Series de Tiempo - Trabajo 2

Integrantes:

Brahian Steven Serna Restrepo. CC 1007396943

Jimena Uribe Giraldo. CC 1001022793

Julián Saavedra Echavarría. CC 1000883721

Nataly García Osorio. CC 1007239212

Cargar paquetes

```
library(ggplot2)
library(car)
library(lmtest)
library(tseries)
library(quantmod)
library(foreign)
library(astsa)
library(forecast)
library(urca)
library(fUnitRoots)
```

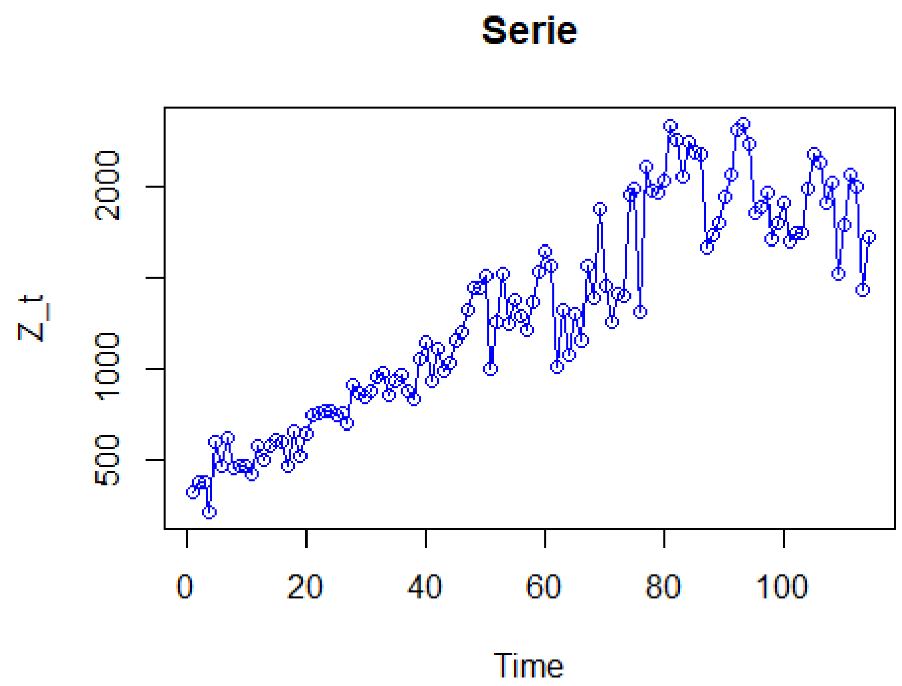
Punto A.

```
datos <- read.csv("Serie01_We_02_W6.csv", sep = ";")
datos_ts = ts(datos) # Conversión a serie de tiempo
```

Introducción:

Como siempre, debemos hacer un primer acercamiento a los datos, mirar de qué forma se comportan para así identificar el proceso a seguir. Así, la gráfica de los datos se presenta a continuación:

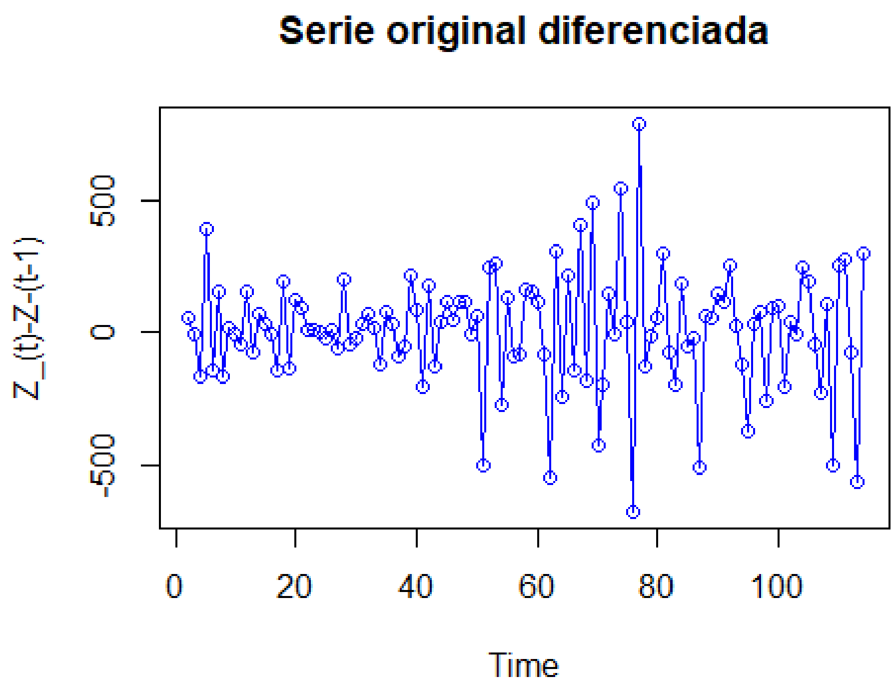
```
# Primera vista graficamente de los datos:
plot(datos_ts, col = "blue", main = "Serie", ylab = "Z_t", type = "o")
```



Primeras observaciones:

- No se trata de un proceso estacionario.
- Parece ser un proceso integrado, es decir que los valores de la serie de tiempo parecen estar relacionados entre sí
- la varianza de la serie no es homogénea y si diferenciamos esto se verá más evidente:

```
plot.ts(diff(datos_ts), ylab = "Z_(t)-Z-(t-1)", col = "blue",
        main = "Serie original diferenciada", type = "o")
```



##

Transformación Box-Cox

Observamos que efectivamente estamos ante un caso donde la varianza NO es homogénea.

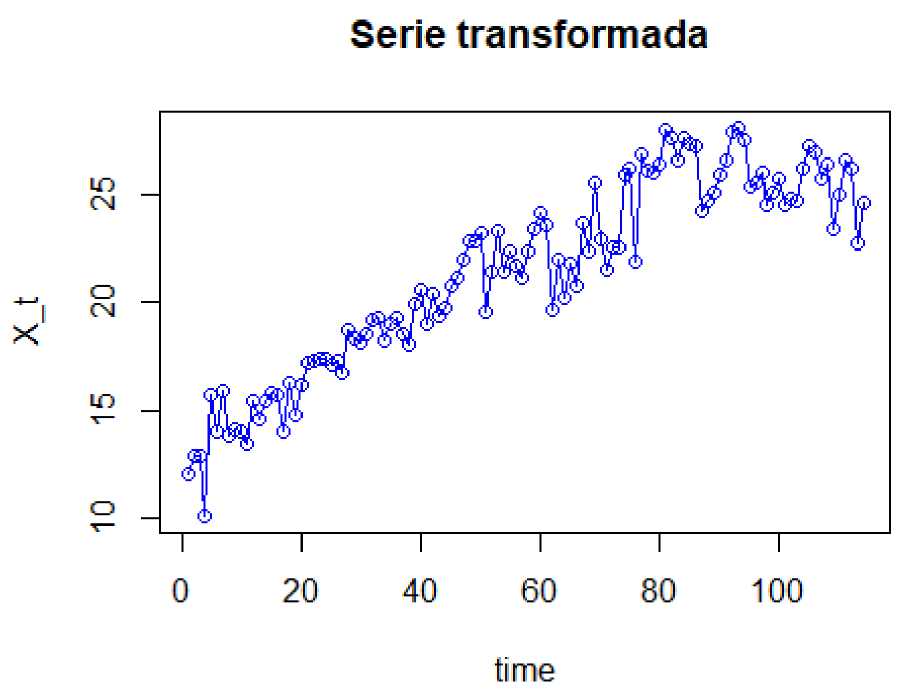
Como la varianza de la serie no es homogénea, se estimará “Lambda” de la transformación Box-Cox.

```
(tBoxCox=powerTransform(datos_ts))

## Estimated transformation parameter
##  datos_ts
## 0.7496944

summary(tBoxCox)

## bcPower Transformation to Normality
##          Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## datos_ts    0.7497           1       0.3633       1.1361
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                             LRT df       pval
## LR test, lambda = (0) 16.24151  1 5.5759e-05
##
## Likelihood ratio test that no transformation is needed
##                             LRT df    pval
## LR test, lambda = (1) 1.551829  1 0.21287
```

```r
BoxCox.lambda(datos_ts, method=c("guerrero"))

## [1] 0.4348836
```

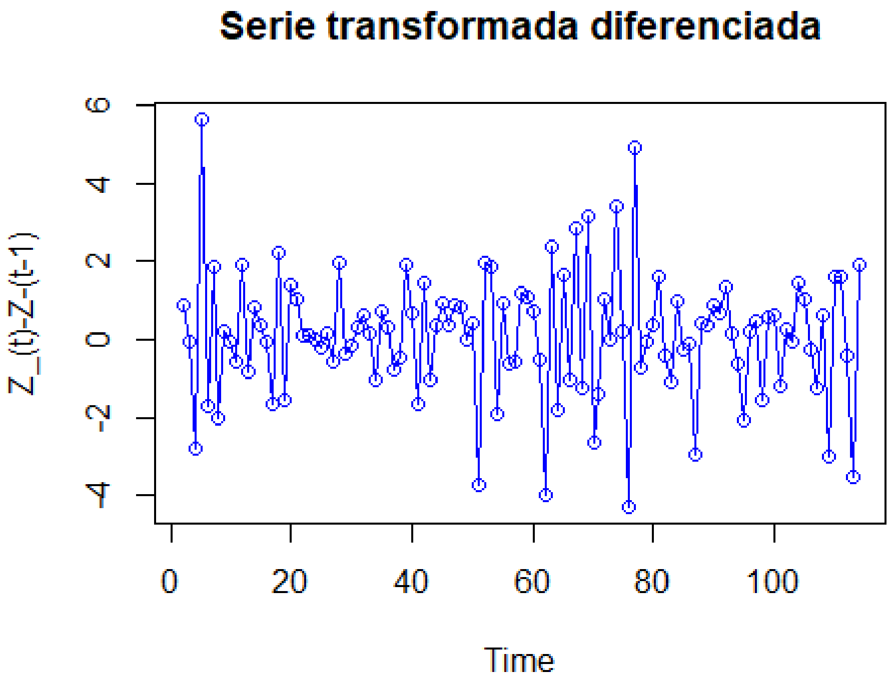
Se selecciona la estimación de lambda del método “guerrero” el cual es aproximadamente 0.435. Vamos entonces a transformar la serie, utilizando la raíz cuarta:

```r
plot.ts(datos_ts^(.43) ,main = " Serie transformada",ylab = "X_t", xlab =
"time", type = "o",col='blue', lwd = 1 )
```



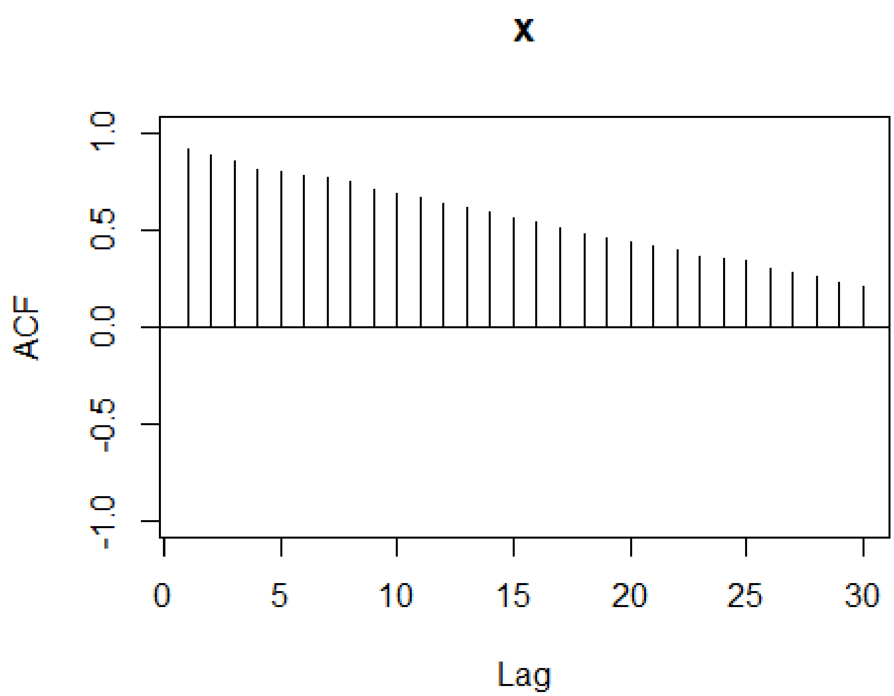
Con esto el problema de la varianza mejora con relación al primer observamiento que se le hizo a la serie, para observar esta mejoría diferenciemos la transformación y note esto mismo:

```r
plot.ts(diff(datos_ts^.43), ylab = "Z_(t)-Z-(t-1)", col = "blue",
        main = "Serie transformada diferenciada", type = "o")
```

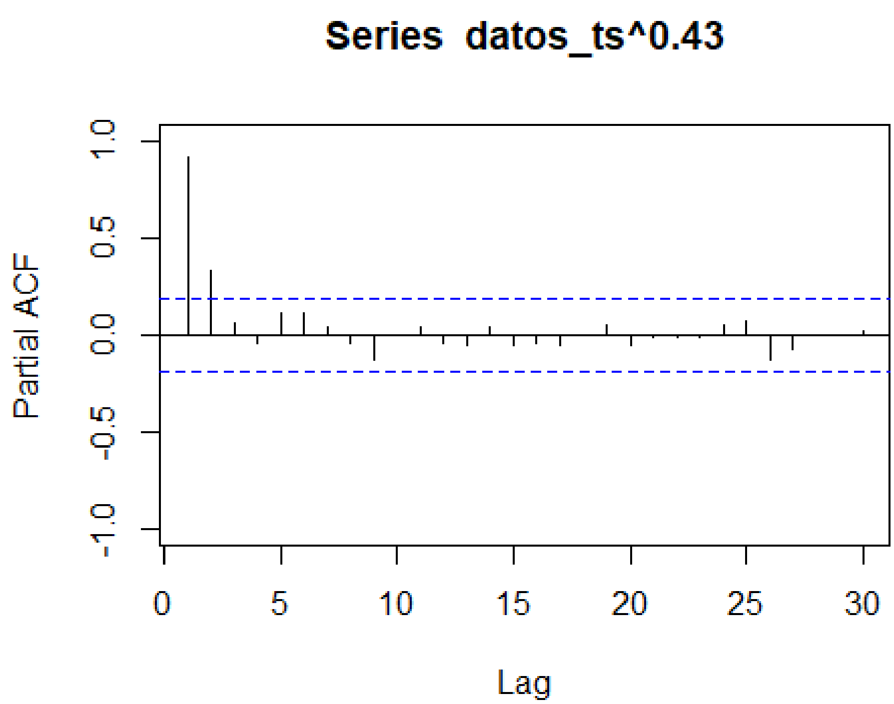


Continuaremos usando la transformación de la serie como si fuera la serie original usando lambda de 0.43, ahora vamos a graficar los correlogramas para buscar evidencia para diferenciar la serie.

```
Acf(datos_ts^0.43, lag.max=30, ci=0,ylim=c(-1,1))
```

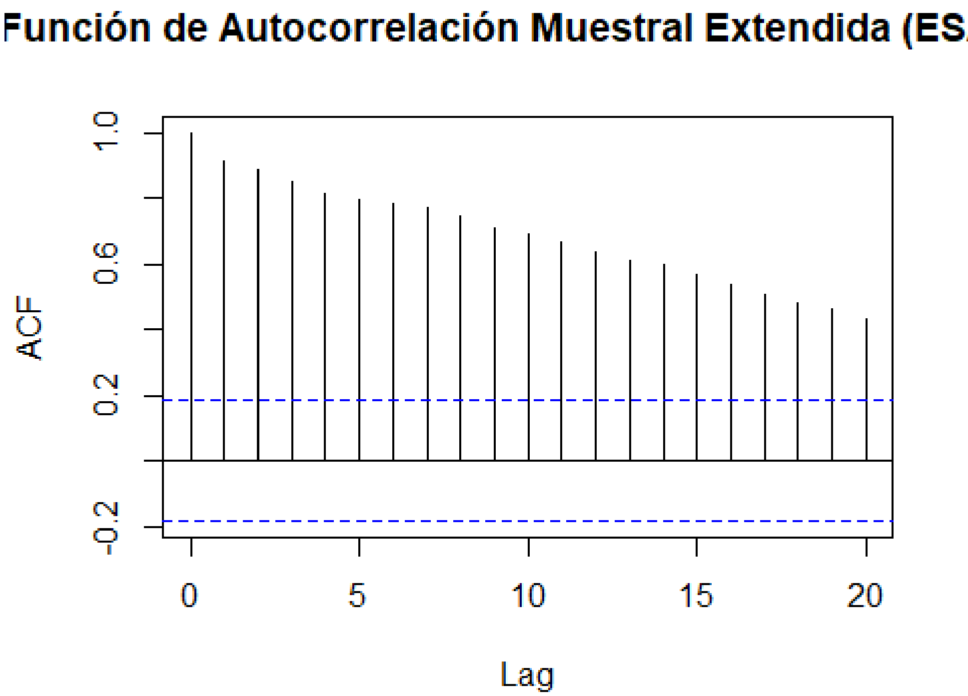



```
pacf(datos_ts^0.43, lag.max=30, ylim=c(-1,1))
```




```
# Calcular la ESACF
esacf_result <- stats::acf(datos_ts^0.43, plot = FALSE)
```

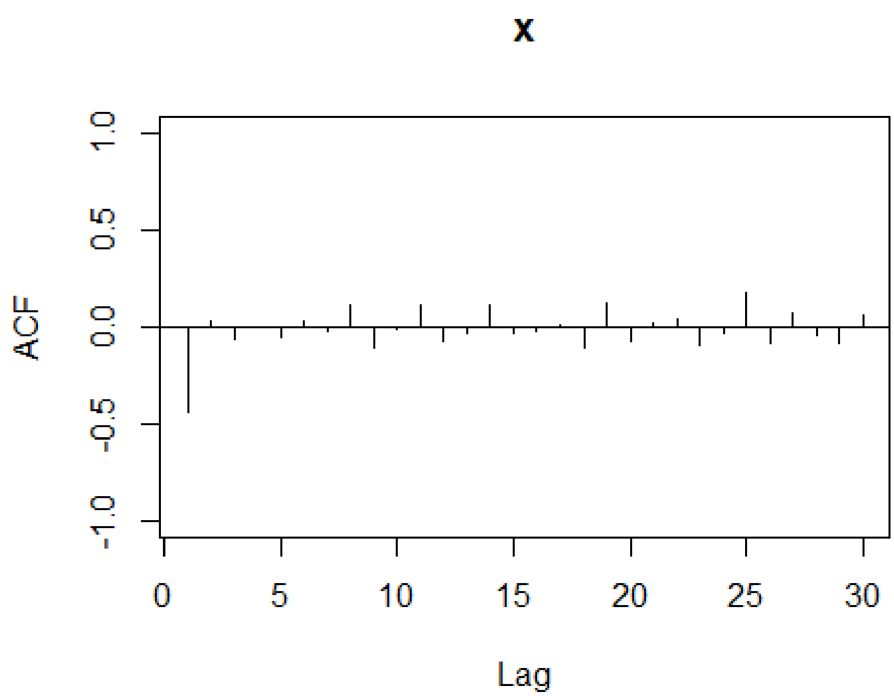
```
plot(esacf_result, main = "Función de Autocorrelación Muestral Extendida (ESACF)")
```



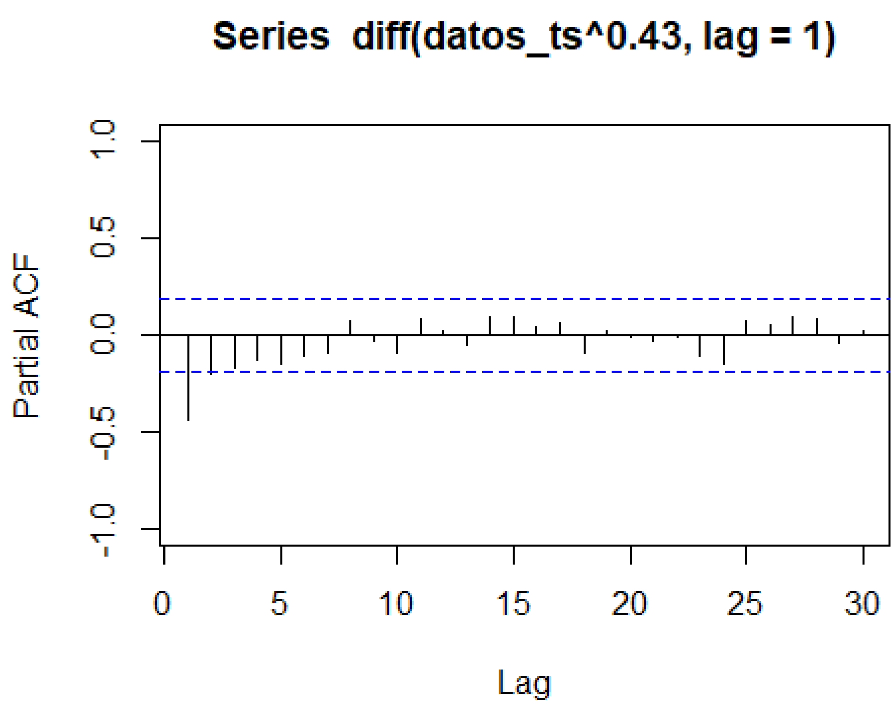
En la ACF se demora mucho en caer y en la PACF tiene un pico alto en el primer rezago, esto es una evidencia de que la serie necesite ser diferenciada (Más adelante se corroborará esto con la prueba de raíces unitarias).

El modelo que propongo por ahora es un ARIMA(p,1,q) porque a simple vista la tendencia es lineal positiva de orden 1, Esto se corrobora en la prueba de raíces unitarias, ahora miremos el ACF y PACF con la serie diferenciada para proponer un p y q

```
Acf(diff(datos_ts^.43,lag = 1), lag.max=30, ci=0,ylim=c(-1,1))
```




```
pacf(diff(datos_ts^.43,lag = 1), lag.max=30, ylim=c(-1,1))
```



El análisis de la PACF parece indicar que hay decaimiento exponencial tanto en los rezagos no estacionales como estacionales; por su parte la ACF señala que hay un corte después del primer rezago.

Por lo anterior el modelo que propongo es un ARIMA(0,1,1)

Punto B.

```
auto.arima(datos_ts^0.43,max.p=5,max.q=5)

## Series: datos_ts^0.43
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.6595  0.1102
## s.e.   0.0823  0.0449
##
## sigma^2 = 1.929:  log likelihood = -196.75
## AIC=399.5   AICc=399.72   BIC=407.68
```

Observe que efectivamente se estima que es un modelo ARIMA(0,1,1)

Punto C

De igual manera, realicemos una prueba de Dickey- Fuller para confirmar que no hay estacionariedad en la serie original, para esta prueba estamos ante el siguiente juego de hipótesis:

$$H_0 : \text{La serie no es estacionaria} \quad vs. \quad H_1 : \text{La serie es estacionaria}$$

Así, se tiene el test:

```
adf.test(datos_ts^0.43, alternative = "stationary")

##
##  Augmented Dickey-Fuller Test
##
## data:  datos_ts^0.43
## Dickey-Fuller = -1.7134, Lag order = 4, p-value = 0.6956
## alternative hypothesis: stationary
```

No se rechazar H_0, ya que $\alpha = 0.05 < p - value = 0.6956$ y esto significa que existe una o más raíces unitarias en la serie.

- Como esta serie no es estacionaria debemos convertirla a estacionaria, podemos hacerlo con diferencias o logaritmos. En nuestro caso, vamos a trabajar con diferencias.

Diferenciamos los datos originales y los trabajamos como si fueran los datos originales y aplicamos la prueba de Dickey-Fuller

```
adf.test(diff(datos_ts^0.43), alternative = "stationary")

## Warning in adf.test(diff(datos_ts^0.43), alternative = "stationary"):
p-value
## smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  diff(datos_ts^0.43)
## Dickey-Fuller = -7.2755, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Se observa que la prueba se acepta que la serie es estacionaria por ende se comprueba que solo hay una raíz unitaria.

```
serie_transf <- datos_ts^0.43

(maxlag=floor(12*(length(datos_ts)/100)^(0.75)))

## [1] 13

ru_tz = ur.df(serie_transf, type = c("trend"), lags=maxlag, selectlags =
c("BIC"))
summary(ru_tz)

##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0133 -0.6106  0.0747  0.8604  3.3266
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.95680    1.46923   3.374  0.00107 **
## z.lag.1     -0.30440    0.09588  -3.175  0.00201 **
## tt           0.03081    0.01212   2.542  0.01262 *
## z.diff.lag  -0.22909    0.10444  -2.194  0.03068 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.375 on 96 degrees of freedom
## Multiple R-squared:  0.238,  Adjusted R-squared:  0.2142
## F-statistic: 9.994 on 3 and 96 DF,  p-value: 8.565e-06
##
##
## Value of test-statistic is: -3.1749 3.9345 5.4944
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

```
ru_tz=ur.df(serie_transf, type = c("trend"), lags=maxlag, selectlags =
c("AIC"))
summary(ru_tz)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0133 -0.6106  0.0747  0.8604  3.3266
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.95680    1.46923   3.374  0.00107 **
## z.lag.1     -0.30440    0.09588  -3.175  0.00201 **
## tt           0.03081    0.01212   2.542  0.01262 *
## z.diff.lag  -0.22909    0.10444  -2.194  0.03068 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.375 on 96 degrees of freedom
## Multiple R-squared:  0.238,  Adjusted R-squared:  0.2142
## F-statistic: 9.994 on 3 and 96 DF,  p-value: 8.565e-06
##
##
## Value of test-statistic is: -3.1749 3.9345 5.4944
##
```

```
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

punto D.

```
mod1_CSS_ML=Arima(datos_ts, c(0, 1, 1), include.drift=TRUE, lambda=.43,
method = c("CSS-ML"))
summary(mod1_CSS_ML)
```

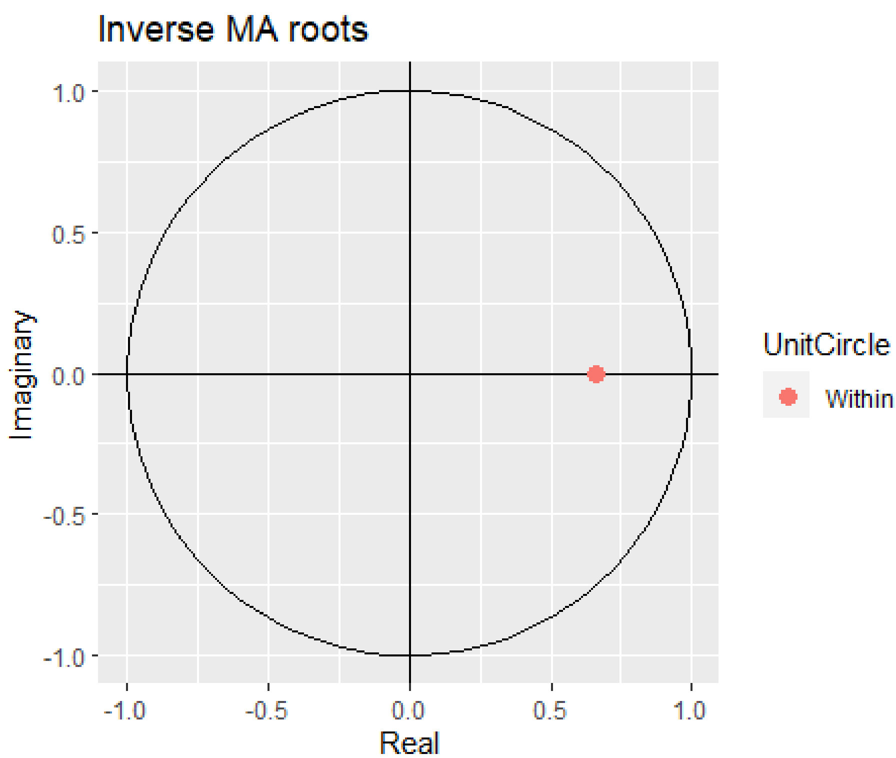
```
## Series: datos_ts
## ARIMA(0,1,1) with drift
## Box Cox transformation: lambda= 0.43
##
## Coefficients:
##           ma1   drift
##       -0.6595  0.2562
## s.e.   0.0823  0.1044
##
## sigma^2 = 10.43:  log likelihood = -292.12
## AIC=590.24   AICc=590.46   BIC=598.42
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
ACF1
## Training set -1.445827 199.3047 144.5044 -1.009664 11.77491 0.9005126
0.1384639
```

El modelo con los parámetros estimados seria el siguiente:

$$(1\text{-b})Z_t^{0.43} = (\mathrm{I} + 0.6595\mathrm{b})a_t + 0.2562$$

Punto E

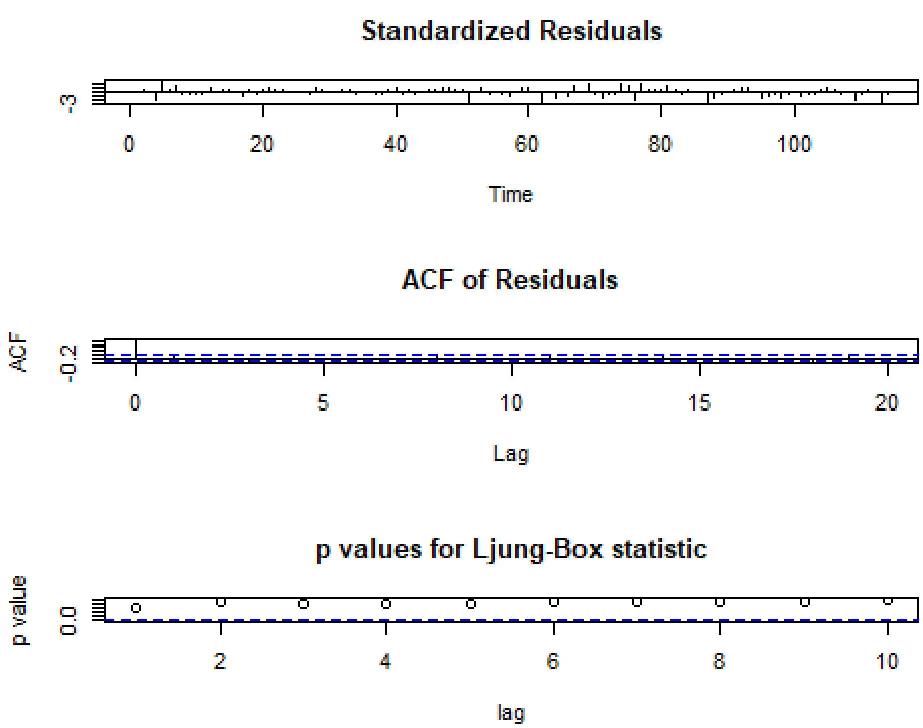
```
autoplot(mod1_CSS_ML)
```



Como la raíz está adentro del círculo de unidad quiere decir que la serie es estacionaria.

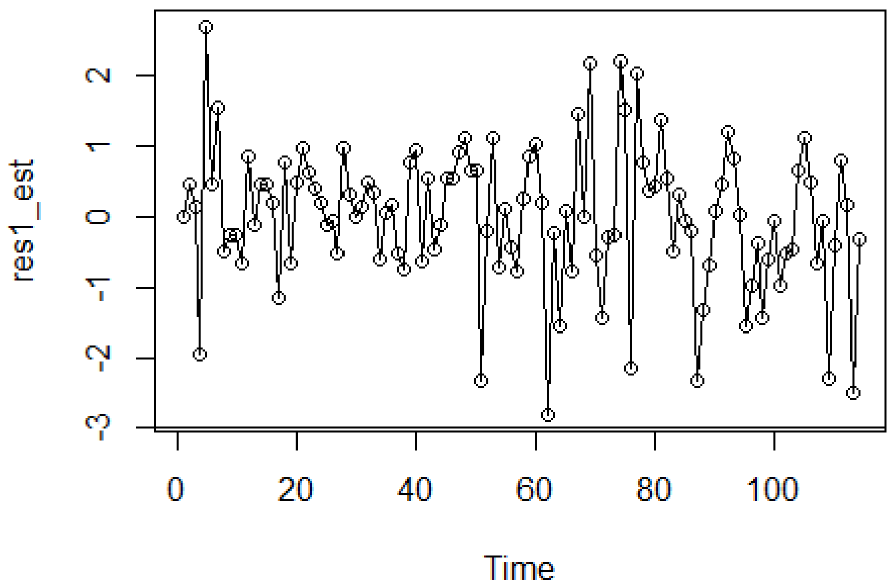
Analísis de los residuales

```
tsdiag(mod1_CSS_ML)
```



Fluctúa alrededor de un valor fijo, es decir la media y la varianza parece ser constante y según la ACF el primer rezago esta correlacionado consigo mismo obviamente y luego cae cuando se calcula la correlación con los demás rezagos.

```
res1_CSS_ML=residuals(mod1_CSS_ML)

res1_est=res1_CSS_ML/(mod1_CSS_ML$sigma2^0.5)
plot.ts(res1_est, type="o")
abline(a=-3, b=0)
abline(a=3, b=0)
```



Con el grafico anterior corroboramos lo anterior dicho.

Bajo la hipótesis de normalidad el número esperado A de observaciones atípicas es:

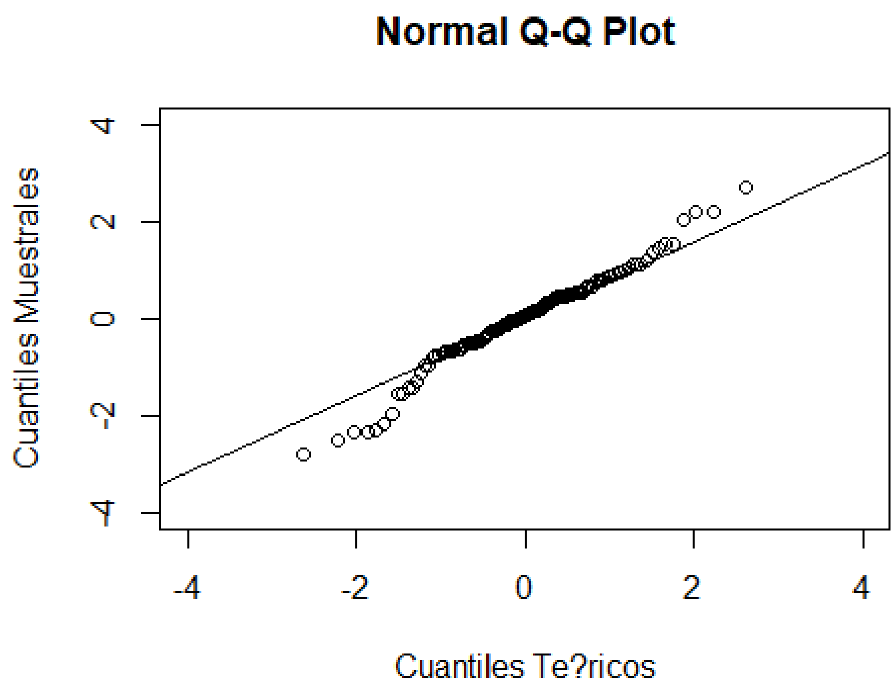
```
(Nobs_Esp=round(length(datos_ts)*2*pnorm(-3, mean = 0, sd = 1, lower.tail
= TRUE)))

## [1] 0
```

Se detectan las observaciones atípicas

```
ind=(abs(res1_est)>3.0)
sum(ind)

## [1] 0
```

Se verifica la normalidad de los residuales con un q-q plot

```
qqnorm(res1_est, xlab = "Cuantiles Te?ricos", ylab = "Cuantiles
Muestrales",
xlim=c(-4,4), ylim=c(-4,4))
qqline(res1_est)
```




```
shapiro.test(res1_est)

##
##  Shapiro-Wilk normality test
##
## data:  res1_est
## W = 0.97458, p-value = 0.02857
```

No se cumple el test de normalidad de Shapiro-Wilk y esto se corrobora con la gráfica donde los cuantiles Teóricos no contienen bien a los cuantiles Muéstrales, por tanto no se puede hacer observaciones de los datos atipicos con la distribución normal.

Punto F

¿La tendencia de la serie posee componentes determinística y aleatoria, solo una de ellas (cúal) o ambas?.

Las componentes de una serie temporal pueden ser de naturaleza determinista o aleatoria. En el caso de la tendencia de la serie analizada y la forma en que se integra, se determina que se trata de una serie con componentes aleatorios y al mismo tiempo determinísticos, pues se utilizan modelos autorregresivos- medias móviles para el modelamiento en la serie temporal, además de ser no estacionario y tener la presencia de una raíz unitaria en el componente autorregresivo del proceso generado,

¿El proceso adecuado para modelar la serie se trata de un proceso estacionario en tendencia o un proceso de diferencias estacionarias?

El proceso adecuado para modelar la serie se trata de un proceso de diferencias estacionarias, pues la serie de tiempo muestra una variabilidad que cambia a lo largo del tiempo, siendo un proceso no estacionario cuya no estacionaridad está motivada por la presencia de raíces unitarias.