A pequeños problemas soluciones baratas

Jesús M. Rodríguez Rodríguez
jesusmaria.rodriguez@jcyl.es

D. G. de Presupuestos y Estadística

Junta de Castilla y León

RESUMEN

En el día a día de nuestro trabajo en un organismo estadístico tenemos que enfrentarnos a pequeños problemas de lectura y tratamiento de datos, problemas que se pueden abordar individualmente o intentar automatizarlos.

La automatización en el organismo estadístico de la Junta de Castilla y León se ha hecho tradicionalmente utilizando Excel o SAS, ambas licencias tienen un coste, aunque bastante diferente, pero en la actualidad se están realizando pequeños programas en R para implementar alguno de los procesos.

En esta ponencia se presentan alguno de los programas desarrollados en R.

Uno de ellos encaminado a la lectura de datos en formato fijo, de una forma más rápida y flexible que la que hay por defecto en R y que se basa en un fichero auxiliar de formato, que puede ser un fichero de texto, Excel o simplemente un data.frame o matriz de R. Dicho formato contendrá el nombre de cada campo, donde empieza y termina o donde empieza y su longitud, el tipo de datos que contiene y, si es necesario, el número de decimales.

Otro para el tratamiento del Indicador de Confianza del Consumidor y abarca todo el proceso, desde la lectura de los datos hasta la obtención de las tablas y gráficos, que además mantiene los datos históricos para poder realizar los gráficos de series y cálculos de variaciones. En este programa se ha abordado también el problema del ajuste de los datos cuando se presentan redondeados, para que en las tablas que se distribuyen todo este perfectamente ajustado.

Para la realización de gráficos se han realizado varios programas, que aunque haya paquetes que hagan algo similar, se han desarrollado para cubrir nuestras necesidades específicas. Entre ellos se puede destacar el que sirve para la elaboración de pirámides de población, con la particularidad de que es posible de superponer varias de ellas para poder ser comparadas. Se ha programado también un pequeño interfaz gráfico para facilitar su uso por personas que desconozcan el lenguaje. Además da la posibilidad de leer datos directamente del Sistema de Información Estadística de la Junta de Castilla y León.

INTRODUCCIÓN

En esta ponencia se van a presentar algunas pequeñas aplicaciones escritas en R cuyo objetivo es resolver pequeños problemas que nos hemos ido encontrando, problemas que estaban casi todos resueltos con otras aplicaciones en SAS, Excel, … pero para los que se ha decidido utilizar un programa gratuito, orientado al cálculo estadístico y ampliamente utilizado en ámbitos académicos, científicos y profesionales por todo el mundo, es decir R.

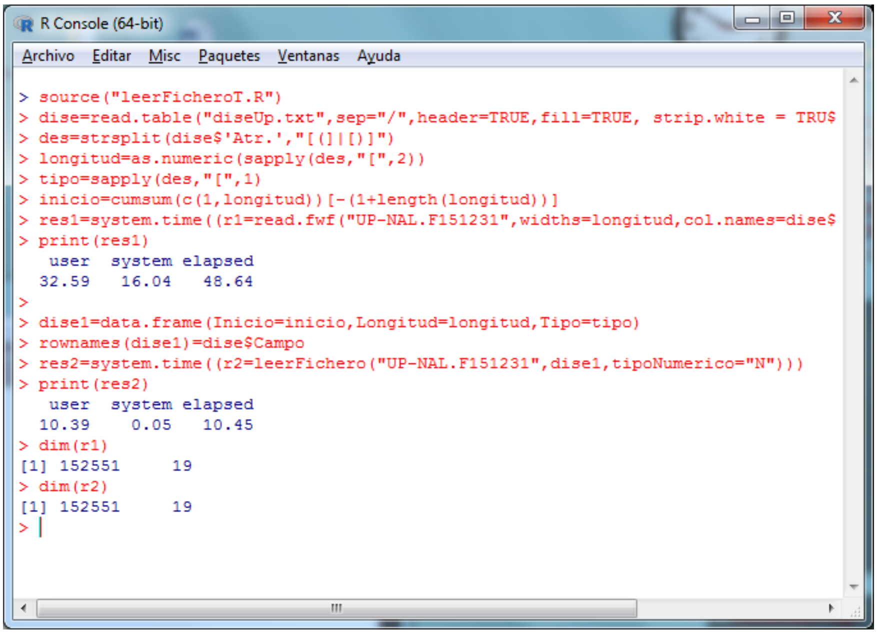
LECTURA DE FICHEROS DE TEXTO EN FORMATO FIJO

En R existe la función **read.fwf** que sirve para leer este tipo de ficheros, función que es fácil de usar y que cuando no hay muchos registros y cada registro no tiene muchos campos no da grandes problemas, pero cuando el número de registros es grande se hace muy lenta y cuando el número de campos es grande puede dar problemas de memoria.

La función que se ha diseñado lo que hace es leer el fichero como texto, del vector obtenido se extrae cada campo en función de lo que se indica en el diseño de registro y se crea un data.frame.

El diseño de registro es un data.frame, que se puede obtener de un fichero de texto, de Excel o de cualquier otra fuente. El nombre de las filas del diseño contendrá el nombre de los campos a leer, aunque también puede estar en uno de los campos, cuyo nombre se pasará a la función de darse este caso.

Para comparar la velocidad de ambas funciones se han tomado los datos del fichero de Unidades Poblacionales del callejero del INE, que tiene 152.551 registros.



```
R Console (64-bit)
Archivo  Editar  Misc  Paquetes  Ventanas  Ayuda

> source("leerFicheroT.R")
> dise=read.table("diseUp.txt",sep="/",header=TRUE,fill=TRUE, strip.white = TRU$
> des=strsplit(dise$'Atr.',"[(]|[)]")
> longitud=as.numeric(sapply(des,"[",2))
> tipo=sapply(des,"[",1)
> inicio=cumsum(c(1,longitud))[-(1+length(longitud))]
> res1=system.time((r1=read.fwf("UP-NAL.F151231",widths=longitud,col.names=dise$
> print(res1)
   user  system elapsed
  32.59   16.04   48.64
>
> dise1=data.frame(Inicio=inicio,Longitud=longitud,Tipo=tipo)
> rownames(dise1)=dise$Campo
> res2=system.time((r2=leerFichero("UP-NAL.F151231",dise1,tipoNumerico="N")))
> print(res2)
   user  system elapsed
  10.39    0.05   10.45
> dim(r1)
[1] 152551     19
> dim(r2)
[1] 152551     19
> 
```

Como se ve en la salida que se acompaña el tiempo total empleado por la función es menos de la cuarta parte del empleado por la **read.fwf**.

Si se realiza lo mismo con el fichero de Tramos del callejero, con 2.102.759 registros y 39 campos, la diferencia es más notable, un tiempo total de 105.27 segundos contra 853.14 segundos de read.fwf. Todo esto en un ordenador con Windows 7 de 64 bits, 4Gb de RAM y un procesador I5.

Por otro lado está función permite, entre otras cosas, leer trozos de campos, a la vez que los campos enteros, cambiar el orden y leer números con decimales separados por comas y separadores de miles.

Las funciones utilizadas son las siguientes:

```
trim<-function(x){   gsub("^ *| *$","",x) }

extraer<-function(campo,dise,lineas,
                 campoInicio="Inicio",campoFin="Fin",campoTipo="Tipo",
                 campoLongitud="Longitud",
                 campoDecimales="Decimales",calcularDecimales=TRUE,
                 separadorDecimales=NULL,marcaMiles=NULL,
                 nas=".",tipoNumerico="numeric"){
   if(!is.null(campoFin) & (campoFin %in% colnames(dise))){
     res=substr(lineas,dise[campo,campoInicio],dise[campo,campoFin])
   }else{
     res=substr(lineas,dise[campo,campoInicio],dise[campo,campoInicio]+dise[campo,campoLongitud]-1)
   }
   res[trim(res) %in% nas]=NA
   if(as.character(dise[campo,campoTipo]) %in% tipoNumerico){
     res[trim(res) %in% nas]=NA
     if(!is.null(marcaMiles)) gsub(marcaMiles,"",res)
     if(is.null(marcaMiles) & !is.null(separadorDecimales)){
        if(separadorDecimales!=",") gsub(",","",res)
     }
     if(!is.null(separadorDecimales)) gsub(separadorDecimales,".",res)
     res=as.numeric(res)
     if(calcularDecimales){
       if(!is.null(campoDecimales) & (campoDecimales %in% colnames(dise)) ){
         if(!is.na(dise[campo,campoDecimales])) res=res/10^dise[campo,campoDecimales]
       }
     }
   }
   return(res)
}

leerFichero=function(fichero,dise,campoNombre=NULL,...){
   lineas=readLines(fichero,warn=FALSE)
   if(!is.null(campoNombre)) rownames(dise)=as.character(dise[,campoNombre])
   dat1=do.call(data.frame,lapply(rownames(dise),extraer,dise,lineas,...))
   colnames(dat1)=rownames(dise)
   return(dat1)
}
```

La función **extraer**, que es la base del proceso, permite dar el nombre de los campos con los diversos elementos, así como la forma de indicar que campos son numéricos y los separadores de miles y decimales utilizados. Asimismo, se puede poner como se marcan los valores faltantes, por defecto se utiliza un punto.

INDICADOR DE CONFIANZA DEL CONSUMIDOR

El Indicador de Confianza del Consumidor es una operación estadística cuyo objetivo consiste en conocer el grado de confianza de los consumidores a través de sus expectativas personales de gasto y de su percepción de la situación económica actual y futura.

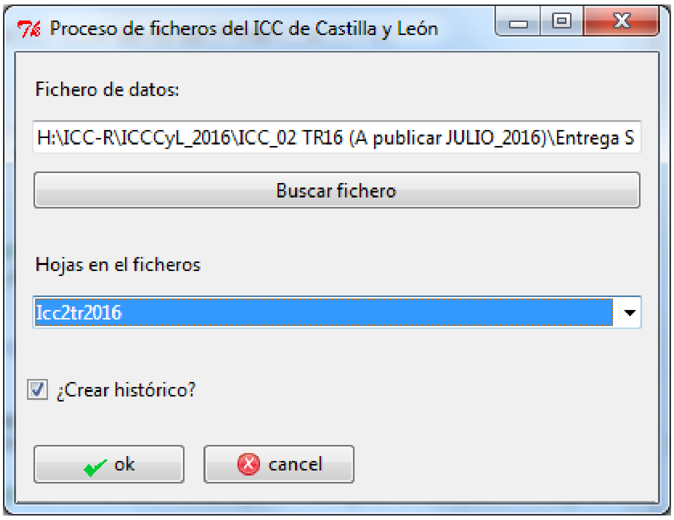
Esta operación se viene realizando desde el año 2008. En ella se entrevista, trimestralmente, a 1000 personas, a las que se realizan una serie de preguntas que se contestan con un máximo de 4 opciones (Mejor, igual, peor, NS/NC), que es con las que se calculan los saldos que dan lugar a los indicadores. Contando las variables de estratificación y algunas adicionales informativas, de cada encuestado se tienen 43 variables. Por lo tanto es un conjunto pequeño de datos, que se podría tratar con cualquier programa, incluso Excel.

Originalmente se tenía un programa desarrollado en SAS, pero al surgir la necesidad de actualizarlo y adaptarlo a nuevas formas de presentación, se pensó en desarrollar la aplicación en R. Desde que se desarrolló se han utilizado ambas en paralelo a modo de comprobación.

Uno de los problemas que había que solucionar era el de los redondeos, ya que el dar los indicadores con una cifra decimal en algunos casos los datos no casaban y se ajustaban a mano.

En la aplicación desarrollada en R se ha tratado dicho problema, de forma que las tablas obtenidas todo suma. La forma de hacerlo ha sido efectuar los redondeos en los indicadores antes de obtener los resultados que se calculan a partir de ellos.

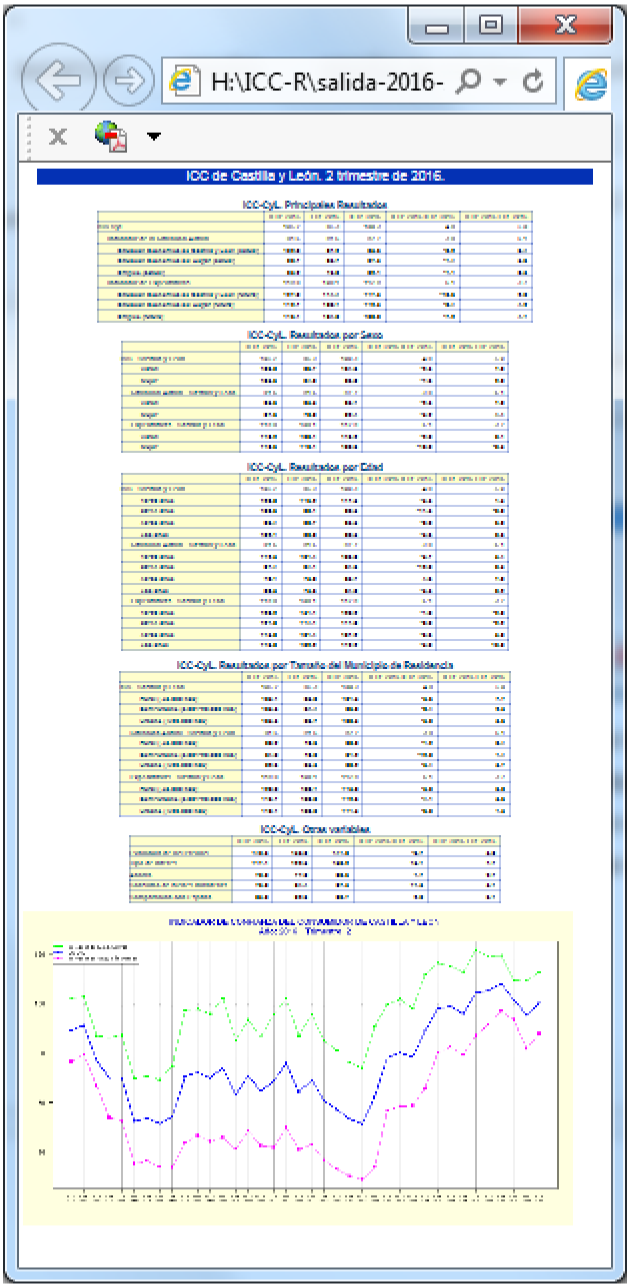
Las personas que realizan esta operación no son especialistas en programación, ni en SAS, ni en R, por lo que se ha diseñado un interfaz en R para que no haya que modificar nada en el programa al correrlo los sucesivos trimestres.

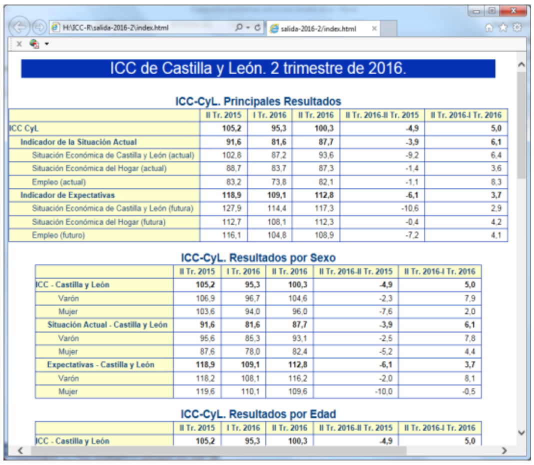


En esta ventana sólo hay que elegir el libro de Excel y la hoja del mismo en la que están los datos. A mayores se puede decidir crear un fichero con los datos históricos o no, simplemente con el fin de realizar pruebas sin machacar el fichero de datos históricos que se habitualmente se crea y que sirve para calcular las variaciones y dibujar las series.

Para que el uso de R sea más transparente se utiliza un acceso directo para ejecutar el programa en modo de línea de comando, por lo que el usuario no se enfrenta con R en ningún momento.

En este caso se ha optado por obtener una salida en HTML, que fácilmente se puede copiar a cualquier aplicación. Las tablas ya se presentan formateadas de acuerdo a los estándares de nuestra Dirección General. Salida que se presenta al terminar la ejecución, y que también se guarda en el directorio correspondiente.





H:\ICC-R\salida-2016-2\index.html

salida-2016-2/index.html

ICC de Castilla y León. 2 trimestre de 2016.

ICC-CyL. Principales Resultados

	II Tr. 2015	I Tr. 2016	II Tr. 2016	II Tr. 2016-II Tr. 2015	II Tr. 2016-I Tr. 2016
ICC CyL	**105,2**	**95,3**	**100,3**	**-4,9**	**5,0**
Indicador de la Situación Actual	**91,6**	**81,6**	**87,7**	**-3,9**	**6,1**
Situación Económica de Castilla y León (actual)	102,8	87,2	93,6	-9,2	6,4
Situación Económica del Hogar (actual)	88,7	83,7	87,3	-1,4	3,6
Empleo (actual)	83,2	73,8	82,1	-1,1	8,3
Indicador de Expectativas	**118,9**	**109,1**	**112,8**	**-6,1**	**3,7**
Situación Económica de Castilla y León (futura)	127,9	114,4	117,3	-10,6	2,9
Situación Económica del Hogar (futura)	112,7	108,1	112,3	-0,4	4,2
Empleo (futuro)	116,1	104,8	108,9	-7,2	4,1

ICC-CyL. Resultados por Sexo

	II Tr. 2015	I Tr. 2016	II Tr. 2016	II Tr. 2016-II Tr. 2015	II Tr. 2016-I Tr. 2016
ICC - Castilla y León	**105,2**	**95,3**	**100,3**	**-4,9**	**5,0**
Varón	106,9	96,7	104,6	-2,3	7,9
Mujer	103,6	94,0	96,0	-7,6	2,0
Situación Actual - Castilla y León	**91,6**	**81,6**	**87,7**	**-3,9**	**6,1**
Varón	95,6	85,3	93,1	-2,5	7,8
Mujer	87,6	78,0	82,4	-5,2	4,4
Expectativas - Castilla y León	**118,9**	**109,1**	**112,8**	**-6,1**	**3,7**
Varón	118,2	108,1	116,2	-2,0	8,1
Mujer	119,6	110,1	109,6	-10,0	-0,5

ICC-CyL. Resultados por Edad

	II Tr. 2015	I Tr. 2016	II Tr. 2016	II Tr. 2016-II Tr. 2015	II Tr. 2016-I Tr. 2016
ICC - Castilla y León	**105,2**	**95,3**	**100,3**	**-4,9**	**5,0**

El proceso en R tarda bastante menos que el correspondiente en SAS y la salida permite una más rápida creación de los informes y documentos.

PIRÁMIDES DE POBLACIÓN

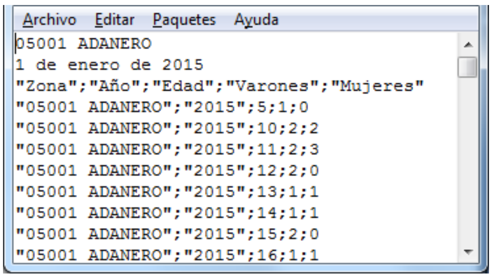
Los gráficos son un elemento importante a la hora de presentar la información estadística, por lo que R, con todas las posibilidades gráficas que tiene, se convierte en una herramienta imprescindible a la hora de realizarlos, sobre todo cuando se quiere hacer algo personalizado, que no está disponible habitualmente en otras aplicaciones.

Aunque las pirámides de población es un tipo de gráfico ampliamente utilizado y disponible en muchas aplicaciones, bien directamente o mediante añadidos, siempre se quiere darlas un toque particular, por lo que hace tiempo que venimos haciéndolas utilizando R.

Los sucesivos cambios en los programa utilizados, nos han llevado a una versión que pretende ser lo bastante flexible para realizar cualquiera de los tipos que utilizamos y a la vez se ha intentado que puede ser utilizada por personas que no sepan nada de R, para lo que se hecho una interfaz para poder trabajar de forma sencilla. También se puede iniciar el programa mediante un acceso directo, de forma que su base en R es totalmente transparente al usuario.

El programa completo obtiene los datos de ficheros de texto, aunque en la función que realmente dibuja las pirámides los datos se han de pasar como una matriz o data.frame y por lo tanto la ampliación para obtener datos en otros formatos se ha de hacer solamente en el interfaz.

El fichero de datos original ha de tener una columna con la edad, otra con el número de hombres y otra con el de mujeres. Además puede tener una cabecera con información sobre la fecha y el lugar de los datos. Se ha previsto que los nombres de las columnas puedan variar (en singular y en plural y en algunos idiomas distintos, de momento español, inglés y francés) y la lista de posibilidades es fácil de ampliar. Una muestra de un fichero es:


Archivo Editar Paquetes Ayuda

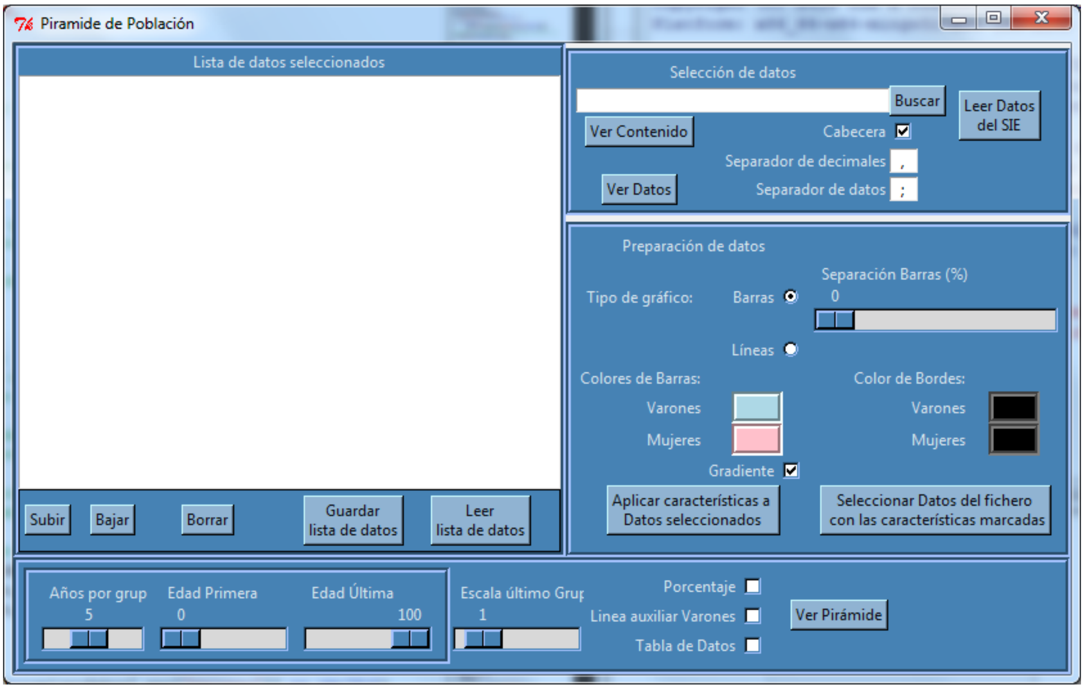
```
05001 ADANERO
1 de enero de 2015
"Zona";"Año";"Edad";"Varones";"Mujeres"
"05001 ADANERO";"2015";5;1;0
"05001 ADANERO";"2015";10;2;2
"05001 ADANERO";"2015";11;2;3
"05001 ADANERO";"2015";12;2;0
"05001 ADANERO";"2015";13;1;1
"05001 ADANERO";"2015";14;1;1
"05001 ADANERO";"2015";15;2;0
"05001 ADANERO";"2015";16;1;1
```

En este ejemplo se puede ver que hay una columna adicional, que no se va a utilizar.

También se ha habilitado la posibilidad de obtener los datos directamente del Sistema de Información Estadística de la Junta de Castilla y León, del cual se pueden obtener a nivel de municipio, provincia o comunidad. Los datos se guardan en un fichero en el formato requerido para después ser utilizado.

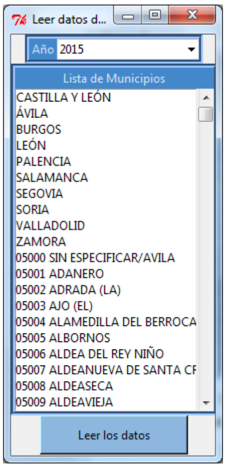
Los datos en el fichero pueden venir ya agrupados o en edades individuales, en cuyo caso el programa puede establecer los grupos que se consideren oportunos.

El interfaz que se ha desarrollado se muestra a continuación.



En la parte superior derecha se puede seleccionar el fichero con los datos, verlo tal cual (Ver contenido), indicar sus características y ver como quedarían los datos una vez leídos.

También se pueden leer los datos del SIE (Sistema de Información Estadística de Castilla y León). En este caso se accede a otra ventana para seleccionar el municipio, la provincia o la Comunidad.



Cuando se selecciona y se leen los datos, estos se guardan en un fichero en el formato adecuado y el nombre se pone en el campo correspondiente (Selección de datos).

Debajo de esta zona está la que sirve para indicar las características de la pirámide: tipo, colores, …

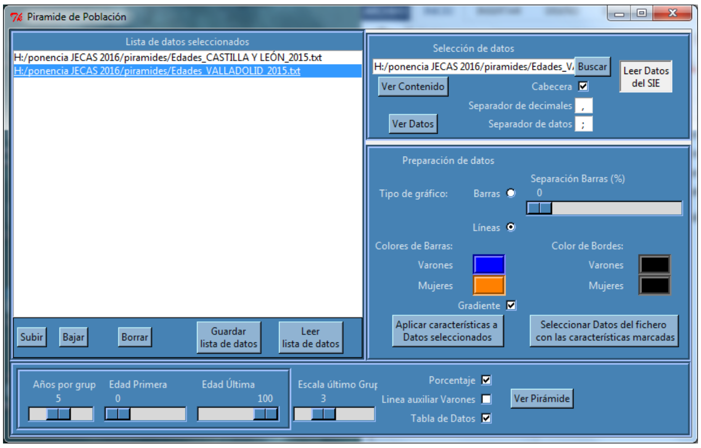
Estas características se aplican a los datos seleccionados en la parte izquierda (Botón de la izquierda) o a los seleccionados en la parte superior y a la vez pasar estos a la parte izquierda (Botón de la derecha).

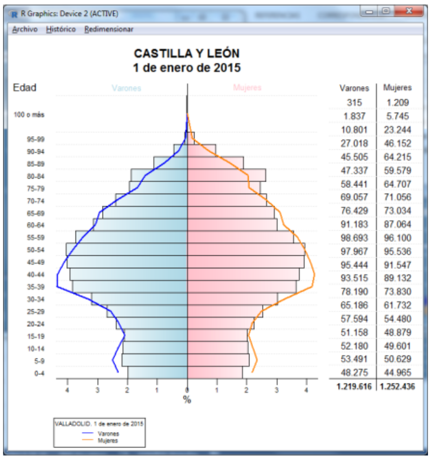
En el cuadro de la parte superior izquierda aparecerán los datos que se han idos seleccionando. El primero de ellos es el principal y en base a él se establecen los límites del gráfico de la pirámide, el resto se superponen sobre ella.

Es posible alterar el orden y las características (tipo y color) de los datos en dicha lista, así como guardar la lista o recuperar una lista guardada.

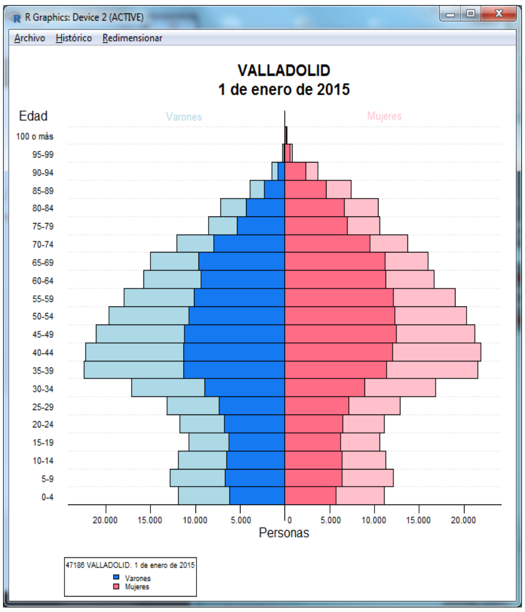
Con los datos leídos se puede proceder a dibujar la pirámide, de la que se puede seleccionar algunas características adicionales en la parte inferior, como son el número de años por grupo (si es que los datos están en años individuales), la primera y última edad a considerar, si el último grupo se dibuja y cuál será su alto, si se van a utilizar porcentajes o valores absolutos, si se dibujará la línea auxiliar de varones o si se va a mostrar la tabla de datos (sólo del primer conjunto de datos).

Hecho todo ello se puede ya mandar dibujarla.





También es sencillo obtener otra para comparar la población que vive en la provincia con la de la capital



La imagen, como cualquiera de R, se puede copiar al portapapeles o guardar en distintos formatos.

Entre todas las funciones del programa desarrollado la principal es aquella que dibuja las pirámides, que puede ser utilizada en cualquier otro contexto.

La sintaxis de la misma es:

```
piramide=function(datos,datosaux=NULL,grupos=5,tipo="Barras",ultimoGrupo=0,
                colores=c(Varones="lightblue",Mujeres="pink"),gradiente=TRUE,
                borde=par("fg"),separacion=0,fecha="",lugar="",
                porcentaje=FALSE,grid=TRUE,lineaAuxVarones=FALSE,
                tablaDatos=FALSE)
```

Se ve que los parámetros recogen los aspectos que se han comentado anteriormente.

datos ha de ser un data.frame con los datos principales (al menos ha de tener una columna de edad, otra de varones y otra de mujeres), que además puede tener una serie de atributos con las características de la pirámide. Si tiene los atributos éstos sustituyen a lo que se pase como parámetros a la función. Es decir que si datos tiene un atributo que se llama *colores*, lo que tenga este atributo sustituye a lo que se pase en el parámetro *colores*.

datosaux es una lista de data.frame con los datos para el resto de las pirámides a superponer, que también pueden tener atributos con sus características.

Si en *datos* se tiene un conjunto de datos, con *piramide(datos)* se hace la pirámide, si se quiere guardar directamente en un fichero, por ejemplo en pdf, se puede hacer:

```
pdf("prueba.pdf")
piramide(datos)
dev.off()
```

CONCLUSIONES

Cuando se realiza un trabajo para profesionales no estadísticos a veces sorprende que la pregunta que hacen es *"¿Con qué programa se han hecho los cálculos?"*, como si eso fuese un aspecto fundamental para la validación de los resultados, cuando en realidad lo importante es como se han hecho., los algoritmos empleados y su adecuación a los datos de que se dispone.

Es cierto que no todos los programas utilizan los mismos algoritmos, pero la mayoría de los habitualmente usados permiten la elección entre distintas alternativas. Lo bueno de programas como R es la posibilidad de utilizar un gran número de funciones para cada problema y la facilidad de crear las nuestras si no encontramos la que nos interesa. Otros programas estadísticos también tienen esta posibilidad, pero generalmente su precio mucho mayor.

En consecuencia, la conclusión más destacable es que lo importante a la hora de tratar estadísticamente datos no es la *MARCA* del programa empleado, si no la calidad estadística de las personas que lo utilizan, aunque el programa utilizado no haya costado nada o sea de propósito general, como Excel.

En relación con el coste de una solución a un problema, éste no sólo es la parte de la licencia del programa, sino también el tiempo empleado en el desarrollo de las aplicaciones. En el caso de los programas comentados en esta ponencia y de los de la mayoría de los empleados en nuestra Dirección General, el tiempo de desarrollo no ha sido grande y la mayoría se han hecho en base a ampliar pequeñas soluciones para otros problemas.

Y por último decir que los grandes programas siguen siendo útiles para ciertas cosas, ya que con ellos se pueden abordar soluciones difíciles de implementar con otro tipo de programas.