

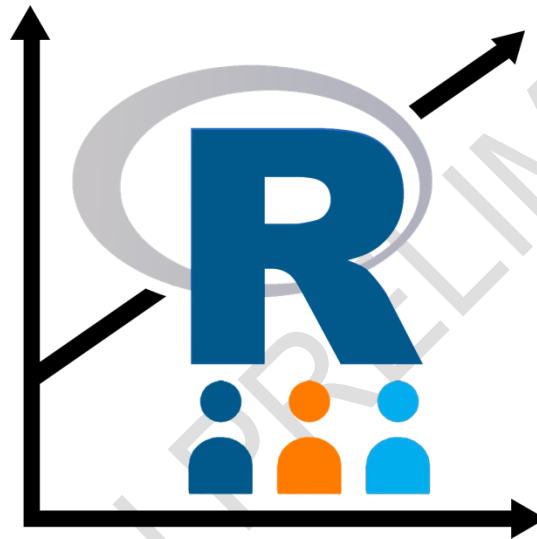


Introducción al uso de "R" en emergencias de salud

Módulo 2:

Tratamiento de datos y características relacionadas con las personas

Clase 6: Importación y exportación de datos



```
modulo_2 <- ("Clase 6")
```

- > Importación de datos con el paquete rio
- > Importación de datos de forma manual
- > Importar datos del REDCap
- > Formatos estándar de R
- > Exportación de datos con el paquete rio

FICHA_TECNICA

Introducción al uso de "R" en emergencias de salud

Módulo 2: Tratamiento de datos y características relacionadas con las personas

Clase 6: Importación y exportación de datos

ORGANIZACIÓN PANAMERICANA DE LA SALUD/ORGANIZACIÓN MUNDIAL DE LA SALUD (OPS/OMS)

Coordinación General

Dra. Socorro Gross Galiano – Representante de la OPS/OMS – Brasília/Brasil

Dra. María Almirón – Advisor, Detection, Verification and Risk Assessment/Health Emergency Information & Risk Assessment (HIM)/ PAHO Health Emergencies Department (PHE) – Washington/United States

Coordinación Ejecutiva

Unidad Técnica de Vigilancia, Preparación y Respuesta a Emergencias y Desastres/ OPS/OMS – Brasília/Brasil

Walter Massa Ramalho

Juan Cortez-Escalante

Laís de Almeida Relvas-Brandt

Mábia Milhomem Bastos

Amanda Coutinho de Souza

Health Emergency Information & Risk Assessment (HIM)/ PAHO Health Emergencies Department (PHE) – Washington/United States

Cristian Hertlein

Wildo Navegantes de Araújo

Equipo Técnico

Unidad Técnica de Vigilancia, Preparación y Respuesta a Emergencias y Desastres/ OPS/OMS

Contenido

Laís de Almeida Relvas-Brandt

Cleonice Andrade Holanda

Revisión

Flávia Reis de Andrade

Laís de Almeida Relvas-Brandt

Equipo Pedagógico

Mônica Diniz Durães – Colaboradora – Consultora Nacional de Capacidades Humanas para la Salud/ OPS/OMS

Creación Digital

Pedro Augusto Jorge de Queiroz

Flávia Reis de Andrade

> 1

> Para seguir la clase

Para seguir esta clase, abra el proyecto del curso haciendo doble clic en el archivo `r_opas.Rproj`, dentro de la carpeta `r_opas` que descargó con el material del curso. Cuando se inicie RStudio, abra el script `r_opas_mod2_clase6.R`, ubicado en la carpeta `r_opas/scripts`. O cree un *script* en blanco para que pueda ingresar los códigos, como aprendió en la clase anterior.

Luego ejecute la función `pacman::p_load()` para instalar y/o cargar los paquetes que se utilizarán en la clase. Son ellos: paquete `pacman` para instalar y cargar todas las demás bibliotecas que se necesitarán en esta clase: `rio` (para la importación y exportación de datos), `tidyverse` (para el manejo y tratamiento de datos) y `REDCapR` (para la importación de datos del REDCap). Los paquetes `rio` y `REDCapR` se presentarán en esta clase.

```
# ORGANIZACIÓN PANAMERICANA DE LA SALUD (OPS) - OFICINA DE BRASIL
# UNIDAD TÉCNICA DE VIGILANCIA, PREPARACIÓN Y RESPUESTA A EMERGENCIAS Y DESASTRES
# CURSO PRINCIPIANTE DE R APLICADO A EMERGENCIAS DE SALUD
# MÓDULO 2 - TRATAMIENTO DE DATOS Y CARACTERÍSTICAS RELACIONADAS CON LAS PERSONAS
# CLASE 6 - IMPORTACIÓN Y EXPORTACIÓN DE ARCHIVOS A R

# 1. Para seguir la clase #####
# Instalar/Cargar paquetes
pacman::p_load(rio,          # Importación y exportación de datos
               tidyverse,    # Manejo y tratamiento de datos
               REDCapR)      # Importación de datos del REDCap
```

> 2

> Importación de datos con el paquete `rio`

Importar una base de datos para analizarse en R ha sido probablemente uno de los pasos más desafiantes para los usuarios principiantes del lenguaje. Esto se debe a

que era necesario utilizar una función de importación específica para cada tipo de extensión del archivo que se importará (como .csv., .dbf, .xlsx, etc.). Si usted ha tenido algún contacto con códigos en R antes de este curso, probablemente ha encontrado las funciones `read.csv()`, `read.csv2()`, `read.xlsx()`, `read_excel()`, `read.dbf()`, `read.dbf()`, entre otras.

Cada una de estas funciones tiene sus argumentos específicos para la importación adecuada del archivo donde se almacenaron los datos. Como resultado, un proceso aparentemente simple de "abrir la base de datos" requiere mucha atención, ya que la importación de cada tipo de extensión requería diferentes pasos para configurar la importación. Muchas veces alguna particularidad pasaba inadvertida y provocaba una pérdida de tiempo en una actividad inicial y mucha frustración para el(la) principiante en el lenguaje R. Después de todo, si no se puede importar la base de datos, no se puede iniciar el análisis.

El paquete `rio` fue creado para simplificar este proceso. Podemos verlo como si fuera una navaja suiza, pues unifica los pasos de importación y exportación de datos de diferentes extensiones en dos funciones simples: `import()` y `export()`, que en español significan importar y exportar. De hecho, el nombre "`rio`" es una referencia a *Input/Output* en R (R I/O), que en español podemos traducir como Entrada/Salida. Creemos que el uso de este paquete favorece la difusión de R en el día a día de los servicios de vigilancia en salud, donde es común manejar bases de datos con diferentes extensiones.

La lista completa de extensiones reconocidas por el paquete `rio` se puede encontrar en el enlace: <https://cran.rproject.org/web/packages/rio/vignettes/rio.html>. Para más detalles, recuerde que usted puede consultar la documentación del paquete en el panel *Help* o escribir `?rio` en el panel *console*.

Ahora comencemos a explorar el funcionamiento de la función `rio::import()` para importar los datos que se utilizarán en este curso y algunos otros ejemplos. Los archivos están disponibles en la carpeta de datos, dentro de la carpeta `r_opas`

proporcionada con el material del curso¹. Los archivos importados se refieren a los siguientes conjuntos de datos:

- Notificaciones de Síndrome Respiratorio Agudo Grave (SRAG) realizadas por un estado brasileño entre 2020 y 2022 (archivo `srag_20_22_.xlsx`). Los datos originales provienen del Sistema de Vigilancia Epidemiológica de la Influenza (SIVEP-Gripe) y se obtuvieron en el sitio web <https://opendatasus.saude.gov.br/> el 17/06/2022.
- Notificaciones de síndrome gripal realizadas por un estado brasileño (`sg_2022.csv`) en 2022. Los datos originales provienen del sistema en eSUS Notifica y se obtuvieron en el sitio web <https://opendatasus.saude.gov.br/> el 17/06/2022.
- Estimaciones de población de los municipios de un estado brasileño en 2020 y 2021. Las estimaciones fueron calculadas por el Ministerio de Salud y se obtuvieron en el tabulador de datos del departamento de informática del SUS (TabNet/DataSUS) a través del <https://datasus.saude.gov.br/populacao-residente> el 17/06/2022.

Importar desde un archivo

Para importar los datos desde un archivo, usted debe elegir un nombre para el objeto que se creará y asignarle la función `rio::import()`, como en el código siguiente. Como argumento de la función, debe proporcionar, entre comillas, la ruta de la carpeta donde está guardado más el nombre de su archivo de datos (incluida la extensión). Ejecute el siguiente código y observe el panel *environment* (Figura 1).

```
# 2. Importación de datos con el paquete rio ####  
# Importar desde un archivo las Notificaciones de síndrome gripal  
# en eSUS, 2022 (Descarga: 17/06/2022)  
sg <- import("datos/sg_2022.csv")
```

¹ Si es necesario, consulte el documento `r_opas_instrucciones_material.pdf`, disponible en la misma carpeta de las clases (`r_opas/clases`).

```
# Importar desde un archivo las Notificaciones de síndrome
# respiratorio agudo grave en el SIVEP-Gripe, 2020-2022
# (Descarga: 17/06/2022)
srag <- import("datos/srag_20_22.xlsx")
```

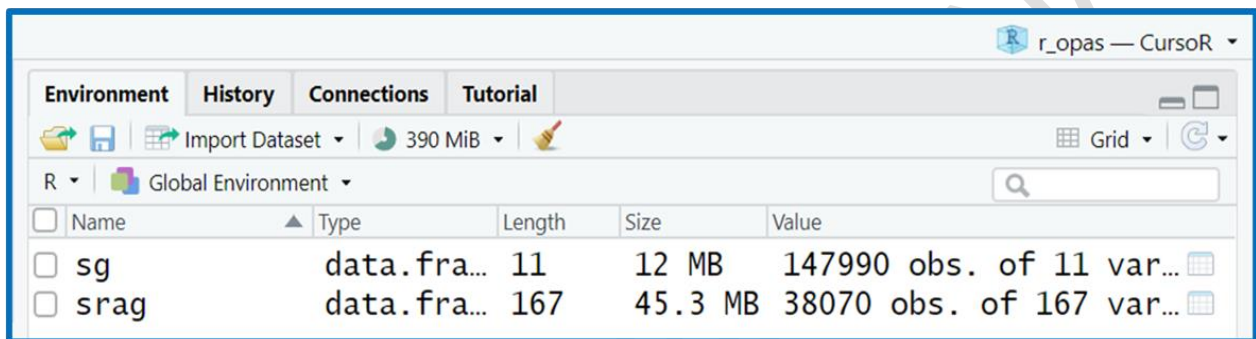


Figura 1: Vista de los objetos `sg` y `srag` dentro del panel *environment*.

Se crearon dos objetos de tipo *data.frame*. El objeto `sg` es una lista de casos SG que incluye 147.990 registros y 11 variables, mientras que el objeto `srag` es una lista de 38.070 casos notificados de SRAG con 167 variables.

Tenga en cuenta que no fue necesario dirigir R a la carpeta donde usted guardó el material del curso. Al usar el entorno de proyecto, R trabajará con rutas relativas a partir del directorio donde se guardó el archivo `.RProj`². En este caso, independientemente de dónde usted guardó la carpeta `r_opas`, R buscará la carpeta de datos dentro de ella y encontrará los archivos de interés. Vea también que la función `rio::import()` se escribió de la misma manera para ambos archivos, pero el primero con la extensión `.xlsx` y el segundo con la extensión `.csv`.

Para archivos del Excel que tienen más de una pestaña, use el argumento `which` = e ingrese el nombre de la pestaña de interés entre comillas para especificar qué

² Contenido discutido en la clase 05, módulo 1.

pestaña debe importarse. Si no se declara el argumento `which`, se importará la primera pestaña del archivo de forma predeterminada.

¡Atención! ¡No olvide usar comas para separar los argumentos de una función! En el siguiente código, observe que los objetos `pop_21` y `pop_20`, referentes a las estimaciones de población municipales para 2020 y 2021, se crearon importando datos del mismo archivo en diferentes pestañas.


```
# 2. Importación de datos con el paquete rio #####  
# Importar desde un archivo las estimaciones de población municipales,  
# 2020-2021 (Fuente: Ministerio de Salud)  
pop_21 <- import("datos/pop_20_21.xls",  
                 which = "pop_ma_2021")  
pop_20 <- import("datos/pop_20_21.xls")
```




Importar desde Google Sheets

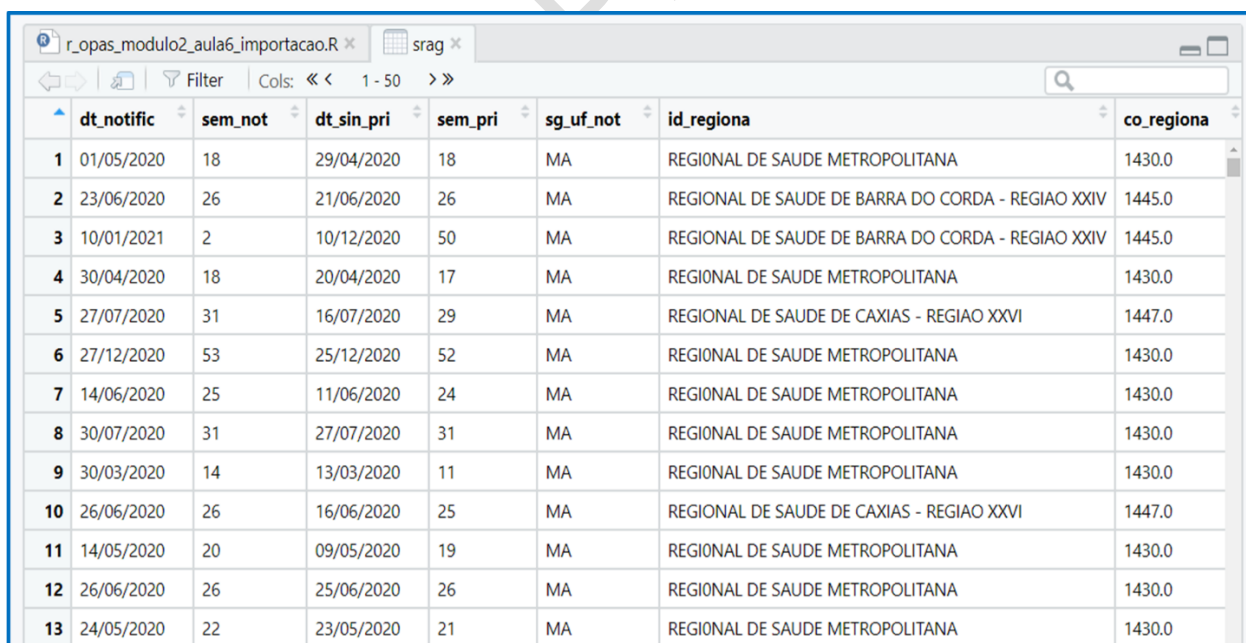
Otra forma comúnmente utilizada para el almacenamiento de datos son las bases provenientes del intercambio de información en la nube, como [Google Sheets](#) (Hojas de cálculo de Google). En esta situación, el dato que se importará se origina de un enlace y no de un directorio o carpeta. Afortunadamente, la función `rio::import()` también funciona para importar datos de *Google Sheets*. Si los datos son públicos, solo hay que proporcionar el enlace de la hoja de cálculo entre comillas como argumento de la función. En el siguiente código, creamos un objeto llamado `google` e importamos un ejemplo ficticio de una lista de casos con tres variables de signos y síntomas.

```
# 2. Importación de datos con el paquete rio #####  
# Importar desde Google Sheets  
google <-  
import("https://docs.google.com/spreadsheets/d/1VZgTTA7cNP9TLo8weH9wp7g5POdUkY  
aN49AhNewA2ys/edit#gid=0")
```

¡Atención! En el caso que no se abra la base de datos es posible que se le solicite que autentique su cuenta de Google. Si esto sucede, siga las instrucciones para otorgar los permisos en RStudio o *pop-ups* en su navegador de Internet. Para editar, crear y eliminar sus hojas de cálculo en *Google Sheets*, entre otros, explore las funcionalidades de los paquetes *googlesheets4* y *googledrive*.

En este momento, usted tiene cinco objetos en su panel *environment*. Puede explorar su visualización en columnas y filas haciendo clic en el icono  dentro del panel *environment*, a la derecha del nombre del objeto (observe la posición del icono en la Figura 1). Automáticamente se abrirá una nueva pestaña en el panel *editor*, al lado del *script*, donde se visualizará la base de datos con un *layout* similar a una hoja de cálculo electrónica, con sus filas y columnas (Figura 2).

En esta vista se puede: abrir los datos en una nueva pestaña con el icono ; establecer filtros en los datos con el icono  **Filter** (solo para visualización, no se filtrará en su objeto); migrar entre las variables con los iconos **Cols: << >> 1 - 50** y con la barra de desplazamiento horizontal, buscar un valor con la lupa  , entre otros. Para cerrar, haga clic en la "x" que está junto al nombre del objeto.




	dt_notific	sem_not	dt_sin_pri	sem_pri	sg_uf_not	id_regiona	co_regiona
1	01/05/2020	18	29/04/2020	18	MA	REGIONAL DE SAUDE METROPOLITANA	1430.0
2	23/06/2020	26	21/06/2020	26	MA	REGIONAL DE SAUDE DE BARRA DO CORDA - REGIAO XXIV	1445.0
3	10/01/2021	2	10/12/2020	50	MA	REGIONAL DE SAUDE DE BARRA DO CORDA - REGIAO XXIV	1445.0
4	30/04/2020	18	20/04/2020	17	MA	REGIONAL DE SAUDE METROPOLITANA	1430.0
5	27/07/2020	31	16/07/2020	29	MA	REGIONAL DE SAUDE DE CAXIAS - REGIAO XXVI	1447.0
6	27/12/2020	53	25/12/2020	52	MA	REGIONAL DE SAUDE METROPOLITANA	1430.0
7	14/06/2020	25	11/06/2020	24	MA	REGIONAL DE SAUDE METROPOLITANA	1430.0
8	30/07/2020	31	27/07/2020	31	MA	REGIONAL DE SAUDE METROPOLITANA	1430.0
9	30/03/2020	14	13/03/2020	11	MA	REGIONAL DE SAUDE METROPOLITANA	1430.0
10	26/06/2020	26	16/06/2020	25	MA	REGIONAL DE SAUDE DE CAXIAS - REGIAO XXVI	1447.0
11	14/05/2020	20	09/05/2020	19	MA	REGIONAL DE SAUDE METROPOLITANA	1430.0
12	26/06/2020	26	25/06/2020	26	MA	REGIONAL DE SAUDE METROPOLITANA	1430.0
13	24/05/2020	22	23/05/2020	21	MA	REGIONAL DE SAUDE METROPOLITANA	1430.0

Figura 2: Vista de los datos del objeto *srag* en una nueva pestaña en el panel editor.

Observe que al abrir la vista del *data.frame* `srag`, como vimos arriba, el comando se mostrará automáticamente `View(srag)` en la *console*. También puede escribir esta función en el *script* o en la consola para visualizar los datos como en la Figura 2, recordando informar el nombre del objeto como argumento. La visualización de datos de esta manera suele ser una actividad muy realizada por los usuarios principiantes en el lenguaje R, pero tenga cuidado con bases de datos muy grandes, ya que pueden requerir mucho procesamiento de su computadora.

> 3

> Importación de datos de forma manual

Además, hay la opción de importar los datos de forma manual, haciendo clic en el icono  `Import Dataset` del panel *environment*. Como ejemplo, volvamos a importar el archivo `sg_2022_2022.csv` (notificaciones de síndrome gripal) que se encuentra en la carpeta `r_opas/datos`. Al ser una extensión `.csv`, elijamos la opción *From Text (base)*, como en la Figura 3.

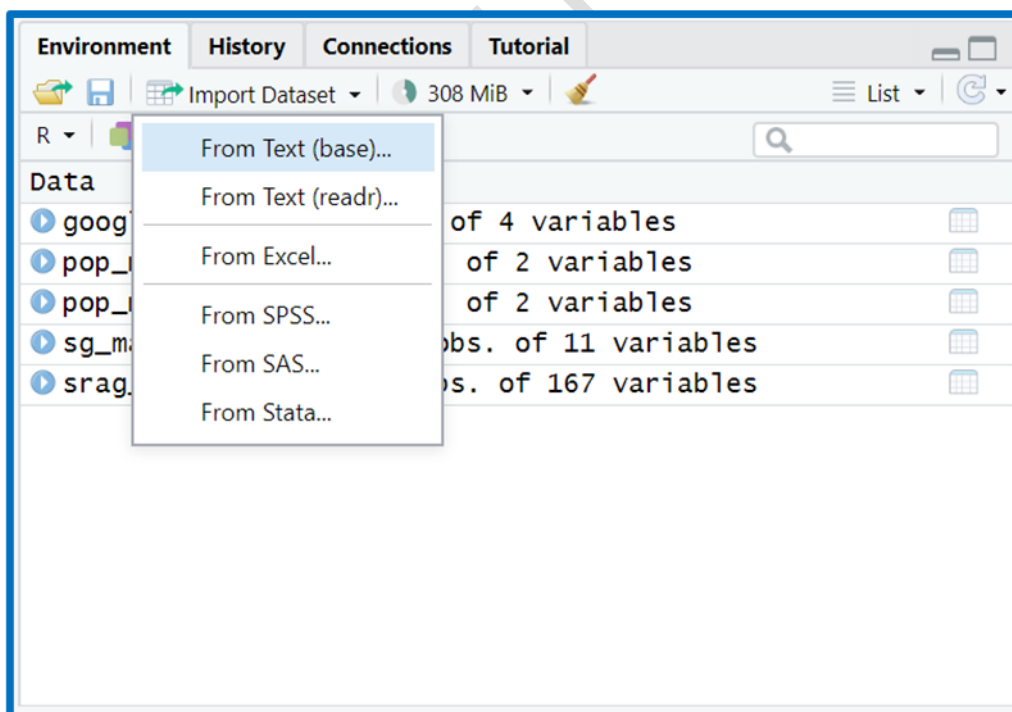


Figura 3: Vista del icono *Import Dataset* en el panel *environment*.

Se abrirá automáticamente la ventana *Select File to Import*, que en español significa: seleccione el archivo para importar. Usted debe identificar la carpeta r_opas/datos en su computadora, seleccionar el archivo y hacer clic en *Open* (Figura 4).

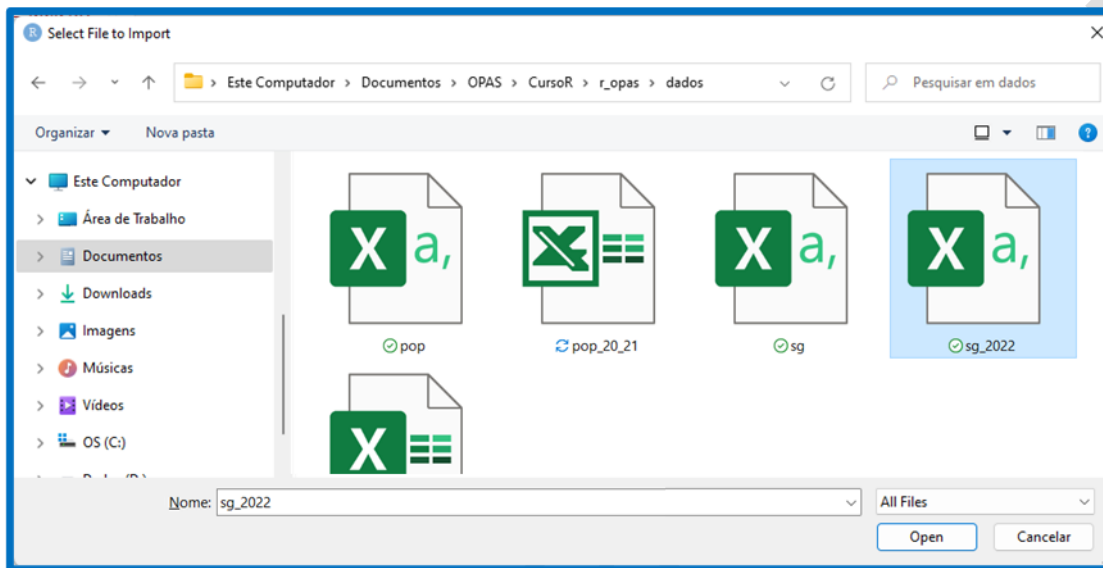


Figura 4: Vista de la pantalla *Select File to Import* para seleccionar el archivo.

Se abrirá una nueva ventana titulada *Import Dataset* (Importar conjunto de datos). En ella usted puede definir el nombre del *data.frame* que se creará con los datos importados, la codificación (*Encoding*), si la primera columna contiene el nombre de las variables (*Heading*), el separador de los datos en el archivo .csv (*Separator*), separador decimal, entre otros. A la derecha, también se puede obtener una vista de cómo se almacenan los datos brutos y cómo quedarán en el *data.frame* (Figura 5). Elija un nombre para el objeto que se creará o mantenga el nombre que se generó automáticamente (el mismo nombre del archivo) y haga clic en *Import* para importar.

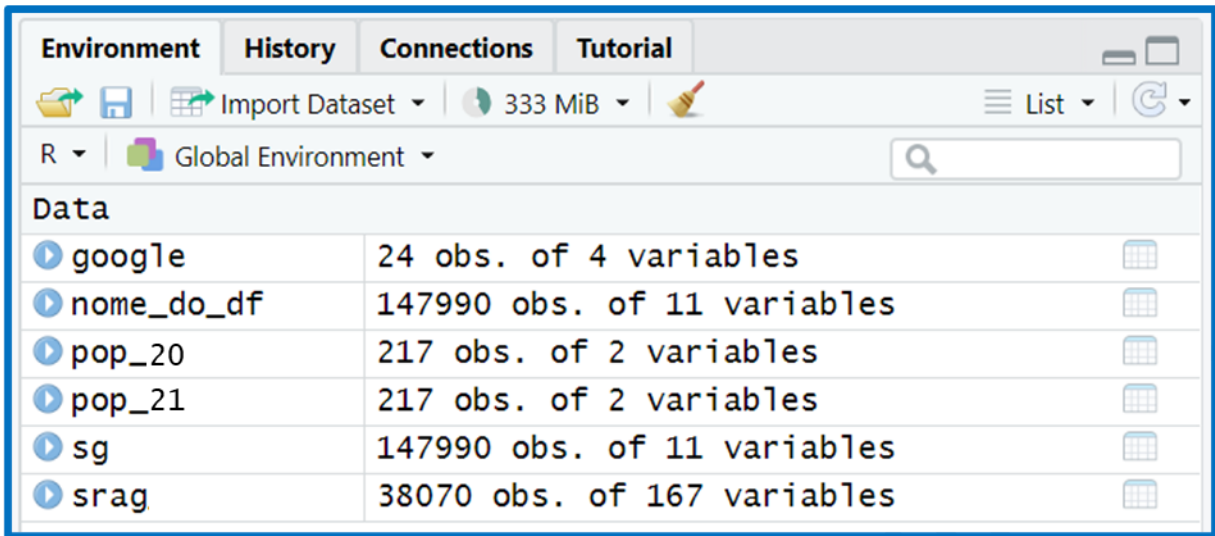


Figura 6: Observación de los objetos *data.frames* en el panel *environment*.

Si bien puede parecer más fácil importar datos de esta manera, no recomendamos esta estrategia, ya que es probable que la función `rio::import()` tiene suficiente configuración interna para evitar la mayoría de los errores que podríamos cometer en estos pasos manuales.

> 4 > Importar datos de REDCap

REDCap es un acrónimo de *Research Electronic Data Capture*. Es una aplicación web para construir y gestionar encuestas y bases de datos en línea, desarrollada por la Universidad de Vanderbilt, en Tennessee, Estados Unidos. En los formularios del REDCap se producen notificaciones nacionales de Monkeypox (Viruela del mono), de Síndrome Inflamatorio Multisistémico Pediátrico (SIMP) y de Síndrome Inflamatorio Multisistémico del Adulto (SIMA) asociadas temporalmente al COVID-19, entre otras condiciones de emergencia de salud pública en Brasil.

Desafortunadamente, el paquete `rio` no tiene interfaz con REDCap. Además, para importar los datos de REDCap directamente a R, usted debe comprobar que tiene acceso a esta información a través de un *token*, que se puede obtener en la sección API, cuando esté conectado(a). Cada *token* es específico para un(a) solo(a) usuario(a) en un

solo proyecto REDCap. **¡Atención!** Es una contraseña personal e intransferible. Por ello, tenga mucho cuidado de nunca compartir su *script* con *token*, ¡especialmente cuando tenga acceso a datos de identificación de los sujetos o cualquier otro dato sensible!

Los datos del REDCap se pueden importar con la función `redcap_read()` del paquete REDCapR (ya cargado al inicio). Informe al argumento `redcap_uri=` el enlace de la API del proyecto REDCap al que accederá entre comillas; al argumento `token =` su clave específica, también entre comillas; y al argumento `raw_or_label=` la opción que quiera, siendo `"label"` para importar las etiquetas de los datos o `"raw"` para los códigos. Termine con `$data` para especificar que el interés es descargar los datos. Sin el `$data` el comando solo devolverá un objeto de lista con múltiples componentes.

```
# 4. Importar datos del REDCap ####  
# Importar datos de RedCAP  
simp <- redcap_read(redcap_uri="https://redcap.saude.gov.br/api/",  
                    token="",  
                    raw_or_label="label")$data
```

El código anterior es un ejemplo y no importará ningún dato, ya que no tiene *token*. Pero si usted tiene el suyo, puede insertarlo dentro de las comillas y ejecutar el comando. También puede incluir el argumento `fields =` para especificar las variables que desea importar. Por ejemplo, si usa `fields = c("sexo", "ufnot")`, solo se importarán las variables `sexo` y `ufnot`. Para conocer más sobre el paquete REDCapR, acceda a: <https://cran.r-project.org/web/packages/REDCapR/REDCapR.pdf>.

¡Un consejo! También es posible descargar los datos a un archivo para su computadora (por ejemplo, en `.csv` o `.R`, específico de R) e importar con la función `rio::import()`. Aunque requiera este paso manual, a veces puede ser más fácil.

Existen dos formatos estándar de lenguaje R que quizás le interese utilizar en sus análisis. El primero es el formato `.rds`, que almacena un solo objeto de R, como un *data.frame*. Es útil para almacenar datos limpios, ya que mantiene las clases de columnas R y todas las transformaciones que usted realice (por ejemplo, cambiar el nombre de una columna, convertir una variable en fecha o crear la variable grupo de edad).

A partir de la próxima clase trabajaremos con esta extensión para que no sea necesario ejecutar todo el *script* de tratamiento de los datos cada vez que se abra RStudio. Por ejemplo, si usted tiene que interrumpir el estudio de una clase en su tercera sección y regresa al día siguiente, podrá guardar sus datos ya tratados con la extensión `.rds` y volver al día siguiente sin tener que ejecutar los comandos que ya estudió antes de parar.

Es muy probable que usted abandone el formato `.rds` a medida que adquiera experiencia con R, ya que seguramente le resultará más fácil volver a ejecutar sus comandos que guardar archivos paralelos. De hecho, el uso de archivos con esta extensión no es una regla en la rutina de análisis de datos. Pero hemos visto que guardar los archivos en `.rds` es una estrategia que funciona bien con los usuarios principiantes de R.

Al almacenar los pasos ya realizados, evitamos confusiones, errores de código y, en consecuencia, evitamos frustraciones en sus primeras inmersiones con el lenguaje R. Afortunadamente, la función `rio::import()` también puede cargar la extensión `.rds` y no tendremos grandes problemas para traer nuestros datos ya tratados a RStudio, como veremos en la siguiente clase.

El segundo formato de exportación de archivos estándar de R es `.RData`, que puede almacenar varios objetos o incluso un entorno de trabajo de R completo. Con esta extensión, todos sus objetos presentes en el panel *environment* se guardarán con todos los cambios realizados previamente en cada uno. Si usted configuró R como sugerimos en la clase 3, cada vez que cierre RStudio, se le preguntará si quiere guardar el entorno de trabajo (*workspace*) en un archivo `.RData`. También es posible guardar el archivo `.RData` del *workspace* haciendo clic en el icono del *disket*, dentro del panel *environment*.

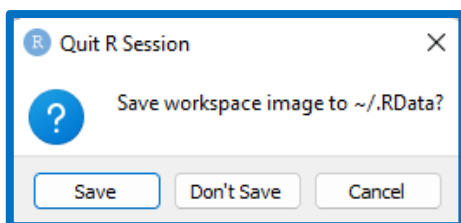


Figura 7: Vista de la pantalla *Quit R Session* con la opción de guardar el entorno de trabajo en un archivo .RData.

Sin embargo, la práctica de guardar archivos .Rdata puede afectar el procesamiento de los análisis cuando sus datos son muy pesados (archivos grandes). Por ello, su uso debe ser cauteloso y no rutinario. Para cargar archivos .Rdata al reiniciar una sesión en R, use la función `load()` del paquete `base` informando la ruta y el nombre del archivo entre comillas. O haga clic en el icono 📁 para encontrar el archivo .Rdata en su computadora.

> 6

> Exportación de datos con el paquete `rio`

El paquete `rio` facilitó mucho la exportación de datos al introducir la función `rio::export()`, que es versátil para exportar diferentes tipos de extensiones de archivo, así como la función `rio::import()`. La función `export()` tiene dos argumentos obligatorios. El primero es el objeto que se exportará y el segundo es la ruta completa de la carpeta donde quiere guardar este *output* en su computadora, la cual debe informarse entre comillas, incluyendo el nombre del archivo y su extensión. Con el siguiente código, vamos a exportar los `data.frames` `sg` y `srag` para los archivos `sg.rds` y `srag.rds`, respectivamente. Aunque no se ha realizado ninguna transformación, ejecutemos los códigos para comprender cómo funciona la función `rio::export()`.

```
# 6. Exportación de datos con el paquete rio ####
```

```
# Exportar en .rds las Notificaciones de síndrome gripal en esUS, 2022
```

```
# (Descarga: 17/06/2022)
export(sg, "datos/sg.rds")

# Exportar en .rds las Notificaciones de síndrome respiratorio agudo
# grave en SIVEP-Gripe, 2020-2022 (Descarga: 17/06/2022)
export(srag, "datos/srag.rds")
```

Nótese que no fue necesario informar a R la ruta completa de la carpeta `r_opas/datos` en su computadora. Como estamos trabajando en entorno de proyecto, solo necesitamos informar la ruta deseada desde la carpeta raíz del proyecto, como vimos en la sección de importación. Para finalizar esta clase, abra la carpeta `r_opas/datos` en su computadora y verifique los archivos exportados.

> 7

> Referencias

Batra, Neale, et al. The Epidemiologist R Handbook. 2021. DOI:10.5281/zenodo.475264610.528 Disponible en: <https://epirhandbook.com/en/>.

Silva, Henrique Alvarenga da. Manual Básico da Linguagem R: Introdução à análise de dados com a linguagem R e o RStudio para área da saúde. [recurso electrónico] / Henrique Alvarenga da Silva - 1ª edición – São João del Rei: Editora do Autor, 2018. ISBN: 978-65-900132-0-0 (e-book).

Wickham, H., & Grolemund, G. R for data science: import, tidy, transform, visualize, and model data. O'Reilly Media, Inc. 2016. ISBN: 9781491910399. Disponible en: <https://r4ds.had.co.nz/>