Modelos de Box y Jenkins

Trabajo 2. Análisis de Series de Tiempo

Bladimir Valerio Morales Torrez

Enero 2022

Contents

1 Introducción **2**

2 Indice de consumo de agua potable ICAP **2**

2.1 Datos 2

2.2 Periodo de estudio 2

2.3 Fuente de datos 2

3 Gráfico **3**

4 Datos de entrenamiento y test **3**

5 Ajuste del Modelo de Box y Jenkins **4**

5.1 Estacionariedad 5

5.2 Estacionalidad 10

5.3 Transformación de la serie temporal 12

5.4 Ajuste del Modelo 18

5.5 Validación de supuestos 23

5.6 Conclusión ajuste del modelo 30

6 Gráfico serie original y el ajustado **30**

7 Predicción **32**

8 MAPE **34**

9 Comparación de la predicción **35**

10 Conclusiones **35**

1 Introducción

Para este trabajo de análisis de series de tiempo se aplicarán técnicas de modelamiento en series temporales, específicamente los modelos de Box y Jenkins.

La serie de tiempo de estudio para este trabajo es:

- Indice de consumo de agua potable ICAP (enero 1990 a julio 2021).

Se puede encontrar el repositorio de datos y del informe en el siguiente enlace (https://github.com/bladimir-morales/modelos_box_jenkins).

Se puede visualizar el presente informe en formato pdf, en el siguiente enlace:

- https://bladimir-morales.github.io/modelos_box_jenkins/trabajo2.pdf

Se puede visualizar el presente informe en formato html, en el siguiente enlace:

- https://bladimir-morales.github.io/modelos_box_jenkins/trabajo2.html

2 Indice de consumo de agua potable ICAP

2.1 Datos

El indice mensual de consumo de agua potable de Bolivia ICAP, es un indicador que nos permite conocer *la evolución y comportamiento* del consumo de agua potable de los sectores privado y público a nivel general con año base de 1990, así para este año el indice será igual a 100 y para las siguientes gestiones presentará una variación (incremento o decremento) respecto al año base de acuerdo al consumo de agua potable del mes a tratarse.

2.2 Periodo de estudio

La serie de tiempo esta con periodicidad mensual, comprendidos desde enero de 1990 hasta julio de 2021, teniendo en total 379 observaciones.

2.3 Fuente de datos

La información del indice mensual de consumo de agua potable de Bolivia se puede encontrar en la página oficial del Instituto Nacional de Estadística (INE)[1], sección de "Estadísticas Económicas" y subsección "Servicios básicos". Específicamente se puede descargar los datos en formato establecido por la institución en excel del siguiente enlace: (https://nube.ine.gob.bo/index.php/s/M1H9axannIL7leg/download).

Los metadatos están disponibles en el Catálogo del Archivo Nacional de Datos (ANDA) del INE (http://anda4.ine.gob.bo/ANDA4_2/index.php/catalog/254).

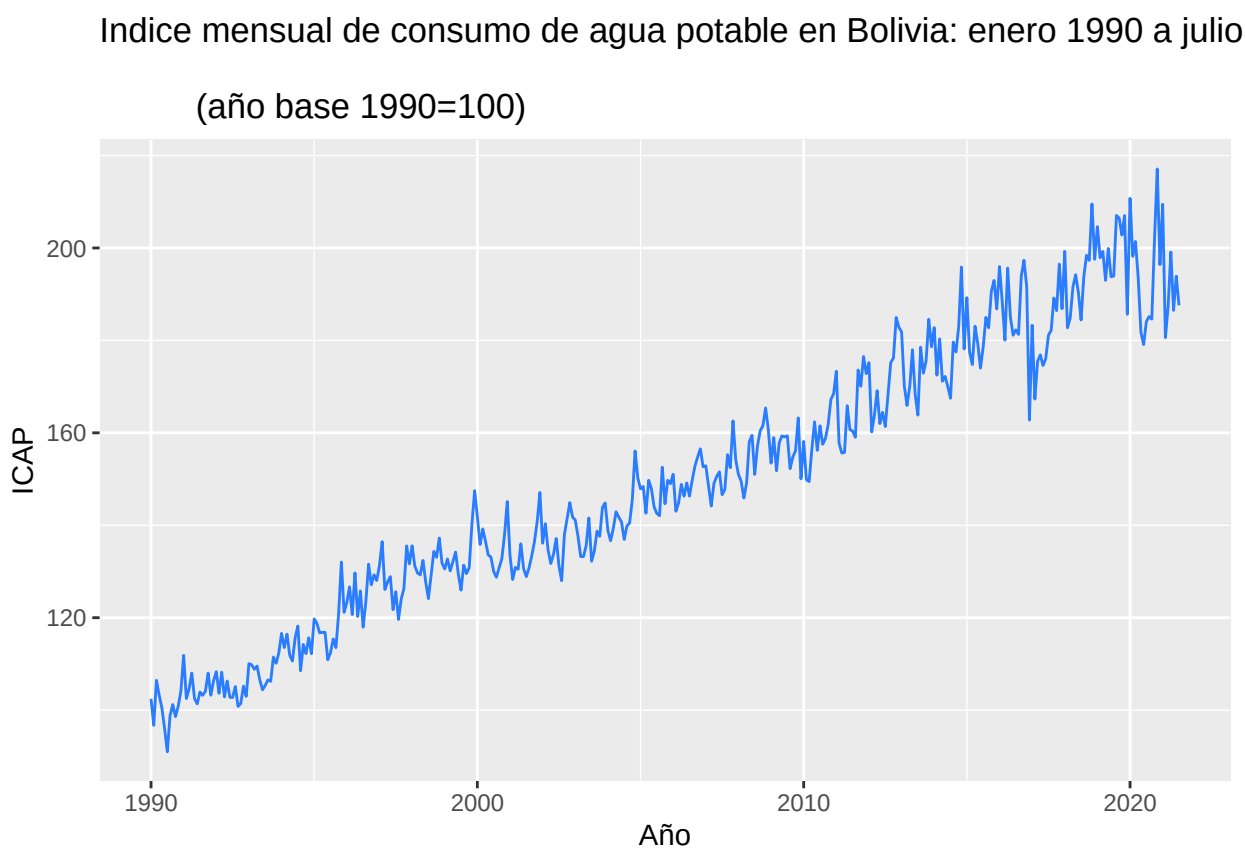
Para fines prácticos se puso la variable en estudio en formato *.txt, el cual puede ser descargada del siguiente enlace (https://raw.githubusercontent.com/bladimir-morales/series_de_tiempo/main/data/agua.txt).

[1] www.ine.gob.bo

3 Gráfico

```
url<-"https://raw.githubusercontent.com/bladimir-morales/modelos_box_jenkins/main/agua.txt?token=AOEHMZ
agua<-read.table(url,head=T)
serie<-ts(agua$agua,start = c(1990,1),frequency = 12)

autoplot(serie,series = "ICAP")+
  ggtitle("Indice mensual de consumo de agua potable en Bolivia: enero 1990 a julio 2021 \n
          (año base 1990=100)")+
  xlab("Año")+ylab("ICAP")+
  scale_color_manual(values="#2B7DFF")+
  theme(legend.position = "none")
```



En el gráfico visualmente se puede observar que la serie de tiempo en estudio tendría tendencia aditiva y un posible efecto estacional.

4 Datos de entrenamiento y test

Para efectos de obtener un modelo óptimo y lo más preciso posible, se dividirá la serie de tiempo en dos conjuntos:

- Conjunto de datos de entrenamiento:

Se tomará en cuenta los datos desde enero de 1990 hasta diciembre de 2019, contando con 360 observaciones.

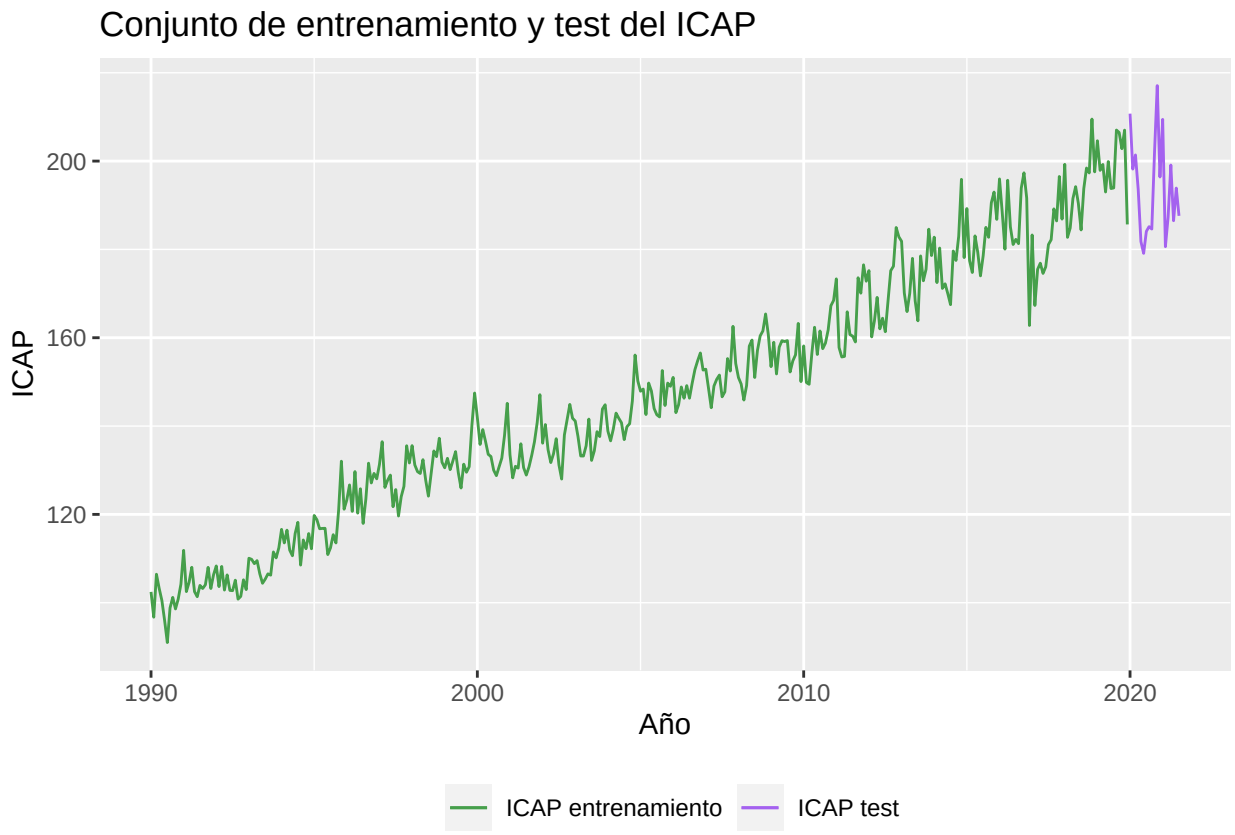
- Conjunto de datos de test.

Se tomará en cuenta los datos desde enero de 2020 hasta julio de 2021, contando con 19 observaciones.

En el siguiente gráfico se puede observar la serie de entrenamiento y de test.

```
serie_ent<-ts(agua$agua,start = c(1990,1),end = c(2019,12), frequency=12)
serie_test<-ts(agua$agua[361:379],start = c(2020,1), frequency=12)

autoplot(serie_ent,series = "ICAP entrenamiento")+
  autolayer(serie_test,series="ICAP test")+
  ggtitle("Conjunto de entrenamiento y test del ICAP")+
  xlab("Año")+ylab("ICAP")+
  scale_color_manual(values=c("#469F4B","#A462EF"))+
  theme(legend.position = "bottom",legend.title = element_blank() )
```



5 Ajuste del Modelo de Box y Jenkins

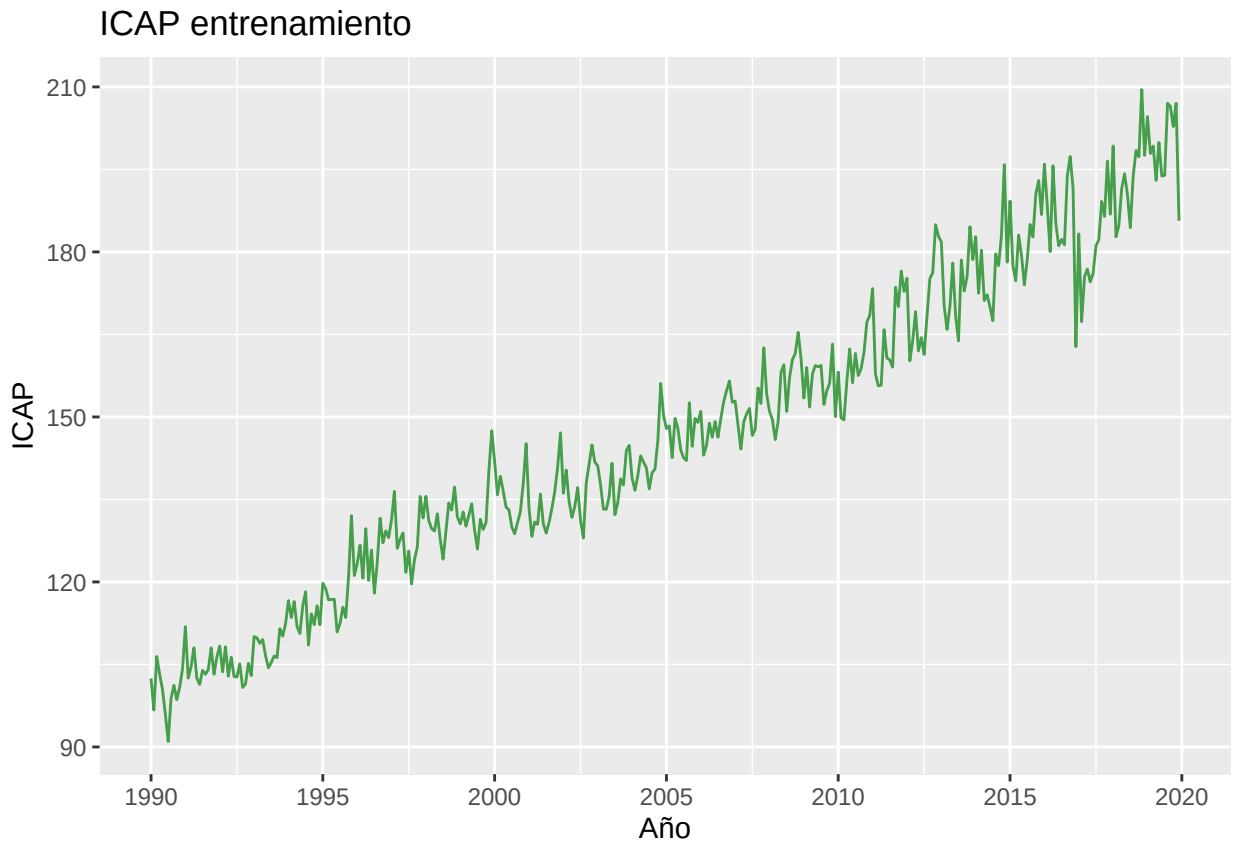
En esta sección se ajustará un modelo de Box y Jenkins a la serie del ICAP de entrenamiento, para el cual lo primero que se verificara es la estacionariedad (media y varianza) y la estacionalidad de la serie. Posteriormente se realizarán las transformaciones necesarias si es el caso para poder ajustar una serie estacionaria a un modelo de Box y Jenkins.

5.1 Estacionariedad

Para poder determinar si la serie del ICAP de entrenamiento es estacionaria en media y varianza se visualizará primero el gráfico de la serie realizando una descomposición de la misma, segundo la función de autocorrelación y tercero se harán pruebas de hipótesis para rechazar o no la hipótesis nula.

5.1.1 Análisis gráfico de la serie de entrenamiento y descomposición

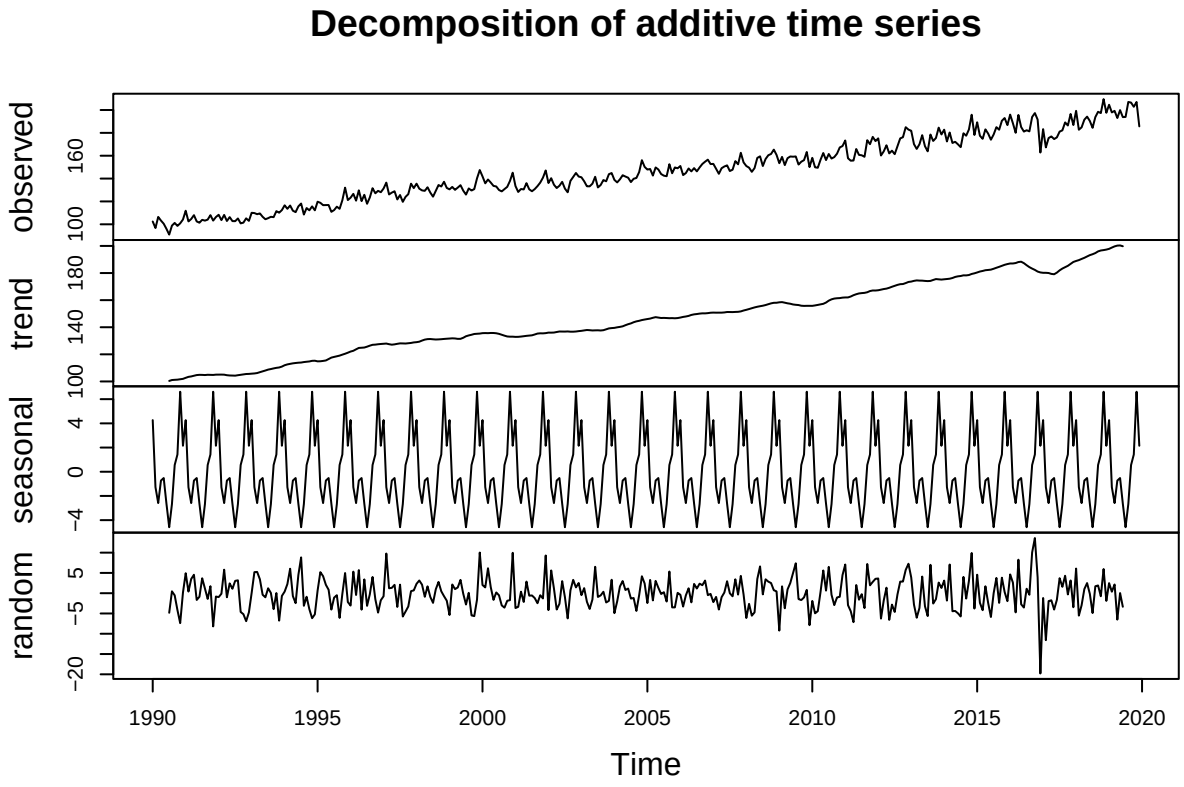
```
autoplot(serie_ent,series = "ICAP entrenamiento")+
  ggtitle("ICAP entrenamiento")+
  xlab("Año")+ylab("ICAP")+
  scale_color_manual(values=c("#469F4B"))+
  theme(legend.position = "none")
```



Visualmente se puede observar que se tiene una tendencia aditiva y posiblemente estacionalidad multiplicativam, además de presentar varianza no constante en el tiempo.

Para tener mejor precisión de lo mencionado se va a descomponer la serie de tiempo.

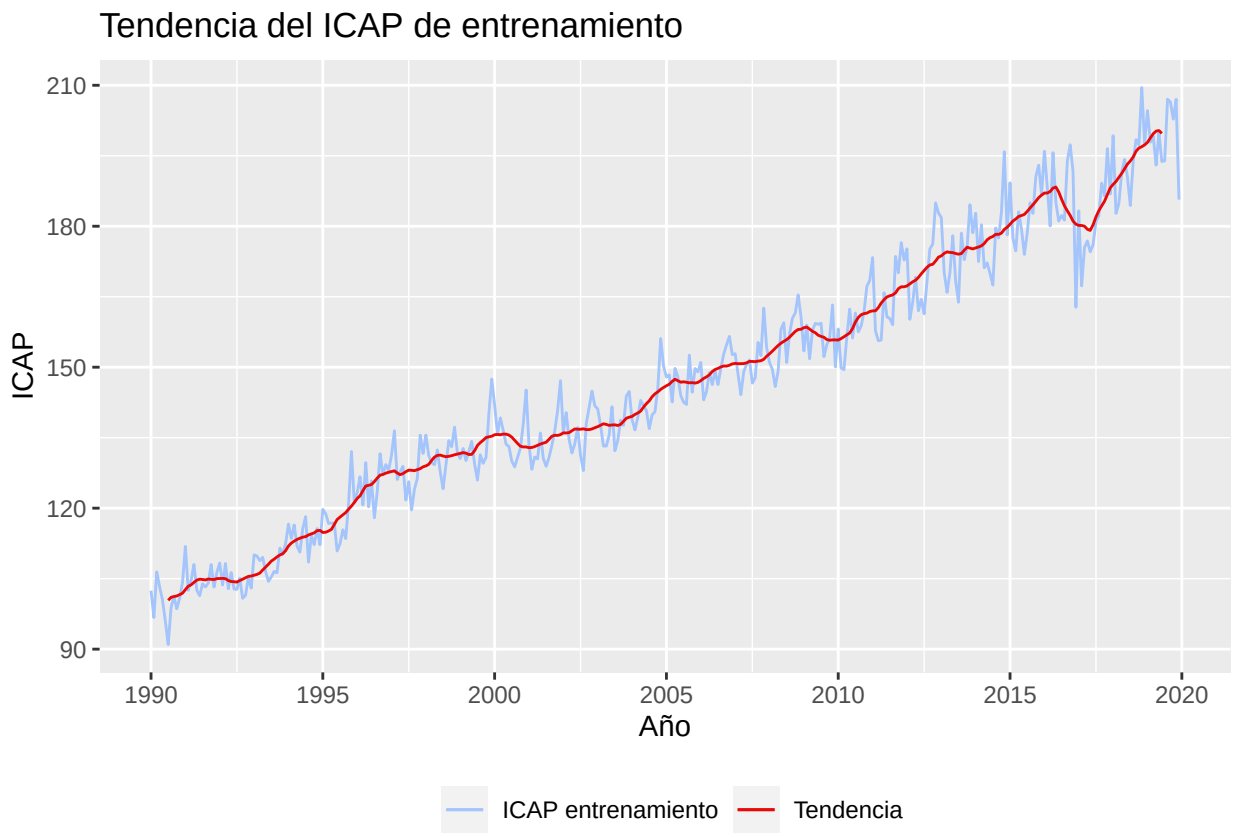
```
descomposicion<-decompose(serie_ent)
plot(descomposicion)
```



En la descomposición se puede observar que existe una tendencia aditiva y efecto estacional.

Para visualizar de mejor manera la tendencia que es la aplicación de una media móvil de orden 12 debido a que las observaciones son mensuales se tiene el siguiente gráfico.

```
autoplot(serie_ent, series = "ICAP entrenamiento")+
  autolayer(descomposicion$trend,series = "Tendencia")+
  ggtitle("Tendencia del ICAP de entrenamiento")+
  xlab("Año")+ylab("ICAP")+
  scale_color_manual(values=c("#A4C4FC","#E80808"))+
  theme(legend.position = "bottom",legend.title = element_blank())
```



Con la media móvil de orden 12 se puede observar en el gráfico la tendencia aditiva de la serie de tiempo, cabe mencionar que en diciembre del 2016 a aproximadamente junio del 2017 hay una rampa decreciente que puede explicarse por un fenómeno climático de sequía y falta de lluvias en Bolivia, lo cual derivó a un desabastecimiento de agua en ocho de los nueve departamentos del país y en 94 barrios del departamento de La Paz existió un racionamiento de agua, esto se detalla de mejor manera el reportaje de CNN en español en su portal web[2].

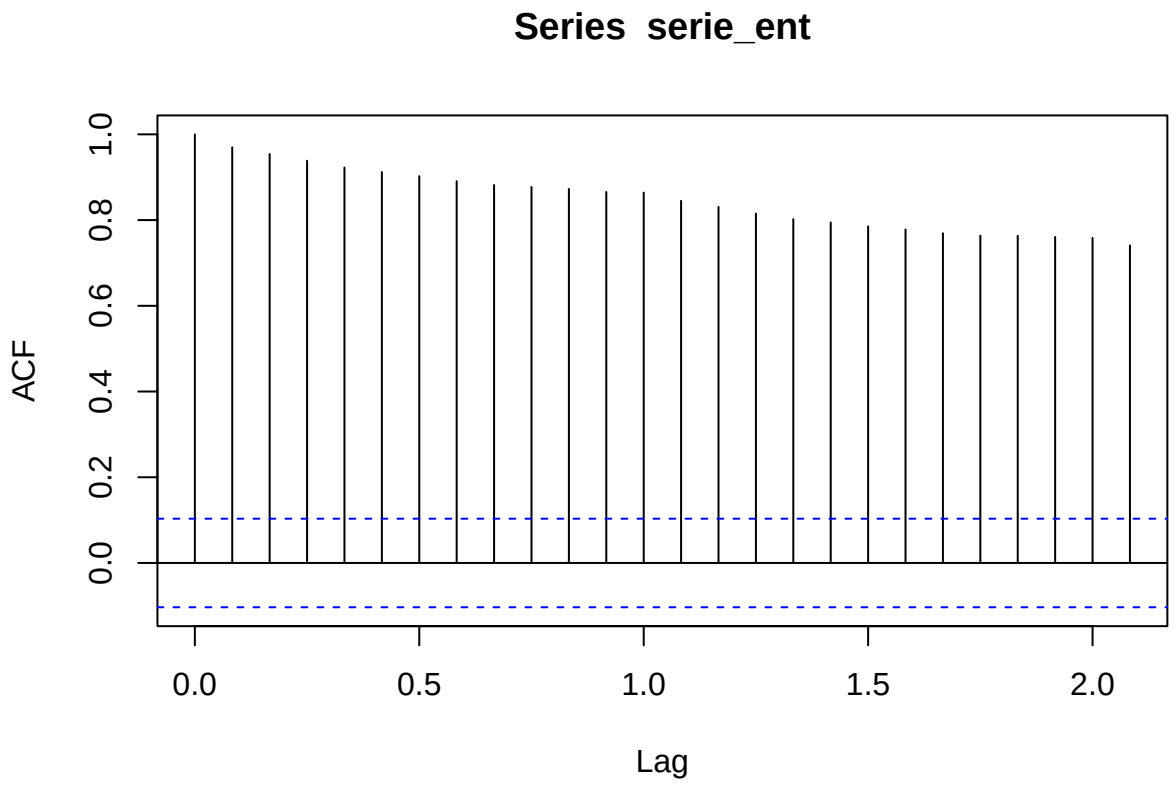
El efecto estacional se estudiara más adelante.

5.1.2 Función de autocorrelación FAC

Ahora se vera el gráfico de la función de autocorrelación (FAC) de la serie de entrenamiento del ICAP.

```
acf(serie_ent)
```

[2]https://cnnespanol.cnn.com/2016/11/21/declaran-emergencia-nacional-en-bolivia-por-sequia-y-falta-de-agua/



El gráfico de la FAC, no presenta un decaimiento exponencial a 0, resultando como conclusión la no estacionariedad de la serie.

5.1.3 Dócimas de estacionariedad

Para la prueba de estacionariedad se utilizará el test de Dickey Fuller Aumentado donde las hipótesis a contrastar son:

$$H_0 : \gamma = 0 \text{ por consiguiente } (\phi_1 = 1)$$
$$H_0 : \gamma \neq 0 \text{ por consiguiente } (\phi_1 < 1)$$

o equivalentemente

$$H_0 : \text{ Existe raíz unitaria (no es estacionaria)}$$
$$H_1 : \text{ No existe raíz unitaria (es estacionaria)}$$

Se probará el contraste con el siguiente comando:

```
adf.test(serie_ent)
```

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

```
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  serie_ent
## Dickey-Fuller = -5.1683, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

Se puede observar que el $p - valor$ es menor a 0.01 por el mensaje de alerta que imprime el software, llegando así a la decisión de rechazar la hipótesis nula, lo cual deriva a que la serie seria estacionaria. Lo cual entra en contradicción con el análisis del gráfico, la descomposición y la FAC, hay que tener en cuenta que los rezagos que esta tomando esta prueba por defecto son 7 y como nuestra serie es mensual realizaremos el test con almenos 12 rezagos.

```
adf.test(serie_ent,k=12)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  serie_ent
## Dickey-Fuller = -2.4064, Lag order = 12, p-value = 0.4054
## alternative hypothesis: stationary
```

Ahora se puede observar que el $p - valor$ es igual a 0.4054 llegando a la decisión de no rechazar la hipóstesis nula, lo cual deriva a que la serie no es estacionaria, no entrando en contradicción con los análisis previos.

Observación:

Para poder determinar desde qué número de rezagos no se rechaza la hipótesis nula se crea la siguiente iteración de la prueba de Dickey Fuller Aumentado:

```
for(i in 1:24){
  test<-adf.test(serie_ent,k=i)
  print(paste0("Nro. de rezagos: ",i,", P-valor:",test$p.value))
}
```

```
## [1] "Nro. de rezagos: 1, P-valor:0.01"
## [1] "Nro. de rezagos: 2, P-valor:0.01"
## [1] "Nro. de rezagos: 3, P-valor:0.01"
## [1] "Nro. de rezagos: 4, P-valor:0.01"
## [1] "Nro. de rezagos: 5, P-valor:0.01"
## [1] "Nro. de rezagos: 6, P-valor:0.01"
## [1] "Nro. de rezagos: 7, P-valor:0.01"
## [1] "Nro. de rezagos: 8, P-valor:0.01"
## [1] "Nro. de rezagos: 9, P-valor:0.0694827876230502"
## [1] "Nro. de rezagos: 10, P-valor:0.210253403771682"
## [1] "Nro. de rezagos: 11, P-valor:0.673748480029247"
## [1] "Nro. de rezagos: 12, P-valor:0.405387633132588"
## [1] "Nro. de rezagos: 13, P-valor:0.30119597888132"
## [1] "Nro. de rezagos: 14, P-valor:0.105985971185532"
## [1] "Nro. de rezagos: 15, P-valor:0.059784928367456"
## [1] "Nro. de rezagos: 16, P-valor:0.0891392926003197"
## [1] "Nro. de rezagos: 17, P-valor:0.0774720272028518"
## [1] "Nro. de rezagos: 18, P-valor:0.156593570792597"
## [1] "Nro. de rezagos: 19, P-valor:0.151222714106116"
## [1] "Nro. de rezagos: 20, P-valor:0.175806338988105"
```

```
## [1] "Nro. de rezagos: 21, P-valor:0.35232953748354"
## [1] "Nro. de rezagos: 22, P-valor:0.524929774977473"
## [1] "Nro. de rezagos: 23, P-valor:0.782796109884554"
## [1] "Nro. de rezagos: 24, P-valor:0.662107301202331"
```

Como se puede observar desde el rezago 9 para adelante no se rechaza la hipótesis nula, pensando así que desde ese número de rezagos la serie presenta no estacionariedad.

5.1.4 Conclusión de estacionariedad

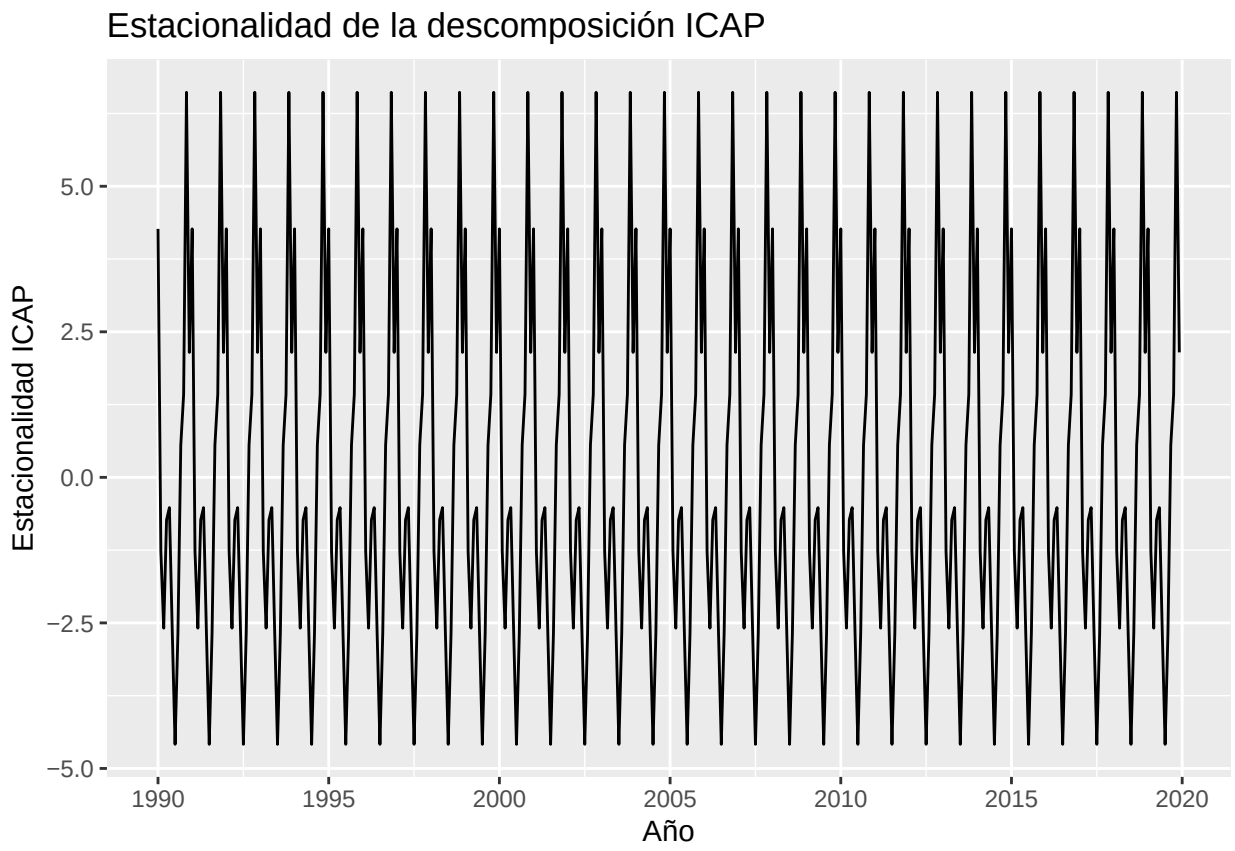
Posterior a realizar un análisis de la serie del ICAP de Bolivia tanto gráfico, por descomposición, la función de autocorrelación FAC y realizar la prueba de Dickey Fuller Aumentado se concluye que la serie es no estacionaria tanto en media como en varianza, presentando así tendencia positiva y varianza no constante.

5.2 Estacionalidad

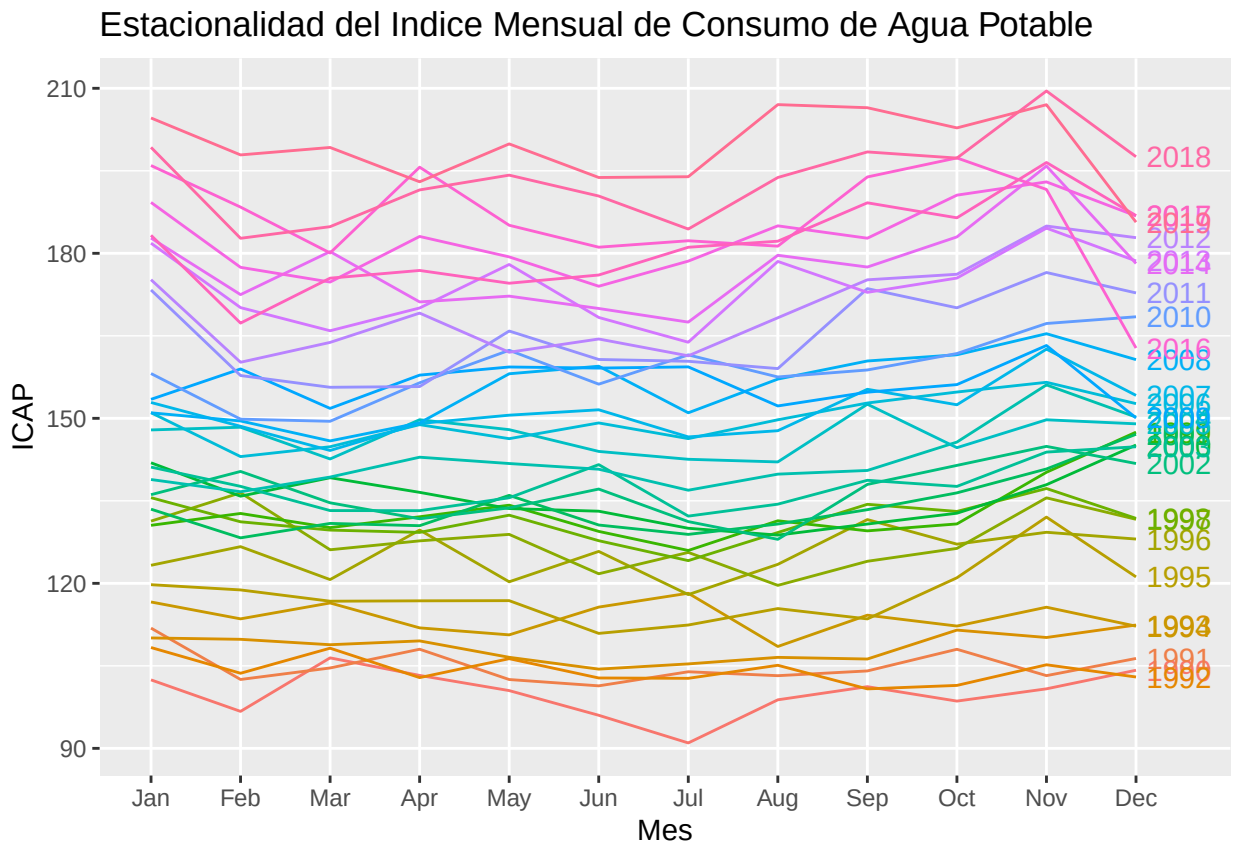
5.2.1 Análisis gráfico

Para determinar de mejor manera el efecto estacional que en primera instancia resulto de la descomposición de la serie de tiempo el cual se refleja en el siguiente gráfico.

```
autoplot(descomposicion$seasonal)+
  ggtitle("Estacionalidad de la descomposición ICAP")+
  xlab("Año")+ylab("Estacionalidad ICAP")
```

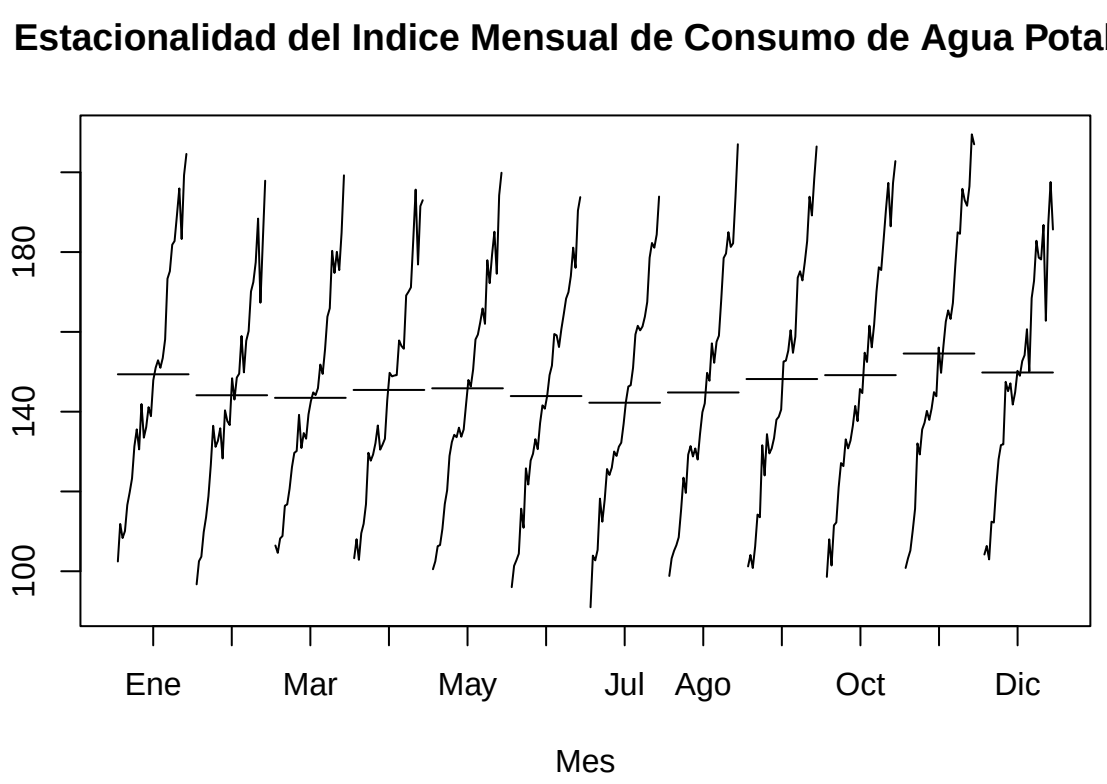


```
ggseasonplot(serie_ent,main="Estacionalidad del Indice Mensual de Consumo de Agua Potable",
          year.labels = T,xlab="Mes",ylab="ICAP")
```



En este gráfico se puede observar en primera instancia que existe una tendencia aditiva lo cual verifica lo mostrado anteriormente, ya que a medida que pasa cada año el consumo de agua potable va aumentando, por otro lado no se presente de manera clara el efecto estacional, pero en el mes de noviembre se puede ver un leve incremento de la serie en estudio.

```
monthplot(serie_ent,ylab="",main="Estacionalidad del Indice Mensual de Consumo de Agua Potable",
          labels = c("Ene","Feb","Mar","Abr","May","Jun","Jul","Ago","Sep","Oct","Nov","Dic"),
          xlab="Mes")
```



En este último gráfico se puede observar que podría existir un efecto estacional ya que en los meses de enero y noviembre asciende de manera muy leve el consumo de agua potable en Bolivia, pero en los demás meses se mantiene casi constantemente.

5.2.2 Conclusión estacionalidad

Se puede concluir entonces con un análisis visual que la serie en estudio del indice de consumo de agua potable en Bolivia tiene efecto de tendencia aditiva con estacionalidad.

5.3 Transformación de la serie temporal

Como se pudo concluir en los análisis previos de estacionariedad y estacionalidad la serie del ICAP de entrenamiento es no estacionario en media y tampoco en varianza, además de tener efecto estacional, por esos motivos se procederá a realizar transformaciones para poder tener una serie de tiempo estacionaria en media y varianza y sin efecto estacional.

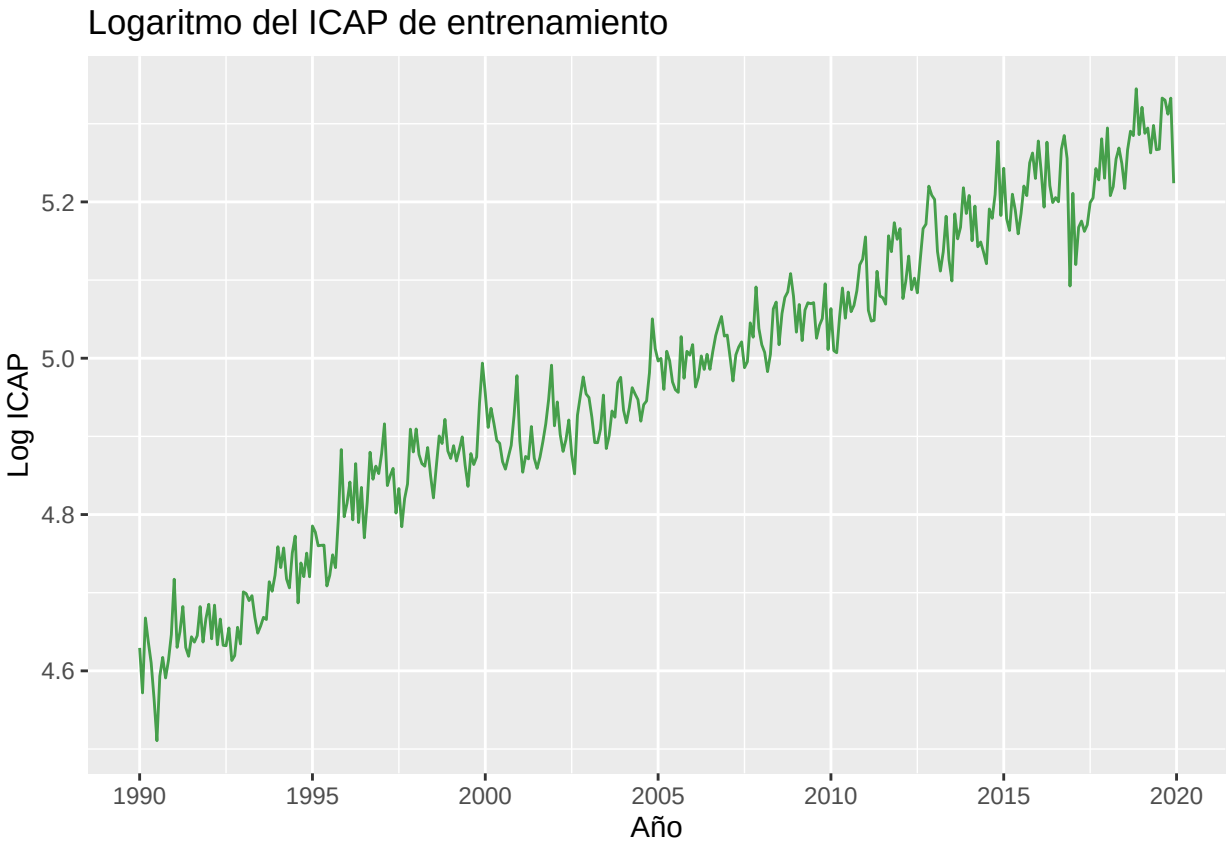
5.3.1 Transformación logaritmo

Para poder tener varianza constante se utilizará la transformación logaritmo a la serie del ICAP de entrenamiento.

```
log_serie_ent<-log(serie_ent)

autoplot(log_serie_ent,series="Logaritmo de ICAP entrenamiento")+
```

```
  ggtitle("Logaritmo del ICAP de entrenamiento")+
  xlab("Año")+ylab("Log ICAP")+
  scale_color_manual(values=c("#469F4B"))+
  theme(legend.position = "none")
```



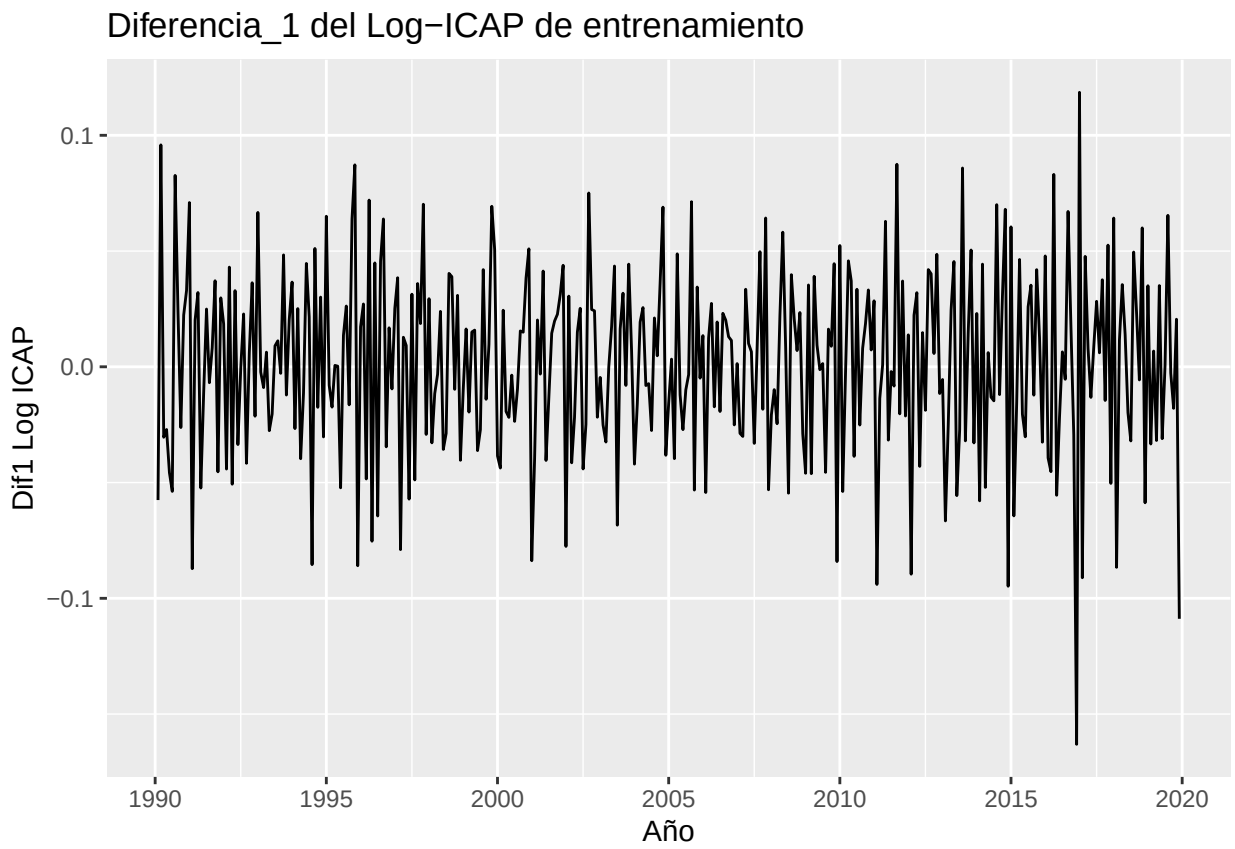
Se puede visualizar que la serie ya no cuenta con varianza creciente conforme pasa el tiempo.

5.3.2 Diferenciación

Como la serie aplicada a logaritmo aún presenta tendencia se hará una diferencia con rezago 1.

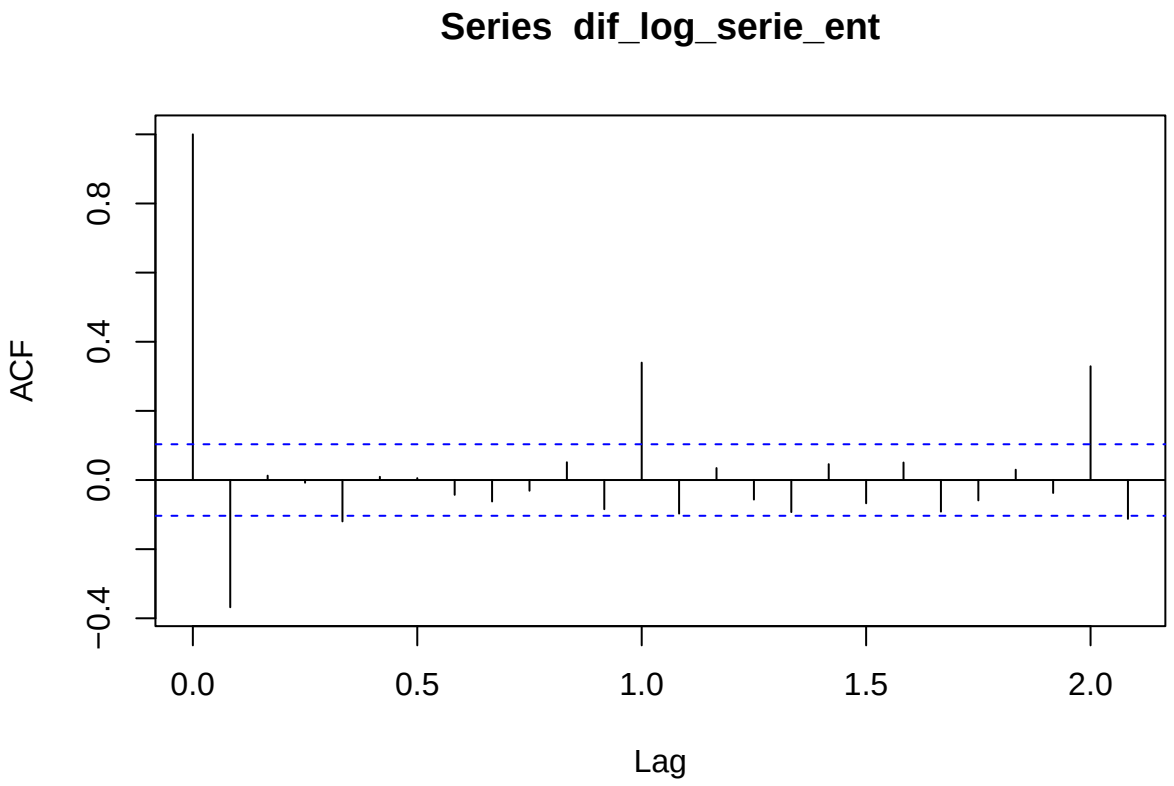
```
dif_log_serie_ent<-diff(log_serie_ent,1)

autoplot(dif_log_serie_ent)+
  ggtitle("Diferencia_1 del Log-ICAP de entrenamiento")+
  xlab("Año")+ylab("Dif1 Log ICAP")+
  scale_color_manual(values=c("#469F4B"))+
  theme(legend.position = "none")
```



La serie gráficamente presenta media constante y su varianza también excpeto por el periodo entre el 2016 al 2017 que aumenta su variabilidad.

```
acf(dif_log_serie_ent)
```



Teniendo el gráfico de la función de autocorrelación si existe decaimiento exponencial a 0, lo cual indica la estacionariedad de la serie de tiempo.

Se hará la prueba de Dickey Fuller Aumentada para verificar estacionariedad.

```
adf.test(dif_log_serie_ent,k=12)
```

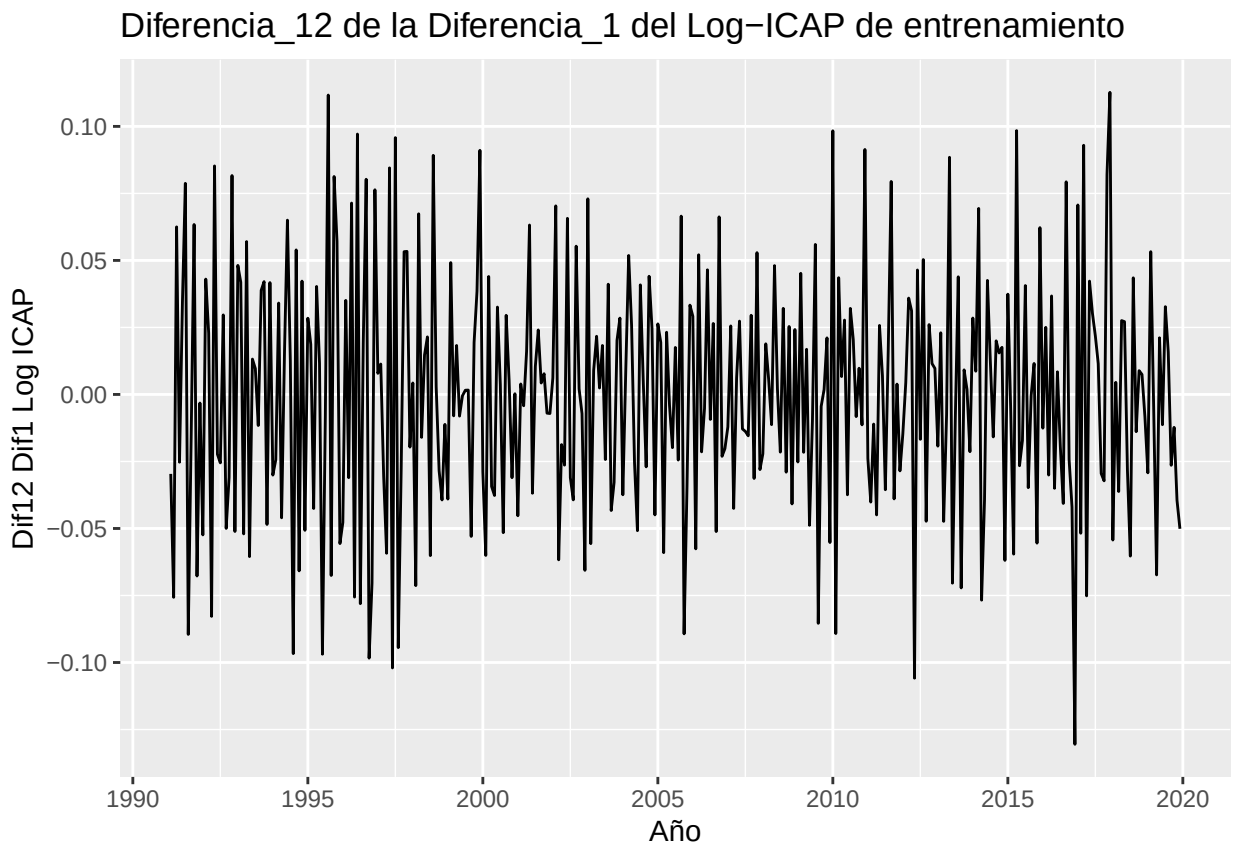
```
##
##  Augmented Dickey-Fuller Test
##
## data:  dif_log_serie_ent
## Dickey-Fuller = -7.6033, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

Se rechaza la hipótesis nula donde se concluye que la serie es estacionaria, tomando solo una diferencia.

Pero como se analizó anteriormente la serie tenia presumiblemente estacionalidad lo cual se podría quitar con una diferenciación de 12 rezagos, a la serie aplicada con logaritmo y diferenciada una vez con 1 rezago.

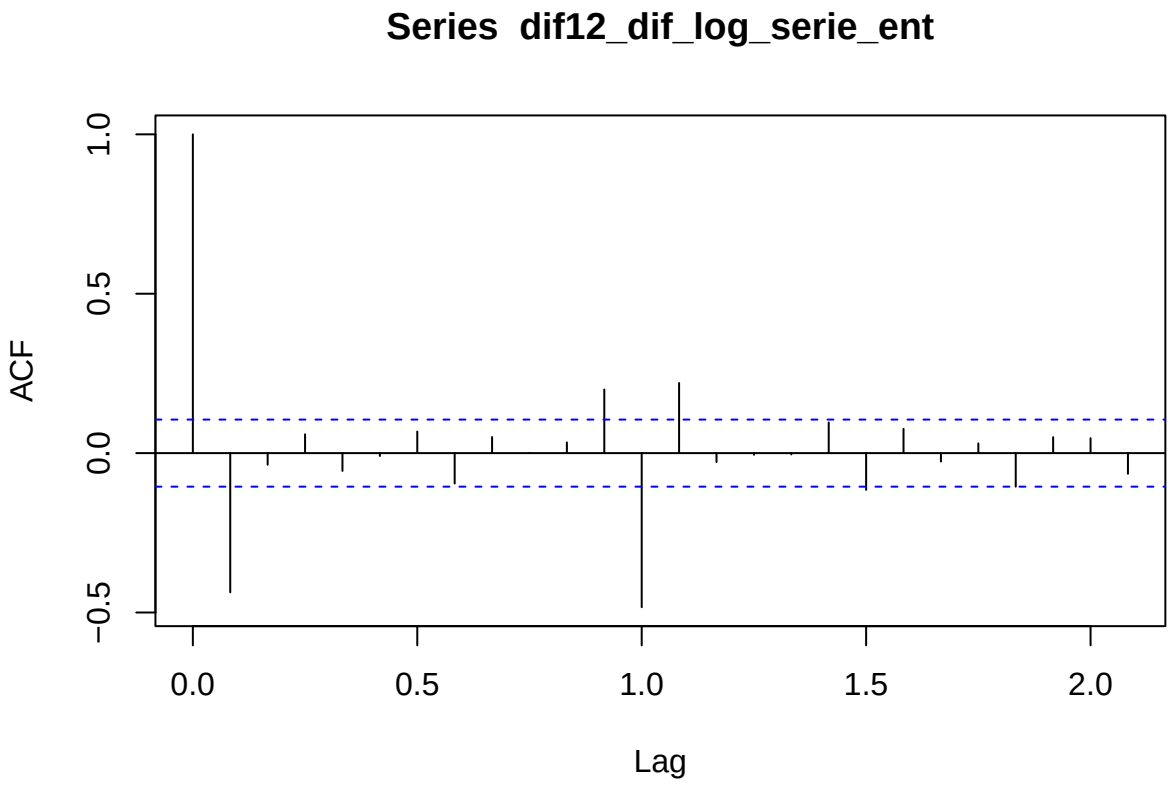
```
dif12_dif_log_serie_ent<-diff(dif_log_serie_ent,12)

autoplot(dif12_dif_log_serie_ent)+
  ggtitle("Diferencia_12 de la Diferencia_1 del Log-ICAP de entrenamiento")+
  xlab("Año")+ylab("Dif12 Dif1 Log ICAP")+
  scale_color_manual(values=c("#469F4B"))+
  theme(legend.position = "none")
```



La serie no presenta tendencia, teniendo media y varianza constante, y tampoco presenta efecto estacional, donde visualmente estaríamos en frente una serie estacionaria.

```
acf(dif12_dif_log_serie_ent)
```



Teniendo el gráfico de la función de autocorrelación existe decaimiento exponencial a 0, lo cual indica la estacionariedad de la serie de tiempo.

Para comprobar esto probamos el test de Dickey Fuller Aumentado.

```
adf.test(dif12_dif_log_serie_ent,k=12)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dif12_dif_log_serie_ent
## Dickey-Fuller = -6.7913, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

Se rechaza la hipótesis nula de raíz unitaria teniendo así una serie estacionaria.

5.3.3 Conclusión de las transformaciones

Se puede concluir que se tiene una serie estacionaria aplicando logaritmo y una diferenciación con rezago igual a 1 a la serie de tiempo del ICAP de Bolivia. No se ve necesario aplicar la diferenciación con rezago igual a 12 posterior al logaritmo y a la diferenciación con 1 rezago.

Para consolidar lo que se menciono se procederá a utilizar el comando en R que nos ayuda a ver cuantas diferencias se necesita y así también en la estacionalidad.

```
paste0("Se necesita diferenciar: ",ndiffs(log_serie_ent)," vez")
```

```
## [1] "Se necesita diferenciar: 1 vez"
```

```
paste0("Se necesita diferenciar en la parte estacional: ",nsdiffs(log_serie_ent)," veces")
```

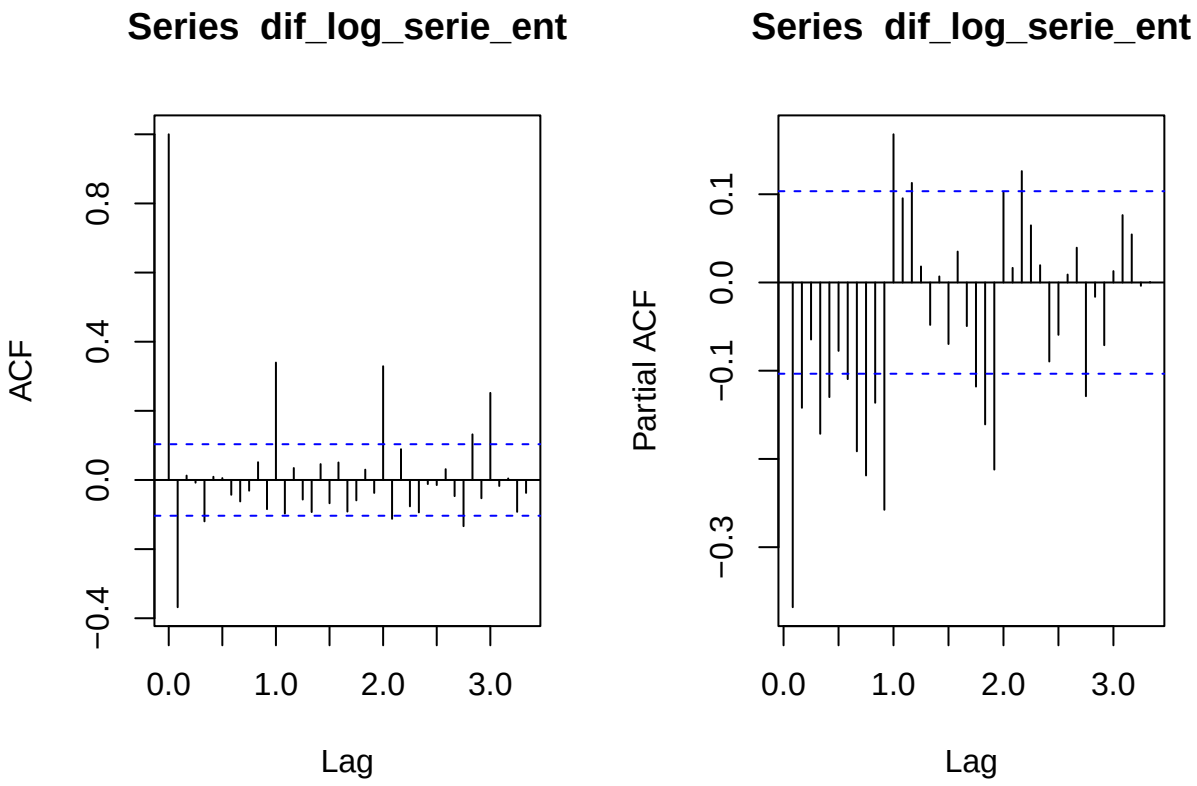
```
## [1] "Se necesita diferenciar en la parte estacional: 0 veces"
```

5.4 Ajuste del Modelo

Como se pudo determinar en la transformación de la serie se aplico logaritmo para estabilizar varianza y una diferencia de rezago 1 para que sea estacionaria. No es necesario diferenciar con rezago 12.

Se grafica las FAC y FACP de la serie transformada.

```
par(mfrow=c(1,2))
acf(dif_log_serie_ent,lag=40)
pacf(dif_log_serie_ent,lag=40)
```



De estas gráficas se puede pensar en estimar los siguientes modelos:

1) $SARIMA(2,1,1)(1,0,1)_{12}$
2) $SARIMA(4,1,1)(0,0,1)_{12}$

5.4.1 Modelo 1

$SARIMA(2,1,1)(1,0,1)_{12}$

```
significancia<-function(modelo,se_modelo){
  prueba_coef<-c()
  for (i in 1:length(modelo)) {
    prueba_coef[i]<-modelo[i]/sqrt(se_modelo[i,i])
    print(paste0(names(modelo[i]),"  :  ", prueba_coef[i]))
  }
}
```

```
sarima1<-Arima(log_serie_ent,order = c(2,1,1),seasonal = c(1,0,1))
sarima1
```

```
## Series: log_serie_ent
## ARIMA(2,1,1)(1,0,1)[12]
##
## Coefficients:
##          ar1     ar2      ma1    sar1     sma1
##       0.1886  0.0816  -0.8416  0.9494  -0.7238
## s.e.  0.0890  0.0764   0.0680  0.0229   0.0552
##
## sigma^2 estimated as 0.0008972:  log likelihood=747.76
## AIC=-1483.52   AICc=-1483.28   BIC=-1460.22
```

```
significancia(sarima1$coef,sarima1$var.coef)
```

```
## [1] "ar1  :  2.11860618467722"
## [1] "ar2  :  1.06775297381627"
## [1] "ma1  :  -12.37978439386"
## [1] "sar1  :  41.4111705610035"
## [1] "sma1  :  -13.1123828669638"
```

Al verificar la significancia de los parámetros estimados que resultan de la división del parámetro estimado entre la desviación estándar del mismo se tiene aproximadamente que todos los mayores a 2 en valor absoluto son significativos.

Teniendo así que en el ajuste del primer modelo $SARIMA(2,1,1)(1,0,1)_{12}$ el coeficiente autoregresivo de orden 2 AR(2) es no significativo para el modelo, se estima el mismo quitando ese estimador resultando así un modelo $SARIMA(1,1,1)(1,0,1)_{12}$

```
sarima1<-Arima(log_serie_ent,order = c(1,1,1),seasonal = c(1,0,1))
sarima1
```

```
## Series: log_serie_ent
## ARIMA(1,1,1)(1,0,1)[12]
##
## Coefficients:
##          ar1      ma1    sar1     sma1
##       0.1282  -0.7780  0.9499  -0.7213
## s.e.  0.0860   0.0624  0.0223   0.0544
```

```
##
## sigma^2 estimated as 0.0008969:  log likelihood=747.24
## AIC=-1484.48   AICc=-1484.31   BIC=-1465.06
```

```
significancia(sarima1$coef,sarima1$var.coef)
```

```
## [1] "ar1 :  1.49046872443294"
## [1] "ma1 :  -12.4698284920309"
## [1] "sar1 :  42.5731003947495"
## [1] "sma1 :  -13.2617744415541"
```

Como se puede observar ahora el coeficiente AR(1) es no significativo para el modelo, por tanto se estima un modelo $SARIMA(0,1,1)(1,0,1)_{12}$

```
sarima1<-Arima(log_serie_ent,order = c(0,1,1),seasonal = c(1,0,1))
sarima1
```

```
## Series: log_serie_ent
## ARIMA(0,1,1)(1,0,1)[12]
##
## Coefficients:
##           ma1   sar1     sma1
##       -0.6971  0.949  -0.7131
## s.e.   0.0463  0.022   0.0535
##
## sigma^2 estimated as 0.0008996:  log likelihood=746.16
## AIC=-1484.32   AICc=-1484.21   BIC=-1468.79
```

```
significancia(sarima1$coef,sarima1$var.coef)
```

```
## [1] "ma1 :  -15.0573242513155"
## [1] "sar1 :  43.1679830584975"
## [1] "sma1 :  -13.3163637305963"
```

Así para el modelo $SARIMA(0,1,1)(1,0,1)_{12}$ todos los parámetros estimados son significativos. Para este modelo estimado se tiene un Criterio de información de Akaike AIC

```
AIC_sarima1<-sarima1$aic
AIC_sarima1
```

```
## [1] -1484.323
```

5.4.2 Modelo 2

$SARIMA(4,1,1)(0,0,1)_{12}$

```
sarima2<-Arima(log_serie_ent,order = c(4,1,1),seasonal = c(0,0,1))
sarima2
```

```
## Series: log_serie_ent
## ARIMA(4,1,1)(0,0,1)[12]
##
## Coefficients:
##          ar1     ar2     ar3      ar4      ma1    sma1
##       0.2641  0.1091  0.0088  -0.1117  -0.8083  0.2739
## s.e.  0.0763  0.0621  0.0591   0.0582   0.0579  0.0447
##
## sigma^2 estimated as 0.001149:  log likelihood=707.86
## AIC=-1401.72   AICc=-1401.4   BIC=-1374.53
```

```
significancia(sarima2$coef,sarima2$var.coef)
```

```
## [1] "ar1 :  3.46271282429991"
## [1] "ar2 :  1.75565415777994"
## [1] "ar3 :  0.148199798046032"
## [1] "ar4 :  -1.91822355678645"
## [1] "ma1 :  -13.9645264590549"
## [1] "sma1 :  6.12723271581687"
```

Se tiene que en el ajuste del modelo $SARIMA(4,1,1)(0,0,1)_{12}$ el coeficiente autoregresivo de orden $2,3,4$ AR(2), AR(3) y AR(4) es no significativo para el modelo, se estima el mismo quitando ese estimador resultando así un modelo $SARIMA(1,1,1)(0,0,1)_{12}$

```
sarima2<-Arima(log_serie_ent,order = c(1,1,1),seasonal = c(0,0,1))
sarima2
```

```
## Series: log_serie_ent
## ARIMA(1,1,1)(0,0,1)[12]
##
## Coefficients:
##          ar1      ma1    sma1
##       0.289  -0.8072  0.2823
## s.e.  0.076   0.0468  0.0437
##
## sigma^2 estimated as 0.001164:  log likelihood=704.08
## AIC=-1400.17   AICc=-1400.05   BIC=-1384.63
```

```
significancia(sarima2$coef,sarima2$var.coef)
```

```
## [1] "ar1 :  3.80309121747723"
## [1] "ma1 :  -17.2641244648079"
## [1] "sma1 :  6.45876424966251"
```

Así para el modelo $SARIMA(1,1,1)(0,0,1)_{12}$ todos los parámetros estimados son significativos.

El Criterio de información de Akaike AIC es:

```
AIC_sarima2<-sarima2$aic
AIC_sarima2
```

```
## [1] -1400.166
```

5.4.3 Modelo 3

Para el tercer modelo se estimará un con un comando en R que genera un proceso automático en el ajuste de los modelos Box y Jenkins.

```
arima_automatico<-auto.arima(log_serie_ent)
arima_automatico
```

```
## Series: log_serie_ent
## ARIMA(2,1,2)(1,0,0)[12] with drift
##
## Coefficients:
##          ar1     ar2      ma1     ma2    sar1   drift
##       0.6291  0.0582  -1.2687  0.2870  0.4291  0.0019
## s.e.  0.2795  0.1264   0.2804  0.2655  0.0543  0.0002
##
## sigma^2 estimated as 0.001018:  log likelihood=728.64
## AIC=-1443.28   AICc=-1442.96   BIC=-1416.1
```

```
significancia(arima_automatico$coef,arima_automatico$var.coef)
```

```
## [1] "ar1 :  2.25088242862455"
## [1] "ar2 :  0.46019747200534"
## [1] "ma1 :  -4.52468215774098"
## [1] "ma2 :  1.08093830360389"
## [1] "sar1 :  7.90064003626897"
## [1] "drift :  9.5732496543853"
```

Los parámetros estimados AR(2) y MA(2) no son significativos.

```
AIC_arima_automatico<-arima_automatico$aic
AIC_arima_automatico
```

```
## [1] -1443.281
```

5.4.4 Decisión

Se pudo encontrar tres posibles modelos dos de los cuales tienen todos sus coeficientes significativos, para tomar la decisión de cual modelo elegir se utilizará el Criterio de Información de Akaike AIC.

$SARIMA(0,1,1)(1,0,1)_{12}$

```
AIC_sarima1
```

```
## [1] -1484.323
```

$SARIMA(1,1,1)(0,0,1)_{12}$

```
AIC_sarima2
```

```
## [1] -1400.166
```

$SARIMA(5,1,0)(2,0,0)_{12}$

```
AIC_arima_automatico
```

```
## [1] -1443.281
```

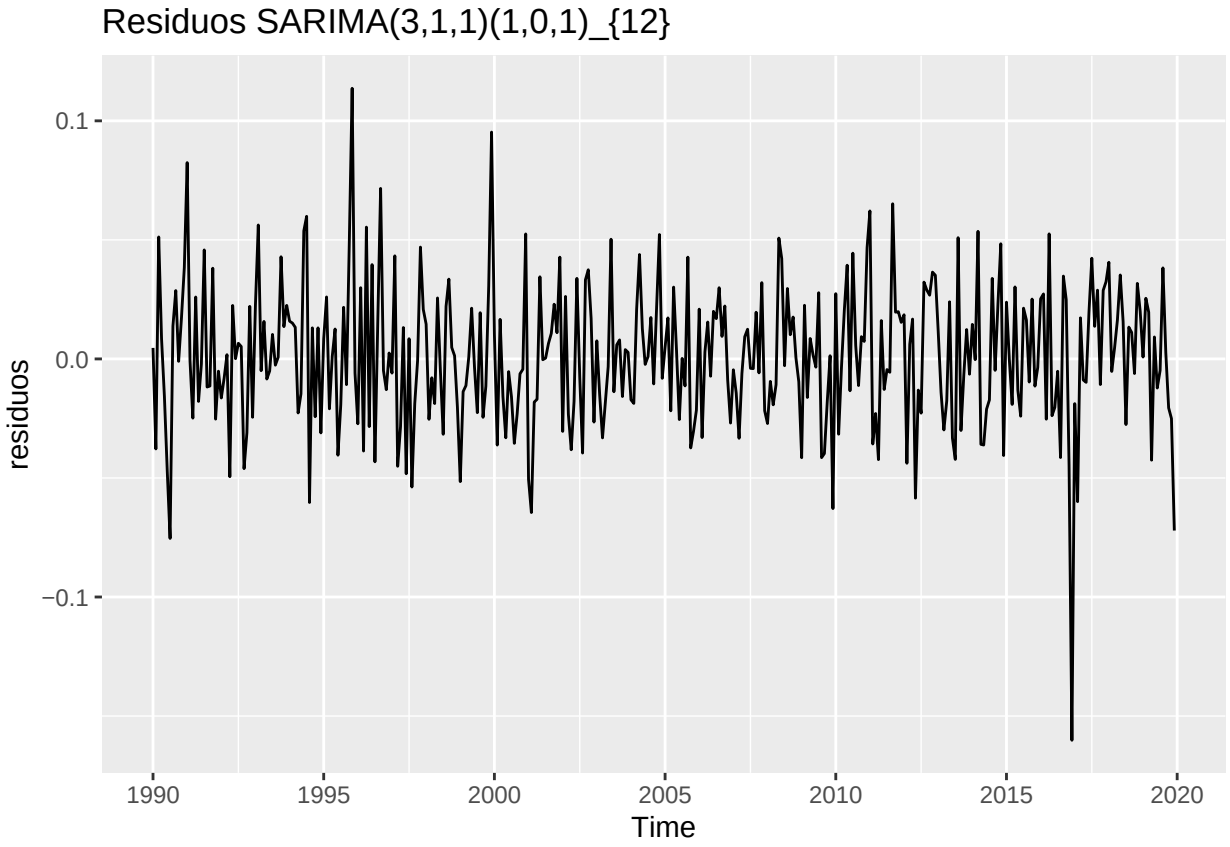
El que tiene mejor AIC es el modelo $SARIMA(0,1,1)(1,0,1)_{12}$, por tal motivo tomaremos este para realizar la validación de supuestos.

5.5 Validación de supuestos

5.5.1 Independencia de Residuos

Graficamos los residuos.

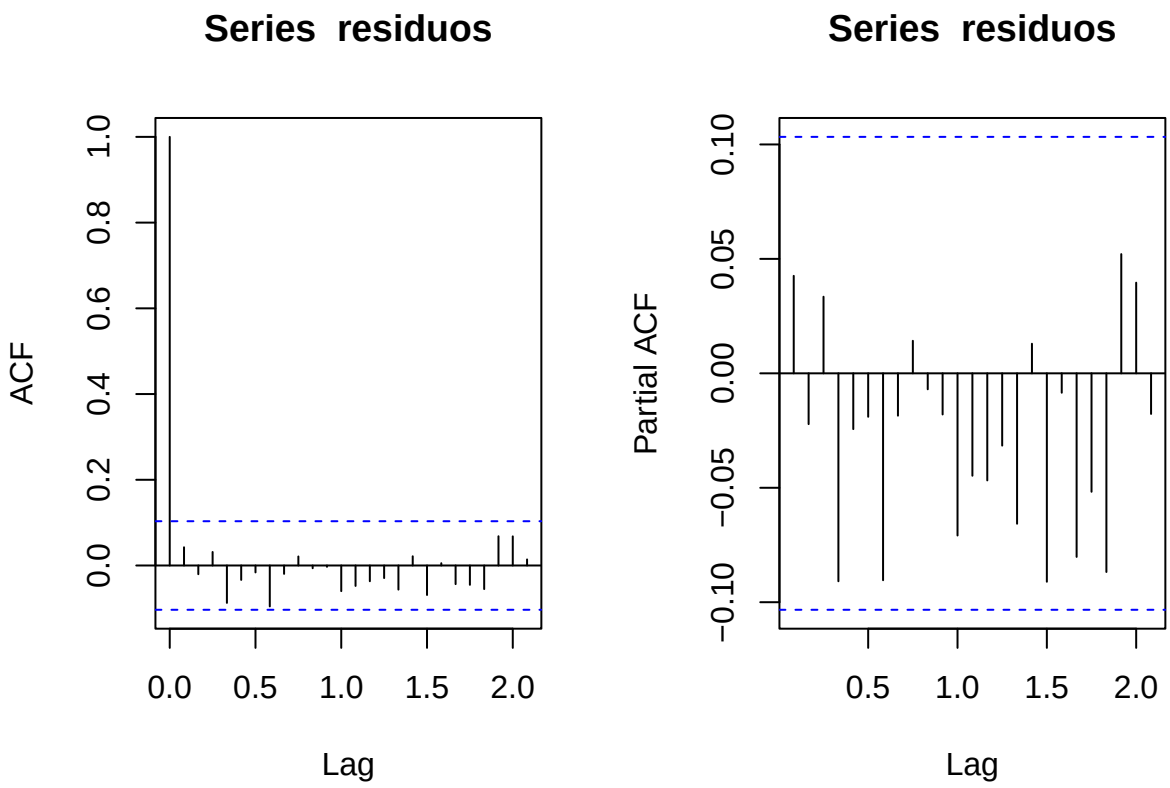
```
residuos<-sarima1$residuals
autoplot(residuos)+
  ggtitle("Residuos SARIMA(3,1,1)(1,0,1)_{12}")
```



- **FAC y FACP**

En primera instancia los residuos deben ser semejantes a un ruido blanco, donde los coeficientes estimados de la FAC y FACP no deben ser significativamente distintos de cero.

```
par(mfrow=c(1,2))
acf(residuos)
pacf(residuos)
```



Coomo se puede visualizar todos los coeficientes estimados de la FAC y FACP de los residuos son aproximadamente cero, teniendo así posiblemente ruido blanco.

- **Dócima de Box-Pierce y Ljung-Box**

Se utilizará en primera instancia la prueba de Box-Pierce y Ljung-Box que tienen como hipótesis:

$$H_0 : \rho_1 = \rho_2 = ... = \rho_k = 0 \text{ (independencia)}$$

```
Box.test(residuos,type = "Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  residuos
## X-squared = 0.65304, df = 1, p-value = 0.419
```

Para la prueba de Box-Pierce no se rechaza H_0 teniendo así independencia en los residuos.

```r
Box.test(residuos,type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuos
## X-squared = 0.6585, df = 1, p-value = 0.4171
```

Para la prueba de Ljung-Box no se rechaza H_0 teniendo así independencia en los residuos.

- **Dócima de Durbin-Watson**

Ahora se utilizará la prueba de Durbin-Watson donde se tendrían los siguientes supuestos:

a) El modelo debe incluir un intercepto
b) Los errores están generados mediante un esquema autoregresivo de orden 1.

La hipótesis serán:

$$H_0 : \phi_1 = 0 \text{ (No hay autocorrelación)}$$
$$H_1 : \phi_1 \neq 0 \text{ (Hay autocorrelación)}$$

```r
residuos_fit<-lm(residuos[-c(1,length(residuos))]~1)
dwtest(residuos_fit,alternative="two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  residuos_fit
## DW = 1.9177, p-value = 0.4352
## alternative hypothesis: true autocorrelation is not 0
```

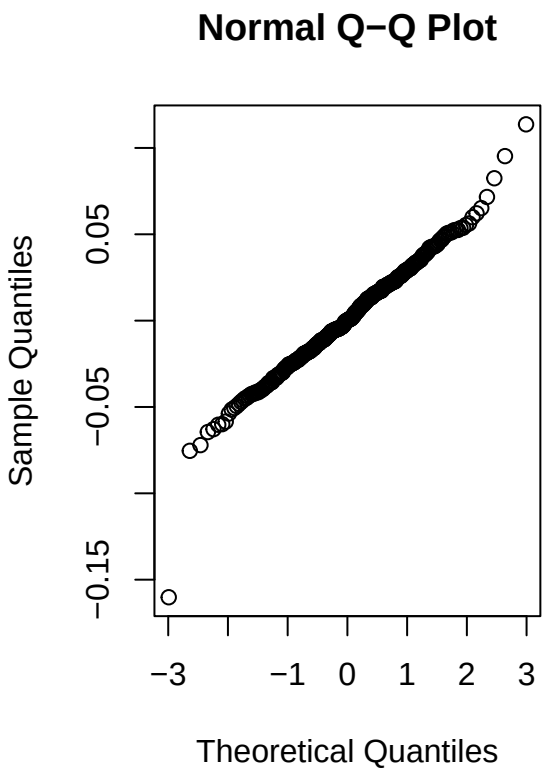
Como $DW = 1.9177$ y el $p - valor > 0.05$ por lo tanto no se rechaza la hipótesis nula, no existiendo autocorrelación de orden 1.

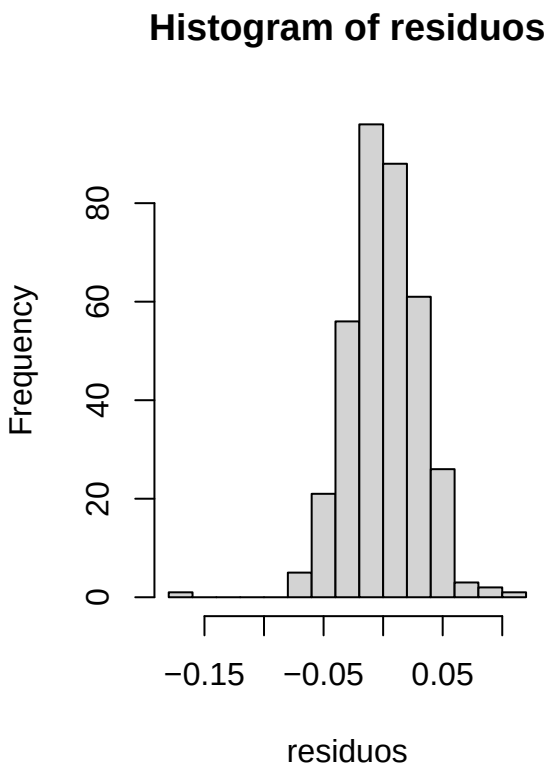
Conclusión: Existe evidencia estadística para decir que los residuos son independientes y no autocorrelacionados.

5.5.2 Normalidad de Residuos

Se verifica en los residuos QQ-normal y el histograma.

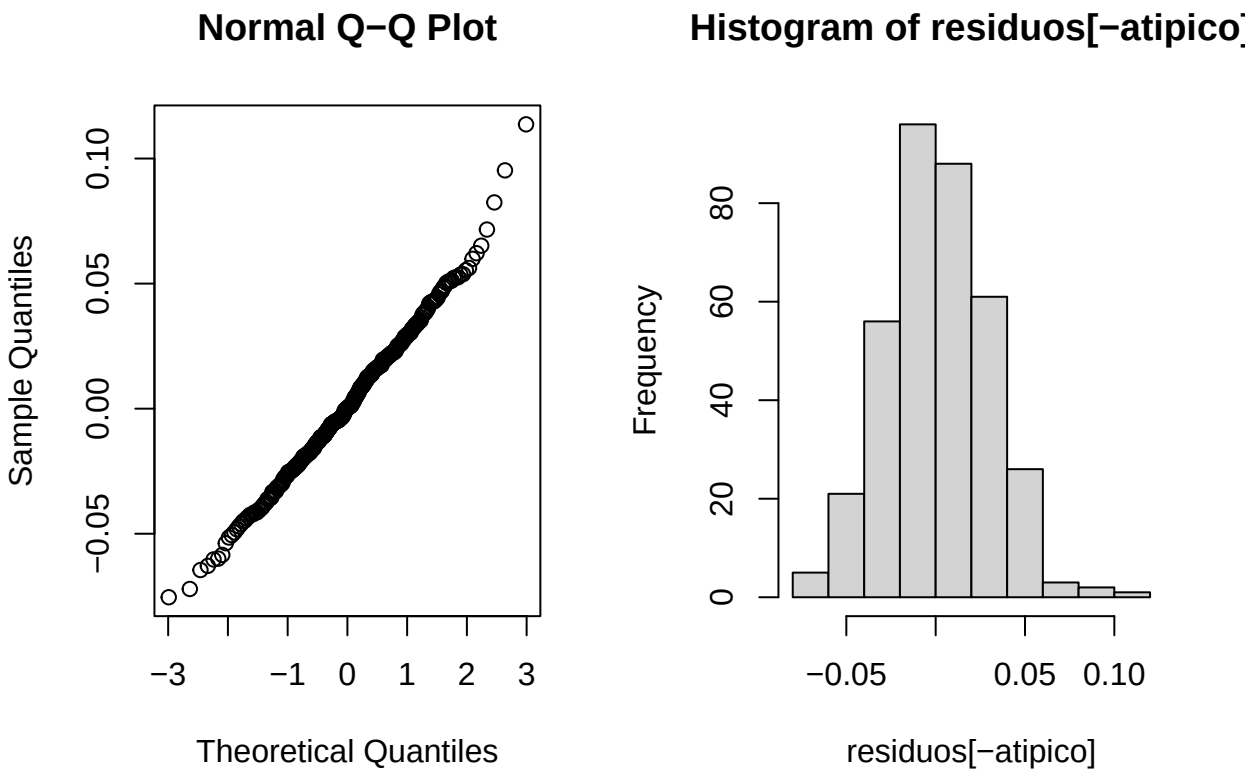
```r
par(mfrow=c(1,2))
qqnorm(residuos)
hist(residuos)
```





Se puede observar la existencia de datos atípicos, efectivamente si se ven los residuos a más detalle existe un valor atípico en diciembre del 2016, el cual se mencionó en el estudio de tendencia un decaimiento bastante fuerte a lo usual de la serie temporal. Este valor atípico puede que este produciendo algún tipo de sesgo en el estudio de normalidad.

```
atipico<-which.min(residuos)
par(mfrow=c(1,2))
qqnorm(residuos[-atipico])
hist(residuos[-atipico])
```



Visualmente al quitar este valor atípico se puede observar que posiblemente exista normalidad en los residuos. Para poder determinar el mismo se realizarán las dócimas de normalidad con este valor atípico y sin él.

Se realizará las pruebas de normalidad.

- **Dócima de Jarque-Bera**

La dócima de Jarque-Bera tiene la siguiente hipótesis nula

$$H_0: \text{ Los residuos son normales}$$

```
normtest::jb.norm.test(residuos)
```

```
##
##  Jarque-Bera test for normality
##
## data:  residuos
## JB = 76.727, p-value < 2.2e-16
```

Se rechaza la hipótesis nula donde se concluye que los residuos no cumplen normalidad como se pudo observar en el gráfico.

```
normtest::jb.norm.test(residuos[-atipico])
```

```
##
##  Jarque-Bera test for normality
##
## data:  residuos[-atipico]
## JB = 4.8267, p-value = 0.077
```

Sin contar con este dato se puede observar que no se rechaza la hipótesis nula, teniendo así normalidad en los residuos.

- **Dócima de Shapiro-Wilk**

La dócima de Shapiro-Wilk tiene la siguiente hipótesis nula

$$H_0 : \text{La distribución es normal}$$

```
shapiro.test(residuos)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos
## W = 0.98086, p-value = 0.0001029
```

Se rechaza la hipótesis nula donde se concluye que los residuos no cumplen normalidad.

```
shapiro.test(residuos[-atipico])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos[-atipico]
## W = 0.99465, p-value = 0.2451
```

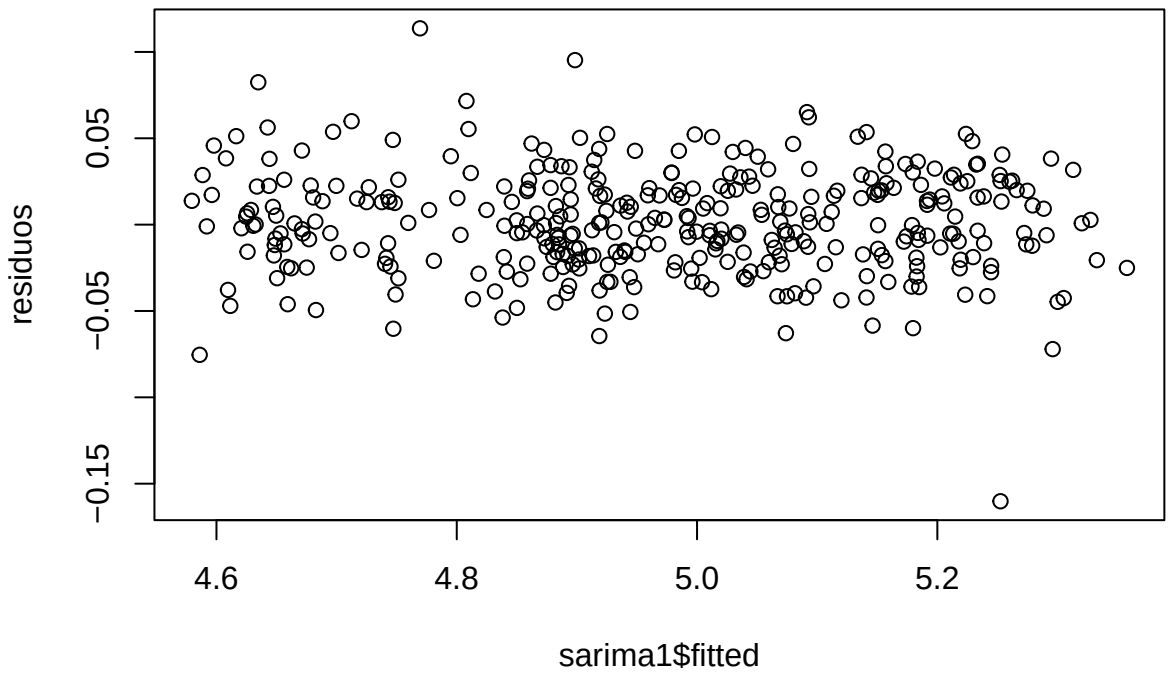
No se rechaza la hipótesis nula donde se concluye que los residuos cumplen normalidad.

Conclusión: Se puede evidenciar que no existe normalidad en los residuos teniendo en cuenta el valor atípico sucitado en diciembre del 2016. No tomando en cuenta este valor se puede decir que los residuos son normales

5.5.3 Homocedasticidad de Residuos

Se hará primero un análisis gráfico.

```
plot(sarima1$fitted,residuos)
```



Se puede evidenciar que existe un valor atípico el cual corresponde a diciembre del 2016, pero según el gráfico existiria homocedasticidad, sin contar con ese valor atípico.

Dócima de White

La hipótesis a contrastar es:

$$H_0: \text{ Residuos son homocedasticos}$$

```
white.test(residuos)
```

```
##
##  White Neural Network Test
##
## data:  residuos
## X-squared = 2.1013, df = 2, p-value = 0.3497
```

Se concluye que no se rechaza la hipótesis nula, teniendo así evidencia estadística para decir que los residuos son homocedásticos.

Dócima de Breush-Pagan

La hipótesis a contrastar es:

$$H_0: \text{ Residuos son homocedasticos}$$

```
residuos_fit<-lm(residuos[-1] ~ residuos[-length(residuos)])
bptest(residuos_fit)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  residuos_fit
## BP = 0.0039868, df = 1, p-value = 0.9497
```

No se rechaza la hipótesis nula.

Conclusión: Existe evidencia estadística para decir que los residuos son homocedasticos, no presentando así heterocedasticidad.

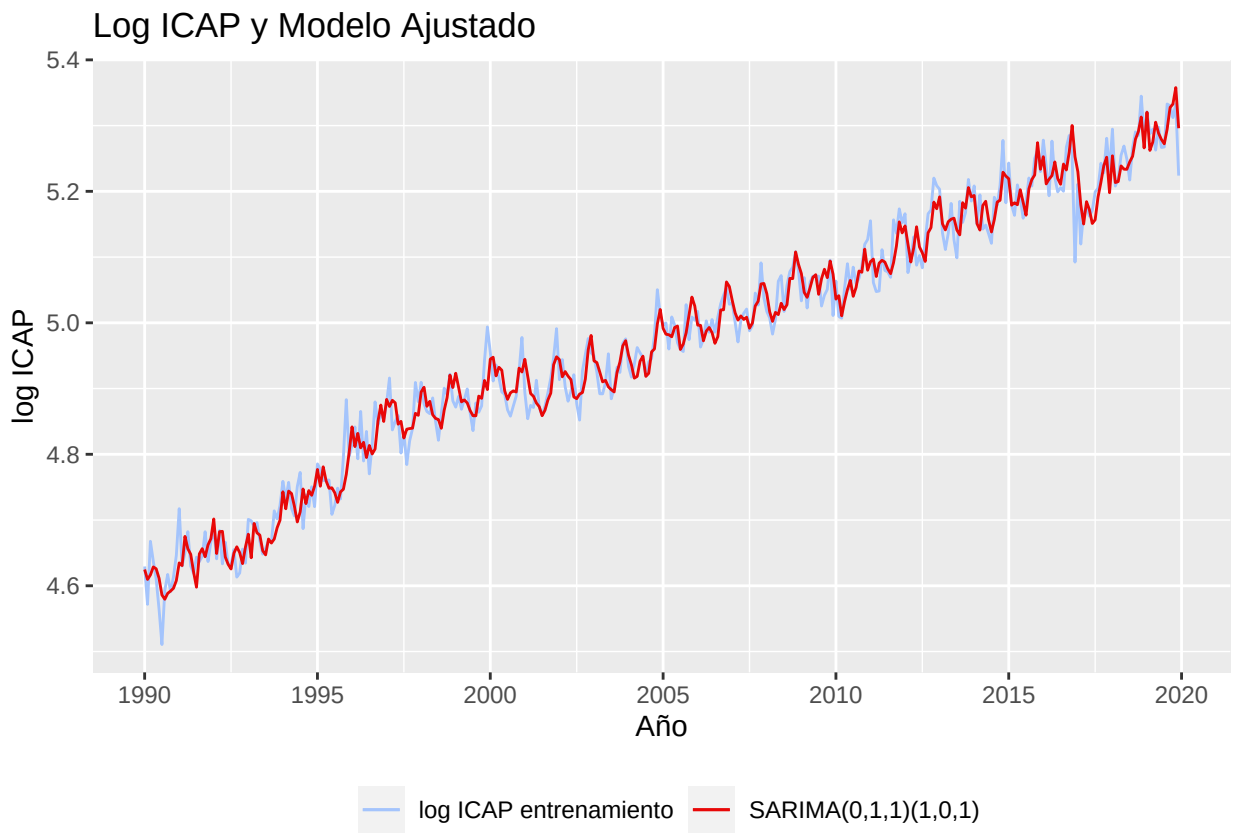
5.6 Conclusión ajuste del modelo

Se puede concluir entonces que el modelo elegido $SARIMA(0,1,1)(1,0,1)_{12}$ tiene el menor AIC teniendo residuos independientes y homocedásticos, en cuanto a la normalidad sin contar con el valor atípico del 2016 se puede asumir este supuesto.

6 Gráfico serie original y el ajustado

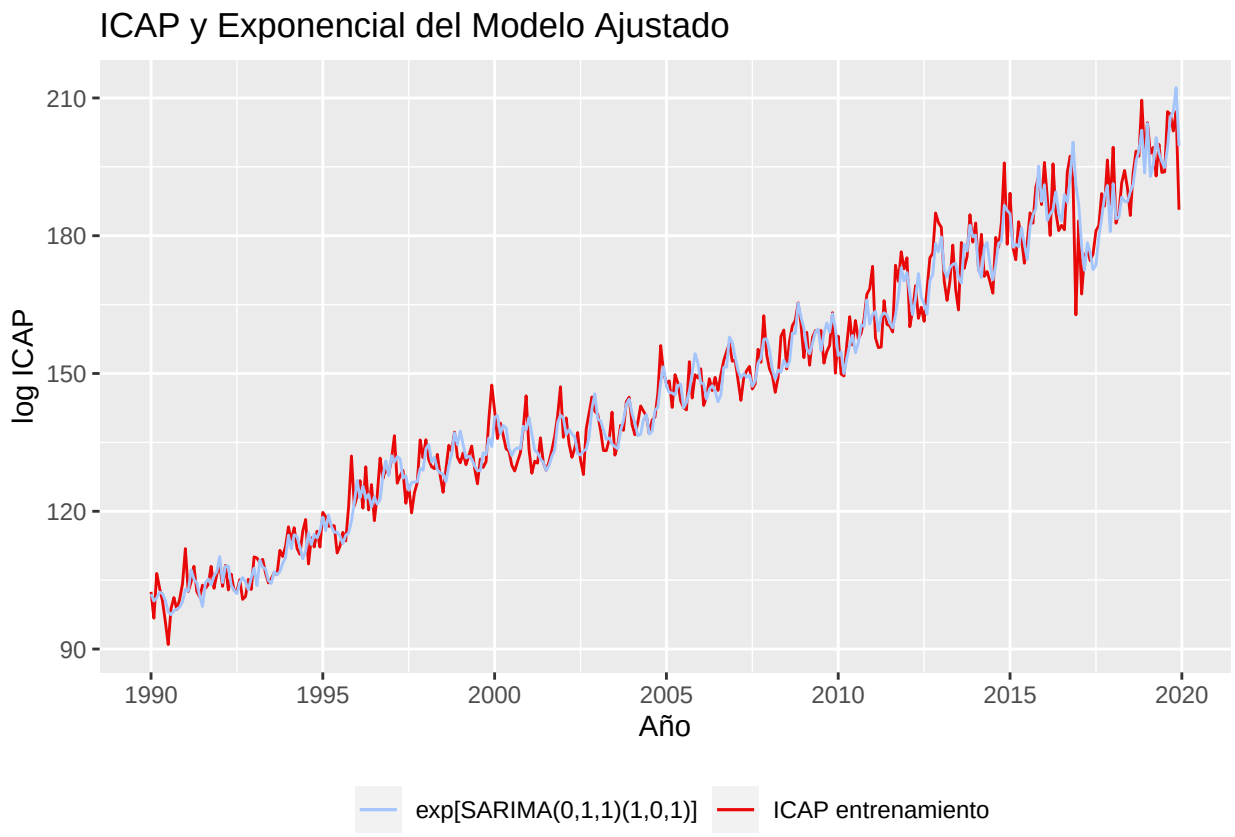
Primero gráficamos la serie del ICAP aplicado a logaritmo y el ajuste del modelo $SARIMA(0,1,1)(1,0,1)_{12}$

```
autoplot(log_serie_ent,series="log ICAP entrenamiento")+
  autolayer(sarima1$fitted,series = "SARIMA(0,1,1)(1,0,1)")+
  ggtitle("Log ICAP y Modelo Ajustado")+
  xlab("Año")+ylab("log ICAP")+
  scale_color_manual(values=c("#A4C4FC","#E80808"))+
  theme(legend.position = "bottom",legend.title = element_blank() )
```



Ahora gráficamos la serie original del ICAP y el inverso del logaritmo (exponencial) al modelo ajustado.

```
autoplot(serie_ent,series="ICAP entrenamiento")+
  autolayer(exp(sarima1$fitted),series = "exp[SARIMA(0,1,1)(1,0,1)]")+
  ggtitle("ICAP y Exponencial del Modelo Ajustado")+
  xlab("Año")+ylab("log ICAP")+
  scale_color_manual(values=c("#A4C4FC","#E80808"))+
  theme(legend.position = "bottom",legend.title = element_blank() )
```



7 Predicción

Se realizará la predicción para el conjunto test del ICAP.

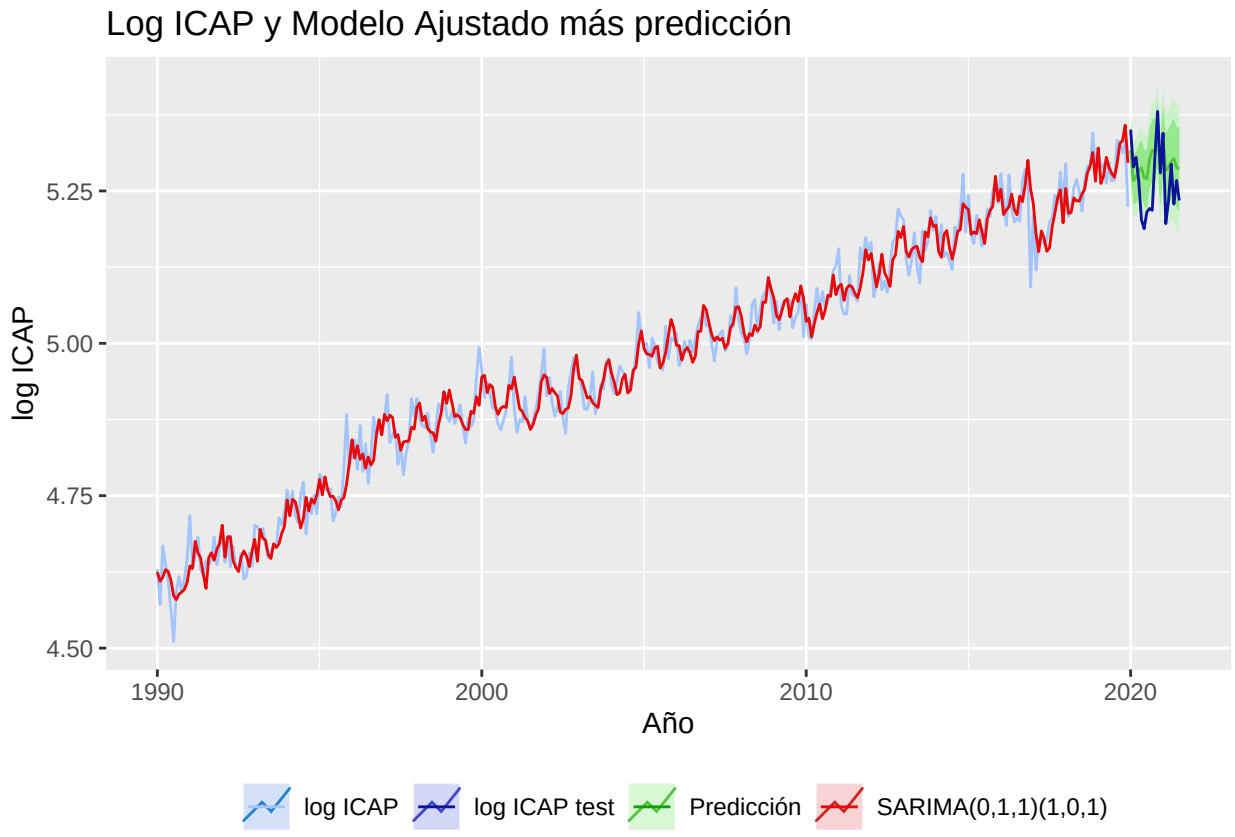
```
prediccion<-forecast(sarima1,19)
prediccion
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2020       5.316774 5.278336 5.355212 5.257988 5.375560
## Feb 2020       5.267360 5.227197 5.307523 5.205936 5.328784
## Mar 2020       5.272496 5.230679 5.314313 5.208543 5.336449
## Apr 2020       5.282364 5.238956 5.325771 5.215978 5.348750
## May 2020       5.287944 5.243002 5.332886 5.219211 5.356677
## Jun 2020       5.271778 5.225352 5.318204 5.200776 5.342780
## Jul 2020       5.269846 5.221982 5.317710 5.196645 5.343047
## Aug 2020       5.302260 5.253001 5.351520 5.226924 5.377596
## Sep 2020       5.316745 5.266128 5.367362 5.239333 5.394157
## Oct 2020       5.315744 5.263805 5.367683 5.236310 5.395178
## Nov 2020       5.342889 5.289661 5.396117 5.261483 5.424294
## Dec 2020       5.278540 5.224054 5.333027 5.195210 5.361870
## Jan 2021       5.330827 5.272537 5.389117 5.241680 5.419974
## Feb 2021       5.283934 5.223894 5.343974 5.192111 5.375757
## Mar 2021       5.288808 5.227068 5.350549 5.194384 5.383232
## Apr 2021       5.298173 5.234777 5.361568 5.201218 5.395128
```

```
## May 2021       5.303468 5.238460 5.368476 5.204047 5.402889
## Jun 2021       5.288127 5.221545 5.354709 5.186299 5.389955
## Jul 2021       5.286294 5.218174 5.354413 5.182114 5.390473
```

Primero gráficamos la serie del ICAP tanto de entrenamiento como de test aplicado a logaritmo y el ajuste del modelo $SARIMA(0,1,1)(1,0,1)_{12}$ más su predicción.

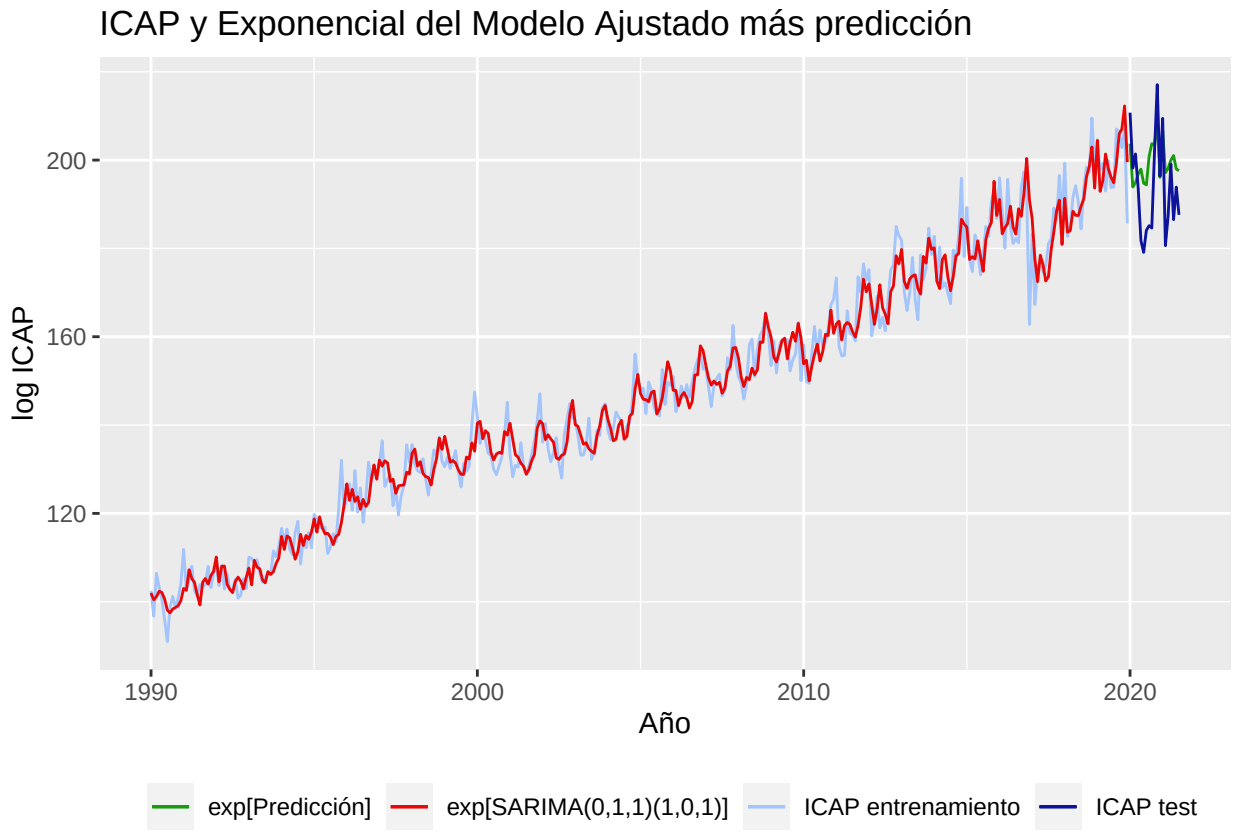
```
autoplot(log_serie_ent,series="log ICAP")+
  autolayer(sarima1$fitted,series = "SARIMA(0,1,1)(1,0,1)")+
  autolayer(prediccion,series = "Predicción")+
  autolayer(log(serie_test),series = "log ICAP test")+
  ggtitle("Log ICAP y Modelo Ajustado más predicción")+
  xlab("Año")+ylab("log ICAP")+
  scale_color_manual(values=c("#A4C4FC","#0D139B","#189B0D","#E80808"))+
  theme(legend.position = "bottom",legend.title = element_blank() )
```



Ahora gráficamos la serie original del ICAP tanto de entrenamiento como de test y el inverso del logaritmo (exponencial) al modelo ajustado más su predicción.

```
autoplot(serie_ent,series="ICAP entrenamiento")+
  autolayer(exp(sarima1$fitted),series = "exp[SARIMA(0,1,1)(1,0,1)]")+
  autolayer(exp(prediccion$mean),series = "exp[Predicción]")+
  autolayer(serie_test,series = "ICAP test")+
  ggtitle("ICAP y Exponencial del Modelo Ajustado más predicción")+
  xlab("Año")+ylab("log ICAP")+
```

```
  scale_color_manual(values=c("#189B0D","#E80808","#A4C4FC","#0D139B"))+
  theme(legend.position = "bottom",legend.title = element_blank() )
```



8 MAPE

Se calcula el MAPE para la predicción de los datos del conjunto del test.

```
mape<-function(y,f){
  pe<-((y-f)/y)
  mape1<-(sum(abs(pe))/length(y))*100
  return(mape1)
}
mape(log(serie_test),prediccion$mean)
```

```
## [1] 0.8683145
```

```
mape_test<-mape(serie_test,exp(prediccion$mean))
mape_test
```

```
## [1] 4.680337
```

El MAPE es igual a 4.6803368, esto aplicado a la exponencial de las predicciones.

9 Comparación de la predicción

El MAPE obtenido por el método de suavizamiento exponencial específicamente al modelo Holt-Winters aplicado a la serie de entrenamiento fue de 4.5956761 y el MAPE obtenido por el modelo $SARIMA(0,1,1)(1,0,1)_{12}$ de Box y Jenkins es 4.6803368 teniendo una diferencia de 0.084 a favor del modelo Holt-Winters.

Se podría concluir que la diferencia es mínima en cuanto al MAPE se refiere, por tal motivo se puede decir que ambos modelos son óptimos para poder predecir de manera eficiente y consistente la serie del ICAP de Bolivia.

10 Conclusiones

Se estudio la serie temporal del indice de consumo de agua potable en Bolivia desde el periodo de enero de 1990 hasta julio del 2021, teniendo en total 379 observaciones con periodicidad mensual, existiendo un valor atípico en diciembre del 2016. Posteriormente se pudo determinar la existencia de tendencia aditiva con estacionalidad, lo cual derivó a tener no estacionariedad en media y varianza, probando así la Función de Autocorrelación FAC y las dócimas de Dickey Fuller Aumentado.

Se transformó la serie con logaritmo para estabilizar varianza y posteriormente se hizo una diferencia de rezago 1 que era el único necesario teniendo una serie estacionaria, ya que con rezago 12 también se cumplia estacionariedad pero solo bastaba con la primera diferencia.

Se ajustó con la metodología de Box y Jenkins a un modelo $SARIMA(0,1,1)(1,0,1)_{12}$ que presentaba estimadores de parámetros significativos y el menor AIC de todos los modelos propuestos, se realizó la validación de supuestos teniendo así independencia y homocedasticidad en los residuos. En cuanto a la normalidad es necesario comentar que no cumple este supuesto al tener el dato atípico de diciembre de 2016, obviando este dato el supuesto de normalidad en los residuos se cumple.

Al pronosticar los datos con el modelo $SARIMA(0,1,1)(1,0,1)_{12}$ y obtener el MAPE del mismo, se comparó con el MAPE del modelo Holt Winters realizado en el anterior trabajo y se obtuvo una diferencia de 0.084, mostrando así una mínima diferencia entre ambos, lo cual deriva a decir que tanto el modelo de Box y Jenkins y el de suavizamiento exponencial son eficientes para realizar pronósticos a la serie del ICAP de Bolivia.