PRÁCTICA 3: MÉTODOS DE ALISADO EXPONENCIAL

PROCESOS ESTOCÁSTICOS Y SERIES TEMPORALES

GRADO EN CIENCIA E INGENIERÍA DE DATOS

Sumario: En esta práctica veremos cómo realizar el análisis y predicción de series temporales con métodos de alisado exponencial, tales como el alisado exponencial simple, método de Holt o el método de Holt-Winters, incidiendo en la adecuación de cada uno de ellos según las caractarísticas de la serie en estudio. Además, veremos cómo seleccionar y estimar los modelos ETS (Error, Trend, Seasonal), así como obtener intervalos de predicción.

1. Resumen de los métodos de alisado exponencial y su uso con R

Recordemos que los métodos de alisado exponencial se clasifican dependiendo de si la serie en estudio presenta o no componente estacional.

1. **Si la serie no presenta componente estacional:**

   (a) Método de alisado exponencial simple: se usa cuando la tendencia se considera constante (localmente). En este caso, interviene un único parámetro de alisado (**alpha**) y se tiene la siguiente fórmula recurrente para estimar el nivel de la serie en cada instante:

   $$\boxed{a_t = \alpha x_t + (1-\alpha)a_{t-1} \qquad \forall t, \text{ con } \alpha \in [0,1]}$$

   (b) Método de Holt: se usa cuando la tendencia se considera lineal (localmente). En este caso, intervienen dos parámetros de alisado (**alpha** y **beta**) y se tienen las siguientes fórmulas recurrentes para estimar el nivel de la serie y la pendiente de la tendencia en cada instante:

   $$\boxed{a_t = \alpha x_t + (1-\alpha)\left(a_{t-1} + b_{t-1}\right)}$$

   $$\boxed{b_t = \beta\left(a_t - a_{t-1}\right) + (1-\beta)b_{t-1}}$$

2. **Si la serie presenta componente estacional:**

   (a) Método de Holt-Winters para modelos multiplicativos: se usa cuando la serie presenta estacionalidad, la tendencia se considera lineal localmente y ambas componentes tendencia y estacional se integran con esquema multiplicativo. En este caso, intervienen tres parámetros de alisado (**alpha**, **beta** y **gamma**) y se tienen las siguientes fórmulas recurrentes para estimar el nivel de la serie, la pendiente de la tendencia y la estacionalidad en cada instante:

   $$\boxed{a_t = \alpha \frac{x_t}{S_{t-L}} + (1-\alpha)\left(a_{t-1} + b_{t-1}\right)}$$

   $$\boxed{b_t = \beta\left(a_t - a_{t-1}\right) + (1-\beta)b_{t-1}}$$

   $$\boxed{S_t = \gamma \cdot \frac{x_t}{a_t} + (1-\gamma)\, S_{t-L}}$$

(b) Método de Holt-Winters para modelos aditivos: se usa cuando la serie presenta estacionalidad, la tendencia se considera lineal localmente y ambas componentes tendencia y estacional se integran con esquema aditivo. En este caso, intervienen tres parámetros de alisado (**alpha**, **beta** y **gamma**) y se tienen las siguientes fórmulas recurrentes para estimar el nivel de la serie, la pendiente de la tendencia y la estacionalidad en cada instante:

$$\boxed{a_t = \alpha\left(x_t - S_{t-L}\right) + (1-\alpha)\left(a_{t-1} + b_{t-1}\right)}$$

$$\boxed{b_t = \beta\left(a_t - a_{t-1}\right) + (1-\beta)b_{t-1}}$$

$$\boxed{S_t = \gamma \cdot \left(x_t - a_t\right) + (1-\gamma)\, S_{t-L}}$$

Con la función *HoltWinters()* de R, podemos realizar cualquiera de los cuatro métodos de alisado mencionados anteriormente. Para ello, bastará con indicar adecuadamente los argumentos que intervienen, como se indica a continuación.

1. Para el **método de alisado exponencial simple**, indicaremos que los parámetros de alisado *beta* y *gamma* no intervienen, así que el formato será:

   HoltWinters(datos, beta = FALSE, gamma = FALSE)

2. Para el **método de Holt**, indicaremos que el parámetro de alisado *gamma* no interviene, así que el formato será:

   HoltWinters(datos, gamma = FALSE)

3. Para el **método de Holt-Winters multiplicativo**, se usarán los tres parámetros de alisado y sólo tendremos que especificar cómo se combinan tendencia y componente estacional, así que el formato será:

   HoltWinters(datos, seasonal = "multiplicative")

4. Para el **método de Holt-Winters aditivo**, se usarán los tres parámetros de alisado y sólo tendremos que especificar cómo se combinan tendencia y componente estacional, así que el formato será:

   HoltWinters(datos, seasonal = "additive")

IMPORTANTE: La función *HoltWinters()* de R se aplica sobre objetos de la clase *ts* (time series). Sólo en los casos de datos sin componente estacional, se podrán usar objetos de tipo numérico en lugar de ts.

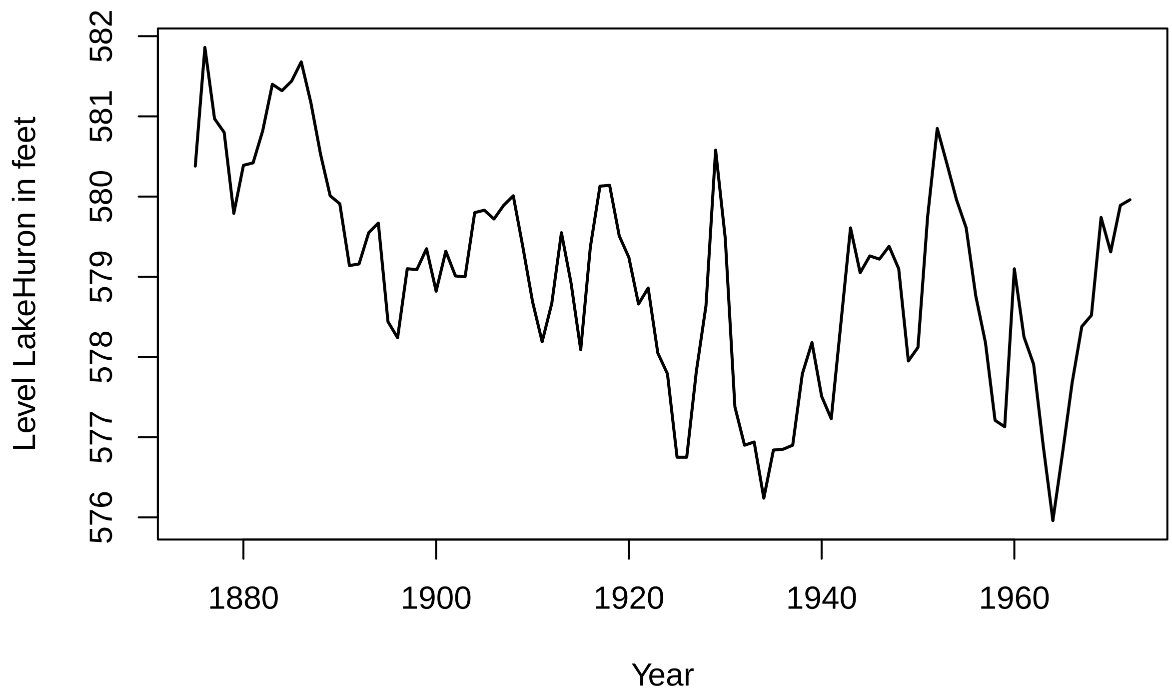
En la siguiente sección, mostramos un ejemplo concreto para cada método de alisado.

2. Ejemplos de los métodos de alisado exponencial con R

2.1. Alisado exponencial simple

Consideremos la serie de datos "LakeHuron" de R, que contiene las medidas anuales del nivel del lago Huron desde 1875 hasta 1972.

```
ts.plot(LakeHuron,
        gpars = list(xlab = "Year",
                     ylab = "Level LakeHuron in feet",
                     lwd = 1.5))
```



Esta serie no presenta estacionalidad y la tendencia se puede considerar constante localmente, así que aplicaremos el método de alisado exponencial simple.

```
class(LakeHuron)  #comprobamos que si LakeHuron es de clase ts, aunque en este caso no sería imprescind
```

```
## [1] "ts"
```

```
ejemplo_aes <- HoltWinters(LakeHuron, beta = FALSE, gamma = FALSE)
# summary(ejemplo_aes)
ejemplo_aes
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = LakeHuron, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.9999339
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##     [,1]
## a 579.96
```

En la salida anterior, observamos que el parámetro alpha óptimo, es decir, el que minimiza la suma de cuadrados de los errores vale casi uno: $alpha = 0.9999339$. Esto significa que, para estimar el nivel de la serie en cada instante, casi todo el peso recae en la última observación, careciendo de relevancia las observaciones anteriores. Según se indica, el valor estimado del nivel en el último instante es $a = 579.96$, que se corresponderá con las predicciones futuras que hagamos de la serie.

Veamos el valor de la suma de cuadrados residual para el alpha óptimo con:

```
ejemplo_aes$SSE
```

```
## [1] 53.86594
```

Podemos realizar predicciones futuras con *predict()*, por ejemplo, para los siguientes 5 años. Comprobamos que coinciden con el último nivel de la serie estimado $a = 579.96$:

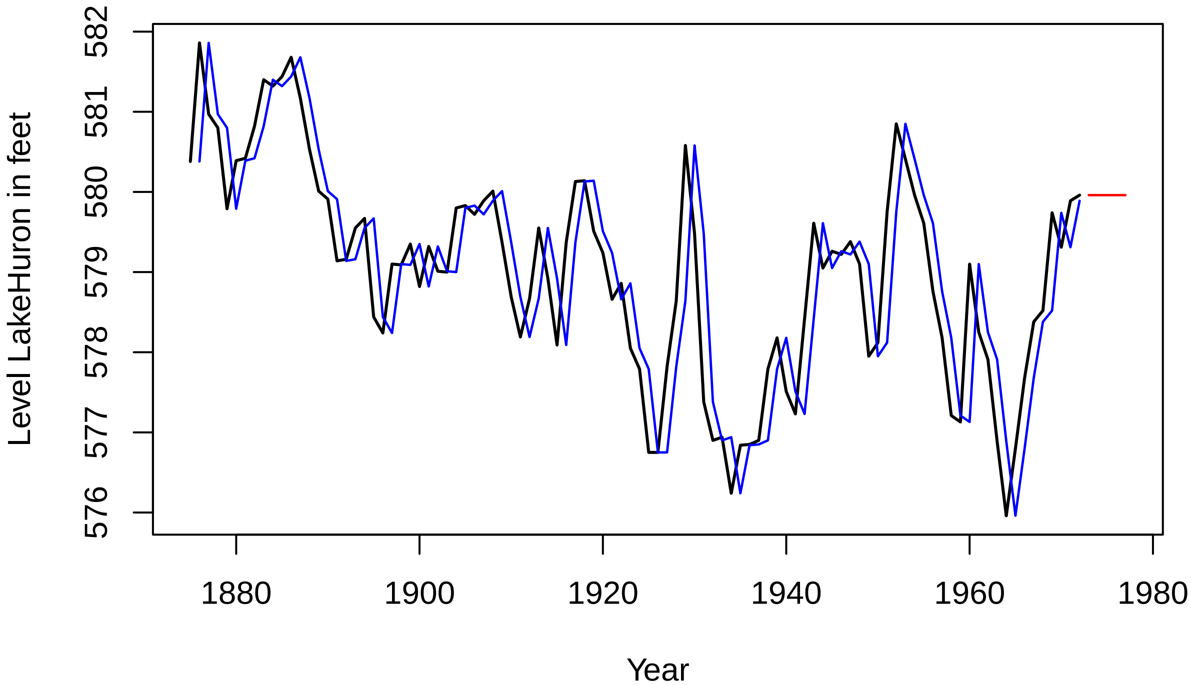
```
predict(ejemplo_aes, n.ahead = 5)
```

```
## Time Series:
## Start = 1973
## End = 1977
## Frequency = 1
##         fit
## [1,] 579.96
## [2,] 579.96
## [3,] 579.96
## [4,] 579.96
## [5,] 579.96
```

Representemos los datos reales (negro), junto valores ajustados (azul) y predicciones para los próximos 5 años (rojo):

```
ts.plot(LakeHuron, xlim = c(1875, 1977),
        gpars = list(xlab = "Year",
                     ylab = "Level LakeHuron in feet",
                     lwd = 1.5))

lines(ejemplo_aes$fitted[ , 1], col = "blue", lwd = 1.2)
lines(predict(ejemplo_aes, n.ahead = 5), col = "red", lwd = 1.2)
```

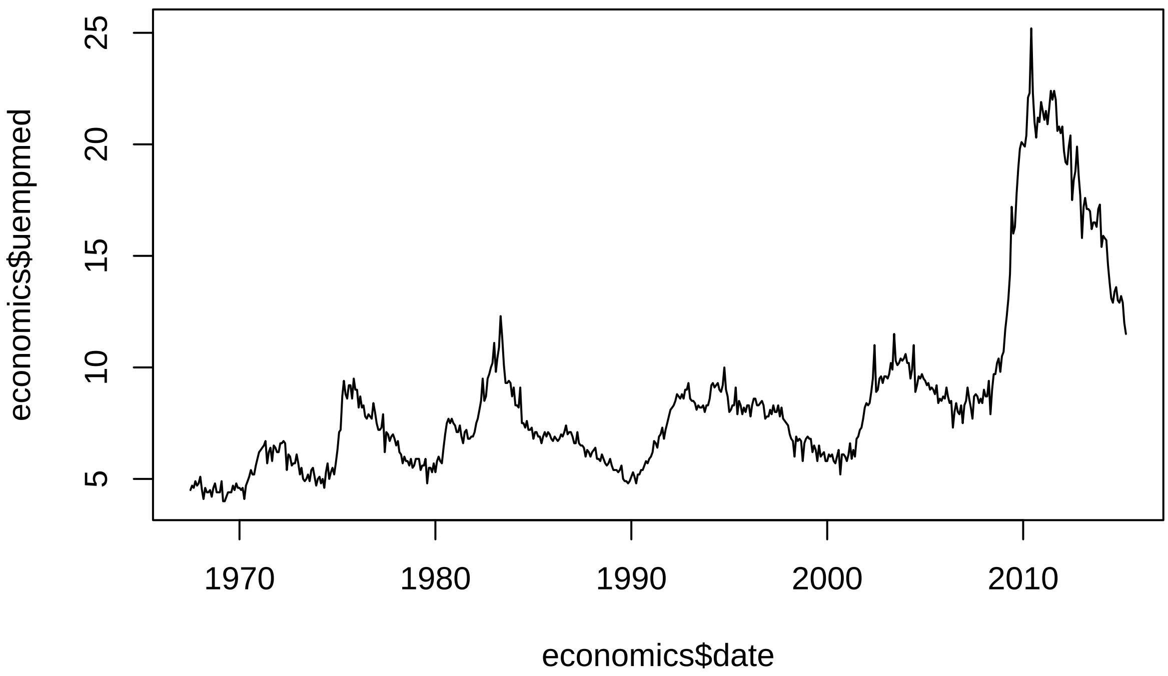


En este caso, los valores ajustados y las predicciones obtenidas con el método de alisado exponencial simple serían casi los mismos que los obtenidos con el método naive consistente en predecir el próximo instante con el último valor observado de la serie. Este ejemplo permite ilustrar además que los métodos de alisado exponencial tienen sentido para predicciones a muy corto plazo, siendo conveniente volver a ajustar el modelo conforme se obtienen nuevas observaciones.

2.2. Método de Holt

Consideraremos en esta ocasión un conjunto de datos disponible en el paquete *ggplot2*. Se trata del dataset *economics*, y en particular, nos quedaremos con la variable *uempmed* que representa la mediana de tiempo en desempleo (https://fred.stlouisfed.org/series/UEMPMED). Se trata de datos mensuales desde julio de 1967 hasta abril de 2015.

```
library(ggplot2)
# ts.plot(economics$uempmed)
plot(economics$date,economics$uempmed, type = "l")
```



En este caso, no se observa una componente estacional anual, es decir, no vemos un patrón periódico que se repita cada año (recordemos que se trata de datos mensuales). Por tanto, podemos aplicar el método de Holt, pues la serie no presenta estacionalidad y la tendencia de la serie se considera lineal (localmente). Indicar que, a pesar de que estos datos no son de clase *ts*, el método funciona al no haber componente estacional (gamma = FALSE).

```
class(economics$uempmed) #comprobamos que si los datos son de clase ts, aunque en este caso no sería im
```

```
## [1] "numeric"
```

```
ejemplo_Holt <- HoltWinters(economics$uempmed, gamma = FALSE)
ejemplo_Holt
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = economics$uempmed, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.7050079
##  beta : 0.0925097
##  gamma: FALSE
##
## Coefficients:
##           [,1]
## a 11.6412203
## b -0.2600279
```

En la salida anterior, observamos que los parámetros de alisado óptimos son: $alpha = 0.7050079$ y $beta = 0.0925097$. Según se indica, la estimación del último nivel de la serie es $a = 11.6412203$, mientras que para la pendiente de la tendencia es $b = -0.2600279$. Por tanto, las predicciones futuras de la serie vendrán dadas por la expresión: $\hat{x}_{T+t} = 11.6412203 - 0.2600279\ t$, donde T es el número de observaciones de la serie y t es el horizonte de predicción.

Veamos el valor de la suma de cuadrados residual para los parámetros de alisado óptimos:

```
ejemplo_Holt$SSE
```

```
## [1] 174.6968
```

Podemos realizar predicciones futuras con *predict()*, por ejemplo, para los siguientes 6 meses. Comprobamos que coinciden con sustituir los horizontes $t = 1, 2, ..., 6$ en la expresión: $\hat{x}_{T+t} = 11.6412203 - 0.2600279\ t$

```
predict(ejemplo_Holt, n.ahead = 6)
```

```
## Time Series:
## Start = 575
## End = 580
## Frequency = 1
##           fit
## [1,] 11.38119
## [2,] 11.12116
## [3,] 10.86114
## [4,] 10.60111
## [5,] 10.34108
## [6,] 10.08105
```

```
print("predicciones manuales:")
```

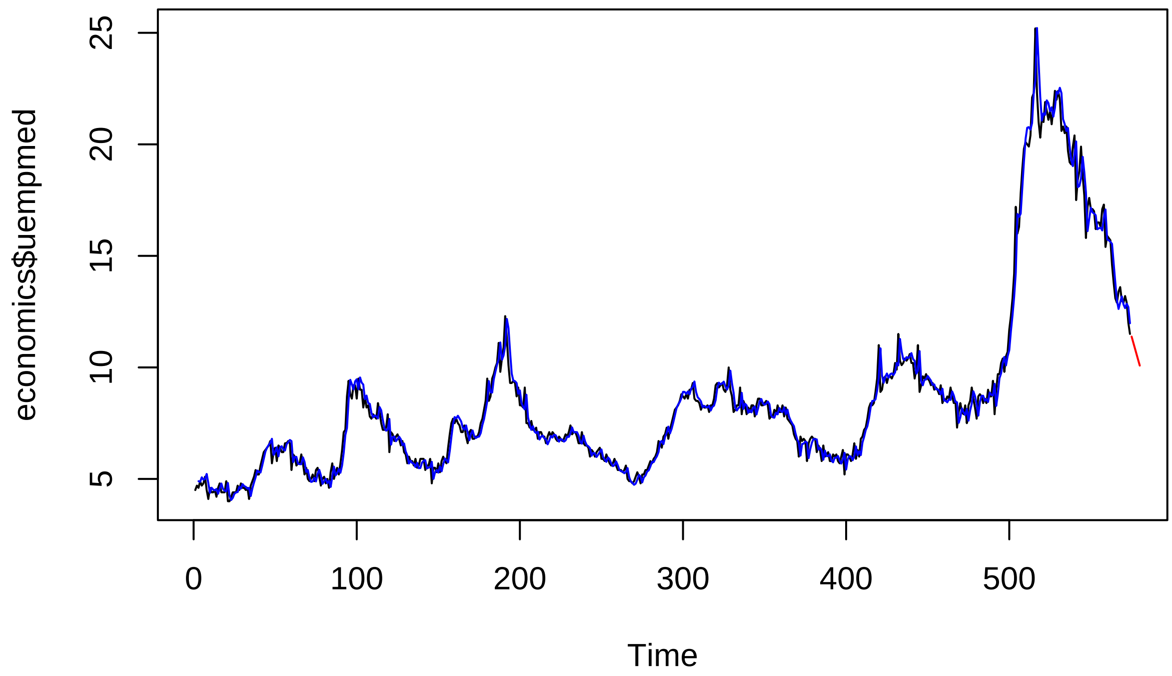
```
## [1] "predicciones manuales:"
```

```
horizonte <- 1:6
11.6412203-0.2600279*horizonte
```

```
## [1] 11.38119 11.12116 10.86114 10.60111 10.34108 10.08105
```

En el siguiente gráfico se muestra la serie original (negro), la serie ajustada (azul) usando el método de Holt y las predicciones (rojo) para los próximos 6 meses.

```
ts.plot(economics$uempmed)
lines(ejemplo_Holt$fitted[ , 1], col = "blue")
lines(predict(ejemplo_Holt, n.ahead = 6), col = "red")
```

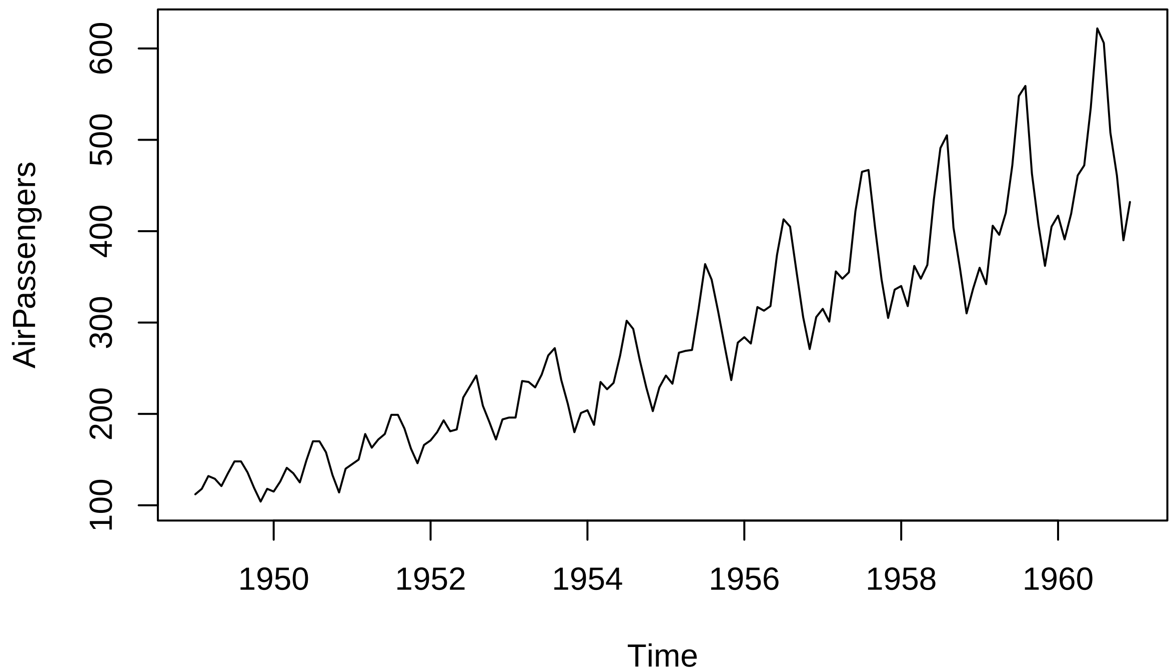


A medida que disponemos de nuevas observaciones, se deberían calcular los valores de las series alisadas para los nuevos instantes y por tanto se revisan las predicciones de manera automática.

2.3. Método de Holt-Winters multiplicativo

Vamos a considerar los datos de la serie temporal "AirPassengers" incluida en R. Contiene el número de pasajeros mensuales en aerolíneas internacionales, desde enero de 1949 hasta diciembre de 1960. Esta serie temporal ya se analizó en una práctica anterior con los métodos de descomposición estacional.

```
ts.plot(AirPassengers)
```



En este caso, sí se observa una clara componente estacional anual. Además, con esquema multiplicativo, pues aumenta la dispersión anual al aumentar el nivel de la serie. Por tanto, aplicaremos el método de Holt-Winters multiplicativo. Indicar que, en este caso, sí es necesario que los datos sean de clase *ts* (time series).

```
class(AirPassengers) #comprobamos que sí es de clase ts
```

```
## [1] "ts"
```

```
ejemplo_HW_mult <- HoltWinters(AirPassengers, seasonal = "multiplicative")
ejemplo_HW_mult
```

```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
##
## Call:
## HoltWinters(x = AirPassengers, seasonal = "multiplicative")
##
## Smoothing parameters:
##  alpha: 0.2755925
##  beta : 0.03269295
##  gamma: 0.8707292
##
## Coefficients:
##            [,1]
## a   469.3232206
## b     3.0215391
## s1    0.9464611
## s2    0.8829239
```

```
## s3    0.9717369
## s4    1.0304825
## s5    1.0476884
## s6    1.1805272
## s7    1.3590778
## s8    1.3331706
## s9    1.1083381
## s10   0.9868813
## s11   0.8361333
## s12   0.9209877
```

En la salida anterior, observamos que los parámetros de alisado óptimos son: $alpha = 0.2755925$, $beta = 0.03269295$ y $gamma = 0.8707292$. El valor de gamma próximo a 1 nos indica que las observaciones más recientes tienen mucho más peso que las pasadas a la hora de estimar la componente estacional (la forma de la estacionalidad no es muy estable en el tiempo). Según se indica, la estimación del último nivel de la serie es $a = 469.3232206$, la última pendiente de la tendencia es $b = 3.0215391$ y los últimos valores de la componente estacional son $s1 = 0.9464611$ (enero), $\ldots$, $s12 = 0.9209877$ (diciembre).

Por tanto, las predicciones futuras de la serie vendrán dadas por la expresión: $\hat{x}_{T+t} = (469.3232206 + 3.0215391\ t) \times S_{T+t-12}$, donde T es el número de observaciones de la serie, t es el horizonte de predicción y S_{T+t-12} se corresponde con uno de los 12 valores estacionales según el mes. Por ejemplo, para la predicción del próximo enero se usará $S_{T+t-12} = s1 = 0.9464611$.

Veamos el valor de la suma de cuadrados residual para los parámetros de alisado óptimos:

```
ejemplo_HW_mult$SSE
```

```
## [1] 16570.78
```

Podemos realizar predicciones futuras con *predict()*, por ejemplo, para los siguientes 12 meses. Comprobamos que coinciden con sustituir los horizontes $t = 1, 2, ..., 12$ en la expresión que se indicó más arriba.

```
predict(ejemplo_HW_mult, n.ahead = 12)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1961 447.0559 419.7123 464.8671 496.0839 507.5326 575.4509 666.5923 657.9137
##           Sep      Oct      Nov      Dec
## 1961 550.3088 492.9853 420.2073 465.6345
```

```
print("predicciones manuales:")
```

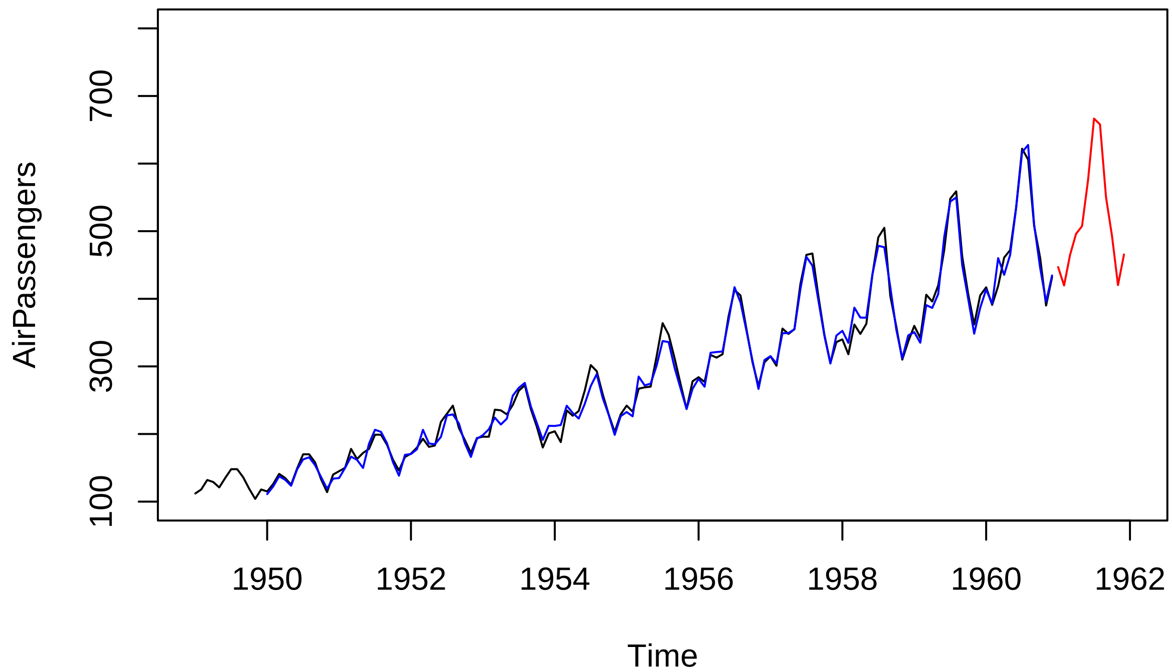
```
## [1] "predicciones manuales:"
```

```
horizonte <- 1:12
valores_estacionales <- ejemplo_HW_mult$coefficients[3:14]
(469.3232206+3.0215391*horizonte)*valores_estacionales
```

```
##       s1       s2       s3       s4       s5       s6       s7       s8
## 447.0559 419.7123 464.8671 496.0839 507.5326 575.4509 666.5923 657.9137
##       s9      s10      s11      s12
## 550.3088 492.9853 420.2073 465.6345
```

En el siguiente gráfico se muestra la serie original (negro), la serie ajustada (azul) y las predicciones (rojo) para los próximos 12 meses.

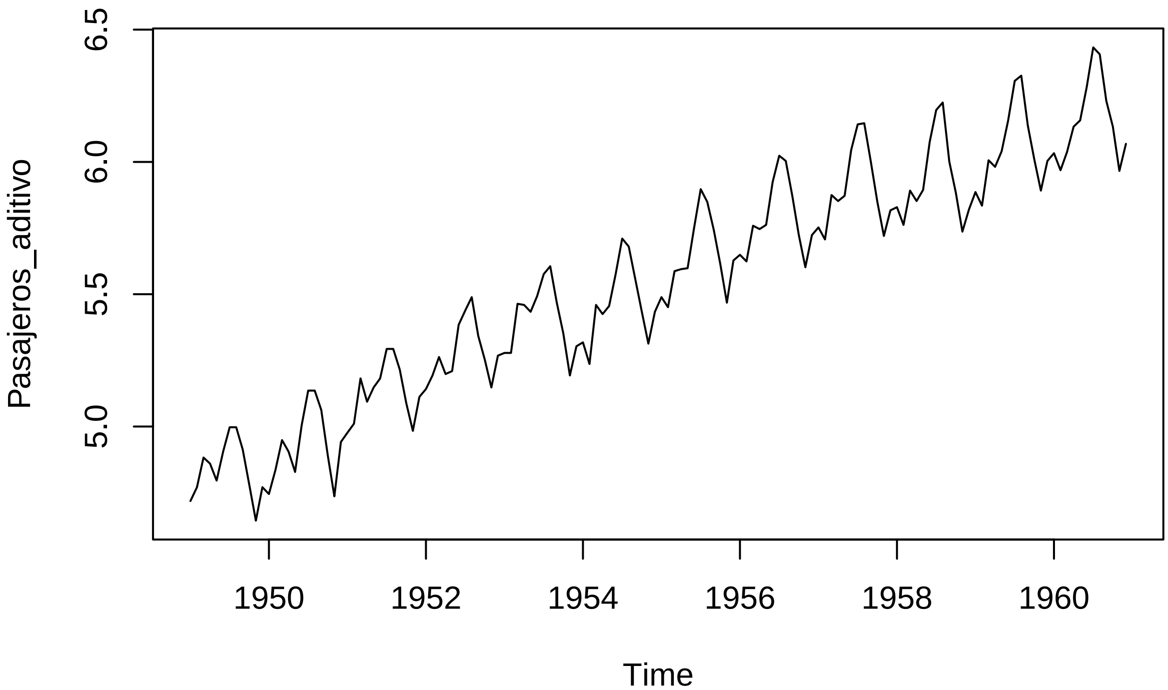
```
ts.plot(AirPassengers, xlim = c(1949, 1962), ylim = c(100, 800))
lines(ejemplo_HW_mult$fitted[ , 1], col = "blue")
lines(predict(ejemplo_HW_mult, n.ahead = 12), col = "red")
```



2.4. Método de Holt-Winters aditivo

Para ver un ejemplo de este tipo, usaremos de nuevo los datos de "AirPassengers", pero transformados tomando logaritmos neperianos. De esta forma el esquema con el que se combinan tendencia y estacionalidad será aditivo.

```
Pasajeros_aditivo <- log(AirPassengers)
ts.plot(Pasajeros_aditivo)
```



Repitiendo los pasos que hicimos en la sección anterior, tendremos ahora:

```
ejemplo_HW_aditivo <- HoltWinters(Pasajeros_aditivo, seasonal = "additive")
ejemplo_HW_aditivo
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = Pasajeros_aditivo, seasonal = "additive")
##
## Smoothing parameters:
##  alpha: 0.3266015
##  beta : 0.005744138
##  gamma: 0.8206654
##
## Coefficients:
##             [,1]
## a    6.172308435
## b    0.008981893
## s1  -0.073201087
## s2  -0.140973564
## s3  -0.036703294
## s4   0.014522733
## s5   0.032554237
## s6   0.154873570
## s7   0.294317062
## s8   0.276063997
```

```
## s9   0.088237657
## s10 -0.032657089
## s11 -0.198012716
## s12 -0.102863837
```

La medida de error es:

```
ejemplo_HW_aditivo$SSE
```

```
## [1] 0.2030765
```

```
residuos <- Pasajeros_aditivo - ejemplo_HW_aditivo$fitted[ , 1]
sum(residuos^2)
```

```
## [1] 0.2030765
```

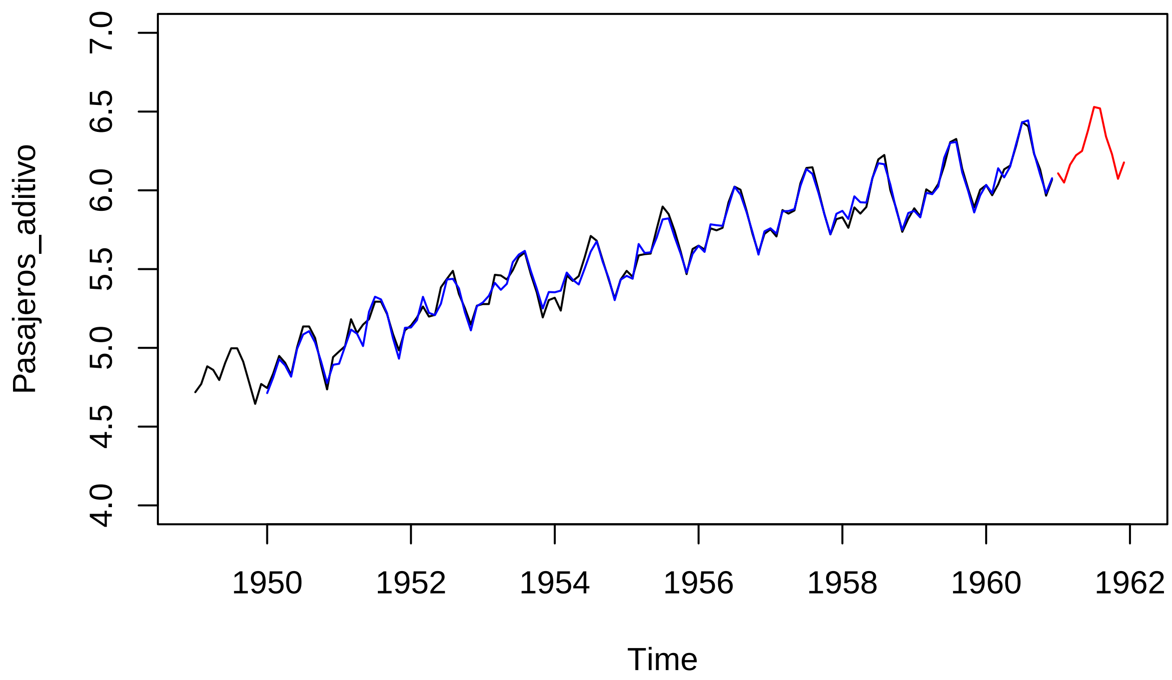
Y las predicciones del próximo año serán:

```
predict(ejemplo_HW_aditivo, n.ahead = 12)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1961 6.108089 6.049299 6.162551 6.222759 6.249772 6.381073 6.529499 6.520228
##           Sep      Oct      Nov      Dec
## 1961 6.341383 6.229470 6.073097 6.177227
```

En el siguiente gráfico se muestra la serie original (negro), la serie ajustada (azul) y las predicciones (rojo) para los próximos 12 meses.

```
ts.plot(Pasajeros_aditivo, xlim = c(1949, 1962), ylim = c(4, 7))
lines(ejemplo_HW_aditivo$fitted[ , 1], col = "blue")
lines(predict(ejemplo_HW_aditivo, n.ahead = 12), col = "red")
```



3. Modelos ETS (Error, Trend, Seasonal)

Recordemos que los modelos ETS representan los modelos estadísticos que subyacen bajo los métodos de alisado exponencial, dando lugar a las mismas estimaciones puntuales y permitiendo la obtención de intervalos de predicción.

En R, los modelos ETS se pueden aplicar usando la función *ets()* del paquete *forecast*.

Veamos a modo de ejemplo cómo abordar el análisis de la serie "Pasajeros_aditivo" que vimos en una sección anterior. Recordemos que, para dicha serie, se empleó el método de alisado Holt-Winters aditivo. Por tanto, en el contexto de modelos ETS, usaremos el modelo ETS(A,A,A). Es decir, errores aditivos, tendencia aditiva y estacionalidad aditiva.

```
library(forecast)
```

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

```
ejemplo_ETS_AAA <- ets(Pasajeros_aditivo, model = "AAA")
ejemplo_ETS_AAA
```

```
## ETS(A,A,A)
##
## Call:
## ets(y = Pasajeros_aditivo, model = "AAA")
##
##   Smoothing parameters:
```

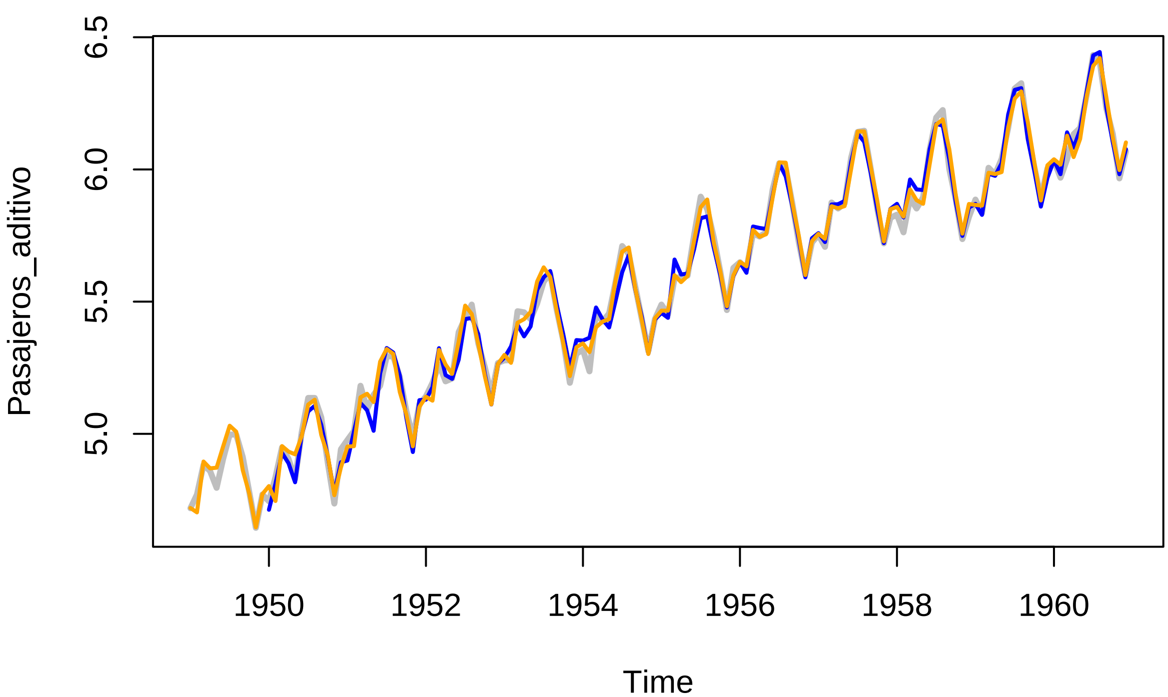
```
##     alpha = 0.6975
##     beta  = 0.0031
##     gamma = 1e-04
##
##   Initial states:
##     l = 4.7925
##     b = 0.0111
##     s = -0.1045 -0.2206 -0.0787 0.0562 0.2049 0.2149
##            0.1146 -0.0081 -0.0059 0.0225 -0.1113 -0.0841
##
##   sigma:  0.0383
##
##       AIC      AICc       BIC
## -207.1694 -202.3123 -156.6826
```

```
# summary(ejemplo_ETS_AAA)
```

Obsérvese que los parámetros de alisado óptimos que resultan del modelo ETS no coinciden con los obtenidos en el método Holt-Winters aditivo, puesto que ambos enfoques usan criterios diferentes para seleccionar los parámetros óptimos (véase https://robjhyndman.com/hyndsight/estimation2/index.html).

Podemos comparar los valores ajustados obtenidos con cada enfoque. Vemos que, a pesar de no coincidir, tienen comportamientos muy similares.

```
ts.plot(Pasajeros_aditivo, col = "grey", lwd = 3)
lines(ejemplo_HW_aditivo$fitted[ , 1], col = "blue", lwd = 2)
lines(ejemplo_ETS_AAA$fitted, col = "orange", lwd = 2)
```



Y podemos comparar las medidas de error (SSE) en ambos casos:

```
ejemplo_HW_aditivo$SSE
```

```
## [1] 0.2030765
```

```
sum(ejemplo_ETS_AAA$residuals^2)
```

```
## [1] 0.1873479
```

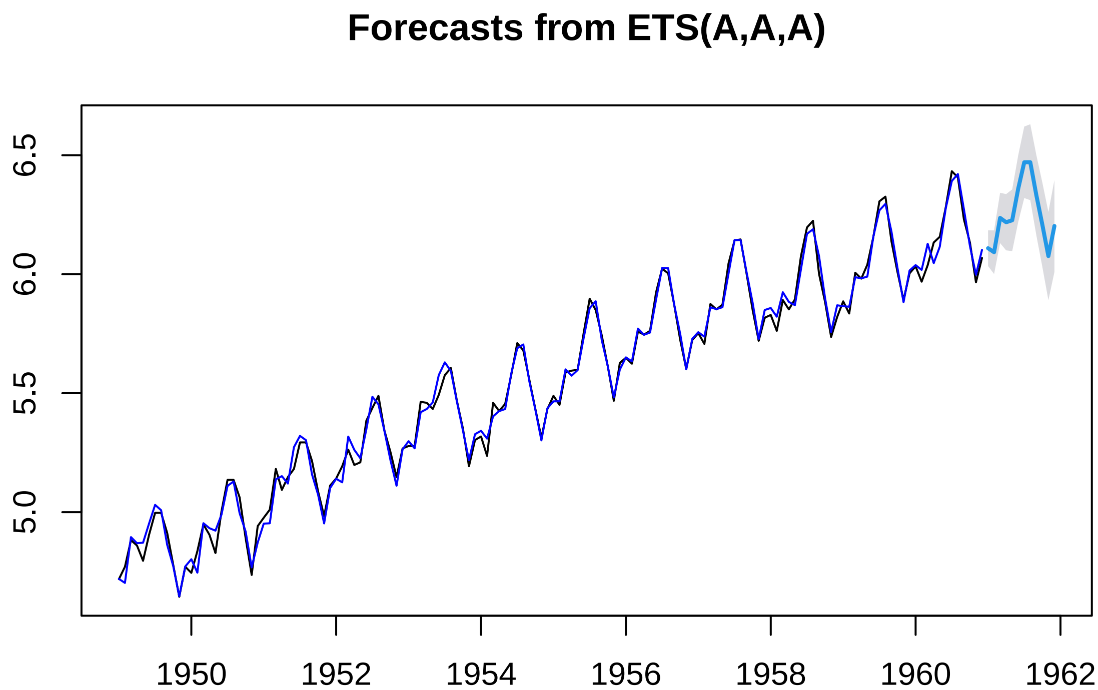
Una ventaja de los modelos ETS frente a los métodos de alisado, es que permiten obtener intervalos de predicción además de las predicciones puntuales. Por ejemplo, realizamos la predicción para los próximos 12 meses con el modelo ETS.

```
forecast(ejemplo_ETS_AAA, h = 12, level = 0.95)
```

```
##          Point Forecast    Lo 95    Hi 95
## Jan 1961       6.109335 6.034351 6.184319
## Feb 1961       6.092542 6.000989 6.184094
## Mar 1961       6.236626 6.130960 6.342292
## Apr 1961       6.218531 6.100323 6.336738
## May 1961       6.226734 6.097100 6.356369
## Jun 1961       6.359734 6.219514 6.499954
## Jul 1961       6.470379 6.320237 6.620521
## Aug 1961       6.470714 6.311189 6.630240
## Sep 1961       6.332402 6.163942 6.500863
## Oct 1961       6.207747 6.030732 6.384762
## Nov 1961       6.076240 5.890997 6.261482
## Dec 1961       6.202710 6.009525 6.395894
```

Representamos ahora la serie original (negro) con los valores ajustados (azul oscuro) y los intervalos de predicción para los meses del próximo año (zona sombreada en gris):

```
plot(forecast(ejemplo_ETS_AAA, h = 12, level = 0.95))
lines(ejemplo_ETS_AAA$fitted, col = "blue")
```



Otra gran ventaja de los modelos ETS es que se puede realizar la selección del modelo usando criterios de información (AIC, AIC corregido y BIC). Veamos para la serie "Pasajeros_aditivo" la búsqueda automática del modelo:

```
busqueda_ETS <- ets(Pasajeros_aditivo)
busqueda_ETS
```

```
## ETS(M,A,M)
##
## Call:
## ets(y = Pasajeros_aditivo)
##
##   Smoothing parameters:
##     alpha = 0.5871
##     beta  = 1e-04
##     gamma = 1e-04
##
##   Initial states:
##     l = 4.8055
##     b = 0.0091
##     s = 0.9814 0.96 0.9866 1.0116 1.0376 1.0379
##            1.02 0.9985 0.9986 1.004 0.9804 0.9833
##
##   sigma:  0.0069
##
##        AIC       AICc        BIC
## -208.3403 -203.4832 -157.8535
```

En este caso, el modelo que proporciona menor valor del AIC corregido es el ETS(M,A,M), que no coincide con el que ajustamos manualmente. Se puede comprobar que errores, valores ajustados y predicciones son muy similares en ambos casos:

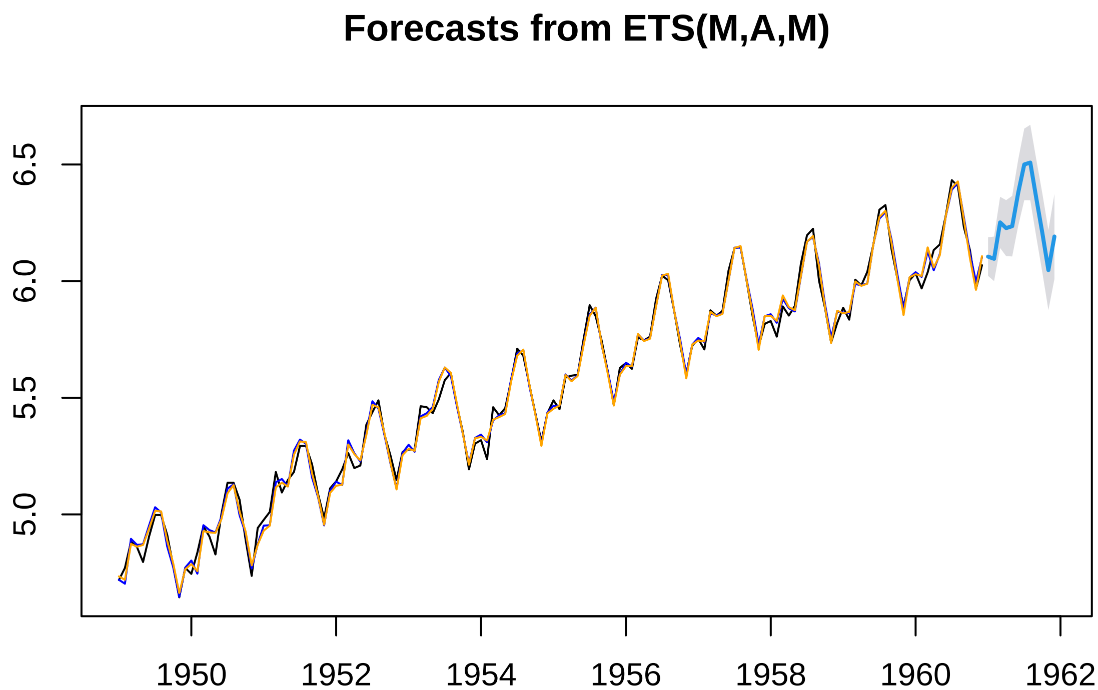
```
sum((Pasajeros_aditivo - busqueda_ETS$fitted)^2)
```

```
## [1] 0.178084
```

```
sum(ejemplo_ETS_AAA$residuals^2)
```

```
## [1] 0.1873479
```

```
plot(forecast(busqueda_ETS, h = 12, level = 0.95))
lines(ejemplo_ETS_AAA$fitted, col = "blue")
lines(busqueda_ETS$fitted, col = "orange")
```



Como ejercicio, se deja al lector el ajuste con modelos ETS de cada una de las series analizadas en esta práctica.