

# La predicción de series temporales mediante el modelo de regresión

*(Time series prediction using the regression model)*

**Dr. Agustín ALONSO RODRIGUEZ**  
Real Centro Universitario  
“Escorial-María Cristina”  
San Lorenzo del Escorial

**Resumen:** En este trabajo se expone cómo la predicción en las series temporales se puede obtener acudiendo al modelo de regresión. Utilizando la serie *Índice* de la Red Eléctrica de España, REE, en su apartado: demanda eléctrica para la construcción de edificios, desde enero de 2010 hasta agosto del 2020, datos corregidos, se muestra paso a paso cómo lograrlo.

**Abstract:** In this paper, it is shown how the prediction of a time series can be achieved using the regression model. The time series *Índice* of Red Eléctrica de España, REE, section: demand of electricity for buildings construction, from January 2020 to August 2020, it is used as an illustration.

**Palabras clave:** Serie *Índice*, demanda de electricidad, construcción de edificios, Red Eléctrica de España, modelo de regresión, predicción, series temporales, paquetes estadísticos: forecast, R, TSstudio.

**Keywords:** Time series: *Índice* of Red Eléctrica de España, demand of electricity, buildings construction, regression model, time series prediction, software: forecast, R, TSstudio.

## Sumario:

- I. **Introducción.**
- II. **La Demanda de energía eléctrica para la construcción de edificios, la serie Índice de REE.**
- III. **El estudio de las series de tiempo.**

- IV. Modelos de regresión para la serie Demanda de energía eléctrica.**
- V. Modelos validados y ajuste de sus predicciones.**
- VI. Un modelo ARIMA.**
- VII. Conclusiones.**
- VIII. Bibliografía.**

**Recibido: noviembre 2020.**

**Aceptado: enero 2021.#**

## I. INTRODUCCIÓN

La descomposición de una serie temporal en sus elementos *estructurales*: la tendencia, la estacionalidad, el ciclo, y el componente irregular, permiten incorporar, en el marco del modelo de regresión, el análisis de las series temporales.

En la descomposición de una serie temporal en sus elementos estructurales, cabe distinguir una doble estructura, a saber: aditiva y multiplicativa.

Si designamos por  $Y_t$  la serie temporal, podemos escribir la estructura *aditiva* como

$$Y_t = T_t + S_t + C_t + I_t \quad (1)$$

siendo:

$T_t$ : la tendencia, *trend* en inglés

$S_t$ : el componente estacional, *season*, relacionado con la frecuencia de la serie: mensual, trimestral, etc.

$C_t$ : el componente cíclico

$I_t$ : el componente irregular.

Y por lo que respecta a la estructura *multiplicativa* tenemos

$$Y_t = T_t \times S_t \times C_t \times I_t \quad (2)$$

siendo el significado de los símbolos como en el caso anterior.

La estructura *aditiva* en (1), permite asociarla con el modelo de regresión lineal al escribir, sin tener en cuenta el componente cíclico, incorporado al elemento estacional

$$Y_t = \beta_1 + \beta_2 T_t + \beta_3 S_t + \varepsilon_t \quad (3)$$

siendo ahora

$Y_t$ : la variable respuesta, y  $T_t$  y  $S_t$  las variables explicativas, asociando el término *irregular* con el término de *error* del modelo de regresión:  $\mathcal{E}_t$ .

Por lo que respecta a la estructura *multiplicativa* en (2), mediante la transformación logarítmica de sus componentes, podemos hacerla *aditiva* escribiendo

$$\log(Y_t) = \beta_1 + \beta_2 \log(T_t) + \beta_3 \log(S_t) + \mathcal{E}_t$$

Advertencia. Dada la amplitud del tema y las restricciones de espacio de toda publicación, para facilitar la comprensión, se ha optado por las representaciones gráficas a costa de las explicaciones pormenorizadas de los resultados. De esta manera, espero haber logrado el objetivo fijado en el título del artículo.

## II. LA SERIE ÍNDICE DE RED ELÉCTRICA DE ESPAÑA.

A fin de mostrar el lado aplicado de este enfoque para el análisis de las series temporales, utilizaré la serie *Índice* de la Red Eléctrica de España, en su apartado: demanda de energía eléctrica para la construcción de edificios, una serie mensual (datos corregidos) de 128 observaciones, desde enero del 2010 hasta agosto del 2020 (datos accedidos el 12-09-2020). Con el nombre *demanda* la serie aparece representada en la figura 1.

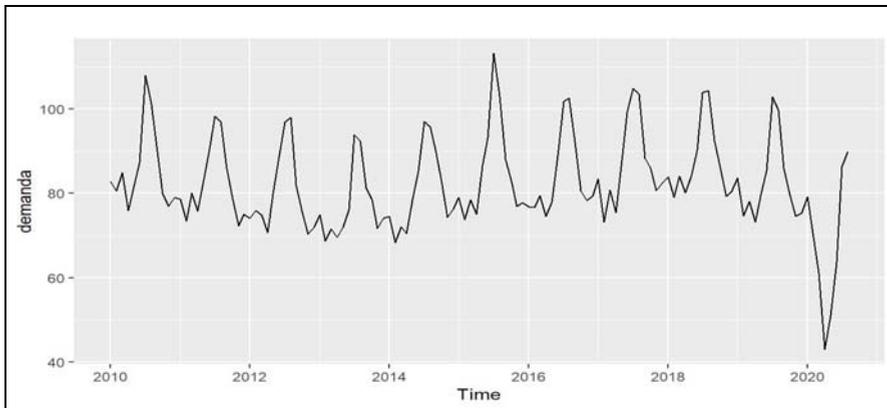


Figura 1. Demanda de electricidad para la construcción de edificios.

### III. EL ESTUDIO DE LAS SERIES DE TIEMPO

A lo largo de los años, el análisis de las series temporales se ha llevado a cabo tanto en el ámbito de la frecuencia, como en el ámbito del tiempo. En este último ámbito, y desde la aparición en 1976 de la obra de *Box y Jenkins, Time Series Analysis: Forecasting and Control*, el análisis de las series de tiempo se ha llevado a cabo, preferentemente, sobre la base de la información contenida en la función de autocorrelación y de autocorrelación parcial de la serie en estudio: los modelos **ARIMA**.

Gracias a Rob. J. Hyndman y colaboradores, 2008, el paquete *forecast*, parte del sistema estadístico *fpp (Forecasting: Principles and Practice)* permite, gracias su función *tslm* aplicar el análisis de las series temporales en el marco de la regresión. Así mismo, hay que destacar los trabajos de Rami Krispin, quien en 2019, publica *Hands-on Time Series Analysis with R*, acompañado por su paquete de software *TSstudio*. Más recientemente R. Krispin ha presentado el software *forecastLM*, todavía en desarrollo, dedicado todo él al análisis de las series temporales en el marco de la regresión.

Tanto Hyndman como Krispin presentan todo un sistema de *software* para el análisis de las series temporales, con excelentes capacidades gráficas. La diferencia fundamental es que *forecast* utiliza *ggplot2* para las representaciones gráficas, con su fácil adaptación al formato PDF para publicaciones. Por su parte, Krispin utiliza *Plotly* para sus gráficos. Destacar también que Krispin posibilita la incorporación de *forecast* a su software, permitiendo que algunas funciones de ambos sistemas funcionen conjuntamente.

### IV. MODELOS DE REGRESIÓN PARA LA SERIE *DEMANDA DE ENERGÍA ELÉCTRICA*

Comenzamos nuestro análisis particionando la serie *demanda* en dos subseries: la subserie *train*, sobre la que se estima el modelo, y la subserie *test* que permite obtener estadísticos sobre lo acertado del ajuste. Por tratarse de datos mensuales, la subserie *test* comprende las 12 últimas observaciones, todo un año, figura 2.

Hecha esta partición, vamos a estimar con la función *tslm* tres modelos para nuestra serie: un modelo para la *tendencia*, un modelo para el *componente estacional* y un tercer modelo, incorporando ambos componentes.

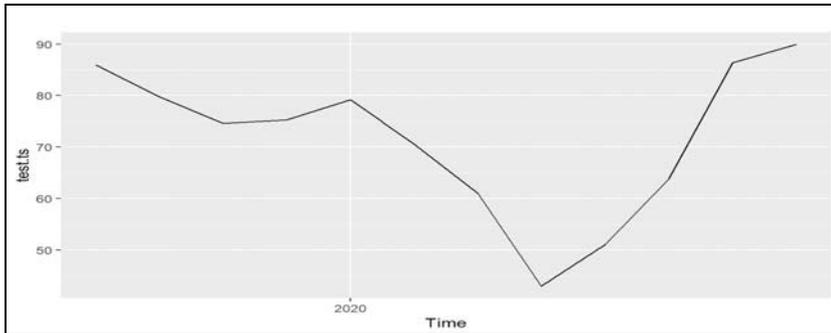


Figura 2. Las 12 observaciones de la subserie test.

### Modelo para la tendencia: *modelo1*

El modelo a estimar es:

$$Y_t = \beta_1 + \beta_2 trend_t + \varepsilon_t$$

obteniendo como resultado

$$\hat{Y}_t = 80.303 + 0.053 trend_t$$

(1.856) (0.028)

$$R^2 = 0.031$$

(entre paréntesis los errores estándar de los coeficientes)

Podemos representar el ajuste de este modelo junto a los datos muestrales como se muestra en la figura 3.

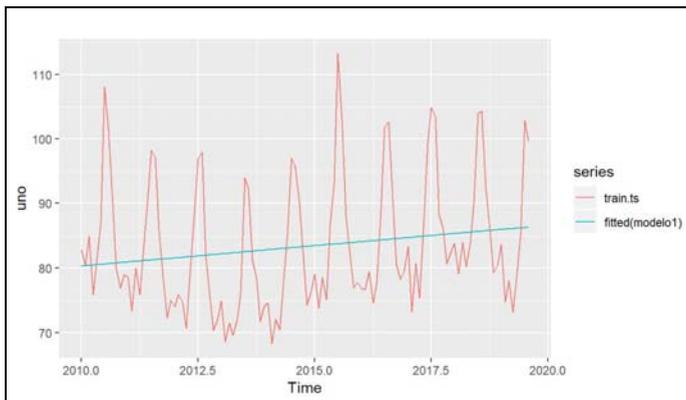


Figura 3. Datos muestrales y ajuste del modelo1,

### Modelo para el comportamiento estacional: *modelo2*

El modelo a estimar es:

$$Y_t = \beta_1 + \beta_2 \text{season}_t + \mathcal{E}_t$$

con el resultado:

```
#
# Call:
# tslm(formula = train.ts ~ season)
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -12.3464  -3.0134   0.3247   2.6897  11.1845
#
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# Intercept)      79.193      1.363  58.111 < 2e-16 ***
# season2         -4.756      1.927  -2.468  0.01523 *
# season3         -0.779      1.927  -0.404  0.68690
# season4         -5.106      1.927  -2.649  0.00932 **
# season5          1.700      1.927   0.882  0.37991
# season6          9.310      1.927   4.831  4.70e-06 ***
# season7         22.901      1.927  11.882 < 2e-16 ***
# season8         20.547      1.927  10.661 < 2e-16 ***
# season9          8.636      1.980   4.362  3.05e-05 ***
# season10         2.042      1.980   1.031  0.30473
# season11        -3.552      1.980  -1.794  0.07571 .
# season12        -1.798      1.980  -0.908  0.36604
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 4.309 on 104 degrees of freedom
# Multiple R-squared:  0.8335, Adjusted R-squared:  0.8159
# F-statistic: 47.33 on 11 and 104 DF, p-value: < 2.2e-16
```

Podemos también representar los datos muestrales junto al ajuste de este modelo como se muestra en la figura 4

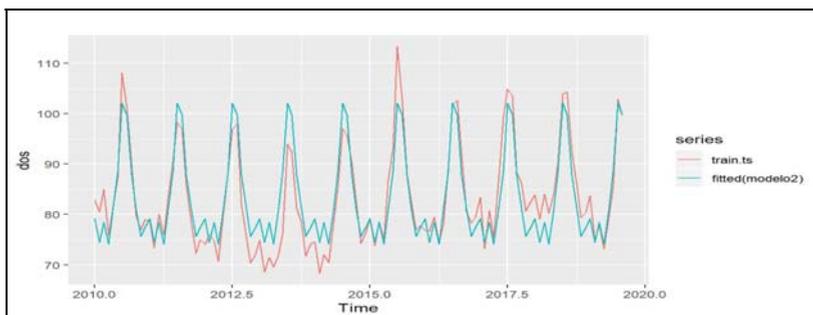


Figura 4. Datos muestrales y ajuste del modelo2.

### Modelo completo: *modelo3*

En este modelo incorporamos ambos componentes: la tendencia y el componente estacional:

$$Y_t = \beta_1 + \beta_2 trend_t + \beta_3 season_t + \mathcal{E}_t$$

obteniendo como resultado:

```
#
# Call:
# tslm(formula = train.ts ~ trend + season)
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -11.5937  -2.6815   0.0601   2.8104  10.9336
#
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  76.89309    1.42883   53.815 < 2e-16 ***
# trend         0.04181    0.01130    3.700 0.000348 ***
# season2      -4.79751    1.81949   -2.637 0.009666 **
# season3      -0.86263    1.81959   -0.474 0.636448
# season4      -5.23164    1.81977   -2.875 0.004911 **
# season5       1.53224    1.82002    0.842 0.401804
# season6       9.10143    1.82033    5.000 2.36e-06 ***
# season7      22.64971    1.82072   12.440 < 2e-16 ***
# season8      20.25460    1.82117   11.122 < 2e-16 ***
# season9       8.55280    1.86945    4.575 1.33e-05 ***
# season10      1.91688    1.86962    1.025 0.307634
# season11     -3.71960    1.86986   -1.989 0.049326 *
# season12     -2.00675    1.87017   -1.073 0.285763
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 4.068 on 103 degrees of freedom
# Multiple R-squared:  0.853, Adjusted R-squared:  0.8359
# F-statistic: 49.82 on 12 and 103 DF, p-value: < 2.2e-16
```

Los datos muestrales junto al ajuste de este modelo aparecen en la figura 5.

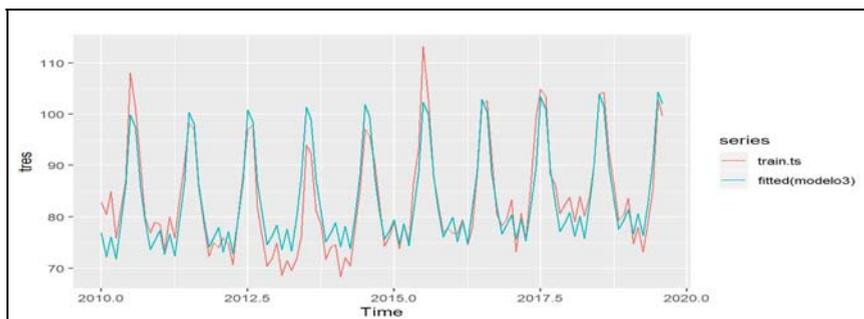


Figura 5. Datos muestrales y ajuste del modelo3

Llegados aquí, se impone comprobar la validez de este modelo3, el más completo. Analizando los residuos, tenemos, la figura 6.

Residuals from Linear regression model

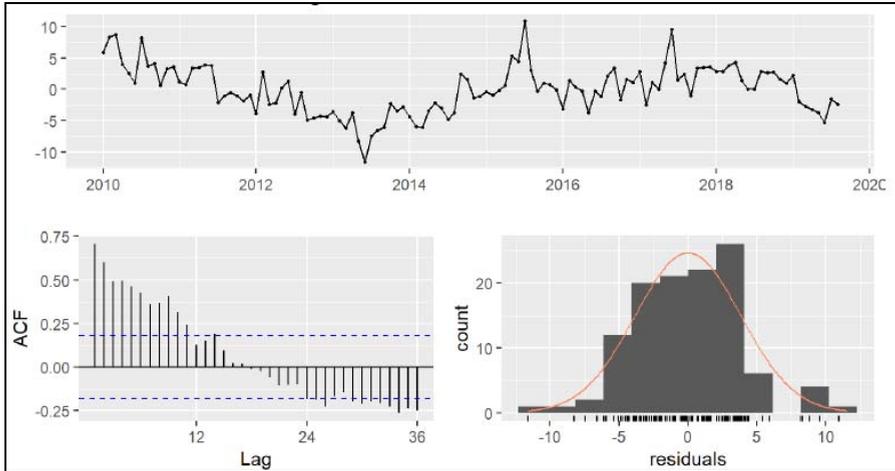


Figura 6. Residuos del modelo3.

```
#
# Breusch-Godfrey test for serial correlation of order up to 23
#
# data: Residuals from Linear regression model
# LM test = 73.696, df = 23, p-value = 3.226e-07
```

La alta autorrelación presente en estos residuos invalida, estadísticamente, el modelo estimado.

Para corregir este problema caben dos opciones. La primera consiste en modificar el modelo introduciendo retardos, *lags*, como regresores. La segunda consiste en modelizar los residuos, utilizando las facilidades que brinda el paquete *forecast*, acudiendo a la función *auto.arima*. Veamos ambas posibilidades.

## V. MODELOS VALIDADOS Y AJUSTES DE SUS PREDICIONES

### 1. Retardos como regresores de un modelo

Como guía para establecer el número de retardos a tener presente, estimemos la función de autocorrelación parcial de los residuos: figura 7.

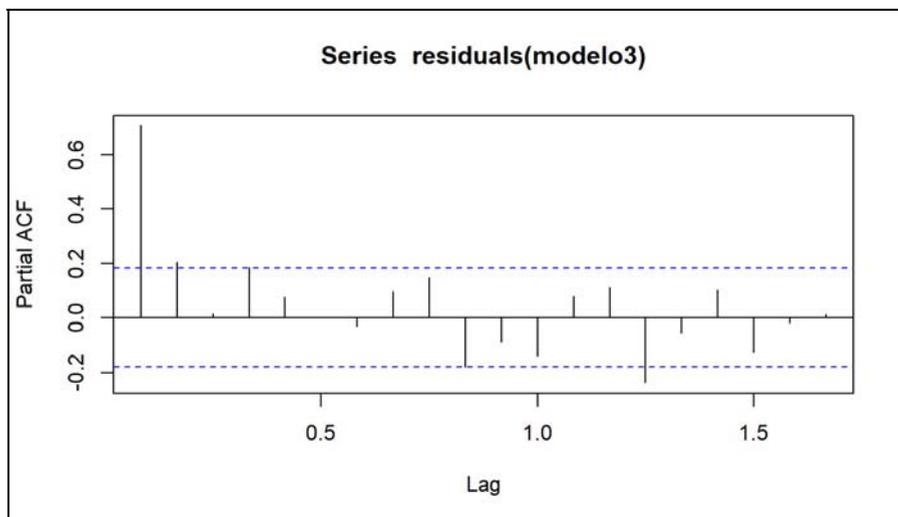


Figura 7. Función de autocorrelación parcial de los residuos del modelo3.

A la vista de este resultado, podemos considerar, inicialmente, un modelo  $AR(1)$  para los residuos, y postular un retardo:  $lag1$ , como posible regresor de un nuevo modelo.

Siguiendo las sugerencias de R. Krispin, y con ayuda de su paquete *TSstudio* formamos una tabla o *data.frame* con los datos de la serie *demanda*, en la que incorporamos la fecha de la serie, *date*, la serie, *y*, el retardo, *lag1* y los meses, *month*. Como resultado, tenemos

#	date	y	lag1	month
# 1	2010-01-01	82.833	NA	ene\\.
# 2	2010-02-01	80.522	82.833	feb\\.
# 3	2010-03-01	84.960	80.522	mar\\.
# 4	2010-04-01	75.876	84.960	abr\\.
# 5	2010-05-01	81.247	75.876	may\\.
# 6	2010-06-01	87.239	81.247	jun\\.

Hecha esta reorganización de la serie original, volvemos a realizar la partición, como antes, en dos subseries: *train* y *test*

Tras las particiones, estimamos el nuevo modelo, *mod1*, y comprobamos sus residuos, figura 8.

$$Y_t = \beta_1 + \beta_2 trend_t + \beta_3 season_t + \beta_4 lag1_t + \epsilon_t$$

Residuals from Linear regression model

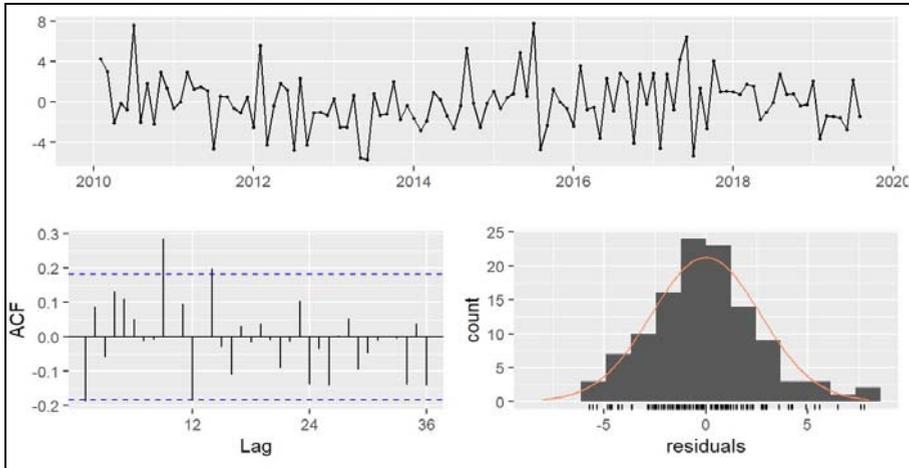


Figura 8: Residuos del modelo *mod1*.

```
#
# Breusch-Godfrey test for serial correlation of order up to 23
#
# data: Residuals from Linear regression model
# LM test = 35.91, df = 23, p-value = 0.04213
```

Ahora es posible aceptar que los residuos validan el modelo *mod1*

Validado el modelo, podemos pasar a la predicción puntual con el mismo, obteniendo, como resultado, la siguiente tabla, y su representación en la figura 9.

# Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
# Sep 2019	88.57244	84.66015	92.48472	82.55617	94.58871
# Oct 2019	80.75074	76.82594	84.67555	74.71523	86.78626
# Nov 2019	75.47439	71.55341	79.39537	69.44475	81.50403
# Dec 2019	77.50099	73.58295	81.41902	71.47588	83.52610
# Jan 2020	78.15379	74.22704	82.08053	72.11529	84.19229
# Feb 2020	75.38330	71.47928	79.28732	69.37975	81.38686
# Mar 2020	76.60482	72.66361	80.54602	70.54408	82.66556
# Apr 2020	62.70594	58.41677	66.99511	56.11010	69.30178
# May 2020	59.75136	54.82425	64.67847	52.17451	67.32821
# Jun 2020	68.31445	63.45941	73.16948	60.84843	75.78046
# Jul 2020	85.49090	80.88972	90.09209	78.41526	92.56655
# Aug 2020	89.56463	85.33635	93.79290	83.06244	96.06681

```
# Warning: `arrange_()` is deprecated as of dplyr 0.7.0.
# Please use `arrange()` instead.
# See vignette('programming') for more help
# This warning is displayed once every 8 hours.
# Call `lifecycle::last_warnings()` to see where this warning was generated.
```

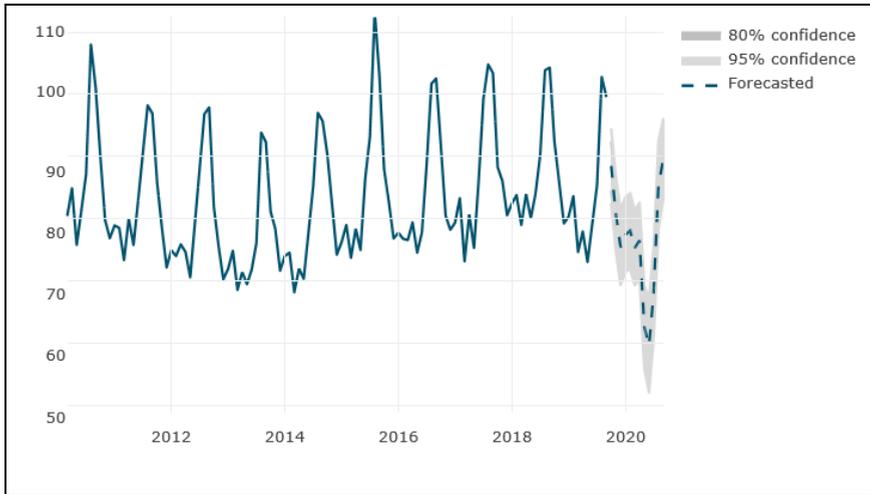


Figura 9. Predicciones con el modelo *mod1*.

Con ayuda de *TSstudio*, es posible representar, con mayor detalle, los datos de la serie completa, los datos de la predicción, junto a los datos de la submuestra *test*. El problema es que al imprimirse el gráfico en blanco y negro, el resultado queda poco realzado, véase la figura 10.

demanda - Actual vs Forecasted and Fitted

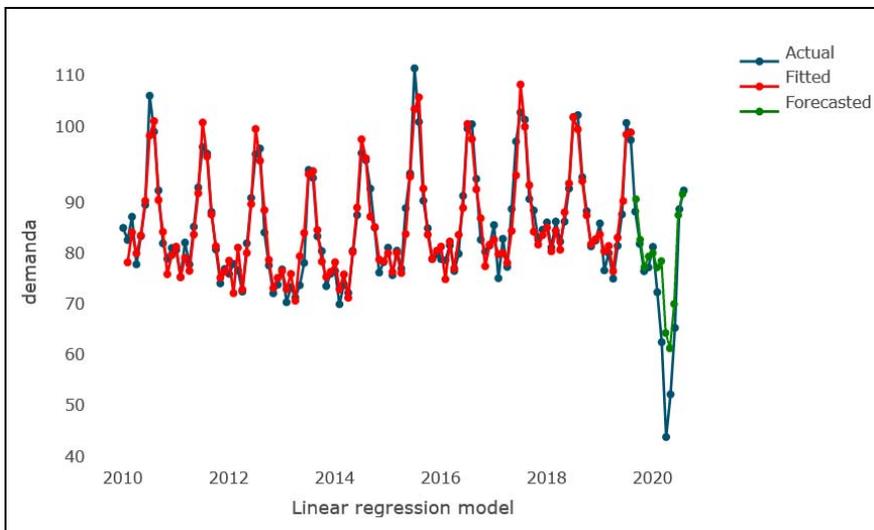


Figura 10. Datos muestrales, predicciones *fc1* y submuestra *test*.

## 2. Modelización de los residuos mediante auto.arima

El segundo enfoque para eliminar la alta autocorrelación presente en los residuos consiste en establecer un modelo para los mismos.

Hyndman y Athanasopoulos en el capítulo 9 de su *Forecasting, Principles and Practice*, segunda edición, 2018, al tratar los modelos de regresión dinámicos, en el apartado segundo, 9.2, considera el tema *Regression with ARIMA errors in R*, que permite estimar un modelo de regresión con residuos que, a su vez, son modelizados siguiendo un modelo ARIMA. Cf. p. 279.

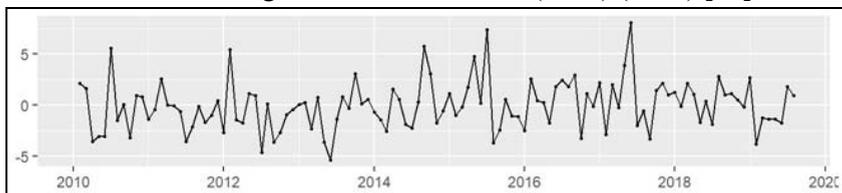
En esta ocasión, siguiendo a Rami Krispin, pp. 382 - 386, en su adaptación de lo expuesto por Hyndman y Athanasopoulos, se estima el modelo *mod2*.

Tras unos minutos, obtenemos el modelo estimado

```
# Series: train
# Regression with ARIMA(0,0,1)(2,0,0)[12] errors
#
# Coefficients:
#      ma1      sar1      sar2  intercept  monthfeb\\.\.  monthmar\\.\.
#      -0.4731  -0.3320  -0.2131    4.8666     -5.9225        2.473
# s.e.    0.1005   0.0998   0.0963    2.4873     0.9752        0.837
#      monthabr\\.\.  monthmay\\.\.  monthjun\\.\.  monthjul\\.\.  monthago\\.\.
#      -5.4061      5.4565      6.6104     12.4549     -2.3821
# s.e.    0.8335      0.8385      0.8411      0.9177      1.1391
#      monthsep\\.\.  monthoct\\.\.  monthnov\\.\.  monthdic\\.\.
#      -12.2481     -7.3520     -6.7876     0.3524     0.9540
# s.e.    1.1087      0.9187      0.8669      1.0114     0.0314
#
# sigma^2 estimated as 6.669:  log likelihood=-265.4
# AIC=564.79  AICc=571.1  BIC=611.45
#
# Training set error measures:
#      ME      RMSE      MAE      MPE      MAPE      MASE
# Training set -0.02318316  2.408081  1.849689 -0.1103042  2.187734  0.4772455
#      ACF1
# Training set 0.01587788
```

Los residuos de este modelo, han sido modelizados, de manera automática, como ARIMA (0,0,1) (2,0,0) [12], y una vez modelizados, los residuos del modelo *mod2* aparecen en la figura 11.

Residuals from Regression with ARIMA (0,0,1) (2,0,0) [12] errors.



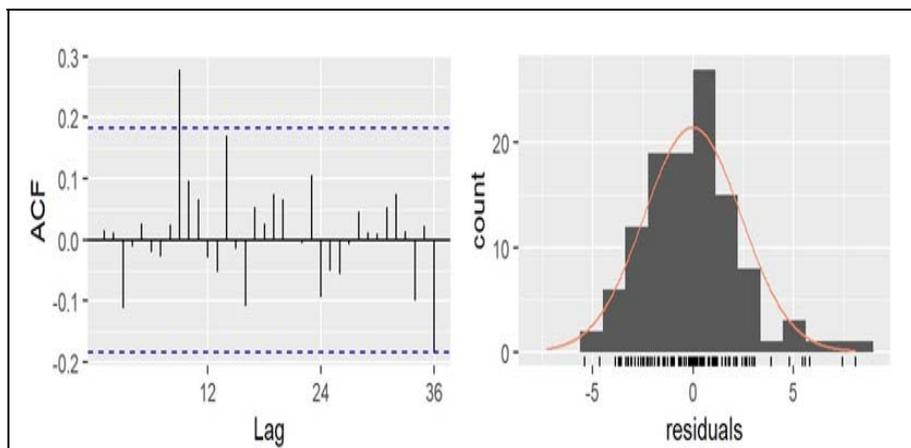


Figura 11. Residuos del modelo *mod2*.

```
#
#   Ljung-Box test
#
# data: Residuals from Regression with ARIMA(0,0,1)(2,0,0)[12] errors
# Q* = 22.974, df = 7, p-value = 0.001722
#
# Model df: 16. Total lags used: 23
```

Como puede observarse, los residuos validan el modelo *mod2*.

Validado el modelo, se pasa a la predicción con el mismo. Sus valores aparecen en la siguiente tabla.

# Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
# Sep 2019	87.71992	84.41046	91.02937	82.65855	92.78129
# Oct 2019	78.48729	74.82616	82.14841	72.88808	84.08649
# Nov 2019	74.41469	70.75356	78.07581	68.81548	80.01390
# Dec 2019	76.46612	72.80499	80.12724	70.86691	82.06533
# Jan 2020	75.93447	72.27334	79.59559	70.33526	81.53368
# Feb 2020	75.82794	72.16681	79.48906	70.22873	81.42714
# Mar 2020	74.59486	70.93373	78.25598	68.99565	80.19407
# Apr 2020	57.81758	54.15645	61.47870	52.21837	63.41679
# May 2020	52.11895	48.45782	55.78007	46.51974	57.71816
# Jun 2020	61.09431	57.43318	64.75543	55.49510	66.69351
# Jul 2020	76.71646	73.05533	80.37758	71.11725	82.31567
# Aug 2020	84.68098	81.01986	88.34211	79.08177	90.28019

La representación gráfica se recoge en la figura 12.

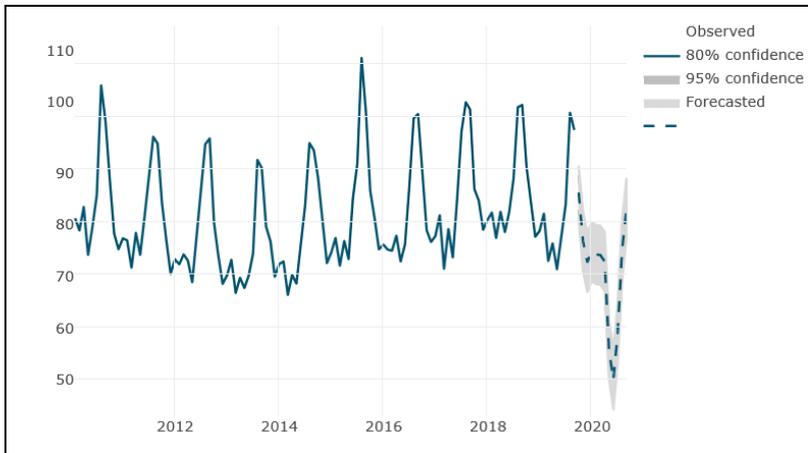


Figura 12. Predicción con el modelo mod2.

### 3. Ajustes de sus predicciones

Podemos comparar las predicciones de ambos modelos acudiendo a los estadísticos de ajuste. Para el *mod1* tenemos,

$$accuracy(fc1, test)$$

que da como resultado

#	ME	RMSE	MAE	MPE	MAPE
# Training set	1.112325e-15	2.657333	2.054429	-0.09860793	2.436091
# Test set	-4.803312e+00	7.998320	5.202259	-8.99698601	9.473989
#	MASE	ACF1	Theil's U		
# Training set	0.5305327	-0.1881583	NA		
# Test set	1.3434240	0.6075475	0.7636603		

Para el *mod2*,

$$accuracy(fc2, test)$$

obteniendo

#	ME	RMSE	MAE	MPE	MAPE	MASE
# Training set	-0.02318316	2.408081	1.849689	-0.1103042	2.187734	0.4772455
# Test set	-1.27046198	6.953342	5.007931	-3.5538501	8.104801	1.2921159
#	ACF1	Theil's U				
# Training set	0.01587788	NA				
# Test set	0.56005863	0.6122006				

Para la explicaciones de todos estos estadísticos, ver Hyndman y Athanasopoulos, *Forecasting: Principles and Practice*, segunda edición, pp. 62 - 68.

Tomado en consideración el estadístico MAPE (Mean absolute percentage error), podemos ver que el porcentaje de error del primer modelo es del 2.4% en la submuestra *train* y del 9.5% en la submuestra *test*. Y para el segundo modelo, el porcentaje de error es del 2.2% en la submuestra *train* y del 8.1% en la submuestra *test*. Las diferencias son prácticamente insignificantes, quedando así aceptados ambos modelos.

## VI. UN MODELO ARIMA

Como complemento de todo lo expuesto, y con ayuda de la función *auto.arima* del paquete *forecast* podemos estimar un modelo de series temporales para nuestra serie *demanda*.

Una vez particionada la serie en las submestras *train.ts* y *test.ts*, como anteriormente, estimamos el modelo mediante la instrucción.

$$mod3 = \text{auto.arima}(\text{train.ts})$$

obteniendo

```
# Series: train.ts
# ARIMA(0,1,1)(2,1,0)[12]
#
# Coefficients:
#      ma1      sar1      sar2
#    -0.5861 -0.7625 -0.4131
# s.e.  0.0957  0.0954  0.0948
#
# sigma^2 estimated as 10.5: log likelihood=-270.26
# AIC=548.52  AICc=548.93  BIC=559.06
#
# Training set error measures:
#      ME      RMSE      MAE      MPE      MAPE      MASE
# Training set 0.1127655 3.009196 2.307399 0.08174433 2.731622 0.5953411
#
#      ACF1
# Training set 0.05269271
```

Los residuos de este modelo aparecen en la figura 13.

Residuals from ARIMA (0,1,1)(2,1,0)[12]

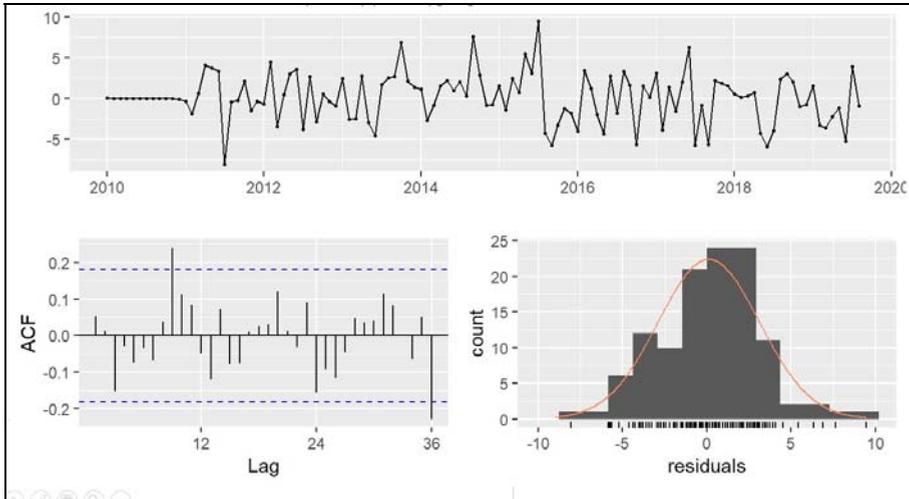


Figura 13. Residuos del modelo mod3.

```
#
# Ljung-Box test
#
# data: Residuals from ARIMA(0,1,1)(2,1,0)[12]
# Q* = 22.993, df = 20, p-value = 0.2891
#
# Model df: 3. Total lags used: 23
```

Estos residuos validan el modelo. Validado el modelo, se pasa a utilizarlo para predicción. Los valores aparecen en la siguiente tabla, y su representación gráfica en la figura 14.

# Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
# Sep 2019	87.74220	83.58869	91.89571	81.38995	94.09445
# Oct 2019	80.64205	76.14677	85.13734	73.76711	87.51700
# Nov 2019	76.17640	71.36355	80.98925	68.81579	83.53702
# Dec 2019	77.55218	72.44146	82.66290	69.73601	85.36835
# Jan 2020	80.44323	75.05108	85.83539	72.19664	88.68982
# Feb 2020	2.45185	66.79223	78.11146	63.79621	81.10748
# Mar 2020	78.06989	72.15490	83.98489	69.02369	87.11610
# Apr 2020	73.35945	67.19965	79.51924	63.93885	82.78004
# May 2020	80.74522	74.34998	87.14045	70.96455	90.52588
# Jun 2020	89.70237	83.08007	96.32467	79.57443	99.83031
# Jul 2020	100.87338	94.03153	107.71522	90.40968	111.33707
# Aug 2020	99.68618	92.63162	106.74073	88.89717	110.47519

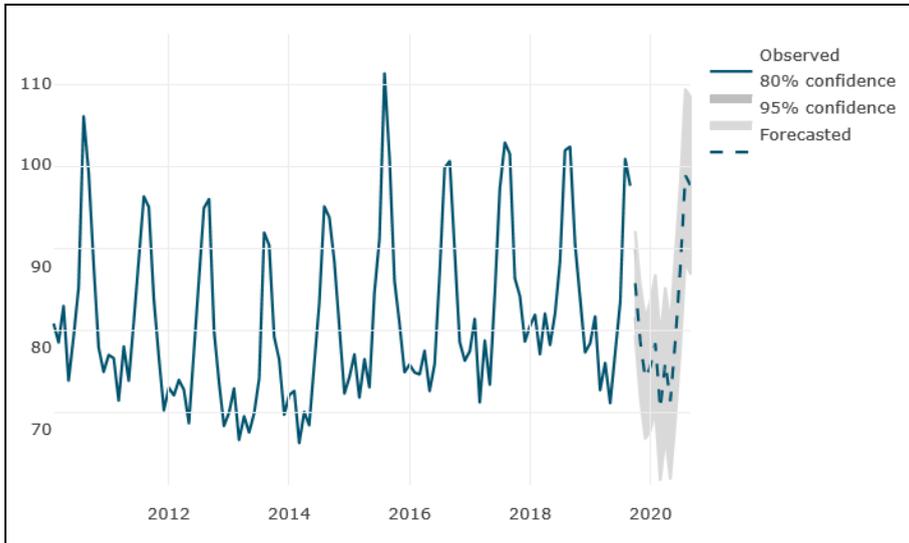


Figura 14. Predicción con el modelo mod3.

Como último paso, veamos los estadísticos del ajuste.

#	ME	RMSE	MAE	MPE	MAPE
# Training set	0.1127655	3.009196	2.307399	0.08174433	2.731622
# Test set	-11.4013658	16.041550	11.401366	-19.79214092	19.792141
#	MASE	ACF1	Theil's U		
# Training set	0.5953411	0.05269271	NA		
# Test set	2.9417113	0.77880878	1.764728		

Si acudimos de nuevo al estadístico *MAPE*, el porcentaje de error en la submuestra *train* es del 2.7%, y del 19.8% en la submuestra *test*, porcentajes que podemos *considerar* parecidos a los de los modelos de regresión *mod1* y *mod2*. Así, pues, estos dos modelos de regresión logran predicciones comparables a las del modelo ARIMA.

## VII. CONCLUSIONES

A la vista de estos resultados parece correcto afirmar que el modelo de regresión, tan utilizado a lo largo de la historia, puede ser también utilizado para la predicción de las series temporales.

**VIII. BIBLIOGRAFÍA**

- BOX, G. E. P. y JENKINS, G. M., *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day, 1976.
- BOX, G. E. P.; JENKINS, G. M. y REINSEL, G. C., *Time Series Analysis: Forecasting and Control*, tercera edición. Englewood Cliffs: Prentice-Hall, 1994.
- CRAN, The Comprehensive R Archive Network. <http://cloud.r-project.org/> (<http://cloud.r-project.org/>)
- HYNDMAN, R. J. y colaboradores. *fpp2*, programa descargable desde CRAN.
- HYNDMAN, R. J. y ATHANASOPOULOS, G., *Forecasting, Principles and Practice*, segunda edición. O.Texts. [www.otexts.org/fpp2/](http://www.otexts.org/fpp2/), 2018.
- HYNDMAN R. J. y KHANDAKAR, Y., Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*, 27(2008)1-22
- KRISPIN, R., *Hands-On Time Series Analysis with R.*, Birmingham: Packt Publishing, 2019.
- KRISPIN, R., *TSstudio*, programa descargable desde CRAN.
- Red Eléctrica de España, [www.ree.es/es/data/publicaciones/indice-red-eléctrica](http://www.ree.es/es/data/publicaciones/indice-red-eléctrica).

