

## 13. -Data Visualization\_04\_06\_turismo\_gasto\_completo\_v\_01

June 15, 2023

```
#
CU55_Modelo agregado de estimación del gasto medio por turista
Citizenlab Data Science Methodology > II - Data Processing Domain *** > # 13.- Data Visual-
ization
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\_55\_08\_03\_gasto\_municipio.csv
- Sampled Input File: CU\_45\_07\_03\_gasto\_municipio.csv
- 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.

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

'LC_COLLATE=es_ES.UTF-8;LC_CTYPE=es_ES.UTF-8;LC_MONETARY=es_ES.UTF-
8;LC_NUMERIC=C;LC_TIME=es_ES.UTF-8'
```

### 0.3 Settings

#### 0.3.1 Libraries to use

```
In [3]: library(readr)
library(dplyr)
library(tidyr)
```

```
library(ggplot2)
library(summarytools)
library(GGally)
library(sf)
library(nortest)
library(lubridate)
library(leaflet)
```

### 0.3.2 Paths

```
In [10]: 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. Ucomment the line if using this option

```
In [11]: # file_data <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

```
In [12]: iFile <- "CU_55_08_03_gasto_municipio.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\_55\_08\_03\_gasto\_municipio.csv

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

```
In [13]: data <- read_csv(file_data)
```

Rows: 50294 Columns: 10

Column specification

Delimiter: ","

chr (5): mes, pais\_orig\_cod, pais\_orig, mun\_dest, CMUN

dbl (4): mun\_dest\_cod, turistas, gasto, Target

lgl (1): is\_train

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

```
In [14]: # file_data <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

```
In [15]: iFileg <- "CU_45_05_01_municipios_geo.json"
         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.")
         }
```

Se leerán datos del archivo: Data/Input/CU\_45\_05\_01\_municipios\_geo.json

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

```
In [16]: datageo <- st_read(file_geo)
```

Reading layer `CU\_45\_05\_01\_municipios\_geo' from data source

`C:\Vicente\_SPIKA\casos\_urjc\notebooks\dominios\_II\_y\_III\55\_turismo\_gasto\Data\Input\CU\_45\_05\_01\_municipios\_geo.json' using driver `GeoJSON'

Simple feature collection with 179 features and 7 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: -4.57908 ymin: 39.88476 xmax: -3.05282 ymax: 41.16499

Geodetic CRS: WGS 84

## 0.5 Data join

Unimos los dos data frames

```
In [17]: data
```

	mes <chr>	pais_orig_cod <chr>	pais_orig <chr>	mun_dest_cod <dbl>	mun_dest <chr>
	2019-08	110	Francia	28161	Valdemoro
	2021-07	010	Total Europa	28176	Villanueva
	2021-07	010	Total Europa	28132	San Martí
	2022-01	000	Total	28141	Sevilla la N
	2019-08	128	Rumania	28130	San Fernan
	2022-07	000	Total	28126	Robregord
	2022-08	011	Total Unión Europea	28075	Loeches
	2022-03	126	Alemania	28005	Alcalá de H
	2020-01	121	Países Bajos	28066	Griñón
	2021-09	110	Francia	28005	Alcalá de H
	2020-05	121	Países Bajos	28047	Collado Vi
	2020-06	351	Venezuela	28065	Getafe
	2021-05	010	Total Europa	28100	Nuevo Baz
	2021-02	123	Portugal	28113	Pinto
	2020-01	010	Total Europa	28005	Alcalá de H
	2021-08	131	Suecia	28074	Leganés
	2021-07	000	Total	28034	Canencia
	2019-12	302	EE.UU.	28047	Collado Vi
	2020-04	213	Egipto	28079	Madrid
	2022-10	110	Francia	28066	Griñón
	2021-03	102	Austria	28007	Alcorcón
	2019-11	126	Alemania	28005	Alcalá de H
	2020-08	128	Rumania	28148	Torrejón de
	2021-07	117	Luxemburgo	28080	Majadahon
	2021-07	407	China	28007	Alcorcón
	2022-04	340	Argentina	28134	San Sebast
	2019-07	228	Marruecos	28005	Alcalá de H
	2020-05	117	Luxemburgo	28080	Majadahon
	2022-03	110	Francia	28903	Tres Cantos
A spec_tbl_df: 50294 CE 10	2019-11	343	Colombia	28130	San Fernan
	2022-10	000	Total	28162	Valdeolmo
	2022-10	010	Total Europa	28162	Valdeolmo
	2022-10	000	Total	28164	Valdetorre
	2022-10	011	Total Unión Europea	28164	Valdetorre
	2022-10	000	Total	28167	Velilla de S
	2022-10	000	Total	28168	Vellón, El
	2022-10	126	Alemania	28168	Vellón, El
	2022-10	000	Total	28169	Venturada
	2022-10	011	Total Unión Europea	28169	Venturada
	2022-10	000	Total	28171	Villa del Pr
	2022-10	011	Total Unión Europea	28171	Villa del Pr
	2022-10	121	Países Bajos	28171	Villa del Pr
	2022-10	010	Total Europa	28172	Villalbilla
	2022-10	000	Total	28174	Villamanta
	2022-10	011	Total Unión Europea	28174	Villamanta
	2022-10	011	Total Unión Europea	28176	Villanueva
	2022-10	020	Total África	28176	Villanueva
	2022-10	033	Total Sudamérica	28176	Villanueva
	2022-10	102	Austria	28176	Villanueva
	2022-10	115	Italia	28176	Villanueva

```
In [18]: dataj <- data |>
  full_join(datageo, by = c("CMUN" = "cmun"))
```

## 0.6 Data Structure

Estructura de los datos:

```
In [19]: dataj |> glimpse()
```

```
Rows: 50,301
Columns: 17
$ mes          <chr> "2019-08", "2021-07", "2021-07", "2022-01", "2019-08", "
$ pais_orig_cod <chr> "110", "010", "010", "000", "128", "000", "011", "126",
$ pais_orig     <chr> "Francia", "Total Europa", "Total Europa", "Total", "Rum
$ mun_dest_cod  <dbl> 28161, 28176, 28132, 28141, 28130, 28126, 28075, 28005,
$ mun_dest      <chr> "Valdemoro", "Villanueva de la Cañada", "San Martín de l
$ turistas      <dbl> 466, 1375, 465, 54, 135, 30, 285, 768, 31, 1646, 116, 36
$ CMUN          <chr> "161", "176", "132", "141", "130", "126", "075", "005",
$ gasto         <dbl> 76.360, 99.650, 99.650, 107.820, 109.210, 118.230, 118.2
$ Target        <dbl> 76.360, 99.650, 99.650, 107.820, 109.210, 118.230, 118.2
$ is_train      <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TR
$ codauto       <chr> "13", "13", "13", "13", "13", "13", "13", "13", "13", "1
$ ine.ccaa.name <chr> "Madrid, Comunidad de", "Madrid, Comunidad de", "Madrid,
$ cpro          <chr> "28", "28", "28", "28", "28", "28", "28", "28", "28", "2
$ ine.prov.name <chr> "Madrid", "Madrid", "Madrid", "Madrid", "Madrid", "Madri
$ name          <chr> "Valdemoro", "Villanueva de la Cañada", "San Martín de l
$ LAU_CODE      <chr> "28161", "28176", "28132", "28141", "28130", "28126", "2
$ geometry      <MULTIPOLYGON [ř]> MULTIPOLYGON (((-3.5914 40..., MULTIPOLYGO
```

Muestra de los primeros datos:

```
In [20]: dataj |> slice_head(n = 5)
```

	mes	pais_orig_cod	pais_orig	mun_dest_cod	mun_dest
	<chr>	<chr>	<chr>	<dbl>	<chr>
A spec_tbl_df: 5 of 17	2019-08	110	Francia	28161	Valdemoro
	2021-07	010	Total Europa	28176	Villanueva de la Cañada
	2021-07	010	Total Europa	28132	San Martín de la Vega
	2022-01	000	Total	28141	Sevilla la Nueva
	2019-08	128	Rumania	28130	San Fernando de Henares

## 0.7 Data Visualization

### 0.7.1 Map

Filtrar y agrupar los datos a mostrar en el mapa

```
In [21]: mdata <- dataj |>
  group_by(geometry) |>
  summarise(valor = sum(Target, na.rm = TRUE))
```

```

In [22]: pal <- colorNumeric(palette = "Blues",
                             domain = mdata$valor)

sf::st_as_sf(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(DESBDT, " (", GEOCODIGO, ")"),
              label = ~paste0(valor, " Turistas")) |>
  addLegend("bottomright",
            pal = pal,
            values = ~valor,
            title = "Número de vacunas",
            labFormat = labelFormat(big.mark = " "),
            opacity = 1
  )

```

HTML widgets cannot be represented in plain text (need html)

## 0.8 REPORT

A continuación se realizará un informe de las acciones realizadas

### 0.9 Main Actions Carried Out

- Se han realizado visualizaciones adicionales de los datos del caso de uso

### 0.10 Main Conclusions

- Los datos son adecuados para el caso de uso