

Simulación de múltiples modelos lineales

Marcos Bujosa

20 de octubre de 2025

1. Objetivo

Que el estudiante tenga una gran batería de series para ejercitar la identificación. Los nombres de los ficheros impiden saber a priori de qué modelo se trata en cada caso (cada serie tiene un hash único). El alumno tendrá que observar el gráfico *rango-media* para decidir primero si la serie necesita la transformación logarítmica. Después tendrá que ir mirando el gráfico de las series, de los residuos, los correlogramas (también es útil mirar el gráfico del espectro del modelo junto al periodograma) para intentar indentificar el modelo lineal con el que se ha simulado la serie temporal.

Los ficheros `Etiqueta.txt` almacenan el tipo de modelo que se ha simulado en cada caso. Por tanto, una vez se ha identificado un modelo, se puede verificar si efectivamente ese ha sido el modelo usado al crear la serie buscando el hash del fichero en la primera columna de `Etiqueta.txt`.

2. Qué hace el código de más abajo

Simula desde procesos de ruido blanco a procesos $ARIMA(p, d, q) \times (P, D, Q)_S$, con transformaciones exponenciales en algunos casos y con medias distintas de cero en otros.

En total se simulan 320 modelos con periodicidad trimestral y 320 con periodicidad mensual.

Las series se simulan con la librería `tfarima` de R escrita por José Luis Gallego y cada una de ellas se graba en un fichero `csv` con un nombre anonimizado (mediante un hash) para ocultar cualquier pista sobre cuál es el modelo subyacente.

Para mayor comodidad del estudiante, cada serie es leída posteriormente con GRETL y guardada en formato Gretl (`*.gdt`) en el directorio [IdentificaEstosARIMA](#) (manteniendo los nombres anonimizados pero añadiendo tras un guión la supuesta periodicidad de los datos).

```
import itertools
import hashlib
import subprocess
import os

ar = [ '(1 - 0.8B)', '(1 + 0.8B)', '(1 - 0.8BS)' ]
ma = [ '(1 - 0.65B)', '(1 + 0.65B)', '(1 - 0.65BS)' ]
I   = [ '(1 - B)', '(1 - BS)' ]

def simulacion(lista_AR, lista_MA, lista_I, PeriodoEstacional=12):
    S = str(PeriodoEstacional)
    directorio = "SeriesSimuladas"+S
    d = os.path.join(directorio)
    if not os.path.exists(d):
        os.makedirs(d)

    def polinomios(monomios, S, NOLimita=1):
        polinomios = []
```

```

for i in range(3):
    for c in itertools.combinations(monomios, i):
        cc = ''.join(c).replace("S", "s")
        #polinomios = polinomios + [cc] if ((i < len(monomios)) / NOLimita) else polinomios
        polinomios = polinomios + [cc]
    return polinomios

SampleSize = 40*PeriodoEstacional
numero_de_modelos = 0
etiqueta = []
for ca,ma in enumerate(polinomios(lista_MA,str(S),0)):
    for cb,ar in enumerate(polinomios(lista_AR,str(S),0)):
        for cc,i in enumerate(polinomios(lista_I,str(S))):
            for l in range(2):
                if len(ar)<18 or len(ma)<18:
                    numero_de_modelos += 1
                    parametros = "ar = '%s', ma = '%s', i = '%s'" % (ar,ma,i)
                    modelo = "ARIMA <- um('%s)" % (parametros)
                    if l:
                        sim = 'y = exp((sim(ARIMA, n = %d)) * 0.1 + 2.5)' % (SampleSize,)
                        logs = 'logs'
                        media = 'mu = 2.5'
                    else:
                        sim = 'y = (sim(ARIMA, n = 200))'
                        logs = ' '
                        media = 'mu = 0.0'

                # script de R
                caso = chr(ca*1000+cb*100+cc*10+l)
                label = (hashlib.shake_128(caso.encode()).hexdigest(3))
                caracteristicas = logs + ',\t' + media + ',\t' + parametros
                etiqueta = etiqueta + [label + ',\t' + caracteristicas]
                fichero = "write.csv(y, file='%s/%s.csv')" % (directorio,label)
                with open('Rscript.r', 'w') as f:
                    f.write('library(tsfarima)+'\n')
                    f.write(modelo+'\n')
                    f.write(sim+'\n')
                    f.write(fichero+'\n')
                subprocess.call (["/usr/bin/Rscript", "--vanilla", "./Rscript.r"])

                # script de Gretl
                TamañoMuestral = 'nulldata %d' % SampleSize
                Dir = 'string directory = $(pwd) ~ "/IdentificaEstosARIMA"'
                DirTrabajo = 'set workdir "@directory"'
                periodicidad = 'setobs %s 1985:1 --time-series' % (S,)
                datos = 'open %s/%s.csv' % (directorio,label,)
                guardado = 'store %s-%s.gdt x' % (label,S)
                Guion = "GRETLScript"+S+".inp"
                with open(Guion, 'w') as f:
                    f.write(Dir+'\n'+DirTrabajo+'\n'+datos+'\n'+periodicidad+'\n'+guardado)

                subprocess.call (["/usr/bin/gretlcli", "-b", Guion])

print(numero_de_modelos)

with open(directorio+'/Etiquetas.txt', 'w') as f:
    f.write('etiqueta' + ',\t' + 'logs' + ',\t' + 'mu' + ',\t' + 'ar' + ',\t' + 'ma'+',\t' + 'nabla' + '\n')
    for e in etiqueta:
        f.write(e+'\n')

```

```

ar = [ '(1 - 0.8B)', '(1 + 0.8B)', '(1 - 0.8BS)']
ma = [ '(1 - 0.55B)', '(1 + 0.55B)', '(1 + 0.55BS)']
I = [ '(1 - B)', '(1 - BS)']
_ = simulacion(ar, ma, I, 4)
_ = simulacion(ar, ma, I, 12)

```
