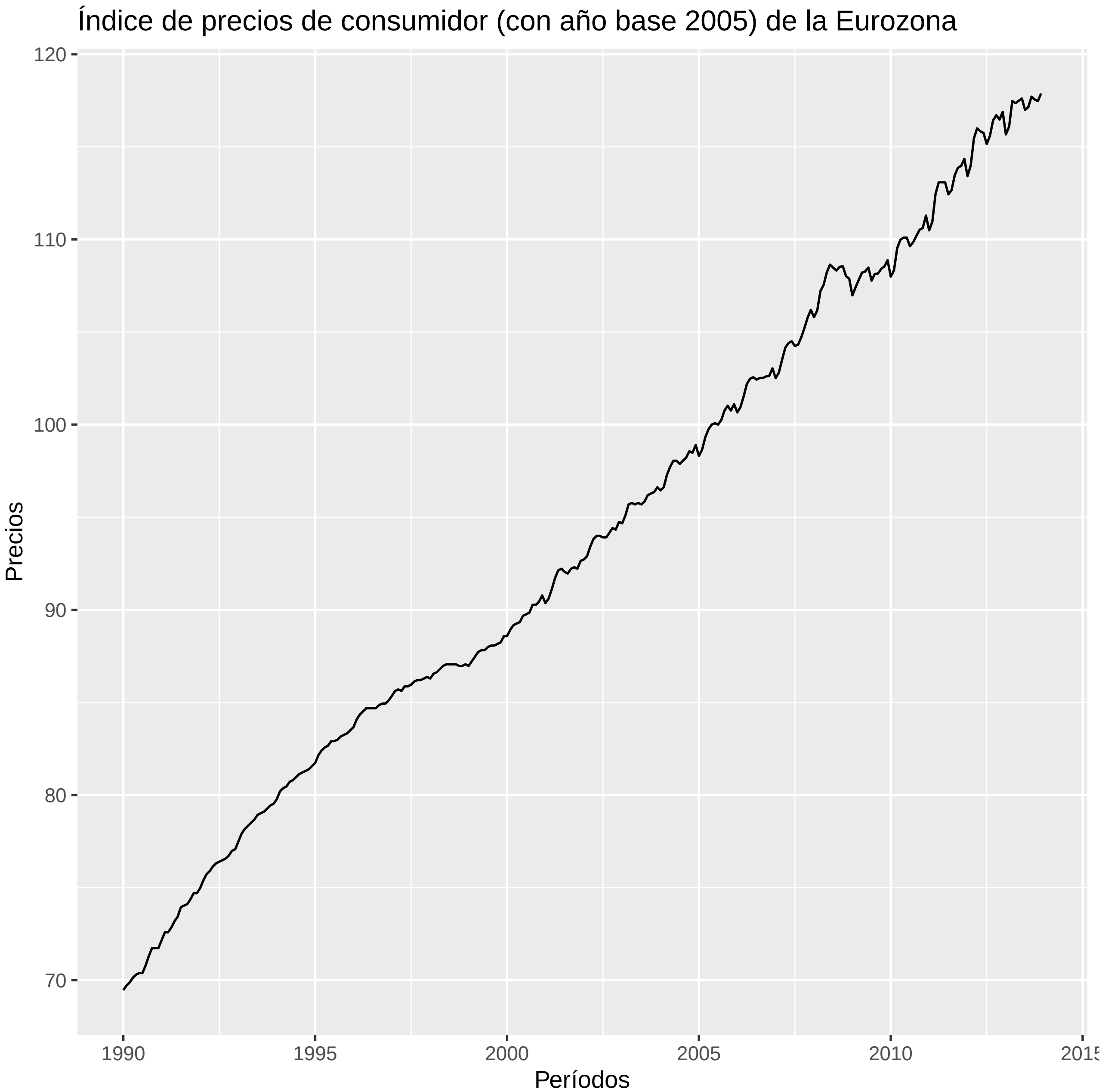
1. Choques y efectos externos: Aplicación de los modelos ARIMAX y SARIMAX

Al igual que con el modelo SARIMA, la mejor forma de abordar el temaa de los modelos ARIMAX es con un sencillo ejemplo. Primero, empezaremos a cargar las paqueterías que se ocuparan para estimar el modelo que ocuparemos. La serie de tiempo que ocuparemos para este ejemplo se llama "hicp", y se trata del índice de precios del consumidor de la Eurozona (considerando como año base el 2006). Utilizaremos esta serie como ejemplo debido a que las series temporales de tipo económicas normalmente se encuentran influenciadas por choques externos que afectan el comportamiento de la serie. De igual manera, realizaremos de manera rápida una gráfica sobre la serie para observar de mejor manera las características de la misma utilizando la función *autoplot()*.

```
library(tsoutliers)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

library(tseries)
library(forecast)
library(ggplot2)
data("hicp")
Precios <- hicp[[1]]
autoplot(Precios)+
  labs(title = "Índice de precios de consumidor (con año base 2005) de la Eurozona",x="Períodos"
```



A simple vista podemos observar algunas características importantes: La serie presenta una tendencia creciente, prácticamente bajo una relación lineal entre la serie y el paso del tiempo. Otra característica importante es que existe un importante aumento en el índica por el 2008-2009 (el cual podría explicarse fácilmente por el efecto de la crisis financiera del 2008). Este efecto podría impedir las estimaciones del modelo sean buenas, ya que es un choque externo que ocasiona el comportamiento de la serie sea diferente al que debería de haber sido.
Para estimar el modelo, procedemos a utilizar la metodología ***Box-Jenkins***. Primero nprocedereos a reaalizar la prueba de estacionariedad de *Dickey-Fuller Aumentada y Phillips-Perron* (las cuales es casi seguro que ambas indiquen que la serie NO es estacionaria, debido a que existe una tendencia casi perfecta entre la serie y el paso del tiempo). Para realizar la *Prueba Dickey-Fuller Aumentada* utilizamos el siguiente código:

```
adf.test(Precios,alternative = "stationary", k=0 )

##
##  Augmented Dickey-Fuller Test
##
## data:  Precios
```

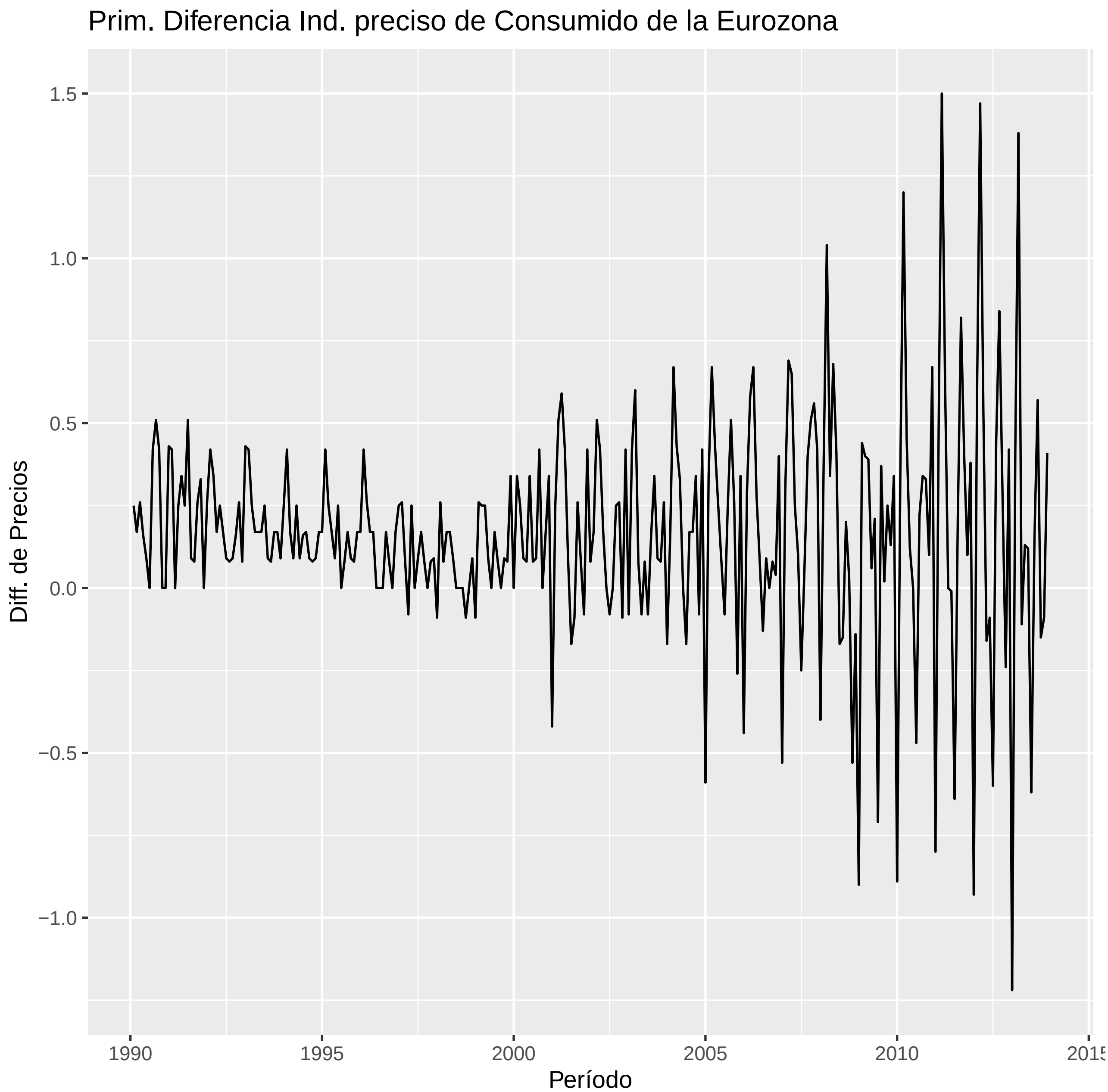
```
## Dickey-Fuller = -2.7611, Lag order = 0, p-value = 0.2555
## alternative hypothesis: stationary
```

La *Prueba Dickey-Fuller Aumentada,* con un estadístico de -2.7611 y un p-valor del 0.2555, hay evidencia estadística de NO rechazar la hipótesis nula, por lo que podemos suponer que la serie NO es estacionaria (taal cual ya habíamos intuido debido a la existencia de la tendencia). Ahora, procedemos a realizar la prueba *Phillips-Perron* utilizando el siguiente código:

```
pp.test(Precios , alternative = "stationary")

##
##  Phillips-Perron Unit Root Test
##
## data:  Precios
## Dickey-Fuller Z(alpha) = -9.5386, Truncation lag parameter = 5, p-value
## = 0.5754
## alternative hypothesis: stationary
```

La *Prueba Phillips-Perron* con un estadístico de -9.5385 y un p-valor del 0.5759, existe evidencia estadística para NO rechazar la hipótesis nula, por lo que podemos suponer que la serie NO es estacionaria. El resultado de la mano del estadístico *Dickey-Fuller*, nos da certidumbre de que la serie, en efecto, NO es estacionaria. El siguiente paso, es hacer que la serie sea estacionaria, por lo que procedemos a realizar una transformación, en este caso, aplicar a primera diferencia de la serie, con el fin de eliminar la tendencia estocástica que existe en la serie. Podemos ver la nueva diferencia utilizando el siguiente código:

```
autoplot(diff(Precios))+
  labs(title = "Prim. Diferencia Ind. preciso de Consumido de la Eurozona",
       x="Período", y="Diff. de Precios")+theme_gray()
```



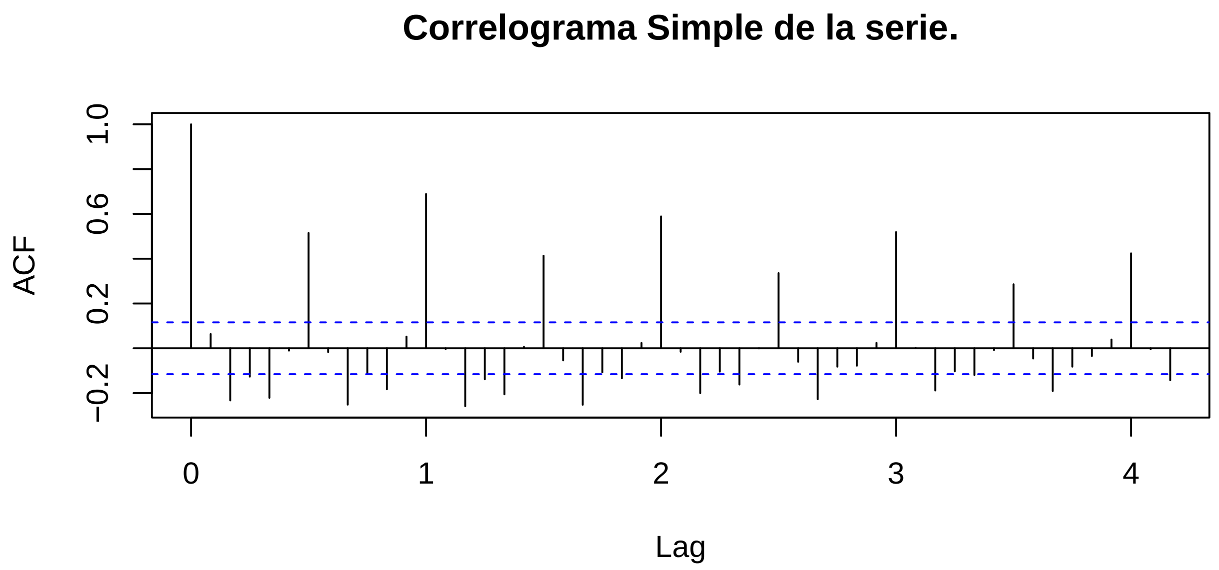
Podemos observar que la nueva serie ya no presenta la tendencia estocástica; sin embargo, podemos observar que si presenta importantes variaciones respecto a su media en los últimos períodos. Este tipo de comportamiento nos indica que la serie tuvo un importante cambio en su comportamiento normal a causa de algún factor o suceso que el modelo no está considerando. También puede indicarnos que existe un período de volatilidad en el período. Conforme avancemos con la estimación, podemos observar de mejor manera cual podría ser la causa de dicho problema. Procederemos a continuar con los pasos de la metodología; comprobaremos de igual manera que la nueva serie diferenciada sea estacionaria realizando las *Pruebas de Dickey-Fuller Aumentada y Phillips-Perron*. Los códigos que ocuparemos para ambas pruebas son las siguientes:

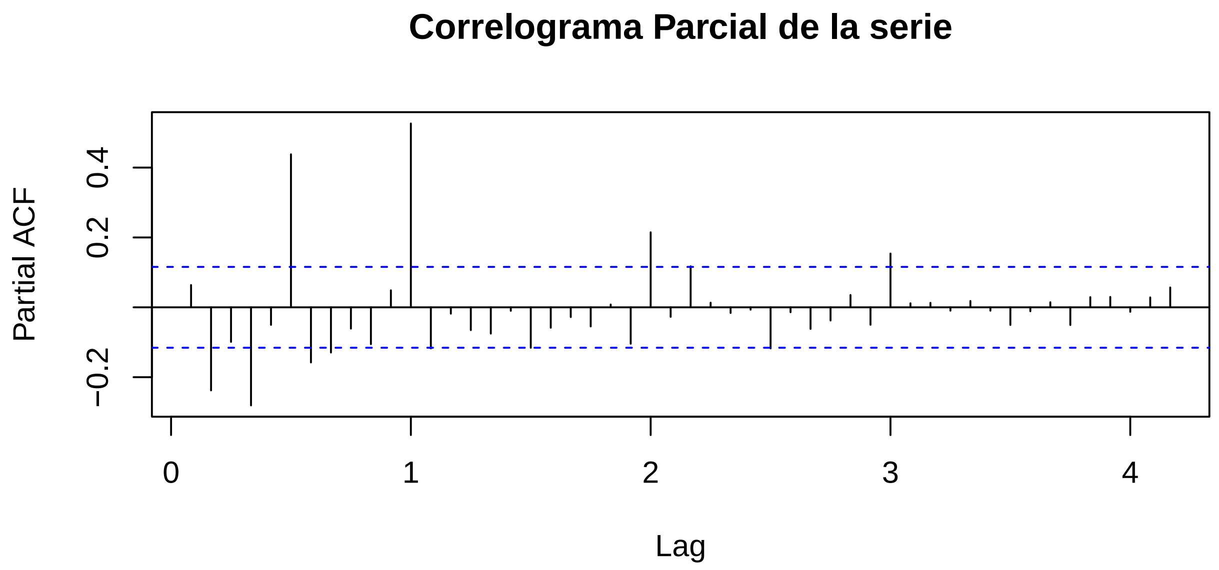
```
adf.test(diff(Precios),alternative = "stationary", k=0)

## Warning in adf.test(diff(Precios), alternative = "stationary", k = 0): p-value smaller
than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  diff(Precios)
## Dickey-Fuller = -15.766, Lag order = 0, p-value = 0.01
```

```
## alternative hypothesis: stationary
```

Con p-valor de 0.01 y 0.01, ambos menores al nivel de significancia del 0.05, rechazamos la hipótesis nula de NO estacionariedad; confirmando que la nueva serie diferenciada es estacionaria. Procedemos a confirmar la naturaleza de la serie, y ver que tipo de modelo podemos ocupar para la serie, utilizando los correlogramas. Para realizar el correlograma simple y parcial de la serie, utilizaremos el siguiente código:

```
layout(1:2)
acf(diff(Precios), 50, main ="Correlograma Simple de la serie.")
pacf(diff(Precios), 50, main = "Correlograma Parcial de la serie" )
```



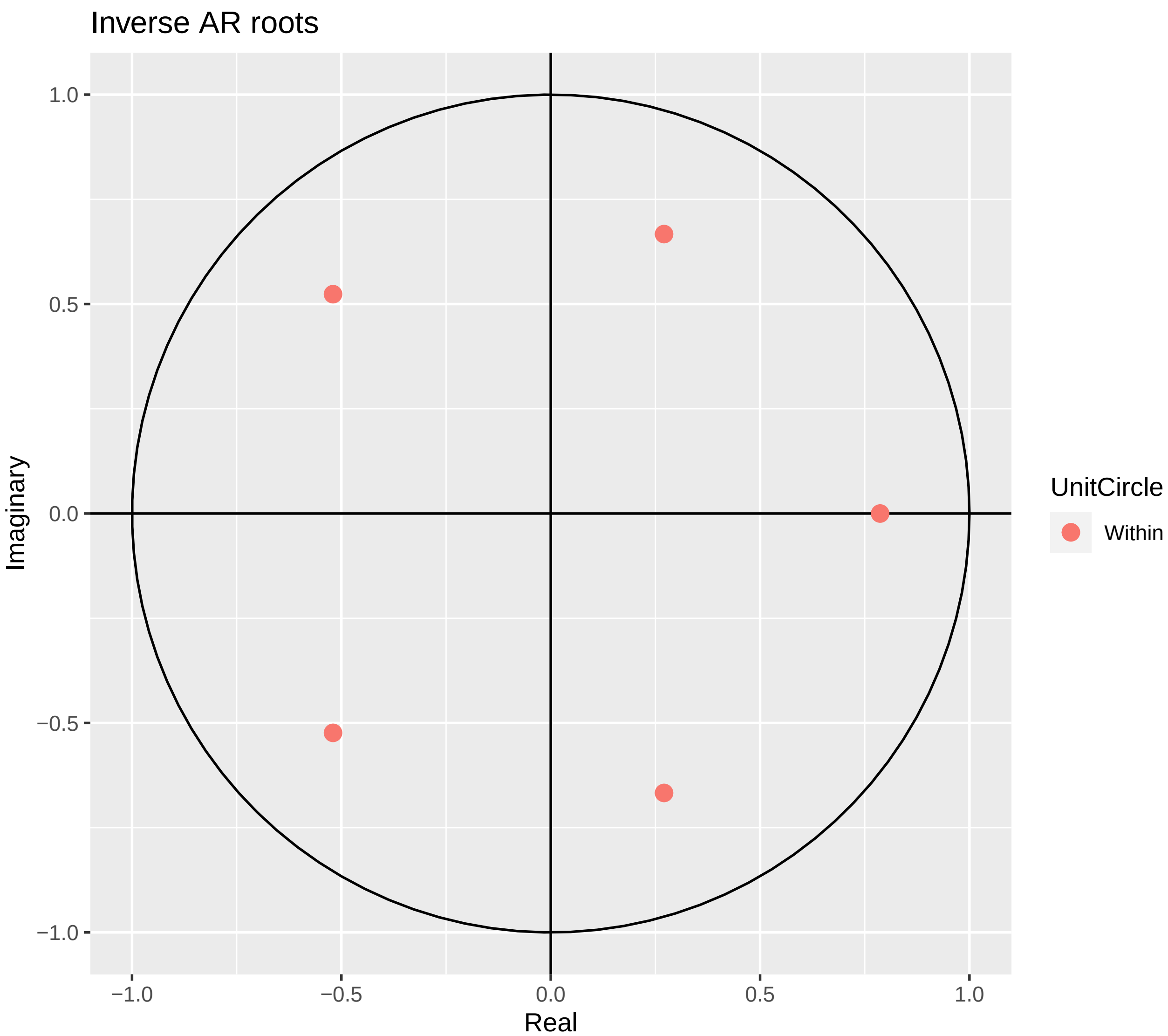


Podemos observar que el correlograma simple presenta un comportamiento muy significativo en el componente auto-regresivo (AR) de la serie; por otro lado, no parece que existan efectos considerables en el comportamiento de Media Móvil (MA). Procederemos a realizaar unas 6 propuestas de modelos (donde mayoritariamente utilizaremos procesos auto-regresivos en ves de proceso MA debido al peso que indican los correlogramas sobre el componente auto-regresivo) y utilizaremos los *criterios de información de Akaike* para confirmar el modelo que mejor se adecue al comportamiento de la serie con base en los errores de estimación. Para realizar todos los modelos y comparar los criterios de información podemos utilizar el siguiente código:

```
#Calculamos los modelos ARIMA propuestos
ModAPre_ARIMA <- arima(Precios, order = c(1,1,0))
ModBPre_ARIMA <- arima(Precios, order = c(2,1,0))
ModCPre_ARIMA <- arima(Precios, order = c(1,1,1))
ModDPre_ARIMA <- arima(Precios, order = c(3,1,0))
ModEPre_ARIMA <- arima(Precios, order = c(4,1,0))
ModFPre_ARIMA <- arima(Precios, order = c(5,1,0))
ModAPre_ARIMA$aic

## [1] 220.5501

ModBPre_ARIMA$aic

## [1] 222.0049

ModCPre_ARIMA$aic

## [1] 220.4274

ModDPre_ARIMA$aic

## [1] 219.5403

ModEPre_ARIMA$aic

## [1] 221.3298

ModFPre_ARIMA$aic

## [1] 208.8321
```

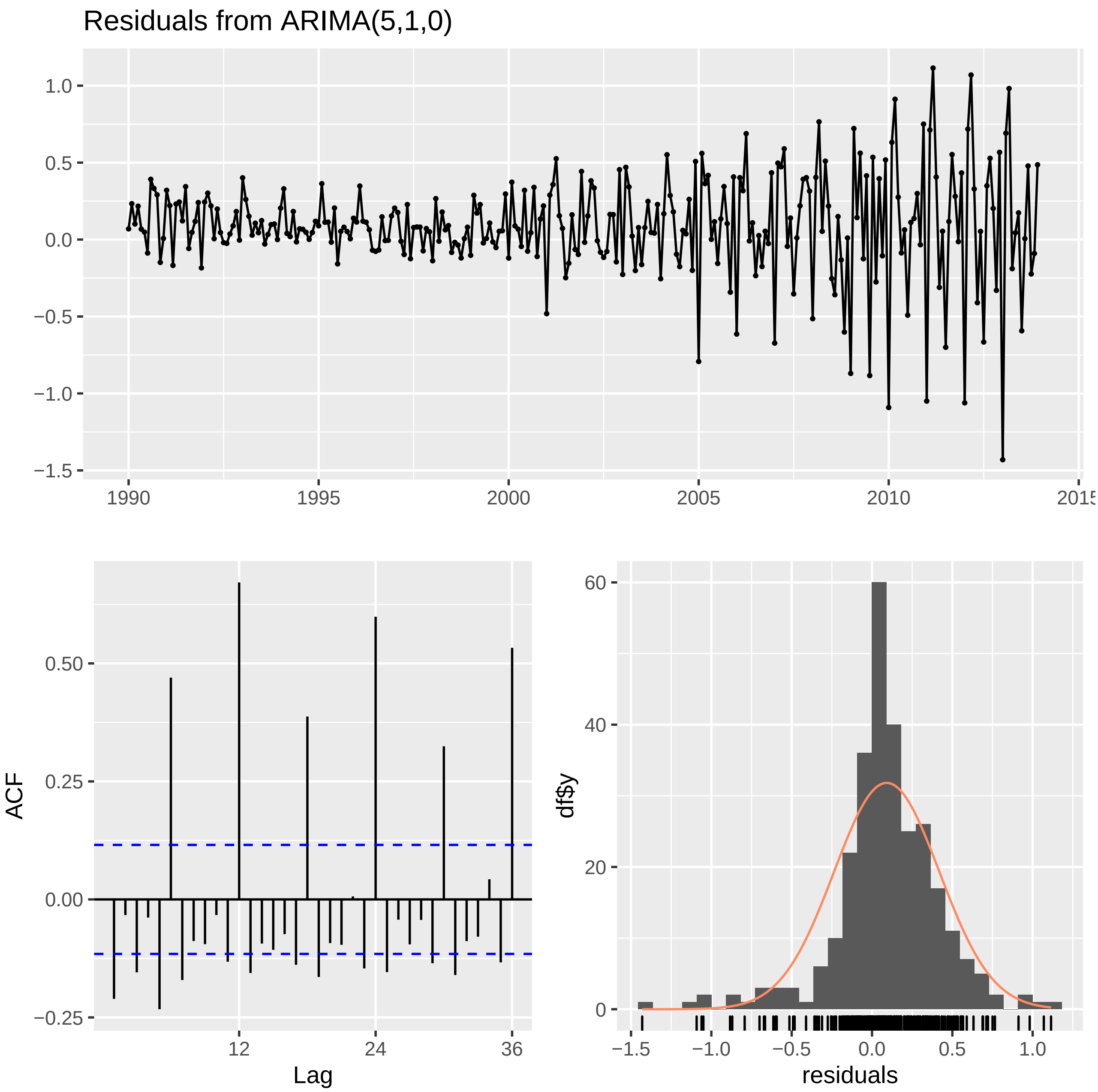
Confirmamos que el modelo que mejor se aadecua de acuerdo a los errores de estimación es el modelo ARIMA(5,1,0). Tal cual como nos indicó el correlograma Simple; existe un predominante comportamiento auto-regresivo en la serie. Utilizaremos este modelo como el indicado para empezar a realizaer las pruebas sobre las especificaciones y sobre el análisis del cumplimiento de los supuestos de los errores. Primero, verificaremos que el modelo sea estable en los parámetros estimados utilizando el siguiente código

```
autoplot(ModFPre_ARIMA)+theme_gray()
```



Podemos confirmar con la prueba de raíces unitarias que las raíces de las estimaciones de los parámetros del modelo son estables, de tal manera que los pronósticos que se realizaran con dicho modelo serán estables y no se alejaran de la media estimada, por lo que procederemos a realizar el análisis sobre el cumplimiento de los supuestos de los errores del modelo. Como un primer acercamiento, podemos utilizar la función *checkresiduas()* para tener un primer vistazo sobre el comportamiento general de los errores del modelo, tal y como lo haremos con el siguiente código:

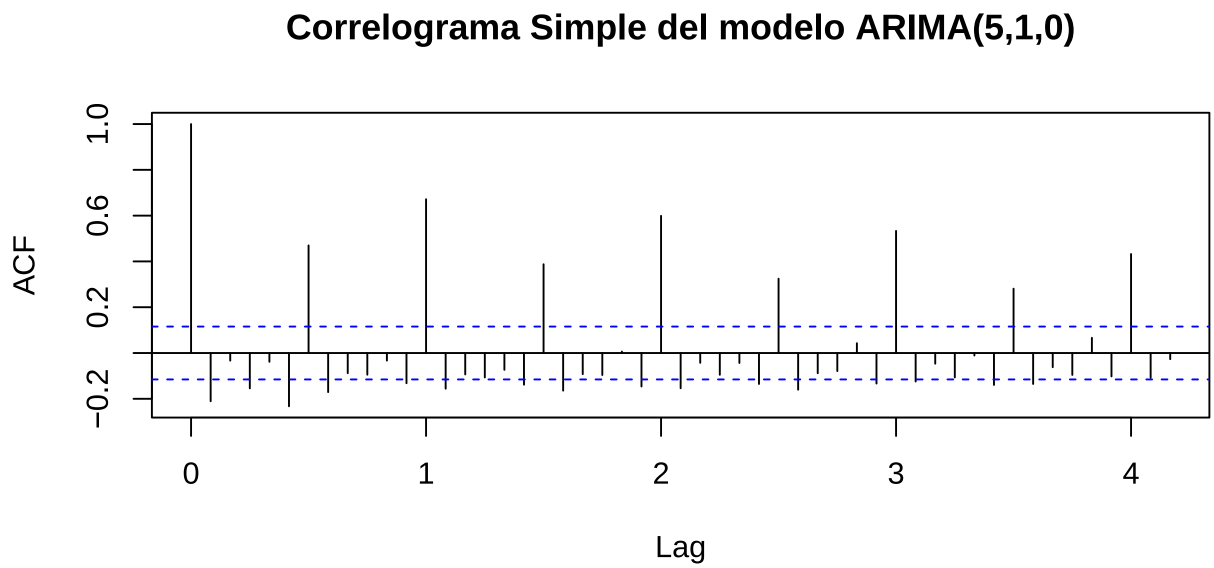
```
checkresiduals(ModFPre_ARIMA)
```

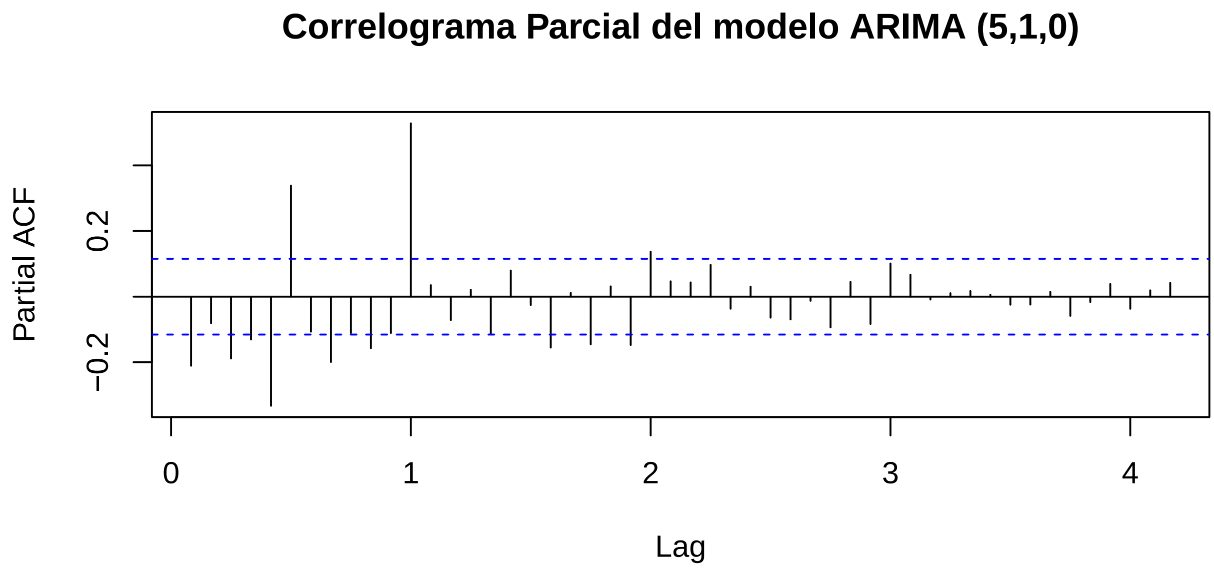



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(5,1,0)
## Q* = 459.5, df = 19, p-value < 2.2e-16
##
## Model df: 5.   Total lags used: 24
```

Podemos observar que la serie de los errores de estimación del modelo presentan, de acuerdo con el histograma, un comportamiento parecido al normal, y que este puede ser explicado por el comportamiento estacional que presenta la serie (ya que el correlograma simple indica que existe un comportamiento estacional en el componente autor-regresivo del modelo). Por otra parte, la gráfica temporal de los errores indica que aún existe una importante variación en el comportamiento de la serie que existe después del 2008, y que aun estimando el modelo no ha logrado corregir; esto quiere decir que el modelo aun no considera este efecto o este factor externo en dicha serie. Para confirmar un poco mejor la idea de la existencia de un comportamiento estacional en la serie, y por lo tanto intentar estimar un modelo SARIMA, procedemos a realizar los correlogramas de los errores utilizando el siguiente código:

```
layout(1:2)
acf(ModFPre_ARIMA$residuals , 50, main = "Correlograma Simple del modelo ARIMA(5,1,0)")
pacf(ModFPre_ARIMA$residuals , 50, main = "Correlograma Parcial del modelo ARIMA (5,1,0)" )
```





Ambos correlogramas confirman que existe un comportamiento estacional en ambos componentes del modelo (tanto del proceso AR como del proceso MA) por lo que es necesario empezar a estimar un modelo SARIMA que considere estos comportamientos en la serie original. Primero, veremos si la serie necesita diferenciarse de manera estacional sobre la serie utilizando la función *nsdiffs()* con el siguiente código:

```
nsdiffs(Precios)

## [1] 1
```

Podemos confirmar que la serie necesita diferenciarse estacionalmente una vez. de tal forma que existe una tendencia estacional que afecta el comportamiento de la serie y tenemos que eliminarla para realizar estimaciones significativas sobre los valores futuros de la serie. Procederemos a realizar varias propuestas de modelos SARIMA (nosotros consideremos al menos unos 9 (modelos) que consideren el comportamiento estacional que presenta la serie del Índice de precios del consumidor de la Eurozona, y a su vez, procederemos a utilizar los *Criterios información de Akaike* para seleccionar el mejor modelo de acuerdo a los errores. Para realizar todos los pasos, podemos ocupar el siguiente código:

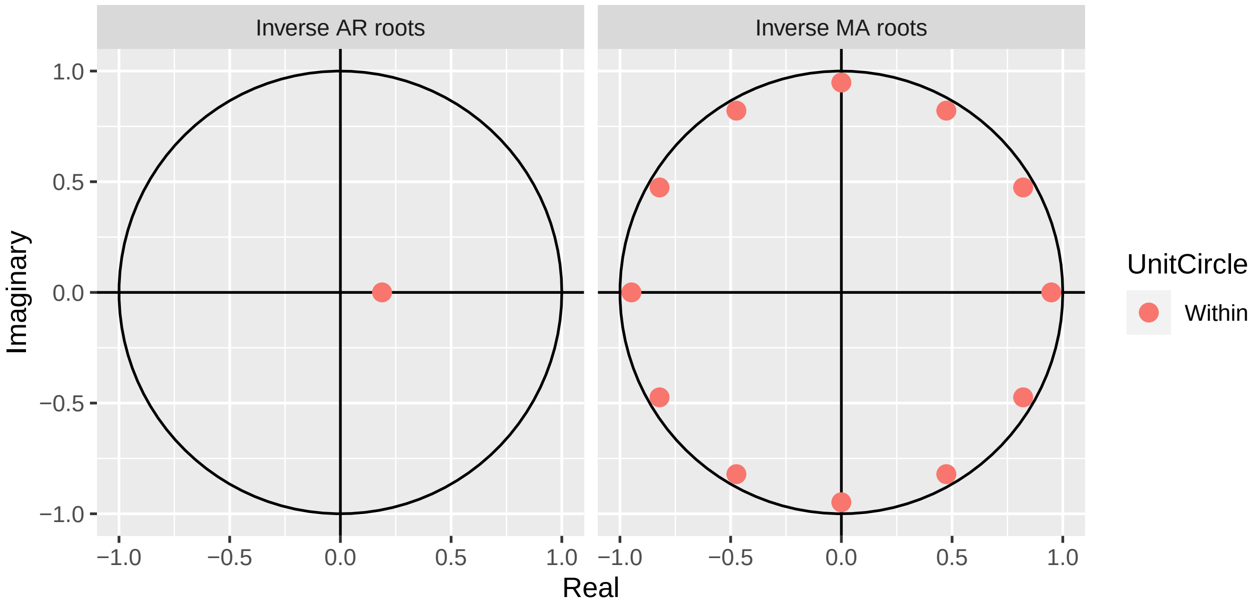
```
#Los modelos seleecionamos son los siguientes:
SarAPre_Mod <- arima(Precios ,order = c(5,1,0), seasonal = list(order = c(1,1,1)) )
SarBPre_Mod <- arima(Precios ,order = c(5,1,0), seasonal = list(order = c(1,1,0)) )
SarCPre_Mod <- arima(Precios ,order = c(5,1,0), seasonal = list(order = c(0,1,1)) )
SarDPre_Mod <- arima(Precios ,order = c(3,1,0), seasonal = list(order = c(1,1,1)))
SarEPre_Mod <- arima(Precios ,order = c(3,1,0), seasonal = list(order = c(1,1,0)) )
SarFPre_Mod <- arima(Precios ,order = c(3,1,0), seasonal = list(order = c(0,1,1)) )
SarGPre_Mod <- arima(Precios ,order = c(1,1,0), seasonal = list(order = c(1,1,1)) )
SarHPre_Mod <- arima(Precios ,order = c(1,1,0), seasonal = list(order = c(1,1,0)) )
SarIPre_Mod <- arima(Precios ,order = c(1,1,0), seasonal = list(order = c(0,1,1)) )
```

Los critetrios de información son los siguientes:

```
SarAPre_Mod$aic

## [1] -111.323

SarBPre_Mod$aic

## [1] -98.7062

SarCPre_Mod$aic

## [1] -111.2368

SarDPre_Mod$aic

## [1] -113.8136

SarEPre_Mod$aic

## [1] -98.63429

SarFPre_Mod$aic

## [1] -113.8789

SarGPre_Mod$aic

## [1] -117.5969

SarHPre_Mod$aic

## [1] -102.3711

SarIPre_Mod$aic

## [1] -117.7129
```

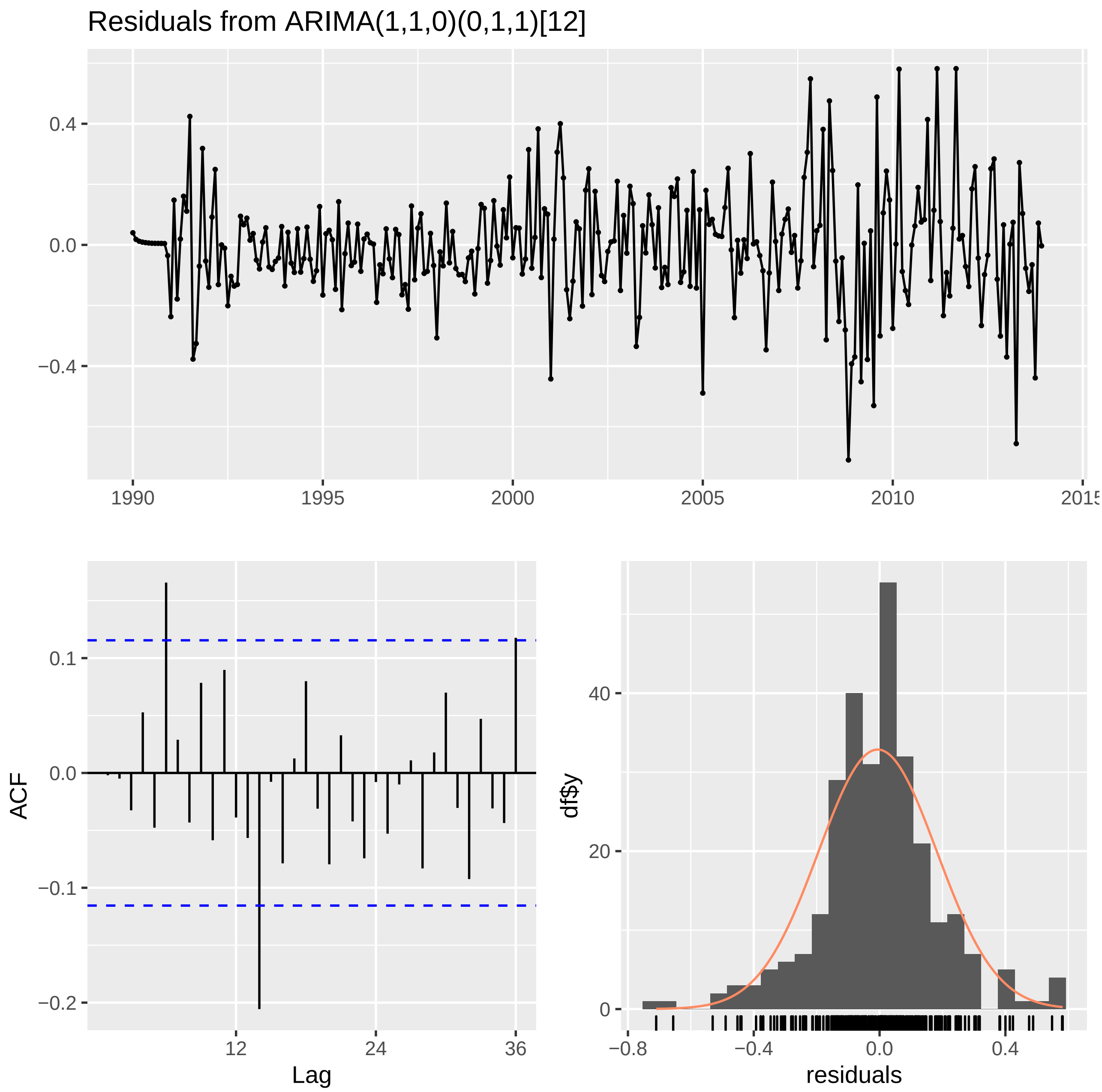
Podemos confirmar que la serie necesita diferenciarse estacionalmente una vez. de tal forma que existe una tendencia estacional que afecta el comportamiento de la serie y tenemos que eliminarla para realizar estimaciones significativas sobre los valores futuros de la serie. Procederemos a realizar varias propuestas de modelos SARIMA (nosotros consideremos al menos unos 9 (modelos) que consideren el comportamiento estacional que presenta la serie del Índice de precios del consumidor de la Eurozona, y a su vez, procederemos a utilizar los *Criterios información de Akaike* para seleccionar el mejor modelo de acuerdo a los errores. Para realizar todos los pasos, podemos ocupar el siguiente código:

```
autoplot(SarIPre_Mod)+theme_gray()
```



Tal parece que los parámetros estimados del modelo, tanto de los componentes AR y MA como de los componentes SAR y SMA, son estables a lo largo del período comprendido; por lo que las estimaciones futuras del modelo convergerán a una media y no tendrá un comportamiento explosivo. Procederemos a realizar el análisis sobre el cumplimiento de los supuestos de los errores de predicción del modelo. Como un primer acercamiento, utilizaremos la función *checkresiduals()* para visualizar de manera rápida el cumplimiento de algunos de los supuestos sobre los residuos. Para esto podemos utilizar el siguiente código:

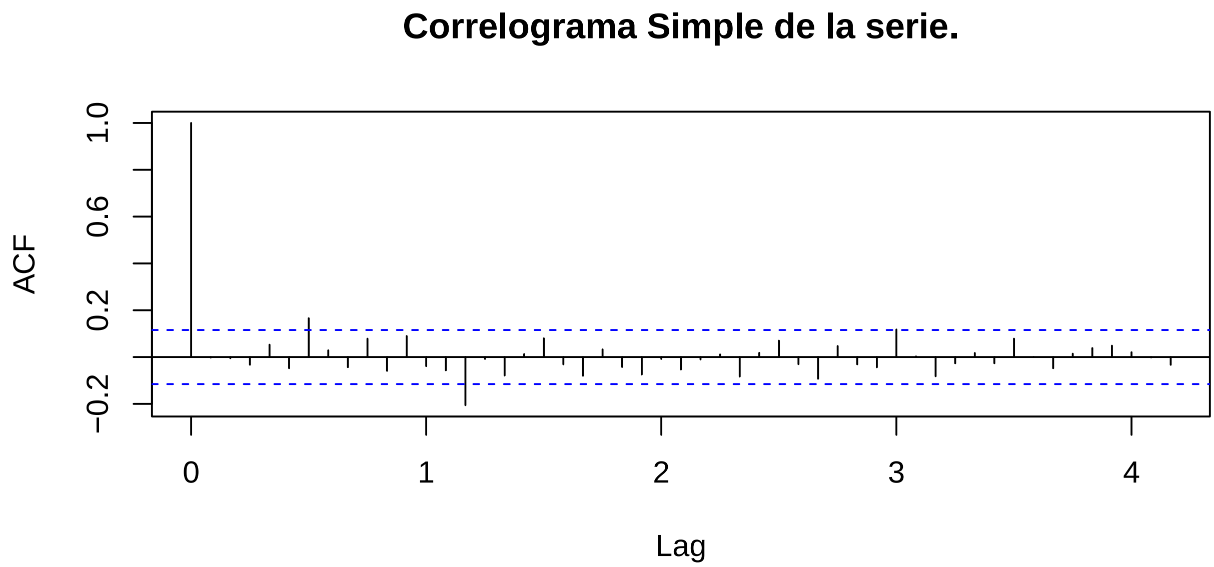
```r
checkresiduals(SarIPre_Mod)
```

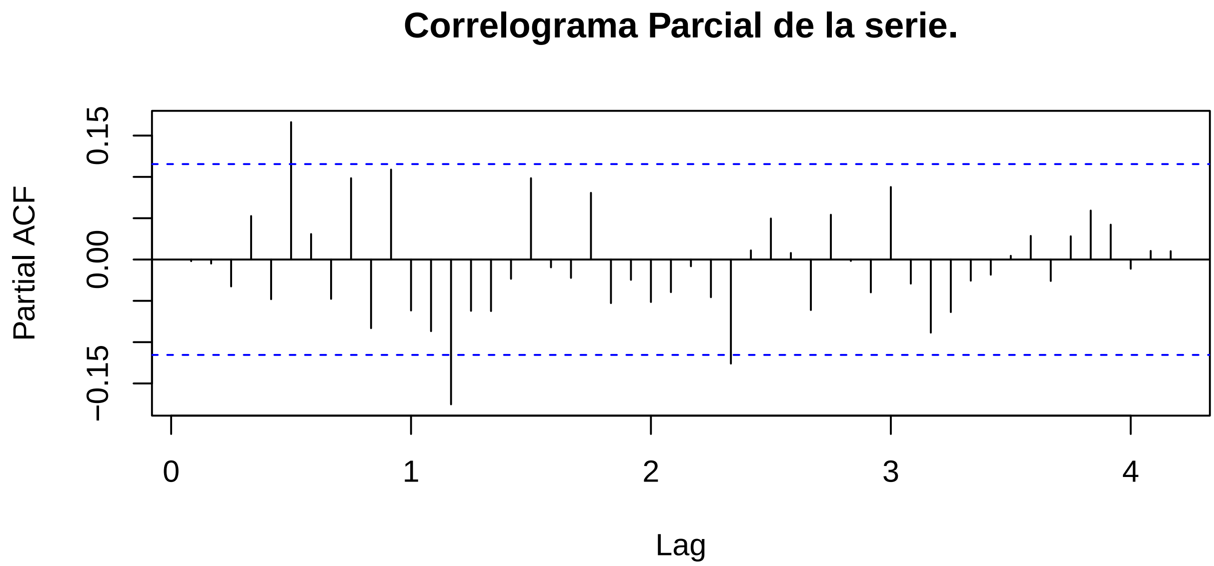



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)(0,1,1)[12]
## Q* = 39.24, df = 22, p-value = 0.01325
##
## Model df: 2.   Total lags used: 24
```

A simple vista podemos observar algunos factores importantes a considerar primero, que la serie de los errores presentan un comportamiento más cercano al normal que en los anteriores modelos, sin embargo, aún no se podría considerar como normal debido a algunos picos en el histograma. Por otra parte, también podemos confirmar que aún existen factores que no estamos considerando debido a que el correlograma simple indica que aún la serie no es del todo NO auto-correlacionada; esto podría indicarnos que, en efecto existen algunos períodos del tiempo dentro de la serie que afectaron de manera exógena a la serie y que por lo mismo no estamos considerando dentro del modelo en sí. Para confirmar esta afirmación, podemos realizar el análisis de correlogramas, tanto el simple como el parcial, para verificar si existen o no aun factores que no estamos considerando dentro de la serie. Para esto, podemos ocupar el siguiente código:

```
layout(1:2)
acf(SarIPre_Mod$residuals , 50, main = "Correlograma Simple de la serie." )
pacf(SarIPre_Mod$residuals , 50, main = "Correlograma Parcial de la serie." )
```





Tal parece que no existen otros posibles factores que alteren el comportamiento de la serie, a excepción de dos valores atípicos en uno de los rezagos de los correlogramas. La presencia de valores atípicos que se encuentran en un único de los correlogramas es un indicio importante sobre la existencia de un efecto **exones** sobre el comportamiento de la serie, ya que visto de una manera estadística, indica que existe un punto " no normal". en el comportamiento de la serie. Para comprobar esta hipótesis, procederemos a realizar las pruebas de normalidad sobre los residuos del modelo y verificaremos que estos se distribuyen o no de manera normal. Utilizaremos las dos pruebas de normalidad que hemos estado usando para confirmar este supuesto. Para esto utilizaremos los siguientes códigos:

```
JarqueBera.test(SarIPre_Mod$residuals)

##
```

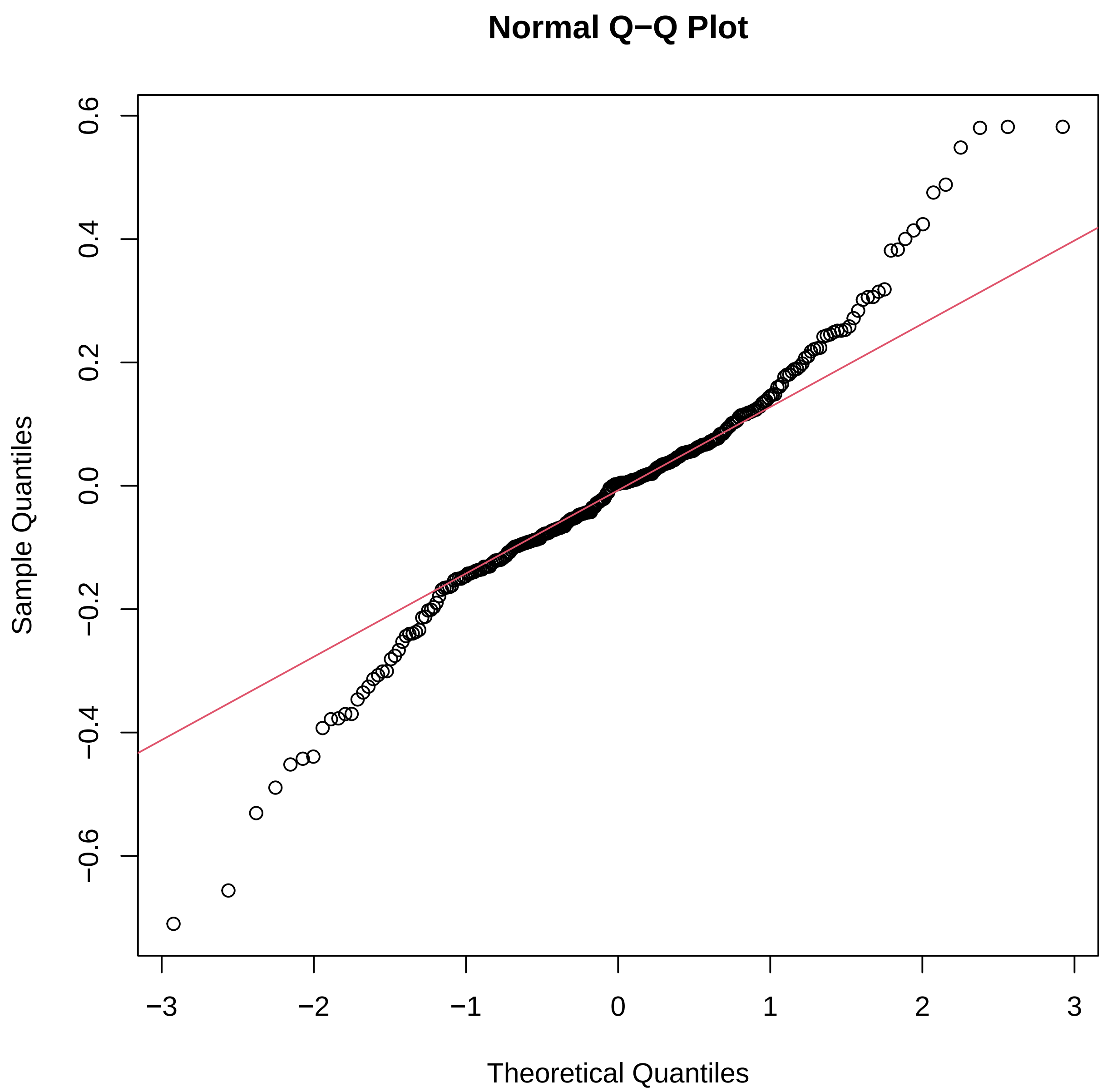
```
##  Jarque Bera Test
##
## data:  SarIPre_Mod$residuals
## X-squared = 38.612, df = 2, p-value = 4.125e-09
##
##
##  Skewness
##
## data:  SarIPre_Mod$residuals
## statistic = 0.0022534, p-value = 0.9875
##
##
##  Kurtosis
##
## data:  SarIPre_Mod$residuals
## statistic = 4.7938, p-value = 5.17e-10

shapiro.test(SarIPre_Mod$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  SarIPre_Mod$residuals
## W = 0.96817, p-value = 5.414e-06
```

Realizando las pruebas de *Jarque-Bera y Shapiro*, con valores p-valore menores al 0.05 nivel de significancia, rechazamos la hipótesis nula de Normalidad sobre ambas pruebas, confirmando a través de ambas que la serie de los residuos no se distribuye de manera normal. Lo siguiente seria realizar la prueba de auto-correlación utilizando la prueba *Ljung-Box* para verificar el supuesto de NO auto-correlación de los residuos. Para esto utilizaremos el siguiente código:

```
Box.test(SarIPre_Mod$residuals , lag=20, type = "Ljung-Box" )

##
##  Box-Ljung test
##
## data:  SarIPre_Mod$residuals
## X-squared = 36.583, df = 20, p-value = 0.01312
```

Los resultados de la prueba de Ljung-Box, con un p-valor menor al 0.05 nivel de significancia, tenemos suficiente evidencia para rechazar la hipótesis nula de NO auto-correlación, confirmamos que la serie de los residuos, en efecto, si presenta problemas de auto-correlación serial entre los residuos del modelo.
Por último, procederemos a utilizar la Q-Q plot como último estadpistico que nos ayude a confirmar la existencia o no de normalidad sobre los residuos de la serie. Para realizar la Q-Q plot, podemos utilizar el siguiente código:

```
qqnorm(SarIPre_Mod$residuals )
qqline(SarIPre_Mod$residuals , col ="2" )
```



Confirmamos la no existencia de normalidad y con problemas de auto-correlación sobre la serie de los residuos del modelo propuesto. En este punto resulta conveniente observar de nuevo la gráfica de los residuos de la serie y proponer alguna transformación o, en su caso, proponer utilizar una variable exógena que permita disminuir el efecto exógeno que la serie pudiera haber presentado en algún punto del tiempo. Esta variable exógena ayudara a encerrar el efecto directo del periodo que consideramos que ocurrió un choque exógeno (como la crisis del 2018). Para esto, utilizaremos un modelo **SARIMAX** o (**ARIMAX**, en caso de que no exista un componente estacional), el cual estableceremos de la siguiente manera:

$$SARIMAX(p,d,q)(P,D,Q)_m$$

$$\begin{aligned}(1-L)^d(1-L)^D Y_t = {} & \alpha + \rho X_t + \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \cdots + \Phi p Y_{t-p} + \beta_1 Y_{t-1} p + \beta_2 Y_{t-2} + \\ & \cdots + \beta_P Y_{t-P} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \eta_1 \varepsilon_{t-1} + \eta_2 \varepsilon_{t-2} + \\ & \cdots \eta_Q \varepsilon_{t-Q}\end{aligned}$$

Donde:

- Y_t es la variable de interés.

- αes la ordenada al origen y constante del modelo.
- ρX_t es el componente exógeno del modelo; donde ρ es el parámetro para estimar y conocer el efecto exógeno, y X_t es la variable que crearemos para especificar el período de tiempo.
- $\Phi_1, \Phi_2, \ldots, \Phi_p$ son los parámetros para estimar del modelo Auto-regresivo normal de orden p.
- $\theta_1, \theta_2, \ldots, \theta_q$ son los parámetros del modelo MA normal de orden q.
- $\beta_1, \beta_2, \ldots, \beta_P$ son los parámetros para estimar del modelo Auto-regresivo estacional del orden P.
- $\eta_1, \eta_2, \ldots, \eta_Q$ son los parámetros para estimar del modelo MA estacional de orden Q.
- $(1-L)^d$ es el factor que determina el número de diferencias normales que se deba aplicar a la variable de interés de acuerdo con el orden de integración d.
- $(1-L)^D$ es el factor que determina el número de diferencias estacionales que se debe de aplicar a la variable de interés de acuerdo con el orden de integración D.
- ε_t es la variable aleatoria o error del modelo.

Una vez establecido el modelo, podemos reducir la formula a una simple expresión que sea más sencilla de leer:

$$\text{SARIMAX con efectos exógenos: } (1-L)^d(1-L)^D Y_t = \rho X_t + U_t$$

Donde:

$$\begin{aligned} U_t =& \alpha + \rho X_t + \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \cdots + \Phi p Y_{t-p} + \beta_1 Y_{t-1} p + \beta_2 Y_{t-2} + \cdots + \beta_P Y_{t-P} + \\ & \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \eta_1 \varepsilon_{t-1} + \eta_2 \varepsilon_{t-2} + \cdots \eta_Q \varepsilon_{t-Q} \end{aligned}$$

Para poder identificar posibles Outliers (o choques exógenos) en la serie podemos ocupar la función *tso()* que nos brinda la paquetería *tsoutliers*, el cual nos ayudará a dos cosas:

- Identificar los períodos de tiempo dentro de la serie que presentan desviaciones importantes que no se presentan recurrentemenete en la serie (por lo que podemos suponer que se dio por un choque externo).
- Nos permite eliminar tales efectos, dándonos una seria limpia de los mismos

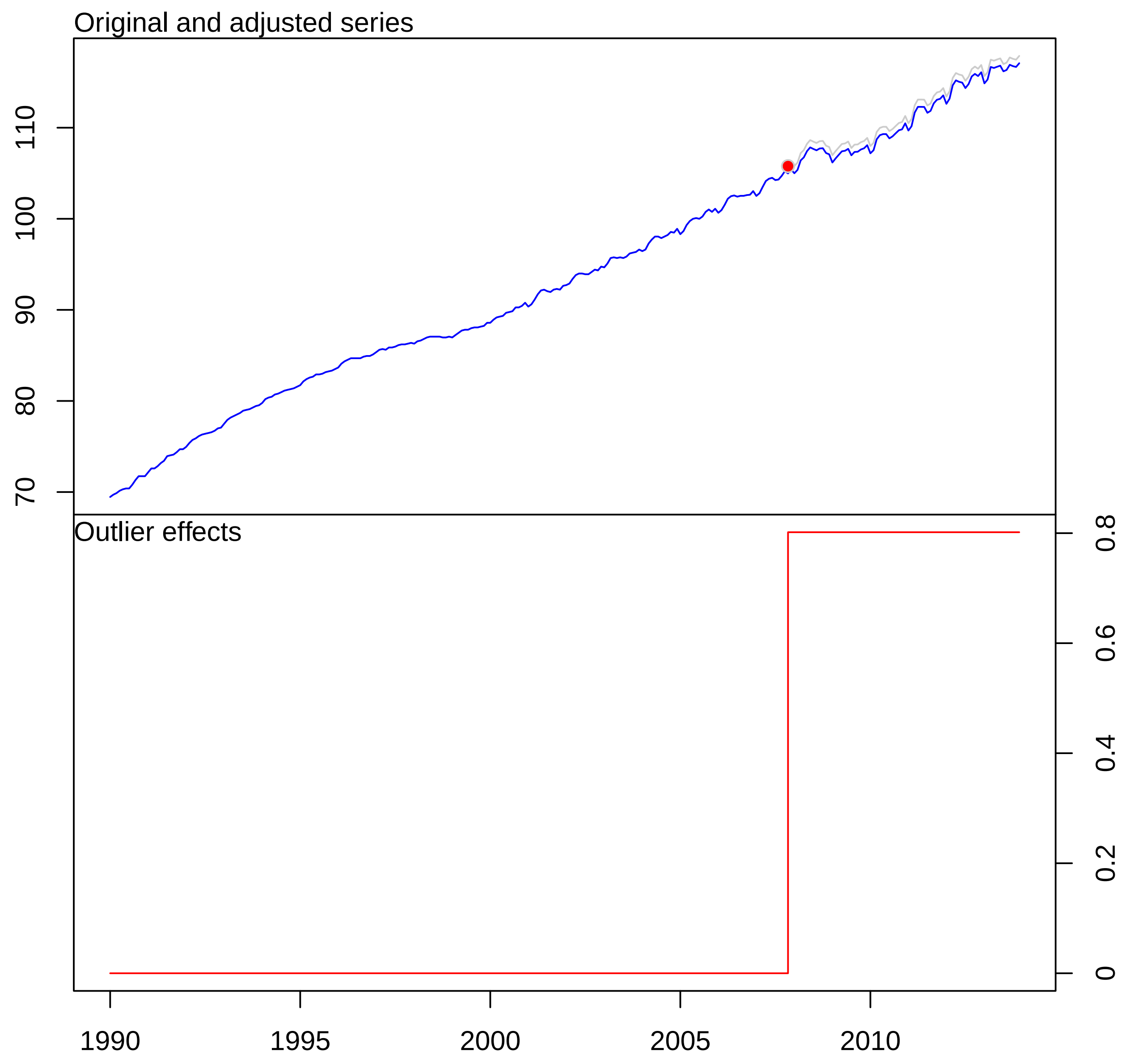
Primero necesitamos identificar estos choques, por lo cual ocuparemos el siguiente código:

```
Outliers <- tso(y=hicp[[1]] )

## Warning in locate.outliers.iloop(resid = resid, pars = pars, cval = cval, : stopped
when 'maxit.iloop'was reached
## Warning in locate.outliers.iloop(resid = resid, pars = pars, cval = cval, : stopped
when 'maxit.iloop'was reached
## Warning in locate.outliers.oloop(y = y, fit = fit, types = types, cval = cval, : stopped
when 'maxit.oloop = 4'was reached

Outliers

## Series: hicp[[1]]
## Regression with ARIMA(0,1,0)(1,0,0)[12] errors
##
## Coefficients:
##          sar1   LS215
##        0.8461  0.8017
## s.e.   0.0314  0.1575
##
```

```
## sigma^2 = 0.04288:  log likelihood = 38.21
## AIC=-70.42   AICc=-70.33   BIC=-59.44
##
## Outliers:
##   type ind    time coefhat tstat
## 1   LS 215 2007:11  0.8017  5.09
```

Para observar de mejor maqnera loa Outliers que la función *tso()* identifico, podemos graficarlos utilizando el siguiente código:

```
plot(Outliers)
```



Pudimos identificar que existe un periodo en el tiempo dentro de la serie que afecto de manera exógena al comportamiento de esta; podemos ver que el efecto del mismo fue importante. Existen dos formas de preceder con este problema; podemos intentar eliminar el Outlier de dicha serie, o bien considerarla dentro de un modelo Exógeno diferente. Pra nuestro caso, vamos a guardar el Outlier en el objeto llamado "Outliers", y a su vez, utilizaremos dichos Outliers para estimar nuevamente algunas propuestas de modelos SARIMA exógenos. Lo que procederemos a hacer es estimar estas propuestas

de modelos SARIMA Exógenos guardando los Outliers, y utilizándolos en los modelos que vamos a estimar. Los modelos propuestos para dicha tarea serán exactamente los mismos que propusimos con anterioridad, con la diferencia de que agregaremos en la opción xreg el data frame de los Outiliers que descubrimos en la serie.
Para eso podemos ocupar el siguiente código:

```
Outliers <- Outliers[["fit"]][["call"]][["xreg"]]
SarAPreOut_Mod <- arima(Precios , order = c(5,1,0), seasonal = list(order= c(1,1,1)),xreg = Outl
SarBPreOut_Mod <- arima(Precios , order = c(5,1,0), seasonal = list(order= c(1,1,0)),xreg = Outl
SarCPreOut_Mod <- arima(Precios , order = c(5,1,0), seasonal = list(order= c(0,1,1)),xreg = Outl
SarDPreOut_Mod <- arima(Precios , order = c(3,1,0), seasonal = list(order= c(1,1,1)),xreg = Outl
SarEPreOut_Mod <- arima(Precios , order = c(3,1,0), seasonal = list(order= c(1,1,0)),xreg = Outl
SarFPreOut_Mod <- arima(Precios , order = c(3,1,0), seasonal = list(order= c(0,1,1)),xreg = Outl
SarGPreOut_Mod <- arima(Precios , order = c(1,1,0), seasonal = list(order= c(1,1,1)),xreg = Outl
SarHPreOut_Mod <- arima(Precios , order = c(1,1,0), seasonal = list(order= c(1,1,0)),xreg = Outl
SarIPreOut_Mod <- arima(Precios , order = c(1,1,0), seasonal = list(order= c(0,1,1)),xreg = Outl
```

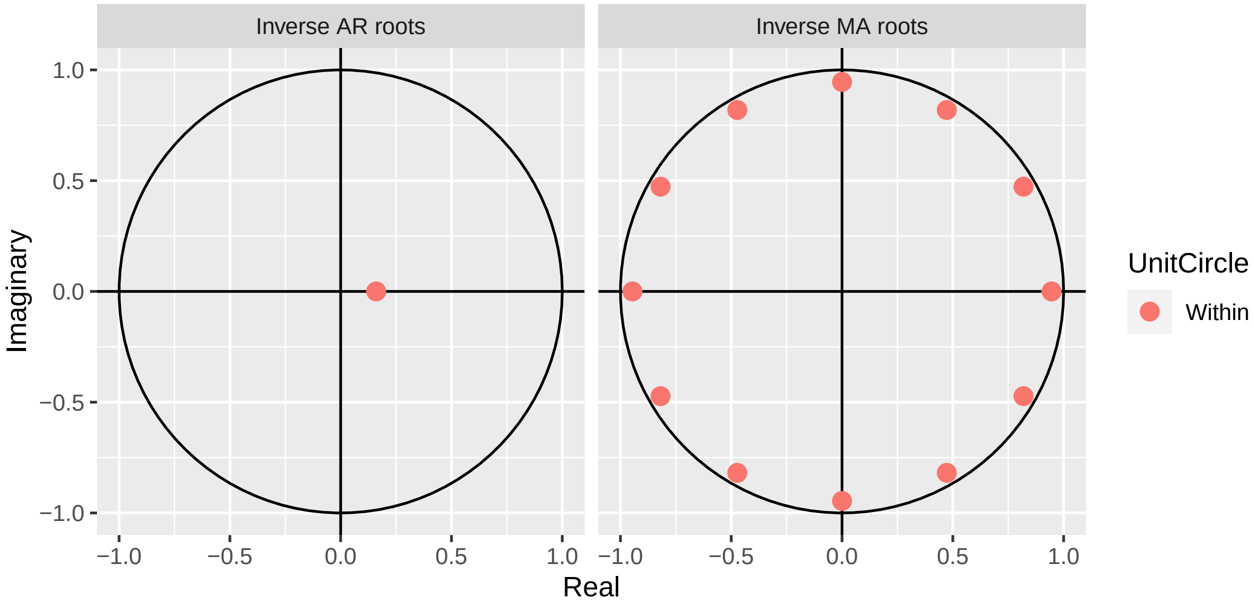
Los criterios de información son los siguientes:

```
SarAPreOut_Mod$aic

## [1] -122.0578

SarBPreOut_Mod$aic

## [1] -109.2959

SarCPreOut_Mod$aic

## [1] -122.6353

SarDPreOut_Mod$aic

## [1] -125.4117

SarEPreOut_Mod$aic

## [1] -111.3796

SarFPreOut_Mod$aic

## [1] -126.0907

SarGPreOut_Mod$aic

## [1] -128.9146

SarHPreOut_Mod$aic

## [1] -115.2257

SarIPreOut_Mod$aic

## [1] -129.7032
```

Los resultados indican que el mejor modelo para estimar el comportamiento de la serie es el modelo SARIMA(1,1,0)(0,1,1,1) con Outliers como variables exógenas. Una vez que tenemos estimado el

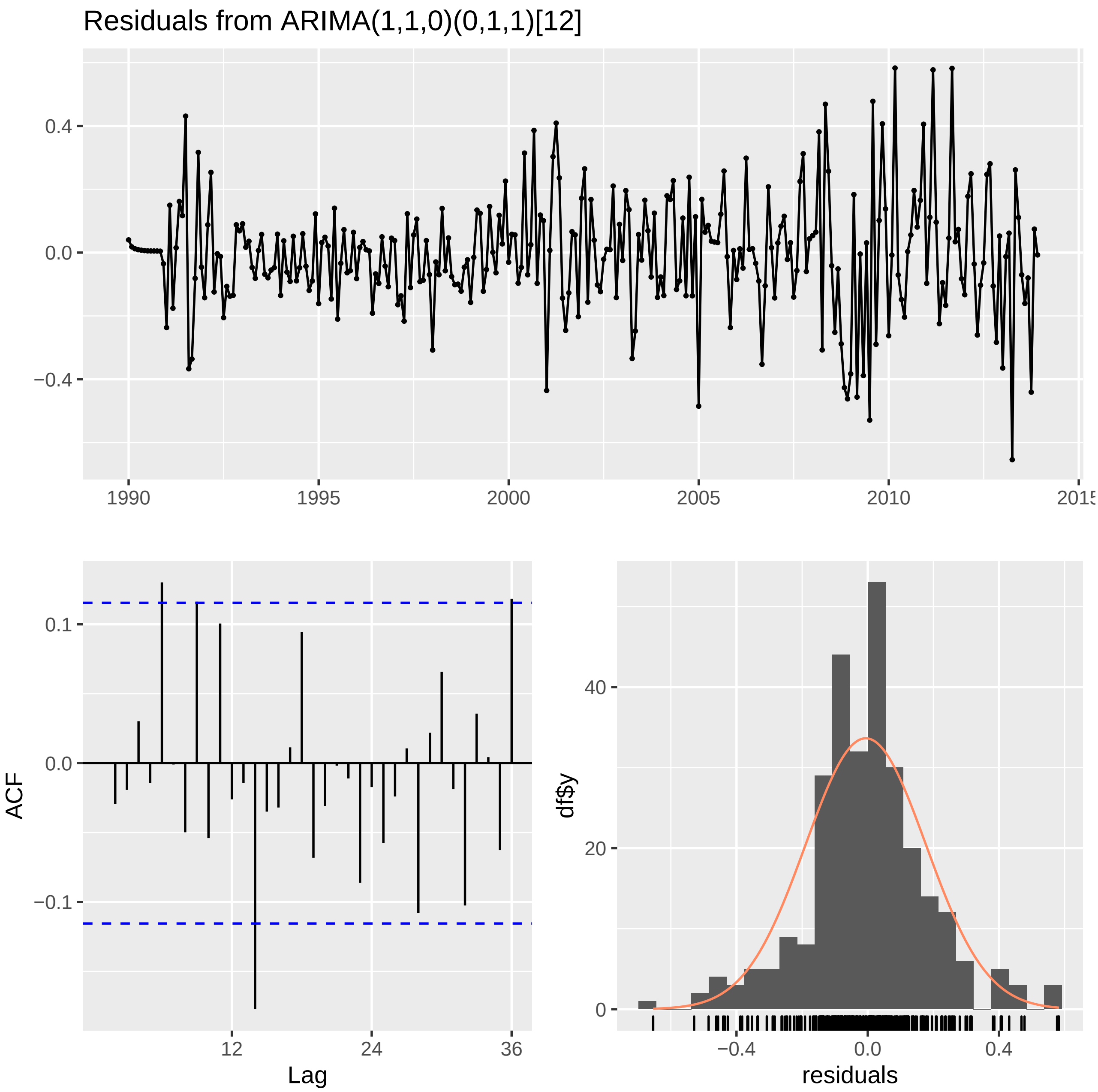
modelo que ocuparemos para realizar las proyecciones sobre la serie, necesitamos de igual manera realizar todas y cada una de las pruebas sobre la estabilidad del modelo y sobre el cumplimiento de los supuestos de los residuos de la serie.
La prueba con la que empezaríamos es la *Prueba de Raíces unitarias* para verificar la estabilidad del modelo, la cual podemos realizar con el siguiente código:

```
autoplot(SarIPreOut_Mod )+theme_gray()
```



Podemos comprobar que los parámetros estimados del modelo SARIMA que seleccionamos son estables dentro del mismo, de tal manera que las proyecciones de la serie no presentaran un comportamiento explosivo que aleje a los valores de su media. Una vez cumplido este supuesto, podemos proceder a realizar el análisis sobre los residuos de la serie, e ir comprando poco a coco que estos se cumplan para realizar proyecciones eficientes sobre la serie.
De igual manera que en la estimaciones anteriores, lo primero sería proceder a utilizar la función *checkresiduals()* sobre el modelo seleccionado para ver a grandes rasgos las principales características de los residuos del modelo exógeno. Para esto podemos ocupar el siguiente código:

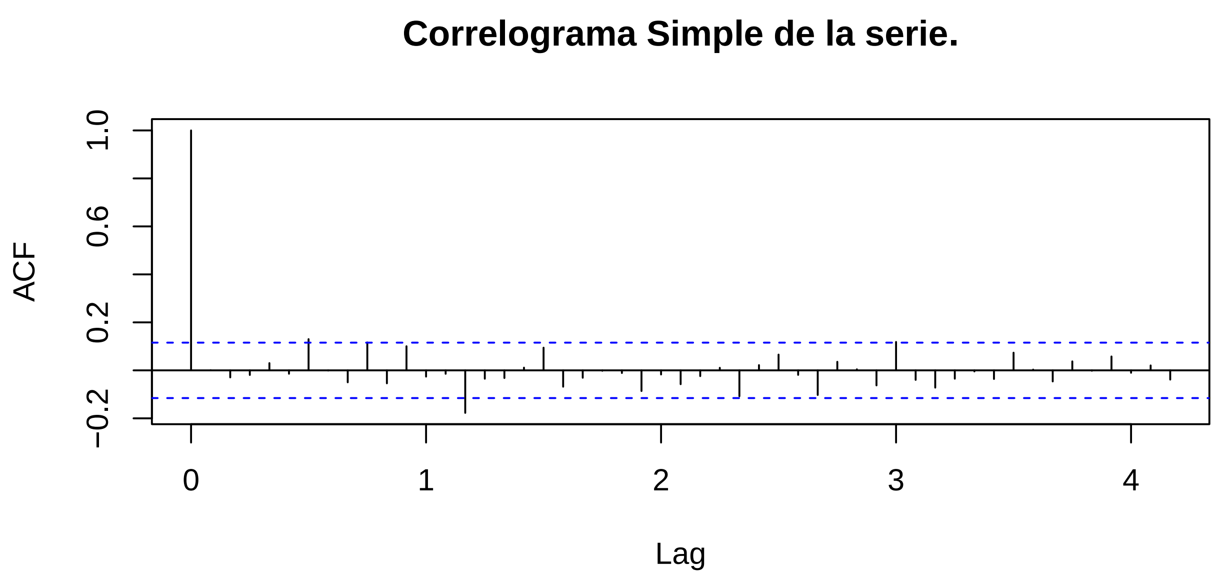
```
checkresiduals(SarIPreOut_Mod)
```

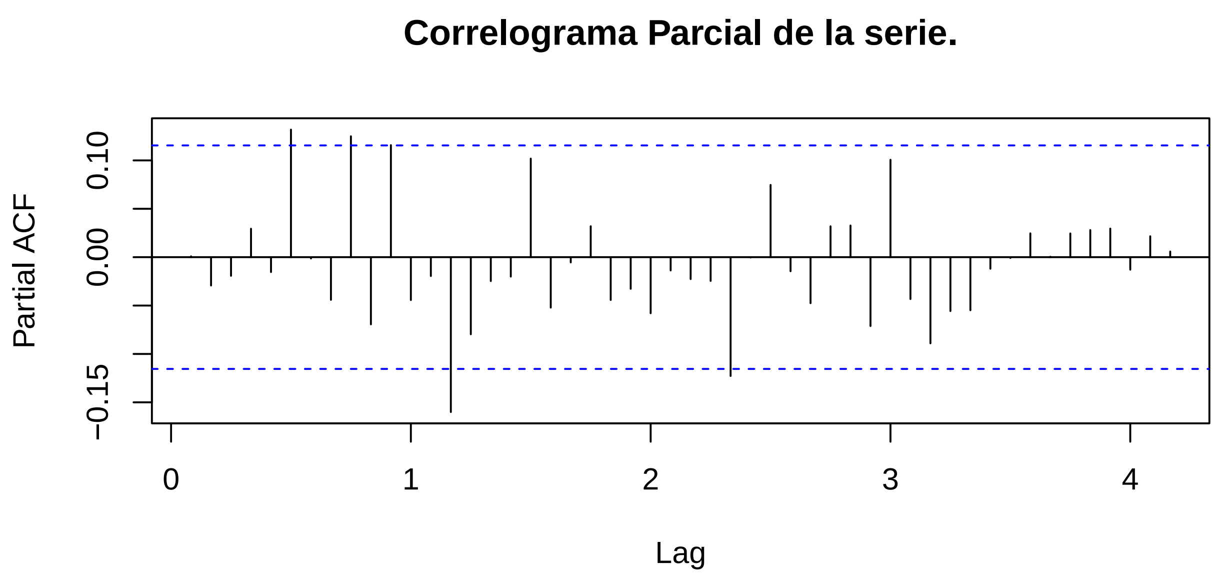



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)(0,1,1)[12]
## Q* = 31.963, df = 21, p-value = 0.05906
##
## Model df: 3.   Total lags used: 24
```

Podemos apreciar que el incluir un Outlier en la estimación del modelo como tl no mejoro mucho los resultados de los residuos del modelo. A grandes rasgos podemos observar tanto que aún existen valores fuera de los rangos que estable el correlograma simple, y que la distribución de los errores es NO normal debido a los picos que esta tiene en el histograma.
Intentaremos proceder con las siguientes pruebas para confirmar nuestras sospechas: Primer realizamos los correlogramas simple y parcial con el siguiente código:

```
layout(1:2)
acf(SarIPreOut_Mod$residuals , 50, main = "Correlograma Simple de la serie.")
pacf(SarIPreOut_Mod$residuals, 50, main = "Correlograma Parcial de la serie." )
```





Al observar los correlogramas simple y parcial, podemos observar que existen rezagos dentro de la serie de residuos que indican que aún existen otros factores dentro de la serie que aún pueden afectar a la proyección de la serie. Estos rezagos están tanto en el correlograma simple como en el parcial, de tal forma que ese efecto tanto al proceso AR como al proceso MA de la serie. Lo siguiente seria realizar las pruebas sobre el cumplimiento de normalidad en la serie.
Primero, realizamos la prueba de *Jarque-Bera* sobre los residuos del modelo con siguiente código:

```
jarque.bera.test(SarIPreOut_Mod$residuals )

##
##  Jarque Bera Test
##
## data:  SarIPreOut_Mod$residuals
## X-squared = 25.455, df = 2, p-value = 2.968e-06
```

Los resultados de la prueba *Jarque-Bera*, con un p-valor muchísimo menor al 0.05 nivel de significancia, se rechaza la hipótesis nula de normalidad sobre la serie, de tal manera que existe evidencia para afirmar que dicha serie no se distribuye de manera normal. Este resultado no sorprende tanto ya que pudimos observar que existen picos importantes dentro del histograma de los residuos. Procedemos a realizar la *prueba Shapiro* para contrastar normalidad utilizando el siguiente código:

```
shapiro.test(SarIPreOut_Mod$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  SarIPreOut_Mod$residuals
## W = 0.97312, p-value = 3.12e-05
```

De igual manera que con la prueba de *Jarque-Bera*, la *prueba de Shapiro* indica, con un p-valor mucho menos al 0.05 nivel de significancia. que hay suficiente evidencia para rechazar la hipótesis nula de normalidad sobre la serie de los residuos del modelo, confirmando que los residuos no se distribuyen de manera normal.

Muy comúnmente las series de tiempo de tipo financieras (como la inflación, el precio de las acciones o los índices bursátiles) suelen presentar problemas de heteroscedasticidad, lo cual significa que la varianza de la serie tiene un comportamiento no constante, de tal forma que puede crecer conforme pasa el tiempo o viceversa. Para estimar este tipo de modelos se requiere de utilizar una serie de modelos Heterocedasticos que permitan considerar el efecto de volatilidad de serie; sin embargo, este tipo de modelos no se planean abordar en este texto. Por lo cual, existe otro procedimiento que suele funcionar en la mayoría de los casos: aplicar transformaciones sobre la serie para suavizar el comportamiento de la varianza. Es bastante común que cuando trabajamos con datos estos tengan una varianza muy grande que, por consecuencia, ocasione que la serie no se distribuya de manera normal. Es conveniente en muchas ocasiones transformar estas series para que sea más fácil modelarlas. Existen distintas técnicas que pueden utilizarse cuando muchos de los datos se concentran en un extremo de la distribución y todos los valores son positivos. Una de estas técnicas comúnmente utilizadas es la transformación logarítmica. Para transformar la serie original en el logaritmo natural de la misma, podemos ocupar la función *log()* sobre la serie. Para ahorrarnos pasos, empezaremos con conseguir los Outliers de la serie transformada, y los guardaremos de igual manera en el objeto Outliers, tal y como hacemos en el siguiente código:

```
Outliers <- tso(y=log(hicp[[1]]))

## Warning in locate.outliers.oloop(y = y, fit = fit, types = types, cval = cval, : the
first 13 residuals were set to zero
## Warning in locate.outliers.oloop(y = y, fit = fit, types = types, cval = cval, : the
first 13 residuals were set to zero
## Warning in locate.outliers.oloop(y = y, fit = fit, types = types, cval = cval, : the
first 13 residuals were set to zero

Outliers

## Series: log(hicp[[1]])
## Regression with ARIMA(1,1,0)(1,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1    TC19    TC133   LS215    AO220
##       0.1773  0.8499  0.0062  -0.0047  0.0063  -0.0034
## s.e.  0.0606  0.0314  0.0013   0.0013  0.0015   0.0010
##
## sigma^2 = 3.887e-06:  log likelihood = 1378.18
## AIC=-2742.36   AICc=-2741.96   BIC=-2716.75
##
## Outliers:
##   type ind    time   coefhat  tstat
## 1   TC  19 1991:07  0.006203  4.747
## 2   TC 133 2001:01 -0.004726 -3.607
```

```
## 3   LS 215 2007:11  0.006323  4.204
## 4   AO 220 2008:04 -0.003441 -3.618
```