

# Análisis Estadístico con R

Series de Tiempo

*Víctor Morales-Oñate*

*21 de marzo de 2018*

## Contents

<b>Introducción</b>	<b>1</b>
<b>Rezagos y operadores en diferencia</b>	<b>2</b>
Operadores de rezagos . . . . .	2
<b>Descomposición de una serie de tiempo</b>	<b>8</b>
Suavizamiento: Holt-Winters . . . . .	14
<b>Modelos de series de tiempo</b>	<b>17</b>
Ruido blanco . . . . .	17
Serie estacionaria (en covarianza) . . . . .	18
<b>Procesos ARMA(p,q)</b>	<b>18</b>
<b>El modelo Autorregresivo AR(p)</b>	<b>19</b>
<b>Proceso de Medias Móviles (MA)</b>	<b>30</b>
<b>Proceso ARMA</b>	<b>35</b>
<b>Buscando el <i>mejor</i> modelo</b>	<b>37</b>
Test de Dickey Fuller . . . . .	49
<b>Cointegración</b>	<b>53</b>
Estacionariedad . . . . .	53
Regresión espúrea . . . . .	55
Estacionariedad . . . . .	57
Ejemplo 2 . . . . .	60
<b>Introducción a los modelos de vectores autorregresivos: VAR</b>	<b>62</b>
<b>Referencias</b>	<b>67</b>

---

## Introducción

Una serie de tiempo es una sucesión de variables aleatorias ordenadas de acuerdo a una unidad de tiempo,  $Y_1, \dots, Y_T$ .

¿Por qué usar series de tiempo?

- Pronósticos
- Entender el mecanismo de generación de datos (no visible al inicio de una investigación)

# Rezagos y operadores en diferencia

## Operadores de rezagos

Definición:

$$\Delta Y_{t-i} = Y_t - Y_{t-i}$$

Ejemplos:

$$\Delta Y_t = Y_t - Y_{t-1}$$

Caso general:

$$L^j Y_t = Y_{t-j}$$

Ejemplos:

$$L^1 Y_t = L Y_t = Y_{t-1}$$

$$L^2 Y_t = Y_{t-2}$$

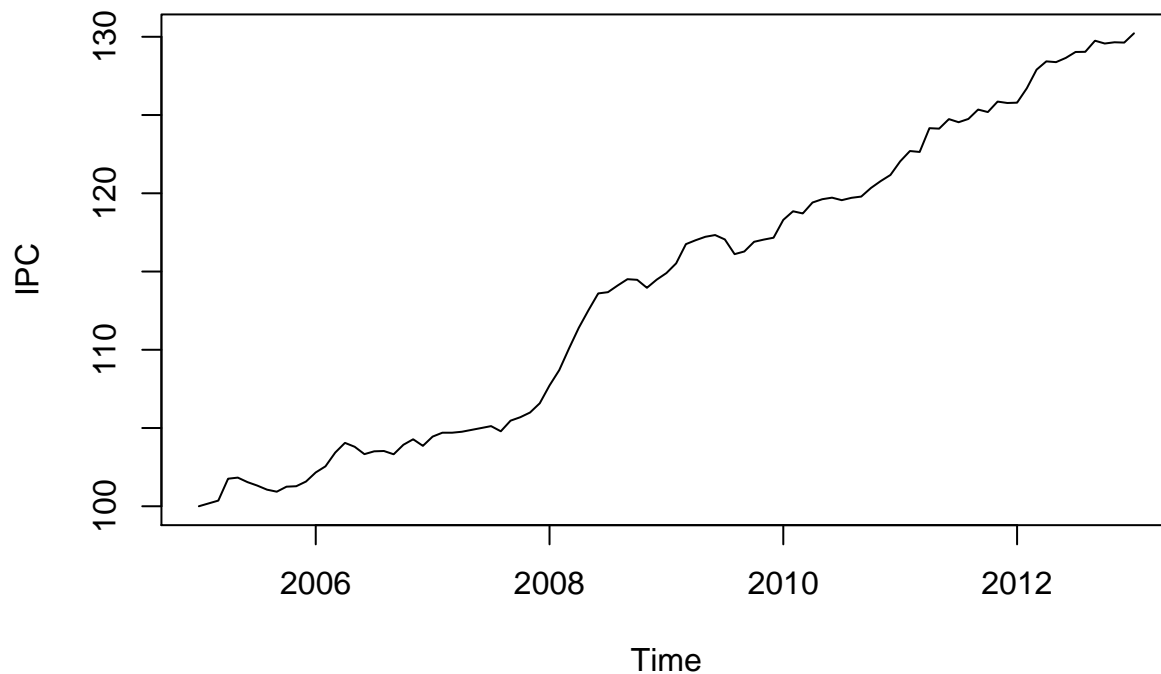
$$L^{-2} Y_t = Y_{t+2}$$

$$L^i L^j = L^{i+j} = Y_{t-(i+j)}$$

# Manipulando ts en 'R

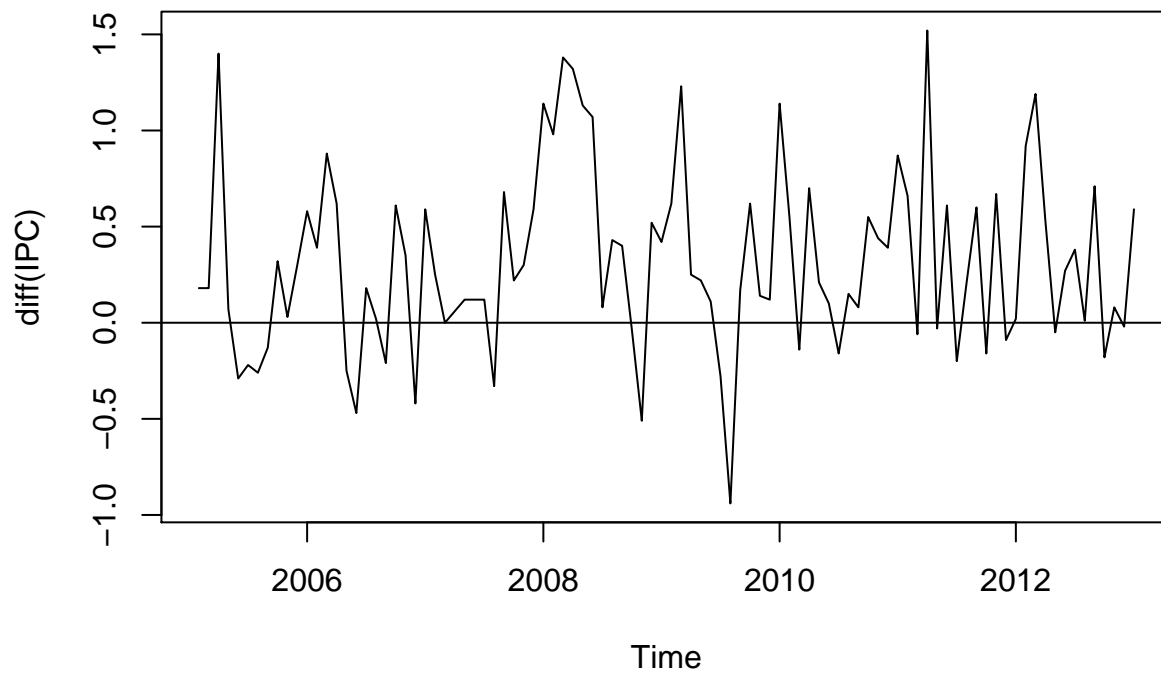
- Abrir IPCEcuador.csv
- Se puede ver una inflación variable

```
uu <- "https://raw.githubusercontent.com/vmoprojs/DataLectures/master/IPCEcuador.csv"
datos <- read.csv(url(uu),header=T,dec=".",sep=",")
IPC <- ts(datos$IPC,start=c(2005,1),freq=12)
plot(IPC)
```



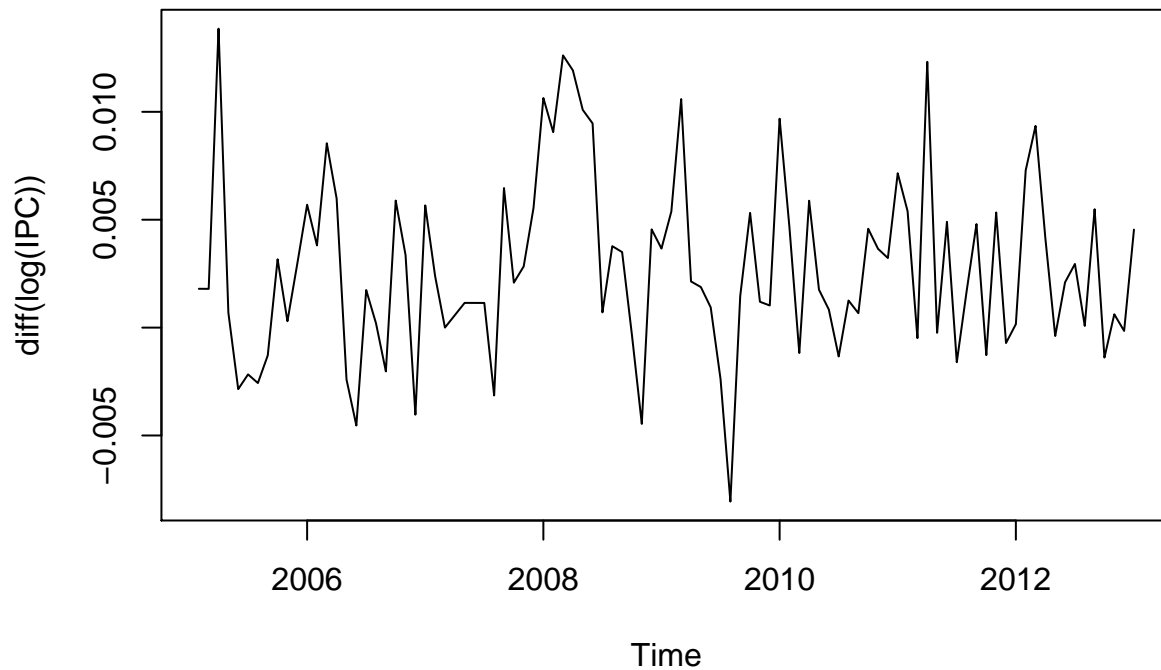
La serie tiene tendencia creciente. Tratemos de quitar esa tendencia:

```
plot(diff(IPC)) # Se puede ver una inflacion estable
abline(h=0)
```



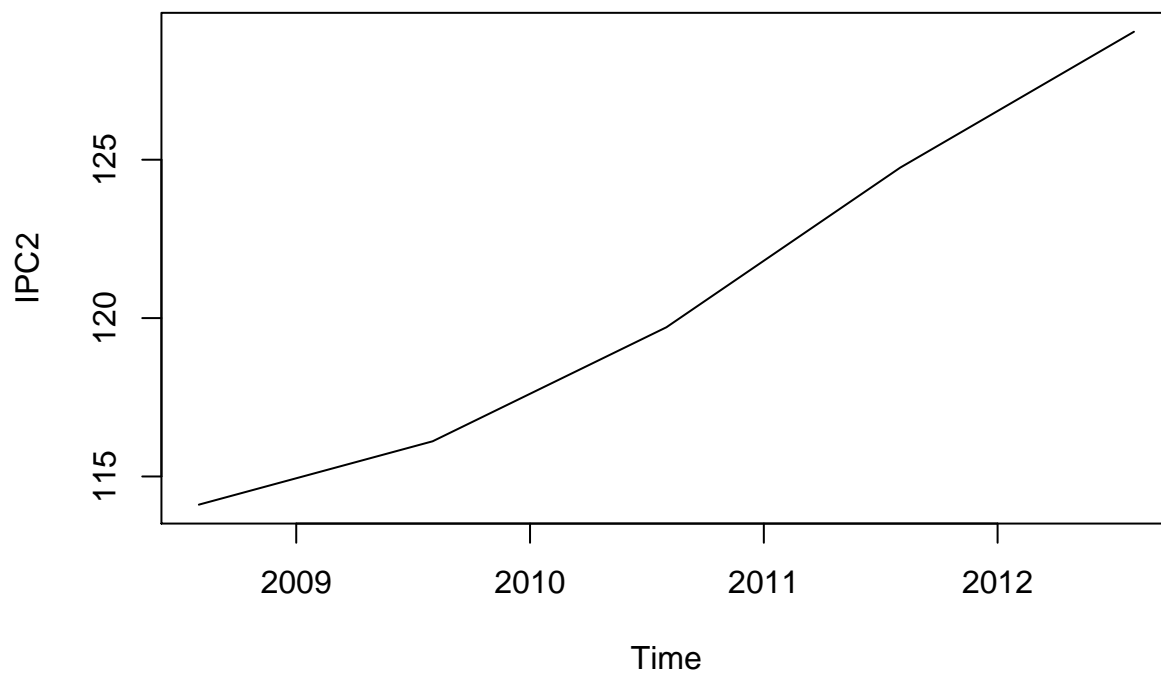
Se ha estabilizado, pero podemos hacerlo aún más con el logaritmo de la diferencia:

```
plot(diff(log(IPC))) #Tasa de variacion del IPC
```

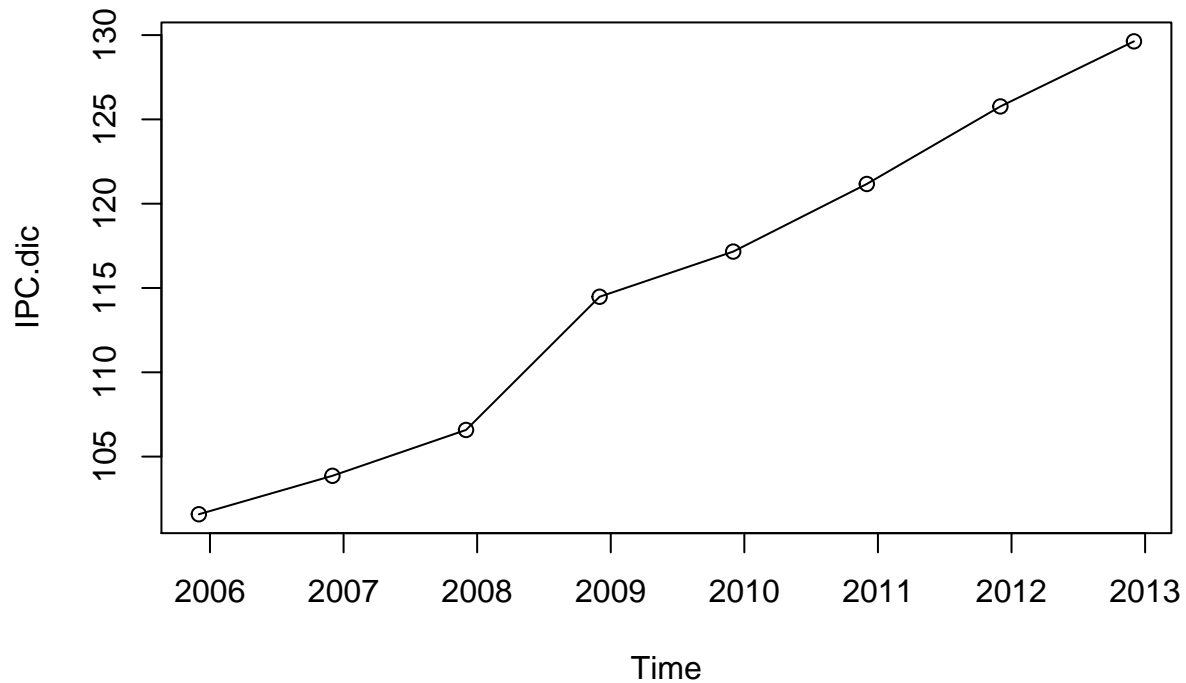


La serie no tiene tendencia y es estable. Ahora, si deseo trabajar con un subconjunto de datos, puedo...

```
# Solo quiero trabajar con los datos de agosto 2008
IPC2 <- window(IPC,start=c(2008,8),freq=1)
plot(IPC2)
```



```
# IPC de todos los diciembres
IPC.dic <- window(IPC,start=c(2005,12),freq=T)
plot(IPC.dic)
points(IPC.dic)
```



Si tengo mensuales y necesito trabajar con el IPC anual:

```
aggregate(IPC)
```

```
## Time Series:
## Start = 2005
## End = 2012
## Frequency = 1
## [1] 1213.09 1241.75 1262.13 1349.24 1399.26 1435.96 1491.87 1542.53
```

A continuación algunas transformaciones frecuentes y su interpretación:

Transformación	Interpretación
$z_t = \nabla y_t = y_t - y_{t-1}$	Cambio en $y_t$ . Es un indicador de crecimiento absoluto.
$z_t = \ln(y_t) - \ln(y_{t-1}) \approx \frac{y_t - y_{t-1}}{y_{t-1}}$	Es la tasa logarítmica de variación de una variable. Es un indicador de crecimiento relativo. Si se multiplica por 100 es la tasa de crecimiento porcentual de la variable
$z_t = \nabla [\ln(y_t) - \ln(y_{t-1})]$	Es el cambio en la tasa logarítmica de variación de una variable. Es un indicador de la aceleración de la tasa de crecimiento relativo de una variable.

## Ejemplo

Veamos un gráfico más interesante usando un conjunto de datos anterior, vamos a:

- Abrir la base `estadisticas_Turismo.csv`
- Agregar de manera mensual
- Convertir a `ts` y graficar

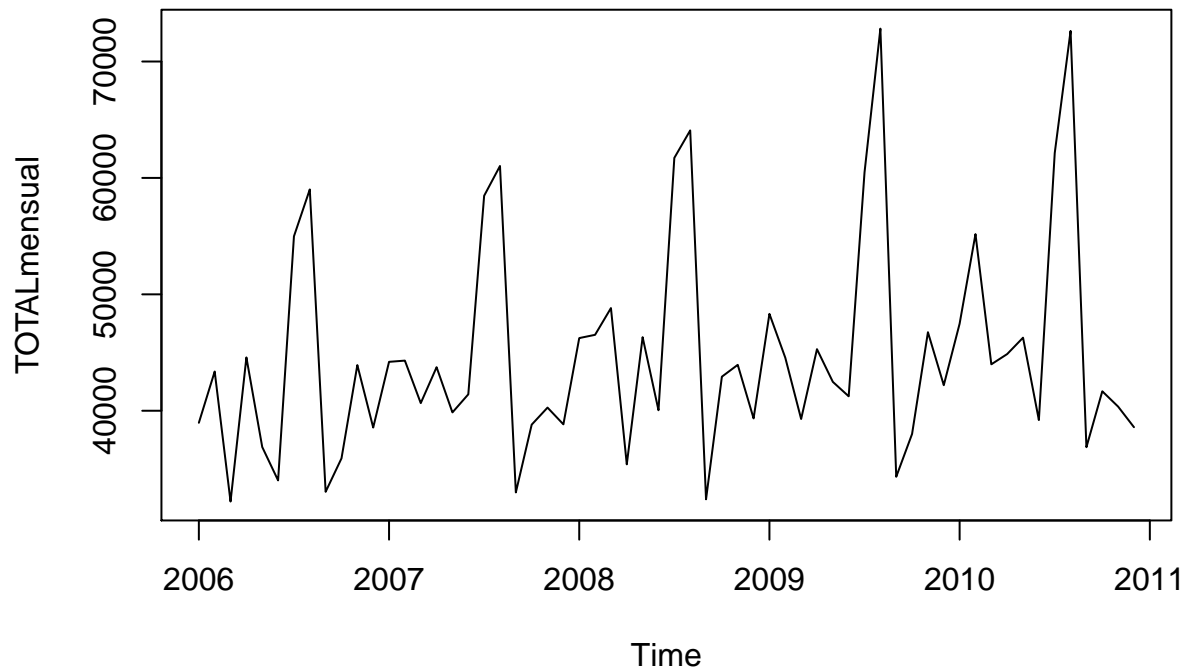
```
uu <- "https://raw.githubusercontent.com/vmoprojs/DataLectures/master/estadisticas%20Turismo.csv"
datos<-read.csv(url(uu),header=T,dec=".",sep=";")
attach(datos)
```

```
# Visitas a Areas Naturales Protegidas

# Sumar por mes y año

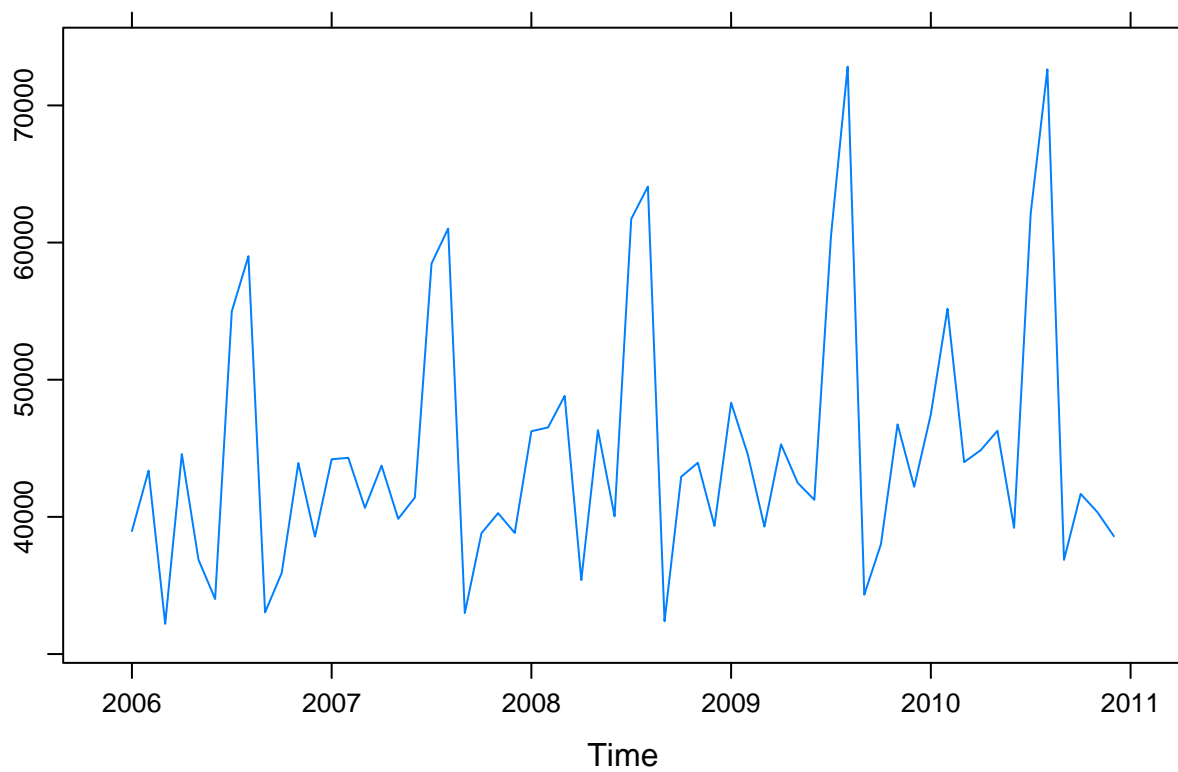
mensual<-aggregate(TOTALMENSUAL,by=list(mesnum,Year),FUN="sum") # Los datos sin mes es el total de ese año

TOTALmensual<-ts(mensual[,3],start=c(2006,1),freq=12)
plot(TOTALmensual)
```



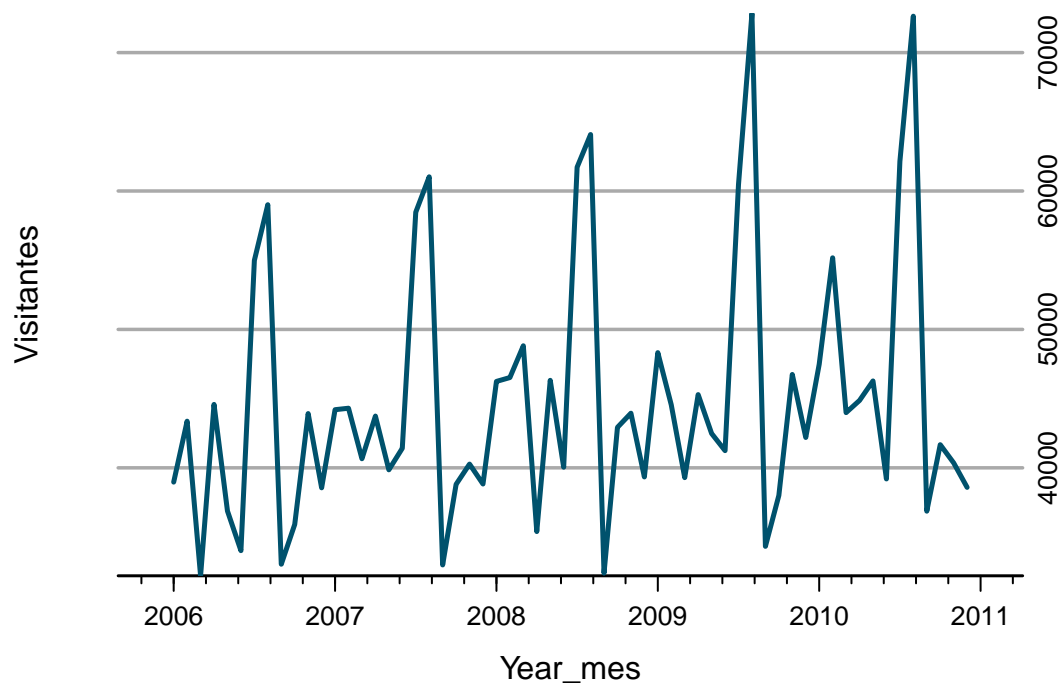
Se ve una tendencia creciente y también una cierta estacionalidad. Veamos la misma serie en gráficos más atractivos:

```
library(latticeExtra)
library(RColorBrewer)
library(lattice)
xyplot(TOTALmensual)
```



```
asTheEconomist(xyplot(TOTALmensual,
                      main="TOTAL VISITAS MENSALES \n AREAS PROTEGIDAS")
               ,xlab="Year_mes",ylab="Visitantes")
```

TOTAL VISITAS MENSALES  
AREAS PROTEGIDAS



# Descomposición de una serie de tiempo

## Componentes

- Tendencia-ciclo: representa los cambios de largo plazo en el nivel de la serie de tiempo
- Estacionalidad: Caracteriza fluctuaciones periódicas de longitud constante causadas por factores tales como temperatura, estación del año, periodo vacacional, políticas, etc.

$$Y_t = f(S_t, T_t, E_t)$$

donde  $Y_t$  es la serie observada,  $S_t$  es el componente estacional,  $T_t$  es la tendencia y  $E_t$  es el término de error.

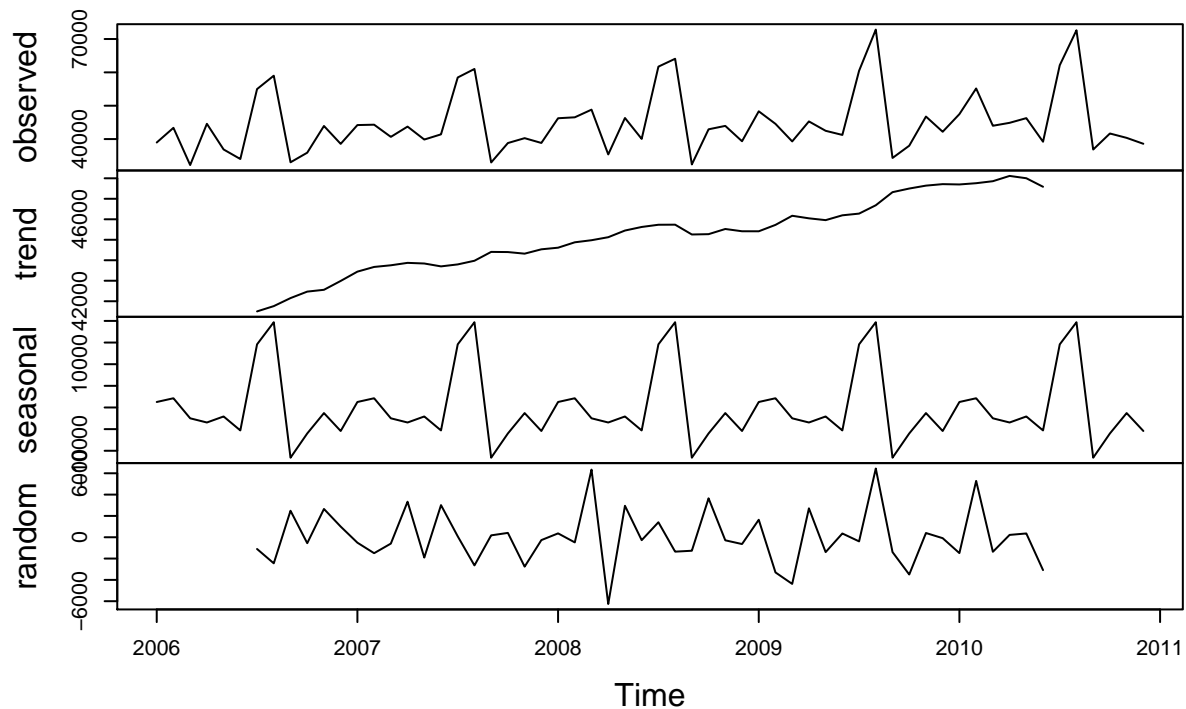
La forma de  $f$  en la ecuación anterior determina tipos de descomposiciones:

Descomposición	Expresión
Aditiva	$Y_t = S_t + T_t + E_t$
Multiplicativa	$Y_t = S_t * T_t * E_t$
Transformación logarítmica	$\log(Y_t) = \log(S_t) + \log(T_t) + \log(E_t)$
Ajuste estacional	$Y_t - S_t = T_t + E_t$

## Ejemplo

```
visitas.descompuesta<-decompose(TOTALmensual, type="additive")  
plot(visitas.descompuesta)
```

### Decomposition of additive time series



Dentro de `visitas.descompuesta` tenemos los siguientes elementos:

- `$x` = serie original



- `$seasonal` = componente estacional de los datos EJ: en marzo hay un decremento de 2502 (para cada dato)
- `$trend` = tendencia
- `$random` = visitas no explicadas por la tendencia o la estacionalidad
- `$figure` = estacionalidad (mismo que `seasonal` pero sin repetición)

## Descomposición: ¿aditiva o multiplicativa?

Visualmente:

- Aditivo:
  - las fluctuaciones estacionales lucen aproximadamente constantes en tamaño con el tiempo y
  - no parecen depender del nivel de la serie temporal,
  - y las fluctuaciones aleatorias también parecen ser más o menos constantes en tamaño a lo largo del tiempo
- Multiplicativo
  - Si el efecto estacional tiende a aumentar a medida que aumenta la tendencia
  - la varianza de la serie original y la tendencia aumentan con el tiempo

Forma alternativa de elegir: ver cuál es la que tiene un componente aleatorio menor.

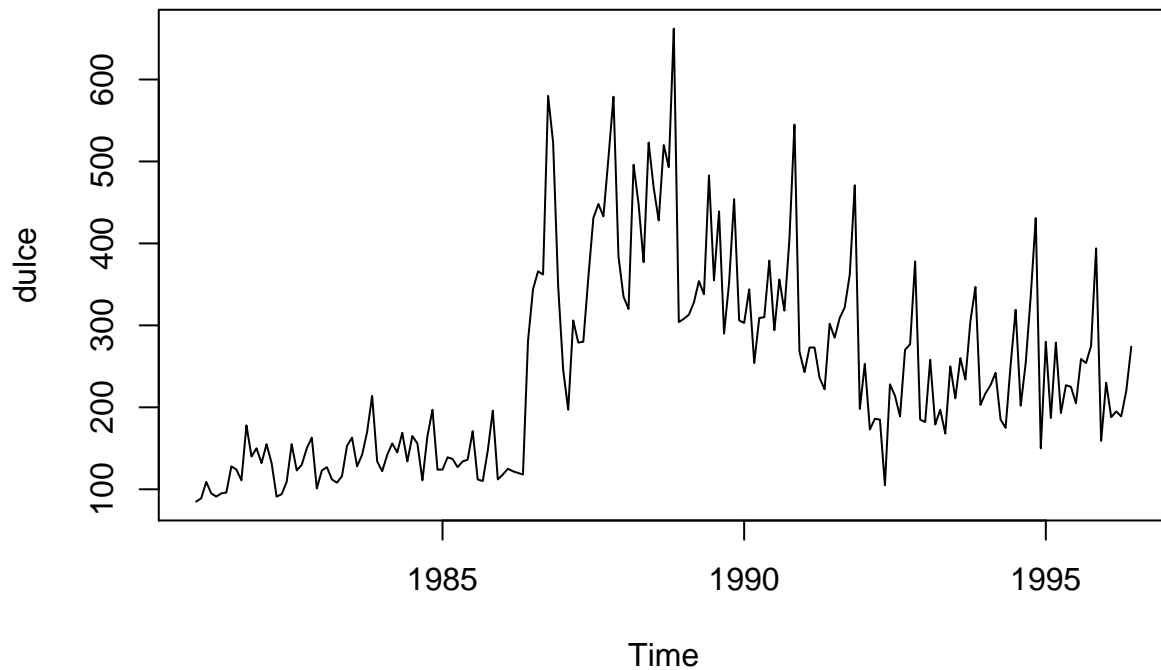
## Ejemplo:

Los datos en el archivo `wine.dat` son ventas mensuales de vino australiano por categoría, en miles de litros, desde enero de 1980 hasta julio de 1995. Las categorías son blanco fortificado (`fortw`), blanco seco (`dryw`), blanco dulce (`sweetw`), rojo (`red`), rosa (`rose`) y espumoso (`spark`).

```
direccion<- "https://raw.githubusercontent.com/dallascard/Introductory_Time_Series_with_R_datasets/master/wine.dat"
wine<-read.csv(direccion,header=T,sep="")
attach(wine)
head(wine)
```

```
##   winet fortw dryw sweetw  red rose spark
## 1     1  2585 1954     85  464  112 1686
## 2     2  3368 2302     89  675  118 1591
## 3     3  3210 3054    109  703  129 2304
## 4     4  3111 2414     95  887   99 1712
## 5     5  3756 2226     91 1139  116 1471
## 6     6  4216 2725     95 1077  168 1377
```

```
dulce <- ts(sweetw,start=c(1980,12), freq=12)
plot(dulce)
```

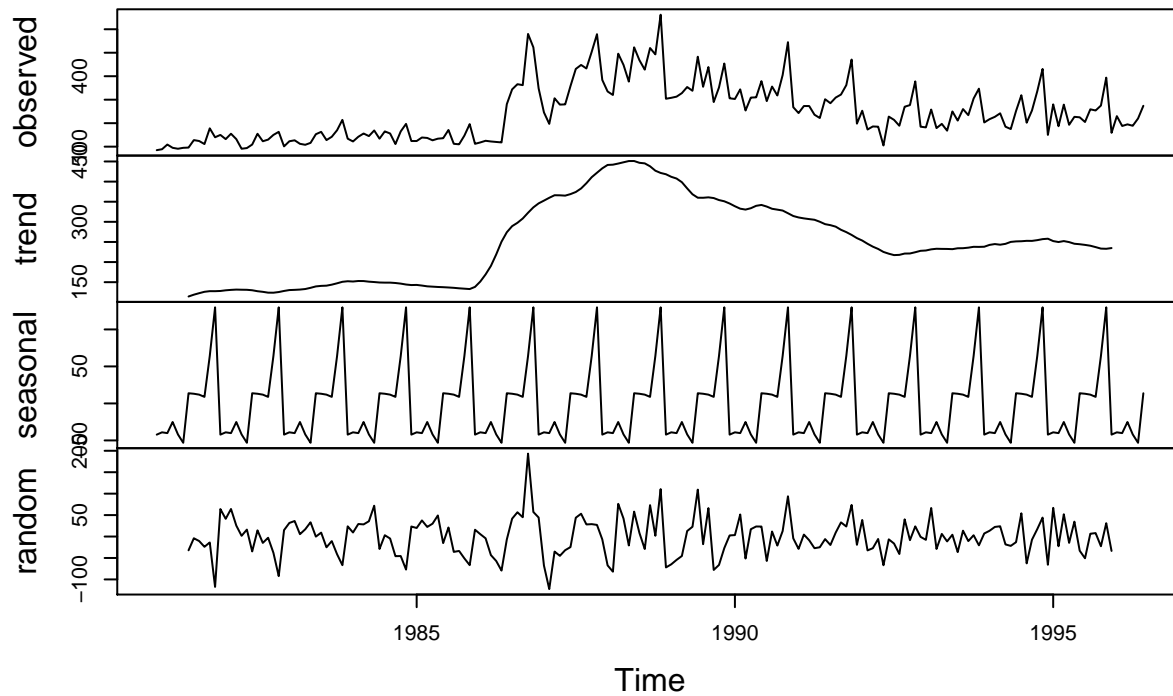


Tratemos la serie como un caso aditivo:

En funcion del grafico de la variable, se decide el “type” de la descomposicion La estacionalidad tiene valores negativos porque se plantea respecto de la tendencia

```
dulce.descompuesta<-decompose(dulce, type="additive")
plot(dulce.descompuesta)
```

### Decomposition of additive time series



```
a<-dulce.descompuesta$trend[27] # La tendencia era de 130 en mayo del 82
b<-dulce.descompuesta$seasonal[27] # El componente de la estacionalidad era este
c<-dulce.descompuesta$random[27] # Es el componente aleatorio
```

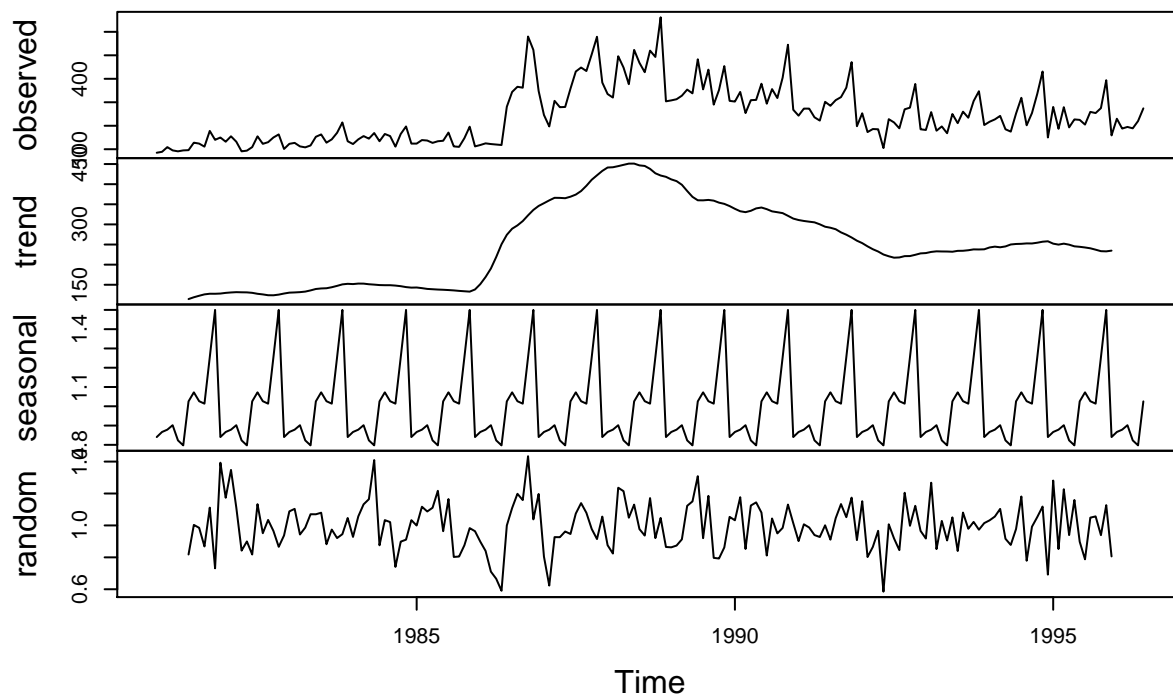
```
a+b+c # La sumatoria de la descomposicion de la serie da el valor real, si es aditiva
```

```
## [1] 127
```

Veamos el caso multiplicativo:

```
# Multiplicativa
dulce.descompuesta1<-decompose(dulce, type="multiplicative")
plot(dulce.descompuesta1)
```

## Decomposition of multiplicative time series



```
a<-dulce.descompuesta1$trend[27] # La tendencia era de 130 en mayo del 82
b<-dulce.descompuesta1$seasonal[27] # El componente de la estacionalidad era este
c<-dulce.descompuesta1$random[27] # Es el componente aleatorio
```

```
a*b*c # La sumatoria de la descomposicion de la serie da el valor real, si es aditiva
```

```
## [1] 127
```

Veamos la forma alternativa de elección:

```
u1<-var(dulce.descompuesta1$random,na.rm=T)
u2<-var(dulce.descompuesta1$seasonal,na.rm=T)
cbind(u1,u2)
```

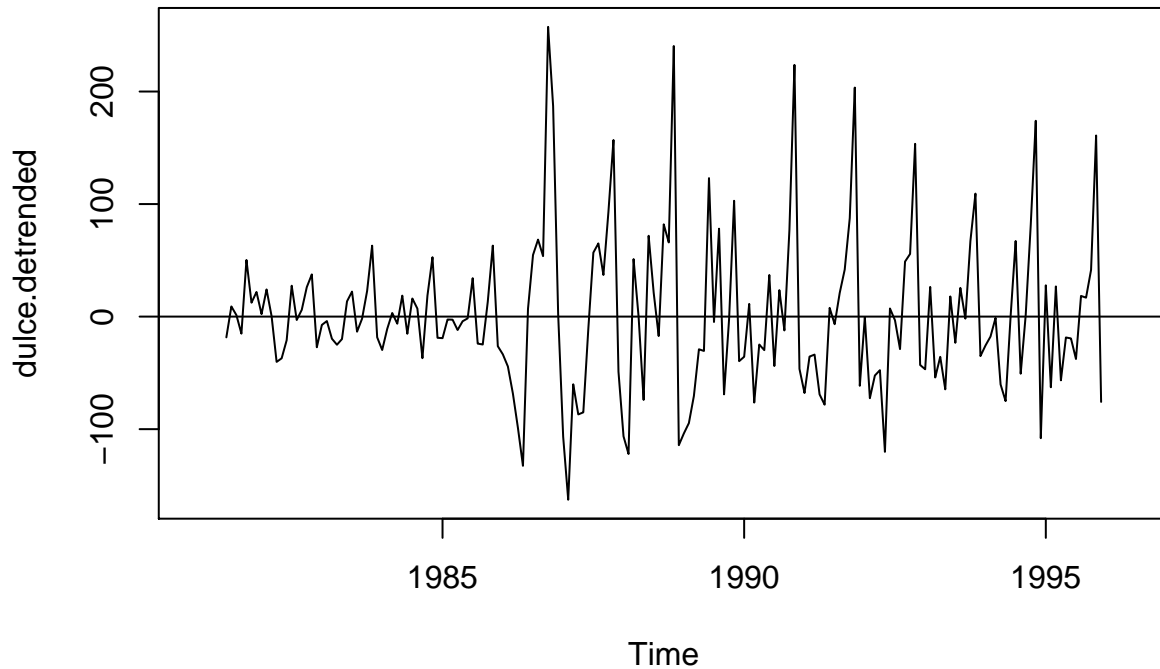
```
##           u1           u2
## [1,] 1970.235 0.02247602
```

Se escoge la multiplicativa en este caso.

### Detrend:

Las series se ofrecen generalmente sin tendencia ni estacionalidad. Veamos la serie sin tendencia:

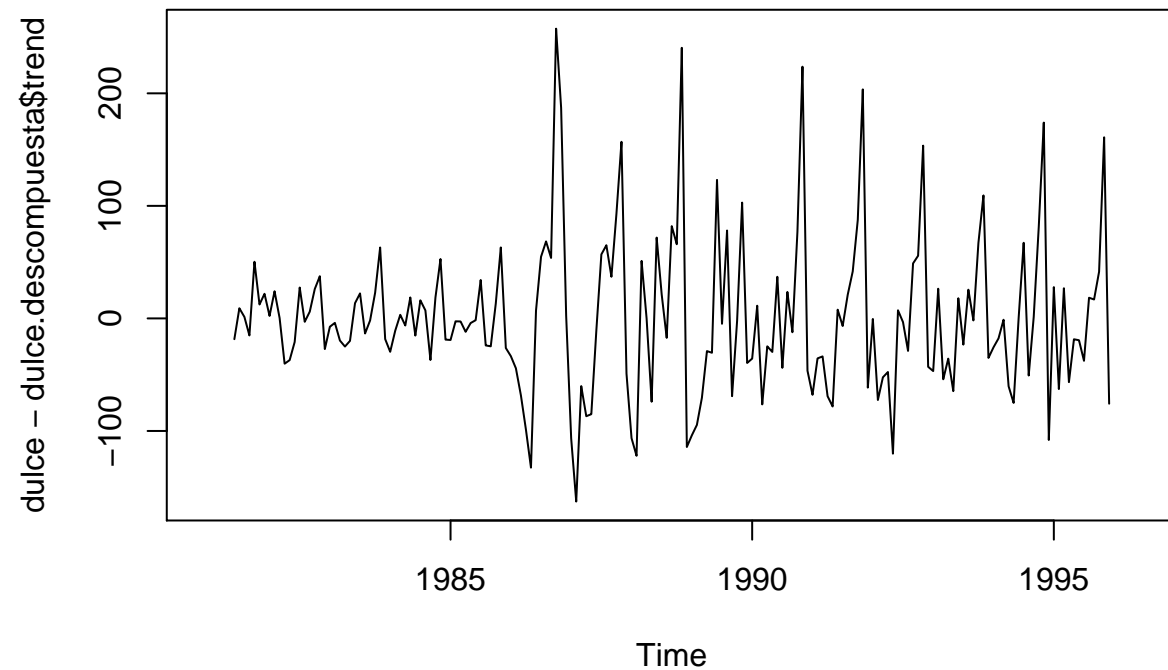
```
dulce.detrended <- dulce-dulce.descompuesta$trend  
plot(dulce.detrended)  
abline(h=0)
```



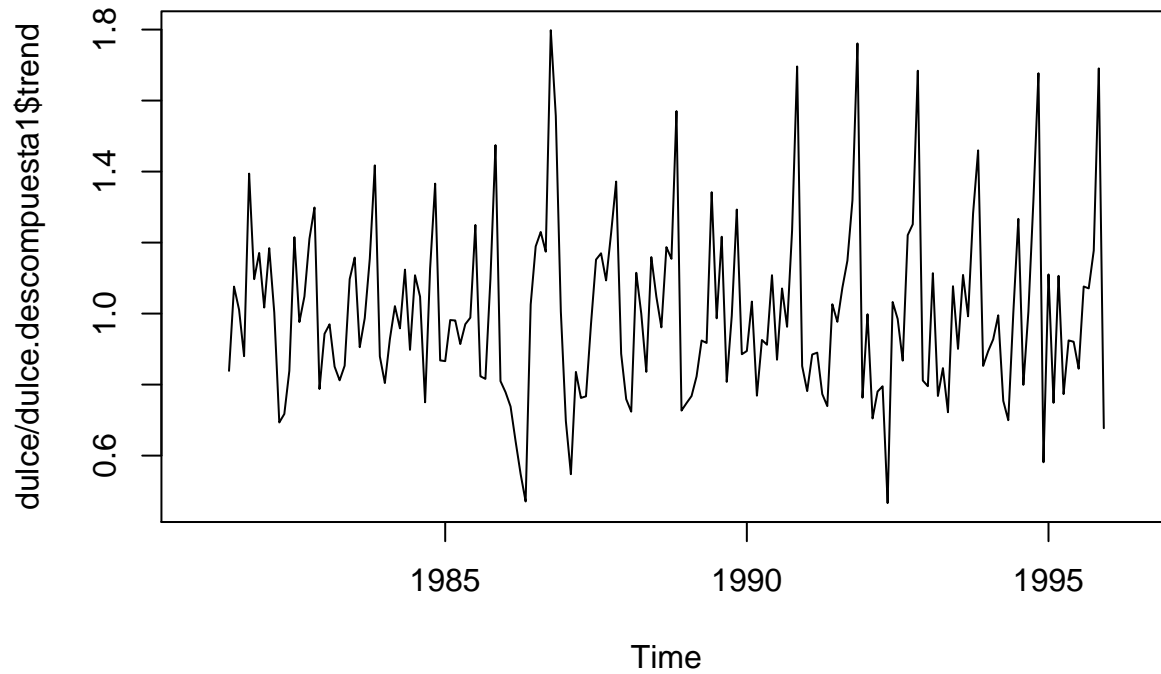
Parece ser que hay un cambio en la varianza desde el 85.

Si descomponemos multiplicativamente en vez de restar se debe dividir.

```
plot(dulce-dulce.descompuesta$trend)
```

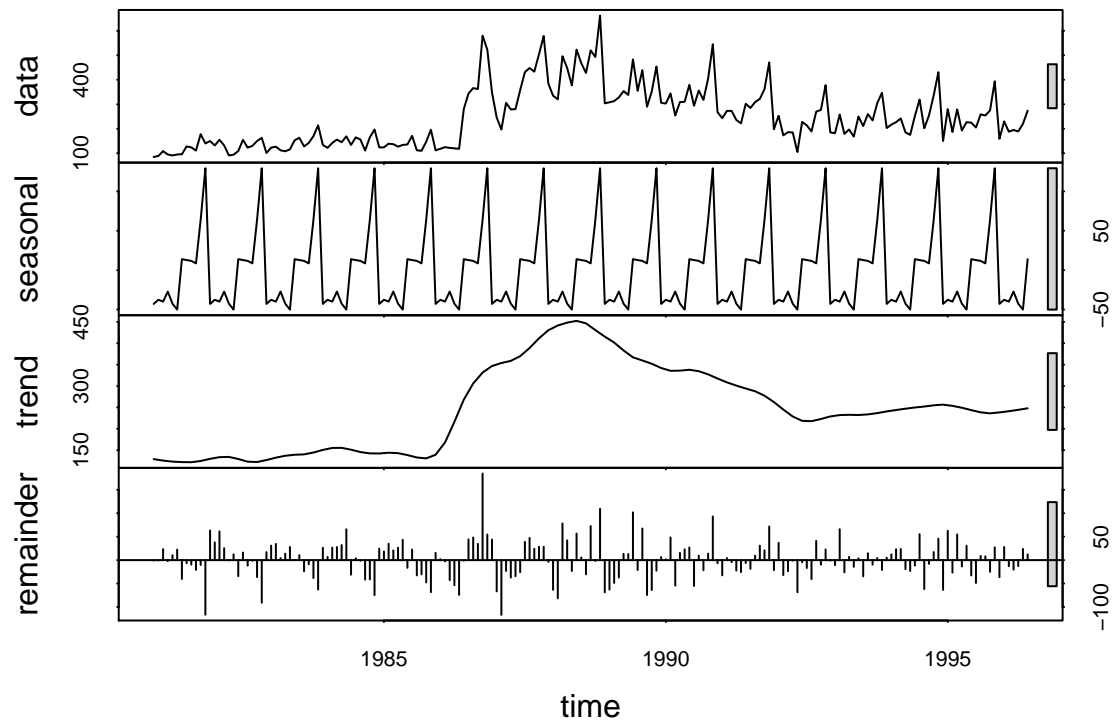


```
plot(dulce/dulce.descompuesta1$trend)
```



Existen formas de descomponer más sofisticadas, por ejemplo, usando la función `stl`.

```
dulce.stl<-stl(dulce,s.window="per")
plot(dulce.stl)
```



En este caso el calculo de la tendencia cambia, se calcula con formas no paramétricas. La barra del final es la desviacion estándar.

## Suavizamiento: Holt-Winters

El método se resume en las fórmulas siguientes:

$$\begin{aligned}a_t &= \alpha(x_t - s_{t-p}) + (1 - \alpha)(a_{t-1} + b_{t-1}) \\b_t &= \beta(a_t - a_{t-1}) + (1 - \beta)b_{t-1} \\s_t &= \gamma(x_t - a_t) + (1 - \gamma)s_{t-p}\end{aligned}$$

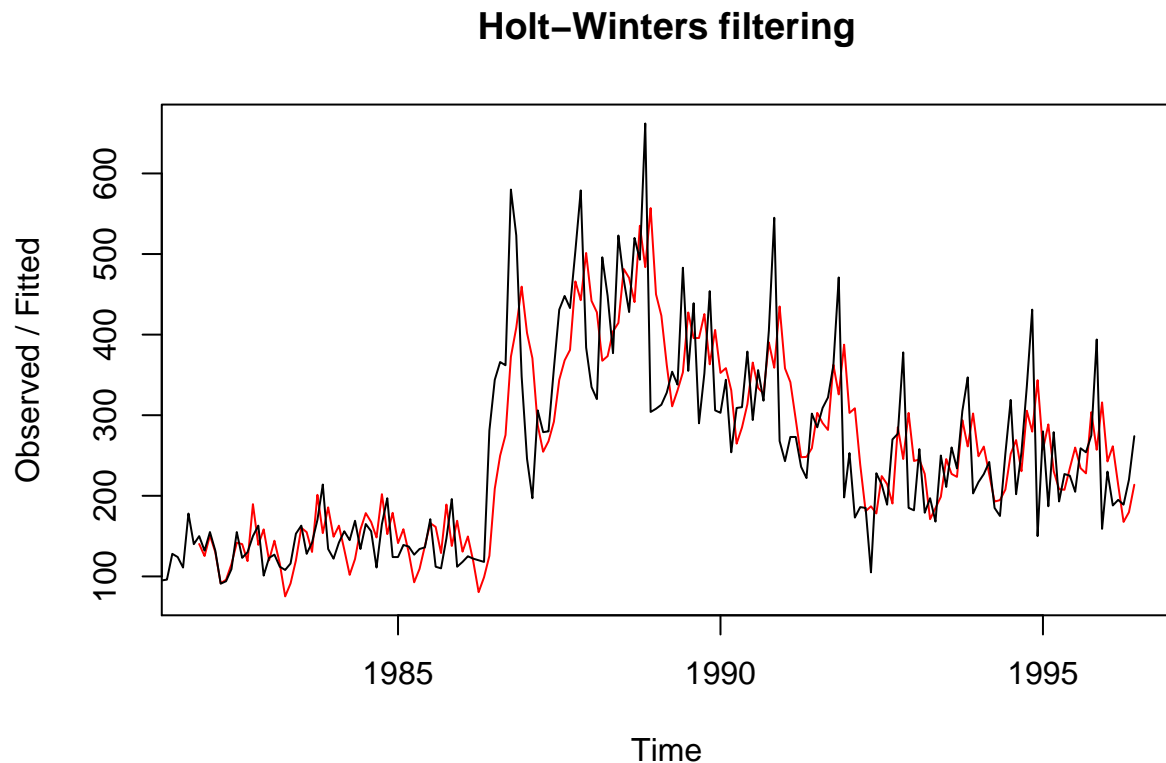
El método de Holt-Winters generaliza el método de suavizamiento exponencial.

### Ejemplo

Veamos un modelo más sencillo:

$$\begin{aligned}x_t &= \mu_t + w_t \\ \mu_t = a_t &= \alpha x_t + (1 - \alpha)a_{t-1}\end{aligned}$$

```
dulce.se <- HoltWinters(dulce,beta=0,gamma=0)
plot(dulce.se)
```

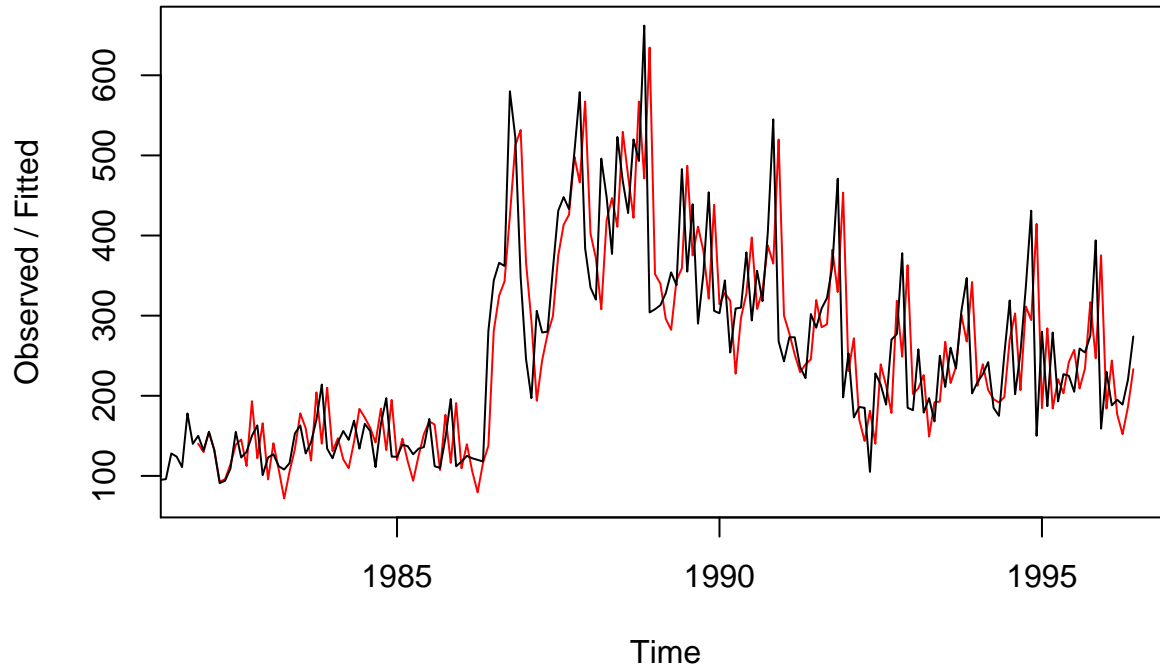


Es un suavizamiento HW sin tendencia y sin componente estacional. La serie roja son los datos con suavizamiento exponencial y la negra son los observados. R buscó el alpha que le pareció apropiado.

Usemos un alpha deliberado:

```
dulce.se1 <- HoltWinters(dulce,alpha=0.8,beta=0,gamma=0)
plot(dulce.se1)
```

## Holt-Winters filtering



¿Qué pasó con los errores?

```
dulce.se$SSE # Suma de los residuos al cuadrado (de un paso)
```

```
## [1] 963408.2
```

```
dulce.se1$SSE # Suma de los residuos al cuadrado (de un paso)
```

```
## [1] 1132577
```

Es decir, el criterio para la búsqueda de los parámetros es la minimización del SSE.

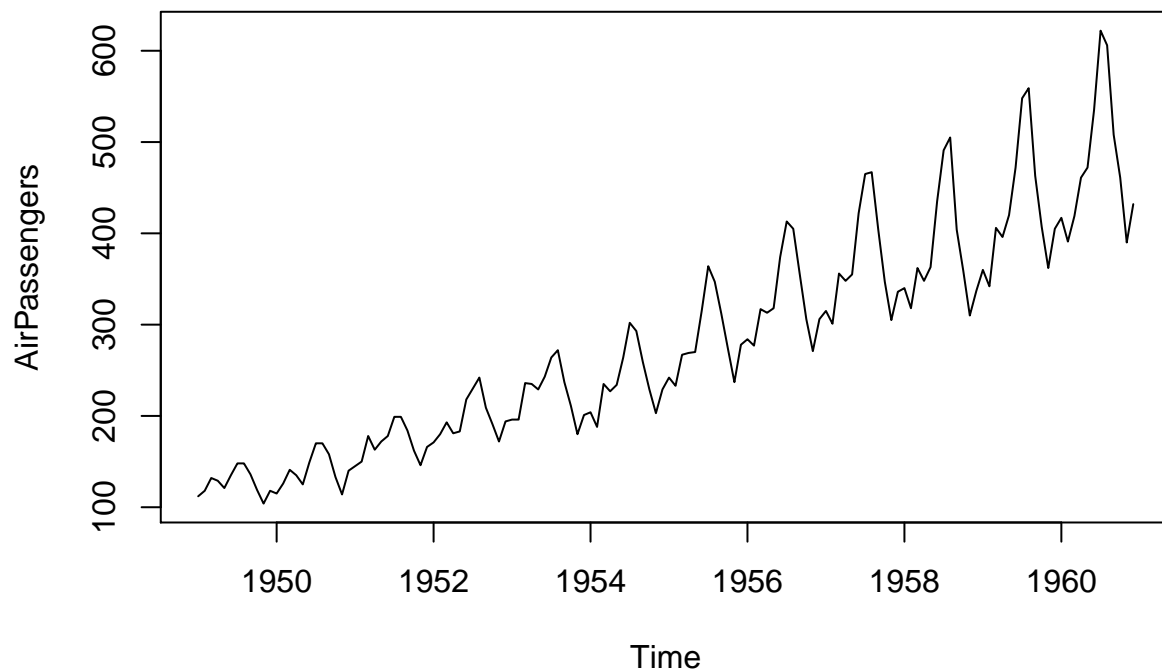
## Ejemplo

Total mensual de pasajeros (en miles) de líneas aéreas internacionales, de 1949 a 1960.

```
data(AirPassengers)
str(AirPassengers)
```

```
## Time-Series [1:144] from 1949 to 1961: 112 118 132 129 121 135 148 148 136 119 ...
```

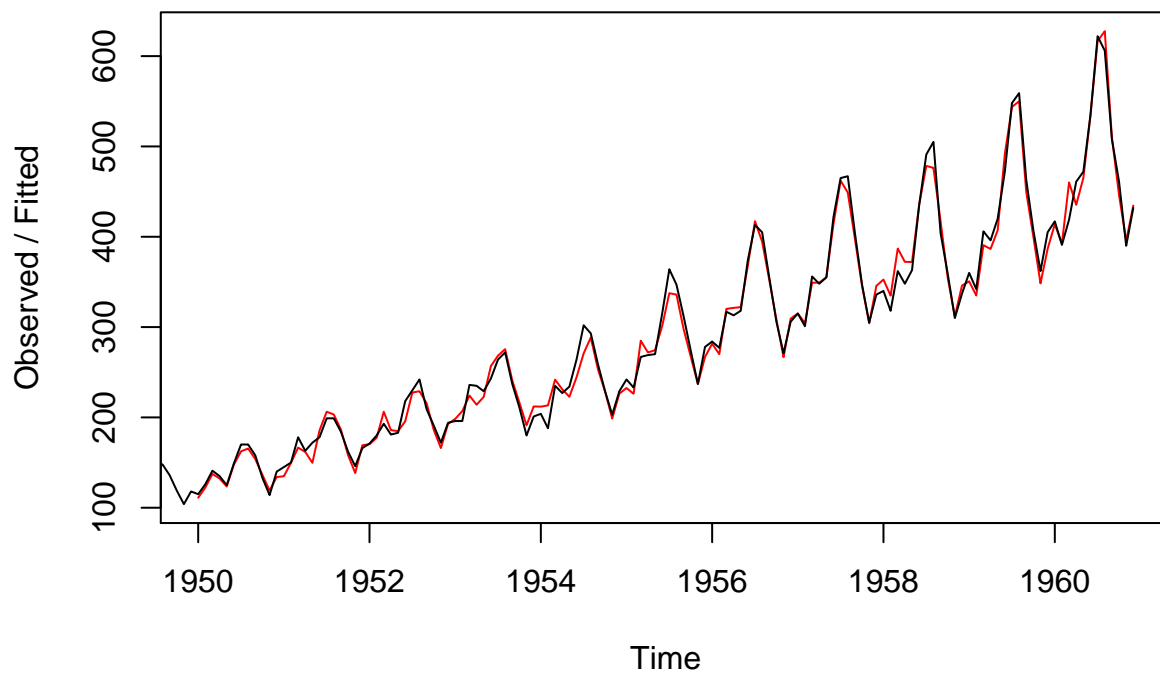
```
plot(AirPassengers)
```



Se aprecia tendencia y variabilidad. Podemos usar HW para predicción:

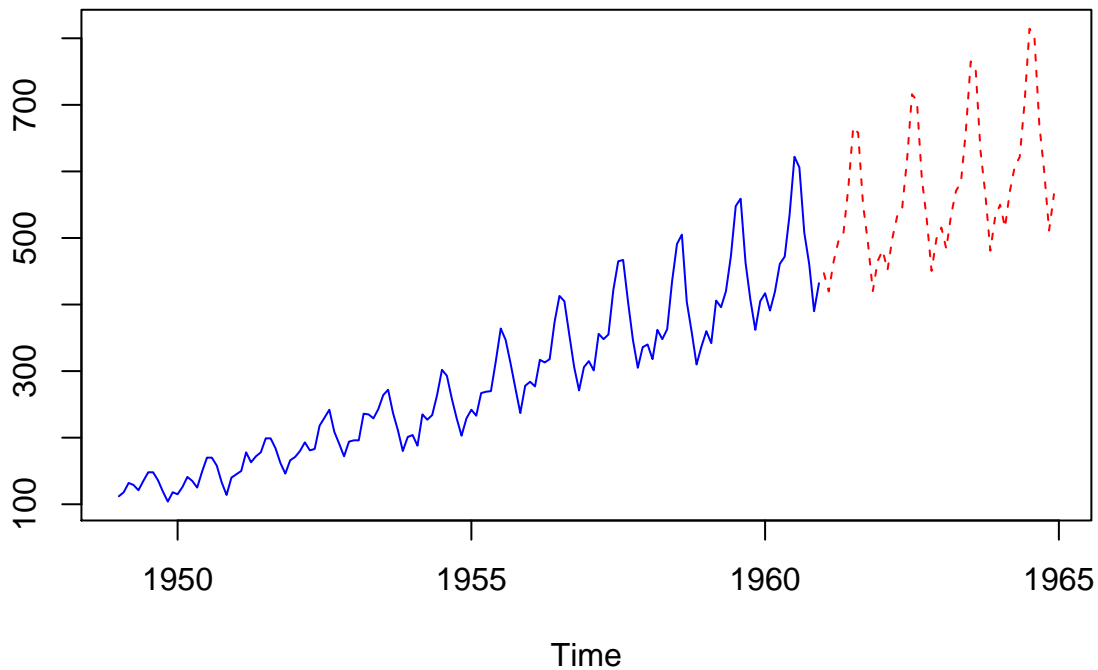
```
ap.hw<- HoltWinters(AirPassengers,seasonal="mult")
plot(ap.hw)
```

### Holt-Winters filtering



```
ap.prediccion <- predict(ap.hw,n.ahead=48)
ts.plot(AirPassengers,ap.prediccion,lty=1:2,
col=c("blue", "red"))
```





## Modelos de series de tiempo

### Ruido blanco

Una serie  $(\epsilon_t, t \in \mathbb{Z})$  se dice que es Ruido Blanco si cumple

- $E(\epsilon_t) = 0$  (media cero)
- $Var(\epsilon_t) = \sigma^2$  (varianza constante)
- $\forall k \neq 0, Cov(\epsilon_t, \epsilon_{t+k}) = 0$  (Incorrelación)

Si además cumple que  $\epsilon_t \sim N(0, \sigma^2)$  se dice que  $\epsilon_t$  es Ruido Blanco Gaussiano (RBG).

```
n <- 200
mu <- 0
sdt <- 3
w <- rnorm(n, mu, sdt)
```

¿Cómo se si algo tiene ruido blanco? : Analizo la función de autocorrelación muestral.

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0}$$

donde

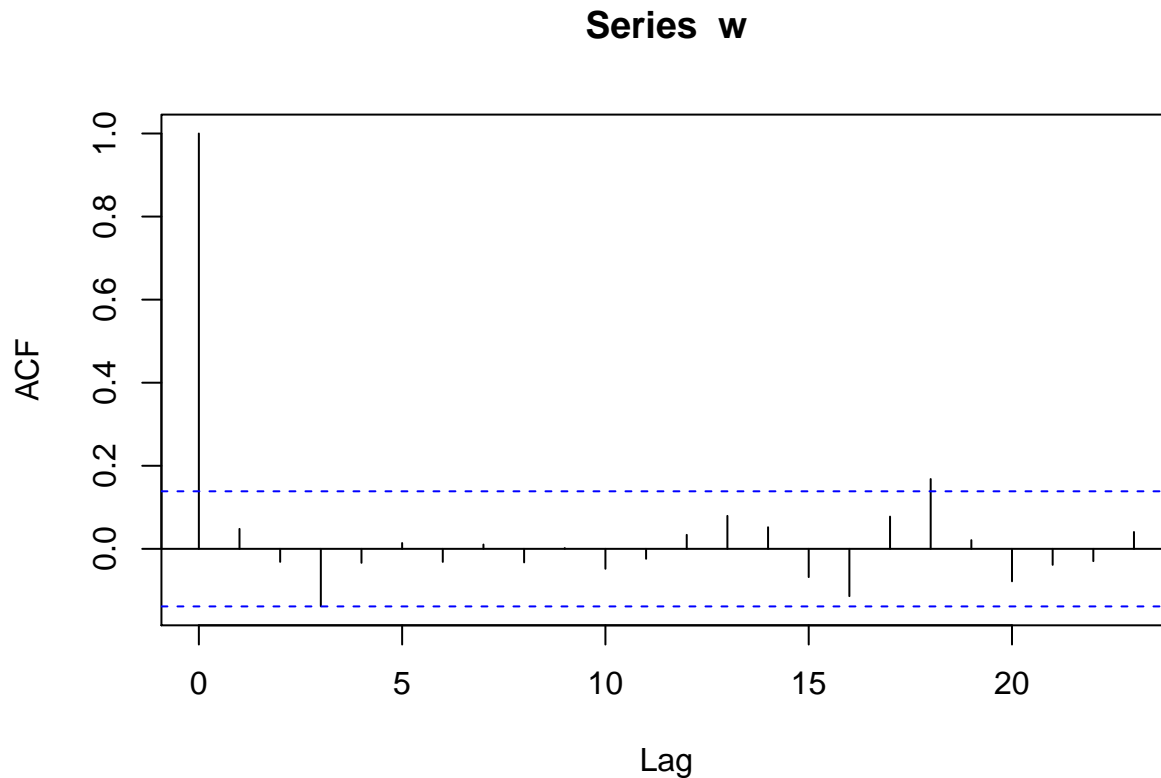
$$\hat{\gamma}_k = \frac{\sum (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{n}$$

$$\hat{\gamma}_0 = \frac{\sum (Y_t - \bar{Y})^2}{n}$$

Se asume que  $\hat{\rho}_k \sim N(0, 1/n)$ . Es decir:

$$\hat{\rho}_k = \frac{\sum_{t=k+1}^T (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^T (Y_t - \bar{Y})^2}$$

`acf(w)`



Si se sale de las franjas, si hay correlación y no hay ruido blanco

### Serie estacionaria (en covarianza)

Una serie  $(Y_t, t \in \mathbb{Z})$  se dice estacionaria en covarianza o simplemente estacionaria si cumple dos condiciones:

1.  $E(Y_t) = \mu$
2.  $Cov(Y_{t_1}, Y_{t_2}) = R(t_2 - t_1)$  con  $R$  función par ( $f(-x) = f(x)$ )

Es decir, la covarianza entre  $Y_{t_1}$  y  $Y_{t_2}$  depende únicamente de la distancia entre los tiempo  $t_2$  y  $t_1$ ,  $|t_2 - t_1|$ .

### Procesos ARMA(p,q)

En los modelos de descomposición  $Y_t = T_t + S_t + \epsilon_t$ ,  $t = 1, 2, \dots$  se estima  $\hat{\epsilon}_t$  y se determina si es o no ruido blanco mediante, por ejemplo, las pruebas LjungBox y DurbinWatson.

En caso de encontrar que  $\hat{\epsilon}_t$  no es ruido blanco, el siguiente paso es modelar esta componente mediante tres posibles modelos:

1. Medias Móviles de orden  $q$ ,  $MA(q)$ .
2. Autorregresivos de orden  $q$ ,  $AR(p)$ .
3. Medias Móviles Autorregresivos,  $ARMA(p, q)$ .

## El modelo Autorregresivo AR(p)

Se dice que  $Y_n$ ,  $n \in \mathbb{Z}$  sigue un proceso  $AR(p)$  de media cero si

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

donde  $\epsilon_t \sim RB(0, \sigma^2)$  y  $p = 1, 2, \dots$ . Usando el operador de rezago  $L$  se puede escribir como:

$$\phi_p(L)(Y_n) = \epsilon_n$$

con  $\phi_p(z) = 1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p$ , el polinomio autorregresivo.

### Condición Suficiente para que un $AR(p)$ sea Estacionario en Covarianza

La condición suficiente para que  $Y_t \sim AR(p)$  sea estacionario en covarianza es que las  $p$  raíces de la ecuación  $\phi_p(z) = 0$ ,  $z_i$ ,  $i = 1, 2, \dots, p$  cumplan

$$|z_i| > 1.$$

En palabras, la condición se describe como *para que un proceso autorregresivo de orden  $p$  sea estacionario en covarianza, es suficiente que las raíces del polinomio autorregresivo estén por fuera del círculo unitario*

Si el proceso  $Y_t$  es estacionario en covarianza se cumple que su media es constante,  $Y_t = \mu$

### Propiedades

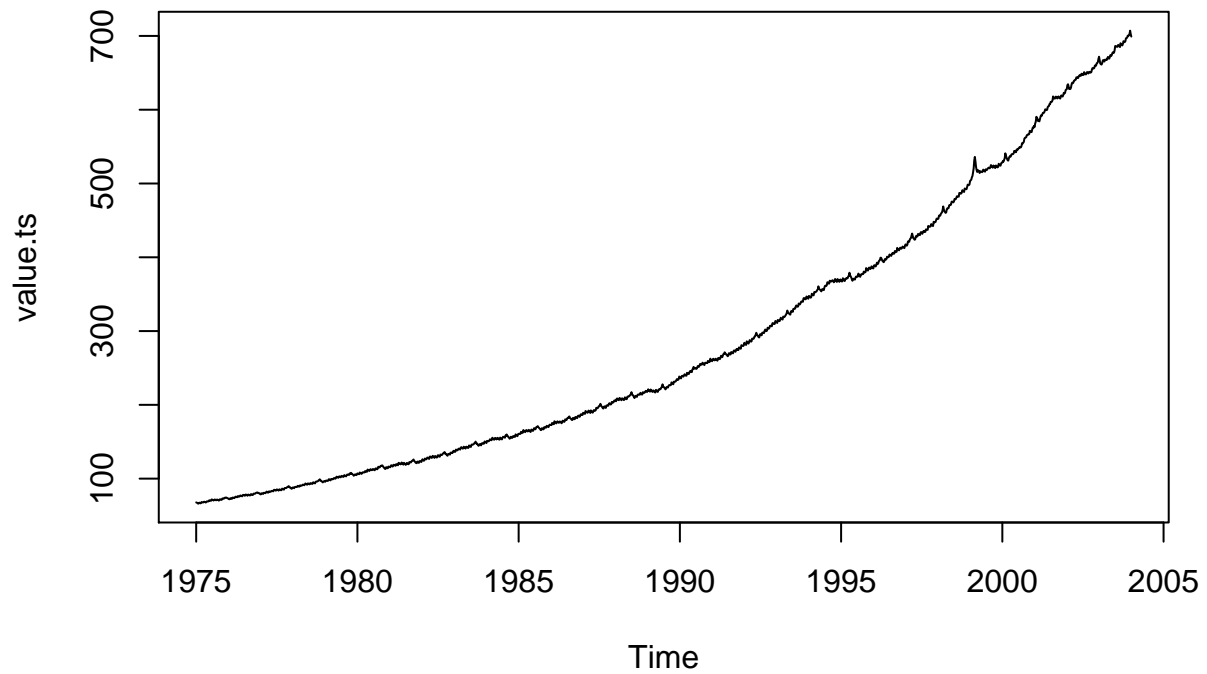
1.  $E(Y_t) = 0$
2.  $\sum_{j=1}^p \phi_j < 1$

Trabajaremos con datos de  $M1$  (WCURRNS dinero en circulación fuera de los Estados Unidos) semanales de los Estados Unidos desde enero de 1975.

```
uu <- "https://raw.githubusercontent.com/vmoprojs/DataLectures/master/WCURRNS.csv"
datos <- read.csv(url(uu), header=T, sep=";")
names(datos)

## [1] "DATE" "VALUE"

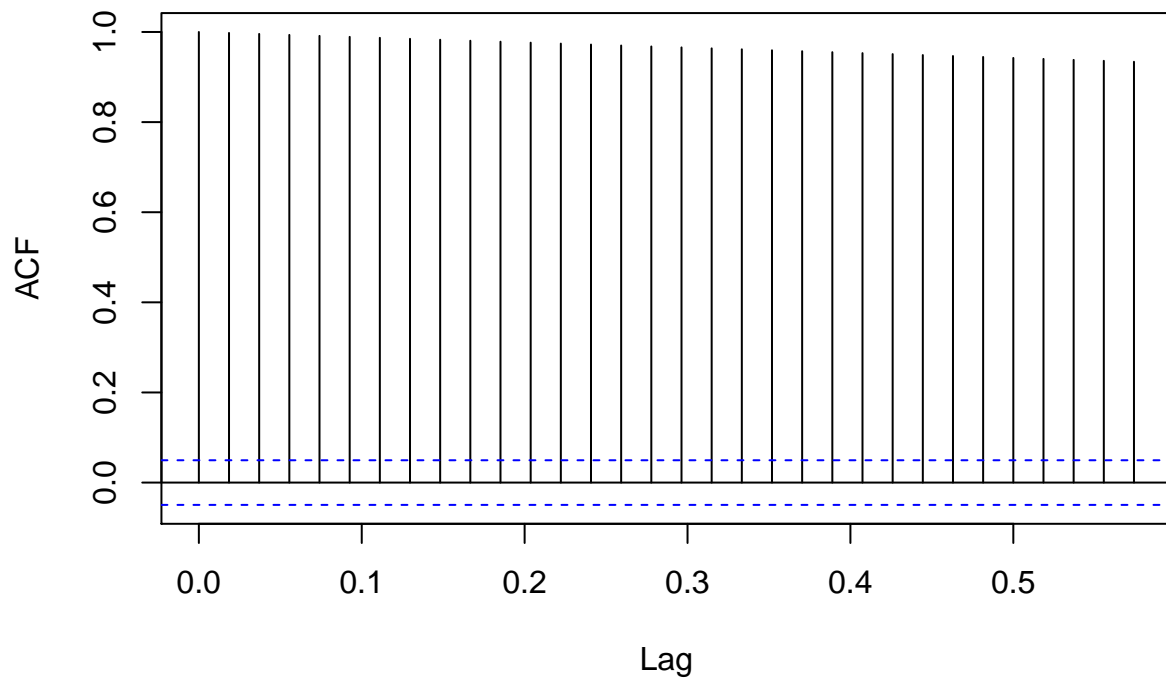
attach(datos)
value.ts <- ts(VALUE, start=c(1975,1), freq=54)
ts.plot(value.ts)
```



Estacionariedad: La serie es estacionaria si la varianza no cambia

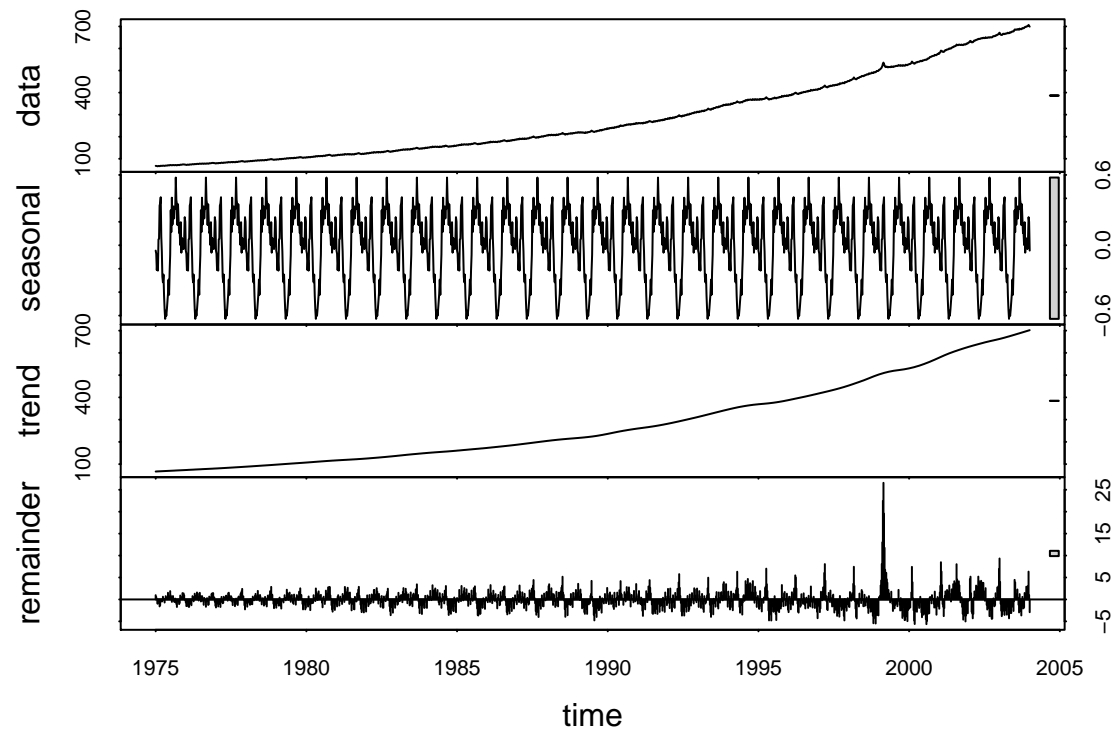
```
acf(value.ts)
```

### Series value.ts



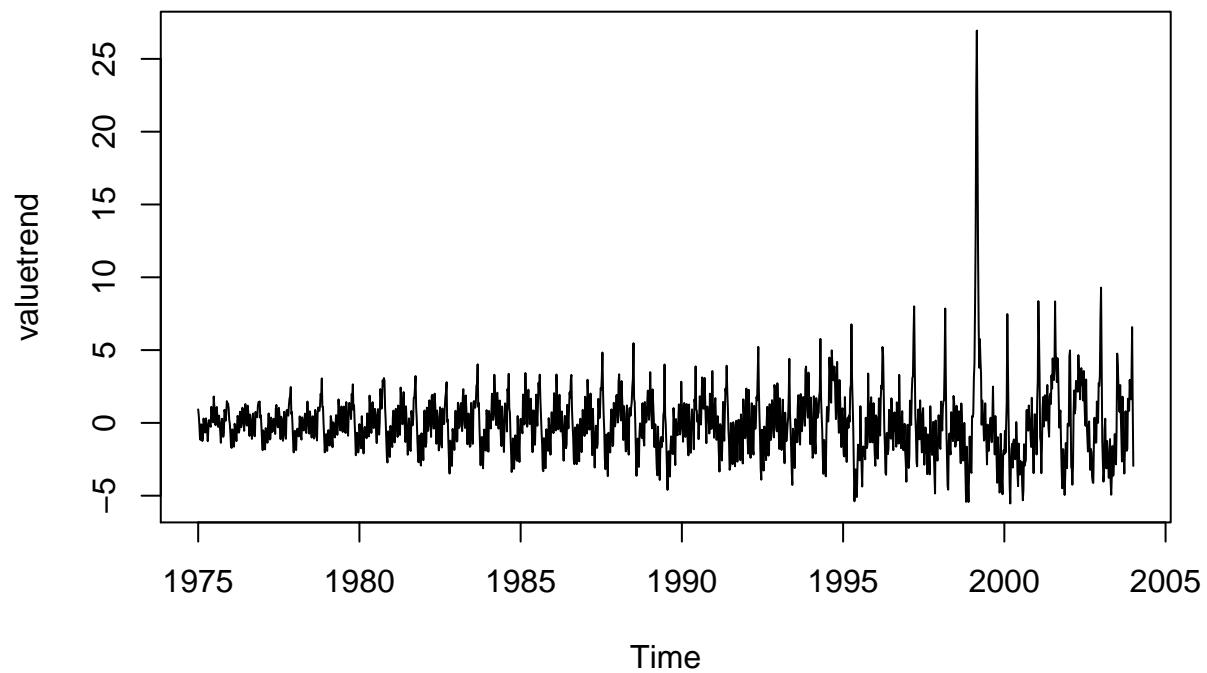
Esta es la marca de una serie que NO es estacionaria, dado que la autocorrelación decrece muy lentamente.

```
plot(stl(value.ts,s.window="per"))
```



Una forma de trabajar con una serie esacionaria es quitarle el *trend*

```
valuetrend<- value.ts- stl(value.ts,s.window="per")$time.series[,2]  
plot(valuetrend)
```



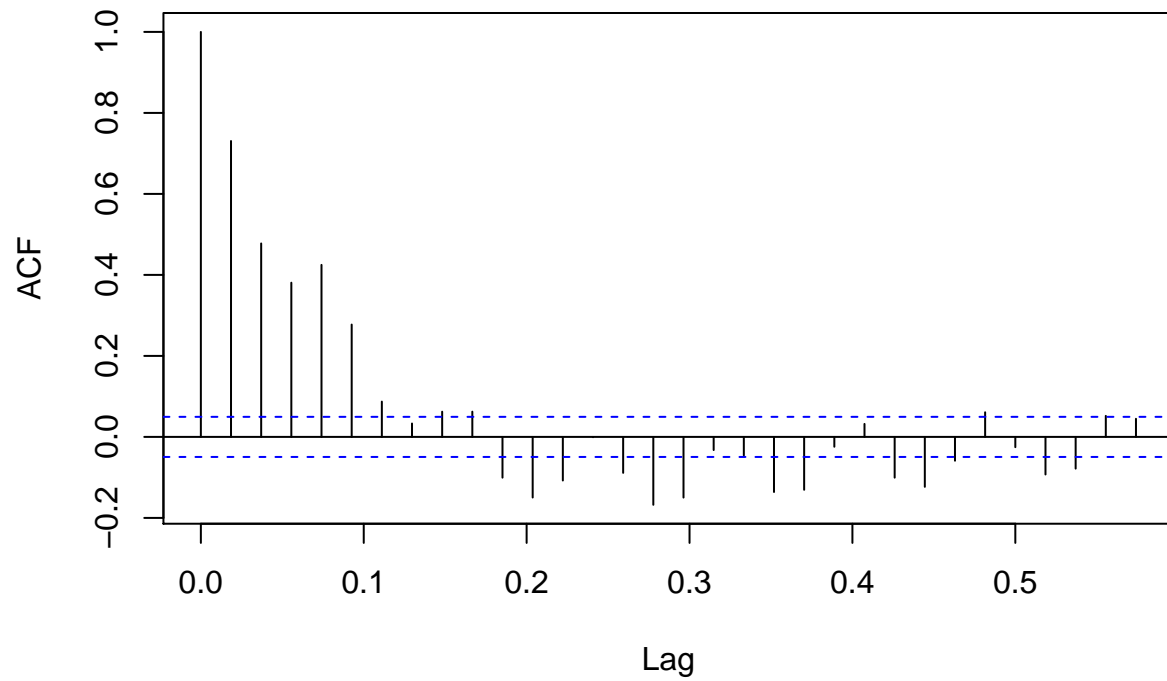
*Reminder* es lo que queda sin tendencia ni estacionalidad

```
valuereminder<-  
stl(value.ts,s.window="per")$time.series[,3]
```

Veamos cómo quedo la serie:

```
acf(valuereminder)
```

### Series valuereminder

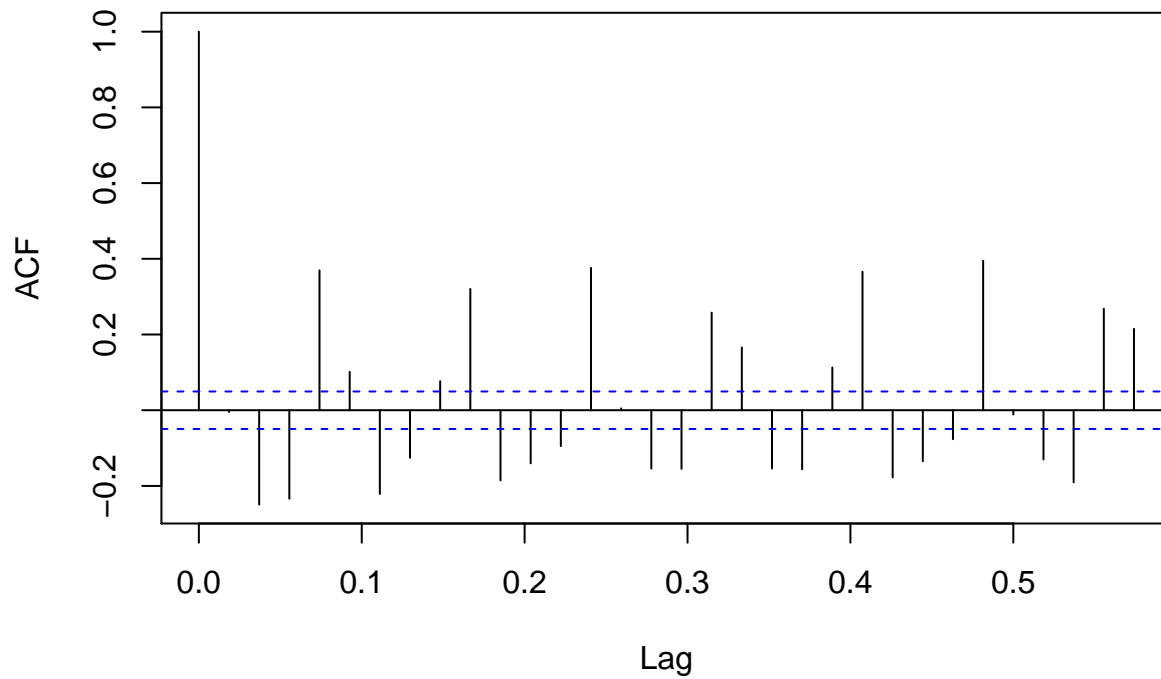


Se puede decir que la hicimos una serie estacionaria

Otra forma de hacer estacionaria una serie es trabajar con las diferencias

```
acf(diff(value.ts))
```

### Series diff(value.ts)

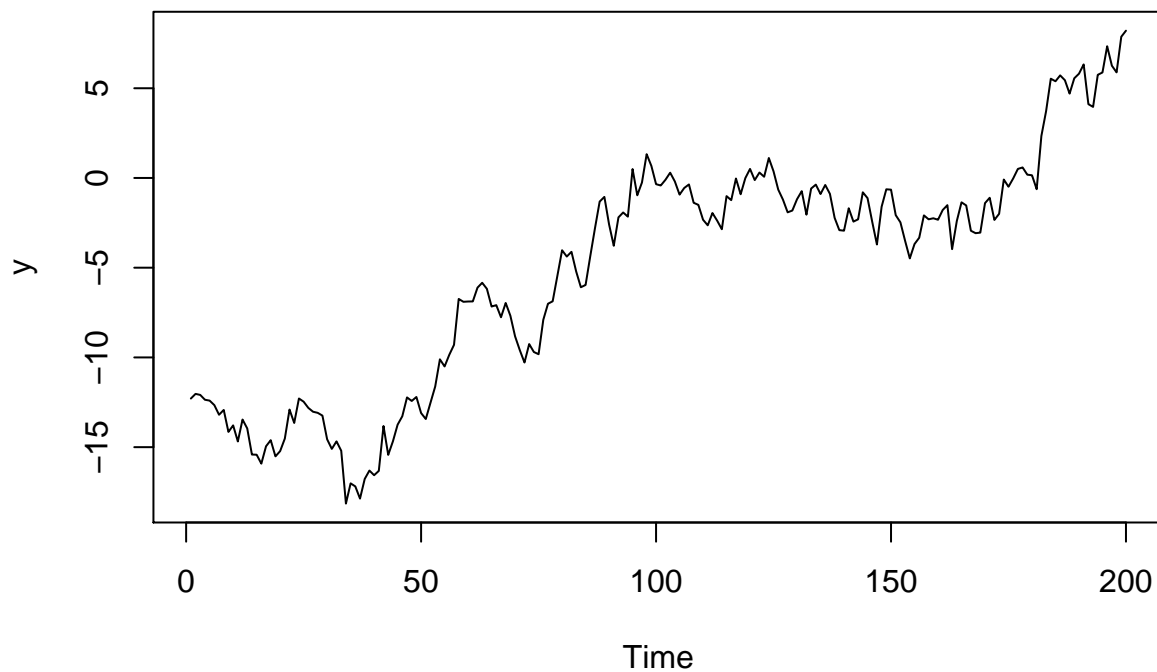


Nos indica que hay una estructura en la serie que no es ruido blanco pero SI estacionaria (cae de 1 a “casi” cero)

#### Simulación:

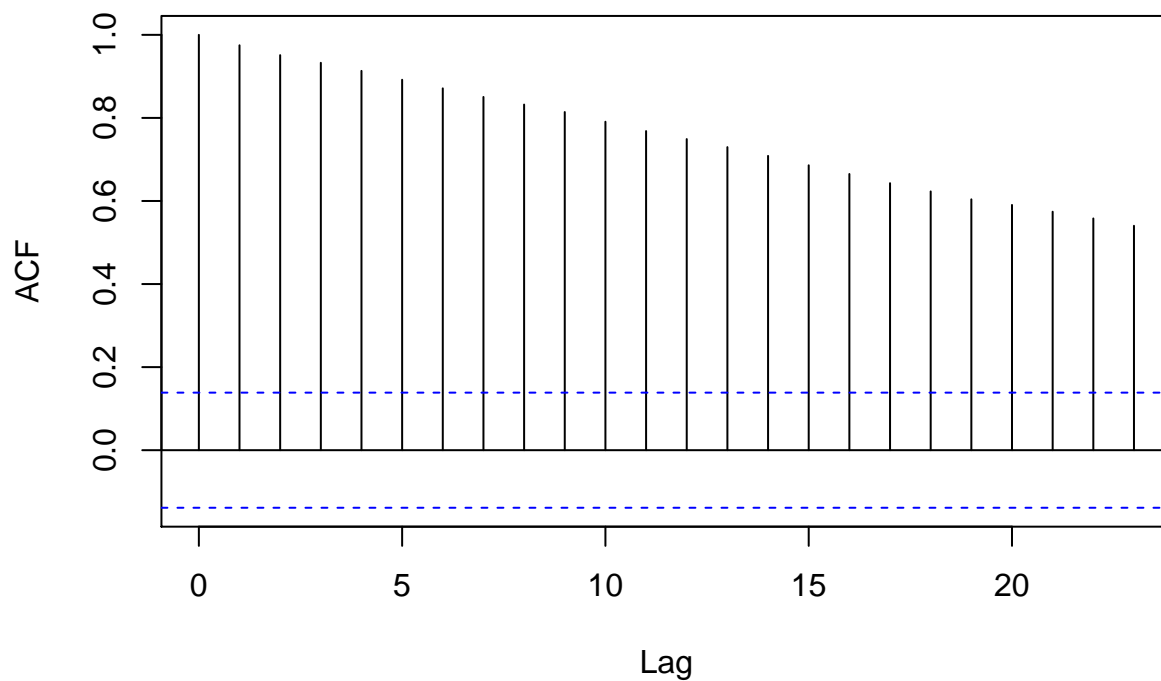
El siguiente paso es modelar esta estructura. Un modelo para ello es un modelo autorregresivo. Simular un  $AR(1)$ .

```
y <- arima.sim(list(ar=c(0.99),sd=1),n=200)
plot(y)
```



```
acf(y)
```

**Series y**

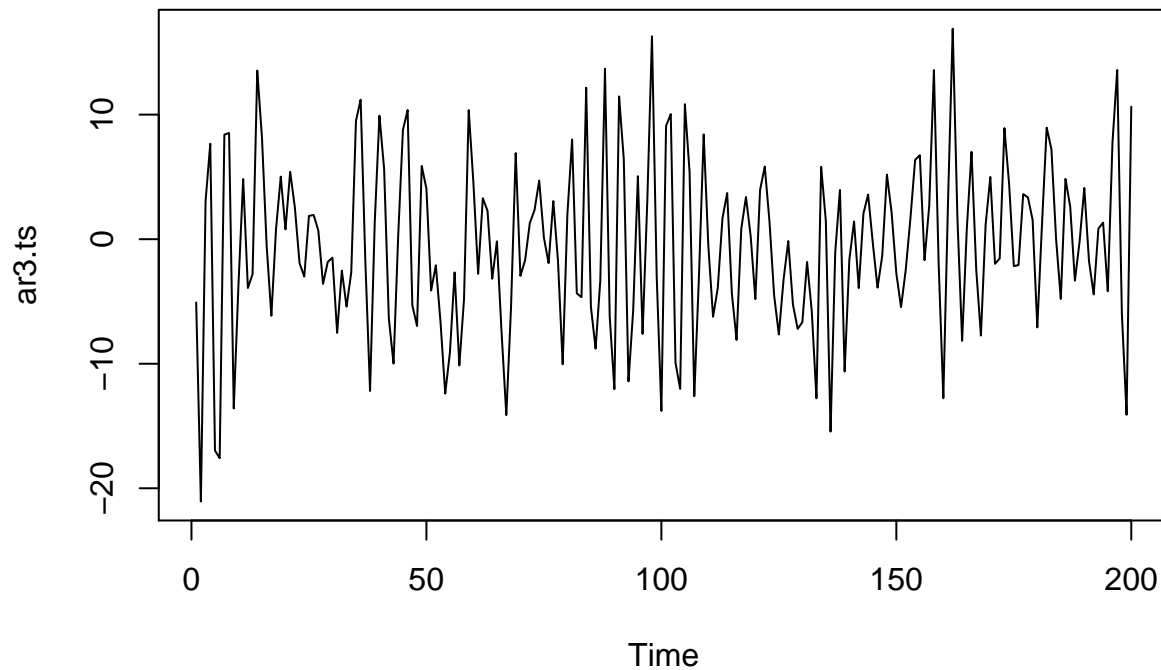


¿Cuáles son los parámetros del `arima.sim`? Hemos simulado  $Y_t = \phi_0 + \phi_1 Y_{t-1} = \phi_0 + 0.99 Y_{t-1}$ .

Simulemos el modelo:  $Y_t = 0.5 Y_{t-1} - 0.7 Y_{t-2} + 0.6 Y_{t-3}$

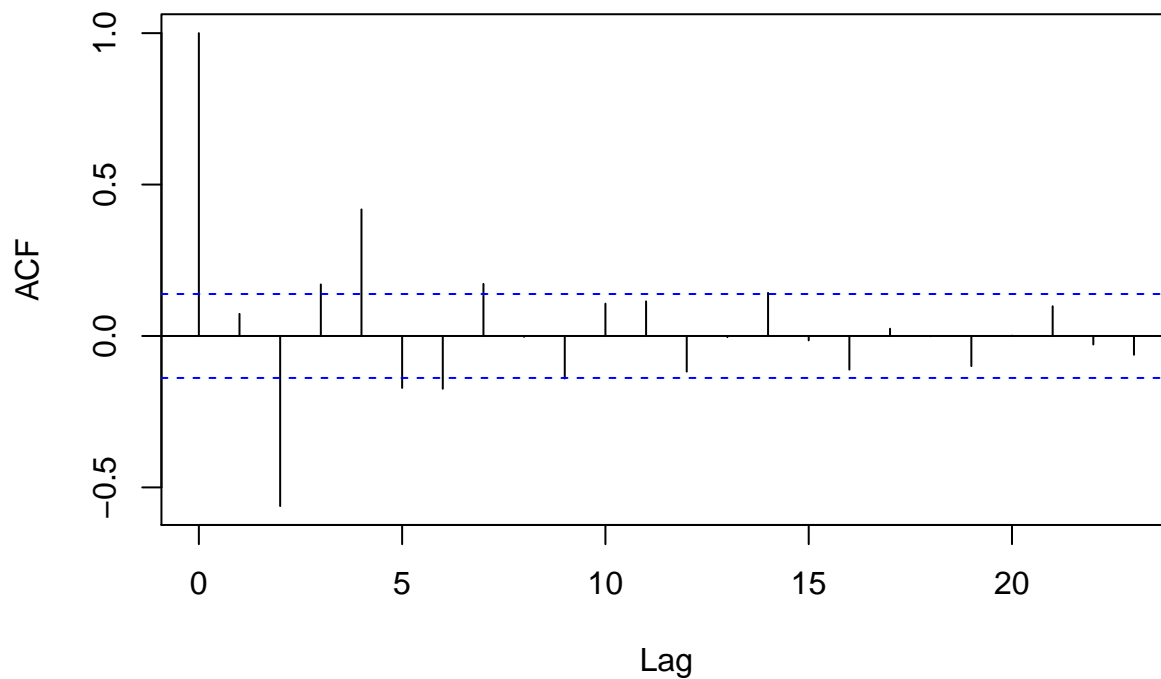
```
ar3 <- arima.sim(n=200,list(ar=c(0.5,-0.7,0.6)),sd=5)
ar3.ts = ts(ar3)
plot(ar3.ts)
```





```
acf(ar3)
```

### Series ar3



Las autocorrelaciones decaen exponencialmente a cero

**Autocorrelaciones parciales:** nos ayuda a determinar el orden del modelo.

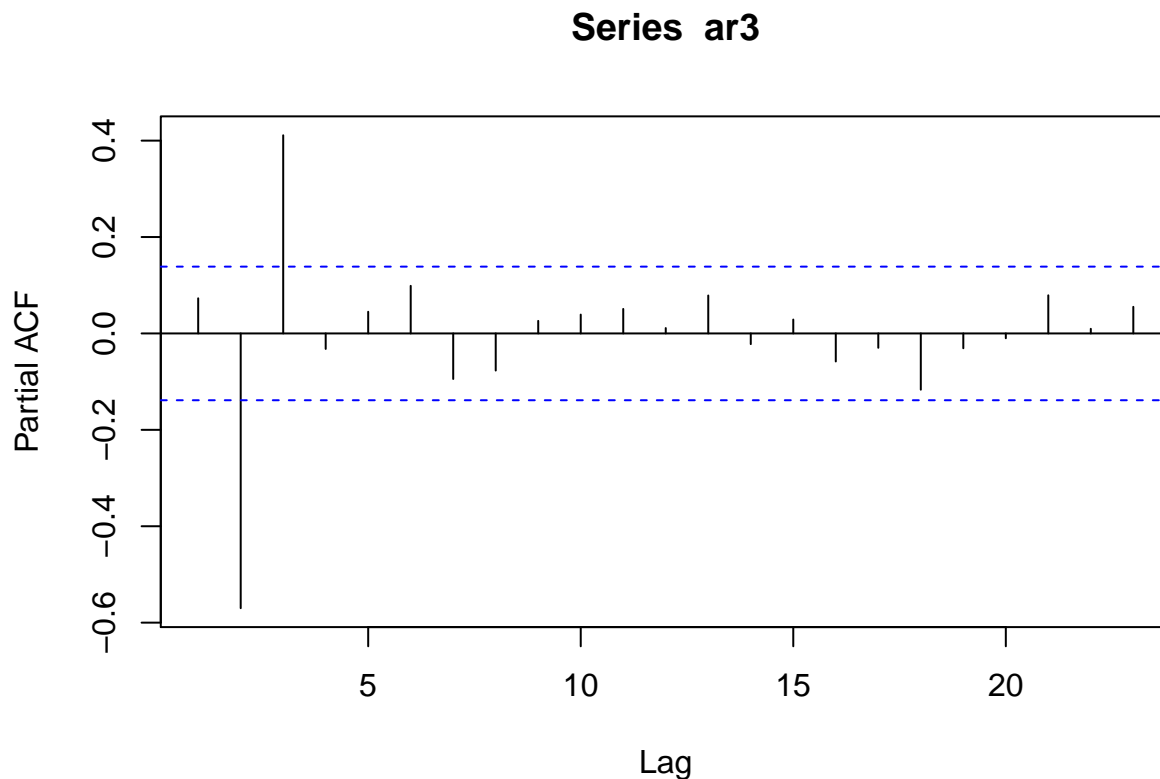
La autocorrelación parcial es la correlación entre  $Y_t$  y  $Y_{t-k}$  después de eliminar el efecto de las  $Y$  intermedias.

**Definición** Suponga que  $(Y_t, t \in \mathbb{Z})$  es estacionaria. La pacf muestral es una función de  $k$ ,

1.  $\hat{\alpha}(1) = \hat{\rho}(1)$
2.  $\hat{\alpha}(2)$ : se regresa  $Y_t$  sobre  $Y_{t-1}$  y  $Y_{t-2}$  tal que  $Y_t = \phi_{21}Y_{t-1} + \phi_{22}Y_{t-2} + \epsilon_t$  entonces  $\hat{\alpha}(2) = \phi_{22}$
3.  $\hat{\alpha}(k)$ : se regresa  $Y_t$  sobre  $Y_{t-1} \dots Y_{t-k}$  tal que  $Y_t = \phi_{k1}Y_{t-1} + \dots + \phi_{kk}Y_{t-k} + \epsilon_t$  entonces  $\hat{\alpha}(k) = \phi_{kk}$

En los datos de series de tiempo, una gran proporción de la correlación entre  $Y_t$  y  $Y_{t-k}$  puede deberse a sus correlaciones con los rezagos intermedios  $Y_1, Y_2, \dots, Y_{t-k+1}$ . La correlación parcial elimina la influencia de estas variables intermedias.

```
pacf(ar3)
```



```
ar(ar3)$aic
```

##	0	1	2	3	4	5
##	110.666988	111.600673	35.029208	0.000000	1.793229	3.388323
##	6	7	8	9	10	11
##	3.426863	3.634029	4.445275	6.309119	8.002508	9.484606
##	12	13	14	15	16	17
##	11.459652	12.215914	14.117396	15.951050	17.272178	19.094212
##	18	19	20	21	22	23
##	18.350351	20.161274	22.141235	22.885737	24.867037	26.256271

La tercera autocorrelación es la que esta fuera de las bandas, esto indica que el modelo es un AR(3)

### Ejemplo

Datos: precio de huevos desde 1901

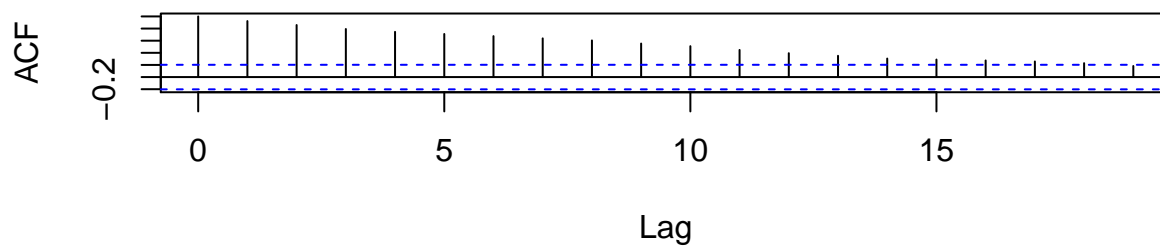
```
uu <- "https://raw.githubusercontent.com/vmoprojs/DataLectures/master/PrecioHuevos.csv"
datos <- read.csv(url(uu),header=T,sep=";")
ts.precio <- ts(datos$precio,start=1901)
plot(ts.precio)
```



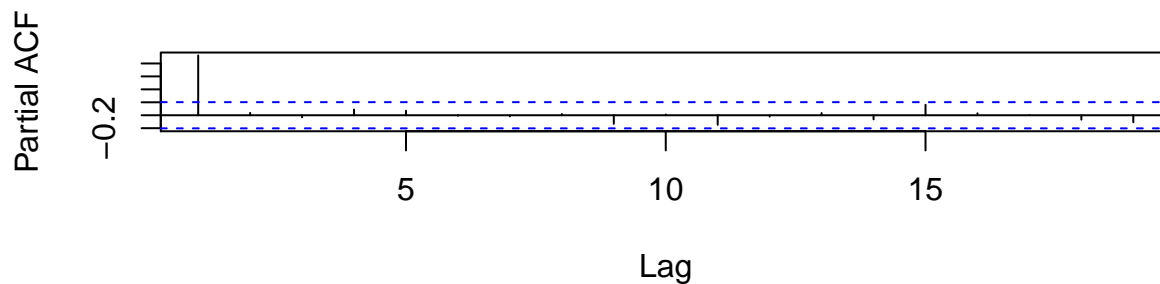
Veamos las autocorrelaciones:

```
par(mfrow=c(2,1))
acf(ts.precio)
pacf(ts.precio)
```

**Series ts.precio**



**Series ts.precio**



```
par(mfrow=c(1,1))
```

Las auto si decaen, no lo hacen tan rápido. No se puede decir si es estacionario o no.

Evaluemos un modelo:

```
modelo1 <- arima(ts.precio, order=c(1,0,0))
print(modelo1)
```

```
##
## Call:
## arima(x = ts.precio, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##      0.9517   195.5066
## s.e.  0.0310    48.1190
##
## sigma^2 estimated as 712.3:  log likelihood = -443.28,  aic = 892.56
modelo1$var.coef
```

```
##          ar1  intercept
## ar1      0.0009588394  -0.1532017
## intercept -0.1532017342 2315.4371739
```

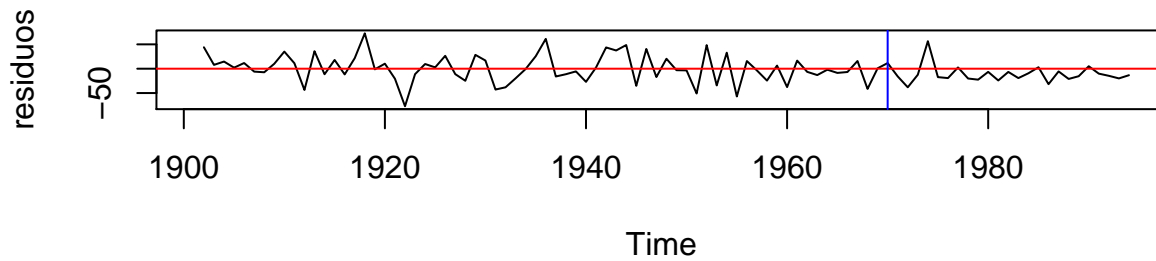
¿Qué nos recomienda R?

```
ar.precio <- ar(ts.precio)
ar.precio
```

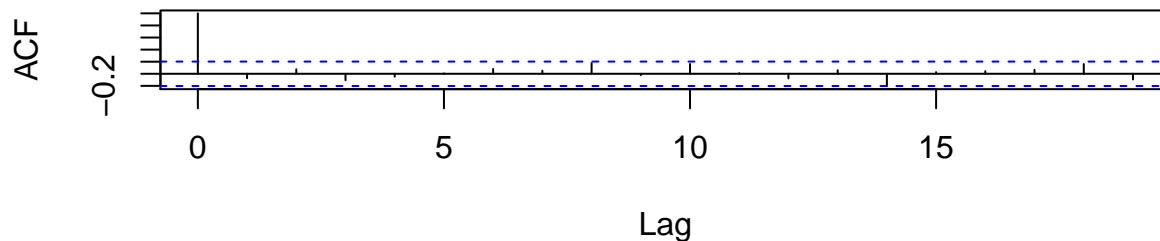
```
##
## Call:
## ar(x = ts.precio)
##
## Coefficients:
##      1
## 0.9237
##
## Order selected 1  sigma^2 estimated as 975.9
```

Analicemos los residuos

```
residuos = ar.precio$resid
# Los residuos debe estar sin ninguna estructura
par(mfrow = c(2,1))
plot(residuos)
abline(h=0,col="red")
abline(v=1970,col="blue")
acf(residuos,na.action=na.pass)
```



**Series residuos**



### Prueba de Ljung-Box

La prueba de Ljung-Box se puede definir de la siguiente manera.

$H_0$ : Los datos se distribuyen de forma independiente (es decir, las correlaciones en la población de la que se toma la muestra son 0, de modo que cualquier correlación observada en los datos es el resultado de la aleatoriedad del proceso de muestreo).

$H_a$ : Los datos no se distribuyen de forma independiente.

La estadística de prueba es:

$$Q = n(n+2) \sum_{k=1}^h \frac{\hat{\rho}_k^2}{n-k}$$

donde  $n$  es el tamaño de la muestra,  $\hat{\rho}_k$  es la autocorrelación de la muestra en el retraso  $k$  y  $h$  es el número de retardos que se están probando. Por nivel de significación  $\alpha$ , la región crítica para el rechazo de la hipótesis de aleatoriedad es

$$Q > \chi_{1-\alpha, h}^2$$

donde  $\chi_{1-\alpha, h}^2$  es la  $\alpha$ -cuantil de la distribución chi-cuadrado con  $m$  grados de libertad.

La prueba de Ljung-Box se utiliza comúnmente en autorregresivo integrado de media móvil de modelado (ARIMA). Tenga en cuenta que se aplica a los residuos de un modelo ARIMA equipada, no en la serie original, y en tales aplicaciones, la hipótesis de hecho objeto del ensayo es que los residuos del modelo ARIMA no tienen autocorrelación. Al probar los residuales de un modelo ARIMA estimado, los grados de libertad deben ser ajustados para reflejar la estimación de parámetros. Por ejemplo, para un modelo  $ARIMA(p, 0, q)$ , los grados de libertad se debe establecer en  $h - p - q$ .

```
Box.test(residuos, lag=20, type="Ljung")
```

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

```
##
## data:  residuos
## X-squared = 20.526, df = 20, p-value = 0.4255

Ho: Ruido Blanco ¿Es ruido blanco?

Probemos un segundo modelo
modelo2 <- arima(ts.precio, order=c(2,0,0))
print(modelo2)

##
## Call:
## arima(x = ts.precio, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##      0.8456  0.1134   193.5287
## s.e.  0.1026  0.1046    53.9009
##
## sigma^2 estimated as 703:  log likelihood = -442.7,  aic = 893.4

Comparemos los resultados:
ar2.precio <- ar(ts.precio,FALSE,2)
ar2.precio$aic

##          0          1          2
## 178.338243  0.000000  1.869475

modelo2$aic

## [1] 893.401

modelo1$aic

## [1] 892.563

Se escoge el modelo de menor AIC.
```

## Proceso de Medias Móviles (MA)

Recordemos el polinomio de rezagos:

$$B_p(L) = \beta_0 + \beta_1 L + \beta_2 L^2 + \cdots + \beta_p L^p$$

combinados con una serie de tiempo:

$$B_p(L)(Y_t) = (\beta_0 + \beta_1 L + \beta_2 L^2 + \cdots + \beta_p L^p)(Y_t)$$

$$B_p(L)(Y_t) = \sum_{j=0}^p \beta_j L^j Y_t$$

$$B_p(L)(Y_t) = \sum_{j=0}^p \beta_j L^j Y_t$$

### Definición

Se dice que una serie  $Y_t$  sigue un proceso  $MA(q)$ ,  $q = 1, 2, \dots$  de media móvil de orden  $q$ , si se cumple que

$$Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q}$$

para constantes  $\theta_1, \dots, \theta_q$  y  $\epsilon_t \sim N(0, \sigma^2)$ . La expresión con el operador  $L$  es, si se define el polinomio.

$$\theta_p(L) = 1 + \theta_1 L + \dots + \theta_q L^q$$

entonces la ecuación queda  $Y_t = \theta_q(L)(\epsilon_t)$

### Propiedades

1.  $E(Y_t) = 0$
2.  $Var(Y_t) = (1 + \theta_1^2 + \dots + \theta_q^2)\sigma^2$

luego  $Var(Y_t) > Var(\epsilon_t)$ , en general. 3.  $Cov(Y_t, Y_{t+k}) = R(k)$ , donde

$$R(K) = \sigma^2 \sum_{j=0}^{q-k} \theta_j \theta_{j+k}$$

donde  $\theta_0 = 1$  y  $k < q + 1$ .  $R(K) = 0$  si  $k \geq q + 1$ .

4. Un  $MA(q)$  siempre es un proceso estacionario con ACF,  $p(k) = \frac{R(k)}{R(0)}$

La ecuación () se puede interpretar como una indicación de que un  $MA(q)$  es un proceso débilmente correlacionado, ya que su autocovarianza es cero a partir de un valor. Por esta razón se puede ver los procesos  $MA(q)$  como alternativas al Ruido Blanco completamente incorrelacionado.

### Ejemplo

Sea  $Y_t \sim MA(2)$  dado por:

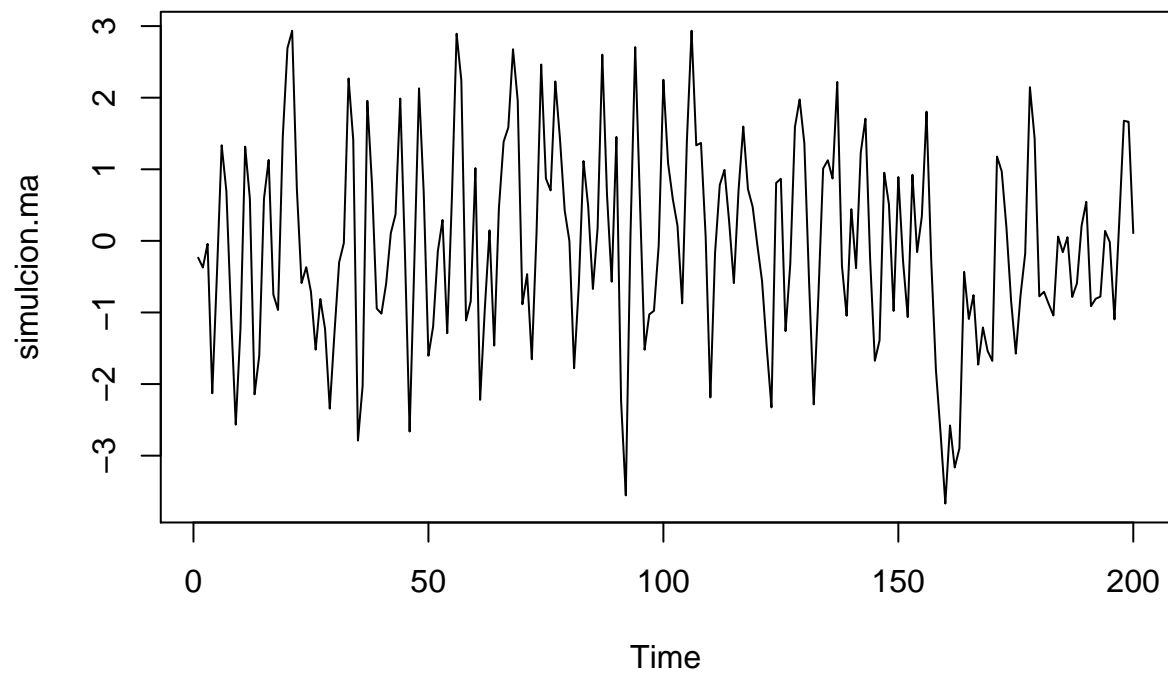
$$y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$$

donde  $\epsilon_t \sim N(0, 9)$ , con  $\theta_1 = -0.4, \theta_2 = 0.4$ .

De acuerdo con (), si la fac muestral de una serie  $Y_t$  termina abruptamente puede tratarse de un  $MA(q)$ .

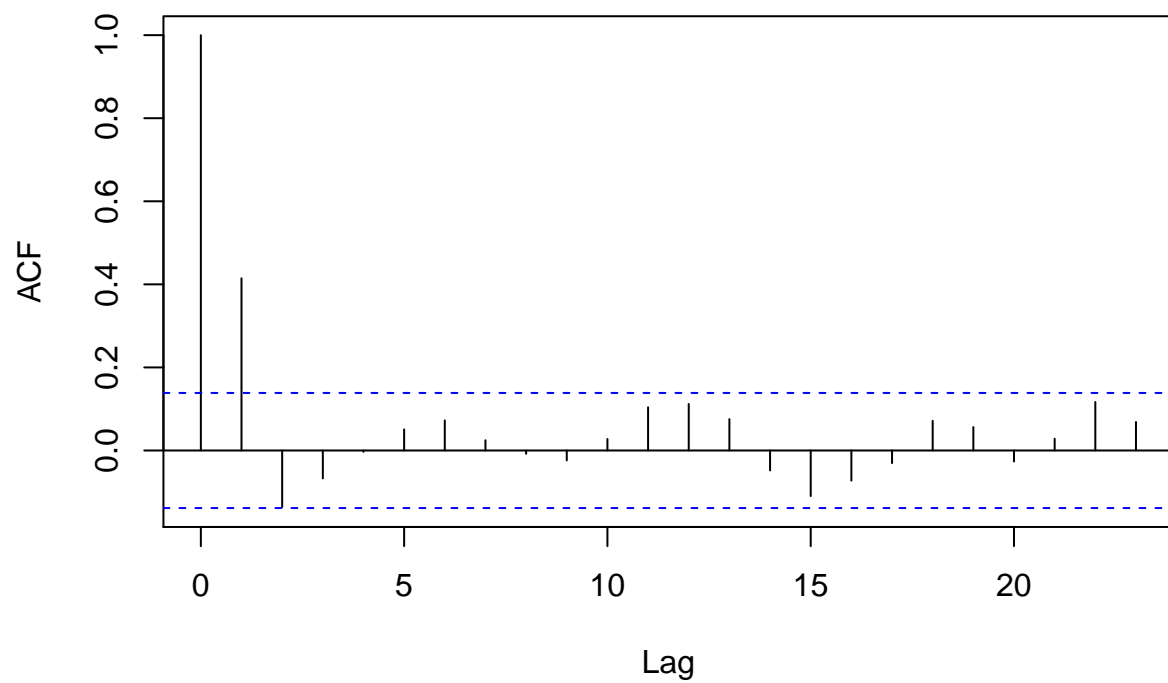
Simulemos un modelo:

```
simulcion.ma <- arima.sim(200,model=list(ma=c(0.8)))  
plot(simulcion.ma)
```



```
acf(simulcion.ma)
```

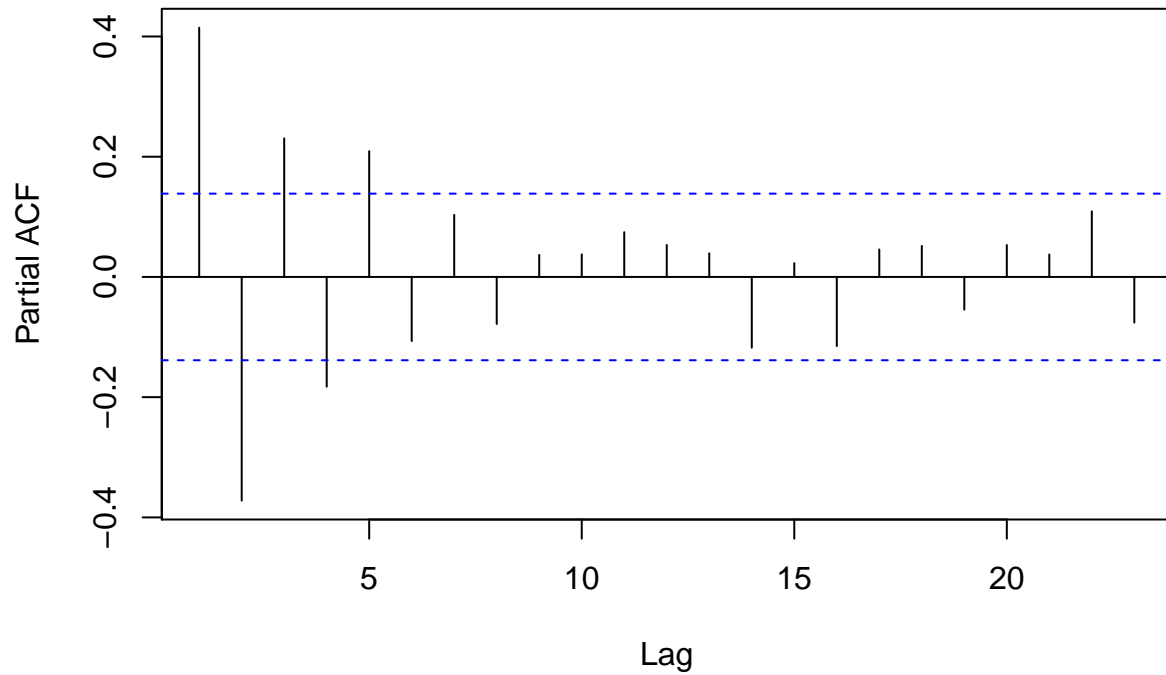
**Series simulcion.ma**



```
pacf(simulcion.ma)
```



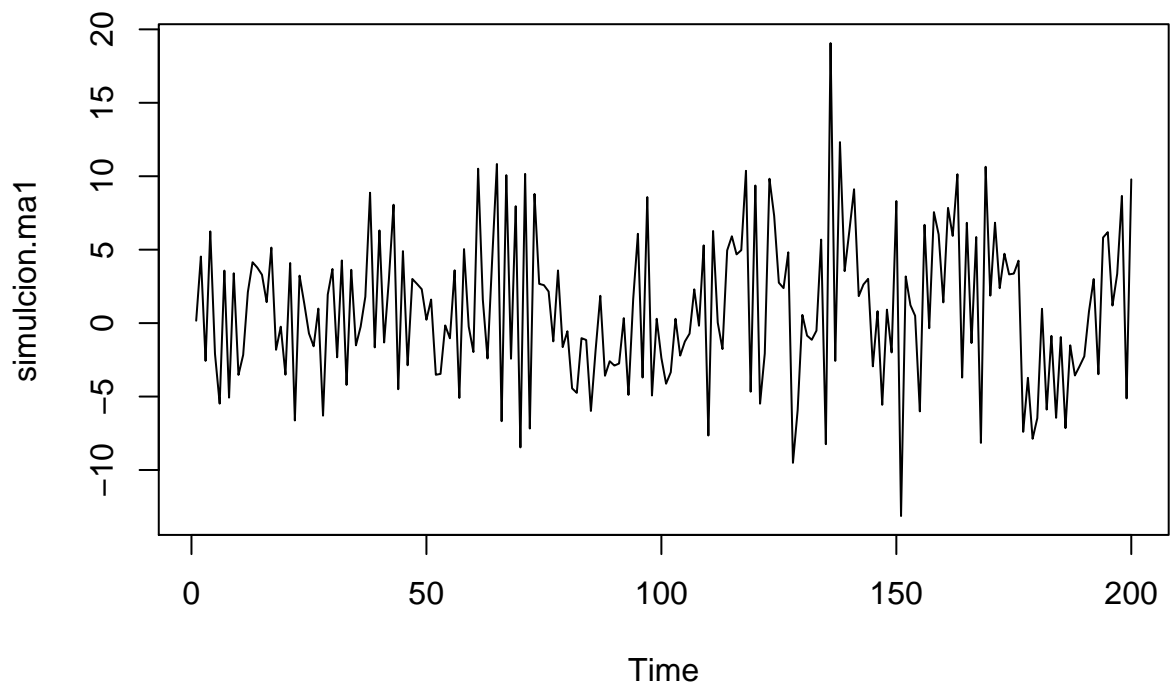
## Series simulcion.ma



- Las  $p$  primeras autocorrelaciones van a ser diferentes de cero
- La autocorrelación parcial decae exponencialmente

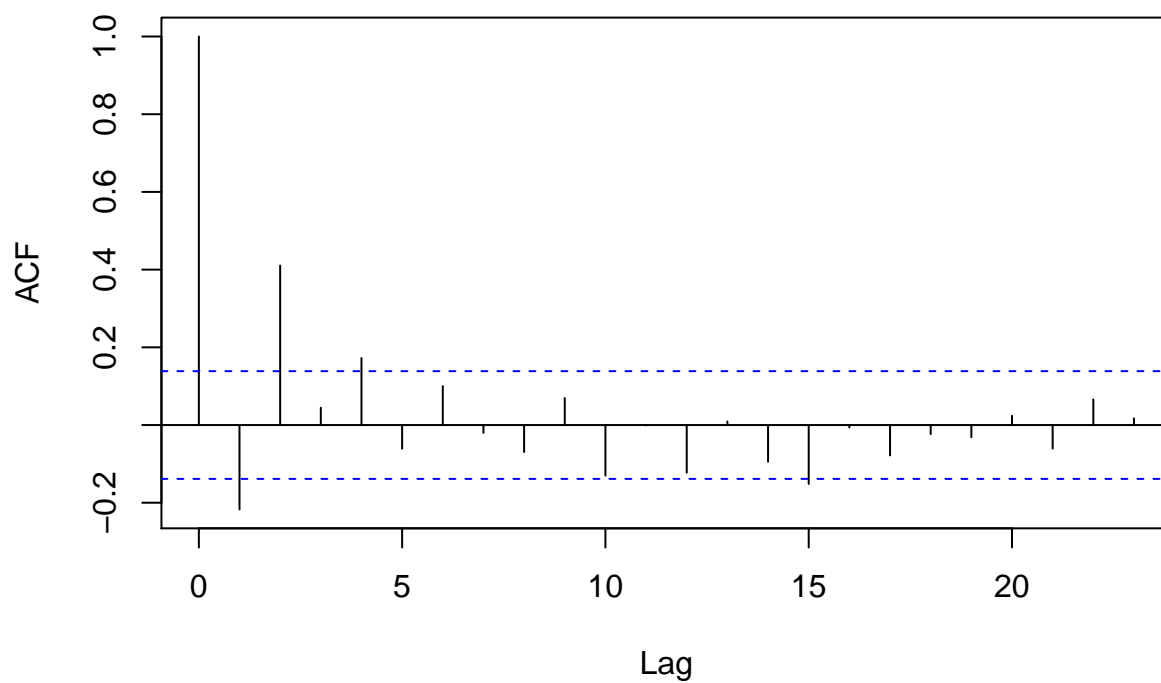
Veamos otro ejemplo

```
simulcion.ma1 <- arima.sim(200, model =list(ma=c(2.1,-0.9,4.7)))  
plot(simulcion.ma1)
```



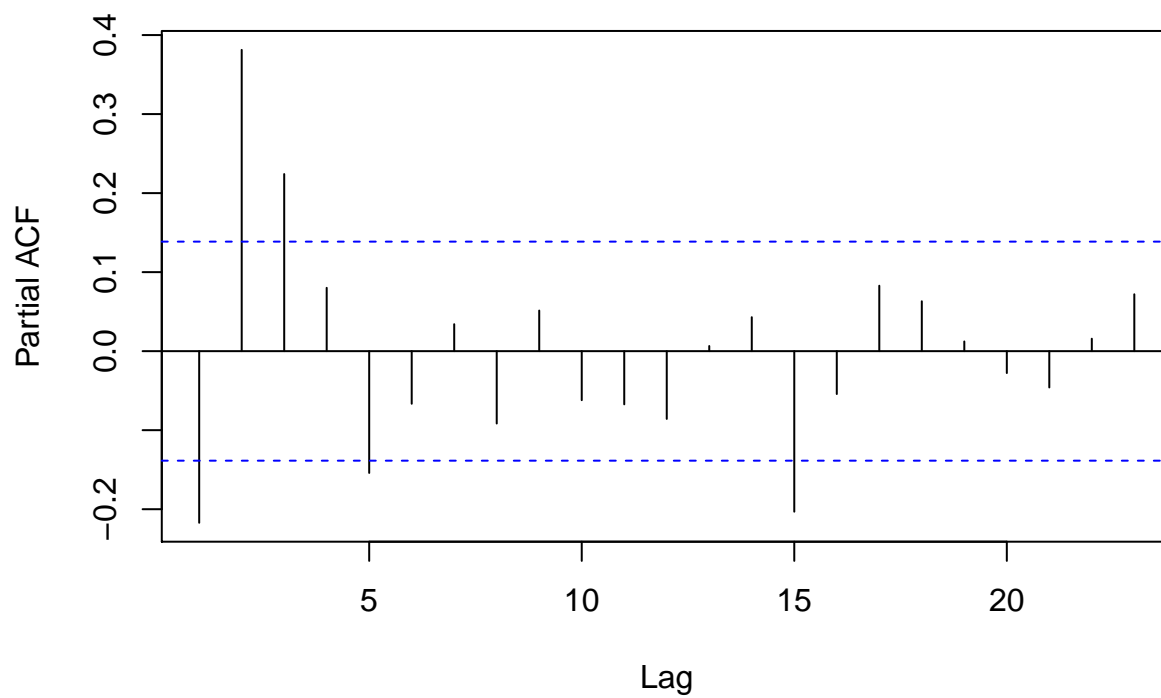
```
acf(simulcion.ma1)
```

### Series simulcion.ma1



```
pacf(simulcion.ma1)
```

### Series simulcion.ma1



# Proceso ARMA

## Definición

Un proceso  $Y_t \sim ARMA(p, q)$  se define mediante

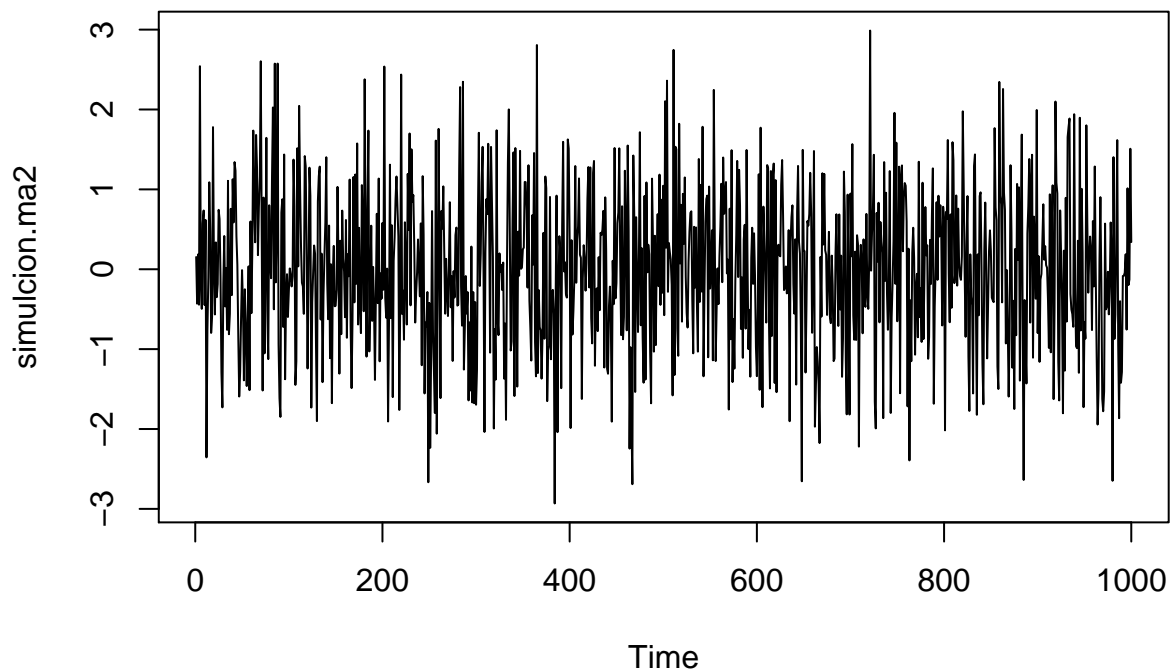
$$\phi_p(L)(Y_t) = \theta_q(L)\epsilon_t$$

donde  $\epsilon_t \sim RB(0, \sigma^2)$  y  $\phi_p(z) = 1 - \sum_{j=1}^p \phi_j z^j$ ,  $\theta_q(z) = 1 + \sum_{j=1}^q \theta_j z^j$  son los polinomios autorregresivo y de media móvil respectivamente.

se asume que las raíces de las ecuaciones  $\phi_p(z) = 0$  y  $\theta_q(z) = 0$  están fuera del círculo unitario. Además se asume que estos polinomios no tienen raíces en común. Si se cumplen estas condiciones el proceso  $Y_t \sim ARMA(p, q)$  es estacionario e identificable.

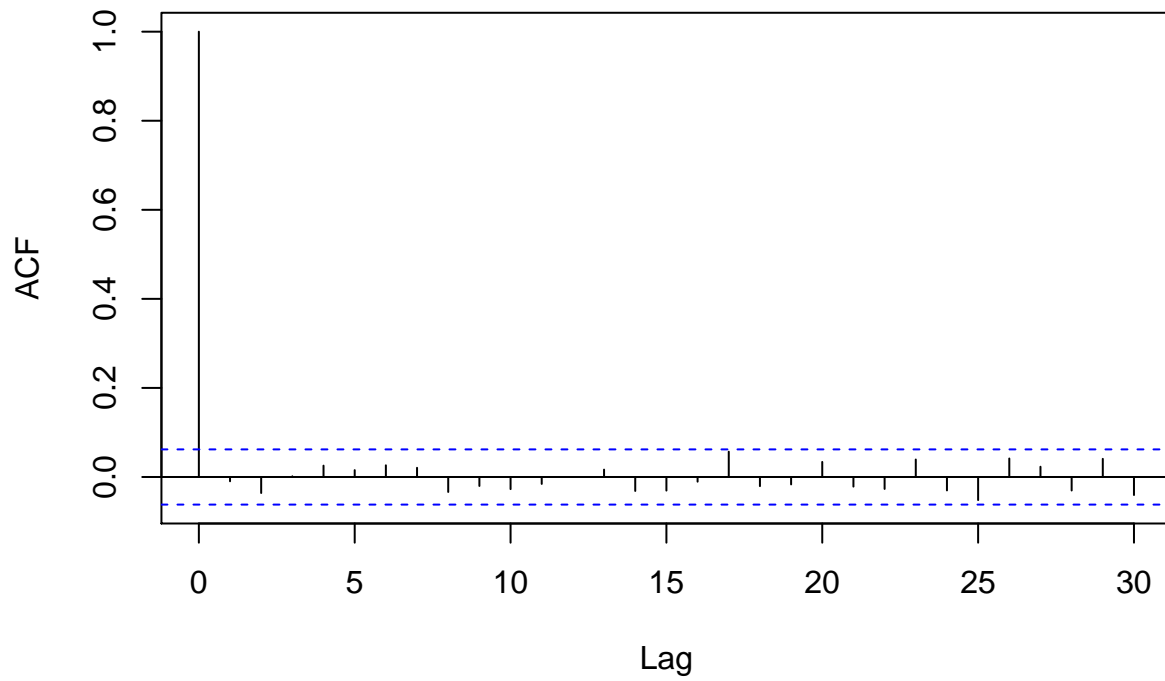
Simulemos el proceso:

```
simulcion.ma2 <- arima.sim(1000, model=list(order=c(1,0,1),ar=c(-0.1),ma=c(0.1)))  
plot(simulcion.ma2)
```



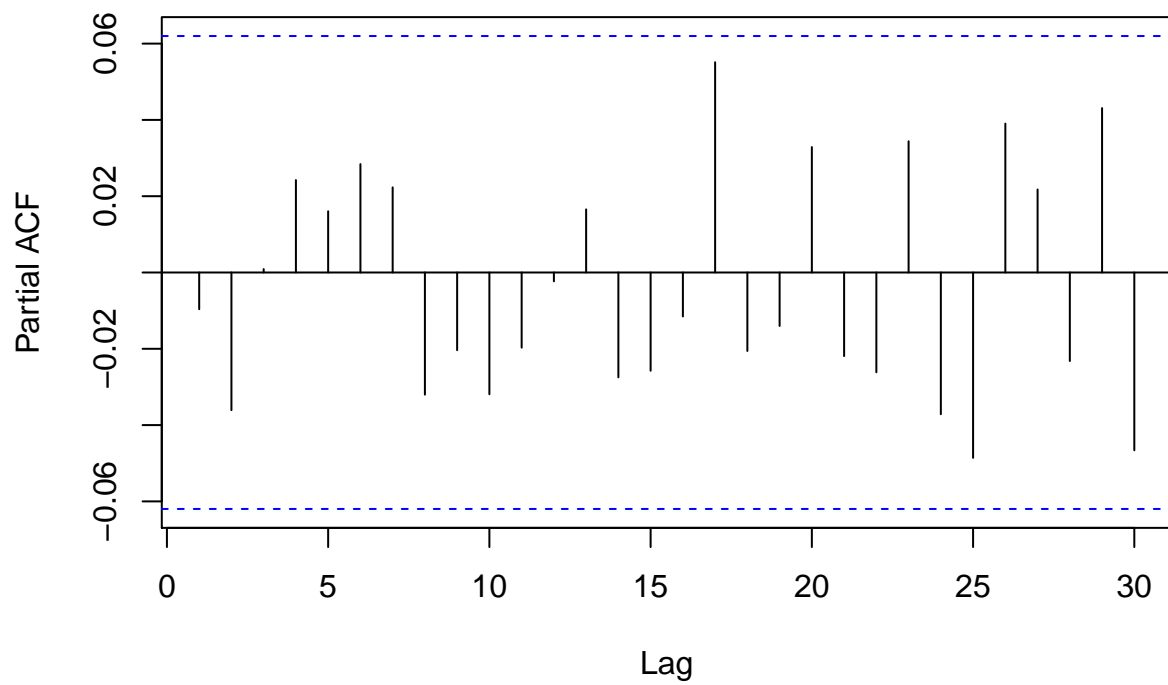
```
acf(simulcion.ma2)
```

### Series simulcion.ma2



```
pacf(simulcion.ma2)
```

### Series simulcion.ma2



## Buscando el *mejor* modelo

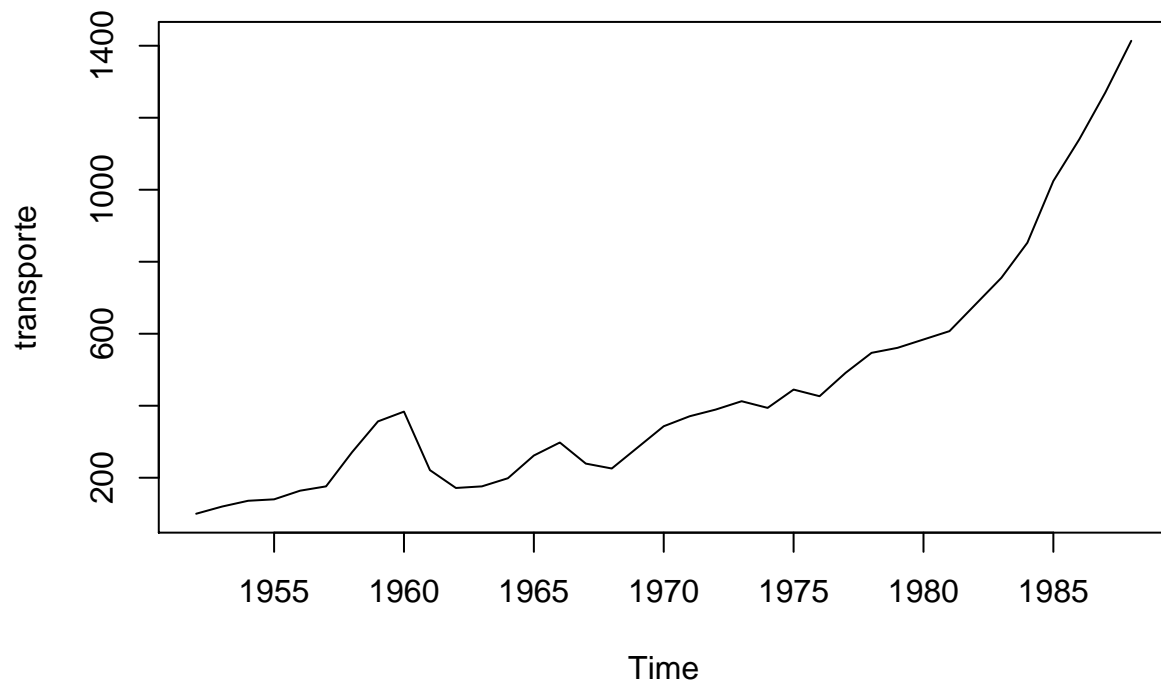
### Ejemplo 1

Datos: Serie de tiempo (1952-1988) del ingreso nacional real en China por sector (año base: 1952)

```
library(AER)
data(ChinaIncome)
str(ChinaIncome)

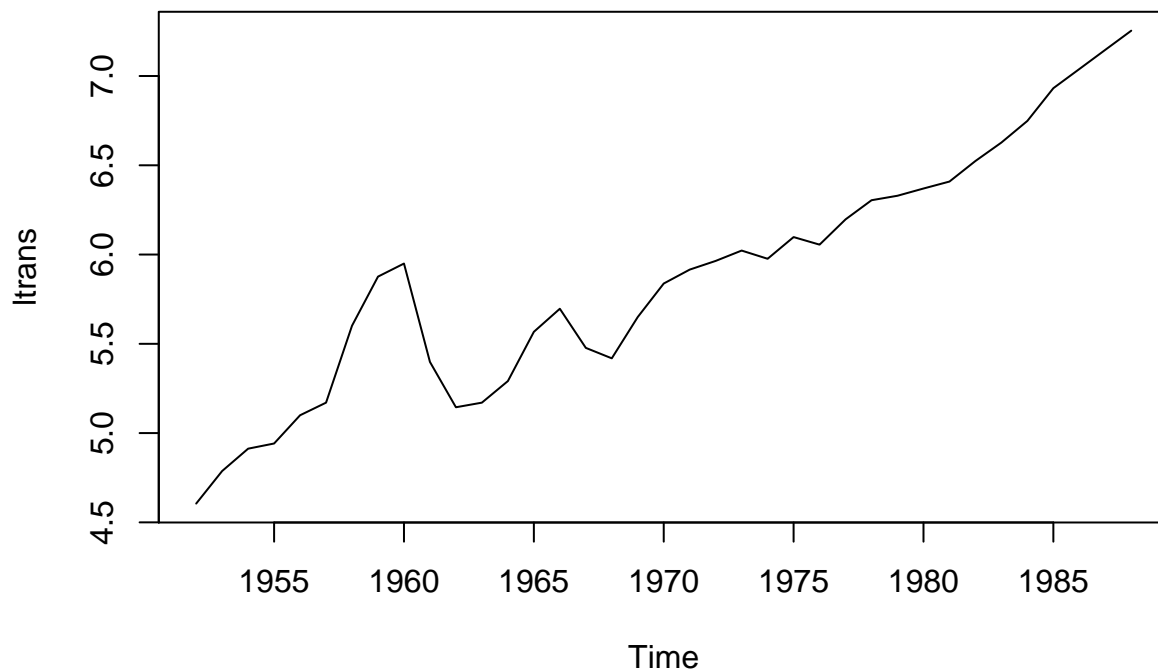
## Time-Series [1:37, 1:5] from 1952 to 1988: 100 102 103 112 116 ...
## - attr(*, "dimnames")=List of 2
## ..$ : NULL
## ..$ : chr [1:5] "agriculture" "commerce" "construction" "industry" ...

transporte <- ChinaIncome[, "transport"]
ts.plot(transporte)
```



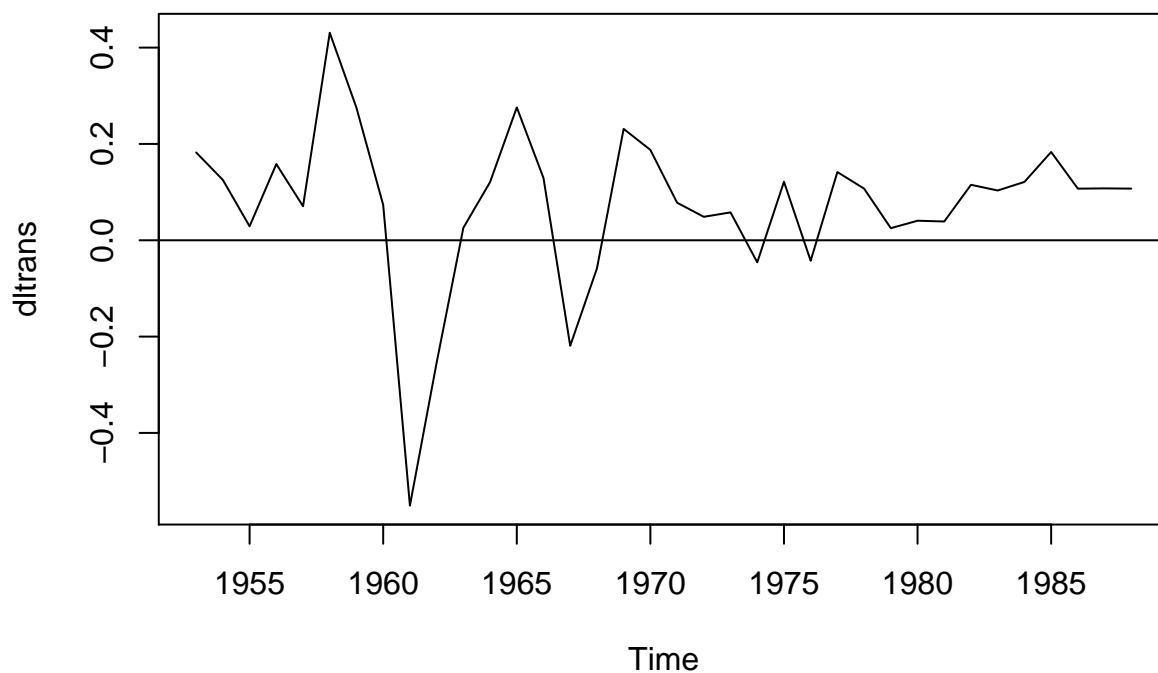
Parece no ser estacionario. Hacemos una transformación para tratar de confirmar la estacionariedad.

```
ltrans <- log(transporte)
ts.plot(ltrans)
```



Notamos que persiste el problema, sigue sin ser estacionario. Probemos con la diferencia:

```
dltrans <- diff(ltrans)
ts.plot(dltrans)
abline(h=0)
```

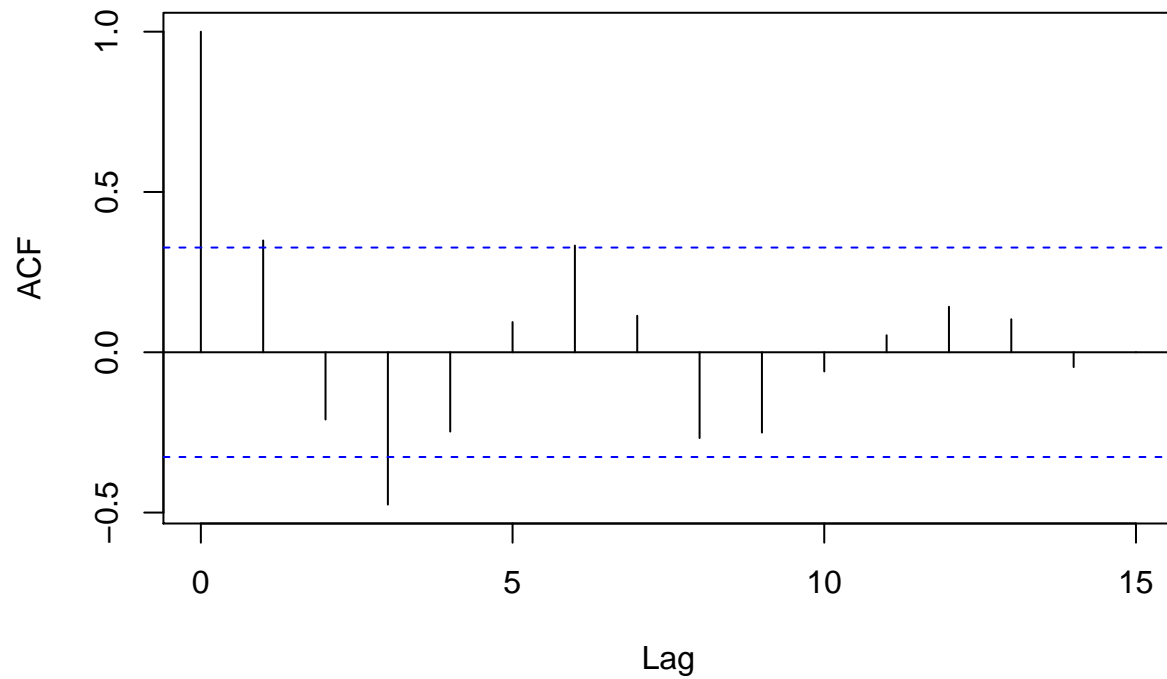


Asumamos estacionariedad (después haremos una prueba específica para verificar estacionariedad) y busquemos el mejor modelo.

Usaremos el `acf` y el `pacf` para evaluar si es MA o AR.

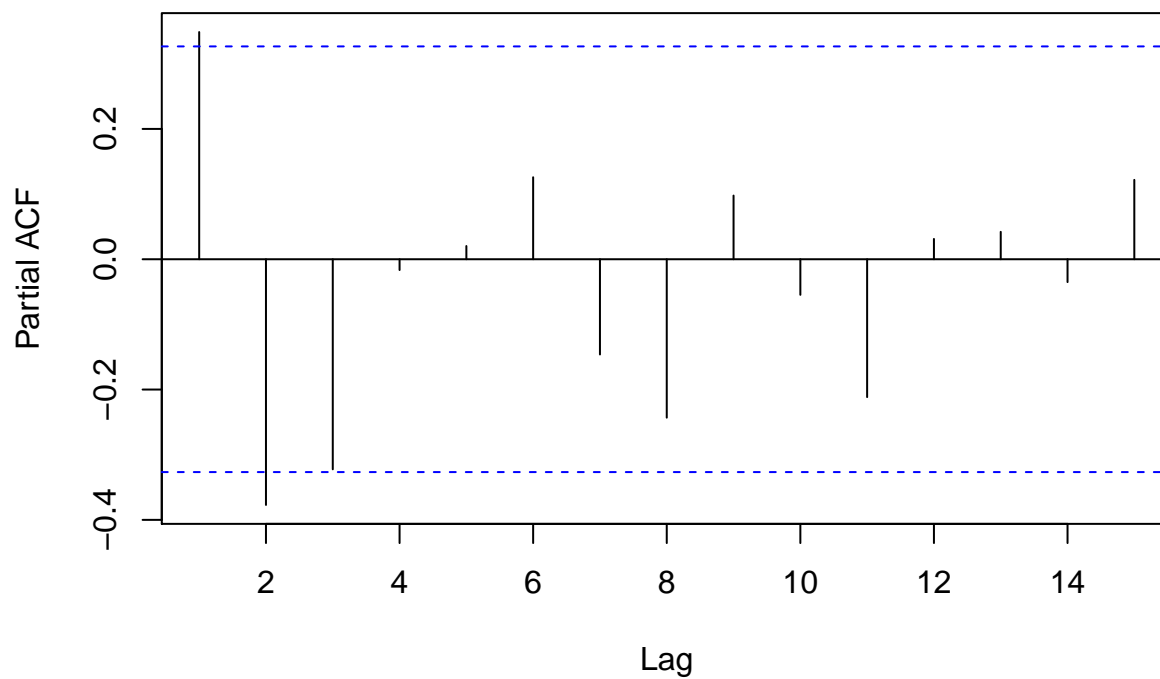
```
acf(dltrans)
```

### Series dltrans



```
pacf(dltrans)
```

### Series dltrans



Ajustando según la gráfica, tendríamos un proceso  $MA(2)$

¿Qué recomienda R?

```
ar(dltrans)$aic
```

```
##          0          1          2          3          4          5          6
## 8.140754 5.473906 1.951624 0.000000 1.990078 3.975236 5.400529
##          7          8          9         10         11         12         13
## 6.622877 6.428684 8.083145 9.975465 10.326488 12.291432 14.227677
##         14         15
## 16.183322 17.644775
```

Según esta recomendación, estamos ante un proceso  $AR(3)$ .

```
modelo1 <- arima(dltrans,order = c(3,0,0))
modelo1$aic
```

```
## [1] -32.75694
```

Ajustemos el  $MA(2)$  y comparemos:

```
modelo2 <- arima(dltrans,order = c(0,0,2))
modelo2$aic
```

```
## [1] -28.01574
```

Recuerda: Un menor AIC es mejor. ¿Con qué modelo te quedas?

Ajustemos un  $MA(3)$ :

```
modelo3 <- arima(dltrans,order = c(0,0,3))
print(modelo3)
```

```
##
## Call:
## arima(x = dltrans, order = c(0, 0, 3))
##
## Coefficients:
##          ma1          ma2          ma3 intercept
##          0.1763 -0.4596 -0.7167          0.0621
## s.e.  0.1883  0.1640  0.1925          0.0053
##
## sigma^2 estimated as 0.01625:  log likelihood = 21.26,  aic = -32.53
modelo3$aic
```

```
## [1] -32.52547
```

Este modelo es mejor que el  $MA(2)$ , pero peor que  $AR(3)$ .

Probemos algunas combinaciones

```
# Ajustando un ARMA(1,1)
modelo4 <- arima(dltrans,order = c(1,0,1))
modelo4$aic
```

```
## [1] -27.49033
```

```
# Ajustando un ARMA(2,1)
modelo5 <- arima(dltrans,order = c(2,0,1))
modelo5$aic
```

```
## [1] -32.45195
```

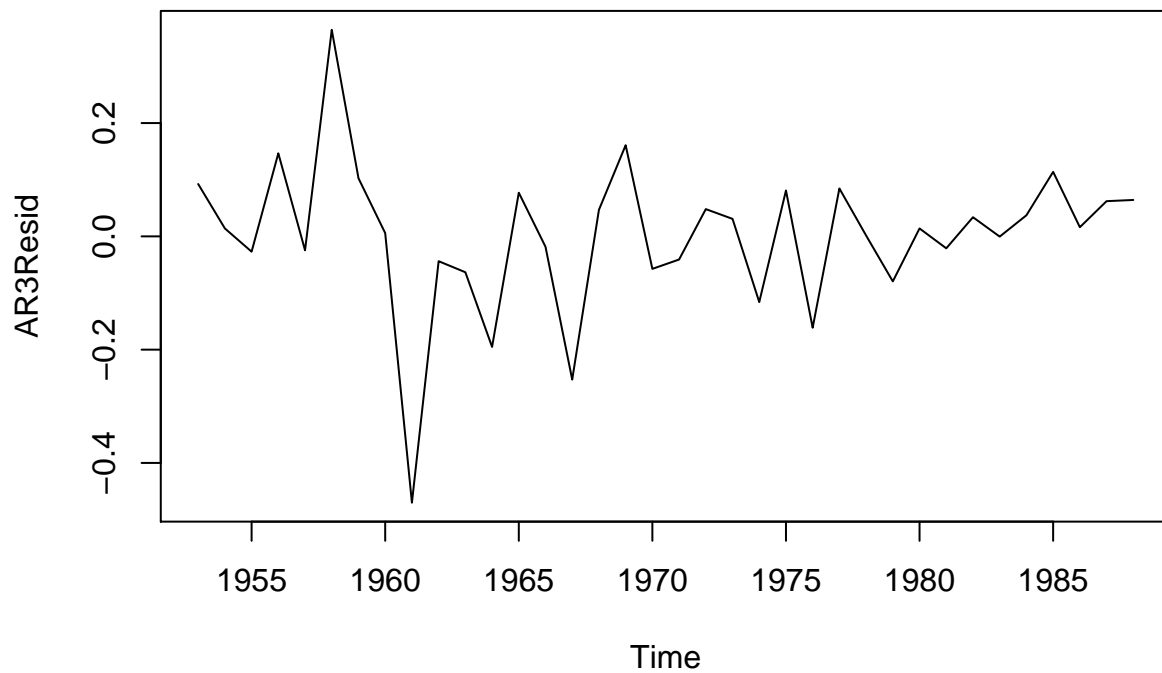


```
# Ajustando un ARMA(1,2)
modelo6 <- arima(dltrans,order = c(1,0,2))
modelo6$aic
```

```
## [1] -29.86264
```

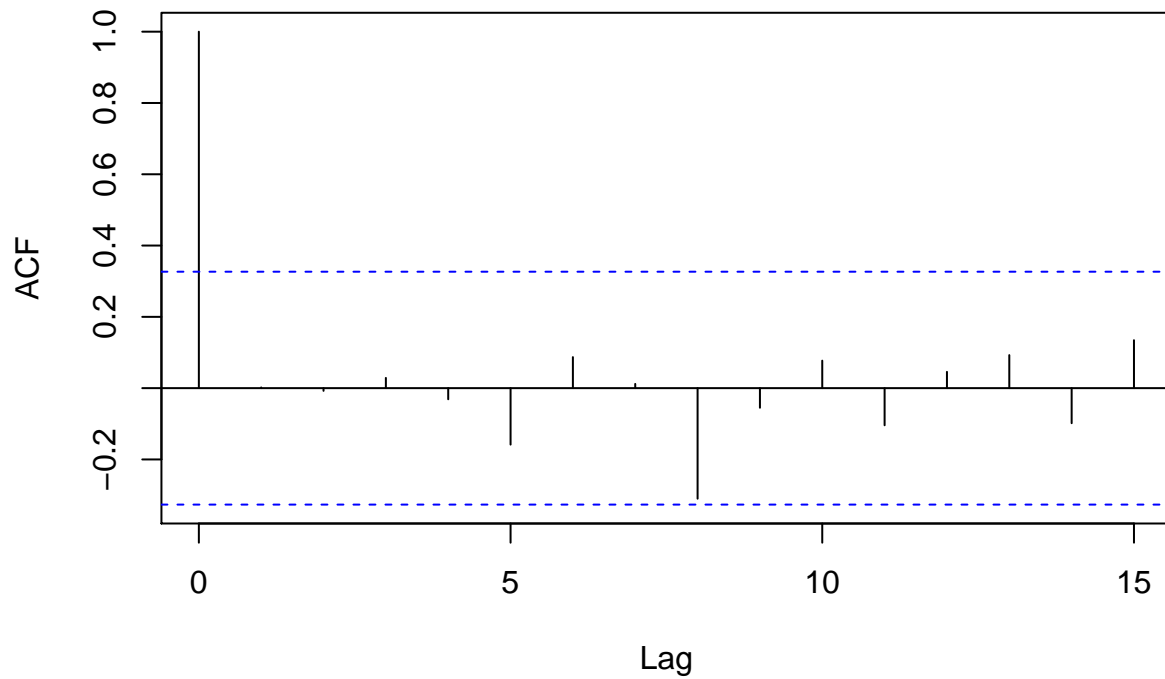
Nos quedamos con el  $AR(3)$ . Revisemos los residuos:

```
AR3Resid <- (modelo1$resid)
ts.plot(AR3Resid)
```



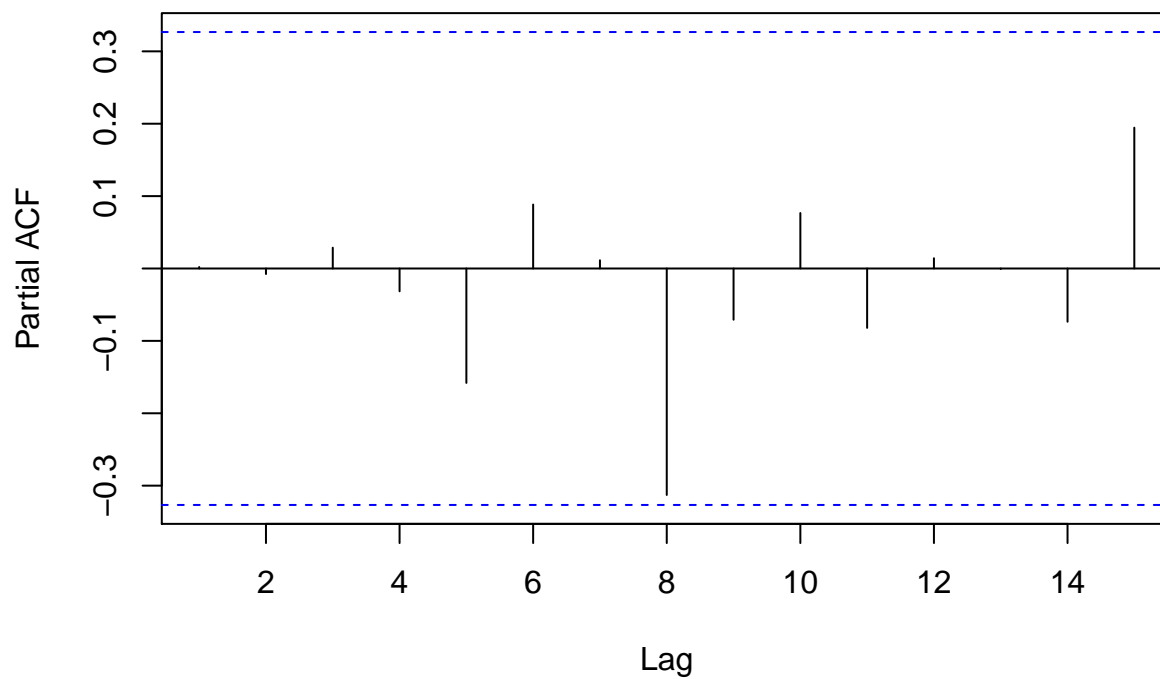
```
acf(AR3Resid)
```

### Series AR3Resid



```
pacf(AR3Resid)
```

### Series AR3Resid



hay autocorrelación, No hay autocorrelación parcial.

Veamos si se trata de un ruido blanco

No

```
Box.test(AR3Resid)
```

```
##  
## Box-Pierce test  
##  
## data: AR3Resid  
## X-squared = 0.00015103, df = 1, p-value = 0.9902
```

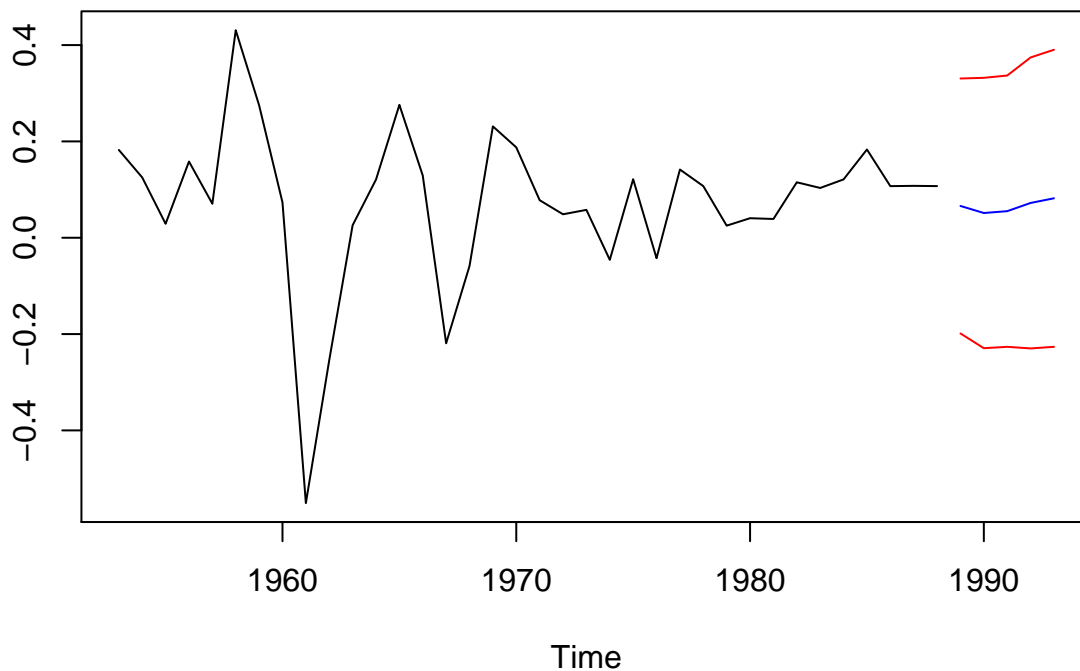
¿Es ruido blanco?

Realicemos una proyección a 5 años

```
pred5 <- predict(modelo1, n.ahead=5, se=T)  
pred5se <- pred5$se
```

intervalos de confianza:

```
ic = pred5$pred + cbind(-2*pred5$se, 2*pred5$se)  
ts.plot(dltrans, pred5$pred, ic,  
col=c("black", "blue", "red", "red"))
```

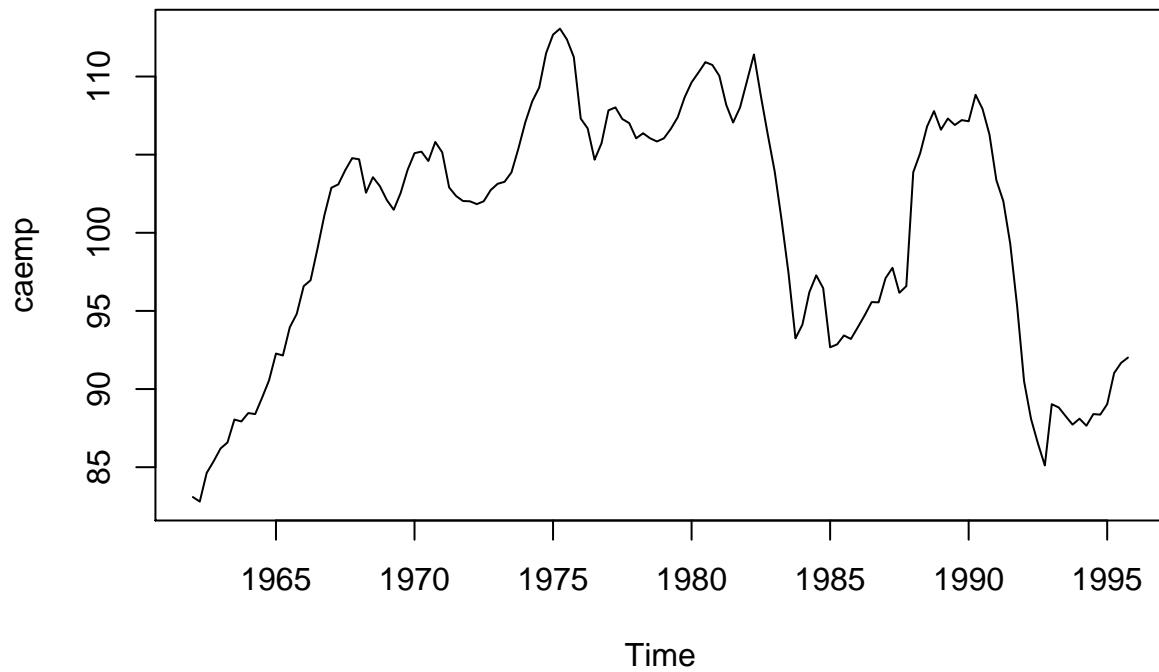


Los intervalos son grandes, podría ser por la cantidad de datos

## Ejemplo 2

Datos: Índice de desempleo trimestral en Canada desde el 62

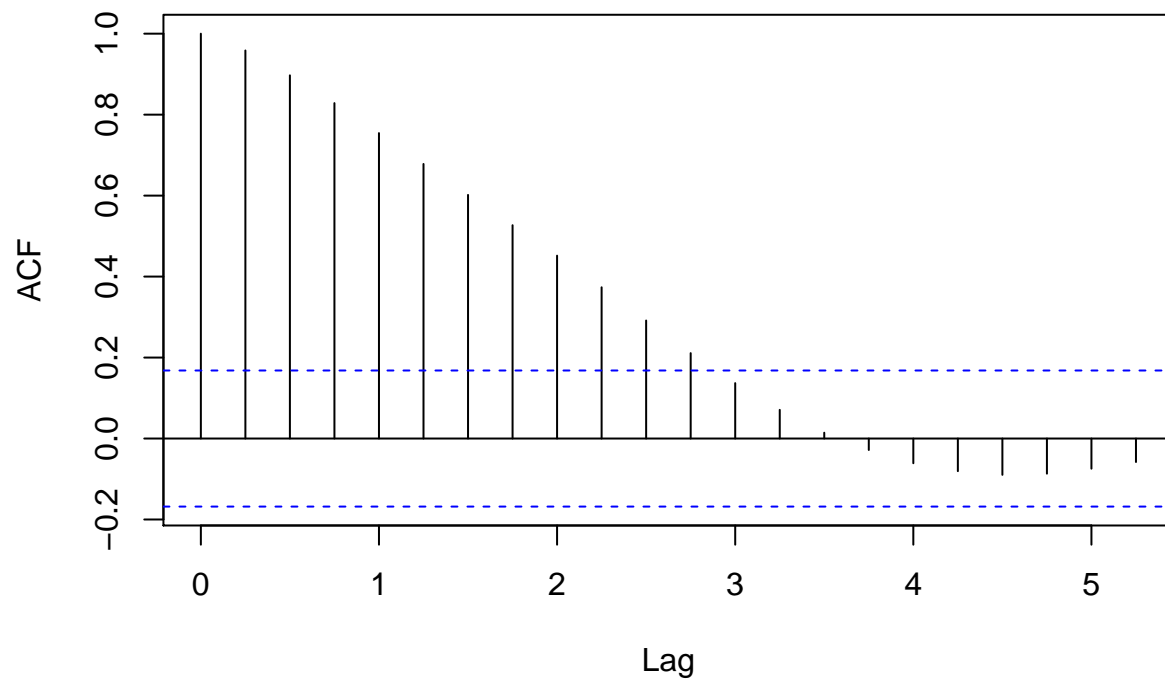
```
uu <- "https://raw.githubusercontent.com/vmoprojs/DataLectures/master/CAEMP.DAT"  
datos <- read.csv(url(uu), sep=",", header=T)  
emp.ts <- ts(datos, st=1962, fr=4)  
plot(emp.ts)
```



Veamos sus autocorrelaciones y AIC:

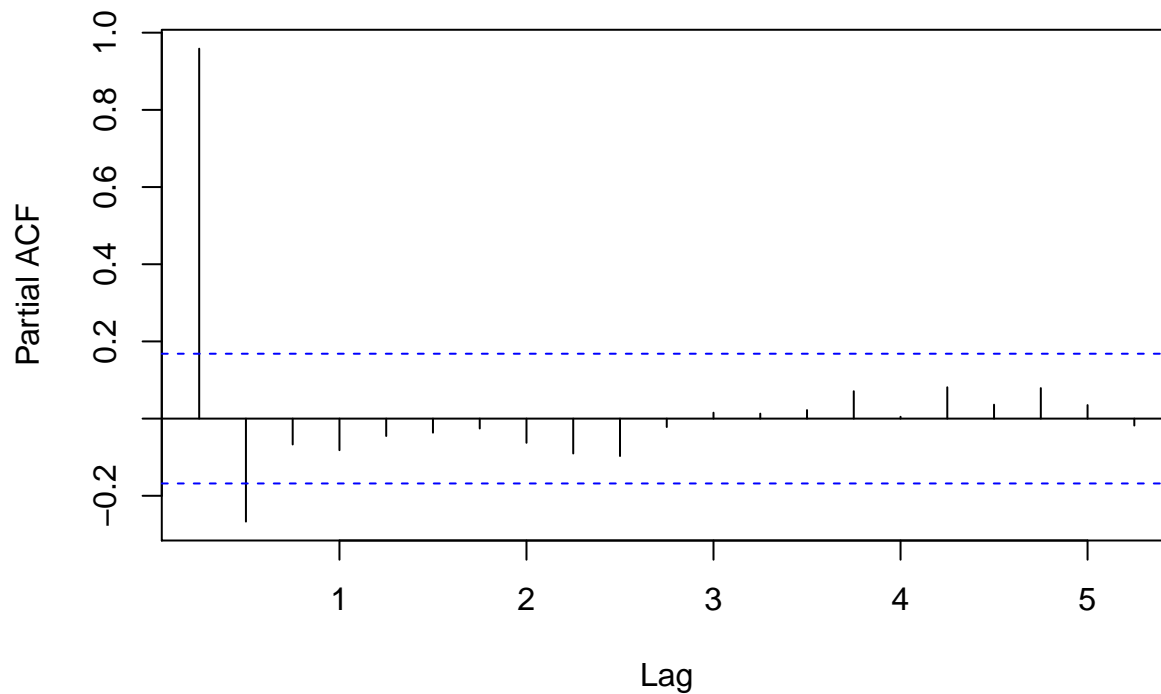
```
acf(emp.ts)
```

**caemp**



```
pacf(emp.ts)
```

## Series emp.ts



```
ar(emp.ts)$aic
```

```
##          0          1          2          3          4          5
## 347.493734  8.048336  0.000000  1.386627  2.471906  4.195572
##          6          7          8          9         10         11
##  6.015058  7.925583  9.390454 10.273560 10.989316 12.924608
##          12         13         14         15         16         17
## 14.892737 16.870041 18.803523 20.117383 22.114279 23.215372
##          18         19         20         21
## 25.038216 26.186319 28.019605 29.975609
```

Decae linealmente el ACF, esto es señal de que no es un proceso AR

Comparemos modelos:

```
model1 <- arima(emp.ts,order=c(2,0,0))
mode2 <- arima(emp.ts,order=c(0,0,4))
Box.test(model1$resid,t="Ljung",lag=20)
```

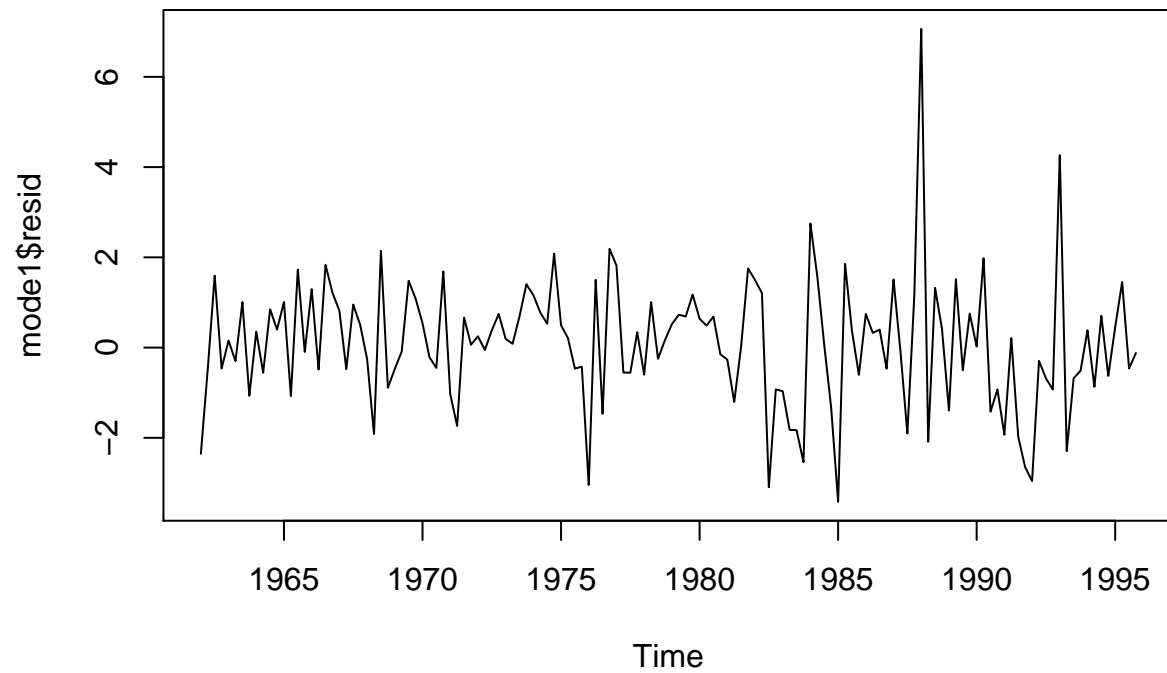
```
##
## Box-Ljung test
##
## data:  model1$resid
## X-squared = 16.546, df = 20, p-value = 0.6822
```

```
Box.test(mode2$resid,t="Ljung",lag=20)
```

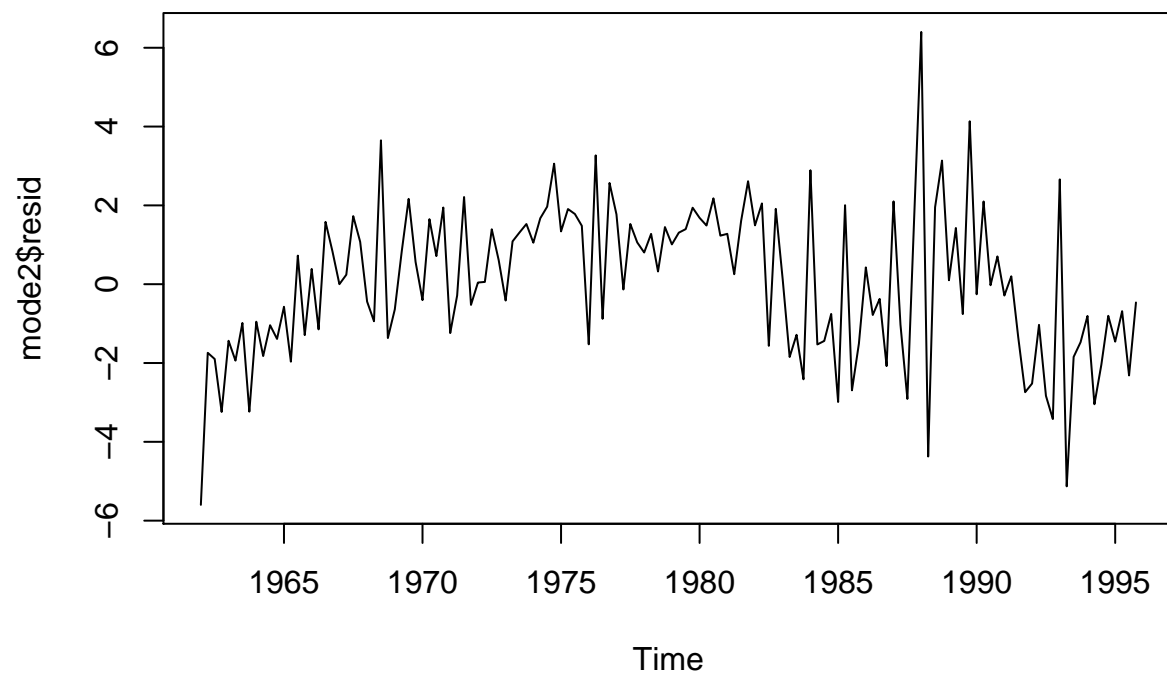
```
##
## Box-Ljung test
##
## data:  mode2$resid
## X-squared = 113.23, df = 20, p-value = 4.996e-15
```

model1 es ruido, pero mode2 no lo es. Analicemos los residuos:

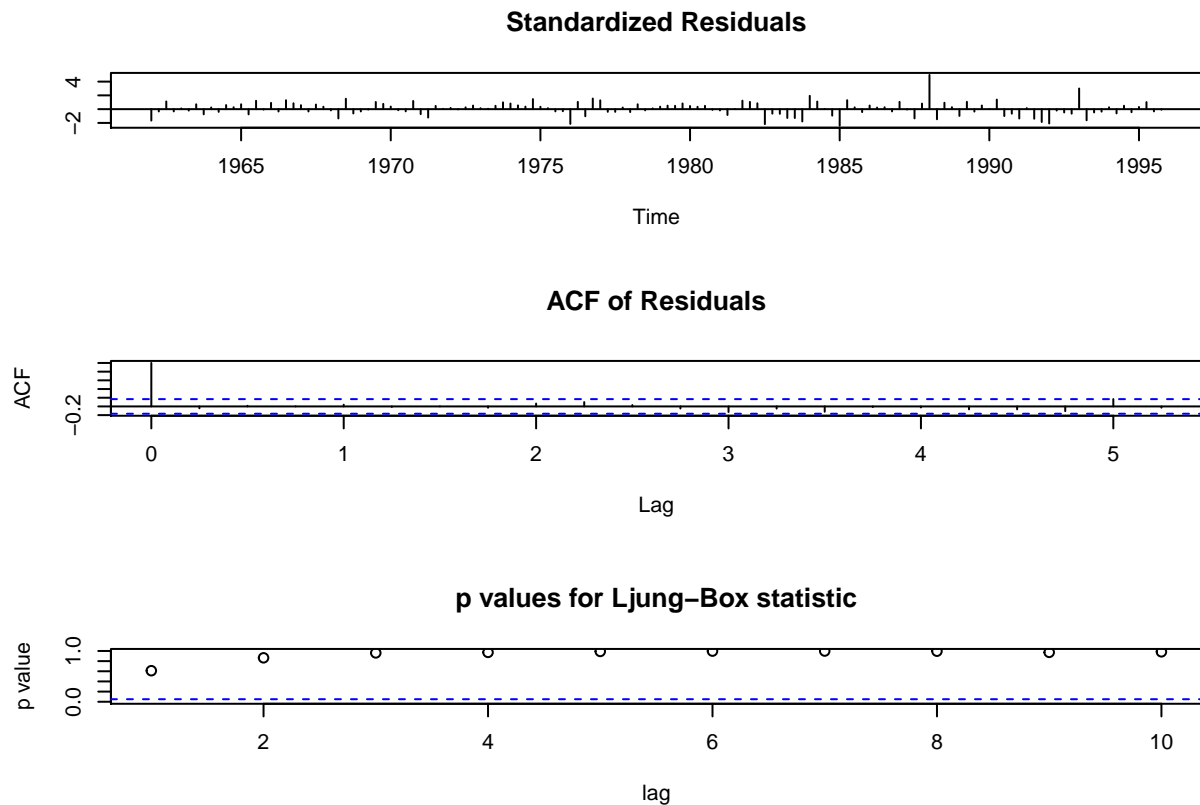
```
ts.plot(model1$resid)
```



```
ts.plot(mode2$resid)
```



```
tsdiag(model1)
```



Probemos un modelo ARMA

```
arma.21 <- arima(emp.ts,order=c(2,0,1))
arma.21$aic
```

```
## [1] 494.8726
```

```
arma.21
```

```
##
```

```
## Call:
```

```
## arima(x = emp.ts, order = c(2, 0, 1))
```

```
##
```

```
## Coefficients:
```

```
##          ar1          ar2          ma1  intercept
```

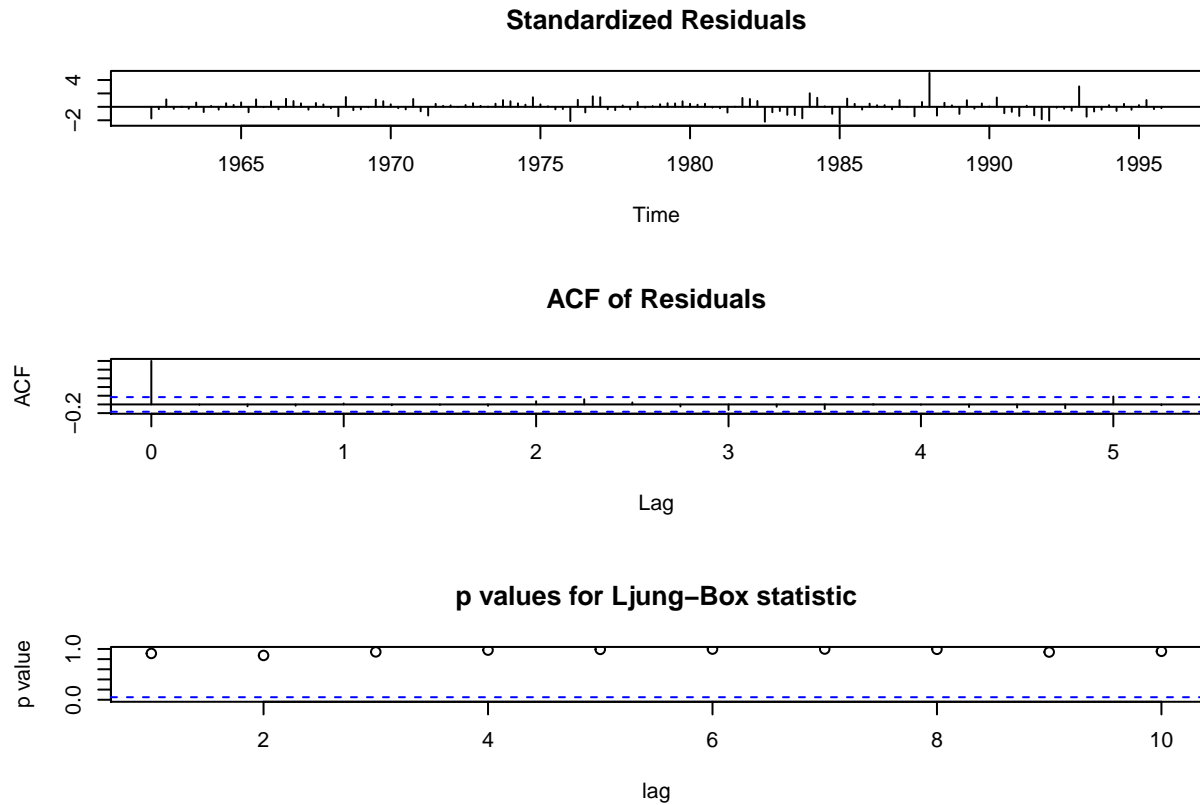
```
##          1.5745 -0.5987 -0.1612    97.9320
```

```
## s.e.  0.1534  0.1522  0.1922    4.0145
```

```
##
```

```
## sigma^2 estimated as 2.011:  log likelihood = -242.44,  aic = 494.87
```

```
tsdiag(arma.21)
```



```
arma.21$coef
```

```
##      ar1      ar2      ma1 intercept
## 1.5744680 -0.5986826 -0.1612365 97.9320375
```

```
arma.21$var.coef
```

```
##              ar1      ar2      ma1  intercept
## ar1      0.02353470 -0.02327060 -0.02640964  0.09975805
## ar2      -0.02327060  0.02316479  0.02602775 -0.11240983
## ma1      -0.02640964  0.02602775  0.03693592 -0.10423546
## intercept 0.09975805 -0.11240983 -0.10423546 16.11644493
```

```
polyroot(c(1,-1.57,0.59)) # Estacionario (Las raíces son |x|>1)
```

```
## [1] 1.056032+0i 1.604985-0i
```

```
polyroot(c(1,-0.16)) # Invertible
```

```
## [1] 6.25+0i
```

```
# Si se cumplen ambas, el proceso ARMA es estacionario.
```

### Condición de invertibilidad del Proceso $MA(q)$

Dado un proceso  $MA(q)$ ,  $Y_t = \theta_q(L)(\epsilon_t)$  donde  $\theta_q(L) = 1 + \theta_1 L + \theta_2 L^2 + \dots + \theta_q L^q$ , entonces considerando el polinomio en  $z \in \mathbb{C}$ ,  $\theta_q(z) = 1 + \theta_1 z + \dots + \theta_q z^q$  y sus  $q$  raíces  $(z_1, z_2, \dots, z_q) \in \mathbb{C}$ , es decir, valores  $z \in \mathbb{C}$  tales que  $\theta_q(z) = 0$ , se dice que el proceso  $Y_t$  es invertible si se cumple

$$|z_j| > 1, \quad \forall j = 1, \dots, q$$

o también, si  $\theta_q(z) \neq 0, \forall z, |z| \leq 1$ . Note que  $()$  es equivalente a



$$\frac{1}{z_j} < 1, \quad \forall j = 1, \dots, q$$

es decir, los inversos de las raíces deben caer dentro del círculo unitario complejo.

## Test de Dickey Fuller

La Prueba de Dickey-Fuller busca determinar la existencia o no de raíces unitarias en una serie de tiempo. La hipótesis nula de esta prueba es que existe una raíz unitaria en la serie.

```
library(tseries)
adf.test(emp.ts)

##
## Augmented Dickey-Fuller Test
##
## data: emp.ts
## Dickey-Fuller = -2.6391, Lag order = 5, p-value = 0.3106
## alternative hypothesis: stationary
adf.test(diff(emp.ts))

##
## Augmented Dickey-Fuller Test
##
## data: diff(emp.ts)
## Dickey-Fuller = -4.0972, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

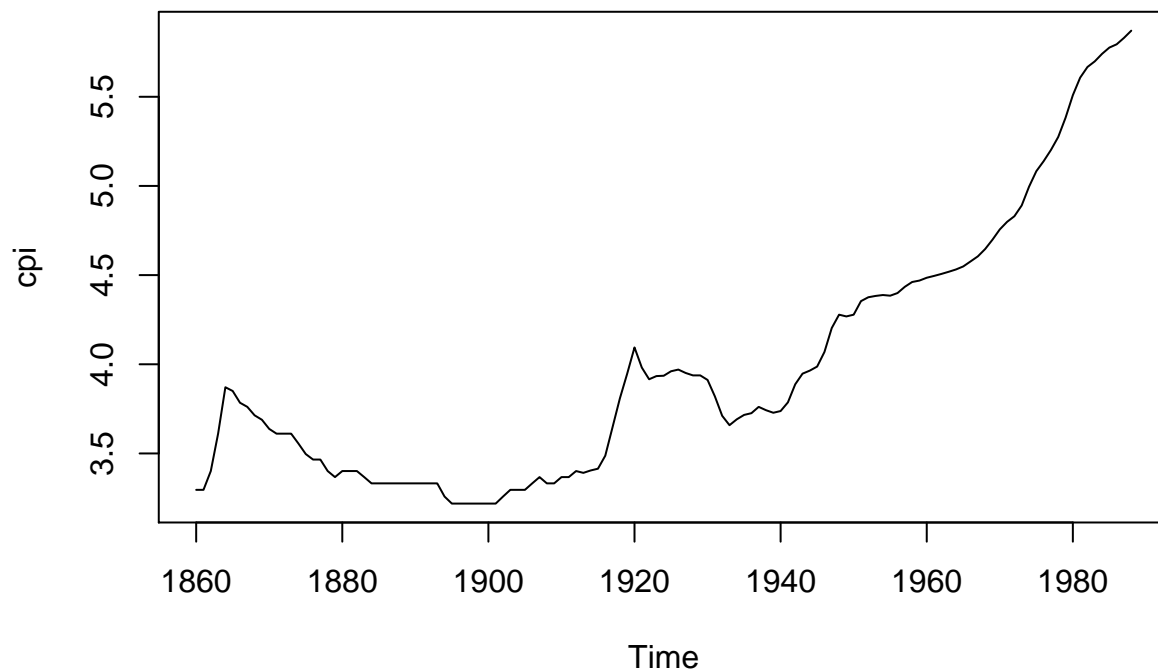
## Ejemplo 3

Datos: 14 series macroeconómicas:

- índice de precios del consumidor (`cpi`),
- producción industrial (`ip`),
- PNB Nominal (`gnp.nom`),
- Velocidad (`vel`),
- Empleo (`emp`),
- Tasa de interés (`int.rate`),
- Sueldos nominales (`nom.wages`),
- Deflactor del PIB (`gnp.def`),
- Stock de dinero (`money.stock`),
- PNB real (`gnp.real`),
- Precios de stock (`stock.prices`),
- PNB per cápita (`gnp.capita`),
- Salario real (`real.wages`), y
- Desempleo (`unemp`).

Tienen diferentes longitudes pero todas terminan en 1988. Trabajaremos con `cpi`

```
data(NelPlo)
plot(cpi)
```



La

serie parece no ser estacionaria ni lineal.

Veamos las raíces unitarias:

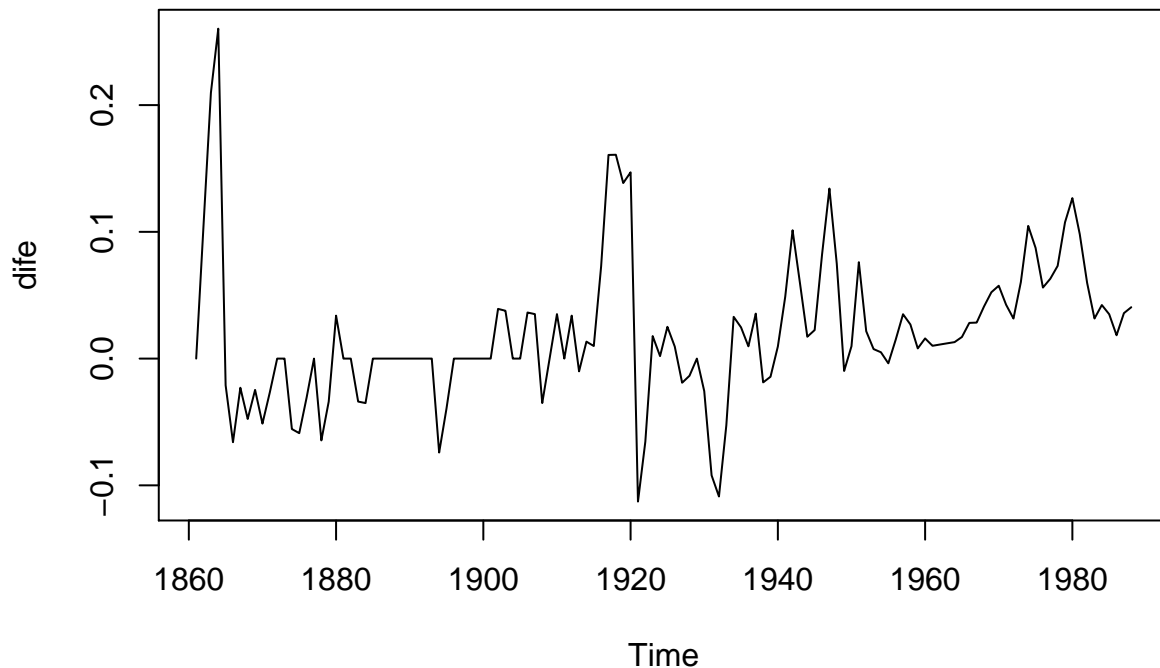
```
adf.test(cpi)
```

```
##
## Augmented Dickey-Fuller Test
##
## data: cpi
## Dickey-Fuller = -1.6131, Lag order = 5, p-value = 0.7374
## alternative hypothesis: stationary
```

¿Es estacionaria?

Probemos con las diferencias

```
dife <- diff(cpi)
plot(dife)
```



```
adf.test(dife)
```

```
##
## Augmented Dickey-Fuller Test
##
## data: dife
## Dickey-Fuller = -4.4814, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

La serie en diferencias si es estacionaria. Veamos qué modelo sugiere R:

```
ar(dife)
```

```
##
## Call:
## ar(x = dife)
##
## Coefficients:
##      1      2      3
## 0.8067 -0.3494 0.1412
##
## Order selected 3  sigma^2 estimated as 0.001875
```

Hasta mi última revisión, no existe una función ma como ar, pero:

```
#### Busquemos el mejor MA ####
```

```
N=10
AICMA=matrix(0,ncol=1,nrow=N)
for (i in 1:N){
  AICMA[i] = arima(diff(cpi),order=c(0,0,i))$aic
}
which.min(AICMA)
```

```
## [1] 3
```

## AICMA

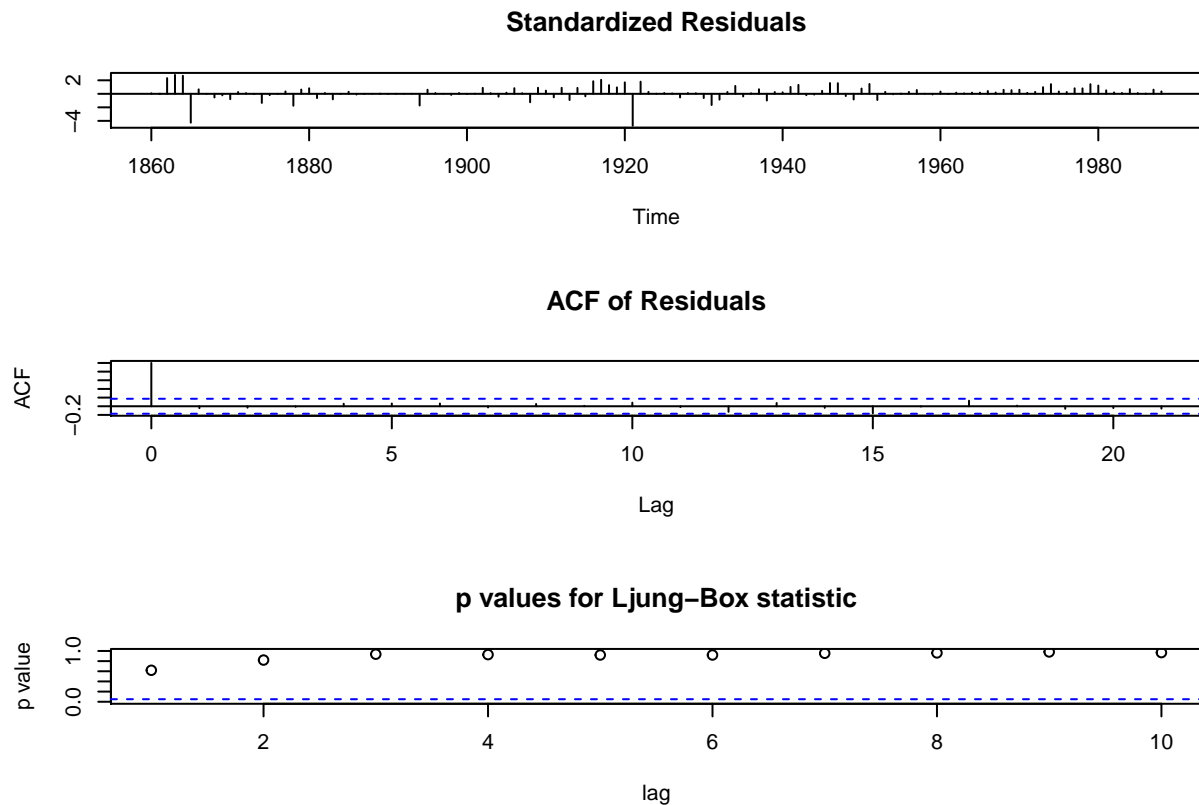
```
##           [,1]
## [1,] -432.7692
## [2,] -435.4208
## [3,] -436.1922
## [4,] -434.4117
## [5,] -432.4410
## [6,] -432.1389
## [7,] -432.3631
## [8,] -430.4594
## [9,] -428.4612
## [10,] -429.8065
```

Se sugiere un  $MA(3)$ .

Evaluemos el modelo MA con una diferencia:

```
model <- arima(cpi,order=c(0,1,3))
model
```

```
##
## Call:
## arima(x = cpi, order = c(0, 1, 3))
##
## Coefficients:
##          ma1      ma2      ma3
##       0.8782  0.3754  0.1898
## s.e.  0.0876  0.1172  0.0890
##
## sigma^2 estimated as 0.00185:  log likelihood = 220.67,  aic = -433.34
tsdiag(model)
```



## Cointegración

Esta sección fue tomada de Colonescu (2016).

- Una serie de tiempo no es estacionaria si su distribución, en particular su media, varianza o covarianza cambian con el tiempo.
- Las series temporales no estacionarias no se pueden usar en los modelos de regresión porque pueden crear una regresión espuria, una relación falsa debido, por ejemplo, a una tendencia común en variables que de otro modo no estarían relacionadas.
- Dos o más series no estacionarias aún pueden ser parte de un modelo de regresión si están cointegradas, es decir, están en una relación estacionaria de algún tipo.

## Estacionariedad

Hasta ahora hemos realizado procedimientos para averiguar si una serie es estacionaria. Recordemos, por ejemplo, el proceso  $AR(1)$ :

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

Este proceso es **estacionario** si  $|\phi| < 1$ ; cuando  $\phi = 1$  el proceso se llama **caminata aleatoria**.

El siguiente código genera procesos  $AR(1)$ :

- con y sin constante
- con y sin tendencia

- $\phi$  menor que 1.

La ecuación genérica para la simulación es:

$$y_t = \alpha - \lambda t + \phi y_{t-1} + \epsilon_t$$

```

N <- 500
a <- 1
l <- 0.01
rho <- 0.7

set.seed(246810)
v <- ts(rnorm(N,0,1))

par(mfrow = c(3,2))

y <- ts(rep(0,N))
for (t in 2:N){
  y[t]<- rho*y[t-1]+v[t]
}
plot(y,type='l', ylab="rho*y[t-1]+v[t]",main= "Sin tendencia")
abline(h=0)

y <- ts(rep(0,N))
for (t in 2:N){
  y[t]<- a+rho*y[t-1]+v[t]
}
plot(y,type='l', ylab="a+rho*y[t-1]+v[t]", main= "Con constante")
abline(h=0)

y <- ts(rep(0,N))
for (t in 2:N){
  y[t]<- a+l*time(y)[t]+rho*y[t-1]+v[t]
}
plot(y,type='l', ylab="a+l*time(y)[t]+rho*y[t-1]+v[t]", main = "Con tendencia y constante")
abline(h=0)

y <- ts(rep(0,N))
for (t in 2:N){
  y[t]<- y[t-1]+v[t]
}
plot(y,type='l', ylab="y[t-1]+v[t]", main = "Caminata aleatoria")
abline(h=0)

a <- 0.1
y <- ts(rep(0,N))
for (t in 2:N){
  y[t]<- a+y[t-1]+v[t]
}
plot(y,type='l', ylab="a+y[t-1]+v[t]", main = "Caminata aleatoria con constante")
abline(h=0)

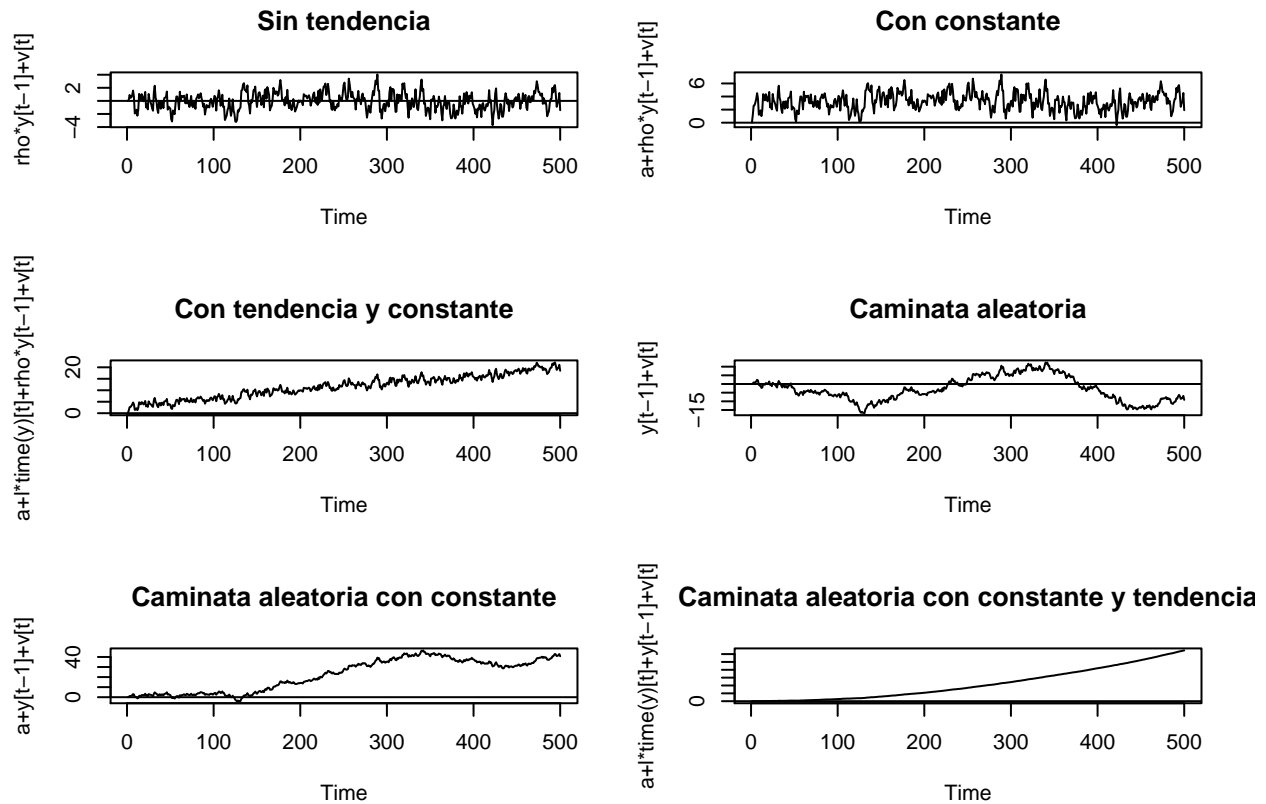
y <- ts(rep(0,N))
for (t in 2:N){

```

```

y[t] <- a + l * time(y)[t] + y[t-1] + v[t]
}
plot(y, type='l', ylab="a+l*time(y)[t]+y[t-1]+v[t]", main = "Caminata aleatoria con constante y tendencia",
abline(h=0))

```



## Regresión espúrea

La no estacionariedad puede conducir a una regresión espuria, una aparente relación entre variables que, en realidad, no están relacionadas. La siguiente secuencia de código genera dos procesos de paseo aleatorio independientes,  $y$  y  $x$ , hacemos la regresión  $y_t = \beta_0 + \beta_1 x_t$ .

Veamos las series y su gráfica de dispersión:

```

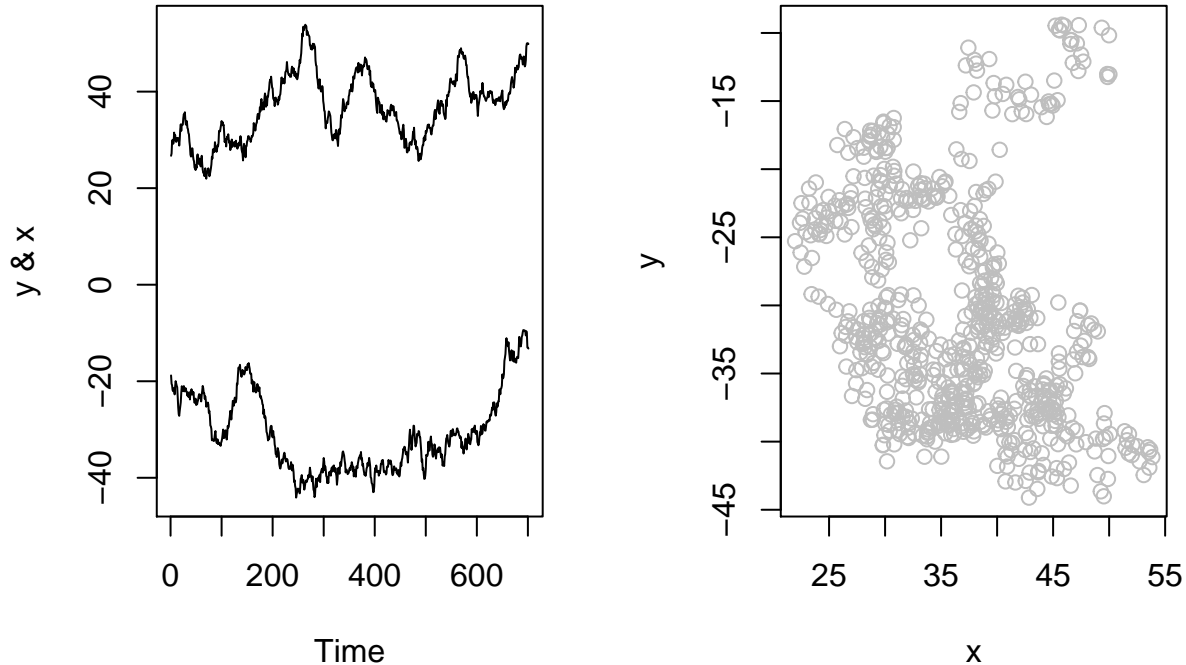
T <- 1000
set.seed(1357)
y <- ts(rep(0,T))
vy <- ts(rnorm(T))
for (t in 2:T){
  y[t] <- y[t-1] + vy[t]
}

set.seed(4365)
x <- ts(rep(0,T))
vx <- ts(rnorm(T))
for (t in 2:T){
  x[t] <- x[t-1] + vx[t]
}
y <- ts(y[300:1000])

```

```
x <- ts(x[300:1000])

par(mfrow = c(1,2))
ts.plot(y,x, ylab="y & x")
plot(x, y, type="p", col="grey")
```



```
spurious.ols <- lm(y~x)
summary(spurious.ols)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.554  -5.973  -2.453   4.508  24.678
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -20.38711    1.61958  -12.588  < 2e-16 ***
## x           -0.28188    0.04331   -6.508  1.45e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.954 on 699 degrees of freedom
## Multiple R-squared:  0.05713,    Adjusted R-squared:  0.05578
## F-statistic: 42.35 on 1 and 699 DF,  p-value: 1.454e-10
```

El resumen muestra una fuerte correlación entre las dos variables, aunque se han generado de forma independiente. (Sin embargo, no es necesario que ninguno de los dos procesos generados aleatoriamente genere una regresión espuria).



## Estacionariedad

Ya hemos trabajado con la prueba de Dickey-Fuller para determinar si una serie es estacionaria. Recuerda que la  $H_o$  de esta prueba es *no estacionariedad*. Es decir, **si rechazamos la hipótesis nula, la serie es estacionaria**.

Un concepto que está estrechamente relacionado con la estacionalidad es el **orden de integración**, que es *cuántas veces necesitamos diferenciar una serie hasta que se vuelva estacionaria*.

Una serie es  $I(0)$ , es decir, integrada de orden 0 si ya es estacionaria (es estacionaria en niveles, no en diferencias); una serie es  $I(1)$  si es no estacionaria en niveles, pero estacionaria en sus primeras diferencias.

## Ejemplo

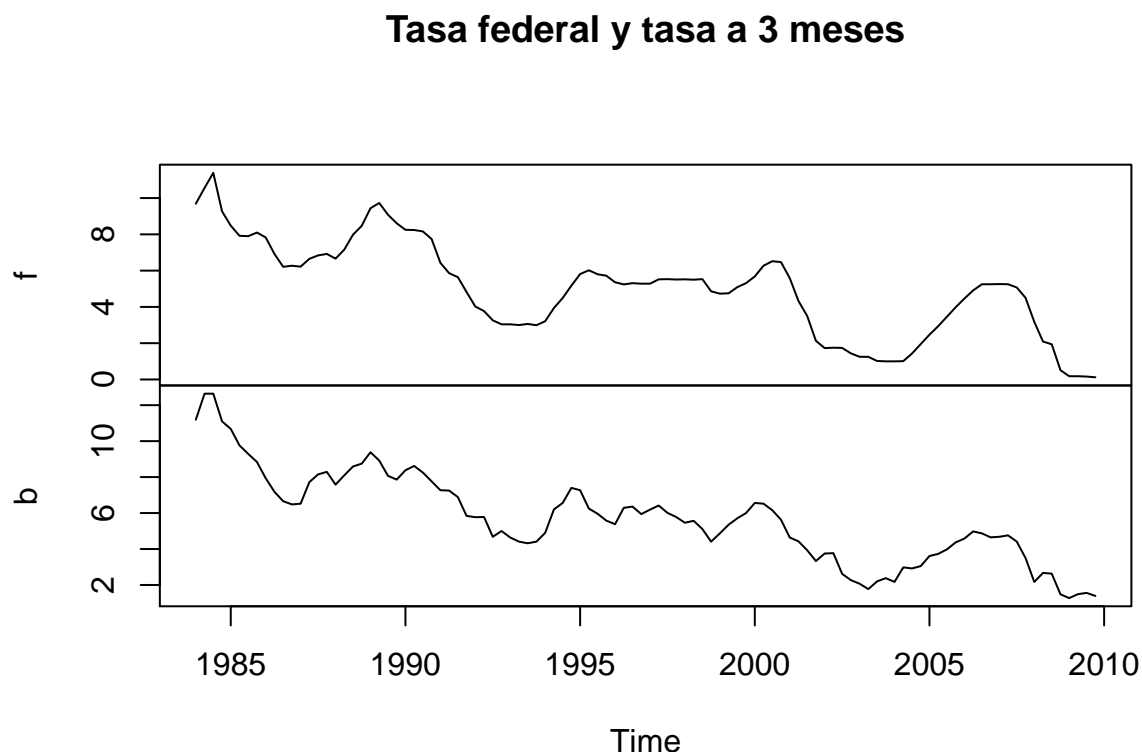
Serie trimestral (1984:1 - 2009:4) de los Estados Unidos de las siguientes variables:

- gdp producto interno bruto
- inf inflacion anual
- f tasa de los fondos federales
- b tasa a tres años

```
dir <- "~/Documents/Consultorias&Cursos/DataLectures"
load(file = paste(dir, "/usa.rda", sep = ""))
usa.ts <- ts(usa, start=c(1984,1), end=c(2009,4),
             frequency=4)
```

Analicemos las tasas: f y b

```
plot.ts(usa.ts[,c("f", "b")], main = "Tasa federal y tasa a 3 meses")
```



Veamos la estacionariedad de las series:

```
adf.test(usa.ts[, "f"], k=10)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: usa.ts[, "f"]  
## Dickey-Fuller = -3.3726, Lag order = 10, p-value = 0.06283  
## alternative hypothesis: stationary
```

```
adf.test(usa.ts[, "b"], k=10)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: usa.ts[, "b"]  
## Dickey-Fuller = -2.9838, Lag order = 10, p-value = 0.1687  
## alternative hypothesis: stationary
```

Las series no son estacionarias. Analicemos sus diferencias

```
adf.test(diff(usa.ts[, "f"]), k=10)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: diff(usa.ts[, "f"])  
## Dickey-Fuller = -3.5291, Lag order = 10, p-value = 0.04291  
## alternative hypothesis: stationary
```

```
adf.test(diff(usa.ts[, "b"]), k=10)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: diff(usa.ts[, "b"])  
## Dickey-Fuller = -3.7181, Lag order = 10, p-value = 0.02601  
## alternative hypothesis: stationary
```

Las series en diferencias son estacionarias.

## Prueba de Phillips-Perron

La hipótesis nula es que la serie tiene una raíz unitaria.

```
# Tets de Phillips Perron
```

```
pp.test(usa.ts[, "f"])
```

```
##  
## Phillips-Perron Unit Root Test  
##  
## data: usa.ts[, "f"]  
## Dickey-Fuller Z(alpha) = -13.21, Truncation lag parameter = 4,  
## p-value = 0.3498  
## alternative hypothesis: stationary
```

```
pp.test(usa.ts[, "b"])
```

```
##
```

```
## Phillips-Perron Unit Root Test
##
## data:  usa.ts[, "b"]
## Dickey-Fuller Z(alpha) = -16.361, Truncation lag parameter = 4,
## p-value = 0.1667
## alternative hypothesis: stationary
```

### Prueba de Cointegración: Phillips-Ouliaris

Peter C. B. Phillips y Sam Ouliaris (1990) muestran que las pruebas de raíz unitaria basadas en residuos aplicadas a los residuos de cointegración estimados no tienen las distribuciones habituales de Dickey-Fuller bajo la **hipótesis nula de no cointegración**.

Debido al fenómeno de regresión espuria bajo la hipótesis nula, la distribución de estas pruebas tiene distribuciones asintóticas que dependen de

1. el número de términos de tendencias deterministas y
2. el número de variables con las que se prueba la cointegración.

Estas distribuciones se conocen como distribuciones de Phillips-Ouliaris.

```
bfx <- as.matrix(cbind(usa.ts[, "f"], usa.ts[, "b"]))
po.test(bfx)
```

```
##
## Phillips-Ouliaris Cointegration Test
##
## data:  bfx
## Phillips-Ouliaris demeaned = -19.877, Truncation lag parameter =
## 1, p-value = 0.05763
```

Las series no son cointegradas

```
fff <- fitted(lm(usa.ts[, "f"] ~ usa.ts[, "b"]))
pp.test(fff)
```

```
##
## Phillips-Perron Unit Root Test
##
## data:  fff
## Dickey-Fuller Z(alpha) = -16.361, Truncation lag parameter = 4,
## p-value = 0.1667
## alternative hypothesis: stationary
adf.test(fff, k = 10)
```

```
##
## Augmented Dickey-Fuller Test
##
## data:  fff
## Dickey-Fuller = -2.9838, Lag order = 10, p-value = 0.1687
## alternative hypothesis: stationary
```

Una relación entre las variables  $I(1)$  cointegradas es una relación a largo plazo, mientras que una relación entre las variables  $I(0)$  es a corto plazo.

## Ejemplo 2

Datos: tasas de cambio mensuales de Estados Unidos, Inglaterra y Nueva Zelanda desde 2004.

```
uu <- "https://raw.githubusercontent.com/vmoprojs/DataLectures/master/us_rates.txt"
datos <- read.csv(url(uu), sep=" ", header=T)
usa.ts <- ts(usa, start=c(1984,1), end=c(2009,4),
             frequency=4)
```

```
# Tasas de cambio, datos mensuales
uk.ts <- ts(datos$UK, st=2004, fr=12)
eu.ts <- ts(datos$EU, st=2004, fr=12)
```

Revisemos si las series son estacionarias:

```
# Tets de Phillips Perron
pp.test(uk.ts)
```

```
##
## Phillips-Perron Unit Root Test
##
## data: uk.ts
## Dickey-Fuller Z(alpha) = -10.554, Truncation lag parameter = 7,
## p-value = 0.521
## alternative hypothesis: stationary
```

```
pp.test(eu.ts)
```

```
##
## Phillips-Perron Unit Root Test
##
## data: eu.ts
## Dickey-Fuller Z(alpha) = -6.812, Truncation lag parameter = 7,
## p-value = 0.7297
## alternative hypothesis: stationary
```

Tienen raíces unitarias

Objetivo: Se piensa que la libra esterlina y el euro tienen alguna relación

```
#Test de Phillips Olliaris
po.test(cbind(uk.ts, eu.ts))
```

```
##
## Phillips-Ouliaris Cointegration Test
##
## data: cbind(uk.ts, eu.ts)
## Phillips-Ouliaris demeaned = -21.662, Truncation lag parameter =
## 10, p-value = 0.04118
```

La  $H_0$ : NO COINTEGRADAS.

Si son cointegradas, es decir que hay una tendencia a largo plazo.

Veamos la relación:

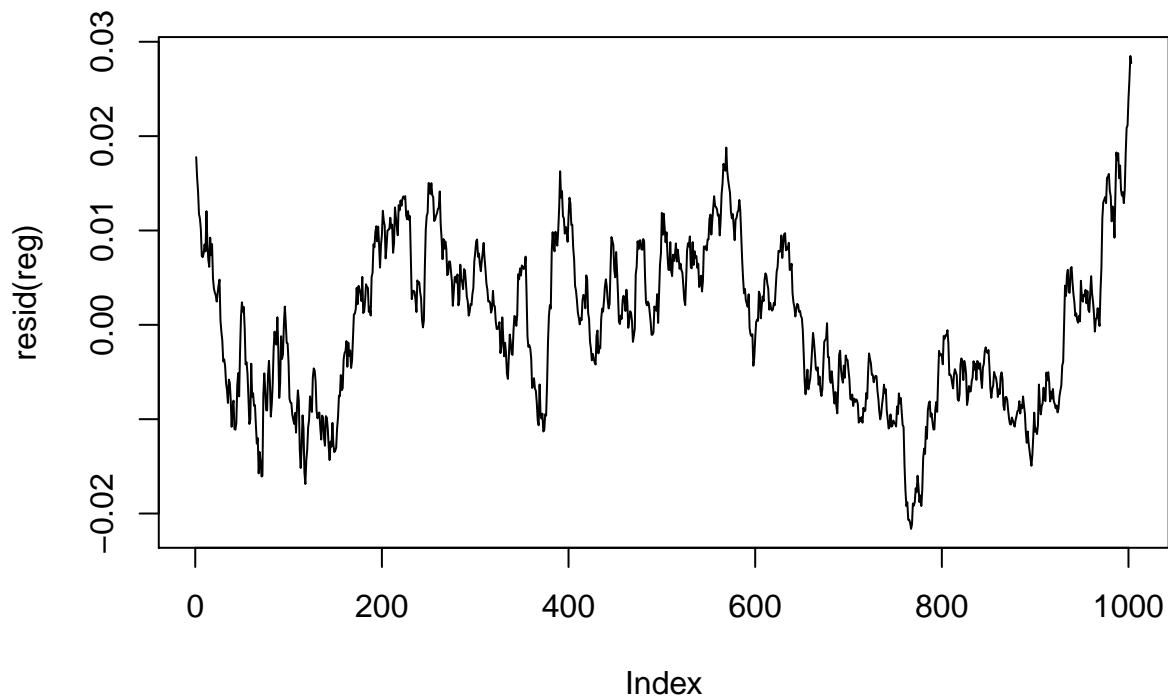
```
reg <- lm(uk.ts~eu.ts)
summary(reg)
```

```
##
## Call:
```

```
## lm(formula = uk.ts ~ eu.ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.0216256 -0.0068351  0.0004963  0.0061439  0.0284938
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.074372   0.004983   14.92  <2e-16 ***
## eu.ts        0.587004   0.006344   92.53  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.008377 on 1001 degrees of freedom
## Multiple R-squared:  0.8953, Adjusted R-squared:  0.8952
## F-statistic: 8561 on 1 and 1001 DF, p-value: < 2.2e-16
```

Analicemos los residuos:

```
residuos = resid(reg)
plot(resid(reg),t="l")
```

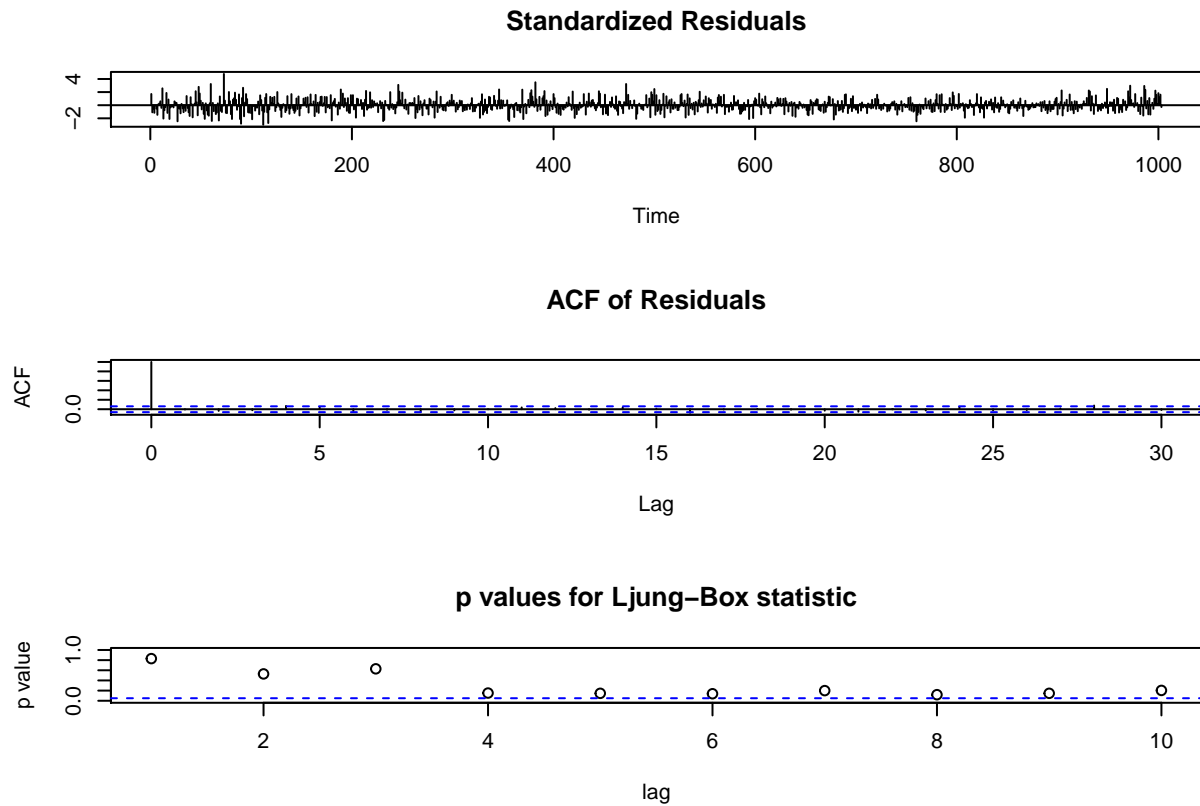


Presentan una estructura, debemos modelizarlos.

```
arma11 <- arima(residuos,order=c(1,0,1))
arma11

##
## Call:
## arima(x = residuos, order = c(1, 0, 1))
##
## Coefficients:
##      ar1      ma1  intercept
##    0.9797  0.1013    0.0015
```

```
## s.e. 0.0072 0.0331 0.0029
##
## sigma^2 estimated as 3.031e-06: log likelihood = 4947.45, aic = -9886.9
tsdiag(arma11)
```



Encontramos un modelo en los errores que si es estacionario, la relación a largo plazo entonces es el coeficiente de la regresión: 0.58.

## Introducción a los modelos de vectores autorregresivos: VAR

Un modelo VAR simple puede ser escrito como:

$$\begin{aligned} y_{1t} &= a_{10} + a_{11}y_{t-1} + a_{12}y_{2t-1} + \epsilon_{1t} \\ y_{2t} &= a_{20} + a_{21}y_{t-1} + a_{22}y_{2t-1} + \epsilon_{2t} \end{aligned}$$

o, en forma matricial

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} a_{1,0} \\ a_{2,0} \end{bmatrix} + \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{bmatrix}$$

donde los términos de error satisfacen

$$\begin{aligned} E(e_{1t}) &= E(e_{2t}) = 0, \forall t \\ E(e_{1t}e_{1s}) &= E(e_{2t}e_{2s}) = E(e_{1t}e_{2s}) = 0, \forall t \neq s \end{aligned}$$

$$\text{Var} \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} = \Sigma, \forall t$$

o, de forma más compacta:

$$y_t = A_0 + A_1 y_t + \epsilon_t$$

donde  $y_t = \begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix}$ ,  $A_1 = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$  y  $\epsilon_t = \begin{bmatrix} e_{1,t} \\ e_{2,t} \end{bmatrix}$

Básicamente, que *todo depende de todo*. Notemos que cada fila puede ser escrita como una ecuación separada:

Mirando un poco más cerca a las ecuaciones individuales, notarás que no aparecen valores contemporáneos (en el tiempo  $t$ ) en el lado derecho (ld) del modelo VAR.

## Ejemplo

Empecemos simulando un proceso VAR (2)

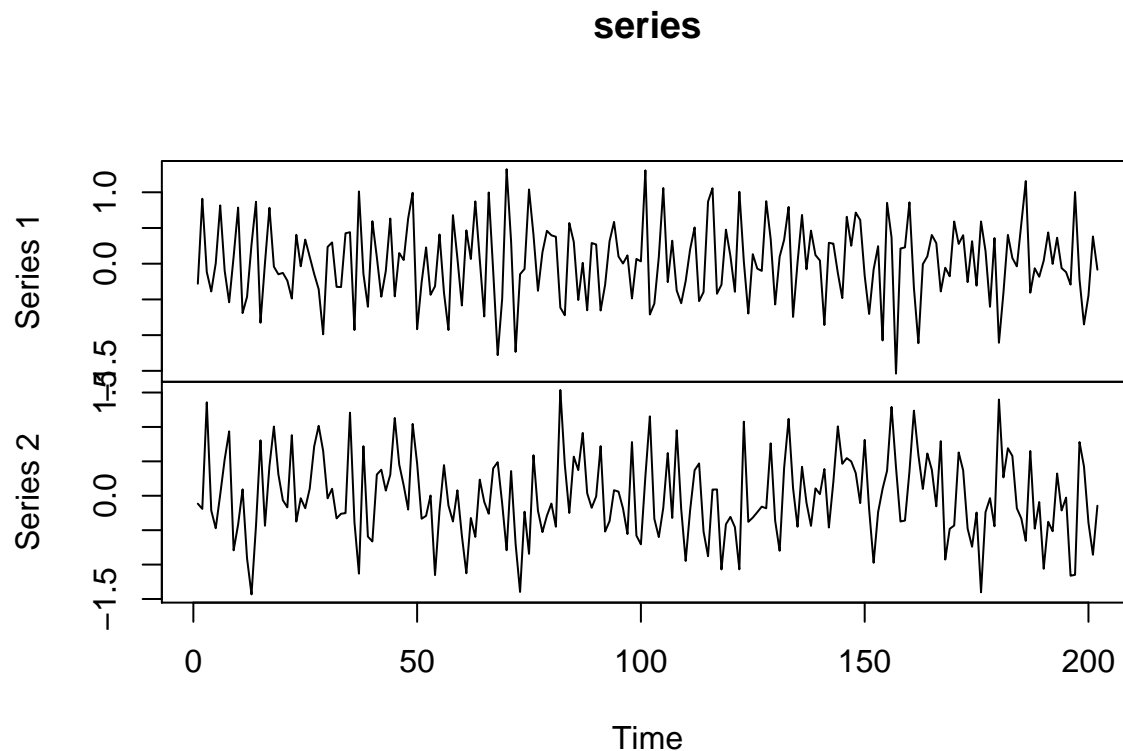
```
set.seed(123) # Misma semilla para tener los mismos resultados

# Generamos muestra
t <- 200 # tamaño de la serie
k <- 2 # Número de variables endógenas
p <- 2 # número de rezagos

# Generamos matriz de coeficientes
A.1 <- matrix(c(-.3, .6, -.4, .5), k) # Matriz de coeficientes del rezago 1
A.2 <- matrix(c(-.1, -.2, .1, .05), k) # Matriz de coeficientes del rezago 2
A <- cbind(A.1, A.2) # Forma compuesta

# Genramos las series
series <- matrix(0, k, t + 2*p) # Inicio serie con ceros
for (i in (p + 1):(t + 2*p)){ # Genramos los errores e ~ N(0,0.5)
  series[, i] <- A.1%*series[, i-1] + A.2%*series[, i-2] + rnorm(k, 0, .5)
}

series <- ts(t(series[, -(1:p)])) # Convertimos a formato ts
names <- c("V1", "V2") # Renombrar variables
plot.ts(series) # Graficos de la serie
```



### Estimación

La función relevante es `VAR` y su uso es directo. Solo tienes que cargar el paquete y especificar los datos (`y`), el orden (`p`) y el tipo del modelo. El tipo de opción determina si se debe incluir un término de intercepción, una tendencia o ambos en el modelo. Como los datos artificiales que hemos generado no contienen términos determinísticos, elegimos no tomarlo en cuenta en la estimación al establecer `type = "none"`.

```
library(vars) # Cargamos el paquete
var.1 <- VAR(series, 2, type = "none") # Estimamos el modelo
```

### Comparación

Un problema central en el análisis de `VAR` es encontrar el número de rezagos que produce los mejores resultados. La comparación de modelos generalmente se basa en criterios de información como el AIC, BIC o HQ. Por lo general, el AIC es preferible a otros criterios, debido a sus características favorables de pronóstico de muestras pequeñas. El BIC y HQ, sin embargo, funcionan bien en muestras grandes y tienen la ventaja de ser un estimador consistente, es decir, converge a los valores verdaderos.

El parámetro clave es `lag.max = 5` en el código siguiente:

```
var.aic <- VAR(series, type = "none", lag.max = 5, ic = "AIC")
```

Veamos los resultados:

```
summary(var.aic)

##
## VAR Estimation Results:
## =====
## Endogenous variables: Series.1, Series.2
## Deterministic variables: none
```



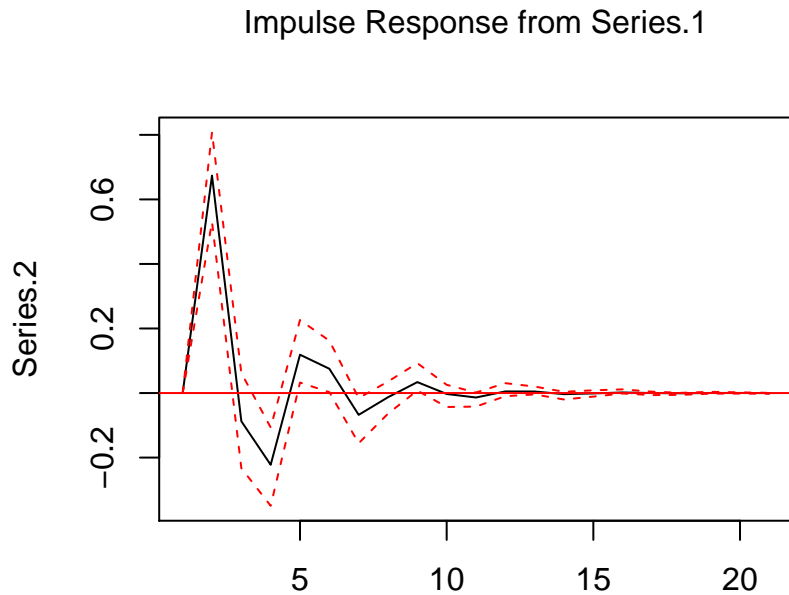
```

## Sample size: 200
## Log Likelihood: -266.065
## Roots of the characteristic polynomial:
## 0.6611 0.6611 0.4473 0.03778
## Call:
## VAR(y = series, type = "none", lag.max = 5, ic = "AIC")
##
##
## Estimation results for equation Series.1:
## =====
## Series.1 = Series.1.l1 + Series.2.l1 + Series.1.l2 + Series.2.l2
##
##           Estimate Std. Error t value Pr(>|t|)
## Series.1.l1 -0.19750    0.06894  -2.865  0.00463 **
## Series.2.l1 -0.32015    0.06601  -4.850  2.51e-06 ***
## Series.1.l2 -0.23210    0.07586  -3.060  0.00252 **
## Series.2.l2  0.04687    0.06478   0.724  0.47018
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.4638 on 196 degrees of freedom
## Multiple R-Squared: 0.2791, Adjusted R-squared: 0.2644
## F-statistic: 18.97 on 4 and 196 DF, p-value: 3.351e-13
##
##
## Estimation results for equation Series.2:
## =====
## Series.2 = Series.1.l1 + Series.2.l1 + Series.1.l2 + Series.2.l2
##
##           Estimate Std. Error t value Pr(>|t|)
## Series.1.l1  0.67381    0.07314   9.213 < 2e-16 ***
## Series.2.l1  0.34136    0.07004   4.874  2.25e-06 ***
## Series.1.l2 -0.18430    0.08048  -2.290  0.0231 *
## Series.2.l2  0.06903    0.06873   1.004  0.3164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.4921 on 196 degrees of freedom
## Multiple R-Squared: 0.3574, Adjusted R-squared: 0.3443
## F-statistic: 27.26 on 4 and 196 DF, p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##           Series.1 Series.2
## Series.1  0.21417 -0.03116
## Series.2 -0.03116  0.24154
##
## Correlation matrix of residuals:
##           Series.1 Series.2
## Series.1    1.000  -0.137
## Series.2  -0.137    1.000

```

### Impulso respuesta

```
ir.1 <- irf(var.1, impulse = "Series.1", response = "Series.2", n.ahead = 20, ortho = FALSE)
plot(ir.1)
```



### 95 % Bootstrap CI, 100 runs

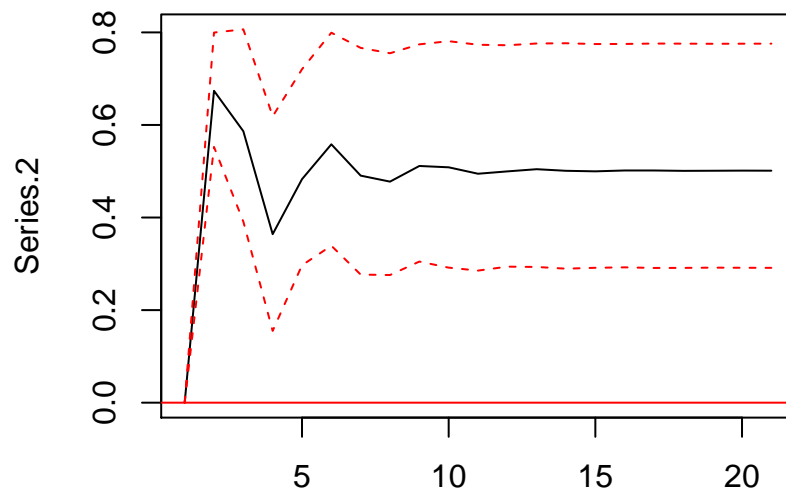
La opción `ortho` es importante, porque dice algo sobre las relaciones contemporáneas entre las variables. En nuestro ejemplo, ya sabemos que tales relaciones no existen, porque la matriz verdadera de varianza-covarianza, o simplemente matriz de covarianza, es diagonal con ceros en los elementos fuera de la diagonal.

Sin embargo, dado que los datos de series temporales limitadas con 200 observaciones restringen la precisión de las estimaciones de los parámetros, la matriz de covarianza tiene valores positivos en sus elementos fuera de la diagonal, lo que implica efectos contemporáneos distintos de cero de un choque. Para descartar esto establezco `ortho = FALSE`. El resultado de esto es que la respuesta al impulso comienza en cero en el período 0. También puedes probar la alternativa y establecer `ortho = TRUE`, lo que da como resultado un gráfico que comienza bajo cero. No quiero entrar en más detalles aquí, pero basta decir que el problema de los llamados errores ortogonales es uno de los problemas centrales en el análisis de VAR y definitivamente deberías leer más al respecto, si tienes la intención de configurar tus propios modelos VAR.

A veces es importante obtener el efecto a largo plazo. Para esto se calcula el gráfico de la función impulso-respuesta acumulada:

```
ir.2 <- irf(var.1, impulse="Series.1", response="Series.2", n.ahead = 20, ortho = FALSE,
cumulative = TRUE)
plot(ir.2)
```

Impulse Response from Series.1 (cumulative)



95 % Bootstrap CI, 100 runs

Vemos que, pese a variaciones negativas en algunos períodos, en el largo plazo el efecto es positivo.

## Referencias

Colonescu, Constantin. 2016. "Principles of Econometrics with R."