

Econometría Aplicada con



```
R Console (32-bit)
Archivo Editar Misc. Ejecutar Ventanas Ayuda

> x <- c(1,2,3,4,5,6)
> y <- x^2
> print(y)
[1] 1 4 9 16 25 36
> mean(y)
[1] 15.16667
> var(y)
[1] 178.9444
> lm_1 <- lm(y ~ x)
> print(lm_1)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept) -9.3333
x             7.0000

> summary(lm_1)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept) -9.3333
x             7.0000

Residuals:
1      2      3      4      5      6
3.3333 -0.6667 -2.6667 -2.6667 -0.6667  3.3333

Coefficients:
(Intercept) Estimate Std. Error t value Pr(>|t|)
1             -9.3333      2.8441    -3.282 0.030453 *
2              7.0000      0.7303     9.585 0.000662 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.055 on 4 degrees of freedom
Multiple R-squared:  0.9583,    Adjusted R-squared:  0.9478
F-statistic: 91.87 on 1 and 4 DF,  p-value: 0.000662

> |
```



SESIÓN 4: Gestión de datos

Contenido

Introducción	4
Crear nueva variable	5
Etiquetar una variable	6
Renombrar variables	8
Recodificar variables.....	9
Ordenar datos	11
Fusión de datos.....	13
Añadir columnas	13
Añadir filas	13
Bibliografía	14
Recursos informáticos	14

Introducción

Con un conjunto de datos en R, se pueden hacer diversas modificaciones a la base de datos, como crear variables, recodificar variables y renombrarlas, todo usando solamente el código de R.

En la siguiente sesión se explicarán las diversas formas de gestionar una base de datos en dicho programa. Desde ahora se usará la GUI RStudio y se trabajará con los Scripts, a fin de que los procedimientos sean más sencillos y amigables para el alumno.

Crear nueva variable

Para la creación de nuevas variables se pueden usar dos operadores, `<-` y `=`, la sintaxis para la creación de una nueva variable en una base de datos viene dada por la siguiente expresión:

Suponiendo que se tiene una base de datos que contiene información sobre los precios de autos y sus diferentes características se quiere crear una variable que represente a los precios en miles de dólares.

```
auto$pmil = with(auto, price/1000) # Creando un variable para representar
Los miles
```

Esta sentencia indica en **auto\$pmil**, que con dicho conjunto de datos se creará una variable de nombre **pmil**, luego viene el argumento de la nueva variable con el comando **with()**, aquí se indica la data nuevamente seguida por una coma (,) y el argumento de la nueva variable a crearse, que será **price/1000**, es decir, el precio dividido entre 1000.

Para ejecutar una línea en un Script de R se debe hacer la combinación de teclas **Ctrl+Enter**, sombreando la línea de código se ejecutará dicha orden y se creará una nueva variable para la base de datos:

Para visualizar la nueva variable se debe escribir la siguiente sentencia:

```
print(pmil) # Mostrar Los valores en La consola
```

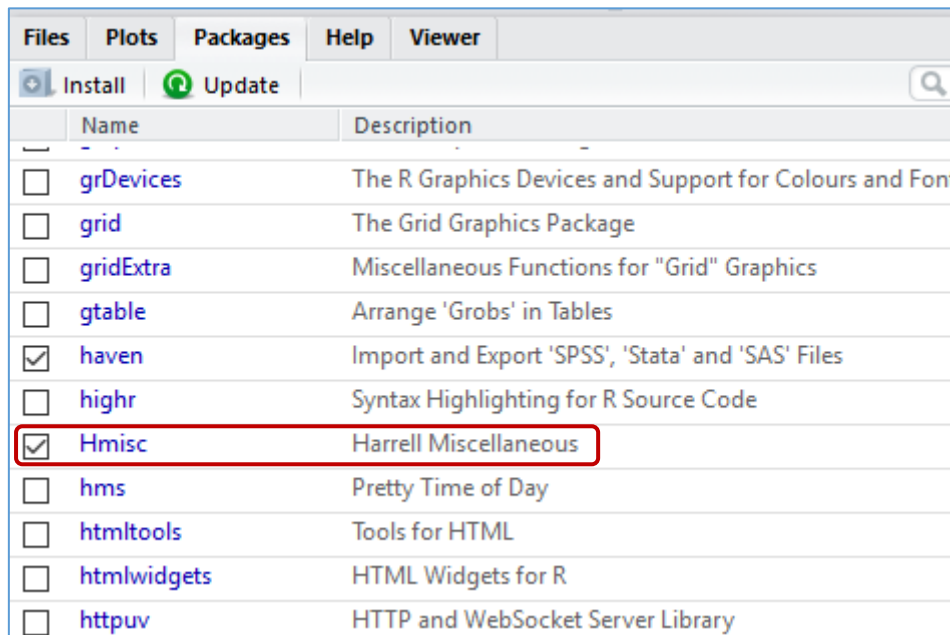
Esto mostrará los valores de la variable creada en la consola, pero antes se debe usar la función **attach()**, esta permite habilitar el conjunto de datos con el que se trabajará, es decir, carga en la memoria las variables de dataframe.

La consola mostrará lo siguiente:

```
> print(pmil) # Mostrar Los valores en La consola
[1] 4.099 4.749 3.799 4.816 7.827 5.788 4.453 5.189 10.372
4.082 11.385
[12] 14.500 15.906 3.299 5.705 4.504 5.104 3.667 3.955 3.984
4.010 5.886
[23] 6.342 4.389 4.187 11.497 13.594 13.466 3.829 5.379 6.165
4.516 6.303
[34] 3.291 8.814 5.172 4.733 4.890 4.181 4.195 10.371 4.647
4.425 4.482
[45] 6.486 4.060 5.798 4.934 5.222 4.723 4.424 4.172 9.690
6.295 9.735
[56] 6.229 4.589 5.079 8.129 4.296 5.799 4.499 3.995 12.990
3.895 3.798
[67] 5.899 3.748 5.719 7.140 5.397 4.697 6.850 11.995
attr(,"label")
[1] "Price"
attr(,"format.stata")
```

Etiquetar una variable

Para etiquetar una variable se suele hacer uso del paquete **Hmisc** y su comando **label()**, para usarlo basta con buscar en RStudio el nombre del paquete en la sección de Paquetes instalados y marcarlo:



Esto hará que automáticamente se cargue el paquete y se muestre en la consola la salida:

```
> library("Hmisc", lib.loc="~/R/win-library/3.3")
Loading required package: lattice
Loading required package: survival
Loading required package: Formula
Loading required package: ggplot2
```

Attaching package: 'ggplot2'

The following object is masked from 'auto':

mpg

Attaching package: 'Hmisc'

The following objects are masked from 'package:base':

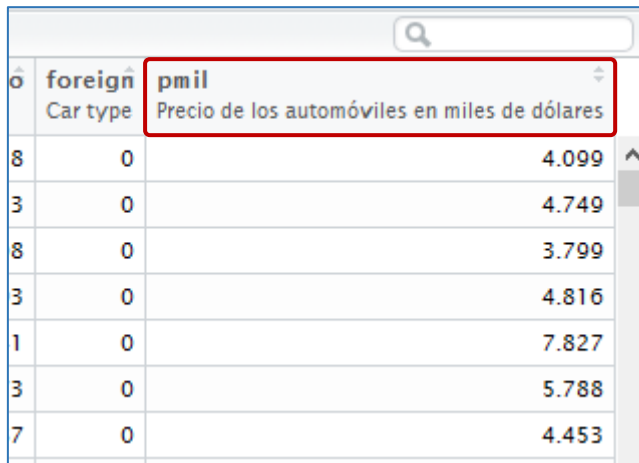
format.pval, round.POSIXt, trunc.POSIXt, units

Una vez que se ha cargado el paquete se deberá introducir la siguiente sintaxis:

```
label(auto$pmil)='Precio de los automóviles en miles de dólares'
```

Esta sintaxis indica que se quiere hacer una etiqueta para la variable **pmil** de la base de datos **auto**, luego del símbolo '=', se indicará entre comillas simples la etiqueta para dicha variable.

En la vista de la base de datos aparecerá la variable con la etiqueta que anteriormente se indicó:



id	foreign	Car type	pmil
8	0		4.099
3	0		4.749
8	0		3.799
3	0		4.816
1	0		7.827
3	0		5.788
7	0		4.453

Renombrar variables

Para renombrar variables se suele recurrir a la función **fix(df)** (edición de la tabla) sustituyendo “a mano” los nombres. Para automatizar en renombrado de variables se usa el comando **rename**; en RStudio podemos editar los dataframes haciendo clic en la descripción del objeto, en la pestaña **Environment**.

Una vez que se ha cargado el paquete **reshape**, se usará la función **rename()**, de la siguiente manera:

```
auto=rename(auto,c(length="longitud"))
```

En la vista de la base de dato se verá la variable con su nuevo nombre:

	weight	longitud	turn
ft.)	Weight (lbs.)	Length (in.)	Turn Circle (ft.)
16	1990	156	36
15	1930	155	35
15	2040	155	35
12	2160	172	36
14	3170	193	37
11	2670	175	36
9	2200	165	35

Recodificar variables

Para recodificar variables es interesante hablar, en primer lugar, de los operadores lógicos, pues a menudo se recodifica valores según una regla lógica.

Operador	Descripción
<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que
==	Exactamente igual a
!=	No igual a/que
!x	Diferente de x
x y	x o y
x & y	x e y
isTRUE(x)	Evalúa si x es una expresión verdadera

Con la base de datos de los autos se debe seguir la forma para hacer la recodificación de la variable **price**:

```
auto$pricecat[ auto$price > 10000 ] = "Precio alto"
auto$pricecat[ auto$price <= 10000 & auto$price > 6000 ] = "Precio
medio"
auto$pricecat[ auto$price <= 6000 ] = "Precio bajo"
```

Ese código indica que con la base de datos **auto**, se generará una variable llamada **pricecat**, que será las categorías de los valores de los precios, donde se considerará que un hay un precio alto cuando el valor de la variable **price** sea mayor a 10000, precio medio cuando esta contenido entre 6000 y 10000, y finalmente precio bajo cuando el valor de la variable Price sea menor o igual a 6000.

En la base de datos se podrá observar la variable con los valores de la recodificación:

in.)	gear_ratio Gear Ratio	foreign Car type	pricecat
00	2.47	0	Precio alto
00	2.47	0	Precio alto
02	2.47	0	Precio alto
40	2.73	0	Precio bajo
02	2.75	0	Precio bajo
02	2.26	0	Precio medio
50	2.43	0	Precio bajo
02	2.75	0	Precio medio

Ordenar datos

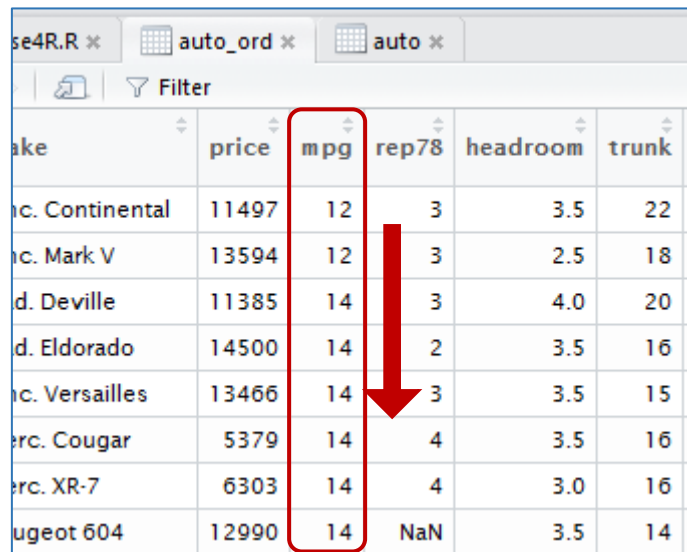
Si se quiere ordenar para mostrar solamente una variable se debe hacer uso de la función **sort()**, pero para ordenar las filas de un dataframe se debe hacer uso de la función **order()**, que genera un índice de orden, que por defecto será ascendente, aunque podrá modificarse con el signo menos al cambiar el sentido.

La sintaxis vendrá dada por:

```
auto_ord = auto[ order( auto$mpg ), ]
```

Esto indicará que se creará un nuevo conjunto de datos llamado **auto_ord**, que, basado en la información de la base **auto**, estará ordenada de acuerdo a la variable **mpg**.

En la parte de la vista de los datos se podrá visualizar la nueva base de datos creada y se notará que estará ordenada por la variable **mpg**:



make	price	mpg	rep78	headroom	trunk
ac. Continental	11497	12	3	3.5	22
ac. Mark V	13594	12	3	2.5	18
d. Deville	11385	14	3	4.0	20
d. Eldorado	14500	14	2	3.5	16
ic. Versailles	13466	14	3	3.5	15
erc. Cougar	5379	14	4	3.5	16
erc. XR-7	6303	14	4	3.0	16
ugeot 604	12990	14	NaN	3.5	14

Cuando se quiere ordenar los datos con más de una variable simplemente se indica la misma base y la variable:

```
auto_ord2 = auto[ order( auto$mpg, -auto$length ), ]
```

Esta orden le indica a R que se quiere crear un nuevo conjunto de datos llamado **auto_ord2**, que estará ordenado por la variable **mpg** de la base de datos **auto** y también por la variable **length**, pero en la segunda variable el ordenamiento será de forma descendente.

En el programa se verá el ordenamiento con el nuevo conjunto de datos:

mpg	rep78	headroom	trunk	weight	length
12	3	3.5	22	4850	233
12	3	2.5	18	4700	230
14	3	4.0	20	4300	221
14	4	3.5	16	4000	221
14	4	3.0	16	4100	217
14	2	3.5	16	3900	204

Es decir, en primer lugar, los datos estarán ordenados por **mpg**, y a su vez habrá un orden inferior por **length**.

Fusión de datos

Podemos ampliar una dataframe añadiendo variables (columnas) o casos (filas). Para esto trabajará con las funciones **merge()** y **rbind()**.

Añadir columnas

Para unir dos bases de datos de forma horizontal, se usa la función **merge()**. En la mayoría de casos las bases de datos se unirán de acuerdo a una o más variables de llave o de identificación.

La función será la siguiente:

```
total = merge(dataframeA, dataframeB, by="ID")
```

La sentencia está indicando que se quiere fusionar dos conjuntos de datos (**dataframeA** y **dataframeB**) utilizando como variable clave a **ID**.

Para fusionar de acuerdo a dos variables:

```
total = merge(dataframeA, dataframeB, by=c("ID", "Country"))
```

El código anterior es la fusión de los datos **dataframeA** y **dataframeB**, pero esta vez se unirán por dos variables, las cuales serán **ID** y **Country**.

Añadir filas

Para unir dos conjuntos de datos de forma de forma vertical, se usa la función **rbind()**. Para esto los dos conjuntos de datos deben tener las mismas variables, pero no tienen que estar necesariamente en el mismo orden.

```
total <- rbind(dataframeA, dataframeB)
```

En la unión de datos de forma vertical se debe tener en cuenta lo siguiente, antes de usar el comando **rbind()**:

- Se deben borrar las variables extra que están en la base de datos A o,
- Se deben crear variables adicionales en el conjunto de datos B y poner los valores como perdidos (NA).

Bibliografía

Recursos informáticos

Quick R - Data Management:

<http://www.statmethods.net/management/merging.html>

Universidad de Murcia - Entorno de trabajo R:

<http://www.um.es/ae/FEIR/10/#por-que-emplear-r>