05. - Data Collection_CU_04_03_indicadores_secciones_v_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 03. Secciones censales y variables socioeconómicas

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.

[ ]:
```
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

- {readr} from *tidyverse* for reading and writing csv files
- {tcltk} for selecting files and paths (if needed)
- {dplyr} for data exploration
- {readxl} for reading excel files
- {stringr} for manipulating strings

[19]:
```
library(readr)
library(tcltk)
library(dplyr)
library(readxl)
library(stringr)
```

0.1.3 Paths

[20]:
```
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.

1. Indicadores por sección censal 2021 OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

[21]:
```
# file_data_01 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

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)

[22]:
```
iFile_01 <- "C2021_indicadores.csv"
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/C2021_indicadores.csv
```

Data file to dataframe

[23]:
```
data_01_secciones <- read_csv(file_data_01)
```

```
Rows: 36333 Columns: 20
  Column specification

Delimiter: ","
chr  (5): ccaa, CPRO, CMUN, dist, secc
dbl (15): t1_1, t2_1, t2_2, t3_1, t4_1, t4_2, t4_3, t5_1, t6_1, t7_1,
t8_1, …

 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:

[24]:
```
glimpse(data_01_secciones)
```

```
Rows: 36,333
Columns: 20
$ ccaa  <chr> "01", "01", "01", "01", "01", "01", "01", "01",
"01", "01", "01"…
$ CPRO  <chr> "04", "04", "04", "04", "04", "04", "04", "04",
"04", "04", "04"…
$ CMUN  <chr> "001", "002", "003", "003", "003", "003",
"003", "003", "003", "…
$ dist  <chr> "01", "01", "01", "01", "01", "01", "01", "01",
"01", "02", "02"…
$ secc  <chr> "001", "001", "001", "002", "003", "004",
"005", "006", "007", "…
$ t1_1  <dbl> 1260, 1212, 869, 1765, 2239, 2528, 2040, 1519,
888, 1190, 1463, …
$ t2_1  <dbl> 0.4857, 0.4719, 0.5086, 0.5008, 0.5092, 0.4869,
0.5015, 0.5069, …
$ t2_2  <dbl> 0.5143, 0.5281, 0.4914, 0.4992, 0.4908, 0.5131,
0.4985, 0.4931, …
$ t3_1  <dbl> 49.0008, 48.9695, 44.5984, 39.7705, 37.5775,
36.9256, 38.7466, 4…
$ t4_1  <dbl> 0.1048, 0.0891, 0.1208, 0.1773, 0.1894, 0.1998,
0.1730, 0.1646, …
$ t4_2  <dbl> 0.6008, 0.6287, 0.6789, 0.6929, 0.7034, 0.7017,
0.6941, 0.6471, …
$ t4_3  <dbl> 0.2944, 0.2822, 0.2002, 0.1297, 0.1072, 0.0985,
0.1328, 0.1883, …
$ t5_1  <dbl> 0.0984, 0.0941, 0.1231, 0.0793, 0.1215, 0.1250,
0.0912, 0.1238, …
$ t6_1  <dbl> 0.0984, 0.0899, 0.1277, 0.0782, 0.1170, 0.1274,
0.1010, 0.1284, …
$ t7_1  <dbl> 0.0833, 0.0598, 0.0759, 0.0482, 0.0562, 0.0302,
```

```
0.0670, 0.0654, …
$ t8_1  <dbl> 0.0762, 0.0498, 0.0628, 0.0386, 0.0435, 0.0222,
0.0433, 0.0512, …
$ t9_1  <dbl> 0.2163, 0.1812, 0.2356, 0.2231, 0.1730, 0.1265,
0.2116, 0.2648, …
$ t10_1 <dbl> 0.1951, 0.2052, 0.1576, 0.1258, 0.1432, 0.1705,
0.1561, 0.1438, …
$ t11_1 <dbl> 0.3768, 0.3614, 0.4686, 0.5455, 0.5802, 0.5289,
0.5513, 0.4925, …
$ t12_1 <dbl> 0.4681, 0.4547, 0.5563, 0.6240, 0.6771, 0.6377,
0.6532, 0.5753, …
```

Muestra de datos:

[25]:
```
slice_head(data_01_secciones, n = 5)
```

A spec_tbl_df: 5 × 20

ccaa <chr>	CPRO <chr>	CMUN <chr>	dist <chr>	secc <chr>	t1_1 <dbl>	t2_1 <dbl>	t2_2 <dbl>	t3_1 <dbl>	t4_1 <dbl
01	04	001	01	001	1260	0.4857	0.5143	49.0008	0.104
01	04	002	01	001	1212	0.4719	0.5281	48.9695	0.089
01	04	003	01	001	869	0.5086	0.4914	44.5984	0.120
01	04	003	01	002	1765	0.5008	0.4992	39.7705	0.177
01	04	003	01	003	2239	0.5092	0.4908	37.5775	0.189

0.2.1 2. Nombres de indicadores socioeconómicos

OPCION A: Seleccionar fichero en ventana para mayor comodidad

Data load using the {tcltk} package. Uncomment if using this option.

[26]:
```
# file_data_02 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

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) - Comentar si se usa la opción anterior

[27]:
```
iFile_02 <- "indicadores_seccen_c2021.xlsx"
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/indicadores_seccen_c2021.xlsx
```

Data file to dataframe

[28]:
```
data_02_indicadores <- read_excel(file_data_02,
                                  sheet = "indicadores",
                                  range = "A8:B23")
```

Estructura de los datos:

[29]:
```
glimpse(data_02_indicadores)
```

```
Rows: 15
Columns: 2
$ Tabla     <chr> "t1_1", "t2_1", "t2_2", "t3_1", "t4_1",
"t4_2", "t4_3", "t5_…
$ Indicador <chr> "Total Personas", "Porcentaje de mujeres",
"Porcentaje de ho…
```

Muestra de datos:

[30]:
```
slice_head(data_02_indicadores, n = 5)
```

A tibble: 5 × 2

Tabla <chr>	Indicador <chr>
t1_1	Total Personas
t2_1	Porcentaje de mujeres
t2_2	Porcentaje de hombres
t3_1	Edad media
t4_1	Porcentaje de personas menores de 16 años

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:

- Indicadores del INE por sección censal
- Descripciones de indicadores del INE de tabla publicada en Excel

0.3.2 Extract data

- Filtrar datos de indicadores solo de la Comunidad de Madrid

[31]:
```
data_01_secciones <- data_01_secciones |>
    filter(ccaa == 13)
```

Datos extraídos:

[32]:
```
glimpse(data_01_secciones)
```

```
Rows: 4,417
Columns: 20
$ ccaa  <chr> "13", "13", "13", "13", "13", "13", "13", "13",
"13", "13", "13"…
$ CPRO  <chr> "28", "28", "28", "28", "28", "28", "28", "28",
```

```
"28", "28", "28"…
$ CMUN  <chr> "001", "002", "002", "003", "004", "004",
"004", "004", "005", "…
$ dist  <chr> "01", "01", "01", "01", "01", "01", "01", "01",
"01", "01", "01"…
$ secc  <chr> "001", "001", "002", "001", "001", "002",
"003", "004", "001", "…
$ t1_1  <dbl> 55, 2358, 2435, 248, 2886, 3376, 2161, 1523,
1648, 1015, 1728, 1…
$ t2_1  <dbl> NA, 0.4724, 0.4830, 0.3952, 0.5135, 0.4793,
0.5016, 0.5279, 0.54…
$ t2_2  <dbl> NA, 0.5276, 0.5170, 0.6048, 0.4865, 0.5207,
0.4984, 0.4721, 0.45…
$ t3_1  <dbl> NA, 40.5161, 38.3339, 47.4597, 44.4099,
40.6810, 38.0088, 42.537…
$ t4_1  <dbl> NA, 0.1425, 0.1959, 0.1371, 0.1611, 0.1739,
0.2230, 0.1753, 0.11…
$ t4_2  <dbl> NA, 0.7443, 0.7002, 0.6573, 0.6067, 0.6727,
0.6395, 0.6448, 0.69…
$ t4_3  <dbl> NA, 0.1132, 0.1039, 0.2056, 0.2322, 0.1534,
0.1374, 0.1799, 0.18…
$ t5_1  <dbl> NA, 0.1569, 0.1544, 0.1129, 0.1639, 0.1440,
0.1749, 0.1326, 0.16…
$ t6_1  <dbl> NA, 0.1968, 0.1951, 0.1411, 0.2128, 0.1730,
0.2138, 0.1589, 0.22…
$ t7_1  <dbl> NA, 0.4016, 0.4244, 0.1542, 0.1470, 0.1596,
0.1668, 0.1584, 0.09…
$ t8_1  <dbl> NA, 0.3971, 0.4224, 0.1449, 0.1409, 0.1506,
0.1602, 0.1545, 0.09…
$ t9_1  <dbl> NA, 0.4377, 0.4438, 0.5280, 0.2602, 0.3234,
0.2859, 0.3057, 0.51…
$ t10_1 <dbl> NA, 0.0912, 0.1226, 0.0932, 0.1814, 0.1478,
0.1658, 0.1824, 0.10…
$ t11_1 <dbl> NA, 0.6063, 0.5955, 0.5000, 0.4213, 0.5170,
0.4943, 0.4745, 0.52…
$ t12_1 <dbl> NA, 0.6672, 0.6788, 0.5514, 0.5147, 0.6067,
0.5926, 0.5804, 0.58…
```

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

El archivo se ha obtenido de una fuente pública, ver apartado conclusiones

0.7 Data Save

Este proceso, puede copiarse y repetirse en aquellas partes del notebbok que necesiten guardar datos. Recuerde cambiar la extensión añadida del fichero para diferenciarlas

Identificamos los datos a guardar

0.7.1 1. Archivo de indicadores

[33]:
```
data_to_save_01 <- data_01_secciones
```

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

0.7.2 Proceso 05

[34]:
```
caso <- "CU_04"
proceso <- '_05'
tarea <- "_03"
archivo <- "_01"
proper <- "_indicadores_secciones"
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
- Descomentar líneas si se usa esta opción

[35]:
```
# file_save_01 <- paste0(caso, proceso, tarea, tcltk::tkgetSaveFile(), proper,␣
↪extension)
# path_out_01 <- paste0(oPath, file_save_01)
# write_csv(datos, path_out_01)

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

OPCION B: Especificar el nombre de archivo

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

[36]:
```
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_04_05_03_01_indicadores_secciones.csv'

Copia del fichero a Input Será usado en otros notebooks

[37]:
```
path_in_01 <- paste0(iPath, file_save_01)
file.copy(path_out_01, path_in_01)
```

TRUE

0.7.3 2. Descripción indicadores

[38]:
```
data_to_save_02 <- data_02_indicadores
```

[39]:
```
archivo <- "_02"
proper <- "_indicadores_nombres"
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
- Descomentar líneas si se usa esta opción

[40]:
```
# file_save_02 <- paste0(caso, proceso, tarea, tcltk::tkgetSaveFile(), proper,␣
 ↪extension)
# path_out_02 <- paste0(oPath, file_save_02)
# write_csv(datos, path_out_02)

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

OPCION B: Especificar el nombre de archivo

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

[41]:
```
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_04_05_03_02_indicadores_nombres.csv'

Copia del fichero a Input Será usado en otros notebooks

[42]:
```
path_in_02 <- paste0(iPath, file_save_02)
file.copy(path_out_02, path_in_02)
```

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}, {readxl} and {stringr} packages must be installed. They are part of the *tidyverse*.
- The {sf} package must be installed. It needs some system requierements, check the package documentation at https://r-spatial.github.io/sf/
- 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 * readr 2.1.3 * sf 1.0.9 * dplyr 1.0.10 * readxl 1.4.1 * stringr 1.5.0

0.8.3 Data structures

Objeto `data_01_secciones` (Indicadores INE por sección censal)

- Los datos de origen fueron descargados de https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadist
  csv con todas las comunidades, archivo 'C2021_Indicadores.csv'. Indicadores por sección censal Comunidad de Madrid censo 2021.
- Hay 15 indicadores numéricos de 36.233 secciones censales de toda España (4.147 de la Comunidad de Madrid), más 5 columnas indicadoras de comunidad autónoma, provincia, municipio, distrito y sección censal.

Objeto `data_02_indicadores` (descripciones indicadores INE)

- Los datos fueron obtenidos de https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=
  Archivo `indicadores_seccen_c2021.xlsx`: Códigos y nombres de indicadores censo 2021 en hoja indicadores (desde fila 8, 15 indicadores); tablas disponibles en hoja 'Tablas disponibles'.
- Hay 15 filas con las descripciones de los indicadores y el código que se utiliza como encabezado de columna en la tabla de indicadores.

0.8.4 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 filtrado los datos de indicadores del INE solo de la Comunidad de Madrid, ya que de vacunas no tendremos de otras comunidades

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

- Se deben unir los datos de indicadores a las zonas sanitarias básicas
- Se deben agregar los datos por zonas sanitarias

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

[43]:
```
# incluir código
```