

13. -Data Visualization_CU_18_20_infra_meteo_v_01

June 13, 2023

#

CU18__Infraestructuras_eventos

Citizenlab Data Science Methodology > II - Data Processing Domain *** > # 13.- Data Visualization

Data Visualization is the process of performing a statistical graphical analysis of the data.

0.1 Notas

- La visualización es parte del análisis exploratorio de datos
- En los notebooks del proceso 12 se incluyen gráficos que constituyen la visualización de los datos del caso de uso, y no se repiten aquí
- En los notebooks del proceso 13 se añaden algunas visualizaciones espaciales no incluidas en el proceso 12

0.2 File

En este notebook se importan dos archivos: el csv con los datos y el json con las geometrías

- Input File: CU_18_09.3_20_diario_infra
- Output File: No aplica

0.2.1 Encoding

Con la siguiente expresión se evitan problemas con el encoding al ejecutar el notebook. Es posible que deba ser eliminada o adaptada a la máquina en la que se ejecute el código.

```
[1]: Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8")
```

```
'LC_CTYPE=es_ES.UTF-8;LC_NUMERIC=C;LC_TIME=es_ES.UTF-8;LC_COLLATE=es_ES.UTF-8;LC_MONETARY=es_ES.UTF-8;LC_MESSAGES=en_US.UTF-8;LC_PAPER=es_ES.UTF-8;LC_NAME=C;LC_ADDRESS=C;LC_TELEPHONE=C;LC_MEASUREMENT=es_ES.UTF-8;LC_IDENTIFICATION=C'
```

0.3 Settings

0.3.1 Libraries to use

```
[8]: library(readr)
library(dplyr)
# library(sf)
library(tidyr)
library(ggplot2)
# library(summarytools)
library(GGally)
library(nortest)
library(lubridate)
# library(leaflet)
```

0.3.2 Paths

```
[9]: iPath <- "Data/Input/"
oPath <- "Data/Output/"
```

0.4 Data Load

1. Archivo de datos CSV

OPCION A: Seleccionar fichero en ventana para mayor comodidad

Data load using the {tcltk} package. Uncomment the line if using this option

```
[ ]: # file_data <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

```
[10]: iFile <- "CU_18_09.3_20_diario_infra.csv"
file_data <- paste0(iPath, iFile)

if(file.exists(file_data)){
  cat("Se leerán datos del archivo: ", file_data)
} else{
  warning("Cuidado: el archivo no existe.")
}
```

Se leerán datos del archivo: Data/Input/CU_18_09.3_20_diario_infra.csv

Data file to dataframe Usar la función adecuada según el formato de entrada (xlsx, csv, json, ...)

```
[11]: data <- read_csv(file_data)
```

Rows: 377727 Columns: 10
Column specification

Delimiter: ","

```
dbl (9): id_inf, capacidad, demanda, evento_infra, evento_zona, tmed,  
prec,...
```

```
date (1): fecha
```

Use ``spec()`` to retrieve the full column specification for this data.

Specify the column types or set ``show_col_types = FALSE`` to quiet this message.

2. Archivo de geometrías JSON

OPCION A: Seleccionar fichero en ventana para mayor comodidad

Data load using the `{tcltk}` package. Uncomment the line if using this option

```
[12]: # file_data <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

```
[ ]: # iFileg <- "xxxx"  
# file_geo <- paste0(iPath, iFileg)  
  
# if(file.exists(file_geo)){  
#   cat("Se leerán datos del archivo: ", file_geo)  
# } else{  
#   warning("Cuidado: el archivo no existe.")  
# }
```

Data file to dataframe Usar la función adecuada según el formato de entrada (xlsx, csv, json, ...)

```
[ ]: # datageo <- st_read(file_geo)
```

0.5 Data join

Unimos los dos data frames

```
[13]: # dataj <- data |>  
#   full_join(datageo, by = c("GEOCODIGO", "DESBDT"))
```

0.6 Data Structure

Estructura de los datos:

```
[ ]: # dataj |> glimpse()
```

Muestra de los primeros datos:

```
[ ]: # dataj |> slice_head(n = 5)
```

0.7 Data Visualization

0.7.1 Map

Filtrar y agrupar los datos a mostrar en el mapa

```
[ ]: # mdata <- dataj |>
#   group_by(geometry) |>
#   summarise(valor = sum(n_vacunas, na.rm = TRUE))

[ ]: # pal <- colorNumeric(palette = "Blues",
#                         domain = mdata$valor)

# mdata |>
#   leaflet() |>
#   addTiles() |>
#   addPolygons(color = "#444444",
#               weight = 1,
#               smoothFactor = 0.5,
#               fillOpacity = 1,
#               fillColor = ~pal(valor),
#               highlightOptions = highlightOptions(color = "white", weight = 2,
#                                                    bringToFront = TRUE),
#               # popup = ~paste0(DESBTD, " (", GEOCODIGO, ")"),
#               label = ~paste0(valor, " vacunas")) |>
#   addLegend("bottomright",
#           pal = pal,
#           values = ~valor,
#           title = "Número de vacunas",
#           labFormat = labelFormat(big.mark = " "),
#           opacity = 1
#   )
```

0.8 REPORT

A continuación se realizará un informe de las acciones realizadas

0.9 Main Actions Carried Out

- No aplica la visualización de mapas en este caso de uso

0.10 Main Conclusions

- Los datos son adecuados para el caso de uso

0.11 CODE TO DEPLOY (PILOT)

A continuación se incluirá el código que deba ser llevado a despliegue para producción, dado que se entiende efectúa operaciones necesarias sobre los datos en la ejecución del prototipo

Description

- No hay nada que desplegar en el piloto, ya que estos datos son estáticos o en todo caso cambian con muy poca frecuencia, altamente improbable durante el proyecto.

CODE

[]: