

Práctica 2

Juan Cantero Jiménez

3/30/2022

1 Introducción.

En el presente trabajo se aplica la metodología de alisado exponencial a una serie temporal, AustriaT, obtenida de Eurostat que contiene las exportaciones totales de Austria hacia los 27 países de la Unión Europea (con Reino Unido ya excluida). Dicha serie mensual comienza en enero de 1999 y finaliza en diciembre de 2020. También se propone un algoritmo de optimización de parámetros mediante búsqueda por cuadrícula, grid search.

2 Lectura de los datos.

```
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo

library(ggplot2)
austria <- read.csv2("AustriaT.csv")
austriat <- ts(austria, start = c(1999, 1), frequency = 12)
```

3 Ajusta del modelo a los datos y descripción.

Para obtener el modelo de alisado exponencial más adecuado a los datos se puede usar la función ets() del paquete forecast. Notese que este es el modelo más óptimo según el criterio de selección usado internamente por la función. Estos modelos son obtenidos maximizando la función de verosimilitud, seleccionando entre las distintas variantes de los modelos mediante el criterio de información Akaike corregido para muestras pequeñas

```
austriatETS <- ets(austriat, damped = F)
summary(austriatETS)
```

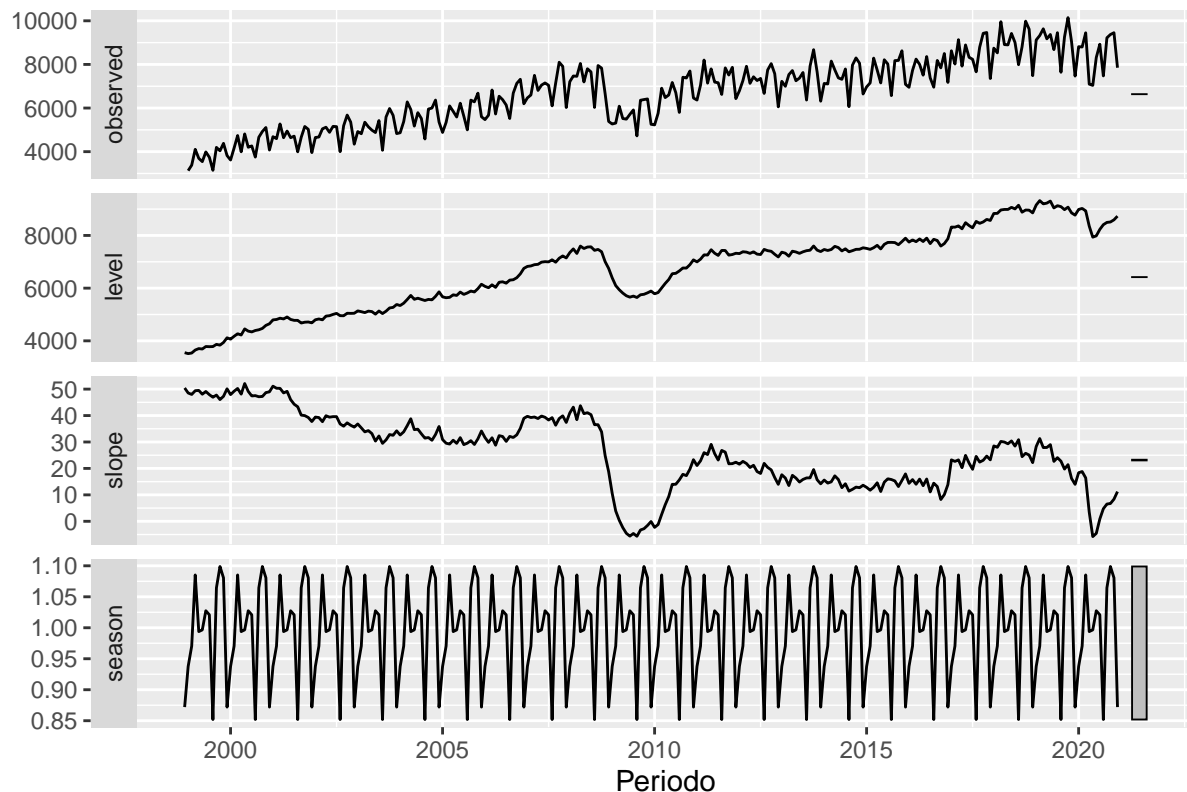
```
## ETS(M,A,M)
##
## Call:
## ets(y = austriat, damped = F)
##
## Smoothing parameters:
##   alpha = 0.3277
##   beta  = 0.0071
##   gamma = 1e-04
##
```

```
## Initial states:
## l = 3552.0793
## b = 50.44
## s = 0.872 1.0806 1.0989 1.0645 0.8519 1.0209
##      1.0275 0.9966 0.9938 1.0852 0.9706 0.9376
##
## sigma: 0.0563
##
##      AIC      AICc      BIC
## 4606.440 4608.928 4667.232
##
## Training set error measures:
##      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -23.63791 377.9256 288.9148 -0.599816 4.421472 0.5612612 0.0401729
```

El modelo obtenido por la función es ETS(M, A, M) en otras palabras, un modelo de alisado exponencial de Holt-Winters multiplicativo o también $y_{t+1} = (l_t + b_t)s_{t+1-m}(1 + \epsilon_{t+1})$. Se ha forzado el modelo a evitar tendencia amortiguada puesto que en pruebas no mostradas, el valor obtenido para ϕ era de 0.98, lo que aconseja volver a realizar el proceso de selección restringiendo a los que no posea amortiguamiento. Los valores óptimos de los parámetros que describen el modelo $\alpha = 0.3277$, $\beta = 0.0071$ y $\gamma = 1e10^{-4}$, los valores bajos en estos dos últimos, indican que la pendiente y la estacionalidad modifican su valor muy lentamente. En consecuencia, el modelo tiene pendiente y estacionalidad, pero estas permanecen constantes en el tiempo. Esto se puede comprobar si se atiende a la Figura 1.

```
autoplot(austriatETS,
  xlab = "Periodo",
  main = "Figura 1. Componentes del modelo óptimo para AustriaT")
```

Figura 1. Componentes del modelo óptimo para AustriaT



Si nos centramos ahora en la calidad del modelo, el valor MAPE del 4.42 % indica un buen ajuste del modelo a los datos. El criterio MASE arroja un valor de 0.56 lo que indica que el modelo reduce un 0.44 el error del método ingenuo con estacionalidad.

También podemos observar los últimos valores que se han obtenido para el nivel, la pendiente y la estacionalidad.

```
TT <- nrow(austriatETS$states)
austriatETS$states[TT,]
```

```
##          l          b          s1          s2          s3          s4
## 8733.0830233 11.2335838 0.8719829 1.0805667 1.0988811 1.0644465
##          s5          s6          s7          s8          s9          s10
## 0.8519491 1.0208858 1.0274256 0.9965816 0.9938097 1.0852142
##          s11          s12
## 0.9705503 0.9376174
```

Según estos valores, marzo es el mes con más exportaciones, teniendo 1.098 veces más respecto a la media anual, en contraposición mayo es el mes con menos exportaciones teniendo 0.851 veces menos respecto a la media anual.

4 Predicción.

La predicción puede ser llevada a cabo haciendo uso de la función `forecast()`, solo se muestra el primer año:

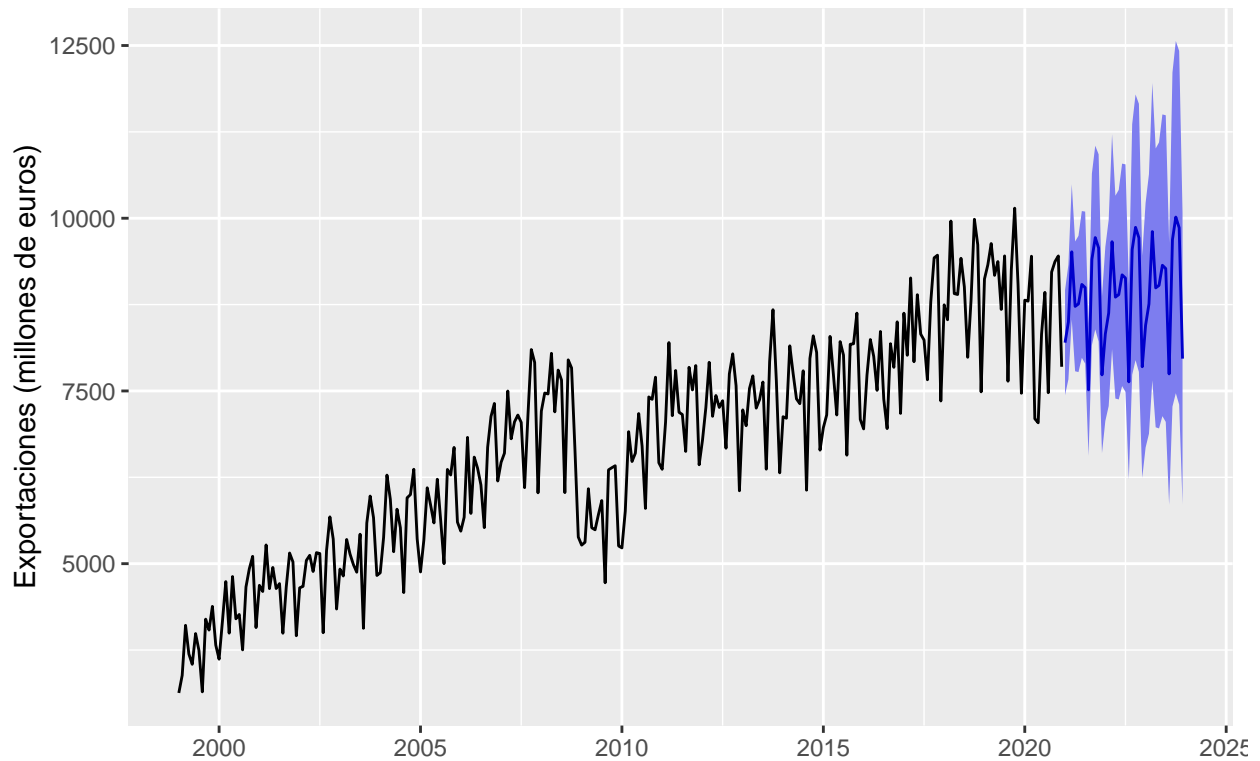
```
austriatETSpre <- forecast(austriatETS, h = 36, level = 0.90)
austriatETSpre
```

```
##          Point Forecast      Lo 90      Hi 90
## Jan 2021      8198.824 7439.374 8958.273
## Feb 2021      8497.702 7667.580 9327.823
## Mar 2021      9513.838 8536.876 10490.800
## Apr 2021      8723.679 7784.640 9662.719
## May 2021      8759.206 7773.275 9745.136
## Jun 2021      9041.843 7979.911 10103.774
## Jul 2021      8995.758 7895.487 10096.029
## Aug 2021      7516.706 6560.915 8472.498
## Sep 2021      9403.518 8162.368 10644.668
## Oct 2021      9720.064 8390.257 11049.871
## Nov 2021      9570.204 8214.804 10925.603
## Dec 2021      7732.645 6600.275 8865.015
## [ reached 'max' / getOption("max.print") -- omitted 24 rows ]
```

gráficamente, se puede obtener

```
autoplot(austriatETSpre,
         xlab = "",
         ylab = "Exportaciones (millones de euros)",
         main = "Figura 2. Exportaciones de Austria y predicción")
```

Figura 2. Exportaciones de Austria y predicción



Véase como la longitud del intervalo de predicción aumenta al alejarnos del horizonte de predicción.

5 Evaluación de la calidad de las previsiones con origen de previsión móvil.

A continuación, se procede a evaluar la calidad de las predicciones haciendo uso de la metodología de origen de predicción móvil. Se asumirá que se requieren de 10 años para hacer una buena estimación $k = 120$. Además se evaluará la calidad de las predicciones a tres años mediante el criterio de calidad MAPE.

```
k <- 120
h <- 36
TT <- length(austriat)
s <- TT - k - h

mapeEts <- matrix(NA, s + 1, h)
for (i in 0:s) {
  train.set <- subset(austriat, start = i + 1, end = i + k)
  test.set <- subset(austriat, start = i + k + 1, end = i + k + h)

  fit <- ets(train.set, model = "MAM", damped = FALSE)
  fcast <- forecast(fit, h = h)
  mapeEts[i + 1,] <- 100*abs(test.set - fcast$mean)/test.set
}

mapeEts <- colMeans(mapeEts)
round(mapeEts, 2)

## [1] 5.35 5.52 5.78 6.78 7.24 7.83 8.95 9.21 9.70 10.90 10.75 10.74
```

```
## [13] 12.69 13.22 13.40 14.73 14.37 15.11 16.45 16.17 16.73 17.48 16.72 17.77
## [25] 18.60 18.90 20.18 20.62 21.53 21.98 22.35 22.84 23.50 23.77 23.73 24.60
```

Como se puede observar, el MAPE extra-muestral a un periodo vista, 5.35 %, y el intra-muestral, 4.42 %, son similares, aunque el primero es superior. Conforme nos alejamos del horizonte de predicción, la calidad de las predicciones va disminuyendo, llegando a un MAPE de 24.60 % para la predicción a 36 meses vista.

6 Residuo e identifica la presencia de valores extremos.

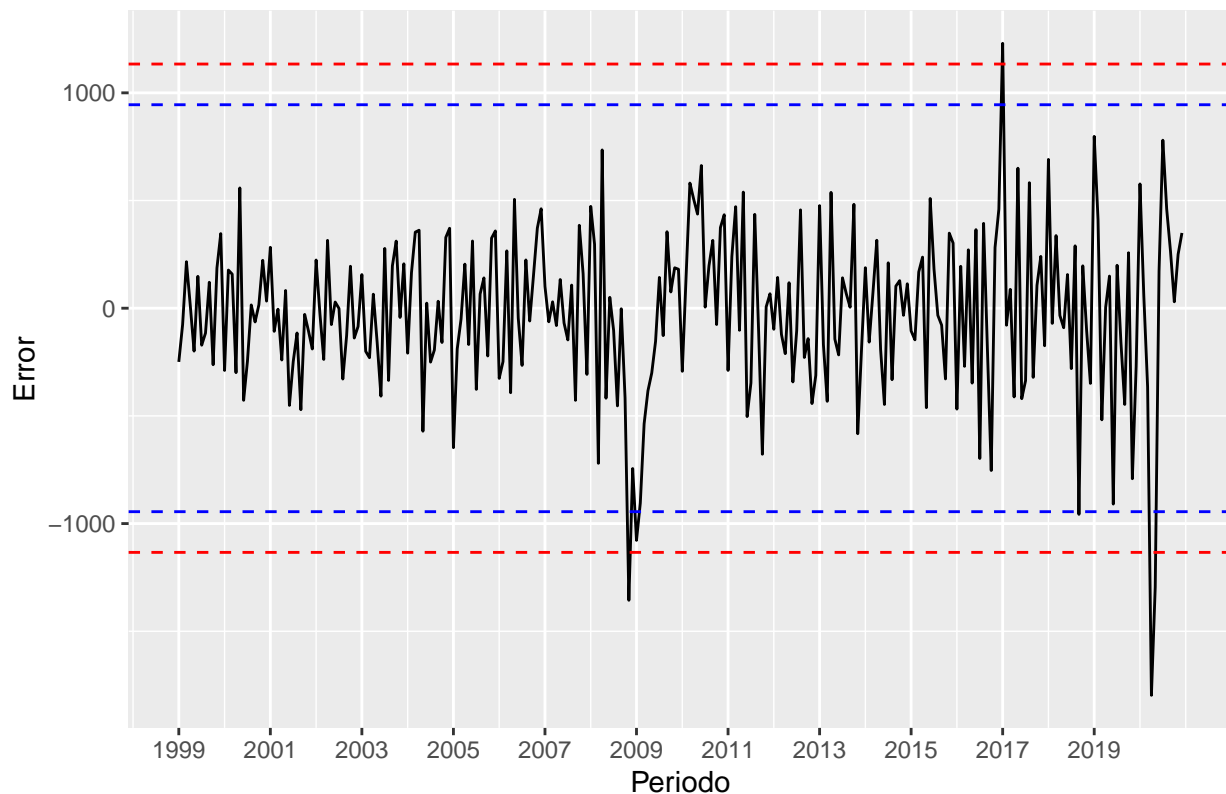
El error del modelo puede ser obtenido mediante el código

```
error <- residuals(austriatETS, type = "response")
sderror <- sd(error)

autoplot(error, series="Error",
          colour = "black",
          xlab = "Periodo",
          ylab = "Error",
          main = "Figura 3: Error + Intervención") +
  geom_hline(yintercept = c(-3, -2.5, 2.5, 3)*sderror,
            colour = c("red", "blue", "blue", "red"), lty = 2) +
  scale_x_continuous(breaks= seq(1999, 2020, 2))
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

Figura 3: Error + Intervención



En la Figura 3 se puede observar la presencia de tres valores atípicos que superan las 3 desviaciones típicas, estos corresponden a 2009, 2017 y 2020. También se pueden observar otros valores atípicos, que en este caso

se sitúan próximos a las 2 desviaciones típicas, alrededor del año 2019.

7 Comparación del modelo obtenido con el método ingenuo con estacionalidad.

El método sencillo con estacionalidad se puede obtener con la función `snaive()`

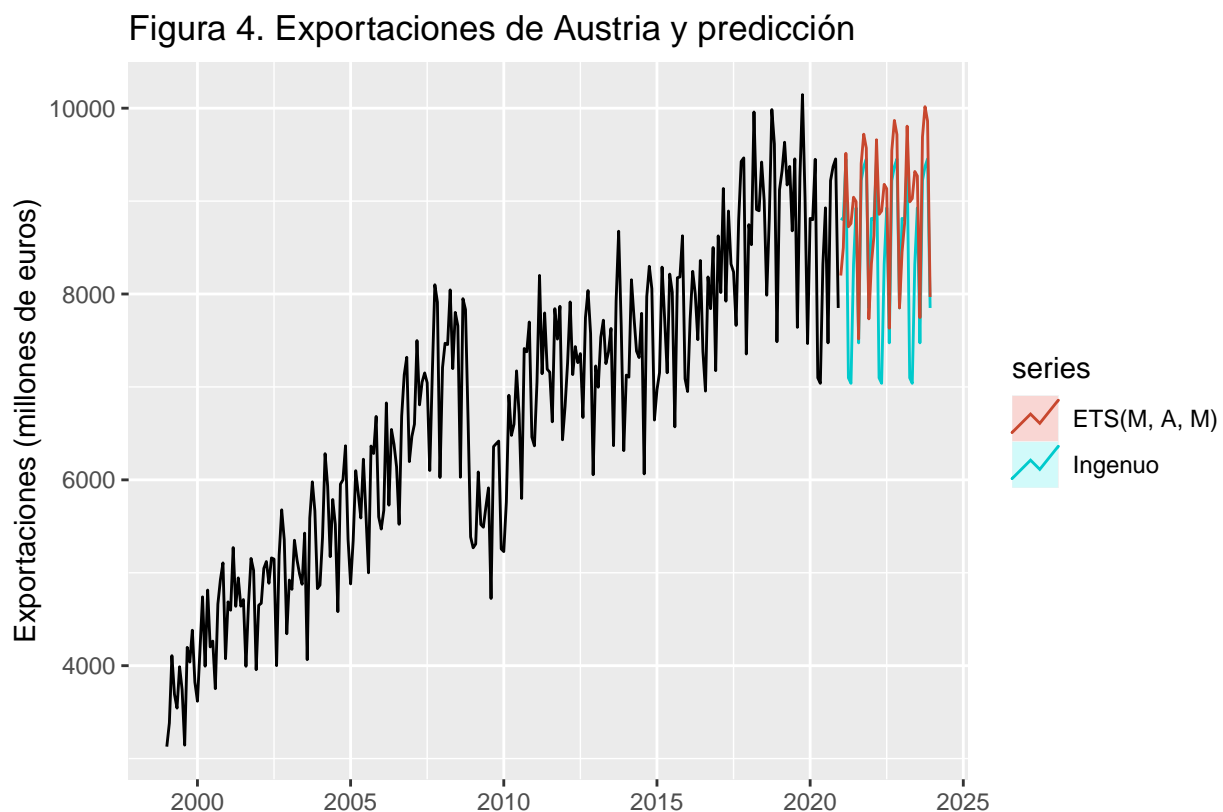
```
austria.naiv.pred <-snaive(austriat, h = 36)
accuracy(austria.naiv.pred)
```

```
##                ME      RMSE      MAE      MPE      MAPE  MASE      ACF1
## Training set 224.792 693.4888 514.76 3.288469 7.885207    1 0.6222893
```

```
accuracy(austriatETSpre)
```

```
##                ME      RMSE      MAE      MPE      MAPE  MASE      ACF1
## Training set -23.63791 377.9256 288.9148 -0.599816 4.421472 0.5612612 0.0401729
```

```
autoplot(austriat,
          xlab = "",
          ylab = "Exportaciones (millones de euros)",
          main = "Figura 4. Exportaciones de Austria y predicción") + autolayer(austria.naiv.pred, series = "naive")
```



Resulta aparente la reducción en la calidad de predicción del método ingenuo con estacionalidad frente al obtenido mediante la función `ets()`. Cabe destacar, el valor de los criterios ME y MPE indica un error sistemático en la predicción, sesgo. Así se puede concluir que el modelo obtenido es superior al modelo ingenuo con estacionalidad.

8 Grid Search para la obtención del modelo con mejor capacidad de predicción.

A continuación se propone una metodología para la obtención de los parámetros más optimo para la función ETS de cara a la creación de un modelo optimo para predicción. La función `create_matrix` creará tantas cadenas de texto como distintas combinaciones de parámetros (estos son `model`, `damped`, `lambda`, `biasadj`, `opt.crit` y `ic`, aunque se pueden seleccionar menos) existan, en estas se detalla la llamada a la función `ets()`. El resultado de `create_matrix` es una lista con las llamadas a la función `ets` que será introducida en la función `grid_search`. Esta función ejecuta las llamadas y obtiene distintos parámetros de calidad, además de una evaluación de la predicción mediante origen de predicción móvil, usando como indicador de calidad MAPE extra-muestral. El resultado de `grid_search` es un matriz con tantas filas como llamadas a la función `ets` se hagan, y con variables que indican los parámetros de calidad del modelo. Debido al coste computacional del análisis realizado, la función `grid_search` ha sido paralelizada de forma que pueda usar mas cantidad de nucleos de CPU. Esta implementación paralelizada es optima para sistemas Linux. Por último destacar que la función `ets` genera errores cuando para las condiciones introducidas no se puede crear un modelo, este error se ha manejado gracias a una sentencia `try`.

```
create_matrix <- function(model = TRUE,
                           damped = TRUE,
                           lambda = TRUE,
                           biasadj = TRUE,
                           opt.crit = TRUE,
                           ic = TRUE){

  matrix_to_functions <- function(x){
    expressions <- c("ets(serie,")
    arguments <- c("model", "lambda", "damped", "biasadj", "opt.crit", "ic")
    for (arg in arguments){
      sen <- try({x[[arg]]}, silent = T)
      if(!( class(sen) == "try-error")){
        if( arg == "opt.crit" | arg == "ic" | arg == "model"){
          expressions <- c(expressions, arg, "=", "\"" ,x[[arg]], "\"" ,",")
        }else{
          expressions <- c(expressions, arg, "=", x[[arg]], ",")
        }
      }
    }
    result<- paste(expressions, collapse = "")
    result<- substring(result,1, nchar(result)-1)
    result<- paste(result, ")", sep = "")
    return(result)
  }

  options <- list()
  if (model){
    options[["level"]] <- c("A", "M", "N")
    options[["tendency"]] <- c("A", "M", "N")
    options[["error"]] <- c("A", "M", "N")
  }
  if (ic){
    options[["ic"]] <- c("aicc", "aic", "bic")
  }
  if (damped){
    options[["damped"]] <- c("TRUE", "FALSE")
  }
}
```

```

if (biasadj){
  options[["biasadj"]] <- c("TRUE", "FALSE")
}
if (opt.crit){
  options[["opt.crit"]] <- c("lik", "amse", "mse", "sigma", "mae")
}
if (lambda){
  options[["lambda"]] <- c("0", "NULL")
}
res <- expand.grid(options)
if (model){
  models <- apply(res, 1, function(x) paste(x[ c("level", "tendency", "error")], collapse = ""))
  result <- res[, -which(colnames(res) == "level" | colnames(res) == "tendency" | colnames(res) == "error")]
  result$model <- models
  result <- as.matrix(result)
}else{
  result <- as.matrix(res)
}
return(as.list(apply(result, 1, matrix_to_functions)))
}
grid_search <- function(list_functions, datos){
  require(parallel)
  require(forecast)
  analisis <- function(x, data, k = 120, h = 36){
    serie <- data
    res<-try(eval(parse(text = x)), silent=TRUE)
    if (class(res) == "ets"){
      TT <- length(data)
      s <- TT - k - h
      mapeEts <- matrix(NA, s + 1, h)
      for (i in 0:s) {
        train.set <- subset(data, start = i + 1, end = i + k)
        test.set <- subset(data, start = i + k + 1, end = i + k + h)
        serie <- train.set
        fit<-try(eval(parse(text = x)), silent=TRUE)
        fcast <- forecast(fit, h = h)
        mapeEts[i + 1,] <- 100*abs(test.set - fcast$mean)/test.set
      }
      mapeEts <- colMeans(mapeEts)
      result <- cbind(accuracy(res), matrix(mapeEts,
                                             nrow = 1,
                                             ncol= h,
                                             byrow = T))
      colnames(result) <- c(colnames(accuracy(res)), 1:h)
      return(result)
    }else{
      return(rep(NA, 7+h))}
  }
  previo <- parallel::mclapply(list_functions,
                                function(x, y) analisis(x, data = y),
                                y = datos,
                                mc.cores = 11)

```



```

previo <- do.call(rbind.data.frame, previo)
result <- cbind(unlist(list_functions), previo)
colnames(result) <- c("model", colnames(previo))
return(result)

}

```

Así si ejecutamos ambas rutinas descritas anteriormente con el fin de buscar aquel modelo que posea menor MAPE extra-muestral, para el mes 36 desde el horizonte de previsión, obtenemos

```

functions <- create_matrix()
start <- Sys.time()
modelos <- grid_search(functions, austriat)

## Loading required package: parallel

end <- Sys.time()
time <- end - start
time

## Time difference of 22.05005 mins

modelos_fil <- modelos[!is.na(modelos$ME), ]

modelos_fil[which.min(abs(modelos_fil$"36")),]

##                                                                 model
## Training set43 ets(serie,model="ANA",lambda=0,damped=FALSE,biasadj=TRUE,opt.crit="amse",ic="aicc")
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set43 20.62544 388.0509 300.7479 0.08964217 4.591972 0.5842487
##           ACF1      1      2      3      4      5      6
## Training set43 -0.1986366 4.649172 4.391013 4.19656 5.075951 4.816458 4.841182
##           7      8      9      10      11      12      13
## Training set43 5.320441 5.374006 5.242788 6.237764 5.731251 5.602738 6.830604
##           14      15      16      17      18      19      20
## Training set43 6.592114 6.885114 7.421992 6.816226 7.436794 7.543843 7.351129
##           21      22      23      24      25      26      27
## Training set43 7.710254 7.796245 7.428694 7.885475 7.98363 7.738501 8.479761
##           28      29      30      31      32      33      34
## Training set43 8.388225 8.398176 8.763193 8.567318 8.707618 9.027843 8.542575
##           35      36
## Training set43 8.589501 8.919007

```

que el modelo optimo de cara a la previsión es ets(A,N,A) con transformación logarítmica y sin amortiguación. Además el criterio de optimización es el AMSE y el criterio de información AICC. Como se puede observar, este modelo ofrece un MAPE de 8.91 % muy inferior al ofrecido por el modelo ets(M,A,M) 24.60 % obtenido tras aplicar la función ets con los parámetros por defecto. Se puede concluir que la función ets con el conjunto de parametros por defecto no obtiene el modelo que realice las mejores predicciones. Cabe destacar que esta búsqueda para el modelo que posea menor MAPE extra-muestral para el periodo 36 desde el horizonte de previsión puede ser repetido para otro periodo u parametro de calidad intra-muestral.