Colgate vs Crest

Luis Vaciero

21/11/2020

Cargamos las librerías necesarias

```r
library(openxlsx)
library(skimr)
library(fpp2)
library(ggplot2)
library(zoo)
library(ggfortify)
library(tseries)
require(forecast)
require(xts)
library(readr)
library(tidyverse)
library(dplyr)
library(TSA)
library(Hmisc)
library(astsa)
library(tsoutliers)
library(normtest)
```

Marcamos la base de datos

```r
library(readxl)
data <- read_excel('data.xlsx')
View(data)
```

Indicamos las semanas

```r
data$Date <- as.Date(paste(data$Year, data$Week, 1, sep = "-"), "%Y-%U-%u")
data <- dplyr::select(data, -Year, -Week)
```

Summary

```r
skim(data)
```

Data summary

Name	data
Number of rows	276
Number of columns	3
Column type frequency:	
Date	1
numeric	2
Group variables	None

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
Date	0	1	1958-01-06	1963-04-22	1960-08-25	276

Variable type: numeric

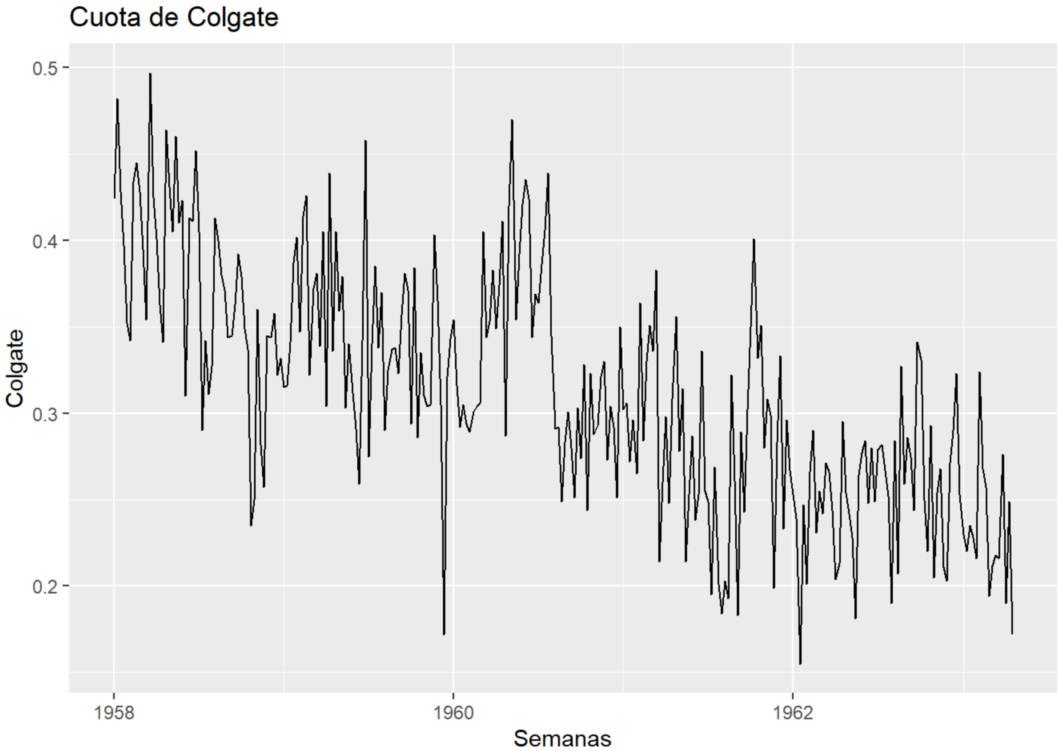
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Crest	0	1	0.26	0.13	0.05	0.13	0.25	0.37	0.53	▇▅▃▇▂
Colgate	0	1	0.31	0.07	0.16	0.26	0.31	0.36	0.50	▂▇▇▅▁

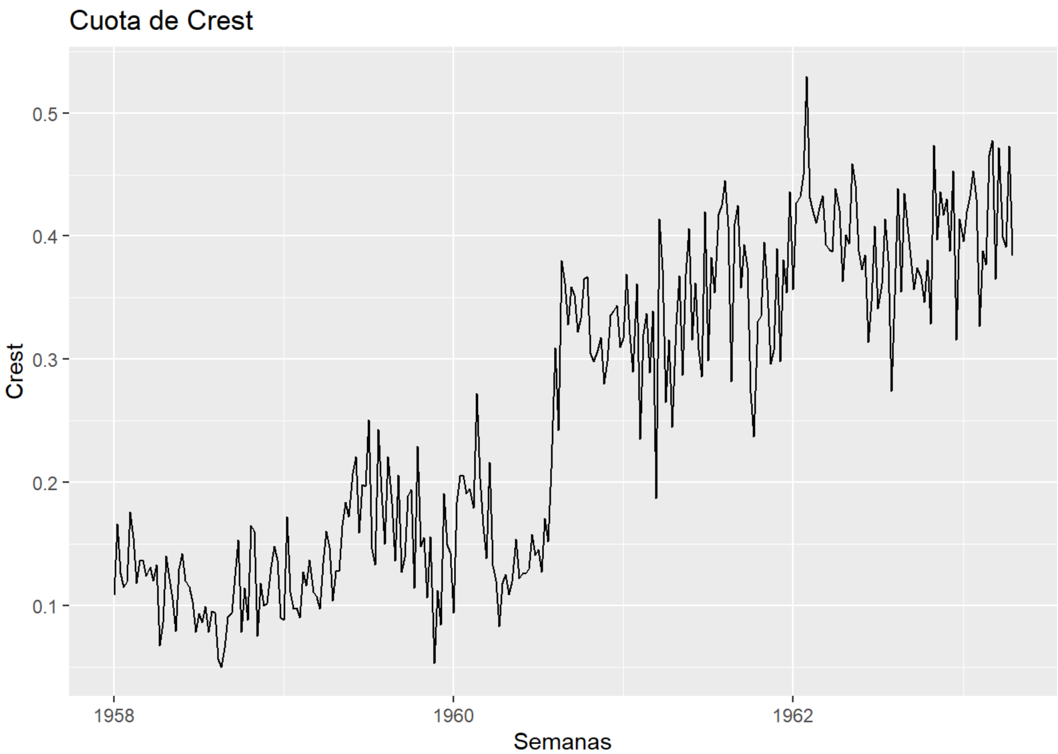
Dividimos la serie en 2 (Colgate y Crest).

Convertimos los datos.

```
acolgate <- as.zoo(colgate)
acrest <- as.zoo(crest)
```

Visualizamos ambas series





Test Dickey-Fuller

Test Dickey-Fuller (La Prueba de Dickey-Fuller busca determinar la existencia o no de raíces unitarias en una serie de tiempo. La hipótesis nula de esta prueba es que existe una raíz unitaria en la serie.)

```
##
##  Augmented Dickey-Fuller Test
##
## data:  acolgate
## Dickey-Fuller = -4.1783, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  acrest
## Dickey-Fuller = -3.4715, Lag order = 6, p-value = 0.04591
## alternative hypothesis: stationary
```

Ambos valores están cerca de superar el margen de significación marcado, pero sin sobrepasarlo, por lo que concluimos que existe estacionalidad.

Eliminamos las 16 semanas del año 1963, para las dos cuotas de mercado.

```
cOmit=16

nObsColgate=length(acolgate)
nObsCrest= length(acrest)
```

Seleccionamos el training set

```
colgatetrain <- window(acolgate, start=index(acolgate[1]),end = index(acolgate[nObsColgate- cOmit]))
cresttrain <- window(acrest, star= index(acrest[1]), end = index(acrest[nObsCrest-cOmit]))
```

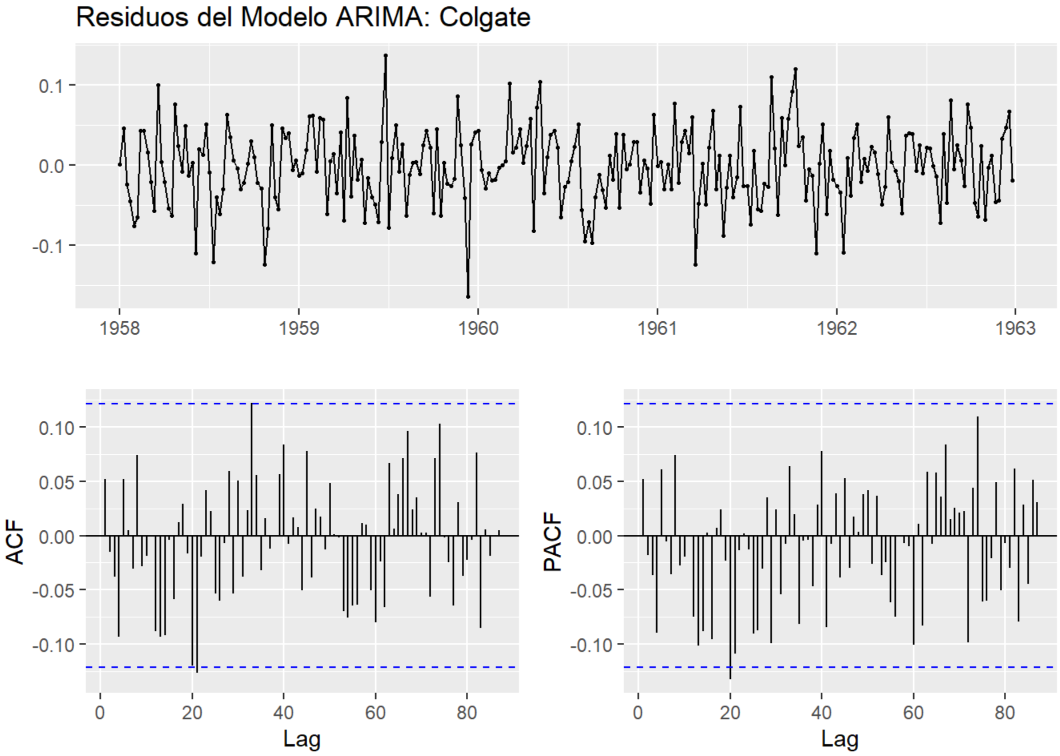
MODELO ARIMA

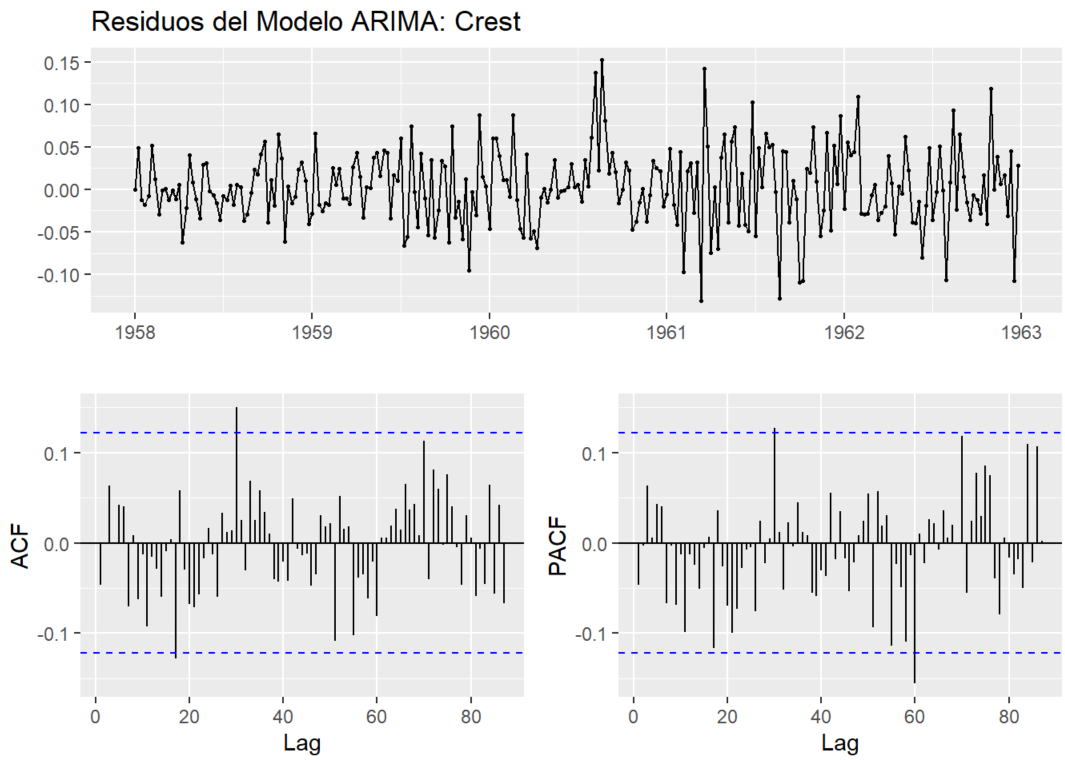
Usamos la funcion autoarima, que realiza las transformaciones necesarias para el modelo estacionario.

```
## Series: colgatetrain
## ARIMA(0,1,1)(1,0,0)[52]
##
## Coefficients:
##           ma1    sar1
##       -0.7571  0.0232
## s.e.   0.0470  0.0682
##
## sigma^2 estimated as 0.00232:  log likelihood=418.63
## AIC=-831.25   AICc=-831.16   BIC=-820.58
##
## Training set error measures:
##                        ME       RMSE        MAE       MPE     MAPE      MASE
## Training set -0.002534821 0.04788855 0.03773876 -3.019377 12.77398 0.6038201
##                    ACF1
## Training set 0.05216089
```

```
## Series: cresttrain
## ARIMA(0,1,1)
##
## Coefficients:
##           ma1
##       -0.6494
## s.e.   0.0448
##
## sigma^2 estimated as 0.002054:  log likelihood=434.08
## AIC=-864.15   AICc=-864.1   BIC=-857.04
##
## Training set error measures:
##                       ME       RMSE        MAE       MPE     MAPE      MASE
## Training set 0.003018071 0.04514444 0.03445351 -2.924537 17.27068 0.3800355
##                     ACF1
## Training set -0.04605961
```

Análisis de los residuos





Se puede observar en las funciones de autocorrelación, que existen retardos fuera de las bandas de confianza, debido a Outliers.

las series son estacionarias por lo que los residuos son “ruido blanco”. Ambos modelos son ARIMA(0,1,1) y los valores AIC son muy representativos, aunque como hemos observado hay información desajustada, eso es debido a los valores atípicos.

Box-Ljung Test (comprueba correlaciones)

```
Box.test(fit_colgate$residuals,lag=3, fitdf=1, type="Lj")
```

```
##
##  Box-Ljung test
##
## data:  fit_colgate$residuals
## X-squared = 1.1567, df = 2, p-value = 0.5608
```

```
Box.test(fit_crest$residuals,lag=3, fitdf=1, type="Lj")
```

```
##
##  Box-Ljung test
##
## data:  fit_crest$residuals
## X-squared = 1.6314, df = 2, p-value = 0.4423
```

```
#los valores son superiores al p-value, por lo que no rechazamos la correlación en los datos, lo que  result
a en aleatoriedad del proceso de muestreo o independencia)
```

Predicción

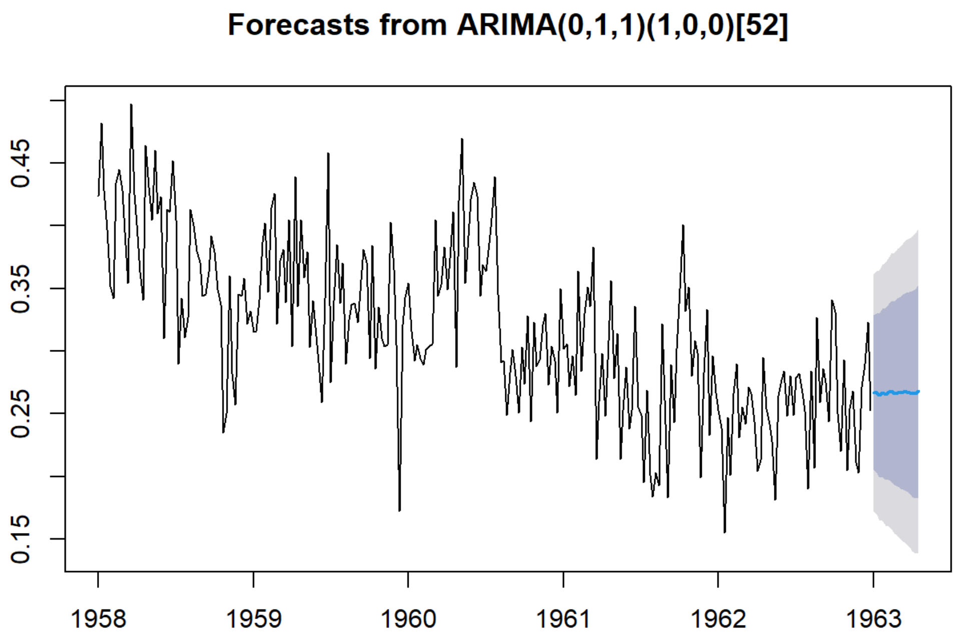
Aplicamos forecast

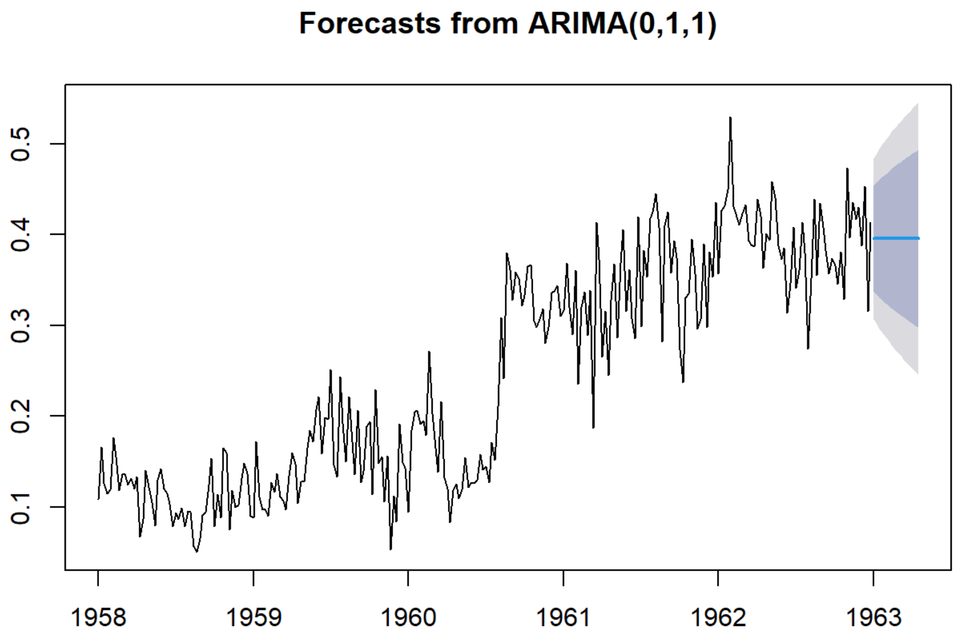
```
summary(cuota.arima.colgate)
```

```
##
## Forecast method: ARIMA(0,1,1)(1,0,0)[52]
##
## Model Information:
## Series: colgatetrain
## ARIMA(0,1,1)(1,0,0)[52]
##
## Coefficients:
##           ma1    sar1
##       -0.7571  0.0232
## s.e.   0.0470  0.0682
##
## sigma^2 estimated as 0.00232:  log likelihood=418.63
## AIC=-831.25   AICc=-831.16   BIC=-820.58
##
## Error measures:
##                        ME       RMSE        MAE       MPE     MAPE      MASE
## Training set -0.002534821 0.04788855 0.03773876 -3.019377 12.77398 0.6038201
##                    ACF1
## Training set 0.05216089
##
## Forecasts:
##          Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 1963.000      0.2669631 0.2052343 0.3286919 0.1725571 0.3613692
## 1963.019      0.2666376 0.2031138 0.3301615 0.1694863 0.3637890
## 1963.038      0.2646847 0.1994152 0.3299542 0.1648636 0.3645058
## 1963.058      0.2668236 0.1998539 0.3337933 0.1644023 0.3692449
## 1963.077      0.2657542 0.1971264 0.3343819 0.1607971 0.3707113
## 1963.096      0.2672654 0.1970186 0.3375121 0.1598323 0.3746985
## 1963.115      0.2678234 0.1959942 0.3396525 0.1579701 0.3776766
## 1963.135      0.2664516 0.1930741 0.3398292 0.1542304 0.3786729
## 1963.154      0.2670096 0.1921157 0.3419035 0.1524693 0.3815500
## 1963.173      0.2667074 0.1903272 0.3430875 0.1498940 0.3835207
## 1963.192      0.2673816 0.1895436 0.3452196 0.1483386 0.3864246
## 1963.212      0.2672654 0.1879963 0.3465345 0.1460338 0.3884970
## 1963.231      0.2667539 0.1860791 0.3474287 0.1433725 0.3901353
## 1963.250      0.2658239 0.1837675 0.3478803 0.1403295 0.3913183
## 1963.269      0.2660332 0.1826180 0.3494483 0.1384607 0.3936056
## 1963.288      0.2679396 0.1831875 0.3526917 0.1383225 0.3975567
```

```
summary(cuota.arima.crest)
```

```
##
## Forecast method: ARIMA(0,1,1)
##
## Model Information:
## Series: cresttrain
## ARIMA(0,1,1)
##
## Coefficients:
##           ma1
##       -0.6494
## s.e.   0.0448
##
## sigma^2 estimated as 0.002054:  log likelihood=434.08
## AIC=-864.15   AICc=-864.1   BIC=-857.04
##
## Error measures:
##                       ME       RMSE        MAE       MPE     MAPE      MASE
## Training set 0.003018071 0.04514444 0.03445351 -2.924537 17.27068 0.3800355
##                     ACF1
## Training set -0.04605961
##
## Forecasts:
##          Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 1963.000      0.3958305 0.3377518 0.4539092 0.3070067 0.4846543
## 1963.019      0.3958305 0.3342850 0.4573760 0.3017048 0.4899563
## 1963.038      0.3958305 0.3310033 0.4606577 0.2966859 0.4949751
## 1963.058      0.3958305 0.3278800 0.4637810 0.2919092 0.4997519
## 1963.077      0.3958305 0.3248940 0.4667670 0.2873425 0.5043185
## 1963.096      0.3958305 0.3220288 0.4696322 0.2829605 0.5087005
## 1963.115      0.3958305 0.3192707 0.4723903 0.2787424 0.5129186
## 1963.135      0.3958305 0.3166086 0.4750525 0.2746710 0.5169900
## 1963.154      0.3958305 0.3140330 0.4776280 0.2707321 0.5209290
## 1963.173      0.3958305 0.3115362 0.4801249 0.2669134 0.5247476
## 1963.192      0.3958305 0.3091111 0.4825499 0.2632047 0.5284563
## 1963.212      0.3958305 0.3067521 0.4849089 0.2595969 0.5320642
## 1963.231      0.3958305 0.3044540 0.4872070 0.2560822 0.5355789
## 1963.250      0.3958305 0.3022122 0.4894488 0.2526537 0.5390073
## 1963.269      0.3958305 0.3000229 0.4916381 0.2493054 0.5423556
## 1963.288      0.3958305 0.2978825 0.4937785 0.2460320 0.5456290
```





Detectamos Outliers

```
## [1] "No AO detected"
```

```
##                [,1]       [,2]       [,3]
## ind      135.000000 136.000000 138.000000
## lambda2    3.918954   4.372891   4.005427
```

```
## [1] "No IO detected"
```

```
## [1] "No IO detected"
```

El AO detectado en 135 es el impulso explicado por el comunicado hecho el 1 de agosto de 1960, cuando el Consejo de Terapéutica Dental de la American Dental Association (ADA) aprobó a Crest como una "ayuda importante en cualquier programa de higiene dental".Los restantes, ambos 136 y 138 pueden interpretarse como efecto rebote, sin justicacion clara.

ARIMAX

Realizamos el modelo arimax con los datos obtenidos con el modelo ARIMA

```
## 
## Call:
## arimax(x = as.double(acolgate), order = c(0, 1, 1), method = "ML", xtransf = data.frame(seqx = 1 *
##     (seq(acolgate)), seqy = 1 * (seq(acolgate))), transfer = list(c(0, 0), c(1,
##     0)))
## 
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): Se han producido NaNs
```

```
##           ma1  seqx-MA0  seqy-AR1  seqy-MA0
##       -0.7730    -4e-04         0   -0.0004
## s.e.   0.0455       NaN       NaN    0.0072
## 
## sigma^2 estimated as 0.002256:  log likelihood = 447.31,  aic = -886.61
```

Introducimos los AOs dentro del modelo Arimax.

```
##
## Call:
## arimax(x = as.double(acrest), order = c(0, 1, 1), xreg = data.frame(I136 = 1 *
##     (seq(acrest) == 136), I138 = 1 * (seq(acrest) == 138)), method = "ML", xtransf = data.frame(seqx = 1
*
##     (seq(acrest) >= 135), seqy = 1 * (seq(acrest))), transfer = list(c(0, 0),
##     c(0, 0)))
##
## Coefficients:
##           ma1    I136    I138  seqx-MA0  seqy-MA0
##       -0.7674  0.0205  0.0766    0.1346     5e-04
## s.e.   0.0476  0.0436  0.0420    0.0318     6e-04
##
## sigma^2 estimated as 0.001899:  log likelihood = 470.97,  aic = -931.94
```

Comprobamos si hemos conseguido limpiar los outliers

```
## [1] "No AO detected"
```

```
## [1] "No IO detected"
```

```
## [1] "No AO detected"
```

```
## [1] "No IO detected"
```

Podemos observar como han desaparecido

Funcion de transferencia

```
##
## Call:
## arimax(x = as.double(acolgate), order = c(0, 1, 1), include.mean = TRUE, method = "ML",
##     xtransf = acrest, transfer = list(c(0, 0)))
##
## Coefficients:
##           ma1  xtransf-MA0
##       -0.8381      -0.4884
## s.e.   0.0346       0.0530
##
## sigma^2 estimated as 0.001751:  log likelihood = 481.99,  aic = -959.97
```

```
##
## Call:
## arimax(x = as.double(acolgate), order = c(0, 1, 1), include.mean = TRUE, method = "ML",
##     xtransf = acrest, transfer = list(c(0, 0)))
##
## Coefficients:
##           ma1  xtransf-MA0
##       -0.8381      -0.4884
## s.e.   0.0346       0.0530
##
## sigma^2 estimated as 0.001751:  log likelihood = 481.99,  aic = -959.97
##
## Training set error measures:
##                       ME      RMSE        MAE       MPE     MAPE      MASE
## Training set -0.00136496 0.0417676 0.03229819 -2.196369 10.97291 0.6993702
##                    ACF1
## Training set 0.06132245
```