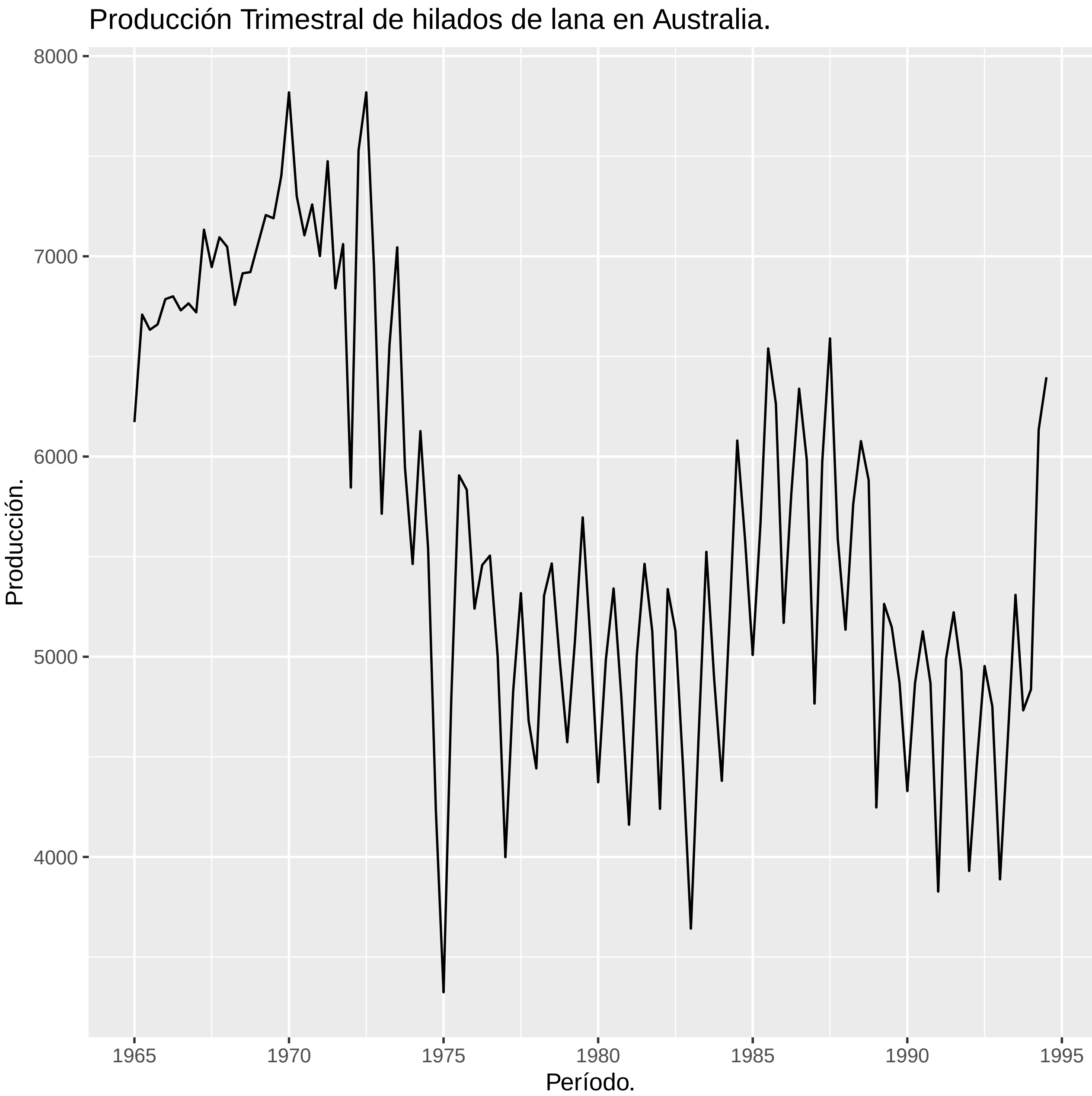
1. Componente estacional: Modelo multiplicativo SARIMA

Para empezar a abordar el tema del modelo SARIMA como tal utilizaremos un ejemplo intentando estimar un modelo ARIMA sobre una serie de tiempo.

```
library(forecast)

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

library(tseries)
library(ggplot2)
Produccion <- forecast::woolyrnq
autoplot(Produccion)+
  labs(title = "Producción Trimestral de hilados de lana en Australia.",
       x= "Período.", y="Producción." )+theme_gray()
```



La importancia de las gráficas temporales para el análisis de series de tiempo es resultado de que, a simple vista, es posible observar distintos fenómenos en la serie. Por ejemplo:

- Podemos observar que la serie tiene una tendencia, la cual puede que haya cambiado de un período a otro.
- Es posible que la serie sea algo volátil, debido a que tiene imoortantes caídas y bajadas en los valores. Por lo tanto; a simple vista podemos casi asegurar que la serie presenta un comportamiento NO estacionario en la media (Ya que no se mantiene en una media constante a causa de la presencia de una tendencia estocástica), y posiblemente, tampoco es constante en varianza. Sin embargo, para confirmar dichos fenómenos debemos de utilizar las pruebas de Estacionariedad que ya habíamos estudiado con anterioridad; *la prueba Dickey-Fuller Aumentada*, y la *Prueba Phillips-Perron*. Para realizar la prueba *Dickeey-Fuller Aumentada*, podemos utilizar el siguiente código sobre la serie.

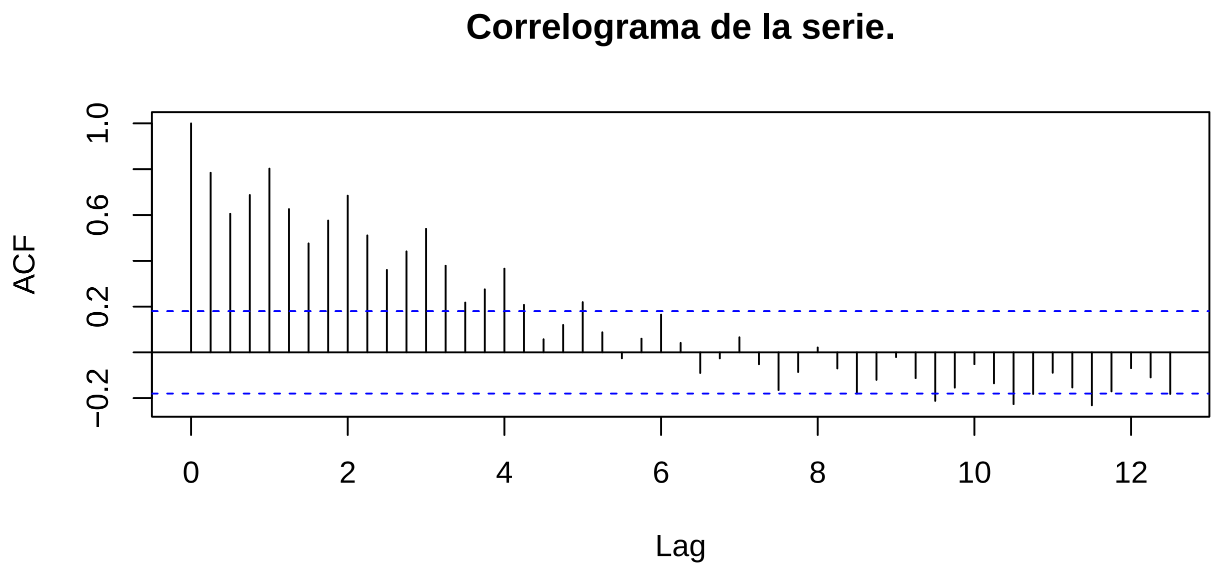
```
adf.test(Produccion,alternative = "stationary", k=0 )

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

##
##  Augmented Dickey-Fuller Test
##
## data:  Produccion
## Dickey-Fuller = -4.707, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

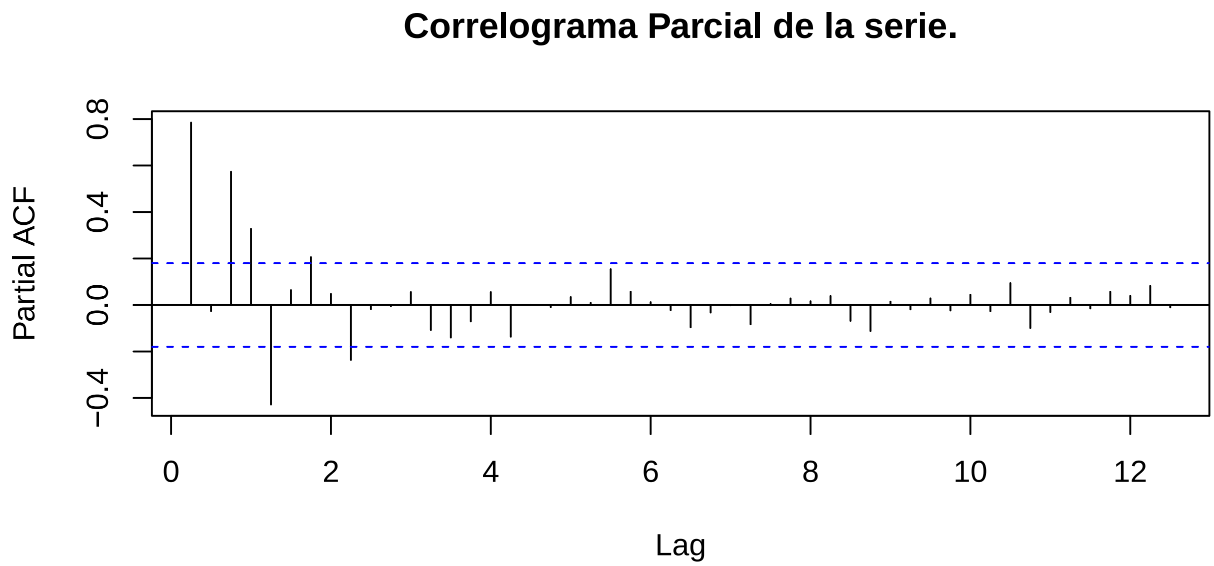
Con un p-valor de 0.01, menor al 0.05 nivel de significancia de la prueba, se rechaza la hipótesis nula, lo que confirma que la serie tiene un comportamiento estacionario. esto a simple vista podría sorprender; sin embargo, podemos utilizar a su vez la prueba *Phillips-Perron* para confirmar si la serie ya presenta un comportamiento estacionario. Para realizar la *prueba Phillips-Perron* sobre la serie de producción, podemos ocupar el siguiente código:

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

## Warning in pp.test(Produccion, alternative = "stationary"): p-value smaller than printed
p-value

##
##  Phillips-Perron Unit Root Test
##
## data:  Produccion
## Dickey-Fuller Z(alpha) = -38.211, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

Confirmamos con la *Prueba Phillips-Perron* que, en efecto, la serie original de producción es una serie estacionaria. Esto significa que no es necesario aplicar la primera diferencia de la serie, y que, además, no existe un factor de integración sobre la serie.Proseguimos a realizar el análisis de correlogramas de la serie utilizando el siguiente código:

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





No observamos que existe un comportamiento explosivo sobre el correlograma simple de la serie, de tal manera que la serie es en efecto estacionaria. Por otra parte, el hecho de que tanto el correlograma simple como en el correlograma parcial existan rezagos que sobrepasen los intervalos de correlación significa que existe la posibilidad de que la serie se componga tanto de un proceso estocástico AR como un proceso MA en conjunto, es decir, un proceso ARMA. Los modelos ARMA que propondremos para estimar el comportamiento de la serie de producción son:

- ARIMA (1,0,1)
- ARIMA (1,0,0)
- ARIMA (0,0,1)
- ARIMA (2,0,0)

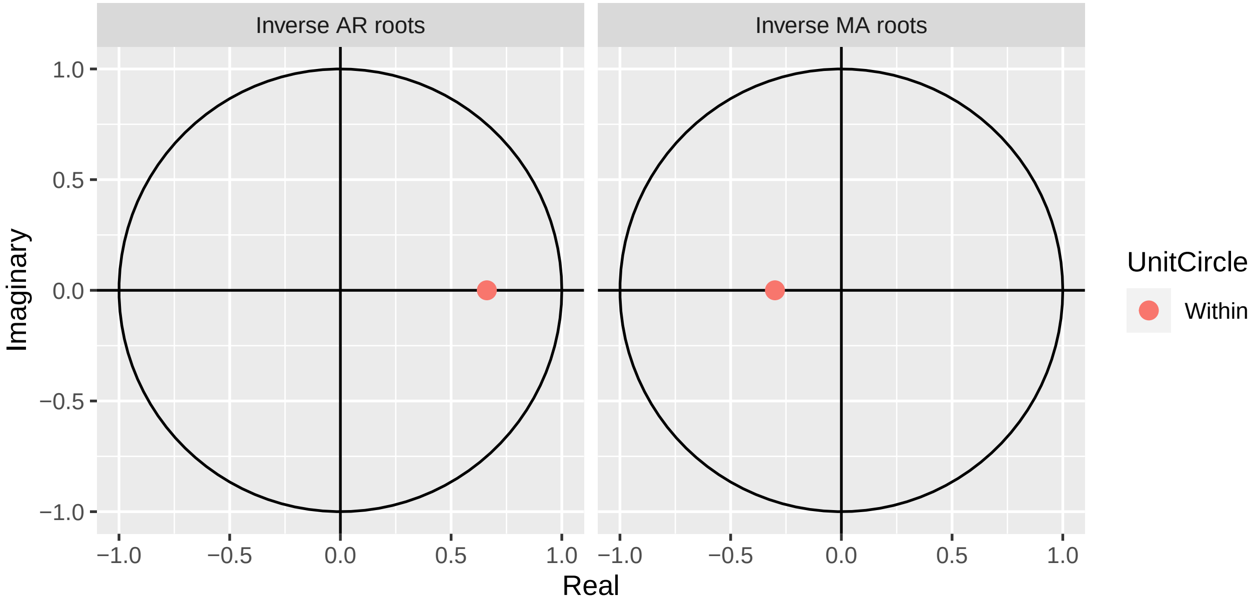
Ojo: Debido a que la serie resulto en un proceso estacionario, no fue necesario diferenciar la serie para buscar los componentes AR y MA de la serie, por lo que al momento de estimar el modelo ARIMA correspondiente debemos de dejar en 0 el factor de integración (el valor que se encuentra en medio del valor rezago AR y MA). Al igual que en las estimaciones anteriores, utilizaremos la función *arima()* para estimar los modelos seleccionados, y posterior a eso, calcularemos los ***criterios de información***

Akaike para verificar y seleccionar el mejor modelo para modelar el comportamiento de la serie de interés. Para realizar eso utilizaremos el siguiente código:

```
#Calculamos los modelos ARIMA propuestos con base en los correlogramas:
ModAProd_ARIMA <- arima(Produccion , order = c(1,0,1) )
ModBProd_ARIMA <- arima(Produccion , order = c(1,0,0) )
ModCProd_ARIMA <- arima(Produccion , order = c(0,0,1) )
ModDProd_ARIMA <- arima(Produccion , order = c(2,0,0) )
ModAProd_ARIMA$aic

## [1] 1882.007

ModBProd_ARIMA$aic

## [1] 1882.118

ModCProd_ARIMA$aic

## [1] 1918.45

ModDProd_ARIMA$aic

## [1] 1884.058
```

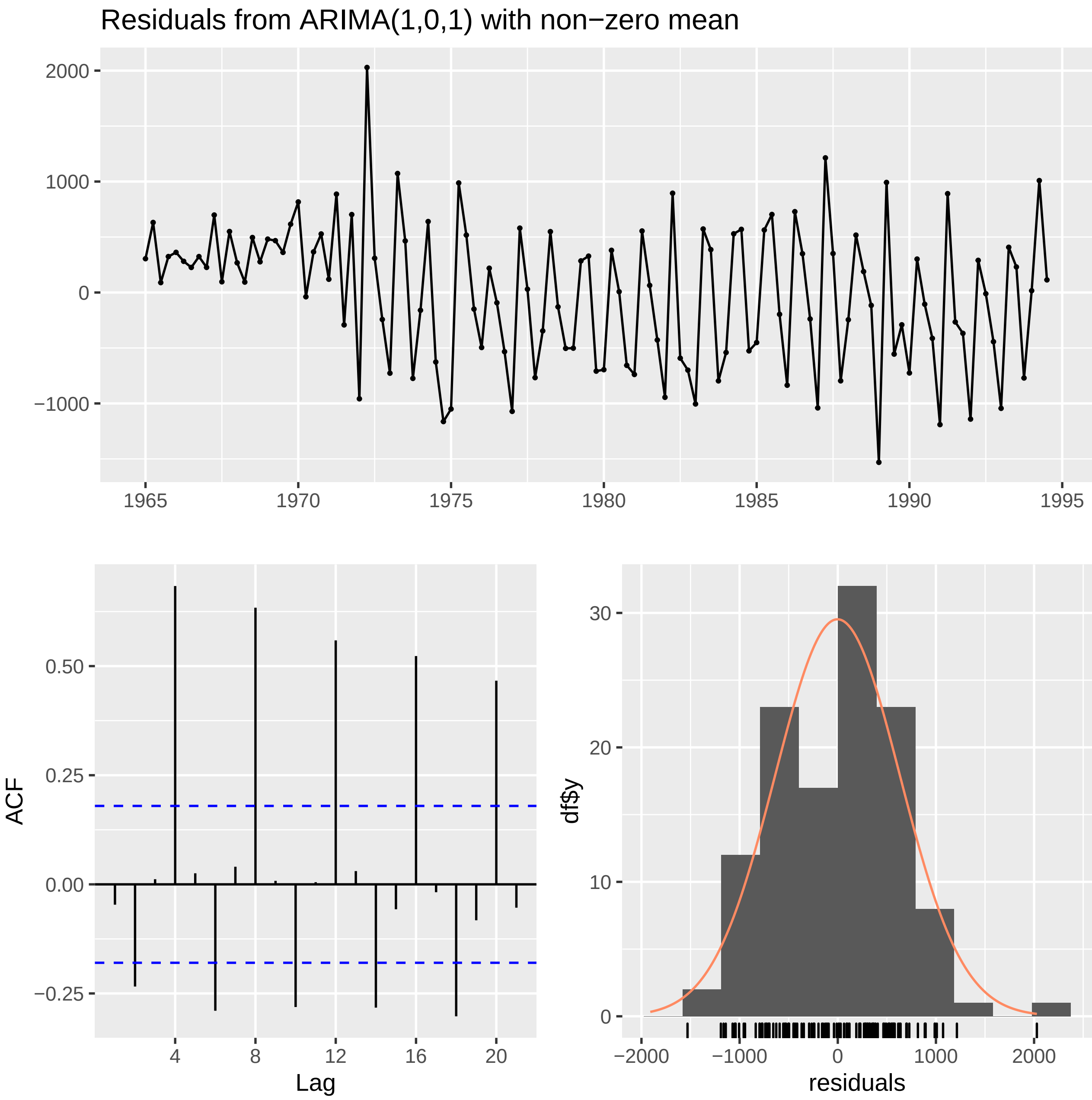
Utilizando *los criterios de información de Akaike*, es posible observar que el mejor modelo para estimar el comportamiento de la serie de Producción es el modelo ARIMA (1,0,1). Ahora, comprobaremos que los parámetros estimados sean estables utilizando la función *autoplot()* sobre el modelo seleccionado:

```
autoplot(ModAProd_ARIMA)+theme_gray()
```



Podemos observar que las raíces del modelo se encuentran dentro del círculo unitario, de tal forma que los resultados convergerán a una media y, por lo tanto, no tendrán un comportamiento explosivo que reste eficiencia la pronóstico del modelo.
Procederemos a realizar el análisis de los residuos sobre los errores de estimación del modelo. Primerio, un pequeño análisis visual utilizando la función *checkresiduals()*:

```
checkresiduals(ModAProd_ARIMA)
```



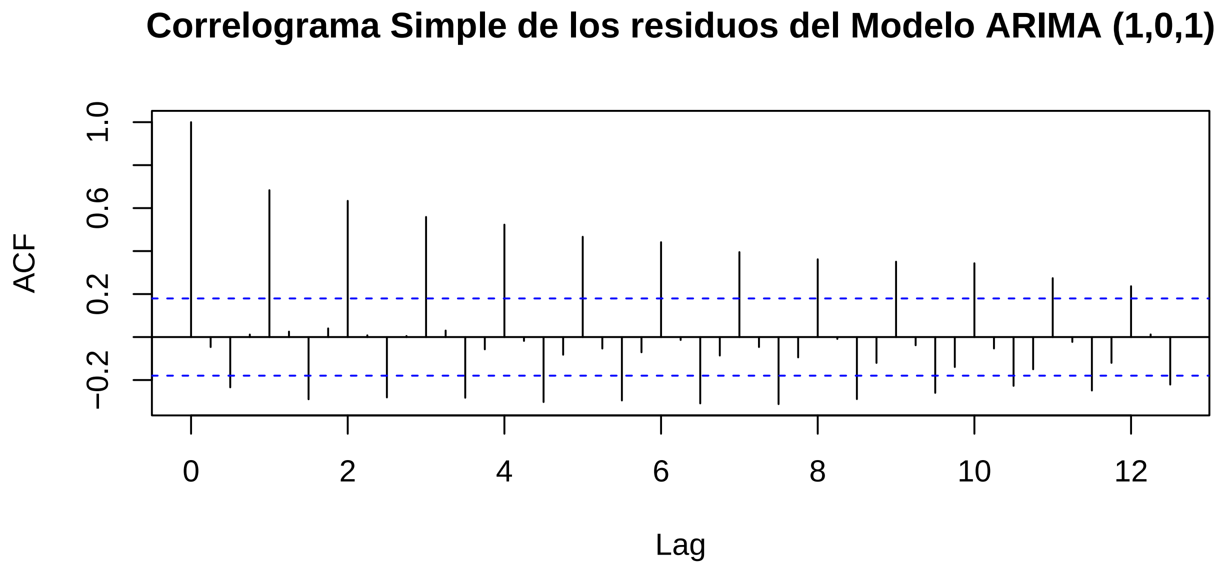

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1) with non-zero mean
## Q* = 128.59, df = 5, p-value < 2.2e-16
##
## Model df: 3.   Total lags used: 8
```

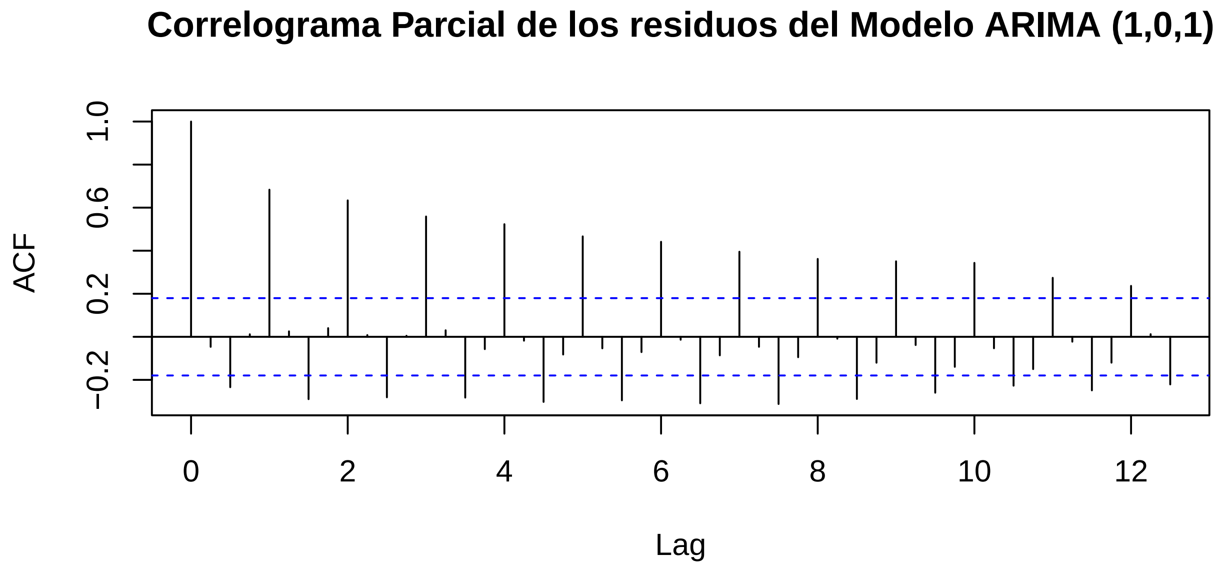
Con lo anterior podemos observar bastantes cosas interesantes:

- Tal parece que la serie se distribuye de manera norma, debido a que el histograma y la gráfica de densidad de la serie tienen un parecido cercano al de una distribución.
- Tal parece que existe un problema de auto-correlación serial importanye, ya que varios rezagos consecutivos (que parecen salirse cada determinado tiempo) se sale de su intervalo.

Para reforzar esta idea, podemos realizar el análisis de correlogramas utilizando el siguiente código:

```
layout(1:2)
acf(ModAProd_ARIMA$residuals , 50, main = "Correlograma Simple de los residuos del Modelo ARIMA
acf(ModAProd_ARIMA$residuals , 50, main = "Correlograma Parcial de los residuos del Modelo ARIMA
```





Al igual en el Correlograma Simple, también existen fuera del intervalo del Corrrelogram Parcial. Este tipo de comportamientos en los correlogramas indican que áun hay factores que no están considerando en los modelos y que afectan de manera importante al comportamiento de la serie temporal. Es importante observar que, de acuerdo con el correlograma simple, los valores fuera del intervalo presentan un comportamiento similar, ya que se repiten cada determinado tiempo de acuerdo al rezago de la serie. Este tipo de comportamiento es bastante común en las series temporales que presentan un comportamiento estacional; es decir, que tiene un efecto temporal que se repite cada determinado período de tiempo. Podemos observar este tipo de fenómenos, por ejemplo, en las ventas de los productos orgánicos, como las frutas o verduras, ya que su venta y producción dependen indispensablemente del período estacional del año. Para poder estimar modelos que consideren el comportamiento estacional de las series podemos ocupar los modelos **SARIMA**.
Para incorporar el comportamiento estacional de la serie, podemos anexar un factor de diferenciación adicional al de integración, con el fin de diferencial de acuerdo con el componete estacional del modelo. Para esto formulamos un modelo **SARIMA** con diferentes ordenes para el proceso AR, MA y I de

acuerdo con el orden estacional de la variable. Para esto podemos formular el siguiente modelo:

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

Donde:

- Y_t es la variable de interés-
- α es la ordenada al origen y constante del modelo.
- Φ_p son los parámetros para estimar del modelo Auto-regresivo normal de orden p.
- θ_q son los parámetros para estimar del modelo MA normal de orden q.
- β_P son los parámetros para estimar del modelo auto-regresivo estacional de orden P.
- η_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 debe aplicar a la variable de interés de acuerdo al orden de integración d.
- $(1-D)^D$ es el factor que determina el número de diferencias estacionales que se debe de aplicar a la variable de interés de acuerdo al orden de integración d.
- ε_t es la variable aleatoria o error del modelo.

Para empezar a estimar modelos SARIMA, procedemos a identificar cuantas diferencias estacionales deberán de aplicarse a la serie. Para poder encontrar este valor existe la función *nsdiffs()*. Para identificar el número de diferencias estacionales sobre la serie, procedemos a utilizar la función *nsdiffs()* sobre a serie de "Producción" que venimos trabajando, utilizando el siguiente código:

```
nsdiffs(Produccion )

## [1] 1
```

De acuerdo con el resultado de la función, pudimos identificar que se requiere únicamente una diferenciación estacional (lo que significa que debemos de diferenciar sobre 12 unidades ya que la serie es una serie mensual). Procederemos a realizar la estimación de varios modelos SARIMA utilizando la función *arima()* sobre la serie de "Producción". Para esto seleccionamos al menos 9 propuestas de modelos SARIMA en donde únicamente variaremos el número de valores auto-regresivos y de medias móviles, así como el orden de integración normal del modelo en la opción *order*, mientras que para variemos las órdenes del modelo SARIMA, así como su rango de integración estacional (el cual ya habíamos descubierto que era 1 diferenciación estacional) en el apartado de *seasonal*. Para realizar la estimación de estos modelos propuestos utilizaremos el siguiente código:

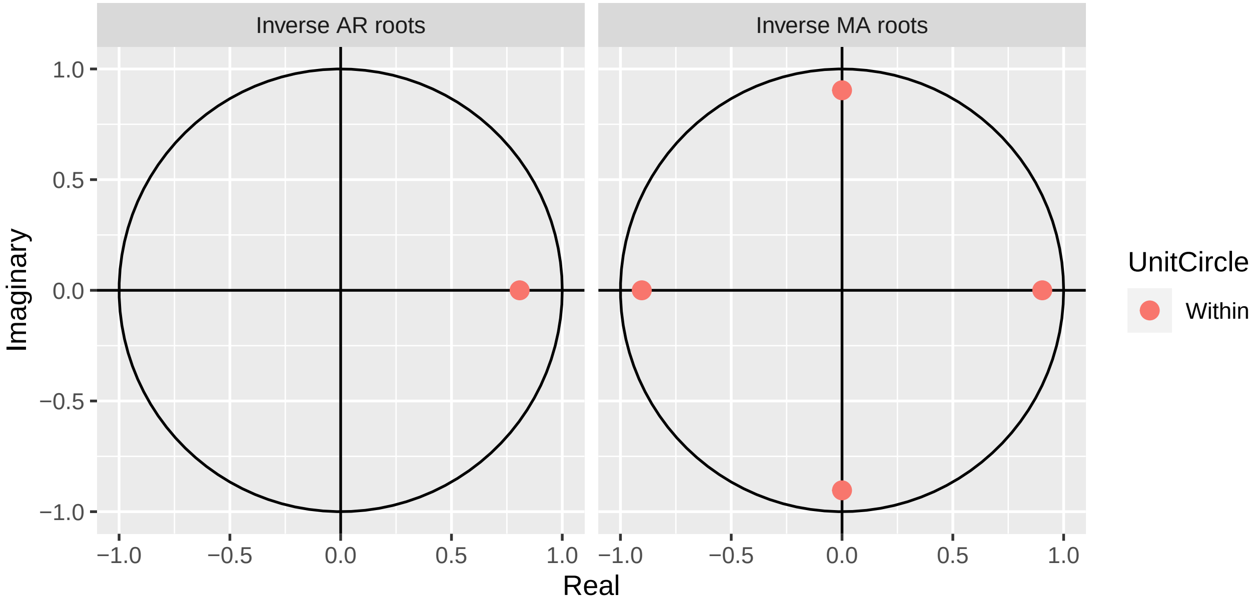
```
SarA_Mod <- arima(Produccion , order = c(1,0,1), seasonal = list(order = c(0,1,1)) )
SarB_Mod <- arima(Produccion , order = c(0,0,1), seasonal = list(order = c(0,1,1)) )
SarC_Mod <- arima(Produccion , order = c(1,0,0), seasonal = list(order = c(0,1,1)) )
SarD_Mod <- arima(Produccion , order = c(1,0,1), seasonal = list(order = c(1,1,1)) )
SarE_Mod <- arima(Produccion , order = c(0,0,1), seasonal = list(order = c(1,1,1)) )
SarF_Mod <- arima(Produccion , order = c(1,0,0), seasonal = list(order = c(1,1,1)) )
SarG_Mod <- arima(Produccion , order = c(1,0,1), seasonal = list(order = c(1,1,0)) )
SarH_Mod <- arima(Produccion , order = c(0,0,1), seasonal = list(order = c(1,1,0)) )
SarI_Mod <- arima(Produccion , order = c(1,0,0), seasonal = list(order = c(1,1,0)) )
```

Vamos a utilizar el mejor modelo de acuerdo con *los criterios de información de Akaike* utilizando el siguiente código:

```
SarA_Mod$aic

## [1] 1723.992

SarB_Mod$aic

## [1] 1753.157

SarC_Mod$aic

## [1] 1721.998

SarD_Mod$aic

## [1] 1725.809

SarE_Mod$aic

## [1] 1753.945

SarF_Mod$aic

## [1] 1723.809

SarG_Mod$aic

## [1] 1732.207

SarH_Mod$aic

## [1] 1753.853

SarI_Mod$aic

## [1] 1730.321
```

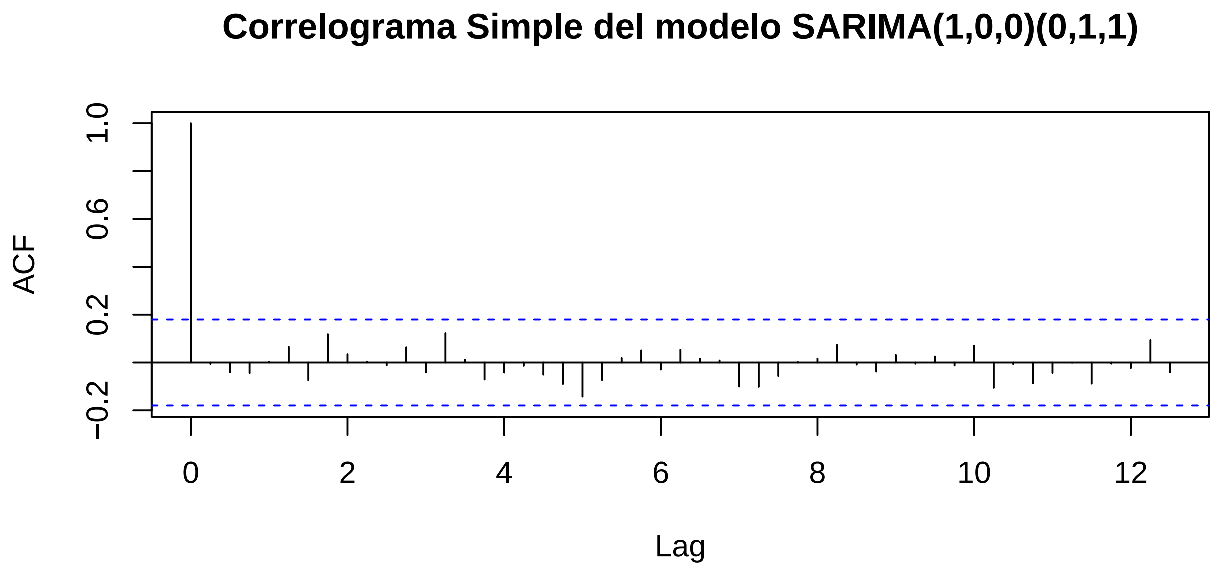
De acuerdo a los resultados de cada modelo, el mejor modelo obtenido es el modelo obtenido es el modelo SARIMA(1,0,0)(0,1,1), por lo que es este el modelo que ocuparemos para pronosticar nuestra serie. Primero realizaremos la prueba de estabilidad sobre los valores de los parámetros del modelo utilizando la función *autoplot()* sobre el modelo propuesto, tal y como realizamos en el siguiente código:

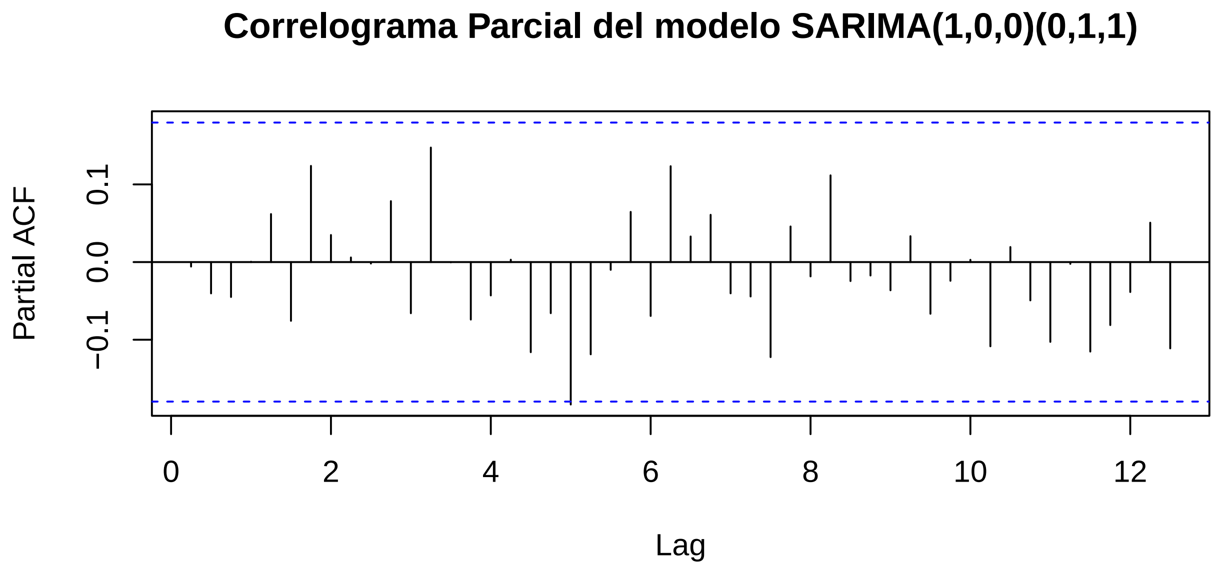
```
autoplot(SarC_Mod )+theme_gray()
```



Podemos observar a simple vista que mejoró considerablemente el comportamiento de los residuos, ya que ya no existen los valores rezagados en el correlograma simple; a su vez, parece ser que la serie de los residuos presentan un comportamiento normal, ya que únicamente un pico fuera de la distribución. A simple vista podemos confirmar que el modelo presentara resultados positivos.
Para obtener un panorama más completo sobre el comportamiento de dichos residuos, procederemos a realizar todas las pruebas ya mencionadas. Primero, realizamos los correlogramas simples y parciales utilizando el siguiente código:

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





Tanto el correlograma simple como el correlograma parcial indican que ya no existen otros posibles factores o fenómenos que afecten a la serie temporal, ya que no existen rezagos que se salgan de los intervalos de significancia de la prueba. Como siguiente prueba, necesitamos contrastar el cumplimiento del supuesto de NO auto-correlación, realizamos la prueba "Ljung-Box" sobre los errores, utilizando el siguiente código:

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

##
##  Box-Ljung test
##
## data:  SarC_Mod$residuals
## X-squared = 11.913, df = 20, p-value = 0.919
```

Al realizar la prueba Ljung-Box sobre los residuos del modelo, con un p-valor mucho mayor al 0.05 nivel de significancia, podemos aceptar la hipótesis nula de que la serie no tiene problemas de auto-correlación.
Para realizar la prueba de normalidad, utilizaremos la prueba *Jarque-Bera*, y la prueba *Shapiro*. El

uso de ambas pruebas, a igual que el uso de las pruebas de raíces unitarias, servirán para confirmar de mejor manera el cumplimiento o el incumplimiento del supuesto de normalidad de la serie. Para realizar la prueba de *Jarque-Bera* utilizaremos de siguiente código:

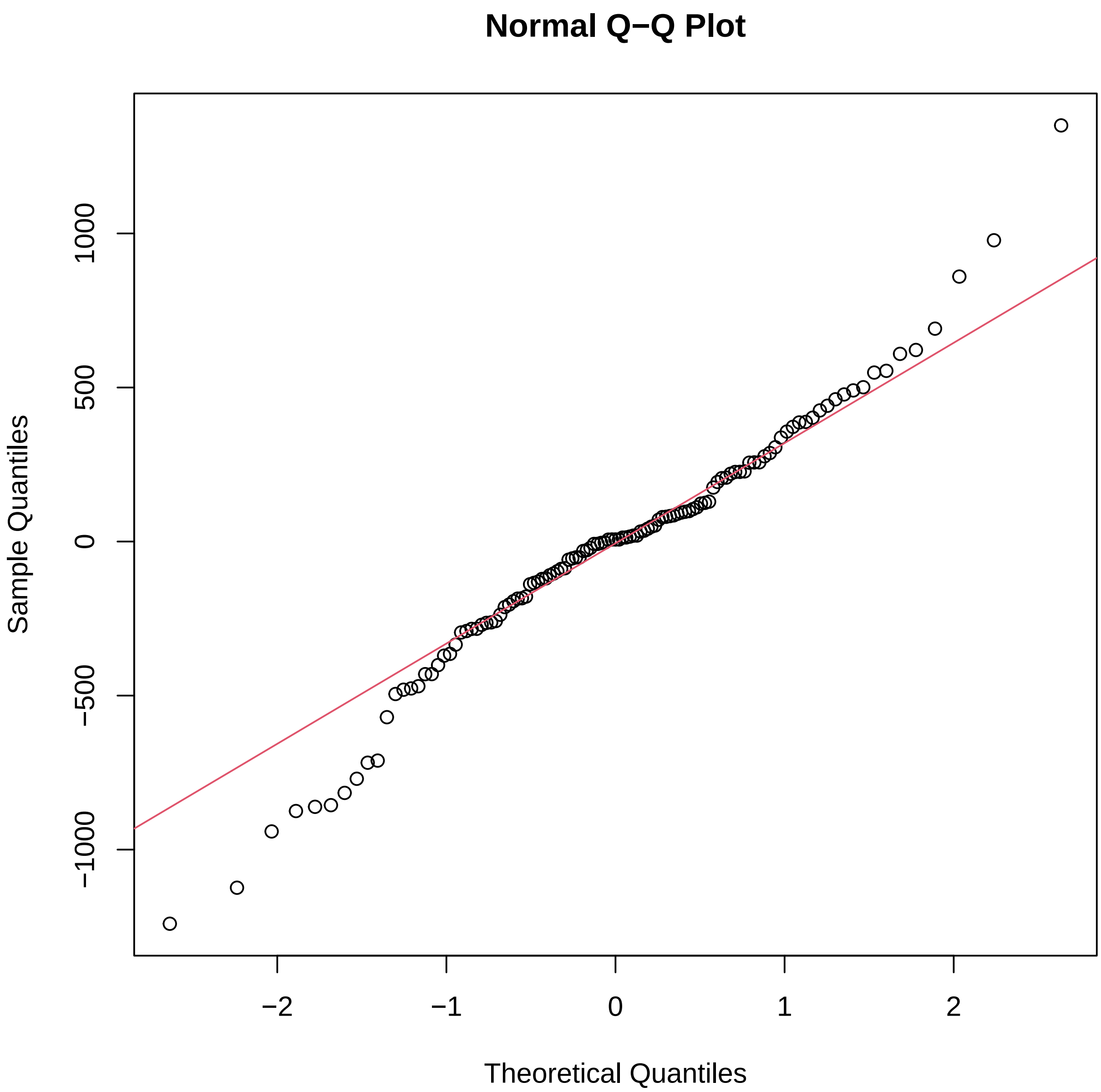
```
library(tseries)
jarque.bera.test(SarC_Mod$residuals)

##
##  Jarque Bera Test
##
## data:  SarC_Mod$residuals
## X-squared = 8.4141, df = 2, p-value = 0.01489
```

El resultado de la prueba, al igual que otros contrastes que hemos mencioando, se puede confirmar observando el p-valor obtenido. De acuerdo con los resultados de la prueba, con un p-valor ligeramente menor al 0.05, se rechaza la hipótesis nula de normalidad sobre los residuos del modelo seleccionado. Confirmamos que el modelo tiene residuos que se distribuyen de manera no normal de acuerdo con la *prueba de Jarque-Bera*. Intentaremos conformar dicho resultado utilizando de manera paralela la prueba de *Shapiro*. Para esto, utilizaremos la función *shapiro.test()* sobre los residuos del modelo utilizando el siguiente código:

```
shapiro.test(SarC_Mod$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  SarC_Mod$residuals
## W = 0.97261, p-value = 0.0156
```

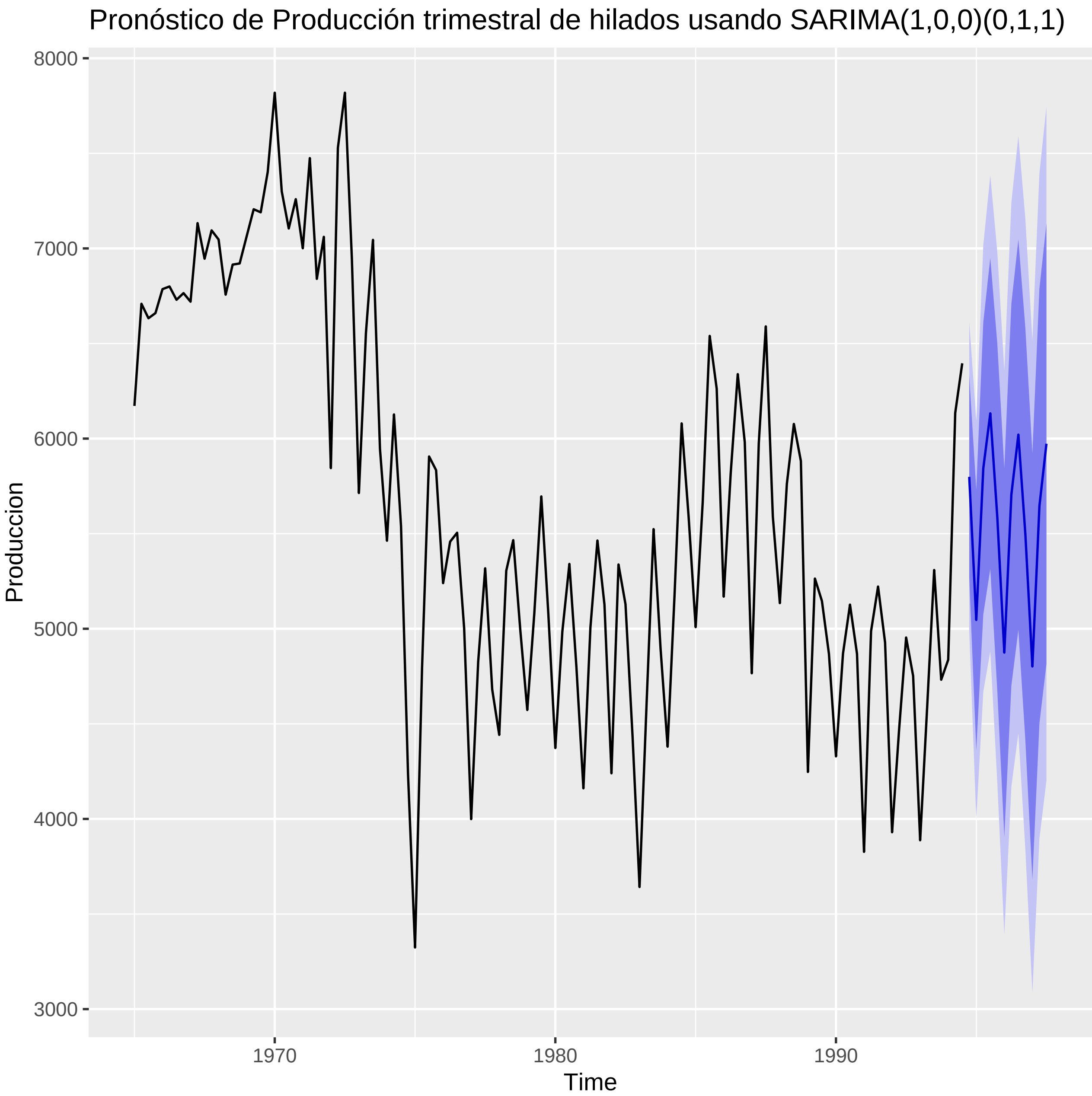
El p-valor de la prueba, con un valor de 0.0156, menor al nivel de significancia del 0.05, se rechza la hipótesis nula de normalidad sobre los residuos del Modelo. Este comportamiento no normal se confirma por al menos dos pruebas. Intentaremos realizar un tercer contraste de normailidad con la Q-Q plot sobre los residuos, la cual nos permite visualizar si los residuos siguen una distribución de una variable simulada normal. Para esto utilizaremos el siguiente código.

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



Podemos confirmar que la **serie de los residuos NO presenta un comportamiento normal**. Esto podría deberse a diversos tipos de efectos que aún no hemos considerado en la serie. Por el momento vamos a proceder con el pronóstico de la serie, aunque indicando siempre que el pronóstico podría no ser eficiente debido a que no cumple con el supuesto de normalidad. Para realizar el pronóstico utilizaremos el modelo seleccionad, y a su ves, lo graficaremos junto con sus intervalos de confianza utilizando el siguiente código:

```
SARIMA_ModC <- forecast(SarC_Mod, 12)
autoplot(SARIMA_ModC)+
  labs(title ="Pronóstico de Producción trimestral de hilados usando SARIMA(1,0,0)(0,1,1)"  )+th
```


Pronóstico de Producción trimestral de hilados usando SARIMA(1,0,0)(0,1,1)
8000
7000
6000
5000
4000
3000
Produccion
1970
1980
1990
Time