Series Temporales y Predicción

Práctica 3B

Simulación y identificación de procesos AR y MA

Solución orientativa a los ejercicios propuestos

Práctica 1.1

Consideramos un proceso AR(1) dado por $X_t = \alpha \cdot X_{t-1} + E_t$. Simula un proceso como el anterior (por ejemplo, n = 1.000 observaciones) y representa las autocorrelaciones y autocorrelaciones parciales. ¿Cómo cambian si cambia el valor de α?

En cada caso, mira qué modelo propone la función auto.arima(). ¿Es razonable?

Solución orientativa

Código R

```
nobs <- 10000
serie <- vector()
serie[1] <- 0
alpha <- 0.6
for (i in 2:nobs)
{
  serie[i] <- alpha*serie[i-1]+rnorm(1)
}
plot(serie)
par(mfrow = c(2,1))
acf(serie)
pacf(serie)
fitarima = auto.arima(serie, allowdrift=FALSE)
fitarima
coeftest(fitarima)
```

IMPORTANTE: Antes de simular recordad que $|\phi|$ ha de ser menor que 1 para que la serie sea estacionaria. Podéis realizar una prueba de cómo evoluciona una serie con $|\phi|$ mayor que 1.

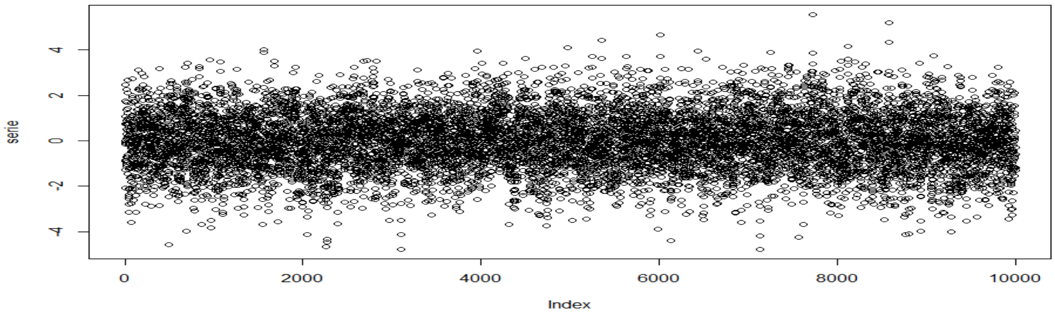
Caso 1

ϕ = 0,6

Modelo teórico

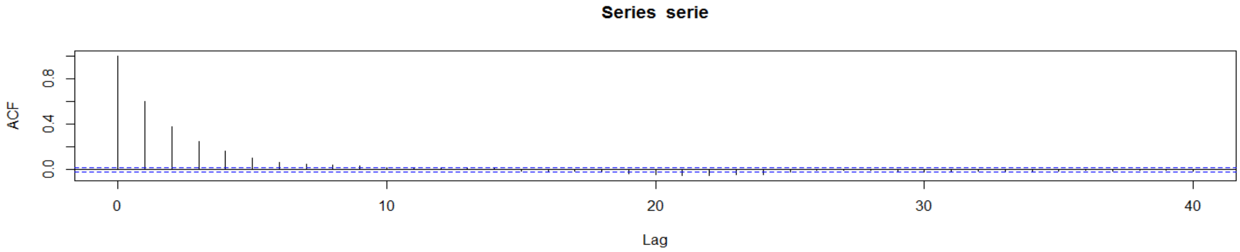
$X_t = 0.6 \cdot X_{t-1} + E_t$

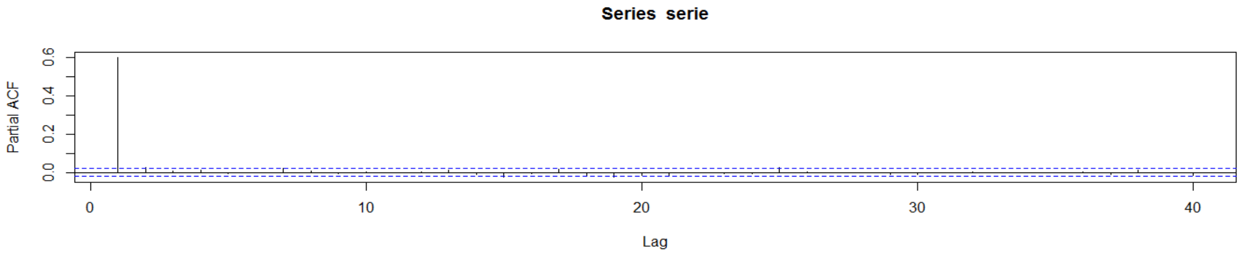
La serie que obtenemos de la simulación está centrada en el 0, sin tendencia ni estacionalidad, con una varianza manteniéndose constante a lo largo del tiempo. Por lo que estamos ante una serie estacionaria.



En el correlograma (gráfica ACF) observamos que ρ(1) = 0.6, ρ(k) = 0.6^k. Por lo que vemos como la función desciende potencialmente a cero a medida que va aumentando el lag.

En la función de autocorrelación parcial observamos como el primer lag toma el valor 0.6, α(1) = 0.6, mientras que el resto de lags toma valores muy cercanos a cero (dentro de las dos bandas).





En nuestra simulación la función autoarima() nos propone:

```
> auto.arima(serie, allowdrift=FALSE)
Series: serie
ARIMA(1,0,1) with zero mean

Coefficients:
         ar1      ma1
      0.6327  -0.0461
s.e.  0.0129   0.0167

sigma^2 estimated as 1.017:  log likelihood=-14273.53
AIC=28553.05   AICc=28553.05   BIC=28574.68
> fitarima = auto.arima(serie, allowdrift=FALSE)
> coeftest(fitarima)

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ar1  0.632682   0.012887 49.0952 < 2.2e-16 ***
ma1 -0.046053   0.016695 -2.7584  0.005808 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

descomponer la serie en un AR(1) con coeficiente 0.6327 más una media móvil con coeficiente -0.0461. En fórmula seria:

Xt = 0.6327 · Xt−1 - 0.0461 · Et-1 + Et

donde Et es ruido blanco.

Ahora bien, si miramos el test sobre los coeficientes, el coeficiente que acompaña al componente autoregresivo (0,0637) es significativamente distinto a 0, puesto que el p-valor es < 2.2.e-16. Mientras que el coeficiente que acompaña a la media móvil (-0.046053) se podría considerar casi nulo con un nivel de significación de 0.001, puesto que el p-valor es 0.005808 > 0.001.

Si anuláramos la media móvil del modelo, el resultado sería:

```
arima1<-arima(serie,c(1,0,0))
arima1
coeftest(arima1)
```

```
> arima1<-arima(serie,c(1,0,0))
> arima1

Call:
arima(x = serie, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.6034     0.0050
s.e.  0.0080     0.0254

sigma^2 estimated as 1.018:  log likelihood = -14277.3,  aic = 28560.6
> coeftest(arima1)

z test of coefficients:

           Estimate Std. Error z value Pr(>|z|)
ar1       0.6034208  0.0079682  75.729   <2e-16 ***
intercept 0.0050107  0.0254339   0.197   0.8438
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$X_t = 0.0050107 + 0.6034 \cdot X_{t-1} + E_t$

donde Et es ruido blanco.

Nuevamente el coeficiente autoregresivo nos da significativamente distinto a 0 (p-valor < 2e-16), pero nos ha aparecido un coeficiente constante (0.0050107) “igual a 0”. Veámoslo:

Hipótesis nula: Coeficiente (intercept) = 0;
Hipótesis nula: Coeficiente (intercept) ≠ 0

Puesto que p-valor = 0.8438 > 0.05 entonces aceptamos la hipótesis nula; por lo que aceptamos que el coeficiente es igual a 0.

Quitemos la constante:

```
arima2<-arima(serie,c(1,0,0),  include.mean = FALSE)
arima2
coeftest(arima2)
```

```
Call:
arima(x = serie, order = c(1, 0, 0), include.mean = FALSE)

Coefficients:
         ar1
      0.6034
s.e.  0.0080

sigma^2 estimated as 1.018:  log likelihood = -14277.32,  aic = 28558.64
> coeftest(arima2)

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ar1 0.6034271  0.0079681   75.73 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$X_t = 0.6034271 \cdot X_{t-1} + E_t$

donde Et es ruido blanco.

Observamos que el AIC del primer modelo propuesto por la auto.arima tenía un AIC = 28.553,03 que no está tan lejos del AIC del modelo autoregresivo que acabamos de proponer (AIC = 28.558,64).

Teniendo en cuenta el principio de parsimonia (el modelo más simple es el mejor, explicando casi lo mismo) el modelo autorregresivo $X_t = 0.6034271 \cdot X_{t-1} + E_t$ sería adecuado para modelar nuestra simulación.

Caso 2

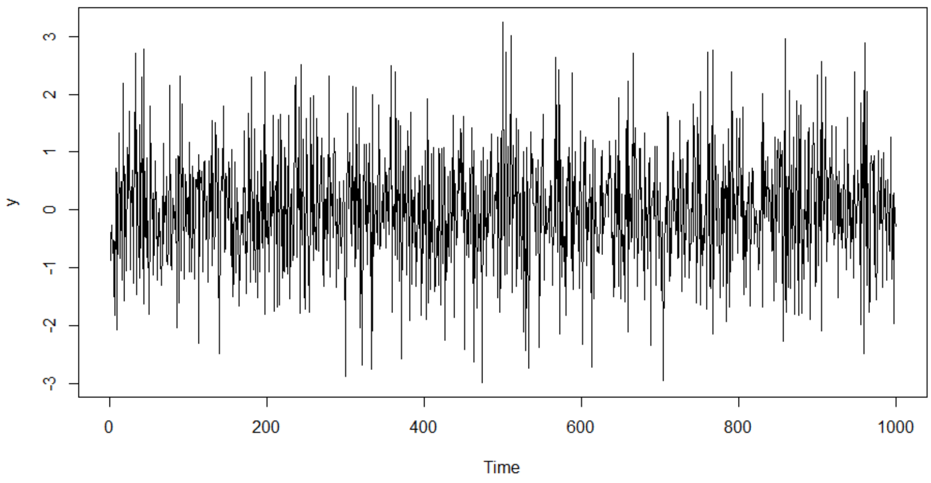
ϕ = - 0,3

Modelo teórico

$X_t = -0.3 \cdot X_{t-1} + E_t$

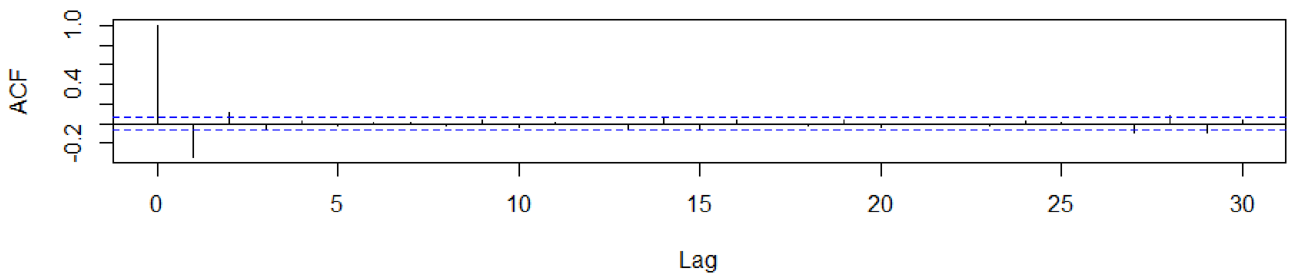
```
alpha <- -0.3
y <- arima.sim(n = 10000, list(ar = alpha), innov=rnorm(10000))
par(mfrow = c(1,1))
plot(y)
par(mfrow = c(2,1))
acf(y)
pacf(y)
fitarima = auto.arima(y, allowdrift=FALSE)
fitarima
coeftest(fitarima)
```

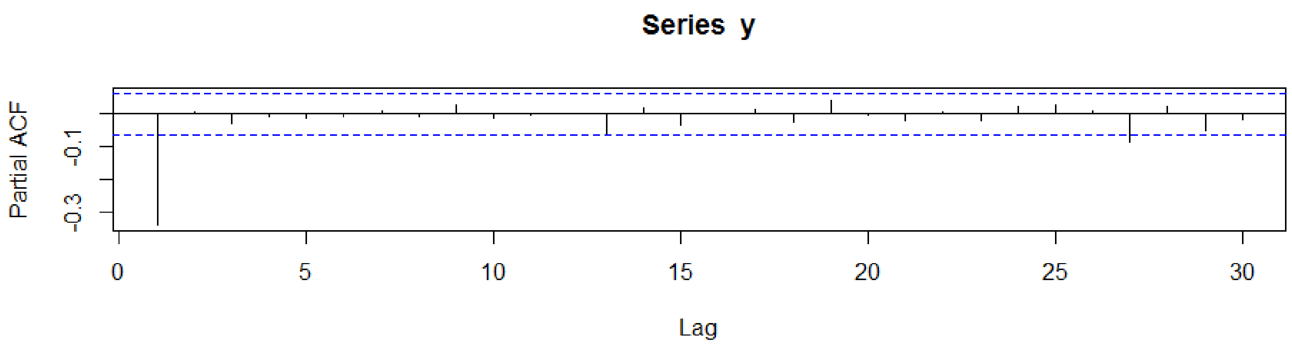
La serie que obtenemos de la simulación está centrada en el 0, sin tendencia ni estacionalidad, con una varianza manteniéndose constante a lo largo del tiempo. Por lo que estamos ante una serie estacionaria.



En el correlograma (gráfica ACF) observamos que $\rho(1) = -0.3$, $\rho(k) = (-0.3)^k$. Por lo que vemos como la función muestra picos que se van alternando de positivo a negativo desciendo potencialmente a cero a medida que va aumentando el lag.

En la función de autocorrelación parcial observamos como el primer lag toma el valor -0.3, $\alpha(1) = -0.3$, mientras que el resto de lags toma valores muy cercanos a cero (dentro de las dos bandas).





En nuestra simulación, la función autoarima() nos propone:

```
> pacf(y)
> fitarima = auto.arima(y, allowdrift=FALSE)
> fitarima
Series: y
ARIMA(2,0,1) with zero mean

Coefficients:
         ar1     ar2      ma1
      0.5540  0.2405  -0.8550
s.e.  0.0864  0.0303   0.0849

sigma^2 estimated as 0.9987:  log likelihood=-14181.4
AIC=28370.8   AICc=28370.8   BIC=28399.64
> coeftest(fitarima)

z test of coefficients:

     Estimate Std. Error  z value  Pr(>|z|)
ar1  0.553964   0.086380   6.4131 1.426e-10 ***
ar2  0.240451   0.030261   7.9459 1.928e-15 ***
ma1 -0.855031   0.084924 -10.0681 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

descomponer la serie en un AR(2) más una media móvil con coeficiente -0.855031. En fórmula seria:

Xt = 0,553964 · Xt−1 + 0,240451 Xt-2 - 0.855031 · Et-1 + Et

dónde Et es ruido blanco.

En este caso el test sobre los coeficientes nos da todos ellos significativamente distinto a 0.

Por otro lado, si calculamos un modelo autoregresivo de orden 1 obtenemos:

```
arima2<-arima(y,c(1,0,0),  include.mean = FALSE)
arima2
coeftest(arima2)
```

```
> arima2<-arima(y,c(1,0,0),  include.mean = FALSE)
> arima2

Call:
arima(x = y, order = c(1, 0, 0), include.mean = FALSE)

Coefficients:
          ar1
      -0.2967
s.e.   0.0096

sigma^2 estimated as 0.999:  log likelihood = -14184.56,  aic = 28373.12
> coeftest(arima2)

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ar1 -0.296657   0.009552 -31.057 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Xt = -0,2967 · Xt−1 + Et

donde Et es ruido blanco

Además el coeficiente autoregresivo de orden 1 (-0,296657) nos da significativamente distinto a 0 con un AIC de 28373,12, no muy lejano al AIC que nos ofrecía el modelo propuesto por la función auto.arima() de 28370,8.

Teniendo en cuenta el principio de parsimonia (el modelo más simple es el mejor) el modelo autorregresivo $X_t = -0{,}2967 \cdot X_{t-1} + E_t$ sería adecuado para modelar nuestra simulación.

Práctica 1.2

Consideramos un proceso MA(1) dado por $X_t = Z_t + \theta \cdot Z_{t-1}$, donde $Z_t \sim N(0, \sigma^2)$. Simula un proceso como el anterior (por ejemplo, n = 1.000 observaciones) y representa las autocorrelaciones y autocorrelaciones parciales. ¿Cómo cambian si cambia el valor de α?

En cada caso, mira qué modelo propone la función auto.arima(). ¿Es razonable?

Solución orientativa

Código R

```
### MA(1) process
nobs <- 1000
serie <- vector()
zt    <- vector()
serie[1] <- 0
zt[1]    <- 0
alpha <- -0.8
for (i in 2:nobs)
{
  zt[i]    <- rnorm(1)
  serie[i] <- zt[i]+alpha*zt[i-1]
}
par(mfrow = c(1,1))
plot(serie)
par(mfrow = c(2,1))
acf(serie)
pacf(serie)
fitarima = auto.arima(serie, allowdrift=FALSE)
fitarima
coeftest(fitarima)
```

Caso 1

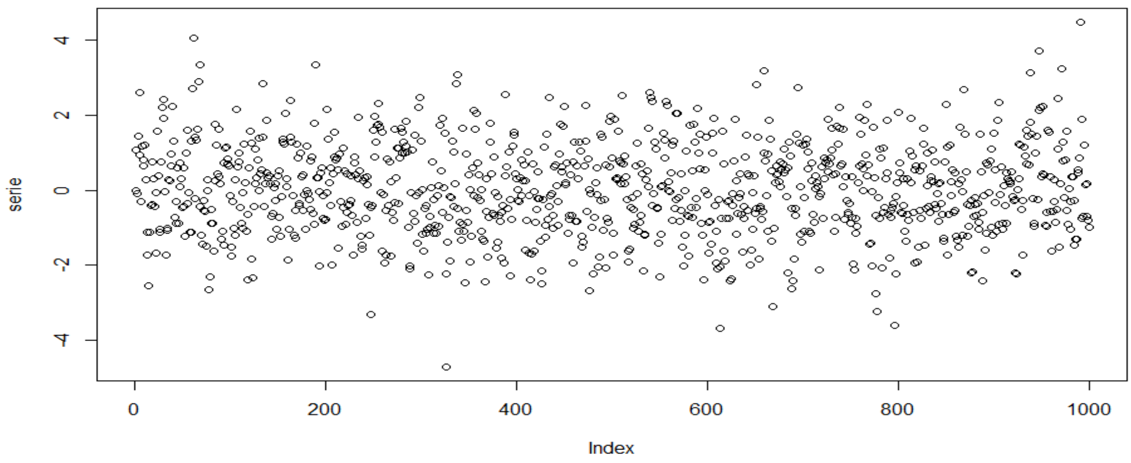
θ = 0,6

Modelo teórico

$X_t = 0.6 \cdot E_{t-1} + E_t$

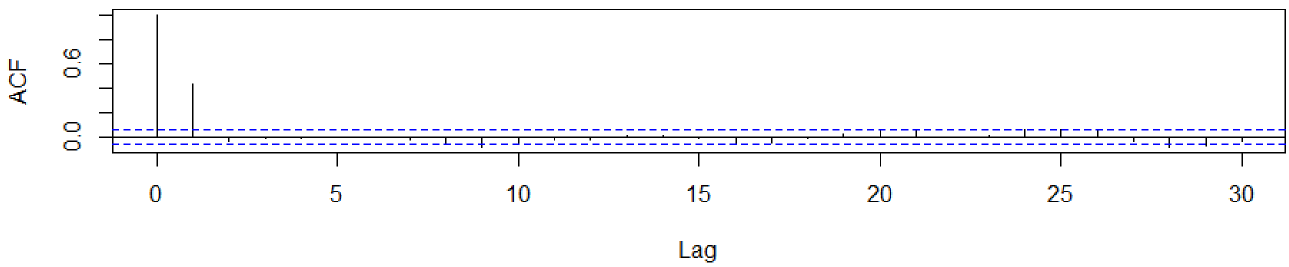
Donde E_t es el ruido blanco.

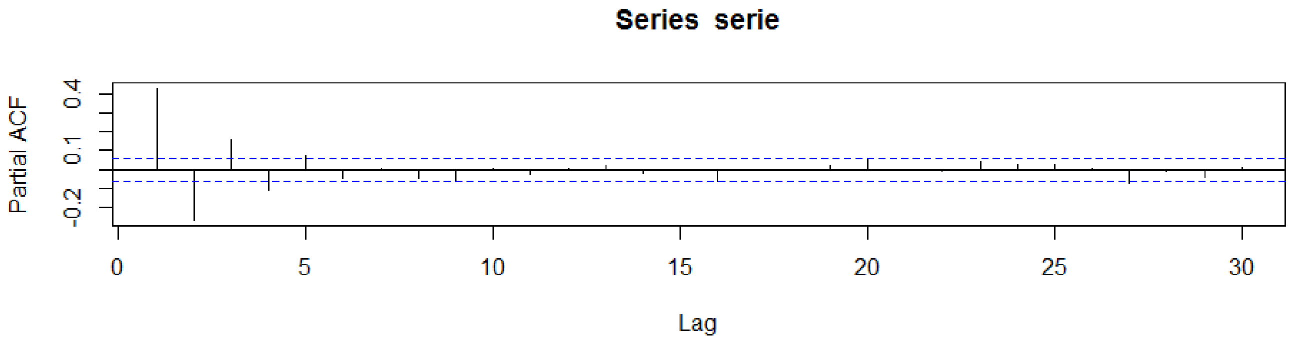
La serie que obtenemos de la simulación está centrada en el 0, sin tendencia ni estacionalidad, con una varianza manteniéndose constante a lo largo del tiempo. Por lo que estamos ante una serie estacionaria.



En el correlograma (gráfica ACF) observamos como el primer lag toma un valor distinto a 0 y positivo mientras que el resto de lags toma valores muy cercanos a cero (dentro de las dos bandas).

En la función de autocorrelación parcial (gráfica PACF) observamos que α (1) = $-\theta/(1+\theta^2)$ = $-0,6/(1+0,6^2)$ = -0,44 negativo y diferente de 0 y como la función desciende potencialmente a cero a medida que va aumentando el lag.





En nuestra simulación la función autoarima() nos propone:

```
> fitarima = auto.arima(serie, allowdrift=FALSE)
> fitarima
Series: serie
ARIMA(0,0,1) with zero mean

Coefficients:
         ma1
      0.6334
s.e.  0.0246

sigma^2 estimated as 1.077:  log likelihood=-1455.59
AIC=2915.17   AICc=2915.19   BIC=2924.99
> coeftest(fitarima)

z test of coefficients:

    Estimate Std. Error z value  Pr(>|z|)
ma1 0.633427   0.024575  25.775 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

descomponer la serie en un MA(1) con coeficiente 0,6334. En fórmula seria:

Xt = 0.6334 · Et−1 + Et

donde Et es ruido blanco.

Además el test del coeficiente de orden 1 de la media móvil resulta significativamente distinto 0

Hipótesis nula: Coeficiente (ma1) = 0;
Hipótesis nula: Coeficiente (ma1) ≠ 0

Puesto que p-valor <2.2e-16 < 0.01 entonces rechazamos la hipótesis nula; por lo que rechazamos que el coeficiente es igual a 0.

Caso 2

ϕ = - 0,5

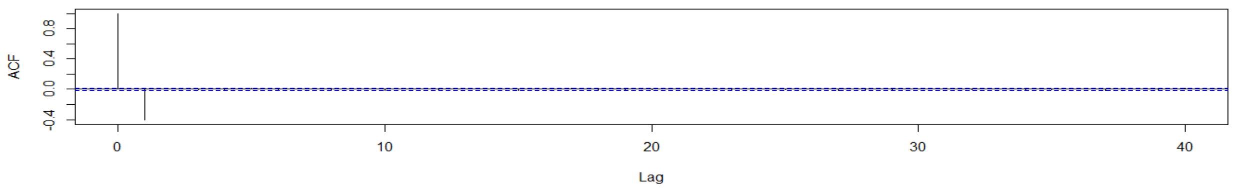
Modelo teórico

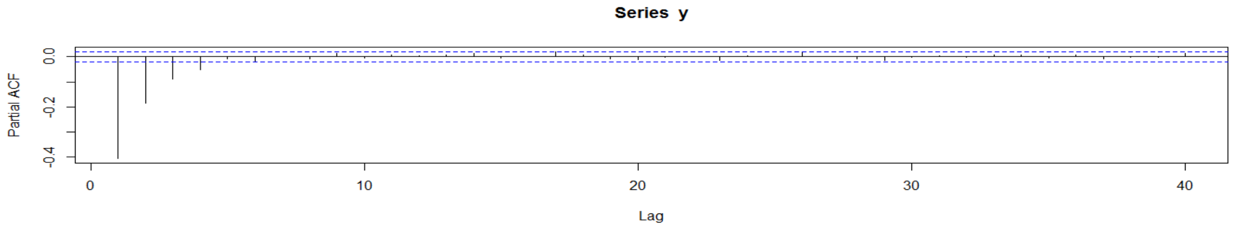
Xt = -0.5 · Xt−1 + Et

```
alpha <- -0.5
y <- arima.sim(n = 10000, list(ma = alpha), innov=rnorm(10000))
par(mfrow = c(1,1))
plot(y)
par(mfrow = c(2,1))
acf(y)
pacf(y)
fitarima = auto.arima(y, allowdrift=FALSE)
fitarima
coeftest(fitarima)
```

En el correlograma (gráfica ACF) de la serie simulada observamos como el primer lag toma un valor negativo y distinto a 0, mientras que el resto de lags toma valores muy cercanos a cero (dentro de las dos bandas).

En la función de autocorrelación parcial (gráfica PACF) observamos que la función toma valores negativos que van acercándose potencialmente a cero a medida que va aumentando el lag.





En nuestra simulación, la función auto.arima() nos propone:

```
> fitarima
Series: y
ARIMA(0,0,1) with zero mean

Coefficients:
          ma1
      -0.4966
s.e.   0.0086

sigma^2 estimated as 1.013:  log likelihood=-14255.76
AIC=28515.52   AICc=28515.53   BIC=28529.95
> coeftest(fitarima)

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)
ma1 -0.4966077  0.0085926 -57.795 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

descomponer la serie en un MA(1) con coeficiente -0.4966. En fórmula seria:

Xt = - 0.4966 · Et-1 + Et

donde Et es ruido blanco.

En este caso el test sobre el coeficiente nos da que resulta ser significativamente distinto a 0.

A tener en cuenta…

Informalmente de estos dos ejercicios hemos aprendido:

La forma de **acf ()** para definir valores de p y q es mirar los gráficos y repasando la tabla podemos determinar qué tipo de modelo seleccionar y cuáles serán los valores de p, d y q.

Forma	Modelo
Serie exponencial decayendo a 0	Modelo Auto Regresivo (AR). Función pacf () que se utilizará para identificar el orden del modelo
Picos alternativos positivos y negativos, decayendo a 0	Modelo Auto Regresivo (AR). Función pacf () que se utilizará para identificar el orden del modelo
Uno o más picos en serie, resto todos son 0	Modelo de media móvil (MA), identifica el orden donde el gráfico se convierte en 0
Después de algunos retrasos en general la serie va decayendo.	Modelo mezclado AR & MA
La serie total es 0 o casi 0	Datos aleatorios
Valores medios a intervalos fijos	Necesitamos incluir el término AR de estacionalidad
Picos visibles que no decaen a 0	Series no son estacionarias

Enlace de interés: https://www.diegocalvo.es/analisis-de-series-temporales-en-r-arima/

Práctica 2.1

El fichero *prac3TS.txt* disponible en el Campus Virtual de la asignatura contiene datos relacionados con la magnitud de terremotos alrededor del mundo, desde 1916 hasta 2015. Se pide:

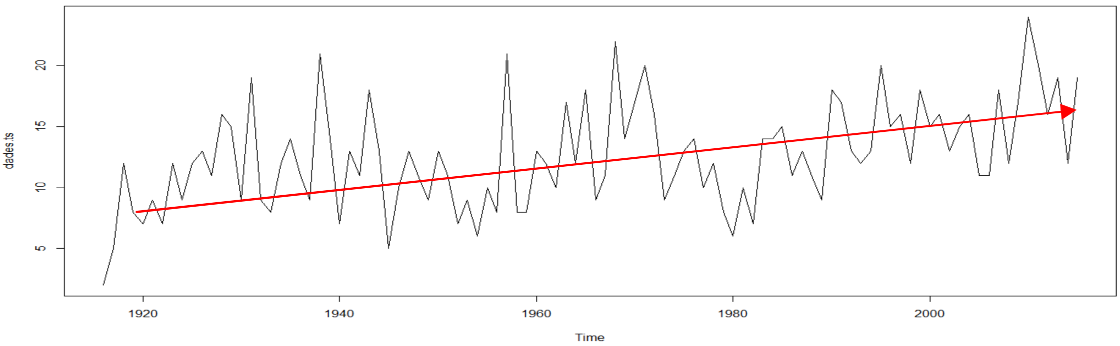
Solución orientativa:

i. Convierte los datos en un objeto serie temporal a R y realiza un gráfico.

Código R

```
library(zoo)
library(forecast)

dades <- read.table("/Users/POR740051/Desktop/UOC/SeriesTemporalsUAB/prac3TS.txt", header=T)
start <- as.yearmon("1916-01-01")
end   <- as.yearmon("2015-01-01")
dades.ts <- ts(dades$Quakes, start=start, end=end, frequency=1)
ts.plot(dades.ts)
```

ii. ¿Tiene tendencia?
Se detecta una ligera tendencia ascendente en el tiempo sobretodo en la parte final de la gráfica.



iii. ¿Tiene estacionalidad?
Dando un vistazo a la gráfica no se detecta una estacionalidad destacada.

iv. Realiza la regresión lineal de la serie respecto el tiempo. ¿Cuál es la bondad del ajuste?

Código R

```
### Treure tendencia
t <- 1:length(dades$Quakes)
```

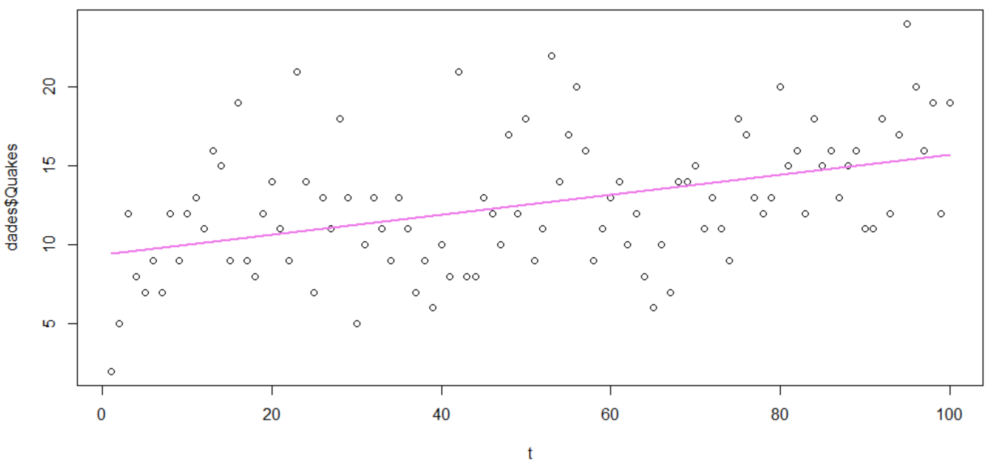
```
### temps indexat
dades.lm=lm(Quakes~t , data=dades)
summary(dades.lm)
beta <- dades.lm$coefficients
plot(t, dades$Quakes);
lines(t, beta[1]+beta[2]*t,col="orchid2", lwd=2)
```

```
Call:
lm(formula = Quakes ~ t, data = dades)

Residuals:
    Min      1Q  Median      3Q     Max
-7.5279 -2.7521 -0.2683  2.1594 10.1308

Coefficients:
            Estimate Std. Error t value   Pr(>|t|)
(Intercept)  9.41333    0.77295  12.178    < 2e-16 ***
t            0.06330    0.01329   4.764 0.00000659 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.836 on 98 degrees of freedom
Multiple R-squared:  0.188,	Adjusted R-squared:  0.1797
F-statistic: 22.69 on 1 and 98 DF,  p-value: 0.000006589
```

El modelo de regresión obtenido es el siguiente:

Quakes = 9,41333 + 0,06330 · t + Et

donde Et son los residuos.



Los coeficientes son ambos significativamente distintos a 0 con p-valores mucho más pequeños a 0,01, tanto para la constante de intersección con el eje de la y (9,41333) como el que acompaña a la variable tiempo (0,06330).

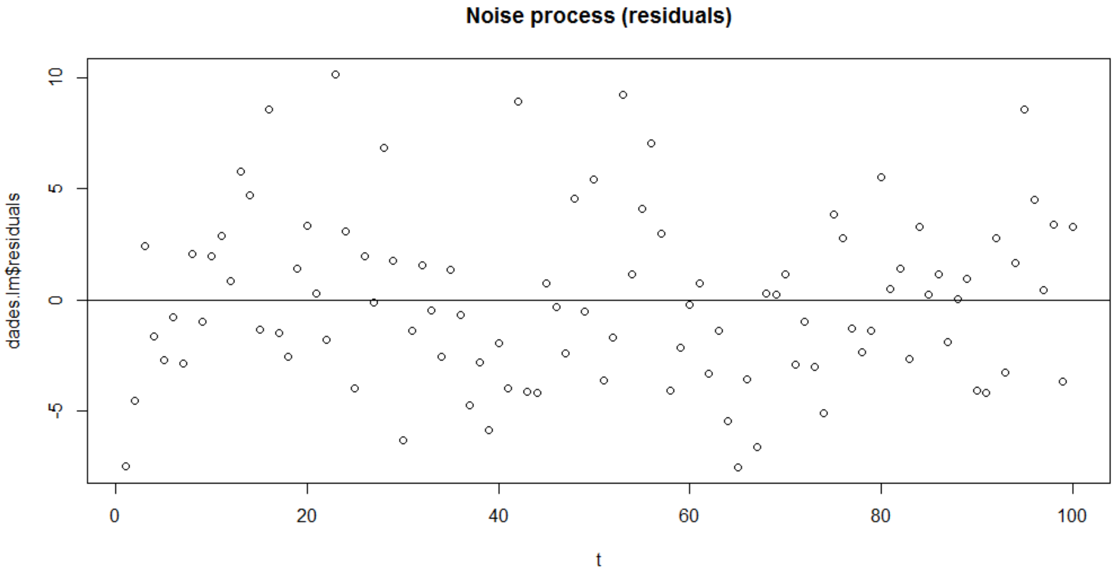
El coeficiente de determinación es 18,8%, lo que nos dice que la variable tiempo explica un 18,8% de la variabilidad de la serie terremoto (Quakes).

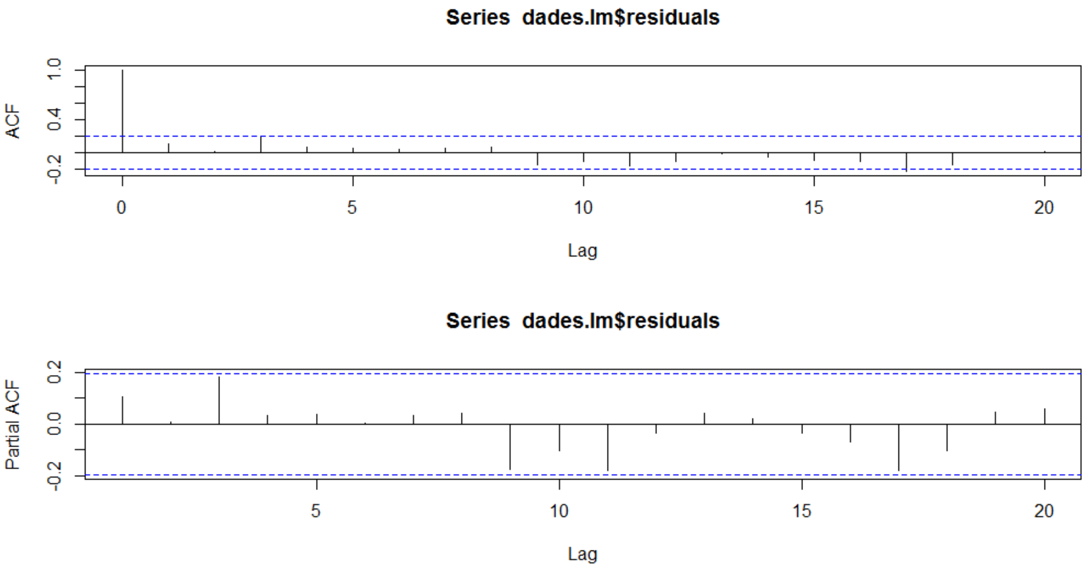
Todo y que la variable t explica la tendencia in crescendo de la serie en el tiempo, la bondad de ajuste es pequeña de tan sólo un 18,8%.

v. Tomad los residuos de la regresión lineal y dibuja las autocorrelaciones y autocorrelaciones parciales, ¿qué tipo de proceso te parece que estamos tratando?

```
plot(t, dades.lm$residuals, main="Noise process (residuals)");
```

```
abline(h=0)
par(mfrow = c(2,1))
acf(dades.lm$residuals)
pacf(dades.lm$residuals)
```





Tanto la gráfica de los residuos como el correlograma y la función de autocorrelación parcial muestran un ruido blanco. Tanto en el correlograma como en la función de autocorrelación todos los valores están dentro de las bandas que delimitan el intervalo de los valores muy cercanos a “cero”.

vi. La función auto.arima() del paquete forecast permite ajustar automáticamente cuál es el tipo de modelo más adecuado en base a diversos criterios de información. Prueba si el modelo que propone esta función para los residuos te parece el más razonable.

Código R

```
auto.arima(dades.lm$residuals)
```

La función auto.arima() de R nos confirma que los residuos siguen un ruido blanco.

```
> auto.arima(dades.lm$residuals)
Series: dades.lm$residuals
ARIMA(0,0,0) with zero mean

sigma^2 estimated as 14.42:  log likelihood=-275.32
AIC=552.64   AICc=552.68   BIC=555.25
```