LAB 27 (MD)

AnaGSanjuanM

2023-02-23

————————LABORATORIO 27————————

Tidy data - datos ordenados - Parte 3

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")
```

Cargar las tablas del paquete de 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
```

```
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
```

Se obtienen los dataframe de las tablas

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

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

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

Exportar los dataframes originales

```
write.csv(df1,file="df1.csv")
write.csv(df2,file="df2.csv")
write.csv(df3,file="df3.csv")
write.csv(df4a,file="df4a.csv")
write.csv(df4b,file="df4b.csv")
```

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
```

1. Pivotar tabla 4b

Ordenar datos con la tabla4b (PIVOTAR). Se añade el operador pipe %>% (presionando ctrl Shift M)

Se genera objeto llamado t4b_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 poblacion.

```
t4b_PIVOTANTE = tabla4b %>%
  pivot_longer(cols=c("1999", "2000"), names_to="anio", values_to = "poblacion")
```

Para visualizar

```
t4b_PIVOTANTE
```

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

2. Combinar las versiones ordenadas de tabla4a y tabla4b (ocupando dplyr)

Generar objeto llamado unión de las tablas 4 (a y b) que será igual a una unión a partir de la izquierda (left_join)

Se unen la t4a_PIVOTANTE y la t4b_PIVOTANTE

```
union_t4 = left_join(t4a_PIVOTANTE, t4b_PIVOTANTE)
```

```
## Joining with `by = join_by(pais, anio)`
```

Para visualizar

```
union_t4
```

```
## # A tibble: 6 × 4
##   pais       anio   casos  poblacion
##   <chr>      <chr>  <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
```

Exportar resultados tabla4a + tabla4b (ordenada)

```
write.csv(union_t4, file="union_t4.csv")
```

3. DATOS ANCHOS CON TABLA 2

Para visualizar

```
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
```

Es una base de datos larga, por lo que debemos hacerla ancha

Pivotar tabla 2 “A LO ANCHO”

Se crea objeto llamdo t2_ancha. Será igual a la tabla2 original pero con filtro (operador pipe)

Como el pivotaje será a lo ancho, se ocupará pivot_wider.

Se establecen nombres (names_from) que será igual a tipo. Values_from será igual a cuenta

```
t2_ancha = tabla2 %>%
  pivot_wider(names_from = tipo, values_from = cuenta)
```

Para vsisualizar

```
t2_ancha
```

```
## # A tibble: 6 × 4
##   pais        anio  casos  población
##   <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
```

Exportar la tabla ordenada recien creada

```
write.csv(t2_ancha, file = "t2_ancha.csv")
```

————-INICIO DE LAB27———

FUNCIÓN SEPARAR

PRIMERA SEPARACIÓN

La tabla3, que tiene un problema diferente

Ver tabla3

```
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
```

Tenemos una columna (tasa) en la que se ve una variable partida en dos (númerador: número de casos y denominador:la población)

Para tener una base ordenada, necesitamos separar la columna tasa en casos y población

El objeto se llamará SEPARADO_1 el cual tendrá como input la tabla3

Se coloca el operador pipe para filtrar: separar (separate) la columna tasa, es decir, partirla en dos columnas: “casos” y “poblacion”

```
SEPARADO_1 = tabla3 %>%
  separate(tasa, into = c("casos", "poblacion"))
```

Para visualizar

```
SEPARADO_1
```

```
## # A tibble: 6 × 4
##   pais        anio casos  poblacion
##   <chr>      <dbl> <chr>  <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
```

SEGUNDA SEPARACIÓN: POR CARACTERES QUE SEPARAN LOS DATOS

Tenemos una columna (tasa) en la que se ve una variable partida en dos (númerador: número de casos y denominador:la población)

Para tener una base ordenada, necesitamos separar la columna tasa en casos y población

El objeto se llamará SEPARADO_2 el cual tendrá como input la tabla3

Se coloca el operador pipe para filtrar: separar (separate) la columna tasa, es decir, partirla en dos columnas: “casos” y “poblacion”

Se añadirá que la separación sea dada por el caracter / (la diagonal) que aparece en el dataframe (df3)

```
SEPARADO_2 = tabla3 %>%
  separate(tasa, into = c("casos", "poblacion") , sep = "/")
```

Para visualizar

```
SEPARADO_2
```

```
## # A tibble: 6 × 4
##   pais        anio casos  poblacion
##   <chr>      <dbl> <chr>  <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
```

TERCERA SEPARACIÓN: DIVISIÓN DE AÑOS: COLUMNA SIGLO Y COLUMNA ANIO

Tenemos una columna (anio) en la que se ven dos tipos de datos (1999 y 2000)

Para tener una base ordenada, necesitamos separar la columna anio en siglos y anios

El objeto se llamará SEPARADO_3 el cual tendrá como input la tabla3

Se coloca el operador pipe para filtrar: separar (separate) la columna anio, es decir, partirla en dos columnas: “siglo” y “anio”

La separación estará dada por dos primeros números

```
SEPARADO_3 = tabla3 %>%
  separate(anio, into = c("siglo", "anio"), sep = 2)
```

Para visualizar

```
SEPARADO_3
```

```
## # A tibble: 6 × 4
##   pais       siglo anio  tasa
##   <chr>      <chr> <chr> <chr>
## 1 Afganistán 19    99    745/19987071
## 2 Afganistán 20    00    2666/20595360
## 3 Brasil     19    99    37737/172006362
## 4 Brasil     20    00    80488/174504898
## 5 China      19    99    212258/1272915272
## 6 China      20    00    213766/1280428583
```

Ahora vamos a unir la tabla generada anteriormente

Podemos usar unite() para unir las columnas siglo y anio creadas en el ejemplo anterior

Creamos onjeto llamado UNION_1, que será igual a la tabla SEPARADO_3.

Se activa el operador pipe %>% para que una (unit), es decir, se hará una nueva columna que esté integrada por la columna siglo y la couna anio

```
UNION_1 = SEPARADO_3 %>%
  unite(nueva, siglo, anio)
```

Para visualizar

```
UNION_1
```

```
## # A tibble: 6 × 3
##   pais       nueva tasa
##   <chr>      <chr> <chr>
## 1 Afganistán 19_99 745/19987071
## 2 Afganistán 20_00 2666/20595360
## 3 Brasil     19_99 37737/172006362
## 4 Brasil     20_00 80488/174504898
## 5 China      19_99 212258/1272915272
## 6 China      20_00 213766/1280428583
```

La nueva columna muestra un guion bajo, por lo que debemos retirarlo

En este caso también necesitamos el argumento sep por defecto

Pondrá un guión bajo (_) entre los valores de las distintas columnas. Si no queremos ningún separador usamos “”

Podemos usar unite() para unir las columnas siglo y anio creadas

Creamos objeto llamado UNION_2, que será igual a la tabla SEPARADO_3.

Se activa el operador pipe %>% para que una (unit), es decir, se hará una nueva columna que esté integrada por la columna siglo y la couna anio

Pero que en la separación omita cualquier caracter

```
UNION_2 = SEPARADO_3 %>%
  unite(nueva, siglo, anio, sep="")
```

Para visualizar

```
UNION_2
```

```
## # A tibble: 6 × 3
##   pais       nueva tasa
##   <chr>      <chr> <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
```

————————————FIN DE LABORATORIO 27——————-