Series de tiempo univariadas - Presentación 22

Mauricio Alejandro Mazo Lopera

Universidad Nacional de Colombia
Facultad de Ciencias
Escuela de Estadística
Medellín

En muchas situaciones las series de tiempo se ven afectadas por ciertos eventos externos tales como días festivos, huelgas, promociones, cambios políticos, etc. A dichos eventos se les suele llamar **intervenciones**.

Cuando hablamos de **análisis de intervenciones**, nos referimos a la técnica que evalúa el efecto de eventos externos.

En muchos casos se conocen los eventos que producen las intervenciones, pero en otros casos no se conoce y por tanto, lleva a valores dentro de la serie que se conocen como datos **outliers**.

Modelos de intervención

Dado que sabemos que ocurre una intervención en el tiempo T, la pregunta que surge es ¿dicho evento generó un cambio en la serie de tiempo (como por ejemplo un incremento en la media)? y si es así, ¿qué tan grande fue dicho efecto?

Modelos de intervención

Dado que sabemos que ocurre una intervención en el tiempo T, la pregunta que surge es ¿dicho evento generó un cambio en la serie de tiempo (como por ejemplo un incremento en la media)? y si es así, ¿qué tan grande fue dicho efecto?

Existen dos tipos de variables que se asocian con el análisis de intervenciones:

- La **función paso:**

$$S_t^{(T)} = \begin{cases} 0, & t < T \\ 1, & t \geq T \end{cases}$$

- La **función pulso:**

$$P_t^{(T)} = \begin{cases} 0, & t = T \\ 1, & t \neq T \end{cases}$$

Note que existe una relación entre las funciones paso y pulso dado por:

$$P_t^{(T)} = S_t^{(T)} - S_{t-1}^{(T)} = (1 - B)S_t^{(T)}$$

Esto implica que una intervención puede ser representada por la función pulso o por la función paso. El uso de una forma u otra depende de la conveniencia a la hora de la interpretación.

Entre las varias formas de respuesta a intervenciones de tipo pulso o paso se destacan las siguientes:

Modelos de intervención

1. Cuando un impacto desconociso de una intervención se "siente" b periodos después de la intervención. Matemáticamente se puede escribir como:

$$\omega B^b S_t^{(T)} \quad \text{ó} \quad \omega B^b P_t^{(T)}$$

2. El impacto de una intervención se "siente" b después de la intervención, pero la respuesta es gradual. Matemáticamente se puede escribir como:

$$\frac{\omega B^b}{(1-\delta B)} S_t^{(T)} \quad \text{ó} \quad \frac{\omega B^b}{(1-\delta B)} P_t^{(T)}$$

donde $0 \leq \delta \leq 1$. Cuando $\delta = 0$, estas expresiones son equivalentes a las del item (1). Cuando $\delta = 1$, el impacto incrementa linealmente sin una cota. En la mayoría de los casos $0 < \delta < 1$ y la respuesta es gradual.

De manera mucho más general, una respuesta a intervenciones puede plantearse como una función racional:

$$\frac{\omega(B)B^b}{\delta(B)}$$

donde $\omega(B) = \omega_0 - \omega_1 B - \cdots - \omega_s B^s$ y $\delta(B) = 1 - \delta_1 B - \cdots - \delta_r B^r$ son polinímios, b es el tiempo de demora del efecto de la intervención, y los pesos ω_j representan los efectos iniciales esperados de la intervención.

Modelos de intervención

De manera mucho más general, una respuesta a intervenciones puede plantearse como una función racional:

$$\frac{\omega(B)B^b}{\delta(B)}$$

donde $\omega(B) = \omega_0 - \omega_1 B - \cdots - \omega_s B^s$ y $\delta(B) = 1 - \delta_1 B - \cdots - \delta_r B^r$ son polinímios, b es el tiempo de demora del efecto de la intervención, y los pesos ω_j representan los efectos iniciales esperados de la intervención.

Por otra parte, el polinómio $\delta(B)$ mide el comportamiento del efecto permanente de la intervención. Una raíz unitaria de este polinómio indica un efecto que va creciendo de manera lineal y raíces por fuera del círculo unitario representa que el evento tiene una respuesta gradual.

Modelos de intervención

Para múltiples intervenciones, tenemos el siguiente modelo general:

$$X_t = \theta_0 + \sum_{j=1}^{k} \frac{\omega_j(B)B^{b_j}}{\delta_j(B)} I_{jt} + \frac{\theta(B)}{\psi(B)} w_t$$

donde I_{jt}, $j = 1, 2, \ldots, k$ son variables de intervención que pueden ser funciones paso o pulso. Además,

$$\psi(B) = (1 - B)^d \phi(B)$$

El objetivo de este modelo es medir los efectos de las intervenciones.

Ejemplo aplicado a datos reales: Accidentes

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:

```r
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
```

Le damos 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: Accidentes

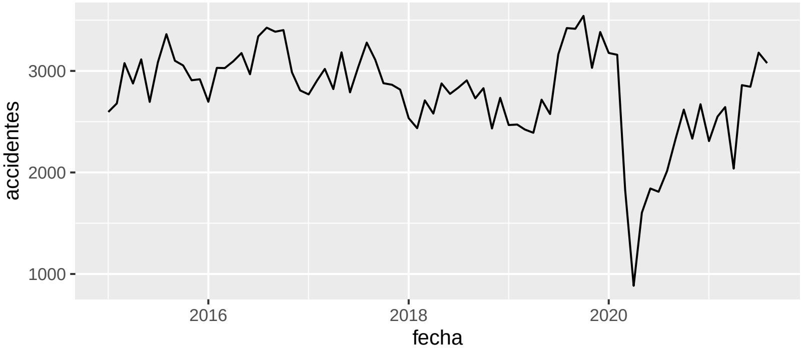
Obtenemos el conteo de accidentes por día:

```
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: Accidentes

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



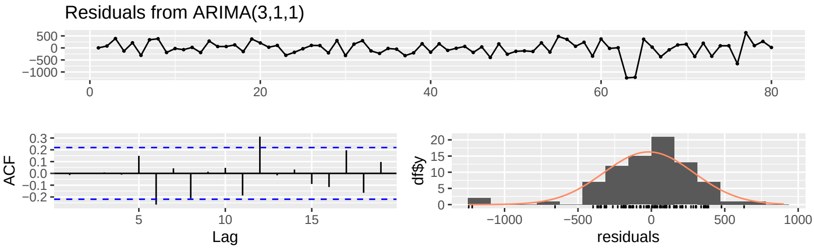
En abril de 2020 se disminuyeron drásticamente los accidentes y es claro que fue por las mediadas relacionadas con la pandemia.

Ejemplo aplicado a datos reales: Accidentes

Inicialmente ajustamos un modelo $ARIMA(p, d, q)$ sin preocuparnos por la caída en los valores en abril de 2020:

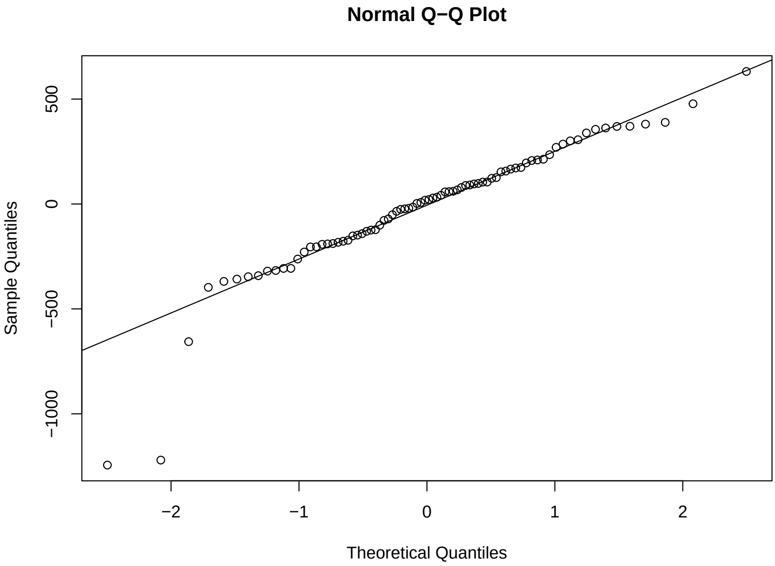
```
require(forecast)
require(lmtest)
modelo0 <- auto.arima(resum1$accidentes, stepwise = FALSE,
          approximation = FALSE)
modelo0 %>% coeftest()
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.720577   0.116761   6.1714 6.769e-10 ***
## ar2   0.121460   0.137898   0.8808    0.3784
## ar3 -0.117051   0.117485  -0.9963    0.3191
## ma1 -0.981396   0.071823 -13.6641 < 2.2e-16 ***
## ---
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,1)
## Q* = 44.329, df = 21, p-value = 0.002119
##
## Model df: 4.   Total lags used: 25
```

Ejemplo aplicado a datos reales: Accidentes



Ejemplo aplicado a datos reales: Accidentes

Dividimos la base de datos pre-intervención para ver el orden del modelo a ser ajustado:

```
resum1_pre <- resum1[1:63,]
```

Usamos la función **auto.arima** del paquete **forecast**:

```
require(forecast)
auto.arima(resum1_pre$accidentes, stepwise = FALSE,
           approximation = FALSE)
```

```
## Series: resum1_pre$accidentes
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##          ma1     ma2       mean
##       0.5211  0.4746  2895.1645
## s.e.  0.1431  0.1096    64.9484
```

Ejemplo aplicado a datos reales: Accidentes

Como vimos antes, el modelo a ajustar es un $ARIMA(0, 0, 2)$. Usamos la función **arimax** del paquete **TSA** para ver el efecto de la intervención dada por el modelo:

$$X_t = \theta_0 + \frac{\omega_1}{1 - \delta_1 B} P_{1t}^{(64)} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2}$$

```
require(TSA)
modelo1 <- arimax(resum1$accidentes, order=c(0, 0, 2),
  xtransf=data.frame(
   abril2020a=1 * (seq_along(resum1$accidentes) == 64)),
              transfer=list(c(1, 0)))
```

Vemos el resumen del modelo:

```
require(lmtest)
modelo1 %>% coeftest()
```

```
##
## z test of coefficients:
##
##                    Estimate  Std. Error z value  Pr(>|z|)
## ma1              5.2019e-01  1.2036e-01  4.3218 1.548e-05 ***
## ma2              5.8228e-01  7.9554e-02  7.3193 2.492e-13 ***
## intercept        2.8005e+03  6.5673e+01 42.6427 < 2.2e-16 ***
## abril2020a-AR1   3.1766e-02  1.1639e-01  0.2729    0.7849
## abril2020a-MA0 -1.1510e+03  2.4568e+02 -4.6851 2.798e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

Como vimos antes, el δ_1 estimado $(3.1766e-02)$ no es significativo con un nivel de significancia del 5 %. En cambio, el ω_1 estimado (-1151) sí es significativo.

Ejemplo aplicado a datos reales: Accidentes

Como vimos antes, el δ_1 estimado ($3.1766e-02$) no es significativo con un nivel de significancia del 5 %. En cambio, el ω_1 estimado (-1151) sí es significativo. Ajustamos nuevamente el modelo sin tener en cuenta el δ_1:

```
require(TSA)
modelo2 <- arimax(resum1$accidentes, order=c(0, 0, 2),
  xtransf=data.frame(
   abril2020a=1 * (seq_along(resum1$accidentes) == 64)),
              transfer=list(c(0, 0)))
```

Ejemplo aplicado a datos reales: Accidentes

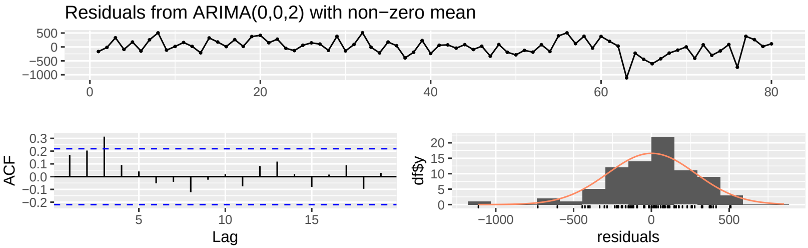
Vemos el resumen del modelo:

```
require(lmtest)
modelo2 %>% coeftest()
```

```
##
## z test of coefficients:
##
##                  Estimate  Std. Error z value  Pr(>|z|)
## ma1            5.1808e-01  1.1837e-01  4.3768 1.204e-05 ***
## ma2            5.7696e-01  8.0815e-02  7.1394 9.377e-13 ***
## intercept      2.8061e+03  6.5429e+01 42.8882 < 2.2e-16 ***
## abril2020a-MA0 -1.1397e+03  2.4515e+02 -4.6491 3.334e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

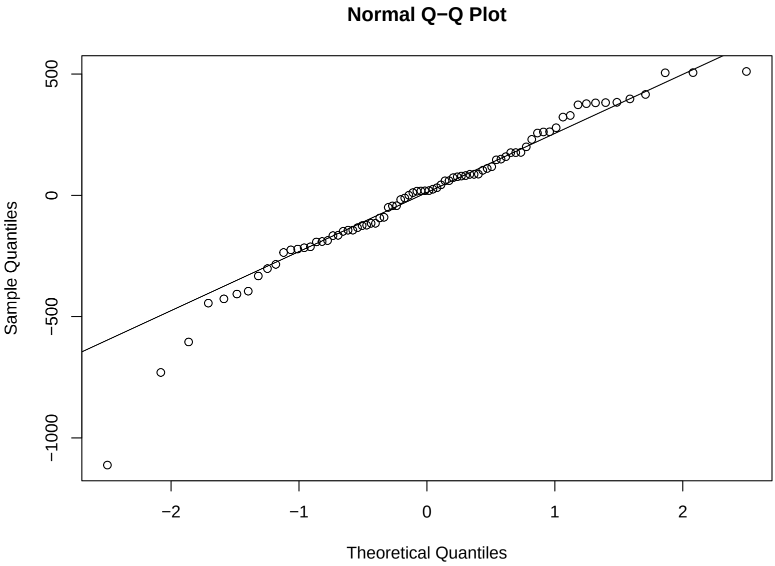
El valor -1139.7 se puede iterpretar como la caída que se presentó en el número de accidentes producto de las restricciones por pandemia.

Ejemplo aplicado a datos reales: Accidentes




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,2) with non-zero mean
## Q* = 25.3, df = 21, p-value = 0.2344
##
## Model df: 4.   Total lags used: 25
```

Ejemplo aplicado a datos reales: Accidentes



Ejemplo aplicado a datos reales: Accidentes

```
require(tseries)
modelo2$residuals %>% shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.95447, p-value = 0.006263
```

```
modelo2$residuals %>% jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  .
## X-squared = 22.954, df = 2, p-value = 1.036e-05
```

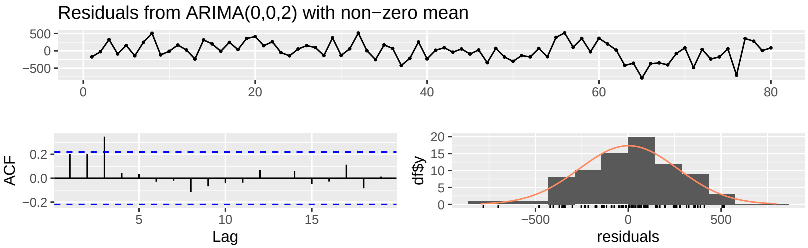
Ejemplo aplicado a datos reales: Accidentes

Podemos ajustar un tercer modelo considerando dos intervenciones: una en el mes de abril de 2020 y otra un mes antes, es decir, en marzo de 2020:

$$X_t = \theta_0 + \omega_1 P_{1t}^{(63)} + \omega_2 P_{2t}^{(64)} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2}$$

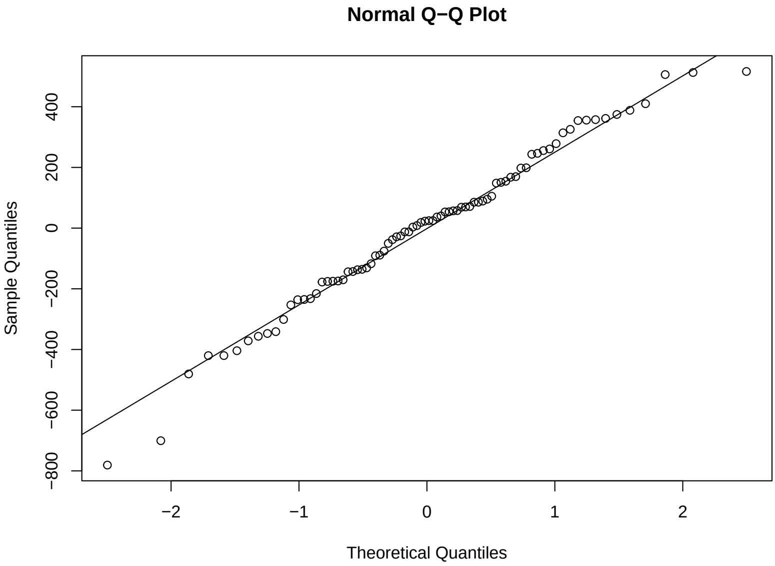
```
modelo3 <- arimax(resum1$accidentes, order=c(0, 0, 2),
  xtransf=data.frame(
   marzo2020a=1 * (seq_along(resum1$accidentes) == 63),
   abril2020a=1 * (seq_along(resum1$accidentes) == 64)),
              transfer=list(c(0, 0), c(0, 0)))
```

Ejemplo aplicado a datos reales: Accidentes




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,2) with non-zero mean
## Q* = 28.508, df = 20, p-value = 0.09791
##
## Model df: 5.   Total lags used: 25
```

Ejemplo aplicado a datos reales: Accidentes



Ejemplo aplicado a datos reales: Accidentes

```
modelo3$residuals %>% shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.98286, p-value = 0.3609
```

```
modelo3$residuals %>% jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  .
## X-squared = 1.8462, df = 2, p-value = 0.3973
```

Ejemplo aplicado a datos reales: Desempleo

Considere la BD relacionada con el Empleo y Desempleo en Colombia desde 2021 reportada por el DANE y que se encuentra en el link (AQUÍ) (también la pueden encontrar en la carpeta de DATOS del Moodle del curso como **anexo_empleo_ago_22.xlsx**).

puntos porcentuales respecto al mismo mes de 2021 (54,9%).

Documento	Fecha de publicación	Formato	Tamaño	Acción
Boletín técnico	30/09/2022	PDF	353 KB	Descargar
Comunicado de prensa	30/09/2022	PDF	466 KB	Descargar
Presentación (rueda de prensa)	30/09/2022	PDF	5,80 MB	Descargar
Presentación extendida (rueda de prensa)	30/09/2022	PDF	7,53 MB	Descargar
Presentación empalme de series GEIH 2005-2018	30/09/2022	PDF	1,18 MB	Descargar
Anexos	30/09/2022	XLSX	2.59 MB	Descargar
Anexo desestacionalizado	30/09/2022	XLSX	403 KB	Descargar
Nota técnica de empalme serie	30/09/2022	PDF	146 KB	Descargar

Ejemplo aplicado a datos reales: Desempleo

```r
require(tidyverse)
require(readxl)
direccion <- "../../DATOS/anexo_empleo_ago_22.xlsx"
bd_original <- read_excel(direccion,
                          sheet = "Total nacional",
                          skip=12)
# Seleccionamos la fila 4, relacionada con la tasa
# de desempleo y las columnas 2 hasta 261:
td <- bd_original %>%
       slice(4) %>%
       select(2:ncol(bd_original)) %>%
       t() %>% as.numeric()
tasa_desemp <- data.frame(fecha=seq(as.Date("2001/1/1"),
                                     as.Date("2022/8/1"),
                                     "months"), td =td)
```

Ejemplo aplicado a datos reales: Desempleo

```
tasa_desemp %>% head(4)
```

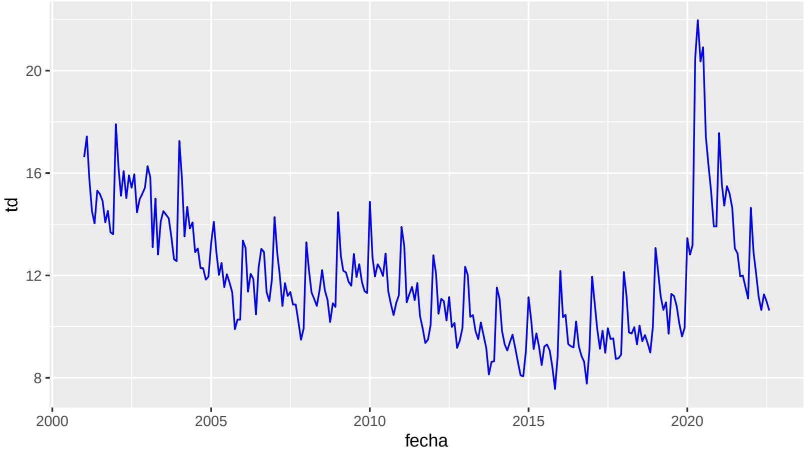
```
##        fecha      td
## 1 2001-01-01 16.6223
## 2 2001-02-01 17.4342
## 3 2001-03-01 15.8119
## 4 2001-04-01 14.5151
```

```
tasa_desemp %>% tail(4)
```

```
##          fecha       td
## 257 2022-05-01 10.64826
## 258 2022-06-01 11.26120
## 259 2022-07-01 10.98892
## 260 2022-08-01 10.63128
```

Ejemplo aplicado a datos reales: Desempleo

```
tasa_desemp %>% ggplot(aes(x=fecha, y=td))+
                  geom_line(col="blue")
```



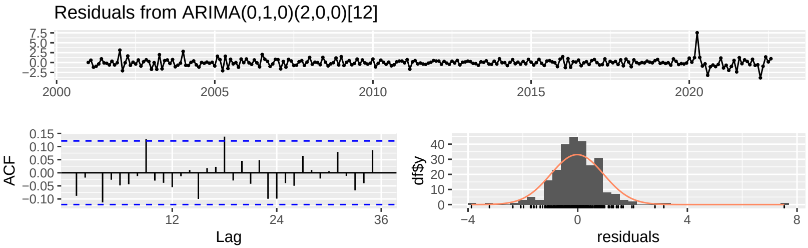
Ejemplo aplicado a datos reales: Desempleo

Inicialmente ajustamos un modelo $SARIMA(p, d, q) \times (P, D, Q)_s$ sin preocuparnos por la caída en los valores en abril de 2020:

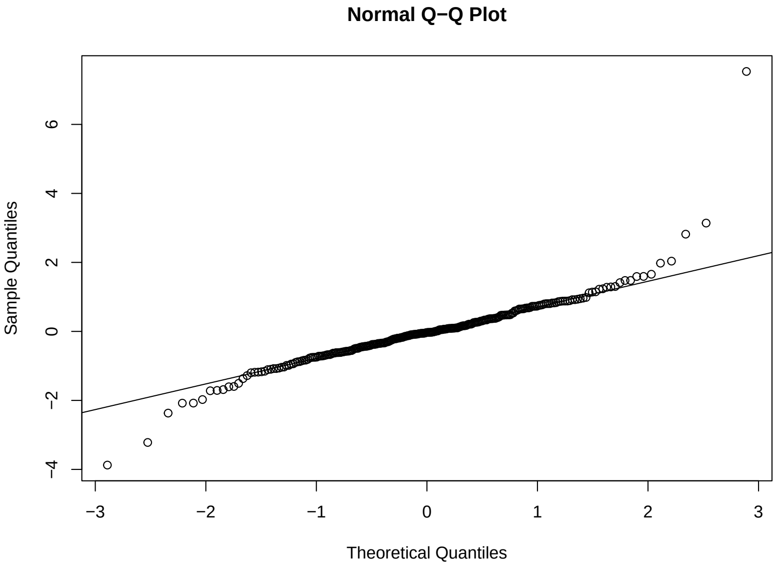
```
ts_desemp <- ts(tasa_desemp$td, start=c(2001,1),
                frequency = 12)
modelo0_des <- auto.arima(ts_desemp, stepwise = FALSE,
           approximation = FALSE)
modelo0_des %>% coeftest()
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## sar1 0.400275   0.058449  6.8483 7.474e-12 ***
## sar2 0.357977   0.060347  5.9320 2.993e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

Ejemplo aplicado a datos reales: Desempleo




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0)(2,0,0)[12]
## Q* = 29.382, df = 23, p-value = 0.168
##
## Model df: 2.   Total lags used: 25
```


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4
6
−3
−2
−1
0
1
2
3

Ejemplo aplicado a datos reales: Desempleo

```
modelo0_des$residuals %>% shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.87217, p-value = 6.4e-14
```

```
modelo0_des$residuals %>% jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  .
## X-squared = 2489.4, df = 2, p-value < 2.2e-16
```

Ejemplo aplicado a datos reales: Desempleo

Dividimos la base de datos pre-intervención para ver el orden del modelo a ser ajustado:

```
ts_desemp_pre <- window(ts_desemp, start = c(2001,1),
                                   end=c(2020,3))
```

Usamos la función **auto.arima** del paquete **forecast**:

```
require(forecast)
auto.arima(ts_desemp_pre, stepwise = FALSE,
           approximation = FALSE)
```

```
## Series: ts_desemp_pre
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##           ma1     sma1
##       -0.6151  -0.8241
## s.e.   0.0508   0.0570
##
```

Ejemplo aplicado a datos reales: Desempleo

Como vimos antes, el modelo a ajustar es un $SARIMA(0,1,1) \times (0,1,1)_{12}$. Usamos la función **arimax** del paquete **TSA** para ver el efecto de la intervención dada por el modelo:

$$(1-B^{12})(1-B)X_t = \frac{\omega_1}{1-\delta_1 B} S_{1t}^{(232)} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-12}$$

```
require(TSA)
modelo1_des <- arimax(ts_desemp, order=c(0, 1, 1),
                  seasonal = list(order = c(0, 1, 1)),
  xtransf=data.frame(
   abril2020a=c(rep(0, 231), rep(1, 29))),
               transfer=list(c(1, 0)))
```

Ejemplo aplicado a datos reales: Desempleo

Vemos el resumen del modelo:

```
require(lmtest)
modelo1_des %>% coeftest()
```

```
##
## z test of coefficients:
##
##                  Estimate Std. Error  z value  Pr(>|z|)
## ma1             -0.377206   0.055103  -6.8455 7.621e-12 ***
## sma1            -0.846801   0.054865 -15.4343 < 2.2e-16 ***
## abril2020a-AR1  0.077919   0.066277   1.1757    0.2397
## abril2020a-MA0  8.230505   0.663544  12.4039 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

Ejemplo aplicado a datos reales: Desempleo

Como vimos antes, el δ_1 estimado (0.077919) no es significativo con un nivel de significancia del 5 %. En cambio, el ω_1 estimado (8.230505) sí es significativo.

Ejemplo aplicado a datos reales: Desempleo

Como vimos antes, el δ_1 estimado (0.077919) no es significativo con un nivel de significancia del 5 %. En cambio, el ω_1 estimado (8.230505) sí es significativo. Ajustamos nuevamente el modelo sin tener en cuenta el δ_1:

```
require(TSA)
modelo1_des <- arimax(ts_desemp, order=c(0, 1, 1),
                  seasonal = list(order = c(0, 1, 1)),
  xtransf=data.frame(
   abril2020a=c(rep(0, 231), rep(1, 29))),
              transfer=list(c(0, 0)))
```

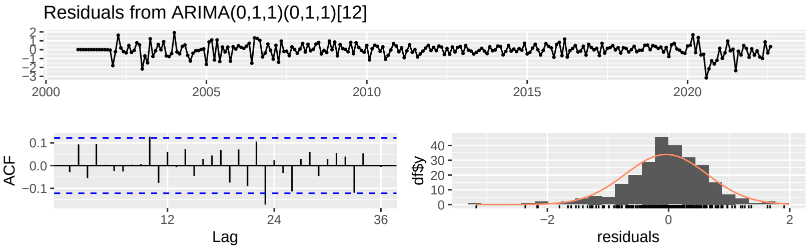
Vemos el resumen del modelo:

```
require(lmtest)
modelo1_des %>% coeftest()
```

```
##
## z test of coefficients:
##
##                  Estimate Std. Error z value  Pr(>|z|)
## ma1             -0.389639   0.054244  -7.183 6.818e-13 ***
## sma1            -0.847439   0.054375 -15.585 < 2.2e-16 ***
## abril2020a-MA0   8.492282   0.634071  13.393 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

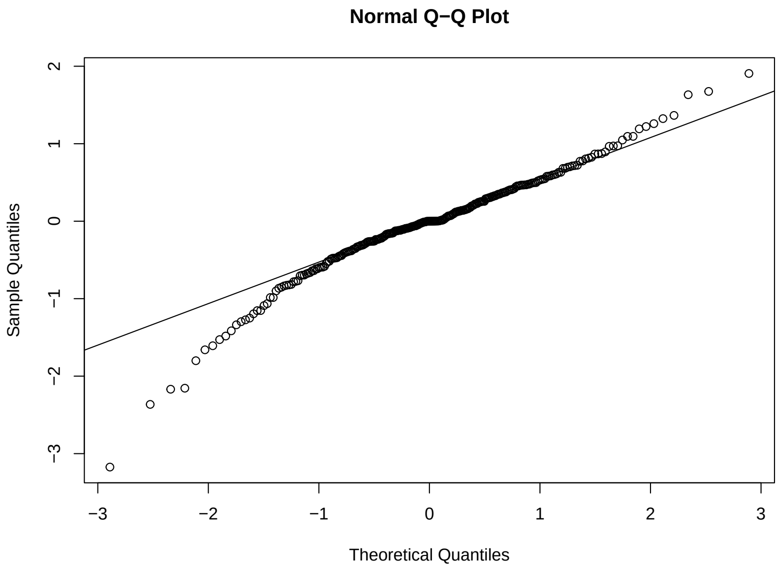
El valor 8.492282 se puede iterpretar el aumento que se presentó en el porcentaje de desempleo producto de la pandemia.

Ejemplo aplicado a datos reales: Desempleo




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(0,1,1)[12]
## Q* = 34.621, df = 22, p-value = 0.04242
##
## Model df: 3.   Total lags used: 25
```

Ejemplo aplicado a datos reales: Desempleo



Ejemplo aplicado a datos reales: Desempleo

```
require(tseries)
modelo1_des$residuals %>% shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.96037, p-value = 1.446e-06
```

```
modelo1_des$residuals %>% jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  .
## X-squared = 88.022, df = 2, p-value < 2.2e-16
```

Ensayamos el efecto de la intervención pulso dada por el modelo:

$$(1 - B^{12})(1 - B)X_t = \frac{\omega_1}{1 - \delta_1 B} P_{1t}^{(232)} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-12}$$

```
require(TSA)
modelo2_des <- arimax(ts_desemp, order=c(0, 1, 1),
                  seasonal = list(order = c(0, 1, 1)),
  xtransf=data.frame(
   abril2020a=1*(seq_along(ts_desemp) == 232)),
               transfer=list(c(1, 0)))
```

Ejemplo aplicado a datos reales: Desempleo

Vemos el resumen del modelo:

```
require(lmtest)
modelo2_des %>% coeftest()
```

```
##
## z test of coefficients:
##
##                   Estimate Std. Error z value  Pr(>|z|)
## ma1              -0.553726   0.064248 -8.6185 < 2.2e-16 ***
## sma1             -0.674933   0.085892 -7.8579 3.905e-15 ***
## abril2020a-AR1    0.907042   0.017255 52.5680 < 2.2e-16 ***
## abril2020a-MA0    9.597458   0.583212 16.4562 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

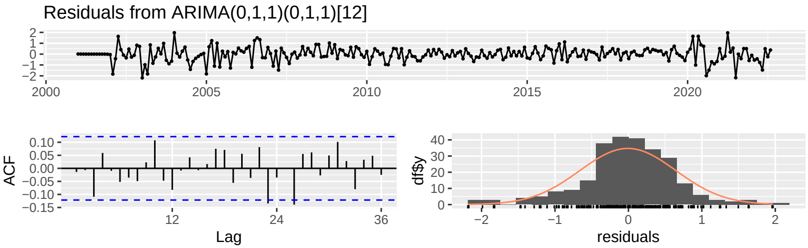
Como vimos antes, el δ_1 estimado (0.907042) es significativo con un nivel de significancia del 5 %. El ω_1 estimado (9.597458) es también significativo.

Ejemplo aplicado a datos reales: Desempleo

Como vimos antes, el δ_1 estimado (0.907042) es significativo con un nivel de significancia del 5 %. El ω_1 estimado (9.597458) es también significativo.

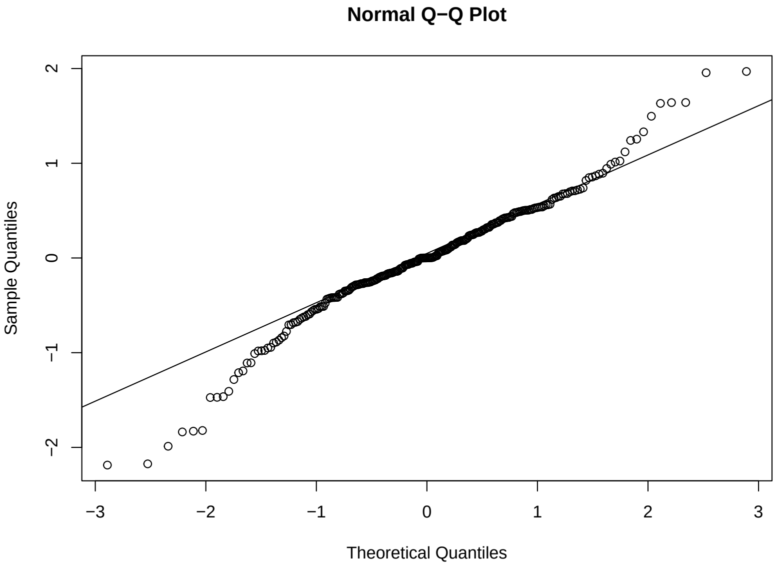
Que el δ_1 sea significativo se puede ver como que después de la intervención el efecto fue cayendo gradualmente.

El valor 9.597458 se puede iterpretar el aumento que se presentó en el porcentaje de desempleo producto de la pandemia.




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(0,1,1)[12]
## Q* = 24.807, df = 21, p-value = 0.2556
##
## Model df: 4.   Total lags used: 25
```

Ejemplo aplicado a datos reales: Desempleo



Ejemplo aplicado a datos reales: Desempleo

```
require(tseries)
modelo2_des$residuals %>% shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.97052, p-value = 3.329e-05
```

```
modelo2_des$residuals %>% jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  .
## X-squared = 29.017, df = 2, p-value = 5.001e-07
```

Ejemplo aplicado a datos reales: Desempleo

Ensayamos un nuevo modelo aplicando la transformación logaritmo natural, es decir, modelamos la serie $Y_t = \ln(X_t)$:

$$(1 - B^{12})(1 - B)Y_t = \frac{\omega_1}{1 - \delta_1 B} P_{1t}^{(232)} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-12}$$

```
require(TSA)
modelo3_des <- arimax(log(ts_desemp), order=c(0, 1, 1),
                  seasonal = list(order = c(0, 1, 1)),
  xtransf=data.frame(
   abril2020a=1*(seq_along(ts_desemp) == 232)),
              transfer=list(c(1, 0)))
```

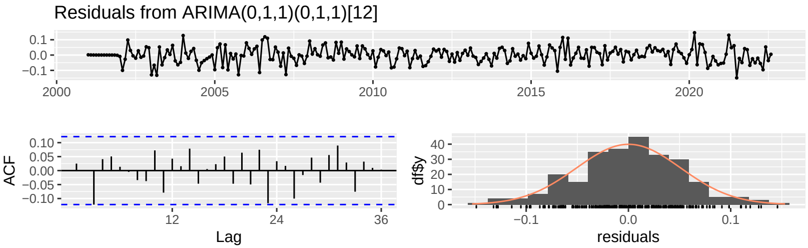
Vemos el resumen del modelo:

```
require(lmtest)
modelo3_des %>% coeftest()
```

```
##
## z test of coefficients:
##
##                  Estimate Std. Error z value  Pr(>|z|)
## ma1             -0.590655   0.056288 -10.493 < 2.2e-16 ***
## sma1            -0.801131   0.056488 -14.182 < 2.2e-16 ***
## abril2020a-AR1  0.924217   0.018131  50.973 < 2.2e-16 ***
## abril2020a-MA0  0.617083   0.044536  13.856 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

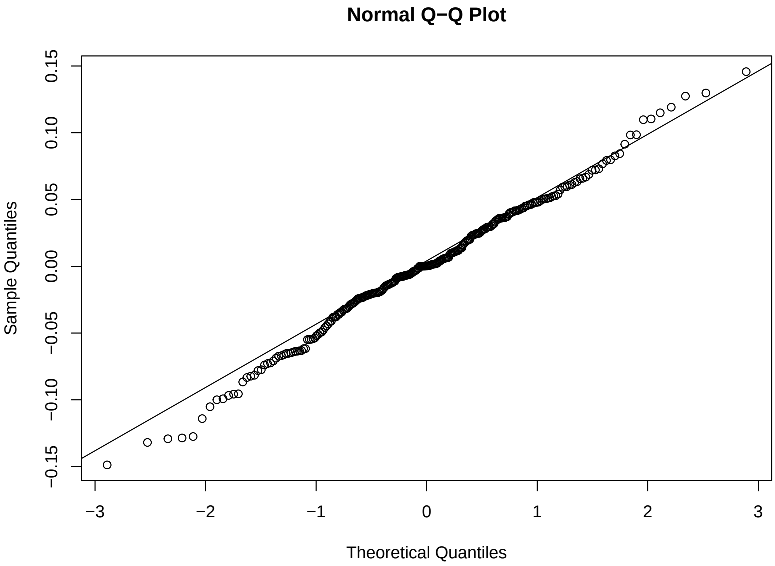
¿Cómo interpretar estos parámetros estimados?

Ejemplo aplicado a datos reales: Desempleo




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(0,1,1)[12]
## Q* = 21.265, df = 21, p-value = 0.4429
##
## Model df: 4.   Total lags used: 25
```

Ejemplo aplicado a datos reales: Desempleo



Ejemplo aplicado a datos reales: Desempleo

```
require(tseries)
modelo3_des$residuals %>% shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.99353, p-value = 0.3248
```

```
modelo3_des$residuals %>% jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  .
## X-squared = 1.3392, df = 2, p-value = 0.5119
```