05. - Data Collection_CU_18_12_pois_distrito_v_01

June 13, 2023

#

CU18_Infraestructuras_eventos

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 04. Asignar distrito censal a POIs

- Dadas las coordenadas de cada POI, se asigna el distrito censal al que pertenece.

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 Packages to use

- {tcltk} para selección interactiva de archivos locales
- {sf} para trabajar con georeferenciación
- {readr} para leer y escribir archivos csv
- {dplyr} para explorar datos

[3]:
```
library(sf)
library(readr)
library(dplyr)
```

```
Linking to GEOS 3.10.2, GDAL 3.4.2, PROJ 8.2.1; sf_use_s2() is TRUE


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
```

0.1.2 Paths

[4]:
```
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. Archivo de POIs OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

[5]:
```
# file_data_01 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[6]:
```
iFile_01 <- "CU_18_05_11_pois_csv.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 leer<U+00E1>n datos del archivo:  Data/Input/CU_18_05_11_pois_csv.csv
```

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

[7]:
```
data_01 <- read_csv(file_data_01)
```

```
Rows: 24780 Columns: 5
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): grupo, tipo, nombre
dbl (2): X, Y

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.
```

Estructura de los datos:

[8]:
```
glimpse(data_01)
```

```
Rows: 24,780
Columns: 5
$ grupo  <chr> "turismo", "hosteleria", "hosteleria",
"hosteleria", "comercio"~
$ tipo   <chr> "hotel", "restaurant", "pub", "pub",
"supermarket", "fast_food"~
$ nombre <chr> "NH Ciudad de la Imagen", "Caf<U+00E9>
Comercial", "Sidrer<U+00ED>a 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.~
```

Muestra de datos:

[9]:
```
slice_head(data_01, n = 5)
```

A spec_tbl_df: 5 x 5

grupo <chr>	tipo <chr>	nombre <chr>	X <dbl>	Y <dbl>
turismo	hotel	NH Ciudad de la Imagen	-3.788176	40.39844
hosteleria	restaurant	Caf<U+00E9> Comercial	-3.702002	40.42873
hosteleria	pub	Sidrer<U+00ED>a la Camocha	-3.701686	40.42703
hosteleria	pub	Gran Cafe Santander	-3.696329	40.42760
comercio	supermarket	Alcampo	-3.706888	40.48035

2. Archivo de geolocalización distritos censales OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

[10]:
```
# file_data_02 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[11]:
```
iFile_02 <- "CU_18_05_03_distritos_geo.json"
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 leer<U+00E1>n datos del archivo:  Data/Input/CU_18_05_03_distritos_geo.json
```

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

[12]:
```
data_02 <- st_read(file_data_02)
```

```
Reading layer `CU_18_05_03_distritos_geo' from data source
  `/Users/emilio.lcano/academico/gh_repos/__transferencia/citizenlab/CitizenLab-
Research-and-Development/casos_urjc/notebooks/II_data_processing/18_infraestruct
uras/Data/Input/CU_18_05_03_distritos_geo.json'
  using driver `GeoJSON'
Simple feature collection with 247 features and 2 fields
Geometry type: GEOMETRY
Dimension:     XY
Bounding box:  xmin: -4.579006 ymin: 39.8848 xmax: -3.052983 ymax: 41.16584
Geodetic CRS:  WGS 84
```

Estructura de los datos:

[13]:
```
glimpse(data_02)
```

```
Rows: 247
Columns: 3
$ CMUN     <chr> "001", "002", "003", "004", "005", "005",
"005", "005", "005"~
$ CDIS     <chr> "01", "01", "01", "01", "01", "02", "03",
"04", "05", "01", "~
$ geometry <POLYGON [arc_degree]> POLYGON ((-3.64502
41.12129…, POLYGON ((-3~
```

Muestra de datos:

[14]:
```
data_02 |> tibble() |> slice_head(n = 5)
```

A tibble: 5 x 3

CMUN <chr>	CDIS <chr>	geometry <POLYGON [arc_degree]>
001	01	POLYGON ((-3.64502 41.12129...
002	01	POLYGON ((-3.503032 40.526,...
003	01	POLYGON ((-3.808664 40.8921...
004	01	POLYGON ((-4.00197 40.25642...
005	01	POLYGON ((-3.361691 40.4762...

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:

- Archivo de POIs
- Archivo de georeferenciación distritos censales

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

Data Transform Convertir coordenadas numéricas a objeto espacial

[15]:
```
tdata_01 <- data_01 |>
    st_as_sf(coords = c("X", "Y"), crs = 4326,
           remove = FALSE)
```

[16]:
```
tdata_01 |> tibble() |> slice_head(n = 5)
```

A tibble: 5 x 6

grupo <chr>	tipo <chr>	nombre <chr>	X <dbl>	Y <dbl>	geometry <POINT [ar
turismo	hotel	NH Ciudad de la Imagen	-3.788176	40.39844	POINT (-3.7
hosteleria	restaurant	Caf<U+00E9> Comercial	-3.702002	40.42873	POINT (-3.7
hosteleria	pub	Sidrer<U+00ED>a la Camocha	-3.701686	40.42703	POINT (-3.7
hosteleria	pub	Gran Cafe Santander	-3.696329	40.42760	POINT (-3.6
comercio	supermarket	Alcampo	-3.706888	40.48035	POINT (-3.7

- Corregir geometrías erróneas de distritos

[17]:
```
tdata_02 <- data_02 |> st_make_valid()
```

- Unir data frames de POIs y distritos por intersección de coordenadas

[18]:
```
data <- tdata_01 |>
  st_join(tdata_02) |>
  st_drop_geometry()
```

[19]:
```
data |> slice_head(n = 5)
```

	grupo	tipo	nombre	X	Y	CMUN	CD
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<chr>	<c
	turismo	hotel	NH Ciudad de la Imagen	-3.788176	40.39844	115	01
A tibble: 5 x 7	hosteleria	restaurant	Caf<U+00E9> Comercial	-3.702002	40.42873	079	01
	hosteleria	pub	Sidrer<U+00ED>a 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

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

0.4 Synthetic Data Generation

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

0.5 Fake Data Generation

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

0.6 Open Data

Los datos provienen de fuentes abiertas obtenidas en taras anteriores.

0.7 Data Save

Este proceso, puede copiarse y repetirse en aquellas partes del notebbok que necesiten guardar datos. Recuerde cambiar las cadenas añadida del fichero para diferenciarlas

Identificamos los datos a guardar

[20]:
```
data_to_save <- data
```

Estructura de nombre de archivos:

- Código del caso de uso, por ejemplo “CU_04”
- Número del proceso que lo genera, por ejemplo ”_05”.
- Número de la tarea que lo genera, por ejemplo ”_01”
- En caso de generarse varios ficheros en la misma tarea, llevarán _01 _02 … después
- Nombre: identificativo de “properData”, por ejemplo ”_zonasgeo”
- Extensión del archivo

Ejemplo: ”CU_04_05_01_01_zonasgeo.json, primer fichero que se genera en la tarea 01 del proceso 05 (Data Collection) para el caso de uso 04 (vacunas)

Importante mantener los guiones bajos antes de proceso, tarea, archivo y nombre

0.7.1 Proceso 05

[21]:
```
caso <- "CU_18"
proceso <- '_05'
tarea <- "_12"
archivo <- ""
proper <- "_pois_distrito"
```

```
extension <- ".csv"
```

OPCION A: Uso del paquete “tcltk” para mayor comodidad

- Buscar carpeta, escribir nombre de archivo SIN extensión (se especifica en el código)
- Especificar sufijo2 si es necesario
- Cambiar datos por datos_xx si es necesario

[22]:
```
# file_save <- paste0(caso, proceso, tarea, tcltk::tkgetSaveFile(), proper,␣
↪extension)
# path_out <- paste0(oPath, file_save)
# write_csv(data_to_save, path_out)

# cat('File saved as: ')
# path_out
```

OPCION B: Especificar el nombre de archivo

- Los ficheros de salida del proceso van siempre a Data/Output/.

[23]:
```
file_save <- paste0(caso, proceso, tarea, archivo, proper, extension)
path_out <- paste0(oPath, file_save)
write_csv(data_to_save, path_out)

cat('File saved as: ')
path_out
```

```
File saved as:
```

’Data/Output/CU_18_05_12_pois_distrito.csv’

Copia del fichero a Input Si el archivo se va a usar en otros notebooks, copiar a la carpeta Input

[24]:
```
path_in <- paste0(iPath, file_save)
file.copy(path_out, path_in, overwrite = TRUE)
```

TRUE

0.8 Main Conclusions

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

0.8.1 Prerequisites

This working code needs the following conditions:

- For using the interactive selection of file, the {tcltk} package must be installed. It is not needed in production.
- The {readr}, {dplyr} and {sf} packages must be installed.
- The data paths `Data/Input` and `Data/Output` must exist (relative to the notebook path)

0.8.2 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 4.2.2 * dplyr 1.0.10 * sf 1.0.9

0.8.3 Data structures

Objeto `data`

- Los datos de origen contienen datos de infraesctructuras
- Hay 1633 filas con las variables:
  - grupo
  - tipo
  - nombre
  - X
  - Y
  - CMUN
  - CDIS

[25]:
```
glimpse(data)
```

```
Rows: 24,780
Columns: 7
$ grupo  <chr> "turismo", "hosteleria", "hosteleria",
"hosteleria", "comercio"~
$ tipo   <chr> "hotel", "restaurant", "pub", "pub",
"supermarket", "fast_food"~
$ nombre <chr> "NH Ciudad de la Imagen", "Caf<U+00E9>
Comercial", "Sidrer<U+00ED>a 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~
```

Observaciones generales sobre los datos

- Hay algunas discrepancias entre los códigos de municipios de las infraestructuras y los de los distritos censales que habría que revisar
- En lo sucesivo se tomarán como buenos los códigos del INE al ser la fuente primaria

0.8.4 Consideraciones para despliegue en piloto

- xxx

0.8.5 Consideraciones para despliegue en producción

- Se deben crear los procesos ETL en producción necesarios para que los datos de entrada estén actualizados

0.9 Main Actions

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

- Se han transformado las coordenadas numéricas de infraestructuras a objeto espacial 'sf"
- Se han corregido las geometrías erróneas de distritos
- Se ha obtenido el distrito al que pertenece la infraestructura por intersección espacial

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

- Se debe revisar la codificación de municipios de la Cumunidad de Madrid

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

[26]:
```
# incluir código
```