

Votantes del rechazo

Matías Deneken & Roberto Cantillán

Librerías

```
library(tidyverse)
library(haven)
library(readxl)
library(stringi)
library(stringr)
library(here)
library(tidymodels)
library(broom)
library(kableExtra)
library(marginaleffects)
library(knitr)
library(stargazer)
library(texreg)
```

Introducción

El plebiscito constitucional chileno del 4 de octubre de 2022 reveló un patrón de votación inesperado entre la población indígena. Contrario a las expectativas, los votantes indígenas no mostraron un apoyo significativamente mayor a la propuesta constitucional en comparación con los votantes no indígenas. Este fenómeno plantea preguntas cruciales sobre los factores que influyen en las decisiones electorales de las comunidades indígenas en Chile. Nuestra investigación se centra en dos preguntas principales:

- ¿Qué factores predicen el voto de rechazo entre las personas indígenas?
- ¿Existen diferencias significativas entre los predictores del voto para poblaciones indígenas y no indígenas?

Para abordar estas cuestiones, proponemos un análisis comparativo utilizando modelos de regresión logística. Nuestro objetivo es identificar las variables socioeconómicas, culturales y políticas que tienen mayor influencia en la decisión de voto, comparando específicamente entre votantes indígenas y no indígenas.

Datos

```
# Panel data.
getwd()

## [1] "/home/rober/Documents/elri_code_analysis/code/rechazo-constitutional"

load(here("data/BDD_ELRI_LONG.RData"))
elri <- BDD_ELRI_LONG

# Comunas cercanas al conflicto.
#09201 #Angol
#08202 #Arauco
```

```

#08203 #Canete
#09121 #cholchol
#09203 #curacautin
#08205 #curanilahue
#09104 #currarhue
#09204 #Ercilla
#09105 #Freire
#10104 #Fresia
#09106 #Galvarino
#09107 #Gorbea
#09108 #Lautaro
#08201 #Lebu
#09109 #Loncoche
#09205 #Lonquimai
#09111 #Nueva imperial
#09112 #Padre de las casas
#09101 #Temuco
#08207 #Tirúa
#09211 #Victoria
#09119 #Vilcún

```

Preparamos datos I

```

elri <- elri %>% filter(ano == 2023) %>%
  dplyr::mutate (mujer = case_when(g2 == 1 ~ "0",
                                   g2 == 2 ~ "1")) %>%
  dplyr::mutate (edad = case_when(g18 %in% 18:24 ~ "18_24",
                                   g18 %in% 25:34 ~ "25_34",
                                   g18 %in% 35:44 ~ "35_44",
                                   g18 %in% 45:54 ~ "45_54",
                                   g18 %in% 55:64 ~ "55_64",
                                   g18 %in% 65:89 ~ "65+")) %>%
  mutate(indigena_es = case_when(a1 >= 10 ~ "No indígena",
                                   a1 <= 12 ~ "Indígena")) %>%
  select(edad,
         mujer,
         urbano_rural,
         d14, #Voto apruebo - rechazo
         indigena_es, # ser indígena
         d1_1, #Grados de conflicto con el estado
         d1_2, # Grado de conflicto inter étnico.
         c2, #Cuanto confía en los chilenos no indígenas
         c5, #Confianza hacia pueblos originarias
         c23, #Diferencia justa o injusta entre indígenas
         # d11, #Posición política. Mejor no.
         a6, #identificación con chile
         a4, #identificacion pp.oo
         d6_1, #identificación con la causa indígena
         c7_2, #Frecuencia de contacto con no indígenas.
         c7_3, #experiencia positiva con no indígena
         c14, #experiencia positiva con indígenas
         c13, #Frecuencia de contacto con indígenas
         comuna

```

```

    )

elri <- elri %>% filter(d14 <= 2) #Filtrar por apruebo/rechazo.

# Anreponer el 0

elri$comuna <- ifelse(nchar(elri$comuna) == 4,
                     str_pad(elri$comuna,
                             width = 5, pad = "0"),
                     elri$comuna)

# Suponiendo que tu dataframe se llama ELRI y la variable de interés es comuna
# Crear un vector con los códigos de las comunas que están cerca del conflicto
comunas_conflicto <- c("09201", "08202", "08203", "09121", "09203", "08205",
                      "09104", "09204", "09105", "10104", "09106", "09107",
                      "09108", "08201", "09109", "09111", "09205", "09112",
                      "09101", "08207", "09211", "09119")

# Crear la nueva variable cerca_conflicto
elri$cerca_conflicto <- ifelse(elri$comuna %in% comunas_conflicto,
                              "cerca_conflicto", "lejos_conflicto")

# Ver el resultado
#print(elri)
#elri %>% select(comuna, cerca_conflicto)

```

El descriptivo aprecia que las personas que viven cerca del conflicto tienden a votar más por el rechazo.

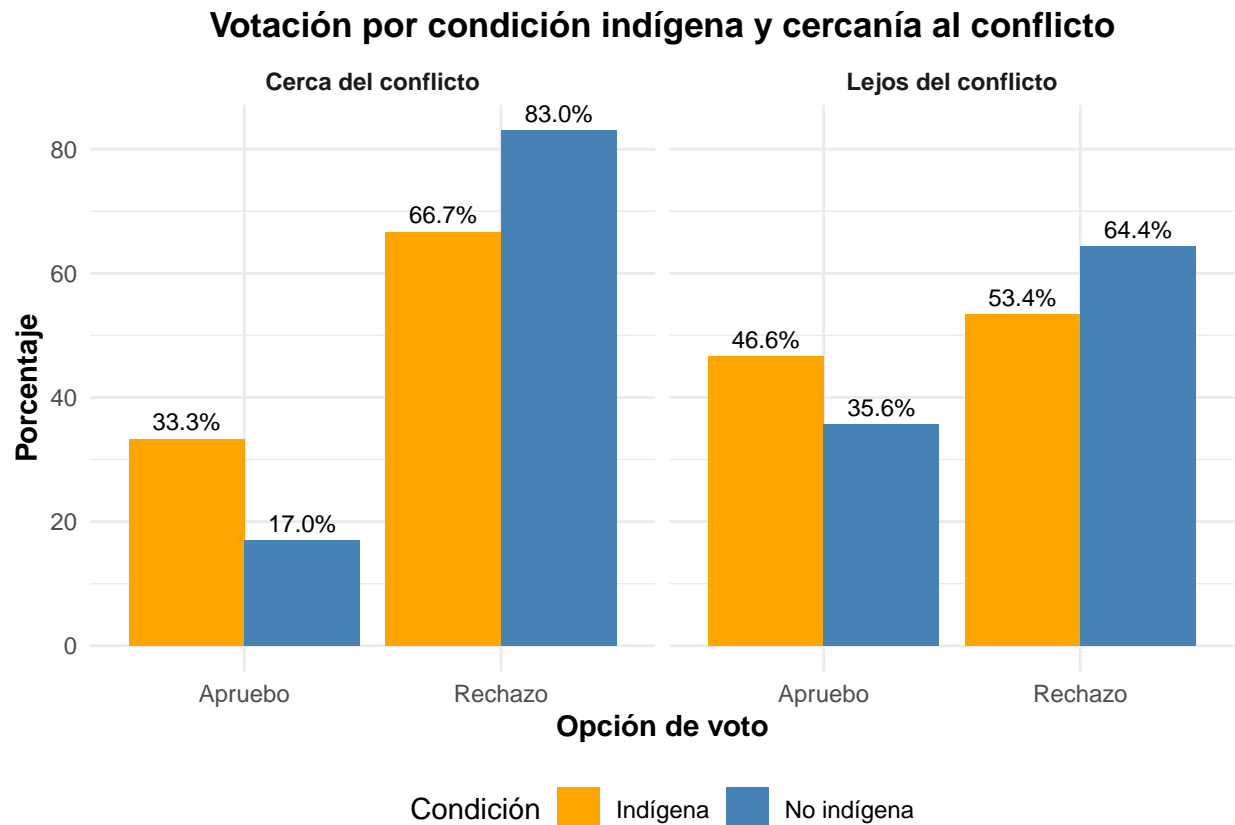
plot

```

elri %>% select(cerca_conflicto, indigena_es, d14) %>%
  group_by(indigena_es, cerca_conflicto, d14) %>%
  summarise(count = n()) %>%
  mutate(percentage = count / sum(count) * 100) %>%
  ggplot(aes(x = factor(d14), y = percentage, fill = indigena_es)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~ cerca_conflicto,
            labeller = labeller(cerca_conflicto = c("cerca_conflicto" = "Cerca del conflicto",
                                                    "lejos_conflicto" = "Lejos del conflicto")))) +
  scale_fill_manual(values = c("Indígena" = "#FFA500", "No indígena" = "#4682B4")) +
  scale_x_discrete(labels = c("1" = "Apruebo", "2" = "Rechazo")) +
  labs(title = "Votación por condición indígena y cercanía al conflicto",
       x = "Opción de voto",
       y = "Porcentaje",
       fill = "Condición") +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.title = element_text(face = "bold"),
        strip.text = element_text(face = "bold")) +
  geom_text(aes(label = sprintf("%.1f%%", percentage)),
            position = position_dodge(width = 0.9),

```

```
vjust = -0.5,  
size = 3)
```



Otra literatura sugiere que la frecuencia de contacto y experiencia positiva podría influir. Personas con mayor experiencia positiva, tenderían a votar a aprueba

Preparamos datos II

```
elri_prep <- elri %>%  
  # Convertir "88" y "99" a NA en todas las columnas  
  mutate(across(everything(), ~if_else(. %in% c("88", "99"), NA_character_,  
                                         as.character(.)))) %>%  
  # Convertir variables de carácter a factor  
  mutate(across(where(is.character), as.factor)) %>%  
  # Convertir d14 a factor y recodificar  
  mutate(d14 = as.factor(d14)) %>%  
  mutate(d14 = fct_recode(d14, "Apruebo" = "1", "Rechazo" = "2")) %>%  
  # Convertir variables sociodemográficas a factor  
  mutate(  
    mujer = as.factor(mujer),  
    edad = factor(edad, levels = c("18_24", "25_34", "35_44", "45_54", "55_64", "65+")),  
    urbano_rural = as.factor(urbano_rural)  
  ) %>%  
  # Convertir otras variables predictoras a factores ordenados  
  mutate(across(c(d1_1, d1_2, c2, c5, c23, a6, a4, d6_1, c7_2, c7_3, c13, c14),  
                ~factor(.)))
```

```

# El resto del código permanece igual
# Función para crear y ajustar modelo
fit_logistic <- function(data, formula) {
  logistic_model <- logistic_reg() %>%
    set_engine("glm")

  workflow <- workflow() %>%
    add_model(logistic_model) %>%
    add_formula(formula)

  fit(workflow, data)
}

# Función para extraer y formatear resultados
extract_results <- function(model) {
  model %>%
    extract_fit_parsnip() %>%
    tidy() %>%
    mutate(odds_ratio = exp(estimate),
           lower_ci = exp(estimate - 1.96 * std.error),
           upper_ci = exp(estimate + 1.96 * std.error)) %>%
    select(term, estimate, odds_ratio, lower_ci, upper_ci, p.value)
}

```

Modelos simples (sin teracciones.)

```

# 1. Modelo base con variables sociodemográficas
model_base <- fit_logistic(elri_prep, d14 ~ mujer + edad + urbano_rural)

# 2. Modelo con variables de identidad y confianza
model_identity <- fit_logistic(elri_prep, d14 ~ mujer + edad + urbano_rural +
                              indigena_es + a6 + a4 + d6_1 + c2 + c5)

# 3. Modelo con variables de conflicto
model_conflict <- fit_logistic(elri_prep, d14 ~ mujer + edad + urbano_rural +
                              d1_1 + d1_2 + cerca_conflicto)

# 4. Modelo con variables de contacto intergrupar
model_contact <- fit_logistic(elri_prep, d14 ~ mujer + edad + urbano_rural +
                              c7_2 + c7_3 + c13 + c14)

# Extraer y mostrar resultados
results_base <- extract_results(model_base)
results_identity <- extract_results(model_identity)
results_conflict <- extract_results(model_conflict)
results_contact <- extract_results(model_contact)

# Mostrar resultados
#print("Modelo Base:")
#print(results_base)
#
#print("Modelo de Identidad y Confianza:")

```

```
#print(results_identity)
#
#print("Modelo de Conflicto:")
#print(results_conflict)
#
#print("Modelo de Contacto Intergrupar:")
#print(results_contact)
```

Tabla

```
texreg(list(model_identity, model_conflict, model_contact),
        float.pos = "H", return.string = TRUE, bold = 0.05, stars = 0,
        custom.note = "Coefficients with  $p < 0.05$  in \\textbf{bold}.",
        digits = 3, leading.zero = FALSE, single.row = TRUE)
```

	Model 1	Model 2	Model 3
(Intercept)	1.034 (.810)	1.813 (.558)	.284 (.898)
mujer1	−.078 (.124)	−.125 (.119)	−.114 (.130)
edad25_34	−.304 (.377)	−.434 (.367)	−.284 (.439)
edad35_44	.357 (.386)	.123 (.374)	.347 (.448)
edad45_54	−.122 (.377)	−.315 (.366)	−.268 (.437)
edad55_64	.033 (.380)	−.145 (.368)	−.003 (.441)
‘edad65+‘	.084 (.376)	.032 (.365)	.070 (.435)
urbano_rural2	.353 (.138)	.200 (.136)	.392 (.145)
‘indigena_esNo indigena‘	−.031 (.161)		
a62	−.256 (.545)		
a63	−.438 (.509)		
a64	−.019 (.470)		
a65	.053 (.469)		
a42	−.048 (.220)		
a43	−.501 (.219)		
a44	−.391 (.228)		
a45	−.692 (.255)		
d6_12	−.326 (.301)		
d6_13	−.872 (.289)		
d6_14	−.839 (.278)		
d6_15	−.668 (.293)		
c22	.784 (.510)		
c23	1.169 (.500)		
c24	1.154 (.502)		
c25	1.165 (.521)		
c52	−.171 (.545)		
c53	−.572 (.506)		
c54	−.631 (.509)		
c55	−1.015 (.529)		
d1_12		−.733 (.493)	
d1_13		−.266 (.440)	
d1_14		−.490 (.419)	
d1_15		−.340 (.420)	
d1_22		−.009 (.250)	
d1_23		−.536 (.231)	
d1_24		−.285 (.221)	
d1_25		−.508 (.244)	
cerca_conflictolejos_conflicto		−.567 (.156)	
c7_22			.033 (.271)
c7_23			−.162 (.181)
c7_24			−.277 (.153)
c7_25			
c7_32			.445 (.665)
c7_33			.256 (.622)
c7_34			−.214 (.622)
c7_35			−.230 (.635)
c132			.173 (.225)
c133			.119 (.200)
c134			.164 (.213)
c135			
c142			.364 (.468)
c143			.686 (.447)
c144			.149 (.444)
c145			.054 (.463)
Deviance (Null)	1919.620	1971.684	1688.025
df.null	1434	1474	1265
Log Likelihood	−904.398	−958.914	−809.454
AIC	1866.796	1951.828	1662.907
BIC	2019.595	2041.867	1776.067

Modelos con interacciones (indígenas y no indígena * variables indep. de interés) y efectos marginales

```
# Función para ajustar modelos con interacciones usando glm
fit_interaction_model <- function(data, formula) {
  glm(formula, data = data, family = binomial(link = "logit"))
}

# Ajustar modelos con interacciones
demographic_vars <- c("mujer", "edad", "urbano_rural")

formula_identity <- as.formula(paste("d14 ~",
                                     paste(demographic_vars, collapse = " + "), "+",
                                     "indigena_es * (a6 + a4 + d6_1 + c2 + c5)"))

formula_conflict <- as.formula(paste("d14 ~",
                                     paste(demographic_vars, collapse = " + "), "+",
                                     "indigena_es * (d1_1 + d1_2 + cerca_conflicto)"))

formula_contact <- as.formula(paste("d14 ~",
                                     paste(demographic_vars, collapse = " + "), "+",
                                     "indigena_es * (c7_2 + c7_3 + c13 + c14)"))

model_identity <- fit_interaction_model(elri_prep, formula_identity)
model_conflict <- fit_interaction_model(elri_prep, formula_conflict)
model_contact <- fit_interaction_model(elri_prep, formula_contact)
```

```
# Función para calcular efectos marginales manualmente
calculate_manual_margins <- function(model, var_name) {
  # Extraer datos del modelo
  data <- model$model

  # Crear un dataframe con todas las combinaciones de niveles
  levels_indigena <- levels(data$indigena_es)
  levels_var <- levels(data[[var_name]])
  grid <- expand.grid(indigena_es = levels_indigena, var = levels_var)
  names(grid)[2] <- var_name

  # Añadir valores promedio para otras variables
  for (col in names(data)) {
    if (!(col %in% c("indigena_es", var_name, "d14"))) {
      if (is.factor(data[[col]])) {
        grid[[col]] <- levels(data[[col]])[1] # Usar el primer nivel para factores
      } else {
        grid[[col]] <- mean(data[[col]], na.rm = TRUE)
      }
    }
  }
}

# Predecir probabilidades
grid$prob <- predict(model, newdata = grid, type = "response")

# Calcular efectos marginales
grid %>%
```



```

    group_by(indigena_es) %>%
    mutate(effect = prob - lag(prob)) %>%
    filter(!is.na(effect)) %>%
    mutate(variable = var_name,
           level = !!sym(var_name)) %>%
    select(indigena_es, variable, level, effect)
}

# Calcular efectos marginales para todas las variables de interés
calculate_all_margins <- function(model) {
  vars_of_interest <- names(model$model)[!(names(model$model) %in% c("d14", "mujer", "edad", "urbano_ru
  margins_list <- lapply(vars_of_interest, function(var) calculate_manual_margins(model, var))
  do.call(rbind, margins_list)
}

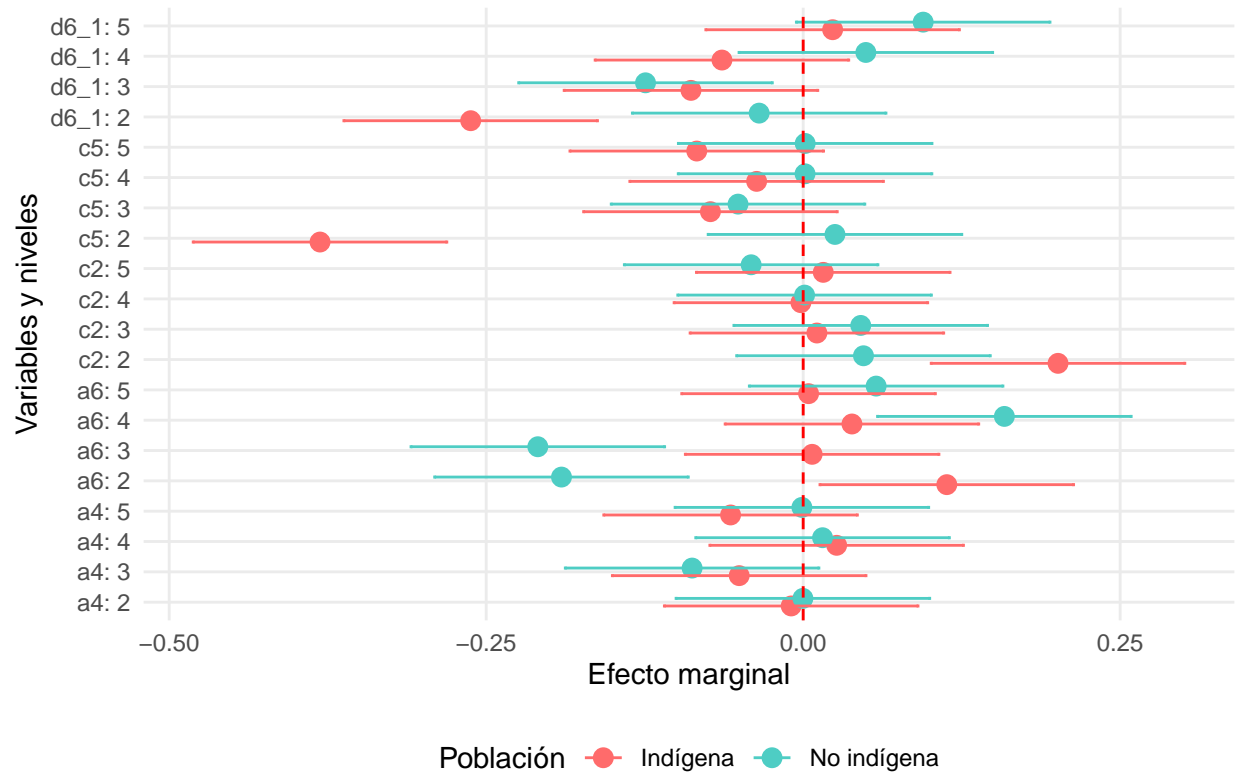
# Función para crear el gráfico
plot_margins <- function(margins, title) {
  margins$y_axis <- paste(margins$variable, margins$level, sep = ": ")

  ggplot(margins, aes(x = effect, y = y_axis, color = indigena_es)) +
    geom_point(position = position_dodge(width = 0.5), size = 3) +
    geom_errorbar(aes(xmin = effect - 0.1, xmax = effect + 0.1), # Ajusta estos valores según sea neces
                  position = position_dodge(width = 0.5), width = 0.2) +
    scale_color_manual(values = c("Indígena" = "#FF6B6B", "No indígena" = "#4ECDC4"),
                      name = "Población") +
    labs(title = title,
         x = "Efecto marginal",
         y = "Variables y niveles") +
    theme_minimal() +
    theme(legend.position = "bottom",
          panel.grid.minor = element_blank(),
          axis.text.y = element_text(hjust = 1),
          plot.title = element_text(hjust = 0.5, face = "bold")) +
    geom_vline(xintercept = 0, linetype = "dashed", color = "red")
}

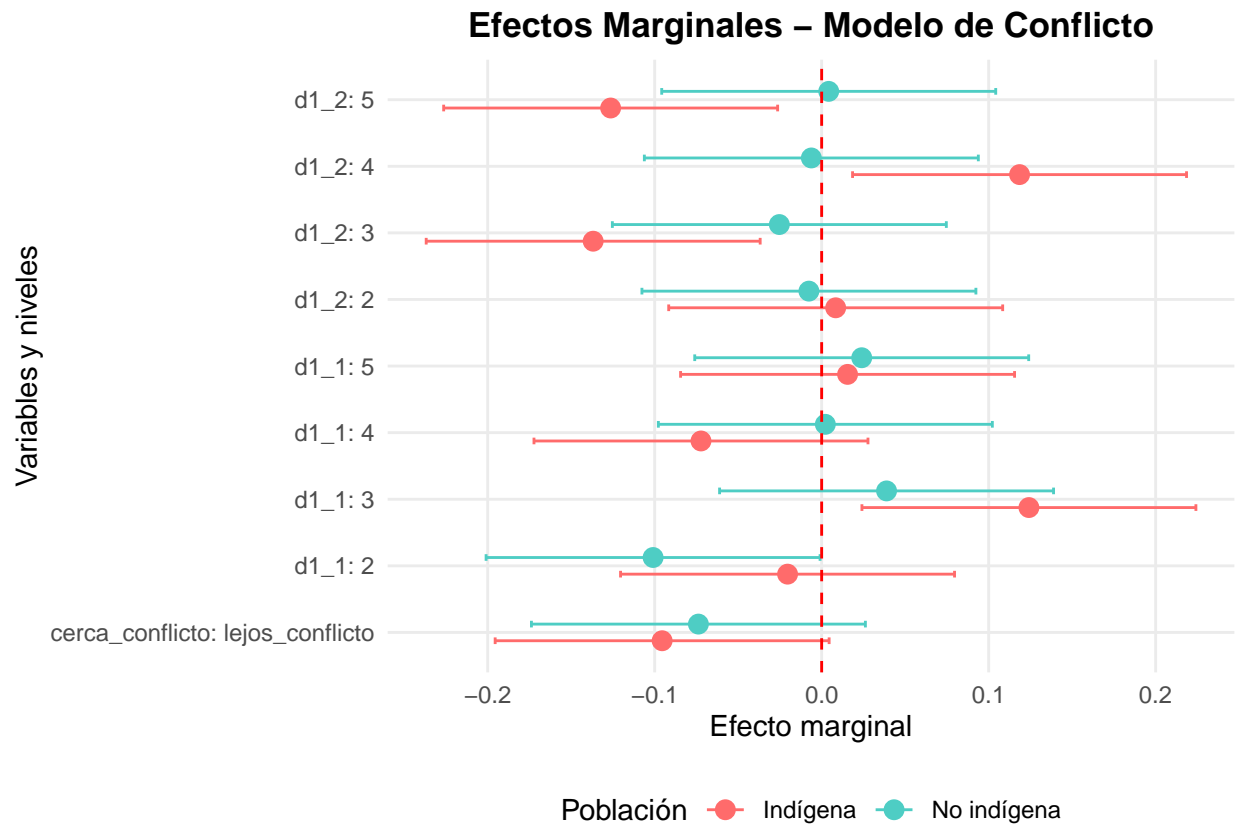
# Aplicar a cada modelo
margins_identity <- calculate_all_margins(model_identity)
plot_identity <- plot_margins(margins_identity, "Efectos Marginales - Modelo de Identidad y Confianza")
print(plot_identity)

```

Efectos Marginales – Modelo de Identidad y Confianza



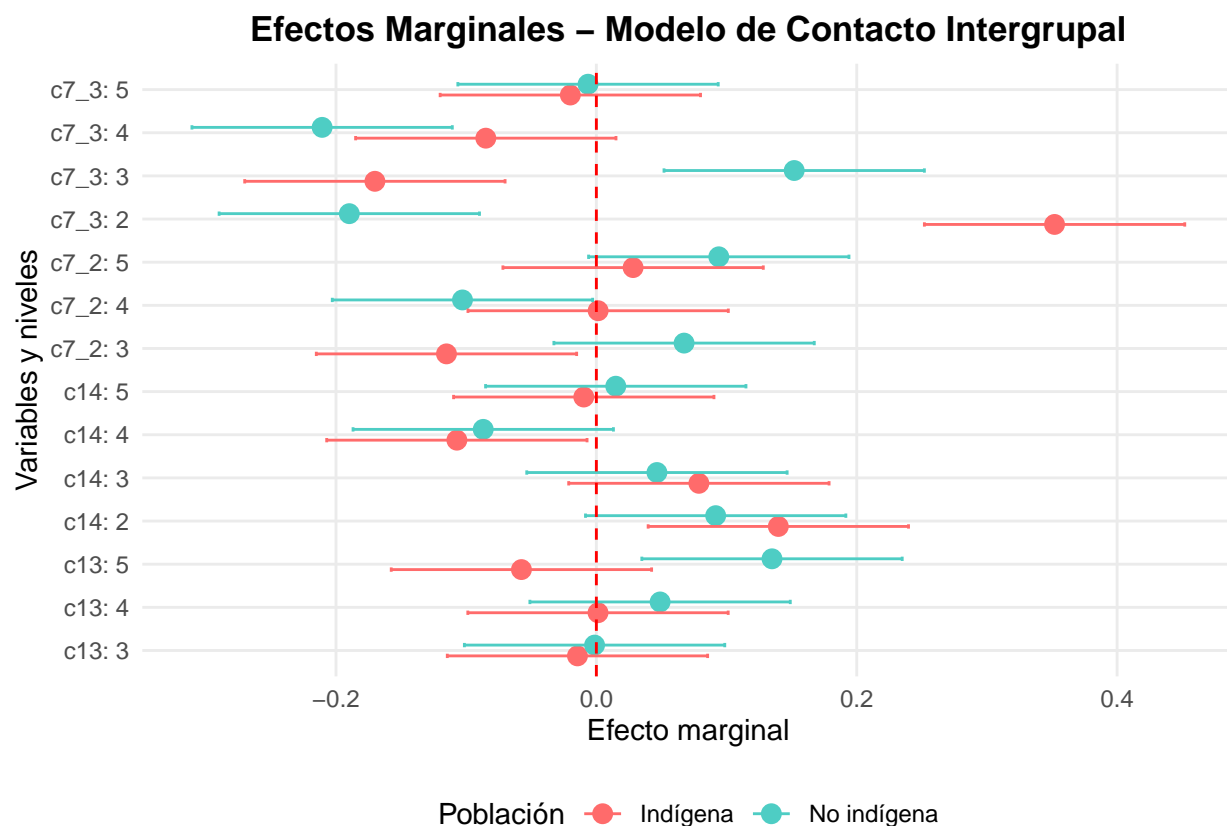
```
margins_conflict <- calculate_all_margins(model_conflict)
plot_conflict <- plot_margins(margins_conflict, "Efectos Marginales – Modelo de Conflicto")
print(plot_conflict)
```



```

margins_contact <- calculate_all_margins(model_contact)
plot_contact <- plot_margins(margins_contact, "Efectos Marginales - Modelo de Contacto Intergrupar")
print(plot_contact)

```



```
# Función para extraer y formatear resultados
extract_results <- function(model) {
  tidy(model) %>%
    mutate(odds_ratio = exp(estimate),
           lower_ci = exp(estimate - 1.96 * std.error),
           upper_ci = exp(estimate + 1.96 * std.error)) %>%
    select(term, estimate, odds_ratio, lower_ci, upper_ci, p.value)
}

# Extraer resultados de los modelos
results_identity <- extract_results(model_identity)
results_conflict <- extract_results(model_conflict)
results_contact <- extract_results(model_contact)

# Mostrar resultados
print("Modelo de Identidad y Confianza:")

## [1] "Modelo de Identidad y Confianza:"
results_identity

## # A tibble: 49 x 6
##   term                estimate odds_ratio lower_ci upper_ci p.value
##   <chr>                <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
## 1 (Intercept)          1.09         2.96   0.0714  123.    0.568
## 2 mujer1             -0.0883         0.915   0.715    1.17   0.483
## 3 edad25_34          -0.286         0.751   0.355    1.59   0.453
```

```
## 4 edad35_44          0.408      1.50    0.699      3.23  0.296
## 5 edad45_54         -0.0833    0.920    0.436      1.94  0.827
## 6 edad55_64          0.0584    1.06    0.499      2.25  0.879
## 7 edad65+            0.125     1.13    0.537      2.39  0.743
## 8 urbano_rural2      0.338     1.40    1.07      1.85  0.0155
## 9 indigena_esNo indígena 0.675     1.96    0.0314    123.  0.749
## 10 a62               0.737     2.09    0.439      9.95  0.355
## # i 39 more rows
```

```
print("Modelo de Conflicto:")
```

```
## [1] "Modelo de Conflicto:"
```

```
results_conflict
```

```
## # A tibble: 27 x 6
##   term                estimate odds_ratio lower_ci upper_ci p.value
##   <chr>              <dbl>      <dbl>   <dbl>   <dbl>   <dbl>
## 1 (Intercept)         0.690        1.99    0.508    7.82  0.322
## 2 mujer1             -0.0881       0.916    0.722    1.16  0.468
## 3 edad25_34          -0.410       0.664    0.321    1.37  0.268
## 4 edad35_44           0.195       1.22    0.580    2.55  0.605
## 5 edad45_54          -0.265       0.767    0.372    1.58  0.472
## 6 edad55_64          -0.101       0.904    0.436    1.87  0.785
## 7 edad65+            0.0684       1.07    0.521    2.20  0.852
## 8 urbano_rural2      0.222       1.25    0.952    1.64  0.109
## 9 indigena_esNo indígena 2.36        10.6     1.72    65.4  0.0111
## 10 d1_12             -0.0905      0.913    0.225    3.70  0.899
## # i 17 more rows
```

```
print("Modelo de Contacto Intergrupar:")
```

```
## [1] "Modelo de Contacto Intergrupar:"
```

```
results_contact
```

```
## # A tibble: 37 x 6
##   term                estimate odds_ratio lower_ci upper_ci p.value
##   <chr>              <dbl>      <dbl>   <dbl>   <dbl>   <dbl>
## 1 (Intercept)        -0.317       0.728    0.0686    7.73  0.793
## 2 mujer1             -0.0935       0.911    0.702    1.18  0.481
## 3 edad25_34          -0.221       0.802    0.332    1.94  0.624
## 4 edad35_44           0.436       1.55    0.630    3.80  0.341
## 5 edad45_54          -0.183       0.833    0.346    2.01  0.683
## 6 edad55_64          0.0691       1.07    0.442    2.60  0.879
## 7 edad65+            0.124       1.13    0.472    2.71  0.781
## 8 urbano_rural2      0.413       1.51    1.13    2.02  0.00513
## 9 indigena_esNo indígena 1.20        3.34    0.136    81.6  0.460
## 10 c7_23             -0.501       0.606    0.293    1.25  0.176
## # i 27 more rows
```

Tabla II (interacciones)

```
#texreg(list(model_identity, model_conflict, model_contact),
#         float.pos = "H", return.string = TRUE, bold = 0.05, stars = 0,
#         custom.note = "Coefficients with $p < 0.05$ in \\textbf{bold}.",
#         digits = 3, leading.zero = FALSE, single.row = TRUE)
```

