

Serie de tiempo - Fecha de creación de FRE

Daniel S. Parra G.

01-01-2021

```
## Propósito del Script: Fecha de creación de FRE
##
##
## Copyright (c) Fondo Nacional de Estupefacientes, 2021
##
## Email: dsparra@minsalud.gov.co

require(vistime)

## Loading required package: vistime
## Warning: package 'vistime' was built under R version 3.6.3

require(plotly)

## Loading required package: plotly
## Warning: package 'plotly' was built under R version 3.6.3
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.6.3
##
## Attaching package: 'plotly'
## The following object is masked from 'package:ggplot2':
##
##   last_plot
## The following object is masked from 'package:stats':
##
##   filter
## The following object is masked from 'package:graphics':
##
##   layout

require(tidyverse)

## Loading required package: tidyverse
## Warning: package 'tidyverse' was built under R version 3.6.3
## -- Attaching packages ----- tidyverse 1.3.0 --
## v tibble  3.1.0      v dplyr    1.0.5
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1
## v purrr   0.3.4
```

```

## Warning: package 'tibble' was built under R version 3.6.3
## Warning: package 'tidyr' was built under R version 3.6.3
## Warning: package 'readr' was built under R version 3.6.3
## Warning: package 'purrr' was built under R version 3.6.3
## Warning: package 'forcats' was built under R version 3.6.3

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks plotly::filter(), stats::filter()
## x dplyr::lag() masks stats::lag()

require(lubridate)

## Loading required package: lubridate
## Warning: package 'lubridate' was built under R version 3.6.3
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

df <- read_csv(file.path('data', 'processed', '001_Herramienta_Procesada.csv'))

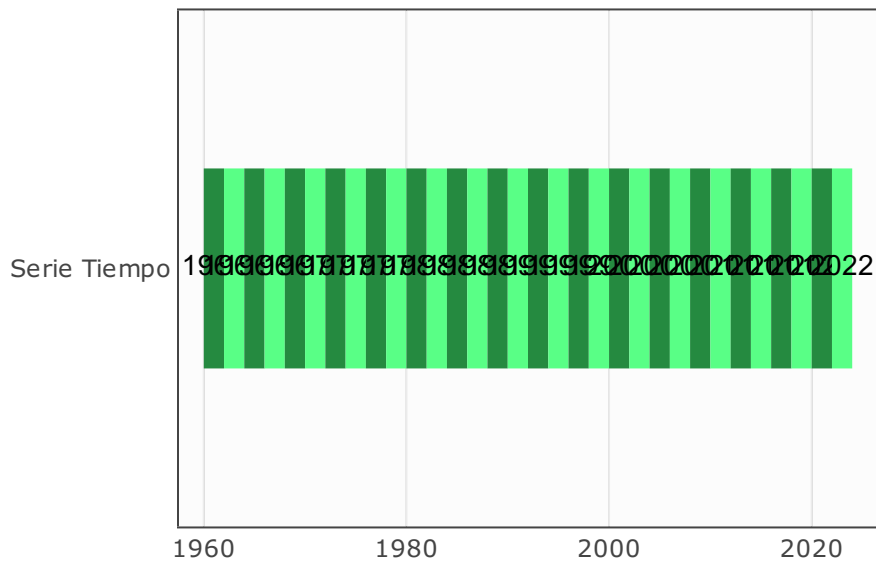
##
## -- Column specification -----
## cols(
##   .default = col_character(),
##   `Marca temporal` = col_datetime(format = ""),
##   `Fecha de la visita` = col_datetime(format = ""),
##   `Fecha creación FRE` = col_datetime(format = ""),
##   `Si la respuesta a la pregunta anterior fue "otro", indique cual:...19` = col_logical(),
##   `Si la respuesta a la pregunta anterior fue "otro", indique cual:...24` = col_logical(),
##   `Si la respuesta a la pregunta anterior fue "otro", indique cual:...27` = col_logical(),
##   `Si la respuesta a la pregunta anterior fue "otro", indique cual:...29` = col_logical(),
##   `Si la respuesta a la pregunta anterior fue "otro", indique cual:...34` = col_logical(),
##   `Si la respuesta a la pregunta anterior fue "otro", indique cual:...43` = col_logical(),
##   `3.01 Existencias actuales de recetarios en el FRE` = col_double(),
##   `3.03 Tiempo de duración proyectada de las actuales existencias de recetarios (semanas).` = col_double(),
##   `3.04 N.º folios que tiene el recetario` = col_double(),
##   `3.05 N.º de prescripciones por recetario` = col_double(),
##   `3.06 Costo de adquisición del recetario (COP)` = col_double(),
##   `4.05. ¿Cuántos equipos de cómputo tiene el FRE para el desarrollo de sus actividades?` = col_double(),
##   `Si la respuesta anterior fue otra, indique cual` = col_logical(),
##   `Si la respuesta anterior fue otro, indique cual...179` = col_logical(),
##   CodigoDepartamento = col_double()
## )
## i Use `spec()` for the full column specifications.

year_df <- tibble(start = seq(1960, 2022, 2),
                  end = seq(1962, 2024, 2)) %>%
  mutate(start = as.character(start)) %>%
  add_column(color = rep(c('#258a40', '#59ff86'), 32 / 2),
             event = paste0('ID', formatC(1:32, width = 2, flag = '0')),
             group = 'Serie Tiempo') %>%

```

```
mutate(across(c(start, end), list(date = function(x) ymd(paste(x, '01', '01', sep = '-')))))

vistime(year_df,
  col.event = 'start',
  col.start = 'start_date',
  col.end = 'end_date')
```



```
paleta <- RColorBrewer::brewer.pal(12, 'Set3')

p <- df %>%
  select(Departamento_1, `Fecha creación FRE`) %>%
  drop_na() %>%
  mutate(
    start_date = as.Date(`Fecha creación FRE`),
    end_date   = as.Date(`Fecha creación FRE`),
    start = Departamento_1,
    group = 'Fecha de Creación',
    color = map_chr(Departamento_1, ~paleta[runif(1, 1, 12)])
  ) %>%
  bind_rows(year_df) %>%
  vistime(col.event = 'start',
    col.start = 'start_date',
    col.end = 'end_date')
```

```

pp <- plotly::plotly_build(p)

marker_idx <- which(purrr::map_chr(pp$$data, "mode") == "markers")
for(i in marker_idx) pp$$data[[i]]$marker$size <- 12
text_idx <- which(purrr::map_chr(pp$$data, "mode") == "text")

for(i in text_idx){
  pp$$data[[i]]$textfont$size <- 10
  pp$$data[[i]]$text <- str_to_sentence(pp$$data[[i]]$text)
}

# pp$$config$displaylogo

text_idx_1 <-
  which((map(pp$$data, "text") %>% str_detect('Tolima')) &
    (map_chr(pp$$data, "mode") == "text"))

## Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
## opts(pattern)): argument is not an atomic vector; coercing
for(i in text_idx_1) pp$$data[[i]]$y <- pp$$data[[i]]$y %>% {. + rnorm(length(.), 0, 0.2)}

pp$$config$displaylogo <- FALSE

pp

```

