Trabajo No 2 de Estadística III - 3009137
Análisis de Serie de Tiempo: Modelos de regresión con errores estructurales ARMA(p,q) estacionarios

Índice

1. Características del Trabajo **1**

2. Puntos a Desarrollar **1**

3. Programación R **5**

4. Los Grupos para el Trabajo y Asignaciones **5**

5. Exportar tablas de parámetros estimados y las predicciones a archivo .csv **7**

5.1. En modelos ajustados con función R `Arima()` 7

5.2. En modelos ajustados y pronosticados con la función de usuario `regexpo.ErrorARMA()`. 7

APÉNDICE A. Métodos de Identificación automática de modelos ARMA en los errores de MR en R **10**

APÉNDICE B. Ajustes, validación supuestos y pronósticos en R de MR con errores ARMA **14**

APÉNDICE C. Gráficos función armasubsets sobre residuos de ajuste del modelo global en la Tabla 1 **25**

1. Características del Trabajo

Este trabajo es una continuación del Trabajo No 1. Consiste en ajustar con validación cruzada uno de los modelos de regresión global para la serie, escogido entre los dos que fueron propuestos en el trabajo 1 (ver en la Tabla 1 cuál de los dos modelos deben considerar), luego, validar supuesto de ruido blanco para los errores estructurales y determinar en caso de no ser ruido blanco, si estos errores definen un proceso estacionario de media cero e identificar posibles modelos, esto último, suponiendo estacionariedad en covarianza y media cero al usar los diferentes métodos de identificación. De acuerdo a la asignación en Tabla 1, cada grupo deberá considerar cuatro modelos de regresión manteniendo misma estructura de regresión en la tendencia y estacionalidad formulada en el modelo global pero con errores estructurales ARMA (recuerde que el término ARMA también incluye a procesos AR y MA como casos particulares pues un AR(p)=ARMA(p,0) y un MA(q) es un ARMA(0,q)), con el fin de ajustarlos, validar sus supuestos y compararlos en ajuste y pronóstico. Finalmente, el mejor modelo de regresión con errores ARMA deberá ser comparado con el mejor de los dos modelos de ajuste local del trabajo 1, al cual también deberá evaluar supuestos sobre sus errores de ajuste.

2. Puntos a Desarrollar

La presentación de la solución de los puntos a desarrollar y que se enuncian a continuación, deberá acomodarse al formato y al contenido de Secciones descrito en la plantilla de los trabajos del curso (descargar de moodle el archivo *PlantillaTrabajosv04.docx*) y ***máximo número de páginas, 18***.

1. Introducción: Presente nuevamente la definición DANE de la serie asignada. Luego, deberá relatar brevemente los resultados alcanzados en el trabajo 1: Cuáles modelos globales y locales fueron propuestos (debe dar las ecuaciones teóricas), cuál es el mejor modelo global, qué logró explicar este modelo con relación a los patrones que fueron observados sobre la serie, y si resultó mejor el ajuste y pronóstico global vs. lo local (modelos 3 y 4 en el trabajo 1) y por qué.

2. Análisis descriptivo de la serie, modelo global asignado y sus resultados:

   a) Presente y analice **en forma breve** la gráfica de la serie (y su logaritmo natural si la serie es multiplicativa) indicando los patrones observables de tendencia, estacionalidad, varianza, ciclos. Grafique y analice además la ACF estimada con la serie (para el caso multiplicativo sólo presente y analice la ACF del logaritmo natural) con $m = 36$ y concluya en términos de estacionariedad o no y por qué, contrastando además con lo que a partir de la gráfica de la serie se concluye al respecto.

   b) Para el modelo de regresión global señalado en la Tabla 1, reporte la ecuación teórica con sus supuestos y con la estrategia de validación cruzada usando la misma longitud de ajuste del trabajo anterior ($n = 113$), ajuste nuevamente este modelo y reporte los resultados de ajuste ***(solo la tabla de parámetros estimados, medidas de ajuste, gráfico del ajuste)*** y pronósticos ***(solo la tabla con las medidas de cobertura, amplitud media de los I.P, MAE, MAPE y RMSE)***. Dé una conclusión breve sobre la calidad del ajuste y de los pronósticos con este modelo. Use $\exp(C_n^*(p))$ para el cálculo de AIC y BIC.

3. Evaluación del supuesto de ruido blanco e identificación de procesos estocásticos sobre los errores estructurales del modelo global:

   a) Validación de supuestos: Guarde los residuos estructurales $\widehat{E}_t$ en la escala en que ajustó la serie. Analice inicialmente las gráficas de estos residuales en términos de los supuestos sobre los errores E_t: media constante en cero, varianza constante, y determine si hay ciclos evidentes no explicados, rachas en signos $\pm$ o cualquier otro patrón en el tiempo, y qué concluye frente a la existencia de estos patrones. Realice las pruebas de incorrelación con: Ljung-Box, Durbin-Watson (recuerde que no aplica si el modelo no es lineal en parámetros de la regresión), y gráficas de la ACF y PACF con límites de Bartlett. Concluya sobre si los errores estructurales E_t son ruido blanco o no y en este último caso, evalúe si por lo menos son estacionarios con media cero. Use $m = 36$ en ACF, PACF y $m = 6, 12, 18, 24, 30$ y 36 en Ljung-Box.

   b) Identificación de posibles modelos para los errores estructurales E_t: En caso de la existencia de correlaciones y suponiendo que media cero, varianza constante y ergodicidad se cumplen en los errores estructurales, determine cuáles modelos ARMA provisionales (recuerde de nuevo que procesos AR y MA son casos particulares de los ARMA), podrían ser los adecuados para el error estructural, con base en ***acf(), pacf(), auto.arima(), eacf(), SelectModel(), armasubsets(). Para los métodos automáticos y la EACF debe presentar un print-screen de la consola R donde sea visible la línea de programa R ejecutada junto con la salida que aparece en la consola R***. En cada caso debe dar la ecuación teórica del modelo ARMA identificado para los errores estructurales y presentar la salida R original que arroje la función de identificación y gráficas dónde aplique.

      - la ACF y la PACF.
      - La EACF. Realizar la gráfica con orden máximo p, q iguales a 24.
      - `SelectModel()` cargada desde repositorio en github, úsela sólo en caso de identificar un patrón tipo cola en ACF. Tenga en cuenta que esta función sólo identifica procesos AR(p). Use un p máximo de 36, variando además, el argumento `criterion=" AIC"` y con `criterion="BIC"`. Con cada criterio puede estar obteniendo modelos AR(p) distintos.
      - `auto.arima()` de la librera forecast. Use esta función sobre el vector de residuales `residuals(modelo)`, y sobre el objeto de serie de tiempo creado con los residuales con la función `ts()`, usando la misma frecuencia

y fecha de inicio de la serie asignada. En cada caso, varíe también el argumento `ic=" aic"` e `ic="bic"`. De esta forma es posible obtener máximo cuatro modelos distintos.

- La función `armasubsets()` de la librería `TSA`, usando criterio BIC y argumento `ar.method="ml"`, sobre residuos estructurales, tomando el modelo en el renglón que se indica en la Tabla 1. Además, si en la Tabla 1 se indica agregar algún parámetro, complete la ecuación del modelo ingresando los términos correspondientes. Tenga en cuenga que en los modelos identificados con esta función algunos coeficientes deben ser fijados en cero. Verifique que la gráfica que obtiene con su programación R es igual a la que se exhibe en el Apéndice C de esta guía (identifique la respectiva gráfica en el Apéndice C con el nombre de la serie: Datos1, Datos2, etc.)

NOTA 1: Verifique que en todos los modelos que identifique la función auto.arima sean ARIMA(p, d, q) con $d = 0$ ó ARIMA$(p, d, q) \times (P, D, Q)_{[12]}$, con $d = 0$ y $D = 0$, lo cual indica un proceso estacionario, y verifique también que sean procesos con media cero. Por otra parte, como es posible que con la función auto.arima() se identifiquen procesos ARMA estacionales estacionarios o ARMA$(p, q) \times (P, Q)_{[12]}$, estos también deben ser considerados. Ver en Apéndice A la explicación acerca de la estructura y la ecuación de este tipo de modelos.

NOTA 2: Con métodos de identificación diferentes a ACF-PACF, recuerde que no se deben considerar como candidatos modelos que no son consistentes con lo que ya se ha identificado con ACF-PACF; por ejemplo, si es muy claro que la ACF es cola, entonces no son admisibles procesos MA(q), ni MA(Q)[12], y si además, la ACF es tipo cola exponencial-sinusoidal, sólo son admisibles procesos AR(p) y ARMA(p,q), con $p \geq 2$, y procesos ARMA(p,q)(P,Q)[12] tales que $(p + 12 \times P) \geq 2$.

4. Modelos de regresión global con errores estructurales E_t ARMA: Considerando misma estructura de regresión del modelo global, pero errores estructurales según los modelos que se indican en la Tabla 1, dé las ecuaciones teóricas con sus supuestos y ajústelos con validación cruzada, usando el n del trabajo 1. ***Tenga en cuenta para la definición de estos modelos, tanto en sus ecuaciones teóricas, ajustadas y de predicción y para validar sus supuestos, la actividad de lectura señalada en el ítem 9.1 del programa de la asignatura (documento en Moodle: Definición modelo de regresión con errores ARMA)***. Tengan en cuenta también que es posible que algunos o todos los modelos propuestos en la Tabla 1 para el error estructural no coincidan con los identificados en 3-b), sino que son una modificación obtenida tras probar con modelos preliminares y chequear que no hay validez del supuesto de RB por corelaciones no nulas, por lo que resultó necesario hacer cambios en algunos de ellos. Tenga en cuenta además lo siguiente:

   a) En modelos de regresión lineal múltiple (sobre Y_t ó $\log(Y_t)$, según sea el caso) con errores estructurales ARMA, debe ajustar juntas tendencia, estacionalidad y errores E_t ARMA, usando la función `Arima()` de la librería `forecast`; para ello, en esta función se debe usar el argumento `xreg` igual a una matriz X cuyas columnas son los valores de las variables predictoras en el modelo global, matriz que debe crearse previamente. También, en la función `Arima()` se deben definir los órdenes del ARMA para el error E_t, usando los argumentos correspondientes en esta función. En modelos de regresión exponenciales con errores estructurales ARMA, debe ajustar y obtener los pronósticos (solo puntuales, no hay posibilidad de pronósticos por I.P) usando la función de usuario `regexpo.ErrorARMA()` disponible en el repositorio de github del curso; esta función también va a requerirle la matrix X de predictores para el ajuste, la matriz con los valores de los predictores en el pronóstico y un vector con los nombres de los parámetros en la función de regresión exponencial.

   b) Recuerde además que si ajusta sobre $\log(Y_t)$, debe calcular $\widehat{Y}_t \approx \exp\left(\widehat{\log(Y_t)}\right) \times \exp(\hat{\sigma}_a^2/2)$ y reportar el valor del factor de corrección por transformación lognormal, donde $\hat{\sigma}_a^2$ es la varianza estimada del error de ajuste a_t del modelo, la cual se invoca en R como `modelo1$sigma2`, `modelo2$sigma2`, `modelo3$sigma2`, `modelo4$sigma2`, donde `modelo1, modelo2, modelo3, modelo4`, son objetos R donde se ha guardado el resultado del ajuste de los modelos 1 a 4 usando la función R `Arima()`.

c) Para la programación R estudie los ejemplos presentados en el Apéndice B de esta guía, Secciones I, II y III para modelos con ecuación de regresión lineal en los parámetros de tendencia y estacionalidad (modelos polinomiales estacionales y modelos log polinomiales estacionales, con error estructural ARMA) y Sección IV para modelos de regresión con estructura no lineal en los parámetros de tendencia y estacionalidad (modelos exponenciales polinomial estacional con error estructural ARMA).

NOTA 3: Para todos los modelos presente las tablas de parámetros estimados, las gráficas de ajustes, tabla de medidas de ajuste, pero la ecuación ajustada solo para el modelo 3. Use $\exp(C_n^*(p))$ para AIC y BIC en todos los modelos. También se deben formular y realizar los tests de significancia de parámetros de interés en la tendencia y en la estacionalidad y de ***todos*** los parámetros en el modelo del error estructural, además, en modelos de regresión lineal ***(es decir, polinomiales estacionales y log polinomiales estacionales)*** usando indicadoras en la componente estacional, debe interpretar y comparar las estimaciones de los parámetros estacionales como se hizo en el trabajo 1 del curso (consulte en asesoría cómo realizar las gráficas pertinentes).

5. Análisis de residuales y validación de supuestos: Realice sobre los residuos de ajuste $\widehat{a}_t$ en los modelos de regresión con errores E_t ARMA, los análisis de las respectivas gráficas, de manera comparativa, como también indicar si hay o no evidencia fuerte de carencia de ajuste, e indique cuáles modelos de los ajustados presentan mejor comportamiento en sus residuales de ajuste $\widehat{a}_t$. Recuerde que con estos residuos se desea evaluar para los errores de ajuste a_t si no hay evidencia muestral fuerte en contra de media es cero y varianza constante y además chequear ausencia de patrones contrarios a la independencia. Haga la validación de supuestos sobre los errores de ajuste a_t: Tests ACF, PACF, Ljung-Box (con el mismo m usado en el punto 3-a) de esta guía), Test de normalidad y gráfico de normalidad (estos dos últimos sólo si no se detectan correlaciones entre los errores de ajuste a_t). Concluya acerca de la validez del supuesto de que los errores de ajuste a_t definen un proceso de ruido blanco con distribución normal.

6. Pronósticos para la validación cruzada: Para todos los modelos asignados presente, interprete, analice y compare los pronósticos puntuales y por intervalos del 95 % (recuerde que I.P no son posibles en modelos exponenciales con error estructural ARMA), y la gráfica comparativa de los valores reales y pronósticos puntuales, pero respecto a la ecuación de los pronósticos puntuales, presente solo la del modelo 3. Tenga en cuenta que para modelos ajustados sobre logaritmo de la serie también es necesario aplicar el factor de corrección por transformación lognormal al traer valores pronosticados a la escala original, es decir, $\widehat{Y}_n(L) \approx \exp\left(\widehat{\log(Y_n(L))}\right) \times \exp(\hat{\sigma}_a^2/2)$ y de la misma forma proceder con los intervalos de predicción.

7. Comparación de modelos de regresión con errores ARMA vs. mejor local: Para el mejor modelo local entre los dos considerados en el trabajo 1, deberá presentar:

   - Gráfica de su ajuste, y valores de los criterios de información AIC y BIC,
   - Gráficos de sus residuales,
   - Medidas de precisión de los pronósticos puntuales (MAE, MAPE, RMSE) y por intervalos del 95 % de confianza (amplitud media y cobertura de los IP, si los hay),
   - Evaluación del supuesto de ruido blanco (tests ACF y PACF) para sus errores de ajuste E_t y test de normalidad (si no rechaza en tests ACF y PACF).
   - Compare el mejor modelo de regresión con error ARMA con el modelo local, tanto en calidad de ajustes, pronósticos y validez de supuestos.

8. Elabore conclusiones: En esta sección debe presentar:

   - Un resumen de los resultados encontrados en el respectivo trabajo,
   - Enunciar los problemas enfrentados en la modelación,

- Postular cuál ha sido el mejor modelo en ajuste y pronóstico entre los tratados y comentar acerca de lo que usted crea que logró este mejor modelo: ¿capturó la dinámica de la serie: Su tendencia, estacionalidad y sus variaciones cíclicas son bien ajustadas?, ¿Los pronósticos parecen realistas y confiables?, ¿Críticas al mejor modelo que encontró en el trabajo actual?
- Exprese claramente qué recomienda para la serie en cuanto a ajustes globales con errores estructurales ARMA o locales según lo realizado hasta el momento, considerando sobre todo qué tan bien representa cada modelo los patrones observados en la serie.

3. Programación R

Ver el siguiente material para la programación R necesaria.

- Taller 7 de monitoría sobre validación de supuestos e identificación de procesos estacionarios y el Apéndice A de esta guía "Uso de métodos de Identificación automática de modelos ARMA en los errores estructurales de modelos de regresión en R".
- Para el ajuste, pronósticos y validación de supuestos de los modelos 1 a 4:
    - En modelos de regresión lineal sobre Y_t o sobre $\log(Y_t)$, con errores estructurales ARMA, remitirse a los ejemplos en el Apéndice B de esta guía, así: Sección I para casos como modelos 1 y 2; Sección II para casos como el modelo 4 y Sección III para casos como el modelo 3.
    - La Sección IV es solo para modelos exponenciales con errores estructurales ARMA, allí ver literal **a.** para casos como modelos 1 y 2; literal **b.** para casos como el modelo 4 y literal **c.** para casos como el modelo 3.
- Recuerde que si está modelando a $\log(Y_t)$, deberá luego obtener ajustes y pronósticos en la escala original.

4. Los Grupos para el Trabajo y Asignaciones

De acuerdo a los grupos y series definidas en el primer trabajo. El modelo de regresión global con error estructural RB y con errores ARMA que deberán trabajar, son los siguientes:

Tabla 1: Modelo global y modelos con error estructural ARMA

					Modelos de regresión con error estructural ARMA			
					modelo 1	modelo 2	modelo 3	modelo 4
serie	modelo global	estacionalidad	polinomio	EACF	Error	Error	Error	Error según armasubsets (1)
Datos1	exponencial polinomial estacional	trigon. $F_j = j/12$, $j = 1, \ldots, 5$	2	24x24	AR(18)+	ARMA(4,5)+	ARMA(4,1)(1,1)[12]+	12x12, renglón 1 y agregar θ_5,+
Datos2	log polinomial estacional	trigon. $F_j = j/12$, $j = 1, \ldots, 6$	3	24x24	AR(10)	ARMA(2,4)	ARMA(3,0)(0,2)[12]	12x12, renglón 4
Datos3	log polinomial estacional	trigon. $F_j = j/12$, $j = 1, \ldots, 6$	3	24x24	AR(10)	ARMA(2,4)	ARMA(3,0)(0,2)[12]	12x12, renglón 3
Datos4	log polinomial estacional	trigon. $F_j = j/12$, $j = 1, \ldots, 6$	3	24x24	AR(10)	ARMA(4,3)	ARMA(3,3)(1,2)[12]	12x12, renglón 4 y agregar ϕ_{10}
Datos5	log polinomial estacional	trigon. $F_j = j/12$, $j = 3, 4, 5$	5	24x24	AR(15)	ARMA(7,7)	ARMA(6,2)(1,1)[12]	18x18, renglón 1, agregar ϕ_{13}, **
Datos6	log polinomial estacional	Indicadoras	3	24x24	AR(16)	ARMA(3,1)	ARIMA(4,0)(1,2)[12]	24x24, renglón 1 e incluir a ϕ_1
Datos7	exponencial polinomial estacional	Indicadoras	1	24x24	AR(17)+	ARMA(14,10)++	ARMA(5,5)(0,1)[12]+	12x12, renglón 4 y agregar ϕ_{10},+

					Modelos de regresión con error estructural ARMA			
					modelo 1	modelo 2	modelo 3	modelo 4
serie	modelo global	estacionalidad	polinomio	EACF	Error	Error	Error	Error según armasubsets **(1)**
Datos8	Polinomial estacional	Indicadoras	3	24x24	AR(15)	ARMA(2,10)	ARMA(2,4)(1,1)[12]	18x18, renglón 6 y agregar ϕ_1, ϕ_{10}
Datos9	log polinomial estacional	Trigon. $F_j = j/12$, $j = 2, 3, 4, 5$	6	24x24	AR(11)	ARMA(3,3)$*$	ARMA(1,3)(1,1)[12]$*$	18x18, renglón 6 y agregar ϕ_1
Datos10	log polinomial estacional	Trigon. $F_j = j/12$, $j = 2, 3, 4, 5$	4	24x24	AR(15)	ARMA(9,7)	ARMA(1,0)(2,0)[12]	24x24, renglón 1
Datos11	log polinomial estacional	Indicadoras	3	24x24	MA(1)	ARMA(7,1)	ARMA(7,2)(0,1)[12]	12x12, renglón 4
Datos12	exponencial polinomial estacional	Trigon. $F_j = j/12$, $j = 2, 3, 4, 5$	3	24x24	AR(8)+	ARMA(5,5)+	ARMA(5,0)(0,1)[12]+	12x12, renglón 4,+
Datos13	Polinomial estacional	Indicadoras	4	24x24	MA(11)	ARMA(1,11)	ARMA(0,10)(0,1)[12]	12x12, renglón 6 y agregar θ_1 y θ_9
Datos14	log polinomial estacional	Indicadoras	3	24x24	AR(10)	ARMA(5,3)	ARMA(3,0)(0,1)[12]	12x12, renglón 1 y agregar θ_4
Datos16	exponencial polinomial estacional	Trigon. $F_j = j/12$, $j = 1, 2, 3, 4, 5, 6$	2	24x24	AR(10)+	ARMA(2,4)+	ARMA(5,5)(0,1)[12]+	18x18, renglón 1 y agregar ϕ_2 y θ_{13},+
Datos17	log polinomial estacional	Indicadoras	2	24x24	AR(10)	ARMA(2,4)	ARMA(3,4)(0,1)[12]	12x12, renglón 3 y agregar θ_4
Datos18	log polinomial estacional	Indicadoras	2	24x24	AR(10)	ARMA(5,4)	ARMA(1,4)(2,0)[12]	12x12,renglón 1 y agregar ϕ_4
Datos19	Polinomial estacional	trigon. $F_j = j/12$, $j = 3, 4, 5$	3	24x24	AR(13)	ARMA(1,2)	ARMA(1,0)(0,1)[12]	18x18, renglón 5
Datos 20	log polinomial estacional	Indicadoras	3	24x24	AR(16)	ARMA(3,1)	ARMA(3,0)(1,1)[12]	24x24, renglón 5, agregar ϕ_1, $***$
Datos22	Polinomial estacional	Indicadoras	3	24x24	AR(16)	ARMA(2,10)	ARMA(4,8)(1,1)[12]	12x12, renglón 3 y agregar ϕ_5, ϕ_6
Datos23	Polinomial estacional	Trigon. $F_j = j/12$, $j = 2, 3, 4, 5$	5	24x24	MA(3)	ARMA(3,2)	ARMA(0,3)(1,1)[12]	24x24, renglón 1 y agregar θ_{22}
Datos24	Polinomial estacional	Trigon. $F_j = j/12$, $j = 2, 3, 4, 5$	4	24x24	AR(15)	ARMA(9,5)	ARMA(1,0)(2,0)[12]	18x18, renglón 1 y agregar θ_5
Datos27	Polinomial estacional	Indicadoras	3	24x24	MA(10)	ARMA(1,12)	ARMA(2,1)(0,1)[12]	12x12, renglón 2 y agregar θ_1

Observaciones:

(1) En la función armasubsets aplicada sobre los residuos estructurales del modelo global, debe usar argumento ar.method="ml", y cada grupo deberá comparar la gráfica obtenida con la que se muestra en el Apéndice C de esta guía. Para tableros 12x12 se fijan argumentos nar=12 y nma=12; para tableros 18x18 se fijan argumentos nar=18 y nma=18 y para tableros 24x24 se fijan argumentos nar=24 y nma=24.

En los casos donde aparezca alguno de los siguientes símbolos, tener en cuenta lo que se indica a continuación:

$*$ Deben ser ajustados fijando los valores estimados de los parámetros de tendencia y estacionalidad en los valores hallados en el modelo global. Consultar en asesoría cómo hacerlo, pero previamente, prepare la programación como si fuera a correr el modelo tal como explican los ejemplos en el Apéndice B.

$**$ Usar en `Arima()` argumento `method=`"CSS-ML".

$***$Debe ser ajustados fijando los valores estimados de los parámetros de tendencia y estacionalidad en los valores hallados en el modelo global. Consultar en asesoría cómo hacerlo, pero previamente, prepare la programación como si fuera a correr el modelo tal como explican los ejemplos en el Apéndice B, además debe estimar en `Arima()` por `method=`"CSS-ML"

+ Se ajusta con la función de usuario `regexpo.ErrorARMA()`.

++ Se ajusta con la función de usuario `regexpo.ErrorARMA()` y en la cual debe usarse el argumento `optim.method=`"Nelder"

5. Exportar tablas de parámetros estimados y las predicciones a archivo .csv

5.1. En modelos ajustados con función R `Arima()`

Suponga que desea exportar la tabla de parámetros estimados y las predicciones de un modelo de regresión con error ARMA, ajustado mediante función `Arima()` bajo un objeto R de nombre `modelo2`, por ejemplo, un modelo log cuadrático estacional con trigonométricas en frecuencias $F_j = j/12$, con $j = 1, \ldots, 6$, y error estructural ARMA(2,4) (previamente se ha definido la matriz de predictores `X`), cuyo ajuste es guardado en un objeto R de nombre `modelo2` y sus pronósticos en un objeto R de nombre `predmod2`, ilustrados en la Figura 1.

```
> modelo2=Arima(log(yt),order=c(2,0,4),xreg=as.matrix(X),method="ML")
> k2=length(coef(modelo2)[coef(modelo2)!=0])
> df2=n-k2
> coeftest(modelo2,df=df2)
t test of coefficients:

             Estimate  Std. Error  t value  Pr(>|t|)
ar1        4.5514e-01  2.1536e-01   2.1133 0.0372498 *
ar2       -5.4714e-01  1.3467e-01  -4.0627 0.0001012 ***
ma1        3.1999e-01  2.0080e-01   1.5936 0.1144143
ma2        5.2833e-01  1.1002e-01   4.8021 5.989e-06 ***
ma3        5.0149e-01  8.2648e-02   6.0678 2.777e-08 ***
ma4        4.5950e-01  8.9936e-02   5.1092 1.718e-06 ***
intercept  4.3889e+00  3.3174e-02 132.2991 < 2.2e-16 ***
t          1.0519e-03  1.0865e-03   0.9682 0.3354714
t2         1.6728e-05         NaN      NaN       NaN
sen1      -4.1392e-02  1.4831e-02  -2.7909 0.0063786 **
cos1       6.4412e-02  1.2553e-02   5.1314 1.568e-06 ***
sen2       7.5420e-04  4.9889e-03   0.1512 0.8801644
cos2       5.1537e-02  5.0336e-03  10.2387 < 2.2e-16 ***
sen3      -2.6501e-02  9.6629e-03  -2.7425 0.0073134 **
cos3       4.3191e-02  9.8339e-03   4.3921 2.970e-05 ***
sen4      -5.0633e-03  4.4185e-03  -1.1460 0.2547539
cos4       4.9016e-02  4.5915e-03  10.6753 < 2.2e-16 ***
sen5       1.8104e-02  2.9247e-03   6.1901 1.607e-08 ***
cos5       4.7793e-02  2.9255e-03  16.3365 < 2.2e-16 ***
cos6       9.9049e-03  2.7590e-03   3.5900 0.0005303 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> predmod2=ts(predmod2,freq=12,start=c(2022,6))
> predmod2
         Point Forecast     Lo 95    Hi 95
Jun 2022       113.9436 102.11858 127.1380
Jul 2022       119.7237 104.22540 137.5266
Aug 2022       116.8874 101.27421 134.9076
Sep 2022       113.2281  97.88942 130.9702
Oct 2022       116.3426  99.99309 135.3654
Nov 2022       120.8098 103.82306 140.5759
Dec 2022       152.3520 130.75024 177.5226
Jan 2023       113.4122  97.28452 132.2136
Feb 2023       109.3286  93.77188 127.4663
Mar 2023       116.9197 100.25494 136.3546
Apr 2023       106.3734  91.21144 124.0557
May 2023       112.8991  96.80047 131.6750
```

Figura 1: Vista salidas R: tabla de parámetros estimados y pronósticos de un MRLM con error estructural ARMA estacionario de media cero.

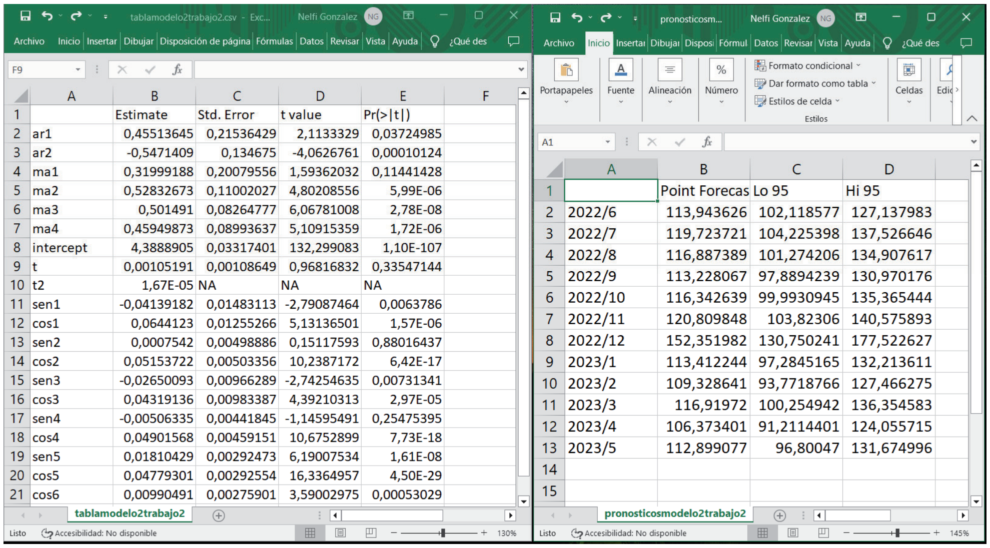
Proceda como se ilustra a continuación usando la función R `write.csv2()`,

```
#Exportando a excel tabla de parámetros estimados
write.csv2(coeftest(modelo2,df=df2),file="C:/Documents/trabajo2/tablamodelo2trabajo2.csv",row.names = TRUE)

#Exportando a excel pronosticos
write.csv2(predmod2,file="C:/Documents/trabajo2/pronosticosmodelo2trabajo2.csv",row.names = paste(trunc(time(predmod2)),cycle(predmod2),sep="/"))
```

donde se ha especificado guardar la tabla de parámetros estimados en un archivo de nombre `tablamodelo2trabajo2.csv` y la tabla de pronósticos en un archivo de nombre `pronosticosmodelo2trabajo2.csv`, ambos en la ruta de archivos `C:/Documents/trabajo2/`. Adapte ruta y nombre de archivo como sea necesario. Los archivos resultantes se ilustran en la Figura 2.

5.2. En modelos ajustados y pronosticados con la función de usuario `regexpo.ErrorARMA()`.

Suponga ahora que se desea exportar la tabla de parámetros estimados y los pronósticos de un modelo de regresión exponencial con error ARMA, por ejemplo un exponencial polinomial grado 4, estacional con trigonométricas en frecuencias $F_j = j/12$, para $j = 2, 3, 4, 5$, y con error ARMA(2,2), ajustado y pronosticado de forma aproximada mediante la función de usuario `regexpo.ErrorARMA()`, como se muestra en la Figura 3. Previamente, se han definido la matriz con valores de los predictores en el ajuste (`X`), en el pronóstico (`Xnuevo`) y el vector de nombres para los parámetros en la tendencia y estacionalidad (`param2`). Para exportar estos resultados, proceda de la siguiente manera,



	Estimate	Std. Error	t value	Pr(>|t|)
ar1	0,45513645	0,21536429	2,1133329	0,03724985
ar2	-0,5471409	0,134675	-4,0626761	0,00010124
ma1	0,31999188	0,20079556	1,59362032	0,11441428
ma2	0,52832673	0,11002027	4,80208556	5,99E-06
ma3	0,501491	0,08264777	6,06781008	2,78E-08
ma4	0,45949873	0,08993637	5,10915359	1,72E-06
intercept	4,3888905	0,03317401	132,299083	1,10E-107
t	0,00105191	0,00108649	0,96816832	0,33547144
t2	1,67E-05	NA	NA	NA
sen1	-0,04139182	0,01483113	-2,79087464	0,0063786
cos1	0,0644123	0,01255266	5,13136501	1,57E-06
sen2	0,0007542	0,00498886	0,15117593	0,88016437
cos2	0,05153722	0,00503356	10,2387172	6,42E-17
sen3	-0,02650093	0,00966289	-2,74254635	0,00731341
cos3	0,04319136	0,00983387	4,39210313	2,97E-05
sen4	-0,00506335	0,00441845	-1,14595491	0,25475395
cos4	0,04901568	0,00459151	10,6752899	7,73E-18
sen5	0,01810429	0,00292473	6,19007534	1,61E-08
cos5	0,04779301	0,00292554	16,3364957	4,50E-29
cos6	0,00990491	0,00275901	3,59002975	0,00053029

	Point Forecas	Lo 95	Hi 95
2022/6	113,943626	102,118577	127,137983
2022/7	119,723721	104,225398	137,526646
2022/8	116,887389	101,274206	134,907617
2022/9	113,228067	97,8894239	130,970176
2022/10	116,342639	99,9930945	135,365444
2022/11	120,809848	103,82306	140,575893
2022/12	152,351982	130,750241	177,522627
2023/1	113,412244	97,2845165	132,213611
2023/2	109,328641	93,7718766	127,466275
2023/3	116,91972	100,254942	136,354583
2023/4	106,373401	91,2114401	124,055715
2023/5	112,899077	96,80047	131,674996

Figura 2: Vista archivos `tablamodelo2trabajo2.csv`, `pronosticosmodelo2trabajo2.csv`, en un MRLM con error ARMA.

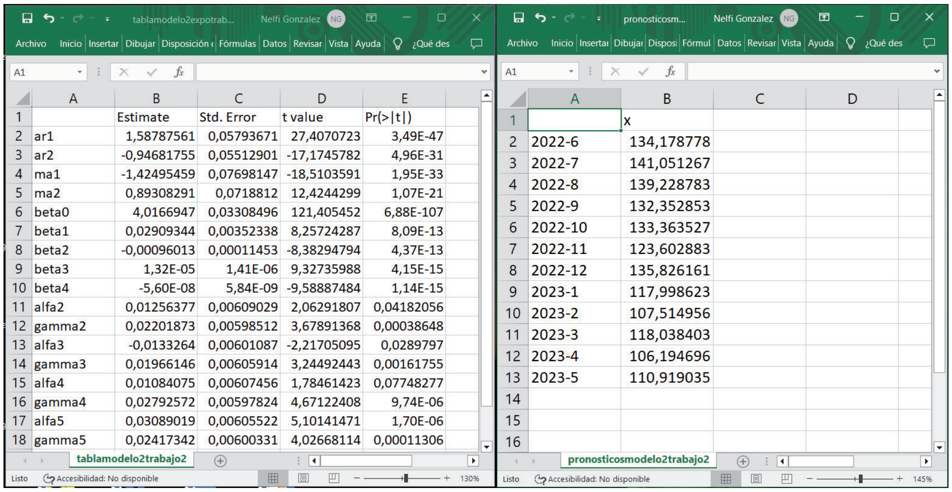
```
#Exportando a excel tabla de parámetros estimados
write.csv2(coef(modelo2),file="C:/Documents/trabajo2/tablamodelo2trabajo2.csv",row.names = TRUE)

#Exportando a excel pronosticos
write.csv2(as.numeric(ytpron2),file="C:/Documents/trabajo2/pronosticosmodelo2trabajo2.csv",
          row.names = paste(trunc(time(ytpron2)),cycle(ytpron2),sep="-"))
```

donde se ha especificado guardar la tabla de parámetros estimados en un archivo de nombre `tablamodelo2trabajo2.csv` y los pronósticos en un archivo de nombre `pronosticosmodelo2trabajo2.csv`, ambos archivos guardados en la ruta `C:/Documents/trabajo2/`. Adapte ruta y nombre de archivos como sea necesario. Los archivos resultantes se ilustran en la Figura 4.

```
> modelo2=regexpo.ErrorARMA(respuesta=yt,names.param=param2,data=X,newdata=Xnuevo,order=c(2,0,2),method="ML")
> coef(modelo2)
            Estimate   Std. Error    t value      Pr(>|t|)
ar1     1.587876e+00 5.793675e-02  27.407055  3.485232e-47
ar2    -9.468175e-01 5.512905e-02 -17.174564  4.963550e-31
ma1    -1.424955e+00 7.698151e-02 -18.510348  1.950893e-33
ma2     8.930828e-01 7.188124e-02  12.424422  1.069918e-21
beta0   4.016695e+00 3.308497e-02 121.405433 6.884599e-107
beta1   2.909343e-02 3.523385e-03   8.257239  8.088369e-13
beta2  -9.601311e-04 1.145339e-04  -8.382944  4.368280e-13
beta3   1.317095e-05 1.412077e-06   9.327356  4.147567e-15
beta4  -5.601061e-08 5.841210e-09  -9.588871  1.137081e-15
alfa2   1.256377e-02 6.090317e-03   2.062909  4.182144e-02
gamma2  2.201872e-02 5.985126e-03   3.678907  3.864912e-04
alfa3  -1.332640e-02 6.010841e-03  -2.217061  2.897901e-02
gamma3  1.966146e-02 6.059142e-03   3.244924  1.617554e-03
alfa4   1.084075e-02 6.074576e-03   1.784610  7.748353e-02
gamma4  2.792571e-02 5.978244e-03   4.671223  9.740641e-06
alfa5   3.089021e-02 6.055219e-03   5.101419  1.697114e-06
gamma5  2.417340e-02 6.003312e-03   4.026677  1.130619e-04
```

```
> ytpron2=modelo2$forecast; ytpron2
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2022                                              134.1788 141.0513 139.2288 132.3529 133.3635 123.6029 135.8262
2023 117.9986 107.5150 118.0384 106.1947 110.9190
```

Figura 3: Vista salidas R:tabla de parámetros estimados y pronósticos de un modelo de regresión exponencial con error ARMA.



	Estimate	Std. Error	t value	Pr(>|t|)
ar1	1,58787561	0,05793671	27,4070723	3,49E-47
ar2	-0,94681755	0,05512901	-17,1745782	4,96E-31
ma1	-1,42495459	0,07698147	-18,5103591	1,95E-33
ma2	0,89308291	0,0718812	12,4244299	1,07E-21
beta0	4,0166947	0,03308496	121,405452	6,88E-107
beta1	0,02909344	0,00352338	8,25724287	8,09E-13
beta2	-0,00096013	0,00011453	-8,38294794	4,37E-13
beta3	1,32E-05	1,41E-06	9,32735988	4,15E-15
beta4	-5,60E-08	5,84E-09	-9,58887484	1,14E-15
alfa2	0,01256377	0,00609029	2,06291807	0,04182056
gamma2	0,02201873	0,00598512	3,67891368	0,00038648
alfa3	-0,0133264	0,00601087	-2,21705095	0,0289797
gamma3	0,01966146	0,00605914	3,24492443	0,00161755
alfa4	0,01084075	0,00607456	1,78461423	0,07748277
gamma4	0,02792572	0,00597824	4,67122408	9,74E-06
alfa5	0,03089019	0,00605522	5,10141471	1,70E-06
gamma5	0,02417342	0,00600331	4,02668114	0,00011306

	x
2022-6	134,178778
2022-7	141,051267
2022-8	139,228783
2022-9	132,352853
2022-10	133,363527
2022-11	123,602883
2022-12	135,826161
2023-1	117,998623
2023-2	107,514956
2023-3	118,038403
2023-4	106,194696
2023-5	110,919035

Figura 4: Vista archivos `tablamodelo2trabajo2.csv`, `pronosticosmodelo2trabajo2.csv`, regresión exponencial con error ARMA.

Referencias

[1] Bowerman, B. L, O'Connell, R. T y Koehler, A. B. (2009) *Pronósticos, Series de Tiempo y Regresión. Un Enfoque Aplicado. 4 ed*. CENGAGE Learning

[2] Chatfield, C. (2019) *The Analysis of Time Series. An Introduction with R, Seventh edition*. CRC Press-USA.

[3] Diebold, F. (2001) *Elementos de Pronósticos*. International Thomson Editores, México.

[4] Cryer, J. D. and Chan, K-S. (2008) *Time Series Analysis With Applications in R*. Springer.

[5] González, N. G. (2013) *Notas de Clase Estadística III 3009137*. Escuela de Estadística, Universidad Nacional de Colombia Sede Medellín.

[6] Shumway, R. H. and Stoffer, D. S. (2017) *Time Series Analysis and Its Applications. With R Examples*. Fourth ed. Springer

APÉNDICE A. Uso de métodos de Identificación automática de modelos ARMA en los errores estructurales de modelos de regresión en R

Considere la serie mensual de muertes en accidente de tránsito en Ontario, enero 1960 a diciembre 1974 (Fuente: Ledolter, 1983) que se ilustra a continuación junto con su logaritmo natural. Para el logaritmo natural de la serie, $\log(Y_t)$, se ajustó un modelo de regresión lineal (MODELO 1) de tendencia cuadrática y estacionalidad con funciones trigonométricas en las frecuencias F_j=j/12, j=1,2,…,6, usando los primeros n=168 datos (de enero 1960 a diciembre 1973).

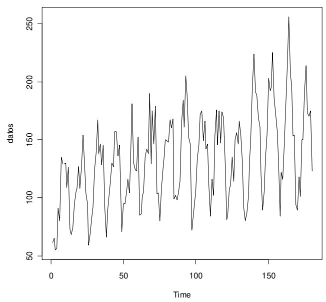


Fig. A1. Serie en escala original

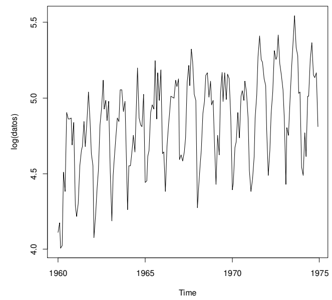


Fig. A2. Serie en escala log. natural

Modelo 1:

$\log(Y_t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \sum_{j=1}^{5}\left[\alpha_j \sin(\pi j t/6) + \gamma_j \cos(\pi j t/6)\right] + \gamma_6 \cos(\pi t) + E_t, \quad \{E_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0,\sigma^2)$

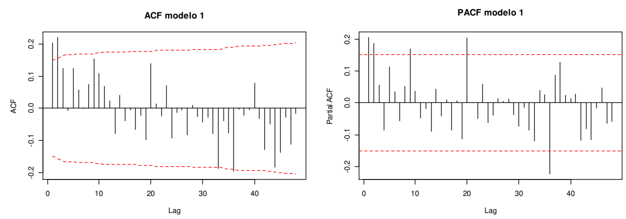


Fig. A3. ACF, PACF construidos con residuos del modelo 1

```
library(forecast);library(TSA);
datos=scan(file.choose()) #leer abraham5.dat
datos=ts(datos,freq=12,start=c(1960,1))
n=length(datos)-12; t=1:n; t2=t^2
sen1=sin(pi*t/6);cos1=cos(pi*t/6);sen2=sin(pi*t/3)
cos2=cos(pi*t/3);sen3=sin(pi*t/2);cos3=cos(pi*t/2)
sen4=sin(2*pi*t/3);cos4=cos(2*pi*t/3);sen5=sin(5*pi*t/6)
cos5=cos(5*pi*t/6);cos6=cos(pi*t)
X=data.frame(t,t2,sen1,cos1,sen2,cos2,sen3,cos3,sen4,cos4,sen5,cos5,cos6)
yt=ts(datos[t],freq=12,start=c(1960,1))

mod1=lm(log(yt)~.,data=X)
summary(mod1)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.540e+00  3.011e-02 150.772  < 2e-16 ***
t            4.789e-03  8.222e-04   5.825 3.22e-08 ***
t2          -1.200e-05  4.712e-06  -2.548 0.011826 *
sen1        -3.160e-01  1.404e-02 -22.512  < 2e-16 ***
cos1        -7.059e-02  1.402e-02  -5.036 1.32e-06 ***
sen2        -5.587e-02  1.402e-02  -3.985 0.000104 ***
cos2         5.810e-02  1.402e-02   4.145 5.60e-05 ***
sen3        -2.431e-02  1.402e-02  -1.734 0.084887 .
cos3         4.632e-02  1.402e-02   3.305 0.001183 **
sen4        -1.081e-02  1.402e-02  -0.772 0.441535
cos4         2.229e-02  1.402e-02   1.590 0.113914
sen5         2.766e-02  1.402e-02   1.974 0.050222 .
cos5         3.783e-02  1.402e-02   2.699 0.007740 **
cos6         1.599e-02  9.911e-03   1.613 0.108822
```

- Print-Screen de la identificación con auto.arima sobre vector de residuales sin fechas, usando criterios AIC y BIC

```
> auto.arima(residuals(mod1),ic="aic")
Series: residuals(mod1)
ARIMA(4,0,1) with zero mean

Coefficients:
          ar1     ar2     ar3      ar4     ma1
      -0.2463  0.2611  0.1421  -0.0809  0.4128
s.e.   0.3998  0.0978  0.1026   0.0888  0.3958

sigma^2 estimated as 0.01375:  log likelihood=121.6
AIC=-231.19   AICc=-230.67   BIC=-212.45
> auto.arima(residuals(mod1),ic="bic")
Series: residuals(mod1)
ARIMA(2,0,0) with zero mean

Coefficients:
         ar1     ar2
      0.1667  0.1876
s.e.  0.0757  0.0758

sigma^2 estimated as 0.01397:  log likelihood=120.31
AIC=-234.62   AICc=-234.48   BIC=-225.25
>
```

Se identifican dos modelos:
Con AIC: E_t un ARIMA(4,0,1)=ARMA(4,1) estacionario y de media cero, entonces

$$E_t = \sum_{j=1}^{4} \phi_j E_{t-j} + a_t + \theta_1 a_{t-1}, \text{con } \{a_t\}_{t\in\mathbb{Z}^+} \text{ un } RB \sim N(0,\sigma_a^2)$$

Con BIC: E_t un ARIMA(2,0,0)=AR(2) estacionario de media cero, entonces

$$E_t = \phi_1 E_{t-1} + \phi_2 E_{t-2} + a_t, \text{con } \{a_t\}_{t\in\mathbb{Z}^+} \text{ un } RB \sim N(0,\sigma_a^2)$$

Los coeficientes en la salida R denominados ar1, ar2, ar3 y ar4 hacen referencia a ϕ_1, ϕ_2, ϕ_3 y ϕ_4 respectivamente, mientras que el coeficiente ma1 hace referencia a θ_1. La función da además los valores estimados de los parámetros y sus errores estándar, pero estos no son los que tomaremos en la estimación del modelo global con errores ARMA.

- Print-Screen de la identificación con auto.arima sobre vector de residuales con fechas, usando criterios AIC y BIC

```
> SerieEt=ts(residuals(mod1),freq=12,start=c(1960,1))
>
> auto.arima(SerieEt,ic="aic")
Series: SerieEt
ARIMA(2,0,0)(2,0,0)[12] with zero mean

Coefficients:
         ar1     ar2     sar1     sar2
      0.1776  0.1857  -0.0044  -0.1326
s.e.  0.0761  0.0769   0.0831   0.0838

sigma^2 estimated as 0.01373:  log likelihood=121.55
AIC=-233.1   AICc=-232.73   BIC=-217.48
> auto.arima(SerieEt,ic="bic")
Series: SerieEt
ARIMA(2,0,0)(2,0,0)[12] with zero mean

Coefficients:
         ar1     ar2     sar1     sar2
      0.1776  0.1857  -0.0044  -0.1326
s.e.  0.0761  0.0769   0.0831   0.0838

sigma^2 estimated as 0.01373:  log likelihood=121.55
AIC=-233.1   AICc=-232.73   BIC=-217.48
>
```

Aquí se identifican modelos ARMA estacionales estacionarios, así
Por AIC: E_t es un ARIMA(2,0,0)(2,0,0)[12]=AR(2)×AR(2)[12] de media cero y estacionario
Por BIC: Identifica el mismo modelo dado con AIC.

NOTA: E_t es un ARMA(p,q)(P,Q)[s] corresponde a un proceso ARMA donde hay la interacción de dos estructuras de dependencia:

a) Una estructura ARMA(p,q) que explica la dependencia entre los valores sucesivos de E_t, es decir, entre $\ldots, E_{t-3}, E_{t-2}, E_{t-1}, E_t, E_{t+1}, E_{t+2}, E_{t+3}, \ldots$. Esta estructura tiene un polinomio AR(p) dado por $\phi_p(B) = 1 - \sum_{j=1}^{p} \phi_j B^j$ y un polinomio MA(q) dado por $\theta_q(B) = 1 + \sum_{i=1}^{q} \theta_i B^i$

b) Una estructura ARMA(P,Q)[s] que explica la dependencia entre valores de E_t separados s períodos en el tiempo, es decir, entre $\ldots, E_{t-3s}, E_{t-2s}, E_{t-s}, E_t, E_{t+s}, E_{t+2s}, E_{t+3s}, \ldots$. Esta estructura tiene un polinomio AR(P) de período s dado por $\Phi_P(B^s) = 1 - \sum_{k=1}^{P} \Phi_k B^{ks}$ y un polinomio MA(Q) de período *s* dado por $\Theta_Q(B^s) = 1 + \sum_{l=1}^{Q} \Theta_l B^{ls}$

c) El modelo general tiene la siguiente ecuación $\phi_p(B)\Phi_P(B^s)E_t = \theta_q(B)\Theta_Q(B^s)a_t$, con $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0, \sigma_a^2)$

Luego, para el modelo identificado se tiene, $\phi_2(B) = 1 - \phi_1 B - \phi_2 B^2$, $\Phi_2(B^{12}) = 1 - \Phi_1 B^{12} - \Phi_2 B^{24}$; no hay polinomios MA y por tanto, $\phi_2(B)\Phi_2(B^{12})E_t = a_t$ de donde

$$(1 - \phi_1 B - \phi_2 B^2)(1 - \Phi_1 B^{12} - \Phi_2 B^{24})E_t = a_t$$
$$\Rightarrow (1 - \phi_1 B - \phi_2 B^2 - \Phi_1 B^{12} + \phi_1\Phi_1 B^{13} + \phi_2\Phi_1 B^{14} - \Phi_2 B^{24} + \phi_1\Phi_2 B^{25} + \phi_2\Phi_2 B^{26})E_t = a_t$$

Despejando E_t, obtenemos,

$$E_t = \phi_1 E_{t-1} + \phi_2 E_{t-2} + \Phi_1 E_{t-12} - \phi_1\Phi_1 E_{t-13} - \phi_2\Phi_1 E_{t-14} + \Phi_2 E_{t-24} - \phi_1\Phi_2 E_{t-25} - \phi_2\Phi_2 E_{t-26} + a_t,$$
$$\text{con } \{a_t\}_{t\in\mathbb{Z}^+} \text{ un } RB \sim N(0, \sigma_a^2)$$

Los coeficientes en la salida R denominados ar1, ar2, hacen referencia a ϕ_1 y ϕ_2 de la estructura ARMA(2,0) mientras que los coeficientes denominados sar1 y sar2 corresponden a Φ_1 y Φ_2 de la estructura ARMA(2,0)[12]. Si hubiera parámetros θ_i aparecerían con nombre mai mientras que parámetros Θ_k aparecerían con el nombre smak, siendo las letras i y k los respectivos subíndices.

- Print-Screen de la identificación de modelos AR(p) con SelectModel sobre vector de residuales, usando criterios AIC y BIC y orden p máximo de 36. Primero cargue la función desde github así,

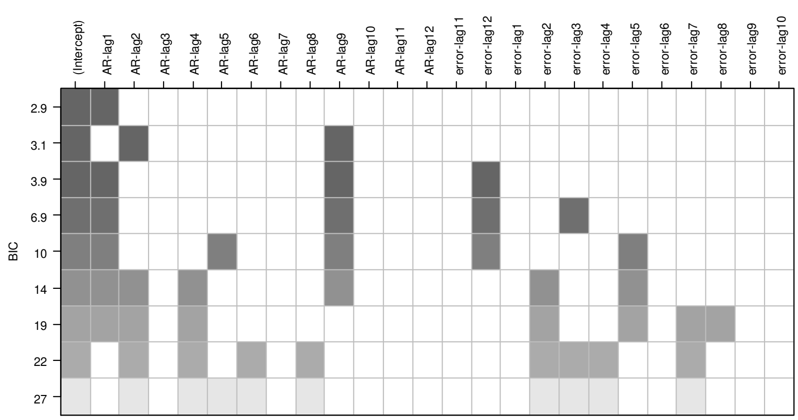
```
source("https://raw.githubusercontent.com/NelfiGonzalez/Funciones-de-Usuario-Estadistica-III/main/Funcion-SelectModel.R")
```

```
> SelectModel(residuals(mod1), lag.max = 36, Criterion="AIC",ARModel = "AR")
  p AIC-Exact AIC-Approx
1 2 -711.3872  -5.882468
2 3 -709.9035  -5.115701
3 4 -709.1081  -5.333515
> SelectModel(residuals(mod1), lag.max = 36, Criterion="BIC",ARModel = "AR")
  p BIC-Exact BIC-Approx
1 2 -702.0153   6.613388
2 1 -701.1310   2.014936
3 0 -699.0217   2.971439
>
```

Se identifican los siguientes modelos para E_t:
Por AIC: el mejor modelo es un AR(2): $E_t = \phi_1 E_{t-1} + \phi_2 E_{t-2} + a_t$, el segundo mejor un AR(3): $E_t = \phi_1 E_{t-1} + \phi_2 E_{t-2} + \phi_3 E_{t-3} + a_t$ y el tercer mejor es un AR(4): $E_t = \phi_1 E_{t-1} + \phi_2 E_{t-2} + \phi_3 E_{t-3} + \phi_4 E_{t-4} + a_t$, y en todos los casos bajo el supuesto de $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0, \sigma_a^2)$.
Por BIC: el mejor modelo es un AR(2): $E_t = \phi_1 E_{t-1} + \phi_2 E_{t-2} + a_t$, el segundo mejor es un AR(1): $E_t = \phi_1 E_{t-1} + a_t$ y en ambos casos bajo el supuesto de $\{a_t\}_{t\in\mathbb{Z}^+}$ *un* $RB \sim N(0, \sigma_a^2)$. El tercero dice que p=0 pero indicaría que no hay estructura AR(p) (no necesariamente que haya un ruido blanco!!)

- Identificación con armasubsets, fijando un valor máximo para p y q en 12

```
win.graph(width=10,height=5)
plot(armasubsets(residuals(mod1),nar=12,nma=12,y.name='AR',ar.method='ml'))
```



En cada renglón se identifica un posible ARMA, de modo que en el renglón superior está el modelo ARMA de menor BIC y hacia abajo va aumentando el BIC. Esta función internamente ajusta de modo aproximado modelos arma para la serie sobre la que se aplica, a través de una regresión con intercepto, donde las variables explicatorias son rezagos de esa serie con coeficientes de regresión iguales a los ϕ_j y rezagos de un proceso del cual se aproximan los valores de los a_{t-i}, cuyos coeficientes de regresión son los θ_i; por tanto, en los modelos que se muestran en esta gráfica aparece siempre un intercepto, pero sólo interesan los coeficientes denominados AR-lagj, que representan a los ϕ_j y los coeficientes denominados error-lagi que representan a los θ_i, respectivamente. Para cada renglón, las casillas sombreadas indican cuáles coeficientes deben ir en el modelo mientras que las casillas en blanco indican cuáles de los coeficientes no van en el modelo, es decir, valen cero. Entonces para los tres primeros modelos (renglón 1, 2 y 3 de arriba hacia abajo), esta función nos dice que

Renglón 1: $E_t = \phi_1 E_{t-1} + a_t$, con $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0, \sigma_a^2)$, es decir, E_t es un AR(1)

Renglón 2: $E_t = \phi_2 E_{t-2} + \phi_9 E_{t-9} + a_t$, con $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0, \sigma_a^2)$, es decir, E_t es un AR(9) pero sólo ϕ_2 y ϕ_9 van en el modelo.

Renglón 3: $E_t = \phi_1 E_{t-1} + \phi_9 E_{t-9} + a_t + \theta_{12} a_{t-12}$, con $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0, \sigma_a^2)$, es decir, E_t es un ARMA(9,12) pero sólo ϕ_1, ϕ_9 y θ_{12} van en el modelo.

- Print-screen de la identificación con EACF fijando un valor máximo de 36 para los órdenes p y q

```
> eacf(residuals(mod1),ar.max=36,ma.max=36)
AR/MA
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
0  x x o o o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  x  o  o  x  o
1  x o o o o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  x  o  o  x  o
2  x x o o o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  x  o  o  x  o
3  x o x o o o o o o o o  o  o  o  o  o  o  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  x  o
4  x o x o o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
5  x x x o o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
6  x o x o o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
7  x o x o o x o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
8  x x x o o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
9  x x o x o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
10 x x o x o o o x x o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
11 x x o x x o o o x o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
12 x x x x o o o o o x o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
13 x x o x o o o o o x o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
14 x x o x o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
15 x x o x o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
16 o x x o o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
17 o x x o o o o o o o x  x  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
18 o x x o o o o o o x x  x  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
19 x x x x x x x o o o x  x  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
20 o o x x o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
21 o o o x o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
22 x o o x o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
23 x x x x x o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
24 x x o o o o o o o o o  o  o  o  o  o  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
25 x o o o o o o o o o o  o  o  o  o  o  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
26 o o o o o o o o o o o  o  o  o  o  o  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
27 x o o o o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
28 o o o o o o o o o o o  o  o  o  o  o  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
29 x o o o o o o o o o o  o  o  o  o  o  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
30 x x x x x o o o o o o  o  o  o  o  o  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
31 x x x x o o o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
32 x o x o o x o o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
33 x x x o o o o o o o o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
34 x o x o o o o o o o o  x  o  o  o  o  o  o  o  o  o  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
35 o x x o o o o o o o x  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
36 x x x x x o o o o o x  x  x  x  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
```

¿Qué modelo se identifica? *Para hacer uso de esta gráfica debe estudiar con la lectura programada: Función de autocorrelación extendida o EACF*

APÉNDICE B. Sobre ajustes, validación supuestos y pronósticos en R de modelos de regresión con errores ARMA

Los ejemplos en las Secciones I, II y III de este Apéndice, consideran una estructura de regresión lineal en los parámetros de tendencia y estacionalidad, es decir, modelos aditivos polinomiales estacionales y modelos log polinomiales estacionales, con errores estructurales $\mathrm{E_t}$ correlacionados según algún modelo estocástico estacionario de media cero. ***Para los casos de modelos de regresión no lineal en los parámetros de tendencia y estacionalidad (modelos exponenciales), vaya a la Sección IV página 20.***

Pasos iniciales: Cargar librerías y funciones de usuario necesarias:

```
library(forecast);library(lmtest);library(TSA)

source("https://raw.githubusercontent.com/NelfiGonzalez/Funciones-de-Usuario-Estadistica-III/main/Funciones-Criterios.Informacion-Calidad.Intervalos.R")
source("https://raw.githubusercontent.com/NelfiGonzalez/Funciones-de-Usuario-Estadistica-III/main/Funciones-BP.LB.test-pruebaDW1.R")
```

Nota: En los siguientes ejemplos, el paso 1) solo debe realizarlo una vez, pues en todos los modelos que ud. ajustará en este segundo trabajo, se mantiene igual la estructura de tendencia y estacionalidad. En los pasos 2) a 7), tenga en cuenta que, por cada modelo distinto a considerar en los errores, debe crear con diferentes nombres los objetos que guarden valores ajustados, residuos, pronósticos, etc.

I. Modelos lineales en los parámetros de tendencia y estacionalidad, y errores ARMA(p,q) con todos los parámetros. Por ejemplo, considere los siguientes dos modelos de regresión lineal con errores estructurales ARMA:

a. $Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \sum_{i=1}^{11} \delta_i I_{i,t} + E_t$, donde $E_t = \sum_{j=1}^{p} \phi_j E_{t-j} + a_t + \sum_{i=1}^{q} a_{t-i}$, con $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0,\sigma_a^2)$,

b. $\log(Y_t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \sum_{j=1}^{5}\{\alpha_j \sin(\pi j t/6) + \gamma_j \cos(\pi j t/6)\} + \gamma_6 \cos(\pi t) + E_t$, donde $E_t = \sum_{j=1}^{p} \phi_j E_{t-j} + a_t + \sum_{i=1}^{q} \theta_i a_{t-i}$, con $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0,\sigma_a^2)$.

Veamos cada caso:

a. Para modelo $Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \sum_{i=1}^{11} \delta_i I_{i,t} + E_t$, donde $E_t = \sum_{j=1}^{p} \phi_j E_{t-j} + a_t + \sum_{i=1}^{q} \theta_i a_{t-i}$, con $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0,\sigma_a^2)$, se procede de la siguiente manera:

1) Lea datos y defina variables necesarias para el ajuste y el pronóstico (en este ejemplo se asumen 12 pronósticos ex – post):

```
datos=read.table(file.choose(),....) #leer conforme a la base de datos asignada
datos=ts(datos,freq=12,start=c(1960,1)) #convertir en serie según frecuencia y fecha inicial
n=length(datos)-12 #Defina longitud de los datos para ajuste con validación cruzada. Aquí se recorta en 12 datos
#Defina el índice de tiempo y sus potencias según polinomio planteado para Tt
t=1:n; t2=t^2
yt=ts(datos[t],freq=12,start=c(1960,1)) #serie recortada
#Defina la matriz de variables indicadoras
Mes=seasonaldummy(yt)

#Defina la matriz de variables predictoras de tendencia y estacionalidad:
X=data.frame(t,t2,Mes)

#Definiendo variables para pronósticos
tnuevo=(n+1):length(datos)
t2nuevo=tnuevo^2
Mesnuevo=seasonaldummy(yt,h=12) #Horizonte de pronóstico h igual a longitud de recorte de la serie
Xnuevo=data.frame(t=tnuevo,t2=t2nuevo,Mesnuevo) #matriz de predictores en el pronóstico
ytf=ts(datos[tnuevo],freq=12,start=c(1974,1)) #los datos no usados en ajuste. Fechas y frecuencias según su caso
```

2) Ajuste el modelo con función Arima de librería forecast

```
#indique el valor correspondiente a p y q en la función Arima, así como la matriz de predictores X en xreg=as.matrix(X)
modelo=Arima(yt,order=c(p,0,q),xreg=as.matrix(X),method="ML")
```

3) Construya tabla completa de parámetros estimados, valores P corresponden a $P(|t_{n-k}| > |T_0|)$, con k la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA.

```
#Calcule grados de libertad del MSE del modelo
k=length(coef(modelo)[coef(modelo)!=0]);k #número de parámetros del modelo
dfmodelo=n-k
#Construya tabla de parámetros estimados con estadísticos T0 y valores P para cada parámetro
coeftest(modelo,df=dfmodelo)
```

4) Obtenga gráfica del ajuste de la serie y de los residuales de ajuste $\hat{a}_t$

```
#Gráfico de la serie y su ajuste
ythat=modelo$fitted #Este objeto ya tiene fechas
plot(datos)
lines(ythat,col=2)
legend("topleft",legend=c("datos","ajuste modelo"),lty=1,col=1:2)
#Gráficos de residuales
plot(residuals(modelo));abline(h=0)
abline(h=c(-2*sqrt(modelo$sigma2),2*sqrt(modelo$sigma2)),lty=2)
plot(as.numeric(modelo$fitted),residuals(modelo));abline(h=0)
abline(h=c(-2*sqrt(modelo$sigma2),2*sqrt(modelo$sigma2)),lty=2)
```

5) Cálculo de AIC y BIC, versión $\exp(C_n^*(p))$, con el número de parámetros k siendo la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA, previamente calculado en 3).

```
Criteriosmodelo=exp.crit.inf.resid(residuales=residuals(modelo),n.par=k); Criteriosmodelo
```

6) Valide supuestos sobre el error de ajuste a_t

```
#ACF sobre residuales de ajuste en el modelo. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
acf(as.numeric(residuals(modelo)),ci.type="ma",lag.max=m,main="ACF modelo",ci.col=2)

#PACF sobre residuales de ajuste en el modelo. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
pacf(as.numeric(residuals(modelo)),lag.max=m,main="PACF modelo",ci.col=2)

BP.LB.test(residuals(modelo),maxlag=m,type="Ljung") #test Ljung-Box use máximo m igual al de ACF y PACF

#Normalidad sobre residuales de ajuste en el modelo. Sólo si no se rechaza supuesto de ruido blanco
shapiro.test(residuals(modelo))

win.graph()
qqnorm(residuals(modelo),main="Gráfico de normalidad residuos de ajuste modelo")
qqline(residuals(modelo),col=2)
```

7) Pronósticos para la validación cruzada (en este ejemplo se asume 12 pronósticos ex – post)

```
predmodelo=ts(as.data.frame(forecast(modelo,xreg=as.matrix(Xnuevo),level=95)),freq=12, start=c(1974,1))
predmodelo
ytpronmodelo=predmodelo[,1] #Tomando el pronóstico puntual. Este objeto tiene fechas

#Medidas precisión pronósticos
accuracy(ytpronmodelo,ytf)
amplitud.cobertura(real=ytf,LIP=predmodelo[,2],LSP=predmodelo[,3])
```

b. Modelo $\log(Y_t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \sum_{j=1}^{5}\{\alpha_j \sin(\pi j t/6) + \gamma_j \cos(\pi j t/6)\} + \gamma_6 \cos(\pi t) + E_t$, donde $E_t = \sum_{j=1}^{p} \phi_j E_{t-j} + a_t + \sum_{i=1}^{q} \theta_i a_{t-i}$, con $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0,\sigma_a^2)$

1) Lea datos y defina variables necesarias para el ajuste y el pronóstico (en este ejemplo se asume 12 pronósticos ex – post)

```
datos=read.table(file.choose(),....) #leer conforme a la base de datos asignada
datos=ts(datos,freq=12,start=c(1960,1)) #convertir en serie según frecuencia y fecha inicial
n=length(datos)-12 #Defina longitud de los datos para ajuste con validación cruzada, según su caso
#Defina el índice de tiempo y sus potencias según polinomio
t=1:n; t2=t^2; t3=t^3
yt=ts(datos[t],freq=12,start=c(1960,1)) #serie recortada

#Defina las variables trigonométricas necesarias según modelo postulado
sen1=sin(pi*t/6)
cos1=cos(pi*t/6)
sen2=sin(pi*t/3)
cos2=cos(pi*t/3)
sen3=sin(pi*t/2)
cos3=cos(pi*t/2)
sen4=sin(2*pi*t/3)
cos4=cos(2*pi*t/3)
sen5=sin(5*pi*t/6)
cos5=cos(5*pi*t/6)
cos6=cos(pi*t)
#Defina la matriz de variables predictoras de tendencia y estacionalidad:
X=data.frame(t,t2,t3,sen1,cos1,sen2,cos2,sen3,cos3,sen4,cos4,sen5,cos5,cos6)

#Definiendo variables para pronósticos
tnuevo=(n+1):length(datos)
t2nuevo=tnuevo^2
t3nuevo=tnuevo^3
sen1n=sin(pi*tnuevo/6)
cos1n=cos(pi*tnuevo/6)
sen2n=sin(pi*tnuevo/3)
cos2n=cos(pi*tnuevo/3)
sen3n=sin(pi*tnuevo/2)
cos3n=cos(pi*tnuevo/2)
sen4n=sin(2*pi*tnuevo/3)
cos4n=cos(2*pi*tnuevo/3)
sen5n=sin(5*pi*tnuevo/6)
cos5n=cos(5*pi*tnuevo/6)
cos6n=cos(pi*tnuevo)

#matriz de predictores en el pronóstico
Xnuevo=data.frame(t=tnuevo,t2=t2nuevo,t3=t3nuevo,sen1=sen1n,cos1=cos1n,sen2=sen2n,cos2=cos2n,sen3=sen3n,cos3=cos3n,
                  sen4=sen4n,cos4=cos4n,sen5=sen5n,cos5=cos5n,cos6=cos6n)
ytf=ts(datos[tnuevo],freq=12,start=c(1974,1)) #los datos no usados en ajuste. Fechas y frecuencias según datos
```

2) Ajuste el modelo con función Arima de librería forecast

```
#indique el valor correspondiente a p y q en la función Arima, así como la matriz de predictores X en xreg=as.matrix(X)
modelo=Arima(log(yt),order=c(p,0,q),xreg=as.matrix(X),method="ML")
```

3) Construya tabla completa de parámetros estimados, valores P corresponden a $P(|t_{n-k}| > |T_0|)$, con k la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA.

```
#Calcule grados de libertad del MSE del modelo
k=length(coef(modelo)[coef(modelo)!=0]);k #número de parámetros del modelo
dfmodelo=n-k

#Construya tabla de parámetros estimados con estadísticos T0 y valores P para cada parámetro
coeftest(modelo,df=dfmodelo)
```

4) Obtenga gráfica del ajuste de la serie en escala original y de los residuales de ajuste $\hat{a}_t$. Observe que en el factor de corrección por transformación lognormal, se invoca a modelo$sigma2 el cual es igual a $\hat{\sigma}_a^2$

```
ythat=exp(modelo$fitted)*exp(modelo$sigma2/2) #este objeto ya queda con las fechas de la serie
#Gráfico de la serie y su ajuste
plot(datos)
lines(ythat,col=2)
legend("topleft",legend=c("datos","ajuste modelo"),lty=1,col=1:2)

#Gráficos de residuales
plot(residuals(modelo));abline(h=0)
abline(h=c(-2*sqrt(modelo$sigma2),2*sqrt(modelo$sigma2)),lty=2)
plot(as.numeric(modelo$fitted),residuals(modelo));abline(h=0)
abline(h=c(-2*sqrt(modelo$sigma2),2*sqrt(modelo$sigma2)),lty=2)
```

5) Cálculo de AIC y BIC, versión $\exp(C_n^*(p))$, con el número de parámetros k siendo la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA, previamente calculado en 3).

```
Res.orig=yt-ythat #Cálculo de pseudo residuales
Criteriosmodelo=exp.crit.inf.resid(residuales= Res.orig,n.par=k); Criteriosmodelo
```

6) Valide supuestos sobre el error de ajuste a_t

```
#ACF sobre residuales de ajuste en el modelo. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
acf(as.numeric(residuals(modelo)),ci.type="ma",lag.max=m,main="ACF modelo",ci.col=2)

#PACF sobre residuales de ajuste en el modelo. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
pacf(as.numeric(residuals(modelo)),lag.max=m,main="PACF modelo",ci.col=2)
BP.LB.test(residuals(modelo),maxlag=m,type="Ljung") #test Ljung-Box use máximo m igual al de ACF y PACF
#Normalidad sobre residuales de ajuste en el modelo. Sólo si no se rechaza supuesto de ruido blanco
shapiro.test(residuals(modelo))

win.graph()
qqnorm(residuals(modelo),main="Gráfico de normalidad residuos de ajuste modelo")
qqline(residuals(modelo),col=2)
```

7) Pronósticos para la validación cruzada. Observe de nuevo que en el factor de corrección por transformación lognormal, se invoca a modelo$sigma2 el cual es igual a $\hat{\sigma}_a^2$ (en este ejemplo se asume 12 pronósticos ex – post)

```
#Cálculo del pronóstico con I.P del 95%, en escala original
predmodelo=exp(as.data.frame(forecast(modelo,xreg=as.matrix(Xnuevo),level=95)))*exp(modelo$sigma2/2)
predmodelo=ts(predmodelo,freq=12,start=c(1974,1)) #Fechas y frecuencia según datos en pronósticos
predmodelo
ytpronmodelo=predmodelo[,1] #Tomando el pronóstico puntual

#Medidas precisión pronósticos
accuracy(ytpronmodelo,ytf)
amplitud.cobertura(real=ytf,LIP=predmodelo[,2],LSP=predmodelo[,3])
```

II. Modelos lineales en los parámetros de tendencia y estacionalidad, y errores ARMA(p,q) con sólo alguno de los parámetros de esta estructura ARMA. Por ejemplo,
$\log(Y_t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \sum_{j=1}^{5}\{\alpha_j \sin(\pi j t/6) + \gamma_j \cos(\pi j t/6)\} + \gamma_6 \cos(\pi t) + E_t$, donde $E_t = \phi_1 E_{t-1} + \phi_9 E_{t-9} + a_t + \theta_{12} a_{t-12}$, con $\{a_t\}_{t \in \mathbb{Z}^+}$ un $RB \sim N(0, \sigma_a^2)$. Se procede de la siguiente manera:

1) Lea datos y defina variables necesarias para el ajuste y el pronóstico (en este ejemplo se asume 12 pronósticos ex – post)

```
datos=read.table(file.choose(),....) #leer conforme a la base de datos asignada
datos=ts(datos,freq=12,start=c(1960,1)) #convertir en serie según frecuencia y fecha inicial
n=length(datos)-12 #Defina longitud de los datos para ajuste con validación cruzada, según su caso
#Defina el índice de tiempo y sus potencias según polinomio
t=1:n; t2=t^2; t3=t^3
yt=ts(datos[t],freq=12,start=c(1960,1)) #serie recortada

#Defina las variables trigonométricas necesarias según modelo postulado
sen1=sin(pi*t/6)
cos1=cos(pi*t/6)
sen2=sin(pi*t/3)
cos2=cos(pi*t/3)
sen3=sin(pi*t/2)
cos3=cos(pi*t/2)
```

```
sen4=sin(2*pi*t/3)
cos4=cos(2*pi*t/3)
sen5=sin(5*pi*t/6)
cos5=cos(5*pi*t/6)
cos6=cos(pi*t)

#Defina la matriz de variables predictoras de tendencia y estacionalidad:
X=data.frame(t,t2,t3,sen1,cos1,sen2,cos2,sen3,cos3,sen4,cos4,sen5,cos5,cos6)

#Definiendo variables para pronósticos
tnuevo=(n+1):length(datos)
t2nuevo=tnuevo^2
t3nuevo=tnuevo^3
sen1n=sin(pi*tnuevo/6)
cos1n=cos(pi*tnuevo/6)
sen2n=sin(pi*tnuevo/3)
cos2n=cos(pi*tnuevo/3)
sen3n=sin(pi*tnuevo/2)
cos3n=cos(pi*tnuevo/2)
sen4n=sin(2*pi*tnuevo/3)
cos4n=cos(2*pi*tnuevo/3)
sen5n=sin(5*pi*tnuevo/6)
cos5n=cos(5*pi*tnuevo/6)
cos6n=cos(pi*tnuevo)
#matriz de predictores en el pronóstico
Xnuevo=data.frame(t=tnuevo,t2=t2nuevo,t3=t3nuevo,sen1=sen1n,cos1=cos1n,sen2=sen2n,cos2=cos2n,sen3=sen3n,cos3=cos3n,
                  sen4=sen4n,cos4=cos4n,sen5=sen5n,cos5=cos5n,cos6=cos6n)
ytf=ts(datos[tnuevo],freq=12,start=c(1974,1)) #los datos no usados en ajuste. Fechas y frecuencias según su caso
```

2) Ajuste el modelo con función Arima de librería forecast. Dado que no se usan todos los parámetros del modelo ARMA(p,q) en los errores estructurales E_t, en la función Arima debe especificarse cuáles parámetros van en el modelo y cuáles se fijan en cero. Para ello, se usa el argumento fixed igual a un vector de longitud (p+q)+(No. Parámetros tendencia)+(No. Parámetros estacionalidad). Las entradas o valores de este vector se ordenan de la siguiente manera: Los primeros p valores son relativos a los coeficientes ϕ_j, los siguientes q valores hacen referencia a los coeficientes θ_i, los siguientes r valores (r= No. Parámetros tendencia+ No. Parámetros estacionalidad) hacen referencia a los parámetros de la tendencia y la componente estacional de acuerdo a la estructura de regresión postulada. En cada entrada del vector se coloca un 0 si el parámetro no va en el modelo y un NA, si el respectivo parámetro debe ir en el modelo. Por tanto, para los últimos r valores del vector se debe asignar un NA, mientras que para los primeros (p+q) valores correspondientes a los coeficientes del ARMA, se coloca 0 ó NA, según la ecuación específica del ARMA. Para el ejemplo, como el error estructural es un ARMA(9,12) pero sólo los coeficientes ϕ_1, ϕ_9 y θ_{12} van en la estructura ARMA, mientras que en la parte de tendencia y estacionalidad se tienen 15 parámetros, el argumento fixed en la función Arima debe ser especificado así (OJO: rep(a,b) repite b veces el argumento a),

fixed=c(NA,rep(0,7),NA,rep(0,11),NA,rep(NA,15))

que es equivalente a escribir de forma extendida lo siguiente,

fixed=c(NA,0,0,0,0,0,0,0,NA,0,0,0,0,0,0,0,0,0,0,0,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA)

donde el color del carácter indica

- Las entradas para los parámetros parte AR(9)
- Las entradas para los parámetros parte MA(12)
- Las entradas para los parámetros de tendencia y estacionalidad

```
#indique el valor correspondiente a p y q en la función Arima, así como la matriz de predictores X en xreg=as.matrix(X) y
#el vector para argumento fixed
modelo=Arima(log(yt),order=c(9,0,12),xreg=as.matrix(X),fixed= c(NA,rep(0,7),NA,rep(0,11),NA,rep(NA,15)),method="ML")
```

3) Construya tabla completa de parámetros estimados, valores P corresponden a $P(|t_{n-k}| > |T_0|)$, con k la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA.

```
#Calcule grados de libertad del MSE del modelo
k=length(coef(modelo)[coef(modelo)!=0]);k #número de parámetros del modelo
dfmodelo=n-k
#Construya tabla de parámetros estimados con estadísticos T0 y valores P para cada parámetro
coeftest(modelo,df=dfmodelo)
```

4) Obtenga gráfica del ajuste de la serie en escala original y de los residuales de ajuste $\hat{a}_t$. Observe que en el factor de corrección por transformación lognormal, se invoca a modelo$sigma2 el cual es igual a $\hat{\sigma}_a^2$

```
ythat=exp(modelo$fitted)*exp(modelo$sigma2/2) #este objeto ya queda con las fechas de la serie
#Gráfico de la serie y su ajuste
plot(datos)
lines(ythat,col=2)
legend("topleft",legend=c("datos","ajuste modelo"),lty=1,col=1:2)

#Gráficos de residuales
plot(residuals(modelo));abline(h=0)
abline(h=c(-2*sqrt(modelo$sigma2),2*sqrt(modelo$sigma2)),lty=2)
plot(as.numeric(modelo$fitted),residuals(modelo));abline(h=0)
abline(h=c(-2*sqrt(modelo$sigma2),2*sqrt(modelo$sigma2)),lty=2)
```

5) Cálculo de AIC y BIC, versión $\exp(C_n^*(p))$, con el número de parámetros k siendo la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA, previamente calculado en 3).

```
Res.orig=yt-ythat #Cálculo de pseudo residuales
Criteriosmodelo=exp.crit.inf.resid(residuales=Res.orig,n.par=k); Criteriosmodelo
```

6) Valide supuestos sobre el error de ajuste a_t

```
#ACF sobre residuales de ajuste en el modelo. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
acf(as.numeric(residuals(modelo)),ci.type="ma",lag.max=m,main="ACF modelo",ci.col=2)

#PACF sobre residuales de ajuste en el modelo. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
pacf(as.numeric(residuals(modelo)),lag.max=m,main="PACF modelo",ci.col=2)

BP.LB.test(residuals(modelo),maxlag=m,type="Ljung") #test Ljung-Box use máximo m igual al de ACF y PACF
#Normalidad sobre residuales de ajuste en el modelo. Sólo si no se rechaza supuesto de ruido blanco
shapiro.test(residuals(modelo))
win.graph()
qqnorm(residuals(modelo),main="Gráfico de normalidad residuos de ajuste modelo")
qqline(residuals(modelo),col=2)
```

7) Pronósticos para la validación cruzada. Observe de nuevo que en el factor de corrección por transformación lognormal, se invoca a modelo\$sigma2 el cual es igual a $\hat{\sigma}_a^2$

```
#Cálculo del pronóstico con I.P del 95%, en escala original
predmodelo=exp(as.data.frame(forecast(modelo,xreg=as.matrix(Xnuevo),level=95)))*exp(modelo$sigma2/2)
predmodelo=ts(predmodelo,freq=12,start=c(1974,1))
predmodelo
ytpronmodelo=predmodelo[,1] #Tomando el pronóstico puntual

#Medidas precisión pronósticos
accuracy(ytpronmodelo,ytf)
amplitud.cobertura(real=ytf,LIP=predmodelo[,2],LSP=predmodelo[,3])
```

III. Modelos lineales en los parámetros de tendencia y estacionalidad, y errores ARMA(p,q)(P,Q)[s]

Por ejemplo,

$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \sum_{i=1}^{11} \delta_i I_{i,t} + E_t$, donde E_t es un ARMA(p,q)(P,Q)[12], es decir

$\phi_p(B)\Phi_P(B^{12})E_t = \theta_q(B)\Theta_Q(B^{12})a_t$, con $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0,\sigma_a^2)$. Entonces, el modelo queda como

$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \sum_{i=1}^{11} \delta_i I_{i,t} + E_t$, con

$$E_t = \sum_{j=1}^{p} \phi_j E_{t-j} + \sum_{k=1}^{P} \Phi_k\, E_{t-12*k} - \sum_{k=1}^{P} \Phi_k \sum_{j=1}^{p} \phi_j E_{t-j-12*k} + a_t + \sum_{i=1}^{q} \theta_i a_{t-i} + \sum_{l=1}^{Q} \Theta_l a_{t-12*l} + \sum_{l=1}^{Q} \Theta_l \sum_{i=1}^{q} \theta_i a_{t-i-12*l}$$

y $\{a_t\}_{t\in\mathbb{Z}^+}$ un $RB \sim N(0,\sigma_a^2)$

1) Lea datos y defina variables necesarias para el ajuste y el pronóstico (en este ejemplo se asume 12 pronósticos ex – post)

```
datos=read.table(file.choose(),....) #leer conforme a la base de datos asignada
datos=ts(datos,freq=12,start=c(1960,1)) #convertir en serie según frecuencia y fecha inicial
n=length(datos)-12 #Defina longitud de los datos para ajuste con validación cruzada, según su caso
#Defina el índice de tiempo y sus potencias según polinomio
t=1:n; t2=t^2; t3=t^3
yt=ts(datos[t],freq=12,start=c(1960,1)) #serie recortada

#Defina la matriz de variables indicadoras
Mes=seasonaldummy(yt)

#Defina la matriz de variables predictoras de tendencia y estacionalidad:
X=data.frame(t,t2,t3,Mes)

#Definiendo variables para pronósticos
tnuevo=(n+1):length(datos)
t2nuevo=tnuevo^2
t3nuevo=tnuevo^3
Mesnuevo=seasonaldummy(yt,h=12) #Horizonte de pronóstico h igual a longitud de recorte de la serie
Xnuevo=data.frame(t=tnuevo,t2=t2nuevo,t3=t3nuevo,Mesnuevo) #matriz predictores en el pronóstico
ytf=ts(datos[tnuevo],freq=12,start=c(1974,1)) #los datos no usados en ajuste. Fechas y frecuencias según su caso
```

2) Ajuste el modelo con función Arima de librería forecast. El orden p, q se especifica con order=c(p,0,q), mientras que el orden P, Q se especifica con seasonal=list(order=c(P,0,Q))

```
#indique valor correspondiente a p, q, P y Q en la función Arima, así como la matriz de predictores X en xreg=as.matrix(X)
modelo=Arima(yt,order=c(p,0,q),seasonal=list(order=c(P,0,Q)),xreg=as.matrix(X),method="ML")
```

3) Construya tabla completa de parámetros estimados, valores P corresponden a $P(|t_{n-k}| > |T_0|)$, con k la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA.

```
#Calcule grados de libertad del MSE del modelo,
k=length(coef(modelo)[coef(modelo)!=0]);k #número de parámetros del modelo
dfmodelo=n-k

#Construya tabla de parámetros estimados con estadísticos T0 y valores P para cada parámetro
coeftest(modelo,df=dfmodelo)
```

4) Obtenga gráfica del ajuste de la serie en escala original y de los residuales de ajuste $\hat{a}_t$.

```
ythat=modelo$fitted #este objeto ya queda con las fechas de la serie
#Gráfico de la serie y su ajuste
plot(datos)
lines(ythat,col=2)
legend("topleft",legend=c("datos","ajuste modelo"),lty=1,col=1:2)

#Gráficos de residuales
plot(residuals(modelo));abline(h=0)
abline(h=c(-2*sqrt(modelo$sigma2),2*sqrt(modelo$sigma2)),lty=2)
plot(as.numeric(modelo$fitted),residuals(modelo));abline(h=0)
abline(h=c(-2*sqrt(modelo$sigma2),2*sqrt(modelo$sigma2)),lty=2)
```

5) Cálculo de AIC y BIC, versión $\exp(C_n^*(p))$, con el número de parámetros k siendo la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA, previamente calculado en 3).

```
Criteriosmodelo=exp.crit.inf.resid(residuales=residuals(modelo),n.par=k); Criteriosmodelo
```

6) Valide supuestos sobre el error de ajuste a_t

```
#ACF sobre residuales de ajuste en el modelo. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
acf(as.numeric(residuals(modelo)),ci.type="ma",lag.max=m,main="ACF modelo",ci.col=2)

#PACF sobre residuales de ajuste en el modelo. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
pacf(as.numeric(residuals(modelo)),lag.max=m,main="PACF modelo",ci.col=2)

BP.LB.test(residuals(modelo),maxlag=m,type="Ljung") #test Ljung-Box use máximo m igual al de ACF y PACF
#Normalidad sobre residuales de ajuste en el modelo. Sólo si no se rechaza supuesto de ruido blanco
shapiro.test(residuals(modelo))

win.graph()
qqnorm(residuals(modelo),main="Gráfico de normalidad residuos de ajuste modelo")
qqline(residuals(modelo),col=2)
```

7) Pronósticos para la validación cruzada (en este ejemplo se supone que son 12 los pronósticos)

```
#Cálculo del pronóstico con I.P del 95%, en escala original
predmodelo=ts(as.data.frame(forecast(modelo,xreg=as.matrix(Xnuevo),level=95)),freq=12, start=c(1974,1))
predmodelo
ytpronmodelo=predmodelo[,1] #Tomando el pronóstico puntual. Este objeto tiene fechas

#Medidas precisión pronósticos
accuracy(ytpronmodelo,ytf)
amplitud.cobertura(real=ytf,LIP=predmodelo[,2],LSP=predmodelo[,3])
```

IV. Modelos exponenciales con errores ARMA.
Ningún modelo donde la estructura de regresión o ecuación estructural, es no lineal en los parámetros de tendencia y estacionalidad, podrá ajustarse con la función Arima(). Por otro lado, la función disponible en R para modelos de regresión no lineal con errores ARMA, gnls() de la librería nlme, no logra correr para muchos casos y es difícil ajustar sus argumentos de control para lograr que estime. La solución aproximada para este inconveniente es ajustar y pronosticar separadamente la estructura de regresión no lineal (el modelo exponencial), luego usar los residuales del modelo exponencial para ajustar y pronosticar la estructura ARMA y finalmente, juntar los valores ajustados y pronosticados de estas dos partes, así:

1. Ajustar sobre la serie de datos asignados (serie recortada para el ajuste con validación cruzada) y pronosticar la estructura exponencial.
2. Ajustar y pronosticar la estructura ARMA operando con los residuos estructurales que genera el ajuste exponencial.
3. Finalmente procedemos a construir una estimación y pronóstico total para la serie, sumando los respectivos ajustes y pronósticos de la regresión no lineal con los correspondientes ajustes y pronósticos del ARMA: es decir, $\hat{Y}_t \approx \widehat{\mathrm{EXP}}_t + \widehat{\hat{\mathrm{E}}}_t$, $\hat{Y}_n(L) \approx \widehat{\mathrm{EXP}}_n(L) + \hat{\mathrm{E}}_n(L)$, donde $\widehat{\mathrm{EXP}}_t$ y $\widehat{\mathrm{EXP}}_n(L)$ son respectivamente, el ajuste y el pronóstico de la estructura de regresión exponencial, y $\hat{\hat{E}}_t$ y $\hat{\mathrm{E}}_n(L)$ son respectivamente, el ajuste y el pronóstico de la estructura ARMA de media cero y estacionaria usando sólo los residuos del ajuste del modelo exponencial del trabajo 1.

Para implementar lo anterior se ha creado la función de usuario de nombre ***`regexpo.ErrorARMA()`*** que maneja los siguientes argumentos:

- **`respuesta`**: Un objeto ts con los datos a ser ajustados.
- **`names.param`**: Un vector de caracteres con los nombres para los parámetros en la función de regresión de las componentes estructurales.
- **`data`**, **`newdata`**: Objetos tipo data.frame, cuyas columnas corresponden a los valores de las variables predictoras del modelo en el ajuste y en el pronóstico, respectivamente.
- **`control`**: una lista opcional de configuraciones de control para la función nls(). Consulte nls.control() para conocer los nombres de los valores de control configurables y su efecto.
- **`order`**: Una especificación de la parte no estacional del modelo ARIMA, donde los tres componentes (p, d, q) son el orden AR, el grado de diferenciación y el orden MA, respectivamente.
- **`seasonal`**: Una especificación de la parte estacional del modelo ARIMA, más el período (que por defecto es frequency(respuesta)). Este argumente debe ser dado como una lista con el orden de los componentes y el período, pero una especificación de solo un vector numérico de longitud 3 se convertirá en una lista adecuada con las especificaciones del orden en la parte estacional.
- **`fixed`**: Vector numérico opcional de la misma longitud que la suma de los órdenes de la ecuación del ARMA estacionario de media cero a ajustar sobre residuos estructurales del ajuste exponencial. Ver ayuda de función **`arima()`** sobre detalles de este argumento.
- **`method`**: Método de ajuste: máxima verosimilitud ("**`ML`**"), Máxima verosilmilitud combinada con mínimos cuadrados condicionales ("**`CSS-ML`**") o mínimos de cuadrados condicionales ("**`CSS`**"). El valor predeterminado (a menos que existan valores faltantes) es "**`CSS-ML`**", en el cual se usan mínimos cuadrados condicionales para encontrar los valores iniciales y luego aplica máxima verosimilitud.
- **`optim.method`**: El valor pasado al argumento '**`method`**' para '**`optim`**' en la función **`Arima()`**. Por defecto es "**`BFGS`**".
- **`optim.control`**: Una lista de parámetros de control para '**`optim`**', usados en la función **`Arima()`**.

Resultados: La función produce una lista con las siguientes componentes

- **`coefficients`**: Matriz con la tabla de parámetros estimados, sus errores estándar, estadístico T0 y valor P asociado. *Tenga en cuenta que no hay una estimación conjunta de los parámetros de regresión de la función exponencial y de los parámetros del modelo ARMA del error estructural, de modo que los errores estándar provienen del ajuste separado de las dos estructuras, pero los valores p son calculados bajo una distribución t-student cuyos grados de libertad son df=n - total de parámetros*. El objeto tabla puede ser obtenido con la función **`coef()`** aplicada al objeto R donde se guarde el resultado de la función **`regexpo.ErrorARMA()`**.
- **`fitted`**: Objeto tipo ts con los valores ajustados de la serie. Estos valores pueden ser extraídos mediante la función fitted() sobre el objeto R donde se guarde el resultado de la función **`regexpo.ErrorARMA()`**.
- **`residuals`**: Objeto tipo ts con los residuos del ajuste total (los $\hat{a}_t$). Estos valores pueden ser extraídos mediante la función **`residuals()`** sobre el objeto R donde se guarde el resultado de la función **`regexpo.ErrorARMA()`**.
- **`sigma2`**: numérico, con la estimación de la varianza de las innovaciones del modelo ARMA (es decir $\hat{\sigma}_a^2$), definido para el error estructural del modelo de regresión. Puede ser extraído de la siguiente manera: **`nombre_objeto$sigma2`**, donde '**`nombre_objeto`**' es el nombre del objeto R donde se guarda el resultado de la función **`regexpo.ErrorARMA()`**.
- **`forecast`**: Objeto tipo ts con los pronósticos puntuales de la respuesta para **`h=ncol(newdata)`** periodos después del ajuste. Puede ser extraído de la siguiente manera: **`nombre_objeto$forecast`**, donde '**`nombre_objeto`**' es el nombre del objeto R donde se guarda el resultado de la función **`regexpo.ErrorARMA()`**.

Pasos iniciales: Cargar librerías y funciones de usuario necesarias

```
library(forecast);library(lmtest);library(TSA)
source("https://raw.githubusercontent.com/NelfiGonzalez/Funciones-de-Usuario-Estadistica-III/main/Funciones-Criterios.Informacion-Calidad.Intervalos.R")
source("https://raw.githubusercontent.com/NelfiGonzalez/Funciones-de-Usuario-Estadistica-III/main/Funciones-BP.LB.test-pruebaDW1.R")
source("https://raw.githubusercontent.com/NelfiGonzalez/Funciones-de-Usuario-Estadistica-III/main/Funcion-regexpo.ErrorARMA.R")
```

Nota: En los siguientes ejemplos, el paso 1) solo se realiza una vez, no es necesario correrlo con cada modelo distinto en los errores. En los pasos 2) a 7), tenga en cuenta que, por cada modelo distinto a considerar en los errores, debe crear con diferentes nombres los objetos que guarden valores ajustados, residuos, pronósticos, etc.

a. Modelo de regresión exponencial con errores ARMA(p,q) con todos los parámetros. Considere el siguiente ejemplo, $Y_t = \exp\left(\beta_0 + \beta_1 t + \beta_2 t^2 + \sum_{i=1}^{11} \delta_i I_{i,t}\right) + E_t$, donde $E_t = \sum_{j=1}^{p} \phi_j E_{t-j} + a_t + \sum_{i=1}^{q} \theta_i a_{t-i}$, con a_t un $RB \sim N(0, \sigma_a^2)$,

1) Lectura datos, definición de variables para ajustes y pronósticos (en este ejemplo se asume 12 pronósticos ex – post).

```
datos=read.table(file.choose(),....) #leer conforme a la base de datos asignada
datos=ts(datos,freq=12,start=c(1960,1)) #convertir en serie según frecuencia y fecha inicial
n=length(datos)-12 #Defina longitud de los datos para ajuste con validación cruzada. Aquí se recorta en 12 datos
#Defina el índice de tiempo y sus potencias según polinomio planteado para Tt
t=1:n; t2=t^2
yt=ts(datos[t],freq=12,start=c(1960,1)) #serie recortada
#Defina la matriz de variables indicadoras
Mes=seasonaldummy(yt)

#Defina la matriz de variables predictoras de tendencia y estacionalidad:
X=data.frame(t,t2,Mes)

Definiendo variables para pronósticos de la estructura exponencial
tnuevo=(n+1):length(datos)
t2nuevo=tnuevo^2
Mesnuevo=seasonaldummy(yt,h=12) #Horizonte de pronóstico h igual a longitud de recorte de la serie

Xnuevo=data.frame(t=tnuevo,t2=t2nuevo,Mesnuevo) #matriz predictores en el pronóstico

ytf=ts(datos[tnuevo],freq=12,start=c(1974,1)) #los datos no usados en ajuste. Fechas y frecuencias según su caso

#Crear vector de nombres para los parámetros en la estructura exponencial (como en el trabajo 1)
parammod=c(paste0("beta",0:2),paste0("delta",1:11))
```

2) Ajuste del modelo de regresión exponencial con error ARMA.

```
#indique el valor correspondiente a p y q en la función de usuario regexpo.ErrorARMA, y los demás argumento de esta función
modelo=regexpo.ErrorARMA(respuesta=yt,names.param=parammod,data=X,newdata=Xnuevo,order=c(p,0,q),method="ML")
```

3) Construya tabla de parámetros estimados, valores P corresponden a $P(|t_{n-k}| > |T_0|)$, con k la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA).

```
#Construya tabla de parámetros estimados con estadísticos T0 y valores P para cada parámetro
coef(modelo)
```

4) Obtenga valor ajustado y gráfica del ajuste de la serie en escala original y de los residuales de ajuste $\hat{a}_t$.

```
ythat=fitted(modelo) #Ajuste total de la serie

#Gráfico de la serie y su ajuste
plot(datos)
lines(ythat,col=2)
legend("topleft",legend=c("datos","ajuste modelo"),lty=1,col=1:2)

#Gráficos de residuales del ajuste
plot(residuals(modelo))
abline(h=c(-2*sqrt(modelo$sigma2),0,2*sqrt(modelo$sigma2)),lty=2)
plot(as.numeric(fitted(modelo)),residuals(modelo))
abline(h=c(-2*sqrt(modelo$sigma2),0,2*sqrt(modelo$sigma2)),lty=2)
```

5) Cálculo de AIC y BIC, versión $\exp\left(C_n^*(p)\right)$, con el número de parámetros k siendo la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA).

```
k=modelo$p; k #número parámetros
Criteriosmodelo=exp.crit.inf.resid(residuales=residuals(modelo),n.par=k); Criteriosmodelo
```

6) Valide supuestos sobre el error de ajuste a_t

```
#ACF sobre residuales de ajuste. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
acf(as.numeric(residuals(modelo)),ci.type="ma",lag.max=m,main="ACF modelo",ci.col=2)

#PACF sobre residuales de ajuste. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
pacf(as.numeric(residuals(modelo)),lag.max=m,main="PACF modelo",ci.col=2)

BP.LB.test(residuals(modelo),maxlag=m,type="Ljung") #test Ljung-Box use máximo m igual al de ACF y PACF

#Normalidad sobre residuales de ajuste en el modelo. Sólo si no se rechaza supuesto de ruido blanco
shapiro.test(residuals(modelo))

win.graph()
qqnorm(residuals(modelo),main="Gráfico de normalidad residuos de ajuste modelo")
qqline(residuals(modelo),col=2)
```

7) Pronósticos para la validación cruzada. No es posible obtener pronósticos por I.P

```
ytpron=modelo$forecast; ytpron

#Medidas precisión pronósticos puntuales
accuracy(ytpron,ytf)
```

b. Modelo de regresión exponencial con errores ARMA(p,q) con algunos de los parámetros fijos en cero. Considere el siguiente ejemplo,
$Y_t = \exp\left(\beta_0 + \beta_1 t + \beta_2 t^2 + \sum_{i=1}^{11} \delta_i I_{i,t}\right) + E_t$, donde $E_t = \phi_1 E_{t-1} + \phi_9 E_{t-9} + a_t + \theta_{12} a_{t-12}$, con a_t un $RB \sim N(0, \sigma_a^2)$. Se procede de la siguiente manera:

1) Lectura datos, definición de variables para ajustes y pronósticos (en este ejemplo se asume 12 pronósticos ex – post).

```
datos=read.table(file.choose(),....) #leer conforme a la base de datos asignada
datos=ts(datos,freq=12,start=c(1960,1)) #convertir en serie según frecuencia y fecha inicial
n=length(datos)-12 #Defina longitud de los datos para ajuste con validación cruzada. Aquí se recorta en 12 datos
#Defina el índice de tiempo y sus potencias según polinomio planteado para Tt
t=1:n; t2=t^2
yt=ts(datos[t],freq=12,start=c(1960,1)) #serie recortada
#Defina la matriz de variables indicadoras
Mes=seasonaldummy(yt)

#Defina la matriz de variables predictoras de tendencia y estacionalidad:
X=data.frame(t,t2,Mes)

Definiendo variables para pronósticos de la estructura exponencial
tnuevo=(n+1):length(datos)
t2nuevo=tnuevo^2
Mesnuevo=seasonaldummy(yt,h=12) #Horizonte de pronóstico h igual a longitud de recorte de la serie

Xnuevo=data.frame(t=tnuevo,t2=t2nuevo,Mesnuevo) #matriz predictores en el pronóstico

ytf=ts(datos[tnuevo],freq=12,start=c(1974,1)) #los datos no usados en ajuste. Fechas y frecuencias según su caso

Crear vector de nombres para los parámetros en la estructura exponencial
parammod=c(paste0("beta",0:2),paste0("delta",1:11))
```

2) Ajuste del modelo de regresión exponencial con error ARMA. Dado que no se usan todos los parámetros del modelo ARMA(p,q) en los errores estructurales E_t, en la función de usuario **regexpo.ErrorARMA()** debe especificarse cuáles parámetros van en el modelo y cuáles se fijan en cero. Para ello, se usa el argumento fixed igual a un vector de longitud (p+q). Las entradas o valores de este vector se ordenan de la siguiente manera: Los primeros p valores son relativos a los coeficientes ϕ_j, los siguientes q valores hacen referencia a los coeficientes θ_i. En cada entrada del vector se coloca un 0 si el parámetro no va en el modelo y un NA, si el respectivo parámetro debe ir en el modelo. Para el ejemplo, como el error estructural es un ARMA(9,12) pero sólo los coeficientes ϕ_1, ϕ_9 y θ_{12} van en la estructura ARMA, el argumento fixed en la función Arima debe ser especificado así (OJO: rep(a,b) repite b veces el argumento a),
fixed=c(NA,rep(0,7),NA,rep(0,11),NA)
que es equivalente a escribir de forma extendida lo siguiente,

fixed=c(NA,0,0,0,0,0,0,0,NA,0,0,0,0,0,0,0,0,0,0,0,NA)
donde el color del caracter indica
Las entradas para los parámetros parte AR(9)
Las entradas para los parámetros parte MA(12)

```
#indique el valor correspondiente a p y q en la función de usuario regexpo.ErrorARMA, y los demás argumento de esta función
#incluyendo el vector para el argumento fixed
modelo=regexpo.ErrorARMA(respuesta=yt,names.param=parammod,data=X,newdata=Xnuevo,order=c(9,0,12),
                         fixed=c(NA,rep(0,7),NA,rep(0,11),NA),method="ML")
```

3) Construya tabla de parámetros estimados, valores P corresponden a $P(|t_{n-k}| > |T_0|)$, con k la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA).

```
#Construya tabla de parámetros estimados con estadísticos T0 y valores P para cada parámetro
coef(modelo)
```

4) Obtenga valor ajustado y gráfica del ajuste de la serie en escala original y de los residuales de ajuste $\hat{a}_t$.

```
ythat=fitted(modelo) #Ajuste total de la serie

#Gráfico de la serie y su ajuste
plot(datos)
lines(ythat,col=2)
legend("topleft",legend=c("datos","ajuste modelo"),lty=1,col=1:2)

#Gráficos de residuales del ajuste
plot(residuals(modelo))
abline(h=c(-2*sqrt(modelo$sigma2),0,2*sqrt(modelo$sigma2)),lty=2)
plot(as.numeric(fitted(modelo)),residuals(modelo))
abline(h=c(-2*sqrt(modelo$sigma2),0,2*sqrt(modelo$sigma2)),lty=2)
```

5) Cálculo de AIC y BIC, versión $\exp(C_n^*(p))$, con el número de parámetros k siendo la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA).

```
k=modelo$p; k #número parámetros
Criteriosmodelo=exp.crit.inf.resid(residuales=residuals(modelo),n.par=k); Criteriosmodelo
```

6) Valide supuestos sobre el error de ajuste a_t

```
#ACF sobre residuales de ajuste. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
acf(as.numeric(residuals(modelo)),ci.type="ma",lag.max=m,main="ACF modelo",ci.col=2)

#PACF sobre residuales de ajuste. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
pacf(as.numeric(residuals(modelo)),lag.max=m,main="PACF modelo",ci.col=2)

BP.LB.test(residuals(modelo),maxlag=m,type="Ljung") #test Ljung-Box use máximo m igual al de ACF y PACF

#Normalidad sobre residuales de ajuste en el modelo. Sólo si no se rechaza supuesto de ruido blanco
shapiro.test(residuals(modelo))

win.graph()
qqnorm(residuals(modelo),main="Gráfico de normalidad residuos de ajuste modelo")
qqline(residuals(modelo),col=2)
```

7) Pronósticos para la validación cruzada. No es posible obtener pronósticos por I.P

```
ytpron=modelo$forecast; ytpron

#Medidas precisión pronósticos puntuales
accuracy(ytpron,ytf)
```

c. Modelo de regresión exponencial con errores ARMA(p,q)(P,Q)[s]. Considere el siguiente ejemplo, $Y_t = \exp\left(\beta_0 + \beta_1 t + \beta_2 t^2 + \sum_{i=1}^{11} \delta_i I_{i,t}\right) + E_t$, donde E_t es un ARMA(p,q)(P,Q)[12], es decir $\phi_p(B)\Phi_P(B^{12})E_t = \theta_q(B)\Theta_Q(B^{12})a_t$, con a_t un $RB \sim N(0,\sigma_a^2)$. Entonces, el modelo queda como

$Y_t = \exp\left(\beta_0 + \beta_1 t + \beta_2 t^2 + \sum_{i=1}^{11} \delta_i I_{i,t}\right) + E_t$, con

$$E_t = \sum_{j=1}^{p} \phi_j E_{t-j} + \sum_{k=1}^{P} \Phi_k\, E_{t-12*k} - \sum_{k=1}^{P} \Phi_k \sum_{j=1}^{p} \phi_j E_{t-j-12*k} + a_t + \sum_{i=1}^{q} \theta_i a_{t-i} + \sum_{l=1}^{Q} \Theta_l a_{t-12*l} + \sum_{l=1}^{Q} \Theta_l \sum_{i=1}^{q} \theta_i a_{t-i-12*l}$$

y a_t un $RB \sim N(0,\sigma_a^2)$

1) Lectura datos, definición de variables para ajustes y pronósticos (en este ejemplo se asume 12 pronósticos ex – post).

```
datos=read.table(file.choose(),....) #leer conforme a la base de datos asignada
datos=ts(datos,freq=12,start=c(1960,1)) #convertir en serie según frecuencia y fecha inicial
n=length(datos)-12 #Defina longitud de los datos para ajuste con validación cruzada. Aquí se recorta en 12 datos
#Defina el índice de tiempo y sus potencias según polinomio planteado para Tt
t=1:n; t2=t^2
yt=ts(datos[t],freq=12,start=c(1960,1)) #serie recortada
#Defina la matriz de variables indicadoras
Mes=seasonaldummy(yt)

#Defina la matriz de variables predictoras de tendencia y estacionalidad:
X=data.frame(t,t2,Mes)

Definiendo variables para pronósticos de la estructura exponencial
tnuevo=(n+1):length(datos)
t2nuevo=tnuevo^2
```

```
Mesnuevo=seasonaldummy(yt,h=12) #Horizonte de pronóstico h igual a longitud de recorte de la serie

Xnuevo=data.frame(t=tnuevo,t2=t2nuevo,Mesnuevo) #matriz predictores en el pronóstico

ytf=ts(datos[tnuevo],freq=12,start=c(1974,1)) #los datos no usados en ajuste. Fechas y frecuencias según su caso

Crear vector de nombres para los parámetros en la estructura exponencial (como en el trabajo 1)
parammod=c(paste0("beta",0:2),paste0("delta",1:11))
```

2) Ajuste del modelo de regresión exponencial con error ARMA.. El orden p, q se especifica con order=c(p,0,q), mientras que el orden P, Q se especifica con seasonal=list(order=c(P,0,Q))

```
#indique el valor correspondiente a p, q, P y Q en la función de usuario regexpo.ErrorARMA, y los demás argumento de esta
#función
Modelo=regexpo.ErrorARMA(respuesta=yt,names.param=parammod,data=X,newdata=Xnuevo,order=c(p,0,q),seasonal=list(order=c(P,0,Q)),
                         method="ML")
```

3) Construya tabla de parámetros estimados, valores P corresponden a $P(|t_{n-k}| > |T_0|)$, con k la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA).

```
#Construya tabla de parámetros estimados con estadísticos T0 y valores P para cada parámetro
coef(modelo)
```

4) Obtenga valor ajustado y gráfica del ajuste de la serie en escala original y de los residuales de ajuste $\hat{a}_t$.

```
ythat= fitted(modelo) #Ajuste total de la serie

#Gráfico de la serie y su ajuste
plot(datos)
lines(ythat,col=2)
legend("topleft",legend=c("datos","ajuste modelo"),lty=1,col=1:2)

#Gráficos de residuales del ajuste
plot(residuals(modelo))
abline(h=c(-2*sqrt(modelo$sigma2),0,2*sqrt(modelo$sigma2)),lty=2)
plot(as.numeric(fitted(modelo)),residuals(modelo))
abline(h=c(-2*sqrt(modelo$sigma2),0,2*sqrt(modelo$sigma2)),lty=2)
```

5) Cálculo de AIC y BIC, versión $\exp(C_n^*(p))$, con el número de parámetros k siendo la suma del número de parámetros en las estructuras de tendencia, estacionalidad y errores ARMA).

```
k=modelo$p; k #número parámetros
Criteriosmodelo =exp.crit.inf.resid(residuales=residuals(modelo),n.par=k); Criteriosmodelo
```

6) Valide supuestos sobre el error de ajuste a_t

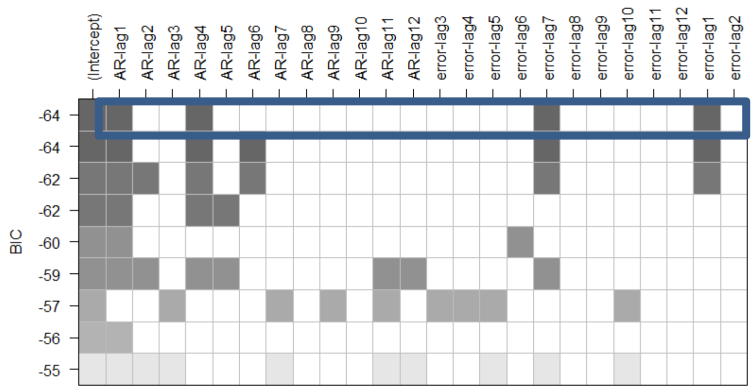
```
#ACF sobre residuales de ajuste. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
acf(as.numeric(residuals(modelo)),ci.type="ma",lag.max=m,main="ACF modelo",ci.col=2)

#PACF sobre residuales de ajuste. Use valor para m el que se indica en la guía del trabajo
win.graph(width=4.875,height=3.5,pointsize=8)
pacf(as.numeric(residuals(modelo)),lag.max=m,main="PACF modelo",ci.col=2)

BP.LB.test(residuals(modelo),maxlag=m,type="Ljung") #test Ljung-Box use máximo m igual al de ACF y PACF

#Normalidad sobre residuales de ajuste en el modelo. Sólo si no se rechaza supuesto de ruido blanco
shapiro.test(residuals(modelo))

win.graph()
qqnorm(residuals(modelo),main="Gráfico de normalidad residuos de ajuste modelo")
qqline(residuals(modelo),col=2)
```

7) Pronósticos para la validación cruzada. No es posible obtener pronósticos por I.P

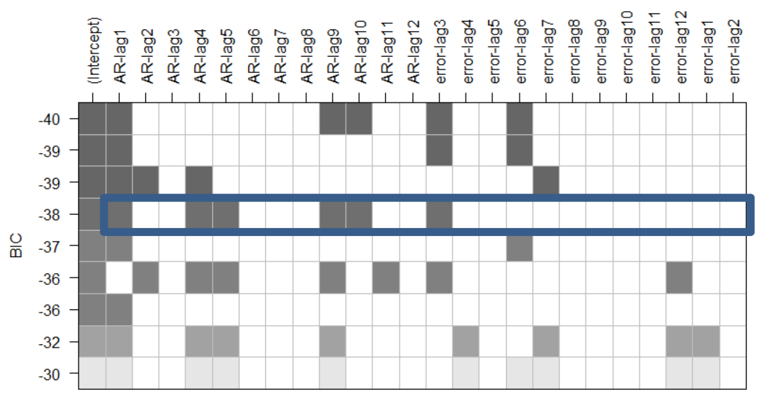
```
ytpron=modelo$forecast; ytpron

#Medidas precisión pronósticos puntuales
accuracy(ytpron,ytf)
```

APÉNDICE C. Gráficos generados con función armasubsets sobre residuos de ajuste del modelo global listado en la Tabla 1.
Para las gráficas siguientes, tenga en cuenta que, método ‘ml’ se refiere a usar en función armasubsets el argumento ar.method=’ml’.

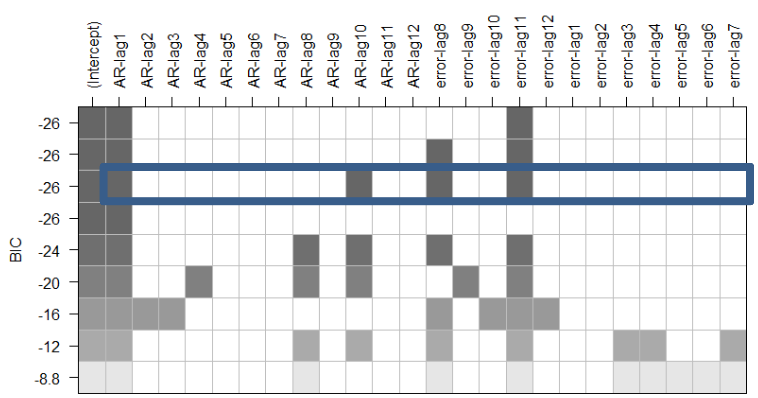
En Datos 1: armasubsets 12x12, método ‘ml’, renglón 1 y agregar θ_5



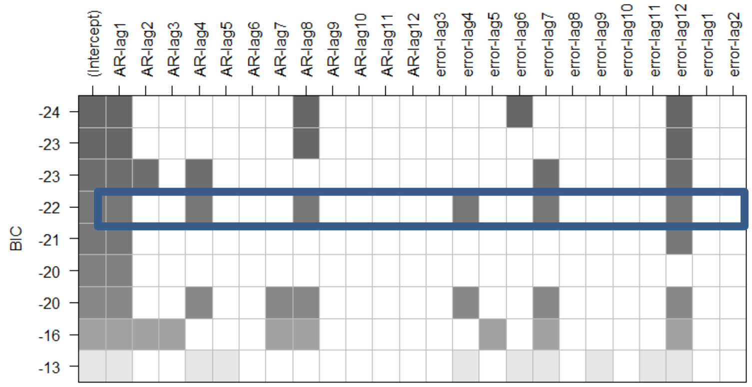
En Datos 2: armasubsets 12x12, método ‘ml’, renglón 4:



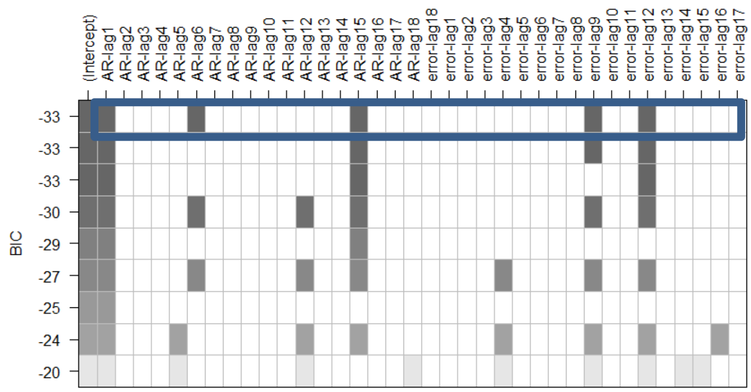
En Datos 3: armasubsets 12x12, método ‘ml’, renglón 3:



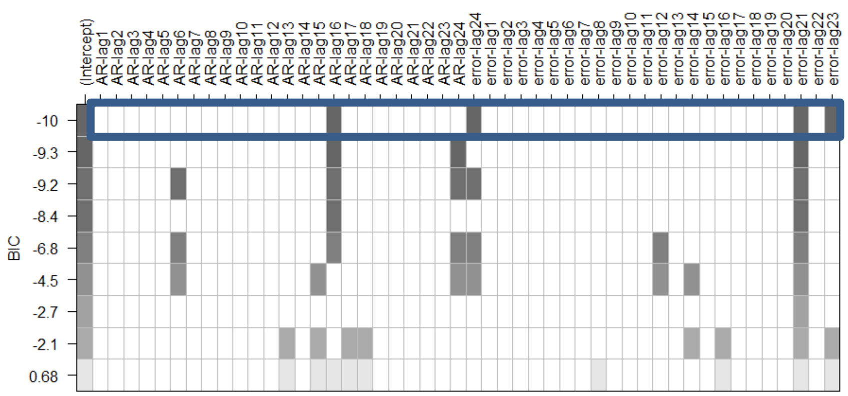
En Datos 4: armasubsets 12x12, método ‘ml’, renglón 4 y agregar a ϕ_{10}:



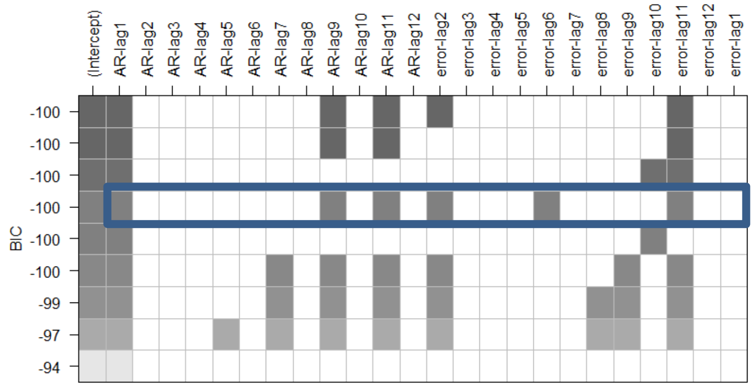
En Datos 5: armasubsets 18x18, método ‘ml’, renglón 1, agregar ϕ_{13} y estimar con método CSS-ML en Arima()



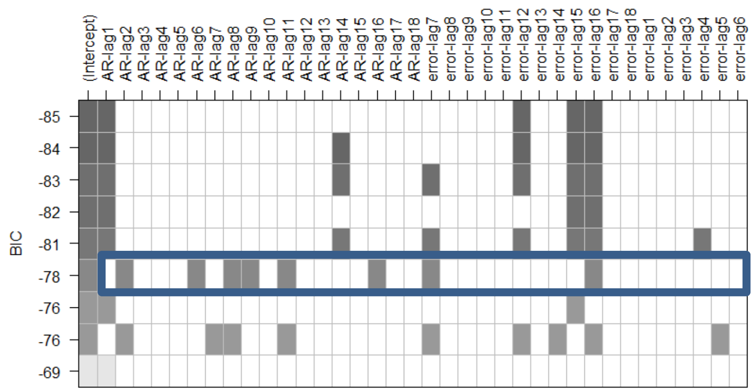
En Datos 6: armasubsets 24x24, método ‘ml’, renglón 1 y agregar ϕ_1



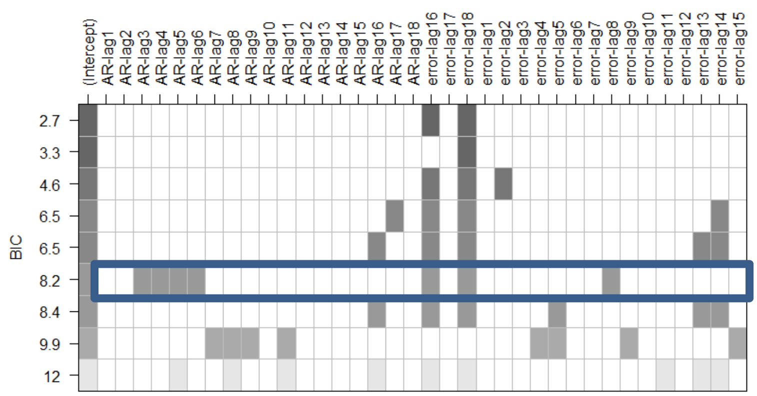
En Datos 7: armasubsets 12x12, método 'ml', renglón 4 y agregar ϕ_{10}



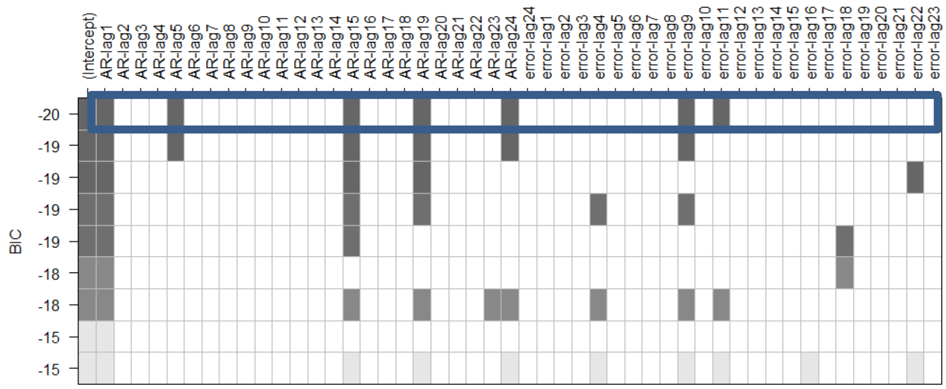
En Datos 8: armasubsets 18x18, método 'ml', renglón 6 y agregar ϕ_1, ϕ_{10}



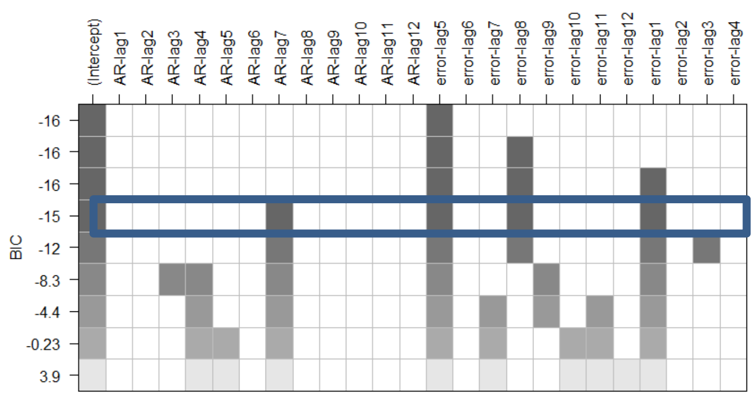
En Datos 9: armasubsets 18x18, método 'ml', renglón 6 y agregar ϕ_1



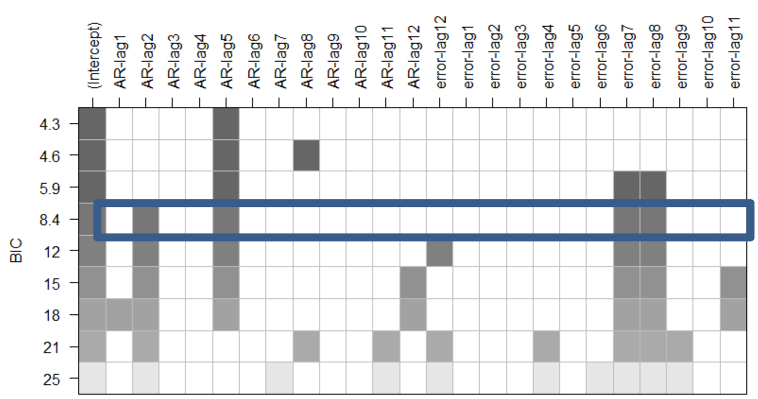
En Datos 10: armasubsets 24x24, método ‘ml’, renglón 1



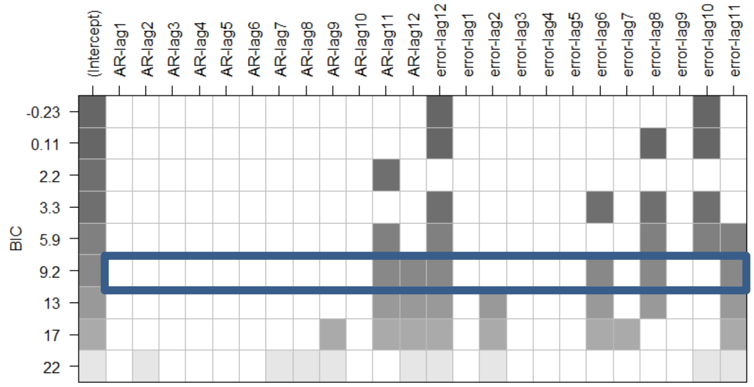
En Datos 11: armasubsets 12x12, método ‘ml’, renglón 4



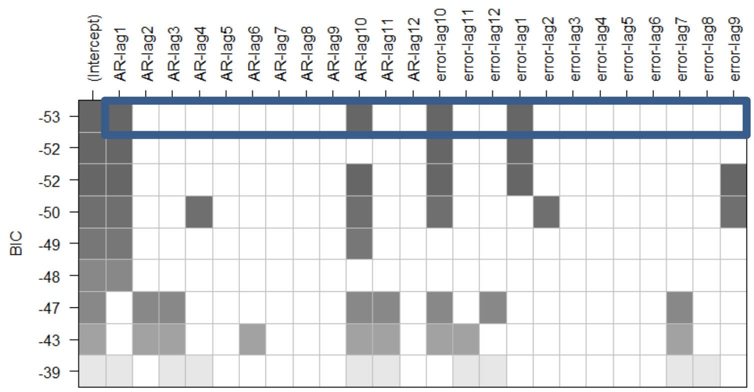
En Datos 12: armasubsets 12x12, método ‘ml’, renglón 4



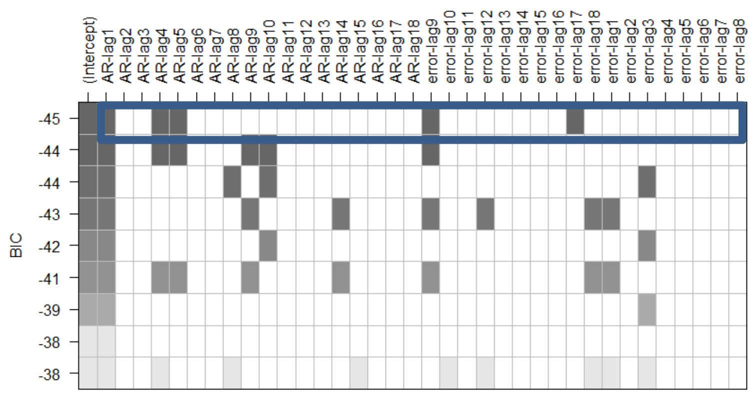
En Datos13: armasubsets 12x12, método 'ml', renglón 6 y agregar θ_1, θ_9



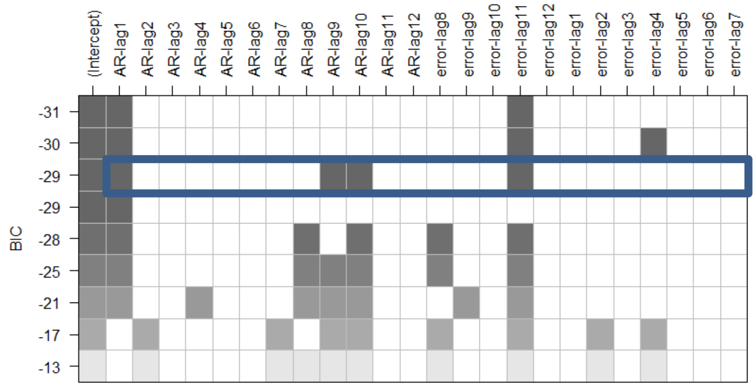
En Datos 14: armasubsets 12x12, método 'ml', renglón 1 y agregar θ_4



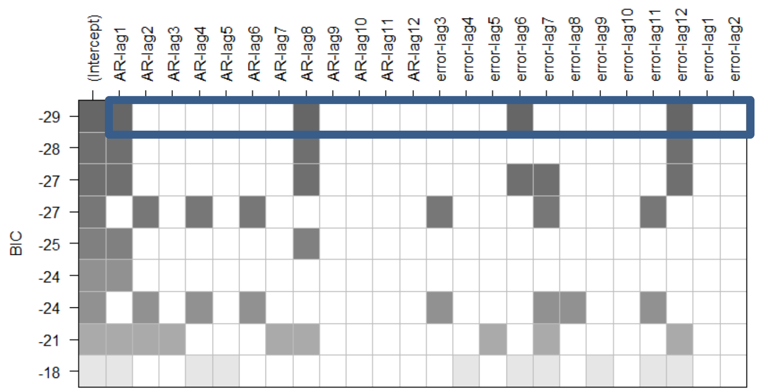
En Datos 16: armasubsets 18x18, método 'ml', renglón 1 y agregar ϕ_2, θ_{13}



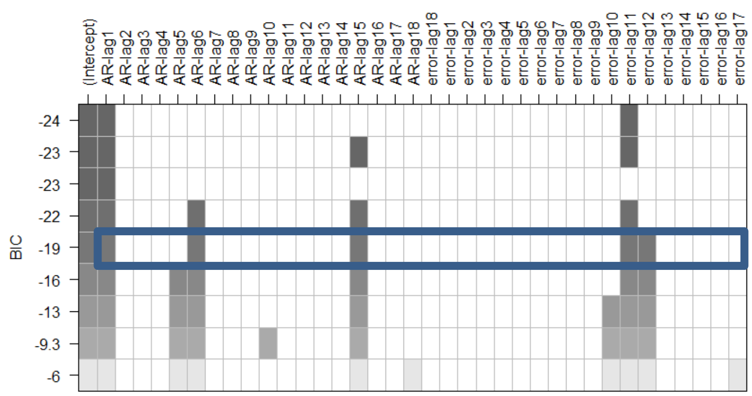
En Datos 17: armasubsets 12x12, método 'ml', renglón 3 y agregar θ_4



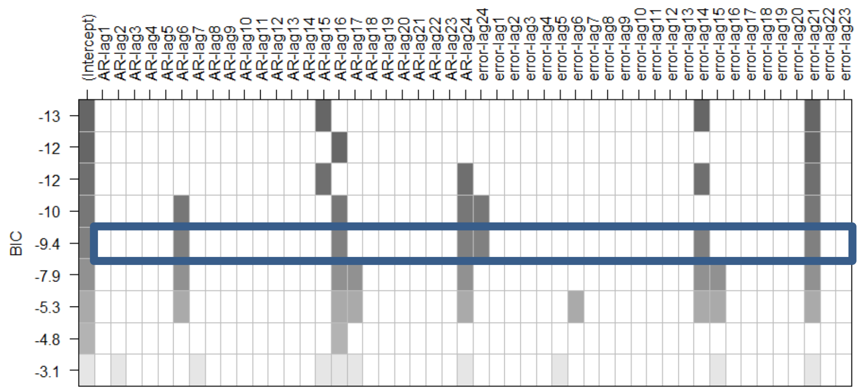
En Datos 18: armasubsets 12x12, método 'ml', renglón 1 y agregar ϕ_4



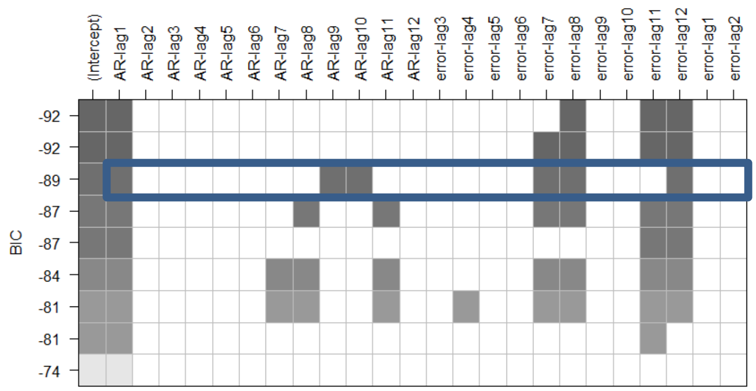
En Datos 19: armasubsets 18x18, método 'ml', renglón 5



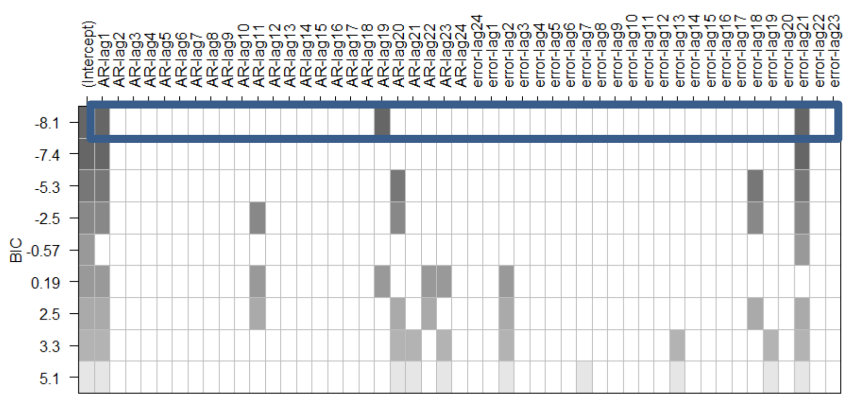
En Datos 20: armasubsets 24x24, método ‘ml’, renglón 5, agregar ϕ_1, fijar valores de estimaciones de parámetros estructurales iguales a los del modelo global y estimar con método ‘CSS-ML’ en función Arima()



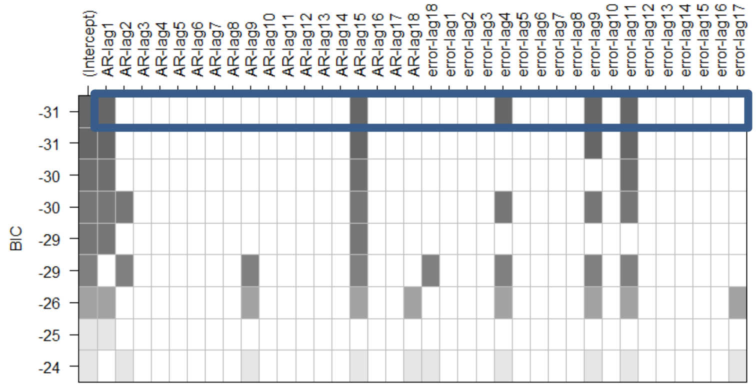
En Datos 22: armasubsets 12x12, método ‘ml’, renglón 3 y agregar ϕ_5, ϕ_6



En Datos 23: armasubsets 24x24, método ‘ml’, renglón 1 y agregar θ_{22}



En Datos 24: armasubsets 18x18, método ‘ml’, renglón 1 y agregar θ_5



En Datos 27: armasubsets 12x12, método ‘ml’, renglón 2 y agregar θ_1