

# Modelos Lineales con R (I)

## Regresión Lineal

Xavier Barber

Joint Research Unit UMH-FISABIO (StatSalut) & Valencia Bayesian Research group (VaBaR)  
Center of Operations Research. Universidad Miguel Hernández de Elche



UNIVERSITAS  
Miguel Hernández  
RESEARCH INSTITUTE

MH UNIVERSITAS  
Miguel Hernández



# Ejercicios Regresión Lineal

# El problema a resolver

- Como ejemplo se utiliza el conjunto de datos de “temperatura mundial”. (aida)
- Vamos a ajustar un modelo de regresión de avg\_temp en función de year para abordar la “pregunta de investigación” de si el mundo se está calentando o no.

# El problema a resolver

- Más concretamente, evaluaremos si los datos ofrecen motivos para creer que, asumiendo una relación lineal

$$y = \beta_0 + \beta_1 x,$$

donde  $x$  es el año calendario y  $y$  es la temperatura superficial promedio de ese año, el coeficiente  $\beta_1$  es creíblemente positivo.<sup>1</sup>

---

<sup>1</sup>Para no dejar que el elefante se cuele en la habitación: sí, hay modelos mucho mejores para este tipo de datos y pregunta de investigación que un simple modelo de regresión lineal. Pero lo primero es lo primero.

# Objetivos de Aprendizaje

Los objetivos de aprendizaje de este ejercicio son: - Ser capaz de utilizar el paquete `brms` para ajustar modelos de regresión lineal y, en particular, para:

- Especificar un modelo de regresión con una fórmula en R.
- Interpretar el resumen del modelo.
- Extraer muestras posteriores.
- Modificar los priors por defecto.
- Probar hipótesis sobre los coeficientes de regresión.
- La función principal del paquete `brms` es `brm` (abreviatura de Estadística Bayesiana)

# Función Principal de brms

- Aquí se muestra un ejemplo del caso de estudio actual basado en el conjunto de datos de temperatura mundial:

```
#remotes::install_github("michael-franke/aida-package")
library(brms)
library(aida)
fit_temperature <- brm(
  # especificar qué variable explicar en términos de cuál,
  # usando la sintaxis de fórmulas
  formula = avg_temp ~ year,
  # qué datos usar
  data = aida::data_WorldTemp)
```

# Función Principal de brms

- La función principal del paquete `brms` es `brm` (abreviatura de **Bayesian Regression Model**).
- Su comportamiento es muy similar al de la función `glm` que vimos anteriormente <sup>3</sup>.

---

<sup>3</sup>En realidad, `brm` es similar a la función `lmer` del paquete `lme4`, que es más general que `glm`. Tanto `lmer` como `brm` abarcan también los llamados modelos de regresión jerárquica.

# Ejemplo del Caso de Estudio

```
library(brms)
library(aida)
fit_temperature <- brm(
  # Especificar qué variable explicar en términos de cuál,
  # usando la sintaxis de fórmulas
  formula = avg_temp ~ year,
  # Qué datos usar
  data = aida::data_WorldTemp)
```

# Sintaxis de la Fórmula

- La sintaxis de la fórmula  $y \sim x$  le indica a R que queremos explicar o predecir la variable dependiente  $y$  en términos de las mediciones asociadas de  $x$ , tal como se almacenan en el conjunto de datos (ya sea un tibble o un data.frame) suministrado en el argumento data.

# Objeto brmsfit

- El objeto devuelto por la llamada a `brm()` es un objeto especial de la clase `brmsfit`. Si imprimimos este objeto en pantalla, obtenemos un resumen (lo cual también podemos producir mediante la llamada explícita `summary(fit_temperature)`).

```
fit_temperature
```

# Resumen del Modelo

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: avg_temp ~ year
## Data: aida::data_WorldTemp (Number of observations: 269)
## Draws: 4 chains, each with iter = 10000; warmup = 5000; thin = 1;
##         total post-warmup draws = 20000
##
## Regression Coefficients:
##             Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept     -3.51      0.60    -4.71    -2.33 1.00
## year          0.01      0.00     0.01     0.01 1.00
##                 Bulk_ESS Tail_ESS
## Intercept     22965     15721
## year          22962     15591
##
## Further Distributional Parameters:
##             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sigma        0.40      0.02     0.37     0.44 1.00     10768
##                 Tail_ESS
## sigma        8697
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

# Interpretación del Resumen

- Esta salida nos indica qué modelo se ajustó y expone algunas propiedades de la rutina de muestreo MCMC utilizada para obtener muestras de la distribución posterior.
- Los elementos más importantes para extraer conclusiones del análisis son los resúmenes de los parámetros estimados, aquí denominados:
  - **Intercept**: el  $\beta_0$  del modelo de regresión.
  - **year**: el coeficiente de pendiente  $\beta_1$  para la variable year.
  - **sigma**: la desviación estándar de la función de error gaussiano alrededor del predictor central.

El valor “Estimate” de cada parámetro representa su media posterior, y las columnas “l-95% CI” y “u-95% CI” indican el rango intercuantil del 95% de la distribución marginal posterior.

# Obteniendo las distribuciones posteriores

# Extracción de Muestras Posteriore

La función `brms::posterior_samples` extrae las muestras del posterior que forman parte del objeto `brmsfit`.<sup>4</sup>

---

<sup>4</sup>La columna `lp__` proporciona el logaritmo de la probabilidad de los datos para los valores de parámetro correspondientes en cada fila. Es información útil para la comprobación y comparación de modelos, pero aquí la omitiremos.

# Código para Extraer las Muestras

```
post_samples_temperature <-
  as_draws_df(fit_temperature) %>%
  dplyr::select(-lp__, -lprior)
head(post_samples_temperature)
```

# Ejemplo de Salida

```
## # A draws_df: 6 iterations, 1 chains, and 4 variables
##   b_Intercept b_year sigma Intercept
## 1      -2.8 0.0059  0.40      8.3
## 2      -2.9 0.0060  0.41      8.3
## 3      -3.0 0.0060  0.40      8.3
## 4      -2.8 0.0058  0.40      8.3
## 5      -3.7 0.0064  0.39      8.4
## 6      -3.8 0.0064  0.43      8.3
## # ... hidden reserved variables {'chain', 'iteration', 'draw'}
```

# Resumen de las Muestras Extraídas

Estas muestras extraídas se pueden usar, por ejemplo, para calcular nuestro propio resumen en formato tibble:

```
map_dfr(post_samples_temperature[, 1:3],  
        aida::summarize_sample_vector) %>%  
        mutate(Parameter =  
              colnames(post_samples_temperature[, 1:3])))
```

# Ejemplo de Salida del Resumen

```
## # A tibble: 3 x 4
##   Parameter      `|95%`      mean     `|95%|`
##   <chr>          <dbl>      <dbl>      <dbl>
## 1 b_Intercept  -4.65     -3.51     -2.29
## 2 b_year        0.00563   0.00627   0.00689
## 3 sigma         0.370     0.405     0.439
```

# Gráfica Manual de las Muestras

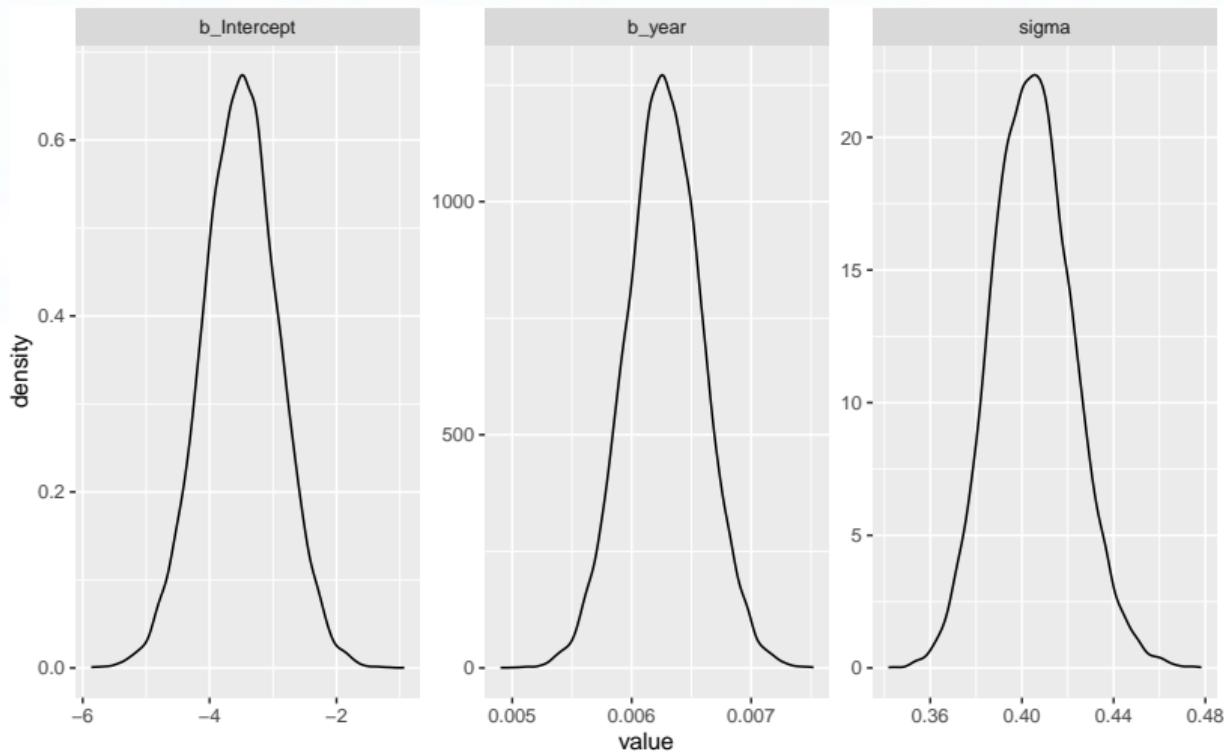
Se puede graficar manualmente:<sup>5</sup>

```
post_samples_temperature %>%
  pivot_longer(cols = everything()) %>%
  ggplot(aes(x = value)) +
  geom_density() +
  facet_wrap(~name, scales = "free")
```

---

<sup>5</sup>Existen paquetes especializados para graficar la salida de modelos Stan y ajustes de modelos brms, como los excelentes paquetes tidybayes y ggdist.

# Ejemplo de Gráfica



# Subamos un peldaño “modelizador”

```
modelo<-brms::stancode(fit_temperature)
## // generated with brms 2.22.0
## functions {
## }
## data {
##   int<lower=1> N;  // total number of observations
##   vector[N] Y;  // response variable
##   int<lower=1> K;  // number of population-level effects
##   matrix[N, K] X;  // population-level design matrix
##   int<lower=1> Kc;  // number of population-level effects after centering
##   int prior_only;  // should the likelihood be ignored?
## }
## transformed data {
##   matrix[N, Kc] Xc;  // centered version of X without an intercept
##   vector[Kc] means_X;  // column means of X before centering
##   for (i in 2:K) {
##     means_X[i - 1] = mean(X[, i]);
##     Xc[, i - 1] = X[, i] - means_X[i - 1];
##   }
## }
```

# subamos un peldaño “modelizador”

```
##  
## parameters {  
##   vector[Kc] b; // regression coefficients  
##   real Intercept; // temporary intercept for centered predictors  
##   real<lower=0> sigma; // dispersion parameter  
## }  
## transformed parameters {  
##   real lprior = 0; // prior contributions to the log posterior  
##   lprior += student_t_lpdf(Intercept | 3, 8.3, 2.5);  
##   lprior += student_t_lpdf(sigma | 3, 0, 2.5)  
##     - 1 * student_t_lccdf(0 | 3, 0, 2.5);  
## }
```

# subamos un peldaño “modelizador”

```
## }
## model {
##   // likelihood including constants
##   if (!prior_only) {
##     target += normal_id_glm_lpdf(Y | Xc, Intercept, b, sigma);
##   }
##   // priors including constants
##   target += lprior;
## }
## generated quantities {
##   // actual population-level intercept
##   real b_Intercept = Intercept - dot_product(means_X, b);
## }
```

# Buscando las priors del modelo

# Priors en Modelos Bayesianos

- Los modelos bayesianos requieren priors para todos los parámetros.
- La función `brms::prior_summary` muestra qué priors ha asumido (implícitamente) un modelo ajustado con `brms`.

# Resumen de Priors del Modelo Original

```
brms::prior_summary(fit_temperature)
```

```
##          prior    class coef group resp dpar
##      (flat)      b
##      (flat)      b year
## student_t(3, 8.3, 2.5) Intercept
## student_t(3, 0, 2.5)   sigma
## nlpar lb ub     source
##           default
##      (vectorized)
##           default
##        0     default
```

# Interpretación de la Salida

- Esta salida nos indica que brms usó una distribución Student's  $t$  para el intercepto y la desviación estándar.<sup>6</sup>
- También muestra que todos los coeficientes de pendiente (abreviados aquí como "b") tienen un prior plano (no informativo).

---

<sup>6</sup>En realidad, la prior sobre la desviación estándar es una distribución Student's  $t$  truncada, ya que no se permiten valores negativos para una desviación estándar.

# Modificando la prior para la Pendiente

- Si queremos cambiar la prior para cualquier parámetro o grupo de parámetros, podemos usar el argumento `prior` en la función `brm` junto con la función `prior()`.
- La sintaxis para las distribuciones dentro de `prior()` sigue la de Stan, según la referencia de funciones de Stan.
- El ejemplo siguiente establece la prior para el coeficiente de la pendiente a una distribución Student's  $t$  muy estrecha con media -0.01 y desviación estándar 0.001.

# Modificando la prior para la Pendiente

```
output2 <- capture.output(fit_temperature_skeptical <- brm(  
  # specify what to explain in terms of what using the formula syntax  
  formula = avg_temp ~ year,  
  # which data to use  
  data = aida::data_WorldTemp,  
  # hand-craft priors for slope  
  prior = prior(student_t(1, -0.01, 0.001), coef = year)  
)
```

- Este prior es un *prior escéptico* en el sentido de que asume que una pendiente negativa es más probable, es decir, que el mundo se ha enfriado a lo largo de los años.

# Comparación de Resúmenes de Ajuste: Modelo Original

- Para el ajuste original, se puede obtener el siguiente resumen:

```
map_dfr(post_samples_temperature[, 1:3],  
        aida::summarize_sample_vector) %>%  
mutate(Parameter =  
       colnames(post_samples_temperature[, 1:3]))
```

# Comparación de Resúmenes de Ajuste: Modelo Original

- Resultado esperado:

```
## # A tibble: 3 x 4
##   Parameter      `|95%`      mean     `|95%|`
##   <chr>          <dbl>      <dbl>      <dbl>
## 1 b_Intercept   -4.65     -3.51     -2.29
## 2 b_year        0.00563   0.00627   0.00689
## 3 sigma         0.370     0.405     0.439
```

# Comparación de Resúmenes de Ajuste: Modelo con Prior Escéptica

- Para el modelo con prior escéptica:

```
post_samples_temperature_skeptical <-
  as_draws_df(fit_temperature_skeptical) %>%
    select(-lp__, -lprior)

summary_table <- map_dfr(post_samples_temperature_skeptical,
  aida::summarize_sample_vector) %>%
  mutate(Parameter = colnames(post_samples_temperature_skeptical))
```

# Comparación de Resúmenes de Ajuste: Modelo con Prior Escéptica

Resultado esperado:

Parameter	P2.5%	Mean	P97.5%
b_Intercept	-4.7244	-3.4968	-2.3224
b_year	0.0056	0.0063	0.0069
sigma	0.3715	0.4048	0.4386

- Observamos que los datos han anulado la prior escéptica inicial, lo que sugiere que la evidencia en los datos para que el coeficiente de la pendiente sea positivo es más fuerte que la suposición original.

# Conclusión

- La comparación de los resúmenes demuestra que el ajuste del modelo con prior escéptico fue sobrepasado por la información contenida en los datos.

# Ejercicio

- **Ejercicio:** ¿Qué esperas que suceda con la estimación del intercepto al usar un prior muy fuerte para el coeficiente de la pendiente de year, por ejemplo, una distribución normal con media 5 y desviación estándar 0.01?

# Ajuste con Prior Muy Fuerte

```

output3 <- capture.output(fit_temperature_ridiculous <- brm(
  # specify what to explain in terms of what using the formula syntax
  formula = avg_temp ~ year,
  # which data to use
  data = aida::data_WorldTemp,
  # hand-craft a very strong prior for slope
  prior = prior(normal(5, 0.01), coef = year)
))
# Usar as_draws_df() (alternativa recomendada a posterior_samples)
post_samples_temperature_ridiculous <- as_draws_df(fit_temperature_ridiculous) %>%
  select(-lp__, -lprior)

# Crear el resumen y asignar los nombres correctos (suponiendo que el resumen tiene 4 filas)
summary_table <- map_dfr(post_samples_temperature_ridiculous, aida::summarize_sample_vector) %>%
  mutate(Parameter = colnames(post_samples_temperature_ridiculous))

# Ver el resumen
kable(summary_table[1:3,], col.names = c("Parameter", "P2.5%", "Mean", "P97.5%"), digits=4)

```

# Ajuste con una Prior Muy Fuerte

- Un resumen de este ajuste podría ser:

Parameter	P2.5%	Mean	P97.5%
b_Intercept	-9444.1613	-9406.5359	-9369.7767
b_year	4.9763	4.9946	5.0150
sigma	354.6828	386.0746	419.5911

# Ajuste con una Prior Muy Fuerte

```
library(bayesplot)
mcmc_hist(post_samples_temperature_ridiculous,
           pars = c("b_Intercept", "b_year", "sigma"))
```

# Ajuste con una Prior Muy Fuerte

```
library(bayesplot)
mcmc_hist(post_samples_temperature_ridiculous,
           pars = c("b_Intercept", "b_year", "sigma"))
```

# Predicciones Posteriores para Datos Ajustados

- La función `brms::posterior_predict` devuelve muestras de la distribución predictiva posterior de un objeto `brms_fit`.
- Por ejemplo, el siguiente código genera 4000 predicciones muestrales para cada uno de los 269 valores de `year` en el conjunto de datos de temperatura mundial.

```
samples_post_pred_temperature <- brms::posterior_predict(fit_temperature)
dim(samples_post_pred_temperature)

## [1] 20000    269
```

# Predicciones Posteriores para Nuevos Datos

- La función `brms::posterior_predict` también se puede usar para obtener muestras de la distribución predictiva posterior de un modelo de regresión ajustado para nuevos valores de los predictores del modelo.
- Si nos interesan las predicciones de la temperatura superficial promedio mundial para los años 2025 y 2040, solo necesitamos suministrar un data frame (o tibble) con los valores de los predictores de interés.

# Predicciones Posteriores para Nuevos Datos

```
# Crear un tibble con nuevos valores de predictores
X_new <- tribble(
  ~year,
  2025,
  2040
)

# Obtener predicciones muestrales del modelo bayesiano
post_pred_new <- brms::posterior_predict(fit_temperature, X_new)

# Obtener un resumen (bayesiano) de estas muestras posteriores
rbind(
  aida::summarize_sample_vector(post_pred_new[,1], "2025"),
  aida::summarize_sample_vector(post_pred_new[,2], "2040")
)
```

# Predicciones Posteriores para Nuevos Datos

```
## # A tibble: 2 x 4
##   Parameter `|95%`    mean `|95%|`
##   <chr>      <dbl> <dbl>  <dbl>
## 1 2025       8.40  9.19   9.99
## 2 2040       8.50  9.29   10.1
```

# Predicciones Posteriores para Nuevos Datos

```

library(tidyverse)

# Supongamos que los datos originales son:
data_original <- aida::data_WorldTemp

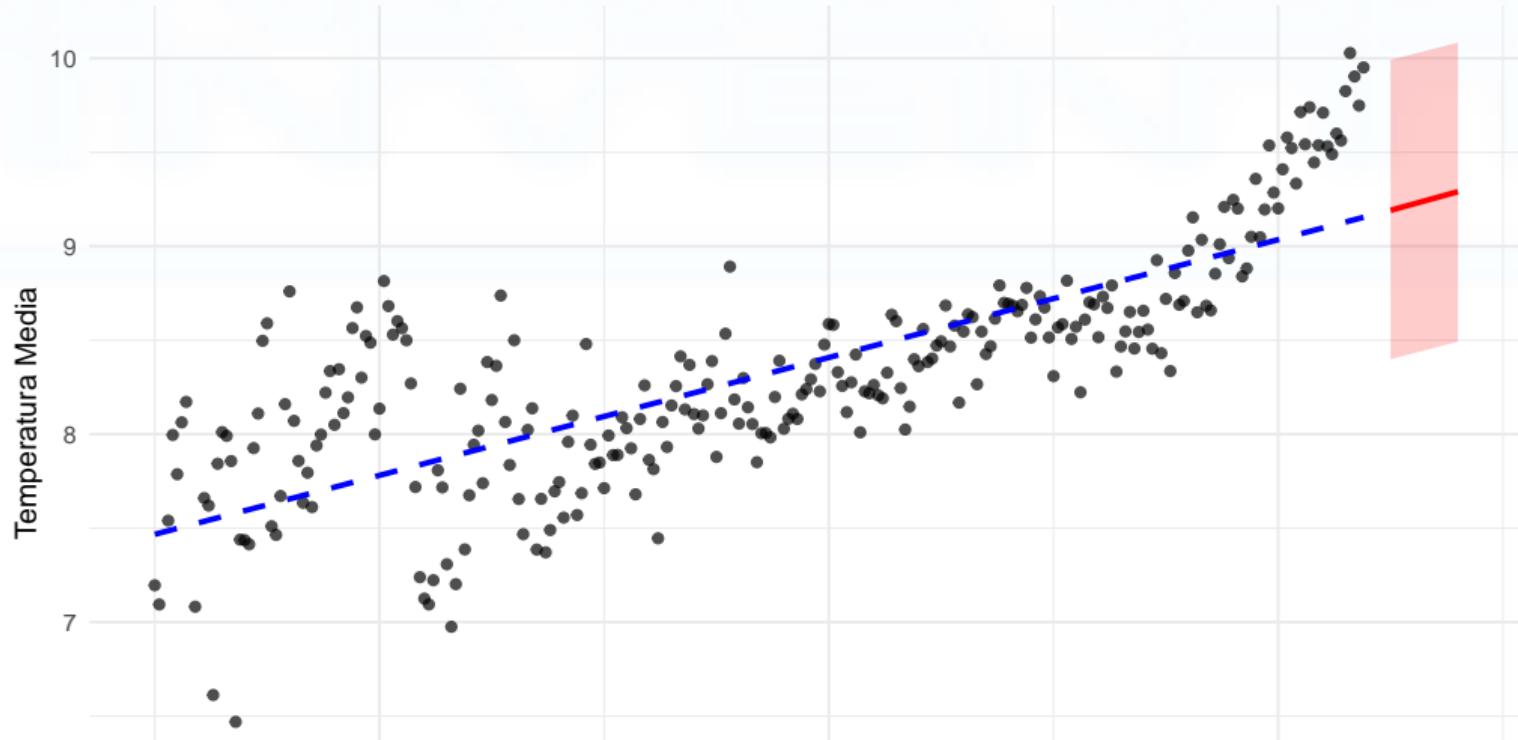
# Obtener resumen de la predicción (media y IC al 95%)
pred_summary <- as_tibble(post_pred_new) %>% #pred_summary # post_pred_new
  pivot_longer(cols = everything(), names_to = "id", values_to = "pred") %>%
  group_by(id) %>%
  summarise(
    avg_temp = mean(pred),
    lower = quantile(pred, 0.025),
    upper = quantile(pred, 0.975)
  ) %>%
  mutate(year = X_new$year)

# Gráfico con ggplot2
ggplot(data_original, aes(x = year, y = avg_temp)) +
  geom_point(color = "black", alpha = 0.7) + # Datos originales
  geom_smooth(method = "lm", se = FALSE, color = "blue", linetype = "dashed") + # Tendencia original
  geom_line(data = pred_summary, aes(x = year, y = avg_temp), color = "red", linewidth = 1) + # Predicción
  geom_ribbon(data = pred_summary, aes(x = year, ymin = lower, ymax = upper))

```

# Predicciones Posteriores para Nuevos Datos

Predicción Bayesiana con Región de Credibilidad



# Contraste de Hipótesis

# Factor Bayes

- El **Factor Bayes (BF)** es una medida que cuantifica la evidencia aportada por los datos a favor de una hipótesis en comparación con otra.

Se define como la razón de verosimilitud entre dos modelos (por ejemplo, el modelo alternativo vs. el modelo nulo).

- $BF > 1$  indica que los datos favorecen la hipótesis alternativa;  $BF < 1$  favorece la hipótesis nula.

# Ventajas y Aplicaciones

- **Comparación directa de modelos:** Permite evaluar la plausibilidad relativa de diferentes hipótesis.
- **No depende de valores p:** Ofrece una alternativa más informativa a las pruebas frecuentistas.
- **Integración de información previa:** Incorpora conocimiento previo en el análisis.

El Factor Bayes se ha popularizado en el análisis bayesiano como una herramienta para comparar modelos y evaluar hipótesis de manera más directa y cuantitativa.

# Definición y Fórmula

El **Factor Bayes (BF)** se define como:

$$BF = \frac{P(D | H_1)}{P(D | H_0)}$$

donde: -  $P(D | H)$  es la **verosimilitud marginal** bajo la hipótesis  $H$ . - Se calcula integrando la verosimilitud multiplicada por el prior:

$$P(D | H) = \int P(D | \theta, H) P(\theta | H) d\theta.$$

# Cálculo Paso a Paso

## 1 Calcular la verosimilitud condicional:

Para cada modelo ( $H_1$  y  $H_0$ ), estima  $P(D | \theta, H)$ .

## 2 Integrar sobre el espacio de parámetros:

Combina la verosimilitud con el prior  $P(\theta | H)$  para obtener la verosimilitud marginal:

$$P(D | H) = \int P(D | \theta, H) P(\theta | H) d\theta.$$

# Cálculo Paso a Paso

## ③ Obtener el Factor Bayes:

Una vez calculadas las evidencias marginales para  $H_1$  y  $H_0$ , se obtiene:

$$BF = \frac{P(D | H_1)}{P(D | H_0)}.$$

# Ejemplo: Escenario Hipotético

- Supongamos que queremos comparar dos hipótesis:
  - $H_1$ : Existe un efecto.
  - $H_0$ : No existe efecto.
- Se han obtenido las siguientes verosimilitudes marginales:
  - $P(D | H_1) = 0.20$
  - $P(D | H_0) = 0.05$

# Cálculo del Factor Bayes

- Aplicamos la fórmula:

$$BF = \frac{P(D | H_1)}{P(D | H_0)} = \frac{0.20}{0.05} = 4.$$

- Esto indica que los datos son 4 veces más probables bajo  $H_1$  que bajo  $H_0$ .

# Interpretación del Ejemplo

- Con un Factor Bayes de 4 se tiene evidencia moderada a favor de  $H_1$ :
  - Los datos respaldan la hipótesis de que existe un efecto.
  - En otras palabras, es razonable preferir  $H_1$  sobre  $H_0$  según la evidencia disponible.

# Contraste de Hipótesis con brms

- El paquete **brms** también contiene una función útil para abordar hipótesis sobre los parámetros del modelo.
- La función `brms::hypothesis` puede calcular ***Bayes Factor*** para hipótesis puntuales utilizando el método de Savage-Dickey.
- Además, calcula una prueba binaria de si una hipótesis es creíble, basándose en su inclusión en un Intervalo de Credibilidad bayesiano.

# Contraste de Hipótesis

- Para hipótesis de intervalo,  $\theta \in [a; b]$ , la función `brms::hypothesis` calcula las probabilidades posteriores (llamadas **ratio de evidencia** en el contexto de esta función):<sup>7</sup>

$$\frac{P(\theta \in [a; b] | D)}{P(\theta \notin [a; b] | D)}$$

---

<sup>7</sup>Nótese que para priors donde  $P(\theta \in [a; b]) = 0.5$ , las probabilidades posteriores son iguales al factor de Bayes.

Para otros priors, habría que corregir las probabilidades posteriores con los priors para obtener factores de Bayes, algo que el paquete **brms** no realiza actualmente.

# Factores de Bayes para Hipótesis Puntuales

- Calcular el ***Bayes Factor*** para hipótesis puntuales con `brms::hypothesis` requiere priors adecuados para todos los parámetros que formen parte de la hipótesis.
- También es necesario tomar muestras de los priors de los parámetros.<sup>8</sup>

---

<sup>8</sup>Puede parecer innecesario tomar muestras de la prior para los parámetros, ya que podríamos consultar la definición cerrada de la prior para ese parámetro. Sin embargo, esto solo funciona para parámetros de primer nivel, no para parámetros en modelos jerárquicos que dependen de otros parámetros.

# Especificando Priors y Obteniendo Muestras de la prior

- A continuación se muestra una llamada a `brm` que:
  - ➊ Especifica un parámetro razonablemente poco restringido pero adecuado para el coeficiente de pendiente de `year`.
  - ➋ Obtiene muestras de la prior (configurando la opción `sample_prior = "yes"`):

# Especificando Priors y Obteniendo Muestras de la prior

```
output <- capture.output(fit_temperature_weakinfo <- brm(  
  # Especificar qué variable explicar en términos de cuál (sintaxis de fórmula)  
  formula = avg_temp ~ year,  
  # Datos a utilizar  
  data = aida::data_WorldTemp,  
  # Prior poco informativo para la pendiente  
  prior = prior(student_t(1, 0, 1), coef = year),  
  # Obtener muestras de la prior  
  sample_prior = "yes",  
  # Aumentar el número de iteraciones para mayor precisión  
  iter = 20000  
)
```

# Resumen de la Distribución Posterior para $\beta_1$

- Antes de abordar hipótesis sobre el parámetro de pendiente para `year`, recordemos las estadísticas resumen de la distribución posterior:

```
# Obtener las muestras posteriores usando as_draws_df
posterior_draws <- as_draws_df(fit_temperature_weakinfo)

# Extraer la columna correspondiente al parámetro 'b_year' y resumirla
posterior_draws %>%
  pull(b_year) %>%
  aida::summarize_sample_vector()
```

# Resumen de la Distribución Posterior para $\beta_1$

```
## # A tibble: 1 x 4
##   Parameter `|95%`      mean  `95%|` 
##   <chr>       <dbl>     <dbl>    <dbl>
## 1 ""          0.00564  0.00627  0.00688
```

# Prueba de Hipótesis: Intervalo para year

- La hipótesis principal de interés es que la pendiente de year es creíblemente positiva.
- Esta es una hipótesis de intervalo y podemos probarla de la siguiente manera:

```
hypothesis(fit_temperature_weakinfo, "year > 0")
```

# Prueba de Hipótesis: Intervalo para year

```
## Hypothesis Tests for class b:  
##   Hypothesis Estimate Est.Error CI.Lower CI.Upper  
## 1 (year) > 0     0.01        0     0.01     0.01  
##   Evid.Ratio Post.Prob Star  
## 1       Inf        1     *  
## ---  
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.  
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;  
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.  
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

# Prueba de Hipótesis: Hipótesis Puntual para year

A continuación se prueba una hipótesis puntual:

```
hypothesis(fit_temperature_weakinfo,  
           "year = 0.005")
```

# Prueba de Hipótesis: Hipótesis Puntual para year

```
## Hypothesis Tests for class b:  
##          Hypothesis Estimate Est.Error CI.Lower CI.Upper  
## 1 (year)-(0.005) = 0      0        0        0        0  
##   Evid.Ratio Post.Prob Star  
## 1       0.56      0.36    *  
## ---  
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.  
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;  
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.  
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

# Interpretación

- La tabla muestra la estimación para la pendiente de `year`, junto con su error, límites inferior y superior del intervalo creíble (95% por defecto).
- El **ratio de evidencia** (`evid.ratio`) para una hipótesis de intervalo no es el factor de Bayes, sino las probabilidades posteriores.
- En este caso, un ratio de evidencia de Inf significa que todas las muestras posteriores para el coeficiente de pendiente fueron positivas.
- En la prueba de hipótesis puntual, la estimación (y su error e intervalo creíble) se calcula como una comparación contra 0.

# Interpretación

- El ratio de evidencia dado es el factor de Bayes de la hipótesis puntual frente al modelo de referencia, calculado mediante el método Savage-Dickey.
- La “Estrella” en la tabla indica que la hipótesis puntual se excluye del intervalo creíble calculado, por lo que, bajo una lógica binaria, rechazaríamos la hipótesis.

# Predictor Categórico

# Introducción

- Los modelos de regresión anteriores se aplicaron a casos en los que se quería predecir una variable métrica  $y$  a partir de mediciones métricas asociadas  $x_i$  ( $1 \leq i \leq n$ ).
- En esta sección, se generaliza este enfoque para cubrir también el caso en que un predictor  $x_i$  es una variable categórica, es decir, un indicador que muestra a qué grupo o condición experimental pertenece una medición de  $y$ .
- Al final de la sección, podremos aplicar modelos de regresión lineal al análisis de mediciones métricas, por ejemplo, en un diseño factorial, muy común en experimentos psicológicos.

# Contraste de Predictores Categóricos

- El truco para generalizar la regresión lineal e incluir predictores categóricos es mapear los niveles de la variable categórica a valores numéricos.
- Por ejemplo, si tenemos dos grupos en un predictor  $x$ , digamos grupo  $A$  y grupo  $B$ , podríamos codificar el grupo  $A$  como  $x = 0$  y el grupo  $B$  como  $x = 1$ .
- Existen muchas codificaciones sensatas (y otras, ridículamente absurdas).
- El término técnico es **contrast coding** (codificación de contrastes): un esquema para transformar las distinciones categóricas en representaciones numéricas que permitan probar fácilmente los contrastes teóricamente interesantes mediante el modelo de regresión resultante.

# Variable categórica con 2 niveles

- Un total de 213 participantes tomaron parte en una versión en línea de una tarea Simon. Los participantes eran estudiantes de Ciencias Cognitivas de la Universidad de Osnabrück, que participaban en los cursos «Introducción a la (Neuro)Psicología Cognitiva» o «Prácticas de laboratorio de Psicología Experimental» en el trimestre de verano de 2019.
- Examinaremos la hipótesis de que, entre todas las respuestas correctas, los tiempos de reacción medios en la condición congruente son menores que los de la condición incongruente.

# Extracción de Datos

Extraemos las columnas relevantes del conjunto de datos:

```
# Extraer únicamente las columnas relevantes del conjunto de datos
data_ST_excerpt <- aida::data_ST %>%
  filter(correctness == "correct") %>%
  select(RT, condition)

# Mostrar las primeras 5 líneas
head(data_ST_excerpt, 5)
```

# Extracción de Datos

```
## # A tibble: 5 x 2
##       RT condition
##   <dbl> <chr>
## 1    735 incongruent
## 2    557 incongruent
## 3    455 congruent
## 4    376 congruent
## 5    626 incongruent
```

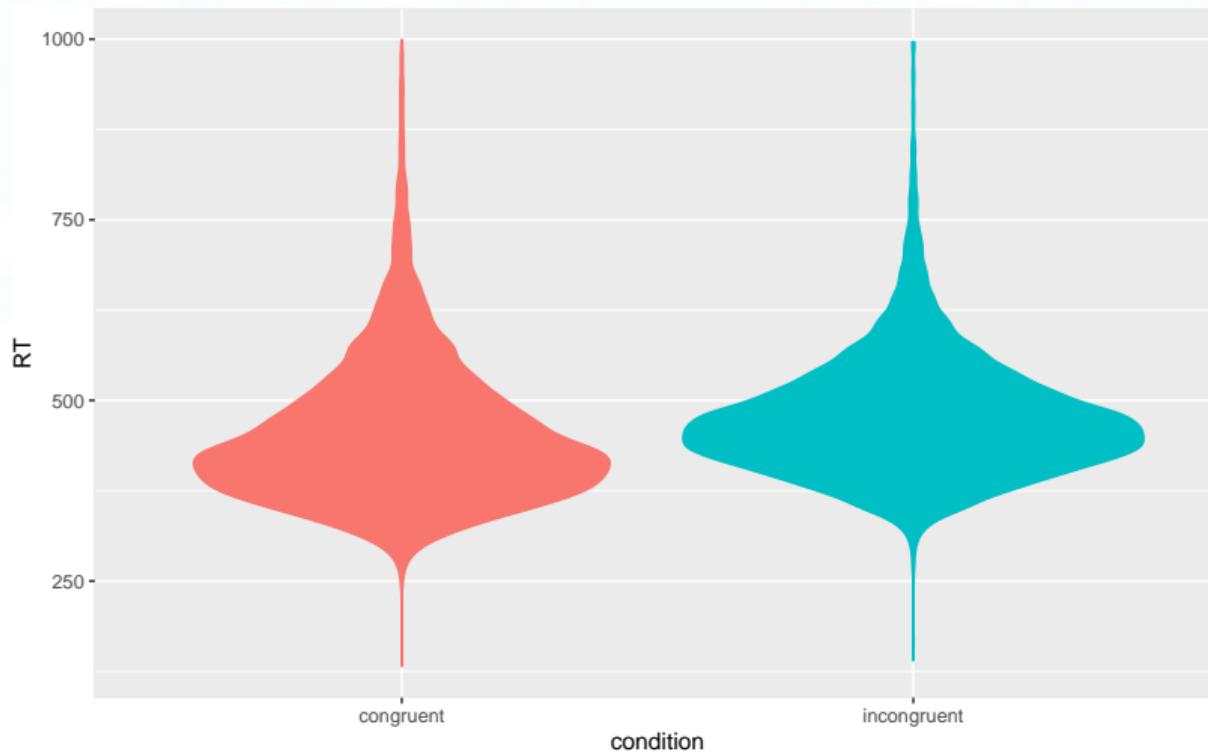
# Visualización de la Distribución de RT

- Visualizamos la distribución de los tiempos de reacción (RT) en cada condición:

```
data_ST_excerpt %>%
  ggplot(aes(x = condition, y = RT,
             color = condition, fill = condition)) +
  geom_violin() +
  theme(legend.position = "none")
```

(Se muestra una imagen que ilustra la distribución de RT en cada condición.)

# Visualización de la Distribución de RT



# Medias por Condición

- Calculamos las medias de RT para cada condición:

```
data_ST_excerpt %>%  
  group_by(condition) %>%  
  summarize(mean_RT = mean(RT))
```

# Medias por Condición

```
## # A tibble: 2 x 2
##   condition   mean_RT
##   <chr>        <dbl>
## 1 congruent    453.
## 2 incongruent 477.
```

# Diferencia Entre Medias

- La diferencia entre las medias de las condiciones es:

```
data_ST_excerpt %>%
  filter(condition == "incongruent") %>%
  pull(RT) %>%
  mean() -
  data_ST_excerpt %>%
  filter(condition == "congruent") %>%
  pull(RT) %>%
  mean()
```

# Diferencia Entre Medias

```
## [1] 23.63348
```

- Aunque numéricamente esta diferencia es considerable, la pregunta es si es lo suficientemente grande como para confiar en ella.
- Abordamos esta cuestión mediante inferencia posterior basada en un modelo de regresión.
- Utilizaremos la misma sintaxis de fórmula que antes: queremos un modelo que explique RT en función de condition.

# Ajuste del Modelo de Regresión

Ajustamos un modelo de regresión lineal con el predictor categórico:

```
fit_brms_ST <- brm(  
  formula = RT ~ condition,  
  data = data_ST_excerpt  
)
```

# Resumen del Modelo

- Revisamos la información resumen de las muestras posteriores de los efectos fijos:

```
summary(fit_brms_ST)$fixed[, c("l-95% CI", "Estimate", "u-95% CI")]
```

```
##                               1-95% CI   Estimate   u-95% CI
## Intercept                 450.91917 452.85721 454.70644
## conditionincongruent    20.96212  23.64485  26.27604
```

# Interpretación

- La variable **Intercept** corresponde a la media de RT en la condición de referencia, es decir, la condición congruente.
- La variable **conditionincongruent** representa la diferencia entre la media de la condición incongruente y la condición congruente.
  - Su valor coincide aproximadamente con la diferencia calculada previamente (23.63348).

# Codificación Interna del Predictor

- ¿Cómo se obtienen estos resultados?
- Para usar un modelo de regresión lineal simple, el predictor categórico  $x$  se codifica como 0 o 1.
- Concretamente, **brms** introduce una nueva variable (llamada, por ejemplo, `new_predictor`) que vale 0 para la condición congruente y 1 para la condición incongruente.
- Por defecto, **brms** elige el nivel alfabéticamente primero (aquí “congruent”) como nivel de referencia, asignándole el valor 0.

# Codificación Interna del Predictor

Ejemplo de codificación:

```
data_ST_excerpt %>%
  mutate(new_predictor = ifelse(condition == "congruent", 0, 1)) %>%
  head(5)

## # A tibble: 5 x 3
##       RT condition  new_predictor
##     <dbl> <chr>        <dbl>
## 1    735 incongruent     1
## 2    557 incongruent     1
## 3    455 congruent      0
## 4    376 congruent      0
## 5    626 incongruent     1
```

# Codificación Interna del Predictor

- Con esta codificación, el modelo de regresión se expresa como:

$$\begin{aligned}\xi_i &= \beta_0 + \beta_1 x_i \\ y_i &\sim \text{Normal}(\mu = \xi_i, \sigma)\end{aligned}$$

- El parámetro  $\beta_0$  (intercepto) se interpreta como la media predicha para el nivel de referencia (condición congruente).
- Para  $x_i = 1$  (condición incongruente), la predicción es  $\beta_0 + \beta_1$ , de modo que  $\beta_1$  representa la diferencia  $\delta$  entre los grupos.
- En otras palabras, podemos considerar una prueba  $t$  como un caso

# Representación Esquemática de la Codificación

La codificación se puede representar esquemáticamente como:

```
## # A tibble: 2 × 3
##   condition      x_0     x_1
##   <chr>        <dbl>  <dbl>
## 1 congruent     1       0
## 2 incongruent   1       1
```

# Variable categórica con múltiples niveles

- El esquema de codificación 0/1 mostrado anteriormente funciona bien para un predictor categórico con dos niveles.
- Sin embargo, es posible utilizar regresión lineal también para predictores categóricos con más de dos niveles.
- En ese caso, existen diversos esquemas de **codificación de contrastes**, es decir, formas de elegir números para codificar los niveles del predictor.

# Variable categórica con múltiples niveles

- El conjunto de datos de **cronometría mental** contiene un único predictor categórico, llamado `block`, con tres niveles:
  - “reaction”
  - “goNoGo”
  - “discrimination”
- Nos interesa ajustar un modelo de regresión en el que se explique el tiempo de reacción (variable `RT`) en función de `block`.

# Variable categórica con múltiples niveles

- Nuestra principal pregunta de interés es si se apoyan en los datos las siguientes desigualdades:

$RT \text{ en "reaction"} < RT \text{ en "goNoGo"} < RT \text{ en "discrimination"}$

- Es decir, nos interesan las diferencias ( $\delta$ ) entre “reaction” y “goNoGo” y entre “discrimination” y “goNoGo”<sup>9</sup>.

---

<sup>9</sup>Para ser precisos, es posible también probar variables aleatorias derivadas de las muestras posteriores. Por lo tanto, no es *necesario* codificar directamente los contrastes de interés. Sin embargo, en la mayoría de los análisis bayesianos tiene sentido asignar priors exactamente a estos  $\delta$  (por ejemplo, priors escépticos sesgados en contra de una hipótesis a probar) y, para eso, es (casi) prácticamente necesario que los contrastes relevantes se expresen como coeficientes de pendiente en el modelo.

# Extracción de Datos

- Consideramos sólo los datos relevantes para nuestros fines actuales:

```
# Seleccionar las columnas relevantes del conjunto de datos
data_MC_excerpt <- aida::data_MC_cleaned %>%
  select(RT, block)

# Mostrar las primeras 5 líneas
head(data_MC_excerpt, 5)
```

# Extracción de Datos

```
## # A tibble: 5 × 2
##       RT block
##   <dbl> <ord>
## 1     311 reaction
## 2     269 reaction
## 3     317 reaction
## 4     325 reaction
## 5     240 reaction
```

# Medias por Nivel de block

- Calculamos las medias de los tiempos de reacción para cada nivel del predictor:

```
data_MC_excerpt %>%
  group_by(block) %>%
  summarize(mean_RT = mean(RT))
```

# Medias por Nivel de block

```
## # A tibble: 3 × 2
##   block      mean_RT
##   <ord>      <dbl>
## 1 reaction    300.
## 2 goNoGo     427.
## 3 discrimination 488.
```

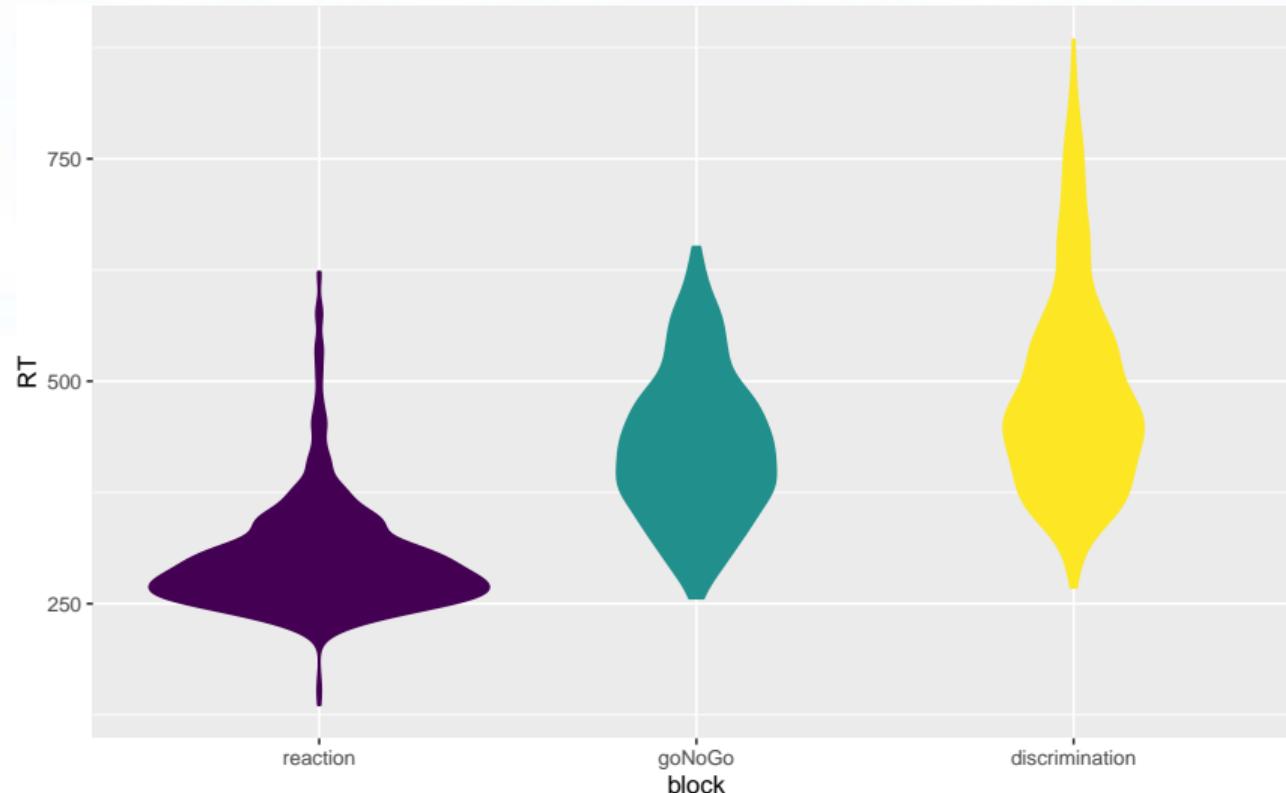
# Visualización de la Distribución de RT

- Visualizamos la distribución de las mediciones de RT en cada bloque:

```
data_MC_excerpt %>%
  ggplot(aes(x = block, y = RT,
             color = block, fill = block)) +
  geom_violin() +
  theme(legend.position = "none")
```

*(Se mostrará una imagen ilustrativa con la distribución de RT para cada nivel de block.)*

# Visualización de la Distribución de RT



# Ajuste del Modelo de Regresión (Codificación Original)

- Utilizando la misma sintaxis de fórmula, ajustamos un modelo de regresión que explica RT en función de block:

```
fit_brms_mc <- brm(  
  formula = RT ~ block,  
  data = data_MC_excerpt  
)
```

# Resumen de los Efectos Fijos

- Inspeccionamos la información resumen para las muestras posteriores de los efectos fijos:

```
summary(fit_brms_mc)$fixed[,  
  c("l-95% CI", "Estimate", "u-95% CI")]
```

# Resumen de los Efectos Fijos

	1-95% CI	Estimate	vu-95% CI
## Intercept	400.98876	404.90896	408.87119
## block.L	127.21126	132.80967	138.33805
## block.Q	-34.71398	-27.28713	-19.86181

# Resumen de los Efectos Fijos

## Interpretación:

- El término **Intercept** corresponde a la media de RT en el nivel de referencia.
  - Por defecto, **brms** selecciona el nivel alfabéticamente primero, que en este caso es “discrimination”.
- Existen dos coeficientes de pendiente:
  - **block.L**: La diferencia entre el nivel de referencia (“discrimination”) y otro nivel (en este caso, “goNoGo”).
  - **block.Q**: La diferencia entre el nivel restante (“reaction”) y el nivel de referencia.
- Estos resultados sugieren que los tiempos de reacción son mayores en la condición “discrimination” y las diferencias entre los niveles son significativas.

# Cambiando el Nivel de Referencia

- Para poder interpretar directamente las comparaciones que nos interesan (por ejemplo, queremos que el nivel de referencia sea el “nivel medio”), basta con cambiar el nivel de referencia.
- Convertimos `block` en factor y ordenamos manualmente sus niveles de acuerdo con la hipótesis ordenada:

```
data_MC_excerpt <- data_MC_excerpt %>%
  mutate(block_reordered = factor(block,
    levels = c("goNoGo", "reaction", "discrimination")))
```

# Cambiando el Nivel de Referencia

- Luego, ajustamos otro modelo de regresión usando esta nueva variable:

```
fit_brms_mc_reordered <- brm(  
  formula = RT ~ block_reordered,  
  data = data_MC_excerpt  
)
```

# Resumen del Modelo con Nuevo Nivel de Referencia

- Inspeccionamos el resumen de los efectos fijos:

```
summary(fit_brms_mc_reordered)$fixed[,  
  c("l-95% CI", "Estimate", "u-95% CI")]
```

```
##                               l-95% CI Estimate u-95% CI  
## Intercept             401.16332 404.89229 408.64559  
## block_reordered.L   35.94682  42.79215  49.73385  
## block_reordered.Q 122.74315 128.61676 134.46136
```

# Resumen del Modelo con Nuevo Nivel de Referencia

## Interpretación:

- Ahora el **Intercept** corresponde a la media de RT en el nivel de referencia “goNoGo”.
- Los coeficientes de pendiente:
  - **block\_reordered.L** representa la diferencia entre “reaction” y “goNoGo”.
  - **block\_reordered.Q** representa la diferencia entre “discrimination” y “goNoGo”.
- De esta manera, los parámetros del modelo se interpretan directamente como las diferencias que nos interesa evaluar.

# Codificación Esquemática de las Variables

- La codificación que conduce a este resultado se puede representar de forma esquemática como:

```
## # A tibble: 3 × 4
##   block      x_0     x_1     x_2
##   <chr>    <dbl>   <dbl>   <dbl>
## 1 goNoGo     1       0       0
## 2 reaction    1       1       0
## 3 discrimination  1       0       1
```

# Codificación Esquemática de las Variables

- Aquí:
  - $x_0 = 1$  para todas las observaciones (constante).
  - $x_1 = 1$  si la observación pertenece a “reaction” y 0 en caso contrario.
  - $x_2 = 1$  si la observación pertenece a “discrimination” y 0 en caso contrario.
- De esta forma, el modelo se expresa como:

$$\xi_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$$

- Siendo:
  - $\beta_0$  es la media de RT en “goNoGo” (nivel de referencia).

# Conclusión

- La codificación de niveles en función de un nivel de referencia (llamada **codificación de tratamiento** o **dummy coding**) permite que los coeficientes del modelo expresen directamente las diferencias entre grupos.
- Esto facilita la interpretación y el testeo de hipótesis de contraste directamente a partir del modelo de regresión lineal.

# Multiple predictores

# Bayesian Facotrial Designs

- Los diseños factoriales con múltiples predictores categóricos son comunes en la psicología experimental, en Agronomía, en ciencias de la salud, en macroeconomía, etc.
- Cada predictor categórico puede codificarse con diferentes esquemas, y su interacción debe considerarse en los modelos.

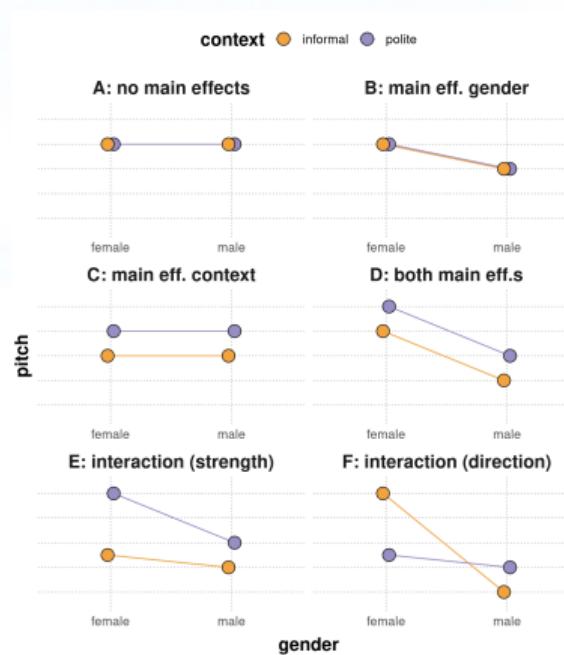
# Ejemplo: Sabrosidad de un Snack

- Dos factores categóricos influyen en la percepción de sabrosidad:
  - **Mayo:** 0 = sin mayonesa, 1 = con mayonesa
  - **Chocolate:** 0 = sin chocolate, 1 = con chocolate
- Se encuentran efectos positivos para ambos factores por separado, pero ¿qué pasa si el snack contiene ambos?

# Interacción entre Predictores

- Cuando hay múltiples predictores categóricos, se pueden incluir términos de interacción en el modelo.
- Un término de interacción captura cómo los efectos de un predictor cambian dependiendo del otro.

# Interacción entre Predictores



# Ejemplo: Datos de Politeness

- Se estudia el tono de voz en un diseño factorial  $2 \times 2$  con:
  - **Género:** Hombre / Mujer
  - **Contexto:** Informal / Cortés

# Carga de Datos en R

```
politeness_data <- aida::data_polite  
head(politeness_data, 5)
```

```
## # A tibble: 5 x 5  
##   subject gender sentence context pitch  
##   <chr>    <chr>    <chr>    <chr>    <dbl>  
## 1 F1       F        S1       pol      213.  
## 2 F1       F        S1       inf      204.  
## 3 F1       F        S2       pol      285.  
## 4 F1       F        S2       inf      260.  
## 5 F1       F        S3       pol      204.
```

# Hipótesis de Investigación

- ① **H1:** El tono de voz de los hombres es menor que el de las mujeres.
- ② **H2:** El tono de voz en contextos corteses es menor que en contextos informales.
- ③ **H3:** El efecto del contexto es más fuerte en mujeres que en hombres.

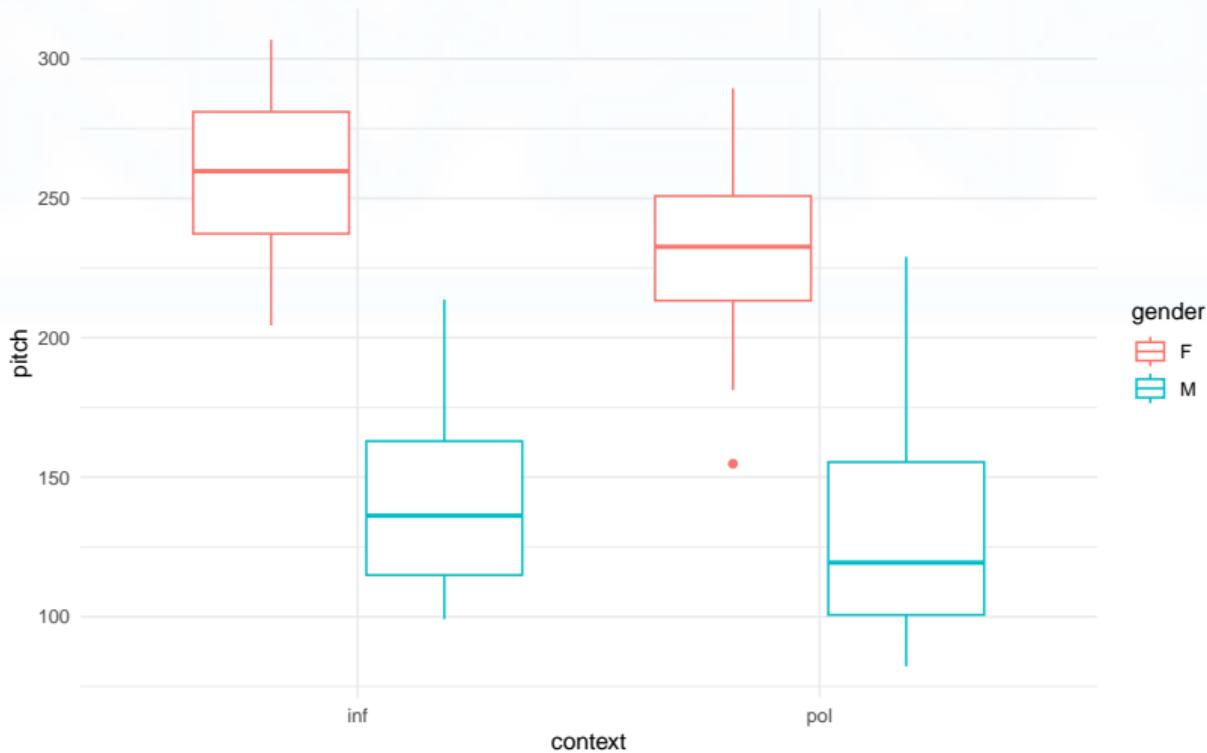
# Diagramas de Interpretación

- Existen varios tipos de interacciones posibles en un diseño factorial  $2 \times 2$ :
  - **Efecto de género**
  - **Efecto de contexto**
  - **Interacción entre ambos**

# Inspección Visual de los Datos

```
ggplot(politeness_data, aes(x = context,  
                             y = pitch, color = gender)) +  
  geom_boxplot() +  
  theme_minimal()
```

# Inspección Visual de los Datos



# Modelado en brms

- Se ajusta un modelo de regresión bayesiano con términos de interacción:

```
fit_brms_politeness <- brm(  
  pitch ~ gender * context,  
  data = politeness_data  
)
```

# Resumen de Resultados

```
summary(fit_brms_politeness)$fixed[,  
  c("l-95% CI", "Estimate", "u-95% CI")]
```

	l-95% CI	Estimate	u-95% CI
## Intercept	244.65022	260.29315	275.780319
## genderM	-137.76195	-115.55019	-93.222550
## contextpol	-48.08341	-26.95697	-4.222726
## genderM:contextpol	-16.42808	14.96744	45.031342

# Pruebas de Hipótesis en brms

## H1: Efecto de género

```
brms::hypothesis(fit_brms_politeness,  
                  "genderM + 0.5 * genderM:contextpol < 0")
```

```
## Hypothesis Tests for class b:  
##                                Hypothesis Estimate Est.Error CI.Lower  
## 1 (genderM+0.5*gend... < 0    -108.07      8.13   -121.72  
##   CI.Upper Evid.Ratio Post.Prob Star  
## 1     -95.01        Inf         1      *  
## ---  
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.  
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;  
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.  
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

# Pruebas de Hipótesis en brms

## H2: Efecto de contexto

```
brms::hypothesis(fit_brms_politeness,  
                  "contextpol + 0.5 * genderM:contextpol < 0")
```

```
## Hypothesis Tests for class b:  
##                                Hypothesis Estimate Est.Error CI.Lower  
## 1 (contextpol+0.5*g... < 0    -19.47      8.18   -32.68  
##   CI.Upper Evid.Ratio Post.Prob Star  
## 1     -5.79      113.29      0.99      *  
## ---  
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.  
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;  
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.  
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

# Pruebas de Hipótesis en brms

## H3: Interacción

```
brms::hypothesis(fit_brms_politeness,  
                  "genderM:contextpol > 0")
```

```
## Hypothesis Tests for class b:  
##                                Hypothesis Estimate Est.Error CI.Lower  
## 1 (genderM:contextpol) > 0      14.97      15.6   -11.07  
##   CI.Upper Evid.Ratio Post.Prob Star  
## 1     40.08      5.14      0.84  
## ---  
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.  
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;  
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.  
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

# Conclusiones

- Podemos interpretar esto como que, dados el modelo y los datos, es plausible pensar que los hablantes masculinos presentan un tono de voz más bajo que las hablantes femeninas (al promediar ambos contextos).
- También podemos concluir que, dados el modelo y los datos, es plausible pensar que el tono de voz es menor en contextos corteses que en contextos informales (promediado sobre ambos niveles del factor gender).

# Conclusiones

- La posterior del término de interacción genderM:contextpol no indica que 0, o algún valor cercano a él, sea poco plausible.
- Esto se puede interpretar como que, dados el modelo y los datos, no hay indicación de creer que el cambio en el tono de voz de los hablantes masculinos al pasar de contextos informales a corteses difiere del cambio observado en las hablantes femeninas.

# Conclusiones

- Se encuentra evidencia fuerte para los efectos principales de género y contexto.
- No se encuentra evidencia significativa para la interacción.
- La codificación de contrastes puede influir en la interpretación de los coeficientes.

# Referencias

- Franke, M. & Roettger, T. B. (2019). *Bayesian Regression Modeling (for Factorial Designs): A Tutorial.*
- Jaeger, T. F. (2008). *Categorical Data Analysis: Away from ANOVAs and Towards Logit Mixed Models.*

