05. - Data Collection_CU_18_13_distrito_pois_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 05. Agrupar datos de POIs por distrito

- A partir de los datos puntuales, agregar por distrito censal y contar POIs para hacer mapas de regiones y usar en modelos
- Adicionalmente crear metadatos con descripción y agrupamiento de variables.

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 archivos csv

- {dplyr} para explorar datos
- {dityr} para tranformar datos
- {janitor} para limpiar datos

[1]:
```
library(readr)
library(dplyr)
library(tidyr)
library(janitor)
```

```
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



Attaching package: 'janitor'


The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

0.1.2 Paths

[2]:
```
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)

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

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

OPCION B: Especificar el nombre de archivo

[3]:
```
iFile <- "CU_18_05_12_pois_distrito.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 leer<U+00E1>n datos del archivo:  Data/Input/CU_18_05_12_pois_distrito.csv
```

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

[4]:
```
data <- read_csv(file_data)
```

```
Rows: 24780 Columns: 7
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (5): grupo, tipo, nombre, CMUN, CDIS
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:

[5]:
```
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~
```

Muestra de datos:

[6]:
```
slice_head(data, n = 5)
```

A spec_tbl_df: 5 x 7

grupo <chr>	tipo <chr>	nombre <chr>	X <dbl>	Y <dbl>	CMU <chr>
turismo	hotel	NH Ciudad de la Imagen	-3.788176	40.39844	115
hosteleria	restaurant	Caf<U+00E9> Comercial	-3.702002	40.42873	079
hosteleria	pub	Sidrer<U+00ED>a la Camocha	-3.701686	40.42703	079
hosteleria	pub	Gran Cafe Santander	-3.696329	40.42760	079
comercio	supermarket	Alcampo	-3.706888	40.48035	079

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

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

- Contar número de infraestructuras de cada tipo por distrito
- Extender en columnas para caracterizar distritos

[7]:
```
tdata_01 <- data |>
  count(CMUN, CDIS, tipo) |>
  pivot_wider(names_from = tipo,
              values_from = n,
              values_fill = 0)
```

[8]:
```
glimpse(tdata_01)
```

```
Rows: 224
Columns: 60
$ CMUN              <chr> "002", "003", "004", "005", "005",
"005", "005", "00~
$ CDIS              <chr> "01", "01", "01", "01", "02", "03",
"04", "05", "01"~
$ bakery            <int> 2, 0, 0, 4, 5, 1, 2, 7, 4, 1, 1, 0,
2, 0, 0, 1, 0, 0~
$ bar               <int> 2, 0, 1, 22, 10, 2, 3, 11, 7, 1, 1,
2, 6, 0, 2, 17, ~
$ butcher           <int> 1, 0, 0, 2, 2, 0, 1, 2, 0, 0, 1, 0,
0, 0, 0, 2, 0, 0~
```

```
$ cafe              <int> 5, 1, 0, 12, 6, 1, 3, 19, 12, 3, 2,
0, 6, 0, 1, 4, 0~
$ car_dealership    <int> 1, 0, 0, 1, 1, 3, 0, 3, 5, 1, 0, 0,
0, 0, 0, 0, 0, 0~
$ clothes           <int> 1, 0, 0, 13, 1, 0, 1, 28, 0, 0, 0,
0, 0, 0, 1, 1, 0,~
$ convenience       <int> 2, 0, 0, 4, 2, 2, 2, 13, 8, 1, 1,
2, 6, 0, 0, 1, 0, ~
$ fast_food         <int> 1, 0, 0, 8, 1, 2, 5, 11, 10, 7, 4,
0, 6, 0, 2, 5, 0,~
$ food_court        <int> 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ furniture_shop    <int> 2, 0, 0, 3, 0, 0, 1, 8, 2, 1, 0, 0,
0, 0, 0, 0, 0, 0~
$ gift_shop         <int> 1, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0~
$ greengrocer       <int> 1, 0, 0, 1, 1, 0, 0, 8, 2, 0, 1, 0,
1, 0, 0, 0, 0, 0~
$ guesthouse        <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0~
$ hairdresser       <int> 2, 0, 0, 16, 4, 0, 4, 21, 7, 1, 0,
1, 5, 0, 0, 1, 0,~
$ hotel             <int> 1, 1, 0, 3, 0, 3, 0, 1, 3, 3, 0, 0,
0, 0, 2, 1, 0, 1~
$ pub               <int> 2, 0, 0, 24, 4, 2, 2, 31, 7, 0, 0,
0, 4, 0, 0, 2, 1,~
$ restaurant        <int> 5, 4, 3, 53, 5, 18, 10, 22, 73, 26,
13, 1, 4, 2, 2, ~
$ stationery        <int> 1, 0, 0, 1, 3, 0, 1, 2, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1~
$ vending_any       <int> 1, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ bicycle_rental    <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0~
$ hostel            <int> 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0~
$ supermarket       <int> 0, 1, 1, 6, 6, 2, 3, 10, 8, 12, 3,
4, 3, 1, 3, 3, 0,~
$ beauty_shop       <int> 0, 0, 0, 3, 0, 0, 2, 10, 6, 0, 0,
0, 0, 0, 1, 0, 0, ~
$ beverages         <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ bicycle_shop      <int> 0, 0, 0, 3, 1, 1, 0, 2, 4, 0, 1, 1,
1, 0, 0, 2, 0, 0~
$ bookshop          <int> 0, 0, 0, 7, 0, 0, 2, 3, 3, 1, 0, 0,
0, 0, 0, 0, 0, 1~
$ chemist           <int> 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0~
```

```
$ computer_shop     <int> 0, 0, 0, 3, 3, 0, 0, 2, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0~
$ doityourself      <int> 0, 0, 0, 2, 2, 2, 0, 3, 3, 0, 0, 0,
0, 0, 0, 1, 0, 0~
$ florist           <int> 0, 0, 0, 2, 0, 0, 2, 2, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ jeweller          <int> 0, 0, 0, 2, 1, 0, 0, 3, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ kiosk             <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ mall              <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ mobile_phone_shop <int> 0, 0, 0, 2, 1, 0, 0, 7, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ optician          <int> 0, 0, 0, 4, 1, 0, 2, 6, 2, 1, 1, 0,
0, 0, 0, 0, 0, 2~
$ shoe_shop         <int> 0, 0, 0, 10, 1, 0, 0, 7, 3, 0, 0,
0, 1, 0, 0, 1, 0, ~
$ toy_shop          <int> 0, 0, 0, 1, 1, 0, 0, 4, 3, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ travel_agent      <int> 0, 0, 0, 3, 0, 0, 0, 1, 1, 1, 0, 0,
0, 0, 0, 0, 0, 0~
$ car_rental        <int> 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0,
0, 0, 1, 0, 0, 0~
$ car_wash          <int> 0, 0, 0, 0, 1, 0, 0, 2, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ outdoor_shop      <int> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ sports_shop       <int> 0, 0, 0, 0, 0, 1, 1, 4, 1, 2, 0, 0,
0, 0, 0, 0, 0, 0~
$ caravan_site      <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ biergarten        <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ garden_centre     <int> 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ laundry           <int> 0, 0, 0, 0, 0, 0, 0, 2, 4, 0, 0, 0,
0, 0, 1, 0, 0, 0~
$ department_store  <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 1, 0, 0~
$ newsagent         <int> 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0,
0, 0, 0, 0, 0, 1~
$ vending_parking   <int> 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ chalet            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ shelter           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
```

```
$ motel             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ camp_site         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ general           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ alpine_hut        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ video_shop        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ vending_machine   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ car_sharing       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
```

[9]:
```
tdata_01 |> slice_head(n = 5)
```

A tibble: 5 x 60

CMUN <chr>	CDIS <chr>	bakery <int>	bar <int>	butcher <int>	cafe <int>	car_dealership <int>	clothes <int>	convenience <int>
002	01	2	2	1	5	1	1	2
003	01	0	0	0	1	0	0	0
004	01	0	1	0	0	0	0	0
005	01	4	22	2	12	1	13	4
005	02	5	10	2	6	1	1	2

Data Extract

- Extraer nombres de columnas

[10]:
```
tdata_02 <- data.frame(desc_var = colnames(tdata_01)[3:ncol(tdata_01)])
```

[11]:
```
tdata_02
```

A data.frame: 58 x 1

desc_var <chr>
bakery
bar
butcher
cafe
car_dealership
clothes
convenience
fast_food
food_court
furniture_shop
gift_shop
greengrocer
guesthouse
hairdresser
hotel
pub
restaurant
stationery
vending_any
bicycle_rental
hostel
supermarket
beauty_shop
beverages
bicycle_shop
bookshop
chemist
computer_shop
doityourself
florist
jeweller
kiosk
mall
mobile_phone_shop
optician
shoe_shop
toy_shop
travel_agent
car_rental
car_wash
outdoor_shop
sports_shop
caravan_site
biergarten
garden_centre
laundry
department_store
newsagent
vending_parking
chalet
shelter

Data Transform

- Completar metadatos

[13]:
```
tdata_02 <- tdata_02 |>
    mutate(nombre_var = colnames(tdata_01)[3:ncol(tdata_01)]) |>
    left_join(data |> count(grupo, tipo),
                by = c("desc_var" = "tipo"))
```

[14]:
```
glimpse(tdata_02)
```

```
Rows: 58
Columns: 4
$ desc_var   <chr> "bakery", "bar", "butcher", "cafe",
"car_dealership", "clot~
$ nombre_var <chr> "bakery", "bar", "butcher", "cafe",
"car_dealership", "clot~
$ grupo      <chr> "comercio", "hosteleria", "comercio",
"hosteleria", "comerc~
$ n          <int> 627, 2044, 289, 1577, 261, 1435, 1200,
1049, 24, 286, 129, ~
```

[15]:
```
tdata_02
```

A data.frame: 58 x 4

desc_var <chr>	nombre_var <chr>	grupo <chr>	n <int>
bakery	bakery	comercio	627
bar	bar	hosteleria	2044
butcher	butcher	comercio	289
cafe	cafe	hosteleria	1577
car_dealership	car_dealership	comercio	261
clothes	clothes	comercio	1435
convenience	convenience	comercio	1200
fast_food	fast_food	hosteleria	1049
food_court	food_court	hosteleria	24
furniture_shop	furniture_shop	comercio	286
gift_shop	gift_shop	comercio	129
greengrocer	greengrocer	comercio	548
guesthouse	guesthouse	turismo	285
hairdresser	hairdresser	comercio	1366
hotel	hotel	turismo	340
pub	pub	hosteleria	1201
restaurant	restaurant	hosteleria	5772
stationery	stationery	comercio	190
vending_any	vending_any	comercio	55
bicycle_rental	bicycle_rental	comercio	235
hostel	hostel	turismo	90
supermarket	supermarket	comercio	1135
beauty_shop	beauty_shop	comercio	358
beverages	beverages	comercio	62
bicycle_shop	bicycle_shop	comercio	146
bookshop	bookshop	comercio	252
chemist	chemist	comercio	60
computer_shop	computer_shop	comercio	131
doityourself	doityourself	comercio	354
florist	florist	comercio	179
jeweller	jeweller	comercio	200
kiosk	kiosk	comercio	255
mall	mall	comercio	19
mobile_phone_shop	mobile_phone_shop	comercio	213
optician	optician	comercio	339
shoe_shop	shoe_shop	comercio	411
toy_shop	toy_shop	comercio	116
travel_agent	travel_agent	comercio	180
car_rental	car_rental	comercio	56
car_wash	car_wash	comercio	89
outdoor_shop	outdoor_shop	comercio	27
sports_shop	sports_shop	comercio	111
caravan_site	caravan_site	turismo	4
biergarten	biergarten	hosteleria	28
garden_centre	garden_centre	comercio	30
laundry	laundry	comercio	197
department_store	department_store	comercio	32
newsagent	newsagent	comercio	62
vending_parking	vending_parking	comercio	608
chalet	chalet	turismo	19
shelter	shelter	turismo	54

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

Estos datos no requieren tareas de este tipo.

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

1. Infraestructuras agregadas por distrito

[16]:
```
data_to_save_01 <- tdata_01
```

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

[18]:
```
caso <- "CU_18"
proceso <- '_05'
tarea <- "_13"
archivo <- "_01"
proper <- "_ditrito_pois"
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_01 <- paste0(caso, proceso, tarea, tcltk::tkgetSaveFile(), proper,␣
↪extension)
# path_out_01 <- paste0(oPath, file_save_01)
# write_csv(data_to_save_01, path_out_01)

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

OPCION B: Especificar el nombre de archivo

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

[19]:
```
file_save_01 <- paste0(caso, proceso, tarea, archivo, proper, extension)
path_out_01 <- paste0(oPath, file_save_01)
write_csv(data_to_save_01, path_out_01)

cat('File saved as: ')
path_out_01
```

```
File saved as:
```

'Data/Output/CU_18_05_13_01_ditrito_pois.csv'

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

[20]:
```
path_in_01 <- paste0(iPath, file_save_01)
file.copy(path_out_01, path_in_01, overwrite = TRUE)
```

TRUE

2. Metadatos de infraestructuras

[21]:
```
data_to_save_02 <- tdata_02
```

[22]:
```
archivo <- "_02"
proper <- "_pois_meta"
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_02 <- paste0(caso, proceso, tarea, tcltk::tkgetSaveFile(), proper,␣
↪extension)
# path_out_02 <- paste0(oPath, file_save_02)
# write_csv(data_to_save_02, path_out_02)

# 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_02 <- paste0(caso, proceso, tarea, archivo, proper, extension)
path_out_02 <- paste0(oPath, file_save_02)
write_csv(data_to_save_02, path_out_02)

cat('File saved as: ')
path_out_02
```

```
File saved as:
```

'Data/Output/CU_18_05_13_02_pois_meta.csv'

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

[24]:
```
path_in_02 <- paste0(iPath, file_save_02)
file.copy(path_out_02, path_in_02, 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}, {tidyr} and {janitor} 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 * tidyr 1.3.0 * dplyr 1.0.10 * janitor 2.1.0 * readr 2.1.3

0.8.3 Data structures

Objeto `tdata_01`

- Tenemos 224 filas, una por distrito, con los recuentos de cada uno de los tipos de pois (58)

[25]:
```
glimpse(tdata_01)
```

```
Rows: 224
Columns: 60
$ cmun              <chr> "002", "003", "004", "005", "005",
"005", "005", "00~
$ cdis              <chr> "01", "01", "01", "01", "02", "03",
"04", "05", "01"~
$ bakery            <int> 2, 0, 0, 4, 5, 1, 2, 7, 4, 1, 1, 0,
2, 0, 0, 1, 0, 0~
$ bar               <int> 2, 0, 1, 22, 10, 2, 3, 11, 7, 1, 1,
2, 6, 0, 2, 17, ~
$ butcher           <int> 1, 0, 0, 2, 2, 0, 1, 2, 0, 0, 1, 0,
0, 0, 0, 2, 0, 0~
$ cafe              <int> 5, 1, 0, 12, 6, 1, 3, 19, 12, 3, 2,
0, 6, 0, 1, 4, 0~
$ car_dealership    <int> 1, 0, 0, 1, 1, 3, 0, 3, 5, 1, 0, 0,
0, 0, 0, 0, 0, 0~
$ clothes           <int> 1, 0, 0, 13, 1, 0, 1, 28, 0, 0, 0,
0, 0, 0, 1, 1, 0,~
$ convenience       <int> 2, 0, 0, 4, 2, 2, 2, 13, 8, 1, 1,
2, 6, 0, 0, 1, 0, ~
$ fast_food         <int> 1, 0, 0, 8, 1, 2, 5, 11, 10, 7, 4,
0, 6, 0, 2, 5, 0,~
$ food_court        <int> 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ furniture_shop    <int> 2, 0, 0, 3, 0, 0, 1, 8, 2, 1, 0, 0,
0, 0, 0, 0, 0, 0~
$ gift_shop         <int> 1, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0~
$ greengrocer       <int> 1, 0, 0, 1, 1, 0, 0, 8, 2, 0, 1, 0,
1, 0, 0, 0, 0, 0~
$ guesthouse        <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0~
$ hairdresser       <int> 2, 0, 0, 16, 4, 0, 4, 21, 7, 1, 0,
1, 5, 0, 0, 1, 0,~
$ hotel             <int> 1, 1, 0, 3, 0, 3, 0, 1, 3, 3, 0, 0,
0, 0, 2, 1, 0, 1~
$ pub               <int> 2, 0, 0, 24, 4, 2, 2, 31, 7, 0, 0,
0, 4, 0, 0, 2, 1,~
$ restaurant        <int> 5, 4, 3, 53, 5, 18, 10, 22, 73, 26,
13, 1, 4, 2, 2, ~
$ stationery        <int> 1, 0, 0, 1, 3, 0, 1, 2, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1~
$ vending_any       <int> 1, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ bicycle_rental    <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0~
$ hostel            <int> 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0~
```

```
$ supermarket       <int> 0, 1, 1, 6, 6, 2, 3, 10, 8, 12, 3,
4, 3, 1, 3, 3, 0,~
$ beauty_shop       <int> 0, 0, 0, 3, 0, 0, 2, 10, 6, 0, 0,
0, 0, 0, 1, 0, 0, ~
$ beverages         <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ bicycle_shop      <int> 0, 0, 0, 3, 1, 1, 0, 2, 4, 0, 1, 1,
1, 0, 0, 2, 0, 0~
$ bookshop          <int> 0, 0, 0, 7, 0, 0, 2, 3, 3, 1, 0, 0,
0, 0, 0, 0, 0, 1~
$ chemist           <int> 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ computer_shop     <int> 0, 0, 0, 3, 3, 0, 0, 2, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0~
$ doityourself      <int> 0, 0, 0, 2, 2, 2, 0, 3, 3, 0, 0, 0,
0, 0, 0, 1, 0, 0~
$ florist           <int> 0, 0, 0, 2, 0, 0, 2, 2, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ jeweller          <int> 0, 0, 0, 2, 1, 0, 0, 3, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ kiosk             <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ mall              <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ mobile_phone_shop <int> 0, 0, 0, 2, 1, 0, 0, 7, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ optician          <int> 0, 0, 0, 4, 1, 0, 2, 6, 2, 1, 1, 0,
0, 0, 0, 0, 0, 2~
$ shoe_shop         <int> 0, 0, 0, 10, 1, 0, 0, 7, 3, 0, 0,
0, 1, 0, 0, 1, 0, ~
$ toy_shop          <int> 0, 0, 0, 1, 1, 0, 0, 4, 3, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ travel_agent      <int> 0, 0, 0, 3, 0, 0, 0, 1, 1, 1, 0, 0,
0, 0, 0, 0, 0, 0~
$ car_rental        <int> 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0,
0, 0, 1, 0, 0, 0~
$ car_wash          <int> 0, 0, 0, 0, 1, 0, 0, 2, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ outdoor_shop      <int> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ sports_shop       <int> 0, 0, 0, 0, 0, 1, 1, 4, 1, 2, 0, 0,
0, 0, 0, 0, 0, 0~
$ caravan_site      <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ biergarten        <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ garden_centre     <int> 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0~
```

```
$ laundry           <int> 0, 0, 0, 0, 0, 0, 0, 2, 4, 0, 0, 0,
0, 0, 1, 0, 0, 0~
$ department_store  <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 1, 0, 0~
$ newsagent         <int> 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0,
0, 0, 0, 0, 0, 1~
$ vending_parking   <int> 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ chalet            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ shelter           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ motel             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ camp_site         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ general           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ alpine_hut        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ video_shop        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ vending_machine   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
$ car_sharing       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0~
```

0.9 Objeto `tdata_02`

- Tenemos 58 filas, una por tipo de poi, con sus metadatos

[26]:
```
glimpse(tdata_02)
```

```
Rows: 58
Columns: 4
$ desc_var   <chr> "bakery", "bar", "butcher", "cafe",
"car_dealership", "clot~
$ nombre_var <chr> "bakery", "bar", "butcher", "cafe",
"car_dealership", "clot~
$ grupo      <chr> "comercio", "hosteleria", "comercio",
"hosteleria", "comerc~
$ n          <int> 627, 2044, 289, 1577, 261, 1435, 1200,
1049, 24, 286, 129, ~
```

Observaciones generales sobre los datos

- No aplica

0.9.1 Consideraciones para despliegue en piloto

- Utilizar los metadatos para agrupar variables si hiciera falta.

0.9.2 Consideraciones para despliegue en producción

- No aplica

0.10 Main Actions

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

- Se han agrupado las infraesctructuras por distrito
- Se han calculado el número de infraestructuras por tipo en cada distrito

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

- Se deben unir al resto de datos por distrito

0.11 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
```