05. - Data Collection_CU_18_14_aemet_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 14. Obtener datos de estaciones AEMET

- Descarga de datos desde AEMET

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

- {stringr} para manipulación de strings
- {climaemet} para obtener los datos de aemet

[10]:
```
library(sf)
library(readr)
library(dplyr)
library(stringr)
library(climaemet)
```

0.1.2 Paths

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

0.2 Data Load

No aplica

0.3 Open Data

0.3.1 Datos de AEMET

1. Obtener API key de AEMET
2. Descargar datos directamente con el paquete `climaemet`
3. Transformar para guardar en CSV

[3]:
```
# browseURL("https://opendata.aemet.es/centrodedescargas/obtencionAPIKey")
```

[4]:
```
aemet_api_key("eyJhbGciOiJIUzI1NiJ9.
↪eyJzdWIiOiJlbWlsaW8ubG9wZXpAdXJqYy5lcyIsImp0aSI6ImI4ZmEyNjMyLTI3ODktNDcwNi1hMGM1LWNmNjI1MWU
↪uf2TizyDvDXQ4G12POBoNFZirdGtT1EW09emo5mvtj8")
```

```
To install your API key for use in future sessions, run this function with
`install = TRUE`.
```

Estaciones que usaremos

[5]:
```
estacionesm <- aemet_stations() |>
  filter(provincia == "MADRID") |>
  pull(indicativo)
```

[6]:
```
data <-
  aemet_daily_clim(estacionesm,
                   start = "2019-01-01",
                   end = "2019-12-31",
                   return_sf = TRUE)
```

```
Warning message in is.na(tbl[[lat]]) || any(is.na(tbl[[lon]])):
"'length(x) = 4547 > 1' in coercion to 'logical(1)'"
```

Estructura de los datos:

[7]:
```
glimpse(data)
```

```
Rows: 4,547
Columns: 21
$ fecha       <date> 2019-01-01, 2019-01-02, 2019-01-03,
2019-01-04, 2019-01-0~
$ indicativo  <chr> "2462", "2462", "2462", "2462", "2462",
"2462", "2462", "2~
$ nombre      <chr> "PUERTO DE NAVACERRADA", "PUERTO DE
NAVACERRADA", "PUERTO ~
$ provincia   <chr> "MADRID", "MADRID", "MADRID", "MADRID",
"MADRID", "MADRID"~
$ altitud     <dbl> 1894, 1894, 1894, 1894, 1894, 1894, 1894,
1894, 1894, 1894~
$ tmed        <dbl> 6.4, 5.2, 7.6, 6.0, 7.2, 4.6, 4.1, 6.2,
-0.8, -4.1, -5.2, ~
$ prec        <chr> "0,0", "0,0", "0,0", "0,0", "0,0", "0,0",
"0,0", "0,0", "0~
$ tmin        <dbl> 1.6, -0.8, 2.7, 1.7, 1.8, 0.9, 1.2, 1.4,
-3.5, -7.5, -10.3~
$ horatmin    <chr> "23:40", "08:00", "23:59", "01:30",
"18:00", "07:10", "Var~
$ tmax        <dbl> 11.3, 11.3, 12.6, 10.2, 12.5, 8.2, 7.0,
10.9, 2.0, -0.7, -~
$ horatmax    <chr> "12:40", "12:40", "12:00", "12:10",
"12:50", "11:40", "13:~
$ dir         <chr> "36", "04", "15", "15", "03", "35", "36",
"02", "36", "36"~
$ velmedia    <dbl> 3.3, 2.8, 1.9, 1.4, 3.1, 3.3, 2.8, 2.8,
5.3, 3.1, 2.8, 1.9~
$ racha       <dbl> 8.9, 11.4, 8.9, 8.3, 8.3, 11.9, 7.2,
11.1, 12.8, 8.6, 8.9,~
$ horaracha   <chr> "23:59", "17:10", "04:00", "00:40",
"06:50", "08:10", "06:~
$ sol         <dbl> 7.9, 7.9, 7.3, 8.0, 8.2, 8.0, 8.1, 8.1,
3.6, 1.2, 8.2, 7.8~
$ presMax     <dbl> 823.1, 822.0, 819.5, 822.7, 824.3, 823.3,
821.8, 821.3, 81~
$ horaPresMax <chr> "10", "00", "10", "23", "10", "00", "11",
"Varias", "00", ~
$ presMin     <dbl> 821.7, 818.8, 818.0, 818.5, 822.4, 821.0,
820.0, 818.4, 81~
$ horaPresMin <chr> "05", "24", "05", "05", "Varias", "16",
```

```
"14", "24", "24", ~
$ geometry    <POINT [arc_degree]> POINT (-4.010556
40.79306), POINT (-4.01055~
```

Muestra primeros datos

[12]:
```
data |> tibble() |> slice_head(n = 5)
```

A tibble: 5 x 21

fecha <date>	indicativo <chr>	nombre <chr>	provincia <chr>	altitud <dbl>	tmed <dbl>	prec <chr
2019-01-01	2462	PUERTO DE NAVACERRADA	MADRID	1894	6.4	0,0
2019-01-02	2462	PUERTO DE NAVACERRADA	MADRID	1894	5.2	0,0
2019-01-03	2462	PUERTO DE NAVACERRADA	MADRID	1894	7.6	0,0
2019-01-04	2462	PUERTO DE NAVACERRADA	MADRID	1894	6.0	0,0
2019-01-05	2462	PUERTO DE NAVACERRADA	MADRID	1894	7.2	0,0

0.4 ETL Processes

Transform data

- Asignar precipitación 0 a los valores inapreciables
- Convertir la precipitación en numérica
- Convertir objeto espacial en coordenadas numéricas para exportar a csv

[13]:
```
tdata <- data |>
    mutate(prec = str_replace(prec, "Ip", "0"),
           prec = as.numeric(str_replace(prec, ",", "."))) |>
    bind_cols(st_coordinates(data)) |>
    st_drop_geometry()
```

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

```
Rows: 4,547
Columns: 22
$ fecha       <date> 2019-01-01, 2019-01-02, 2019-01-03,
2019-01-04, 2019-01-0~
$ indicativo  <chr> "2462", "2462", "2462", "2462", "2462",
"2462", "2462", "2~
$ nombre      <chr> "PUERTO DE NAVACERRADA", "PUERTO DE
NAVACERRADA", "PUERTO ~
$ provincia   <chr> "MADRID", "MADRID", "MADRID", "MADRID",
"MADRID", "MADRID"~
$ altitud     <dbl> 1894, 1894, 1894, 1894, 1894, 1894, 1894,
1894, 1894, 1894~
$ tmed        <dbl> 6.4, 5.2, 7.6, 6.0, 7.2, 4.6, 4.1, 6.2,
-0.8, -4.1, -5.2, ~
$ prec        <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0~
$ tmin        <dbl> 1.6, -0.8, 2.7, 1.7, 1.8, 0.9, 1.2, 1.4,
```

```
-3.5, -7.5, -10.3~
$ horatmin    <chr> "23:40", "08:00", "23:59", "01:30",
"18:00", "07:10", "Var~
$ tmax        <dbl> 11.3, 11.3, 12.6, 10.2, 12.5, 8.2, 7.0,
10.9, 2.0, -0.7, -~
$ horatmax    <chr> "12:40", "12:40", "12:00", "12:10",
"12:50", "11:40", "13:~
$ dir         <chr> "36", "04", "15", "15", "03", "35", "36",
"02", "36", "36"~
$ velmedia    <dbl> 3.3, 2.8, 1.9, 1.4, 3.1, 3.3, 2.8, 2.8,
5.3, 3.1, 2.8, 1.9~
$ racha       <dbl> 8.9, 11.4, 8.9, 8.3, 8.3, 11.9, 7.2,
11.1, 12.8, 8.6, 8.9,~
$ horaracha   <chr> "23:59", "17:10", "04:00", "00:40",
"06:50", "08:10", "06:~
$ sol         <dbl> 7.9, 7.9, 7.3, 8.0, 8.2, 8.0, 8.1, 8.1,
3.6, 1.2, 8.2, 7.8~
$ presMax     <dbl> 823.1, 822.0, 819.5, 822.7, 824.3, 823.3,
821.8, 821.3, 81~
$ horaPresMax <chr> "10", "00", "10", "23", "10", "00", "11",
"Varias", "00", ~
$ presMin     <dbl> 821.7, 818.8, 818.0, 818.5, 822.4, 821.0,
820.0, 818.4, 81~
$ horaPresMin <chr> "05", "24", "05", "05", "Varias", "16",
"14", "24", "24", ~
$ X           <dbl> -4.010556, -4.010556, -4.010556,
-4.010556, -4.010556, -4.~
$ Y           <dbl> 40.79306, 40.79306, 40.79306, 40.79306,
40.79306, 40.79306~
```

0.5 Synthetic Data Generation

No aplica

0.6 Fake Data Generation

No aplica

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

[15]:
```
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

[16]:
```
caso <- "CU_18"
proceso <- '_05'
tarea <- "_14"
archivo <- ""
proper <- "_aemet"
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/.
- Pero en este caso lo guardamos solo en Input porque es una descarga directa y no viene de otro input

[17]:
```
file_save <- paste0(caso, proceso, tarea, archivo, proper, extension)
path_out <- paste0(iPath, file_save)
write_csv(data_to_save, path_out)

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

```
File saved as:
```

'Data/Input/CU_18_05_14_aemet.csv'

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 {xxx} 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 * climaemet 1.0.2 * dplyr 1.0.10 * readr 2.1.3 * stringr 1.5.0 * sf 1.0.9

0.8.3 Data structures

Objeto `tdata`

- Los datos de origen se descargan en formato `sf`
- Hay 4293 filas con las variables:
  - fecha
  - indicativo
  - nombre
  - provincia
  - altitud
  - tmed
  - prec
  - tmin
  - horatmin
  - tmax
  - horatmax
  - dir
  - velmedia
  - racha
  - horaracha
  - sol
  - presMax
  - horaPresMax
  - presMin
  - horaPresMin
  - X
  - Y

Observaciones generales sobre los datos

- Se han obtenido datos de 2022.

- Hay varias variables meteorológicas, suele haber bastantes valores perdidos

0.8.4 Consideraciones para despliegue en piloto

- Se ha utilizado una api key particular: conseguir una específica para proyecto antes de ejecutar

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
- Se ha utilizado una api key particular: conseguir una específica para proyecto antes de ejecutar

0.9 Main Actions

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

- Se han descargado los datos de AEMET por estaciones

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

- Se debe interpolar la temperatura y precipitaciones a los distritos censales

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
```