

Covid 19 en México

Murpholinox Peligro

16 07 2020

```
# Carga los paquetes necesarios.
library(ggplot2)
#library(ggdark) # No necesario si solo se crea el PDF.
library(latex2exp)
library(lubridate)
library(dplyr)
# Configura el directorio de trabajo.
setwd("/home/murphy/Repos/plotcovid19mx")

# Descarga datos del European CDC.
wget -O full.csv https://opendata.ecdc.europa.eu/covid19/casedistribution/csv
# Obtiene las líneas correspondientes a México.
grep Mex full.csv > mex.csv
# Solo ocupamos la columna 1 y 6 (fecha y decesos por día).
awk -F "," '{print $1,"" $6}' mex.csv > clean.csv
# Añade nombre a las columnas.
echo "fecha,decesos" >> clean.csv
# Revierte los datos en la lista.
tac clean.csv > clean_r.csv
```

```
## --2020-07-16 19:54:42-- https://opendata.ecdc.europa.eu/covid19/casedistribution/csv
## Resolving opendata.ecdc.europa.eu (opendata.ecdc.europa.eu)... 212.181.0.63
## Connecting to opendata.ecdc.europa.eu (opendata.ecdc.europa.eu)|212.181.0.63|:443... connected.
## HTTP request sent, awaiting response... 301 Moved Permanently
## Location: https://opendata.ecdc.europa.eu/covid19/casedistribution/csv/ [following]
## --2020-07-16 19:54:43-- https://opendata.ecdc.europa.eu/covid19/casedistribution/csv/
## Reusing existing connection to opendata.ecdc.europa.eu:443.
## HTTP request sent, awaiting response... 200 OK
## Length: 2041865 (1.9M) [application/octet-stream]
## Saving to: 'full.csv'
##
##      OK ..... 2% 122K 16s
##    50K ..... 5% 238K 12s
##   100K ..... 7% 238K 10s
##   150K ..... 10% 242K 9s
##   200K ..... 12% 241K 9s
##   250K ..... 15% 243K 8s
##   300K ..... 17% 5.54M 7s
##   350K ..... 20% 250K 7s
##   400K ..... 22% 247K 6s
##   450K ..... 25% 245K 6s
##   500K ..... 27% 238K 6s
##   550K ..... 30% 3.18M 5s
```

```
## 600K ..... 32% 238K 5s
## 650K ..... 35% 248K 5s
## 700K ..... 37% 245K 5s
## 750K ..... 40% 244K 5s
## 800K ..... 42% 2.53M 4s
## 850K ..... 45% 261K 4s
## 900K ..... 47% 245K 4s
## 950K ..... 50% 245K 4s
## 1000K ..... 52% 911K 3s
## 1050K ..... 55% 244K 3s
## 1100K ..... 57% 284K 3s
## 1150K ..... 60% 263K 3s
## 1200K ..... 62% 257K 3s
## 1250K ..... 65% 905K 2s
## 1300K ..... 67% 286K 2s
## 1350K ..... 70% 243K 2s
## 1400K ..... 72% 244K 2s
## 1450K ..... 75% 1.37M 2s
## 1500K ..... 77% 240K 2s
## 1550K ..... 80% 278K 1s
## 1600K ..... 82% 256K 1s
## 1650K ..... 85% 239K 1s
## 1700K ..... 87% 1.20M 1s
## 1750K ..... 90% 239K 1s
## 1800K ..... 92% 271K 1s
## 1850K ..... 95% 269K 0s
## 1900K ..... 97% 239K 0s
## 1950K ..... 100% 981K=6.9s
```

```
##
```

```
## 2020-07-16 19:54:50 (289 KB/s) - 'full.csv' saved [2041865/2041865]
```

```
# Carga los datos limpios a R.
```

```
datos <- read.csv("~/Repos/plotcovid19mx/clean_r.csv")
```

```
# Cambia el formato de la fecha de d/m/y a y-m-d.
```

```
datos$newdate <- lubridate::dmy(datos$fecha)
```

```
# Crea una nueva variable con un nuevo formato para la fecha (a números enteros).
```

```
# Esto se hace porque x con formato de fecha impide que el algoritmo para
```

```
# obtener el modelo exponencial llegue a una solución.
```

```
xmax <- max(length(datos$fecha))
```

```
datos$number <- seq(0,xmax-1)
```

```
# Necesitamos los días del brote en México (después del 20-marzo).
```

```
smalldf<-datos %>%
```

```
  filter(number >= 72)
```

```
# Ordena los datos a usar en un tibble.
```

```
x<-smalldf$number
```

```
x<-x-71
```

```
y<-smalldf$decesos
```

```
y2<-y+1
```

```
nice<-tibble(x,y)
```

```
# Se crean dos data frames, porque el modelo lineal con logaritmos se quiebra
```

```
# al tener ceros en y, por eso tenemos y2.
```

```
dummy<-tibble(x,y2)
```

```
# Guarda datos finales en formato CSV.
```

```
write.csv(nice, file=~ /Repos/plotcovid19mx/nice.csv")
```

```

# Crea una gráfica base.
p <- ggplot(data = nice, aes(x=x, y=y)) + geom_point()

# Crea modelo lineal con el data frame dummy que contiene y2=y+1.
linm <- lm(log(y2)~x, data = dummy)
# Obtiene los parámetros del modelo lineal.
a1 <- exp(coef(linm)[1])
b1 <- coef(linm)[2]
# Aplica los parámetros del modelo lineal en la creación del modelo exponencial.
expm <- nls(y ~ a * exp(b * x), start = list(a=a1, b=b1), data = nice)
summary(expm)

##
## Formula: y ~ a * exp(b * x)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## a 83.632517  14.528331   5.757 7.03e-08 ***
## b  0.019144   0.001783  10.734 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 172 on 117 degrees of freedom
##
## Number of iterations to convergence: 12
## Achieved convergence tolerance: 4.278e-06

a2 <- coef(expm)[1]
b2 <- coef(expm)[2]
# Obtiene el número de decesos acumulados
cum <- sum(y)

# Agrega el ajuste exponencial a la gráfica base,
p2 <- p +
stat_smooth(method = 'lm', aes(colour = 'linear'), se = FALSE) +
stat_smooth(method = 'lm', formula = y ~ poly(x,2), aes(colour = 'polynomial'), se= FALSE) +
stat_smooth(method = 'nls', formula = y ~ a * log(x) +b, aes(colour = 'logarithmic'), se = FALSE, start =
stat_smooth(method = 'nls', formula = y ~ a*exp(b *x), aes(colour = 'exponential'), se = FALSE, start =
# stat_smooth(method = 'nls', formula = y ~ a * exp(b * x), se=FALSE,
# method.args = list(start = list(a = a2, b = b2))) +
# la ecuación de la exponencial,
# annotate("label", x = 30, y = 700,
# label = TeX(sprintf("$y = %.2f e ^{%.2fx}$", a2, b2))) +
# una etiqueta para el número de decesos acumulados,
annotate("label", x = 30, y = 900,
label = TeX(sprintf("Decesos acumulados, %d", cum))) +
# los títulos necesarios,
ylab("Decesos") + xlab("Día") +
ggtitle("Decesos por Covid-19 en México, a partir del 20 de marzo.")
p2

```

Decesos por Covid-19 en México, a partir del 20 de marzo.

