Curso Primeros Pasos en R

Clase 5: Introducción a Tidyverse

Profesora: Ana María Alvarado

Pontificia Universidad Católica de Chile

Noviembre 2021

Clase 5: Introducción a `tidyverse`

- ¿Qué es `tidyverse`?
- Sintaxis en `tidyverse`
- Transformación de datos con `dplyr`
- Taller práctico

¿Qué es `tidyverse`?

¿Qué es `tidyverse`?

- Es un universo de paquetes de R
- Diseñados para las tareas de programación, importación, ordenación, manipulación y visualización de datos.
- Todos utilizan la misma **filosofía** de **diseño**, **gramática** y **estructuras de datos**.
- Resuelve problemas complicados combinando diferentes piezas consistentes unas con otras.

Ciclo de un proyecto de ciencia de datos

Dentro del ciclo de un proyecto de ciencia de datos, **tidyverse** cubre las etapas de leer, ordenar, transformar y visualizar datos.

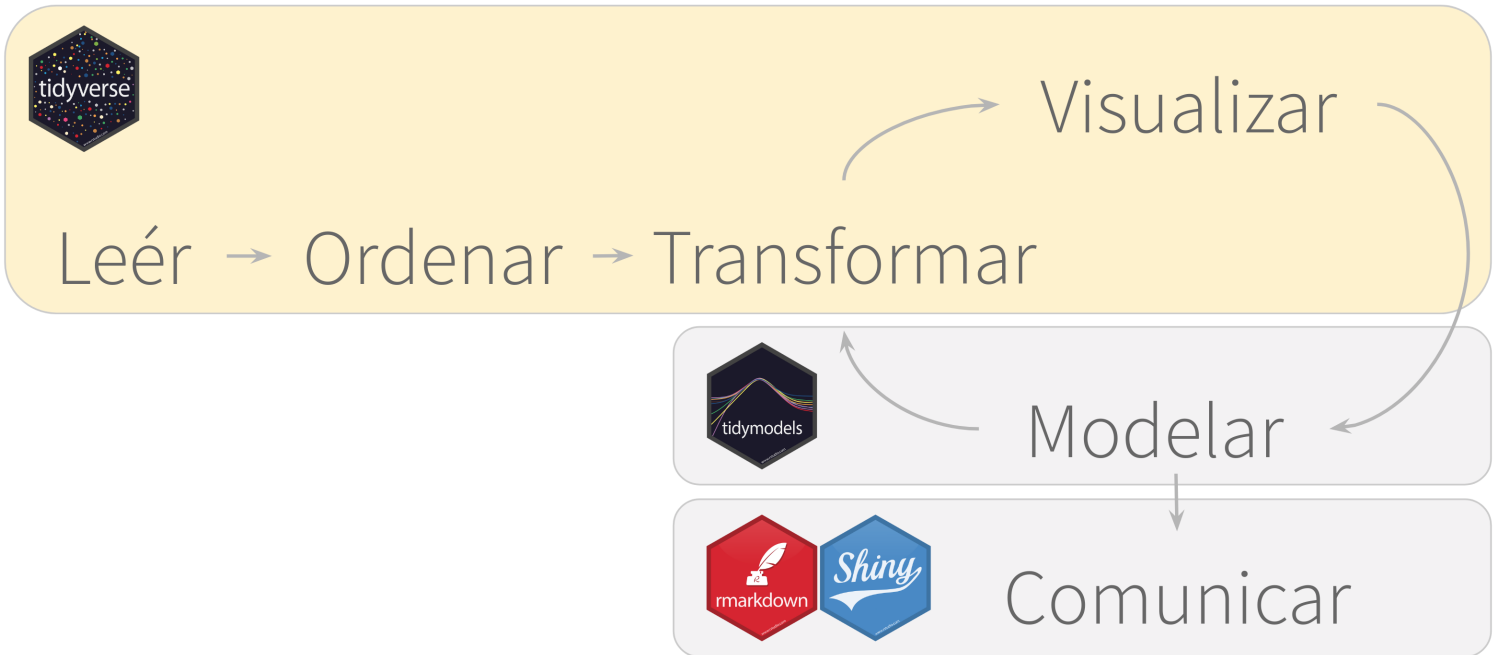


Imagen: https://github.com/rstudio-education

Paquetes de `tidyverse`

El corazón de **tidyverse** son los paquetes:

dplyr - Transformar datos

ggplot2 - Visualizar datos

readr - Leer archivos csv, txt..

forecast - Factorización

tibble - Marco de datos

stringr - Manejo de texto

tidyr - Ordernar datos

purrr - Programación funcional

Paquetes de `tidyverse`

Algunos paquetes adyacentes a tidyverse, no centrales, son:

Lectura de datos

- readxl
- googlesheets4
- googledrive
- jsonlite
- xml2
- rvest
- haven
- httr

Ordenamiento y transformación

- janitor
- lubridate
- glue
- skimr
- tidytext
- hms
- feather
- blob
- modelr

Paquetes de `tidyverse`

Algunos paquetes adyacentes a tidyverse, no centrales, son:

Visualización de datos

- kableExtra
- ggrepel
- cowplot
- patchwork
- gganimate

Programación

- reprex
- magrittr

Sintaxis en `tidyverse`

Sintaxis en `tidyverse`

La sintaxis del `tidyverse` se ha popularizado en los últimos años porque permite leer el código y programar de forma más parecida a como leemos (de izquierda a derecha y de arriba hacia abajo). Esto es posible gracias al operador "pipe" `%>%`

funcion(objeto, ...)

objeto %>% funcion(...)

Ejemplo R base:

```
mean(paises[paises$anio == 2007,
            ]$poblacion)
```

Ejemplo `tidyverse`:

```
paises %>%
  filter(anio == 2007) %>%
  summarize(mean(poblacion))
```

...con tidyverse es más fácil interpretar.

Operador Pipe %>%

El operador pipe %>% permite encadenar funciones sin tener que ir creando variables para uso temporal o sin tener que anidar funciones.

Puedes usar los siguientes atajos con el teclado...

CTRL + SHIFT + M (Windows)

CMD + SHIFT + M (Mac)

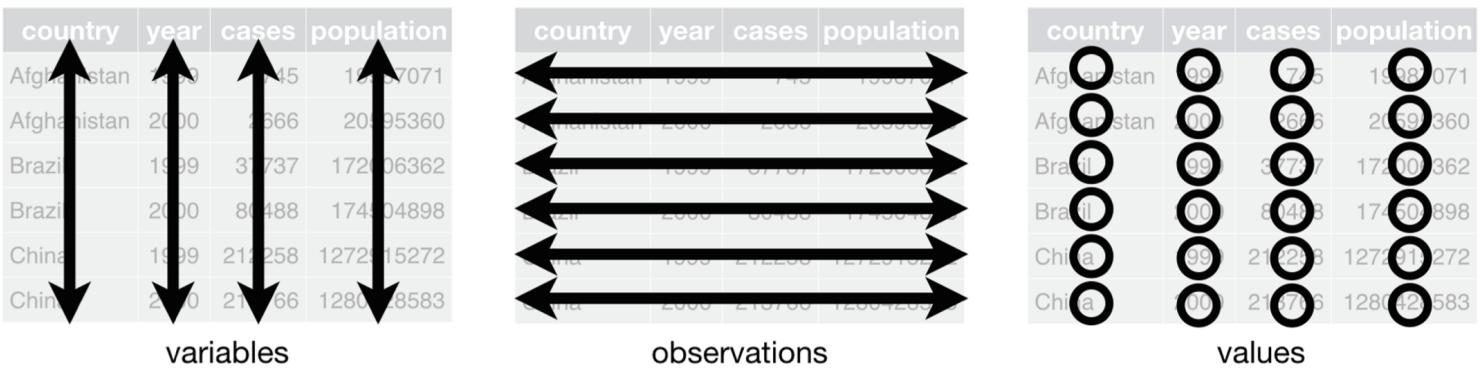
Los principios de `tidyverse`

1. Cada función es un paso

2. Las funciones se combinan con %>%

3. Use datos *ordenados*

- Cada columna es una variable
- Cada línea es una observación
- Cada celda es un valor

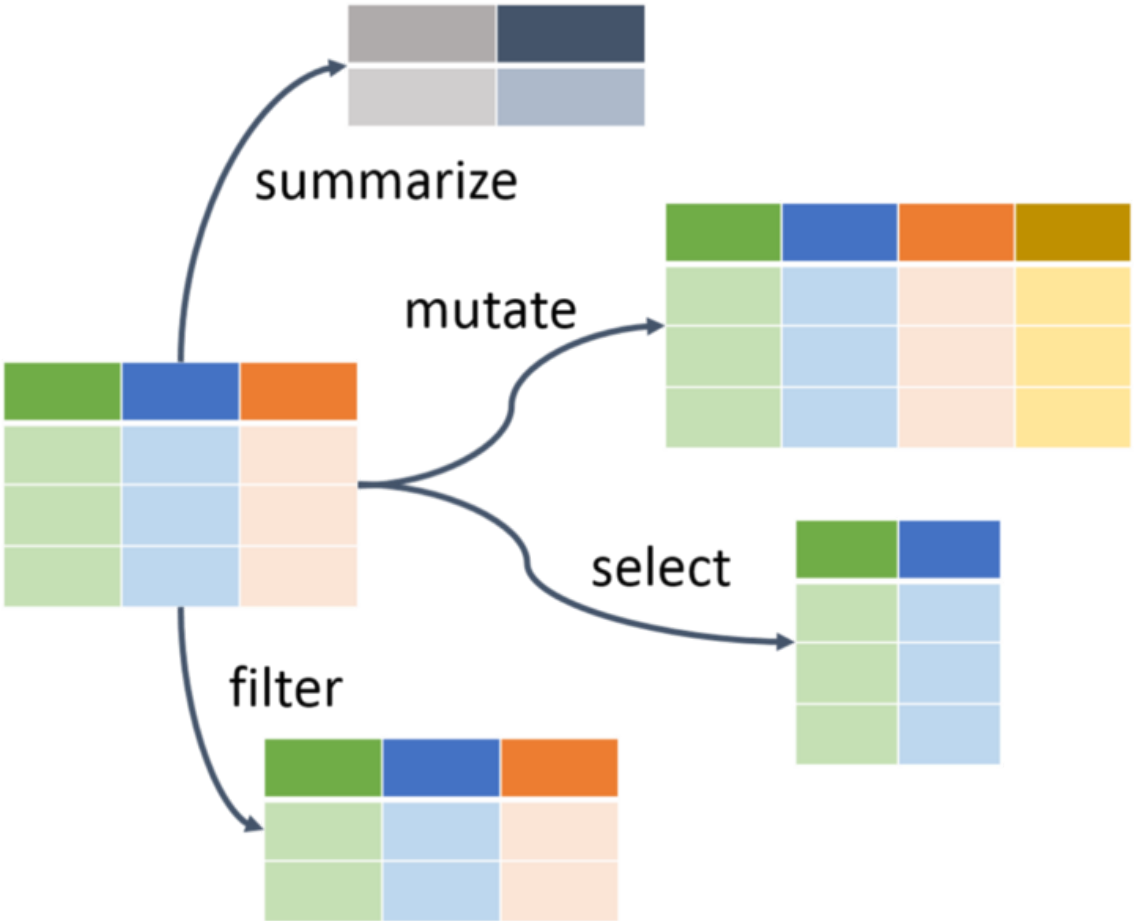


Transformación de datos con `dplyr`

El paquete `dplyr`

Es una gramática de manipulación de datos que proporciona un conjunto consistente de verbos que le ayudan a resolver los desafíos más comunes de manipulación de datos.

Gramática de `dplyr`



Gramática de `dplyr`

Algunas de las principales funciones o "verbos" del paquete `dplyr` son:

- `select()`: Selecciona, renombra y reordena columnas.
- `filter()`: Extrae filas que cumplen un criterio lógico.
- `arrange()`: Ordena las filas según los valores de una columnas .
- `mutate()`: Crea nuevas variables o transforma variables existentes.
- `rename()`: Renombra variables.
- `summarise()`: Calcula un resumen en base a funciones.
- `group_by()`: Agrupa filas a partir de un conjunto de categorías de una o más variables.
- `count()`: Cuenta el número de filas en cada grupo definido por una variable.
- `slice()`: Extraer observaciones (filas) de acuerdo a su posición.

Gramática de `dplyr`

Para llevar a cabo estas acciones debemos tener en cuenta algunas características comunes:

- El primer argumento siempre es un tibble o data.frame
- El resto de los argumentos indican los parametros de lo que queremos hacer
- El resultado siempre tiene estructura de tibble o data frame

Función `case_when`

La función `case_when` de la librería `dplyr`, permite vectorizar múltiples declaraciones *if* y *else if*. Es el equivalente de la sentencia `CASE WHEN` de SQL. Este funciona de la siguiente forma:

```
case_when(<Condición 1> ~ <Resultado si condición 1 es TRUE>,
          <Condición 2> ~ <Resultado si condición 2 es TRUE>,
          ...)
```

Es un comando especialmente útil para recodificar variables categóricas.

Ejemplo:

```
case_when(nota < 1 ~ "Error, ingrese un número entre 1 y 7.",
          nota > 7 ~ "Error, ingrese un número entre 1 y 7.",
          is.numeric(nota) == FALSE ~ "Error, ingrese un número.",
          nota >= 4 ~ "¡Felicitaciones!",
          nota <4 ~ "Reprobaste")
```

Taller práctico

Taller práctico

Utilice el conjunto de datos disponible en `Lock5Data::HollywoodMovies` que contiene datos de películas estrenadas en Hollywood entre el 2012 y 2018.

- Seleccione las variables `Year`, `Movie`, `LeadStudio`, `Genre`, `Budget` y `Profitability`.
- Filtre las películas de los estudios: *"Warner Bros. ", "Universal Pictures ", "Lionsgate" y "Twentieth Century Fox "*.
- Construya una variable llamada `pbudgetmax` calculada como el budget de la película dividida por el máximo valor de budget observado.
- Calcule el promedio, mínimo, máximo, desviación estándar, Q1, Q2 y Q3 de la variable `pbudgetmax` para cada uno de los estudios seleccionados. Luego ordénelos de mayor a menor promedio. ¿Qué puede comentar al respecto?.
- Guarde los resultados anteriores en un archivo .csv.
- Construya una variable categórica llamada `bud_cat` que tendrá la categoría **"Alto"** si `budget >= 100` y **"Bajo"** en caso contrario.

Referencias y Material Complementario

Referencias y Material Complementario

- Sitios Web de tibble, readr, dplyr, tidyr, purrr, stringr y forcats.
- Información del package googlesheets4 si almacena datos en Google Sheets o trabaja con personas que lo hacen.
- Información package para datos en los formatos JSON y XML, para datos almacenados y recuperados de Internet. También, puede ser de interés los packages rvest y httr.
- Data Transformation Cheat Sheet with dplyr.
- Módulo Work with data de RStudio Primers, para practicar el uso de las principales funciones del paquete dplyr.

¡Gracias!

Ana María Alvarado Celis
amalvara@uc.cl

Maite Vergara - Esteban Rucán
maite.vergara@uc.cl - errucan@uc.cl