



Departamento de Economía

**Facultad de Ciencias
Administrativas y Económicas**



Economics Lecture Notes

**Introducción (empleando R) a los modelos univariados no lineales
de series de tiempo: Modelo TAR**

Julio César Alonso Cifuentes

Icesi ECONOMICS LN No. 17
Diciembre 2022

Introducción (empleando R) a los modelos univariados no lineales de series de tiempo: Modelo TAR

Julio César Alonso Cifuentes

Icesi
ECONOMICS LN
No.17
Diciembre 2022

Universidad Icesi

Editor:

Carlos Giovanni González Espitia
Profesor Titular, Universidad Icesi
cggonzalez@icesi.edu.co

Asistente editorial:

Valentina Perdomo

Gestión Editorial

Departamento de Economía - Universidad Icesi

Apuntes de Economía [Icesi Economics Lecture Notes] es una publicación del Departamento de Economía de la Universidad Icesi, cuya finalidad es divulgar las notas de clase de los docentes y brindar material didáctico para la instrucción en el área económica a diferentes niveles. El contenido, errores u omisiones de este documento son responsabilidad absoluta de los autores y no compromete al Editor ni a la Universidad Icesi.

www.icesi.edu.co

Tel: 5552334. Fax: 5551441

Calle 18 # 122-135 Cali, Valle del Cauca, Colombia

Introducción (empleando R) a los modelos
univariados no lineales de series de tiempo:
Modelo TAR
Notas de clase.

Julio César Alonso ^{*}
jcalonso@icesi.edu.co
CIENFI - Departamento de Economía
Universidad Icesi
Cali - Colombia

23 de diciembre de 2022



1. Objetivos de Aprendizaje

Al finalizar la lectura de este documento el estudiante estará en capacidad de:

- Explicar en sus propias palabras qué es un modelo Autoregresivo con umbral (*Threshold Autoregressive Models*), más conocido como **TAR**, estimarlo y generar pronósticos con este modelo en R.
- Explicar en sus propias palabras qué es un modelo *Self- Exciting TAR* (**SETAR**), estimarlo y generar pronósticos con este modelo en R.

^{*}Grupo de Investigación Economía, Métodos Cuantitativos y Políticas Públicas

- Explicar en sus propias palabras la diferencia entre un modelo **SETAR** de dos y tres regímenes.
- Explicar en sus propias palabras qué es un modelo *Momentum TAR* (**M-TAR**), estimarlo y generar pronósticos con este modelo en R.
- Explicar en sus propias palabras qué es un modelo *Logistic Smooth Transition Autoregressive* (**LSTAR**), estimarlo y generar pronósticos con este modelo en R.
- Explicar en sus propias palabras qué es un modelo *Additive Non-lineal Autoregressive models* (**AAR**), estimarlo y generar pronósticos con este modelo en R.
- Explicar en sus propias palabras qué es un modelo de Red Neuronal no-lineal Autoregresiva *nonlinear autoregressive neural network* (**NARNN**), estimarlo y generar pronósticos con este modelo en R.

2. Introducción

Hasta el momento hemos discutido modelos que suponen un comportamiento lineal. Por ejemplo, para modelos autoregresivos de orden 1 (AR(1)) es igual y de la misma magnitud la respuesta de la variable ante un cambio positivo o negativo de una unidad de la variable rezagada. Pero el mundo no necesariamente es lineal. Se podría esperar que las variables económicas reaccionen de manera diferente ante variaciones positivas o negativas. O se puede esperar una reacción diferente cuando una variable sobrepasa un determinado umbral. En este documento nos concentraremos en los modelos no-lineales; en especial el modelo Autoregresivo con umbral **TAR** (por su sigla del inglés *Threshold Autoregressive Models*) y otros modelos derivados de este. Así mismo, discutiremos cómo estimarlos en R.

3. El modelo TAR de dos regímenes

En general un modelo **TAR** (de dos regímenes) corresponde a un modelo en el cuál el valor actual de la variable y_t es función de un proceso autoregresivo de orden (p_1) cuando la variable de umbral z_t rezaga es igual o superior a un umbral (r) o a un proceso autoregresivo de orden (p_2) cuando la variable z_t es mayor al umbral. Es decir,

$$y_t = \begin{cases} \phi_1 + \phi_{1,1}y_{t-d} + \dots + \phi_{1,p_1}y_{t-(p_1-1)d} + \sigma_1\varepsilon_t & z_t \leq r \\ \phi_2 + \phi_{2,1}y_{t-d} + \dots + \phi_{2,p_2}y_{t-(p_2-1)d} + \sigma_2\varepsilon_t & z_t > r \end{cases} \quad (1)$$

donde z_t puede ser cualquier variable. Si bien las condiciones para que este proceso sea estacionario no son sencillas, se puede demostrar fácilmente que una condición necesaria es que la suma del valor absoluto de los coeficientes autoregresivos de cada régimen sea menor a uno.

Si se emplea como variable de umbral la misma serie ($z_t = y_{t-\delta d}$) y se asume que $d = 1$, entonces el modelo **TAR** se convierte en un modelo **SETAR** (Self-Exciting TAR) con retraso de umbral δ (threshold delay)¹. Es decir, el modelo **SETAR** corresponde a:

$$y_t = \begin{cases} \phi_1 + \phi_{1,1}y_{t-1} + \dots + \phi_{1,p_1}y_{t-(p_1-1)} + \sigma_1\varepsilon_t & y_{t-\delta} \leq r \\ \phi_2 + \phi_{2,1}y_{t-1} + \dots + \phi_{2,p_2}y_{t-(p_2-1)} + \sigma_2\varepsilon_t & y_{t-\delta} > r \end{cases} \quad (2)$$

Una forma más compacta de escribir este modelo es empleando una función indicador. Es decir, sea $I(y_{t-1} \leq r)$ una función que toma el valor de uno si $y_{t-1} \leq r$ y cero en caso contrario. Empleando esta función indicador, el anterior modelo SETAR se puede escribir como:

$$y_t = (\phi_1 + \phi_{1,1}y_{t-1} + \dots + \phi_{1,p_1}y_{t-(p_1-1)} + \sigma_1\varepsilon_t) I(y_{t-1} \leq r) + (\phi_2 + \phi_{2,1}y_{t-1} + \dots + \phi_{2,p_2}y_{t-(p_2-1)} + \sigma_2\varepsilon_t) (1 - I(y_{t-1} \leq r))$$

Este modelo tiene varias características importantes. Por ejemplo, en cada uno de los regímenes el proceso autoregresivo puede ser de un orden diferente. La varianza de y_t puede ser diferente en cada régimen. Otra de las características interesantes de este modelo, es que en caso de ser un proceso estacionario, para cada régimen el proceso tiende a revertir a una media diferente. Para entender este último punto mejor, supongamos un modelo **SETAR** de dos regímenes con $p_1 = p_2 = 1$ y $\delta = 1$. En este caso, el modelo **SETAR** se puede reescribir como:

$$y_t = \begin{cases} \phi_1 + \phi_{1,1}y_{t-1} + \sigma_1\varepsilon_t & y_{t-1} \leq r \\ \phi_2 + \phi_{2,1}y_{t-1} + \sigma_2\varepsilon_t & y_{t-1} > r \end{cases} \quad (3)$$

Es fácil mostrar que:

$$E[y_t] = \begin{cases} \frac{\phi_1}{1-\phi_{1,1}} & y_{t-1} \leq r \\ \frac{\phi_2}{1-\phi_{2,1}} & y_{t-1} > r \end{cases}$$

En la Figura 1 se puede observar cómo la serie puede revertir a una media diferente y con velocidades de ajustes diferentes dependiendo del valor que toma la variable en el periodo anterior. Para aclarar un poco más este resultado teórico, consideremos una serie simulada del siguiente DGP (de la sigla del inglés proceso generador de datos):

$$y_t = \begin{cases} -2 + 0,35y_{t-1} + \varepsilon_t & y_{t-1} \leq 0 \\ 2 + 0,5y_{t-1} + \varepsilon_t & y_{t-1} > 0 \end{cases}$$

Una simulación de 5000 observaciones se presentan en la Figura 2. Como se puede observar, en este caso hay dos regímenes claramente marcados en la serie. Las primeras realizaciones de esta serie son negativas, y la serie se mantiene en este régimen. Pero una vez la serie sobrepasa el umbral de cero, la serie cambia de régimen y se mantiene en este nuevo régimen. Una característica interesante

¹Este parámetro también es conocido como el parámetro de retraso.

de este tipo de series generadas por un DGP **SETAR** es la presencia de colas pesadas. Colas que son producidas por la existencia de dos polos de atracción de la series, gracias a la existencia de los dos regímenes. En la Figura 3 se puede observar un gráfico q-q para esta serie simulada. Del gráfico se observa claramente la existencia de colas pesadas.

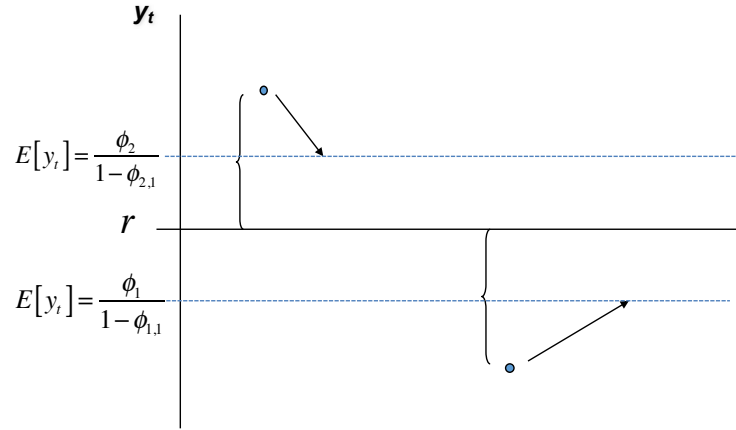


Figura 1: Esquema de un Modelo **SETAR**

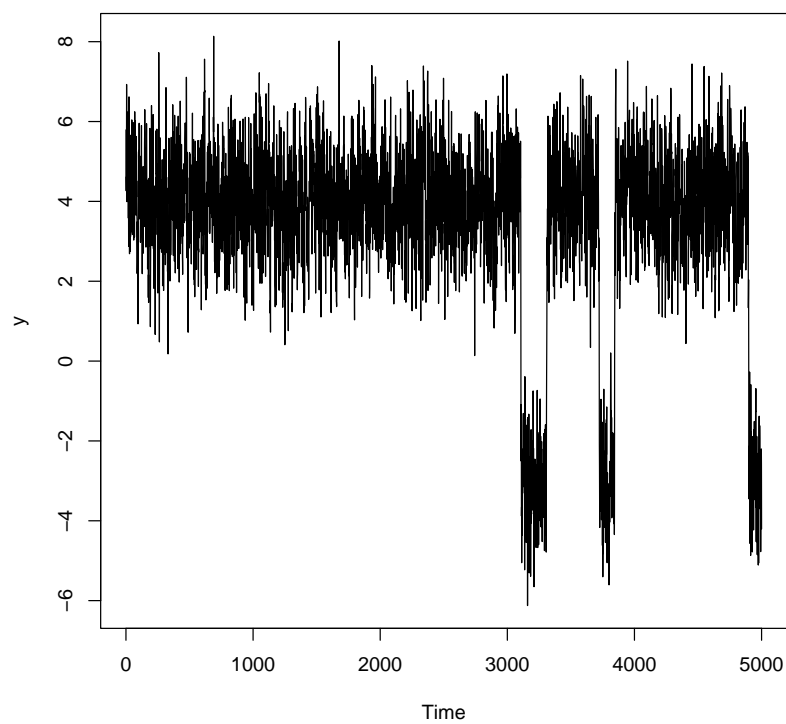


Figura 2: Serie simulada para un modelo SETAR $y_t = (-2 + 0,35y_{t-1} + \varepsilon_t) I(y_{t-1} \leq 0) + (2 + 0,5y_{t-1} + \varepsilon_t) (1 - I(y_{t-1} \leq 0))$

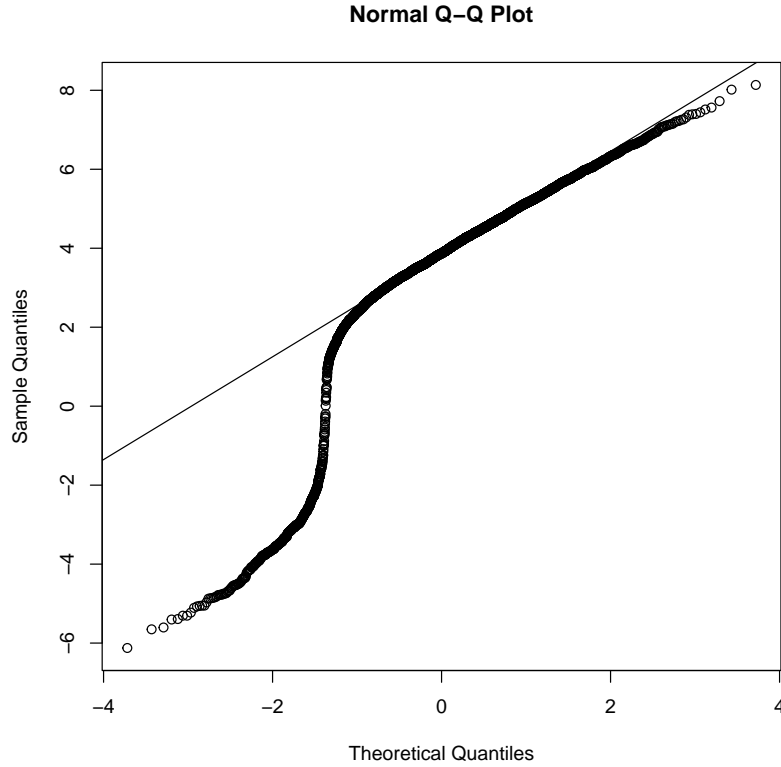


Figura 3: Gráfico qq de la serie simulada del modelo SETAR $y_t = (-2 + 0,35y_{t-1} + \varepsilon_t) I(y_{t-1} \leq 0) + (2 + 0,5y_{t-1} + \varepsilon_t) (1 - I(y_{t-1} \leq 0))$

Por otro lado, es importante no llevarse una falsa idea de que estos modelos **SETAR** son fácilmente identificables a ojo. Consideremos una simulación a partir del siguiente DGP:

$$y_t = \begin{cases} 0,35y_{t-1} + \varepsilon_t & y_{t-1} \leq 0 \\ 0,5y_{t-1} + \varepsilon_t & y_{t-1} > 0 \end{cases}$$

En este caso una simulación de 5000 realizaciones se puede observar en la Figura 4 y el correspondiente gráfico q-q se reporta en la Figura 5. En este caso, si bien es difícil diferenciar a partir de la Figura 4 la existencia de dos regímenes, la serie sigue presentando colas pesadas.

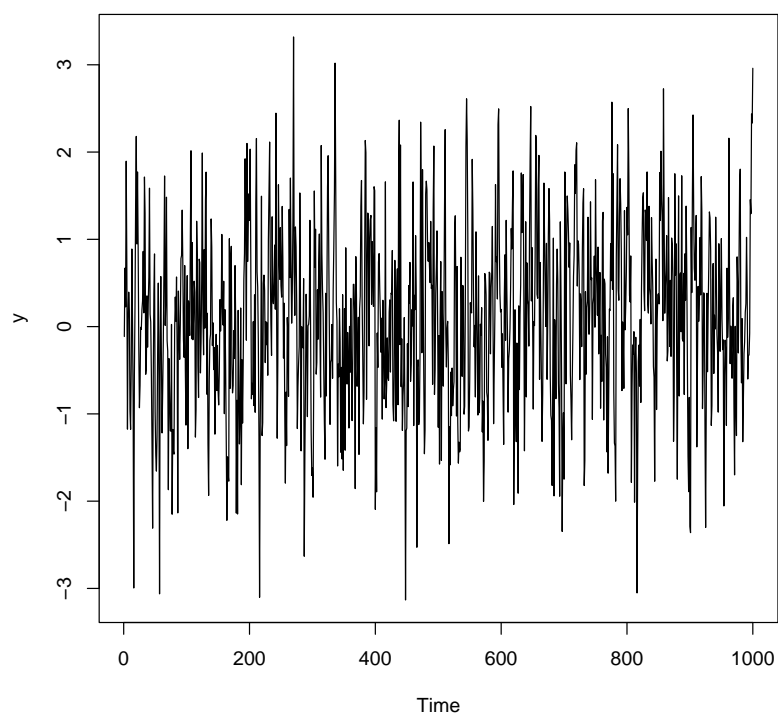


Figura 4: Serie simulada para un modelo SETAR $y_t = (0,35y_{t-1} + \varepsilon_t) I(y_{t-1} \leq 0) + (0,5y_{t-1} + \varepsilon_t) (1 - I(y_{t-1} \leq 0))$

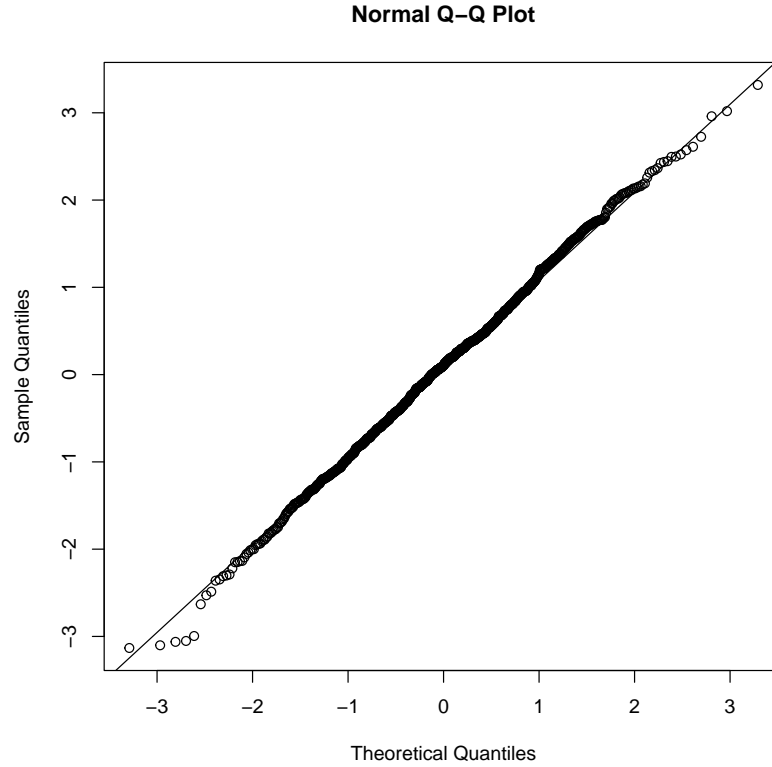
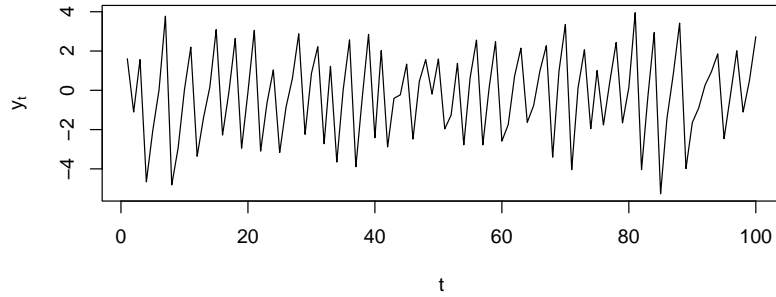


Figura 5: Gráfico qq de la serie simulada del modelo SETAR $y_t = (0,35y_{t-1} + \varepsilon_t) I(y_{t-1} \leq 0) + (0,5y_{t-1} + \varepsilon_t) (1 - I(y_{t-1} \leq 0))$

Las anteriores simulaciones son un ejemplo de lo difícil que puede resultar para un investigador diferenciar una serie que tiene un comportamiento lineal con una de comportamiento no lineal. Antes de pasar a pruebas formales que permitan identificar la existencia o no de un comportamiento no lineal, vemos un último ejemplo. En la Figura 6 se presentan simulaciones para un modelo ARMA(1,1) y un modelo TAR con dos regímenes y un término autoregresivo en cada uno de estos regímenes. Las dos series presentan un comportamiento relativamente similar, no obstante fueron generadas a partir de dos mundos totalmente diferentes.

Modelo SETAR con $\delta = 1$, $\phi_{10} = 0$, $\phi_{11} = 0.5$, $\phi_{20} = 0$, $\phi_{21} = -1.8$, $\sigma_1 = 1$ y $\sigma_2 =$



Modelo ARMA(1,1) $y_t = -0.75y_{t-1} + 0.25\varepsilon_{t-1} + \varepsilon_t$

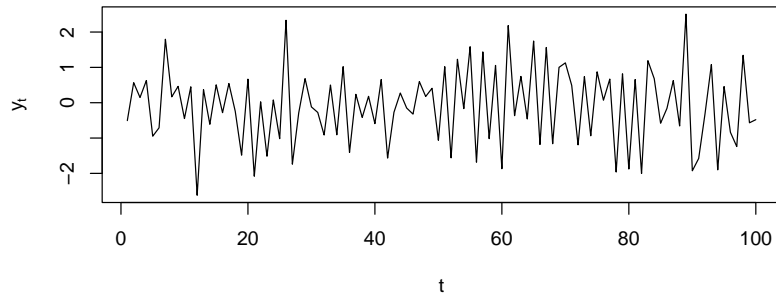


Figura 6: Series simuladas de un modelo TAR y un modelo ARMA(1,1)

4. Pruebas para detectar la no linealidad

En esta sección presentaremos diferentes aproximaciones formales e informales para determinar si existe un comportamiento no lineal o no en una serie de tiempo. Para ejemplificar cada uno de los procedimientos, emplearemos las dos series simuladas que se reportan en la Figura 6. La serie proveniente del modelo $ARMA(1,1)$ fue creada empleando la función `arima.sim` del paquete base de R.

```
set.seed(99)
x= arima.sim(n=100, list(ma=0.1, ar=-0.75), innov=rnorm(100))
```

La anterior línea de código crea una muestra de tamaño 100 del siguiente DGP:

$$x_t = -0.75x_{t-1} + 0.1\varepsilon_{t-1} + \varepsilon_t \quad (4)$$

donde ε_t sigue una distribución estándar normal.

Por otro lado, la muestra del modelo SETAR fue creado empleando la función `tar.sim` del paquete **TSA** (Chan y Ripley, 2012) . Esta serie fue creada empleando la siguiente línea.

```
library(TSA)
set.seed(99)
y= tar.sim(n=100,Phi1=c(2.,0.6),Phi2=c(-2,-0.4),
          p=1,d=1,sigma1=1,thd=1,sigma2=1)$y
```

Esta función crea 100 realizaciones del siguiente DGP:

$$y_t = \begin{cases} 2 + 0,6y_{t-1} + \varepsilon_t & y_{t-1} \leq 1 \\ -2 - 0,4y_{t-1} + \varepsilon_t & y_{t-1} > 1 \end{cases} \quad (5)$$

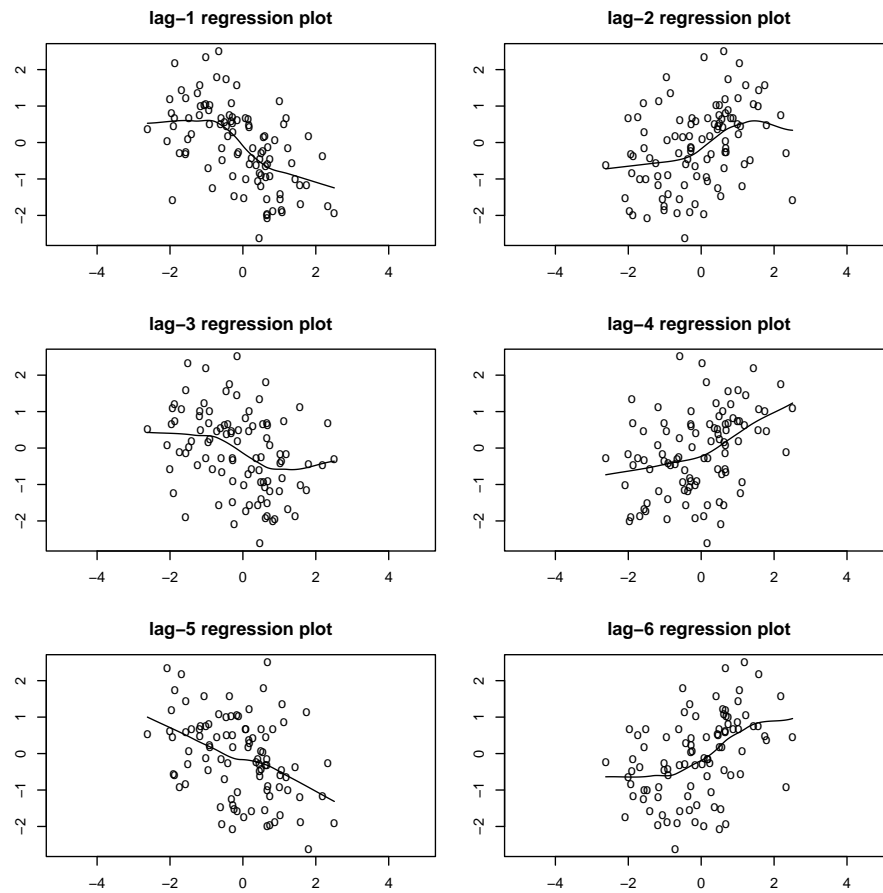
donde ε_t sigue una distribución estándar normal.

4.1. Aproximación gráfica: gráfico de rezagos

Una primera aproximación para determinar la existencia de linealidades o no es graficar la serie bajo estudio y_t con respecto a diferentes rezagos de esta (y_{t-s}). Un gráfico como este puede ser acompañado de la estimación de una regresión no paramétrica que permita determinar la existencia de patrones no lineales entre las variables. Este tipo de gráficos se pueden realizar con medio de la función `lagplot` del paquete **TSA**.

Empleemos esta función en las dos series generadas anteriormente. Por ejemplo, para el caso de la serie simulada a partir de (4) el correspondiente gráfico de rezagos se puede obtener de la siguiente manera:

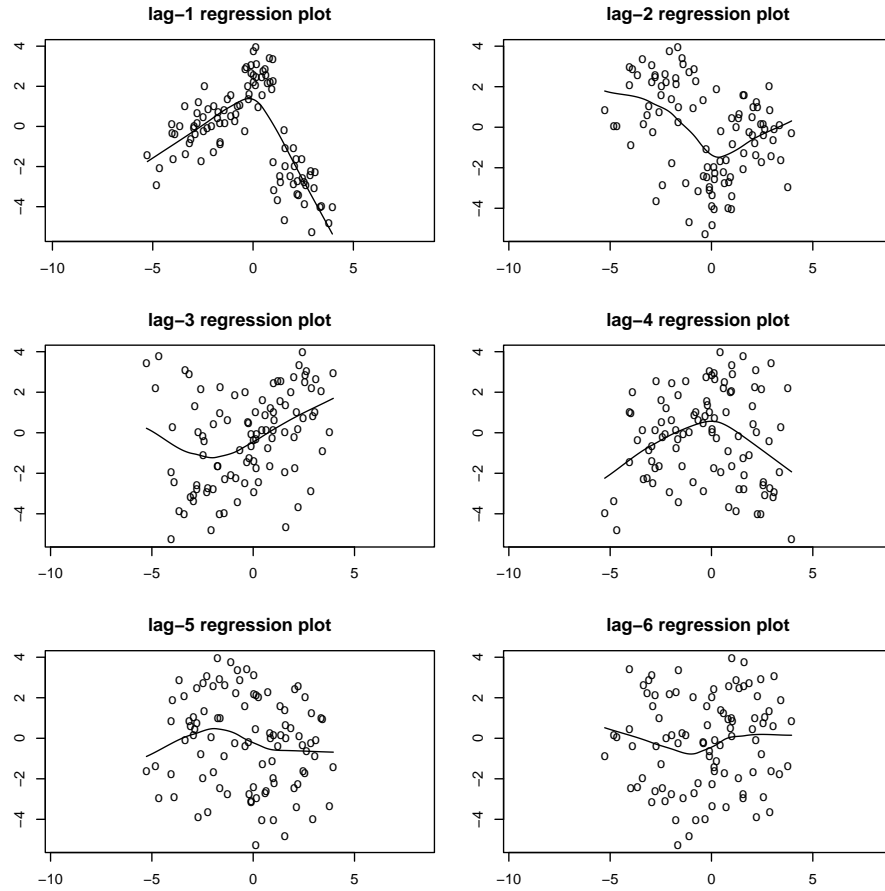
```
library(locfit)
lagplot(x)
```



En el primer panel de la anterior Figura se puede observar la relación negativa y muy cerca a ser lineal entre la serie y su primer rezago. Esto es resultado del relativamente alto coeficiente autoregresivo y de su signo negativo. Para la relación con los demás rezagos no se puede evidenciar una relación (línea de regresión no paramétrica relativamente plana).

Y para el caso de la serie SETAR generada a partir de (4) se tendrá que:

```
lagplot(y)
```



Esta última figura muestra una relación no lineal entre la serie y su primer rezago. Para los demás rezagos no se observa una relación. Claramente, estos gráficos no permiten generar una conclusión clara sobre la linealidad o no de las series. Pero los gráficos sí pueden ser parte de un análisis que se complemente con pruebas formales, como la que se discuten adelante.

4.2. Prueba de McLeod-Li (1983)

La prueba de McLeod y Li (1983) fue inicialmente diseñada para detectar la presencia de un comportamiento ARCH en los errores de un modelo ARMA. La idea inicial de los autores era realizar una prueba similar a la de Ljung-Box para detectar autocorrelación entre los errores. Pero en vez de determinar la significancia conjunta de las autocorrelaciones entre los errores separados por diferentes rezagos, la idea inicial era detectar las correlaciones entre los errores al cuadrado de diferentes rezagos. Esta prueba permitía detectar el comportamiento no lineal de los residuos. Por otro lado, se ha demostrado que si la prueba de McLeod y Li (1983) se aplica no a los residuos sino a una serie y_t ,

esta prueba tendrá un poder relativamente alto para detectar diferentes formas de no-linealidad en los datos.

Es decir, si se define $\hat{\rho}_i$ como el coeficiente de correlación estimado entre y_t^2 y y_{t-i}^2 , el estadístico Q de Ljung-Box correspondiente a la prueba de McLeod y Li (1983) de no linealidad en las series para s rezagos será:

$$Q = T(T+2) \sum_{i=1}^s \frac{\hat{\rho}_i}{T-i} \quad (6)$$

El estadístico Q tiene sigue una distribución asintótica χ^2 con s grados de libertad bajo la nula de que la serie no presenta autocorrelación. En otras palabras, este estadístico permite probar la nula de la existencia de linealidad versus la alterna de no linealidad. Es importante aclarar que rechazar la hipótesis nula no permite determinar el tipo de no linealidad que presenta la serie. Esta prueba típicamente se realiza para diferentes rezagos para determinar la existencia o no de linealidad en los datos.

La prueba de McLeod y Li (1983) puede estimarse en *R* empleando la función `McLeod.Li.test` del paquete **TSA**. Esta función permite realizar la prueba para diferentes rezagos al mismo tiempo y reporta los correspondientes valores *p* para cada rezago. Solamente es necesario incluir el argumento *y* que corresponde a la serie que será probada por linealidad. Adicionalmente, si se incluye el argumento *plot* y se escoge la opción "FALSE", no se presentará el gráfico y se guardarán los valores *p* en un objeto. Por ejemplo, la siguiente línea de código permite realizar la prueba sobre la serie *ARMA(1,1)* y obtener los *p* valores para la prueba de los primeros 20 rezagos.

```
ML<-McLeod.Li.test(y=x, plot = FALSE)
ML

## $p.values
## [1] 0.5131858 0.7928646 0.4185121 0.5865697 0.4414807 0.4748009 0.5495988
## [8] 0.6546382 0.5926614 0.5792716 0.6667507 0.6979690 0.7522939 0.6921652
## [15] 0.7230818 0.7579247 0.7788288 0.5635777 0.6294041 0.6186141
```

Estos *p* valores no permiten rechazar la nula de linealidad a todos los niveles de confianza convencionales. Si no se escribe el argumento *plot*, se obtiene una versión gráfica de los resultados como se muestra en la Figura 7.

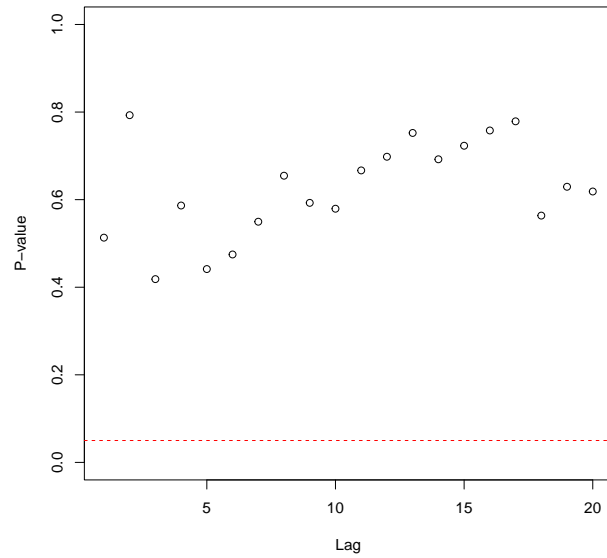


Figura 7: Prueba McLeod-Li (1983) para la serie ARMA simulada

Para el caso de la serie SETAR simulada se tienen los siguientes resultados:

```
ML<-McLeod.Li.test(y=y, plot = FALSE)
ML

## $p.values
## [1] 8.065170e-01 1.414262e-02 9.722010e-05 1.439216e-05 1.237823e-05
## [6] 1.558032e-05 3.820823e-05 7.917953e-05 5.531763e-05 9.243249e-05
## [11] 1.605586e-04 2.703442e-04 4.140321e-04 1.996696e-04 3.490358e-04
## [16] 2.349795e-04 2.217263e-04 3.035543e-04 3.488253e-04 5.164184e-04
```

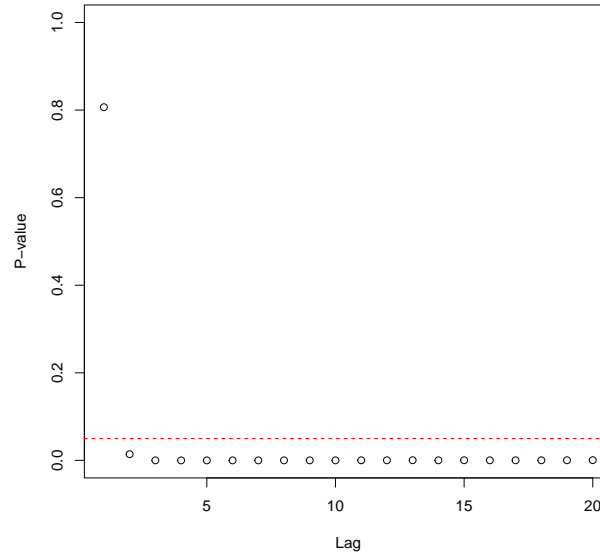



Figura 8: Prueba McLeod-Li (1983) para la serie SETAR simulada

Estos resultados muestran que para la mayoría de rezagos es posible rechazar la nula de linealidad de la serie. Es decir, parece que la serie no es lineal.

4.3. Prueba de Keenan (1985)

Keenan(1985) sugiere una prueba de no linealidad cuya hipótesis nula es que la serie sigue un proceso $AR(p)$. La prueba implica los siguientes pasos:

1. Seleccionar el orden p del posible modelo lineal $AR(p)$. Esto se puede realizar por medio de un criterio de información; como por ejemplo AIC.
2. Correr una regresión de y_t en función de los p rezagos de y_t . Es decir, se ajusta una regresión lineal de y_t en función de $(1, y_{t-1}, \dots, y_{t-p})$.
3. Encontrar los valores ajustados de y_t (\hat{y}_t) y calcularlos residuos ($\hat{a}_t = y_t - \hat{y}_t$).
4. Calcular la suma cuadrada de los residuos (SSR por su sigla en inglés)
5. Correr una regresión de y_t^2 en función de los p rezagos de y_t y obtenga los correspondientes residuos ($\hat{\zeta}_t$).

6. Calcular

$$\hat{\eta}_t = \frac{\sum_{t=p+1}^T \hat{a}_t \hat{\zeta}_t}{\sum_{t=p+1}^T \hat{a}_t \hat{\zeta}_t^2}$$

7. Construir el estadístico de prueba

$$\hat{F} = \frac{(T - 2p - 2) \hat{\eta}_t^2}{(SSR - \hat{\eta}_t^2)}$$

Keenan(1985) demuestra que bajo la hipótesis nula de linealidad (modelo AR(p)) y suponiendo que a_t es un proceso i.i.d Gausiano seguirá una distribución F con 1 grado de libertad en el numerado y $(T - 2p - 2)$ en el denominador. En este caso, rechazar la hipótesis nula implicará que la serie sigue un proceso no lineal.

La prueba de Keenan (1985) puede estimarse en R empleando la función `Keenan.test()` del paquete **TSA**. Esta función realiza todos los pasos descritos arriba automáticamente, solo es necesario especificar cual es la serie que será evaluada. Por ejemplo, para nuestra serie simulada a partir del modelo ARMA(1,1), la prueba de Keenan (1985) puede efectuarse de la siguiente manera:

```
Keenan.test(x)

## $test.stat
## [1] 0.1766519
##
## $p.value
## [1] 0.6753148
##
## $order
## [1] 6
```

El p-valor de la prueba implica que no es posible rechazar la hipótesis nula de que la serie sigue un proceso lineal. De manera similar, para la serie simulada desde un modelo SETAR se tiene que:

```
Keenan.test(y)

## $test.stat
## [1] 35.60251
##
## $p.value
## [1] 4.243602e-08
##
## $order
## [1] 2
```

En este caso, se rechaza la hipótesis de que el proceso es lineal. Es decir, existe evidencia de que la serie es no lineal.

4.4. Prueba de Tsay (1986)

Tsay (1986) diseña una prueba que mejora el poder de la prueba de Keenan (1985) permitiendo una estructura no lineal en las regresiones auxiliares de la prueba de Keenan (1985). En vez de emplear y_t^2 en el paso 5, Tsay (1986) sugiere emplear como variable dependiente $y_{t-i}y_{t-j}$ (para $i, j = 1, \dots, p$). En este caso, el correspondiente estadístico (\tilde{F}) seguirá una distribución asintótica F con m grados de libertad en el numerador y $T - m - p - 1$ en el denominador, donde $m = p(p - 1)/2$.

La función `Tsay.test()` del paquete **TSA** permite efectuar esta prueba de manera automática. Para nuestras dos series simuladas tenemos que:

```
Tsay.test(x)

## $test.stat
## [1] 1.181
##
## $p.value
## [1] 0.2963
##
## $order
## [1] 6

Tsay.test(y)

## $test.stat
## [1] 19.77
##
## $p.value
## [1] 5.624e-10
##
## $order
## [1] 2
```

Para la serie generada desde el DGP ARMA no es posible rechazar la nula de que un modelo lineal es mejor que uno no lineal con un 99% de confianza. Por otro lado, para la muestra generada por medio del modelo SETAR la hipótesis nula se puede rechazar. Por tanto, existe evidencia de no linealidad en esta segunda serie.

4.5. Razón de máxima verosimilitud para no linealidad de umbral

Chan y Tong (1986)² proponen una prueba mucho más específica que las que hemos considerado hasta ahora. En este caso ellos construyen una prueba cuya hipótesis nula es que la muestra proviene de un proceso $AR(p)$ frente a la alterna que proviene de un modelo SETAR de dos regímenes como el siguiente:

$$y_t = \begin{cases} \phi_1 + \phi_{1,1}y_{t-1} + \sigma_1\varepsilon_t & y_{t-\delta} \leq r \\ \phi_2 + \phi_{2,1}y_{t-1} + \sigma_2\varepsilon_t & y_{t-\delta} > r \end{cases}$$

Chan y Tong (1986) construyen el siguiente estadístico de prueba

$$LR = \left(\frac{\hat{\sigma}_{SETAR}^2}{\hat{\sigma}_{AR}^2} \right)^{\frac{T-p+1}{2}}$$

donde $\hat{\sigma}_{SETAR}^2$ y $\hat{\sigma}_{AR}^2$ son las varianzas estimadas para el modelo SETAR y AR respectivamente. Chan (1991) sugiere una forma de aproximar la distribución de este estadístico de prueba y provee un algoritmo para calcular el correspondiente valor p. Dado que el retraso de umbral δ y el p son desconocidos, esta prueba se realiza para diferentes valores de estos dos parámetros.

La prueba de Chan y Tong (1986) puede estimarse en *R* empleando la función `tlrt()` del paquete **TSA**. Para la muestra simulada a partir del DGP *ARMA* y para $p = 1, 2$ tenemos que:

```
# p=1
pvaluem.x.p1=NULL

for (d in 1:5) {
  res=tlrt(x,p=1,d=d,a=0.25,b=0.75)
  pvaluem.x.p1= cbind( pvaluem.x.p1,
                      c(d,res$test.statistic,res$p.value))
}

rownames(pvaluem.x.p1)=c('d','test statistic','p-value')
round(pvaluem.x.p1,3)

##           [,1] [,2] [,3] [,4] [,5]
## d           1.000 2.000 3.000 4.000 5.000
## test statistic 14.139 7.648 11.426 10.901 8.914
## p-value       0.016 0.172 0.046 0.056 0.117

# p=2
pvaluem.x.p2=NULL
```

²Esta prueba también se conoce como $LR - TAR$.

```

for (d in 1:5) {
  res=tlrt(x,p=2,d=d,a=0.25,b=0.75)
  pvaluem.x.p2= cbind( pvaluem.x.p2,
                      c(d,res$test.statistic,res$p.value))
}

rownames(pvaluem.x.p2)=c('d','test statistic','p-value')
round(pvaluem.x.p2,3)

##           [,1] [,2] [,3] [,4] [,5]
## d           1.000 2.000 3.000 4.000 5.000
## test statistic 16.529 8.048 10.338 12.284 8.170
## p-value      0.016 0.248 0.137 0.073 0.245

```

Y para el caso de la muestra del modelo SETAR se tiene que:

```

# p=1
pvaluem.y.p1=NULL

for (d in 1:5) {
  res=tlrt(y,p=1,d=d,a=0.25,b=0.75)
  pvaluem.y.p1= cbind( pvaluem.y.p1,
                      c(d,res$test.statistic,res$p.value))
}

rownames(pvaluem.y.p1)=c('d','test statistic','p-value')
round(pvaluem.y.p1,3)

##           [,1] [,2] [,3] [,4] [,5]
## d           1.000 2.000 3.000 4.000 5.000
## test statistic 423.885 59.358 22.673 9.091 15.081
## p-value      0.000 0.000 0.000 0.106 0.011

# p=2
pvaluem.y.p2=NULL

for (d in 1:5) {
  res=tlrt(y,p=2,d=d,a=0.25,b=0.75)
  pvaluem.y.p2= cbind( pvaluem.y.p2,
                      c(d,res$test.statistic,res$p.value))
}

rownames(pvaluem.y.p2)=c('d','test statistic','p-value')
round(pvaluem.y.p2,3)

```

```
##           [,1] [,2] [,3] [,4] [,5]
## d           1.000 2.000 3.000 4.000 5.000
## test statistic 319.807 20.039 19.989 15.304 8.157
## p-value       0.000 0.004 0.004 0.025 0.245
```

De acuerdo a esta prueba, existe evidencia en favor de la no linealidad de la serie generada por el modelo **SETAR**. Y con un 99% de confianza no se rechaza la hipótesis de linealidad para la serie generada por el modelo *ARMA*.

5. Estimación de los modelos SETAR y TAR

Para estimar un modelo **TAR** o su caso especial **SETAR** se debe reconocer que para un régimen determinado (r y z_t) el modelo es lineal. Esto hace que los vectores $\Phi_1 = (\phi_1, \phi_{1,1}, \dots, \phi_{1,p_1})$ y $\Phi_2 = (\phi_2, \phi_{2,1}, \dots, \phi_{2,p_2})$ puedan ser estimados por medio de Mínimos Cuadrados Condicionales (CLS por su sigla en inglés). Los errores estándar de los elementos de Φ_1 y Φ_2 se pueden calcular de la manera como se hace en las regresiones múltiples.

La estimación de modelos **SETAR** en R se puede realizar por medio de al menos dos paquetes: **TSA** y **tsDyn** (Di Narzo, Aznarte y Stigler (2009)). La función **tar** del paquete **TSA** permite estimar modelos **SETAR** de dos regímenes. Por otro lado, la función **setar** del paquete **tsDyn** es mucho más flexible. Esta última función permite estimar modelos **TAR** con un número de regímenes mayor a dos. Concentremos nuestra atención a esta última función por ser más flexible.

Empleando (3), el modelo **SETAR** para dos regímenes se puede escribir de una manera más general como:

$$y_t = \begin{cases} \phi_1 + \phi_{1,1}y_{t-d} + \dots + \phi_{1,p_1}y_{t-(p_1-1)d} + \sigma_1\varepsilon_t & z_t \leq r \\ \phi_2 + \phi_{2,1}y_{t-d} + \dots + \phi_{2,p_2}y_{t-(p_2-1)d} + \sigma_2\varepsilon_t & z_t > r \end{cases} \quad (7)$$

donde la variable de umbral puede estar definida como una combinación lineal de $m - 1$ rezagos de y_t . Es decir,

$$z_t = \beta_1 y_{t-1} + \dots + \beta_{m-1} y_{t-(m-1)\delta d} \quad (8)$$

La función **setar()** del paquete **tsDyn** permite estimar esta especificación. Los argumentos más relevantes de la función son:

- **x**: la series de tiempo. Es decir, y_t en (5) y (5).
- **m**: $m - 1$ es el número de rezagos de la variable que se emplearán para construir la variable de umbral. Por ejemplo, si $m = 2$ tendremos que (5) se transforma en $z_t = y_{t-\delta d}$. Esta es la opción más común para estimar un modelo **SETAR**.

- **d**: representa a d en (5). Los modelos **SETAR** que se presentaron con anterioridad en este documento corresponde al caso en que $d = 1$. Es más, este es el valor por defecto que tiene **d** en esta función. Por ejemplo, si $m = 2$ y $d = 1$ tendremos que (5) se transforma en $z_t = y_{t-\delta}$.
- **mL**: el orden autoregresivo del primer régimen o régimen bajo (Low en inglés y por eso la L). Esto corresponde a p_1 en (5). Si no se especifica este argumento, la función podrá tomar los rezagos del argumento **ML**. Si ninguno de los dos argumento se especifica (i.e. **mL** y **ML**) la función asignará por defecto a este argumento el valor del argumento **m**.
- **mM**: el orden autoregresivo del segundo régimen de un **SETAR** de tres regímenes o régimen medio (Middle en inglés y por eso la M). Este argumento solo se emplea si **nthresh=2**; es decir, si se establece un modelo con dos umbrales y por tanto tres regímenes. Este argumento funciona de igual manera que el argumento **mL**.
- **mH**: el orden autoregresivo del segundo régimen o régimen alto (High en inglés y por eso la H). Esto corresponde a p_2 en (5). Si no se especifica este argumento, la función podrá tomar los rezagos del argumento **MH**. Si ninguno de los dos argumentos se especifican (i.e. **hL** y **HL**) la función asignará por defecto a este argumento el valor del argumento **m**.
- **thDelay**: el umbral de retraso o parámetro de retraso. En (5) representa δ . Por ejemplo, si $m = 2$ y $d = 1$, $\delta = 1$ tendremos que (5) se transforma en $z_t = y_{t-1}$.
- **mTh**: los coeficientes de la combinación lineal de los rezagos que construyen la variable de umbral. Es decir corresponde a los β en (5). Si se emplea $m = 1$ este coeficiente se define automáticamente como uno.
- **thVar**: variable externa que se desea emplear como variable de umbral. En caso de emplearse este argumento, se estimará un modelo **TAR**. Si no se especifica una variable externa, se estimará un modelo **SETAR**. Es decir, la variable de umbral será la misma serie estudiada.
- **th**: valor o valores de los umbrales. Corresponde a r en (5). Si no se especifica este argumento, la función hará una búsqueda de grilla para encontrar el mejor valor del umbral.
- **include**: este argumento permite incluir regresores determinísticos en el modelo **SETAR**. Las opciones son `"const"`, `"trend"`, `"none"` y `"both"`. Es decir, se puede incluir solo un intercepto (esta es la opción predeterminada), una tendencia, ni tendencia ni intercepto y tendencia e intercepto.
- **model**: permite determinar si se desea estimar un **SETAR** ("**TAR**") o un **M-TAR** ("**MTAR**").

- **ML**: rezagos de la variable y_t para ser incluidos en el régimen bajo. Esto permite no tener que incluir todos los rezagos que se encuentran entre 1 y p_1 . Por ejemplo, si **ML=c(2,4)** esto implicará que se incluirán en el primer régimen (régimen bajo) únicamente el segundo y cuarto rezago de y_t . Si no se especifica este argumento se emplearán todos los rezagos especificados en **mL**.
- **MM**: rezagos de la variable y_t para ser incluidos en el régimen de medio. Solo aplica para modelos de tres regímenes.
- **MH**: rezagos de la variable y_t para ser incluidos en el régimen alto. Funciona de manera similar a **ML**.
- **nthresh**: el número de umbrales. Por defecto este argumento es uno. Es decir, un modelo con dos regímenes. Para un modelo de tres regímenes se debe tener **nthresh = 2**.
- **trim**: este parámetro permite determinar el porcentaje mínimo de observaciones que se conservará por régimen. Por defecto este parámetro está establecido en 0.15. Esto es lo que sugiere la mayoría de la literatura empírica.
- **type**: este argumento permite determinar cómo se tratarán las series en el modelo **SETAR**. Se pueden emplear la serie original en niveles (lo preestablecido por la función) si **type="level"**, en primeras diferencias ($\Delta y_t = y_t - y_{t-1}$) si **type = "diff"** o una mezcla como la prueba ADF si **type = "ADF"**. La opción similar a la prueba ADF implica que a la izquierda de (5) se encuentra la variable en diferencias (Δy_t) y al lado derecho en cada uno de los regímenes se tendrá a la serie original rezagada un periodo ($y_{(t-1)}$) y los rezagos de las primeras diferencias (Δy_{t-1} hasta Δy_{p_i-p+1}).

Empleando esta función podemos estimar por ejemplo el modelo expresado en (4) con nuestra serie generada aleatoriamente. Por ejemplo,

```
library(tsDyn)
res1<-setar(y, m=2,, mL=1, mH=1, thDelay=1)
res1

##
## Non linear autoregressive model
##
## SETAR model ( 2 regimes)
## Coefficients:
## Low regime:
##   const.L      phiL.1
## 0.7735744 -1.1476555
##
```



```
## High regime:
##   const.H   phiH.1
## 1.6103043 0.6244573
##
## Threshold:
## -Variable: Z(t) = + (0) X(t)+ (1)X(t-1)
## -Value: 0.9976
## Proportion of points in low regime: 68.37%   High regime: 31.63%
```

Este paquete también permite emplear la función `summary` para presentar unos resultados más organizados

```
summary(res1)

##
## Non linear autoregressive model
##
## SETAR model ( 2 regimes)
## Coefficients:
## Low regime:
##   const.L   phiL.1
## 0.7735744 -1.1476555
##
## High regime:
##   const.H   phiH.1
## 1.6103043 0.6244573
##
## Threshold:
## -Variable: Z(t) = + (0) X(t)+ (1)X(t-1)
## -Value: 0.9976
## Proportion of points in low regime: 68.37%   High regime: 31.63%
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.11557 -1.14745 -0.14556  1.24078  3.71492
##
## Fit:
## residuals variance = 2.764,  AIC = 112, MAPE = 389.6%
##
## Coefficient(s):
##
##           Estimate Std. Error t value Pr(>|t|)
## const.L    0.77357    0.24097   3.2103 0.001804 **
## phiL.1   -1.14766    0.13526  -8.4849 2.648e-13 ***
## const.H    1.61030    0.81819   1.9681 0.051937 .
## phiH.1     0.62446    0.26950   2.3171 0.022622 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold
## Variable: Z(t) = + (0) X(t) + (1) X(t-1)
##
## Value: 0.9976
```

También se puede emplear la función `toLatex()` para obtener una versión en \LaTeX del modelo estimado. En este caso, la siguiente línea de código permite obtener la expresión en \LaTeX que tras compilar genera la ecuación estimada.

```
toLatex(res1)
```

$$X_{t-1} = \begin{cases} +0,774 - 1,148X_{t-0} & Z_t \leq +0,998 \\ +1,610 + 0,624X_{t-0} & Z_t > +0,998 \end{cases}$$

$$Z_t = X_{t-1}$$

Como era de esperarse los resultados de estimar los coeficientes de (4) son diferentes a los valores poblacionales. Si bien hemos determinado cómo estimar un modelo **SETAR** en R aún queda una pregunta fundamental por aclarar cómo determinar en la práctica el número de rezagos óptimo de cada régimen y el parámetro de retraso que se desconoce en la práctica. En la siguiente sección discutiremos una estrategia que permita encontrar el mejor modelo.

6. Selección del mejor modelo

Como es usual en la tarea de un investigador que emplea series de tiempo, los parámetros del modelo son desconocidos. Por ejemplo, para un modelo $ARMA(p, q)$ se debe determinar el p y el q . Esta tarea típicamente se desarrolla empleando criterios de información como el AIC , el SBC y el SSR . Para el caso de los modelos no lineales como los tratados aquí, es necesario determinar más parámetros. Para un modelo **SETAR** se debe determinar el número de regímenes, el número de rezagos para incluir en el umbral (m), el número de rezagos en cada régimen (p_1, p_2 , etc.), el parámetro de rezago (Δ) y el valor del o de los umbrales (r). Para los modelos **SETAR** se pueden calcular los AIC , el SBC y el SSR . Adicionalmente, se cuenta con una versión ampliada del criterio de información de Akaike denominado *pooled AIC* o AIC total. La función `selectSETAR()` del paquete **tsDyn** permite calcular automáticamente el AIC , el SBC , el SSR y el AIC total de un conjunto de modelos **SETAR** y compararlos. Al igual que las contrapartes lineales, el modelo con el menor criterio de información será el mejor candidato para iniciar la modelación de la serie.

Los argumentos de la función `selectSETAR()` son similares a los de la función `setar()` que se describieron anteriormente. Los valores por defecto de esta

función son los mismos de la función `setar()`. Esta función además incluye otros argumentos como:

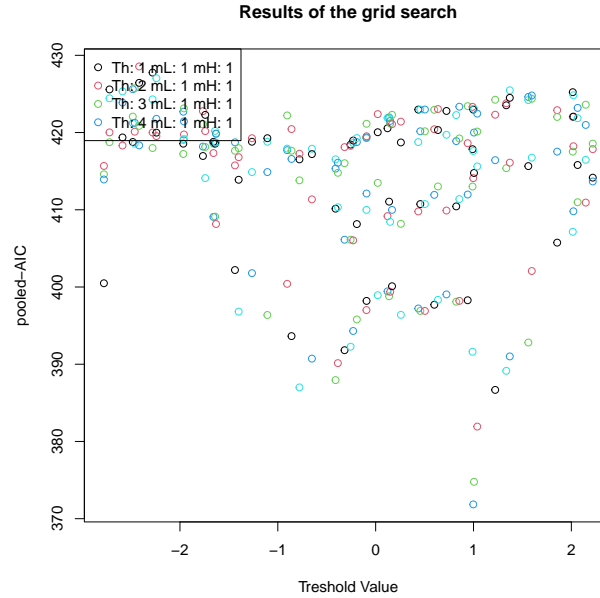
- **criterion**: el criterio que se empleará para escoger el modelo. Las opciones son: "pooled-AIC", "AIC", "BIC" y "SSR".
- **plot**: si se escoge la opción `TRUE` (predeterminada) al finalizar la ejecución de la función se presentará un resumen gráfico de los resultados.
- **same.lags**: Si se emplean los criterios de información *AIC* o *AIC* total y se establece el argumento m^3 , este argumento determinará si la búsqueda se realiza únicamente entre modelos con igual número de rezagos en todos los regímenes (`same.lags = TRUE`, opción predeterminada) o se permite que el número de rezagos sea diferente entre regímenes (`same.lags = FALSE`).
- **hpc**: Esta opción permite determinar si el procedimiento de búsqueda se distribuye de manera paralela en los núcleos del computador. Si se emplea la función "none", la búsqueda se realizará en un solo núcleo del computador. Con la opción "foreach" se puede distribuir el cálculo en los diferentes núcleos agilizando el proceso.

Para nuestra serie simulada por el modelo **SETAR**, podemos buscar el mejor modelo empleando el criterio *AIC* total buscando hasta 11 rezagos tanto en la variable de umbral como en los rezagos de cada régimen, hasta un parámetro de rezago de 4 y sin la necesidad de que los rezagos sean simétricos entre regímenes con la siguiente línea de código.

```
selectSETAR(y, m=11, mL=1:10, mH=1:10, same.lags = FALSE,
            thDelay=1:4)
```

```
## Searching on 61 possible threshold values within regimes with sufficient ( 15% ) number of
## Searching on 244 combinations of thresholds ( 61 ), thDelay ( 4 ), mL ( 1 ) and MM ( 11 )
## Results of the grid search for 1 threshold
##      thDelay mL mH      th pooled-AIC
## 1         1  1  1  0.9975909   371.8494
## 2         1  1  1  1.0051700   374.7743
## 3         1  1  1  1.0383376   381.9255
## 4         1  1  1  1.2217248   386.6761
## 5         1  1  1 -0.7785670   387.0010
## 6         1  1  1 -0.4097433   387.9454
## 7         1  1  1  1.3349528   389.1471
## 8         1  1  1 -0.3862790   390.1424
## 9         1  1  1 -0.6513558   390.7286
## 10        1  1  1  1.3708774   391.0130
```

³Ver la sección anterior para una explicación de este argumento.



De acuerdo con los resultados obtenidos el mejor modelo es el **SETAR** con un $\delta = 1$, $p_1 = p_2 = 1$.

7. Pronósticos fuera de muestra

Los modelos no lineales pueden ser empleados para realizar pronósticos. La función `predict` del paquete **tsDyn** permite realizar pronósticos fuera de muestra de un objeto no-lineal (clase `nlar`). Los argumentos de esta función son:

- **object**: Objeto de clase `nlar` a partir del cual se harán las proyecciones. Es decir, el objeto que se obtiene después de emplear la función `setar()`.
- **newdata**: Nuevos datos, si se desea hacer una proyección a partir de otra muestra.
- **n.ahead**: El número de periodos que se quiere pronosticar. El valor por defecto es uno.
- **type**: Método que se quiere emplear para generar el pronóstico. Una de las opciones es `"naive"`. Esta opción produce pronósticos que ignoran el comportamiento de los residuales. La literatura muestra que si se trata de dos o más periodos a pronosticar, si se ignora los residuos se produce un tipo de sesgo conocido en la literatura como ingenuo (en inglés `naive`). Otra opción es `"MC"` que implica emplear el método de Monte-Carlo para generar los residuos a partir de una distribución normal con desviación estándar igual al error estándar de los residuales. La opción `"bootstrap"`

implica generar los errores a partir de un muestreo de los residuales estimados del modelo. Finalmente, la opción **"block-bootstrap"** es igual a **"bootstrap"** pero se emplea remuestreos por bloques.

- **ci**: El nivel confianza del intervalo de proyección. Por defecto está predefinido **ci = 0.95**.

Empleando el objeto **res1**, que tiene el modelo **SETAR** estimado, podemos crear pronósticos. Supongamos que queremos crear pronósticos para los siguientes 10 periodos y empleando diferentes métodos para generar el error. El siguiente código produce los pronósticos.

```
pred_setar_naive <- predict(res1, n.ahead=10)
pred_setar_MC <- predict(res1, n.ahead=10,
                        type="MC")
pred_setar_boot <- predict(res1, n.ahead=10,
                        type="bootstrap")
pred_setar_Bboot <- predict(res1, n.ahead=10,
                        type="block-bootstrap")
```

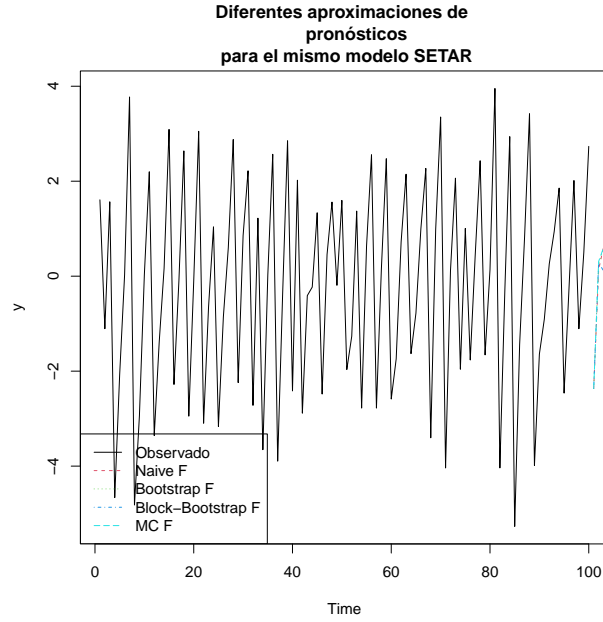
Cada uno de estos objetos contiene tanto el pronóstico, como el respectivo intervalo de confianza a excepción del método **"naive"**. Es decir,

```
pred_setar_naive

## Time Series:
## Start = 101
## End = 110
## Frequency = 1
## [1] -2.36352166  0.13438580  0.61934576  0.06277881  0.70152591 -0.03153569
## [7]  0.80976646 -0.15575854  0.95233150 -0.31937409
```

Finalmente, podemos hacer un gráfico que compare los pronósticos de los tres métodos empleando el siguiente código.

```
pred_range <- range(pred_setar_naive,
                    pred_setar_boot$pred,
                    pred_setar_MC$pred, na.rm=TRUE)
plot.ts(y, type= "l", main="Diferentes aproximaciones de
      pronósticos \n para el mismo modelo SETAR")
lines(pred_setar_naive, lty=2, col=2)
lines(pred_setar_boot$pred, lty=3, col=3)
lines(pred_setar_Bboot$pred, lty=4, col=4)
lines(pred_setar_MC$pred, lty=5, col=5)
legLabels <- c("Observado", "Naive F", "Bootstrap F",
              "Block-Bootstrap F", "MC F")
legend("bottomleft", leg=legLabels, lty=1:5, col=1:5)
```



8. Extensiones del modelo TAR

8.1. El modelo Momentum-TAR (M-TAR)

Enders y Granger (1998) proponen emplear como variable de umbral en el modelo SETAR de dos regímenes la misma variable en diferencias. Este modelo se conoce como Momentum-TAR (M-TAR). Es decir, el M-TAR de dos regímenes y parámetro de retraso uno será:

$$y_t = \begin{cases} \phi_1 + \phi_{1,1}y_{t-d} + \dots + \phi_{1,L}y_{t-(p_1-1)d} + \sigma_1\varepsilon_t & \Delta y_{t-1} \leq r \\ \phi_2 + \phi_{2,1}y_{t-d} + \dots + \phi_{2,H}y_{t-(p_2-1)d} + \sigma_2\varepsilon_t & \Delta y_{t-1} > r \end{cases} \quad (9)$$

La idea de este modelo es que el proceso presenta diferente “momentum” o movimiento dependiendo del cambio del periodo anterior. Si el umbral es cero ($r = 0$) esto implica que la serie se comporta diferente si el cambio de la variable en el periodo pasado fue positivo o negativo. Este tipo de modelos se puede estimar empleando la función `setar` del paquete **tsDyn** empleando el argumento `model` igual a "MTAR".

Por ejemplo, si quisiéramos estimar el modelo M-TAR de dos regímenes con $p_1 = p_2 = 1$ y parámetro de rezago uno para la serie generada aleatoriamente del modelo **SETAR** se puede emplear la siguiente línea de código.

```

res2<-setar(y, m=2,, mL=1, mH=1, thDelay=1, model="MTAR")
summary(res2)

##
## Non linear autoregressive model
##
## SETAR model ( 2 regimes)
## Coefficients:
## Low regime:
##   const.L      phiL.1
## 1.23826028 0.02067541
##
## High regime:
##   const.H      phiH.1
## -0.9746614 -0.3592869
##
## Threshold:
## Momentum Threshold (MTAR) Adjustment
## -Variable: Z(t) = + (0)D X(t)+ (1)DX(t-1)
## -Value: -2.77
## Proportion of points in low regime: 29.9%   High regime: 70.1%
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.91752 -1.04185 -0.22457  1.21912  4.75907
##
## Fit:
## residuals variance = 3.463,  AIC = 134, MAPE = 203%
##
## Coefficient(s):
##
##           Estimate Std. Error  t value Pr(>|t|)
## const.L  1.238260    0.363399   3.4074 0.0009595 ***
## phiL.1    0.020675    0.326347   0.0634 0.9496163
## const.H -0.974661    0.231462  -4.2109 5.736e-05 ***
## phiH.1   -0.359287    0.088877  -4.0425 0.0001067 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold
## Variable: Z(t) = + (0) X(t) + (1) X(t-1)
##
## Value: -2.77

```

Empleando la función `toLatex()` permite obtener la siguiente expresión en \LaTeX que reporta el modelo estimado.

$$X_{t-1} = \begin{cases} +1,2383 + 0,0207X_{t-0} & Z_t \leq -2,77 \\ -0,975 - 0,359X_{t-0} & Z_t > -2,77 \end{cases}$$

$$Z_t = \Delta X_{t-1}$$

8.2. TAR con tres regímenes

Una generalización del modelo TAR estudiado hasta este momento es el modelo TAR con más de dos regímenes. Por ejemplo, si nos restringimos a los modelos **SETAR**, un modelo **SETAR** de tres regímenes estará dado por:

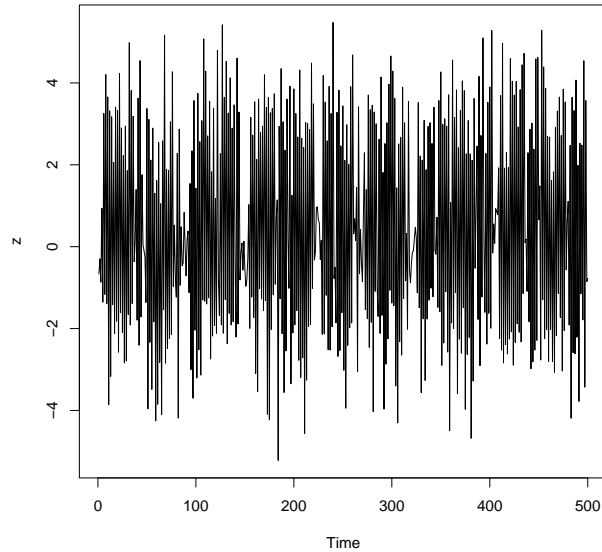
$$y_t = \begin{cases} \phi_1 + \phi_{1,1}y_{t-1} + \phi_{1,2}y_{t-2} + \dots + \phi_{1,2}y_{t-p_1} & y_{t-\delta} < r_1 \\ \phi_2 + \phi_{2,1}y_{t-1} + \phi_{2,2}y_{t-2} + \dots + \phi_{2,2}y_{t-p_2} & r_1 \leq y_{t-\delta} \leq r_2 \\ \phi_3 + \phi_{3,1}y_{t-1} + \phi_{3,2}y_{t-2} + \dots + \phi_{3,2}y_{t-p_3} & y_{t-\delta} > r_2 \end{cases} \quad (10)$$

Este tipo de modelos puede ser estimado por medio de la función **setar** del paquete **tsDyn**, especificando el número de umbrales igual a dos (**nthresh**= 2). Adicionalmente, el mejor modelo se puede seleccionar empleando la función **selectSETAR**. En este caso es necesario especificar (**nthresh**= 2).

Por ejemplo, generemos una muestra aleatoria de tamaño 500 del siguiente DGP:

$$y_t = \begin{cases} -3 + 0,25y_{t-1} & y_{t-1} < -1 \\ 0 + 0,3y_{t-1} & -1 \leq y_{t-1} \leq 1 \\ 4 + 0,35y_{t-1} & y_{t-1} > 1 \end{cases} \quad (11)$$

```
set.seed(2324113)
z <- setar.sim(B=c(4, 0.35, 0, 0.3, -3, 0.25), n = 500,
               lag=1, type="simul", nthresh=2,
               Thresh= c(-1,1))
ts.plot(z)
```

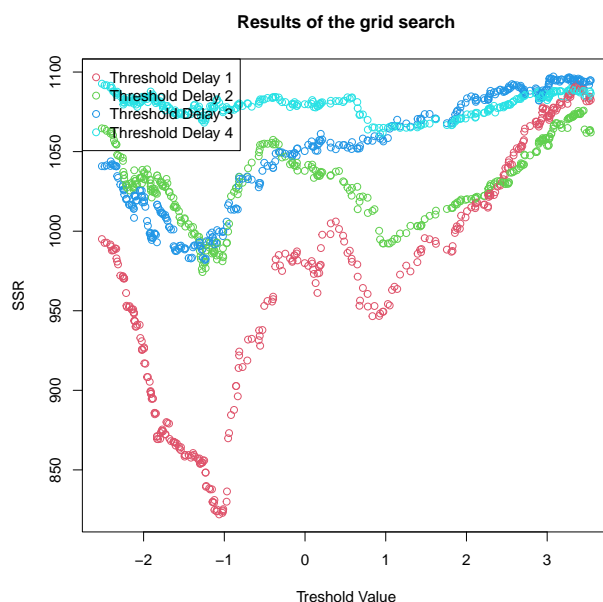



Una vez generada la muestra que se presenta en la Figura anterior, supongamos que se quiere encontrar el mejor modelo que ajuste a la muestra. Esto se puede lograr mediante las siguientes líneas de código.

```
selectSETAR(z, m=11, mL=1:10, mH=1:10, same.lags = FALSE,
            thDelay=1:4, nthresh=2)

## Using maximum autoregressive order for middle regime: mM = 11
## Searching on 341 possible threshold values within regimes with sufficient ( 15% ) number
## Searching on 1364 combinations of thresholds (341) and thDelay (4)
##
## 1 T: Trim not respected:  0.1492843 0.8507157 from th: -2.516548
## 1 T: Trim not respected:  0.1492843 0.8507157 from th: -2.516548Result of the one thresh
## -Thresh: -1.064092 -Delay:  1 - SSR 822.0227
## Second best: 0.9187119 (conditionnal on th= -1.064092 and Delay= 1 )    SSR/AIC: 742.3308
## Second best: -1.013405 (conditionnal on th= 0.9187119 and Delay= 1 )    SSR/AIC: 734.1667
## Results of the grid search for 1 threshold
##   Conditional on ML=  1 and MH=  1
##   thDelay      th      SSR
## 1         1 -1.064092 822.0227
## 2         1 -1.022385 822.9829
## 3         1 -1.091299 823.9356
## 4         1 -1.017043 823.9670
## 5         1 -1.098258 824.0423
##
## Results of the grid search for 2 thresholds
```

```
## Conditional on thDelay = 1 , and on ML= 1 , MM= 1 2 3 4 5 6 7 8 9 10 11 , MH= 1
##      th1      th2      SSR
## 1 -1.013405 0.9187119 734.1667
##
## Overall best results:
##      thDelay      th1      th2      SSR
## 1.0000000 -1.0134053 0.9187119 734.1666652
## With lags:
## -ML: 1
## -MM: 1 2 3 4 5 6 7 8 9 10 11
## -MH: 1
```



Como era de esperarse, los resultados muestran que según el criterio *SSR* el mejor modelo para esta muestra es el SETAR de tres regímenes, con un rezago para los regímenes de los extremos, y 11 para el de la mitad. Este modelo se puede estimar por medio de la siguiente línea.

```
res3<-setar(z, m=12, nthresh=2, thDelay=1, mL=1, mM=11 ,mH=1)
summary(res3)

##
## Non linear autoregressive model
##
## SETAR model ( 3 regimes)
## Coefficients:
## Low regime:
```

```

##      const.L      phiL.1
## -3.3693597  0.3870062
##
## Mid regime:
##      const.M      phiM.1      phiM.2      phiM.3      phiM.4      phiM.5
##  0.131851563 -1.250250464  0.072865600  0.010042315 -0.025402387  0.053690866
##      phiM.6      phiM.7      phiM.8      phiM.9      phiM.10      phiM.11
##  0.041263235  0.001521735 -0.023697036  0.083981581 -0.022016263 -0.058334529
##
## High regime:
##      const.H      phiH.1
##  1.8971753 -0.4340165
##
## Threshold:
## -Variable: Z(t) = + (0) X(t)+ (1)X(t-1)+ (0)X(t-2)+ (0)X(t-3)+ (0)X(t-4)+ (0)X(t-5)+ (0)X(t-6)
## -Value: -1.0134  0.9187
## Proportion of points in low regime: 41.19%  Middle regime: 15.16%  High regime: 43.65%
##
## Residuals:
##      Min      1Q      Median      3Q      Max
## -4.265143 -0.759943  0.082522  0.854372  3.233627
##
## Fit:
## residuals variance = 1.458,  AIC = 224,  MAPE = 160.9%
##
## Coefficient(s):
##
##      Estimate Std. Error t value Pr(>|t|)
## const.L -3.3693597  0.3077133 -10.9497 < 2.2e-16 ***
## phiL.1  0.3870062  0.0917875  4.2163 2.964e-05 ***
## const.M  0.1318516  0.2443450  0.5396 0.5897
## phiM.1 -1.2502505  0.1554576 -8.0424 6.831e-15 ***
## phiM.2  0.0728656  0.3290138  0.2215 0.8248
## phiM.3  0.0100423  0.1078921  0.0931 0.9259
## phiM.4 -0.0254024  0.1276775 -0.1990 0.8424
## phiM.5  0.0536909  0.1184645  0.4532 0.6506
## phiM.6  0.0412632  0.1363530  0.3026 0.7623
## phiM.7  0.0015217  0.1284769  0.0118 0.9906
## phiM.8 -0.0236970  0.1293833 -0.1832 0.8548
## phiM.9  0.0839816  0.1298196  0.6469 0.5180
## phiM.10 -0.0220163  0.1212108 -0.1816 0.8559
## phiM.11 -0.0583345  0.1021362 -0.5711 0.5682
## const.H  1.8971753  0.1815305 10.4510 < 2.2e-16 ***
## phiH.1 -0.4340165  0.0766936 -5.6591 2.609e-08 ***
## ---

```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold
## Variable: Z(t) = + (0) X(t) + (1) X(t-1)+ (0) X(t-2)+ (0) X(t-3)+ (0) X(t-4)+ (0) X(t-5)+
##
## Value: -1.0134  0.9187
```

Noten que en el régimen de la mitad de los rezagos 2 a 11 no son individualmente significativos con un 95 % de confianza. Al estimar el modelo con un rezago en los tres regímenes, obtenemos los siguientes resultados.

```
res4<-setar(z, m=2, nthresh=2, thDelay=1, mL=1, mM=1 ,mH=1)
summary(res4)

##
## Non linear autoregressive model
##
## SETAR model ( 3 regimes)
## Coefficients:
## Low regime:
##   const.L   phiL.1
## -3.3642268  0.3844842
##
## Mid regime:
##   const.M   phiM.1
##  0.1783205 -1.2432612
##
## High regime:
##   const.H   phiH.1
##  1.9413599 -0.4205604
##
## Threshold:
## -Variable: Z(t) = + (0) X(t)+ (1)X(t-1)+ (0)X(t-0)
## -Value: -1.0134  0.9187
## Proportion of points in low regime: 40.96%   Middle regime: 15.46%   High regime: 43.57%
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.304364 -0.806677  0.084677  0.822085  3.422473
##
## Fit:
## residuals variance = 1.492,  AIC = 216, MAPE = 151.8%
##
## Coefficient(s):
##
##           Estimate Std. Error  t value Pr(>|t|)
```

```
## const.L -3.364227    0.307717 -10.9329 < 2.2e-16 ***
## phiL.1   0.384484    0.091630   4.1960 3.221e-05 ***
## const.M  0.178321    0.140294   1.2710  0.2043
## phiM.1  -1.243261    0.140666  -8.8384 < 2.2e-16 ***
## const.H  1.941360    0.179867  10.7933 < 2.2e-16 ***
## phiH.1  -0.420560    0.076065  -5.5290 5.225e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold
## Variable: Z(t) = + (0) X(t) + (1) X(t-1)
##
## Value: -1.0134  0.9187
```

El modelo estimado, será entonces (para tres regímenes no trabaja bien la función `toLatex()`):

$$X_{t-1} = \begin{cases} -3,364 + 0,384X_{t-0} & Z_t \leq -1,013 \\ 0 - 1,243X_{t-0} & -1,013 < Z_t \leq 0,919 \\ 1,941 - 0,421X_{t-0} & Z_t > 0,919 \end{cases}$$

$$Z_t = X_{t-1}$$

8.2.1. Prueba de Hansen (1999)

Hansen (1999) construye una prueba que permite comparar un modelo **SETAR** con un modelo *AR* (i en este sentido es una prueba de linealidad) o con otro **SETAR** con diferente número de umbrales (en este sentido es una prueba de especificación). La idea detrás de esta prueba es comparar es la Suma Cuadrada de los Residuales (*SSR*) de los modelos por medio de una prueba del tipo F o prueba de Wald (también podría considerarse una prueba de razón de máxima verosimilitud).

El estadístico de prueba es

$$FH_{jk} = T \frac{(SSR_j - SSR_k)}{SSR_k}$$

Este estadístico permite probar la hipótesis nula de que el modelo j es mejor que el modelo k (alterna) $SETAR(j)$ contra $SETAR(k)$ ($k > j$). La distribución de este modelo no es estándar y por eso se aconseja tomar la decisión empleando técnicas de bootstrapping.

En R, podemos emplear la función `setarTest()` del paquete **tsDyn**. para efectuar esta prueba. Esta función emplea los argumentos `m` y `thDelay` definidos de igual manera que en la función `setar()`. Es decir, estos dos parámetros se refieren a la variable de umbral. Por otro lado, está el argumento `nboot` que es el número de de réplicas de las simulaciones de bootstrap y `test` que especifica los modelos que se compararán. Si `test = "1vs"` entonces se compara el modelo

AR lineal con el **SETAR** de un umbral y a parte también se compara con el **SETAR** de 2 umbrales y "2vs3" lo hace entre los **SETAR** de uno y 2 umbrales.

Por ejemplo, para probar la hipótesis nula de que el modelo **SETAR** con dos régimen es mejor que con tres regímenes, creemos un **SETAR** de con 2 umbrales podemos emplear el siguiente código:

```
# recuerda que m = m-1

setarTest(z, m=2, thDelay=1, nboot=100, test="2vs3")

## Test of setar(2) against setar(3)
##
##          Test Pval
## 2vs3 43.65764      0
```

Estos resultados muestran que existe suficiente evidencia para rechazar la nula de que el modelo con un umbral es mejor que el modelo con dos umbrales para la serie simulada.

```
setarTest(z, m=2, thDelay=1, nboot=100, test="1vs")

## Test of linearity against setar(2) and setar(3)
##
##          Test Pval
## 1vs2 103.1829      0
## 1vs3 155.8861      0
```

Y también podemos ver que los modelos no lineales son mejores que el AR para esta serie simulada.

8.3. Modelo LSTAR de dos regímenes

Una generalización del modelo TAR expresado en (5) es emplear como condición para estar en uno de los dos regímenes una probabilidad⁴ en vez de una desigualdad. En este caso el proceso estará en el régimen uno con una probabilidad que depende de una variable de umbral. Si suponemos que la función de probabilidad está descrita por una función logística tendremos que el modelo **LSTAR** (*Logistic Smooth Transition AutoRegressive*) corresponde a:

$$y_t = (\phi_1 + \phi_{1,1}y_{t-d} + \dots + \phi_{1,L}y_{t-(L-1)d}) (1 - G(z_t, \gamma, r)) + (\phi_2 + \phi_{2,1}y_{t-d} + \dots + \phi_{2,H}y_{t-(H-1)d}) G(z_t, \gamma, r) + \varepsilon_t \quad (12)$$

donde $G(Z_t, \gamma, r)$ es la función logística. Es decir,

$$G(Z_t, \gamma, r) = \frac{1}{1 + e^{-\gamma(z_t - r)}}$$

⁴También conocida como función de transición.

con $\gamma > 0$. γ representa el parámetro de suavizamiento, que en este caso también deberá ser estimada.

Este tipo de especificación permite una transición suave entre los regímenes. A diferencia de los modelos **TAR** que habíamos visto antes que implican un salto entre los regímenes.

Este tipo de modelos se puede estimar empleando la función `lstar()` del paquete **tsDyn**. Los argumentos son similares a los de las funciones empleadas hasta ahora.

Veamos un ejemplo empleando una de las series generadas con anterioridad.

```
res5 <- lstar(y, m=2, mTh=c(0,1), control=list(maxit=3000))

## Using maximum autoregressive order for low regime: mL = 2
## Using maximum autoregressive order for high regime: mH = 2
## Performing grid search for starting values...
## Starting values fixed: gamma = 100 , th = -2.755414 ; SSE = 265.6521
## Grid search selected lower/upper bound gamma (was: 1 100 ]).
## Might try to widen bound with arg: 'starting.control=list(gammaInt=c(1,200))'
## Convergence problem code 1. You might want to increase maximum number of iterations by se
## Optimized values fixed for regime 2 : gamma = 125.1245 , th = -2.763748 ; SSE = 265.1

summary(res5)

##
## Non linear autoregressive model
##
## LSTAR model
## Coefficients:
## Low regime:
##      const.L      phiL.1      phiL.2
## 0.07132667 1.39822232 -0.63099991
##
## High regime:
##      const.H      phiH.1      phiH.2
## -0.4708535 -2.0722307 -0.1046062
##
## Smoothing parameter: gamma = 125.1
##
## Threshold
## Variable: Z(t) = + (0) X(t) + (1) X(t-1)
##
## Value: -2.764
##
## Residuals:
##      Min      1Q    Median      3Q     Max
```

```
## -4.44325 -0.86450 -0.07059 0.98782 3.31762
##
## Fit:
## residuals variance = 2.651, AIC = 113, MAPE = 205.6%
##
## Coefficient(s):
##           Estimate Std. Error t value Pr(>|z|)
## const.L    0.071327   2.466138  0.0289 0.9769265
## phiL.1     1.398222   0.539482  2.5918 0.0095479 **
## phiL.2    -0.631000   0.719584 -0.8769 0.3805434
## const.H   -0.470854   2.473983 -0.1903 0.8490568
## phiH.1    -2.072231   0.547510 -3.7848 0.0001538 ***
## phiH.2    -0.104606   0.730053 -0.1433 0.8860645
## gamma    125.124491  164.868971  0.7589 0.4478927
## th       -2.763748   0.039211 -70.4838 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Non-linearity test of full-order LSTAR model against full-order AR model
## F = 2.5761 ; p-value = 0.08149
##
## Threshold
## Variable: Z(t) = + (0) X(t) + (1) X(t-1)
```

En la anterior línea de código incluimos el argumento `control = list(maxit=3000)` que le manda la orden al algoritmo de maximización de parar después de 3000 intentos.

El lector puede además explorar las siguientes funciones con los objetos que hemos estimado (Modelos **TAR**, **SETAR** de dos o tres regímenes, **M-TAR** y **LSTAR**) para realizar los correspondientes diagnósticos de los modelos que hemos estudiado hasta ahora:

- `residuals()`: para obtener los residuales.
- `AIC()`: para obtener el *AIC* del modelo.
- `BIC()`: para obtener el *BIC* del modelo.
- `plot()` para graficar los residuales y las funciones de autocorrelación y autocorrelación parcial estimadas.

9. Otros modelos no lineales

En el mundo de modelos no lineales podemos encontrar muchos otros modelos, en especial en el mundo no paramétrico existen muchos desarrollos que comúnmente son empleados para pronosticar; por ejemplo, los modelos *GAM*

(Modelos Aditivos Generalizados) y las redes neuronales. A continuación se presentan estos dos otros modelos para hacernos una idea de este mundo no lineal.

9.1. Generalized Additive Model (GAM)

Los modelos GAM son modelos no paramétricos también conocidos como *Additive Non-lineal Autoregressive models* (**AAR**). Un modelo GAM tienen la siguiente forma:

$$y_t = \mu + \sum_{i=1}^m s_i(y_{t-(i-1)d}) \quad (13)$$

donde $s_i(\cdot)$ corresponde a una función de suavización que emplea una regresión cúbica de *spline*⁵.

Este tipo de modelos se pueden estimar empleando la función `aar()` del paquete **tsDyn**. Los argumentos son similares a los de las funciones empleadas hasta ahora. Estimemos el modelo **AAR** con $m = 2$ para una de nuestras series simuladas.

```
res6 <- aar(y, m=2)

summary(res6)

##
## Non linear autoregressive model
##
## AAR model
##
## Family: gaussian
## Link function: identity
##
## Formula:
## y ~ s(V1.0, bs = "cr") + s(V1..1, bs = "cr")
##
## Estimated degrees of freedom:
## 7.86 7.90 total = 16.77
##
## GCV score: 1.560082
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

⁵Una regresión cúbica de *spline* (también conocido como *B-spline*) es un método de suavización de una serie de tiempo que implica interpolaciones por pedazos. Se ajusta un polinomio cúbico a cada porción de la serie a diferencia de las tendencias polinómicas tradicionales que ajustan un polinomio a toda la muestra. La regresión cúbica de *spline* es un caso especial de los modelos lineales generalizados (GLS).

```
## -2.950359 -0.589596 -0.055224 0.653771 2.862557
##
## Fit:
## residuals variance = 1.05, AIC = 43, MAPE = 111.3%
##
## Family: gaussian
## Link function: identity
##
## Formula:
## y ~ s(V1.0, bs = "cr") + s(V1..1, bs = "cr")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.23104    0.11487  -2.0113  0.04761 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## s(V1.0)    7.8615 8.6128 16.3566 <2e-16 ***
## s(V1..1)   7.9047 8.6521  0.9944  0.5861
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.749 Deviance explained = 79%
## GCV = 1.5601 Scale est. = 1.2932 n = 98
```

Noten que el segundo término no suavizamiento no es significativo, podemos entonces estimar el modelo sin el segundo término ($m = 1$).

```
res7 <- aar(y, m=1)

summary(res7)

##
## Non linear autoregressive model
##
## AAR model
##
## Family: gaussian
## Link function: identity
##
## Formula:
## y ~ s(V1.0, bs = "cr")
##
## Estimated degrees of freedom:
```

```
## 7.73 total = 8.73
##
## GCV score: 1.453494
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.433985 -0.583102  0.037007  0.770084  2.871934
##
## Fit:
## residuals variance = 1.196,  AIC = 38, MAPE = 112%
##
## Family: gaussian
## Link function: identity
##
## Formula:
## y ~ s(V1.0, bs = "cr")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.23988    0.11570  -2.0733  0.04099 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(V1.0)  7.7326  8.5247 33.197 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.74  Deviance explained = 76.1%
## GCV = 1.4535  Scale est. = 1.3253    n = 99
```

Adicionalmente, con las métricas podemos ver que el modelo con solo el primer término es mejor.

```
AIC(res6)
## [1] 42.92642
AIC(res7)
## [1] 37.92322
BIC(res6)
## [1] 92.42465
BIC(res7)
## [1] 63.97492
```

En la mayoría de los casos, estos modelos univariados son empleados para generar pronósticos, y por tanto la bondad de estos modelos implicará hacer una evaluación de sus pronósticos fuera de muestra.

9.2. Red Neuronal no-lineal Autoregresiva

En el mundo del aprendizaje de máquina también se han generado aproximaciones para construir modelos univariados para pronosticar. Un ejemplo es el modelo de Red Neuronal Artificial (NN) con una salida lineal, D unidades escondidas y función de activación g que se puede representar de la siguiente forma:

$$y_t = \beta_0 + \sum_{j=1}^D \beta_j g \left(\gamma_{0j} + \sum_{i=1}^m \gamma_{ij} y_{t-(i-1)d} \right) \quad (14)$$

A este modelo se le conoce como Red Neuronal no-lineal Autoregresiva *nonlinear autoregressive neural network* (**NARNN**). Como la mayoría de modelos de Redes Neuronales, funcionan como “cajas negras” en las que los datos entran y la relación entre las diferentes capas de neuronas generan un resultado, sin que podamos ver claramente cómo se producen dichas interacciones. No obstante ese problema, las redes neuronales presentan en algunas aplicaciones buenos resultados al momento de generar pronósticos.

Este tipo de modelos se pueden estimar empleando la función `nnetTS()` del paquete **nnet** (Venables y Ripley, 2002). Los argumentos son similares a los de las funciones empleadas hasta ahora, el único argumento diferente es `size` que determina el número de capas ocultas de la red neuronal. Este será el parámetro que el investigador tendrá que calibrar hasta encontrar el resultado deseado.

Veamos un ejemplo

```
library(nnet)
res8 <- nnetTs(y, d=1, m=2, size=2)

summary(res8)

##
## Non linear autoregressive model
##
## NNET time series model
## a 2-2-1 network with 9 weights
## options were - linear output units
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.40063922 -0.63143359  0.07073162  0.61779234  2.77512107
##
## Fit:
## residuals variance = 0.9437,  AIC = 12, MAPE = 141.8%
```

Veamos qué ocurre si aumentamos el número de capas ocultas en la red neuronal.

```
library(nnet)
res9 <- nnetTs(y, d=1, m=4, size=2)
AIC(res8)

## [1] 12.20095

AIC(res9)

## [1] 57.58496

# usemos menos capas
res10 <- nnetTs(y, d=1, m=1, size=2)
AIC(res10)

## [1] 9.471791
```

Así parece que según esta métrica interna es mejor emplear dos capas. ¡Noten que este modelo no tiene supuestos estadísticos para ser validados!

10. Comparando los modelos

En algunas situaciones prácticas nos vemos en la necesidad de escoger entre cuál de estos modelos explica mejor la muestra estudiada⁶.

Veamos rápidamente como escoger el mejor modelo que explica la muestra generada a partir del modelo **SETAR** de dos regímenes y un rezago en cada periodo y parámetro de rezago uno descrito en 4 (objeto *y*). Hemos estimado diferentes modelos a lo largo de este documento.

En el objeto *res1* guardamos la estimación de un modelo **SETAR** con un rezago en los dos regímenes y parámetro de rezago uno (el modelo del cuál se generaron los datos). También se estimó un modelo **M-TAR** que se guardó en el objeto *res2*. Y en los objetos *res5*, *res7* y *res8* se guardaron las estimaciones de los modelos *LSTAR*, mejor **AAR** y mejor **NARNN**, respectivamente.

Estos modelos se pueden comparar en su comportamiento en la muestra empleando criterios de información como el *AIC* o métricas de ajuste como el *MAPE*. Por ejemplo, en este caso podemos emplear el siguiente código.

```
modelos <- list()
modelos[["SETAR"]] <- res1
modelos[["M-SETAR"]] <- res2
modelos[["LSTAR"]] <- res5
```

⁶Nota que esto no es lo mismo que escoger el mejor modelo para pronosticar. Podríamos tener un problema de **over fitting** y el modelo es muy bueno para explicar la muestra pero no lo que pasará en el futuro.

```

modelos[["AAR"]] <- res7
modelos[["ANN"]] <- res8
sapply(modelos, AIC)

##      SETAR    M-SETAR      LSTAR      AAR      ANN
## 111.65880 134.20897 113.49668  37.92322 12.20095

sapply(modelos, BIC)

##      SETAR    M-SETAR      LSTAR      AAR      ANN
## 124.68465 147.23482 134.33804  63.97492 35.64748

sapply(modelos, MAPE)

##      SETAR    M-SETAR      LSTAR      AAR      ANN
## 3.895723 2.029556 2.055656 1.120264 1.417765

sapply(modelos, mse)

##      SETAR    M-SETAR      LSTAR      AAR      ANN
## 2.7637488 3.4628422 2.6510792 1.1962985 0.9436589

```

Noten que en este caso, y solo por el azar, las métricas dicen que el modelo **ANN** se ajusta mejor a los datos.

11. Comentarios finales

En este documento hemos discutido los modelos no lineales **TAR**, **SETAR**, **M-TAR** y **LSTAR** que pueden ser útiles para modelar series de tiempo. Estos modelos paramétricos podrían generar interpretaciones interesantes del comportamiento de una serie de tiempo, en especial si se comparan con sus contrapartes lineales de la familia *ARIMA*. No obstante la interpretabilidad de estos modelos no es su fuerte. Es importante recordar que en la mayoría de los casos que se emplean los modelos univariados el principal objetivo es generar un buen pronóstico.

Por otro lado, existen otros modelos no-paramétricos (no lineales) que también pueden ser útiles para modelar una serie de tiempo. De este mundo no paramétrico discutimos los modelos **AAR** y **NARNN**, pero existen muchas más aproximaciones paramétricas que no hemos discutido aún. Estas aproximaciones no-paramétricas típicamente tienen menos interpretabilidad que los modelos paramétricos, pero lo más importante es su capacidad de generar buenos pronósticos.

En la práctica nos vemos en la necesidad de escoger entre diferentes modelos lineales empleados. Esto típicamente se hará fuera de muestra⁷ con ventanas móviles o recursivas. De esta manera se podría seleccionar el mejor modelo

⁷En la terminología de los científicos de datos se le llamaría validación cruzada.

para pronosticar. Este es un tema más amplio que seguramente ya estudiaste o estudiarás en un curso de pronósticos.

Referencias

- [1] Antonio, Fabio Di Narzo, Jose Luis Aznarte, and Matthieu Stigler. *tsDyn: Time series analysis based on dynamical systems theory*, 2009.
- [2] K . S . Chan. Percentage Points of Likelihood Ratio Tests for Threshold Autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):691–696, 1991.
- [3] Kung-Sik Chan and Brian Ripley. *TSA: Time Series Analysis*, 2022. R package version 1.3.1.
- [4] Kung Sik Chan and Howell Tong. On estimating thresholds in autoregressive models. *Journal of time series analysis*, 7(3):179–190, 1986.
- [5] Walter Enders and C. W. J. Granger. Unit-Root Tests and Asymmetric Adjustment with an Example Using the Term Structure of Interest Rates. *Journal of Business & Economic Statistics*, 16(3):304–311, 1998.
- [6] Bruce Hansen. Testing for linearity. *Journal of Economic Surveys*, 13(5):551–576, 1999.
- [7] Daniel MacRae Keenan. A Tukey Nonadditivity-Type Test for Time Series Nonlinearity. *Biometrika*, 72(1):39–44, 1985.
- [8] A I McLeod and W K Li. DIAGNOSTIC CHECKING ARMA TIME SERIES MODELS USING SQUARED-RESIDUAL AUTOCORRELATIONS. *Journal of Time Series Analysis*, 4(4):269–273, 1983.
- [9] Ruey S. Tsay. Nonlinearity Tests for Time Series. *Biometrika*, 73(2):461–466, 1986.
- [10] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.