Proyecto 2

Rodrigo Zepeda (rodrigo.zepeda@itam.mx)

Septiembre 2021

NOTA Utilizo un estimador y un método para generar intervalos de confianza que NO hemos visto en clase justo para que no se empalme con el trabajo de nadie. En sus ejemplos, usen otros estimadores (para así garantizar que no sólo copy-pastearon mi código)

```
library(rtweet)
library(tidyverse)
library(magrittr)
```

Objetivo

En este proyecto busco responder la siguiente pregunta (en su proyecto son 3 preguntas):

> ¿Cuántos tweets se generan en México?

Para responder la pregunta consideraré a todo usuario registrado en Twitter como una persona. En particular, para restringir los resultados a México utilizo las coordenadas del país con el API de Google de acuerdo a como se especifica en este link

```
mexico_coord <- lookup_coords(address = "mexico",
                              components = "country:Mexico",
                              apikey = Sys.getenv("maps.apikey"))
```

Se obtiene entonces una muestra de tamaño n del total de Tweets generados desde esta localización durante 1 minuto:

```
#Se obtiene la muestra por 1 minuto
rt <- stream_tweets(geocode = mexico_coord, timeout = 60)
```

```
## Streaming tweets for 60 seconds...
```

```
## Finished streaming tweets!
```

> **NOTA** Recordamos que la app de streaming nos da el 1% de los Tweets que se generan.

De ahí obtenemos a los repetidos y al total de la muestra:

```
unicos      <- rt %>% distinct(user_id) %>% tally() %>% as.numeric()
total       <- rt %>% tally() %>% as.numeric()
```

El estimador que utilizaremos se deduce como sigue: Sean ν la cantidad de usuarios **únicos** obtenidos de la muestra. Sea $\{u_1, u_2, \ldots, u_N\}$ los usuarios de todo el universo. Sea $\mathcal{S}^* = \{u_i | u_i \in \mathcal{S}\}$ el conjunto de usuarios únicos de la muestra $\mathcal{S}$. Luego utilizamos el hecho de que:

$$\mathbb{E}\big[\#\mathcal{S}^*\big] = \sum_{i=1}^{N} \mathbb{E}\big[\mathbb{I}_{\mathcal{S}}(u_i)\big] = \sum_{i=1}^{N} \mathbb{P}\big(u_i \in \mathcal{S}\big) = \sum_{i=1}^{N} \left(1 - \left(1 - \frac{1}{N}\right)^{\#\mathcal{S}}\right) = N\left(1 - \left(1 - \frac{1}{N}\right)^{\#S}\right) \approx N(1 - e^{-\#\mathcal{S}/N})$$

Para argumentar que un estimador de N es la solución $\hat{N}$ a:

$$\#S^* = \hat{N}(1 - e^{-\#S/\hat{N}})$$

donde $\#S^*$ es la cantidad de elementos únicos (2167) de la muestra de tamaño $\#S$ (2186).

Dicho estimador no es un estimador insesgado (se basa en una aproximación límite).[1] Resolvemos mediante `uniroot`:

```
fun.optim <- function(N){unicos - N*(1 - exp(-total/N))}
raiz      <- uniroot(fun.optim, lower = 1, upper = 1.e12)$root
N.val     <- round(raiz)
```

De donde tenemos que nuestro N gorrito es: 125,023

Podemos estimar sus intervalos de confianza mediante bootstrap esto es una adaptación de Buckland y Garthwaite[2].

```
nsim         <- 1000
Nvec         <- rep(N.val, nsim)
total_length <- total
for (i in 1:nsim){
  synthetic_population  <- 1:N.val #Muestrear de población tamaño N
  unique_sample         <- total_length
  while (unique_sample == total_length){ #Keep resampling until obtaining different sizes
    bootstrap_sample      <- sample(synthetic_population, total_length, replace = TRUE)
    unique_sample         <- bootstrap_sample %>% unique() %>% length() %>% as.numeric()
  }
  fun.optim    <-  function(N){unique_sample - N*(1 - exp(-total/N))}
  Nvec[i]      <- floor(uniroot(fun.optim, lower = 1, upper = 1.e12)$root)
  N.val        <- floor(Nvec[i])
}
ic <- quantile(Nvec, c(0.025, 0.975))
```

De donde obtenemos que el intervalo de confianza al 95% es [14,008, 2,388,569].

[1] En caso de que fuera insesgado habría que demostrarlo.

[2] **Ojo** No hemos visto bootstrap pero lo veremos aunque no en este caso. No espero que lo hagan así. Lo puse sólo para garantizar que no me copian mi estimador.