

05. - Data Collection_CU_25_07_datos_sanitarios_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 07. Obtener un fichero con los datos sanitarios relevantes para las listas de espera.

- Los datos buenos habría que pedirlos al SNS. Mientras tanto, este archivo los simula en función de los datos existentes.
- En este notebook se guarda también un fichero de datos completo

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(readr)
library(dplyr)
library(tidyr)

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

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

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)

1. Listas de espera para simular una capacidad acorde

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

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

OPCION B: Especificar el nombre de archivo

```
[5]: iFile_01 <- "CU_25_05_05_02_lista_espera.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/CU_25_05_05_02_lista_espera.csv

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

```
[6]: data_01 <- read_csv(file_data_01)
```

Rows: 55680 Columns: 7
Column specification

Delimiter: ","

chr (2): Hospital, Especialidad

dbl (5): total_pacientes, media_tiempo_dias, ano, semana, CODCNH

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:

```
[7]: data_01 |> glimpse()
```

Rows: 55,680

Columns: 7

\$ Hospital <chr> "HOSPITAL CENTRAL DE LA CRUZ ROJA"

```

SAN JOSE Y SANTA A...
$ Especialidad      <chr> "Angiología y Cirugía Vascular",
"Angiología 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, ...
$ CODCNH           <dbl> 280148, 280148, 280148, 280148,
280148, 280148, 2801...

```

Muestra de datos:

```
[8]: data_01 |> slice_head(n = 5)
```

	Hospital <chr>	Especialidad <chr>
A spec_tbl_df: 5 x 7	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiología
	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiología
	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiología
	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiología
	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiología

2. Catálogo de hospitales incluyendo área

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

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

OPCION B: Especificar el nombre de archivo

```
[10]: iFile_02 <- "CU_25_05_05_01_hospitales.csv"
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/CU_25_05_05_01_hospitales.csv

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

```
[11]: data_02 <- read_csv(file_data_02)
```

Rows: 87 Columns: 24
Column specification

Delimiter: ","

chr (6): Hospital, id_area, nombre_area, Municipio, Clase,
Dependencia

dbl (18): CODCNH, cmunicipio, CAMAS, TAC, RM, GAM, HEM, ASD, LIT, BCO,
ALI, ...

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:

```
[12]: data_02 |> glimpse()
```

```
Rows: 87
Columns: 24
$ CODCNH      <dbl> 281371, 280745, 281124, 281071, 281292,
281281, 281090, 28...
$ Hospital    <chr> "HOSPITAL HM VALLES", "HOSPITAL
UNIVERSITARIO PRINCIPE DE ...
$ id_area     <chr> "03", "03", "08", "08", "11", "01", "06",
"06", "06", "11"...
$ nombre_area <chr> "Este", "Este", "Sur-Oeste I", "Sur-Oeste
I", "Sur Ii", "S...
$ cmunicipio  <dbl> 280053, 280053, 280072, 280072, 280133,
280148, 280225, 28...
$ Municipio   <chr> "Alcalá de Henares", "Alcalá de Henares",
"Alcorcón", "Alc...
$ CAMAS       <dbl> 20, 507, 81, 402, 98, 132, 170, 53, 192,
339, 344, 217, 13...
$ Clase       <chr> "Otros Centros con Internamiento",
"Hospitales Generales",...
$ Dependencia <chr> "PRIVADOS", "SERVICIOS E INSTITUTOS DE
SALUD DE LAS COMUNI...
$ TAC         <dbl> 1, 3, 1, 2, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1,
1, 2, 0, 0, 0, 0...
$ RM          <dbl> 1, 2, 2, 2, 1, 1, 3, 2, 0, 0, 0, 2, 2, 1,
2, 2, 0, 0, 0, 0...
$ GAM         <dbl> 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0...
$ HEM         <dbl> 0, 1, 1, 1, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0,
1, 1, 0, 0, 0, 0...
$ ASD         <dbl> 0, 1, 1, 1, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0,
1, 1, 0, 0, 0, 0...
$ LIT         <dbl> 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
```

```

0, 0, 0, 0, 0, 0...
$ BCO          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
$ ALI          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
2, 0, 0, 0, 0, 0...
$ SPECT        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
1, 2, 0, 0, 0, 0...
$ PET          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 0, 0, 0, 0...
$ MAMOS        <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 0, 0, 0, 1, 0, 2,
1, 1, 0, 0, 0, 0...
$ DO           <dbl> 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0,
1, 0, 0, 0, 0, 0...
$ DIAL         <dbl> 0, 28, 0, 28, 13, 18, 1, 0, 0, 0, 0, 18,
0, 25, 0, 24, 0, ...
$ X            <dbl> -3.367372, -3.347602, -3.849035,
-3.837394, -3.610795, -3...
$ Y            <dbl> 40.48302, 40.51025, 40.34248, 40.34887,
40.05726, 40.29697...

```

Muestra de datos:

```
[13]: data_02 |> slice_head(n = 5)
```

	CODCNH <dbl>	Hospital <chr>	id_area <chr>	nom <chr>
	281371	HOSPITAL HM VALLES	03	Est
A spec_tbl_df: 5 x 24	280745	HOSPITAL UNIVERSITARIO PRINCIPE DE ASTURIAS	03	Est
	281124	HOSPITAL QUIRONSAUD SUR	08	Sur
	281071	HOSPITAL UNIVERSITARIO FUNDACION ALCORCON	08	Sur
	281292	HOSPITAL DEL TAJO S.A.	11	Sur

3. Indicadores áreas

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

```
[14]: # file_data_03 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

```
[15]: iFile_03 <- "CU_25_05_06_indicadores_area.csv"
file_data_03 <- paste0(iPath, iFile_03)

if(file.exists(file_data_03)){
  cat("Se leerán datos del archivo: ", file_data_03)
} else{
  warning("Cuidado: el archivo no existe.")
}
```

Se leerán datos del archivo: Data/Input/CU_25_05_06_indicadores_area.csv

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

```
[16]: data_03 <- read_csv(file_data_03)
```

Rows: 11 Columns: 17

Column specification

Delimiter: ","

chr (2): id_area, nombre_area

dbl (15): t3_1, t1_1, t2_1, t2_2, 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:

```
[17]: data_03 |> glimpse()
```

Rows: 11

Columns: 17

\$ id_area <chr> "01", "02", "03", "04", "05", "06", "07", "08", "09", "10"...

\$ nombre_area <chr> "Sur-Este", "Centro-Norte", "Este", "Noreste", "Norte", "0"...

\$ t3_1 <dbl> 41.89330, 44.85910, 41.16384, 43.50412, 41.38840, 40.89333...

\$ t1_1 <dbl> 862340, 404085, 383079, 615157, 912485, 763506, 511605, 53...

\$ t2_1 <dbl> 0.5207432, 0.5419660, 0.5091281, 0.5321335, 0.5202002, 0.5...

\$ t2_2 <dbl> 0.4792568, 0.4580340, 0.4908719, 0.4678665, 0.4791324, 0.4...

\$ t4_1 <dbl> 0.1585425, 0.1293202, 0.1671827, 0.1495244, 0.1700457, 0.1...

\$ t4_2 <dbl> 0.6732152, 0.6456867, 0.6697046, 0.6530478, 0.6590844, 0.6...

\$ t4_3 <dbl> 0.1682410, 0.2249913, 0.1631133, 0.1974297, 0.1702033, 0.1...

\$ t5_1 <dbl> 0.1399664, 0.1279791, 0.1520691, 0.1266269, 0.1248427, 0.1...

\$ t6_1 <dbl> 0.2042778, 0.1854376, 0.1903965, 0.2132414, 0.1950003, 0.1...

\$ t7_1 <dbl> 0.05120722, 0.05930819, 0.05762767,

```
0.05909755, 0.06406634...
$ t8_1      <dbl> 0.03895576, 0.05200647, 0.04557209,
0.04845935, 0.05318346...
$ t9_1      <dbl> 0.3600324, 0.5572380, 0.3196460,
0.4600102, 0.4872258, 0.5...
$ t10_1     <dbl> 0.14408131, 0.09486110, 0.14520694,
0.11547830, 0.10328147...
$ t11_1     <dbl> 0.5193163, 0.5069128, 0.5250289,
0.5117620, 0.5420084, 0.5...
$ t12_1     <dbl> 0.6047756, 0.5605799, 0.6109222,
0.5773072, 0.6035461, 0.5...
```

Muestra de datos:

```
[18]: data_03 |> slice_head(n = 5)
```

	id_area	nombre_area	t3_1	t1_1	t2_1	t2_2	t4_1	t4_2
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
A spec_tbl_df: 5 x 17	01	Sur-Este	41.89330	862340	0.5207432	0.4792568	0.1585425	0.6732
	02	Centro-Norte	44.85910	404085	0.5419660	0.4580340	0.1293202	0.6456
	03	Este	41.16384	383079	0.5091281	0.4908719	0.1671827	0.6697
	04	Noreste	43.50412	615157	0.5321335	0.4678665	0.1495244	0.6530
	05	Norte	41.38840	912485	0.5202002	0.4791324	0.1700457	0.6590

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 reales
- Catálogo de hospitales

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.

Data transformation

- Unión de tablas y agrupar por área.

```
[19]: tdata_01 <- data_01 |>
      inner_join(data_02 |> select(CODCNH, id_area, nombre_area), by = "CODCNH")
      ↪|>
      group_by(id_area, nombre_area, Hospital, Especialidad) |>
      summarise(total_pacientes = mean(total_pacientes, na.rm = TRUE),
                media_tiempo_dias = mean(media_tiempo_dias, na.rm = TRUE)) |>
      ungroup()
```


`summarise()` has grouped output by 'id_area', 'nombre_area', 'Hospital'. You can override using the `.groups` argument.

```
[20]: tdata_01 |> glimpse()
```

```
Rows: 464
Columns: 6
$ id_area      <chr> "01", "01", "01", "01", "01", "01",
"01", "01", "01"...
$ nombre_area  <chr> "Sur-Este", "Sur-Este", "Sur-Este",
"Sur-Este", "Sur...
$ Hospital     <chr> "HOSPITAL INFANTIL UNIVERSITARIO
NIÑO JESUS", "HOSPI...
$ Especialidad <chr> "Angiología y Cirugía Vascular",
"Cirugía Cardíaca",...
$ total_pacientes <dbl> 0.00000, 0.00000, 0.00000,
41.93277, 315.46218, 93.2...
$ media_tiempo_dias <dbl> 0.00000, 0.00000, 0.00000,
39.46882, 53.02748, 60.17...
```

```
[21]: tdata_01 |> slice_head(n = 5)
```

	id_area	nombre_area	Hospital	Especialidad
	<chr>	<chr>	<chr>	<chr>
A tibble: 5 x 6	01	Sur-Este	HOSPITAL INFANTIL UNIVERSITARIO NIÑO JESUS	Angiología
	01	Sur-Este	HOSPITAL INFANTIL UNIVERSITARIO NIÑO JESUS	Cirugía Ca
	01	Sur-Este	HOSPITAL INFANTIL UNIVERSITARIO NIÑO JESUS	Cirugía Ge
	01	Sur-Este	HOSPITAL INFANTIL UNIVERSITARIO NIÑO JESUS	Cirugía Or
	01	Sur-Este	HOSPITAL INFANTIL UNIVERSITARIO NIÑO JESUS	Cirugía Pe

0.4 Synthetic Data Generation

- Asignar capacidad
- Para cada hospital y especialidad, asignar la capacidad. Se asume la misma capacidad en el tiempo, personas / tiempo

```
[22]: sdata_01 <- tdata_01 |>
      mutate(capacidad = ceiling(total_pacientes/media_tiempo_dias)) |>
      replace_na(list(capacidad = 0))
```

```
[23]: sdata_01 |> glimpse()
```

```
Rows: 464
Columns: 7
$ id_area      <chr> "01", "01", "01", "01", "01", "01",
"01", "01", "01"...
$ nombre_area  <chr> "Sur-Este", "Sur-Este", "Sur-Este",
"Sur-Este", "Sur...
$ Hospital     <chr> "HOSPITAL INFANTIL UNIVERSITARIO
```

```

NIÑO JESUS", "HOSPI...
$ Especialidad      <chr> "Angiología y Cirugía Vascular",
"Cirugía Cardíaca",...
$ total_pacientes   <dbl> 0.00000, 0.00000, 0.00000,
41.93277, 315.46218, 93.2...
$ media_tiempo_dias <dbl> 0.00000, 0.00000, 0.00000,
39.46882, 53.02748, 60.17...
$ capacidad         <dbl> 0, 0, 0, 2, 6, 2, 0, 0, 0, 1, 4, 2,
3, 23, 5, 3, 0, ...

```

```
[24]: sdata_01 |> slice_head(n = 5)
```

	id_area	nombre_area	Hospital	Especialidad
	<chr>	<chr>	<chr>	<chr>
A tibble: 5 x 7	01	Sur-Este	HOSPITAL INFANTIL UNIVERSITARIO NIÑO JESUS	Angiología
	01	Sur-Este	HOSPITAL INFANTIL UNIVERSITARIO NIÑO JESUS	Cirugía Ca
	01	Sur-Este	HOSPITAL INFANTIL UNIVERSITARIO NIÑO JESUS	Cirugía Ge
	01	Sur-Este	HOSPITAL INFANTIL UNIVERSITARIO NIÑO JESUS	Cirugía Or
	01	Sur-Este	HOSPITAL INFANTIL UNIVERSITARIO NIÑO JESUS	Cirugía Pe

0.4.1 Data transformation

- Agrupar capacidad por área

```
[25]: tdata_01_area <- sdata_01 |>
  group_by(id_area, nombre_area, Especialidad) |>
  summarise(capacidad = sum(capacidad)) |>
  ungroup()
```

`summarise()` has grouped output by 'id_area', 'nombre_area'. You can override using the `.groups` argument.

```
[26]: tdata_01_area |> glimpse()
```

```

Rows: 160
Columns: 4
$ id_area      <chr> "01", "01", "01", "01", "01", "01",
"01", "01", "01", "01...
$ nombre_area  <chr> "Sur-Este", "Sur-Este", "Sur-Este",
"Sur-Este", "Sur-Este...
$ Especialidad <chr> "Angiología y Cirugía Vascular",
"Cirugía Cardíaca", "Cir...
$ capacidad    <dbl> 0, 0, 20, 2, 6, 2, 0, 9, 6, 1, 4, 27,
11, 99, 18, 14, 6, ...

```

```
[27]: tdata_01_area |> slice_head(n = 5)
```

	id_area <chr>	nombre_area <chr>	Especialidad <chr>	capacidad <dbl>
A tibble: 5 x 4	01	Sur-Este	Angiología y Cirugía Vascular	0
	01	Sur-Este	Cirugía Cardíaca	0
	01	Sur-Este	Cirugía General y del Aparato Digestivo	20
	01	Sur-Este	Cirugía Oral y Maxilofacial	2
	01	Sur-Este	Cirugía Pediátrica General	6

0.5 Fake Data Generation

Proceso de simulación de pacientes, consultas y hospitalizaciones:

1. Unión de lista con indicadores
2. Simulación de pacientes teniendo en cuenta tamaño de la cola y población del área
3. Simulación de consultas teniendo en cuenta el tamaño de la cola
4. Simulación de hospitalizaciones teniendo en cuenta el tamaño de la cola
5. Quitar información de indicadores

```
[28]: set.seed(1)
      tdata_03 <- data_01 |>
        inner_join(data_02 |> select(CODCNH, id_area, nombre_area), by = c(
          ↪c("CODCNH")) |>
        inner_join(data_03, by = c("id_area", "nombre_area")) |>
        mutate(pacientes = ceiling((max(c(total_pacientes, 10), na.rm = TRUE)/
          ↪t1_1)*(1+rnorm(n(), 0.6, 0.1))*t1_1*0.1),
              consultas = ceiling(total_pacientes*1.5 + sample(c(-1, 1),
          ↪1)*ceiling(rnorm(n(), 50, 10)))),
              hospitalizaciones = ceiling(total_pacientes*runif(n(), 0, 0.05))) |>
        select(-(t3_1:t12_1))
```

```
[29]: tdata_03 |> glimpse()
```

```
Rows: 55,680
Columns: 12
$ Hospital      <chr> "HOSPITAL CENTRAL DE LA CRUZ ROJA
SAN JOSE Y SANTA A...
$ Especialidad  <chr> "Angiología y Cirugía Vascular",
"Angiología 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, ...
$ CODCNH       <dbl> 280148, 280148, 280148, 280148,
280148, 280148, 2801...
$ id_area      <chr> "05", "05", "05", "05", "05", "05",
```

```
"05", "05", "05"...
$ nombre_area      <chr> "Norte", "Norte", "Norte", "Norte",
"Norte", "Norte"...
$ pacientes         <dbl> 1392, 1465, 1373, 1593, 1478, 1374,
1493, 1515, 1501...
$ consultas         <dbl> 717, 703, 684, 686, 682, NA, 656,
657, 646, 641, 625...
$ hospitalizaciones <dbl> 8, 3, 12, 3, 17, NA, 10, 20, 14, 8,
19, 8, 18, 6, 5,...
```

0.5.1 Data Extract

- Consolidar todos los datos en una sola tabla

```
[30]: tdata_03 |> select(CODCNH, pacientes:hospitalizaciones, ano, semana)
```

	CODCNH <dbl>	pacientes <dbl>	consultas <dbl>	hospitalizaciones <dbl>	ano <dbl>	semana <dbl>
	280148	1392	717	8	2022	44
	280148	1465	703	3	2022	43
	280148	1373	684	12	2022	42
	280148	1593	686	3	2022	41
	280148	1478	682	17	2022	40
	280148	1374	NA	NA	2022	39
	280148	1493	656	10	2022	38
	280148	1515	657	20	2022	37
	280148	1501	646	14	2022	36
	280148	1421	641	8	2022	35
	280148	1585	625	19	2022	34
	280148	1484	619	8	2022	33
	280148	1392	607	18	2022	32
	280148	1248	581	6	2022	31
	280148	1550	553	5	2022	30
	280148	1445	539	8	2022	29
	280148	1447	496	15	2022	28
	280148	1534	495	14	2022	27
	280148	1523	484	4	2022	25
	280148	1502	478	5	2022	24
	280148	1532	460	6	2022	23
	280148	1519	478	6	2022	22
	280148	1455	514	6	2022	21
	280148	1269	569	14	2022	20
	280148	1505	572	15	2022	19
	280148	1444	578	6	2022	18
	280148	1435	598	4	2022	17
	280148	1316	624	1	2022	16
	280148	1405	630	15	2022	15
A tibble: 55680 x 6	280148	1486	625	10	2022	14

	280838	1275	3742	62	2021	1
	280838	1445	4156	86	2020	51
	280838	1499	4128	104	2020	50
	280838	1396	4017	8	2020	49
	280838	1380	3932	109	2020	48
	280838	1566	3884	71	2020	47
	280838	1506	3855	56	2020	46
	280838	1317	3740	110	2020	45
	280838	1491	3729	81	2020	43
	280838	1439	3660	62	2020	42
	280838	1369	3588	23	2020	41
	280838	1556	3438	65	2020	40
	280838	1471	3331	64	2020	39
	280838	1488	3254	70	2020	38
	280838	1388	3225	100	2020	37
	280838	1434	3187	30	2020	36
	280838	1498	3263	22	2020	35
	280838	1620	3258	59	2020	34
	280838	1489	3254	33	2020	33
	280838	1408	3252	66	2020	32

```
[31]: a <- data_01 |>
      inner_join(data_02 |> select(-Hospital), by = c("CODCNH")) |>
      inner_join(data_03, by = c("id_area", "nombre_area")) |>
      inner_join(tdata_01_area, by = c("id_area", "nombre_area", "Especialidad"))
b <- tdata_03 |> select(CODCNH, pacientes:hospitalizaciones, ano, semana,
↳Especialidad)
names(a); nrow(a)
names(b); nrow(b)
c <- inner_join(a, b)
names(c); nrow(c)
nrow(c)/nrow(b)
length(unique(a$Especialidad))
```

1. 'Hospital' 2. 'Especialidad' 3. 'total_pacientes' 4. 'media_tiempo_dias' 5. 'ano' 6. 'semana' 7. 'CODCNH' 8. 'id_area' 9. 'nombre_area' 10. 'cmunicipio' 11. 'Municipio' 12. 'CAMAS' 13. 'Clase' 14. 'Dependencia' 15. 'TAC' 16. 'RM' 17. 'GAM' 18. 'HEM' 19. 'ASD' 20. 'LIT' 21. 'BCO' 22. 'ALI' 23. 'SPECT' 24. 'PET' 25. 'MAMOS' 26. 'DO' 27. 'DIAL' 28. 'X' 29. 'Y' 30. 't3_1' 31. 't1_1' 32. 't2_1' 33. 't2_2' 34. 't4_1' 35. 't4_2' 36. 't4_3' 37. 't5_1' 38. 't6_1' 39. 't7_1' 40. 't8_1' 41. 't9_1' 42. 't10_1' 43. 't11_1' 44. 't12_1' 45. 'capacidad'

55680

1. 'CODCNH' 2. 'pacientes' 3. 'consultas' 4. 'hospitalizaciones' 5. 'ano' 6. 'semana' 7. 'Especialidad'

55680

Joining, by = c("Especialidad", "ano", "semana", "CODCNH")

1. 'Hospital' 2. 'Especialidad' 3. 'total_pacientes' 4. 'media_tiempo_dias' 5. 'ano' 6. 'semana' 7. 'CODCNH' 8. 'id_area' 9. 'nombre_area' 10. 'cmunicipio' 11. 'Municipio' 12. 'CAMAS' 13. 'Clase' 14. 'Dependencia' 15. 'TAC' 16. 'RM' 17. 'GAM' 18. 'HEM' 19. 'ASD' 20. 'LIT' 21. 'BCO' 22. 'ALI' 23. 'SPECT' 24. 'PET' 25. 'MAMOS' 26. 'DO' 27. 'DIAL' 28. 'X' 29. 'Y' 30. 't3_1' 31. 't1_1' 32. 't2_1' 33. 't2_2' 34. 't4_1' 35. 't4_2' 36. 't4_3' 37. 't5_1' 38. 't6_1' 39. 't7_1' 40. 't8_1' 41. 't9_1' 42. 't10_1' 43. 't11_1' 44. 't12_1' 45. 'capacidad' 46. 'pacientes' 47. 'consultas' 48. 'hospitalizaciones'

55680

1

16

```
[32]: data_completo <- data_01 |>
      inner_join(data_02 |> select(-Hospital), by = c("CODCNH")) |>
      inner_join(data_03, by = c("id_area", "nombre_area")) |>
      inner_join(tdata_01_area, by = c("id_area", "nombre_area", "Especialidad"))
↳|>
      inner_join(tdata_03 |> select(CODCNH, Especialidad, ano, semana, pacientes:
↳hospitalizaciones),
      by = c("CODCNH", "Especialidad", "ano", "semana"))
```

```
[33]: data_completo |> glimpse()
```

```
Rows: 55,680
Columns: 48
$ Hospital      <chr> "HOSPITAL CENTRAL DE LA CRUZ ROJA
SAN JOSE Y SANTA A...
$ Especialidad  <chr> "Angiología y Cirugía Vascular",
"Angiología 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, ...
$ CODCNH       <dbl> 280148, 280148, 280148, 280148,
280148, 280148, 2801...
$ id_area      <chr> "05", "05", "05", "05", "05", "05",
"05", "05", "05"...
$ nombre_area   <chr> "Norte", "Norte", "Norte", "Norte",
"Norte", "Norte"...
$ cmunicipio   <dbl> 280796, 280796, 280796, 280796,
280796, 280796, 2807...
$ Municipio     <chr> "Madrid", "Madrid", "Madrid",
"Madrid", "Madrid", "M...
$ CAMAS        <dbl> 156, 156, 156, 156, 156, 156, 156,
156, 156, 156, 15...
$ Clase        <chr> "Otros Centros con Internamiento",
"Otros Centros co...
$ Dependencia   <chr> "SERVICIOS E INSTITUTOS DE SALUD DE
LAS COMUNIDADES ...
$ TAC          <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1...
$ RM           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
$ GAM          <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1...
$ HEM          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
$ ASD          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
$ LIT          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
$ BCD          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
$ ALI          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
```

```

$ SPECT      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
$ PET        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
$ MAMOS      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1...
$ DO         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
$ DIAL       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0...
$ X          <dbl> -3.706932, -3.706932, -3.706932,
-3.706932, -3.70693...
$ Y          <dbl> 40.44715, 40.44715, 40.44715,
40.44715, 40.44715, 40...
$ t3_1       <dbl> 41.3884, 41.3884, 41.3884, 41.3884,
41.3884, 41.3884...
$ t1_1       <dbl> 912485, 912485, 912485, 912485,
912485, 912485, 9124...
$ t2_1       <dbl> 0.5202002, 0.5202002, 0.5202002,
0.5202002, 0.520200...
$ t2_2       <dbl> 0.4791324, 0.4791324, 0.4791324,
0.4791324, 0.479132...
$ t4_1       <dbl> 0.1700457, 0.1700457, 0.1700457,
0.1700457, 0.170045...
$ t4_2       <dbl> 0.6590844, 0.6590844, 0.6590844,
0.6590844, 0.659084...
$ t4_3       <dbl> 0.1702033, 0.1702033, 0.1702033,
0.1702033, 0.170203...
$ t5_1       <dbl> 0.1248427, 0.1248427, 0.1248427,
0.1248427, 0.124842...
$ t6_1       <dbl> 0.1950003, 0.1950003, 0.1950003,
0.1950003, 0.195000...
$ t7_1       <dbl> 0.06406634, 0.06406634, 0.06406634,
0.06406634, 0.06...
$ t8_1       <dbl> 0.05318346, 0.05318346, 0.05318346,
0.05318346, 0.05...
$ t9_1       <dbl> 0.4872258, 0.4872258, 0.4872258,
0.4872258, 0.487225...
$ t10_1      <dbl> 0.1032815, 0.1032815, 0.1032815,
0.1032815, 0.103281...
$ t11_1      <dbl> 0.5420084, 0.5420084, 0.5420084,
0.5420084, 0.542008...
$ t12_1      <dbl> 0.6035461, 0.6035461, 0.6035461,
0.6035461, 0.603546...
$ capacidad  <dbl> 14, 14, 14, 14, 14, 14, 14, 14, 14, 14,
14, 14, 14, 14, ...
$ pacientes  <dbl> 1392, 1465, 1373, 1593, 1478, 1374,
1493, 1515, 1501...

```



```
$ consultas          <dbl> 717, 703, 684, 686, 682, NA, 656,
657, 646, 641, 625...
$ hospitalizaciones <dbl> 8, 3, 12, 3, 17, NA, 10, 20, 14, 8,
19, 8, 18, 6, 5,...
```

```
[34]: data_completo |> slice_head(n = 5)
```

	Hospital <chr>	Especiali <chr>
A spec_tbl_df: 5 x 48	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiolog
	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiolog
	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiolog
	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiolog
	HOSPITAL CENTRAL DE LA CRUZ ROJA SAN JOSE Y SANTA ADELA	Angiolog

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

1. Datos de capacidad (solo por área y especialidad)

Identificamos los datos a guardar

```
[35]: data_to_save_01 <- tdata_01_area
```

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

```
[36]: caso <- "CU_25"
proceso <- '_05'
tarea <- "_07"
archivo <- "_01"
proper <- "_capacidad"
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

```
[37]: # file_save_01 <- paste0(caso, proceso, tarea, tcltk::tkgetSaveFile(), 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_02
```

OPCION B: Especificar el nombre de archivo

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

```
[38]: 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_25_05_07_01_capacidad.csv'

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

```
[39]: path_in_01 <- paste0(iPath, file_save_01)
      file.copy(path_out_01, path_in_01, overwrite = TRUE)
```

TRUE

2. Listas de espera con datos sanitarios

Identificamos los datos a guardar

```
[40]: data_to_save_02 <- tdata_03
```

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

```
[41]: archivo <- "_02"
      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

```
[42]: # file_save_02 <- paste0(caso, proceso, tarea, tcltk::tkgetSaveFile(), 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
```

OPCION B: Especificar el nombre de archivo

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

```
[43]: 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_25_05_07_02_lista_espera.csv’

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

```
[44]: path_in_02 <- paste0(iPath, file_save_02)
      file.copy(path_out_02, path_in_02, overwrite = TRUE)
```

TRUE

3. Datos completos en un archivo

Identificamos los datos a guardar

```
[45]: data_to_save_03 <- data_completo
```

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

0.7.3 Proceso 05

```
[46]: archivo <- "_03"
      proper <- "_lista_espera_completo"
      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

```
[47]: # file_save_03 <- paste0(caso, proceso, tarea, tcltk::tkgetSaveFile(), proper,
      ↪extension)
      # path_out_03 <- paste0(oPath, file_save_03)
      # write_csv(data_to_save_03, path_out_03)

      # cat('File saved as: ')
      # path_out_03
```

OPCION B: Especificar el nombre de archivo

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

```
[48]: file_save_03 <- paste0(caso, proceso, tarea, archivo, proper, extension)
      path_out_03 <- paste0(oPath, file_save_03)
      write_csv(data_to_save_03, path_out_03)

      cat('File saved as: ')
      path_out_03
```

File saved as:

’Data/Output/CU_25_05_07_03_lista_espera_completo.csv’

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

```
[49]: path_in_03 <- paste0(iPath, file_save_03)
      file.copy(path_out_03, path_in_03, 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, 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 * tcltk 4.2.2 * readr 2.1.3 * dplyr 1.0.10 * tidyr 1.3.0

0.8.3 Data structures

Objeto tdata_01_area

- Hay 160 filas con información de las siguientes variables:
 - id_area
 - nombre_area
 - Especialidad
 - capacidad

Observaciones generales sobre los datos

- Se dispone de la capacidad global para cada especialidad en cada área
- Se entiende que la capacidad dependerá de los quirófanos/equipos de especialistas disponibles en los hospitales

Objeto tdata_03

- Hay 464 filas con información de las siguientes variables:
 - id_area
 - nombre_area
 - Hospital
 - Especialidad

- total_pacientes
- media_tiempo_dias
- capacidad
- pacientes
- consultas
- hospitalizaciones

```
[50]: cat("- Hay", nrow(data_to_save_02), "filas con información de las siguientes_
      ↪variables:\n\t*", paste(colnames(data_to_save_02), collapse = "\n\t* "))
```

- Hay 55680 filas con información de las siguientes variables:
 - * Hospital
 - * Especialidad
 - * total_pacientes
 - * media_tiempo_dias
 - * ano
 - * semana
 - * CODCNH
 - * id_area
 - * nombre_area
 - * pacientes
 - * consultas
 - * hospitalizaciones

Objeto data_completo

- Los datos con distinta granularidad se repiten
- Hay 55680 filas con información de las siguientes variables:
 - Hospital
 - Especialidad
 - total_pacientes
 - media_tiempo_dias
 - ano
 - semana
 - CODCNH
 - id_area
 - nombre_area
 - cmunicipio
 - Municipio
 - CAMAS
 - Clase
 - Dependencia
 - TAC
 - RM
 - GAM
 - HEM
 - ASD
 - LIT
 - BCO

- ALI
- SPECT
- PET
- MAMOS
- DO
- DIAL
- X
- Y
- t3_1
- t1_1
- t2_1
- t2_2
- t4_1
- t4_2
- t4_3
- t5_1
- t6_1
- t7_1
- t8_1
- t9_1
- t10_1
- t11_1
- t12_1
- capacidad
- pacientes
- consultas
- hospitalizaciones

Observaciones generales sobre los datos

- Se dispone de la capacidad global para cada especialidad en cada área
- Se entiende que la capacidad dependerá de los quirófanos/equipos de especialistas disponibles en los hospitales

0.8.4 Consideraciones para despliegue en piloto

- Los datos con distinta resolución se guardan en archivos distintos. Se han unido también los datos en una tabla completa repitiendo datos en filas cuando procede. A la hora de extraer estos datos hay que estar seguro de la operación de agregación a usar (media, suma, otra)

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
- Se ha asumido el área sanitaria como agregación de hospitales en todo el caso. En el despliegue en producción se podría cambiar modificando los notebooks de la forma adecuada

0.9 Main Actions

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

- Se han asignado capacidades por especialidad y zona
- Se han simulado datos sanitarios por zona y semana

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

- Se deben unir los datos a la tabla completa

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

[52]: `# incluir código`