Crest y Colgate

José María Álvarez Silva

21/11/2019

CUNEF - Master en Data Science para Finanzas Predicción

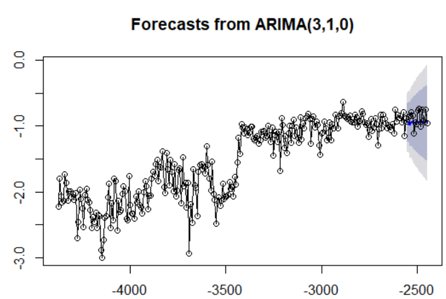
Propósito

Predicción del *Market Share* de crest y colgate. El objetivo es predecir las 16 semanas del año 1963, para las dos cuotas de mercado, tomando en cuenta el evento del ADA (1 de agosto de 1960, el Consejo de Terapéutica Dental de la American Dental Association (ADA) aprobó a Crest como una "ayuda importante en cualquier programa de higiene dental"). Este evento parece afectar la participación de mercado de ambos competidores por lo que es importante tenerlo en cuenta a la hora de predecir. La predicción se realizó con modelos ARIMA. Adicionalmente, un modelo de función de tranferencia entre las dos cuotas.

Predicción

Se llevo a cabo la predicción de la participación de mercado de Crest y Colgate para las próximas 16 semanas. La predicción de Crest:

1963	MS	MS	MS	MS
1-4	0.387001	0.395869	0.376461	0.391011
5-8	0.387204	0.388488	0.385771	0.387830
9-12	0.387281	0.387468	0.387088	0.387377
13-16	0.387298	0.387325	0.387272	0.387313

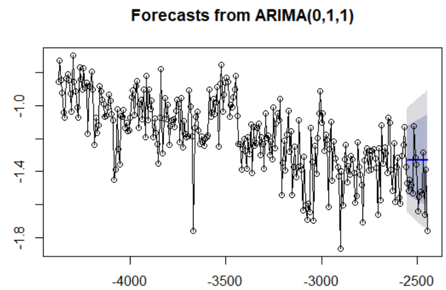


El modelo utilizado fue un modelo "ARIMA(3,1,0)" sobre el *Market Share* de Crest. (escala logarítmica)

La predicción de Colgate:

1963	MS	MS	MS	MS
1-4	0.264273	0.264273	0.264273	0.264273
5-8	0.264273	0.264273	0.264273	0.264273
9-12	0.264273	0.264273	0.264273	0.264273

1963	MS	MS	MS	MS
13-16	0.264273	0.264273	0.264273	0.264273



El modelo utilizado fue un modelo “ARIMA(0,1,1)” sobre el *Market Share* de Colgate. (escala logarítmica)

Resumen Ejecutivo

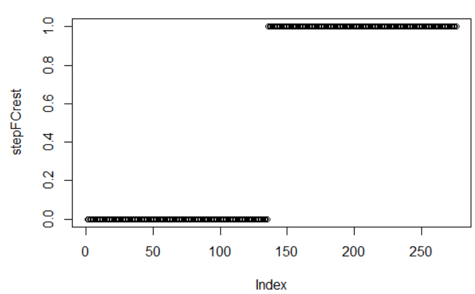
Proceso General

En este análisis utilizamos dos enfoques para la predicción del *Market Share* de las siguientes 16 semanas de **Crest** y **Colgate**. Se trabajó con el logaritmo de las series de tiempo del *Market Share* semanal de cada compañía desde 1958 a 1963. Los enfoques utilizados para predicir:

- ARIMA.
- ARIMAX - ARIMA con intervención.

Resultados

Para tomar encuenta ADA se realizara un ajuste al modelar cada serie introduciendo una intervencion de tipo step:

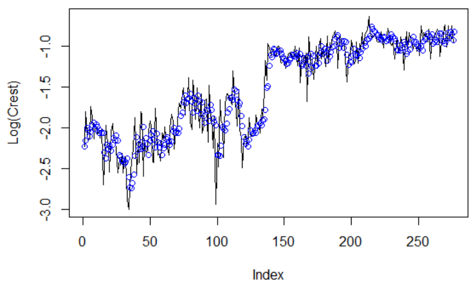


CREST

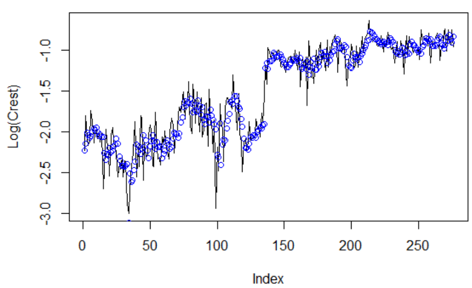
Comparando el poder predictivo de cada uno de los modelos (generados con el set de entrenamiento) a través de la métricas de error de predicción (MSE, MAE y Bias) en el set de Test (*Market Share* por compañia menos las últimas 16 semanas), el modelo que tuvo mejor desempeño fue el "" sobre el *Market Share* de Crest (escala logarítmica). Como se muestra en la tabla a continuación:

	MSE	MAE	Bias
Crest ARIMA	0.015499	0.098578	-0.06508
Crest ARIMAX	" "	" "	" "

Ajuste ARIMA



Ajuste ARIMA con intervención

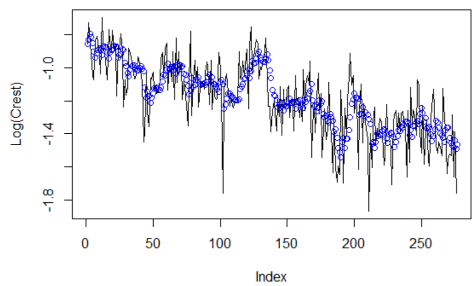


COLGATE

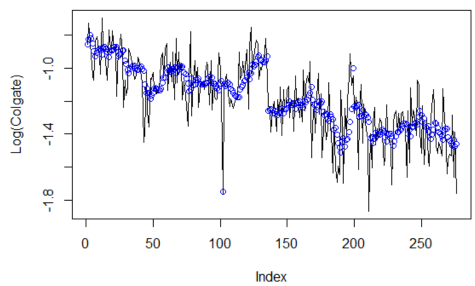
Comparando el poder predictivo de cada uno de los modelos (generados con el set de entrenamiento) a través de la métricas de error de predicción (MSE, MAE y Bias) en el set de Test (*Market Share* por compañia menos las últimas 16 semanas), el modelo que tuvo mejor desempeño fue el "" sobre el *Market Share* de Colgate (escala logarítmica). Como se muestra en la tabla a continuación:

	MSE	MAE	Bias
Colgate ARIMA	0.043269	0.17629	0.143651
Colgate ARIMAX	" "	" "	" "

Ajuste ARIMA



Ajuste ARIMA con intervención



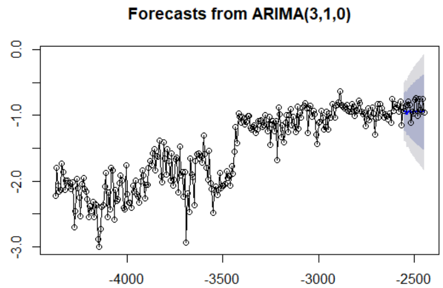
Predicción

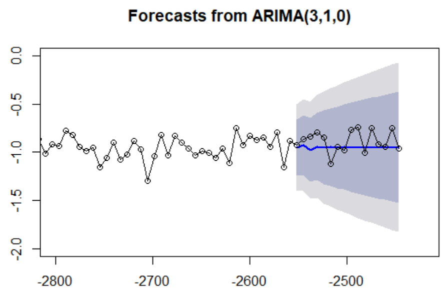
Predicción Crest

Se llevo a cabo la predicción de la participación de mercado de Crest para las próximas 16 semanas. La predicción de Crest:

1963	MS	MS	MS	MS
1-4	0.387001	0.395869	0.376461	0.391011
5-8	0.387204	0.388488	0.385771	0.387830
9-12	0.387281	0.387468	0.387088	0.387377
13-16	0.387298	0.387325	0.387272	0.387313

Predicción ARIMA



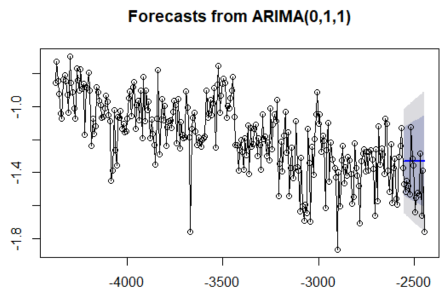


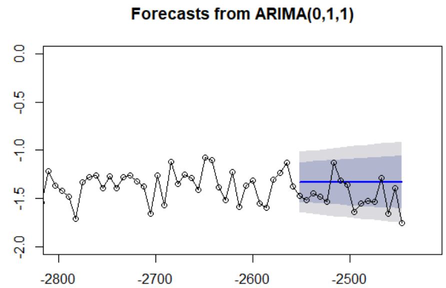
Predicción Colgate

Se llevo a cabo la predicción de la participación de mercado de Colgate para las próximas 16 semanas (escala logarítmica). La predicción de Colgate:

1963	MS	MS	MS	MS
1-4	-1.330771	-1.330771	-1.330771	-1.330771
5-8	-1.330771	-1.330771	-1.330771	-1.330771
9-12	-1.330771	-1.330771	-1.330771	-1.330771
13-16	-1.330771	-1.330771	-1.330771	-1.330771

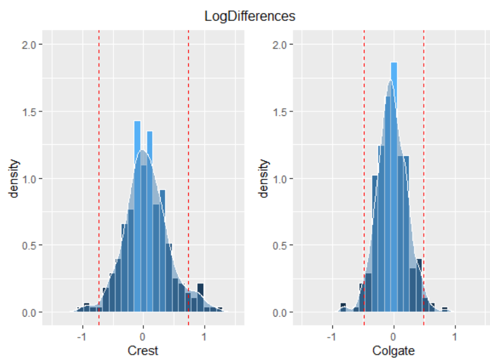
Predicción ARIMA



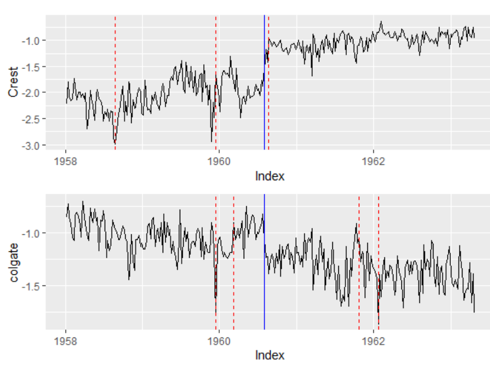


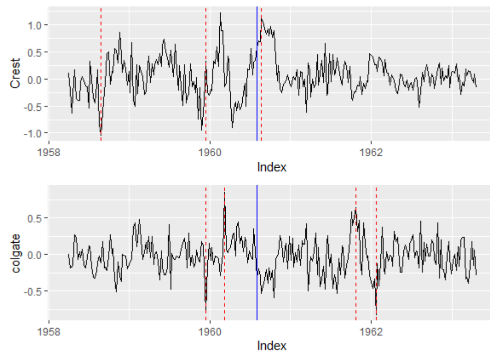
Detección de Outliers

AL analizar las diferencias notamos un comportamiento normal (aproximadamente) por lo que una forma de buscar outliers es buscando valores en las colas.



Al establecer una regla para detección de outliers encontramos cuatro fechas para colgate; pero, el primero esta relacionado con el segundo y el tercero con el cuarto, por lo que al introducirlo al ARIMAX solo tomamos en cuenta el primero y el tercero (además de ADA). En el caso de crest, encontramos dos y uno que coincide con ADA.





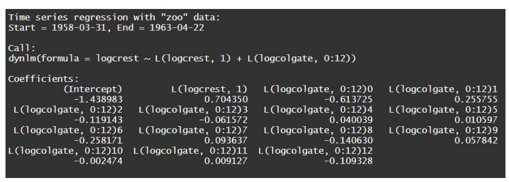
Detección Automática

La Deteccion automatica resulta en uno de los mismos outliers antes encontrados para las dos series.

Modelo de función de tranferencia

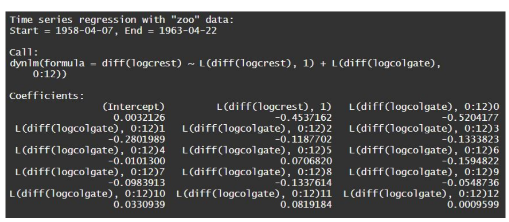
Modelo Dinámico

Explicar Crest con Crest pasado y Colgate



```
Time series regression with "zoo" data:
Start = 1958-03-31, End = 1963-04-22

Call:
dynlm(formula = logcrest ~ L(logcrest, 1) + L(logcolgate, 0:12))

Coefficients:
          (Intercept)        L(logcrest, 1)  L(logcolgate, 0:12)0  L(logcolgate, 0:12)1
            -1.438983              0.704350             -0.613725              0.255755
 L(logcolgate, 0:12)2  L(logcolgate, 0:12)3  L(logcolgate, 0:12)4  L(logcolgate, 0:12)5
            -0.119143             -0.061572              0.040039              0.010597
 L(logcolgate, 0:12)6  L(logcolgate, 0:12)7  L(logcolgate, 0:12)8  L(logcolgate, 0:12)9
            -0.258171              0.093637             -0.140630              0.057842
L(logcolgate, 0:12)10 L(logcolgate, 0:12)11 L(logcolgate, 0:12)12
            -0.002474              0.009127             -0.109328
```

Explicar las diferencias de Crest con las de Crest pasado y Colgate

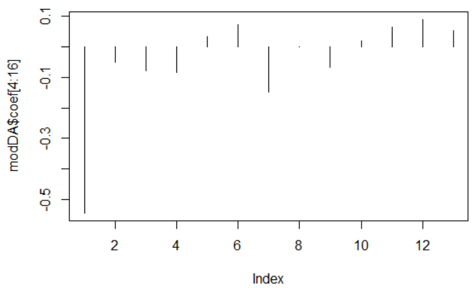


```
Time series regression with "zoo" data:
Start = 1958-04-07, End = 1963-04-22

Call:
dynlm(formula = diff(logcrest) ~ L(diff(logcrest), 1) + L(diff(logcolgate),
    0:12))

Coefficients:
                (Intercept)         L(diff(logcrest), 1)  L(diff(logcolgate), 0:12)0
                  0.0032126                   -0.4537162                  -0.5204177
 L(diff(logcolgate), 0:12)1   L(diff(logcolgate), 0:12)2  L(diff(logcolgate), 0:12)3
                 -0.2801989                   -0.1187702                  -0.1333823
 L(diff(logcolgate), 0:12)4   L(diff(logcolgate), 0:12)5  L(diff(logcolgate), 0:12)6
                 -0.0101300                    0.0706820                  -0.1594822
 L(diff(logcolgate), 0:12)7   L(diff(logcolgate), 0:12)8  L(diff(logcolgate), 0:12)9
                 -0.0983913                   -0.1337614                  -0.0548736
L(diff(logcolgate), 0:12)10  L(diff(logcolgate), 0:12)11 L(diff(logcolgate), 0:12)12
                  0.0330939                    0.0819184                   0.0009599
```

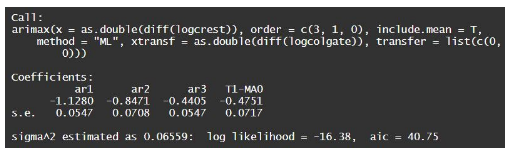
Modelo de Función de Tranferencia

Buscamos los coeficientes significativos:

Con suficientes lags:



Coeficientes significativos:



```
Call:
arimax(x = as.double(diff(logcrest)), order = c(3, 1, 0), include.mean = T,
    method = "ML", xtransf = as.double(diff(logcolgate)), transfer = list(c(0,
        0)))

Coefficients:
          ar1      ar2      ar3   T1-MA0
      -1.1280  -0.8471  -0.4405  -0.4751
s.e.   0.0547   0.0708   0.0547   0.0717

sigma^2 estimated as 0.06559:  log likelihood = -16.38,  aic = 40.75
```

Todos son significativos (Coeficiente entre s.e. es mayor a 2).

Observaciones

Al existir un cambio tan radical en el comportamiento de las cuotas del mercado de productos dentales es importante tener encuenta el evento de ADA ya que puede afectar el ajuste que se hace al modelor. Al observar el ajuste de las series, tanto de Crest como de Colgate, se observa claramente como se logra un mejor ajuste con los modelos ARIMA con intervención. La misma Lógica aplica para eventos aleatorios que pueden afectar el ajuste (outliers) por lo que es importante, después de analizarlos, considerarlos a la hora de modelar.

No fue posible realizar una predicción con el modelo ARIMAX.

Referencias

- https://stats.stackexchange.com/questions/18375/how-to-fit-an-arimax-model-with-r
- https://stackoverflow.com/questions/25224155/transfer-function-models-arimax-in-tsa
- https://rpubs.com/simasiami/378726
- https://cran.r-project.org/web/packages/TSA/TSA.pdf
- https://stats.stackexchange.com/questions/169564/arimax-prediction-using-forecast-package

Anexos

```
##########################################################################################################
## Start Date: 18/11/2019
## End Date:   -
## Author:     José María Álvarez Silva
## School:     CUNEF
## Class:      Predicción
```

```r
## Assigment:  Crest y Colgate
## Language:   Spanish
##
########################################################################################################
## Predicción
########################################################################################################
## Primas Mapfre ######################################################################################
##     ARIMAX y ARIMAS

## Propósito ##########################################################################################
##     Predicción del MS de Crest y Colgate con intervencion y sin intervención.
```

Paquetes

```r
## Paquetes ###########################################################################################
##
library(dplyr)
library(tidyverse)
library(forecast)
library(xts)
library(ggplot2)
library(zoo)
library(ggfortify)
library(skimr)
library(gridExtra)
library(ggpubr)
library(TSA)
library(Hmisc)
library(astsa)
library(dynlm)
```

Datos

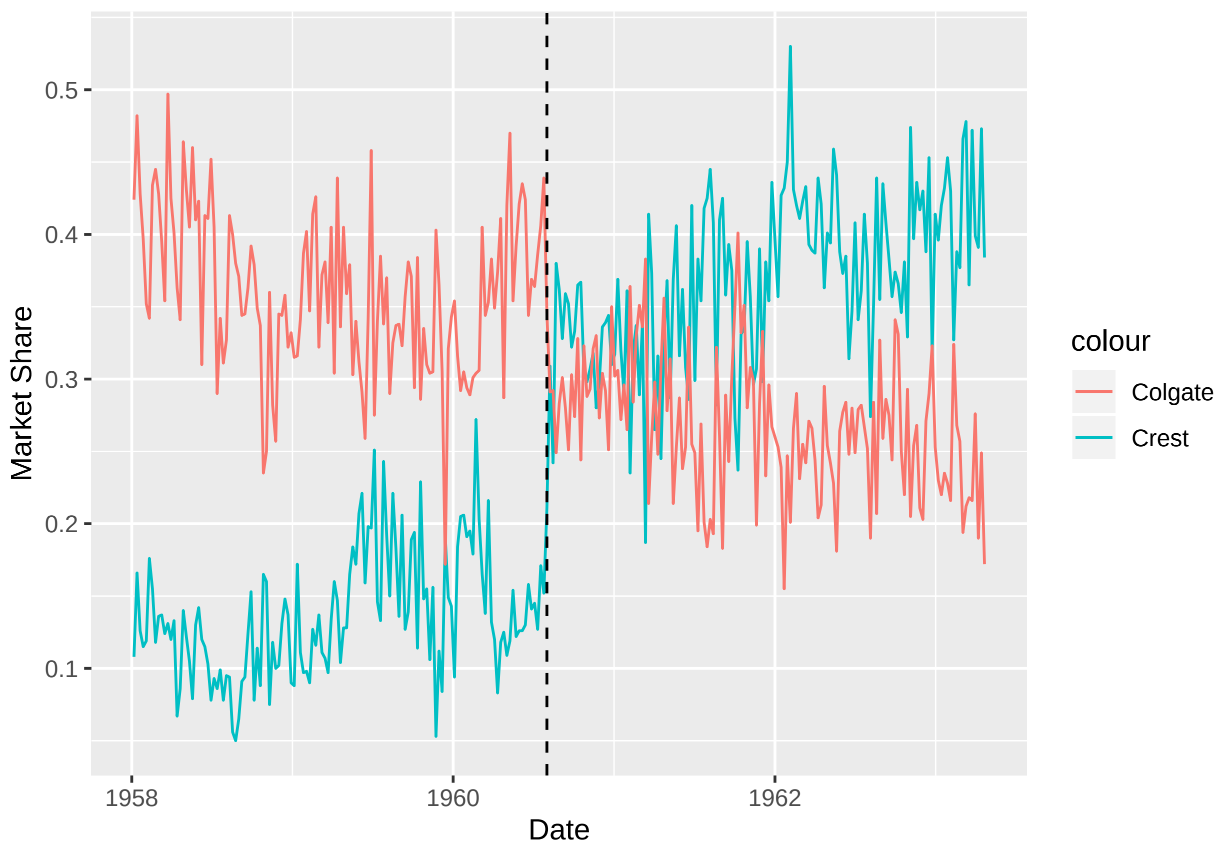
```r
## Datos ##############################################################################################
##
datos <- read.csv("data.csv")

datos$Date <- as.Date(paste(datos$Year, datos$Week, 1, sep = "-"), "%Y-%U-%u")

skim(datos)
```

```
## Skim summary statistics
##  n obs: 276
##  n variables: 5
##
## -- Variable type:Date ---------------------------------------------------------------------------
##  variable missing complete   n        min        max     median n_unique
##      Date       0      276 276 1958-01-06 1963-04-22 1960-08-25      276
##
```

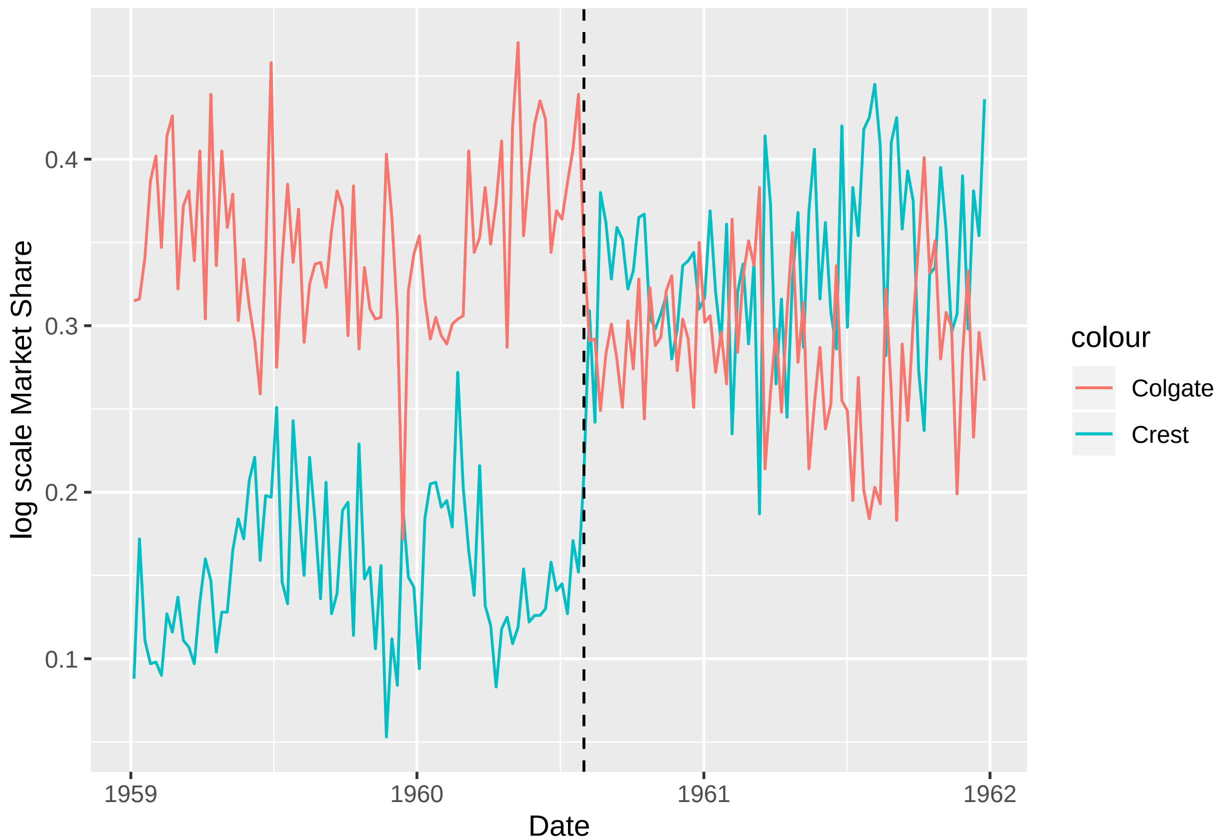
```
## -- Variable type:integer --------------------------------------------------------------------------------
##  variable missing complete   n    mean    sd   p0  p25  p50  p75 p100
##      Week       0      276 276   25.46 15.23    1   12   25   39   52
##      Year       0      276 276 1960.17  1.54 1958 1959 1960 1961 1963
##      hist
##  <U+2587><U+2587><U+2587><U+2586><U+2586><U+2587><U+2586><U+2587>
##  <U+2587><U+2587><U+2581><U+2587><U+2587><U+2581><U+2587><U+2582>
##
## -- Variable type:numeric --------------------------------------------------------------------------------
##  variable missing complete   n mean    sd   p0  p25  p50  p75 p100
##   Colgate       0      276 276 0.31 0.069 0.15 0.26 0.31 0.36 0.5
##     Crest       0      276 276 0.26 0.13  0.05 0.13 0.25 0.37 0.53
##      hist
##  <U+2582><U+2583><U+2587><U+2587><U+2587><U+2585><U+2582><U+2581>
##  <U+2585><U+2587><U+2583><U+2582><U+2585><U+2586><U+2583><U+2581>
```

Análisis Exploratorio de Datos

```r
## Series
ggplot(data = datos, aes(x = Date)) +
  geom_line(aes(y = Crest, colour = "Crest")) +
  geom_line(aes(y = Colgate, colour = "Colgate")) +
  ylab("Market Share") +
  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 2) ## 1 agosto 1960
```



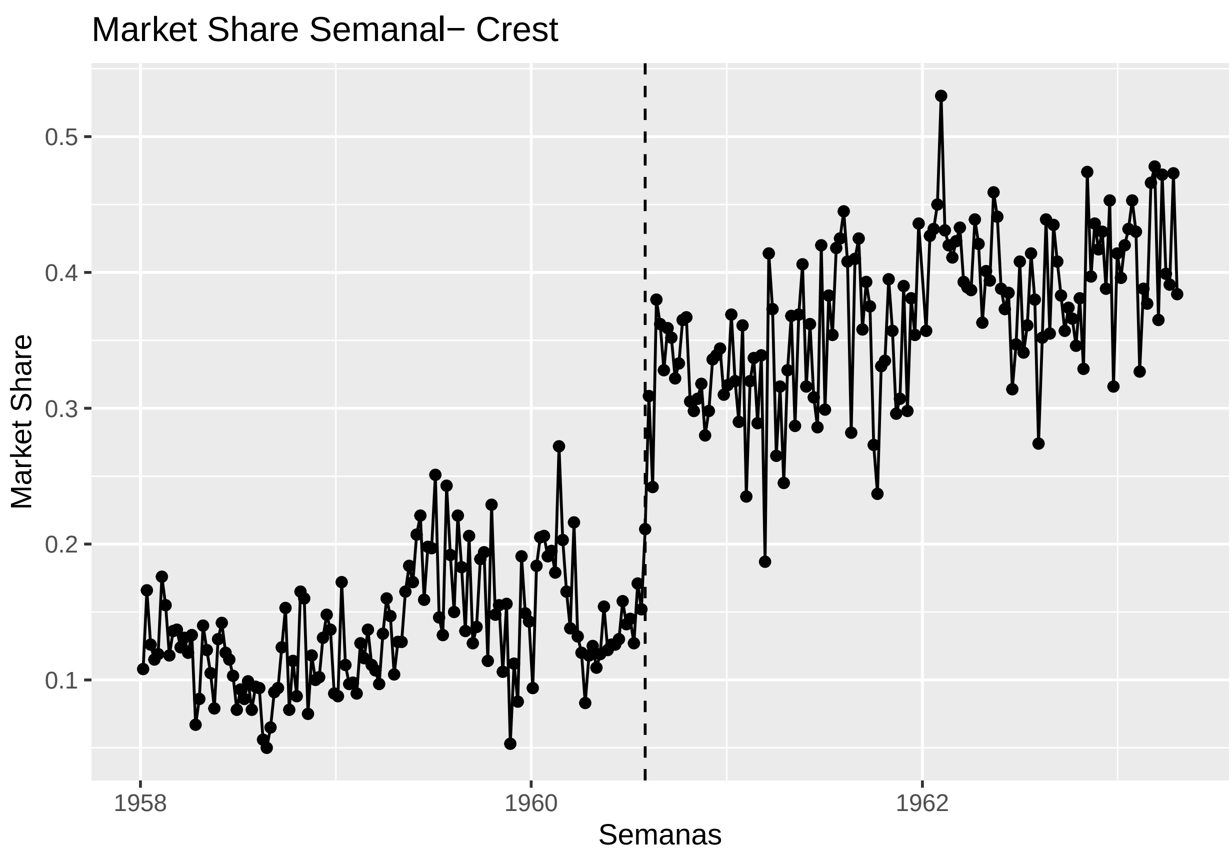
```r
ggplot(data = filter(datos, Year == 1959 | Year == 1960 | Year == 1961), aes(x = Date)) +
  geom_line(aes(y = Crest, colour = "Crest")) +
  geom_line(aes(y = Colgate, colour = "Colgate")) +
  ylab("log scale Market Share") +
  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 2) ## 1 agosto 1960
```



Crest

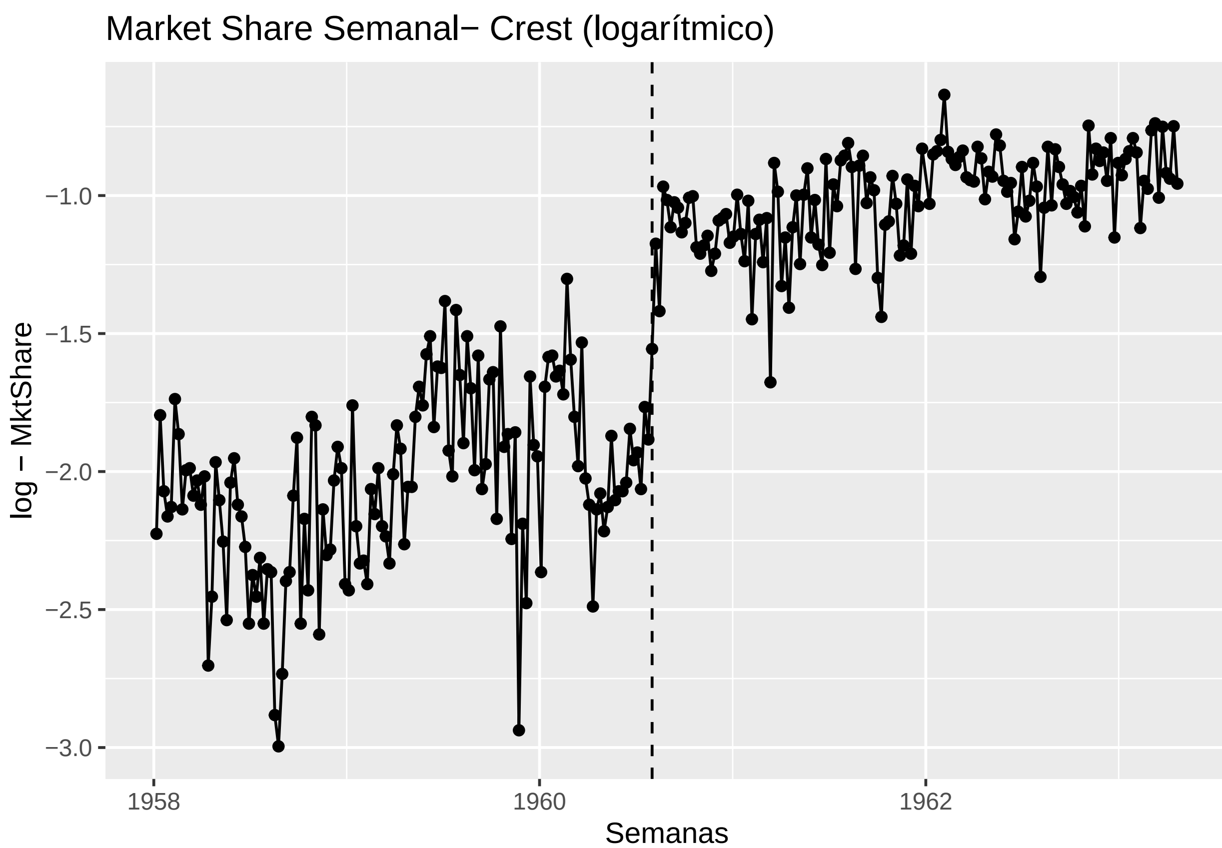
```r
## Plot Serie crest
crest = xts((datos$Crest), order.by = datos$Date)
colnames(crest) <- "Crest"
## paqueteria zoo para mejor funcionamiento
crest = as.zoo(crest$Crest)
#autoplot(crest) + ggtitle("Market Share Semanal - Crest") + xlab("Semanas") + ylab("Market Share") +
#  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 2)

## Nuestra ts de market share de Crest de llama Crest

df_crest <- data.frame(value = as.vector(crest),
                       time = time(crest))
ggplot(df_crest) + geom_point(aes(x = time, y = value)) +
  geom_line(aes(x = time, y = value)) +
  ylab("Market Share") +
```

```
  ggtitle("Market Share Semanal- Crest") +
  xlab("Semanas") +
  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 2)
```



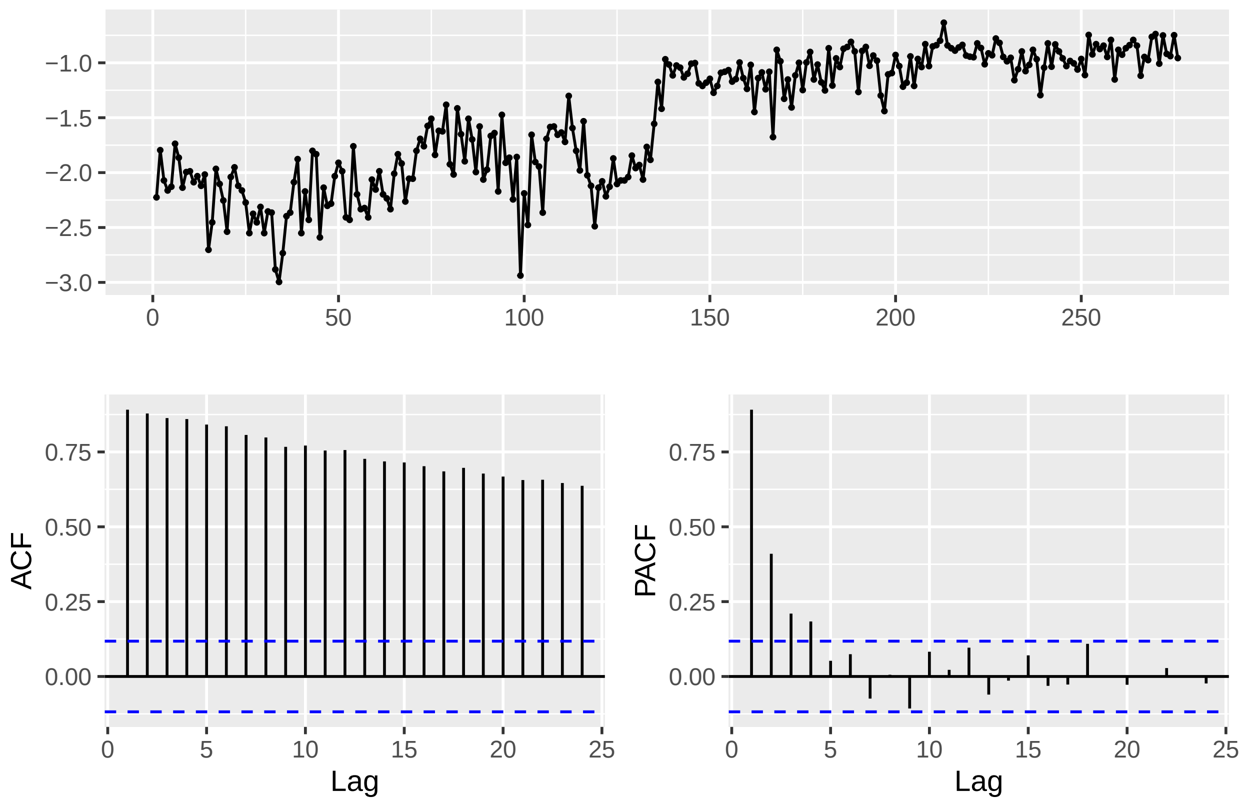
LogCrest

```
## trabajamos con transformacion logaritmica
logcrest <- log(crest)
df_logcrest <- data.frame(value = as.vector(logcrest),
                          time = time(logcrest))
ggplot(df_logcrest) + geom_point(aes(x = time, y = value)) +
  geom_line(aes(x = time, y = value)) +
  ylab("log - MktShare") +
  ggtitle("Market Share Semanal- Crest (logarítmico)") +
  xlab("Semanas") +
  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 2)
```

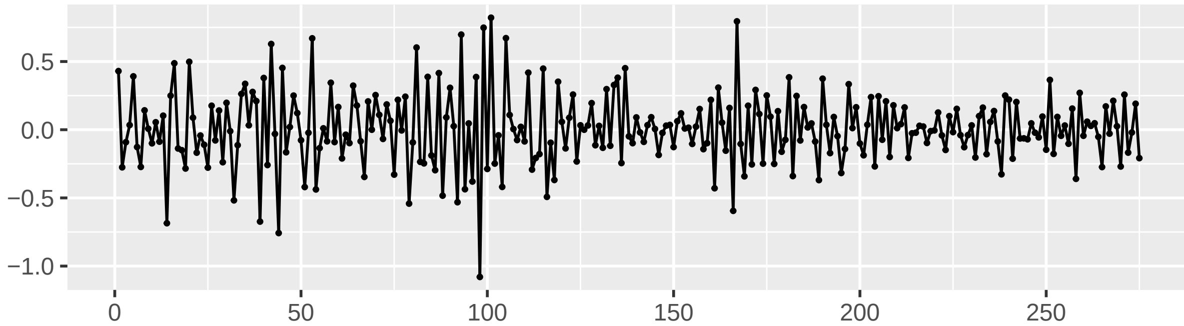


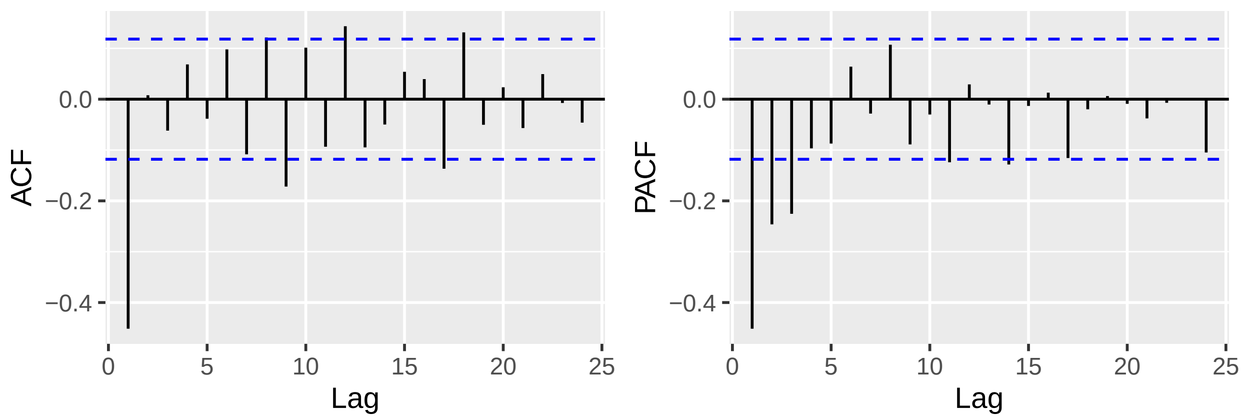
Diferenciando la serie

```r
## Difference
ggtsdisplay(logcrest)
```




```
ggtsdisplay(diff(logcrest))
```






```
which(diff(logcrest) == max(diff(logcrest)))
```

```
## [1] 101
```

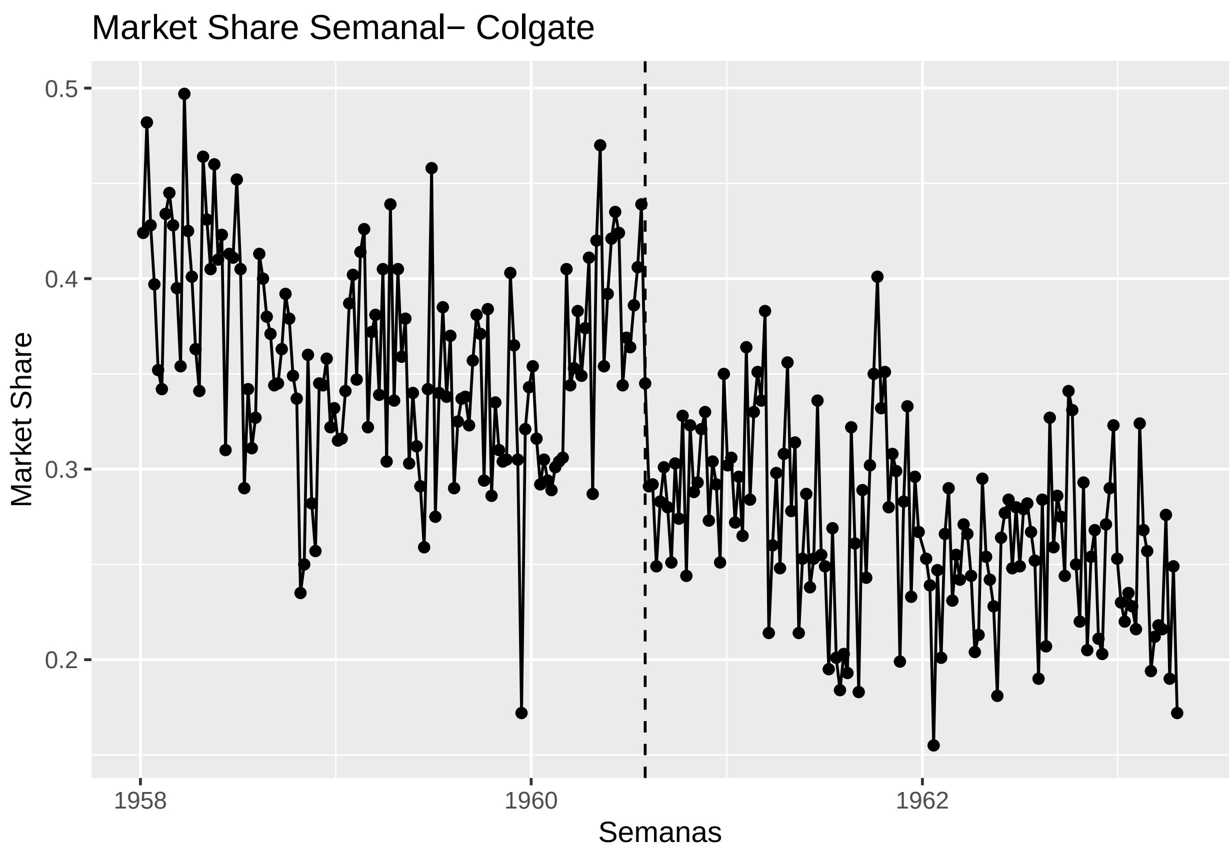
```
which(diff(logcrest) == min(diff(logcrest)))
```

```
## [1] 98
```

Colgate

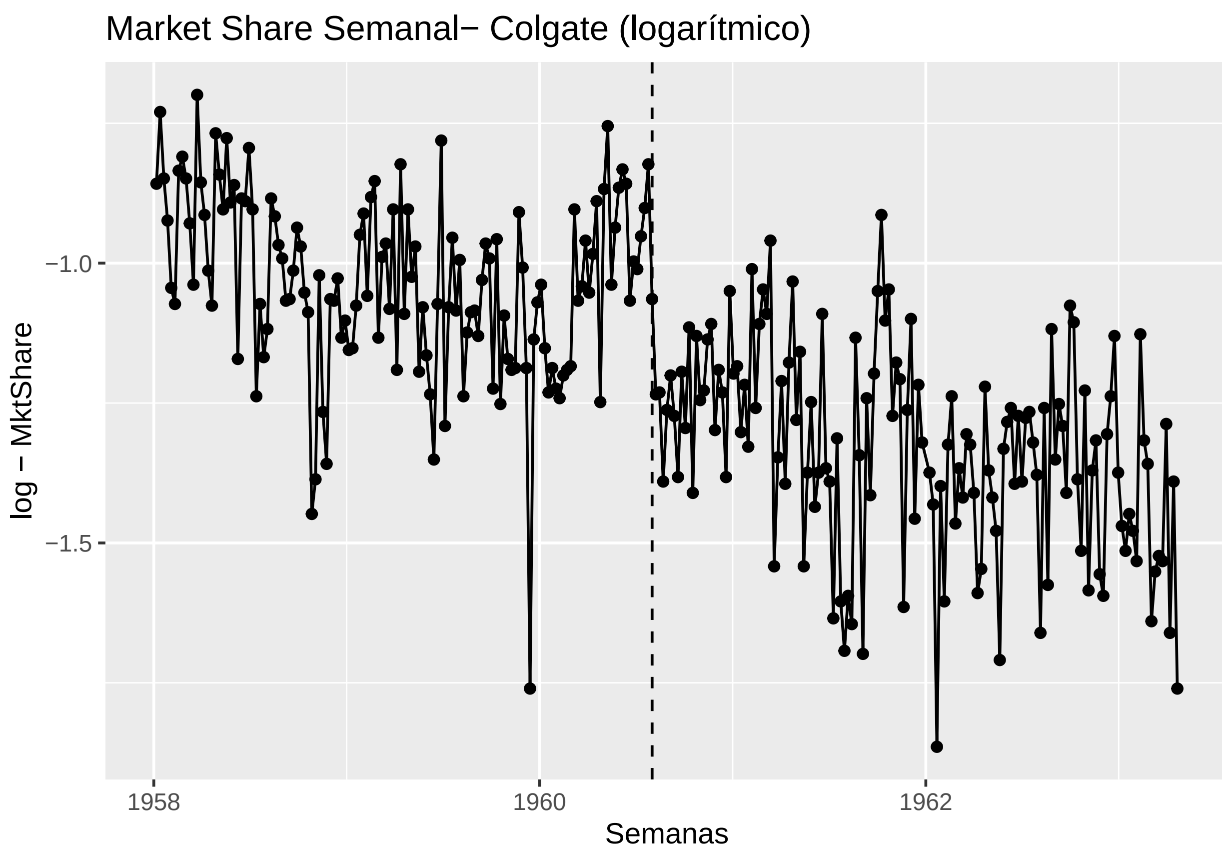
```
#-#- Colgate #-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-

## Plot Serie colgate
colgate = xts((datos$Colgate), order.by = datos$Date)
colnames(colgate) <- "colgate"
## paqueteria zoo para mejor funcionamiento
colgate = as.zoo(colgate$colgate)
#autoplot(colgate) + ggtitle("Market Share Semanal - Colgate") + xlab("Semanas") + ylab("Market Share")
#  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 2)

## Nuestra ts de market share de colgate de llama colgate

df_colgate <- data.frame(value = as.vector(colgate),
```

```r
                         time = time(colgate))
ggplot(df_colgate) + geom_point(aes(x = time, y = value)) +
  geom_line(aes(x = time, y = value)) +
  ylab("Market Share") +
  ggtitle("Market Share Semanal- Colgate") +
  xlab("Semanas") +
  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 2)
```



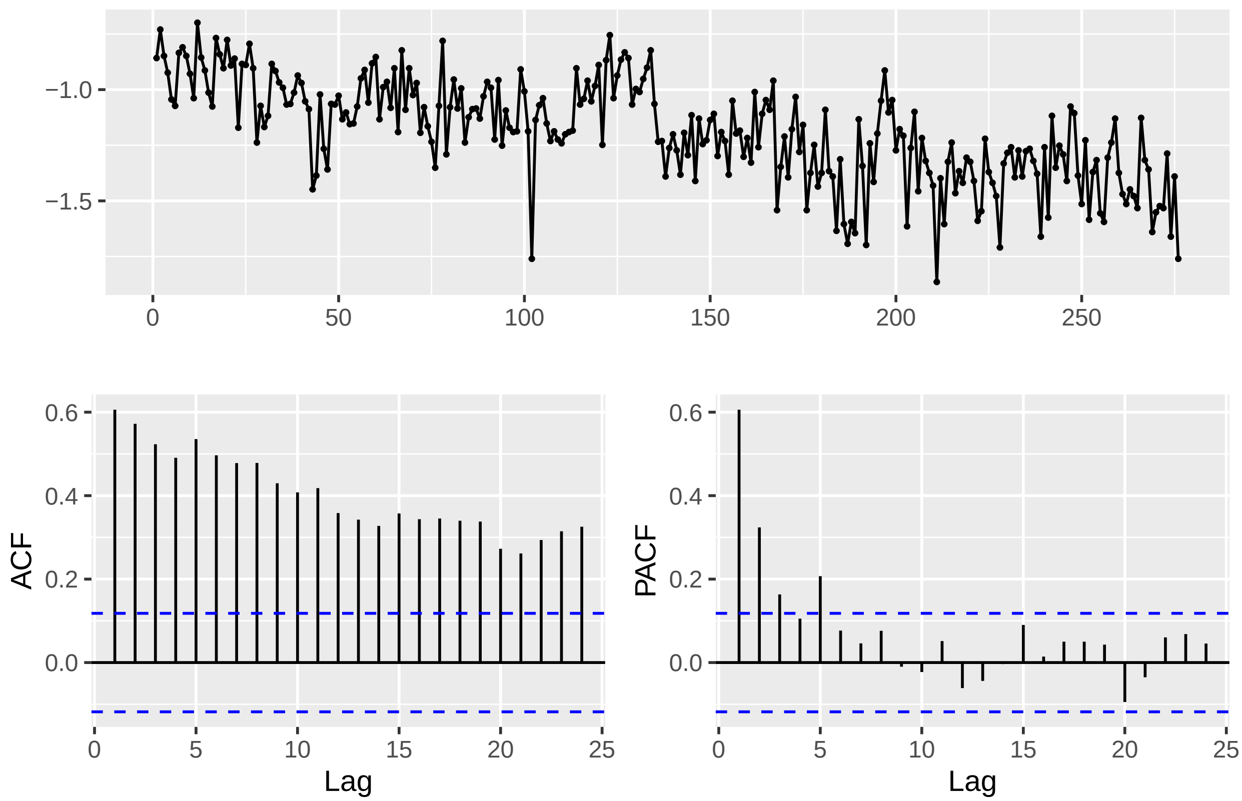
LogColgate

```r
## trabajamos con transformacion logaritmica
logcolgate <- log(colgate)
df_logcolgate <- data.frame(value = as.vector(logcolgate),
                            time = time(logcolgate))
ggplot(df_logcolgate) + geom_point(aes(x = time, y = value)) +
  geom_line(aes(x = time, y = value)) +
  ylab("log - MktShare") +
  ggtitle("Market Share Semanal- Colgate (logarítmico)") +
  xlab("Semanas") +
  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 2)
```

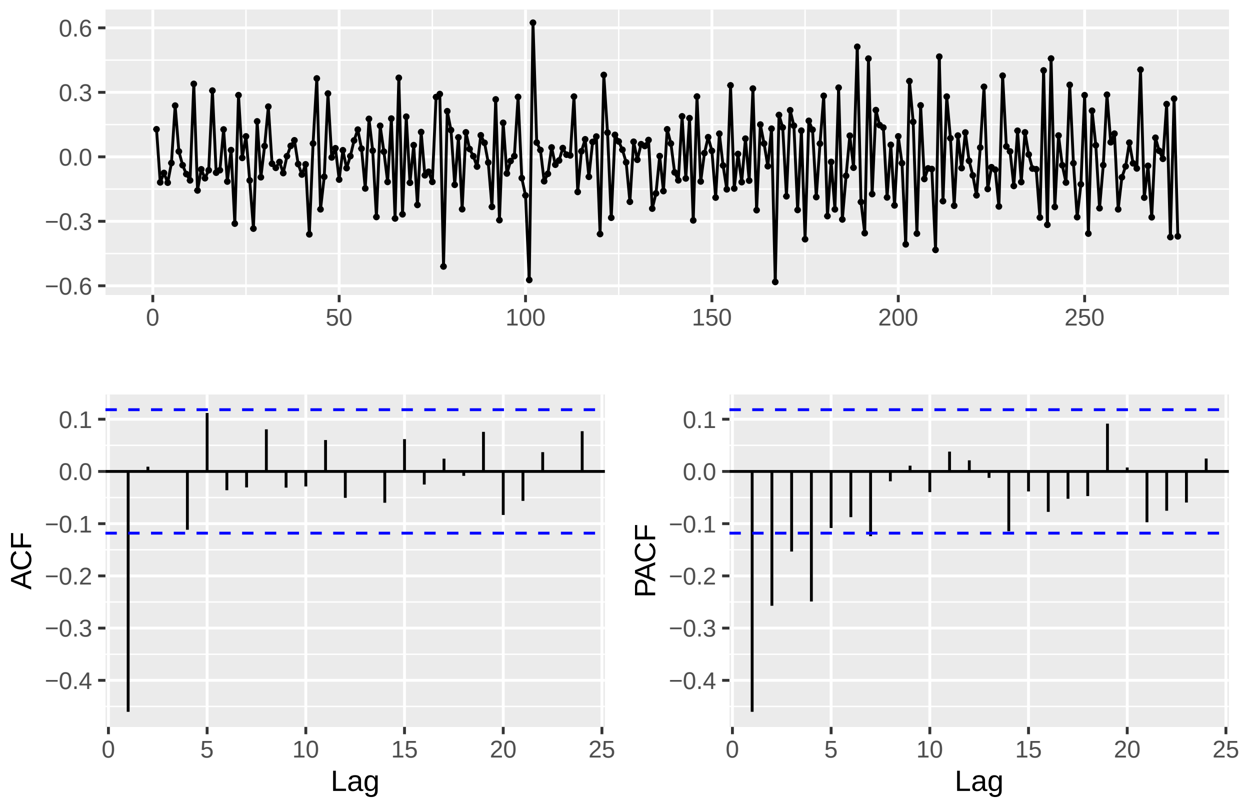


Diferenciando la serie

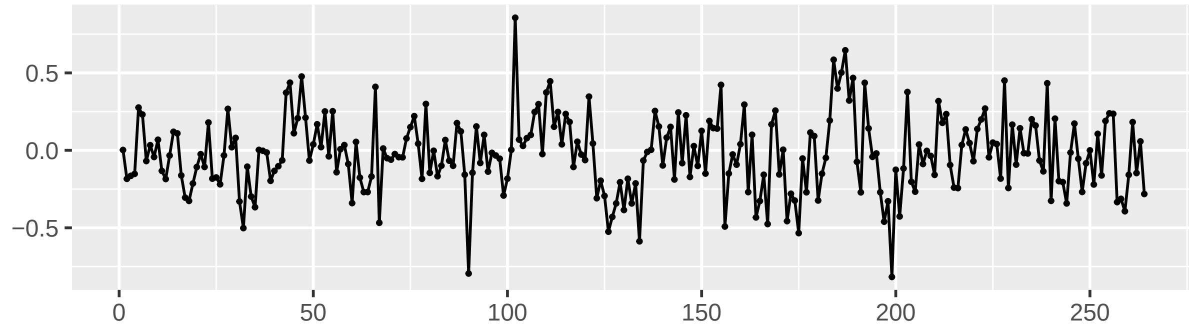
```
## Difference
ggtsdisplay(logcolgate)
```

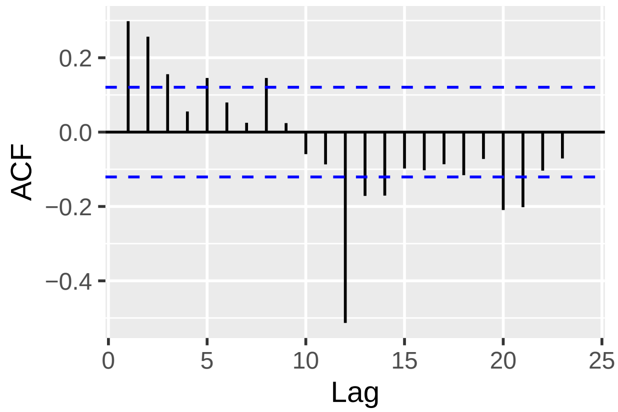


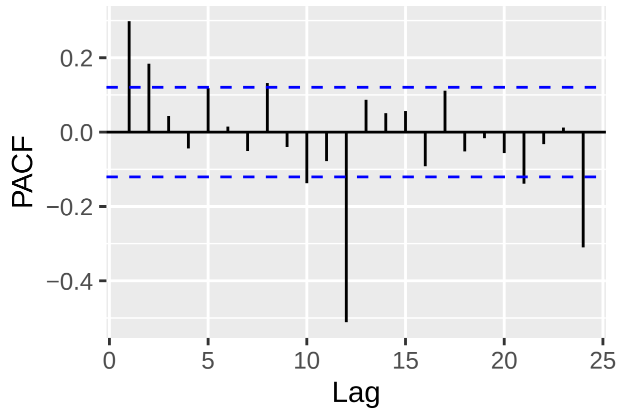

```
ggtsdisplay(diff(logcolgate))
```




```r
ggtsdisplay(diff(logcolgate, 12))
```








```
which(diff(logcolgate) == max(diff(logcolgate)))
```

```
## [1] 102
```

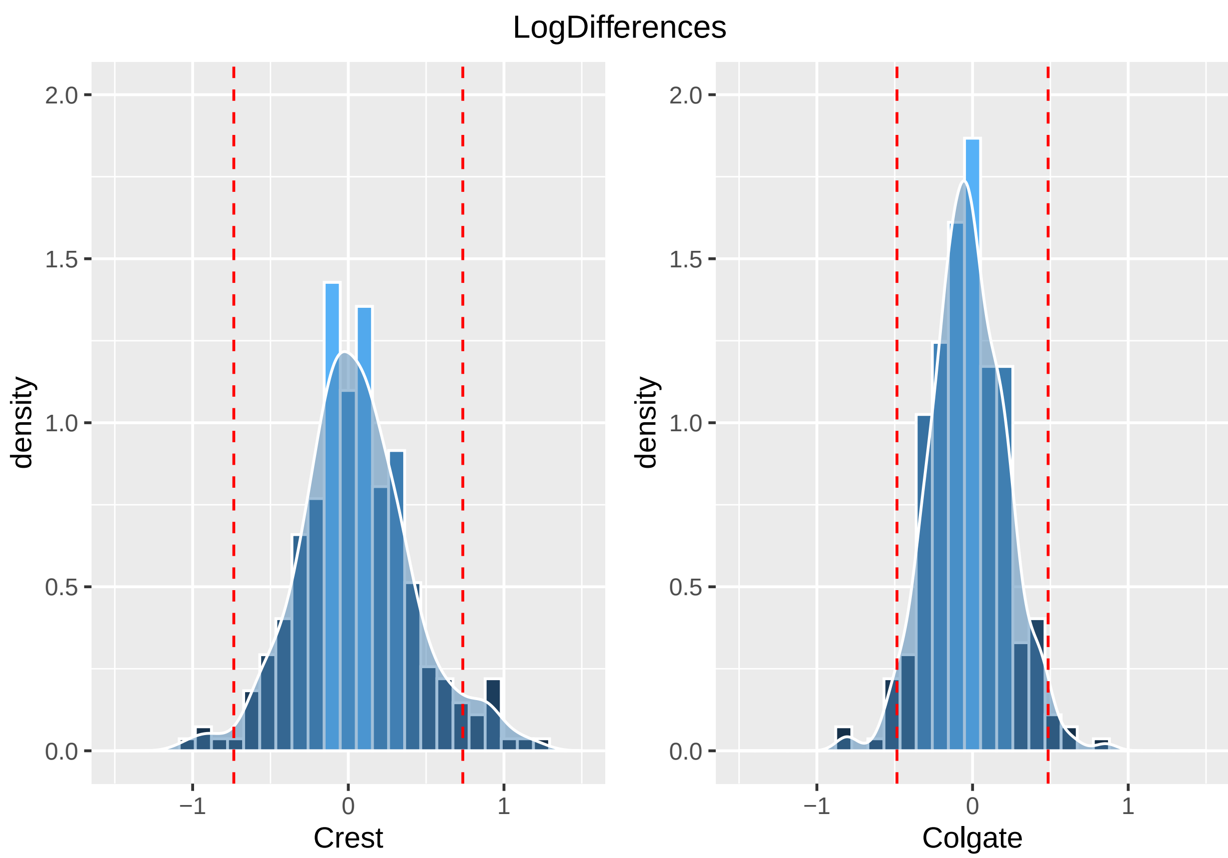
```
which(diff(logcolgate) == min(diff(logcolgate)))
```

```
## [1] 167
```

Detectando Outliers

```
#-#- Fechas Importantes #-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-


## histograms


p1 <- ggplot(data = diff(logcrest,12), aes(Crest)) +
  geom_histogram(aes(y = ..density.., fill = ..count..), color = "white") +
  xlab("Crest") +
  geom_density(fill = "steelblue", alpha = 0.5, color = "white") +
  theme(legend.position = "None")  + ylim(0,2) + xlim(-1.5,1.5) +
  geom_vline(xintercept = c(2 * sd(diff(logcrest,12)),-2 * sd(diff(logcrest,12))),
```

```r
               linetype = 2, colour = "red")
p2 <- ggplot(data = diff(logcolgate,12), aes(colgate)) +
  geom_histogram(aes(y = ..density.., fill = ..count..), color = "white") +
  xlab("Colgate") +
  geom_density(fill = "steelblue", alpha = 0.5, color = "white") +
  theme(legend.position = "None")  + ylim(0,2) + xlim(-1.5,1.5) +
  geom_vline(xintercept = c(2 * sd(diff(logcolgate,12)),-2 * sd(diff(logcolgate,12))),
               linetype = 2, colour = "red")

grid.arrange(
  p1, p2,
  widths = c( 1, 1),
  top = text_grob("LogDifferences"),
  layout_matrix = rbind(c(1, 2),
                        c(1, 2))
)
```



Outliers

```r
k = 4

which(abs(diff(logcrest,12)) > k*sd(abs(diff(logcrest,12))))
```

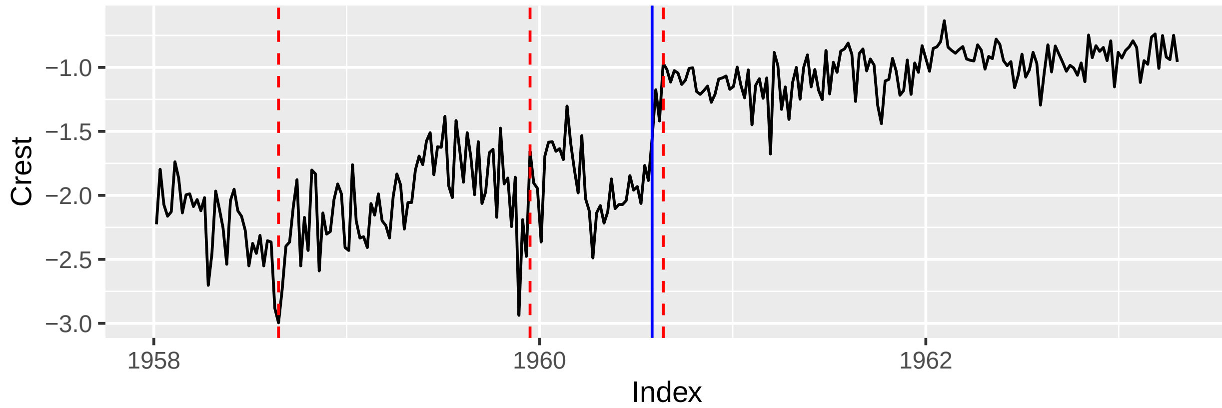
```
## [1]  22  99 126 127
```

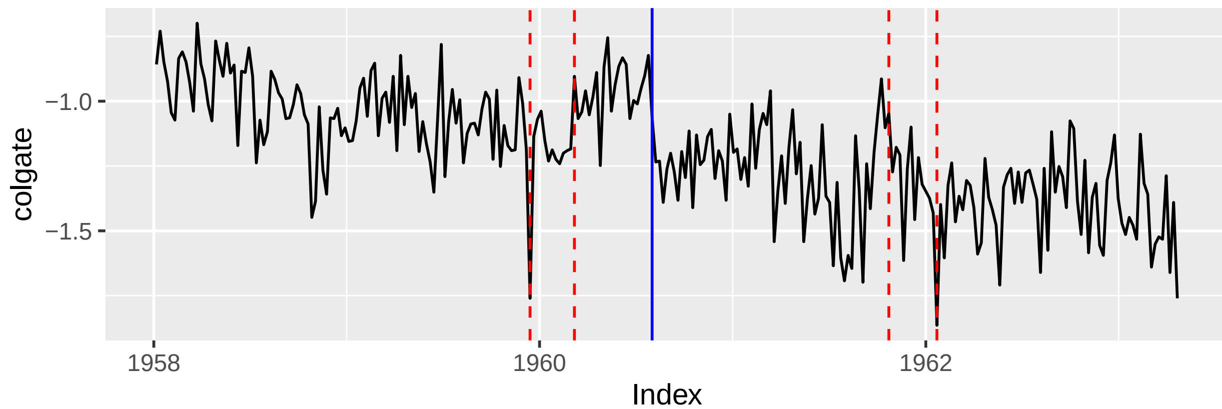
```r
which(abs(diff(logcolgate,12)) > k*sd(abs(diff(logcolgate,12))))
```

```
## [1]  90 102 187 199
```

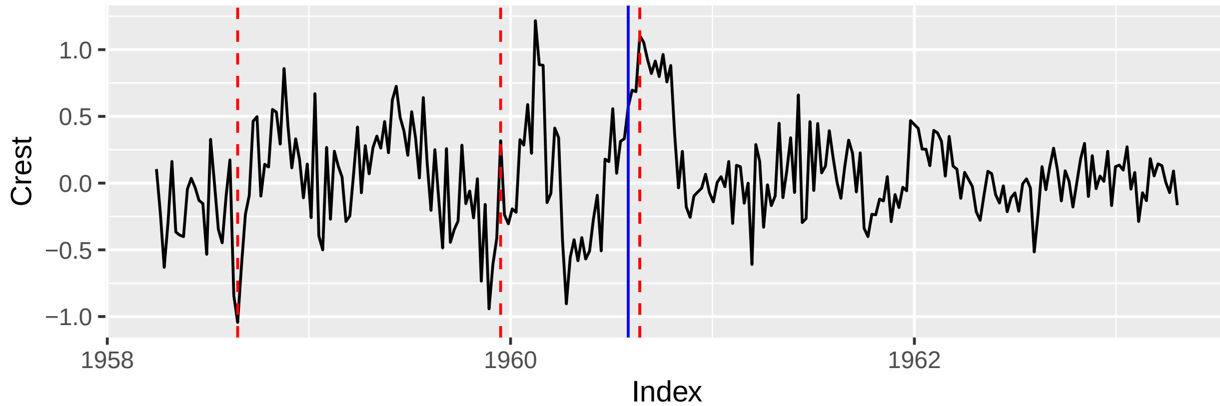
Outlier Grafocamente

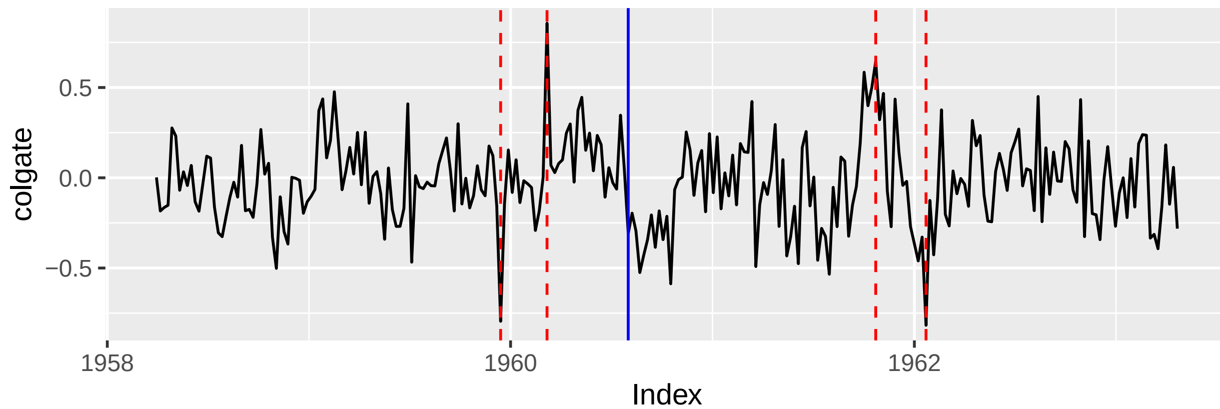
```r
## trabajando con logaritmos

p1 <- autoplot(((logcrest)))  +
  geom_vline(xintercept = as.numeric(datos$Date[126 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[126 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[90 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[22 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 1, colour = "blue")
p2 <- autoplot(((logcolgate))) +
  geom_vline(xintercept = as.numeric(datos$Date[90 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[102 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[187 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[199 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 1, colour = "blue")

grid.arrange(
  p1, p2,
  widths = c( 1, 1),
  #top = text_grob(Character),
  layout_matrix = rbind(c(1, 1),
                        c(2, 2))
)
```






```
## 12 lags (trimestre)

p1 <- autoplot((diff(logcrest,12)))  +
  geom_vline(xintercept = as.numeric(datos$Date[126 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[126 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[90 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[22 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 1, colour = "blue")
p2 <- autoplot((diff(logcolgate,12))) +
  geom_vline(xintercept = as.numeric(datos$Date[90 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[102 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[187 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[199 + 12]), linetype = 2, colour = "red") +
  geom_vline(xintercept = as.numeric(datos$Date[135]), linetype = 1, colour = "blue")

grid.arrange(
  p1, p2,
  widths = c( 1, 1),
  #top = text_grob(Character),
  layout_matrix = rbind(c(1, 1),
                        c(2, 2))
)
```



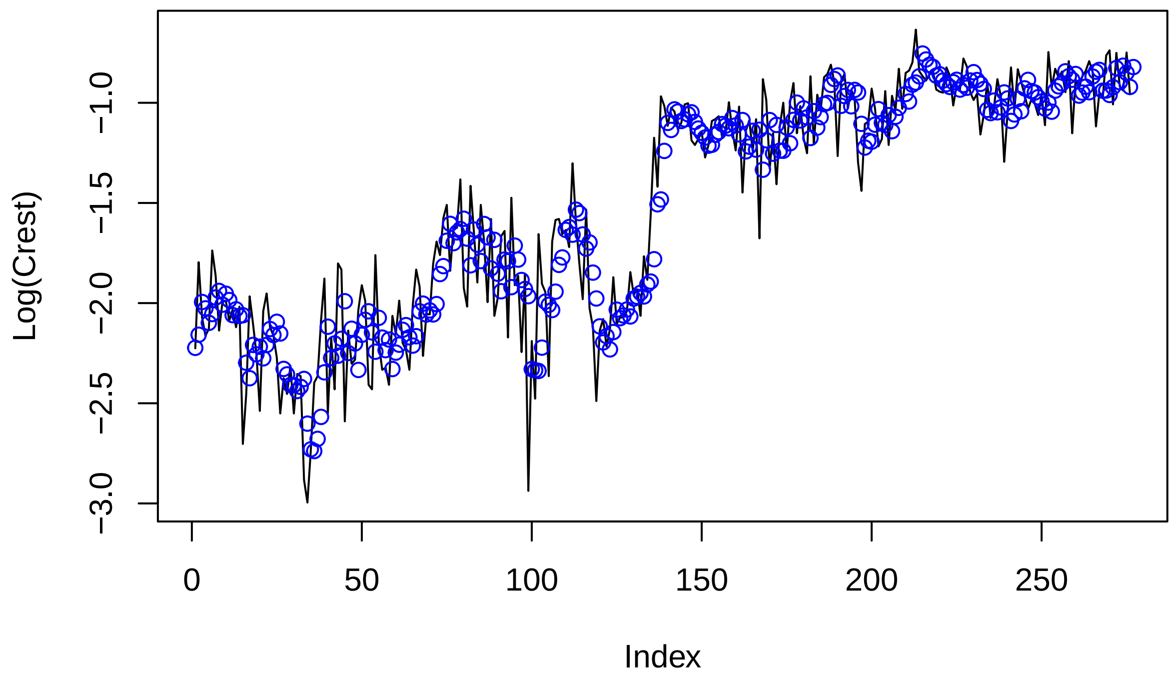


Crest ARIMA

```r
## Crest ARIMA ##########################################################################################
modCrest <- auto.arima(logcrest)
summary(modCrest)
```

```
## Series: logcrest
## ARIMA(3,1,0)
##
## Coefficients:
##           ar1      ar2      ar3
##       -0.6188  -0.3730  -0.2216
## s.e.   0.0588   0.0659   0.0588
##
## sigma^2 estimated as 0.04944:  log likelihood=23.89
## AIC=-39.78   AICc=-39.63   BIC=-25.3
##
## Training set error measures:
##                      ME      RMSE       MAE       MPE     MAPE      MASE
## Training set 0.01007853 0.2211341 0.1672474 -2.288999 11.55329 0.1103725
##                    ACF1
## Training set -0.0223424
```

```r
plot(as.double(logcrest), ylab = "Log(Crest)", type = "l")
points(as.double(fitted(modCrest)), col = "blue")
```



Training

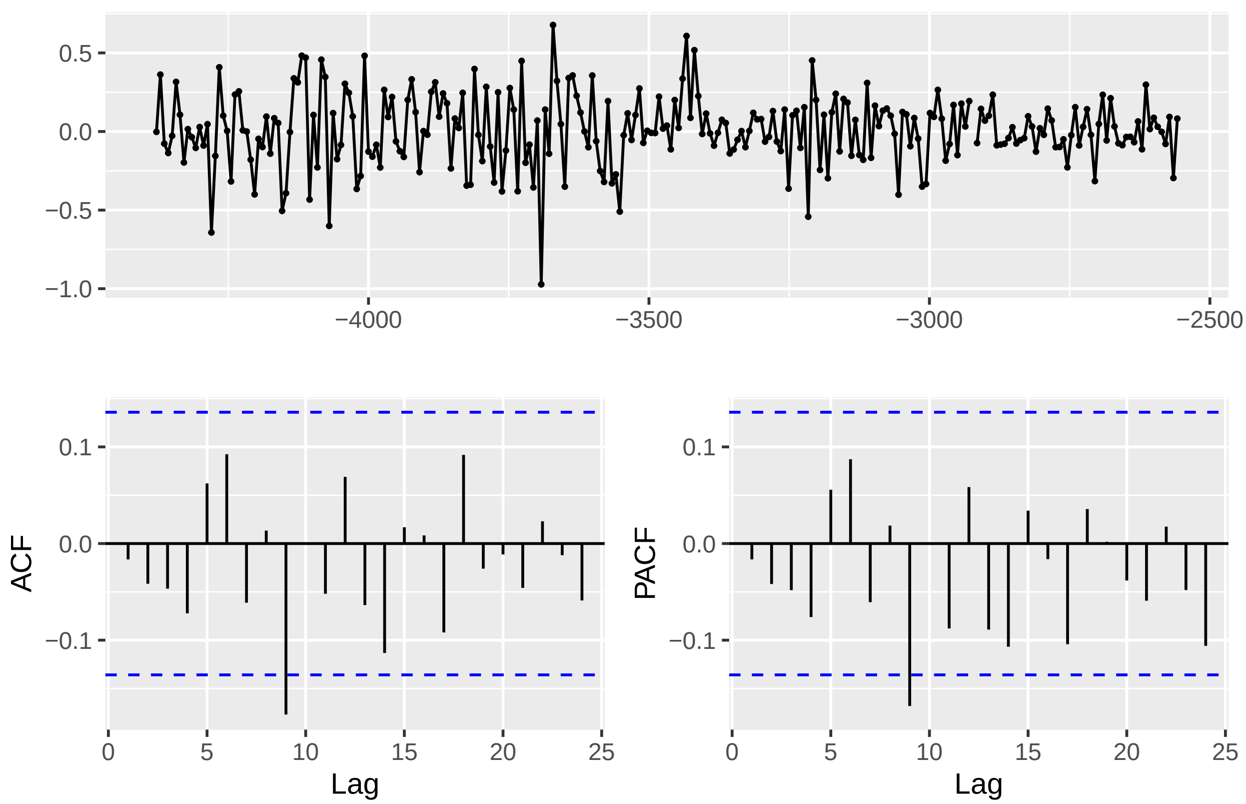
```r
#- Training set     -#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

## Select number of observation to compare forecast (16 semanas)
cOmit = 16

## Data Size
nObs = length(logcrest)

## sub_sample
oCrest <- window(logcrest,start = index(logcrest[1]),end = index(logcrest[nObs - cOmit]))

## ARIMA MODEL Automatic selection####
crest.train.arima = auto.arima(oCrest) ## lamnda cero is log transformation
summary(crest.train.arima)
```

```
## Series: oCrest
## ARIMA(3,1,0)
##
```

```
## Coefficients:
##           ar1      ar2      ar3
##       -0.6175  -0.3738  -0.2276
## s.e.   0.0605   0.0678   0.0607
##
## sigma^2 estimated as 0.05136:  log likelihood=17.59
## AIC=-27.17   AICc=-27.02   BIC=-12.93
##
## Training set error measures:
##                     ME      RMSE       MAE       MPE     MAPE     MASE
## Training set 0.0102414 0.2253215 0.1707961 -2.271995 11.48468 0.109894
##                     ACF1
## Training set -0.02012989
```

Análisis de Residuales

```r
#residual analysis
ggtsdisplay(crest.train.arima$residuals)
```




```r
#box-Ljung Test
Box.test(crest.train.arima$residuals,lag = 4, fitdf = 3, type = "Lj")
```

```
##
```

```
##  Box-Ljung test
##
## data:  crest.train.arima$residuals
## X-squared = 2.6221, df = 1, p-value = 0.1054
```

```r
Box.test(crest.train.arima$residuals,lag = 8, fitdf = 3, type = "Lj")
```

```
##
##  Box-Ljung test
##
## data:  crest.train.arima$residuals
## X-squared = 6.4423, df = 5, p-value = 0.2655
```

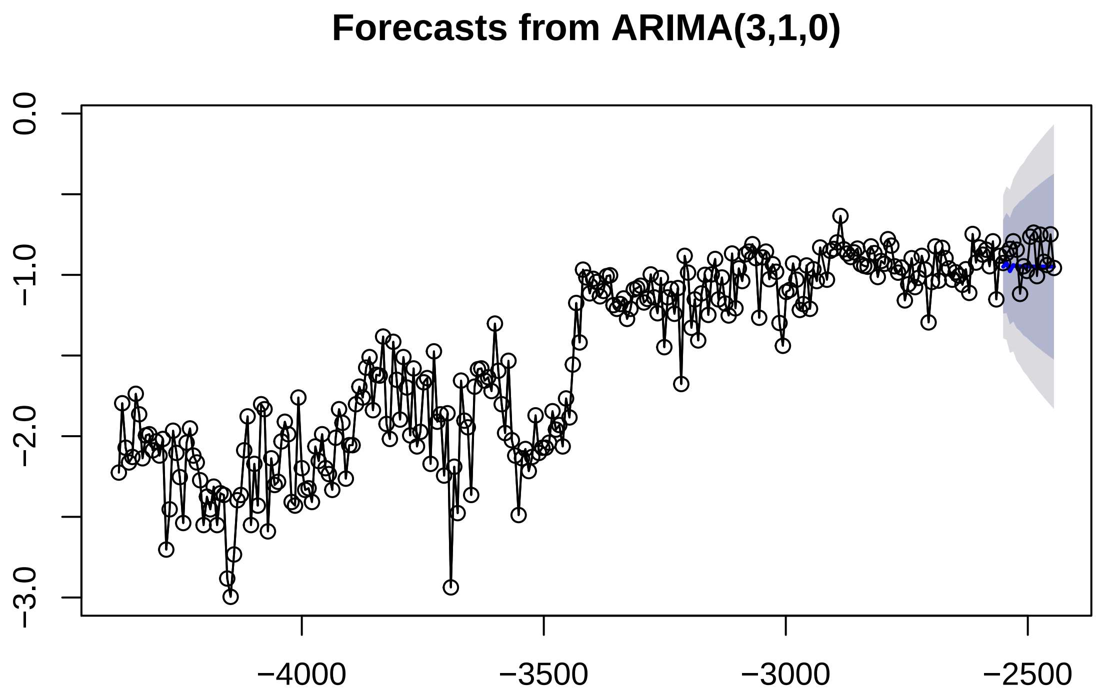
```r
Box.test(crest.train.arima$residuals,lag = 12, fitdf = 3, type = "Lj")
```

```
##
##  Box-Ljung test
##
## data:  crest.train.arima$residuals
## X-squared = 15.964, df = 9, p-value = 0.06764
```

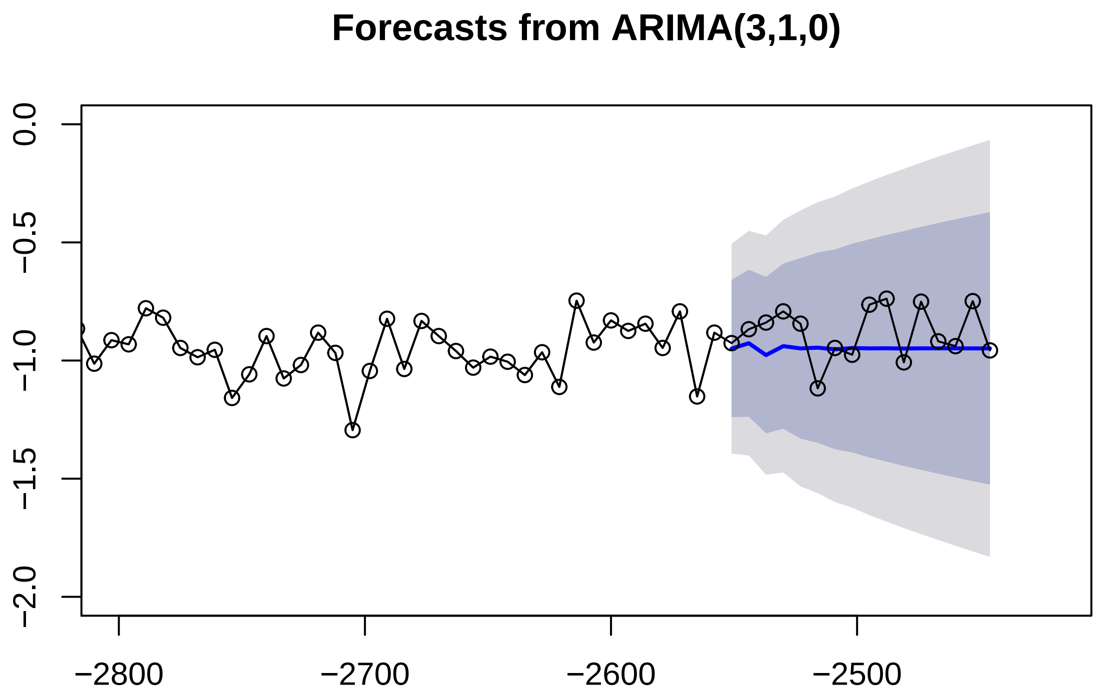
```r
## Residuales independientes
```

Forecast

```r
plot(forecast(crest.train.arima, h = 16))
lines(window(logcrest),type = "o")
```




```
plot(forecast(crest.train.arima, h = 16), xlim = c(-2800, -2420), ylim = c(-2,0))
lines(window(logcrest),type = "o")
```




```
fcrest_arima <- forecast(crest.train.arima, h = 16) ## predecimos 16 semanas
```

Métricas de Predición

```
crestArimaMatrix <- matrix(c(fcrest_arima$mean[1:16], as.double(tail(logcrest,16))), ncol = 2)
crestArimaMatrix
```

```
##             [,1]       [,2]
##  [1,] -0.9493279 -0.9263411
##  [2,] -0.9266708 -0.8675006
##  [3,] -0.9769404 -0.8393297
##  [4,] -0.9390185 -0.7918632
##  [5,] -0.9488025 -0.8439701
##  [6,] -0.9454925 -1.1177951
##  [7,] -0.9525113 -0.9467499
##  [8,] -0.9471875 -0.9755101
##  [9,] -0.9486049 -0.7635696
## [10,] -0.9481219 -0.7381445
## [11,] -0.9491022 -1.0078579
## [12,] -0.9483548 -0.7507763
## [13,] -0.9485598 -0.9187939
## [14,] -0.9484894 -0.9390477
## [15,] -0.9486264 -0.7486599
## [16,] -0.9485215 -0.9571127
```

```r
## MSE
mean((crestArimaMatrix[,1] - crestArimaMatrix[,2])^2)
```

```
## [1] 0.01549924
```

```r
## MAE
mean(abs(crestArimaMatrix[,1] - crestArimaMatrix[,2]))
```

```
## [1] 0.09857839
```

```r
## Bias
mean(crestArimaMatrix[,1] - crestArimaMatrix[,2])
```

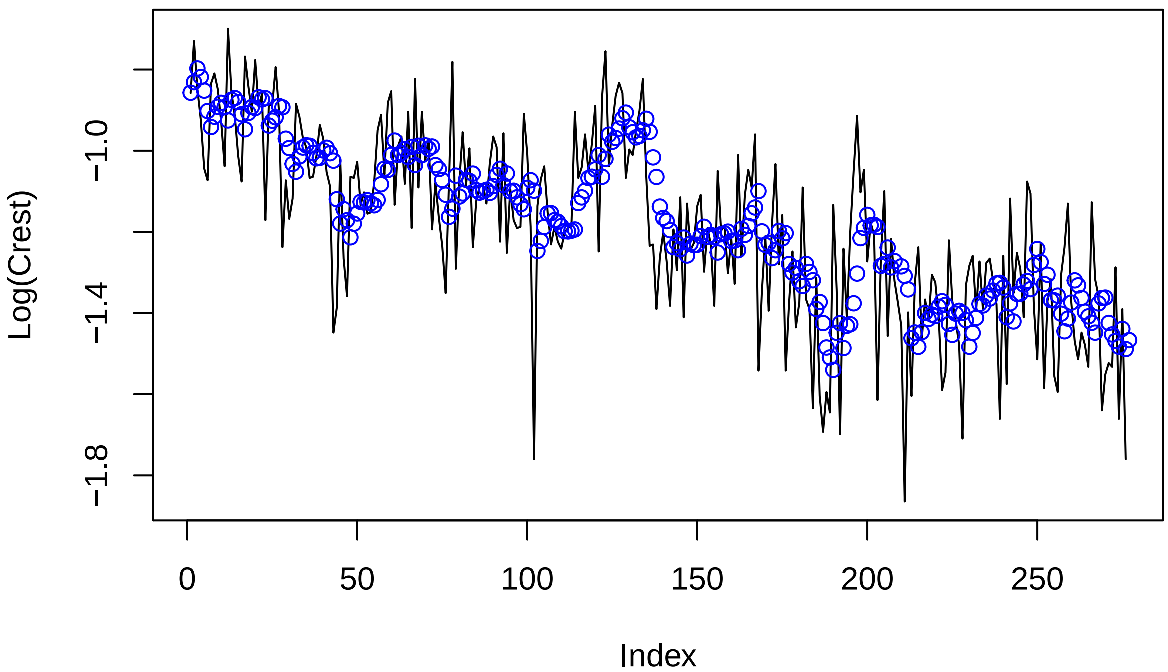
```
## [1] -0.06508188
```

Colgate ARIMA

```r
## Colgate ARIMA ###################################################################################
modColgate <- auto.arima(logcolgate)
summary(modColgate)
```

```
## Series: logcolgate
## ARIMA(0,1,1)
##
## Coefficients:
##           ma1
##       -0.7756
## s.e.   0.0451
##
## sigma^2 estimated as 0.02574:  log likelihood=112.47
## AIC=-220.95   AICc=-220.9   BIC=-213.71
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set -0.01086176 0.1601509 0.1244834 -0.6296119 10.53961 0.1048155
##                    ACF1
## Training set 0.04232292
```

```r
plot(as.double(logcolgate), ylab = "Log(Crest)", type = "l")
points(as.double(fitted(modColgate)), col = "blue")
```



Training

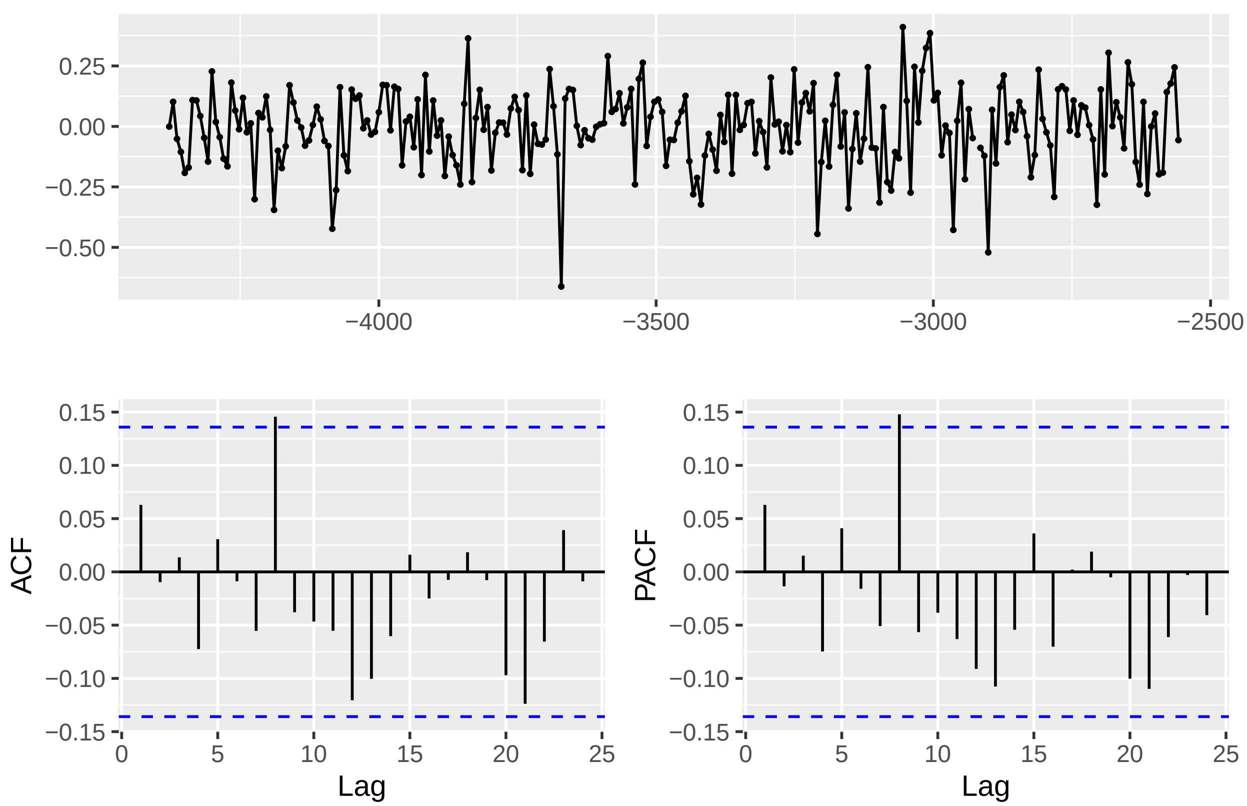
```
#- Training set     -#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

## Select number of observation to compare forecast (16 semanas)
cOmit = 16

## Data Size
nObs = length(logcolgate)

## sub_sample
ocolgate <- window(logcolgate,start = index(logcolgate[1]),end = index(logcolgate[nObs - cOmit]))

## ARIMA MODEL Automatic selection####
colgate.train.arima = auto.arima(ocolgate) ## lamnda cero is log transformation
summary(colgate.train.arima)
```

```
## Series: ocolgate
## ARIMA(0,1,1)
##
## Coefficients:
##           ma1
##       -0.7691
## s.e.   0.0477
```

```
##
## sigma^2 estimated as 0.02558:  log likelihood=106.73
## AIC=-209.45   AICc=-209.4   BIC=-202.33
##
## Training set error measures:
##                        ME      RMSE       MAE        MPE     MAPE
## Training set -0.007851298 0.1596299 0.1234751 -0.8500588 10.59601
##                   MASE      ACF1
## Training set 0.1055347 0.0452135
```

Análisis de Residuales

```r
#residual analysis
ggtsdisplay(colgate.train.arima$residuals)
```




```r
#box-Ljung Test
Box.test(colgate.train.arima$residuals,lag = 4, fitdf = 3, type = "Lj")
```

```
##
##  Box-Ljung test
##
## data:  colgate.train.arima$residuals
## X-squared = 2.5333, df = 1, p-value = 0.1115
```

```r
Box.test(colgate.train.arima$residuals,lag = 8, fitdf = 3, type = "Lj")
```

```
##
##  Box-Ljung test
##
## data:  colgate.train.arima$residuals
## X-squared = 4.836, df = 5, p-value = 0.4362
```

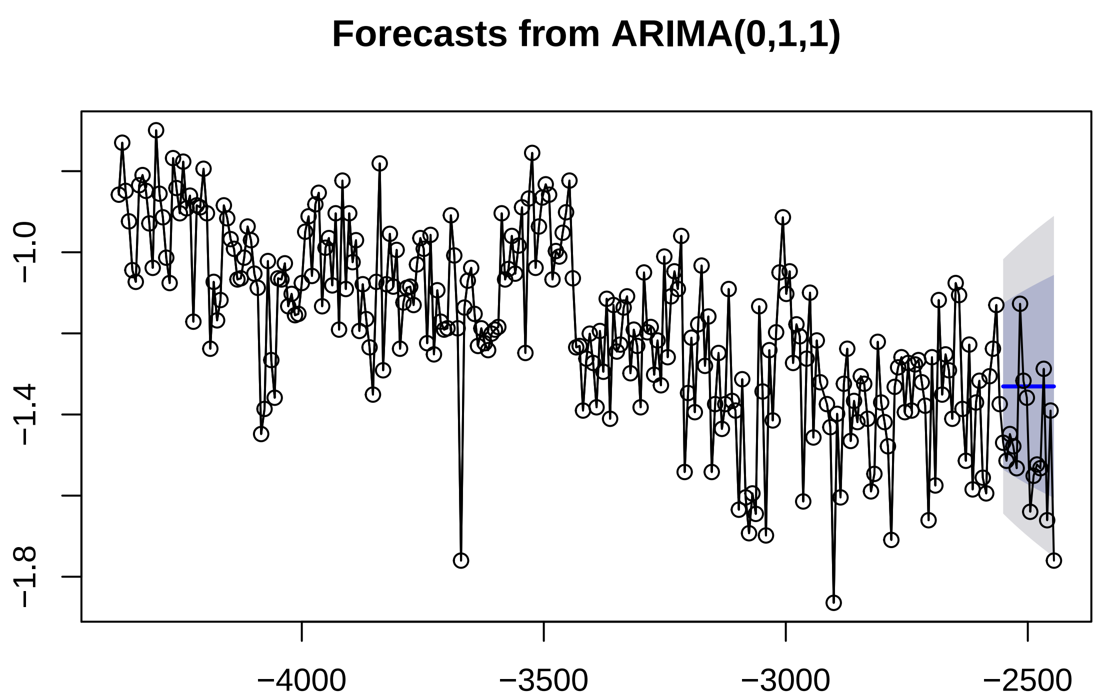
```r
Box.test(colgate.train.arima$residuals,lag = 12, fitdf = 3, type = "Lj")
```

```
##
##  Box-Ljung test
##
## data:  colgate.train.arima$residuals
## X-squared = 7.8228, df = 9, p-value = 0.5521
```

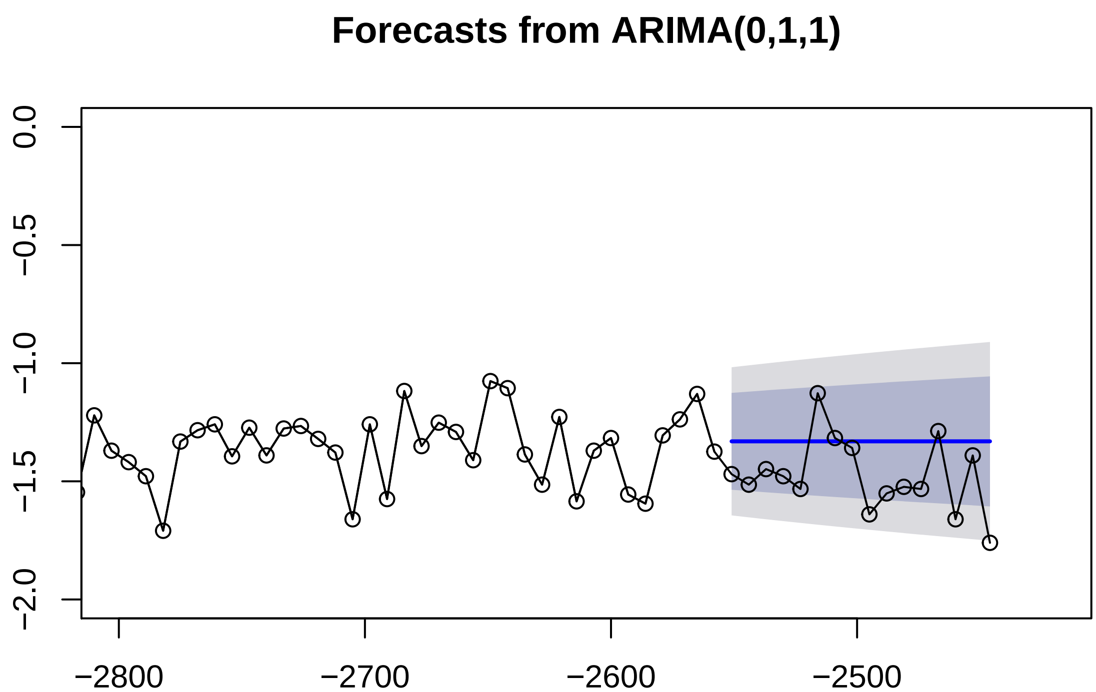
```r
## Residuales independientes
```

Forecast

```r
plot(forecast(colgate.train.arima, h = 16))
lines(window(logcolgate),type = "o")
```



```r
plot(forecast(colgate.train.arima, h = 16), xlim = c(-2800, -2420), ylim = c(-2,0))
lines(window(logcolgate),type = "o")
```




```r
fcolgate_arima <- forecast(colgate.train.arima, h = 16) ## predecimos 16 semanas
```

Métricas de Predición

```r
colgateArimaMatrix <- matrix(c(fcolgate_arima$mean[1:16], as.double(tail(logcolgate,16))), ncol = 2)
colgateArimaMatrix
```

```
##            [,1]      [,2]
##  [1,] -1.330771 -1.469676
##  [2,] -1.330771 -1.514128
##  [3,] -1.330771 -1.448170
##  [4,] -1.330771 -1.478410
##  [5,] -1.330771 -1.532477
##  [6,] -1.330771 -1.127012
##  [7,] -1.330771 -1.316768
##  [8,] -1.330771 -1.358679
##  [9,] -1.330771 -1.639897
## [10,] -1.330771 -1.551169
## [11,] -1.330771 -1.523260
```

```
## [12,] -1.330771 -1.532477
## [13,] -1.330771 -1.287354
## [14,] -1.330771 -1.660731
## [15,] -1.330771 -1.390302
## [16,] -1.330771 -1.760261
```

```
## MSE
mean((colgateArimaMatrix[,1] - colgateArimaMatrix[,2])^2)
```

```
## [1] 0.04326913
```

```
## MAE
mean(abs(colgateArimaMatrix[,1] - colgateArimaMatrix[,2]))
```

```
## [1] 0.1762993
```

```
## Bias
mean(colgateArimaMatrix[,1] - colgateArimaMatrix[,2])
```

```
## [1] 0.1436519
```

Outliers automáticos

```
## Outliers ##############################################################################################
##

## Crest
detectAO(modCrest)
```

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

```
detectIO(modCrest)
```

```
##              [,1] [,2]
## ind     99.000000   NA
## lambda1 -4.636878   NA
```

```
## Colgate
detectAO(modColgate)
```

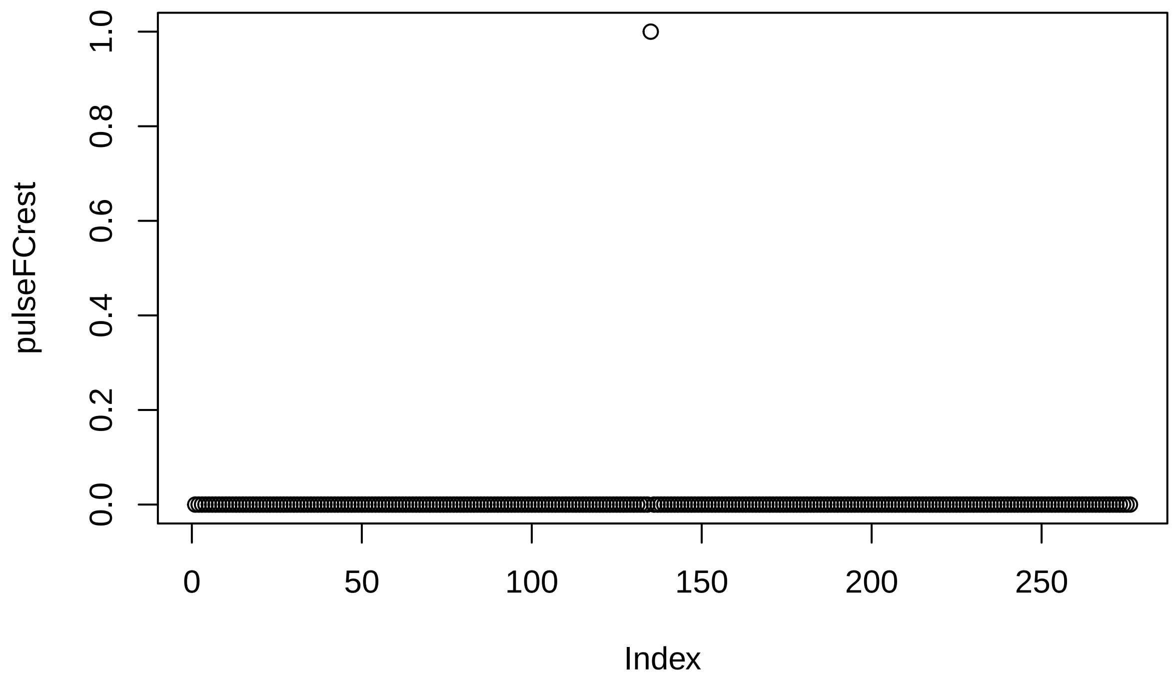
```
## [1] "No AO detected"
```

```
detectIO(modColgate)
```

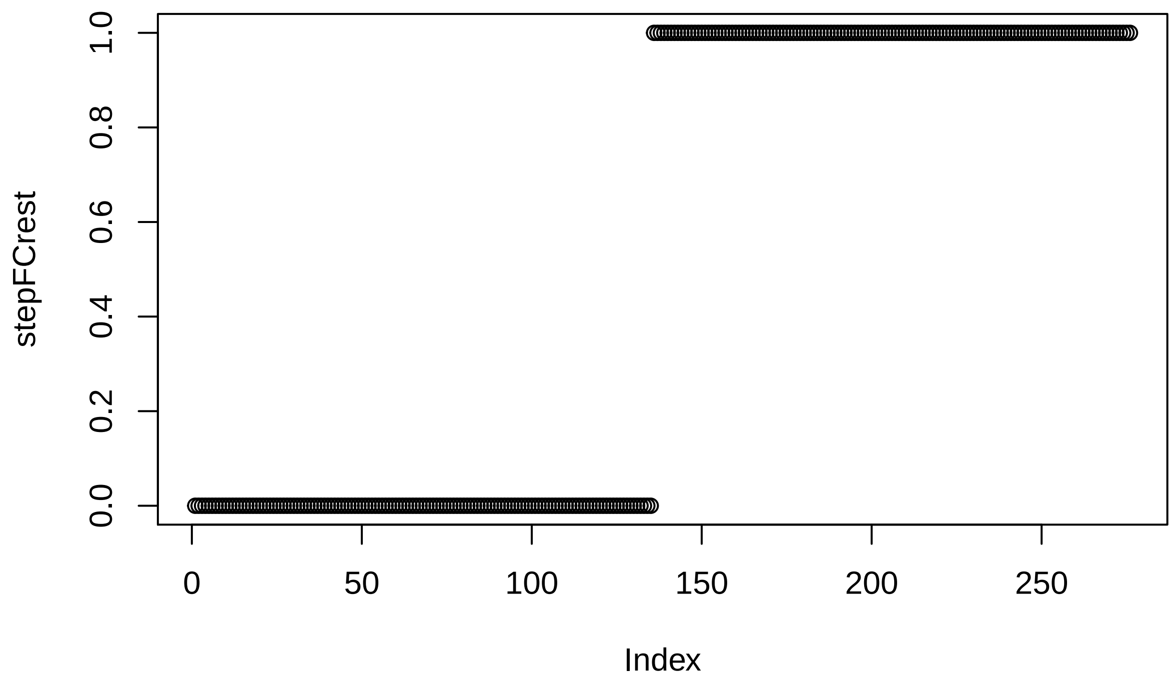
```
##               [,1] [,2]
## ind     102.000000   NA
## lambda1  -4.241699   NA
```

Crest ARIMAX

```r
## Crest ARIMAX ###########################################################################################

pulseFCrest <- data.frame(ADA = 1*(seq(logcrest) == which(datos$Date == "1960-08-01")))[,1]
plot(pulseFCrest)
```



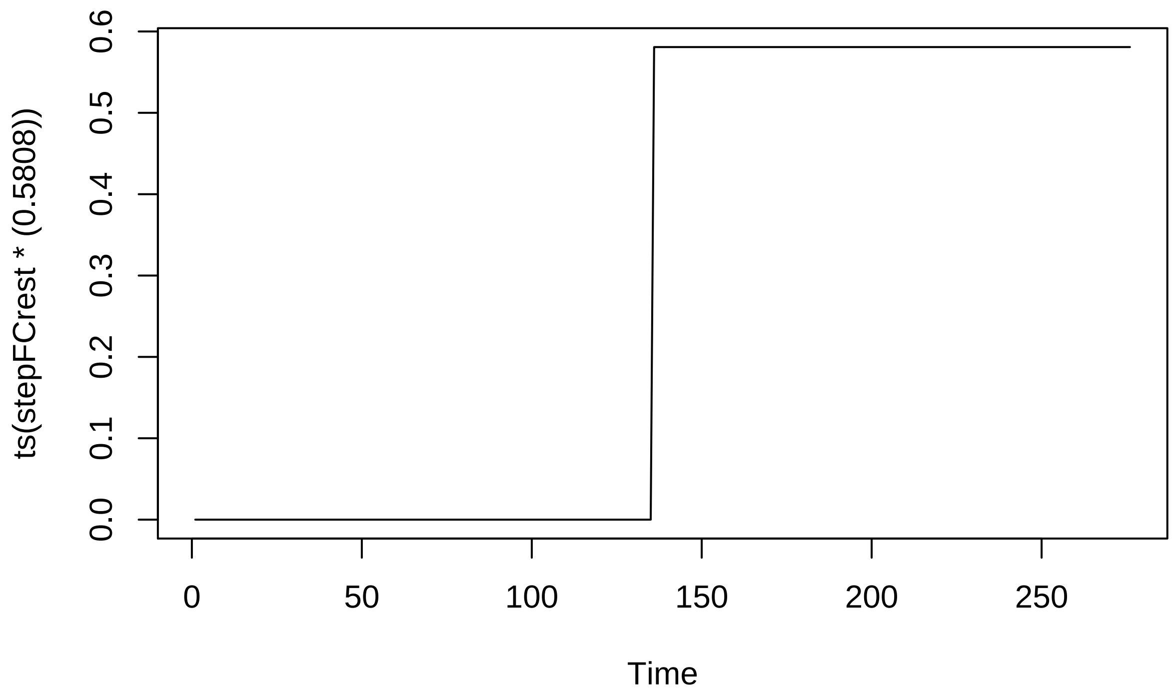

```r
stepFCrest <- data.frame(ADA = 1*(seq(logcolgate) > which(datos$Date == "1960-08-01")))[,1]
plot(stepFCrest)
```



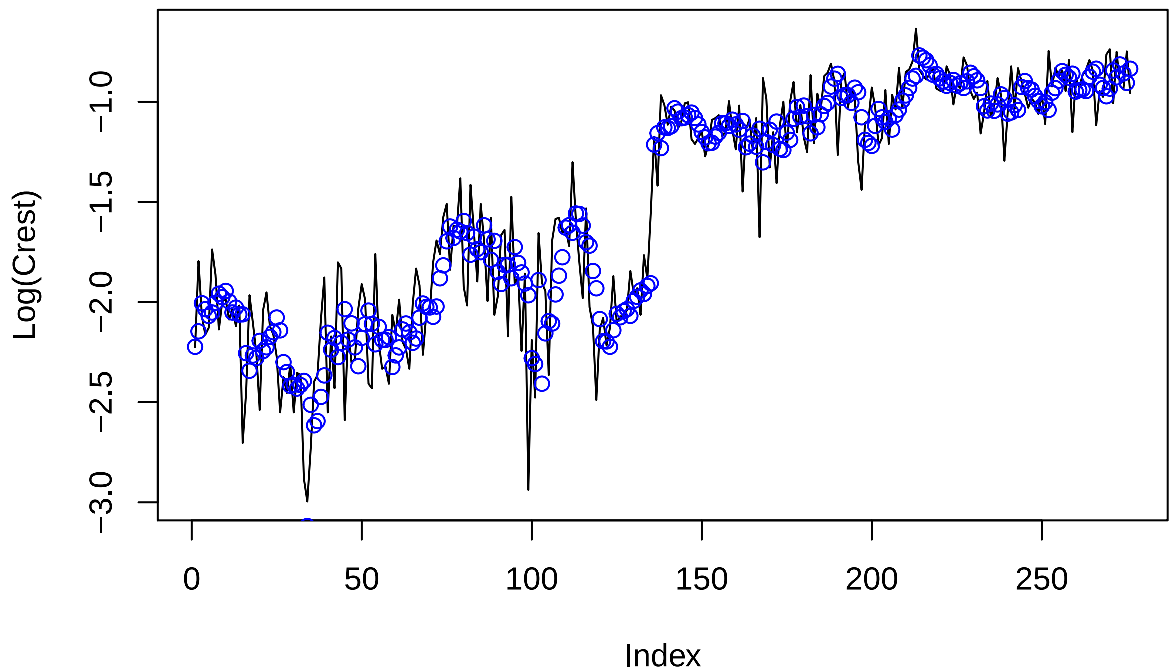

```r
df_crest <- data.frame(pulseFCrest,stepFCrest)

## Solo step
crest.m1 = arimax(as.double(logcrest$Crest),
                  order = c(3,1,0), method = 'ML',
                  xtransf = data.frame(ADA = stepFCrest),
                  transfer = list(c(0,0)),
                  xreg = data.frame(Imp1 = 1*(seq(logcolgate) == (22 + 12)),
                                    Imp2 = 1*(seq(logcolgate) == (90 + 12)))
                  )
crest.m1
```

```
##
## Call:
## arimax(x = as.double(logcrest$Crest), order = c(3, 1, 0), xreg = data.frame(Imp1 = 1 *
##     (seq(logcolgate) == (22 + 12)), Imp2 = 1 * (seq(logcolgate) == (90 + 12))),
##     method = "ML", xtransf = data.frame(ADA = stepFCrest), transfer = list(c(0,
##         0)))
##
## Coefficients:
##           ar1      ar2      ar3     Imp1    Imp2  ADA-MA0
##       -0.6925  -0.4457  -0.2157  -0.5565  0.4882   0.5808
## s.e.   0.0611   0.0680   0.0633   0.1960  0.2095   0.1626
##
## sigma^2 estimated as 0.04485:  log likelihood = 36.36,  aic = -60.72
```

```r
plot(ts(stepFCrest*(0.5808)))
```




```r
plot(as.double(logcrest), ylab = "Log(Crest)", type = "l")
points(fitted(crest.m1), col = "blue")
```



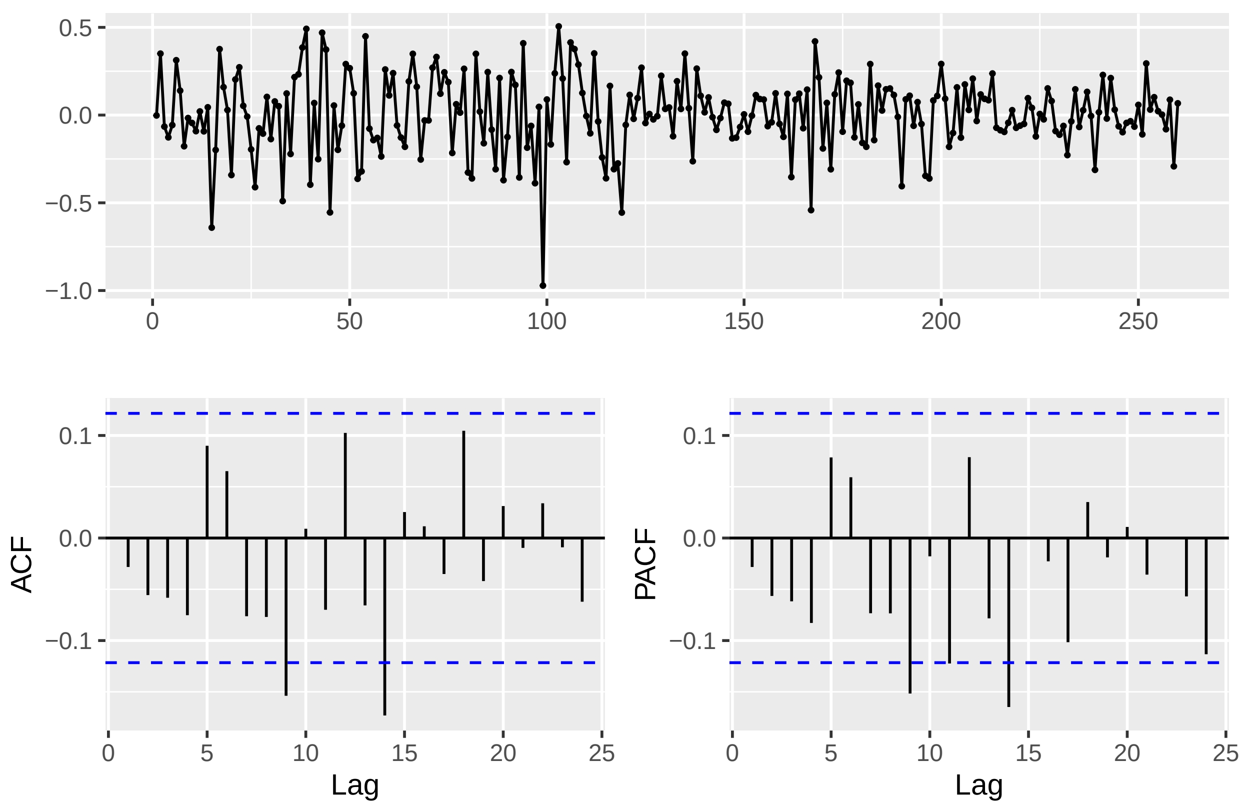
Training

```r
#- Training set     -#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

## Select number of observation to compare forecast (16 semanas)
cOmit = 16

## Data Size
nObs = length(logcrest)

## sub_sample
oCrest <- window(logcrest,start = index(logcrest[1]),end = index(logcrest[nObs - cOmit]))

stepFCrestARIMAX <- data.frame(ADA = 1*(seq(oCrest) > which(datos$Date == "1960-08-01")))[,1]

## ARIMAX MODEL CREST
crest.train.arimax = arimax(as.double(oCrest$Crest),
                            order = c(3,1,0), method = 'ML',
                            xtransf = data.frame(ADA = stepFCrestARIMAX),
                            transfer = list(c(0,0)),
                            xreg = data.frame(Imp1 = 1*(seq(oCrest) == (22 + 12)),
                                              Imp2 = 1*(seq(oCrest) == (90 + 12)))
)
summary(crest.train.arimax)
```

```
##
## Call:
## arimax(x = as.double(oCrest$Crest), order = c(3, 1, 0), xreg = data.frame(Imp1 = 1 *
##     (seq(oCrest) == (22 + 12)), Imp2 = 1 * (seq(oCrest) == (90 + 12))), method = "ML",
##     xtransf = data.frame(ADA = stepFCrestARIMAX), transfer = list(c(0, 0)))
##
## Coefficients:
##           ar1      ar2      ar3     Imp1    Imp2  ADA-MA0
##       -0.6928  -0.4484  -0.2227  -0.5560  0.4793   0.5829
## s.e.   0.0629   0.0700   0.0654   0.1997  0.2141   0.1653
##
## sigma^2 estimated as 0.0465:  log likelihood = 29.54,  aic = -47.08
##
## Training set error measures:
##                         ME      RMSE       MAE       MPE     MAPE      MASE
## Training set 0.005501084 0.2152195 0.1648899 -1.829959 10.97426 0.8247673
##                     ACF1
## Training set -0.02822744
```

Análisis Residuales

```r
#residual analysis
ggtsdisplay(crest.train.arimax$residuals)
```



```r
#box-Ljung Test
Box.test(crest.train.arimax$residuals,lag = 4, fitdf = 3, type = "Lj")
```

```
##
##  Box-Ljung test
##
## data:  crest.train.arimax$residuals
## X-squared = 3.4308, df = 1, p-value = 0.06399
```

```r
Box.test(crest.train.arimax$residuals,lag = 8, fitdf = 3, type = "Lj")
```

```
##
##  Box-Ljung test
##
## data:  crest.train.arimax$residuals
## X-squared = 9.8993, df = 5, p-value = 0.07814
```

```r
Box.test(crest.train.arimax$residuals,lag = 12, fitdf = 3, type = "Lj")
```

```
##
##  Box-Ljung test
##
## data:  crest.train.arimax$residuals
## X-squared = 20.568, df = 9, p-value = 0.01471
```

```r
## Residuales independientes
```

Predicción

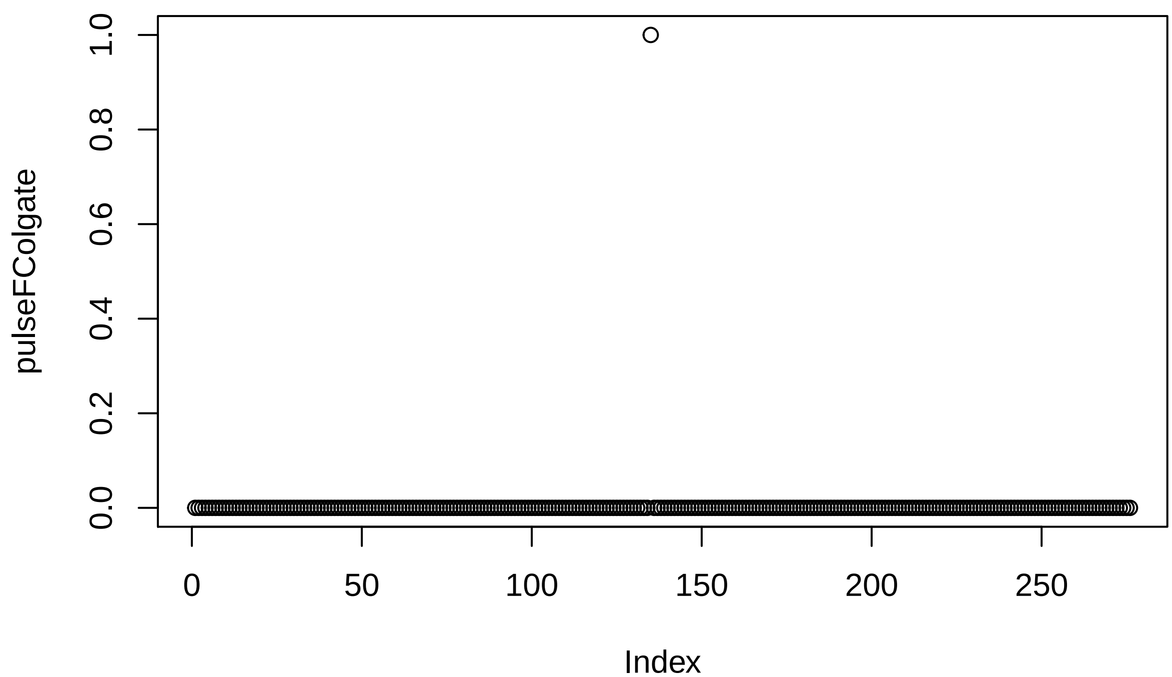
```r
#plot(forecast(crest.train.arimax, h = 16, newxreg = c(0,0)))
#lines(window(logcrest),type = "o")

#plot(forecast(crest.train.arimax, h = 16), xlim = c(-2800, -2420), ylim = c(-2,0))
#lines(window(logcrest),type = "o")

#fcrest_arimax <- forecast(crest.train.arimax, h = 16) ## predecimos 16 semanas
```

Métricas de Predicción

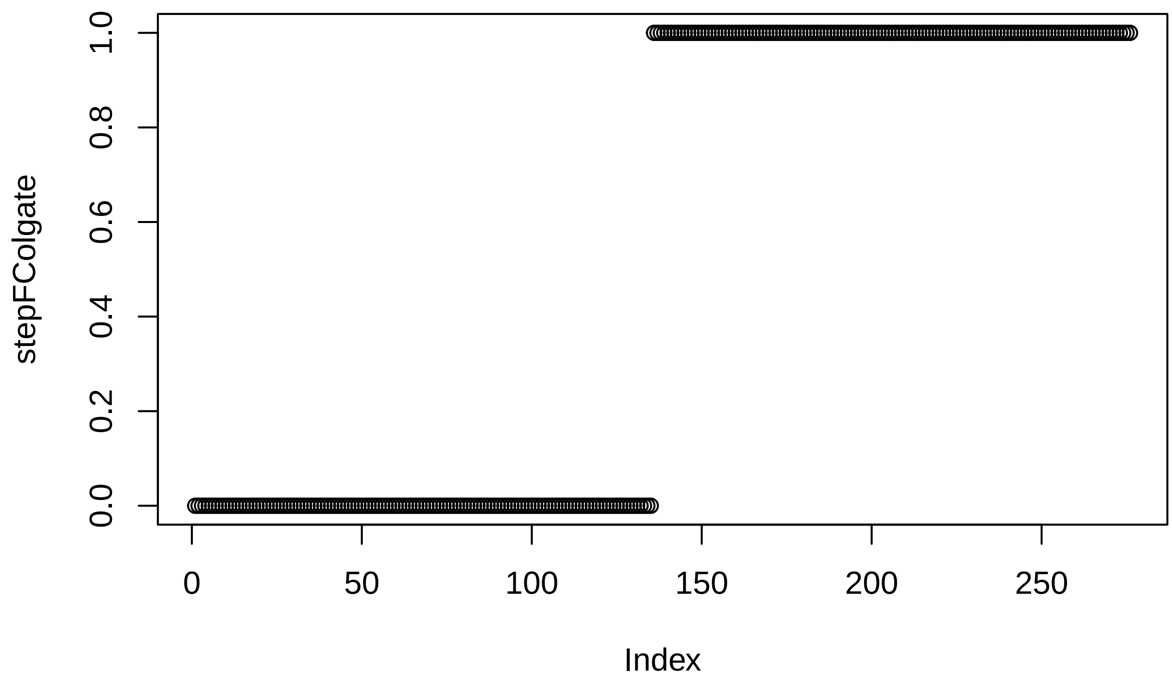
```r
#crestArimaxMatrix <- matrix(c(fcrest_arimax$mean[1:16], as.double(tail(logcrest,16))), ncol = 2)
#crestArimaxMatrix
## MSE
#mean((crestArimaxMatrix[,1] - crestArimaxMatrix[,2])^2)
## MAE
#mean(abs(crestArimaxMatrix[,1] - crestArimaxMatrix[,2]))
## Bias
#mean(crestArimaxMatrix[,1] - crestArimaxMatrix[,2])
```

Crest ARIMAX

```
## Colgate ARIMAX ######################################################################################

pulseFColgate <- data.frame(ADA = 1*(seq(logcolgate) == which(datos$Date == "1960-08-01")))[,1]
plot(pulseFColgate)
```




```
stepFColgate <- data.frame(ADA = 1*(seq(logcolgate) > which(datos$Date == "1960-08-01")))[,1]
plot(stepFColgate)
```




```r
dfcolgate <- data.frame(pulseFColgate,stepFColgate)

colgate.m1 = arimax(as.double(colgate$colgate),
                  order = c(0,1,1), method = 'ML',
                  xtransf = dfcolgate,
                  transfer = list(c(2,0),c(0,0)))
colgate.m1
```

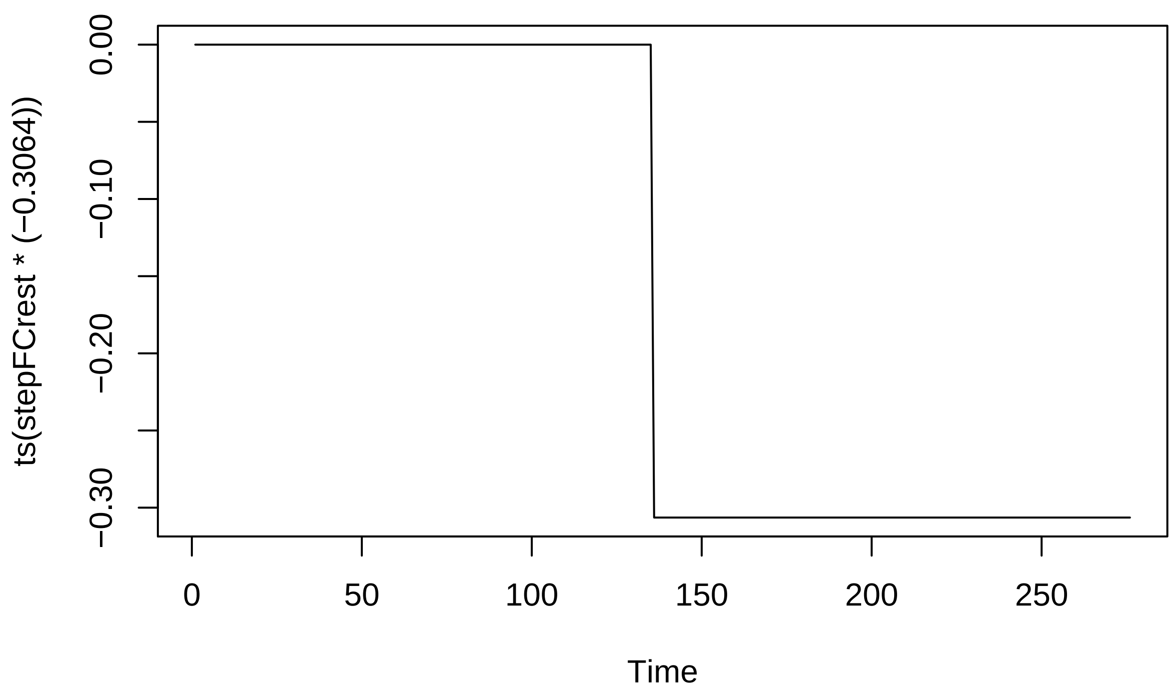
```
##
## Call:
## arimax(x = as.double(colgate$colgate), order = c(0, 1, 1), method = "ML", xtransf = dfcolgate,
##     transfer = list(c(2, 0), c(0, 0)))
##
## Coefficients:
##           ma1  pulseFColgate-AR1  pulseFColgate-AR2  pulseFColgate-MA0
##       -0.8049            -0.4341            -0.7455            -0.0439
## s.e.   0.0414             0.2999             0.3262             0.0396
##       stepFColgate-MA0
##                -0.1166
## s.e.            0.0301
##
## sigma^2 estimated as 0.002144:  log likelihood = 454.19,  aic = -898.39
```

```r
## Solo step
colgate.m1 = arimax(as.double(logcolgate$colgate),
```

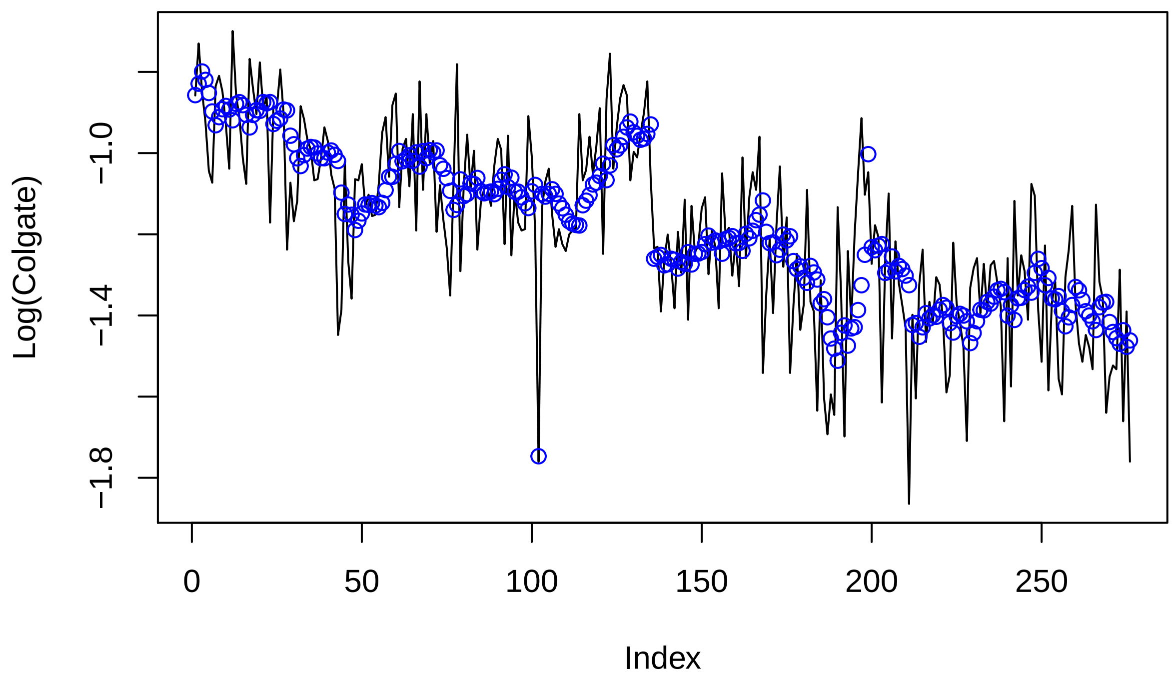
```r
                    order = c(0,1,1), method = 'ML',
                    xtransf = data.frame(ADA = stepFColgate),
                    transfer = list(c(0,0)),
                    xreg = data.frame(Imp1 = 1*(seq(logcolgate) == (90 + 12)),
                                      Imp2 = 1*(seq(logcolgate) == (187 + 12)))
                    )
colgate.m1
```

```
##
## Call:
## arimax(x = as.double(logcolgate$colgate), order = c(0, 1, 1), xreg = data.frame(Imp1 = 1 *
##     (seq(logcolgate) == (90 + 12)), Imp2 = 1 * (seq(logcolgate) == (187 + 12))),
##     method = "ML", xtransf = data.frame(ADA = stepFColgate), transfer = list(c(0,
##         0)))
##
## Coefficients:
##           ma1     Imp1    Imp2  ADA-MA0
##       -0.8181  -0.6487  0.2211  -0.3064
## s.e.   0.0441   0.1444  0.1467   0.0873
##
## sigma^2 estimated as 0.02294:  log likelihood = 128.27,  aic = -248.55
```

```r
plot(ts(stepFCrest*(-0.3064)))
```



```r
plot(as.double(logcolgate), ylab = "Log(Colgate)", type = "l")
points(fitted(colgate.m1), col = "blue")
```



Training

Análisis Residuales

Predicción

Métricas de Predicción

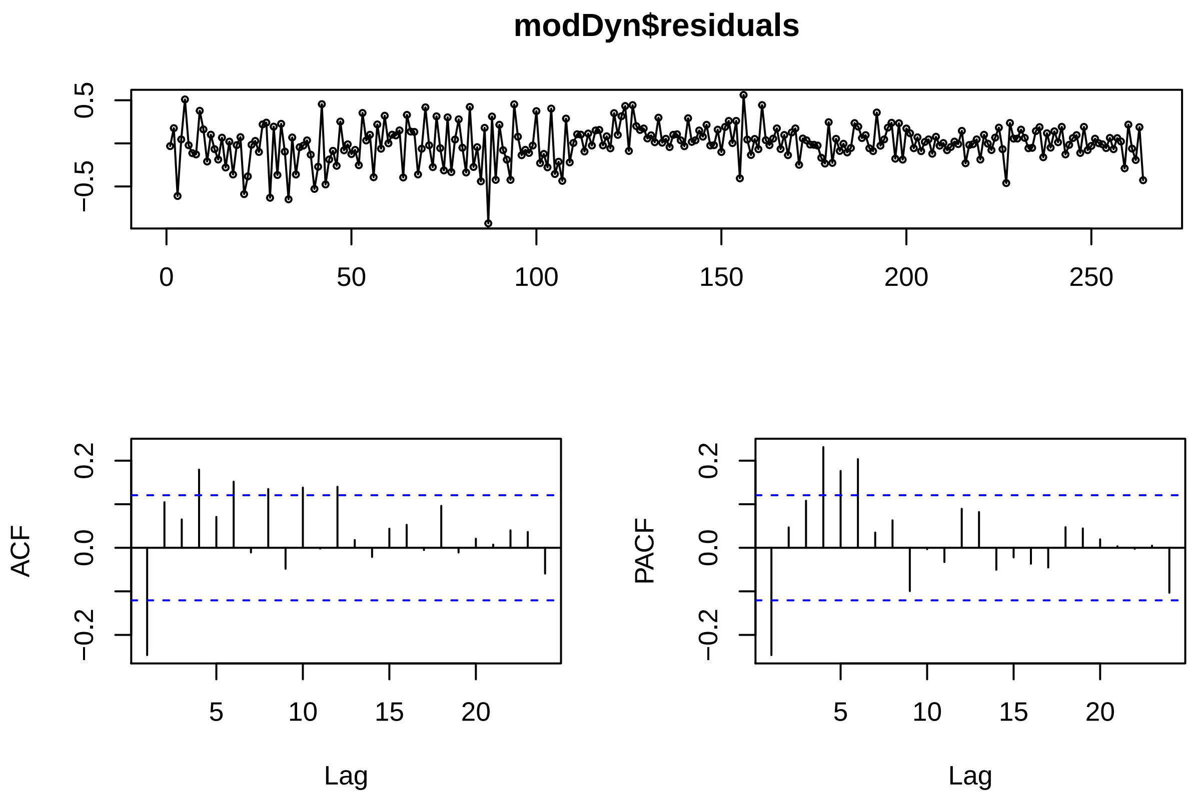
Regresión dinámica y función de transferencia

```r
modDyn <- dynlm(logcrest ~ L(logcrest, 1) + L(logcolgate, 0:12))
modDyn
```

```
##
## Time series regression with "zoo" data:
## Start = 1958-03-31, End = 1963-04-22
##
## Call:
## dynlm(formula = logcrest ~ L(logcrest, 1) + L(logcolgate, 0:12))
```

```
##
## Coefficients:
##           (Intercept)        L(logcrest, 1)   L(logcolgate, 0:12)0
##             -1.438983              0.704350              -0.613725
##  L(logcolgate, 0:12)1  L(logcolgate, 0:12)2   L(logcolgate, 0:12)3
##              0.255755             -0.119143              -0.061572
##  L(logcolgate, 0:12)4  L(logcolgate, 0:12)5   L(logcolgate, 0:12)6
##              0.040039              0.010597              -0.258171
##  L(logcolgate, 0:12)7  L(logcolgate, 0:12)8   L(logcolgate, 0:12)9
##              0.093637             -0.140630               0.057842
## L(logcolgate, 0:12)10 L(logcolgate, 0:12)11 L(logcolgate, 0:12)12
##             -0.002474              0.009127              -0.109328
```

```
tsdisplay(modDyn$residuals)
```

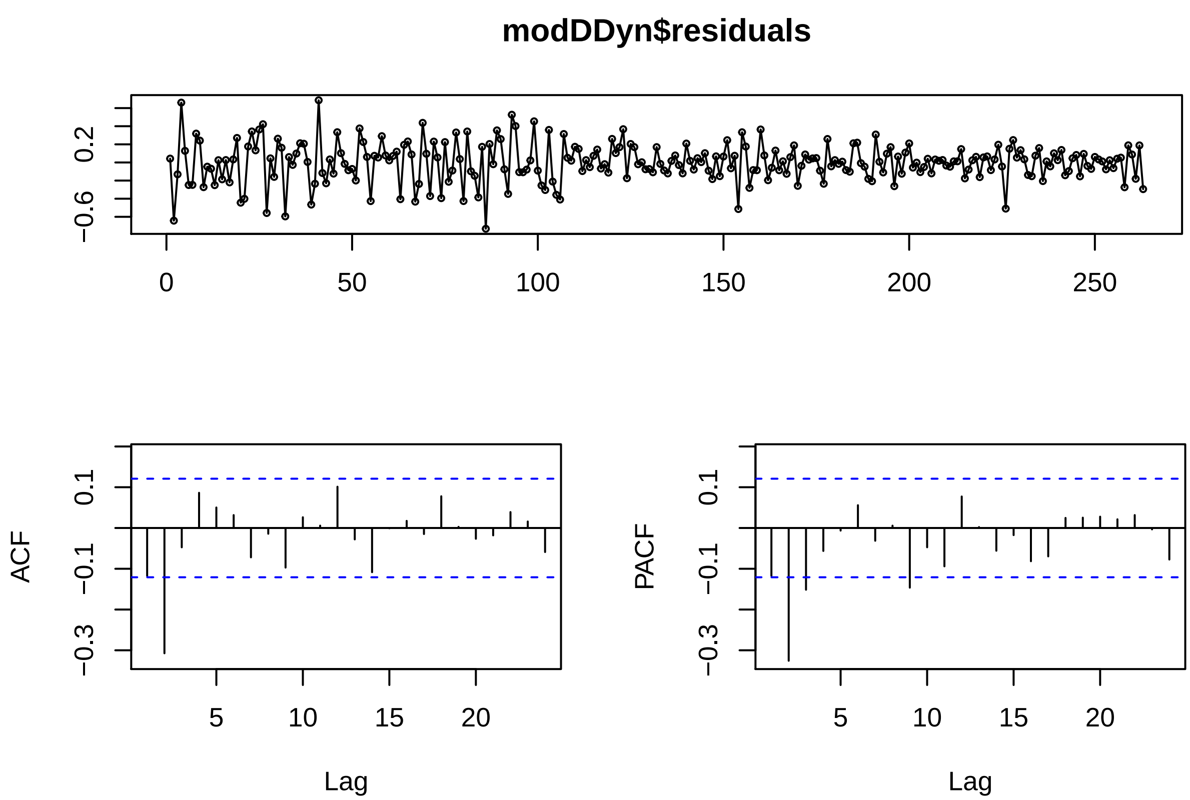



```
## Diff

modDDyn <- dynlm(diff(logcrest) ~ L(diff(logcrest), 1) + L(diff(logcolgate), 0:12))
modDDyn
```

```
##
## Time series regression with "zoo" data:
## Start = 1958-04-07, End = 1963-04-22
##
## Call:
```

```
## dynlm(formula = diff(logcrest) ~ L(diff(logcrest), 1) + L(diff(logcolgate),
##     0:12))
##
## Coefficients:
##                (Intercept)        L(diff(logcrest), 1)
##                  0.0032126                  -0.4537162
##  L(diff(logcolgate), 0:12)0   L(diff(logcolgate), 0:12)1
##                 -0.5204177                  -0.2801989
##  L(diff(logcolgate), 0:12)2   L(diff(logcolgate), 0:12)3
##                 -0.1187702                  -0.1333823
##  L(diff(logcolgate), 0:12)4   L(diff(logcolgate), 0:12)5
##                 -0.0101300                   0.0706820
##  L(diff(logcolgate), 0:12)6   L(diff(logcolgate), 0:12)7
##                 -0.1594822                  -0.0983913
##  L(diff(logcolgate), 0:12)8   L(diff(logcolgate), 0:12)9
##                 -0.1337614                  -0.0548736
## L(diff(logcolgate), 0:12)10  L(diff(logcolgate), 0:12)11
##                  0.0330939                   0.0819184
## L(diff(logcolgate), 0:12)12
##                  0.0009599
```

```
tsdisplay(modDDyn$residuals)
```

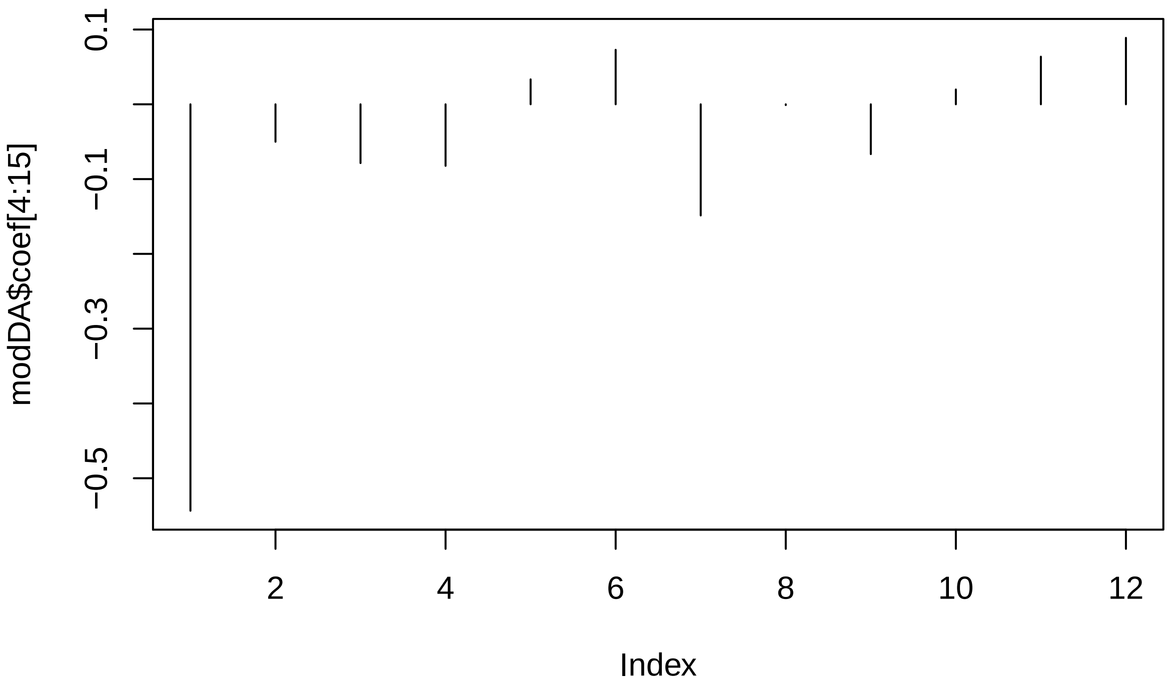


ARIMAX

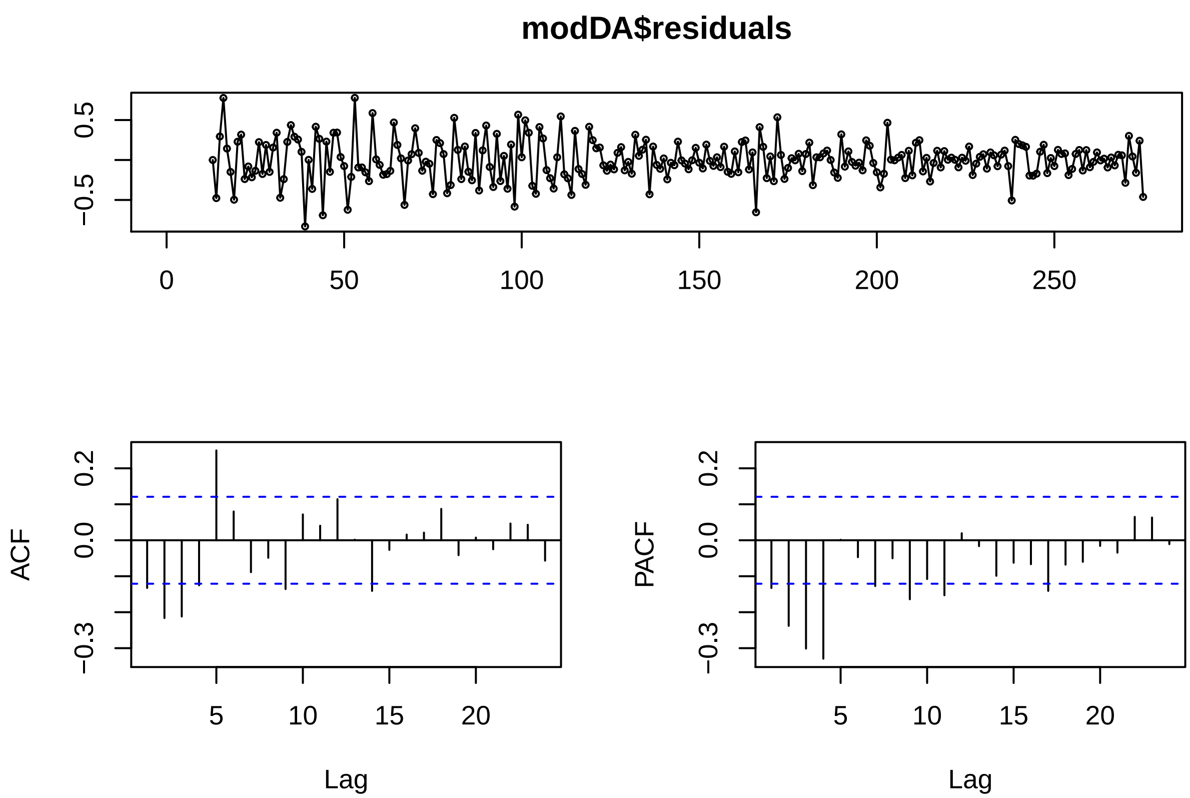
```r
### ARIMAX



modDA <- arimax(as.double(diff(logcrest)),
                order = c(3,1,0),
                include.mean = T,
                xtransf = as.double(diff(logcolgate)),
                transfer = list(c(0,12)),
                method = "ML")
modDA
```

```
##
## Call:
## arimax(x = as.double(diff(logcrest)), order = c(3, 1, 0), include.mean = T,
##     method = "ML", xtransf = as.double(diff(logcolgate)), transfer = list(c(0,
##         12)))
##
## Coefficients:
##           ar1      ar2      ar3   T1-MA0   T1-MA1   T1-MA2   T1-MA3
##       -1.1435  -0.8775  -0.4533  -0.5434  -0.0501  -0.0786  -0.0823
## s.e.   0.0559   0.0740   0.0575   0.0912   0.1007   0.1117   0.1268
##       T1-MA4  T1-MA5   T1-MA6   T1-MA7   T1-MA8  T1-MA9  T1-MA10  T1-MA11
##       0.0333  0.0729  -0.1486  -0.0010  -0.0666  0.0198   0.0637   0.0888
## s.e.  0.1466  0.1495   0.1512   0.1492   0.1472  0.1274   0.1134   0.1009
##       T1-MA12
##        0.0522
## s.e.   0.0925
##
## sigma^2 estimated as 0.06194:  log likelihood = -8.22,  aic = 48.45
```

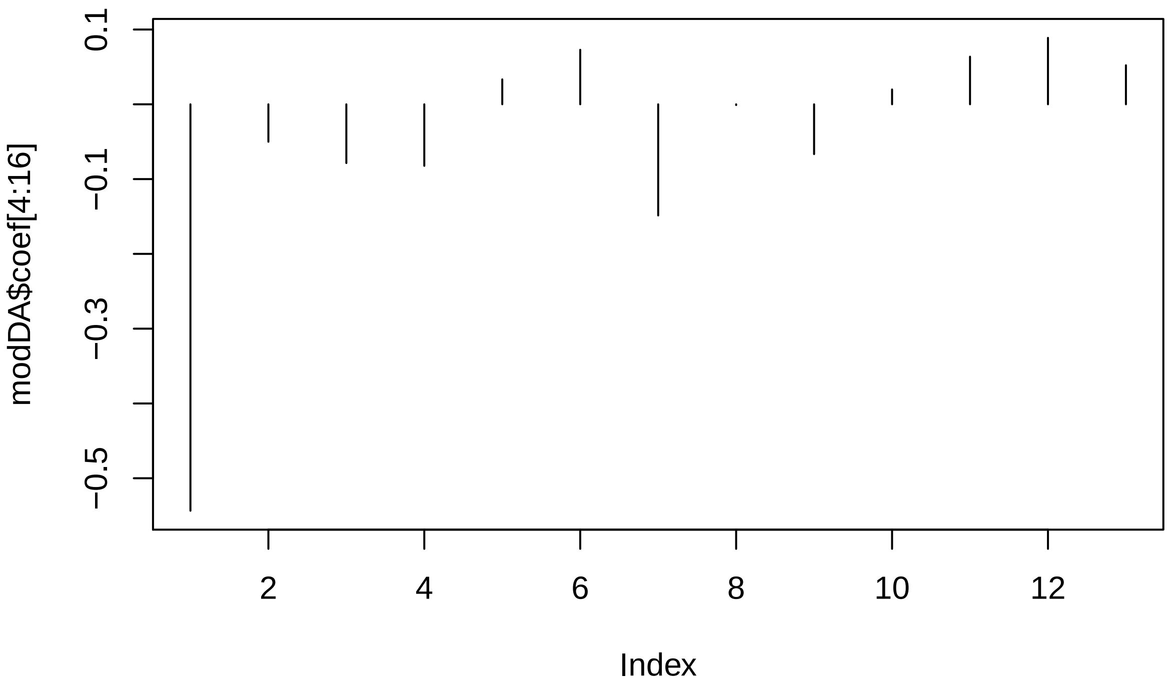
```r
plot(modDA$coef[4:15], type = "h")
```




```
tsdisplay(modDA$residuals)
```




```
plot(modDA$coef[4:16], type = "h")
```

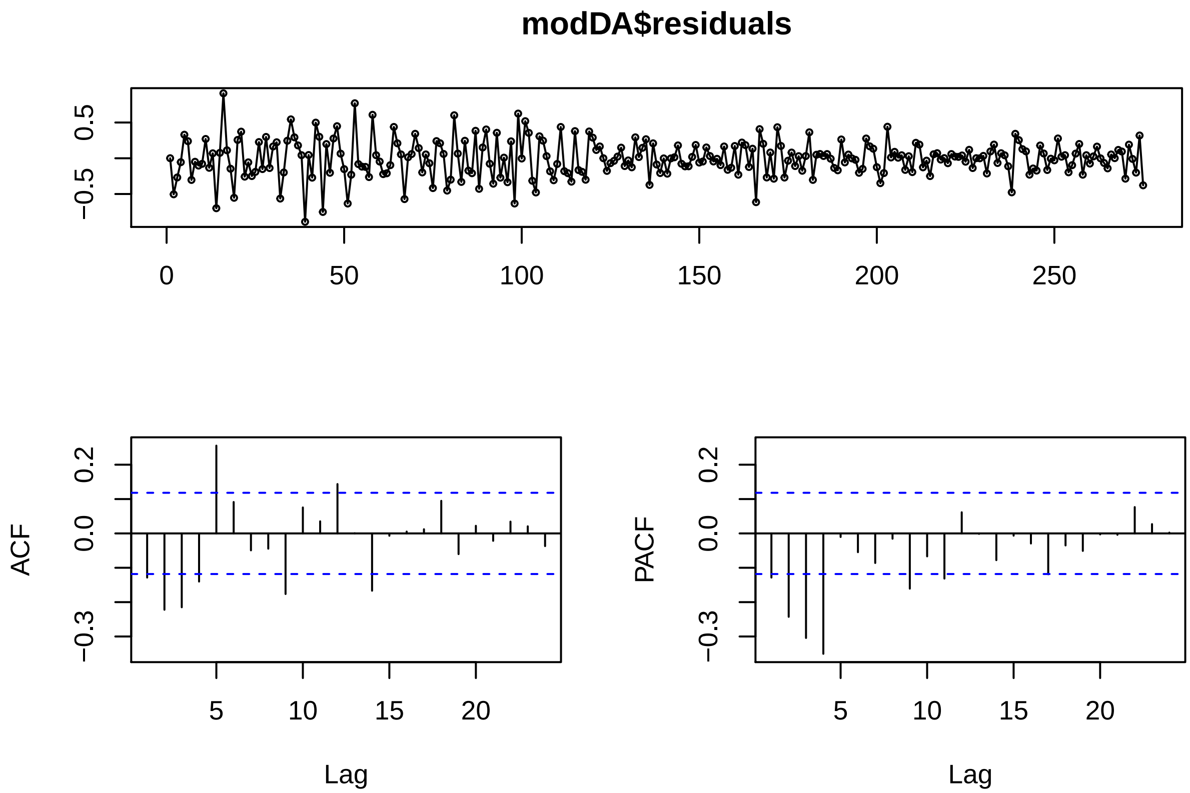


Todos Signifcativos

```r
modDA <- arimax(as.double(diff(logcrest)),
                order = c(3,1,0),
                include.mean = T,
                xtransf = as.double(diff(logcolgate)),
                transfer = list(c(0,0)),
                method = "ML")
modDA
```

```
##
## Call:
## arimax(x = as.double(diff(logcrest)), order = c(3, 1, 0), include.mean = T,
##     method = "ML", xtransf = as.double(diff(logcolgate)), transfer = list(c(0,
##         0)))
##
## Coefficients:
##           ar1      ar2      ar3   T1-MA0
##       -1.1280  -0.8471  -0.4405  -0.4751
## s.e.   0.0547   0.0708   0.0547   0.0717
##
## sigma^2 estimated as 0.06559:  log likelihood = -16.38,  aic = 40.75
```

```r
tsdisplay(modDA$residuals)
```


modDA$residuals
0.5
-0.5
0
50
100
150
200
250
ACF
0.2
0.0
-0.3
5
10
15
20
Lag
PACF
0.2
0.0
-0.3
5
10
15
20
Lag