Modelo pronóstico elecciones presidenciales Colombia 2022

Sergio Eduardo Calvo Mazuera

Junio 16 2022

1. Cocinero

1.1. Información general

Cocinero: Sergio Eduardo Calvo Mazuera
Twitter: @SCalvo25.

2. Introducción

Se elabora un modelo estadístico para predecir la segunda vuelta de las elecciones presidenciales de Colombia 2022. El modelo consiste en simulación de Monte Carlo utilizando el margen de error y la librería de pronóstico en series de tiempo Prophet. Se utilizan los datos para encuestas de opinión desde las consultas presidenciales del 13 de marzo como datos con un ajuste para generar una serie de tiempo sin más de una observación por fecha.

3. Ingredientes

En esta sección detallo los datos utilizados para la elaboración del modelo. Los datos pueden ser encontrados en el repositorio de github: `https://github.com/SergioECalvoM/Pronostico-Presidencial-2022`.

3.1. Encuestas

Se elabora un achivo de texto con nombre: `EncuestasSV2022.txt` que contiene información sobre los sondeos de intención de voto para la segunda vuelta de las elecciones presidenciales de 2022 en Colombia. El archivo corresponde a los datos de las encuestas públicas con tiempos de recolección posteriores a las consultas presidenciales del 13 de marzo de 2022 para escenarios de segunda vuelta entre los candidatos Gustavo Petro y

Rodolfo Hernández, así como las encuestas posteriores a la primera vuelta del 29 de mayo de 2022. Este archivo cuenta con las siguientes columnas:

- **Pollster**: Nombre de la firma encuestadora.
- **Date**: Corresponde a la fecha mediana de recolección de la información de la encuesta. En caso de una cantidad par de recolección se toma el primer día de la segunda parte (Ejemplo: Si se tienen cuatro días de recolección, se tomará como fecha el tercer día). La única excepción es para los datos del tracking de la firma encuestadora GAD3, se toma el última día de medición.
- **SS**: Tamaño de la muestra. Denotado n para cálculos.
- **MoE**: Márgen de error para muestras cercanas al 50 % con un nivel de confianza del 95 %. Para los datos de Bogotá, este cálculo se realizó utilizando:

$$\text{MOE}_i(0{,}5) = z_{0{,}95}\sqrt{\frac{0{,}5^2}{n}}, \tag{1}$$

  para cada encuesta $i = 1, \dots N$ donde $z_{0,95}$ corresponde al cuantil de la distribución normal tal que para una variable aleatoria $X \sim \mathcal{N}(\mu, \sigma^2)$,la probabilidad $\mathbb{P}\left(X \in [\mu - z_{0,95}\sigma, \mu + z_{0,95}\sigma]\right) = 0{,}95$ (Wonnacott and Wonnacott, 1990).
- **Candidatos**: La proporción de votantes con intención de voto por cada uno de los candidatos a la presidencia que aparecen en el tarjetón presidencial para la segunda vuelta de las elecciones:
  - Gustavo_Petro.
  - Rodolfo_Hernandez.
- **En_Blanco**: La proporción de encuestados con intención de voto por el voto en blanco.
- **None**: La proporción de encuestados con intención de no votar por ninguno.
- **Uncertain**: La proporción de encuestados indecisos.
- **Source**: Cadena informativa o sitio web de donde se extrae la información de cada encuesta.
- **Link**: El enlace a cada una de las encuestas.

3.2. Serie de tiempo

Ahora bien, tomando como base el archivo de encuestas previamente descrito, se elabora un archivo de datos `SerieTemporalSV2022.txt` con las mismas columnas exceptuando Pollster, MoE, Source y Link. Posteriormente, se realiza para el caso de varias encuestas con la misma fecha, un promedio ponderado por el tamaño de muestra. Lo anterior garantiza que para cada fecha se obtenga un registro único que consiste de tamaño de muestra, proporción de la intención de voto para cada candidato, el voto en blanco, indecisos y aquellos que no planean votar por ninguna opción. Los cálculos pueden encontrarse en el archivo de Excel `AjusteSV`.

4. Cocina

4.1. Ajuste y margen de error

El modelo fue elaborado en el paquete estadístico R (R Core Team, 2020). Como primer paso cargamos las librerías necesarias para el manejo de datos: magrittr, plyr, Rmisc y tidyverse; las necesarias para los cálculos del modelo: lubridate, prophet y matrixStats y aquellas utilizadas para la visualización de tablas y gráficas: kableExtra y ggplot2. Adicionalmente, se define la ruta donde se encuentra el archivo `SerieTemporalSV2022.txt` para poder definir un data frame con este.

```
rm(list=ls(all=TRUE))
library(magrittr)
library(plyr)
library(Rmisc)
library(tidyverse)
library(kableExtra)
library(lubridate)
library(prophet)
library(ggplot2)
library(matrixStats)
setwd("C:/Users/sergi/OneDrive/Desktop/42/Work/Research/Election Forecasting
df<-read.table("SerieTemporalSV2022.txt", T)
```

Ahora, se edita el data frame original para calcular las proporciones de intención de voto de lo candidatos y voto en blanco omitiendo la proporción de encuestados que aseguran no tener intención de votar por ninguna de las opciones, pues se considera que dichos encuestados probablemente serán abstencionistas o anularán el voto.

```
clean_df<-df%>%
  mutate(Gustavo_Petro=Gustavo_Petro/(1-None),
         Rodolfo_Hernandez=Rodolfo_Hernandez/(1-None),
```

```
    En_Blanco=En_Blanco/(1-None))
```

Posteriormente, se define la función `SVSimulation` cuyo argumento será el número S de simulaciones que se desee. Definimos tres listas donde se insertarán los resultados de cada simulación en la predicción puntual de voto, y los límites inferiores y superiores del intervalo de confianza del 95 %. Además, se inicia el ciclo `for` para iterar los cálculos S-veces.

```
SVSimulation<-function(S){
  dflist=list()
  dflowlist=list()
  dfupplist=list()
for(i in 1:S){
```

En este punto incorporamos el elemento estocástico del modelo que justifica que sea una simulación iterada S-veces. Considerando que para cada proporción de intención de voto (Candidatos y voto en blanco) se obtiene un diferente margen de error, añadimos al data frame tres columnas con la mitad de un valor aleatorio obtenido de una distribución normal estándar y otras con los margenes de error de cada proporción en cada encuesta. El fundamento teórico es el siguiente:

En una encuesta i el margen de error calculado con un nivel de confianza del 95 % indica que en el 95 % de las muestras la proporción de intención de voto p_j de la opción j se encontrará entre $\mu_j - MOE_i(p_j)$ y $\mu_j - MOE_i(p_j)$, donde μ_j es la estimación puntual y el margen de error se calcula como:

$$MOE_i(p_j) = z_{0,95}\sqrt{\frac{p_j(1-p_j}{n}}, \tag{2}$$

análogo a la ecuación 1 usualmente reportada en las fichas técnicas (Sudman and Bradburn, 1982). Luego, se busca hallar en el 95 % de las simulaciones que la proporción p_j esté en ese intervalo, por lo cual si el valor aleatorio calculado se encuentra a dos desviaciones estándar de la media, donde se concentra el 95 % de la información, es deseable que al ser multiplicado por el margen de error el resultado sea $\pm MOE_i(p_j)$. Este último resultado se añade a cada p_j correspondiente.

```
TS<-clean_df
n_polls=nrow(TS)
TS$GPrand<-rnorm(n_polls)/2
TS$RHrand<-rnorm(n_polls)/2
TS$EBrand<-rnorm(n_polls)/2


TS<-TS%>%
  mutate(GPMoE=qnorm(0.975)*sqrt((Gustavo_Petro*
  (1-Gustavo_Petro))/SS),
```

```
  RHMoE=qnorm(0.975)*sqrt((Rodolfo_Hernandez*
        (1-Rodolfo_Hernandez))/SS),
  EBMoE=qnorm(0.975)*sqrt((En_Blanco*
         (1-En_Blanco))/SS))%>%
  mutate(Gustavo_Petro=Gustavo_Petro+GPrand*GPMoE,
         Rodolfo_Hernandez=Rodolfo_Hernandez+RHrand*RHMoE,
         En_Blanco=En_Blanco+EBrand*EBMoE)
```

Como último paso previo a la predicción con Prophet, transformamos los valores que hayan quedado por debajo de cero posterior al proceso aleatorio de sumar el margen de error, se excluye a los indecisos de la muestra y se normalizan las proporciones para obtener una suma de 1.

```
TS<-TS %>%
  mutate(Gustavo_Petro=if_else(Gustavo_Petro<0, 0, Gustavo_Petro),
         Rodolfo_Hernandez=if_else(
         Rodolfo_Hernandez<0, 0, Rodolfo_Hernandez),
         En_Blanco=if_else(En_Blanco<0, 0, En_Blanco))

TS<-TS%>%
  mutate(Total=Gustavo_Petro+
           Rodolfo_Hernandez+
           En_Blanco)%>%
  mutate(across(Gustavo_Petro:En_Blanco, ~ ./Total))%>%
  mutate(Gustavo_Petro=Gustavo_Petro/Total,
         Rodolfo_Hernandez=Rodolfo_Hernandez/Total,
         En_Blanco=En_Blanco/Total)
```

4.2. Series de tiempo y Prophet

Taylor and Letham (2017) proponen un modelo de pronóstico a escala para series de tiempo con tres componentes, el cual tiene la siguiente especificación:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \tag{3}$$

donde $g(t)$ corresponde a la función de tendencia que modela cambios sin periodicidad en la serie de tiempo; $s(t)$ es un componente de estacionalidad (anual, semanal, diaria); $h(t)$ incorpora efectos de días festivos y ϵ_t corresponde a los cambios en la serie de tiempo no ajustados por el modelo, se asume que $\epsilon_t \sim \mathcal{N}(0,1)$ y son i.i.d. Prophet ya había sido utilizado por Cuervo and Guerrero (2019) como parte de un modelo híbrido para predecir la primera vuelta de las elecciones presidenciales de Colombia en 2018 directamente en los datos de las encuestas de dicha elección. Para las series de tiempo de los datos para cada candidato, se obtiene una tendencia g que consiste de una función lineal a trozos, así como

una estacionalidad semanal, $h(t) = 0$ para todo t. En el caso de este modelo, se utiliza la librería en R de Prophet (Taylor and Letham, 2021) de la siguiente forma:

```
Petro<-TS%>%
    select(Date, Gustavo_Petro)
  colnames(Petro)=c('ds','y')
  head(Petro)
  ModelPetro<-prophet(Petro)
  FuturePetro<-make_future_dataframe(ModelPetro, periods=9)
  tail(FuturePetro)
  ForecastPetro<-predict(ModelPetro, FuturePetro)

  Hernandez<-TS%>%
    select(Date, Rodolfo_Hernandez)
  Hernandez
  colnames(Hernandez)=c('ds','y')
  head(Hernandez)
  ModelHernandez<-prophet(Hernandez)
  FutureHernandez<-make_future_dataframe(ModelHernandez, periods=9)
  tail(FutureHernandez)
  ForecastHernandez<-predict(ModelHernandez, FutureHernandez)

  Blanco<-TS%>%
    select(Date, En_Blanco)
  Blanco
  colnames(Blanco)=c('ds','y')
  head(Blanco)
  ModelBlanco<-prophet(Blanco)
  FutureBlanco<-make_future_dataframe(ModelBlanco, periods=9)
  tail(FutureBlanco)
  ForecastBlanco<-predict(ModelBlanco, FutureBlanco)
```

4.3. Consolidación de datos

Ahora bien, se añade en cada iteración un data frame diferente en cada una de las listas creadas previo al ciclo `for`, que corresponden a los resultados de la iteración para el pronóstico puntual de la proporción de voto de cada opción y los límites inferiores y superiores de los intervalos de confianza de dichos pronósticos. Resalto que dichos data frame son previamente transformados para que los valores negativos se conviertan en cero y el pronóstico puntual es normalizados para obtener una suma de 1 sobre las opciones, cerrandose el mencionado ciclo `for`.

```
DF<-TS%>%
      filter(!row_number() %in% c(1:nrow(TS)))%>%
      add_row(Gustavo_Petro=ForecastPetro$yhat[nrow(ForecastPetro)],
              Rodolfo_Hernandez=ForecastHernandez$yhat
              [nrow(ForecastHernandez)],
              En_Blanco=ForecastBlanco$yhat[nrow(ForecastBlanco)])%>%
      pivot_longer(cols=contains("_"), names_to="nombre"
      values_to="Int_voto")%>%
      mutate(Candidato=case_when(nombre=="Gustavo_Petro"~
      "Gustavo Petro",
            nombre=="Rodolfo_Hernandez"~"Rodolfo Hern ndez",
            nombre=="En_Blanco"~"Voto en Blanco")%>%
               factor())%>%
      group_by(Candidato)%>%
      summarize(Predicci n=Int_voto*100)%>%
      arrange(desc(Predicci n))
    DF$Predicci n[DF$Predicci n<0]<-0
    scale<-function(x) (x/sum(x))*100
    DF<-DF%>%
      mutate(across(!Candidato, scale))
    DFCIupp<-TS%>%
      filter(!row_number() %in% c(1:nrow(TS)))%>%
      add_row(Gustavo_Petro=ForecastPetro$yhat_upper
      [nrow(ForecastPetro)],
      Rodolfo_Hernandez=ForecastHernandez$yhat_upper
             [nrow(ForecastHernandez)],
      En_Blanco=ForecastBlanco$yhat_upper[nrow(ForecastBlanco)])%>%
      pivot_longer(cols=contains("_"), names_to="nombre",
      values_to="Int_voto")%>%
      mutate(Candidato=case_when(nombre=="Gustavo_Petro"~
      "Gustavo Petro",
      nombre=="Rodolfo_Hernandez"~"Rodolfo Hern ndez"
      nombre=="En_Blanco"~"Voto en Blanco")%>%
               factor())%>%
      group_by(Candidato)%>%
      summarize(CISup=Int_voto*100)%>%
      arrange(desc(CISup))
    DFCIupp$CISup[DFCIupp$CISup<0]<-0
    DFCIlow<-TS%>%
      filter(!row_number() %in% c(1:nrow(TS)))%>%
```

```
    add_row(Gustavo_Petro=ForecastPetro$yhat_lower
    [nrow(ForecastPetro)],
    Rodolfo_Hernandez=ForecastHernandez$yhat_lower
    [nrow(ForecastHernandez)],
    En_Blanco=ForecastBlanco$yhat_lower[nrow(ForecastBlanco)])%>%
    pivot_longer(cols=contains("_"), names_to="nombre",
    values_to="Int_voto")%>%
    mutate(Candidato=case_when(nombre=="Gustavo_Petro"~"Gustavo Petro",
    nombre=="Rodolfo_Hernandez"~"Rodolfo Hern ndez",
    nombre=="En_Blanco"~"Voto en Blanco")%>%
              factor())%>%
    group_by(Candidato)%>%
    summarize(CIInf=Int_voto*100)%>%
    arrange(desc(CIInf))
  DFCIlow$CIInf[DFCIlow$CIInf<0]<-0
  dflist[[i]]=DF
  dflowlist[[i]]=DFCIlow
  dfupplist[[i]]=DFCIupp
}
```

Cerrado el ciclo, se procede a convertir cada una de las listas en un data frame utilizando la función `join_all()` sobre cada lista, cambiando el nombre de cada columna para que corresponda al valor de la simulación i y calculando sobre cada opción la mediana sobre las S simulaciones. Como último paso se genera un data frame con el consolidado de pronóstico e intervalos de confianza obtenidos con la mediana sobre sus valores en cada simulación, el cual será el objeto que retorne la función.

```
Result<-join_all(dflist, by="Candidato",
type = "left", match = "all")
namesdf=c("Candidato")
for(i in 1:S){
  namesdf[[i+1]] <- paste("Simulation", i, sep = "")}
colnames(Result)<-namesdf


Result<-Result%>%
  mutate(Promedio=rowMedians(as.matrix(Result
  [,2:ncol(Result)])))%>%
  arrange(desc(Promedio))%>%
  mutate(Pronostico=signif(Promedio, digits=4))

Result<-Result%>%
```

```
  select(c(Candidato, Pronostico))

Resultinf<-join_all(dflowlist, by="Candidato",
type = "left", match = "all")
namesdf=c("Candidato")
for(i in 1:S){
  namesdf[[i+1]] <- paste("Simulation", i, sep = "")}
colnames(Resultinf)<-namesdf


Resultinf<-Resultinf%>%
  mutate(CIInf=rowMedians(as.matrix(Resultinf
  [,2:ncol(Resultinf)])))%>%
  arrange(desc(CIInf))%>%
  mutate(CIInf=signif(CIInf, digits=4))

Resultinf<-Resultinf%>%
  select(c(Candidato, CIInf))

Resultsup<-join_all(dfupplist, by="Candidato",
type = "left", match = "all")
namesdf=c("Candidato")
for(i in 1:S){
  namesdf[[i+1]] <- paste("Simulation", i, sep = "")}
colnames(Resultsup)<-namesdf


Resultsup<-Resultsup%>%
  mutate(CISup=rowMedians(as.matrix(Resultsup
  [,2:ncol(Resultsup)])))%>%
  arrange(desc(CISup))%>%
  mutate(CISup=signif(CISup, digits=4))

Resultsup<-Resultsup%>%
  select(c(Candidato, CISup))
Result<-Result%>%
  left_join(Resultinf)%>%
  left_join(Resultsup)

return(Result)
```

```
}
```

5. Resultados

Se utilizó la función creada para simular 5000 veces la elección con Prophet, y utilizando los votos validos de la primera vuelta presidencial como base se obtienen los valores estimados de votos de cada opción en el pronóstico puntual y los límites de intervalos de confianza. Este resultado se consolida en una tabla `tableaverageSV.html` elaborada usando la librería `kableExtra` y en un gráfico elaborado con `ggplot2` como se muestra a continuación

```
 Simresulttablets<-Simresult%>%
  mutate(Votos=formatC(2173157*Pronostico/100, digits = 0,
  format = "f"))%>%
  mutate(VotosInf=formatC(2173157*CIInf/100, digits = 0,
  format = "f"))%>%
  mutate(VotosSup=formatC(2173157*CISup/100, digits = 0,
  format = "f"))
Simresulttablets<-Simresulttablets%>%
  kable("html",
        digits=2,
        caption = "Predicci n: % y votos por candidato") %>%
  kable_styling(full_width = F) %>%
  footnote(number = c("Autor: Sergio Calvo","Twitter:
  @Scalvo25","Fecha pron stico: 2022-06-14"))%>%
  kable_paper() %>%
  save_kable(file = "tableaverageSV.html", self_contained = T)


dataplot <- data.frame(Candidato <-Simresult$Candidato,
                       F <- Simresult$Pronostico,
                       L <- Simresult$CIInf,
                       U <- Simresult$CISup)

ggplot(dataplot, aes(x = reorder(Candidato, -F), y = F,
colour=Candidato)) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L))+scale_color_manual(
  values=c("purple", "orange", "grey"))+
  ggtitle("Pron stico Segunda Vuelta Presidencial Colombia
  2022 (Prophet)")+
```

```
  ylab("Porcentaje")+xlab("Candidato")+labs(fill = "Candidato")+
  theme(axis.text=element_text(size=6))
ggsave("SegVuelta.png",width = 20, height = 20, units = "cm")
```

5.1. Consolidado

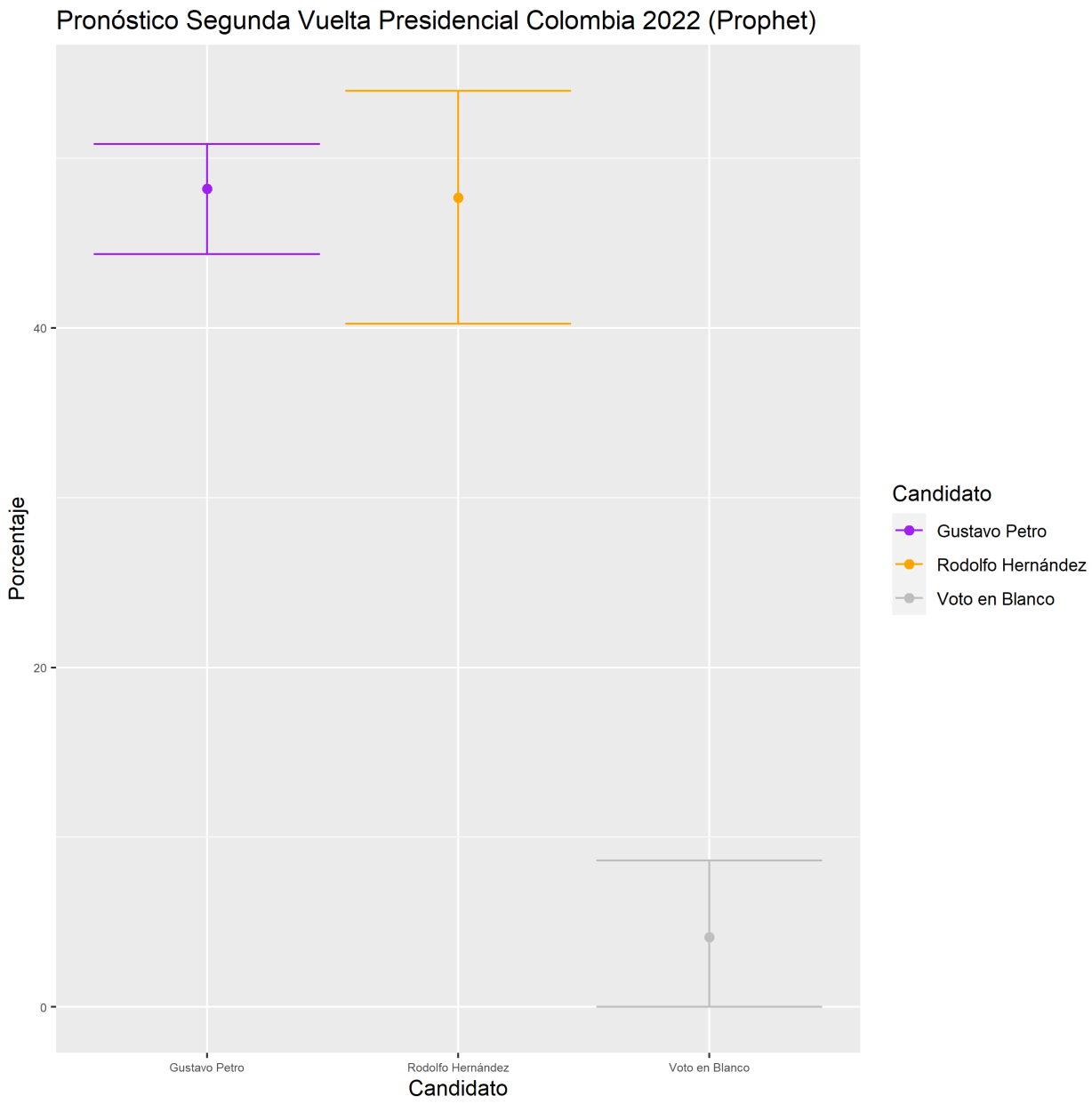
A continuación podemos observar los resultados de las 5000 simulaciones en formato de tabla:

Predicción: % y votos por candidato

Candidato	Pronostico	CIInf	CISup	Votos	VotosInf	VotosSup
Gustavo Petro	48.19	44.36	50.83	1047244	964012	1104616
Rodolfo Hernández	47.69	40.24	53.98	1036379	874478	1173070
Voto en Blanco	4.11	0.00	8.62	89338	0	187239

[1] Autor: Sergio Calvo
[2] Twitter: @Scalvo25
[3] Fecha pronóstico: 2022-06-14

Así mismo, podemos observar la gráfica de los resultados para los intervalos de confianza medianos y la estimación puntual en la siguiente página:



Referencias

M. C. Cuervo and M. A. V. Guerrero. Predicción electoral usando un modelo híbrido basado en análisis sentimental y seguimiento a encuestas: elecciones presidenciales de colombia. *Revista Politécnica*, 15(30):94–104, 2019. ISSN 2256-5353.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL `https://www.R-project.org/`.

S. Sudman and N. Bradburn. *Asking Questions: A Practical Guide to Questionnaire Design*. Jossey Bass, San Francisco, 1982. ISBN 0-87589-546-8.

S. Taylor and B. Letham. *prophet: Automatic Forecasting Procedure*, 2021. URL `https://CRAN.R-project.org/package=prophet`. R package version 1.0.

S. J. Taylor and B. Letham. Forecasting at scale. *PeerJ Preprints*, 2017.

T. Wonnacott and R. Wonnacott. *Introductory Statistics*. Wiley, San Francisco, 5 edition, 1990. ISBN 0-471-61518-8.