

# Importación de Datos en R

Jorge Meneses y Paulo Peña

# ¿Qué aprenderemos?

En esta sesión entenderemos los conceptos de paquetes y librerías. Asimismo, aprenderemos a escribir los códigos para importar datos y como corroborar los datos que hemos importado a nuestros proyectos.



# Paquetes más usados

- dplyr: para organizar y manipular datos
- ggplot2: para gener gráficos de alto impacto
- tidy: para manipular dataframes (bases de datos)
- openxlsx: para editar datos de microsoft excel

# Cómo utilizo un paquete

Usaremos como ejemplo el paquete dplyr, un paquete fundamental en el uso de R

```
## Para instalar un paquete  
install.packages("dplyr")
```

```
# Importante colocar el nombre del paquete entre comillas ""  
# si vamos a instalar dplyr, hay que colocarlo ("dplyr")
```

```
## Para usar un paquete
```

```
library(dplyr)
```

```
# Una vez instalado el paquete usamos la función library  
# Con library podremos utilizar las funciones en nuestra sesión de trabajo
```

# Importar datos

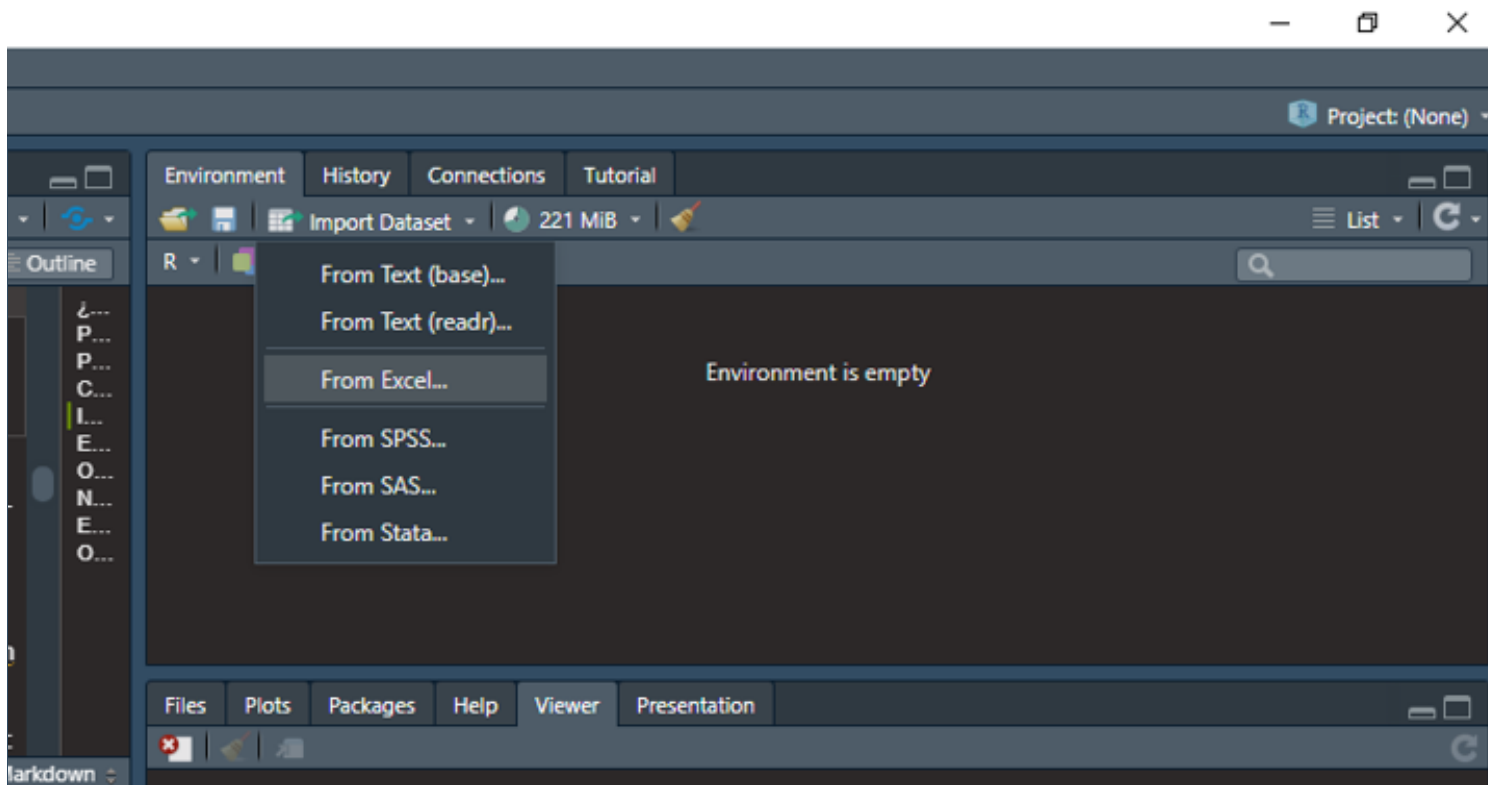
Cuando importamos un archivo estamos guardando contenido en nuestra sesión como un objeto. El procedimiento que usemos determinará el tipo de objeto creado.

De manera análoga, podemos exportar nuestros objetos de R a archivos.

Fuente: <https://bookdown.org/jboscomendoza/r-principiantes4/importar-y-exportar-datos.html>

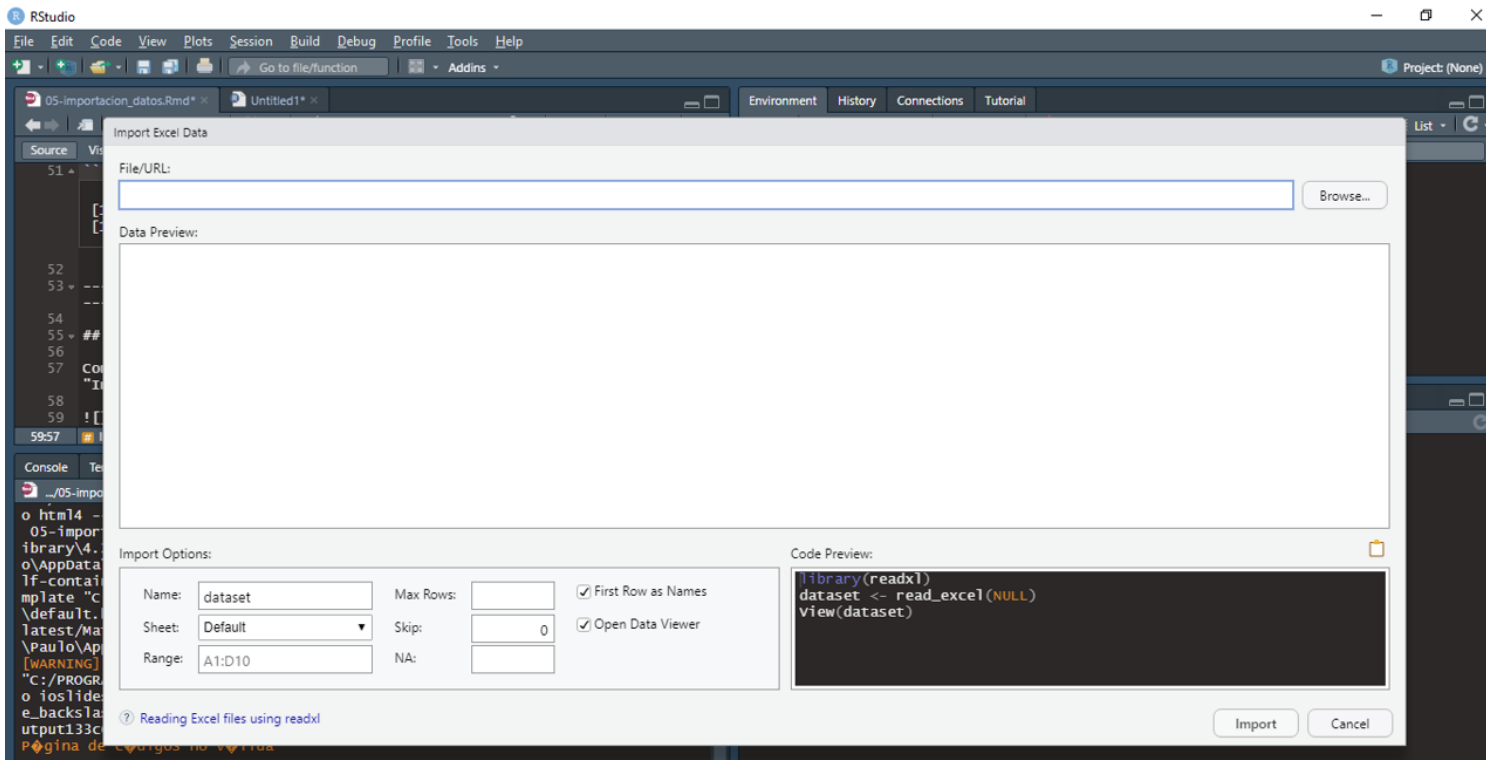
# Importar datos: botones en RStudio

Como primera opción, para importar datos podemos utilizar el botón “Import Dataset” dentro del panel de Entorno y escoger el tipo de archivo.



# Botón de Importación

Luego aparecerá una ventana para que podamos buscar el documento dentro de nuestros archivos. En el lado inferior derecho aparece automáticamente las líneas de código necesarias a ingresar si se quisiera escribir en el editor de códigos.





# Escribiendo el código

Si vamos por la opción de escribir el código en el guión (opción más recomendada) estas serían las líneas de código usuales. Usaremos como ejemplo la base del programa juntos:

```
# Código para importar .csv
```

```
link <- "https://datosabiertos.midis.gob.pe/sites/default/files/03-Dataset-JUNTOS-informacion-usuarios-I-bimestre-2020"
data <- read.csv(link, sep = ";")
```

```
# Código para importar .xlsx
```

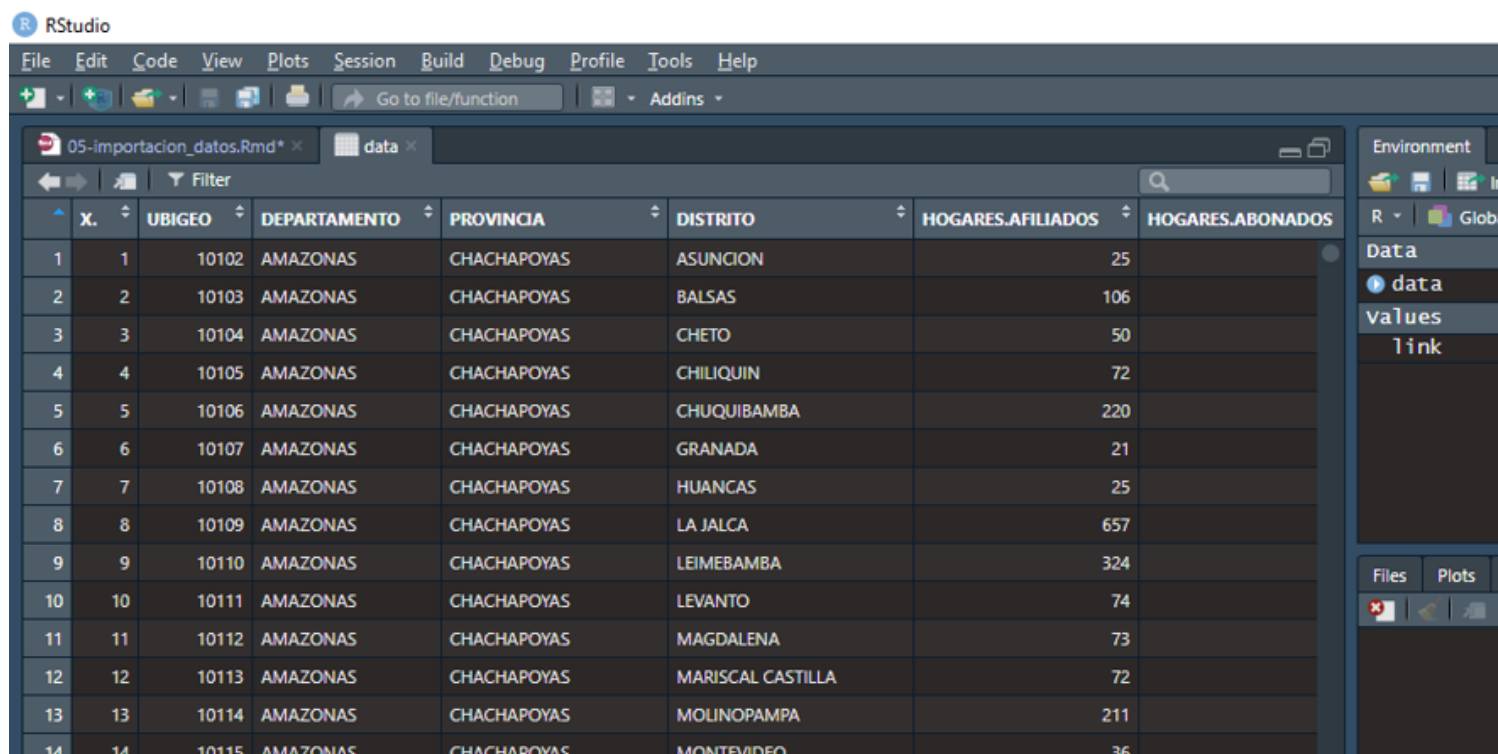
```
library(openxlsx)
data <- read.xlsx("archivo.xlsx")
```

```
# para exportar en formato de excel se recomienda usar el paquete xlsx
```

# Verificar los datos importados

Una vez tengamos la data podemos revisar su contenido con los siguientes comandos:

`View(data)`



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

05-importacion\_datos.Rmd\* data

Filter

	X	UBIGEO	DEPARTAMENTO	PROVINCIA	DISTRITO	HOGARES.AFILIADOS	HOGARES.ABONADOS
1	1	10102	AMAZONAS	CHACHAPOYAS	ASUNCION	25	
2	2	10103	AMAZONAS	CHACHAPOYAS	BALSAS	106	
3	3	10104	AMAZONAS	CHACHAPOYAS	CHETO	50	
4	4	10105	AMAZONAS	CHACHAPOYAS	CHILIQUN	72	
5	5	10106	AMAZONAS	CHACHAPOYAS	CHUQUIBAMBA	220	
6	6	10107	AMAZONAS	CHACHAPOYAS	GRANADA	21	
7	7	10108	AMAZONAS	CHACHAPOYAS	HUANCAS	25	
8	8	10109	AMAZONAS	CHACHAPOYAS	LA JALCA	657	
9	9	10110	AMAZONAS	CHACHAPOYAS	LEIMBAMBA	324	
10	10	10111	AMAZONAS	CHACHAPOYAS	LEVANTO	74	
11	11	10112	AMAZONAS	CHACHAPOYAS	MAGDALENA	73	
12	12	10113	AMAZONAS	CHACHAPOYAS	MARISCAL CASTILLA	72	
13	13	10114	AMAZONAS	CHACHAPOYAS	MOLINOPAMPA	211	
14	14	10115	AMAZONAS	CHACHAPOYAS	MONTEVIDEO	36	

Environment

R Global

Data

data

Values

link

Files Plots

# Verificar los datos importados, continuando...

Si desean explorar los datos también pueden usar los siguientes comandos:

```
str(data)
```

```
# str viene de estructura, sirve para saber la estructura del objeto
```

```
dim(data)
```

```
# para saber las dimensiones en el objeto: número de filas x número de columnas
```

```
length(data)
```

```
# número de columnas
```

```
dplyr::count(data)
```

```
# número de filas en la tabla
```

```
summary(data)
```

```
#resumir los datos
```

# Escribir los datos

R nos permite exportar los objetos que creamos en diferentes formatos (.xlsx, .csv, .sav).

*# Exportar en .csv: se puede realizar con las funciones utils*

```
write.csv(data, "C:/ubicación/nombreadarchivo.csv")
```

*# Exportar en .xlsx: esto solo se hace con ayuda de otros paquetes.*

*# Recomendamos altamente el uso del paquete openxlsx*

```
write.xlsx(data, "C:/ubicación/nombreadarchivo.xlsx")
```