05. - Data Collection_CU_18_07_admon_comercio_actividad_educacion_v_01

June 13, 2023

#

CUxx_xxxxx

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 07. Crear csv con datos de administración y empresas

- Crear archivo csv con datos de stablecimientos de administración, colectivo empresarial, comercio y educación, incluyendo coordenadas, tipo y distrito censal
- Consolidar los datos en formato shp en data.frames con coordenadas
- Total 12 ficheros de entrada y uno de salida

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

- {sf} para trabajar con georeferenciación
- {readr} para leer y escribir archivos csv
- {dplyr} para explorar y manipular datos
- {dplyr} para organizar datos

[1]:
```
library(sf)
library(readr)
library(dplyr)
library(tidyr)
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/admon_comercio_actividad_educacion/"
oPath <- "Data/Output/"
```

0.2 Data Load

- En esta tarea, se realizan a la vez los procesos Data Load y Data Transform por conveniencia al ejecutar en un bucle sobre las carpetas que contienen las infraestructuras. Por tanto no está la opción de cargar ficheros locales, ver apartado ETL.
- Los datos de entrada descargados (zip descomprimidos) se han copiado a la carpeta `admon_comercio_actividad_educacion/` dentro de Data/Input/

OPCION B: Especificar el nombre de archivo

No aplica

Data file to dataframe No aplica

0.3 ETL Processes

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

- La carga y transformación se hacen a la vez

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

Obtener lista de carpetas con shp

[3]:
```
carpetas <- list.dirs(iPath, recursive = FALSE)
```

[4]:
```
carpetas
```

1. 'Data/Input/admon_comercio_actividad_educacion//Administración pública_ Agencia Tributaria' 2. 'Data/Input/admon_comercio_actividad_educacion//Administración pública_ Ayuntamientos, Consejerías, Ministerios, etc.' 3. 'Data/Input/admon_comercio_actividad_educacion//Administración pública_ Embajadas y consulados' 4. 'Data/Input/admon_comercio_actividad_educacion//Administración pública_ Oficinas de empleo' 5. 'Data/Input/admon_comercio_actividad_educacion//Administración pública_ Seguridad Social' 6. 'Data/Input/admon_comercio_actividad_educacion//Administración Pública. Administración de Justicia' 7. 'Data/Input/admon_comercio_actividad_educacion//Colectivo empresarial por tamaño y actividad' 8. 'Data/Input/admon_comercio_actividad_educacion//Comercio_ Centros comerciales' 9. 'Data/Input/admon_comercio_actividad_educacion//Comercio_ Galerías de alimentación' 10. 'Data/Input/admon_comercio_actividad_educacion//Comercio_ Grandes superficies especializadas' 11. 'Data/Input/admon_comercio_actividad_educacion//Comercio_ Hipermercados' 12. 'Data/Input/admon_comercio_actividad_educacion//Comercio_ Mercadillos' 13. 'Data/Input/admon_comercio_actividad_educacion//Comercio_ Mercados de abastos' 14. 'Data/Input/admon_comercio_actividad_educacion//Comercio_ Otros servicios. Bancos' 15. 'Data/Input/admon_comercio_actividad_educacion//Educación_ Campus universitarios' 16. 'Data/Input/admon_comercio_actividad_educacion//Educación_ Centros educativos no universitarios. Centros privados' 17. 'Data/Input/admon_comercio_actividad_educacion//Educación_ Centros educativos no universitarios. Centros públicos' 18. 'Data/Input/admon_comercio_actividad_educacion//Centros educativos no universitarios. Servicios educativos' 19. 'Data/Input/admon_comercio_actividad_educacion//Educación_ Centros educativos universitarios' 20. 'Data/Input/admon_comercio_actividad_educacion//Educación_ Colegios mayores'

Obtener grupos, limpiando los nombres

[5]:
```
grupos <- carpetas |>
  str_replace_all("\\. ", "\\_ ") |>
  str_replace_all("Colectivo", "Empresas\\_ Colectivo") |>
  str_replace_all("Administración Pública", "Administración pública") |>
  str_split("//", simplify = TRUE) |>
  as.data.frame() |>
  select(2) |>
  separate(V2, c("Grupo", "Tipo"), sep = "\\_ ", extra = "merge") |>
```

```
  mutate(Tipo = str_replace_all(Tipo, "\\_ ", " - "))
```

[6]:
```
grupos
```

A data.frame: 20 × 2

Grupo <chr>	Tipo <chr>
Administración pública	Agencia Tributaria
Administración pública	Ayuntamientos, Consejerías, Ministerios, etc.
Administración pública	Embajadas y consulados
Administración pública	Oficinas de empleo
Administración pública	Seguridad Social
Administración pública	Administración de Justicia
Empresas	Colectivo empresarial por tamaño y actividad
Comercio	Centros comerciales
Comercio	Galerías de alimentación
Comercio	Grandes superficies especializadas
Comercio	Hipermercados
Comercio	Mercadillos
Comercio	Mercados de abastos
Comercio	Otros servicios - Bancos
Educación	Campus universitarios
Educación	Centros educativos no universitarios - Centros privados
Educación	Centros educativos no universitarios - Centros públicos
Educación	Centros educativos no universitarios - Servicios educativos
Educación	Centros educativos universitarios
Educación	Colegios mayores

Crear lista de data.frames importando los shp

Se realizan las siguientes transformaciones en cada uno:

- Transformación de crs a 4326
- Asignación de grupo y tipo
- Asignación de ETIQUETA según el contenido
- Hacer válidas las geometrías y calcular centroide (campus universitarios)
- Extraer solo Tipo, Grupo, Etiqueta y geometría

[7]:
```
lpuntos <- purrr::map(seq_along(carpetas), ~{
  df <- st_read(carpetas[.x],
                        options = "ENCODING=latin1",
                quiet = TRUE) |>
      st_transform(4326) |>
    mutate(Grupo = grupos$Grupo[.x],
           Tipo = grupos$Tipo[.x])
  if(.x == 1){
    df <- df |> mutate(ETIQUETA = BUSQUEDA)
  }
  if(.x == 5){
    df <- df |> mutate(ETIQUETA = BUSCA)
```

```
  }
  if(.x %in% c(6, 9, 12, 15)){
    df <- df |> mutate(ETIQUETA = NOMBRE)
  }
  if(.x == 7){
    df <- df |> mutate(Grupo = Tipo,
                       Tipo = DSCODR28)
  }
  if(.x == 19){
    df <- df |> mutate(ETIQUETA = paste0(ETIQUETA, " - ", UNIVERSIDA))
  }
  if(.x == 15){
    df <- df |> st_make_valid() |> st_centroid()
  }
  return(df |> select(Grupo, Tipo, ETIQUETA))
    }
)
```

```
Warning message in st_centroid.sf(st_make_valid(df)):
"st_centroid assumes attributes are constant over geometries of x"
```

Apilar data frames

[19]:
```
data <- bind_rows(lpuntos)
```

Añadir coordendas como columnas numéricas y eliminar columna sf

[20]:
```
tdata <- data |>
  bind_cols(st_coordinates(data)) |>
  st_drop_geometry()
```

Eliminar dos puntos sin cooredenadas

[21]:
```
tdata |> filter(is.na(X))
```

A data.frame: 2×5

Grupo	Tipo	ETIQUETA
<chr>	<chr>	<chr>
Educación	Campus universitarios	Campus UPCO (Madrid) Universidad Pontificia de Co
Educación	Campus universitarios	Campus UEM (Alcobendas)

[22]:
```
tdata <- tdata |>
  drop_na(X)
```

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

Estructura de datos

[28]:
```
glimpse(tdata)
```

```
Rows: 120,280
Columns: 5
$ Grupo    <chr> "Administración pública", "Administración
pública", "Administ…
$ Tipo     <chr> "Agencia Tributaria", "Agencia Tributaria",
"Agencia Tributar…
$ ETIQUETA <chr> "Administración de la Agencia Tributaria
Oficina Central", "A…
$ X        <dbl> -3.698678, -3.712513, -3.745598, -3.690993,
-3.672477, -3.710…
$ Y        <dbl> 40.45554, 40.44553, 40.37049, 40.41795,
40.45483, 40.48316, 4…
```

Muestra primeros datos

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

A data.frame: 5 × 5

Grupo <chr>	Tipo <chr>	ETIQUETA <chr>
Administración pública	Agencia Tributaria	Administración de la Agencia Tributaria Ofi
Administración pública	Agencia Tributaria	Administración de la Agencia Tributaria Guz
Administración pública	Agencia Tributaria	Administración de la Agencia Tributaria Sur
Administración pública	Agencia Tributaria	Administración de la Agencia Tributaria Mo
Administración pública	Agencia Tributaria	Administración de la Agencia Tributaria Ciu

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 originales fueron descargados de fuentes públicas

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

[23]:
```
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

[24]:
```
caso <- "CU_18"
proceso <- '_05'
tarea <- "_07"
archivo <- ""
proper <- "_admon_comercio_actividad_educacion"
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, 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/.

[25]:
```
file_save <- paste0(caso, proceso, tarea, archivo, proper, extension)
path_out <- paste0(oPath, file_save)
write_csv(tdata, path_out)

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

```
File saved as:
```

’Data/Output/CU_18_05_07_admon_comercio_actividad_educacion.csv’

Copia del fichero a Input Si el archivo se va a usar en otros notebooks, copiar a la carpeta Input Se pone específicamente porque los ficheros de Input estaban en una carpeta dentro de Input

[26]:
```
path_in <- paste0("Data/Input/", 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}, {sf}, {dplyr}, {tidyr}, {stringr} package 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 * tidyr 1.3.0 * dplyr 1.0.10 * readr 2.1.3 * stringr 1.5.0

0.8.3 Data structures

Objeto `data`

- Los datos de origen son archivos shp con geometrías y otros datos
- Hay 120.280 filas con las variables:
  - Grupo
  - Tipo
  - ETIQUETA
  - X
  - Y

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 obtenido las coordenadas del centroide de POIs que eran polígonos
- Se han homogeneizado las etiquetas y clasificación de los POIs

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

- Indicar en procesos de limpieza de datos que se han eliminado NA

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
```