

Estimación de poder estadístico con base en simulaciones para la determinación de tamaños de muestra

Proyecto *Funciones evolutivas del habla dirigida a bebés: Impacto en la atención, las preferencias auditivas y el desarrollo lingüístico y musical temprano*

Juan David Leongómez ^{1,*}

08 de julio de 2025

¹ CODEC: Ciencias Cognitivas y del Comportamiento, Facultad de Psicología, Universidad El Bosque.

* Correspondencia: jleongomez@unbosque.edu.co

Description

Enfoque de simulación de datos y análisis de poder:

En este apéndice técnico presentamos análisis de poder estadístico, tanto por simulación como analíticos, realizados con el fin de justificar los tamaños muestrales propuestos para los dos estudios del proyecto *Funciones evolutivas del habla dirigida a bebés: Impacto en la atención, las preferencias auditivas y el desarrollo lingüístico y musical temprano*.

En el **Estudio 1** utilizamos un diseño intra-sujeto $2 \times 2 \times 2$, en el que modelamos el tiempo total de fijación como función de tres predictores acústicos binarios: variabilidad tonal ($f_0 SD$), tono medio ($f_0 mean$) y dispersión de formantes (D_f). Realizamos una simulación Monte Carlo utilizando tamaños del efecto y niveles de ruido realistas, en línea con las condiciones experimentales esperadas. Analizamos cada conjunto de datos simulado mediante modelos lineales mixtos con interceptos aleatorios por participante. Los resultados muestran que, con aproximadamente 150–160 participantes, se alcanza un poder estadístico superior al 80 % para detectar todos los efectos principales. En contraste, los efectos de interacción son considerablemente más pequeños y requerirían muestras mucho mayores y probablemente no realistas para ser detectados de forma confiable. No realizamos una prueba de equivalencia para este estudio, dado que los efectos se analizarán mediante modelos mixtos, para los cuales no existe actualmente un procedimiento estandarizado para llevar a cabo ni planear pruebas de equivalencia.

En el **Estudio 2** examinamos la relación entre variables acústicas del habla dirigida a bebés (IDS) en madres y las habilidades lingüísticas y musicales de sus hijos, utilizando correlación de Pearson. Calculamos el poder estadístico de forma analítica, tanto para la detección tradicional como para pruebas de equivalencia. En particular, estimamos el tamaño muestral necesario para (a) detectar una correlación de al menos $\rho = 0.30$, y (b) concluir equivalencia estadística con cero dentro de un margen predefinido de ± 0.15 . Los resultados indican que una muestra de aproximadamente 480 participantes proporciona un poder estadístico de al menos el 90 % para ambos objetivos. Adicionalmente, realizamos una simulación Monte Carlo para estimar la probabilidad de confirmar la hipótesis nula o la asociación esperada, o de obtener resultados inconclusos (es decir, correlaciones estadísticamente significativas pero equivalentes a cero).

Reproducibilidad:

Este documento incluye todo el código y explicaciones paso a paso de los análisis, figuras y tablas (incluyendo figuras y tablas suplementarias) correspondientes al análisis de poder basado en simulaciones para el proyecto *Funciones evolutivas del habla dirigida a bebés: Impacto en la atención, las preferencias auditivas y el desarrollo lingüístico y musical temprano*.

Los datos están disponibles en el Open Science Framework (OSF): <https://doi.org/10.XXXXXXX>. La estrategia de análisis de poder y los análisis/modelos principales fueron diseñados por Juan David Leongómez. Este documento y todo su código subyacente fueron elaborados en R Markdown por Juan David Leongómez utilizando R y L^AT_EX.

Índice

1. Preliminares	2
1.1. Cargar paquetes	2
2. Estudio 1	3
2.1. Estrategia de simulación	3
2.2. Simular los datos	3
2.2.1. Ajustar el modelo lineal mixto y estimar tamaños del efecto	4
2.2.2. Visualización de efectos: figura tipo raincloud	6
2.3. Estimación de poder mediante simulación de Monte Carlo	8
2.3.1. Resumen del poder estadístico	9
2.3.2. Curvas de poder	9
2.3.3. Estimación del tamaño muestral	12
2.3.4. Distribución del poder con el tamaño muestral final	12
3. Estudio 2	14
3.1. Estrategia de análisis	14
3.2. Curvas de poder para detección y equivalencia	14
4. Información de la sesión (para reproducibilidad)	19
5. Referencias	20

1. Preliminares

1.1. Cargar paquetes

Este archivo fue creado utilizando `knitr` (Xie, 2014, 2025), y los análisis se realizaron con paquetes del ecosistema `tidyverse` para la manipulación de datos y la generación de figuras (Wickham et al., 2019), incluyendo `dplyr` (Wickham et al., 2023), `stringr` (Wickham, 2023) y `ggplot2` (Wickham, 2016). Las simulaciones de poder se ejecutaron con `purrr` y `furrr` (Vaughan & Dancho, 2022) para programación funcional y en paralelo, y los modelos fueron ajustados con `lmerTest` (Kuznetsova et al., 2017), una extensión de `lme4` que proporciona valores *p* para modelos mixtos. Las estimaciones de tamaño del efecto se calcularon con el paquete `effectsize` (Ben-Shachar et al., 2020), y las visualizaciones incluyeron figuras mejoradas usando `ggdist` (Kay, 2024, 2025) para figuras tipo `raincloud`, y `tidyquant` (Dancho & Vaughan, 2025) para paletas de colores personalizadas. Todos los paquetes pueden instalarse desde el Comprehensive R Archive Network (CRAN). Para una lista completa de las versiones de los paquetes utilizados en este análisis, consulte la sección *Session Info* al final del documento.

```
# Cargar paquetes necesarios
library(tidyverse) # Manipulación de datos y figuras
library(ggdist) # Para figuras tipo raincloud
library(tidyquant) # Temas personalizados y escalas de color
library(effectsize) # Estimación del tamaño del efecto
library(lmerTest) # Modelos lineales mixtos con valores p
library(furrr) # Procesamiento paralelo con purrr
library(scales) # Funciones de escalado para figuras
library(stringr) # Manejo de cadenas de texto
library(ggpibr) # Organización de figuras
```

```
library(truncnorm) # Distribución normal truncada
library(TOSTER) # Para análisis de poder y pruebas de equivalencia
library(kableExtra) # Para crear tablas
```

2. Estudio 1

2.1. Estrategia de simulación

Esta simulación modela el **tiempo total de fijación** (en milisegundos) como función de tres factores binarios intra-sujeto que se manipulan experimentalmente:

- $f_0 SD$: Variabilidad tonal (Baja vs Alta)
- $f_0 mean$: Tono medio (Baja vs Alta)
- D_f : Dispersión de formantes (Baja vs Alta)

Cada uno de los **10 000 participantes** observa estímulos que representan las **8 combinaciones posibles** de estos factores (para un total de 80 000 datos). El tiempo de fijación se extrae de una distribución normal, siguiendo la siguiente lógica:

- **Efectos principales**:
 - $f_0 SD$: Efecto más fuerte. Suma, en promedio, 100 ms cuando es Alta.
 - $f_0 mean$: Suma 75 ms cuando es Alta.
 - D_f : Suma 50 ms cuando es Alta.
- **Interacciones**:
 - $f_0 SD \times f_0 mean$: Suma 100 ms cuando ambos son Altas.
 - $f_0 SD \times D_f$: Suma 100 ms cuando ambos son Altas.
 - $f_0 mean \times D_f$: Suma 50 ms cuando ambos son Altas.
 - Interacción triple: Suma 120 ms cuando los tres son Altas.

Se utiliza una distribución normal con una desviación estándar de 1 000 ms para generar datos ruidosos a nivel de ensayo. Los valores se restringen al rango entre 0 y 5000 ms.

2.2. Simular los datos

```
set.seed(123)

# Crear las 8 combinaciones de condiciones
stimulus_conditions <- expand_grid(
  f0_sd      = c("Baja", "Alta"),
  f0_mean    = c("Baja", "Alta"),
  Df         = c("Baja", "Alta")
)

# Generar 10000 identificadores de participantes
participant_ids <- str_c("P", str_pad(1:10000, width = 4, pad = "0"))

# Diseño completamente cruzado: cada participante ve todas las condiciones
design <- expand_grid(
  ID = participant_ids,
  stimulus_conditions
)

# Simular tiempos de fijación con base en los efectos principales e interacciones
```

```

simulated_data <- design |>
  mutate(
    f0_sd_val = if_else(f0_sd == "Baja", 0, 1),
    f0_mean_val = if_else(f0_mean == "Baja", 0, 1),
    Df_val = if_else(Df == "Baja", 0, 1),
    base_mean = 2500 + 100 * f0_sd_val,
    effect_mean = 75 * f0_mean_val,
    effect_df = 50 * Df_val,
    interaction_effect = 100 * f0_sd_val * f0_mean_val +
      100 * f0_sd_val * Df_val +
      50 * f0_mean_val * Df_val +
      120 * f0_sd_val * f0_mean_val * Df_val,
    mu = base_mean + effect_mean + effect_df + interaction_effect,
    fixation_time = round(rnorm(n(), mean = mu, sd = 1000)),
    fixation_time = pmin(pmax(fixation_time, 0), 5000)
  ) |>
  select(ID, f0_sd, f0_mean, Df, fixation_time)

```

2.2.1. Ajustar el modelo lineal mixto y estimar tamaños del efecto

Ajustamos un modelo lineal mixto considerando un intercepto aleatorio por participante, y evaluamos todos los efectos principales y las interacciones entre factores acústicos. También estimamos el tamaño del efecto mediante omega parcial al cuadrado (ω_p^2) para cada término fijo del modelo.

La ecuación del modelo es:

$$\begin{aligned}
\text{Tiempo de fijación}_i &\sim N(\mu, \sigma^2) \\
\mu &= \alpha_j[i] + \\
&\quad \beta_1(f_0 \text{ SD, Baja}) + \\
&\quad \beta_2(f_0 \text{ mean, Baja}) + \\
&\quad \beta_3(D_f, \text{ Baja}) + \\
&\quad \beta_4(f_0 \text{ mean, Baja} \times f_0 \text{ SD, Baja}) + \\
&\quad \beta_5(D_f, \text{ Baja} \times f_0 \text{ SD, Baja}) + \\
&\quad \beta_6(D_f, \text{ Baja} \times f_0 \text{ mean, Baja}) + \\
&\quad \beta_7(D_f, \text{ Baja} \times f_0 \text{ mean, Baja} \times f_0 \text{ SD, Baja}) \\
\alpha_j &\sim N(\mu_{\alpha_j}, \sigma_{\alpha_j}^2), \quad \text{para ID } j = 1, \dots, J
\end{aligned}$$

```

# Ajustar el modelo mixto
mod <- lmer(fixation_time ~ f0_sd * f0_mean * Df + (1 | ID), data = simulated_data)

# Calcular omega parcial al cuadrado para cada término fijo
pop_effect <- omega_squared(mod, partial = TRUE) |>
  rename("Efecto" = "Parameter")

# Tabla tipo ANOVA de efectos fijos con interpretación de tamaños del efecto
anova(mod) |>
  rownames_to_column(var = "Efecto") |>
  left_join(interpret_omega_squared(pop_effect)) |>
  mutate(
    # Reemplazar nombres por notación matemática
    Efecto = str_replace_all(Efecto, "f0_sd", "$f_0$"),
    Efecto = str_replace_all(Efecto, "f0_mean", "$f_0$"),
    Efecto = str_replace_all(Efecto, "Df", "$D_f$"),
    Efecto = str_replace_all(Efecto, ":" , " $\times $"),
  )

```

```

# Traducir interpretaciones del tamaño del efecto
Interpretation = str_replace_all(Interpretation, "very small", "Muy pequeño"),
Interpretation = str_replace_all(Interpretation, "small", "Pequeño"),

# Redondear grados de libertad y valores F
DenDF = round(DenDF, 2),
`F value` = round(`F value`, 2),

# Dar formato al valor p según su significancia
`Pr(>F)` = ifelse(`Pr(>F)` < 0.0001,
  "\\textbf{< 0.0001}",
  ifelse(`Pr(>F)` < 0.001,
    "\\textbf{< 0.001}",
    ifelse(`Pr(>F)` < 0.05,
      paste0("\\textbf{", round(`Pr(>F)`, 4), "}"),
      round(`Pr(>F)`, 2)
    )
  )
),

# Redondear omega parcial al cuadrado
Omega2_partial = round(Omega2_partial, 3)
) |>
select(Efecto, "F value", "NumDF", "DenDF", "Pr(>F)", "Omega2_partial", "Interpretation") |>
unite("$g1$", NumDF:DenDF, sep = " ", ") |>
rename(
  "Efectos fijos" = "Efecto",
  "$F$" = "F value",
  "$p$" = `Pr(>F)` ,
  "$\\omega^2_p$" = "Omega2_partial",
  "Interpretación" = "Interpretation"
) |>
kable(
  booktabs = TRUE,
  align = "lcccc",
  linesep = "",
  caption = "Efectos principales e interacciones entre factores acústicos  
en la población simulada",
  escape = FALSE
) |>
kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
footnote(
  general = "El tamaño de efecto simulado está representado como omega parcial al cuadrado ($\\\\omega^2_p$). Para los efectos,  
$f_0$~SD$: variabilidad de la frecuencia fundamental;  
$f_0$~mean$: frecuencia fundamental media;  
$D_f$: dispersión de las formantes.  
Siguiendo los estándares  
recomendados por los creadores del paquete \\\texttt{effectsize}  
(\\\\\\cite{effectsize}; ver  
\\\\\\url{https://easystats.github.io/effectsize/reference/interpret_omega_squared.html}),  
en la columna \\\textit{Interpretación} un valor de $\\\\omega^2_p$ menor a 0.01  
se considera muy pequeño,  
entre 0.01 y 0.06 como pequeño,  
entre 0.06 y 0.14 como mediano,
```

```
y mayor o igual a 0.14 como grande (de \\\cite{field2013}).  
Efectos significativos están en negrilla.",  
threeparttable = TRUE,  
footnote_as_chunk = TRUE,  
escape = FALSE,  
general_title = "Nota: "  
)
```

Cuadro 1. Efectos principales e interacciones entre factores acústicos en la población simulada

Efectos fijos	F	gl	p	ω_p^2	Interpretación
$f_0 SD$	1062.15	1, 69992.95	<0.0001	0.015	Pequeño
$f_0 mean$	646.53	1, 69992.95	<0.0001	0.009	Muy pequeño
D_f	492.11	1, 69992.95	<0.0001	0.007	Muy pequeño
$f_0 SD \times f_0 mean$	138.96	1, 69992.96	<0.0001	0.002	Muy pequeño
$f_0 SD \times D_f$	151.63	1, 69992.96	<0.0001	0.002	Muy pequeño
$f_0 mean \times D_f$	59.79	1, 69992.96	<0.0001	0.001	Muy pequeño
$f_0 SD \times f_0 mean \times D_f$	10.56	1, 69992.93	0.0012	0.000	Muy pequeño

Nota: El tamaño de efecto simulado está representado como omega parcial al cuadrado (ω_p^2). Para los efectos, $f_0 SD$: variabilidad de la frecuencia fundamental; $f_0 mean$: frecuencia fundamental media; D_f : dispersión de las formantes. Siguiendo los estándares recomendados por los creadores del paquete **effectsize** (Ben-Shachar et al., 2020; ver https://easystats.github.io/effectsize/reference/interpret_omega_squared.html), en la columna Interpretación un valor de ω_p^2 menor a 0.01 se considera muy pequeño, entre 0.01 y 0.06 como pequeño, entre 0.06 y 0.14 como mediano, y mayor o igual a 0.14 como grande (de Field, 2013). Efectos significativos están en negrilla.

2.2.2. Visualización de efectos: figura tipo raincloud

A continuación se muestra una figura tipo *raincloud* que representa la distribución de los tiempos de fijación para cada condición. Ilustra las diferencias según $f_0 SD$, dividido por columnas según $f_0 mean$ y por filas según D_f .

```
# Preparar los datos con niveles de factores ordenados
plot_data <- simulated_data |>
  mutate(
    f0_mean = factor(f0_mean, levels = c("Baja", "Alta")),
    Df = factor(Df, levels = c("Baja", "Alta")),
    f0_sd = factor(f0_sd, levels = c("Baja", "Alta"))
  )

# Etiquetas para los paneles
label_f0_mean <- c(
  "Baja" = "'Baja '*italic(f)[0]*' mean'",
  "Alta" = "'Alta '*italic(f)[0]*' mean'"
)
label_Df <- c(
  "Baja" = "'Baja '*italic(D)[f]",
  "Alta" = "'Alta '*italic(D)[f]"
)

# Gráfico tipo raincloud
ggplot(plot_data, aes(x = f0_sd, y = fixation_time, fill = f0_sd)) +
  # Nube de distribución
  stat_halfeye(
    adjust = 0.5,
    justification = -0.3,
```

```
.width = 0,
point_colour = NA,
alpha = 0.6
) +
# Puntos individuales
geom_jitter(
  aes(color = f0_sd),
  width = 0.1,
  alpha = 0.05,
  size = 0.7
) +
# Punto del promedio
stat_summary(
  fun = mean,
  geom = "point",
  size = 2,
  color = "black",
  position = position_nudge(x = 0.2)
) +
# Barra de error
stat_summary(
  fun.data = mean_se,
  geom = "errorbar",
  width = 0.1,
  color = "black",
  position = position_nudge(x = 0.2)
) +
# Paneles por f0_mean × Df
facet_wrap(
  Df ~ f0_mean,
  nrow = 1,
  labeller = labeller(
    f0_mean = as_labeller(label_f0_mean, default = label_parsed),
    Df = as_labeller(label_Df, default = label_parsed)
  )
) +
labs(
  title = "Distribución de tiempos de fijación en la población simulada",
  x = expression(italic(f)[0] * " SD"),
  y = "Tiempo de fijación (ms)",
  fill = expression(italic(f)[0] * " SD"),
  color = expression(italic(f)[0] * " SD")
) +
scale_colour_tq() +
scale_fill_tq() +
theme_tq(base_size = 14)
```

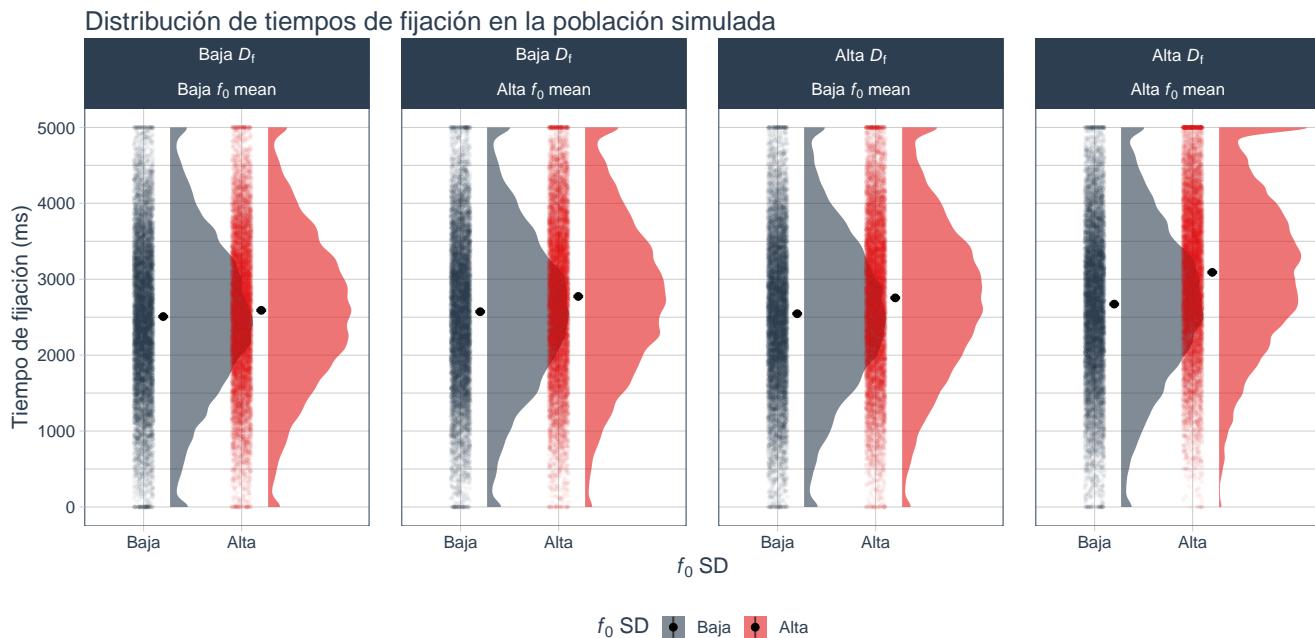


Figura 1. Distribución de los tiempos de fijación en función de $f_0 SD$ (eje x), dividida por $f_0 mean$ (columnas) y D_f (filas). Cada panel muestra la distribución de densidad, los puntos individuales (dispersados), y los promedios por condición con sus barras de error. Los tiempos de fijación aumentan con mayor $f_0 SD$, y los efectos se ven modulados por las demás características acústicas.

2.3. Estimación de poder mediante simulación de Monte Carlo

Estimamos el poder estadístico para detectar cada efecto simulando 1 000 muestras aleatorias de la población simulada en distintos tamaños muestrales. Específicamente, a partir de la población simulada ($N = 10\,000$), extraemos 1 000 muestras aleatorias de tamaño $n = 10$, luego otras 1 000 de tamaño $n = 20$, y así sucesivamente hasta obtener 1 000 muestras de tamaño $n = 200$. Para cada una de estas 20 000 muestras, ajustamos el mismo modelo lineal mixto, examinamos la distribución de los valores p , y estimamos la probabilidad de detectar un efecto estadísticamente significativo para cada término del modelo.

```
# Definir el nivel alfa
alpha_lev <- 0.05

# Función para ejecutar múltiples simulaciones de poder para cada n
run_power_sim <- function(dat, n_sample, n_sim, alpha = alpha_lev) {
  # Ejecutar `n_sim` simulaciones para cada tamaño muestral
  map_dfr(seq_len(n_sim), \i) {
    # Seleccionar participantes
    ids <- dat |>
      distinct(ID) |>
      slice_sample(n = n_sample) |>
      pull(ID)
    sampled_data <- dat |> filter(ID %in% ids)

    # Ajustar modelo
    mod <- lmer(fixation_time ~ f0_sd * f0_mean * Df + (1 | ID), data = sampled_data)

    # Extraer resultados ANOVA
    anova_res <- anova(mod)

    # Obtener todos los términos fijos
  }
}
```

```

term_names <- rownames(anova_res)

# Iterar sobre los términos
map_dfr(term_names, \((term) {
  tibble(
    sim = i,
    term = term,
    n_sample = n_sample,
    p_value = anova_res[term, "Pr(>F)"],
    signif = ifelse(p_value < alpha, "Significant", "Non-significant")
  )
})})
}) |>
  # Limpiar y agregar resultados
  filter(!is.na(p_value))
}

# Definir tamaños muestrales y ejecutar las simulaciones
sample_sizes <- seq(10, 200, by = 10)
term_order <- c(
  "f0_sd",
  "f0_mean",
  "Df",
  "f0_sd:f0_mean",
  "f0_sd:Df",
  "f0_mean:Df",
  "f0_sd:f0_mean:Df"
)

plan(multisession) # Paralelizar si está disponible

power_results <- map_dfr(
  sample_sizes,
  ~ run_power_sim(simulated_data, n_sample = .x, n_sim = 1000),
  .id = "sample_step"
) |>
  mutate(term = factor(term, levels = term_order))

```

2.3.1. Resumen del poder estadístico

Se agregan los resultados de las simulaciones para estimar el poder estadístico medio asociado a cada término del modelo, a través de distintos tamaños muestrales. El poder se define como la proporción de simulaciones en las que se obtuvo un valor p inferior al nivel de significancia α .

```

power_summ <- power_results |>
  group_by(term, n_sample) |>
  summarise(
    power = mean(p_value < alpha_lev, na.rm = TRUE),
    .groups = "drop"
  )

```

2.3.2. Curvas de poder

Esta figura muestra las curvas de poder para detectar los efectos principales y las interacciones, resaltando los tamaños muestrales en los que el poder estadístico supera el 80%.

```

# Etiquetas para los términos en notación matemática
term_labels <- c(
  "f0_sd" = italic(f)[0]*' SD',
  "f0_mean" = italic(f)[0]*' mean',
  "Df" = italic(D)[f],
  "f0_sd:f0_mean" = italic(f)[0]*' SD × *italic(f)[0]*' mean',
  "f0_sd:Df" = italic(f)[0]*' SD × *italic(D)[f]',
  "f0_mean:Df" = italic(f)[0]*' mean × *italic(D)[f]',
  "f0_sd:f0_mean:Df" = italic(f)[0]*' SD × *italic(f)[0]*' mean × *italic(D)[f]"
)

# Dividir entre efectos principales e interacciones
main_power_summ <- power_summ |>
  filter(!str_detect(term, ":"))

int_power_summ <- power_summ |>
  filter(str_detect(term, ":"))

# Encontrar el mínimo n donde el poder supera 0.8
sample_power_08 <- main_power_summ |>
  filter(power > 0.8) |>
  group_by(term) |>
  filter(power == min(power, na.rm = TRUE))

# Usar el mayor de esos tamaños muestrales como recomendación final,
# para asegurar poder adecuado en todos los efectos principales
final_sample_size_st1 <- max(sample_power_08$n_sample)

# Poder en el tamaño muestral final
sample_fin <- main_power_summ |>
  filter(n_sample == final_sample_size_st1) |>
  arrange(desc(power))

# Figura: curvas de poder para efectos principales e interacciones
ggarrange(
  ggplot(main_power_summ, aes(x = n_sample, y = power)) +
    geom_line(linewidth = 0.5) +
    geom_point(aes(color = power > 0.8), size = 2) +
    geom_hline(yintercept = 0.8, linetype = "dashed", color = "red") +
    scale_colour_tq(
      labels = c("FALSE" = "<= 80%", "TRUE" = "> 80%"),
      name = "Umbral de poder"
    ) +
    scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
    labs(
      title = "Curvas de poder para efectos principales",
      x = "Número de participantes",
      y = "Poder estimado"
    ) +
    theme_tq() +
    facet_wrap(~term, labeller = as_labeller(term_labels, label_parsed)),
  ggplot(int_power_summ, aes(x = n_sample, y = power)) +
    geom_line(linewidth = 0.5) +
    geom_point(aes(color = power > 0.8), size = 2) +
    geom_hline(yintercept = 0.8, linetype = "dashed", color = "red") +
    scale_colour_tq(
      labels = c("FALSE" = "<= 80%", "TRUE" = "> 80%"),
      name = "Umbral de poder"
    )
)

```

```

    name = "Umbral de poder"
) +
scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
labs(
  title = "Curvas de poder para interacciones",
  x = "Número de participantes",
  y = "Poder estimado"
) +
theme_tq() +
facet_wrap(~term,
  labeller = as_labeller(term_labels, label_parsed),
  nrow = 1
),
labels = "AUTO",
common.legend = TRUE,
legend = "bottom",
nrow = 2
)

```

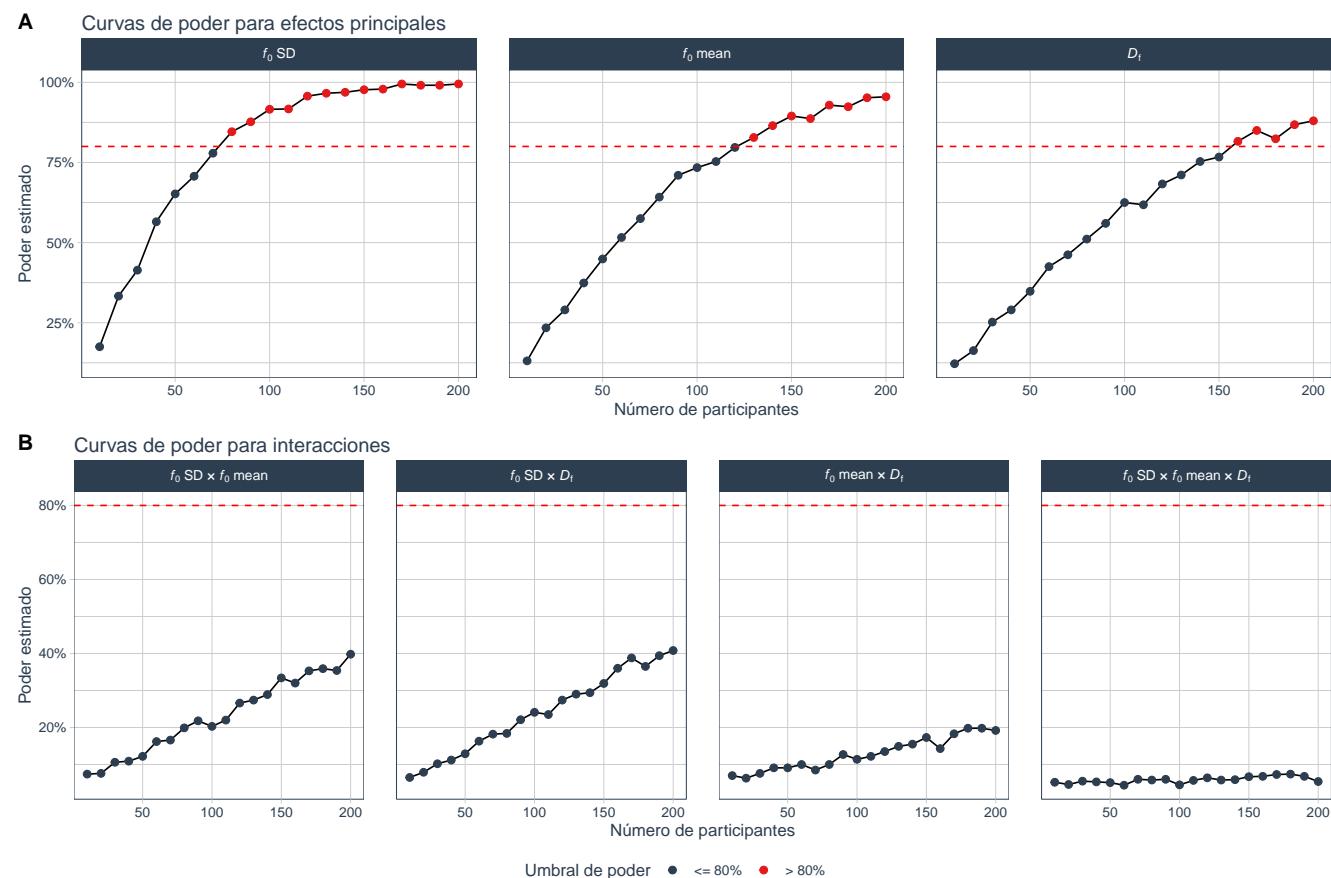


Figura 2. Curvas de poder para detectar efectos principales e interacciones. Los paneles muestran el poder estadístico como función del tamaño muestral para cada efecto fijo (A) e interacción (B). Los puntos indican las estimaciones de poder basadas en simulaciones para cada tamaño muestral, con líneas punteadas rojas que marcan el umbral del 80 %. La mayoría de los efectos principales alcanzan un alto poder con menos de 160 participantes, mientras que las interacciones permanecen con bajo poder en este rango, reflejando sus tamaños de efecto más pequeños.

2.3.3. Estimación del tamaño muestral

En la población simulada, los efectos de interacción son extremadamente débiles y requerirían tamaños muestrales muy grandes para ser detectados de forma fiable. En cambio, los efectos principales de $f_0 SD$, $f_0 mean$ y D_f pueden detectarse con un tamaño muestral mucho más razonable. Si los efectos poblacionales verdaderos son como los simulados ($\omega_p^2 = 0.015, 0.009, 0.007$, respectivamente), se alcanzaría un poder estadístico de 0.8 con $n = 80$ para $f_0 SD$, $n = 130$ para $f_0 mean$ y $n = 160$ para D_f .

Según las simulaciones:

- El poder supera el 80 % para los efectos principales con $n = 160$.
- Los valores correspondientes de omega parcial al cuadrado fueron: 0.015, 0.009, 0.007.

Por tanto, un tamaño muestral de $n = 160$ participantes sería suficiente para detectar de forma fiable todos los efectos principales. Los efectos de interacción requieren muestras mucho más grandes, debido a que sus tamaños de efecto reales son más pequeños.

2.3.4. Distribución del poder con el tamaño muestral final

A continuación, visualizamos la distribución de los valores p para todos los efectos utilizando el tamaño muestral final, lo cual permite ilustrar el perfil esperado de errores Tipo I y II.

```
# Filtrar por n final
final_power <- power_summ |>
  filter(n_sample == final_sample_size_st1)

# Obtener todos los valores p en ese n
main_power_results <- power_results |>
  filter(!str_detect(term, ":")) |>
  group_by(term) |>
  filter(n_sample == final_sample_size_st1) |>
  left_join(final_power, by = c("term", "n_sample"))

int_power_results <- power_results |>
  filter(str_detect(term, ":")) |>
  group_by(term) |>
  filter(n_sample == final_sample_size_st1) |>
  left_join(final_power, by = c("term", "n_sample"))

# Figura: distribuciones de los valores p
ggarrange(
  ggplot(main_power_results, aes(x = p_value, fill = signif)) +
    geom_histogram(
      bins = 100,
      breaks = seq(0, 1, 0.01),
      alpha = 0.8,
      colour = NA
    ) +
    geom_text(
      data = main_power_results |> distinct(term, power),
      aes(
        x = Inf, y = Inf,
        label = paste0("Poder = ", round(power, 2))
      ),
      vjust = 1.5, hjust = 1.1,
      inherit.aes = FALSE
    ) +
    labs(
      x = "Valor p",
      y = "Frecuencia",
    )
)
```

```

fill = "Significancia",
title = "Distribución de valores p para efectos principales"
) +
scale_fill_tq(name = "Significancia") +
scale_x_continuous(breaks = scales::pretty_breaks(n = 10)) +
facet_wrap(~term, labeller = as_labeller(term_labels, label_parsed)) +
theme_tq() +
guides(
  fill = guide_legend(override.aes = list(colour = NA), reverse = TRUE),
  colour = "none"
),
ggplot(int_power_results, aes(x = p_value, fill = signif)) +
  geom_histogram(
    bins = 100,
    breaks = seq(0, 1, 0.01),
    alpha = 0.8,
    colour = NA
  ) +
  geom_text(
    data = int_power_results |> distinct(term, power),
    aes(
      x = Inf, y = Inf,
      label = paste0("Poder = ", round(power, 2))
    ),
    vjust = 1.5, hjust = 1.1,
    inherit.aes = FALSE
  ) +
  labs(
    x = "Valor p",
    y = "Frecuencia",
    fill = "Significancia",
    title = "Distribución de valores p para interacciones"
  ) +
  scale_fill_tq(name = "Significancia") +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 10)) +
  facet_wrap(~term,
    labeller = as_labeller(term_labels, label_parsed),
    nrow = 1
  ) +
  theme_tq() +
  guides(
    fill = guide_legend(override.aes = list(colour = NA), reverse = TRUE),
    colour = "none"
  ),
  labels = "AUTO",
  common.legend = TRUE,
  legend = "bottom",
  nrow = 2
)

```

A Distribución de valores p para efectos principales

Figura 3. Distribución de los valores p en las simulaciones con $n = 160$. Esta figura muestra la distribución de los valores p para cada efecto, basada en 1000 simulaciones utilizando el tamaño muestral final recomendado. Los efectos principales muestran una fuerte asimetría hacia valores p bajos, consistente con alto poder. Las interacciones producen distribuciones más uniformes, lo que indica una sensibilidad limitada para detectar estos efectos con este tamaño muestral.

3. Estudio 2

3.1. Estrategia de análisis

En el Estudio 2 investigamos el valor predictivo de las características acústicas del habla materna dirigida a bebés prelingüísticos sobre las habilidades lingüísticas y musicales de sus hijos o hijas a los 3 años de edad. Para estimar la probabilidad de detectar asociaciones bivariadas entre las características del habla dirigida a bebés (IDS) y el desarrollo infantil, utilizamos correlaciones de Pearson. Evaluamos dos aspectos estadísticos complementarios:

1. **Detección de una correlación** de al menos $\rho = 0.30$ (prueba direccional, unilateral).
2. **Prueba de equivalencia**, para determinar si una correlación observada es estadísticamente indistinguible de cero dentro de un margen predefinido de ± 0.25 .

Para esto usamos la función `TOSTER::power_z_cor()`, que permite calcular el poder estadístico de forma analítica en función del tamaño muestral (Caldwell, 2022; Lakens, 2017).

3.2. Curvas de poder para detección y equivalencia

```
# Parámetros clave para el análisis
eq_bounds <- 0.15 # Margen de equivalencia (±)
```

```

rho_pop <- 0.30 # Tamaño del efecto poblacional esperado (correlación)
alpha <- 0.05 # Nivel de significancia
n_sim <- 100000 # Número de simulaciones (usado en otra sección)

# Panel A: Curvas de poder según el tamaño de muestra
sample_sizes <- seq(10, 600, by = 10) # Rango de tamaños muestrales evaluados

# Calcular poder para detectar una correlación positiva (prueba unilateral)
power_detect <- map_dbl(sample_sizes, ~
  power_z_cor(
    alternative = "greater", # Prueba unilateral
    alpha = alpha,
    rho = rho_pop, # Correlación poblacional esperada
    n = .x,
    power = NULL
  )$power)

# Calcular poder para concluir equivalencia (correlación equivalente a 0 dentro del margen)
power_equiv <- map_dbl(sample_sizes, ~
  power_z_cor(
    alternative = "equivalence", # Prueba de equivalencia
    alpha = alpha,
    null = eq_bounds, # Margen de equivalencia
    rho = 0, # Correlación nula
    n = .x,
    power = NULL
  )$power)

# Unir ambos resultados en un solo data frame largo
power_df <- tibble(
  n = sample_sizes,
  Detection = power_detect,
  Equivalence = power_equiv
) |>
  pivot_longer(-n, names_to = "Test", values_to = "Power") |>
  mutate(
    Test = str_replace_all(Test, "Detection", "Detección"),
    Test = str_replace_all(Test, "Equivalence", "Equivalencia")
  )

# Texto dinámico para el subtítulo de la figura (con parámetros del análisis)
subtitle_text <- substitute(
  expression("Tamaño del efecto para detección: " * rho == r * ",
            Límites de equivalencia: ±" * eq),
  list(r = rho_pop, eq = eq_bounds)
)

# Tamaño muestral mínimo requerido para alcanzar 90% de poder en equivalencia
final_sample_size_st2 <- power_df |>
  filter(Test == "Equivalencia", Power >= 0.9) |>
  summarise(min_n = min(n)) |>
  pull(min_n)

# Tamaño muestral mínimo requerido para alcanzar 90% de poder en detección
p_sample_size_st2 <- power_df |>
  filter(Test == "Detección", Power >= 0.9) |>

```

```

summarise(min_n = min(n)) |>
pull(min_n)

# Crear figura con curvas de poder para ambas pruebas
ggplot(power_df, aes(x = n, y = Power, color = Test)) +
  geom_line(linewidth = 0.5) + # Línea de poder
  geom_point(size = 1) + # Puntos individuales
  geom_hline(yintercept = 0.8, linetype = "dashed") + # Línea de referencia: 80%
  geom_hline(yintercept = 0.9, linetype = "dashed") + # Línea de referencia: 90%
  geom_vline(xintercept = final_sample_size_st2, colour = "#E31A1C", linetype = "dotted") +
  annotate("text",
    y = 0.5, x = final_sample_size_st2,
    label = bquote(italic(n) == .(final_sample_size_st2)),
    hjust = 0.5, vjust = 1.3, angle = 90,
    color = "#E31A1C", size = 2.5
  ) +
  geom_vline(xintercept = p_sample_size_st2, colour = "#2c3e50", linetype = "dotted") +
  annotate("text",
    y = 0.5, x = p_sample_size_st2,
    label = bquote(italic(n) == .(p_sample_size_st2)),
    hjust = 0.5, vjust = 1.3, angle = 90,
    color = "#2c3e50", size = 2.5
  ) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(
    title = "Curvas de poder para detección y prueba de equivalencia",
    subtitle = eval(subtitle_text),
    x = "Tamaño muestral (n)",
    y = "Poder estadístico",
    color = "Tipo de prueba"
  ) +
  scale_colour_tq() +
  theme_tq()

```

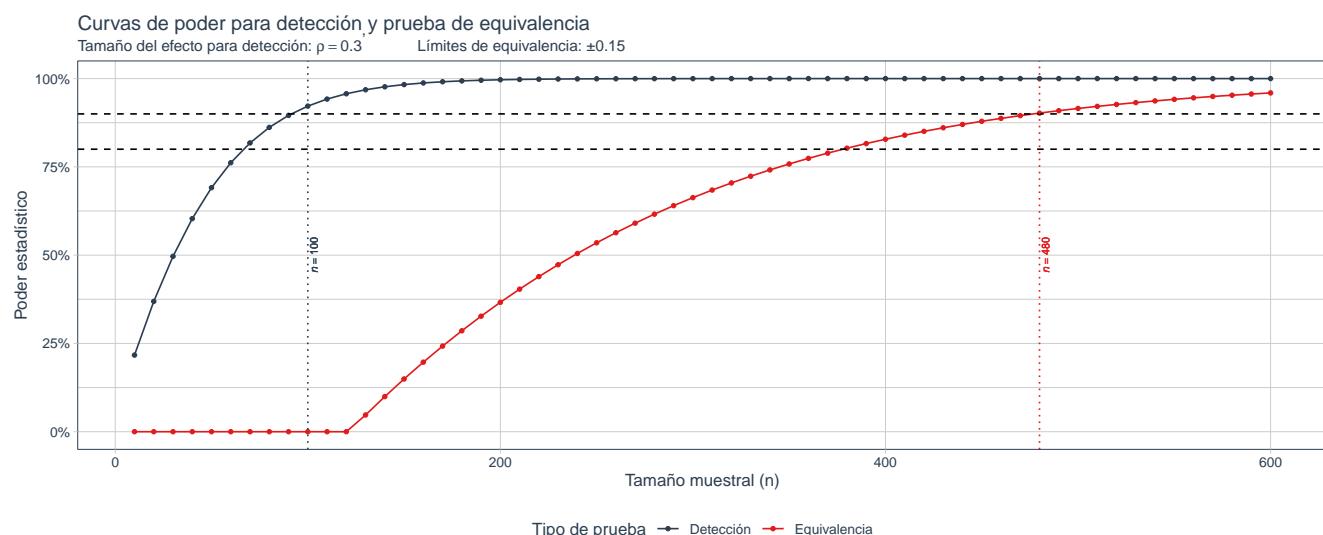


Figura 4. Curvas de poder para detectar una correlación positiva (línea gris) y para concluir equivalencia estadística con cero (línea roja) según el tamaño muestral. Las líneas horizontales punteadas representan los umbrales de poder del 80 % y 90 %. Las líneas verticales punteadas indican el tamaño muestral requerido para alcanzar un poder de $1 - \beta = 0.9$.

En general, con una muestra de 477 participantes, correlaciones de $\rho \geq 0.23$ permiten concluir un efecto positivo distinto de cero; correlaciones de $\rho \leq 0.07$ llevan a concluir equivalencia con cero. Correlaciones entre $\rho = 0.08$ y 0.22 tienden a producir resultados inconclusos (posiblemente significativos, pero también equivalentes a cero).

```
# Panel A: Resultados simulados bajo rho = 0
set.seed(123)
z_sim_0 <- rnorm(n_sim, mean = atanh(0), sd = 1 / sqrt(final_sample_size_st2 - 3))
r_sim_0 <- tanh(z_sim_0)
ci_low_0 <- tanh(z_sim_0 - qnorm(1 - alpha) * (1 / sqrt(final_sample_size_st2 - 3)))
ci_high_0 <- tanh(z_sim_0 + qnorm(1 - alpha) * (1 / sqrt(final_sample_size_st2 - 3)))
equiv_0 <- ci_low_0 > -eq_bounds & ci_high_0 < eq_bounds

df_sim_0 <- tibble(r_obs = r_sim_0, equiv = equiv_0)

# Panel B: Detecciones positivas simuladas bajo rho = 0.3
# Simulamos correlaciones observadas asumiendo una rho verdadera de 0.3
z_sim_3 <- rnorm(n_sim, mean = atanh(0.3), sd = 1 / sqrt(final_sample_size_st2 - 3))
r_sim_3 <- tanh(z_sim_3)

# Calculamos el IC del 90%
se_z <- 1 / sqrt(final_sample_size_st2 - 3)
ci_low_3 <- tanh(z_sim_3 - qnorm(1 - alpha) * se_z)
ci_high_3 <- tanh(z_sim_3 + qnorm(1 - alpha) * se_z)

# Clasificamos los resultados
test_outcome <- case_when(
  ci_low_3 > eq_bounds ~ "Efecto positivo",
  ci_low_3 < 0 & ci_high_3 > 0 ~ "No significativo",
  TRUE ~ "Inconcluso"
)

df_sim_3 <- tibble(
  r_obs = r_sim_3,
  outcome = factor(test_outcome,
    levels = c("Efecto positivo", "Inconcluso", "No significativo")
  )
)

# Cálculo de porcentajes por categoría con columnas x e y para texto
df_sim_0_summary <- df_sim_0 |>
  count(equiv) |>
  mutate(
    perc = scales::percent(n / sum(n), accuracy = 1),
    label = ifelse(equiv, "Equivalencia concluida", "Inconcluso"),
    colour = ifelse(equiv, "#D55E00", "gray40"),
    x = -0.30, # coordenada horizontal para colocar el texto
    y = c(8500, 9000) # ajusta si es necesario según altura de barras
  )

df_sim_3_summary <- df_sim_3 |>
  count(outcome) |>
  mutate(
    perc = scales::percent(n / sum(n), accuracy = 1),
    colour = case_when(
      outcome == "Efecto positivo" ~ "forestgreen",
      outcome == "Inconcluso" ~ "gray40"
    ),
  )
```

```

x = -0.13,
y = c(9000, 8500)
)

# Figura corregida
ggarrange(
  # Panel A
  ggplot(df_sim_0, aes(x = r_obs, fill = equiv)) +
    annotate("rect", xmin = -eq_bounds, xmax = eq_bounds, ymin = 0, ymax = Inf,
            fill = "gray90", alpha = 0.4) +
    annotate("text", x = 0, y = Inf, label = "Equivalente a 0", vjust = 1.5, fontface = "italic", colour =
    geom_histogram(binwidth = 0.01, center = 0, alpha = 0.8, color = "gray30") +
    geom_vline(xintercept = c(-eq_bounds, eq_bounds), linetype = "dashed", color = "gray40") +
    geom_text(data = df_sim_0_summary, aes(x = x, y = y, label = paste(label, ":", perc), colour = colour),
              inherit.aes = FALSE, hjust = 0) +
    scale_colour_identity() +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 10), limits = c(-0.3, 0.3)) +
    scale_fill_manual(
      values = c("TRUE" = "#D55E00", "FALSE" = "gray70"),
      labels = c("Inconcluso", "Equivalencia concluida"),
      name = NULL
    ) +
    labs(
      title = expression("Resultados simulados del test de equivalencia (" * rho == 0 * ")"),
      subtitle = "Las barras sombreadas muestran ICs completamente dentro de ±0.15 (IC del 90%)",
      x = "Correlación observada (r)",
      y = "Frecuencia"
    ) +
    ylim(c(0, 10000)) +
    theme_tq(),
  # Panel B
  ggplot(df_sim_3, aes(x = r_obs, fill = outcome)) +
    annotate("rect", xmin = -eq_bounds, xmax = eq_bounds, ymin = 0, ymax = Inf,
            fill = "gray90", alpha = 0.4) +
    annotate("text", x = 0, y = Inf, label = "Equivalente a 0", vjust = 1.5, fontface = "italic", colour =
    geom_histogram(binwidth = 0.01, center = 0, alpha = 0.85, color = "gray30") +
    geom_vline(xintercept = c(-eq_bounds, eq_bounds), linetype = "dashed", color = "gray40") +
    geom_text(data = df_sim_3_summary, aes(x = x, y = y, label = paste(outcome, ":", perc), colour = colour),
              inherit.aes = FALSE, hjust = 0) +
    scale_colour_identity() +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 10), limits = c(-0.15, 0.45)) +
    scale_fill_manual(
      values = c(
        "Efecto positivo" = "forestgreen",
        "Inconcluso" = "gray70",
        "No significativo" = "red"
      ),
      name = NULL
    ) +
    labs(
      title = expression("Resultados simulados bajo " * rho == 0.3),
      subtitle = "Las barras sombreadas muestran ICs completamente dentro de ±0.15 (IC del 90%)",
      x = "Correlación observada (r)",
      y = "Frecuencia"
    )
)

```

```

ylim(c(0, 10000)) +
theme_tq(),

labels = "AUTO",
nrow = 1,
common.legend = FALSE
)

```

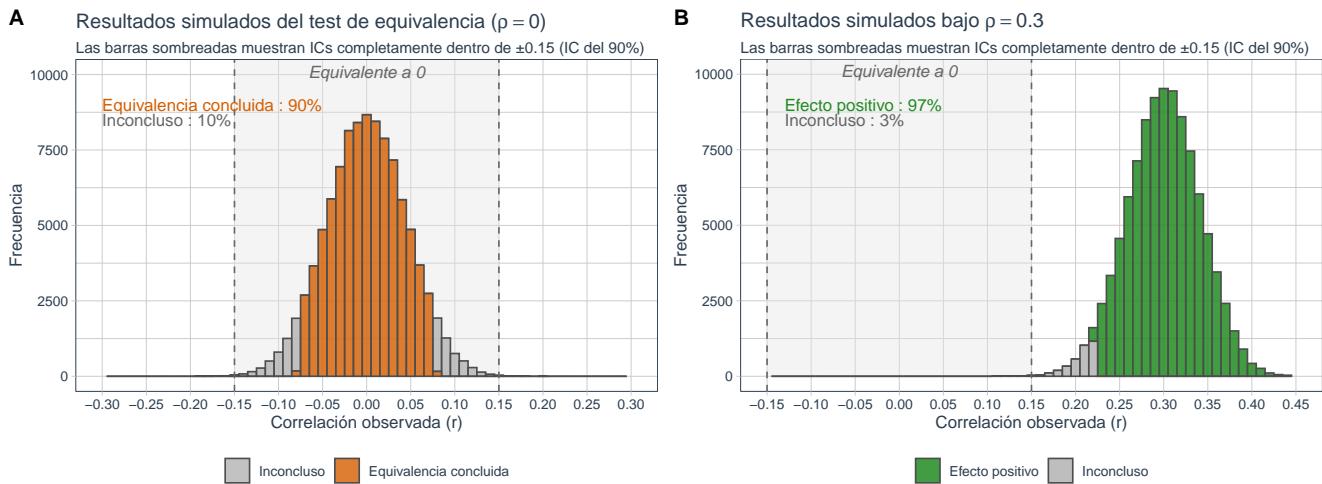


Figura 5. Análisis de poder para detectar o rechazar una correlación de $\rho = 0.30$. (A) Correlaciones observadas simuladas asumiendo que el valor poblacional verdadero es $\rho = 0$ (es decir, bajo la hipótesis nula). Las barras naranjas indican casos en los que el intervalo de confianza (IC) del 90 % cae completamente dentro de los límites de equivalencia (± 0.15), permitiendo concluir equivalencia. Las barras grises representan resultados inconclusos. (B) Correlaciones observadas simuladas asumiendo que la correlación poblacional verdadera es $\rho = 0.3$. Las barras verdes indican efectos positivos claros (el límite inferior del IC del 90 % > 0.15), las rojas muestran resultados no significativos (el IC incluye el 0) y las grises representan resultados inconclusos (el IC está completamente por encima de 0 pero dentro de los límites de equivalencia). En ambos paneles, la banda sombreada y la etiqueta *Equivalente a 0* indican la región definida por los límites de equivalencia, y los porcentajes de probabilidad de cada resultado se muestran en el margen izquierdo.

4. Información de la sesión (para reproducibilidad)

```

library(pander)
pander(sessionInfo(), locale = FALSE)

```

R version 4.5.1 (2025-06-13)

Platform: x86_64-pc-linux-gnu

attached base packages: *stats, graphics, grDevices, utils, datasets, methods* and *base*

other attached packages: *pander(v.0.6.6)*, *kableExtra(v.1.4.0)*, *TOSTER(v.0.8.4)*, *truncnorm(v.1.0-9)*, *ggpubr(v.0.6.0)*, *scales(v.1.4.0)*, *furrr(v.0.3.1)*, *future(v.1.58.0)*, *lmerTest(v.3.1-3)*, *lme4(v.1.1-37)*, *Matrix(v.1.7-3)*, *effectsize(v.1.0.0)*, *PerformanceAnalytics(v.2.0.8)*, *quantmod(v.0.4.28)*, *TTR(v.0.24.4)*, *xts(v.0.14.1)*, *zoo(v.1.8-14)*, *tidyquant(v.1.0.11)*, *ggdist(v.3.3.3)*, *lubridate(v.1.9.4)*, *forcats(v.1.0.0)*, *stringr(v.1.5.1)*, *dplyr(v.1.1.4)*, *purrr(v.1.0.4)*, *readr(v.2.1.5)*, *tidyrr(v.1.3.1)*, *tibble(v.3.2.1)*, *ggplot2(v.3.5.2)*, *tidyverse(v.2.0.0)* and *knitr(v.1.50)*

loaded via a namespace (and not attached): *Rdpack(v.2.6.4)*, *gridExtra(v.2.3)*, *sandwich(v.3.1-1)*, *rlang(v.1.1.6)*, *magrittr(v.2.0.3)*, *multcomp(v.1.4-28)*, *compiler(v.4.5.1)*, *systemfonts(v.1.2.3)*, *vctrs(v.0.6.5)*, *quadprog(v.1.5-8)*, *pkgconfig(v.2.0.3)*, *fastmap(v.1.2.0)*, *backports(v.1.5.0)*, *labeling(v.0.4.3)*, *rmarkdown(v.2.29)*, *tzdb(v.0.5.0)*, *nloptr(v.2.2.1)*, *xfun(v.0.52)*, *broom(v.1.0.8)*, *parallel(v.4.5.1)*, *R6(v.2.6.1)*, *stringi(v.1.8.7)*,

RColorBrewer(v.1.1-3), parallelly(v.1.45.0), car(v.3.1-3), boot(v.1.3-31), numDeriv(v.2016.8-1.1), estimability(v.1.5.1), Rcpp(v.1.0.14), bookdown(v.0.43), parameters(v.0.24.2), splines(v.4.5.1), timechange(v.0.3.0), tidyselect(v.1.2.1), rstudioapi(v.0.17.1), abind(v.1.4-8), yaml(v.2.3.10), codetools(v.0.2-20), curl(v.6.2.2), listenv(v.0.9.1), lattice(v.0.22-7), withr(v.3.0.2), bayestestR(v.0.15.2), coda(v.0.19-4.1), evaluate(v.1.0.4), survival(v.3.8-3), xml2(v.1.3.8), pillar(v.1.10.2), carData(v.3.0-5), reformulas(v.0.4.0), insight(v.1.1.0), distributional(v.0.5.0), generics(v.0.1.3), hms(v.1.1.3), mireqa(v.1.2.8), globals(v.0.18.0), xtable(v.1.8-4), glue(v.1.8.0), emmeans(v.1.11.0), tools(v.4.5.1), ggsignif(v.0.6.4), mvtnorm(v.1.3-3), cowplot(v.1.1.3), grid(v.4.5.1), rbindutils(v.2.3), RobStatTM(v.1.0.11), datawizard(v.1.0.2), nlme(v.3.1-168), Formula(v.1.2-5), cli(v.3.6.5), textshaping(v.1.0.1), viridisLite(v.0.4.2), svglite(v.2.2.1), gtable(v.0.3.6), rstatix(v.0.7.2), digest(v.0.6.37), TH.data(v.1.1-3), farver(v.2.1.2), htmltools(v.0.5.8.1), lifecycle(v.1.0.4) and MASS(v.7.3-65)

5. Referencias

- Ben-Shachar, M. S., Lüdecke, D., & Makowski, D. (2020). *effectsize: Estimation of Effect Size Indices and Standardized Parameters*. *Journal of Open Source Software*, 5(56), 2815. <https://doi.org/10.21105/joss.02815>
- Caldwell, A. R. (2022). Exploring Equivalence Testing with the Updated TOSTER R Package. *PsyArXiv*. <https://doi.org/10.31234/osf.io/ty8de>
- Dancho, M., & Vaughan, D. (2025). *tidyquant: Tidy Quantitative Financial Analysis* [R package version 1.0.1]. <https://doi.org/10.32614/CRAN.package.tidyquant>
- Field, A. (2013). *Discovering statistics using IBM SPSS statistics: And sex and drugs and rock 'n' roll* (4th edition). Sage.
- Kay, M. (2024). *ggdist: Visualizations of Distributions and Uncertainty in the Grammar of Graphics*. *IEEE Transactions on Visualization and Computer Graphics*, 30(1), 414-424. <https://doi.org/10.1109/TVCG.2023.3327195>
- Kay, M. (2025). *ggdist: Visualizations of Distributions and Uncertainty* [R package version 3.3.3]. <https://doi.org/10.5281/zenodo.3879620>
- Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). *lmerTest Package: Tests in Linear Mixed Effects Models*. *Journal of Statistical Software*, 82(13), 1-26. <https://doi.org/10.18637/jss.v082.i13>
- Lakens, D. (2017). Equivalence tests: A practical primer for t-tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 1, 1-8. <https://doi.org/10.1177/1948550617697177>
- Vaughan, D., & Dancho, M. (2022). *furrr: Apply Mapping Functions in Parallel using Futures* [R package version 0.3.1]. <https://doi.org/10.32614/CRAN.package.furrr>
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>
- Wickham, H. (2023). *stringr: Simple, Consistent Wrappers for Common String Operations* [R package version 1.5.1]. <https://doi.org/10.32614/CRAN.package.stringr>
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., ... Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686. <https://doi.org/10.21105/joss.01686>
- Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *dplyr: A Grammar of Data Manipulation* [R package version 1.1.4]. <https://doi.org/10.32614/CRAN.package.dplyr>
- Xie, Y. (2014). *knitr: A Comprehensive Tool for Reproducible Research in R* [ISBN 978-1466561595]. En V. Stodden, F. Leisch & R. D. Peng (Eds.), *Implementing Reproducible Computational Research*. Chapman and Hall/CRC. <https://doi.org/10.1201/9781315373461-1>
- Xie, Y. (2025). *knitr: A General-Purpose Package for Dynamic Report Generation in R* [R package version 1.50]. <https://yihui.org/knitr/>