

# CU55\_MODEL\_DEVELOPMENT\_02\_ESCENARIOS

June 13, 2023

#

CU55\_Modelo agregado de estimación del gasto medio por turista

## 1 IV. Model development

En este anexo se incluye el código utilizado durante el desarrollo de los modelos incluidos en el caso de uso.

### 1.1 Generación de escenarios

```
[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'
```

```
[2]: ## Generador de escenarios
```

```
library(readr)
library(dplyr)
library(stringr)

gasto_municipio <- read_csv("CU_55_05_02_gasto_municipio.csv")

## Escenario origen

gasto_municipio |>
  filter(mes %in% c("2021-08", "2021-09", "2021-10"),
         pais_orig == "Francia",
         str_starts(mun_dest, "M")) |>
  select(-c(pais_orig_cod, mun_dest_cod, CMUN, gasto)) |>
  write_csv("ESCENARIO_ORIGEN.csv")

## Escenario destino

gasto_municipio |>
  filter(mes %in% c("2021-08"),
         str_starts(pais_orig, "Total", negate = TRUE),
         mun_dest == "Alcalá de Henares") |>
```

```
select(-c(pais_orig_cod, mun_dest_cod, CMUN, gasto)) |>
write_csv("ESCENARIO_DESTINO.csv")
```

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

Rows: 50294 Columns: 8  
Column specification

Delimiter: ","

chr (5): mes, pais\_orig\_cod, pais\_orig, mun\_dest, CMUN

dbl (3): mun\_dest\_cod, turistas, gasto

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.

Error in `filter()`:

In argument: `str\_starts(mun\_dest, "M")`.

Caused by error in `str\_starts()`:

! no se pudo encontrar la función "str\_starts"

Traceback:

```
1. write_csv(select(filter(gasto_municipio, mes %in% c("2021-08",
.      "2021-09", "2021-10"), pais_orig == "Francia", str_starts(mun_dest,
.      "M")), -c(pais_orig_cod, mun_dest_cod, CMUN, gasto)),
↳ "ESCENARIO_ORIGEN.csv")

2. write_delim(x, file, delim = ",", na = na, append = append, col_names
↳ = col_names,
.      quote = quote, escape = escape, eol = eol, num_threads = num_threads,
.      progress = progress)

3. stopifnot(is.data.frame(x))
```

```

4. is.data.frame(x)

5. select(filter(gasto_municipio, mes %in% c("2021-08", "2021-09",
.   "2021-10"), pais_orig == "Francia", str_starts(mun_dest,
.   "M")), -c(pais_orig_cod, mun_dest_cod, CMUN, gasto))

6. filter(gasto_municipio, mes %in% c("2021-08", "2021-09", "2021-10"),
.   pais_orig == "Francia", str_starts(mun_dest, "M"))

7. filter.data.frame(gasto_municipio, mes %in% c("2021-08", "2021-09",
.   "2021-10"), pais_orig == "Francia", str_starts(mun_dest,
.   "M"))

8. filter_rows(.data, dots, by)

9. filter_eval(dots, mask = mask, error_call = error_call)

10. withCallingHandlers(mask$eval_all_filter(dots, env_filter), error = ␣
↪dplyr_error_handler(dots = dots,
.   mask = mask, bullets = filter_bullets, error_call = error_call),
.   warning = function(cnd) {
.     local_error_context(dots, i, mask)
.     warning_handler(cnd)
.   }, `dplyr:::signal_filter_one_column_matrix` = function(e) {
.     warn_filter_one_column_matrix(call = error_call)
.   }, `dplyr:::signal_filter_across` = function(e) {
.     warn_filter_across(call = error_call)
.   }, `dplyr:::signal_filter_data_frame` = function(e) {
.     warn_filter_data_frame(call = error_call)
.   })

11. mask$eval_all_filter(dots, env_filter)

12. eval()

13. .handleSimpleError(function (cnd)
. {
.   local_error_context(dots, i = frame[[i_sym]], mask = mask)
.   if (inherits(cnd, "dplyr:::internal_error")) {
.     parent <- error_cnd(message = bullets(cnd))
.   }
.   else {
.     parent <- cnd
.   }
.   message <- c(cnd_bullet_header(action), i = if ␣
↪(has_active_group_context(mask)) cnd_bullet_cur_group_label())

```

```

        .      abort(message, class = error_class, parent = parent, call =
↪error_call)
        . }, "no se pudo encontrar la función \"str_starts\"", base::
↪quote(str_starts(mun_dest,
        .      "M"))))

14. h(simpleError(msg, call))

15. abort(message, class = error_class, parent = parent, call =
↪error_call)

16. signal_abort(cnd, .file)

```