

L24 LAB 05 (MD)

AnaGSanjuanM

2023-02-22

Cargar paquete tidyverse y readr

```
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
```

```
library(readr)
```

CARGAR DATOS

En la ventana de Data (al frente) damos clic en Import Dataset y buscamos la ubicación del archivo que generamos, el ICE 2014 en csv. Se puede importar desde la ventana que se abrió. También, en la parte inferior se mostrará el código, lo copiamos y pegamos aquí para correrlo pero antes sustituimos lo que dice ICE_2014 por tabla1

Este es el código original obtenido

```
library(readr)
```

```
ICE_2014 <- read_csv("ICE 2014.csv")
```

```
## Rows: 83 Columns: 15
## — Column specification —
## Delimiter: ","
## chr (1): MUNICIPIO
## dbl (14): Km,0, Km,2, Km,4, Km,6, Km,8, Km,10, Km,12, Km,14, Km,16, Km,18, K...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Este es el código modificado que se correrá

```
tabla1 <- read_csv("ICE 2014.csv")
```

```
## Rows: 83 Columns: 15
## — Column specification —
## Delimiter: ","
## chr (1): MUNICIPIO
## dbl (14): Km,0, Km,2, Km,4, Km,6, Km,8, Km,10, Km,12, Km,14, Km,16, Km,18, K...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Para ver los nombres de la tabla

```
names(tabla1)
```

```
## [1] "MUNICIPIO" "Km,0"      "Km,2"      "Km,4"      "Km,6"      "Km,8"
## [7] "Km,10"     "Km,12"     "Km,14"     "Km,16"     "Km,18"     "Km,20"
## [13] "Km,22"     "Km,24"     "Km,26"
```

INSTRUCCIONES

Se tiene una matriz ordenada a lo ancho

1. Se requiere pivotar la tabla "a lo largo". Generar tabla que se llama ti_PIVOTANTE. Como input debe tener la tabla1 que se acaba de cargar. Colocar operador pip %>% (que es un filtro). Entonces se pivotea a lo largo las columnas de Km. Se llamará iteración y su valor será el ranking

```
t1_PIVOTANTE = tabla1 %>%  
  pivot_longer(cols = c("Km,0" , "Km,2" , "Km,4" , "Km,6" , "Km,8" , "Km,10" , "Km,12" , "Km,14" , "Km,16" , "Km,  
18" , "Km,20" , "Km,22" , "Km,24" , "Km,26" ),names_to = "iteracion", values_to = "ranking")
```

Ahora se va a exportar

En formato separado por comas csv. El input será la tabla pivotante t1_PIVOTANTE que se acaba de generar. El archivo se llamará ice_CHARTICULATOR.csv

```
write.csv(t1_PIVOTANTE, file = "ice_CHARTICULATOR.CSV")
```

Para saber en qué ruta se guardó ejecuto el comando. Está en los documentos.

```
getwd()
```

```
## [1] "C:/Users/Lenovo/Documents/GitHub/LAB-24/L24 Output/HTML"
```