05. - Data Collection_CU_53_02_spi_v_01

June 13, 2023

#

CU53_impacto de las políticas de inversión en sanidad, infraestructuras y promoción turística en el SPI

Citizenlab Data Science Methodology > II - Data Processing Domain *** > # 05.- Data Collection

Data Collection is the process to obtain and generate (if required) necessary data to model the problem.

0.0.1 02. Transformar datos del SPI

- Convertir a csv los datos del SPI, que vienen en libro Excel con indicadores en columnas

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(readxl)
library(readr)
library(dplyr)
library(stringr)
library(tidyr)

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

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 <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[5]:
```
iFile <- "2022_social_progress_index_dataset-1663250262.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/2022_social_progress_index_dataset-1663250262.xlsx
```

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

- Se leen primero los metadatos de los índices

[6]:
```
metadata_01 <- read_excel(file_data,
                          sheet = "2011-2022 SPI data", col_names = FALSE,
                          skip = 1, n_max = 2) |>
  t() |> as.data.frame(row.names = FALSE, col.names = c("id_var", "name_var"))␣
↪|>
  rename(id_var = V1, name_var = V2)
```

```
New names:
• `` -> `...1`
• `` -> `...2`
• `` -> `...3`
• `` -> `...4`
• `` -> `...5`
• `` -> `...6`
• `` -> `...7`
• `` -> `...8`
• `` -> `...9`
• `` -> `...10`
• `` -> `...11`
• `` -> `...12`
• `` -> `...13`
• `` -> `...14`
• `` -> `...15`
• `` -> `...16`
```

- `` -> `…17`
- `` -> `…18`
- `` -> `…19`
- `` -> `…20`
- `` -> `…21`
- `` -> `…22`
- `` -> `…23`
- `` -> `…24`
- `` -> `…25`
- `` -> `…26`
- `` -> `…27`
- `` -> `…28`
- `` -> `…29`
- `` -> `…30`
- `` -> `…31`
- `` -> `…32`
- `` -> `…33`
- `` -> `…34`
- `` -> `…35`
- `` -> `…36`
- `` -> `…37`
- `` -> `…38`
- `` -> `…39`
- `` -> `…40`
- `` -> `…41`
- `` -> `…42`
- `` -> `…43`
- `` -> `…44`
- `` -> `…45`
- `` -> `…46`
- `` -> `…47`
- `` -> `…48`
- `` -> `…49`
- `` -> `…50`
- `` -> `…51`
- `` -> `…52`
- `` -> `…53`
- `` -> `…54`
- `` -> `…55`
- `` -> `…56`
- `` -> `…57`
- `` -> `…58`
- `` -> `…59`
- `` -> `…60`
- `` -> `…61`
- `` -> `…62`
- `` -> `…63`
- `` -> `…64`

- `` -> `…65`
- `` -> `…66`
- `` -> `…67`
- `` -> `…68`
- `` -> `…69`
- `` -> `…70`
- `` -> `…71`
- `` -> `…72`
- `` -> `…73`
- `` -> `…74`
- `` -> `…75`
- `` -> `…76`
- `` -> `…77`
- `` -> `…78`
- `` -> `…79`
- `` -> `…80`
- `` -> `…81`

[7]:
```
metadata_01 |> glimpse()
```

```
Rows: 81
Columns: 2
$ id_var   <chr> "rank_score_spi", "country",
"spicountrycode", "spiyear", "st…
$ name_var <chr> "SPI \r\nRank", "Country", "SPI country
code", "SPI \r\nyear"…
```

[8]:
```
metadata_01 |> slice_head(n = 10)
```

A data.frame: 10 x 2

id_var	name_var
<chr>	<chr>
rank_score_spi	SPI Rank
country	Country
spicountrycode	SPI country code
spiyear	SPI year
status	Status
score_spi	Social Progress Index
score_bhn	Basic Human Needs
score_fow	Foundations of Wellbeing
score_opp	Opportunity
score_nbmc	Nutrition & Basic Medical Care

- Ahora se leen los índices, con los nombres de columnas de los metadatos

[9]:
```
data <- read_excel(file_data,
                   sheet = "2011-2022 SPI data",
                   col_names = metadata_01$id_var,
                   skip = 3)
```

Estructura de los datos:

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

```
Rows: 2,364
Columns: 81
$ rank_score_spi       <dbl> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, 1…
$ country              <chr> "World", "World", "World",
"World", "World", "Wor…
$ spicountrycode       <chr> "WWW", "WWW", "WWW", "WWW",
"WWW", "WWW", "WWW", …
$ spiyear              <dbl> 2022, 2021, 2020, 2019, 2018,
2017, 2016, 2015, 2…
$ status               <chr> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, "…
$ score_spi            <dbl> 65.24, 64.87, 64.55, 64.31,
63.62, 63.29, 62.65, …
$ score_bhn            <dbl> 75.80, 75.35, 74.91, 74.48,
73.98, 73.43, 72.88, …
$ score_fow            <dbl> 63.62, 63.18, 62.91, 62.46,
61.03, 60.52, 59.26, …
$ score_opp            <dbl> 56.28, 56.07, 55.83, 55.98,
55.84, 55.92, 55.80, …
$ score_nbmc           <dbl> 82.09, 82.01, 81.79, 81.53,
81.23, 80.90, 80.53, …
$ score_ws             <dbl> 80.74, 79.84, 78.69, 77.99,
77.20, 76.59, 76.17, …
$ score_sh             <dbl> 79.36, 78.86, 78.50, 77.63,
76.92, 75.92, 74.97, …
$ score_ps             <dbl> 61.02, 60.68, 60.66, 60.78,
60.56, 60.31, 59.86, …
$ score_abk            <dbl> 72.42, 72.50, 72.36, 72.27,
71.90, 71.67, 71.16, …
$ score_aic            <dbl> 75.96, 74.44, 73.71, 72.87,
68.90, 67.40, 63.91, …
$ score_hw             <dbl> 58.21, 57.91, 57.84, 57.25,
56.72, 56.19, 55.78, …
$ score_eq             <dbl> 47.90, 47.87, 47.73, 47.45,
46.58, 46.82, 46.19, …
$ score_pr             <dbl> 60.40, 60.38, 61.01, 61.49,
61.97, 62.86, 63.30, …
$ score_pfc            <dbl> 62.27, 62.36, 62.39, 62.58,
62.46, 62.39, 62.05, …
$ score_incl           <dbl> 42.97, 42.14, 41.39, 41.86,
41.48, 41.62, 41.70, …
$ score_aae            <dbl> 59.50, 59.39, 58.52, 58.00,
57.46, 56.81, 56.17, …
$ nbmc_stunting        <dbl> 13.8027, 14.0202, 14.2332,
14.4525, 14.6888, 14.9…
```

```
$ nbmc_infectiousdaly  <dbl> 5412.513, 5585.193, 5808.714,
6038.574, 6314.341,…
$ nbmc_matmort         <dbl> 97.0507, 97.9615, 99.0098,
100.1971, 101.5897, 10…
$ nbmc_childmort       <dbl> 26.2493, 27.0843, 27.9841,
28.9533, 30.0464, 31.1…
$ nbmc_undernourish    <dbl> 9.5083, 8.9530, 8.8949, 8.9118,
8.9593, 9.0725, 9…
$ nbmc_dietlowfruitveg <dbl> 52.4891, 52.5851, 52.7105,
52.8507, 53.0054, 53.1…
$ ws_washmortdalys     <dbl> 1110.5481, 1163.1967, 1226.1396,
1257.8374, 1313.…
$ ws_sanitation        <dbl> 0.7853, 0.7724, 0.7550, 0.7559,
0.7462, 0.7339, 0…
$ ws_water             <dbl> 0.9112, 0.9078, 0.9033, 0.9041,
0.9019, 0.8990, 0…
$ ws_watersat          <dbl> 0.7424, 0.7315, 0.7182, 0.6976,
0.6868, 0.6847, 0…
$ sh_hhairpolldalys    <dbl> 1325.6415, 1400.8573, 1465.7838,
1528.8021, 1596.…
$ sh_affhousingdissat  <dbl> 0.3704, 0.3676, 0.3593, 0.3699,
0.3737, 0.3876, 0…
$ sh_electricity       <dbl> 90.0831, 89.5766, 89.1320,
88.1540, 87.2604, 86.1…
$ sh_cleanfuels        <dbl> 65.6136, 64.5201, 63.2636,
62.1685, 61.0557, 59.9…
$ ps_politicalkillings <dbl> 0.5855, 0.5653, 0.5829, 0.5956,
0.6022, 0.6017, 0…
$ ps_intpersvioldaly   <dbl> 341.1930, 348.6951, 350.5466,
349.5562, 355.8503,…
$ ps_transportdaly     <dbl> 995.2075, 1005.0394, 1008.2297,
1023.3297, 1043.7…
$ ps_intimpartnviol    <dbl> 13.2783, 13.3475, 13.4232,
13.4976, 13.5729, 13.6…
$ ps_moneystolen       <dbl> 0.1319, 0.1308, 0.1345, 0.1319,
0.1340, 0.1355, 0…
$ abk_qualeduc         <dbl> 1.5663, 1.6016, 1.6172, 1.6413,
1.6513, 1.6568, 1…
$ abk_propnoeduc       <dbl> 0.1677, 0.1714, 0.1753, 0.1792,
0.1833, 0.1874, 0…
$ abk_popsomesec       <dbl> 59.3857, 59.3257, 58.9104,
58.7618, 57.9001, 57.5…
$ abk_totprimenrol     <dbl> 92.5845, 92.6930, 92.7453,
92.8029, 92.7666, 92.7…
$ abk_educpar          <dbl> 0.1895, 0.1892, 0.1913, 0.1944,
0.2000, 0.2031, 0…
$ aic_altinfo          <dbl> 0.5581, 0.5581, 0.5619, 0.5620,
0.5664, 0.5837, 0…
```

```
$ aic_mobiles          <dbl> 105.7527, 106.1647, 103.5512,
102.5812, 100.4686,…
$ aic_internet         <dbl> 58.4423, 52.9280, 48.5497,
45.4282, 42.9375, 39.9…
$ aic_eparticip        <dbl> 0.7534, 0.7550, 0.7725, 0.7738,
0.6500, 0.6513, 0…
$ hw_qualityhealth     <dbl> 1.5816, 1.5880, 1.6693, 1.6806,
1.6953, 1.7167, 1…
$ hw_lifex60           <dbl> 20.3304, 20.2712, 20.2290,
20.1313, 20.0435, 19.9…
$ hw_ncdmort           <dbl> 374.7873, 380.2000, 386.0096,
392.0828, 398.3793,…
$ hw_univhealthcov     <dbl> 66.4853, 66.0378, 65.5144,
64.9584, 64.5299, 64.2…
$ hw_qualhealthsat     <dbl> 0.6394, 0.6381, 0.6359, 0.6232,
0.6110, 0.5961, 0…
$ eq_airpolldalys      <dbl> 1564.3610, 1551.2358, 1555.8151,
1599.3593, 1647.…
$ eq_leadexpdalys      <dbl> 326.7876, 333.9709, 341.1407,
349.1149, 356.5378,…
$ eq_pm25              <dbl> 45.7065, 45.6731, 45.6643,
45.3535, 47.3430, 45.9…
$ eq_spindex           <dbl> 61.1883, 61.1169, 60.9789,
60.8695, 60.7703, 60.7…
$ pr_freerelig         <dbl> 2.4025, 2.4020, 2.4122, 2.4692,
2.4784, 2.5806, 2…
$ pr_proprightswomen   <dbl> 3.9055, 3.8970, 3.9027, 3.8866,
3.8830, 3.8944, 3…
$ pr_peaceassemb       <dbl> 1.9866, 2.0530, 2.1117, 2.1161,
2.1515, 2.1745, 2…
$ pr_accessjustice     <dbl> 0.5941, 0.5831, 0.5916, 0.5914,
0.5999, 0.5998, 0…
$ pr_freediscuss       <dbl> 0.5978, 0.5849, 0.5917, 0.6027,
0.6100, 0.6179, 0…
$ pr_polrights         <dbl> 19.2707, 19.6530, 19.9072,
20.2304, 20.4150, 20.9…
$ pfc_freedomestmov    <dbl> 0.6054, 0.5953, 0.6169, 0.6535,
0.6467, 0.6577, 0…
$ pfc_earlymarriage    <dbl> 10.3461, 10.5139, 10.6854,
10.8554, 11.0282, 11.2…
$ pfc_contracept       <dbl> 74.4316, 74.1702, 73.9048,
73.6566, 73.3902, 73.2…
$ pfc_neet             <dbl> 22.2264, 20.7324, 20.9768,
20.9641, 20.9421, 20.9…
$ pfc_vulnemploy       <dbl> 46.2278, 46.5302, 46.9376,
47.2727, 47.5735, 47.9…
$ pfc_corruption       <dbl> 41.1208, 40.7652, 40.8253,
40.4453, 40.9192, 40.6…
```

```
$ incl_equalprotect    <dbl> 0.5141, 0.5096, 0.5183, 0.5462,
0.5550, 0.5578, 0…
$ incl_equalaccess     <dbl> 0.5446, 0.5460, 0.5225, 0.5603,
0.5557, 0.5666, 0…
$ incl_sexualorient    <dbl> 0.8111, 0.7890, 0.7612, 0.8102,
0.7837, 0.8245, 0…
$ incl_accpubsersocgr  <dbl> 1.9767, 1.9979, 2.0361, 2.0183,
2.0772, 2.0809, 2…
$ incl_gayslesb        <dbl> 0.4127, 0.3851, 0.3506, 0.3064,
0.2839, 0.2645, 0…
$ incl_discrimin       <dbl> 7.1472, 7.3180, 7.2646, 7.4035,
7.5010, 7.5329, 7…
$ aae_acadfreed        <dbl> 0.4374, 0.4784, 0.4647, 0.4733,
0.4836, 0.4861, 0…
$ aae_femterteduc      <dbl> 0.3450, 0.3342, 0.3235, 0.3132,
0.3032, 0.2936, 0…
$ aae_tertschlife      <dbl> 2.1666, 2.1218, 2.0671, 2.0268,
2.0050, 1.9681, 1…
$ aae_citabledocs      <dbl> 0.5862, 0.5438, 0.5146, 0.4859,
0.4609, 0.4489, 0…
$ aae_qualuniversities <dbl> 217.6475, 225.5760, 211.4173,
203.6117, 185.5802,…
```

Muestra de datos:

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

A tibble: 5 x 81

rank_score_spi <dbl>	country <chr>	spicountrycode <chr>	spiyear <dbl>	status <chr>	score_spi <dbl>	score_bhn <dbl>	score_ <dbl>
NA	World	WWW	2022	NA	65.24	75.80	63.62
NA	World	WWW	2021	NA	64.87	75.35	63.18
NA	World	WWW	2020	NA	64.55	74.91	62.91
NA	World	WWW	2019	NA	64.31	74.48	62.46
NA	World	WWW	2018	NA	63.62	73.98	61.03

- Obtenemos una segunda parte de los metadatos

[12]:
```
metadata_02 <- read_excel(file_data,
                          sheet = "DEFINITIONS") |>
  drop_na(`Indicator name`)
```

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:

- Datos de índices
- Metadatos de índices (diccionario de datos)

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

- Transformar metadatos:
  - Eliminar saltos de línea
  - Separar unidades
  - Sustituir & por and
  - Unir dimensiones
  - Añadir indicador del rol del valor

[13]:
```
tmetadata <- metadata_01 |>
  mutate(name_var = str_remove_all(name_var, "\\r\\n")) |>
  separate(name_var, into = c("name_var", "tmp_units"), sep = " \\(") |>
  mutate(name_var = str_trim(name_var)) |>
  mutate(name_var = str_replace_all(name_var, "\\&", "and")) |>
  select(-tmp_units) |>
  mutate(name_var = str_to_sentence(name_var)) |>
  left_join(metadata_02 |> select(`Indicator name`, Dimension,
                                  Component, `Unit of measurement`,
                                  Definition, Source),
            by = c("name_var" = "Indicator name")) |>
  mutate(role = c(rep("aux", 5), "SPI",
                  rep("Dimension", 3),
                  rep("Component", 3*4),
                  rep("Indicator", 60)),
         .after = "name_var")
```

```
Warning message:
"Expected 2 pieces. Missing pieces filled with `NA` in 21 rows [1, 2,
3, 4, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, …]."
```

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

```
Rows: 81
Columns: 8
$ id_var                <chr> "rank_score_spi", "country",
"spicountrycode", "…
$ name_var              <chr> "Spi rank", "Country", "Spi
country code", "Spi …
$ role                  <chr> "aux", "aux", "aux", "aux",
"aux", "SPI", "Dimen…
$ Dimension             <chr> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, …
$ Component             <chr> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, …
```

```
$ `Unit of measurement` <chr> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, …
$ Definition            <chr> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, …
$ Source                <chr> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, …
```

[15]:
```
tmetadata |> slice(c(1:3, 22:26))
```

A data.frame: 8 x 8

id_var <chr>	name_var <chr>	role <chr>	Dimension <chr>	Compon <chr>
rank_score_spi	Spi rank	aux	NA	NA
country	Country	aux	NA	NA
spicountrycode	Spi country code	aux	NA	NA
nbmc_stunting	Child stunting	Indicator	Basic Human Needs	Nutritio
nbmc_infectiousdaly	Infectious diseases	Indicator	Basic Human Needs	Nutritio
nbmc_matmort	Maternal mortality rate	Indicator	Basic Human Needs	Nutritio
nbmc_childmort	Child mortality rate	Indicator	Basic Human Needs	Nutritio
nbmc_undernourish	Undernourishment	Indicator	Basic Human Needs	Nutritio

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 del SPI

Identificamos los datos a guardar

[16]:
```
data_to_save_01 <- data
```

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

[17]:
```
caso <- "CU_53"
proceso <- '_05'
tarea <- "_02"
archivo <- "_01"
proper <- "_spi"
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

[18]:
```
# 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_01
```

OPCION B: Especificar el nombre de archivo

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

[19]:
```
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_53_05_02_01_spi.csv’

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

[20]:
```
path_in_01 <- paste0(iPath, file_save_01)
file.copy(path_out_01, path_in_01, overwrite = TRUE)
```

TRUE

2. Metadatos del SPI

Identificamos los datos a guardar

[22]:
```
data_to_save_02 <- tmetadata
```

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

[23]:
```
archivo <- "_02"
proper <- "_spi_metadata"
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

[24]:
```
# 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/.

[25]:
```
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_53_05_02_02_spi_metadata.csv’

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

[26]:
```
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 readxl, readr, dplyr, stringr, tidyr deben estar instalados.

0.8.2 Configuration Management

This notebook has been tested with the following versions of R and packages. It cannot be assured that later versions work in the same way: * R 4.2.2 * tcltk 4.2.2 * readxl 1.4.1 * readr 2.1.3 * dplyr 1.0.10 * stringr 1.5.0 * tidyr 1.3.0

0.8.3 Data structures

Objeto `data`

- Hay 2364 filas con información de las siguientes variables:
  - rank_score_spi
  - country
  - spicountrycode
  - spiyear
  - status
  - score_spi
  - score_bhn
  - score_fow
  - score_opp
  - score_nbmc
  - score_ws
  - score_sh
  - score_ps
  - score_abk
  - score_aic
  - score_hw
  - score_eq
  - score_pr

- score_pfc
- score_incl
- score_aae
- nbmc_stunting
- nbmc_infectiousdaly
- nbmc_matmort
- nbmc_childmort
- nbmc_undernourish
- nbmc_dietlowfruitveg
- ws_washmortdalys
- ws_sanitation
- ws_water
- ws_watersat
- sh_hhairpolldalys
- sh_affhousingdissat
- sh_electricity
- sh_cleanfuels
- ps_politicalkillings
- ps_intpersvioldaly
- ps_transportdaly
- ps_intimpartnviol
- ps_moneystolen
- abk_qualeduc
- abk_propnoeduc
- abk_popsomesec
- abk_totprimenrol
- abk_educpar
- aic_altinfo
- aic_mobiles
- aic_internet
- aic_eparticip
- hw_qualityhealth
- hw_lifex60
- hw_ncdmort
- hw_univhealthcov
- hw_qualhealthsat
- eq_airpolldalys
- eq_leadexpdalys
- eq_pm25
- eq_spindex
- pr_freerelig
- pr_proprightswomen
- pr_peaceassemb
- pr_accessjustice
- pr_freediscuss
- pr_polrights
- pfc_freedomestmov
- pfc_earlymarriage

  - pfc_contracept
  - pfc_neet
  - pfc_vulnemploy
  - pfc_corruption
  - incl_equalprotect
  - incl_equalaccess
  - incl_sexualorient
  - incl_accpubsersocgr
  - incl_gayslesb
  - incl_discrimin
  - aae_acadfreed
  - aae_femterteduc
  - aae_tertschlife
  - aae_citabledocs
  - aae_qualuniversities

Objeto `tmetadata`

- Hay 81 filas con información de las siguientes variables:
  - id_var
  - name_var
  - role
  - Dimension
  - Component
  - Unit of measurement
  - Definition
  - Source

Observaciones generales sobre los datos

- Los datos son jerárquicos: los indicadores forman componentes, los componentes dimensiones y las dimensiones el índice.

0.8.4 Consideraciones para despliegue en piloto

- Los metadatos son útiles para la representación en tablas y gráficos con etiqueteas amigables

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
- Los datos de entrada se deben adquirir para usar en producción

0.9 Main Actions

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

- Se han dispuesto los índices en formato rectangular con nombres de columna adecuados

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

- Se deben relacionar estos datos con los de inversiones

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
```