05. - Data Collection_CU_25_04_lista_espera_v_01

June 10, 2023

#

CU25_Modelo de gestión de Lista de Espera Quirúrgica

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 04. Homogeneizar datos del CSV descargado de listas de espera

- Valores perdidos como vacíos
- Datos de semanas como número

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 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

[33]:
```
library(readr)
library(dplyr)
library(lubridate)

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

0.1.2 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.

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)

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

[4]:
```
iFile <- "listas_espera.csv"
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 leer<U+00E1>n datos del archivo:  Data/Input/listas_espera.csv
```

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

[5]:
```
data <- read_csv(file_data, na = "na")
```

```
New names:
* `` -> `...1`
```

```
Rows: 55680 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): Hospital, Especialidad, Semana
dbl (3): …1, Total pacientes, Media tiempo (dias)

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

Estructura de los datos:

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

```
Rows: 55,680
Columns: 6
$ …1                  <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14~
$ Hospital              <chr> "HOSPITAL CENTRAL DE LA CRUZ
ROJA SAN JOSE Y SAN~
$ Especialidad          <chr> "Angiolog<U+00ED>a y
Cirug<U+00ED>a Vascular", "Angiolog<U+00ED>a y C~
$ Semana                <chr> "SEMANAL - 04/11/2022",
"SEMANAL - 28/10/2022", ~
$ `Total pacientes`     <dbl> 439, 437, 429, 419, 418, NA,
408, 399, 396, 393,~
$ `Media tiempo (dias)` <dbl> 56.29, 55.13, 53.99, 56.29,
55.82, NA, 57.45, 57~
```

Muestra de datos:

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

A spec_tbl_df: 5 x 6

	...1	Hospital	H
	<dbl>	<chr>	<
0	0	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	A
1	1	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	A
2	2	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	A
3	3	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	A
4	4	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	A

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:

- Listas de espera obtenidas por webscrapping

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

0.3.2 Data extract & adequacy

- Quitar columna identificadora
- Quitar espacios a nombres de columnas

Por conveniencia al ser un archivo intermedio, se incluyen aquí tareas propias del proceso 06 *Data Adequacy*

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

[23]:
```
edata <- data |>
    select(-1) |>
    rename(total_pacientes = `Total pacientes`,
        media_tiempo_dias = `Media tiempo (dias)`)
```

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

```
Rows: 55,680
Columns: 5
$ Hospital          <chr> "HOSPITAL CENTRAL DE LA CRUZ ROJA
SAN JOSE Y SANTA A~
$ Especialidad      <chr> "Angiolog<U+00ED>a y Cirug<U+00ED>a
Vascular", "Angiolog<U+00ED>a y Cirug~
$ Semana            <chr> "SEMANAL - 04/11/2022", "SEMANAL -
28/10/2022", "SEM~
$ total_pacientes   <dbl> 439, 437, 429, 419, 418, NA, 408,
399, 396, 393, 378~
$ media_tiempo_dias <dbl> 56.29, 55.13, 53.99, 56.29, 55.82,
NA, 57.45, 57.61,~
```

Data transformation

- Extraer la semana y el año de la columna Semana, quitando columna tipo texto

[25]:
```
tdata <- edata |>
    mutate(ano = year(as.Date(Semana, format = "SEMANAL - %d/%m/%Y")),
        semana = isoweek(as.Date(Semana, format = "SEMANAL - %d/%m/%Y"))) |>
    select(-Semana)
```

Estructura de los datos a guardar

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

```
Rows: 55,680
Columns: 6
$ Hospital          <chr> "HOSPITAL CENTRAL DE LA CRUZ ROJA
SAN JOSE Y SANTA A~
$ Especialidad      <chr> "Angiolog<U+00ED>a y Cirug<U+00ED>a
```

```
Vascular", "Angiolog<U+00ED>a y Cirug~
$ total_pacientes   <dbl> 439, 437, 429, 419, 418, NA, 408,
399, 396, 393, 378~
$ media_tiempo_dias <dbl> 56.29, 55.13, 53.99, 56.29, 55.82,
NA, 57.45, 57.61,~
$ ano               <dbl> 2022, 2022, 2022, 2022, 2022, 2022,
2022, 2022, 2022~
$ semana            <dbl> 44, 43, 42, 41, 40, 39, 38, 37, 36,
35, 34, 33, 32, ~
```

Muestra de las primeras filas

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

A tibble: 5 x 6

Hospital <chr>	Especialidad <chr>
HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiolog<U+00
HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiolog<U+00
HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiolog<U+00
HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiolog<U+00
HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiolog<U+00

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 las cadenas añadida del fichero para diferenciarlas

Identificamos los datos a guardar

[28]:
```
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

[29]:
```
caso <- "CU_25"
proceso <- '_05'
tarea <- "_04"
archivo <- ""
proper <- "_lista_espera"
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/.

[30]:
```
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_25_05_04_lista_espera.csv’

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

[31]:
```
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, lubridate 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 * lubridate 1.9.1

0.8.3 Data structures

Objeto `data`

- Hay 55680 filas con informaciónn de las siguientes variables:
  - Hospital
  - Especialidad
  - total_pacientes
  - media_tiempo_dias
  - ano
  - semana

[36]:
```
cat("- Hay", nrow(data), "filas con información de las siguientes variables:
 ↪\n\t*", paste(colnames(data_to_save), collapse = "\n\t* "))
```

```
- Hay 55680 filas con informaci<U+00F3>n de las siguientes variables:
        * Hospital
        * Especialidad
        * total_pacientes
        * media_tiempo_dias
        * ano
        * semana
```

Observaciones generales sobre los datos

- Se toma la semana según norma ISO 8601.

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 extraído las semanas de la información tipo carácter obtenida
- Se han estandarizado los nombres de columna

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

- Se deben unir los datos de lista de espera a la geolocalización y a los datos del catálogo nacional de hospitales

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
```