

# Trabajo No 2 de Técnicas de Pronósticos

## Modelos ARMA-SARMA para los Residuos Estructurales

Diciembre de 2020

### 1. Descripción del Trabajo

1. El trabajo consiste de cuatro (4) puntos y tiene un valor de 33% de la nota final. Desarrollar cada punto por separado. En Moodle está el tema y la fecha límite de entrega.
2. El objetivo es encontrar el posible mejor modelo ARMA-SARMA para los residuos estructurales,  $\epsilon_t$ , del modelo de componentes utilizado para pronosticar en el trabajo No 1. Hay que identificar una pareja  $p, q = 0, 1, 2 \dots$  tal que  $\epsilon_t \sim ARMA(p, q)$ , ó parejas  $(p, q), (p_s, q_s)$  tales que  $\epsilon_t \sim SARMA(p, q)(p_s, q_s)[s]$ . Estimar el modelo escogido y calcular los pronósticos.
3. Finalmente, comparar los pronósticos del modelo estructural, los que se obtienen al sumarles los pronósticos calculados el modelo ARMA-SARMA y los del modelo de Espacio de Estados asignado.

# Puntos del Trabajo

Asuma la estrategia de validacion cruzada utilizada en el Trabajo No 1.

1. (25/25) **Pruebas de incorrelación.** Reporte la gráfica de la fac con las bandas de Bartlett, para los residuos estructurales obtenidos en el Trabajo No 1. Interprete el resultado. Realice las Pruebas de incorrelación Ljung-Box y Durbin-Watson. Concluya sobre si los residuos son ruido blanco o no.
2. (25/25) **Identifique** un posible modelo ARMA-SARMA , usando: 1) la función `auto.arima()` de la librería `forecast`. Use el código siguiente.

```
auto.arima(y,stationary=TRUE, seasonal=TRUE,ic= "aicc")
```

- 2) la función `armasubsets()` de la librería `TSA`. Reporte ambos modelos.
3. (25/25) **Estimación** Estime los dos modelos con la función `arima()`. Escoja el de menor AIC. Se calcula con `AIC(modelo)`. Con el modelo escogido valide los residuos con la fac y la prueba Ljung-Box. En caso de no obtener ruido blanco reporte este resultado.
4. (25/25) Calcule los pronósticos para la validación cruzada con los 3 modelos C, C+ARMA y EE. Reporte MAPE, RMSE, U-Theil para éstos. Concluya cuál modelo pronostica mejor.

Opciones	Modelos
(C) Componentes	(1.1) cuadrático + indicadoras (1.2) cúbico + indicadoras (1.3) exponencial (lineal + indicadoras) (1.4) exponencial ( cuadrático + indicadoras)
(EE) Espacio de Estados, Suavizadores	(2) Híbrido Loess + indicadoras (3) Holt-Winters (4) Modelo ETS-AAA (5) Espacio de Estados BSM (6) Red neuronal autoregresiva NNAR

Cuadro 1: Modelos para Pronosticar

Cuadro 2: Asignación de modelos y serie por grupo

Grupo = Serie No	C	EE	Estudiantes
1 ✓	1.1	2	Manuela Cuartas – Yesica Velasquez
2 ✓	1.1	3	Sebastián González
3 ✓	1.1	4	Cristian Gaviria – Katerine Mejia
4 ✓	1.1	5	Luis Fernando Marin – Giselle Viviana Martínez
5 ✓	1.1	6	Hernán Dario Jaramillo – Vanessa Sarrazola
6 ✓	1.2	2	Christian Areiza – Giovanny Soto
7 ✓	1.2	3	Ana María Hincapié – Madelin Uribe
8 ✓	1.2	4	Valentina Aguirre
9 ✓	1.2	5	Christian Quintero Arias – Santiago González
10 ✓	1.2	6	Maria Isabel Londoño – Juan Esteban Arroyave
11 ✓	1.3	2	Sebastian Lopez Betancur
12 ✓	1.3	3	David Morales Rave
13 ✓	1.3	4	Juan Camilo Palacio – Yesica Andrea Velasquez
14 ✓	1.3	5	Julian Stivens Usuga Ramirez
8 ✓	1.2	4	Pablo Andres Jarava
20 ✓	1.4	6	Rodolfo Casadiego – Erika Johnnela Gómez

# Series para el Trabajo

La serie asignada a cada integrante está en la lista siguiente. Los archivos están en Moodle en la carpeta del tema del trabajo ó se pueden cargar desde las librerías indicadas en cada serie.

## (1) producción total vino

```
# Instrucciones de lectura. Archivo en Moodle
Wine sales
#Australia, monthly
#Jan 1980-Jun 1994
#Total --- Total Wine sales (th. of litres)
#by wine makers in bottles <= 1L.
#ABS (Australian Bureau of Statistics)
# Time Series Data Library
Vino = read.table("Wine.dat", header = TRUE, stringsAsFactors = FALSE)
attach(Vino)
y = ts(Vino$Total,frequency=12,start=c(1980,01))
```

## (2) Demanda diaria energía eléctrica en Bogotá,

```
Empresa Codensa S.A., en Gwh,
entre 1995-07-20 y 1997-10-23.
Archivo en Moodle
S = read.table("codensa.diaria.dat",
header = TRUE, stringsAsFactors = FALSE)
```

## (3) precio de carne pavo

```
turkey.price.ts: Monthly Average Turkey Price, January 2001 to April 2008
his time series shows the average retail price of turkey in the United States be
S = read.table("turkey.price.dat",
header = TRUE, stringsAsFactors = FALSE)
attach(S)
```

## (4) La serie h02,

```

en la librería fpp,
se describe como: ‘‘Monthly cortecosteroid
drug sales in Australia from 1992 to 2008’’.
Se carga con las instrucciones
require(fpp)
y = h02
# generar un vector de fechas, clase 'Date'
fechas = seq(as.Date("1992/1/1"),
length.out = length(y), by = "months")

```

### (5) producción vino tinto

```

Instrucciones de lectura. Archivo en Moodle
Wine sales
#Australia, monthly
#Jan 1980-Jun 1994
#Total --- Total Wine sales (th. of litres)
#by wine makers in bottles <= 1L.
#ABS (Australian Bureau of Statistics)
# Time Series Data Library
Vino = read.table("Wine.dat", header = TRUE, stringsAsFactors = FALSE)
attach(Vino)
y = ts(Vino$Red,frequency=12,start=c(1980,01))

```

### (6) Empleo masculino Suiza.

```

Time series from http://www.dataserries.org,
a comprehensive and up-to-date
collection of open data from Switzerland
require(dataserries)
D=dataserries::ds("ch_comb_jobs.596.tot.1.0")
y = D$ch_comb_jobs.596.tot.1.0
names(y) = "empleo total hombres trimestral Suiza"
fechas = D$time
y=ts(y,frequency=4)
ts.plot(y)

```

### (7) Turistas Suiza.

```

Time series from http://www.dataserie.org,
a comprehensive and up-to-date
collection of open data from Switzerland
require(dataseries)
D=dataseries::ds("ch_comb_jobs.596.tot.1.0")
y = D$ch_comb_jobs.596.tot.1.0
names(y) = "empleo total hombres trimestral Suiza"
fechas = D$time
y=ts(y,frequency=4)
ts.plot(y)

```

**(8) accidentes fatales auto en UK,**

```

# leer datos de la serie Seatbelts:
UKDriverDeaths is a time series giving the monthly
totals of car drivers in Great Britain killed or
seriously injured Jan 1969 to Dec 1984.
Compulsory wearing of seat belts was
introduced on 31 Jan 1983.
data(Seatbelts)
# car drivers killed.
y = Seatbelts[,2]
ts.plot(y)

```

**(9) La serie IPC,**

```

Variación porcentual mensual del Índice de precios
al consumidor, 2000-2012.
library(readxl)
res <- read_excel("IPCporcentual.xlsx", 1) # lee el primer libro
attach(res)
y = ts(IPC, frequency=12,start=c(2000,01))
ts.plot(y)

```

**(10) La serie PIB Colombia,**

```

PIB Trimestral Colombia, Sin desestacionalizar
a precios constantes 2005.

```

```
G = read.table("PIBtrimestral.dat", header = TRUE,
stringsAsFactors=FALSE)
attach(G)
fechas = seq(as.Date("2000/1/1"),
length.out = length(pib), by = "quarter")
y = ts(pib,frequency=4,start=c(2000,01))
```

#### (11) precio carne hamburguesa

A time series objects consisting of average monthly retail prices per pound of ham in the United States between January 2001 and April 2008

```
S = read.table("ham.price.dat",
header = TRUE, stringsAsFactors = FALSE)
attach(S)
```

#### (12) turismo Chipre

```
# leer turismo chipre: 1 serie
tu.chipre = read.table("turismo.chipre.dat",
header = TRUE, stringsAsFactors = FALSE)
y = tu.chipre$y
# generar un vector de fechas, clase 'Date'
fechas = seq(as.Date("1979/1/1"),
length.out = length(y), by = "month")
```

#### (13) indice vivienda USA Schiller San Diego

```
# leer indice SDXRSA indice Schiller
# sobre variación de los precios de vivienda
# en la ciudad de San Diego, EUA.
# ver https://fred.stlouisfed.org/series/SDXRSA
D = read.csv("SDXRSA.csv", header = TRUE,
stringsAsFactors = FALSE)
# usar la variacion mensual
y = diff(log(D$SDXRSA),1,1)
```

#### (14) generacion energía eléctrica

```

en la librería fpp, se describe como:
‘‘ Electricity monthly total net
generation. January 1973–October 2010.’’
require(fpp)
data(usmelec)
y = ts(usmelec,frequency=12, start=c(1973,01))
# generar un vector de fechas, clase 'Date'
fechas = seq(as.Date("1973/1/1"),
length.out = length(y), by = "months")
plot(fechas,y,type='l')

```

### (15) ventas antidiabético

```

en la librería fpp, se describe como
‘‘Monthly anti-diabetic drug sales
in Australia from 1992 to 2008’’.
require(fpp)
data(a10)
y = ts(a10,frequency=12,start=c(1992,01))
# generar un vector de fechas, clase 'Date'
fechas = seq(as.Date("1992/1/1"),
length.out = length(y), by = "months")
plot(fechas,y,type='l')

```

### (16) Producción industrial España

```

# descripcion
Time-Series [1:61] from 1978 to 1983:
Brockwell and Davis (1991, Series E, p. 556)"
"Industrial production, Spain: monthly"
# instalacion
devtools::install_github("FinYang/tsdl")
require(tsdl)
# lectura
D <- subset(tsdl,12,"Production")
y = ts(D[[17]],frequency=12,start=c(1978,01))
par(mfrow=c(1,1))
t= seq(1,length(y))

```



```
ts.plot(y,type='b')
```

**(17) serie turismo**

```
# descripcion
The Tcomp package provides data from the tourism
forecasting competition described in George Athanasopolous,
Rob J. Hyndman, Haiyan Song, Doris C. Wu (2011)
‘‘The tourism forecasting competition’’,
International Journal of Forecasting 27 (2011) 822-844.
# instalacion
require(Tcomp)
data(package='Tcomp')
# lectura
y1 = tourism$M82$x
ts.plot(y,type='b')
```

**(18) turismo Australia**

```
# descripcion
Quarterly visitor nights (in millions) spent by
international tourists to Australia. 1999-2015
# instalacion
require(fpp2)
# lectura
data(austourists )
y = ts(austourists,frequency=4,start=c(1999,01))
ts.plot(y,type='b')
```

**(19) producción cerveza Australia,**

```
en la librería fpp, se describe como:
Producción total de cerveza en Australia: ‘‘
Total quarterly beer production in Australia
(in megalitres) from 1956:Q1 to 2008:Q3.’’.
Se carga con las instrucciones
require(fpp)
y = ts(ausbeer,frequency=4,start=c(1956,01))
ts.plot(y)
```

## (20) poblacion insectos

```
# descripcion
poblacion de un tipo de mosca (blowfly)
en el estudio Brillinger, Guckenheimer, Guttorp and Oster (1980)
# instalacion
devtools::install_github("FinYang/tsdl")
require(tsdl)
D <- subset(tsdl,1,"Ecology")
# definir un periodo s=18 usando
require(sazedR)
S(D[[7]])
# lectura
y = ts(D[[7]],frequency=18)
ts.plot(y,type='b')
```