

# Más Allá de Máxima Verosimilitud

*Introducción a la Psicometría Bayesiana*

Ricardo Rey-Sáez



*ricardoreysaez95@gmail.com*

*Universidad Autónoma de Madrid*

February 27, 2025



# Índice

- **Universo** frecuentista, **multiverso** bayesiano
  - Método de los momentos y máxima verosimilitud
  - Estimación bayesiana
- Introducción a la psicometría bayesiana
- Estimación a través de **{blavaan}**
  - Prior predictive checks
  - Convergencia y eficiencia
  - Bondad de ajuste
  - Comparación de modelos
  - Posterior predictive checks
- Estimación a través de *Stan* y **{cmdstanr}**

# **Universo frecuentista, multiverso bayesiano**

## Frecuentismo: ¿qué edad tenéis?

- La media de edad es **una cantidad fija, pero desconocida** ( $\mu_{\text{edad}}$ ).
- El objetivo es estimar  $\mu_{\text{edad}}$  **usando una muestra aleatoria**.
  1. **Elegimos un estimador:** la media muestral.
  2. **Cuantificamos la incertidumbre:** el error típico de la media (SE).
  3. Usamos el SE y construimos un **intervalo de confianza** (IC) para inferir el valor de  $\mu_{\text{edad}}$ .
- Un frequentista concluirá:
  - La media de edad estimada es  $\hat{\mu}_{\text{edad}}$  con un IC de  $[a, b]$ .
  - Si repetimos este experimento infinitas veces, el IC contendrá el valor de  $\mu_{\text{edad}}$  en el 95% de los casos.

## Bayesiano: ¿qué edad tenéis?

- La media de edad es **una variable aleatoria** porque incluimos **nuestra incertidumbre sobre su valor con una distribución de probabilidad.**
- El objetivo es combinar (1) los datos y (2) lo que creemos.
  1. Elegimos una **distribución a priori** para  $\mu_{\text{edad}}$ :  $\Pr(\mu_{\text{edad}})$

### ¿Cómo incluimos nuestra incertidumbre en un modelo estadístico?

Si creo que vuestra edad está **entre 22 y 26 años**, puedo expresar esta creencia usando una distribución de probabilidad. Por ejemplo, puedo utilizar **la distribución normal** para reflejar lo que creo:

$$\Pr(\mu_{\text{edad}}) \sim \mathcal{N}(\mu = 24, \sigma = 1)$$

Esto significa que espero que la media de edad esté en torno a **24 años** y que, al mismo tiempo, creo que la edad del 95% de vosotros estará, aproxiadamente, entre 22 y 26 años:

- Intervalo de confianza:  $\mu \pm 1.96 \cdot \sigma \implies 24 \pm 1.96 \times 1 \approx [22, 26]$

## Bayesiano: ¿qué edad tenéis?

- La media de edad es **una variable aleatoria** porque incluimos **nuestra incertidumbre sobre su valor con una distribución de probabilidad.**
- El objetivo es combinar (1) los datos y (2) lo que creemos.
  1. Elegimos una **distribución a priori** para  $\mu_{\text{edad}}$ :  $\Pr(\mu_{\text{edad}})$
  2. Combinamos la prior y los datos con el **Teorema de Bayes**.

$$\Pr(\mu_{\text{edad}} \mid D) \propto \Pr(D \mid \mu_{\text{edad}}) \times \Pr(\mu_{\text{edad}})$$

## Bayesiano: ¿qué edad tenéis?

- La media de edad es **una variable aleatoria** porque incluimos **nuestra incertidumbre sobre su valor con una distribución de probabilidad.**
- El objetivo es combinar (1) los datos y (2) lo que creemos.
  1. Elegimos una **distribución a priori** para  $\mu_{\text{edad}}$ :  $\Pr(\mu_{\text{edad}})$
  2. Combinamos la prior y los datos con el **Teorema de Bayes**.
  3. Esta combinación es la **distribución posterior**.
- Un bayesiano concluirá:
  - Tras ver los datos, mi antigua creencia (**prior**) sobre  $\mu_{\text{edad}}$  se ha actualizado (**posterior**) .
  - El **intervalo de credibilidad** al 95% nos dice que **hay un 95% de probabilidad** de que  $\mu_{\text{edad}}$  esté en  $[a, b]$

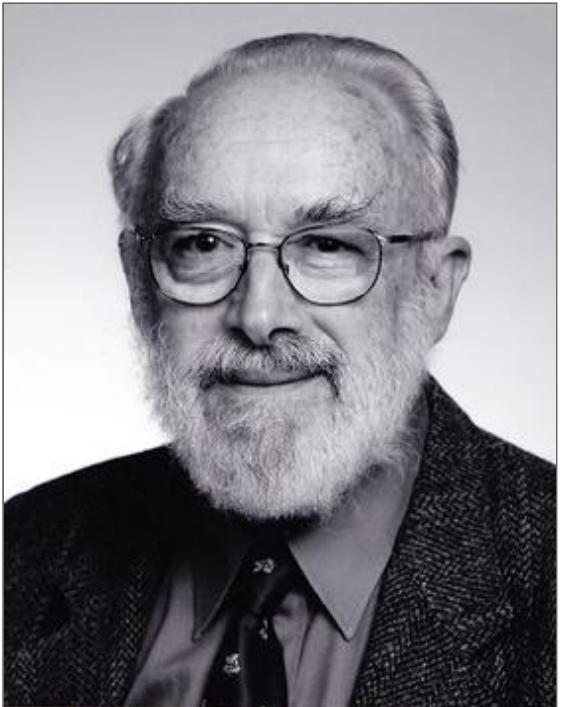
# Universo frecuentista, multiverso bayesiano

Pierre-Simon Laplace



*La inferencia Bayesiana es sentido común expresado en números.*

Dennis Lindley



*La única estadística buena.*

# Universo frecuentista, multiverso bayesiano

## Frecuentismo

- **Probabilidad:** Frecuencia de eventos en repeticiones infinitas
- **Parámetros:** Constantes desconocidas
- **Incertidumbre:** Variabilidad del muestreo

## Bayesiano

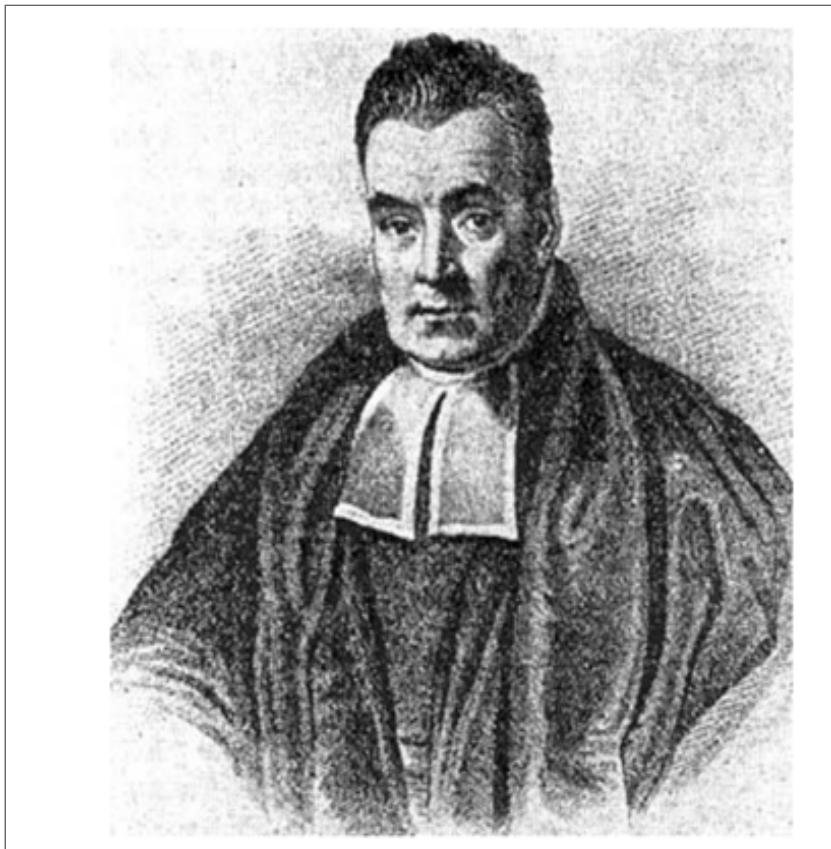
- **Probabilidad:** Grado de creencia sobre un evento o parámetro
- **Parámetros:** Variables con una distribución de probabilidad
- **Incertidumbre:** Refleja el conocimiento antes y después de observar los datos

¿Cuál de las dos es más antigua?

Formalmente, la **estadística bayesiana**, que surge en el siglo XVIII.

# Universo frecuentista, multiverso bayesiano

Thomas Bayes



*El único procedimiento estadístico que es coherente, es decir, que evita afirmaciones*

*que sean internamente inconsistentes.*

### **El hombre de la foto no es Thomas Bayes**

Aun así, este autorretrato sigue usándose en los cursos de estadística Bayesiana.

## Universo frecuentista, multiverso bayesiano

Cada año, Carmen Ximénez y Javier Revuelta, como coordinadores del Máster de Metodología de las Ciencias del Comportamiento y de la Salud, deben decidir cuántos Trabajos de Fin de Máster (TFM) de investigación y de prácticas van a ofertar.

Para ello, necesitan conocer la proporción esperada de estudiantes que optarán por un TFM de investigación. Denotemos esta proporción como  $\theta$ , un valor desconocido que podemos estimar a partir de los datos del curso anterior.

```
1 # TFM de investigación el último año  
2 TFM <- c(1,0,0,1,1,1,0,1,1,0,  
3           0,0,1,0,1,1,1,1,0,1)
```

## Estimando $\theta$ : método de los momentos

En esta situación, el *método de los momentos* es la media aritmética de la variable [TFM](#).

```
1 # Estimación de \theta con el método de los momentos  
2 mean(TFM)
```

[1] 0.6

Como  $\hat{\theta} = 0.6$ , el 60% de los alumnos escogieron un TFM de investigación el año pasado.

## Estimando $\theta$ : máxima verosimilitud

- Necesitamos un **modelo de probabilidad** adecuado.
  - Dado que cada estudiante elige entre dos opciones, podemos modelar su decisión con la **distribución de Bernoulli**

$$\Pr(\text{TFM}_i = 1) \sim \text{Bernoulli}(\theta)$$

- Probabilidad de escoger un TFM de investigación:  $\theta$
- Probabilidad de escoger un TFM de prácticas:  $1 - \theta$
- En R, la ecuación anterior se representa con `dbinom`, indicando `size=1`.

```
1 # Función de densidad de bernoulli  
2 dbinom(x = "TFM_i", size = 1, prob = "theta")
```

## Estimando $\theta$ : máxima verosimilitud

- Sin embargo, nuestro interés no está en cada estudiante por separado, sino en la **verosimilitud conjunta de los datos**.
- Cada alumno elige un TFM **independientemente** de lo que elijan los demás.
- La **verosimilitud conjunta** se obtiene **multiplicando las verosimilitudes individuales de los alumnos**.

$$\Pr(D \mid \theta) = \prod_{i=1}^N \text{Bernoulli}(\text{TFM}_i \mid \theta)$$

```
1 # Función de verosimilitud de los datos  
2 prod(dbinom(x = "TFM_i", size = 1, prob = "theta"))
```

## Estimando $\theta$ : máxima verosimilitud

- ¿Cómo de verosímil es que  $\theta = 0.2$  en nuestros datos?

```
1 # Verosimilitud de los datos cuando \theta = 0.2  
2 prod(dbinom(TFM, size = 1, prob = 0.2))
```

[1] 6.871948e-10

- El valor 6.871948e-10 representa la verosimilitud de los datos dado un valor de  $\theta = 0.2$ . ¿Qué tal si probamos ahora  $\theta = 0.3$ ?

```
1 # Verosimilitud de los datos cuando \theta = 0.3  
2 prod(dbinom(TFM, size = 1, prob = 0.3))
```

[1] 3.063652e-08

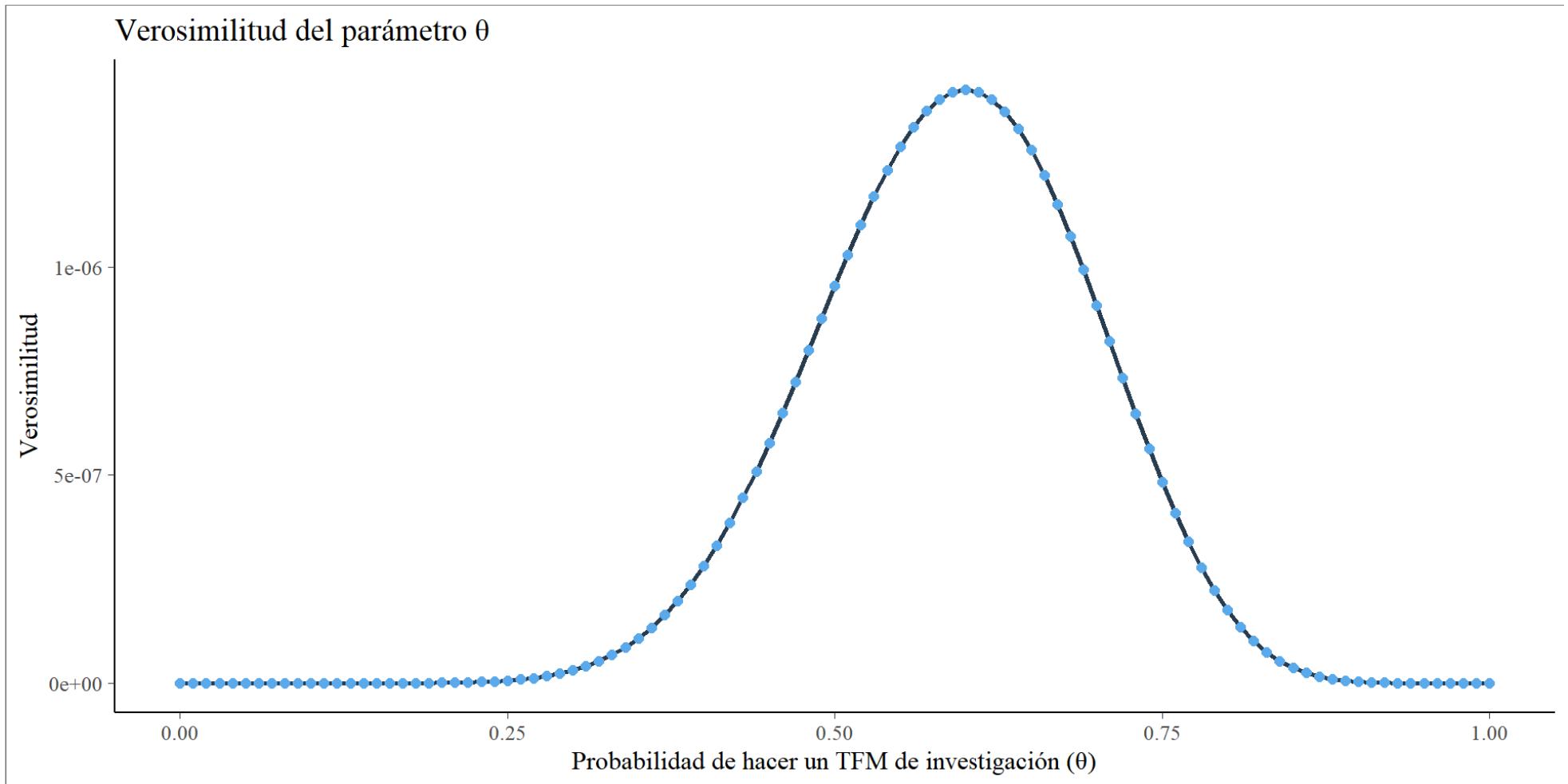
- Como 3.063652e-08 es mayor que 6.871948e-10,  $\theta = 0.3$  explica mejor los datos que  $\theta = 0.2$ .

## Estimando $\theta$ : máxima verosimilitud

¿Qué pasa si repetimos este mismo proceso para todos los valores de  $\theta$  entre 0 y 1?

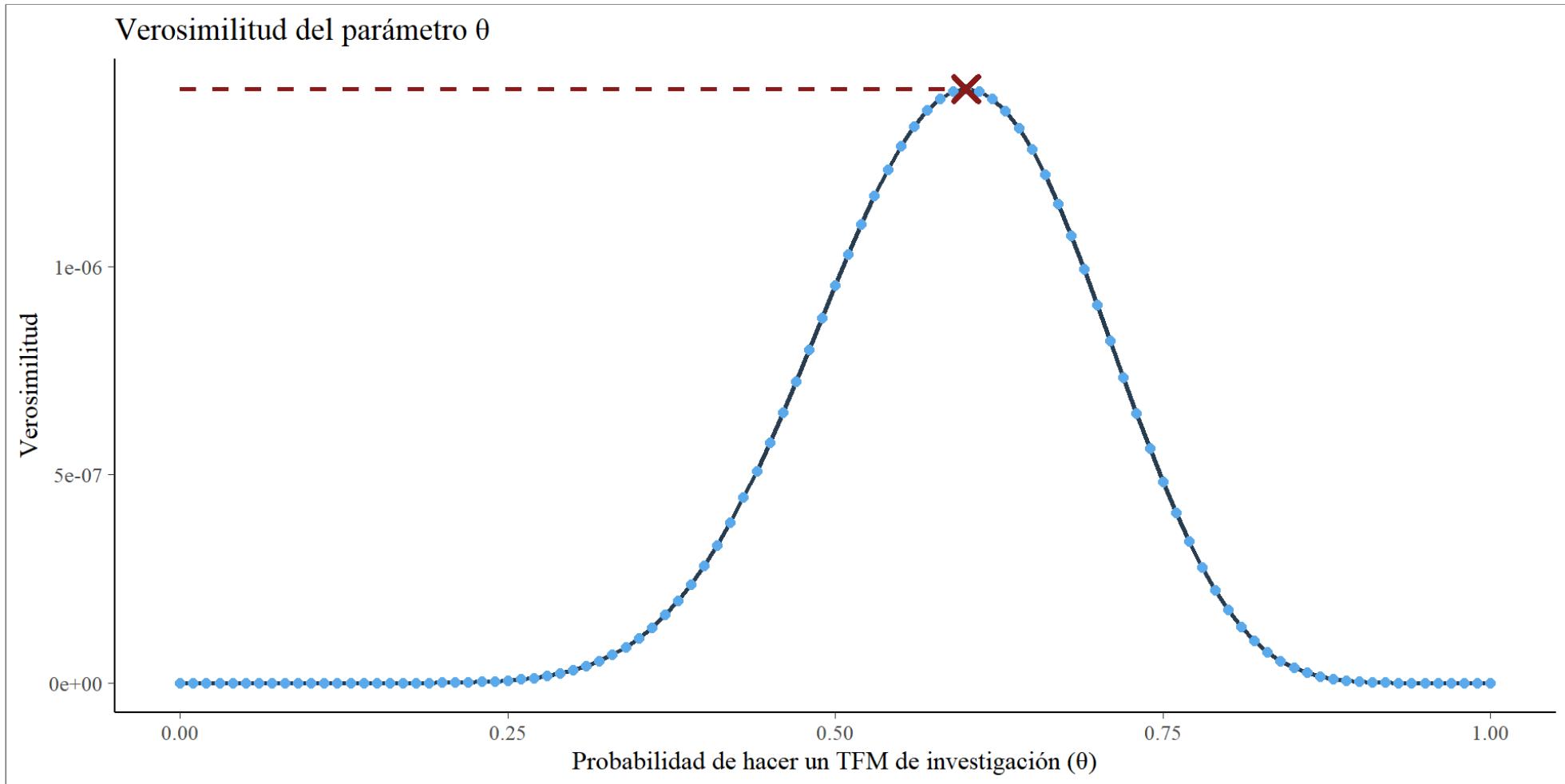
```
1 # Valores de theta de 0 a 1
2 theta <- seq(0, 1, 0.01)
3 # Vector vacío para guardar la verosimilitud con cada theta
4 likelihood <- vector(length = length(theta))
5 # Verosimilitud para cada theta
6 for(i in 1:length(theta)){
7   likelihood[i] <- prod(dbinom(TFM, size = 1, prob = theta
8 })
9 # Gráfico de densidad
10 plot(theta, likelihood)
```

## Estimando $\theta$ : máxima verosimilitud



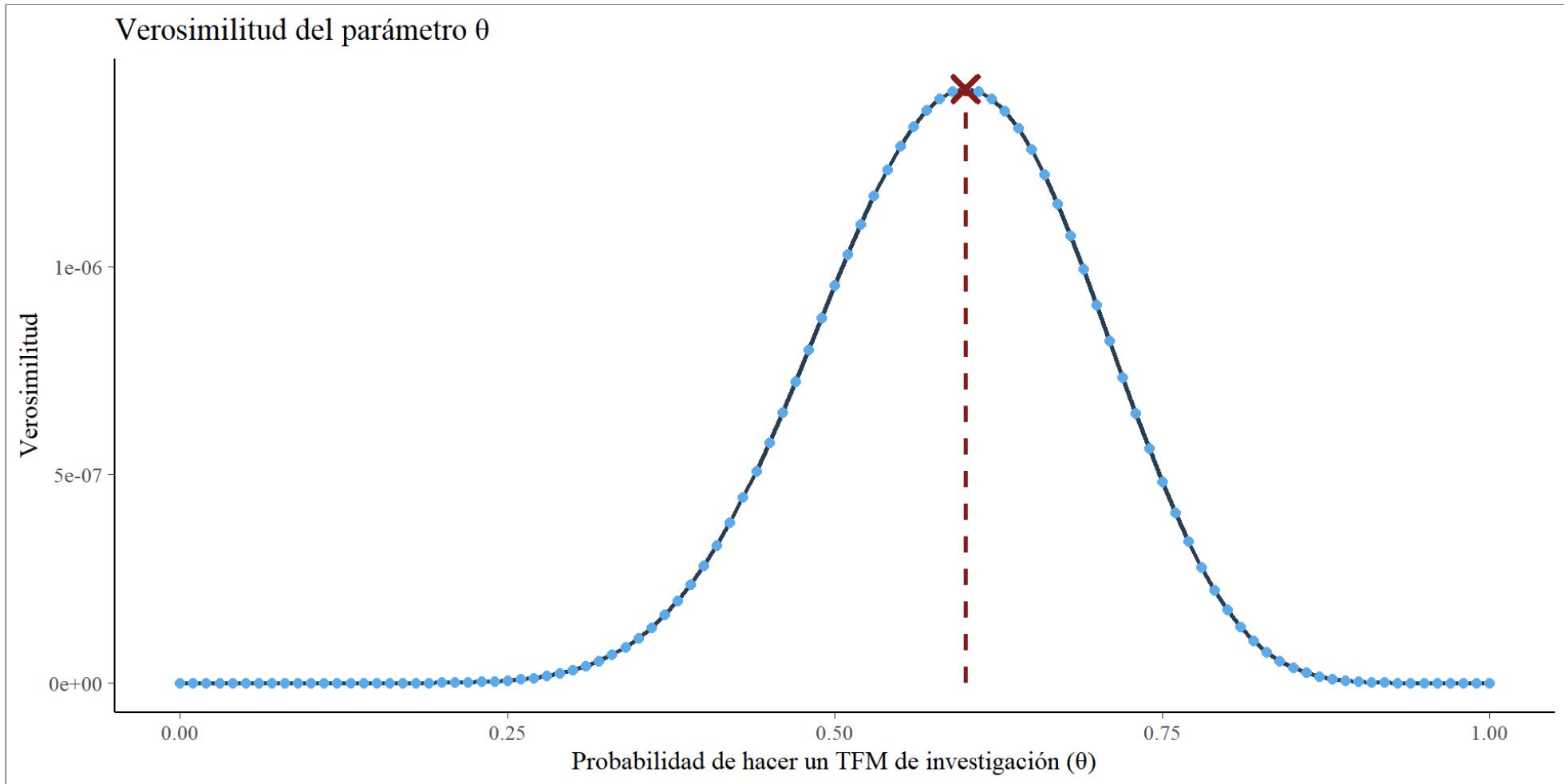
- ¿Cuál es el valor de  $\theta$  que maximiza la verosimilitud?

## Estimando $\theta$ : máxima verosimilitud



- ¿Cuál es el valor de  $\theta$  que maximiza la verosimilitud?

## Estimando $\theta$ : máxima verosimilitud



- ¿Cuál es el valor de  $\theta$  que maximiza la verosimilitud?

## Estimando $\theta$ : máxima verosimilitud

- ¿Cuál es el valor de  $\theta$  que maximiza la verosimilitud?

```
1 # Valor de theta que maximiza la verosimilitud  
2 theta[which.max(likelihood)]
```

[1] 0.6

- Este valor de  $\hat{\theta} = 0.6$  coincide con el que hemos obtenido antes utilizando el *método de los momentos*.

## Estimando $\theta$ : estimación bayesiana

- La **estimación bayesiana** es, ante todo, estimación basada en la verosimilitud, y su corazón es el **Teorema de Bayes**:

$$\Pr(\theta | D) \propto \Pr(D | \theta) \times \Pr(\theta)$$

- **Verosimilitud:**  $\Pr(D | \theta)$
- **Probabilidad a priori:**  $\Pr(\theta)$
- **Probabilidad posterior:**  $\Pr(\theta | D)$

### Note

El símbolo  $\propto$  significa “*proporcional a*”, y no “*igual a*”. Esto se debe a que la distribución posterior es una distribución de probabilidad, es decir, la suma de sus valores es 1. Sin embargo, esto no se cumple con el producto de la verosimilitud y la distribución a priori.

## Estimando $\theta$ : estimación bayesiana

- Alternativamente, el **Teorema de Bayes** también se representa como

$$\Pr(\theta | D) = \frac{\Pr(D | \theta) \times \Pr(\theta)}{\Pr(D)}$$

- $\Pr(D)$  recibe el nombre de *verosimilitud marginal*, y no es más que el sumatorio de todos los valores de  $\Pr(D | \theta) \times \Pr(\theta)$ .

```
1 # Ejemplo  
2 likelihood <- c(.2, .8, .8); prior <- c(.2, .5, .5)  
3 # Sin dividir entre su sumatorio  
4 likelihood * prior
```

```
[1] 0.04 0.40 0.40
```

## Estimando $\theta$ : estimación bayesiana

- Alternativamente, el **Teorema de Bayes** también se representa como

$$\Pr(\theta | D) = \frac{\Pr(D | \theta) \times \Pr(\theta)}{\Pr(D)}$$

- $\Pr(D)$  recibe el nombre de *verosimilitud marginal*, y no es más que el sumatorio de todos los valores de  $\Pr(D | \theta) \times \Pr(\theta)$ .

```
1 # Ejemplo
2 likelihood <- c(.2, .8, .8); prior <- c(.2, .5, .5)
3 # Dividiendo entre su sumatorio
4 likelihood * prior/sum(likelihood * prior)
```

```
[1] 0.04761905 0.47619048 0.47619048
```

## Estimando $\theta$ : estimación bayesiana

- Alternativamente, el **Teorema de Bayes** también se representa como

$$\Pr(\theta | D) = \frac{\Pr(D | \theta) \times \Pr(\theta)}{\Pr(D)}$$

- $\Pr(D)$  recibe el nombre de *verosimilitud marginal*, y no es más que el sumatorio de todos los valores de  $\Pr(D | \theta) \times \Pr(\theta)$ .
- Esto garantiza que la **distribución posterior**,  $\Pr(\theta | D)$ , sea una **distribución de probabilidad válida**.
- La *verosimilitud marginal* es **imprescindible** en la comparación de modelos a través del **factor de Bayes**.

## Estimando $\theta$ : estimación bayesiana

- Carmen y Javier han recuperado la selección de TFM de los cursos académicos.

---

### Selección de TFM de prácticas e investigación

---

Curso	Investigación	Prácticas	$\hat{\theta}$
2018/2019	10	14	0.417
2019/2020	5	15	0.250
2020/2021	17	13	0.567
2021/2022	13	15	0.464
2022/2023	8	18	0.308

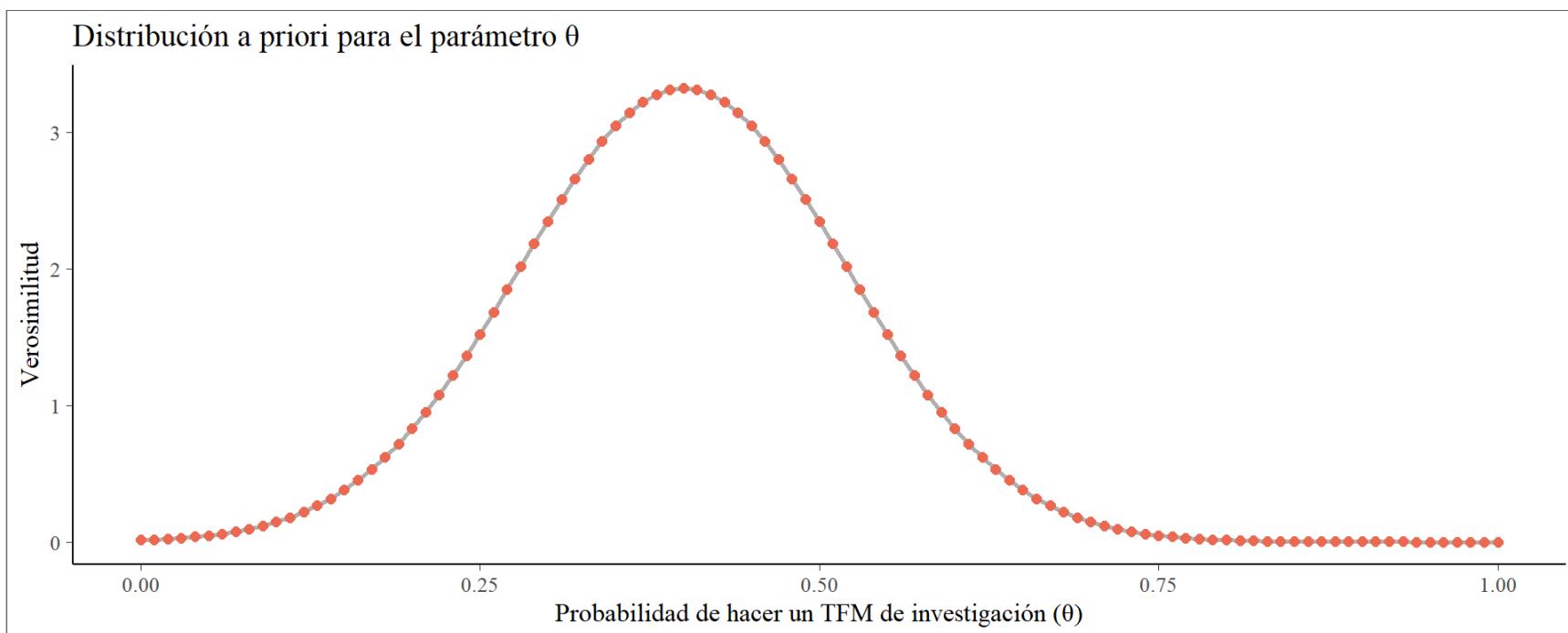
---

- $\hat{\theta}$  es nuestro **conocimiento previo** de la selección de TFM.

## Estimando $\theta$ : estimación bayesiana

- La media de  $\hat{\theta}$  es 0.40, y su desviación típica es 0.12.
- Podemos utilizar esta información con la **distribución a priori**.

$$\theta \sim \mathcal{N} (\mu_\theta = 0.4, \sigma_\theta = 0.12)$$



## Estimando $\theta$ : estimación bayesiana

- El siguiente código de R muestra cómo surge toda la magia que crea la famosa distribución posterior.
- En primer lugar, calculamos (otra vez) **la verosimilitud de los datos**

```
1 # Valores de theta de 0 a 1
2 theta <- seq(0, 1, 0.01)
3 # Vector vacío para guardar la verosimilitud con cada theta
4 likelihood <- vector(length = length(theta))
5 # Verosimilitud para cada theta
6 for(i in 1:length(theta)){
7   likelihood[i] <- prod(dbinom(TFM, size = 1, prob = theta
8 }
```

## Estimando $\theta$ : estimación bayesiana

- Aplicamos el **Teorema de Bayes**:

$$\Pr(\theta | D) = \frac{\Pr(D | \theta) \times \Pr(\theta)}{\Pr(D)} = \frac{\text{Verosimilitud} \times \text{Prior}}{\text{Verosimilitud marginal}}$$

```
1 # Distribución a priori de theta  
2 prior <- dnorm(theta, mean = 0.4, sd = 0.12)  
3 # Distribución posterior (proporcional)  
4 posterior_prop <- likelihood * prior  
5 # Verosimilitud marginal  
6 marginal <- sum(likelihood * prior)  
7 # Distribución posterior escalada a [0,1]  
8 posterior <- posterior_prop/marginal  
9 plot(theta, posterior) # Gráfico de densidad posterior
```

## Estimando $\theta$ : estimación bayesiana

- Aplicamos el **Teorema de Bayes**:

$$\Pr(\theta | D) = \frac{\Pr(D | \theta) \times \text{Pr}(\theta)}{\Pr(D)} = \frac{\text{Verosimilitud} \times \text{Prior}}{\text{Verosimilitud marginal}}$$

```
1 # Distribución a priori de theta  
2 prior <- dnorm(theta, mean = 0.4, sd = 0.12)  
3 # Distribución posterior (proporcional)  
4 posterior_prop <- likelihood * prior  
5 # Verosimilitud marginal  
6 marginal <- sum(likelihood * prior)  
7 # Distribución posterior escalada a [0,1]  
8 posterior <- posterior_prop/marginal  
9 plot(theta, posterior) # Gráfico de densidad posterior
```

## Estimando $\theta$ : estimación bayesiana

- Aplicamos el **Teorema de Bayes**:

$$\Pr(\theta | D) = \frac{\Pr(D | \theta) \times \Pr(\theta)}{\Pr(D)} = \frac{\text{Verosimilitud} \times \text{Prior}}{\text{Verosimilitud marginal}}$$

```
1 # Distribución a priori de theta  
2 prior <- dnorm(theta, mean = 0.4, sd = 0.12)  
3 # Distribución posterior (proporcional)  
4 posterior_prop <- likelihood * prior  
5 # Verosimilitud marginal  
6 marginal <- sum(likelihood * prior)  
7 # Distribución posterior escalada a [0,1]  
8 posterior <- posterior_prop/marginal  
9 plot(theta, posterior) # Gráfico de densidad posterior
```

## Estimando $\theta$ : estimación bayesiana

- Aplicamos el **Teorema de Bayes**:

$$\Pr(\theta | D) = \frac{\Pr(D | \theta) \times \Pr(\theta)}{\Pr(D)} = \frac{\text{Verosimilitud} \times \text{Prior}}{\text{Verosimilitud marginal}}$$

```
1 # Distribución a priori de theta  
2 prior <- dnorm(theta, mean = 0.4, sd = 0.12)  
3 # Distribución posterior (proporcional)  
4 posterior_prop <- likelihood * prior  
5 # Verosimilitud marginal  
6 marginal <- sum(likelihood * prior)  
7 # Distribución posterior escalada a [0,1]  
8 posterior <- posterior_prop/marginal  
9 plot(theta, posterior) # Gráfico de densidad posterior
```

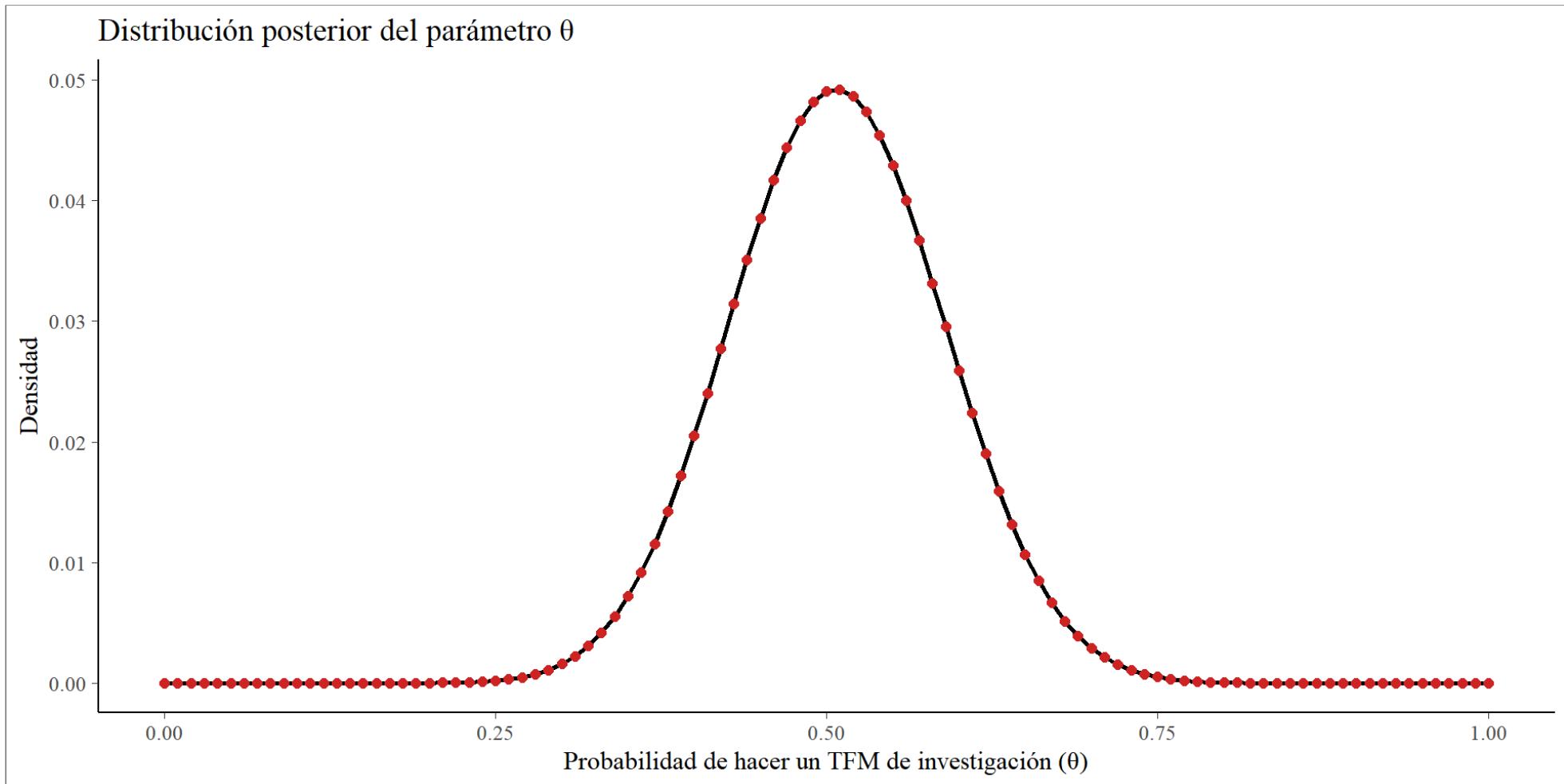
## Estimando $\theta$ : estimación bayesiana

- Aplicamos el **Teorema de Bayes**:

$$\Pr(\theta | D) = \frac{\Pr(D | \theta) \times \Pr(\theta)}{\Pr(D)} = \frac{\text{Verosimilitud} \times \text{Prior}}{\text{Verosimilitud marginal}}$$

```
1 # Distribución a priori de theta  
2 prior <- dnorm(theta, mean = 0.4, sd = 0.12)  
3 # Distribución posterior (proporcional)  
4 posterior_prop <- likelihood * prior  
5 # Verosimilitud marginal  
6 marginal <- sum(likelihood * prior)  
7 # Distribución posterior escalada a [0,1]  
8 posterior <- posterior_prop/marginal  
9 plot(theta, posterior) # Gráfico de densidad posterior
```

## Estimando $\theta$ : estimación bayesiana



## Estimando $\theta$ : estimación bayesiana

- La **moda posterior** suele utilizarse como estadísticos de resumen de la **distribución posterior**.

```
1 # Moda de la distribución posterior  
2 theta[which.max(posterior)]
```

[1] 0.51

- Es distinto del valor que obtendríamos por **máxima verosimilitud**

```
1 # Estimación máximo-verosímil  
2 theta[which.max(likelihood)]
```

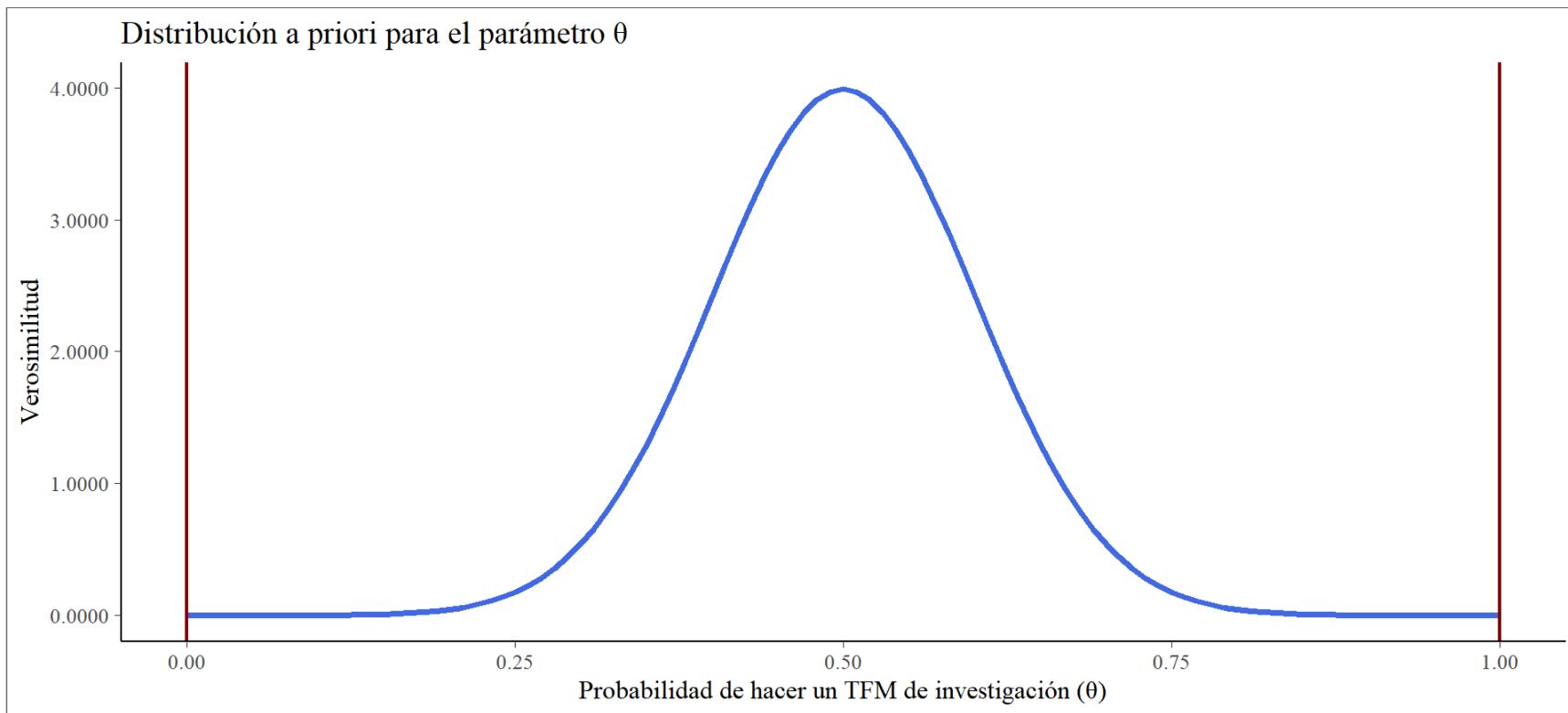
[1] 0.6

- La **distribución a priori** es responsable de esta diferencia

## Estimando $\theta$ : estimación bayesiana

- ¿Qué pasa si utilizamos una distribución a priori menos *informativa*?

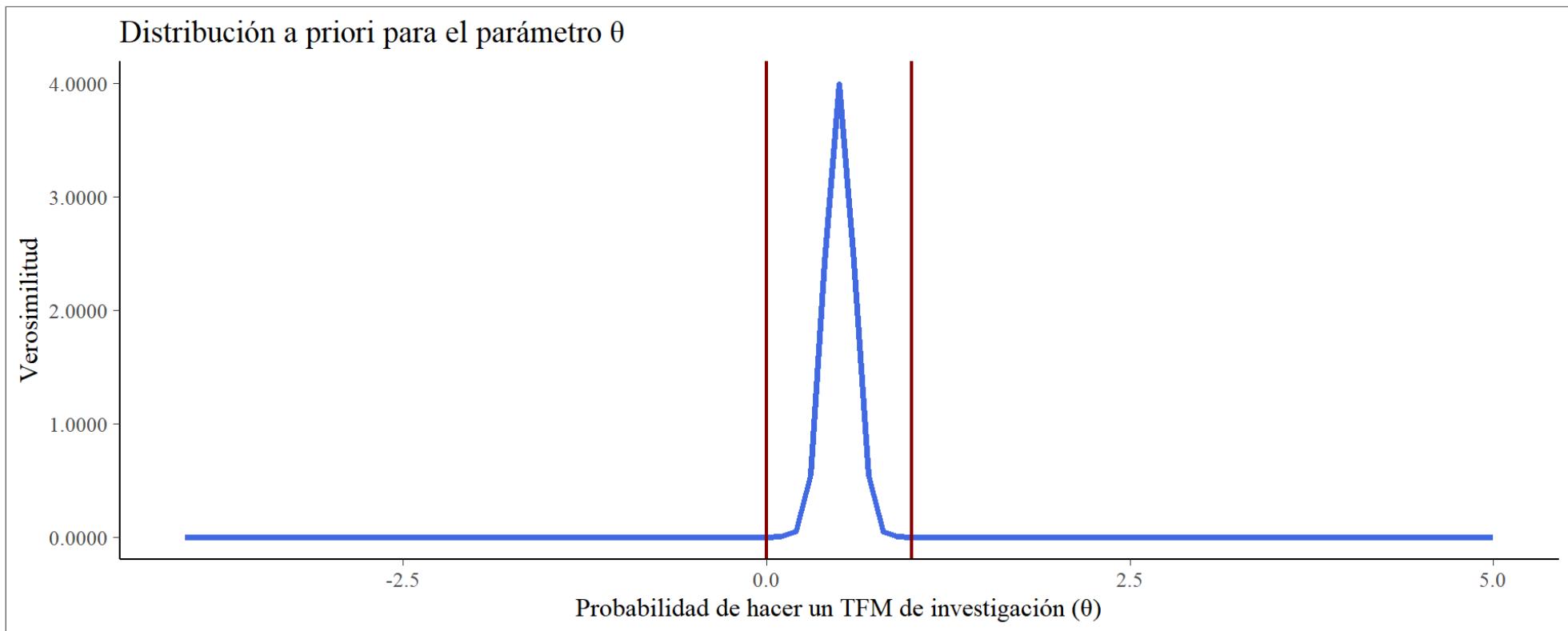
$$\theta \sim \mathcal{N}(\mu_\theta = 0.5, \sigma_\theta = 0.1)$$



## Estimando $\theta$ : estimación bayesiana

- ¿Qué pasa si utilizamos una distribución a priori menos *informativa*?

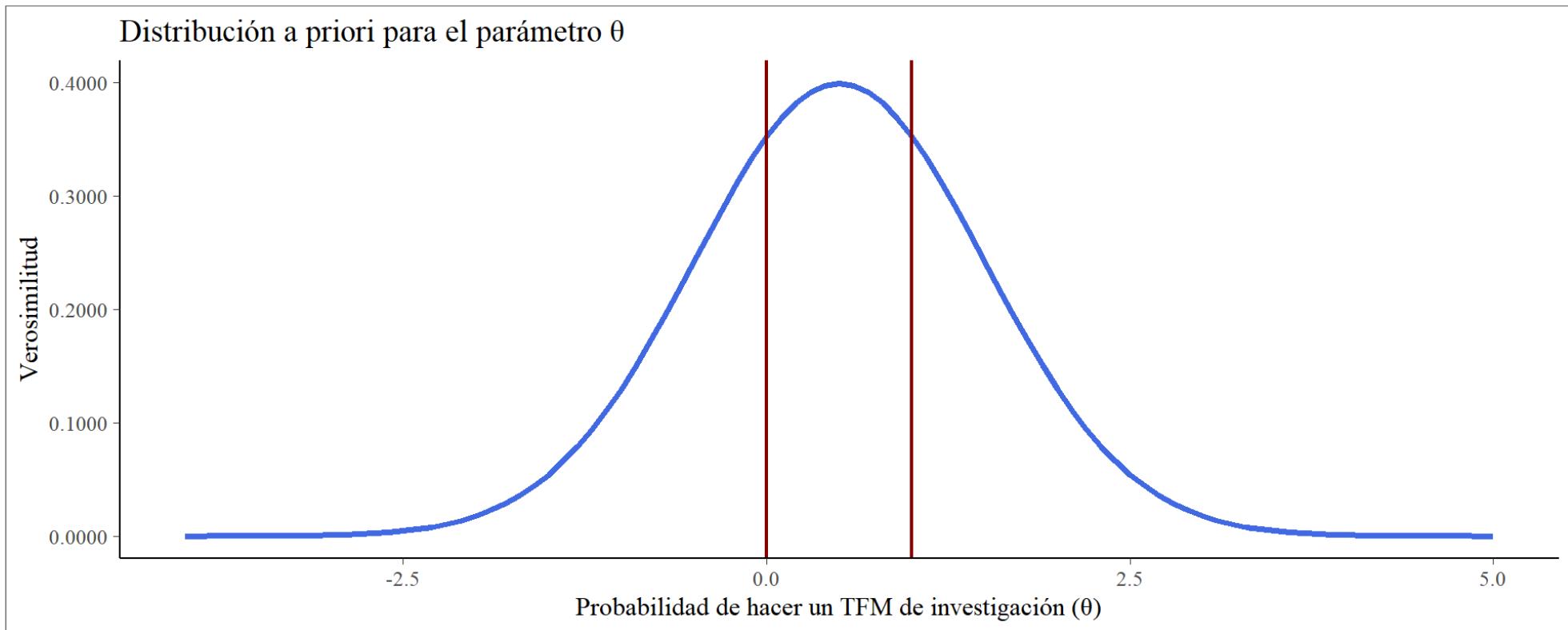
$$\theta \sim \mathcal{N}(\mu_\theta = 0.5, \sigma_\theta = 0.1)$$



## Estimando $\theta$ : estimación bayesiana

- ¿Qué pasa si utilizamos una distribución a priori menos *informativa*?

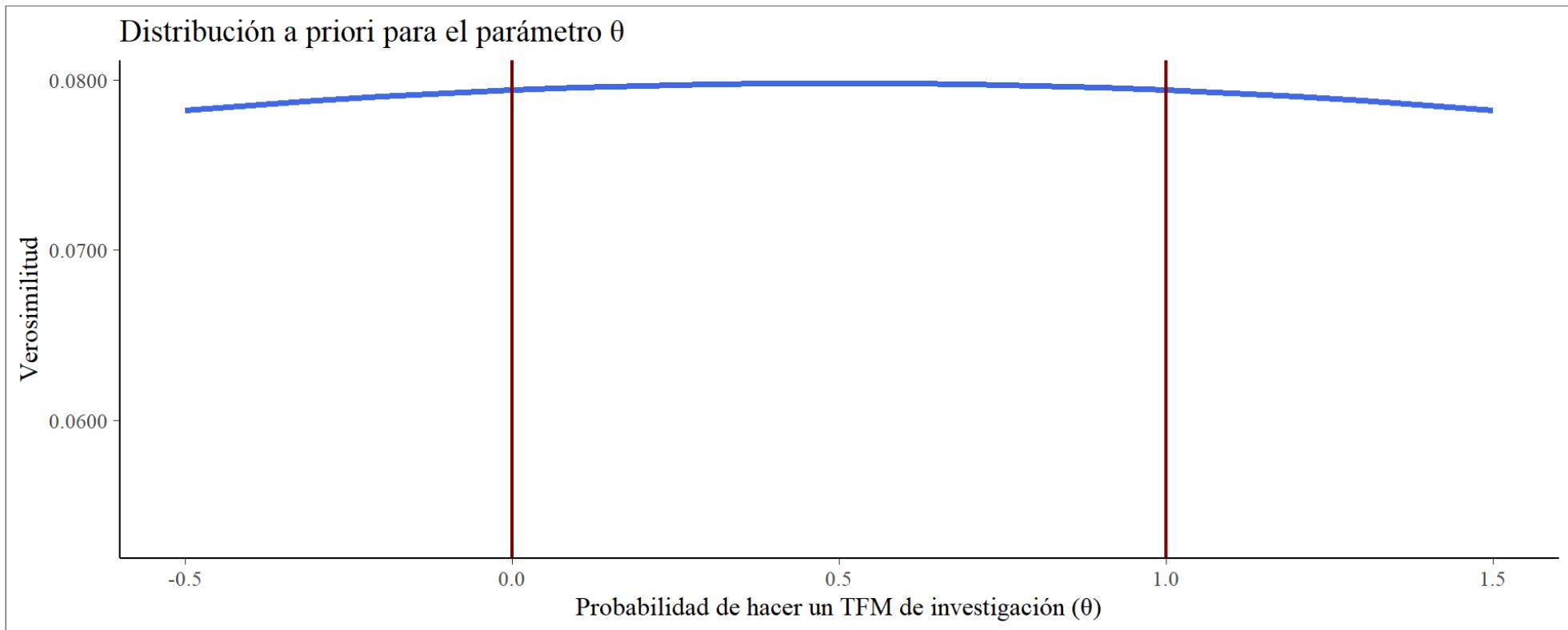
$$\theta \sim \mathcal{N}(\mu_\theta = 0.5, \sigma_\theta = 1)$$



## Estimando $\theta$ : estimación bayesiana

- ¿Qué pasa si utilizamos una distribución a priori menos *informativa*?

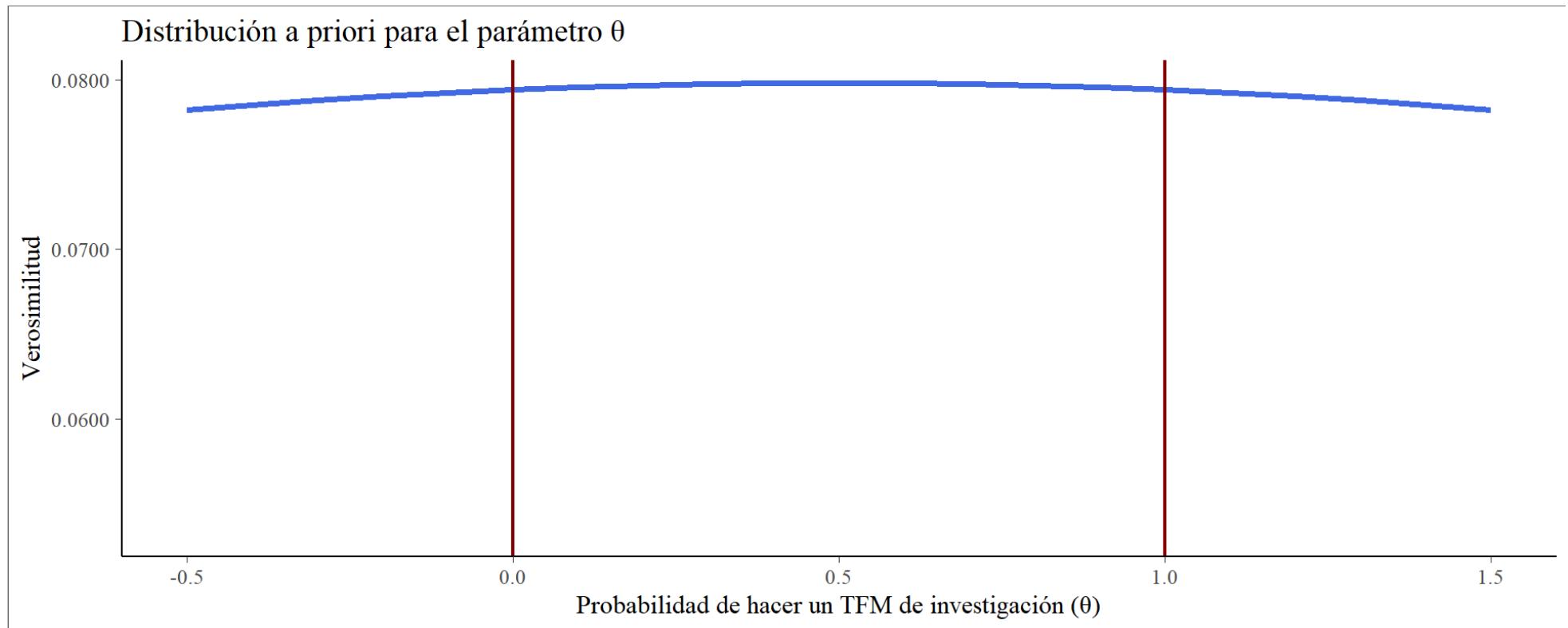
$$\theta \sim \mathcal{N}(\mu_\theta = 0.5, \sigma_\theta = 5)$$



## Estimando $\theta$ : estimación bayesiana

- ¿Qué pasa si utilizamos una distribución a priori menos *informativa*?

$$\theta \sim \mathcal{N}(\mu_\theta = 0.5, \sigma_\theta = 5)$$



## Estimando $\theta$ : estimación bayesiana

- ¿Qué pasa si utilizamos una distribución a priori menos *informativa*?

$$\theta \sim \mathcal{N}(\mu_\theta = 0.5, \sigma_\theta = 5)$$

- La probabilidad a priori para cualquier valor de  $\theta$  es **prácticamente la misma**.

```
1 # Prior no informativa sobre theta  
2 theta <- seq(0, 1, .1)  
3 round(dnorm(theta, 0.5, 5), 4)
```

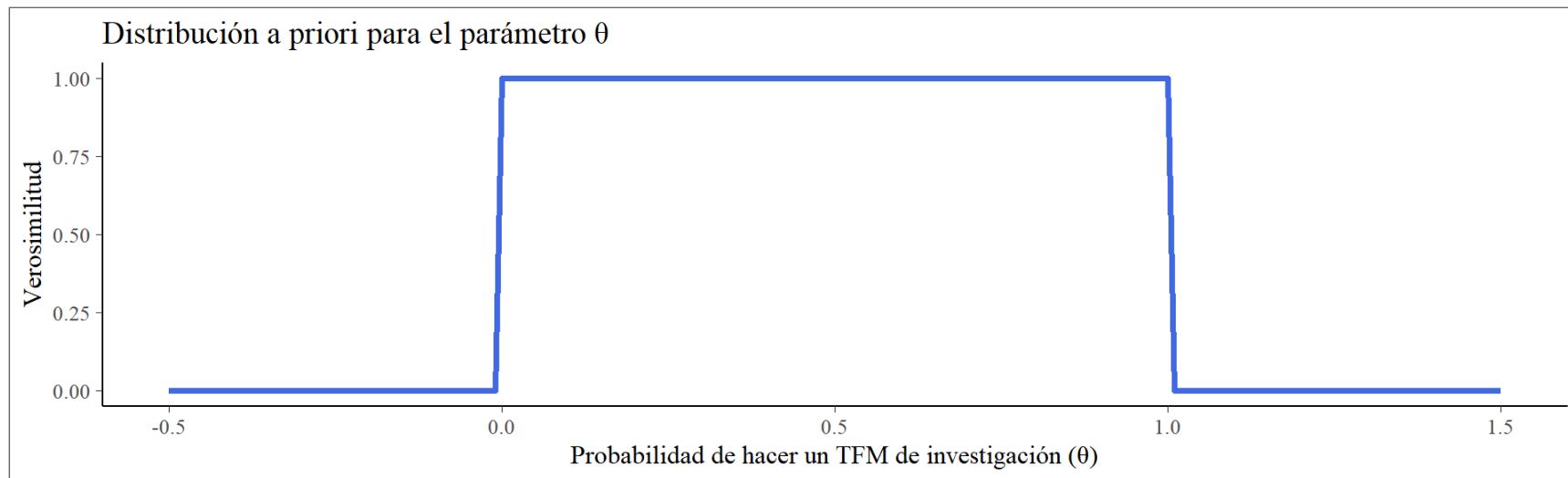
```
[1] 0.0794 0.0795 0.0796 0.0797 0.0798 0.0798 0.0798 0.0797  
0.0796 0.0795  
[11] 0.0794
```

## Estimando $\theta$ : estimación bayesiana

- Las **distribuciones a priori planas**, como la **distribución uniforme**, son los casos más extremos de distribuciones a priori no informativas.

$$\theta \sim \text{Uniform}(a = 0, b = 1)$$

- Se asigna una **probabilidad constante** a todo valor en el rango  $[a, b]$



## Estimando $\theta$ : estimación bayesiana

```
1 theta <- seq(0, 1, .1) # Valores de theta  
2 dunif(theta, min = 0, max = 1) # Prior uniforme
```

```
[1] 1 1 1 1 1 1 1 1 1 1 1 1
```

- ¿Qué implicaciones tiene esto para el **Teorema de Bayes**?

$$\Pr(\theta | D) \propto \Pr(D | \theta) \times \Pr(\theta)$$

## Estimando $\theta$ : estimación bayesiana

```
1 theta <- seq(0, 1, .1) # Valores de theta  
2 dunif(theta, min = 0, max = 1) # Prior uniforme
```

```
[1] 1 1 1 1 1 1 1 1 1 1 1 1
```

- ¿Qué implicaciones tiene esto para el **Teorema de Bayes**?

$$\Pr(\theta | D) \propto \Pr(D | \theta) \times 1$$

## Estimando $\theta$ : estimación bayesiana

```
1 theta <- seq(0, 1, .1) # Valores de theta  
2 dunif(theta, min = 0, max = 1) # Prior uniforme
```

```
[1] 1 1 1 1 1 1 1 1 1 1 1
```

- ¿Qué implicaciones tiene esto para el **Teorema de Bayes**?

$$\Pr(\theta | D) = \Pr(D | \theta)$$

- La **distribución posterior** es **lo mismo** que la verosimilitud.

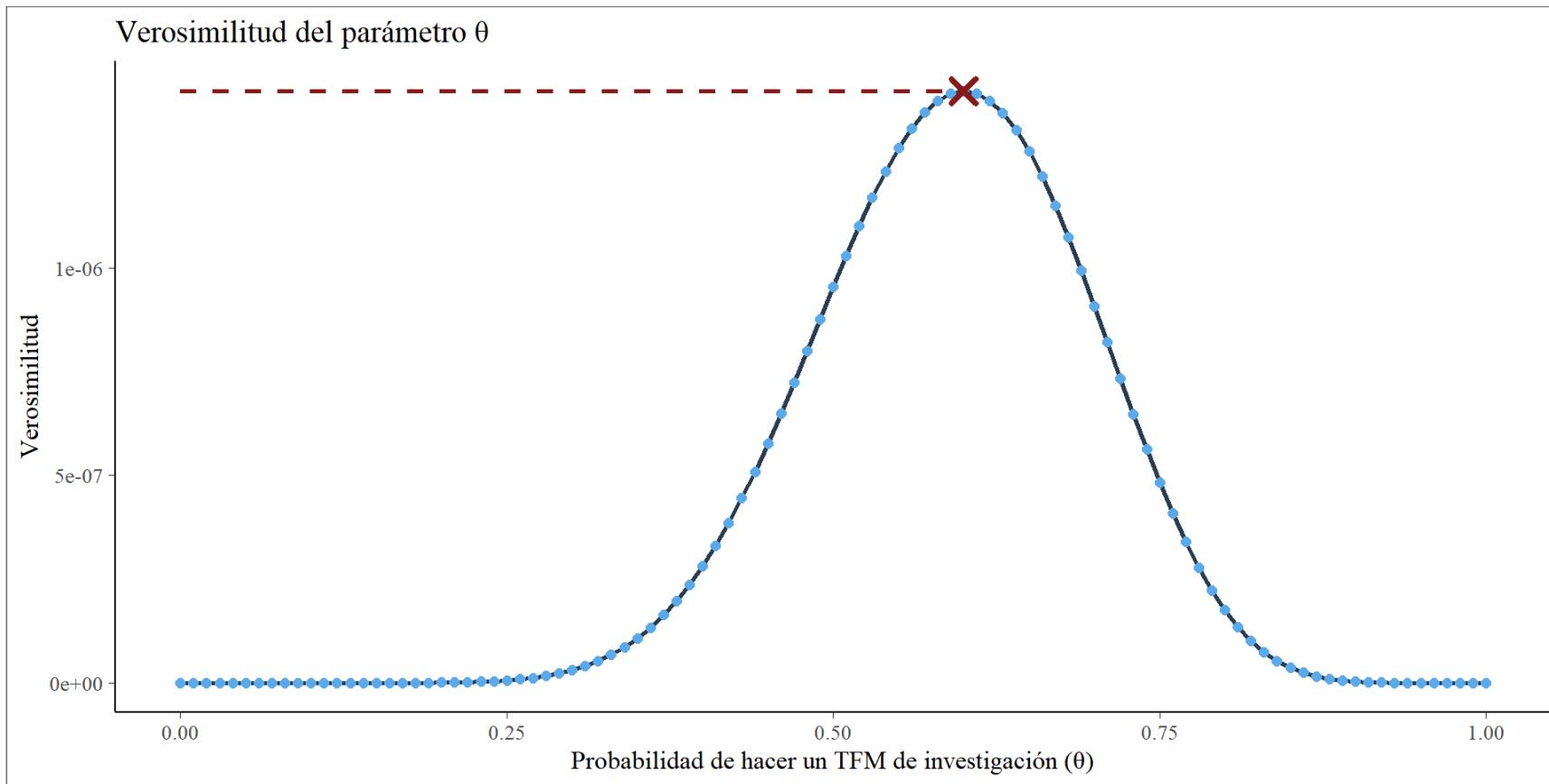
### Universo frecuentista, multiverso bayesiano

La **estadística frecuentista** sólo es **un caso particular** dentro de la **estadística bayesiana**.

# **Máxima verosimilitud Vs. Cadenas de Markov**

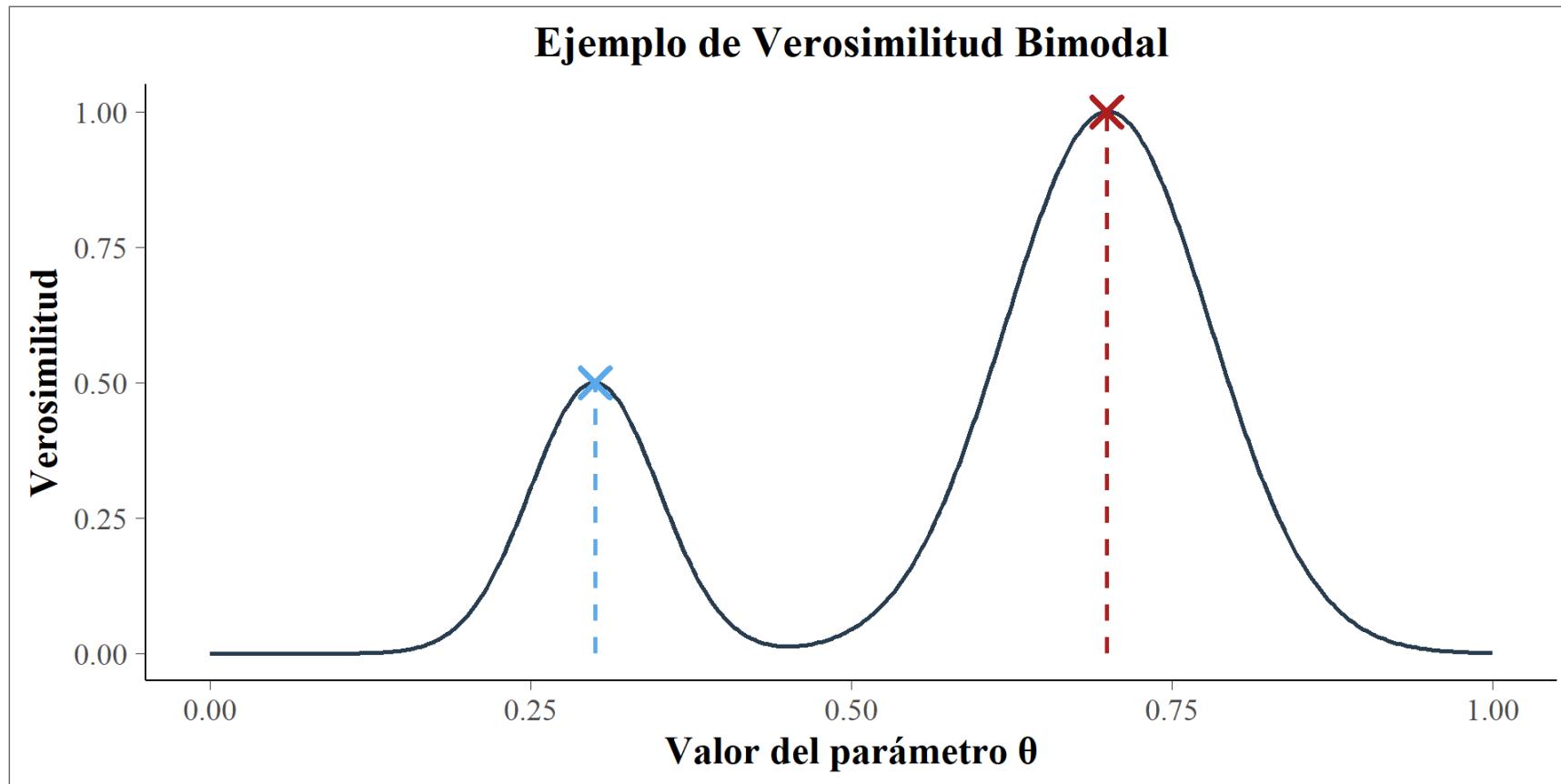
# Máxima verosimilitud Vs. Cadenas de Markov

- Máxima verosimilitud utiliza **algoritmos de optimización numérica** para encontrar **un valor máximo**.



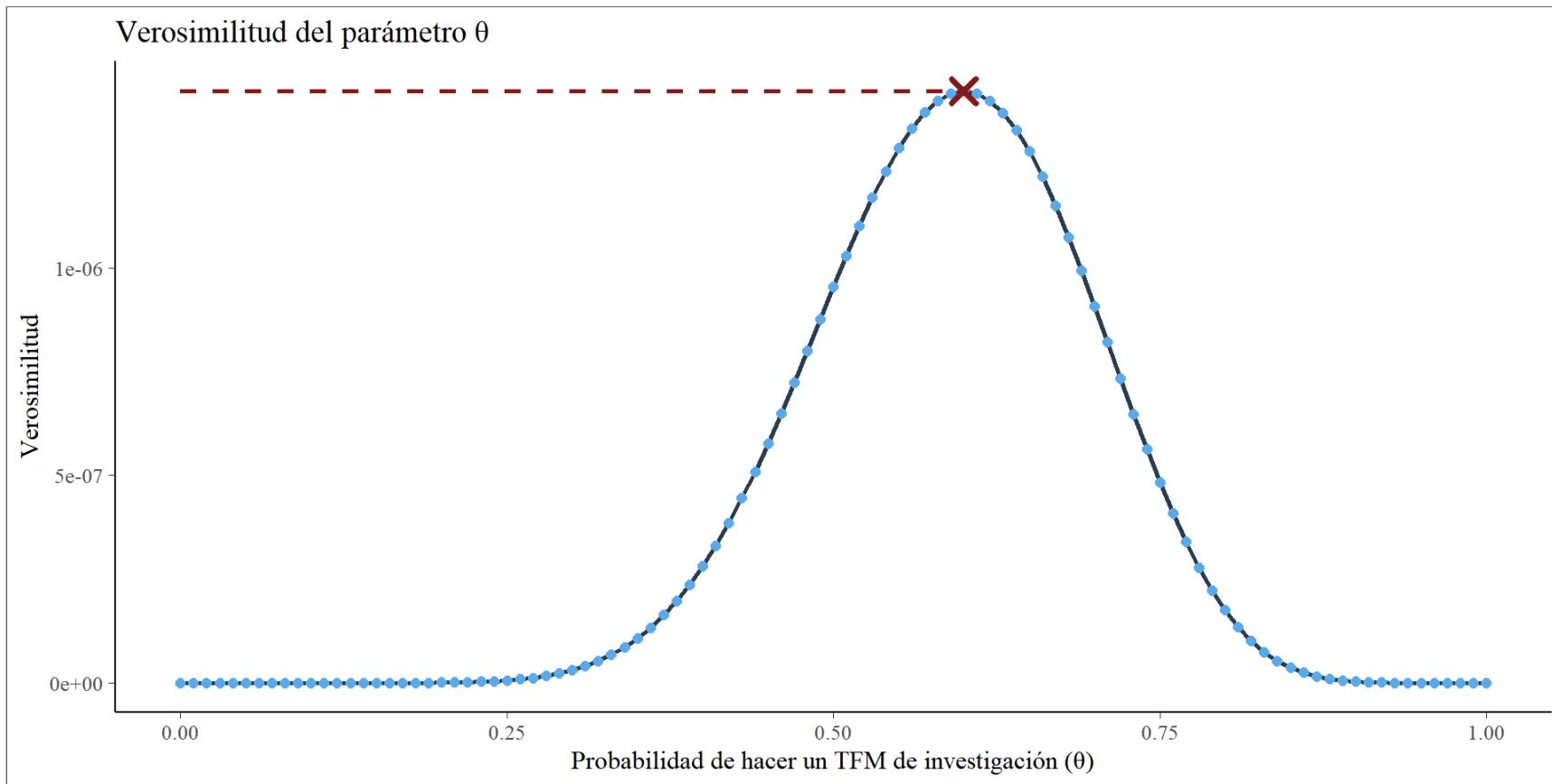
## Máxima verosimilitud Vs. Cadenas de Markov

- En ocasiones, puede encontrar un máximo local, pero no un *máximo absoluto*.



## Máxima verosimilitud Vs. Cadenas de Markov

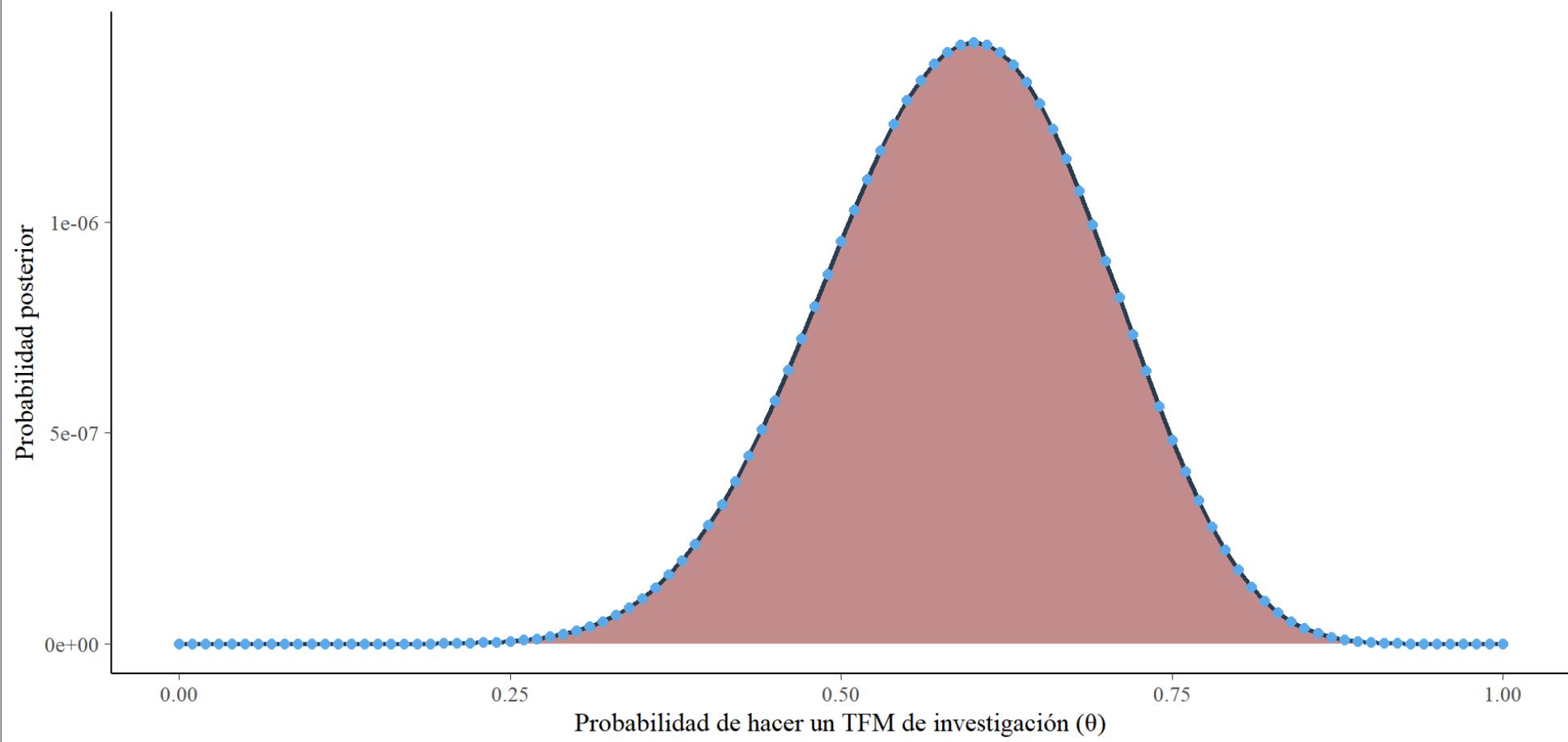
- Las Cadenas de Markov utilizan métodos de simulación Monte Carlo para recuperar **toda una distribución, no un único valor**



## **Máxima verosimilitud Vs. Cadenas de Markov**

- Las Cadenas de Markov utilizan métodos de simulación Monte Carlo para recuperar **toda una distribución, no un único valor**

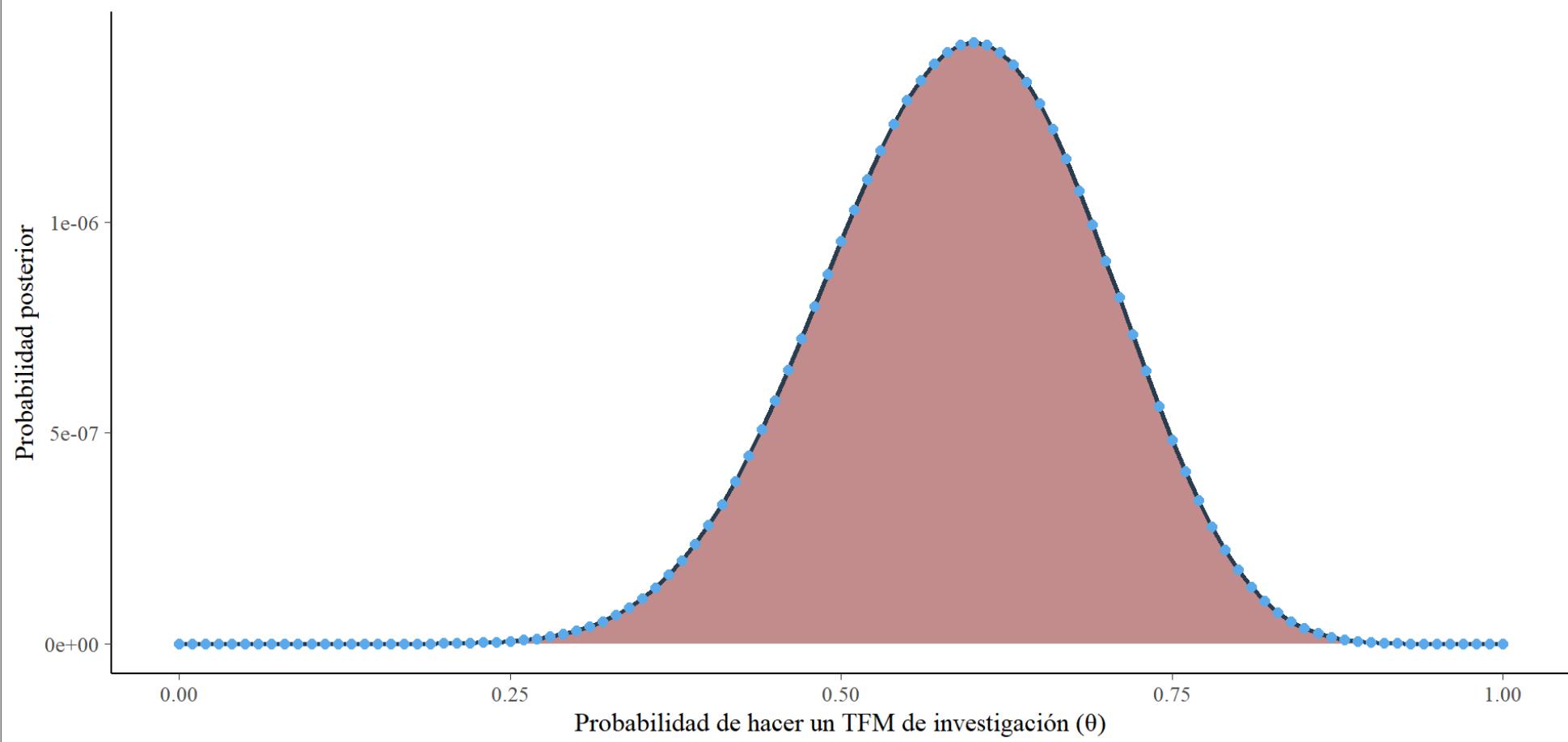
## Probabilidad posterior del parámetro $\theta$



## Máxima verosimilitud Vs. Cadenas de Markov

- Las Cadenas de Markov utilizan métodos de simulación Monte Carlo para recuperar **toda una distribución, no un único valor**
- Se realizan  $n$  **extracciones** de la distribución posterior.

## Probabilidad posterior del parámetro $\theta$



## Máxima verosimilitud Vs. Cadenas de Markov

- Las Cadenas de Markov utilizan métodos de simulación Monte Carlo para recuperar **toda una distribución, no un único valor**
- Se realizan  $n$  **extracciones** de la distribución posterior.
- Los **valores más probables** aparecerán con **mayor frecuencia**.
- Los algoritmos que se suelen aplicar **difieren en eficiencia y flexibilidad**.
  - Sampleador de Gibbs (implementado en **JAGS**).
  - Monte Carlo Hamiltoniano (implementado en **Stan**).
- Visualmente podemos ver cómo **difieren en su forma de extraer muestras**.
  - [Visualizador de algoritmos MCMC](#)

# **Introducción a la Psicometría Bayesiana**

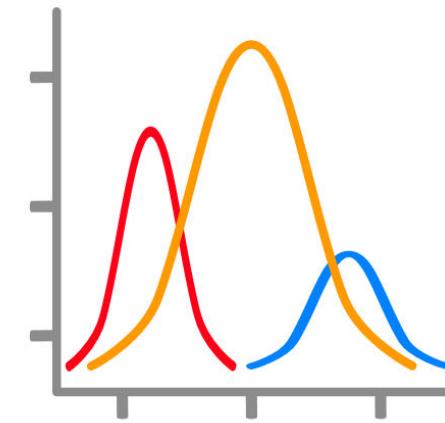
**¿Qué es modelar?**



Variable  
dependiente

~

“se distribuye  
como”

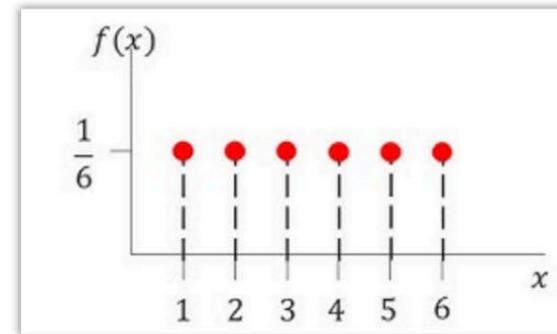


Distribución de  
probabilidad

**¿Qué es modelar?**



~



Variable  
dependiente

“se distribuye  
como”

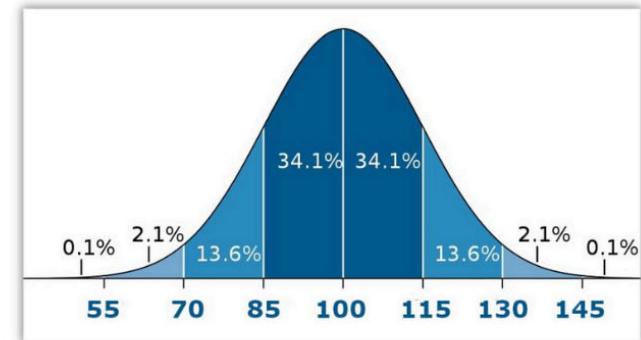
Distribución de  
probabilidad

**¿Qué es modelar?**



Variable  
dependiente

~



“se distribuye  
como”

Distribución de  
probabilidad

## ¿Qué es modelar?

- Es habitual expresar un **modelo de regresión lineal** como

$$Y_i = \beta_0 + \beta_1 \cdot X_i + \varepsilon_i, \quad \text{donde} \quad \varepsilon_i \sim \mathcal{N}(\mu = 0, \sigma = \sigma_\varepsilon)$$

- Esto es **idéntico** a

$$Y_i \sim \mathcal{N}(\mu = \beta_0 + \beta_1 \cdot X_i, \sigma = \sigma_\varepsilon)$$

- Por tanto, el **modelo estadístico** cuenta con:

- Una **variable dependiente** y una **distribución de probabilidad**,
- los **parámetros** de la **distribución de probabilidad** seleccionada.
- los **pronósticos** de la ecuación lineal (i.e., la parte **determinista**).

## El modelo psicométrico del factor común

- La psicometría se asienta sobre el modelo de regresión lineal.

$$Y_i = \beta_0 + \beta_1 \cdot X_i + \varepsilon_i, \quad \text{donde} \quad \varepsilon_i \sim \mathcal{N}(\mu = 0, \sigma = \sigma_\varepsilon)$$

- Sin embargo, en psicometría:
  - se asume que la variable  $X$  es **latente en lugar de observable**. Se suele representar con la letra griega  $\eta$ .
  - suelen utilizarse las letras griegas  $\nu$ ,  $\lambda$  y  $\theta$  en lugar de  $\beta_0$ ,  $\beta_1$  y  $\sigma_\varepsilon$ .
- Por eso, el mismo modelo se representa como

$$Y_i = \nu + \lambda \cdot \eta_i + \varepsilon_i, \quad \text{donde} \quad \varepsilon_i \sim \mathcal{N}(\mu = 0, \sigma = \theta)$$

## El modelo psicométrico del factor común

$$Y_i = \nu + \lambda \cdot \eta_i + \varepsilon_i, \quad \text{donde } \varepsilon_i \sim \mathcal{N}(\mu = 0, \sigma = \theta)$$

- $Y_i$  es **una medida imperfecta** de una variable no observable,  $\eta_i$ .
- $\varepsilon_i$  recibe el nombre de **unicidad** (i.e., varianza no explicada).
  - Una parte de la **unicidad** es el **error de medida**
- Es habitual contar con **varias medidas de la variable no observable**.

$$Y_{i1} = \nu_1 + \lambda_1 \cdot \eta_i + \varepsilon_{i1}, \quad \text{donde } \varepsilon_{i1} \sim \mathcal{N}(\mu = 0, \sigma = \theta_1)$$

## El modelo psicométrico del factor común

$$Y_i = \nu + \lambda \cdot \eta_i + \varepsilon_i, \quad \text{donde } \varepsilon_i \sim \mathcal{N}(\mu = 0, \sigma = \theta)$$

- $Y_i$  es **una medida imperfecta** de una variable no observable,  $\eta_i$ .
- $\varepsilon_i$  recibe el nombre de **unicidad** (i.e., varianza no explicada).  
→ Una parte de la **unicidad** es el **error de medida**
- Es habitual contar con **varias medidas de la variable no observable**.

$$Y_{i1} = \nu_1 + \lambda_1 \cdot \eta_i + \varepsilon_{i1}, \quad \text{donde } \varepsilon_{i1} \sim \mathcal{N}(\mu = 0, \sigma = \theta_1)$$

$$Y_{i2} = \nu_2 + \lambda_2 \cdot \eta_i + \varepsilon_{i2}, \quad \text{donde } \varepsilon_{i2} \sim \mathcal{N}(\mu = 0, \sigma = \theta_2)$$

## El modelo psicométrico del factor común

$$Y_i = \nu + \lambda \cdot \eta_i + \varepsilon_i, \quad \text{donde } \varepsilon_i \sim \mathcal{N}(\mu = 0, \sigma = \theta)$$

- $Y_i$  es **una medida imperfecta** de una variable no observable,  $\eta_i$ .
- $\varepsilon_i$  recibe el nombre de **unicidad** (i.e., varianza no explicada).  
→ Una parte de la **unicidad** es el **error de medida**
- Es habitual contar con **varias medidas de la variable no observable**.

$$Y_{i1} = \nu_1 + \lambda_1 \cdot \eta_i + \varepsilon_{i1}, \quad \text{donde } \varepsilon_{i1} \sim \mathcal{N}(\mu = 0, \sigma = \theta_1)$$

$$Y_{i2} = \nu_2 + \lambda_2 \cdot \eta_i + \varepsilon_{i2}, \quad \text{donde } \varepsilon_{i2} \sim \mathcal{N}(\mu = 0, \sigma = \theta_2)$$

$$Y_{i3} = \nu_3 + \lambda_3 \cdot \eta_i + \varepsilon_{i3}, \quad \text{donde } \varepsilon_{i3} \sim \mathcal{N}(\mu = 0, \sigma = \theta_3)$$

## El modelo psicométrico del factor común

$$Y_i = \nu + \lambda \cdot \eta_i + \varepsilon_i, \quad \text{donde } \varepsilon_i \sim \mathcal{N}(\mu = 0, \sigma = \theta)$$

- $Y_i$  es **una medida imperfecta** de una variable no observable,  $\eta_i$ .
- $\varepsilon_i$  recibe el nombre de **unicidad** (i.e., varianza no explicada).

→ Una parte de la **unicidad** es el **error de medida**

- Es habitual contar con **varias medidas de la variable no observable**.

$$Y_{i1} = \nu_1 + \lambda_1 \cdot \eta_i + \varepsilon_{i1}, \quad \text{donde } \varepsilon_{i1} \sim \mathcal{N}(\mu = 0, \sigma = \theta_1)$$

$$Y_{i2} = \nu_2 + \lambda_2 \cdot \eta_i + \varepsilon_{i2}, \quad \text{donde } \varepsilon_{i2} \sim \mathcal{N}(\mu = 0, \sigma = \theta_2)$$

$$Y_{i3} = \nu_3 + \lambda_3 \cdot \eta_i + \varepsilon_{i3}, \quad \text{donde } \varepsilon_{i3} \sim \mathcal{N}(\mu = 0, \sigma = \theta_3)$$

## El modelo psicométrico del factor común

$$Y_i = \nu + \lambda \cdot \eta_i + \varepsilon_i, \quad \text{donde } \varepsilon_i \sim \mathcal{N}(\mu = 0, \sigma = \theta)$$

- $Y_i$  es **una medida imperfecta** de una variable no observable,  $\eta_i$ .
- $\varepsilon_i$  recibe el nombre de **unicidad** (i.e., varianza no explicada).

→ Una parte de la **unicidad** es el **error de medida**

- Es habitual contar con **varias medidas de la variable no observable**.
- Este mismo modelo para  $I$  personas y  $J$  variables observables es

$$Y_{ij} = \nu_j + \lambda_j \cdot \eta_i + \varepsilon_{ij}, \quad \text{donde } \varepsilon_{ij} \sim \mathcal{N}(\mu = 0, \sigma = \theta_j)$$

- ¿Y si tenemos **más de una variable latente**?

## El modelo psicométrico del factor común

- ¿Cómo cambia el modelo al ir añadiendo variables latentes?

$$Y_{ij} = \nu_j + \lambda_{j1} \cdot \eta_{i1} + \varepsilon_{ij}$$

## El modelo psicométrico del factor común

- ¿Cómo cambia el modelo al ir añadiendo variables latentes?

$$Y_{ij} = \nu_j + \lambda_{j1} \cdot \eta_{i1} + \varepsilon_{ij}$$

$$Y_{ij} = \nu_j + \lambda_{j1} \cdot \eta_{i1} + \lambda_{j2} \cdot \eta_{i2} + \varepsilon_{ij}$$

## El modelo psicométrico del factor común

- ¿Cómo cambia el modelo al ir añadiendo variables latentes?

$$Y_{ij} = \nu_j + \lambda_{j1} \cdot \eta_{i1} + \varepsilon_{ij}$$

$$Y_{ij} = \nu_j + \lambda_{j1} \cdot \eta_{i1} + \lambda_{j2} \cdot \eta_{i2} + \varepsilon_{ij}$$

$$Y_{ij} = \nu_j + \lambda_{j1} \cdot \eta_{i1} + \lambda_{j2} \cdot \eta_{i2} + \lambda_{j3} \cdot \eta_{i3} + \varepsilon_{ij}$$

# El modelo psicométrico del factor común

- ¿Cómo cambia el modelo al ir añadiendo variables latentes?

$$Y_{ij} = \nu_j + \lambda_{j1} \cdot \eta_{i1} + \varepsilon_{ij}$$

$$Y_{ij} = \nu_j + \lambda_{j1} \cdot \eta_{i1} + \lambda_{j2} \cdot \eta_{i2} + \varepsilon_{ij}$$

$$Y_{ij} = \nu_j + \lambda_{j1} \cdot \eta_{i1} + \lambda_{j2} \cdot \eta_{i2} + \lambda_{j3} \cdot \eta_{i3} + \varepsilon_{ij}$$

$$Y_{ij} = \nu_j + \lambda_{j1} \cdot \eta_{i1} + \lambda_{j2} \cdot \eta_{i2} + \lambda_{j3} \cdot \eta_{i3} + \dots + \varepsilon_{ij}$$

$$Y_{ij} = \nu_j + \sum_{m=1}^M \lambda_{jm} \cdot \eta_{im} + \varepsilon_{ij}$$

- Cuando hay  $M$  variables latentes, el modelo se puede escribir como

## El modelo psicométrico del factor común

- Los psicómetras prefieren la **versión matricial** del modelo.

$$\underbrace{\begin{bmatrix} Y_{i1} \\ Y_{i2} \\ Y_{i3} \end{bmatrix}}_{\mathbf{y}_i} = \underbrace{\begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{bmatrix}}_{\boldsymbol{\nu}} + \underbrace{\begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \\ \lambda_{31} & \lambda_{32} \end{bmatrix}}_{\boldsymbol{\Lambda}} \times \underbrace{\begin{bmatrix} \eta_{i1} \\ \eta_{i2} \end{bmatrix}}_{\boldsymbol{\eta}_i} + \underbrace{\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}}_{\boldsymbol{\varepsilon}_i}$$

- En la **versión matricial**, para cada sujeto  $i$  tenemos:

- Un vector con  $J$  respuestas observadas:  $\mathbf{y}_i$ .
- Un vector con  $J$  intersecciones:  $\boldsymbol{\nu}$ .
- Una matriz con  $J \times M$  pesos factoriales:  $\boldsymbol{\Lambda}$ .
- Un vector con  $M$  puntuaciones latentes:  $\boldsymbol{\eta}_i$ .
- Un vector con  $J$  puntuaciones error:  $\boldsymbol{\varepsilon}_i$ .

## El modelo psicométrico del factor común

- Ambas versiones representan **dos formas distintas** del mismo modelo
- La versión **condicional**

$$Y_{ij} \sim \mathcal{N} \left( \mu = \nu_j + \sum_{m=1}^M \lambda_{jm} \cdot \eta_{im}, \sigma = \theta \right)$$

- La versión **marginal**

$$\mathbf{y}_i \sim \mathcal{MVN} (\mu = \boldsymbol{\nu} + \boldsymbol{\Lambda} \cdot \boldsymbol{\alpha}, \Sigma = \boldsymbol{\Lambda} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Lambda}' + \boldsymbol{\Theta})$$

- $\boldsymbol{\alpha}$  es el vector con las medias de  $\boldsymbol{\eta}_i$ .  $\boldsymbol{\Psi}$  y  $\boldsymbol{\Theta}$  son las matrices de covarianzas de  $\boldsymbol{\eta}_i$  y  $\boldsymbol{\varepsilon}_i$ , respectivamente.

## El modelo psicométrico del factor común

- Ambas versiones representan **dos formas distintas** del mismo modelo
- La versión **condicional**

$$Y_{ij} \sim \mathcal{N} \left( \mu = \nu_j + \sum_{m=1}^M \lambda_{jm} \cdot \eta_{im}, \sigma = \theta \right)$$

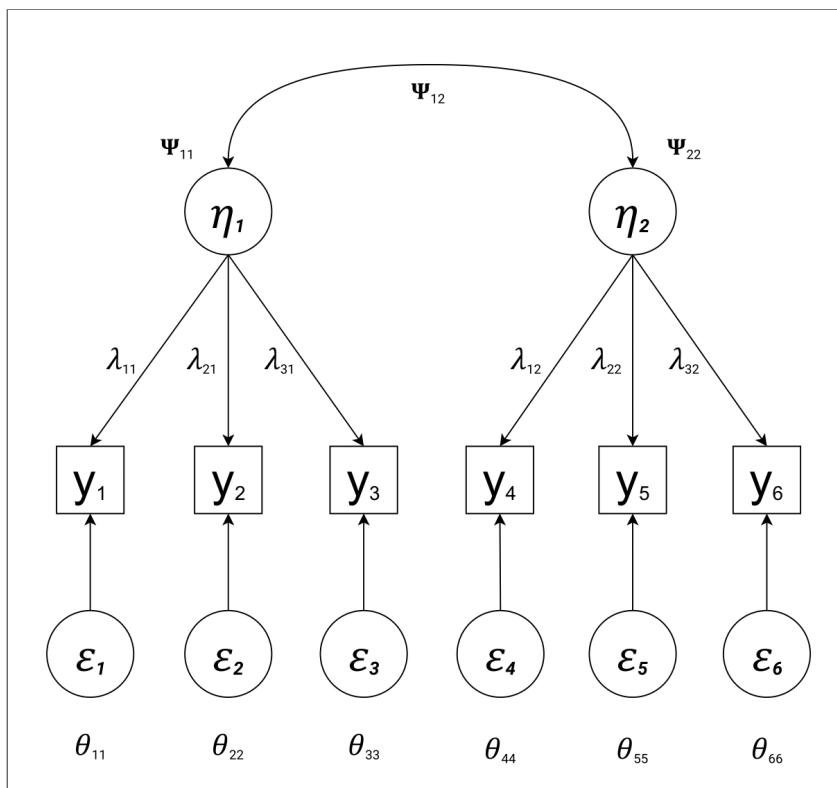
- La versión **marginal**

$$\mathbf{y}_i \sim \mathcal{MVN} (\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \cdot \boldsymbol{\alpha}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Lambda}' + \boldsymbol{\Theta})$$

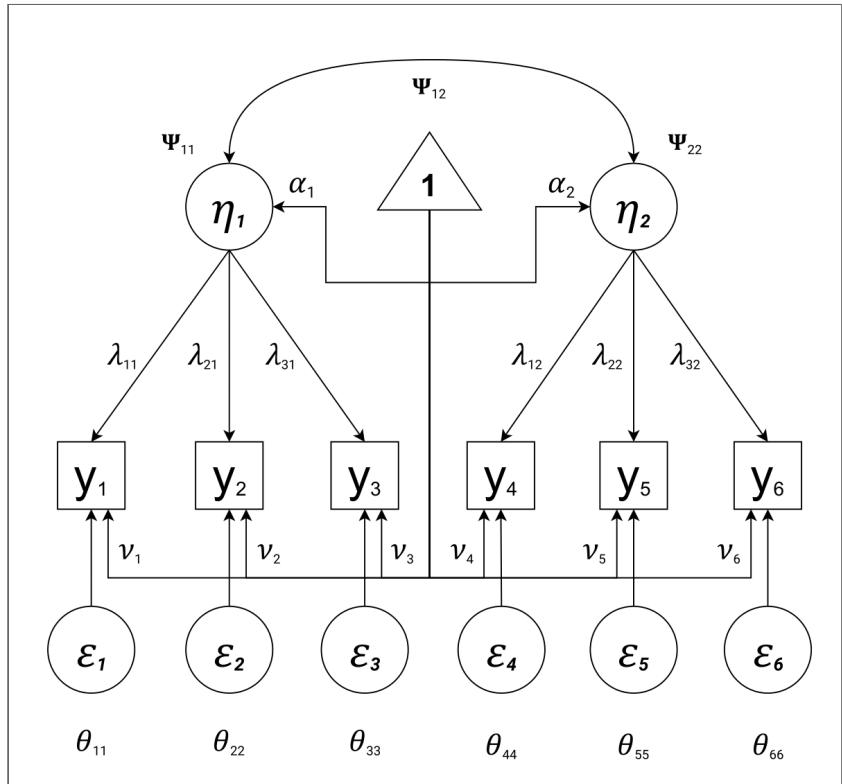
- $\boldsymbol{\alpha}$  es el vector con las medias de  $\boldsymbol{\eta}_i$ .  $\boldsymbol{\Psi}$  y  $\boldsymbol{\Theta}$  son las matrices de covarianzas de  $\boldsymbol{\eta}_i$  y  $\boldsymbol{\varepsilon}_i$ , respectivamente.

# Simulando datos desde el modelo en R

## Modelo sin estructura de medias



## Modelo con estructura de medias



## Simulando datos desde el modelo en R

$$\mathbf{y}_i \sim \mathcal{MVN} (\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \cdot \boldsymbol{\alpha}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Lambda}' + \boldsymbol{\Theta})$$

- Intersecciones de los ítems:  $\boldsymbol{\nu}$

$$\boldsymbol{\nu} = [-1, 0, 1, -1, 0, 1]$$

```
1 # Vector de intersecciones poblacional  
2 Nu <- c(-1, 0, 1, -1, 0, 1)
```

## Simulando datos desde el modelo en R

$$\mathbf{y}_i \sim \mathcal{MVN} (\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \cdot \boldsymbol{\alpha}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Lambda}' + \boldsymbol{\Theta})$$

- Pesos factoriales no estandarizados:  $\boldsymbol{\Lambda}$

$$\boldsymbol{\Lambda}' = \begin{bmatrix} 1.40 & 1.75 & 2.10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.40 & 1.75 & 2.10 \end{bmatrix}$$

```
1 Lambda <- matrix(c(1.40, 1.75, 2.10, 0, 0, 0,
2                               0, 0, 0, 1.40, 1.75, 2.10), ncol = 2)
3 t(Lambda) # Lambda transpuesta
```

```
[,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1.4 1.75 2.1 0.0 0.00 0.0
[2,] 0.0 0.00 0.0 1.4 1.75 2.1
```

## Simulando datos desde el modelo en R

$$\mathbf{y}_i \sim \mathcal{MVN} (\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \cdot \boldsymbol{\alpha}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Lambda}' + \boldsymbol{\Theta})$$

- Medias de las variables latentes:  $\boldsymbol{\alpha}$

$$\boldsymbol{\alpha} = [0, 0]$$

```
1 # Medias latentes poblacionales  
2 alpha <- c(0, 0)
```

## Simulando datos desde el modelo en R

$$\mathbf{y}_i \sim \mathcal{MVN} (\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \cdot \boldsymbol{\alpha}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Lambda}' + \boldsymbol{\Theta})$$

- Matriz de covarianzas entre las variables latentes:  $\boldsymbol{\Psi}$

$$\boldsymbol{\Psi} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

```
1 # Matriz de covarianzas latente poblacional  
2 Psi <- matrix(c(1, .5, .5, 1), ncol = 2)
```

## Simulando datos desde el modelo en R

$$\mathbf{y}_i \sim \mathcal{MVN} (\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \cdot \boldsymbol{\alpha}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Lambda}' + \boldsymbol{\Theta})$$

- Matriz de varianzas únicas:  $\boldsymbol{\Theta}$

$$\boldsymbol{\Theta} = \begin{bmatrix} \theta_{11}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \theta_{22}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \theta_{33}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \theta_{44}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \theta_{55}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \theta_{66}^2 \end{bmatrix}$$

## Simulando datos desde el modelo en R

$$\mathbf{y}_i \sim \mathcal{MVN} (\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \cdot \boldsymbol{\alpha}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Lambda}' + \boldsymbol{\Theta})$$

- Matriz de varianzas únicas:  $\boldsymbol{\Theta}$

$$\boldsymbol{\Theta} = \begin{bmatrix} 2.04 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.1875 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.59 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.04 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.1875 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.59 \end{bmatrix}$$

## Simulando datos desde el modelo en R

$$\mathbf{y}_i \sim \mathcal{MVN} (\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \cdot \boldsymbol{\alpha}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Lambda}' + \boldsymbol{\Theta})$$

- Matriz de varianzas únicas:  $\boldsymbol{\Theta}$

```
1 # Matriz de covarianzas únicas poblacional
2 (Theta <- diag(c(2.04, 3.1875, 4.59, 2.04, 3.1875, 4.59), (
```

```
[,1]   [,2] [,3] [,4]   [,5] [,6]
[1,] 2.04 0.0000 0.00 0.00 0.0000 0.00
[2,] 0.00 3.1875 0.00 0.00 0.0000 0.00
[3,] 0.00 0.0000 4.59 0.00 0.0000 0.00
[4,] 0.00 0.0000 0.00 2.04 0.0000 0.00
[5,] 0.00 0.0000 0.00 0.00 3.1875 0.00
[6,] 0.00 0.0000 0.00 0.00 0.0000 4.59
```

## Simulando datos desde el modelo en R

$$\mathbf{y}_i \sim \mathcal{MVN} (\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \cdot \boldsymbol{\alpha}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Lambda}' + \boldsymbol{\Theta})$$

- ¡Ahora podemos calcular  $\boldsymbol{\mu}$  y  $\boldsymbol{\Sigma}$  tal y como indica el modelo!

```
1 # Parámetros del modelo multivariante  
2 mu <- Nu      # Como alpha = 0, Lambda * alpha = 0.  
3 Sigma = Lambda %*% Psi %*% t(Lambda) + Theta
```

- Finalmente, simulamos las respuesta de  $n = 300$  participantes.

```
1 # Simulamos datos desde el modelo  
2 library(MASS) # Tiene la distribución normal multivariante  
3 set.seed(2025) # Simulación reproducible  
4 Y <- mvrnorm(n = 300, mu = mu, Sigma = Sigma)  
5 Y <- as.data.frame(Y) # Guardamos como data frame
```

# **Estimación a través de {blavaan}**

## Sintaxis del modelo

- La sintaxis de `{blavaan}` es idéntica a la de `{lavaan}`.
- Especificaremos dos modelos: uno con **un único factor** común y otro con **dos factores relacionados** (el verdadero).

```
1 # Sintaxis del modelo: un factor común
2 model.1f <- '
3 F1 =~ V1 + V2 + V3 + V4 + V5 + V6
4 '
5 # Sintaxis del modelo: dos factores relacionados
6 model.2f <- '
7 F1 =~ V1 + V2 + V3
8 F2 =~ V4 + V5 + V6
9 F1 ~~ F2
10 '
```

## Prior predictive check

- ¿Qué priors está utilizando `{blavaan}` para los parámetros del modelo?

```
1 # By-default prior distributions  
2 dpriors()
```

Distribución a priori de los parámetros del CFA

Parámetro	priors	Parámetro	priors
$\nu$	normal(0,32)	$\theta$	gamma(1,.5)[sd]
$\alpha$	normal(0,10)	$\psi$	gamma(1,.5)[sd]
$\lambda$	normal(0,10)	$\rho$	beta(1,1)

## Prior predictive check

- `{blavaan}` tiene un atajo para realizar **prior predictive checks**.

```
1 # Ajusta un modelo "vacío" prisamp = TRUE!
2 prior_predictive <- bcfa(model.2f, data = Y, std.lv = TRUE,
3                           meanstructure = TRUE, test = "none",
4                           sample = 2e4, prisamp = TRUE,
5                           bcontrol = list(cores = 3))
```

- Acto seguido, podemos guardar la **distribución a priori** con la siguiente línea de código

```
1 # Guardamos la distribución a priori
2 prior.draws <- blavInspect(prior_predictive, "mcmc")
```

- Graficaremos las distribuciones con el paquete `{bayesplot}`

## Prior predictive check: intersecciones

```
1 # Prior predictive distribution: intersecciones  
2 mcmc_hist(x = prior.draws, regex_pars = "~1", alpha = .5)
```

## Prior predictive check: pesos factoriales

```
1 # Prior predictive distribution: pesos factoriales  
2 mcmc_hist(x = prior.draws, regex_pars = "=~", alpha = .5)
```

## Prior predictive check: correlaciones latentes

```
1 # Prior predictive distribution: correlación latente  
2 mcmc_hist(x = prior.draws, pars = "F1~F2", alpha = .5)
```

## Prior predictive check: varianzas únicas

```
1 # Prior predictive distribution: correlación latente  
2 pars <- paste0("V", 1:6, "~", "V", 1:6)  
3 mcmc_hist(x = prior.draws, pars = pars, alpha = .5)
```

---

## Cambiando la distribución prior

- En `{blavaan}` no se puede cambiar la distribución a priori, sólo sus parámetros.

```
1 # Nueva distribución a priori
2 (new_priors <- dpriors(nu = "normal(0,15)",
3                         lambda = "normal(0,5)",
4                         rho = "beta(2,2)",
5                         psi = "gamma(1,.2)"))
6 # Muestreamos de la nueva distribución a priori
7 new_prior_pred <- bcfa(model.2f, data = Y, std.lv = TRUE,
8                         meanstructure = TRUE, test = "none",
9                         sample = 2e4, prisamp = TRUE,
10                        dp = new_priors, # Added new priors
11                        bcontrol = list(cores = 3))
```

## Cambiando la distribución prior

- Volvemos a extraer la distribución a priori

```
1 new.prior.draws <- blavInspect(new_prior_pred, "mcmc")
```

## Prior predictive check: intersecciones

```
1 # Prior predictive distribution: intersecciones  
2 mcmc_hist(x = new.prior.draws, regex_pars = "~1", alpha =
```

## Prior predictive check: pesos factoriales

```
1 # Prior predictive distribution: pesos factoriales  
2 mcmc_hist(x = new.prior.draws, regex_pars = "=~", alpha =
```

## Prior predictive check: correlaciones latentes

```
1 # Prior predictive distribution: correlación latente  
2 mcmc_hist(x = new.prior.draws, pars = "F1~F2", alpha = .5)
```

## Prior predictive check: varianzas únicas

```
1 # Prior predictive distribution: correlación latente  
2 pars <- paste0("V", 1:6, "~", "V", 1:6)  
3 mcmc_hist(x = new.prior.draws, pars = pars, alpha = .5)
```

## Estimación del modelo

- Estimaremos los modelos de uno y dos factores **con las nuevas priors**.
  - Utilizaremos **3 cadenas de Markov con 1500 iteraciones**, descartando las primeras 500 iteraciones.
  - **bcontrol** permite **parallelizar** las cadenas de Markov.

```
1 # Modelo de un factor  
2 bcfa.1f.fit <- bcfa(model.1f, data = Y, burnin = 500,  
3 sample = 1000, meanstructure = TRUE,  
4 std.lv = TRUE, bcontrol = list(cores =  
5 # Modelo de dos factores  
6 bcfa.2f.fit <- bcfa(model.2f, data = Y, burnin = 500,  
7 sample = 1000, meanstructure = TRUE,  
8 std.lv = TRUE, bcontrol = list(cores =
```

## Convergencia del modelo

- La **validez** de las inferencias depende de la **convergencia** del modelo.
- *Potential Scale Reduction Factor* (PSRF,  $\hat{R}$ ) con valores inferiores a 1.05

```
1 # Potential Scale Reduction Factor (PSRF): 1 factor  
2 which(blavInspect(bcfa.1f.fit, what = "rhat") > 1.05)
```

```
named integer(0)
```

```
1 # Potential Scale Reduction Factor (PSRF): 2 factores  
2 which(blavInspect(bcfa.2f.fit, what = "rhat") > 1.05)
```

```
named integer(0)
```

- Esto sugiere que las cadenas de Markov **están muestreando de la misma distribución posterior de manera estable**.

## Convergencia del modelo

- Podemos **evaluar visualmente la convergencia de las cadenas de Markov** a través de los gráficos de trayectoria (**traceplots**).
- Para ello, necesitamos **extraer la distribución posterior** de los parámetros.

```
1 # Distribución posterior de ambos modelos  
2 posterior.1f <- blavInspect(bcfa.1f.fit, "mcmc")  
3 posterior.2f <- blavInspect(bcfa.2f.fit, "mcmc")
```

- El siguiente código genera **traceplots** para los pesos factoriales

```
1 # Gráficos de trayectoria  
2 mcmc_trace(x = posterior.1f, regex_pars = "F1=~")  
3 mcmc_trace(x = posterior.2f, regex_pars = c("F1=~", "F2=~"))
```

## **Convergencia del modelo**

---

## **Convergencia del modelo**

---

## Convergencia del modelo

- Podemos destacar una de las cadenas para evaluar su desempeño.

```
1 mcmc_trace_highlight(x = posterior.2f, regex_pars = "F1~~F10")
2                               highlight = 3)
```

## Convergencia del modelo

- ¿Qué aspecto tiene este gráfico cuando **la convergencia es mala**?

## Eficiencia

- Los valores de los parámetros de iteración a iteración suelen estar correlacionados.
- Cuanto **menor sea su correlación, más informativo será el valor** en cada muestra.
- Se evalúa a través del **tamaño muestral efectivo**.  
→ Valor mínimo recomendable:  $100 \times N^o$  de cadenas de Markov.

```
1 min(blavInspect(bcfa.1f.fit, "neff"))
```

```
[1] 947.9307
```

```
1 min(blavInspect(bcfa.2f.fit, "neff"))
```

```
[1] 2093.932
```

## Ajuste del modelo

- Similares a los índices de ajuste frecuentistas, ¡pero **con intervalos de credibilidad!**
- **Índices Bayesianos de ajuste global**
  - *Bayesian Root Mean Square Error of Approximation* (BRMSEA)
  - *Bayesian Unbiased Goodness-of-fit Index* ( $\Gamma$ )
  - *Bayesian Unbiased Goodness-of-fit Index* ajustado a muestras pequeñas ( $\Gamma_{adj}$ )
  - *Bayesian McDonald Index* (BMc)
- **Índices Bayesianos de ajuste incremental**
  - *Bayesian Comparative Fit Index* (BCFI)
  - *Bayesian Tucker-Lewis Index* (BTLI)

## Ajuste del modelo

## Índices de ajuste incremental en [{blavaan}](#)

Los índices de ajuste incremental bayesianos, **BCFI** y **BTLI**, comparan el ajuste del modelo propuesto con el de un modelo nulo. Sin embargo, **{blavaan}** no estima ese modelo nulo por defecto, como sí hace **{lavaan}**. Por ello, debemos especificar manualmente el modelo nulo y estimarlo.

```
1 # Sintaxis del modelo nulo  
2 model.null <- paste0("V",1:6,"~~","V",1:6,collapse = " \n"  
3 # Ajustamos el modelo nulo  
4 bcfa.null.fit <- bcfa(model.null, data = Y, burnin = 500,  
5 sample = 1000, meanstructure = TRUE,  
6 std.lv = TRUE, bcontrol = list(cores
```

## Ajuste del modelo: un factor común

```
1 # Indices de ajuste: un factor  
2 blav_fit_1f <- blavFitIndices(object = bcfa.1f.fit,  
3                                 baseline.model = bcfa.null.f:  
4 summary(blav_fit_1f, hpd = TRUE, prob = .95,  
5         central.tendency = c("mean","median","mode"))
```

	EAP	Median	MAP	SD	lower	upper
BRMSEA	0.313	0.313	0.312	0.007	0.301	0.327
BGammaHat	0.873	0.874	0.874	0.005	0.864	0.882
adjBGammaHat	0.228	0.230	0.235	0.030	0.169	0.282
BMc	0.804	0.805	0.806	0.008	0.789	0.818
BCFI	0.714	0.715	0.716	0.013	0.688	0.738



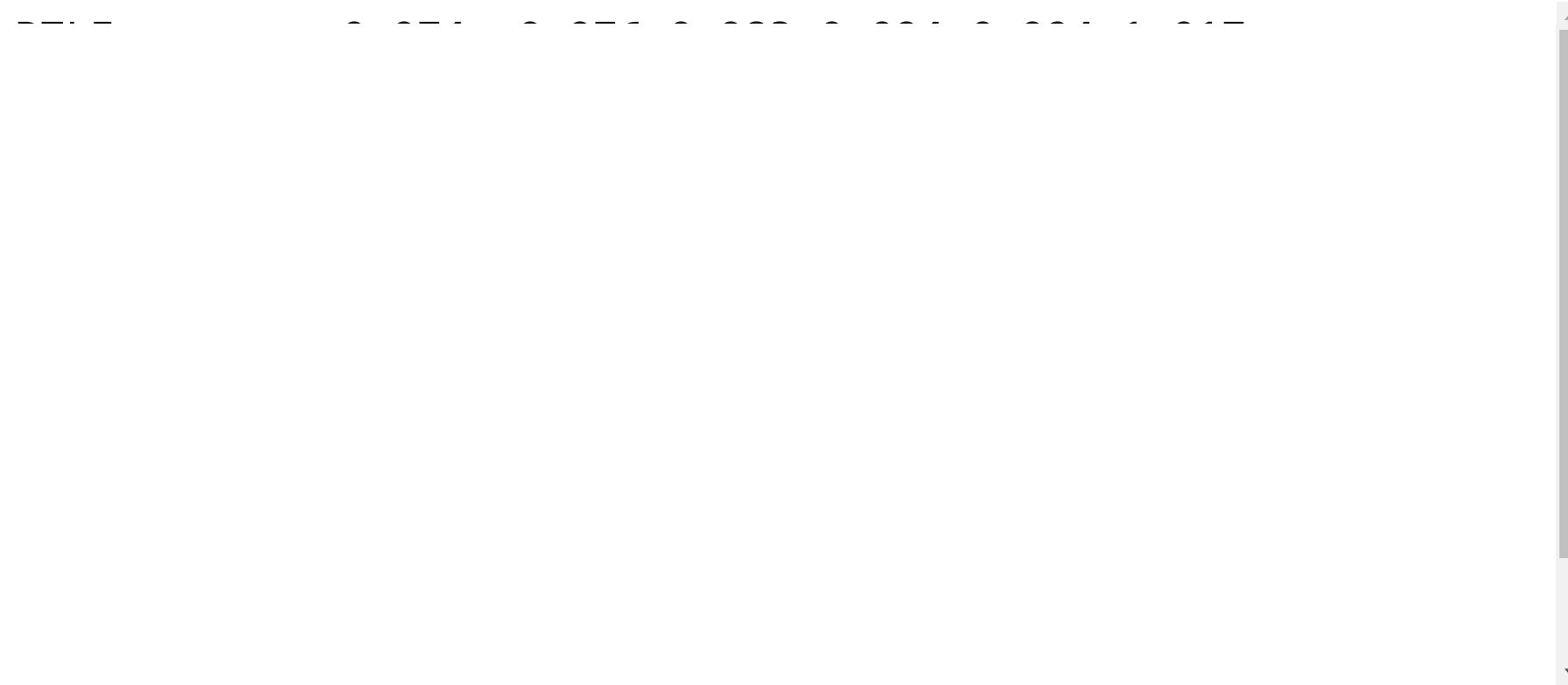
## Ajuste del modelo: un factor común

```
1 mcmc_hist(data.frame(blav_fit_1f@indices), alpha = 0.5,  
2           pars = c("BRMSEA", "BCFI", "BTLI", "BGammaHat"))
```

## Ajuste del modelo: dos factores correlacionados

```
1 # Indices de ajuste: dos factores
2 blav_fit_2f <- blavFitIndices(object = bcfa.2f.fit,
3                                baseline.model = bcfa.null.f:
4 summary(blav_fit_2f, hpd = TRUE, prob = .95,
5         central.tendency = c("mean","median","mode"))
```

	EAP	Median	MAP	SD	lower	upper
BRMSEA	0.046	0.050	0.057	0.026	0.000	0.085
BGammaHat	0.993	0.993	1.000	0.006	0.981	1.000
adjBGammaHat	0.975	0.978	0.998	0.021	0.936	1.000
BMc	0.989	0.990	0.999	0.009	0.972	1.000
BCFI	0.985	0.987	0.999	0.012	0.962	1.000



## Ajuste del modelo: dos factores correlacionados

```
1 mcmc_hist(data.frame(blav_fit_2f@indices), alpha = 0.5,  
2           pars = c("BRMSEA", "BCFI", "BTLI", "BGammaHat"))
```

# Comparación de modelos

- `{blavaan}` incluye dos métodos para comparar modelos:

## 1. Métodos basados en la **capacidad predictiva del modelo**:

- *Leave-One-Out (LOO) cross-validation.*
- *Watanabee-Akaike Information Criteria (WAIC).*
- Ambos incluidos en el paquete `{loo}`.
- Cuentan con **pruebas de significacion**.

## 2. Métodos basados en la **capacidad explicativa del modelo**:

- Factor de Bayes (aproximación de Laplace-Metrópolis).
- `{blavaan}` no recomienda su uso.

## Comparación de modelos

```
1 # Comparación de modelos  
2 blav_com_1f_2f <- blavCompare(object1 = bcfa.1f.fit,  
3                                 object2 = bcfa.2f.fit)  
4 # Leave-One-Out:  
5 blav_com_1f_2f$loo[[1]] # modelo de un factor  
6 blav_com_1f_2f$loo[[2]] # modelo de dos factores
```

Un factor		Dos factores		
	Estimación	Error típico	Estimación	Error típico
elpd_loo	-4048.296	34.544	-3984.390	32.102
p_loo	22.568	1.663	18.993	1.202
looic	8096.592	69.087	7968.780	64.204

## Comparación de modelos

```
1 # Comparación de ELPDs  
2 blav_com_1f_2f$diff_loo
```

	elpd_diff	se_diff
model2	0.0	0.0
model1	-63.9	12.2

- Model2 es el modelo que hemos indicado previamente en object 2

```
1 # Comparación de modelos  
2 blav_com_1f_2f <- blavCompare(object1 = bcfa.1f.fit,  
3                                     object2 = bcfa.2f.fit)
```

- Por tanto, el modelo de dos factores tiene mejor capacidad predictiva que el de un factor porque la diferencia  $\Delta\text{ELPD} = -63.9$  está fuera del rango  $\pm 2 \times 12.2$ .

## Estimaciones de los parámetros: medias posteriores

```
1 parameterEstimates(bcfa.2f.fit, standardized = TRUE,  
2                               output = "text")
```

Latent Variables:

	Estimate	Std.lv	Std.all
F1 =~			
V1	1.452	1.452	0.711
V2	2.091	2.091	0.759
V3	2.031	2.031	0.673
F2 =~			
V4	1.566	1.566	0.722
V5	1.671	1.671	0.708

## Distribución posterior: intersecciones

```
1 # Posterior distribution: histograms  
2 mcmc_hist(x = posterior.2f, regex_pars = "~1", alpha = .5)
```

## Distribución posterior: correlación latente

```
1 # Posterior distribution: density plot  
2 mcmc_dens(x = posterior.2f, pars = "F1~F2", alpha = .5)
```

## Distribución posterior: pesos factoriales

```
1 # Posterior distribution: pairs plots  
2 mcmc_pairs(posterior.2f, regex_pars = "F1=~", diag_fun = "|"  
3           off_diag_fun = "scatter", off_diag_args = list(a
```

## Distribución posterior: varianzas únicas

```
1 # Posterior distribution: pairs plots  
2 mcmc_intervals(posterior.2f, pars = paste0("V", 1:6, "~~",
```

## Distribución posterior: pesos factoriales

```
1 # Posterior distribution: pairs plots  
2 mcmc_areas(posterior.2f, regex_pars = c("F1=~", "F2=~"),  
3   prob = 0.8, prob_outer = 0.95, point_est = "median")
```

## Posterior Predictive Model Checks (PPMC)

- Método para evaluar la capacidad predictiva de un modelo.
- La idea central es **verificar si el model puede generar datos similares a los datos observados.**
- ¿Podemos distinguir los **datos observados** de **datos simulados desde el modelo?**
- En `{blavaan}` podemos utilizar la función `sampleData` para generar datos desde el modelo.
- Tras simularlo, se **comparan gráficamente** los datos observados y simulados

## Posterior Predictive Model Checks: $y^{\text{rep}}$

- Finalmente, podemos ver el parecido entre  $y$  con  $y^{\text{rep}}$

```
1 # Simulamos 100 bases de datos
2 yrep <- sampleData(bcfa.2f.fit, nrep = 100, simplify = TRUE)
3 # Guardamos los resultados para cada ítem
4 posterior_plots_list <- vector(mode = "list", length = ncol(Y))
5 for(i in 1:ncol(Y)){
6   posterior_plots_list[[i]] <- ppc_dens_overlay(
7     y = Y[,i], yrep = t(sapply(yrep, function(x) x[,i])))
8   )
9   labs(title = paste("Ítem", i))
10 }
11 # Gráfico comparando y e yrep
12 cowplot::plot_grid(plotlist = posterior_plots_list)
```

## Posterior Predictive Model Checks: $y^{\text{rep}}$

---

## Posterior Predictive Model Checks (PPMC)

- El PPMC también puede aplicarse a estadísticos específicos (SRMR,  $\chi^2$ ).
- Durante la estimación, en cada iteración:
  1. Se **extrae una muestra** de los parámetros  $\theta^{(s)}$  de la **distribución posterior** (esto lo hacemos **siempre**).
  2. Se utilizan estos parámetros para **generar datos simulados**:  $y^{\text{rep}}$ .
  3. Se **calcula algún estadístico** de interés en  $y^{\text{rep}}$  e  $y$ .
  4. Se **compara la distribución del estadístico** entre  $y$  e  $y^{\text{rep}}$ .
- En `{blavaan}` pueden realizarse con la función `ppmc`

# Posterior Predictive Model Checks (PPMC)

- En el modelo bayesiano, en cada iteracion:
  - Estimamos  $\Lambda^{(s)}$ ,  $\Theta^{(s)}$  y  $\Psi^{(s)}$ .
  - Estimamos  $\Sigma^{(s)} = \Lambda^{(s)} \cdot \Psi^{(s)} \cdot \Lambda'^{(s)} + \Theta^{(s)}$ .
  - Siempre tenemos la misma matriz de covarianzas observada:  $\text{Cov}(Y)$ .
  - Calculamos el estadístico  $\chi^{2(s)}$ , que refleja la discrepancia entre  $\text{Cov}(Y)$  y  $\Sigma^{(s)}$ .
  - Simulamos  $y^{rep(s)} \sim \mathcal{MVN}(\mu^{(s)}, \Sigma^{(s)})$
  - Calculamos la media y covarianza “observada” de  $y^{rep(s)}$ .
  - Calculamos el estadístico  $\chi^{2(s),rep}$ , que refleja la discrepancia entre  $\text{Cov}(y^{rep(s)})$  y  $\Sigma^{(s)}$ .

## Posterior Predictive Model Checks (PPMC): $\chi^2$ y SRMR

- La función `ppmc` puede utilizarse para calcular el estadístico  $\chi^2$  y SRMR en los datos observados y simulados en cada iteración.

```
1 # Posterior Predictive Model Checks  
2 ppmc_fit_1f <- ppmc(object = bcfa.1f.fit, thin = 1,  
3                         fit.measures = c("srmr","chisq"))  
4 ppmc_fit_2f <- ppmc(object = bcfa.2f.fit, thin = 1,  
5                         fit.measures = c("srmr","chisq"))
```

- Las funciones `plot` e `hist` comparan gráficamente los datos observados y simulados.

## Posterior Predictive Model Checks (PPMC): $\chi^2$

```
1 plot(ppmc_fit_2f, element = "chisq") # Dos factores
```

## Posterior Predictive Model Checks (PPMC): $\chi^2$

```
1 hist(ppmc_fit_2f, element = "chisq") # Dos factores
```

## Posterior Predictive Model Checks (PPMC): $\chi^2$

```
1 plot(ppmc_fit_1f, element = "chisq") # Un factor
```

---

## Posterior Predictive Model Checks (PPMC): $\chi^2$

```
1 hist(ppmc_fit_1f, element = "chisq")
```

## Posterior Predictive Model Checks (PPMC): SRMR

```
1 plot(ppmc_fit_2f, element = "srmr") # Dos factores
```

---

## Posterior Predictive Model Checks (PPMC): SRMR

```
1 hist(ppmc_fit_2f, element = "srmr") # Dos factores
```

---

## Posterior Predictive Model Checks (PPMC): SRMR

```
1 plot(ppmc_fit_1f, element = "srmr") # Un factor
```

---

## Posterior Predictive Model Checks (PPMC): SRMR

```
1 hist(ppmc_fit_1f, element = "srmr") # Un factor
```

---

## PPMC y Fiabilidad: $\omega$

- Podemos utilizar `ppmc` con cualquier función que pueda utilizarse para un modelo de {lavaan}.
- Podemos utilizar `compRelSEM` de `{SemTools}` para obtener la distribución posterior de  $\omega$ .

```
1 # Función para obtener la fiabilidad en cada iteración
2 bcfa.reliability <- function(fit){ semTools::compRelSEM(fi-
3 # Fiabilidad en el modelo unidimensional
4 ppmc_omega_1f <- ppmc(object = blavaan.1f.fit,
5                         discFUN = bcfa.reliability)
6 # Fiabilidad en el modelo de dos factores
7 ppmc_omega_2f <- ppmc(object = blavaan.2f.fit,
8                         discFUN = bcfa.reliability)
```

## PPMC y Fiabilidad: $\omega$

```
1 # Distribución posterior de la fiabilidad: un factor  
2 hist(unlist(posterior_omega_1f@obsDist$discFUN1))
```

---

## **PPMC y Fiabilidad:** $\omega$

---

# **Estimación a través de Stan y `{cmdstanr}`**

## **Stan: state of the art**

- Stan representa el *state of the art* de la estadística bayesiana.
- Algoritmo Monte Carlo Hamiltoniano más eficiente: el *No-U-Turn Sampler* (NUTS)
- Le debe su nombre a **Stanislaw Marcin Ulam**,
  - Matemático polaco, miembro del **proyecto Manhattan**
  - **Co-autor del método de Montecarlo.**

## Stan: state of the art

- En R, podemos usar `{Rstan}` o `{cmdstanr}`.
- Es **muy recomendable** utilizar `{cmdstanr}`:
  1. La elección de los creadores de Stan.
  2. Sólo `{cmdstanr}` utiliza la **última versión de Stan**.
  3. Evita saturaciones en la memoria RAM.
  4. **Más rápido y eficiente** en la estimación.
  5. Conectado con otros paquetes: `{posterior}`, `{loo}`, `{bayesplot}`...
- De hecho, es **más fácil** instalar Stan usando `{cmdstanr}`.

# Programación en Stan.

- Puede hacerse en [Rstudio](#).
- Cuenta con **siete** posibles bloques:
  1. **Functions**: espacio para crear funciones específicas.
  2. **Data**: se introducen los datos desde [R](#) como una lista.
  3. **Transformed data**: transformaciones sobre los datos introducidos.
  4. **Parameters**: parámetros que muestrearemos de la posterior.
  5. **Transformed parameters**: transformaciones sobre los parámetros muestreados.
  6. **Model**: priors y log-verosimilitud.
  7. **Generated quantities**: parámetros transformados post-estimación.

# Programación en Stan.

```
1 // Espacio para funciones
2 functions {
3
4 }
5 // Espacio para los datos
6 data {
7
8 }
9 // Espacio para transformar los datos
10 transformed data {
11
12 }
```

## Programación en Stan: Data

- Programaremos el código para un **modelo factorial confirmatorio**.
- Los datos necesarios son:
  1. Número de participantes:  $N$
  2. Número de ítems:  $J$
  3. Número de latentes:  $M$
  4. Matriz con los datos observados:  $Y$
  5. Matriz que indique qué ítem satura en qué factor.
- Esta es **mi forma de hacerlo**, pero hay infinitas versiones alternativas!

## Programación en Stan: Data

```
1 data {  
2     int<lower=0> N;                      // Number of observations  
3     int<lower=0> J;                      // Number of items  
4     int<lower=0> M;                      // Number of latent factors  
5     matrix[N, J] Y;                      // Observed data  
6     matrix[J, M] L_ind;                  // Lambda index matrix  
7 }
```

## Programación en Stan: Parámetros

- Vamos a especificar que el modelo muestree
  1. El **vector de medias** del modelo:  $\mu$ .
  2. La **matriz de pesos factoriales**:  $\Lambda$ .
  3. El **vector de varianzas únicas**:  $\theta$ .
  4. La **matriz triangular inferior** de la descomposición de Cholesky para las correlaciones latentes:  $L_\Psi$ 
    - Garantiza que la matriz de correlaciones es válida
    - Recuperamos  $\Psi$  como  $\Psi = L_\Psi \cdot L'_\Psi$

# Programación en Stan: Parámetros

- Vamos a especificar que el modelo muestree
  1. El **vector de medias** del modelo:  $\mu$ .
  2. La **matriz de pesos factoriales**:  $\Lambda$ .
  3. El **vector de varianzas únicas**:  $\theta$ .
  4. La **matriz triangular inferior** de la **descomposición de Cholesky** para las correlaciones latentes:  $L_{\Psi}$ .

```
1 parameters {  
2   vector[J] mu;  
3   matrix[J, M] L_unc;  
4   vector<lower=0>[J] Theta;  
5   cholesky_factor_corr[M] Psi_chol;  
6 }
```

# Programación en Stan: Parámetros transformados

1. Indicar qué peso va con cada factor latente.
2. Construir  $\Psi$  como  $L_\Psi \cdot L'_\Psi$ .
3. Construir  $\Sigma$  como  $\Sigma = \Lambda \cdot \Psi \cdot \Lambda' + \Theta$ .

```
1 transformed parameters{  
2     // Constrain Lambda matrix  
3     matrix[J, M] L_mat = L_unc .* L_ind;  
4     // Model-implied latent-variable correlation matrix  
5     corr_matrix[M] Psi_b = Psi_chol + Psi_chol';  
6     // Model-implied covariance matrix  
7     cov_matrix[J] Sigma;  
8     Sigma = L_mat * Psi_b * L_mat' + diag_matrix(Theta);  
9 }
```

## Programación en Stan: Modelo

- Sólo debemos **poner priors** en los **parámetros muestreados**.
- En este ejemplo usaremos priors poco informativas.

$$\boldsymbol{\mu} \sim \text{Normal}(\mu = 0, \sigma = 10)$$

$$\boldsymbol{\Lambda} \sim \text{Normal}(\mu = 0, \sigma = 5)$$

$$\boldsymbol{\theta} \sim \text{Gamma}(\alpha = 1, \beta = 0.5)$$

$$L_{\Psi} \sim \text{LkjCholesky}(\eta = 1)$$

- Finalmente, especificaremos la **función de verosimilitud** del modelo.

$$\mathbf{y}_i \sim \mathcal{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

# Programación en Stan: Modelo

```
1 model {
2     // Model priors
3     mu ~ normal(0, 10);           // Mu prior
4     to_vector(L_unc) ~ normal(0, 5); // Lambda prior
5     Theta ~ gamma(1, 0.5);        // Theta prior
6     Psi_chol ~ lkj_corr_cholesky(1); // Psi prior
7
8     // Model log-likelihood
9     for(i in 1:N){
10         Y[i,] ~ multi_normal(mu, Sigma);
11     }
12 }
```

# Programación en Stan: Cantidades generadas

- Este apartado permite realizar **cálculos tras ajustar el modelo**.
  - Otras transformaciones de parámetros.
  - Simulación de datos ( $y^{\text{rep}}$ )
  - Posterior Predictive Model Checks
  - Cálculo de log-verosimilitudes
- Nosotros **guardaremos la log-verosimilitud** para comparar modelos.

```
1 generated quantities {  
2     vector[N] log_lik;          // Model log-likelihood  
3     for(i in 1:N){  
4         log_lik[i] = multi_normal_lpdf(Y[i,] | mu, Sigma);  
5     }  
6 }
```

## Programación en Stan: Cantidades generadas

- El modelo estima los parámetros  $\mu$  y  $\Sigma$  en función de los datos,  $y$ .
- Sin embargo, podemos obtener **la misma  $\Sigma$  con distintas  $\Lambda$** :
  - Obtendríamos la misma  $\Sigma$  usando  $\Lambda$  o  $-\Lambda$ .
- **{blavaan}** soluciona este problema de signos en las cantidades generadas.
- El código de este modelo incluye esta corrección.
  - No lo mostraremos en detalle, basta con saber que se aplica para que el modelo esté identificado.

## Usando {cmdstanr}

- El primer paso es **compilar** nuestro modelo programado en **Stan**

```
1 # Compilamos el modelo  
2 BCFA <- cmdstan_model(stan_file = "CFA_marginal.stan")
```

- Una vez compilado, **{cmdstanr}** creará un **programa ejecutable** en vuestro ordenador.
- Al mismo tiempo, tendremos acceso a este **programa** con el objeto donde hayamos guardado el modelo en **R**.
- El objeto **BCFA** es **una especie de lista con todas las funciones que pueden utilizarse dentro**.

## Usando {cmdstanr}

- Por ejemplo, `BCFA$sample()` es la función para estimar el modelo bayesiano.
- Para **introducir los datos** en el modelo, **utilizaremos una lista**.
- Los objetos de la lista **deben coincidir en nombre con los del apartado de Datos en Stan**

```
1 sdata.1f <- list(  
2   N = 300,  
3   J = 6,  
4   M = 1,  
5   Y = Y,  
6   L_ind = matrix(rep(1, 6), ncol=1)  
7 )
```

## Usando {cmdstanr}

- BCFA\$sample() cuenta con muchos argumentos, pero no son similares a los de {blavaan}

```
1 BCFA_1f <- BCFA$sample(  
2   data = sdata.1f,           # Stan data  
3   chains = 4,              # Number of chains  
4   parallel_chains = 4,     # Number of parallel chains  
5   iter_warmup = 500,       # Adaptation iterations  
6   iter_sampling = 1500,     # Sampled iterations  
7   refresh = 500,           # Progress bar at 500 iterations  
8   init = 0)                # All starting values = 0
```

- BCFA-1f será una lista que contenga los resultados y funciones para almacenarlos y resumirlos.

## Usando {cmdstanr}

- La función `BCFA_1f$summary()` genera tablas resumen de los parámetros que indiquemos.

```
1 BCFA_1f$summary("Lambda") # Estimated factor loadings
```

variable	mean	median	sd	mad	q5	q95	rhat	ess_bu
Lambda[1,1]	0.72	0.72	0.13	0.13	0.52	0.93	1	8877
Lambda[2,1]	1.17	1.17	0.17	0.17	0.90	1.45	1	6893
Lambda[3,1]	1.30	1.29	0.20	0.20	0.97	1.63	1	6140
Lambda[4,1]	1.53	1.53	0.13	0.13	1.32	1.74	1	6879
Lambda[5,1]	1.99	1.99	0.16	0.16	1.72	2.26	1	7426
Lambda[6,1]	1.89	1.89	0.18	0.18	1.58	2.19	1	7576

## Usando {cmdstanr}

- Estimaremos también el modelo de dos factores.

```
1 sdata.2f <- list(  
2   N = 300,  
3   J = 6,  
4   M = 2,  
5   Y = Y,  
6   L_ind = ifelse(Lambda == 0, 0, 1)  
7 )
```

## Usando {cmdstanr}

- Estimaremos también el modelo de dos factores.

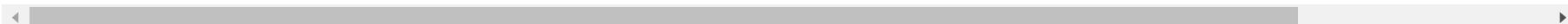
```
1 BCFA_2f <- BCFA$sample(  
2   data = sdata.2f,           # Stan data  
3   chains = 4,              # Number of chains  
4   parallel_chains = 4,     # Number of parallel chains  
5   iter_warmup = 500,       # Adaptation iterations  
6   iter_sampling = 1500,     # Sampled iterations  
7   refresh = 500,           # Progress bar  
8   seed = 2025,             # Reproducible results  
9   init = 0)                 # All starting values = 0
```

## Usando {cmdstanr}

- Estimaremos también el modelo de dos factores.

```
1 BCFA_2f$summary("Lambda") # Estimated factor loadings
```

variable	mean	median	sd	mad	q5	q95	rhat	ess_bu
Lambda[1,1]	1.09	1.09	0.12	0.12	0.90	1.29	1	6068
Lambda[2,1]	1.71	1.71	0.17	0.17	1.44	1.98	1	4970
Lambda[3,1]	2.20	2.20	0.18	0.19	1.90	2.50	1	4619
Lambda[4,2]	1.57	1.57	0.12	0.12	1.37	1.78	1	5773
Lambda[5,2]	2.05	2.05	0.16	0.16	1.78	2.31	1	5678
Lambda[6,2]	2.02	2.01	0.18	0.18	1.73	2.31	1	5434



## Comparando modelos

- Hemos guardado la log-verosimilitud de ambos modelos.
- Podemos acceder a todos sus valores usando `$draws()`
- Los utilizaremos para comparar los modelos con `{loo}`

```
1 loo_1f <- loo(x = BCFA_1f$draws("log_lik"))
2 loo_2f <- loo(x = BCFA_2f$draws("log_lik"))
3 loo_compare(loo_1f, loo_2f)
```

	elpd_diff	se_diff
model2	0.0	0.0
model1	-55.2	12.0



## **Bonus track 1: Comparación de modelos**

# Comparación de modelos

**Ojalá esto fuese más sencillo**

La comparación de modelos en estadística bayesiana ha sido (y sigue siendo) un tema de fervoroso debate entre dos disciplinas: **los psicólogos matemáticos y los estadísticos bayesianos.**

## Psicólogos matemáticos

**Objetivo explicativo:** teorías cognitivas y conductuales como modelos cuantitativos.

**Comparación de modelos:** métodos basados en la verosimilitud marginal.

**Métodos de comparación:** Factor de Bayes, *Bayesian Model Averaging...*

**Algoritmo MCMC:** Suelen utilizar el **sampleador de Gibbs** (e.g., JAGS).

## Estadísticos bayesianos

**Objetivo predictivo:** modelos empíricos con predicciones precisas.

**Comparación de modelos:** métodos basados en la validación cruzada.

**Métodos de comparación:** *Leave-One-Out cross-validation, Bayesian stacking...*

**Algoritmo MCMC:** Suelen usar el algoritmo **Monte Carlo Hamiltoniano** (e.g., Stan).

## Comparación de modelos: el Factor de Bayes

- La *verosimilitud marginal* es **la probabilidad de los datos observados** con el modelo que hemos ajustado.

$$\Pr(\theta | D) = \frac{\Pr(D | \theta) \times \Pr(\theta)}{\Pr(D)}$$

- El **Factor de Bayes** es la ratio entre la *verosimilitud marginal* de dos modelos  $\mathcal{M}$ .

$$BF_{12} = \frac{\Pr(D | \mathcal{M}_1)}{\Pr(D | \mathcal{M}_2)}$$

## Comparación de modelos: el Factor de Bayes

$$BF_{12} = \frac{\Pr(D | \mathcal{M}_1)}{\Pr(D | \mathcal{M}_2)}$$

- El Factor de Bayes **compara la probabilidad de los datos observados en cada modelo.**
- Permite responder a la pregunta: **¿Con qué modelo es más probable que se hayan generado los datos observados?.**
- $BF_{12} = 3$  significa que **los datos observados son tres veces más probables** con el modelo  $\mathcal{M}_1$  que con  $\mathcal{M}_2$ .
- Su cálculo exacto **no está disponible en modelos complejos**, pero **puede aproximarse** con métodos numéricos.

## **Limitaciones del factor de Bayes.**

- Las críticas más relevantes contra el Factor de Bayes son:
  1. Las aproximaciones pueden ser **inestables y sesgadas** en modelos complejos.
  2. Es **muy sensible** a las **distribuciones a priori**.
  3. Puede requerir **muchas extracciones de la posterior**.
- Existen recomendaciones para:
  1. Evaluar la inestabilidad y posible sesgo del Factor de Bayes.
  2. Evaluar el impacto de la distribución a priori.

## Comparación de modelos: Leave-One-Out

- El método *Leave-One-Out* (LOO) evalúa la capacidad de un modelo para predecir datos que no fueron utilizados en su ajuste.
- Permite responder a la pregunta: **¿Cómo de buenas son las predicciones del modelo fuera de la muestra original?**
- El procedimiento es sencillo:
  1. Quitamos una observación de la base de datos.
  2. Ajustamos el modelo con los datos restantes.
  3. Predecimos la observación que hemos quitado al principio.
  4. Calculamos la log-densidad predictiva
  5. Repetimos esto para todas las observaciones.
  6. Sumamos las log-densidades predictivas.

## Comparación de modelos: Leave-One-Out

- Ejemplo: **regresión lineal simple** ( $Y_i = \beta_0 + \beta_1 \cdot X_i + \varepsilon_i$ )

```
1 for (i in 1:n) {  
2   # Ajuste sin la i-ésima observación  
3   fit_loo <- lm(y[-i] ~ x[-i])  
4   # Predicción para la observación i  
5   y_pred_i <- predict(fit_loo, newdata = data.frame(x = x  
6   # Error de estimación sin la i-ésima obs  
7   sigma_i <- summary(fit_loo)$sigma  
8   # Cálculo del log-densidad predictiva de y[i]  
9   log_pdens[i] <- dnorm(y[i], mean = y_pred_i, sd = sigma_i,  
10                           log = TRUE)  
11 }  
12 # ELPD: suma de las log-densidades
```

## Comparación de modelos: Leave-One-Out

- ELPD son las siglas de *Expected log pointwise predictive density*, y su valor refleja la **calidad predictiva del modelo**.
- La comparación entre ELPDs permite saber **qué modelo predice mejor los datos fuera de la muestra**.

$$\Delta\text{ELPD}_{\mathcal{M}_1, \mathcal{M}_2} = \text{ELPD}_{\mathcal{M}_1} - \text{ELPD}_{\mathcal{M}_2}$$

- El paquete `{loo}` estima  $\Delta\text{ELPD}_{\mathcal{M}_1, \mathcal{M}_2}$  y su error típico.
- Si  $\Delta\text{ELPD}_{\mathcal{M}_1, \mathcal{M}_2}$  es mayor que  $\pm 2 \times$  el error típico, **la diferencia predictiva entre ambos modelos es significativa**.

## Limitaciones de Leave-One-Out

- Las críticas más relevantes son:
  - Computacionalmente costoso, **intratable en bases de datos grandes.**
  - **Sigue siendo sensible a las priors**, aunque menos que el  $BF$ .
  - Asumen que los datos son intercambiables (i.e., no hay un efecto del tiempo).
- Hay alternativas para algunos de los problemas:
  - **PSIS-LOO**: versión adaptada a bases de datos grandes.
  - **Leave-future-one-out** para tener en cuenta la secuencia temporal.

## Conectando ambos mundos: Pseudo-factor de Bayes

- El Pseudo-factor de Bayes (PsBF) **compara la capacidad predictiva de dos modelos de forma similar al Factor de Bayes.**
- Permite responder a la pregunta: **¿Qué modelo predice mejor los datos observados?**

$$\text{PsBF}_{12} = \exp(\Delta\text{ELPD}_{\mathcal{M}_1, \mathcal{M}_2})$$

- $\text{PsBF}_{12} = 3$  significa que las predicciones con el modelo  $\mathcal{M}_1$  son tres veces mejores que las predicciones con el modelo  $\mathcal{M}_2$ .

## Comparación de modelos

- `{blavaan}` incluye una estimación de la verosimilitud marginal y del factor de bayes basada en la **aproximación de Laplace-Metrópolis**.

```
1 # Comparación de ELPDs  
2 blav_com_1f_2f$bf
```

	bf	mll1	mll2
-62.32224	-4102.84295	-4040.52072	

- Como está en escala logarítmica, **debemos exponenciarlo**.

```
1 c(BF12 = exp(blav_com_1f_2f$bf[1]), # BF12  
2 BF21 = 1/exp(blav_com_1f_2f$bf[1])) # BF21
```

	BF12.bf	BF21.bf
	8.586119e-28	1.164671e+27

## Comparación de modelos

- Los datos observados son  $1.16 \times 10^{27}$  veces más probables con el modelo de dos factores que con el modelo de un factor.
- Finalmente, el **Pseudo-Factor de Bayes** se obtiene exponenciando  $\Delta\text{ELPD}$ .

```
1 # Incremento el ELPD  
2 ELPD_diff <- blav_com_1f_2f$diff_loo[2,"elpd_diff"]  
3 c(PsBF12 = exp(ELPD_diff), PsBF21 = 1/exp(ELPD_diff))
```

PsBF12	PsBF21
1.761748e-28	5.676181e+27

- La **capacidad predictive** del modelo de dos factores es  $5.67 \times 10^{27}$  veces mejor que la del modelo de un factor.