

CU04_MODEL_DEVELOPMENT_03_pasos

June 9, 2023

#

CU04_Optimización de vacunas

1 IV. Model development

Se incluye aquí otro código, sin ejecutar, que se ha utilizado durante el desarrollo de los modelos para su visualización en la implementación.

1.1 Paso 2

```
[ ]: ## ZONAS SANITARIAS E HISTÓRICO DE VACUNACIÓN

source("LIBRERIAS.R")

source("VARIABLES.R")

file_name <- "CU_04_05_01_zonasgeo.json"
ZONAS <- st_read(file_name)

file_name <- "CU_04_05_16_vacunacion_gripe.csv"
HISTORICO <- read_csv(file_name)

## MAPA

ldata <- ZONAS |>
  left_join(HISTORICO, by = c("GEOCODIGO", "DESBDT"),
            multiple = "all") |>
  filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 & semana <=
  ↪ SEMANA_FIN)) |>
  group_by(GEOCODIGO, DESBDT) |>
  summarise(n_vacunas = sum(n_vacunas), .groups = "drop")

# COL1 <- rgb(33/255, 150/255, 243/255)
pal <- colorNumeric(palette = "Blues",
                   domain = ldata$n_vacunas)
```

```

ldata |>
  leaflet() |>
  addTiles() |>
  addPolygons(color = "#444444",
              weight = 1,
              smoothFactor = 0.5,
              fillOpacity = 1,
              fillColor = ~pal(n_vacunas),
              highlightOptions = highlightOptions(color = "white", weight = 2,
                                                    bringToFront = TRUE),
              popup = ~paste0(DESBDT, " (", GEOCODIGO, ")"),
              label = ~paste0(n_vacunas, " vacunas")) |>
  addLegend("bottomright",
            pal = pal,
            values = ~n_vacunas,
            title = "Número de vacunas",
            labFormat = labelFormat(big.mark = " "),
            opacity = 1
  )

## SERIE

if(is.na(ZONA)){
  sdata <- HISTORICO |>
    filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 & semana <=
↪SEMANA_FIN)) |>
    group_by(ano, semana) |>
    summarise(n_vacunas = sum(n_vacunas, na.rm = TRUE), .groups = "drop")
  NZONA <- NA
} else{
  sdata <- HISTORICO |>
    filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 & semana <=
↪SEMANA_FIN),
           GEOCODIGO == ZONA)
  NZONA <- ZONAS |>
    filter(GEOCODIGO == ZONA) |>
    pull(DESBDT)
}
sdata <- sdata |>
  mutate(ano_semana = paste0(ano, "-", semana),
         fecha = as.Date(parse_date_time(paste(ano, semana, 1, sep="/"), 'Y/W/
↪W'))))

sdata |>
  ggplot() +
  aes(x = fecha,

```

```

    y = n_vacunas) +
  geom_line(col = COL1) +
  labs(title = paste0("Histórico campaña ", ANO, "/", ANO + 1),
        subtitle = if_else(is.na(ZONA), "Total zonas",
                           paste0("Zona ", ZONA,
                                   " (", NZONA, ")")),

        x = "Semana",
        y = "Total vacunas") +
  scale_x_date(date_breaks = "1 month",
               date_minor_breaks = "1 week",
               labels = function(x) month(x, label = TRUE)) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5))

## TABLA

HISTORICO |>
  filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 & semana <=
→SEMANA_FIN)) |>
  group_by(GEOCODIGO, DESBDT) |>
  summarise(n_vacunas = sum(n_vacunas, na.rm = TRUE), .groups = "drop") |>
  datatable(rownames = FALSE, colnames = c("Código zona", "Nombre zona", "Total
→vacunas campaña")) |>
  formatRound(3, dec.mark = ",", mark = ".", digits = 0)

```

1.1.1 Paso 3

```

[ ]: ## PROYECCIÓN CAMPAÑA

source("LIBRERIAS.R")

source("VARIABLES.R")

## ZONAS
file_name <- "ZONAS.json"
ZONAS <- st_read(file_name)

## ESCENARIO
file_name <- "NEWDATA.csv"
ESCENARIO <- read_csv(file_name)

## MODELO
modelo <- read_rds("mod_04.rds")

```

```

## PREDICCIÓN
file_name <- "PREDICTION.csv"
prediction <- predict.gam(modelo, ESCENARIO, se.fit = TRUE, type = "response")
ESCENARIO.pred <- ESCENARIO |> bind_cols(data.frame(prediction) )
ESCENARIO.pred |> write_csv(file_name)

## VISUALIZACIÓN

## Se muestra solo la variable respuesta

## MAPA

ldata <- ZONAS |>
  left_join(ESCENARIO.pred, by = c("GEOCODIGO"),
            multiple = "all") |>
  # filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 & semana <=
  ↪ SEMANA_FIN)) |>
  group_by(GEOCODIGO, DESBDT) |>
  summarise(n_vacunas = sum(fit, na.rm = TRUE), .groups = "drop")

# COL1 <- rgb(33/255, 150/255, 243/255)
pal <- colorNumeric(palette = "Blues",
                   domain = ldata$n_vacunas)

ldata |>
  leaflet() |>
  addTiles() |>
  addPolygons(color = "#444444",
              weight = 1,
              smoothFactor = 0.5,
              fillOpacity = 1,
              fillColor = ~pal(n_vacunas),
              highlightOptions = highlightOptions(color = "white", weight = 2,
                                                    bringToFront = TRUE),
              popup = ~paste0(DESBDT, " (", GEOCODIGO, ")"),
              label = ~paste0(round(n_vacunas), " vacunas")) |>
  addLegend("bottomright",
            pal = pal,
            values = ~n_vacunas,
            title = "Número de vacunas<br/>Predicción escenario",
            labFormat = labelFormat(big.mark = " "),
            opacity = 1
  )

## SERIE

```

```

if(is.na(ZONA)){
  sdata <- ESCENARIO.pred |>
    group_by(scampana) |>
    summarise(fit = sum(fit, na.rm = TRUE), .groups = "drop")
  NZONA <- NA
} else{
  sdata <- ESCENARIO.pred |>
    filter(GEOCODIGO == ZONA)
  NZONA <- ZONAS |>
    filter(GEOCODIGO == ZONA) |>
    pull(DESBDT)
}

sdata |>
  ggplot() +
  aes(x = scampana,
      y = fit) +
  geom_line(col = COL1) +
  labs(title = paste0("Predicción campaña ", ANO, "/", ANO + 1),
       subtitle = if_else(is.na(ZONA), "Total zonas",
                          paste0("Zona ", ZONA,
                                " (", NZONA, ")")),
       x = "Semana",
       y = "Total vacunas") +
  theme_bw()

## TABLA

ESCENARIO.pred |> right_join(tibble(ZONAS) |> select(2:3), by = "GEOCODIGO") |>
  group_by(GEOCODIGO, DESBDT) |>
  summarise(fit = sum(fit, na.rm = TRUE), .groups = "drop") |>
  datatable(rownames = FALSE, colnames = c("Código zona", "", "Total predicción",
→vacunas escenario")) |>
  formatRound(3, dec.mark = ",", mark = ".", digits = 0)

#
# ## MAPA
# ldata <- ZONAS |>
#   left_join(ESCENARIO.pred, by = c("GEOCODIGO"),
#             multiple = "all") |>
#   # filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 &
→semana <= SEMANA_FIN)) |>

```

```

#   group_by(GEOCODIGO) />
#   summarise(across(fit:se.fit, ~sum(.x, na.rm = TRUE)),
#             across(tmed:so2, ~mean(.x, na.rm = TRUE)),
#             .groups = "drop")
#   pal <- colorNumeric(palette = "Blues",
#                       domain = ldata /> pull(fit))
#   ldata />
#   leaflet() />
#   addTiles() />
#   addPolygons(color = "#444444",
#               weight = 1,
#               smoothFactor = 0.5,
#               fillOpacity = 1,
#               fillColor = ~pal(fit),
#               highlightOptions = highlightOptions(color = "white", weight =
→2,
#
#               bringToFront = TRUE),
#               popup = ~paste0(GEOCODIGO),
#               label = ~paste0(fit, " vacunas")) />
#   addLegend("bottomright",
#             pal = pal,
#             values = ~fit,
#             title = paste0("Predicción vacunas: "),
#             labFormat = labelFormat(big.mark = " "),
#             opacity = 1
#   )
#
#   ## GRÁFICO
#   if(is.na(ZONA)){
#     sdata <- HISTORICO />
#     filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 &
→semana <= SEMANA_FIN)) />
#     group_by(ano, semana) />
#     summarise("{PREDICTOR}" := mean(eval(parse(text = PREDICTOR))), na.rm =
→TRUE), .groups = "drop")
#     NZONA <- NA
#   } else{
#     sdata <- HISTORICO />
#     filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 &
→semana <= SEMANA_FIN),
#           GEOCODIGO == ZONA)
#     NZONA <- ZONAS />
#     filter(GEOCODIGO == ZONA) />
#     pull(DESBDT)
#   }
#   sdata <- sdata />
#   mutate(ano_semana = paste0(ano, "-", semana),

```

```

#           fecha = as.Date(parse_date_time(paste(ano, semana, 1, sep="/"), 'Y/
→W/w'))))
#
#   sdata |>
#     ggplot() +
#       aes(x = fecha,
#           y = eval(parse(text = PREDICTOR))) +
#       geom_line(col = COL1) +
#       labs(title = paste0("Histórico campaña ", ANO, "/", ANO + 1),
#            subtitle = if_else(is.na(ZONA), "Media zonas",
#                               paste0("Zona ", ZONA,
#                                       " (", NZONA, ")")),
#            x = "Semana",
#            y = paste("Media de ", PREDICTOR)) +
#       scale_x_date(date_breaks = "1 month",
#                    date_minor_breaks = "1 week",
#                    labels = function(x) month(x, label = TRUE)) +
#       theme_bw() +
#       theme(axis.text.x = element_text(angle = 45, vjust = 0.5))
#
#   ## TABLA
#   sdata |>
#     select(-ano_semana, -fecha) |>
#     datatable(rownames = FALSE,
#               colnames = c("Año", "Semana",
#                           PREDICTOR)) |>
#     formatRound(3, dec.mark = ",", mark = ".", digits = 2)
#
#
#
#
# } else if (PREDICTOR %in% colnames(ESCUCHA)){
#
#   ## MAPA
#   ## NO HAY MAPA PUESTO QUE LOS DATOS DE ESCUCHA NO ESTÁN GEOLOCALIZADOS
#
#   ## GRÁFICO
#   sdata <- ESCUCHA |>
#     filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 & semana_
→<= SEMANA_FIN)) |>
#     mutate(ano_semana = paste0(ano, "-", semana),
#            fecha = as.Date(parse_date_time(paste(ano, semana, 1, sep="/"), 'Y/
→W/w'))))
#
#   NZONA <- NA
#

```

```

# sdata />
# ggplot() +
# aes(x = fecha,
#     y = eval(parse(text = PREDICTOR))) +
# geom_line(col = COL1) +
# labs(title = paste0("Histórico campaña ", ANO, "/", ANO + 1),
#     subtitle = "Total zonas",
#     x = "Semana",
#     y = paste("Total de ", PREDICTOR)) +
# scale_x_date(date_breaks = "1 month",
#     date_minor_breaks = "1 week",
#     labels = function(x) month(x, label = TRUE)) +
# theme_bw() +
# theme(axis.text.x = element_text(angle = 45, vjust = 0.5))
#
# ## TABLA
# sdata />
# select(-ano_semana, -fecha) />
# datatable(rownames = FALSE,
#     colnames = c("Año", "Semana", "Tuits gripe", "Interés gripe"))
→ />
# formatRound(4, dec.mark = ",", mark = ".", digits = 0)
#
#
#
# } else if(PREDICTOR %in% colnames(INDICADORES)){
#
# ## MAPA
# ldata <- ZONAS />
# left_join(INDICADORES, by = c("GEOCODIGO", "DESBDT"),
#     multiple = "all")
# pal <- colorNumeric(palette = "Blues",
#     domain = ldata /> pull(PREDICTOR))
# ldata />
# leaflet() />
# addTiles() />
# addPolygons(color = "#444444",
#     weight = 1,
#     smoothFactor = 0.5,
#     fillOpacity = 1,
#     fillColor = ~pal(eval(parse(text = PREDICTOR))),
#     highlightOptions = highlightOptions(color = "white", weight =
→ 2,
#     bringToFront = TRUE),
#     popup = ~paste0(DESBDT, " (", GEOCODIGO, ")"),
#     label = ~paste0(round(eval(parse(text = PREDICTOR)), 2)*100))
→ />

```



```

#   addLegend("bottomright",
#             pal = pal,
#             values = ~eval(parse(text = PREDICTOR)),
#             title = paste0("Predictor: ", PREDICTOR),
#             labFormat = labelFormat(big.mark = " "),
#             opacity = 1
#   )
#
#   ## SERIE
#   ## No hay serie puesto que los indicadores son estáticos
#   ## Se muestra gráfico de barras con el top 10
#   gdata <- ZONAS |> tibble() |>
#     left_join(INDICADORES, by = c("GEOCODIGO", "DESBDT")) |>
#     slice_max(desc(eval(parse(text = PREDICTOR))), n = 10)
#   gdata |>
#     ggplot() +
#     aes(y = fct_reorder(DESBDT, eval(parse(text = PREDICTOR))),
#         x = eval(parse(text = PREDICTOR))) +
#     geom_col(fill = COL1) +
#     theme_bw() +
#     labs(title = paste0("Top 10 zonas por ", PREDICTOR),
#          # subtitle = "Total zonas",
#          x = PREDICTOR,
#          y = "")
#
#   ## TABLA
#
#   tdata <- ZONAS |> tibble() |>
#     left_join(INDICADORES, by = c("GEOCODIGO", "DESBDT"),
#               multiple = "all")
#
#   tdata |>
#     datatable(rownames = FALSE) |>
#     formatRound(5:ncol(tdata), dec.mark = ",", mark = ".", digits = 2)
# }
#
#
#
#
#
#
#
#

```

1.1.2 Paso 4

```
[ ]: ## VARIABLES PREDICTIVAS

source("LIBRERIAS.R")

source("VARIABLES.R")


## ZONAS
file_name <- "CU_04_05_01_zonasgeo.json"
ZONAS <- st_read(file_name)
st_set

## HISTORICO
file_name <- "HISTORICO.csv"
HISTORICO <- read_csv(file_name)

## ESCUCHA
file_name <- "CU_04_05_18_escucha_gripe.csv"
ESCUCHA <- read_csv(file_name)

## INDICADORES
file_name <- "CU_04_05_17_indicadores_vacunacion.csv"
INDICADORES <- read_csv(file_name)

if (PREDICTOR %in% colnames(HISTORICO)){

  ## MAPA
  ldata <- ZONAS |>
    left_join(HISTORICO, by = c("GEOCODIGO", "DESBDT"),
              multiple = "all") |>
    filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 & semana <=
    SEMANA_FIN)) |>
    group_by(GEOCODIGO, DESBDT) |>
    summarise(across(n_vacunas:n_citas, ~sum(.x, na.rm = TRUE)),
              across(tmed:so2, ~mean(.x, na.rm = TRUE)),
              .groups = "drop")
  pal <- colorNumeric(palette = "Blues",
                     domain = ldata |> pull(PREDICTOR))
  ldata |>
    leaflet() |>
    addTiles() |>
```

```

addPolygons(color = "#444444",
             weight = 1,
             smoothFactor = 0.5,
             fillOpacity = 1,
             fillColor = ~pal(eval(parse(text = PREDICTOR))),
             highlightOptions = highlightOptions(color = "white", weight = 2,
                                                  bringToFront = TRUE),
             popup = ~paste0(DESBDT, " (", GEOCODIGO, ")"),
             label = ~paste0(eval(parse(text = PREDICTOR)), " vacunas")) |>
addLegend("bottomright",
          pal = pal,
          values = ~eval(parse(text = PREDICTOR)),
          title = paste0("Predictor: ", PREDICTOR),
          labFormat = labelFormat(big.mark = " "),
          opacity = 1
)

## GRÁFICO
if(is.na(ZONA)){
  sdata <- HISTORICO |>
    filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 & semana_
↪<= SEMANA_FIN)) |>
    group_by(ano, semana) |>
    summarise("{PREDICTOR}" := mean(eval(parse(text = PREDICTOR))), na.rm =_
↪TRUE), .groups = "drop")
  NZONA <- NA
} else{
  sdata <- HISTORICO |>
    filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 & semana_
↪<= SEMANA_FIN),
           GEOCODIGO == ZONA)
  NZONA <- ZONAS |>
    filter(GEOCODIGO == ZONA) |>
    pull(DESBDT)
}
sdata <- sdata |>
  mutate(ano_semana = paste0(ano, "-", semana),
         fecha = as.Date(parse_date_time(paste(ano, semana, 1, sep="/"), 'Y/W/
↪W'))))

sdata |>
  ggplot() +
  aes(x = fecha,
      y = eval(parse(text = PREDICTOR))) +
  geom_line(col = COL1) +
  labs(title = paste0("Histórico campaña ", ANO, "/", ANO + 1),
       subtitle = if_else(is.na(ZONA), "Media zonas",

```

```

        paste0("Zona ", ZONA,
              " (", NZONA, ")"),
      x = "Semana",
      y = paste("Media de ", PREDICTOR)) +
    scale_x_date(date_breaks = "1 month",
                 date_minor_breaks = "1 week",
                 labels = function(x) month(x, label = TRUE)) +
    theme_bw() +
    theme(axis.text.x = element_text(angle = 45, vjust = 0.5))

## TABLA
sdata |>
  select(-ano_semana, -fecha) |>
  datatable(rownames = FALSE,
            colnames = c("Año", "Semana",
                        PREDICTOR)) |>
  formatRound(3, dec.mark = ",", mark = ".", digits = 2)

} else if (PREDICTOR %in% colnames(ESCUCHA)){

## MAPA
## NO HAY MAPA PUESTO QUE LOS DATOS DE ESCUCHA NO ESTÁN GEOLOCALIZADOS

## GRÁFICO
sdata <- ESCUCHA |>
  filter((ano == ANO & semana >= SEMANA_INICIO) | (ano == ANO + 1 & semana <=
→SEMANA_FIN)) |>
  mutate(ano_semana = paste0(ano, "-", semana),
        fecha = as.Date(parse_date_time(paste(ano, semana, 1, sep="/"), 'Y/W/
→W'))))

NZONA <- NA

sdata |>
  ggplot() +
  aes(x = fecha,
      y = eval(parse(text = PREDICTOR))) +
  geom_line(col = COL1) +
  labs(title = paste0("Histórico campaña ", ANO, "/", ANO + 1),
       subtitle = "Total zonas",
       x = "Semana",
       y = paste("Total de ", PREDICTOR)) +
  scale_x_date(date_breaks = "1 month",

```

```

        date_minor_breaks = "1 week",
        labels = function(x) month(x, label = TRUE)) +
theme_bw() +
theme(axis.text.x = element_text(angle = 45, vjust = 0.5))

## TABLA
sdata |>
  select(-ano_semana, -fecha) |>
  datatable(rownames = FALSE,
            colnames = c("Año", "Semana", "Tuits gripe", "Interés gripe")) |>
  formatRound(4, dec.mark = ",", mark = ".", digits = 0)

} else if(PREDICTOR %in% colnames(INDICADORES)){

## MAPA
ldata <- ZONAS |>
  left_join(INDICADORES, by = c("GEOCODIGO", "DESBDT"),
            multiple = "all")
pal <- colorNumeric(palette = "Blues",
                  domain = ldata |> pull(PREDICTOR))

ldata |>
  leaflet() |>
  addTiles() |>
  addPolygons(color = "#444444",
              weight = 1,
              smoothFactor = 0.5,
              fillOpacity = 1,
              fillColor = ~pal(eval(parse(text = PREDICTOR))),
              highlightOptions = highlightOptions(color = "white", weight = 2,
                                                    bringToFront = TRUE),
              popup = ~paste0(DESBDT, " (", GEOCODIGO, ")"),
              label = ~paste0(round(eval(parse(text = PREDICTOR)), 2)*100)) |>
  addLegend("bottomright",
            pal = pal,
            values = ~eval(parse(text = PREDICTOR)),
            title = paste0("Predictor: ", PREDICTOR),
            labFormat = labelFormat(big.mark = " "),
            opacity = 1
  )

## SERIE
## No hay serie puesto que los indicadores son estáticos
## Se muestra gráfico de barras con el top 10
gdata <- ZONAS |> tibble() |>
  left_join(INDICADORES, by = c("GEOCODIGO", "DESBDT")) |>

```

```

    slice_max(desc(eval(parse(text = PREDICTOR))), n = 10)
gdata |>
  ggplot() +
  aes(y = fct_reorder(DESBDT, eval(parse(text = PREDICTOR))),
      x = eval(parse(text = PREDICTOR))) +
  geom_col(fill = COL1) +
  theme_bw() +
  labs(title = paste0("Top 10 zonas por ", PREDICTOR),
       # subtitle = "Total zonas",
       x = PREDICTOR,
       y = "")

## TABLA

tdata <- ZONAS |> tibble() |>
  left_join(INDICADORES, by = c("GEOCODIGO", "DESBDT"),
            multiple = "all")

tdata |>
  datatable(rownames = FALSE) |>
  formatRound(5:ncol(tdata), dec.mark = ",", mark = ".", digits = 2)
}

```

1.1.3 Paso 5

```

[ ]: ## PROYECCIÓN CAMPAÑA

source("LIBRERIAS.R")

source("VARIABLES.R")

## ZONAS
file_name <- "ZONAS.json"
ZONAS <- st_read(file_name)

## ESCENARIO
file_name <- "ACTUAL_TS.csv"
ESCENARIO <- read_csv(file_name)

## ESTIMACIÓN MODELO
# Preparación objeto tsibble
actualts <- ESCENARIO |>
  mutate(tweek = make_yearweek(ano, semana)) |>

```

```

as_tsibble(key = GEOCODIGO, index = tweek) |>
fill_gaps()

# Ajuste modelo ARIMA
mod_04_ts <- actualts |>
  model(arima = ARIMA(n_vacunas))

## PREDICCIÓN
h <- as.numeric(make_yearweek(min(actualts$ano) + 1, SEMANA_FIN) -
  ↪min(actualts$tweek))

prediction <- mod_04_ts |> forecast(h = h) |> hilo(level = CONF.LEVEL)

## ARREGLAR DATOS Y GUARDAR
PREDICCION <- prediction |>
  mutate(across(last_col(), ~paste(.x$lower, .x$upper, sep = ";"))) |>
  separate(`90%`, into = c("lower", "upper"), sep = ";") |>
  as_tibble() |>
  mutate(n_vacunas = .mean,
         ano = year(tweek),
         semana = isoweek(tweek),
         dato = "Predicción") |>
  select(GEOCODIGO, ano, semana, n_vacunas, lower, upper, dato) |>
  bind_rows(ESCENARIO |>
    mutate(dato = "Actual"))

write_csv(PREDICCION, "PREDICCION_TS.csv")

## VISUALIZACIÓN

## Se muestra solo la variable respuesta

## MAPA

ldata <- ZONAS |>
  left_join(PREDICCION, by = c("GEOCODIGO"),
            multiple = "all") |>
  group_by(GEOCODIGO, DESBDT) |>
  summarise(n_vacunas = sum(n_vacunas, na.rm = TRUE), .groups = "drop")

# COL1 <- rgb(33/255, 150/255, 243/255)
pal <- colorNumeric(palette = "Blues",
                   domain = ldata$n_vacunas)

ldata |>

```

```

leaflet() |>
addTiles() |>
addPolygons(color = "#444444",
             weight = 1,
             smoothFactor = 0.5,
             fillOpacity = 1,
             fillColor = ~pal(n_vacunas),
             highlightOptions = highlightOptions(color = "white", weight = 2,
                                                  bringToFront = TRUE),
             popup = ~paste0(DESDT, " (", GEOCODIGO, ")"),
             label = ~paste0(round(n_vacunas), " vacunas")) |>
addLegend("bottomright",
          pal = pal,
          values = ~n_vacunas,
          title = "Número de vacunas<br/>Predicción resto campaña",
          labFormat = labelFormat(big.mark = " "),
          opacity = 1
)

## SERIE

if(is.na(ZONA)){
  sdata <- PREDICCION |>
  mutate(ano_semana = paste0(ano, "-", semana),
         fecha = as.Date(parse_date_time(paste(ano, semana, 1, sep="/"), 'Y/W/
↪W')))) |>
  group_by(fecha, dato) |>
  summarise(n_vacunas = sum(n_vacunas, na.rm = TRUE), .groups = "drop")
  NZONA <- NA
} else{
  sdata <- PREDICCION |>
  filter(GEOCODIGO == ZONA) |>
  # select(GEOCODIGO, tweek, .mean) |>
  mutate(ano_semana = paste0(ano, "-", semana),
         fecha = as.Date(parse_date_time(paste(ano, semana, 1, sep="/"), 'Y/W/
↪W')))) |>
  group_by(fecha, dato) |>
  summarise(n_vacunas = sum(n_vacunas, na.rm = TRUE), .groups = "drop")
  NZONA <- ZONAS |>
  filter(GEOCODIGO == ZONA) |>
  pull(DESDT)
}

p <- sdata |>
ggplot() +

```



```

aes(x = fecha,
    y = n_vacunas) +
geom_line(aes(col = dato)) +
labs(x = "Semana",
     y = "Total vacunas",
     col = "Tipo de dato") +
theme_bw() +
theme(plot.margin = unit(c(1.2, 1, 1, 1), "cm"))
ggplotly(p) |>
  layout(title = list(text = paste0("Predicción resto campaña ",
    min(year(sdata$fecha)), "/",
    min(year(sdata$fecha)) + 1,
    "<br><sup>",
    if_else(is.na(ZONA), "Total zonas",
      paste0("Zona ", ZONA,
        " (", NZONA, ")")),
    "</sup>"),
    x = 0.05))

## TABLA

PREDICCION |> right_join(tibble(ZONAS) |> select(2:3), by = "GEOCODIGO") |>
  group_by(GEOCODIGO, DESBDT, dato) |>
  summarise(n_vacunas = sum(n_vacunas, na.rm = TRUE), .groups = "drop") |>
  datatable(rownames = FALSE, colnames = c("Código zona", "", "Tipo de dato",
  ↪ "Total predicción vacunas escenario")) |>
  formatRound(4, dec.mark = ",", mark = ".", digits = 0)

```

1.1.4 Paso 6 data

```

[ ]: ## MAPA DE RIESGO

source("LIBRERIAS.R")

source("VARIABLES.R")

## DATOS

data_04_completo <- read_csv("CU_04_05_19_vacunacion_gripe_completo.csv")

capacidad <- data_04_completo |>
  group_by(GEOCODIGO) |>
  summarise(capacidad = mean(capacidad_zona, na.rm = TRUE))

write_csv(capacidad, "CAPACIDAD.csv")

```

```

## GEO
mapSpain::esp_get_prov("28") |>
  st_bbox() |>
  st_as_stars()
st_bbox(RIESGO, crs = "EPSG:4258") |>
  st_as_stars(dx = 100)

```

1.1.5 Paso 6

```

[ ]: ## MAPA DE RIESGO

source("LIBRERIAS.R")

source("VARIABLES.R")

## ZONAS
file_name <- "ZONAS.json"
ZONAS <- st_read(file_name)

## CAPACIDAD
file_name <- "CAPACIDAD.csv"
CAPACIDAD <- read_csv(file_name)

## PROYECCIÓN CAMPAÑA ANTERIOR
file_name <- "PREDICTION.csv"
PROYECCION <- read_csv(file_name)

## PREDICCIÓN CAMPAÑA EN CURSO
file_name <- "PREDICCION_TS.csv"
PREDICCION <- read_csv(file_name)

## UNIÓN DE DATOS Y CÁLCULO DEL RIESGO
RIESGO <- ZONAS |>
  inner_join(
    PROYECCION |>
      group_by(GEOCODIGO) |>
      summarise(fit = sum(fit, na.rm = TRUE)),
    by = "GEOCODIGO") |>
  inner_join(
    PREDICCION |>
      group_by(GEOCODIGO) |>

```

```

    summarise(n_vacunas = sum(n_vacunas, na.rm = TRUE)),
    by = "GEOCODIGO") |>
inner_join(
  CAPACIDAD, by = "GEOCODIGO"
) |>
mutate(saturacion_proy = fit/capacidad,
       saturacion_pred = n_vacunas / capacidad)

RIESGO.cent <- st_centroid(RIESGO)

# RIESGO.cent |>
#   ggplot(aes(col = saturacion_proy)) +
#   geom_sf()

## VARIOGRAMA Y KRIGING

# Malla

grd <- st_bbox(RIESGO) |>
  st_as_stars() |>
  st_crop(RIESGO)
# grd

## Variograma empírico robusto
v0 <- variogram(saturacion_proy ~ 1,
               RIESGO.cent |> drop_na(),
               cressie = TRUE)

## Modelo variograma
v.m <- fit.variogram(v0, vgm(c("Exp", "Mat", "Sph", "Ste", "Gau")),
                    fit.kappa = TRUE)

## kriging
b <- krige(saturacion_proy ~ 1,
          locations = RIESGO.cent |> drop_na(),
          newdata = grd,
          model = v.m)

## VISUALIZACIÓN

pal <- colorNumeric(c("#FFFFCC", "#41B6C4", "#EE2088"), b$var1.pred,
                  na.color = "transparent")

leaflet() |>
  addTiles() |>
  addStarsImage(b, colors = pal, opacity = 0.8) |>
  addLegend(pal = pal, values = b$var1.pred,

```

```
        title = "Riesgo de saturación")  
  
write_stars(b, "kk.tif")
```