Tutoría 5

- **Refuerzo Tarea 3**:
  Programación funcional y manejo de tablas con datos reales

Sofía Madariaga

Pontificia Universidad Católica de Chile
Diplomado Ciencia de Datos para Políticas Públicas
Taller de Análisis de Datos I

25 de octubre de 2023

Objetivo tutoría

En esta tutoría 6, vamos a repasar y aprender algunos conceptos y funciones fundamentales para afrontar un ejercicio final, similar a la Tarea 3, que nos dará las herramientas para decidir y afrontar la automatización del procesamiento de datos.

Materiales

Encontrará el material en el portal del diplomado Materiales del Curso > Tutorías.

- **Link:** https://ep.ingenieriauc.cl/mod/folder/view.php?id=87282

Tabla de Contenidos

Limpiar y correcto formato
- Clase de las variables
- Funciones para manipular texto
- Función paste y paste0

Manejo de tablas de datos
- Unir bases de datos
- Pivotar tablas de datos

Programación funcional
- Introducción
- Elementos de la programación funcional

Manejo de directorios y documentos

Ejercicios

Limpiar y correcto formato

Clase de las variables

- **Clase:** el tipo de objeto que R identifica (en el objeto mismo). Verifique para el objeto global con **class** o **glimpse** para cada variable de la base datos.

Siempre es importante **cuidar el formato de nuestros datos**. Utilice la función `glimpse` o `str` para comprobar cada formato.

`c("0", "1", "1", "0", "0")` $\xrightarrow{\texttt{as.numeric()}}$ `c(0, 1, 1, 0, 0)`

Recomendación

- Siempre es recomendable mantener las variables como `factor`, en caso de ser categórica, ya que por coerción siempre puede volver a su valor numérico. Esto no es obligatorio.

¿Dónde está el problema?

```
> glimpse(titanic)
Rows: 891
Columns: 12
$ PassengerId <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,  ...
$ Survived    <chr> "0", "1", "1", "1", "0", "0", "0", "0", "1", ...
$ Pclass      <dbl> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, ...
$ Name        <chr> "Braund, Mr. Owen Harris", "Cumings,  ..." ...
$ Sex         <chr> "male", "female", "female", "female", "male", ...
$ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, ...
$ SibSp       <dbl> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, ...
$ Parch       <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, ...
$ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", ...
$ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, ...
$ Cabin       <chr> NA, "C85", NA, "C123", NA, NA, "E46", NA, NA, ...
$ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", ...
```

Vamos a R →

Limpiar y correcto formato

Funciones para manipular texto

- **Stringr documentación:** https://stringr.tidyverse.org/

En R, podemos manipular el texto con las funciones de **stringr** (funciones **gsub**, **grep**, **grepl** en R Base), permitiendonos reemplazar, eliminar o encontrar texto.

- **str_to_lower.** Permite cambiar un string o vector de strings en *lower case* (todas las letras en minúsculas).
- **str_to_upper.** Permite cambiar un string o vector de strings en *upper case* (todas las letras en mayúsculas).
- **str_replace_all.** Reemplaza un valor por otro.
- **str_detect.** Genera unvector booleano. Busca si el string contiene el patrón.

Limpiar y correcto formato

Funciones para manipular texto

```
# str_to_lower
c("Manzana", "Naranja", "Plátano", "Palta") %>% str_to_lower

# str_to_upper
c("Manzana", "Naranja", "Plátano", "Palta") %>% str_to_upper

# str_replace_all
c("Manzana", "Naranja", "Plátano", "Palta") %>% str_replace_all("a", "4")

# str_detect
c("Manzana", "Naranja", "Plátano", "Palta") %>% str_detect("a$")
```

Recomendación

Estudiar expresiones regulares:
https://cran.r-project.org/web/packages/stringr/vignettes/regular-expressions.html
No relevante para el curso, pero si para su formación en R.

Limpiar y correcto formato

Función paste y paste0

La función **paste** y **paste0** me permite unir dos o más vectores por cada observación en el texto. La diferencia está en que **paste** permite espacios entre los caracteres y la función **paste0** no admite espacio.

```
# A tibble: 6 × 3
  PassengerId   Age id_age
        <dbl> <dbl> <chr>
1           1    22 1_22
2           2    38 2_38
3           3    26 3_26
4           4    35 4_35
5           5    35 5_35
6           6    NA 6_NA
```

Similar a la función **unite** de **tidyr**. **Importante:** cusar **remove = FALSE** para no sobreescribir la primera variable.

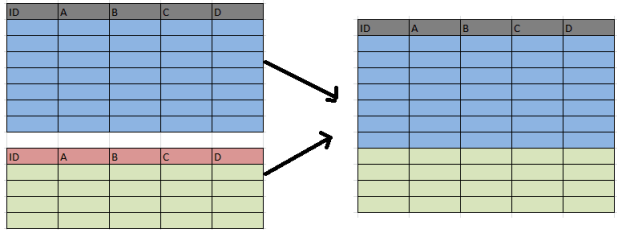
Manejo de tablas de datos

Unir bases de datos

R nos permite unir por **filas** y **columnas**.

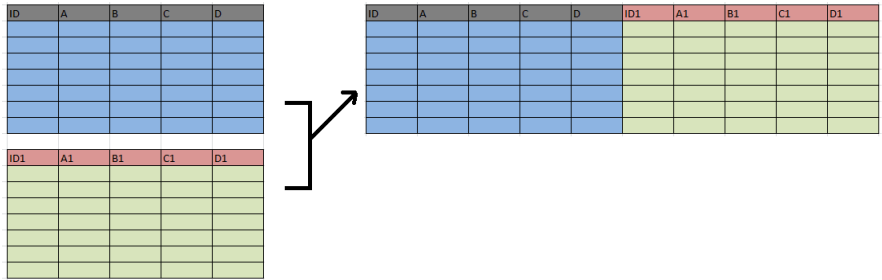
- **rbind().** Método de Rbase. Función para pegar por fila, tanto matrices como bases de datos.
- **bind_rows().** Método de **dplyr**. Es una función para pegar por fila, flexible para bases de datos.
- **cbind().** Método de Rbase. Función para pegar por columna, tanto matrices como bases de datos. **No recomendada para bases de datos, ya que no respeta el carácter relacional de los datos**.
- **funciones join.** Para cualquier unión en donde se requiera unir dos bases de datos que compartan información con alguna varable (key) (revisar tutoría pasada).

Unir por fila
`rbind`
`bind_rows`



Unir por columna
`cbind`

Uniones relacionales
`{.sufijo}_join`

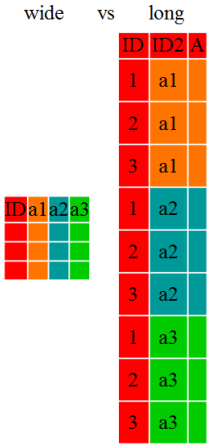


Imágenes de https://www.analytics-tuts.com/reshaping-data-in-r/

Manejo de tablas de datos

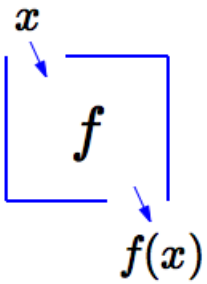
Pivotar tablas de datos

- **pivot_longer:** nos permite transformar datos anchos (wide), a datos largos.
- **pivot_wider:** nos permite transformar datos largos a datos anchos.



Programación funcional

Introducción



- En R, el paradigma predominante de programación es la **programación funcional**.
- No solo implica la aplicación de funciones, sino que implica la creación de funciones. Para crear funciones o construir nuestras propias mecánicas necesitamos algunos procesos básicos: iterativos y/o vectorizados.

Programación funcional

Elementos de la programación funcional

Bucles: procesos iterativos basados en un índice.

- **Bulcle for:** recorriendo un número definido de elementos.
- **Bucle while:** iterando hasta que se cumpla una condición.

Familia apply: aplicación de funciones a todos los elementos de la estructura.

- **sapply:** nos permite aplicar una misma función a todas las variables de una base de datos y devuelve una matriz.
- **lapply:** nos permite aplicar una misma función a todas las variables de una base de datos y nos devuelve una lista.
- **apply:** aplicamos una función a través de filas o columnas en una matriz.
- **tapply:** aplicamos una función a subconjuntos de datos basados en una categoría o factor.

categoría o factor.

Manejo de directorios y documentos

Ver los archivos de un directorio

```
dir("directorio")
list.files("directorio, pattern = ".csv$")
```

Crear carpetas y archivos (o abrir archivso existentes)

```
dir.create("Output")
file.edit("Output/script.R")
```

Ejercicios

Ejercicio final

A continuación, usted trabajará con 4 bases de datos, aplicando programación funcional.

- `amazon_prime_titles.csv`
- `disney_plus_titles.csv`
- `hulu_titles.csv`
- `netflix_titles.csv`

Ejercicios

Las cuatro cuentan con las mismas variables:

- `show_id`: identificador por show (no coincide entre las bases).
- `type`: tipo de audiovisual: TV Show o Movie.
- `title`: título del audiovisual.
- `director`: director del audiovisual.
- `cast`: reparto del audiovisual.
- `ountry`: país de filmación.
- `date_added`: día añaddo a la plataforma.
- `release_year`: año de realización.
- `rating`: calificación del audiovisual.
- `duration`: duración del audiovisual.
- `listed_in`: categorías en las que ha sido catalogada.
- `description`: reseña del audiovisual.
- `source`: a qué plataforma pertenecen: Netflix, Amazon prime, Hulu, Disney+

Ejercicios

El cliente desea contar con una base de datos eficiente que le permita buscar sus películas favoritas en la plataforma de su elección. Dado que todavía hay servicios de transmisión por agregar, se busca automatizar un proceso que consolide y fusione estas tablas de datos, proporcionando al cliente una visión completa de todas sus opciones, incluso para el futuro.

1. Proponga un método para unir las bases de datos.
2. Al proceso anterior, proponga un procesamiento para Para uniformar el nombre de las películas transformando en minúculas y elimine cualquier signo de puntuación, tilde y carácter extraño.
3. Al proceso anterior añada: genere una variable "fecha_acotada"que sea: "`mes, año`" de la variable date_added.
4. Al proceso anterior, añada: elimine la "s"del identificador (show_id).

5. Con la base de datos unida, genere una base de datos **wide** usando la variable source, y donde se indique qué películas pertenecen a qué servicio de *streming*.
6. Esa misma tabla de datos: vuelva a transformar a long data.
7. Genere una función en el que automatice el proceso indicando el directorio de los datos. Al final, la función tiene que entregar una **lista** con las bases de datos: `data_master` (base dedatos original), `data_master_wide` y `data_master_long`.

Función final

procesamiento $<-$ **function(**directorio**)**{

1. Guardo el nombre de los archivos.
2. Genero una lista vacía

   for(i in elementos){

   3. Cargo el archivo *de cada base de datos*.
   4. Limpio el texto de los títulos *de cada base de datos*.
   5. Genero la variable de fecha *de cada base de datos*.
   6. Limpio la "s"del identificador *de cada base de datos*.
   7. Guardo *cada base de datos* en una lista.

   }

8. Uno la base de datos generando la base `data_master`.
9. Pivoteo la base de datos generando una base de datos wide llamada `data_master_long`.
10. Pivoteo la base anterior generando una base de datos long llamada `data_master_long`.

    return(`list(data_master, data_master_wide, data_master_long)` **)**

}

Propuesto

PROPUESTO: haga el mismo ejercicio, pero aplicando lapply y siguiendo el ejemplo del script que se presenta en la sección de lapply (líneas 237-260) y siguiendo el ejercicio (a) en cuanto a la estrategia para importar varias tablas de este Ejercicio final.

Bibliografía recomendada

- **Advacned R**. Link: https://adv-r.hadley.nz/
- **Modern R with the tidyverse**. Link: https://modern-rstats.eu/
- **R for Data Science**. Link: https://r4ds.had.co.nz/