

Lab27 (Markdown)

Alejandra Guzmán

2024-03-12

LABORATORIO: Tidy data (datos ordenados) 1 Prerrequisitos: Instalar paquete tidyverse Instalar paquete de datos

Cargar paquete tidyverse

```
library(tidyverse)

## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr   1.5.1
## ✓ ggplot2    3.5.0      ✓ tibble     3.2.1
## ✓ lubridate  1.9.3      ✓ tidyr      1.3.1
## ✓ purrr      1.0.2
## — Conflicts — tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Cargar paquete de datos

```
library(datos)
```

Ver datos como tibble (tabla 1 hasta tabla 4b)

```
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
```

Ver datos como dataframe

```
dtf<-data_frame(tabla1)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## i Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
dtf2<-data_frame(tabla2)
dtf3<-data_frame(tabla3)
dtf4a<-data_frame(tabla4a)
dtf4b<-data_frame(tabla4b)
```

Exportar los dataframes originales

```
write.csv(dtf,file="dtf1.csv")
write.csv(dtf2,file="dtf2.csv")
write.csv(dtf3,file="dtf3.csv")
write.csv(dtf4a,file="dtf4a.csv")
write.csv(dtf4b,file="dtf4b.csv")
```

Explicación de tibble

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

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

Ordenar datos con la tabla4a (PIVOTAR)

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

Exportar resultado: tabla ordenada

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

PARTE 2 EJERCICIO 2: PIVOTAR Y UNIR TABLAS 1. Pivotar tabla 4b “A LO LARGO”

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

Unir tablas ordenadas

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

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

Exportar resultado: tabla4a + tabla4b (ordenada)

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

EJERCICIO 3: DATOS ANCHOS CON TABLA 2 1. Pivotar tabla 2 “A LO ANCHO” Ordenar datos con la tabla 2 (PIVOTAR a lo ANCHO)

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

Exportar resultado: tabla ordenada

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

PARTE 3 La tabla3, que tiene un problema diferente: tenemos una columna (tasa) que contiene dos variables (casos y población). Separación 1 Separar casos y población por default

```
Separado_1 = tabla3 %>%  
  separate(tasa, into=c("casos", "población"))
```

Separación 2 Separar casos y población por caracter “/” “*” “_”

```
Separado_2 = tabla3 %>%  
  separate(tasa, into=c("casos", "población"), sep="/")
```

Separación 3 Separar siglo y año de la columna “anio”

```
Separado_3 = tabla3 %>%  
  separate(anio, into=c("siglo", "año"), sep=2)
```

Ahora vamos a unir la tabla separada anteriormente Podemos usar unite() para unir las columnas siglo y anio creadas en el ejemplo anterior

```
UNION_1= Separado_3 %>%  
  unite(nueva, siglo, año)
```

En este caso también necesitamos el argumento sep. Por defecto, pondrá un guion bajo (_) entre los valores de las distintas columnas. Si no queremos ningún separador usamos ""

```
UNION_2= Separado_3 %>%  
  unite(nueva, siglo, año, sep="")
```