

óstico12.3228Pronósticos para los siguientes cinco periodos empleando la
función *autoplotfigure.caption*.103

Julio César Alonso

Introducción a los pronósticos con modelos estadístico de series de tiempo para científico dedatos (en R)

Primera Versión para comentarios

26 de enero de 2020

Universidad Icesi

Introducción a los pronósticos con modelos estadístico de series de tiempo para científico dedatos (en R) / Julio C. Alonso.

p. cm.—(Wiley series in survey methodology)

“Wiley-Interscience.”

Includes bibliographical references and index.

ISBN XXXXXXXXXX (pbk.)

1. Modelos lineales—Methodology. 2. Social

sciences—Research—Statistical methods. I. XXXXX II. Series.

HA31.2.S873 2007 001.4'33—dc22 2004044064

Índice general

1. Introducción	1
1.1. Diferencia entre planear y pronosticar	2
1.2. Tipos de pronósticos	3
2. Series de tiempo y sus componentes	7
2.1. Componentes más comunes de una serie de tiempo	9
2.2. Componentes de una serie de tiempo en R	13
2.2.1. Construcción de un objeto de series de tiempo.	14
2.2.2. Encontrando los componentes de una serie de tiempo.	16
2.2.3. Otros métodos de descomposición	22
2.3. Ejercicios	31
3. Métodos de suavizamiento	33
3.1. Muestra de estimación y de evaluación	33
3.2. Promedio Móvil	36
3.3. Suavización Exponencial Simple (SES)	39
3.4. Suavización Exponencial Lineal (Holt)	42
3.5. Suavización Exponencial Lineal de Winters (Holt-Winters)	44
3.5.1. Holt-Winters aditivo	44
3.5.2. Holt-Winters multiplicativo	45
3.5.3. Holt-Winters en R	45
3.6. Comentarios finales	51
3.7. Ejercicios	52
4. Estacionariedad	53
4.1. Estacionariedad (débil)	53
4.2. El ruido blanco	56
4.3. ¿Cómo determinar si una serie es ruido blanco?	57
4.3.1. Ejemplo con una serie simulada	58
4.3.2. Autocorrelación	59
4.3.3. Heteroscedasticidad	63

4.3.4. Normalidad	64
4.4. Comentarios Finales	66
4.5. Apéndice: Descripción de pruebas de autocorrelación empleadas en este capítulo	66
4.5.1. Prueba de Rachas (Runs test)	66
4.5.2. Prueba de Box-Pierce y Ljung-Box	67
5. Encontrando la tendencia y la estacionalidad con modelos de regresión	69
5.1. Estimación de la tendencia	69
5.1.1. Tendencia lineal	69
5.1.2. Tendencia cuadrática	74
5.1.3. Tendencia polinómica	78
5.2. Estimación de la estacionalidad	82
5.3. Selección del mejor modelo para pronosticar	86
5.3.1. Evaluación de los supuestos de los residuales	89
5.4. Ejercicio	93
6. Modelos Autoregresivos (AR)	95
6.1. Proceso $AR(1)$	95
6.2. Proceso $AR(2)$	100
6.3. Proceso $AR(p)$	104
6.4. Función de autocorrelación parcial (PACF)	105
7. Modelos de Media Móvil (MA)	111
7.1. Proceso $MA(1)$	112
7.2. Proceso $MA(2)$	114
7.3. Proceso $MA(q)$	116
7.4. Relación entre los procesos MA y AR	117
8. Modelos de ARMA y ARIMA	121
8.1. Características de los procesos $ARMA(p, q)$	123
8.2. Modelos ARIMA	125
8.3. Raíces del polinomio característico del componente AR y el orden de integración	127
8.4. Comentarios finales	130
9. Estimación de modelos ARMA y ARIMA y construcción de pronósticos	131
9.1. La ACF y PACF muestral	133
9.2. Estimación de un modelo ARIMA con la técnica Box-Jenkins	143
9.2.1. Ejemplo con una serie simulada	144
9.2.2. Un ejemplo con le PIB real de Colombia	153
9.3. Criterios de información y selección automática de modelos ARMA	162
9.3.1. Ejemplo con una serie simulada	163
9.3.2. Un ejemplo con le PIB real de Colombia	164

9.4. Construcción del pronóstico y su intervalo	166
9.4.1. Ejemplo con serie simulada	167
9.4.2. Ejemplo con le PIB real de Colombia	169
9.5. Pronósticos e intervalos para sumas de períodos	170
9.6. Comentarios Finales	172
10. Tipos de pronósticos	173
10.1. Diferentes métodos para la generación de pronósticos	173
10.2. Métricas para la evaluación de pronósticos	175
10.3. Ejemplo con datos simulados y modelos ARIMA	177
10.3.1. Pronóstico con ventana fija	183
10.3.2. Pronóstico con ventana móvil (rolling window)	186
10.3.3. Pronóstico con ventana recursiva	187
10.4. Evaluación de pronósticos	187
10.5. Comentarios finales	193
11. Combinación de pronósticos	195
11.1. Métodos simples de combinación de pronósticos	196
11.1.1. Promedio simple	196
11.1.2. Mediana	196
11.1.3. Media truncada	196
11.1.4. Media Winzor	197
11.1.5. Método de Bates y Granger	197
11.1.6. Método de Newbold y Granger	197
11.1.7. Ranking inverso	198
11.2. Métodos basados en regresión	198
11.2.1. Mínimos Cuadrados Ordinarios (MCO)	198
11.2.2. Mínima Desviación Absoluta (MDA)	199
11.2.3. Mínimos Cuadrados Restringidos (MCR)	199
11.2.4. Promedio de todas las posibles regresiones	200
11.3. Métodos basados en vectores propios	200
11.4. Combinando pronósticos en R	201
11.4.1. Primer paso: preparación de los datos	204
11.4.2. Segundo paso: Construcción de los pronósticos combinados	204
11.5. Comentarios finales	209
12. Modelos de Vectores Auto Regresivos (VAR)	211
12.1. Selección del orden del VAR (p)	213
12.2. Chequeo de los supuestos del error	217
12.3. Usando los resultados	226
12.3.1. Pronósticos	227
12.3.2. Prueba de causalidad de Granger	228
12.3.3. Funciones impulso respuesta	230
12.4. Apéndices	232
12.4.1. Prueba ARCH-LM multivariada	232

Índice alfabético	237
--------------------------------	------------

Capítulo 1

Introducción

Muchos han dicho que el hombre es el único animal que ríe, pero se ha mostrado que esto no es cierto. De hecho, una de las características de los humanos que nos distinguen de los otros animales es que somos los únicos que pensamos en el futuro (ver Gilbert, 2009 para una discusión filosófica de este tema).

La mayoría de las decisiones que tomamos los seres humanos tienen asociadas prever el comportamiento futuro. Las decisiones, ya sean a nivel personal o de los negocios, implican “imaginar” el futuro. Solamente imaginémonos la decisión de a qué hora saldremos de nuestras casas para llegar a tiempo al trabajo o el estudio. Tenemos que “pronosticar” lo que ocurrirá en el camino, de tal manera que no lleguemos tarde.

En los negocios, desde la creación de un plan de negocio hasta decisiones mas pequeñas como el funcionamiento diario de una planta implican también pronosticar el comportamiento de uno o varios aspectos de la realidad¹. Por ahora, definamos un pronóstico como *el arte y la ciencia de predecir eventos futuros*. En nuestro caso los eventos futuros estarán relacionados con el mundo de los negocios.

Es común que algunas de las decisiones en el mundo de los negocios se manejen discrecionalmente² y de forma intuitiva, sin separar de manera consiente la elaboración del pronóstico y la toma de la decisión per sé. Por ejemplo, el panadero de la esquina típicamente toma a diario la decisión de cuántos panes producir. Al mismo tiempo que está tomando su decisión emplea su experiencia e intuición para hacer su pronóstico de cuántas unidades venderá.

No obstante, esto es común, es deseable que las organizaciones empleen un proceso de pronóstico no discrecional y de toma de decisión que no sea orientado por pre-concepciones del comportamiento futuro que emplee juicios de valor y difícilmente el proceso sea replicable por otra persona.

Si bien los pronósticos no discrecionales pueden ser deseables y la intuición del comportamiento de las variables de negocio es importante, en este libro nos concen-

¹ En estadística, a un aspecto de la realidad se le denomina variable

² Por pronóstico discrecional entenderemos aquel que no sigue ninguna regla y que depende de la subjetividad de quien o quienes lo hacen.

traremos en técnicas estadísticas que permitan generar pronósticos del comportamiento futuro de una característica de la realidad (variable) a partir de información histórica empleando métodos estadísticos que son replicables. Estos métodos no emplean la intuición o preconcepciones del analista y en este sentido son pronósticos que son guiados por los datos (*Data driven*).

Pero antes de continuar es muy importante reconocer que cualquier proceso de toma de decisiones implicará determinar qué ocurrirá en el futuro. Ya sea que se realice un proceso de **planeación** o de **proyección**.

1.1. Diferencia entre planear y pronosticar

Los procesos de planeación y pronosticar son actividades básicas e importantes que a veces se entienden como sinónimos, pero no lo son. Ambos procesos están estrechamente relacionados entre sí, pero no son lo mismo. La planificación da respuesta a cómo, cuándo y qué hacer para que el futuro se comporte como se desea o como se espera de acuerdo a un escenario. Así, la planeación es una actividad orientada a cumplir unos objetivos que implica futuros cursos de acción y proporciona un futuro entorno “deseado” para la organización.

Como el futuro es incierto, los encargados de la planeación (planificadores) se ven obligados a hacer algunos supuestos sobre el comportamiento de la realidad en el futuro. Estos supuestos sobre el comportamiento futuro se denominan previsiones, los cuáles se basan en hechos, tendencia pasada, condición económica esperada, intuición o información disponible sobre el futuro.

La planeación es el proceso *de pensar en el curso de acción futuro que se requiere para lograr un objetivo específico*. Por tanto la planeación determina qué curso de acción se debe tomar, cuándo es el momento adecuado, por quién y dónde para llegar a un escenario deseado o esperado. Además, la planeación típicamente implica imaginarse el mejor escenario, el peor escenario, el escenario mas probable, etc.

La planeación implica típicamente los siguientes pasos:

- Identificar el objetivo.
- Seleccionar estrategias para alcanzar el objetivo.
- Identificar los recursos necesarios.
- Crear una línea de tiempo
- Determinar el método de evaluación y seguimiento.
- Identificar tareas y responsables.
- Implementar y supervisar todos los pasos cuidadosamente

Por otro lado, el proceso de realizar pronósticos emplea datos pasados y presentes y el análisis de patrones y tendencias históricas para realizar predicciones del comportamiento futuro. La construcción de pronósticos implica típicamente las siguientes tareas:

- Entender el problema e identificar la variable a pronosticar.

- Crear una base de datos con la información histórica.
- Determinar las características mas relevantes del comportamiento histórico.
- Escoger el “mejor” modelo, entre las diferentes posibles aproximaciones, para explicar el comportamiento de la variable a pronosticar en la muestra de evaluación.
- Realizar un monitoreo continuo del comportamiento de los pronósticos.

Todos estos pasos serán discutidos en este libro. Los pronósticos también implica realizar unos supuestos. El principal es que el comportamiento futuro de la variable de interés será igual al comportamiento pasado. Así mismo, los modelos estadísticos que se emplean para la construcción de los pronósticos también implican unos supuestos y por tanto para tener buenos pronósticos tendremos que estar seguros que los supuestos de los modelos estadísticos se cumplen.

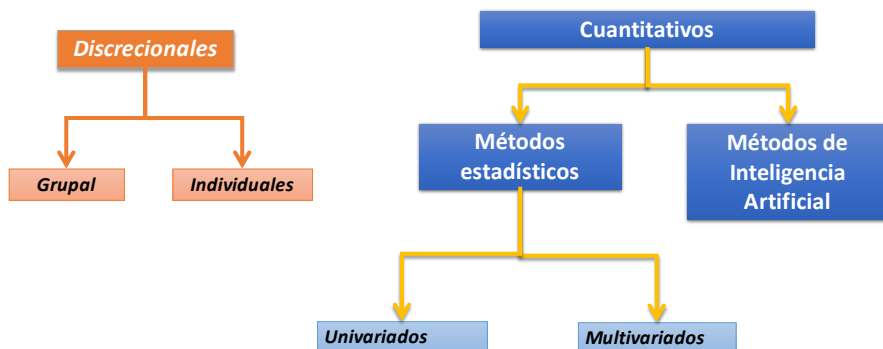
En resumen, la planeación implica el proceso de “imaginarse” las estrategias y tareas necesarias para lograr un objetivo específico. Por otro lado, la construcción de pronósticos es un proceso que emplea datos del pasado para encontrar patrones de comportamiento que permita extrapolar el comportamiento futuro de una variable (característica de la realidad) si el patrón se repite en el futuro. Los pronósticos implican emplear datos del pasado para predecir el comportamiento futuro, bajo el supuesto que se mantienen patrones de comportamientos del pasado en el futuro. Es decir, si todo se mantiene igual o todos los cambios en la realidad actúan de tal manera que se cancelan unos a otros. Comúnmente, los pronósticos ayudan a las organizaciones a reducir la incertidumbre sobre el comportamiento futuro y sirven como base para iniciar un proceso de planeación.

1.2. Tipos de pronósticos

Regresando a la actividad de pronosticar, nuestra gran pregunta a lo largo de este libro será ¿Cómo determinar el comportamiento futuro de la o las variables de interés? Para responder esta pregunta existen básicamente dos tipos de pronósticos (Ver Figura 1.1): los discrecionales y los cuantitativos. Los métodos discrecionales son aquellos que dependen de juicio de personas o grupos y no están sometidos a una regla o norma. Este tipo de pronósticos pueden ser grupales o individuales. Este tipo de pronósticos están por fuera del alcance de este libro.

Por otro lado, los métodos cuantitativos pueden ser de dos clases: los basados en métodos estadísticos y los que emplean métodos de inteligencia artificial.

Este libro se concentra en los pronósticos cuantitativos que emplean métodos estadísticos. Esto se pueden dividir de diferentes maneras pero por ahora creemos dos grupos: los univariados y los multivariados. Los métodos univariados usan técnicas de series de tiempo cuyo objetivo es emplear la información contenida en el comportamiento histórico de la variable para pronosticar su comportamiento futuro. Los métodos multivariados emplean información de una o más variables diferentes a la variable de interés para pronosticar el comportamiento futuro. Un ejemplo de métodos estadísticos multivariados es la regresión múltiple.

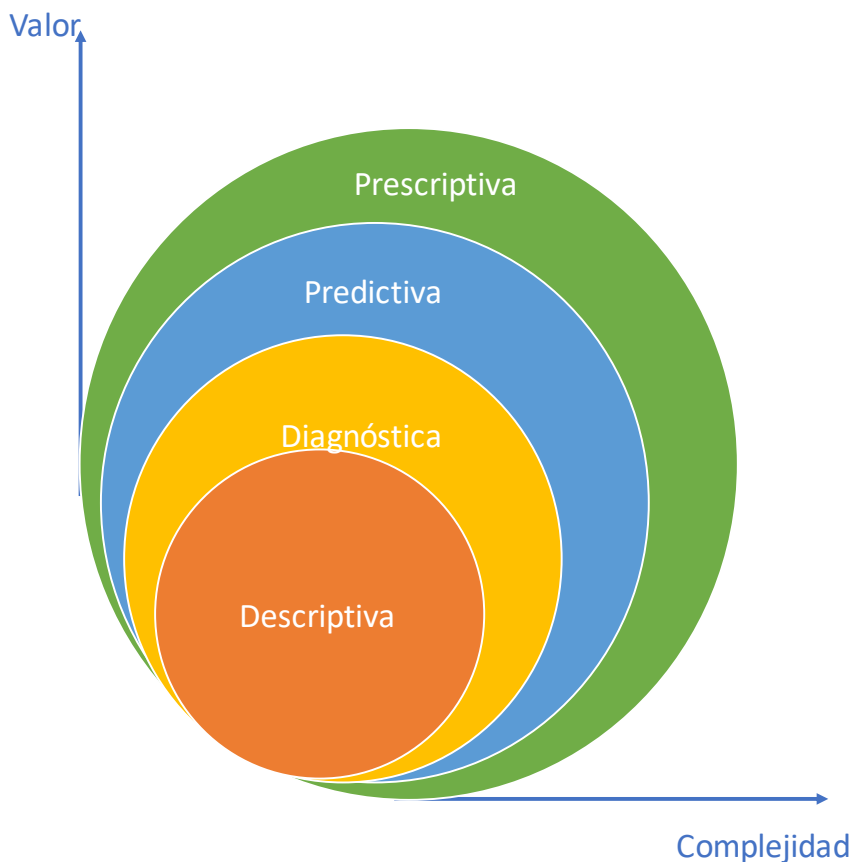
Figura 1.1 Tipos de pronósticos

Antes de continuar es importante mencionar que todo método cuantitativo para pronosticar es por definición una extrapolación de patrones e interrelaciones pasadas. Los buenos resultados de nuestros métodos estadísticos, dependerán de que el futuro sea similar a lo que sucedió en el pasado y que el modelo empleado pueda capturar dicho comportamiento. Si por el contrario, el futuro se comporta de forma diferente al pasado, entonces los modelos no funcionarían bien para generar pronósticos.

El objetivo de este libro será entender las ventajas y limitaciones de los principales métodos estadísticos univariados y multivariados para pronosticar el comportamiento futuro. Empezaremos discutiendo las características de las series de tiempo. Después, estudiaremos los métodos de suavizamiento como los modelos de promedio móvil, suavización exponencial, el método de Holt y el método de Holt-Winters. Posteriormente nos concentraremos en formas de evaluación de modelos. A continuación, discutiremos modelos para pronosticar la tendencia y la estacionalidad. Posteriormente estudiaremos el concepto de estacionariedad y los modelos ARIMA. Finalmente discutiremos los modelos jerárquicos y los modelos multivariados.

Antes de continuar es importante recordar que existen cuatro tipos de analítica: descriptiva, diagnóstica, predictiva y prescriptiva. (Ver Figura 1.2). Estos tipos de analítica responden diferentes preguntas de negocio y no necesariamente una es mejor que otra, cada una cumple una función diferente pero es claro que a medida que pasamos de la analítica descriptiva a la prescriptiva se está generando mayor valor a las organizaciones, pero también se está aumentando el grado de complejidad. La analítica descriptiva.

La analítica descriptiva responde a la pregunta ¿qué está pasando en mi negocio? Para esto usa las bases de datos de la compañía, o las disponibles públicamente y emplea la estadística descriptiva para explorar los datos, resumir información en reportes y visualizaciones que sean útiles para entender qué ha pasado y qué está pasando con el negocio. Este tipo de analítica es la más utilizada en las empresas y es la que requiere menor sofisticación técnica para aplicarlas. No obstante, en el caso de series de tiempo es importante anotar que las estadísticas descriptivas pueden ser un poco más sofisticadas que calcular una media simple o una desviación estándar.

Figura 1.2 Tipos de analítica

Esto será discutido en el Capítulo 2 y Capítulo 4. Por ahora solo imagínese una serie que crece en el tiempo (tiene tendencia creciente), en ese caso calcular una media no tiene mucho sentido. En el Capítulo 2 veremos como descomponer una serie de tiempo en diferentes partes. Este tipo de análisis corresponde a la analítica descriptiva.

La analítica diagnóstica típicamente quiere responder la pregunta de por qué está pasando lo que está pasando en el negocio. Este tipo de análisis tiene la capacidad de encontrar las causas de los problemas, la raíz de las relaciones y es capaz de aislar los efectos de diferentes fenómenos. En nuestro caso, en la primera parte de este libro al concentrarnos en modelos univariados no será posible encontrar relaciones ni podemos aislar efectos. No obstante, en la segunda parte discutiremos modelos multivariados que permitirán realizar analítica diagnóstica; como por ejemplo: la prueba de causalidad de Granger y las funciones impulso respuesta.

La analítica predictiva busca responder la pregunta: ¿qué es posible que ocurra? Para esto se apoya en datos históricos; es decir, con los datos que ya se tiene, se predicen comportamientos que aún no se conocen. En otras palabras, con datos ya existentes se predicen los datos que aún no se tiene o no han ocurrido. Esta obra se centra en este tipo de analítica. Todos los modelos y métodos que estudiaremos quieren responder la pregunta que ocurrirá en el futuro con nuestra variable objetivo. Por ejemplo con este tipo de analítica podemos responder preguntas como: ¿cuánto es lo más probable que venda de un producto el próximo año? o si todo sigue igual ¿cuál sería el comportamiento del precio de uno de mis insumos?

Finalmente, la analítica prescriptiva busca responder la pregunta: ¿qué necesito hacer? Este tipo de analítica hace énfasis en el uso de técnicas de optimización para identificar cuál es la mejor alternativa para minimizar o maximizar algún objetivo así como modelos que permitan simular situaciones. Las técnicas que se estudian aquí pueden hacer parte de una análisis prescriptivo como se discutirá en los respectivos capítulos.

Capítulo 2

Series de tiempo y sus componentes

Buena parte de los datos que se emplean en el mundo de los negocios tienen una estructura de series tiempo. Es decir, son observaciones de una misma variable para diferentes períodos, y generalmente el espacio entre cada observación es siempre el mismo. Por ejemplo: las ventas mensuales, el precio diario del dólar, el PIB anual, la inflación mensual y los rendimientos diarios de un activo. Formalmente, podemos expresar una serie de tiempo de las siguientes formas alternas:

$$\{y_1, y_2, \dots, y_T\} \quad (2.1)$$

$$\{y_t\}, t = 1, 2, \dots, T$$

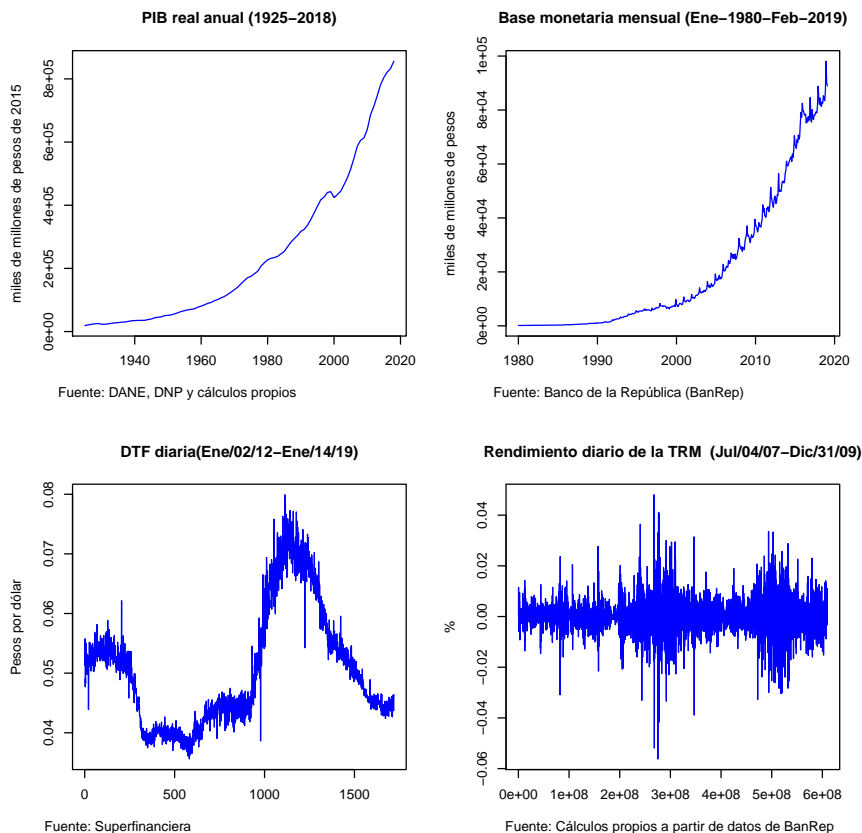
$$\{y_t | t \in 1, 2, \dots, T\}$$

En general, trataremos a $\{y_t\}$ como una variable aleatoria. Antes de continuar es importante resaltar que el término “serie de tiempo” es empleado tanto para referirnos a un conjunto de realizaciones de la variable aleatoria $\{y_t\}$ ¹ como a un modelo probabilístico que describe el comportamiento de la muestra. Por ejemplo, un modelo probabilístico para la variable aleatoria $\{y_t\}$, también conocido como el proceso generador de los datos o DGP (por su sigla del término en inglés Data Generating Process) o ley de movimiento puede ser

$$y_t = \varepsilon_t, \varepsilon_t \sim i.i.d. (0, \sigma_\varepsilon^2) \quad (2.2)$$

donde *i.i.d* representa una variable aleatoria que sigue una distribución idéntica e independiente (en inglés Identically and Independently Distributed). En otras palabras, y_t es una variable aleatoria con media cero y varianza σ_ε^2 y un valor de y_t en un periodo determinado es independiente de los valores pasados o futuros de ella misma.

¹ Es decir, una muestra de $\{y_t\}$. Como por ejemplo los rendimientos diarios de un activo para cada uno de los días estudiados en la muestra.

Figura 2.1 Ejemplo de varias series de tiempo para la economía colombiana

De hecho, una de las principales características de las series de tiempo, tanto macroeconómicas, financieras, como de ventas, es que comúnmente no son i.i.d; típicamente lo que ocurre en el período t depende de períodos anteriores ($t-1, t-2, \dots$). Por ejemplo, para las ventas de un año determinado es muy probable que éstas estarán relativamente cercanas a las ventas observadas en el año inmediatamente anteriores. No es muy común que las ventas de un año sean trascendentalmente diferentes a las de periodos anteriores.

Esto mismo ocurre con las variables macroeconómicas. El PIB de un año tiende a estar muy cerca la del año anterior. Naturalmente existe una diferencia entre un año y otro, pero es relativamente fácil ver como el valor del PIB de un año es relativamente cercano al del año anterior. En la figura 2.1 se presenta el PIB real de Colombia, la base monetaria mensual, la tasa de los certificados a depósitos a término fijo (a 90 días) promedio diaria (DTF) y el rendimiento (logarítmico) de la tasa representativa de mercado (TRM) diaria².

² Es decir, el rendimiento diario es calculado como

Por ejemplo, el PIB anual colombiano presenta una característica predominante: su “tendencia” a crecer. Es decir, a medida que pasa el tiempo la serie crece. En general nuestro objetivo en el análisis de las series de tiempo será determinar la ley de movimiento que gobierna a la muestra bajo análisis y de esta manera poder pronosticar el comportamiento futuro. En otras palabras, queremos encontrar una expresión que describa el comportamiento de la serie. Esa expresión comúnmente implicará parámetros poblacionales (que se consideran determinísticos) y una parte aleatoria. Por ejemplo, el DGP del PIB PIB_t puede ser:

$$PIB_t = \alpha + \beta PIB_{t-1} + \varepsilon_t$$

donde $\varepsilon_t \sim i.i.d. (0, \sigma_\varepsilon^2)$. En otras palabras, es un componente aleatorio. Además, α y β corresponden a parámetros poblacionales (determinísticos). Otro DGP que puede describir ese comportamiento es:

$$PIB_t = \alpha + \beta t + \varepsilon_t$$

Este último DGP implica que la serie crece a medida que el tiempo pasa; es decir, t crece.

Regresando a figura 2.1, la base monetaria también presenta una “tendencia” a crecer, pero además presenta un aumento relativamente regular para los diciembres de cada año (estacionalidad). Por otro lado, la serie de la DTF aparentemente tienen un comportamiento algo diferente, pues no existe una clara tendencia a crecer durante todo el periodo de estudio. Finalmente, el rendimiento de la TRM claramente se mueve al rededor de una media (en este caso cero) pero la serie presenta una desviación con respecto a la media que es cambiante (volatilidad no constante).

2.1. Componentes más comunes de una serie de tiempo

En general, una serie de tiempo, como las presentadas en la sección anterior o las que se observan en el mundo de los negocios, puede estar conformada por uno o varios de los siguientes componentes:

- Tendencia
- Factor Cíclico
- Estacionalidad
- Componente puramente aleatoria

La tendencia de una serie de tiempo corresponde a la dirección de largo plazo de la serie. El factor cíclico recoge los movimientos por encima y por debajo de la tendencia por periodos superiores al año. El componente estacional es el patrón de comportamiento que se repite año tras año. Este componente solo está presente

$$r_t = \ln(TRM_t) - \ln(TRM_{t-1})$$

donde TRM_t representa el promedio diario de la cotización de un dólar americano.

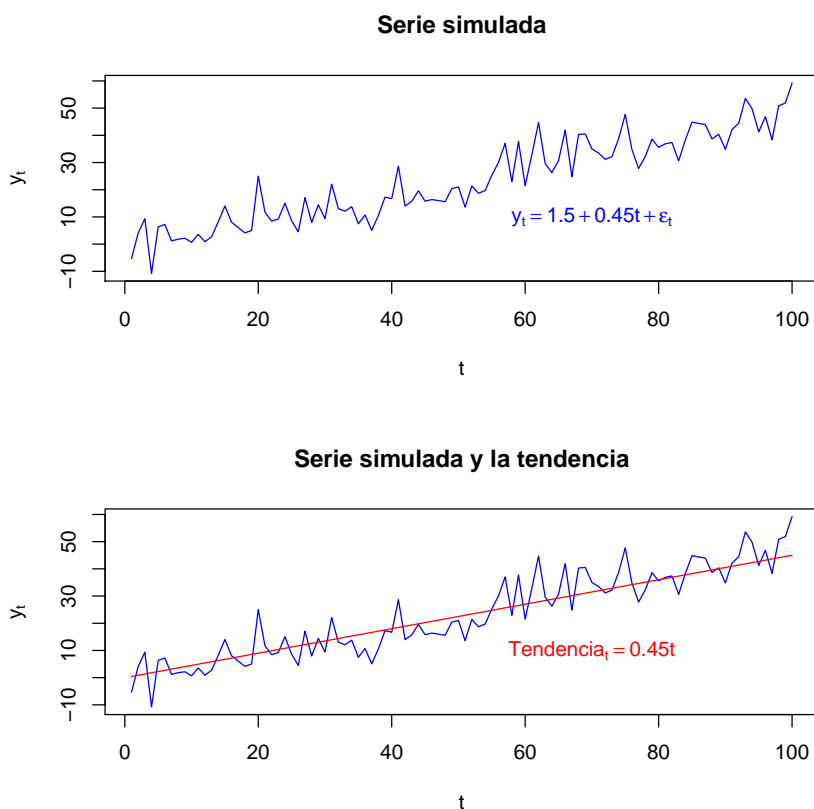
para series de tiempo con una periodicidad menor a un año. Finalmente el componente puramente aleatorio es aquella parte de la serie de tiempo que presenta un comportamiento estocástico. Este componente recoge variaciones en la serie debido a cambios que afectan directamente a la variable bajo estudio u otros eventos importantes que pueden afectar a muchas variables entre ellas la que se encuentra bajo estudio; eventos como guerras y catástrofes nacionales.

En la figura 2.2 se presenta una serie que fue simulada a partir del siguiente DGP que solo contiene una tendencia y el componente puramente aleatorio:

$$y_t = 1,5 + 0,45 \cdot t + \varepsilon_t \quad (2.3)$$

De la figura se puede observar como la serie fluctúa (aleatoriamente) alrededor de la tendencia que es totalmente determinística.

Figura 2.2 Serie de tiempo con tendencia

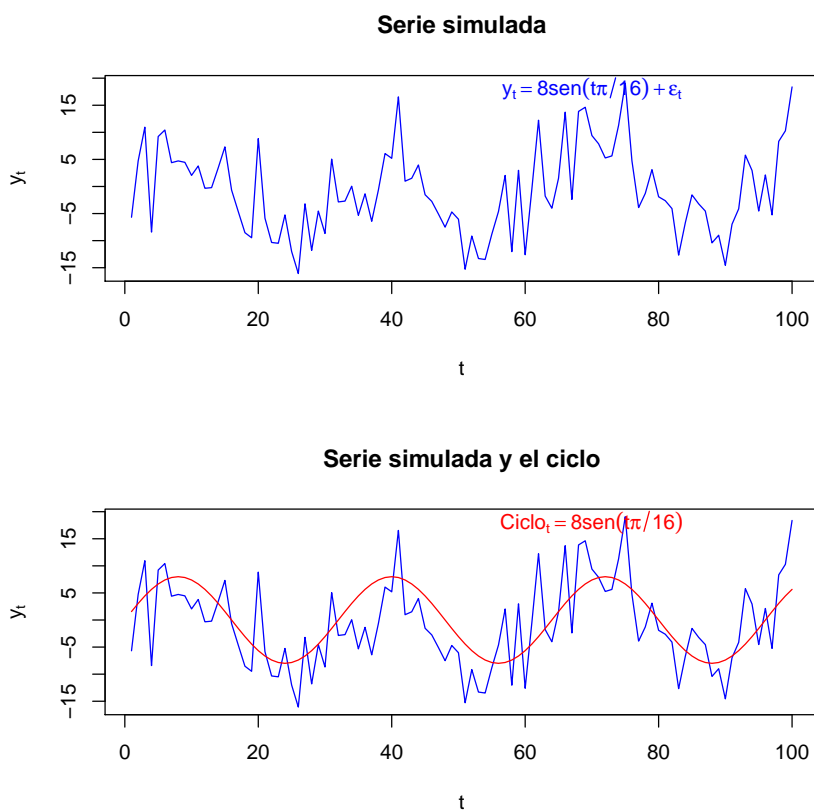


En la figura 2.3 se presenta una serie cuyos únicos componentes son la parte puramente aleatoria y un factor cíclico. En especial, esta serie de tiempo fue simulada para una muestra de 100 períodos a partir de la siguiente ley de movimiento:

$$y_t = 8 \cdot \text{sen} \left(\frac{t \cdot \pi}{16} \right) + \varepsilon_t \quad (2.4)$$

donde $t = 1, 2, \dots, 100$. En este caso se puede observar como la serie fluctúa (aleatoriamente) alrededor de la función senoidal que es totalmente determinística.

Figura 2.3 Serie de tiempo con comportamiento cíclico

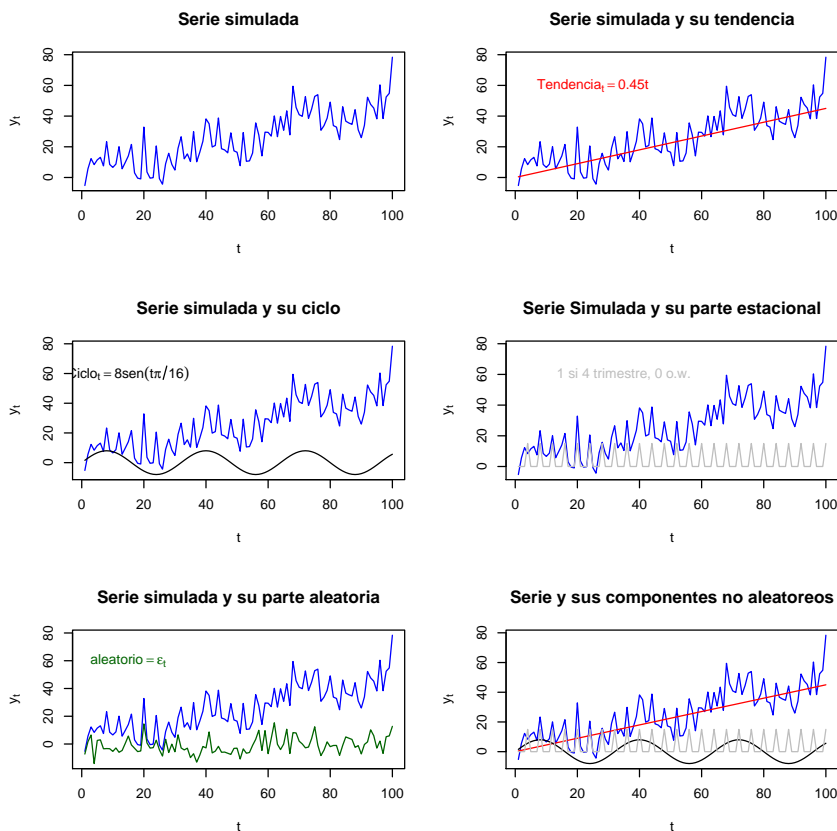


Consideremos ahora un ejemplo más complejo. En la figura 2.4 se puede observar una serie trimestral que posee 4 componentes diferentes: una tendencia, un componente cíclico, un componente estacional y el componente aleatorio. En este caso en especial, el *DGP* que se empleó para generar la muestra para los 100 períodos (y_t para $t = 1, 2, \dots, 100$) es:

$$y_t = 1,5 + 0,45 \cdot t + 8 \cdot \text{sen}\left(\frac{t \cdot \pi}{16}\right) + 15 \cdot D_t + \varepsilon_t \quad (2.5)$$

donde D_t es una variable dummy que toma el valor de 1 cada cuatro periodos y cero en caso contrario. En este caso, observar el gráfico no hace tan evidente la presencia de todos los cuatro componentes.

Figura 2.4 Serie de tiempo con 4 componentes



Hasta aquí los ejemplos de series simuladas que se han presentado corresponde a **sumar cada uno de los componentes**. Esto **implica que la variable se puede descomponer de forma aditiva**. En otras palabras, si definimos el componente cíclico y de tendencia, estacional y aleatorio para un momento t como T_t , S_t y R_t . La descomposición aditiva de una variable de serie de tiempo implica que el DGP corresponde a:

$$y_t = S_t + T_t + R_t \quad (2.6)$$

Pero esta no es la única forma como se combinan los componentes de una serie de tiempo. Otra opción es que la variable pueda ser descompuesta de manera multiplicativa. Es decir, el *DGP* correspondería a

$$y_t = S_t \times T_t \times R_t \quad (2.7)$$

Esta expresión puede parecer un poco más complicada de trabajar; en algunas ocasiones se puede transformar la serie para convertir una descomposición multiplicativa a una aditiva. Una transformación común de las series calculándole el logaritmo de tal manera que:

$$\log y_t = \log S_t + \log T_t + \log R_t \quad (2.8)$$

La pregunta práctica es cuál de las dos descomposiciones emplear. En general, la descomposición aditiva es la que se sugiere si las fluctuaciones estacionales, o la variación en torno al ciclo o la tendencia, es tal que su magnitud no varía con el nivel de la serie. Cuando la variación en el patrón estacional, o la variación alrededor del ciclo o la tendencia, parece ser proporcional al tiempo³, entonces se recomienda una descomposición multiplicativa. No obstante siempre será una buena práctica emplear las dos aproximaciones y chequear si el componente puramente aleatorio si se comporta como tal. En otras, palabras es mejor que los datos nos cuenten cuál de las aproximaciones es la mejor.

En la práctica, nuestra labor de generar pronósticos implicará detectar el *DGP* o ley de movimiento de la variable de interés para posteriormente poder realizar nuestra mejor conjetura del comportamiento futuro. El descomponer una serie puede ser útil para hacer analítica descriptiva; es decir, para describir cuáles son las características dominantes de la serie. Pero la descomposición también puede permitir entender que tipo de modelo sería ideal emplear para realizar pronósticos (analítica predictiva). A continuación discutiremos diferentes formas de descomponer una serie empleando R.

2.2. Componentes de una serie de tiempo en R

Ahora concentrémonos en cómo definir una serie de tiempo en R y cómo encontrar sus componentes. Para este ejercicio emplearemos la información disponible en los archivos `datosEmpleo.xlsx` y `base.csv`. En el primer archivo encontrarán la tasa de desempleo mensual de las 13 principales ciudades en Colombia (TD.13ciudades). El archivo también contiene series mensuales para las 13 principales ciudades de Colombia el número de ocupados en miles de personas (Ocupados), los desocupados (Desocupados) y los inactivos (Inactivos). Estos datos cubren el periodo enero de 2001 a abril de 2019.

El segundo archivo contiene la información de la base monetaria (cantidad de dinero) mensual para el periodo enero de 1980 a febrero de 2019.

³ Es decir, a medida que pasa el tiempo el componente estacional cambia su comportamiento.

2.2.1. Construcción de un objeto de series de tiempo.

Nuestra primera tarea será leer los archivos de Excel. Para el primer archivo podemos emplear el paquete **readxl**. Carguemos los datos en un objeto que denominaremos *data*.

```
library(readxl)
data <- read_excel("datosCOMP/datosEmpleo.xlsx",
  col_types = c("date", "numeric", "numeric",
    "numeric", "numeric") )
class(data)

## [1] "tbl_df"      "tbl"        "data.frame"

head(data)

## # A tibble: 6 x 5
##   mes                TD_13ciudades Ocupados Desocupados
##   <dtm>              <dbl>      <dbl>      <dbl>
## 1 2001-01-01 00:00:00 20.9      6924.      1835.
## 2 2001-02-01 00:00:00 19.9      7038.      1748.
## 3 2001-03-01 00:00:00 19.2      6946.      1653.
## 4 2001-04-01 00:00:00 17.9      6973.      1519.
## 5 2001-05-01 00:00:00 17.9      6994.      1530.
## 6 2001-06-01 00:00:00 18.7      7014.      1608.
## # ... with 1 more variable: Inactivos <dbl>
```

En este caso los datos fueron leídos como un *data frame*. Pero la información correspondiente a la fecha no está indexando la información. En otras palabras, aún no tenemos un objeto que sea reconocido por R como una serie de tiempo.

En R podemos encontrar diferentes clases de objetos para trabajar con series de tiempo. Por ejemplo, existen los objetos *ts* que están asociados al paquete base de R, tal vez el tipo de objeto más común en los paquetes que emplean series de tiempo. Esta clase de objeto es bastante sencilla, pero no permite manipular las series de tiempo de alta frecuencia (minutos, segundos). También encontramos los objetos *xts*, asociados al paquete del mismo nombre que permiten trabajar mucho más cómodamente con series de tiempo de alta frecuencia. Otra clase de objetos que brinda mucha flexibilidad es el **zoo**. En la práctica, tendremos que aprender a usar diferentes clases de objetos de series de tiempo, dado que diferentes paquetes requieren diferentes clases de objeto. Esto lo veremos a lo largo de este libro.

Empecemos creando un objeto de serie de tiempo de clase *ts*, empleando la función con el mismo nombre. Los argumentos de esta función son: un vector o matriz de datos y *frequency* corresponde a la frecuencia de la serie de tiempo. Así, para datos mensuales *frequency* = 12 y para datos trimestrales *frequency* = 4. Para datos diarios, *frequency* = 7. Creemos un objeto de clase *ts*.

```
data.ts <- ts(data[, -1], start = c(2001, 1), frequency = 12)
head(data.ts)

##      TD_13ciudades Ocupados Desocupados Inactivos
```

```
## [1,] 20.94638 6923.604 1834.507 4600.718
## [2,] 19.89421 7037.746 1747.820 4596.805
## [3,] 19.22157 6945.973 1652.823 4807.120
## [4,] 17.88858 6973.079 1519.137 4937.280
## [5,] 17.94565 6994.462 1529.720 4928.911
## [6,] 18.65389 7014.100 1608.439 4854.332
```

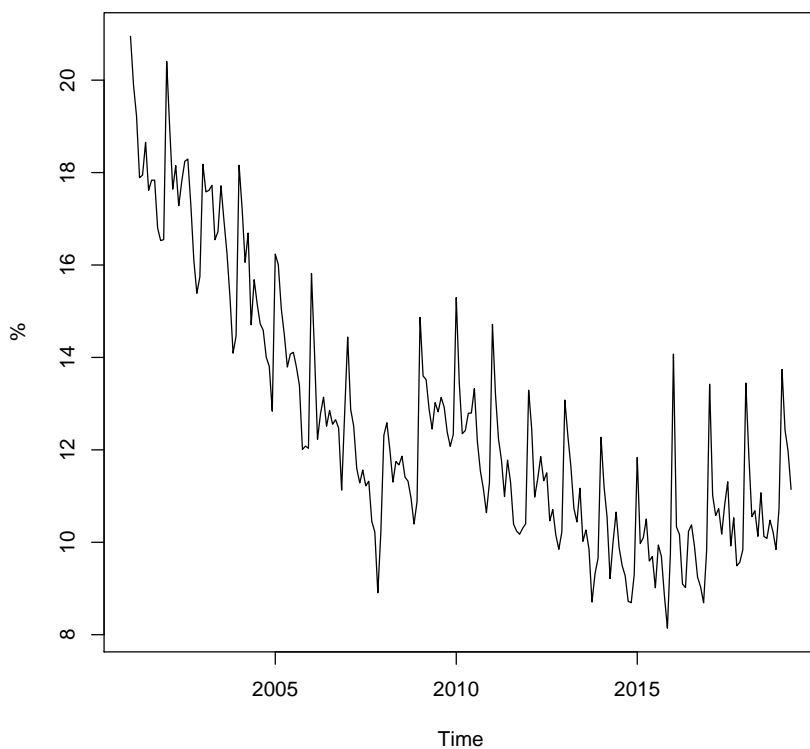
```
library(xts)
periodicity(data.ts)
```

```
## Monthly periodicity from Jan 2001 to Apr 2019
```

Grafiquemos la tasa de desempleo de la siguiente manera:

```
ts.plot(data.ts[,1],
  main = "Tasa de desempleo mensual (13 ciudades)")
```

Figura 2.5 Gráfico de la Tasa de desempleo mensual (13 ciudades) empleando la función `ts.plot`



Noten que si los datos corresponden días laborales, esto no se podrá hacer por medio de esta función. En esos casos es mejor emplear la clase *xts*.

Ahora creemos un objeto *xts*. Para crear un objeto de esta clase, requerimos el vector o matriz de datos y un vector con las fechas que permita ordenar los datos. En la columna *mes* del data frame *data* teníamos las fechas y las leímos como tal. Entonces empleando la función *xts* podemos crear el objeto de series de tiempo de clase del mismo nombre.

```
library(xts)
data.xts <- xts(data[, -1], order.by = data$mes)
class(xts)

## [1] "function"
```

Si la columna *mes* no hubiese sido leída como una fecha, está se puede convertir a fecha empleando el siguiente código

```
data$mes <- as.Date(data$mes)
```

Grafiquemos rápidamente la tasa de desempleo usando la función *plot.xts*.

```
plot.xts(data.xts$TD_13ciudades, main = "Tasa de desempleo mensual (13 ciudades)")
```

Noten que este gráfico es mucho más estético que el anterior. Esta es una de las ventajas de los objetos *xts*.

Por ahora sigamos trabajando solo con la tasa de desempleo.

```
TD.ts <- data.ts[, 1]
TD.xts <- data.xts$TD_13ciudades
```

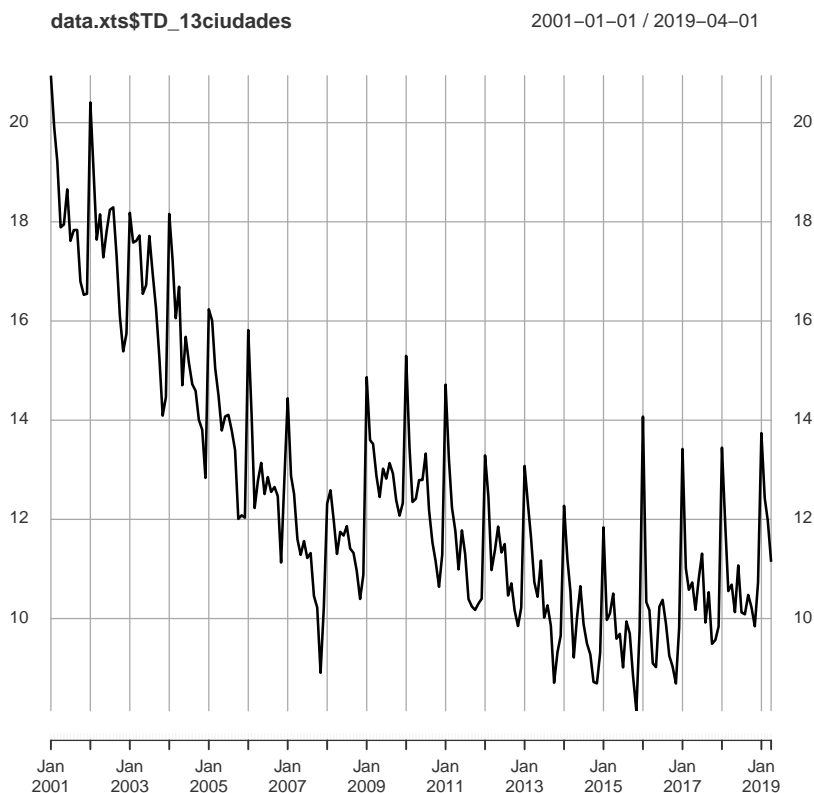
Antes de proceder a descomponer esta serie, carguemos la serie de la mensual de la base monetaria sólo como un objeto *ts*.

```
Base <- read.csv("datosCOMP/base.csv", sep=",")
Base <- ts(Base, start = c(1980, 1), frequency = 12)
```

2.2.2. Encontrando los componentes de una serie de tiempo.

En algunas ocasiones puede ser útil empezar nuestro análisis descomponiendo la serie de tiempo en sus componentes: tendencia, estacionalidad y componente puramente aleatorio. Es decir, hacer un análisis de analítica descriptiva. Una forma de hacer esto es empleando la función *decompose* del paquete *base* de R. Esta función sólo trabaja con objetos de clase *ts*. E incluye dos argumentos, la serie de tiempo que se quiere descomponer y cómo se desea tratar la descomposición de la parte estacional. El método aditivo (*type = "additive"*) se recomienda cuando las variaciones estacionales son aproximadamente constantes a lo largo de la serie, mientras que el método multiplicativo (*type = "multiplicative"*) se emplea cuando las variaciones

Figura 2.6 Gráfico de la Tasa de desempleo mensual (13 ciudades) empleando la función `plot.xts`

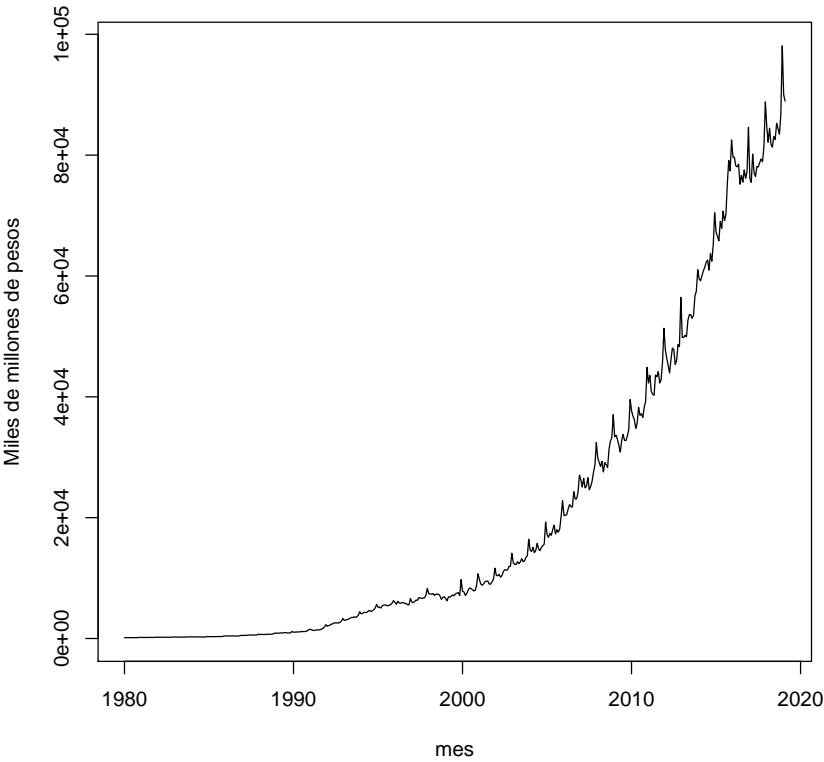


estacionales cambian proporcionalmente al nivel de la serie. Por ejemplo, para la serie de Tasa de desempleo, el componente estacional parece ser constante y por tanto se sugeriría un componente aditivo. Mientras que en el caso de la Base monetaria el componente estacional está creciendo con la serie. En este caso se sugiere emplear un componente estacional multiplicativo.

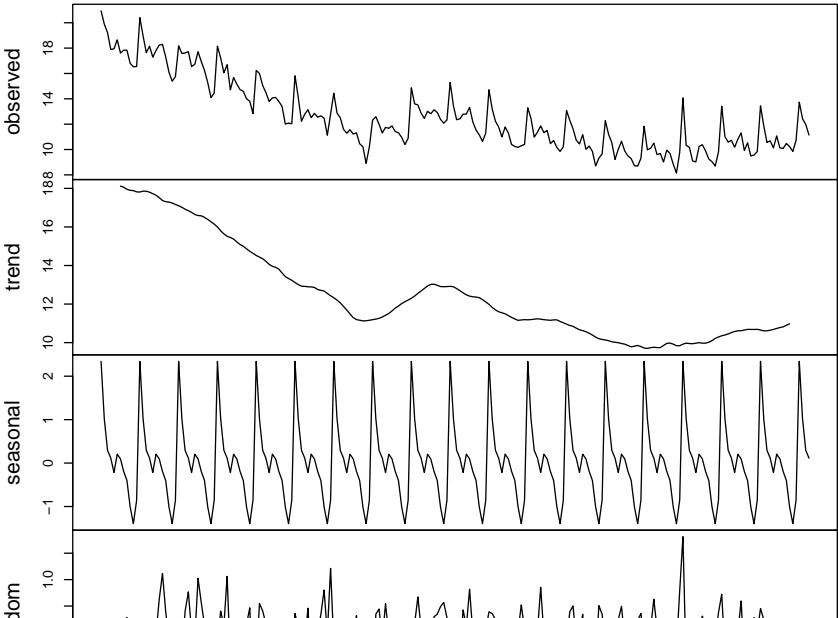
Empecemos con la tasa de desempleo.

```
TD.componentes <- decompose(TD.ts, type = "additive")
plot(TD.componentes)
```

Figura 2.7 Base monetaria mensual



Decomposition of additive time series



Podemos observar el componente estacional marcado en la serie, una tendencia no lineal de los datos y la parte aleatoria.

Es importante mencionar que esta descomposición se emplea solo como referencia para iniciar el análisis y no implica pruebas estadística alguna⁴. Mas adelante estudiaremos herramientas de inferencia que nos permitiría concluir sobre los diferentes componentes.

En algunas ocasiones, puede ser de interés trabajar con las series sin el componente estacional. En este caso una forma sencilla y rápida para quitar el componente estacional (desestacionalizar), cuando la descomposición es aditiva, es restarle a la serie el componente estacional encontrado en la descomposición. Es decir, la serie desestacionalizada será

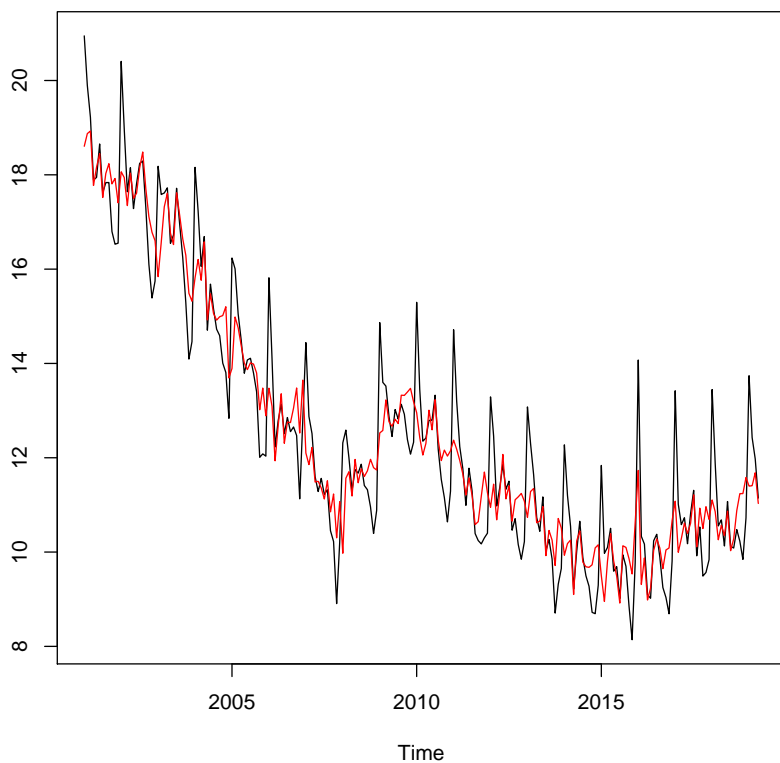
$$y_t - S_t \quad (2.9)$$

En R esto implicaría

```
TD.desestacionalizada <- TD.ts - TD.componentes$seasonal
ts.plot(TD.ts, TD.desestacionalizada, gpars = list(col = c("black", "red")), main="Tasa de de
```

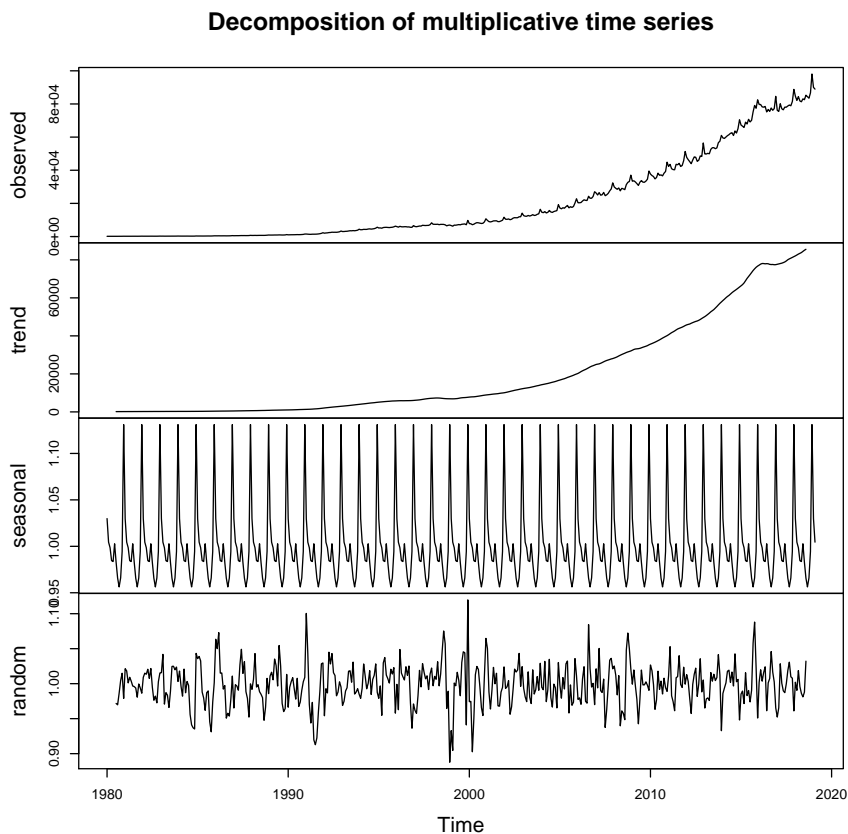
⁴ La función *decompose* calcula la tendencia (T_t) empleando una media móvil. En el Capítulo 3 se discutirá este método. Una vez la tendencia es removida ($y_t - T_t$), el componente estacional (S_t) es estimado sacando un promedio para cada uno de los periodos que conforman una estación. El componente puramente aleatorio corresponde a lo que queda (ε_t). Así, si el método escogido es el aditivo el modelo implícito es $y_t = T_t + S_t + \varepsilon_t$. Si el modelo es multiplicativo el modelo implícito es $y_t = T_t \cdot S_t \cdot \varepsilon_t$.

Tasa de desempleo mensual observada y desestacionalizada (13 ciudades)



Para el caso de la base monetaria la correspondiente descomposición sería:

```
Base.componentes <- decompose(Base, type = "multiplicative")  
plot(Base.componentes)
```



Ustedes podrán constatar que si se emplea el método aditivo para descomponer la serie de la base monetaria, en la parte puramente aleatoria se encontrará un componente que tiene comportamiento estacional. Y por tanto esto es signo de que no se ha removido todo el componente estacional. Cuando se emplea el método multiplicativo la parte aleatoria (el error o residuo) tiene un comportamiento meramente aleatorio. En los siguientes capítulos discutiremos el comportamiento deseable del error en más detalle.

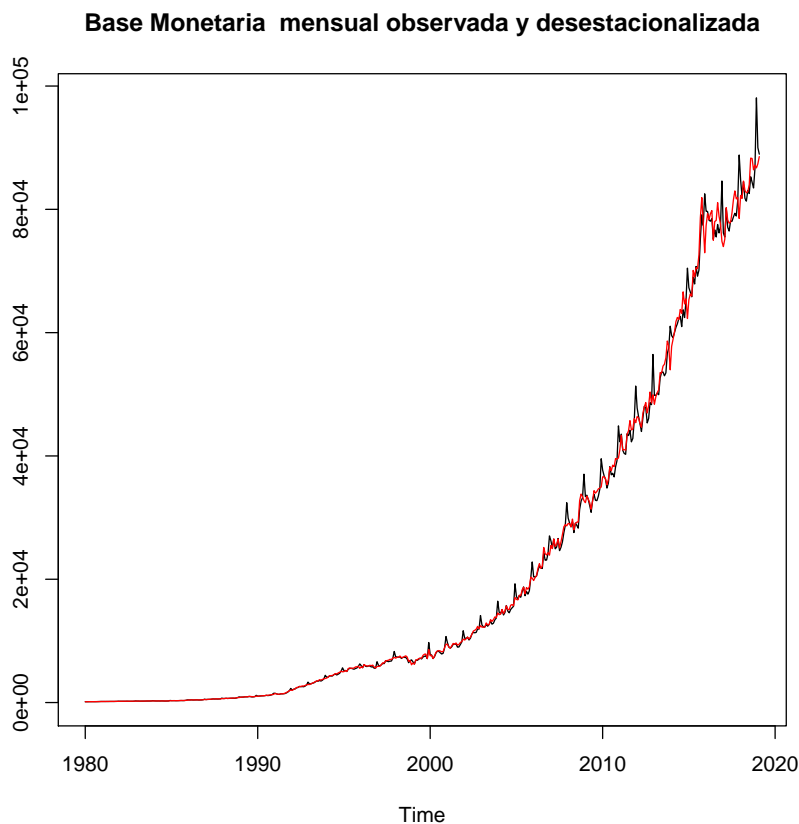
La serie desestacionalizada para este caso de una serie con descomposición multiplicativa sería

$$y_t / S_t \quad (2.10)$$

Lo cuál es equivalente a

```
Base.desestacionalizada <- Base / Base.componentes $seasonal

ts.plot(Base, Base.desestacionalizada,
  gpars = list(col = c("black", "red")),
  main="Base Monetaria mensual observada y desestacionalizada")
```



2.2.3. *Otros métodos de descomposición*

Existen muchos otros métodos para encontrar los componentes de una serie, pero tal vez los mas conocidos son: Descomposición X11 y "SEATS"(que corresponden a la sigla de extracción estacional en modelos ARIMA de series de tiempo que en inglés es Seasonal Extraction in ARIMA Time Series).

Estos métodos de descomposición son típicamente empleados para presentar datos estacionales y no se emplean para generar proyecciones. Como lo menciona Bell y Hillmer, 2002, la desestacionalización se realiza para simplificar los datos para que puedan ser interpretados más fácilmente por usuarios estadísticamente poco sofisticados sin una pérdida significativa de información.

A continuación se presenta una breve descripción de estos dos métodos de descomposición que típicamente se emplean principalmente para desestacionalizar y como implementarlos en R.

2.2.3.1. El método ARIMAX11

El método X11 es una técnica para desestacionalizar desarrollada por el Census Bureau (la oficina de Estadísticas del Gobierno de los Estados Unidos) en la década de los 60 (ver Shiskin, 1965). Este método tiene hoy en día muchas variantes como el método ARIMAX11, ARIMAX12 o ARIMAX13 que implica la combinación de la modelación ARIMA (ver Capítulo ??). Por ahora, concentrémonos en el método ARIMAX11.

Este método implica como primer paso estimar un modelo ARIMA (en el Capítulo ?? se discutirá esto en mayor detalle) y generar pronósticos para uno o tres años, dependiendo del tipo de serie.

Con la serie observada y la proyectada, se le aplican secuencias de promedios móviles centradas para encontrar cada uno de los componentes de la serie.

A manera de ejemplo, para una serie mensual (y_t) esto implica calcular una media móvil para estimar el componente tendencial y cíclico (\hat{T}_t):

$$\hat{T}_t = \frac{1}{13} \sum_{j=-6}^6 y_{t-j}. \quad (2.11)$$

Noten que los pronósticos de la serie calculada en el primer paso son empleados para poder calcular los promedios para todos los datos de la muestra, dado que el promedio móvil implica tanto datos pasados como futuros para cada observación. Si la serie es trimestral, entonces la tendencia se estima empleando el siguiente promedio móvil centrado

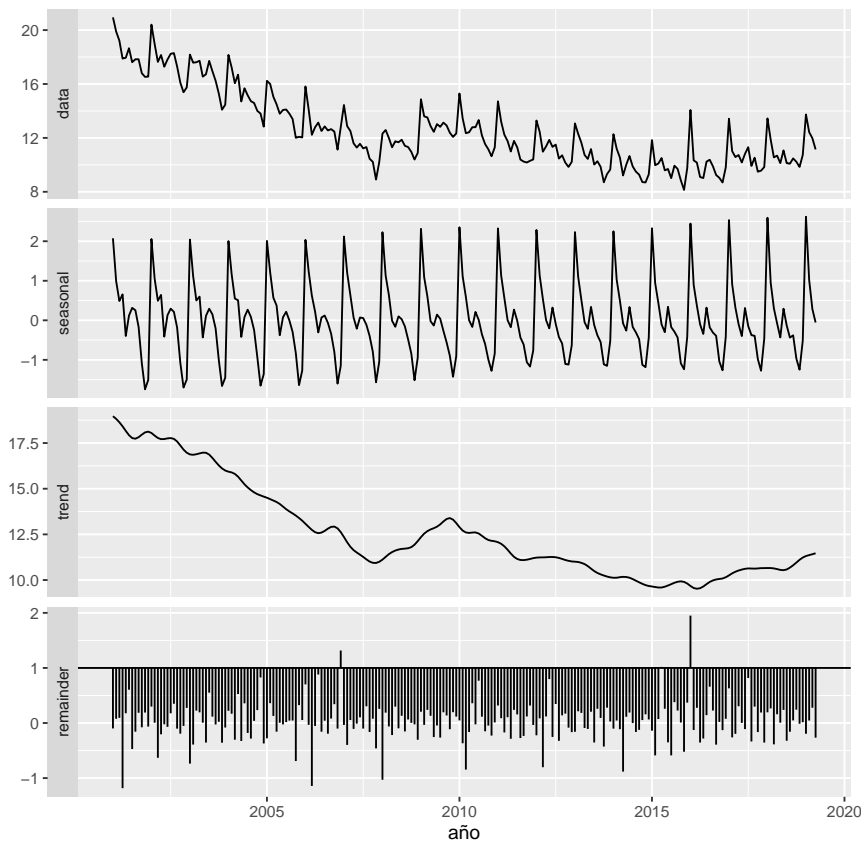
$$\hat{T}_t = \frac{1}{5} \sum_{j=-2}^2 y_{t-j}. \quad (2.12)$$

El siguiente paso es remover la tendencia estimada de la serie original para dejar el componente estacional y aleatorio de la serie. y continuar con medias móviles.

En R se puede realizar esta descomposición empleando la función *seas* del paquete **seasonal**. Esta función realiza estos pasos de manera automática. Y para graficar la descomposición podemos emplear la función *autoplot* del paquete **forecast**. Para realizar la descomposición de la serie de tasa de desempleo podemos emplear el siguiente código:

```
library(seasonal)
TD.componentes.ARIMAX11 <- seas(TD.ts, x11 = "")
library(forecast)
autoplot(TD.componentes.ARIMAX11, facet=T) + xlab("año") +
  ggtitle("Descomposición por el método ARIMAX11 de la tasa de desempleo.")
```

Descomposición por el método ARIMAX11 de la tasa de desempleo.



El paquete **seasonal** permite calcular cada uno de los componentes de la serie de manera muy fácil empleando las funciones *trendcycle*, *seasonal* y *remainder*. Estas funciones necesitan como argumento un objeto de clase *seas*. Y finalmente la serie desestacionalizada se puede obtener empleando la función *seasadj* del mismo paquete.

```
Tt = trendcycle(TD.componentes.ARIMAX11)
head(Tt)

##           Jan           Feb           Mar           Apr           May           Jun
## 2001 18.96990 18.82523 18.63874 18.41650 18.16498 17.92951

St= seasonal(TD.componentes.ARIMAX11)
head(St)

##           Jan           Feb           Mar           Apr           May
## 2001  2.0748862  0.9924369  0.4886943  0.6547792 -0.3965642
##
##           Jun
## 2001  0.1193586

Rt = remainder(TD.componentes.ARIMAX11)
```



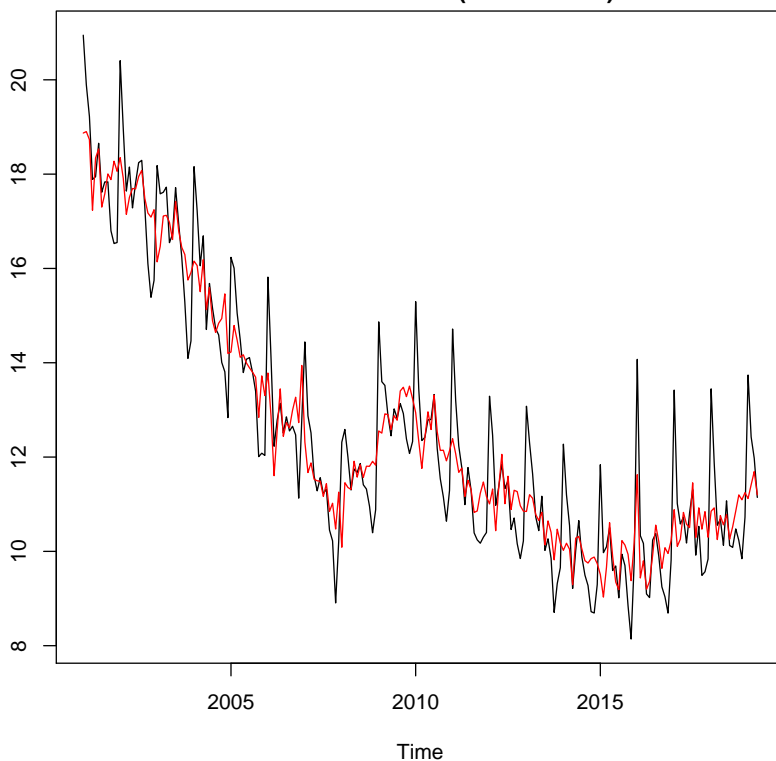
```
head(Rt)

##           Jan           Feb           Mar           Apr           May
## 2001 -0.09840257  0.07654424  0.09413559 -1.18270091  0.17723886
##           Jun
## 2001  0.60502781

TD.desestacionalizada.ARIMAX11 = seasadj(TD.componentes.ARIMAX11)

ts.plot(TD.ts, TD.desestacionalizada.ARIMAX11,
        gpars = list(col = c("black", "red")),
        main="Tasa de desempleo mensual observada y
desestacionalizada \n Método ARIMAX11 (13 ciudades)")
```

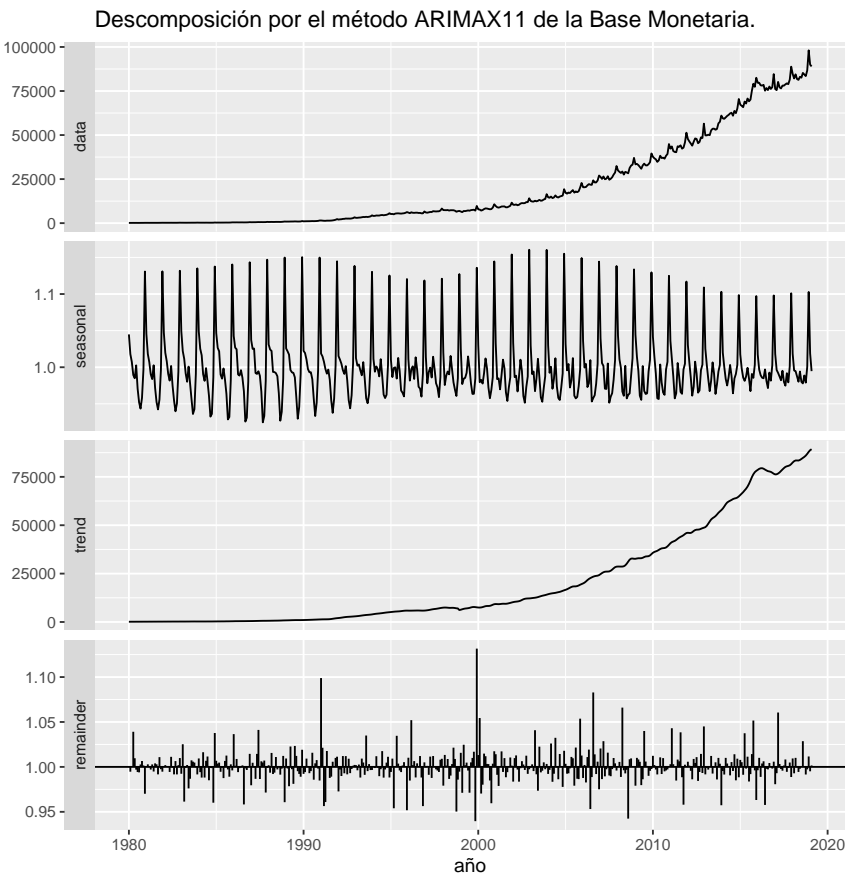
**Tasa de desempleo mensual observada y
desestacionalizada
Método ARIMAX11 (13 ciudades)**



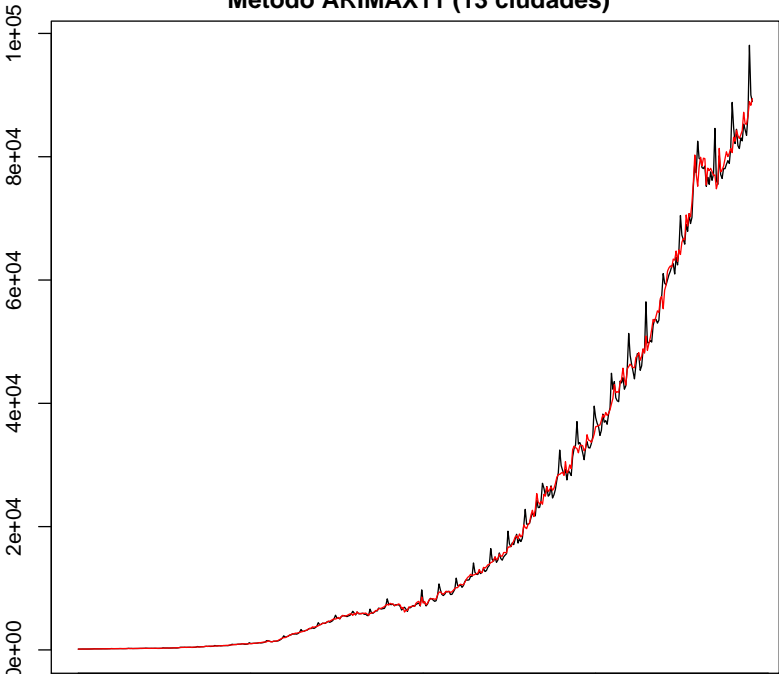
Ahora podemos realizar el mismo ejercicio con la serie de la Base Monetaria.

```
Base.componentes.ARIMAX11 <- seas(Base, x11 = "")
autoplot(Base.componentes.ARIMAX11, facet=T) + xlab("año") +
  ggtitle("Descomposición por el método ARIMAX11 de la Base Monetaria.")
Base.desestacionalizada.ARIMAX11 = seasadj(Base.componentes.ARIMAX11)
```

```
ts.plot(Base, Base.desestacionalizada.ARIMAX11,  
        gpars = list(col = c("black", "red")),  
        main="Base monetaria mensual observada y  
desestacionalizada \n Método ARIMAX11 (13 ciudades)")
```



**Base monetaria mensual observada y
desestacionalizada
Método ARIMAX11 (13 ciudades)**



2.2.3.2. SEATS

Este método fue desarrollado por el Banco Central de España. Este procedimiento extiende lo propuesto por el ARMIAX11 al incluir un mecanismo de extracción del componente estacional en modelos ARIMA. La explicación del detalle de este método está por fuera del alcance de este libro (para ver una descripción de esta aproximación se puede consultar Dagum y Bianconcini, 2016.)

Este procedimiento para descomponer series de tiempo solo funciona con datos trimestrales y mensuales. Por lo tanto, la estacionalidad de otros tipos, como los datos diarios, los datos por hora o los datos semanales, requieren un método diferente.

El procedimiento se implementa fácilmente con la función *seas* del paquete **seasonal**.

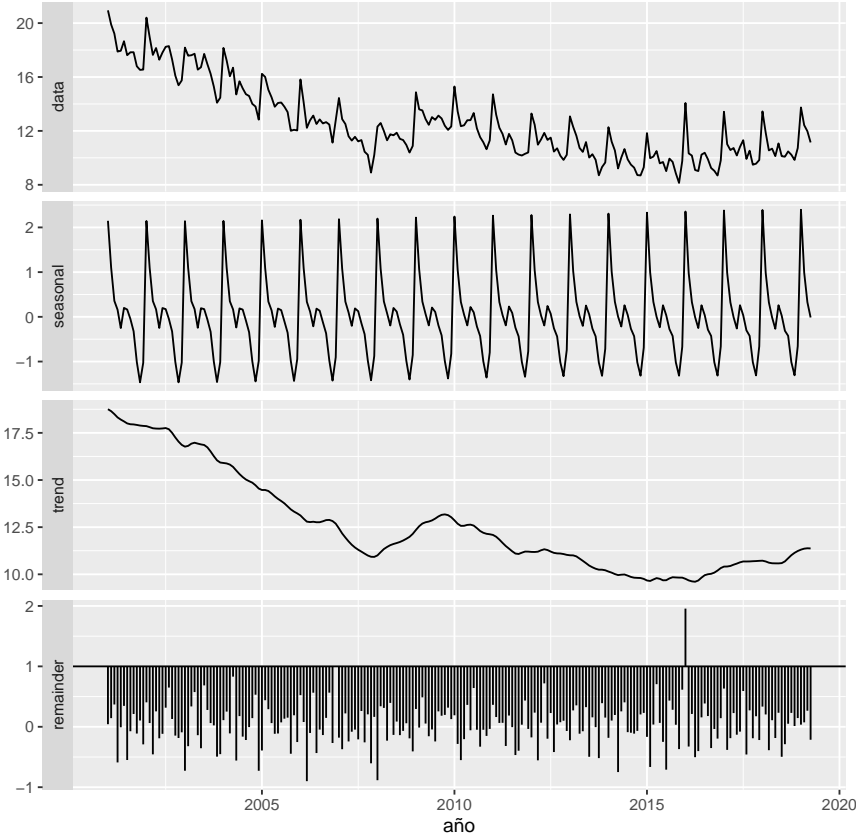
```
TD.componentes.SEATS <- seas(TD.ts)

autoplot(TD.componentes.SEATS, facet=T) + xlab("año") +
  ggtitle("Descomposición por el método SEATS de la tasa de desempleo.")

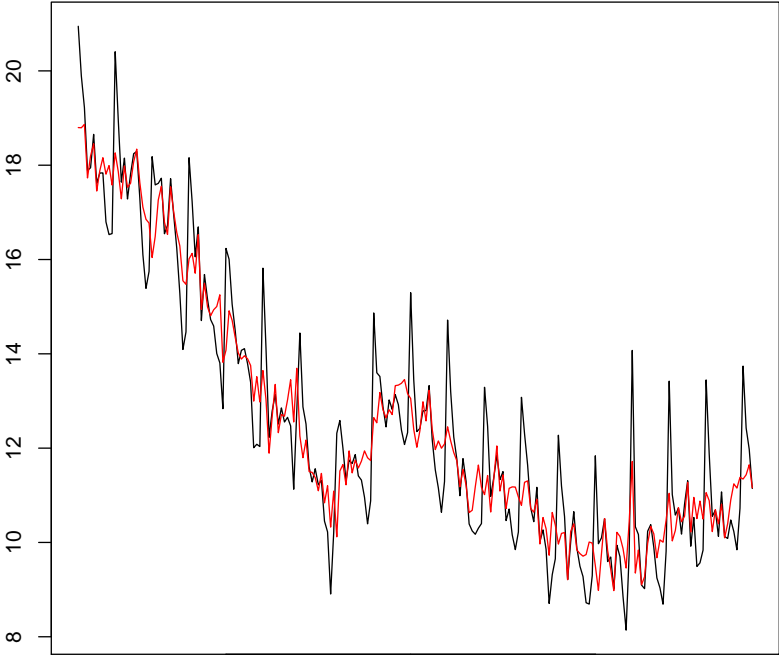
TD.desestacionalizada.SEATS = seasadj(TD.componentes.SEATS)

ts.plot(TD.ts, TD.desestacionalizada.SEATS,
  gpars = list(col = c("black", "red")),
  main="Tasa de desempleo mensual observada y
  desestacionalizada \n Método SEATS (13 ciudades)")
```

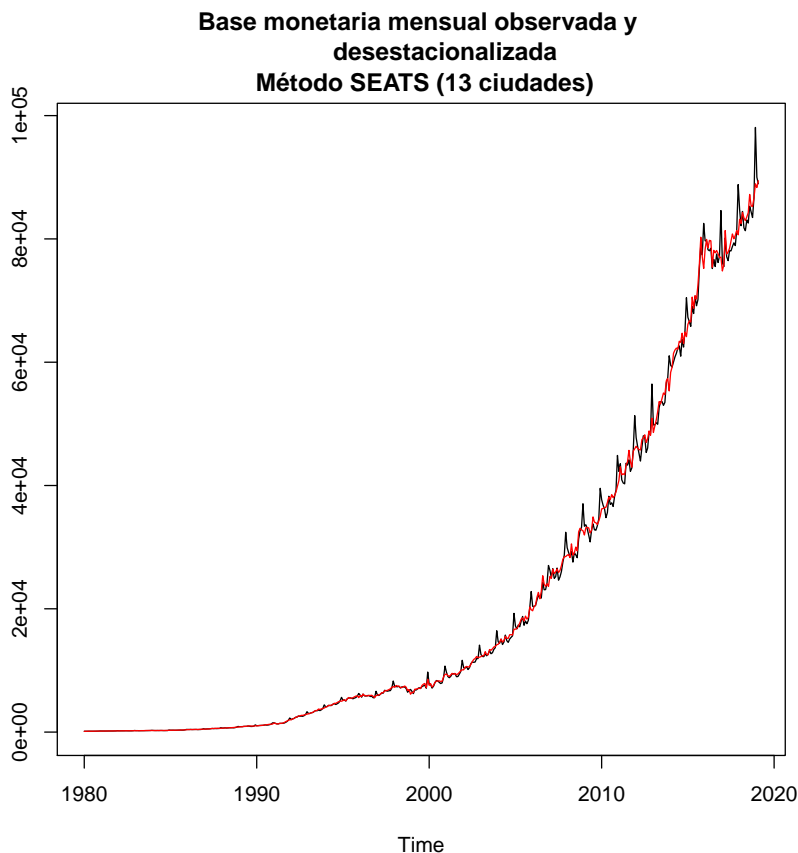
Descomposición por el método SEATS de la tasa de desempleo.



Tasa de desempleo mensual observada y
desestacionalizada
Método SEATS (13 ciudades)



Ustedes pueden desarrollar el mismo procedimiento para la Base monetaria para obtener el siguiente gráfico.



2.2.3.3. Comentarios sobre los métodos de descomposición.

Hemos estudiado tres métodos para descomponer una serie y si es del caso desestacionalizarla. La pregunta natural sería, ¿cuál aproximación emplear? o ¿cómo escoger entre estas aproximaciones? La verdad no existe una buena forma de responder esta pregunta. Como veremos en el resto de este libro, la mejor aproximación será poder modelar el DGP de la mejor manera de tal manera que podamos entender cómo se comporta la serie. Este tipo de análisis se puede entender como una forma de describir la serie y por tanto cercano a la estadística descriptiva o analítica descriptiva. Típicamente estas descomposiciones no se emplean para generar pronósticos sino para simplificar las series. En últimas, la desestacionalización se realiza para simplificar los datos para que puedan ser interpretados más fácil-

mente por usuarios estadísticamente poco sofisticados sin una pérdida significativa de información (ver Bell y Hillmer, 2002).

2.3. Ejercicios

Empleando la información del número de ocupados en miles de personas (Ocupados), para las 13 principales ciudades de Colombia, los desocupados (Desocupados) y los inactivos (Inactivos) que se encuentra en el archivo `datosEmpleo.xlsx`, descomponga cada una de las series en sus componentes.

Capítulo 3

Métodos de suavizamiento

En este capítulo estudiaremos cuatro métodos que se emplean en las series de tiempo para pronosticar series de tiempo. Los métodos que emplearemos en este capítulo emplean relativamente poca información para suavizar el comportamiento de la serie, de ahí su nombre. Estos métodos son:

- Promedio Móvil
- Suavización Exponencial Simple (SES)
- Suavización Exponencial Lineal (Holt)
- Suavización Exponencial Lineal de Winters (Holt-Winters)

En este capítulo emplearemos la información de la tasa de desempleo mensual de las 13 principales ciudades en Colombia (TD_13ciudades) que se encuentra disponible en el archivo `datosEmpleo.xlsx` que fue empleado en el Capítulo 2. Cargue los datos y guárdelos en un objeto de clase *ts* con el nombre *D.ts*.

3.1. Muestra de estimación y de evaluación

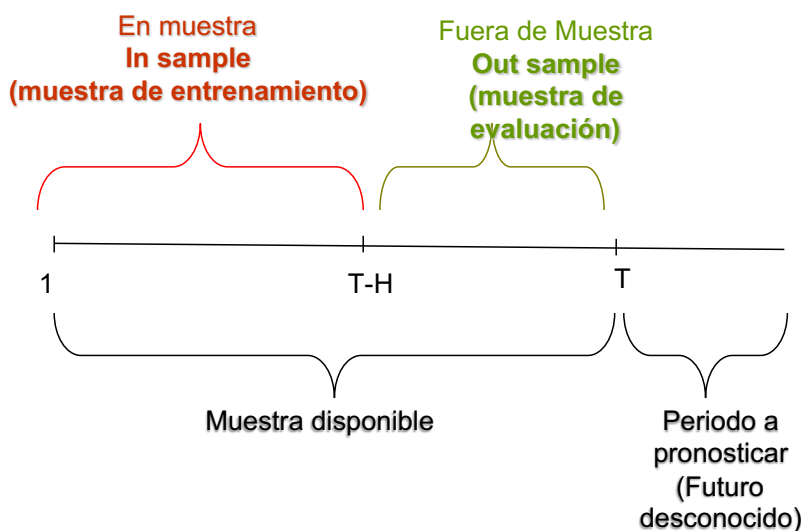
Antes de discutir los métodos de suavizamiento para pronosticar es importante discutir un hecho importante. Dado que los pronósticos son extrapolaciones, la forma ideal de evaluar un modelo de pronóstico es mirando su capacidad para extrapolar. En otras palabras, a diferencia de modelos de regresión múltiple donde es importante el ajuste del modelo a la muestra con la que se estima el modelo, en este tipo de modelos será más importante la capacidad de los modelos de generar buenos pronósticos.

Ahora supongamos que contamos con una muestra de tamaño T y queremos generar pronósticos para esta serie. Dada la discusión anterior, lo ideal sería contar con información del futuro (datos después del periodo T) (ver Figura 3.1). Esto nos lleva a un problema práctico, o generamos pronósticos del futuro (más allá del período T) y esperamos a que pase el tiempo para poder evaluar los pronósticos

al compararlos con valores realmente observados o hacemos algún ejercicio con la muestra de tamaño T . Claramente la primera opción no es práctica.

Cómo se puede observar la Figura 3.1 una opción es guardar las últimas H observaciones y trabajar con $T - H$ datos para estimar el modelo. A esta muestra (desde $t = 1$ hasta $t = T$) la llamaremos el pedido "in-sample", en muestra o muestra entrenamiento. Los restantes H (últimos) períodos los denominaremos "out-of-sample", fuera de muestra o muestra evaluación. Con esta muestra de H observaciones se evaluarán el comportamiento de los modelos para proyectar. Una vez escogido el mejor modelo, volveremos a usar toda la muestra disponible (hasta la observación T) para re-estimar el modelo y generar los pronósticos que se requieran para el futuro (ver Figura 3.1).

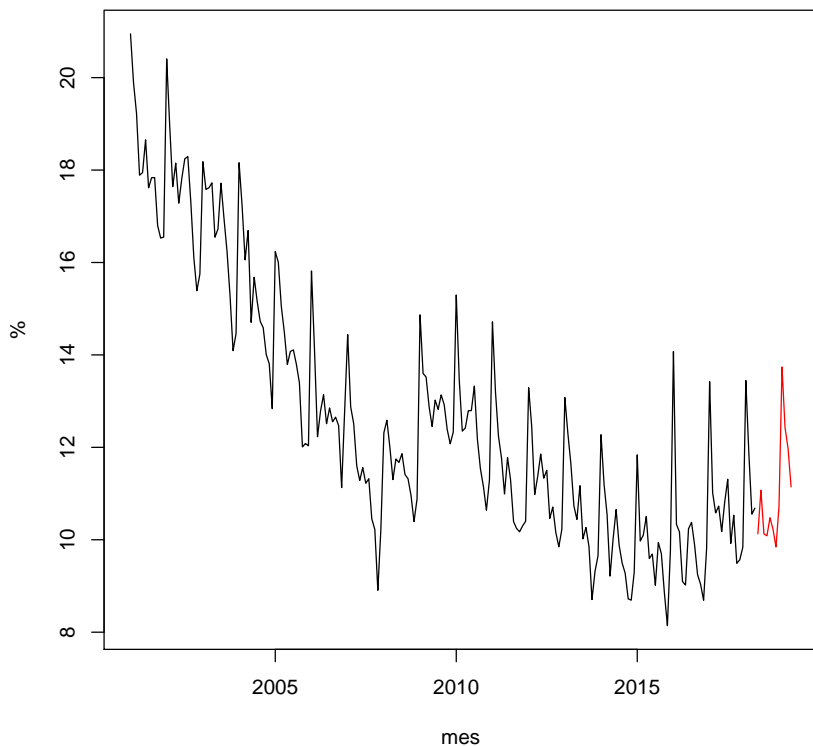
Figura 3.1 Preparación de la muestra



En otras palabras, es importante guardar una parte de la muestra para evaluar el comportamiento de los modelos por fuera de muestra (out-of-sample). Guardemos un año de datos empleando la función *window*.

```
TD.eval <- window(TD.ts, start= c(2018, 5))
TD.in.sample <- window(TD.ts, end= c(2018, 4))
```

Ahora la pregunta es cómo determinar cuál de los modelos que se calcule es mejor. Una forma de hacer esto es mirar el error que comete cada modelo para un periodo determinado. Es decir si y_t es el valor real de la serie para cualquier periodo t y \hat{y}_t es el valor pronosticado para el periodo t , entonces el error de pronóstico para un periodo será $e_t = y_t - \hat{y}_t$. Como estamos pronosticando varios periodos una forma natural de tener en cuenta todos los errores sería sumarlos y dividirlos por

Figura 3.2 Tasa de desempleo mensual muestra de estimación y evaluación (13 ciudades)”

el número de pronósticos (error promedio). Un problema de esta aproximación es que los errores positivos y negativos se pueden cancelar entre sí, y que los errores grandes y pequeños están teniendo el mismo peso. Una opción para resolver esto es considerar el promedio de los residuos al cuadrado¹. Normalmente, se acostumbra sacarle la raíz cuadrada a este promedio para regresar la suma a la unidad original de los datos. En otras palabras, una buena idea es emplear el la raíz cuadrada del error promedio al cuadrado o *RMSE* (por la sigla del inglés Root Mean Square Error). El *RMSE* viene dado por:

$$RMSE = \sqrt{\frac{\sum_{h=1}^H (y_{t+h} - \hat{y}_{t+h})^2}{H}} \quad (3.1)$$

¹ Error promedio al cuadrado o MSE en inglés por Mean Square Error (*MSE*)

En las siguientes secciones discutiremos los diferentes modelos de suavización y los evaluaremos fuera de muestra.

3.2. Promedio Móvil

El promedio móvil es tal vez el método mas sencillo para proyectar la serie por medio del comportamiento en el "vecindario" de la serie. Este método ignora los componentes de una serie de tiempo y suaviza el comportamiento de la serie ignorando posibles tendencias, componentes estacionales y cíclico. El pronóstico para el periodo t y (y_t) está dado por:

$$\hat{y}_{T+1} = \frac{1}{m} \sum_{j=-k}^k y_{t-j} \quad (3.2)$$

donde \hat{y}_{T+1} corresponde al valor suavizado para el periodo t . $m = 2k + 1$ es el orden del promedio móvil (número de observaciones que se incluyen en el promedio móvil) y por tanto $k = (m - 1)/2$

Esto lo podemos hacer fácilmente con la función *ma* del paquete **forecast**. Los argumentos de esta función son el objeto que contiene una serie de clase *ts* y el orden (*order*).

Calculemos el promedio móvil de orden 3, 4, 5, y 6 de la siguiente manera:

```
library(forecast)
ma.3 <- ma(TD.in.sample, order = 3)
ma.4 <- ma(TD.in.sample, order = 4)
ma.5 <- ma(TD.in.sample, order = 5)
ma.6 <- ma(TD.in.sample, order = 6)
```

Cuando se especifica un orden par, las observaciones promediadas incluirán una observación más del futuro que del pasado (k se redondea hacia arriba).

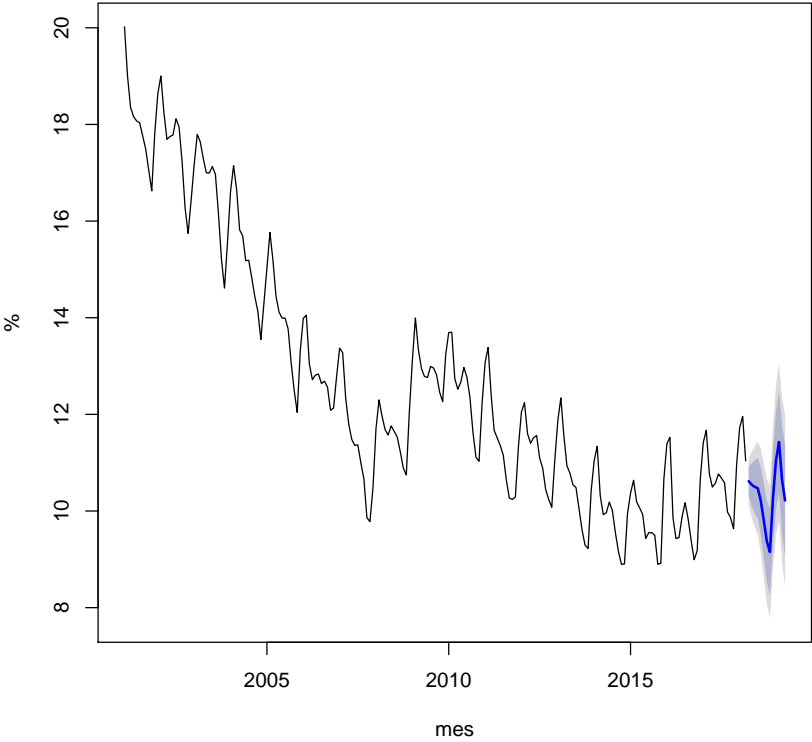
Ahora, generemos los pronósticos para el año que guardamos los datos. Para esto podemos emplear la función *forecast* del paquete del mismo nombre. Esta función solo necesita que se incluya el objeto que contiene el modelo desde el que se quiere hacer el pronóstico y el número de pronósticos que se quieren hacer (también conocido como el horizonte) (h). En este caso,

```
ma.3.f <- forecast(ma.3, h=13)
ma.3.f
plot(ma.3.f)
```

Figura 3.3 Pronóstico de la tasa de desempleo mensual (13 ciudades) empleando promedio móvil de orden 3

Warning in ets(object, lambda = lambda, biasadj = biasadj, allow.multiplicative.trend = allow.multiplicative.trend, : Missing values encountered. Using longest contiguous portion of time series

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## Apr 2018	10.616564	10.303084	10.93004	10.137138	11.09599
## May 2018	10.544038	10.099719	10.98836	9.864512	11.22356
## Jun 2018	10.499162	9.953751	11.04457	9.665028	11.33330
## Jul 2018	10.472309	9.841091	11.10353	9.506944	11.43767
## Aug 2018	10.205942	9.498614	10.91327	9.124176	11.28771
## Sep 2018	9.795148	9.018547	10.57175	8.607439	10.98286
## Oct 2018	9.368072	8.527343	10.20880	8.082288	10.65386
## Nov 2018	9.151304	8.250489	10.05212	7.773627	10.52898
## Dec 2018	10.258826	9.301204	11.21645	8.794269	11.72338
## Jan 2019	11.048374	10.036669	12.06008	9.501104	12.59564
## Feb 2019	11.429372	10.365891	12.49285	9.802918	13.05583
## Mar 2019	10.639011	9.525737	11.75228	8.936405	12.34162
## Apr 2019	10.218048	9.056698	11.37940	8.441916	11.99418



La Figura 3.3 permite observar los pronósticos y sus respectivos intervalos de confianza con un 95 % de confianza. El desempeño del pronósticos fuera de muestra se puede medir con el *RMSE*, que se calcula por medio de la función *accuracy* del paquete *forecast*. Esta última función necesita como argumentos la serie de pronósticos y el valor real. En nuestro caso esto implica:

```
accuracy(ma.3.f, TD.eval)

##                ME      RMSE      MAE      MPE      MAPE
## Training set 0.004128339 0.2349185 0.1870452 0.07553684 1.534388
## Test set    0.695433170 1.0576037 0.8425556 5.82209977 7.275654
##                MASE      ACF1 Theil's U
## Training set 0.2223235 0.3011947      NA
## Test set    1.0014690 0.2120179 0.9433586
```

Finalmente, calculemos los pronósticos para cada una de los ordenes considerados y sus respectivos *RMSE*.

```
ma.4.f <- forecast(ma.4, h=14)
ma.5.f <- forecast(ma.5, h=14)
ma.6.f <- forecast(ma.6, h=15)
RMSE.3 <- accuracy(ma.3.f, TD.eval)[ "Test set", "RMSE"]
RMSE.4 <- accuracy(ma.4.f, TD.eval)[ "Test set", "RMSE"]
RMSE.5 <- accuracy(ma.5.f, TD.eval)[ "Test set", "RMSE"]
RMSE.6 <- accuracy(ma.6.f, TD.eval)[ "Test set", "RMSE"]
tabla <- matrix(c(RMSE.3, RMSE.4, RMSE.5, RMSE.6), 4, 1)
colnames(tabla) <- "RMSE"
names <- c("h=3", "h=4", "h=5", "h=6")
tabla <- data.frame(tabla)
row.names(tabla) <- c("h=3", "h=4", "h=5", "h=6")
```

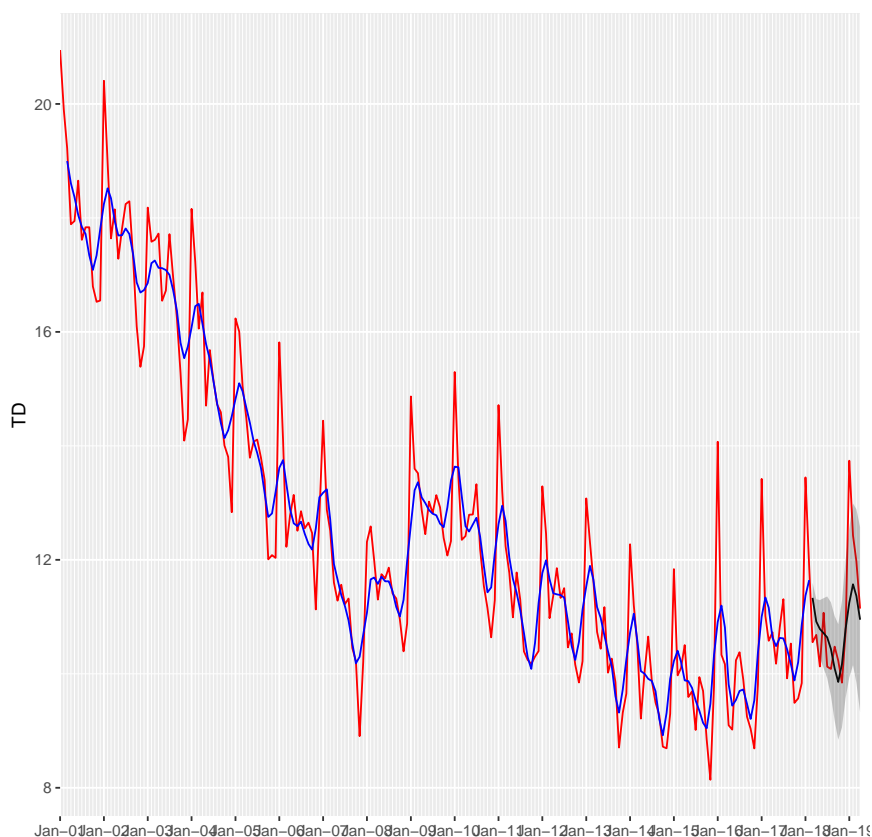
Cuadro 3.1 RMSE para 12 pronósticos empleando promedios móviles de diferentes ore-
nes (h)

	RMSE
h=3	1.06
h=4	0.85
h=5	0.93
h=6	0.96

En este caso encontramos que el mínimo *RMSE* corresponde a h=4. El pronóstico de ese modelo lo encontramos en el siguiente gráfico.

Figura 3.4 Pronóstico para el último año de la TD con promedio móvil ($h=4$) (13 ciudades)
(negro=pronóstico, azul=estimado, rojo=datos, sombra=95 % conf.)

```
## Warning: Removed 16 rows containing missing values (geom_path).
## Warning: Removed 206 rows containing missing values (geom_path).
```



3.3. Suavización Exponencial Simple (SES)

Los métodos de suavizamiento exponencial se desarrollaron durante la década de 1950 (ver por ejemplo Brown, 1959). Los pronósticos producidos utilizando métodos de suavizado exponencial **son promedios ponderados de observaciones pasadas**, algo similar al promedio móvil. La diferencia de estos métodos es que las observaciones son ponderadas de manera diferente. En este caso **los pesos** que se le asignan a los observaciones **van decayendo exponencialmente a medida que las observaciones envejecen (se alejan en el tiempo)**. En otras palabras, **cuanto más reciente sea la observación, mayor será el peso asignado**.

Formalmente,

$$\begin{aligned} \text{Ecuación de pronóstico} \quad & \hat{y}_{t+h|t} = \ell_t \\ \text{Ecuación de suavizamiento} \quad & \ell_t = \alpha y_t + (1 - \alpha) \ell_{t-1}, \end{aligned}$$

El paquete *forecast* tiene la función *ses* que estima directamente α minimizando el SSE. Es decir, esta función encuentra α minimizando la siguiente expresión:

$$\text{SSE} = \sum_{t=1}^T (y_t - \hat{y}_t)^2 = \sum_{t=1}^T e_t^2 \quad (3.3)$$

esto equivale a minimizar *in-sample* el RMSE.

Esta función necesita como argumentos la serie (objeto *ts*) y el horizonte (*h* a pronosticar).

```
SES.TD <- ses(TD.in.sample, h = 12)
summary(SES.TD)

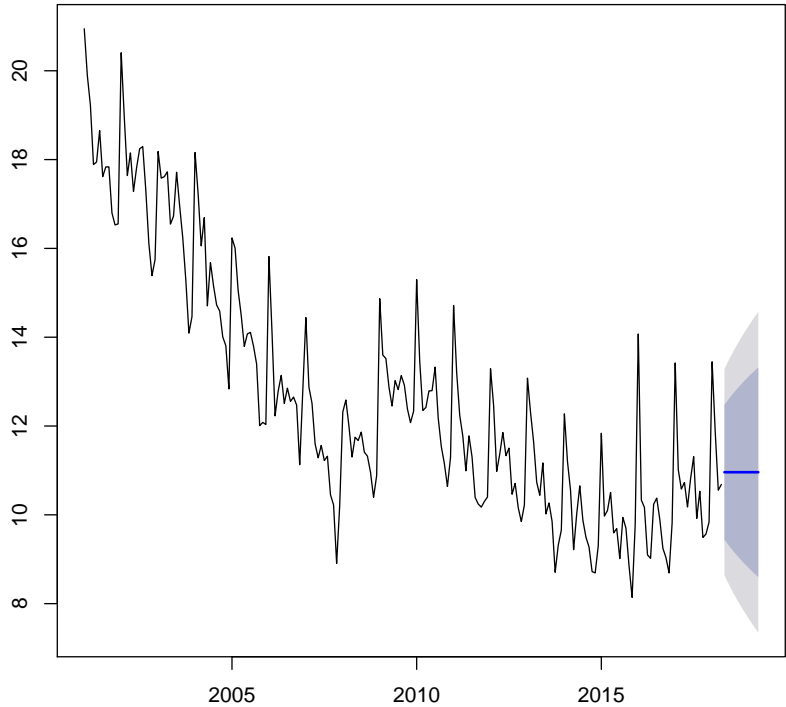
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
## ses(y = TD.in.sample, h = 12)
##
## Smoothing parameters:
##   alpha = 0.3597
##
## Initial states:
##   l = 19.6602
##
##   sigma: 1.184
##
##      AIC      AICc      BIC
## 1184.454 1184.571 1194.466
##
## Error measures:
##              ME      RMSE      MAE      MPE      MAPE
## Training set -0.1163002 1.178276 0.8458051 -1.50297 6.690792
##              MASE      ACF1
## Training set 0.9125669 0.2472514
##
## Forecasts:
##      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
## May 2018      10.95979 9.442452 12.47712 8.639224 13.28035
## Jun 2018      10.95979 9.347296 12.57228 8.493696 13.42588
## Jul 2018      10.95979 9.257451 12.66212 8.356289 13.56328
## Aug 2018      10.95979 9.172115 12.74746 8.225779 13.69379
```



```
## Sep 2018      10.95979  9.090672 12.82890  8.101222 13.81835
## Oct 2018      10.95979  9.012632 12.90694  7.981870 13.93770
## Nov 2018      10.95979  8.937601 12.98197  7.867121 14.05245
## Dec 2018      10.95979  8.865257 13.05432  7.756480 14.16309
## Jan 2019      10.95979  8.795329 13.12424  7.649535 14.27004
## Feb 2019      10.95979  8.727591 13.19198  7.545938 14.37363
## Mar 2019      10.95979  8.661848 13.25772  7.445394 14.47418
## Apr 2019      10.95979  8.597935 13.32164  7.347647 14.57193

plot(SES.TD)
```

Forecasts from Simple exponential smoothing



En esta caso el α estimado es 0.3596639. Y el *RMSE* en la muestra de evaluación es

```
RMSE.SES <- accuracy(SES.TD, TD.eval)[ "Test set", "RMSE" ]
RMSE.SES
## [1] 1.124422
RMSE.4
## [1] 0.8511266
```

Noten que este *RMSE* es mas grande que el del modelo de media móvil con $h=4$.

3.4. Suavización Exponencial Lineal (Holt)

En la sección anterior vimos como el pronóstico con el método de SES se aplana rápidamente; es decir, no tenía en cuenta la inercia de la serie que se encuentra en su tendencia. Holt, 1957 observó este problema del método de suavizado exponencial simple y lo extendió para **incluir una tendencia**. El propuso una ecuación de pronóstico y dos ecuaciones de suavizado (una para el nivel y otra para la tendencia):

Ec. de pronóstico	$\hat{y}_{t+h t} = \ell_t + hb_t$
Ec. de nivel	$\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1})$
Ecuación de tendencia	$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1},$

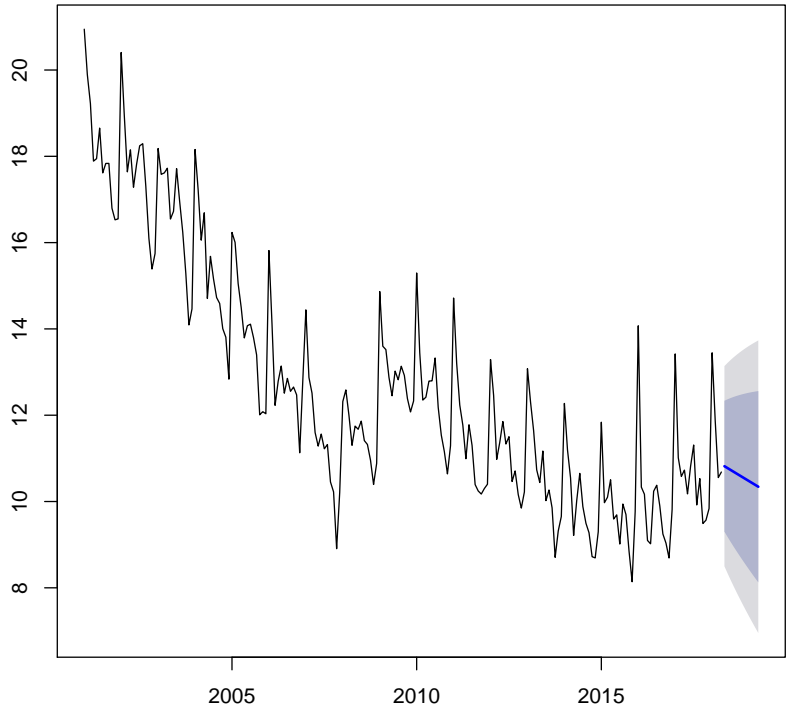
En este caso se tienen dos parámetros α y β . Al igual que en el caso de la suavización exponencial simple, el paquete *forecast* cuenta con la función *holt* que encuentra los valores de estos dos parámetros que minimiza el *SSE* "in-sample". La función tiene los mismo argumentos que la función *ses*.

```
HOLT.TD <- holt(TD.in.sample, h = 12)
summary(HOLT.TD)

##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
## holt(y = TD.in.sample, h = 12)
##
## Smoothing parameters:
##   alpha = 0.3207
##   beta  = 1e-04
##
## Initial states:
##   l = 20.142
##   b = -0.0432
##
## sigma: 1.184
##
##      AIC      AICc      BIC
## 1186.424 1186.721 1203.111
##
## Error measures:
##               ME      RMSE      MAE      MPE      MAPE
## Training set -0.001692839 1.17254 0.8198668 -0.5300256 6.422868
```

```
##                               MASE      ACF1
## Training set 0.8845813 0.2740671
##
## Forecasts:
##      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
## May 2018      10.81807 9.300744 12.33541 8.497517 13.13863
## Jun 2018      10.77484 9.181355 12.36833 8.337813 13.21188
## Jul 2018      10.73161 9.065399 12.39783 8.183358 13.27987
## Aug 2018      10.68838 8.952445 12.42432 8.033495 13.34327
## Sep 2018      10.64515 8.842144 12.44816 7.887689 13.40262
## Oct 2018      10.60192 8.734211 12.46963 7.745504 13.45834
## Nov 2018      10.55869 8.628407 12.48898 7.606576 13.51081
## Dec 2018      10.51546 8.524531 12.50639 7.470597 13.56033
## Jan 2019      10.47223 8.422413 12.52205 7.337305 13.60716
## Feb 2019      10.42900 8.321905 12.53609 7.206476 13.65152
## Mar 2019      10.38577 8.222879 12.54866 7.077914 13.69362
## Apr 2019      10.34254 8.125223 12.55985 6.951447 13.73363
plot(HOLT.TD)
```

Forecasts from Holt's method



En esta caso el α estimado es 0.3206864 y el β estimado es $1,0000013 \times 10^{-4}$. Y el RMSE en la muestra de evaluación es

```
RMSE.HOLT <- accuracy(HOLT.TD, TD.eval)["Test set", "RMSE"]  
RMSE.HOLT  
  
## [1] 1.286171  
  
RMSE.4  
  
## [1] 0.8511266
```

Aún sigue siendo mejor los pronósticos basados en el modelo de media móvil con $h=4$.

3.5. Suavización Exponencial Lineal de Winters (Holt-Winters)

Holt, 1957 y Winters, 1960 extendieron el método de Holt para capturar la **estacionalidad**. El método estacional de Holt-Winters comprende la ecuación de pronóstico y tres ecuaciones de suavizado, una para el **nivel** (ℓ), una para la **tendencia** (b_t) y uno para el componente **estacional** (s_t). Esto implicará un nuevo parámetro asociado a la estacionalidad (γ).

Existen dos opciones para este método que difieren en la naturaleza del componente estacional: Holt-Winters aditivo y Holt-Winters multiplicativo. El método aditivo se recomienda cuando las variaciones estacionales son aproximadamente constantes a lo largo de la serie, mientras que el método multiplicativo se emplea cuando las variaciones estacionales cambian proporcionalmente al nivel de la serie.

Con el método aditivo, el componente estacional se expresa en términos absolutos en la escala de la serie observada, y en la ecuación de nivel, la serie se ajusta estacionalmente restando el componente estacional. Dentro de cada año, el componente estacional se sumará aproximadamente a cero. Con el método multiplicativo, el componente estacional se expresa en términos relativos (porcentajes), y la serie se ajusta estacionalmente dividiendo por el componente estacional. Dentro de cada año, el componente estacional sumará aproximadamente.

3.5.1. *Holt-Winters aditivo*

Para el método aditivo, tenemos que:

$$\begin{aligned}\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\ \ell_t &= \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} \\ s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m},\end{aligned}$$

donde k es la parte entera de $(h - 1)/m$. Esto garantiza que las estimaciones de los índices estacionales utilizados para el pronóstico provienen del último año de la muestra. La ecuación de nivel muestra un promedio ponderado entre la observación ajustada estacionalmente $(y_t - s_{t-m})$ y el pronóstico no estacional $\ell_{t-1} + b_{t-1}$, para el periodo t .

La ecuación de tendencia es idéntica al método lineal de Holt. La ecuación estacional muestra un promedio ponderado entre el índice estacional del periodo t $((y_t - \ell_{t-1} - bt - 1))$ y el índice estacional de la misma temporada del año pasado; es decir, m periodos atrás de tiempo (donde $m = 12$ si los datos son mensuales o $m = 4$ si los datos son trimestrales).

3.5.2. *Holt-Winters multiplicativo*

Para el caso multiplicativo, los componentes se definen como:

$$\begin{aligned}\hat{y}_{t+h|t} &= (\ell_t + hb_t)s_{t+h-m(k+1)} \\ \ell_t &= \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} \\ s_t &= \gamma \frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1 - \gamma)s_{t-m}\end{aligned}$$

3.5.3. *Holt-Winters en R*

En la práctica es común que independientemente de cómo se comporte la estacionalidad, se estimen los dos tipos de modelo. Esto lo podemos hacer con la función `hw` del paquete `forecast`. Los argumentos de esta función son similares a las funciones `ses` y `holt`. En este caso tenemos un tercer argumento que corresponde a `seasonal`, el cual permite escoger entre los dos métodos. Es decir las opciones son `seasonal = "additive"` para el modelo aditivo o `seasonal = "multiplicative"` para el modelo multiplicativo. En este caso, la función estimará los tres parámetros (α , β y γ) minimizando la *SSE* in-sample ”.

```
HOLT_WINTERS.AD.TD <- hw(TD.in.sample, h = 12, seasonal = "additive")
summary(HOLT_WINTERS.AD.TD)

##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
```

```
## Call:
## hw(y = TD.in.sample, h = 12, seasonal = "additive")
##
## Smoothing parameters:
##   alpha = 0.4732
##   beta  = 1e-04
##   gamma = 1e-04
##
## Initial states:
##   l = 18.6055
##   b = -0.0391
##   s = -0.8491 -1.4197 -1.0517 -0.4489 -0.1765 0.1199
##       0.2072 -0.2187 0.1131 0.3628 1.0584 2.3031
##
## sigma: 0.5566
##
##      AIC      AICc      BIC
## 883.8571 887.0782 940.5953
##
## Error measures:
##               ME      RMSE      MAE      MPE
## Training set 0.001513283 0.5348103 0.4223262 -0.02378379
##               MAPE      MASE      ACF1
## Training set 3.449983 0.4556616 0.01322643
##
## Forecasts:
##      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
## May 2018      10.260177  9.546804 10.973551  9.169167 11.35119
## Jun 2018      10.646851  9.857607 11.436096  9.439806 11.85390
## Jul 2018      10.520526  9.662056 11.378996  9.207610 11.83344
## Aug 2018      10.184932  9.262385 11.107479  8.774018 11.59585
## Sep 2018       9.873547  8.891064 10.856030  8.370969 11.37612
## Oct 2018       9.231573  8.192578 10.270568  7.642567 10.82058
## Nov 2018       8.824439  7.731825  9.917054  7.153429 10.49545
## Dec 2018       9.355903  8.212154 10.499651  7.606690 11.10512
## Jan 2019      12.469123 11.276406 13.661840 10.645020 14.29323
## Feb 2019      11.185147  9.945371 12.424924  9.289073 13.08122
## Mar 2019      10.450456  9.165320 11.735592  8.485011 12.41590
## Apr 2019      10.161679  8.832709 11.490650  8.129195 12.19416

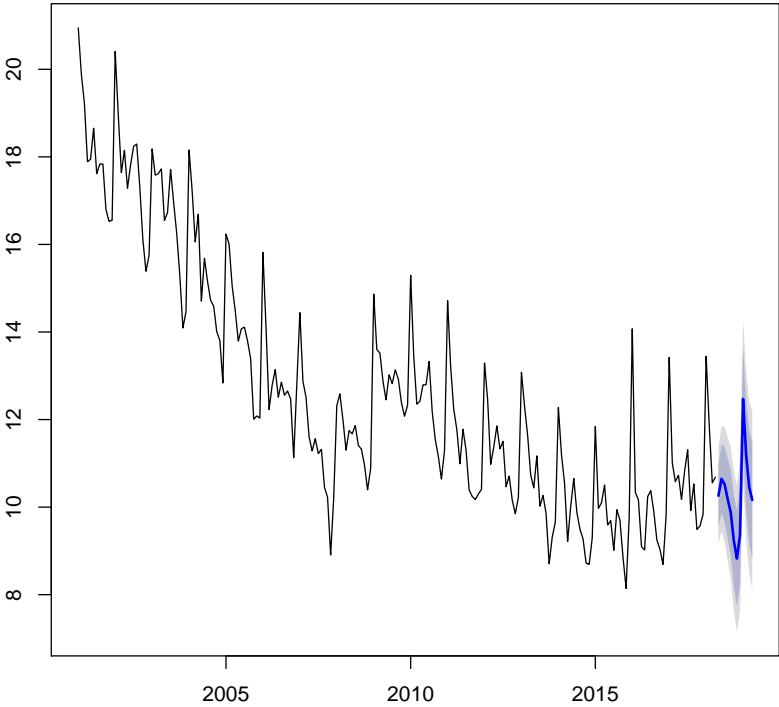
plot(HOLT_WINTERS.AD.TD)

HOLT_WINTERS.MUL.TD <- hw(TD.in.sample, h = 12, seasonal = "multiplicative")
summary(HOLT_WINTERS.MUL.TD)

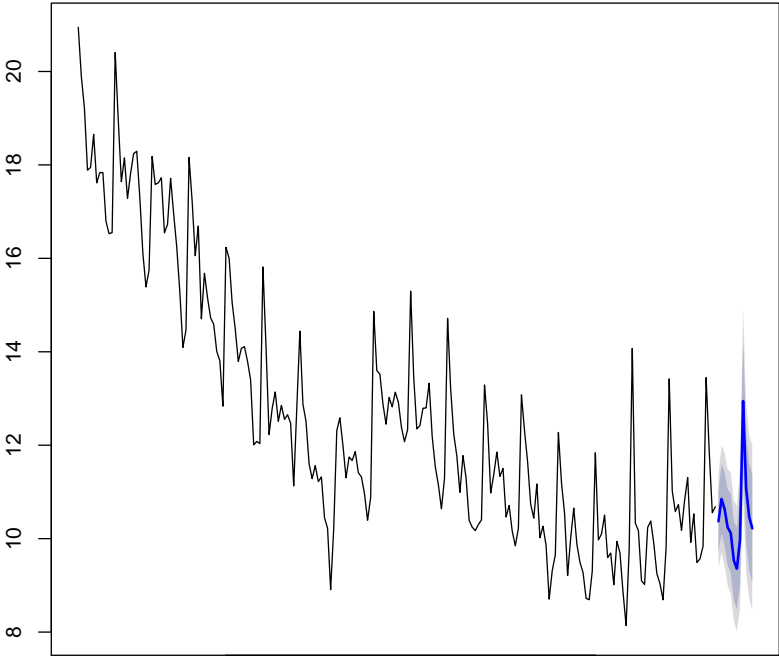
##
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
## hw(y = TD.in.sample, h = 12, seasonal = "multiplicative")
##
```

```
## Smoothing parameters:
##   alpha = 0.4383
##   beta  = 1e-04
##   gamma = 0.1404
##
## Initial states:
##   l = 18.2989
##   b = -0.0312
##   s = 0.9277 0.914 0.9376 0.9816 1.0021 1.0114
##       1.01 0.9736 1.0095 1.0172 1.0671 1.1482
##
##   sigma: 0.0493
##
##       AIC      AICc      BIC
## 923.3945 926.6156 980.1327
##
## Error measures:
##
##           ME      RMSE      MAE      MPE
## Training set -0.005828806 0.5677337 0.4380994 -0.1722464
##
##           MAPE      MASE      ACF1
## Training set 3.593288 0.4726798 0.03648157
##
## Forecasts:
##           Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
## May 2018      10.372251  9.717075 11.02743  9.370245 11.37426
## Jun 2018      10.849427 10.100634 11.59822  9.704247 11.99461
## Jul 2018      10.632909  9.841346 11.42447  9.422317 11.84350
## Aug 2018      10.240501  9.426004 11.05500  8.994836 11.48617
## Sep 2018      10.119467  9.265845 10.97309  8.813965 11.42497
## Oct 2018       9.532044  8.684240 10.37985  8.235439 10.82865
## Nov 2018       9.356057  8.482806 10.22931  8.020534 10.69158
## Dec 2018       9.979593  9.005987 10.95320  8.490592 11.46859
## Jan 2019      12.944186 11.628582 14.25979 10.932143 14.95623
## Feb 2019      11.075207  9.905843 12.24457  9.286819 12.86359
## Mar 2019      10.460972  9.316377 11.60557  8.710465 12.21148
## Apr 2019      10.221315  9.064833 11.37780  8.452628 11.99000
##
plot(HOLT_WINTERS.MUL.TD)
```

Forecasts from Holt–Winters' additive method



Forecasts from Holt–Winters' multiplicative method



Ahora la pregunta práctica es ¿Cuál de estos modelos es mejor para pronosticar? Como se discutió anteriormente, para seleccionar el mejor modelo podemos comparar el *RMSE* de las diferentes aproximaciones en la muestra de evaluación. Esto lo podemos hacer fácilmente empleando la función *accuracy* del paquete *forecast* como lo hemos hecho en las anteriores secciones.

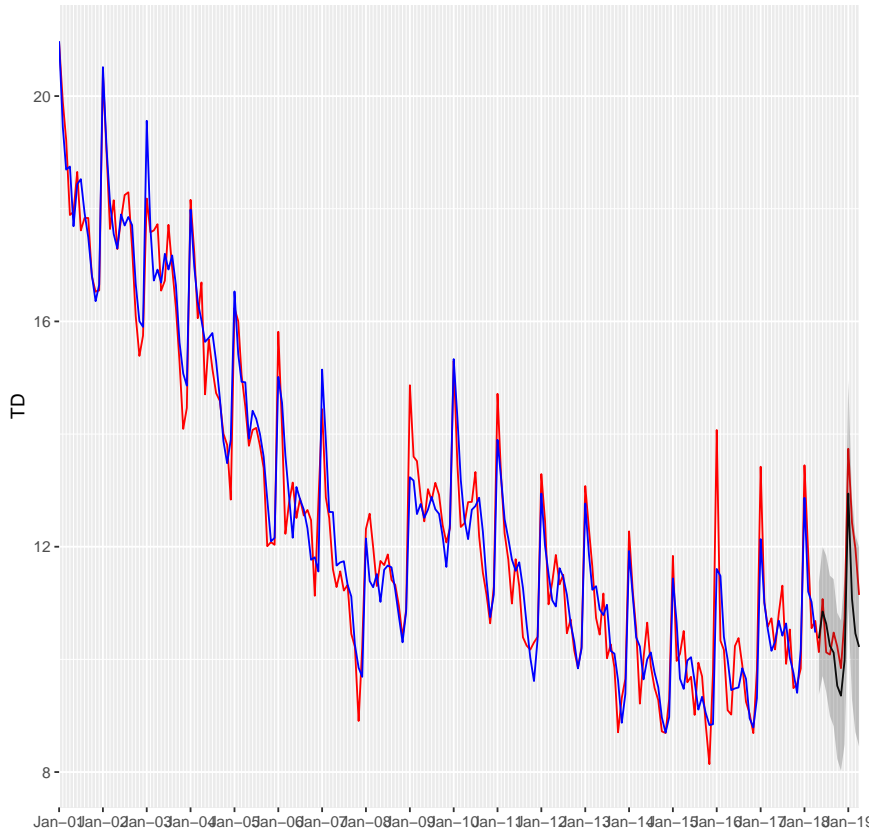
```
RMSE.HOLT_WINTERS.AD.TD <- accuracy(HOLT_WINTERS.AD.TD,
                                     TD.eval)[ "Test set", "RMSE"]
RMSE.HOLT_WINTERS.AD.TD
## [1] 0.9613288

RMSE.HOLT_WINTERS.MUL.TD <- accuracy(HOLT_WINTERS.MUL.TD,
                                     TD.eval)[ "Test set", "RMSE"]
RMSE.HOLT_WINTERS.MUL.TD
## [1] 0.7856407

RMSE.4
## [1] 0.8511266
```

Según estos resultados el mejor método es el Holt-Winters multiplicativo. El pronóstico de ese modelo lo encontramos en el siguiente gráfico.

Figura 3.5 Pronóstico para el último año de la TD con H-W multiplicativo (13 ciudades)
(negro=pronóstico, azul=estimado, rojo=datos, sombra=95 % conf.)

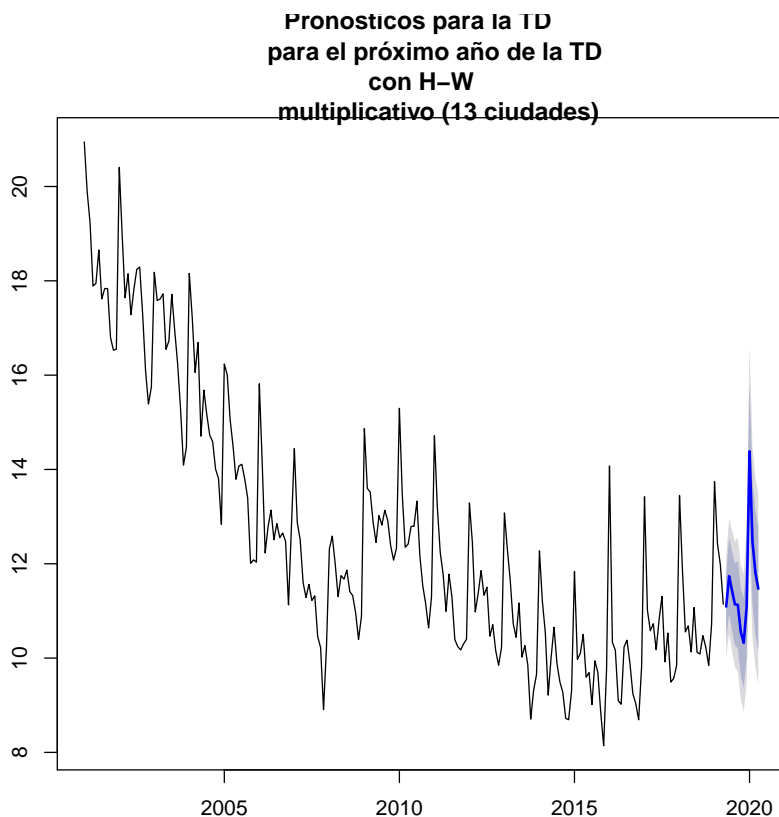


Y por tanto los pronósticos para el siguiente año serán:

```
Pronstoticos <- hw(TD.ts, h = 12, seasonal = "multiplicative")
Pronstoticos
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## May 2019	11.09280	10.395509	11.79009	10.026387	12.15921
## Jun 2019	11.73800	10.938008	12.53798	10.514520	12.96147
## Jul 2019	11.42821	10.590739	12.26568	10.147410	12.70900
## Aug 2019	11.13883	10.266909	12.01075	9.805341	12.47232
## Sep 2019	11.12963	10.203963	12.05531	9.713942	12.54533
## Oct 2019	10.55129	9.622951	11.47963	9.131517	11.97106
## Nov 2019	10.32230	9.365120	11.27948	8.858420	11.78618
## Dec 2019	11.07962	10.000226	12.15901	9.428830	12.73041
## Jan 2020	14.38392	12.915736	15.85210	12.138529	16.62930
## Feb 2020	12.46695	11.136908	13.79700	10.432824	14.50108
## Mar 2020	11.81365	10.499111	13.12819	9.803235	13.82407
## Apr 2020	11.47079	10.142032	12.79955	9.438628	13.50296

```
plot(Pronsoticos, main = "Pronósticos para la TD  
para el próximo año de la TD \n con H-W  
multiplicativo (13 ciudades)")
```



3.6. Comentarios finales

Los métodos que hemos estudiado en esta ocasión son relativamente sencillos, pero esto no implica que no puedan ser poderosos al momento de generar pronósticos. En la práctica del científico de datos estos modelos deben estar siempre en su caja de herramientas. En la práctica siempre comparamos una batería grande de modelos antes de llegar a escoger el mejor modelo para pronosticar. Noten que en este caso, si bien podemos encontrar un buen pronóstico, el modelo no permitirá dar una explicación del por qué el valor pronosticado es obtenido.

3.7. Ejercicios

Empleando la información del número de ocupados en miles de personas (Ocupados), para las 13 principales ciudades de Colombia, los desocupados (Desocupados) y los inactivos (Inactivos) que se encuentra en el archivo `datosEmpleo.xlsx`, genere un pronóstico para los próximos seis meses de cada una de las series.

Capítulo 4

Estacionariedad

En el Capítulo 2 se discutió cómo las series de tiempo que se emplean económicas y financieras y en general las que se emplean en el mundo de los negocios pueden poseer diferentes componentes¹ que implican comportamientos característicos como una tendencia a crecer en el caso de presentarse una tendencia (Ver figura 2.1).

En este capítulo discutiremos una de las propiedades estadísticas más deseables en las series de tiempo: la estacionariedad.

4.1. Estacionariedad (débil)

En general, se dirá que un proceso $\{y_t\}$ es *débilmente estacionario* (weakly stationary en inglés) si:

$$E[y_t] = \mu < \infty \quad (4.1)$$

$$Var[y_t] = \sigma^2 < \infty \quad (4.2)$$

$$Cov[y_t, y_{t-s}] = \gamma_s < \infty \quad (4.3)$$

donde $s \neq t$. Es decir, una serie y_t será estacionario si i) tiene una media que no cambia en el tiempo, ii) la varianza es constante y iii) la covarianza entre dos observaciones separadas por s períodos no depende del tiempo².

Además, un proceso y_t será *estrictamente estacionario* (strictly stationary en inglés) si

$$F\{y_1, y_2, \dots, y_T\} = F\{y_{1+s}, y_{2+s}, \dots, y_{T+s}\}$$

¹ Como por ejemplo la tendencia, el factor cíclico y la estacionalidad.

² Pero si puede depender de s .

donde $F\{\}$ es una función de distribución conjunta. En otras palabras, una serie de tiempo será estrictamente estacionaria si la distribución es la misma independiente del periodo en que se encuentre el proceso. Esto es mucho más restrictivo que el concepto de procesos débilmente estacionario. En este libro nos concentraremos únicamente en procesos débilmente estacionarios. Por eso de aquí en adelante cuando se hable de procesos estacionarios se deberá entender que nos referimos a procesos débilmente estacionarios.

Regresando a los procesos (débilmente) estacionarios, la condición (4.1) implica que el proceso (débilmente) estacionario fluctuará siempre al rededor de μ . Es decir, una serie (débilmente) estacionario no crecerá en el tiempo y por tanto no puede presentar una tendencia. Así, las series del PIB anual y la base monetaria mensual presentadas en la figura 2.1 aparentemente³ no cumplen esta condición y por tanto no parecen ser estacionarias.

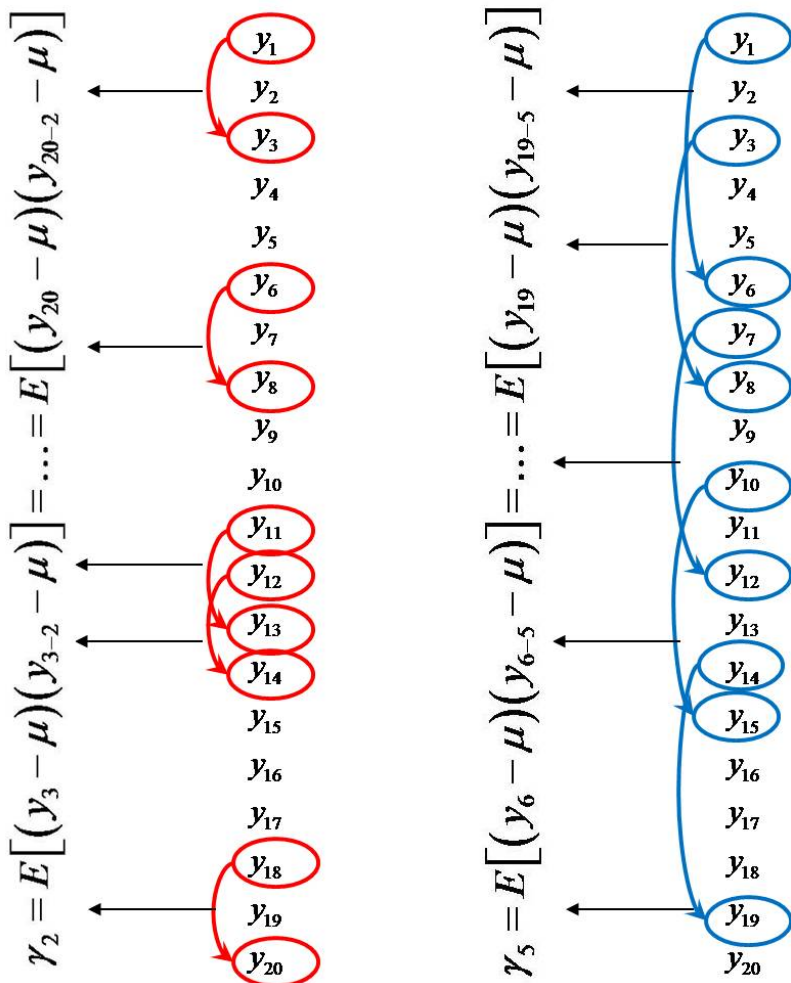
La condición (4.2) implica que el proceso sea homoscedástico para toda observación. Por ejemplo, el rendimiento de la TRM, si bien parece fluctuar para todas las observaciones al rededor de la misma media, la desviación con respecto a la media no es la misma. Es más, parece ser que la desviación con respecto a la media (volatilidad) tiene un patrón de comportamiento tal que menores desviaciones tienden a ser seguidas de menores desviaciones; al igual que periodos de gran volatilidad tienden a ser seguidos de periodos volátiles. A este fenómeno se le conoce como volatilidad agrupada (volatility clustering); fenómeno que es muy común cuando se estudian rendimientos de activos financieros (ver Alonso C. y Arcos, 2006 o Alonso y Torres, 2014 para una mayor discusión). Regresando a la serie de rendimientos diarios del IGBC, si bien aparentemente cumple la condición (4.1), no parece cumplir la condición (4.2); por lo tanto, aparentemente no es un proceso estacionario.

Finalmente, la condición (4.3) tal vez es casi que imposible de ser observada “con el ojo”. Esta condición implica que la relación lineal⁴ entre observaciones se mantengan constantes. En otras palabras, la covarianza (también conocida como la autocovarianza por tratarse de la relación lineal de observaciones de la misma serie de tiempo) no puede depender del tiempo (t), sino de la distancia entre las variables comparadas.

Para entender intuitivamente esta condición, consideremos una serie de tiempo $\{y_t\}$ para $t = 1, 2, \dots, 20$. Ahora detengamos nuestra atención en la covarianza (autocovarianza) entre las observaciones que están separadas por 5 periodos la llamaremos γ_5 , es decir la medida de relación lineal entre variables aleatorias que están rezagadas 5 periodos. Por ejemplo, entre la relación lineal entre la observación 6 y la 1, entre la 19 y la 14, etc. (ver la figura 4.1). Consideremos otra autocovarianza, la que mide la relación (lineal) entre observaciones que están distanciadas por dos periodos; es decir γ_2 (ver la figura 4.1). La condición (4.3) implica que estas autocovarianzas dependerán únicamente del número de rezagos o espacios que separan las variables (s) y no del tiempo (t).

³ Más adelante estudiaremos pruebas formales que permitan determinar si una serie de tiempo observada corresponde a un proceso estacionario o no.

⁴ Recordemos que la covarianza es una medida de la relación lineal entre dos variables.

Figura 4.1 Esquema de la autocovarianza para 2 y 5 rezagos

Antes de continuar es importante reconocer que (4.1) implica que el primer momento de la distribución de la serie de tiempo sea finita y constante en el tiempo. Por otro lado, las condiciones (4.2) y (4.3) implican que el segundo momento (al redor de la media) de la distribución que sigue el proceso sea constantes para todo t y finita. Es decir:

$$E[(y_t - \mu)(y_{t-s} - \mu)] = \gamma_s \quad (4.4)$$

Por eso un proceso (débilmente) estacionario también es conocido como *proceso covarianza estacionario* (covariance stationary en inglés) o *proceso estacionario de segundo orden* (second-order stationary en inglés).

4.2. El ruido blanco

La piedra angular de todos los procesos que estudiaremos a lo largo del libro es el proceso ruido blanco (white noise en inglés)⁵. En general un proceso ruido blanco ε_t deberá cumplir que:

$$E[\varepsilon_t] = 0 \quad (4.5)$$

$$Var[\varepsilon_t] = \sigma_\varepsilon^2 \quad (4.6)$$

$$Cov[\varepsilon_t, \varepsilon_s] = E[\varepsilon_t \varepsilon_s] = 0 \quad (4.7)$$

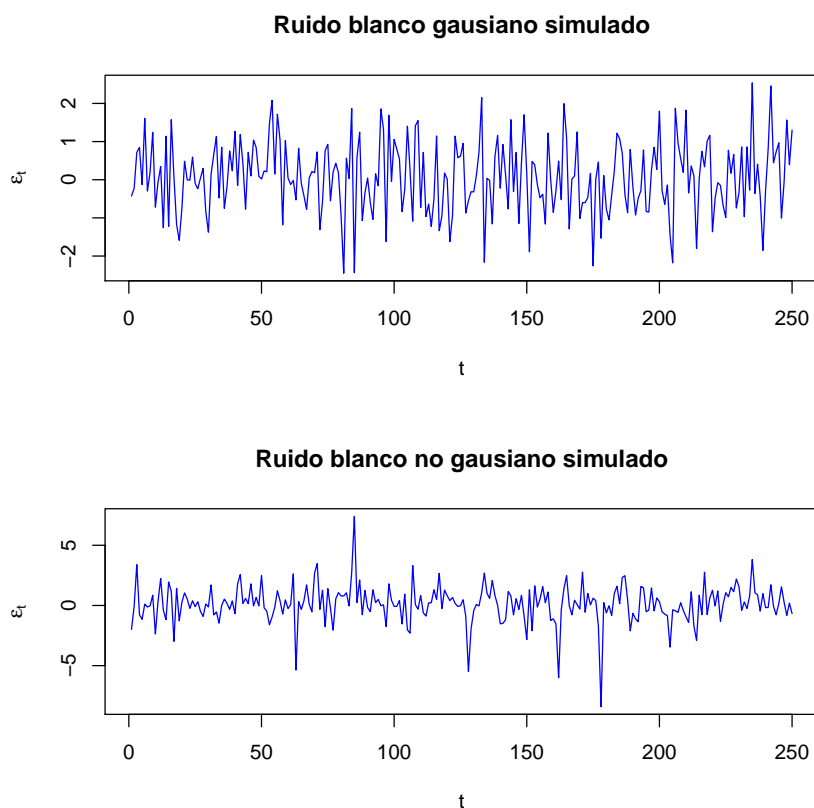
Es decir, un proceso ruido blanco corresponde a una serie de tiempo si la media es cero, la varianza es constante y no hay ninguna relación lineal entre el valor actual y cualquier valor pasado. Es decir, (4.5) y (4.7) implica que la serie de tiempo no se puede predecir (por la ausencia de cualquier tipo de autocorrelación) por medio de alguna combinación lineal de sus valores pasados y (4.6) implica la presencia de homoscedasticidad.

Si además de (4.5), (4.6) y (4.7) se asume que ε_t sigue una distribución normal, entonces tendremos un ruido blanco gaussiano o normal; es decir,

$$\varepsilon_t \sim i.i.d.N(0, \sigma_\varepsilon^2) \quad (4.8)$$

En la figura 4.2 se presentan dos series de tiempo generadas a partir de un ruido blanco gaussiano (con varianza de uno) y un ruido blanco generado a partir de una distribución t de student con 3 grados de libertad. Como se puede ver en dicha figura, sin importar la distribución, las muestras no presentan ningún tipo de patrón de comportamiento. La única diferencia entre los dos procesos es que aquél generado por la distribución t tiene un rango mayor; esto se produce gracias a que la distribución t presenta una mayor probabilidad de observar valores extremos que lo que ocurre con una distribución estándar normal).

⁵ Este proceso toma su nombre de la luz blanca, pues la física ha demostrado que el ruido blanco es un proceso que puede representar la señal que emite este tipo de luz.

Figura 4.2 250 Simulaciones de un proceso ruido blanco

Finalmente, es importante tener en cuenta que un proceso ruido blanco será por definición un proceso estacionario. De aquí en adelante siempre que empleemos ε_t nos referiremos a un proceso de ruido blanco.

4.3. ¿Cómo determinar si una serie es ruido blanco?

En la práctica nos encontraremos ante situaciones en las cuáles será buena idea comprobar si una determinada serie es estacionaria o no. Mas adelante discutiremos esto en detalle. Pero por ahora nos concentraremos en cómo constatar si una serie se comporta como un ruido blanco o no.

Recordemos que las características de un proceso ruido blanco son: **media igual a cero, homoscedasticidad y no autocorrelación entre las observaciones.** Noten que

esto es similar a los supuestos del Teorema de Gauss-Markov sobre el residuo de un modelo de regresión múltiple.

Así podemos emplear algunos elementos que ya conocemos. Iniciemos por la parte mas difícil de detectar a “ ojo ”, la autocorrelación.

Para constatar si una serie **no tienen autocorrelación**, podemos de manera formal emplear las pruebas de:

- Prueba de **Rachas**
- Prueba de **Box-Pierce y Ljung-Box**

Todas estas pruebas tienen en común es que la **hipótesis de alterna** de estas es la existencia de **una serie** (errores en este caso) **no autocorrelacionado** de alguna manera. En no de los apéndices de este capítulo se presenta una discusión de cada uno de estas pruebas de autocorrelación.

Para comprobar la posible existencia de **heteroscedasticidad**, nos concentraremos en el tipo de heteroscedasticidad mas común en las series de tiempo: un comportamiento ARCH (heteroscedástico condicional en el tiempo) o GARCH.

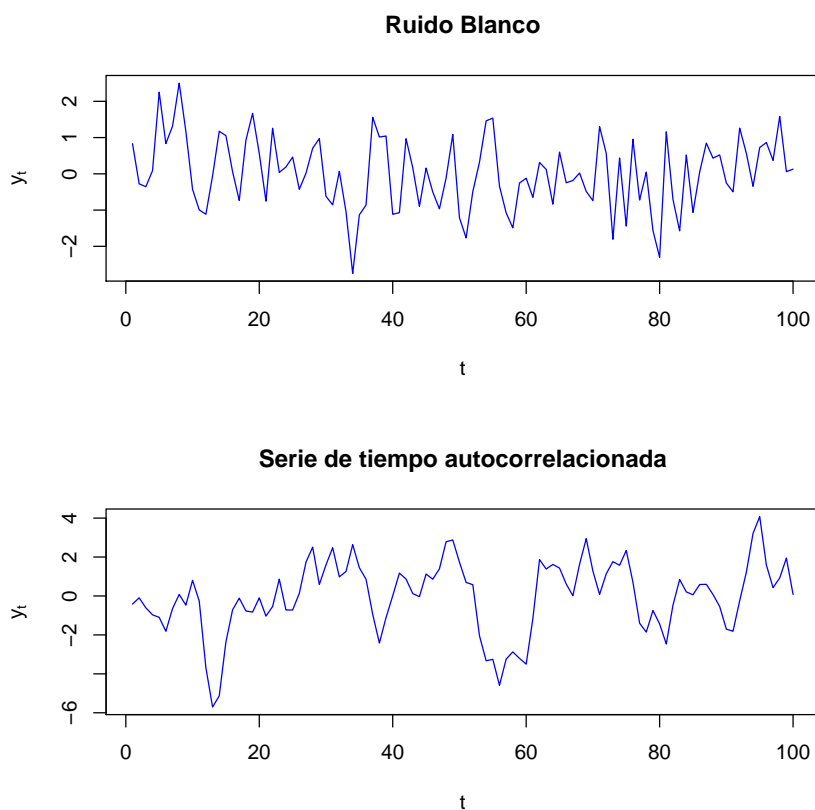
Finalmente, podemos evaluar si los residuos siguen una distribución normal o no. En este caso podemos emplear de manera informal el gráfico de quantiles acumulado de probabilidad (q-q normality plot en inglés). Y formalmente, se pueden emplear diferentes **pruebas de normalidad** como lo discuten Alonso y Montenegro, 2015 y Alonso y Estrada (2018)). En este momento nos concentraremos en las pruebas de Jarque-Bera y Shapiro. Estas dos pruebas tienen en común que su **hipótesis nula** es que los datos provienen de una población **que sigue una distribución normal**, versus la **hipótesis alterna** que los datos **no siguen una distribución normal**. En otras palabras, realmente estas pruebas son de no-normalidad.

4.3.1. Ejemplo con una serie simulada

Para ilustrar esto empleemos dos series simuladas: un ruido blanco (y1) y un proceso que no es ruido blanco y2. Para esto empleemos la función *arima.sim* del paquete base de R.

```
set.seed (123456)
y1 <- arima.sim(model =list(order = c(0, 0, 0)), n= 100)
y2 <- arima.sim(model =list(order = c(1, 0, 1), ar = c(0.5),
                                ma = c(0.8)), n= 100)
```

En la Figura 4.3 se presenta las series simuladas.

Figura 4.3 Procesos simulados.

4.3.2. Autocorrelación

Para determinar si la serie es o no ruido blanco podemos realizar las pruebas que ya conocemos de nuestro curso anterior. Para realizar la **prueba de rachas**, podemos emplear la función `runs.test()` del paquete **tseries**. Esta función tiene dos argumentos; estos argumentos son:

```
runs.test(x, alternative = c("two.sided", "less", "greater"))
```

donde:

- “x”: este argumento es obligatorio y corresponde a un objeto que contenga una variable de clase `factor` que muestre si el error es positivo o negativo

- “alternative”: Este argumento no es obligatorio. Este argumento le provee a la función cuál es la hipótesis alterna que se desea probar. Si *alternative* = “two.sided”, la alterna será que existe algún tipo de autocorrelación. Esta es la opción por defecto, es decir si no se especifica este argumento, se efectuará esta prueba. Si *alternative* = “less” la alterna es que la autocorrelación es positiva (menos cambios de signos que los esperados) y si *alternative* = “greater”, la alterna es que existe autocorrelación negativa (mas cambios de signos que los esperados).

Así, nuestro primer paso para efectuar esta prueba es convertir la serie en una variable dicotómica que muestre el signo del residual para cada periodo.

```
library(tseries)
signo.y1 <- factor(y1>0)
head(signo.y1, 3)

## [1] TRUE  FALSE FALSE
## Levels: FALSE TRUE

runs.test(signo.y1)

##
##  Runs Test
##
## data:  signo.y1
## Standard Normal = -0.74456, p-value = 0.4565
## alternative hypothesis: two.sided
```

Los resultados no permiten rechazar (valor p de 0.4565) la no autocorrelación y por tanto se puede concluir que no existe autocorrelación en la serie. Es decir, con esta prueba y en este aspecto, la serie y1 parece comportarse como un ruido blanco. Ahora hacemos lo mismo con la serie y2.

```
signo.y2 <- factor(y2>0)
head(signo.y2, 3)

## [1] FALSE FALSE FALSE
## Levels: FALSE TRUE

runs.test(signo.y2)

##
##  Runs Test
##
## data:  signo.y2
## Standard Normal = -6.5998, p-value = 4.119e-11
## alternative hypothesis: two.sided
```

Los resultados permiten rechazar (valor p de 0) la no autocorrelación y por tanto se puede concluir que existe autocorrelación en la serie. Es decir, con esta prueba la serie y2 no parece comportarse como un ruido blanco.

Este análisis debe complementarse con otras pruebas. La prueba de Box-Pierce y la modificación de Ljung-Box pueden calcularse empleando la misma función (*Box.test()*) de la base de R. Esta función tiene tres argumentos importantes,

Box.test(x, lag = 1, type = c("Box-Pierce", "Ljung-Box"))

donde:

- “x”: el vector al que se le quiere realizar la prueba
- “lag”: el número de rezagos (*lag*) para incluir en la hipótesis nula y alterna. Por defecto *lag* = 1
- “type”: el tipo de prueba (*type*). Las opciones son *type* = “Box-Pierce” (la opción por defecto de la función) y *type* = “Ljung-Box”. En el segundo caso se realiza la prueba de Ljung-Box que introduce la corrección para muestras pequeñas.

Entonces para probar la hipótesis $H_0 : \gamma_1 = 0$ con la prueba de Box-Pierce podemos emplear el siguiente código

```
Box.test(y1, lag = 1)

##
## Box-Pierce test
##
## data: y1
## X-squared = 5.252, df = 1, p-value = 0.02192
```

Así, no podemos rechazar la hipótesis nula de no autocorrelación (para el primer rezago). Como se mencionó anteriormente, es común probar esta hipótesis para los primeros rezagos, por lo menos los primeros 20. Es decir,

$$\begin{aligned} H_0 : \gamma_1 &= 0 \\ H_0 : \gamma_1 &= \gamma_2 = 0 \\ &\vdots \\ H_0 : \gamma_1 &= \gamma_2 = \dots \gamma_{20} = 0 \end{aligned}$$

Para realizar estas prueba podemos crear una función para construir una tabla con todos los rezagos que se deseen emplear .

```
tabla.Box.Pierce <- function(residuo, max.lag = 20,
                             type = "Box-Pierce"){
  # se crean objetos para guardar los resultados
  BP.estadistico <- matrix(0,max.lag,1)
  BP.pval <-matrix(0,max.lag,1)

  # se calcula la prueba para los diferentes rezagos
  for (i in 1:max.lag) {
    BP<- Box.test(residuo, lag = i, type = type)
    BP.estadistico[i]<-BP$statistic
    BP.pval[i]<-round(BP$p.value,5)
  }
  labels<- c( "Rezagos", type, "p-valor")
```

```

Cuerpo.Tabla <- cbind(matrix(1:max.lag,max.lag,1),
                        BP.estadistico, BP.pval)
TABLABP <- data.frame(Cuerpo.Tabla)
names(TABLABP) <- labels
return(TABLABP)
}

```

Ahora podemos emplear la función para crear el siguiente cuadro.

Cuadro 4.1 Prueba de Box-Pierce para la serie y_1 para los primeros rezagos

Rezagos	Box-Pierce	p-valor
1.00	5.25	0.02
2.00	5.46	0.07
3.00	5.46	0.14
4.00	5.87	0.21
5.00	6.17	0.29
6.00	7.61	0.27
7.00	7.77	0.35
8.00	8.37	0.40
9.00	8.38	0.50
10.00	12.50	0.25
11.00	13.19	0.28
12.00	13.71	0.32
13.00	13.71	0.39
14.00	13.73	0.47
15.00	14.33	0.50
16.00	14.66	0.55
17.00	18.13	0.38
18.00	18.16	0.44
19.00	18.65	0.48
20.00	19.48	0.49

Los resultados nos permiten concluir que las autocorrelaciones de la serie y_1 son cero para todos los rezagos analizados. Así, podemos concluir que no existe autocorrelación en los residuos de acuerdo con esta prueba.

El lector puede generar fácilmente la siguiente tabla que contienen los resultados de la prueba de Ljung-Box aplicada a la misma serie.

No es sorprendente que los resultados de esta prueba sean los mismo que los obtenidos con la prueba de Box-pierce, pues en este caso la muestra es grande. Así, la corrección para muestras pequeñas no era importante en este caso.

Las dos pruebas coinciden que la serie no está autocorrelacionada.

Para el caso de la serie y_2 el lector puede encontrar los resultados reportados en la Tabla 4.3.

Esto ratifica que la serie y_2 presenta autocorrelación y por tanto no es un ruido blanco.

Cuadro 4.2 Prueba de Ljung-Box para la serie y1 para los primeros rezagos

Rezagos	Ljung-Box	p-valor
1.00	5.41	0.02
2.00	5.63	0.06
3.00	5.63	0.13
4.00	6.07	0.19
5.00	6.38	0.27
6.00	7.95	0.24
7.00	8.12	0.32
8.00	8.79	0.36
9.00	8.80	0.46
10.00	13.47	0.20
11.00	14.26	0.22
12.00	14.86	0.25
13.00	14.86	0.32
14.00	14.88	0.39
15.00	15.60	0.41
16.00	16.01	0.45
17.00	20.27	0.26
18.00	20.31	0.32
19.00	20.92	0.34
20.00	21.98	0.34

Cuadro 4.3 Prueba de Ljung-Box para la serie y2 para los primeros rezagos

Rezagos	Ljung-Box	p-valor
1.00	62.30	0.00
2.00	84.46	0.00
3.00	92.38	0.00
4.00	94.15	0.00
5.00	94.23	0.00
6.00	94.27	0.00
7.00	94.64	0.00
8.00	97.03	0.00
9.00	101.63	0.00
10.00	105.21	0.00
11.00	107.72	0.00
12.00	110.28	0.00
13.00	113.02	0.00
14.00	116.84	0.00
15.00	121.52	0.00
16.00	124.63	0.00
17.00	125.83	0.00
18.00	126.25	0.00
19.00	126.32	0.00
20.00	126.50	0.00

4.3.3. *Heteroscedasticidad*

Ahora nos queda por chequear si existe problemas de heteroscedasticidad, en especial un comportamiento tipo ARCH o GARCH (en un próximo capítulo dis-

cutiremos en detalle este tema). Esto implica que la varianza de la serie tenga un comportamiento no autocorrelacionado.

Una aproximación para determinar si existe un comportamiento GARCH o ARCH es emplear la prueba de Ljung-Box sobre la serie (sin media) al cuadrado. En este caso,

En este caso esto implica la siguiente linea de código.

```
tabla.Box.Pierce((y1-mean(y1))^2, type = "Ljung-Box")
```

Los resultados del la Tabla 4.4, permiten rechazar que la nula de un comportamiento homoscedastico (no auto-correlación de las varianzas) y por tanto no hay comportamiento ARCH o GARCH.

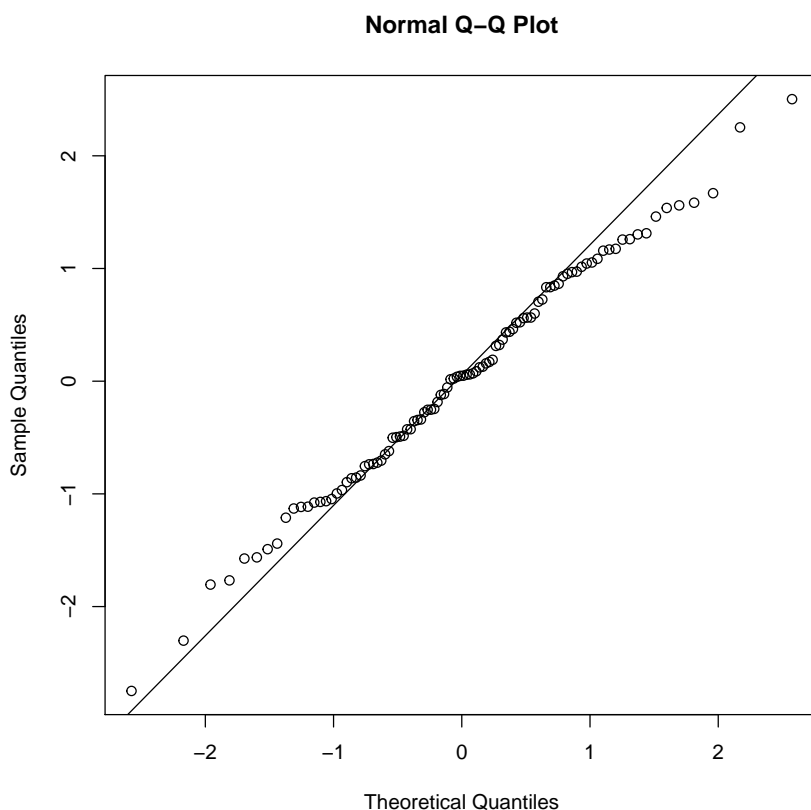
Cuadro 4.4 Prueba de Ljung-Box para la serie y1 al cuadrado para los primeros rezagos

Rezagos	Ljung-Box	p-valor
1.00	0.57	0.45
2.00	0.92	0.63
3.00	6.29	0.10
4.00	6.41	0.17
5.00	7.43	0.19
6.00	7.73	0.26
7.00	7.89	0.34
8.00	9.77	0.28
9.00	11.27	0.26
10.00	11.78	0.30
11.00	13.72	0.25
12.00	14.30	0.28
13.00	15.31	0.29
14.00	15.65	0.33
15.00	15.66	0.40
16.00	15.91	0.46
17.00	15.94	0.53
18.00	16.34	0.57
19.00	17.24	0.57
20.00	17.28	0.63

4.3.4. Normalidad

Finalmente, podemos evaluar si la serie sigue una distribución normal o no. Noten que esto no es necesario para ser ruido blanco, solo para ser ruido blanco gaussiano. Primero hagamos un análisis gráfico empleando las siguientes lineas de código.

```
qqnorm(y1)
qqline(y1)
```


Figura 4.4 Gráficos q-q para la serie y1

El gráfico q-q muestra en el eje horizontal el cuantil teórico de la distribución normal y en el eje vertical el cuantil en la muestra. Además se incluye una línea recta que muestra cómo debería ser el cuantil de la muestra para que sea igual al teórico. Es decir, lo que debería ocurrir si la muestra fuese producida por una distribución normal. La Figura 4.4 muestra con puntos las observaciones de la serie. En este caso vemos que hay unas observaciones extremas que distan relativamente poco de la distribución normal. Esto podría generar algún tipo de alarma sobre el no cumplimiento del supuesto de normalidad. Pero la decisión final la deberíamos tomar con pruebas estadísticas.

Para realizar las pruebas de normalidad de Jarque-Bera y Shapiro, emplearemos las funciones `jarque.bera.test()` del paquete `tseries` y `shapiro.test` disponible en el paquete central de R. Estas dos pruebas solo requieren como argumento la serie a la cual se le quiere realizar la prueba.

En las siguientes línea de código se puede observar como emplear estas dos pruebas.

```
library(tseries)
jarque.bera.test(y1)

##
##  Jarque Bera Test
##
## data:  y1
## X-squared = 0.31453, df = 2, p-value = 0.8545

shapiro.test(y2)

##
##  Shapiro-Wilk normality test
##
## data:  y2
## W = 0.97056, p-value = 0.02439
```

Para ambas pruebas, no se puede rechazar la hipótesis de que los errores provienen de una distribución normal. Así, parece ser que la serie `y1` proviene de un proceso ruido blanco gaussiano.

4.4. Comentarios Finales

En este capítulo hemos estudiado el concepto de estacionariedad y un proceso muy importante para el modelaje estadístico de las series de tiempo: el ruido blanco. En general para determinar si una serie es estacionaria o no existen pruebas que veremos más adelante.

Pero por ahora, una buena práctica es graficar las serie de tiempo. Si esta tiene una tendencia clara, probablemente no será estacionaria. Una vez se descarte la existe de una tendencia, debemos proceder a chequear las condiciones para que una serie sea estacionaria. En este capítulo solo estudiamos las condiciones para que un proceso sea ruido blanco, pero más adelante veremos las condiciones para que una serie sea estacionaria.

4.5. Apéndice: Descripción de pruebas de autocorrelación empleadas en este capítulo

4.5.1. Prueba de Rachas (*Runs test*)

La prueba de rachas, propuesta por Wald y Wolfowitz, 1940, es una prueba de independencia lineal no paramétrica cuya idea es relativamente sencilla. Si no hay

autocorrelación, entonces no deberían haber muchos errores seguidos con el mismo signo (autocorrelación positiva), ni tampoco muchos cambios de signo seguido (autocorrelación negativa). En otras palabras, debe existir la cantidad adecuada de cambios de signo en una serie de datos: ni muchos, ni pocos.

Esta prueba tiene además la ventaja de no necesitar suponer una distribución de los errores. Para probar la hipótesis nula de que los errores son totalmente aleatorios ($H_0 : \rho = 0$) versus la alterna de que existe algún tipo de autocorrelación en los errores, se requiere seguir los siguientes pasos a partir de los errores estimados:

1. Cuente el número de errores con signo positivo (N_+) y con signo (N_-)
2. Cuente el número rachas (k), es decir de "seguidillas" de signo. Por ejemplo, si tenemos que los signos de los errores son: - - - - + + - + + - + +. Entonces se tendrán seis rachas ($k = 6$). (- - - -)(+ + +)(-)(+ + +)(-)(+ +). Note que el número de rachas es igual al número de cambios de signo
3. Si N_+ y/o N_- son menores que 20, entonces se puede construir un intervalo de confianza del 95 % para el número de rachas "razonable" bajo la hipótesis nula.
4. Si N_+ y/o N_- son mayores que 20, entonces se puede construir un intervalo de confianza del $(1 - \alpha)100\%$ para el número de rachas "razonable" bajo la hipótesis de la siguiente manera:

$$[E[k] \pm z_{\frac{\alpha}{2}} \sqrt{\text{Var}[k]}] \quad (4.9)$$

donde, el valor esperado y la varianza de k (las rachas) son:

$$E(k) = \frac{2N_+N_-}{N_+ + N_-} + 1 \quad (4.10)$$

$$\text{Var}(k) = \frac{2N_+N_- (2N_+N_- - N_+ - N_-)}{(N_+ + N_-)^2 (N_+ + N_- - 1)} \quad (4.11)$$

La hipótesis nula se puede rechazar si el número de rachas observadas no están contenidas en el intervalo de confianza.⁶

4.5.2. Prueba de Box-Pierce y Ljung-Box

Otra aproximación para comprobar la existencia o no de autocorrelación es determinar si las autocorrelaciones a diferentes rezagos son o no iguales a cero. Box y Pierce, 1970 diseñan una prueba basada en la autocorrelación muestral de los errores que permite detectar la existencia de errores con procesos más persistentes que AR(1). Recordemos que la autocorrelación poblacional se define de la siguiente forma:

⁶ Una manera alternativa para comprobar la hipótesis nula, cuando se tienen más de 20 observaciones de un mismo signo es emplear como estadístico de prueba $RA = \frac{k - E(k)}{\sqrt{\text{Var}(k)}}$. La hipótesis nula puede ser rechazada si $|RA| > z_{\frac{\alpha}{2}}$.

$$\gamma_j = \frac{\text{Cov}(\varepsilon_t, \varepsilon_{t-j})}{\sqrt{\text{Var}(\varepsilon_t) \text{Var}(\varepsilon_{t-j})}} = \frac{\text{Cov}(\varepsilon_t, \varepsilon_{t-j})}{\sigma_\varepsilon^2}$$

Y la correspondiente autocorrelación muestral es:

$$\hat{\gamma}_j = \frac{\sum_{t=k+1}^n (\hat{\varepsilon}_t - \bar{\varepsilon})(\hat{\varepsilon}_{t-j} - \bar{\varepsilon})}{\sum_{t=1}^n (\hat{\varepsilon}_t - \bar{\varepsilon})^2} = \frac{\sum_{t=j+1}^n \hat{\varepsilon}_t \hat{\varepsilon}_{t-j}}{\sum_{t=1}^n (\hat{\varepsilon}_t)^2}$$

Box y Pierce, 1970 definen una prueba que permite determinar si las primeras s autocorrelaciones son conjuntamente iguales a cero o no. Es decir, permite comprobar la **hipótesis nula** de un error **no autocorrelacionado** (las correlaciones a los s rezagos son cero), versus la **hipótesis alterna** de la existencia de **algún** tipo de **autocorrelación** (por lo menos una autocorrelación no es cero). Para comprobar esta hipótesis nula, Box y Pierce, 1970 sugieren el estadístico Q :

$$Q = n \sum_{j=1}^s r_k^2 \sim_a \chi_s^2$$

donde s corresponden al número de rezagos que se desean considerar dentro de la prueba. Ellos demuestran que su estadístico sigue una distribución Chi-cuadrado con s grados de libertad (χ_s^2). Por lo tanto, ser? posible rechazar la H_0 (error no autocorrelacionado) si se cumple que $Q > \chi_s^2$.

Sin embargo, la prueba de Box-Pierce sólo es válida para muestras grandes ($n > 20$), para resolver este inconveniente Ljung y Box, 1979 proponen una modificación del estadístico anterior para que presente un mejor comportamiento en muestras pequeñas. El estadístico de Ljung-Box corresponde a:

$$Q' = n(n+2) \sum_{k=1}^s \frac{r_j^2}{n+j}$$

Este estadístico funciona y posee la misma distribución que el de la prueba de Box-Pierce.

Finalmente, es importante mencionar que una práctica muy común es realizar esta prueba para un número relativamente grande de rezagos. Es decir, hacer las correspondientes pruebas para diferentes rezagos; por ejemplo, se calculan los correspondientes estadísticos para comprobar las siguientes hipótesis alternas:

$$\begin{aligned} H_0 : \gamma_1 &= 0 \\ H_0 : \gamma_1 &= \gamma_2 = 0 \\ &\vdots \\ H_0 : \gamma_1 &= \gamma_2 = \dots \gamma_{n/3} = 0 \end{aligned}$$

La decisión de si los errores están o no autocorrelacionados se toma teniendo en cuenta las decisiones de cada una de estas pruebas.

Capítulo 5

Encontrando la tendencia y la estacionalidad con modelos de regresión

En el Capítulo 2 discutimos como las series de tiempo pueden tener diferentes componentes: el componente cíclico y de tendencia T_t , estacional S_t y aleatorio R_t . Además, estudiamos la descomposición aditiva de una variable de serie de tiempo (y_t) de la forma:

$$y_t = S_t + T_t + R_t \quad (5.1)$$

Este Capítulo presenta una introducción a la estimación de la tendencia T_t y el componente estacional S_t en el contexto de la regresión múltiple. Nos concentraremos en la estimación de una tendencia lineal, cuadrática, polinómica y exponencial. Y posteriormente como estimar el componente estacional por medio de variables dummy.

Para este capítulo emplearemos la información la serie del número de tiquetes mensuales vendidos por una empresa de transporte terrestre disponible en el archivo `Tiquetes.csv`. Los datos corresponden al periodo junio de 2008 a mayo de 2018 (120 observaciones).

Antes de iniciar, carguemos los datos de los tiquetes terrestres como un objeto `ts` que denominaremos `TT` y construyamos un gráfico.

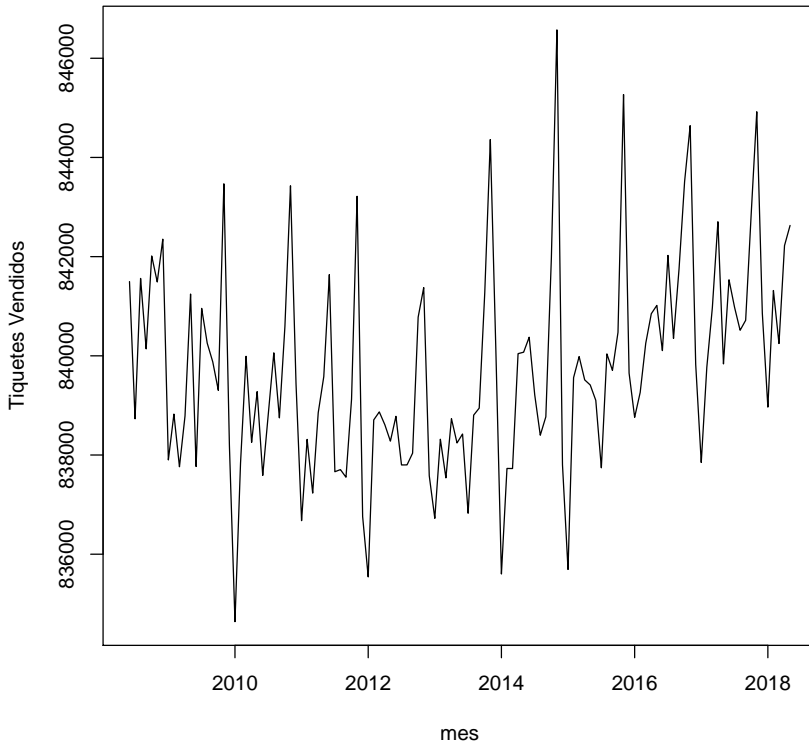
De acuerdo con la figura 5.1 la serie presenta una tendencia que aparentemente no es lineal y un componente estacional.

5.1. Estimación de la tendencia

5.1.1. Tendencia lineal

Un modelo con una tendencia lineal implica modelar el componente T_t de 5.1 suponiendo el siguiente DGP :

$$y_t = \beta_1 + \beta_2 t + \varepsilon_t \quad (5.2)$$

Figura 5.1 Tiquetes terrestres vendidos mensualmente

donde $t = 1, 2, \dots, T$. Noten que esto corresponde a una regresión de la variable en función del tiempo. Esto ya sabemos como hacerlo. Primero debemos crear la variable t y luego corre la respectiva regresión.

```
t <- 1:length(TT.ts)

data1 <- ts(cbind(TT.ts, t), start = c(2008,6), frequency = 12)

res.t.lineal <- lm(TT.ts ~ t, data = data1)

summary(res.t.lineal)

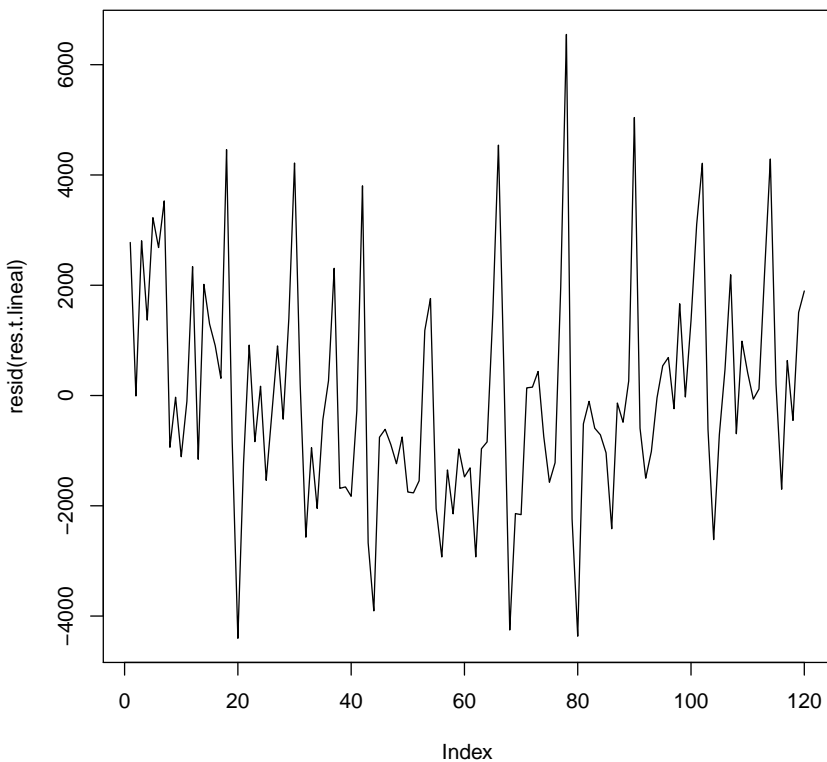
##
## Call:
## lm(formula = TT.ts ~ t, data = data1)
##
## Residuals:
```

##	Min	1Q	Median	3Q	Max
----	-----	----	--------	----	-----

```
## -4403 -1230 -298 1034 6549
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.387e+05  3.732e+02 2247.297 < 2e-16 ***
## t           1.691e+01  5.353e+00   3.159  0.00201 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2031 on 118 degrees of freedom
## Multiple R-squared:  0.07796, Adjusted R-squared:  0.07015
## F-statistic: 9.978 on 1 and 118 DF, p-value: 0.002013
```

Antes de proceder a analizar los resultados de esta regresión, se debe constatar que los residuos cumplan los supuestos del Teorema de Gauss-Markov. Recuerden que los resultados de esta regresión solo serán válidas si se cumplen dichos supuestos.

```
plot(resid(res.t.lineal), type="l")
```



En este gráfico podemos ver que los residuales no son totalmente aleatorios. Se puede observar una tendencia no lineal en ellos y un componente estacional. Por lo tanto aún existe información por extraer de esta serie. Así, no procederemos a constatar si los supuestos se cumplen, claramente estos no se cumplirán. Es mas, si graficamos la tendencia lineal encontrada (\hat{y}_t que lo guardaremos en el objeto *t.lineal*) y la serie original es evidente que el fit del modelo no es bueno.

```
t.lineal <- fitted(res.t.lineal)
t.lineal <- start = c(2008,6), frequency = 12
ts.plot(TT.ts, t.lineal, col=1:2)

## Error: <text>:2:30: unexpected ', '
## 1: t.lineal <- fitted(res.t.lineal)
## 2: t.lineal <- start = c(2008,6),
##
```

No obstante, si quisiéramos tener el pronóstico para los siguientes periodos con esta tendencia lineal, esto lo podríamos hacerlo de manera mas sencilla si empleamos el paquete **forecast** (Hyndman y Khandakar, 2008) y la función *tslm* que estima modelos lineales como la función *lm*, pero está diseñada para trabajar mas fácilmente con series de tiempo. Los argumentos de esta función son similares a los de la función *lm*. Una diferencia sustancial es que con esta función no se debe crear la variable de la tendencia. Esto lo hace automáticamente la función¹. Veamos como funciona.

```
library(forecast)
res.t.lineal.2 <- tslm(TT.ts ~ trend, data = TT.ts)
summary(res.t.lineal.2)

##
## Call:
## tslm(formula = TT.ts ~ trend, data = TT.ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4403  -1230   -298    1034    6549
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.387e+05  3.732e+02 2247.297 < 2e-16 ***
## trend      1.691e+01  5.353e+00   3.159  0.00201 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2031 on 118 degrees of freedom
## Multiple R-squared:  0.07796, Adjusted R-squared:  0.07015
## F-statistic: 9.978 on 1 and 118 DF, p-value: 0.002013
```

Noten que los resultados son exactamente iguales a los obtenidos anteriormente, es mas el objeto *res.t.lineal* es de la clase *lm*. (Compruébelo) Ahora podemos crear

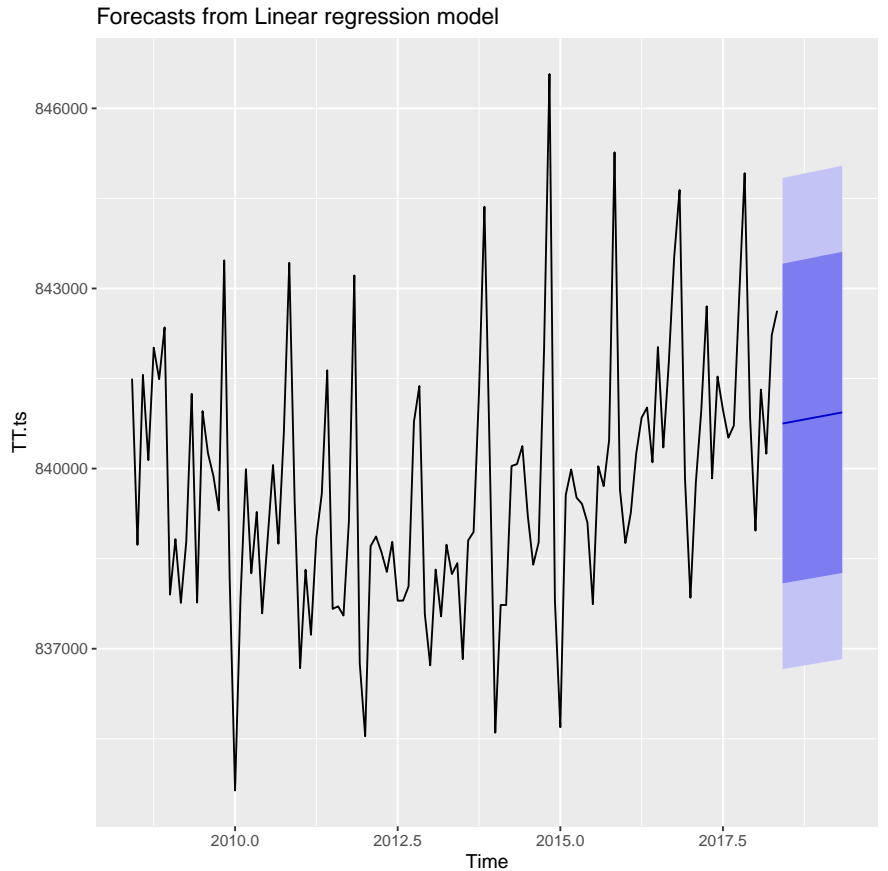
¹ Esta función también puede crear automáticamente las variables dummy para la estacionalidad.

un pronóstico rápidamente empleando la función *forecast* del paquete del mismo nombre.

```
forecast(res.t.lineal.2 , h=12)
```

##	Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## Jun 2018		840750.1	838088.3	843411.9	836660.1	844840.1
## Jul 2018		840767.0	838104.2	843429.9	836675.4	844858.7
## Aug 2018		840783.9	838120.0	843447.9	836690.6	844877.3
## Sep 2018		840800.9	838135.8	843466.0	836705.8	844896.0
## Oct 2018		840817.8	838151.5	843484.0	836720.9	844914.6
## Nov 2018		840834.7	838167.3	843502.1	836736.0	844933.3
## Dec 2018		840851.6	838183.0	843520.2	836751.1	844952.0
## Jan 2019		840868.5	838198.7	843538.3	836766.2	844970.8
## Feb 2019		840885.4	838214.4	843556.4	836781.3	844989.5
## Mar 2019		840902.3	838230.1	843574.5	836796.3	845008.3
## Apr 2019		840919.2	838245.8	843592.7	836811.3	845027.2
## May 2019		840936.1	838261.4	843610.9	836826.2	845046.0

```
autoplot(forecast(res.t.lineal.2 , h=12))
```



5.1.2. Tendencia cuadrática

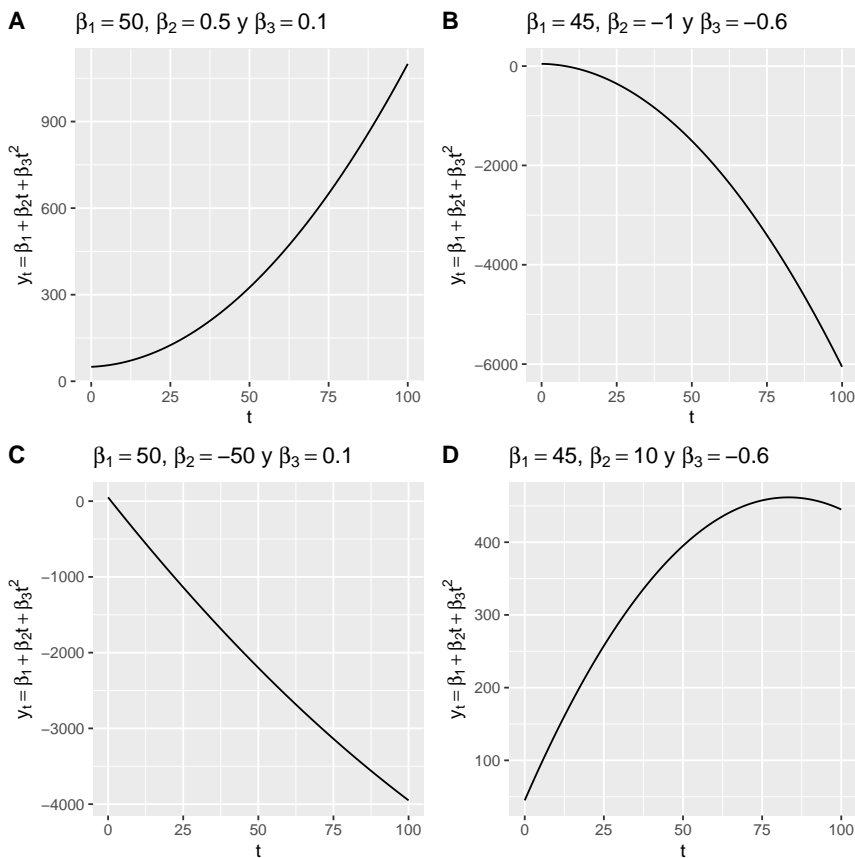
La tendencia no necesariamente tiene que ser lineal, otra opción para modelar el componente T_t de 5.1 es tener una tendencia cuadrática. Es decir, en este caso el DGP sería:

$$y_t = \beta_1 + \beta_2 t + \beta_3 t^2 + \varepsilon_t \quad (5.3)$$

donde $t = 1, 2, \dots, T$.

Esta forma funcional provee muchas mas opciones como se puede ver en la figura 5.2.

Figura 5.2 Forma funcional de la tendencia cuadrática para diferentes valores de β_1 , β_2 y β_3



Otra vez, esto se puede estimar por medio del método de mínimos cuadrados ordinarios (MCO). Noten que no hay riesgo de tener multicolinealidad perfecta, ¿Por qué?

Esto se puede estimar de la siguiente manera:

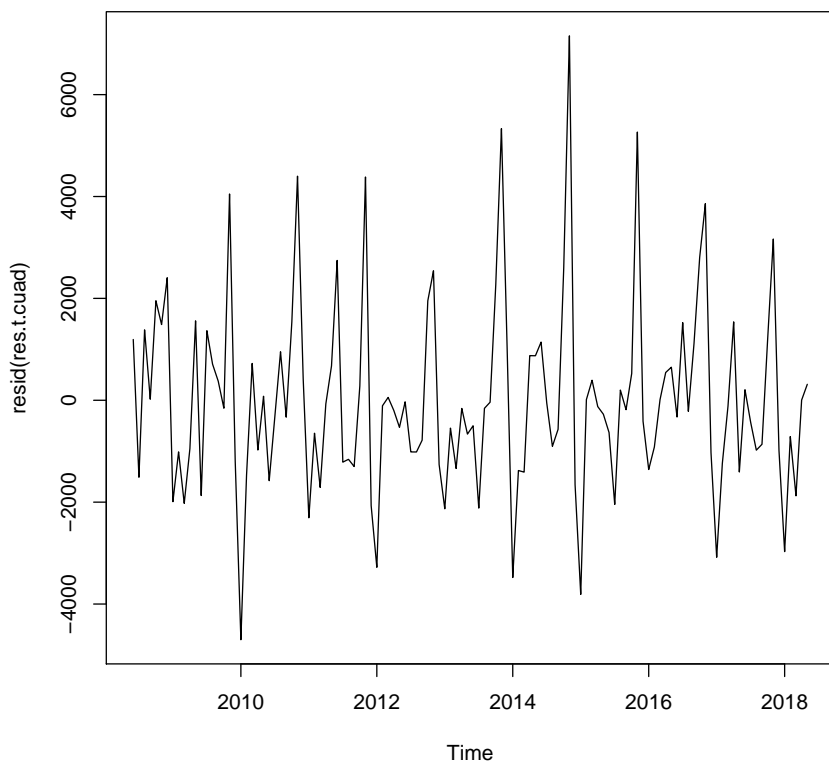
```
#res.t.cuad <- lm(TT.ts ~ t + I(t^2), data = data1)
res.t.cuad <- tslm(TT.ts ~ trend + I(trend^2), data = TT.ts)

summary(res.t.cuad)

##
## Call:
## tslm(formula = TT.ts ~ trend + I(trend^2), data = TT.ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4701.0 -1166.2  -172.4   874.1  7153.3
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.404e+05  5.300e+02  1585.661 < 2e-16 ***
## trend        -6.492e+01  2.022e+01   -3.210  0.00171 **
## I(trend^2)    6.762e-01  1.619e-01    4.177  5.71e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1903 on 117 degrees of freedom
## Multiple R-squared:  0.1976, Adjusted R-squared:  0.1839
## F-statistic: 14.41 on 2 and 117 DF, p-value: 2.546e-06
```

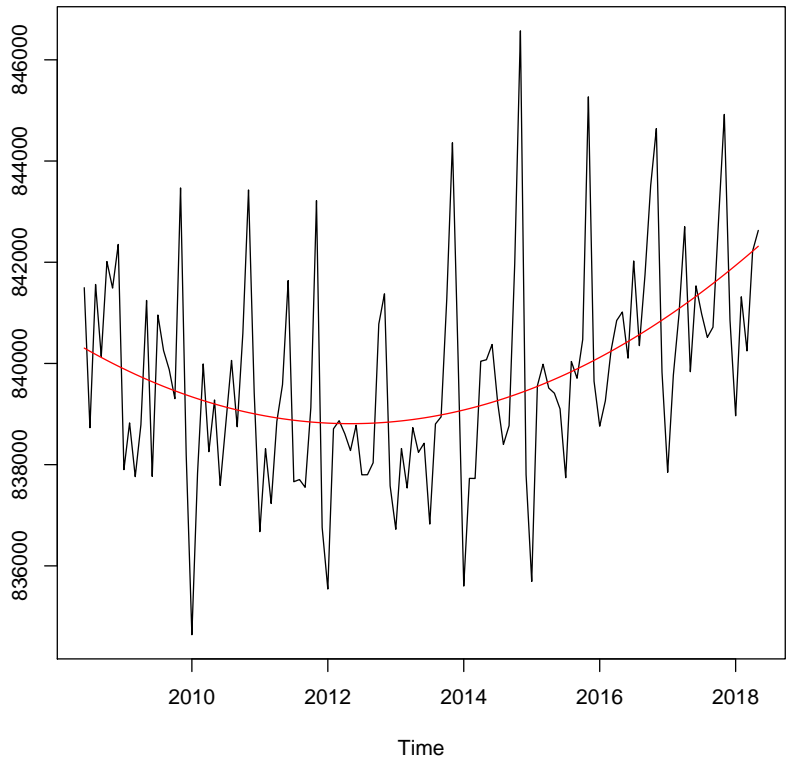
Igual que en el caso anterior, antes de proceder a analizar los resultados de esta regresión, debemos mirar si los residuos cumplen o no los supuestos del Teorema de Gauss-Markov.

```
plot(resid(res.t.cuad), type="l")
```



En este gráfico podemos ver que aún los residuales no son totalmente aleatorios. Aún se puede observar una tendencia no lineal en ellos y un componente estacional. Por lo tanto aún existe información por extraer de esta serie. Por eso no procederemos a constatar si los supuestos se cumplen, claramente estos no se cumplirán. Esto es evidente si graficamos la tendencia cuadrática encontrada (\hat{y}_t que lo guardaremos en el objeto *t.cuad*) y la serie original. Aun es evidente que el fit del modelo no es bueno.

```
t.cuad <- fitted(res.t.cuad)
ts.plot(TT.ts, t.cuad, col=1:2)
```

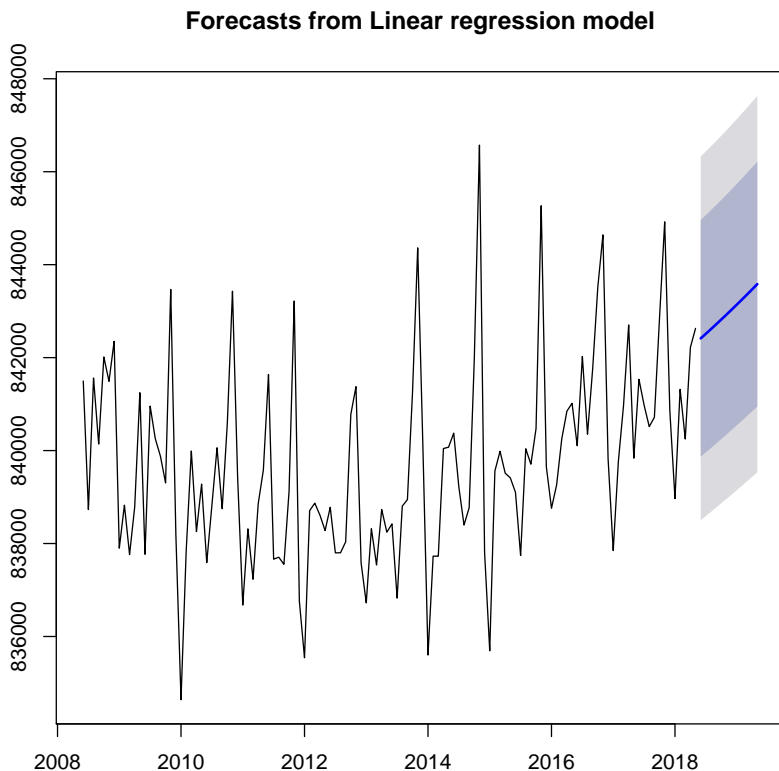


De igual manera que en el caso anterior, si se quisiera encontrar un pronóstico para los siguientes 12 meses esto se podría hacer de la siguiente manera:

```
forecast(res.t.cuad , h=12)
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## Jun 2018	842413.9	839867.9	844960.0	838501.6	846326.2
## Jul 2018	842513.3	839961.0	845065.6	838591.4	846435.2
## Aug 2018	842614.1	840055.2	845173.0	838682.0	846546.2
## Sep 2018	842716.2	840150.4	845282.1	838773.5	846659.0
## Oct 2018	842819.7	840246.5	845392.8	838865.7	846773.7
## Nov 2018	842924.5	840343.6	845505.4	838958.7	846890.3
## Dec 2018	843030.7	840441.7	845619.6	839052.4	847008.9
## Jan 2019	843138.2	840540.8	845735.6	839146.9	847129.5
## Feb 2019	843247.1	840640.8	845853.4	839242.2	847252.0
## Mar 2019	843357.3	840741.7	845972.9	839338.1	847376.5
## Apr 2019	843468.9	840843.6	846094.2	839434.7	847503.0
## May 2019	843581.8	840946.3	846217.3	839532.1	847631.6

```
plot(forecast(res.t.cuad , h=12))
```



5.1.3. Tendencia polinómica

Otra opción para estimar una tendencia no lineal es emplear un polinomio de orden superior. Por ejemplo un polinomio de orden tres correspondería a:

$$y_t = \beta_1 + \beta_2 t + \beta_3 t^2 + \beta_4 t^3 + \varepsilon_t \quad (5.4)$$

donde $t = 1, 2, \dots, T$. Y en general se puede tener un polinomio de orden p :

$$y_t = \beta_1 + \beta_2 t + \beta_3 t^2 + \beta_4 t^3 + \dots + \beta_{p+1} t^p + \varepsilon_t \quad (5.5)$$

Esta opción genera mucha más versatilidad en la tendencia.

Estimemos los modelos con tendencia polinómica de orden 3 a 8. Para facilidad podemos emplear en la fórmula la función *poly* que tiene dos argumentos: la variable para la cual se quiere crear el polinomio y el orden del polinomio.

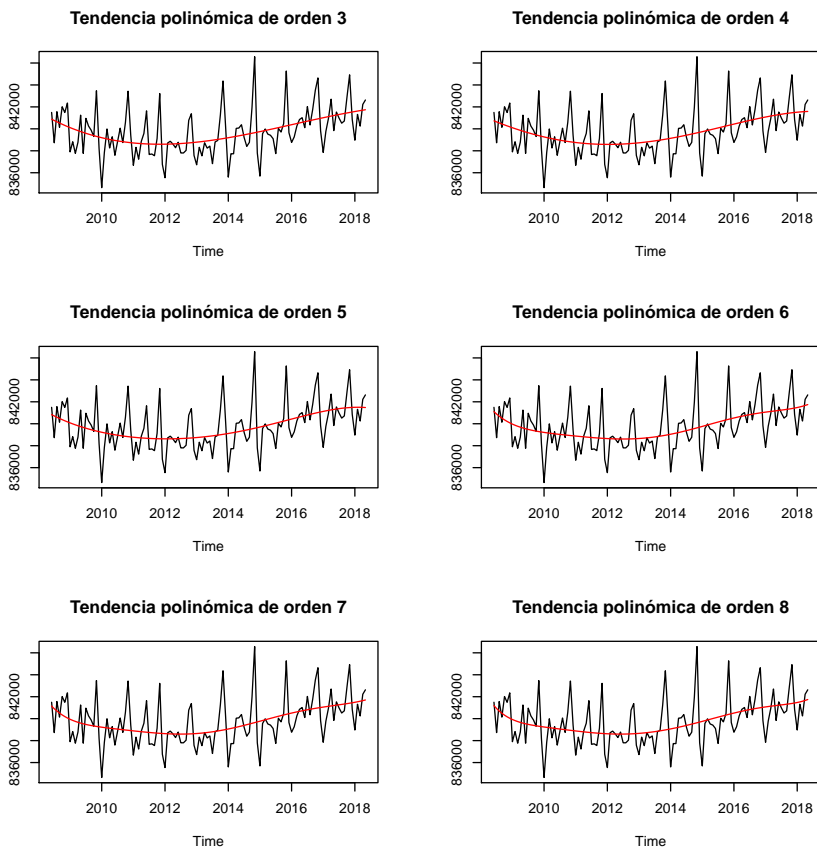
```
#res.t.cuad <- lm(TT.ts ~ t + I(t^2), data = data1)
res.t.pol.3 <- tslm(TT.ts ~ trend + I(trend^2) + I(trend^3), data = TT.ts)
res.t.pol.4 <- tslm(TT.ts ~ poly(trend, 4), data = TT.ts)
res.t.pol.5 <- tslm(TT.ts ~ poly(trend, 5), data = TT.ts)
res.t.pol.6 <- tslm(TT.ts ~ poly(trend, 6), data = TT.ts)
res.t.pol.7 <- tslm(TT.ts ~ poly(trend, 7), data = TT.ts)
res.t.pol.8 <- tslm(TT.ts ~ poly(trend, 8), data = TT.ts)
summary(res.t.pol.4 )

##
## Call:
## tslm(formula = TT.ts ~ poly(trend, 4), data = TT.ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4624.3 -1123.8  -229.3   845.2  6940.1
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)    839727.1     173.8  4830.422  < 2e-16 ***
## poly(trend, 4)1    6416.5     1904.3    3.369  0.00103 **
## poly(trend, 4)2    7949.6     1904.3    4.174  5.83e-05 ***
## poly(trend, 4)3   -2511.5     1904.3   -1.319  0.18985
## poly(trend, 4)4    -609.4     1904.3   -0.320  0.74953
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1904 on 115 degrees of freedom
## Multiple R-squared:  0.2103, Adjusted R-squared:  0.1828
## F-statistic: 7.655 on 4 and 115 DF,  p-value: 1.666e-05
```

En la siguiente gráfica podemos ver cada una de estas tendencias estimadas

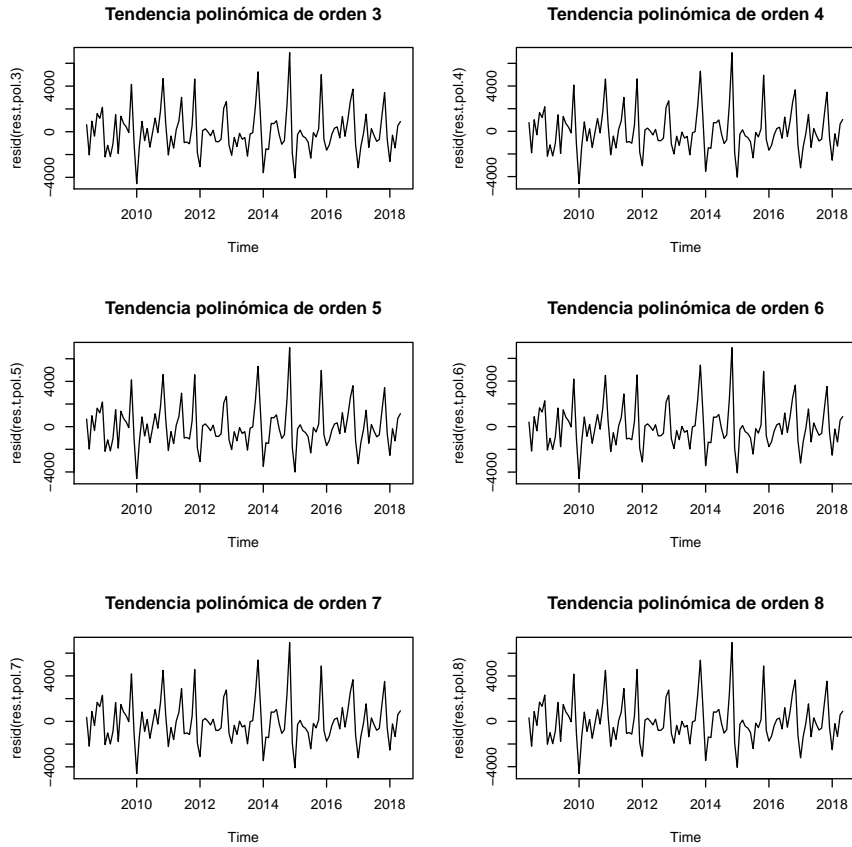
```
t.pol.3 <- fitted(res.t.pol.3)
t.pol.4 <- fitted(res.t.pol.4)
t.pol.5 <- fitted(res.t.pol.5)
t.pol.6 <- fitted(res.t.pol.6)
t.pol.7 <- fitted(res.t.pol.7)
t.pol.8 <- fitted(res.t.pol.8)

par(mfrow=c(3,2))
ts.plot(TT.ts, t.pol.3, col=1:2, main = "Tendencia polinómica de orden 3")
ts.plot(TT.ts, t.pol.4, col=1:2, main = "Tendencia polinómica de orden 4")
ts.plot(TT.ts, t.pol.5, col=1:2, main = "Tendencia polinómica de orden 5")
ts.plot(TT.ts, t.pol.6, col=1:2, main = "Tendencia polinómica de orden 6")
ts.plot(TT.ts, t.pol.7, col=1:2, main = "Tendencia polinómica de orden 7")
ts.plot(TT.ts, t.pol.8, col=1:2, main = "Tendencia polinómica de orden 8")
```



Para entender un poco el efecto de emplear estos polinomios, veamos el comportamiento de los residuales para cada uno de estos modelos.

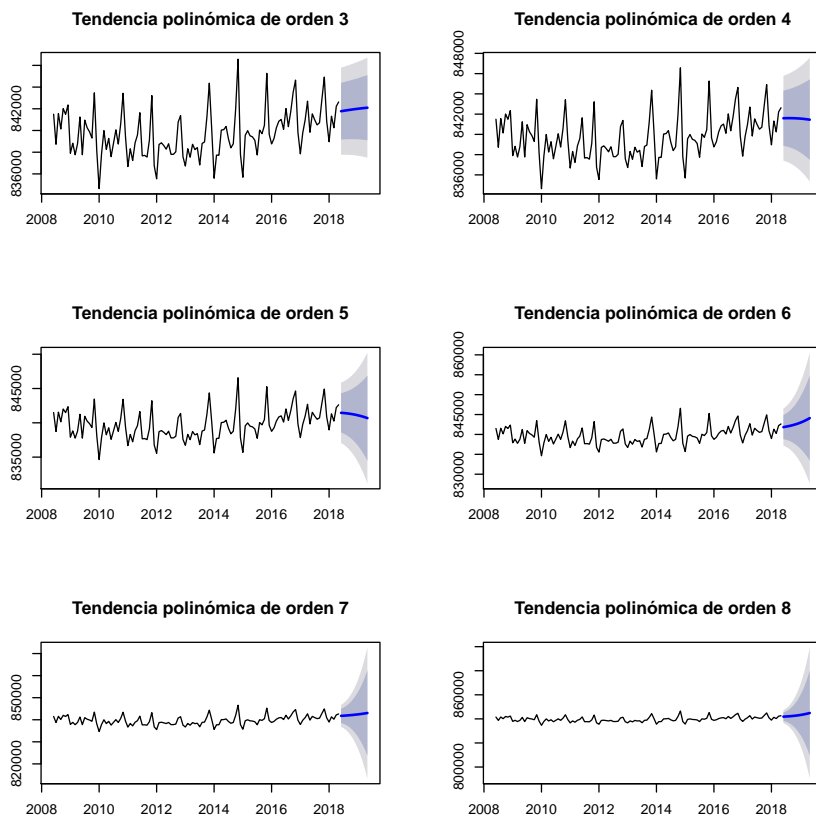
```
par(mfrow=c(3,2))
plot(resid(res.t.pol.3), type="l", main = "Tendencia polinómica de orden 3")
plot(resid(res.t.pol.4), type="l", main = "Tendencia polinómica de orden 4")
plot(resid(res.t.pol.5), type="l", main = "Tendencia polinómica de orden 5")
plot(resid(res.t.pol.6), type="l", main = "Tendencia polinómica de orden 6")
plot(resid(res.t.pol.7), type="l", main = "Tendencia polinómica de orden 7")
plot(resid(res.t.pol.8), type="l", main = "Tendencia polinómica de orden 8")
```

PaEn todos los casos, aún parece persistir un componente estacional.

Finalmente, note que la forma de construir los pronósticos es similar a lo que hemos hecho anteriormente.

```
par(mfrow=c(3,2))
plot(forecast(res.t.pol.3, h=12), main = "Tendencia polinómica de orden 3")
plot(forecast(res.t.pol.4, h=12), main = "Tendencia polinómica de orden 4")
plot(forecast(res.t.pol.5, h=12), main = "Tendencia polinómica de orden 5")
plot(forecast(res.t.pol.6, h=12), main = "Tendencia polinómica de orden 6")
plot(forecast(res.t.pol.7, h=12), main = "Tendencia polinómica de orden 7")
plot(forecast(res.t.pol.8, h=12), main = "Tendencia polinómica de orden 8")
```



Noten que los pronósticos son muy diferentes dependiendo del orden del polinomio seleccionado. Por eso será muy importante encontrar cuál es el mejor modelo. En caso de no existir la estacionalidad, deberíamos chequear los supuestos y realizar las pruebas que permitan descartar (con pruebas estadísticas conjuntas o individuales como las estudiadas anteriormente) los diferentes órdenes del polinomio. Es decir, algo equivalente a eliminar variables.

5.2. Estimación de la estacionalidad

Ahora concentremos en el componente S_t de 5.1. La estacionalidad puede ser de muchos tipos. Tal vez la más sencilla es la estacionalidad determinística. Es decir, aquella que ocurre con seguridad todos los periodos. Un tipo de estacionalidad determinística es la aditiva que puede ser modelada por medio de variables dummy que afecten el intercepto. Por ejemplo, para el caso de datos mensuales, este tipo de estacionalidad implicaría el siguiente DGP:

$$y_t = \beta_1 + \sum_{i=2}^{12} \beta_i \cdot D_{i,t} + \varepsilon_t \quad (5.6)$$

donde $D_{i,t}$ corresponde a una variable dummy que toma el valor de uno si t corresponde al mes i donde $i = 2, 3, \dots, 12$.

En *R* estimar este modelo es relativamente fácil con la función *tslm* empleando en la fórmula del modelo a estimar el componente *season*. Esto calculará automáticamente las variables dummy para cada uno de los meses como se presenta en la ecuación anterior.

```
res.t.season <- tslm(TT.ts ~ season, data = TT.ts)
summary(res.t.season)

##
## Call:
## tslm(formula = TT.ts ~ season, data = TT.ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2498.1 -1171.1  -148.6   972.3  3095.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 836834.9      463.8  1804.462 < 2e-16 ***
## season2      2128.4       655.9    3.245 0.001563 **
## season3      2222.7       655.9    3.389 0.000980 ***
## season4      3022.3       655.9    4.608 1.12e-05 ***
## season5      3124.5       655.9    4.764 5.94e-06 ***
## season6      2846.1       655.9    4.340 3.23e-05 ***
## season7      2240.5       655.9    3.416 0.000896 ***
## season8      2714.3       655.9    4.139 6.95e-05 ***
## season9      2589.8       655.9    3.949 0.000140 ***
## season10     4356.9       655.9    6.643 1.29e-09 ***
## season11     7039.2       655.9   10.733 < 2e-16 ***
## season12     2421.6       655.9    3.692 0.000350 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1467 on 108 degrees of freedom
## Multiple R-squared:  0.5602, Adjusted R-squared:  0.5154
## F-statistic: 12.5 on 11 and 108 DF, p-value: 6.416e-15
```

Otra opción para crear las variables dummy es emplear la función *seasonal.dummies* del paquete **uroot** (de Lacalle, 2019). Con esta función se pueden crear las variables dummy y después incluirlas en el objeto que contenga los datos.

```
library(uroot)
head(seasonal.dummies(TT.ts))

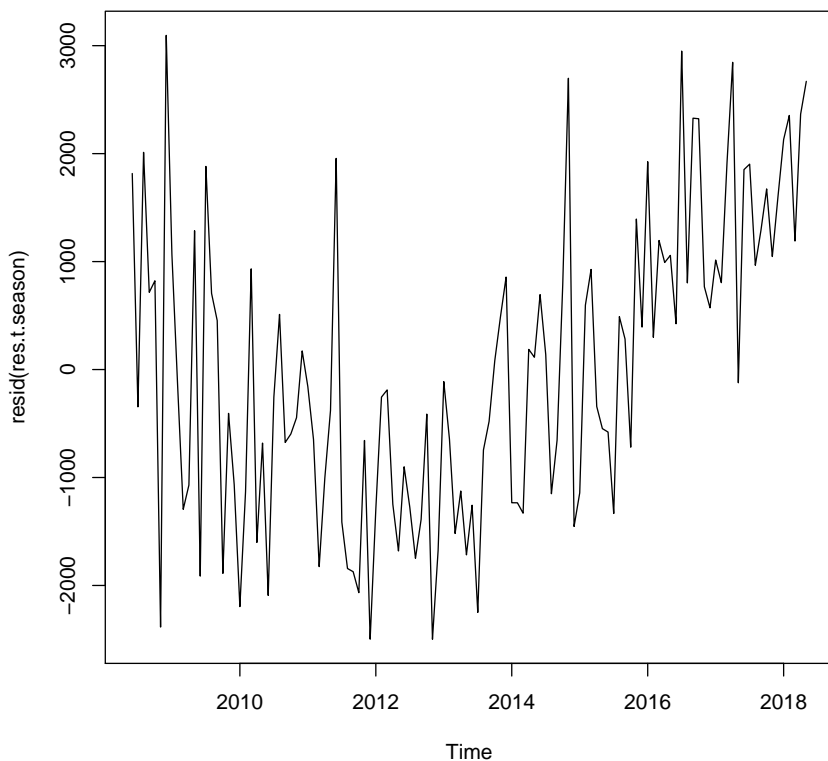
##           SD1 SD2 SD3 SD4 SD5 SD6 SD7 SD8 SD9 SD10 SD11 SD12
## Jun 2008    0  0  0  0  0  1  0  0  0  0  0  0
## Jul 2008    0  0  0  0  0  0  1  0  0  0  0  0
```

## Aug 2008	0	0	0	0	0	0	0	1	0	0	0	0
## Sep 2008	0	0	0	0	0	0	0	0	1	0	0	0
## Oct 2008	0	0	0	0	0	0	0	0	0	1	0	0
## Nov 2008	0	0	0	0	0	0	0	0	0	0	1	0

Noten que en este caso tendrán que descartar una de las variables dummy.

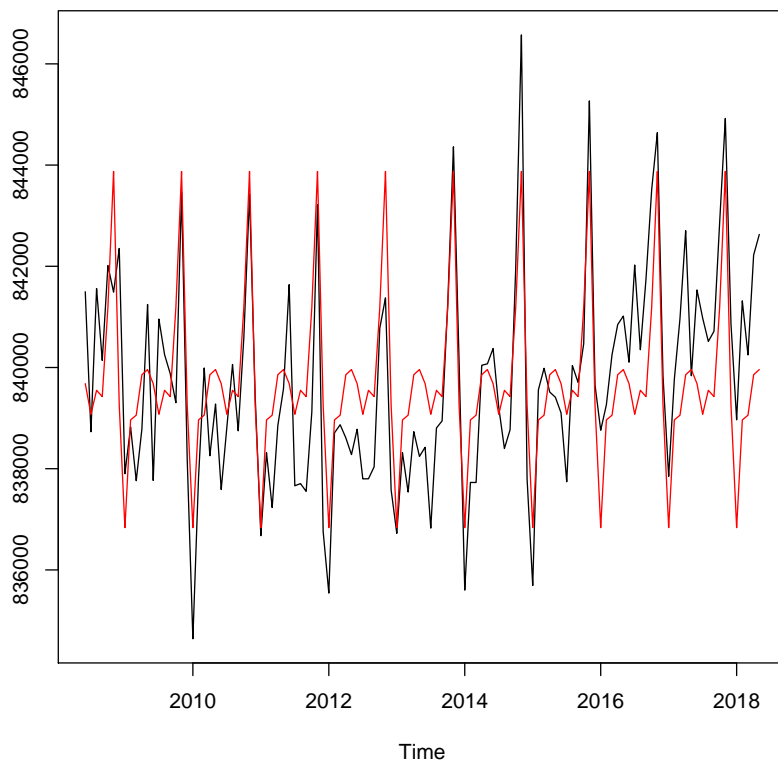
Igual que en los casos anteriores, antes de proceder a analizar los resultados de esta regresión, debemos mirar si los residuos cumplen o no los supuestos del Teorema de Gauss-Markov.

```
plot(resid(res.t.season), type="l")
```



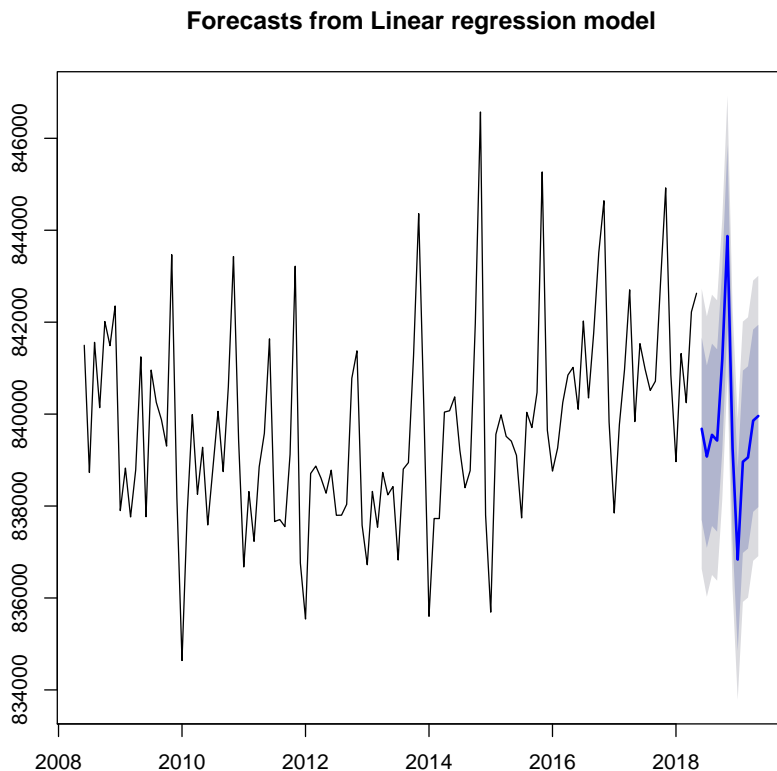
En este gráfico podemos ver que los residuos no son totalmente aleatorios. Ahora se puede observar una tendencia no lineal en ellos, pero el componente estacional parece que ya se removió. Por lo tanto aún existe información por extraer de esta serie. Por eso procederemos a eliminar la tendencia. Pero antes veamos como se comporta en muestra este modelo. Es evidente que el fit del modelo no es bueno.

```
t.season <- fitted(res.t.season)
ts.plot(TT.ts, t.season, col=1:2)
```



De igual manera que en el caso anterior, si se quisiera encontrar un pronóstico para los siguientes 12 meses esto se podría hacer de la siguiente manera:

```
plot(forecast(res.t.season , h=12))
```



5.3. Selección del mejor modelo para pronosticar

Ahora pongamos junto lo estudiado sobre la tendencia y la estacionalidad y en el Capítulo 3. Dado que tendremos diferentes opciones para calcular la tendencia, será necesario que para cada modelo se constate que los supuestos se cumplan y para aquellos modelos que se cumplan los supuestos, deberemos evaluar el comportamiento de estos fuera de muestra.

Siguiendo lo estudiando en el Capítulo 3, primero guardaremos el último año de la muestra para la evaluación fuera de muestra.

```
TT.eval <- window(TT.ts, start= c(2017, 6))
TT.in.sample <- window(TT.ts, end= c(2017, 5))
```

Dados los resultados encontrados anteriormente, compararemos modelos con tendencia polinómica de orden superior a 2 con componente estacional y hasta el orden 9. Es decir,

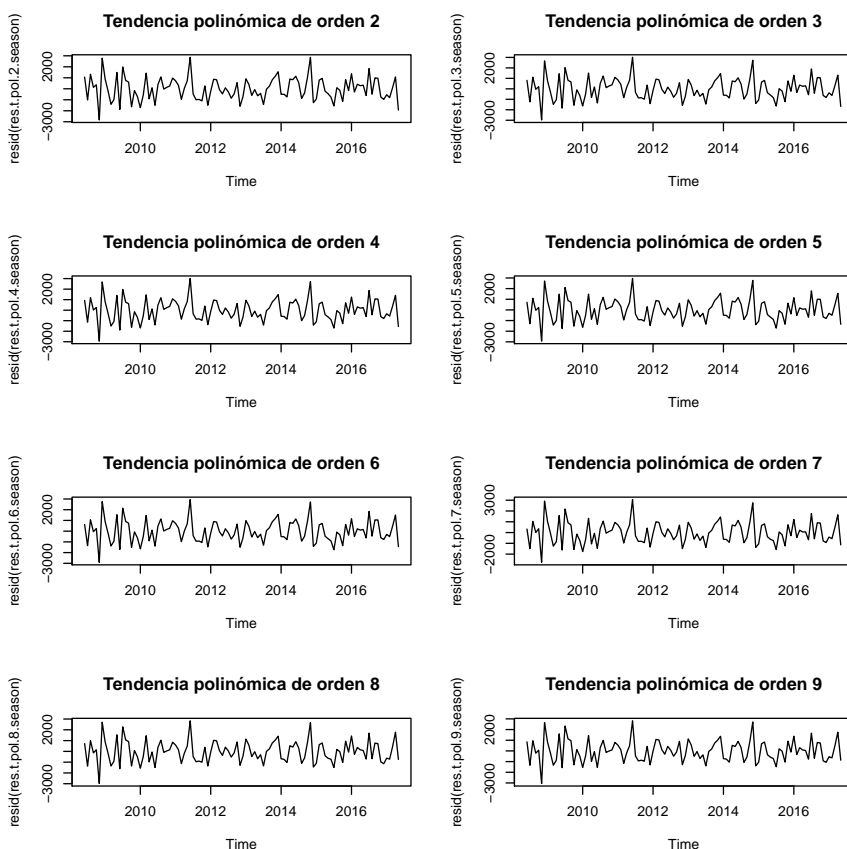
```
res.t.pol.2.season <- tslm(TT.in.sample ~ poly(trend, 2) + season, data = TT.in.sample)
res.t.pol.3.season <- tslm(TT.in.sample ~ poly(trend, 3) + season, data = TT.in.sample)
res.t.pol.4.season <- tslm(TT.in.sample ~ poly(trend, 4) + season, data = TT.in.sample)
res.t.pol.5.season <- tslm(TT.in.sample ~ poly(trend, 5) + season, data = TT.in.sample)
res.t.pol.6.season <- tslm(TT.in.sample ~ poly(trend, 6) + season, data = TT.in.sample)
res.t.pol.7.season <- tslm(TT.in.sample ~ poly(trend, 7) + season, data = TT.in.sample)
res.t.pol.8.season <- tslm(TT.in.sample ~ poly(trend, 8) + season, data = TT.in.sample)
res.t.pol.9.season <- tslm(TT.in.sample ~ poly(trend, 9) + season, data = TT.in.sample)

summary(res.t.pol.9.season )

##
## Call:
## tslm(formula = TT.in.sample ~ poly(trend, 9) + season, data = TT.in.sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3053.01  -736.59   -41.15    759.00   2816.94
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    836582.7      379.3  2205.311  < 2e-16 ***
## poly(trend, 9)1    3813.1      1142.3     3.338  0.001244 **
## poly(trend, 9)2    7396.4      1134.6     6.519  4.46e-09 ***
## poly(trend, 9)3   -1215.0      1152.3    -1.054  0.294636
## poly(trend, 9)4    -494.6      1136.2    -0.435  0.664421
## poly(trend, 9)5    -857.3      1158.7    -0.740  0.461349
## poly(trend, 9)6     310.8      1144.4     0.272  0.786580
## poly(trend, 9)7   -1082.6      1155.8    -0.937  0.351486
## poly(trend, 9)8   -1532.8      1161.8    -1.319  0.190532
## poly(trend, 9)9     475.9      1148.0     0.415  0.679487
## season2          2112.8        535.0     3.949  0.000159 ***
## season3          2350.2        535.9     4.386  3.23e-05 ***
## season4          3039.0        538.4     5.644  2.04e-07 ***
## season5          3134.9        545.1     5.752  1.29e-07 ***
## season6          2916.8        547.2     5.330  7.64e-07 ***
## season7          2284.4        540.6     4.226  5.84e-05 ***
## season8          2848.7        537.7     5.298  8.72e-07 ***
## season9          2680.0        536.4     4.996  2.99e-06 ***
## season10         4400.7        535.7     8.215  1.81e-12 ***
## season11         7151.9        535.2    13.363  < 2e-16 ***
## season12         2474.9        534.9     4.627  1.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1134 on 87 degrees of freedom
## Multiple R-squared:  0.7573, Adjusted R-squared:  0.7016
## F-statistic: 13.58 on 20 and 87 DF,  p-value: < 2.2e-16
```

Para entender un poco el efecto de emplear estos polinomios y la estacionalidad, veamos el comportamiento de los residuales para cada uno de estos modelos.

```
par(mfrow=c(4,2))
plot(resid(res.t.pol.2.season), type="l", main = "Tendencia polinómica de orden 2")
plot(resid(res.t.pol.3.season), type="l", main = "Tendencia polinómica de orden 3")
plot(resid(res.t.pol.4.season), type="l", main = "Tendencia polinómica de orden 4")
plot(resid(res.t.pol.5.season), type="l", main = "Tendencia polinómica de orden 5")
plot(resid(res.t.pol.6.season), type="l", main = "Tendencia polinómica de orden 6")
plot(resid(res.t.pol.7.season), type="l", main = "Tendencia polinómica de orden 7")
plot(resid(res.t.pol.8.season), type="l", main = "Tendencia polinómica de orden 8")
plot(resid(res.t.pol.9.season), type="l", main = "Tendencia polinómica de orden 9")
```



Ahora se nota que el componente estacional y las tendencias se han removido. El siguiente paso implicará comprobar si se cumplen los supuestos del Teorema de Gauss-Markov para los residuales. Noten que esto es equivalente a probar que ε_t es un ruido blanco.

5.3.1. Evaluación de los supuestos de los residuales

5.3.1.1. No autocorrelación

Empleemos la función *tabla.Box.Pierce* creada en Capítulo 4 (ver sección 4.3.2) para realizar la prueba de Box-Pierce y Ljung-Box para crear el Cuadro 5.1.

Cuadro 5.1 Prueba de Box-Pierce de los errores para los residuos del modelo con tendencia polinómica de orden 2

Rezagos	Box-Pierce	p-valor
1.00	0.43	0.51
2.00	1.28	0.53
3.00	4.00	0.26
4.00	5.23	0.26
5.00	5.33	0.38
6.00	5.42	0.49
7.00	5.88	0.55
8.00	6.05	0.64
9.00	6.05	0.73
10.00	6.14	0.80
11.00	6.74	0.82
12.00	10.89	0.54
13.00	12.00	0.53
14.00	12.06	0.60
15.00	13.26	0.58
16.00	15.07	0.52
17.00	15.23	0.58
18.00	15.61	0.62
19.00	16.21	0.64
20.00	16.21	0.70

El lector puede constatar que para todos los modelos no existe autocorrelación.

5.3.1.2. No comportamiento heteroscedástico

Como se discutió en el Capítulo 4 (ver sección 4.3.3), una aproximación para determinar si existe un comportamiento GARCH o ARCH es emplear la prueba de Ljung-Box sobre la serie (sin media) al cuadrado. En este caso al ser residuales de una regresión las series ya tienen media cero. El Cuadro 5.2 presenta los resultados para los residuales al cuadrado del modelo de tendencia polinómica de orden 3.

En este caso, el lector también puede constatar que los residuales de todos los modelos con tendencia polinómica y estacionalidad considerados no tienen comportamiento GARCH o ARCH.

Cuadro 5.2 Prueba de Ljung-Box de los errores al cuadrado para los residuos del modelo con tendencia polinómica de orden 2

Rezagos	Ljung-Box	p-valor
1.00	2.23	0.14
2.00	3.44	0.18
3.00	3.51	0.32
4.00	3.51	0.48
5.00	3.52	0.62
6.00	5.45	0.49
7.00	6.32	0.50
8.00	6.97	0.54
9.00	8.04	0.53
10.00	8.07	0.62
11.00	8.61	0.66
12.00	8.64	0.73
13.00	8.75	0.79
14.00	8.84	0.84
15.00	9.05	0.87
16.00	9.11	0.91
17.00	9.21	0.93
18.00	9.79	0.94
19.00	9.80	0.96
20.00	10.08	0.97

5.3.1.3. Normalidad

En este caso la normalidad será necesaria para poder construir intervalos de confianza para los pronósticos empleando la aproximación tradicional. En caso que el supuesto no se cumpla, se podrá construir intervalos de confianza empleando la metodología de bootstrapping.

```
library(tseries)
jarque.bera.test(resid(res.t.pol.2.season))

##
##  Jarque Bera Test
##
## data:  resid(res.t.pol.2.season)
## X-squared = 1.4709, df = 2, p-value = 0.4793

shapiro.test(resid(res.t.pol.2.season))

##
##  Shapiro-Wilk normality test
##
## data:  resid(res.t.pol.2.season)
## W = 0.98564, p-value = 0.3006
```

Ustedes pueden constatar que así como para la tendencia polinómica de orden 2, para los modelos de orden 3 a 9 no se puede rechazar la nula de normalidad.

Así compararemos el comportamiento fuera de muestra de los modelos con tendencia polinómica de orden 2 a 9

5.3.1.4. Comparación de los modelos fuera de muestra

Calculemos los pronósticos para cada una de los ordenes considerados y sus respectivos *RMSE*. Para esto podemos crear el siguiente loop.

```
RMSE <- as.data.frame(matrix(NA,nrow=1,ncol=length(2:9)))

for (j in 2:9) {

  colnames(RMSE)[j-1] <- noquote(paste("pol",j, "season", sep = "."))

  Pronosticos <- forecast(tslm(TT.in.sample ~ poly(trend, order = j) +
                                season, TT.in.sample),h=12)$mean

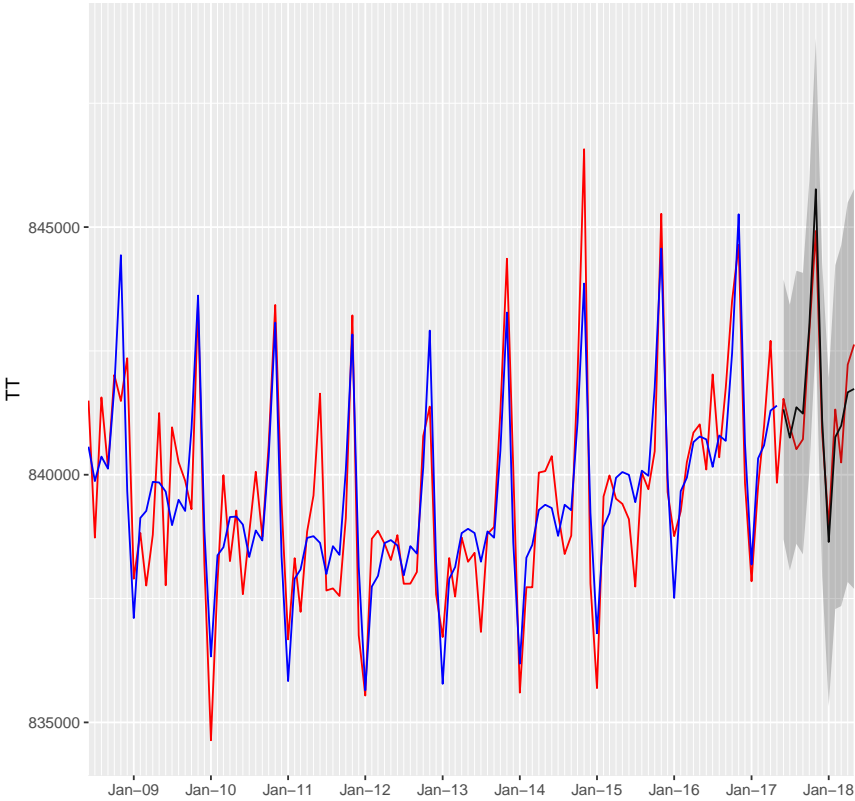
  RMSE[,j-1] <- accuracy(Pronosticos, TT.eval)["Test set", "RMSE"]
}
RMSE
```

##	pol.2.season	pol.3.season	pol.4.season	pol.5.season
## 1	1065.279	659.2347	574.0859	1330.168
##	pol.6.season	pol.7.season	pol.8.season	pol.9.season
## 1	646.1447	5103.975	16860.65	10228.58

En este caso encontramos que el mínimo *RMSE* corresponde al polinomio de orden 4.

El ajuste de ese modelo lo encontramos en la figura 5.3.

Figura 5.3 Pronóstico en muestra de evaluación y de estimación de las ventas de tiquetes terrestres (negro=pronóstico, azul=estimado, rojo=datos, sombra=95 % conf.)
(black=forecast, blue=fitted, red=data, shadow=95% conf. interval)



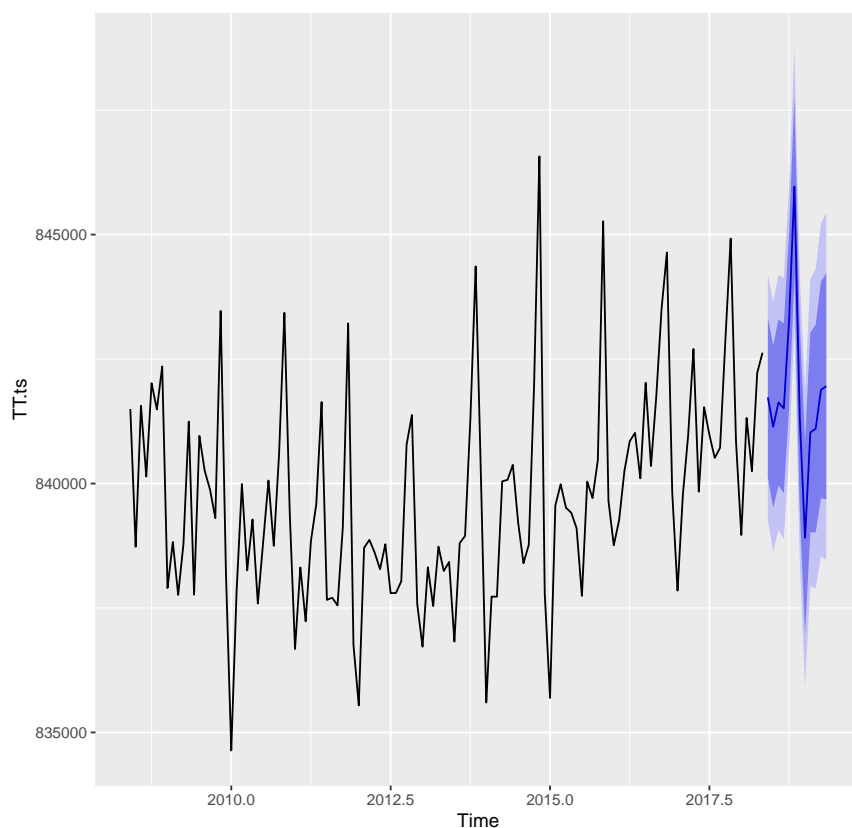
Las proyecciones de este mejor modelo para el siguiente año serán:

```
pronostico <- forecast(tslm(TT.ts ~ poly(trend, order = j) + season, TT.ts),h=12)
pronostico
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## Jun 2018	841731.3	840129.9	843332.8	839269.0	844193.7
## Jul 2018	841141.6	839509.9	842773.2	838632.8	843650.3
## Aug 2018	841627.3	839960.3	843294.2	839064.2	844190.3
## Sep 2018	841510.7	839802.9	843218.4	838885.0	844136.4
## Oct 2018	843281.6	841527.3	845036.0	840584.2	845979.0
## Nov 2018	845963.6	844156.5	847770.8	843185.0	848742.2
## Dec 2018	841341.5	839475.0	843208.0	838471.7	844211.4
## Jan 2019	838911.1	836978.5	840843.8	835939.6	841882.7
## Feb 2019	841026.4	839020.5	843032.3	837942.2	844110.6
## Mar 2019	841103.1	839016.6	843189.6	837895.0	844311.2
## Apr 2019	841880.5	839705.9	844055.2	838536.9	845224.2
## May 2019	841956.0	839685.4	844226.6	838464.9	845447.2

Y el correspondiente grafico se reporta en la Figura 5.4

Figura 5.4 Pronóstico de las ventas de tiquetes terrestres para el siguiente año con el modelo seleccionado



5.4. Ejercicio

Empleando la información de la tasa de desempleo y el número de ocupados en miles de personas (Ocupados) para las 13 principales ciudades disponibles en el archivo *datosEmpleo.xlsx*, encuentre el mejor pronóstico para los próximos 6 meses de cada una de las variables empleando tanto los métodos de este Capítulo. Compare los resultados con el mejor modelo encontrado en el cuerpo y ejercicio del capítulo 3. Escriba un breve informe de máximo una página de texto que explique cómo llega a sus proyecciones y presente las proyecciones. Aclare en el texto cuáles serían las limitaciones de sus pronósticos.

Capítulo 6

Modelos Autoregresivos (AR)

En el capítulo anterior se discutió la estacionariedad de una serie de tiempo. En especial, se definió un proceso estacionario, como aquel que tiene una media y varianza que no cambian en el tiempo y que la covarianza no depende del tiempo. En este capítulo nos dedicaremos a estudiar una familia de procesos conocidos como los procesos Auto-Regresivos o también conocidos como AR por su sigla.

La característica principal de esta familia de procesos es el de suponer que la historia de la variable aleatoria afecta su comportamiento futuro y presente de la serie. En general una secuencia de variables aleatorias

$$\{y_t | t \in 1, 2, \dots, T\}$$

se definirá como un proceso autoregresivo de orden p ($AR(p)$) si el valor de la secuencia hoy (y_t) corresponde a la suma de i) un promedio ponderado de los últimos p periodos, ii) un término de error y iii) en algunas ocasiones una constante. En otras palabras,

$$y_t = \delta + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t$$

Es decir,

$$y_t = \delta + \sum_{i=1}^p \phi_i y_{t-i} + \varepsilon_t$$

A continuación estudiaremos las características y propiedades de los procesos AR, empezando por los más sencillos. Primero prestaremos atención a los procesos $AR(1)$ y $AR(2)$, para luego estudiar el caso más general $AR(p)$.

6.1. Proceso $AR(1)$

En general, los proceso $AR(1)$ se caracterizan porque el valor actual de serie $\{y_t\}$ es función del valor inmediatamente pasado de la serie, de un término de error y de una constante. En otras palabras,

$$y_t = \delta + \phi_1 y_{t-1} + \varepsilon_t \quad (6.1)$$

Remplazando ¹ y_{t-1} en la anterior expresión obtenemos:

$$y_t = \delta + \phi_1 (\delta + \phi_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t$$

$$y_t = \delta(1 + \phi_1) + \phi_1^2 y_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t$$

y remplazando y_{t-2} obtenemos:

$$y_t = \delta(1 + \phi_1 + \phi_1^2) + \phi_1^3 y_{t-3} + \phi_1 \varepsilon_{t-1} + \phi_1^2 \varepsilon_{t-2} + \varepsilon_t$$

Si seguimos remplazando recurrentemente, tendremos²:

$$y_t = \frac{\delta}{(1 - \phi_1)} + \sum_{i=0}^{\infty} \phi_1^i \varepsilon_{t-i} \quad (6.2)$$

Es fácil demostrar que el proceso $AR(1)$ tiene las siguientes media, varianza y autocovarianza:

$$E[y_t] = \frac{\delta}{(1 - \phi_1)} = \mu \quad (6.3)$$

$$Var[y_t] = \frac{\sigma_\varepsilon^2}{1 - \phi_1^2} = \gamma_0 \quad (6.4)$$

$$Cov[y_t, y_{t-s}] = \phi_1^s \gamma_0 = \frac{\phi_1^s}{1 - \phi_1^2} \sigma_\varepsilon^2 = \gamma_s \quad (6.5)$$

De estas expresiones, es claro que el proceso $AR(1)$ será estacionario siempre y cuando $|\phi_1| < 1$. De hecho, en el caso que $|\phi_1| \geq 1$, entonces cualquier choque (o perturbación) se acumulará periodo tras periodo de tal manera que la serie (y su media) crecerá en el tiempo sin ningún límite, por eso no será un proceso estacionario³.

Antes de continuar, consideramos nuevamente la parte no aleatoria del proceso $AR(1)$ y descartemos la constante, es decir

$$y_t = \phi_1 y_{t-1}$$

$$y_t - \phi_1 y_{t-1} = 0$$

con esta expresión podemos construir lo que se conoce como el polinomio característico del proceso que corresponde a⁴

¹ Noten que $y_{t-1} = \delta + \phi_1 y_{t-2} + \varepsilon_{t-1}$.

² Para llegar a esta expresión empleamos una serie de Taylor. Recuerden que una serie de Taylor implica que si $|\phi| < 1$, entonces $1 + \phi + \phi^2 + \phi^3 + \dots = \sum_{i=0}^{\infty} \phi^i$ tiende a $\frac{1}{1-\phi}$.

³ Mas adelante, cuando se discutan los procesos no estacionarios se elaborará más sobre el tema.

⁴ Este polinomio se construye remplazando la serie de tiempo en el periodo t por el número 1, la serie en el periodo $t-1$ por una variable z , la serie en el periodo $t-2$ por z^2 , la serie en el periodo $t-3$ por z^3 y así sucesivamente.

$$1 - \phi_1 z = 0$$

La condición para estacionariedad del proceso AR(1) ($|\phi_1| < 1$) es equivalente a decir que la raíz del polinomio característico es mayor que uno⁵. En otras palabras, dado el polinomio característico de un proceso AR(1):

$$(1 - \phi_1 z)$$

Si encontramos la raíz de ese polinomio⁶, tenemos que:

$$(1 - \phi_1 z) = 0$$

$$\phi_1 z = 1$$

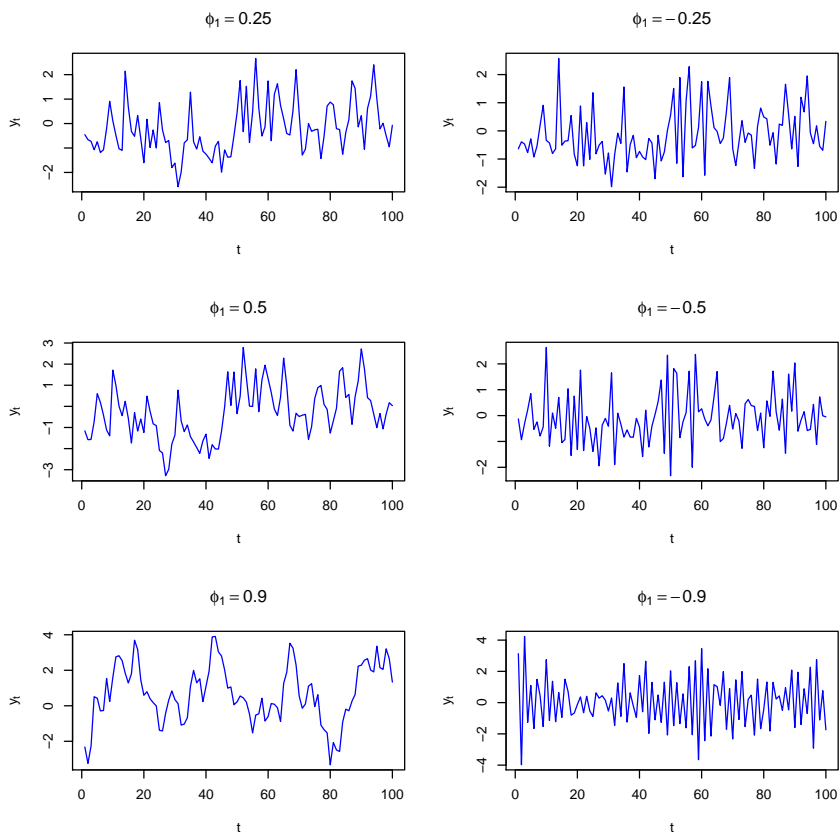
$$z = \frac{1}{\phi_1}$$

Por lo tanto, la condición $|\phi_1| < 1$, implica que $z > 1$. Es decir, la condición para que un proceso AR(1) sea estacionario es que $|\phi_1| < 1$ o, lo que es equivalente, que la raíz del polinomio característico sea mayor que cero.

⁵ En la literatura, se emplea la expresión, “la raíz está por fuera del círculo unitario”. Más adelante se aclarará el origen de esta expresión.

⁶ Es decir, encontramos el valor de z que satisface $(1 - \phi_1 z) = 0$.

Figura 6.1 Procesos AR(1) simulados para diferentes valores ϕ_1 y las mismas realizaciones del proceso ruido blanco (ε_t).



En la figura 6.1 se presentan 100 realizaciones de seis diferentes procesos $AR(1)$, cada uno emplea un ϕ_1 diferente, pero las mismas realizaciones de ε_t . La característica predominante en estos procesos es que no importa la magnitud de la perturbación (ε_t) el proceso tiende regresar a su media, esta característica se conoce como reversión a la media (en inglés mean reverting).

Función de Autocorrelación (ACF) (teórica)

La covarianza es una medida de la relación lineal entre dos variables aleatorias que depende del valor en que esté medida la serie. Convencionalmente, como medida de la relación lineal entre variables aleatorias se emplea el coeficiente de correlación que está libre de unidades y se encuentra entre uno y cero. Análogamente, a la correlación, podemos encontrar la autocorrelación, a partir de la autocovarianza, para diferentes rezagos. En general, la autocorrelación para un proceso con varianza constante corresponde a:

$$\rho_s = \frac{Cov(y_t, y_{t-s})}{\sqrt{Var(y_t) Var(y_{t-s})}} = \frac{Cov(y_t, y_{t-s})}{\sqrt{(Var(y_t))^2}} = \frac{\gamma_s}{\gamma_0}$$

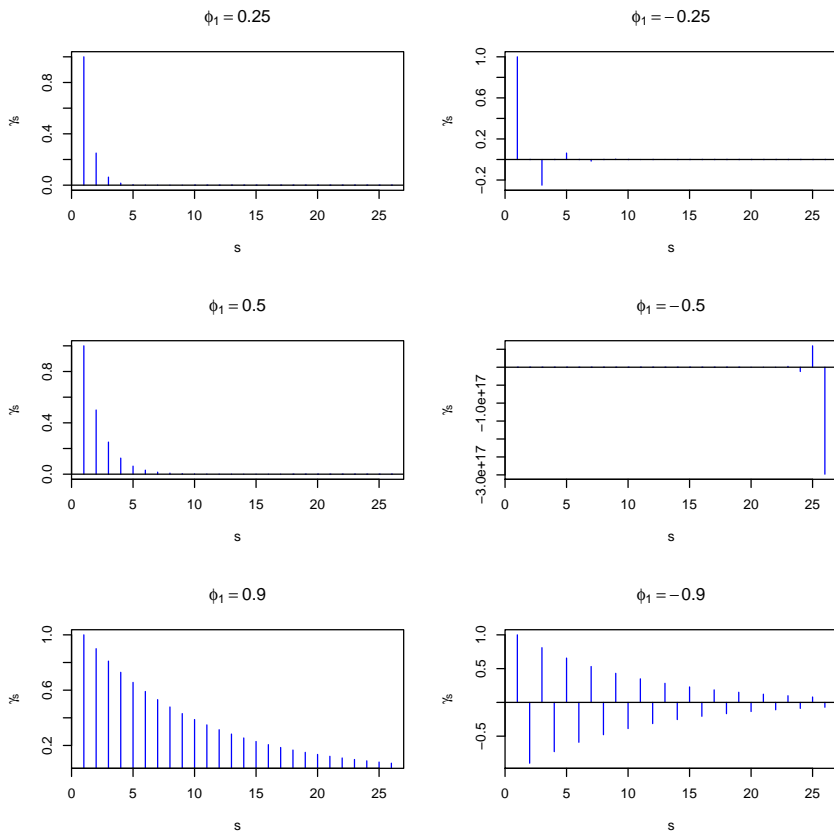
Así, la función que relaciona la autocorrelación y el número de rezagos se denomina la función de autocorrelación. Dos propiedades importantes de la función de autocorrelación:

- $\rho_0 = 1$
- $\rho_{-s} = \rho_s$

A partir de 6.4 y 6.5 podemos construir la función de autocorrelación (ACF) teórica para un proceso AR(1). En este caso,

$$\rho_s = \frac{\phi_1^s \gamma_0}{\gamma_0} = \phi_1^s \quad (6.6)$$

De esta manera la ACF teórica para un proceso AR(1) estacionario tenderá a cero a medida que se consideren mas rezagos (s crece), dado que $|\phi_1| < 1$. Si ϕ_1 es positivo (y menor que 1), entonces la ACF teórica decaerá exponencialmente a cero (ver figura 6.2). Por el contrario sí ϕ_1 es negativo (y mayor que -1), entonces la ACF teórica decaerá exponencialmente pero de manera oscilante a cero (ver figura 6.2).

Figura 6.2 AFC teórica para procesos AR(1) con diferentes valores ϕ_1 .

6.2. Proceso AR(2)

Un proceso $AR(2)$ corresponde a una serie de tiempo tal que su DGP corresponde a:

$$y_t = \delta + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t \quad (6.7)$$

En este caso, es fácil mostrar que⁷:

$$\mu = \frac{\delta}{1 - \phi_1 - \phi_2} \quad (6.8)$$

⁷ Ver la respuesta del Ejercicio 1 del Capítulo de Estacionariedad para una demostración de este resultado.

$$\gamma_s = \begin{cases} \frac{(1-\phi_2)\sigma_\varepsilon^2}{(1+\phi_2)[(1-\phi_2)^2-\phi_1^2]} & s = 0 \\ \frac{\phi_1\gamma_0}{1-\phi_2} & s = 1, -1 \\ \phi_1\gamma_{s-1} + \phi_2\gamma_{s-2} & o.w. \end{cases} \quad (6.9)$$

En este caso una condición necesaria pero no suficiente para que el proceso AR(2) sea estacionario es $\phi_1 + \phi_2 < 1$. Pero una condición necesaria y suficiente para la estacionariedad de un proceso AR(2) es que las raíces del polinomio característico estén por fuera del círculo unitario⁸. En otras palabras necesitamos que las dos posibles soluciones de:

$$1 - \phi_1 z - \phi_2 z^2 = 0 \quad (6.10)$$

sean mayores que uno en valor absoluto⁹ ($|z_j| > 1$).

Ejemplo 1: ¿Es el siguiente proceso AR(2) estacionario?

Consideremos el siguiente proceso:

$$y_t = \delta + 0,25y_{t-1} + 0,125y_{t-2} + \varepsilon_t$$

¿Es este proceso estacionario?

Respuesta: Noten que los coeficientes autoregresivos de este proceso suman menos que uno (en valor absoluto), así se cumple una condición necesaria pero no suficiente. Para determinar si el proceso AR(2) es estacionario o no, solo tenemos que encontrar las raíces del polinomio característico. En este caso el polinomio característico es:

$$[1 - 0,25z - 0,125z^2]$$

Así, debemos resolver el siguiente polinomio característico:

$$\left[1 - \frac{1}{4}z - \frac{1}{8}z^2\right] = 0$$

Factorizando, se tiene que:

$$\left(1 - \frac{1}{2}z\right)\left(1 + \frac{1}{4}z\right) = 0$$

Esto implica que $(1 - \frac{1}{2}z) = 0$ o $(1 + \frac{1}{4}z) = 0$. Es decir, $z = 2$ o $z = -4$. Ambas raíces están por fuera del círculo unitario, por tanto el proceso AR(2) es estacionario.

⁸ Es decir, que el valor absoluto de las raíces sean mayores que cero.

⁹ En caso de obtener un par de raíces complejas de la forma $z_j = z_a + iz_b$, entonces se debe cumplir que $\sqrt{z_a^2 + z_b^2} = |z_j| > 1$. En los siguientes dos ejemplos se presenta una representación intuitiva de este concepto.

Ejemplo 2: ¿Es el siguiente proceso AR(2) estacionario?

Consideremos el siguiente proceso:

$$y_t = \delta + 0,25y_{t-2} + \varepsilon_t$$

¿Es este proceso estacionario?

Respuesta: En este caso el polinomio característico:

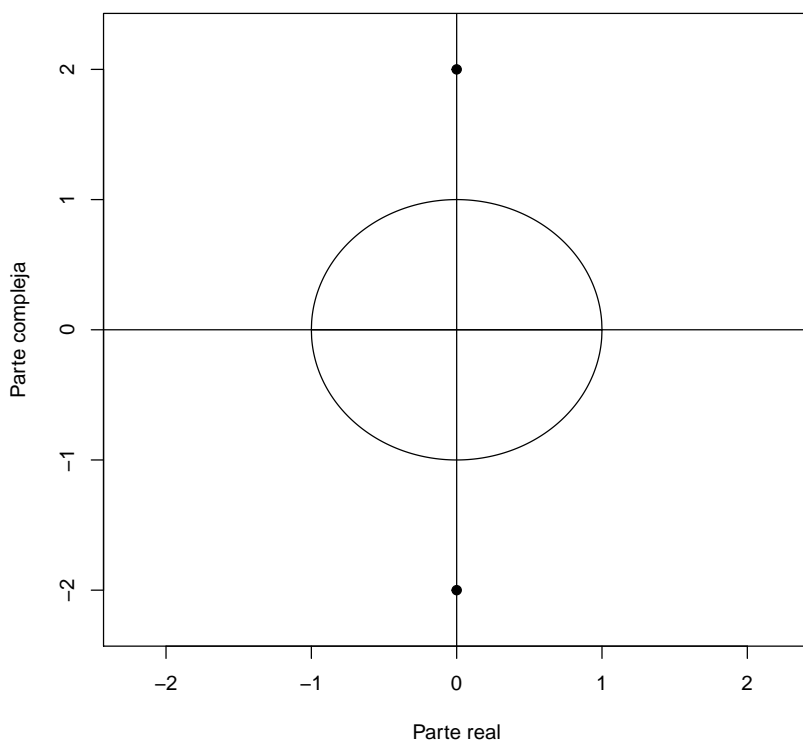
$$\left[1 + \frac{1}{4}z^2\right] = 0$$

Factorizando, se tiene que:

$$\left(1 - \frac{1}{2}i \cdot z\right) \left(1 + \frac{1}{2}i \cdot z\right) = 0$$

Esto implica que $1 - \frac{1}{2}i \cdot z = 0$ o $1 + \frac{1}{2}i \cdot z = 0$. Es decir, $z = 2i$ o $z = -2i$. Noten que $z_1 = z_a + iz_b = 0 + 2i$, por tanto $\sqrt{z_a^2 + z_b^2} = \sqrt{0 + 2^2} = |2| > 1$ tenemos que las raíces están por fuera del círculo unitario. En la figura 6.3 se representan las raíces de este proceso.

Figura 6.3 Raíces imaginarias del proceso $y_t = \delta + 0,25y_{t-2} + \varepsilon_t$.

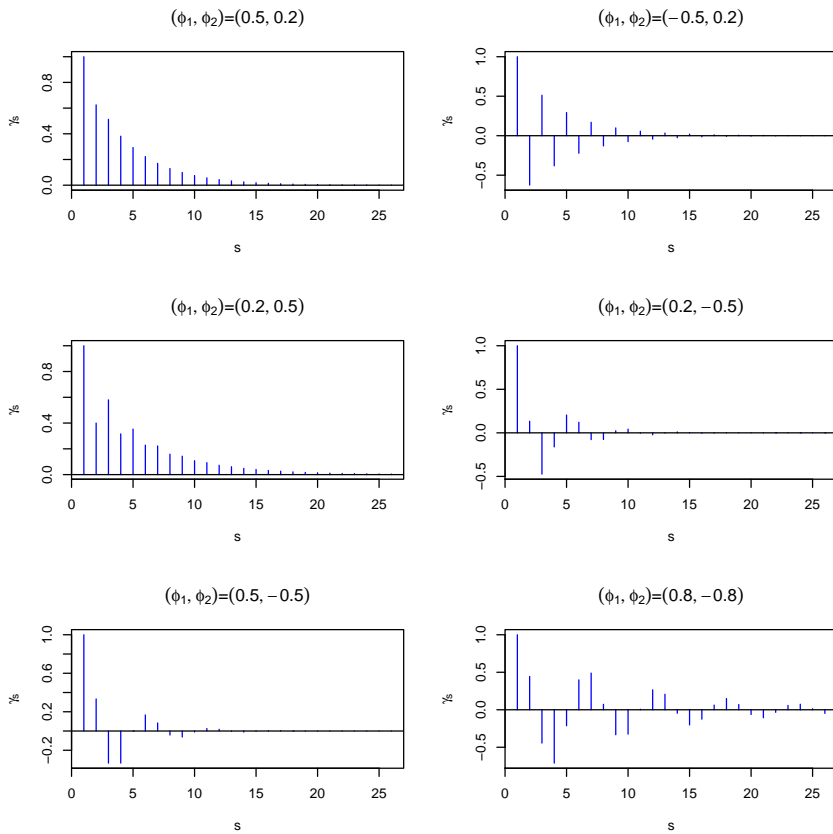


Empleando 6.9 podemos derivar la ACF de un proceso AR(2), es fácil demostrar que:

$$\rho_s = \begin{cases} 1 & s = 0 \\ \frac{\phi_1}{1-\phi_2} & s = 1, -1 \\ \phi_1 \rho_{s-1} + \phi_2 \rho_{s-2} & o.w. \end{cases} \quad (6.11)$$

En la figura 6.4 se presentan 6 diferentes combinaciones de valores de ϕ_1 y ϕ_2 . Como se puede observar, el comportamiento del ACF depende de los valores de estos dos parámetros; en todo caso, la ACF tiende a cero a medida que el número de rezagos considerados aumenta, pero menos rápido que en el caso del proceso AR(1).

Figura 6.4 AFC teórica para procesos AR(2) con diferentes valores ϕ_1 y ϕ_2 .



6.3. Proceso $AR(p)$

En las dos secciones anteriores estudiamos bajo que condiciones son los procesos $AR(1)$ y $AR(2)$ estacionarios, adicionalmente encontramos el comportamiento declinante de la ACF para ambos casos. En esta sección generalizaremos nuestros resultados. Empecemos por definir un proceso $AR(p)$ como:

$$y_t = \delta + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (6.12)$$

Para este proceso tenemos que:

$$\mu = \frac{\delta}{1 - \phi_1 - \phi_2 - \dots - \phi_p} \quad (6.13)$$

$$\gamma_s = \begin{cases} \phi_1 \gamma_1 + \phi_2 \gamma_2 + \dots + \phi_p \gamma_p + \sigma_\varepsilon^2 & s = 0 \\ \phi_1 \gamma_{s-1} + \phi_2 \gamma_{s-2} + \dots + \phi_p \gamma_{s-p} & s = 1, 2, \dots \end{cases} \quad (6.14)$$

Una condición necesaria, pero no suficiente para que un proceso $AR(p)$ sea estacionario es:

$$\phi_1 + \phi_2 + \dots + \phi_p < 1$$

La condición suficiente para que un proceso $AR(p)$ sea estacionario es que las raíces del polinomio característico:

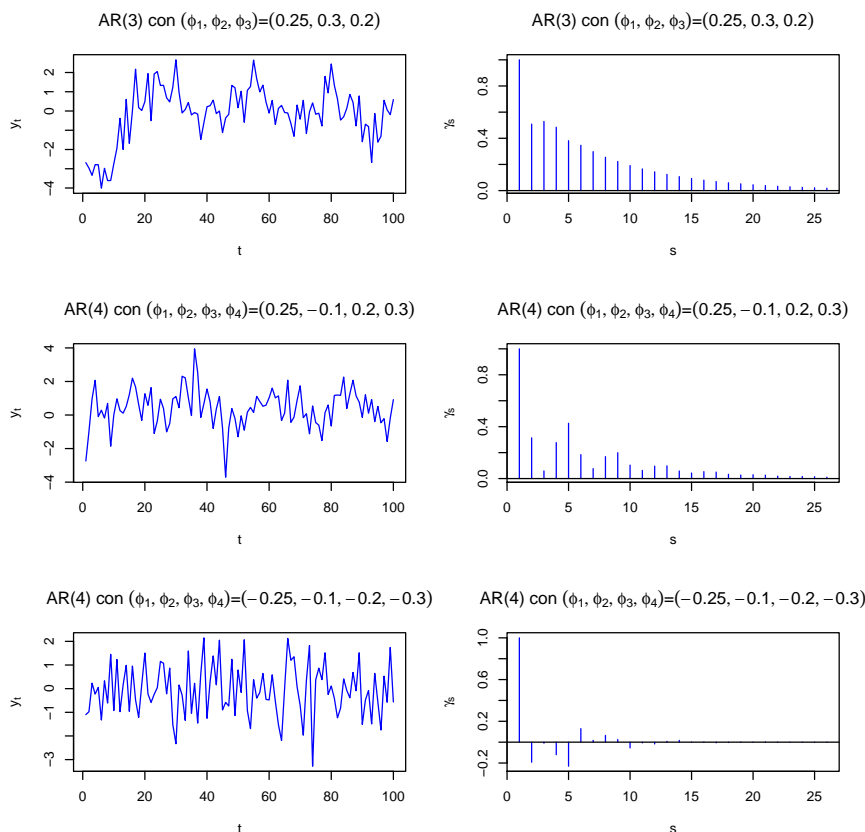
$$\phi_p(z) = (1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p) = 0$$

estén por fuera del círculo unitario. Es decir, para el caso de raíces reales necesitamos que $|z_j| > 1$ y para el caso de raíces complejas de la forma $z_j = z_a + iz_b$, entonces se debe cumplir que $\sqrt{z_a^2 + z_b^2} = |z_j| > 1$. Por otro lado, empleando 6.14 podemos encontrar la siguiente expresión general para la ACF de un $AR(p)$ que es conocida como las ecuaciones de Yule-Walker:

$$\rho_s = \phi_1 \rho_{s-1} + \phi_2 \rho_{s-2} + \dots + \phi_p \rho_{s-p} \quad (6.15)$$

para $s = 1, 2, \dots$. Empleando las ecuaciones de Yule-Walker permiten crear un sistema de ecuaciones del cual se puede encontrar cada una de las autocorrelaciones.

En la figura 6.5 se presentan tres muestras simuladas a partir de procesos AR y sus correspondientes ACF teóricas. Se presenta un proceso $AR(3)$ y dos $AR(4)$. Este gráfico permite entender una característica. En general, al observar las realizaciones de un proceso $AR(p)$ o su ACF será imposible determinar su orden.

Figura 6.5 Series simuladas y AFC teórica para diferentes procesos AR(P).

6.4. Función de autocorrelación parcial (PACF)

Como se discutió anteriormente, no existe una característica única observable en la ACF que permita diferenciar un proceso AR de otro. La razón está en la misma naturaleza de los procesos AR.

Por ejemplo, para el caso de un proceso $AR(1)$, la observación actual está relacionada con las anteriores de manera indirecta vía y_{t-1} . Es decir, existe una relación entre y_t y y_{t-2} , porque y_t depende de y_{t-1} y y_{t-1} depende de y_{t-2} . Así, y_t y y_{t-2} están relacionadas así y_{t-2} no está presente en el DGP de y_t para el caso de un proceso $AR(1)$.

Así, la ACF mide tanto la correlación directa e indirecta entre rezagos de una serie de tiempo. Para deshacernos de la relación indirecta entre rezagos, se puede emplear la autocorrelación parcial. La función de autocorrelación parcial (PACF por su sigla en inglés) mide la relación “neta” entre dos diferentes rezagos, eliminando

las posibles relaciones indirectas. En otras palabras, la autocorrelación parcial para el rezago s de una serie de tiempo y_t , denominada como ρ_s^* corresponde a la relación lineal entre y_t y y_{t-s} dejando $y_{t-1}, y_{t-2}, \dots, y_{t-(s-1)}$ constantes.

Es decir, esto es equivalente a correr una regresión múltiple en el que la desviación con respecto a la media del periodo actual es explicada por las s desviaciones con respecto a la media. La autocorrelación parcial corresponde solo para el rezago s , corresponderá únicamente al coeficiente de la desviación s .

Es decir, la autocorrelación parcial para el rezago s para la serie de tiempo y_t corresponde al coeficiente $\phi_s^{(s)}$ ($\phi_s^{(s)} = \rho_s^*$) de la siguiente combinación lineal:

$$\hat{y}_t - \mu = \phi_1^{(s)} (y_{t-1} - \mu) + \phi_2^{(s)} (y_{t-2} - \mu) + \dots + \phi_s^{(s)} (y_{t-s} - \mu) \quad (6.16)$$

donde

$$\begin{bmatrix} \phi_1^{(s)} \\ \phi_2^{(s)} \\ \vdots \\ \phi_s^{(s)} \end{bmatrix} = \begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{s-1} \\ \gamma_1 & \gamma_0 & \cdots & \gamma_{s-2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{s-1} & \gamma_{s-2} & \cdots & \gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_s \end{bmatrix} \quad (6.17)$$

En el Anexo 1 de este capítulo se explica el origen de esta expresión.

Por ejemplo, la primera autocorrelación parcial de y_t (ρ_1^*) corresponde al coeficiente $\phi_1^{(1)}$ en:

$$\hat{y}_t - \mu = \phi_1^{(1)} (y_{t-1} - \mu)$$

y la segunda autocorrelación parcial de y_t (ρ_2^*) corresponde al coeficiente $\phi_2^{(2)}$ en:

$$\hat{y}_t - \mu = \phi_1^{(2)} (y_{t-1} - \mu) + \phi_2^{(2)} (y_{t-2} - \mu)$$

Así, empleando 6.17 para el caso de un proceso $AR(1)$ para el rezago 1 la PACF (ρ_1^*) será:

$$\phi_1^{(1)} = \gamma_0^{-1} \gamma_1 = \frac{\gamma_1}{\gamma_0} = \rho_1$$

Para el caso del segundo rezago la PACF ρ_2^* corresponde al coeficiente $\phi_2^{(2)}$ de la siguiente operación:

$$\begin{bmatrix} \phi_1^{(2)} \\ \phi_2^{(2)} \end{bmatrix} = \begin{bmatrix} \gamma_0 & \gamma_1 \\ \gamma_1 & \gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}$$

En este caso¹⁰,

$$\phi_2^{(2)} = \frac{\gamma_0 (\phi_1 \gamma_0)}{\gamma_0^2 - (\phi_1 \gamma_0)^2} - \frac{(\phi_1 \gamma_0)^2}{\gamma_0^2 - (\phi_1 \gamma_0)^2} = \gamma_0^2 \left(\frac{1}{1 - \phi_1^2} - \frac{1}{1 - \phi_1^2} \right) = 0$$

¹⁰ En el Anexo de este capítulo se muestra como se llega a este resultado.

Siguiendo un procedimiento similar se puede derivar la siguiente PACF de los procesos AR(1):

$$\rho_s^* = \begin{cases} 1 & s = 0 \\ \rho_1 & s = 1 \\ 0 & o.w. \end{cases} \quad (6.18)$$

De esta manera, la PACF será diferente de cero para el primer rezago¹¹ y será igual a cero para los demás rezagos.

Para el caso del proceso AR(2), la PACF será diferente de cero para los dos primeros rezagos y cero para los demás rezagos. En general, para el caso de un proceso AR(p), tendremos que para los primeros p rezagos la PACF será diferente de cero y para los demás rezagos la PACF será cero.

Ejercicios

1. Encuentre la media, varianza y covarianza de un proceso AR(1). Muestre todos los pasos.
2. Demuestre que $\rho_{-s} = \rho_s$.
3. Encuentre la PACF para el proceso AR(2). Evalúe y grafique la función PACF que encuentre para los siguientes casos:
 - a. $(\phi_1, \phi_2) = (0,5, -0,5)$
 - b. $(\phi_1, \phi_2) = (0,8, -0,8)$
 - c. $(\phi_1, \phi_2) = (0,8, 0,2)$

Anexo 1. Derivación de la expresión 6.17

Partiendo de 6.16:

$$\hat{y}_t - \mu = \phi_1^{(s)} (y_{t-1} - \mu) + \phi_2^{(s)} (y_{t-2} - \mu) + \dots + \phi_s^{(s)} (y_{t-s} - \mu)$$

Esto lo podemos expresar matricialmente de la siguiente manera:

$$\begin{bmatrix} \hat{y}_{1+s} - \mu \\ \hat{y}_{2+s} - \mu \\ \vdots \\ \hat{y}_T - \mu \end{bmatrix} = \begin{bmatrix} y_{1+s} - \mu & y_{2+s} - \mu & \cdots & y_1 - \mu \\ y_{2+s} - \mu & y_{3+s} - \mu & \cdots & y_2 - \mu \\ \vdots & \vdots & \ddots & \vdots \\ y_T - \mu & y_{T-1} - \mu & \cdots & y_{T-s} - \mu \end{bmatrix} \begin{bmatrix} \phi_1^{(s)} \\ \phi_2^{(s)} \\ \vdots \\ \phi_s^{(s)} \end{bmatrix}$$

De hecho esto es de la forma $W = X\phi$, lo cual implica que los coeficientes de la proyección de y en sus s valores más recientes es tan dados por $\phi = (X^T X)^{-1} X^T W$,

¹¹ La PACF para el rezago cero es uno por definición.

donde

$$\phi = \begin{bmatrix} \phi_1^{(s)} \\ \phi_2^{(s)} \\ \vdots \\ \phi_s^{(s)} \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} \sum (\hat{y}_t - \mu)(y_t - \mu) & \sum (\hat{y}_t - \mu)(y_{t-1} - \mu) & \cdots & \sum (\hat{y}_t - \mu)(y_{t-(s-1)} - \mu) \\ \sum (y_{t-1} - \mu)(\hat{y}_t - \mu) & \sum (\hat{y}_t - \mu)(y_t - \mu) & \cdots & \sum (\hat{y}_t - \mu)(y_{t-(s-2)} - \mu) \\ \vdots & \vdots & \ddots & \vdots \\ \sum (y_{t-(s-1)} - \mu)(\hat{y}_t - \mu) & \sum (y_{t-(s-2)} - \mu)(\hat{y}_t - \mu) & \cdots & \sum (\hat{y}_t - \mu)(y_t - \mu) \end{bmatrix}$$

$$X^T W = \begin{bmatrix} \sum (\hat{y}_t - \mu)(y_{t-1} - \mu) \\ \sum (\hat{y}_t - \mu)(y_{t-2} - \mu) \\ \vdots \\ \sum (\hat{y}_t - \mu)(y_{t-s} - \mu) \end{bmatrix}$$

Si, todas las T observaciones son aleatorias y \hat{y}_T es un estimador insesgado de y_t , entonces $\sum (\hat{y}_t - \mu)(y_{t-s} - \mu)$ corresponde a la auto covarianza entre y_t y y_{t-s} (γ_s). Esto es equivalente a:

$$\begin{bmatrix} \phi_1^{(s)} \\ \phi_2^{(s)} \\ \vdots \\ \phi_s^{(s)} \end{bmatrix} = \begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{s-1} \\ \gamma_1 & \gamma_0 & \cdots & \gamma_{s-2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{s-1} & \gamma_{s-2} & \cdots & \gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_s \end{bmatrix}$$

Anexo 2. Cálculo de la segunda autocorrelación parcial del proceso AR(2)

Empleando 6.17 para el caso de un proceso $AR(1)$ para el rezago 2 la PACF (ρ_2^*) corresponde al coeficiente $\phi_2^{(2)}$ de la siguiente operación:

$$\begin{bmatrix} \phi_1^{(2)} \\ \phi_2^{(2)} \end{bmatrix} = \begin{bmatrix} \gamma_0 & \gamma_1 \\ \gamma_1 & \gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}$$

La primera tarea es encontrar la inversa de esta expresión. Así,

$$\begin{bmatrix} \gamma_0 & \gamma_1 \\ \gamma_1 & \gamma_0 \end{bmatrix}^{-1} = \frac{1}{\begin{vmatrix} \gamma_0 & \gamma_1 \\ \gamma_1 & \gamma_0 \end{vmatrix}} \begin{bmatrix} \gamma_0 & -\gamma_1 \\ -\gamma_1 & \gamma_0 \end{bmatrix} = \frac{1}{\gamma_0^2 - \gamma_1^2} \begin{bmatrix} \gamma_0 & -\gamma_1 \\ -\gamma_1 & \gamma_0 \end{bmatrix}$$

$$\begin{bmatrix} \gamma_0 & \gamma_1 \\ \gamma_1 & \gamma_0 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{\gamma_0}{\gamma_0^2 - \gamma_1^2} & -\frac{\gamma_1}{\gamma_0^2 - \gamma_1^2} \\ -\frac{\gamma_1}{\gamma_0^2 - \gamma_1^2} & \frac{\gamma_0}{\gamma_0^2 - \gamma_1^2} \end{bmatrix}$$

Reemplazando en la expresión original, se obtiene:

$$\begin{bmatrix} \phi_1^{(2)} \\ \phi_2^{(2)} \end{bmatrix} = \begin{bmatrix} \frac{\gamma_0}{\gamma_0^2 - \gamma_1^2} & -\frac{\gamma_1}{\gamma_0^2 - \gamma_1^2} \\ -\frac{\gamma_1}{\gamma_0^2 - \gamma_1^2} & \frac{\gamma_0}{\gamma_0^2 - \gamma_1^2} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}$$

Empleando 6.5, se obtiene

$$\phi_2^{(2)} = \frac{\gamma_0 (\phi_1 \gamma_0)}{\gamma_0^2 - (\phi_1 \gamma_0)^2} - \frac{(\phi_1 \gamma_0)^2}{\gamma_0^2 - (\phi_1 \gamma_0)^2} = \gamma_0^2 \left(\frac{1}{1 - \phi_1^2} - \frac{1}{1 - \phi_1^2} \right) = 0$$

Capítulo 7

Modelos de Media Móvil (MA)

En el capítulo anterior se discutieron las características de la familia de procesos AR(p) y las características que estos deben cumplir para que el proceso sea estacionario. También se discutieron las características de la función de autocorrelación (ACF) y autocorrelación parcial (PACF) para dicha familia. En este capítulo nos concentraremos en otra familia de procesos conocida como de media móvil (Moving Average en inglés) o también conocidos como MA por su sigla.

Al igual que en el caso de la familia AR, la familia MA tiene como característica principal suponer que la historia de la variable aleatoria afecta su comportamiento futuro y presente de la serie. En general una secuencia de variables aleatorias

$$\{y_t | t \in 1, 2, \dots, T\}$$

se definirá como un proceso de media móvil de orden q ($MA(q)$) si el valor de la secuencia hoy (y_t) corresponde a la suma de i) un promedio ponderado de los últimos q periodos pasados de un proceso ruido blanco, ii) un término de error y iii) en algunas ocasiones una constante. En otras palabras,

$$y_t = \delta + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Es decir,

$$y_t = \delta + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t$$

A continuación estudiaremos las características y propiedades de los procesos MA, empezando por los más sencillos. Primero prestaremos atención a los procesos MA(1) y MA(2), para luego estudiar el caso más general MA(p). Finalmente, estudiaremos la relación que existe entre los procesos MA y AR.

7.1. Proceso $MA(1)$

En general, los proceso $MA(1)$ se caracterizan porque el valor actual de serie $\{y_t\}$ es función del valor inmediatamente pasado del término de error del período anterior, del término de error actual y de una constante. En otras palabras,

$$y_t = \delta + \theta_1 \varepsilon_{t-1} + \varepsilon_t \quad (7.1)$$

Es relativamente facil encontrar la media y la autocovarianza del proceso $MA(1)$:

$$E[y_t] = \delta + \theta_1 E[\varepsilon_{t-1}] + E[\varepsilon_t] = \delta = \mu \quad (7.2)$$

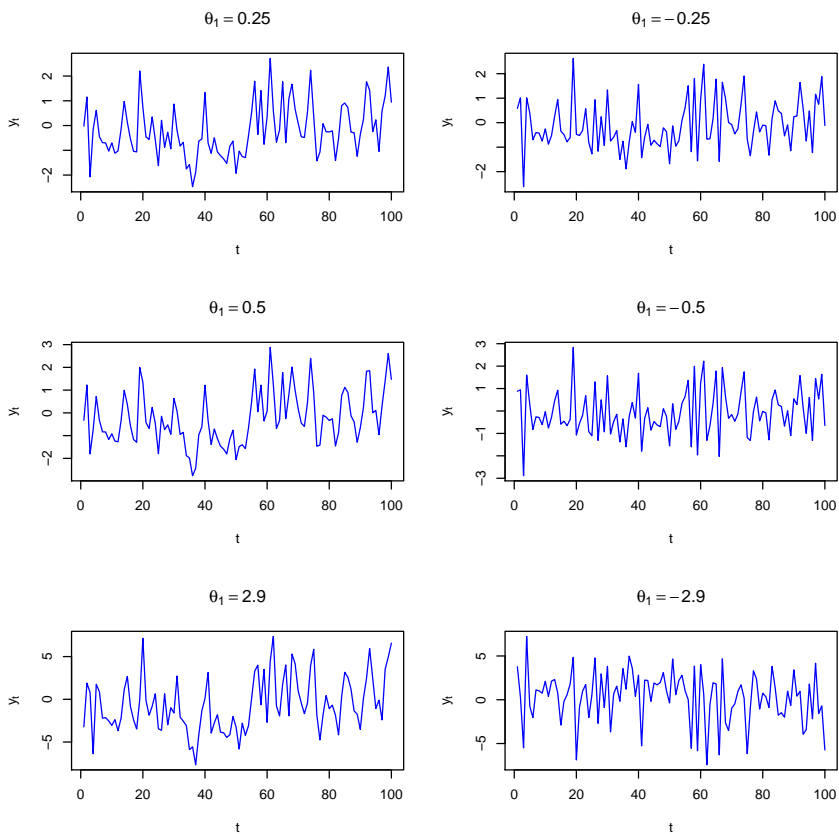
$$\gamma_s = \begin{cases} \sigma_\varepsilon^2 (1 + \theta_1^2) & s = 0 \\ \theta_1 \sigma_\varepsilon^2 & s = 1, -1 \\ 0 & o.w. \end{cases} \quad (7.3)$$

De estas expresiones, es claro que el proceso $MA(1)$ tiene una media y varianza constante γ_0 y la autocovarianza no depende del tiempo. Es decir, el proceso $MA(1)$ es estacionario. Es importante anotar que no fue necesario suponer nada sobre la naturaleza de θ_1 para que que el proceso $MA(1)$ sea estacionario. En otras palabras, todo proceso $MA(1)$ será estacionario.

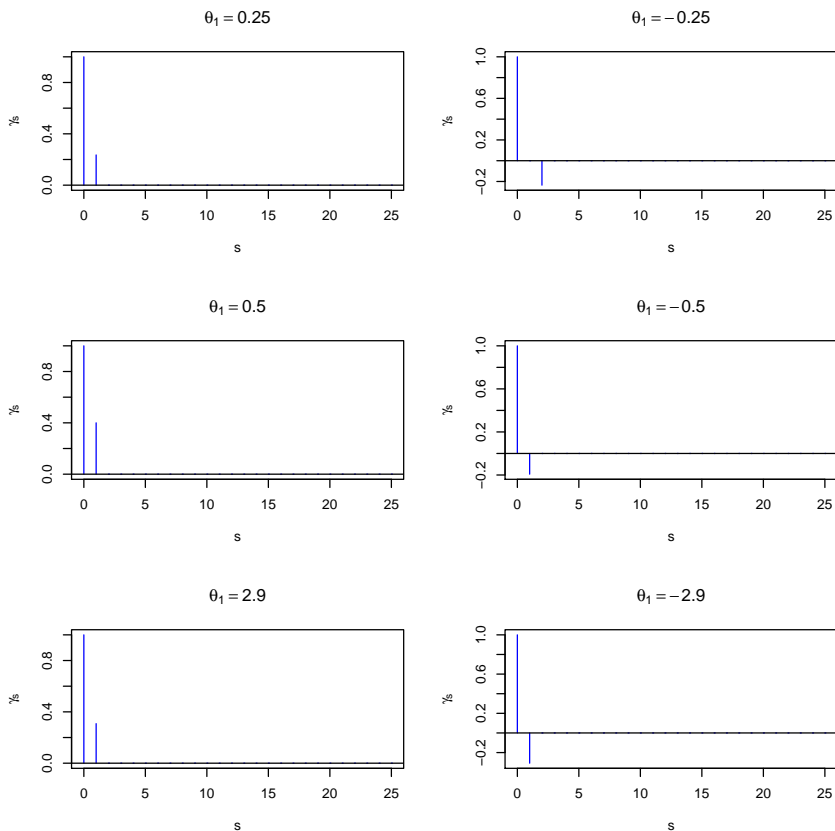
Otra característica interesante de los procesos $MA(1)$ que se desprende de 7.3 es que su función de autocorrelación (ACF) será diferente de cero hasta el primer rezago y cero para rezagos superiores o iguales a dos. Es fácil mostrar que:

$$\rho_s = \begin{cases} 1 & s = 0 \\ \frac{\theta_1}{(1 + \theta_1^2)} & s = 1, -1 \\ 0 & o.w. \end{cases} \quad (7.4)$$

Figura 7.1 Procesos $MA(1)$ simulados para diferentes valores θ_1 y las mismas realizaciones del proceso ruido blanco (ε_t).



En la figura 7.1 se presentan 100 realizaciones de seis diferentes procesos $MA(1)$, cada uno emplea un θ_1 diferente, pero las mismas realizaciones de ε_t . Al igual que en el caso de un proceso $AR(1)$, la característica predominante en estos procesos es que no importa la magnitud de la perturbación (ε_t) el proceso tiende regresar a su media, esta característica se conoce como reversión a la media (en inglés mean reverting).

Figura 7.2 AFC teórica para procesos MA(1) con diferentes valores θ_1 .

7.2. Proceso MA(2)

Un proceso $MA(2)$ corresponde a una serie de tiempo tal que su DGP corresponde a:

$$y_t = \delta + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \varepsilon_t \quad (7.5)$$

En este caso, es fácil mostrar que:

$$E[y_t] = \delta = \mu \quad (7.6)$$

$$\gamma_s = \begin{cases} \sigma_\varepsilon^2 (1 + \theta_1^2 + \theta_2^2) & s = 0 \\ \sigma_\varepsilon^2 (\theta_1 + \theta_2 \theta_1) & s = 1, -1 \\ \sigma_\varepsilon^2 (\theta_2) & s = 2, -2 \\ 0 & o.w. \end{cases} \quad (7.7)$$

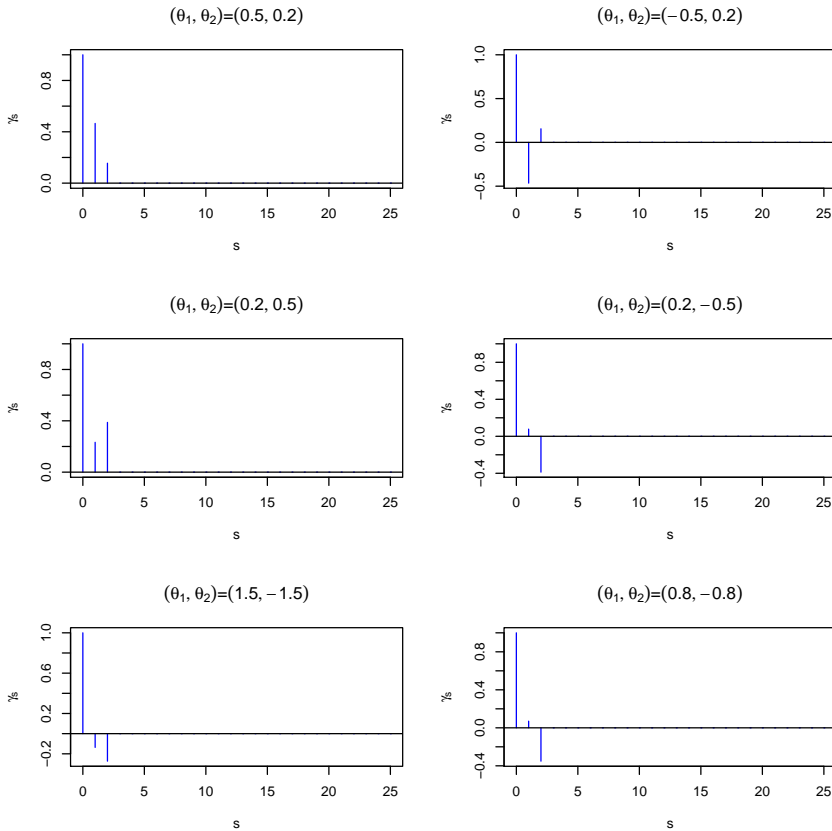
Es decir, la media y la varianza son constantes y la covarianza no depende del tiempo. En otras palabras, el proceso $MA(2)$ será estacionario. Este resultado, no depende de los valores de los parámetros de media móvil (θ_1 o θ_2). En otras palabras, todo proceso $MA(2)$ será estacionario.

Empleando 7.7 podemos derivar la ACF de un proceso $MA(2)$, es fácil demostrar que:

$$\rho_s = \begin{cases} 1 & sis = 0 \\ \frac{(\theta_1 + \theta_2 \theta_1)}{(1 + \theta_1^2 + \theta_2^2)} & sis = 1, -1 \\ \frac{\theta_2}{(1 + \theta_1^2 + \theta_2^2)} & s = 2, -2 \\ 0 & o.w. \end{cases} \quad (7.8)$$

En la figura 7.3 se presentan las ACF teóricas para 6 diferentes combinaciones de valores de θ_1 y θ_2 . Como se puede observar, el comportamiento del ACF depende de los valores de estos dos parámetros; pero en todo caso, la ACF será cero a partir del segundo rezago..

Figura 7.3 AFC teórica para procesos $MA(2)$ con diferentes valores θ_1 y θ_2 .



7.3. Proceso $MA(q)$

En las dos secciones anteriores encontramos que todo proceso $MA(1)$ y $MA(2)$ serán estacionarios. Adicionalmente encontramos una característica típica en la ACF de ambos procesos, la ACF es igual a cero para los rezagos mayores que el correspondiente orden del respectivo proceso. En esta sección generalizaremos nuestros resultados. Empecemos por definir un proceso $MA(q)$ como:

$$y_t = \delta + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (7.9)$$

Para este proceso tenemos que:

$$E[y_t] = \mu = \delta \quad (7.10)$$

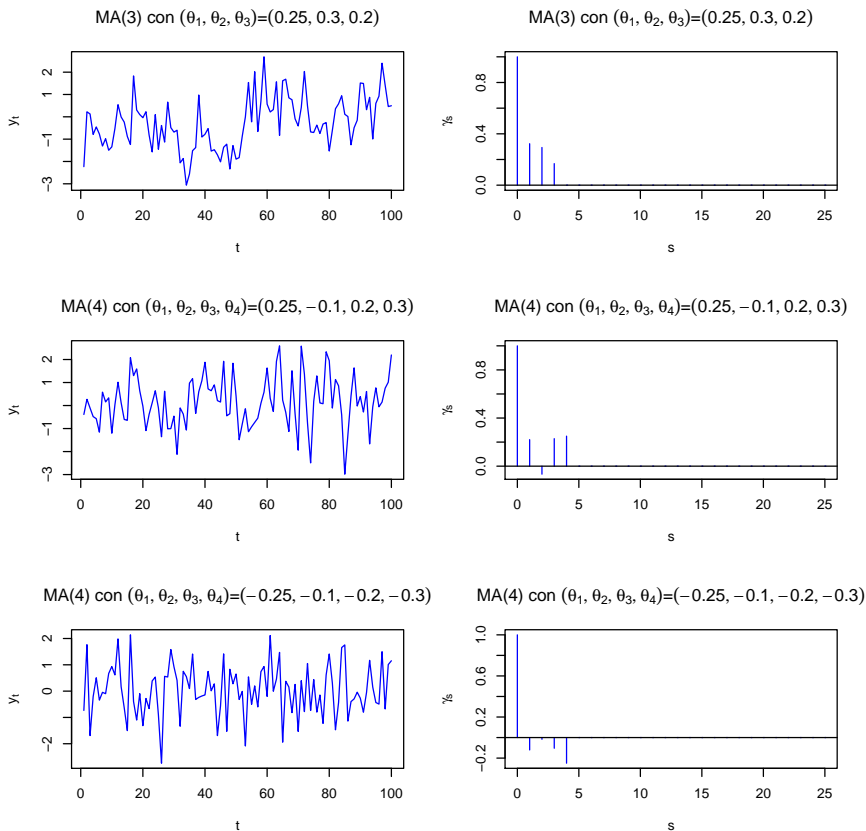
$$\gamma_s = \begin{cases} \sigma_\varepsilon^2 \left(\sum_{i=0}^{q-s} \theta_i \theta_{i+s} \right) & |s| \leq q \\ 0 & o.w. \end{cases} \quad (7.11)$$

Donde $\theta_0 = 1$. Así, no será necesario suponer algo sobre los coeficientes del proceso $MA(q)$ para obtener una media y varianza constante y que la covarianza no dependa del tiempo. Así, todo proceso $MA(q)$ será estacionario.

Otro resultado interesante es el comportamiento de la ACF del proceso $MA(q)$ es que ésta será igual a cero para todo rezago mayor a q . Este resultado se puede demostrar rápidamente partiendo de 7.11, de esta expresión podemos encontrar que:

$$\rho_s = \begin{cases} \frac{\left(\sum_{i=0}^{q-k} \theta_i \theta_{i+s} \right)}{\left(\sum_{i=0}^{q-k} \theta_i^2 \right)} & |s| \leq q \\ 0 & o.w. \end{cases} \quad (7.12)$$

En la figura 7.4 se presentan tres muestras simuladas a partir de procesos MA y sus correspondientes ACF teóricas. Se presenta un proceso $MA(3)$ y dos $MA(4)$. A diferencia de lo que observamos en los procesos $AR(p)$, al observar la ACF de las realizaciones de un proceso $MA(q)$ podríamos determinar el orden q .

Figura 7.4 Series simuladas y AFC teórica para diferentes procesos MA(q).

7.4. Relación entre los procesos MA y AR

Antes de continuar nuestro análisis de los procesos MA es importante observar que existe una relación entre los procesos MA y AR. Por ejemplo, consideremos el proceso $MA(1)$ y por simplicidad supongamos que $\delta = 0$:

$$y_t = \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

Ahora note que ε_t puede ser expresado como:

$$\varepsilon_t = y_t - \theta_1 \varepsilon_{t-1}$$

Remplazando ε_{t-1} en la expresión del MA(1) tenemos que:

$$y_t = \theta_1(y_{t-1} - \theta_1 \varepsilon_{t-2}) + \varepsilon_t$$

$$y_t = \theta_1 y_{t-1} - \theta_1^2 \varepsilon_{t-2} + \varepsilon_t$$

Ahora podemos remplazar ε_{t-2} , para obtener

$$y_t = \theta_1 y_{t-1} - \theta_1^2 y_{t-2} + \theta_1^3 \varepsilon_{t-3} + \varepsilon_t$$

Podemos seguir remplazando de manera recursiva hasta obtener

$$y_t = \theta_1 y_{t-1} - \theta_1^2 y_{t-2} - \theta_1^3 y_{t-3} + \cdots + \varepsilon_t$$

Lo cual implica

$$y_t = \sum_{i=1}^{\infty} \theta_1^i y_{t-i} + \varepsilon_t$$

Esta expresión representa un proceso AR de orden infinito ($AR(\infty)$). Esto quiere decir que si se cumple que $|\theta_1| < 1$, entonces un proceso $MA(1)$ se puede expresar como un $AR(\infty)$ estacionario. Esta cualidad se conoce como la propiedad de invertibilidad.

Ahora este resultado nos lleva a otro resultado interesante. Dado que un proceso $AR(p)$ tiene una PACF que es diferente de cero hasta el rezago cero y para los demás rezagos es cero, tenemos que para el caso de un $MA(1)$ (que es equivalente a un $AR(\infty)$) tendrá una PACF diferente de cero para todos los rezagos.

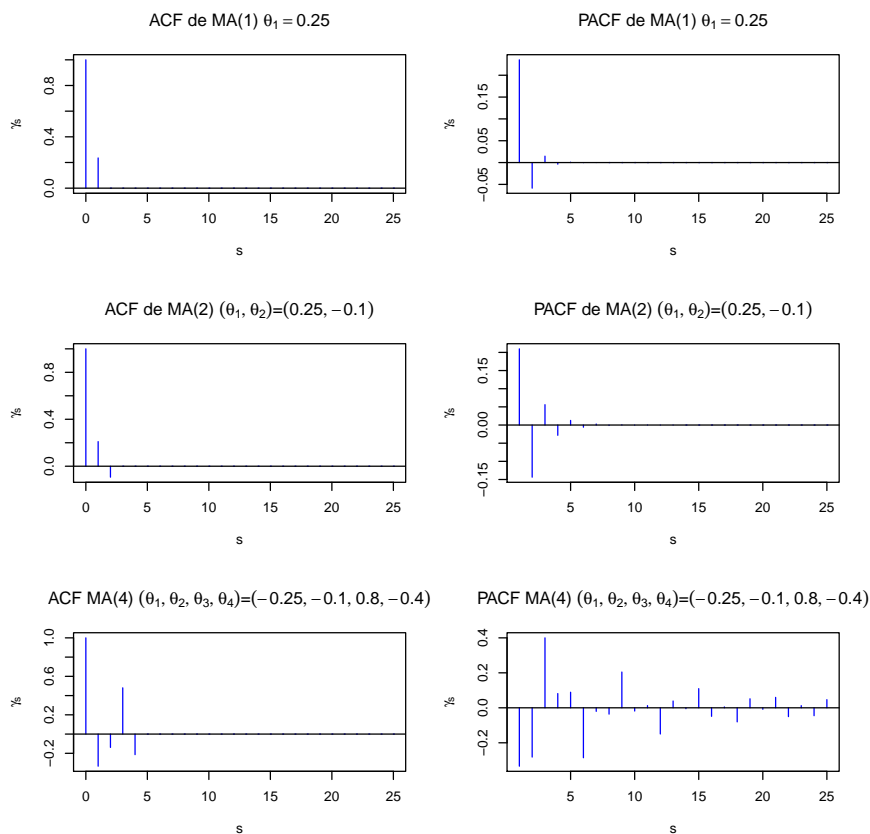
De hecho, en general tenemos que cualquier proceso $MA(q)$ dado por:

$$y_t = \delta + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t$$

podrá expresarse como un proceso $AR(\infty)$ si las raíces del polinomio característico¹ y así sucesivamente. $[1 + \theta_1 z + \theta_2 z^2 + \cdots + \theta_q z^q]$ están por fuera del círculo unitario. En otras palabras, si las raíces del polinomio característico del proceso $MA(q)$ están por fuera del círculo unitario, entonces el proceso $MA(q)$ es invertible.

Regresando a las implicaciones de que un proceso MA se puede escribir como un proceso AR de orden infinito, tendremos que la PACF de cualquier proceso $MA(q)$ tenderá a cero. En la figura 7.5 se presenta la ACF y PACF para diferentes proceso MA .

¹ En este caso el polinomio característico se construye remplazando ε_t por 1, ε_{t-1} por z , ε_{t-2} por z^2

Figura 7.5 ACF y PAFC teórica para diferentes procesos MA(q).

Ejercicios

1. Encuentre la media, varianza y covarianza de un proceso $MA(1)$ y $MA(2)$. Muestre todos los pasos.
2. Encuentre la ACF para el proceso $MA(3)$.

Capítulo 8

Modelos de ARMA y ARIMA

En los dos capítulos anteriores se discutieron las características de la familia de procesos AR(p) y MA(q) y las características que estos deben cumplir para que el proceso sea estacionario. También se discutieron las características de la función de autocorrelación (ACF) y autocorrelación parcial (PACF) para dichas familias. En este capítulo nos concentraremos en otra familia derivada de estas dos: la familia ARMA. Finalmente discutiremos los modelos ARIMA.

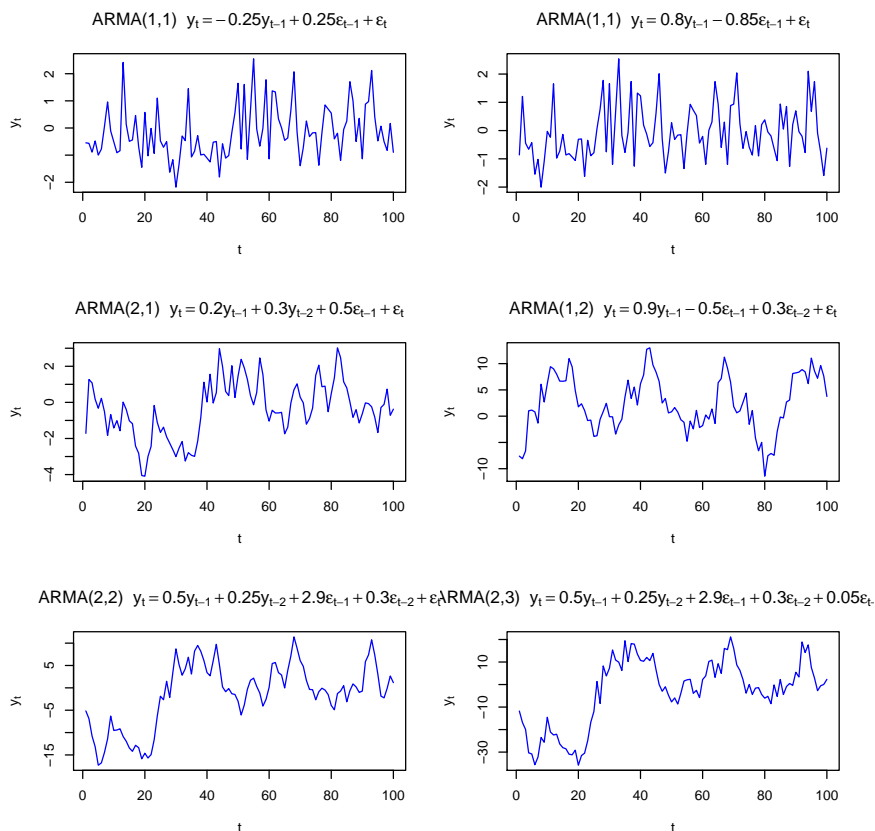
En general una secuencia de variables aleatorias

$$\{y_t | t \in 1, 2, \dots, T\}$$

se definirá como un proceso de $ARMA(p, q)$ ($\{y_t | t \in 1, 2, \dots, T\} \sim ARMA(p, q)$) si el valor de la secuencia hoy (y_t) corresponde a la suma de i) un promedio ponderado de los últimos p periodos, ii) un promedio ponderado de los últimos q periodos pasados de un proceso ruido blanco, iii) un término de error y iv) en algunas ocasiones una constante. En otras palabras,

$$y_t = \delta + \sum_{i=1}^p \phi_i y_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (8.1)$$

Estos procesos ARMA son una combinación de procesos AR y MA y por tanto las características de los procesos ARMA serán una combinación de sus correspondientes partes AR y MA. En la figura 8.1 se presentan realizaciones de diferentes modelos ARMA.

Figura 8.1 Procesos ARMA(p,q) simulados.

Antes de continuar será importante recordar las características más importantes de los procesos $AR(p)$ y $MA(q)$. Recordemos que los procesos $AR(p)$ serán estacionarios si las raíces del polinomio característico están por fuera del círculo unitario; por otro lado, todos los procesos $MA(q)$ son estacionarios. En cuanto a la ACF, los procesos $AR(p)$ tienen una ACF que tiende a cero de forma exponencial u oscilante. En el caso de los procesos $MA(q)$ la ACF es diferente de cero hasta el rezago q y de ahí en adelante es igual a cero. Finalmente, en el caso de la PACF, tenemos un comportamiento totalmente contrario al de la ACF. Es decir, en el caso de los $AR(p)$ la PACF es únicamente diferente para rezagos iguales o menores a p ; para el caso de los $MA(q)$ la PACF tiende a cero de manera exponencial u oscilante.

Finalmente, todo proceso $AR(p)$ se puede invertir (expresar como un $MA(\infty)$). En el caso de los procesos $MA(q)$, solo se puede invertir (convertir en un proceso $AR(\infty)$) si las raíces del polinomio característico están por fuera del círculo unitario.

Cuadro 8.1 Características de los procesos $AR(p)$ y $MA(q)$.

	$AR(p)$	$MA(q)$
	$\phi_p(L)y_t = \varepsilon_t$	$y_t = \theta_q(L)\varepsilon_t$
Estacionariedad	Todas las raíces de $\phi_p(z) = 0$ están por fuera del círculo unitario	No existe restricción
ACF	$\rho_s \rightarrow 0$ exponencialmente u oscilante	$\rho_s \neq 0$ para $s \leq q$ $\rho_s = 0$ para $s > q$
PACF	$\rho_s^* \neq 0$ para $s \leq p$ $\rho_s^* = 0$ para $s > p$	$\rho_s^* \rightarrow 0$ exponencialmente u oscilante
Invertibilidad	No existe restricción para procesos $AR(p)$ estacionarios	Todas las raíces de $\theta_q(z) = 0$ están por fuera del círculo unitario

8.1. Características de los procesos $ARMA(p, q)$

En general las características de los proceso $ARMA(p, q)$ dependerán de las característica de sus respectivos componentes $AR(p)$ y $MA(q)$. Así, para tener un proceso $ARMA(p, q)$ estacionario necesitamos que tanto el proceso $AR(p)$ como el $MA(q)$ sean estacionarios.

En general, el proceso $ARMA(p, q)$:

$$y_t = \delta + \sum_{i=1}^p \phi_i y_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (8.2)$$

será estacionario si las raíces del polinomio característico de orden p de la parte AR están por fuera del círculo unitario.

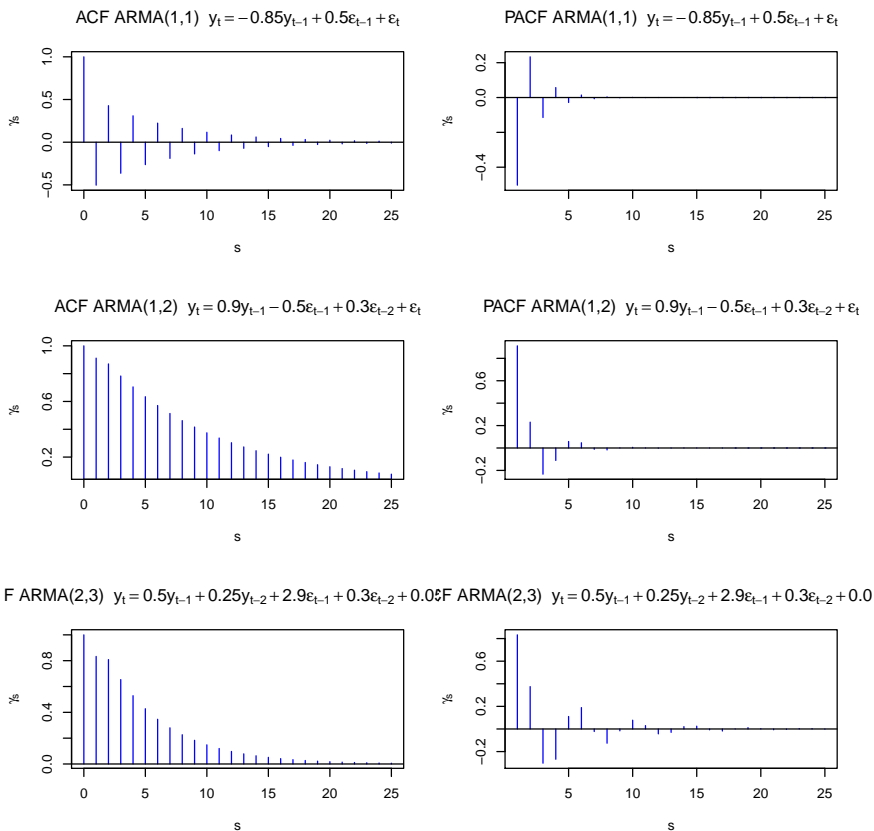
Por otro lado, las características de la ACF y PACF de los modelos $ARMA(p, q)$ corresponden a la suma de las característica de las ACF y PACF de sus respectivos componentes $AR(p)$ y $MA(q)$. Así, en general la ACF de un proceso $ARMA(p, q)$ será claramente diferente de cero para los primeros q rezagos y de ahí en adelante tenderá a cero. Para la PACF, tendremos que ésta será claramente diferente de cero para los primeros p rezagos y de ahí en adelante tenderá a cero.

Finalmente, los procesos $ARMA(p, q)$ pueden ser expresados como un proceso $AR(\infty)$ o $MA(\infty)$. Para expresar un $ARMA(p, q)$ como un proceso $MA(\infty)$ se necesita que las raíces del polinomio característico de orden p de la parte autoregresiva estén por fuera del círculo unitario. Es decir, si el proceso $ARMA(p, q)$ es estacionario. Así, todo proceso estacionario $ARMA(p, q)$ puede ser expresado como un $MA(\infty)$.

Si se desea expresar un $ARMA(p, q)$ como un proceso $AR(\infty)$ se necesita que las raíces del polinomio de orden q de la parte MA estén por fuera del círculo unitario. Así, todo proceso estacionario $ARMA(p, q)$ puede ser expresado como un $AR(\infty)$ si las raíces del polinomio característico de la parte $MA(q)$ tiene sus

raíces por fuera del círculo unitario. Estos resultados se resumen en el siguiente cuadro; así mismo en la figura 8.2.

Cuadro 8.2 Características de los procesos $ARMA(p, q)$.	
$ARMA(p, q)$	
$\phi_p(L)y_t = \theta_q(L)\varepsilon_t$	
Estacionariedad	Todas las raíces de $\phi_p(z) = 0$ están por fuera del círculo unitario.
ACF	$\rho_s \neq 0$ para $s \leq q$ $\rho_s \rightarrow 0$ para $s > q$
PACF	$\rho_s^* \neq 0$ para $s \leq p$ $\rho_s^* \rightarrow 0$ para $s > p$
Invertibilidad	No existe restricción para convertirse en $MA(\infty)$ Todas las raíces de $\theta_q(z) = 0$ están por fuera del círculo unitario para convertirse en $AR(\infty)$.

Figura 8.2 ACF y PACF teóricas para diferentes procesos ARMA.

ARMAPACFacf

8.2. Modelos ARIMA

Hasta el momento hemos discutido series de tiempo que son estacionarias, pero en la práctica es común encontrar series que no son estacionarias. Por ejemplo, es común encontrar series que crecen en el tiempo y que por tanto no pueden tener una media que no cambia en el tiempo (ver los ejemplos del Capítulo 2 en la Figura 2.1).

En esos casos es posible que diferenciando¹ la serie encontremos una serie estacionaria.

Supongamos que tenemos una secuencia de variables aleatorias

$$\{y_t | t \in 1, 2, \dots, T\}$$

¹ En el contexto de las series de tiempo, diferenciar una serie o sacar las primeras diferencias de una serie es equivalente a integrar la serie.

que no es estacionaria. Entonces, **sí diferenciamos una vez la serie y obtenemos una serie estacionaria, entonces diremos que la serie original es integrada de orden uno.** Es decir, si y_t no es estacionario pero $\Delta y_t = y_t - y_{t-1}$ si lo es, entonces diremos que y_t es integrada de orden uno ($y_t \sim I(1)$).

En general, si la serie y_t debe ser diferenciada d veces para obtener una serie estacionaria, entonces diremos que la serie y_t es integrada de orden d ($y_t \sim I(d)$).

Ahora, bien diremos que una serie y_t proviene de un modelo $ARIMA(p, d, q)$ si el correspondiente DGP es:

$$\Delta^d y_t = \delta + \phi_1 \Delta^d y_{t-1} + \phi_2 \Delta^d y_{t-2} + \dots + \phi_p \Delta^d y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (8.3)$$

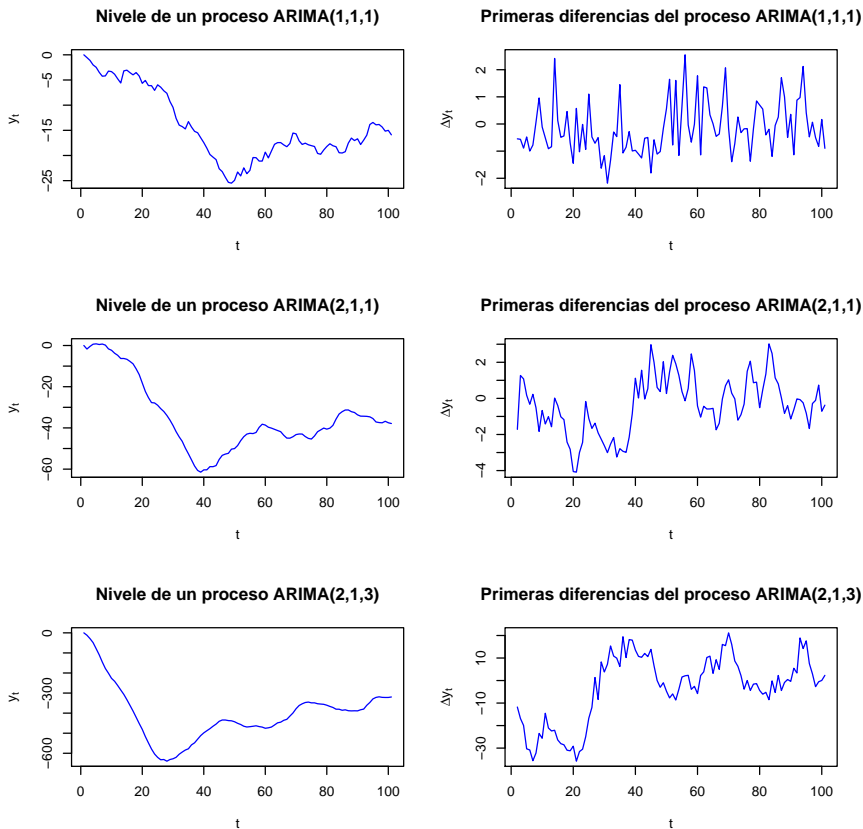
Una forma equivalente de escribir un proceso $ARIMA(p, d, q)$ es:

$$\Delta^d y_t = \delta + \sum_{i=1}^p \phi_i \Delta^d y_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (8.4)$$

Así, **un proceso $ARIMA(p, d, q)$ es una serie que después de ser diferenciada d veces sigue un proceso $ARMA(p, q)$.** En la siguiente figura se presentan varios ejemplos de procesos $ARIMA(p, 1, q)$ y de las primeras diferencias del proceso. Como se puede observar como la media de las series² aparentemente cambian en el tiempo y como las primeras diferencias³ ya tienen un comportamiento más acorde a una serie estacionaria.

² La serie sin diferenciar se conoce comúnmente como el nivel de la serie o la serie en niveles.

³ Es decir, la serie diferenciada Δy_t .

Figura 8.3 Procesos ARIMA(p,d,q) simulados.

8.3. Raíces del polinomio característico del componente AR y el orden de integración

En la sección anterior discutimos la definición de un proceso integrado de orden d ($I(d)$). En esta sección discutiremos la relación entre las raíces del polinomio característico del componente AR y el orden de integración de una serie. Para entender este concepto, empleemos un ejemplo.

Consideremos el siguiente proceso:

$$y_t = 0,5 + 1,2y_{t-1} - 0,2y_{t-2} + \varepsilon_t$$

Claramente este proceso no es estacionario, pues el coeficiente que acompaña el primer término AR es mayor que uno. En este caso el polinomio característico:

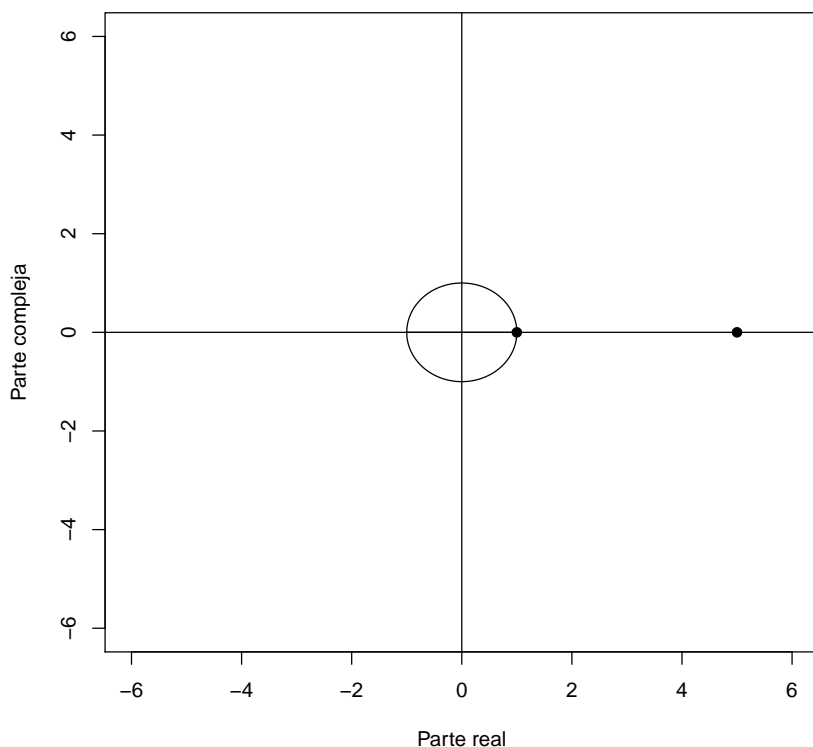
$$\left[1 - \frac{6}{5}z + \frac{1}{5}z^2\right] = 0$$

Factorizando, se tiene que:

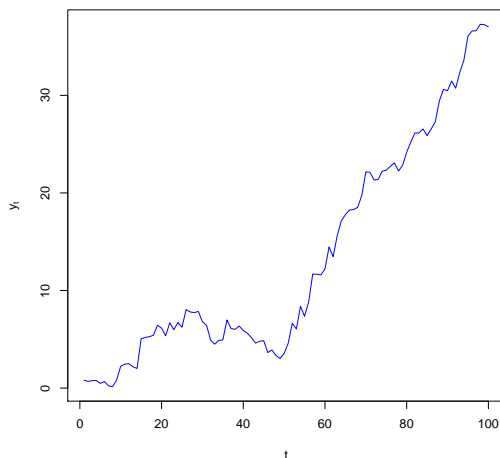
$$(1 - z) \left(1 - \frac{1}{5} \cdot z\right) = 0$$

Esto implica que $1 - z = 0$ o $1 - \frac{1}{5} \cdot z = 0$. Es decir, $z = 1$ o $z = 5$. En la figura 8.4 se representan las raíces de este proceso.

Figura 8.4 Raíces del proceso $y_t = 0,5 + 1,2y_{t-1} - 0,2y_{t-2} + \varepsilon_t$.



En este caso una de las raíces del proceso está sobre el círculo unitario, esto es lo que se conoce como una raíz unitaria. Es decir, el proceso tiene una raíz unitaria. En la figura 8.5 se presenta una serie simulada de este proceso. Claramente esta serie no es estacionaria pues crece en el tiempo.

Figura 8.5 Serie simulada del procesos $y_t = 0,5 + 1,2y_{t-1} - 0,2y_{t-2} + \varepsilon_t$.

Por otro lado, el número de raíces unitarias indica el orden de integración de la serie. Es decir, el número de raíces unitarias es igual al número de veces que se tiene que diferenciar la serie. En este caso, dado que existe una raíz unitaria, la serie es $I(1)$. Noten lo siguiente, si diferenciamos el proceso tendremos un proceso estacionario.

Para entender por qué este es el caso, restemos a ambos lados y_{t-1} :

$$y_t - y_{t-1} = 0,5 + 1,2y_{t-1} - 0,2y_{t-2} + \varepsilon_t - y_{t-1}$$

$$y_t - y_{t-1} = 0,5 + 0,2y_{t-1} - 0,2y_{t-2} + \varepsilon_t$$

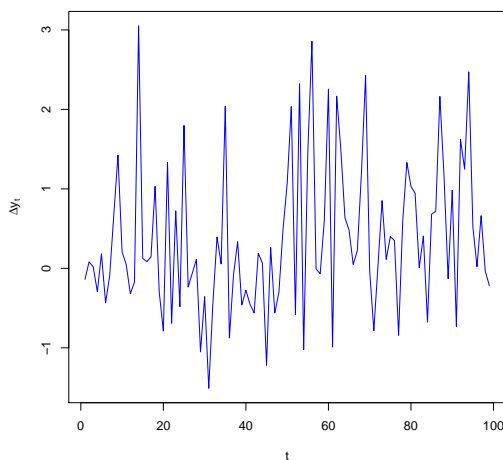
$$y_t - y_{t-1} = 0,5 + 0,2(y_{t-1} - y_{t-2}) + \varepsilon_t$$

Como $\Delta y_t = y_t - y_{t-1}$, entonces la anterior expresión se puede re-expresar como

$$\Delta y_t = 0,5 + 0,2\Delta y_{t-1} + \varepsilon_t$$

Noten que esta serie ya es estacionaria. En otras palabras, Δy_t es un proceso AR estacionario, pues el coeficiente que acompaña la parte autoregresiva (Δy_{t-1}) es menor que uno. En la figura 8.6 se puede observar las primeras diferencias de la serie. Estas ya se comportan como una serie estacionaria.

Figura 8.6 Primeras diferencias (Δy_t) de la serie simulada del procesos $y_t = 0,5 + 1,2y_{t-1} - 0,2y_{t-2} + \varepsilon_t$.



En general, **sí una serie cuenta con d de raíces unitarias en su polinomio característico de la parte AR y las otras raíces están por fuera del círculo unitario, entonces la serie será integrada de orden d ($I(d)$).** En otras palabras, se debe diferenciar d veces para obtener un proceso estacionario.

8.4. Comentarios finales

En el capítulo 9 discutiremos como determinar, a partir de una muestra cuál es el DGP más adecuado y como estimarlo. Así mismo, en el capítulo ?? discutiremos sobre las pruebas estadísticas disponibles para detectar si una serie tiene una raíz unitaria o no.

Ejercicios

1. Encuentre la media y varianza de un proceso $ARIMA(0, 1, 1)$ y $ARIMA(1, 1, 1)$. Encuentre la covarianza de las primeras diferencias de este proceso.

Capítulo 9

Estimación de modelos ARMA y ARIMA y construcción de pronósticos

En los tres capítulo anteriores se discutieron las características de la familia de procesos AR(p), MA(q), ARMA(p,q) y ARIMA. En este capítulo nos concentramos en el problema empírico de estimar un modelo de ARMA o ARIMA a partir de una muestra.

Habíamos discutido que un modelo AR(p), Ma(q) o ARMA (p,q) puede ser identificado por las características de sus respectivas funciones (poblacionales) de autocorrelación y autocorrelación parcial. En la Tabla 9.1 se resumen dichas características.

Cuadro 9.1 Características de las ACF y PACF poblacionales de los $AR(p)$, $MA(q)$ y $ARMA(p,q)$.

	$AR(p)$	$MA(q)$	$ARMA(p,q)$
	$\phi_p(L)y_t = \varepsilon_t$	$y_t = \theta_q(L)\varepsilon_t$	$\phi_p(L)y_t = \theta_q(L)\varepsilon_t$
ACF	$\rho_s \rightarrow 0$	$\rho_s \neq 0$ para $s \leq q$ $\rho_s = 0$ para $s > q$	$\rho_s \neq 0$ para $s \leq q$ $\rho_s \rightarrow 0$ para $s > q$
PACF	$\rho_s^* \neq 0$ para $s \leq p$ $\rho_s^* = 0$ para $s > p$	$\rho_s^* \rightarrow 0$	$\rho_s^* \neq 0$ para $s \leq p$ $\rho_s^* \rightarrow 0$ para $s > p$

El paso natural para encontrar el verdadero DGP de una muestra será identificar el comportamiento de la ACF y PACF del proceso que ha generado dicha muestra. Naturalmente, no conocemos el proceso ARMA o ARIMA que produjo la muestra bajo estudio y por tanto no conocemos el comportamiento de la ACF y PACF poblacionales y por eso tendremos que estimarlas a partir de la muestra.

En este capítulo discutiremos cómo estimar la ACF y PACF para una muestra y cómo encontrar el orden del proceso ARMA (o ARIMA) empleando dos técnicas comúnmente empleadas. Pero antes de iniciar, generemos una muestra aleatoria de un proceso ARIMA para emplearlo a lo largo de los ejemplos.

Una forma de crear una serie y_t de un DGP ARIMA es empleando la función **arima.sim()** del paquete central de *R* (R Core Team, 2019)¹. Los argumentos más importantes de esta función son

$$\text{arima.sim}(\text{model}, n)$$

donde:

- **model**: una lista que incluye el orden del proceso ARIMA a simular y los coeficientes poblacionales que se emplearán. Por ejemplo, si se quiere crear una muestra a partir del siguiente modelo $AR(2)$: $y_t = 0,25y_{t-1} - 0,5y_{t-2} + \varepsilon_t$, tendremos **model = list(order = c(2, 0, 0), ar = c(0.25, -.5))** o de manera alternativa **model = list(ar = c(0.25, -.5))**.
- **n**: tamaño de la muestra que se creará. Es decir, T .

Esta función crea un objeto de clase *ts* que contienen por defecto una muestra de una series de tiempo proveniente del ARIMA especificado con un término de error que proviene de un ruido blanco gaussiano. La distribución del proceso ruido blanco se puede modificar con el argumento **innov**.

Ahora procedamos a crear tres muestras de tamaño 100 ($T = 100$) para los siguientes procesos ARIMA:

$$y_t = 0,5y_{t-1} + \varepsilon_t \quad (9.1)$$

$$y_t = 0,8\varepsilon_{t-2} + \varepsilon_t \quad (9.2)$$

$$y_t = 0,5y_{t-1} + 0,8\varepsilon_{t-2} + \varepsilon_t \quad (9.3)$$

$$\begin{aligned} y_t &= y_{t-1} + 0,8\varepsilon_{t-2} + \varepsilon_t \\ \Delta y_t &= 0,8\varepsilon_{t-2} + \varepsilon_t \end{aligned} \quad (9.4)$$

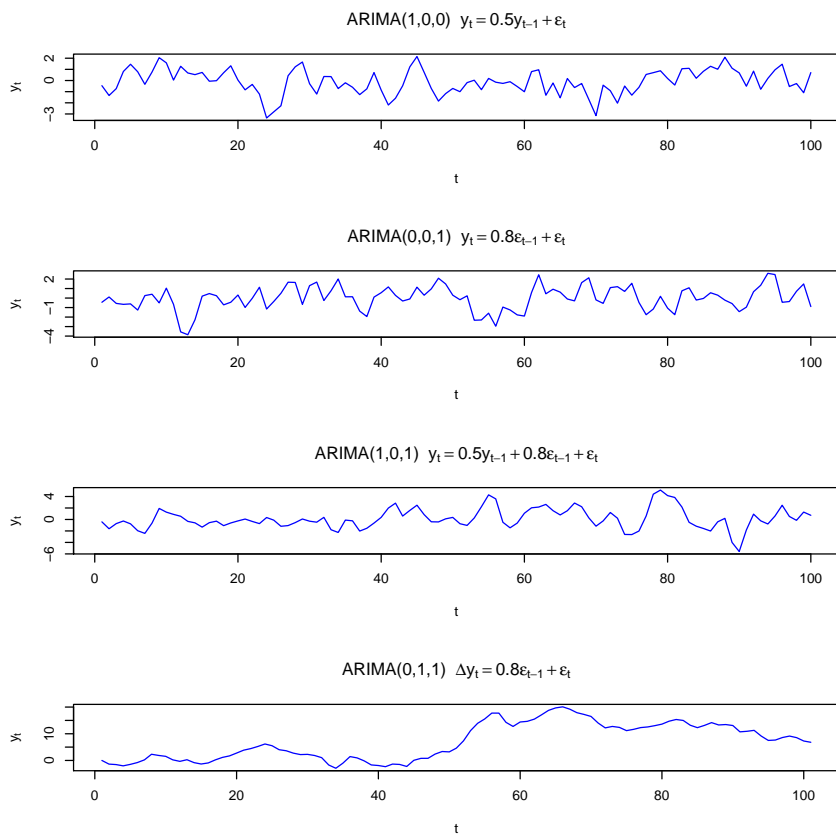
y asignémoslo a los objetos **AR1**, **MA1**, **ARMA11** y **ARIMA011**, respectivamente. Para garantizar que obtendremos siempre los mismo valores, emplearemos una semilla (**set.seed()**)

```
set.seed(123456)
AR1 <- arima.sim(model = list(order = c(1, 0, 0), ar = c(0.5)),
                  n = 100)
MA1 <- arima.sim(model = list(ma = c(0.8)), n = 100)
ARMA11 <- arima.sim(model = list(order = c(1, 0, 1), ar = c(0.5),
                                ma = c(0.8)), n = 100)
ARIMA011 <- arima.sim(model = list(order = c(0, 1, 1), ma = c(0.8)),
                      n = 100)
class(AR1)
## [1] "ts"
```

¹ Es decir, no se requiere cargar ningún paquete para acceder a dicha función.

En la siguiente figura se presenta las series simuladas.

Figura 9.1 Procesos ARMIA(p,d,q) simulados.



9.1. La ACF y PACF muestral

En la práctica no contamos ni con las ACF, ni con la PACF teórica, pero tampoco contamos con la correspondiente media y varianza poblacional. Contamos con una muestra de tamaño T proveniente de un proceso estacionario (o que parece estacionario), entonces podremos estimar la media y la varianza de la manera convencional. Es decir, la media poblacional (μ) puede estimarse por medio del siguiente estimador:

$$\bar{y} = \frac{\sum_{t=1}^T y_t}{T} \quad (9.5)$$

y la varianza poblacional (σ) por medio del siguiente estimador² :

$$\hat{\sigma}^2 = \frac{\sum_{t=1}^T (y_t - \bar{y})^2}{T - 1} \quad (9.6)$$

En el caso de la autocorrelación para el rezago s (ρ_s) , está también se puede estimar empleando el correspondiente análogo muestral. En otras palabras, si la series es estacionaria, el estimador para ρ_s será:

$$\hat{\rho}_s = r_s = \frac{\sum_{t=1}^T (y_t - \bar{y}) (y_{t-s} - \bar{y})}{\sum_{t=1}^T (y_t - \bar{y})^2} \quad (9.7)$$

Así, estimando la autocorrelación para diferentes rezagos se obtendrá la función de autocorrelación muestral (estimada) o SACF (pos su sigla en inglés). Naturalmente, (9.7) corresponde a un estimador puntual, Box y Jenkins (1976) encontraron que la distribución muestral de la SACF es normal cuando el DGP es un proceso MA con errores normales. En la actualidad, para determinar si una autocorrelación poblacional es diferente de cero se acostumbra construir un intervalo de confianza $\pm 2 \frac{1}{\sqrt{T}}$ al rededor de cero. En caso que una autocorrelación esté por fuera de dicho intervalo, podremos rechazar la hipótesis nula individual de que dicha autocorrelación es cero con un nivel de confianza del 95 %.

En *R* podemos calcular la SACF de manera muy sencilla empleando la función **acf** que está en el paquete central (es decir no se requiere cargar ningún paquete para acceder a dicha función). Esta función típicamente incluye los siguientes argumentos:

$$acf(y, lag.max = NULL)$$

donde:

- **y**: la serie de tiempo a la cuál se le quiere estimar la ACF
- **lag.max**: es el número de máximo para el cuál se quiere estimar la ACF. Por defecto la función calcula la SACF para los primeros $10 * \log_{10}(T)$ rezagos. De hecho, Enders (2004) recomienda emplear un máximo de $T/4$ rezagos.

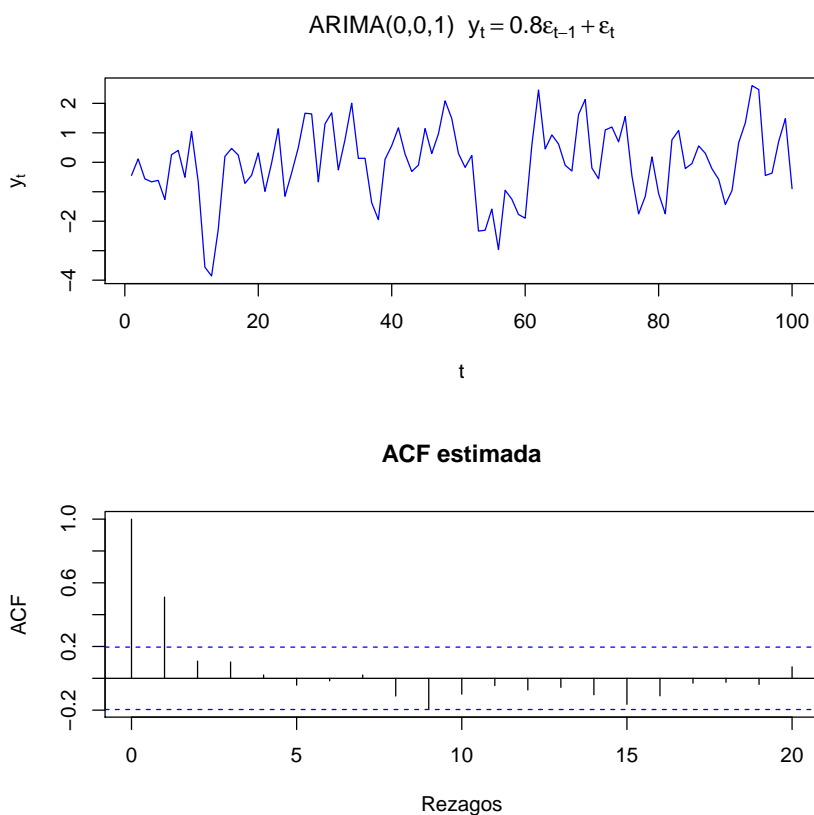
Es importante resaltar que esta función por defecto muestra un gráfico con las SACF y los respectivos intervalos de confianza con un 95 %.

² En algunas ocasiones se acostumbra dividir por T y no por $T - 1$. En ese caso el estimador será sesgado pero consistente.

Veamos un ejemplo empleando la serie de tiempo que guardamos en el objeto **MA1** que generamos de manera aleatoria.

```
layout(matrix(c(1,2), 2,1))
plot.ts(MA1, ylab=expression(y[t]), xlab="t", col="blue",
main = expression(paste("ARIMA(0,0,1) ",
y[t]== 0.8*epsilon[t-1]+epsilon[t])))
acf(MA1, main='ACF estimada', xlab='Rezagos')
```

Figura 9.2 SACF de una serie simulada de un MA(1)



Noten que en este caso, como era de esperarse por las características de la ACF poblacional de un MA(1), solo la autocorrelación para el primer rezago es individualmente estadísticamente diferente de cero.

Por otro lado, es importante reconocer que el intervalo de confianza sólo está construido para controlar el tamaño del error tipo I para pruebas individuales, si se

desea realizar pruebas conjuntas sobre un grupo de autocorrelaciones, lo indicado sería emplear una prueba como la de Box-Pierce o la modificación de Ljung-Box. De hecho, estas últimas pruebas pueden ser muy útiles si lo que se desea es determinar si la serie es un ruido blanco ¿por qué?

Por otro lado, la PACF se puede estimar fácilmente dada su definición. Recuerden que la PACF es equivalente a correr una regresión múltiple en el que la desviación con respecto a la media del periodo actual es explicada por las s desviaciones con respecto a la media. La autocorrelación parcial correspondiente al rezago s es el coeficiente de la desviación s .

Es decir, la autocorrelación parcial para el rezago s para la serie de tiempo y_t corresponde al coeficiente $\phi_s^{(s)}$ ($\phi_s^{(s)} = \rho_s^*$) de la siguiente combinación lineal:

$$\hat{y}_t - \mu = \phi_1^{(s)} (y_{t-1} - \mu) + \phi_2^{(s)} (y_{t-2} - \mu) + \dots + \phi_s^{(s)} (y_{t-s} - \mu) \quad (9.8)$$

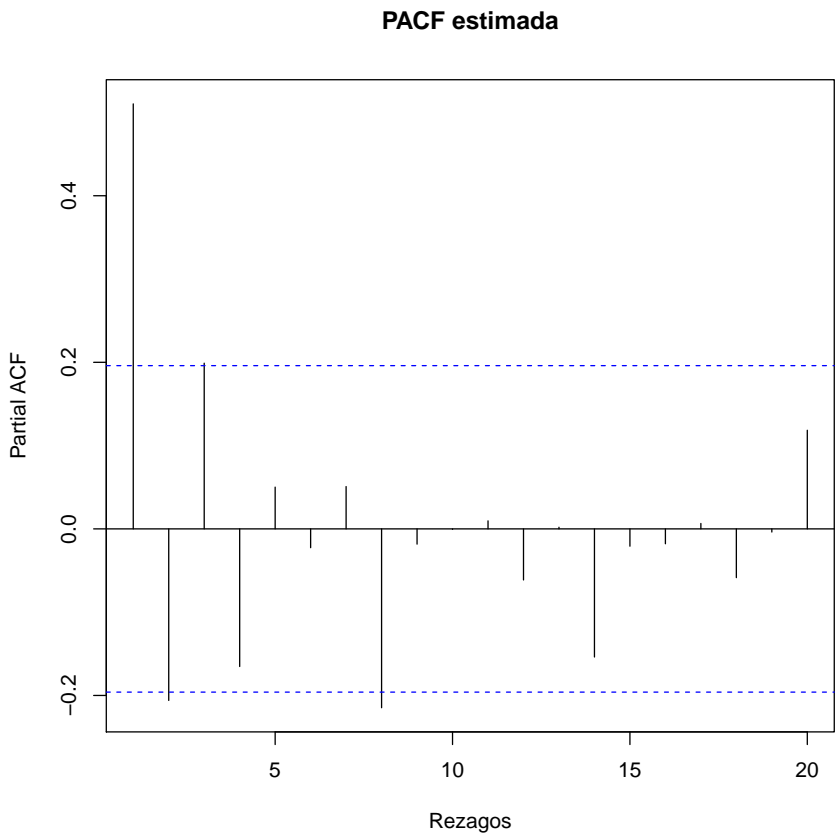
Esta estimación se puede realizar fácilmente en *R* empleando la función **pacf** que también está en el paquete central. Esta función típicamente incluye los siguientes argumentos:

$$\text{pacf}(y, \text{lag.max} = \text{NULL})$$

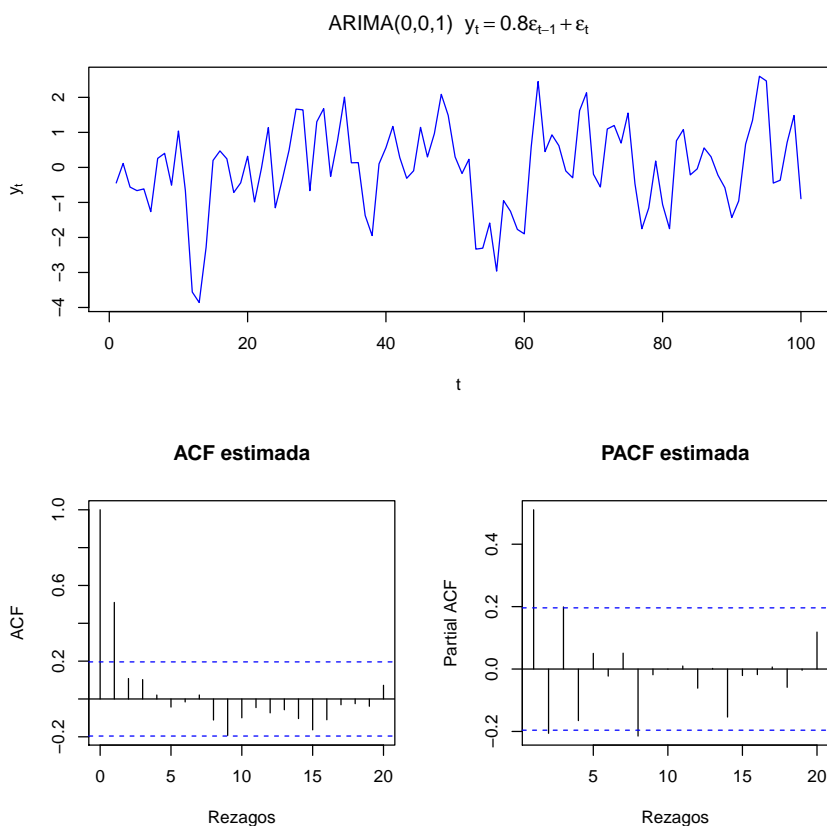
donde los argumentos son similares a los que presenta la función **acf**. Para este caso también es posible construir un intervalo de confianza del 95 % para cada una de las SPACF empleando el rango $\pm 2 \frac{1}{\sqrt{T}}$ al rededor de cero.

Continuando con el ejemplo anterior, la PACF de la serie guardada en el objeto **MA1** se puede estimar por medio de la siguiente línea de código.

```
pacf(MA1, main='PACF estimada', xlab='Rezagos')
```

Poniendo todo junto podemos obtener un gráfico que tenga la serie original, la SACF y SPACF de la siguiente manera

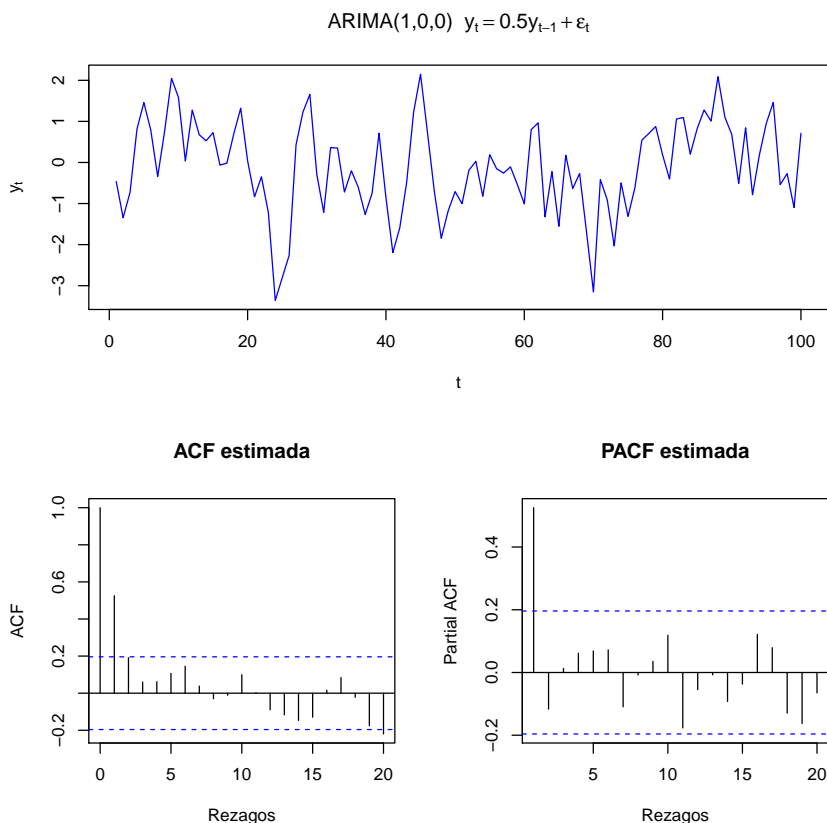
Figura 9.3 SACF y SPACF de una serie simulada de un MA(1)

En la figura 9.3 se presentan los resultados tanto de la estimación de la ACF como de la PACF. Noten que en el caso de la PACF estábamos esperando que ninguna de las autocorrelaciones parciales fuera cero, pues sabemos que para el caso de un MA(1) con un término de media móvil relativamente grande (0.8), las autocorrelaciones parciales tienden a cero pero no son cero. Este ejemplo, muestra lo difícil que es en la práctica distinguir a partir de la ACF y PACF el verdadero DGO de los datos. Para ampliar esta idea consideremos ahora el caso del proceso AR(1) simulado y guardado en el objeto **AR1** con coeficiente autoregresivo 0.5. El correspondiente gráfico donde se muestran las ACF y PACF estimadas se presenta en la figura 9.4.

```
layout(matrix(c(1,1,2,3), 2,2, byrow=TRUE))
plot.ts(AR1, ylab=expression(y[t]), xlab="t", col="blue",
main = expression(paste("ARIMA(1,0,0) ",
y[t]== 0.5*y[t-1]+epsilon[t])))
```

```
acf(AR1, main='ACF estimada', xlab='Rezagos')
pacf(AR1, main='PACF estimada', xlab='Rezagos')
```

Figura 9.4 SACF y SPACF de una serie simulada de un AR(1)



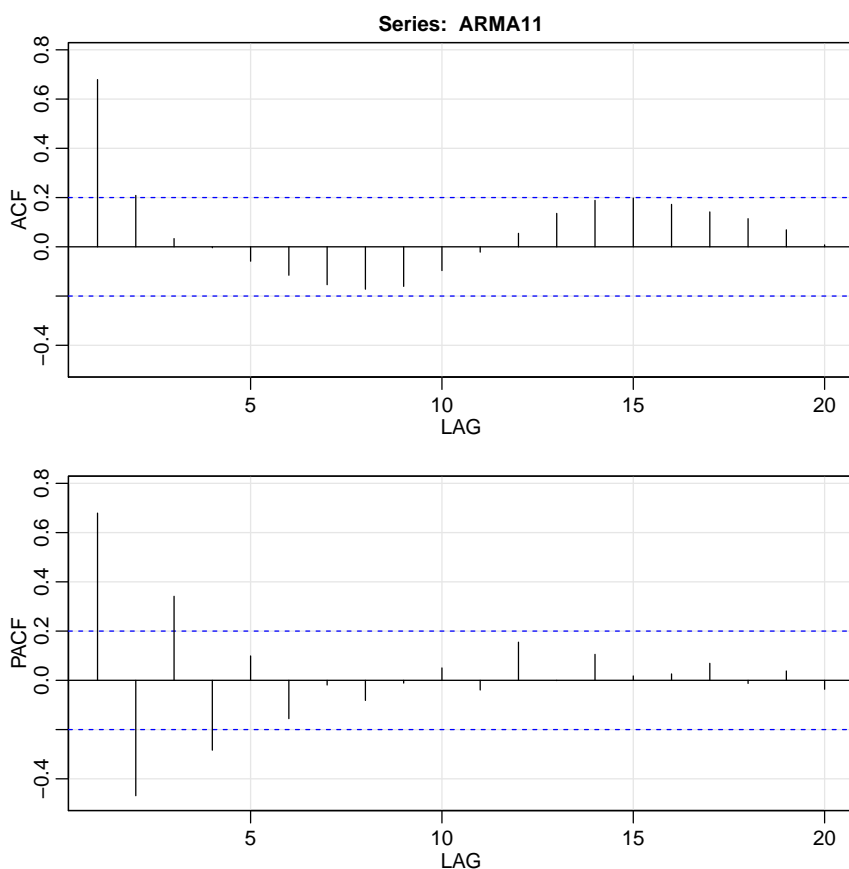
Para el AR(1), sabemos que la PACF poblacional es diferente de cero para el primer rezago e igual a cero para los demás rezagos. En el caso de nuestra simulación la SPACF concuerda con lo que se esperaba de esta muestra, pues sólo para el primer rezago dicha es (individualmente) diferente de cero. Por otro lado, sabemos que para el caso del AR(1) la ACF poblacional nunca será cero, si bien tenderá a cero. Esta característica se puede intuir de cierta manera en nuestra simulación, pero claramente es mucho más fácil llegar a dicha conclusión si se conoce el DGP real. Siguiendo estrictamente las conclusiones de los intervalos de confianza para las

SACF, podríamos llegar a la conclusión que sólo para los primeros cuatro rezagos las autocorrelaciones son diferentes de cero.

Antes de continuar, es importante mencionar que es posible obtener gráficos más estéticos de la ACF y PACF estimadas empleando la función `acf2` del paquete `astsa` (Stoffer, 2017). Esta función genera al mismo tiempo la ACF y la PACF y el argumento más importante es la serie de tiempo a la que se le quiere calcular estas cantidades. A continuación veamos un ejemplo con el objeto **ARMA11**.

```
library(astsa)
acf2(ARMA11)
```

Figura 9.5 SACF y SPACF de una serie simulada de un ARMA(1,1) empleando el paquete `astsa`.



Otra opción es emplear gráficos del paquete `ggplot2` (Wickham, 2016) vía las funciones `ggAcf` y `ggPacf` que viene en el paquete `forecast` (Hyndman y Khan-

dakar, 2008). Estas funciones tienen como argumento principal la serie a la que se quiere calcular la ACF y PACF. En las siguientes dos figuras se observa la serie simulada para el proceso ARIMA en sus niveles (objeto **ARIMA11** y sus respectivas ACF y PACF estimadas, así como para las primeras diferencias de las series.

Figura 9.6 SACF y SPACF de la serie simulada de un ARIMA(0,1,1) empleando el paquete forecast.

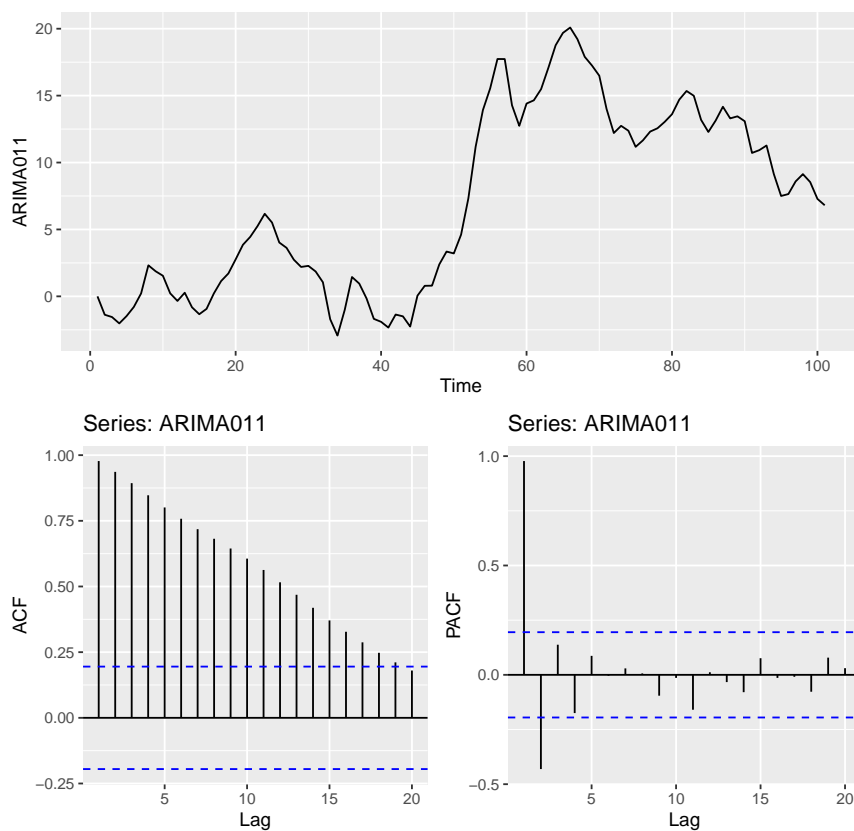
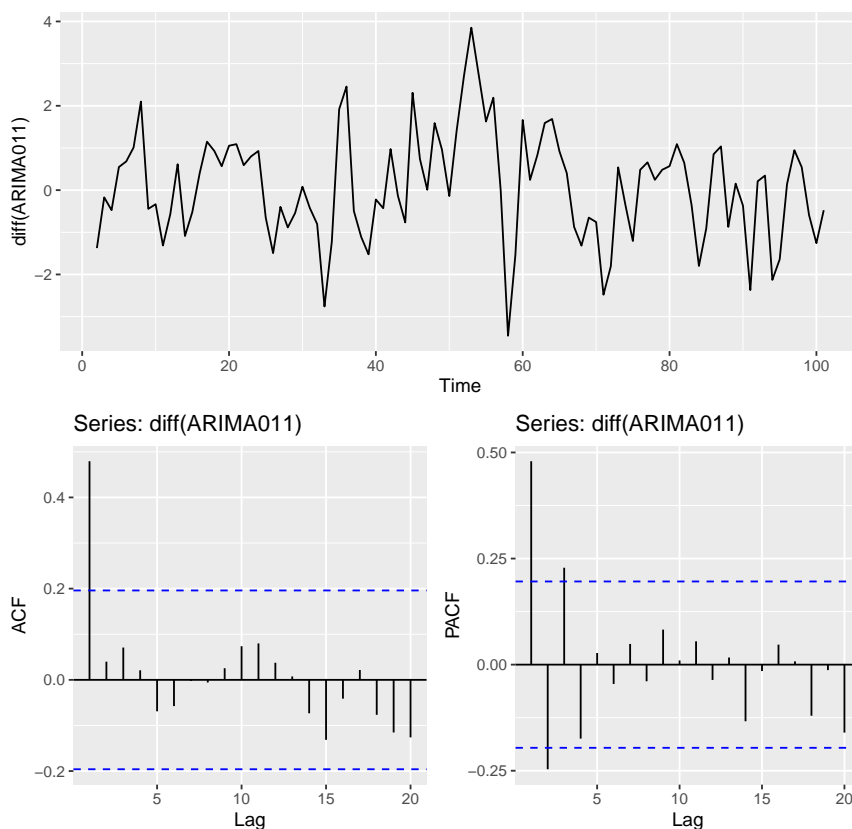


Figura 9.7 SACF y SPACF de las primeras diferencias de la serie simulada de un $\text{ARIMA}(0,1,1)$ empleando el paquete forecast.



Hasta ahora hemos discutido cómo estimar la ACF y PACF para una serie determinada. En teoría, una vez uno conoce el comportamiento de la ACF y de la PACF, se podría determinar el p , d y q del proceso ARIMA y la estimación de los respectivos coeficientes debería ser una tarea fácil. Pero como hemos visto anteriormente, no es una tarea fácil determinar el orden del proceso ARIMA. A continuación se describen dos procedimientos comunes en la literatura para encontrar el modelo ARIMA dada una serie. La primera es la técnica de Box-Jenkins y la segunda es encontrar el mejor modelo empleando criterios de información. La segunda técnica se puede automatizar y por esto en la actualidad es la más empleada. Pero una buena práctica es hacer una combinación de estas dos técnicas como se discutirá más adelante.

9.2. Estimación de un modelo ARIMA con la técnica Box-Jenkins

En la década de los 70, Box y Jenkins popularizaron una técnica de tres pasos para la identificación del orden de los modelos ARMA o ARIMA. La técnica permitió que estos modelos se popularizaran y fueran empleados en diferentes ambientes como una herramienta importante para realizar pronósticos.

La técnica Box-Jenkins implica tres pasos:

- Identificación del modelo
- Estimación de los parámetros
- Diagnóstico del modelo

La fase de *Identificación del modelo* implica determinar visualmente si la serie es estacionaria. En caso que la muestra no fuera estacionaria lo convencional era realizar las transformaciones necesarias para que la serie lo fuera. Por ejemplo, si la serie presentaba una tendencia (i.e. crecía o decrecía en el tiempo como es el caso de la serie que simulamos y guardamos en el objeto **ARIMA11** entonces se sugería diferenciar la serie³; por otro lado, si la serie presentaba una varianza no constante, su sugería emplear una transformación logarítmica para cambiar la escala y en algunas ocasiones la variabilidad disminuye. Otra parte importante de este proceso de “limpiar” las series de la técnica de Box-Jenkins, es determinar si existe comportamientos estacionales que deban ser modelados o removidos⁴. En algunos casos emplear diferenciaciones estacionales⁵ podría quitar el comportamiento estacional.

Finalmente, una vez se tiene una serie estacionaria, por medio de las SACF y SPACF se determina si se deben emplear o no componentes autoregresivos y/o de media móvil en el modelo. En otras palabras, por medio de las SACF y SPACF se determina el p y el q del modelo.

La fase de *Estimación de los parámetros* implica emplear estimadores de Máxima verosimilitud para encontrar tanto los componentes autoregresivos como los de media móvil. Este método de estimación implica un supuesto sobre la distribución de los errores, que convencionalmente se asumen normales, y el uso de rutinas de optimización para encontrar el máximo de la función (Ver Anexos para una discusión mas amplia al respecto).

La etapa del *Diagnóstico del modelo* implica determinar si el residuo del modelo estimado corresponde a un ruido blanco. En otras palabras, esta fase implica determinar si el residuo estimado está libre de autocorrelación y heteroscedasticidad. Así mismo los residuos deberían seguir una distribución normal, si el modelo fue estimado bajo ese supuesto.

³ En la actualidad se emplean pruebas formales para determinar si una serie debe ser diferenciada o no. Más adelante discutiremos este tema en detalle.

⁴ Este tema lo estudiaremos a fondo más adelante.

⁵ Por diferenciación estacional se entiende sacar diferencias a la frecuencia de los datos. Es decir, para datos trimestrales la diferenciación estacional implica restar la observación del periodo t la observación del periodo $t - 4$.

Para constatar si los residuos del modelo estimado no tienen autocorrelación, podemos de manera informal ver las SACF y SPACF de las series y de manera formal emplear las pruebas de:

- Prueba de Rachas
- Prueba de Box-Pierce y Ljung-Box

Todas estas pruebas tienen en común es que la hipótesis de alterna de estas es la existencia de una serie (errores en este caso) no autocorrelacionado de alguna manera. En no de los apéndices de este capítulo se presenta una discusión de cada uno de estas pruebas de autocorrelación.

Para comprobar la posible existencia de heteroscedasticidad, nos concentraremos en el tipo de heteroscedasticidad mas común en las series de tiempo: un comportamiento ARCH (heteroscedástico condicional en el tiempo) o GARCH. En capítulos siguientes nos concentraremos en este tema. Por ahora es importante mencionar que, informalmente, una forma gráfica de chequear si existe o no un comportamiento heteroscedástico condicional en el tiempo es empleando gráficos de las SACF y SPACF de los residuos elevados al cuadrado⁶ o los residuos absolutos. Las pruebas formales para este supuesto las veremos más adelante.

Finalmente, podemos evaluar si los residuos siguen una distribución normal o no. En este caso podemos emplear de manera informal el gráfico de cuantiles acumulado de probabilidad (q-q normality plot en inglés). Y formalmente, se pueden emplear diferentes pruebas de normalidad como lo discuten Alonso y Montenegro, 2015 y Alonso y Estrada (2018)). En este momento nos concentraremos en las pruebas de Jarque-Bera y Shapiro. Estas dos pruebas tienen en común que su hipótesis nula es que los datos provienen de una población que sigue una distribución normal, versus la hipótesis alterna que los datos no siguen una distribución normal. En otras palabras, realmente estas pruebas son de no-normalidad.

Finalmente, es importante resaltar que esta técnica puede implicar detectar mas de un modelo que puede concordar con la SACF y SPACF. en ese caso la técnica de Box-Jenkins sugieren emplear el modelo más **parsimonioso**; es decir, el modelo con menor número de parámetros que explique igual de “bien” a la muestra.

9.2.1. *Ejemplo con una serie simulada*

Para ilustrar esta técnica, empleemos la serie simulada que guardamos en el objeto **ARIMA11**. En la práctica no conoceríamos el DGP que generó los datos, y por tanto por ahora supondremos que no sabemos dicho DGP.

⁶ Asegúrese que entiende porqué esta es una buena idea.

9.2.1.1. Identificación del modelo

El primer paso implica la identificación del modelo, para esto podemos emplear la Figura 9.6. Claramente la serie crece y por tanto aparentemente no es estacionaria. Entonces, siguiendo la recomendación de esta técnica podemos proceder a diferenciar la serie. Los resultados se encuentran en la Figura 9.7. El código que generó dicha figura es el siguiente.

```
g1 <- autoplot(diff(ARIMA011))
g2 <- ggAcf(diff(ARIMA011))
g3 <- ggPacf(diff(ARIMA011))

ggarrange(g1, ggarrange(g2, g3, nrow=1, ncol = 2), nrow = 2)
```

En este caso la serie no crece en el tiempo y parece tener una media constante, tampoco parece tener una volatilidad no constante. Es decir, tras diferenciar una vez, obtenemos una serie que parece estacionaria. Por tanto, podríamos definir a $d = 1$. Ahora miremos la respectivas ACF y PACF estimadas (Ver Figura 9.7).

Al observar las SACF y SPACF podemos ver que las autocorrelaciones parciales para el primer rezagos es estadísticamente diferentes de cero (de manera individual y con un 95 % de confianza), al igual que las autocorrelación parcial. No obstante parece que la autocorrelaciones parciales parecen decaer de manera oscilatoria. Este comportamiento concuerda con lo esperado en un modelo MA(1). Así que un posible candidato es un modelo ARIMA(0,1,1)⁷.

9.2.1.2. Estimación de los parámetros

La fase de *Estimación de los parámetros* se puede llevar a cabo rápidamente con una línea de código empleando la función **arima** que está en el paquete central de R. Esta función típicamente incluye los siguientes argumentos:

arima(y, order = c(0, 0, 0), xreg = NULL, include.mean = TRUE)

donde:

- “y”: la serie de tiempo a la cuál se le estimará el modelo ARIMA
- “order”: es un listado que muestra el orden del modelo ARIMA. Por ejemplo, si se desea estimar un modelo ARIMA(2,0,3), tendremos que escribir **order=c(2,0,3)**.
- “xreg”: corresponde a una matriz de variables exógenas explicativas. De hecho, cuando se emplean variables explicativas en un modelo ARIMA, este se conoce con el nombre de modelos ARIMAX. En este libro no trataremos este tipo de modelos.

⁷ Noten que esta observación no es obvia y dos investigadores podrían llegar a conclusiones diferentes. Por esto esta técnica tiene cierta parte de Arte. Normalmente, la experiencia del investigador lo va llevando a poder visualizar el comportamiento de modelos teóricos en la muestra. Pero claramente esta es la mayor debilidad de esta técnica.

- “include.mean”: este argumento nos permite incluir o no un intercepto en el modelo. Por defecto los modelos incluyen intercepto.

Continuando con el ejemplo, estimemos el modelo ARIMA(0,1,1) para la muestra generada aleatoriamente y guardemos el resultado en el objeto **modelo1**.

```
modelo1 <- arima(ARIMA011, order=c(0, 1, 1))
```

Los resultados de la estimación del modelo guardados en el objeto **modelo1** se pueden visualizar de diferentes maneras, por ejemplo

```
summary(modelo1)

##
## Call:
## arima(x = ARIMA011, order = c(0, 1, 1))
##
## Coefficients:
##          ma1
##          0.7445
## s.e.      0.0733
##
## sigma^2 estimated as 1.049:  log likelihood = -144.69,  aic = 293.38
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE
## Training set 0.03958595 1.019144 0.8249496 42.53019 86.64006
##              MASE      ACF1
## Training set 0.8102038 -0.01186388

library(lmtest)
coeftest(modelo1)

##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 0.744476   0.073267  10.161 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

9.2.1.3. Diagnóstico del modelo

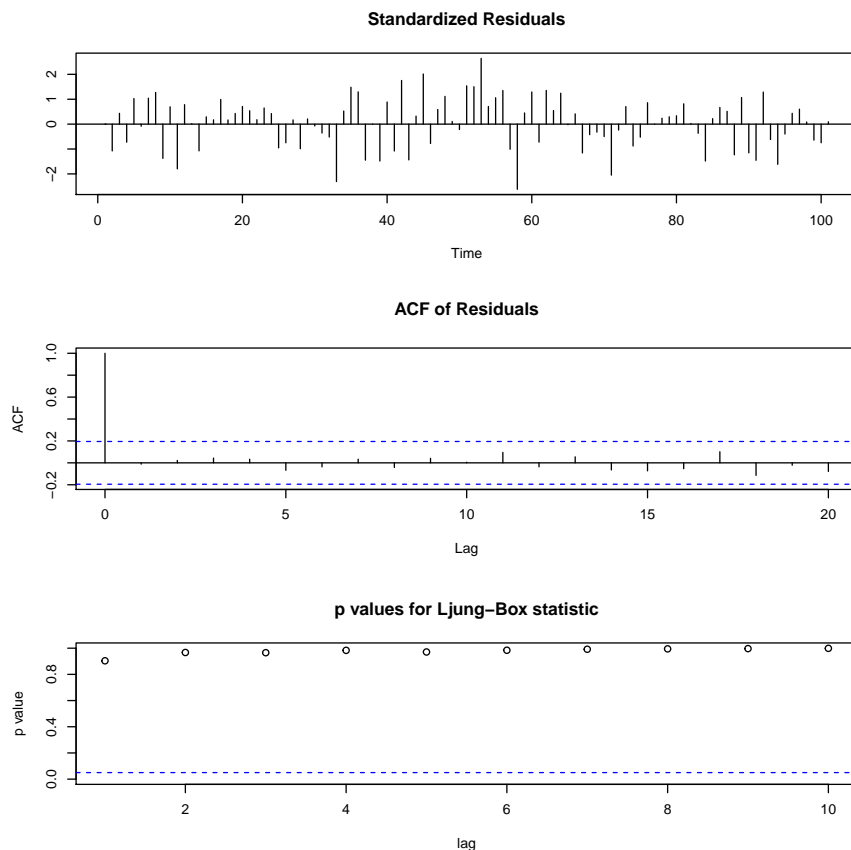
En el objeto **modelo1** se ha guardado todos los resultados de la estimación del modelo, pero antes de pasar a sacar conclusiones o emplear el modelo para por ejemplo realizar proyecciones, es importante realizar el diagnóstico de los residuos.

Autocorrelación

Para la etapa del *Diagnóstico del modelo* tenemos que determinar si los residuos estimados son o no ruido blanco. Una manera no exhaustiva de chequear rápidamente si el modelo tiene o no autocorrelación es emplear la función **tsdiag** que muestra

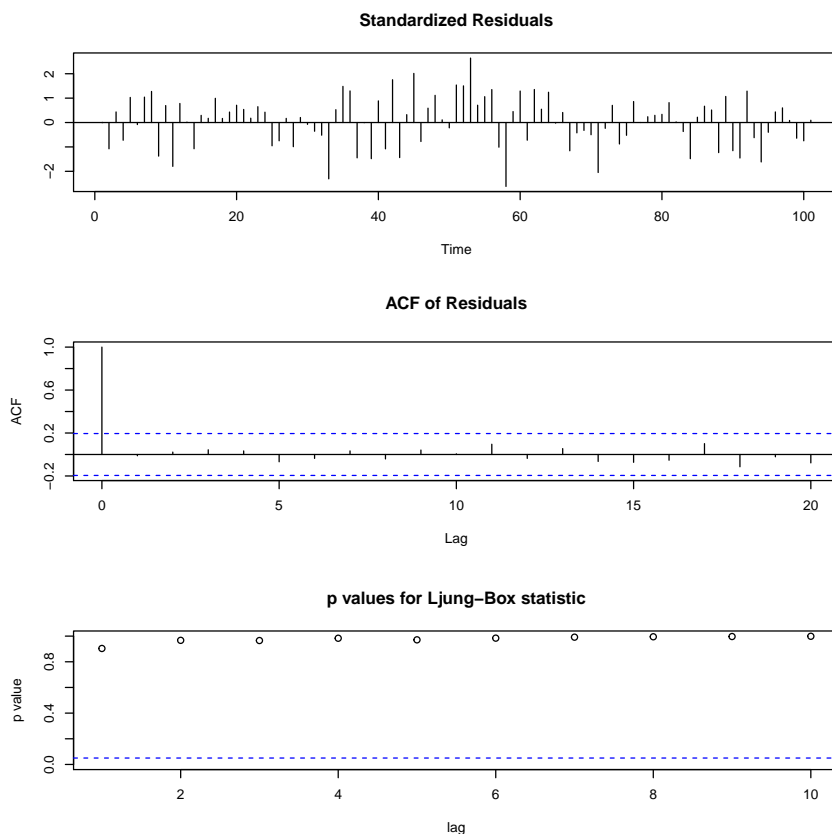
un gráfico de los residuos estandarizados⁸, la ACF de los residuos y el valor p para la prueba Ljung-Box para los primeros diez rezagos. En este caso la función se puede aplicar de la siguiente manera.

```
tsdiag(modelo1)
```



Empleando la figura 9.8 no se puede rechazar para ninguno de los rezagos la hipótesis individual que la autocorrelación es cero. En el tercer panel podemos ver que las pruebas conjuntas, no permiten rechazar la hipótesis de que todas las autocorrelaciones (hasta el rezago 10) son conjuntamente iguales a cero. Es decir, parece que no existe autocorrelación en los residuos.

⁸ Otra opción es la función `checkresults()` del paquete `forecast`. La diferencia es que en vez de incluir el objeto en el que se tiene el modelo, esta función requiere la serie de los residuos. Mas adelante se muestra como obtenerla.

Figura 9.8 Prueba Ljung-Box para el modelo estimado (Función `tsdiag()`)

Para continuar con nuestro diagnóstico, debemos emplear los residuos del modelo estimado. Los residuos los podemos extraer rápidamente con la función **residuals**.

```
res <- residuals (modelo1)
```

Para realizar la prueba de rachas, podemos emplear la función **runs.test()** del paquete **tseries** (Trapletti y Hornik, 2018). Ya discutimos esta función en el Capítulo 4 (ver sección 4.3.2)

Nuestro primer paso para efectuar esta prueba es convertir los residuos estimados en una variable dicotómica que muestre el signo del residual para cada periodo.

```
library(tseries)
signo.error <- factor(res>0)
head(signo.error, 3)
```

```
## [1] FALSE FALSE TRUE
```

```
## Levels: FALSE TRUE
runs.test(signo.error)

##
##  Runs Test
##
## data: signo.error
## Standard Normal = -0.48813, p-value = 0.6255
## alternative hypothesis: two.sided
```

Los resultados no permiten rechazar (valor p de 0.6255) la no autocorrelación y por tanto se puede concluir que no existe autocorrelación en los residuos del modelo estimado. Es decir, con esta prueba y en este aspecto, los residuos parecen un ruido blanco.

Este análisis debe complementarse con otras pruebas. La prueba de Box-Pierce y la modificación de Ljung-Box pueden calcularse empleando la misma función (**Box.test()**) de la base de R. *Box.test*. Esta función fue discutida en el Capítulo 4 (ver sección 4.3.2).

Entonces para probar la hipótesis $H_0 : \gamma_1 = 0$ con la prueba de Box-Pierce podemos emplear el siguiente código

```
Box.test(res, lag = 1)

##
##  Box-Pierce test
##
## data: res
## X-squared = 0.014216, df = 1, p-value = 0.9051
```

Así, no podemos rechazar la hipótesis nula de no autocorrelación (para el primer rezago). Como se mencionó anteriormente, es común probar esta hipótesis para los primeros rezagos, por lo menos los primeros 20. Es decir,

$$\begin{aligned} H_0 : \gamma_1 &= 0 \\ H_0 : \gamma_1 &= \gamma_2 = 0 \\ &\vdots \\ H_0 : \gamma_1 &= \gamma_2 = \dots = \gamma_{20} = 0 \end{aligned}$$

Para realizar estas prueba podemos emplear la función *tabla.Box.Pierce* creada en Capítulo 4 (ver sección 4.3.2). Empleando dicha función podemos crear el el Cuadro 9.2.

Los resultados nos permiten concluir que las autocorrelaciones de los errores son cero para todos los rezagos analizados. Así, podemos concluir que no existe autocorrelación en los residuos de acuerdo con esta prueba.

El lector puede generar fácilmente los resultados de la prueba de Ljung-Box aplicada a los mismos residuos. No debe sorprender que los resultados de esta última prueba sean los mismo que los obtenidos con la prueba de Box-pierce, pues en este caso la muestra es grande. Así, la corrección para muestras pequeñas no era

Cuadro 9.2 Prueba de Box-Pierce de los errores para los primeros rezagos

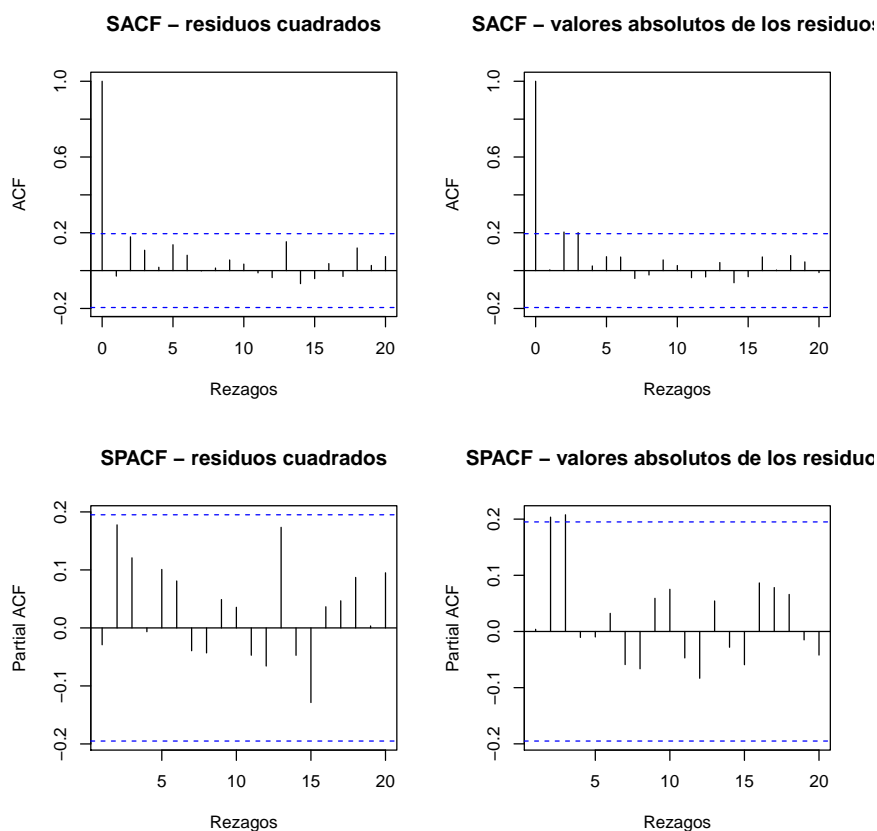
Rezagos	Box-Pierce	p-valor
1.00	0.01	0.91
2.00	0.06	0.97
3.00	0.26	0.97
4.00	0.37	0.98
5.00	0.84	0.97
6.00	0.97	0.99
7.00	1.08	0.99
8.00	1.26	1.00
9.00	1.43	1.00
10.00	1.43	1.00
11.00	2.35	1.00
12.00	2.49	1.00
13.00	2.79	1.00
14.00	3.22	1.00
15.00	3.76	1.00
16.00	4.04	1.00
17.00	5.08	1.00
18.00	6.40	0.99
19.00	6.44	1.00
20.00	7.06	1.00

importante en este caso. Las dos pruebas coinciden que los residuales del modelo ARIMA estimado no están autocorrelacionados.

Homoscedasticidad

Ahora nos queda por chequear si existe problemas de heteroscedasticidad, en especial un comportamiento tipo ARCH o GARCH (en un próximo capítulo discutiremos en detalle este tema). Una forma gráfica de chequear si existe o no un comportamiento heteroscedástico condicional en el tiempo es empleando gráficos de las SACF y SPACF de los residuos elevados al cuadrado y los residuos absolutos.

```
layout(matrix(c(1,2,3,4), 2,2))
acf(res^2, main='SACF - residuos cuadrados', xlab='Rezagos')
pacf(res^2, main='SPACF - residuos cuadrados', xlab='Rezagos')
acf(abs(res), main='SACF - valores absolutos de los residuos',
    xlab='Rezagos')
pacf(abs(res), main='SPACF - valores absolutos de los residuos',
    xlab='Rezagos')
```

Figura 9.9 Gráficos para detectar comportamiento ARCH

De acuerdo a la figura 9.9, no parece existir problemas de autocorrelación o autocorrelación parcial entre los residuos al cuadrado o los residuos absolutos. Es decir, no tenemos problemas de heteroscedasticidad.

Como se mencionó en el Capítulo 4 (ver sección 4.3.3), una aproximación para determinar si existe un comportamiento GARCH o ARCH es emplear la prueba de Ljung-Box sobre la serie (sin media) al cuadrado. En este caso, al tratarse de unos residuales de un modelo estimado con intercepto, no es necesario restar la media.

Los resultados de la Tabla 9.3, no permiten concluir que existe una autocorrelación de las varianzas y por tanto no hay comportamiento ARCH o GARCH.

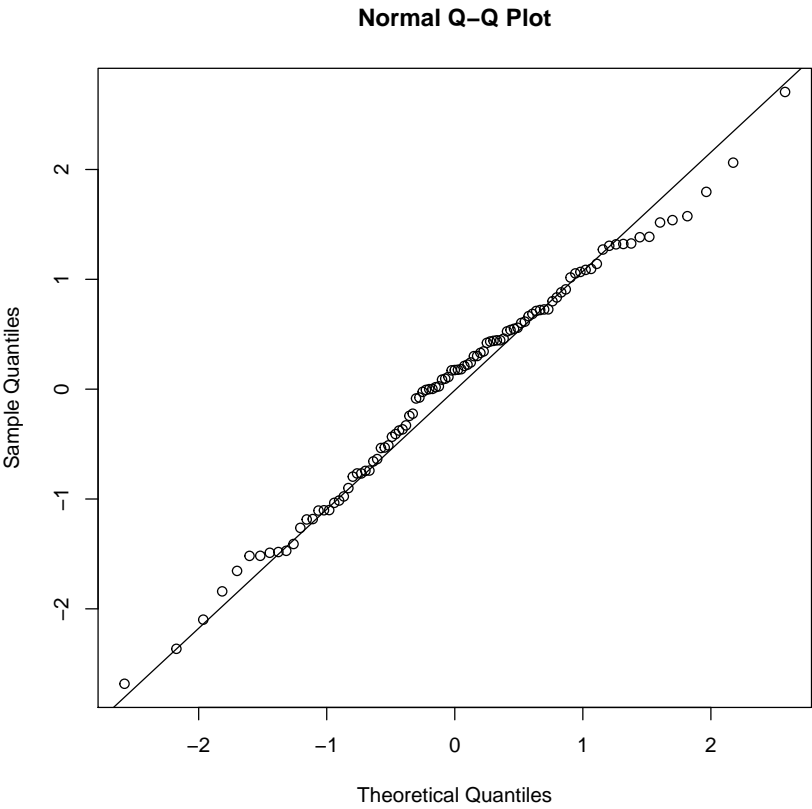
Normalidad

Finalmente, podemos evaluar si los residuos siguen una distribución normal o no. Primero hagamos un análisis gráfico empleando el gráfico q-q que fue discutido en el Capítulo 4 (ver sección 4.3.4)

Cuadro 9.3 Prueba de Ljung-Box para los errores al cuadrado (primeros 20 rezagos)

Rezagos	Ljung-Box	p-valor
1.00	0.09	0.77
2.00	3.43	0.18
3.00	4.66	0.20
4.00	4.69	0.32
5.00	6.71	0.24
6.00	7.43	0.28
7.00	7.43	0.39
8.00	7.45	0.49
9.00	7.80	0.55
10.00	7.94	0.63
11.00	7.96	0.72
12.00	8.12	0.78
13.00	10.86	0.62
14.00	11.43	0.65
15.00	11.65	0.71
16.00	11.82	0.76
17.00	11.93	0.80
18.00	13.72	0.75
19.00	13.82	0.79
20.00	14.52	0.80

Figura 9.10 Gráficos q-q para los residuos



La Figura 9.10 permite observar que existen unas observaciones extremas que distan mucho de la distribución normal. Esto podría generar algún tipo de alarma sobre el no cumplimiento del supuesto de normalidad. Pero la decisión final la tomaremos con las pruebas estadísticas discutidas en Capítulo 4 (ver sección 4.3.4).

```
##  
## Jarque Bera Test  
##  
## data: res  
## X-squared = 1.1124, df = 2, p-value = 0.5734  
##  
## Shapiro-Wilk normality test  
##  
## data: res  
## W = 0.98958, p-value = 0.6241
```

Para ambas pruebas, no se puede rechazar la hipótesis de que los errores provienen de una distribución normal. Así, parece ser que los errores son ruido blanco gaussiano.

Noten que hasta aquí hemos probado que nuestro modelo candidato, el ARI-MA(0,1,1), cumple los supuestos realizados sobre el término de error. Entonces podríamos emplear este modelo para hacer inferencia o realizar proyecciones.

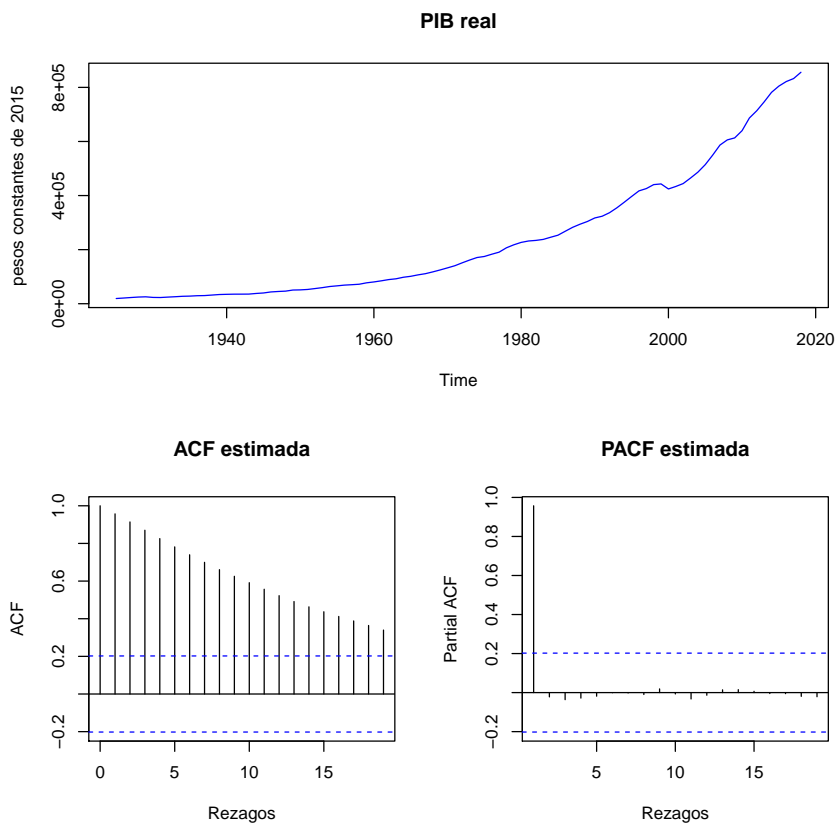
Por otro lado, en esta aproximación deberíamos tener varios modelos candidatos y realizar este mismo análisis para cada modelo y después seleccionar entre los modelos que cumplen los supuestos el mas parsimonioso.

9.2.2. *Un ejemplo con le PIB real de Colombia*

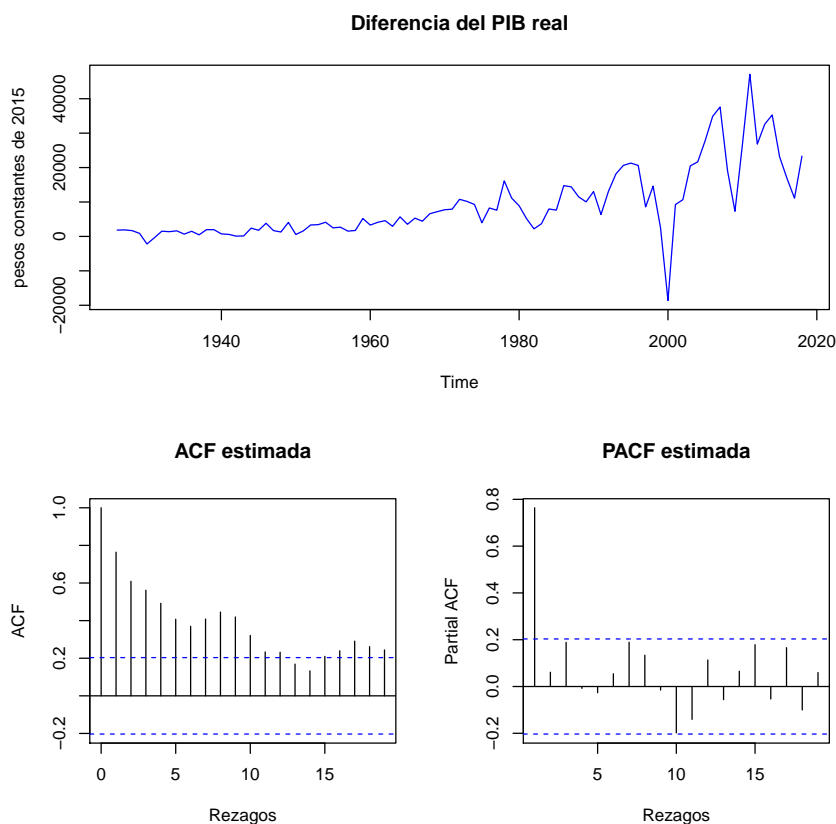
Ahora repliquemos el proceso anterior, pero empleando la serie del PIB real de Colombia a precios constantes de 2015 que se encuentra disponible en el archivo *pib.csv*. Los datos tiene como fuente el DANE y el empalme fue realizado en el Cienfi. Nuestra tarea es encontrar el mejor modelo ARIMA para describir el comportamiento del PIB real empleando la la técnica Box-Jenkins.

9.2.2.1. Identificación del modelo

Empecemos cargando los datos y graficando la ACF y PACF muestral. Deberán obtener un gráfico como el siguiente.

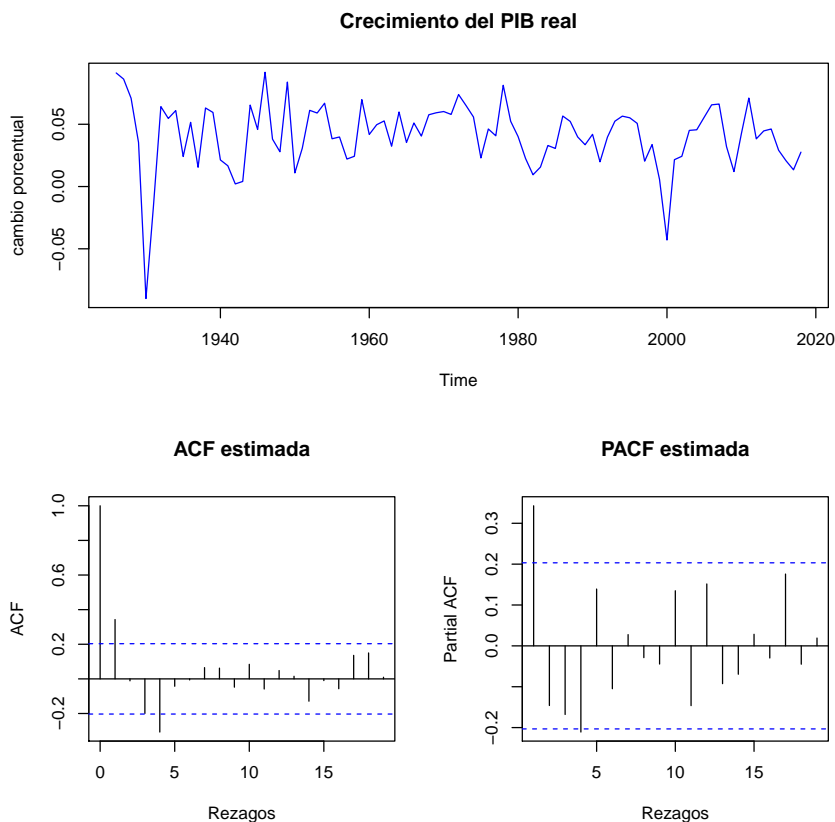
Figura 9.11 SACF y SPACF de la serie del PIB real de Colombia

En la figura anterior podemos ver como la serie aparentemente no es estacionaria, pues crece en el tiempo. Por tanto, diferenciamos la serie y repetamos el procedimiento. Obtendremos el siguiente gráfico.

Figura 9.12 SACF y SPACF de la diferencia del PIB real de Colombia

Esta serie ya parece no crecer en el tiempo, pero parece no tener una volatilidad constante. Para solucionar este problema, calculemos el logaritmo (natural) de la serie⁹ y diferenciémoslo. Obtendremos el siguiente gráfico.

⁹ En R el logaritmo natural se calcula empleando la función **log()**

Figura 9.13 SACF y SPACF del crecimiento del logaritmo natural del PIB real de Colombia

Esta serie ya tiene un comportamiento más parecido al de una serie estacionaria. Trabajaremos con esta serie. De acuerdo a la ACF y PACF parece que tenemos un modelo $\text{ARIMA}(0,1,1)$ para el logaritmo del PIB, pues la ACF tiene solo el primer rezago claramente diferente de cero y la PACF muestral parece estar cayendo a medida que aumentamos los rezagos. Por otro lado, también podríamos pensar en un modelo $\text{ARIMA}(1,1,1)$. Estimemos estos dos los compararemos.

9.2.2.2. Estimación de los parámetros y diagnóstico del modelo

Estimemos los dos modelos para el logaritmo del PIB real y guardémoslos en los objetos **ModeloPIB.1** y **ModeloPIB.2**, respectivamente. Además guardemos la serie de los residuales de cada modelo. Esto lo podemos hacer con las siguientes

```
ModeloPIB.1 <- arima(log(PIBreal), order=c(0, 1, 1))
res.1 <- residuals (ModeloPIB.1)
ModeloPIB.2 <- arima(log(PIBreal), order=c(1, 1, 1))
res.2 <- residuals (ModeloPIB.2)
```

Ahora empleemos la prueba de rachas para determinar si los errores de estos modelos están o no autocorrelacionados.

```
signo.error.1 <- factor(res.1>0)
runs.test(signo.error.1)

##
##  Runs Test
##
## data: signo.error.1
## Standard Normal = 0.70273, p-value = 0.4822
## alternative hypothesis: two.sided

signo.error.2 <- factor(res.2>0)
runs.test(signo.error.2)

##
##  Runs Test
##
## data: signo.error.2
## Standard Normal = -1.2047, p-value = 0.2283
## alternative hypothesis: two.sided
```

En ambos casos la nula de no autocorrelación no se puede rechazar. Ahora miremos la prueba de Ljung-Box.

En los dos modelos ocurre lo mismo: existe autocorrelación en los residuales. El lector puede verificar las SACF y SPACF de los residuos, esto refuerza la idea que los modelos aún tienen problemas de autocorrelación. La mejor forma de resolver este problema es incluyendo más términos MA o AR.

El lector puede jugar un poco con diferentes especificaciones. Una especificación con relativamente pocos parámetros que puede funcionar es el ARIMA(3,1,2)¹⁰. En este caso los resultados de la prueba de rachas y de Ljung-Box permiten concluir con un nivel de confianza del 99% que no hay autocorrelación en este modelo.

```
ModeloPIB.3 <- arima(log(PIBreal), order=c(3, 1, 2))
res.3 <- residuals (ModeloPIB.3)
signo.error.3 <- factor(res.3>0)
runs.test(signo.error.3)

##
##  Runs Test
##
## data: signo.error.3
## Standard Normal = 0.90685, p-value = 0.3645
## alternative hypothesis: two.sided
```

¹⁰ El lector encontrará que el modelo ARIMA(3,1,1) también resuelve el problema de autocorrelación, pero tiene un problema de errores heteroscedásticos.

Cuadro 9.4 Prueba de Ljung-Box de los errores para el modelo ARIMA(0,1,1) para los primeros rezagos

Rezagos	Ljung-Box	p-valor
1.00	10.36	0.00
2.00	14.74	0.00
3.00	17.69	0.00
4.00	20.46	0.00
5.00	21.25	0.00
6.00	21.36	0.00
7.00	21.37	0.00
8.00	23.64	0.00
9.00	28.97	0.00
10.00	36.46	0.00
11.00	41.66	0.00
12.00	43.38	0.00
13.00	43.46	0.00
14.00	46.90	0.00
15.00	49.48	0.00
16.00	52.44	0.00
17.00	54.18	0.00
18.00	56.11	0.00
19.00	59.24	0.00
20.00	66.11	0.00

Cuadro 9.5 Prueba de Ljung-Box de los errores para el modelo ARIMA(0,1,1) para los primeros rezagos

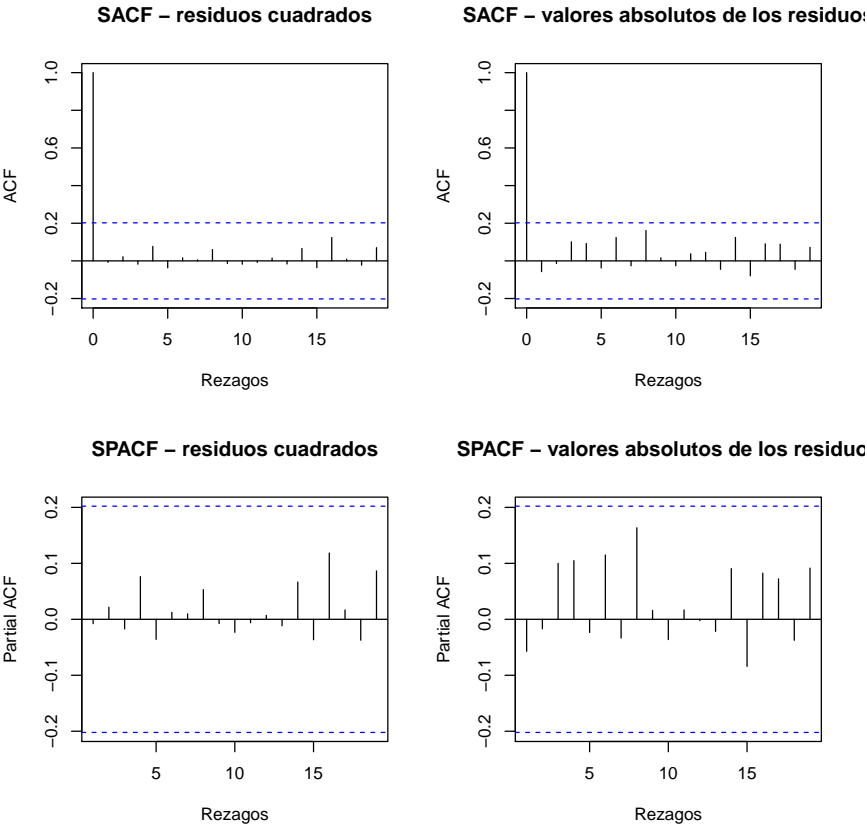
Rezagos	Ljung-Box	p-valor
1.00	0.20	0.65
2.00	3.04	0.22
3.00	7.24	0.06
4.00	17.89	0.00
5.00	18.11	0.00
6.00	18.15	0.01
7.00	19.08	0.01
8.00	19.59	0.01
9.00	21.08	0.01
10.00	22.96	0.01
11.00	24.27	0.01
12.00	25.00	0.01
13.00	25.40	0.02
14.00	28.52	0.01
15.00	28.59	0.02
16.00	30.68	0.01
17.00	32.28	0.01
18.00	33.68	0.01
19.00	34.78	0.01
20.00	40.16	0.00

Ahora miremos si los errores tienen heteroscedasticidad, en especial un comportamiento tipo ARCH o GARCH. En la siguiente figura se presenta las SACF y SPACF de los residuos elevados al cuadrado y los residuos absolutos.

Cuadro 9.6 Prueba de Ljung-Box de los errores para el modelo ARIMA(3,1,2) para los primeros rezagos

Rezagos	Ljung-Box	p-valor
1.00	0.02	0.90
2.00	0.04	0.98
3.00	2.11	0.55
4.00	9.49	0.05
5.00	9.52	0.09
6.00	9.53	0.15
7.00	9.79	0.20
8.00	10.71	0.22
9.00	12.61	0.18
10.00	15.24	0.12
11.00	17.23	0.10
12.00	18.05	0.11
13.00	18.52	0.14
14.00	20.79	0.11
15.00	21.37	0.13
16.00	23.25	0.11
17.00	24.74	0.10
18.00	26.52	0.09
19.00	27.63	0.09
20.00	32.81	0.04

Figura 9.14 Gráficos para detectar comportamiento ARCH del modelo ARIMA(3,1,2)



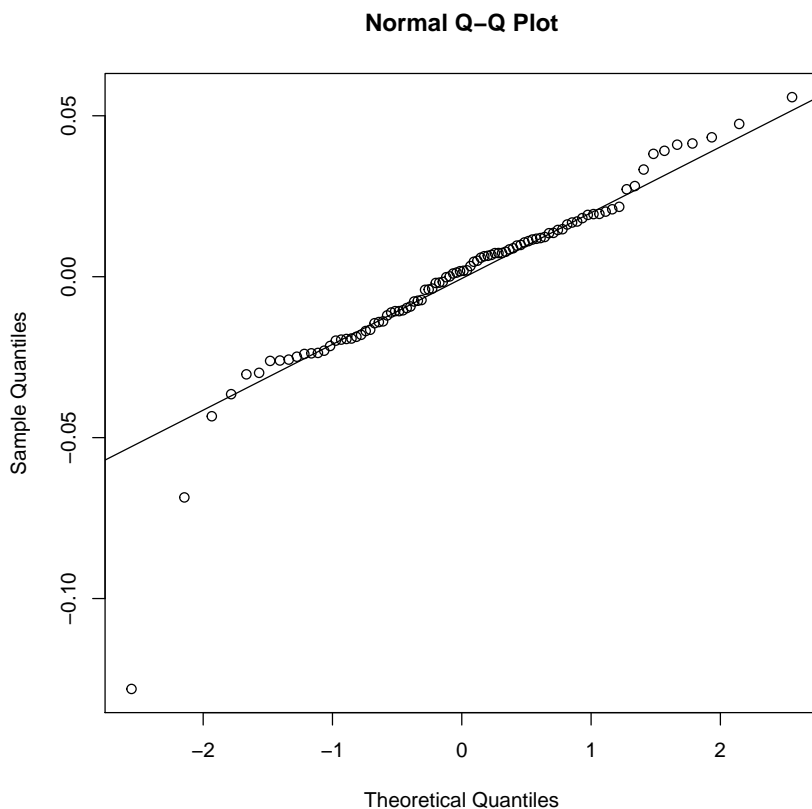
De acuerdo a la figura 9.14, parece no existir problemas de autocorrelación o autocorrelación parcial entre los residuos al cuadrado o su valor absoluto. Podemos concluir que este modelo (el ARIMA(3.1.2)) no tiene autocorrelación ni heteroscedasticidad.

En el Cuadro 9.7 se encuentra la prueba para de Ljung-Box para los residuales al cuadrado. Los resultados corroboran las conclusiones que se obtienen con los gráficos: no existe comportamiento GARCH.

Cuadro 9.7 Prueba de Ljung-Box de los errores al cuadrado para el modelo ARIMA(3,1,2)

Rezagos	Ljung-Box	p-valor
1.00	0.01	0.94
2.00	0.05	0.97
3.00	0.08	0.99
4.00	0.68	0.95
5.00	0.82	0.98
6.00	0.84	0.99
7.00	0.85	1.00
8.00	1.23	1.00
9.00	1.25	1.00
10.00	1.28	1.00
11.00	1.29	1.00
12.00	1.31	1.00
13.00	1.34	1.00
14.00	1.83	1.00
15.00	1.98	1.00
16.00	3.76	1.00
17.00	3.77	1.00
18.00	3.83	1.00
19.00	4.42	1.00
20.00	5.33	1.00

Finalmente, podemos evaluar si los residuos siguen una distribución normal o no.

Figura 9.15 Gráficos q-q para los residuos del modelo ARIMA(3,1,2)

```
jarque.bera.test(res.3)
```

```
##
```

```
## Jarque Bera Test
```

```
##
```

```
## data: res.3
```

```
## X-squared = 181.43, df = 2, p-value < 2.2e-16
```

```
shapiro.test(res.3)
```

```
##
```

```
## Shapiro-Wilk normality test
```

```
##
```

```
## data: res.3
```

```
## W = 0.90757, p-value = 5.955e-06
```

Para ambas pruebas y el gráfico q-q, podemos concluir que los errores no siguen una distribución normal. Esto tiene varias implicaciones, el estimador de Máxima verosimilitud puede no comportarse bien en esta situación. Podemos emplear otros métodos como el de Minimizar la suma condicional de los residuos al cuadrado (CSS por el inglés conditional-sum-of-squares). El problema mas delicado estará cuando realicemos intervalos para los pronósticos en ese caso tendremos que emplear técnicas como el Bootstrapping para generar los intervalos. En la sección 9.4 discutiremos esta opción.

9.3. Criterios de información y selección automática de modelos ARMA

La sección anterior, es un buen ejemplo de lo difícil que puede ser seleccionar un modelo ARMA o ARIMA a partir de las SACF y SPACF. Otra opción, que es la más empleada en la actualidad, es preguntarnos que tan bien ajusta un modelo a los datos y probar un conjunto grande de modelos y seleccionar el “mejor” modelo de acuerdo a su ajuste.

De hecho ya sabemos que existe un compromiso entre reducir la suma cuadrada de los errores y la perdida de grados de libertad por incluir más parámetros estimados. Así, podemos construir diferentes criterios que penalicen la mejora en el ajuste por estar incluyendo más parámetros en el modelo. Por otro lado, dado que estamos en un ambiente de Máxima Verosimilitud, podemos emplear criterios que incluyan el valor de la función de máxima verosimilitud (qué tan bueno es el modelo). Los criterios más empleados son:

- *criterio de Akaike (Hirotugu AKAIKE, 1981)* (AIC, por su sigla en inglés):

$$AIC(p, q) = \ln(\hat{\sigma}^2) + \frac{2(p+q)}{T}$$

- *criterio Bayesiano de Schwarz (Schwarz, 1978)* (SBC, por su sigla en inglés):

$$SBC(p, q) = \ln(\hat{\sigma}^2) + \frac{\ln(T) 2(p+q)}{T}$$

- *criterio de Hannan y Quinn (Hannan y Quinn, 1979)* (HQ):

$$HQ(p, q) = \ln(\hat{\sigma}^2) + \frac{\ln(\ln(T)) 2(p+q)}{T}$$

Noten que dada la forma como están definidos estos criterios de información, será deseable un modelo con un criterio de información menor. Es decir, la idea es minimizar el criterio de información seleccionado.

En general, hoy en día es muy común emplear la rapidez de los computadores para estimar un número relativamente grande de modelos y a cada uno calcularle los

criterios de información. Y después seleccionar el modelo que minimice el respectivo criterio de información. Naturalmente, los residuos del modelo seleccionado deben ser probados para determinar si son ruido blanco o no. La librería *forecast* tiene la función **auto.arima** que permite realizar dicha operación automáticamente. Esta función típicamente incluye los siguientes argumentos:

```
auto.arima(y, max.p=5, max.q=5, max.order=5, stationary=FALSE,
           ic=c("aic", "aicc", "bic"), stepwise=TRUE, xreg=NULL)
```

donde:

- “y”: la serie de tiempo a la cuál se le estimará el “mejor” modelo ARIMA.
- “max.p”: el número máximo de términos autoregresivos que se incluirán en la búsqueda.
- “max.q”: el número máximo de términos de media móvil que se incluirán en la búsqueda.
- “stationary”: TRUE si se quiere restringir la búsqueda a solo modelo estacionarios.
- “ic”: criterio de información que se empleará para realizar la búsqueda. Hay dos versiones de AIC disponibles.
- “xreg”: corresponde a una matriz de variables exógenas explicativas. En este libro no trataremos este tipo de modelos.
- “stepwise”: FALSE si se desea que se evalúen todas las posibles combinaciones de modelos. Esto puede ser un poco demorado. TRUE sí se desea que la búsqueda del modelo que minimiza el correspondiente criterio de información se realice de manera “inteligente”. En otras palabras no se evalúen combinaciones para las cuales se preve que el correspondiente criterio de información está disminuyendo.

9.3.1. Ejemplo con una serie simulada

Continuando con nuestro ejemplo de la serie simulado, podemos emplear las siguientes líneas de código para automatizar el proceso de selección del mejor modelo empleando los criterios de información AICC, AICC y BIC.

```
auto.arima(ARIMA011, max.p = 10, max.q = 10,
           ic = 'aic', stepwise=FALSE)

## Series: ARIMA011
## ARIMA(0,1,1)
##
## Coefficients:
##      ma1
##      0.7445
## s.e.  0.0733
##
## sigma^2 estimated as 1.06: log likelihood=-144.69
## AIC=293.38   AICc=293.51   BIC=298.59
```

```

auto.arima(ARIMA011, max.p = 10, max.q = 10,
           ic = 'aicc', stepwise=FALSE)

## Series: ARIMA011
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##          0.7445
## s.e.    0.0733
##
## sigma^2 estimated as 1.06:  log likelihood=-144.69
## AIC=293.38   AICc=293.51   BIC=298.59

auto.arima(ARIMA011, max.p = 10, max.q = 10,
           ic = 'bic', stepwise=FALSE)

## Series: ARIMA011
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##          0.7445
## s.e.    0.0733
##
## sigma^2 estimated as 1.06:  log likelihood=-144.69
## AIC=293.38   AICc=293.51   BIC=298.59

```

Noten que, para este caso, los tres criterios de información coinciden en seleccionar como mejor modelo el ARIMA(0,1,1). Esto coincide con uno de los modelos que intentamos probar en la sección anterior. En ese caso encontramos que dicho modelo cumplía todos los supuestos. Es decir llegamos a una respuesta similar, pero sin tanto esfuerzo. Pero no siempre será así, como veremos con el siguiente ejemplo.

9.3.2. *Un ejemplo con le PIB real de Colombia*

Ahora regresemos a la serie del PIB. Ya sabemos que es buena idea trabajar con el logaritmo de la serie, así que podemos emplear el siguiente código para encontrar el mejor modelo de manera automática empleando los criterios de información AIC, AICC y BIC.

```

auto.arima(log(PIBreal), max.p = 10, max.q = 10,
           ic = 'aic', stepwise=FALSE)

## Series: log(PIBreal)
## ARIMA(4,1,0) with drift
##
## Coefficients:
##          ar1          ar2          ar3          ar4      drift
##          0.3322   -0.1005   -0.0992   -0.2312   0.0407

```

```
## s.e. 0.1015 0.1074 0.1068 0.1023 0.0023
##
## sigma^2 estimated as 0.0005916: log likelihood=216.06
## AIC=-420.12 AICc=-419.14 BIC=-404.92

auto.arima(log(PIBreal), max.p = 10, max.q = 10,
            ic = 'aicc', stepwise=FALSE)

## Series: log(PIBreal)
## ARIMA(4,1,0) with drift
##
## Coefficients:
##      ar1      ar2      ar3      ar4  drift
##      0.3322 -0.1005 -0.0992 -0.2312 0.0407
## s.e. 0.1015 0.1074 0.1068 0.1023 0.0023
##
## sigma^2 estimated as 0.0005916: log likelihood=216.06
## AIC=-420.12 AICc=-419.14 BIC=-404.92

auto.arima(log(PIBreal), max.p = 10, max.q = 10, ic = 'bic',
            stepwise=FALSE)

## Series: log(PIBreal)
## ARIMA(0,1,1) with drift
##
## Coefficients:
##      ma1  drift
##      0.3790 0.0409
## s.e. 0.0979 0.0035
##
## sigma^2 estimated as 0.0006298: log likelihood=211.76
## AIC=-417.52 AICc=-417.25 BIC=-409.92
```

Otra vez, los criterios AIC y AICC seleccionan el mismo modelo: ARIMA(4,1,0). El criterio BIC selecciona el modelo ARIMA(0,1,1). Este último modelo lo habíamos seleccionado con el otro método y encontramos que no cumplía el supuesto de no autocorrelación del error. Procedamos a chequear si el modelo ARIMA(4,1,0) cumplen los supuestos.

```
ModeloPIB.4 <- arima(log(PIBreal), order=c(4, 1, 0))
res.4 <- residuals (ModeloPIB.4)
signo.error.4 <- factor(res.4>0)
runs.test(signo.error.4)

##
##  Runs Test
##
## data: signo.error.4
## Standard Normal = -0.27496, p-value = 0.7833
## alternative hypothesis: two.sided
```

La prueba de rachas provee evidencia de la no autocorrelación de los residuales. La prueba de Ljung-Box permite rechazar la nula de no autocorrelación para a partir del rezago 10 (Ver Cuadro 9.8). Así este modelo no es un buen modelo. Lo que

Cuadro 9.8 Prueba de Ljung-Box de los errores para el modelo ARIMA(3,1,2) para los primeros rezagos

Rezagos	Ljung-Box	p-valor
1.00	0.44	0.51
2.00	1.28	0.53
3.00	2.83	0.42
4.00	16.22	0.00
5.00	16.80	0.00
6.00	16.88	0.01
7.00	17.66	0.01
8.00	18.57	0.02
9.00	20.79	0.01
10.00	24.32	0.01
11.00	26.72	0.01
12.00	27.58	0.01
13.00	28.02	0.01
14.00	32.23	0.00
15.00	33.08	0.00
16.00	36.50	0.00
17.00	38.19	0.00
18.00	40.08	0.00
19.00	42.28	0.00
20.00	49.06	0.00

debemos hacer en este caso es buscar un modelo cercano a este que cumpla los supuestos. Eso ya lo habíamos hecho en la sección anterior, encontrando que un mejor modelo es el ARIMA(3,1,0).

9.4. Construcción del pronóstico y su intervalo

En la sección anterior discutimos como estimar un modelo ARIMA que cumpla los supuestos del error. En esta sección construiremos un pronóstico y su respectivo intervalo de confianza a partir de un modelo ARIMA estimado.

Antes de discutir en el detalle de cómo realizar el pronóstico en R, es importante reconocer que para construir un pronóstico y su respectivo intervalo de confianza para de un modelo ARIMA cuyos residuales se comportan como un ruido blanco será necesario:

1. determinar el número de periodos para los que se construirá el pronóstico; es decir, el horizonte (h).
2. suponer una distribución del error

Si se supone una distribución normal, entonces un intervalo de confianza del $(1 - \alpha)\%$ para el pronóstico para h pasos adelante (h -steps-ahead en inglés) está dado por:

$$\hat{y}_{T+h|T} \pm z_{\alpha/2} \hat{\sigma}_h \quad (9.9)$$

donde $z_{\alpha/2}$ es el correspondiente valor de la distribución normal y $\hat{\sigma}_h$ es el error estándar del pronóstico.

Por otro lado, si la distribución del error no es normal, entonces una opción para construir el intervalo es simular la distribución de los errores. Esto se puede realizar con una técnica estadística denominada *bootstrapping*. El bootstrapping implica generar un número relativamente grande de muestras aleatorias con reposición de los errores y reestimar el correspondiente parámetro de interés. Esto implicaría tener una muestra simulada relativamente grande del parámetro de interés de la cuál podemos derivar el correspondiente intervalo, que por cierto no necesariamente será simétrico.

Para construir los pronósticos podemos emplear la función *forecast* del paquete **forecast** (Hyndman y Khandakar, 2008) del paquete del mismo nombre.

Esta función típicamente incluye los siguientes argumentos:

forecast(object, h, level = c(80, 95), bootstrap = FALSE)

donde:

- **object**: el objeto que contiene el modelo a partir del cual se quiere realizar las proyecciones.
- **h**: es el número de periodos que se quiere proyectar (este es el horizonte de pronósticos) Es decir, determina el orden de los pronósticos para h-step-ahead.
- **level = c(80, 95)**: El nivel de confianza para los intervalos de confianza de la predicción. Por defecto se calculan los límites inferiores y superiores para el pronóstico para los niveles de 80 % y 95 %.
- **bootstrap**: determina si se emplea bootstrapping para construir los intervalos para los pronósticos o no. En caso que los datos no sigan una distribución normal es posible emplear la técnica de *bootstrapping* (**bootstrap = TRUE**). En este caso la distribución del residuo es simulada a partir de los residuos estimados y por tanto no implica ningún supuesto de distribución. El valor por defecto de este argumento es **FALSE**.

A continuación construyamos los respectivos pronósticos para los dos ejemplos que veníamos trabajando.

9.4.1. Ejemplo con serie simulada

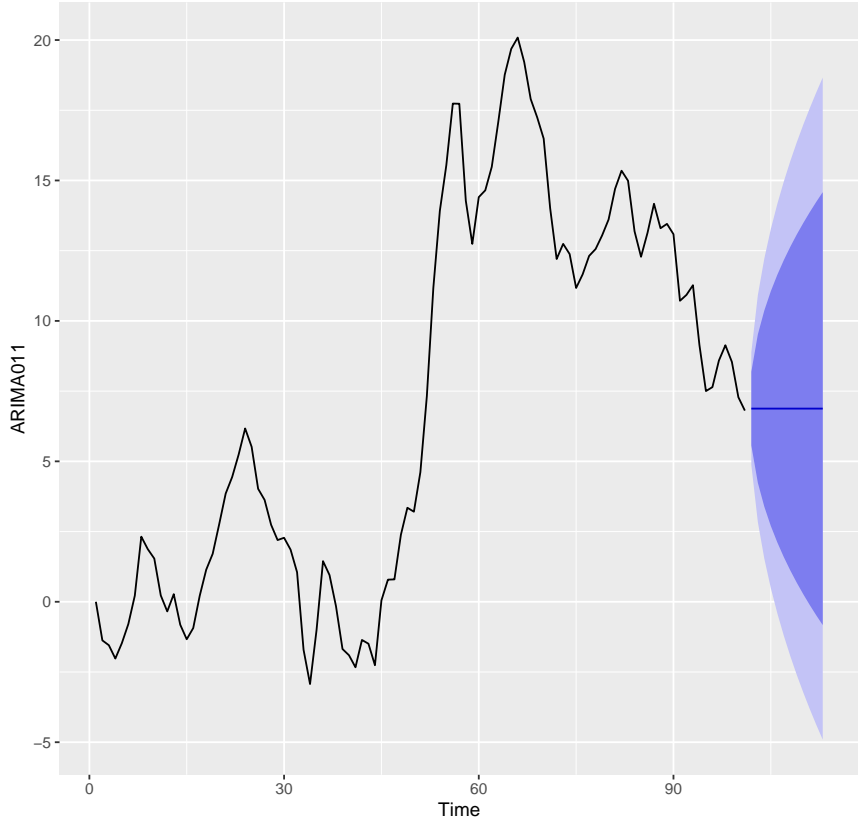
En las secciones anteriores determinamos que el mejor modelo para la serie simulada es el ARIMA(0,1,1). El modelo fue guardado en el objeto **modelo1**. Además, encontramos que los residuales del modelo era ruido blanco y además seguía una distribución normal. Así, podemos emplear (9.9) para construir el intervalo de confianza. Si deseamos construir un pronóstico para los siguientes 12 períodos podemos emplear el siguiente código.

```
f.modelo1 <- forecast(modelo1, h= 12)
f.modelo1
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## 102	6.877614	5.5650140	8.190214	4.8701656	8.885062
## 103	6.877614	4.2382767	9.516951	2.8410960	10.914132
## 104	6.877614	3.3834354	10.371792	1.5337291	12.221499
## 105	6.877614	2.6999991	11.055229	0.4885034	13.266724
## 106	6.877614	2.1136190	11.641609	-0.4082876	14.163515
## 107	6.877614	1.5918945	12.163333	-1.2061964	14.961424
## 108	6.877614	1.1172309	12.637997	-1.9321316	15.687359
## 109	6.877614	0.6788080	13.076420	-2.6026416	16.357869
## 110	6.877614	0.2694086	13.485819	-3.2287639	16.983992
## 111	6.877614	-0.1160660	13.871294	-3.8182965	17.573524
## 112	6.877614	-0.4813766	14.236604	-4.3769908	18.132219
## 113	6.877614	-0.8293909	14.584619	-4.9092328	18.664461

Los resultados se presenta en la Figura 9.16.

Figura 9.16 Pronóstico de la serie simulada para los siguientes 12 períodos



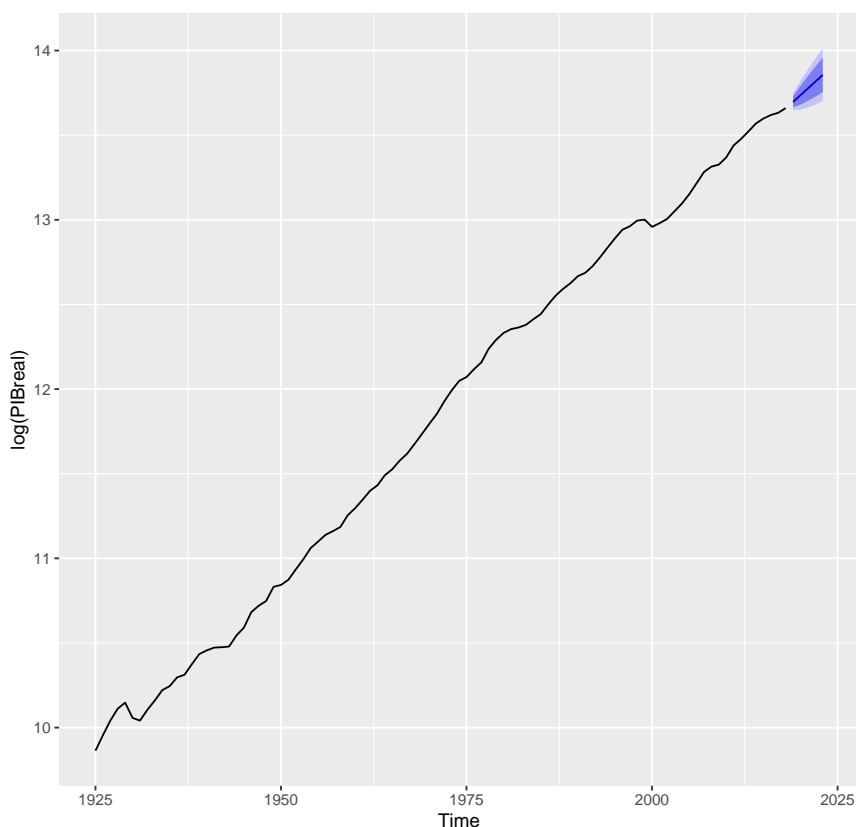
9.4.2. Ejemplo con le PIB real de Colombia

En las secciones anteriores encontramos que la serie anual del PIB podía representarse por medio de un modelo ARIMA(3,1,0) que guardamos en el objeto **ModeloPIB.3**. También mostramos que los residuales eran ruido blanco, pero no seguían una distribución normal. Así, en este caso para construir el intervalo de confianza la mejor opción sería emplear bootstrapping. Supongamos que queremos construir pronósticos para los siguientes 5 años, esto lo podemos hacer de la siguiente manera:

```
f.ModeloPIB.3 <- forecast(ModeloPIB.3, h= 5)
f.ModeloPIB.3
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## 2019	13.69688	13.66481	13.72895	13.64783	13.74592
## 2020	13.73594	13.68037	13.79151	13.65095	13.82093
## 2021	13.77598	13.70293	13.84902	13.66426	13.88770
## 2022	13.81606	13.72763	13.90449	13.68082	13.95130
## 2023	13.85630	13.75466	13.95794	13.70086	14.01174

Los resultados se presentan en la Figura 9.17.

Figura 9.17 Pronóstico del PIB real para los siguientes 5 períodos

9.5. Pronósticos e intervalos para sumas de períodos

En algunas ocasiones se desea tener un pronóstico que corresponde a la suma de unos períodos. Por ejemplo, si se tiene 12 pronósticos para cada mes y se desea tener el pronóstico del año. En estos casos el estimador puntual (el centro del intervalo de confianza) corresponde a la suma de los 12 pronósticos mensuales. No obstante, los límites del intervalo de confianza no corresponde a la suma de los 12 meses, ya que en este caso existe una correlación entre los errores de pronóstico. Así en estos casos es mejor emplear una técnica de simulación.

Por ejemplo, supongamos por un momento que la serie simulada en este capítulo corresponde a datos mensuales y se quiere construir un pronóstico semestral a partir de la suma de los primeros 6 meses ya pronosticados. En este caso el pronóstico y su respectivo intervalo de confianza se puede obtener simulando un número grande de trayectorias (algo similar al bootstrapping). Por ejemplo, empleemos 10000 si-

mulaciones. Esto se puede realizar con la función la función *simulate* del paquete **forecast** (Hyndman y Khandakar, 2008).

Esta función típicamente incluye los siguientes argumentos:

simulate(object, nsim, seed = NULL, bootstrap = FALSE)

donde:

- **object**: el objeto que contiene el modelo a partir del cual se quiere realizar las simulaciones.
- **nsim**: es el número de periodos que se quiere proyectar (este es el horizonte de pronósticos) Es decir, determina el orden de los pronósticos para h-step-ahead.
- **seed**: Permite definir una semilla para las simulaciones. El valor por defecto es NULL.
- **bootstrap**: determina si se emplea bootstrapping para construir los intervalos para cada una de las simulaciones o no. El valor por defecto de este argumento es **FALSE**.

El código corresponde a:

```
n.simula <- 10000
h <- 6
simulaciones <- NULL
for(i in 1:n.simula) {
  simulaciones[i] <- sum(simulate(modelo1, nsim=h))
}
mean(simulaciones)

## [1] 41.37979

sum(f.modelo1$mean[1:6])

## [1] 41.26568
```

Pronóstico para la suma de los siguientes 6 meses con la simulación corresponde a 41.38, mientras que la suma de los pronósticos realizados anteriormente es 41.266. Noten que estos dos números son relativamente similares. Entre más grande sea el número de periodos agregados, estos dos métodos se pueden separar un poco mas. El intervalo de confianza simulado para este agregado se puede obtener con la función *quantile* del paquete **forecast** (Hyndman y Khandakar, 2008). El código para calcular intervalos de confianza del 90 % y 85 % es

```
quantile(simulaciones, prob=c(0.05, 0.95))

##          5%          95%
## 15.99766 66.29094

quantile(simulaciones, prob=c(0.025, 0.975))

##          2.5%          97.5%
## 11.01749 71.61792
```

9.6. Comentarios Finales

En este capítulo hemos estudiado dos técnicas para encontrar el mejor modelo ARIMA que potencialmente generó una muestra. La técnica Box-Jenkins que implica tener que adivinar a partir de la SACF y SPACF cuál sería el modelo ARIMA. Por otro lado, está la técnica de selección automática del modelo empleando criterios de información. En la práctica la mejor aproximación es emplear una mezcla de los dos.

Una buena práctica es graficar las serie de tiempo y sus respectivas ACF y PACF. Hacer las transformaciones que se requiera para encontrar una serie aparentemente estacionaria. Esto nos dará una idea que es modelo podría ser factible o por lo menos el vecindario en el cuál estaría el mejor modelo. Después podemos emplear los algoritmos de selección automática de modelos que emplean criterios de información. Una vez se tengan los o el modelo seleccionado automáticamente, debemos proceder a chequear los supuestos del modelo. Si los supuestos no se cumplen, podemos intentar aumentando la parte autoregresiva o de media móvil, siguiendo la intuición que provee el análisis inicial de las ACF y PACF muestrales. Siempre tendremos que chequear que los supuestos se cumplan.

Una vez tengamos un modelo que cumple los supuestos, podemos proceder a emplear el modelo para chequear la significancia de los coeficientes, realizar predicciones u otros análisis requeridos. En el próximo capítulo discutiremos la comparación de diferentes modelos teniendo en cuenta su capacidad de predecir.

Capítulo 10

Tipos de pronósticos

El uso más común de los modelos ARMA o ARIMA discutidos en el capítulo 8 y 9 y los modelos de suavización discutidos en el Capítulo 3 es generar proyecciones o pronósticos para los siguientes períodos. En este capítulo discutiremos los elementos más importantes sobre efectuar pronósticos con estos modelos modelos ARIMA. Naturalmente, el paso anterior a construir un pronóstico es tener un buen modelo que represente el DGP que generó la muestra bajo estudio. Es decir un modelo que tenga un buen ajuste (o fit en inglés) para la muestra.

De hecho es muy importante tener en cuenta que no necesariamente un modelo que se comporte bien en la muestra se vaya a comportar bien por fuera de la muestra. Así, es común que cuando se construyen modelos para pronosticar, se emplee como criterio de selección entre diferentes modelos (que cumplan los supuestos) seleccionar aquél que se comporta mejor fuera de muestra.

Cuando empleamos modelos univariados para generar pronósticos para los siguientes períodos existe siempre una disyuntiva de cómo actualizar nuestro modelo. Este capítulo discute este problema¹.

10.1. Diferentes métodos para la generación de pronósticos

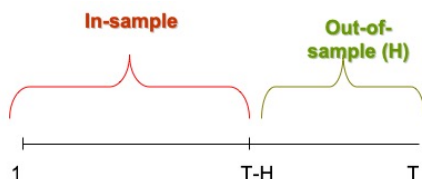
Existen numerosas formas de evaluar el comportamiento de unos pronósticos pero en todo caso, la evaluación siempre implicará poder tener las realizaciones del proceso para aquellos periodos que se generan los pronósticos. En otras palabras, para evaluar si un pronóstico es bueno o no se requiere tener el valor real futuro. De esta manera se puede comparar el valor pronosticado y el real.

Para resolver el problema práctico de que no contamos con observaciones del futuro, es común que se guarde una parte muy pequeña de la muestra para emplear-

¹ En este capítulo solo emplearemos modelos ARIMA para ilustrar los conceptos que desarrollaremos, pero el lector debería hacer el ejercicio con los otros modelos estudiados en el libro. Por simplicidad solo nos restringiremos a los modelos ARIMA en este capítulo, pero todas las técnicas que estudiamos aquí se pueden aplicar a todos los modelos cubiertos en este libro.

la como muestra de evaluación. Los pronósticos que se hagan en esta muestra se llaman pronósticos por fuera de la muestra o pronósticos **out-of-sample**. Llamemos al número de periodos que se guardarán de la muestra H^2 . Así en este caso lo que se acostumbra es reestimar los modelos con $T - H$ observaciones y realizar el correspondiente pronóstico para los H periodos. (ver figura 10.1).

Figura 10.1 Fuera de muestra y en la muestra



Para construir los H pronósticos fuera de muestra podemos tomar diferentes aproximaciones. Una aproximación es emplear el mismo modelo y coeficientes estimados con la muestra de tamaño $T - H$. A este tipo de pronóstico se le conoce de ventana fija (ver Figura 10.2). En este caso se realizarán pronósticos hasta H pasos adelante³. Otra opción implica que cada vez que se genera un nuevo pronóstico, actualizar los coeficientes del modelo ARIMA empleando una muestra actualizada con la siguiente observación ($T - H + 1$) y quitando la primera observación (1), de tal manera que siempre se tiene una muestra del mismo tamaño ($T - H$) para actualizar los valores de los parámetros y pronosticar solo un periodo adelante con cada reestimación del modelo⁴. Este procedimiento se realiza hasta agotar la muestra de evaluación. Es decir, hasta tener H pronósticos, cada uno se realizó para un horizonte de un periodo adelante. A este procedimiento se le conoce como **pronósticos con una ventana móvil** o **rolling window** (ver Figura 10.3). Finalmente, otra opción es actualizar los coeficientes empleando una observación más después de hacer cada pronóstico para un periodo adelante y dejando el inicio de la muestra siempre igual, de tal manera que la muestra crece. Este procedimiento se conoce como **pronóstico recursivo** (ver Figura 10.4).

² Las evaluaciones in-sample no son recomendables si el modelo se empleará para hacer proyecciones, pues no es lo mismo tener un modelo que se comporte bien en la muestra que uno que sea capaz de comportarse por fuera de esta. Por eso es recomendable emplear las evaluaciones fuera de muestra para seleccionar modelos para pronosticar.

³ En inglés esto es lo que se denomina un pronóstico de **H-step-ahead**.

⁴ estos son pronósticos de un paso adelante (**one-step-ahead**).

Figura 10.2 Pronósticos fuera de muestra con ventana fija

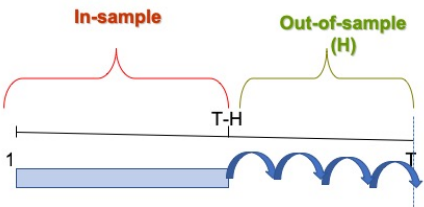


Figura 10.3 Pronósticos fuera de muestra con ventana móvil

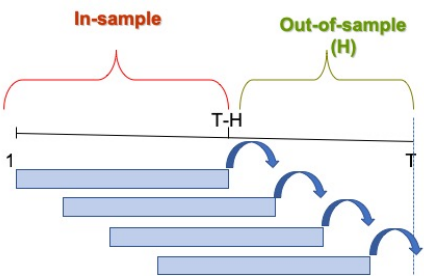
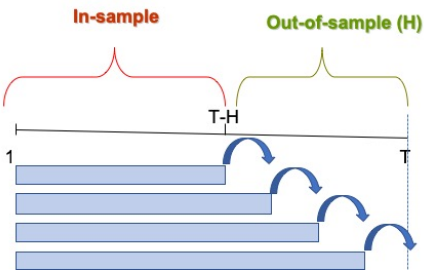


Figura 10.4 Pronósticos fuera de muestra con ventana recursiva



10.2. Métricas para la evaluación de pronósticos

Independientemente del método empleado para generar las predicciones fuera de muestra, existen unas métricas que permiten evaluar que tan cerca está cada

pronóstico del valor real observado para los I periodos. Las métricas más empleadas para evaluar el comportamiento de los pronósticos son:

- *RMSE* (Raíz media cuadrada del error):

$$RMSE = \sqrt{\frac{1}{h} \sum_{t=T-H}^T (\hat{y}_t - y_t)^2}$$

- *MAE* (error absoluto promedio):

$$MAE = \frac{1}{h} \sum_{t=T-H}^T |\hat{y}_t - y_t|$$

- *MAPE* (error absoluto porcentual promedio):

$$MAPE = \frac{1}{h} \sum_{t=T-H}^T \frac{|\hat{y}_t - y_t|}{\hat{y}_t}$$

donde \hat{y}_t representa el valor proyectado por el respectivo modelo para el periodo t .

Todos estas métricas tienen en común que implican tomar decisiones sin tener en cuenta ninguna distribución. Por otro lado, existe la posibilidad de realizar una prueba de hipótesis que permita determinar si estadísticamente la serie pronosticada es diferente del valor observado o no.

Diebold y Mariano, 1995 propusieron una prueba que compara la precisión de dos diferentes métodos de pronóstico. La hipótesis nula de esta prueba es que, al comparar dos métodos, ambos tengan la misma precisión estadística. Esta prueba parte de definir el error de pronóstico (forecast error) del método j para el periodo t como $e_{j,t} = \hat{y}_{j,t} - y_t$, para $j = 1, 2$. Adicionalmente, es necesario definir una función de pérdida (loss function en inglés) $g()$ que es la que penalizará el tamaño del error de cada uno de los modelos⁵.

Diebold y Mariano, 1995 definen el diferencial entre los dos pronósticos como $d_t = g(e_{1,t}) - g(e_{2,t})$. Intuitivamente es muy fácil argumentar que los dos pronósticos tienen la misma precisión si y solamente si el valor esperado de la diferencia es cero. Esto quiere decir que

$$H_0 : E[d_t] = 0$$

en otras palabras los dos pronósticos tienen la misma precisión y

$$H_A : E[d_t] \neq 0$$

Los dos pronósticos no tienen la misma precisión. Para realizar esta prueba, Harvey, Leybourne y Newbold, 1997 proponen una versión simplificada del estadístico original dada por:

⁵ Una función de pérdida se caracteriza por: *i.* $g(e_{j,t}) = 0$ si no hay error, *ii.* $g(e_{j,t}) \geq 0$ y *iii.* $g(e_{j,t})$ aumenta en valor cuando el error crece independiente del signo de este. Ejemplos de funciones de pérdida son $g(e_{j,t}) = |e_{j,t}|$ y $g(e_{j,t}) = |e_{j,t}|^2$

$$DM = \frac{\bar{d}}{\gamma_0 + \sum_{k=1}^{\infty} \gamma_k} \quad (10.1)$$

donde $\gamma_k = \text{Cov}[d_t, d_{t-j}]$ y $= 1 \over H \sum_{t=1}^H d_t$

Adicionalmente, q es la última covarianza diferente de cero. Los autores demuestran que $|DM|$ sigue una distribución estándar normal.

En la próxima sección discutiremos como calcular los pronósticos y evaluarlos fuera de muestra en R.

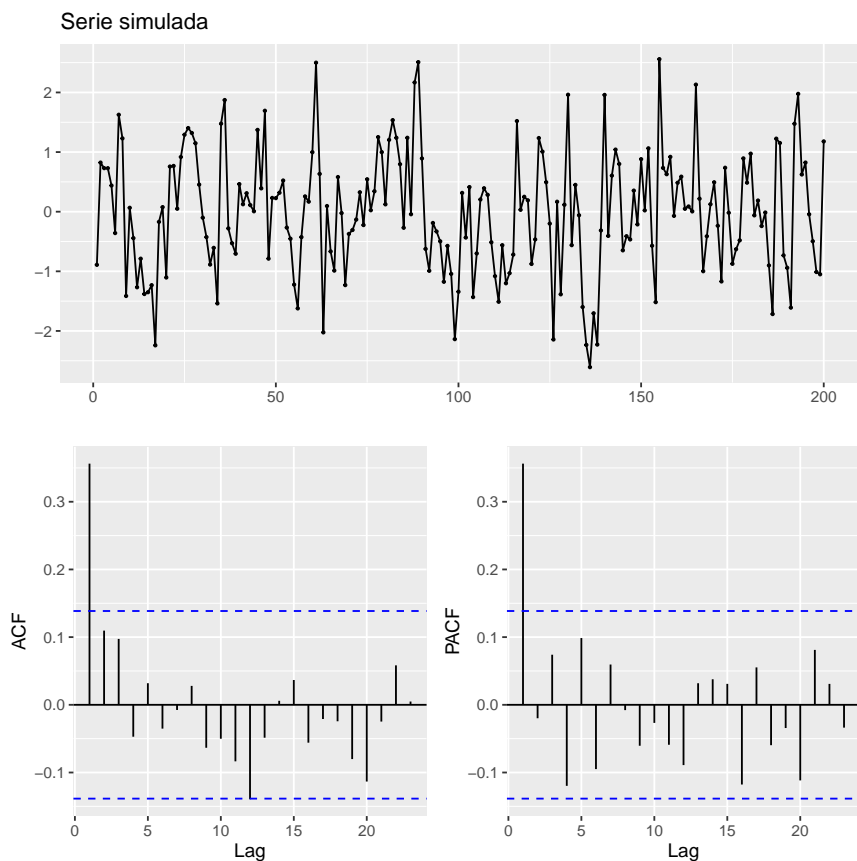
10.3. Ejemplo con datos simulados y modelos ARIMA

Para mostrar como calcular pronósticos y cómo evaluarlos, emplearemos una muestra de tamaño 200 ($T = 200$) simulada del siguiente DGP:

$$y_t = 0,45y_{t-1} + \varepsilon_t$$

La Figura 10.5 presenta la serie simulada y la respectivas SACF y APACF.

```
set.seed(123)
y <- arima.sim(model = list(order = c(1, 0, 0), ar = c(0.45)),
               n = 200)
library(forecast)
```

Figura 10.5 SACF y SPACF de la muestra simulada

Para realizar la evaluación de los modelos empleemos las últimas 50 observaciones. Es decir, $H = 50$ y los modelos serán estimados inicialmente con $T - H = 150$ observaciones. Ahora encontremos el mejor modelo de forma automática empleando los criterios de información.

```
auto.arima(y[1:150], max.p = 10, max.q = 10, ic = 'aic', stepwise=FALSE)

## Series: y[1:150]
## ARIMA(0,0,3) with zero mean
##
## Coefficients:
##      ma1      ma2      ma3
##      0.4499  0.1848  0.2610
## s.e.  0.0806  0.0923  0.0872
##
## sigma^2 estimated as 0.8515: log likelihood=-199.47
## AIC=406.94  AICc=407.22  BIC=418.99
```

```

auto.arima(y[1:150], max.p = 10, max.q = 10, ic = 'aicc', stepwise=FALSE)

## Series: y[1:150]
## ARIMA(0,0,3) with zero mean
##
## Coefficients:
##          ma1      ma2      ma3
##      0.4499  0.1848  0.2610
## s.e.  0.0806  0.0923  0.0872
##
## sigma^2 estimated as 0.8515:  log likelihood=-199.47
## AIC=406.94  AICc=407.22  BIC=418.99

auto.arima(y[1:150], max.p = 10, max.q = 10, ic = 'bic', stepwise=FALSE)

## Series: y[1:150]
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##          ar1
##      0.4246
## s.e.  0.0739
##
## sigma^2 estimated as 0.8768:  log likelihood=-202.58
## AIC=409.16  AICc=409.24  BIC=415.18

```

Los criterios AIC y AICC seleccionan el mismo modelo: ARIMA(0,0,3). El criterio BIC selecciona el modelo ARIMA(1,0,1). Recuerden que los datos fueron generados empleando el modelo ARIMA(1,0,0).

Trabajemos con estos dos modelos y veamos si los residuales siguen los supuestos.

```

Modelo.1 <- arima(y[1:150], order=c(0, 0, 3))
res.1 <- residuals (Modelo.1)
signo.error.1 <- factor(res.1>0)
library(tseries)
runs.test(signo.error.1)

##
## Runs Test
##
## data: signo.error.1
## Standard Normal = 0.34792, p-value = 0.7279
## alternative hypothesis: two.sided

Modelo.2 <- arima(y[1:150], order=c(1, 0, 0))
res.2 <- residuals (Modelo.2)
signo.error.2 <- factor(res.2>0)
runs.test(signo.error.2)

##
## Runs Test
##
## data: signo.error.2

```

```
## Standard Normal = -0.30854, p-value = 0.7577
## alternative hypothesis: two.sided
```

Cuadro 10.1 Prueba de Box-Pierce de los errores para el modelo ARIMA(0,0,3) para los primeros rezagos

Rezagos	Box-Pierce	p-valor
1.00	0.02	0.89
2.00	0.02	0.99
3.00	0.10	0.99
4.00	0.23	0.99
5.00	0.23	1.00
6.00	0.24	1.00
7.00	0.41	1.00
8.00	0.46	1.00
9.00	1.15	1.00
10.00	1.18	1.00
11.00	2.39	1.00
12.00	6.70	0.88
13.00	7.58	0.87
14.00	7.63	0.91
15.00	7.83	0.93
16.00	8.91	0.92
17.00	9.80	0.91
18.00	10.17	0.93
19.00	10.39	0.94
20.00	10.99	0.95

Para los dos modelos las pruebas muestran que los errores no están correlacionados. Ahora miremos el comportamiento de la varianza.

Cuadro 10.2 Prueba de Box-Pierce de los errores para el modelo ARIMA(1,0,0) para los primeros rezagos

Rezagos	Box-Pierce	p-valor
1.00	0.02	0.89
2.00	0.02	0.99
3.00	3.06	0.38
4.00	5.46	0.24
5.00	5.46	0.36
6.00	5.50	0.48
7.00	5.54	0.59
8.00	5.59	0.69
9.00	6.85	0.65
10.00	6.92	0.73
11.00	7.77	0.73
12.00	11.57	0.48
13.00	13.01	0.45
14.00	13.43	0.49
15.00	13.43	0.57
16.00	14.39	0.57
17.00	15.16	0.58
18.00	15.36	0.64
19.00	15.61	0.68
20.00	16.11	0.71

Figura 10.6 Gráficos para detectar comportamiento ARCH del modelo ARIMA(0,0,3)

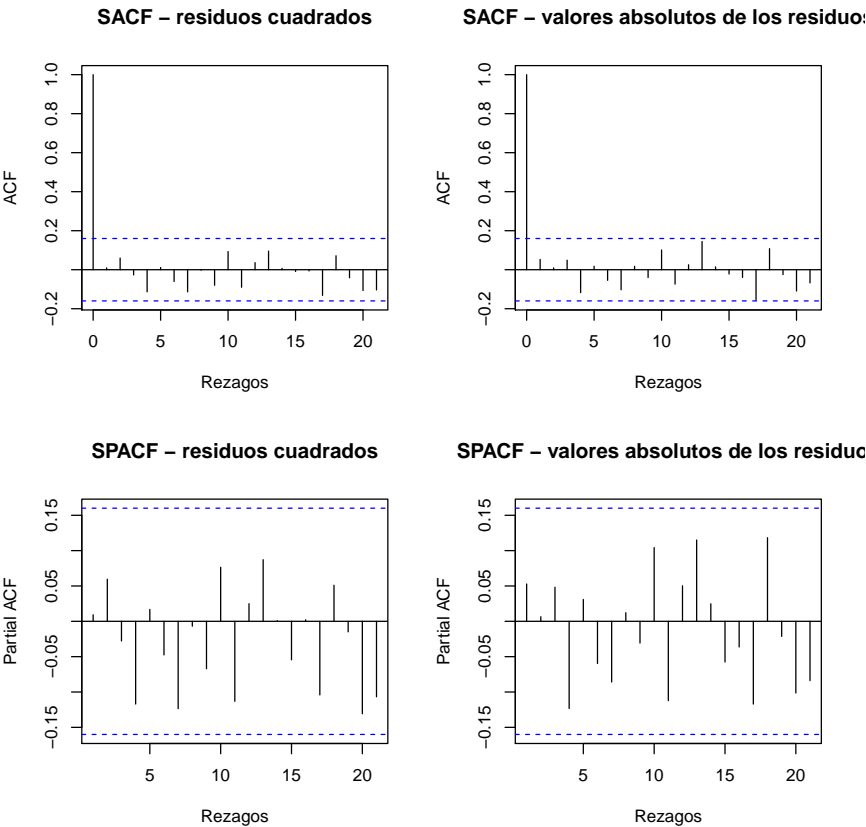
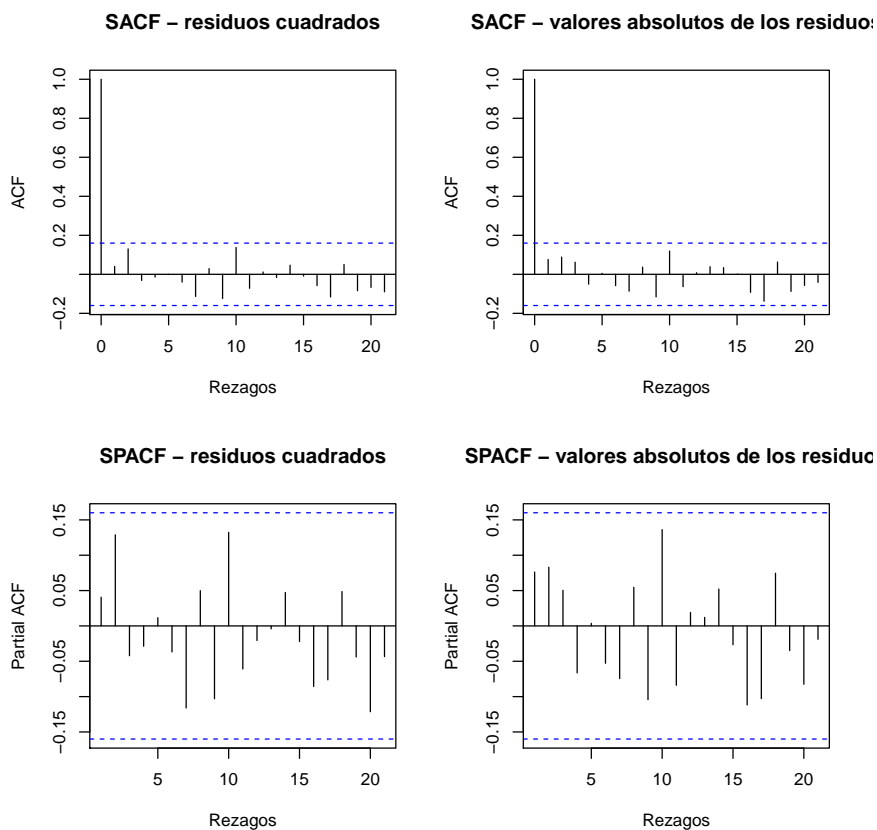


Figura 10.7 Gráficos para detectar comportamiento ARCH del modelo ARIMA(1,0,0)

Así los dos modelos no tienen problema de comportamiento ARCH. Ahora miremos la normalidad de los errores estimados.

```
##
##  Jarque Bera Test
##
## data:  res.1
## X-squared = 0.3927, df = 2, p-value = 0.8217
##
##  Shapiro-Wilk normality test
##
## data:  res.1
## W = 0.99476, p-value = 0.8689
##
##  Jarque Bera Test
##
```

```
## data: res.2
## X-squared = 0.7783, df = 2, p-value = 0.6776
##
## Shapiro-Wilk normality test
##
## data: res.2
## W = 0.99397, p-value = 0.7893
```

Para los dos modelos y las dos pruebas no es posible rechazar la nula de que los residuos siguen una distribución normal.

Ahora podemos proceder a construir los pronósticos con cada uno de los dos modelos para los 50 periodos de la muestra de evaluación. Esto se puede hacer con la función *forecast* del paquete **forecast** (Hyndman y Khandakar, 2008) del paquete del mismo nombre. Esta función típicamente incluye los siguientes argumentos:

$$\text{forecast}(\text{object}, h, \text{level} = c(80, 95))$$

donde:

- **object**: el objeto que contiene el modelo a partir del cual se quiere realizar las proyecciones.
- **h**: es el número de periodos que se quiere proyectar (estos son pronósticos de un periodo adelante).
- **level = c(80, 95)**: El nivel de confianza para los intervalos de confianza de la predicción. Por defecto se calculan los límites inferiores y superiores para el pronóstico para los niveles de 80 % y 95 %.

Es importante mencionar que esta función asume que el residuo sigue una distribución normal. En caso que los datos no sigan una distribución normal o que la muestra sea pequeña y se tema que el resultado asintótico no aplique, es posible emplear la técnica de *bootstrapping*. En este caso la distribución del residuo es simulada a partir de los residuos estimados y por tanto no implica ningún supuesto de distribución. Esto se puede realizar con el argumento **bootstrap** igualándolo a **TRUE**⁶ (Ver sección 9.4). El valor por defecto de este argumento es **FALSE**.

Antes de continuar, recordemos que tenemos tres formas de hacer este pronóstico, con ventana fija, ventana móvil y ventana recursiva.

10.3.1. Pronóstico con ventana fija

En este caso la ventana de estimación de los parámetros se mantiene fija para los primeros 150 datos y con ese modelo estimado realizamos los pronósticos para el siguiente periodo durante 50 periodos. Esto lo podemos hacer de la siguiente manera.

⁶ Es decir, **bootstrap = TRUE**.

```
library(forecast)
Modelo1.f.vf<- forecast(Modelo.1, h=50)
attributes(Modelo1.f.vf)

## $names
## [1] "method"      "model"      "level"      "mean"      "lower"
## [6] "upper"      "x"          "series"     "fitted"     "residuals"
##
## $class
## [1] "forecast"

Modelo2.f.vf<- forecast(Modelo.2, h=50)
```

El objeto en el que guardamos los resultados tiene muchos compartimientos "slots". Estos se pueden acceder empleando el signo \$. Por ejemplo si deseamos el pronóstico puntual para el primer modelo podemos emplear **Modelo.1.vf\$mean**. Si deseamos los límites inferiores de los intervalos de confianza, podemos invocar el compartimiento **lower** y así sucesivamente. Como solo emplearemos el pronóstico puntual para la evaluación de estos por fuera de muestra, recojamos dichos pronósticos en un vector.

```
Modelo.1.f.vf <- Modelo1.f.vf$mean
Modelo.2.f.vf <- Modelo2.f.vf$mean
```

Podemos graficar el objeto con los pronósticos para obtener la Figura 10.8 por medio del siguiente código.

```
plot(Modelo2.f.vf)
```


Forecasts from ARIMA(1,0,0) with non-zero mean

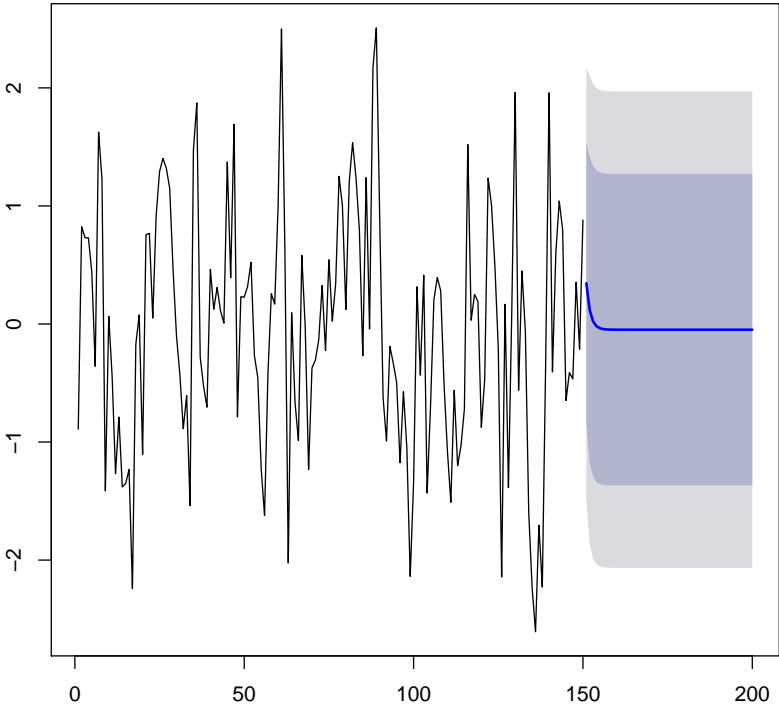
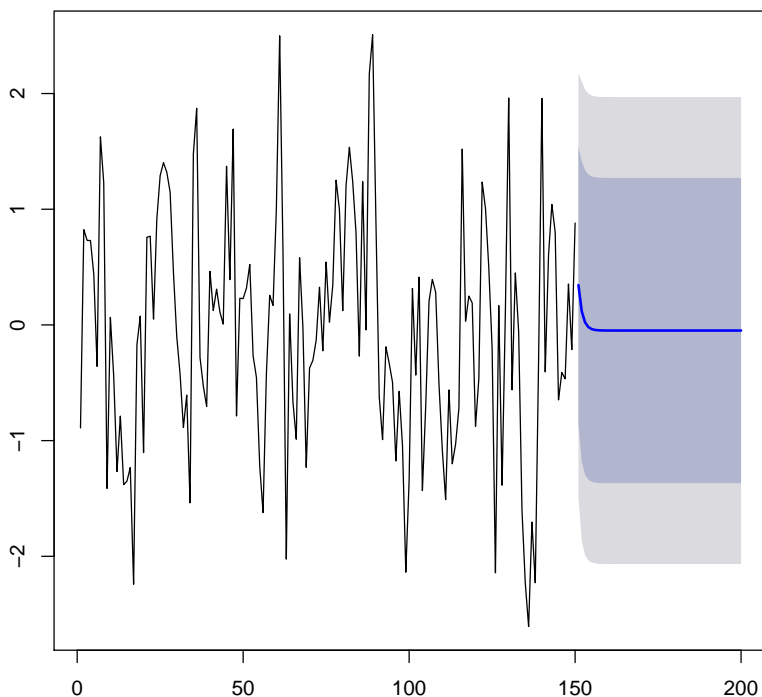


Figura 10.8 Gráficos de los pronósticos para el modelo ARIMA(1,0,0) para 50 periodos con una ventana fija



10.3.2. *Pronóstico con ventana móvil (rolling window)*

En este caso la ventana de estimación de los parámetros se mantiene del mismo tamaño, pero se mueve una observación a la vez hasta agotar toda la muestra. Esto lo podemos hacer empleando la función **window** del paquete central de R (R Core Team, 2019). Esta función obtiene un subconjunto de una serie de tiempo. Esta función típicamente incluye tres argumentos:

$$\text{window}(x, \text{start} = \text{NULL}, \text{end} = \text{NULL})$$

donde:

- **x**: un objeto de serie de tiempo al que se le quiere sacar un subconjunto.

- **start**: inicio de la ventana (de la submuestra).
- **end**: final de la ventana (de la submuestra).

Esta función se puede emplear de la siguiente manera:

```
Modelo.1.f.v.m <- NULL
Modelo.2.f.v.m <- NULL
for(i in 1:50){
  ventana<- window(y, start=1+i, end=150+i)
  Modelo.1.movil<- arima(ventana, order=c(0, 0, 3))
  Modelo.1.f.v.m[i] <- forecast(Modelo.1.movil, h=1)$mean

  Modelo.2.movil<- arima(ventana, order=c(1, 0, 0))
  Modelo.2.f.v.m[i] <- forecast(Modelo.2.movil, h=1)$mean
}
```

10.3.3. Pronóstico con ventana recursiva

En este caso la ventana de estimación de los parámetros se amplía una observación a la vez después de construir el pronóstico hasta agotar toda la muestra. Esto lo podemos hacer de una manera similar al caso de la ventana móvil.

```
Modelo.1.f.v.r <- NULL
Modelo.2.f.v.r <- NULL
for(i in 1:50){
  ventana<- window(y, start=1, end=150+i)
  Modelo.1.recursiva<- arima(ventana, order=c(0, 0, 3))
  Modelo.1.f.v.r[i] <- forecast(Modelo.1.recursiva, h=1)$mean

  Modelo.2.recursiva<- arima(ventana, order=c(1, 0, 0))
  Modelo.2.f.v.r[i] <- forecast(Modelo.2.recursiva, h=1)$mean
}
```

10.4. Evaluación de pronósticos

Las métricas de comparación de pronósticos los podemos calcular empleando la función **error** del paquete **ftsa** (Shang, 2013).

Esta función típicamente incluye tres argumentos:

error(forecast, true, , method = c("me", "mpe", "mae", "mse", "sse", "rmse", "mdae", "mdse", "mape", "mdape", "smape", "smdape", "rmspe", "rmdspe", "mrae", "mdrae", "gmrae", "relmae", "relmse", "mase", "mdase", "rmsse")

donde:

- **forecast**: objeto que contiene los valores pronosticados fuera de muestra.

- **true:** objeto que contiene los valores observados de la serie.
- **method:** lista que contiene una de las métricas a calcular.

Con las siguientes líneas de código podemos calcular para el primer modelo cada una de las métricas.

```
library(ftsa)

## Loading required package: rainbow
## Loading required package: MASS
## Loading required package: pcaPP
## Loading required package: sde
## Loading required package: stats4
## Loading required package: fda
## Loading required package: splines
## Loading required package: Matrix
##
## Attaching package: 'fda'
## The following object is masked from 'package:forecast':
##
##   fourier
## The following object is masked from 'package:graphics':
##
##   matplot
## sde 2.0.15
## Companion package to the book
## 'Simulation and Inference for Stochastic Differential Equations With R
## Examples'
## Iacus, Springer NY, (2008)
## To check the errata corrrige of the book, type vignette("sde.errata")
##
## Attaching package: 'ftsa'
## The following object is masked from 'package:magrittr':
##
##   extract
## The following objects are masked from 'package:stats':
##
##   sd, var

y.eval <- window(y, start=151)

# ventana fija
metricas.modelo1.vf.mae <- error(forecast = Modelo.1.f.vf, true =y.eval,
                                method = "mae")
metricas.modelo1.vf.mape <- error(forecast = Modelo.1.f.vf, true =y.eval,
                                method = "mape")
metricas.modelo1.vf.rmse <- error(forecast = Modelo.1.f.vf, true =y.eval,
                                method = "rmse")

# ventana móvil
metricas.modelo1.vm.mae <- error(forecast = Modelo.1.f.vm, true =y.eval,
                                method = "mae")
metricas.modelo1.vm.mape <- error(forecast = Modelo.1.f.vm, true =y.eval,
                                method = "mape")
metricas.modelo1.vm.rmse <- error(forecast = Modelo.1.f.vm, true =y.eval,
```

```
method = "rmse")
# ventana recursiva
metricas.modelo1.vr.mae <- error(forecast = Modelo.1.f.vr, true =y.eval,
                                method = "mae")
metricas.modelo1.vr.mape <- error(forecast = Modelo.1.f.vr, true =y.eval,
                                method = "mape")
metricas.modelo1.vr.rmse <- error(forecast = Modelo.1.f.vr, true =y.eval,
                                method = "rmse")
```

Este mismo procedimiento se puede realizar para el segundo modelo. Los resultados de los dos modelos se resumen en la siguiente tabla.

```
# ventana fija
metricas.modelo2.vf.mae <- error(forecast = Modelo.2.f.vf, true =y.eval,
                                method = "mae")
metricas.modelo2.vf.mape <- error(forecast = Modelo.2.f.vf, true =y.eval,
                                method = "mape")
metricas.modelo2.vf.rmse <- error(forecast = Modelo.2.f.vf, true =y.eval,
                                method = "rmse")

# ventana móvil
metricas.modelo2.vm.mae <- error(forecast = Modelo.2.f.vm, true =y.eval,
                                method = "mae")
metricas.modelo2.vm.mape <- error(forecast = Modelo.2.f.vm, true =y.eval,
                                method = "mape")
metricas.modelo2.vm.rmse <- error(forecast = Modelo.2.f.vm, true =y.eval,
                                method = "rmse")

# ventana recursiva
metricas.modelo2.vr.mae <- error(forecast = Modelo.2.f.vr, true =y.eval,
                                method = "mae")
metricas.modelo2.vr.mape <- error(forecast = Modelo.2.f.vr, true =y.eval,
                                method = "mape")
metricas.modelo2.vr.rmse <- error(forecast = Modelo.2.f.vr, true =y.eval,
                                method = "rmse")
```

Cuadro 10.3 Métricas para los pronósticos con ventana fija

	ARIMA(0,0,3)	ARMA(1,0,0)
MAE	0.77	0.76
MAPE	171.83	150.16
RMSE	0.97	0.96

Cuadro 10.4 Métricas para los pronósticos con ventana móvil

	ARIMA(0,0,3)	ARMA(1,0,0)
MAE	0.49	0.48
MAPE	143.69	71.21
RMSE	0.64	0.60

Cuadro 10.5 Métricas para los pronósticos con ventana móvil

	ARIMA(0,0,3)	ARMA(1,0,0)
MAE	0.50	0.47
MAPE	168.20	67.36
RMSE	0.65	0.59

Como se puede observar, con los tres criterios y en todas las formas de calcular los pronósticos podemos concluir que el modelo ARIMA(0,0,1) presenta un mejor comportamiento fuera de muestra.

Ahora consideremos la prueba de Diebold y Mariano (1995), la cual podemos realizar empleando la función **dm.test** del paquete **forecast** (Hyndman y Khandakar, 2008).

Esta función típicamente incluye cuatro argumentos:

dm.test(e1, e2, alternative = c("two.sided", "less", "greater"), power = 2)

donde:

- **e1**: objeto que contiene los errores de pronóstico del modelo 1.
- **e2**: objeto que contiene los errores de pronóstico del modelo 2.
- **alternative**: lista que determina que tipo de hipótesis alterna y nula se desea probar. Si **alternative = "two.sided"**, entonces la hipótesis nula será que los dos métodos tienen la misma precisión. Esta es la opción por defecto. Si **alternative = "less"**, entonces la hipótesis nula será que el método 2 es igual o mas preciso que el método 1. Si **alternative = "greater"**, entonces la hipótesis nula será que el método 2 es igual o menos preciso que el método 1.
- **power**: un numero igual a 1 o 2 que determina la potencia a la cual se eleva la función de perdida. Si **power = 1**, entonces la función de perdida será: $g(e_{j,t}) = |e_{j,t}|$. Si **power = 2**, entonces la función de perdida será: $g(e_{j,t}) = |e_{j,t}|^2$.

Así, podemos emplear el siguiente código para comparar los dos modelos fuera de la muestra empleando como función de perdida $g(e_{j,t}) = |e_{j,t}|$.

```
# ventana fija
e1.vf <- Modelo.1.f.vf - y.eval
e2.vf <- Modelo.2.f.vf - y.eval
dm.test(e1.vf, e2.vf, power=1)
dm.test(e1.vf, e2.vf, alternative="greater", power=1)
dm.test(e1.vf, e2.vf, alternative="less", power=1)
```

Con esta función de pérdida y para el caso de la ventana fija, no se puede rechazar la nula de que los dos modelos son igual de precisos. Pero, si es posible rechazar la hipótesis nula de que el método 2 es igual o menos preciso que el método 1. Es decir el modelo 2 es más preciso que el 1.

Para la ventana móvil y la recursiva tendremos que:

```
# ventana móvil
e1.vm <- Modelo.1.f.vm - y.eval
e2.vm <- Modelo.2.f.vm - y.eval
```

```

dm.test(e1.vm, e2.vm, power=1)
dm.test(e1.vm, e2.vm, alternative="greater", power=1)
dm.test(e1.vm, e2.vm, alternative="less", power=1)

# ventana recursiva
e1.vr <- Modelo.1.f.vr - y.eval
e2.vr <- Modelo.2.f.vr - y.eval
dm.test(e1.vr, e2.vr, power=1)
dm.test(e1.vr, e2.vr, alternative="greater", power=1)
dm.test(e1.vr, e2.vr, alternative="less", power=1)

```

Tanto para la venta recursiva como la móvil, se puede rechazar la nula de que los dos modelos tengan la misma precisión. No obstante, no es posible concluir con claridad cuál de los dos modelos es mejor.

Poniendo todos los resultados juntos, encontramos que con las métricas, sin importar el método, es mejor el modelo 2. Por otro lado, para la ventana fija es mejor el modelo 2. Y para los otros dos métodos no es posible determinar que modelo es mejor, no obstante sabemos que no son igual de precisos los dos métodos. Por lo tanto tendremos que concluir que el modelo 2 es mejor. Aunque la decisión no es tan clara desde el punto estadístico, las métricas si favorecen este modelo. Así para pronosticar “nuevos períodos” podremos emplear el modelo ARMA(1,0) estimado con toda la muestra (los T datos). Por tanto nuestro pronóstico para el siguiente periodo es:

```
f<-forecast(arima(y, order=c(1, 0, 0)) , h=1)
```

```
plot(f)
```

Forecasts from ARIMA(1,0,0) with non-zero mean

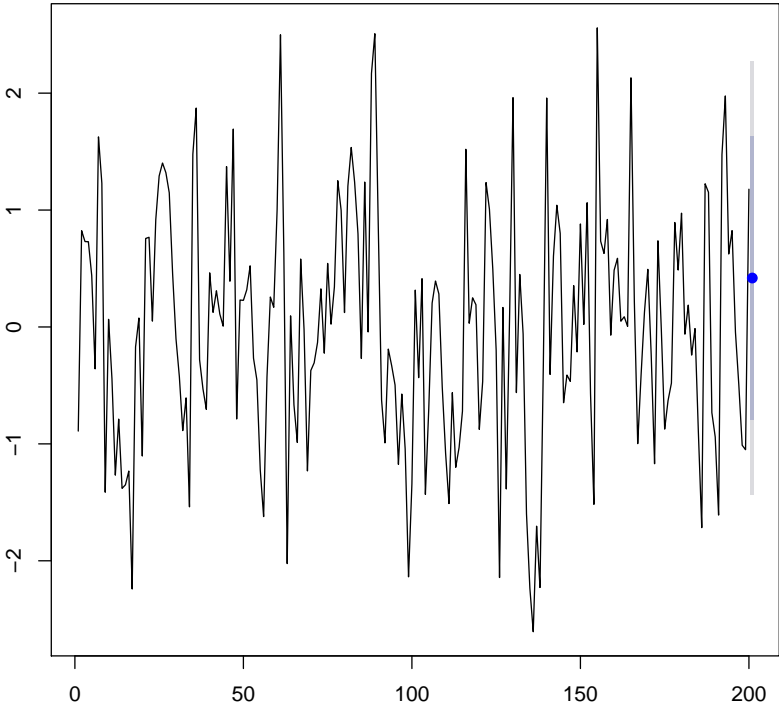
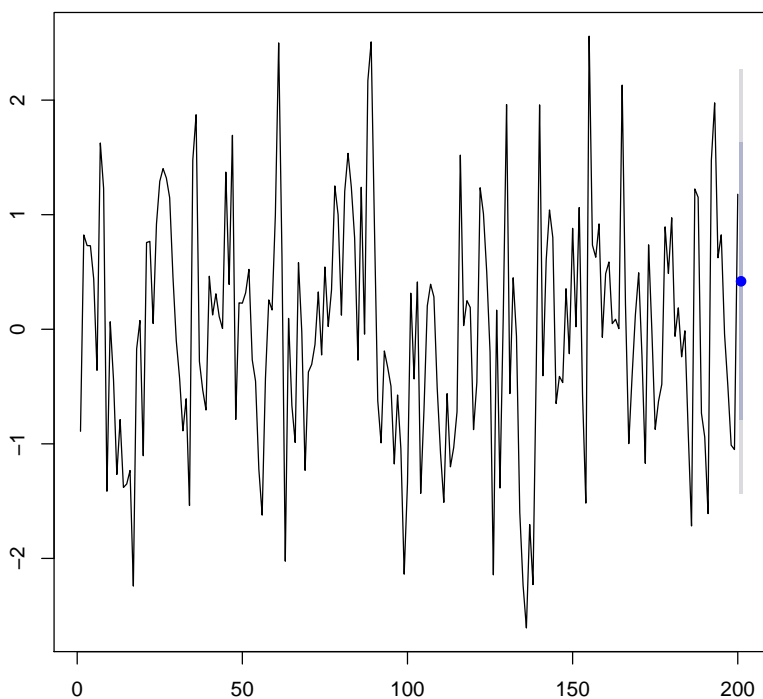


Figura 10.9 Pronóstico de la serie ARMA(1,0) para el siguiente periodo

10.5. Comentarios finales

En este capítulo hemos visto como construir pronósticos y sus correspondientes intervalos de confianza. Es importante recordar siempre la necesidad de validar los supuestos de los modelos automáticos, pues los algoritmos simplemente generan una forma rápida de comprar modelos basados en una métrica, pero no están diseñados para encontrar los modelos que necesariamente cumplan los supuestos.

Ejercicios

1. En el archivo *DataCDTs.xlsx* se encuentra datos para la DTF a 90 días en Colombia. Cargue los datos y genere una predicción y su respectivo intervalo de confianza para el siguiente día.
2. En el archivo *IPC.csv* se encuentra datos mensuales para IPC total nacional en Colombia desde enero de 2009. Cargue los datos y genere una predicción y su respectivo intervalo de confianza para el siguiente mes.

Capítulo 11

Combinación de pronósticos

Típicamente el objetivo de un científico de datos es encontrar el mejor modelo par pronosticar. Hasta aquí hemos visto diferente métodos para estimar una serie de tiempo. Hemos estudiado métodos que enfatizan en el comportamiento de la tendencia, otros en la estacionalidad y otros en las autocorrelaciones. Una buena forma de mejorar nuestros pronósticos es combinarlos o ensamblarlos. Tal vez de los primeros autores en documentar las bondades de combinar pronósticos fue Bates y Granger (1969).

la idea de ensamblar pronósticos es tratar de sacar ventaja de las bondades de cada uno de los métodos empleados. Existen diferentes formas de combinar modelos que van desde sacar promedios ponderados de diferentes métodos de pronósticos hasta métodos que ensamblan pronósticos empleando regresiones. En este capítulo discutiremos diferentes aproximaciones que están disponibles en el paquete **ForecastComb** (Weiss, Roetzer y Raviv, 2018b) Siguiendo a Weiss, Raviv y Roetzer (2018a) dividiremos esta discusión en métodos sencillos para ensamblar pronósticos, métodos que emplean regresiones y métodos que emplean vectores propios.

Antes de iniciar la discusión de los métodos, necesitaremos emplear una notación algo diferente en este capítulo para evitar confusiones. Definamos $f_{i,t}$ como el pronóstico (en muestra o fuera de muestra) del método i para el periodo t , f_t^c como el pronóstico combinado, H el número de período pronosticados y N como el número total de métodos de pronósticos que serán combinados. Ahora veamos diferentes técnicas para crear dichas combinaciones, pero antes es importante anotar que será imposible determinar a-priori cuál es la mejor forma de combinar los pronósticos. Por eso seguiremos un procedimiento análogo al discutido hasta ahora que implica, si es del caso encontrar los parámetros requeridos en la muestra de estimación y escoger el mejor método de ensamble con la muestra de evaluación.

11.1. Métodos simples de combinación de pronósticos

11.1.1. Promedio simple

Esa es tal vez la aproximación más sencilla para combinar predicciones. No necesita la estimación de ningún parámetro. En este caso el pronóstico ensamblando f_t^c está dado por:

$$f_t^c = \frac{1}{N} \sum_{i=1}^N f_{i,t}$$

(11.1)

Asombrosamente, este método que pesa a todos los pronósticos por igual funciona muy bien. Esta combinación se puede implementar con la función *comb_SA* del paquete **ForecastComb**.

11.1.2. Mediana

En algunas ocasiones, algunos pronósticos pueden estar relativamente alejados de los otros métodos y del valor observado, en estos casos emplear un método que no sea sensible a valores extremos podría ser mejor idea. En este caso el pronóstico ensamblado f_t^c para cada periodo corresponde a la mediana. Es decir,

$$f_t^c = \begin{cases} f_{(\frac{N}{2}+0.5),t} & \text{N es impar} \\ \frac{1}{2} (f_{(\frac{N}{2}),t} + f_{(\frac{N}{2}+1),t}) & \text{N es par} \end{cases} \quad (11.2)$$

donde $f_{(i),t}$ corresponde al pronóstico en la posición i -ésima después de organizar de mayor a menor todos los N pronósticos del periodo t . Este tipo de ensamble se puede realizar por medio de la función *comb_MED* del paquete **ForecastComb**.

11.1.3. Media truncada

Otra manera de evitar que los valores extremos afecten el pronóstico combinado es descartar aquellos pronósticos que son los más grandes y más bajos. Por ejemplo, podemos quitar el 5% de los pronósticos más bajos y el 5% más grande. Sea λ la proporción de datos que serán eliminados en cada uno de los extremos (en la parte superior y en la parte inferior por igual), entonces el pronóstico ensamblado será:

$$f_t^c = \frac{1}{N(1-2\lambda)} \sum_{i=\lambda N+1}^{(1-\lambda)N} f_{i,t} \quad (11.3)$$

Note que si $\lambda = 0$ entonces tenemos el método de la media y si $\lambda = 0,5$ tenemos el método de la mediana. La función *comb_TA* del paquete **ForecastComb**. permite hacer este ensamble.

11.1.4. Media Winzor

Esta es una versión de media que también está diseñada para evitar que valor extremos influyan la media, pero a diferencia de la media truncada que elimina observaciones, esta lo que hace es ponderar diferente a dichos valores. En este caso tenemos:

$$f_t^c = \frac{1}{N} \left[K f_{(K+1),t} + \sum_{i=K+1}^{N-K} K f_{(N-K),t} \right] \quad (11.4)$$

donde $K = \lambda N$. La función *comb_WA* del paquete **ForecastComb** permite hacer esta combinación.

11.1.5. Método de Bates y Granger

Bates y Granger (1969) proponen una aproximación que viene de las finanzas y en especial de las técnicas de diversificación de los portafolios. La idea es emplear la matriz de media cuadrada de los errores de predicción para ponderar cada pronóstico. Es decir,

$$f_t^c = \sum_{i=1}^N f_{i,t} \frac{\hat{\sigma}_{(i),t}^2}{\sum_{j=1}^N \hat{\sigma}_{(j),t}^2} \quad (11.5)$$

donde $\hat{\sigma}_{(i),t}^2$ es el valor estimado de la media cuadrada de los errores de predicción para ponderar el método i en el periodo t . La función *comb_BG* del paquete **ForecastComb** permite hacer este ensamble de pronósticos.

11.1.6. Método de Newbold y Granger

Newbold y Granger (1974) siguen por la misma línea de Bates y Granger (1969), pero proponen ponderar los pronósticos con una minimización restringida de la media cuadrada de los errores de pronóstico. En este caso, el pronóstico ensamblado se puede expresar más fácilmente empleando álgebra matricial. Es decir,

$$f_t^c = \mathbf{F}_t \times \frac{\Sigma^{-1}e}{e^T \Sigma^{-1}e} \quad (11.6)$$

donde \mathbf{F}_t es una matriz de dimensiones $H \times N$ que contiene en cada columna los pronósticos de cada uno de los i métodos para los H periodos. Σ es la matriz de la media cuadrada de los errores de pronóstico. y finalmente, e es un vector de dimensiones $N \times 1$ que cumple la condición de que $e^T \times w = 1$. De manera análoga a las otras funciones de este capítulo, existe la función *comb_NG* del paquete **ForecastComb** permite hacer este ensamble de pronósticos.

11.1.7. Ranking inverso

Esta aproximación fue propuesta por Aiolfi y Timmermann (2006) implica ordenar los métodos de muestreo de acuerdo con el comportamiento en muestra (hasta la muestra de estimación), siendo la del primer puesto la del menor *MSE* (in-sample) ($Rank_1$), la N -ésima posición corresponde al método de pronóstico con el *MSE* mas alto ($Rank_N$). El pronóstico combinado le asignará mas peso a los métodos con mejor posición en el ranking¹. En este caso, el pronóstico ensamblado será:

$$f_t^c = \sum_{i=1}^N f_{i,t} \frac{Rank_i^{-1}}{\sum_{j=1}^N Rank_j^{-1}} \quad (11.7)$$

Esta aproximación se puede implementar empleando la función *comb_InvW* del paquete **ForecastComb**.

11.2. Métodos basados en regresión

11.2.1. Mínimos Cuadrados Ordinarios (MCO)

Crane y Crotty (1967) propuso inicialmente emplear un modelo de regresión como el siguiente para encontrar la relación (y por tanto los pesos a asignar) entre cada uno de los pronósticos de cada uno de los métodos y el valor realmente observado (y_t):

$$y_t = \beta_0 + \sum_{i=1}^N \beta_i f_{i,t} + \varepsilon_t \quad (11.8)$$

donde ε_t es un término de error que cumple todos los supuestos del Teorema de Gauss-Makow. Los coeficientes β s pueden ser estimados por medio de MCO (OLS en inglés) de tal manera que el pronóstico combinado estará dado por:

¹ Recuerden que mejor posición en el ranking implica un menor número, de ahí el nombre del método: ranking inverso.

$$f_t^c = \hat{\beta}_0 + \sum_{i=1}^N \hat{\beta}_i f_{i,t} \quad (11.9)$$

Noten que los β s estimados ($\hat{\beta}$ s) asociados a pendientes actúan como pesos muy similar a los promedio móviles que empleamos en las técnicas de combinación de pronósticos discutidos en la sección anterior. Es decir, si definimos $\hat{\omega}_i = \hat{\beta}_i$, es más fácil ver como este método termina siendo un promedio ponderado, pero los pesos son producto de emplear el método MCO. Es decir,

$$f_t^c = \hat{\omega}_0 + \sum_{i=1}^N \hat{\omega}_i f_{i,t} \quad (11.10)$$

Una de las ventajas de emplear este método es que los pesos serán insesgados y eficientes (si los supuestos del error se cumplen), y por lo tanto f_t^c será insesgado, aun si uno de los métodos de pronósticos presenta sesgo. Por otro lado, una desventaja del método es que las ponderaciones no necesariamente sumaran uno y por tanto la interpretación de dichos pesos no es tan sencilla. Podemos realizar esta combinación de pronósticos empleando la función *comb.OLS* del paquete **ForecastComb**.

11.2.2. *Mínima Desviación Absoluta (MDA)*

Un método alternativo para encontrar los pesos asociados a la expresión 11.8 es en vez de minimizar la suma de todos los errores al cuadrado, minimizar la suma de los valores absolutos de los errores. A este último método se le denomina *Mínima Desviación Absoluta (MDA)* o *LAD* por su sigla en inglés. En este caso, los errores grandes son igualmente penalizados que los errores pequeños a diferencia del método MCO que penaliza a los errores de pronóstico grandes. En algunas aplicaciones, dependiendo del dominio de aplicación, puede ser buena idea penalizar igual errores grandes que pequeños. El paquete **ForecastComb** permite implementar esta aproximación de combinación de pronósticos empleando la función *comb.LAD*.

11.2.3. *Mínimos Cuadrados Restringidos (MCR)*

Para evitar el problema de que los pesos no sumen uno, el método de MCO puede incluir la restricción de que $\sum_{i=1}^N \omega_i = 1$. La versión de los MCO que permite imponer restricciones a los parámetros se conoce como *Mínimos Cuadrados Restringidos (MCR o CLS por su sigla en inglés)*. El paquete **ForecastComb** implementa esta aproximación por medio de la función *comb.CLS*.

11.2.4. Promedio de todas las posibles regresiones

Es aproximación parte de reconocer que si empleamos N pronósticos se podrían tener

$$\sum_{i=1}^N \binom{N}{i} = 2^N - 1$$

modelos. Por ejemplo, si se tienen 5 métodos de pronóstico, entonces podrían existir 31 posibles modelos que estarán anidados en el equivalente al modelo 11.8 para 5 regresores. es decir, existirían 31 posibles ensambles de pronósticos. En otras palabras, nada garantiza que los 5 métodos tengan que hacer parte del pronóstico combinado y en principio estaríamos en un problema de seleccionar el mejor modelo.

Elliott, Gargano y Timmermann (2013) proponen emplear un promedio simple de todos los posibles ensambles. El paquete **ForecastComb** implementa esta idea realizando una modificación. La función *comb_CSR* no calcula el ensamble final empleando el promedio simple. Por el contrario construye pesos para cada uno de los pronósticos que emplean el valor del criterios de información para cada posible regresión. Los criterios de información disponibles son el AIC (criterio de información de Akaike), AICC (AIC modificado), BIC (Criterio Bayesiano de Información o criterio de Schwarz) y HQ (Hannan Quinn). Para cada uno de estos criterios se reporta los respectivos ensambles.

11.3. Métodos basados en vectores propios

Estas aproximaciones siguen la idea propuesta por Hsiao y Wan (2014) para la construcción de ensambles óptimos de pronósticos. Los autores parten de proponer ensambles que permitan minimizar el promedio del error cuadrático de predicción (MSPE)². Ellos proponen emplear una normalización de los pesos que en la práctica implica una restricción para los pesos en ese problema de minimización. La normalización propuesta corresponde a:

$$\sum_{i=1}^N \omega_i^2 = 1$$

Los autores proponen 4 aproximaciones que emplean valores propios para encontrar los pesos óptimos dada la anterior restricción que se encuentran implementadas en el paquete **ForecastComb**. Las aproximaciones son³:

- estándar (función *comb_EIG1*)
- con corrección de sesgo (función *comb_EIG2*)
- truncada (función *comb_EIG43*)

² Esto corresponde al MSE fuera de muestra que hemos discutido hasta ahora

³ Para ver el detalle de estas aproximaciones ver Hsiao y Wan (2014)

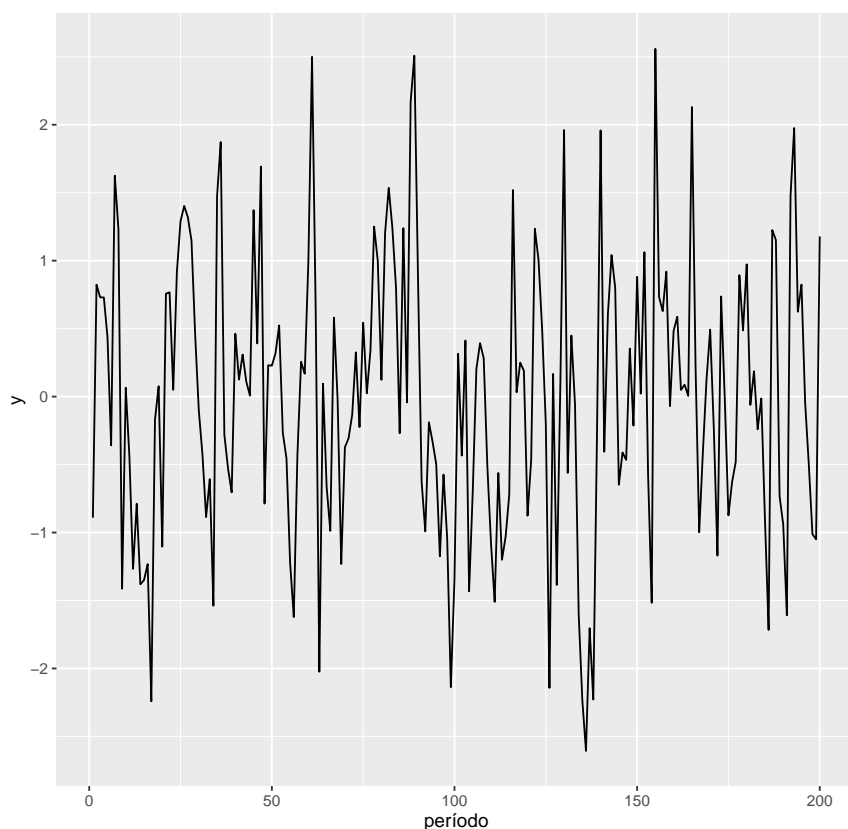
- truncada y con corrección de sesgo (función *comb.EIG4*)

11.4. Combinando pronósticos en R

En R podemos encontrar varios paquetes para ensamblar pronósticos como **ForecastComb** (Weiss y col., 2018b) , **forecastHybrid** (Shaub y Ellis, 2019) y **opera** (Gaillard y Goude, 2016) . En esta sección, continuando con lo discutido en la anterior nos concentraremos en emplear el paquete **ForecastComb**.

Para mostrar como emplear este paquete, empleemos la misma serie simulada que se emplea en el capítulo 10.

```
set.seed (123)
y <- arima.sim(model = list(order = c(1, 0, 0), ar = c(0.45)),
               n = 200)
```

Figura 11.1 Muestra simulada

Ahora supongamos que se desean construir pronósticos para un periodo y para realizar la evaluación de los modelos empleemos las últimas 50 observaciones. Es decir, $H = 50$ y los modelos serán estimados inicialmente con $T - H = 150$ observaciones. Todas las evaluaciones se realizarán con ventanas recursivas. Además, por simplicidad supongamos que se tiene los siguientes métodos de proyección:

1. Promedio móvil con ventana de 7 (compruebe que esta es la mejor ventana para este método)
2. SES
3. ARMA(1,0) (en el Capítulo 10 se concluyó que este era el mejor modelo ARIMA con residuales que son ruido blanco)

Nuestra tarea en lo que resta de este capítulo será generar combinaciones de pronósticos de esos métodos.

Iniciemos creando i) las muestra de evaluación y de estimación, ii) los respectivos 150 pronósticos en muestra y iii) 50 pronósticos fuera de muestra para el siguiente periodo empleando una ventana recursiva.

```
# i) Muestra de evaluación y de estimación
y.eval <- window(y, start= 151)
y.est <- window(y, end= 150)
library(forecast)

# ii) 150 pronósticos en muestra para cada modelo

#pronósticos de ventana recursiva
Th <- 150
f.ma.7 <- NULL
f.ses <- NULL
f.ARIMA <- NULL

ventana<- subset(y, end=150)

f.ma.7.enMuestra <- ma(ventana , order = 7)
f.ma.7.enMuestra[148:150] <-forecast(ma(ventana , order = 7), h=3)$mean
f.ses.enMuestra <- ses(ventana)$fitted
f.ARIMA.enMuestra <- Arima(ventana , order = c(1, 0, 0))$fitted

f.enMuestra <- cbind(f.ma.7.enMuestra, f.ses.enMuestra, f.ARIMA.enMuestra)

# iii) 50 pronósticos fuera de muestra para el siguiente periodo empleando una ventana recursiva
for(i in 1:50){
  ventana<- subset(y, start=1, end= Th +i -1)

  f.ma.7[i] <- tail(forecast(ma(ventana , order = 7), h=4)$mean,1)
  f.ses[i] <- tail(ses(ventana, h =2)$mean,1)
  f.ARIMA[i] <- forecast(Arima(ventana , order = c(1, 0, 0)), h=1)$mean
}

f.fueraMuestra <- cbind(f.ma.7, f.ses, f.ARIMA)
```

Antes de entrar en el detalle de la combinación de los pronósticos, construyamos una tabla con las métricas del desempeño de los modelos fuera de muestra. Esto nos permitirá comparar los resultados con los modelos de ensamble.

```
tabla <- rbind(accuracy(f.ma.7, y.eval)["Test set", c("RMSE", "MAPE", "MAE")],
accuracy(f.ses, y.eval)["Test set", c("RMSE", "MAPE", "MAE")],
accuracy(f.ARIMA, y.eval)["Test set", c("RMSE", "MAPE", "MAE")])

tabla <- data.frame(tabla)

row.names(tabla) <- c("MA", "SES", "ARIMA")

tabla
```

##		RMSE	MAPE	MAE
##	MA	1.1023316	350.1312	0.8789579
##	SES	1.0622242	318.4533	0.8263543
##	ARIMA	0.9886362	212.9988	0.7586818

11.4.1. Primer paso: preparación de los datos

El primer paso para realizar los ensambles es preparar los datos que le pasaremos a las funciones de combinación de pronósticos del paquete **ForecastComb**, dado que estas necesitan como entrada un objeto de clase **ForecastComb**. Para transformar datos a esta clase podemos emplear la función **foreccomb**. Esta función típicamente incluye los siguientes argumentos:

foreccomb(observed_vector, prediction_matrix, newobs = NULL, newpreds = NULL)
donde:

- **observed_vector**: un vector con la serie original para la muestra de estimación.
Para nuestro caso, construyamos el objeto **datos.entrada** de la siguiente manera:

```
library(ForecastComb)
datos.entrada <- foreccomb(observed_vector = y.est,
                           prediction_matrix = f.enMuestra, newobs = y.eval,
                           newpreds = f.fueraMuestra)

## A subset of the individual forecasts included NA values and has been imputed.
class(datos.entrada)

## [1] "foreccomb"
```

Noten que por defecto esta función imputa todos los valores perdidos que existan en la matriz de predicciones. Ahora podemos emplear cualquiera de los 15 métodos de ensamble disponibles.

11.4.2. Segundo paso: Construcción de los pronósticos combinados

El paquete **textbfForecastComb** cuenta con los 15 métodos de ensamble discutidos en la sección anterior. Las funciones que implementan dichos métodos tienen la misma estructura y requieren únicamente un argumento, un objeto de clase **textbfForecastComb**. Así, aplicar cualquiera de las 15 técnicas es muy sencillo.

Cuadro 11.1 Funciones del paquete `textbfForecastComb` para los 15 métodos estudiantos

Método	función
Métodos simples	
Promedio simple	<code>comb_SA</code>
Mediana	<code>comb_MED</code>
Media troncada	<code>comb_TA</code>
Media Winzor	<code>comb_WA</code>
Método de Bates y Granger	<code>comb_BG</code>
Método de Newbold y Granger	<code>comb_NG</code>
Ranking inverso	<code>comb_InvW</code>
Métodos basados en regresión	
Mínimos Cuadrados Ordinarios (MCO)	<code>comb_OLS</code>
Mínima Desviación Absoluta	<code>comb_LAD</code>
Mínimos Cuadrados Restringidos	<code>comb_CLS</code>
Promedio de todas las posibles regresiones	<code>comb_CSR</code>
Métodos basados en vectores propios	
Estándar	<code>comb_EIG1</code>
Con corrección de sesgo	<code>comb_EIG2</code>
Troncada	<code>comb_EIG3</code>
Troncada y con corrección de sesgo	<code>comb_EIG4 height</code>

Por ejemplo si se desea emplear el método de MCO para combinar los pronósticos podemos emplear el siguiente código

```
f.comb.OLS <- comb_OLS(datos.entrada)
summary(f.comb.OLS)

##
## Summary of Forecast Combination
## -----
##
## Method: Ordinary Least Squares Regression
##
## Individual Forecasts & Combination Weights:
##
##              Combination Weight
## f.ma.7.enMuestra      1.4130736
## f.ses.enMuestra      -0.6958283
## f.ARIMA.enMuestra     0.7634489
##
## Intercept (Bias-Correction): 0.03260275
##
## Accuracy of Combined Forecast:
##
##              ME      RMSE      MAE      MPE
## Training Set 1.804582e-17 0.7385989 0.5892611 -58.76327
## Test set    -5.184065e-02 1.1451885 0.9162992 47.87985
##              MAPE      NA      NA
## Training Set 250.2116 1.804582e-17 0.7385989
## Test set    347.5390 -7.048328e-02 1.0058895
##
## Additional information can be extracted from the combination object:
## For fitted values (training set): f.comb.OLS$Fitted
```

```
## For forecasts (test set): f.comb.OLS$Forecasts_Test
## See str(f.comb.OLS) for full list.
```

En los objetos creados por las funciones de este paquete podemos encontrar siempre una posición con los pronósticos ensamblados en la muestra que corresponde al compartimiento `$Fitted`. Los pronósticos de la muestra de evaluación están en el compartimiento `$Forecasts_Test`. Las métricas de precisión de los pronósticos para la muestra de estimación y evaluación se encuentran en `$Accuracy_Train` y `$Accuracy_Test`, respectivamente. En este caso

```
f.comb.OLS$Accuracy_Train

##              ME      RMSE      MAE      MPE      MAPE
## Test set 1.804582e-17 0.7385989 0.5892611 -58.76327 250.2116
##              ACF1 Theil's U
## Test set -0.05831707 0.5906766

f.comb.OLS$Accuracy_Test

##              ME      RMSE      MAE      MPE      MAPE
## Test set -0.05184065 1.145189 0.9162992 47.87985 347.539
##              ACF1 Theil's U
## Test set -0.07048328 1.00589
```

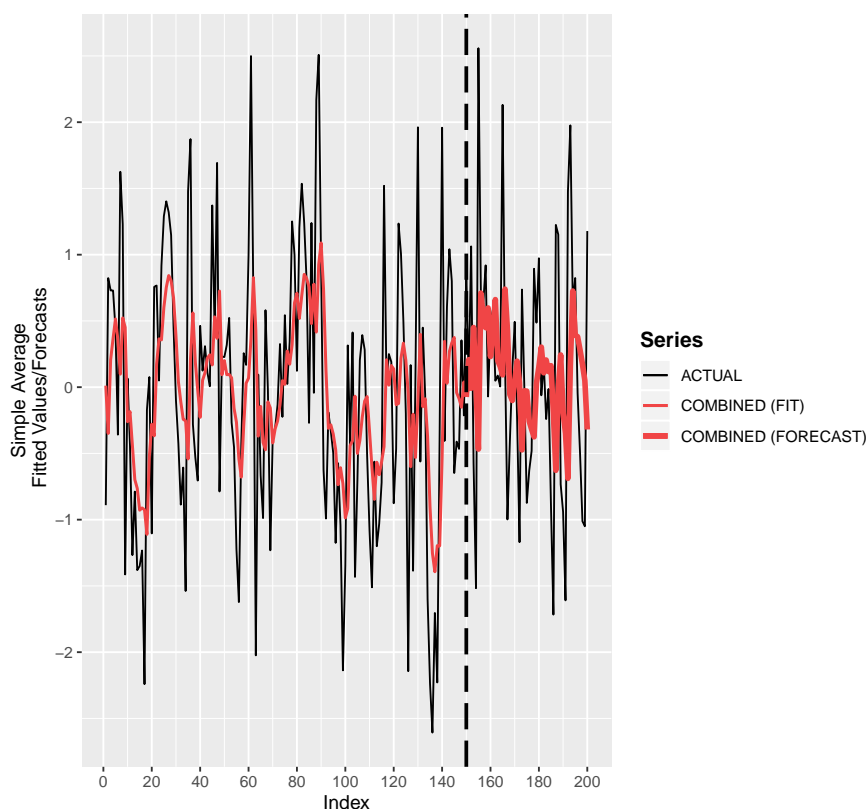
Miremos ahora un ejemplo con el método de media móvil.

```
f.comb.SA <- comb_SA(datos.entrada)
f.comb.SA$Accuracy_Test

##              ME      RMSE      MAE      MPE      MAPE
## Test set 0.01725634 1.036378 0.8054689 40.47328 257.8958
##              ACF1 Theil's U
## Test set -0.03269751 0.9670918

plot(f.comb.SA)
```

Simple Average Forecast Combination Actual vs. Fitted/Test Set Forecast



Noten e el método de media permite obtener mejores resultados fuera de muestra. Por otro lado, en este paquete existe la función **auto.combine**, que permite comparar todos los 15 métodos y seleccionar el mejor modelo basado en el RMSE de la muestra de estimación. Adicionalmente, se puede realizar una combinación que emplee la idea de crear ventanas repulsivas. Para esto se puede emplear la función **textbfrolling.combine** que tiene dos atributos, los datos y el método de combinación (este segundo se debe expresar entre comillas) Aquí es importante anotar que esa búsqueda con ventana móvil (y recursiva) implica agotar toda la muestra de estimación y en este caso sería conveniente incluir todas las observaciones como datos de entrada tanto para estimación como para evaluación. Veamos un ejemplo.

```
f.ma.7.TODAMuestra <- ma(y , order = 7)

f.ses.TODAMuestra <- ses(y)$fitted
f.ARIMA.TODAMuestra <- Arima(y , order = c(1, 0, 0))$fitted

f.TODAMuestra <- cbind(f.ma.7.TODAMuestra, f.ses.TODAMuestra, f.ARIMA.TODAMuestra)
```

```

datos.entrada.2 <- foreccomb(observed_vector = y,
                             prediction_matrix = f.TODAMuestra, newobs = y,
                             newpreds = f.TODAMuestra)

## A subset of the individual forecasts included NA values and has been imputed.

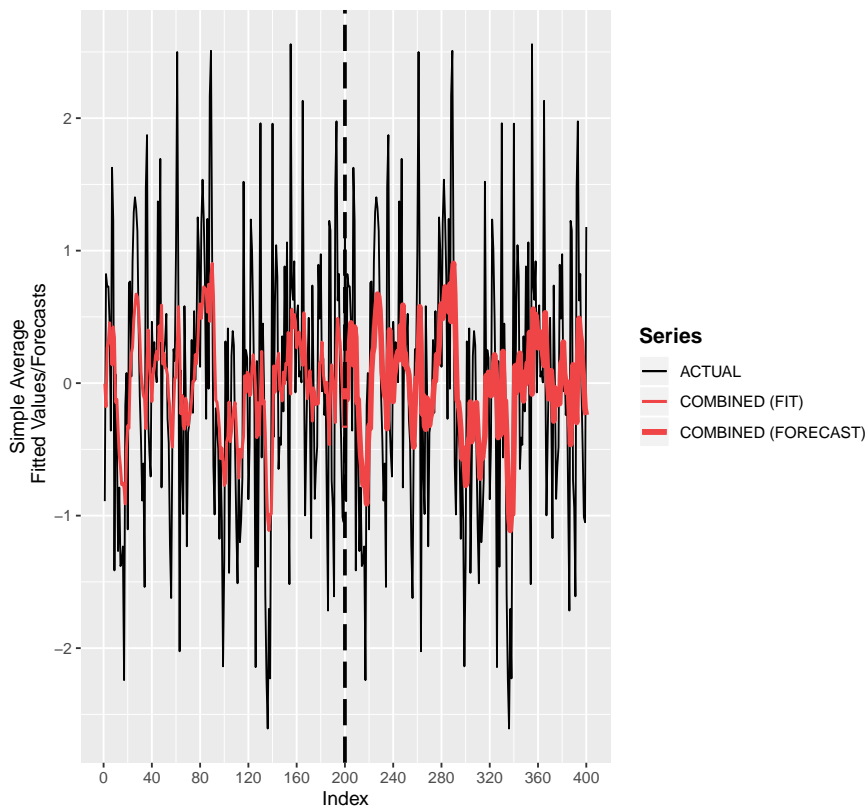
f.comb.SA.recurativa <- rolling_combine(datos.entrada.2, comb_method = "comb_SA")
f.comb.SA.recurativa$Accuracy_Test

##              ME      RMSE      MAE      MPE      MAPE
## Test set -0.0005682729 0.9000545 0.7132966 47.46215 172.6312
##              ACF1 Theil's U
## Test set 0.06190392 0.9264699

plot(f.comb.SA.recurativa)

```

Simple Average Forecast Combination Actual vs. Fitted/Test Set Forecast



Finalmente, cuando queramos crear el pronóstico, podemos emplear la función `textbfpredict` que necesita el argumento con el objeto que tenga el ensamble seleccionado de clase *foreccomb* y la matriz que tenga los pronósticos de los métodos que se están ensamblando (argumento `newpreds`).

11.5. Comentarios finales

En este capítulo hemos discutido como combinar pronósticos empleando diferentes métodos para ponderar cada metodología individual de pronóstico. Esta técnica puede ayudar a mejorar la precisión de sus estimaciones. Nos concentramos en el paquete **ForecastComb** por su flexibilidad y permitir incluir una buena cantidad de métodos. Por otro lado, este paquete tiene una gran limitación al no calcular intervalos de confianza para la predicción combinada. La verdad esta es una tarea no trivial y en estos momentos aún falta mucha investigación al respecto. La literatura sobre el tema se ha concentrado en el comportamiento del estimador puntual y no de los correspondientes intervalos. No obstante, el paquete **forecastHybrid** (Shaub y Ellis, 2019) si permite obtener intervalos de confianza para combinación de pronósticos, pero éste es mucho mas limitado en los métodos de ensamble que permite.

Capítulo 12

Modelos de Vectores Auto Regresivos (VAR)

En los dos capítulo anteriores discutimos modelos univariados en los que se emplea el comportamiento pasado de la variable bajo estudio para pronosticar el comportamiento futuro. También se discutió lo complicado que se puede tornar hacer pronósticos con modelos de regresión múltiple que emplean diferentes variables explicativas para explicar el comportamiento futuro de una variable de interés. Existe una opción intermedia que es emplear una generalización del modelo univariado AR(p) al caso de dos o más variables.

En estos modelos en vez de considerar una única variable, como en el modelo AR(p), se pueden considerar un vector de k series de tiempo que depende de p periodos pasados. A este modelo se le denomina **Vectores Autoregresivos**, más conocidos por la sigla VAR.

En general un modelo de Vectores Autoregresivos de orden p ($VAR(p)$) está dado por:

$$\mathbf{y}_t = \Phi_0 + \Phi_1 \mathbf{y}_{t-1} + \Phi_2 \mathbf{y}_{t-2} + \boldsymbol{\varepsilon}_t \quad (12.1)$$

donde \mathbf{y}_t es un vector de k series de tiempo en el periodo t ; es decir,

$$\mathbf{y}_t = \begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{k,t} \end{bmatrix} \quad (12.2)$$

Así mismo, $\boldsymbol{\varepsilon}_t$ es el vector de errores para cada variable; en otras palabras

$$\boldsymbol{\varepsilon}_t = \begin{bmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \\ \vdots \\ \varepsilon_{k,t} \end{bmatrix} \quad (12.3)$$

y Φ_0 y Φ_i son matrices de coeficientes de dimensiones $k \times 1$ y $k \times k$.

Adicionalmente, de manera similar a lo ocurrido en el caso del modelo AR(p), tendremos unos supuestos equivalentes sobre el comportamiento de los errores. Supondremos que el error tienen las siguientes propiedades:

1. El error para cada variable (y por tanto del vector) tiene media cero.

$$E[\varepsilon_t] = \mathbf{0}_{(k \times 1)} \quad (12.4)$$

2. La matriz de varianzas y covarianzas contemporánea no depende del tiempo. Noten que por fuera de la diagonal se encuentran la covarianzas entre las diferentes variables.

$$E[\varepsilon_t(\varepsilon_t)^T] = \Sigma_{(k \times k)} \quad (12.5)$$

3. No existe correlación en el tiempo entre los errores. No solo no existe autocorrelación de los errores para cada variable, sino también no hay correlación entre diferentes rezados de los errores entre variables.

$$E[\varepsilon_t(\varepsilon_{t-s})^T] = \mathbf{0}_{(k \times k)} \quad (12.6)$$

Adicionalmente a estos supuestos se puede incluir la condición de que los errores sigan una distribución multivariada normal. Este supuesto solo será necesario para realizar los intervalos de confianza para los pronósticos.

El modelo VAR expresado en 12.1 incluye una matriz de constantes (interceptos) también podría incluirse variables dummy para modelar la estacionalidad, una tendencia (time trend para capturar la tendencia creciente) u otras variables explicativas que entrarán de manera contemporánea en el modelo. Por ahora, nos concentraremos en el modelo VAR con una constante como se presenta en la ecuación 12.1.

Por ejemplo, si suponemos un VAR(2) con $k = 2$. En ese caso el VAR corresponde a:

$$\mathbf{y}_t = \Phi_0 + \Phi_1 \mathbf{y}_{t-1} + \Phi_2 \mathbf{y}_{t-2} + \varepsilon_t \quad (12.7)$$

Es decir,

$$\mathbf{y}_t = \begin{bmatrix} y_t \\ x_t \end{bmatrix} = \begin{bmatrix} \phi_{0,1} \\ \phi_{0,2} \end{bmatrix} + \begin{bmatrix} \phi_{1,11} & \phi_{1,12} \\ \phi_{1,21} & \phi_{1,22} \end{bmatrix} \begin{bmatrix} y_{t-1} \\ x_{t-1} \end{bmatrix} + \begin{bmatrix} \phi_{2,11} & \phi_{2,12} \\ \phi_{2,21} & \phi_{2,22} \end{bmatrix} \begin{bmatrix} y_{t-2} \\ x_{t-2} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{bmatrix} \quad (12.8)$$

Y los supuestos del error implican que $E[\varepsilon_{i,t}] = 0$ para $i = 1, 2$, además

$$E[\varepsilon_t(\varepsilon_t)^T] = \Sigma_{(2 \times 2)} = \begin{bmatrix} (\sigma_1)^2 & \sigma_{1,2} \\ \sigma_{2,1} & (\sigma_2)^2 \end{bmatrix} \quad (12.9)$$

y

$$E[\varepsilon_t(\varepsilon_{t-s})^T] = \begin{bmatrix} \gamma(s)_{1,1} & \gamma(s)_{1,2} \\ \gamma(s)_{2,1} & \gamma(s)_{2,2} \end{bmatrix} = \mathbf{0}_{(2 \times 2)} \quad (12.10)$$

donde, $\gamma(s)_{i,j}$ es la “auto”covarianza para el rezago s (para todo $s > 0$) entre los errores de las ecuaciones i y j .

Si miramos el sistema “ecuación por ecuación” tenemos

$$y_t = \phi_{0,1} + \phi_{1,1}y_{t-1} + \phi_{1,2}x_{t-1} + \phi_{2,1}y_{t-2} + \phi_{2,2}x_{t-2} + \varepsilon_{1,t} \quad (12.11)$$

$$x_t = \phi_{0,2} + \phi_{1,2}y_{t-1} + \phi_{1,22}x_{t-1} + \phi_{2,21}y_{t-2} + \phi_{2,22}x_{t-2} + \varepsilon_{2,t} \quad (12.12)$$

Esto implica no solo que la serie de tiempo y_t depende de p valores pasados de ella misma, como en el caso del modelo AR(p), sino que está también depende de p valores pasados de la variable x_t . Y lo mismo ocurrirá con la serie de tiempo x_t . Por eso al inicio de este capítulo decíamos que este modelo es un intermedio entre un modelo ARIMA y un modelo de regresión múltiple. Claramente se parece al primero en depender solo de valores pasados y no contemporáneos y se parece al segundo en emplear información de otras variables.

Dado que el modelo VAR(p) es una generalización de los AR(p), encontraremos que los problemas que aparecen en la estimación de los modelos VAR(p) son muy similares a los que nos enfrentamos al estimar los AR(p). Por ejemplo, el primer problema es encontrar p y posteriormente se el En este capítulo estudiaremos como encontrar p , cómo estimar un modelo VAR(p).

Adicionalmente, el VAR nos permite hacer algo que no es posible con los modelos univariados estudiados hasta ahora. El VAR permite determinar si cambios en una variable anteceden los cambios en una variable de interés. A este comportamiento en el que cambios de una variable anteceden los cambios de otras variables se le conoce como **causalidad a lo Granger**. En este capítulo estudiaremos este concepto.

Antes de continuar es importante mencionar que para que las propiedades estadísticas del VAR estimado sean las deseables es necesario que las series que conforman el vector y_t cumpla alguna de estas condiciones:

- Todas las variables son estacionarias
- Si las series no son estacionarias, todas deben tener el mismo orden de integración y tener una relación de largo plazo¹ (estar cointegradas).

Por ahora emplearemos series estacionarias para el resto de este capítulo.

12.1. Selección del orden del VAR (p)

Emplearemos un ejemplo con datos simulados para mostrar cómo es el procedimiento para seleccionar p , estimar el modelo, chequear los supuestos, pronosticar y sacar conclusiones en el contexto del modelo VAR.

Para empezar, consideremos el siguiente DGP:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} -0,3 & -0,4 \\ 0,6 & 0,5 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} -0,1 & 0,1 \\ -0,2 & 0,05 \end{bmatrix} \begin{bmatrix} y_{1,t-2} \\ y_{2,t-2} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{bmatrix} \quad (12.13)$$

¹ En el Apéndice de este capítulo se discute las pruebas de cointegración que permiten determinar si series I(d) tienen una relación de largo plazo o no. Es decir, si están cointegradas.

Además por simplicidad supongamos que $\varepsilon_{1,t} \sim N(0, 0.5)$ y $\varepsilon_{2,t} \sim N(0, 0.5)$. Adicionalmente tendremos que los errores de las dos ecuaciones no están relacionados de ninguna manera. Es decir, $\sigma_{1,2} = \sigma_{2,1} = 0$ y $\gamma(s)_{1,2} = 0$ para todo $s > 0$.

Estos datos los podemos generar de muchas formas, veamos una relativamente simple. Iniciemos definiendo los parámetros del proceso expresado en 12.1.

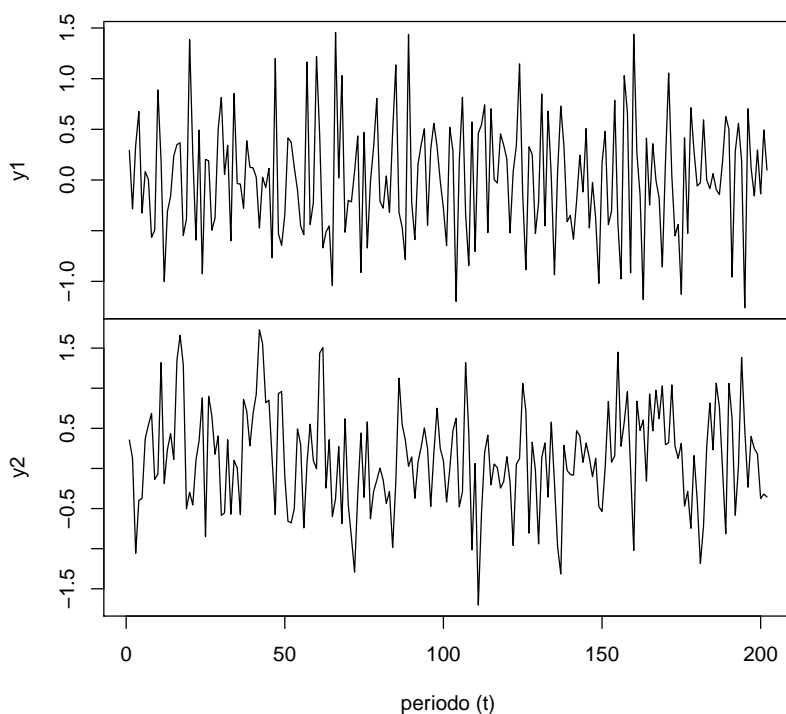
```
set.seed(12345)
T <- 200
k <- 2
p <- 2

Theta.1 <- matrix(c(-.3, .6, -.4, .5), k)
Theta.2 <- matrix(c(-.1, -.2, .1, .05), k)
Theta <- cbind(Theta.1, Theta.2)
```

Ahora, construyamos la muestra empleando el siguiente código.

```
series <- matrix(0, k, T + 2*p) # Serie inicial con solo ceros
for (i in (p + 1):(T + 2*p)){ # Genera las series con  $\varepsilon \sim N(0, 0.5)$ 
  series[, i] <- Theta.1%*series[, i-1] + Theta.2%*series[, i-2] +
    rnorm(k, 0, .5)
}
series <- as.data.frame(t(series))
colnames(series) <- c("y1", "y2")
series <- ts(series[-(1:p),])
```

En la figura 12.1 se presentan las series simuladas.

Figura 12.1 Series simuladas a partir del modelo 12.1

Al igual que en el caso de los modelos $AR(p)$, la primera tarea es seleccionar el número de rezagos óptimo. En el caso de los modelos $AR(p)$, estudiamos en el capítulo 11. En este método se emplean las ACF y PACF estimadas para encontrar el orden del proceso. La segunda aproximación emplea criterios de información para sugerir el mejor p . En ambos casos una vez detectado el posible orden del modelo (p) se constata que los supuestos de los residuales se cumplan. Si se encontraba unos residuales autocorrelacionados se incluía un rezago más al modelo.

En el caso de los VAR, vamos a seguir una aproximación similar a la segunda. Es decir, vamos a emplear criterios de información para seleccionar el número de rezagos que emplearemos o por lo menos tener un punto de arranque para seleccionar el mejor número de rezagos. Después chequearemos si los residuales cumplen los supuestos. En caso contrario se introducirán rezagos hasta eliminar los problemas de

autocorrelación, o se emplearán transformaciones de las variables para solucionar posibles problemas de comportamiento GARCH o ARCH.

Com o ya lo vimos en el capítulo refch:EstimaciónARMA, emplear los criterios de información para encontrar el p óptimo es una aproximación que ahorra tiempo. En el caso de los modelos VAR, el paquete **vars** (Pfaff, 2008) tiene una función que permite hacer esto para los siguientes criterios de información: *AIC*, *HQ*, *SC* (*BIC*) y *FPE*. Estos criterios son versiones adaptadas al mundo multivariado de sus contrapartes univariadas que estudiamos en el Capítulo ??ónARMAch:EstimaciónARMA sección 9.3). Los criterios de información dependerán del número de parámetros estimados en el respectivo VAR (p^*) que a su vez depende del número de rezagos (p) y de las variables incluidas en el VAR (k). Los criterios de información en este caso corresponden a:

- *criterio de Akaike (Hirotugu AKAIKE, 1981) (AIC):*

$$AIC(p) = \ln(\det(\tilde{\Sigma})) + \frac{2}{T}pk^2$$

- *criterio Bayesiano de Schwarz (Schwarz, 1978) (SBC):*

$$SBC(p) = \ln(\det(\tilde{\Sigma})) + \frac{\ln(T)}{T}pk^2$$

- *criterio de Hannan y Quinn (Hannan y Quinn, 1979) (HQ):*

$$HQ(p) = \ln(\det(\tilde{\Sigma}(n))) + \frac{2\ln(\ln(T))}{T}pk^2$$

donde $\tilde{\Sigma} = \frac{\sum_{t=1}^T \hat{\epsilon}_t \hat{\epsilon}_t'}{T}$ es la matriz de varianzas y covarianzas estimada del vector de residuales.

Adicionalmente, se acostumbra emplear el Error Final de Predicción final prediction error (*FPE* por la sigla del término en inglés final prediction error). La expresión del *FPE* corresponde a:

$$FPE(p) = \left(\frac{T + p^*}{T - p^*} \right)^k \det(\tilde{\Sigma}).$$

La función que implementa esto en *R* es *VARselect*. Esta función típicamente incluye los siguientes argumentos:

VARselect(y, lag.max = 10, type = c("const", "trend", "both", "none"), season = NULL, exogen = NULL)

donde:

- “y”: El objeto que contiene las observaciones de las variables endógenas del VAR
- “lag.max”: el número máximo de rezagos (p) para realizar la búsqueda del mejor modelo. Por defecto se emplea un número máximo de 10 rezagos.

- “type”: Este argumento permite a la función saber si se incluirá un vector de constantes o interceptos como en la expresión 12.1 (`type = "const"`), un vector de tendencias (`type = "trend"`), tanto un vector de constantes como de tendencia (`type = "both"`), o ningún tipo de variable determinística (no intercepto no tendencia) (`type = "none"`)..
- “season”: Si se desea incluir variables dummy estacionales. Si se desea emplear variables dummy para datos mensuales entonces “season = 12” y para datos trimestrales “season = 4”.

Dado que de entrada es imposible saber si el modelo VAR debe o no incluir interceptos, en este caso la mejor idea es emplear ésta función de la siguiente manera:

```
library(vars)

## Loading required package: strucchange
## Loading required package: sandwich
## Loading required package: urca

VARselect(series, lag.max = 8, type = "const")

## $selection
## AIC(n)  HQ(n)  SC(n)  FPE(n)
##      3      1      1      3
##
## $criteria
##              1              2              3              4
## AIC(n) -2.61600680 -2.62626723 -2.63056845 -2.60889103
## HQ(n)  -2.57508162 -2.55805859 -2.53507635 -2.48611547
## SC(n)  -2.51493902 -2.45782094 -2.39474364 -2.30568770
## FPE(n)  0.07309452 0.07234967 0.07204202 0.07362595
##              5              6              7              8
## AIC(n) -2.59790279 -2.56683346 -2.55934034 -2.54760172
## HQ(n)  -2.44784377 -2.38949098 -2.35471440 -2.31569232
## SC(n)  -2.22732093 -2.12887309 -2.05400145 -1.97488431
## FPE(n)  0.07444764 0.07680918 0.07740361 0.07833973
```

El criterio *BIC* (*SC*) es el único que propone un modelo *VAR*(1), los otros tres criterios sugieren un *VAR*(2). El siguiente paso será chequear los supuestos de los dos modelos candidatos para decidir qué *p* emplear.

12.2. Chequeo de los supuestos del error

Iniciemos analizando el *VAR*(2). Estimemos dicho modelo y veamos los gráficos de diagnóstico predeterminados en el paquete **vars**. Pero antes será necesario estimar el modelo *VAR*(2) con la función *VAR* que tiene los mismos argumentos de la función *VARselect*. En este caso podemos estimar el modelo y guardarlo en el objeto *var.2c* que será de clase *varest*.

```

var.2c <- VAR(series, p = 2, type = "const")
summary(var.2c)

##
## VAR Estimation Results:
## =====
## Endogenous variables: y1, y2
## Deterministic variables: const
## Sample size: 200
## Log Likelihood: -293.52
## Roots of the characteristic polynomial:
## 0.5487 0.5487 0.4823 0.01472
## Call:
## VAR(y = series, p = 2, type = "const")
##
##
## Estimation results for equation y1:
## =====
## y1 = y1.l1 + y2.l1 + y1.l2 + y2.l2 + const
##
##      Estimate Std. Error t value Pr(>|t|)
## y1.l1 -0.28062    0.07070  -3.969 0.000101 ***
## y2.l1 -0.37761    0.06864  -5.501 1.17e-07 ***
## y1.l2 -0.11446    0.07818  -1.464 0.144792
## y2.l2  0.05752    0.07051   0.816 0.415624
## const  0.07125    0.03707   1.922 0.056039 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.5025 on 195 degrees of freedom
## Multiple R-Squared: 0.2412, Adjusted R-squared: 0.2257
## F-statistic: 15.5 on 4 and 195 DF, p-value: 5.017e-11
##
##
## Estimation results for equation y2:
## =====
## y2 = y1.l1 + y2.l1 + y1.l2 + y2.l2 + const
##
##      Estimate Std. Error t value Pr(>|t|)
## y1.l1  0.58910    0.07302   8.068 7.15e-14 ***
## y2.l1  0.42902    0.07089   6.052 7.21e-09 ***
## y1.l2 -0.10568    0.08075  -1.309  0.192
## y2.l2  0.07178    0.07282   0.986  0.326
## const  0.04836    0.03828   1.263  0.208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.519 on 195 degrees of freedom
## Multiple R-Squared: 0.3396, Adjusted R-squared: 0.3261
## F-statistic: 25.07 on 4 and 195 DF, p-value: < 2.2e-16
##
##

```

```
##
## Covariance matrix of residuals:
##      y1      y2
## y1  0.25253 -0.01177
## y2 -0.01177  0.26937
##
## Correlation matrix of residuals:
##      y1      y2
## y1  1.00000 -0.04512
## y2 -0.04512  1.00000

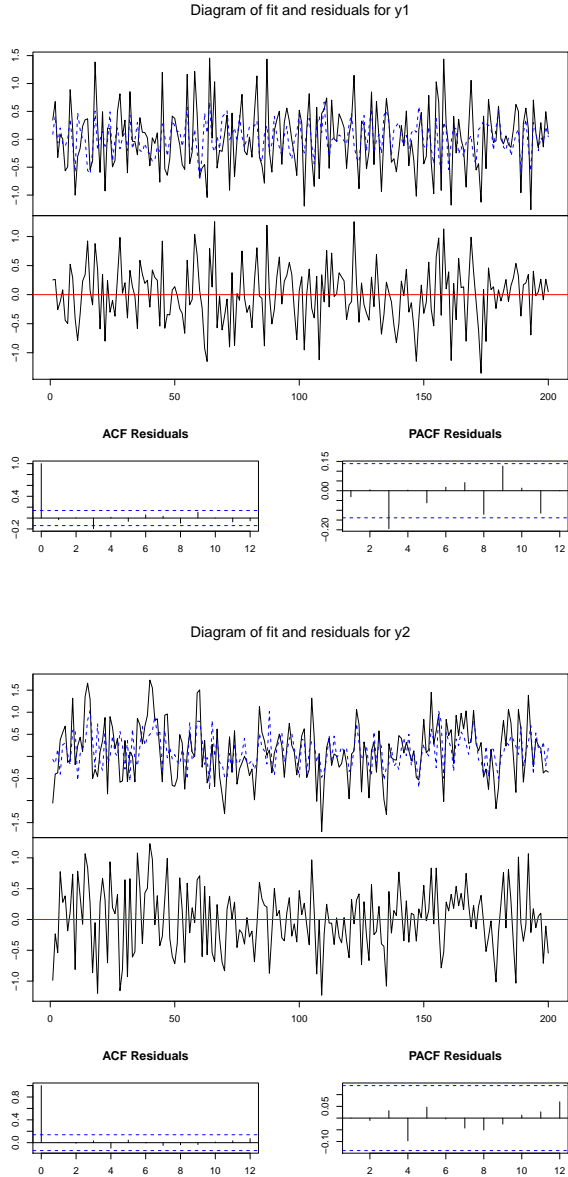
print(var.2c)

##
## VAR Estimation Results:
## =====
##
## Estimated coefficients for equation y1:
## =====
## Call:
## y1 = y1.l1 + y2.l1 + y1.l2 + y2.l2 + const
##
##      y1.l1      y2.l1      y1.l2      y2.l2      const
## -0.28062162 -0.37760799 -0.11446076  0.05752004  0.07124784
##
##
## Estimated coefficients for equation y2:
## =====
## Call:
## y2 = y1.l1 + y2.l1 + y1.l2 + y2.l2 + const
##
##      y1.l1      y2.l1      y1.l2      y2.l2      const
##  0.5891035  0.4290193 -0.1056815  0.0717795  0.0483635
```

Ahora, podemos emplear los gráficos preestablecidos de diagnostico con la función *plot* aplicada al objeto de clase *varest*.

```
plot(var.2c)
```

Figura 12.2 Residuales de las dos series de los residuales



Existen varias adaptaciones de las pruebas de autocorrelación anteriormente estudiadas para las series univariadas (Ver Capítulo 4, sección 4.3.2) para este ambiente multivariado. Dos de las más comunes son las versiones multivariadas de las pruebas de Portmanteau y Breusch-Godfrey.

La prueba de Portmanteau está diseñada para probar la hipótesis nula de no existencia de correlación de orden s (par los s primeros rezagos) para los errores estimados de un $VAR(p)$ estable, versus la alterna de que existe algún tipo de dependencia (correlación). El correspondiente estadístico es:

$$Q_s = T \sum_{j=1}^s tr(\hat{\Sigma}_j^T \hat{\Sigma}_0^{-1} \hat{\Sigma}_j \hat{\Sigma}_0^{-1})$$

, donde $\hat{\Sigma}_i = \frac{1}{T} \sum_{t=i+1}^T \hat{\epsilon}_t \hat{\epsilon}_{t-i}^T$. Está bien documentado que este estadístico sigue aproximadamente una distribución ² con $k^2(s-p)$ grados de libertad.

También está bien documentado que para muestras pequeñas o valores pequeños de s , el comportamiento de la prueba de Portmanteau no es el mejor y por eso existe una corrección al estadístico que implica:

$$Q_s^* = T^2 \sum_{j=1}^s \frac{1}{T-j} tr(\hat{\Sigma}_j^T \hat{\Sigma}_0^{-1} \hat{\Sigma}_j \hat{\Sigma}_0^{-1})$$

La prueba Breusch-Godfrey (estadístico LM) tiene como idea básica que si los errores estuviesen correlacionados, entonces puede existir un VAR que los represente. Por tanto la hipótesis nula de esta prueba es que todas las matrices que acompañarían al vector de errores estimados rezagos son conjuntamente cero para los primeros s rezagos, versus la alterna que alguna de estas matrices no son cero. En otras palabras, la nula implica errores no relacionados y la alterna algún tipo de relación.

Esta prueba implica estimar la siguiente regresión auxiliar:

$$\hat{\epsilon}_t = A_1 y_{t-1} + \dots + A_p y_{t-p} + CD_t + B_1 \hat{\epsilon}_{t-1} + \dots + B_h \hat{\epsilon}_{t-h} + u_t$$

donde CD_t representa las variables dummy para estacionalidad (si se emplearon en la estimación del modelo). El estadístico de esta prueba implica:

$$LM_h = T(k - tr(\tilde{\Sigma}_R^{-1} \tilde{\Sigma}_U)),$$

donde $\tilde{\Sigma}_R$ y $\tilde{\Sigma}_U$ representan las matrices de varianzas y covarianzas de los residuales de los modelos restringido (R) y no restringido (U), respectivamente. El estadístico LM_h sigue una distribución ² con sk^2 grados de libertad.

Edgerton y Shukur, 1999 propusieron una corrección a la prueba anterior para muestras pequeñas. Esta corrección implica el siguiente estadístico:

$$LMF_s = \frac{1 - (1 - R_r^2)^{1/r}}{(1 - R_r^2)^{1/r}} \frac{Nr - q}{km},$$

donde $R_r^2 = 1 - |\tilde{\Sigma}_U|/|\tilde{\Sigma}_R|$, $r = ((k^2 m^2 - 4)/(k^2 + m^2 - 5))^{1/2}$, $q = 1/2Km - 1$, $N = T - k - m - 1/2(k - m + 1)$, y $m = ks$. Este estadístico modificado sigue una distribución F con sk^2 grados de libertad en el numerados y $int(Nr - q)$ en el denominador.

La función *serial.test* del paquete **vars** permite realizar estas pruebas. Esta función típicamente incluye los siguientes argumentos:

```
serial.test(x, lags.pt = 16, lags.bg = 5, type = c("PT.asymptotic", "PT.adjusted",
"BG", "ES"))
```

donde:

- “x”: objeto de clase *varest* (objeto donde se encuentre el modelo VAR estimado por medio del paquete *vars*).
- “lags.pt”: el número de rezagos máximo para realizar la prueba de Portmanteau. Por defecto es igual a 16.
- “lags.bg”: el número de rezagos máximo para realizar la prueba de Breusch-Godfrey. Por defecto es igual a 5.
- “type”: el tipo de prueba. Si `type = "PT.asymptotic"`, entonces se efectuará la prueba de Portmanteau. Si `type = "PT.adjusted"`, entonces se calculará la corrección para muestras pequeñas o pocos rezagos de la prueba de Portmanteau. Si `type = "BG"` se efectúa la prueba LM de Breusch-Godfrey. Y finalmente, si `type = "ES"`, se estima la corrección para muestras pequeñas de la prueba LM.

En este caso, tenemos que la prueba de Portmanteau para el primer rezago se puede obtener de la siguiente manera:

```
serial.test(var.2c, lags.pt = 1, type = "PT.asymptotic")

## Warning in pchisq(STATISTIC, df = PARAMETER): NaNs produced
## Warning in pchisq(STATISTIC, df = PARAMETER): NaNs produced
##
## Portmanteau Test (asymptotic)
##
## data: Residuals of VAR object var.2c
## Chi-squared = 0.38803, df = -4, p-value = NA
```

De manera análoga a la versión univariada, podemos escribir una función que haga esta prueba para muchos rezagos al tiempo.

```
tabla.Auto.VAR <- function(VAR, max.lag = 20,
                           type = "PT.asymptotic"){
  # se crean objetos para guardar los resultados
  estadistico <- matrix(0,max.lag,1)
  pval <-matrix(0,max.lag,1)

  # se calcula la prueba para los diferentes rezagos
  for (i in 1:max.lag) {
    SerialCorrelation <- serial.test(VAR, lags.pt=i, lags.bg=i, type = type)
    estadistico[i] <- SerialCorrelation$serial$statistic
    pval[i]<-round(SerialCorrelation$serial$p.value,5)
  }
  labels<- c( "Rezagos", type, "p-valor")

  Cuerpo.Tabla <- cbind(matrix(1:max.lag,max.lag,1),
                        estadistico, pval)
```

```
TABLBP <- data.frame(Cuerpo.Tabla)
names(TABLBP) <- labels
return(TABLBP)
}
```

Cuadro 12.1 Prueba de Portmanteau para los errores del modelo VAR estimado (primeros 10 rezagos)

Rezagos	PT.asymptotic	p-valor
1.00	0.39	
2.00	0.86	0.00
3.00	10.73	0.03
4.00	14.54	0.07
5.00	15.96	0.19
6.00	17.44	0.36
7.00	23.40	0.27
8.00	32.12	0.12
9.00	34.76	0.18
10.00	35.49	0.31

Cuadro 12.2 Prueba de Portmanteau ajustada para muestras pequeñas para los errores del modelo VAR estimado (primeros 10 rezagos)

Rezagos	PT.adjusted	p-valor
1.00	0.39	
2.00	0.86	0.00
3.00	10.89	0.03
4.00	14.78	0.06
5.00	16.23	0.18
6.00	17.76	0.34
7.00	23.93	0.25
8.00	33.02	0.10
9.00	35.78	0.15
10.00	36.55	0.27

Los resultados soportan que los residuos no parecen tener correlaciones que dependan del rezago. Noten que si bien para los primeros dos rezagos las pruebas de Portmanteau y Portmanteau ajustada permiten rechazar la nula, para rezagos superiores no se puede rechazar la nula de que todos los rezagos las autocovarianzas son cero.

Ahora consideremos la posibilidad de un comportamiento ARCH en los residuales (en todos o alguno de ellos). Existe la prueba ARCH-LM multivariada que permite chequear este comportamiento heteroscedástico o no. En el Apéndice al final de este capítulo se presenta el detalle de la prueba (ver sección 12.4.1).

La hipótesis nula de esta prueba es que no existe comportamiento ARCH multivariado para los primeros q rezagos. La hipótesis alterna implica que existe algún tipo de comportamiento ARCH. El estadístico de prueba $VARCH_{LM}(q)$ sigue una

Cuadro 12.3 Prueba de Breusch-Godfrey para los errores del modelo VAR estimado (primeros 10 rezagos)

Rezagos	BG	p-valor
1.00	11.07	0.03
2.00	12.68	0.12
3.00	20.73	0.05
4.00	23.02	0.11
5.00	25.69	0.18
6.00	28.43	0.24
7.00	36.87	0.12
8.00	42.93	0.09
9.00	45.36	0.14
10.00	45.87	0.24

Cuadro 12.4 Prueba de Edgerton y Shukur (1999) ajustada para muestras pequeñas para los errores del modelo VAR estimado (primeros 10 rezagos)

Rezagos	ES	p-valor
1.00	2.74	0.03
2.00	1.56	0.14
3.00	1.72	0.06
4.00	1.43	0.13
5.00	1.27	0.19
6.00	1.17	0.27
7.00	1.31	0.14
8.00	1.34	0.11
9.00	1.26	0.15
10.00	1.13	0.28

distribución χ^2 con $qk^2(k+1)^2/4$ grados de libertad(ver sección 12.4.1 para el detalle de la prueba).

Esta prueba se puede realizar empleando la función *arch.test* del paquete **vars**. Esta función típicamente incluye los siguientes argumentos:

arch.test(x, lags.multi = 5)

donde:

- "x": el objeto de clase *varest* al que se le hará la prueba.
- "lags.multi": El número de rezagos para el que se realizará la prueba. Por defecto "lags.multi = 5".

La prueba multivariada para los primeros 10 rezagos se puede calcular de la siguiente manera.

```
arch.test(var.2c, lags.multi = 10)

##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object var.2c
## Chi-squared = 96.745, df = 90, p-value = 0.2945
```


De manera análoga, podemos construir una función que permita realizar esta prueba para varios rezagos a la vez. En este caso:

```

tabla.ARCH.VAR <- function(VAR, max.lag = 20){
  # se crean objetos para guardar los resultados
  estadistico <- matrix(0,max.lag,1)
  pval <-matrix(0,max.lag,1)

  # se calcula la prueba para los diferentes rezagos
  for (i in 1:max.lag) {

    ARCH<- arch.test(VAR, lags.multi=i)
    estadistico[i]<-ARCH$sarch.mul$statistic
    pval[i]<-round(ARCH$sarch.mul$p.value,5)
  }
  labels<- c( "Rezagos", "ARCH LM-Estadístico", "p-valor")

  Cuerpo.Tabla <- cbind(matrix(1:max.lag,max.lag,1),
                        estadistico, pval)
  TABLABP <- data.frame(Cuerpo.Tabla)
  names(TABLABP) <- labels
  return(TABLABP)
}
tabla.ARCH.VAR(var.2c, 10)

##      Rezagos ARCH LM-Estadístico p-valor
## 1          1          5.070769 0.82810
## 2          2         13.578046 0.75615
## 3          3         34.570316 0.15009
## 4          4         43.975902 0.16962
## 5          5         62.245895 0.04502
## 6          6         66.306903 0.12147
## 7          7         73.643663 0.16899
## 8          8         77.048082 0.32043
## 9          9         86.583582 0.31522
## 10         10         96.745327 0.29455

```

En el Cuadro 12.5 podemos ver que no existe evidencia de comportamiento ARCH multivariado en los residuales.

Ahora miremos si existe normalidad. Esta es necesaria para la inferencia en muestras pequeñas y para las proyecciones. Podemos realizar una versión multivariada de la prueba JB de normalidad de la siguiente manera:

```

normality.test(var.2c)

## $JB
##
## JB-Test (multivariate)
##
## data: Residuals of VAR object var.2c
## Chi-squared = 1.4608, df = 4, p-value = 0.8336
##

```

Cuadro 12.5 Prueba de ARCH-LM Multivariada para los errores del modelo VAR estimado (primeros 10 rezagos)

Rezagos	ARCH LM-Estadístico	p-valor
1.00	5.07	0.83
2.00	13.58	0.76
3.00	34.57	0.15
4.00	43.98	0.17
5.00	62.25	0.05
6.00	66.31	0.12
7.00	73.64	0.17
8.00	77.05	0.32
9.00	86.58	0.32
10.00	96.75	0.29

```
##
## $Skewness
##
## Skewness only (multivariate)
##
## data: Residuals of VAR object var.2c
## Chi-squared = 0.044585, df = 2, p-value = 0.978
##
##
## $Kurtosis
##
## Kurtosis only (multivariate)
##
## data: Residuals of VAR object var.2c
## Chi-squared = 1.4162, df = 2, p-value = 0.4926
```

En este caso la nula de que todas las series de los errores provienen de distribuciones normales no se puede rechazar. Es decir, existe evidencia que los errores siguen una distribución normal multivariada.

Ya tenemos suficiente evidencia para afirmar que el modelo $VAR(2)$ está libre de problemas. Podemos compararlo con el modelo $VAR(1)$ que fue seleccionado por el criterio de información BIC . El lector puede verificar que con un $VAR(1)$ se encuentra problemas de autocorrelación en los residuos (ver prueba de Portamenteu corregida). Así que seguiremos trabajando con el $VAR(2)$.

12.3. Usando los resultados

Ahora podemos hacer diferente tipos de análisis. Por ejemplo podemos hacer: pronósticos, las pruebas de causalidad de Granger o calcular la función impulso respuesta.

12.3.1. Pronósticos

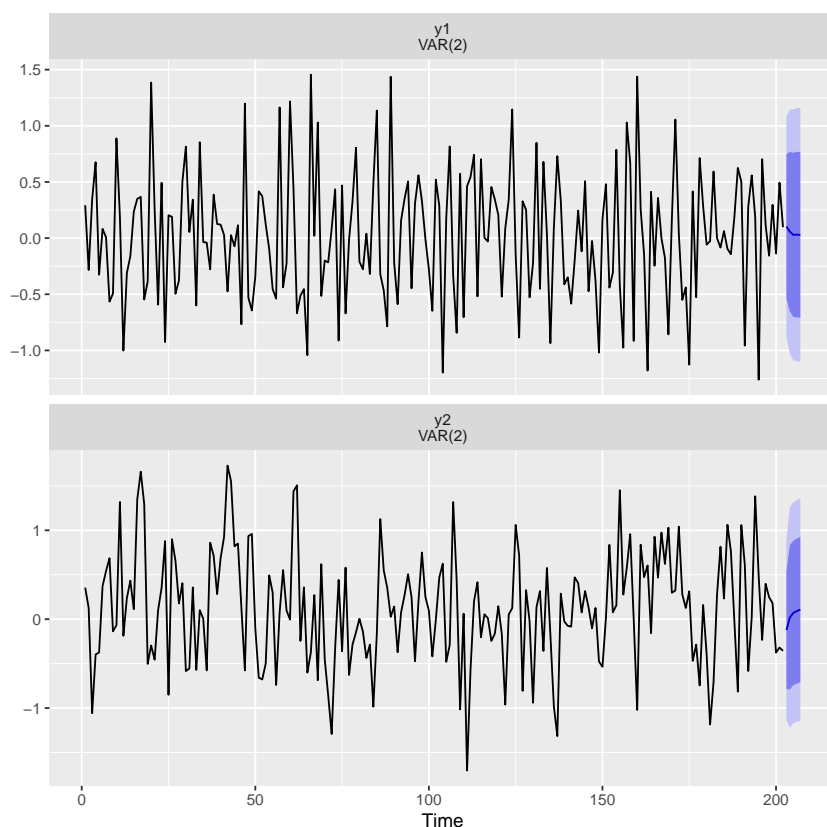
Los pronósticos se pueden construir fácilmente con la función *forecast* del paquete **forecast** (Hyndman y Khandakar, 2008). Esta es la misma función que empleamos en los Capítulos 3, 5 y ??ónARMAch:EstimaciónARMAronóstico para los siguientes 5 períodos se puede obtener de la siguiente manera.

```
var.2c.prd <- forecast(var.2c, h=5)
var.2c.prd
```

## y1	##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## 203		0.10318533	-0.5408252	0.7471958	-0.8817437	1.088114
## 204		0.05657081	-0.6550450	0.7681866	-1.0317516	1.144893
## 205		0.02860243	-0.7021882	0.7593931	-1.0890454	1.146250
## 206		0.03111178	-0.7063438	0.7685674	-1.0967292	1.158953
## 207		0.02887587	-0.7093648	0.7671166	-1.1001659	1.157918
## y2	##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## 203		-0.12165718	-0.7867966	0.5434822	-1.138900	0.8955858
## 204		0.02108542	-0.7900907	0.8322615	-1.219501	1.2616722
## 205		0.07109834	-0.7404814	0.8826781	-1.170106	1.3123025
## 206		0.09125087	-0.7244870	0.9069887	-1.156313	1.3388143
## 207		0.10792059	-0.7104577	0.9262989	-1.143681	1.3595223

Con las funciones *plot* o *autoplot* se pueden obtener gráficos de los pronósticos².

² Otra opción de gráfico es el denominado "fanchart". Este se puede obtener realizando los pronósticos con la función *predict* y luego graficar esos pronósticos con la función *fanchart*. En este caso sería crear el objeto con `var.2c.prd <- predict(var.2c, n.ahead = 8, ci = 0.95)` y luego el gráfico con `fanchart(var.2c.prd)`.

Figura 12.3 Pronósticos para los siguientes cinco periodos empleando la función *autoplot*

Es importante mencionar que aquí también aplica las evaluaciones fuera de muestra cómo se discutieron en el capítulo 10. Este puede ser otro método de pronosticar que puede competir con los modelos univariados para conseguir el mejor método de pronóstico.

12.3.2. Prueba de causalidad de Granger

La causalidad de Granger (Granger, 1969) es muy empleada por los economistas para encontrar la causalidad entre series de tiempo. En este contexto la causalidad (a lo Granger) está asociada a la idea de causa y efecto, aunque no es exactamente la misma. Una variable x_t causa a la variable y_t si movimientos de x_t anteceden en el tiempo a los cambios en y_t .

Sin embargo, con la causalidad a lo Granger, no se está probando una verdadera relación de causa y efecto. Lo que se puede saber es si una variable en particular se mueve antes que otra. En otras palabras, si encontramos causalidad a lo Granger, no hay un vínculo causal en el verdadero sentido de la palabra. No obstante esta limitación, esta prueba puede ser de gran utilidad en algunos ambientes.

En el ejemplo que venimos trabajando tenemos que nuestro VAR(2) expresado “ecuación por ecuación” es

$$y_{1,t} = \phi_{0,1} + \phi_{1,11}y_{1,t-1} + \phi_{1,12}y_{2,t-1} + \phi_{2,11}y_{1,t-2} + \phi_{2,12}y_{2,t-2} + \varepsilon_{1,t} \quad (12.14)$$

$$y_{2,t} = \phi_{0,2} + \phi_{1,21}y_{1,t-1} + \phi_{1,22}y_{2,t-1} + \phi_{2,21}y_{1,t-2} + \phi_{2,22}y_{2,t-2} + \varepsilon_{2,t} \quad (12.15)$$

Si queremos probar si $y_{2,t}$ causa a lo Granger a $y_{1,t}$, esto es equivalente a probar la hipótesis nula que $\phi_{1,12} = \phi_{2,12} = 0$. Si la hipótesis nula es verdadera, entonces esto significa que los cambios en periodos pasados de $y_{2,t}$ no afecta a $y_{1,t}$ y por tanto diremos que $y_{2,t}$ no causa a lo Granger a $y_{1,t}$. Por el contrario si rechazamos la hipótesis nula, esto implicará que al menos uno de esos coeficientes es diferente de cero y por tanto $y_{2,t}$ causa a lo Granger a $y_{1,t}$.

La función *causality* del paquete **vars** permite realizar tanto la prueba de causalidad de Granger como una prueba conocida como la prueba de causalidad instantánea. Esta función típicamente incluye los siguientes argumentos:

causality(x, cause = NULL, vcov.=NULL, boot=FALSE, boot.runs=100)

donde:

- “x”: el objeto de clase *varest* al que se le hará la prueba.
- “cause”: permite especificar qué variable o variables queremos probar que cause a las demás. Si no se escribe algo, se empleará la primera variable del objeto *vars*.
- “vcov.”: permite emplear estimadores HC estimadores para la matriz de varianzas y covarianzas de los coeficientes en caso que exista heteroscedasticidad.
- “boot”: permite pedirle a la función que calcule los valores p empleado bootstrapping. Por defecto *boot=FALSE*.
- “boot.runs”: permite especificar el número de veces que se replicará el ejercicio para generar los estadísticos por medio del bootstrapping.

Para nuestro caso, si queremos probar si y_2 causa a y_1 tenemos:

```
causality(var.2c, cause = "y2")

## $Granger
##
## Granger causality H0: y2 do not Granger-cause y1
##
## data: VAR object var.2c
## F-Test = 16.109, df1 = 2, df2 = 390, p-value = 1.897e-07
##
##
## $Instant
##
```

```
## H0: No instantaneous causality between: y2 and y1
##
## data: VAR object var.2c
## Chi-squared = 0.40626, df = 1, p-value = 0.5239
```

y para el caso contrario

```
causality(var.2c, cause = "y1")

## $Granger
##
## Granger causality H0: y1 do not Granger-cause y2
##
## data: VAR object var.2c
## F-Test = 38.655, df1 = 2, df2 = 390, p-value = 4.441e-16
##
##
## $Instant
##
## H0: No instantaneous causality between: y1 and y2
##
## data: VAR object var.2c
## Chi-squared = 0.40626, df = 1, p-value = 0.5239
```

Noten que se puede concluir que las dos variables se causan mutuamente pero no instantáneamente.

12.3.3. Funciones impulso respuesta

La función de Impulso-Respuesta nos permite conocer, de acuerdo al modelo VAR estimado, cuál sería la respuesta de las variables involucradas en el VAR ante una perturbación inesperada en sólo una de las variables involucradas.

Es decir, es una manera de determinar como las variables involucradas en el VAR reaccionan a cambios inesperados en una de las variables³ Es importante anotar que la relaciones a las perturbaciones que se grafican en la función impulso respuesta se acompañan de intervalos de confianza del 95 %, los cuales permiten determinar si la reacción de las variables si son estadísticamente diferente de cero.

La función *irf* del paquete **vars** permite calcular la función impulso respuesta. Esta función típicamente incluye los siguientes argumentos:

irf(x, impulse = NULL, response = NULL, n.ahead = 10)

donde:

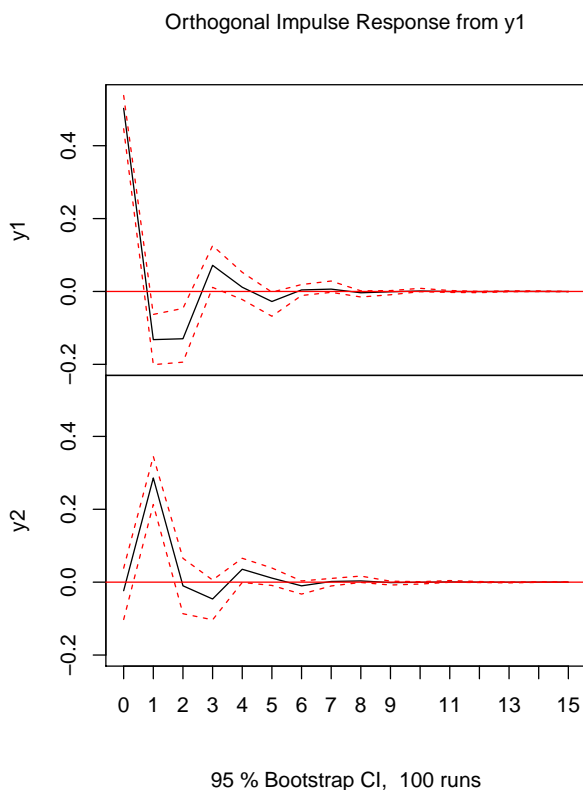
- "x": el objeto de clase *varest* del que se calculará la función impulso respuesta.
- "impulse": un vector que contiene el nombre de las variables que tendrán el impulso. Es decir, las variables que tendrán las perturbaciones inesperadas.

³ Los cambios inesperados también se conocen como perturbaciones o choques.

- "response": un vector que contiene el nombre de las variables para las cuales se calculará la respuesta.
- "n.ahead": el número de periodos para los cuales se calcularán las respuestas.

Esta función por defecto emplea bootstrapping para calcular los intervalos de confianza que por defecto serán del 95 %. El impulso de y1 sobre las dos variables para los siguientes 15 periodos se puede calcular de la siguiente manera

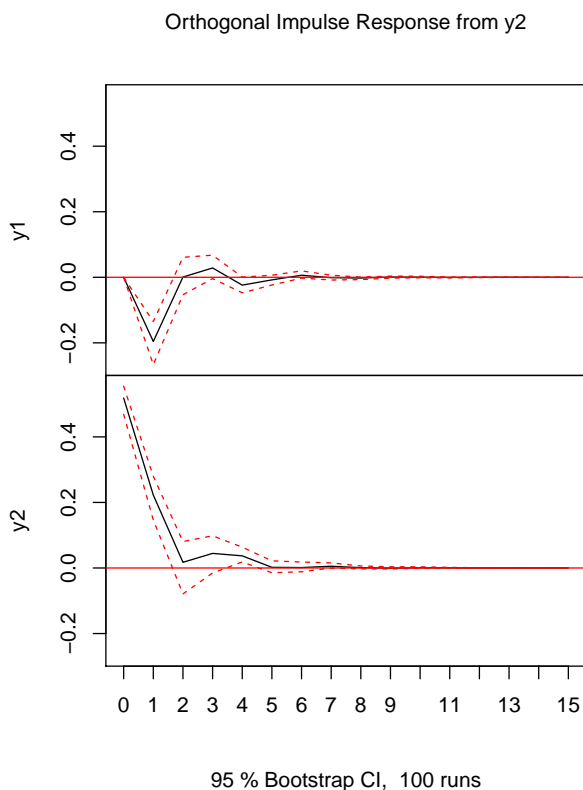
```
Imp.Res1<-irf(var.2c, impulse = "y1", response = c("y1", "y2"), n.ahead = 15)
plot(Imp.Res1)
```



Estos resultados implican que un cambio positivo inesperado en y1 generará un cambio positivo únicamente un periodo después del choque. Los siguientes periodos no hay una respuesta (Ver intervalos). El efecto sobre la misma variable y1 es negativo por los dos primeros periodos. De ahí en adelante no hay efecto.

La reacción ante choques en y2 la podemos calcular de la siguiente manera:

```
Imp.Res2<-irf(var.2c, impulse = "y2", response = c("y1", "y2"), n.ahead = 15)
plot(Imp.Res2)
```



En este caso el efecto del choque y_2 sobre ella misma se observa en el periodo 1, 2 y 4. Mientras que para la variable y_1 se observa un efecto negativo solo para el primer periodo después del choque.

12.4. Apéndices

Prueba de Cointegración.

12.4.1. Prueba ARCH-LM multivariada

La versión multivariada de la prueba ARCH-LM implica la siguiente regresión auxiliar :

$$vech(\hat{\varepsilon}_t \hat{\varepsilon}_t^T) = \beta_0 + B_1 vech(\hat{\varepsilon}_{t-1} \hat{\varepsilon}_{t-1}^T) + \dots + B_q vech(\hat{\varepsilon}_{t-q} \hat{\varepsilon}_{t-q}^T) + \mathbf{v}_t$$

donde \mathbf{v}_t corresponde a un error esférico y $vech$ es el "operador de apilado de columnas" (column-stacking operator) para matrices simétricas que apila las columnas a partir de la diagonal principal hacia abajo. β_0 tiene dimensiones $\frac{1}{2}k(k+1) \times 1$ y las matrices de coeficientes B_i con $i = 1, \dots, q$ tiene dimensiones $\frac{1}{2}k(k+1) \times \frac{1}{2}k(k+1)$.

La hipótesis nula de esta prueba es que no existe comportamiento ARCH multivariado para los primeros q rezagos. Es decir, $B_1 = B_2 = \dots = B_q = \mathbf{0}$. La hipótesis alterna implica que alguna de esas matrices no tiene ceros y por tanto existe algún tipo de comportamiento ARCH. El estadístico de prueba corresponde a:

$$VARCH_{LM}(q) = \frac{1}{2} T k(k+1) R_m^2 \quad ,$$

donde

$$R_m^2 = 1 - \frac{2}{k(k+1)} tr(\hat{\Omega} \hat{\Omega}_0^{-1}) \quad ,$$

y $\hat{\Omega}$ representa la matriz de varianzas y covarianzas de la regresión auxiliar. Este estadístico sigue una distribución χ^2 con $qk^2(k+1)^2/4$ grados de libertad.

Referencias

- Gilbert, D. (2009). *Stumbling on happiness*. Vintage Canada.
- Bell, W. R. & Hillmer, S. C. (2002). Issues Involved With the Seasonal Adjustment of Economic Time Series. *Journal of Business & Economic Statistics*, 20(1), 98-127. doi:10.1198/073500102753410426. eprint: <https://doi.org/10.1198/073500102753410426>
- Shiskin, J. (1965). *The X-11 variant of the census method II seasonal adjustment program*. US Government Printing Office.
- Dagum, E. B. & Bianconcini, S. (2016). *Seasonal Adjustment Methods and Real Time Trend- Cycle Estimation*. doi:10.1007/978-3-319-31822-6
- Brown, R. G. (1959). *Statistical forecasting for inventory control*. McGraw/Hill.
- Holt, C. (1957). Forecasting seasonals and trends by exponentially weighted moving averages, ONR Memorandum (Vol. 52), Pittsburgh, PA: Carnegie Institute of Technology. Available from the Engineering Library, University of Texas at Austin.
- Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3), 324-342.
- Alonso C., J. C. & Arcos, M. A. (2006). 4 Hechos Estilizados de las series de rendimientos: Una ilustración para Colombia. *Estudios Gerenciales, En imprenta*.
- Alonso, J. C. & Torres, G. (2014). Características estadísticas del índice general de la Bolsa de Valores de Colombia (IGBC) en sus primeros 10 años. *Journal of Economics Finance and Administrative Science*, 19(36), 45-54. Recuperado desde <http://www.sciencedirect.com/science/article/pii/S2077188614000031>
- Alonso, J. C. & Montenegro, S. (2015). Estudio de Monte Carlo para comparar 8 pruebas de normalidad sobre residuos de mínimos cuadrados ordinarios en presencia de procesos autorregresivos de primer orden. *Estudios Gerenciales*, 31(In press), 253-265. doi:10.1016/j.estger.2014.12.003
- Wald, A. & Wolfowitz, J. (1940). On a Test Whether Two Samples are from the Same Population. *The Annals of Mathematical Statistics*, 11(2), 147-162.
- Box, G. E. P. & Pierce, D. A. (1970). Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association*, 65(332), 1509-1526.
- Ljung, G. M. & Box, G. E. P. (1979). The Likelihood Function of Stationary Autoregressive-Moving Average Models. *Biometrika*, 66(2), 265-270.
- Hyndman, R. J. & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3), 1-22. Recuperado desde <http://www.jstatsoft.org/article/view/v027i03>
- de Lacalle, J. L. (2019). *uroot: Unit Root Tests for Seasonal Time Series*. R package version 2.0-9.1. Recuperado desde <https://CRAN.R-project.org/package=uroot>
- R Core Team. (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. Recuperado desde <https://www.R-project.org/>
- Stoffer, D. (2017). *astsa: Applied Statistical Time Series Analysis*. R package version 1.8. Recuperado desde <https://CRAN.R-project.org/package=astsa>
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. Recuperado desde <http://ggplot2.org>
- Trapletti, A. & Hornik, K. (2018). *tseries: Time Series Analysis and Computational Finance*. R package version 0.10-46. Recuperado desde <https://CRAN.R-project.org/package=tseries>
- Hirotsugu AKAIKE. (1981). Likelihood of a Model and Information Criteria. *Journal of Econometrics*, 16, 3-14.
- Schwarz, G. E. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2), 461-464. doi:10.1214/aos/1176344136
- Hannan, E. J. & Quinn, B. G. (1979). The Determination of the Order of an Autoregression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2), 190-195. doi:10.1111/j.2517-6161.1979.tb01072.x

- Diebold, F. X. & Mariano, R. S. (1995). Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*, 13(3), 253-263. Recuperado desde <http://www.jstor.org/stable/1392185>
- Harvey, D., Leybourne, S. & Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of forecasting*, 13(2), 281-291.
- Shang, H. L. (2013). ftsa: An R Package for Analyzing Functional Time Series. *The R Journal*, 5(1), 64-72. Recuperado desde <https://journal.r-project.org/archive/2013-1/shang.pdf>
- Bates, J. M. & Granger, C. W. J. (1969). The Combination of Forecast. *Journal of the Operational Research Society*, 20(4), 451-468. doi:<https://doi.org/10.1057/jors.1969.103>
- Weiss, C. E., Roetzer, G. R. & Raviv, E. (2018b). *ForecastComb: Forecast Combination Methods*. R package version 1.3.1. Recuperado desde <https://CRAN.R-project.org/package=ForecastComb>
- Weiss, C. E., Raviv, E. & Roetzer, G. (2018a). Forecast Combinations in R using the ForecastComb Package. *The R Journal*, 10(2), 262-281. doi:10.32614/RJ-2018-052
- Newbold, P. & Granger, C. W. J. (1974). Experience with Forecasting Univariate Time Series and the Combination of Forecasts. *Journal of the Royal Statistic Society. Series A (General)*, 137(2), 131-165. doi:<https://doi.org/10.2307/2344546>
- Aiolfi, M. & Timmermann, A. (2006). Persistence in forecasting performance and conditional combination strategies. *Journal of Econometrics*, 135(1-2), 31-53. doi:10.1016/j.jeconom.2005.07.015
- Crane, D. B. & Crotty, J. R. (1967). A Two-Stage Forecasting Model: Exponential Smoothing and Multiple Regression. *Management Science*, 13(8), B-501-B-507. doi:10.1287/mnsc.13.8.b501
- Elliott, G., Gargano, A. & Timmermann, A. (2013). Complete subset regressions. *Journal of Econometrics*, 177(2), 357-373. doi:10.1016/j.jeconom.2013.04.017
- Hsiao, C. & Wan, S. K. (2014). Is there an optimal forecast combination? *Journal of Econometrics*, 178(PART 2), 294-309. doi:10.1016/j.jeconom.2013.11.003
- Shaub, D. & Ellis, P. (2019). *forecastHybrid: Convenient Functions for Ensemble Time Series Forecasts*. R package version 4.2.17. Recuperado desde <https://CRAN.R-project.org/package=forecastHybrid>
- Gaillard, P. & Goude, Y. (2016). *opera: Online Prediction by Expert Aggregation*. R package version 1.0. Recuperado desde <http://pierre.gaillard.me/opera.html>
- Pfaff, B. (2008). VAR, SVAR and SVEC Models: Implementation Within R Package vars. *Journal of Statistical Software*, 27(4). Recuperado desde <http://www.jstatsoft.org/v27/i04/>
- Edgerton, D. & Shukur, G. (1999). Testing autocorrelation in a system perspective testing autocorrelation. *Econometric Reviews*, 18(4), 343-386. doi:10.1080/07474939908800351
- Granger, C. W. J. (1969). Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3), 424-438.

Índice alfabético

analítica descriptiva, 4, 13, 30

analítica diagnóstica, 5

analítica predictiva, 6, 13

analítica prescriptiva, 6

ARIMAX11, 23

autocorrelación muestral, 68

bootstrapping, 167

componente aleatorio, 9

descomposición aditiva, 12, 17

descomposición multiplicativa, 13, 21

desestacionalización, 22, 30

desestacionalizar, 19, 21

estacionalidad, 9

factor cíclico, 9, 11

función accuracy, 49

función as.Date, 16

función auto.combine, 207

función comb.BG, 197

función comb.CLS, 199

función comb.CSR, 200

función comb.EIG1, 200

función comb.EIG2, 200

función comb.EIG3, 200

función comb.EIG4, 201

función comb.InvW, 198

función comb.LAD, 199

función comb.MED, 196

función comb.NG, 198

función comb.OLS, 199

función comb.SA, 196

función comb.TA, 197

función comb.WA, 197

función decompose, 16

función forecast, 167

función foreccomb, 204

función predict, 208

función remainder, 24

función rolling.combine, 207

función seas, 23, 28

función seasadj, 24

función seasonal, 24

función simulate, 171

función trendcycle, 24

función ts, 14, 16

función window, 34

función xts, 16

modelo Holt, 33

modelo Holt-Winters, 33

modelo SES, 33

MSE, 35

muestra de estimación, 33

muestra de evaluación, 33

objeto seas, 24

objeto ts, 14

objeto xts, 14

objeto zoo, 14

paquete forecast, 49, 167, 171

paquete ForecastComb, 195–201, 204

paquete forecastHybrid, 201, 209

paquete opera, 201

paquete seasonal, 23, 28

paquete xts, 16

promedio móvil, 33

prueba de Box-Pierce, 67

prueba de Ljung-Box, 67

prueba de Rachas, 66

quantile, 171

RMSE, 35

serie de tiempo, 7

tendencia, 9, 10