LAB 25 (MD)

AnaGSanjuanM

2023-02-23

————————LABORATORIO 25————————————

Tidy data - datos ordenados - Parte 1

Objetivo: Introducción práctica a los datos ordenados (o tidy data) y a las herrameintas que provee el paquete tidyr.

En este ejecicio vamos a:

1. Cargar datos (tibbles)
2. Convesrtir nuestros tibbles en datadrames (para su exportación)
3. Exportar dataframes originales
4. Pivotar tabla 4a
5. Exportar resultado (TABLA PIVOTANTE)

Prerrequisitos. instalar paquete tidyverse

install.packages(“tidyverse”)

Instalar paquete de datos

install.packages(“remotes”)

remotes::install_github(“cienciadedatos/datos”)

install.packages(“datos”)

Cargar paquete tidyverse

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.0     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.1.8
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors
```

Cargar paquete de datos

```
library("datos")
```

Tabla 1 hasta tabla 4b. Ver datos como tibble

```
datos::tabla1
```

```
## # A tibble: 6 × 4
##   pais        anio  casos  poblacion
##   <chr>      <dbl>  <dbl>      <dbl>
## 1 Afganistán  1999    745   19987071
## 2 Afganistán  2000   2666   20595360
## 3 Brasil      1999  37737  172006362
## 4 Brasil      2000  80488  174504898
## 5 China       1999 212258 1272915272
## 6 China       2000 213766 1280428583
```

Cargar las otras cuatro tablas del paquete de datos

```
datos::tabla2
```

```
## # A tibble: 12 × 4
##    pais        anio tipo          cuenta
##    <chr>      <dbl> <chr>          <dbl>
##  1 Afganistán  1999 casos            745
##  2 Afganistán  1999 población   19987071
##  3 Afganistán  2000 casos           2666
##  4 Afganistán  2000 población   20595360
##  5 Brasil      1999 casos          37737
##  6 Brasil      1999 población  172006362
##  7 Brasil      2000 casos          80488
##  8 Brasil      2000 población  174504898
##  9 China       1999 casos         212258
## 10 China       1999 población 1272915272
## 11 China       2000 casos         213766
## 12 China       2000 población 1280428583
```

```
datos::tabla3
```

```
## # A tibble: 6 × 3
##   pais        anio tasa
##   <chr>      <dbl> <chr>
## 1 Afganistán  1999 745/19987071
## 2 Afganistán  2000 2666/20595360
## 3 Brasil      1999 37737/172006362
## 4 Brasil      2000 80488/174504898
## 5 China       1999 212258/1272915272
## 6 China       2000 213766/1280428583
```

```
datos::tabla4a
```

```
## # A tibble: 3 × 3
##   pais       `1999` `2000`
##   <chr>       <dbl>  <dbl>
## 1 Afganistán    745   2666
## 2 Brasil      37737  80488
## 3 China      212258 213766
```

```
datos::tabla4b
```

```
## # A tibble: 3 × 3
##   pais           `1999`     `2000`
##   <chr>           <dbl>      <dbl>
## 1 Afganistán   19987071   20595360
## 2 Brasil      172006362  174504898
## 3 China      1272915272 1280428583
```

La visualización de los datos es como tibble, es como un dataframe pero no se puede exportar, cambiar variables (hay ciertas limitantes).

Convirtiendo tibble a dataframe

Primer datadrame (df1) que sea igual a la importación de un data_frame de la tabla1.

```
df1 <- data_frame(tabla1)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## ℹ Please use `tibble()` instead.
```

Ahora ya aparece en el Environment y se puede exportar

Se ejecuta la misma indicación con las tablas restantes, asigando nombre de dataframe con base en la tabla.

```
df2 <- data_frame(tabla2)
df3 <- data_frame(tabla3)
df4a <- data_frame(tabla4a)
df4b <- data_frame(tabla4b)
```

Para visualizar

```
df1
```

```
## # A tibble: 6 × 4
##   pais        anio  casos  poblacion
##   <chr>      <dbl>  <dbl>      <dbl>
## 1 Afganistán  1999    745   19987071
## 2 Afganistán  2000   2666   20595360
## 3 Brasil      1999  37737  172006362
## 4 Brasil      2000  80488  174504898
## 5 China       1999 212258 1272915272
## 6 China       2000 213766 1280428583
```

```
df2
```

```
## # A tibble: 12 × 4
##    pais        anio tipo          cuenta
##    <chr>      <dbl> <chr>          <dbl>
##  1 Afganistán  1999 casos            745
##  2 Afganistán  1999 población   19987071
##  3 Afganistán  2000 casos           2666
##  4 Afganistán  2000 población   20595360
##  5 Brasil      1999 casos          37737
##  6 Brasil      1999 población  172006362
##  7 Brasil      2000 casos          80488
##  8 Brasil      2000 población  174504898
##  9 China       1999 casos         212258
## 10 China       1999 población 1272915272
## 11 China       2000 casos         213766
## 12 China       2000 población 1280428583
```

```
df3
```

```
## # A tibble: 6 × 3
##   pais        anio tasa
##   <chr>      <dbl> <chr>
## 1 Afganistán  1999 745/19987071
## 2 Afganistán  2000 2666/20595360
## 3 Brasil      1999 37737/172006362
## 4 Brasil      2000 80488/174504898
## 5 China       1999 212258/1272915272
## 6 China       2000 213766/1280428583
```

```
df4a
```

```
## # A tibble: 3 × 3
##   pais       `1999` `2000`
##   <chr>       <dbl>  <dbl>
## 1 Afganistán    745   2666
## 2 Brasil      37737  80488
## 3 China      212258 213766
```

```
df4b
```

```
## # A tibble: 3 × 3
##   pais           `1999`     `2000`
##   <chr>           <dbl>      <dbl>
## 1 Afganistán   19987071   20595360
## 2 Brasil      172006362  174504898
## 3 China      1272915272 1280428583
```

¿Cuál de las bases está ordenada y cuál está desordenada?

Los datos irdenados tienen tres características esenciales

1. Cada variable debe contener su propia columna
2. Cada observación debe tener su propia fila
3. Cada valor debe tener su propia celda

Exportar los dataframes originales

```
write.csv(df1,file="df1.csv")
```

Se hace lo mismo con los otros dataframes

```
write.csv(df2,file="df2.csv")
write.csv(df3,file="df3.csv")
write.csv(df4a,file="df4a.csv")
write.csv(df4b,file="df4b.csv")
```

Ordenando los datos.

Explicación de tibble en cuadrante de visualizaciones

```
vignette("tibble")
```

```
## starting httpd help server ... done
```

La mayoría de las funciones que usarás en este libro, producen tibbles, ya que son una de las caracterísitcas transversales del tidyverse.

Si ya te has familiarizado con data.frame(), es importante que tomes en cuanta que tibble() hace menos cosas.

Nunca cambia el tipo de los inputs (p. ej., ¡nunca convierte caracteres en factores!)

Nunca cambia el nombre de las variables y nunca asigna nombres a las filas.

Ordenar datos con la tabla4a (PIVOTAR). Se añade el operador pipe %>% (presionando ctrl Shift M)

Se genera objeto llamado t4a_PIVOTANTE (será una tabla ordenada), para pivotear a lo largo (pivot_longer)

Las columnas están dadas por los años y se reemplazará el nombre por anio en los que englobará los dos momentos en el tiempo.

Los valores se tomarán como casos.

```
t4a_PIVOTANTE = tabla4a %>%
  pivot_longer(cols=c("1999", "2000"), names_to="anio", values_to = "casos")
```

Para visualizar

```
t4a_PIVOTANTE
```

```
## # A tibble: 6 × 3
##   pais       anio   casos
##   <chr>      <chr>  <dbl>
## 1 Afganistán 1999     745
## 2 Afganistán 2000    2666
## 3 Brasil     1999   37737
## 4 Brasil     2000   80488
## 5 China      1999  212258
## 6 China      2000  213766
```

Exportar resultados: tabla ordenada

```
write.csv(t4a_PIVOTANTE, file="t4a_PIVOTANTE.csv")
```

————————————FIN DE LABORATORIO 25————————————————