

Librerías de R para Estadística Bayesiana

Autor - Luis Bautista

12/09/2025



Introducción

La **Estadística Bayesiana** ha cobrado gran relevancia en los últimos años gracias al incremento del poder computacional y al desarrollo de algoritmos de simulación como *Markov Chain Monte Carlo (MCMC)*. En R, existen múltiples librerías que facilitan la implementación de modelos Bayesianos, desde enfoques clásicos hasta técnicas de última generación.

En este documento se presentan las principales librerías de R utilizadas para la **inferencia bayesiana**, con ejemplos prácticos y comparaciones.

Librerías Principales

1. rstan

- Interfaz de R para el lenguaje [Stan](#).
- Permite especificar modelos bayesianos complejos.
- Utiliza **Hamiltonian Monte Carlo (HMC)**, más eficiente que Gibbs en muchos casos.

```

# Instalación
install.packages("rstan", repos = "https://cloud.r-project.org/")
library(rstan)

# Ejemplo: Regresión Lineal bayesiana
model_code <- "
data {
  int<lower=0> N;
  vector[N] x;
  vector[N] y;
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}
model {
  y ~ normal(alpha + beta * x, sigma);
}
"

N <- 100
x <- rnorm(N)
y <- 1 + 2 * x + rnorm(N, 0, 1)
data_list <- list(N=N, x=x, y=y)

fit <- stan(model_code = model_code, data = data_list, iter=1000,
print(fit)

```

2. rstanarm

- Paquete oficial del equipo de Stan.
- Permite ajustar modelos bayesianos con sintaxis **similar a glm, lmer, glmer**, pero usando **Stan por detrás**.
- Ideal para quienes desean migrar de la estadística frecuentista a la bayesiana sin aprender el lenguaje Stan.

```

install.packages("rstanarm")
library(rstanarm)

# Ejemplo: regresión Logística bayesiana
data(mtcars)

```

```
fit_rstanarm <- stan_glm(vs ~ mpg + hp, data=mtcars, family=binomial)
summary(fit_rstanarm)
plot(fit_rstanarm, plotfun="areas", pars=c("mpg","hp"))
```

3. brms

- Basado en `rstan`, pero con **sintaxis estilo fórmula (como `lm/glm`)**.
- Muy usado por su facilidad y potencia.
- Ideal para modelos jerárquicos, GLMs, modelos de supervivencia, entre otros.

```
install.packages("brms")
library(brms)

fit_brms <- brm(y ~ x, data=data.frame(x,y), family=gaussian())
summary(fit_brms)
plot(fit_brms)
```

4. BayesFactor

- Diseñado para **comparación de modelos** mediante *Bayes Factors*.
- Fácil de usar en diseños experimentales (ANOVA, t-test, regresiones).

```
install.packages("BayesFactor")
library(BayesFactor)
t_test <- ttestBF(x = rnorm(20,0,1), y = rnorm(20,0.5,1))
t_test
```

5. MCMCpack

- Colección de funciones para realizar modelos bayesianos con MCMC.
- Incluye regresión lineal, logística y multinomial.
- Buena opción para **aprendizaje y docencia**.

```
install.packages("MCMCpack")
library(MCMCpack)
fit_mcmc <- MCMCregress(y ~ x)
summary(fit_mcmc)
```

6. rjags

- Interfaz a **JAGS (Just Another Gibbs Sampler)**.
- Similar a Stan pero con Gibbs Sampling.
- Muy usado en enseñanza y aplicaciones clásicas.

```
install.packages("rjags")
library(rjags)

model_string <- "
model {
  for (i in 1:N) {
    y[i] ~ dnorm(mu[i], tau)
    mu[i] <- alpha + beta * x[i]
  }
  alpha ~ dnorm(0, 0.001)
  beta ~ dnorm(0, 0.001)
  tau ~ dgamma(0.1, 0.1)
}
"

data_jags <- list(x=x, y=y, N=length(y))
model <- jags.model(textConnection(model_string), data=data_jags,
  update(model, 1000)
samples <- coda.samples(model, variable.names=c("alpha", "beta", "tau"),
  summary(samples)
```

7. bayesm

- Librería especializada en **econometría y marketing**.
- Soporta modelos de elección discreta, datos de panel, modelos de mezcla y jerárquicos.
- Ampliamente utilizada en investigación de mercados y modelos de demanda.

```
install.packages("bayesm")
library(bayesm)

# Ejemplo: regresión bayesiana simple
data(riceFarms)
y <- riceFarms$lprod
X <- cbind(1, riceFarms$size, riceFarms$seed)
out <- runireg(y, X)
summary(out$betadraw)
```

8. nimble

- Extensión de BUGS/JAGS con capacidad de programar algoritmos MCMC personalizados.
- Más flexible y rápido que JAGS en algunos casos.
- Ideal para **modelos jerárquicos complejos**.

```
# Instalación
install.packages("nimble")
library(nimble)

# Modelo en BUGS Language
code <- nimbleCode({
  for(i in 1:N){
    y[i] ~ dnorm(mu[i], tau) # modelo de regresión
    mu[i] <- alpha + beta * x[i]
  }
  alpha ~ dnorm(0, 0.001) # priori para intercepto
  beta ~ dnorm(0, 0.001) # priori para pendiente
  tau ~ dgamma(0.1, 0.1) # priori para precisión (1/varianza)
  sigma <- 1 / sqrt(tau)
})

# Datos simulados
set.seed(123)
N <- 100
x <- rnorm(N, 0, 1)
y <- 1 + 2 * x + rnorm(N, 0, 1)

data <- list(y = y, x = x)
constants <- list(N = N)
inits <- list(alpha = 0, beta = 0, tau = 1)
```

```

# Crear modelo en nimble
model <- nimbleModel(code, data = data, constants = constants, ini

# Configuración de MCMC
conf <- configureMCMC(model, monitors = c("alpha","beta","sigma"))
mcmc <- buildMCMC(conf)

# Compilar modelo y MCMC
cModel <- compileNimble(model)
cMCMC <- compileNimble(mcmc, project = model)

# Ejecutar MCMC
samples <- runMCMC(cMCMC, niter = 5000, nburnin = 1000, nchains =

# Resultados
summary(samples)
plot(samples)

```

Comparación de Librerías

Librería	Método de inferencia	Ventajas	Limitaciones
rstan	HMC/NUTS	Muy rápido, flexible, modelos complejos	Curva de aprendizaje alta
brms	HMC vía Stan	Sintaxis amigable, gran variedad de familias	Menos control que rstan
rstanarm	HMC vía Stan	Sintaxis idéntica a glm/lmer, oficial	Menos flexible que brms

Librería	Método de inferencia	Ventajas	Limitaciones
BayesFactor	Bayes Factors	Fácil de usar, útil en experimentos	No para modelos muy complejos
MCMCpack	Gibbs Sampling	Fácil de usar, educativo	Limitado en complejidad
rjags	Gibbs Sampling	Popular, buena documentación	Más lento que HMC
nimble	Personalizable	Algoritmos propios, gran flexibilidad	Requiere más programación
bayesm	Gibbs Sampling	Potente en econometría y marketing	Enfoque muy específico

Visualización y Diagnóstico

Paquetes complementarios:

- **bayesplot** → gráficos para diagnóstico MCMC.
- **loo** → *Leave-One-Out cross-validation* para comparar modelos.
- **tidybayes** → manipulación y visualización elegante de resultados bayesianos.

```
install.packages("bayesplot")
library(bayesplot)
mcmc_trace(as.array(fit), pars=c("alpha","beta"))
```

Conclusiones

- R ofrece una amplia gama de librerías para **Estadística Bayesiana**, desde herramientas básicas para la docencia (**MCMCpack** , **BayesFactor**) hasta librerías avanzadas para investigación (**rstan** , **brms** , **rstanarm** , **nimble** , **bayesm**).

- La elección depende del **nivel de complejidad del modelo**, **velocidad requerida** y **área de aplicación**.
- La tendencia actual se orienta hacia librerías basadas en **Stan** (**rstan**, **brms**, **rstanarm**) por su eficiencia y escalabilidad.
- Para aplicaciones en **econometría y marketing**, **bayesm** es una referencia obligada.

Referencias

- Carpenter et al. (2017). *Stan: A Probabilistic Programming Language*.
- Kruschke, J. (2015). *Doing Bayesian Data Analysis*.
- McElreath, R. (2020). *Statistical Rethinking*.
- Rossi, Allenby & McCulloch (2012). *Bayesian Statistics and Marketing*.
- [Stan Documentation](#)
- [brms vignette](#)
- [rstanarm CRAN](#)
- [bayesm CRAN](#)
- [BayesFactor CRAN](#)

Conéctate conmigo

