

Un modelo *state space* para la producción de energía en España

A model state space for the monthly Spanish energy production

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

Resumen: En este trabajo se estima un modelo *state space* para la serie de observaciones mensuales que recoge la producción y distribución de la energía total en España, desde enero de 2013 hasta enero de 2021. Tras una breve descripción de estos modelos, se pasa a su aplicación, y para destacar la valía de los mismos se comparan sus predicciones con las de otros modelos utilizados con series temporales.

Abstract: In this paper a model *state space* is estimated for the *Monthly Spanish total production and distribution of energy*, for the period January 2013 to January 2021. After a brief presentation of these models *state space*, the estimated model is used for prediction. To enhance the utility of these models, its predictions are compared with those generated with other models used in Time Series Analysis.

Palabras clave: Modelos *state space*, Filtro de Kalman, estado filtrado, estado suavizado, ARIMA, tslm, ets, paquete KFAS, paquete forecast, precisión de las predicciones.

Keywords: State space models, Kalman Filter, filtered state, smoothed state, ARIMA, tslm, ets, KFAS statistical package, forecast, statistical package, accuracy of forecasts

Sumario:

- I. **Introducción.**
- II. **Modelos *state space*.**

III. El total de la energía en España.

IV. Estimación del modelo *state space*, paquete KFAS.

V. Comparaciones.

5.1. *Modelo ARIMA.*

5.2. *Modelo de regresión.*

5.3. *Estimación del modelo ets, paquete forecast.*

VI. Conclusiones.

VII. Bibliografía.

Recibido: septiembre 2021.

Aceptado: noviembre 2021.

I. INTRODUCCIÓN

En este trabajo se estima un modelo *state space* para las observaciones mensuales de la serie temporal sobre la producción y distribución de la energía total en España, desde enero de 2013 hasta enero de 2021.

Para mejor evaluar la precisión de las predicciones con el modelo estimado, la serie original es particionada en dos subseries, la serie *train.ts* sobre la que se estima el modelo, y la serie *test.ts* con la que se compararán las predicciones.

Así mismo, para poner de relieve la valía del modelo *state space* estimado, se comparan sus predicciones con las que se obtienen con otros modelos utilizados para series temporales. En concreto, con un modelo ARIMA y con un modelo de regresión.

A pesar de ser los modelos *state space* un marco de referencia para los modelos de series temporales, hay que decir que hasta sólo recientemente los modelos *state space* no han estado a disposición de los investigadores en el ámbito del software libre, y concretamente, en el entorno del proyecto R. En este entorno, existen actualmente varios paquetes para la estimación de modelos *state space*. Un ejemplo notable, es el paquete *KFAS*, que es el utilizado en este trabajo.

Hay que destacar también que con un enfoque ligeramente diferente al del paquete *KFAS*, el Prof. Robert Hyndman y colaboradores tratan el modelo *state space* en su paquete *forecast*. En el último apartado de este trabajo, se estima el modelo *state space* con este paquete, y como se puede comprobar, la estimación *suavizada* o *alisada* del vector de estados prácticamente coincide con la del paquete *KFAS*.

II. MODELOS *STATE SPACE*

Los modelos *state space* constituyen un marco de referencia para un gran número de modelos utilizados con las series temporales.

La formulación más simple de un modelo *state space* del tipo Gauss puede expresarse

$$y_t = Z_t \alpha_t + \epsilon_t \quad (\text{ecuación de observaciones}) \quad (1)$$

$$\alpha_{t+1} = T_t \alpha_t + R_t \eta_t \quad (\text{ecuación de estados}) \quad (2)$$

Siendo

$$\epsilon_t \sim N(0, H_t)$$

$$\eta_t \sim N(0, Q_t)$$

con distribución normal, e independientes entre sí.

La ecuación (1) describe la relación entre el vector no observado de estados, α_t , y el vector de observaciones y_t . La ecuación (2) describe la evolución en el tiempo del vector de estados.

El vector y_t de dimensiones $p \times 1$ contiene las observaciones de la serie temporal, mientras que α_{t+1} de dimensiones $m \times 1$ contiene el vector de valores *latentes* o no observados de los *estados*.

Las ecuaciones (1) y (2) junto con las distribuciones de los errores, constituyen un modelo *state space* correctamente especificado.

Las matrices Z_t, T_t y R_t dependen del tipo de modelo, y las matrices de varianzas y covarianzas H_t y Q_t suelen ser invariantes en el tiempo.

El objetivo de la modelización *state space* consiste en lograr el conocimiento de los estados *latentes* α_t , dadas las observaciones recogidas en y_t . El objetivo se logra, en la actualidad, gracias a los algoritmos recurrentes de Kalman: *filtering* y *smoothing*. Del primero obtenemos las predicciones *un paso adelante* y los errores de la predicción, es decir:

$$\alpha_{t+1} = E(\alpha_{t+1} | y_t, \dots, y_1)$$

$$v_t = y_t - Z_t \alpha_t$$

así como las matrices de varianzas y covarianzas asociadas.

Sobre estos resultados, y con el algoritmo *smoothing* obtenemos los valores ajustados, *suavizados* o *alisados* de α_t

$$\hat{\alpha}_t = E(\alpha_t | y_n, \dots, y_1)$$

$$V_t = E(\text{var}(\alpha_t | y_n, \dots, y_1))$$

Con ayuda del paquete *KFAS* en R, del profesor Jouni Helske, de la Universidad de Jyväskylä, Finlandia, realizamos este trabajo. Hay que señalar que *KFAS* sigue muy de cerca la notación de Durbin y Koopman, los autores de referencia en los modelos *state space*.

Llegados aquí, y a modo de paréntesis, como ya hice en mi anterior trabajo, creo interesante relacionar lo expuesto con el concepto básico de *serie temporal*.

Una serie temporal es una serie de observaciones ordenadas en el tiempo.

El modelo básico para representar una serie de tiempo es

$$y_t = \mu_t + \gamma_t + \epsilon_t; \quad t = 1, \dots, n \quad (3)$$

significando los símbolos

μ_t : el componente tendencial, el *trend*

γ_t : el componente periódico, el componente *estacional*

ϵ_t : el término de error o perturbación del modelo.

En economía se utiliza también la versión multiplicativa del modelo:

$$y_t = \mu_t \gamma_t \epsilon_t$$

y, tomando logaritmos se pasa a la formulación aditiva en (3)

Para establecer modelos para μ_t y γ_t se necesita el concepto de *random walk*, que designamos como α_t y que se expresa como

$$\alpha_{t+1} = \alpha_t + \eta_t; \quad \eta_t \sim iii(0, \sigma_\eta^2)$$

Si en (3) sustituimos μ_t por α_t , olvidamos el componente γ_t , y consideramos que α_t tiene distribución normal, llegamos a

$$y_t = \alpha_t + \epsilon_t; \quad \epsilon_t \sim N(0, \sigma_\epsilon^2) \quad (4)$$

$$\begin{aligned} \alpha_{t+1} &= \alpha_t + \eta_t; \quad \eta_t \sim N(0, \sigma_\eta^2) \quad (5) \\ t &= 1, 2, \dots, n \end{aligned}$$

que constituye la formulación más sencilla de un modelo *state space*, con ϵ_t y η_t incorrelacionadas serialmente e independientes entre sí. La representación en (4) y (5) recibe el nombre técnico de *modelo de nivel local*.

En la formulación del paquete *forecast* antes aludido, las anteriores ecuaciones (4) y (5) pasan a escribirse como

$$y_t = \alpha_t + \epsilon_t \quad (6)$$

$$\alpha_{t+1} = \alpha_t + a\epsilon_t \quad (7)$$

con un único término de error ϵ_t . Esta formulación se denomina el modelo *state space* de *innovación*, siendo a el factor de alisado o suavizado de la ecuación de estados.

III. EL TOTAL DE LA ENERGÍA DE ESPAÑA

En este artículo se establece un modelo *state space* para la serie: *Production: Energy: Production and distribution of electricity, gas, steam and air conditioning: Total for Spain*, datos de la OCDE, tomados del Federal Reserve Bank of Saint Louis, con fecha: 8-05-2021.

Esta serie *Indice*, de datos mensuales, ajustada estacionalmente y con el índice 100, en 2015 tiene un total de 553 observaciones: desde enero de 1975 hasta enero de 2021. Para facilitar las representaciones gráficas, en este trabajo se toman en consideración las observaciones desde enero del 2013, hasta enero de 2021: un total de 97 observaciones.

Por tratarse de una serie de observaciones mensuales, la serie se particiona en dos subseries, la subserie *train.ts* con la que se estima el modelo, y la subserie *test.ts* con las 12 últimas observaciones de la serie original, con cuyos valores se compararán las predicciones que se hagan

En la figura 1, podemos observar la serie completa y la subserie *train.ts*.

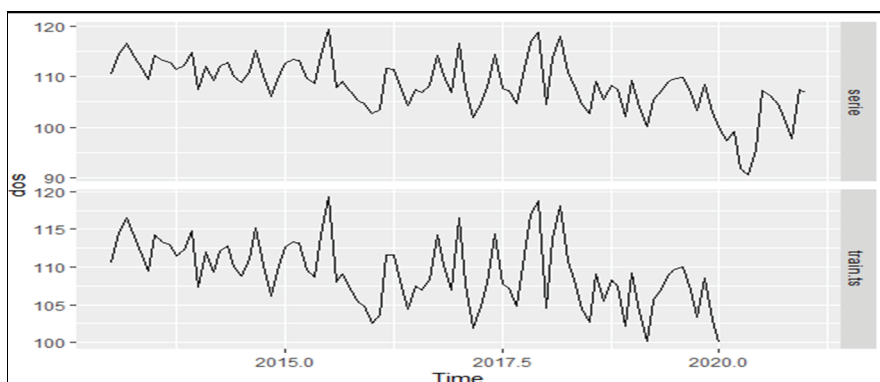


Figura 1. Serie original y subserie *train.ts*

Dado el aspecto de la serie *train.ts*, estimamos un modelo teniendo en cuenta el *nivel* como el *estado* de nuestro modelo.

IV. ESTIMACIÓN DEL MODELO STATE SPACE, PAQUETE KFAS

Con ayuda del paquete *KFAS* establecemos el modelo como

```
modelo = SSModel(train.ts ~ SSMtrend(1,Q = list(matrix(NA))),
                 H = matrix(NA))
```

Los valores *NA* representan las varianzas σ_ϵ^2 y σ_η^2 , que serán estimadas con la función *fitSSM*

Establecido el modelo, estimamos sus coeficientes mediante

```
fit = fitSSM(modelo, inits = c(0,0), method = "BFGS")
```

En nuestro caso, las estimaciones de *máxima verosimilitud* de las varianzas son: $\sigma_\epsilon^2 = 0.61$ y $\sigma_\eta^2 = 12.71$

Aplicamos los algoritmos *filtering* and *smoothing*, que en nuestro caso podemos simplificar como

```
out = KFS(fit$model, smoothing = "state")
```

En la figura 2 representamos la serie *train.ts* y la estimación alisada del vector de estados α_t

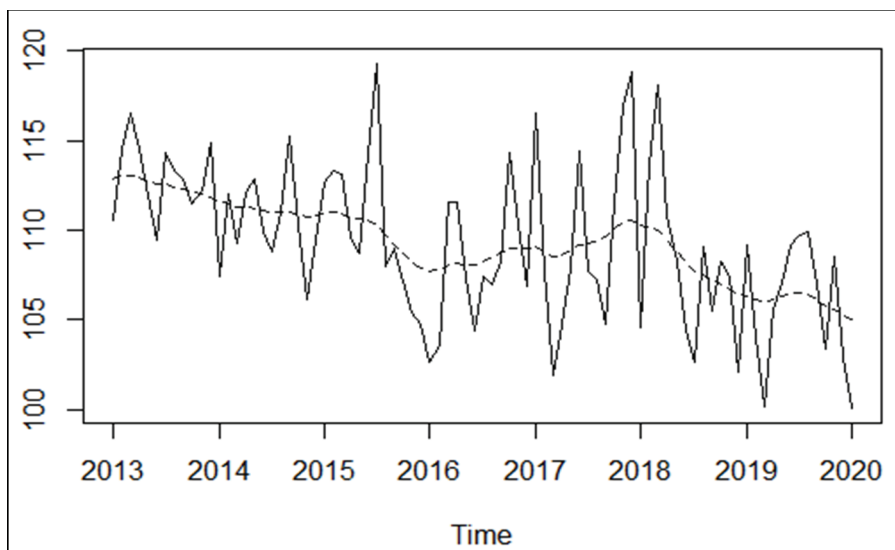


Figura 2. Series: train.ts y vector de estados suavizado

Como se puede comprobar, el vector de estados suavizados, α_t sigue de cerca a la serie *train.ts*.

Como primer paso para la comprobar la validez estadística del modelo estimado, examinemos sus residuos. Dado que el algoritmo de Kalman es un algoritmo recurrente, varios son los tipos de residuos generados.

La instrucción:

```
plot(fit$model)
```

permite ver los tres tipos de residuos generados. Como es una representación gráfica *interactiva*, que requiere el activar los distintos pasos, aquí nos limitamos a considerar los *residuos recurrentes*

```
rres = rstandard(out)
```

residuos que aparecen representados en la figura 3, junto con la representación de la correspondiente función de autocorrelación.

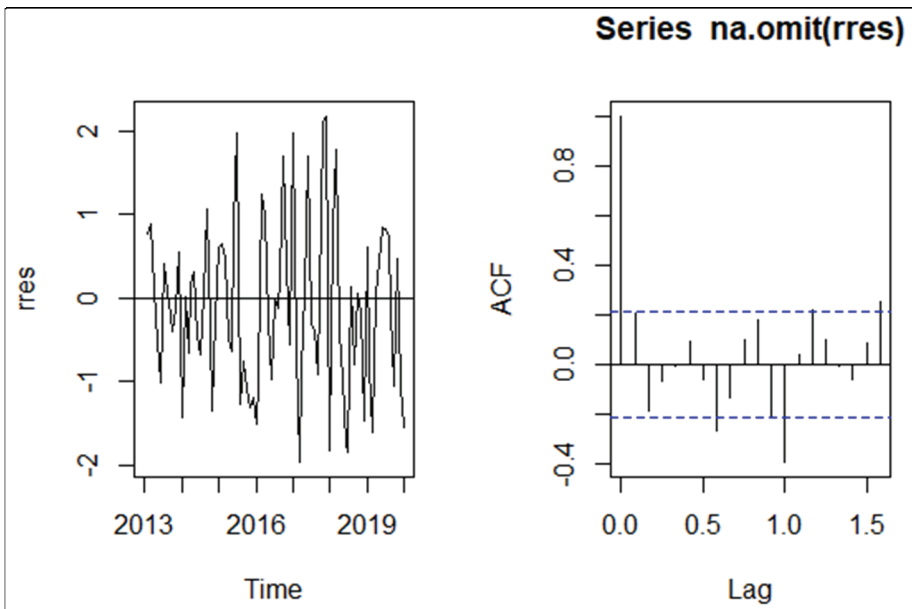


Figura 3. Residuos recurrentes y acf. de los mismos

La figura 3 muestra unos residuos que validan el modelo estimado.

Pasemos a la fase de predicción.

Dado que en la subserie *test.ts* hay 12 observaciones, hagamos con nuestro modelo la predicción de 12 valores futuros. Además de las predicciones, obtenemos las bandas de confianza del 90%

```
Pred = predict(out$model, n.ahead = 12, interval = "prediction",
              level = 0.9)
```

Las predicciones aparecen en la siguiente tabla, junto con los valores de la banda de confianza. Y en la figura 4, vienen representados los valores de la serie *train.ts* junto con las predicciones y la banda de confianza del 90%

##		fit	lwr	upr
##	Feb 2020	105.0094	98.46562	111.5532
##	Mar 2020	105.0094	98.34019	111.6786
##	Apr 2020	105.0094	98.21708	111.8017
##	May 2020	105.0094	98.09617	111.9226
##	Jun 2020	105.0094	97.97733	112.0415
##	Jul 2020	105.0094	97.86047	112.1583
##	Aug 2020	105.0094	97.74549	112.2733
##	Sep 2020	105.0094	97.63229	112.3865
##	Oct 2020	105.0094	97.52082	112.4980
##	Nov 2020	105.0094	97.41097	112.6078
##	Dec 2020	105.0094	97.30269	112.7161
##	Jan 2021	105.0094	97.19591	112.8229

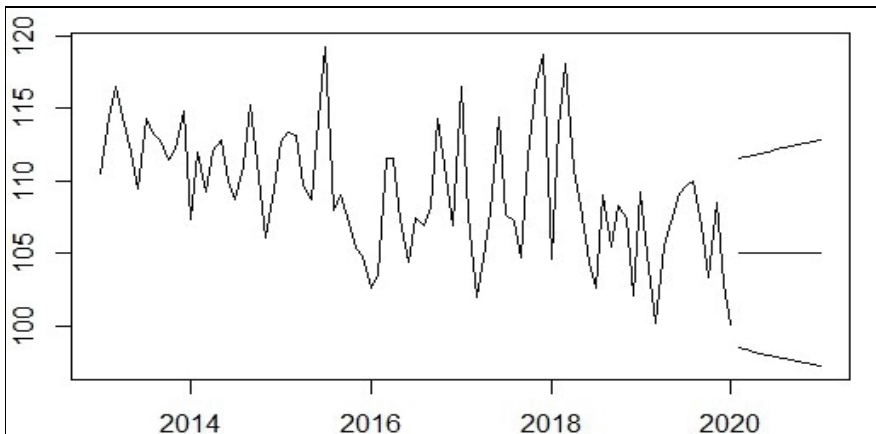


Figura 4. Subserie *train.ts*, y predicciones

La predicción muestra unos valores en línea recta, es decir constantes.

Para comprobar la precisión de las predicciones, comparémoslas con los valores de la serie *test.ts*. Para ello acudimos a la función *accuracy* del paquete *forecast*, que genera los estadísticos

```
accuracy(pred, test.ts)
```

```
##           ME      RMSE      MAE      MPE      MAPE      ACF1 Theil's U
## Test set -4.452109  7.324739  5.743247 -4.78534  5.991692  0.5169549  1.331814
```

Entre estos estadísticos para establecer la precisión de las predicciones, aparece el *MAPE*: *Mean absolute percentage error*, que es uno de los más utilizados, por ser un porcentaje, independiente de la escala de medición de las observaciones y por la facilidad de su interpretación. Aunque tiene algunos inconvenientes, no obstante, se utiliza para comparar las predicciones entre distintos modelos. Hyndeman, R. y Athanasopoulos, G., en su *Forecasting, Principles and Practice*, segunda edición, dedican las páginas 62 a 71 para exponer el significado y las propiedades de los estadísticos arriba calculados, los más utilizados al respecto.

En nuestro caso, un error del 5.99% podemos considerarlo como un gran ajuste de las predicciones de nuestro modelo. Es decir, la diferencia entre las predicciones de nuestro modelo y las observaciones reales es del 5.99%.

V. COMPARACIONES

En vistas a mejor valorar el modelo *state space* estimado, vamos a comparar sus predicciones con las de otros dos modelos utilizados con series temporales: un modelo *ARIMA*, y un modelo de regresión.

5.1. Modelo *ARIMA*

El ya mencionado paquete *forecast* brinda la función *auto.arima* que de manera automática determina los valores de los distintos elementos de un modelo *ARIMA*, tomando como criterio de selección el estadístico de Akaike: *AICc*.

```
## Series: train.ts
## ARIMA(0,1,2)(0,0,1)[12]
##
## Coefficients:
##          ma1          ma2          sma1
##      -0.5333   -0.2856   -0.4304
## s.e.   0.1178    0.1204    0.1448
##
## sigma^2 estimated as 12.64: log likelihood=-226
## AIC=459.99  AICc=460.5  BIC=469.72
```

Si pasamos a los residuos de este modelo, tenemos la figura 5.

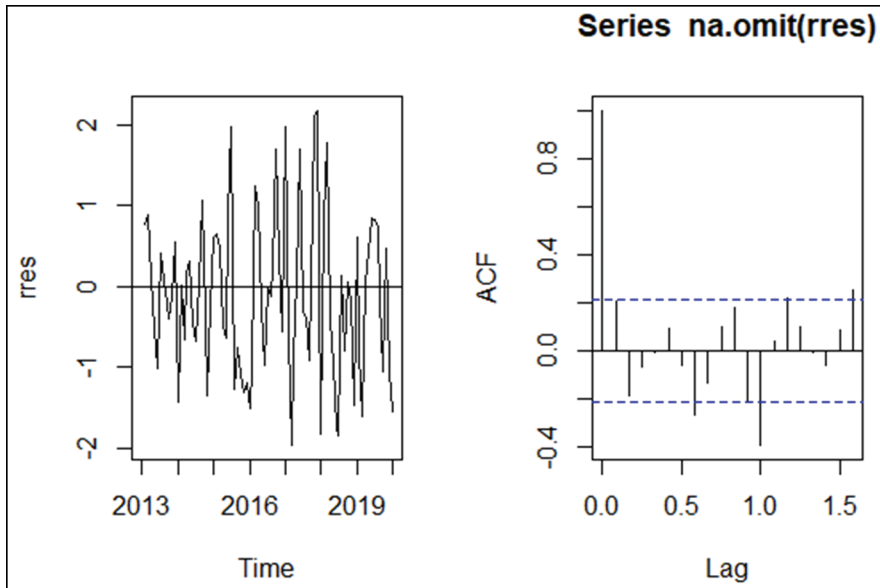


Figura 5. residuos del modelo ARIMA, y acf de los mismos

De nuevo, los residuos validan el modelo estimado.

Pasemos a las predicciones.

En la siguiente tabla aparecen los valores de las predicciones junto a las bandas del 80% y 95% de confianza. Y en la figura 6, su representación.

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## Feb 2020	105.4487	100.89318	110.0043	98.48162	112.4158
## Mar 2020	107.6898	102.66262	112.7169	100.00140	115.3782
## Apr 2020	106.5395	101.44513	111.6339	98.74832	114.3307
## May 2020	106.2667	101.10599	111.4275	98.37406	114.1594
## Jun 2020	105.6272	100.40091	110.8534	97.63431	113.6200
## Jul 2020	106.0694	100.77841	111.3603	97.97756	114.1612
## Aug 2020	104.9837	99.62884	110.3386	96.79415	113.1732
## Sep 2020	107.0401	101.62211	112.4581	98.75399	115.3263
## Oct 2020	107.3864	101.90591	112.8668	99.00474	115.7680
## Nov 2020	105.0842	99.54203	110.6264	96.60818	113.5602
## Dec 2020	108.4379	102.83471	114.0412	99.86854	117.0073
## Jan 2021	109.1690	103.50534	114.8326	100.50720	117.8307

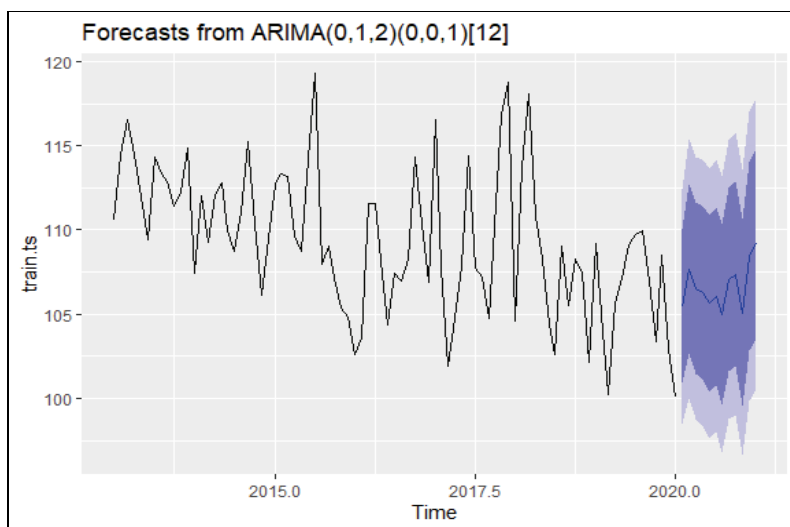


Figura 6. Serie *train.ts* y predicciones del modelo ARIMA

En la figura 6 aparecen la serie *train.ts*, las predicciones del modelo, y las bandas de confianza.

Pasemos a comparar las predicciones con las observaciones de *test.ts*

```
##           ME           RMSE           MAE           MPE           MAPE           ACF1           Theil's U
## Test set -6.087924  8.177472  6.504054  -6.389754  6.779615  0.6026834  1.488421
```

Acudiendo de nuevo al estadístico *MAPE*: 6.7796. En esta ocasión, el porcentaje de error que se comete es del 6.8%. El modelo *state space* lo hace mejor.

5.2. Modelo de regresión

En el paquete *fpp2* del profesor Robert Hyndman colaboradores se encuentra la función *tslm* que ajusta el modelo de regresión a series temporales.

```
##
## Call:
## tslm(formula = train.ts ~ trend)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.1989 -2.4848 -0.5331  2.0761 10.8176
##
## Coefficients:
```

```
##          Estimate Std. Error t value Pr(>|t|)
## (Intercept) 112.77387    0.84272 133.821 < 2e-16 ***
## trend      -0.08012     0.01702  -4.707 9.95e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.851 on 83 degrees of freedom
## Multiple R-squared:  0.2107, Adjusted R-squared:  0.2012
## F-statistic: 22.15 on 1 and 83 DF,  p-value: 9.949e-06
```

Los residuos de este modelo, aparecen en la figura 7.

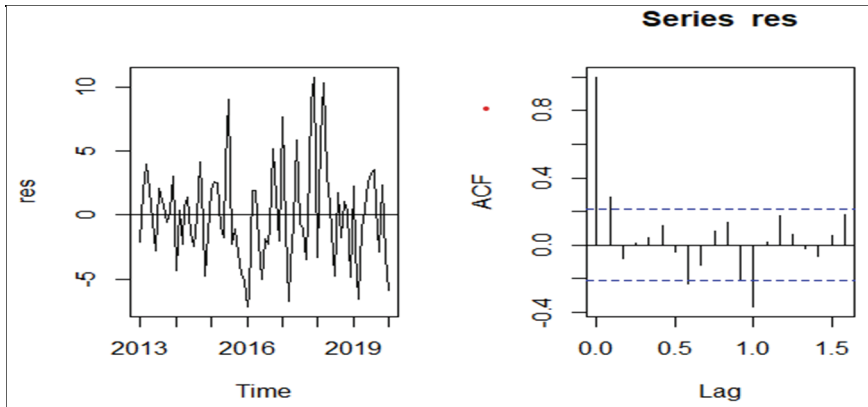


Figura 7. Residuos del modelo de regresión y acf de los mismos

Considerando que la ligera autocorrelación de los residuos no invalida el modelo estimado, pasemos a la predicción con el mismo.

En la tabla siguientes, aparecen las predicciones, y en la figura 8, su representación.

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## Feb 2020	105.8835	100.79154	110.9755	98.04370	113.7234
## Mar 2020	105.8034	100.70729	110.8995	97.95722	113.6496
## Apr 2020	105.7233	100.62295	110.8236	97.87060	113.5760
## May 2020	105.6432	100.53852	110.7478	97.78384	113.5025
## Jun 2020	105.5630	100.45399	110.6721	97.69695	113.4291
## Jul 2020	105.4829	100.36938	110.5965	97.60991	113.3559
## Aug 2020	105.4028	100.28467	110.5209	97.52273	113.2829
## Sep 2020	105.3227	100.19988	110.4455	97.43541	113.2100
## Oct 2020	105.2426	100.11500	110.3701	97.34795	113.1372
## Nov 2020	105.1624	100.03002	110.2949	97.26036	113.0645
## Dec 2020	105.0823	99.94496	110.2197	97.17263	112.9920
## Jan 2021	105.0022	99.85980	110.1446	97.08476	112.9197

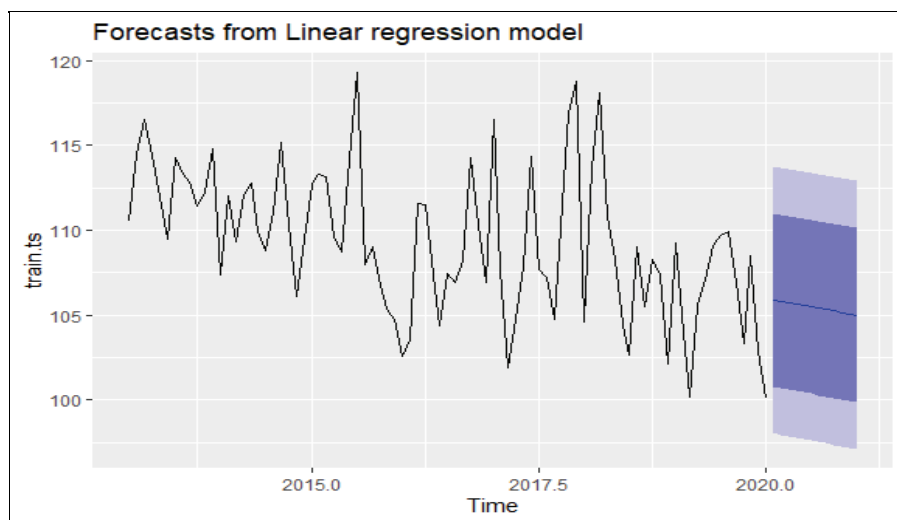


Figura 8. Subserie train.ts, y predicciones

Veamos la precisión de las predicciones.

##	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
## Test set	-4.885577	7.733354	6.02127	-5.228066	6.288949	0.5333002	1.398282

Para este caso, el valor de *MAPE* es 6.2889.

5.3. Modelo ets del paquete forecast

La función *ets* permite la estimación automática de un modelo *state space* para series temporales, teniendo como criterio de selección la minimización del *AICc*.

El modelo estimado, llamémoslo *fit2*

```
fit2 = ets(train.ts)
```

genera el resultado siguiente

```
## ETS(A,N,N)
##
## Call:
## ets(y = train.ts)
##
## Smoothing parameters:
```

```
##      alpha = 0.1737
##
##      Initial states:
##      l = 112.8125
##
##      sigma: 4.0009
##
##      AIC      AICc      BIC
## 617.3110 617.6073 624.6390
```

Los residuos de este modelo aparecen en la figura 9

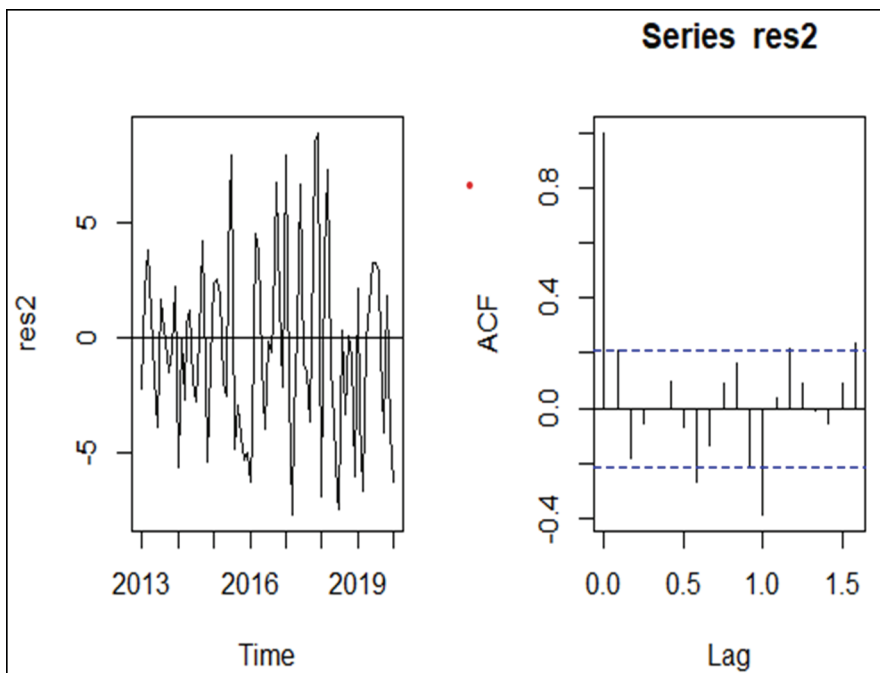


Figura 9.- Residuos del modelo ets y función acf de los mismos.

Estos residuos también validan el modelo *fit2*.

La función *ets* genera como resultado la serie *states*, que es la serie *suavizada* del vector de estados. Si representamos en el mismo gráfico las estimaciones suavizadas del vector de estados obtenido con el paquete *FKAS* y con el paquete *forecast*, tenemos la figura 10.

Como puede observarse, ambas estimaciones son casi coincidentes.

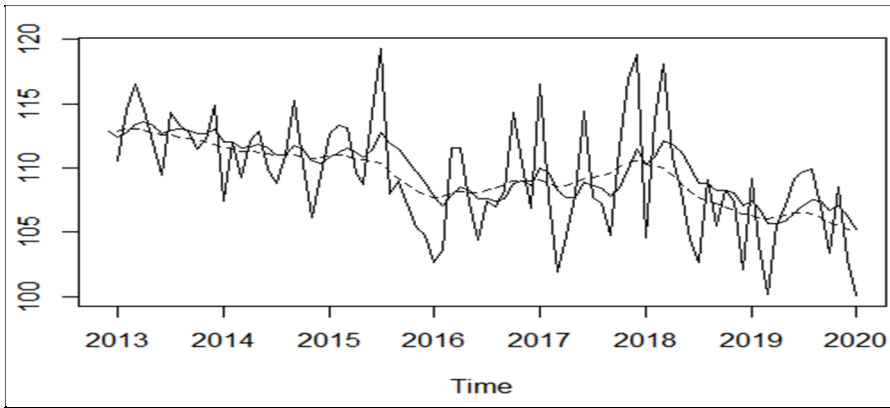


Figura 10. Serie train.ts junto a los dos estados suavizados

Pasando a la predicción, tenemos la tabla siguiente, con su representación en la figura 11

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## Feb 2020	105.2302	100.10278	110.3576	97.38850	113.0719
## Mar 2020	105.2302	100.02598	110.4344	97.27104	113.1893
## Apr 2020	105.2302	99.95029	110.5101	97.15529	113.3051
## May 2020	105.2302	99.87568	110.5847	97.04117	113.4192
## Jun 2020	105.2302	99.80208	110.6583	96.92862	113.5317
## Jul 2020	105.2302	99.72948	110.7309	96.81758	113.6428
## Aug 2020	105.2302	99.65782	110.8025	96.70799	113.7524
## Sep 2020	105.2302	99.58707	110.8733	96.59978	113.8606
## Oct 2020	105.2302	99.51719	110.9432	96.49292	113.9674
## Nov 2020	105.2302	99.44816	111.0122	96.38735	114.0730
## Dec 2020	105.2302	99.37995	111.0804	96.28302	114.1773
## Jan 2021	105.2302	99.31252	111.1478	96.17990	114.2805

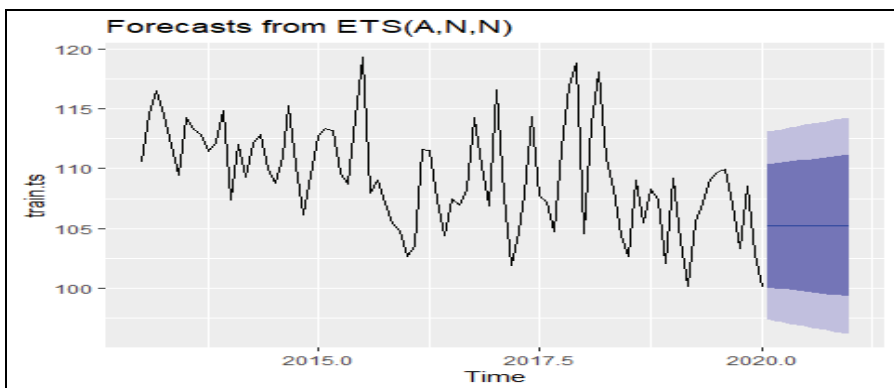


Figura 11. Serie train.ts y predicción del modelo ets.

Veamos la precisión de estas predicciones.

##	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
## Test set	-4.67289	7.460994	5.816841	-5.00565	6.074372	0.5169549	1.355197

En esta ocasión, el valor del $MAPE = 6.07$ es casi coincidente con el $MAPE=5.99$ del modelo estimado con el paquete *FKAS*.

VI. CONCLUSIONES

A manera de resumen podemos destacar los siguientes resultados:

En primer lugar, se ha mostrado la utilidad de los modelos *state space* para el análisis y predicción con series temporales;

Así mismo, se ha mostrado la versatilidad de los paquetes *KFAS* y *forecast* para la estimación y predicción con estos modelos;

Por último, no obstante las semejanzas, hay que señalar la diferencia entre ambos paquetes: el paquete *forecast* es todo un sistema para el análisis de datos.

VII. BIBLIOGRAFÍA

- ALONSO-RODRIGUEZ, A., La predicción de series temporales mediante el modelo de regresión, en *Anuario Jurídico y Económico Escorialense* (San Lorenzo del Escorial), LIV (2021) 501-520.
- ANDERSON B. D. O., y MOORE, J., *Optimal Filtering*, Prentice-Hall, 1979.
- DURBIN, J., y KOOPMAN, *Time Series Analysis by State Space Methods*, segunda edición, Oxford University Press, 2012.
- COMMANDEUR, J. J. F.; KOOPMAN, S. J., y OOMS, M., “Statistical Software for State Space Methods”, in *Journal of Statistical Software*, 41(2011)1-18.
- COMMANDEUR, J. J. F., y KOOPMAN, S. J., *An introduction to State Space Time Series Analysis*, Oxford University Press, 2007.
- COWPERTWAIT, P. S. P., y METCALFE, A. V., *Introductory Time series with R*, Springer, 2009.

- CRYER, J. D., y CHAN, K-S., *Time Series Analysis with Applications in R*, segunda edición, Springer, 2008.
- FRED, Federal Reserve Economic Data, <https://fred.stlouisfed.org>.
- HAMILTON, J. D., *Time Series Analysis*, Princeton University Press, 1994.
- HELSKE, J., “KFAS: Exponential Family State Space Models in R”, *Journal of Statistical Software*, 78(2017)1-39.
- HYNDMAN, R. J.; KOEHLER, A. B.; ORD, J. K., y SNYDER, R. D., *Forecasting with Exponential Smoothing, The State Space Approach*, Springer-Verlag Berlin Heidelberg, 2008.
- HYNDMAN, R.J., y ATHANASOPOULOS, G., *Forecasting, Principles and Practice*, segunda edición, OTexts, 2018.
- HYNDMAN, R. J., y ATHANASOPOULOS, G., *Forecasting, Principles and Practice*, tercera edición, OTexts, 2021.
- KITAGAWA, G., *Introduction to Time Series Modeling*, CRC Press, 2010.
- KLEIBER, CH., y ZEILEIS, A., *Applied Econometrics with R*, Springer, 2008.
- OCDE, <https://www.oecd.org>.
- MARTIN, V.; HURN, S., y HARRIS, D., *Econometric Modelling with Time Series, Specification, Estimation and Testing*, Cambridge University Press, 2013.
- PETRIS, G.; PETRONE, S., y CAMPAGNOLI, P., *Dynamic Linear Models with R*, Springer, 2009.
- PELAGATTI, M. M., *Time Series Modelling with Unobserved Components*, CRC Press, 2016.
- R. R CORE TEAM (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- SHUMWAY, R. H., y STOFFER, D. S., *Time Series Analysis and Its Applications, with R Examples*, cuarta edición, Springer, 2017.
- TUSELL, F., y “Kalman filtering in R”, *Journal of Statistical Software*, 39(2011)1-27.

- WEI, W. S., *Time Series Analysis, Univariate and Multivariate Methods*, Addison-Wesley Publishing Company, Inc., 1990.
- WOODWARD, W. A.; GRAY, H.L., y ELLIOTT, A. C., *Applied Time Series Analysis with R*, segunda edición, CRC Press, 2017.
- ZIVOT, E., y WANG, J., *Modeling Financial Time Series with S-PLUS*, segunda edición, Springer, 2006.

