# 05. - Data Collection_CU_45_06_turismo_origen_completo_v_01

June 11, 2023

#

CU45_Planificación y promoción del destino en base a los patrones en origen de los turistas

Citizenlab Data Science Methodology > II - Data Processing Domain *** > # 05.- Data Collection

Data Collection is the process to obtain and generate (if required) necessary data to model the problem.

### 0.0.1 06. Consolidación de los datos de turismo por origen

- Consolidar en un único archivo los datos obtenidos de turistas por origen (pais, provincia) y los datos de escucha simulados.
- No se incluyen los de turistas por municipio de origen y provincia de destino ya que solo analizamos una

Table of Contents

Settings

Data Load

ETL Processes

Import data from: CSV, Excel, Tab, JSON, SQL, and Parquet files

Synthetic Data Generation

Fake Data Generation

Open Data

Data Save

Main Conclusions

Main Actions

Acciones done

Acctions to perform

## 0.1 Settings

### 0.1.1 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")
```

```
Warning message in Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8"):
"OS reports request to set locale to "es_ES.UTF-8" cannot be honored"
```

”

### 0.1.2 Packages to use

*ELIMINAR O AÑADIR LO QUE TOQUE. COPIAR VERSIONES AL FINAL Y QUITAR CÓDIGO DE VERSIONES*

- {tcltk} para selección interactiva de archivos locales
- {sf} para trabajar con georeferenciación
- {readr} para leer y escribir archivos csv
- {dplyr} para explorar datos
- {stringr} para manipulación de cadenas de caracteres
- {tidyr} para organización de datos

[2]:
```
library(readr)
library(dplyr)

p <- c("tcltk",  "readr", "dplyr")
```

```
Attaching package: ‘dplyr’


The following objects are masked from ‘package:stats’:

    filter, lag


The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

### 0.1.3 Paths

[3]:
```
iPath <- "Data/Input/"
oPath <- "Data/Output/"
```

## 0.2 Data Load

If there are more than one input file, make as many sections as files to import.

Instrucciones - Los ficheros de entrada del proceso están siempre en Data/Input/.
- Si hay más de un fichero de entrada, se crean tantos objetos iFile_xx y file_data_xx como ficheros de entrada (xx número correlativo con dos dígitos, rellenar con ceros a la izquierda)

1. Datos de valoraciones

OPCION A: Seleccionar fichero en ventana para mayor comodidad

Data load using the {tcltk} package. Ucomment the line if not using this option

[6]:
```
# file_data_01 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[7]:
```
iFile_01 <- "CU_45_05_02_valoracion_sim.csv"
file_data_01 <- paste0(iPath, iFile_01)

if(file.exists(file_data_01)){
    cat("Se leerán datos del archivo: ", file_data_01)
} else{
    warning("Cuidado: el archivo no existe.")
}
```

```
Se leerán datos del archivo:  Data/Input/CU_45_05_02_valoracion_sim.csv
```

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

[8]:
```
data_01 <- read_csv(file_data_01)
```

```
Rows: 24780 Columns: 9
  Column specification



Delimiter: ","
chr (5): grupo, tipo, nombre, CMUN, CDIS
dbl (4): X, Y, t3_1, puntos

 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.
```

Estructura de los datos:

[9]:
```
data_01 |> glimpse()
```

```
Rows: 24,780
Columns: 9
$ grupo  <chr> "turismo", "hosteleria", "hosteleria",
"hosteleria", "comercio"…
$ tipo   <chr> "hotel", "restaurant", "pub", "pub",
"supermarket", "fast_food"…
$ nombre <chr> "NH Ciudad de la Imagen", "Café Comercial",
"Sidrería la Camoch…
$ X      <dbl> -3.788176, -3.702002, -3.701686, -3.696329,
-3.706888, -3.60722…
$ Y      <dbl> 40.39844, 40.42873, 40.42703, 40.42760,
40.48035, 40.43337, 40.…
$ CMUN   <chr> "115", "079", "079", "079", "079", "079",
"079", "079", "079", …
$ CDIS   <chr> "01", "01", "01", "01", "08", "20", "01",
"01", "01", "01", "01…
$ t3_1   <dbl> 41.46522, 43.76242, 43.76242, 43.76242,
43.76242, 43.76242, 43.…
$ puntos <dbl> 2, 4, 4, 1, 5, 2, 1, 3, 3, 5, 5, 5, 3, 4, 2,
4, 3, 1, 4, 2, 1, …
```

Muestra de los primeros datos:

[10]:
```
data_01 |> slice_head(n = 5)
```

A spec_tbl_df: 5 × 9

| grupo <chr> | tipo <chr> | nombre <chr> | X <dbl> | Y <dbl> | CMUN <chr> | CD <c |
|---|---|---|---|---|---|---|
| turismo | hotel | NH Ciudad de la Imagen | -3.788176 | 40.39844 | 115 | 01 |
| hosteleria | restaurant | Café Comercial | -3.702002 | 40.42873 | 079 | 01 |
| hosteleria | pub | Sidrería la Camocha | -3.701686 | 40.42703 | 079 | 01 |
| hosteleria | pub | Gran Cafe Santander | -3.696329 | 40.42760 | 079 | 01 |
| comercio | supermarket | Alcampo | -3.706888 | 40.48035 | 079 | 08 |

2. Datos de turismo receptor

OPCION A: Seleccionar fichero en ventana para mayor comodidad

Data load using the {tcltk} package. Ucomment the line if not using this option

[11]:
```
# file_data_02 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[12]:
```
iFile_02 <- "CU_45_05_03_receptor.csv"
file_data_02 <- paste0(iPath, iFile_02)

if(file.exists(file_data_02)){
    cat("Se leerán datos del archivo: ", file_data_02)
} else{
    warning("Cuidado: el archivo no existe.")
}
```

```
Se leerán datos del archivo:  Data/Input/CU_45_05_03_receptor.csv
```

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

[13]:
```
data_02 <- read_csv(file_data_02)
```

```
Rows: 50294 Columns: 7
  Column specification



Delimiter: ","
chr (5): mes, pais_orig_cod, pais_orig, mun_dest, CMUN
dbl (2): mun_dest_cod, turistas

 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.
```

Estructura de los datos:

[14]:
```
data_02 |> glimpse()
```

```
Rows: 50,294
Columns: 7
$ mes           <chr> "2019-07", "2019-07", "2019-07",
"2019-07", "2019-07", "…
$ pais_orig_cod <chr> "000", "010", "011", "030", "110",
"121", "123", "126", …
$ pais_orig     <chr> "Total", "Total Europa", "Total Unión
Europea", "Total A…
$ mun_dest_cod  <dbl> 28002, 28002, 28002, 28002, 28002,
28002, 28002, 28002, …
$ mun_dest      <chr> "Ajalvir", "Ajalvir", "Ajalvir",
"Ajalvir", "Ajalvir", "…
$ turistas      <dbl> 338, 290, 268, 37, 56, 54, 37, 40, 157,
116, 109, 8461, …
```

```
$ CMUN          <chr> "002", "002", "002", "002", "002",
"002", "002", "002", …
```

Muestra de los primeros datos:

[15]:
```
data_02 |> slice_head(n = 5)
```

A spec_tbl_df: 5 × 7

| mes <chr> | pais_orig_cod <chr> | pais_orig <chr> | mun_dest_cod <dbl> | mun_dest <chr> | turistas <dbl> |
|---|---|---|---|---|---|
| 2019-07 | 000 | Total | 28002 | Ajalvir | 338 |
| 2019-07 | 010 | Total Europa | 28002 | Ajalvir | 290 |
| 2019-07 | 011 | Total Unión Europea | 28002 | Ajalvir | 268 |
| 2019-07 | 030 | Total América | 28002 | Ajalvir | 37 |
| 2019-07 | 110 | Francia | 28002 | Ajalvir | 56 |

3. Datos de turismo nacional por provincia de origen

OPCION A: Seleccionar fichero en ventana para mayor comodidad

Data load using the {tcltk} package. Ucomment the line if not using this option

[16]:
```
# file_data_03 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[17]:
```
iFile_03 <- "CU_45_05_04_interno_prov.csv"
file_data_03 <- paste0(iPath, iFile_03)

if(file.exists(file_data_03)){
    cat("Se leerán datos del archivo: ", file_data_03)
} else{
    warning("Cuidado: el archivo no existe.")
}
```

```
Se leerán datos del archivo:  Data/Input/CU_45_05_04_interno_prov.csv
```

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

[18]:
```
data_03 <- read_csv(file_data_03)
```

```
Rows: 521280 Columns: 8
  Column specification



Delimiter: ","
chr (6): total_nacional, total_ccaa, provincia, municipio_destino,
cmun, mes
dbl (2): turistas, secreto
```

```
  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.
```

Estructura de los datos:

[28]:
```
data_03 |> glimpse()
```

```
Rows: 521,280
Columns: 8
$ total_nacional    <chr> "Total Nacional", "Total Nacional",
"Total Nacional"…
$ total_ccaa        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, …
$ provincia         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, …
$ municipio_destino <chr> "Acebeda, La", "Acebeda, La",
"Acebeda, La", "Acebed…
$ turistas          <dbl> NA, NA, NA, 39, 125, 31, NA, NA,
NA, NA, NA, NA, NA,…
$ cmun              <chr> "001", "001", "001", "001", "001",
"001", "001", "00…
$ mes               <chr> "2022-10", "2022-09", "2022-08",
"2022-07", "2022-06…
$ secreto           <dbl> NA, NA, NA, 0, 0, 0, NA, NA, NA, 1,
NA, NA, NA, NA, …
```

Muestra de los primeros datos:

[29]:
```
data_03 |> slice_head(n = 5)
```

A spec_tbl_df: 5 × 8

| total_nacional <chr> | total_ccaa <chr> | provincia <chr> | municipio_destino <chr> | turistas <dbl> | cmun <chr> | mes <chr> |
|---|---|---|---|---|---|---|
| Total Nacional | NA | NA | Acebeda, La | NA | 001 | 2022-10 |
| Total Nacional | NA | NA | Acebeda, La | NA | 001 | 2022-09 |
| Total Nacional | NA | NA | Acebeda, La | NA | 001 | 2022-08 |
| Total Nacional | NA | NA | Acebeda, La | 39 | 001 | 2022-07 |
| Total Nacional | NA | NA | Acebeda, La | 125 | 001 | 2022-06 |

## 0.3 ETL Processes

### 0.3.1 Import data from: CSV, Excel, Tab, JSON, SQL, and Parquet files

Se han importado en el apartado Data Load anterior:

- Datos de puntuaciones por POI
- Datos de turismo de entrada por país
- Datos de turismo de entrada por provincia origen
- Datos de turismo de entrada por municipio origen

Incluir apartados si procede para: Extracción de datos (select, filter), Transformación de datos, (mutate, joins, …). Si es necesario tratar datos perdidos, indicarlo también en NB 09.2

Si no aplica: Estos datos no requieren tareas de este tipo.

**Unión de las tablas**

[ ]:
```
data <- data_01 |>
    full_join(data_02, by = c("CMUN" = "CMUN")) |>
    full_join(data_03, by = c("CMUN" = "cmun",
                              "mes" = "mes"))
nrow(data)
```

```
Warning message in full_join(data_01, data_02, by = c(CMUN = "CMUN")):
"Detected an unexpected many-to-many relationship between `x` and `y`.
 Row 1 of `x` matches multiple rows in `y`.
 Row 653 of `y` matches multiple rows in `x`.
 If a many-to-many relationship is expected, set `relationship =
"many-to-many"` to silence this warning."
```

## 0.4 Synthetic Data Generation

No aplica

## 0.5 Fake Data Generation

No aplica

## 0.6 Open Data

Los datos se han obtenido de fuentes públicas abiertas

## 0.7 Data Save

No procede porque el conjunto de datos consolidado no se ha podido generar por problemas de tamaño. Se usarán los archivos por separado siempre

## 0.8 Main Conclusions

List and describe the general conclusions of the analysis carried out.

### 0.8.1 Configuration Management

This notebook has been tested with the following versions of R and packages. It cannot be assured that later versions work in the same way: * R 4.2.2 * tcltk 4.2.3 * readr 2.1.3 * dplyr 1.1.0

**Observaciones generales sobre los datos**

- El tamaño es demasiado grande para consolidar en un único archivo.
- Se usarán siempre los ficheros por separado

### 0.8.2 Consideraciones para despliegue en piloto

- No aplica

### 0.8.3 Consideraciones para despliegue en producción

- No aplica

## 0.9 Main Actions

**Acciones done** Indicate the actions that have been carried out in this process

- Se ha intentado consolidar los datos en un único archivo

**Acctions to perform** Indicate the actions that must be carried out in subsequent processes

- Se deben hacer las uniones como corresponda en la modelización y despliegue

## 0.10 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

[ ]:
```
# incluir código
```