



2013, Vol. 18 No 2, pp. 12-33

<http://www.uniovi.es/reunido/index.php/Rema>

Revistas de la Universidad de Oviedo, Versión electrónica

REUNIDO

Paquetes estadísticos con licencia libre (I)**Free statistical software (I)**

Carlos Carleos Artime y Norberto Corral Blanco

Departamento de Estadística e I. O. y D.M.. Universidad de Oviedo

RESUMEN

El presente artículo, el primero de una serie de trabajos, presenta un panorama de los principales paquetes estadísticos disponibles en la actualidad con licencia libre, entendiendo este concepto a partir de los grandes proyectos informáticos así autodefinidos que han cristalizado en los últimos treinta años. En esta primera entrega se presta atención fundamentalmente al programa R.

Palabras clave: paquetes estadísticos, software libre, R.

ABSTRACT

This article, the first in a series of works, presents an overview of the major statistical packages available today with free license, understanding this concept from the large computer and self-defined projects that have crystallized in the last thirty years. In this first paper we pay attention mainly to the program R.

Keywords: statistical software, free software, R.

Contacto:

Norberto Corral Blanco

Email: norbert@uniovi.es

1.- Introducción

La palabra “libre” es usada, y no siempre de manera adecuada, en múltiples campos de conocimiento así que la informática no iba a ser una excepción. La palabra es especialmente problemática porque su traducción al inglés, “free”, es aun más polisémica, e incluye el concepto “gratis”. En 1985 se fundó la Free Software Foundation (FSF) para defender una definición rigurosa del s^ofguar libre.

La propia palabra “s^ofguar” (del inglés “software”, por oposición a “hardware”, quincalla, aplicado al soporte físico de la informática) es en sí misma problemática. Para la FSF un s^ofguar es un programa de ordenador, es decir, un fichero ejecutable o una biblioteca. Para el otro “líder moral” de la informática libre, el proyecto Debian, s^ofguar es casi sinónimo de información; el significado de lo escrito en un libro, por ejemplo, sería s^ofguar.

A pesar de las diferencias sobre qué quiere decir “s^ofguar”, que tienen mucha importancia a la hora de redactar las licencias, la FSF y Debian están de acuerdo en qué significa “libre”; un s^ofguar será libre cuando todo usuario suyo esté autorizado para: (libertad 0) ejecutar el programa para cualquier objetivo; (libertad 1) modificar el programa; (libertad 2) redistribuir copias, gratis o cobrando; (libertad 3) redistribuir copias modificadas, gratis o cobrando.

Las libertades 1 y 3 implican que un programa de ordenador “libre” debe distribuirse con su código fuente. Esto ha llevado a la popularización de otra denominación, la de “programas de código abierto” (“open source”). Aunque la definición usada por la Open Source Initiative coincide con las definiciones de “libre” de la FSF y Debian, el uso de “abierto” ha terminado por ser incluso más ambiguo que el de “libre”, por lo que en general se desaconseja¹.

En este contexto, el antónimo de “libre” suele ser “privativo”, aunque Debian prefiere usar un término más preciso (“no-libre”).

Aunque se insiste mucho en que “libre” no significa “gratis” (se puede pagar por la programación de un s^ofguar libre) la popularización de internet ha supuesto que todo programa libre acabe siendo descargable... gratis. En todo caso, es imparable subrayar que el desarrollo de sistemas bien integrados (como Debian, Ubuntu...) ha sido fruto de un trabajo duro durante años en pos de la unificación de licencias y de la observación del estricto cumplimiento de las libertades antes citadas.

En los últimos treinta años la informática libre ha ido consiguiendo éxitos notables. Ya se dispone de sistemas operativos libres, de paquetes ofimáticos libres, etc. En el campo de la estadística, el proyecto R ha conseguido en pocos años convertirse en el entorno informático de referencia para las implementaciones de viejos y nuevos algoritmos.

2.- El paquete estadístico R

2.1.- Historia

Entre 1975 y 1991 se estuvo diseñando en los Laboratorios Bell un lenguaje de programación llamado S (inicial de “statistics”, imitando el nombre del exitoso lenguaje C). El objetivo era disponer de una sintaxis que pudiera expresar cómodamente las operaciones realizadas en el tratamiento y exploración de datos.

En 1988 apareció la primera versión de la implementación privativa más exitosa de S, S-Plus, que fue la más extendida durante los años noventa. Sin embargo, a finales de la década se empezó a popularizar el proyecto R.

¹ Véanse <http://www.gnu.org/philosophy/open-source-misses-the-point.es.html> y <http://www.debian.org/intro/free.es.html>.

R es una implementación de S con ligeras diferencias conceptuales. Externamente parecen el mismo lenguaje, pero R tiene un modelo de evaluación en que las definiciones de funciones anidadas tienen ámbito léxico, mientras que en S el modelo es dinámico. La consecuencia es que la programación de ciertas funciones complicadas es mucho más fácil (por natural) en R que en S; para ver ejemplos concretos, visítase <http://cran.r-project.org/doc/FAQ/R-FAQ.html#Lexical-scoping>.

El proyecto R decidió adoptar como licencia la Licencia Pública General de GNU (GPL), la misma que los proyectos GNU y Linux; en ese sentido, hay quien apunta que “R” se pronuncia como “nuestro” en inglés. Sea por su superioridad técnica, sea por su carácter libre, R se convirtió en el dialecto de S dominante y hoy día es la “lingua franca” de la computación estadística.

2.2. – Prestaciones

R es un lenguaje de programación y una implementación de un intérprete (y, desde la versión 2.13, de un compilador a código intermedio) de ese mismo lenguaje.

Por lo dicho en la sección anterior, se puede afirmar que cada nuevo algoritmo surgido en el dominio de la estadística cuenta con una implementación en R, o bien en C o Fortran enlazada con R.

Dado que R fue diseñado como un lenguaje que resultara cómodo para el análisis de datos, la mayoría de sus usuarios tienen experiencia en la programación y trabajan con R mediante ficheros de instrucciones. Esto ha provocado que los proyectos para desarrollar un interfaz gráfica de usuario no hayan llevado el mismo ritmo que el lenguaje.

Esta situación tiene consecuencias importantes en la difusión de este paquete. A pesar del enorme potencial de R para el análisis estadístico, tanto en docencia como en investigación, una buena parte de sus posibles usuarios prefiere emplear otras aplicaciones estadísticas más fáciles de utilizar, aun pagando bastante por ellas.

Los numerosos intentos de desarrollo de interfaces gráficas se listan en http://www.sciviews.org/_rgui/.

Resumiendo mucho, podríamos decir que el más exitoso hasta el momento ha sido R-Commander. Se trata de una interfaz sin muchas pretensiones cuya mayor limitación es el editor de los ficheros de datos, con un aspecto y uso poco amigable. Sin embargo, como durante mucho tiempo ha sido la única forma de usar R “con el ratón”, se han desarrollado muchas extensiones (al cierre de esta edición, hay treinta paquetes de R cuyos nombres comienzan por “RcmdrPlugin...”).

La segunda interfaz gráfica de R en importancia puede ser Rkward. Está disponible en los repositorios de Debian y Ubuntu. Su aspecto promete; frente a las salidas de texto puro para las tablas de R-Commander, Rkward genera tablas HTML. Sin embargo, le quedan muchos aspectos sin pulir o desarrollar (p.ej. no se pueden hacer transformaciones de variables desde los menús) y todavía los cambios de una versión a otra son bastante desconcertantes.

Por último, citaremos Rattle, que ha gozado de un desarrollo notable en los últimos tiempos. Sin embargo, no incluye un editor de datos, por lo que quien busque una solución completa debería decantarse por Rkward.

2.3.- R-Commander a vista de pájaro

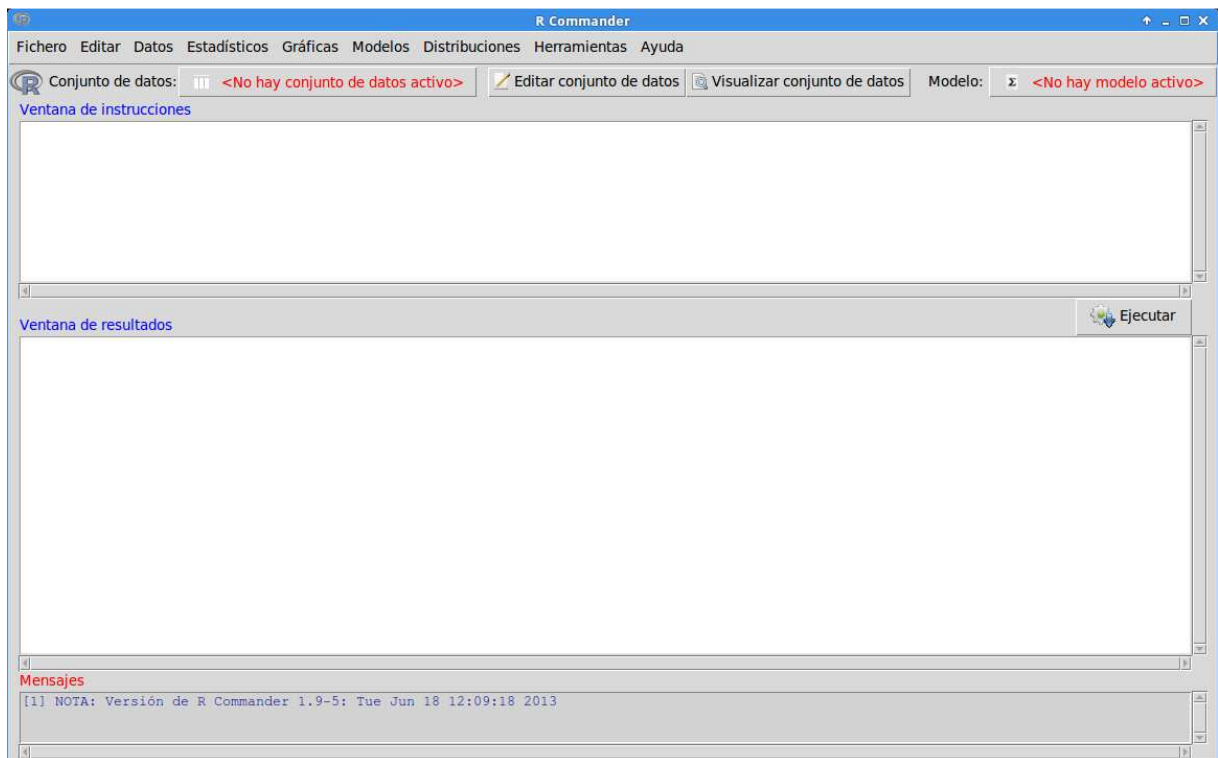
En esta sección haremos un recorrido visual por la interfaz más exitosa de R denominada R-Commander. Con sus limitaciones, el uso de R-Commander es una opción razonable para empezar a utilizar el entorno R sin tener que emplear mucho tiempo en el aprendizaje.

Antes de nada, comentemos cómo instalarlo. En sistemas operativos libres es habitual que esté empaquetado; por ejemplo, aparece en el Centro de Software de Ubuntu y, en Debian, en el paquete “r-cran-rcmdr”. Para el sistema operativo privativo más popular, Windows, lo más cómodo es usar el ejecutable preparado por la Universidad de Cádiz, R-UCA, disponible en http://knuth.uca.es/R/doku.php?id=instalacion_de_r_y_rcmdr:r-uca.

El aspecto general de la interfaz es un poco anticuado, pues se basa en la veterana biblioteca Tcl/Tk.

El acceso a R-Comander se puede hacer pinchando dos veces sobre el icono de esta aplicación o arrancando el programa R y escribiendo directamente la instrucción correspondiente:

> **library(Rcmdr)** (el símbolo > aparece por defecto)



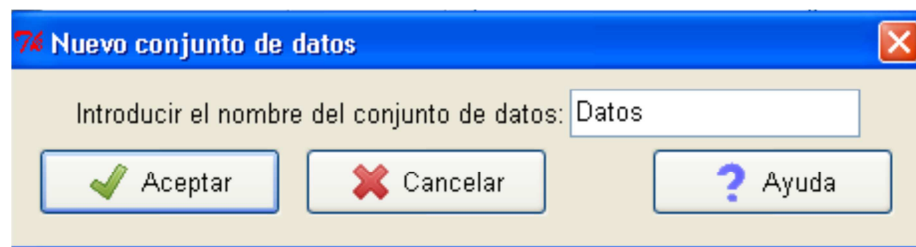
En esta gráfica se aprecian, de arriba abajo, los elementos fundamentales de la interfaz:

- la barra de menús;
- la barra de elementos activos;
- el área de instrucciones (donde aparece el código R asociado a las acciones elegidas mediante menús);
- el área de resultados, escritos en texto puro, lo que puede dificultar su lectura e interpretación (los gráficos aparecen en ventanas diferentes);
- el área de mensajes (donde se avisa, entre otras cosas, de los errores cometidos).

2.3.1.- Edición de datos

El editor de R-Comander permite escribir datos en una cuadrícula de apariencia similar a la de SPSS, Excel, etc. Sin embargo, resulta poco amigable (por ejemplo, no permite deshacer ninguna acción) y sólo cabe aconsejarla para ejemplos muy pequeños.

Datos → Nuevo conjunto de datos

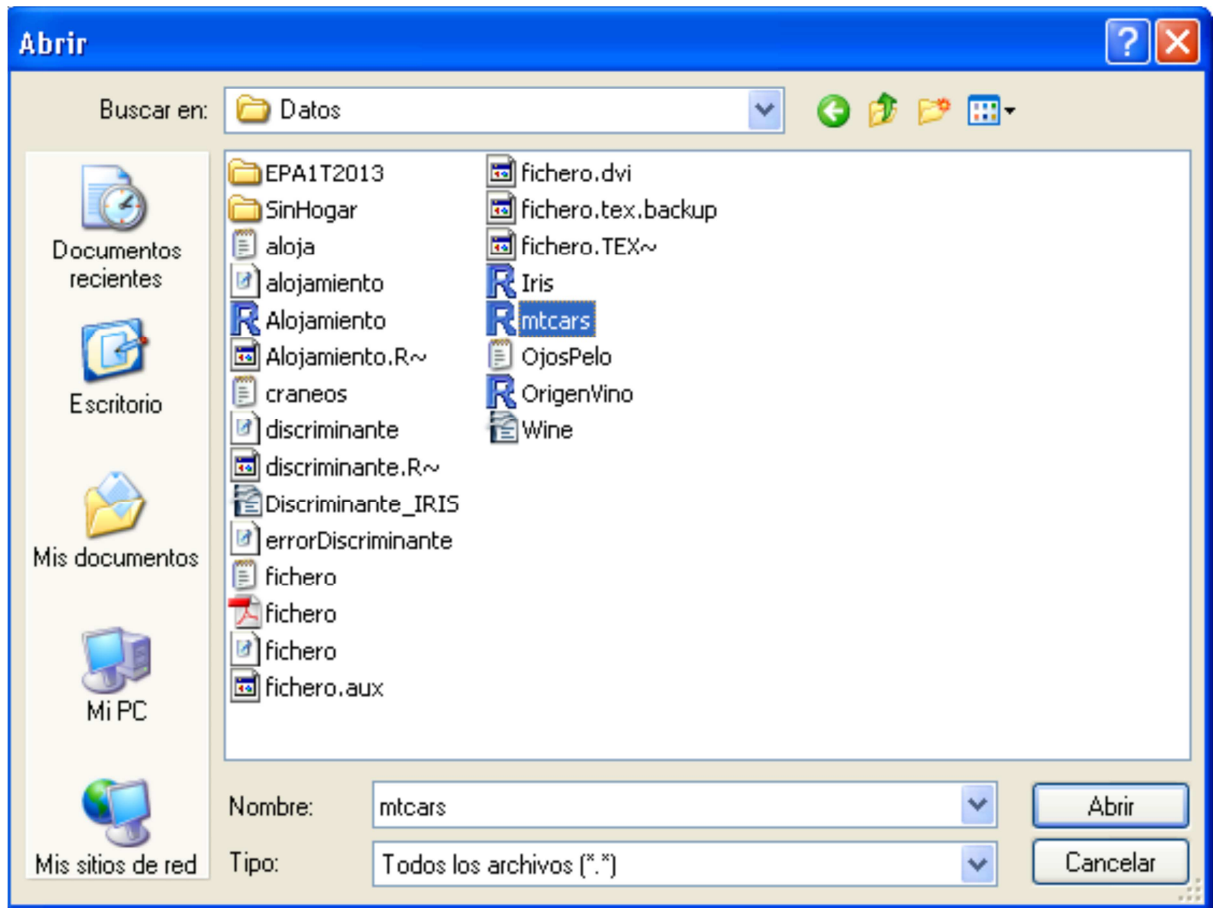


Después de poner el nombre que se desee al conjunto de datos y aceptar aparece una rejilla en la que se pueden escribir los datos.

	var1	var2	var3	var4	var5	var6
1						
2						
3						
4						
5						
6						
7						
8						

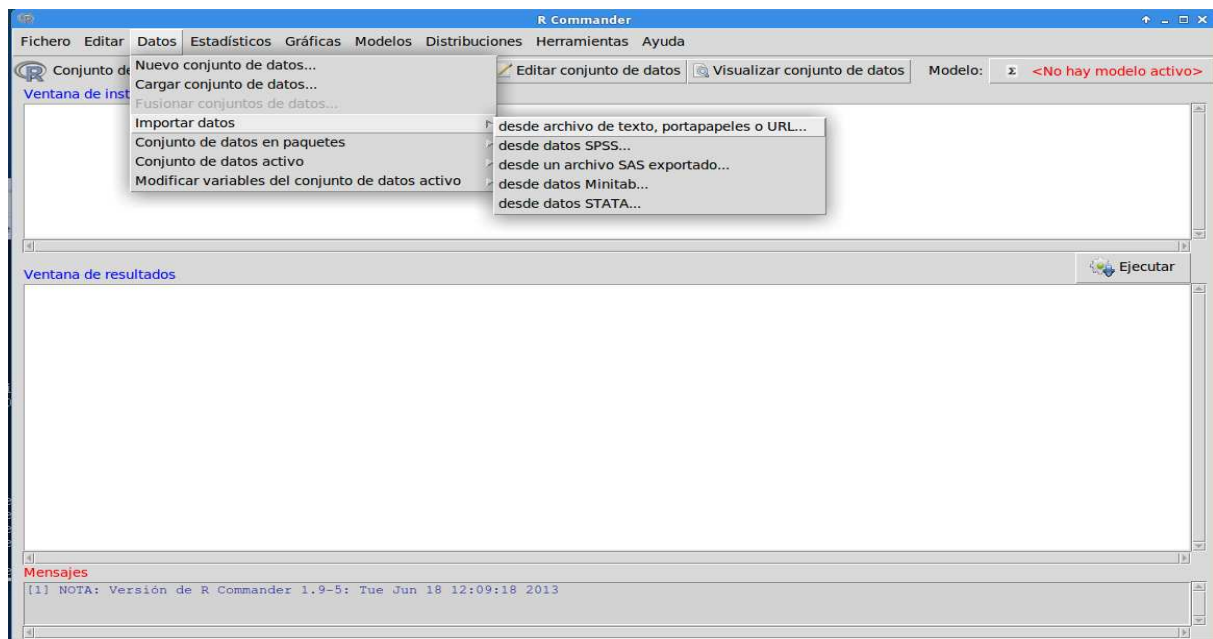
Como decíamos, lo más usual no es editar datos en R-Comander, sino cargarlos de un fichero ya preparado. Una posibilidad es abrir un fichero con datos almacenados en el formato nativo de R (normalmente con extensión .RData o .rda). Los pasos a seguir son

:
Menú Datos → Cargar conjunto de datos → Dirección del fichero ...



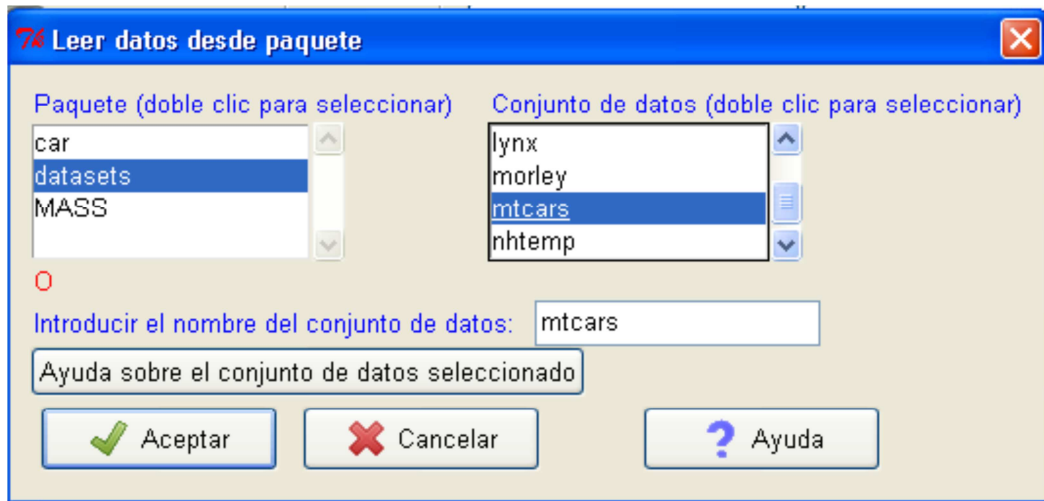
Se pueden importar datos guardados en otros formatos, como texto CSV, SPSS, SAS, Minitab, Stata...

Datos → Importar datos → Tipo de datos → Dirección del fichero ...



También se puede recurrir a usar ficheros distribuidos directamente por R en paquetes de datos. El paquete de datos y el fichero concreto se pueden seleccionar pinchando dos veces sobre el nombre.

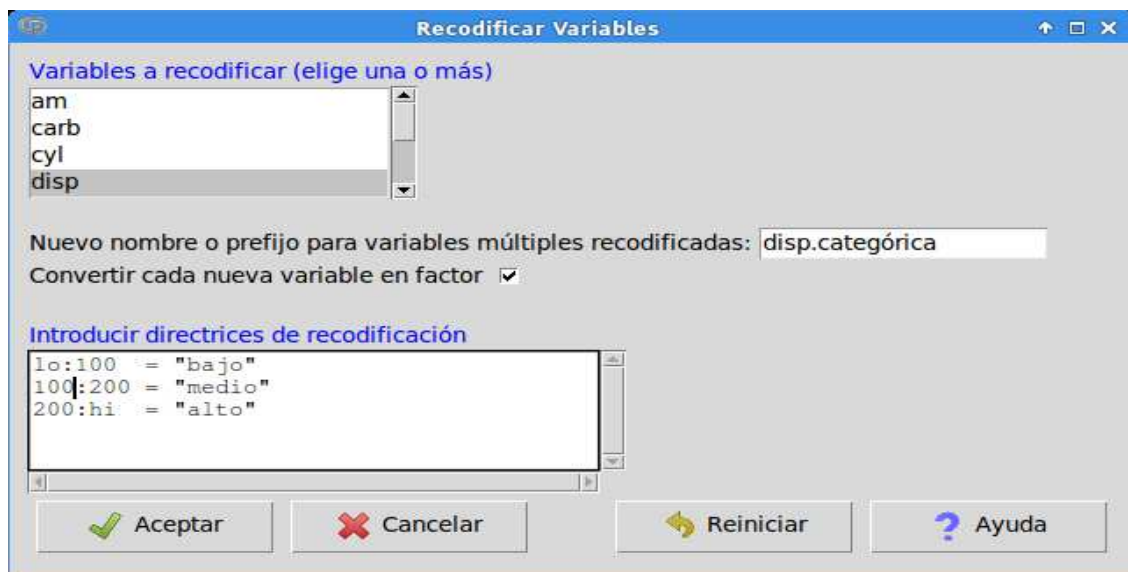
Menú Datos → Conjunto de datos en paquetes → Leer conjunto de datos ...



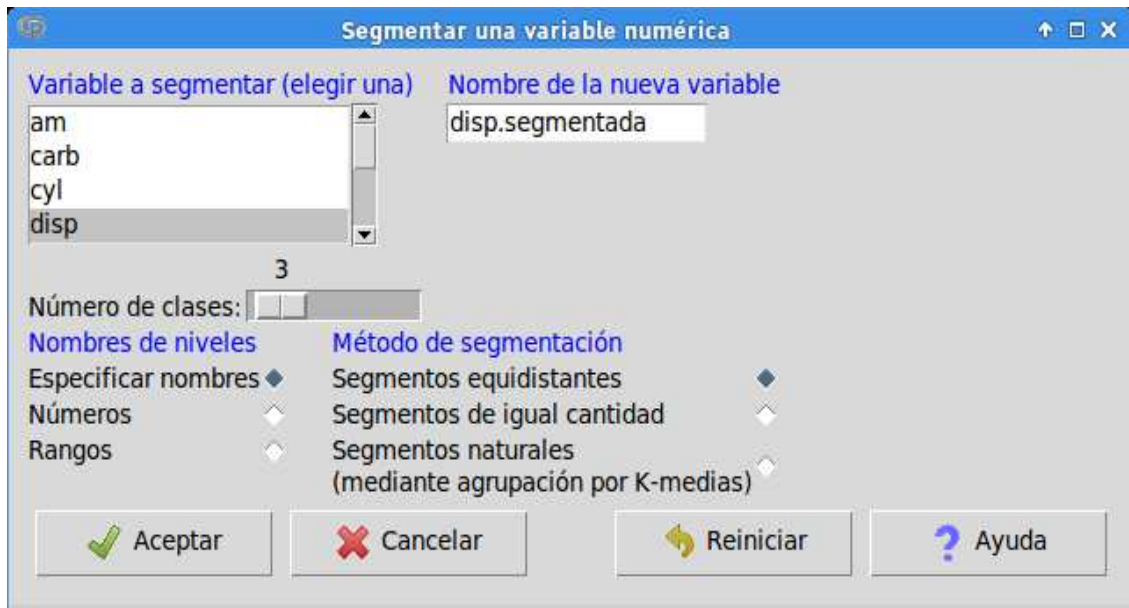
Un conjunto de datos, en lenguaje R, es simplemente un objeto más. De este modo, R-Commander permite tener cargados, de manera simultánea, distintos conjuntos de datos, y el usuario decide cuál de ellos es el “activo” con total libertad en cada momento.

La creación de nuevas variables o modificación de las ya existentes resulta razonablemente cómoda. En este punto conviene comentar que R-Commander insiste mucho en distinguir las variables cuantitativas (llamadas aquí “numéricas”) de las cualitativas (llamadas “factores”). Para crear un factor a partir de otro o de una variable numérica se puede emplear el diálogo:

Datos → Modificar variables del conjunto de ... → Recodificar variables...



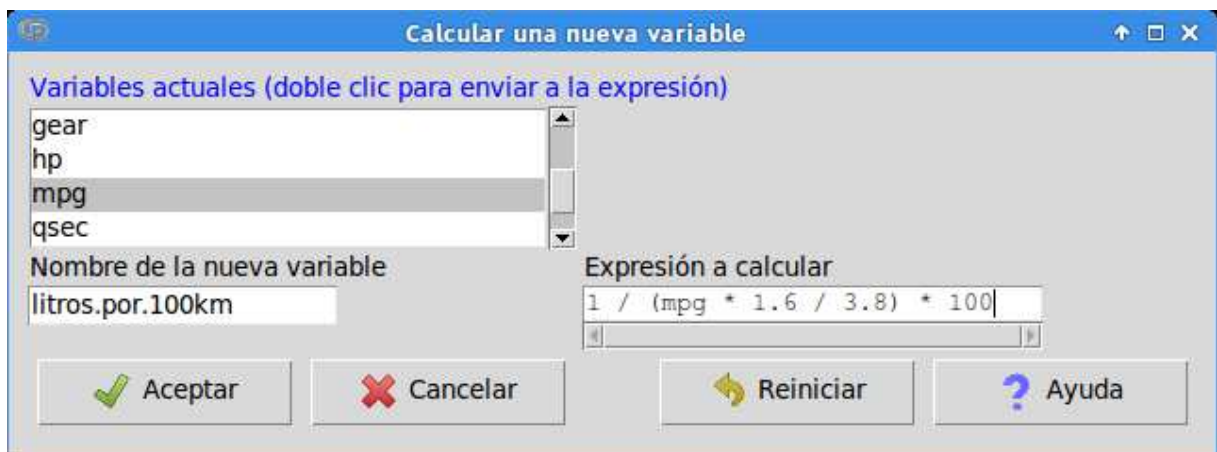
Cuando se tengan dudas sobre cualquier aspecto de este proceso, conviene leer con atención la ayuda del diálogo de recodificar, porque la sintaxis es sencilla, aunque bastante rígida. Si se prefiere una manera automática de recodificación, puede ser útil la opción de Segmentar una variable numérica:



Para crear nuevas variables numéricas se utiliza:

Datos → *Modificar variables del conjunto de datos...* → *Calcular una nueva variable*

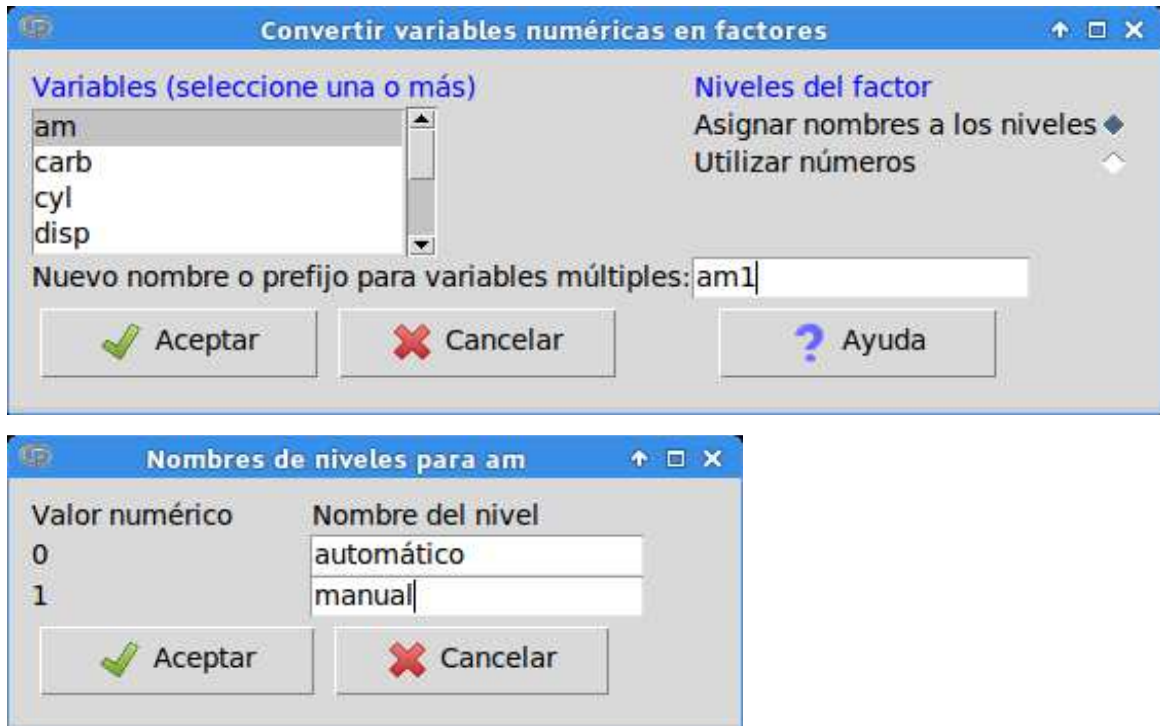
Como ejemplo, veamos cómo pasar de la variable “mpg” (millas por galón) de “mtcars” a litros de combustible consumidos por cada 100 kilómetros:



Un caso bastante frecuente es el paso de una variable numérica a factor. Esto se puede ser resuelto cómodamente con

Datos → *Modificar...* → *Convertir variables numéricas en factores*

Esta situación afecta generalmente a variables cualitativas discretas (p.ej. “número de hermanos del encuestado”) y a variables cualitativas codificadas mediante números, las cuales R-Commander va a considerar como numéricas equivocadamente. Por ejemplo, la variable “am” del conjunto de datos toma valor 0 para indicar “automático” y valor 1 para “manual”:



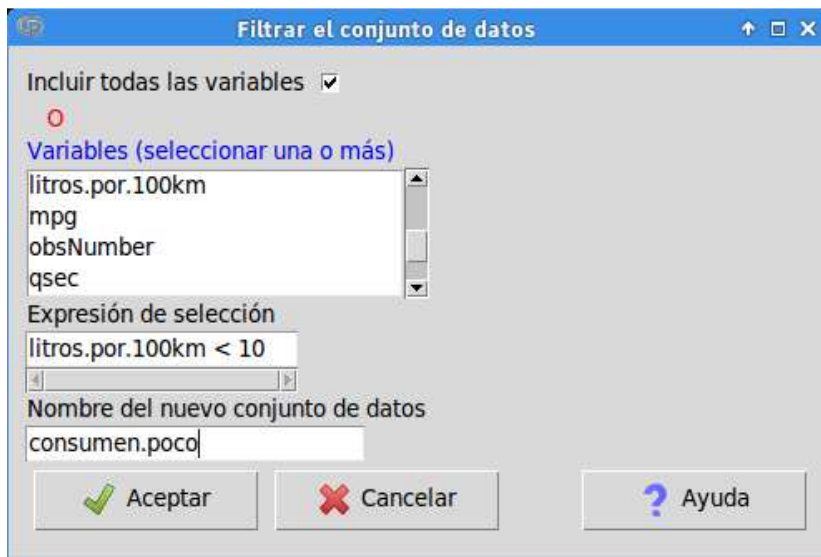
Al convertir una variable numérica en factor suele ser conveniente guardarla con un nombre diferente (en el ejemplo, “am1”), lo que permite conservar la versión original por si es útil más adelante.

La conversión de factor a variable numérica es muy sencilla de hacer, aunque puede resultar difícil que se le ocurra a uno: Se selecciona el submenú “Editar conjunto de datos”; después se pincha en el nombre de la variable y se elige “numérico”. Esta operación, sin embargo, está sujeta a una serie de excepciones que nos lleva a aconsejar, en general, que se tenga cuidado con el formato de las variables numéricas en los datos antes de importarlos a R-Commander.

Al igual que creamos variables, podemos eliminarlas con la misma facilidad:



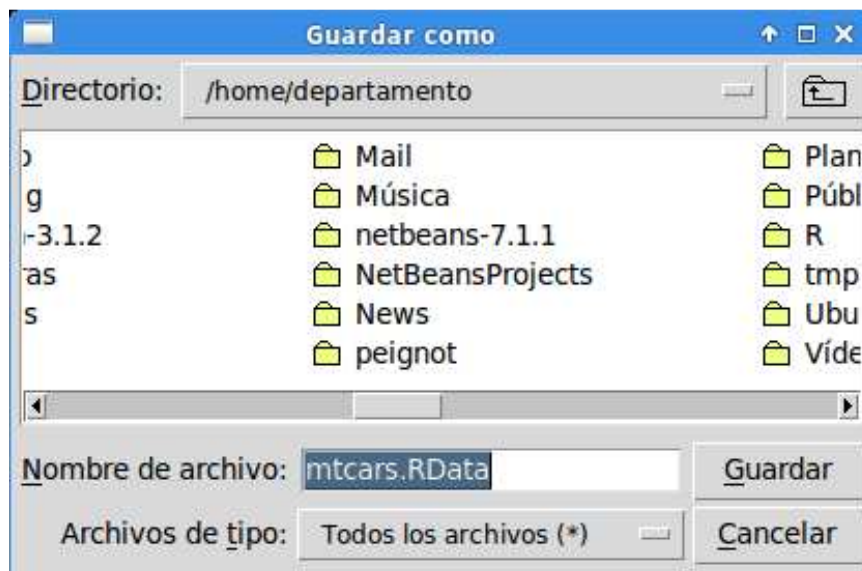
R-Commander permite filtrar los datos actuales según un criterio arbitrario (redactado mediante los operadores lógicos y de comparación del lenguaje R: &, |, ==, !=, <=, >=, <, >) para generar un nuevo conjunto de datos:



A la hora de conservar nuestros datos, hay distintas alternativas:

- Guardar el conjunto de datos activo en un fichero usando el formato .RData, propio de R, que conserva toda la información que R asigna a los datos:

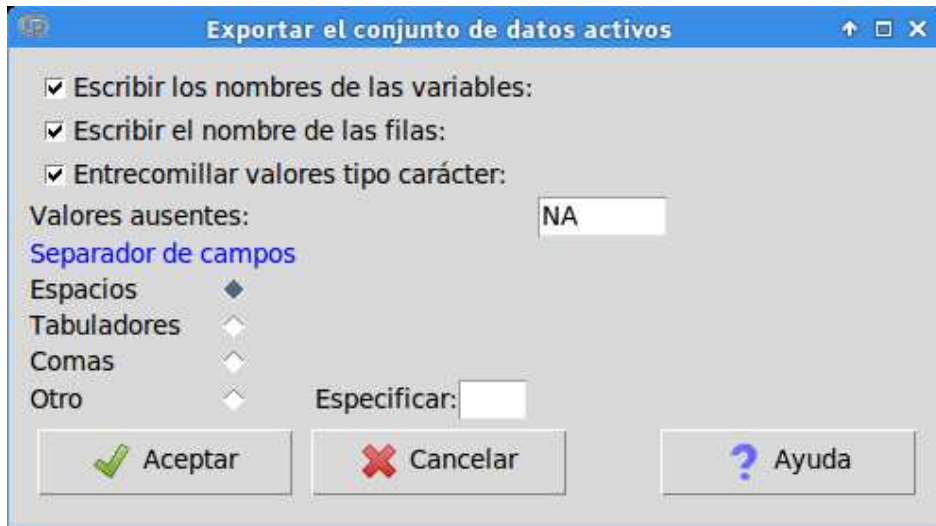
Datos → Conjunto de datos activo → Guardar el conjunto de datos activo → ...



- Guardar todos los conjuntos de datos definidos en la presente sesión:

Fichero → Guardar el entorno de trabajo R como ...

- Exportar los datos en formato de texto, que es el más universal y nos permitirá cargar esos datos en cualquier otro programa. Los ficheros de texto suelen usar extensiones como TXT, CSV (valores separados por comas), TSV (valores separados por tabuladores), etcétera. Al exportar a texto se puede escoger si incluir una cabecera con los nombres de las variables, los nombres de los casos, cómo indicar los valores ausentes, cómo separar los campos de cada registro/renglón y si poner los valores de variables cualitativas entre comillas, para evitar problemas en caso de que puedan contener el mismo carácter que se usa para separar campos.



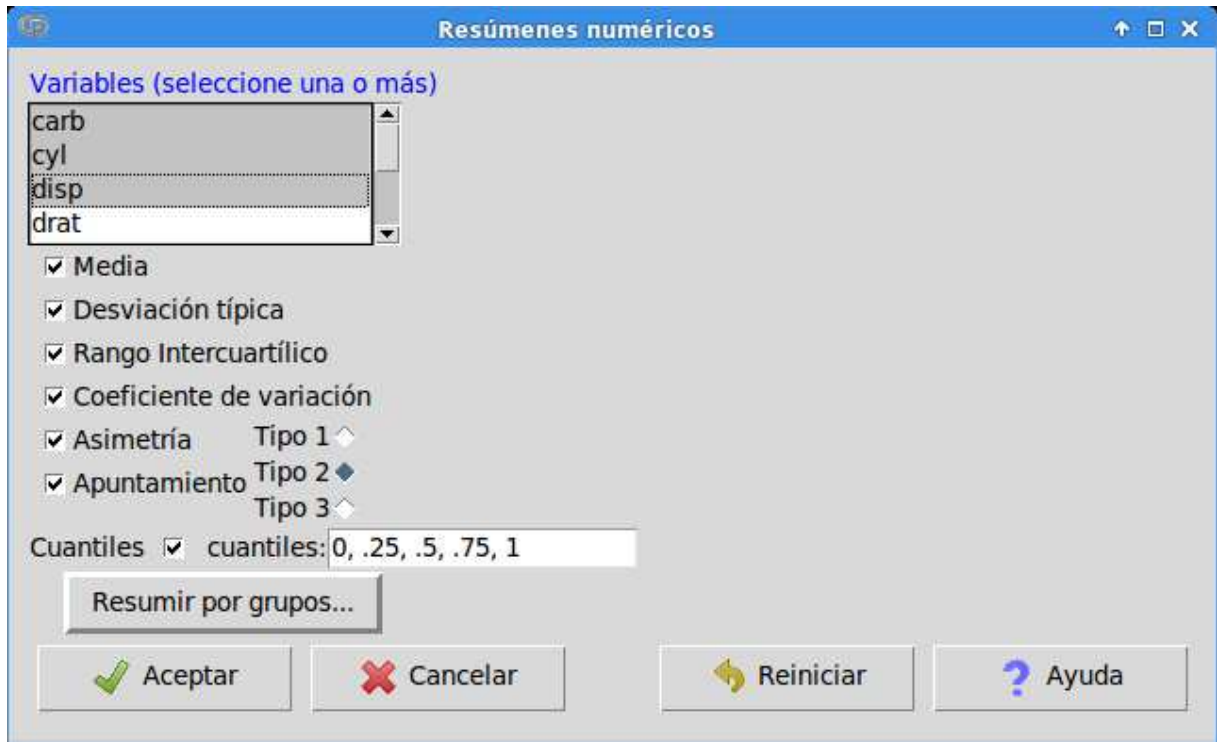
Como ya indicamos, la edición de datos directamente en R-Commander no es aconsejable y resulta más conveniente hacerlo con otras aplicaciones como hojas de cálculo (LibreOffice Calc, Microsoft Excel, etc.), editores de texto, bases de datos, etc.

2.3.2.- Análisis Estadísticos

Veamos ahora qué tipo de análisis estadísticos se pueden hacer directamente desde R-Commander.

En R es muy importante distinguir entre variables cuantitativas y cualitativas. Para las variables numéricas se pueden pedir medidas descriptivas globales o por grupos definidos mediante una variable declarada como factor:

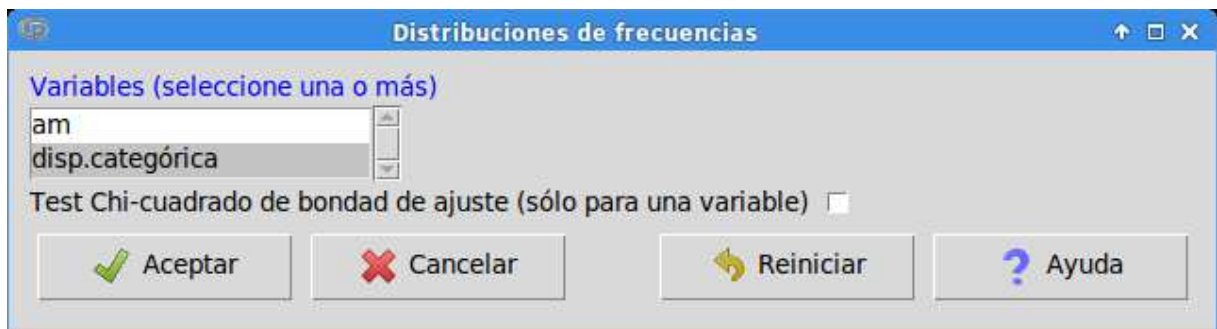
Estadísticos → Resúmenes → Resúmenes numéricos



	mean	sd	IQR	cv	skewness	kurtosis	0%	25%	50%	75%	100%	n
carb	2.8125	1.615200	2.000	0.5742933	1.1570911	2.020059	1.0	2.000	2.0	4	8	32
cyl	6.1875	1.785922	4.000	0.2886338	-0.1922609	-1.762794	4.0	4.000	6.0	8	8	32
disp	230.7219	123.938694	205.175	0.5371779	0.4202331	-1.067523	71.1	120.825	196.3	326	472	32

Para los factores se pueden obtener distribuciones de frecuencias absolutas y relativas (porcentajes):

Estadísticos → *Resúmenes* → *Distribución de frecuencias*

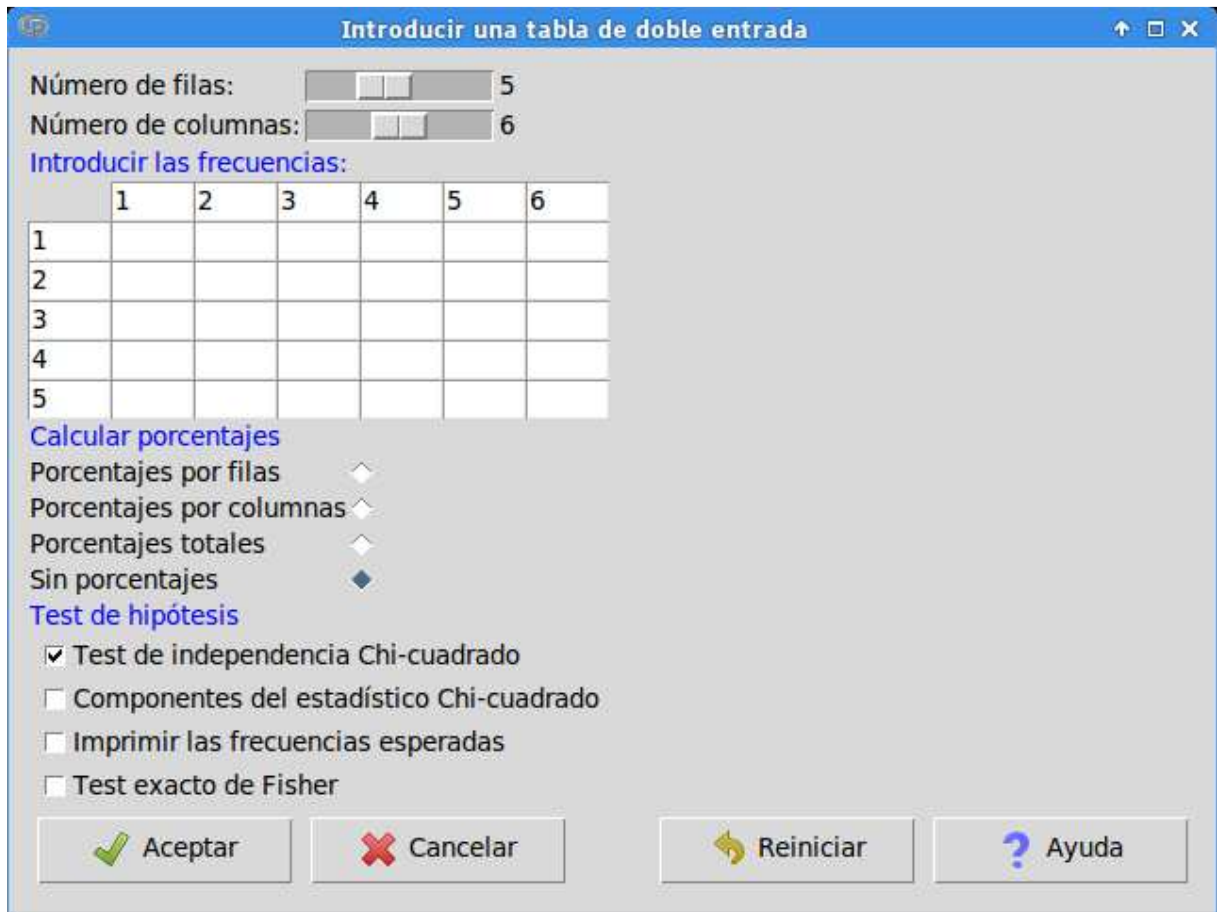


```
> .Table # counts for disp.categorica
alto bajo medio
  16   5   11

> round(100*.Table/sum(.Table), 2) # percentages
alto bajo medio
50.00 15.62 34.38
```

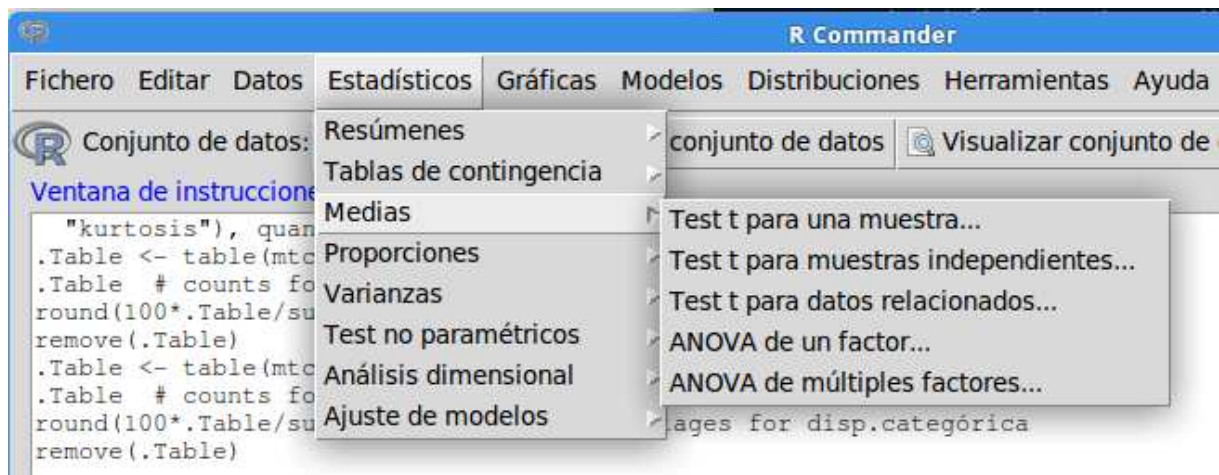
Un cuadro de diálogo bastante útil es el que permita la entrada directa de frecuencias en una tabla de contingencia:

Estadísticos → *Tablas de contingencia* → *Introducir y analizar una tabla ...*



En lo que respecta a inferencia estadística clásica las alternativas son bastante completas. Por ejemplo se pueden comparar medias mediante contrastes paramétricos (t de Student y análisis de varianza) y hacer test de hipótesis sobre proporciones:

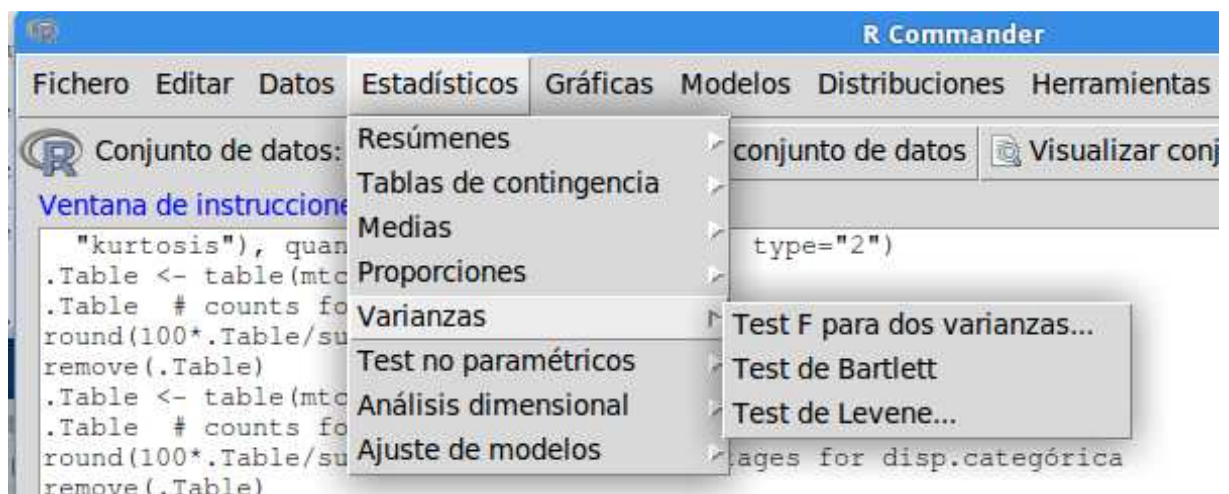
Estadísticos → *Medias*



Cabe comentar que el análisis de la varianza con medidas repetidas no está explícitamente recogido en R-Commander. Es necesario aprender a manejar el lenguaje de modelos o fórmulas implementado en R para expresar la descripción correcta del modelo. Por ejemplo, “respuesta ~ color * forma + Error (sujeto / (color + forma))” significa un modelo en que la variable respuesta depende linealmente de las variables color y forma y de sus interacciones, y en que la variable sujeto indica las medidas repetidas. Es un poco incómodo al principio, pero el lenguaje de fórmulas de R permite la expresión de modelos muy complejos con cierta facilidad.

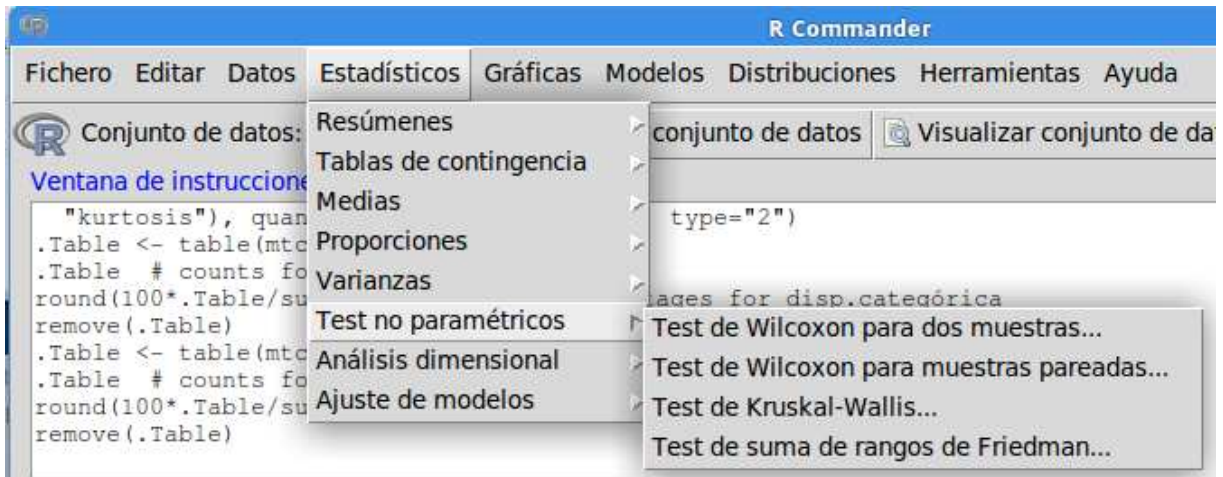
La comparación paramétrica de medias requiere a menudo la hipótesis de igualdad de varianzas que se puede hacer con tres métodos distintos:

Estadísticos → Varianzas



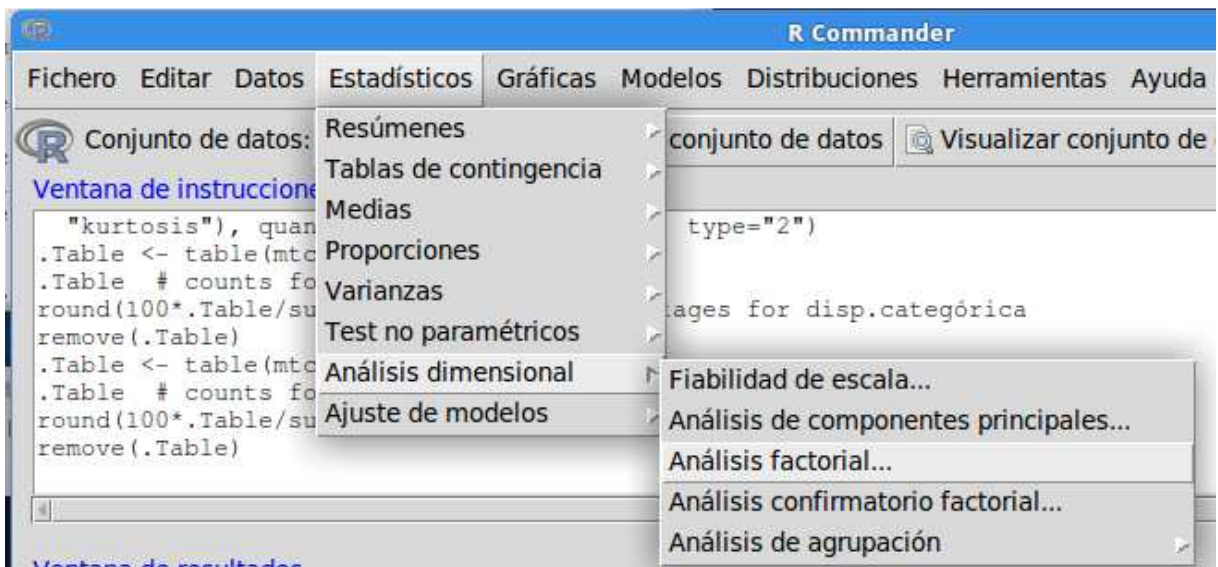
R-Commander también ofrece una selección de las pruebas no paramétricas más habituales:

Estadísticos → Test no paramétricos



En el submenú “Análisis dimensional” se encuentran algunas técnicas de análisis multivariado, cuyo objetivo es representar la estructura de un gran número de variables mediante un número reducido de nuevas variables construidas mediante combinaciones lineales de las originales:

Estadísticos → Análisis dimensional



Las técnicas ofrecidas son:

- Fiabilidad de escala. Calcula el alfa de Cronbach.
- Componentes principales. Se puede elegir trabajar con la matriz de varianzas-covarianzas o con la matriz de correlaciones. Permite guardar las puntuaciones de las componentes.
- Análisis factorial. La estimación de las cargas factoriales se hace suponiendo que los datos proceden de una distribución normal multidimensional y emplea el algoritmo desarrollado por Lawley y Maxwell. Las rotaciones que se pueden hacer son la ortogonal varimax y la oblicua promax. Una vez realizado un análisis factorial, permite realizar un análisis confirmatorio factorial.

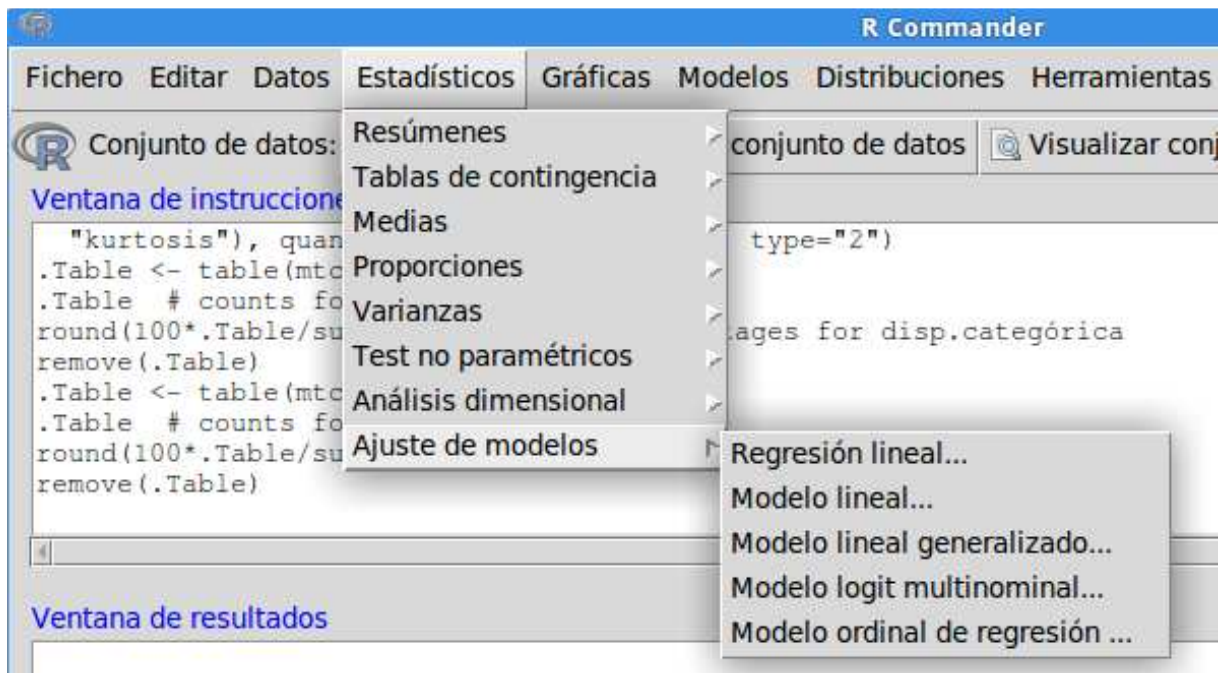


- Análisis de agrupación.
 - Clasificación de las k-medias: Las opciones disponibles permiten seleccionar el número de semillas iniciales, el número máximo de iteraciones, crear una nueva variable con el grupo de cada individuo, etc.
 - Clasificación jerárquica ascendente. Dispone de los procedimientos de agregación más usuales: Ward, enlace simple (salto mínimo), enlace completo, enlace de medianas, enlace de centroides. El abanico de posibilidades para las distancias entre individuos es muy poco variado, ya que sólo ofrece la euclídea, la euclídea al cuadrado y la de Manhattan.



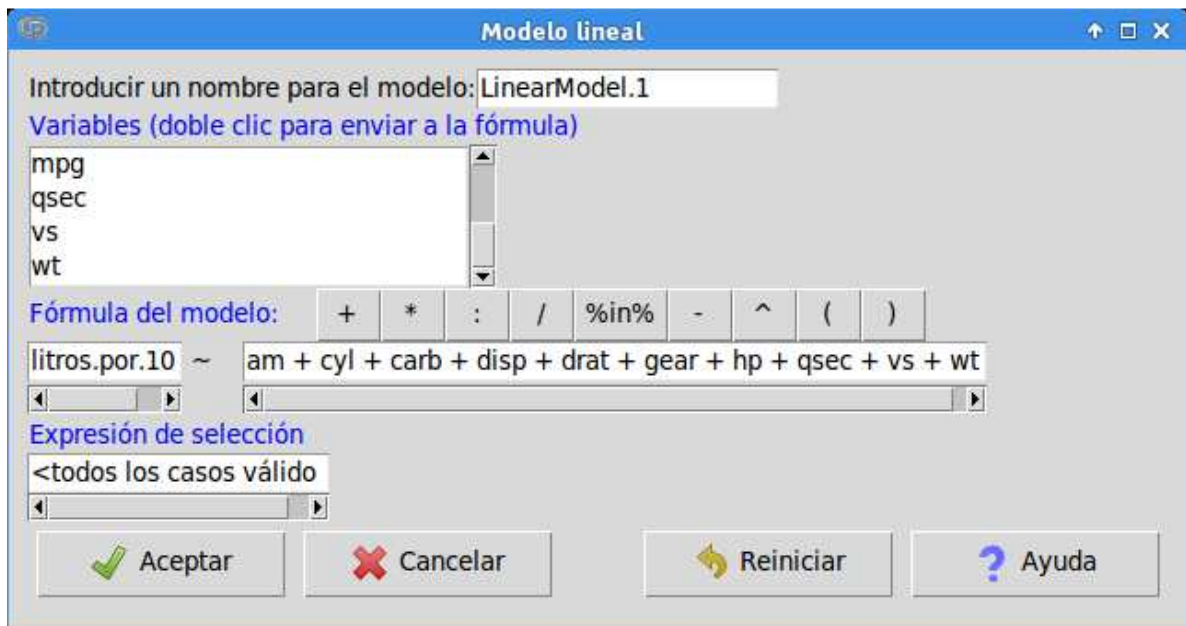
El módulo de ajuste de modelos es bastante completo y dispone de muchos tipos de gráficos que ayudan a buscar el modelo más adecuado a los datos del experimento o a contrastar las hipótesis previas:

Estadísticos → *Ajuste de modelos*



La forma de utilizar estos procedimientos con R commander tiene dos etapas:

1. Se construye el modelo de regresión lineal o logística.



La salida del análisis solicitado aparece así en R-Commander:

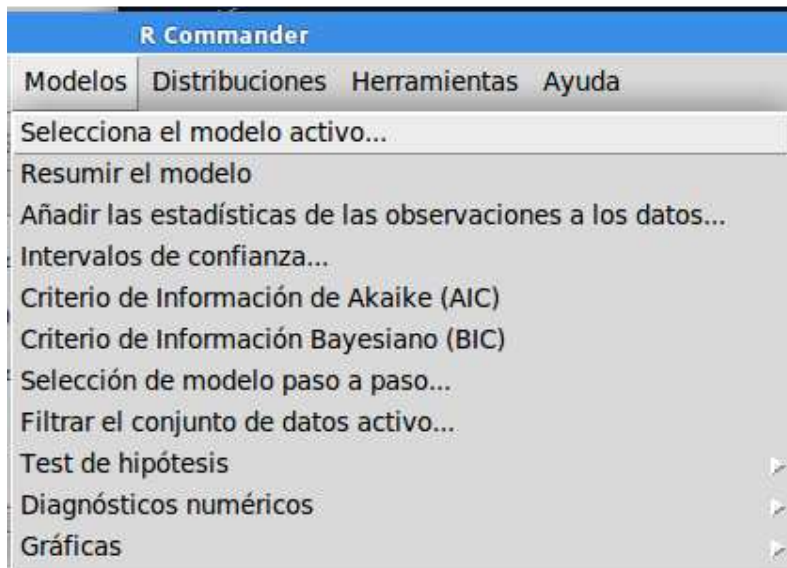
```
Call:
lm(formula = litros.por.100km ~ am + cyl + carb + disp + drat +
    gear + hp + qsec + vs + wt, data = mtcars)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-4.0494 -0.7863  0.1125  0.9087  2.7991
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.660317  12.302257   1.110   0.279
am           0.436406   1.351726   0.323   0.750
cyl        -0.433091   0.686838  -0.631   0.535
carb        0.454490   0.544694   0.834   0.413
disp        0.009497   0.011737   0.809   0.427
drat       -0.150081   1.074843  -0.140   0.890
gear       -1.100244   0.981439  -1.121   0.275
hp          0.006832   0.014307   0.477   0.638
qsec       -0.271162   0.480345  -0.565   0.578
vs          0.216074   1.383180   0.156   0.877
wt          2.048924   1.245096   1.646   0.115
```

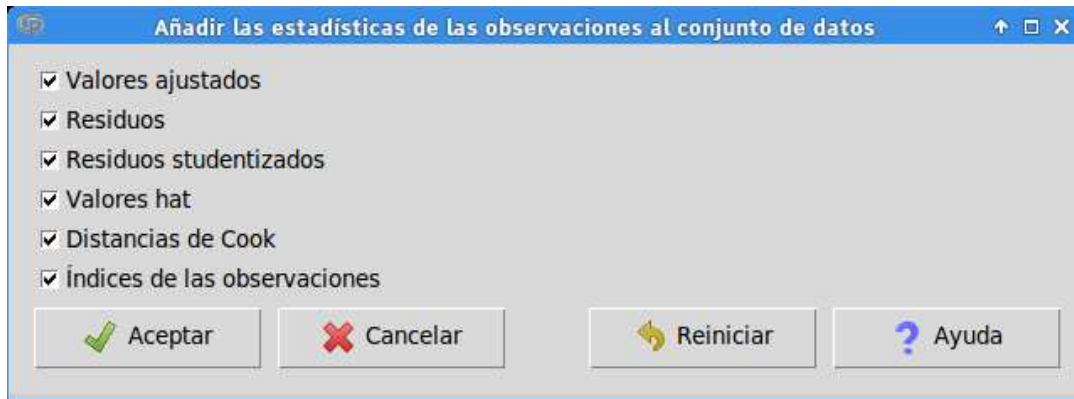
```
Residual standard error: 1.742 on 21 degrees of freedom
Multiple R-squared: 0.8649, Adjusted R-squared: 0.8006
F-statistic: 13.45 on 10 and 21 DF, p-value: 5.15e-07
```

2. En el menú Modelos se pueden seleccionar las opciones necesarias para estudiar la adecuación del modelo construido y analizarlo en detalle.

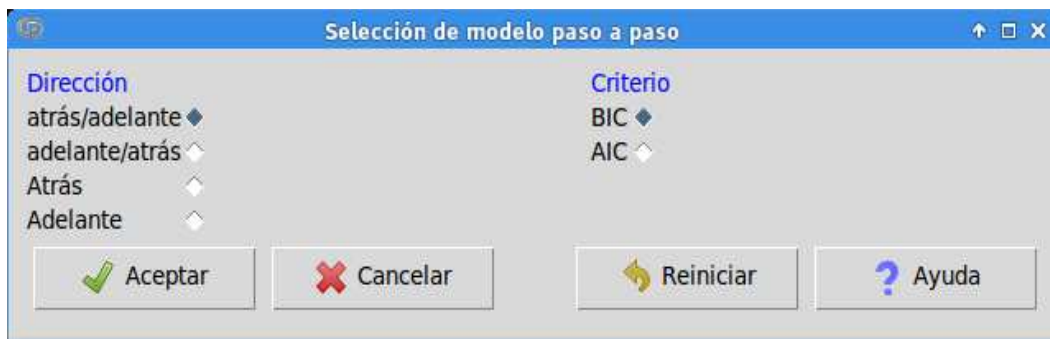


Entre las opciones más comunes se pueden citar las siguientes:

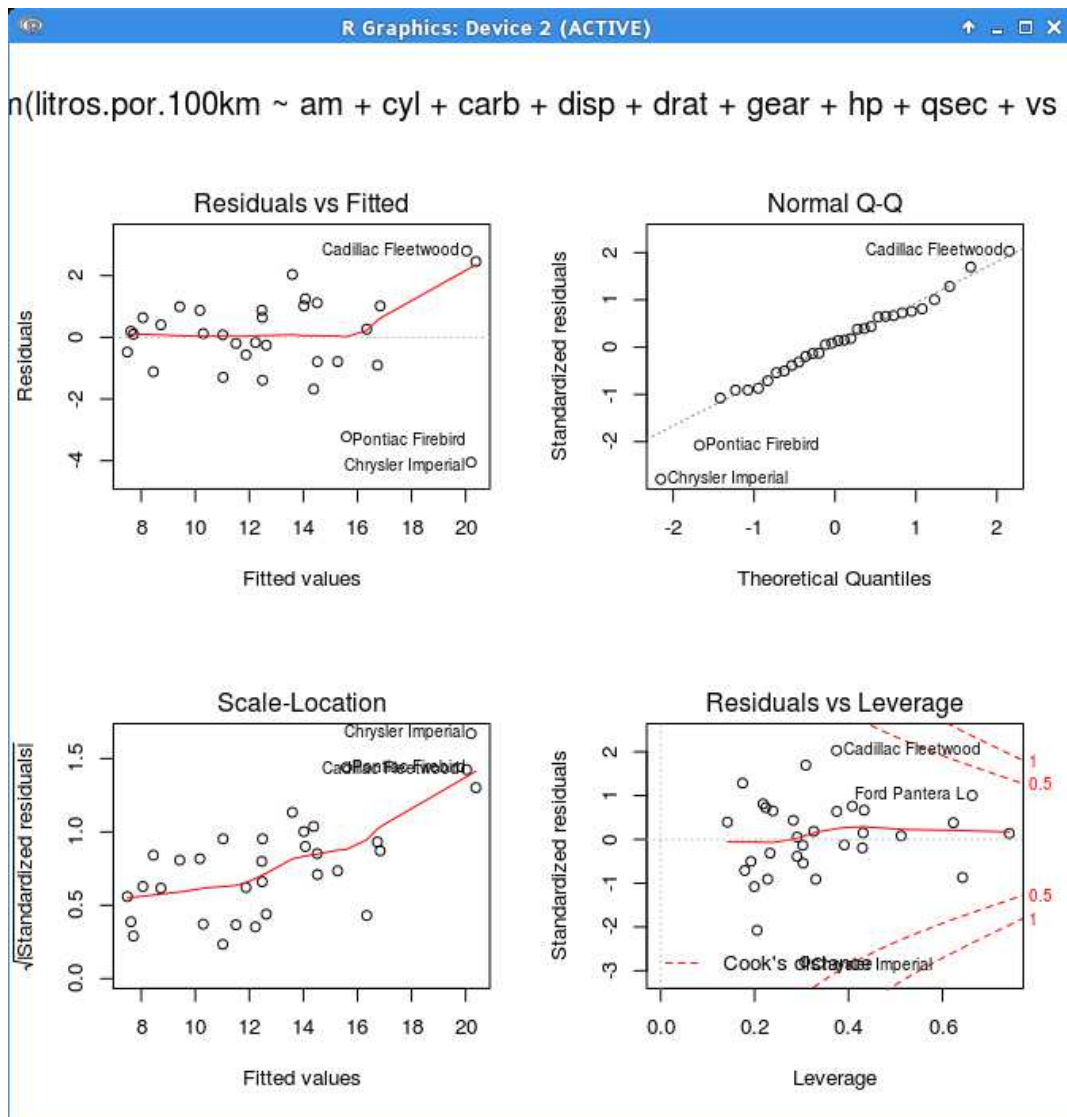
- Añadir las estadísticas de las observaciones a los datos.



- Selección de modelo paso a paso.



- Gráficas varias: de residuos, de regresión parcial, de puntos influyentes, etc.



3.- Otros proyectos

PSPP

Se trata de un proyecto que intenta desarrollar un clon libre del programa SPSS. De momento, PSPP implementa un buen número de las capacidades básicas de SPSS mediante los menús, y la interfaz gráfica es fácil de manejar y más amigable que la R-Commander. Por ello, puede recomendarse para un curso de estadística básica. Tiene versiones para Linux y para Windows que son muy sencillas de instalar. En el siguiente número de la revista se

OpenEpi

Está escrito en Javascript, por lo que la “instalación” supone descomprimir un ZIP y apuntar el navegador al directorio descomprimido. Por desgracia, para los nombres de los ficheros no se ha tenido cuidado en diferenciar mayúsculas y minúsculas, por lo que para sistemas de estilo Unix (como Ubuntu) habría que descomprimirlo en una partición con un sistema de ficheros que no diferenciara entre mayúsculas y minúsculas (por ejemplo, una memoria USB con sistema VFAT, el habitual).

Hace algunos cálculos que pueden ser interesantes en epidemiología, pero es bastante incómodo de usar en general.

Dataplot

Aunque se trata de un programa veterano, no ha sido distribuido con los sistemas operativos libres, por lo que su instalación es algo más difícil. Tiene una versión para Windows.

Se trata de un programa bastante completo, aunque hay que dirigirlo mediante comandos, pues no dispone de interfaz gráfica.

OpenBUGS

El programa WinBUGS es un clásico de la estadística bayesiana. Su desarrollo se ha congelado y actualmente el proyecto se centra en OpenBUGS, la versión libre. Fuera del ámbito bayesiano no ha tenido mucho éxito.

Xlispstat

Se trata de un sistema de estadística avanzada de alta calidad, que precedió en unos años a R. A partir de él se han desarrollado dos interfaces gráficas, ARC y ViSta. ARC se escribió como un apoyo para facilitar el uso de métodos de regresión, y no está mal en ese sentido, pero carece de editor de datos y su forma de distribución no favorece mucho la instalación. ViSta tiene buena pinta, pero sus autores entienden de forma un tanto particular el concepto de “abierto y libre”, lo que ha impedido su distribución en sistemas operativos libres.

Macanova

Tiene una interfaz gráfica que permite acceder a parte de sus prestaciones. Funciona en Windows y Mac; no hemos podido compilarlo en Ubuntu.

Gretl

Especialmente útil en econometría. Es bastante pobre en descriptivos gráficos.

Adamsoft, Weka

Estos dos son programas para minería de datos. Ambos dependen de Java. Weka es una colección de cientos de algoritmos (regresiones lineales, árboles, redes neuronales, bayes ingenuo...) que ha sido muy popular durante años.

Incanter

El lenguaje R, aunque sintácticamente es del estilo Algol (con notación infija, llaves para bloques, etc.) semánticamente es similar a Scheme (el segundo dialecto más popular de Lisp). Se está desarrollando un lenguaje de programación estadístico basado en Clojure, otro dialecto de Lisp, que está en boga y se ejecuta sobre la máquina virtual de Java. A pesar de su diseño más elegante que R, carece de su enorme acervo de paquetes y es incluso más lento.

Julia

Entre los adeptos de R está adquiriendo fama rápidamente este nuevo lenguaje. Pretende tener una sintaxis más “limpia” que la de R (imitando la de Octave) y generar código muy rápido. El tiempo dirá si sus prestaciones para la computación estadística lo hacen alcanzar la relevancia que tiene R actualmente.

4.- Bibliografía

Arriaza, A. J., Fernández, F., López, M. A., Muñoz, M., Pérez, S., y Sánchez, A. (2008). *Estadística básica con R y R-Commander*. <http://knuth.uca.es/moodle/course/view.php?id=37>

- Baron, J. (2011). *Notes on the use of R for psychology experiments and questionnaires*.
<http://www.psych.upenn.edu/~baron/rpsych/rpsych.html>
- Elosua, P. (2009). ¿Existe vida más allá de SPSS? Descubre R. *Psicothema*, 21, 652–655.
- Elosua, P. (2011). *Introducción al entorno R*. Bilbao: Euskal Herriko Unibertsitateko Argitalpen Zerbitzua / Servicio Editorial de la Universidad del País Vasco.
- Elosua, P., y Etxeberria, J. (2012). *R Commander. Gestión y análisis de datos*. Madrid: La Muralla.
- Matas, A., Franco, P.D. y Atorrasagasti, L. (2011). *Estadística básica con R-Commander*. Descargado de <http://www.bubok.es/libros/203887/Estadistica-basica-con-RCommander>
- Sáez, J. A. (2010). *Métodos estadísticos con R y R commander*. Descargado de <http://cran.r-project.org/doc/contrib/Saez-Castillo-RRCmdrv21.pdf>.