Curso Técnicas de Pronóstico

Pronóstico de Series de Tiempo con Modelos para tendencia y estacionalidad

Camilo Cabrera Meneses[1] y John Mario Montoya Zapata[2]

Grupo No.2

Recepción: 26 de septiembre de 2022

Se aceptan comentarios y/o discusiones al escrito.

Resumen

Se evalúan 2 modelos para la predicción de la serie de tiempo relacionada con la producción de cerveza en Australia: 1. Exponencial Lineal + Indicadors, 2. Holt-Winters Componentes. Se evidencia que el modelo 1 no es capaz de capturar la variación en la tendencia para los datos de entrenamiento, la cual se da alrededor del año 1980, obteniendo por ejemplo un R2 ajustado de 0.53. El modelo 2 realiza un muy buen ajuste con los datos de entrenamiento presentando, por ejemplo, un R2 ajustado de 0.96. Se selecciona el modelo 2 por sobre el 1 dado que en la parte de pronóstico (3 años) también es superior, presentando un RMSE de 9.78, un MAPE de 2.199 y un U-Theil de 0.023, mientras que el modelo 1 reporta un RMSE de 80.50, un MAPE 15.39 de y un U-Theil de 0.19.

Palabras claves: Holt-Winters, series de tiempo, pronóstico, modelo exponencial.

Introducción

En el presente trabajo se desarrollan 2 modelos de ajuste y predicción para la serie de tiempo denominada "Producción de cerveza en Australia", la cual se encuentra disponible dentro de la librería ffp perteneciente al lenguaje R. Esta serie tiene una resolución trimestral y cuenta con un registro de 54 años

---

[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.

comenzando en el primer trimestre de 1956 hasta el segundo trimestre del año 2010.

Los modelos desarrollados para el ajuste y pronóstico de la serie mencionada son:

1. Modelo exponencial lineal + indicadoras
2. Holt-Winters componentes

Para realizar el proceso de validación cruzada se han seleccionado los últimos 3 años de la serie para evaluar la calidad de los pronósticos. Si bien la recomendación dada para el desarrollo del presente trabajo indica usar 8 trimestres, dada la extensión de la serie (54 años) se decidió tomar un año más para un total de 12 trimestres, que equivalen aproximadamente a un 5% del total de los datos. La figura 1 muestra la serie de producción de cerveza en Australia separada en datos de entrenamiento y testeo.

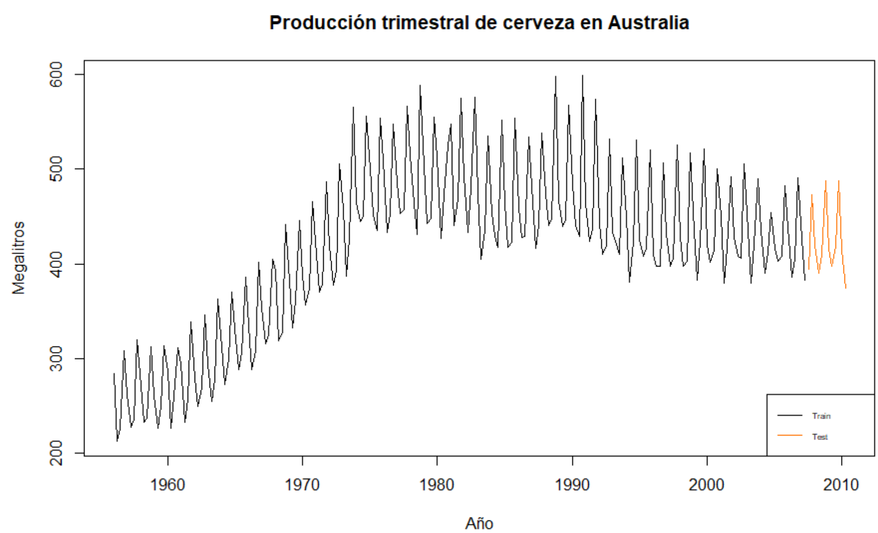

Fig. 1. Serie de producción trimestral de cerveza en Australia dividida en conjuntos de entrenamiento y testeo.

La sección (1) presenta la estimación empleando el modelo 1. La sección (2) presenta la estimación empleando el modelo 2. La sección (3) presenta una comparativa entre 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 de los 2 modelos 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 Modelo Exponencial Lineal + Indicadoras

El primer modelo de componentes empleado para ajuste y pronóstico se expresa a través de la ecuación 1.1

$$Y_t = exp(\beta_0 + \beta_1 t + \sum_{j=1}^{4-1} \delta_i I_t(t)) + \varepsilon_t \tag{1.1}$$

$$log(Y_t) = \beta_0 + \beta_1 t + \sum_{j=1}^{4-1} \delta_i I_t(t) \tag{1.2}$$

El desarrollo de éste requiere determinar la matriz de variables indicadoras estacionales $I_j(t)$. Las dimensiones de esta matriz se relacionan con el periodo de la serie, por lo tanto se debe hacer una inspección de la serie para validar el periodo de la misma y determinar el número de columnas que tendrá la matriz $I_j(t)$. La figura 2 muestra los primeros 3 años de la serie donde podemos observar que el periodo de la serie corresponde a 1 año (4 trimestres).

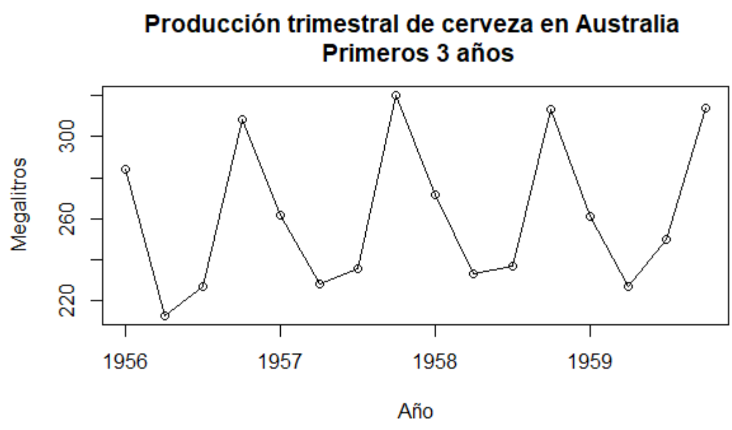

Fig. 2. Inspección visual del periodo presente en la serie de tiempo de producción de cerveza en Australia empleando los primeros 3 años

Para este trabajo se hace uso de la componente estructural en la ecuación 1.1. Inicialmente se ha calculado un modelo logarítmico lineal con componente estacional con variables indicadoras (ver ecuación 1.2) los coeficientes de dicho modelo son significativos a nivel del 5% (Ver Tabla 1), dicho modelo retorna coeficientes ajustados que servirán como entrada para el modelo 1.1.

Tabla.1. Resultados de ajuste del modelo logarítmico-lineal + indicadoras

Parámetro	**Valor Estimado**	**Error Estd.**	**Valor t**	**Valor Pr(>|t|)**
Intercept	5.9234689	0.0284376	208.297	< 2e-16
ti	0.0023465	0.0001795	13.072	< 2e-16
ItQ1	-0.1506184	0.0301924	-4.989	1.31e-06
ItQ2	-0.2718739	0.0301918	-9.005	< 2e-16
ItQ3	-0.2284957	0.0303386	-7.532	1.66e-12

Con estos coeficientes iniciales se desarrolla el ajuste exponencial lineal más variables indicadoras donde nuevamente se han obtenido coeficientes ajustados con significancia a un nivel del 5 % (Ver tabla 2).

Tabla.2. Resultados de ajuste del modelo Exponencial - Lineal + indicadoras

Parámetro	**Valor Estimado**	**Error Estd.**	**Valor t**	**Valor Pr(>|t|)**
Theta1	5.994477	0.026519	226.047	< 2e-16
Theta2	0.001783	0.000172	10.370	< 2e-16
Theta3	-0.151775	0.026601	-5.706	4.11e-08
Theta4	-0.265572	0.028445	-9.336	< 2e-16
Theta5	-0.225603	0.027956	-8.070	6.30e-14

Una vez ajustado el modelo en el periodo de entrenamiento, se ha elaborado la figura 3 en donde podemos visualizar que dicho ajuste no es capaz de capturar el cambio de la tendencia que presenta la serie original alrededor del año 1980 pasando de ser creciente a decreciente, aunque sí muestra la captura de la componente estacional que conserva el periodo original de la serie.

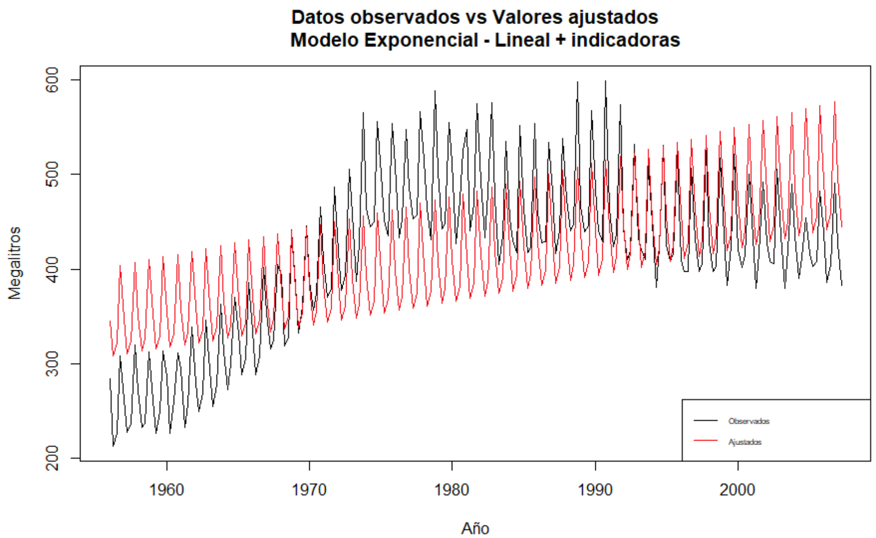

Fig. 3. Ajuste de modelo exponencial-lineal + indicadoras.

Como parte de un ejercicio académico se decidió ajustar un modelo exponencial cuadrático con variables indicadoras para efectuar una comparación contra el modelo 1. El ajuste del modelo exponencial cuadrático para los datos de entrenamiento se muestra en la figura 4. Como se puede observar este ajuste es mejor que el obtenido con el modelo 1, dado que no solo es capaz de capturar la componente estacional de la serie sino también logra modelar el cambio en la tendencia localizada alrededor del año 1980. Este ejercicio se efectuó dado que el modelo exponencial lineal no se ajusta de buena forma a este tipo de series con cambio en su tendencia por lo que se quiso explorar otro modelo dentro de la familia de modelos de componentes aditivas y evaluar su ajuste de manera visual.

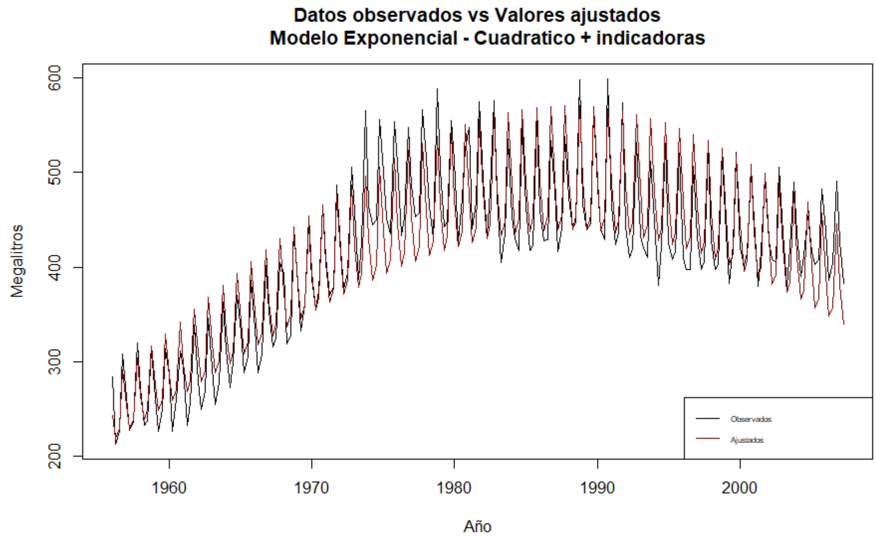

Fig. 4. Ejercicio académico: Ajuste del modelo exponencial-cuadrático más componente estacional con variables indicadoras.

2 Modelo Holt-Winters Componentes

El segundo modelo de componentes empleado para ajuste y pronóstico se expresa a través del conjunto de ecuaciones 2.1, 2.2, 2.3 y 2.4.

$$Y_t = \mu_t + \beta_t + S_t \tag{2.1}$$
$$\mu_t = \alpha\left(Y_t - S_{t-s}\right) + (1 - \alpha)\left(\mu_{t-1} + \beta_{t-1}\right) \tag{2.2}$$
$$\beta_t = \beta\left(\mu_t - \mu_{t-1}\right) + (1 - \beta)\beta_{t-1} \tag{2.3}$$
$$S_t = \gamma\left(Y_t - \mu_t\right) + (1 - \gamma)S_{t-s} \tag{2.4}$$

En el ajuste de este modelo se ha empleado la función *HoltWinters* perteneciente a la librería *Stats* del lenguaje de programación R. Dicha función es capaz de auto-determinar los coeficientes de ajuste α, β y γ minimizando el error de predicción al cuadrado; además para la estimación de la componente estacional la función está configurada para un modelo aditivo.

Una vez realizado el ajuste, los valores de los parámetros obtenidos son:

α: 0,2139425
β: 0,1340575
γ: 0,3220664

Bajo esta parametrización se presenta la figura 4 en la cual es apreciable que el modelo realiza un buen ajuste tanto en la componente de tendencia como en la componente estacional. Es de especial interés notar que el modelo captura el cambio de tendencia en los datos, pasando de una serie creciente a una decreciente.

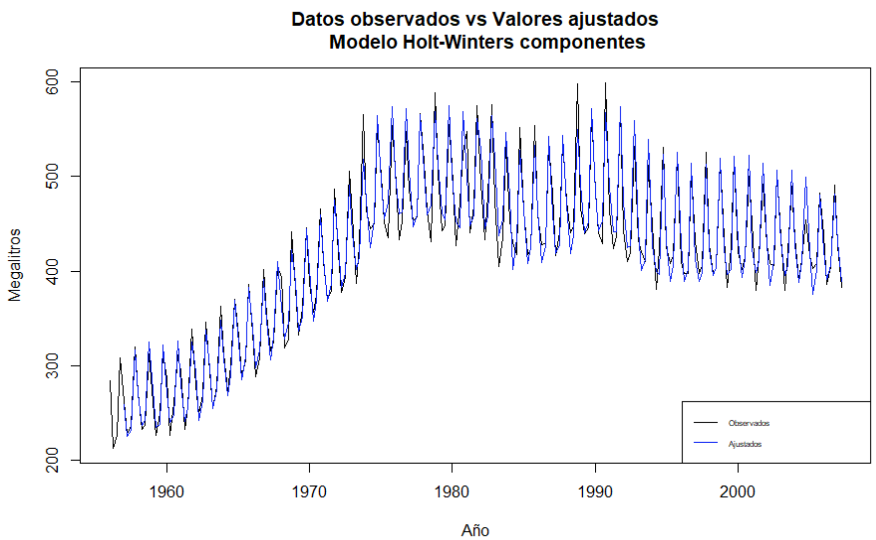

Fig. 5. Ajuste del modelo Holt-Winters componentes

3 Estadísticos de calidad de ajuste: MSE, AIC, BIC y R2-Ajustado

En esta sección se presentan los estadísticos de ajuste para los modelos 1 y 2 desarrollados en la sección (1) y (2), dichos valores se muestran en la tabla 3.

Tabla 3. Estadísticos calidad de ajuste para los modelos exponencial lineal + indicadoras VS Holt-Winters Componentes

Modelo	MSE	AIC	BIC	R2-Ajustado
1	3635.29148	8.20811	8.24042	0.52934
2	257.72383	5.56635	5.614809	0.96663

Estos valores corroboran el hecho de que el ajuste realizado a través del modelo 2 (Holt-Winters Componentes) es mucho mejor que el obtenido por el modelo 1 (exponencial lineal + indicadoras), para ello podemos ver que el modelo 2 presenta un MSE, un AIC y un BIC más pequeño en el modelo 2 además de tener un R2 de ajuste más alto. Estos valores son una consecuencia de los comportamientos mencionados en las secciones 1 y 2, en las cuales se hizo referencia a que el modelo 1 no es capaz de capturar las variaciones en las tendencias de los datos originales.

4 Pronóstico y medidas de precisión en pronósticos

Como parte final del proceso de modelado se han tomado ambos modelos y con ellos se han realizado pronósticos en un horizonte de 3 años (12 trimestres), para ello se emplea la función *predict* obteniendo los pronósticos mostrados en las figuras 6 y 7.

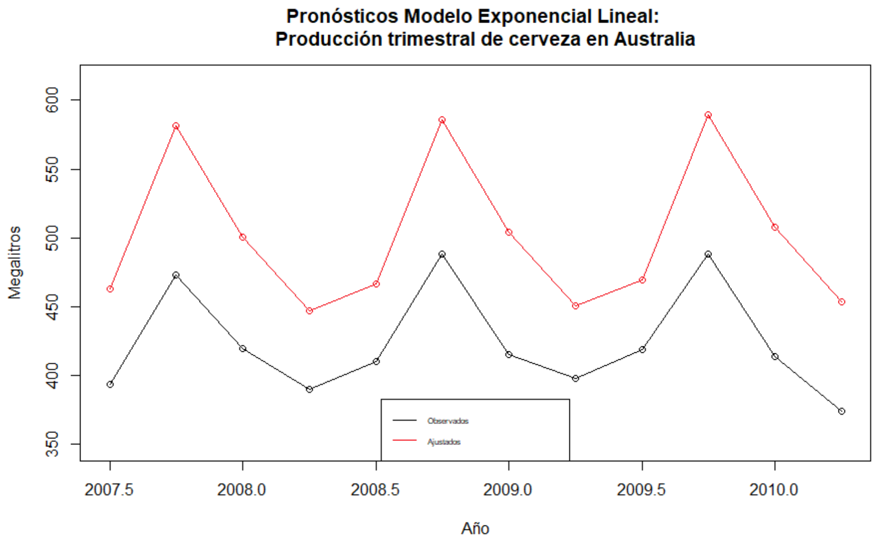

Fig. 6 Pronóstico en un horizonte de 3 años empleando el modelo exponencial-lineal + indicadoras.

Como es apreciable, el modelo que visualmente realizó los mejores pronósticos fue Holt-winters, dicha apreciación visual es corroborada a través de los estadísticos de calidad de pronóstico presentados en la tabla 4. Como podemos observar, el MAPE y el U-Theil tienen una cuantía menor para el modelo 2, en particular el bajo valor de U-Theil por debajo de 1 nos confirma que nuestros pronósticos no son "ingenuos", además se cuenta con un RMSE de 9.78 en el modelo 2 mientras que el modelo 1 alcanza una valor de 80.50.

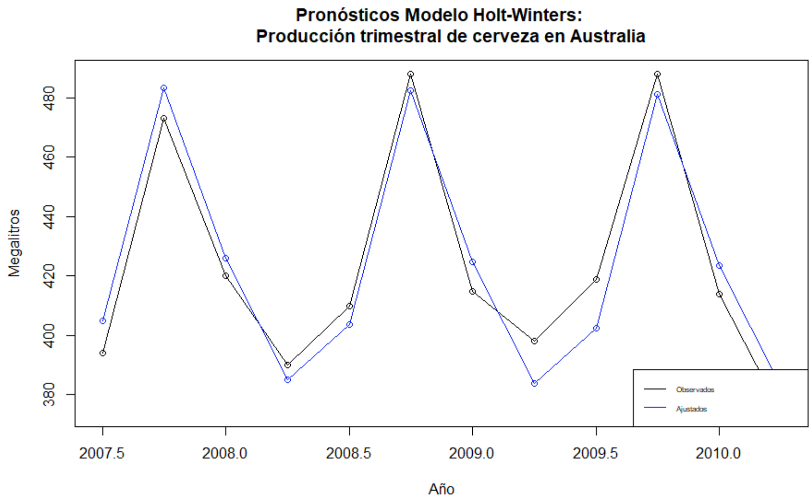

Fig. 7. Pronóstico en un horizonte de 3 años empleando el modelo Holt-Winters Componentes.

Con tales características se confirma que entre estos 2 modelos el que mejor realiza pronósticos es el Holt-Winters.

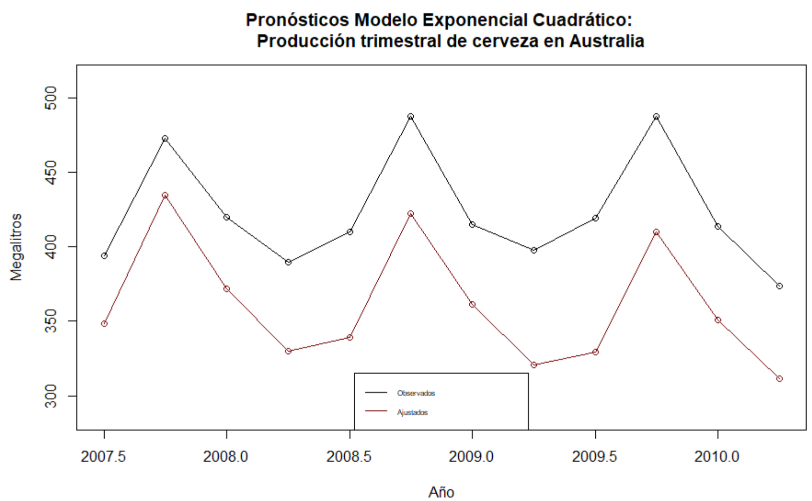

Fig. 8 Pronóstico en un horizonte de 3 años empleando el modelo exponencial-cuadrático + indicadoras.

Finalmente, y como parte del ejercicio de mejora del modelo planteado en la sección (1), se presenta la figura 8 con los pronósticos realizados por el modelo exponencial-cuadrático + indicadoras. Es apreciable que en la parte de pronóstico dicho modelo tampoco realiza un buen trabajo, presentando una subestimación de los valores que en definitiva están lejos de superar al modelo Holt-Winters, por lo cual se corrobora la idea de que un modelo que ajuste bien en los datos de entrenamiento no necesariamente lo hace bien en la parte de pronóstico.

Referencias

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

Anexo

```r
# Development of project No. 1 Forecasting Techniques
# Authors: Camilo Cabrera - John Mario Montoya

# Installing libraries
install.packages("ggplotw")
install.packages("fpp2")
install.packages("DescTools")

# 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)
```

```r
    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)) # frecuency = 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
```

```r
# 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
)

# Component using moving averages
components_yi <- decompose(yi)
plot(components_yi)

#----------------------------------------------------------------------------#
#---------------- 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))

M.exp_lin <- medidas(mod.exp_lin, yi, 2)
```

```r
(M.exp_lin)

yhat_exp_lin <- fitted(mod.exp_lin)
yhat_exp_lin <- ts(yhat_exp, frequency = 4, start = c(1956, 01))

# Visually evaluate the fitted
plot(yi,
    type = "l", col = "#000000", # Plot first time series
    ylim = c(min(yi, yhat_exp_lin), max(yi, yhat_exp_lin)),
    xlim = c(time(yi)[1], tail(time(yi), n = 1)),
    main = "Datos observados vs Valores ajustados
    Modelo Exponencial - Lineal + indicadoras",
    xlab = "Año",
    ylab = "Megalitros"
)
lines(yhat_exp_lin, , type = "l", col = "#f30914")

# Add legend to plot
text_legend <- c("Observados", "Ajustados")
legend("bottomright",
    legend = text_legend, ,
    text.width = strwidth(text_legend)[1] * 2,
    lty = 1, ,
    col = c("#000000", "#f30914")
)

#-----------------------------------#
#------------ Forecast -------------#
#-----------------------------------#

Itf <- seasonaldummy(yi, len_yf)
tf <- seq(T + 1, T + len_yf, 1)

tf2 <- tf * tf
Xtf <- cbind(rep(1, len_yf), tf, Itf)

pron_exp_lin <- predict(mod.exp_lin, data.frame(Xt = I(Xtf)))
y_pron_explin <- ts(pron_exp_lin, start = time(yf)[1], frequency = 4)

plot(yf,
    type = "o", col = "black", # Plot first time series
    ylim = c(min(y_pron_explin, yf), max(y_pron_explin, yf)),
    xlim = c(time(yf)[1], tail(time(yf), n = 1)),
```

```r
    main = "Pronósticos Modelo Exponencial Lineal:
    Producción trimestral de cerveza en Australia",
    xlab = "Año",
    ylab = "Megalitros"
)

lines(y_pron_explin, type = "o", col = "#f30914")

# Add legend to plot
text_legend <- c("Observados", "Ajustados")
legend("bottom",
    legend = text_legend, ,
    text.width = strwidth(text_legend)[1] * 2,
    lty = 1, ,
    col = c("#000000", "#f30914")
)

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


#-----------------------------------------------------------------------------#
#-------------Test Model No1 Variation: Quadratic - Exponential --------------#
#-----------------------------------------------------------------------------#

mod.llin2 <- lm(lyi ~ ti + ti2 + It)

# Seasonal linear exponential model
T <- length(yi)
Xt <- cbind(rep(1, T), ti, ti2, It)
Ds <- data.frame(yi, Xt)
theta.0 <- mod.llin2$coefficient

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

M.exp_q <- medidas(mod.exp_q, yi, 2)

(M.exp_q)
```

```r
yhat_exp_q <- fitted(mod.exp_q)
yhat_exp_q <- ts(yhat_exp_q, frequency = 4, start = c(1956, 01))

# Visually evaluate the fitted
plot(yi,
    type = "l", col = "#000000", # Plot first time series
    main = "Datos observados vs Valores ajustados
    Modelo Exponencial - Cuadratico + indicadoras",
    xlab = "Año",
    ylab = "Megalitros"
)
lines(yhat_exp_q, , type = "l", col = "#790a0a")

# Add legend to plot
text_legend <- c("Observados", "Ajustados")
legend("bottomright",
    legend = text_legend, ,
    text.width = strwidth(text_legend)[1] * 2,
    lty = 1, ,
    col = c("#000000", "#790a0a")
)

#-----------------------------------#
#------------ Forecast -------------#
#-----------------------------------#

Itf <- seasonaldummy(yi, len_yf)
tf <- seq(T + 1, T + len_yf, 1)

tf2 <- tf * tf
Xtf <- cbind(rep(1, len_yf), tf, tf2, Itf)

pron_exp_q <- predict(mod.exp_q, data.frame(Xt = I(Xtf)))
y_pron_exp_q <- ts(pron_exp_q, start = time(yf)[1], frequency = 4)

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

```r
lines(y_pron_exp_q, type = "o", col = "#790a0a")

# Add legend to plot
text_legend <- c("Observados", "Ajustados")
legend("bottom",
    legend = text_legend, ,
    text.width = strwidth(text_legend)[1] * 2,
    lty = 1, ,
    col = c("#000000", "#790a0a")
)


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

#-------------------- Model with auto-tunning of parameters-------------------#

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

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

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

# Visually evaluate the fitted
plot(yi,
    type = "l", col = "#000000", # Plot first time series
    main = "Datos observados vs Valores ajustados
     Modelo Holt-Winters componentes",
    xlab = "Año",
    ylab = "Megalitros"
)
lines(Yt_hat, , type = "l", col = "#0d24f3")

# Add legend to plot
text_legend <- c("Observados", "Ajustados")
```

```r
legend("bottomright",
    legend = text_legend, ,
    text.width = strwidth(text_legend)[1] * 2,
    lty = 1, ,
    col = c("#000000", "#0d24f3")
)

#-----------------------------------#
#------------ Forecast -------------#
#-----------------------------------#


y_pron <- predict(model, len_yf, prediction.interval = TRUE)

plot(yf,
    type = "o", col = "black", # Plot first time series
    ylim = c(min(y_pron[, 1], yf), max(y_pron[, 1], yf)),
    xlim = c(time(yf)[1], tail(time(yf), n = 1)),
    main = "Pronósticos Modelo Holt-Winters:
    Producción trimestral de cerveza en Australia",
    xlab = "Año",
    ylab = "Megalitros"
)
lines(y_pron[, 1], , type = "o", col = "#0d24f3")

text_legend <- c("Observados", "Ajustados")
legend("bottomright",
    legend = text_legend, ,
    text.width = strwidth(text_legend)[1] * 2,
    lty = 1, ,
    col = c("#000000", "#0d24f3")
)

# Metric for quality forecast MAPE, RMSE, UTHEIL
R <- rbind(accuracy(yf, y_pron[, 1]))[, c(2, 5)]
Utheil <- c(TheilU(yf, y_pron[, 1]))
R <- c(R, setNames(Utheil, "Utheil"))
```