Análisis y Predicción de Series Temporales

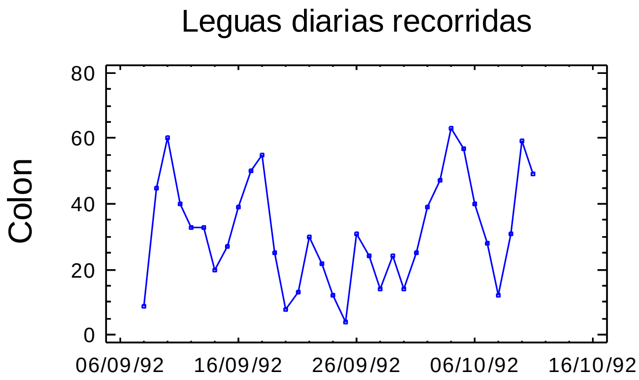
TEMA 1. ANÁLISIS DESCRIPTIVO DE UNA SERIE TEMPORAL

Autor: Juana María Alonso Revenga

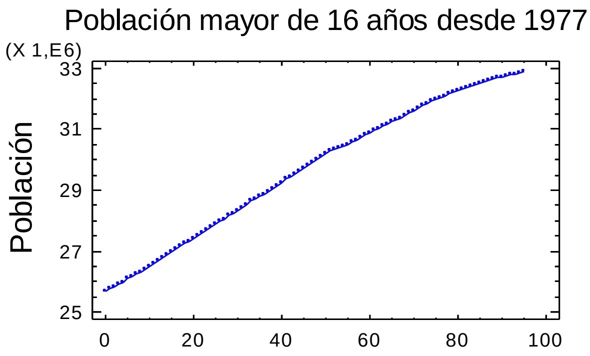
1.1 INTRODUCCIÓN

Una serie temporal es el resultado de observar los valores de una variable a lo largo del tiempo en intervalos regulares (cada día, cada mes, cada año,…)

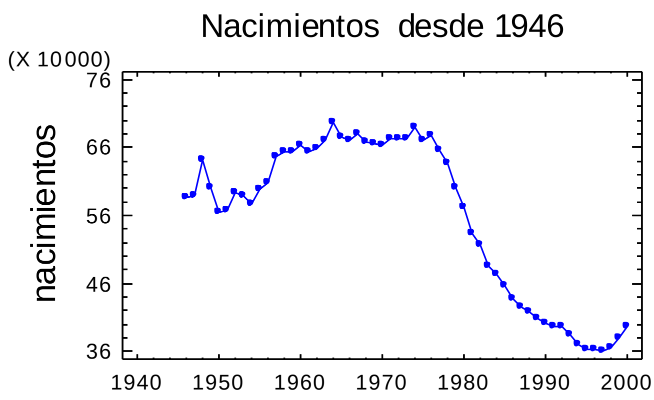
Diremos que una serie es **estacionaria** cuando **fluctúa alrededor de un nivel constante.** Por ejemplo si representamos las leguas recorridas diariamente por la flota de Colón en su primer viaje se observa que la serie es estable con valores que oscilan alrededor de un nivel fijo que en este caso son 30 leguas marinas.



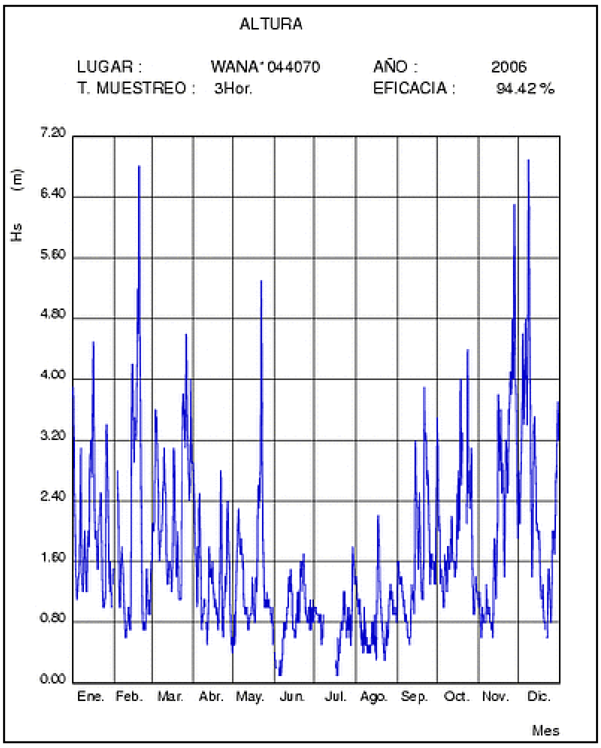
Algunas series no son estacionarias sino que presentan una **tendencia** clara que **podemos modelizar con alguna función matemática (lineal, cuadrática exp,..),** como la población mayor de 16 años en España desde el primer cuatrimestre de 1977 al cuarto trimestre del 2000.



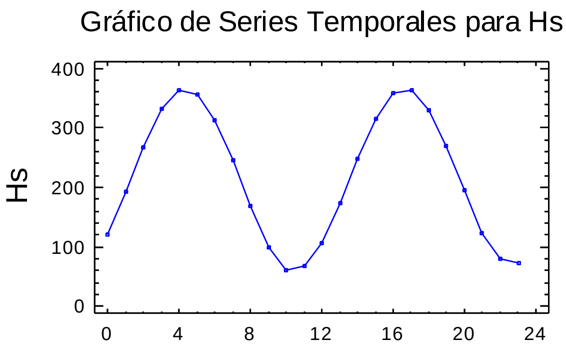
Sin embargo, la serie siguiente que representa el número de nacimientos en España entre los años 1946 al 2000 no es estacionaria y además no presenta una tendencia clara en todo el periodo de estudio. Se observa que la serie no es estable ya que su nivel al principio aumenta ligeramente y a partir de 1976 tiene una tendencia claramente negativa. La mayoría de las series económicas y sociales son no estacionarias.

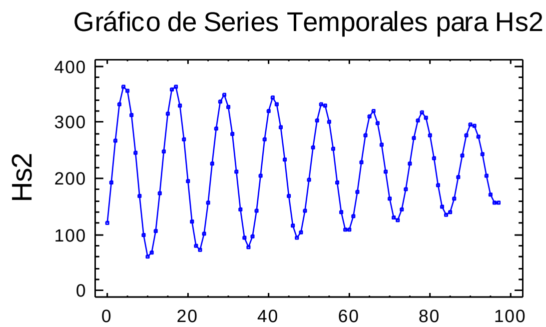


En muchas ocasiones las series oscilan alrededor de un valor central pero **se comportan de forma diferente dependiendo del periodo** esto es lo que llamaremos **estacionalidad.** Un ejemplo típico es la altura de ola medida en las boyas marinas. El siguiente gráfico recoge la altura de ola recogida en la boya de Villagarcía de Arosa de Enero a Diciembre del 2006 con una frecuencia horaria.

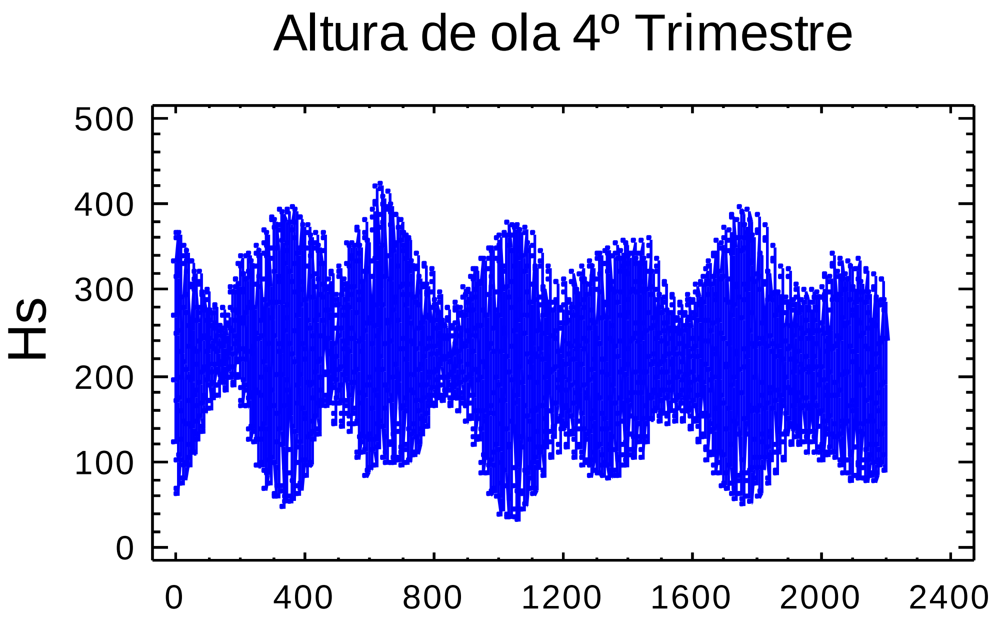


Aún así existen otras series con periodos de repetición aún más claros, como el nivel del mar, con fluctuaciones periódicas debidas a las mareas. Si representamos 24 horas de la variable nivel del mar en un puerto del Atlántico, como por ejemplo en el Ferrol, observamos claramente que presenta un comportamiento estacional cuyo periodo es de 12 horas.

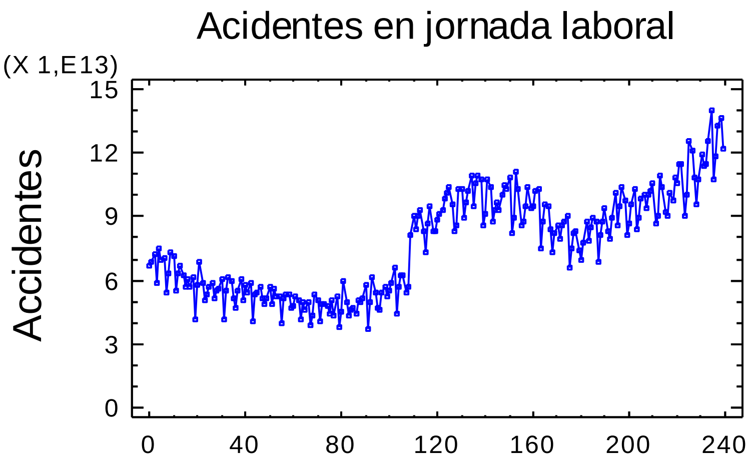




En ocasiones además de la estacionalidad, que se repite en periodos relativamente cortos (horas, días, meses,..), se presentan repeticiones del comportamiento de la serie en **periodos más largos que se denominan ciclos.** En la serie del nivel del mar se observa que en un trimestre tenemos seis repeticiones de ciclo que corresponden con los ciclos lunares (aproximadamente cada 28 días).



Además de estacionalidad podemos encontrar que nuestra serie no tiene tendencia constante, como por ejemplo el número de accidentes en jornada laboral en el periodo de enero de 1979 hasta diciembre de 1998.



En resumen, las series temporales pueden tener o no un nivel estable, y si no lo tienen pueden presentar o no tendencias más o menos constantes. Cuando el nivel de la serie no es estable decimos que la serie no es **estacionaria** y si se repite su comportamiento en periodos diremos que presenta **estacionalidad.** Además se pueden combinar tendencias claras con estacionalidad o ciclos. El gráfico de la serie es siempre una herramienta muy valiosa para entender su comportamiento.

1.2. REPRESENTACIÓN DE SERIES TEMPORALES.

Para representar series en R la función específica es plot ó autoplot, aunque la representación de la serie está incluida en los procedimientos de análisis de series como el forecast. Utilizaremos un ejemplo que contiene los datos diarios del precio de Bitcoin en dólares y el número de transacciones diarias.

Para cualquier análisis de series lo primero es crear un objeto timeseries, esto lo haremos con la **función ts** a partir de una tabla de datos.

La tabla de datos Bitcoin_A es importada desde Excel y a partir de ella creamos tres objetos time series, uno bidimensional y dos unidimensionales.

```
#Convert the data to time series

Bitcoin <- ts(bitcoin_A[,-1], start=c(2018,258), frequency=365)

precio <- ts(bitcoin_A[,2], start=c(2018,258), frequency=365)

ntrans <- ts(bitcoin_A[,3], start=c(2018,258), frequency=365)
```

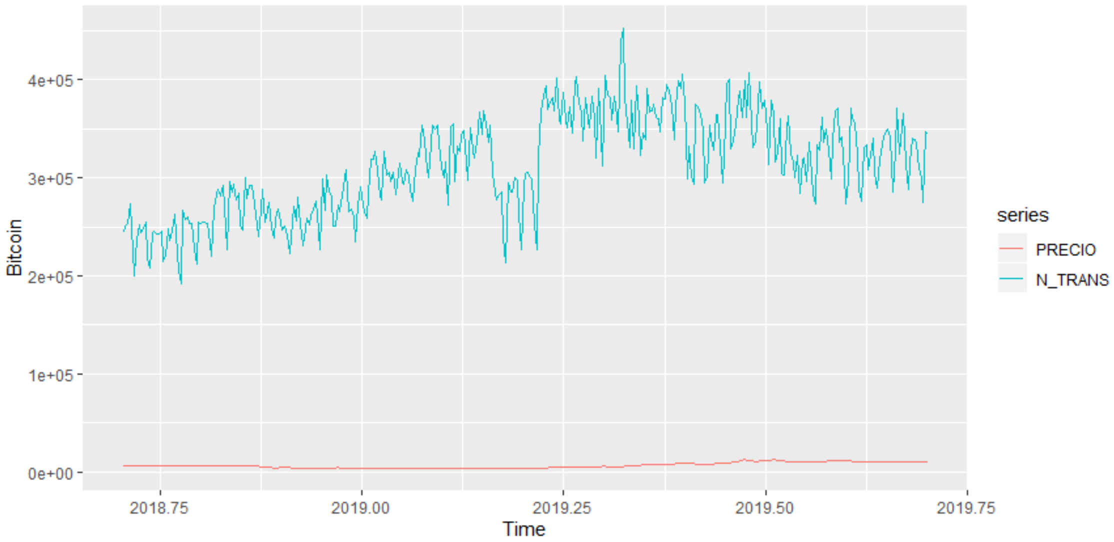
La opción start creará la variable fecha comenzando en la fecha indicada. El problema es que esta función está pensada con base un año y solo tiene dos argumentos: c(año, inicio). En nuestro ejemplo los datos están tomados por días, con la opción, *frequency=365*, y la fecha de inicio es 18/09/2018. Si por ejemplo los datos son meses y empieza en marzo sería start=c(2018,3), frecuency=12. En general los valores de frecuencia temporal más utilizados son:

Data	frequency
Annual	1
Quarterly	4
Monthly	12
Weekly	52

Para representar las dos series juntas utilizamos la sintaxis:

```
#Construcción de gráficas de series juntas
```

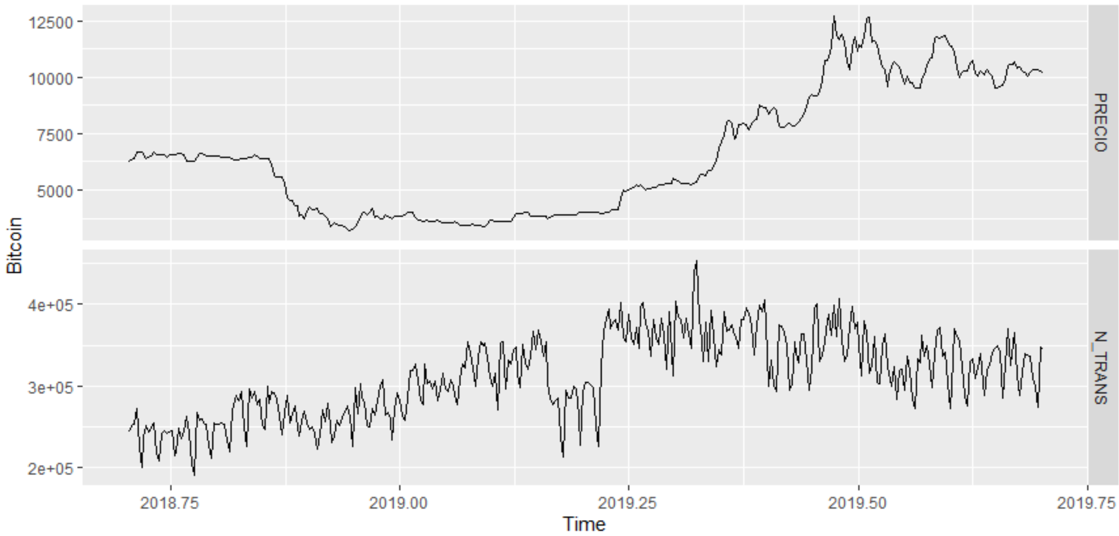
autoplot(Bitcoin))



Observemos que no es conveniente representar las series juntas por la diferencia de sus valores. Por esto las representamos por separado, en dos versiones, o las dos juntas en un mismo marco, o cada una individual:

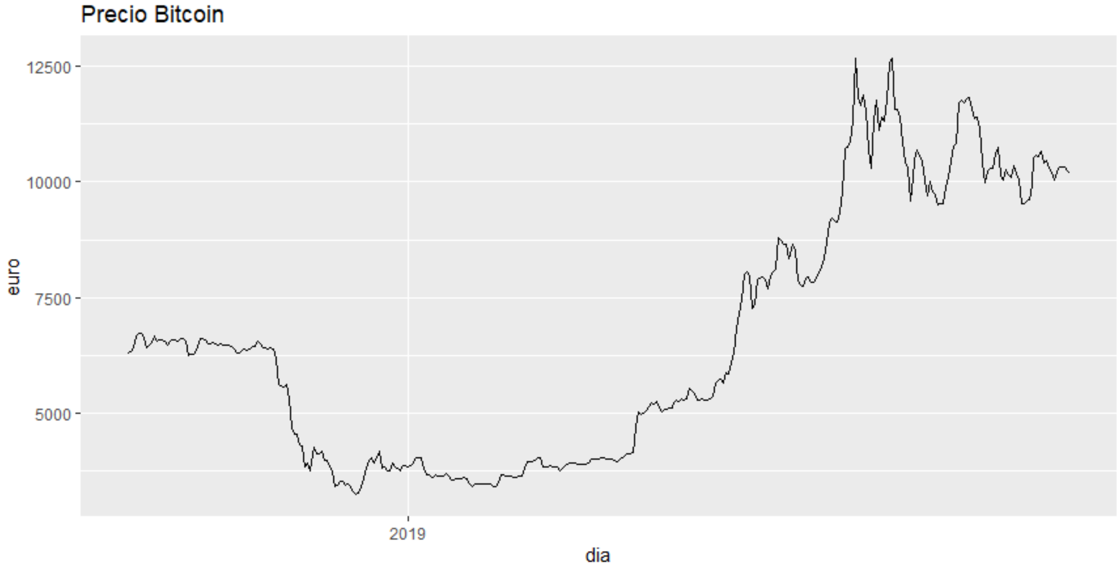
#Para representarlas juntas pero cada una con su escala

autoplot(Bitcoin, facets=TRUE)



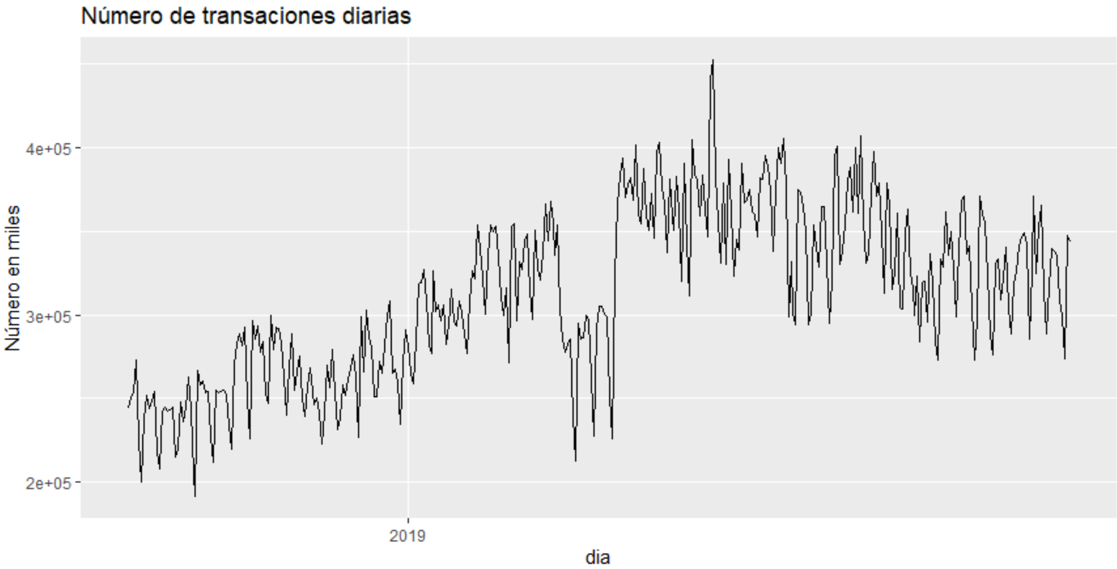
También podemos representarlas por separado añadiendo titulo a la gráfica y a los ejes.

```
autoplot(precio)+  ggtitle("Precio Bitcoin") +  xlab("dia") +  ylab("euro")
```



De la misma forma representamos el número de transacciones diarias

```
autoplot(ntrans)+  ggtitle("Número de transaciones diarias") +
  xlab("dia") +  ylab("Número en miles")
```



1.3. DESCOMPOSICIÓN ESTACIONAL

En este capítulo estudiaremos los métodos descriptivos cuyo objetivo es explicar la evolución pasada de la serie para predecir sus valores futuros. Para estudiar más a fondo una serie se descompone en varios elementos: **tendencia, componente estacional, componente cíclica y componente aleatoria,** como ya comentamos en el tema anterior**.** En general una serie puede ser expresada como suma de varios componentes

$$X_t = T_t + S_t + C_t + Z_t$$

Donde T_t representa la tendencia, S_t la componente estacional, C_t la componente cíclica y Z_t la innovación o término de error. Las series que además del comportamiento estacional presentan tendencia o la componente estacional aumenta con el tiempo se ajustan mejor al modelo multiplicativo

$$X_t = T_t \cdot S_t \cdot C_t \cdot Z_t$$

Además este modelo tiene la ventaja de una más sencilla interpretación de sus factores estacionales ya que están expresados en tanto por uno.

Debido a que el comportamiento cíclico de una serie no suele tener periodos tan claros como el estacional e involucrar grandes periodos de tiempo, esta componente se suele incluir dentro de la tendencia. Cuando la serie además de la tendencia y la componente aleatoria tiene una componente estacional se puede describir mediante el modelo aditivo o multiplicativo

$$X_t = T_t + S_t + Z_t \qquad X_t = T_t * S_t * Z_t$$

Si tenemos una serie con periodo *s,* para estimar cada una de estas componentes se utilizan **medias móviles para estimar la tendencia** y **las medias de cada "mes" para estimar la componente estacional.** Este método sigue los siguientes pasos:

1. Se ajusta la tendencia de la serie (T_t) calculando medias móviles de grado igual al periodo. A continuación se obtiene la serie $\hat{E}_t$ eliminando la tendencia $\left(\hat{E}_t = X_t - \hat{T}_t\right)$ ó $\left(\hat{E}_t = X_t / \hat{T}_t\right)$ que contiene la estacionalidad y el

componente aleatorio. Donde la tendencia se estima a partir de las medias móviles de orden *s*, periodo de la serie

$$\hat{T}_t = \sum_{k=-[s/2]}^{k=[s/2]} \frac{X_{t+k}}{s}$$

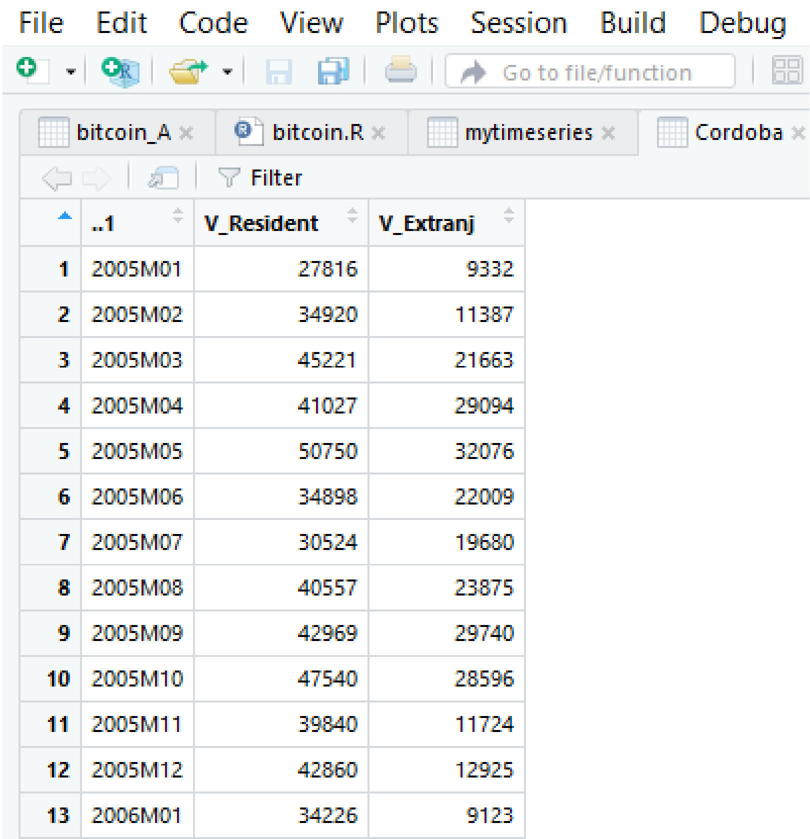
2. Los coeficientes estacionales se estiman en la serie sin tendencia como la diferencia, el cociente en el modelo multiplicativo, entre las medias de cada periodo, $j = 1, \ldots, s$, y la media global $\left(\hat{S}_j = \bar{E}_j - \bar{E}\right)$ó$\left(\hat{S}_j = \bar{E}_j / \bar{E}\right)$.
3. Calculamos la serie de innovaciones estimada restando (ó dividiendo) a la serie sin tendencia la componente estacional de cada observación$\left(\hat{Z}_t = E_t - \hat{S}_j\right)ó\left(\hat{Z}_t = E_t / \hat{S}_j\right)$
4. Si restamos (dividimos) de la serie original el coeficiente estacional de cada mes tenemos la serie desestacionalizada $\left(X_t - \hat{S}_t\right)ó\left(X_t / \hat{S}_t\right)$.
5. La predicción de la serie se realiza sumando la estimación de la tendencia y la desviación estacional correspondiente.

Tipo de modelo. El procedimiento ***Descomposición Estacional*** ofrece varios métodos para modelar los factores estacionales, nosotros estudiaremos los más utilizados: multiplicativo y aditivo.

- **Multiplicativo.** El componente estacional es un factor por el que se multiplica a la serie original para obtener la serie corregida estacionalmente.
- **Aditivo.** Para obtener los valores desestacionalizados se suman a la serie las correcciones estacionales.

En R podemos hacer la descomposición estacional utilizando varias funciones. Veamos su sintaxis básica sobre un ejemplo.

Ejemplo: El fichero Córdoba contiene datos mensuales de viajeros alojados en hoteles en Córdoba.



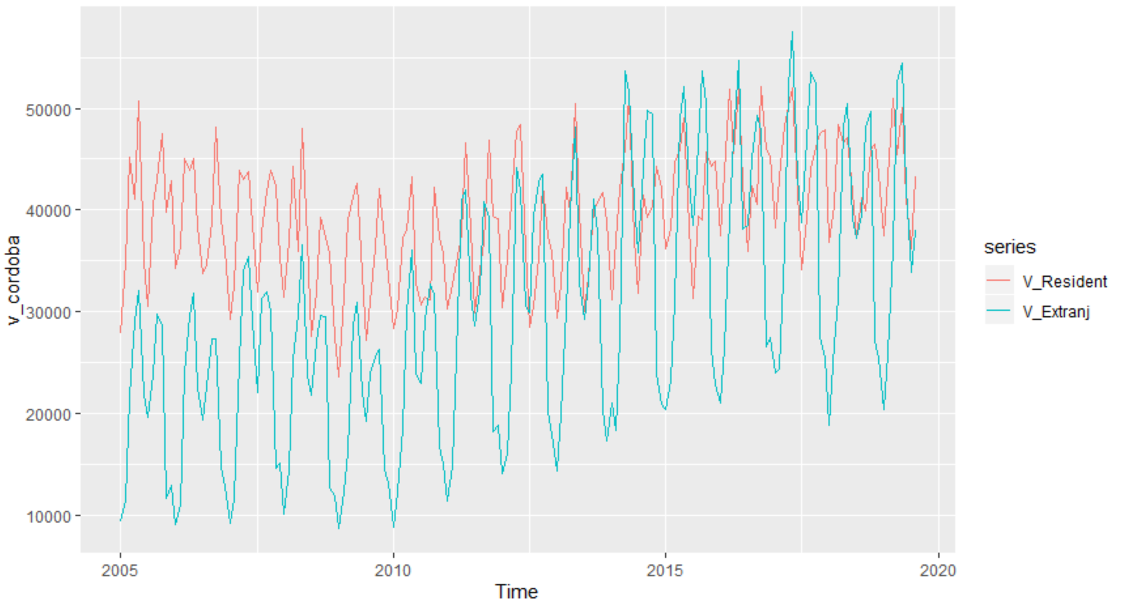
	..1	V_Resident	V_Extranj
1	2005M01	27816	9332
2	2005M02	34920	11387
3	2005M03	45221	21663
4	2005M04	41027	29094
5	2005M05	50750	32076
6	2005M06	34898	22009
7	2005M07	30524	19680
8	2005M08	40557	23875
9	2005M09	42969	29740
10	2005M10	47540	28596
11	2005M11	39840	11724
12	2005M12	42860	12925
13	2006M01	34226	9123

A partir de este fichero creamos tres objetos R tipo serie temporal. El primero contiene un serie bidimensional y los otros dos contienen las series unidimensionales de la columna 2 y 3. Observemos que la fecha es indicada por la opción start y le indicamos que son meses por la frecuencia 12.

```
#Dtos mensuales de viajeros alojados en hoteles de Córdoba
v_cordoba <- ts(Cordoba[,-1], start=(2005,1), frequency=12)
v_cordoba_R  <- ts(Cordoba[,2], start=(2005,1), frequency=12)
v_cordoba_E  <- ts(Cordoba[,3], start=(2005,1), frequency=12)
```

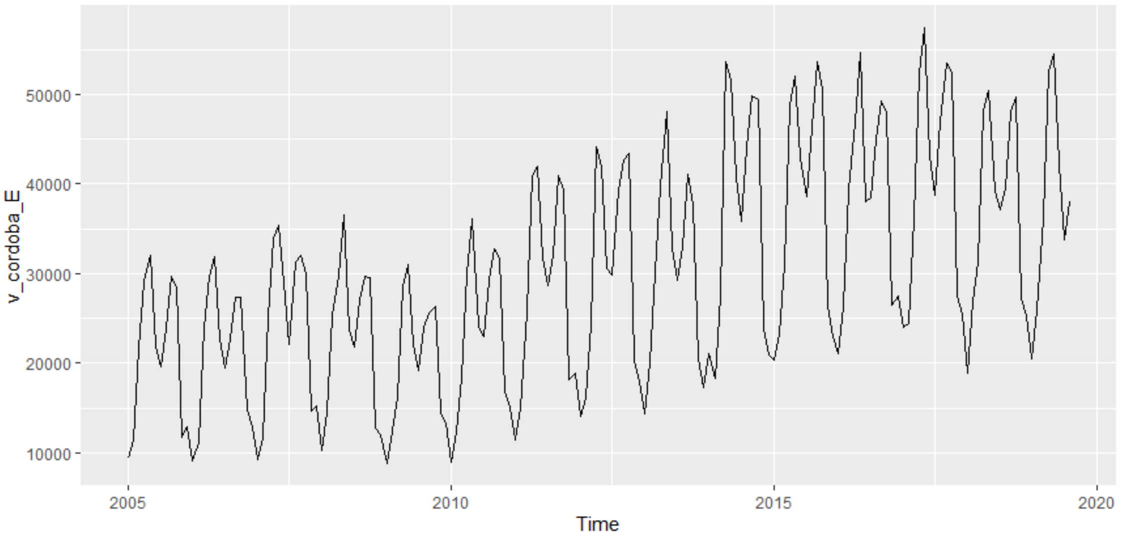
Comenzamos por representar las series de forma conjunta:

```
autoplot(v_cordoba)
```



Nos centramos en la serie viajeros extranjeros. La representamos:

autoplot(v_cordoba_E)



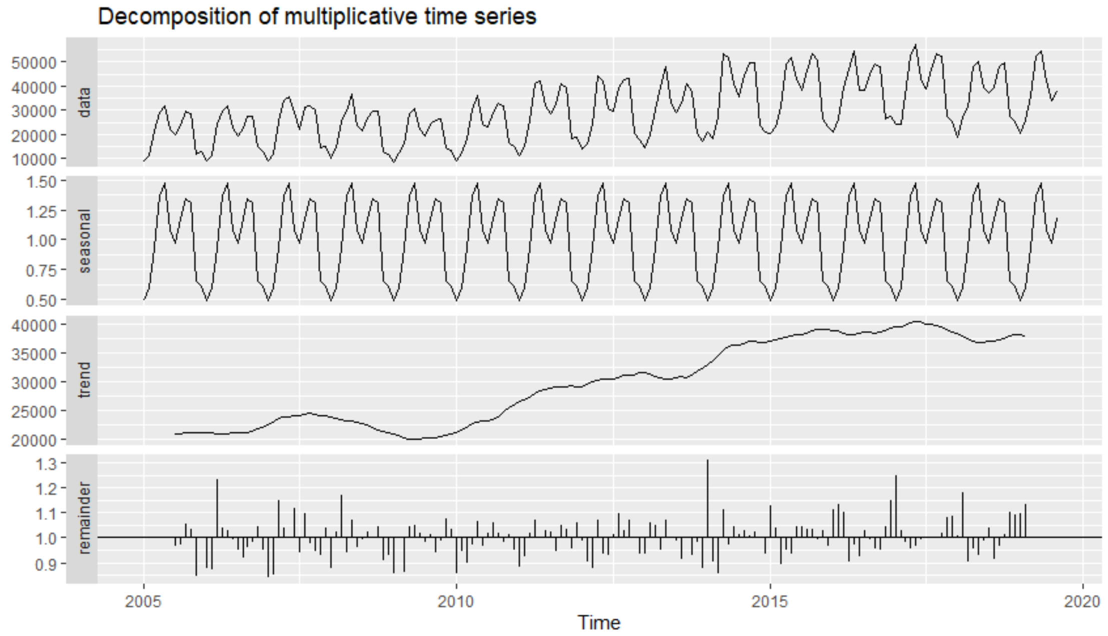
Realizamos la descomposición estacional según el modelo multiplicativo, esto debemos indicarlo porque por defecto ajusta el modelo aditivo y representamos sus componentes

#Guardamos los componentes de la descomposición estacional en

v_cordoba_E_Comp<- decompose(v_cordoba_E,type=c("multiplicative"))

Representamos los componentes de la serie obtenidos.

autoplot(v_cordoba_E_Comp)



Si observamos con más detalle la gráfica que representa la estacionalidad, podemos ver que es la representación de los coeficientes estacionales, por lo que se repite de manera constante. La tercera gráfica representa la tendencia y la cuarta la componente irregular

En v_cordoba_E_Comp tenemos guardadas las tablas de los diferentes componentes que se obtienen de la descomposición estacional

print(v_cordoba_E_Comp)

Por ejemplo, los coeficientes de estacionalidad son los siguientes.

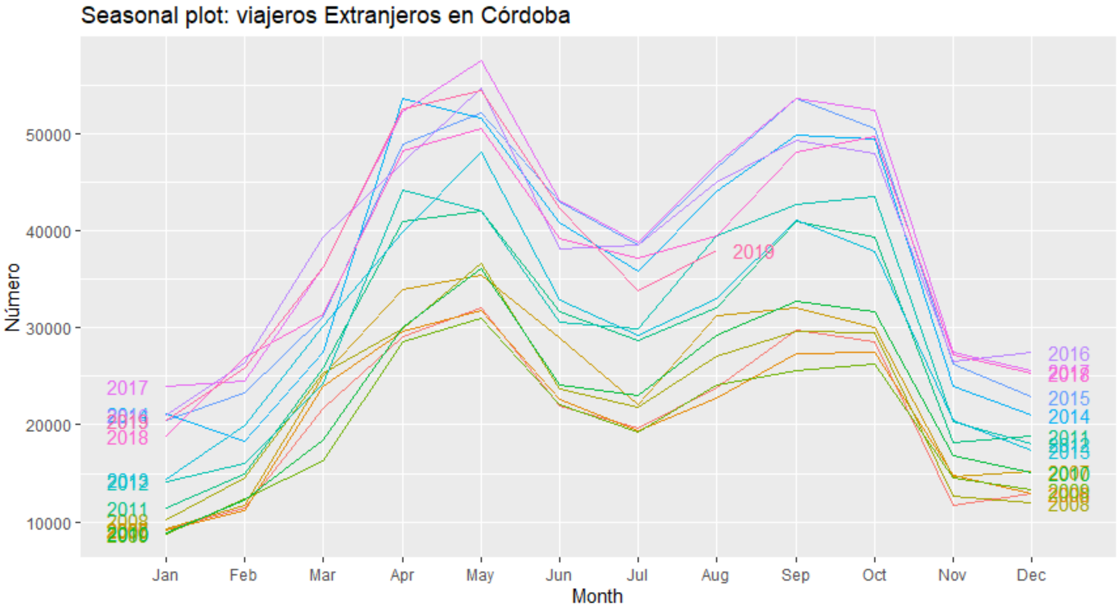
```
$seasonal

0.4896350 0.6019633 0.9276545 1.3666168 1.4767595 1.0800872 0.9695678
1.1740330 1.3413241 1.3078890 0.6538099 0.6106598
```

Observemos que, el mayor es 1.47, que corresponde al mes de mayo. Esto significa que en el mes de mayo hay un 47% más de viajeros en Córdoba que en la media del año.El menor de los coeficientes es el del mes de enero, 0.49, lo que significa que en el mes de enero hay un 51% menos de viajeros que en la media del año.

Para estudiar el comportamiento estacional resulta util la representación gráfica de los valores de la serie, dibujando cada año en un color diferente. Como podemos ver, todos los años tienen la misma estructura y se ve también que va aumentando el número de turistas cada año.

```
ggseasonplot(v_cordoba_E, year.labels=TRUE, year.labels.left=TRUE) +
 ylab("Número") +
 ggtitle("Seasonal plot: viajeros Extranjeros en Córdoba")
```



1.4. MÉTODOS DE SUAVIZADO.

En algunas series los parámetros de la función que ajusta la tendencia presentan pequeñas variaciones con el tiempo. En estos casos existe la posibilidad de utilizar modelos con parámetros variables que son estimados dando más importancia a los datos más recientes que a los antiguos, estos métodos se denominan de **suavizado exponencial porque los pesos que les damos a los valores de la serie decrecen de forma exponencial**. El objetivo es hacer predicciones utilizando los datos de la serie eliminando las fluctuaciones aleatorias y quedándonos solo con la componente tendencia o tendencia- estacionalidad

1.4.1. El modelo de alisado simple.

Este método se utiliza cuando la serie no presenta tendencia creciente o decreciente, es decir podemos modelizarla como

$$X_t = L_t + z_t$$

Incluso si la tendencia cambia lentamente con el tiempo podemos realizar su estimación dando mayor importancia a los valores más próximos al instante en el que estamos y menos a los más alejados en el tiempo. Comenzamos con una estimación inicial del valor medio de L_t que calculamos como la media de los primeros n_0 datos.

Supongamos que en instante t tenemos una estimación que es $\hat{x}_t$, entonces el valor actualizado de la estimación del parámetro para el instante $t+1$ se calcula como la media ponderada de la estimación en el del instante t y el valor de la serie en dicho instante:

$$\hat{x}_{t+1} = \alpha x_t + (1-\alpha)\hat{x}_t$$

Donde $0 \le \alpha \le 1$ determina la importancia que le damos a la estimación y al valor real. Para entender mejor este modelo sustituimos el valor de la estimación en el instante t

$$\hat{x}_{t+1} = \alpha x_t + (1-\alpha)\left(\alpha x_{t-1} + (1-\alpha)\hat{x}_{t-1}\right) = \alpha x_t + (1-\alpha)\alpha x_{t-1} + (1-\alpha)^2 \hat{x}_{t-1}$$

Repitiendo este proceso tenemos

$$\hat{x}_{t+1} = \alpha x_t + (1-\alpha)\alpha x_{t-1} + \ldots + \alpha(1-\alpha)^{t-1\cdot\cdot} x_1 + (1-\alpha)^t \hat{x}_1$$

Debemos darnos cuenta de que necesitamos tener una estimación del primer valor de la serie que puede obtenerse como la media de la serie o como la media de unos pocos valores iniciales.

Suponiendo que t es grande y α pequeño podemos aproximar por

$$\hat{x}_{t+1} = \alpha\left(x_t + (1-\alpha)x_{t-1} + (1-\alpha)^2 x_{t-2} + \ldots\right)$$

Que es la media ponderada de las observaciones anteriores con pesos decrecientes que suman uno.

La ecuación de predicción para valores futuros supuesto que hemos observado hasta el tiempo n es $\hat{x}_{n+m} = \hat{x}_{n+1} \quad m \ge 2$.

Ejemplo. La libreria forecast nos permite hacer los métodos de suavizado explicados anteriormente. Utilizaremos los datos del precio de Bitcoin que mostramos antes, pero nos quedamos con los datos finales. Para ello utilizamos la función window() que es muy util para extraer parte de los datos de una serie.

#Seleccionamos los valores de precio desde agosto de 2019 precio_R<-window(precio,start=c(2019,210))

Hacemos un suavizado exponencial simple con la función ses y calculamos la #predicción para 7 dias

precio_s1=ses(precio_R, h=7)

Para ver todo la información relativa al modelo de suavizado hacemos:

print(precio_s1)

Parte de la salida de esta función, en donde están los valores ajustados y las predicciones, son los parámetros estimados. En este caso:

```
$par
     alpha          l
    0.9999 11253.8011
```

Luego la función utilizada para hacer el suavizado es:

$$\hat{x}_{t+1} = 0.999 x_t + 0.001 \hat{x}_t \, .$$

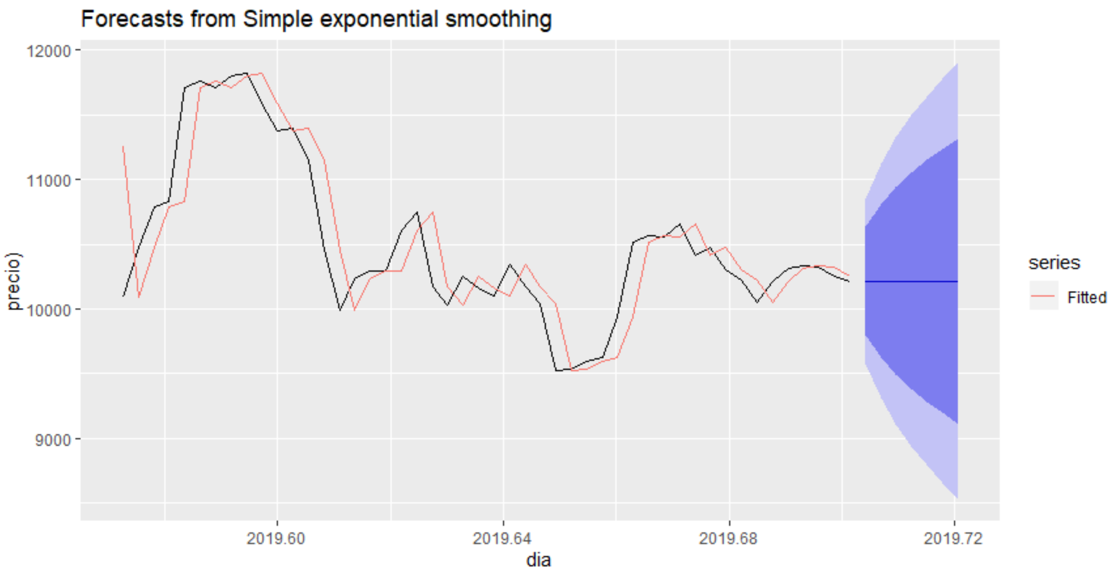
#Representamos los valores observados y los suavizados con la predicción para 7 #días

autoplot(precio_s1) +

autolayer(fitted(precio_s1), series="Fitted") +

ylab("precio)") + xlab("dia")

Observemos que este método de suavizado no se utiliza mucho porque la predicción permanece constante para todos los días.



1.4.2. Método de alisado doble de Holt.

El método de suavizado anterior no nos proporciona buenos resultados cuando los datos tienen una tendencia que varía más rápidamente. Este método supone que la tendencia es lineal $x_t = L_t + b_t t + z_t$ pero su pendiente va variando en el tiempo. De la serie original obtendremos la serie suavizada mediante las ecuaciones:

$$L_t = \alpha x_t + (1-\alpha)(L_{t-1} + b_{t-1})$$

$$b_t = \beta(L_t - L_{t-1}) + (1-\beta) b_{t-1}$$

$$\hat{x}_{t+1} = L_t + b_t$$

Donde α y β son dos constantes entre cero y uno. Cuando se utilizan estas dos constantes iguales se denomina método de Brown. Para aplicar este método debemos determinar las constantes de suavizado y los valores de L_1 y b_1, que normalmente se toman como $L_1 = x_1$ y $b_1 = x_2 - x_1$. Estudios realizados sobre los valores posibles de las constantes de suavizado indican que los valores más adecuados están entre 0.05 y 0.3. Los valores más pequeños se utilizarían para series con poca variación en la tendencia y valores más altos para aquellas en las que las variaciones son mayores. Sas calcula estos coeficientes utilizando optimización, es decir, aquellos que minimizan la suma de los errores (diferencia entre valor real y suavizado) al cuadrado.

La ecuación de predicción para valores futuros, supuesto que hemos observado hasta el tiempo n es:

$$\hat{x}_{n+m} = L_n + b_n m \quad m \geq 1$$

Ejemplo: Continuamos con el ejemplo del precio de Bitcoin y ahora utilizamos la función holt:

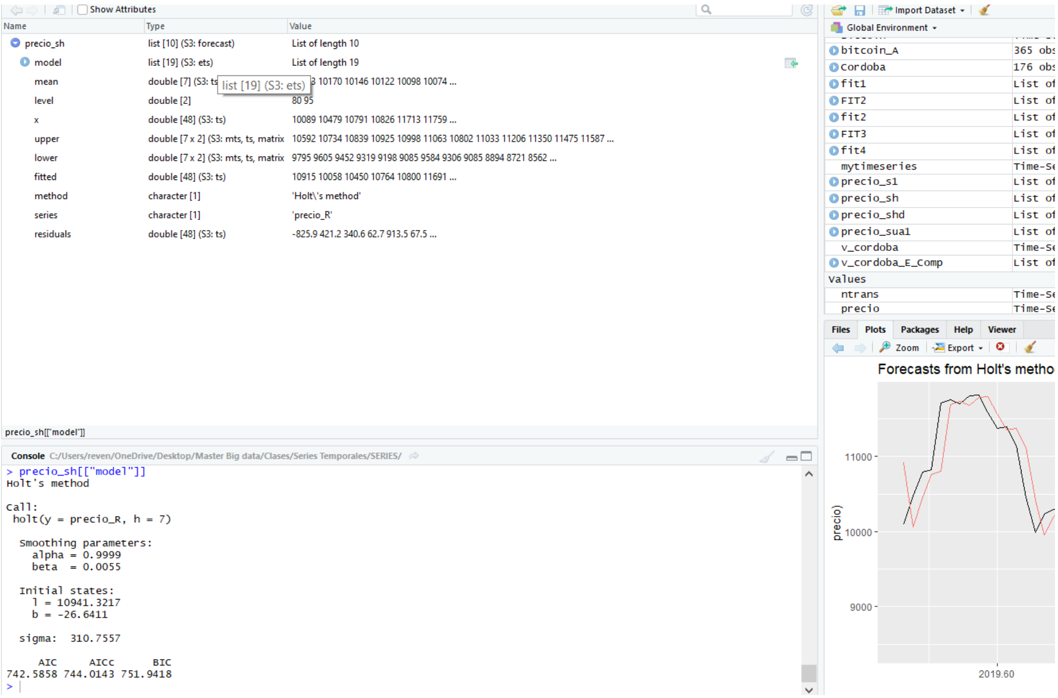
```
precio_sh <- holt(precio_R, h=7)
```

```
print(precio_sh)
```

Con esta función print nos muestra las predicciones con el intervalo de confianza:

Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2019.7041	10193.37	9795.118	10591.62	9584.298	10802.44
2019.7068	10169.53	9604.791	10734.27	9305.835	11033.23
2019.7096	10145.70	9452.132	10839.26	9084.982	11206.41
2019.7123	10121.86	9318.799	10924.92	8893.683	11350.04
2019.7151	10098.03	9197.706	10998.35	8721.106	11474.95
2019.7178	10074.19	9085.232	11063.15	8561.709	11586.67
2019.7205	10050.35	8979.229	11121.48	8412.209	11688.50

Para ver los parámetros del modelo hacemos click en precio_sh y en la model

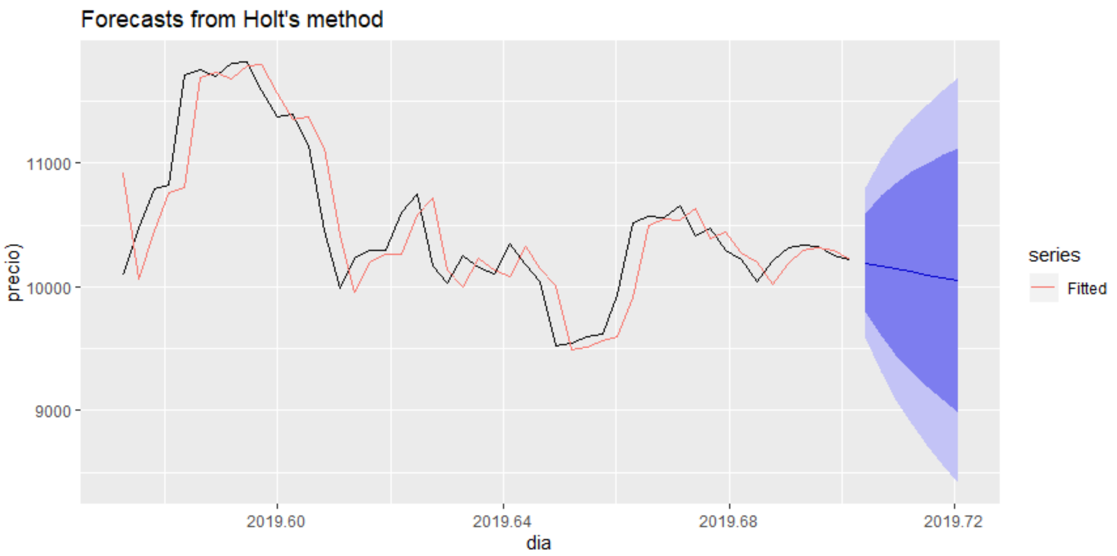



```
> precio_sh[["model"]]
  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.0055

  Initial states:
    l = 10941.3217
    b = -26.6411
```

Por último representamos los valores de la serie con los suavizados y las predicciones

```
autoplot(precio_sh) +
 autolayer(fitted(precio_sh), series="Fitted") +
 ylab("precio)") + xlab("dia")
```



Como podemos observar este método da unas predicciones más adecuadas puesto que tiene en cuanta los cambios en la tendencia.

1.4.3. Método de suavizado para series con estacionalidad: Holt-Winters.

Cuando existe estacionalidad el efecto debe modelarse con un modelo aditivo, $X_t = (L_t + b_t) + S_t + z_t$, si la incidencia de la estacionalidad no aumenta con el tiempo, o con un modelo multiplicativo, $X_t = (L_t + b_t) \cdot S_t + z_t$ si las variaciones estacionales aumentan con el tiempo. La serie suavizada para el modelo multiplicativo se obtiene aplicando las siguientes fórmulas:

$$L_t = \alpha \frac{x_t}{S_{t-s}} + (1-\alpha)(L_{t-1} + b_{t-1})$$

$$b_t = \beta(L_t - L_{t-1}) + (1-\beta) b_{t-1}$$

$$S_t = \gamma \frac{x_t}{L_t} + (1-\gamma) S_{t-s}$$

$$\hat{x}_{t+1} = (L_t + b_t) S_{t-s+1}$$

Para el modelo aditivo es prácticamente igual solo que los coeficientes estacionales se restan en lugar de dividir por ellos:

$$L_t = \alpha\left(x_t - S_{t-s}\right) + \left(1-\alpha\right)\left(L_{t-1} + b_{t-1}\right)$$

$$b_t = \beta\left(L_t - L_{t-1}\right) + \left(1-\beta\right)b_{t-1}$$

$$S_t = \gamma\left(x_t - L_t\right) + \left(1-\gamma\right)S_{t-s}$$

$$\hat{x}_{t+1} = \left(L_t + b_t\right) + S_{t-s+1}$$

En este modelo los valores iniciales de la tendencia se estiman a partir de la media de los valores del primer ciclo

$$L_0 = \frac{z_1 + \ldots.. + z_s}{s}$$

Para estimar la pendiente necesitamos conocer al menos dos ciclos completos para calcular las diferencias, y se inicia con:

$$b_0 = \frac{1}{s}\left(\frac{\left(x_{s+1} - x_1\right)}{s} + \frac{\left(x_{s+2} - x_2\right)}{s} + \ldots + \frac{\left(x_{s+s} - x_s\right)}{s}\right)$$

Finalmente los índices estacionales se estiman con los valores del primer periodo:

En el modelo multiplicativo:

$$S_1 = \frac{x_1}{L_0}, S_2 = \frac{x_2}{L_0}, \ldots S_s = \frac{x_s}{L_0},$$

En el modelo aditivo:

$$S_1 = x_1 - L_0, S_2 = x_2 - L_0, \ldots, S_s = x_s - L_0$$

Las ecuaciones de predicción para valores futuros supuesto que hemos observado hasta el tiempo n que proponen los métodos aditivo y multiplicativo son:

. $\hat{x}_{n+m} = \left(L_n + b_n m\right)S_{n-s+m} \quad m \geq 1$ para el multiplicativo

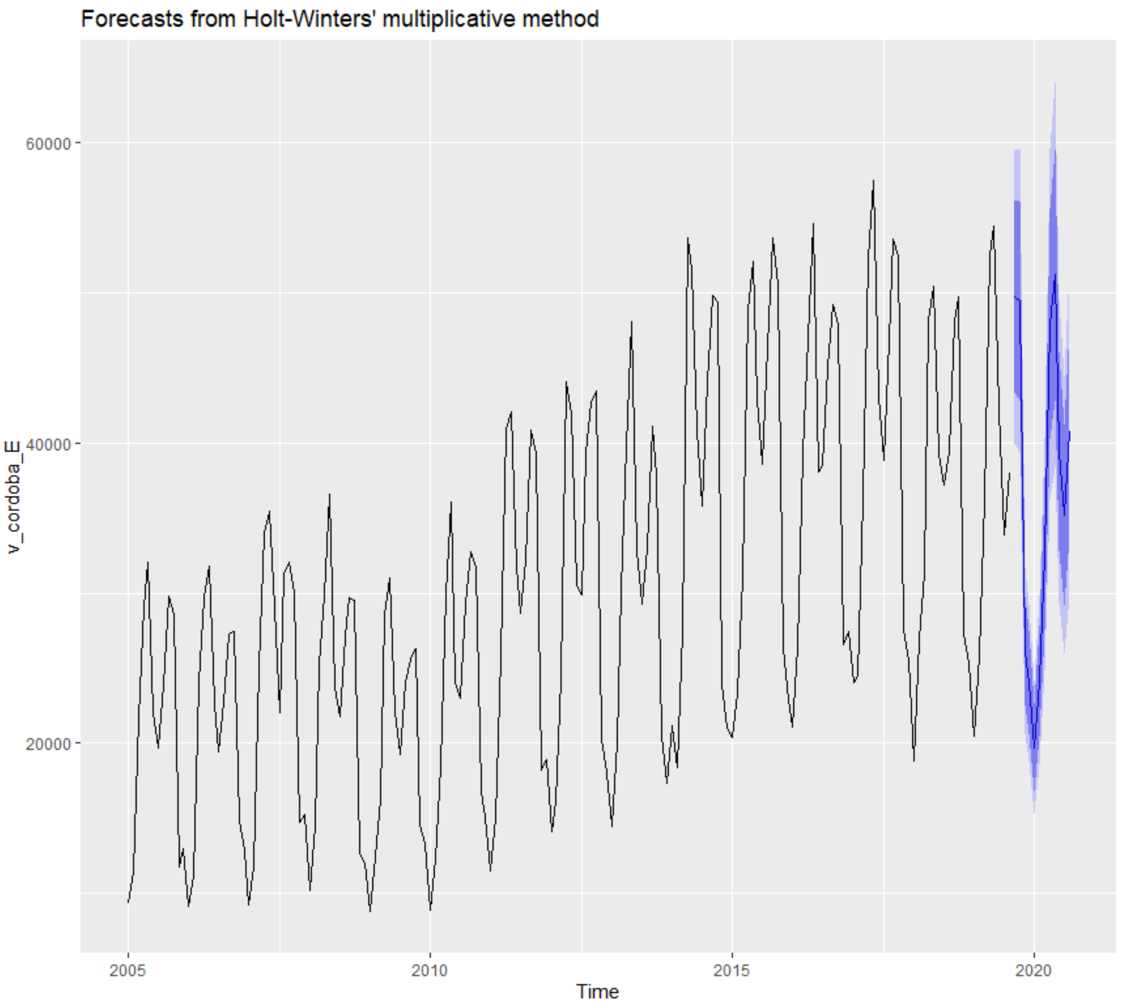
$\hat{x}_{n+m} = \left(L_n + b_n m\right) + S_{n-s+m} \quad m \geq 1$ para el aditivo

Por ejemplo, si queremos predecir el número de viajeros extranjeros que vendrán a Córdoba un año despues del ultimo observado, utilizaremos el modelo de Holt-winters multiplicativo porque la serie es estacional. Guardamos los resultados del modelo ajustado en fit1.

```
fit1 <- hw(v_cordoba_E,h=12, seasonal="multiplicative",level = c(80, 95))
```

```
autoplot(fit1)
```

Observemos que hemos calculado los intervalos de predicción para un nivel de confianza del 80% y de un 95%. Estos intervalos son representados en dos colores azules alrededor de las predicciones. Las predicciones en modelos estacionales se calculan para un periodo más al último observado.



Como podemos ver en el gráfico el método de suavizado de Holt-Winters para series con estacionalidad proporciona unas predicciones muy buenas dado que sus intervalos de predicción son muy ajustados.

```
print(fit1)
```

Esta ópción nos permite ver los valores exactos de las predicciones así como los intervalos de predicción para los dos niveles calculados

		Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Sep	2019	49749.65	43350.41	56148.89	39962.86	59536.44
Oct	2019	49481.59	42875.06	56088.12	39377.77	59585.41
Nov	2019	26136.11	22523.34	29748.87	20610.86	31661.35
Dec	2019	24048.22	20614.43	27482.00	18796.69	29299.74
Jan	2020	19712.70	16810.88	22614.52	15274.75	24150.65
Feb	2020	24148.23	20489.90	27806.57	18553.29	29743.18
Mar	2020	32601.97	27526.92	37677.01	24840.36	40363.57
Apr	2020	47931.60	40275.49	55587.71	36222.59	59640.61
May	2020	51233.81	42847.22	59620.40	38407.63	64060.00
Jun	2020	39472.75	32858.62	46086.87	29357.32	49588.18
Jul	2020	35212.55	29179.09	41246.01	25985.17	44439.93
Aug	2020	40809.86	33666.30	47953.43	29884.73	51735.00
Jun	2020	39472.75	32858.62	46086.87	29357.32	49588.18
Jul	2020	35212.55	29179.09	41246.01	25985.17	44439.93
Aug	2020	40809.86	33666.30	47953.43	29884.73	51735.00

Los parámetros del modelo ajustados son:

```
fit1[["model"]]
```

```
Smoothing parameters:
    alpha = 0.2769
    beta  = 1e-04
    gamma = 0.3342
```

$$L_t = 0.2769\frac{x_t}{S_{t-s}} + (1-0.2769)(L_{t-1}+b_{t-1})$$

$$b_t = 0.0001(L_t - L_{t-1}) + 0.9999b_{t-1}$$

$$S_t = 0.3344\frac{x_t}{L_t} + (1-0.3344)S_{t-s}$$

$$\hat{x}_{t+1} = (L_t + b_t)\cdot S_{t-s+1}$$

Bibliografía:

https://otexts.com/fpp2/

Avril Coghlan. A Little Book of R For Time Series.

Librería Forecast de R