05. - Data Collection_CU_53_03_paisesgeo_v_01

June 13, 2023

#

CU53_impacto de las políticas de inversión en sanidad, infraestructuras y promoción turística en el SPI

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. Obtener geometrías de países para representación SPI

- Obtener archivo json de geometrías de países para representación del SPI

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

ELIMINAR O AÑADIR LO QUE TOQUE. COPIAR VERSIONES AL FINAL Y QUITAR CÓDIGO DE VERSIONES

- {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 cadenas de caracteres
- {tidyr} para organización de datos

[2]:
```
library(spData)
library(sf)
library(countrycode)
library(dplyr)

p <- c("tcltk", "sf", "spData", "dplyr")
```

```
To access larger datasets in this package, install the spDataLarge
package with: `install.packages('spDataLarge',
repos='https://nowosad.github.io/drat/', type='source')`

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
```

0.1.3 Paths

[3]:
```
iPath <- "Data/Input/"
oPath <- "Data/Output/"
```

0.2 Data Load

No aplica

0.3 Open Data

- Obtener georreferenciación países con datos del paquete {spData}

[4]:
```
data <- world
```

0.4 ETL Processes

Data extract

- Quitar columnas no necesarias y dejar solo la georeferenciación
- Añadir código de país ISO de tres dígitos que es el que hay en el SPI

[5]:
```
edata <- data |> select(iso_a2:subregion) |>
  left_join(codelist |> select(iso2c, iso3c),
            by = c("iso_a2" = "iso2c"))
```

[6]:
```
edata |> glimpse()
```

```
Rows: 253
Columns: 7
$ iso_a2    <chr> "FJ", "TZ", "EH", "CA", "US", "KZ", "UZ",
"PG", "ID", "AR", …
$ name_long <chr> "Fiji", "Tanzania", "Western Sahara",
"Canada", "United Stat…
$ continent <chr> "Oceania", "Africa", "Africa", "North
America", "North Ameri…
$ region_un <chr> "Oceania", "Africa", "Africa", "Americas",
"Americas", "Asia…
$ subregion <chr> "Melanesia", "Eastern Africa", "Northern
Africa", "Northern …
$ geom      <MULTIPOLYGON [°]> MULTIPOLYGON (((-180
-16.55…, MULTIPOLYGON ((…
$ iso3c     <chr> "FJI", "TZA", "ESH", "CAN", "USA", "KAZ",
"UZB", "PNG", "IDN…
```

[7]:
```
edata |> tibble() |> head()
```

A tibble: 6 x 7

iso_a2 <chr>	name_long <chr>	continent <chr>	region_un <chr>	subregion <chr>	geom <MULTIPOLYGO
FJ	Fiji	Oceania	Oceania	Melanesia	MULTIPOLYGON
TZ	Tanzania	Africa	Africa	Eastern Africa	MULTIPOLYGON
EH	Western Sahara	Africa	Africa	Northern Africa	MULTIPOLYGON
CA	Canada	North America	Americas	Northern America	MULTIPOLYGON
US	United States	North America	Americas	Northern America	MULTIPOLYGON
KZ	Kazakhstan	Asia	Asia	Central Asia	MULTIPOLYGON

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

[8]:
```
data_to_save <- edata
```

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

[9]:
```
caso <- "CU_53"
proceso <- '_05'
tarea <- "_03"
archivo <- ""
proper <- "_paisesgeo"
extension <- ".json"
```

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

[10]:
```
# 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/.

[13]:
```
file_save <- paste0(caso, proceso, tarea, archivo, proper, extension)
path_out <- paste0(oPath, file_save)
st_write(obj = edata,
         dsn = path_out,
         driver = "GeoJSON",
         delete_dsn = TRUE)


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

```
Deleting source `Data/Output/CU_53_05_03_paisesgeo.json' failed
Writing layer `CU_53_05_03_paisesgeo' to data source
  `Data/Output/CU_53_05_03_paisesgeo.json' using driver `GeoJSON'
Writing 253 features with 6 fields and geometry type Multi Polygon.
File saved as:
```

'Data/Output/CU_53_05_03_paisesgeo.json'

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

[14]:
```
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 sf, spData, 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 * sf 1.0.9 * spData 2.2.1 * dplyr 1.0.10

0.8.3 Data structures

Objeto `edata`

- Hay 253 filas con información de las siguientes variables:
  - iso_a2
  - name_long
  - continent
  - region_un
  - subregion
  - geom
  - iso3c

Observaciones generales sobre los datos

- Estos datos servirán para visualización de los datos de entrada y resultados

0.8.4 Consideraciones para despliegue en piloto

- No aplica

0.8.5 Consideraciones para despliegue en producción

- Es muy posible que los datos no cambien nunca, pero se debería controlar que la codificación es correcta en futuras actualización del SPI

0.9 Main Actions

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

- Se han guardado las georeferencias de países
- Se ha añadido el código de tres letras

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

- Se deben unir las geometrías a los datos del SPI para su representación

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
```