CONTENIDO

1. Técnicas de Suavizado **2**
1.1. SMA 2
1.2. EWMA 4
1.2.1. Suavizado exponencial simple 5
1.2.2. Suavizado exponencial doble 5
1.2.3. Suavizado exponencial triple 6
1.3. Holt-Winters Methods 6
1.3.1. Pronostico a través del suavizado en R 8

2. Pronóstico **12**
2.1. Modelo Auto-Regresivo 12

1. Técnicas de Suavizado

Una media móvil nos muestra el valor medio de una medida en un número de sesiones determinado. Una media móvil de 5 días mostrará el promedio de los datos de los últimos 5 días, una media móvil de 20 días muestra la media de los últimos 20 días, y así sucesivamente. Cuando conecta las medias de cada día, crea una línea de media móvil. El valor de la media móvil depende de dos factores.

1. Los valores que se están promediando.

2. El horizonte temporal.

Las características móvil implica que la media se mueve siguiendo los datos: es decir, recoge el dato que se genera en la última sesión, y a su vez, descarta el dato más antiguo de laa serie temporal. Dentro de todos los indicadores existentes en el mundo del análisis técnico, podría decirse que las medias móviles bien empleadas son un excelente indicador de tendencias.
La media móvil es un indicador de tendencia que nunca particip al movimiento o tendencia de los datso; es decir, simplemente sigue a la curva de datos confirmando la tendencia que hay en vigor en cada momento. No nos adelanta cambios de tendencia, pero si los puede confirmar.

1.1. SMA

Simple Moving Average: Promedio Móvil Simple

$$SMA = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

donde x_i son los valores de laa variable, por tanto SMA es un promedio aritmético de n períodos atrás.

```
library(tseries)

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

setwd("C:\\Users\\81799\\Downloads\\Pronosticos_y_series_de_tiempo\\data")
delitos <- read.csv("Tabla_rp.csv") #Tabla de delitos.
delitos <- delitos[-c(1:24),] #Quitó las primeros 24 filas
delitos_ts <- ts(delitos$totales, start = 2012, frequency = 12 )
```

Recordemos que cuando hiciemos la auto-correlación, nos dió un valor $k = 12$, en donde este valor hacia referencia que se podía predecir el futuro X_t con el dato X_{t-k}

```
library(TTR)
delitos$delitos_month_SMA <- SMA(delitos_ts, n=12)
class(delitos$delitos_month_SMA)# Por default ya nos lo hace de clase TimeSeries

## [1] "ts"
```

Con lo anterior l que estamos haciendo es una nueva columna llamada en la cual, hacemos SMA, de la siguiente manera:

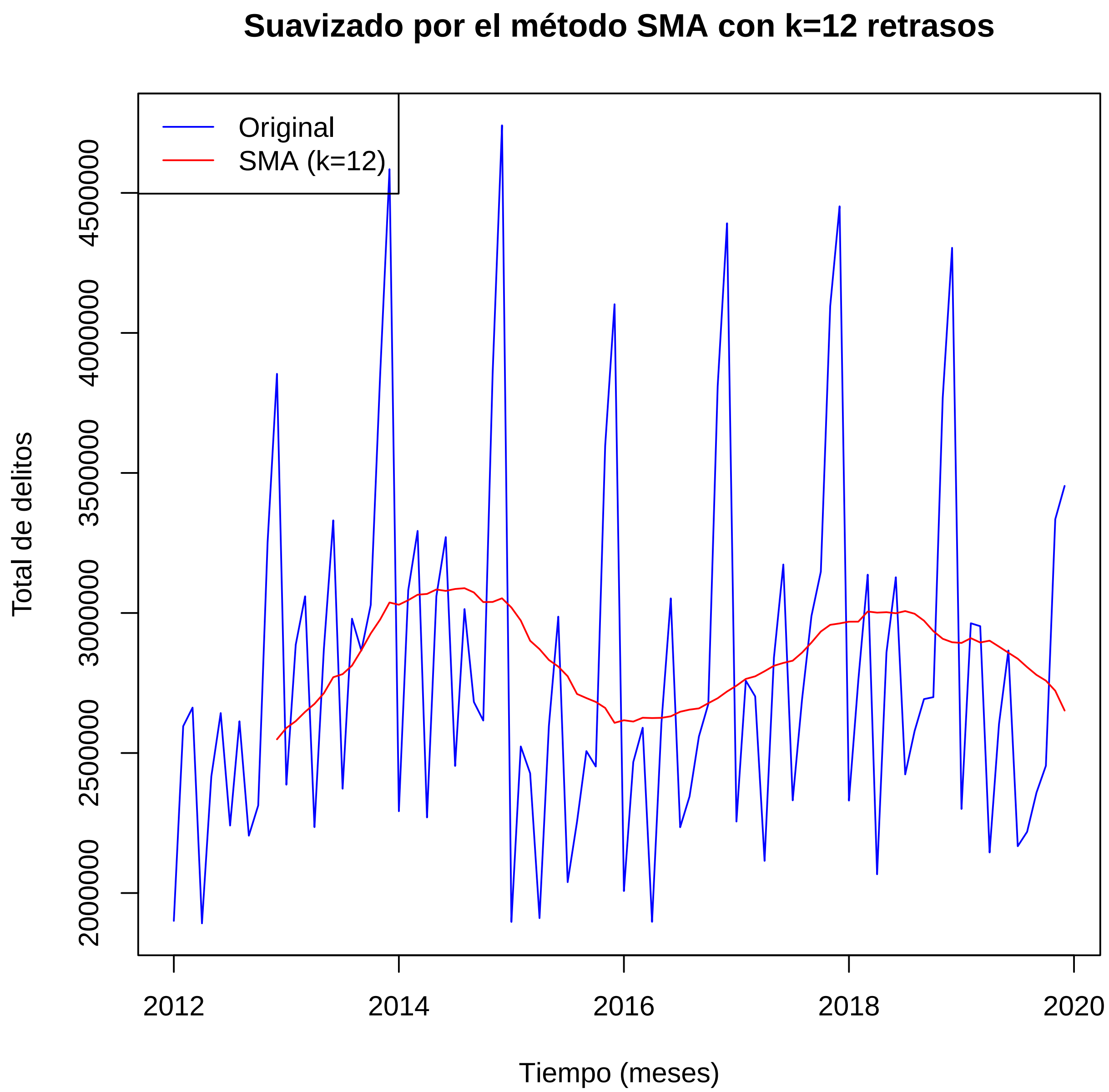
$$\sum_{i=1}^{k-1} y_i = 0 \qquad \textit{Ya que no tiene } k \textit{ datos anteriores}$$

$$y_t = \sum_{i=1}^{k} \frac{y_{t-i}}{k} \qquad t = k, k+1, \ldots$$

```
head(delitos[,c(1,3)], 15)

##    totales delitos_month_SMA
## 25 1900578                NA
## 26 2595733                NA
## 27 2662186                NA
## 28 1891696                NA
## 29 2416640                NA
## 30 2642589                NA
## 31 2240921                NA
## 32 2613253                NA
## 33 2204692                NA
## 34 2312757                NA
## 35 3252110                NA
## 36 3853923           2548923
## 37 2386909           2589451
## 38 2886517           2613683
## 39 3059545           2646796
```

```
delitos_suavizados <- cbind(delitos_ts,delitos$delitos_month_SMA) #Hago un nuevo DataFrame
plot(delitos_suavizados,plot.type = "single", col = c("blue", "red"),
     lwd=1:1, lty=1:1, ylab ="Total de delitos", xlab = "Tiempo (meses)",
     main = "Suavizado por el método SMA con k=12 retrasos")
legend(x="topleft",legend = c("Original", "SMA (k=12)"), col = c("blue", "red"), lty =1:1 )
```



1.2. EWMA

Exponentially Weighted Moving Average: Promedio Móvil Ponderado Exponencialmente

EWMA nos permitirá reducir el efecto de retasp de SMA y pondrá más peso en los valores que ocurrieron más recientemente. La cantidad de peso aplicado a los valores más recientes dependerá de los parámetros utilizados en EWMA, la fórmula con la que se calculan las estimaciones es:

$$y_t = \frac{\sum_{i=0}^{t} w_i x_{t-i}}{\sum_{i=0}^{t} w_i}$$

Donde x_t son los valores de la variable, w_i son los pesos (desde $i = 0$ hasta t), y la y_t es la estimación resultante.

Python

La pregunta es ¿ cómo se deben definir los pesos w_i?
Esto depende del argumento *adjust* dentro de la función *.ewm()*.
Cuando *adjust=True* (por defecto), los pesos se calculan con esta ecuación $w_i = (1-\alpha)^i$
Lo cual resulta en estas estimaciones:

$$y_t = \frac{x_t + (1-\alpha)x_{t-1} + (1-\alpha)^2 x_{t-2} + \cdots + (1-\alpha)^t x_0}{1 + (1-\alpha) + (1-\alpha)^2 + \cdots + (1-\alpha)^t}$$

Cuando *adjust=False* las estimaciones se calculan como:

$$\begin{aligned} y_0 &= x_0 \\ y_t &= (1-\alpha)y_{t-1} + ax_t \end{aligned}$$

que es equivalente a usar estos pesos:

$$w_i = \begin{cases} \alpha(1-\alpha)^i & si\ i < t \\ (1-\alpha)^i & si\ i = t \end{cases}$$

El parámetro de suavizado α tiene que ser un valor $0 < \alpha < 1$. Es posible pasar directamente el valor del parámetro, pero una mejor práctica es pensar en él como una función de estos tres posibles factores.

1. Span (duración).
2. Center of mass (centro de masas).
3. Half-life (vida media).

$$\alpha = \begin{cases} \frac{2}{s+1} & \textit{para una duración (span) } s \geq 1 \\ \frac{1}{1+c} & \textit{para un centro de masa} c \geq 0 \\ 1 - e^{\frac{\log 0.5}{h}} & \textit{para un parámetro half-life} h > 0 \end{cases}$$

- **Span** corresponde a lo que comúnmente se llama " promedio móvil de N-períodos´´.
- **Cennter of mass** tiene una interpretación más física y se puede pensar en términos de la duración: $c = \frac{s-1}{2}$.
- **Half-life** es el período de tiempo para que el peso exponencial se reduzca a la mitad.
- **Alpha** es el parámetro de suavizado directamente.

El ejemplo anterior hemos empleado una especie de *Suavizado exponencial simple* con un factor de suavizado α. desafortunadamente, esta técnica hace un mal trabajo de pronostico cuando hay una tendencia en los datos.

1.2.1. Suavizado exponencial simple

La función del suavizado exponencial de *statsmodels* realiza las mismas estimaciones que el método de promedios ponderados de pandas.

1.2.2. Suavizado exponencial doble

Cuando el suavizado exponencial simple emplea solo un factor de suavizado α(alpha), el suavizado exponencial doble agrega un segundo factor de suavizado β(beta) que aborda las tendencia en los datos. Al igual que el factor alpha, los valores para el factor beta están entre cero y uno ($0 < \beta < 1$). El beneficio aquí es que el modelo puede anticipar futuros aumentos o disminuciones donde el modelo de un sólo factor sólo tendría en cuenta los valores más recientes.
También podemos abordar diferentes tipos de cambio (crecimiento/decadencia) en la tendencia. Si una serie temporal muestra una tendencia inclinada en línea recta, se usaría un ajuste **aditivo.** Si la serie temporal muestra una tendencia exponencial (curva), se usaría un ajuste **multiplicativo.**

1.2.3. Suavizado exponencial triple

Suavizado exponencial triple o Wolt-Winters, añade soporte para la tendencia y la estacionalidad

1.3. Holt-Winters Methods

En la clase anterior vimos los **Promedios móviles ponderados exponencialmente** (EWMA) que es un *Suavizado exponencial simple* usando un solo factor de suavizado α (alpha). Pero no tuvo en cuenta otros factores que contribuyen como la tendencia y la estacionalidad.
En esta clase veremos el *Suavizado exponencial doble y triple* con los Métodos Holt-Winters.

En el **Suavizado exponencial doble** (también conocido como Método de Holt) presentamos un nuevo factor de suavizado β (beta) que aborda la tendencia:

$$\begin{aligned}
l_t &= (1-\alpha)l_{l-1} + \alpha x_t && \textit{nivel.}\\
b_t &= (1-\beta)b_{t-1} + \beta(l_t - l_{t-1}) && \textit{tendencia.}\\
y_t &= l_t + b_t && \textit{modelo estimado.}\\
\hat{y}_{t+h} &= l_t + hb_t && \textit{Modelo de prónosticos (h=\# períodos en el futuro).}
\end{aligned}$$

En el **Suavizado exponencial triple** (también conocido como Método de Holt) presentamos un nuevo factor de suavizado γ (gamma) que aborda la tendencia:

$$\begin{aligned}
l_t &= (1-\alpha)l_{l-1} + \alpha x_t && \textit{nivel.}\\
b_t &= (1-\beta)b_{t-1} + \beta(l_t - l_{t-1}) && \textit{Tendencia.}\\
c_t &= (1-\gamma)c_{t-L} + \gamma(x_t - l_{t-1} - b_{t-1}) && \textit{Estacionalidad}\\
y_t &= (l_t + b_t)c_t && \textit{Modelo estimado.}\\
\hat{y}_{t+m} &= (l_t + mb_t)c_{t-L+1+(m-1)modL} && \textit{Modelo de prónosticos (m= \# períodos en el futuro).}
\end{aligned}$$

Aquí **L** representa el número de divisiones por ciclo. En nuestro caso, mirando los datos mensuales que muestran un patrón repetitivo cada año, usaríamos $L = 12$.
EN general, los valores más altos para α, β y γ (valores más cercanos a 1), ponen más énfasis en los datos recientes.

R

En comparación con Python, R hace este procedimiento de una manera más sencilla con la función *ets*

En model mide "Errores, Tendencia, Estacionalidad", los parámetros a utilizar son:

- N: No hacer caso.
- A: Tipo Aditiva
- M: Tipo Multiplicativa.
- Z: Calcular en automático el tipo que se va a requerir

Cuando se hace un **exponencial simple** solamente considerábamos el error, es decir, no tomábamos en cuenta la tendencia ni la estacionalidad, para lograr eso en R pondremos en la parte model lo siguiente: model = "ZNN".

```r
library(tseries)
library(forecast)
library(TTR)
modelo_simple_aditivo <- ets(delitos_ts, model = "ANN") #Error, Tendencia y Estacionalidad
```

Un **Exponencial doble aditivo** sería de la siguiente manera: model = "AAN" ya que ahora estamos tomando la tendencia aditiva.

```
library(TTR)
modelo_doble_aditivo <- ets(delitos_ts, model = "AAN") #Error, Tendencia y Estacionalidad
```

Un **Exponencial Triple aditivo** sería de la siguiente manera: model = "AAA" ya que ahora estamos tomando la tendencia aditiva y estacionalidad aditiva

```
library(TTR)
modelo_triple_aditivo <- ets(delitos_ts, model = "AAA") #Error, Tendencia y Estacionalidad
```

Cuando se pone model = "ZZZ" nos dará R lo que más conviene a considerar

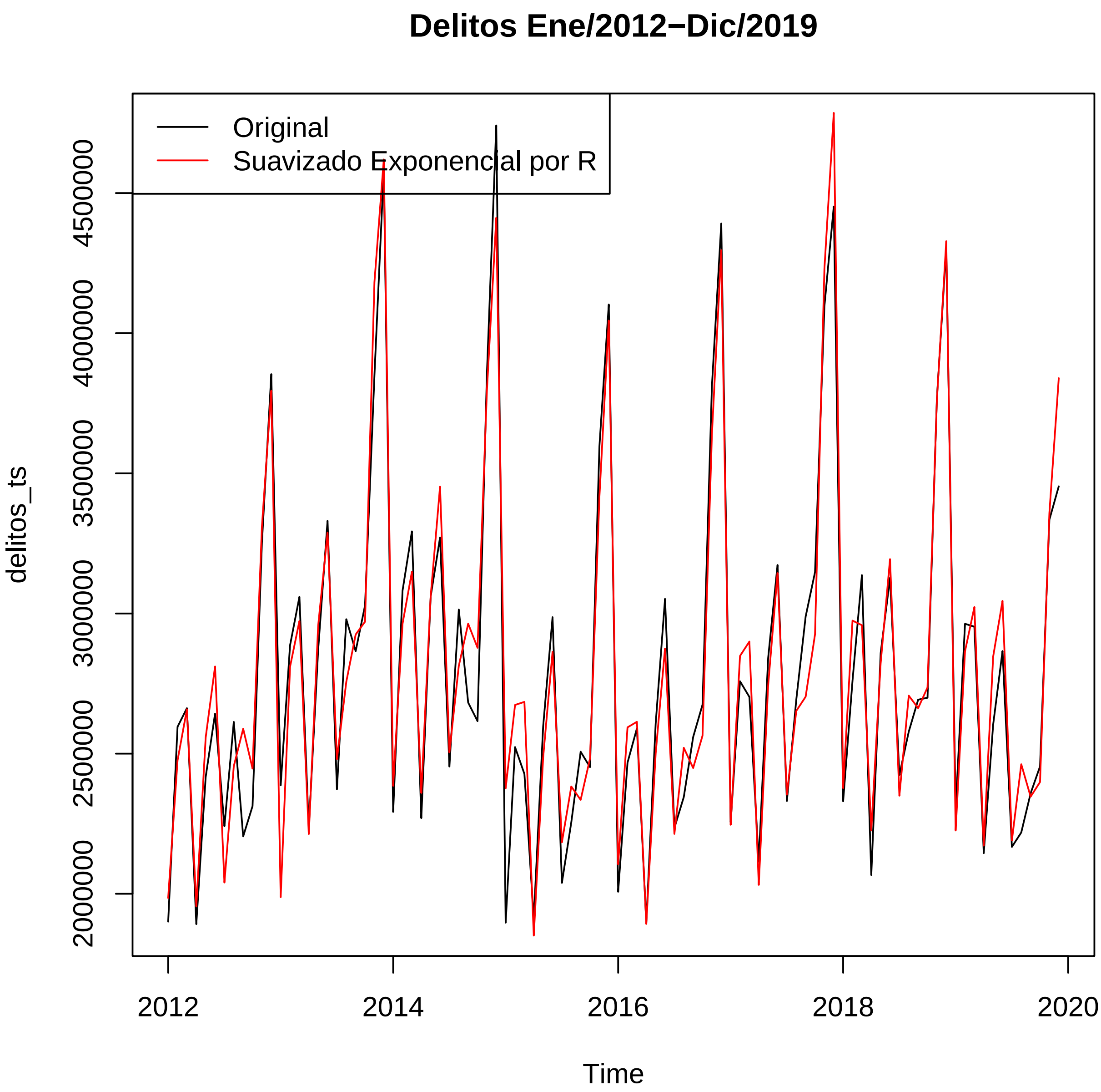
```
library(TTR)
modelo_ets <- ets(delitos_ts, model = "ZZZ") #Error, Tendencia y Estacionalidad
modelo_ets

## ETS(M,Ad,M)
##
## Call:
##  ets(y = delitos_ts, model = "ZZZ")
##
##   Smoothing parameters:
##     alpha = 0.5504
##     beta  = 0.008
##     gamma = 1e-04
##     phi   = 0.9768
##
##   Initial states:
##     l = 2526501.3669
##     b = 30018.4163
##     s = 1.5061 1.3108 0.947 0.9278 0.9193 0.8126
##            1.0926 0.9732 0.738 1.0149 0.981 0.7766
##
##   sigma:  0.0644
##
##      AIC     AICc      BIC
## 2776.788 2785.672 2822.947
```

Recordemos que ETS(Error,Tendencia,Estacionalidad), con lo anterior vemos que R da como resultado ETS(M,Ad,M) lo que significa que:

- Error Multiplicativo.
- Tendencia Aditivo.
- Estacionalidad Multiplicativo.

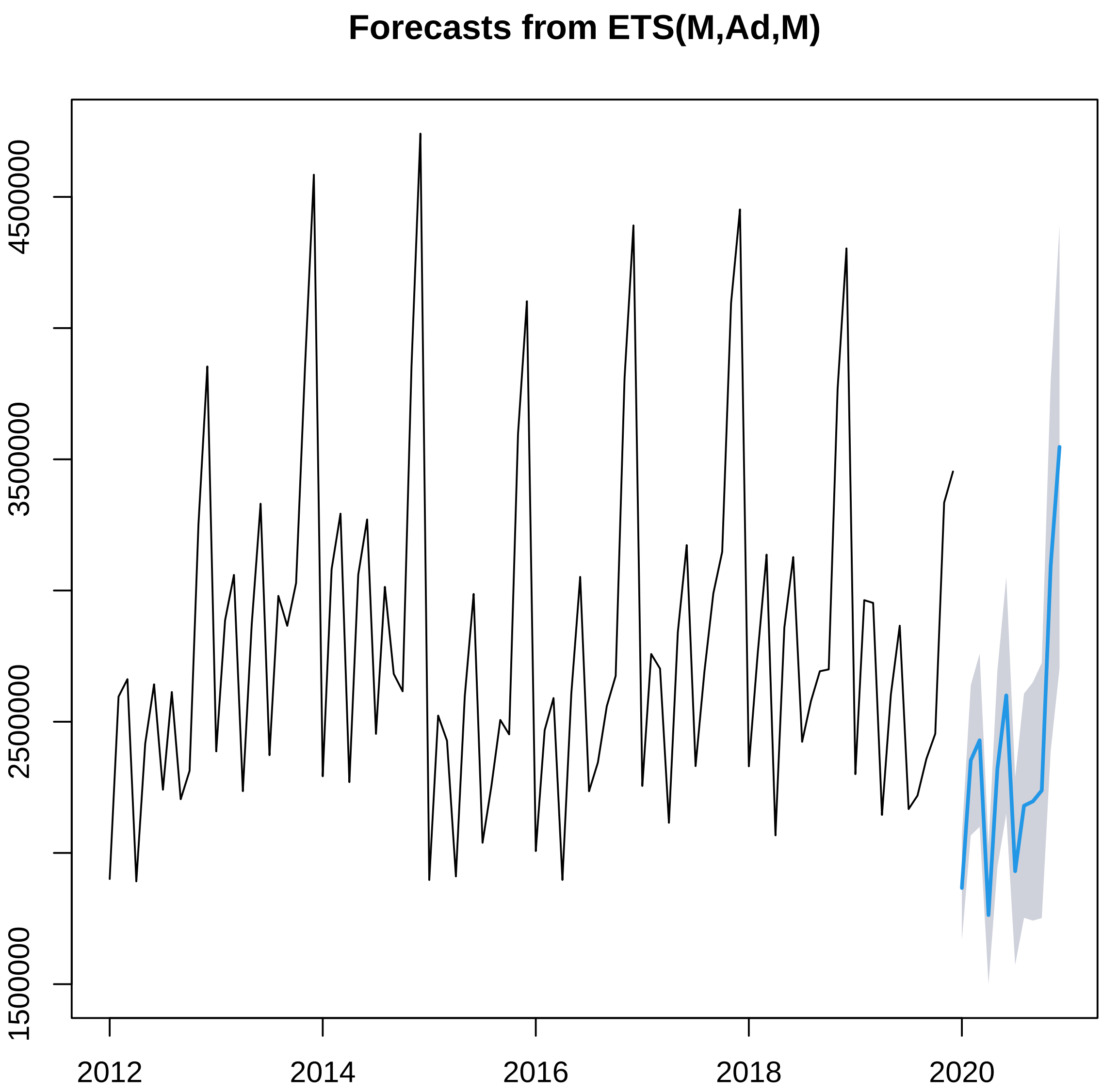
```
plot(delitos_ts, main = "Delitos Ene/2012-Dic/2019") #Datos originales
lines(modelo_ets$fitted, col = "red") #Suavizado Exponencial por R
legend(x="topleft",legend = c("Original", "Suavizado Exponencial por R"), col = c("black", "red"
```



1.3.1. Pronostico a través del suavizado en R

Con lo anterior, podemos hacer una predicción de 1 año a futuro de la siguiente manera:

```
library(forecast)
plot(forecast(modelo_ets, h=12,level = 90))
```



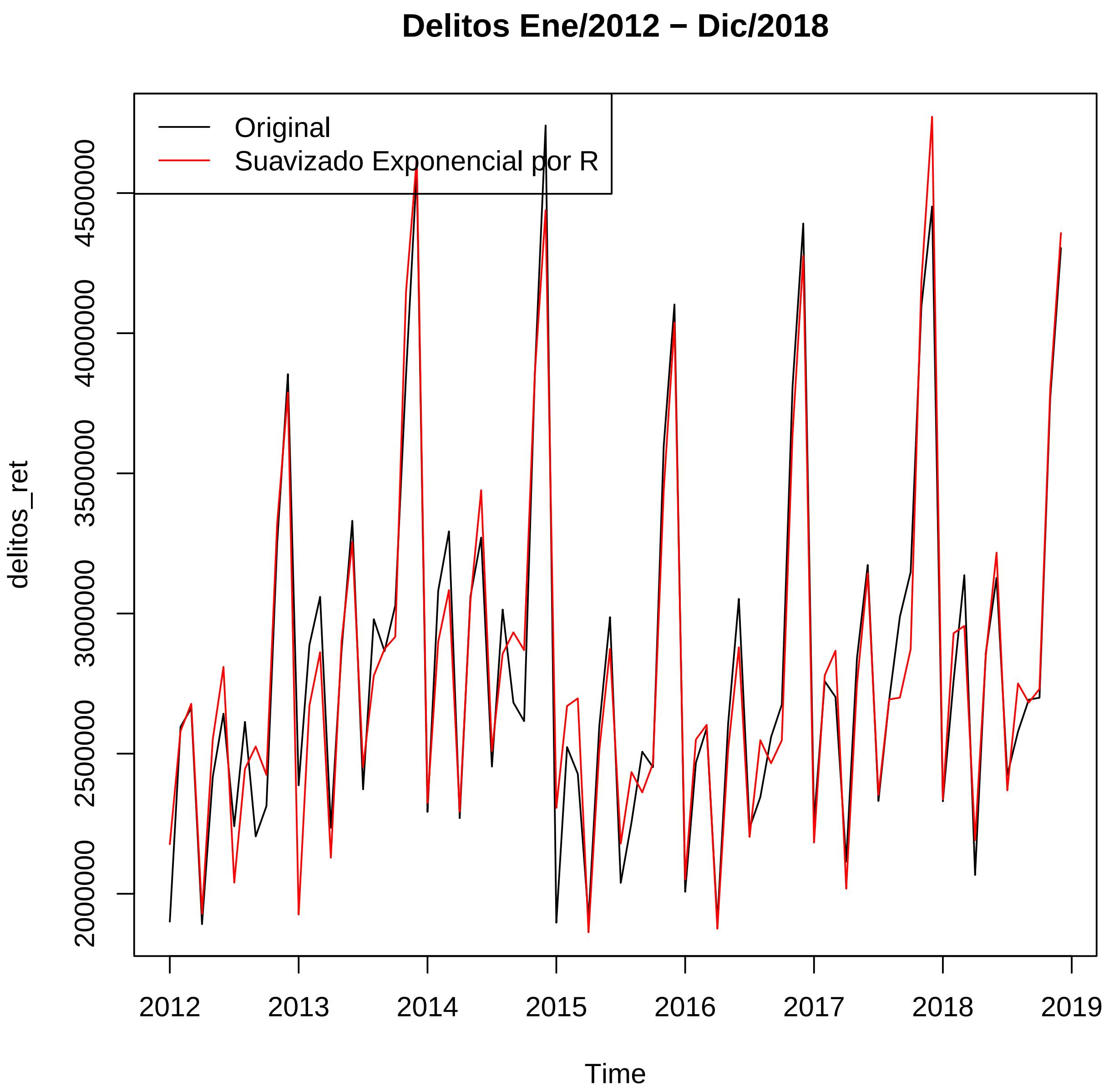
Lo anterior es una predicción con un nivel de confianza del 90 %

Predicción con los datos que tenemos

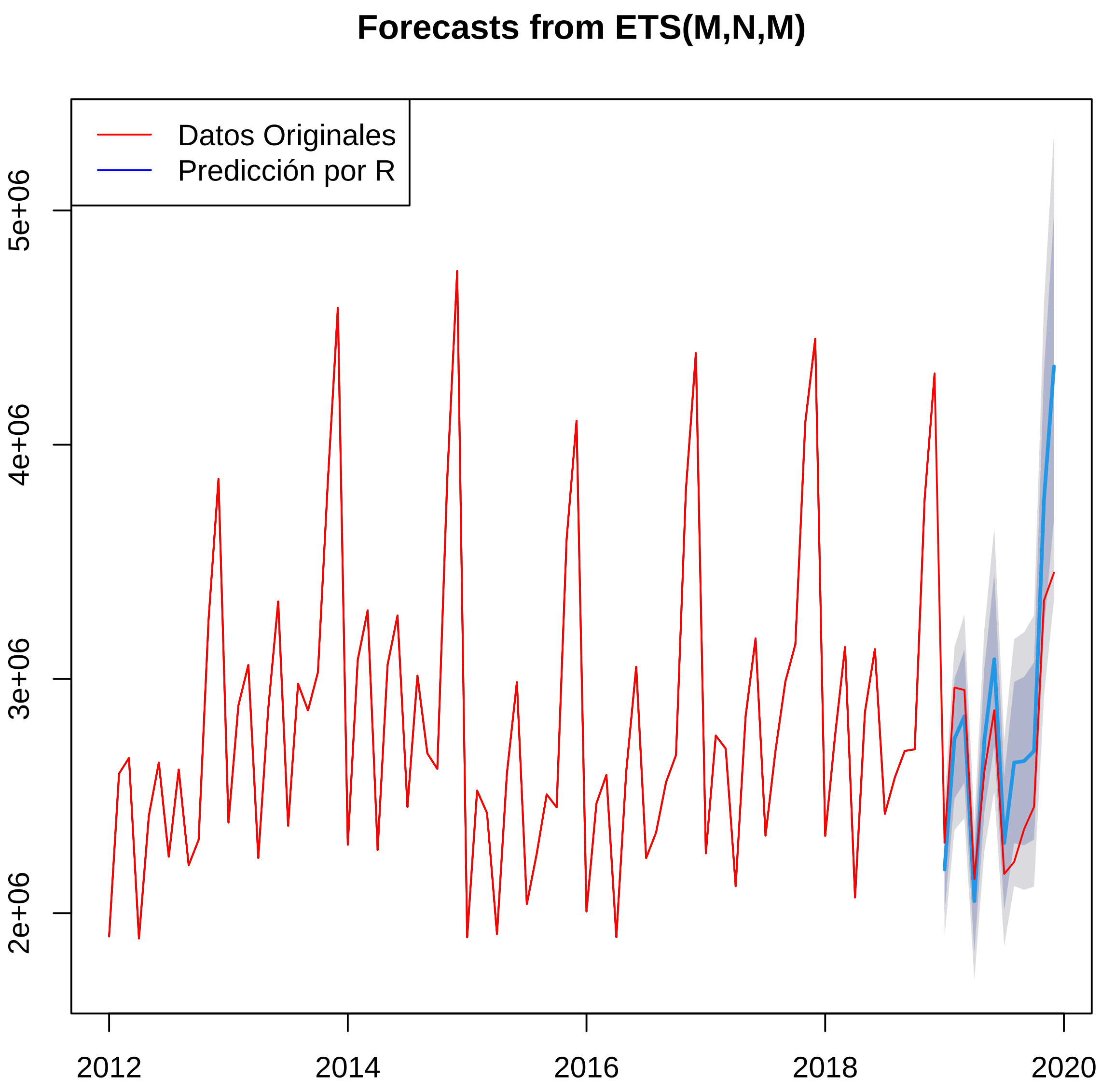
Para darnos una idea de que tan acertado esta nuestro pronostico, podemos eliminar el año 2019 de nuestros datos originales y hacer una predicción de ese año para ver que tan buena es la predicción.

```
library(tseries)
library(TTR)
delitos_ret <-ts(delitos$totales[c(1:84)], start = 2012, frequency = 12) #Quito el año 2019
modelo_ret <- ets(delitos_ret, model = "ZZZ") #Error, Tendencia y Estacionalidad
```

```
plot(delitos_ret, main = "Delitos Ene/2012 - Dic/2018")
lines(modelo_ret$fitted, col = "red") #Suavizado Exponencial por R
legend(x="topleft",legend = c("Original", "Suavizado Exponencial por R"), col = c("black", "red"
```




```
library(forecast)
plot(forecast(modelo_ret, h=12))
lines(delitos_ts, col = "red")
legend(x="topleft",legend = c("Datos Originales", "Predicción por R"), col = c("red", "blue"), l
```



De lo anterior, la predicción no queda exactamente (Linea azul) pero la linea de los datos originales (roja) nunca se sale de la franja gris (Intervalo de confianza)

```
forecast(modelo_ret, h=12) #Valores de la predicción

##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Jan 2019        2186688 2000534 2372842 1901990 2471386
## Feb 2019        2744570 2489477 2999664 2354439 3134702
## Mar 2019        2840769 2556247 3125291 2405630 3275908
## Apr 2019        2051322 1832059 2270585 1715988 2386656
## May 2019        2736055 2426262 3045849 2262267 3209844
## Jun 2019        3084154 2716422 3451885 2521757 3646550
## Jul 2019        2299155 2011865 2586446 1859782 2738529
## Aug 2019        2642509 2297843 2987174 2115388 3169629
## Sep 2019        2649390 2289892 3008887 2099585 3199194
## Oct 2019        2692758 2313730 3071786 2113085 3272431
## Nov 2019        3765003 3216617 4313389 2926319 4603687
## Dec 2019        4334036 3682225 4985847 3337177 5330895
```

2. Pronóstico

2.1. Modelo Auto-Regresivo

Los modelos auto-regresivos predicen el valor actual de nuestra serie temporal en función de los valores pasados. Un modelo auto-regresivo de orden simple, de orden uno, sólo consideraría el valor del período anterior, y el modelo se expresaría de la siguiente manera:

$$x_t = c + \Phi x_{t-1} + \varepsilon_t$$

Donde:

- x_t es el valor de interés, del período actual, de nuestra serie de tiempo.
- c es una constante.
- Φ es el coeficiente que debemos estimar.
- ε_t es el residuo del período actual.
- x_{t-1} es el valor de la serie en el período anterior.

, **Observación:** No funcionan bien si los datos NO son estacionarios.

Sabemos que la ACF (Función de auto-correlación) captura los efectos directos e indirectos del valor anterior sobre el valor presente. Como queremos un modelo eficiente solo queremos considerar aquellos retrasos que tengan un efecto directo y significativo sobre el período presente. Por lo tanto, debemos examinar la PACF (Auto-correlación parcial) antes de construir un modelo con demasiados coeficientes de retraso.
Dada la situación anterior, el modelo se puede hacer más complejo si así se requiere, aumentando el número de retrasos:

$$\begin{array}{ll}
AR(1) & x_t = c + \Phi_1 x_{t-1} + \varepsilon_t \\
AR(2) & x_t = c + \Phi_1 x_{t-1} + \Phi_2 x_{t-2} + \varepsilon_t \\
AR(3) & x_t = c + \Phi_1 x_{t-1} + \Phi_2 x_{t-2} + \Phi_3 x_{t-3} + \varepsilon_t \\
\vdots & \vdots
\end{array}$$