Series de Tiempo - Trabajo 3 y 4

Integrantes:

Brahian Steven Serna Restrepo. CC 1007396943

Jimena Uribe Giraldo. CC 1001022793

Julián Saavedra Echavarría. CC 1000883721

Nataly García Osorio. CC 1007239212

Punto 1:

```
library(astsa)
library(tidyverse)
library(forecast)
library(tseries)
library(car)
library(MASS)
```

El conjunto de datos # 1 es el correspondiente para este trabajo.

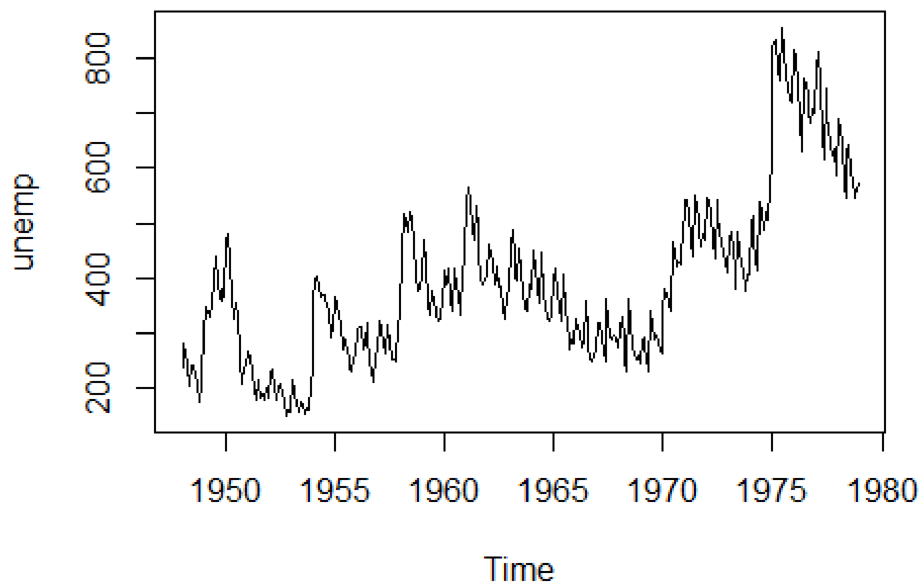
Primero, verifiquemos que los datos “unemp” sean de tipo ts.

```
# Lectura de datos:
class(unemp)
```

a) Gráfica y examen inicial de los datos:

Grafiquemos y notemos qué comportamiento toma la serie de tiempo.

```
plot(unemp)
```



Realicemos la prueba de Dickey-Fuller para confirmar si hay estacionariedad o no.

Para eso, planteamos el siguiente juego de hipótesis:

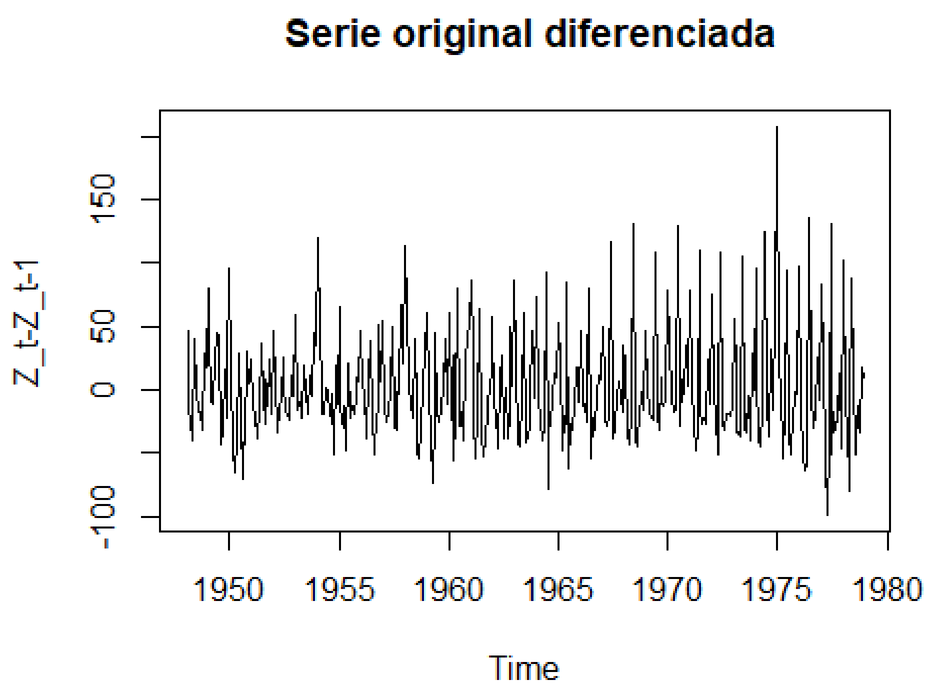
$$H_0 : \text{La serie tiene al menos una raíz unitaria (No estacionaria)}$$
$$vs.$$
$$H_1 : \text{La serie no tiene raíces unitarias (Estacionaria)}$$

```
adf.test(unemp)

##
##  Augmented Dickey-Fuller Test
##
## data:  unemp
## Dickey-Fuller = -3.5175, Lag order = 7, p-value = 0.04112
## alternative hypothesis: stationary
```

Según esta prueba de Dickey-Fuller, se acepta la hipótesis alternativa que nos dice que la serie es estacionaria. NO vamos a aceptar esta prueba como único fundamento para definir estacionariedad en la serie ya que gráficamente podemos ver comportamientos con tendencias. Por lo tanto, diremos que la serie no es estacionaria.

De igual manera, vamos graficar la serie diferenciada para quitar tendencia y poder ver bien el comportamiento de la varianza en la serie:

```
plot.ts(diff(unemp), main = " Serie original diferenciada",
        ylab = "Z_t-Z_t-1")
```



Notemos que según esta grafica, la serie podría tener una media constante (0) pero la varianza parece fluctuar un poco, por lo que no asumimos varianza constante. Así, para el examen inicial de los datos tenemos:

- No se trata de un proceso estacionario.
- La varianza de la serie no es homogénea.
- Posiblemente se trate de un proceso integrado (raíces unitarias).

b) Tranformación de los datos:

Aplicaremos la transformación Box-Cox:

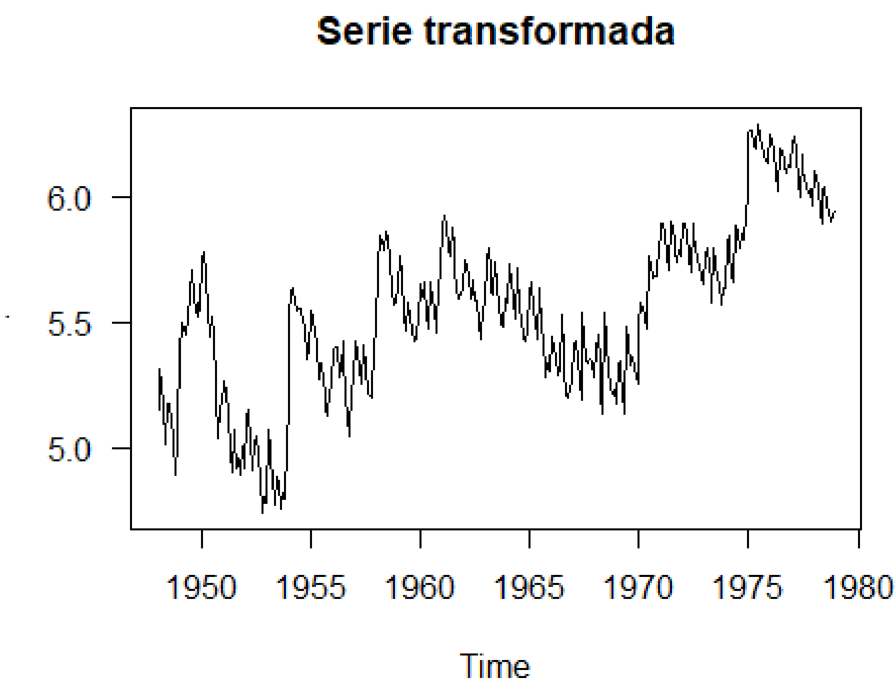
```
# Lo haremos de la siguiente manera, con la libreria forecast:
unemp_lambda <- powerTransform(unemp)
unemp_lambda

## Estimated transformation parameter
##       unemp
## -0.02129048
```

Así, el mejor valor será $\lambda = -0.02129048$. Como lambda es diferente de cero entonces descartamos la transformación por $\ln X_t$

```
# Ahora, aplicamos el el método con el valor de lambda encontrado
unemp_transf <- BoxCox(unemp, lambda= -0.02129048)
```

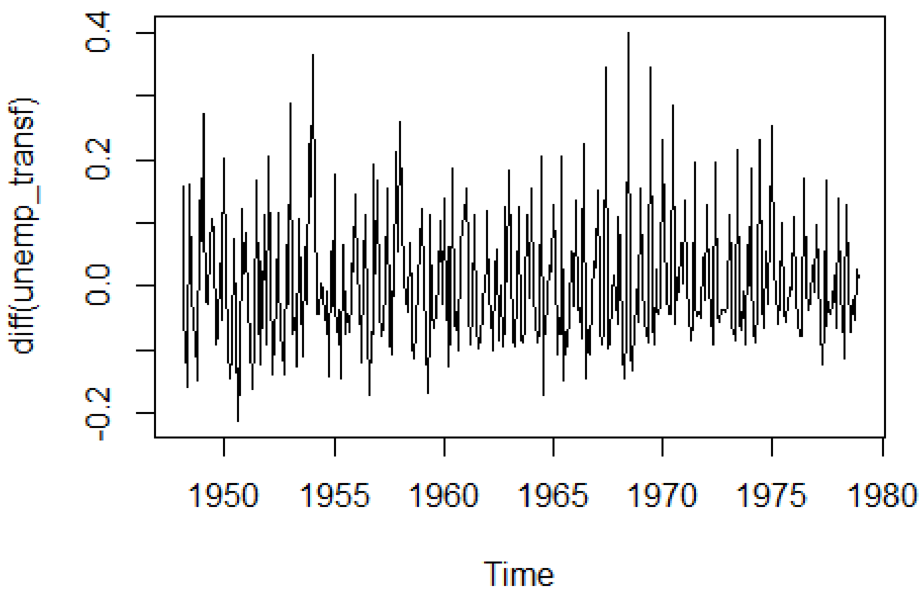
Así, la serie transformada se verá de la siguiente manera:

```
unemp_transf %>% plot(main = "Serie transformada", las = 1)
```



Notemos el comportamiento de la varianza con la tranformación, para eso diferenciamos la serie tranformada:

```
plot.ts(diff(unemp_transf))
```

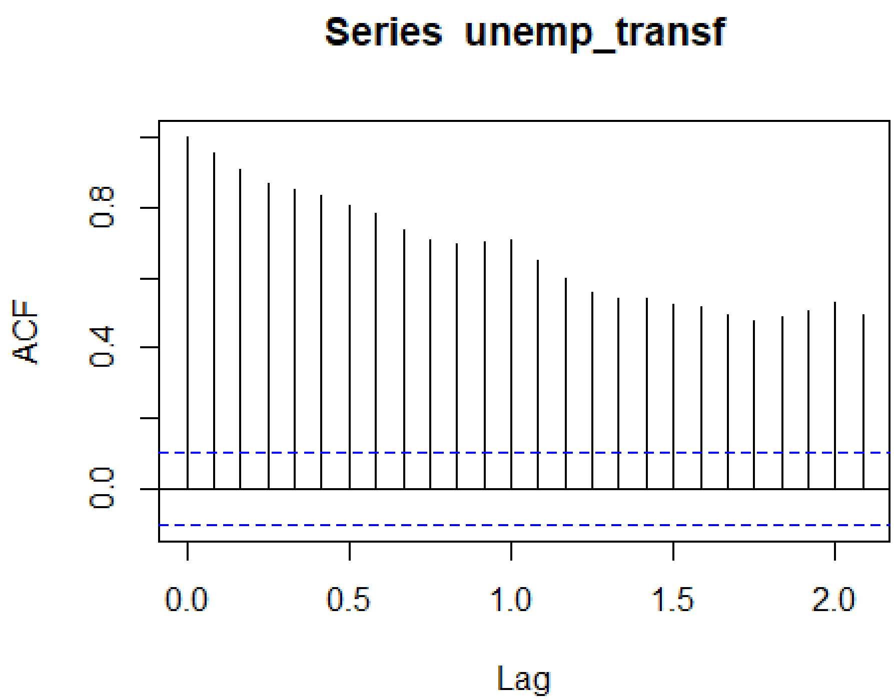


Se puede apreciar una mejoría en varianza y media, la cual se mueve alrededor de 0.

c) Orden de diferenciación e identificación del modelo:

En este caso, seleccionaremos el modelo transformado para trabajar.

```
acf(unemp_transf, type ="correlation")
```



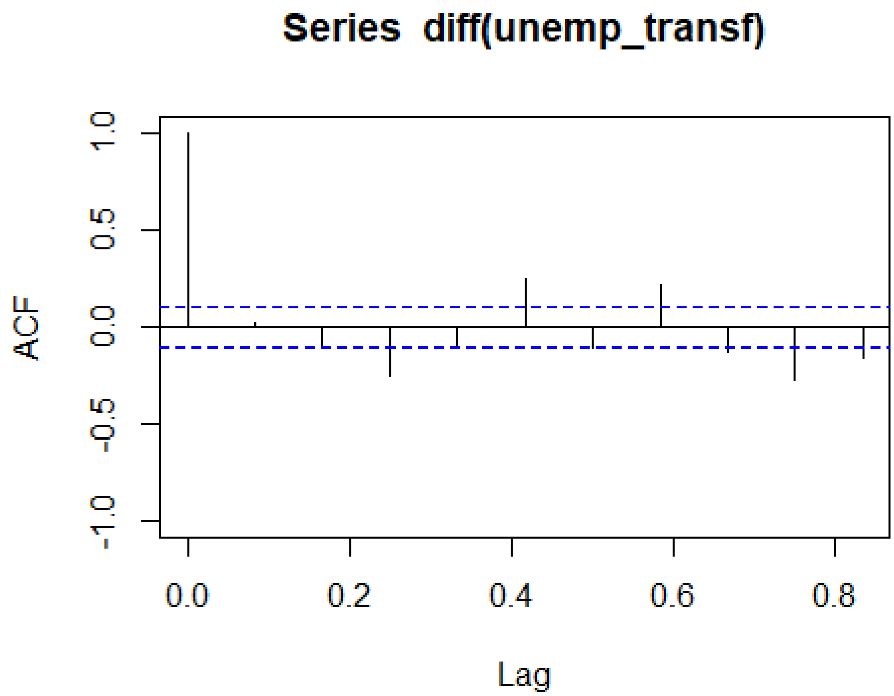
Notemos un comportamiento que decrece lentamente, lo cual nos indica que es necesario aplicar diferencias. Ahora, con la función ndiffs(), consultaremos cuáles son las diferencias recomendadas:

```
ndiffs(unemp_transf)

## [1] 1
```

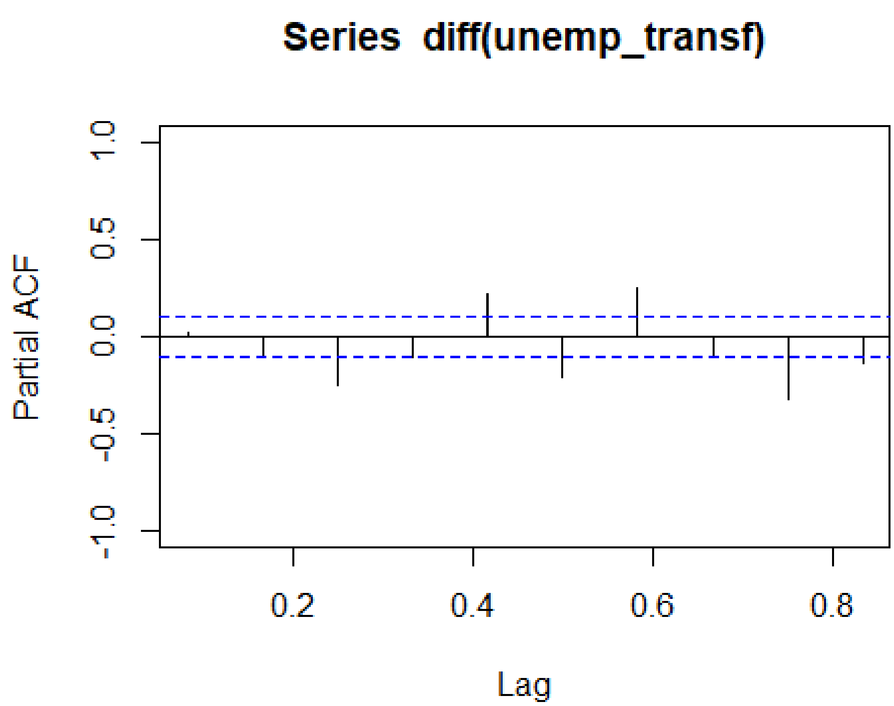
Así, las diferencias necesarias para que el proceso sea estacionario es de 1 diferencia. Por lo que tenemos un proceso Integrado de orden 1.

Así, el ACF y el PACF se verían de la siguiente manera:

```
## ACF:
acf(diff(unemp_transf), lag.max=10, ylim=c(-1,1))
```




```
#PACF:
pacf(diff(unemp_transf), lag.max=10, ylim=c(-1,1))
```



Ahora, verifiquemos con el test de Dickey-Fuller que la serie diferenciada sí sea estacionaria:

```
serie_transf <- diff(unemp_transf)
adf.test(serie_transf)

## Warning in adf.test(serie_transf): p-value smaller than printed p-
value

##
##  Augmented Dickey-Fuller Test
##
## data:  serie_transf
## Dickey-Fuller = -6.9115, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

Por lo que, antes el juego de hipótesis:

$$H_0 : \text{La serie tiene al menos una raíz unitaria (No estacionaria)}$$
$$vs.$$
$$H_1 : \text{La serie no tiene raíces unitarias (Estacionaria)}$$

Rechazamos con seguridad la hipótesis nula y aceptamos la alternativa. Por lo que concluimos que la serie transformada y diferenciada es estacionaria y podemos trabajar con dicho modelo.

Así, el orden del modelo seleccionado será:

```
modelo_con <- auto.arima(serie_transf)
modelo_con

## Series: serie_transf
## ARIMA(2,0,2)(2,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sar1     sar2     sma1
##       1.8803  -0.9377  -1.7947  0.8666  -0.0250  -0.0979  -0.6360
## s.e.  0.0536   0.0512   0.0749  0.0669   0.1251   0.0980   0.1219
##
## sigma^2 = 0.002885:  log likelihood = 539.6
## AIC=-1063.21   AICc=-1062.8   BIC=-1032.14
```

ARIMA (2, 0, 2).

Así, el modelo ARIMA (2, 0, 2) indica que el modelo consta de un polinomio autorregresivo de orden 2, de una diferenciación en la variable de estudio Z_t de orden 0, y de un polinomio de promedios móviles de orden 2. Cabe recalcar que el modelo ha sido diferenciado anteriormente y estamos tomando dicho modelo diferenciado.

d) Estimación de parámetros y diagnóstico de residuales del modelo seleccionado:

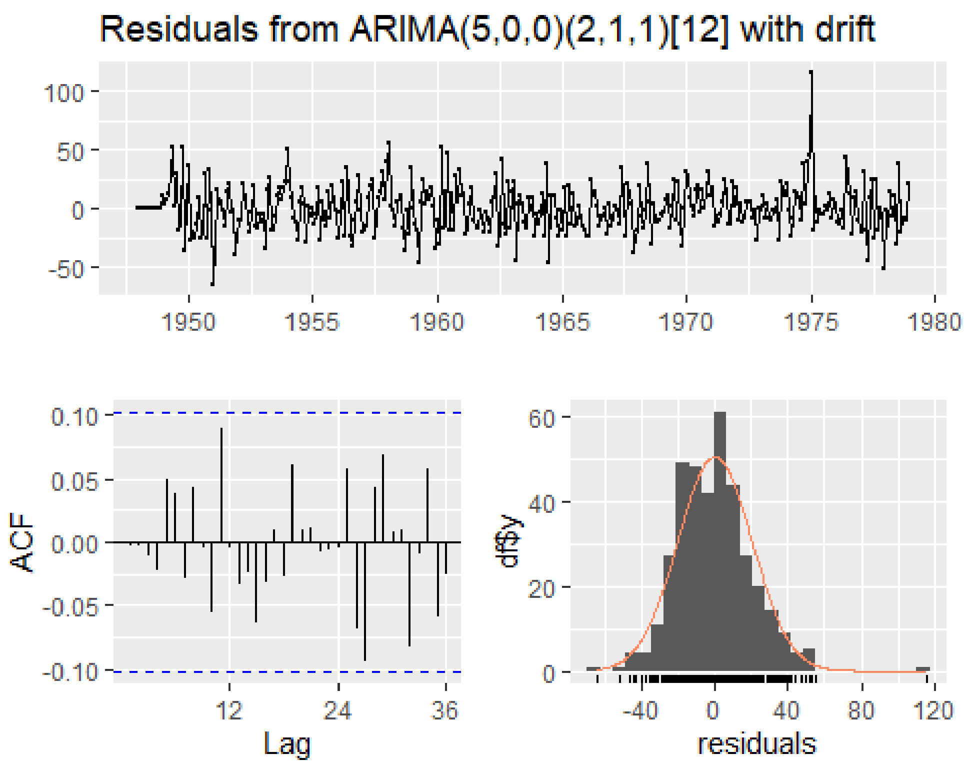
Haremos un comparativo entre el modelo sin transforma y el modelo transformado.

Usando la función auto.arima del paquete forecast para la serie sin transformar:

```
modelo_sin <- auto.arima(unemp)
modelo_sin

## Series: unemp
## ARIMA(5,0,0)(2,1,1)[12] with drift
##
## Coefficients:
##          ar1     ar2      ar3     ar4      ar5     sar1     sar2
sma1
##       1.1093  0.0739  -0.2046  0.0959  -0.1147  -0.1796  -0.1278  -
0.5318
## s.e.  0.0530  0.0806   0.0782  0.0794   0.0529   0.1155   0.0889
0.1076
##        drift
##       1.1571
## s.e.  0.8226
##
## sigma^2 = 437.9:  log likelihood = -1606.32
## AIC=3232.64   AICc=3233.27   BIC=3271.5
```

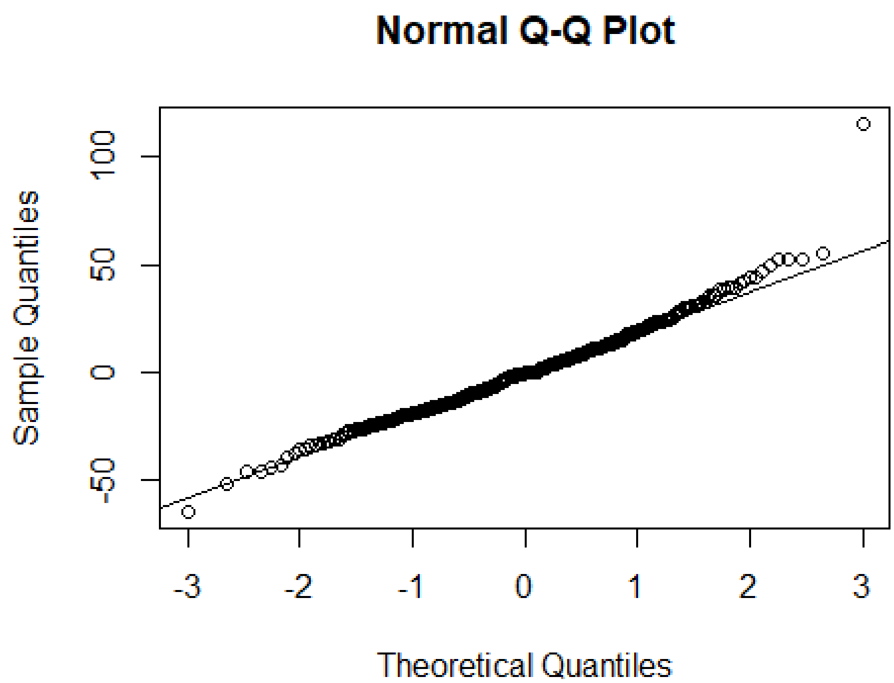
Analizando los residuales del modelo sin transformar:

```
modelo_sin %>% checkresiduals()
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(5,0,0)(2,1,1)[12] with drift
## Q* = 11.466, df = 16, p-value = 0.7799
##
## Model df: 8.   Total lags used: 24
```

```
modelo_sin$residuals %>% qqnorm()
modelo_sin$residuals %>% qqline()
```



Realizando prueba de Shapiro-Wilk para normalidad del modelo sin transformar:

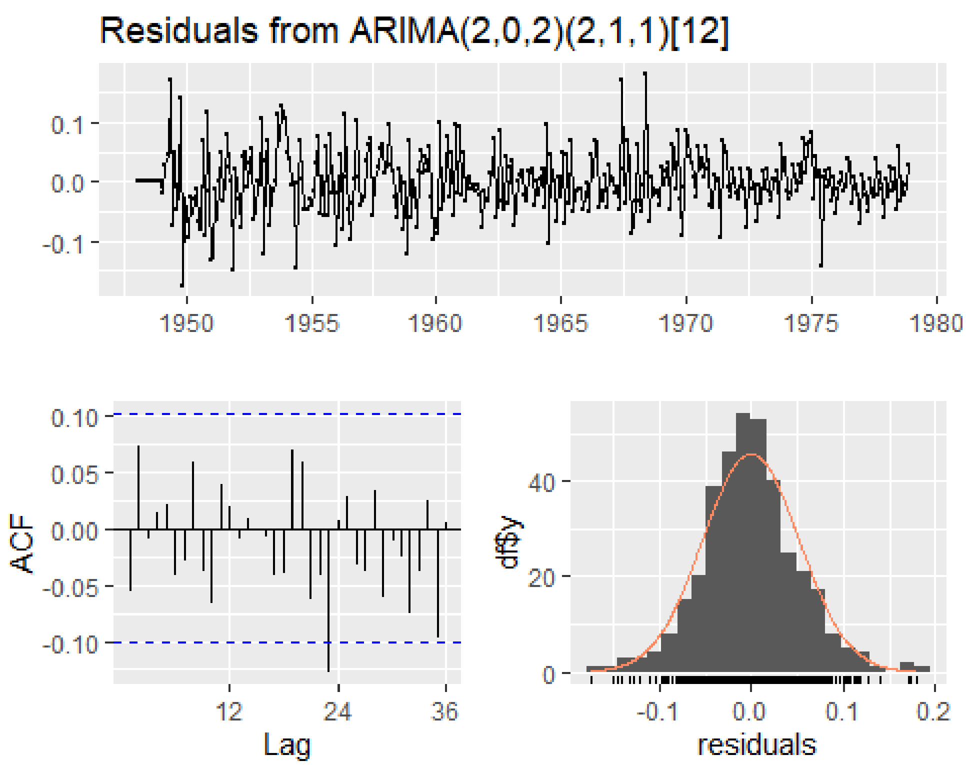
```
modelo_sin$residuals %>% shapiro.test()

##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.97581, p-value = 7.033e-06
```

Ahora, comprobemos lo mismo para el modelo transformado:

```
##arima(serie_transf, order=c(2, 0, 2))
modelo_con <- auto.arima(serie_transf)
modelo_con

## Series: serie_transf
## ARIMA(2,0,2)(2,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sar1     sar2     sma1
##       1.8803  -0.9377  -1.7947  0.8666  -0.0250  -0.0979  -0.6360
## s.e.  0.0536   0.0512   0.0749  0.0669   0.1251   0.0980   0.1219
##
## sigma^2 = 0.002885:  log likelihood = 539.6
## AIC=-1063.21   AICc=-1062.8   BIC=-1032.14
```

Analizando los residuales del modelo transformado:

```
modelo_con %>% checkresiduals(lag = 25)
```

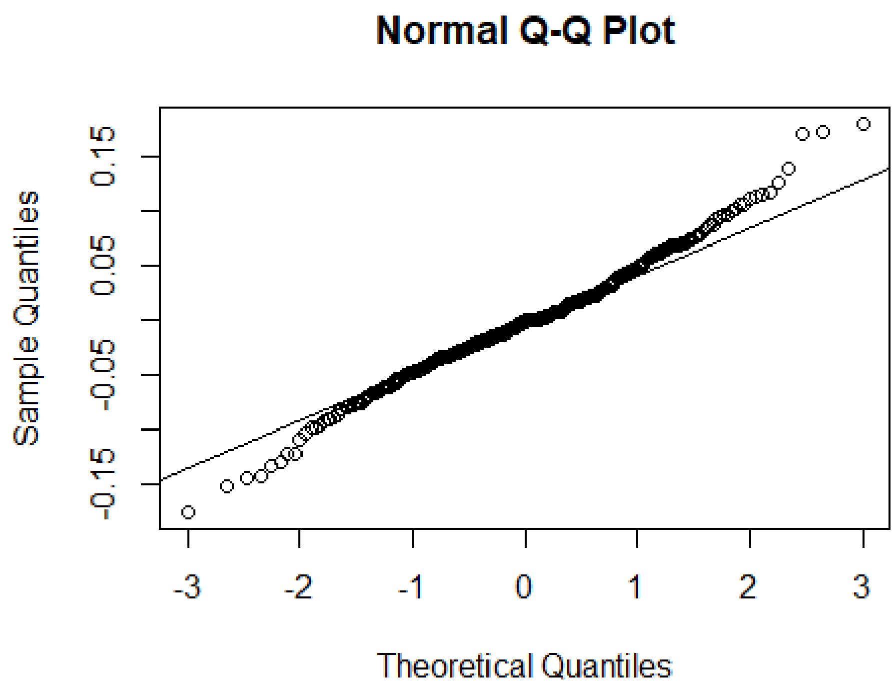



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,2)(2,1,1)[12]
## Q* = 22.414, df = 18, p-value = 0.2141
##
## Model df: 7.   Total lags used: 25
```

H_0: Los datos se distribuyen de forma independiente (es decir, las correlaciones en la población de la que se toma la muestra son 0, de modo que cualquier correlación observada en los datos es el resultado de la aleatoriedad del proceso de muestreo).

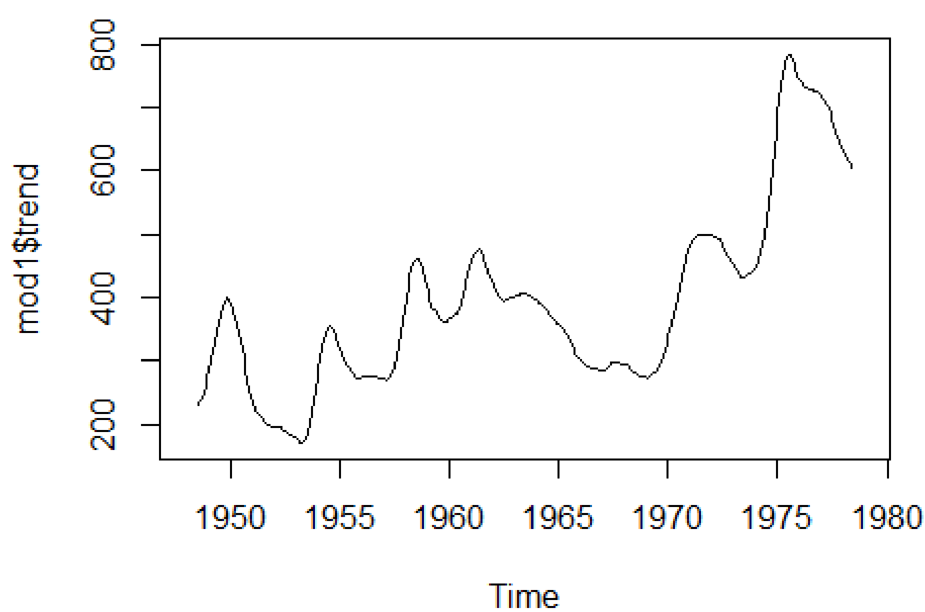
H_a: Los datos no se distribuyen de forma independiente.

```
modelo_con$residuals %>% qqnorm()
modelo_con$residuals %>% qqline()
```



Intento con modelo multiplicativo

```
mod1 <- decompose(unemp, type = "mult")
plot(mod1$trend)
```

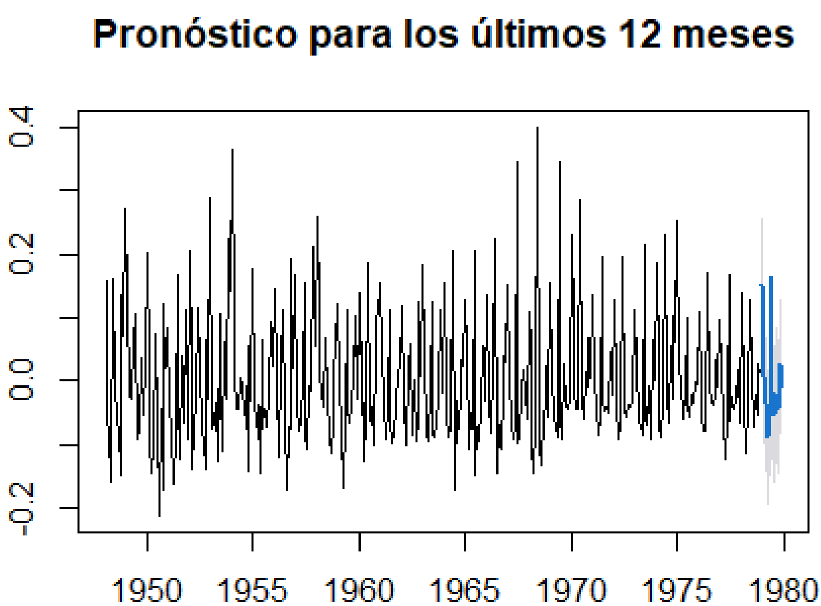


```
datos <- mod1$x
```

Punto 2:

Pronóstico para los últimos 12 meses.

```
pronostico <- forecast(modelo_con,12,level=95)

plot(pronostico, main="Pronóstico para los últimos 12 meses")
```



Ahora, miremos la matriz de pronósticos para los últimos 12 meses, donde exponemos la media y un intervalo de confianza al 95% de confianza para dichas predicciones:

```
matriz_pronosticos <-data.frame(pronostico$mean,pronostico$lower,
                                pronostico$upper)
matriz_pronosticos

##    pronostico.mean        X95.      X95..1
## 1      0.152549723  0.04727804 0.25782141
## 2      0.008883158 -0.09677383 0.11454015
## 3     -0.034581282 -0.14066128 0.07149871
## 4     -0.087671977 -0.19416254 0.01881858
## 5     -0.043913695 -0.15075840 0.06293101
## 6      0.163854944  0.05674142 0.27096846
## 7     -0.015690905 -0.12297855 0.09159674
## 8     -0.052692857 -0.16006977 0.05468405
## 9     -0.021262327 -0.12866838 0.08614373
```

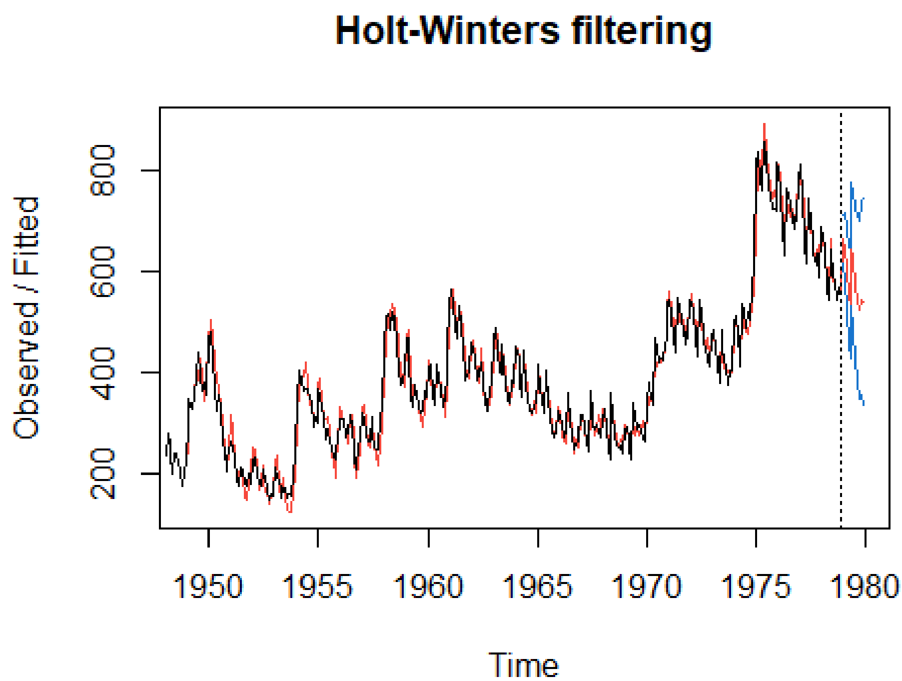
```
## 10    -0.041794881 -0.14920261 0.06561285
## 11     0.025833022 -0.08158162 0.13324766
## 12    -0.008263809 -0.11571634 0.09918872
```

Punto 3

```
#view(unemp)
```

A continuación, calculamos la previsión a 12 meses con un intervalo de confianza de 0,95 y trazamos el pronóstico junto con los valores reales y ajustados.

```
hw <- HoltWinters(unemp)

# Obtener los valores óptimos de alpha, beta y gamma
alpha_optimo <- hw$alpha
beta_optimo <- hw$beta
gamma_optimo <- hw$gamma

# Imprimir los valores óptimos
cat("Alpha óptimo:", alpha_optimo, "\n")

## Alpha óptimo: 0.8590901

cat("Beta óptimo:", beta_optimo, "\n")

## Beta óptimo: 0.07395703

cat("Gamma óptimo:", gamma_optimo, "\n")

## Gamma óptimo: 1

# Realizar el pronóstico
forecast <- predict(hw, n.ahead = 12, prediction.interval = TRUE, level =
0.95)

# Visualizar los resultados
plot(hw, forecast)
```



Podemos observar que los valores de α óptimo: 0.8590901, β óptimo: 0.07395703, γ óptimo: 1.

```
forecast <- forecast(hw, h = 12)
error <- accuracy(forecast)[2]
cat("Minimum Error:", error, "\n")

## Minimum Error: 24.28341

print(forecast)

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1979       662.7290 631.5763 693.8816 615.0851 710.3729
## Feb 1979       649.6001 607.2137 691.9865 584.7757 714.4245
## Mar 1979       609.5576 557.2086 661.9067 529.4967 689.6186
## Apr 1979       552.4326 490.7093 614.1560 458.0349 646.8303
## May 1979       538.2573 467.4584 609.0563 429.9798 646.5349
## Jun 1979       656.1012 576.3741 735.8282 534.1691 778.0332
## Jul 1979       625.7129 537.1163 714.3096 490.2161 761.2098
## Aug 1979       573.2993 475.8355 670.7631 424.2412 722.3573
## Sep 1979       549.8813 443.5153 656.2473 387.2085 712.5541
## Oct 1979       525.1113 409.7822 640.4405 348.7306 701.4921
## Nov 1979       545.0240 420.6524 669.3955 354.8141 735.2338
## Dec 1979       540.4772 406.9708 673.9836 336.2968 744.6577
```

En promedio, los pronósticos realizados por el modelo Holt-Winters difieren de los valores reales en aproximadamente 24.2834 unidades.

Punto 4

Ahora, para comparar el pronostico transformado y el pronostico utilizando Holt-winters, se realiza un summary en donde se arrojen los pronosticos RMSE, MAE Y MAPE, las cuales ayudaran a la hora de evaluar cual es mas preciso en terminos de error absoluto y error relativo.

a continuacion, los datos del pronostico transformado:

```
summary(pronostico)

##
## Forecast method: ARIMA(2,0,2)(2,1,1)[12]
##
## Model Information:
## Series: serie_transf
## ARIMA(2,0,2)(2,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sar1     sar2     sma1
##       1.8803  -0.9377  -1.7947  0.8666  -0.0250  -0.0979  -0.6360
## s.e.  0.0536   0.0512   0.0749  0.0669   0.1251   0.0980   0.1219
##
## sigma^2 = 0.002885:  log likelihood = 539.6
## AIC=-1063.21   AICc=-1062.8   BIC=-1032.14
##
## Error measures:
##                        ME       RMSE        MAE      MPE     MAPE
MASE
## Training set -0.001147539 0.05231761 0.03932498 50.59908 144.5645
0.7355426
##                     ACF1
## Training set -0.05528072
##
## Forecasts:
##          Point Forecast       Lo 95      Hi 95
## Jan 1979    0.152549723  0.04727804 0.25782141
## Feb 1979    0.008883158 -0.09677383 0.11454015
## Mar 1979   -0.034581282 -0.14066128 0.07149871
## Apr 1979   -0.087671977 -0.19416254 0.01881858
## May 1979   -0.043913695 -0.15075840 0.06293101
## Jun 1979    0.163854944  0.05674142 0.27096846
## Jul 1979   -0.015690905 -0.12297855 0.09159674
## Aug 1979   -0.052692857 -0.16006977 0.05468405
## Sep 1979   -0.021262327 -0.12866838 0.08614373
## Oct 1979   -0.041794881 -0.14920261 0.06561285
## Nov 1979    0.025833022 -0.08158162 0.13324766
## Dec 1979   -0.008263809 -0.11571634 0.09918872
```

luego, los datos del pronostico utilizando Holt-winters

```
summary(forecast)

##
## Forecast method: HoltWinters
##
## Model Information:
## Holt-Winters exponential smoothing with trend and additive seasonal
component.
##
## Call:
## HoltWinters(x = unemp)
##
## Smoothing parameters:
##  alpha: 0.8590901
##  beta : 0.07395703
##  gamma: 1
##
## Coefficients:
##           [,1]
## a   605.764372
## b    -2.668564
## s1   59.633164
## s2   49.172878
## s3   11.798946
## s4  -42.657477
## s5  -54.164206
## s6   66.348167
## s7   38.628508
## s8  -11.116599
## s9  -31.865985
## s10 -53.967401
## s11 -31.386199
## s12 -33.264372
##
## Error measures:
##                      ME     RMSE      MAE         MPE     MAPE
MASE
## Training set -0.6473912 24.28341 19.04696 -0.05377982 5.344272
0.261982
##                   ACF1
## Training set 0.2805933
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1979       662.7290 631.5763 693.8816 615.0851 710.3729
## Feb 1979       649.6001 607.2137 691.9865 584.7757 714.4245
## Mar 1979       609.5576 557.2086 661.9067 529.4967 689.6186
## Apr 1979       552.4326 490.7093 614.1560 458.0349 646.8303
## May 1979       538.2573 467.4584 609.0563 429.9798 646.5349
## Jun 1979       656.1012 576.3741 735.8282 534.1691 778.0332
```

```
## Jul 1979       625.7129 537.1163 714.3096 490.2161 761.2098
## Aug 1979       573.2993 475.8355 670.7631 424.2412 722.3573
## Sep 1979       549.8813 443.5153 656.2473 387.2085 712.5541
## Oct 1979       525.1113 409.7822 640.4405 348.7306 701.4921
## Nov 1979       545.0240 420.6524 669.3955 354.8141 735.2338
## Dec 1979       540.4772 406.9708 673.9836 336.2968 744.6577
```

Resultado para el pronostico transformado incialmente RMSE= 0,05231761, MAE= 0,03932498, MAPE= 144,5645

Resultado para el pronostico utilizando Holt-winters RMSE= 24,28341, MAE= 19,04696, MAPE=5.344272

según los datos arrojados para ambos pronosticos, el que resulta ser mas preciso es el pronostico con transformación inicialmente propuesto, pues el RMSE, el MAE y el MAPE son menores y es considerado mas preciso en terminos de estimacion del valor real en comparacion al pronostico Holt- Winters, es por esa razon, que se decide seguir utilizando el modelo inicial.

Taller 4

```
library(forecast)
datos <- read.csv("../../Trabajo 2/Trabajo2/Serie01_We_02_W6.csv", sep =
";")
datos_ts = ts(datos) # Conversión a serie de tiempo
mod1_CSS_ML=Arima(datos_ts, c(0, 1, 1), include.drift=TRUE, lambda=.43,
method = c("CSS-ML"))
datos_ts = ts(datos^.43)
```

Punto A

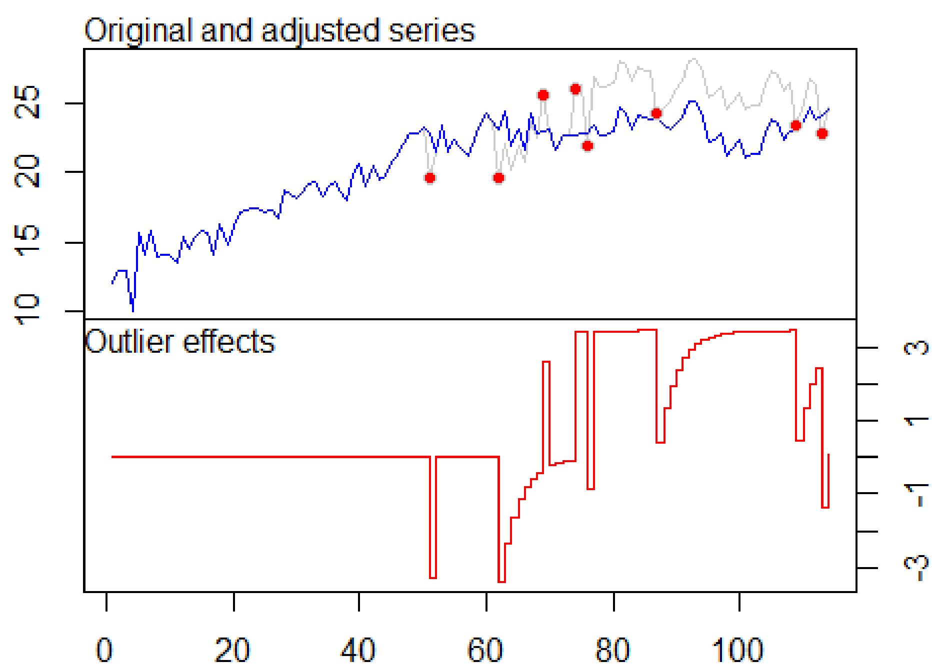
```
library(tsoutliers)

## Warning: package 'tsoutliers' was built under R version 4.2.3

tso(datos_ts)

## Series: datos_ts
## Regression with ARIMA(1,1,0) errors
##
## Coefficients:
##           ar1     AO51     TC62     AO69     LS74     AO76     TC87
TC109
##       -0.4841  -3.2944  -3.3962  2.8936  3.4619  -4.3085  -3.0722  -
3.0157
## s.e.   0.0827   0.8340   0.9008  0.8304  0.9155   0.8319   0.8988
0.9180
##         TC113
##       -4.1011
## s.e.   1.0153
##
```

```
## sigma^2 = 1.124:  log likelihood = -162.39
## AIC=344.77   AICc=346.93   BIC=372.05
## 
## Outliers:
##   type ind time coefhat  tstat
## 1   AO  51   51  -3.294 -3.950
## 2   TC  62   62  -3.396 -3.770
## 3   AO  69   69   2.894  3.484
## 4   LS  74   74   3.462  3.781
## 5   AO  76   76  -4.309 -5.179
## 6   TC  87   87  -3.072 -3.418
## 7   TC 109  109  -3.016 -3.285
## 8   TC 113  113  -4.101 -4.039
```

Se detectan outliers de tipo AO, TC Y LS

Punto B

```
modT4 <- tso(datos_ts)
plot.tsoutliers(modT4)
```



Se notan algunos cambios bruscos y pareciera que tuvieran una duración permante en la serie los outliers, ya que estos no coinciden en la tendencia y en la variabilidad de la misma, estos datos atípicos se ven que influencia la serie en la parte final donde se concentran un poco más estos.