

# Lección 4.C - Simulación de procesos MA(2) en la región de invertibilidad

Marcos Bujosa

## 1. Objetivo de la práctica

Guión: [P-L04-C-simulacion-procesos-MA.inp](#)

### Objetivo

1. Observar la ACF y PACF de distintos modelos MA(2) en las distintas regiones del triángulo de invertibilidad.

### Requerimientos previos

Programe o recupere de una práctica anterior una función que simule procesos MA( $q$ )

```
function series SimuladorMA(matrix theta)
    # SimuladorMA(theta) simula un proceso MA(q),
    # donde theta es el polinomio MA y q es su grado.
    series WN = normal (0,1)
    series X = 0
    loop i=1..cols(theta)
        X = X + theta[i]*WN(1-i)
    endloop
    return X
end function
```

Para que se observe bien la estructura de las ACF y PACF estimadas, establezca un tamaño de muestra suficientemente grande.

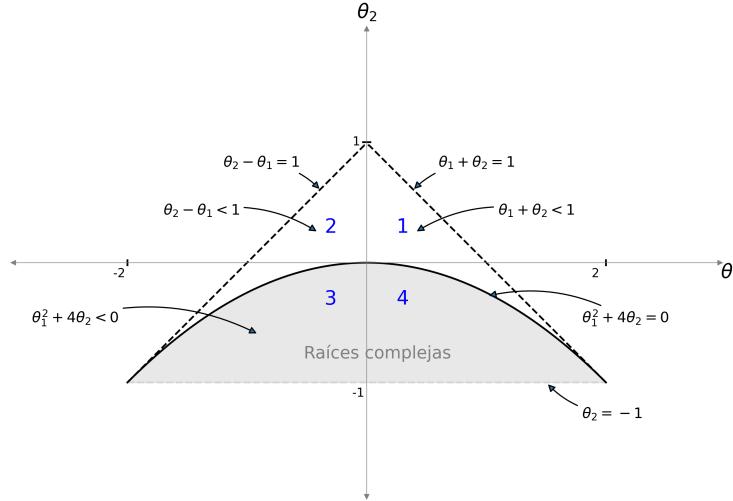
```
# establecemos la muestra
nulldata 3500
setobs 12 1900:01 --time-series
```

Recuerde cómo usar la función

```
scalar theta1 = 0
scalar theta2 = 0.8
series X = SimuladorMA( {1, -theta1, -theta2} )
figura <- corrgm X 12
```

## 2. Actividad 1 - Probando pares de valores en distintas regiones de invertibilidad

Asigne varios pares de valores  $\theta_1$  y  $\theta_2$  que pertenezcan a cada una de las regiones indicadas en la figura y explore cómo se comportan la AFC y PACF en cada caso.

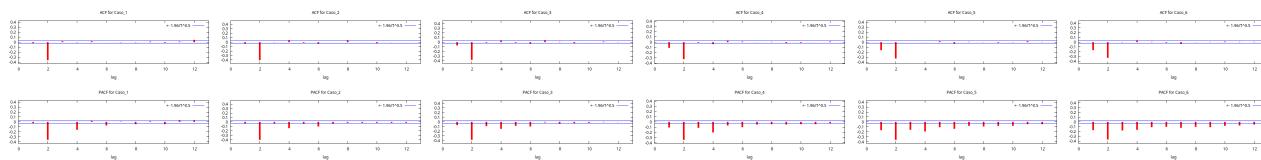


$$\rho_1 = \frac{-\theta_1(1-\theta_2)}{1+\theta_1^2+\theta_2^2}; \quad \rho_2 = \frac{-\theta_2}{1+\theta_1^2+\theta_2^2}; \quad \pi_1 = \rho_1; \quad \pi_2 = \frac{\rho_2 - \rho_1^2}{1-\rho_1^2}; \quad \pi_3 = \frac{\rho_1^3 - \rho_1\rho_2(2-\rho_2)}{1-\rho_1^2 - 2\rho_1^2(1-\rho_2)}; \quad \dots$$

### 2.1. Zona 1

Correlogramas en zona 3

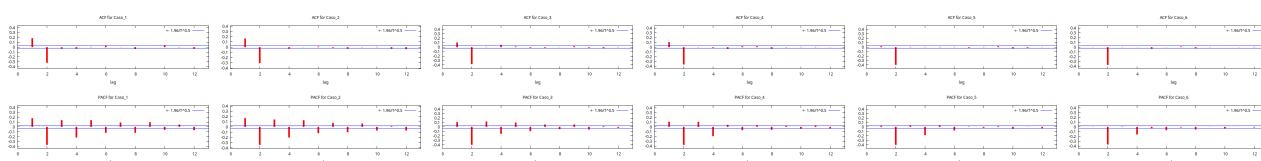
- $\theta_2 = 0,5$
- $\theta_1 \approx 0, 0,1, 0,2, 0,3, 0,4, 0,45.$



### 2.2. Zona 2

Correlogramas en zona 3

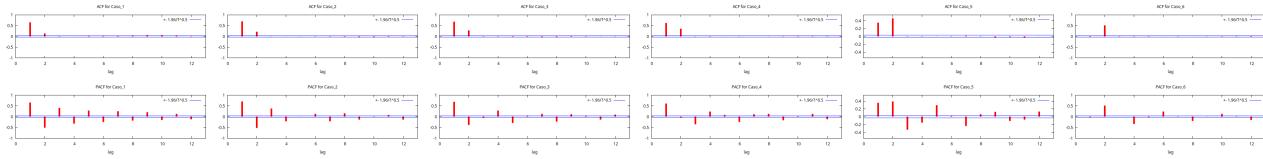
- $\theta_2 = -0,45$
- $\theta_1 = \approx -0,45, -0,4, -0,3, -0,2, -0,1, 0.$



## 2.3. Zona 3

Correlogramas en zona 3

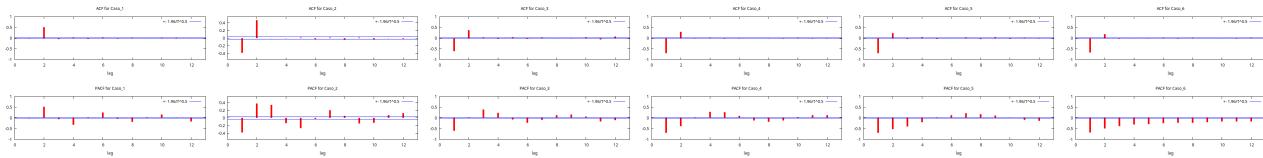
- $\theta_2 = -0,95$
- $\theta_1 \approx -1,95, -1,6, -1,2, -0,8, -0,4, 0.$



## 2.4. Zona 4

Correlogramas en zona 4

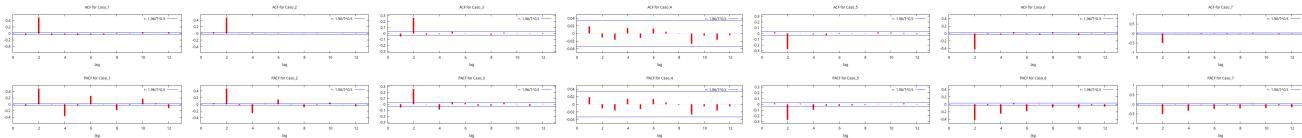
- $\theta_2 = -0,95$
- $\theta_1 \approx 0, 0,4, 0,8, 1,2, 1,6, 1,95.$



## 2.5. Eje vertical

Correlogramas correspondientes a puntos sobre el eje vertical

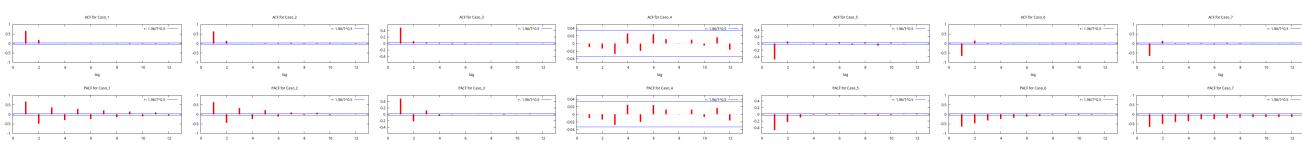
- $\theta_1 = 0$
- $\theta_2 \approx -0,9, -0,6, -0,3, 0, 0,3, 0,6, 0,9.$



## 2.6. Parábola

Correlogramas correspondientes a puntos sobre la parábola

- $\theta_1 \approx -1,9, -1,3, -0,6, 0, 0,6, 1,3, 1,9.$
- $\theta_2 = \frac{-\phi_1^2}{4}$



### 3. Código completo de la práctica

P-L04-C-simulacion-procesos-MA.inp

```
# _____
# Copyright (C) 2025 Marcos Bujosa
# Licencia: GNU General Public License v3.0 o posterior
# Este código es software libre y puede ser redistribuido y/o modificado bajo los términos de la GPL.
# Ver el archivo LICENSE del repositorio para más detalles.
# _____

# Los dos primeros comandos son necesarios para que Gretl guarde los resultados de la práctica en el directorio de trabajo
# al ejecutar lo siguiente desde un terminal (use los nombres y ruta que correspondan)
#
# DIRECTORIO="Nombre_Directorio_trabajo"gretlcli -b ruta/nombre_fichero_de_la_practica.inp
#
# Si esto no le funciona en su sistema, comente las siguientes dos líneas y sítuese en el directorio de trabajo de gretl
# que corresponda (configure dicho directorio de trabajo desde la ventana principal de Gretl).

string directory = getenv("DIRECTORIO")
set workdir "@directory"

function series SimuladorMA(matrix theta)
    # SimuladorMA(theta) simula un proceso MA(q),
    # donde theta es el polinomio MA y q es su grado.
    series WN = normal (0,1)
    series X = 0
    loop i=1..cols(theta)
        X = X + theta[i]*WN(1-i)
    endloop
    return X
end function

# establecemos la muestra
nulldata 3500
setobs 12 1900:01 --time-series

scalar theta1 = 0
scalar theta2 = 0.8
series X = SimuladorMA( {1, -theta1, -theta2} )
figura <- corrgm X 12

scalar theta2 = 0.45
scalar n = 0
loop for (r=0; r<=1-theta2-0.04; r+=.09)
    n += 1
    scalar theta1 = r
    nombre = sprintf("Zona1-%d(%1.2f-%1.2f)", n, theta1, theta2)
    sname = sprintf("Caso_%d", n)
    series @sname = SimuladorMA( {1, -theta1, -theta2} )
    corrgm @sname 12 --plot=@nombre.png
endloop

scalar theta2 = 0.45
scalar n = 0
loop for (r=-theta2; r<=0.03; r+=.09)
    n += 1
    scalar theta1 = r
    nombre = sprintf("Zona2-%d(%1.2f-%1.2f)", n, theta1, theta2)
    sname = sprintf("Caso_%d", n)
    series @sname = SimuladorMA( {1, -theta1, -theta2} )
    corrgm @sname 12 --plot=@nombre.png
endloop

scalar theta2 = -0.95
scalar n = 0
```

```

loop for (r=theta2-1; r<=0.01; r+=.39)
n += 1
scalar theta1 = r
nombre = sprintf("Zona3-%d(%1.2f-%1.2f)", n, theta1, theta2)
sname = sprintf("Caso_%d", n)
series @sname = SimuladorMA( {1, -theta1, -theta2} )
corrgm @sname 12 --plot=@nombre.png
endloop

scalar theta2 = -0.95
scalar n = 0
loop for (r=0; r<=1-theta2+0.01; r+=.39)
n += 1
scalar theta1 = r
nombre = sprintf("Zona4-%d(%1.2f-%1.2f)", n, theta1, theta2)
sname = sprintf("Caso_%d", n)
series @sname = SimuladorMA( {1, -theta1, -theta2} )
corrgm @sname 12 --plot=@nombre.png
endloop

scalar theta1 = 0
scalar n = 0
loop for (r=-.96; r<=1; r+=0.32)
n += 1
scalar theta2 = r
nombre = sprintf("EjeVertical-%d(%1.2f-%1.2f)", n, theta1, theta2)
sname = sprintf("Caso_%d", n)
series @sname = SimuladorMA( {1, -theta1, -theta2} )
corrgm @sname 12 --plot=@nombre.png
endloop

scalar n = 0
loop for (r=-1.91; r<=2; r+=0.636)
n += 1
scalar theta1 = r
scalar theta2 = -theta1*theta1/4
nombre = sprintf("Parabola-%d(%1.2f-%1.2f)", n, theta1, theta2)
sname = sprintf("Caso_%d", n)
series @sname = SimuladorMA( {1, -theta1, -theta2} )
corrgm @sname 12 --plot=@nombre.png
endloop

```