

Cargando datos en R

Sitio: [Plataforma de Formación On line del Instituto Andaluz de
Administración Pública](#)
Curso: (I22F-PT05) Entorno de Programación R
Libro: Cargando datos en R

Imprimido por: ALFONSO LUIS MONTEJO RAEZ
Día: lunes, 11 de abril de 2022, 10:54

Tabla de contenidos

1. Introducción

- 1.1. Archivos de texto CSV o DSV
- 1.2. Archivos de texto desde Internet
- 1.3. Archivos de texto. Ancho fijo
- 1.4. Archivos Excel
- 1.5. Archivos ODS
- 1.6. Archivos DBF
- 1.7. Archivos PX
- 1.8. Archivos PDF

1. Introducción

El formato de datos nativo de R es el **RData** pero, en general, no dispondremos de los datos directamente en él y tendremos que cargarlos en R desde otras fuentes.

Para realizar la carga de datos de otros formatos podemos hacerlo de dos formas: **usando el asistente de RStudio** o **utilizando paquetes específicos** para determinados tipos de ficheros.

En el primer caso, RStudio cuenta con un asistente que nos ayuda en la tarea de importar datos desde la pestaña **Environment**. El asistente carga los datos con la ayuda de varios paquetes que vienen instalados en R.

Los formatos que acepta el asistente son:

- **Paquetes base o readr:** Formato de texto (csv, dsv).
- **Paquete readxl.** Archivos de Excel.
- **Paquete Haven.** Archivos de SPSS, SAS y Stata.

Para los tipos de ficheros que no vienen en el asistente tenemos que usar paquetes específicos, que tendremos que instalar y cargar antes de poder usarlos.

Nosotros vamos a ver la carga de los siguientes formatos:

- **Paquete readODS.** Archivos de Calc de LibreOffice.
- **Paquete foreign.** Archivos de tipo DBF.
- **Paquete pxR.** Archivos de tipo PcAxis.
- **Paquete tabulizer.** Archivos PDF.

Para ver como se cargan los archivos de los distintos tipos vamos a usar ejemplos para cada uno de ellos.

1.1. Archivos de texto CSV o DSV

El formato más habitual para el intercambio de datos es el texto plano.

En este tipo de ficheros los datos de cada variable vienen separados por un carácter, que suele ser una coma o un punto y coma. Los primeros se denominan **ficheros csv** (comma separated values) y los segundos **ficheros dsv** (delimited separated values).

Para cargar este tipo de ficheros en RStudio, podemos hacerlo por dos vías: o usando el asistente o las órdenes **read.csv()** para el formato csv y **read.csv2()** para el formato dsv.

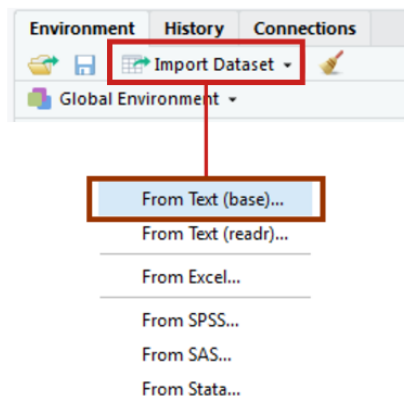
En la orden podemos especificar, entre otros, los siguientes parámetros:

- Valor del separador (, o ;)
- Valor de los decimales (. o ,)
- Cabecera de datos

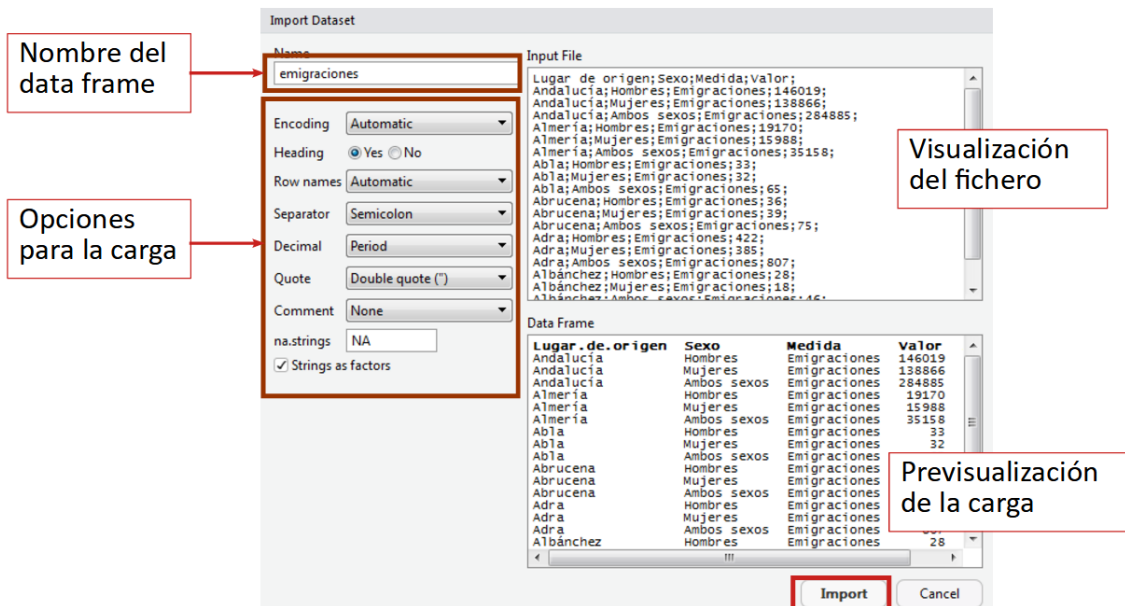
Ejemplo

Vamos a usar el asistente de RStudio para cargar el fichero **emigraciones.txt** con datos de las emigraciones desde Andalucía por municipio y sexo. Este archivo tiene los valores separados por ; y los nombres de las variables están en la primera fila.

Para cargarlo tenemos que pulsar **Import Dataset** de la pestaña **Environment** donde elegimos el tipo de archivo a cargar, que en nuestro caso es de Texto.



Después navegamos para buscar el archivo a cargar en su localización (normalmente en el disco duro o un pendrive) y se abre el asistente donde podremos elegir las opciones adecuadas para la carga del archivo.

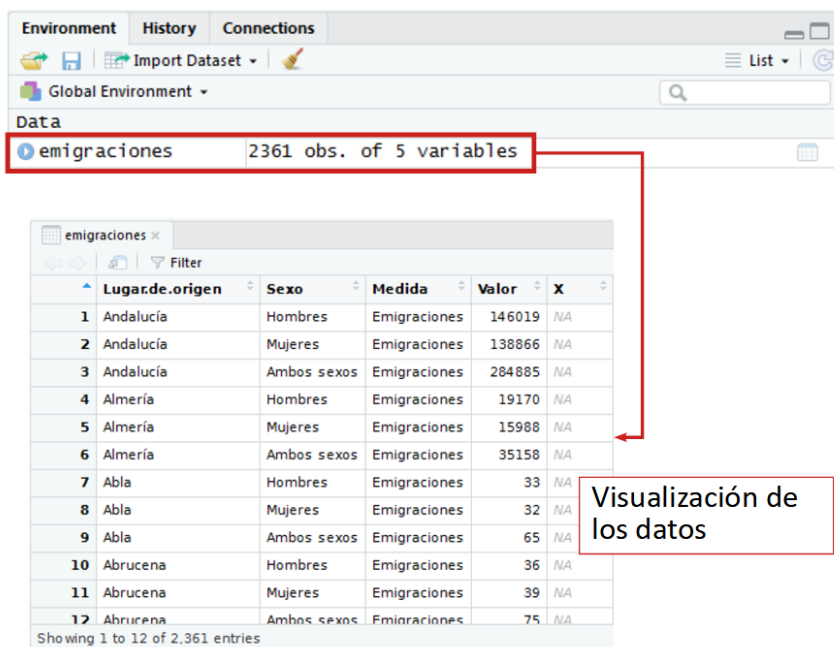


Los parámetros que tenemos que determinar son:

- **Nombre del data frame** que se va a crear en R cuando finalice la carga.
- **Código de caracteres del fichero (Encoding)**. Cual es el sistema de caracteres empleado en el fichero (normalmente UTF-8 o ANSI).
- **Separador de los valores (Separator)**. En nuestro caso los valores están separados por ;
- **Nombres de las variables (Header)**. Si la primera línea del fichero contiene el nombre de las variables.
- **Separador de decimales (Decimal)**. En el caso de datos numéricos con decimales podemos elegir si es un punto o una coma el caracter que separa la parte decimal.

Cuando terminamos de elegir todas las opciones pulsamos el botón **Import**.

Una vez cargado el fichero podemos ver el objeto **emigraciones** creado en la pestaña de Environment.



1.2. Archivos de texto desde Internet

La ubicación del fichero no tiene porque ser un fichero en local si no que puede ser también un fichero que esté en internet.

En este caso podemos cargar el fichero de datos usando la orden **read.csv()** indicando, como origen del archivo, la URL donde se encuentra el fichero.

Ejemplo

En este caso vamos a cargar un fichero de texto que contiene los datos de paro registrado por municipios para el año 2006 que se encuentra en la [URL](https://sede.sepe.gob.es/es/portalttrabajo/resources/sede/datos_abiertos/datos/Paro_por_municipios_2006_csv.csv).

https://sede.sepe.gob.es/es/portalttrabajo/resources/sede/datos_abiertos/datos/Paro_por_municipios_2006_csv.csv

La orden que vamos a usar para cargar el archivo desde internet sería:

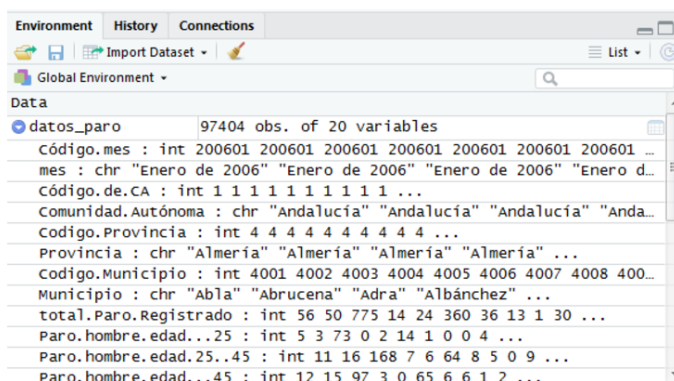
```
> read.csv(url, sep=';', header=T, skip=1, as.is=T) -> datos_paro
```

Opciones empleadas:

- **sep=';'** Usamos separador ;
- **header=T** Los datos tienen el nombre de las variables en la cabecera
- **skip=1** Que se salte la primera línea (que es el título de la tabla)
- **as.is=T** Que mantenga el formato original de los datos.

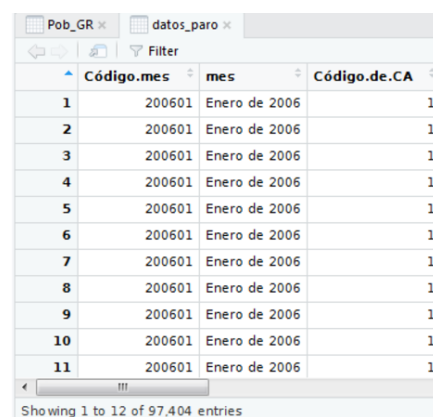
Url es un objeto de texto donde hemos almacenado la dirección desde donde se tienen que bajar los datos.

Una vez cargados los datos se crea el objeto **datos_paro** en la pestaña de **Environment**.



Variable	Value 1	Value 2	Value 3	Value 4	Value 5
Código.mes	200601	200601	200601	200601	200601
mes	"Enero de 2006"	"Enero de 2006"	"Enero de 2006"	"Enero de 2006"	"Enero de 2006"
Código.de.CA	1	1	1	1	1
Comunidad.Autónoma	"Andalucía"	"Andalucía"	"Andalucía"	"Andalucía"	"Andalucía"
Código.Provincia	4	4	4	4	4
Provincia	"Almería"	"Almería"	"Almería"	"Almería"	"Almería"
Código.Municipio	4001	4002	4003	4004	4005
Municipio	"Abla"	"Abrucena"	"Adra"	"Albánchez"	"Albánchez"
total.Paro.Registrado	56	50	775	14	24
Paro.hombre.edad...25	5	3	73	0	2
Paro.hombre.edad...45	11	16	168	7	6
Paro.hombre.edad...45	12	15	97	3	0

Visualización de datos



	Código.mes	mes	Código.de.CA
1	200601	Enero de 2006	1
2	200601	Enero de 2006	1
3	200601	Enero de 2006	1
4	200601	Enero de 2006	1
5	200601	Enero de 2006	1
6	200601	Enero de 2006	1
7	200601	Enero de 2006	1
8	200601	Enero de 2006	1
9	200601	Enero de 2006	1
10	200601	Enero de 2006	1
11	200601	Enero de 2006	1

Se han cargado 97.404 registros con información de 20 variables.

1.3. Archivos de texto. Ancho fijo

Los ficheros de **ancho fijo** tienen la particularidad de que los valores de las variables no están separados por ningún carácter ya que vienen determinados por la posición que ocupa cada valor dentro de una fila.

Para poder cargar este tipo de datos es necesario conocer previamente el **diseño de registro**. Éste es un documento donde se indican las variables que contiene el fichero, junto con la longitud o posición que ocupan y la codificación usada para las variables categóricas.

Este formato es el usado para la difusión de microdatos de encuestas e información estadística oficial.

Para leer un fichero de este tipo se usa la orden **read.fwf()**.

Ejemplo

Vamos a importar los microdatos de la estadística de variaciones residenciales del INE que se encuentran en el archivo EVR_2016.

Lo primero que necesitamos es conocer el diseño de registro para saber que variables contiene el archivo y cual es su ubicación (Longitud de cada campo).

Diseño de registro

Longitud: 37						
Variable	Descripción	Longitud	Inicio	Fin	Observaciones	Valores
SEXO	Sexo	1	1	1		1--> Hombre y 6-->Mujer
PROVNAC	Provincia de nacimiento	2	2	3	Los nacidos en el extranjero se identifican con el código '66'.	01.02....52 -> Consultar listado PROV
MUNINAC	Municipio o país de nacimiento	3	4	6	Para los nacidos en España se codifica el municipio de nacimiento y para los nacidos en el extranjero se codifica el país de nacimiento. Se blanquea el código de los municipios de nacimiento con población inferior a 10.000 habitantes.	Consultar listado MUNI1
EDAD	Edad en el momento de la VR	3	7	9	Edad en el momento de la variación. Se ha calculado para todos los años teniendo en cuenta los meses de nacimiento y variación, no los días.	
MESNAC	Mes de nacimiento	2	10	11		
AÑONAC	Año de nacimiento	4	12	15		
CNAC	Nacionalidad	3	16	18		Consultar listado CNAC

La orden para cargar el archivo es **read.fwf()**,




```
> read.fwf('E:\\Curso R IECA\\Temario\\Sesion2\\Datos\\Variaciones\\EVR_2016',
+          c(1,2,3,3,2,4,3,2,3,2,4,2,3,1,1,1),
+          col.names=c('SEXO', 'PROVNAC', 'MUNINAC', 'EDAD', 'MESNAC', 'ANONAC', 'CNAC', 'PROVALTA', 'MUNIALTA',
+                      'MESVAR', 'ANOVAR', 'PROVBAJA', 'MUNIBAJA', 'TAMUALTA', 'TAMUBAJA', 'TAMUNACI'),
+          colClasses=rep(c('numeric', 'character'), c(1,15)),
+          header=F)->EVR
```

con los siguientes parámetros:

- **Ubicación del fichero a cargar.**
- **Valores de longitud de los campos.** Esta información la encontramos en el diseño de registro e indica la longitud que ocupa cada variable dentro del fichero.
- **Nombres de las variables.** Listado con los nombres que le queremos poner a las variables.
- **Tipo de dato de cada variable.** Tipo de dato (normalmente numerico o carácter).
- **Header.** Indica si el fichero cuenta en su primera con una cabecera con información. En nuestro caso no la lleva y usamos el F (False).

Una vez cargado el fichero podemos ver el objeto **EVR** creado en la pestaña **Environment**.

The screenshot shows the RStudio interface. The top pane is the Environment pane, which displays two data frames: 'emigraciones' and 'EVR'. The 'emigraciones' data frame has 2361 observations and 5 variables. The 'EVR' data frame has 157585 observations and 16 variables. The 'EVR' data frame is expanded, showing its variables: SEXO, PROVINCIA, MUNICIPIO, EDAD, MES, ANO, CATEGORIA, PROVALTA, MUNIALTA, MESVAR, ANOVAR, and PROVALTA. The bottom pane is the Console, which is empty. The left pane is the Source pane, which is empty. The right pane is the Environment pane, which is empty.

Environment	History	Connections
   Import Dataset		
Global Environment		
Data		
emigraciones	2361 obs. of 5 variables	
EVR	157585 obs. of 16 variables	
SEXO : num 6 1 1 1 1 1 6 6 1 ...		
PROVINCIA : chr "01" "01" "01" "01" ...		
MUNICIPIO : chr " " " " " " " " " " ...		
EDAD : chr "024" "002" "003" "002" ...		
MES : chr "10" "04" "07" "12" ...		
ANO : chr "1991" "2014" "2013" "2013" ...		
CATEGORIA : chr "108" "108" "108" "108" ...		
PROVALTA : chr "01" "01" "01" "01" ...		
MUNIALTA : chr "059" "059" "059" "059" ...		
MESVAR : chr "01" "09" "12" "07" ...		
ANOVAR : chr "2016" "2016" "2016" "2016" ...		
PROVALTA : chr "01" "01" "01" "01" ...		

Este objeto tiene 157.585 registros y 16 variables.

1.4. Archivos Excel

Los **ficheros Excel** son propios de las hojas de cálculo de Microsoft Office y son muy usados para el intercambio y difusión de tablas de datos.

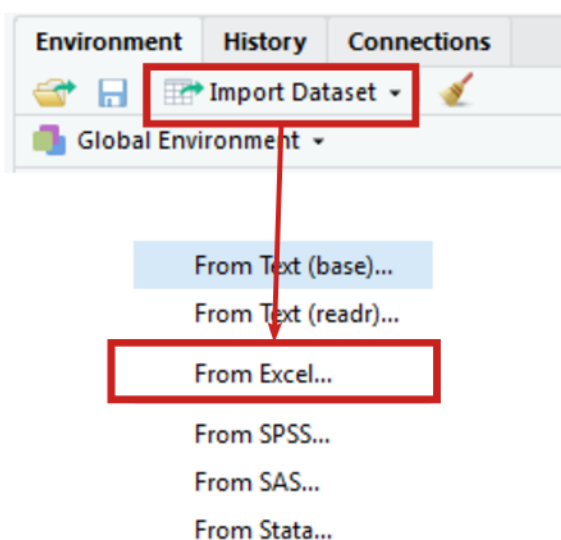
Para leer este tipo de formato necesitamos el **paquete readxl**, por lo que antes de usarlo, debemos instalarlo y cargarlo. La orden para leer este tipo de ficheros es **read_excel()**, que nos sirve para cargar archivos tanto xls como xlsx.

Otra opción para cargar estos ficheros es usar el asistente de RStudio para ficheros Excel.

Ejemplo

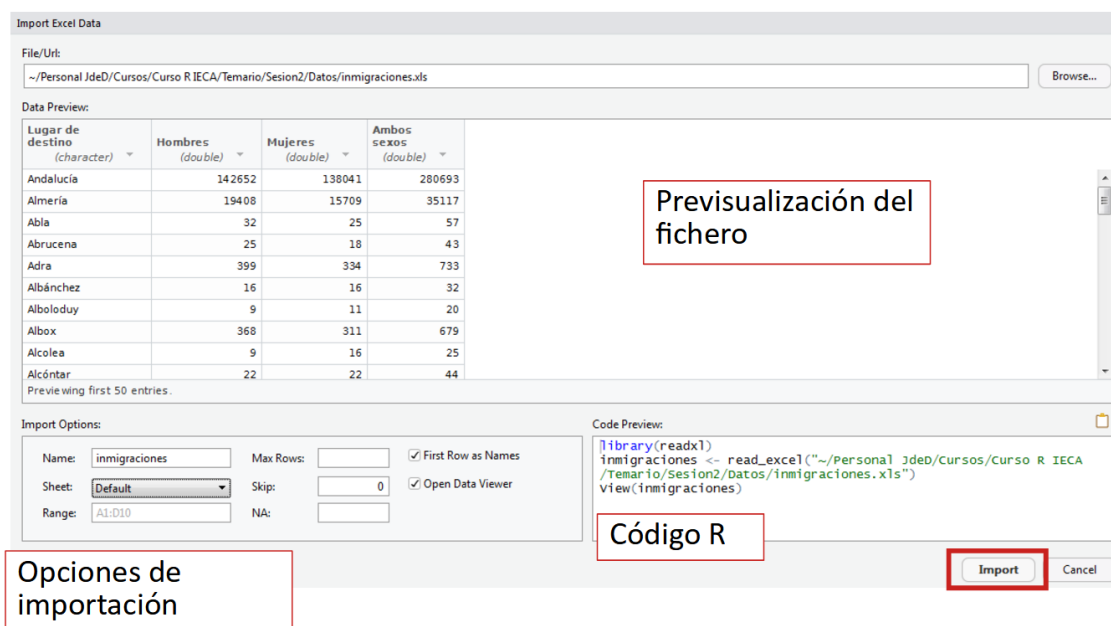
Vamos a usar el asistente de RStudio para cargar el fichero **inmigraciones.xls** que contiene los datos de inmigraciones por municipio y sexo hacia Andalucía para el año 2016.

Para cargarlo tenemos que pulsar **Import Dataset** de la pestaña **Environment** donde elegimos el tipo de archivo a cargar, que en nuestro caso es de Excel.



Una vez elegido el tipo de archivo, se abre el asistente donde podemos elegir las opciones adecuadas para la carga del archivo.

Lo primero es buscar el archivo a cargar pulsando el botón **Browse** y navegando para localizarlo en su ubicación (normalmente en el disco duro o un pendrive) y, a continuación, elegir las opciones adecuadas para la carga del archivo.



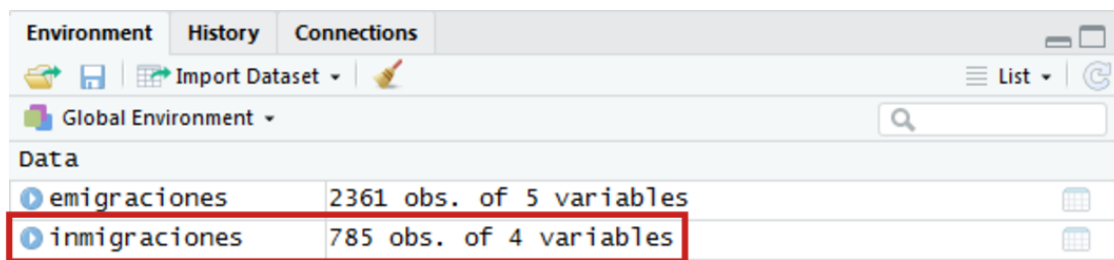
Los parámetros que tenemos que determinar son:

- **Nombre del data frame (Name)** que se va a crear en R cuando finalice la carga.
- **Hoja (Sheet)**. Si el fichero Excel tiene más de una hoja aquí podemos elegir cual de ellas queremos cargar.

En la ventana **Code Preview** podemos ver como se va construyendo la orden que vamos a usar para cargar el fichero Excel. En esta ventana nosotros podemos modificar y escribir las opciones que queramos para adecuarlas a nuestras necesidades.

Cuando terminamos de elegir todas las opciones pulsamos el botón **Import**.

Una vez cargado el fichero podemos ver el objeto **inmigraciones** creado en la pestaña de Environment.



Global Environment	
Data	
emigraciones	2361 obs. of 5 variables
inmigraciones	785 obs. of 4 variables



	Lugar de destino	Hombres	Mujeres	Ambos sexos
1	Andalucía	142652	138041	280693
2	Almería	19408	15709	35117
3	Abla	32	25	57
4	Abrucena	25	18	43
5	Adra	399	334	733
6	Albánchez	16	16	32
7	Alboloduy	9	11	20
8	Albox	368	311	679
9	Alcolea	9	16	25
10	Alcóntar	22	22	44
11	Alcudia de Monteagud	3	2	5

Showing 1 to 11 of 785 entries

Visualización de los datos

Este objeto cuenta con 785 registros y 4 variables.

1.5. Archivos ODS

Los ficheros de formato ods son propios de las hojas de cálculo de Libreoffice y Openoffice.

Para leer este tipo de formato necesitamos el **paquete readODS**, por lo que antes de usarlo, debemos instalarlo y cargarlo.

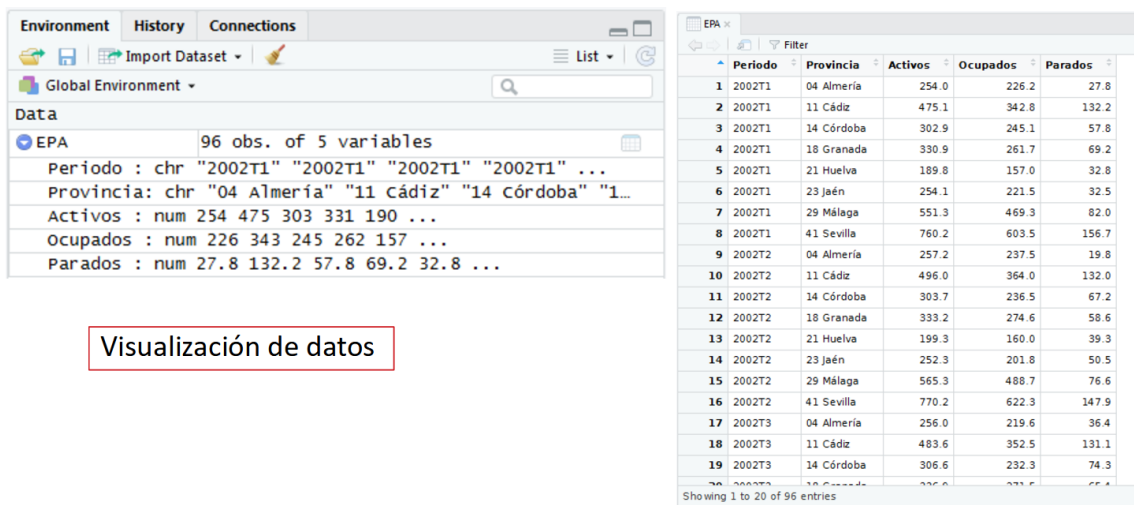
La orden para leer este tipo de ficheros es **read_ods()**.

Ejemplo

Vamos a cargar en R el fichero **EPA.ods** que contiene información sobre la Encuesta de Población Activa por provincias para Andalucía.

```
> library(readODS)
> read_ods('E:\\Curso R IECA\\Temario\\Sesion2\\Datos\\EPA.ods', col_names=T)->EPA
```

Para cargar el fichero ods simplemente tenemos que indicar la ruta donde se encuentra el fichero y usar el parámetro **col_names=T** que indica que la primera línea contiene el nombre de las variables.



The screenshot shows the R Studio interface. On the left, the 'Environment' pane displays the 'EPA' dataset with 96 observations and 5 variables. The variables are: Período (character), Provincia (character), Activos (numeric), Ocupados (numeric), and Parados (numeric). On the right, a preview of the data is shown as a table with 19 rows and 5 columns.

	Período	Provincia	Activos	Ocupados	Parados
1	2002T1	04 Almería	254.0	226.2	27.8
2	2002T1	11 Cádiz	475.1	342.8	132.2
3	2002T1	14 Córdoba	302.9	245.1	57.8
4	2002T1	18 Granada	330.9	261.7	69.2
5	2002T1	21 Huelva	189.8	157.0	32.8
6	2002T1	23 Jaén	254.1	221.5	32.5
7	2002T1	29 Málaga	551.3	469.3	82.0
8	2002T1	41 Sevilla	760.2	603.5	156.7
9	2002T2	04 Almería	257.2	237.5	19.8
10	2002T2	11 Cádiz	496.0	364.0	132.0
11	2002T2	14 Córdoba	303.7	236.5	67.2
12	2002T2	18 Granada	333.2	274.6	58.6
13	2002T2	21 Huelva	199.3	160.0	39.3
14	2002T2	23 Jaén	252.3	201.8	50.5
15	2002T2	29 Málaga	565.3	488.7	76.6
16	2002T2	41 Sevilla	770.2	622.3	147.9
17	2002T3	04 Almería	256.0	219.6	36.4
18	2002T3	11 Cádiz	483.6	352.5	131.1
19	2002T3	14 Córdoba	306.6	232.3	74.3

Visualización de datos

Una vez cargado el fichero se crea un objeto denominado **EPA** que cuenta con 96 registros y cinco variables.

1.6. Archivos DBF

Los ficheros en **formato dbf** son propios de las bases de datos dBase. Éste es usado por las capas cartográficas shape (shp) para almacenar la información alfanumérica.

Para leer este tipo de formato necesitamos el **paquete foreign**, por lo que antes de usarlo, debemos instalarlo y cargarlo.

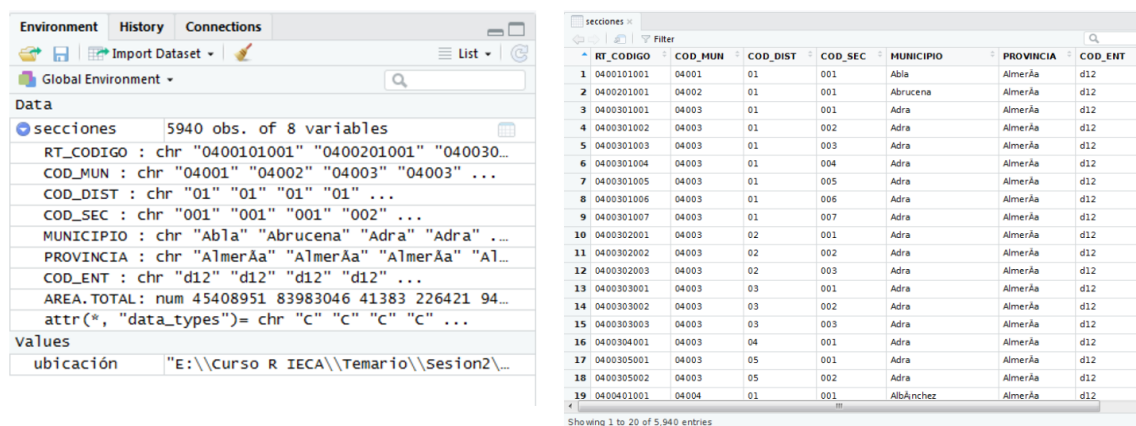
La orden para leer este tipo de ficheros es **read.dbf()**.

Ejemplo

Vamos a cargar en R la información alfanumérica de una capa cartográfica en formato shape, que contiene la información de las secciones censales de 2017.

```
> library(foreign)
> ubicación<-'E:\\Curso R IECA\\Temario\\Sesion2\\Datos\\2017\\sc_2017.dbf'
> read.dbf(ubicación,as.is=T)->secciones
```

Para cargar el fichero dbf simplemente tenemos que indicar la ruta donde se encuentra el fichero y usar el parámetro **as.is=T** para que respete el formato original de los datos.



	RT_CODIGO	COD_MUN	COD_DIST	COD_SEC	MUNICIPIO	PROVINCIA	COD_ENT
1	0400101001	04001	01	001	Abia	Almería	d12
2	0400201001	04002	01	001	Abrucena	Almería	d12
3	0400301001	04003	01	001	Adra	Almería	d12
4	0400301002	04003	01	002	Adra	Almería	d12
5	0400301003	04003	01	003	Adra	Almería	d12
6	0400301004	04003	01	004	Adra	Almería	d12
7	0400301005	04003	01	005	Adra	Almería	d12
8	0400301006	04003	01	006	Adra	Almería	d12
9	0400301007	04003	01	007	Adra	Almería	d12
10	0400302001	04003	02	001	Adra	Almería	d12
11	0400302002	04003	02	002	Adra	Almería	d12
12	0400302003	04003	02	003	Adra	Almería	d12
13	0400303001	04003	03	001	Adra	Almería	d12
14	0400303002	04003	03	002	Adra	Almería	d12
15	0400303003	04003	03	003	Adra	Almería	d12
16	0400304001	04003	04	001	Adra	Almería	d12
17	0400305001	04003	05	001	Adra	Almería	d12
18	0400305002	04003	05	002	Adra	Almería	d12
19	0400401001	04004	01	001	Albánchez	Almería	d12

Visualización de datos

Una vez cargado el fichero se crea un objeto denominado **secciones** que cuenta con 5.940 registros y ocho variables.

1.7. Archivos PX

Los ficheros en **formato px** son tablas de datos del programa PCAxis. Este formato es el que usan los organismos públicos de estadística, como el INE, para la descarga de tablas de datos.

Para leer este tipo de formato necesitamos el **paquete pxR**, por lo que antes de usarlo, debemos instalarlo y cargarlo.

La orden para leer este tipo de ficheros es **read.px()**.

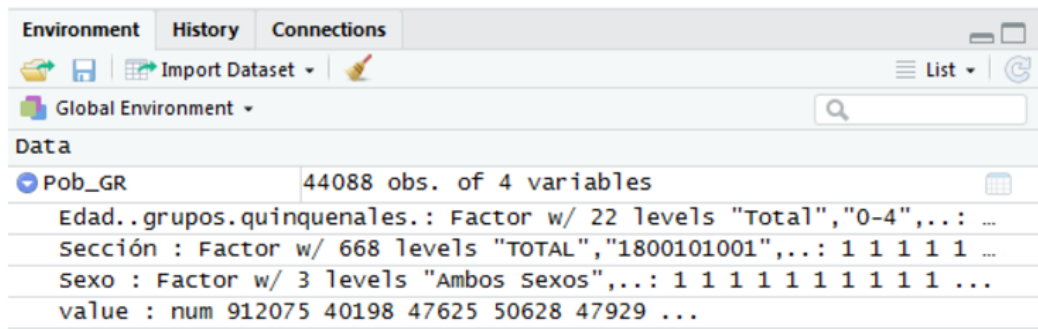
Ejemplo

Vamos a cargar en R la información del **fichero 1801_2018.px** que contiene información de población de la provincia de Granada por sexo, sección censal y grupo de edad.

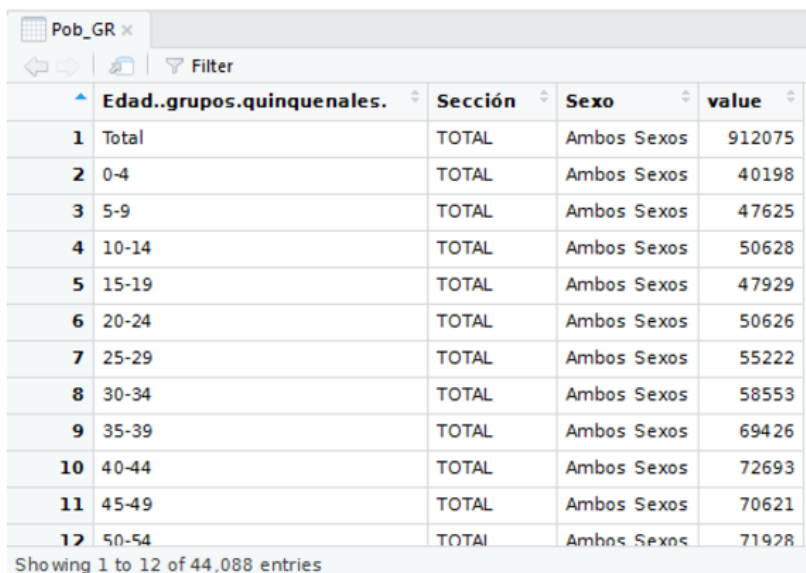
```
> library(pxR)
> read.px('F:\\Curso R IECA\\Temario\\Sesion2\\Datos\\1801_2018.px')->Pob_GR
> as.data.frame(Pob_GR)->Pob_GR
```

Para cargar el fichero px simplemente tenemos que indicar la ruta donde se encuentra el fichero.

Este paquete genera un tipo de dato propio, similar a una lista, que debemos convertir en un data frame para poder trabajar él. Para realizar la conversión usamos la orden **as.data.frame()**.



Environment	History	Connections
Global Environment		
Data		
Pob_GR 44088 obs. of 4 variables		
Edad..grupos.quinquenales.: Factor w/ 22 levels "Total","0-4",... :		
Sección : Factor w/ 668 levels "TOTAL","1800101001",... : 1 1 1 1 1 ...		
Sexo : Factor w/ 3 levels "Ambos Sexos",... : 1 1 1 1 1 1 1 1 1 1 ...		
value : num 912075 40198 47625 50628 47929 ...		



	Edad..grupos.quinquenales.	Sección	Sexo	value
1	Total	TOTAL	Ambos Sexos	912075
2	0-4	TOTAL	Ambos Sexos	40198
3	5-9	TOTAL	Ambos Sexos	47625
4	10-14	TOTAL	Ambos Sexos	50628
5	15-19	TOTAL	Ambos Sexos	47929
6	20-24	TOTAL	Ambos Sexos	50626
7	25-29	TOTAL	Ambos Sexos	55222
8	30-34	TOTAL	Ambos Sexos	58553
9	35-39	TOTAL	Ambos Sexos	69426
10	40-44	TOTAL	Ambos Sexos	72693
11	45-49	TOTAL	Ambos Sexos	70621
12	50-54	TOTAL	Ambos Sexos	71928

Showing 1 to 12 of 44,088 entries

Visualización
de datos

Una vez cargado el fichero se crea un objeto denominado **Pob_GR** que cuenta con 44.088 registros y cuatro variables.

1.8. Archivos PDF

En ocasiones hay organismos que difunden información estadística en formato PDF donde se incluyen tablas.

A través del **paquete tabulizer** se puede extraer de un documento pdf la información contenida en formato de tabla, siempre que ésta tenga una estructura reconocible.

Este paquete no está dentro de la base de R, por lo que tenemos que instalarlo y cargarlo antes de usarlo.

La orden básica para extraer información de un pdf es **extract_tables()** o **extract_text()**.

Ejemplo

Para mostrar el uso de este paquete vamos a utilizar como ejemplo la información que publica el Servicio Andaluz de Empleo en su [informe sobre la mujer y el mercado de trabajo andaluz 2018](#).

En las páginas 114 y 120 de ese informe existe información en formato de tabla que queremos extraer. En concreto se trata de la tabla de los Municipios con más mujeres extranjeras empadronadas y la tabla con los Países de origen de las mujeres contratadas.

Municipio	Mujeres extranjeras	% municipio sobre total
Málaga	22.020	7,20%
Marbella	17.602	5,76%
Sevilla	17.272	5,65%
Mijas	12.833	4,20%
Fuengirola	12.329	4,03%
Roquetas de Mar	11.101	3,63%
Ejido, El	10.981	3,59%
Almería	9.047	2,96%
Benalmádena	8.411	2,75%
Estepona	7.514	2,46%
Granada	7.238	2,37%
Torremolinos	6.879	2,25%
Níjar	5.310	1,74%
Córdoba	4.529	1,48%
Algeciras	4.273	1,40%
Vélez-Málaga	3.811	1,25%
Moguer	2.990	0,98%
Vícar	2.854	0,93%
Huelva	2.811	0,92%
Motril	2.790	0,91%

Vamos a usar la función **extract_tables()** a la que le podemos definir los siguientes parámetros:

- **ubicación**. Donde está el fichero a extraer. Puede ser una URL.
- **pages**. Página del documento donde está la tabla.
- **output**. Formato de la salida de los datos (data frame, matriz, etc).
- **encoding**. Juego de caracteres usado en la codificación del texto.

El resultado de esta función se almacena en una lista dejando cada tabla en un objeto dentro de la lista.

```
> library(tabulizer)
> extract_tables('F:\\Curso R IECA\\Temario\\Sesion3\\Datos\\Mujer_2018.pdf',
+               output = 'data.frame',encoding = 'UTF-8',pages=c(114,120))->datos
```

Environment History Connections

Global Environment

Data

datos List of 2

```

'data.frame': 20 obs. of 3 variables:
..$ Municipio : chr [1:20] "Málaga" "Marbella" "Sevilla" "Mijas" ...
..$ Mujeres.extranjeras : num [1:20] 22 17.6 17.3 12.8 12.3 ...
..$ X..municipio.sobre.total: chr [1:20] "7,20%" "5,76%" "5,65%" "..."
'data.frame': 25 obs. of 4 variables:
..$ X : chr [1:25] "Rumania" "Marruecos" "Bulgaria" "Reino Unido" ...
..$ País : logi [1:25] NA NA NA NA NA NA ...
..$ No.contratos: num [1:25] 74.22 48.3 9.19 6.13 6.12 ...
..$ Porcentaje : chr [1:25] "33,5%" "21,8%" "4,1%" "2,8%" ...

```

El resultado es una lista con dos data frames

Para obtener la información de la lista, donde están almacenadas las tablas, tenemos que acceder usando el doble corchete.

tabla_114

Filter

	Municipio	Mujeres.extranjeras	X..municipio.sobre.total
1	Málaga	22.020	7.20%
2	Marbella	17.602	5.76%
3	Sevilla	17.272	5.65%
4	Mijas	12.833	4.20%
5	Fuengirola	12.329	4.03%
6	Roquetas de Mar	11.101	3.63%
7	Ejido, El	10.981	3.59%
8	Almería	9.047	2.96%
9	Benalmádena	8.411	2.75%
10	Estepona	7.514	2.46%
11	Granada	7.238	2.37%
12	Torremolinos	6.879	2.25%

Showing 1 to 12 of 20 entries

```
> datos[[1]]->tabla_114
> datos[[2]]->tabla_120
```

```
> class(tabla_114)
[1] "data.frame"
> class(tabla_120)
[1] "data.frame"
```

Normalmente, a la información que se obtiene hay que hacerle una limpieza básica ya que se almacena exactamente como aparece en la tabla (con los símbolos de %, los puntos de los miles, etc.)