

Cadenas de Markov COVID-19

Ramiro Hernandez, Fransisco Melchor, Salvador Caracoza, Gerardo Chávez

2023-05-17

```
library(markovchain)

## Package:  markovchain
## Version:  0.9.1
## Date:     2023-01-20
## BugReport: https://github.com/spedygiorgio/markovchain/issues
# Instalar y cargar los paquetes necesarios
library(dplyr)

##
## 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

library(readr)
library(curl)

## Using libcurl 7.84.0 with Schannel

##
## Attaching package: 'curl'

## The following object is masked from 'package:readr':
##
##   parse_date

library(markovchain)

# Descargar y descomprimir el archivo ZIP
url <- "https://datosabiertos.salud.gob.mx/gobmx/salud/datos_abiertos/datos_abiertos_covid19.zip"
destfile <- "datos_abiertos_covid19.zip"
curl_download(url, destfile)
unzip(destfile, exdir = "datos_abiertos_covid19")

# Leer el archivo CSV
csv_file <- list.files("datos_abiertos_covid19", pattern = "*.csv", full.names = TRUE)
covid_data <- read_csv(csv_file, col_types = cols())

## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
```

```

##   dat <- vroom(...)
##   problems(dat)

# Contar los registros existentes
total_registros <- nrow(covid_data)
cat("Total de registros: ", total_registros, "\n")

## Total de registros: 14560769

# Contar los registros en la columna "FECHA_DEF" que sean diferentes a "9999-99-99"
registros_fecha_def <- covid_data %>%
  filter(FECHA_DEF != 9999-99-99) %>%
  nrow()

print(total_registros)

## [1] 14560769

print(registros_fecha_def)

## [1] 93244

cat("Registros con fecha de defunción diferente a '9999-99-99': ", registros_fecha_def, "\n")

## Registros con fecha de defunción diferente a '9999-99-99': 93244

Sanos <- 126014024 #Población de México según el INEGI al 2020
Infectados <- total_registros
Muertos <- registros_fecha_def
Recuperados <- Infectados - Muertos

TasaInfeccion <- Infectados/Sanos
TasaRecuperacion <- Recuperados/Infectados
TasaMuerte <- Muertos/Infectados

TasaInfeccion

## [1] 0.1155488

TasaRecuperacion

## [1] 0.9935962

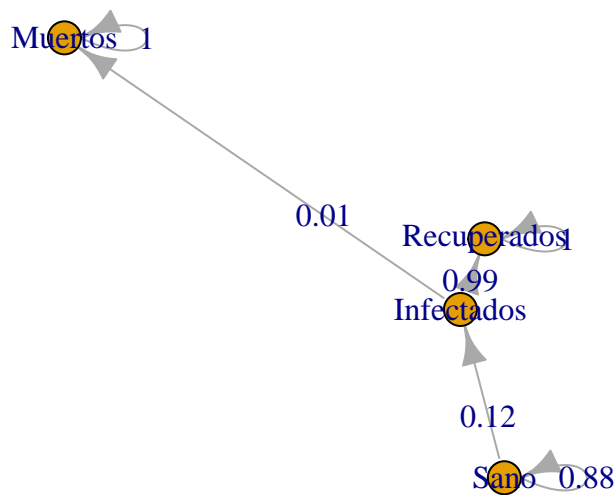
TasaMuerte

## [1] 0.006403783

# Crear una matriz de transición
NombredeEstados3 = c("Sano", "Infectados", "Recuperados", "Muertos")
matrix_transicion <- new("markovchain",
  transitionMatrix = matrix(c(1 - TasaInfeccion, TasaInfeccion, 0, 0,
0, 1 - TasaRecuperacion - TasaMuerte,
TasaRecuperacion, TasaMuerte,
0, 0, 1, 0,
0, 0, 0, 1),
byrow = TRUE, nrow = 4, dimnames = list(NombredeEstados3, NombredeEstados3)))

plot(matrix_transicion)

```



```
summary(matrix_transicion)
```

```
## Unnamed Markov chain Markov chain that is composed by:
## Closed classes:
## Recuperados
## Muertos
## Recurrent classes:
## {Recuperados},{Muertos}
## Transient classes:
## {Sano},{Infectados}
## The Markov chain is not irreducible
## The absorbing states are: Recuperados Muertos
```

```
for (i in 1:100) {
  # Defina el número de pasos de tiempo que desea simular
  n_pasos <- 100 #días

  # Simule la cadena de Markov
  simulacion <- rmarkovchain(n = n_pasos, object = matrix_transicion, t0 = "Sano")

  # Ver la simulación
  print(i)
  print(simulacion)
}
```

```
## [1] 1
## [1] "Sano" "Sano" "Sano" "Sano" "Sano"
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
transitionProbability(object = matrix_transicion, t0 = "Infectados", t1 = "Muertos")
```

```
## [1] 0.006403783
```