

R_Básico_7

J.Ballesta

18/06/25

Resumen

En esta publicación presentamos una introducción al análisis de series temporales, y el método Holt-Winters.

In this publication, we present an introduction to time series analysis and introduce the Holt-Winters method.

In dieser Veröffentlichung geben wir eine Einführung in die Zeitreihenanalyse und stellen die Holt-Winters methode vor.

Introducción.

En esta publicación veremos un análisis sencillo sobre series temporales, usando el lenguaje de programación R(2024), el IDE RStudio(Posit team 2024) y Quarto (RStudio Team 2022) para formatear el fichero de resultados.

Una serie temporal es una secuencia de valores medidos a intervalos de tiempo sucesivos y específicos. Estos valores pueden ser cualquier tipo de dato, desde valores económicos (evolución del precio de venta de un determinado artículo, precio de las acciones de una empresa, el PIB de un país...) hasta datos científicos (temperatura media diaria, población de una especie, ...) La clave es el orden cronológico, ya que el tiempo es una dimensión fundamental que influye en el comportamiento de los datos.

Sin extendernos mucho, las series temporales tienen cuatro componentes :

1. Tendencia T_t : Representa el movimiento a largo plazo de la serie. Puede ser ascendente, descendente o relativamente plana.
2. Estacionalidad S_t : Patrones que se repiten en intervalos fijos y conocidos (diariamente, semanalmente, ...).
3. Ciclo C_t : fluctuaciones que ocurren en periodos más largos que la estacionalidad y no son de duración fija. Suelen estar asociados a ciclos económicos o de negocio.
4. Ruido o residuo ϵ_t : es la parte de la serie temporal que no puede ser explicada por la tendencia la estacionalidad o el ciclo

Estos cuatro componentes se pueden evaluar en dos tipos de modelos :

1. Modelo aditivo $Y_t = T_t + S_t + C_t + \epsilon_t$ (cuando la magnitud de las fluctuaciones estacionales no cambia con el nivel de la serie).
2. Modelo multiplicativo $Y_t = T_t * S_t * C_t * \epsilon_t$ (cuando la magnitud de las fluctuaciones estacionales aumenta o disminuye con el nivel de la serie).

En esta publicación analizaremos un ejemplo sencillo de análisis de una serie univariante.

Librerías.

Como siempre, en primer lugar cargamos las librerías que vamos a necesitar en esta publicación.

```
— Attaching core tidyverse packages — tidyverse 2.0.0
—
✓ dplyr 1.1.4   ✓ readr 2.1.5
✓ forcats 1.0.0 ✓ stringr 1.5.1
✓ ggplot2 3.5.1 ✓ tibble 3.2.1
✓ lubridate 1.9.4 ✓ tidyr 1.3.1
✓ purrr 1.0.2
— Conflicts —
tidyverse_conflicts() —
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag() masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
Registered S3 method overwritten by 'quantmod':
  method      from
as.zoo.data.frame zoo
```

Conjunto de datos.

Analizaremos el conjunto de datos de Eurostat sobre generación de energía eléctrica en España desde el año 2020 a finales del 2024, disponible en el enlace (https://ec.europa.eu/eurostat/databrowser/view/nrg_cb_em_custom_15997382/default/line?lang=en) (Eurostat 2022), descargo los datos de la web de Eurostat y los subo a R con la función `readcsv()`

```
#
# cargo datos del fichero descargado de la web de Eurostat
datos <- read.csv("estat_nrg_cb_em_filtered_en.csv")
# como siempre presento la cabecera de los datos
head(datos, 5)
```

	DATAFLOW	LAST.UPDATE	freq	nrg_bal
1	ESTAT:NRG_CB_EM(1.0)	26/03/25 11:00:00	Monthly	Available to internal market
2	ESTAT:NRG_CB_EM(1.0)	26/03/25 11:00:00	Monthly	Available to internal market
3	ESTAT:NRG_CB_EM(1.0)	26/03/25 11:00:00	Monthly	Available to internal market
4	ESTAT:NRG_CB_EM(1.0)	26/03/25 11:00:00	Monthly	Available to internal market
5	ESTAT:NRG_CB_EM(1.0)	26/03/25 11:00:00	Monthly	Available to internal market

	siec	unit	geo	TIME_PERIOD	OBS_VALUE	OBS_FLAG	CONF_STATUS
1	Electricity	Gigawatt-hour	Spain	2020-01	21548.20		NA
2	Electricity	Gigawatt-hour	Spain	2020-02	18969.58		NA
3	Electricity	Gigawatt-hour	Spain	2020-03	18880.41		NA
4	Electricity	Gigawatt-hour	Spain	2020-04	15387.42		NA
5	Electricity	Gigawatt-hour	Spain	2020-05	16504.28		NA

```
# obtengo información sobre su estructura y tipo de datos
str(datos)

'data.frame': 61 obs. of 11 variables:
 $ DATAFLOW      : chr  "ESTAT:NRG_CB_EM(1.0)" "ESTAT:NRG_CB_EM(1.0)"
"ESTAT:NRG_CB_EM(1.0)" "ESTAT:NRG_CB_EM(1.0)" ...
 $ LAST.UPDATE: chr  "26/03/25 11:00:00" "26/03/25 11:00:00" "26/03/25 11:00:00" "26/03/25
11:00:00" ...
 $ freq      : chr  "Monthly" "Monthly" "Monthly" "Monthly" ...
 $ nrg_bal    : chr  "Available to internal market" "Available to internal market" "Available to
internal market" "Available to internal market" ...
 $ siec      : chr  "Electricity" "Electricity" "Electricity" "Electricity" ...
 $ unit      : chr  "Gigawatt-hour" "Gigawatt-hour" "Gigawatt-hour" "Gigawatt-hour" ...
 $ geo       : chr  "Spain" "Spain" "Spain" "Spain" ...
 $ TIME_PERIOD: chr  "2020-01" "2020-02" "2020-03" "2020-04" ...
 $ OBS_VALUE  : num  21548 18970 18880 15387 16504 ...
 $ OBS_FLAG   : chr  "" "" "" "" ...
 $ CONF_STATUS: logi  NA NA NA NA NA NA ...

#
# Con DataExplorer comprobamos los datos, para ver como están distribuidas las variables y
si # hay datos faltantes
# Preparación de los datos elijo las columnas que me interesan para la serie temporal, la
#fecha y el valor
datos_ts <- datos |>
  select(TIME_PERIOD, OBS_VALUE)
#
plot_intro(datos_ts,
  title = "Dataset Supply, tranformation and consumption of electricity",
  ggtheme = theme_bw())
```

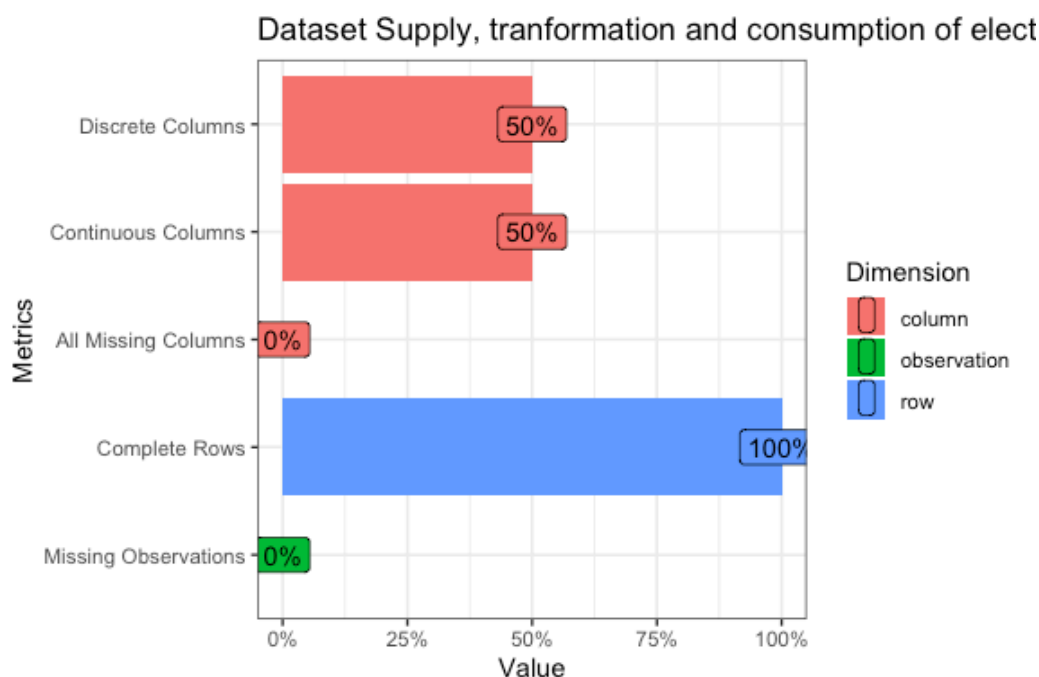


Figura 1: Distribución de los datos

Vemos de [Figura 1](#) no tenemos datos faltantes, y que la información es completa.

Análisis de series temporales.

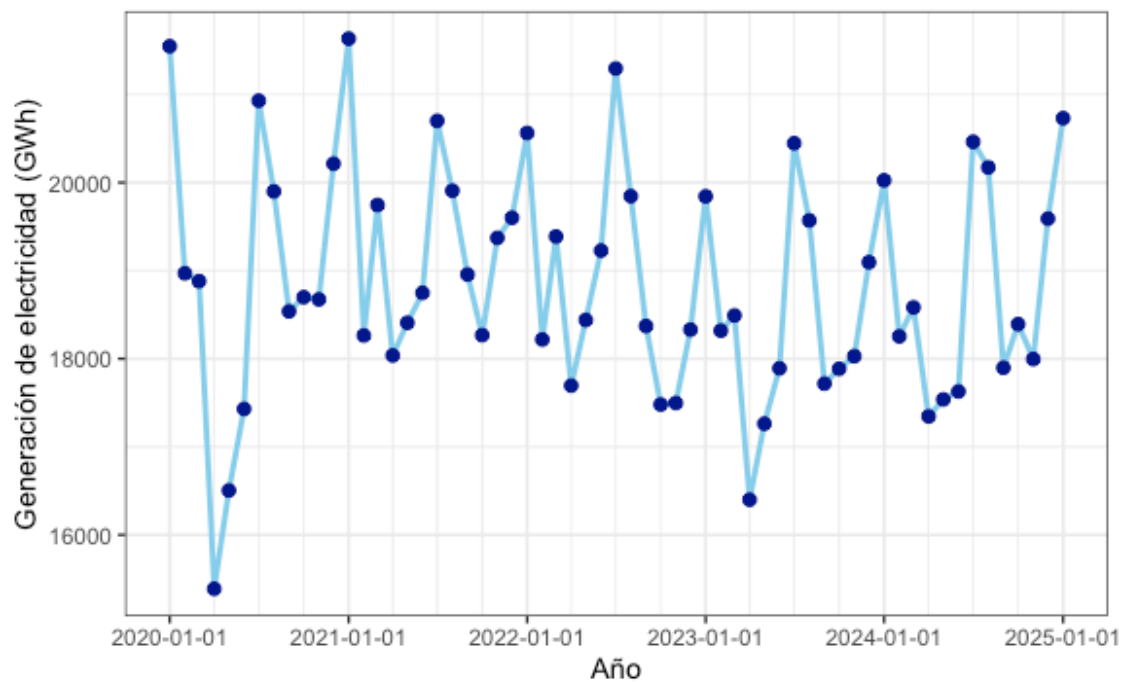
El primer paso es convertir a una serie temporal el dataframe con los datos que hemos obtenido:

```
#
# La cadena de texto yyyy-mm la convierto a formato fecha
datos_ts$TIME_PERIOD <- ym(datos_ts$TIME_PERIOD)
#
# Hago una serie temporal con la función ts(), como tengo 12 meses por año ->
frequency=12
datos_analisis <- ts(datos_ts$OBS_VALUE,
                     start =c(as.numeric(format(datos_ts$TIME_PERIOD[1], "%Y")),
as.numeric(format(datos_ts$TIME_PERIOD[1], "%m"))),
                     frequency = 12)
```

Representamos gráficamente la serie temporal:

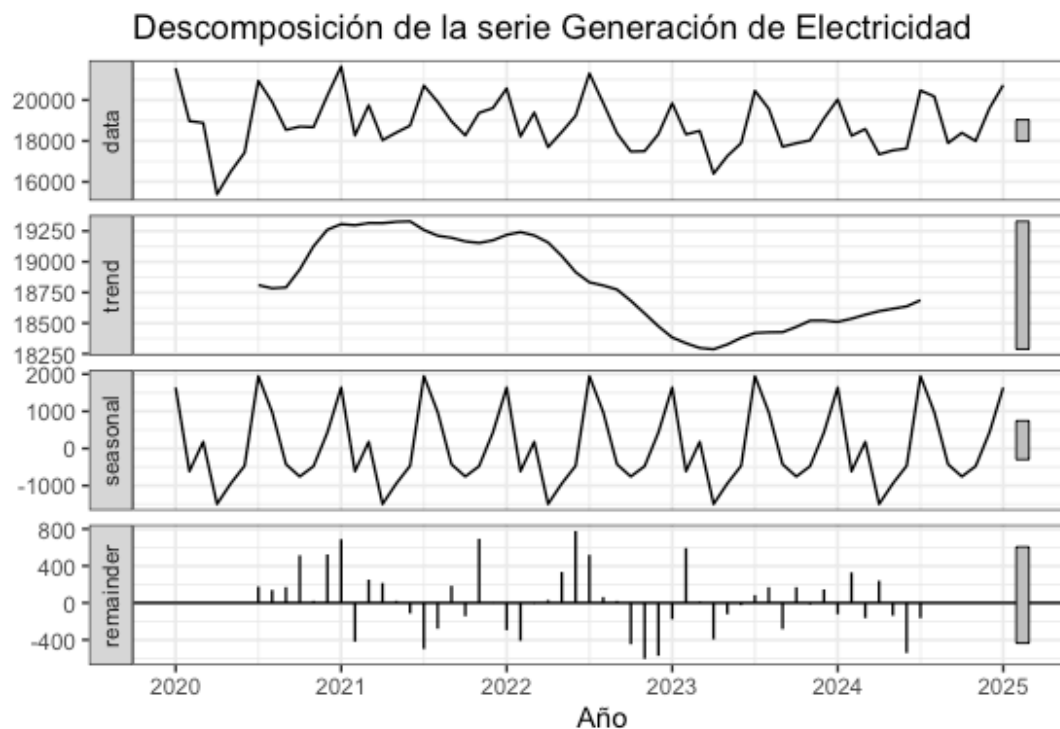
```
#
# representamos gráficamente los datos de la serie. En este caso usaremos la
#capa de scale_x_date() para formatear las fechas en el eje X, mostrando periodos de 1 año y
#trimestres
ggplot(datos_ts, aes(x=TIME_PERIOD, y=OBS_VALUE) )+
  geom_line(color = "skyblue", linewidth = 1) +
  geom_point(color = "darkblue", size = 2) +
  scale_x_date(date_breaks = "1 year",date_minor_breaks = "3 months" )+
  labs(title = "Ejemplo de Serie Temporal: Evolución de la generación de electricidad",
       x = "Año",
       y = "Generación de electricidad (GWh)")
```

Ejemplo de Serie Temporal: Evolución de la generación de electricidad



Vemos que la serie es relativamente estable en el tiempo, con una bajada de valores muy acusada en el 2020. Vamos a visualizar sus componentes, descomponiéndola mediante la función `decompose()` (hay otra función para la descomposición llamada `stl()`):

```
#  
# con la función decompose() descomponemos la serie en sus valores, la tendencia, la  
# estacionalidad y el error  
descomposicion <-decompose(datos_analisis)  
# presentamos la salida gráfica  
autoplot(descomposicion)+  
  labs(title="Descomposición de la serie Generación de Electricidad",  
        x="Año")
```



Estacionariedad, Autocorrelación, ACF y PACF.

Imaginemos que tratamos de predecir el caudal de un río. Si el río cambia constantemente sus parámetros de velocidad, caudal, curso... de forma impredecible ¡será muy difícil hacer predicciones! Pero si el río varía sus parámetros de manera predecible, predecir será mucho más fácil. Las series temporales son así y la clave es la estacionariedad para hacer predicciones confiables.

Una serie temporal es **estacionaria** si sus propiedades estadísticas (medida, varianza y estructura de autocorrelación) permanecen constantes en el tiempo. Si la serie no fuera estacionaria hemos de diferenciarla (calcular la diferencia entre observaciones consecutivas)

La **autocorrelación** es simplemente la correlación de una serie temporal consigo misma en diferentes puntos en el tiempo. Nos dice cuánto se parece un valor actual a un valor pasado.

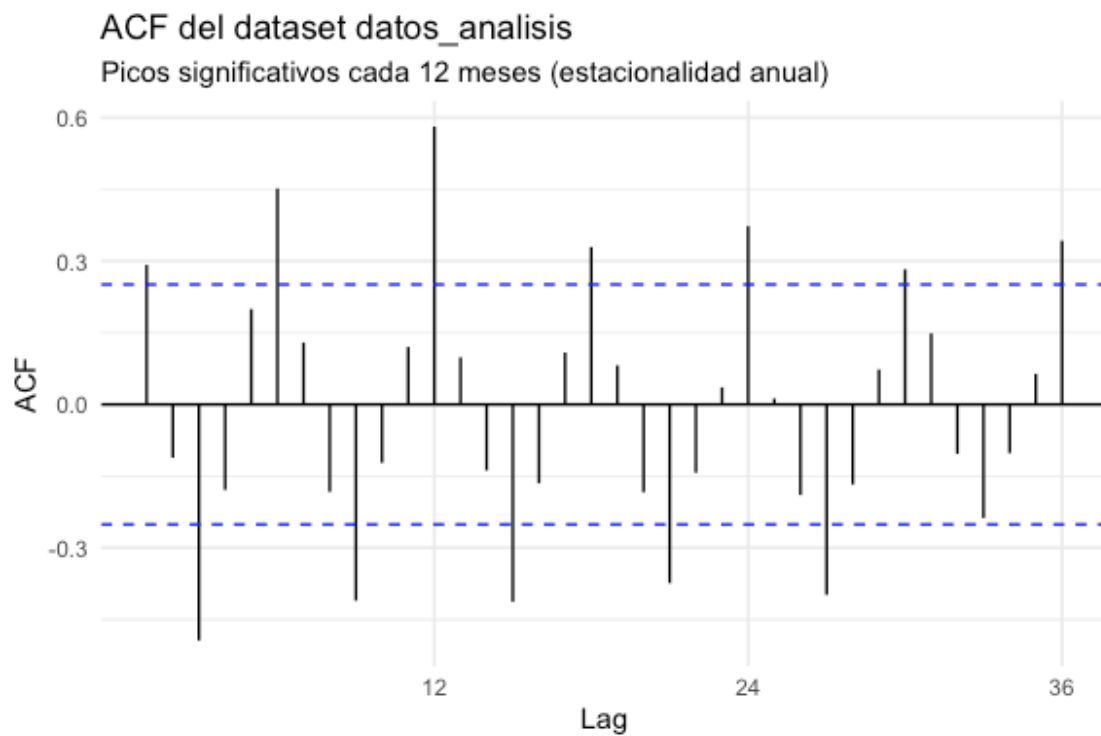
La función de autocorrelación (ACF) mide la correlación entre una observación y una observación anterior, **incluyendo** los efectos de las observaciones intermedias. La ACF nos ayuda a identificar NO estacionariedad, estacionalidad y para un modelo ARIMA el orden MA(q)

La función de autocorrelación parcial (PACF) mide la correlación entre una observación y una observación anterior después de eliminar la influencia de todas las observaciones intermedias. PACF nos ayuda a identificar el orden AR(p) de un modelo ARIMA.

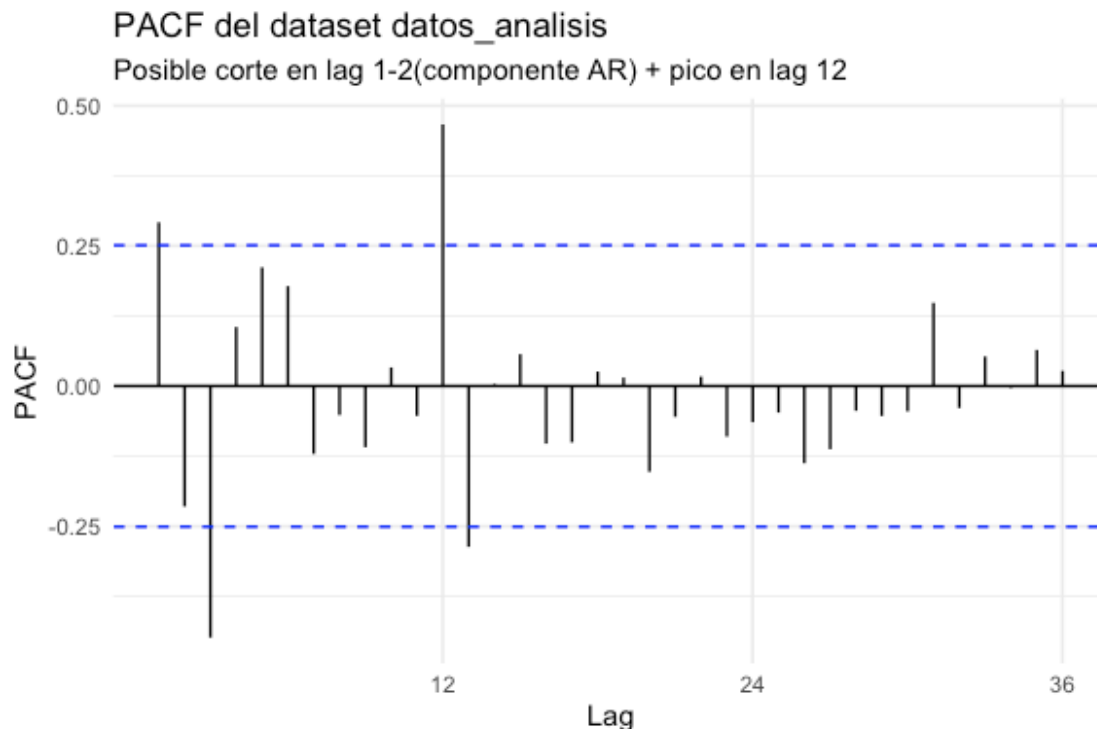
Para nuestra serie, gráficamente podemos obtener ACF y PACF mediante:

```
#
# Visualizamos la ACF de la serie original
```

```
ggAcf(datos_analisis, lag.max=36) +
  labs(title="ACF del dataset datos_analisis",
        subtitle="Picos significativos cada 12 meses (estacionalidad anual)") +
  theme_minimal()
```



```
#
ggPacf(datos_analisis, lag.max=36) +
  labs(title="PACF del dataset datos_analisis",
        subtitle="Posible corte en lag 1-2(componente AR) + pico en lag 12")+
  theme_minimal()
```



Como complemento al análisis gráfico, numéricamente podemos hacer la prueba de estacionariedad con la función `adf.test()`, si el valor de p-value es menor que el valor de significancia v.g. 0,05 se puede rechazar la hipótesis nula y concluir que la serie es estacionaria.

```
#
# Prueba numérica estacionaridad (Prueba de Dickey-Fuller Aumentada)
#Ho - la serie temporal NO es estacionaria
#H1 - La serie temporal es estacionaria
adf.test(datos_analisis)
```

Warning in `adf.test(datos_analisis)`: p-value smaller than printed p-value

Augmented Dickey-Fuller Test

```
data: datos_analisis
Dickey-Fuller = -4.5201, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

Modelo de suavizado exponencial (Holt-Winters).

Los modelos de suavizado exponencial funcionan bajo esta premisa: Las observaciones más recientes de la serie temporal tienen más peso o importancia en el pronóstico futuro que las observaciones más antiguas. La influencia de las observaciones pasadas disminuye exponencialmente a medida que se alejan en el tiempo.

Los modelos Holt-Winters, son útiles para series como la nuestra que exhiben tendencia y/o estacionalidad. los tipos de modelos más comunes son :

1. Suavizado Exponencial Simple (SES) - series temporales que no tienen tendencia ni estacionalidad significativa.
2. Método de Holt (doble suavizado exponencial) - series temporales que presentan una tendencia, pero no estacionalidad.
3. Método de Holt-Winters (triple suavizado exponencial) para series temporales que presentan tanto tendencia como estacionalidad.

```
#
# Lanzamos un modelo Holt-Winters aditivo con la función ets(datos, Error, Tendencia,
#Estacionalidad )
#si observamos la descomposición de la serie, elegimos un modelo aditivo :
# - la amplitud de las oscilaciones estacionales parece ser cte. a lo largo del tiempo
# - la tendencia parece sumarse a un nivel base
# - El error ( remainder) parece cte. a lo largo del tiempo
modelo_hw_aaa <- ets(datos_analisis, model="AAA")
# Mostramos : el tipo de modelo seleccionado, los valores de los parametros de suavizado y
metricas de error utiles para compara distintos modelos
summary(modelo_hw_aaa)

ETS(A,A,A)

Call:
ets(y = datos_analisis, model = "AAA")

Smoothing parameters:
alpha = 0.7329
beta = 1e-04
gamma = 4e-04

Initial states:
l = 19039.6335
b = 27.868
s = 458.3365 -490.9975 -754.4932 -430.3442 981.2538 1929.61
-479.292 -915.2485 -1536.322 181.9904 -533.2817 1588.788

sigma: 612.9454

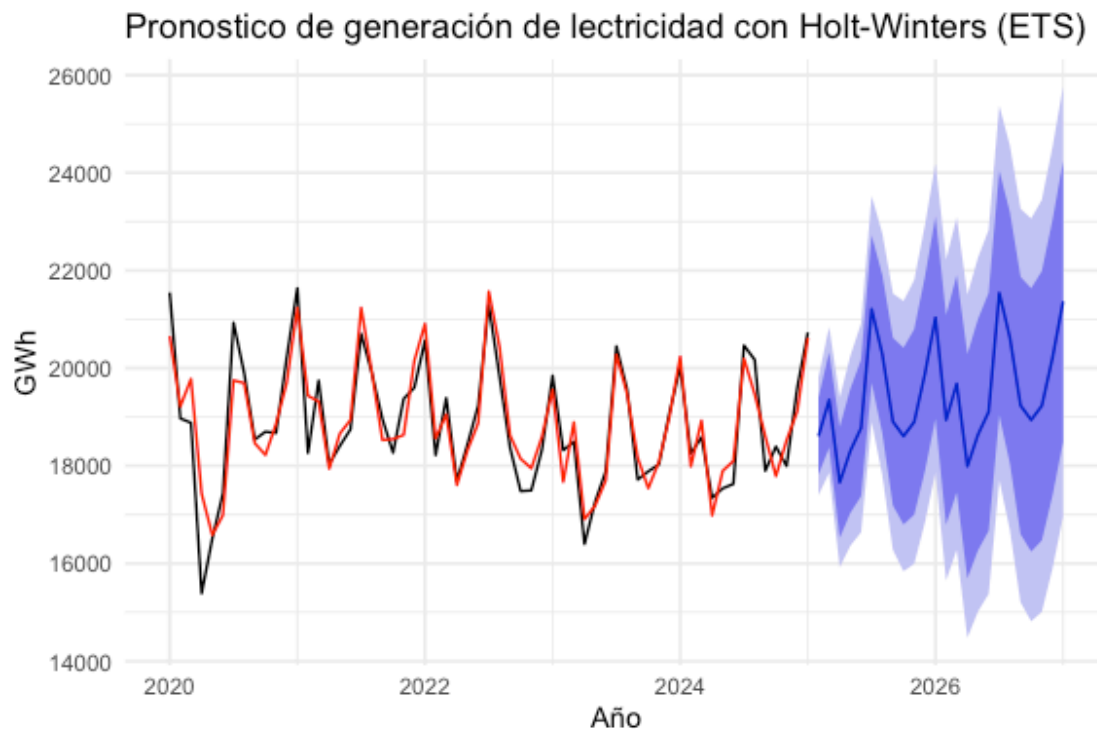
AIC AICc BIC
1049.236 1063.469 1085.121

Training set error measures:
ME RMSE MAE MPE MAPE MASE
Training set -36.1931 526.4573 411.4881 -0.2835218 2.214024 0.6546593
ACF1
Training set 0.02763832

# Pronosticamos la serie para los siguientes 24 meses
pronostico_hw <- forecast(modelo_hw_aaa, h=24)
# Mostramos la serie original y sobre ella en rojo el modelo calculado.
```

#- la linea azul oscuro a partir del final de la serie original, representan los pronosticos
 #- las areas sombreadas azul más claro y más oscuro son los intervalos de confianza (80% y 90% respectivamente)

```
autoplot(pronostico_hw)+
  labs(title="Pronostico de generación de lectricidad con Holt-Winters (ETS)",
        x="Año", y="GWh")+
  theme_minimal()+
  autolayer(fitted(pronostico_hw), series="Valores ajustados", color="red")
```



```
#
print(pronostico_hw)
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Feb 2025	18606.73	17821.21	19392.25	17405.38	19808.08
Mar 2025	19349.63	18375.71	20323.56	17860.14	20839.13
Apr 2025	17658.52	16527.10	18789.94	15928.16	19388.88
May 2025	18307.84	17038.27	19577.40	16366.20	20249.47
Jun 2025	18771.76	17377.64	20165.87	16639.64	20903.88
Jul 2025	21208.50	19700.05	22716.95	18901.53	23515.47
Aug 2025	20287.61	18672.88	21902.35	17818.09	22757.14
Sep 2025	18903.15	17188.68	20617.63	16281.10	21525.21
Oct 2025	18607.20	16798.45	20415.94	15840.96	21373.43
Nov 2025	18897.95	16999.59	20796.31	15994.66	21801.24
Dec 2025	19875.22	17891.26	21859.18	16841.01	22909.42
Jan 2026	21033.66	18967.63	23099.70	17873.93	24193.40
Feb 2026	18938.45	16793.37	21083.53	15657.84	22219.07
Mar 2026	19681.36	17460.10	21902.61	16284.24	23078.47
Apr 2026	17990.24	15695.32	20285.16	14480.46	21500.02
May 2026	18639.56	16273.24	21005.87	15020.59	22258.52
Jun 2026	19103.48	16667.85	21539.11	15378.50	22828.46

Jul 2026	21540.22	19037.17	24043.27	17712.13	25368.31
Aug 2026	20619.34	18050.61	23188.06	16690.82	24547.86
Sep 2026	19234.87	16602.10	21867.65	15208.40	23261.35
Oct 2026	18938.92	16243.60	21634.23	14816.79	23061.05
Nov 2026	19229.67	16473.21	21986.13	15014.03	23445.31
Dec 2026	20206.94	17390.65	23023.23	15899.80	24514.08
Jan 2027	21365.39	18490.49	24240.28	16968.62	25762.15

El modelo Holt-Winters es una herramienta muy potente porque captura bien tanto la tendencia a largo plazo como patrones estacionales repetitivos, lo que lo hace ideal para pronósticos en series como las ventas minoristas, el turismo, el consumo de energía, etc.

En esta publicación hemos analizado un ejemplo sencillo, para los interesados en profundizar en el tema, les recomiendo: [Forecasting: Principles and Practice \(3rd ed\)](#) y su serie de vídeos en Youtube : [YouTube : Forecasting Principles and Practice](#)

Referencias

Eurostat. 2022. «Supply, transformation and consumption of electricity - monthly data». Eurostat. https://doi.org/10.2908/NRG_CB_EM.

Posit team. 2024. *RStudio: Integrated Development Environment for R*. Boston, MA: Posit Software, PBC. <http://www.posit.co/>.

R Core Team. 2024. «R: A Language and Environment for Statistical Computing». <https://www.R-project.org/>.

RStudio Team. 2022. *Quarto: A next-generation publishing system*. RStudio, PBC. <https://quarto.org/>.