

# Trabajo No 1 de Técnicas de Pronósticos

## Pronósticos de Series de Tiempo con modelos de componentes, Holt-Winters y redes MLP

Septiembre de 2023

### Características del Trabajo

El trabajo para cada grupo consiste de 3 puntos enunciados en la sección siguiente. Los puntos se deben resolver con la serie asignada en Tabla 5.5. Las series se describen en la última sección y se indica en dónde se encuentran los datos. Tiene un valor de 40 % de la nota definitiva.

- **Presentación.** Elaborar el reporte en formato Word o Latex, con formato final pdf. Con el nombre de los integrantes del grupo y el número del grupo en el encabezado.
- **Detalles.** Desarrollar cada punto por separado. Las gráficas y las tablas deben contener una numeración, información resumida (caption), que permita entender de qué información se trata.
- **Entrega.** Cargar el informe pdf en el Moodle antes de la fecha de entrega. El archivo pdf debe identificarse con los apellidos de los autores. Tiene un plazo de entrega de 2 semanas (en días hábiles) a partir de la fecha de publicación en Moodle.

### Puntos del Trabajo

Defina la estrategia de validación cruzada escogiendo una parte de los datos para ajustar y otra para comparar los pronósticos. En series mensuales use 12 meses, en trimestrales use 8 trimestres. En frecuencia diaria use 15 días.

1. (40/40) Estime los modelos (1),(2),(3),(4),(5),(6) asignados en la Tabla 5.4. Reporte R-cuadrado ajustado, MSE, AIC, BIC para todos los modelos en una Tabla 6 por 4. Concluya cuál modelo ajustó mejor. Para este mejor modelo reporte: en los casos 1-4, las tablas de parametros estimados, estadísticos t y F, valores p. En los otros casos reporte el informe que genera la función de estimación.
2. (30/30) Calcule los pronósticos para la validación cruzada con los modelos. Reporte RMSE, MAE, MAPE, U-Theil en una Tabla 6 por 4. Concluya cuál modelo pronostica mejor. Reporte la gráfica de los datos observados versus los estimados y los pronosticados para este modelo.
3. (30/30) Reporte el resultado de las pruebas cusum de estabilidad estructural usando el modelo (2). Qué implicaciones puede tener este resultado para los pronósticos?

Modelos	Sección	Codigo R
(1) cuadrático + indicadoras	§3.6	Código R. 3.1, p.79
(2) cúbico + indicadoras	§3.6	Código R. 3.1, p.79
(3) exponencial lineal + indicadoras	§3.6	
(4) exponencial cuadrático + indicadoras	§3.6	
(5) Holt-Winters-amortiguado	§5.2.4 p. 114	Código R. 5.6, p. 114
(6) Red neuronal NNAR	5.22 p. 123	Código R.5.9, p. 124

Tabla 5.4: Modelos de series de tiempo

## Algunas sugerencias

1. En la carpeta “Ejemplos métodos descomposicion” de Moodle hay varios programas de muestra para los puntos 1. y 2. con los modelos (1),..., (4).
2. Use concatenación horizontal (por filas) para formar las matrices 6 por 4 de los puntos 1. y 2. Por ejemplo  

```
(M=rbind(M,medidas(mod2,yi,k)))
```

```
(B = rbind(B,accuracy(pron2,yf)))
```
3. Hay una función `medidas.r` para los modelos (1),..., (4). Y otra `medidas.hw.r` para los modelos (5), (6).

Tabla 5.5: Asignación de modelos y serie por grupo

Grupo = Serie No	Estudiantes
1	Laura Acevedo – Jesús Pejendino – Carolina Gaitán
2	Amador Calle – Carolina Vergara – Carlos Montoya
3	Jesus Castellanos – Deicy Sanchez
4	Santiago Cano – Sebastian Velasco
5	Mario Andrés Amado – Daniel Alzate
6	Daniel Uran Rios – Wilson Graciano Bravo
7	David Arango – Melissa Parra
8	Juan David Valencia – Juan Pablo Estrada
9	Paola Andrea Calle – David Zuleta
10	Laura Maria Caro – Sebastián Ocampo
11	Jashley Chica – Sebastián Ospina
12	Sebastian Caro – Juan David Acosta
13	Laura Marcela Santa – Andrés Mauricio López
14	Pedro Ramon Machado – Laura Juliana Preciado
15	Santiago Agudelo Restrepo
16	Kevin Bravo Bolaño
17	David Sierra Zuluaga
18	Sebastian Restrepo Betancur
19	Hernán Pabón Delgado
20	Adiel Ignacio Restrepo
21	Kaline Andrea Ríos – Cristian Alberto Cortes
22	Juan Sebastian Falcon – Daniela Pico
23	Jhon Alexander Gonzalez Valencia

## 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) Precio tomate USA

Average Price: Tomatoes, Field Grown  
(Cost per Pound/453.6 Grams) in U.S.  
City Average, U.S. Dollars, Monthly, Not Seasonally Adjusted.  
Archivo excel : precio.tomate.us.xls

### (2) Precio naranjas USA

Average Price: Oranges, Navel  
(Cost per Pound/453.6 Grams) in U.S.  
City Average, U.S. Dollars, Monthly, Not Seasonally Adjusted  
Archivo excel : precio.naranja.usa.ciudades.xls  
NOTA: serie con datos faltantes. Usar imputación. Ver el  
programa R: ejemplo1.imputacion.datos.faltantes.r  
en la carpeta Moodle: Cap2. Ejemplos uso básico de series con R

### (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 between January 2001 and April 2008  
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.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)

```

**(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 Indice 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 consiting 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) 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))
```

#### **(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 M82**

```

# 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) Datos de demanda de energía eléctrica horaria, Antioquia, datos de XM**

datos en el archivo: demanda.horaria.dat

### **(21) 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)
```

### **(22) serie turismo M69**

```
# 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$M69$x
ts.plot(y,type='b')
```

### **(23) serie turismo M65**

```
# 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$M65$x
ts.plot(y,type='b')
```

### **(24) 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)
```

### **(25) 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')
```