

Ejemplos

Lizbeth Naranjo Albarrán y Luz Judith Rodríguez Esparza

Paper: *Modelos ocultos de Markov:*

una aplicación de estimación Bayesiana para series de tiempo financieras

Authors: Lizbeth Naranjo Albarrán & Luz Judith Rodríguez Esparza

Journal: Mixba'al

Year: 2023

<https://github.com/lizbethna/HMMBayes.git>

Este archivo muestra las instrucciones para correr los códigos de R y Stan.

Distribución Gamma-Poisson

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
library(extraDistr)
library(rstan)
```

```
## Loading required package: StanHeaders
```

```
## rstan (Version 2.21.3, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
## options(mc.cores = parallel::detectCores()).
```

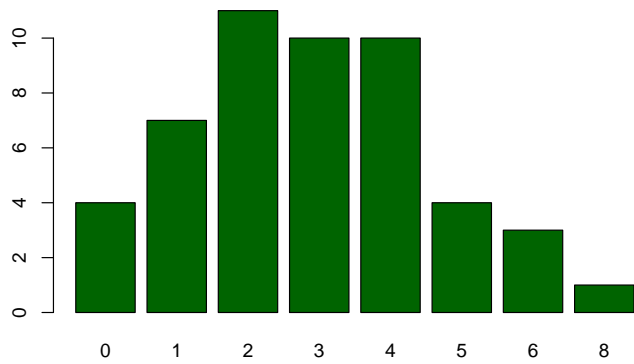
```
## To avoid recompilation of unchanged Stan programs, we recommend calling
```

```
## rstan_options(auto_write = TRUE)
```

Simular datos

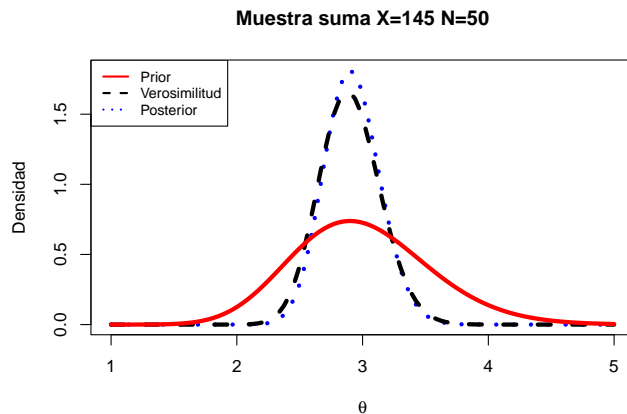
```
N = 50    # tamaño de muestra
theta = 3  # parametro de media
a0 = 30; b0 = 10 # hiperparametros de la distribucion inicial

set.seed(12345)
x = rpois(N, theta)    #  $x \sim \text{Poisson}(\theta)$ 
barplot(table(x), nclass=10, col="darkgreen")
```

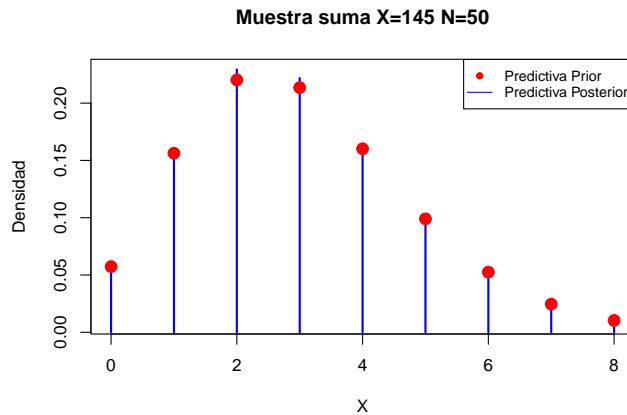


Graficas de las distribuciones final y predictiva final

```
### Posterior
a1 = a0+sum(x)
b1 = b0+N
the0 = seq(quantile(x,0.1),quantile(x,0.9),length.out=100)
plot(the0, dgamma(the0,a1,b1), main=paste0("Muestra suma X=",sum(x)," N=",N),
      xlab=expression(theta), ylab="Densidad", lty=3, lwd=4, col="blue", type="l")
lines(the0, dgamma(the0,sum(x),N), lty=2, lwd=4)
lines(the0,dgamma(the0,a0,b0), lty=1, lwd=4, col="red")
legend("topleft", legend=c("Prior","Verosimilitud","Posterior"),
      lty=c(1,2,3), col=c("red","black","blue"), lwd=2, cex=0.8)
```



```
### Predictiva Prior & Posterior
x0 = (0:max(x))
plot(x0,dgpois(x0, a1, b1), main=paste0("Muestra suma X=",sum(x), " N=",N),
      xlab="X", ylab="Densidad", type="h", lwd=2, col="blue")
points(x0,dgpois(x0, a0, b0), col="red", pch=19, cex=1.5)
legend("topright", legend=c("Predictiva Prior","Predictiva Posterior"),
      lty=c(NA,1), col=c("red","blue"), pch=c(19,NA), cex=0.8)
```



Código Stan

```
datos <- list( "x"=x, "N"=N, # muestra
               "a0"=a0,"b0"=b0) # valores iniciales de la distribucion inicial
param = c("theta","x_star") # parametros a estimar
fit_dist <- stan("dist_poisson_gamma.stan", data=datos,
                chains=2, warmup=1000, iter=2000, thin=2)
```

SAMPLING FOR MODEL 'dist_poisson_gamma' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 1.7e-05 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.17 seconds.

Chain 1: Adjust your expectations accordingly!

Chain 1:

Chain 1:

Chain 1: Iteration: 1 / 2000 [0%] (Warmup)

Chain 1: Iteration: 200 / 2000 [10%] (Warmup)

Chain 1: Iteration: 400 / 2000 [20%] (Warmup)

Chain 1: Iteration: 600 / 2000 [30%] (Warmup)

Chain 1: Iteration: 800 / 2000 [40%] (Warmup)

Chain 1: Iteration: 1000 / 2000 [50%] (Warmup)

Chain 1: Iteration: 1001 / 2000 [50%] (Sampling)

Chain 1: Iteration: 1200 / 2000 [60%] (Sampling)

Chain 1: Iteration: 1400 / 2000 [70%] (Sampling)

Chain 1: Iteration: 1600 / 2000 [80%] (Sampling)

Chain 1: Iteration: 1800 / 2000 [90%] (Sampling)

Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)

Chain 1:

Chain 1: Elapsed Time: 0.010215 seconds (Warm-up)

Chain 1: 0.009925 seconds (Sampling)

Chain 1: 0.02014 seconds (Total)

Chain 1:

SAMPLING FOR MODEL 'dist_poisson_gamma' NOW (CHAIN 2).

Chain 2:

Chain 2: Gradient evaluation took 4e-06 seconds

Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.

Chain 2: Adjust your expectations accordingly!

```
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 2000 [ 0%] (Warmup)
Chain 2: Iteration:   200 / 2000 [10%] (Warmup)
Chain 2: Iteration:   400 / 2000 [20%] (Warmup)
Chain 2: Iteration:   600 / 2000 [30%] (Warmup)
Chain 2: Iteration:   800 / 2000 [40%] (Warmup)
Chain 2: Iteration:  1000 / 2000 [50%] (Warmup)
Chain 2: Iteration: 1001 / 2000 [50%] (Sampling)
Chain 2: Iteration: 1200 / 2000 [60%] (Sampling)
Chain 2: Iteration: 1400 / 2000 [70%] (Sampling)
Chain 2: Iteration: 1600 / 2000 [80%] (Sampling)
Chain 2: Iteration: 1800 / 2000 [90%] (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 0.012651 seconds (Warm-up)
Chain 2:                    0.010188 seconds (Sampling)
Chain 2:                    0.022839 seconds (Total)
Chain 2:
```

Resultados

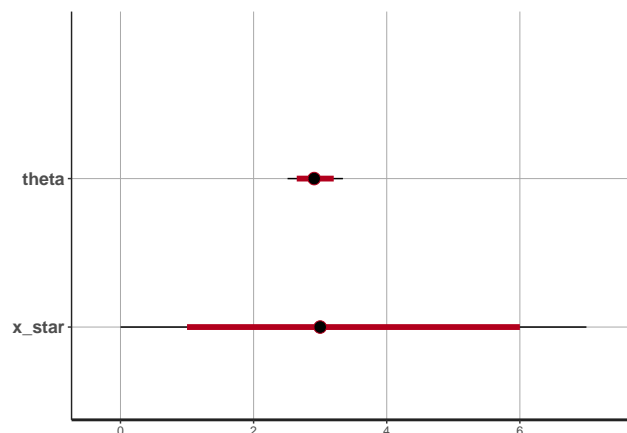
```
print(fit_dist, pars=param)
```

Inference for Stan model: dist_poisson_gamma.
 2 chains, each with iter=2000; warmup=1000; thin=2;
 post-warmup draws per chain=500, total post-warmup draws=1000.

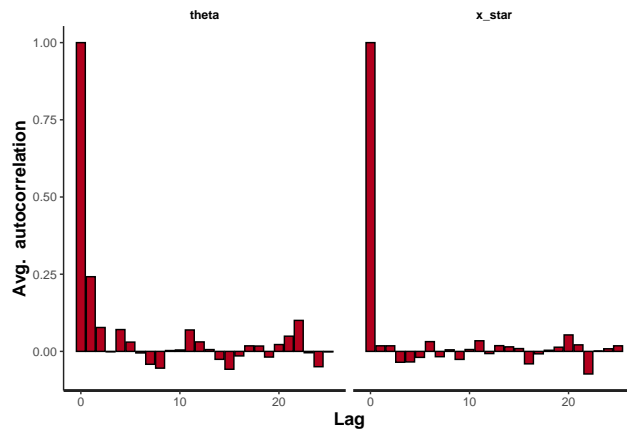
	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
theta	2.92	0.01	0.22	2.51	2.78	2.91	3.07	3.34	551	1
x_star	3.07	0.06	1.81	0.00	2.00	3.00	4.00	7.00	968	1

Samples were drawn using NUTS(diag_e) at Thu Jun 8 12:12:34 2023.
 For each parameter, n_eff is a crude measure of effective sample size,
 and Rhat is the potential scale reduction factor on split chains (at
 convergence, Rhat=1).

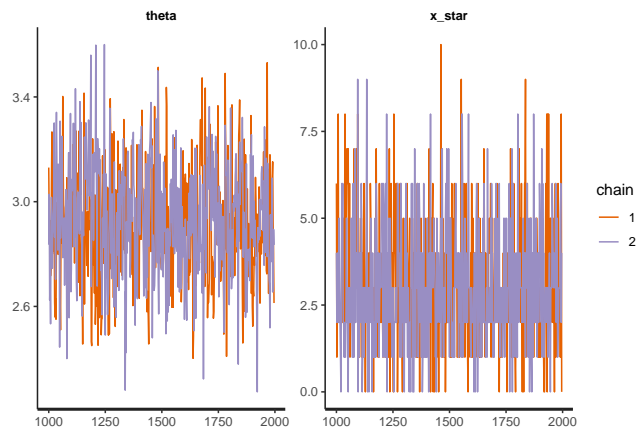
```
stan_plot(fit_dist, pars=param)
```



```
stan_ac(fit_dist, pars=param)
```



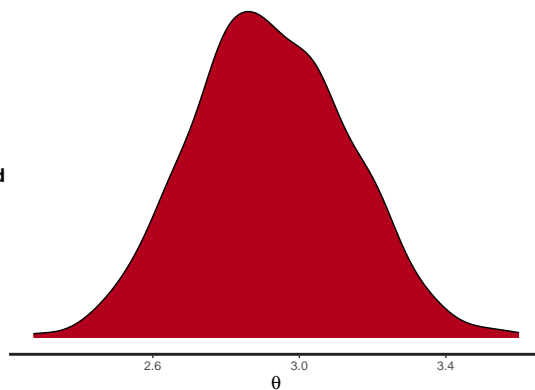
```
stan_trace(fit_dist, pars=param)
```



```
stan_dens(fit_dist, pars="theta", point_est = "mean", show_density = TRUE) +
  ggtitle(expression(paste("Distribución final de ", theta))) +
  ylab("Densidad") + xlab(expression(theta)) +
  theme(axis.title.x=element_text(size=14), axis.title.y=element_text(size=14),
        plot.title = element_text(size=16))
```

Distribución final de θ

Densidad



```
stan_hist(fit_dist, pars="x_star", point_est = "mean", show_density = TRUE) +
  ggtitle(expression(paste("Distribución predictiva final de ", x^{star}))) +
  ylab("Densidad") + xlab(expression(x^{star})) +
  theme(axis.title.x=element_text(size=14), axis.title.y=element_text(size=14),
        plot.title = element_text(size=16))
```

