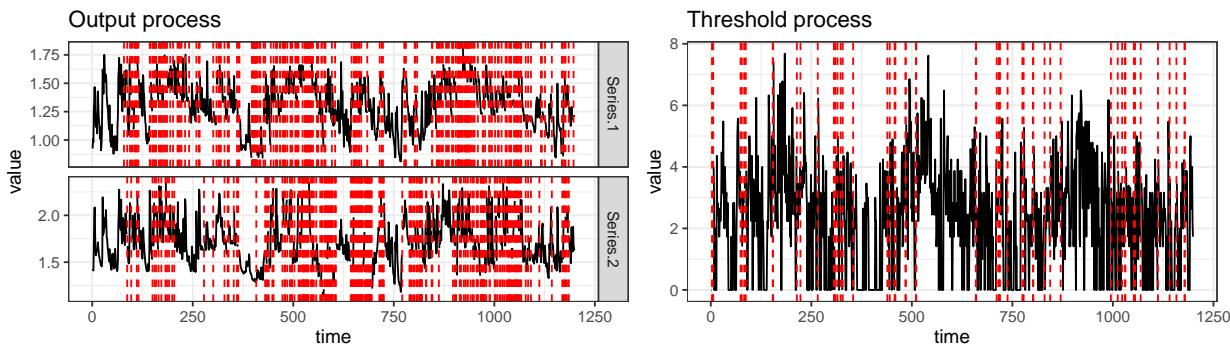


En la aplicación con datos se busca encontrar la relación entre la precipitación diaria (en mm) y el caudal diario del río (en m^3/s) de dos ríos donde un río desemboca en el otro en una región del departamento del Cauca en Colombia. La precipitación se midió en la estación meteorológica San Juan con una altitud de 2400 metros. El primer caudal del río se midió en la estación hidrológica “El Trébol” del río Bedón con una altitud de 1720 metros y el segundo caudal se midió en el río de la Plata en la estación hidrológica de “Villalosada” con una altitud de 1300 metros. Las estaciones están ubicadas cerca del ecuador de la Tierra en una zona geográfica muy seca. Esta última característica permite el control de factores hidrológicos/meteorológicos, los cuales pueden distorsionar el tipo de relación dinámica explicada por el modelo MTAR.

El período de tiempo que se consideró es del 1 de enero de 2006 al 14 de abril de 2009 (1200 puntos de tiempo) que tiene 57 puntos de tiempo con datos faltantes en la serie de precipitaciones, 214 en la serie del caudal del río Bedón y 213 en la serie del cauce del río de la Plata. Estos datos fueron proporcionados por el IDEAM (disponibles en la librería R para los modelos TAR y MTAR).

```
# install.packages("devtools")
# devtools::install_github("adrincont/BMTAR", force = TRUE)
library(BMTAR)
library(zoo)
library(patchwork)
imput_y = function(y){
  ts_prob = data.frame(time = 1:length(y), dat = y)
  ts_prob_2 = ts_prob[!is.na(ts_prob$dat),]
  ks_imput = ksmooth(x = ts_prob_2$time, y = ts_prob_2$dat, 'normal'
    , bandwidth= 10, x.points = 1:length(y))
  ts_prob$dat[is.na(ts_prob$dat)] = ks_imput$y[is.na(ts_prob$dat)]
  return(ts_prob$dat)
}

# [Lectura de Datos]
data("hydrodata")
Yt = hydrodata[,3:4]
Pt = hydrodata$Rainfall
data_miss = list(Yt = as.matrix(log(sqrt(Yt[-1,]))),
                 Zt = as.matrix(sqrt(Pt[-nrow(hydrodata)])))
## Graficas de las series .....#####
datos = tsregime(Yt = data_miss$Yt, Zt = data_miss$Zt)
autoplot(datos,1) + autoplot(datos,2)
```



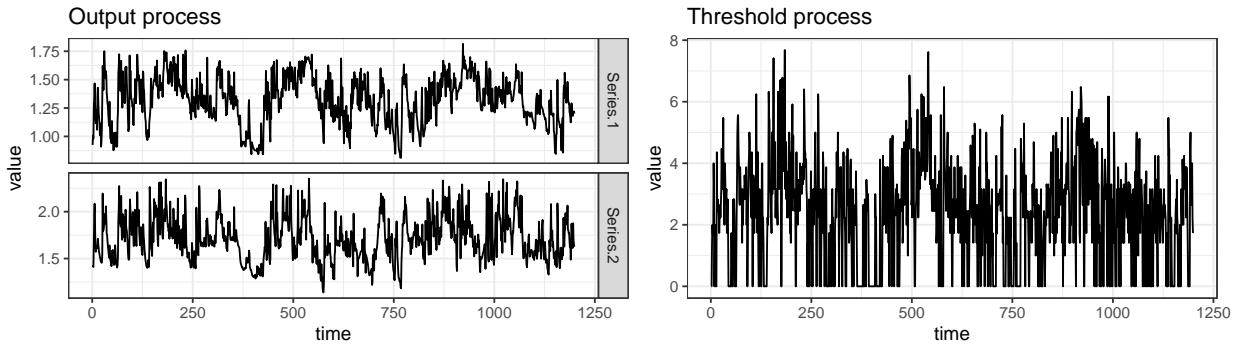
Para la aplicación de la metodología para estimación de modelos MTAR con datos faltantes se recurre a la Librería de R “MTAR” (librería basada en el mismo artículo y que implementa toda la metodología de estimación). Como primer paso y como se propone en el artículo se consideran las series transformadas $Y_t = (\ln \sqrt{Y_{1t}}, \ln \sqrt{Y_{2t}})$ y $Z_t = \sqrt{P_{t-1}}$ (precipitaciones y el caudal bivariado de los ríos Bedón y La Plata el día t). En la metodología de estimación para estos modelos, el primer paso corresponde a la imputación

inicial de datos faltantes que en este caso se realiza por medio de un promedio móvil sobre cada uno de los componentes.

```

Y_temp = datos$Yt
Y_temp[,1] = imput_y(Y_temp[,1])
Y_temp[,2] = imput_y(Y_temp[,2])
Z_temp = datos$Zt
Z_temp[,1] = imput_y(Z_temp[,1])
datos_temp = tsregime(Yt = Y_temp, Zt = Z_temp)
autoplot(datos_temp,1) + autoplot(datos_temp,2)

```



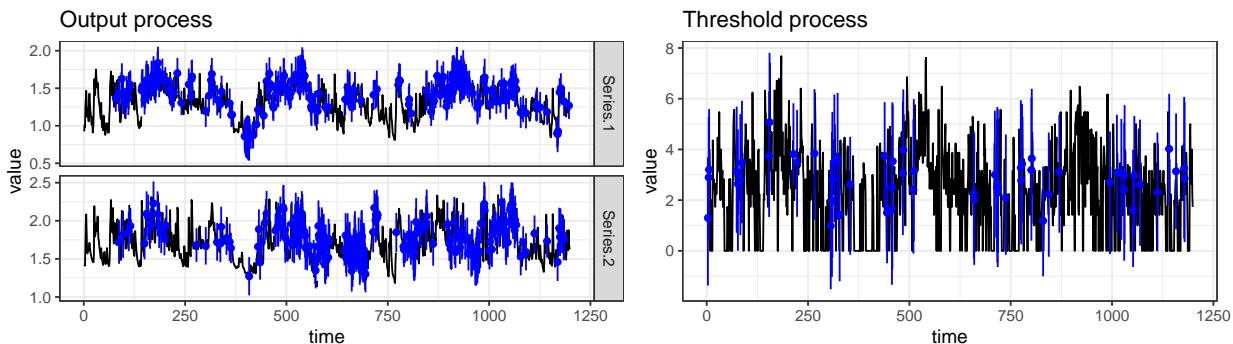
Una vez completada la serie, el siguiente paso corresponde a la estimación del número de regímenes mediante el método de selección de modelos de “Carlin and Chib” y el uso del criterio “NAIC”, seguido de la estimación de ordenes para el modelo por medio de la metodología de “Kuo and Mallick” (ambas metodologías explicadas en la primera parte de estos informes). Para ello, se usan las funciones “mtarnumreg()” y “mtarstr()” que implementan dichas metodologías usando prioris no informativas (el usuario podría dar prioris informativas si es el caso).

```

# initial = mtarinipars(tsregime_obj = datos_temp, list_model = list(l0_max = 3), method = 'KUD')
# estim_nr = mtarnumreg(ini_obj = initial, iterprev = 2000, niter_m = 3000, burn_m = 1000,
#                      list_m = TRUE, ordersprev = list(maxpj = 5, maxqj = 0, maxdj = 5))
# autoplot(estim_nr,1) + autoplot(estim_nr,2)

# list_model = list(pars = list(l = 2, r = 2.6, orders = list(pj = c(5,5), dj = c(4,5))))
# initial = mtarinipars(tsregime_obj = tsregime(Yt = data_miss$Yt, Zt = data_miss$Zt), list_model = list_
# missingest = mtarmissing(ini_obj = initial, chain = TRUE, niter = 5000, burn = 1000)
# data_c = missingest$tsregim
autoplot(missingest,4)[[1]] + autoplot(missingest,4)[[2]]

```



```

# initial = mtarinipars(tsregime_obj = data_c, method = 'KUD',
#                      list_model = list(pars = list(l = 2,

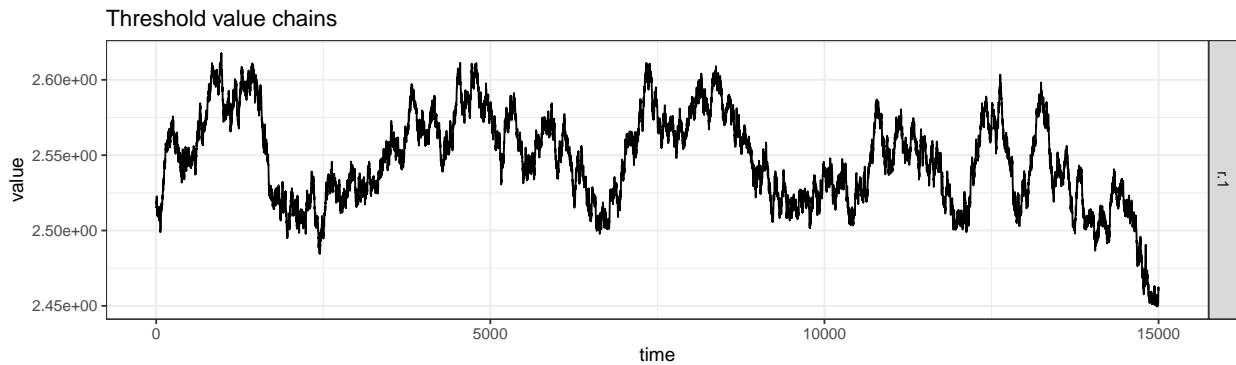
```

```

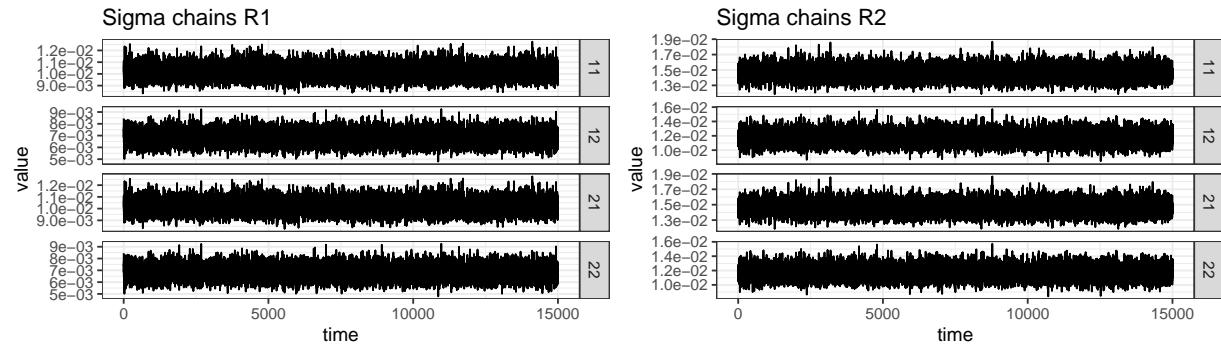
# orders = list(pj = c(5,5), dj = c(5,5)))
# estruc2 = mtarstr(ini_obj = initial, niter = 5000, chain = TRUE, burn = 1000, r_init = 2.6)
#estruc2$orders
#autoplot(estruc2,4)

# initial = mtarinipars(tsregime_obj = data_c, list_model = list(pars = list(l = 2, orders = list(pj = c(5,5), dj = c(5,5))), 
# ns2 = mtarns(ini_obj = initial, niter = 15000, chain = TRUE, burn = 5000, r_init = 2.6)
autoplot(ns2,1)

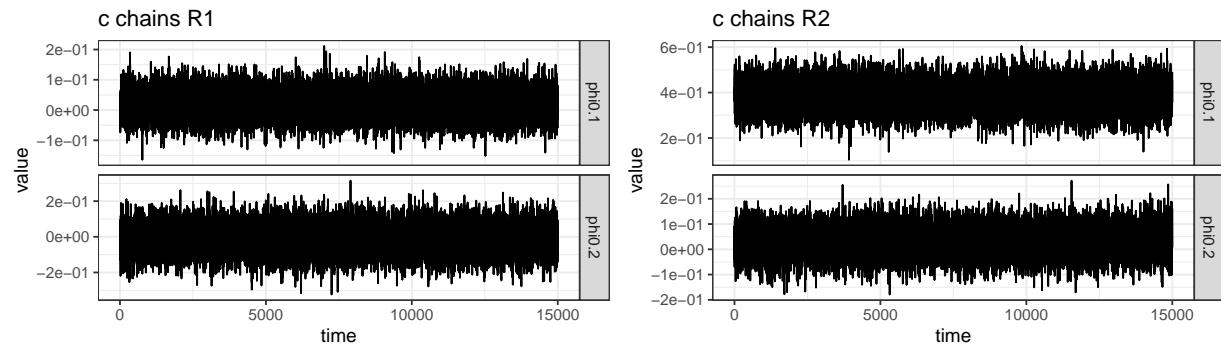
```



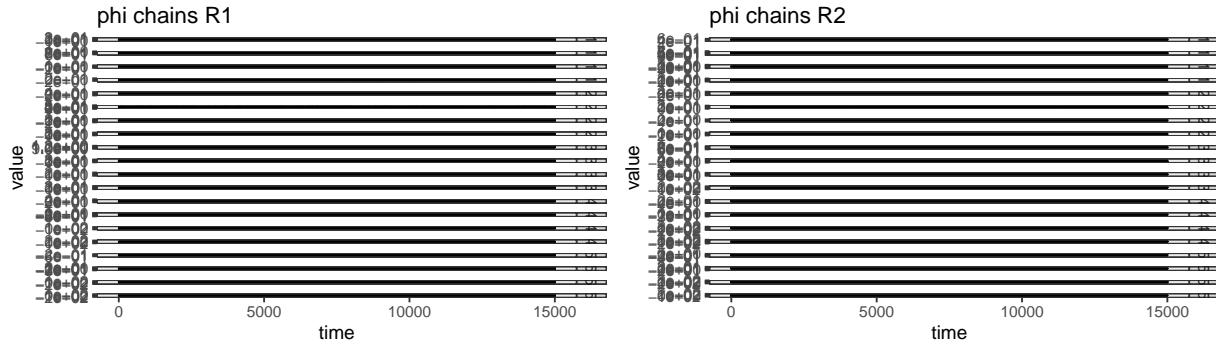
```
autoplot(ns2,2)$R1+autoplot(ns2,2)$R2
```



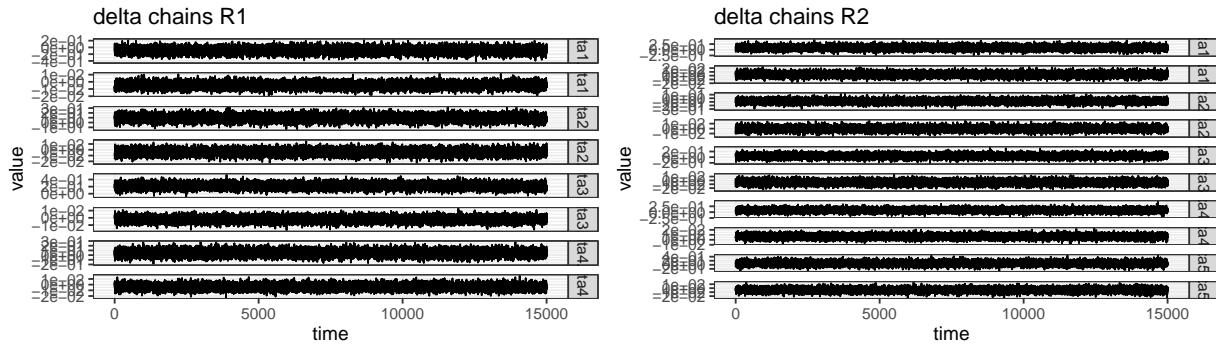
```
autoplot(ns2,3)$R1$c+autoplot(ns2,3)$R2$c
```



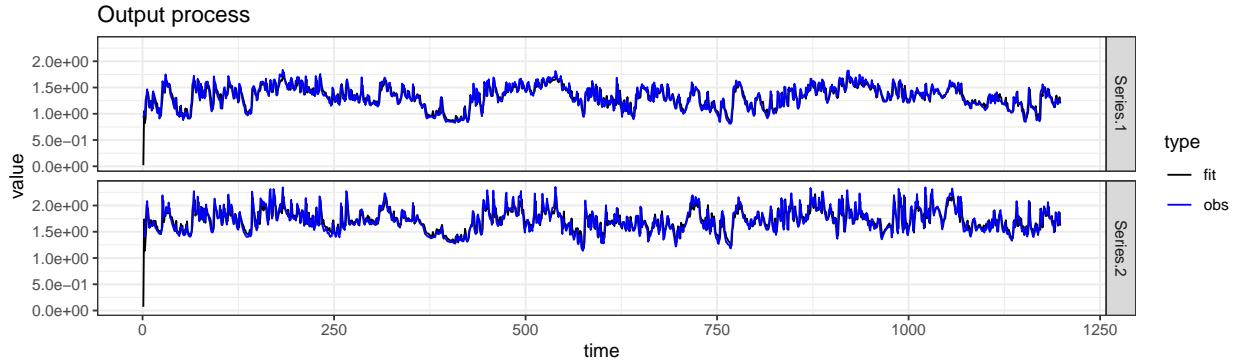
```
autoplot(ns2,3)$R1$phi+autoplot(ns2,3)$R2$phi
```



```
autoplot(ns2,3)$R1$delta+autoplot(ns2,3)$R2$delta
```



```
autoplot(ns2,5)
```



```
#pred1 = mtarforecast(ns2,h = 10)
pred1$forecast$estim
```

```
## $Yt
##      lower limit 2.5%       mean upper limit 97.5%     FNDP
## 1200.1    2.82644647  3.45470323    4.0693048 1.898453
## 1200.2   -0.10040909  0.17532251    0.4502183 1.898453
## 1201.1    1.35819270  2.03207597    2.7033500 1.860717
## 1201.2   -0.25189363  0.17031936    0.6064289 1.860717
## 1202.1    1.14496819  1.72169348    2.2929976 1.897720
## 1202.2   -0.21804761  0.24986784    0.7197960 1.897720
## 1203.1    3.97159859  4.40987018    4.8618904 1.910525
## 1203.2   -0.11280745  0.17109252    0.4596773 1.910525
## 1204.1    2.49106817  3.18607876    3.8794081 1.926146
```

```

## 1204.2      -0.48999243 -0.04297338      0.4056911 1.926146
## 1205.1      0.78056975  1.47831827      2.1803403 1.907639
## 1205.2     -0.40687854  0.14861441      0.7071511 1.907639
## 1206.1      4.47424293  5.02676298      5.5744084 1.871195
## 1206.2     -0.08191974  0.25991463      0.6057236 1.871195
## 1207.1      3.41141532  4.34747826      5.2951206 1.906519
## 1207.2     -0.61752360  -0.10922778      0.4009244 1.906519
## 1208.1      2.23257810  3.38603575      4.5337105 1.916673
## 1208.2     -2.06822489  -0.81768260      0.4357648 1.916673
## 1209.1      4.52211013  5.21629594      5.9292391 1.919916
## 1209.2     -0.17478916  0.33066470      0.8450098 1.919916
##
## $Zt
##      lower limit 2.5%      mean upper limit 97.5%      FNDP
## 1200.1     -0.6466044 2.004539      4.706301 1.898453
## 1201.1     -0.6777297 1.955306      4.621715 1.860717
## 1202.1     -1.0290016 1.672624      4.368263 1.897720
## 1203.1     -1.1337334 1.623128      4.332484 1.910525
## 1204.1     -0.3358724 2.349589      5.056558 1.926146
## 1205.1     -0.4069809 2.309134      5.048625 1.907639
## 1206.1     -1.0726941 1.626381      4.290429 1.871195
## 1207.1     -0.7708655 1.968999      4.703153 1.906519
## 1208.1     -0.5446227 2.143045      4.830108 1.916673
## 1209.1     0.1783136 2.905230      5.601415 1.919916

```