Ejemplos

Lizbeth Naranjo Albarrán y Luz Judith Rodriguez 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.

Referencia

Damiano, L., Peterson, B., y Weylandt, M. (2018). A tutorial on hidden Markov models using Stan. En Stan conference.
https://github.com/luisdamiano/stancon18

Markov switching GARCH

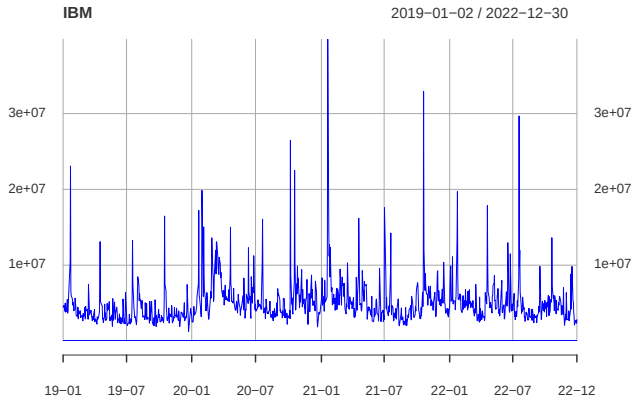
```
library(ggplot2)
library(rstan) # RStan
library(quantmod) # Quantitative Financial Modelling Framework
```

```
plot_statepath <- function(zstar) {
  K <- length(unique(as.vector(zstar)))
  x <- index(zstar)
  t <- 1:dim(zstar)[1]
  opar <- par(no.readonly = TRUE)
  zcol <- (1:K)[zstar]

  layout(matrix(c(1, 2), nrow = 2, ncol = 1), heights = c(0.95, 0.05))
  plot(x = x, y = zstar,
    xlab = bquote(t), ylab = bquote(hat(z)[t]),
    main = bquote("Secuencia mas probable de estados ocultos"),
    ylim = c(1, K), type = 'l', col = 'gray')

  points(x=x, y=zstar,
           pch = 21, bg = zcol, col = zcol, cex = 0.7)

  par(mai = c(0, 0, 0, 0))
  plot.new()
  legend(x = "center",
```

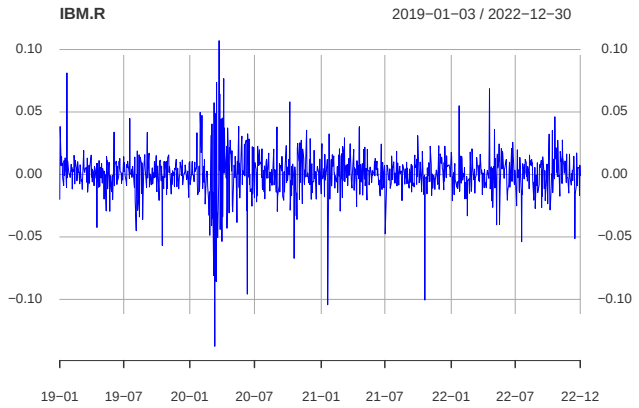
```
          legend = c('Trayectoria mas probable', paste('Estado', 1:K)),
          pch = c(NA, rep(21, K)),
          lwd = c(2, rep(NA, K)),
          col = c('lightgray', 1:K),
          pt.bg = c('lightgray', 1:K),
          bty = 'n', cex = 0.7,
          horiz = TRUE)
  par(opar)
}
```

Datos

```
IBM <- getSymbols("IBM",src='yahoo',
           from = "2019-01-01", to = "2022-12-31", auto.assign = FALSE)    # Obtener los datos
IBM.R <- na.omit(ROC(Ad(IBM)));    # Obtener los retornos

plot(IBM, format.labels="%y-%m", col="blue", lwd=0.5)
```




```
plot(IBM.R, format.labels="%y-%m", col="blue", lwd=0.5)
```



Código Stan

```r
# Markov-switching GARCH
msgarch_fit <- function(y) {
  rstan_options(auto_write = TRUE)
  options(mc.cores = parallel::detectCores())

  stan.model = 'hmm_garch.stan'

  y <- as.vector(coredata(y));
  stan.data = list(
    T = length(y),
    y = y
  )

  stan(file = stan.model,
       data = stan.data, verbose = FALSE,
       iter = 2000, warmup = 1000,
       thin = 2, chains = 2,
       cores = 2, seed = 12345)
}
# Fit GARCH
fit <- msgarch_fit(IBM.R)
```

Resultados

```r
round(summary(fit, pars=c("alpha0","alpha1","beta1","P"))$summary,3)
```

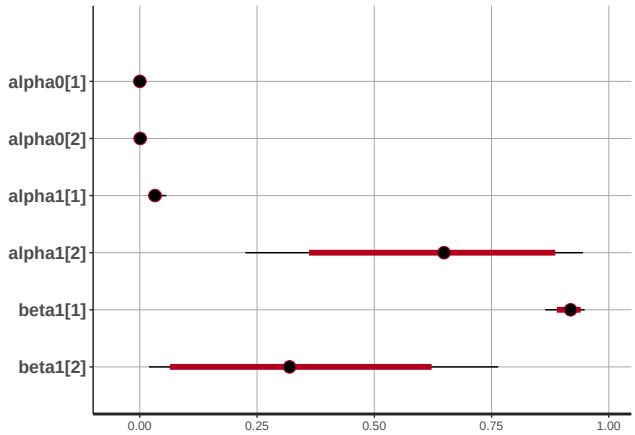
```
           mean se_mean    sd  2.5%   25%   50%   75% 97.5%   n_eff  Rhat
alpha0[1] 0.000   0.000 0.000 0.000 0.000 0.000 0.000 0.000 756.935 1.005
alpha0[2] 0.001   0.000 0.000 0.000 0.001 0.001 0.001 0.002 755.501 0.999
alpha1[1] 0.033   0.000 0.010 0.018 0.026 0.032 0.039 0.057 805.407 1.000
alpha1[2] 0.634   0.007 0.196 0.225 0.494 0.648 0.794 0.945 817.393 0.998
beta1[1]  0.916   0.001 0.022 0.864 0.905 0.919 0.931 0.949 669.287 1.004
beta1[2]  0.333   0.007 0.206 0.020 0.161 0.320 0.485 0.764 820.432 0.998
P[1,1]    0.938   0.001 0.016 0.901 0.928 0.940 0.949 0.964 762.630 1.000
P[1,2]    0.062   0.001 0.016 0.036 0.051 0.060 0.072 0.099 762.630 1.000
P[2,1]    0.544   0.003 0.098 0.367 0.470 0.542 0.612 0.748 875.886 1.000
P[2,2]    0.456   0.003 0.098 0.252 0.388 0.458 0.530 0.633 875.886 1.000
```

```r
round(summary(fit, pars=c("alpha0","alpha1","beta1","P"))$c_summary,3)
```

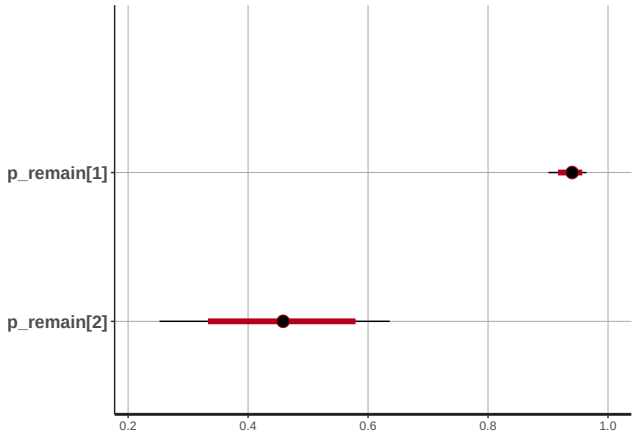
```
, , chains = chain:1

           stats
parameter    mean    sd  2.5%   25%   50%   75% 97.5%
  alpha0[1] 0.000 0.000 0.000 0.000 0.000 0.000 0.000
  alpha0[2] 0.001 0.000 0.000 0.001 0.001 0.001 0.002
  alpha1[1] 0.033 0.010 0.018 0.026 0.032 0.038 0.056
  alpha1[2] 0.631 0.196 0.242 0.490 0.644 0.794 0.944
  beta1[1]  0.917 0.021 0.863 0.907 0.919 0.932 0.949
  beta1[2]  0.336 0.206 0.020 0.167 0.324 0.489 0.751
```

```
  P[1,1]    0.937 0.017 0.900 0.927 0.940 0.949 0.965
  P[1,2]    0.063 0.017 0.035 0.051 0.060 0.073 0.100
  P[2,1]    0.544 0.099 0.369 0.469 0.539 0.611 0.748
  P[2,2]    0.456 0.099 0.252 0.389 0.461 0.531 0.631

, , chains = chain:2

           stats
parameter    mean    sd  2.5%   25%   50%   75% 97.5%
  alpha0[1] 0.000 0.000 0.000 0.000 0.000 0.000 0.000
  alpha0[2] 0.001 0.000 0.000 0.001 0.001 0.001 0.002
  alpha1[1] 0.034 0.009 0.018 0.027 0.033 0.039 0.057
  alpha1[2] 0.637 0.196 0.222 0.496 0.652 0.795 0.944
  beta1[1]  0.915 0.022 0.866 0.904 0.918 0.930 0.948
  beta1[2]  0.330 0.206 0.020 0.154 0.315 0.479 0.769
  P[1,1]    0.939 0.015 0.902 0.930 0.941 0.950 0.963
  P[1,2]    0.061 0.015 0.037 0.050 0.059 0.070 0.098
  P[2,1]    0.543 0.096 0.369 0.471 0.544 0.612 0.745
  P[2,2]    0.457 0.096 0.255 0.388 0.456 0.529 0.631
```

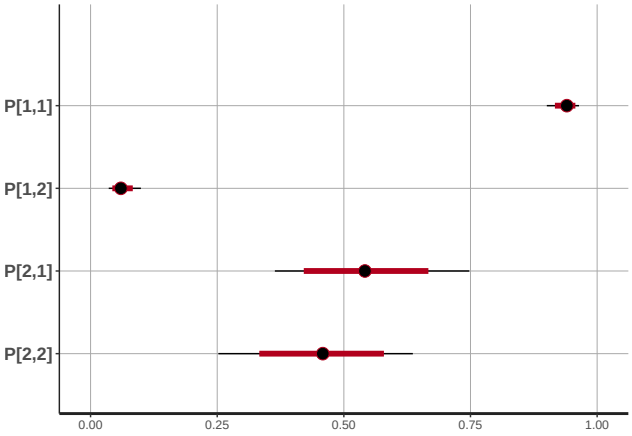
```
plot(fit,pars=c("alpha0","alpha1","beta1"))
```



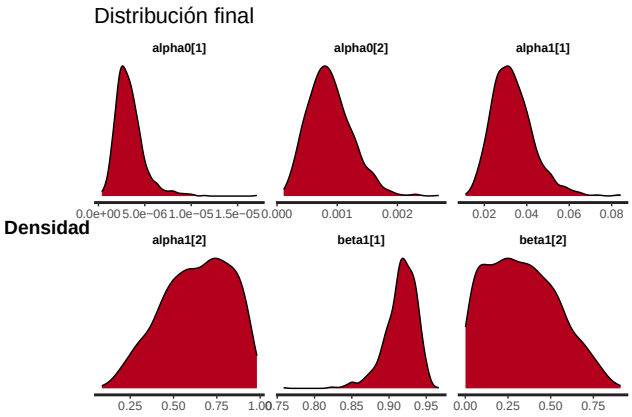

```
plot(fit,pars="p_remain")
```



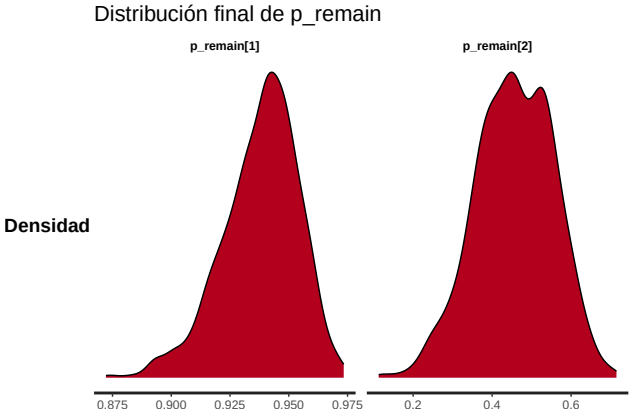
```
plot(fit,pars="P")
```



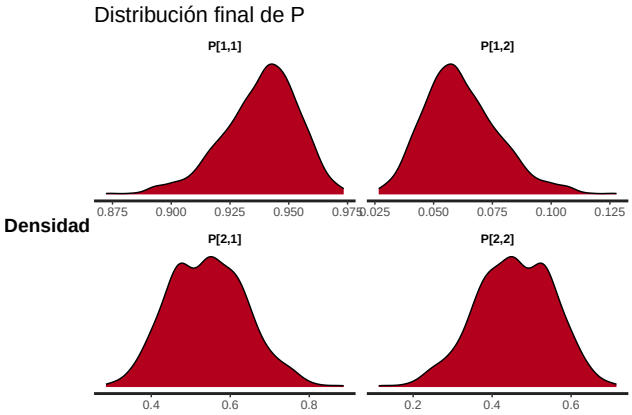

```
stan_dens(fit,pars=c("alpha0","alpha1","beta1"), point_est = "mean", show_density = TRUE) +
  ggtitle(expression("Distribución final",alpha[0],alpha[1],beta[1])) + ylab("Densidad") +
  theme(axis.title.x=element_text(size=14), axis.title.y=element_text(size=14),
        plot.title = element_text(size=16))
```



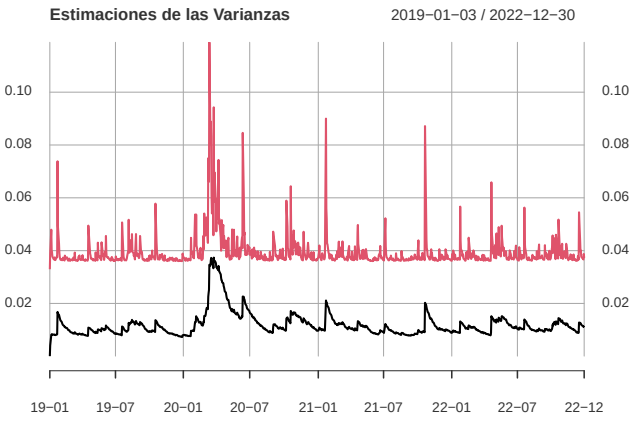

```
stan_dens(fit,pars="p_remain", point_est = "mean", show_density = TRUE) +
  ggtitle("Distribución final de p_remain") + ylab("Densidad") +
  theme(axis.title.x=element_text(size=14), axis.title.y=element_text(size=14),
        plot.title = element_text(size=16))
```



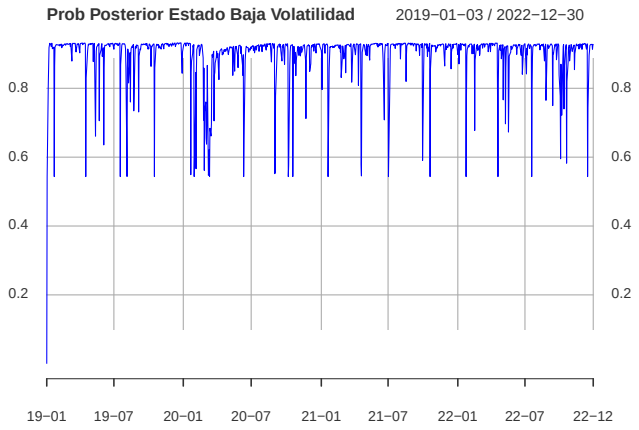
```
stan_dens(fit,pars="P", point_est = "mean", show_density = TRUE) +
  ggtitle("Distribución final de P") + ylab("Densidad") +
  theme(axis.title.x=element_text(size=14), axis.title.y=element_text(size=14),
        plot.title = element_text(size=16))
```



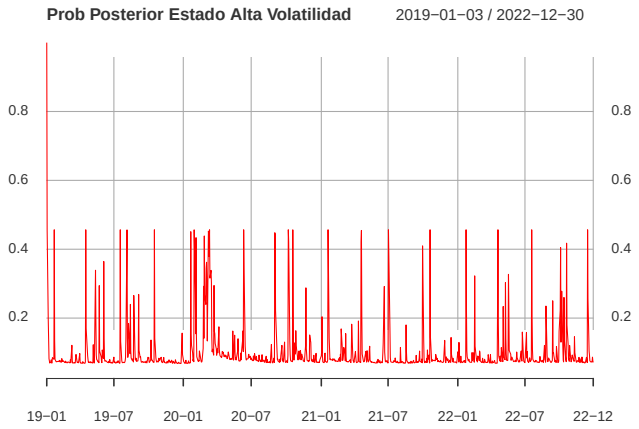

```
garch_posterior_means <- xts(apply(extract(fit, "sigma_t")[[1]], 2:3, mean),
                             index(IBM.R))
colnames(garch_posterior_means) <- c("Low-Vol State", "High-Vol State")

plot(
  garch_posterior_means,
  main = "Estimaciones de las Varianzas",
  format.labels = "%y-%m"
)
```




```
garch_posterior_prob1 <- xts(apply(extract(fit, "alpha")[[1]], 2:3, mean)[,1],
                             index(IBM.R))

plot(
  garch_posterior_prob1,
  main = "Prob Posterior Estado Baja Volatilidad",
  format.labels = "%y-%m",
  col="blue",lwd=1
)
```




```
garch_posterior_prob2 <- xts(apply(extract(fit, "alpha")[[1]], 2:3, mean)[,2],
                             index(IBM.R))
plot(
  garch_posterior_prob2,
  main = "Prob Posterior Estado Alta Volatilidad",
  format.labels = "%y-%m",
  col="red",lwd=1
)
```




```
zstar <- xts(apply(extract(fit, "zstar")[[1]], 2, median),
                             index(IBM.R))
plot_statepath(zstar)
```