Series de tiempo univariadas - Presentación 9

Mauricio Alejandro Mazo Lopera

Universidad Nacional de Colombia
Facultad de Ciencias
Escuela de Estadística
Medellín

Comportamiento de las ACF y PACF teóricas

Proceso	**ACF**	**PACF**
AR(p)	Decaimiento exponencial y/o onda senoidal amortiguada	Corte después del rezago p
MA(q)	Corte después del rezago q	Decaimiento exponencial y/o onda senoidal amortiguada
ARMA(p, q)	Decaimiento exponencial y/0 onda senoidal amortiguada	Decaimiento exponencial y/o onda senoidal amortiguada

Después de ver cuál es el comportamiento teórico de las ACF y PACF de los modelos:

- AR(p): $X_t = \phi_0 + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + w_t$
- MA(q): $X_t = \mu + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}$
- ARMA(p, q): $X_t = \alpha + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}$

Podemos aplicar esto para identificar los ordenes p y/o q del modelo que queremos ajustar a una serie de tiempo, a partir de las ACF y PACF muestrales.

Estimación de modelos ARMA(p, q)

Después de ver cuál es el comportamiento teórico de las ACF y PACF de los modelos:

- AR(p): $X_t = \phi_0 + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + w_t$
- MA(q): $X_t = \mu + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}$
- ARMA(p, q): $X_t = \alpha + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}$

Podemos aplicar esto para identificar los ordenes p y/o q del modelo que queremos ajustar a una serie de tiempo, a partir de las ACF y PACF muestrales.

Luego, el **objetivo** es **estimar** los parámetros ϕ_0, ϕ_1, ϕ_2, ..., ϕ_p, μ, θ_1, θ_2, ..., θ_q y σ_w^2.

Algunos de los métodos de estimación son:

- Método de los momentos.
- Mínimos cuadrados condicional.
- Máxima verosimilitud condicional.

Veamos una breve presentación de estos métodos de estimación (para ver una explicación más detallada ver TEXTO 3 - TEÓRICO - Wei-Time series analysis - Capítulo 7):

Método de los momentos:

Este métodos funciona bien principalmente para el proceso AR(p), ya que para este se cumplen las ecuaciones de Yule-Walker:

$$\begin{aligned}
\rho_1 &= \phi_1 + \phi_2\rho_1 + \cdots + \phi_p\rho_{p-1} \\
\rho_2 &= \phi_1\rho_1 + \phi_2 + \cdots + \phi_p\rho_{p-2} \\
&\vdots \\
\rho_p &= \phi_1\rho_{p-1} + \phi_2\rho_{p-2} + \cdots + \phi_p
\end{aligned}$$

Método de los momentos:

Este métodos funciona bien principalmente para el proceso AR(p), ya que para este se cumplen las ecuaciones de Yule-Walker:

$$\begin{aligned}
\rho_1 &= \phi_1 + \phi_2\rho_1 + \cdots + \phi_p\rho_{p-1} \\
\rho_2 &= \phi_1\rho_1 + \phi_2 + \cdots + \phi_p\rho_{p-2} \\
&\vdots \\
\rho_p &= \phi_1\rho_{p-1} + \phi_2\rho_{p-2} + \cdots + \phi_p
\end{aligned}$$

Reemplazando las ACF $\rho_1, \ldots, \rho_p$ por sus estimaciones muestrales $\widehat{\rho}_1, \ldots, \widehat{\rho}_p$ en este sistema, llegamos a los estimadores:

$$\begin{bmatrix} \widehat{\phi}_1 \\ \widehat{\phi}_2 \\ \vdots \\ \widehat{\phi}_p \end{bmatrix} = \begin{bmatrix} 1 & \widehat{\rho}_1 & \widehat{\rho}_2 & \cdots & \widehat{\rho}_{p-2} & \widehat{\rho}_{p-1} \\ \widehat{\rho}_1 & 1 & \widehat{\rho}_1 & \cdots & \widehat{\rho}_{p-3} & \widehat{\rho}_{p-2} \\ \vdots & & & & & \vdots \\ \widehat{\rho}_{p-1} & \widehat{\rho}_{p-2} & \widehat{\rho}_{p-3} & \cdots & \widehat{\rho}_1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \widehat{\rho}_1 \\ \widehat{\rho}_2 \\ \vdots \\ \widehat{\rho}_p \end{bmatrix}$$

Método de los momentos:

Este métodos funciona bien principalmente para el proceso AR(p), ya que para este se cumplen las ecuaciones de Yule-Walker:

$$\begin{aligned}
\rho_1 &= \phi_1 + \phi_2\rho_1 + \cdots + \phi_p\rho_{p-1} \\
\rho_2 &= \phi_1\rho_1 + \phi_2 + \cdots + \phi_p\rho_{p-2} \\
&\vdots \\
\rho_p &= \phi_1\rho_{p-1} + \phi_2\rho_{p-2} + \cdots + \phi_p
\end{aligned}$$

Reemplazando las ACF $\rho_1, \ldots, \rho_p$ por sus estimaciones muestrales $\widehat{\rho}_1, \ldots, \widehat{\rho}_p$ en este sistema, llegamos a los estimadores:

$$\begin{bmatrix} \widehat{\phi}_1 \\ \widehat{\phi}_2 \\ \vdots \\ \widehat{\phi}_p \end{bmatrix} = \begin{bmatrix} 1 & \widehat{\rho}_1 & \widehat{\rho}_2 & \cdots & \widehat{\rho}_{p-2} & \widehat{\rho}_{p-1} \\ \widehat{\rho}_1 & 1 & \widehat{\rho}_1 & \cdots & \widehat{\rho}_{p-3} & \widehat{\rho}_{p-2} \\ \vdots & & & & & \vdots \\ \widehat{\rho}_{p-1} & \widehat{\rho}_{p-2} & \widehat{\rho}_{p-3} & \cdots & \widehat{\rho}_1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \widehat{\rho}_1 \\ \widehat{\rho}_2 \\ \vdots \\ \widehat{\rho}_p \end{bmatrix}$$

El estimador de σ_w^2 está dado por:

$$\widehat{\sigma}_w^2 = \widehat{\gamma}(0)(1 - \widehat{\phi}_1\widehat{\rho}_1 - \widehat{\phi}_2\widehat{\rho}_2 - \cdots - \widehat{\phi}_p\widehat{\rho}_p) \quad \text{(¿por qué?)}$$

Máxima verosimilitud condicional:

EL método de los momentos tiene como desventaja que no funciona muy bien para estimar los parámetros del modelo MA(q).

Máxima verosimilitud condicional:

EL método de los momentos tiene como desventaja que no funciona muy bien para estimar los parámetros del modelo MA(q).Esto abre paso a otro método conocido como máxima verosimilitud condicional, en donde se asume que w_t es un ruido blanco Gaussiano y se busca maximizar la log-verosimilitud condicional:

$$\ln(L_*(\widetilde{\phi}, \widetilde{\theta}, \alpha, \sigma_w^2)) = -\frac{n}{2}\ln(2\pi\sigma_w^2) - \frac{S_*(\widetilde{\phi}, \alpha, \widetilde{\theta})}{2\sigma_w^2}$$

donde $\widetilde{\phi} = (\phi_1, \phi_2, \ldots, \phi_p)^\top$, $\widetilde{\theta} = (\theta_1, \theta_2, \ldots, \theta_q)^\top$ y la **suma condicional de cuadrados está dada por**:

$$S_*(\widetilde{\phi}, \alpha, \widetilde{\theta}) = \sum_{t=1}^{n} w_t^2$$

con

$$w_t = X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} - \alpha - \theta_1 w_{t-1} - \cdots - \theta_q w_{t-q}$$

Máxima verosimilitud condicional:

El estimador de σ_w^2 está dado por:

$$\widehat{\sigma}_w^2 = \frac{S_*(\widehat{\widetilde{\phi}}, \widehat{\alpha}, \widehat{\widetilde{\theta}})}{g.l.}$$

donde $g.l.$ son los grados de libertad iguales al tamaño de muestra n menos el número de parámetros estimados.

Máxima verosimilitud condicional:

El estimador de σ_w^2 está dado por:

$$\widehat{\sigma}_w^2 = \frac{S_*(\widehat{\widetilde{\phi}}, \widehat{\alpha}, \widehat{\widetilde{\theta}})}{g.l.}$$

donde $g.l.$ son los grados de liberdad iguales al tamaño de muestra n menos el número de parámetros estimados.

En la mayoría de paquetes estadísticos minimizan la **suma condicional de cuadrados** $S_*(\widetilde{\phi}, \alpha, \widetilde{\theta})$ con el fin de obtener valores iniciales en el proceso de maximización de la log-verosimilitud condicional $\ln(L_*(\widetilde{\phi}, \widetilde{\theta}, \alpha, \sigma_w^2))$.

Estimación de modelos ARMA(p, q) en R:

En R exiten distintas funciones para estimar los parámetros de un modelo ARMA(p, q), entre los cuales están:

- **arima**: Hace parte del paquete **stats**. Para mayor información escribir **?stats::arima** en la consola del RStudio.
- **Arima**: Hace parte del paquete **forecast**. Para mayor información escribir **?forecast::Arima** en la consola del RStudio.
- **arima**: Hace parte del paquete **TSA**. Para mayor información escribir **?TSA::arima** en la consola del RStudio.
- **arma**: Hace parte del paquete **tseries**. Para mayor información escribir **?tseries::arma** en la consola del RStudio. Solo minimiza la suma de cuadrados condicionales y no pasa por la verosimilitud.

La tres primeras funciones estiman unos valores iniciales minimizando la suma de cuadrados condicionales y luego usan estos valores para maximizar la función de verosimilitud.

Estimación de modelos ARMA(p, q) en Python:

Dentro del módulo **statmodels** del Python existe un submódulo conocido como **tsa** por sus siglas en inglés *time series analysis* (sitio web AQUÍ).

Este submódulo contiene el proceso **arima_model** cuya función **ARIMA** permite estimar los parámetros del modelo ARMA también utilizando el método de máxima verosimilitud condicional.

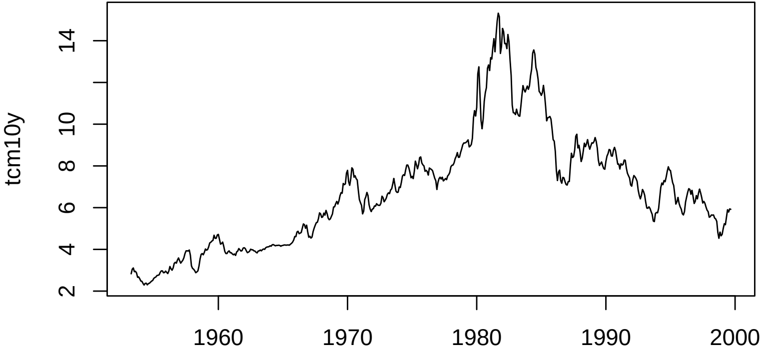
Para cargarlo utilice los códigos

```
from statsmodels.tsa.arima_model import ARIMA
```

Estimación de modelos ARMA(p, q):

Consideremos un ejemplo simple para ver cómo trabajar con las funciones del R que vimos:

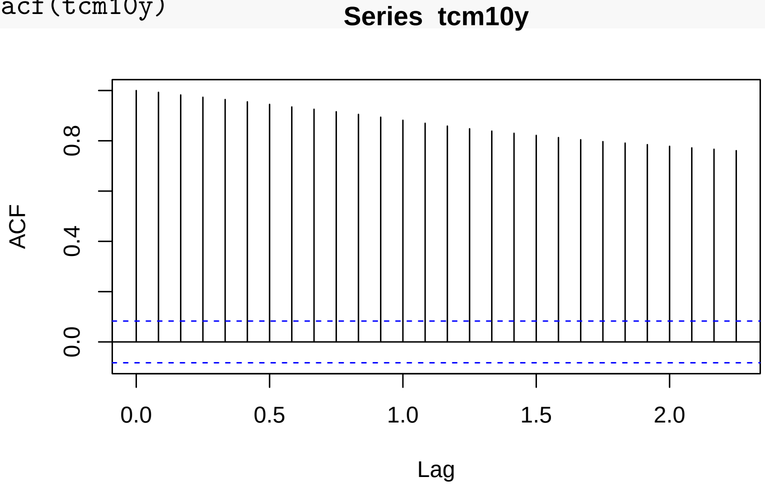
```
require(tseries)
data(tcm)
plot(tcm10y) # Escriba ?tcm para más información
```



Estimación de modelos ARMA(p, q):

Encontremos las ACF y PACF muestrales:

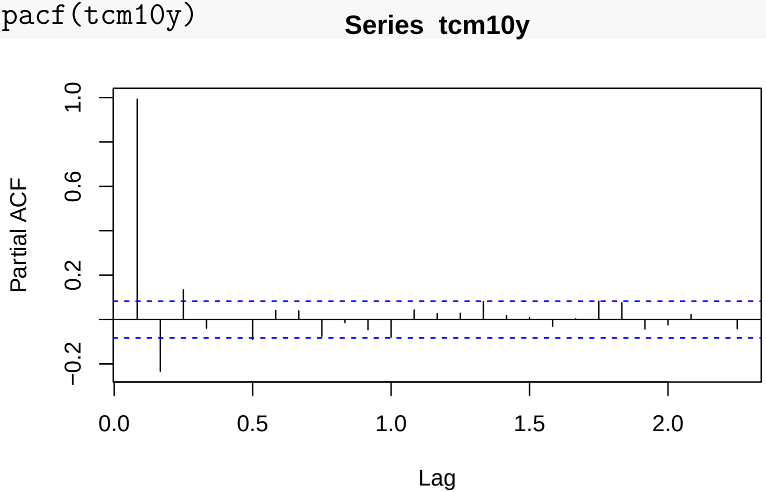
```
acf(tcm10y)
```



Se observa un decaimiento lento debido a la tendencia en la serie.

Estimación de modelos ARMA(p, q):

```
pacf(tcm10y)
```



Se observan los dos primeros lags fuera de la banda de confianza.

Estimación de modelos ARMA(p, q):

Lo anterior sugiere ajustar un un modelo AR(2):

```
modelo1 <- arima(tcm10y, order=c(2, 0, 0)) # order=c()
modelo1
```

```
## Warning in arima(tcm10y, order = c(2, 0, 0)): possible convergence p
## optim gave code = 1
```

```
##
## Call:
## arima(x = tcm10y, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##       1.3250  -0.3320     6.2448
## s.e.  0.0399   0.0399     1.4044
##
## sigma^2 estimated as 0.07231:  log likelihood = -61.28,  aic = 130.5
```

Estimación de modelos ARMA(p, q):

```
names(modelo1)
```

```
##  [1] "coef"      "sigma2"    "var.coef"  "mask"      "loglik"    "aic"
##  [7] "arma"      "residuals" "call"      "series"    "code"      "n.cond"
## [13] "nobs"      "model"
```

Los valores ajustados se pueden obtener como:

```
mod1_ajust <- tcm10y - modelo1$residuals
```

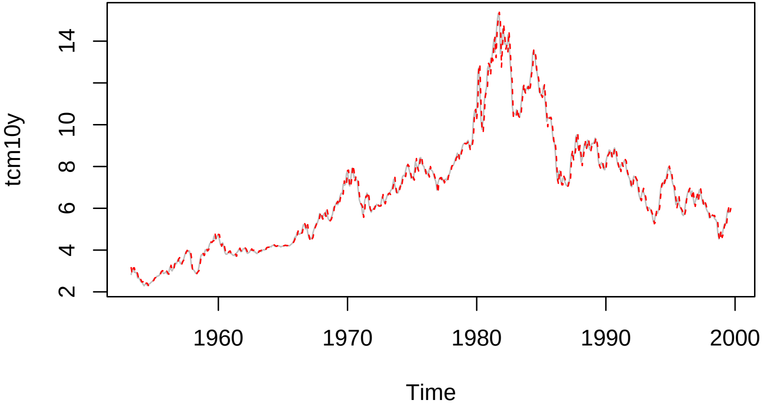
Los coeficientes estimados son:

```
modelo1$coef
```

```
##        ar1        ar2  intercept
##  1.3249527 -0.3320146  6.2448425
```

Grafiquemos la serie **tcm10y** versus los valores ajustados por el modelo:

```
plot(tcm10y, col="gray")
lines(mod1_ajust, col="red", lty=2)
```



En este ejemplo, vemos lo siguiente:

- Las ACF y PACF funcionan para identificar los ordenes p y/o q de modelos estacionarios y vemos que esta serie tiene tendencia y nos lleva a parámetros estimados $\widehat{\phi}_0 = 6.245$, $\widehat{\phi}_1 = 1.325$, $\widehat{\phi}_2 = -0.332$.
- Si utilizamos estos parámetros para analizar la estacionariedad del proceso estimado: $\widehat{X}_t = \widehat{\phi}_0 + \widehat{\phi}_1 X_{t-1} + \widehat{\phi}_2 X_{t-2}$, tenemos el polinomio $\phi(B) = 1 - \widehat{\phi}_1 B - \widehat{\phi}_2 B^2$ cuyas raices son:

```
polyroot(c(1,-1.3250,0.3320))
```

```
## [1] 1.010647+0i 2.980317+0i
```

Las cuales se encuentran fuera del círculo unitario:

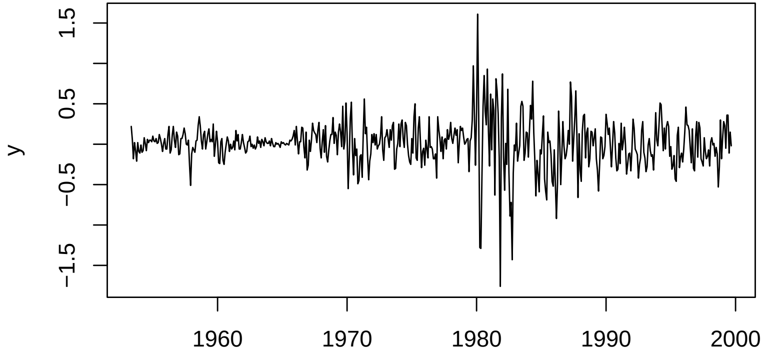
```
abs(polyroot(c(1,-1.3250,0.3320)))
```

```
## [1] 1.010647 2.980317
```

Estimación de modelos ARMA(p, q):

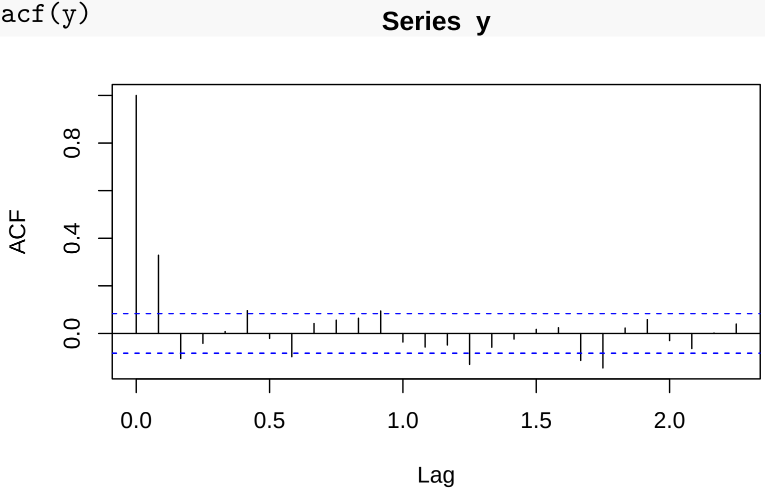
Una opción recomendada en un serie con tendencia es considerar la diferencia de la serie entre el mes t y el mes $t-1$, es decir, si X_t denota el precio mensual de los bonos a 10 años, la serie diferencia es $Y_t = X_t - X_{t-1}$:

```r
y <- diff(tcm10y)
plot(y)
```



Encontremos las ACF y PACF muestrales:

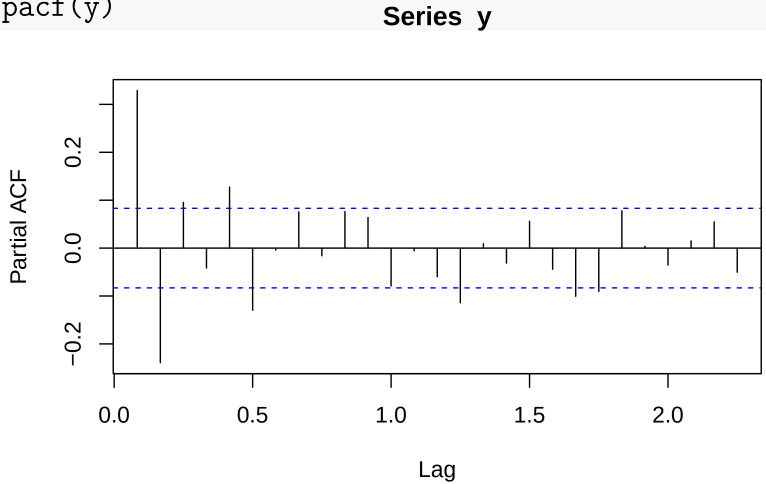
```
acf(y)
```



Se obseva primeros dos lags fuera de la banda de confianza

Encontremos las ACF y PACF muestrales:

```
pacf(y)
```



Decaimiento senoidal.

Estimación de modelos ARMA(p, q):

Lo anterior sugiere ajustar un modelo MA(2):

```
modelo2 <- arima(y, order=c(0, 0, 2))
modelo2
```

```
##
## Call:
## arima(x = y, order = c(0, 0, 2))
##
## Coefficients:
##          ma1      ma2  intercept
##       0.4488  -0.1180     0.0055
## s.e.  0.0424   0.0426     0.0146
##
## sigma^2 estimated as 0.06698:  log likelihood = -37.63,  aic = 83.27
```

Estimación de modelos ARMA(p, q):

```
names(modelo2)
```

```
##  [1] "coef"      "sigma2"    "var.coef"  "mask"      "loglik"    "aic"
##  [7] "arma"      "residuals" "call"      "series"    "code"      "n.cond"
## [13] "nobs"      "model"
```

Los valores ajustados se pueden obtener como:

```
mod2_ajust <- y - modelo2$residuals
```

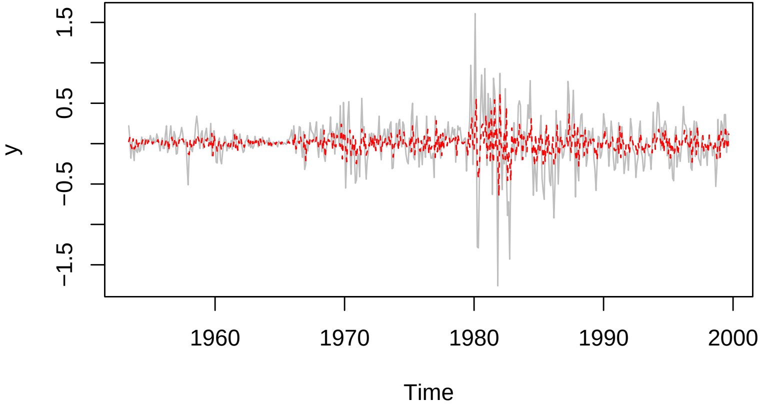
Los coeficientes estimados son:

```
modelo2$coef
```

```
##          ma1          ma2    intercept
##  0.448766481 -0.118042957  0.005514984
```

Grafiquemos la serie Y_t versus los valores ajustados por el modelo:

```
plot(y, col="gray")
lines(mod2_ajust, col="red", lty=2)
```



Estimación de modelos ARMA(p, q):

En estos ajustes surgen varias preguntas:

1. ¿Cómo se estiman los errores estádar (s.e.) de los parámetros estimados? ¿Para qué usarlos?
2. ¿Cómo se calcula el **aic** y cómo usarlo en este contexto?
3. Entre varios modelos, ¿cómo saber cuál es el mejor?
4. ¿Cómo podemos realizar un diagnóstico para saber si el modelo ajusta de manera correcta los datos con respecto a los supuestos realizados?

Errores estándar (s.e.) de los estimadores

Para responder a la primera pregunta, recurrimos al siguiente resultado (visto en inferencia estadística y que se encuentra en TEXTO 2 - EN R - Time-series-analysis-and-its-applications-with-examples-in-r **página 133**) sobre los estimadores de máxima verosimilitud:

Para un processo ARMA(p, q) que es estacionario e invertible se cumple que los estimadores de máxima verosimilitud, además de los estimadores de mínimos cuadrados no condicionales y también condicionales, denotados por $\widehat{\beta}$, tienen distribución asintóticamente normal dada por:

$$\sqrt{n}\left(\widehat{\beta} - \beta\right) \quad \xrightarrow{d} \quad N(\mathbf{0}, \sigma_w^2 \mathbf{\Gamma}_{p,\ q}^{-1})$$

donde $\sigma_w^2 \mathbf{\Gamma}_{p,\ q}^{-1}$ representa la matriz de varianzas-covarianzad de $\widehat{\beta}$ y se obtiene como la inversa de la matriz de información.

Errores estándar (s.e.) de los estimadores

En particular la matriz $(p+q)\times(p+q)$ tiene la forma:

$$\mathbf{\Gamma}_{p,\ q} = \begin{pmatrix} \mathbf{\Gamma}_{\phi\phi} & \mathbf{\Gamma}_{\phi\theta} \\ \mathbf{\Gamma}_{\theta\phi} & \mathbf{\Gamma}_{\theta\theta} \end{pmatrix}$$

Con $\mathbf{\Gamma}_{\phi\phi}$ representando a la matriz de covarianzas de los parámetros AR(p), $\mathbf{\Gamma}_{\theta\theta}$ representando a la matriz de covarianzas de los parámetros MA(q) y $\mathbf{\Gamma}_{\phi\theta}$, $\mathbf{\Gamma}_{\theta\phi}$ representando a la matriz de covarianzas-cruzadas entre los parámetros AR(p) y MA(q).

En particular la matriz $(p+q) \times (p+q)$ tiene la forma:

$$\boldsymbol{\Gamma}_{p,\ q} = \begin{pmatrix} \boldsymbol{\Gamma}_{\phi\phi} & \boldsymbol{\Gamma}_{\phi\theta} \\ \boldsymbol{\Gamma}_{\theta\phi} & \boldsymbol{\Gamma}_{\theta\theta} \end{pmatrix}$$

Con $\boldsymbol{\Gamma}_{\phi\phi}$ representando a la matriz de covarianzas de los parámetros AR(p), $\boldsymbol{\Gamma}_{\theta\theta}$ representando a la matriz de covarianzas de los parámetros MA(q) y $\boldsymbol{\Gamma}_{\phi\theta}$, $\boldsymbol{\Gamma}_{\theta\phi}$ representando a la matriz de covarianzas-cruzadas entre los parámetros AR(p) y MA(q).

La raíz cuadrada de los elementos de la diagonal de la matriz $\sigma_w^2 \boldsymbol{\Gamma}_{p,\ q}^{-1}$ contiene a los errores estándar (s.e.) teóricos de los parámetros de la parte AR(p) y MA(q). Reemplazando σ_w^2 por su estimador $\widehat{\sigma}_w^2$ (además de los otros estimadores en $\boldsymbol{\Gamma}_{p,\ q}^{-1}$) obtenemos los valores en la salida de la función **arima** del R.

Errores estándar (s.e.) de los estimadores

Un uso que se le puede dar a estos valores de errores estándar es plantear las hipotesis individuales:

- En la parte AR(p):

$$H_0 : \phi_i = 0 \quad \text{versus} \quad \phi_i \neq 0, \quad i = 0, 1, 2, \ldots, p$$

- En la parte MA(q):

$$H_0 : \theta_j = 0 \quad \text{versus} \quad \theta_j \neq 0, \quad j = 1, 2, \ldots, q$$

Cuyos estadísticos de prueba son:

$$Z_{\phi,i} = \frac{\widehat{\phi}_i}{\widehat{s.e.}(\phi_i)} \quad \text{y} \quad Z_{\theta,j} = \frac{\widehat{\theta}_j}{\widehat{s.e.}(\theta_j)}$$

Y comparando con la el cuantil $z_{\alpha/2}$ de la normal estándar, rechazamos H_0 si:

$$|Z_{\phi,i}| > z_{\alpha/2} \quad \text{y} \quad |Z_{\theta,j}| > z_{\alpha/2}$$

Errores estándar (s.e.) de los estimadores

Otro uso que se le puede dar a los errores estándar es para construir intervalos de confianza dados por:

- Parámetros de la parte AR(p):

$$\widehat{\phi}_i \pm z_{\alpha/2} \times \widehat{s.e.}(\phi_i), \quad i = 0, 1, 2, \ldots, p$$

- Parámetros de la parte MA(q):

$$\widehat{\theta}_j \pm z_{\alpha/2} \times \widehat{s.e.}(\theta_j), \quad j = 1, 2, \ldots, q$$

NOTA: Recuerde que cuando no hay información sobre el nivel de significancia α, se toma por defecto igual a 0.05 o 5 %.

Ejemplo aplicado a datos reales:

Considere la base de datos de **Incidentes Viales con Motos** registrados en Medellín entre el 2015 y el 2021 (para ver el sitio web donde se puede descargar la base de datos dé click AQUÍ). La BD también se puede descargar de la pestaña de datos del Moodle:

```
require(tidyverse)
require(magrittr)
require(readr)
require(janitor)
direccion <- "../../DATOS/incidentes_viales_motos.csv"
bd_incidentes <- read_delim(direccion,
                            delim = ";")
bd_incidentes %<>% clean_names()
bd_incidentes %>% dim()
```

```
## [1] 223439      9
```

Ejemplo aplicado a datos reales:

Esta base de datos tiene un problema con las fechas y es el siguiente:

```
bd_incidentes %>% head(n=3)
```

```
## # A tibble: 3 x 9
##   nro_radicado ano_incidente fecha_incidente hora_incidente clase_incidente
##          <dbl>         <dbl> <chr>           <chr>          <chr>
## 1      1473523          2015 27/01/15        22:00:00       Choque
## 2      1473525          2015 27/01/15        15:40:00       Choque
## 3      1473526          2015 27/01/15        18:00:00       Choque
## # ... with 4 more variables: gravedad_incidente <chr>, direccion <chr>,
## #   zona <chr>, diseno_via <chr>
```

```
bd_incidentes %>% tail(n=3)
```

```
## # A tibble: 3 x 9
##   nro_radicado ano_incidente fecha_incidente hora_incidente clase_incidente
##          <dbl>         <dbl> <chr>           <chr>          <chr>
## 1      1759665          2021 25/08/2021      20:15:00       Choque
## 2      1759743          2021 25/08/2021      20:25:00       Choque
## 3      1759627          2021 25/08/2021      21:00:00       Otro
## # ... with 4 more variables: gravedad_incidente <chr>, direccion <chr>,
## #   zona <chr>, diseno_via <chr>
```

Ejemplo aplicado a datos reales:

Este problema con las fechas lleva a que si queremos darle el formato adecuado tengamos un problema, ya que para fechas como la siguiente:

```
as.Date("27/01/15", format="%d/%m/%y") # y minúscula
```

```
## [1] "2015-01-27"
```

El formato no funciona si es:

```
as.Date("27/01/2015", format="%d/%m/%y") # y minúscula
```

```
## [1] "2020-01-27"
```

Ejemplo aplicado a datos reales:

Este problema con las fechas lleva a que si queremos darle el formato adecuado tengamos un problema, ya que para fechas como la siguiente:

```
as.Date("27/01/15", format="%d/%m/%y") # y minúscula
```

```
## [1] "2015-01-27"
```

El formato no funciona si es:

```
as.Date("27/01/2015", format="%d/%m/%y") # y minúscula
```

```
## [1] "2020-01-27"
```

Lo correcto es:

```
as.Date("27/01/2015", format="%d/%m/%Y") # Y mayúscula
```

```
## [1] "2015-01-27"
```

Ejemplo aplicado a datos reales:

Una solución a este problema particular es observando que el número de caracteres es diferente en ambos formatos, pero constante:

```
nchar("27/01/15")
```

```
## [1] 8
```

```
nchar("27/01/2015")
```

```
## [1] 10
```

Ejemplo aplicado a datos reales:

Una solución a este problema particular es observando que el número de caracteres es diferente en ambos formatos, pero constante:

```
nchar("27/01/15")
```

```
## [1] 8
```

```
nchar("27/01/2015")
```

```
## [1] 10
```

Así, podemos darle formato a la fecha del incidente considerando un condicional como:

```
aux1 <- as.Date(rep(NA, nrow(bd_incidentes)))
aux2 <- bd_incidentes$fecha_incidente
ind1<-which(nchar(aux2)==8)
ind2<-which(nchar(aux2)==10)
aux1[ind1] <- as.Date(aux2[ind1], format = "%d/%m/%y")
aux1[ind2] <- as.Date(aux2[ind2], format = "%d/%m/%Y")
bd_incidentes$fecha_incidente<-aux1
```

Ejemplo aplicado a datos reales:

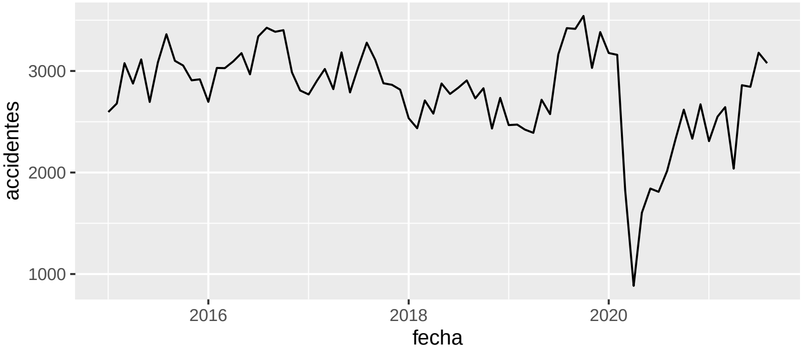
Obtenemos el conteo de accidentes por día:

```
require(lubridate)
bd_incidentes$mes<-month(bd_incidentes$fecha_incidente)
bd_incidentes$anio<-year(bd_incidentes$fecha_incidente)
resum1 <- bd_incidentes %>% group_by(anio, mes) %>%
                            summarise(accidentes=n())
resum1$fecha <- as.Date(
                paste(resum1$anio,resum1$mes,1,sep="-"),
                       format = "%Y-%m-%d")
```

El gráfico de la serie de accidentes mensuales está dado por:

Ejemplo aplicado a datos reales:

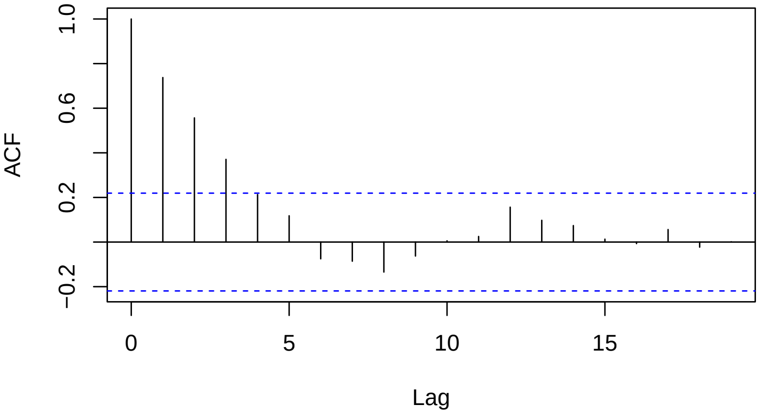
```
resum1 %>% ggplot(aes(x=fecha, y=accidentes))+
  geom_line()
```



Ejemplo aplicado a datos reales:

Encontremos las ACF y PACF muestrales:

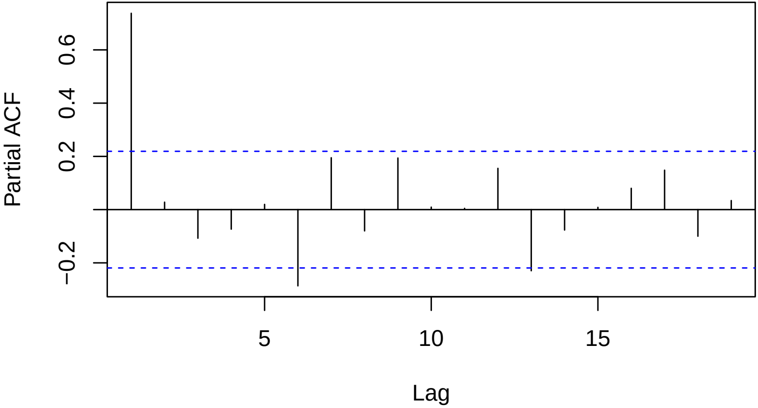
```
acf(resum1$accidentes)
```



Se observa un decaimiento.

Ejemplo aplicado a datos reales:

```
pacf(resum1$accidentes)
```



Los lags 1 y 6 sobresalen de la banda de confianza.

Lo anterior sugiere ajustar un modelo AR(6):

```
modelo3 <- arima(resum1$accidentes, order=c(6, 0, 0))
modelo3
```

```
##
## Call:
## arima(x = resum1$accidentes, order = c(6, 0, 0))
##
## Coefficients:
##          ar1     ar2      ar3      ar4     ar5      ar6  intercept
##       0.7024  0.1013  -0.0705  -0.0515  0.2258  -0.2969  2794.4827
## s.e.  0.1055  0.1307   0.1328   0.1328  0.1409   0.1117    83.4591
##
## sigma^2 estimated as 83525:  log likelihood = -567.52,  aic = 1151.04
```

Ejemplo aplicado a datos reales:

La matriz de covarianzas de los estimadores está dada por

```r
modelo3$var.coef
```

```
##                ar1     ar2     ar3     ar4     ar5     ar6 intercept
## ar1         0.0111 -0.0080 -0.0015  0.0006  0.0011  0.0001   -0.0519
## ar2        -0.0080  0.0171 -0.0073 -0.0013 -0.0007  0.0012   -0.0232
## ar3        -0.0015 -0.0073  0.0176 -0.0075 -0.0012  0.0000    0.0870
## ar4         0.0006 -0.0013 -0.0075  0.0176 -0.0084 -0.0006    0.1867
## ar5         0.0011 -0.0007 -0.0012 -0.0084  0.0198 -0.0098   -0.0470
## ar6         0.0001  0.0012  0.0000 -0.0006 -0.0098  0.0125   -0.0559
## intercept -0.0519 -0.0232  0.0870  0.1867 -0.0470 -0.0559 6965.4181
```

Los erorres estandar (s.e.) son:

```r
se <- modelo3$var.coef %>% diag() %>% sqrt() %>% round(4)
```

Ejemplo aplicado a datos reales:

Los intervalos de confianza están dados por:

```
modelo3$coef-1.96*se
```

```
##        ar1       ar2       ar3       ar4       ar5       ar6 intercept
##      0.496    -0.155    -0.331    -0.312    -0.050    -0.516  2630.903
```

```
modelo3$coef+1.96*se
```

```
##        ar1       ar2       ar3       ar4       ar5       ar6 intercept
##      0.909     0.357     0.190     0.209     0.502    -0.078  2958.063
```

donde $z_{\alpha/2} = z_{0.025} = 1.96$.

¿Qué se puede concluir de estos resultados?

Ejemplo aplicado a datos reales:

Los intervalos de confianza están dados por:

```
modelo3$coef-1.96*se
```

```
##       ar1       ar2       ar3       ar4       ar5       ar6 intercept
##     0.496    -0.155    -0.331    -0.312    -0.050    -0.516  2630.903
```

```
modelo3$coef+1.96*se
```

```
##       ar1       ar2       ar3       ar4       ar5       ar6 intercept
##     0.909     0.357     0.190     0.209     0.502    -0.078  2958.063
```

donde $z_{\alpha/2} = z_{0.025} = 1.96$.

¿Qué se puede concluir de estos resultados? Solo los parámetros ϕ_1, ϕ_6 y el intercepto (ϕ_0) tienen intervalos que no contienen al cero, lo cual puede interpretarse como que son estadísticamente distintos de cero con una confianza del 95 %.