

05. - Data Collection_CU_25_05_hospitales_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 05. Obtener lista de hospitales georeferenciados

- Existen dos fuentes:
 - Catálogo Nacional de Hospitales del SNS (sin referenciar). Archivo Hospitales_de_Espana.csv
 - Hospitales georeferenciados en ERSI. Archivo CNH_2022.xlsx
- Para tener los hospitales georeferenciados se combinan estas fuentes
- Los datos se han descargado a mano y se han copiado a la carpeta **Data/Input** desde los siguientes enlaces:
 - SNS: <https://www.sanidad.gob.es/ciudadanos/prestaciones/centrosServiciosSNS/hospitales/home.htm>
 - ESRI: <https://opendata.esri.es/datasets/ComunidadSIG::hospitales-de-espana/explore?location=39.565169%2C-5.955525%2C6.06>

NOTA: por motivos de eficiencia se realizan en este notebooks tareas propias de otros proceso del dominio II, en particular el proceso 06 *Data Adequacy*

Adicionalmente: se actualizan los nombres de hospitales de las listas de espera para que sean consistentes con la geolocalización y el catálogo nacional.

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(sf)
library(readxl)
library(readr)
library(dplyr)
library(stringr)

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

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

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. Hospitales geolocalizados en ESRI

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 <- "Hospitales_de_Espana.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/Hospitales_de_Espana.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: 924 Columns: 47
Column specification

Delimiter: ","
chr (24): CODCNH, NOMBRE, DIRECCION, CODMU, MUNICIPIOS, CODPROV,
PROVINCIAS,...

dbl (23): OBJECTID, TELEFONO, TELEFONO2, TELEFAX, NCAMAS, CODFI, CODPAT, COD...

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: 924
Columns: 47
$ OBJECTID      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14,...
$ CODCNH        <chr> "010035", "010040", "010053",
"010066", "01008...
$ NOMBRE        <chr> "RED DE SALUD MENTAL DE ARABA
(HOSPITAL PSIQUI...
$ DIRECCION     <chr> "ÁLAVA, 43", "BEATO TOMÁS DE
ZUMÁRRAGA, 10", "...
$ TELEFONO      <dbl> 945006555, 945140900,
945252500, 945142100, 94...
$ TELEFONO2     <dbl> NA, NA, NA, NA, 945006876,
945007007, 96759717...
$ TELEFAX       <dbl> 945006587, 945145709,
945279260, 945143318, 94...
$ CODMU         <chr> "010590", "010590", "010590",
"010590", "01031...
$ MUNICIPIOS    <chr> "Vitoria-Gasteiz", "Vitoria-
Gasteiz", "Vitoria...
$ CODPROV       <chr> "01", "01", "01", "01", "01",
"01", "02", "02"...
$ PROVINCIAS    <chr> "ÁLAVA", "ÁLAVA", "ÁLAVA",
"ÁLAVA", "ÁLAVA", "...
$ CODAUTO       <chr> "16", "16", "16", "16", "16",
"16", "08", "08"...
$ COMUNIDADES   <chr> "PAÍS VASCO", "PAÍS VASCO",
"PAÍS VASCO", "PAÍ...
$ CODPOSTAL     <chr> "01006", "01008", "01002",
"01007", "01309", "...
$ NCAMAS        <dbl> 207, 63, 26, 82, 63, 811,
705, 0, 42, 135, 65,...
$ CODFI         <dbl> 6, 12, 1, 13, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 6,...
$ FINALIDAD_ASISITENCIAL <chr> "PSIQUIÁTRICO", "MÉDICO-
QUIRÚRGICO", "GENERAL"...
$ CODPAT        <dbl> 6, 14, 14, 14, 6, 1, 1, 1,
```

14, 1, 14, 1, 6, 6,...
 \$ DEPENDENCIA_PATRIMONIAL <chr> "COMUNIDAD AUTÓNOMA",
 "PRIVADO NO BENÉFICO", "...
 \$ CODFU <dbl> 4, 23, 23, 23, 4, 4, 31, 31,
 23, 31, 23, 31, 3...
 \$ DEPENDENCIA_FUNCIONAL <chr> "SERVICIO VASCO DE SALUD-
 OSAKIDETZA", "PRIVADO...
 \$ ACREDOCENT <chr> "N", "N", "N", "N", "N", "S",
 "S", "S", "N", "...
 \$ ESCOMPLE <chr> "N", "N", "N", "N", "N", "N",
 "S", "N", "N", "...
 \$ FORCOMPLE <chr> "N", "N", "N", "N", "N", "N",
 "N", "S", "N", "...
 \$ CODIDCOM <chr> NA, NA, NA, NA, NA, NA, NA,
 "020019", NA, NA, ...
 \$ ALTA <chr> "N", "N", "N", "N", "N", "N",
 "N", "N", "N", "...
 \$ CERRADO <chr> "N", "N", "N", "N", "N", "N",
 "N", "N", "N", "...
 \$ CAPITAL <chr> "S", "S", "S", "S", "N", "S",
 "S", "S", "S", "...
 \$ CIERREFECH <chr> NA, NA, NA, NA, NA, NA, NA,
 NA, NA, NA, NA, NA...
 \$ CONCIERTO <chr> "N", "N", "S", "S", "N", "N",
 "N", "N", "S", "...
 \$ EMAIL <chr>
 "SECRETARIA.GERENCIA@osakidetza.eus", "diazdec...
 \$ TAC <dbl> 0, 1, 1, 0, 0, 4, 4, 0, 1, 1,
 1, 0, 1, 1, 0, 3...
 \$ RM <dbl> 0, 1, 2, 0, 0, 0, 1, 0, 2, 1,
 2, 0, 0, 0, 0, 2...
 \$ GAM <dbl> 0, 0, 0, 0, 0, 2, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0...
 \$ HEM <dbl> 0, 1, 0, 0, 0, 2, 1, 0, 0, 0,
 1, 0, 0, 0, 0, 2...
 \$ ASD <dbl> 0, 1, 0, 0, 0, 3, 1, 0, 0, 0,
 0, 0, 0, 0, 0, 5...
 \$ LIT <dbl> 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
 0, 0, 0, 0, 0, 1...
 \$ BCO <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0...
 \$ ALI <dbl> 0, 0, 0, 0, 0, 2, 2, 0, 0, 0,
 0, 0, 0, 0, 0, 0...
 \$ SPECT <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0...
 \$ PET <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0...
 \$ MAMOS <dbl> 0, 1, 1, 0, 0, 3, 2, 0, 0, 1,

```

1, 0, 2, 1, 0, 2...
$ D0 <dbl> 0, 1, 1, 0, 0, 1, 1, 0, 0, 0,
0, 0, 0, 0, 0, 1...
$ DIAL <dbl> 0, 1, 0, 0, 0, 57, 16, 0, 0,
0, 0, 0, 0, 0, 0,...
$ X <dbl> -2.67861197, -2.67664050,
-2.66803516, -2.6803...
$ Y <dbl> 42.83566, 42.84966, 42.84976,
42.83758, 42.575...
$ CalidadGeocodificacion <chr> "Manual", "PointAddress",
"PointAddress", "Poi...

```

Muestra de datos:

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

	OBJECTID	CODCNH	NOMBRE
	<dbl>	<chr>	<chr>
	1	010035	RED DE SALUD MENTAL DE ARABA (HOSPITAL PSIQU
A spec_tbl_df: 5 x 47	2	010040	HOSPITAL SAN JOSÉ
	3	010053	HOSPITAL QUIRÓNSALUD VITORIA
	4	010066	HOSPITAL DE CUIDADOS SAN ONOFRE, S.L. (HOSPITA
	5	010088	HOSPITAL DE LEZA

2. Hospitales del SNS

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 <- "CNH_2022.xlsx"
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/CNH_2022.xlsx

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

```
[11]: data_02 <- read_excel(file_data_02, sheet = "DIRECTORIO DE HOSPITALES")
```

Estructura de los datos:

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

```
Rows: 832
Columns: 22
$ CCN <chr> "1531000730", "1531000737",
"1531000733", "153...
$ CODCNH <chr> "310121", "310044", "310060",
"310057", "31007...
$ `Nombre Centro` <chr> "HOSPITAL GARCIA ORCOYEN",
"HOSPITAL SAN JUAN ...
$ Dirección <chr> "CALLE SANTA SORIA 22 -",
"CALLE BELOSO ALTO 3...
$ Teléfono <chr> "848435000", "948231800",
"948255400", "948140...
$ `Cód. Municipio` <chr> "310977", "312016", "312016",
"312016", "31201...
$ Municipio <chr> "Estella-Lizarra",
"Pamplona/Iruña", "Pamplona...
$ `Cód. Provincia` <chr> "31", "31", "31", "31", "31",
"31", "31", "31"...
$ Provincia <chr> "NAVARRA", "NAVARRA",
"NAVARRA", "NAVARRA", "N...
$ `Cód. CCAA` <chr> "15", "15", "15", "15", "15",
"15", "15", "15"...
$ CCAA <chr> "C. FORAL DE NAVARRA", "C.
FORAL DE NAVARRA", ...
$ `Código Postal` <chr> "31200", "31006", "31008",
"31014", "31006", "...
$ CAMAS <chr> "106", "256", "230", "237",
"72", "84", "1077"...
$ `Cód. Clase de Centro` <chr> "C11", "C11", "C11", "C14",
"C11", "C14", "C11...
$ `Clase de Centro` <chr> "Hospitales Generales",
"Hospitales Generales"...
$ `Cód. Dep. Funcional` <chr> "2", "20", "20", "20", "20",
"2", "2", "2", "2...
$ `Dependencia Funcional` <chr> "SERVICIOS E INSTITUTOS DE
SALUD DE LAS COMUNI...
$ `Forma parte Complejo` <chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "...
$ CODIDCOM <chr> NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA...
$ `Nombre del Complejo` <chr> NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA...
$ ALTA <chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "...
$ Email <chr>
"GERENCIA.AREASALUD.ESTELLA@NAVARRA.ES", "HSJD...
```

Muestra de datos:

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

	CCN <chr>	CODCNH <chr>	Nombre Centro <chr>	Dirección <chr>
A tibble: 5 x 22	1531000730	310121	HOSPITAL GARCIA ORCOYEN	CALLE SAN
	1531000737	310044	HOSPITAL SAN JUAN DE DIOS	CALLE BE
	1531000733	310060	CLINICA UNIVERSIDAD DE NAVARRA	AVENIDA I
	1531000736	310057	CLINICA PSIQUIATRICA PADRE MENNI	CALLE JOA
	1531000738	310076	CLINICA ARCANGEL SAN MIGUEL - PAMPLONA	CALLE BE

3. Lista de espera

OPCION A: Seleccionar fichero en ventana para mayor comodidad

Data load using the {tcltk} package. Uncomment 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_04_lista_espera.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_04_lista_espera.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: 55680 Columns: 6
Column specification

Delimiter: ","

chr (2): Hospital, Especialidad

dbl (4): total_pacientes, media_tiempo_dias, ano, semana

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: 55,680
Columns: 6
$ 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, ...
```

Muestra de datos:

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

	Hospital <chr>	Especialidad <chr>
A spec_tbl_df: 5 x 6	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

4. Áreas sanitarias

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

```
[19]: # file_data_04 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

```
[20]: iFile_04 <- "CU_25_05_03_areasgeo.json"
file_data_04 <- paste0(iPath, iFile_04)

if(file.exists(file_data_04)){
  cat("Se leerán datos del archivo: ", file_data_04)
} else{
  warning("Cuidado: el archivo no existe.")
}
```

Se leerán datos del archivo: Data/Input/CU_25_05_03_areasgeo.json

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

```
[21]: data_04 <- st_read(file_data_04,  
  options = "ENCODING=latin1") |>  
  st_set_crs(23030) |>  
  st_transform(4326)
```

options: ENCODING=latin1

Warning message in CPL_read_ogr(dsn, layer, query, as.character(options), quiet,
:

"GDAL Message 6: driver GeoJSON does not support open option ENCODING"

Reading layer `CU_25_05_03_areasgeo' from data source

`/Users/emilio.lcano/academico/gh_repos/__transferencia/citizenlab/CitizenLab-
Research-and-Development/casos_urjc/notebooks/II_data_processing/25_listas_esper
a/Data/Input/CU_25_05_03_areasgeo.json'

using driver `GeoJSON'

Simple feature collection with 11 features and 3 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: 365635.7 ymin: 4415272 xmax: 495484 ymax: 4557310

Geodetic CRS: WGS 84

Warning message:

"st_crs<- : replacing crs does not reproject data; use st_transform for that"

Estructura de los datos:

```
[22]: data_04 |> glimpse()
```

Rows: 11

Columns: 4

\$ CODBDT <int> 958285, 958286, 958287, 958288, 958289,
958290, 958291, 9582...

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

\$ DESBDT <chr> "Sur-Este", "Centro-Norte", "Este",
"Noreste", "Norte", "Oes...

\$ geometry <MULTIPOLYGON [°]> MULTIPOLYGON (((-3.423161
4..., MULTIPOLYGON ((...

Muestra de datos:

```
[23]: data_04 |> tibble() |> slice_head(n = 5)
```

	CODBDT <int>	GEOCODIGO <chr>	DESBDT <chr>	geometry <MULTIPOLYGON [°]>
A tibble: 5 x 4	958285	01	Sur-Este	MULTIPOLYGON (((-3.423161 4...
	958286	02	Centro-Norte	MULTIPOLYGON (((-3.499207 4...
	958287	03	Este	MULTIPOLYGON (((-3.323953 4...
	958288	04	Noreste	MULTIPOLYGON (((-3.653662 4...
	958289	05	Norte	MULTIPOLYGON (((-3.537981 4...

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:

- Catálogo nacional de hospitales
- Geolocalización hospitales en datos abiertos ESRI

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 transform

- Los datos del Excel del ministerio no tienen tildes
- Los datos de listas de espera de la CM no tienen tildes
- Los datos de ESRI tienen tildes

Quitamos las tildes del archivo de ESRI

```
[24]: tdata_01 <- data_01 |>
      mutate(NOMBRE = chartr("ÁÉÍÓÚ", "AEIOU", NOMBRE))
```

Análisis inconsistencias en nombres

Hospitales cuyos nombres no coinciden:

```
[25]: unique(data_03$Hospital)[!unique(data_03$Hospital) %in% tdata_01$NOMBRE] |>
      sort()
```

1. 'HOSPITAL DE EL ESCORIAL' 2. 'HOSPITAL FUNDACION JIMENEZ DIAZ' 3. 'HOSPITAL UNIVERSITARIO GENERAL DE VILLALBA' 4. 'HOSPITAL UNIVERSITARIO RAMON Y CAJAL' 5. 'HOSPITAL UNIVERSITARIO REY JUAN CARLOS'

Alternativas:

```
[26]: tdata_01 |>
      filter(str_detect(NOMBRE, "ESCORIAL|VILLALBA|JUAN CARLOS|JIMENEZ DIAZ|RAMON Y
↪CAJAL")) |>
      arrange(NOMBRE) |>
      pull(NOMBRE)
```

1. 'HOSPITAL EL ESCORIAL' 2. 'HOSPITAL GENERAL DE VILLALBA' 3. 'HOSPITAL RAMON Y CAJAL' 4. 'HOSPITAL REY JUAN CARLOS' 5. 'HOSPITAL UNIVERSITARIO FUNDACION JIMENEZ DIAZ'

- Los datos de ESRI contienen el código de hospital de CNH
- El nombre definitivo debería ser el de la base de datos oficial de CNH
- Asignamos nombres de ESRI provisionalmente a los de listas de espera

```
[27]: tdata_03 <- data_03 |>
      mutate(Hospital = recode(Hospital,
                                "HOSPITAL DE EL ESCORIAL" = "HOSPITAL EL ESCORIAL",
                                "HOSPITAL FUNDACION JIMENEZ DIAZ" = "HOSPITAL_
↪UNIVERSITARIO FUNDACION JIMENEZ DIAZ",
                                "HOSPITAL UNIVERSITARIO GENERAL DE VILLALBA" =_
↪"HOSPITAL GENERAL DE VILLALBA",
                                "HOSPITAL UNIVERSITARIO RAMON Y CAJAL" = "HOSPITAL_
↪RAMON Y CAJAL",
                                "HOSPITAL UNIVERSITARIO REY JUAN CARLOS" = "HOSPITAL_
↪REY JUAN CARLOS"))
```

- Asignar coordenadas a catálogo de hospitales que están en la Comunidad de Madrid
- Convertir número de camas en numérica

```
[28]: tdata_02 <- data_02 |>
      mutate(CAMAS = as.numeric(CAMAS)) |>
      filter(`Cód. Provincia` == "28") |>
      inner_join(tdata_01 |>
                  select(-NCAMAS, -CODIDCOM, -ALTA),
                  by = "CODCNH")
```

```
[29]: glimpse(tdata_02)
```

```
Rows: 87
Columns: 65
$ CCN                <chr> "1328020932", "1328000034",
"1328000070", "132...
$ CODCNH             <chr> "281371", "280745", "281124",
"281071", "28129...
$ `Nombre Centro`    <chr> "HOSPITAL HM VALLES",
"HOSPITAL UNIVERSITARIO ...
$ Dirección          <chr> "CALLE SANTIAGO 14",
"CARRETERA DE MECO S/N", ...
$ Teléfono           <chr> "918883262", "918878100",
"917810682", "916219...
$ `Cód. Municipio`   <chr> "280053", "280053", "280072",
"280072", "28013...
$ Municipio          <chr> "Alcalá de Henares", "Alcalá
de Henares", "Alc...
$ `Cód. Provincia`   <chr> "28", "28", "28", "28", "28",
```

"28", "28", "28"...
 \$ Provincia <chr> "MADRID", "MADRID", "MADRID",
 "MADRID", "MADRI...
 \$ `Cód. CCAA` <chr> "13", "13", "13", "13", "13",
 "13", "13", "13"...
 \$ CCAA <chr> "MADRID", "MADRID", "MADRID",
 "MADRID", "MADRI...
 \$ `Código Postal` <chr> "28801", "28805", "28922",
 "28922", "28300", "...
 \$ CAMAS <dbl> 20, 507, 81, 402, 98, 132,
 170, 53, 192, 339, ...
 \$ `Cód. Clase de Centro` <chr> "C190", "C11", "C11", "C11",
 "C11", "C11", "C1...
 \$ `Clase de Centro` <chr> "Otros Centros con
 Internamiento", "Hospitales...
 \$ `Cód. Dep. Funcional` <chr> "20", "2", "20", "2", "2",
 "2", "20", "20", "2...
 \$ `Dependencia Funcional` <chr> "PRIVADOS", "SERVICIOS E
 INSTITUTOS DE SALUD D...
 \$ `Forma parte Complejo` <chr> "N", "N", "N", "N", "N", "N",
 "N", "N", "N", "...
 \$ CODIDCOM <chr> NA, NA, NA, NA, NA, NA, NA,
 NA, NA, NA, NA, NA...
 \$ `Nombre del Complejo` <chr> NA, NA, NA, NA, NA, NA, NA,
 NA, NA, NA, NA, NA...
 \$ ALTA <chr> "N", "N", "N", "N", "N", "N",
 "N", "N", "N", "...
 \$ Email <chr> NA, NA, NA,
 "FHALCORCON@FHALCORCON.ES", NA, NA...
 \$ OBJECTID <dbl> 559, 516, 536, 532, 551, 550,
 534, 541, 517, 5...
 \$ NOMBRE <chr> "HM VALLES", "HOSPITAL
 UNIVERSITARIO PRINCIPE ...
 \$ DIRECCION <chr> "C/SANTIAGO, 14", "CTRA.
 ALCALÁ-MECO, S/N", "E...
 \$ TELEFONO <dbl> 917374353, 918878100,
 916496600, 916219400, 91...
 \$ TELEFONO2 <dbl> NA, NA, NA, NA, NA,
 918394000, NA, NA, NA, 646...
 \$ TELEFAX <dbl> 912805932, 918801825,
 916496604, 916219902, 91...
 \$ CODMU <chr> "280053", "280053", "280072",
 "280072", "28013...
 \$ MUNICIPIOS <chr> "Alcalá de Henares", "Alcalá
 de Henares", "Alc...
 \$ CODPROV <chr> "28", "28", "28", "28", "28",
 "28", "28", "28"...
 \$ PROVINCIAS <chr> "MADRID", "MADRID", "MADRID",

"MADRID", "MADRI...	
\$ CODAUTO	<chr> "13", "13", "13", "13", "13",
"13", "13", "13"...	
\$ COMUNIDADES	<chr> "MADRID", "MADRID", "MADRID",
"MADRID", "MADRI...	
\$ CODPOSTAL	<chr> "28801", "28805", "28922",
"28922", "28300", "...	
\$ CODFI	<dbl> 16, 1, 12, 1, 1, 1, 1, 1, 13,
6, 6, 1, 10, 1, ...	
\$ FINALIDAD_ASISITENCIAL	<chr> "OTRA FINALIDAD", "GENERAL",
"MÉDICO-QUIRÚRGIC...	
\$ CODPAT	<dbl> 14, 1, 14, 1, 6, 6, 14, 14,
1, 12, 12, 6, 10, ...	
\$ DEPENDENCIA_PATRIMONIAL	<chr> "PRIVADO NO BENÉFICO",
"SEGURIDAD SOCIAL", "PR...	
\$ CODFU	<dbl> 23, 34, 23, 34, 34, 34, 23,
23, 34, 21, 21, 34...	
\$ DEPENDENCIA_FUNCIONAL	<chr> "PRIVADO NO BENÉFICO",
"SERVICIO MADRILEÑO DE ...	
\$ ACREDOCENT	<chr> "S", "S", "N", "S", "N", "N",
"S", "N", "S", "...	
\$ ESCOMPLE	<chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "...	
\$ FORCOMPLE	<chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "...	
\$ CERRADO	<chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "...	
\$ CAPITAL	<chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "...	
\$ CIERREFECH	<chr> NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA...	
\$ CONCIERTO	<chr> "S", "N", "S", "N", "N", "N",
"S", "S", "N", "...	
\$ EMAIL	<chr> "fgcalleja@hnhospitales.com;
imarco@hnhospital...	
\$ TAC	<dbl> 1, 3, 1, 2, 1, 1, 1, 1, 0, 0,
0, 1, 1, 1, 1, 2...	
\$ RM	<dbl> 1, 2, 2, 2, 1, 1, 3, 2, 0, 0,
0, 2, 2, 1, 2, 2...	
\$ GAM	<dbl> 1, 1, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0...	
\$ HEM	<dbl> 0, 1, 1, 1, 0, 0, 2, 0, 0, 0,
0, 1, 0, 0, 1, 1...	
\$ ASD	<dbl> 0, 1, 1, 1, 1, 0, 2, 0, 0, 0,
0, 0, 0, 0, 1, 1...	
\$ LIT	<dbl> 0, 0, 0, 1, 0, 0, 1, 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...
$ ALI <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 2, 0...
$ SPECT <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 1, 2...
$ PET <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 1...
$ MAMOS <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 0, 0,
0, 1, 0, 2, 1, 1...
$ DO <dbl> 1, 1, 1, 1, 0, 0, 1, 0, 0, 0,
0, 1, 0, 0, 1, 0...
$ DIAL <dbl> 0, 28, 0, 28, 13, 18, 1, 0,
0, 0, 0, 18, 0, 25...
$ X <dbl> -3.367372, -3.347602,
-3.849035, -3.837394, -3...
$ Y <dbl> 40.48302, 40.51025, 40.34248,
40.34887, 40.057...
$ CalidadGeocodificacion <chr> "PointAddress", "Manual",
"StreetAddress", "Ma...

```

Asignación del área sanitaria a los hospitales. Primero usamos objetos sf para hacer el join por geometrías, y después eliminamos estas para tener un data.frame normal y guardar como csv.

```

[30]: tdata_01_sf <- tdata_02 |> st_as_sf(coords = c("X", "Y")) |> st_set_crs(4326)

tdata_01_area <- tdata_01_sf |>
  st_join(data_04 |>
    select(id_area = GEOCODIGO,
           nombre_area = DESBDT)) |>
  bind_cols(st_coordinates(tdata_01_sf)) |>
  st_drop_geometry()

```

```

[31]: glimpse(tdata_01_area)

```

```

Rows: 87
Columns: 67
$ CCN <chr> "1328020932", "1328000034",
"1328000070", "132...
$ CODCNH <chr> "281371", "280745", "281124",
"281071", "28129...
$ `Nombre Centro` <chr> "HOSPITAL HM VALLES",
"HOSPITAL UNIVERSITARIO ...
$ Dirección <chr> "CALLE SANTIAGO 14",
"CARRETERA DE MECO S/N", ...
$ Teléfono <chr> "918883262", "918878100",
"917810682", "916219...
$ `Cód. Municipio` <chr> "280053", "280053", "280072",
"280072", "28013...
$ Municipio <chr> "Alcalá de Henares", "Alcalá

```

de Henares", "Alc...

\$ `Cód. Provincia` <chr> "28", "28", "28", "28", "28",
"28", "28", "28"...

\$ Provincia <chr> "MADRID", "MADRID", "MADRID",
"MADRID", "MADRI..."

\$ `Cód. CCAA` <chr> "13", "13", "13", "13", "13",
"13", "13", "13"...

\$ CCAA <chr> "MADRID", "MADRID", "MADRID",
"MADRID", "MADRI..."

\$ `Código Postal` <chr> "28801", "28805", "28922",
"28922", "28300", "...

\$ CAMAS <dbl> 20, 507, 81, 402, 98, 132,
170, 53, 192, 339, ...

\$ `Cód. Clase de Centro` <chr> "C190", "C11", "C11", "C11",
"C11", "C11", "C1..."

\$ `Clase de Centro` <chr> "Otros Centros con
Internamiento", "Hospitales..."

\$ `Cód. Dep. Funcional` <chr> "20", "2", "20", "2", "2",
"2", "20", "20", "2"...

\$ `Dependencia Funcional` <chr> "PRIVADOS", "SERVICIOS E
INSTITUTOS DE SALUD D..."

\$ `Forma parte Complejo` <chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "..."

\$ CODIDCOM <chr> NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA...

\$ `Nombre del Complejo` <chr> NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA...

\$ ALTA <chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "..."

\$ Email <chr> NA, NA, NA,
"FHALCORCON@FHALCORCON.ES", NA, NA...

\$ OBJECTID <dbl> 559, 516, 536, 532, 551, 550,
534, 541, 517, 5...

\$ NOMBRE <chr> "HM VALLES", "HOSPITAL
UNIVERSITARIO PRINCIPE ...

\$ DIRECCION <chr> "C/SANTIAGO, 14", "CTRA.
ALCALÁ-MECO, S/N", "E..."

\$ TELEFONO <dbl> 917374353, 918878100,
916496600, 916219400, 91...

\$ TELEFONO2 <dbl> NA, NA, NA, NA, NA,
918394000, NA, NA, NA, 646...

\$ TELEFAX <dbl> 912805932, 918801825,
916496604, 916219902, 91...

\$ CODMU <chr> "280053", "280053", "280072",
"280072", "28013..."

\$ MUNICIPIOS <chr> "Alcalá de Henares", "Alcalá
de Henares", "Alc..."

\$ CODPROV <chr> "28", "28", "28", "28", "28",

"28", "28", "28"...	
\$ PROVINCIAS	<chr> "MADRID", "MADRID", "MADRID",
"MADRID", "MADRI...	
\$ CODAUTO	<chr> "13", "13", "13", "13", "13",
"13", "13", "13"...	
\$ COMUNIDADES	<chr> "MADRID", "MADRID", "MADRID",
"MADRID", "MADRI...	
\$ CODPOSTAL	<chr> "28801", "28805", "28922",
"28922", "28300", "...	
\$ CODFI	<dbl> 16, 1, 12, 1, 1, 1, 1, 1, 13,
6, 6, 1, 10, 1, ...	
\$ FINALIDAD_ASISITENCIAL	<chr> "OTRA FINALIDAD", "GENERAL",
"MÉDICO-QUIRÚRGIC...	
\$ CODPAT	<dbl> 14, 1, 14, 1, 6, 6, 14, 14,
1, 12, 12, 6, 10, ...	
\$ DEPENDENCIA_PATRIMONIAL	<chr> "PRIVADO NO BENÉFICO",
"SEGURIDAD SOCIAL", "PR...	
\$ CODFU	<dbl> 23, 34, 23, 34, 34, 34, 23,
23, 34, 21, 21, 34...	
\$ DEPENDENCIA_FUNCIONAL	<chr> "PRIVADO NO BENÉFICO",
"SERVICIO MADRILEÑO DE ...	
\$ ACREDOCENT	<chr> "S", "S", "N", "S", "N", "N",
"S", "N", "S", "...	
\$ ESCOMPLE	<chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "...	
\$ FORCOMPLE	<chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "...	
\$ CERRADO	<chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "...	
\$ CAPITAL	<chr> "N", "N", "N", "N", "N", "N",
"N", "N", "N", "...	
\$ CIERREFECH	<chr> NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA...	
\$ CONCIERTO	<chr> "S", "N", "S", "N", "N", "N",
"S", "S", "N", "...	
\$ EMAIL	<chr> "fgcalleja@hnhospitales.com;
imarco@hnhospital...	
\$ TAC	<dbl> 1, 3, 1, 2, 1, 1, 1, 1, 0, 0,
0, 1, 1, 1, 1, 2...	
\$ RM	<dbl> 1, 2, 2, 2, 1, 1, 3, 2, 0, 0,
0, 2, 2, 1, 2, 2...	
\$ GAM	<dbl> 1, 1, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0...	
\$ HEM	<dbl> 0, 1, 1, 1, 0, 0, 2, 0, 0, 0,
0, 1, 0, 0, 1, 1...	
\$ ASD	<dbl> 0, 1, 1, 1, 1, 0, 2, 0, 0, 0,
0, 0, 0, 0, 1, 1...	
\$ LIT	<dbl> 0, 0, 0, 1, 0, 0, 1, 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...
$ ALI <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 2, 0...
$ SPECT <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 1, 2...
$ PET <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 1...
$ MAMOS <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 0, 0,
0, 1, 0, 2, 1, 1...
$ DO <dbl> 1, 1, 1, 1, 0, 0, 1, 0, 0, 0,
0, 1, 0, 0, 1, 0...
$ DIAL <dbl> 0, 28, 0, 28, 13, 18, 1, 0,
0, 0, 0, 18, 0, 25...
$ CalidadGeocodificacion <chr> "PointAddress", "Manual",
"StreetAddress", "Ma...
$ id_area <chr> "03", "03", "08", "08", "11",
"01", "06", "06"...
$ nombre_area <chr> "Este", "Este", "Sur-Oeste
I", "Sur-Oeste I", ...
$ X <dbl> -3.367372, -3.347602,
-3.849035, -3.837394, -3...
$ Y <dbl> 40.48302, 40.51025, 40.34248,
40.34887, 40.057...

```

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

	CCN <chr>	CODCNH <chr>	Nombre Centro <chr>	Dirección <chr>
A tibble: 5 x 67	1328020932	281371	HOSPITAL HM VALLES	CALLI
	1328000034	280745	HOSPITAL UNIVERSITARIO PRINCIPE DE ASTURIAS	CARR
	1328000070	281124	HOSPITAL QUIRONSAUD SUR	CALLI
	1328000066	281071	HOSPITAL UNIVERSITARIO FUNDACION ALCORCON	CALLI
	1328000084	281292	HOSPITAL DEL TAJO S.A.	AVEN

- Añadir a las listas de espera el código del hospital
- Se mantiene el nombre de la descarga de lista de espera, pero los joins se harán siempre por código

```
[33]: tdata_03_codigos <- tdata_03 |>
      left_join(tdata_01 |> select(NOMBRE, CODCNH),
                by = c("Hospital" = "NOMBRE"))
```

```
[34]: glimpse(tdata_03_codigos)
```

```

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           <chr> "280148", "280148", "280148",
"280148", "280148", "2..."

```

```
[35]: tdata_03_codigos |> slice_head(n = 5)
```

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

- Homogeneizar datos de hospitales

```
[36]: names(tdata_02)
```

```

1. 'CCN' 2. 'CODCNH' 3. 'Nombre Centro' 4. 'Dirección' 5. 'Teléfono' 6. 'Cód. Municipio' 7. 'Municipio' 8. 'Cód. Provincia' 9. 'Provincia' 10. 'Cód. CCAA' 11. 'CCAA' 12. 'Código Postal' 13. 'CAMAS' 14. 'Cód. Clase de Centro' 15. 'Clase de Centro' 16. 'Cód. Dep. Funcional' 17. 'Dependencia Funcional' 18. 'Forma parte Complejo' 19. 'CODIDCOM' 20. 'Nombre del Complejo' 21. 'ALTA' 22. 'Email' 23. 'OBJECTID' 24. 'NOMBRE' 25. 'DIRECCION' 26. 'TELEFONO' 27. 'TELEFONO2' 28. 'TELEFAX' 29. 'CODMU' 30. 'MUNICIPIOS' 31. 'CODPROV' 32. 'PROVINCIAS' 33. 'CODAUTO' 34. 'COMUNIDADES' 35. 'CODPOSTAL' 36. 'CODFI' 37. 'FINALIDAD_ASISITENCIAL' 38. 'CODPAT' 39. 'DEPENDENCIA_PATRIMONIAL' 40. 'CODFU' 41. 'DEPENDENCIA_FUNCIONAL' 42. 'ACREDOCENT' 43. 'ESCOMPLE' 44. 'FORCOMPLE' 45. 'CERRADO' 46. 'CAPITAL' 47. 'CIERREFECH' 48. 'CONCIERTO' 49. 'EMAIL' 50. 'TAC' 51. 'RM' 52. 'GAM' 53. 'HEM' 54. 'ASD' 55. 'LIT' 56. 'BCO' 57. 'ALI' 58. 'SPECT' 59. 'PET' 60. 'MAMOS' 61. 'DO' 62. 'DIAL' 63. 'X' 64. 'Y' 65. 'CalidadGeocodificacion'

```

```

[37]: tdata_02_clean <- tdata_02 |>
      left_join(tdata_01_area |>
                select(CODCNH, id_area, nombre_area),
                by = "CODCNH") |>
      select(CODCNH,
             Hospital = `Nombre Centro`,
             id_area,
             nombre_area,

```

```

cmunicipio = `Cód. Municipio`,
Municipio,
CAMAS,
Clase = `Clase de Centro`,
Dependencia = `Dependencia Funcional`,
TAC:Y)

```

```
[38]: glimpse(tdata_02_clean)
```

```

Rows: 87
Columns: 24
$ CODCNH      <chr> "281371", "280745", "281124", "281071",
"281292", "281281"...
$ 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  <chr> "280053", "280053", "280072", "280072",
"280133", "280148"...
$ 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...
$ ALI         <dbl> 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, 0, 1, 2, 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, 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...
```

```
[39]: tdata_02_clean |> slice_head(n = 5)
```

	CODCNH	Hospital	id_area	nombre_a
	<chr>	<chr>	<chr>	<chr>
	281371	HOSPITAL HM VALLES	03	Este
A tibble: 5 x 24	280745	HOSPITAL UNIVERSITARIO PRINCIPE DE ASTURIAS	03	Este
	281124	HOSPITAL QUIRONSAUD SUR	08	Sur-Oeste
	281071	HOSPITAL UNIVERSITARIO FUNDACION ALCORCON	08	Sur-Oeste
	281292	HOSPITAL DEL TAJO S.A.	11	Sur Ii

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

1. Datos de hospitales

Identificamos los datos a guardar

```
[40]: data_to_save_01 <- tdata_02_clean
```

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

```
[41]: caso <- "CU_25"
      proceso <- '_05'
      tarea <- "_05"
      archivo <- "_01"
      proper <- "_hospitales"
      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_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
```

OPCION B: Especificar el nombre de archivo

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

```
[43]: 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_05_01_hospitales.csv'

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

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

TRUE

2. Lista de espera (actualizada con código de hospital)

Identificamos los datos a guardar

```
[45]: data_to_save_02 <- tdata_03_codigos
```

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

```
[46]: 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

```
[47]: # 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
```

OPCION B: Especificar el nombre de archivo

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

```
[48]: 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_05_02_lista_espera.csv'
```

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

```
[49]: path_in_02 <- paste0(iPath, file_save_02)
      file.copy(path_out_02, path_in_02, 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 tcltk, sf, readr, dplyr, stringr, 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 * sf 1.0.9 * readr 2.1.3 * dplyr 1.0.10 * stringr 1.5.0 * readxl 1.4.1

0.8.3 Data structures

Objeto `tdata_02_clean`

- Los datos de origen son del catálogo nacional de hospitales a los que se ha añadido las coordenadas y el área de salud
- Hay 87 filas con información de las siguientes variables:
 - CODCNH
 - Hospital
 - id_area
 - nombre_area
 - cmunicipio
 - Municipio
 - CAMAS
 - Clase
 - Dependencia
 - TAC
 - RM
 - GAM

- HEM
- ASD
- LIT
- BCO
- ALI
- SPECT
- PET
- MAMOS
- DO
- DIAL
- X
- Y

Objeto tdata_03_codigos

- Son los datos de la lista de espera a los que se ha añadido el código de hospital
- Hay 55680 filas con información de las siguientes variables:
 - Hospital
 - Especialidad
 - total_pacientes
 - media_tiempo_dias
 - ano
 - semana
 - CODCNH

Observaciones generales sobre los datos

- En el fichero de lista de espera se mantiene el nombre de hospital de la descarga
- Pero se añade el código oficial, por el que se harán las uniones. El nombre oficial se tomaría del catálogo nacional
- Si se quisieran usar otras agrupaciones de zonas bastaría cambiar el fichero json

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 homogeneizado nombres de hospitales
- Se han asignado coordenadas al catálogo nacional de hospitales
- Se han asignado áreas de salud a los hospitales
- Se han añadido los códigos de hospitales a las listas de espera

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

- Se deben añadir indicadores a las áreas de salud para usar en los modelos

0.10 CODE TO DEPLOY (PILOT)

A continuación se incluirá el código que deba ser llevado a despliegue para producción, dado que se entienden 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

```
[50]: # incluir código
```