

05. - Data Collection_CU_04_13_búsquedas_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 13. Importar datos de tendencias de búsquedas de Google

- Importar los datos de gripe obtenidos en xlsx y guardar en csv

Ver notas al final para el uso en piloto y producción

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

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

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

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

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

```
[2]: 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.

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

```
[3]: # file_data <- 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)

```
[4]: iFile <- "GoogleTrendsCAMInteres.xlsx"
file_data <- paste0(iPath, iFile)

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

Se leerán datos del archivo: Data/Input/GoogleTrendsCAMInteres.xlsx

Data file to dataframe

```
[5]: data <- read_excel(file_data)
```

Estructura de los datos:

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

```
Rows: 74
Columns: 4
$ fecha      <dtm> 2021-09-05, 2021-09-12, 2021-09-19,
2021-09-26, 2021~
$ ano        <dbl> 2021, 2021, 2021, 2021, 2021, 2021,
2021, 2021, 2021, ~
$ semana     <dbl> 36, 37, 38, 39, 40, 41, 42, 43, 44,
45, 46, 47, 48, 4~
$ `gripe %interes` <dbl> 13, 15, 19, 29, 38, 65, 100, 94, 63,
70, 64, 64, 57, ~
```

Muestra de datos:

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

	fecha	ano	semana	gripe %interes
	<dtm>	<dbl>	<dbl>	<dbl>
A tibble: 5 x 4	2021-09-05	2021	36	13
	2021-09-12	2021	37	15
	2021-09-19	2021	38	19
	2021-09-26	2021	39	29
	2021-10-03	2021	40	38

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:

- Tendencias semanales gripe

Data Extract

- Omitir columnas irrelevantes

Data transform & data extract

- Renombrar nombres de columna

```
[8]: edata <- data |>
      select(2:4)
```

```
[9]: tdata <- edata |>
      rename(interres_gripe = `gripe %interes`)
```

```
[10]: tdata |> glimpse()
```

```
Rows: 74
Columns: 3
$ ano          <dbl> 2021, 2021, 2021, 2021, 2021, 2021,
2021, 2021, 2021, 20~
$ semana       <dbl> 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,
46, 47, 48, 49, ~
$ interes_gripe <dbl> 13, 15, 19, 29, 38, 65, 100, 94, 63,
70, 64, 64, 57, 67,~
```

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

	ano	semana	interes_gripe
	<dbl>	<dbl>	<dbl>
	2021	36	13
A tibble: 5 x 3	2021	37	15
	2021	38	19
	2021	39	29
	2021	40	38

0.4 Synthetic Data Generation

No aplica

0.5 Fake Data Generation

No aplica

0.6 Open Data

No aplica

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

```
[12]: 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

```
[13]: caso <- "CU_04"  
proceso <- '_05'  
tarea <- "_13"  
archivo <- ""  
proper <- "_interes_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

```
[14]: # 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/.

```
[15]: 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_13_interes_gripe.csv'

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

```
[16]: 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 readr, dplyr, readxl 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 * readr 2.1.3 * dplyr 1.0.10 * readxl 1.4.1

0.8.3 Data structures

Objeto tdata

- Hay 74 filas con las variables:
 - ano
 - semana
 - interes_gripe

Observaciones generales sobre los datos

- Los recuentos están agregados para la Comunidad de Madrid

0.8.4 Consideraciones para despliegue en piloto

- Para el piloto se utilizan los datos estáticos

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 usado nombres de columna estándar
- Se ha omitido una columna innecesaria
- Se han guardado los datos en csv

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

- Se deben unir estos datos a los datos de vacunación, teniendo en cuenta que están agregados para toda la comunidad

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
```