

Curso Técnicas de Pronóstico

Modelos ARMA-SARMA para los Residuos Estructurales

Camilo Cabrera Meneses¹ y John Mario Montoya Zapata²

Grupo No.2

Recepción: 02 de noviembre de 2022

Se aceptan comentarios y/o discusiones al escrito.

Resumen

Se evalúan los pronósticos de la serie de residuos determinada a partir del modelo Holt-Winters ajustado en el primer trabajo. Se efectúa las respectivas pruebas para determinar si los residuos están incorrelacionados o no. Posteriormente se identifican y evalúan dos modelos ARMA-SARMA para determinar los pronósticos de los residuos de la serie ajustada Holt-Winters. El mejor modelo se obtuvo por medio de la función `auto.arima()` resultando un modelo $ARIMA(1,0,2)(0,0,0)[4]$. Los pronósticos para la franja de validación cruzada incluyendo los pronósticos de los residuos no presentan mejora apreciable, obteniendo valores de RMSE igual a 9.93, MAPE igual a 2.23 y U-Theil igual a 0.023.

Palabras claves: Residuos estructurales, ARMA, SARMA, series de tiempo, pronóstico.

Introducción

En el presente trabajo se evalúan dos modelos ARMA-SARMA para los residuos estructurales del modelo Holt-Winters Componentes ajustado para la serie de Producción de cerveza en Australia en el trabajo No.1. La figura 1 muestra los valores ajustados por el modelo y la serie de datos originales.

¹ Estudiante de Maestría en Recursos Hidráulicos, Universidad Nacional de Colombia, Medellín, Colombia, cacabreram@unal.edu.co.

² Estudiante de Especialización en Analítica, Universidad Nacional de Colombia, Medellín, Colombia, jmmontoyaz@unal.edu.co.

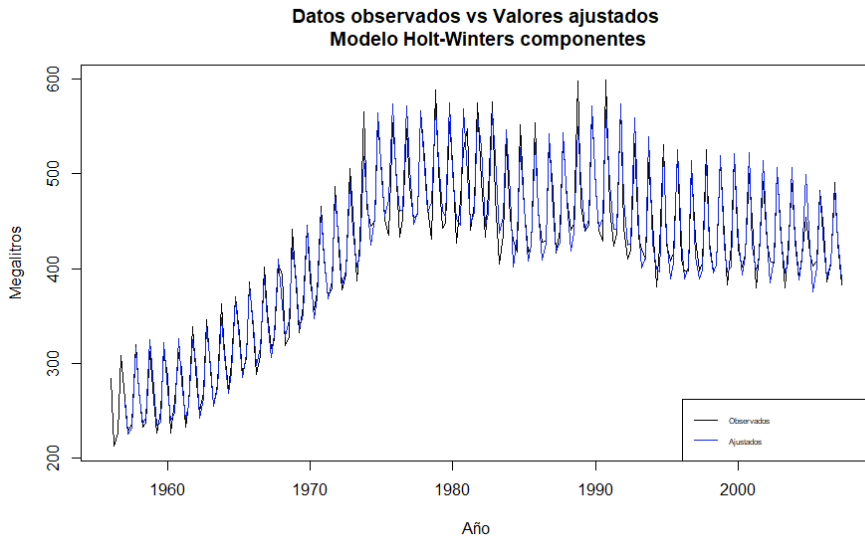


Fig. 1. Ajuste del modelo Holt-Winters componentes

A partir de la serie de valores ajustados por el modelo Holt-Winters se determinan los residuos estructurales por medio de la función *resid* de R. La figura 2 muestra la serie de residuos estructurales a la cual se realizarán los respectivos ajustes de los modelos ARMA-SARMA.

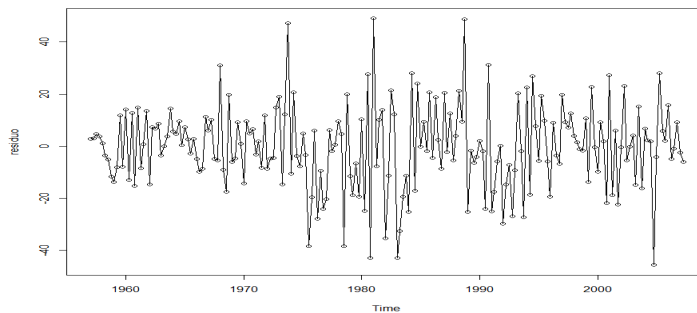


Fig. 2. Serie de residuos estructurales

La sección (1) presenta el resultado de las pruebas de incorrelación aplicadas a la serie de residuos con las respectivas graficas de diagnóstico. La sección (2) presenta la identificación de los modelos ARMA-SARMA. La sección (3) presenta la estimación de los modelos y la selección de aquel que se ajusta mejor; y finalmente en la sección (4) se calculan los pronósticos para la

validación cruzada para el modelo elegido y se concluye sobre cuál pronostica mejor.

En el anexo se encuentra el código desarrollado en lenguaje R para la implementación y evaluación de los modelos.

1 Pruebas de incorrelación

Como primer paso se ejecuta el panel de diagnóstico grafico que incluye las gráficas de la FAC con las respectivas bandas de Bartlett, el QQ plot y el periodograma. La figura 3 muestra el panel de diagnóstico, en el cual se puede observar que la gráfica de la FAC presenta el primer rezago por fuera de las bandas de Bartlett, sin embargo, los demás rezagos se encuentran dentro. Si bien el primer rezago, de los más importantes en este tipo de análisis, está fuera de las bandas, los demás rezagos al encontrarse dentro pueden indicar que los residuos están incorrelacionados. Por esta razón debemos asegurarnos de que los residuos no sean Ruido Blanco (RB) a partir de una prueba de incorrelación.

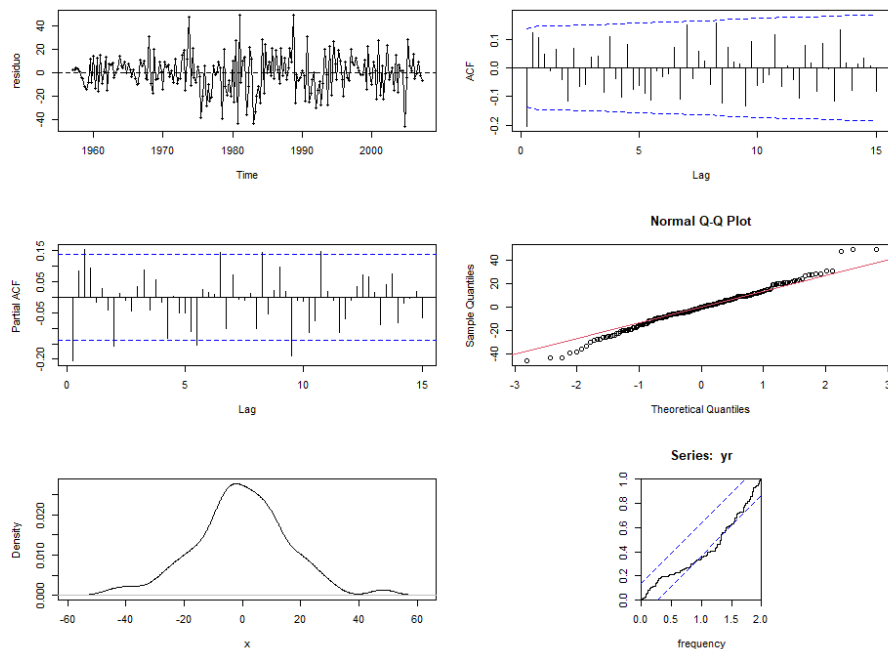


Fig. 3. Panel de diagnóstico para detectar incorrelación.

Ejecutamos la prueba Ljung-Box para 4, 8 y 12 rezagos, todos estos múltiplos del periodo 4 de la serie original que se quiere pronosticar. La tabla 1 muestra el resultado de las pruebas Ljung-Box, en la que se puede apreciar que para todos los rezagos seleccionados se halló un valor $p < 0.05$, con lo cual se rechaza la hipótesis nula, por lo tanto, los residuos **no** están incorrelacionados.

Tabla 1. Resumen pruebas de incorrelación Ljung test

Box-Ljung test			
Lag	X-squared	df	p-value
4	14.664	4	0.005493
8	18.763	8	0.01618
12	21.702	12	0.041

2 Identificación modelos ARMA-SARMA

Para identificar los posibles modelos ARMA-SARMA se hace uso dos funciones dentro de R, la primera *auto.arima()* de la librería forecast y la segunda *armasubsets()* de la librería TSA. Con los modelos ajustados se procede a establecer una comparativa para elegir el mejor modelo para los residuos.

En primer lugar, con la función *auto.arima()* se obtiene un modelo ARIMA(1,0,2)(0,0,0) con media cero. Los parámetros estimados del modelo ARIMA se presentan en la tabla 2, podemos observar que todos los parámetros son significativamente estadísticos (Valor $p < 0.05$).

Tabla 2. Prueba de coeficientes para el modelo ajustado con auto.arima()

Parámetro	Valor estimado	Desv. Estandar	Valor Z	Valor P
ar1	0.592328	0.163774	3.6167	0.0002983
ma1	-0.819230	0.165004	-4.9649	6.873e-07
ma2	0.304869	0.072929	4.1804	2.910e-05

Por otra parte, con la función *armasubsets()* se obtiene el panel de diagnóstico mostrado en la figura 4, el cual contiene los parámetros p y p_s en la parte izquierda, y los parámetros q y q_s en la parte derecha. Para la selección inicial de los parámetros buscamos el arma con menor BIC por lo cual, al ubicarnos en la primera fila del panel, es posible visualizar que el parámetro p es 0 y el parámetro p_s es 0 mientras que el parámetro q es 0 y el parámetro q_s es igual a 3(o 4) obteniendo así el modelo ARIMA(0,0,0)(0,0,3), sin embargo, al ejecutar la prueba de coeficientes se obtuvieron parámetros sin significancia estadística.

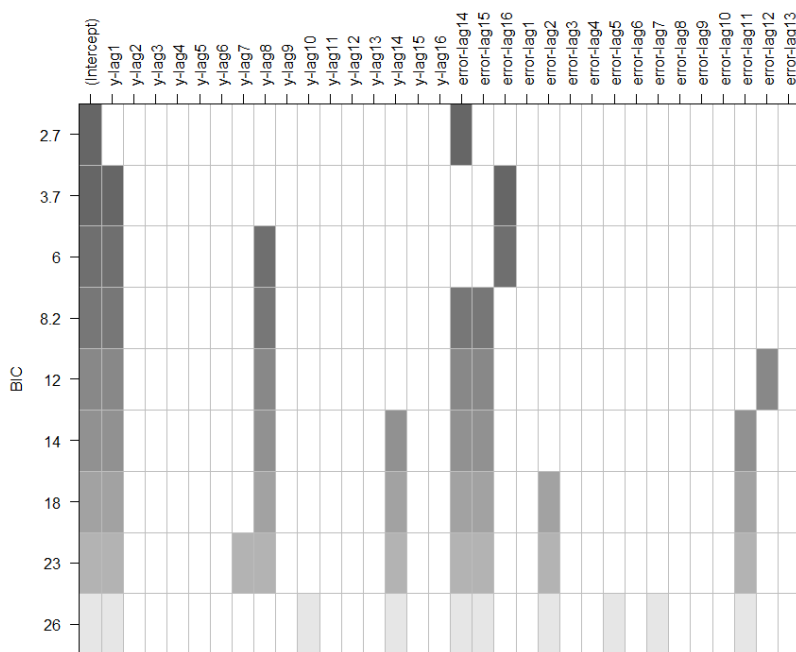


Fig. 4. Panel de diagnóstico obtenido con *armasubsets()*

Por la razón expuesta anteriormente se dio lectura a modelos con un BIC superior llegando al modelo ARIMA(1,0,2)(3,0,3)[4] cuyos parámetros son estadísticamente significativos y se muestran en la tabla 3.

Tabla 3. Prueba de coeficientes para el modelo ajustado con *arimasubsets()*

Parámetro	Valor estimado	Desv. Estandar	Valor Z	Valor P
ar1	0.594571	0.214121	27.768	0.0054897
ma1	-0.810775	0.216737	-37.408	0.0001834
ma2	0.303503	0.079043	38.397	0.0001232
sar1	0.854818	0.136881	62.450	4,24E-07
sar2	0.596044	0.216528	27.527	0.0059099
sar3	-0.813538	0.118738	-68.515	7,31E-09
sma1	-0.852745	0.133149	-64.044	1,51E-07
sma2	-0.739243	0.171465	-43.113	1,62E-02
sma3	0.959475	0.125348	76.545	1,94E-11

3 Selección de modelo

A partir de los modelos determinados con las funciones *auto.arima()* y *armasubset()* se procede a estimar los modelos mediante el uso de la función *arima()*. El resumen de la estimación de ambos modelos se presenta en las tablas 2 y 3, donde se puede evidenciar que los parámetros ajustados para cada modelo son significativos. A partir de los modelos estimados se calcula el AIC obteniendo 1688.756 para el modelo $ARIMA(1,0,2)(0,0,0)$ [4] obtenido con *auto.arima()* y 1692.530 para el modelo $ARIMA(1,0,2)(3,0,3)$ [4] obtenido con *armasubset()*. El modelo $ARIMA(1,0,2)(0,0,0)$ [4] obtuvo un menor AIC, por el cual seleccionamos este para modelo para ajustar a los residuos iniciales. Una vez el modelo se ajusta se ejecuta el panel de diagnóstico gráfico a los residuos obtenidos de este modelo ARMA y la prueba de incorrelación de Ljung-Box. La figura 5 muestra el panel gráfico de diagnóstico aplicado al modelo seleccionado, en el cual podemos observar que para la gráfica de la FAC se observan todos los rezagos dentro de las bandas de Bartlett, además el QQ plot muestra una distribución normal de los residuos, al igual que para el periodograma se observa la serie contenida entre las bandas. Todo lo anterior muestra fuertes indicios de residuos incorrelacionados.

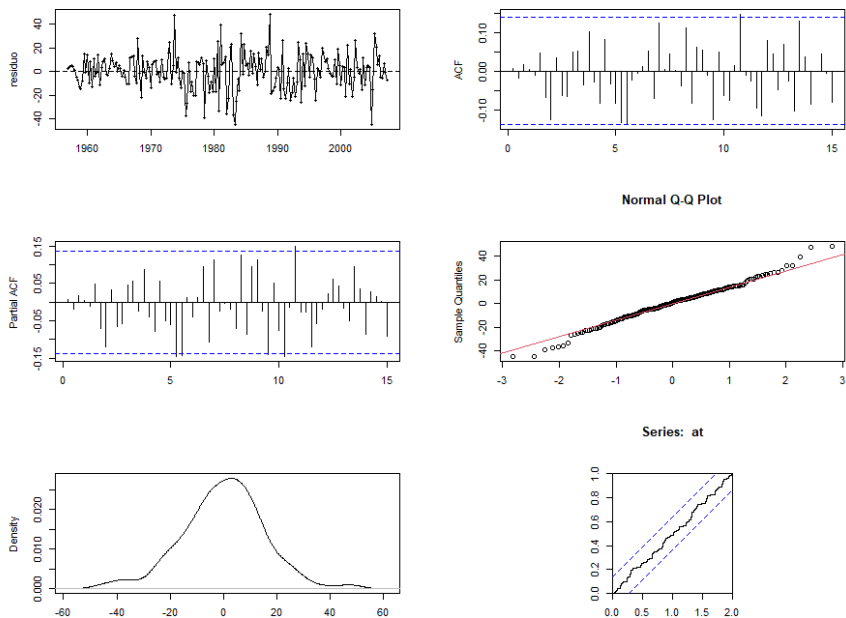


Fig. 5. Panel de diagnóstico aplicado al modelo ARMA para detectar incorrelación.

La tabla 4 muestra el resultado de las pruebas Ljung-Box ejecutadas para los rezagos 4, 8 y 12, en la que se puede apreciar que para todos los rezagos seleccionados se halló un valor $p > 0.05$, con lo cual **no** se rechaza la hipótesis nula, por lo tanto, los residuos **son RB**.

Tabla 4. Resumen pruebas de incorrelación Ljung test para el modelo ARMA.

Box-Ljung test			
Lag	X-squared	df	p-value
4	0.14576	4	0.9975
8	4.872	8	0.7712
12	7.3518	12	0.8335

4 Pronóstico y medidas de precisión en pronósticos

Como parte final del proceso de modelado se reportan los pronósticos para el periodo de validación cruzada, para el modelo Holt-Winters y para el modelo Holt-Winters + ARMA como se muestra en la figura 6. Podemos observar que no se presenta una mejora apreciable en los pronósticos al incorporar el pronóstico de los residuos.

Pronósticos: Producción trimestral de cerveza en Australia

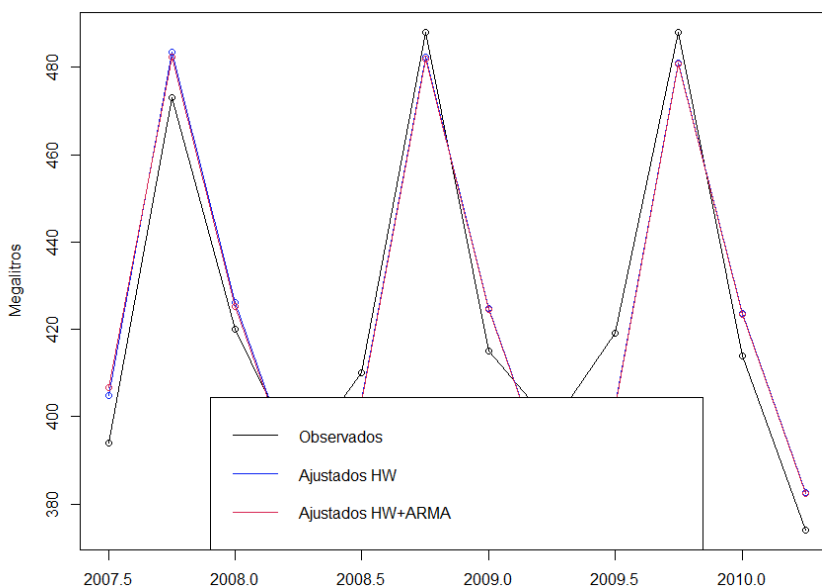


Fig. 6. Comparativa entre pronósticos en un horizonte de 3 años empleando el modelo Holt-Winters y ARMA.

Finalmente se calculan las medidas de precisión en los pronósticos RMSE, MAPE y U-Theil para el modelo HW + ARMA los cuales se presentan a continuación.

RMSE	MAPE	UTheil
9.92784281	2.23191476	0.02334986

Trayendo a colación los estadísticos del modelo Holt-Winters ajustado en el primer trabajo (RMSE de 9.78, MAPE de 2.199 y un U-Theil de 0.023) podemos observar que hay pequeños aumentos en los valores de los errores (RMSE, MAPE) y el U-Theil se mantiene constante, lo cual es muestra de que el incluir los pronósticos de los residuos no siempre da como resultado una mejora en los pronósticos globales de las series de interés.

Referencias

- Giraldo Gomez, N. (2006). TÉCNICAS DE PRONOSTICO Aplicaciones con R. In News.Ge. Universidad Nacional de Colombia.

Anexo

```
# Development of project No. 2 Forecasting Techniques
# Authors: Camilo Cabrera - John Mario Montoya

#-----#
#-----#
#----- Code to reproduce the selected model in project 1 ---#
#-----#
#-----#

# For mario enviroment: "H:\\My
Drive\\UN_Analytics_Specialization\\Forecasting_Techniques\\Projects
\\project2\\src"

# Libraries
library(ggplot2)
require(fpp2)
require(DescTools)

# Remove variables on memory
rm(list = ls())

# Functions for evaluate statistics of model selection
medidas.yest <- function(y, yest, k) {
  # y = serie, m = modelo, k = numero parametros
  T <- length(y)
  sse <- sum((yest - y)^2)
  ssr <- sum((y - mean(y))^2)
  mse <- sse / (T - k)
  R2 <- 1 - sse / ssr
  Ra2 <- 1 - (T - 1) * (1 - R2) / (T - k)
  aic <- log((T - k) * exp(2 * k / T) * mse / T)
  bic <- log(T^(k / T) * (T - k) * mse / T)
```

```

    M <- c(mse, sqrt(mse), Ra2, aic, bic)
    names(M) <- c("MSE", "RMSE", "R2-ajus", "logAIC", "logBIC")
    return(M)
}

medidas <- function(m, y, k) {
  # m = objeto producido con lm()
  # y = variable dependiente
  # k = numero de coeficientes beta
  T <- length(y)
  yest <- fitted(m)
  sse <- sum((yest - y)^2)
  ssr <- sum((y - mean(y))^2)
  mse <- sse / (T - k)
  R2 <- 1 - sse / ssr
  Ra2 <- 1 - (T - 1) * (1 - R2) / (T - k)
  aic <- log((T - k) * exp(2 * k / T) * mse / T)
  bic <- log(T^(k / T) * (T - k) * mse / T)
  M <- c(mse, sqrt(mse), Ra2, aic, bic)
  names(M) <- c("MSE", "RMSE", "R2-ajus", "logAIC", "logBIC")
  return(M)
}

# Load data
# Total quarterly beer production in Australia (in megalitres)
# From 1956:Q1 to 2008:Q3
y <- ts(ausbeer, frequency = 4, start = c(1956, 01)) # frequency = 4
--> Quarters
ts.plot(y)

#-----#
#----- cross-validation -----#
#-----#

n <- length(y)

```

```

len_yf <- 4 * 3 # 4 * num years for test data (10 in this case)
aprox 80-20
yi <- ts(y[1:(n - len_yf)], start = c(time(y)[1]), frequency = 4)
yf <- ts(y[(n - len_yf + 1):n],
        start = c(tail(time(yi), n = 1) + 0.25),
        frequency = 4
)

str(n)
str(length(yi))
str(length(yf))

# Plot data in train and test
plot(yi,
      type = "l", col = "black", # Plot first time series
      ylim = c(min(yi, yf), max(yi, yf)),
      xlim = c(time(y)[1], tail(time(y), n = 1)),
      main = "Producción trimestral de cerveza en Australia",
      xlab = "Año",
      ylab = "Megalitros"
)
lines(yf, , type = "l", col = "#ff7300") # Plot second time series

# Add legend to plot
text_legend <- c("Train", "Test")
legend("bottomright",
      legend = text_legend, ,
      text.width = strwidth(text_legend)[1] * 2,
      lty = 1, ,
      col = 1:2
)

#-----#
-----#

```

```

#----- Model estimate No1: Linear - Exponential -----
-----#
#-----#

lyi <- log(yi)

ti <- seq(1, length(yi))
ti2 <- ti * ti

It <- seasonaldummy(yi)

# Estimate auxiliar model log - linear
mod.llin <- lm(lyi ~ ti + It)
summary(mod.llin)

# Seasonal linear exponential model
T <- length(yi)
Xt <- cbind(rep(1, T), ti, It)
Ds <- data.frame(yi, Xt)
theta.0 <- mod.llin$coefficient

# Use nls function for adjust
mod.exp_lin <- nls(yi ~ exp(Xt %*% theta),
  data = Ds, start = list(theta = theta.0)
)

# Results
(summary(mod.exp_lin))

#-----#
#----- Model estimate No2: Components Holt-Winters -----
-----#
#-----#

```

```

#----- Model with auto-tuning of parameters-----
-----#

model.hw <- HoltWinters(yi)
# summary(model)
params <- (c(model.hw$alpha, model.hw$beta, model.hw$gamma))

# Trend, seasonal component and estimated Y
Tt <- model.hw$fitted[, 2] + model.hw$fitted[, 3]
St <- model.hw$fitted[, 4]
Yt_hat <- model.hw$fitted[, 1]

# Statistics model selection MSE, AIC, BIC R2-Adjust
# source('medidas.yest.r')
medidas.yest(yi, Yt_hat, 3) # OJO QUE DEVUELVE RMSE

#-----
-----#
#----- Project 2 -----
-----#
#-----
-----#

require(lmtest)
require(TSA)

# Test Ljung-Box (Using at least 2 lags) m = T/4, 2s, 3s; T:number of
data, s:Period

#- Structural residuals for model No.2 HW (Selected in project 1
like the best model) -#

# yr = na.omit(yi - Yt_hat)

```

```
yr = resid(model.hw)
# yr = ts(yr, frequency = 4)
```

```
plot(yr,type='o',ylab='residuo')
```

```
par(mfrow=c(3,2))
require(TSA)
plot(yr,type='o',ylab='residuo')
abline(h=0,lty=2)
TSA::acf(yr,60,ci.type="ma",drop.lag.0 = TRUE,main="")
pacf(yr,60,main="")
qqnorm(yr)
qqline(yr,col=2)
plot(density(yr),xlab='x',main= '')
cpgram(yr) # periodograma acumulado
```

La FAC permanece dentro de las bandas de barlet, y además el periodograma se
se mantiene casi siempre dentro de las bandas, lo que sugiere que es RB

```
Box.test(x = yr, lag = 4, type="Ljung-Box")
# p-value = 0.005493, se rechaza la nula luego hay evidencia de que no es RB
```

```
Box.test(x = yr, lag = 8, type="Ljung-Box")
# p-value = 0.01618, se rechaza la nula luego hay evidencia de que no es RB
```

```
Box.test(x = yr, lag = 12, type="Ljung-Box")
# p-value = 0.041, se rechaza la nula luego hay evidencia de que no es RB
```

Test Ljung-Box (Using at least 2 lags) $m = T/4, 2s, 3s$; T :number of data, s :Period

```

#dwtest(model.hw) # de la libreria lmtest

#library(car)
#durbinWatsonTest(model.hw,max.lag=10) # de la libreria car

#----- Structural residuals for model No. 1 Exp.Linea-----
-----#
yr = resid(mod.exp_lin)
yr = ts(yr, frequency = 4, start = c(time(y)[1]))

par(mfrow=c(3,2))
require(TSA)
plot(yr,type='o',ylab='residuo')
abline(h=0,lty=2)
TSA::acf(yr,60,ci.type="ma",drop.lag.0 = TRUE,main="")
pacf(yr,60,main="")
qqnorm(yr)
qqline(yr,col=2)
plot(density(yr),xlab='x',main= '')
cpgram(yr) # periodograma acumulado

# La FAC no permanece dentro de las bandas de barlett, y además el
periodograma
# se sale de las bandas, lo que sugiere que no es RB

Box.test(x = yr, lag = 4, type="Ljung-Box")
# p-value = 2.2e-16, se rechaza la nula, luego hay evidencia de que
no es RB

Box.test(x = yr, lag = 8, type="Ljung-Box")
# p-value = 2.2e-16, se rechaza la nula, luego hay evidencia de que
no es RB

Box.test(x = yr, lag = 12, type="Ljung-Box")

```

```
# p-value = 2.2e-16, se rechaza la nula, luego hay evidencia de que  
no es RB
```

```
#----- Modelos ARMA usando auto.arima() y armasubset -----  
-----#
```

```
#-----identificador armasubsets
```

```
res=armasubsets(y=yr,nar=14,nma=14,y.name='y',ar.method='ols')  
par(mfrow=c(1,1))  
plot(res)
```

```
# Usando el modelo HW  $p = 0$  (1),  $ps = 4$ ,  $q = 0$  (2),  $qs = 0$  (3)  
# ARIMA(1,0,2)(3,0,3)[4] with zero mean
```

```
# Usando el modelo exponencial lineal  $p = 4$ ,  $ps = 3$ ,  $q = 0$  (2),  $qs =$   
 $0$  (3)  
# ARIMA(4,0,0)(3,0,0)[4] with zero mean
```

```
#-----identificador auto.arima  
auto.arima(yr)
```

```
# Usando el modelo HW (Se obtiene un MA creo!!!!)  
# ARIMA(1,0,2)(0,0,0)[4] with zero mean
```

```
# Usando el modelo exponencial lineal (Se obtiene un ARIMA creo !!!)  
# ARIMA(1,1,2)(1,0,1)[4] with zero mean
```

```
# ----- Tuning hyperparameters to armasubsets and comparaing both  
models
```

```
# ---- Modelos usando Holt-Winters
```

```
m.1 =  
arima(yr,order=c(1,0,2),seasonal=list(order=c(3,0,3),period=4))
```



```

#m.1 =
arima(yr,order=c(1,0,2),seasonal=list(order=c(3,0,3),period=4))
coeftest(m.1)

m.2 =
arima(yr,order=c(1,0,2),seasonal=list(order=c(0,0,0),period=4))
coeftest(m.2)

# ---- Modelos usando Exponencial Lineal + indicadoras

m.1 =
arima(yr,order=c(3,0,2),seasonal=list(order=c(3,0,3),period=4))
coeftest(m.1)

m.2 =
arima(yr,order=c(1,0,2),seasonal=list(order=c(1,0,1),period=4))
coeftest(m.2)

# ---- Compare AIC
(c(AIC(m.1),AIC(m.2)))
# HW --- > 1692.530, 1688.756 Por lo cual es mejor el modelo
autoarima
# expo_lin --- > 1726.651, 1724.095 Por lo cual es mejor el modelo
autoarima

# Para modelos con exp ineal
# ES NECESARIO PREGUNTAR QUÉ SUCEDE CUANDO AUTOARIMA SUGIERE UN
MODELO NO ARMA
# INTEGRADO INICIALMENTE CAMBIÉ EL 1 A 0 EN LA PARTE NO ESTACIONAL,
SIN EMBARGO
# ESTO INCREMENTÓ EL AIC DE 1713.232 A 1724.095, SIN EMBARGO EL
COEFICIENTE AR1
# NO RESULTA SIGNIFICATIVO

at = resid(m.2)

```

```

par(mfrow=c(3,2))
plot(at,type='o',ylab='residuo')
abline(h=0,lty=2)
acf(at,60,drop.lag.0 = TRUE,main="")
pacf(at,60,main="")
qqnorm(at)
qqline(at,col=2)
plot(density(at),xlab='x',main= '')
cpgram(at)

```

```

Box.test(x = at, lag = 4, type="Ljung-Box")
# HW --> p-value = 0.9975, No rechaza H0 luego es RB
# exp_lin --> p-value = 0.9739, No rechaza H0 luego es RB

```

```

Box.test(x = at, lag = 8, type="Ljung-Box")
# HW --> p-value = 0.7712, No rechaza H0 luego es RB
# exp_lin --> p-value = 0.9824, No rechaza H0 luego es RB

```

```

Box.test(x = at, lag = 12, type="Ljung-Box")
# HW --> p-value = 0.8335, No rechaza H0 luego es RB
# exp_lin --> p-value = 0.9503, No rechaza H0 luego es RB

```

```

#----- Forecast HW + ARMA -----
-----#

```

```

# Holt-Winters powered by ARMA model
py.hw <- predict(model.hw, len_yf, prediction.interval = TRUE)
p.arma <- predict(m.2,n.ahead=len_yf)$pred
# tsp(p.arma) <- tsp(py.hw)
py.tot <- py.hw[, 1]+p.arma

```

```

se = predict(m.2,n.ahead=len_yf)$se
# tsp(se) <- tsp(py.hw)

```

```

# Visually evaluate the prediction
par(mfrow=c(1,1))

plot(yf,
      type = "o", col = "black", # Plot first time series
      ylim = c(min(py.hw[, 1], yf), max(py.hw[, 1], yf)),
      xlim = c(time(yf)[1], tail(time(yf), n = 1)),
      main = "Pronósticos: Producción trimestral de cerveza en
Australia",
      xlab = "Año",
      ylab = "Megalitros"
)

lines(py.hw[, 1], type = "o", col = "#0d24f3")
lines(py.tot, type = "o", col = "#D82855")

text_legend <- c("Observados", "Ajustados HW", "Ajustados HW+ARMA")

legend("bottom",
      legend = text_legend, ,
      text.width = strwidth(text_legend)[1] * 3,
      lty = 1,
      col = c("#000000", "#0d24f3", "#D82855"),
      text.font = 1,
      pt.cex = 2,
      cex=1
)

# Metric for quality forecast MAPE, RMSE, UTHEIL
R <- rbind(accuracy(yf, py.tot))[, c(2, 5)]
Utheil <- c(TheilU(yf, py.tot))
R <- c(R, setNames(Utheil, "Utheil"))

```