Unidad 3. Procesamiento de datos

Del Curso introductorio al lenguaje de programación R orientado al análisis cuantitativo en Ciencias Sociales por Sarahí Aguilar González

Objetivo: Que el estudiante sea capaz de importar y exportar conjuntos de datos de y en diferentes formatos, así como ejecutar transformaciones básicas sobre estas utilizando dataframes y el paquete data.table.

Agenda

- Paquetes de R
- Data Frames
- data.table

Agenda

Paquetes de R

Data Frames

data.table

corre en

Lenguaje de programación y entorno enfocado al cómputo estadístico y graficación.

R Base

La colección de funciones R que se carga cada vez que inicia R. Estas funciones proporcionan los conceptos básicos del lenguaje y no es necesario cargar un paquete antes de poder usarlas.

Entorno de desarrollo integrado (IDE) para R.

No eres la única persona que escribe sus propias funciones con R.

Muchos profesores, investigadores y programadores usan R para construir y actualizar sus propias funciones que se adecuan a sus necesidades.

Algunos, empaquetan sus funciones y las comparten con el mundo.

R promueve esto y lo permite hacer de forma fácil.

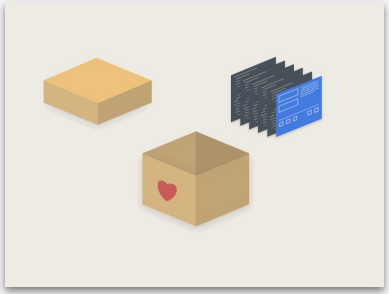
Un paquete R agrupa funciones, conjuntos de datos y archivos de ayuda.

Cualquiera puede usar estas funciones dentro de su propio código R una vez que cargue el paquete en el que residen.

Por lo general, el contenido de un paquete R **está relacionado con un solo tipo de tarea**, que el paquete ayuda a resolver.

Los paquetes necesitan ser **descargados** (una sola vez) y **cargados** al entorno de R (cada vez que se usarán).

Los paquetes son actualizados por sus creadores, y para correr correctamente en tu R, **deben ser compatibles con su versión**.



Paquetes de R

corre en

Lenguaje de programación y entorno enfocado al cómputo estadístico y graficación.

Entorno de desarrollo integrado (IDE) para R.

R Base

La colección de funciones R que se carga cada vez que inicia R. Estas funciones proporcionan los conceptos básicos del lenguaje y no es necesario cargar un paquete antes de poder usarlas.

están escritos en

Paquetes de R

Una colección de funciones escritas en R descargables. Estas funciones proporcionan conceptos más avanzada del lenguaje de forma simplificada, pero es necesario descargarlas y cargarlas antes de poder usarlas.

¿Qué paquetes de R conocen o han utilizado?

Instalando paquetes

(Descargar a tu equipo de cómputo)

La forma más sencilla de instalar un paquete de R es con la función `install.packages("packagename")`

Esto buscará el paquete especificado en la colección de paquetes alojados en el sitio CRAN. Cuando R encuentre el paquete, lo descargara en una carpeta de bibliotecas en su equipo de cómputo. R puede acceder al paquete aquí en futuras sesiones de R sin reinstalarlo.

Cargando paquetes

(Cargar sus funciones en tu sesión)

Para cargar un paquete de R, se utiliza la función R es con la función `library(packagename)`.

Esto pondrá a tu disposición todas las funciones, conjuntos de datos y archivos de ayuda del paquete hasta que cierre su sesión actual de R. La próxima vez que inicies una sesión de R, tendrás que volver a cargar el paquete si deseas usarlo, pero no tendrás que volver a instalarlo.

Algunos puntos que recordar:

- Mientras más dependamos de paquetes de R, más y nuevas reglas tendremos que aprender, y nuestros programas serán menos homogéneos y difíciles de reutilizar.
- Utilicemos únicamente Base R en medida de lo posible.
- Es importante familiarizarse con la documentación de los paquetes que utilizamos y leerla.
- ¡No olvidemos citar los paquetes que utilicemos en nuestra investigación!

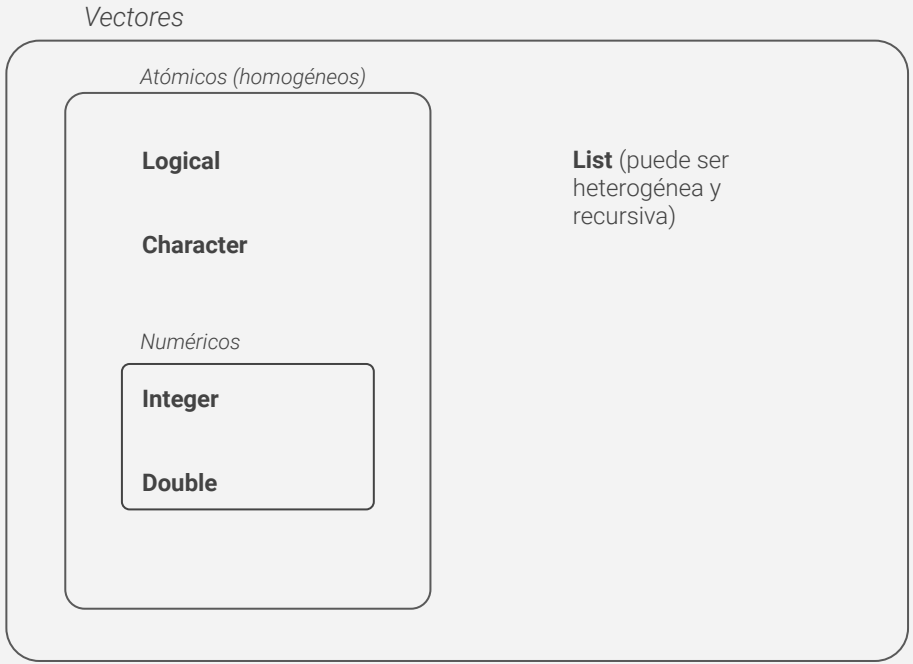
Agenda

Paquetes de R

Data Frames

data.table

Todos los **tipos de objetos en R** están construidos a partir de **vectores**.



NULL *(Ausencia de vector)*

Las 2 **propiedades** de un vector son:

Tipo

Tamaño

Además, pueden también contener **atributos adicionales** que los convierten en **vectores aumentados**.

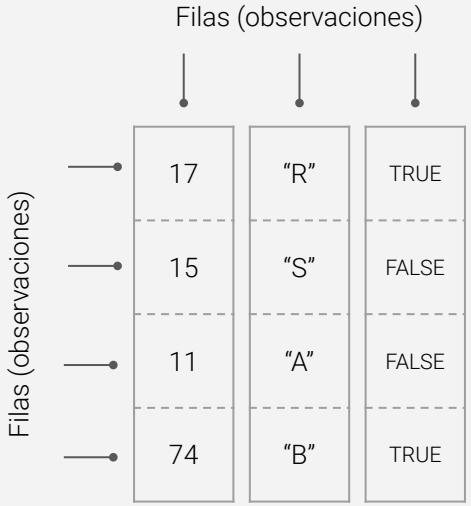
Los **factores** están construidos *encima* de los vectores numéricos de tipo entero.
Las **fechas** están construidas *encima* de los vectores numéricos.
Los **data frames** están construidos *encima* de los vectores de tipo lista.

Los data frames son la versión bidimensional de una lista y son la estructura de almacenamiento más útil para el análisis de datos.

Los data frames agrupan los vectores en una tabla bidimensional. Cada vector se convierte en una columna de la tabla.

Como resultado, cada valor de una **fila** (observación) de un data frame puede contener un **tipo de datos diferente**; pero cada valor de una **columna** (variable) debe tener el **mismo tipo de datos**.

En R, podemos crear data tables desde 0, a partir de archivos externos o de otros data frames.



La extracción de datos, la creación de nuevas columnas, la unión con otros conjuntos de datos y **cualquier operación sobre un data frame están basadas en índices y nombres de columnas**.

	"viv"	"cat"	"flag"
	1	2	3
1	17	"R"	TRUE
2	15	"S"	FALSE
3	11	"A"	FALSE
4	74	"B"	TRUE

Data Frames

La forma general de sintaxis de un data frame es

`df[i, j]`

Textualmente, se leería de la siguiente forma:

1. **Toma df**
2. **Toma el subconjunto de filas i y el subconjunto de columnas j**

¿Cómo seleccionamos la **3ra observación** de la **2da variable**? `df[3, 2]` ó `df[3, "cat"]`

	"viv"	"cat"	"flag"
	1	2	3
1	17	"R"	TRUE
2	15	"S"	FALSE
3	11	"A"	FALSE
4	74	"B"	TRUE

Data Frames

La forma general de sintaxis de un data frame es

`df[i, j]`

Textualmente, se leería de la siguiente forma:

1. **Toma df**
2. **Toma el subconjunto de filas i y el subconjunto de columnas j**

¿Cómo seleccionamos la **2da y 3ra observación** de la **1er variable**? `df[2:3, 1]` ó `df[2:3, "viv"]`

	"viv"	"cat"	"flag"
	1	2	3
1	17	"R"	TRUE
2	15	"S"	FALSE
3	11	"A"	FALSE
4	74	"B"	TRUE

Unión de conjuntos de datos

En ocasiones, vamos a necesitar combinar múltiples conjuntos de datos y agregar nuevas observaciones y/o variables.

Conjunto de datos A

	"viv"	"cat"	"flag"
	1	2	3
1	17	"R"	TRUE
2	15	"S"	FALSE
3	11	"A"	FALSE
4	74	"B"	TRUE

Conjunto de datos B

	"viv"	"cat"	"flag"
	1	2	3
1	19	"A"	FALSE
2	23	"R"	TRUE

¡Nuevo conjunto de datos!

	"viv"	"cat"	"flag"
	1	2	3
1	17	"R"	TRUE
2	15	"S"	FALSE
3	11	"A"	FALSE
4	74	"B"	TRUE
5	19	"A"	FALSE
6	23	"R"	TRUE

Unión de conjuntos de datos

En ocasiones, vamos a necesitar combinar múltiples conjuntos de datos y agregar nuevas observaciones y/o variables.

Conjunto de datos A

	"viv"	"cat"	"flag"
	1	2	3
1	17	"R"	TRUE
2	15	"S"	FALSE
3	11	"A"	FALSE
4	74	"B"	TRUE

Conjunto de datos B

	"viv"	"cat2"
	1	2
1	17	"O"
2	15	"M"
3	11	"N"
4	74	"O"

¡Nuevo conjunto de datos!

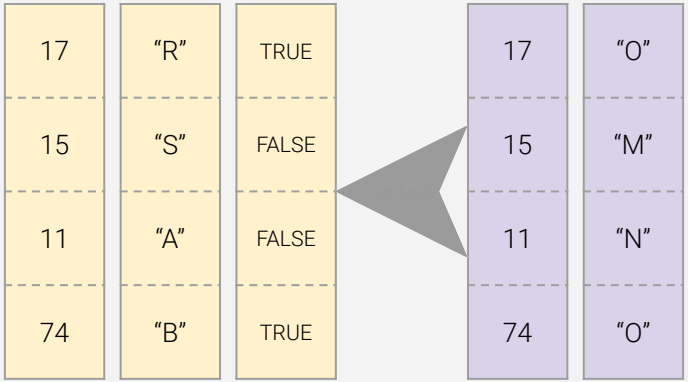
	"viv"	"cat"	"flag"	"cat2"
	1	2	3	4
1	17	"R"	TRUE	"O"
2	15	"S"	FALSE	"M"
3	11	"A"	FALSE	"N"
4	74	"B"	TRUE	"O"

Unión de conjuntos de datos

En R, podemos unir dos conjuntos de datos de múltiples formas…

`rbind` y `cbind`

- Simplemente pegamos "de forma forzada" nuevas filas (`rbind`) y nuevas columnas (`cbind`).



¿Qué pasaría si nuestros conjuntos de datos no están ordenados de la misma forma?

En un `cbind`, ¿qué pasaría si no tuviésemos el mismo número de observaciones en ambos conjuntos de datos?

En un `rbind`, ¿qué pasaría si no tuviésemos el mismo número de columnas en ambos conjuntos de datos?

Unión de conjuntos de datos

En R, podemos unir dos conjuntos de datos de múltiples formas…

`merge`

- Se requiere una o más variables compartidas (🔑 llaves).
- Se debe definir si se requieren conservar todos los elementos del primer conjunto de datos, todos los elementos del segundo conjunto de datos o todos los elementos de ambos conjuntos de datos.

Conjunto de datos `x`

🔑 viv	cat
17	"R"
15	"S"

Conjunto de datos `y`

🔑 viv	cat2
15	"M"
11	"N"

Todos los elementos de `x`

🔑 viv	cat	cat2
17	"R"	NA
15	"S"	"M"

Todos los elementos de `y`

🔑 viv	cat	cat2
15	"S"	"M"
11	NA	"N"

Todos los elementos de `x` y `y`

🔑 viv	cat	cat2
11	NA	"N"
15	"S"	"M"
17	"R"	NA

Reorientación (reshape) de conjuntos de datos

En ocasiones, vamos a necesitar reorientar observaciones y/o variables en un conjunto de datos.
En R, existen dos funciones comunes para reorientar un conjunto de datos…

`melt`

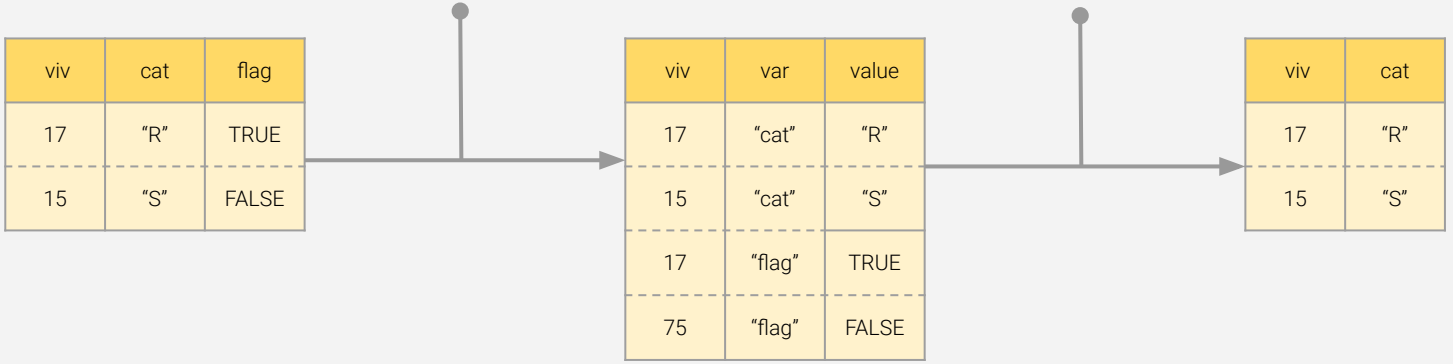
Es necesario definir:

1. uno o varios índices (`"viv"`)
2. la o las variables que queremos reorientar (`"cat"` y `"flag"`)

`cast`

Es necesario definir:

1. qué variable o variables contiene el nombre de la o las variables que queremos reorientar (`"var"`)
2. qué variable o variables contienen los valores de la o las variables que queremos reorientar (`"value"`)



viv	cat	flag
17	"R"	TRUE
15	"S"	FALSE

viv	var	value
17	"cat"	"R"
15	"cat"	"S"
17	"flag"	TRUE
75	"flag"	FALSE

viv	cat
17	"R"
15	"S"

Agenda

- Paquetes de R
- Data Frames
- data.table

Agenda

Paquetes de R

Data Frames

¿Por qué no hemos utilizado el Tidyverse? Espera… ¿qué era el Tidyverse?

data.table

%>%

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.
tidyverse.org

Mientras más dependamos de paquetes de R, más y nuevas reglas tendremos que aprender, y nuestros programas serán menos homogéneos y difíciles de reutilizar.

Lo bonito es sencillo y lo sencillo es bonito.

Agenda

- Paquetes de R
- Data Frames
- data.table

La manipulación de datos utilizando data frames suele ser *confusa* pues **hay muchas maneras** de extraer datos, crear nuevas variables y unir conjuntos de datos.

data.table

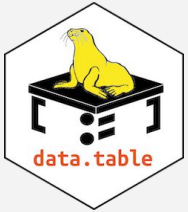
data.table es un paquete R que proporciona una versión *mejorada* de data frames.

data.table permite...

- utilizar una **sintaxis concisa**, **coherente** e **independiente** del conjunto de operaciones,
- realizar análisis de datos de forma fluida **sin la carga cognitiva de tener que asignar cada operación a una función particular** de un conjunto potencialmente enorme de funciones disponibles antes de realizar el análisis de datos,
- y la **optimización automática** de las operaciones que produce un código rápido y eficiente en memoria.

En resumen, si estamos interesados en reducir enormemente el tiempo de programación y computación, este paquete es el indicado.

¡data.table es la segunda librería de manejo de conjuntos de datos más rápida!



¿Por qué un **data.table** proporciona una versión *mejorada* de un data frame?

A diferencia de un data frame, con un data.table se puede hacer mucho más que simplemente crear subconjuntos de filas y seleccionar columnas dentro del marco de `[...]`.

La forma general de sintaxis de data.table es

```
DT[i, j, by = k]
```

Textualmente, se leería de la siguiente forma:

1. **Toma DT**
2. **Toma el subconjunto de filas i u ordena las filas usando las columnas i**
3. **Toma el subconjunto de columnas j o calcula las nuevas columnas j**
4. **Agrupando por las columnas k**