05. - Data Collection_CU_53_05_inversiones_cm_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 05. Consolidar datos de inversiones en CM

- Consolidar los archivos descargados en un csv con las inversiones totales por área de relevancia.

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

[39]:
```
library(purrr)
library(readxl)
library(readr)
library(dplyr)
library(stringr)
library(tidyr)

p <- c("tcltk", "purrr", "readxl", "readr", "dplyr", "stringr", "tidyr")
```

0.1.3 Paths

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

0.2 Data Load

Son muchos archivos y se realiza en bucle. Se incluye la carge en las tareas ETL

0.3 ETL Processes

0.3.1 Import data from: CSV, Excel, Tab, JSON, SQL, and Parquet files

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.

- Se cargan en bucle
- Algunos valores de DISPUESTO vienen con totales, que los importa como caracteres, Como el total no nos hace falta, convertimos a numérico, por lo que se crean NA y se eliminan esas filas

[15]:
```
xlspath <- paste0(iPath, "xls/")
files <- dir(xlspath)
cols <- c("PRESUPUESTO","MES","CENTRO","DESC_CENTRO",
```

```
          "SECCION","DESC_SECCION","CENTRO_GESTOR",
          "DESC_CGESTOR","PROGRAMA","DESC_PROGRAMA","CAPITULO",
          "DESC_CAPITULO","CREDITO_INICIAL","MODIFICACIONES",
          "CREDITO_ACTUAL","AUTORIZADO","C_SALDO_PRESUPUESTO",
          "DISPUESTO","SALDO_AUTORIZACIONES",
          "OBLIGACIONES_RECONOCIDAS","SALDO_DISPOSICIONES")



data <- map_dfr(files,
              ~{read_excel(paste0(ipath, .x), skip = 4,
                           col_names = cols) |>
                select(DESC_SECCION, DESC_CAPITULO, DISPUESTO) |>
                mutate(DISPUESTO = as.numeric(DISPUESTO),
                       mes = str_sub(.x, -6, -5),
                       anyo = str_sub(.x, 11, 14)) |>
                  drop_na(DISPUESTO)})
```

```
Warning message in mask$eval_all_mutate(quo):
"NAs introducidos por coerción"
Warning message in mask$eval_all_mutate(quo):
"NAs introducidos por coerción"
Warning message in mask$eval_all_mutate(quo):
"NAs introducidos por coerción"
Warning message in mask$eval_all_mutate(quo):
"NAs introducidos por coerción"
Warning message in mask$eval_all_mutate(quo):
"NAs introducidos por coerción"
Warning message in mask$eval_all_mutate(quo):
"NAs introducidos por coerción"
Warning message in mask$eval_all_mutate(quo):
"NAs introducidos por coerción"
Warning message in mask$eval_all_mutate(quo):
"NAs introducidos por coerción"
```

[14]:
```
data |> glimpse()
```

```
Rows: 29,661
Columns: 5
$ DESC_SECCION  <chr> "PRESIDENCIA DE LA COMUNIDAD DE
MADRID", "PRESIDENCIA DE…
$ DESC_CAPITULO <chr> "GASTOS DE PERSONAL", "GASTOS
CORRIENTES EN BIENES Y SER…
$ DISPUESTO     <dbl> 93078.87, 0.00, 0.00, 654249.43,
2115219.80, 0.00, 0.00,…
$ mes           <chr> "01", "01", "01", "01", "01", "01",
"01", "01", "01", "0…
$ anyo          <chr> "2017", "2017", "2017", "2017", "2017",
"2017", "2017", …
```

[16]:
```
data |> slice_head(n = 5)
```

A tibble: 5 x 5

DESC_SECCION <chr>	DESC_CAPITULO <chr>
PRESIDENCIA DE LA COMUNIDAD DE MADRID	GASTOS DE PERSONAL
PRESIDENCIA DE LA COMUNIDAD DE MADRID	GASTOS CORRIENTES EN BIENES Y
PRESIDENCIA DE LA COMUNIDAD DE MADRID	ACTIVOS FINANCIEROS
CULTURA Y TURISMO	GASTOS DE PERSONAL
CULTURA Y TURISMO	GASTOS CORRIENTES EN BIENES Y

Data transformation

- Agrupar datos de inversiones

[25]:
```
tdata <- data |>
  filter(str_detect(DESC_CAPITULO, "INVERS")) |>
  group_by(DESC_SECCION, anyo) |>
  summarise(inversion = sum(DISPUESTO), .groups = "drop")
```

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

```
Rows: 85
Columns: 3
$ DESC_SECCION <chr> "ADMINISTRACIÓN LOCAL Y DIGITALIZACIÓN",
"ASAMBLEA DE MAD…
$ anyo         <chr> "2022", "2019", "2020", "2021", "2022",
"2019", "2020", "…
$ inversion    <dbl> 191225660.24, 5379677.43, 6550082.19,
3091085.72, 4625906…
```

[27]:
```
tdata |> slice_head(n = 5)
```

A tibble: 5 x 3

DESC_SECCION <chr>	anyo <chr>	inversion <dbl>
ADMINISTRACIÓN LOCAL Y DIGITALIZACIÓN	2022	191225660
ASAMBLEA DE MADRID	2019	5379677
ASAMBLEA DE MADRID	2020	6550082
ASAMBLEA DE MADRID	2021	3091086
ASAMBLEA DE MADRID	2022	4625906

- Hay secciones que van cambiando, detectamos los tres grupos y agregamos

[31]:
```
tdata_out <- tdata |>
  mutate(grupo = if_else(str_detect(DESC_SECCION, "SAN"),
                         "SANIDAD",
                         if_else(str_detect(DESC_SECCION, "TURISMO|TUR\\."),
                                 "TURISMO",
                                 if_else(str_detect(DESC_SECCION,␣
 ↪"INFR"),
```

```
                                                                    ␣
↪"INFRAESTRUCTURAS",
                                                                     "RESTO")))) |>
 group_by(grupo, anyo) |>
 summarise(inversion = sum(inversion), .groups = "drop")
```

[32]:
```
tdata_out |> glimpse()
```

```
Rows: 24
Columns: 3
$ grupo     <chr> "INFRAESTRUCTURAS", "INFRAESTRUCTURAS",
"INFRAESTRUCTURAS", …
$ anyo      <chr> "2017", "2018", "2019", "2020", "2021",
"2022", "2017", "201…
$ inversion <dbl> 1055334365, 1208736964, 1635747073,
797650078, 910113153, 11…
```

[33]:
```
tdata_out |> slice_head(n = 5)
```

A tibble: 5 x 3

grupo <chr>	anyo <chr>	inversion <dbl>
INFRAESTRUCTURAS	2017	1055334365
INFRAESTRUCTURAS	2018	1208736964
INFRAESTRUCTURAS	2019	1635747073
INFRAESTRUCTURAS	2020	797650078
INFRAESTRUCTURAS	2021	910113153

0.4 Synthetic Data Generation

No aplica

0.5 Fake Data Generation

No aplica

0.6 Open Data

Los datos de entrada se descargaron 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

[34]:
```
data_to_save <- tdata_out
```

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

[35]:
```
caso <- "CU_53"
proceso <- '_05'
tarea <- "_05"
archivo <- ""
proper <- "_inversiones_cm"
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/.

[36]:
```
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_53_05_05_inversiones_cm.csv'

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

[37]:
```
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*)
- Los archivos descargados deben estar en Data/Input/xls
- Los paquetes purrr, readxl, readr, dplyr, stringr, tidyr 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 * purrr 1.0.1 * readxl 1.4.1 * readr 2.1.3 * dplyr 1.0.10 * stringr 1.5.0 * tidyr 1.3.0

0.8.3 Data structures

Objeto `tdata_out`

- Hay 24 filas con información de las siguientes variables:
  - grupo
  - anyo
  - inversion

Observaciones generales sobre los datos

- Los datos no incluyen la ejecución de diciembre porque no hay datos consistentes a lo largo de los años. Se podría estudiar cómo completar este dato.

0.8.4 Consideraciones para despliegue en piloto

- No aplica

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 consolidado los datos de ejecución presupuestaria en un archivo

- Se han normalizado las descripciones de las categorías para poder agrupar

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

- Se deben relacionar las inversiones con el SPI
- Como no hay datos relacionados directamente, hay que buscar qué indicadores están más correlacionados y hacer un modelo de regresión con aquellos que se puedan tener de la Comunidad de Madrid. De ahí hacer predicciones y simulaciones.

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
```