi área de trabajo
save.image("Areal")
#Si quiero visualizar el directorio de trabajo
dir()
rm("test")
rm("test.missing")
rm("x")
ls()
rm("y")
ls()
#Llamo a area de trabajo guardada
load("Areal")
ls()
x
X
X< C(1,2,3,4,5)
x
#Hallo el promedio
mean(x)
#Hallo la medianan
mediar(x)
Hallo la desviación estandar
```


Funciones a cargar en R

Hay tres niveles de funciones en R

- Las que están accesibles por defecto: ejemplos `mean()`, `var()`, `cor()`
- Las que están *instaladas* pero no *inmediatamente accesibles*
- Las que no están instaladas: hay miles disponibles en internet

Funciones a cargar en R

```
> x
[1] 1 2 3 4 5
> ?source
> mean(x)
[1] 3
> y<-c(2,3,4,6,7)
> cor(x,y)
[1] 0.9912407
> #Quiero crear una distribución normal multivariante
> #Tengo información que el comando es mvrnorm
> ?mvrnorm
No documentation for 'mvrnorm' in specified packages and libraries:
you could try '??mvrnorm'
> ??mvrnorm
```

R Information

Help files with alias or concept or title matching 'mvrnorm' using fuzzy matching:

MASS::mvrnorm	Simulate from a Multivariate Normal Distribution
---------------	--

Type '?PKG::FOO' to inspect entries 'PKG::FOO', or 'TYPE?PKG::FOO' for entries like 'PKG::FOO-TYPE'.

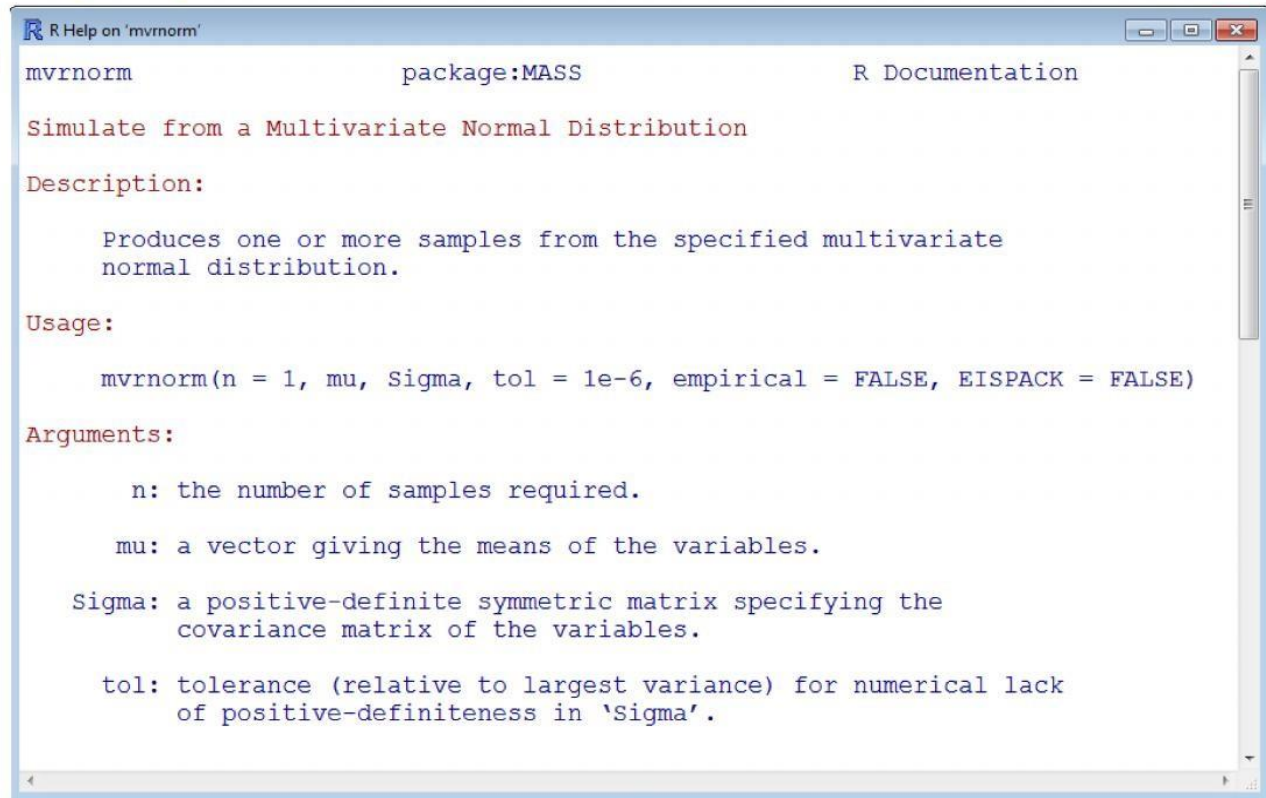
Funciones a cargar en R

```
> installed.packages()
```

base	"base"	"C:/Program Files/R/R-3.1.1/library"
boot	"boot"	"C:/Program Files/R/R-3.1.1/library"
class	"class"	"C:/Program Files/R/R-3.1.1/library"
cluster	"cluster"	"C:/Program Files/R/R-3.1.1/library"
codetools	"codetools"	"C:/Program Files/R/R-3.1.1/library"
compiler	"compiler"	"C:/Program Files/R/R-3.1.1/library"
datasets	"datasets"	"C:/Program Files/R/R-3.1.1/library"
foreign	"foreign"	"C:/Program Files/R/R-3.1.1/library"
graphics	"graphics"	"C:/Program Files/R/R-3.1.1/library"
grDevices	"grDevices"	"C:/Program Files/R/R-3.1.1/library"
grid	"grid"	"C:/Program Files/R/R-3.1.1/library"
KernSmooth	"KernSmooth"	"C:/Program Files/R/R-3.1.1/library"
lattice	"lattice"	"C:/Program Files/R/R-3.1.1/library"
MASS	"MASS"	"C:/Program Files/R/R-3.1.1/library"
Matrix	"Matrix"	"C:/Program Files/R/R-3.1.1/library"
methods	"methods"	"C:/Program Files/R/R-3.1.1/library"
mgcv	"mgcv"	"C:/Program Files/R/R-3.1.1/library"
nlme	"nlme"	"C:/Program Files/R/R-3.1.1/library"
nnet	"nnet"	"C:/Program Files/R/R-3.1.1/library"
parallel	"parallel"	"C:/Program Files/R/R-3.1.1/library"
rpart	"rpart"	"C:/Program Files/R/R-3.1.1/library"
spatial	"spatial"	"C:/Program Files/R/R-3.1.1/library"
splines	"splines"	"C:/Program Files/R/R-3.1.1/library"
stats	"stats"	"C:/Program Files/R/R-3.1.1/library"
stats4	"stats4"	"C:/Program Files/R/R-3.1.1/library"
survival	"survival"	"C:/Program Files/R/R-3.1.1/library"
tcltk	"tcltk"	"C:/Program Files/R/R-3.1.1/library"
tools	"tools"	"C:/Program Files/R/R-3.1.1/library"
translations	"translations"	"C:/Program Files/R/R-3.1.1/library"
utils	"utils"	"C:/Program Files/R/R-3.1.1/library"

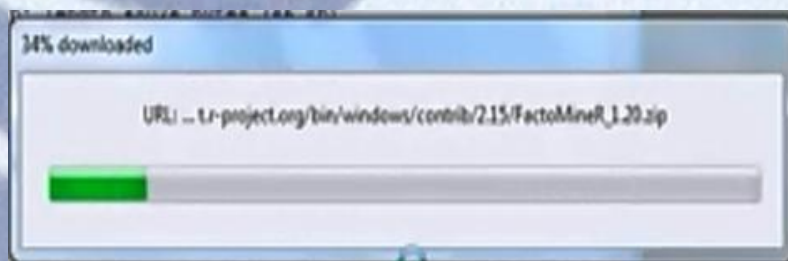
Funciones a cargar en R

```
> require(MASS)
Loading required package: MASS
> ?mvrnorm
> library(MASS)
```



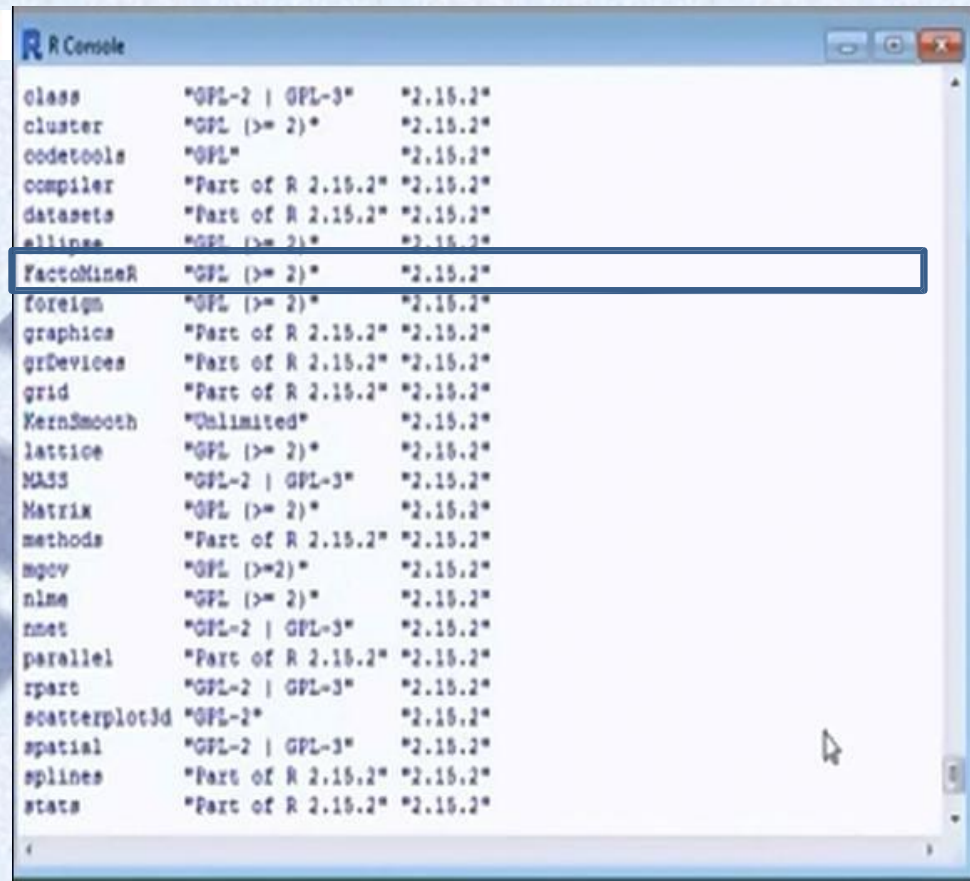
Funciones a Instalar en R

```
> #Imaginemos que queremos utilizar el método de componentes principales
> #Sobre los vectores x e y
> x
[1] 1 2 3 4 5
> y
[1] 2 3 4 6 7
> #Solicitamos a R ayuda sobre PCA (Análisis de componentes principales)
> ?PCA
No documentation for 'PCA' in specified packages and libraries:
you could try '??PCA'
> install.packages("FactoMiner")
Installing package into 'C:/Users/lulloa/Documents/R/win-library/3.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
```



Funciones a Instalar en R

```
> installed.packages()
```

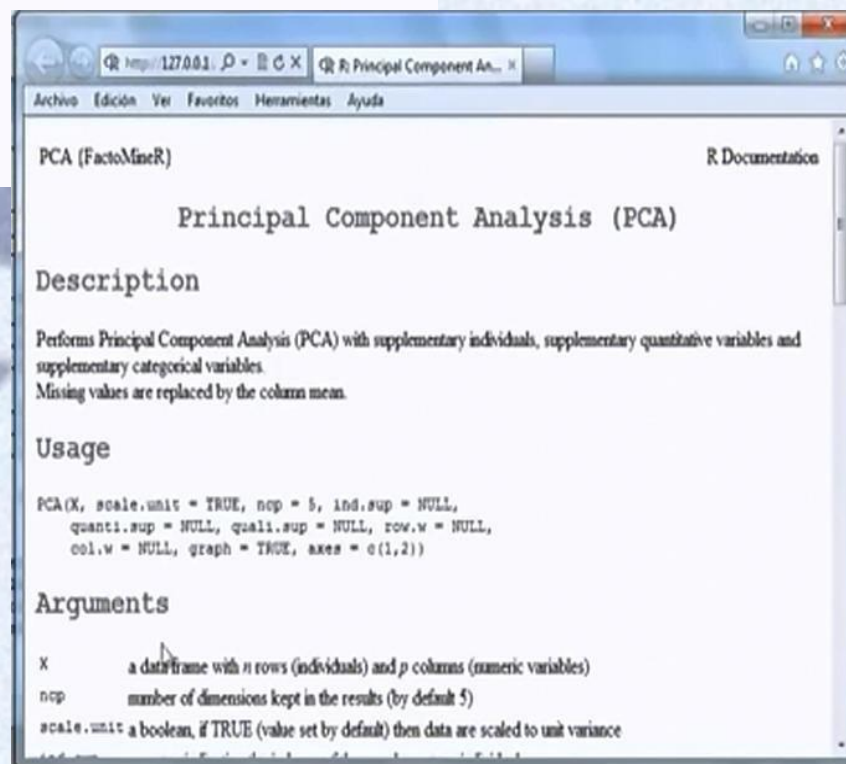


R Console

class	"GPL-2 GPL-3"	"2.15.2"
cluster	"GPL (>= 2)"	"2.15.2"
codetools	"GPL"	"2.15.2"
compiler	"Part of R 2.15.2"	"2.15.2"
datasets	"Part of R 2.15.2"	"2.15.2"
elliptic	"GPL (>= 2)"	"2.15.2"
FactoMineR	"GPL (>= 2)"	"2.15.2"
foreign	"GPL (>= 2)"	"2.15.2"
graphics	"Part of R 2.15.2"	"2.15.2"
grDevices	"Part of R 2.15.2"	"2.15.2"
grid	"Part of R 2.15.2"	"2.15.2"
KernSmooth	"Unlimited"	"2.15.2"
lattice	"GPL (>= 2)"	"2.15.2"
MASS	"GPL-2 GPL-3"	"2.15.2"
Matrix	"GPL (>= 2)"	"2.15.2"
methods	"Part of R 2.15.2"	"2.15.2"
mgcv	"GPL (>= 2)"	"2.15.2"
nme	"GPL (>= 2)"	"2.15.2"
nnet	"GPL-2 GPL-3"	"2.15.2"
parallel	"Part of R 2.15.2"	"2.15.2"
rpart	"GPL-2 GPL-3"	"2.15.2"
scatterplot3d	"GPL-2"	"2.15.2"
spatial	"GPL-2 GPL-3"	"2.15.2"
splines	"Part of R 2.15.2"	"2.15.2"
stats	"Part of R 2.15.2"	"2.15.2"

Funciones a Instalar en R

```
> ?PCA
No documentation for 'PCA' in specified packages and libraries:
you could try '??PCA'
> library(FactoMineR)
Loading required package: ellipse
Loading required package: lattice
Loading required package: cluster
Loading required package: scatterplot3d
> ?PCA
```



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).

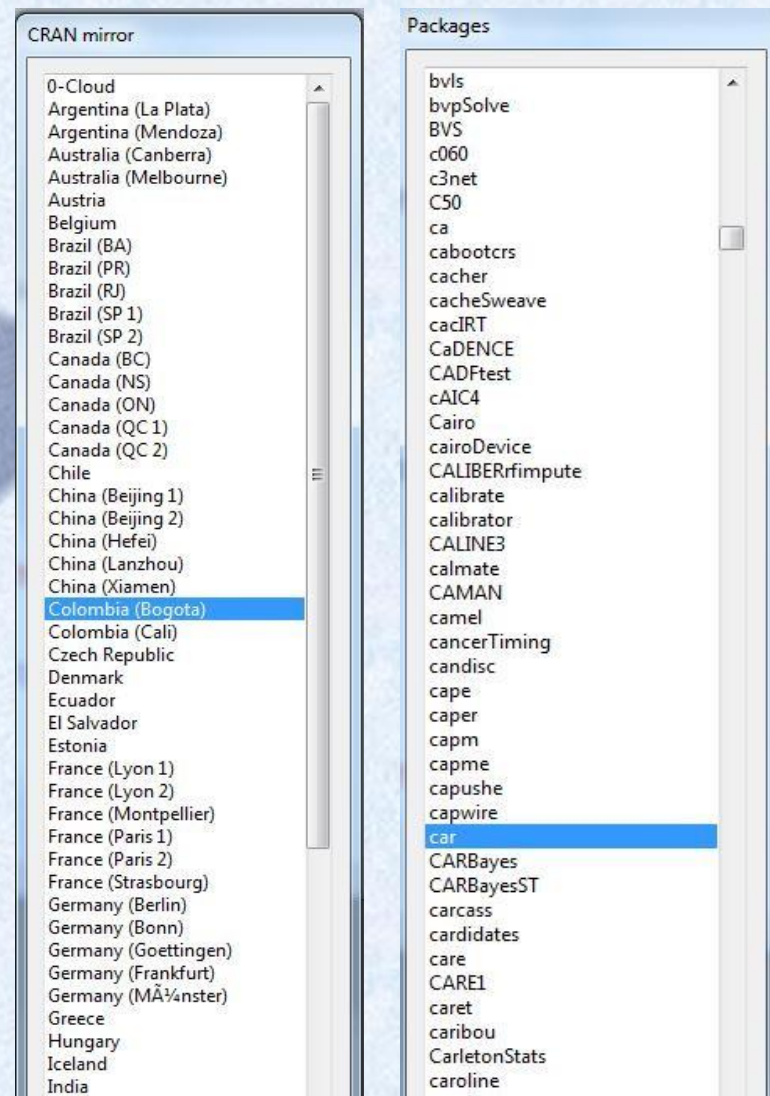
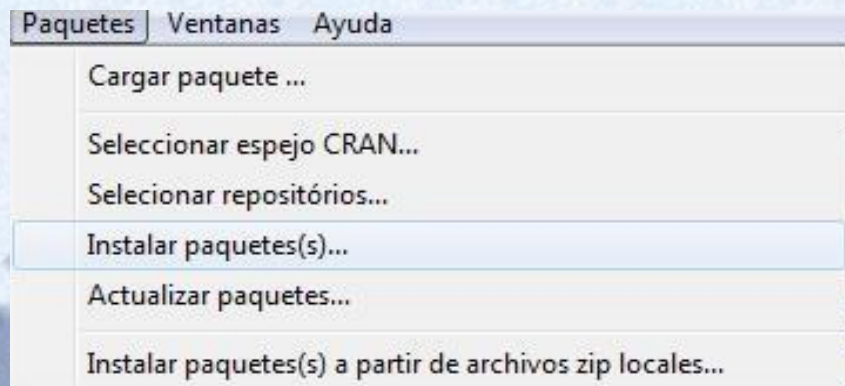
Para activar la función **`recode()`** necesitamos seguir los siguientes pasos:

- Instalar el paquete de algún MIRROR cercano a nuestro país, por ejemplo Colombia(Bogota). (No viene en el paquete básico)
- Una vez instalado, hay que cargar el **paquete car**, de la siguiente manera:

Desplegamos el menú **Paquetes** → **Cargar paquete...**

En la ventana emergente “Select one”, seleccionamos **car**, Finalmente hacemos clic sobre el botón OK

Recodificar una variable numérica usando la función recode() o Recode()



Clases de objetos en R

Todo en R son objetos; los datos también, pero...

- ¿qué tipos de objetos existen en R? : función `class()`

```
> x
[1] 1 2 3 4 5
> class(x)
[1] "numeric"
> v<-c("Saul", "Rosa", "Franklin", "Fidel")
> class(v)
[1] "character"
> is.numeric(x)
[1] TRUE
> is.numeric(v)
[1] FALSE
> is.character(v)
[1] TRUE
```


Clases de objetos en R

Existen muchas clases de datos en R. Veremos:

- números (*numeric*)
- cadenas (*character*)
- factores (*factor*)
- valores lógicos (*logical*)

Existen una serie de funciones para trabajar con clases.

Funciones para clases:

- `class()`
- `is.className(): is.numeric(), is.character()`
- `as.className(): as.numeric(), as.character()`

Clases de objetos en R

Ejemplo

```
> a<-"string"  
> is.character(a)  
> a<-5  
> is.character(a)  
> is.numeric(a)  
> b <-as.character(a)  
> b  
> is.character(b)
```

```
> z<-as.character(x)  
> is.numeric(z)  
[1] FALSE  
> is.character(z)  
[1] TRUE  
> z  
[1] "1" "2" "3" "4" "5"
```


Clases de objetos en R

Operaciones con caracteres

- ¿De qué clase es?: `class()`
- ¿Es de la clase carácter?: `is.character()`
- Convertir en carácter: `as.character()`
- Concatenación de cadenas: `paste(a,b,sep='')`
- Extracción de subcadenas:
`substr(string, comienzo, longitud)`

```
> c1<-"Café"
> c2<-"con"
> c3<-"leche"
> paste(c1,c2,c3)
[1] "Café con leche"
> #Si quiero quitar el separador
> paste(c1,c2,c3, sep="")
[1] "Caféconleche"
> paste(c1,c2,c3, sep="%")
[1] "Café%con%leche"
```

Clases de objetos en R

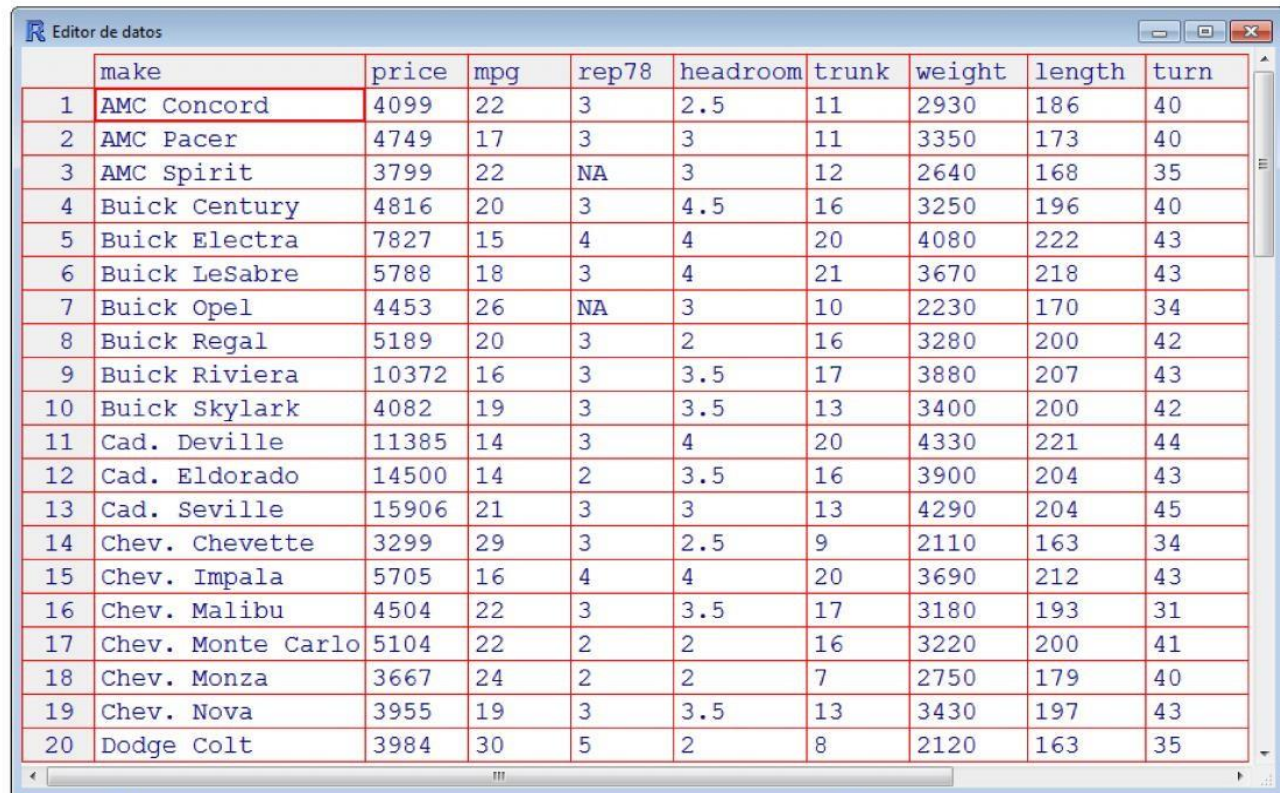
```
> d  
[1] "Café%con%leche"  
> #queremos extraer una porcion de la cadena  
> substr(d,6,8)  
[1] "con"
```


Análisis de datos en R

```
> library("foreign", lib.loc="C:/Program Files/R/R-3.1.1/library")
> dir()
[1] "Areal"                "auto.dta"                "Clase2.R"
[4] "CostaNorte.RData"     "Enaho01-2013-200.dta"    "Enaho2013400d.dat"
[7] "Enaho2013400p.csv"     "Enaho2013400p.sav"       "Enaho2013400p.xlsx"
[10] "Enaho2013400p0.csv"    "Historia4"               "hogares.csv"
[13] "hogares.dat"          "hogares.sav"            "hogares.xls"
[16] "hogares.xlsx"         "Importar.R"             "ImportarSTATA.R"
[19] "MiScript.txt"         "mujeres.dta"            "scan.txt"
[22] "SumariaM.csv"         "SumariaM.sav"           "test.csv"
[25] "test.txt"            "test_missing.txt"        "test_missing_comma.txt"
> read.dta("auto.dta")
      make price mpg rep78 headroom trunk weight length turn
1    AMC Concord  4099  22     3     2.5    11  2930   186   40
2    AMC Pacer   4749  17     3     3.0    11  3350   173   40
3    AMC Spirit  3799  22    NA     3.0    12  2640   168   35
4 Buick Century  4816  20     3     4.5    16  3250   196   40
5 Buick Electra  7827  15     4     4.0    20  4080   222   43
6 Buick LeSabre  5788  18     3     4.0    21  3670   218   43
7 Buick Opel    4453  26    NA     3.0    10  2230   170   34
8 Buick Regal   5189  20     3     2.0    16  3280   200   42
9 Buick Riviera 10372  16     3     3.5    17  3880   207   43
10 Buick Skylark 4082  19     3     3.5    13  3400   200   42
11  Cad. Deville 11385  14     3     4.0    20  4330   221   44
12  Cad. Eldorado 14500 14     2     3.5    16  3900   204   43
13  Cad. Seville 15906 21     3     3.0    13  4290   204   45
14 Chev. Chevette 3299 29     3     2.5     9  2110   163   34
```

Análisis de datos en R

```
> data1<-read.dta("auto.dta")  
> fix(data1)
```



The screenshot shows the 'Editor de datos' window in R, displaying a data table with 20 rows and 10 columns. The columns are: make, price, mpg, rep78, headroom, trunk, weight, length, and turn. The rows represent different car models, with some missing values (NA) in the 'rep78' column for rows 3 and 7.

	make	price	mpg	rep78	headroom	trunk	weight	length	turn
1	AMC Concord	4099	22	3	2.5	11	2930	186	40
2	AMC Pacer	4749	17	3	3	11	3350	173	40
3	AMC Spirit	3799	22	NA	3	12	2640	168	35
4	Buick Century	4816	20	3	4.5	16	3250	196	40
5	Buick Electra	7827	15	4	4	20	4080	222	43
6	Buick LeSabre	5788	18	3	4	21	3670	218	43
7	Buick Opel	4453	26	NA	3	10	2230	170	34
8	Buick Regal	5189	20	3	2	16	3280	200	42
9	Buick Riviera	10372	16	3	3.5	17	3880	207	43
10	Buick Skylark	4082	19	3	3.5	13	3400	200	42
11	Cad. Deville	11385	14	3	4	20	4330	221	44
12	Cad. Eldorado	14500	14	2	3.5	16	3900	204	43
13	Cad. Seville	15906	21	3	3	13	4290	204	45
14	Chev. Chevette	3299	29	3	2.5	9	2110	163	34
15	Chev. Impala	5705	16	4	4	20	3690	212	43
16	Chev. Malibu	4504	22	3	3.5	17	3180	193	31
17	Chev. Monte Carlo	5104	22	2	2	16	3220	200	41
18	Chev. Monza	3667	24	2	2	7	2750	179	40
19	Chev. Nova	3955	19	3	3.5	13	3430	197	43
20	Dodge Colt	3984	30	5	2	8	2120	163	35

Análisis de datos en R

```
> class(datos[,2])
[1] "integer"
> is.numeric(datos[,2])
[1] TRUE
> table(datos[,2])

 1    2    7    8
254 1628    2    1
> class(datos[,3])
[1] "integer"
> table(datos[,3])

 1    2    7    8
132 1751    1    1
```

```
for (i in 1:dim(datos)[2])
{
  print(colnames(datos)[i])
  print(table(datos[,i]))
}
```

Comunicación constante con la Escuela del INEI

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

Área de Educación Virtual
campusvirtual@inei.gob.pe

