05. - Data Collection_CU_18_11_pois_csv_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 11. Obtener fichero csv con POIs de OSM

- Los datos de origen se descargaron en la tarea ETL de turismo de Trello.
- Categorías hostelería, turismo y comercio

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
- {readr} para leer y escribir ficheros csv
- {dplyr} para explorar y transformar datos

- {stringr} para manipular cadenas de caracteres

[1]:
```
library(readr)
library(sf)
library(dplyr)
library(stringr)
```

```
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

[2]:
```
iPath <- "Data/Input/"
oPath <- "Data/Output/"
```

0.2 Data Load

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)

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

[3]:
```
# file_data <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[4]:
```
iFile <- "madrid/gis_osm_pois_free_1.shp"
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 leer<U+00E1>n datos del archivo:  Data/Input/madrid/gis_osm_pois_free_1.shp
```

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

[5]:
```
data <- read_sf(file_data)
```

Estructura de los datos:

[6]:
```
data |>  glimpse()
```

```
Rows: 54,732
Columns: 5
$ osm_id   <chr> "25913327", "25931627", "25931630",
"26065697", "26065699", "~
$ code     <int> 2401, 2203, 2257, 2301, 2304, 2304, 2201,
2001, 2501, 2203, 2~
$ fclass   <chr> "hotel", "cinema", "ice_rink", "restaurant",
"pub", "pub", "t~
$ name     <chr> "NH Ciudad de la Imagen", "mk2 Palacio de
Hielo", NA, "Caf<U+00E9> C~
$ geometry <POINT [arc_degree]> POINT (-3.788176 40.39844),
POINT (-3.636102 4~
```

Muestra de datos:

[7]:
```
data |> tibble() |> slice_head(n = 5)
```

A tibble: 5 x 5

osm_id <chr>	code <int>	fclass <chr>	name <chr>	geometry <POINT [arc_degree]>
25913327	2401	hotel	NH Ciudad de la Imagen	POINT (-3.788176 40.39844)
25931627	2203	cinema	mk2 Palacio de Hielo	POINT (-3.636102 40.46314)
25931630	2257	ice_rink	NA	POINT (-3.637657 40.46304)
26065697	2301	restaurant	Caf<U+00E9> Comercial	POINT (-3.702002 40.42873)
26065699	2304	pub	Sidrer<U+00ED>a la Camocha	POINT (-3.701686 40.42703)

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:

- POIs de la comunidad de Madrid hotelería, turismo y comercio

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 extract

- Filtrar solo categorías 23xx, 24xx y 25xx

- Quitar id

[36]:
```
edata <- data |>
    filter(str_sub(code, 1, 2) %in% c("23", "24", "25")) |>
    select(-osm_id)
```

Data Transformation

- Guardar coordenadas como columnas X e Y
- Estandarizar nombres grupo, tipo, nombre
- Eliminar geometría sf

[39]:
```
tdata <- edata |>
    mutate(grupo = str_sub(code, 1, 2), , .after = 1) |>
    mutate(grupo = case_when(
        grupo == "23" ~ "hosteleria",
        grupo == "24" ~ "turismo",
        grupo == "25" ~ "comercio",
        TRUE ~ "desconocido")) |>
    rename(tipo = fclass,
    nombre = name) |>
    bind_cols(st_coordinates(edata)) |>
    select(-code) |>
    st_drop_geometry()
```

[40]:
```
glimpse(tdata)
```

```
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.~
```

[41]:
```
tdata |> slice_head(n = 5)
```

A tibble: 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

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

0.4 Synthetic Data Generation

Estos datos no requieren tareas de este tipo.

0.5 Fake Data Generation

Estos datos no requieren tareas de este tipo.

0.6 Open Data

Los datos de POIs se han descargado de la fuente abierta OpenStreetMap

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

[47]:
```
data_to_save <- tdata
```

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

[45]:
```
caso <- "CU_18"
proceso <- '_05'
tarea <- "_11"
archivo <- ""
proper <- "_pois_csv"
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

[ ]:
```
# 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/.

[48]:
```
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_11_pois_csv.csv'

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

[49]:
```
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 {dplyr}, {sf}, {readr}, {stringr} 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 * xxx x.x.x

0.8.3 Data structures

Objeto `tdata`

- Los datos de origen son archivos de formas
- Hay 24780 filas con las variables:
  - grupo
  - tipo
  - nombre
  - coordenadas

Observaciones generales sobre los datos

- Ninguna

0.8.4 Consideraciones para despliegue en piloto

- Ninguna

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 convertido las coordenadas espaciales a columnas numéricas
- Se han homogeneizado las categoraizaciones para que coincidan con las de infraestructuras y administración

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

- Se deben asignar distritos censales y contar puntos de cada tipo en cada distrito

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
```