

Introducción a series de tiempo

LAAN

4/16/2019

Contents

Procesos estacionarios en covarianza	1
Función de autocorrelación (ACF o correlograma)	1
Ruido blanco	2
Inferencia	2
Autocorrelación serial individual	2
Autocorrelación serial conjunta	3
Pruebas de autocorrelación serial conjunta: pruebas portmanteau	3
Simulación	5
Procesos de media móvil MA(q)	7
Operador de rezago	8
Características de un proceso MA(q)	8
Media incondicional	8
Varianza incondicional	8
Covarianzas	9
Invertibilidad de un proceso MA(q)	9
Invertibilidad y unicidad de un proceso MA(q)	10
Correlogramas y memoria del proceso	10
Procesos autoregresivos AR(p)	11
Estacionariedad de un modelo AR(p)	11
Ecuación característica	11
Integración	15
Sobrep parametrización	16
Pruebas de hipótesis de estacionariedad	16
Prueba de raíz unitaria	17
Prueba de estacionariedad	19
Análisis de confirmación	20
Características de un proceso AR(p)	20
Teorema de Wold	20
Media incondicional	21
Varianza incondicional	21
Correlaciones seriales	22
Función de autocorrelación parcial (PACF)	23

Correlogramas y memoria de un proceso AR; comparaciones ACF, PACF para procesos MA, AR	23
Ejemplo	24
Modelos ARMA, ARIMA	26
Sobrep parametrización	26
Ejemplo	26
Características de un proceso ARMA(p,q)	27
La metodología de Box-Jenkins	28
Criterios de Información	28
Pronósticos	29
Método de pronóstico simple	29
Pronóstico mediante suavización exponencial	30
Pronóstico para MA(q)	31
Pronóstico para AR(p)	31
Pronósticos con R	32
Ejemplo 1 (Simulación)	32
Ejemplo 2	39
Ejemplo 3	45
Heterocedasticidad condicional y volatilidad	49
Características de la volatilidad	49
Medida de volatilidad	50
Modelación ARCH	51
Procedimiento	51
Ejemplo	52
Pronóstico	60
Modelo ARCH-Innovaciones no gaussianas	62
Innovación t-Student	63
Modelos GARCH	65
Fortalezas y debilidades de la modelación GARCH	65
Estimación GARCH de INTC	66
Diagnóstico	68
ACF, PACF	69
Ljung-Box, lag = 10	71
Ljung-Box, lag = 20	71
Ljung-Box, lag=30	71
Normalidad de las innovaciones	72
Intervalo predictivo alrededor de la media	73
Modelo (G)ARCH con serie ARMA(p,q)	75
Estimación de las ecuaciones de media	76
Detección de heterocedasticidad condicional	77

Estimación de la volatilidad	81
Gráficos con intervalos predictivos	87
Otros modelos	90
Modelo GARCH-M "GARCH in the mean"	90
Modelos EGARCH(1,1)	93
Modelos EGARCH(1,1) con prima de riesgo	95
Modelos VAR (Vectoriales Autoregresivos)	98
Regresiones espurias y estacionariedad	98
Estimación VAR	103
Determinación del orden y estimación	104
Diagnóstico	106
Causalidad según Granger	107
Respuesta a impulso unitario	108
Bibliografía	108

Procesos estacionarios en covarianza

Se dice que una serie (proceso) y_t es **estacionario en covarianza** si

Media $E[y_t] = \mu$

Varianza $E[(y_t - \mu)^2] = \sigma^2$

Autocovarianza $E[(y_{t_1} - \mu)(y_{t_2} - \mu)] = \gamma_{t_1 - t_2}$, para todo t_1, t_2 .

Si

$$t_1 = t_2 + s$$

(s : rezago, “lag”, en inglés), entonces

$$E[(y_{t_2+s} - \mu)(y_{t_2} - \mu)] = \gamma_s$$

es una función del rezago.

El término γ_s es la **función de autocovarianza**.

Nótese que la varianza

$$\text{var}(y_t) = E[y_t^2] - (E[y_t])^2 = \gamma_0.$$

Comentario La autocovarianza γ_s depende de las unidades de medición usadas, por lo que es más conveniente usar la **serie de autocorrelaciones**

$$\tau_s = \frac{\gamma_s}{\gamma_0}, \quad s = 1, 2, \dots$$

Función de autocorrelación (ACF o correlograma)

La **función de autocorrelación** (ACF) es la gráfica de las autocorrelaciones τ_s contra s , $s = 0, 1, \dots$.

Por definición, $\tau_0 = 1$.

La función de autocorrelación también se conoce como **correlograma**.

Ruido blanco

Un **ruido blanco**, u_t , es un proceso estocástico (serie de tiempo) con las siguientes propiedades:

1. $E[u_t] = \mu$, para todo $t > 0$.
2. (**Homocedasticidad**) $var(u_t) = \sigma^2$, para todo $t > 0$.
3. $\gamma_{t_1-t_2} = \begin{cases} \sigma^2, & \text{si } t_1 = t_2; \\ 0, & \text{en cualquier otro caso.} \end{cases}$

Comentarios Se presentan a continuación unas consecuencias de las propiedades de un ruido blanco.

- Para un ruido blanco, si $s = t_1 - t_2$, entonces un ruido blanco tiene función de autocorrelación

$$\tau_s = \begin{cases} 1, & s = 0, \\ 0, & s \neq 0 \end{cases}$$

- Si la media $\mu = 0$, el proceso se conoce como un ruido blanco con media cero. En este caso,

$$\text{corr}(u_t, u_s) = E[u_t u_s] = 0.$$

Además,

$$var(u_t) = \sigma^2 = E[u_t^2],$$

para todo $t > 0$.

- Si el ruido blanco también tiene distribución normal, entonces los coeficientes de autocorrelación muestral

$$\hat{\tau}_s \overset{approx}{\sim} N(0, 1/T)$$

(T es el tamaño muestral).

Inferencia

Si un ruido blanco también tiene distribución normal, entonces se pueden determinar intervalos de confianza para los valores estimados $\hat{\tau}_s$ de la función de autocorrelación.

Autocorrelación serial individual

Los intervalos de confianza se pueden usar para probar la hipótesis de autocorrelación serial individual

$$H_0 : \tau_s = 0$$

$$H_1 : \tau_s \neq 0$$

con nivel de significancia α .

En lo que sigue, se usará $\alpha = .05$ en todas las pruebas de hipótesis.

La prueba de hipótesis es equivalente a

$$H_0 : \text{corr}(u_t, u_{t+s}) = 0$$

$$H_1 : \text{corr}(u_t, u_{t+s}) \neq 0$$

$$s = 1, 2, \dots$$

En la prueba de hipótesis, el estadístico de prueba es el estadístico t usual:

$$T = \frac{\hat{\tau}_s}{\text{ee}(\hat{\tau}_s)}$$

donde $\text{ee}(\hat{\tau}_s)$ es el error estándar del estimador $\hat{\tau}_s$.

Autocorrelación serial conjunta

Una prueba de autocorrelación serial está basada en la siguiente prueba de hipótesis

$$H_0 : \tau_1 = 0, \dots, \tau_m = 0$$

$$H_1 : \tau_1 \neq 0, \dots, \tau_m \neq 0$$

con $m \geq 2$.

En un diagnóstico de ruido blanco más adelante, la prueba de hipótesis es equivalente a

$$H_0 : u_t \text{ es ruido blanco}$$

$$H_1 : u_t \text{ no es ruido blanco}$$

Pruebas de autocorrelación serial conjunta: pruebas portmanteau

Pruebas de autocorrelación serial conjunta reciben el nombre de *pruebas portmanteau* y se conocen por los nombres de los diferentes autores que la descubrieron.

Las pruebas que usaremos son las de

- Box-Pierce [BP]

- Ljung-Box [LB]

Una discusión acerca de pruebas portmanteau a casos más generales se encuentra en [M].

Prueba de Box-Pierce

Si T es el tamaño muestral, el *estadístico de Box-Pierce* es

$$Q = T \sum_{k=1}^m \hat{\tau}_k^2 \stackrel{aprox}{\sim} \chi_m^2$$

El estadístico de Box-Pierce no parece dar resultados buenos con muestras pequeñas.

Prueba de Ljung-Box

Un ajuste a muestras pequeñas es el *estadístico de Ljung-Box*

$$Q^* = T(T+2) \sum_{k=1}^m \frac{\hat{\tau}_k^2}{T-k} \stackrel{aprox}{\sim} \chi_m^2$$

Asintóticamente, $T \rightarrow \infty$, los dos estadísticos Q , Q^* son comparables.

Ejemplo

Usando $T = 100$ observaciones, se estimaron los primeros cinco coeficientes de autocorrelación:

s	1	2	3	4	5
$\hat{\tau}_s$.227	-.013	.0805	.004	-.024

Prueba la significancia estadística individual y conjunta ($m = 5$) de estos coeficientes de autocorrelación. Usar $\alpha = .05$.

Solución: Significancia estadística individual El intervalo de confianza asociado es

$$\pm 1.96 \times \frac{1}{\sqrt{T}} \Rightarrow (-.196, .196)$$

Si $\hat{\tau}_s$ se encuentra en el intervalo de confianza, entonces **no** es significativo; de lo contrario, sí es significativo.

De los valores estimados, solo el primero es significativo.

Solución: significancia estadística en conjunto El estadístico de Box-Pierce

$$Q = 100 \cdot [(.227)^2 + (-.013)^2 + (.0805)^2 + (.004)^2 + (-.024)^2] = \mathbf{5.877025}.$$

El estadístico de Ljung-Box da

$$Q^* = (100 \cdot 102) \cdot [(.227)^2/(100-1) + (-.013)^2/(100-2) + (.0805)^2/(100-3) + (.004)^2/(100-4) + (-.024)^2/(100-5)] = \mathbf{6.0716}.$$

Compararemos el valor de cada estadístico de prueba con el valor crítico de la distribución χ^2_5 :

```
qchisq(.05, 5, lower.tail = F)
```

```
## [1] 11.0705
```

Ambos estadísticos son menores que el valor crítico, por lo que no se puede rechazar la hipótesis nula. Concluimos que

$$\tau_1 = 0, \tau_2 = 0, \tau_3 = 0, \tau_4 = 0, \tau_5 = 0$$

Simulación

Simularemos $T = 100$ observaciones de un ruido blanco con distribución normal con media 0 y varianza 1.

- Se determinará la ACF.
- Se probará la significancia individual de las primeras 20 observaciones.
- Se probará la significancia conjunta con, sucesivamente, $m = 10$; $m = 20$, $m = 50$, usando Q , Q^* .

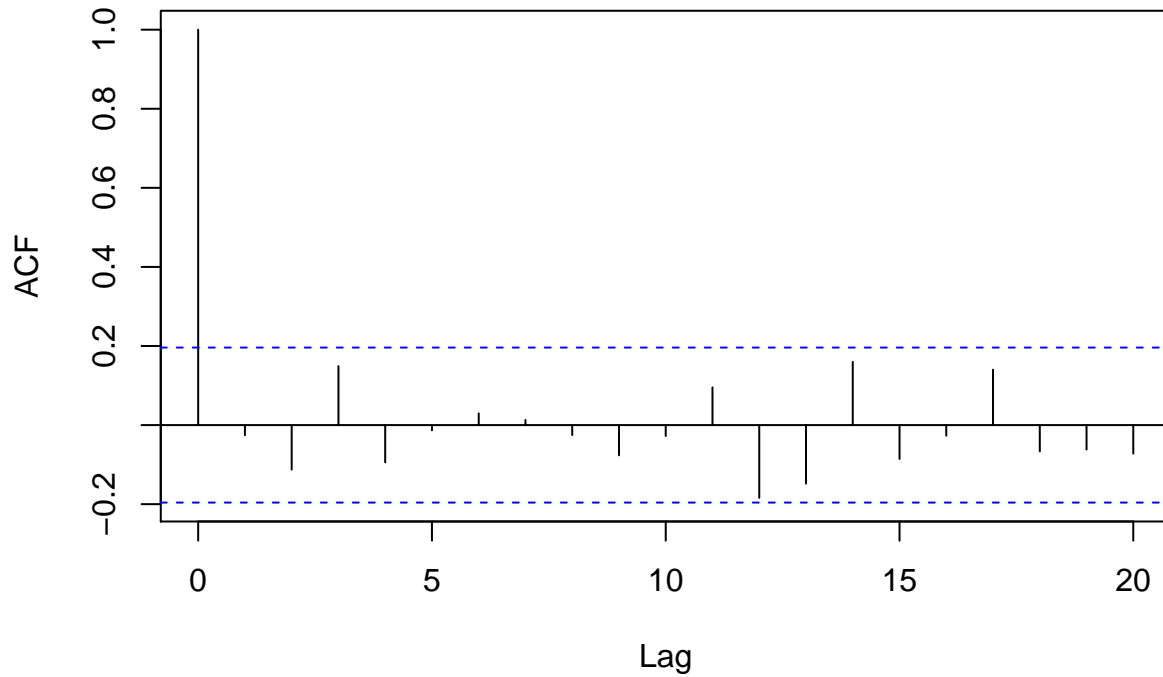
El código se da a continuación. Se usará `set.seed(123)` para obtener resultados reproducibles. El comando `x <- rnorm(100)` genera 100 muestras de una distribución normal estándar:

```
set.seed(123)
x <- rnorm(100)
```

La prueba de significancia estadística individual se lleva a cabo con la función de autocorrelación `acf(as.ts(x))`, en donde se pasa el argumento `as.ts(x)` como serie de tiempo:

```
acf(as.ts(x), main = "Simulación ruido blanco")
```

Simulación ruido blanco



El parámetro opcional `main = "Simulación ruido blanco"` permite poner un título a la gráfica.

En el correlograma (AFC), $\tau_0 = 1$ siempre.

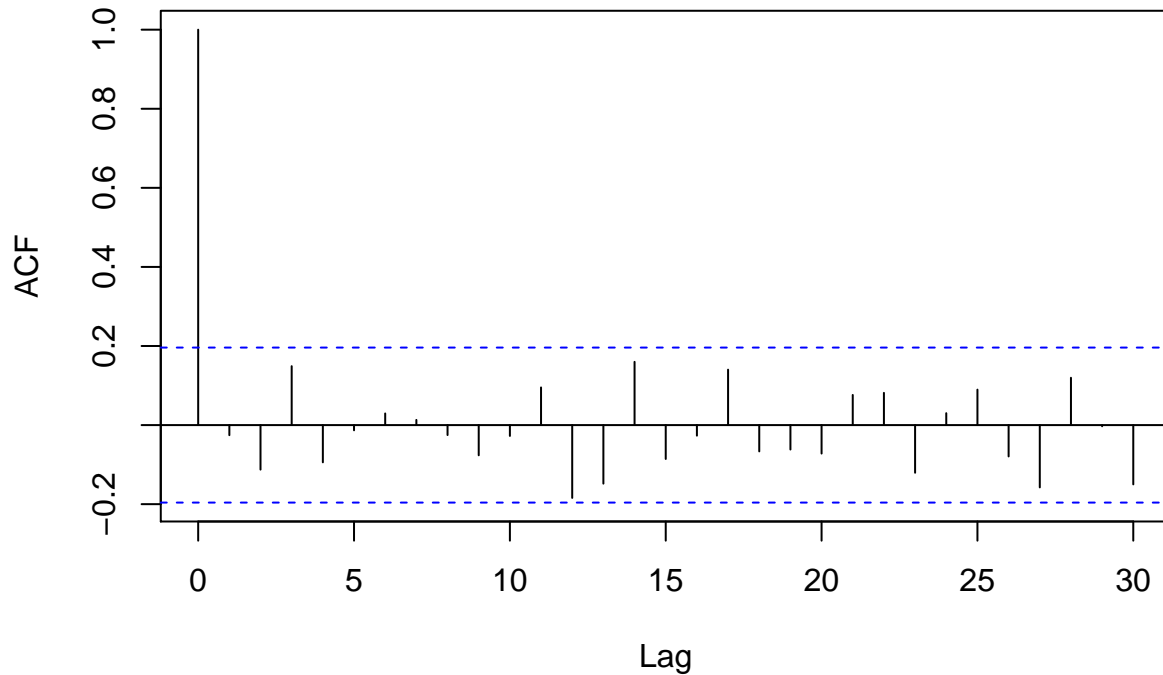
El intervalo de confianza, al 95%, está delimitado por las líneas punteadas en azul. Si un valor $\hat{\tau}_s$ rebasa el intervalo de confianza, entonces $\tau_s \neq 0$.

Podemos ver que $\tau_s = 0$, $s = 1, 2, \dots, 20$.

Lo mismo, pero con más rezagos ($m = 30$):

```
acf(as.ts(x), lag.max = 30, main = "Simulación ruido blanco")
```


Simulación ruido blanco



También en este caso, $\tau_s = 0$, $s = 1, 2, \dots, 30$.

La prueba de significancia conjunta se realiza usando la función `Box.test()`; se especifica el valor del rezago que se quiere probar ($m = 10$; $m = 20$, $m = 50$, en este ejemplo) y el estadístico de prueba (default: Box-Pierce):

```
Box.test(as.ts(x), lag=10)
```

```
##
## Box-Pierce test
##
## data:  as.ts(x)
## X-squared = 5.2889, df = 10, p-value = 0.8711
```

```
Box.test(as.ts(x), lag=20)
```

```
##
## Box-Pierce test
##
## data:  as.ts(x)
## X-squared = 18.46, df = 20, p-value = 0.5572
```

```
Box.test(as.ts(x), lag=50)
```

```
##
## Box-Pierce test
##
```

```
## data:  as.ts(x)
## X-squared = 36.977, df = 50, p-value = 0.9144
```

```
Box.test(as.ts(x), lag=10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
```

```
## data:  as.ts(x)
## X-squared = 5.6107, df = 10, p-value = 0.8468
```

```
Box.test(as.ts(x), lag=20, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
```

```
## data:  as.ts(x)
## X-squared = 21.271, df = 20, p-value = 0.3814
```

```
Box.test(as.ts(x), lag=50, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
```

```
## data:  as.ts(x)
## X-squared = 49.711, df = 50, p-value = 0.4849
```

Concluimos que no existe evidencia de autocorrelación serial, con $m = 10$, $m = 20$, $m = 50$, respectivamente.

Procesos de media móvil MA(q)

Sea u_t un ruido blanco con media $E(u_t) = 0$ y varianza $var(u_t) = \sigma^2$.

La serie y_t es un proceso de **media móvil de orden q** , si

$$y_t = \mu + u_t + \theta_1 u_{t-1} + \cdots + \theta_q u_{t-q}.$$

Una serie de media móvil de orden q se denota con $MA(q)$.

Comentario En ciertos cálculos algebraicos se puede considerar, sin pérdida de generalidad, $\mu = 0$.

Operador de rezago

El **operador de rezago** L (para un periodo) se define

$$Ly_t = y_{t-1}.$$

Para periodos sucesivos, se define de manera inductiva mediante

$$L^i y_t = y_{t-i},$$

de manera que una serie $MA(q)$ se puede escribir

$$y_t = \mu + \sum_{i=1}^q \theta_i L^i u_t + u_t = \mu + \theta(L)u_t,$$

con $\theta(L) = 1 + \theta_1 L + \cdots + \theta_q L^q$.

Características de un proceso MA(q)

Sea $y_t = \mu + u_t + \theta_1 u_{t-1} + \cdots + \theta_q u_{t-q}$ un proceso $MA(q)$. A continuación se probarán las propiedades más básicas de este proceso.

Media incondicional

$$E(y_t) = \mu$$

Solución El valor esperado es lineal, por lo que

$$\begin{aligned} E[y_t] &= E[\mu + u_t + \theta_1 u_{t-1} + \cdots + \theta_q u_{t-q}] \\ &= E[\mu] + E[u_t] + \theta_1 E[u_{t-1}] + \cdots + \theta_q E[u_{t-q}] \\ &= \mu \end{aligned}$$

siendo u_t un ruido blanco con media cero.

Varianza incondicional

$$var(y_t) = \gamma_0 = (1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2)\sigma^2$$

Solución Sin pérdida de generalidad, se puede suponer que $\mu = E[y_t] = 0$. En este caso, la varianza

$$var(y_t) = E[y_t^2] - (E[y_t])^2 = E[y_t^2]$$

El cuadrado del proceso

$$\begin{aligned} y_t^2 &= (u_t + \theta_1 u_{t-1} + \cdots + \theta_q u_{t-q})^2 \\ &= u_t^2 + \theta_1^2 u_{t-1}^2 + \cdots + \theta_q^2 u_{t-q}^2 + \text{productos cruzados} \end{aligned}$$

donde los productos cruzados consisten de términos del tipo $u_{t-j}u_{t-k}$, $j \neq k$. El valor esperado de los productos cruzados es cero, por la tercera propiedad de un ruido blanco. Por lo tanto,

$$\begin{aligned} \text{var}(y_t) &= E[y_t^2] = E[u_t^2 + \theta_1^2 u_{t-1}^2 + \cdots + \theta_q^2 u_{t-q}^2 + \text{productos cruzados}] \\ &= E[u_t^2] + \theta_1^2 E[u_{t-1}^2] + \cdots + \theta_q^2 E[u_{t-q}^2] \\ &= \sigma^2 + \theta_1^2 \sigma^2 + \cdots + \theta_q^2 \sigma^2 \\ &= \sigma^2 (1 + \theta_1^2 + \cdots + \theta_q^2) \end{aligned}$$

Para pasar de la segunda línea a la tercera arriba se usa la propiedad de *homocedasticidad* de un ruido blanco (Propiedad 2).

Covarianzas

Las covarianzas

$$\gamma_s = \begin{cases} (\theta_s + \theta_{s+1}\theta_1 + \cdots + \theta_q\theta_{q-s})\sigma^2, & s = 1, 2, \dots, q \\ 0, & \text{para } s > q \end{cases}$$

La prueba de esta propiedad se encuentra en el libro de Brooks [Br].

Invertibilidad de un proceso MA(q)

Considérese un modelo $MA(q)$

$$y_t = u_t + \theta_1 u_{t-1} + \theta_2 u_{t-2} + \cdots + \theta_q u_{t-q}.$$

Usando el operador de rezago,

$$\begin{aligned} y_t &= u_t + \theta_1 L u_t + \theta_2 L^2 u_t + \cdots + \theta_q L^q u_t \\ &= (1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q) u_t = \Theta_q(L) u_t \end{aligned}$$

El polinomio característico asociado es

$$\Theta(z) = 1 + \theta_1 z + \theta_2 z^2 + \cdots + \theta_q z^q = 0$$

Se dice que el proceso es **invertible** si todas las raíces (posiblemente complejas conjugadas), z_i , $i = 1, \dots, q$, del polinomio característico tienen módulo mayor a uno

$$|z_i| > 1, \quad i = 1, \dots, q.$$

Si un proceso $MA(q)$ es invertible, entonces este proceso se puede expresar mediante un proceso $AR(\infty)$ convergente (véase abajo para procesos autoregresivos)

$$y_t = u_t + c_1 y_{t-1} + c_2 y_{t-2} + \cdots$$

Considérese el caso de un proceso $MA(1)$

$$y_t = u_t + \theta_1 y_{t-1} = (1 + \theta_1 L)u_t$$

El polinomio característico es $1 + \theta_1 z = 0$ y el proceso es invertible si $|\theta_1| < 1$.

Usando la expresión $y_t = (1 + \theta_1 L)u_t$, aplicando formalmente el operador inverso

$$(1 + \theta_1 L)^{-1} y_t = u_t,$$

y expandiendo en series de potencia,

$$\begin{aligned} \left[\sum_{i=0}^{\infty} (-\theta_1)^i L^i \right] y_t &= u_t \\ y_t + \left[\sum_{i=1}^{\infty} (-\theta_1)^i L^i \right] y_t &= u_t \end{aligned}$$

que da la expansión $AR(\infty)$ requerida

$$y_t = u_t + c_1 y_{t-1} + c_2 y_{t-2} + \cdots$$

Invertibilidad y unicidad de un proceso $MA(q)$

La invertibilidad es también importante para identificar un proceso de media móvil.

Considérese, por ejemplo, los siguientes dos procesos de media móvil

$$\begin{aligned} x_t &= u_t + \frac{1}{2} u_{t-1}, \quad u_t \sim N(0, 4) \\ y_t &= v_t + 2v_{t-1}, \quad v_t \sim N(0, 1) \end{aligned}$$

Nótese que u_t, v_t son ambos ruidos blancos. Además, no es difícil verificar que *la función de autocorrelación ACF es la misma para ambos procesos*.

Entre los dos procesos, el primero es invertible y el segundo no lo es.

Correlogramas y memoria del proceso

- La "memoria" de un proceso $MA(q)$ es de q periodos, por lo que cualquier innovación o "shock" desaparecerá para rezagos $s > q$.
- Como consecuencia del inciso anterior, se podrá pronosticar sólo por q periodos adelante.
- La función de autocorrelación de un proceso $MA(q)$ es diferente de cero sólo para los primeros q rezagos (q : orden del proceso).
- Si el proceso $MA(q)$ es invertible, entonces la función de autocorrelación parcial (PACF) tendrá un decaimiento geométrico (exponencial), con rezagos crecientes. Véase más adelante para la definición de función de autocorrelación parcial.

Procesos autoregresivos AR(p)

Un proceso y_t **autoregresivo de orden p** se expresa como

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + u_t$$

donde u_t es el término de error (ruido blanco con media cero).

Usando el operador de rezago, se puede escribir el proceso autoregresivo mediante

$$y_t = \mu + \sum_{i=1}^p L^i y_t + u_t \quad \Leftrightarrow \quad \left(1 - \sum_{i=1}^p L^i\right) y_t = \mu + u_t \quad \Leftrightarrow \quad \Phi(L) y_t = \mu + u_t$$

donde $\Phi(L) = (1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p)$.

Comentario Sin pérdida de generalidad, se puede poner $\mu = 0$.

Estacionariedad de un modelo AR(p)

Se dice que la serie AR(p) y_t es *estacionaria* si

$$y_t = \Phi^{-1}(L) u_t$$

es convergente a cero.

Ecuación característica

La **ecuación característica** asociada al proceso $\Phi(L) y_t = \mu + u_t$ es

$$\Phi(z) = 1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p = 0$$

Resulta que el proceso $AR(p)$ es estacionario si todas las raíces z_i de la ecuación característica

$$1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p = 0$$

cumplen $|z_i| > 1$.

Ejemplo: Caminata aleatoria (Random Walk) Un caso importante de proceso *no estacionario* es una caminata aleatoria, modelada por

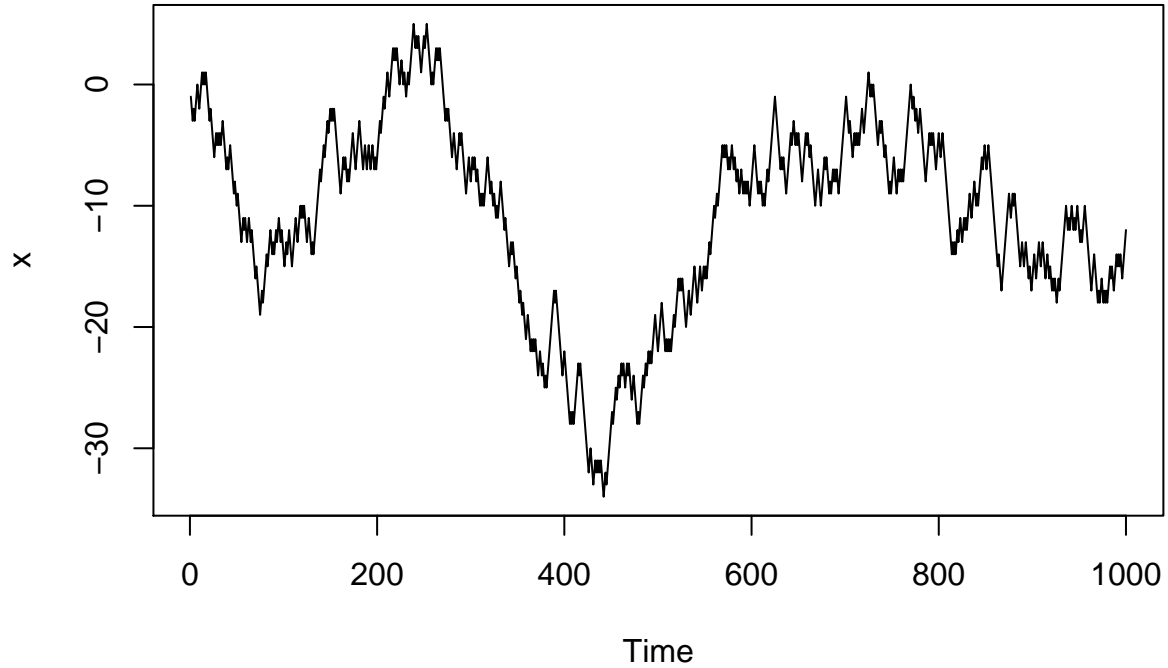
$$y_t = \mu + y_{t-1} + u_t,$$

donde u_t es un ruido blanco con media cero. Si $\mu \neq 0$, el modelo es una caminata aleatoria con deriva (“drift”, en inglés).

Simulaciones de caminatas aleatorias (con/sin deriva) se dan a continuación.

La simulación más sencilla consiste en dar un paso a la izquierda (valor = -1) con probabilidad p y un paso a la derecha (valor 1) con probabilidad $1 - p$. A continuación se presenta el caso $p = .5$:

```
set.seed(123)
n <- 1000
x <- cumsum(sample(c(-1, 1), n, replace = TRUE))
plot.ts(x)
```

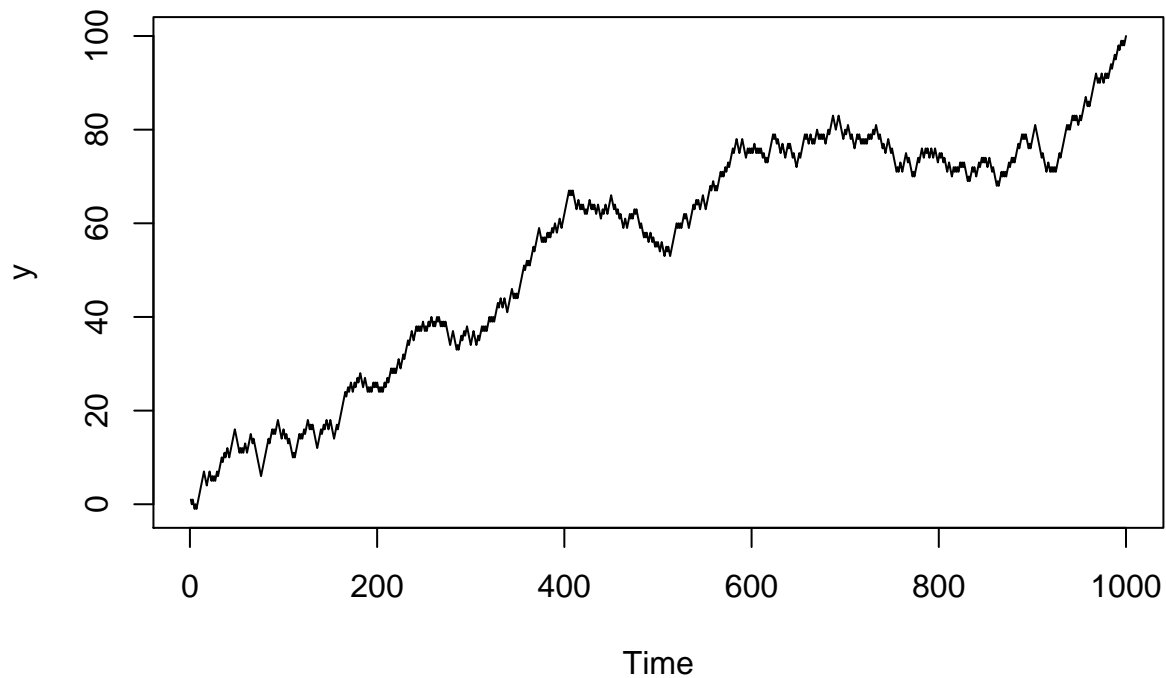


La función `sample()` toma un valor, con remplazo (`replace = TRUE`), del conjunto $\{-1, 1\}$ con probabilidad $p = .5$ (default).

En este caso, el proceso no tiene deriva, $\mu = 0$.

Simulación con probabilidad asimétrica, $p = .45$:

```
n <- 1000
y <- cumsum(sample(c(-1, 1), n, prob = c(.45, .55), replace = TRUE))
plot.ts(y)
```



En

este ejemplo, el valor esperado estimado es

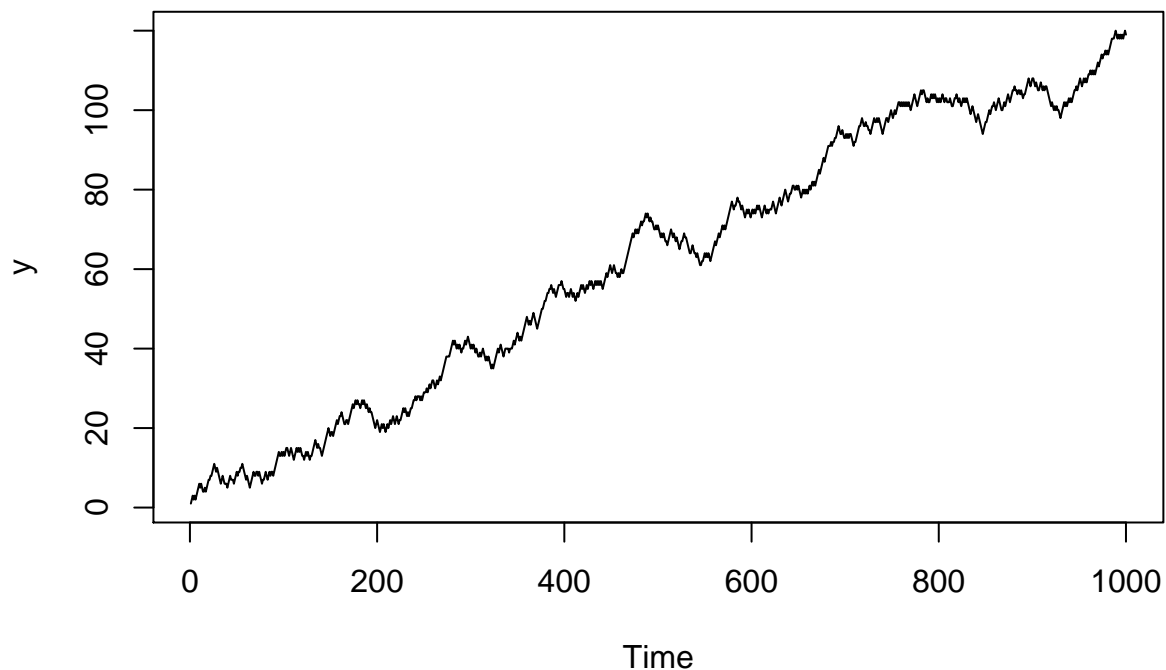
$$E[y_t] = .45(-1) + .55(1) = .1$$

por lo que se concluye que el proceso tiene una deriva estimada de .1.

Simulación de una caminata aleatoria con tres valores

$$y_t = \begin{cases} -1, & \text{con probabilidad} = .4 \\ 0, & \text{con probabilidad} = .1 \\ 1, & \text{con probabilidad} = .5 \end{cases}$$

```
n <- 1000
y <- cumsum(sample(c(-1,0, 1), n, prob = c(.4, .1, .5), replace = TRUE))
plot.ts(y)
```

El

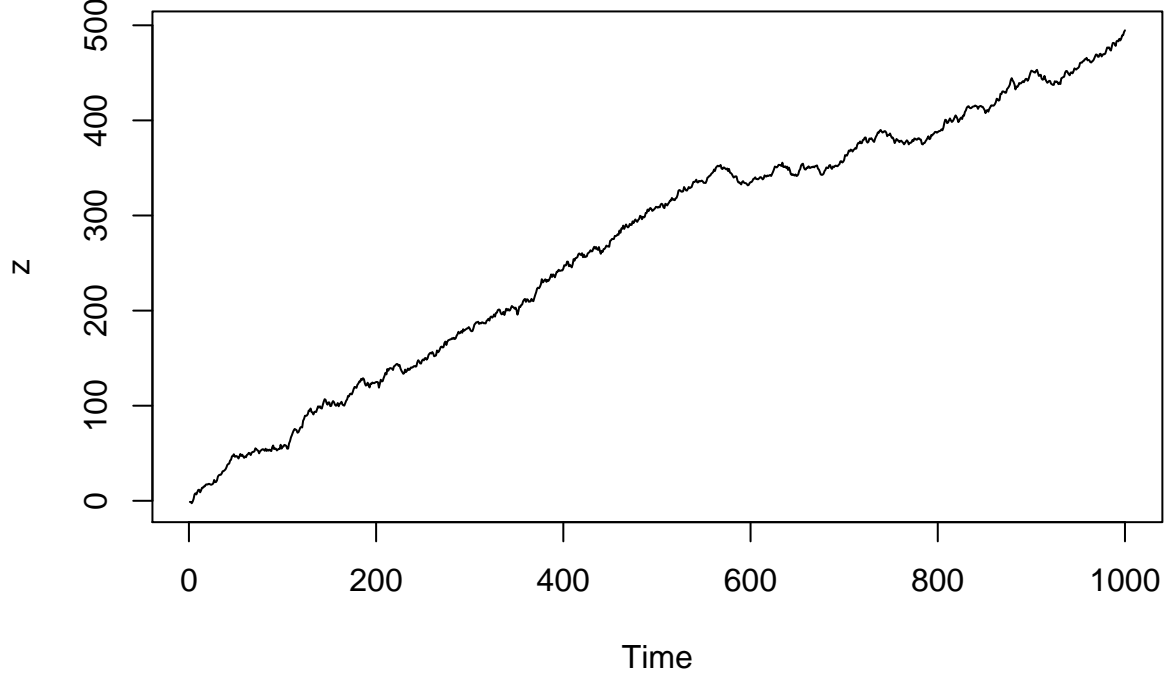
valor esperado estimado es

$$E[y_t] = .4(-1) + 0 + .5(1) = .1$$

por lo que se concluye que el proceso tiene una deriva estimada de .1.

Se puede simular una caminata aleatoria con deriva tomando muestras de una distribución como la normal. En la simulación a continuación, se muestrea una distribución normal, con media .5 y varianza 4. Nótese que en R se pasa el valor correspondiente de la desviación estándar.

```
n <- 1000
z <- cumsum(rnorm(n, mean=.5, sd = 2) )
plot.ts(z)
```



Nótese como en los tres ejemplos con deriva, la tendencia es lineal. Efectivamente, usando sustituciones repetidas en el proceso

$$\begin{aligned}
 y_t &= \mu + y_{t-1} + u_t \\
 &= \mu + (\mu + y_{t-2} + u_{t-1}) + u_t \\
 &= 2\mu + (\mu + y_{t-3} + u_{t-2}) + u_{t-1} + u_t \\
 &\vdots \\
 &= t\mu + y_0 + \sum_{i=0}^t u_{t-i}
 \end{aligned}$$

se encuentra que la tendencia es lineal. El término y_0 corresponde al valor inicial ($t = 0$) del proceso.

Integración

Si una serie no estacionaria y_t se diferencia d veces y el resultado es una serie estacionaria, entonces se dice que y_t es integrada de orden d :

$$y_t \sim I(d) \quad \Rightarrow \quad \Delta^d y_t \sim I(0).$$

Si $d = 0$, entonces la serie es estacionaria. Si $y_t \sim I(1)$, entonces la serie tiene una raíz unitaria; si $y_t \sim I(2)$ tiene dos raíces unitarias y así sucesivamente.

El operador de diferencia $\Delta^d y_t$ se define inductivamente

$$\Delta^d y_t = \Delta \left(\Delta^{d-1} y_t \right)$$

Por ejemplo,

$$\Delta^2 y_t = \Delta(\Delta y_t) = \Delta(y_t - y_{t-1}) = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2}.$$

Ejemplo 1 (Caminata aleatoria) La caminata aleatoria

$$y_t = y_{t-1} + u_t \quad \Delta y_t = u_t$$

por lo que la caminata aleatoria $\sim I(1)$.

Ejemplo 2 Considera $y_t = 3y_{t-1} + 4y_{t-2} + u_t$. La serie tiene polinomio característico

$$1 - 3z - 4z^2 = 0$$

que tiene raíces

```
roots <- polyroot(c(1, -3, -4))
```

$z_1 = 1/4$, $z_2 = -1$, de manera que

$$(1 - 3L - 4L^2)y_t = (1 - z_1^{-1}L)(1 - z_2^{-2}L)y_t = (1 - 4L)(1 + L)y_t = u_t$$

Nótese que $z_t = (1 - 4L)y_t$ es una serie estacionaria, por lo que $y_t \sim I(1)$. Además,

$$z_t = -z_{t-1} + u_t.$$

Sobrep parametrización

Si una serie diferenciada es estacionaria, diferenciarla una, o más veces, puede causar una estructura no invertible $MA(q)$ de las innovaciones.

Por ejemplo, considera la caminata aleatoria $y_t = y_{t-1} + u_t$. La serie diferenciada

$$\Delta y_t = u_t$$

es estacionaria; diferenciando una vez más,

$$\Delta(\Delta y_t) = \Delta u_t = u_t - u_{t-1}$$

que es un proceso $MA(1)$ no invertible.

Pruebas de hipótesis de estacionariedad

Es importante disponer de pruebas de hipótesis de estacionariedad de una serie de tiempo. Hay dos tipos de enfoque:

- Prueba de raíz unitarias, (prueba de Dickey-Fuller aumentada; prueba de Phillips-Perron).
- Prueba de estacionariedad, prueba KPSS (Kwiatkowski-Phillips-Schmidt-Shin).

Prueba de raíz unitaria

En la prueba de raíz unitaria, la prueba de hipótesis asociada es

H_0 : la serie tiene una raíz unitaria

H_1 : la serie no tiene una raíz unitaria

En una prueba de hipótesis de raíz unitaria, cuando se rechaza H_0 se determina que la serie es estacionaria.

La idea atrás de una prueba de raíz unitaria es la siguiente; usaremos una serie $AR(1)$ por simplicidad. Considera

$$y_t = \phi y_{t-1} + u_t.$$

El objetivo es determinar si $\phi = 1$ (presencia de raíz unitaria, es decir, serie no estacionaria), contra la alternativa $\phi < 1$ (serie estacionaria).

Restando y_{t-1} a ambos lados

$$\Delta y_t = \overbrace{(\phi - 1)}^{\psi} y_{t-1} + u_t,$$

por lo que

$$H_0 : \phi = 1 \quad \Leftrightarrow \quad \psi = 0$$

$$H_1 : \phi < 1 \quad \Leftrightarrow \quad \psi < 0$$

La prueba de Dickey-Fuller puede considerar una serie de tiempo con tendencia lineal

$$y_t = \phi y_{t-1} + \alpha + \beta t + u_t.$$

La idea de la prueba de Phillips-Perron es semejante, pero incluye una corrección para residuales autocorrelados.

Las dos pruebas de raíces unitarias tienen resultados semejante y tienen la misma desventaja, porque la *potencia* de la prueba no es muy buena, especialmente con muestra pequeñas y con raíces cercana a uno.

Se ilustrará esta afirmación con dos simulaciones de una serie estacionaria,

$$y_t = .95y_{t-1} + u_t,$$

con $n = 20$, $n = 1000$, respectivamente. Se usará, como de costumbre, $\alpha = .05$ como nivel de significancia.

```
library(tseries)
```

```
## Registered S3 method overwritten by 'xts':  
##   method      from  
##   as.zoo.xts zoo
```

```
## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo

set.seed(123)
z20 <- arima.sim(20, model = list(ar = .95))
adf.test(z20)

##
## Augmented Dickey-Fuller Test
##
## data:  z20
## Dickey-Fuller = -2.4844, Lag order = 2, p-value = 0.3879
## alternative hypothesis: stationary

pp.test(z20)
```

```
##
## Phillips-Perron Unit Root Test
##
## data:  z20
## Dickey-Fuller Z(alpha) = -12.977, Truncation lag parameter = 2,
## p-value = 0.2757
## alternative hypothesis: stationary
```

Con una muestra pequeña, no se rechaza la hipótesis nula cuando es falsa. La potencia de la prueba no es buena.

Por otro lado, con una muestra grande, $n = 1000$, la potencia es buena:

```
library(tseries)
set.seed(123)
z1000 <- arima.sim(1000, model = list(ar = .95))
adf.test(z1000)

## Warning in adf.test(z1000): p-value smaller than printed p-value
##
## Augmented Dickey-Fuller Test
##
## data:  z1000
## Dickey-Fuller = -4.436, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary

pp.test(z1000)

## Warning in pp.test(z1000): p-value smaller than printed p-value
##
## Phillips-Perron Unit Root Test
```

```
##
## data:  z1000
## Dickey-Fuller Z(alpha) = -53.394, Truncation lag parameter = 7,
## p-value = 0.01
## alternative hypothesis: stationary
```

Se rechaza H_0 cuando es falsa.

Prueba de estacionariedad

En la prueba KPSS, la prueba de hipótesis asociada es

H_0 : la serie es estacionaria

H_1 : la serie no es estacionaria

El desarrollo de la prueba KPSS va más allá del alcance de estas notas, pero la idea es de usar mínimos cuadrados ordinarios para estimar el modelo

$$y_t = \beta' \mathbf{D}_t + \mu_t + u_t$$

$$\mu_t = \mu_{t-1} + \varepsilon_t,$$

donde el término $\beta' \mathbf{D}_t$ contiene un término constante o un término con tendencia lineal; u_t es estacionaria, posiblemente heterocedástica; el término ε_t es un ruido blanco con varianza σ_ε^2 . El término central en la prueba KPSS es la segunda ecuación, que modela μ_t mediante una caminata aleatoria. La hipótesis nula de la prueba es $H_0 : \sigma_\varepsilon^2 = 0$, es decir, μ_t es una constante; la alternativa es $H_1 : \sigma_\varepsilon^2 > 0$.

La librería `tseries` incluye una implementación de la prueba KPSS:

```
library(tseries)
set.seed(123)
z20 <- arima.sim(20, model = list(ar = .95))
kpss.test(z20)
```

```
## Warning in kpss.test(z20): p-value greater than printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data:  z20
## KPSS Level = 0.31264, Truncation lag parameter = 2, p-value = 0.1
```

```
library(tseries)
set.seed(123)
z1000 <- arima.sim(1000, model = list(ar = .95))
kpss.test(z1000)
```

```
##
## KPSS Test for Level Stationarity
##
## data: z1000
## KPSS Level = 0.42803, Truncation lag parameter = 7, p-value =
## 0.06507
```

En ambas simulaciones no se rechaza la hipótesis nula de estacionariedad.

Análisis de confirmación

El análisis de confirmación (“confirmatory data analysis”, en inglés) es el uso simultáneo de una prueba de raíz unitaria y de estacionariedad. Siguiendo la notación en el libro de Brooks, p. 365, denotaremos con $y_t \sim I(1)$ una serie no estacionaria y con $y_t \sim I(0)$ una estacionaria.

La hipótesis nula y alternativa de cada enfoque son

ADF/PP	KPSS
$H_0 : y_t \sim I(1)$	$H_0 : y_t \sim I(0)$
$H_1 : y_t \sim I(0)$	$H_0 : y_t \sim I(1)$

Hay cuatro posibles resultados:

ADF/PP	KPSS
Rechazar H_0	No rechazar H_0
No rechazar H_0	Rechazar H_0
Rechazar H_0	Rechazar H_0
No rechazar H_0	No rechazar H_0

Un resultado robusto es el primer par (estacionariedad) o el segundo (no estacionariedad). Los últimos dos pares dan resultados conflictivos.

Características de un proceso AR(p)

Teorema de Wold

Cualquier proceso estacionario se puede escribir mediante la suma de una parte determinística y de un proceso estocástico, este último como $MA(\infty)$.

El Teorema de Wold es intuitivamente claro en el caso de procesos autoregresivos. Considérese, por ejemplo, el caso $AR(1)$,

$$y_t = \phi_1 y_{t-1} + u_t$$

donde, si perder de generalidad, se puede suponer $\mu = 0$. Entonces,

$$\begin{aligned}y_t &= \phi_1 y_{t-1} + u_t \\y_{t-1} &= \phi_1 y_{t-2} + u_{t-1}.\end{aligned}$$

Sustituyendo, se encuentra

$$y_t = \phi_1(\phi_1 y_{t-2} + u_{t-1}) + u_t = \phi_1^2 y_{t-2} + \phi_1 u_{t-1} + u_t$$

Expresando el proceso y_t mediante sus rezagos sucesivos, y pasando al límite, se encuentra que el proceso $AR(1)$ se puede expresar mediante una serie de rezagos sucesivos de un ruido blanco con media cero.

A continuación se probarán las propiedades más básicas de un proceso $AR(p)$. Para simplificar el procedimiento, se ilustrará el caso $p = 1$ y después se dará el caso general $p > 1$.

Media incondicional

Se procede por pasos sucesivos.

$$\begin{aligned}E[y_t] &= E[\mu + \phi_1 y_{t-1} + u_t] \\&= \mu + \phi_1 E[y_{t-1}] = \mu + \phi_1 E[\mu + \phi_1 y_{t-2} + u_{t-1}] \\&= (\mu + \phi_1 \mu) + \phi_1^2 E[y_{t-2}]\end{aligned}$$

Después de n rezagos,

$$E[y_t] = (1 + \phi_1 + \phi_1^2 + \cdots + \phi_1^n) \mu + \phi_1^n E[y_{t-n}]$$

El modelo es estacionario, $|\phi_1| < 1$, por lo que $\phi_1^n \rightarrow 0$, cuando $n \rightarrow \infty$; por ende,

$$E[y_t] = \lim_{n \rightarrow \infty} \mu(1 + \cdots + \phi_1^n) = \mu \sum_{n=1}^{\infty} \phi_1^n = \frac{\mu}{1 - \phi_1}.$$

En el caso general,

$$E[y_t] = \frac{\mu}{1 - \phi_1 - \phi_2 \cdots - \phi_p}.$$

Varianza incondicional

Sin pérdida de generalidad, se supondrá $\mu = E[y_t] = 0$, por lo que

$$\text{var}(y_t) = E[y_t^2].$$

El Teorema de Wold implica

$$\begin{aligned}y_t &= \phi_1 y_{t-1} + u_t \\&= u_t + \phi_1 u_{t-1} + \phi_1^2 u_{t-2} + \cdots\end{aligned}$$

por lo que

$$y_t^2 = u_t^2 + \phi_1^2 u_{t-1}^2 + \phi_1^4 u_{t-2}^2 + \cdots + \text{productos cruzados.}$$

Entonces,

$$\begin{aligned} \text{var}(y_t) &= (1 + \phi_1^2 + \phi_1^4 + \cdots) \sigma^2 = \sum_{i=0}^{\infty} (\phi_1^2)^i \sigma^2 \\ &= \frac{\sigma^2}{1 - \phi_1^2}. \end{aligned}$$

Denotaremos $\text{var}(y_t) = \gamma_0$.

El caso general es semejante.

Correlaciones seriales

Las correlaciones seriales τ_s , $s = 1, 2, \dots$, son los coeficientes de la función de autocorrelación (ACF) y están definidas

$$\tau_s = \frac{\gamma_s}{\gamma_0} = \frac{\text{cov}(y_t, y_{t-s})}{\text{var}(y_t)}.$$

Sin falta de generalidad, se supondrá $\mu = 0 = E[y_t]$.

Analizaremos primeramente el caso $s = 1$.

Usando el Teorema de Wold,

$$\begin{aligned} y_t &= u_t + \phi_1 u_{t-1} + \phi_1^2 u_{t-2} + \cdots \\ y_{t-1} &= u_{t-1} + \phi_1 u_{t-2} + \phi_1^2 u_{t-3} + \cdots \end{aligned}$$

de manera que

$$y_t y_{t-1} = \phi_1 u_{t-1}^2 + \phi_1^3 u_{t-2}^2 + \phi_1^5 u_{t-3}^2 + \cdots + \text{productos cruzados}$$

Calculando los valores esperados,

$$\begin{aligned} E[y_t y_{t-1}] &= E[\phi_1 u_{t-1}^2] + E[\phi_1^3 u_{t-2}^2] + E[\phi_1^5 u_{t-3}^2] \\ &\quad + \cdots + E[\text{productos cruzados}] \end{aligned}$$

por lo que

$$\text{cov}(y_t, y_{t-1}) = \phi_1 \sigma^2 \left[\sum_{i=0}^{\infty} (\phi_1^2)^i \right] = \frac{\phi_1 \sigma^2}{1 - \phi_1^2}$$

Por lo que $\tau_1 = \phi_1$.

El caso $s > 1$ se analiza de una manera semejante y se encuentra que

$$\tau_s = \phi_1^s$$

Función de autocorrelación parcial (PACF)

Usaremos la definición que se encuentra en el libro de [T], Secc. 2.4.2, pp. 59-61.

La PACF (función de autocorrelación parcial) de una serie de tiempo estacionaria es una función de la ACF y se puede introducir de la siguiente manera:

$$\begin{aligned}y_t &= \phi_{0,1} + \phi_{1,1}y_{t-1} + e_{1t} \\y_t &= \phi_{0,2} + \phi_{1,2}y_{t-1} + \phi_{2,2}y_{t-2} + e_{2t} \\y_t &= \phi_{0,3} + \phi_{1,3}y_{t-1} + \phi_{2,3}y_{t-2} + \phi_{3,3}y_{t-3} + e_{3t} \\&\vdots = \vdots\end{aligned}$$

La PACF muestral con rezago s , $\hat{\phi}_{s,s}$, da la contribución adicional del término de rezago, y_{t-s} , a la serie y_t sobre el modelo $AR(s-1)$.

Por lo tanto, en un modelo $AR(p)$, la PACF muestral $\hat{\phi}_{j,j} \approx 0$, para todo $j > p$.

Se puede mostrar, para un modelo $AR(p)$ con errores normales, que

- $\hat{\phi}_{p,p} \rightarrow \phi_p$, si el tamaño muestral, $T \rightarrow \infty$.
- $\hat{\phi}_{j,j} \rightarrow 0$, para todo $j > p$.
- La varianza asintótica de $\hat{\phi}_{j,j} = 1/T$, para $j > p$.

En conclusión, la PACF muestral de un modelo $AR(p)$ es igual a cero, al rezago j , $j > p$.

Correlogramas y memoria de un proceso AR; comparaciones ACF, PACF para procesos MA, AR

Si $y_t = \mu + \sum_{i=1}^p L^i y_t + u_t$ es un proceso $AR(p)$. Entonces

- un modelo $AR(p)$ estacionario tendrá un correlograma que tiende a cero de manera geométrica (exponencial; posiblemente de carácter oscilatorio);
- el Teorema de Wold permite expresar un modelo $AR(p)$ por medio de un $MA(\infty)$, por lo que la "memoria" de las innovaciones u_t persiste de manera indefinida; en el caso de un proceso estacionario, la memoria decaerá de manera geométrica.
- La función de autocorrelación parcial decaerá a cero después de p rezagos (p : orden del proceso autoregresivo), porque existen relaciones entre y_t y sus primeros p rezagos pero y_t no tendrá ninguna relación con y_{t-s} , $s > p$.
- Considerando las relaciones entre procesos $AR(p)$ estacionarios con $MA(\infty)$ y de procesos $MA(q)$ invertibles con $AR(\infty)$ se puede concluir que la ACF de un proceso AR tiene la misma forma de una PACF de un proceso MA ; la ACF de un proceso MA tiene la misma forma de una PACF de un proceso AR .

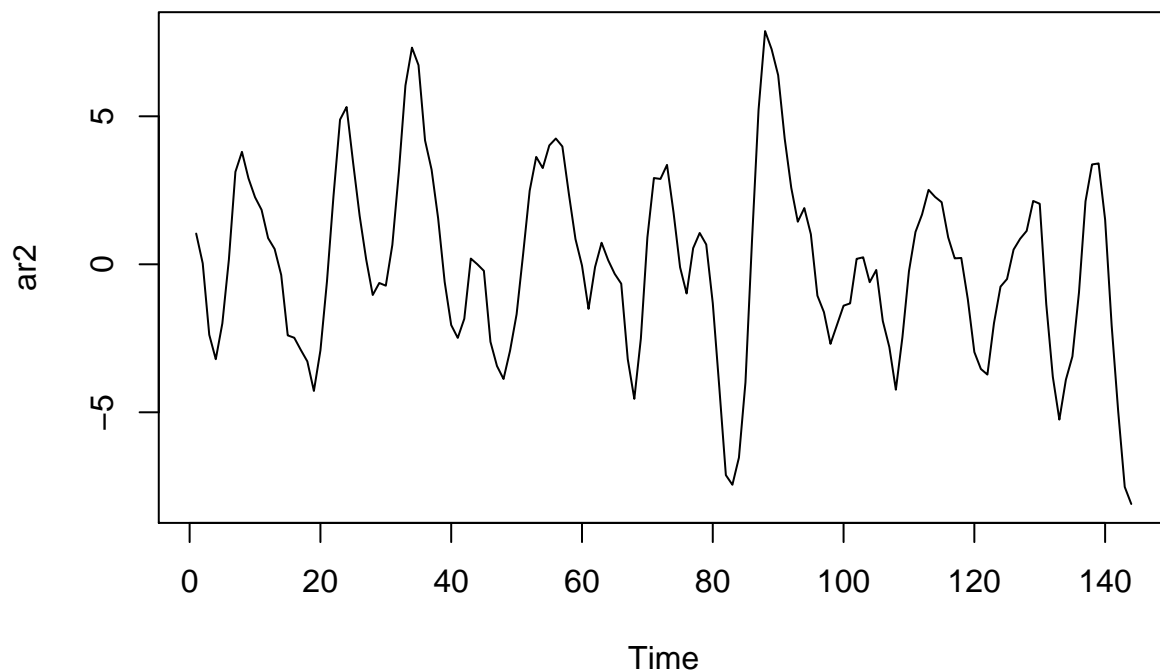
Ejemplo

Se presenta a continuación una simulación de la serie $AR(2)$,

$$y_t = 1.5y_{t-1} - .75y_{t-2} + u_t.$$

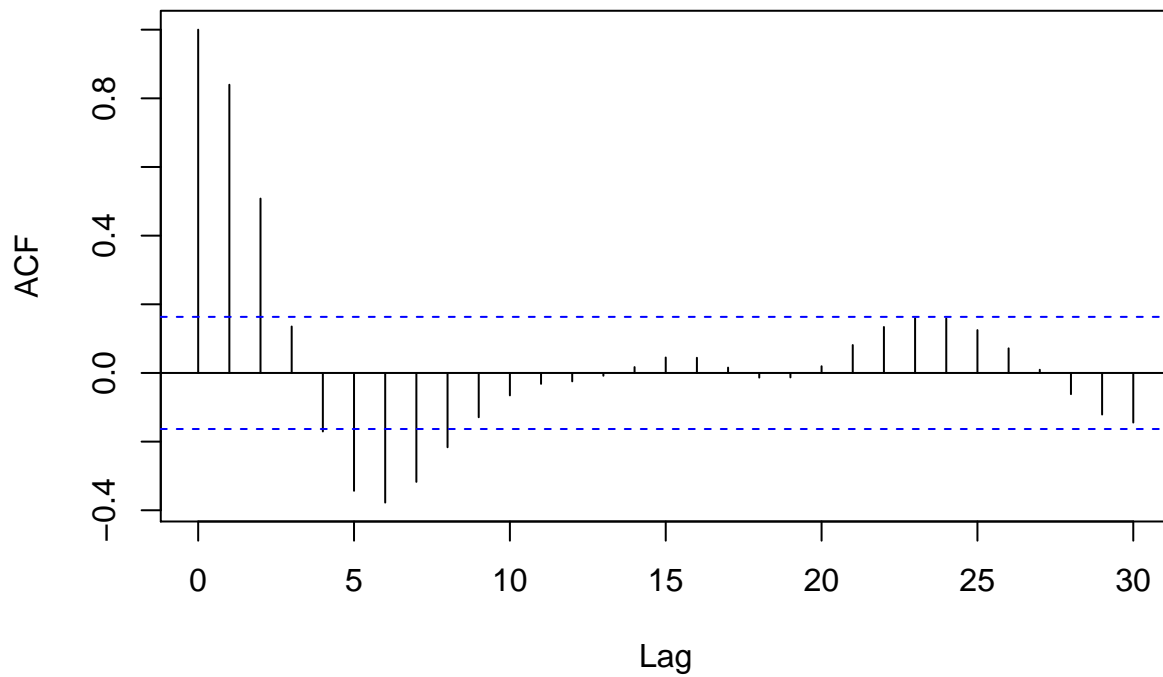
Esta serie presenta las características “ideales” mencionadas arriba; además, la serie presenta un carácter oscilatorio y se mostrará como se pueden usar las raíces complejas conjugadas del polinomio característico para estimar el ciclo.

```
set.seed(12321)
ar2 <- arima.sim(n=144, model = list(ar=c(1.5, -.75)))
plot.ts(ar2)
```



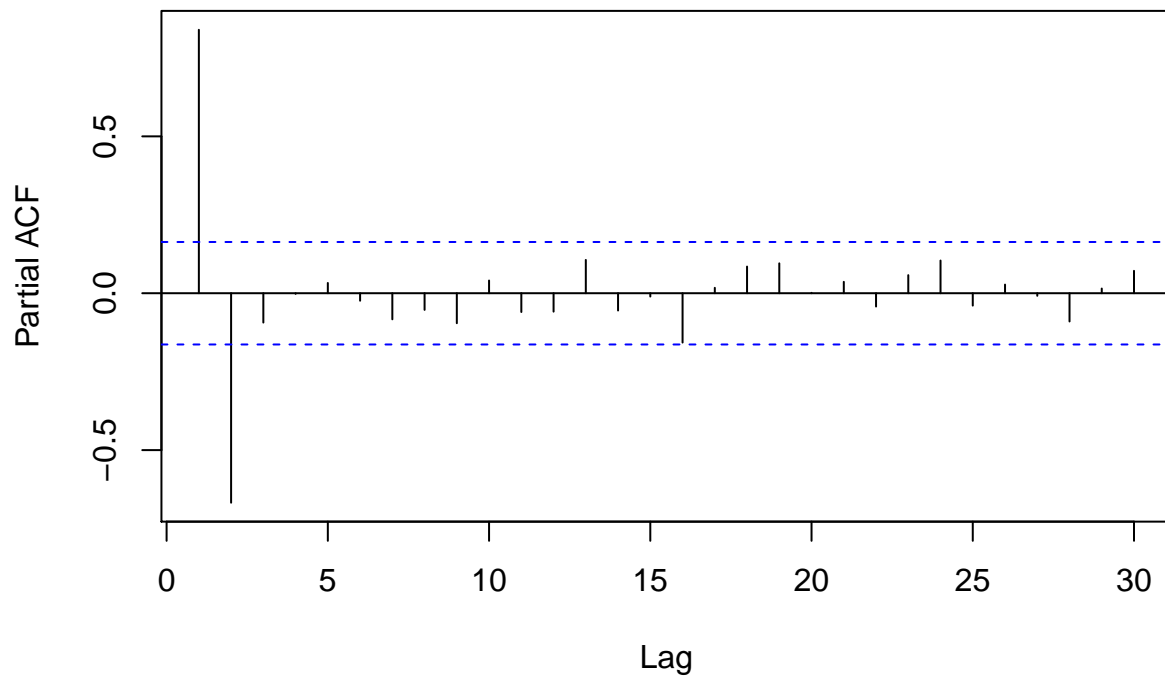
```
acf(ar2, lag.max = 30)
```

Series ar2



```
pacf(ar2, lag.max = 30)
```

Series ar2



El

cálculo de las raíces del polinomio característico asociado

$$\Phi(z) = 1 - 1.5z + .75z^2 = 0$$

```
z = c(1, -1.5, .75)
root <- polyroot(z)
```

El ciclo se estima usando el *argumento* de una de las raíces; el argumento está en radianes per unidad de tiempo, por lo que lo transformamos en ciclos

```
Arg(root[1]) # argumento en radianes/tiempo
```

```
## [1] 0.5235988
```

```
arg <- Arg(root[1])/(2*pi) # argumento en ciclos/tiempo
1/arg # periodo estimado
```

```
## [1] 12
```

Modelos ARMA, ARIMA

Un proceso $ARMA(p, q)$ es una serie de tiempo con características estacionaria autoregresivas y de media móvil:

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 u_{t-1} + \theta_2 u_{t-2} + \cdots + \theta_q u_{t-q} + u_t$$

Usando el operador de rezago, se puede escribir el proceso $ARMA(p, q)$ como

$$\Phi(L)y_t = \mu + \Theta(L)u_t$$

Todas las raíces de la ecuación característica asociada a $\Phi(L)$ tienen módulo menor a 1.

También se pedirá que la parte de media móvil del proceso sea invertible; véase la sección de invertibilidad y unicidad de un proceso $MA(q)$.

Sobrep parametrización

El material de esta sección se puede también encontrar en el libro de [ShS], Sección 3.1.

La sobreparametrización de un proceso ARMA ocurre cuando las ecuaciones características asociadas a $\Phi(L)$, $\Theta(L)$ comparten una raíz.

Ejemplo

Considèrese el modelo $ARMA(1, 1)$

$$y_t = -.8y_{t-1} + .8u_{t-1} + u_t$$

Usando el operador de rezago, podemos escribir

$$(1 + .8L)y_t = (1 + .8L)u_t$$

que es lo mismo que un proceso de ruido blanco

$$y_t = u_t$$

Siguiendo [ShS], se generará una muestra $n = 1000$ de un ruido blanco $u_t \sim N(10, 1)$ y se ajustarán los valores a un modelo $ARMA(1, 1)$:

```
set.seed(1111)
x <- rnorm(1000, mean = 10)
arima(x, order = c(1,0,1))

##
## Call:
## arima(x = x, order = c(1, 0, 1))
##
## Coefficients:
##          ar1          ma1  intercept
##       0.9901   -0.9768       9.9923
## s.e.  0.0105    0.0152    0.0728
##
## sigma^2 estimated as 0.9748:  log likelihood = -1406.24,  aic = 2820.49
```

La estimación del modelo es

$$(1 - .9901L)y_t = (1 - .9768)u_t + 9.9923$$

Se puede observar como los valores estimados $\hat{\phi}_1, \hat{\theta}_1$, se encuentran en el intervalo de confianza (95%) del otro, por lo que podemos deducir que se trata de un modelo sobreparametrizado.

Observando la estimación, vale la pena mencionar que los que se indica como **intercept** (intercepto) es en realidad la estimación de la media μ .

Características de un proceso $ARMA(p, q)$

Las características de un proceso $ARMA(p, q)$ son una combinación de un proceso autoregresivo y de media móvil.

- El valor esperado (media) del proceso es

$$E(y_t) = \frac{\mu}{1 - \phi_1 - \phi_2 - \dots - \phi_p}.$$

- La función de autocorrelación ACF tendrá un patrón decreciente geométricamente.
- La función de autocorrelación parcial PACF tendrá un patrón decreciente geométricamente.
- Después de q rezagos (q : orden de la parte de media móvil) la parte AR dominará el proceso, especialmente las funciones ACF, PACF.

La metodología de Box-Jenkins

Este procedimiento estructura la estimación de los procesos $ARMA(p, q)$:

1. Identificación
2. Estimación
3. Diagnóstico

Identificación En esta fase, se usan la forma de los correlogramas (AFC; PACF) para determinar el posible modelo ($AR(p)$, $MA(q)$, $ARMA(p, q)$), como se explicó arriba.

Estimación Una vez identificado el modelo, se estima el modelo usando técnicas como MCO o máxima verosimilitud. En R, hay también una función `auto.arima()` (librería `forecast`) que permite estimar un modelo minimizando uno de los criterios de información disponibles (véase abajo para la definición de los criterios de información más usados). El uso de esta función puede tener un efecto de sobreestimación del modelo, que no es necesariamente deseable; véase también la fase de diagnóstico.

Diagnóstico Una vez estimado el modelo, con valores p , q apropiados, se determina si el modelo se puede reducir, considerando la significancia estadística de los coeficientes estimados.

Para determinar si el modelo necesita términos adicionales, se puede usar el método de “overfitting”, que consiste en la estimación de un modelo con orden mayor; en la comparación se usarán algunos de los criterios de información y la significancia estadística de los parámetros adicionales.

Una vez determinado, y estimado, el modelo apropiado, se determinará si los residuales tienen autocorrelación serial, individualmente y en conjunto. Cuando se lleva a cabo una prueba portmanteau en R, usando la función `Box.test()`, es importante pasar el parámetro adicional `fitdf = p+q` para restar el número de grados de libertad del modelo estimado (con los valores apropiados p , q).

Puede ser conveniente usar pruebas de normalidad de los errores, aún si, en muestras grandes, se pueden usar versiones del Teorema Central del Límite.

Una regla general es que el modelo sea *parsimonioso*, de manera que capture la *señal* y no el *ruido*.

Para los propósitos de estas notas, se usarán modelos estacionarios, por lo que no se usarán modelos $ARIMA(p, d, q)$ explícitamente.

Criterios de Información

La idea del criterio de información es *penalizar* el uso excesivo de parámetros a estimar; en este sentido, es una herramienta más sofisticada del R^2 ajustado que se usa en regresión lineal.

Su uso es también justificado cuando se usan datos reales “ruidosos” porque el uso de los correlogramas puede no ser tan claros, como menciona la teoría.

La idea general de un criterio de información es de incluir en una fórmula el número de parámetros del modelo y *minimizar* esta fórmula.

A continuación se presentan los criterios de información más usados: AIC (criterio de información de Akaike), (S)BIC (criterio de información bayesiano de Schwarz) y HQIC (criterio de información de Hannan-Quinn).

En todos, T es el tamaño muestral; $k = p + q + 1$ es el número de parámetros a estimar; $\hat{\sigma}^2$ es la varianza residual, $\hat{\sigma}^2 = SRC/T$:

$$\begin{aligned} AIC &= \ln(\hat{\sigma}^2) + \frac{2k}{T} \\ (S)BIC &= \ln(\hat{\sigma}^2) + \frac{k}{T} \ln(T) \\ HQIC &= \ln(\hat{\sigma}^2) + \frac{2k}{T} \ln(\ln(T)) \end{aligned}$$

De los tres criterios, el que más penaliza el número de parámetros a estimar es el SBIC y el que penaliza menos es el AIC; el HQIC se encuentra en el medio.

En términos estadísticos, (S)BIC consistente, pero ineficiente; AIC es, por lo general, eficiente, pero no es consistente.

La varianza de las estimaciones será, por lo general, mayor usando (S)BIC que AIC.

En término de elección del modelo, cuando $T \rightarrow \infty$ (muestras muy grandes) (S)BIC proporciona el modelo correcto y AIC proporciona un modelo no parsimonioso. Si aplica la Ley de los Grandes Números, (S)BIC puede ser la mejor elección, porque la varianza, de todas maneras tiende a cero, cuando $T \rightarrow \infty$.

Pronósticos

Indicamos con Ω_t toda la información disponible hasta el tiempo t (incluido).

Un *pronóstico*, al tiempo $t + h$, se genera por medio de un valor esperado, condicionado sobre Ω_t :

$$f_{t,h} = E[y_{t+h}|\Omega_t]$$

Método de pronóstico simple

1. Si se supone que no habrá cambios, entonces un pronóstico simple, h pasos en adelante, es de usar el valor actual

$$E[y_{t+h}|\Omega_t] = y_t$$

Este tipo de pronóstico es óptimo, si el proceso es una caminata aleatoria:

$$y_{t+1} = y_t + u_t.$$

porque, en este caso,

$$E[y_{t+1}|\Omega_t] = E[y_t + u_t|\Omega_t] = E[y_t|\Omega_t] + E[u_t|\Omega_t] = y_t$$

2. Si la serie es estacionaria (“mean-reverting”), otro método de pronóstico simple es usar la media incondicionada.

Pronóstico mediante suavización exponencial

La técnica de suavización exponencial considera que se puede determinar un pronóstico usando una combinación de las últimas observaciones; la pregunta es cuánto peso se debe asignar a estas observaciones.

Una desventaja de una modelación ARMA (además de la determinación del número de parámetros a estimar), es que un modelo $MA(q)$ tiene una “memoria” limitada sólo a los últimos q rezagos, mientras que un modelo $AR(p)$ tiene una memoria indefinida. De hecho, un suavizado exponencial es un caso especial de un modelo $ARIMA(0, 1, 1)$, con coeficientes MA iguales a $(1 - \alpha)$.

La ventaja de una modelación de suavización exponencial es que permite dar un peso a observaciones en el pasado, sin limitarse necesariamente a las últimas más recientes, como en un modelo de media móvil. El peso se asigna mediante una relación recursiva, de la siguiente manera. Si S_t es el valor actual suavizado, entonces

$$S_t = \alpha y_t + (1 - \alpha)S_{t-1},$$

con $0 < \alpha < 1$.

El valor actual S_t depende de una manera geoméricamente decreciente de los valores pasados; sustituyendo hacia atrás

$$\begin{aligned} S_t &= \alpha y_t + (1 - \alpha)S_{t-1} \\ &= \alpha y_t + (1 - \alpha) [\alpha y_{t-1} + (1 - \alpha)S_{t-2}] \\ &= \alpha [y_t + (1 - \alpha)y_{t-1}] + (1 - \alpha)^2 S_{t-2} \\ &= \alpha [y_t + (1 - \alpha)y_{t-1} + (1 - \alpha)^2 y_{t-2}] + (1 - \alpha)^3 S_{t-3} \\ &\vdots \quad (\text{después de } n \text{ sustituciones hacia atrás}) \\ &= \alpha \left[\sum_{i=0}^n (1 - \alpha)^i y_{t-i} \right] + (1 - \alpha)^{n+1} S_{t-(n+1)}. \end{aligned}$$

Cuando $n \rightarrow \infty$, el valor suavizado actual se expresa mediante una combinación geométrica de sus valores anteriores.

El pronóstico de una suavización exponencial es, entonces,

$$f_{t,s} = S_t, \quad s = 1, 2, \text{ dots}$$

El método de suavización exponencial se puede extender a series con tendencias (método de Holt) o estacionalidades (método de Winter).

Pronóstico para MA(q)

Por simplicidad, se considera el caso $q = 2$; el caso general se deriva de manera directa.

Sea

$$y_t = \mu + \theta_1 u_{t-1} + \theta_2 u_{t-2} + u_t$$

Entonces,

$$\begin{aligned} f_{t,1} &= \mu + \theta_1 u_t + \theta_2 u_{t-1} \\ f_{t,2} &= \mu + \theta_2 u_t \\ f_{t,3} &= f_{t,s} = \mu, \quad s > 3. \end{aligned}$$

La prueba es directa, ya que

$$y_{t+1} = \mu + \theta_1 u_t + \theta_2 u_{t-1} + u_{t+1}$$

por lo que

$$E[y_{t+1}|\Omega_t] = E[\mu + \theta_1 u_t + \theta_2 u_{t-1} + u_{t+1}|\Omega_t] = \mu + \theta_1 u_t + \theta_2 u_{t-1},$$

ya que u_{t+1} es independiente de u_s , $s \leq t$. Un paso más,

$$E[y_{t+2}|\Omega_t] = E[\mu + \theta_1 u_{t+1} + \theta_2 u_t + u_{t+2}|\Omega_t] = \mu + \theta_2 u_t.$$

Finalmente,

$$E[y_{t+s}|\Omega_t] = E[\mu + \theta_1 u_{t+s-1} + \theta_2 u_{t+s-2} + u_{t+s}|\Omega_t] = \mu,$$

para todo $s \geq 3$.

Como ya se comentó arriba, un modelo $MA(q)$ sólo tiene memoria de los últimos q rezagos y esto se ve reflejado también en el pronóstico.

Pronóstico para AR(p)

Por simplicidad, también se considera el caso $p = 2$.

Sea

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + u_t$$

Entonces,

$$\begin{aligned}f_{t,1} &= \mu + \phi_1 y_t + \phi_2 y_{t-1} \\f_{t,2} &= \mu + \phi_1 f_{t,1} + \phi_2 y_t \\f_{t,3} &= \mu + \phi_1 f_{t,2} + \phi_2 f_{t,1} \\f_{t,s} &= \mu + \phi_1 f_{t,s-1} + \phi_2 f_{t,s-2}\end{aligned}$$

La prueba es directa, ya que

$$y_{t+1} = \mu + \phi_1 y_t + \phi_2 y_{t-1} + u_{t+1}$$

por lo que

$$E[y_{t+1}|\Omega_t] = E[\mu + \phi_1 y_t + \phi_2 y_{t-1} + u_{t+1}|\Omega_t] = \mu + \phi_1 y_t + \phi_2 y_{t-1}.$$

Un paso más,

$$\begin{aligned}E[y_{t+2}|\Omega_t] &= E[\mu + \phi_1 y_{t+1} + \phi_2 y_t + u_{t+2}|\Omega_t] \\&= \mu + \phi_1 E[y_{t+1}|\Omega_t] + \phi_2 y_t \\&= \mu + \phi_1 f_{t,1} + \phi_2 y_t,\end{aligned}$$

ya que $f_{t,1} = E[y_{t+1}|\Omega_t]$. Finalmente,

$$E[y_{t+s}|\Omega_t] = E[\mu + \phi_1 u_{t+s-1} + \phi_2 u_{t+s-2} + u_{t+s}|\Omega_t] = \mu + \phi_1 f_{t,s-1} + \phi_2 f_{t,s-2},$$

para todo $s \geq 3$.

Como se ve también en el pronóstico, un proceso autoregresivo tiene una memoria “larga”.

Pronósticos con R

Ejemplo 1 (Simulación)

A continuación presentaremos una implementación en R del pronóstico de una serie simulada $AR(2)$.

Para ilustrar también la metodología Box-Jenkins, se seguirán los pasos de este procedimiento y se estimará la serie simulada. En el caso de una serie simulada, el procedimiento de Box-Jenkins funciona de manera ideal. En el caso de series de tiempo reales, las cosas son un poco más complicadas, por lo que podría ser útil el uso de herramientas como `auto.arima()` (librería `forecast`), `eacf()` (librería `TSA`) para ayudar en la selección del orden apropiado.

Para determinar la bondad del pronóstico, se usarán el 80% de las observaciones para generar el pronóstico y se usará el restante 20% (“holdout sample”) para construir pronósticos fuera de muestra.

```

set.seed(123)
library(tseries) # tseries::adf.test
library(forecast) # forecast::forecast()

## Registered S3 methods overwritten by 'forecast':
##   method          from
##   fitted.fracdiff  fracdiff
##   residuals.fracdiff fracdiff

library(TSA) # tseries::eacf() extended ACF

## Registered S3 methods overwritten by 'TSA':
##   method      from
##   fitted.Arima forecast
##   plot.Arima   forecast

##
## Attaching package: 'TSA'

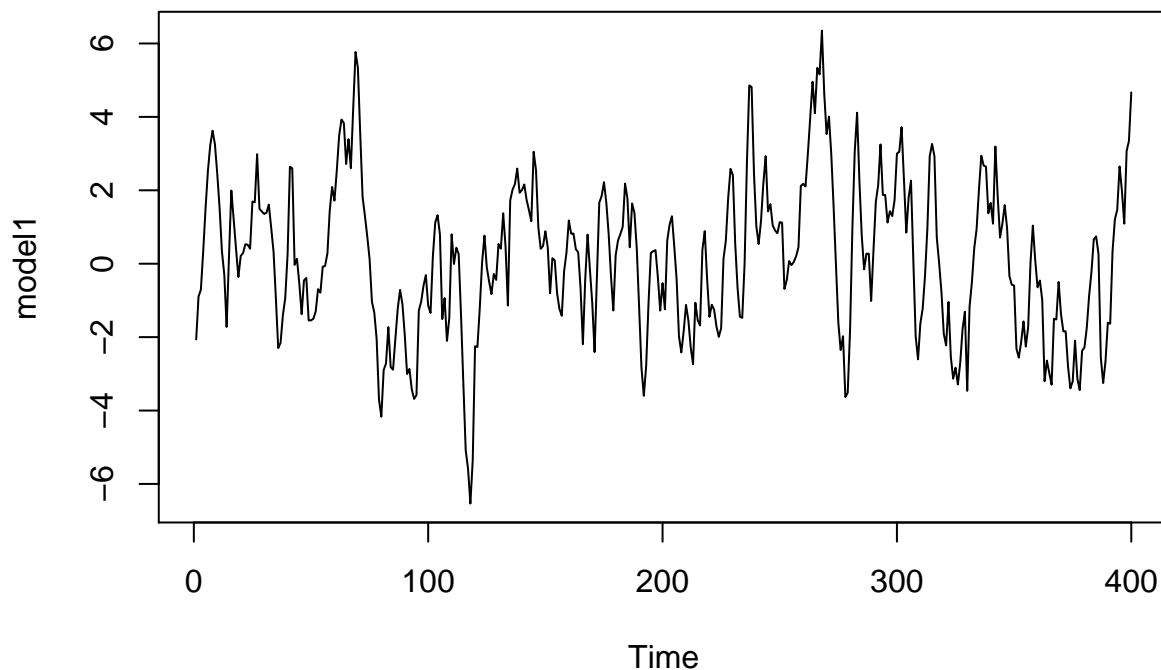
## The following objects are masked from 'package:stats':
##
##   acf, arima

## The following object is masked from 'package:utils':
##
##   tar

modell1 <- arima.sim(model=list(ar=c(1.2,-.32)),n=400)

plot.ts(modell1) # the series looks stationary

```



La serie parece estacionaria; probaremos esta hipótesis mediante una prueba de Dickey-Fuller y una KPSS:

```
adf.test(model1)
```

```
## Warning in adf.test(model1): p-value smaller than printed p-value
##
## Augmented Dickey-Fuller Test
##
## data: model1
## Dickey-Fuller = -4.6915, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(model1)
```

```
## Warning in kpss.test(model1): p-value greater than printed p-value
##
## KPSS Test for Level Stationarity
##
## data: model1
## KPSS Level = 0.1358, Truncation lag parameter = 5, p-value = 0.1
```

Efectivamente, la serie es estacionaria. Para determinar si la serie tiene media cero

```
media <- mean(model1)
t.test(model1)
```

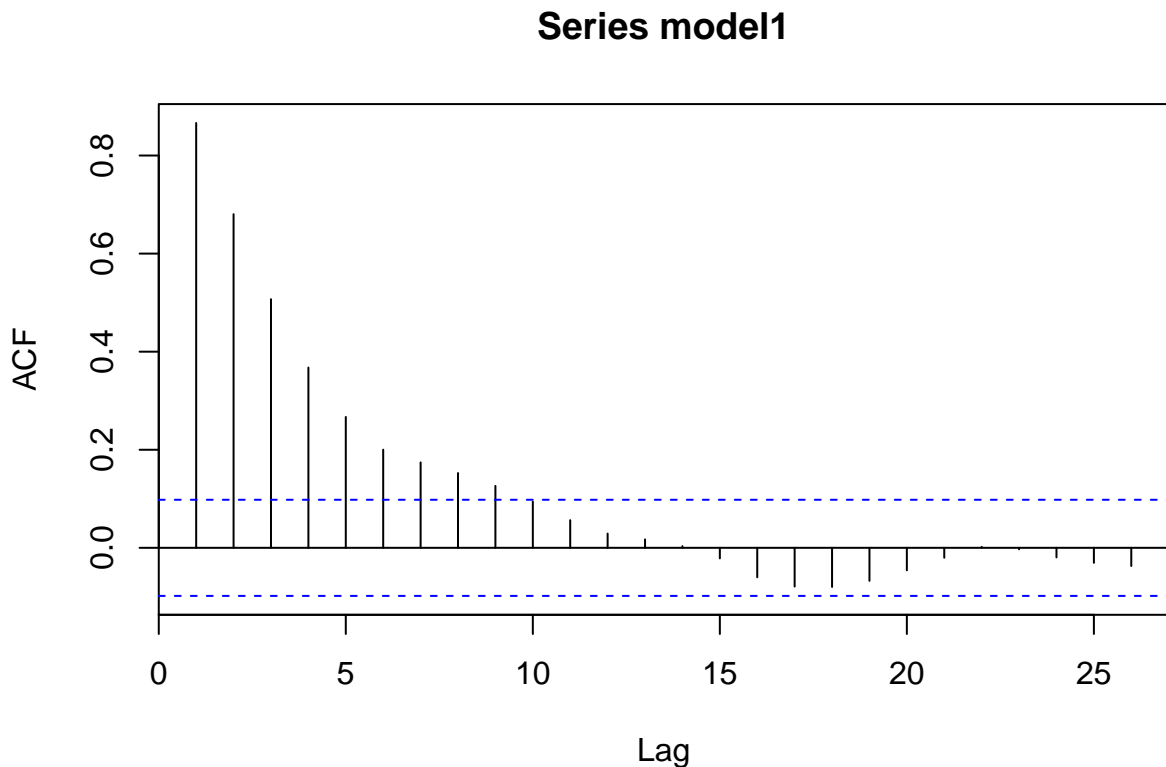
```
##
## One Sample t-test
```

```
##
## data:  model1
## t = 1.2553, df = 399, p-value = 0.2101
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.07402201  0.33555171
## sample estimates:
## mean of x
## 0.1307649
```

La serie tiene media cero, por lo que se eliminará la media en la estimación sucesiva.

Pasamos a la identificación del modelo. Primeramente, la función de autocorrelación:

```
acf(model1)
```

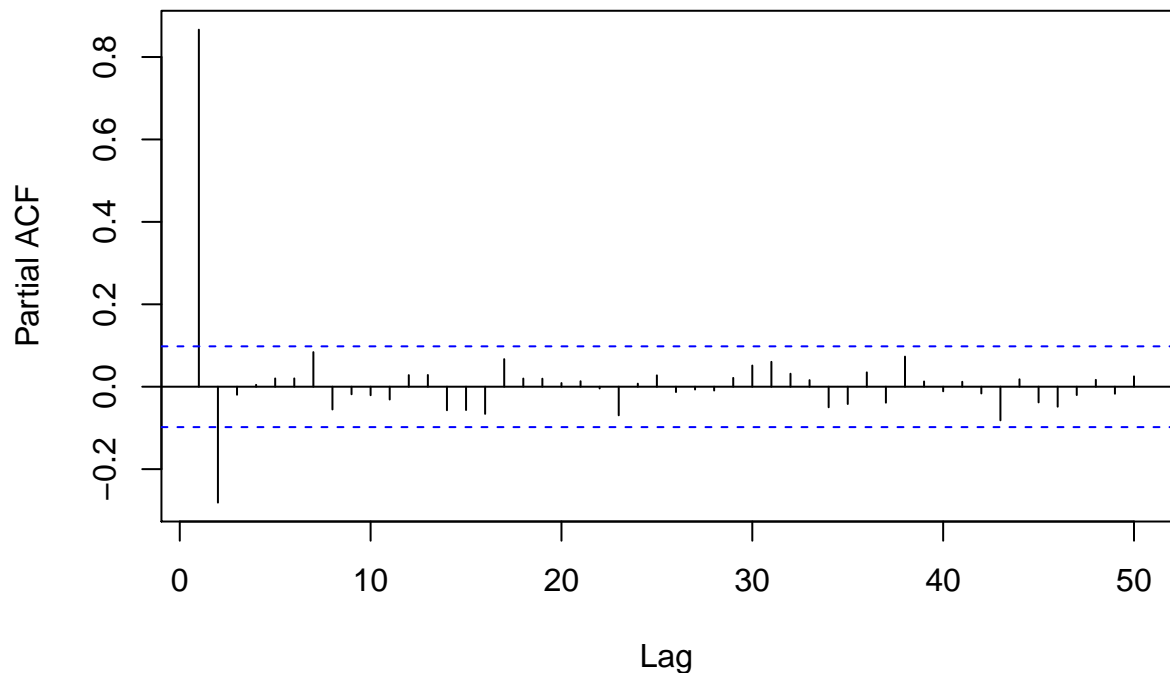


que muestra un patrón geoméricamente decreciente, consistente con una serie autoregresiva.

El orden se puede determinar usando la función de autocorrelación parcial, PACF:

```
pacf(model1, lag.max = 50)
```

Series model1



PACF sugiere $p = 2$.

Pasamos a la estimación del modelo $AR(2)$ sobre el 80% de los datos.

```
model1.train <- Arima(model1[1:320], order = c(2,0,0), include.mean = F)
model1.train
```

```
## Series: model1[1:320]
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##      ar1      ar2
##      1.1862 -0.3611
## s.e.  0.0522  0.0522
##
## sigma^2 estimated as 0.9105:  log likelihood=-438.9
## AIC=883.8   AICc=883.88   BIC=895.11
```

Para el diagnóstico se usarán la función ACF (para determinar si existe autocorrelación serial individual); pruebas de Ljung-Box (con diferentes rezagos, para probarlos de autocorrelación serial conjunta) y pruebas de normalidad (Pruebas de Shapiro-Wilk; Jarque-Bera).

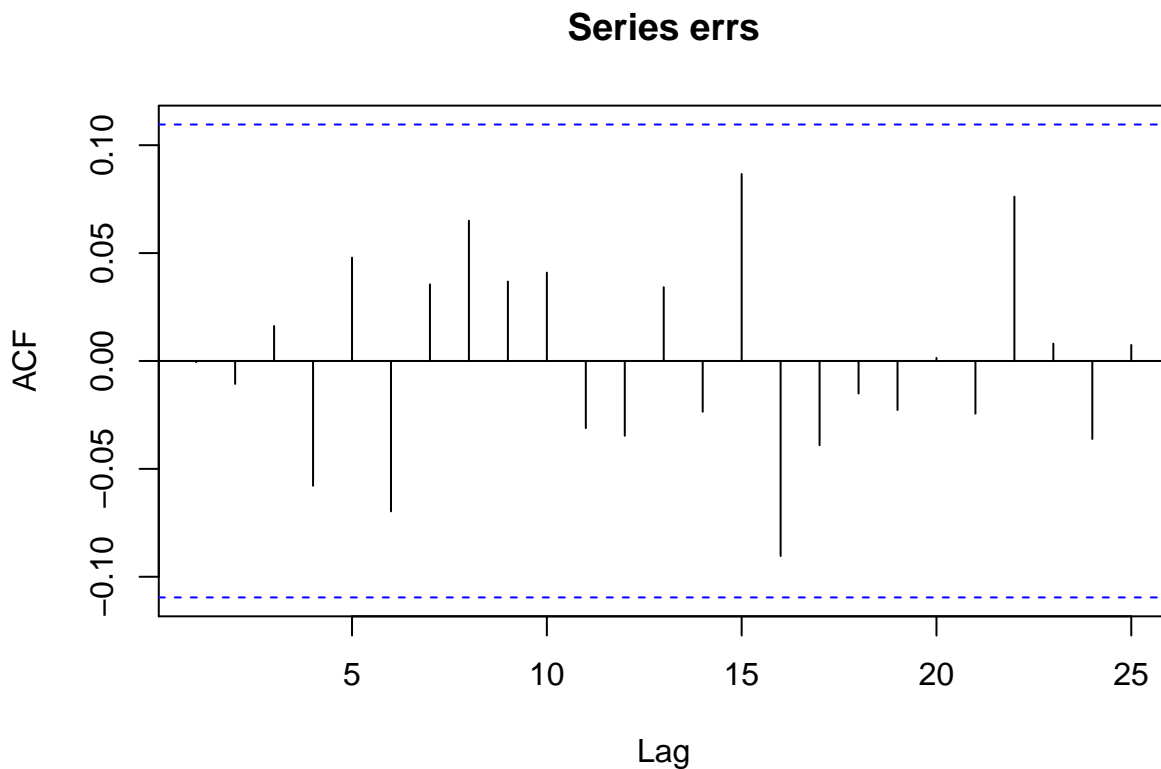
```
errs <- model1.train$residuals
Box.test(errs, lag = 20, type = "Ljung-Box", fitdf = 2)
```

```
##
## Box-Ljung test
```

```
##
## data:  errs
## X-squared = 13.754, df = 18, p-value = 0.745
Box.test(errs, lag = 25, type = "Ljung-Box", fitdf = 2)

##
## Box-Ljung test
##
## data:  errs
## X-squared = 16.461, df = 23, p-value = 0.8348
Box.test(errs, lag = 30, type = "Ljung-Box", fitdf = 2)

##
## Box-Ljung test
##
## data:  errs
## X-squared = 17.268, df = 28, p-value = 0.9431
acf(errs) # individual significance
```



```
shapiro.test(errs)

##
## Shapiro-Wilk normality test
##
```



```
## data:  errs
## W = 0.99232, p-value = 0.09776
```

```
jarque.bera.test(errs)
```

```
##
##  Jarque Bera Test
##
## data:  errs
## X-squared = 4.1334, df = 2, p-value = 0.1266
```

No existe evidencia de autocorrelación serial; además, los errores también parecen tener distribución normal.

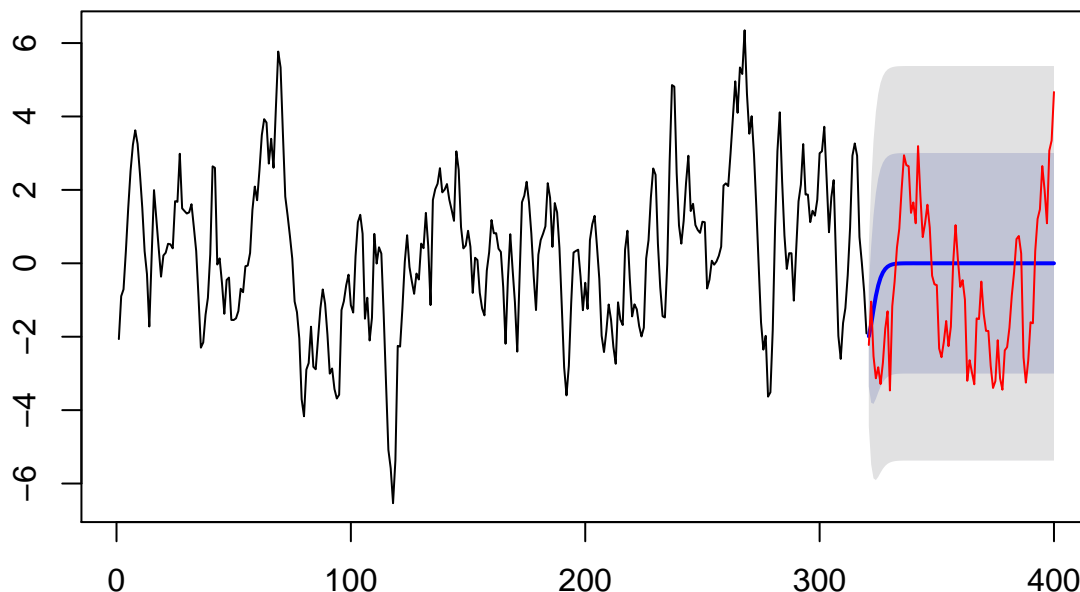
Después, se usará este modelo estimado para crear el pronóstico con intervalos de confianza 85%, 99% (para propósitos ilustrativos; por defecto, los intervalos de confianza son de 80%, 95%):

```
in.sample <- forecast(model1.train, h =80, level = c(85, 99))
```

Después, se creará la serie fuera de muestra para determinar la bondad del pronóstico:

```
model1.test = window(ts(model1), start = 321)
plot(in.sample)
lines(model1.test, col="red")
```

Forecasts from ARIMA(2,0,0) with zero mean



La totalidad de las observaciones fuera de muestra se encuentran en el intervalo de confianza al 99%. El pronóstico parece ser adecuado.

Otras medidas de desempeño son *MAE* (Mean Absolute Error) y *MAPE* (Mean Absolute

Percentage Error) para medir de manera más cuantitativa la discrepancia entre los valores pronosticados con los valores fuera de muestra:

$$MAE = \frac{1}{T} \sum_{i=1}^T | \text{pronóstico} - \text{valor fuera de muestra} |$$

$$MAPE = \frac{1}{T} \sum_{i=1}^T \left| \frac{\text{pronóstico} - \text{valor fuera de muestra}}{\text{valor fuera de muestra}} \right|$$

El objeto `in.sample` contiene los valores promedios del pronóstico; extraemos la media y la comparamos con los valores fuera de muestra:

```
(1/400)*sum(abs((in.sample$mean - model1.test))) # MAE
```

```
## [1] 0.3546297
```

```
(1/400)*sum(abs((in.sample$mean - model1.test)/model1.test)) # MAPE
```

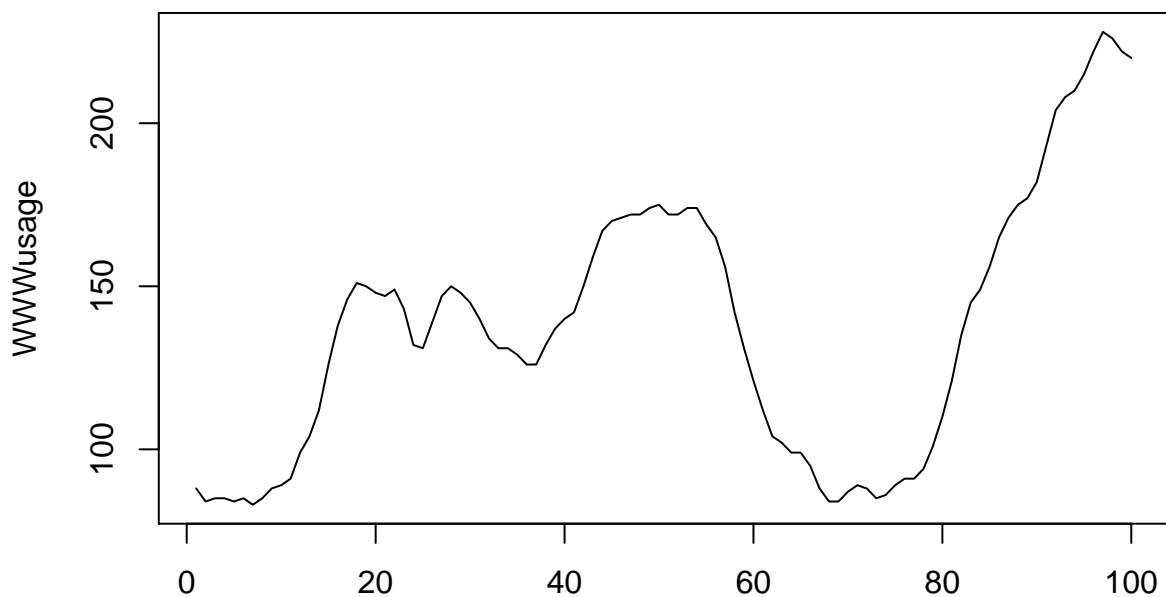
```
## [1] 0.19323
```

Ejemplo 2

Para la autonomía del documento, se usará la serie `WWWusage`, que cuenta el número de usuarios conectado a Internet por minuto; véase la ayuda `?WWWusage`.

La gráfica indica una posible no estacionariedad de la serie de tiempo asociada,

```
plot.ts(WWWusage)
```



que aplicamos las pruebas de estacionariedad por lo

```
adf.test(WWWusage)
```

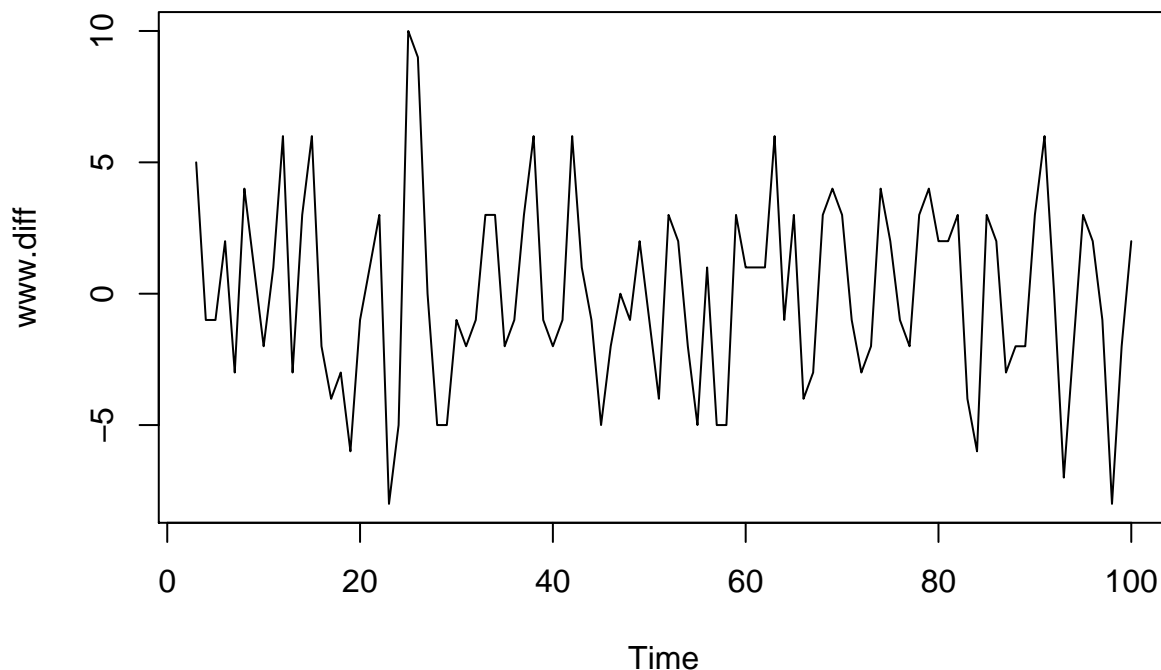
```
##  
## Augmented Dickey-Fuller Test  
##  
## data: WWWusage  
## Dickey-Fuller = -2.6421, Lag order = 4, p-value = 0.3107  
## alternative hypothesis: stationary
```

```
kpss.test(WWWusage)
```

```
##  
## KPSS Test for Level Stationarity  
##  
## data: WWWusage  
## KPSS Level = 0.45424, Truncation lag parameter = 4, p-value =  
## 0.05377
```

Existe cierta evidencia de no estacionariedad, por lo que pasamos a considerar la serie diferenciada dos veces (diferenciando una vez, no se encuentra evidencia robusta de estacionariedad)

```
www.diff <- as.ts(diff(diff(WWWusage)))  
plot.ts(www.diff)
```



```
adf.test(www.diff)
```

```
## Warning in adf.test(www.diff): p-value smaller than printed p-value  
##
```

```
## Augmented Dickey-Fuller Test
##
## data:  www.diff
## Dickey-Fuller = -4.828, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(www.diff)
```

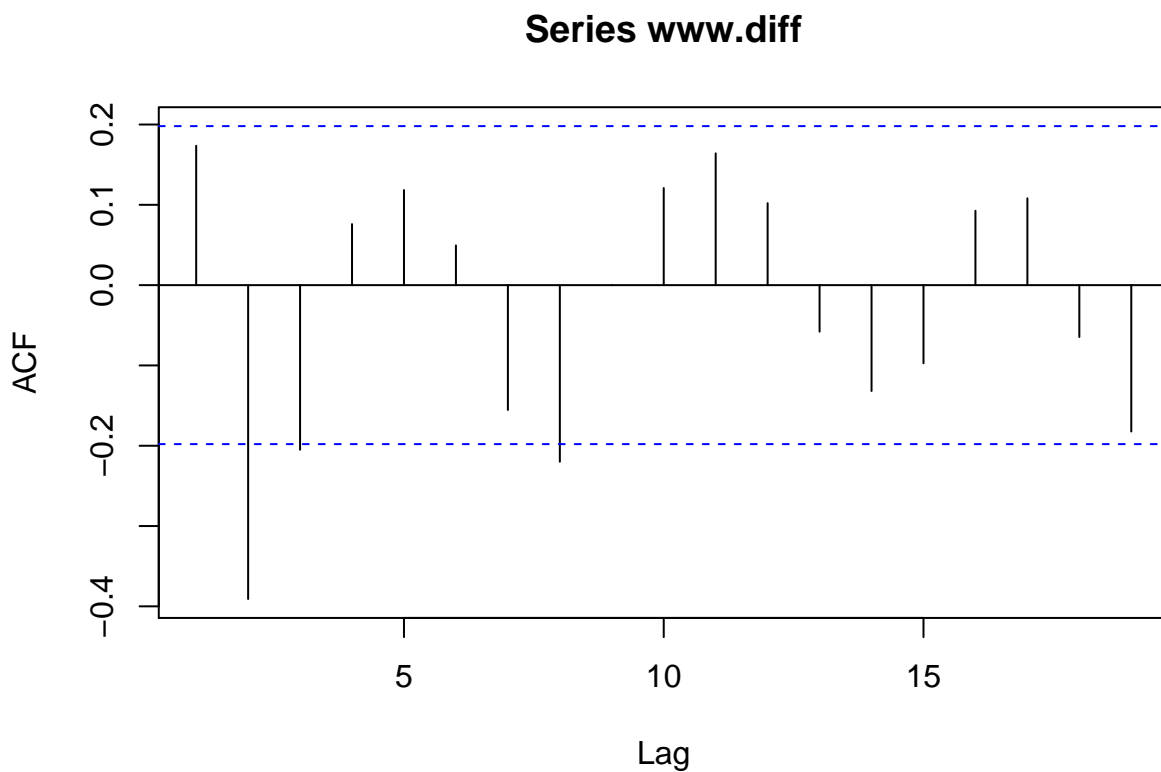
```
## Warning in kpss.test(www.diff): p-value greater than printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
```

```
## data:  www.diff
## KPSS Level = 0.051983, Truncation lag parameter = 3, p-value = 0.1
```

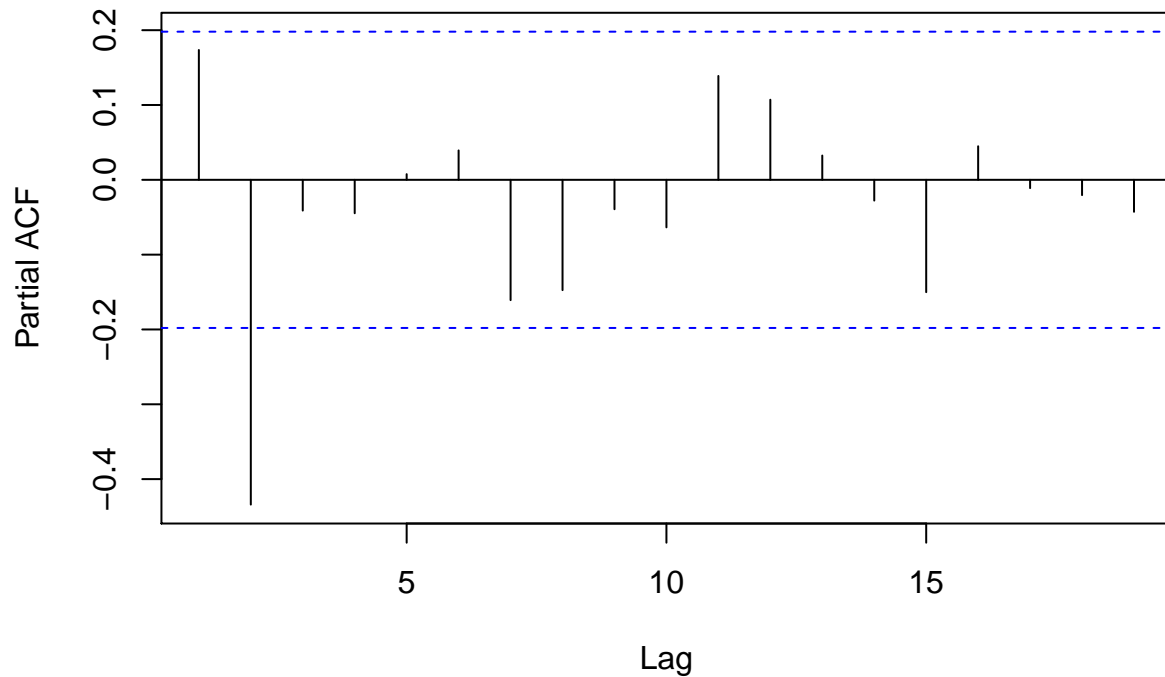
Los correlogramas, ACF, PACF, muestran que un posible modelo $AR(2)$ es apropiado:

```
acf(www.diff)
```



```
pacf(www.diff)
```

Series `www.diff`



Para

determinar si incluir, o no, una media, se prueba la hipótesis de media nula:

```
t.test(www.diff)
```

```
##
## One Sample t-test
##
## data:  www.diff
## t = 0.055747, df = 97, p-value = 0.9557
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.7061641  0.7469804
## sample estimates:
## mean of x
## 0.02040816
```

No hay evidencia que la media sea diferente de cero.

Estimamos el modelo $AR(2)$:

```
www.modelo1.fit <- Arima(www.diff, include.mean = F, order = c(2,0,0))
www.modelo1.fit
```

```
## Series: www.diff
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
```

```
##          ar1      ar2
##      0.2579 -0.4407
## s.e.  0.0915  0.0906
##
## sigma^2 estimated as 10.34:  log likelihood=-252.73
## AIC=511.46   AICc=511.72   BIC=519.22
```

Corremos también la función `auto.arima()` que identifica wl mismo modelo:

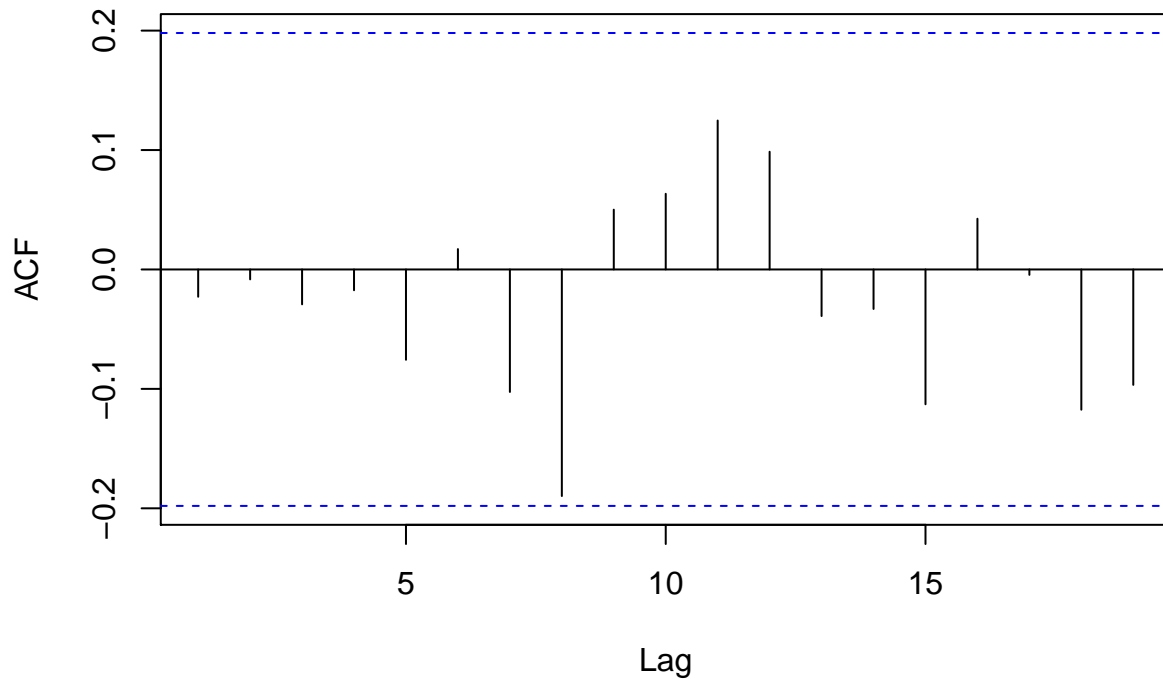
```
www.modelo2.fit <- auto.arima(www.diff, max.p = 6,
                             max.q = 6, allowmean = F,
                             seasonal = F, stationary = T)
www.modelo2.fit
```

```
## Series: www.diff
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##          ar1      ar2
##      0.2579 -0.4407
## s.e.  0.0915  0.0906
##
## sigma^2 estimated as 10.34:  log likelihood=-252.73
## AIC=511.46   AICc=511.72   BIC=519.22
```

Se realiza el diagnóstico de autocorrelación serial, individual

```
acf(www.modelo1.fit$residuals)
```

Series `www.modelo1.fit$residuals`



conjunto, con diferentes rezagos:

```
Box.test(www.modelo1.fit$residuals, fitdf = 2, lag=12)
```

```
##
## Box-Pierce test
##
## data:  www.modelo1.fit$residuals
## X-squared = 8.4363, df = 10, p-value = 0.5863
```

```
Box.test(www.modelo1.fit$residuals, fitdf = 2, lag=20)
```

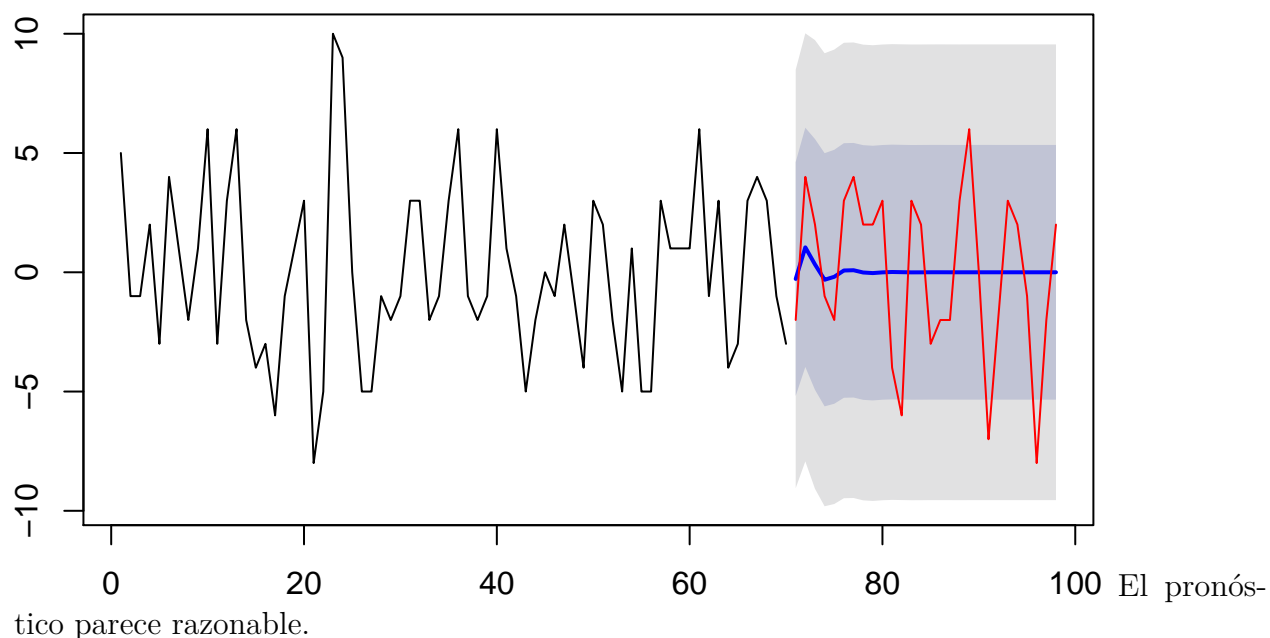
```
##
## Box-Pierce test
##
## data:  www.modelo1.fit$residuals
## X-squared = 14.221, df = 18, p-value = 0.7146
```

No hay evidencia de autocorrelación serial.

El pronóstico se

```
model1.train <- Arima(www.diff[1:70], order = c(2,0,0), include.mean = F)
in.sample <- forecast(model1.train, h = length(www.diff)-70, level = c(85, 99))
www.test = window(ts(www.diff), start = 71)
plot(in.sample)
lines(www.test, col="red")
```

Forecasts from ARIMA(2,0,0) with zero mean



EL cálculo de los parámetros MAE , $MAPE$, se da a continuación:

```
(1/length(www.diff))*sum(abs((in.sample$mean - www.test))) # MAE
```

```
## [1] 0.8240108
```

```
(1/length(www.diff))*sum(abs((in.sample$mean - www.test)/www.test)) # MAPE
```

```
## [1] Inf
```

El cálculo del parámetro $MAPE$ se ve afectado por una división entre cero.

Ejemplo 3

En este tercer ejemplo, se consideran los *retornos mensuales* de Intel Corporation, (ticker INTC), bajados de Yahoo! Finance, desde el 1ero de marzo 1980 hasta el 1ero de febrero 2019. Más adelante analizaremos la volatilidad de estos retornos.

Se prueba que los retornos son estacionario mediante las pruebas de Dickey-Fuller, KPSS:

```
# INTC <- read.csv("/Users/laan/pCloud Sync/2019/EcoFin2/Data/INTC.csv") OSX
INTC <- read.csv("/home/lino/pCloudDrive/pCloud Sync/2019/EcoFin2/Data/INTC.csv")
intc <- ts(INTC$Close, frequency = 12, start = c(1980, 03)) # frecuencia mensual
intc.rtn <- diff(log(intc))
adf.test(intc.rtn)
```

```
## Warning in adf.test(intc.rtn): p-value smaller than printed p-value
```

```
##
```



```
## Augmented Dickey-Fuller Test
##
## data:  intc.rtn
## Dickey-Fuller = -8.4809, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

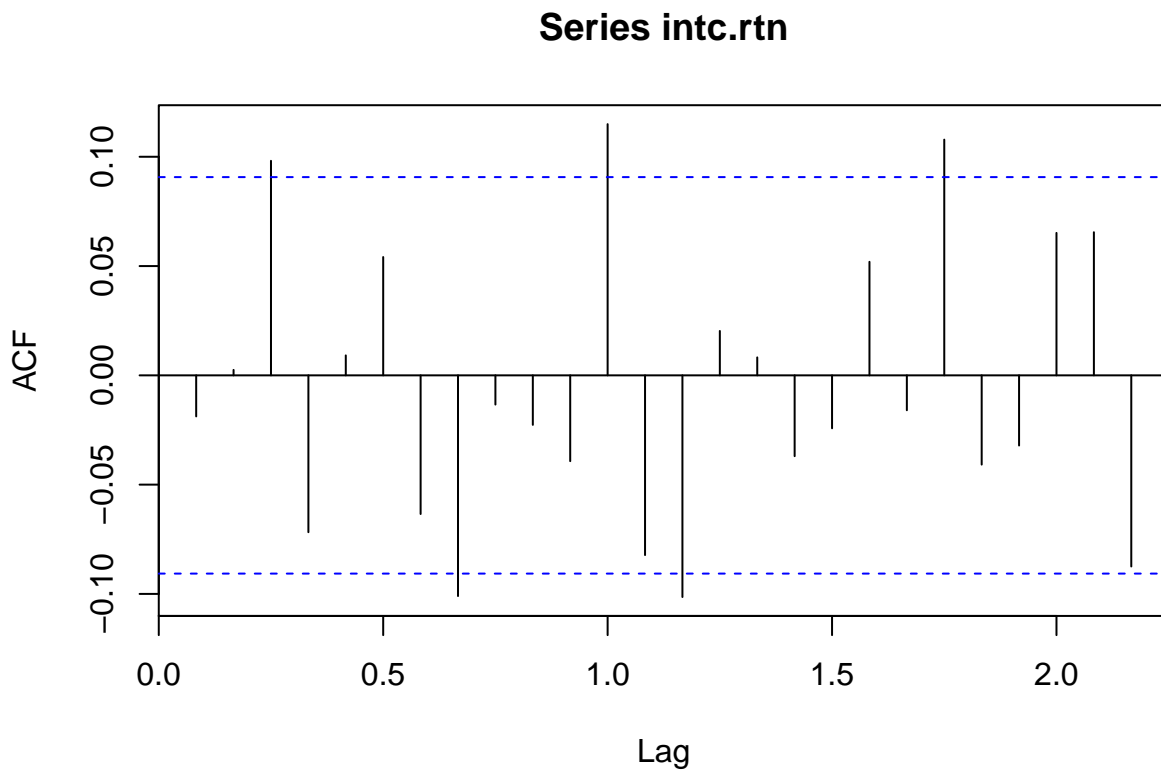
```
kpss.test(intc.rtn)
```

```
## Warning in kpss.test(intc.rtn): p-value greater than printed p-value
```

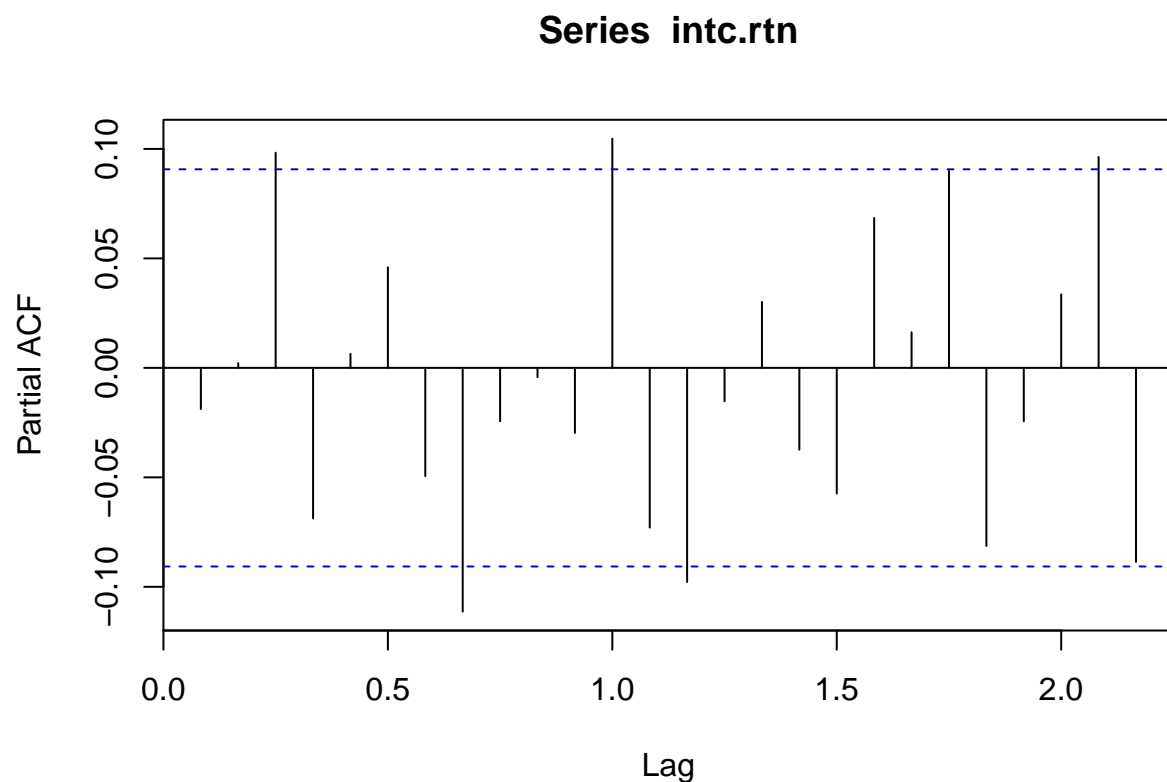
```
##
## KPSS Test for Level Stationarity
##
## data:  intc.rtn
## KPSS Level = 0.15343, Truncation lag parameter = 5, p-value = 0.1
```

Los correlogramas sugieren un modelo mixto $ARMA(p, q)$:

```
acf(intc.rtn)
```



```
pacf(intc.rtn)
```



Para determinar si incluir la media en la estimación del modelo, probaremos la hipótesis de media nula:

```
t.test(intc.rtn)
```

```
##
##  One Sample t-test
##
## data:  intc.rtn
## t = 2.1792, df = 466, p-value = 0.02981
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.001064003 0.020588157
## sample estimates:
##  mean of x
## 0.01082608
```

Hay evidencia que la media es diferente de cero, con valor estimado $\hat{\mu} = 0.0108261$.

Para la estimación usamos la función `auto.arima()`:

```
intc.fit <- auto.arima(intc.rtn, max.p = 10, max.q = 10, stationary = T, seasonal = F)
intc.fit

## Series: intc.rtn
## ARIMA(0,0,0) with non-zero mean
##
```

```
## Coefficients:
##          mean
##          0.0108
## s.e.    0.0050
##
## sigma^2 estimated as 0.01153:  log likelihood=380.02
## AIC=-756.04   AICc=-756.01   BIC=-747.75
```

que resulta en un modelo

$$intc.rtn = \mu + u_t.$$

(El log de los precios mensuales son una caminata aleatoria.)

El diagnóstico,

```
Box.test(intc.fit$residuals, lag=12)
```

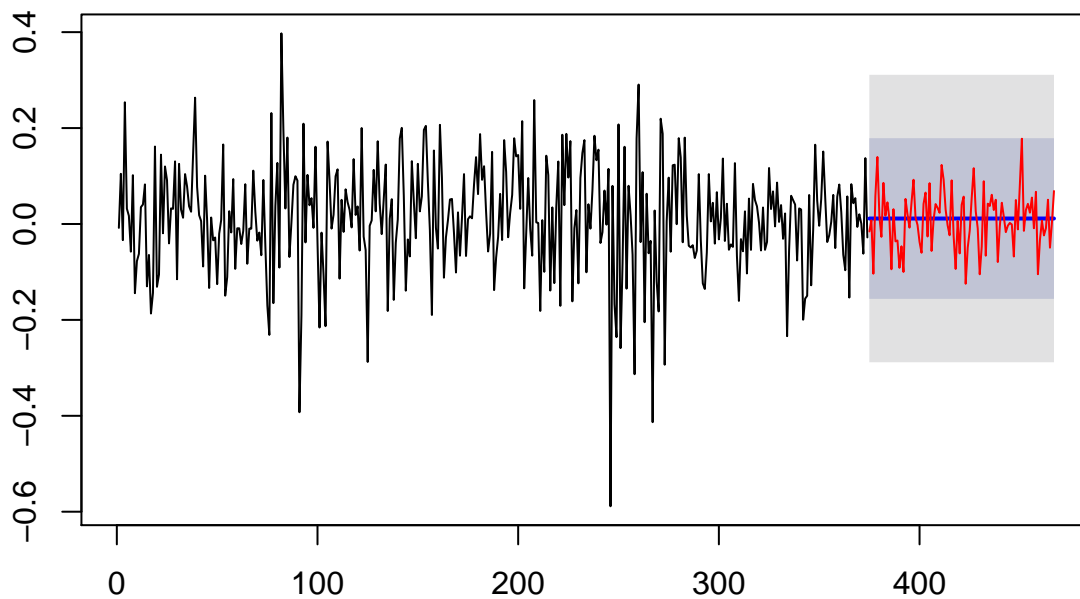
```
##
## Box-Pierce test
##
## data:  intc.fit$residuals
## X-squared = 22.334, df = 12, p-value = 0.03394
```

todavía muestra cierta autocorrelación serial residual que no consideraremos.

Para el pronóstico, usamos el 80% de las observaciones para el pronóstico dentro de muestra y el remanente para el diagnóstico fuera de muestra.

```
intc.train <- Arima(intc.rtn[1:374], order = c(0,0,0), include.mean = T)
in.sample <- forecast(intc.train, h = 467-374, level = c(85, 99))
intc.test = window(ts(intc.rtn), start = 375)
plot(in.sample)
lines(intc.test, col="red")
```

Forecasts from ARIMA(0,0,0) with non-zero mean



El pronóstico parece bueno.

El cálculo de *MAE*, *MAPE* da

```
(1/length(intc.rtn))*sum(abs((in.sample$mean - intc.test))) # MAE
```

```
## [1] 0.009274549
```

```
(1/length(intc.rtn))*sum(abs((in.sample$mean - intc.test)/intc.test)) # MAPE
```

```
## [1] 0.2619759
```

Heterocedasticidad condicional y volatilidad

Seguiremos, por lo general, el Cap. 4 del libro de R. Tsay [T].

La volatilidad se expresa, estadísticamente, en término de desviación estándar del activo bajo consideración.

Características de la volatilidad

Hay ciertas características de la volatilidad que se observan en los datos financieros.

- La volatilidad se agrupa: gran volatilidad en ciertos periodos de tiempo; baja volatilidad en otros.
- La volatilidad evolue de manera continua, por la mayor parte.

- Se observa un **efecto de apalancamiento**. La volatilidad reacciona de manera diferente a fuertes subida y a fuertes caídas de precios: el mayor impacto sobre la volatilidad se percibe durante las caídas de los precios.
 - Para modelar este efecto de apalancamiento, se desarrollaron los modelos EGARCH (Exponential Generalized AutoRegressive Heterocedastic), TGARCH (Threshold Generalized AutoRegressive Heterocedastic).

Medida de volatilidad

La volatilidad se puede medir de diferentes maneras:

1. La volatilidad de un activo se estima usando los precios del mismo, de sus derivados o de ambos.
2. La volatilidad se observa mediante
 - (a) retornos diarios del activo;
 - (b) datos de alta frecuencia (high-frequency data);
3. precios de opciones relacionadas al activo.

Estas diferentes manera de medición dan lugar a diferentes definiciones de volatilidad.

La definición que se usará es la *desviación estándar condicional de los retornos diarios*.

Otra medida de volatilidad es la *volatilidad implícita*, definida mediante la fórmula de Black-Scholes para opciones. El índice VIX del CBOE (Chicago Board Options Exchange) mide la volatilidad implícita. En México, la BMV, a través de MexDer, provee el índice de volatilidad VIMEX.

Para observar un efecto de volatilidad (efecto ARCH, autoregresivo, condicional heterocedástico) sobre un activo, se muestra que para los retornos del activo

1. la autocorrelación de los retornos es débil, o no existente;
2. los retornos no son independientes;
3. la media de la serie es diferente de cero.

La dependencia funcional típicamente se expresa mediante una función cuadrática (véase la modelación ARCH más abajo).

Para propósitos ilustrativos, seguiremos usando los retornos mensuales de Intel Corporation. Ya determinamos que la media de los retornos es diferente de cero.

Modelación ARCH

Un modelo $ARCH(M)$ consiste de una ecuación de media y de varianza como se da a continuación.

La *ecuación de la media* para los retornos, r_t , está dada por

$$\mu_t = E[r_t | \Omega_{t-1}],$$

típicamente un modelo $ARMA(p, q)$ con media.

La *ecuación de la varianza* mide el grado de dependencia estadística de los retornos:

$$\begin{aligned} u_t &= \sigma_t \varepsilon_t \\ \sigma_t^2 &= Var(r_t | \Omega_{t-1}) = E[(r_t - \mu_t)^2 | \Omega_{t-1}] \\ &= \alpha_0 + \alpha_1 u_{t-1}^2 + \cdots + \alpha_m u_{t-m}^2, \end{aligned}$$

con $\alpha_0 > 0$, $\alpha_i \geq 0$, $i = 1, \dots, m$.

En la ecuación de la varianza, el término ε_t es una serie de variables aleatorias independientes, idénticamente distribuidas, con media cero y varianza uno. Distribuciones usadas para ε_t son la normal estándar; la t -Student estandarizada; la GED (Generalized Error Distribution). En datos financieros las colas son, por lo general, más gruesas que las colas de una normal estándar.

Procedimiento

1. Estimar la ecuación de la media para los retornos; si un modelo $ARMA(0, 0)$, con media, todavía tiene dependencia serial residua, entonces usar un modelo $ARMA(p, q)$ apropiado. Empíricamente, la ecuación de la media, $\mu_t = E[r_t | \Omega_{t-1}]$ es sencilla: un modelo $ARMA(1, 1)$ (con media) en muchos casos es suficiente.
2. Usar los residuos de la ecuación de la media para detectar efectos ARCH: se puede usar la prueba de Ljung-Box sobre los residuales al cuadrado, por ejemplo; también la prueba ML (multiplicadores de Lagrange) de Engle. En la prueba de Engle, el estadístico de prueba es el estadístico F de significancia del modelo, con valor-p asociado.
3. Si se observan efectos ARCH, se puede usar la PACF de u_t^2 para determinar el orden m del modelo ARCH.
4. Una vez detectados efectos ARCH, se estima la ecuación de la media y de varianza simultáneamente. La estimación de un modelo ARCH se realiza maximizando la verosimilitud, mediante la distribución de ε_t .

Ejemplo

Usaremos los retornos mensuales r_t de Intel Corporation; ya vimos en el Ejemplo 3, Sección Pronóstico, que estos retornos satisfacen la ecuación

$$r_t = \mu + u_t.$$

Para determinar la existencia del efecto heterocedástico autoregresivo usamos la prueba de Ljung-Box sobre los residuales cuadrados:

```
Box.test((intc.rtn - mean(intc.rtn))^2, lag = 12)
```

```
##
## Box-Pierce test
##
## data: (intc.rtn - mean(intc.rtn))^2
## X-squared = 46.298, df = 12, p-value = 6.163e-06
```

Usaremos también el script `archTest.R` de R.~Tsay para usar la prueba ML de Engle:

```
# source("/Users/laan/pCloud Sync/2019/EcoFin2/Data/archTest.R") OSX
source("/home/lino/pCloudDrive/pCloud Sync/2019/EcoFin2/Data/archTest.R")
archTest(intc.rtn-mean(intc.rtn), 12)
```

```
##
## Call:
## lm(formula = atsq ~ x)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
##	-0.05375	-0.00845	-0.00562	0.00151	0.34886

```
##
## Coefficients:
```

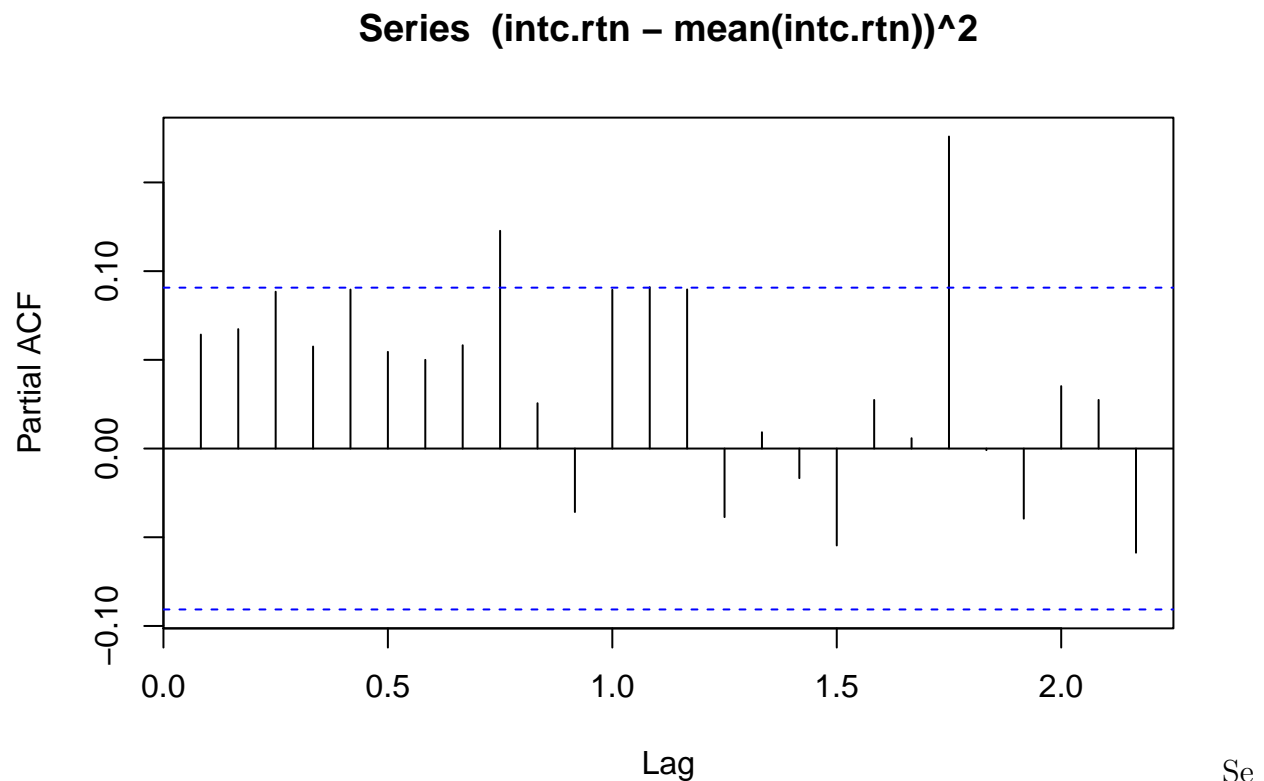
	Estimate	Std. Error	t value	Pr(> t)	
## (Intercept)	0.005234	0.001780	2.941	0.00345	**
## x1	0.029262	0.047360	0.618	0.53699	
## x2	0.036794	0.047347	0.777	0.43751	
## x3	0.055324	0.047367	1.168	0.24345	
## x4	0.029016	0.047098	0.616	0.53816	
## x5	0.067706	0.047056	1.439	0.15090	
## x6	0.039479	0.047135	0.838	0.40271	
## x7	0.036666	0.047131	0.778	0.43702	
## x8	0.055165	0.047039	1.173	0.24153	
## x9	0.117460	0.046889	2.505	0.01260	*
## x10	0.022911	0.047156	0.486	0.62730	
## x11	-0.038180	0.047145	-0.810	0.41847	
## x12	0.089871	0.047154	1.906	0.05731	.

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02484 on 442 degrees of freedom
## Multiple R-squared:  0.06137,    Adjusted R-squared:  0.03589
## F-statistic: 2.408 on 12 and 442 DF,  p-value: 0.004986
```

El valor-p de la prueba de Engle es = .004986.

Existe evidencia de efecto ARCH. Para estimar el orden del modelo $ARCH(m)$ usamos la PACF con los residuales

```
pacf((intc.rtn - mean(intc.rtn))^2)
```



considera que $m = 5$ puede ser adecuado.

Para la estimación del modelo $ARCH(5)$ usamos la función `garchFit()` de la librería `fGarch`.

Nótese que el término α_0 en la ecuación de la varianza se denota con ω en la estimación con R.

Para determinar la distribución condicional, probaremos la hipótesis de normalidad de los errores:

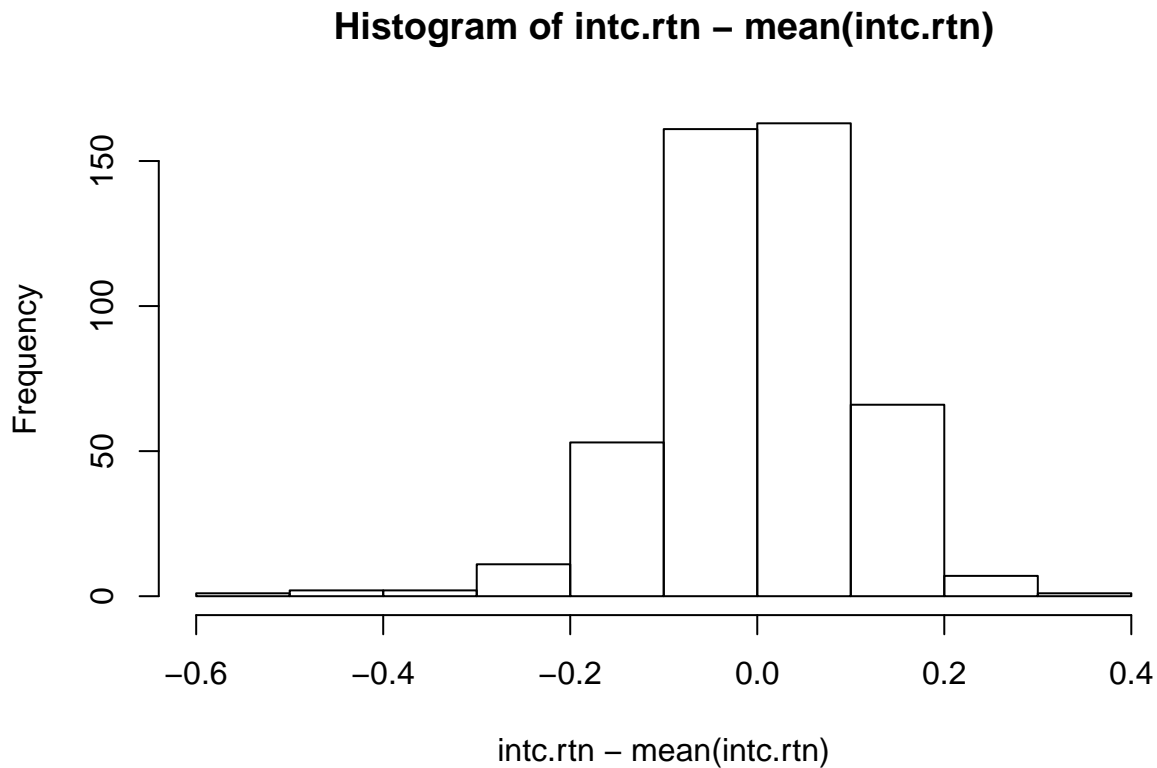
```
shapiro.test((intc.rtn - mean(intc.rtn)))
```

```
##
## Shapiro-Wilk normality test
```



```
##
## data: (intc.rtn - mean(intc.rtn))
## W = 0.97025, p-value = 3.884e-08
```

```
hist(intc.rtn - mean(intc.rtn))
```



Los errores no son normales; usando la información del histograma, se usará una distribución condicional *t*-Student asimétrica.

```
library(fGarch)
```

```
## Loading required package: timeDate
```

```
##
```

```
## Attaching package: 'timeDate'
```

```
## The following objects are masked from 'package:TSA':
```

```
##
```

```
## kurtosis, skewness
```

```
## Loading required package: timeSeries
```

```
## Loading required package: fBasics
```

```
intc.archfit1 <- garchFit(~ 1 + garch(5,0),
                        data = intc.rtn, trace = F, cond.dist = "sstd")
intc.archfit1
```

```
##
```

```

## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(5, 0), data = intc.rtn, cond.dist = "sstd",
##    trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(5, 0)
## <environment: 0x5631197ba588>
##  [data = intc.rtn]
##
## Conditional Distribution:
##  sstd
##
## Coefficient(s):
##      mu      omega    alpha1    alpha2    alpha3    alpha4
## 0.0106938 0.0057569 0.1706409 0.0500237 0.0949283 0.1054819
##   alpha5      skew      shape
## 0.1076036 0.9253258 6.3448764
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.010694   0.004534   2.358 0.018355 *
## omega   0.005757   0.001118   5.148 2.63e-07 ***
## alpha1  0.170641   0.089317   1.911 0.056069 .
## alpha2  0.050024   0.047918   1.044 0.296511
## alpha3  0.094928   0.058758   1.616 0.106183
## alpha4  0.105482   0.071994   1.465 0.142883
## alpha5  0.107604   0.067463   1.595 0.110713
## skew    0.925326   0.061410  15.068 < 2e-16 ***
## shape   6.344876   1.722128   3.684 0.000229 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  411.0137    normalized:  0.8801151
##
## Description:
##  Tue Jul 16 03:26:01 2019 by user: lino

```

Observamos que α_2 , α_3 , α_4 , α_5 no son significativos, por lo que consideramos que un modelo

ARCH(1) puede ser más apropiado:

```
intc.archfit2 <- garchFit(~ 1 + garch(1,0),
                        data = intc.rtn, trace = F, cond.dist = "sstd")
intc.archfit2

##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(1, 0), data = intc.rtn, cond.dist = "sstd",
##    trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(1, 0)
## <environment: 0x56311c385678>
##  [data = intc.rtn]
##
## Conditional Distribution:
##  sstd
##
## Coefficient(s):
##      mu      omega    alpha1      skew      shape
## 0.0117473 0.0089556 0.2416264 0.9320647 6.3052580
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.011747  0.004609   2.549 0.010807 *
## omega   0.008956  0.001084   8.260 2.22e-16 ***
## alpha1  0.241626  0.095949   2.518 0.011793 *
## skew    0.932065  0.058882  15.829 < 2e-16 ***
## shape   6.305258  1.752387   3.598 0.000321 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 402.126      normalized: 0.8610835
##
## Description:
##  Tue Jul 16 03:26:01 2019 by user: lino
```

El modelo estimado es

$$\hat{\mu} = .01174 + u_t$$
$$\hat{\sigma}_t^2 = .00895 + .24162u_{t-1}^2$$

y cada término es significativo. El retorno esperado estimado es $\hat{\mu} = .01174 = 1.174\%$.

Diagnóstico No es necesario usar pruebas de normalidad ya que se modeló con innovaciones no normales. El diagnóstico en este caso consiste en verificar que no hay heterocedasticidad residual.

```
sresi=intc.archfit2@residuals/intc.archfit2@sigma.t
Box.test(sresi,lag=12,type='Ljung')

##
##  Box-Ljung test
##
## data:  sresi
## X-squared = 17.511, df = 12, p-value = 0.1314

archTest(sresi,12)

##
## Call:
## lm(formula = atsq ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.0495 -0.7779 -0.5203  0.1655 30.2351
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.55507    0.17218   3.224  0.00136 **
## x1          -0.05470    0.04738  -1.154  0.24894
## x2           0.02580    0.04742   0.544  0.58672
## x3           0.04664    0.04739   0.984  0.32557
## x4           0.02334    0.04697   0.497  0.61943
## x5           0.03669    0.04691   0.782  0.43462
## x6           0.03919    0.04691   0.835  0.40398
## x7           0.03887    0.04691   0.829  0.40778
## x8           0.05947    0.04691   1.268  0.20556
## x9           0.13935    0.04669   2.984  0.00300 **
## x10          0.04252    0.04712   0.902  0.36736
## x11         -0.03508    0.04716  -0.744  0.45737
## x12          0.08513    0.04710   1.808  0.07134 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.23 on 442 degrees of freedom
## Multiple R-squared:  0.05027,    Adjusted R-squared:  0.02448
## F-statistic:  1.95 on 12 and 442 DF,  p-value: 0.02729
```

Hay poca evidencia de heterocedasticidad residua. Para eliminar el efecto ARCH residuo, se considera un modelo $ARCH(2)$.

```
intc.archfit3 <- garchFit(~ 1 + garch(2,0),
                        data = intc.rtn, trace = F, cond.dist = "sstd")
intc.archfit3
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(2, 0), data = intc.rtn, cond.dist = "sstd",
##          trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(2, 0)
## <environment: 0x56311d8267b0>
##  [data = intc.rtn]
##
## Conditional Distribution:
##  sstd
##
## Coefficient(s):
##          mu      omega    alpha1    alpha2      skew      shape
## 0.0119997 0.0083835 0.2079315 0.0790665 0.9313925 6.2019613
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.012000   0.004617   2.599 0.009346 **
## omega   0.008384   0.001095   7.659 1.89e-14 ***
## alpha1  0.207931   0.096430   2.156 0.031061 *
## alpha2  0.079066   0.061753   1.280 0.200415
## skew    0.931392   0.060047  15.511 < 2e-16 ***
## shape   6.201961   1.677563   3.697 0.000218 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```

## Log Likelihood:
## 403.7362      normalized:  0.8645314
##
## Description:
## Tue Jul 16 03:26:01 2019 by user: lino

sresi3=intc.archfit3@residuals/intc.archfit3@sigma.t
Box.test(sresi3,lag=12,type='Ljung')

##
## Box-Ljung test
##
## data:  sresi3
## X-squared = 16.952, df = 12, p-value = 0.1514

archTest(sresi3,12)

##
## Call:
## lm(formula = atsq ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.129 -0.796 -0.549  0.214 32.876
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.61613    0.18344   3.359 0.000851 ***
## x1          -0.04327    0.04732  -0.914 0.360977
## x2          -0.03330    0.04734  -0.703 0.482130
## x3           0.04255    0.04734   0.899 0.369270
## x4           0.02299    0.04693   0.490 0.624519
## x5           0.03528    0.04687   0.753 0.452019
## x6           0.03079    0.04688   0.657 0.511644
## x7           0.03061    0.04688   0.653 0.514124
## x8           0.06417    0.04687   1.369 0.171685
## x9           0.13487    0.04671   2.888 0.004072 **
## x10          0.03607    0.04711   0.766 0.444318
## x11          -0.02945    0.04712  -0.625 0.532257
## x12          0.09919    0.04707   2.107 0.035655 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.336 on 442 degrees of freedom
## Multiple R-squared:  0.04609,    Adjusted R-squared:  0.02019
## F-statistic:  1.78 on 12 and 442 DF,  p-value: 0.04909

```

El modelo $ARCH(2)$ elimina el efecto ARCH residuo. El retorno esperado estimado es un poco superior al anterior.

Pronóstico

El pronóstico de un modelo ARCH no es muy diferente del pronóstico de un modelo AR. Los detalles se encuentran en el libro de R. Tsay, Sección 4.5.3. Consideramos, para propósitos explicativos, el caso $ARCH(1)$; supón que, el momento $h = 4$, se quiere pronosticar $\sigma_{h=4}^2(1)$, el valor de la volatilidad 1 periodo en adelante. Entonces,

$$\sigma_{h=4}^2(1) = \alpha_0 + \alpha_1 u_{4+1-1}^2 = \alpha_0 + \alpha_1 u_4^2;$$

el pronóstico 2 periodos en adelante,

$$\sigma_{h=4}^2(2) = \alpha_0 + \alpha_1 \sigma_4^2(1)$$

y así, sucesivamente. Véase la fórmula (4.11) en el libro de Tsay.

Se presentarán diferentes situaciones del pronóstico del modelo estimado anteriormente $ARCH(1)$, 10 periodos (meses) en el futuro.

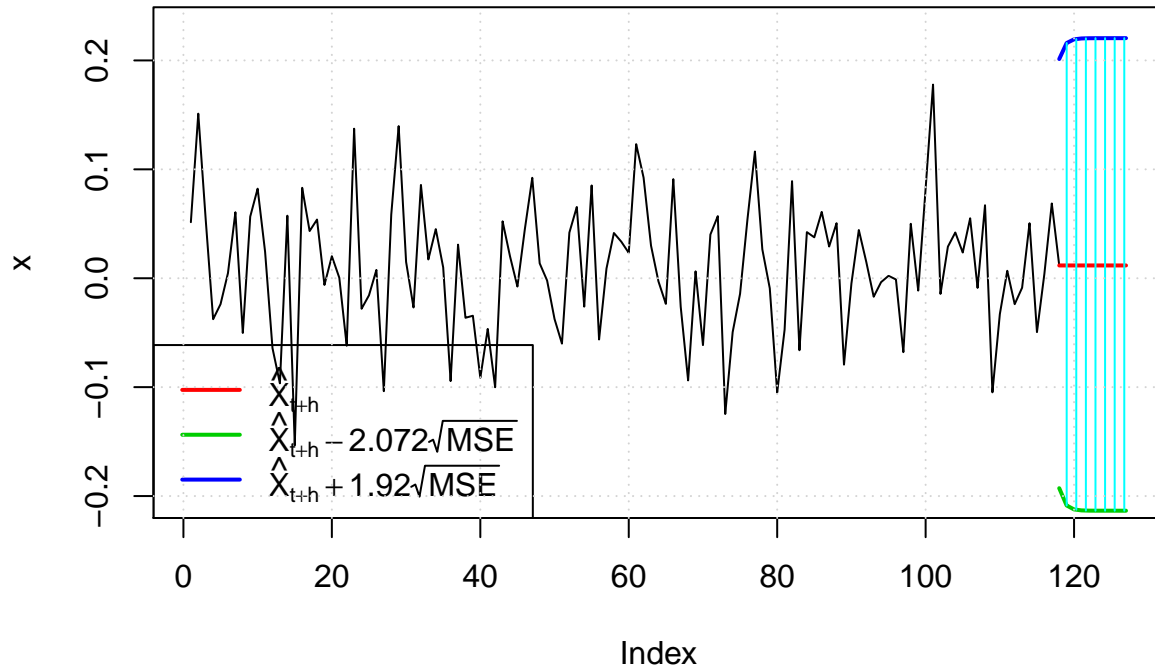
```
predict(intc.archfit2, n.ahead = 10)
```

##	meanForecast	meanError	standardDeviation
## 1	0.01174725	0.09868533	0.09868533
## 2	0.01174725	0.10634240	0.10634240
## 3	0.01174725	0.10811123	0.10811123
## 4	0.01174725	0.10853431	0.10853431
## 5	0.01174725	0.10863629	0.10863629
## 6	0.01174725	0.10866091	0.10866091
## 7	0.01174725	0.10866686	0.10866686
## 8	0.01174725	0.10866830	0.10866830
## 9	0.01174725	0.10866865	0.10866865
## 10	0.01174725	0.10866873	0.10866873

Visualización con ploteo:

```
predict(intc.archfit2, n.ahead = 10, plot = TRUE)
```

Prediction with confidence intervals

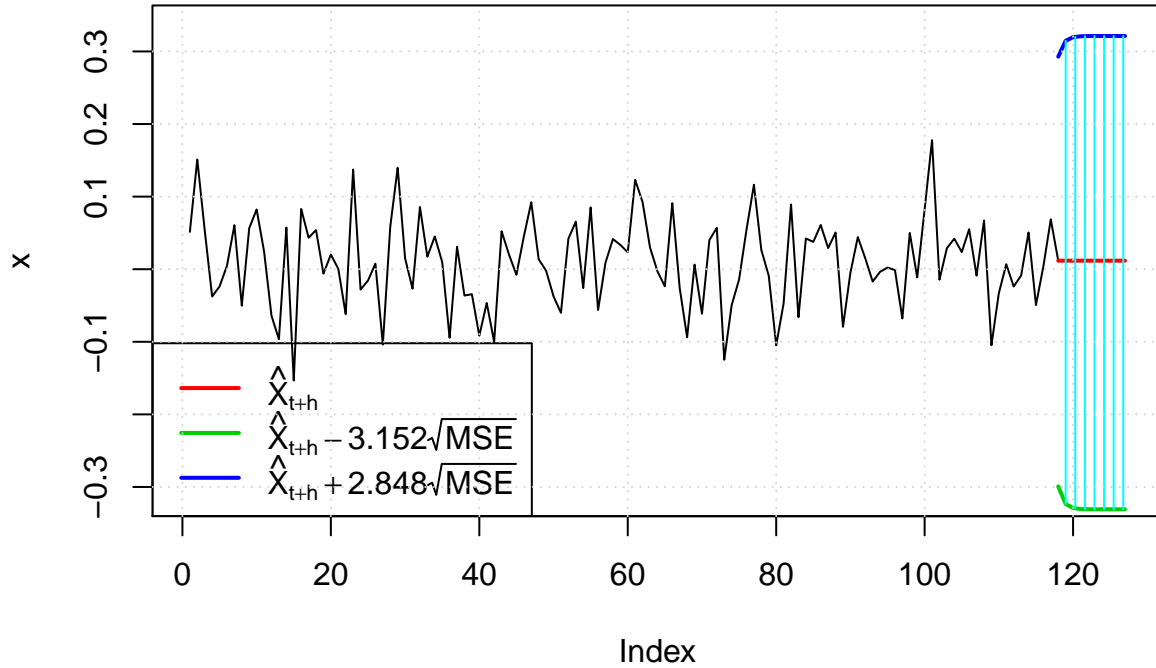


##	meanForecast	meanError	standardDeviation	lowerInterval	upperInterval
## 1	0.01174725	0.09868533	0.09868533	-0.1927625	0.2012618
## 2	0.01174725	0.10634240	0.10634240	-0.2086306	0.2159664
## 3	0.01174725	0.10811123	0.10811123	-0.2122962	0.2193633
## 4	0.01174725	0.10853431	0.10853431	-0.2131730	0.2201757
## 5	0.01174725	0.10863629	0.10863629	-0.2133843	0.2203716
## 6	0.01174725	0.10866091	0.10866091	-0.2134353	0.2204189
## 7	0.01174725	0.10866686	0.10866686	-0.2134477	0.2204303
## 8	0.01174725	0.10866830	0.10866830	-0.2134506	0.2204330
## 9	0.01174725	0.10866865	0.10866865	-0.2134513	0.2204337
## 10	0.01174725	0.10866873	0.10866873	-0.2134515	0.2204339

Visualización con ploteo e intervalo de confianza al 99%.

```
predict(intc.archfit2, n.ahead = 10, plot = TRUE, conf=.99)
```


Prediction with confidence intervals



##	meanForecast	meanError	standardDeviation	lowerInterval	upperInterval
## 1	0.01174725	0.09868533	0.09868533	-0.2992801	0.2928225
## 2	0.01174725	0.10634240	0.10634240	-0.3234130	0.3146314
## 3	0.01174725	0.10811123	0.10811123	-0.3289878	0.3196694
## 4	0.01174725	0.10853431	0.10853431	-0.3303212	0.3208744
## 5	0.01174725	0.10863629	0.10863629	-0.3306426	0.3211648
## 6	0.01174725	0.10866091	0.10866091	-0.3307202	0.3212350
## 7	0.01174725	0.10866686	0.10866686	-0.3307390	0.3212519
## 8	0.01174725	0.10866830	0.10866830	-0.3307435	0.3212560
## 9	0.01174725	0.10866865	0.10866865	-0.3307446	0.3212570
## 10	0.01174725	0.10866873	0.10866873	-0.3307449	0.3212572

Modelo ARCH–Innovaciones no gaussianas

La librería `fGarch` puede manejar innovaciones que siguen la distribución t -Student; t -Student asimétrica; GDE, y GDE asimétrica.

A continuación se presenta el modelo de volatilidad presentado arriba,

$$\begin{aligned}
 \text{intc.returns} &= \mu + u_t \\
 u_t &= \sigma_t \varepsilon_t \\
 \sigma_t^2 &= \alpha_0 + \alpha_1 u_{t-1}^2 + \cdots + \alpha_m u_{t-m}^2,
 \end{aligned}$$

usando algunas de estas innovaciones no gaussianas.

Cada una de estas opciones se pasa usando el parámetro `cond.dist=` con la abreviación correspondiente: `std`, `sstd`, `ged`, `sged`, respectivamente. Otra posibilidad es usar QMLE (Quasi-Maximum Likelihood Estimation), que supone distribución normal, con errores estándares robustos.

Se pueden también pronosticar valores futuros usando estas distribuciones.

Innovación t-Student

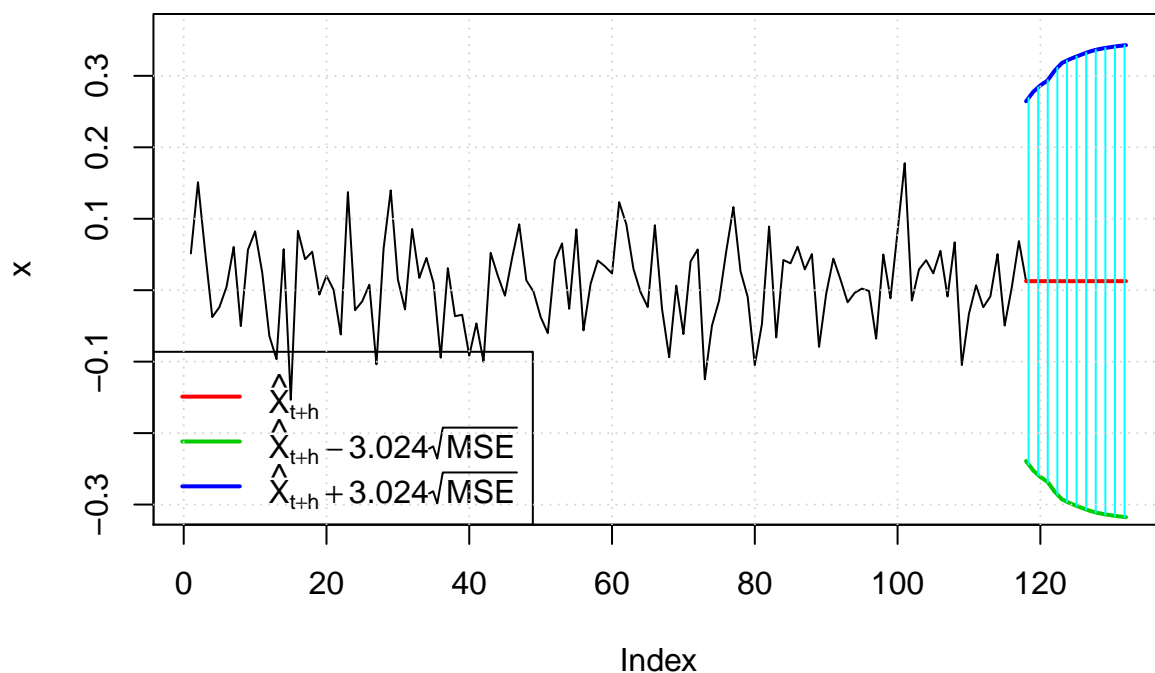
```
intc.archfit.t <- garchFit(~ 1 + garch(5,0), data = intc.rtn, trace = F, cond.dist = "std",
intc.archfit.t

##
## Title:
##   GARCH Modelling
##
## Call:
##   garchFit(formula = ~1 + garch(5, 0), data = intc.rtn, cond.dist = "std",
##     trace = F)
##
## Mean and Variance Equation:
##   data ~ 1 + garch(5, 0)
## <environment: 0x56311c09c520>
##   [data = intc.rtn]
##
## Conditional Distribution:
##   std
##
## Coefficient(s):
##           mu      omega    alpha1    alpha2    alpha3    alpha4
## 0.0126420  0.0058145  0.1684766  0.0523688  0.1031607  0.0961664
##   alpha5      shape
## 0.1059516  6.0413008
##
## Std. Errors:
##   based on Hessian
##
## Error Analysis:
##           Estimate Std. Error  t value Pr(>|t|)
## mu          0.012642    0.004237   2.984  0.00284 **
## omega        0.005814    0.001137   5.115 3.14e-07 ***
## alpha1       0.168477    0.090190   1.868  0.06176 .
## alpha2       0.052369    0.047835   1.095  0.27361
## alpha3       0.103161    0.060639   1.701  0.08890 .
```

```
## alpha4 0.096166 0.071017 1.354 0.17570
## alpha5 0.105952 0.067763 1.564 0.11792
## shape 6.041301 1.533817 3.939 8.19e-05 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 410.3437 normalized: 0.8786803
##
## Description:
## Tue Jul 16 03:26:02 2019 by user: lino
```

```
predict(intc.archfit.t, n.ahead = 15, plot = TRUE, cond.dist = "std", conf = .99)
```

Prediction with confidence intervals



	meanForecast	meanError	standardDeviation	lowerInterval	upperInterval
## 1	0.01264201	0.08325734	0.08325734	-0.2391096	0.2643936
## 2	0.01264201	0.08762152	0.08762152	-0.2523059	0.2775899
## 3	0.01264201	0.09061337	0.09061337	-0.2613526	0.2866366
## 4	0.01264201	0.09286967	0.09286967	-0.2681752	0.2934592
## 5	0.01264201	0.09741086	0.09741086	-0.2819067	0.3071907
## 6	0.01264201	0.10091859	0.10091859	-0.2925133	0.3177973
## 7	0.01264201	0.10256716	0.10256716	-0.2974982	0.3227822
## 8	0.01264201	0.10391547	0.10391547	-0.3015752	0.3268592
## 9	0.01264201	0.10517426	0.10517426	-0.3053815	0.3306655
## 10	0.01264201	0.10636545	0.10636545	-0.3089834	0.3342674

## 11	0.01264201	0.10725919	0.10725919	-0.3116859	0.3369699
## 12	0.01264201	0.10788589	0.10788589	-0.3135809	0.3388649
## 13	0.01264201	0.10841071	0.10841071	-0.3151678	0.3404518
## 14	0.01264201	0.10886176	0.10886176	-0.3165317	0.3418157
## 15	0.01264201	0.10923510	0.10923510	-0.3176606	0.3429446

Para el diagnóstico y el pronóstico, se procede como arriba.

Modelos GARCH

La modelación ARCH puede necesitar órdenes grandes; la modelación *Generalized AutoRegressive Conditional Heteroskedasticity*, GARCH, introducida por Bollerslev (1986), no necesita, por lo general, órdenes grandes, aún si comparte algunas desventajas con la modelación ARCH.

Si $r_t = \text{intc.returns}$ es la serie de retornos, sea

$$a_t = r_t + \mu_t$$

la innovación al tiempo t .

La serie u_t sigue un modelo $GARCH(m, s)$ si

$$u_t = \sigma_t \varepsilon_t$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^m \alpha_i u_{t-i}^2 + \sum_{j=1}^s \beta_j \sigma_{t-j}^2.$$

donde ε_t es una secuencia de variables aleatorias, con la misma distribución, con media 0 y varianza 1.

Una característica importante de la modelación GARCH es que la ecuación de la varianza condicional, σ_t^2 , se puede representar mediante un modelo ARMA para u_t^2 , más una función de una *martingala* (ruido blanco).

A continuación, se presentará el caso $GARCH(1, 1)$ por simplicidad de exposición:

$$u_t = \sigma_t \varepsilon_t$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 u_{t-1}^2 + \beta_1 \sigma_{t-1}^2.$$

Fortalezas y debilidades de la modelación GARCH

Las ventajas y desventajas son semejantes a las de la modelación ARCH.

Ventajas

- Se producen agrupaciones de volatilidad.

- las colas son más gruesas de una normal.
- Los pronósticos, $\sigma_h^2(\ell)$, se obtienen como los pronósticos de la modelación $ARMA(p, q)$.
A la larga,

$$\lim_{\ell \rightarrow \infty} \sigma_h^2(\ell) = \frac{\alpha_0}{1 - \alpha_1 - \beta_1} = Var(u_t).$$

Desventajas

- El modelo $GARCH(1, 1)$ necesita la condición $\alpha_1 + \beta_1 < 1$, para tener varianza finita y positiva.
- No se puede modelar el efecto de apalancamiento.
- En estudios empíricos de volatilidad de series de tiempo con alta frecuencia, la curtosis de los modelos GARCH no parece coincidir con las observaciones.

Estimación GARCH de INTC

A diferencia de la modelación ARCH, no existe evidencia en la determinación del orden de un modelo GARCH. Por lo general, se usan modelos $GARCH(1, 1)$, $GARCH(1, 2)$, $GARCH(2, 1)$ en los estudios empíricos.

A continuación se presenta la estimación $GARCH(1, 1)$ de *intc.returns*, retornos de INTC mensual. Después, se llevará a cabo el diagnóstico, complementando las herramientas usadas en la sección anterior con

- un intervalo predictivo alrededor de la ecuación dinámica de media;
- la serie de tiempo de la volatilidad, usando *volatility* de la librería *fBasics*, cargada por defecto cuando se carga *fGarch*.

```
intc.garchfit <- garchFit(~ 1 + garch(1,1), data = intc.rtn, trace = F)
summary(intc.garchfit)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(1, 1), data = intc.rtn, trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(1, 1)
## <environment: 0x56311e5222e8>
##  [data = intc.rtn]
##
## Conditional Distribution:
```

```

## norm
##
## Coefficient(s):
##      mu      omega      alpha1      beta1
## 8.2857e-03  7.1133e-05  5.7384e-02  9.3667e-01
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate  Std. Error  t value Pr(>|t|)
## mu      8.286e-03   4.212e-03   1.967 0.049140 *
## omega   7.113e-05   8.881e-05   0.801 0.423154
## alpha1  5.738e-02   1.664e-02   3.448 0.000565 ***
## beta1   9.367e-01   1.946e-02  48.125 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 408.8369      normalized: 0.8754538
##
## Description:
## Tue Jul 16 03:26:02 2019 by user: lino
##
##
## Standardised Residuals Tests:
##
##      Statistic p-Value
## Jarque-Bera Test  R      Chi^2 112.054 0
## Shapiro-Wilk Test  R      W      0.9801634 5.380887e-06
## Ljung-Box Test     R      Q(10) 9.230045 0.5104196
## Ljung-Box Test     R      Q(15) 14.81989 0.4644671
## Ljung-Box Test     R      Q(20) 17.82341 0.5990402
## Ljung-Box Test     R^2 Q(10) 3.101413 0.9789357
## Ljung-Box Test     R^2 Q(15) 6.011482 0.9795542
## Ljung-Box Test     R^2 Q(20) 7.323559 0.9954809
## LM Arch Test       R      TR^2 4.711439 0.9669294
##
## Information Criterion Statistics:
##      AIC      BIC      SIC      HQIC
## -1.733777 -1.698262 -1.733922 -1.719801

```

El modelo estimado es (innovaciones gaussianas)

$$r_t = .0082492 + u_t$$

$$\sigma_t^2 = .0001072 + .0587060u_{t-1}^2 + .9324359\sigma_{t-1}^2$$

Vale la pena subrayar que α_0 no es significativo en este modelo; el retorno promedio estimado es al límite de la significancia estadística, con valor-p asociado igual a .052853.

El pronóstico a la larga estimado es

$$\lim_{\ell \rightarrow \infty} \sigma_h^2(\ell) = \frac{.0001072}{1 - .0587060 - .9324359} = 0.01210191802.$$

La varianza muestral es comparable al valor asintótico estimado:

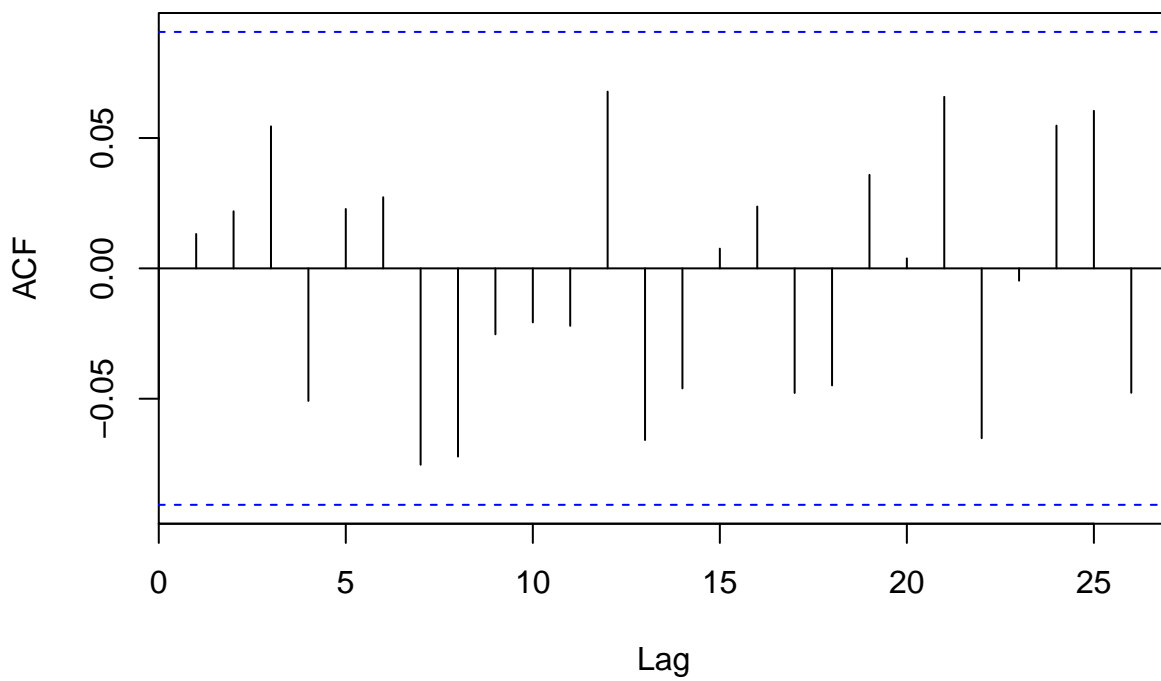
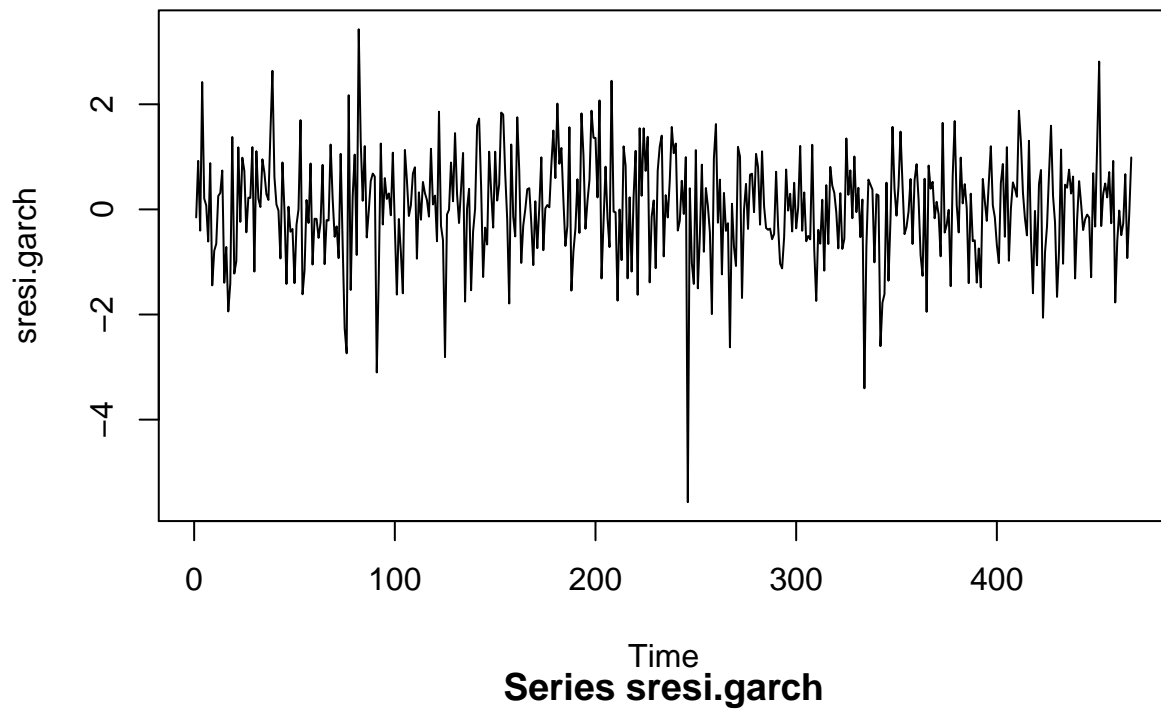
```
var(intc.rtn)
```

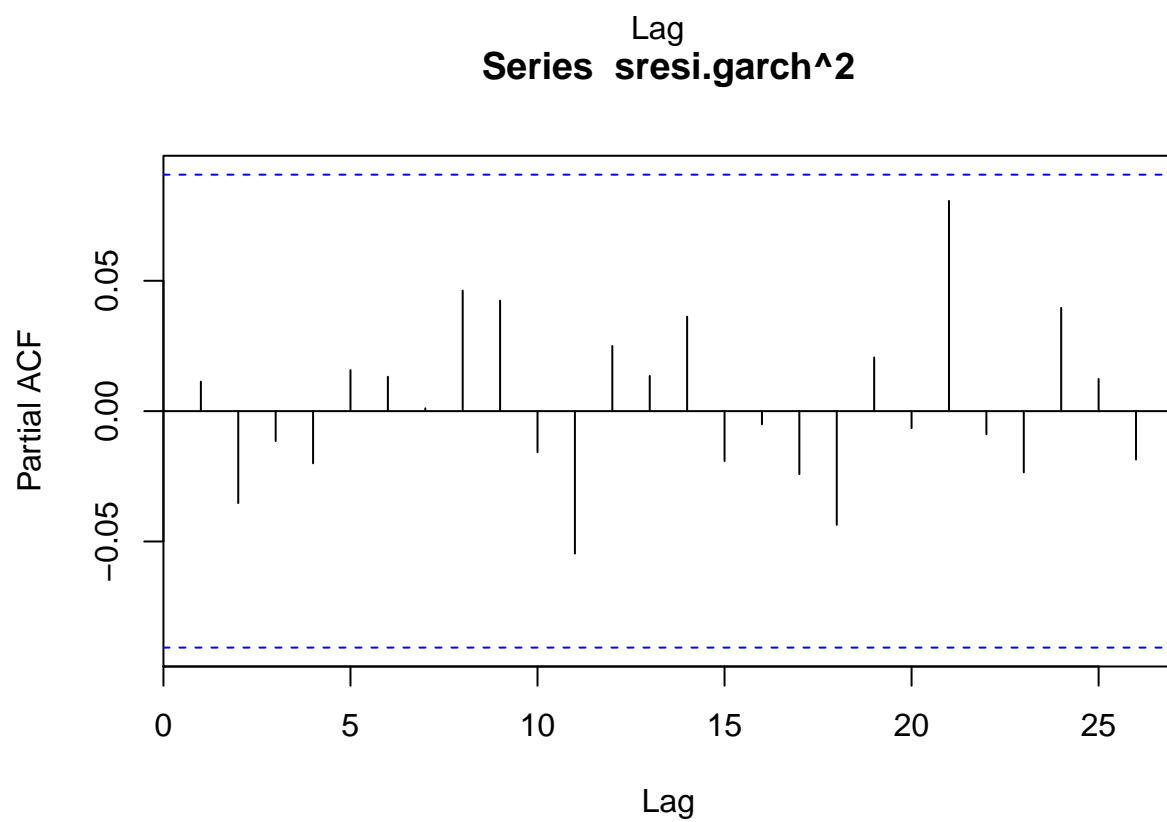
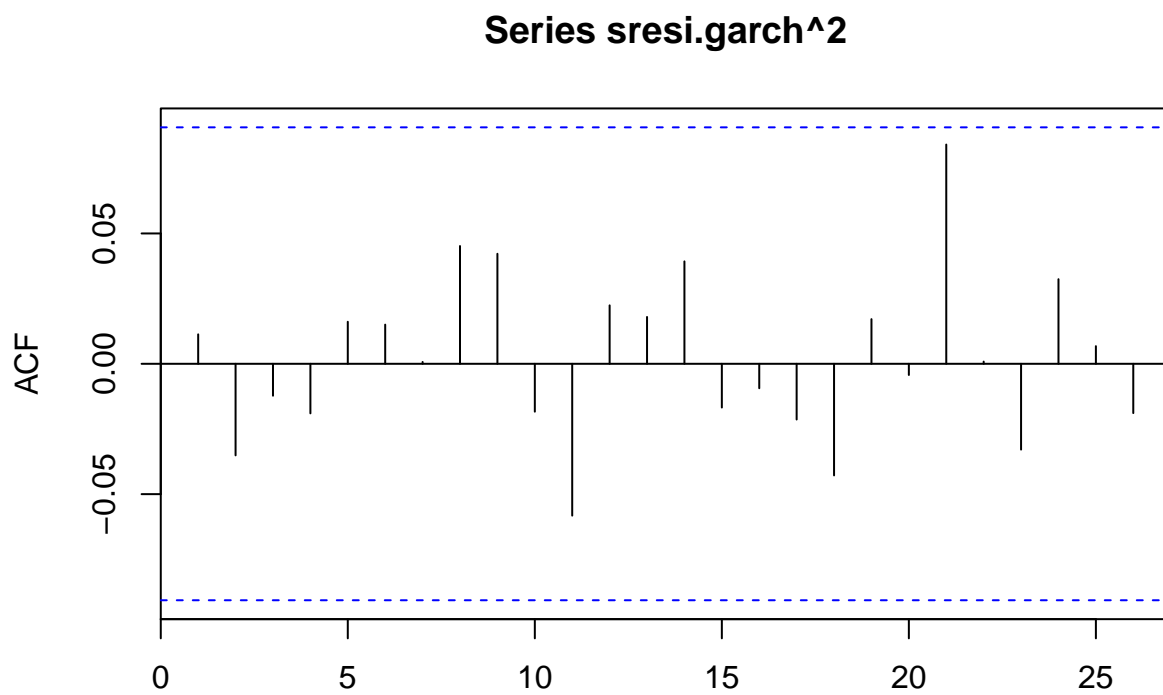
```
## [1] 0.01152514
```

Diagnóstico

Las pruebas de hipótesis a la base del diagnóstico son las mismas que vimos en la modelación ARCH.

ACF, PACF





Ljung-Box, lag = 10

```
Box.test(sresi.garch,lag=10,type='Ljung')
```

```
##  
## Box-Ljung test  
##  
## data: sresi.garch  
## X-squared = 9.23, df = 10, p-value = 0.5104
```

```
Box.test(sresi.garch^2,lag=10,type='Ljung')
```

```
##  
## Box-Ljung test  
##  
## data: sresi.garch^2  
## X-squared = 3.1014, df = 10, p-value = 0.9789
```

Ljung-Box, lag = 20

```
Box.test(sresi.garch,lag=20,type='Ljung')
```

```
##  
## Box-Ljung test  
##  
## data: sresi.garch  
## X-squared = 17.823, df = 20, p-value = 0.599
```

```
Box.test(sresi.garch^2,lag=20,type='Ljung')
```

```
##  
## Box-Ljung test  
##  
## data: sresi.garch^2  
## X-squared = 7.3236, df = 20, p-value = 0.9955
```

Ljung-Box, lag=30

```
Box.test(sresi.garch,lag=30,type='Ljung')
```

```
##  
## Box-Ljung test  
##  
## data: sresi.garch
```

```
## X-squared = 27.381, df = 30, p-value = 0.6032
```

```
Box.test(sresi.garch^2,lag=30,type='Ljung')
```

```
##
```

```
## Box-Ljung test
```

```
##
```

```
## data: sresi.garch^2
```

```
## X-squared = 12.915, df = 30, p-value = 0.9972
```

Normalidad de las innovaciones

Una implementación de la prueba de Jarque-Bera se encuentra en el paquete **fBasics** cargado simultáneamente con la librería **fGarch**, mientras que la prueba de Shapiro-Wilk se encuentra en la distribución **stats**, cargada por defecto.

```
jarqueberaTest(sresi.garch)
```

```
##
```

```
## Title:
```

```
## Jarque - Bera Normalality Test
```

```
##
```

```
## Test Results:
```

```
## STATISTIC:
```

```
## X-squared: 112.054
```

```
## P VALUE:
```

```
## Asymptotic p Value: < 2.2e-16
```

```
##
```

```
## Description:
```

```
## Tue Jul 16 03:26:03 2019 by user: lino
```

```
shapiroTest(sresi.garch)
```

```
##
```

```
## Title:
```

```
## Shapiro - Wilk Normality Test
```

```
##
```

```
## Test Results:
```

```
## STATISTIC:
```

```
## W: 0.9802
```

```
## P VALUE:
```

```
## 5.381e-06
```

```
##
```

```
## Description:
```

```
## Tue Jul 16 03:26:03 2019 by user: lino
```

El modelo parece ser adecuado, entre los modelos de orden bajo; las innovaciones gaussianas no son las correctas.

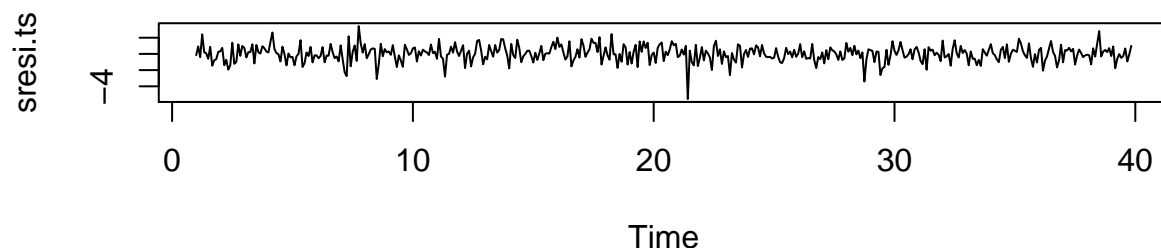
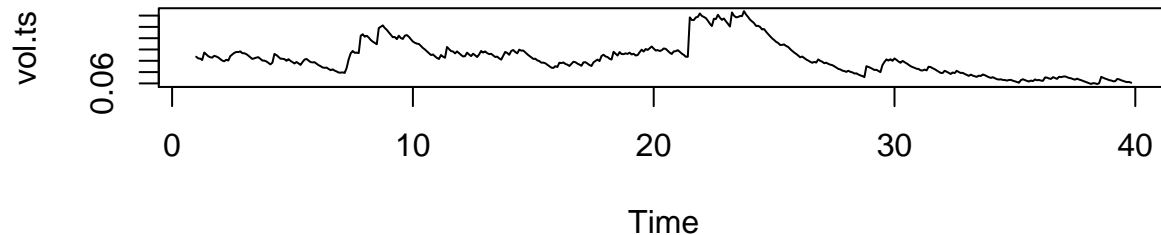
Intervalo predictivo alrededor de la media

En esta parte construiremos un intervalo de confianza, al 95%, alrededor del promedio estimado de los retornos.

Primeramente, se definirá la *serie de volatilidad* del modelo estimado.

A continuación se encuentra el código requerido:

```
v1 <- volatility(intc.garchfit) # volatilidad
sresi <- residuals(intc.garchfit, standardize = T) # residuos estandarizados
vol.ts <- ts(v1, frequency = 12) #volatilidad como serie de tiempo
sresi.ts <- ts(sresi, frequency = 12) # residuos como serie de tiempo
par(mfrow = c(2,1)) # dos gráficas al mismo tiempo
plot.ts(vol.ts)
plot.ts(sresi.ts)
```

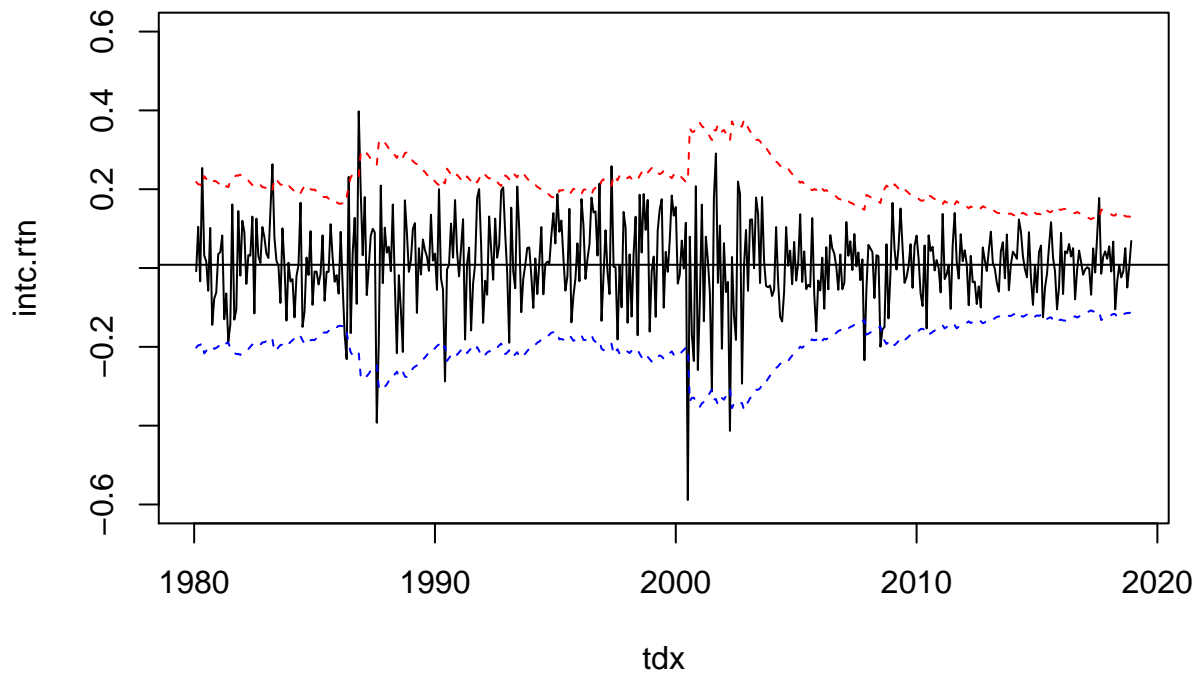


```
par(mfrow = c(1,1)) # regresar al default
```

Después, se define el intervalo de confianza:

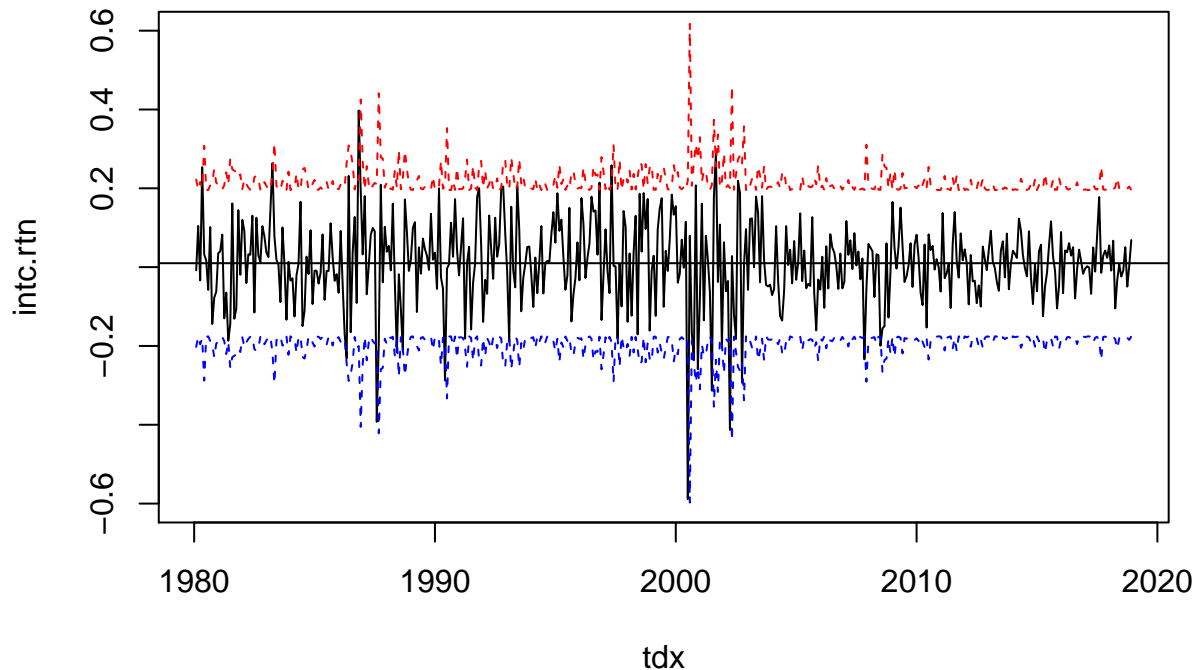
```
upper <- .0082492 + 1.96*v1
lower <- .0082492 - 1.96*v1
tdx <- c(1:467)/12+1980
plot(tdx,intc.rtn, type = "l", ylim = c(-.6,.6))
lines(tdx,upper, lty=2, col = "red")
```

```
lines(tdx,lower, lty=2, col = "blue")
abline(h=c(.0082492))
```



A comparación, se presenta el intervalo predictivo para el modelo $ARCH(1)$:

```
v2 <- volatility(intc.archfit2) # volatilidad
upper2 <- .009915 + 1.96*v2
lower2 <- .009915 - 1.96*v2
tdx <- c(1:467)/12+1980
plot(tdx,intc.rtn, type = "l", ylim = c(-.6,.6))
lines(tdx,upper2, lty=2, col = "red")
lines(tdx,lower2, lty=2, col = "blue")
abline(h=c(.009915))
```



Por lo general, la modelación $ARCH(m)$ sobreestima la volatilidad, mientras que $GARCH(1,1)$ sigue más la tendencia.

Modelo (G)ARCH con serie ARMA(p,q)

En esta parte del documento se analiza la modelación (G)ARCH del índice S&P 100 (símbolo del ticker `^OEX`) en el periodo 2/agosto/1982-4/marzo/2019, con frecuencia semanal. La serie se descargó de Yahoo! Finance, el día 6/marzo/2019.

Más específicamente, se considerarán dos modelos de ecuaciones de media:

$$\begin{aligned} r_t &= \mu + u_t & ARMA(0,0) \\ r_t &= \mu + \phi_1 r_{t-1} + \theta_1 u_{t-1} + u_t & ARMA(1,1) \end{aligned}$$

donde r_t es la serie de los retornos, al cierre, del índice S&P 100. Para cada ecuación de media, se estimará la volatilidad mediante $ARCH(m)$, $GARCH(m,s)$ y se determinará cuál de ellos mejor se ajusta a los retornos. Finalmente, se presentarán gráficas de intervalos predictivos de los modelos de volatilidad.

```
library(readr)
```

```
##
## Attaching package: 'readr'

## The following object is masked from 'package:TSA':
##
## spec
```

```
OEX <- read_csv("OEX.csv")

## Parsed with column specification:
## cols(
##   Date = col_date(format = ""),
##   Open = col_double(),
##   High = col_double(),
##   Low = col_double(),
##   Close = col_double(),
##   `Adj Close` = col_double(),
##   Volume = col_double()
## )

oex.ts <- ts(OEX$Close, frequency = 52)
oex.rtn <- diff(log(oex.ts))
```

Estimación de las ecuaciones de media

Modelo $ARMA(1,1)$ Se estimará el modelo usando la función `Arima` (librería `forecast`):

```
m1 <- Arima(oex.rtn, order = c(1,0,1))
m1

## Series: oex.rtn
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ma1      mean
##      -0.6951  0.6173  0.0017
## s.e.    0.1161  0.1269  0.0005
##
## sigma^2 estimated as 0.000514:  log likelihood=4521.46
## AIC=-9034.93   AICc=-9034.91   BIC=-9012.71
```

Modelo $ARMA(0,0)$ El modelo estimado es

```
m0 <- Arima(oex.rtn, order = c(0,0,0))
m0

## Series: oex.rtn
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##          mean
##      0.0017
## s.e. 0.0005
```

```
##
## sigma^2 estimated as 0.0005195: log likelihood=4510.36
## AIC=-9016.71 AICc=-9016.71 BIC=-9005.6
```

Nótese que la función `auto.arima()` tiene alguna inestabilidad numérica:

```
auto.arima(oex.rtn, stationary = T, seasonal = F)
```

```
## Series: oex.rtn
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
## Warning in sqrt(diag(x$var.coef)): NaNs produced
##          ar1      ar2      ma1      ma2      mean
##          0.1357 0.5722 -0.2132 -0.5064 0.0017
## s.e.      NaN      NaN      NaN      NaN 0.0005
##
## sigma^2 estimated as 0.0005145: log likelihood=4521.47
## AIC=-9030.93 AICc=-9030.89 BIC=-8997.61
```

El mejor ajuste lo proporciona el modelo $ARMA(1,1)$ (mínimo del criterio de información).

Las ecuaciones de media estimadas son

$$r_t = .0017 + u_t$$

$$r_t = .0017 - .6951r_{t-1} + .6173u_{t-1} + u_t$$

Nótese que el retorno promedio estimado es lo mismo,

$$\hat{\mu} = .0017 = .17\% \text{ diario,}$$

en ambos modelos.

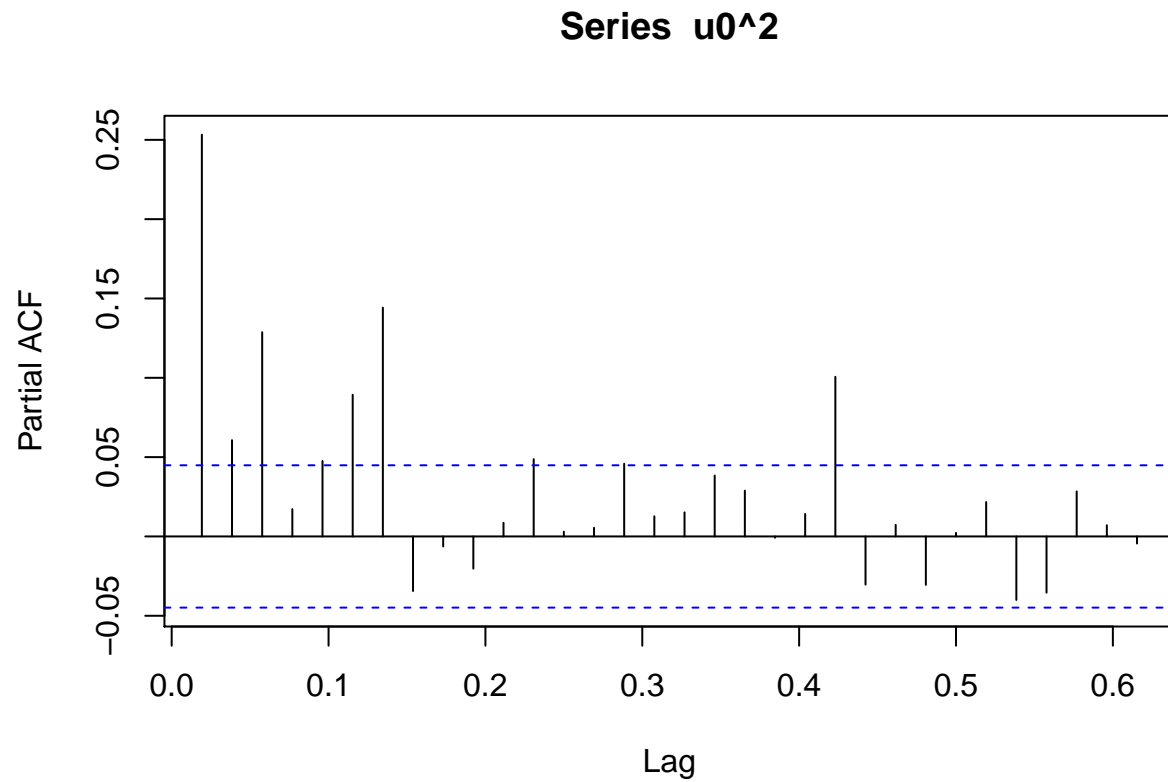
Detección de heterocedasticidad condicional

Para detectar el efecto $ARCH$, se grafica la PACF del cuadrado de los residuales de los dos modelos considerados.

$ARMA(0,0)$ Los residuales, en este caso, son la diferencia

$$u = oex.rtn - mean(oex.rtn)$$

```
u0 = oex.rtn - mean(oex.rtn)
pacf(u0^2)
```

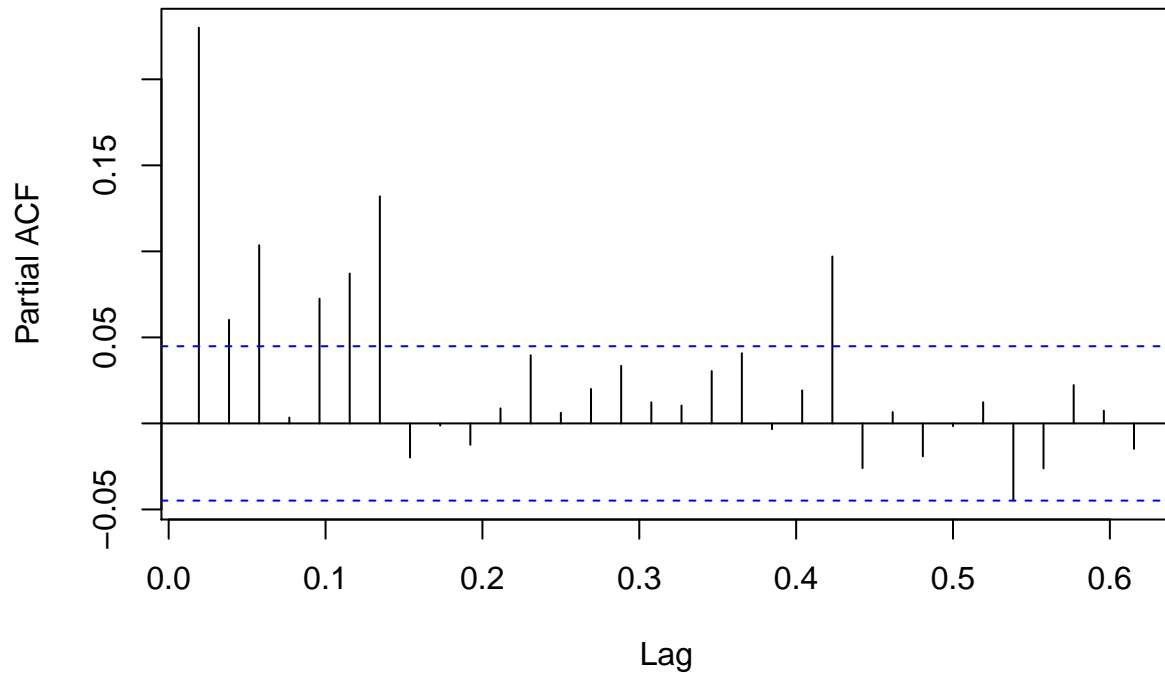



Se detecta el efecto de heterocedasticidad condicional; el orden posible es $m = 7$.

$ARMA(1, 1)$ Los residuales, en este caso, se extraen de `m1`, definido arriba

```
u1 = m1$residuals  
pacf(u1^2)
```

Series $u1^2$



También en este caso se detecta el efecto ARCH, con $m = 7$.

Los dos correlogramas parciales son muy semejantes.

Usando el script de R. Tsay, `archTest.R`:

```
# source("/Users/laan/pCloud Sync/2019/EcoFin2/Data/archTest.R") OSX
source("/home/lino/pCloudDrive/pCloud Sync/2019/EcoFin2/Data/archTest.R")
archTest(u0)
```

```
##
## Call:
## lm(formula = atsq ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.007101 -0.000358 -0.000230  0.000037  0.038337
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.409e-04  3.844e-05   6.268 4.51e-10 ***
## x1           2.213e-01  2.300e-02   9.620 < 2e-16 ***
## x2           1.611e-02  2.350e-02   0.685  0.49313
## x3           1.199e-01  2.349e-02   5.106 3.62e-07 ***
## x4          -1.063e-02  2.338e-02  -0.455  0.64931
## x5           2.715e-02  2.334e-02   1.163  0.24488
```

```
## x6          6.098e-02  2.334e-02   2.613  0.00904 **
## x7          1.447e-01  2.336e-02   6.194  7.16e-10 ***
## x8         -2.834e-02  2.344e-02  -1.209  0.22681
## x9         -2.478e-03  2.331e-02  -0.106  0.91535
## x10        -2.008e-02  2.278e-02  -0.882  0.37806
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001341 on 1888 degrees of freedom
## Multiple R-squared:  0.1148, Adjusted R-squared:  0.1101
## F-statistic: 24.48 on 10 and 1888 DF,  p-value: < 2.2e-16
```

```
archTest(u1)
```

```
##
## Call:
## lm(formula = atsq ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.007709 -0.000355 -0.000234  0.000039  0.040694
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.473e-04  3.963e-05   6.240 5.38e-10 ***
## x1          1.988e-01  2.301e-02   8.642 < 2e-16 ***
## x2          2.160e-02  2.340e-02   0.923  0.35624
## x3          9.938e-02  2.340e-02   4.247 2.27e-05 ***
## x4         -2.547e-02  2.330e-02  -1.093  0.27434
## x5          5.198e-02  2.326e-02   2.234  0.02558 *
## x6          6.211e-02  2.326e-02   2.671  0.00763 **
## x7          1.290e-01  2.329e-02   5.538 3.48e-08 ***
## x8         -1.544e-02  2.337e-02  -0.661  0.50873
## x9          6.775e-04  2.324e-02   0.029  0.97674
## x10        -1.234e-02  2.280e-02  -0.541  0.58854
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001383 on 1888 degrees of freedom
## Multiple R-squared:  0.0958, Adjusted R-squared:  0.09101
## F-statistic:    20 on 10 and 1888 DF,  p-value: < 2.2e-16
```

se obtiene el mismo resultado.

Estimación de la volatilidad

ARCH(7)

Los modelos de volatilidad que corresponden a las ecuación de media *ARMA*(0,0), *ARMA*(1,1), respectivamente, se dan a continuación. Se usarán innovaciones gaussianas, por el momento.

```
m0.arch <- garchFit(~ 1 + garch(7,0), data = oex.rtn, trace=F)
summary(m0.arch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(7, 0), data = oex.rtn, trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(7, 0)
## <environment: 0x563115a6deb8>
##  [data = oex.rtn]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##           mu           omega      alpha1      alpha2      alpha3      alpha4
## 0.00253793 0.00014271 0.23319848 0.10782265 0.17069795 0.08935739
##      alpha5      alpha6      alpha7
## 0.03463355 0.06016999 0.07333925
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##           Estimate Std. Error  t value Pr(>|t|)
## mu      0.0025379  0.0003972   6.389 1.67e-10 ***
## omega   0.0001427  0.0000163   8.755 < 2e-16 ***
## alpha1  0.2331985  0.0335866   6.943 3.83e-12 ***
## alpha2  0.1078227  0.0303336   3.555 0.000379 ***
## alpha3  0.1706979  0.0349662   4.882 1.05e-06 ***
## alpha4  0.0893574  0.0316763   2.821 0.004788 **
## alpha5  0.0346335  0.0289505   1.196 0.231579
## alpha6  0.0601700  0.0260157   2.313 0.020732 *
## alpha7  0.0733392  0.0245708   2.985 0.002837 **
```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 4724.782    normalized: 2.475004
##
## Description:
## Tue Jul 16 03:26:06 2019 by user: lino
##
##
## Standardised Residuals Tests:
##
##               Statistic p-Value
## Jarque-Bera Test   R      Chi^2 169.4947 0
## Shapiro-Wilk Test  R      W      0.9864706 1.954366e-12
## Ljung-Box Test     R      Q(10) 17.31091 0.06776115
## Ljung-Box Test     R      Q(15) 23.33913 0.07720017
## Ljung-Box Test     R      Q(20) 30.20709 0.06656686
## Ljung-Box Test     R^2  Q(10) 6.123565 0.8047771
## Ljung-Box Test     R^2  Q(15) 14.81764 0.4646307
## Ljung-Box Test     R^2  Q(20) 15.84277 0.7263297
## LM Arch Test       R      TR^2 7.584811 0.8166767
##
## Information Criterion Statistics:
##           AIC           BIC           SIC           HQIC
## -4.940578 -4.914392 -4.940622 -4.930941

```

```

m1.arch <- garchFit(~ arma(1,1) + garch(7,0), data = oex.rtn, trace=F)
summary(m1.arch)

```

```

##
## Title:
## GARCH Modelling
##
## Call:
## garchFit(formula = ~arma(1, 1) + garch(7, 0), data = oex.rtn,
##         trace = F)
##
## Mean and Variance Equation:
## data ~ arma(1, 1) + garch(7, 0)
## <environment: 0x563111ec35a0>
## [data = oex.rtn]
##
## Conditional Distribution:
## norm
##

```

```

## Coefficient(s):
##      mu      ar1      ma1      omega      alpha1
## 0.00408762 -0.60542578 0.52565495 0.00014286 0.23207564
##      alpha2      alpha3      alpha4      alpha5      alpha6
## 0.11058220 0.15819782 0.09482535 0.04190674 0.06808244
##      alpha7
## 0.06041675
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      4.088e-03 7.992e-04 5.114 3.15e-07 ***
## ar1     -6.054e-01 1.997e-01 -3.031 0.002435 **
## ma1      5.257e-01 2.145e-01 2.450 0.014269 *
## omega    1.429e-04 1.625e-05 8.789 < 2e-16 ***
## alpha1   2.321e-01 3.371e-02 6.885 5.79e-12 ***
## alpha2   1.106e-01 3.077e-02 3.594 0.000326 ***
## alpha3   1.582e-01 3.498e-02 4.522 6.12e-06 ***
## alpha4   9.483e-02 3.324e-02 2.853 0.004337 **
## alpha5   4.191e-02 2.992e-02 1.401 0.161323
## alpha6   6.808e-02 2.777e-02 2.451 0.014231 *
## alpha7   6.042e-02 2.342e-02 2.579 0.009896 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 4732.627      normalized: 2.479113
##
## Description:
## Tue Jul 16 03:26:07 2019 by user: lino
##
##
## Standardised Residuals Tests:
##
##      Statistic p-Value
## Jarque-Bera Test R Chi^2 181.7267 0
## Shapiro-Wilk Test R W 0.9853776 4.744285e-13
## Ljung-Box Test R Q(10) 6.544325 0.7676486
## Ljung-Box Test R Q(15) 11.4855 0.7174803
## Ljung-Box Test R Q(20) 19.16047 0.5114186
## Ljung-Box Test R^2 Q(10) 6.451189 0.7760399
## Ljung-Box Test R^2 Q(15) 16.60654 0.3429238
## Ljung-Box Test R^2 Q(20) 17.75223 0.603726
## LM Arch Test R TR^2 7.495411 0.8232162

```

```
##
## Information Criterion Statistics:
##      AIC      BIC      SIC      HQIC
## -4.946702 -4.914697 -4.946768 -4.934922
```

Se estima que el retorno promedio diario estimado en el primer modelo es

$$\hat{\mu} = .00253794 = .253794\%;$$

en el segundo modelo es igual a

$$\hat{\mu} = .00408753 / (1 + .60541229) = .002546 = .2546\%.$$

GARCH(1,1)

Los modelos de volatilidad que corresponden a las ecuación de media *ARMA*(0,0), *ARMA*(1,1), respectivamente, se dan a continuación. Se usarán innovaciones gaussianas, por el momento.

```
m0.garch <- garchFit(~ 1 + garch(1,1), data = oex.rtn, trace=F)
summary(m0.garch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(1, 1), data = oex.rtn, trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(1, 1)
## <environment: 0x56311575ad40>
##  [data = oex.rtn]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##      mu      omega      alpha1      beta1
## 2.3739e-03 1.9594e-05 1.6539e-01 8.0393e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      2.374e-03  3.991e-04   5.948 2.71e-09 ***
```

```

## omega 1.959e-05 5.526e-06 3.546 0.000392 ***
## alpha1 1.654e-01 2.452e-02 6.746 1.52e-11 ***
## beta1 8.039e-01 2.919e-02 27.544 < 2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 4729.495 normalized: 2.477472
##
## Description:
## Tue Jul 16 03:26:07 2019 by user: lino
##
##
## Standardised Residuals Tests:
##
## Statistic p-Value
## Jarque-Bera Test R Chi^2 211.1664 0
## Shapiro-Wilk Test R W 0.9853732 4.718079e-13
## Ljung-Box Test R Q(10) 18.74667 0.04360093
## Ljung-Box Test R Q(15) 24.20683 0.06166332
## Ljung-Box Test R Q(20) 31.96962 0.04362316
## Ljung-Box Test R^2 Q(10) 15.16067 0.1263131
## Ljung-Box Test R^2 Q(15) 20.03452 0.1706125
## Ljung-Box Test R^2 Q(20) 21.50074 0.368204
## LM Arch Test R TR^2 18.19986 0.1097554
##
## Information Criterion Statistics:
## AIC BIC SIC HQIC
## -4.950754 -4.939116 -4.950763 -4.946470

m1.garch <- garchFit(~ arma(1,1) + garch(1,1), data = oex.rtn, trace=F)
summary(m1.garch)

##
## Title:
## GARCH Modelling
##
## Call:
## garchFit(formula = ~arma(1, 1) + garch(1, 1), data = oex.rtn,
## trace = F)
##
## Mean and Variance Equation:
## data ~ arma(1, 1) + garch(1, 1)
## <environment: 0x563116f874c8>
## [data = oex.rtn]
##

```



```

## Conditional Distribution:
## norm
##
## Coefficient(s):
##      mu      ar1      ma1      omega      alpha1
## 3.8825e-03 -6.3095e-01  5.4764e-01  1.9659e-05  1.6362e-01
##      beta1
## 8.0488e-01
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      3.882e-03  7.603e-04   5.106 3.29e-07 ***
## ar1     -6.310e-01  1.834e-01  -3.439 0.000583 ***
## ma1      5.476e-01  1.984e-01   2.760 0.005782 **
## omega    1.966e-05  5.529e-06   3.556 0.000377 ***
## alpha1   1.636e-01  2.401e-02   6.813 9.55e-12 ***
## beta1    8.049e-01  2.893e-02  27.825 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 4738.687      normalized: 2.482288
##
## Description:
## Tue Jul 16 03:26:07 2019 by user: lino
##
##
## Standardised Residuals Tests:
##
##      Jarque-Bera Test  R      Chi^2  230.9513  0
##      Shapiro-Wilk Test  R      W      0.9840469 9.271252e-14
##      Ljung-Box Test    R      Q(10)  6.882724  0.7364682
##      Ljung-Box Test    R      Q(15)  11.28151  0.732395
##      Ljung-Box Test    R      Q(20)  19.80829  0.4699782
##      Ljung-Box Test    R^2  Q(10)  13.98725  0.1735742
##      Ljung-Box Test    R^2  Q(15)  19.50866  0.191602
##      Ljung-Box Test    R^2  Q(20)  21.34832  0.3768915
##      LM Arch Test      R      TR^2   16.6009  0.1652365
##
## Information Criterion Statistics:
##      AIC      BIC      SIC      HQIC
## -4.958289 -4.940832 -4.958309 -4.951864

```

El retorno promedio diario estimado en el primer modelo es

$$\hat{\mu} = .002374 = .2374\%;$$

en el segundo modelo es igual a

$$\hat{\mu} = .0038825/(1 + .63095) = .002381 = .2381\%.$$

Diagnóstico

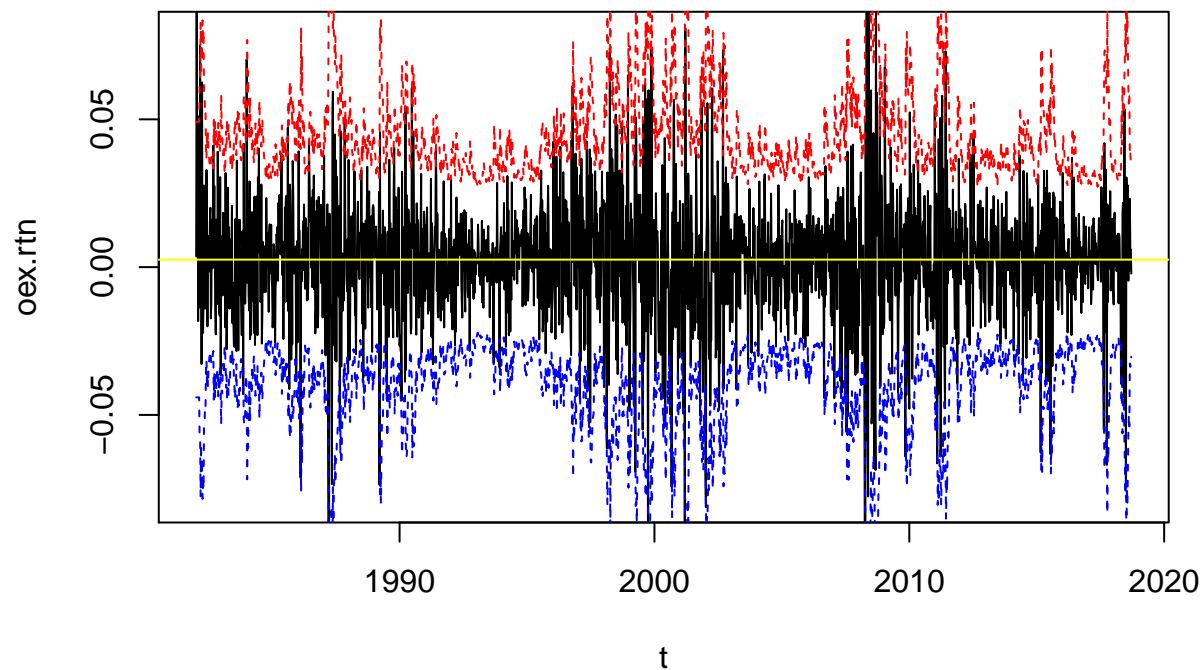
- El único modelo que no pasa uno de los diagnóstico es el $GARCH(1, 1)$, con ecuación de la media constante, `m0.garch`.
- Los criterios de información AIC, BIC son mejores (menores) en el modelo `m1.garch`.
- En ningún caso, las innovaciones son gaussianas.

Conclusión Usando el principio de parsimonia, concluimos que el $GARCH(1, 1)$ con ecuación de la media $ARMA(1, 1)$ es el mejor modelo de volatilidad.

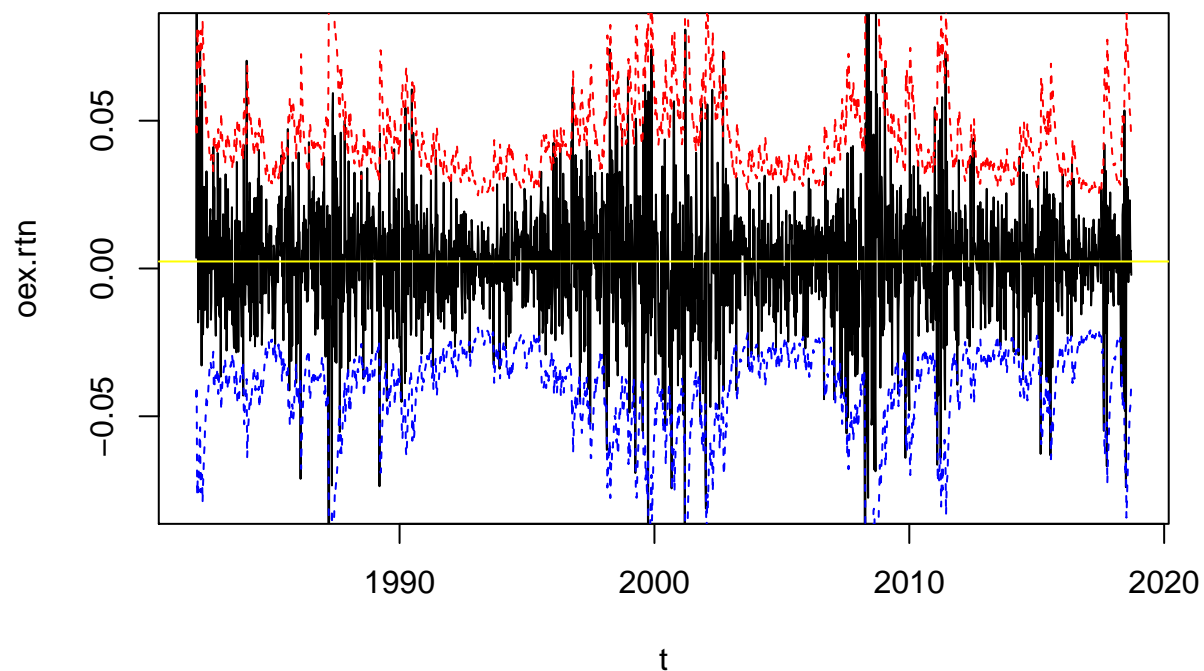
Gráficos con intervalos predictivos

A continuación se presentarán los intervalos predictivos de la volatilidad; por lo general, los modelos $GARCH(1, 1)$ siguen más la tendencia que los modelos $ARCH(7)$.

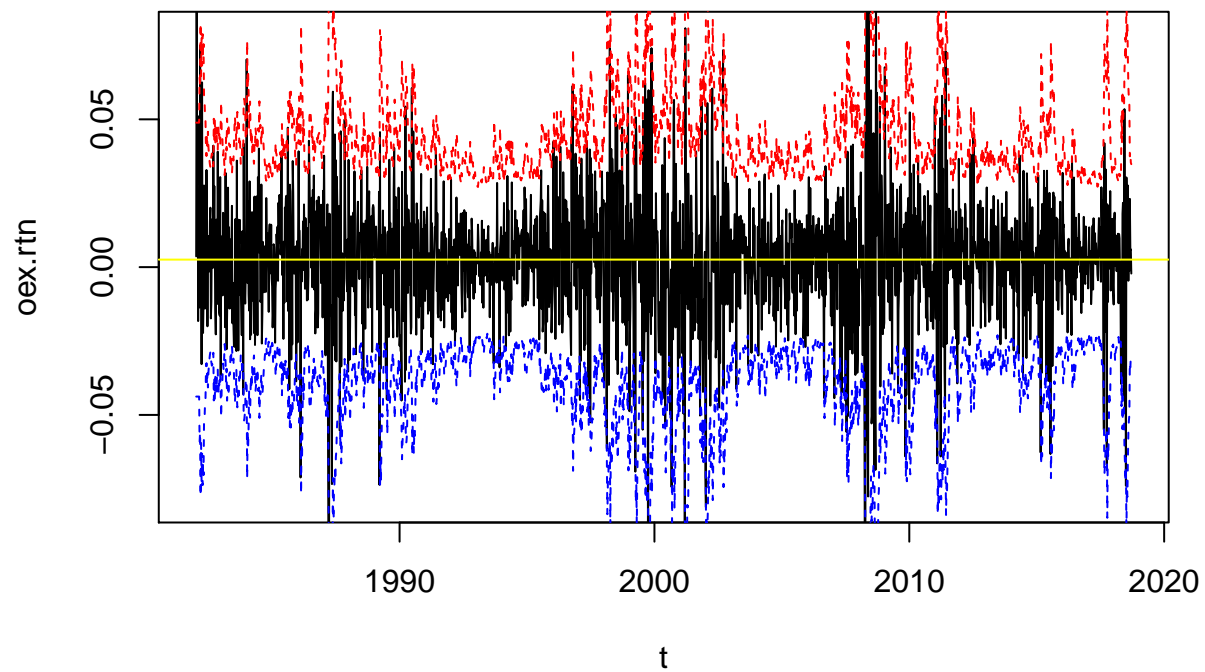
```
v0.arch <- volatility(m0.arch)
v1.arch <- volatility(m1.arch)
v0.garch <- volatility(m0.garch)
v1.garch <- volatility(m1.garch)
upper.v0.arch <- .0025379 + 2*v0.arch
lower.v0.arch <- .0025379 - 2*v0.arch
upper.v1.arch <- .002546 + 2*v1.arch
lower.v1.arch <- .002546 - 2*v1.arch
upper.v0.garch <- .002374 + 2*v0.garch
lower.v0.garch <- .002374 - 2*v0.garch
upper.v1.garch <- .002381 + 2*v1.garch
lower.v1.garch <- .002381 - 2*v1.garch
t <- c(1:1909)/52+1982
plot(t,oex.rtn, type = "l", ylim = c(-.08,.08))
lines(t,upper.v0.arch, lty=2, col = "red")
lines(t,lower.v0.arch, lty=2, col = "blue")
abline(h=c(.0025379), col = "yellow")
```



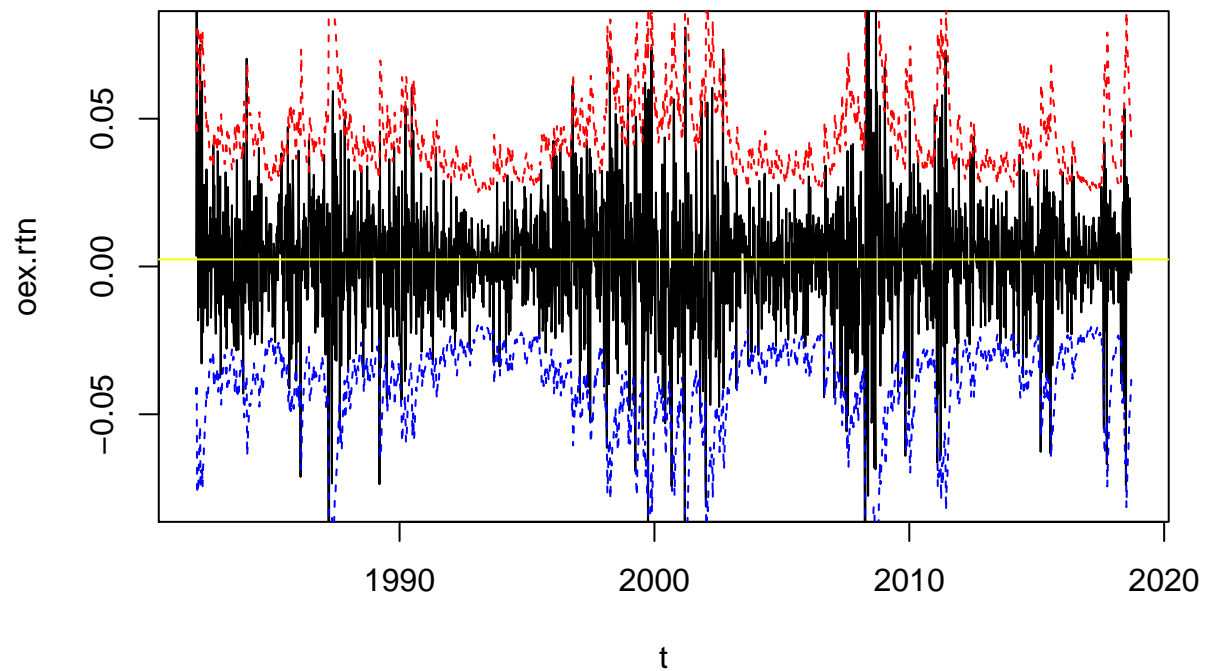
```
plot(t,oex.rtn, type = "l", ylim = c(-.08,.08))
lines(t,upper.v0.garch, lty=2, col = "red")
lines(t,lower.v0.garch, lty=2, col = "blue")
abline(h=c(.002374), col = "yellow")
```



```
plot(t,oex.rtn, type = "l", ylim = c(-.08,.08))
lines(t,upper.v1.arch, lty=2, col = "red")
lines(t,lower.v1.arch, lty=2, col = "blue")
abline(h=c(.002546), col = "yellow")
```



```
plot(t,oex.rtn, type = "l", ylim = c(-.08,.08))
lines(t,upper.v1.garch, lty=2, col = "red")
lines(t,lower.v1.garch, lty=2, col = "blue")
abline(h=c(.002381), col = "yellow")
```



Otros modelos

Modelo GARCH-M "GARCH in the mean"

Un modelo GARCH(1,1)-M es un modelo GARCH(1,1) en donde la ecuación de la media incluye un término de volatilidad:

$$\begin{aligned}r_t &= \mu + c\sigma_t^2 + u_t \\ u_t &= \sigma_t \varepsilon_t \\ \sigma_t^2 &= \alpha_0 + \alpha_1 u_{t-1}^2 + \beta_1 \sigma_{t-1}^2\end{aligned}$$

El coeficiente c se puede interpretar como una *prima de riesgo*.

Si $c > 0$, entonces los retornos son positivamente correlacionados con la volatilidad pasada.

Determina si los retornos de Intel Corporation tienen una prima de riesgo.

La hipótesis estadística es

$$\begin{aligned}H_0 : c &= 0 \\ H_1 : c &\neq 0\end{aligned}$$

```
library(rugarch)
```

```
## Loading required package: parallel
```

```
##
```

```
## Attaching package: 'rugarch'
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##      sigma
```

```
spec.archm = ugarchspec(variance.model=list(model="sGARCH"),
```

```
                      mean.model=list(armaOrder=c(0,0),archm=TRUE)) ## <-- opción "archm=TRUE"
```

```
fit.archm = ugarchfit(data=intc.rtn,spec=spec.archm)
```

```
fit.archm
```

```
##
```

```
## *-----*
```

```
## *          GARCH Model Fit          *
```

```
## *-----*
```

```
##
```

```
## Conditional Variance Dynamics
```

```
## -----
```

```
## GARCH Model   : sGARCH(1,1)
```

```
## Mean Model    : ARFIMA(0,0,0)
```

```
## Distribution   : norm
```

```

##
## Optimal Parameters
## -----
##      Estimate   Std. Error   t value   Pr(>|t|)
## mu      0.010447    0.015457    0.67585    0.49914
## archm  -0.023962    0.165007   -0.14522    0.88454
## omega   0.000070    0.000090    0.77538    0.43812
## alpha1  0.057199    0.016775    3.40988    0.00065
## beta1   0.936924    0.019774   47.38267    0.00000
##
## Robust Standard Errors:
##      Estimate   Std. Error   t value   Pr(>|t|)
## mu      0.010447    0.014564    0.71731    0.473185
## archm  -0.023962    0.166171   -0.14420    0.885344
## omega   0.000070    0.000127    0.55137    0.581381
## alpha1  0.057199    0.020545    2.78406    0.005368
## beta1   0.936924    0.028719   32.62429    0.000000
##
## LogLikelihood : 408.8473
##
## Information Criteria
## -----
##
## Akaike      -1.7295
## Bayes       -1.6851
## Shibata     -1.7298
## Hannan-Quinn -1.7121
##
## Weighted Ljung-Box Test on Standardized Residuals
## -----
##
##              statistic p-value
## Lag[1]              0.08134  0.7755
## Lag[2*(p+q)+(p+q)-1][2] 0.19402  0.8579
## Lag[4*(p+q)+(p+q)-1][5] 1.64742  0.7036
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----
##
##              statistic p-value
## Lag[1]              0.06424  0.7999
## Lag[2*(p+q)+(p+q)-1][5] 0.66850  0.9289
## Lag[4*(p+q)+(p+q)-1][9] 1.19218  0.9769
## d.o.f=2
##

```

```

## Weighted ARCH LM Tests
## -----
##           Statistic Shape Scale P-Value
## ARCH Lag[3]   0.07839 0.500 2.000  0.7795
## ARCH Lag[5]   0.27848 1.440 1.667  0.9459
## ARCH Lag[7]   0.36579 2.315 1.543  0.9889
##
## Nyblom stability test
## -----
## Joint Statistic:  0.9328
## Individual Statistics:
## mu      0.1363
## archm   0.1467
## omega   0.2808
## alpha1  0.1886
## beta1   0.1739
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:      1.28 1.47 1.88
## Individual Statistic:  0.35 0.47 0.75
##
## Sign Bias Test
## -----
##           t-value   prob sig
## Sign Bias      0.1103 0.9123
## Negative Sign Bias 0.1650 0.8690
## Positive Sign Bias 0.5254 0.5996
## Joint Effect    0.5444 0.9090
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----
##   group statistic p-value(g-1)
## 1    20      15.87      0.6660
## 2    30      29.90      0.4193
## 3    40      38.27      0.5031
## 4    50      42.53      0.7313
##
##
## Elapsed time : 0.619523

```

Se concluye que no existe evidencia de una prima de riesgo asociada con la volatilidad en los retornos de Intel.

Modelos EGARCH(1,1)

Un modelo EGARCH(1,1) se escribe como

$$\ln(\sigma_t^2) = \omega + g(\varepsilon_{t-1}) + \beta_1 \ln(\sigma_{t-1}^2),$$

donde

$$g(\varepsilon_{t-1}) = \alpha_1 \varepsilon_{t-1} + \gamma_1 [|\varepsilon_{t-1}| - E(|\varepsilon_{t-1}|)].$$

Nótese que

$$g(\varepsilon_{t-1}) = \begin{cases} (\alpha_1 - \gamma_1)\varepsilon_{t-1} - E(|\varepsilon_{t-1}|), & \varepsilon_{t-1} < 0 \\ (\alpha_1 + \gamma_1)\varepsilon_{t-1} - E(|\varepsilon_{t-1}|), & \varepsilon_{t-1} \geq 0 \end{cases}$$

Se determina que existe efecto de apalancamiento, si $\alpha_1 < 0$: innovaciones negativas tienen más impacto que innovaciones positivas.

Para probar un efecto de apalancamiento se prueba la hipótesis estadística

$$H_0 : \alpha_1 = 0$$

$$H_1 : \alpha_1 < 0$$

Problema Determinar si se observa un efecto de apalancamiento en los retornos de Intel Corporation.

```
spec.egarch = ugarchspec(variance.model=list(model="eGARCH"), ## <-- opción "eGARCH"
                        mean.model=list(armaOrder=c(0,0)))
fit.egarch = ugarchfit(data=intc.rtn,spec=spec.egarch)
fit.egarch
```

```
##
## *-----*
## *          GARCH Model Fit          *
## *-----*
##
## Conditional Variance Dynamics
## -----
## GARCH Model   : eGARCH(1,1)
## Mean Model    : ARFIMA(0,0,0)
## Distribution   : norm
##
## Optimal Parameters
## -----
##      Estimate Std. Error  t value Pr(>|t|)
## mu      0.010749   0.004130   2.6028 0.009247
## omega  -0.022665   0.008636  -2.6244 0.008679
## alpha1  0.032029   0.017668   1.8128 0.069857
## beta1   0.994881   0.001819 546.9206 0.000000
```



```

## gamma1  0.107795    0.018451    5.8422 0.000000
##
## Robust Standard Errors:
##      Estimate Std. Error  t value Pr(>|t|)
## mu      0.010749    0.004338   2.4778 0.013219
## omega  -0.022665    0.014031  -1.6154 0.106230
## alpha1  0.032029    0.026975   1.1873 0.235091
## beta1   0.994881    0.003022 329.1633 0.000000
## gamma1  0.107795    0.018062   5.9680 0.000000
##
## LogLikelihood : 410.5778
##
## Information Criteria
## -----
##
## Akaike      -1.7369
## Bayes       -1.6926
## Shibata     -1.7372
## Hannan-Quinn -1.7195
##
## Weighted Ljung-Box Test on Standardized Residuals
## -----
##
##              statistic p-value
## Lag[1]              0.04017  0.8412
## Lag[2*(p+q)+(p+q)-1][2] 0.08324  0.9316
## Lag[4*(p+q)+(p+q)-1][5] 2.10722  0.5933
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----
##
##              statistic p-value
## Lag[1]              0.9527  0.3290
## Lag[2*(p+q)+(p+q)-1][5] 1.0992  0.8368
## Lag[4*(p+q)+(p+q)-1][9] 2.0324  0.9011
## d.o.f=2
##
## Weighted ARCH LM Tests
## -----
##
##      Statistic Shape Scale P-Value
## ARCH Lag[3]  0.001577 0.500 2.000  0.9683
## ARCH Lag[5]  0.397469 1.440 1.667  0.9134
## ARCH Lag[7]  0.645448 2.315 1.543  0.9634
##
## Nyblom stability test

```

```
## -----
## Joint Statistic: 0.994
## Individual Statistics:
## mu      0.08271
## omega   0.19209
## alpha1  0.15712
## beta1   0.22414
## gamma1  0.02696
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:      1.28 1.47 1.88
## Individual Statistic: 0.35 0.47 0.75
##
## Sign Bias Test
## -----
##              t-value   prob sig
## Sign Bias      0.07595 0.9395
## Negative Sign Bias 0.68897 0.4912
## Positive Sign Bias 0.58188 0.5609
## Joint Effect      0.82339 0.8439
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----
##  group statistic p-value(g-1)
## 1      20      22.55      0.2577
## 2      30      29.12      0.4586
## 3      40      39.30      0.4566
## 4      50      39.75      0.8245
##
##
## Elapsed time : 0.1752253
```

Modelos EGARCH(1,1) con prima de riesgo

Se puede también considerar un modelo EGARCH(1,1), incluyendo una prima de riesgo en la ecuación de la media:

```
spec.egarch.m = ugarchspec(variance.model=list(model="eGARCH"), ## <-- opción "eGARCH"
mean.model=list(armaOrder=c(0,0), archm=T)) ## <-- prima de riesgo
fit.egarch.m = ugarchfit(data=intc.rtn,spec=spec.egarch.m)
fit.egarch.m
```

```
##
```

```

## *-----*
## *          GARCH Model Fit          *
## *-----*
##
## Conditional Variance Dynamics
## -----
## GARCH Model   : eGARCH(1,1)
## Mean Model    : ARFIMA(0,0,0)
## Distribution   : norm
##
## Optimal Parameters
## -----
##           Estimate  Std. Error   t value Pr(>|t|)
## mu         0.013201   0.011674    1.13087 0.258110
## archm      -0.027121   0.129318   -0.20972 0.833883
## omega      -0.023280   0.006472   -3.59698 0.000322
## alpha1     0.032038   0.017466    1.83430 0.066610
## beta1      0.994776   0.001353  735.47308 0.000000
## gamma1     0.106105   0.013115    8.09059 0.000000
##
## Robust Standard Errors:
##           Estimate  Std. Error   t value Pr(>|t|)
## mu         0.013201   0.012874    1.02546 0.305146
## archm      -0.027121   0.140384   -0.19319 0.846809
## omega      -0.023280   0.011056   -2.10554 0.035244
## alpha1     0.032038   0.025727    1.24532 0.213014
## beta1      0.994776   0.002275  437.34938 0.000000
## gamma1     0.106105   0.012286    8.63632 0.000000
##
## LogLikelihood : 410.5908
##
## Information Criteria
## -----
##
## Akaike          -1.7327
## Bayes           -1.6795
## Shibata         -1.7330
## Hannan-Quinn   -1.7118
##
## Weighted Ljung-Box Test on Standardized Residuals
## -----
##                               statistic p-value
## Lag[1]                                0.04369 0.8344
## Lag[2*(p+q)+(p+q)-1][2]          0.08878 0.9276
## Lag[4*(p+q)+(p+q)-1][5]          2.14694 0.5841

```

```

## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----
##
##                statistic p-value
## Lag[1]                1.013  0.3142
## Lag[2*(p+q)+(p+q)-1] [5]    1.159  0.8228
## Lag[4*(p+q)+(p+q)-1] [9]    2.130  0.8889
## d.o.f=2
##
## Weighted ARCH LM Tests
## -----
##                Statistic Shape Scale P-Value
## ARCH Lag[3] 0.0005392 0.500 2.000  0.9815
## ARCH Lag[5] 0.4019234 1.440 1.667  0.9121
## ARCH Lag[7] 0.6714733 2.315 1.543  0.9603
##
## Nyblom stability test
## -----
## Joint Statistic:  1.0607
## Individual Statistics:
## mu      0.08819
## archm   0.09258
## omega   0.18445
## alpha1  0.15763
## beta1   0.21555
## gamma1  0.02736
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:      1.49 1.68 2.12
## Individual Statistic:  0.35 0.47 0.75
##
## Sign Bias Test
## -----
##                t-value   prob sig
## Sign Bias      0.1369 0.8912
## Negative Sign Bias 0.6781 0.4980
## Positive Sign Bias 0.6193 0.5360
## Joint Effect    0.8681 0.8331
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----
## group statistic p-value(g-1)

```

```
## 1      20      20.58      0.3604
## 2      30      31.57      0.3393
## 3      40      45.12      0.2314
## 4      50      35.25      0.9301
##
##
## Elapsed time : 0.4790499
```

Modelos VAR (Vectoriales Autoregresivos)

Modelos VAR analizan dos, o más, series de tiempo en conjunto.

Un modelo $VAR(p)$ para las series de tiempo x_t, y_t es

$$x_t = \beta_{10} + \beta_{11}x_{t-1} + \cdots + \beta_{1p}x_{t-p} + \gamma_{11}y_{t-1} + \cdots + \gamma_{1p}y_{t-p} + u_{1,t}$$

$$y_t = \beta_{20} + \beta_{21}x_{t-1} + \cdots + \beta_{2p}x_{t-p} + \gamma_{21}y_{t-1} + \cdots + \gamma_{2p}y_{t-p} + u_{2,t}$$

A diferencia de la modelación de ecuaciones simultáneas (con variables instrumentales o usando mínimos cuadrados en dos etapas), la modelación VAR no necesita distinguir entre variables exógenas y endógenas: todas las variables son endógenas.

Una desventaja de la modelación VAR, por otro lado, es que las ecuaciones deben tener el mismo número de rezagos, por lo que el número de parámetros a estimar puede ser muy grande. Si el tamaño muestral no es tan grande, una modelación VAR podría reducir el número de grados de libertad de manera importante.

Algunas de las aplicaciones que se verán son si, y de qué manera, una variable (y sus rezagos) afectan la otra variable (Causalidad según Granger); cuánto es sensible una variable a impulsos unitarios y qué proporción de variación de una variable se obtiene por variaciones de la misma (y de la otra) variable. Finalmente, se abordará el tópico de cambio estructural.

Regresiones espurias y estacionariedad

Una situación que se puede presentar, especialmente cuando no se conoce a fondo la naturaleza de los datos que se están manejando, es encontrar relaciones artificiales que son significativas estadísticamente. Véase el artículo clásico de Yule, [Y].

Una regresión espuria es un ejemplo de tales relaciones. El fuerte resultado estadístico es consecuencia de un efecto de tendencia o de no estacionariedad de los datos.

Considérese, por ejemplo, el caso de dos caminatas aleatorias independientes. Si se regresa una sobre la otra, entonces se encontrará una estimación que es estadísticamente significativa que, pero es completamente artificial. Presentamos un ejemplo de esta relación a continuación. El valor de la semilla es para reproducibilidad de los resultados.

Se definen primeramente las dos caminatas aleatorias, x_1 , y_1 :

```
set.seed(123)
n <- 1000
x1 <- cumsum(sample(c(-1, 1), n, TRUE))
y1 <- cumsum(sample(c(-1, 1), n, TRUE))
```

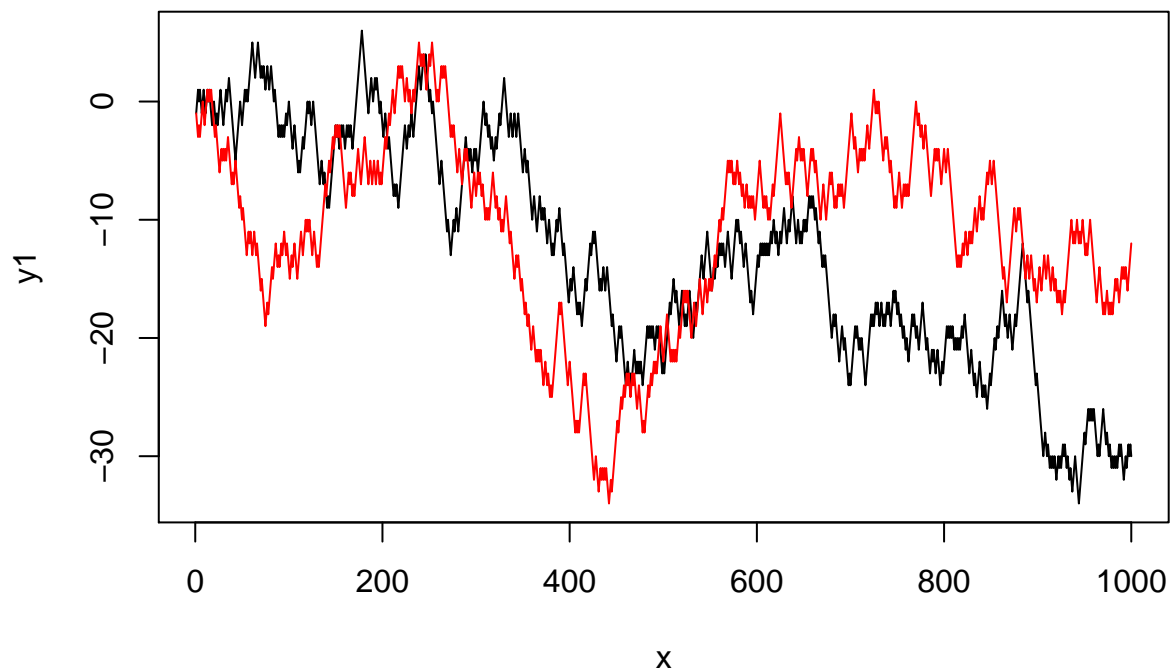
Se calcula un prueba de hipótesis de correlación entre x , y y se grafican:

```
cor.test(x1,y1)

##
## Pearson's product-moment correlation
##
## data: x1 and y1
## t = 11.125, df = 998, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2758382 0.3861925
## sample estimates:
##      cor
## 0.3321515
```

La correlación es positiva, de manera altamente significativa. Las gráficas a continuación de x_1 , y_1 nos permite comprender esta situación.

```
plot(y1, type = "l", xlab = "x")
lines(x1, col="red")
```



Después, regresamos y sobre x y desplegamos el resultado

De-

```
lm1 <- lm(y1~x1)
summary(lm1)
```

```
##
## Call:
## lm(formula = y1 ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6241  -7.2949   0.0017   7.9444  18.7788
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.54151    0.48827  -17.49  <2e-16 ***
## x1           0.40287    0.03621   11.12  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.06 on 998 degrees of freedom
## Multiple R-squared:  0.1103, Adjusted R-squared:  0.1094
## F-statistic: 123.8 on 1 and 998 DF,  p-value: < 2.2e-16
```

Podemos ver el intercepto y la pendiente son significativas, el modelo también lo es y el valor de $R^2 \approx .07$.

Una manera de darse cuenta de que se está tratando de una regresión espuria¹ es verificar que los residuos forman una serie cno estacionaria, como se ve a continuación:

```
library(tseries)
adf.test(as.ts(lm1$residuals))
```

```
## Warning in adf.test(as.ts(lm1$residuals)): p-value smaller than printed p-
## value
##
## Augmented Dickey-Fuller Test
##
## data:  as.ts(lm1$residuals)
## Dickey-Fuller = -4.9177, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

La regresión espuria pierde la “curiosidad” estadística cuando se elimina la tendencia:

```
x1.diff <- diff(x1)
y1.diff <- diff(y1)
```

¹La mejor manera de no incurrir en esto tipo de relaciones artificiales es siempre conocer los datos que se están analizando.

```
lm1.diff <- lm(y1.diff ~ x1.diff)
summary(lm1.diff)
```

```
##
## Call:
## lm(formula = y1.diff ~ x1.diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.9838 -0.9838 -0.9584  1.0162  1.0416
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02889    0.03166  -0.913   0.362
## x1.diff      0.01269    0.03166   0.401   0.688
##
## Residual standard error: 1 on 997 degrees of freedom
## Multiple R-squared:  0.0001613, Adjusted R-squared:  -0.0008416
## F-statistic: 0.1608 on 1 and 997 DF,  p-value: 0.6885
```

```
adf.test(as.ts(lm1.diff$residuals))
```

```
## Warning in adf.test(as.ts(lm1.diff$residuals)): p-value smaller than
## printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data:  as.ts(lm1.diff$residuals)
## Dickey-Fuller = -10.269, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

Por otro lado, cuando se usan variables como el precio del índice S&P's500 y el precio de las acciones de MorganStanley, entonces los resultados de un análisis semejante ya no son una “curiosidad estadística”.

A continuación se definen las series de tiempo, de MorganStanley y de S&P500; se estiman dos modelos de regresión lineal (a nivel de precio; con retornos) y se presentan los resultados de las pruebas de hipótesis de estacionariedad de los residuos.

```
library(readr)
library(forecast)
library(tseries)
# spms <- read.csv("/Users/laan/pCloud Sync/2019/EcoFin2/Data/SPMS.csv") OSX
spms <- read.csv("/home/lino/pCloudDrive/pCloud Sync/2019/EcoFin2/Data/SPMS.csv")
sp.ts <- ts(spms$SP, frequency = 52)
ms.ts <- ts(spms$MS, frequency = 52)
```



```

m.level <- lm(ms.ts ~ sp.ts)
summary(m.level)

##
## Call:
## lm(formula = ms.ts ~ sp.ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.604 -15.287  -6.962  14.260  67.234
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.741105   1.263248   16.42  <2e-16 ***
## sp.ts         0.129279   0.008736   14.80  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.6 on 1359 degrees of freedom
## Multiple R-squared:  0.1388, Adjusted R-squared:  0.1381
## F-statistic: 219 on 1 and 1359 DF, p-value: < 2.2e-16

adf.test(m.level$residuals)

##
## Augmented Dickey-Fuller Test
##
## data: m.level$residuals
## Dickey-Fuller = -1.889, Lag order = 11, p-value = 0.6253
## alternative hypothesis: stationary

ms.rtn <- diff(log(ms.ts))
sp.rtn <- diff(log(sp.ts))
m.rtn <- lm(ms.rtn ~ sp.rtn)
summary(m.rtn)

##
## Call:
## lm(formula = ms.rtn ~ sp.rtn)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46236 -0.02167 -0.00162  0.01994  0.58482
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)

```

```
## (Intercept) -0.001513    0.001221   -1.238    0.216
## sp.rtn      1.998431    0.052479   38.081   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04497 on 1358 degrees of freedom
## Multiple R-squared:  0.5164, Adjusted R-squared:  0.5161
## F-statistic: 1450 on 1 and 1358 DF,  p-value: < 2.2e-16
```

```
adf.test(m.rtn$residuals)
```

```
## Warning in adf.test(m.rtn$residuals): p-value smaller than printed p-value
##
## Augmented Dickey-Fuller Test
##
## data:  m.rtn$residuals
## Dickey-Fuller = -11.351, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

De hecho, el modelo sobre los retornos (segundo modelo estimado) tiene un R^2 más grande que el R^2 de modelo a nivel.

Estimación VAR

En la estimación VAR se usará la librería **vars**:

```
library(vars)
```

```
## Loading required package: MASS
## Loading required package: strucchange
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following object is masked from 'package:timeSeries':
##
##     time<-
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package: sandwich
## Loading required package: urca
```

```
## Loading required package: lmtest
```

Las funciones y métodos de esta librería manejan dataframes, por lo que se necesita crear un dataframe de los retornos:

```
spms.rtn <- data.frame(sp.rtn, ms.rtn)
```

Determinación del orden y estimación

La determinación del orden del modelo $VAR(p)$ se realiza especificando el orden máximo a usar:

```
VARselect(spms.rtn, lag.max = 12, type = "const")
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      11      2      2      11
##
## $criteria
##              1              2              3              4
## AIC(n) -1.377283e+01 -1.378908e+01 -1.378902e+01 -1.378438e+01
## HQ(n)  -1.376415e+01 -1.377461e+01 -1.376877e+01 -1.375835e+01
## SC(n)  -1.374966e+01 -1.375045e+01 -1.373495e+01 -1.371486e+01
## FPE(n)  1.043605e-06  1.026788e-06  1.026846e-06  1.031616e-06
##              5              6              7              8
## AIC(n) -1.378601e+01 -1.379143e+01 -1.380021e+01 -1.379940e+01
## HQ(n)  -1.375418e+01 -1.375382e+01 -1.375682e+01 -1.375021e+01
## SC(n)  -1.370104e+01 -1.369101e+01 -1.368434e+01 -1.366808e+01
## FPE(n)  1.029943e-06  1.024379e-06  1.015416e-06  1.016247e-06
##              9             10             11             12
## AIC(n) -1.379607e+01 -1.379669e+01 -1.380659e+01 -1.380297e+01
## HQ(n)  -1.374110e+01 -1.373593e+01 -1.374005e+01 -1.373064e+01
## SC(n)  -1.364930e+01 -1.363447e+01 -1.362892e+01 -1.360986e+01
## FPE(n)  1.019636e-06  1.019007e-06  1.008971e-06  1.012627e-06
```

Elegimos una estimación sin tendencia, `type = "const"`, ya que los datos son estacionarios.

La elección del orden se realiza con base en uno de los criterios de información. Para empezar, seleccionamos $p = 2$, usando el criterio de Schwarz bayesiano (SC).

Pasamos a la estimación del modelo elegido

```
spms.var1 <- VAR(spms.rtn, p=2, type = "const")
summary(spms.var1)
```

```
##
## VAR Estimation Results:
## =====
```

```

## Endogenous variables: sp.rtn, ms.rtn
## Deterministic variables: const
## Sample size: 1358
## Log Likelihood: 5521.448
## Roots of the characteristic polynomial:
## 0.3532 0.3532 0.2491 0.2491
## Call:
## VAR(y = spms.rtn, p = 2, type = "const")
##
##
## Estimation results for equation sp.rtn:
## =====
## sp.rtn = sp.rtn.l1 + ms.rtn.l1 + sp.rtn.l2 + ms.rtn.l2 + const
##
##           Estimate Std. Error t value Pr(>|t|)
## sp.rtn.l1 -0.0644144  0.0389744  -1.653 0.098616 .
## ms.rtn.l1 -0.0078784  0.0142521  -0.553 0.580497
## sp.rtn.l2 -0.0603850  0.0389048  -1.552 0.120867
## ms.rtn.l2  0.0526934  0.0142280   3.703 0.000221 ***
## const      0.0014581  0.0006284   2.320 0.020474 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.02305 on 1353 degrees of freedom
## Multiple R-Squared: 0.02107, Adjusted R-squared: 0.01817
## F-statistic: 7.279 on 4 and 1353 DF,  p-value: 8.47e-06
##
##
## Estimation results for equation ms.rtn:
## =====
## ms.rtn = sp.rtn.l1 + ms.rtn.l1 + sp.rtn.l2 + ms.rtn.l2 + const
##
##           Estimate Std. Error t value Pr(>|t|)
## sp.rtn.l1  0.154263  0.106972   1.442  0.1495
## ms.rtn.l1 -0.234362  0.039117  -5.991 2.66e-09 ***
## sp.rtn.l2 -0.230491  0.106780  -2.159  0.0311 *
## ms.rtn.l2  0.072916  0.039051   1.867  0.0621 .
## const      0.001426  0.001725   0.827  0.4085
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.06326 on 1353 degrees of freedom
## Multiple R-Squared: 0.04542, Adjusted R-squared: 0.0426

```

```
## F-statistic: 16.09 on 4 and 1353 DF,  p-value: 7.001e-13
##
##
##
## Covariance matrix of residuals:
##          sp.rtn  ms.rtn
## sp.rtn 0.0005312 0.001054
## ms.rtn 0.0010536 0.004002
##
## Correlation matrix of residuals:
##          sp.rtn ms.rtn
## sp.rtn 1.0000 0.7227
## ms.rtn 0.7227 1.0000
```

La matriz de correlación será útil más adelante para la interpretación de la respuesta al impulso y de la descomposición de varianza del error de pronóstico (FEVD, el acrónimo en inglés).

Diagnóstico

El diagnóstico del modelo estimado considera las siguientes pruebas:

- Pruebas de ruido blanco sobre los residuales:
 - Prueba portmanteau(Ljung-Box; Box-Pierce)
 - Prueba de Breusch-Godfrey de correlación serial
- Prueba de efectos ARCH
- Prueba de normalidad
- Prueba de estabilidad

Pruebas de ruido blanco Se considera que los residuales cumplen el supuesto de ruido blanco si no se rechaza la hipótesis nula

$$H_0 : \tau_1 = 0, \dots, \tau_s = 0$$

$$H_1 : \tau_1 \neq 0, \dots, \tau_s \neq 0$$

con $s > 1$.

Prueba portmanteau El estadístico de prueba es una χ_m^2 , con m el número de grados de libertad del modelo. Para muestras pequeñas, es disponible también un estadístico corregido.

```
serial.test(spms.var1, lags.pt = 16, type = "PT.asymptotic")
```

```
##
## Portmanteau Test (asymptotic)
##
```

```
## data: Residuals of VAR object spms.var1
## Chi-squared = 138.56, df = 56, p-value = 6.023e-09
serial.test(spms.var1, lags.pt = 16, type = "PT.adjusted")
```

```
##
## Portmanteau Test (adjusted)
##
## data: Residuals of VAR object spms.var1
## Chi-squared = 139.58, df = 56, p-value = 4.391e-09
```

Existe evidencia de autocorrelación serial (los residuales no son ruido blanco).

Prueba de Breusch-Godfrey Es una prueba ML, con estadístico de prueba es una χ_n^2 , n número de grados de libertad. También hay un estadístico ajustado para muestra pequeña, bajo el nombre prueba F de Edgerton-Shukur:

```
serial.test(spms.var1, lags.bg = 15, type = "BG")
```

```
##
## Breusch-Godfrey LM test
##
## data: Residuals of VAR object spms.var1
## Chi-squared = 142.38, df = 60, p-value = 1.194e-08
```

```
serial.test(spms.var1, lags.bg = 15, type = "ES")
```

```
##
## Edgerton-Shukur F test
##
## data: Residuals of VAR object spms.var1
## F statistic = 2.4385, df1 = 60, df2 = 2644, p-value = 7.414e-09
```

Después de las cuatro pruebas, concluimos que existe fuerte evidencia de autocorrelación serial. Para eliminarla, se aumentará el orden del modelo VAR.

Causalidad según Granger

Considérese el modelo $VAR(p)$

$$\begin{aligned}x_t &= \beta_{10} + \beta_{11}x_{t-1} + \cdots + \beta_{1p}x_{t-p} + \gamma_{11}y_{t-1} + \cdots + \gamma_{1p}y_{t-p} + u_{1,t} \\ y_t &= \beta_{20} + \beta_{21}x_{t-1} + \cdots + \beta_{2p}x_{t-p} + \gamma_{21}y_{t-1} + \cdots + \gamma_{2p}y_{t-p} + u_{2,t}\end{aligned}$$

Definición Se dice que la serie y_t causa (según Granger) la variable x_t si los rezagos de y_t explican el valor actual de x_t .

Hay una definición semejante para x_t que causa y_t .

Para probar si y_t causa x_t , se plantea la siguiente prueba de hipótesis

$$H_0 : \gamma_{11} = 0, \dots \gamma_{1p} = 0$$

$$H_1 : \gamma_{11} \neq 0, \dots \gamma_{1p} \neq 0$$

La hipótesis nula es equivalente a decir que los rezagos de y_t no explican el valor actual de x_t .

Si se rechaza la hipótesis nula, entonces se dice que y_t causa x_t .

Se prueban hipótesis estadísticas semejantes para determinar si

- los rezagos de la variable x_t explican el valor actual de y_t ;
- los rezagos de la variable y_t (resp., x_t) explican el valor actual de la misma variable y_t (resp., x_t).

En la prueba de causalidad de Granger se puede usar el estadístico F si las series sean estacionaria.

Comentarios

- Vale la pena subrayar que la causalidad según Granger no implica que una variable es causa de la otra. Analizando activos financieros, se puede decir, más bien, que un activo “lidera” (o anticipa) el otro en los cambios de sus valores.
- Si x_t causa y_t , y viceversa, entonces se dice que existe “feedback” bidireccional.
- Si, por ejemplo, x_t causa y_t , pero y_t no causa x_t , se dice que x_t es fuertemente exógena en la ecuación para y_t .
- Si ninguna variable causa la otra, se dice que x_t , y_t son independientes.

Respuesta a impulso unitario

En la estimación de un modelo $VAR(p)$, la causalidad de Granger permite determinar si una variable depende de los rezagos de las variables en el modelo, pero no cuantifica esta relación.

La respuesta a impulso unitario permite la cuantificación de esta relación mediante la respuesta del modelo $VAR(p)$ a choques aplicados a cada variable del modelo.

Bibliografía

- Br** Brooks C, *Introductory Econometrics for Finance*, Third edition, Cambridge University Press, 2014.
- BP** Box GEP, Pierce DA., "Distribution of residual autocorrelation in autoregressive-integrated moving average time series models." J Am Stat Assoc. 1970;65:1509–1526. doi: 10.1080/01621459.1970.10481180.

- LB** Ljung GM, Box GEP, "On a measure of lack of fit in time series models." *Biometrika*. 1978;65:297–303. doi: 10.1093/biomet/65.2.297.
- M** Mahdi E, "Portmanteau test statistics for seasonal serial correlation in time series models", Springerplus. 2016;5(1):1485. Published 2016 Sep 5. doi:10.1186/s40064-016-3167-4, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5011475/>.
- ShS** Shumway R.H., Stoffer, D.S., *Time Series and Its Applications with R Examples*, Third edition, Springer Text in Statistics, Springer, 2010.
- T** Tsay R.S., *An introduction to Analysis of Financial Data with R*, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., Hoboken, New Jersey, 2013.
- Y** Yule GU, "Why do we Sometimes get Non-Sense Correlations between Time-Series?—A Study in Sampling and the Nature of Time-Series", *Institute of the Royal Statistical Society*, Vol. 89, No. 1 (January 1926), pp. 1-63