

ANÁLISIS Y PREDICCIÓN DE SERIES DE TIEMPO EN MERCADOS DE ENERGÍA USANDO EL LENGUAJE R

TIME SERIES ANALYSIS AND FORECASTING IN ENERGY MARKETS USING THE R LANGUAGE

JUAN DAVID VELÁSQUEZ HENAO

Profesor Asociado, Escuela de Sistemas, Universidad Nacional de Colombia, jdvelasq@bt.unal.edu.co

YRIS OLAYA MORALES

Profesora Asociada, Escuela de Sistemas, Universidad Nacional de Colombia, yolayam@bt.unal.edu.co

CARLOS JAIME FRANCO CARDONA

Profesor Asociado, Escuela de Sistemas, Universidad Nacional de Colombia, cjfranco@bt.unal.edu.co

Recibido para revisar agosto 19 de 2010, aceptado noviembre 26 de 2010, versión final diciembre 1 de 2010

RESUMEN: El análisis de series de tiempo y la predicción de variables económicas son tópicos centrales de investigación en el campo de la energía. En este artículo, se revisan los principales aspectos del lenguaje R para el computó estadístico, y se discute su utilidad potencial para los investigadores y profesionales en mercados de energía. También, se revisan las principales funciones disponibles para el análisis de series de tiempo y se presentan algunos ejemplos de su uso.

PALABRAS CLAVE: Proyecto-R, software estadístico, demanda, precios, producción.

ABSTRACT: Time series analysis and forecasting of economic variables are central research topics in the energy field. In this paper, we review the main aspects of the R language for statistical computing, and we stress the potential usefulness for researchers and practitioners of energy markets. Also, we review the main available functions for time series analysis and forecasting, and we present some example of their use.

KEYWORDS: R-project, statistical software, demand, prices, production.

1 INTRODUCCIÓN

El modelado y análisis de las relaciones entre las variables económicas y financieras en diferentes mercados de energía (electricidad, petróleo, gas, carbón, biocombustibles, etc.) son dos de los principales ejes de investigación de la economía energética; las metodologías utilizadas provienen de diversos campos del conocimiento que incluyen, entre otros, la economía, la econometría, la investigación de operaciones, la estadística y las finanzas. Gran parte del análisis está centrado en descubrir y entender la relación entre diferentes variables así como su dinámica, a partir de la información histórica. Véase por ejemplo [1] [2] [3].

En contraste, el pronóstico de series de tiempo está mucho más relacionado con las decisiones de inversión y las finanzas. Tras la liberación y desregulación de muchos mercados energéticos en el mundo, el análisis y la predicción de series de precios de la energía han cobrado una innegable importancia, pues estos mercados se comportan de forma volátil. Ya que la energía es un insumo de todas las economías, el pronóstico de precios de los mercados energéticos es importante, no sólo para la toma de decisiones en los mercados de energía, sino para la toma de decisiones en otros sectores económicos; por ejemplo, en [4] se demuestran que los precios del petróleo son una variable explicativa de los movimientos de la tasa de cambio entre el dólar

americano y las monedas de las principales economías mundiales. No resulta sorprendente que se hayan realizado numerosas publicaciones en el área de la predicción de precios y de la demanda de energía[5][6][7].

Consecuentemente, los investigadores y profesionales, que trabajan en el área de la económica energética, deben desarrollar destrezas y competencias en el uso de herramientas computacionales que permitan abordar estas tareas. No obstante, también es necesario contar con herramientas, suficientemente flexibles y potentes, que faciliten la aplicación de nuevos paradigmas y metodologías. Aunque la variedad de paquetes de cómputo para el análisis de series de tiempo es grande (por ejemplo, SAS, SPSS, Eviews, RATS, Stata o Matlab), nuestra experiencia profesional y docente indica que el uso de estos programas entre los profesionales del área de la energía es limitado por factores como el costo de las licencias de software y por la complejidad del software disponible. A diferencia de otras herramientas, R (The R Project for Statistical Computing) es un software libre, popular en la comunidad estadística y lo suficientemente flexible y potente para analizar y predecir el comportamiento de las variables económicas y financieras usadas en mercados energéticos. Este artículo busca difundir el proyecto R en la comunidad de profesionales en mercados de energía y para esto, se ilustran las capacidades de R en uno de los problemas más frecuentes de los mercados de energía, como el modelado y la predicción de series de tiempo.

El estudio de series de tiempo de variables como precios, aportes hídricos y demanda es una de las actividades más realizadas en el área de la energía. La predicción de estas variables por medio de modelos de series de tiempo y modelos de mercado se aplica en la planeación y evaluación de inversiones, en la valoración de los recursos y el análisis de estrategias comerciales entre otras. A diferencia de otros mercados de materias primas y financieros, los precios en los mercados energéticos suelen determinarse en mercados locales y regionales (con excepción del petróleo y el carbón), como resultado de la dificultad o imposibilidad para almacenar la energía y porque se necesitan redes

de transmisión para arbitrar las diferencias de precios. Como consecuencia de estas particularidades, los modelos para la predicción y análisis de series de tiempo en mercados energéticos, así como la interpretación de los resultados, difieren de los de otros tipos de mercados.

Ya que las herramientas computacionales existentes son muy generales, este artículo busca mostrar a los profesionales e investigadores en mercados energéticos cómo utilizar un software libre y flexible para satisfacer sus necesidades de predicción y análisis de series de tiempo. Además de ilustrar las capacidades de la herramienta R y de guiar el aprendizaje, los ejemplos que presentamos muestran las diferencias entre el modelamiento de las series de los mercados de energía y las de otros mercados financieros y de materias primas; este conocimiento es crucial para aplicar correctamente cualquier metodología e interpretar sus resultados.

El lenguaje R posee una amplia variedad de herramientas generales para el análisis y pronóstico de series de tiempo. No obstante, muchas de las funciones están dispersas en diferentes paquetes, por lo que su uso se dificulta para el usuario no experto. Por esta razón, el segundo objetivo de este artículo, es ilustrar el uso de las principales herramientas disponibles para el análisis y predicción de series de tiempo que son de interés para el profesional y el investigador en el campo de los sistemas energéticos.

Para cumplir con los objetivos planteados, el resto de este artículo está organizado así: en la Sección 2, se realiza una descripción general de las fuentes de información y las características del lenguaje R. En la Sección 3, se presentan, discuten y ejemplifican las funciones más importantes para el análisis y la predicción de series de tiempo. Finalmente, se concluye en la Sección 4.

2 EL LENGUAJE R PARA EL COMPUTO ESTADISTICO

El entorno de programación R [8] es un clon de los lenguajes S [9][10][11]y S-plus [12], de tal

forma que muchos programas escritos en S y S-plus pueden ejecutarse en R sin modificaciones. S y S-plus son lenguajes de muy alto nivel diseñados para [12]:La exploración y visualización de datos; el modelado estadístico; y la programación con datos.

2.1 Adquisición y licencia

El entorno R es un software libre en código fuente bajo la definición dada en la licencia GNU (General PublicLicence) de la FSF (Free Software Foundation), el cual puede ser descargado de la Internet ya sea como código fuente o como un ejecutables para los sistemas operativos Linux (Debian, Redhat, SUSE o Ubuntu), Windows o MacOS. A la fecha de escritura de este artículo se encuentra disponible la versión 2.11.0. El entorno y todo el material complementario pueden ser descargados del sitio <http://www.r-project.org/> o en cualquiera de los servidores web o ftp pertenecientes a CRAN.

2.2 Interfaz de usuario

La interacción con el usuario se basa en una interfaz de línea de comandos, que es bastante apropiada para la manipulación interactiva de datos por parte de usuarios experimentados. No obstante, la falta de una interfaz gráfica de usuario más elaborada frena a los nuevos usuarios, ya que es necesario un entrenamiento básico. En respuesta a esta falencia, se han diseñado interfaces alternativas de usuario con el ánimo de facilitar el uso del entorno. Entre las más conocidas se encuentran: R-Commander[13], R-Integrated Computing Environment o R-ICE [14] y Tinn-R.

2.3 Lenguaje de programación

En su gramática, la sintaxis del lenguaje R es similar a la de C y C++ [15], pero su semántica sigue los paradigmas de la programación funcional y la programación orientada a objetos, lo que implica que el lenguaje tiene la capacidad de manipular directamente los objetos del lenguaje, aplicar reglas de sustitución y evaluar expresiones.

R es un lenguaje orientado a objetos, tal que, inclusive los tipos de más básicos datos, tales como: booleanos, enteros, reales, caracteres, vectores, matrices, listas y hojas de datos son objetos mismos. Esta característica permite que el usuario interactúe de forma transparente, ya que las llamadas se realizan a funciones genéricas, como print, summary o plot, las cuales determinan internamente que método debe ser llamado dependiendo de la clase de objetos a las que pertenecen sus argumentos. R soporta internamente dos implementaciones para la programación orientada a objetos llamadas S3 [11], que fue diseñado para su uso interactivo, y S4 [10], el cual supera las deficiencias de S3, y adiciona nuevos elementos. Adicionalmente, R el usuario puede definir sus propias clases y los métodos asociados a ellas [15].

Varias razones explican la popularidad del lenguaje R. Una de ellas es la amplia colección de paquetes de alto nivel para la construcción de gráficos y su posterior análisis [16][17]. Otra es la capacidad de extensión de la funcionalidad del entorno a través de paquetes [18][19] que puede incluir rutinas compiladas en los lenguajes Fortran, C y C++.

2.4 Información disponible

La información sobre el entorno y sus aplicaciones es abundante, destacándose las publicaciones arbitradas y seriadas en las que se describen nuevos paquetes y sus funcionalidades como “The R Journal” y “Journal of Statistical Software”.

3 FUNCIONES DISPONIBLES PARA EL ANÁLISIS Y LA PREDICCIÓN DE SERIES DE TIEMPO

El lenguaje R ha sido utilizado intensivamente para ilustrar diferentes metodologías de análisis y predicción de series de tiempo [20][21][22][23][24]. También existe material de referencia general; véase [18][19][25][26].

Las funciones disponibles para el análisis de series de tiempo se encuentran agrupadas en

paquetes desarrollados por diferentes autores, los cuales pueden ser instalados desde la barra de menú “paquetes/Instalar paquete(s)...”. Los paquetes deben ser cargados manualmente en cada sesión mediante la opción “Cargar paquete...” del menú “Paquetes” o a través de la función `library()`. Los principales paquetes disponibles para el análisis y la predicción de series de tiempo se encuentran listados en la Tabla 1. A continuación, se discute algunas de sus principales funciones, agrupadas de acuerdo con las principales fases de desarrollo un modelo explicativo o predictivo.

3.1 Entrada de los datos

La unidad fundamental de información es el objeto `time-series`, creado con la función `ts()`; este almacena la información relacionada con la secuencia de datos como tal, su periodo, el momento en que ocurre la primera información, etc. En el siguiente fragmento de código, se crea la variable `gasdem` la cual contiene la información de la demanda mensual de gasolina desde enero de 1960 hasta diciembre de 1975 en Ontario (Canadá) [27]:

```
>gasdem<-
+ ts( data = c(
+ 87695, 86890, 96442, 98133,
+ 113615,123924, 128924, 134775,
+ 117357, 114626, 107677, 108087,
+ 92188, 88591, 98683, 99207,
+ 125485, 124677, 132543, 140735,
+ 124008, 121194, 111634, 111565,
+ 101007, 94228, 104255, 106922,
+ 130621, 125251, 140318, 146174,
+ 122318, 128770, 117518, 115492,
+ 108497, 100482, 106140, 118581,
+ 132371, 132042, 151938, 150997,
+ 130931, 137018, 121271, 123548,
+ 109894, 106061, 112539, 125745,
+ 136251, 140892, 158390,148314,
+ 144148, 140138, 124075, 136485,
+ 109895, 109044, 122499, 124264,
+ 142296, 150693, 163331, 165837,
+ 151731, 142491, 140229, 140463,
+ 116963, 118049, 137869, 127392,
+ 154166, 160227, 165869, 173522,
+ 155828, 153771, 143963, 143898,
+ 124046, 121260, 138870, 129782,
+ 162312, 167211, 172897, 189689,
+ 166496, 160754, 155582, 145936,
+ 139625, 137361, 138963, 155301,
+ 172026, 165004, 185861, 190270,
+ 163903, 174270, 160272, 165614,
```

```
+ 146182, 137728, 148932, 156751,
+ 177998, 174559, 198079, 189073,
+ 175702, 180097, 155202, 174508,
+ 154277, 144998, 159644, 168646,
+ 166273, 190176, 205541, 193657,
+ 182617, 189614, 174176, 184416,
+ 158167, 156261, 176353, 175720,
+ 193939, 201269, 218960, 209861,
+ 198688, 190474, 194502, 190755,
+ 166286, 170699, 181468, 174241,
+ 210802, 212262, 218099, 229001,
+ 203200, 212557, 197095, 193693,
+ 188992, 175347, 196265, 203526,
+ 227443, 233038, 234119, 255133,
+ 216478, 232868, 221616, 209893,
+ 194784, 189756, 193522, 212870,
+ 248565, 221532, 252642, 255007,
+ 206826, 233231, 212678, 217173,
+ 199024, 191813, 195997, 208684,
+ 244113, 243108, 255918, 244642,
+ 237579, 237579, 217775, 227621),
+ start = 1960, frequency = 12)
>
```

El diseño del sistema permite que funciones generales como `plot()`, `line()`, `points()` o `sqrt()` puedan ser aplicadas a un objeto `time-series`; adicionalmente, existen funciones específicas que operan directamente sobre él

Funciones como `filter()` o `lag()` permiten transformar la serie de tiempo. Véase la Tabla 2; detalles y opciones específicas para cada función pueden obtenerse consultando el sistema de ayuda. Por ejemplo, en la Figura 1 se grafica la variable `gasdem`, y su tendencia de largo plazo, obtenida mediante la función `decompose()` del paquete `stats`. El código para generar la Figura 1 fue el siguiente:

```
> d <- decompose(gasdem)
>plot(gasdem, type = 'o', pch = 20)
>lines(d$trend, lwd = 2, col = 'red')
>grid()
```

La gráfica presentada (Figura 1) muestra la presencia de un fuerte patrón estacional, consistente con la dependencia del consumo de gasolina de la temperatura promedio, tal como describen en [27], así como una tendencia creciente, resultado del crecimiento del ingreso.

La entrada de los datos no está limitada a la línea de comandos, de tal manera que se puede acceder a información almacenada en archivos

de texto, hojas de cálculo y diferentes bases de datos[25].

3.2 Análisis exploratorio de la información

En el marco de los modelos ARIMA, el análisis exploratorio se basa principalmente en el uso de contrastes de raíces unitarias y el análisis de los gráficos de las funciones de autocorrelación simple y parcial; no obstante, cuando hay presencia de no linealidades, tanto en la media como en la varianza, se deben utilizar otras herramientas exploratorias.

En la Tabla 3, se resumen y describen las principales funciones disponibles para el análisis exploratorio de una serie de tiempo. Para ejemplos detallados véase [20][21], [22][23].

3.3 Especificación y pronóstico

Existen varios modelos disponibles para el modelado y el pronóstico de series de tiempo. Estos abarcan desde aproximaciones más

tradicionales como los modelos de suavizado exponencial y modelos ARIMA (paquetes stats y forecast), hasta modelos no lineales en media como las redes neuronales autoregresivas y modelos de transición de regímenes (paquete tsDyn), y los modelos de varianza condicional. En la Tabla 4, se listan las principales funciones para crear modelos de predicción y el correspondiente paquete donde se encuentran implementadas.

El pronóstico varios pasos adelante es realizado, comúnmente, por la función predict(), la cual se encuentra sobrecargada con una versión específica para cada modelo utilizado. En algunos casos, también se dispone de la función simulate(), que permite generar series simuladas a partir de un modelo ya ajustado.

En el caso de los modelos ARIMA, la función auto.arima() realiza una búsqueda entre un conjunto de posibles modelos y selecciona el mejor de ellos basado en un criterio de información.

Tabla 1. Principales paquetes para el análisis de series de tiempo
Table 1. Main packages for time series analysis

Paquete	Descripción
expsmooth	Datos del libro "Forecasting with exponential smoothing: the state space approach" [24].
FinTS	Funciones, datos y ejemplos del libro "Analysis of Financial Time Series" [28].
forecast	Métodos y herramientas para visualizar y realizar pronósticos de series de tiempo univariadas usando técnicas de suavizado exponencial basado en espacio de estados y la selección automática de modelos ARIMA. El paquete y su funcionalidad son discutidos por Hyndman y Khandakar[29].
stats	Paquete base con funciones estadísticas básicas
tseries	Funciones y datos para el análisis de series de tiempo y finanzas computacionales [30].
uroot	Contrastes de raíces unitarias para series de tiempo con componentes periódicas

Tabla 2. Funciones para transformar una serie de tiempo
Table 2. Functions for time series transformation

Función	Paquete	Descripción
ts()	stats	Crea un objeto time-series
window()	stats	Permite seleccionar un subconjunto de datos de una serie de tiempo
diff()	base	Diferenciación aplicada n veces para un retardo definido por el usuario
lag()	stats	Computa la versión retardada de una serie de tiempo.
stl()	stats	Descompone una serie de tiempo en sus componentes de tendencia, ciclo estacional (aditivo o multiplicativo), e irregular usando LOESS
filter()	stats	Filtrado lineal de una serie de tiempo

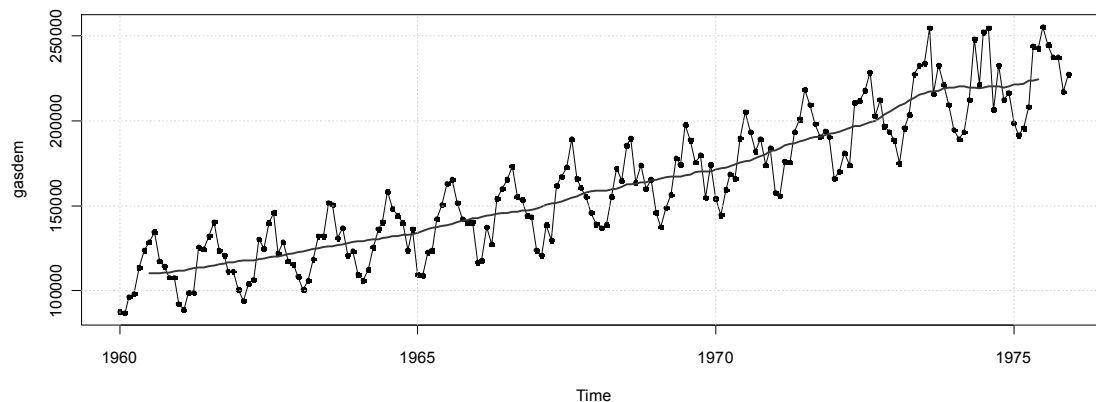


Figura 1. Gráfico de la serie de tiempo de la demanda de gasolina y su tendencia
Figure 1. Plot of the gasoline demand time series dataset and trend

Esta es una clara ventaja, ya que la correcta especificación de un modelo ARIMA es una tarea difícil, aún más cuando la dinámica de la serie de tiempo es compleja; por ejemplo, para la serie de ejemplo utilizada, debería considerarse un modelo SARIMA(p, d, q)(P, D, Q) $_m$ [o ARIMA estacional], donde los valores de los parámetros son escogidos a partir de la experiencia del modelador.

El valor retornado por `auto.arima()` puede ser usado directamente para generar el pronóstico de la serie de tiempo utilizada. En la Figura 2(a), se presenta la comparación entre la serie real y el pronóstico obtenido usando la función `auto.arima()`. Para la información disponible desde el principio de la serie hasta 1973(12), el pronóstico corresponde a la predicción del mes actual utilizando como entrada al modelo la información real; a partir de 1974(1), inclusive,

se realizó el pronóstico con un horizonte de 24 meses hacia adelante usando la información disponible hasta 1973(12).

La línea vertical en la Figura 2(a) corresponde al último dato disponible para el ajuste del modelo, esto es, la observación en 1973(12). La predicción fue generada con el siguiente código:

```
>yfit<- window(gasdem, end = c(1973,
+ 12))
> m1 <- auto.arima(yfit, d = 1, D = 1,
+ max.p = 5, max.q = 5, max.P = 2,
+ max.Q = 2, max.order = 5, start.p=2,
+ start.q=2, start.P=1, start.Q=1)
> p1 <- forecast(m1, h = 24)
>plot(gasdem, col = 'black',
+ ylim = range(c(gasdem,p1$mean)),
+ lwd = 1, pch = 20, type = 'o')
```

Tabla 3. Funciones para graficar y calcular estadísticos descriptivos de una serie de tiempo
Table 3. Functions for plotting and calculating descriptive statistics of a time series

Función	Paquete	Descripción
plot()	graphics	Función genérica para graficar objetos del lenguaje
plot.ts()	stats	Grafica una serie de tiempo
sd()	stats	Desviación estándar
var()	stats	Varianza
hist()	graphics	Histograma
density()	stats	Estimación no paramétrica de la función de densidad de probabilidad
qqnorm()	stats	Grafica Q-Q
qqline()	stats	Agrega un línea a la gráfica Q-Q que pasa por el 1er y el 3r. cuartil.
acf()	stats	Función de autocorrelación simple
pacf()	stats	Función de autocorrelación parcial
ccf()	stats	Función de autocorrelación cruzada
eacf()	TSA	Función de autocorrelación simple extendida
tsdisplay()	forecast	Produce un gráfico de la serie de tiempo, su ACF y su PACF
lag.plot()	stats	Diagrama dirigido de dispersión
adf.test()	tseries	Contraste aumentado de Dickey-Fuller
pp.test()	tseries	Contraste de integración de Phillips-Perron
kpss.test()	tseries	Contraste de estacionalidad de Kwiatkowski-Phillips-Schmidt-Shin.

```
>lines(fitted(m1), col = 'red', lwd = 2)
>lines(p1$mean, col = 'red', lwd = 2)
>grid()
>abline(v=(1973+11/12))
>
```

En este caso el modelo obtenido es:

```
> m1
Series: yfit
ARIMA(2,1,0)(0,1,2)[12]

Call: auto.arima(x = yfit, d = 1, D = 1,
max.p = 5, max.q = 5, max.P = 2,
max.Q = 2, max.order = 5, start.p = 2,
start.q = 2, start.P = 1, start.Q = 1)

Coefficients:
ar1      ar2      sma1      sma2
-1.0198 -0.6720 -0.7255 -0.1540
s.e.0.0611  0.0624  0.1131  0.0924

sigma^2 estimated as 22130445:
log likelihood = -1539.5
AIC = 3089   AICc = 3089.4
BIC = 3104.22
```

El pronóstico de la demanda también puede obtenerse usando suavizado exponencial de Holt-Winters. El siguiente código permite obtener el gráfico de la Figura 2(b).

```
> m2 <- HoltWinters(yfit)
> p2 <- predict(m2, n.ahead = 24)
>plot(gasdem, col = 'black',
+ ylim = range(c(gasdem,p2)), lwd = 1,
+ pch = 20, type = 'o')
>lines(fitted(m2)[,1], col = 'red',
+ lwd = 2)
>lines(p2, col = 'red', lwd = 2)
>grid()
>abline(v=(1973+11/12))
>
```

3.4 Diagnóstico

En la Tabla 5, se listan las principales funciones para realizar el diagnóstico de los residuales

obtenido después de calibrar un modelo de series de tiempo.

4 CONCLUSIONES

Uno de los principales campos de investigación en mercados energéticos es el modelado de diferentes series que incluyen los precios de los energéticos, su demanda en diferentes escalas de tiempo y muchas otras series relacionadas, con el fin de comprender mejor los diferentes hitos históricos que explican sus fluctuaciones en el tiempo. Además de comprender el comportamiento de las series, desde un punto de vista pragmático, deben desarrollarse modelos que para pronosticar su evolución.

El lenguaje R para la computación estadística es un paquete gratuito que provee muchas de las herramientas necesarias para estudiar dichas series de interés. En este artículo se ha realizado

una introducción a dicho entorno de cómputo, haciendo énfasis en las principales fuentes de información detallada, así como también en los principales paquetes y funciones necesarios para realizar las tareas de modelado y predicción. Los principales objetivos de este artículo son: dar a conocer al lector dicha herramienta de trabajo, ilustrando su potencialidad en el área de la econometría energética; y facilitar las primeras experiencias en su uso al presentar una guía detallada de las principales funciones en el análisis y predicción de series de tiempo.

REFERENCIAS

- [1] PARK, J.; RATTI, R. A. 2008. Oil price shocks and stock markets in the U.S. and 13 European countries. *Energy Economics*, 30 (5), pp. 2587–2608.

Tabla 4. Modelos
Table 4. Models

Función	Paquete	Descripción
HoltWinters()	stats	Suavizado exponencial
ar()	stats	Ajusta un modelo autorregresivo (AR) a la serie de tiempo analizada
arima()	stats	Ajusta un modelo ARIMA a la serie de tiempo analizada
auto.arima()	forecast	Busca el mejor modelo ARIMA para una serie de tiempo
StructTS()	stats	Ajusta un modelo estructural a una serie de tiempo
garch()	stats	Ajusta un modelo GARCH a una serie de tiempo
theta()	forecast	Pronóstico usando el método Theta
rwf()	forecast	Pronóstico basado suponiendo que la serie es un paseo aleatorio

Tabla 5. Funciones para el diagnóstico del modelo
Table 5. Model residuals diagnostic functions

Función	Paquete	Descripción
tsdiag()	stats	Diagnostic Plots for Time-Series Fits
Box.test()	stats	Contrastes de Box-Pierce y Ljung-Box
jarque.bera.test()	tseries	Contraste de normalidad de Jarque-Bera
McLeod.Li.test()	TSA	Contraste de no linealidad de McLeod-Li

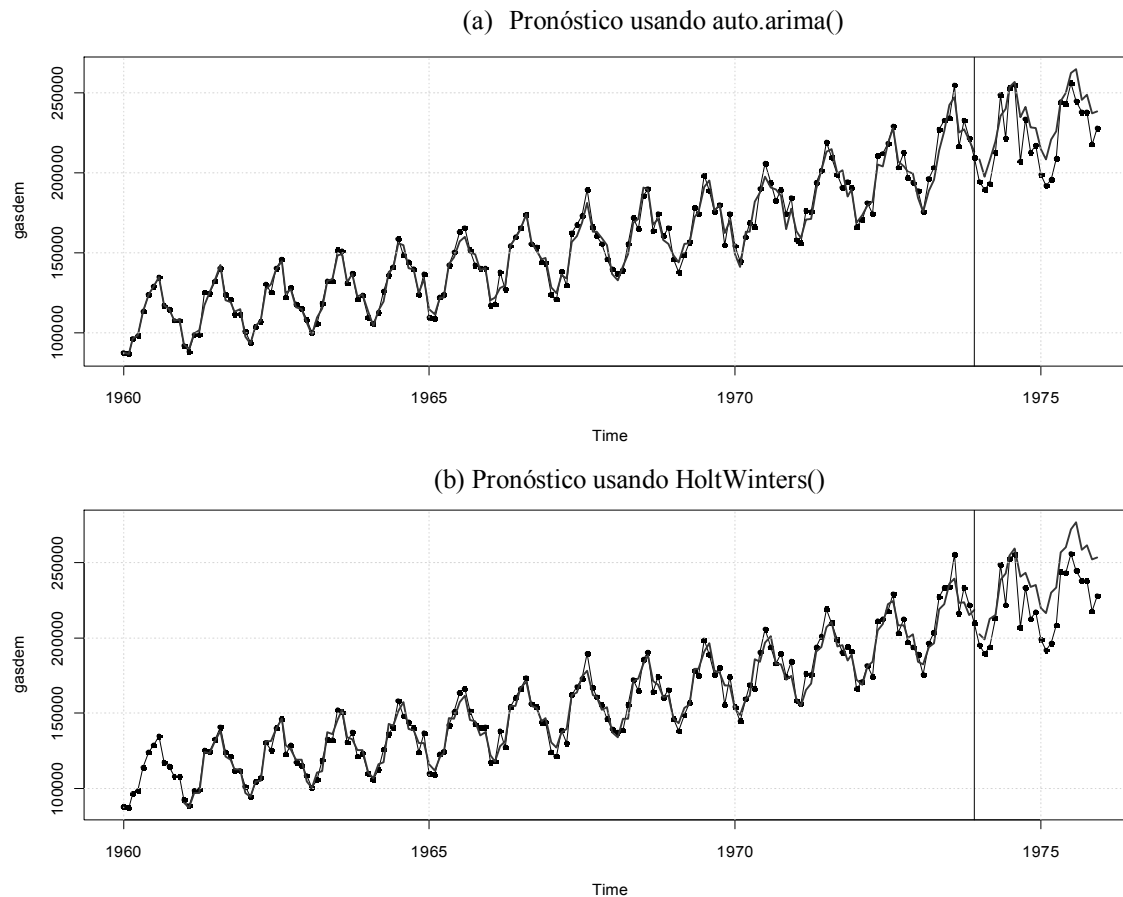


Figura 2. Pronóstico de la demanda mensual de gasolina usando los procedimientos auto.arima() y HoltWinters()
Figure 2. Gasoline monthly demand forecast using the auto.arima() and HoltWinters() procedures

[2] FERDERER, J.P. Oil price volatility and the macroeconomy. *Journal of Macroeconomics*, 18, 1–26, 1996

[3] YU, L.; WANG, S.; LAI, K. K. 2008. Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Economics*, 30, 5, pp. 2623–2635.

[4] LIZARDO, R.A.; MOLLICK, A. V. 2010. Oil price fluctuations and U.S. dollar exchange rates. *Energy Economics* 32, 2, pp. 399–408.

[5] CONEJO, A.J.; CONTREAS, J.; ESPÍNOLA, R.; PLAZAS, M.A. 2005. Forecasting electricity prices for a day-ahead pool-based electric energy market. *International Journal of Forecasting* 21, 3, pp. 435–462.

[6] TAYLOR, J.W.; DE MENEZES, L.M.; MCSHARRY, P.E. 2006. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting* 22, 1, pp. 1–16.

[7] VELÁSQUEZ, J. D.; FRANCO, C. J.; GARCÍA, H. A. 2009. Un modelo no lineal para la predicción de la demanda de electricidad en Colombia, *Estudios Gerenciales*, 25, 112, pp. 37–54.

[8] IHAKA, R., GENTLEMAN, R. 1996. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5, pp. 299–314.

[9] BECKER, R.; CHAMBERS, J. M.; WILKS, A. 1988. The (new) S language: A

programming environment for data analysis and graphics. Pacific Grove: Wadsworth & Brooks/Cole.

[10] CHAMBERS, J. M. 1998. Programming with data: A guide to the S language. New York: Springer-Verlag.

[11] CHAMBERS, J.M.; HASTIE, T. J. 1992. Statistical Models in S. Chapman & Hall, London.

[12] INSIGHTFUL. 2007. S-Plus 8 for Windows. User's Guide. Insightful Corporation, Seattle, WA.

[13] FOX, J. 2005. The R commander: A basic statistics graphical user interface to R. Journal of Statistical Software 14, 9.

[14] SRIPLUNG, H. 2006. Integrated computing environment for R. R package Version 1.0–1. URL: <http://www.r-ice.org>.

[15] GRUNSKY, E. C. 2002. R: a data analysis and statistical programming environment -- an emerging tool for the geosciences. Computers Geosciences 28, 10, pp. 1219–1222.

[16] MURREL, P. 2005. R Graphics. Chapman & Hall/CRC, Boca Raton, FL.

[18] GENTLEMAN, R. 2008. Bioinformatics with R. Chapman & Hall/CRC, Boca Raton, FL.

[19] GENTLEMAN, R. 2008. R Programming for Bioinformatics. Computer Science & Data Analysis. Chapman & Hall/CRC, Boca Raton, FL.

[20] SHUMWAY, R.H.; STOFFER, D.S. 2006. Time Series Analysis and Its Applications With R Examples. Springer, New York.

[21] CRYER, J.D.; CHAN, K.S. 2008. Time series analysis with applications in R. Springer, New York.

[22] COWPERTWAIT, P.S.P.; METCALFE, A. 2009. Introductory Time Series with R. Springer Series in Statistics. Springer.

[23] PFAFF, B. 2008. Analysis of Integrated and Cointegrated Time Series with R. Springer, New York, 2nd edition.

[24] HYNDMAN, R.; ORD, J.K.; KOEHLER, A.B.; SNYDER, R.D. 2008. Forecasting with Exponential Smoothing: The State Space Approach. Springer.

[25] SPECTOR, P. 2008. Data Manipulation with R. Springer, New York.

[26] HEIBERGER, R.M.; HOLLAND, B. 2004. Statistical Analysis and Data Display: An Intermediate Course with Examples in S-Plus, R, and SAS. Springer Texts in Statistics. Springer.

[27] ABRAHAM B.; LEDOLTER, J. 1983. Statistical Methods for Forecasting, John Wiley and Sons, Inc.

[28] TSAY, R. 2005. Analysis of Financial Time Series, 2nd ed. Wiley.

[29] HYNDMAN, R.J.; KHANDAKAR, Y. 2008. Automatic Time Series Forecasting: The forecast Package for R. Journal of Statistical Software 27,3 (<http://www.jstatsoft.org/v27/i03>)

[30] TRAPLETTI, A.; HORNİK, K. 2009. tseries: Time Series Analysis and Computational Finance. R package version 0.10-22.