05. - Data Collection_CU_04_16_vacunacion_gripev_01

June 8, 2023

#

CU04_Optimización de vacunas

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 16. Unión de datos semanales por zonas de vacunación y otras

- Consolidación de todas las variables por semana del caso de uso
- Variables adicionales para la campaña y la semana de campaña
- Vacunación, meteo, contaminación

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 Encoding

Con la siguiente expresión se evitan problemas con el encoding al ejecutar el notebook. Es posible que deba ser eliminada o adaptada a la máquina en la que se ejecute el código.

[1]:
```
Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8")
```

'es_ES.UTF-8/es_ES.UTF-8/es_ES.UTF-8/C/es_ES.UTF-8/C'

0.1.2 Packages to use

- {tcltk} para selección interactiva de archivos locales
- {readxl} para leer archivos de Excel
- {readr} para leer y escribir archivos csv
- {dplyr} para explorar datos

[2]:
```
library(readxl)
library(readr)
library(dplyr)
```

```
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
```

0.1.3 Paths

[3]:
```
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. Datos de vacunación y capacidad

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

[4]:
```
# file_data_01 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[5]:
```
iFile_01 <- "Output.xlsx"
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 leerán datos del archivo:  Data/Input/Output.xlsx
```

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

[6]:
```
data_01 <- read_excel(file_data_01)
```

Estructura de los datos:

[7]:
```
data_01 |> glimpse()
```

```
Rows: 20,868
Columns: 22
$ ID                  <chr> "1", "2", "3", "4", "5", "6",
"7", "8", "9", "10"…
$ CODBDT              <chr> "686213", "686213", "686213",
"686213", "686213",…
$ GEOCODIGO           <chr> "001", "001", "001", "001",
"001", "001", "001", …
$ DESBDT              <chr> "Abrantes", "Abrantes",
"Abrantes", "Abrantes", "…
$ semana              <dbl> 36, 37, 38, 39, 40, 41, 42, 43,
44, 45, 46, 47, 4…
$ ano                 <dbl> 2021, 2021, 2021, 2021, 2021,
2021, 2021, 2021, 2…
$ n_vacunas           <dbl> 0, 0, 0, 0, 0, 328, 344, 353,
371, 341, 389, 349,…
$ n_citas             <dbl> 0, 0, 0, 0, 0, 305, 327, 328,
341, 334, 373, 332,…
$ capacidad_zona      <dbl> 7742, 7654, 7425, 7189, 7566,
7417, 7638, 7711, 7…
$ prop_riesgo         <dbl> 0.1954797, 0.1906348, 0.1930880,
0.1918070, 0.185…
$ tasa_riesgo         <dbl> 0.1950852, 0.1870784, 0.1874322,
0.2031197, 0.188…
$ poblacion_total     <dbl> 29872, 29872, 29872, 29872,
29872, 29872, 29872, …
$ poblacion_mujeres   <dbl> 0.5345094, 0.5345094, 0.5345094,
```

```
0.5345094, 0.534…
$ poblacion_mayores    <dbl> 0.1781619, 0.1781619, 0.1781619,
0.1781619, 0.178…
$ poblacion_inmigrante <dbl> 0.2183993, 0.2183993, 0.2183993,
0.2183993, 0.218…
$ tasa_paro            <dbl> 0.1665607, 0.1665607, 0.1665607,
0.1665607, 0.166…
$ tasa_mayores         <dbl> 0.1644978, 0.1882310, 0.1940977,
0.1781833, 0.181…
$ temperatura          <lgl> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, N…
$ humedad              <lgl> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, N…
$ NO2                  <lgl> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, N…
$ busquedas_gripe      <lgl> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, N…
$ tuits_gripe          <dbl> 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 3…
```

Muestra de datos:

[8]:
```
data_01 |> slice_head(n = 5)
```

A tibble: 5 x 22

ID <chr>	CODBDT <chr>	GEOCODIGO <chr>	DESBDT <chr>	semana <dbl>	ano <dbl>	n_vacunas <dbl>	n_citas <dbl>	ca <c
1	686213	001	Abrantes	36	2021	0	0	77
2	686213	001	Abrantes	37	2021	0	0	76
3	686213	001	Abrantes	38	2021	0	0	74
4	686213	001	Abrantes	39	2021	0	0	71
5	686213	001	Abrantes	40	2021	0	0	75

2. Datos meteorológicos

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

[9]:
```
# file_data_02 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[10]:
```
iFile_02 <- "CU_04_05_11_meteo_zonas.csv"
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 leerán datos del archivo:  Data/Input/CU_04_05_11_meteo_zonas.csv
```

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

[11]:
```
data_02 <- read_csv(file_data_02)
```

```
Rows: 21736 Columns: 7
  Column specification

Delimiter: ","
chr (1): GEOCODIGO
dbl (6): tmed, prec, velmedia, presMax, ano, semana

 Use `spec()` to retrieve the full column specification for this
data.
 Specify the column types or set `show_col_types = FALSE` to quiet
this message.
```

Estructura de los datos:

[12]:
```
data_02 |> glimpse()
```

```
Rows: 21,736
Columns: 7
$ GEOCODIGO <chr> "001", "002", "003", "004", "005", "006",
"007", "008", "009…
$ tmed      <dbl> 22.53414, 22.42739, 22.44443, 22.37906,
21.83238, 22.57096, …
$ prec      <dbl> 1.2086417, 1.2461203, 1.2778878, 1.2615862,
1.2790072, 1.019…
$ velmedia  <dbl> 2.299920, 2.310469, 2.382023, 2.321377,
2.461297, 2.029950, …
$ presMax   <dbl> 940.7107, 938.2132, 941.6089, 939.4657,
951.1882, 944.2638, …
$ ano       <dbl> 2021, 2021, 2021, 2021, 2021, 2021, 2021,
2021, 2021, 2021, …
$ semana    <dbl> 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35,
35, 35, 35, 35, …
```

Muestra de datos:

[13]:
```
data_02 |> slice_head(n = 5)
```

	GEOCODIGO <chr>	tmed <dbl>	prec <dbl>	velmedia <dbl>	presMax <dbl>	ano <dbl>	semana <dbl>
A spec_tbl_df: 5 x 7	001	22.53414	1.208642	2.299920	940.7107	2021	35
	002	22.42739	1.246120	2.310469	938.2132	2021	35
	003	22.44443	1.277888	2.382023	941.6089	2021	35
	004	22.37906	1.261586	2.321377	939.4657	2021	35
	005	21.83238	1.279007	2.461297	951.1882	2021	35

3. Datos de contaminación

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

[14]:
```
# file_data_03 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[15]:
```
iFile_03 <- "CU_04_05_15_zonas_contaminacion.csv"
file_data_03 <- paste0(iPath, iFile_03)

if(file.exists(file_data_03)){
    cat("Se leerán datos del archivo: ", file_data_03)
} else{
    warning("Cuidado: el archivo no existe.")
}
```

```
Se leerán datos del archivo:  Data/Input/CU_04_05_15_zonas_contaminacion.csv
```

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

[16]:
```
data_03 <- read_csv(file_data_03)
```

```
Rows: 21164 Columns: 12
  Column specification

Delimiter: ","
chr  (1): GEOCODIGO
dbl (11): benzene, co, no, no2, nox, o3, pm10, pm2.5, so2, ano, semana

 Use `spec()` to retrieve the full column specification for this
data.
 Specify the column types or set `show_col_types = FALSE` to quiet
this message.
```

Estructura de los datos:

[17]:
```
data_03 |> glimpse()
```

```
Rows: 21,164
Columns: 12
$ GEOCODIGO <chr> "001", "002", "003", "004", "005", "006",
"007", "008", "009…
$ benzene   <dbl> 0.2478896, 0.2351470, 0.2438505, 0.2311297,
0.2582252, 0.326…
$ co        <dbl> 0.2783157, 0.2720947, 0.2564127, 0.2704844,
0.3730037, 0.422…
$ no        <dbl> 4.277267, 4.277267, 4.277267, 4.277267,
4.277267, 4.277267, …
$ no2       <dbl> 30.81135, 30.34314, 30.12568, 30.05632,
25.87842, 27.88230, …
$ nox       <dbl> 38.70242, 38.26907, 37.77193, 37.77965,
32.30093, 35.32128, …
$ o3        <dbl> 53.41865, 55.30295, 55.55106, 57.16719,
57.60870, 59.73277, …
$ pm10      <dbl> 17.55392, 21.35499, 22.63530, 23.80635,
22.43009, 17.30619, …
$ pm2.5     <dbl> 12.77851, 12.84574, 12.85695, 12.86689,
12.18365, 12.24123, …
$ so2       <dbl> 6.747270, 7.411119, 6.654251, 7.439685,
2.376795, 1.334976, …
$ ano       <dbl> 2021, 2021, 2021, 2021, 2021, 2021, 2021,
2021, 2021, 2021, …
$ semana    <dbl> 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35,
35, 35, 35, …
```

Muestra de datos:

[18]:
```
data_03 |> slice_head(n = 5)
```

A spec_tbl_df: 5 x 12

GEOCODIGO <chr>	benzene <dbl>	co <dbl>	no <dbl>	no2 <dbl>	nox <dbl>	o3 <dbl>	pm <d
001	0.2478896	0.2783157	4.277267	30.81135	38.70242	53.41865	17.
002	0.2351470	0.2720947	4.277267	30.34314	38.26907	55.30295	21.
003	0.2438505	0.2564127	4.277267	30.12568	37.77193	55.55106	22.
004	0.2311297	0.2704844	4.277267	30.05632	37.77965	57.16719	23.
005	0.2582252	0.3730037	4.277267	25.87842	32.30093	57.60870	22.

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:

- Datos simulados de vacunación y otros
- Datos meteorológicos
- Datos de contaminación

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

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

Data extract

- Seleccionar columnas de la simulación que son por semana

[19]:
```
edata_01 <- data_01 |>
    select(GEOCODIGO, DESBDT, ano, semana, n_vacunas,
        n_citas)
```

Data transform

- Unión de tablas por semana y zona

[20]:
```
data <- edata_01 |>
    full_join(data_02, by = c("GEOCODIGO", "ano", "semana")) |>
    full_join(data_03, by = c("GEOCODIGO", "ano", "semana"))
```

[21]:
```
glimpse(data)
```

```
Rows: 21,736
Columns: 19
$ GEOCODIGO <chr> "001", "001", "001", "001", "001", "001",
"001", "001", "001…
$ DESBDT    <chr> "Abrantes", "Abrantes", "Abrantes",
"Abrantes", "Abrantes", …
$ ano       <dbl> 2021, 2021, 2021, 2021, 2021, 2021, 2021,
2021, 2021, 2021, …
$ semana    <dbl> 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46,
47, 48, 49, 50, …
$ n_vacunas <dbl> 0, 0, 0, 0, 0, 328, 344, 353, 371, 341,
389, 349, 402, 350, …
$ n_citas   <dbl> 0, 0, 0, 0, 0, 305, 327, 328, 341, 334,
373, 332, 370, 336, …
$ tmed      <dbl> 23.822231, 20.160014, 18.551058, 18.815387,
17.494475, 17.56…
$ prec      <dbl> -0.0063023484, 3.5537258042, 3.9769052178,
1.4806472324, -0.…
$ velmedia  <dbl> 3.573728, 2.494744, 3.316148, 2.384262,
1.850839, 1.527085, …
$ presMax   <dbl> 940.9841, 940.7610, 943.1540, 944.6697,
944.2973, 943.0733, …
$ benzene   <dbl> 0.1713567, 0.1573829, 0.1858059, 0.1486437,
0.1428037, 0.141…
$ co        <dbl> 0.1680325, 0.2138607, 0.2034376, 0.2399882,
0.2693450, 0.283…
$ no        <dbl> 4.098371, 6.515572, 5.477654, 9.593391,
18.860535, 17.000221…
```

```
$ no2       <dbl> 20.09480, 27.42594, 20.74836, 37.08524,
40.19475, 44.42785, …
$ nox       <dbl> 26.48135, 37.45944, 25.61128, 52.43745,
74.04903, 75.16833, …
$ o3        <dbl> 50.03434, 42.41281, 56.29918, 46.79483,
41.06600, 44.01453, …
$ pm10      <dbl> 17.447652, 17.658399, 12.844436, 16.395896,
14.909384, 21.12…
$ pm2.5     <dbl> 3.008675, 10.083070, 7.218588, 9.426029,
8.131753, 12.378902…
$ so2       <dbl> 6.861545, 6.589638, 4.364304, 3.123598,
1.291137, 1.841904, …
```

- Variables auxiliares de campaña. Se asume que la campaña empieza la semana 36 y termina la semana 5

[22]:
```
SEMANA_INICIO <- 36
SEMANA_FIN <- 5
```

[24]:
```
tdata <- data |>
    mutate(
        campana = case_when(semana >= SEMANA_INICIO ~ ano,
                            semana <= SEMANA_FIN ~ ano-1,
                            .default = NA),
        scampana = case_when(semana >= SEMANA_INICIO ~ semana - SEMANA_INICIO␣
 ↪+1,
                             semana <= SEMANA_FIN ~ semana + 52 + 1 -␣
 ↪SEMANA_INICIO,
                             .default = NA))
```

0.4 Synthetic Data Generation

Algunos de los datos se habían generado de forma sintética

0.5 Fake Data Generation

No aplica

0.6 Open Data

Las zonas de salud se habían obtenido de fuentes abiertas

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

[25]:
```
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

[26]:
```
caso <- "CU_04"
proceso <- '_05'
tarea <- "_16"
archivo <- ""
proper <- "_vacunacion_gripe"
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/.

[27]:
```
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_04_05_16_vacunacion_gripe.csv’

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

[28]:
```
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

Para que funcione este código se necesita:

- Las rutas de archivos `Data/Input` y `Data/Output` deben existir (relativas a la ruta del *notebook*)
- El paquete tcltk instalado para seleccionar archivos interactivamente. No se necesita en producción.
- Los paquetes tcltk, readxl, readr, dplyr deben estar instalados.

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 * readxl 1.4.1 * readr 2.1.3 * dplyr 1.0.10

0.8.3 Data structures

Objeto `tdata`

- Los datos de origen
- Hay 21736 filas con la información de las siguientes variables:
  - GEOCODIGO
  - DESBDT
  - ano
  - semana
  - n_vacunas
  - n_citas
  - tmed
  - prec
  - velmedia
  - presMax
  - benzene
  - co
  - no
  - no2
  - nox
  - o3

  - pm10
  - pm2.5
  - so2
  - campana
  - scampana

Observaciones generales sobre los datos

- Había más datos de vacunación que de las otras variables porque las primeras se simularon de las semanas exactas de la campaña, y el resto se obtuvieron de meses completos
- Se han mantenido todas las filas en este conjunto de datos por si son útiles para los modelos
- Los datos son únicos para cada zona, año y semana

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 unido las tablas con datos por semana y zona

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

- Se deben unir a las tablas con otra granularidad, repitiendo datos, para tener un csv que incluya todo

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
```