

Modelos de Espacio Estado

Trabajo 3. 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 Espacio Estado	4
5.1	Local Linear Trend Model	5
5.2	Local Level Model with Seasonal	8
5.3	Local Linear Trend Model with Seasonal	11
5.4	Decisión	15
5.5	Validación de supuestos	16
5.6	Conclusión ajuste del modelo	22
6	Gráfico serie original y el ajustado	23
7	Predicción	24
8	MAPE	26
9	Comparación de la predicción	27
10	Conclusiones	27

1 Introducción

Para este trabajo de análisis de series de tiempo se aplicarán técnicas de modelamiento en series temporales, específicamente los modelos de Espacio Estado.

La serie de tiempo de estudio para este trabajo es:

- Índice 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/modelo_espacio_estado).

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

- https://bladimir-morales.github.io/modelo_espacio_estado/trabajo3.pdf

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

- https://bladimir-morales.github.io/modelo_espacio_estado/trabajo3.html

2 Índice de consumo de agua potable ICAP

2.1 Datos

El índice 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 índice 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 índice 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).

¹www.ine.gob.bo

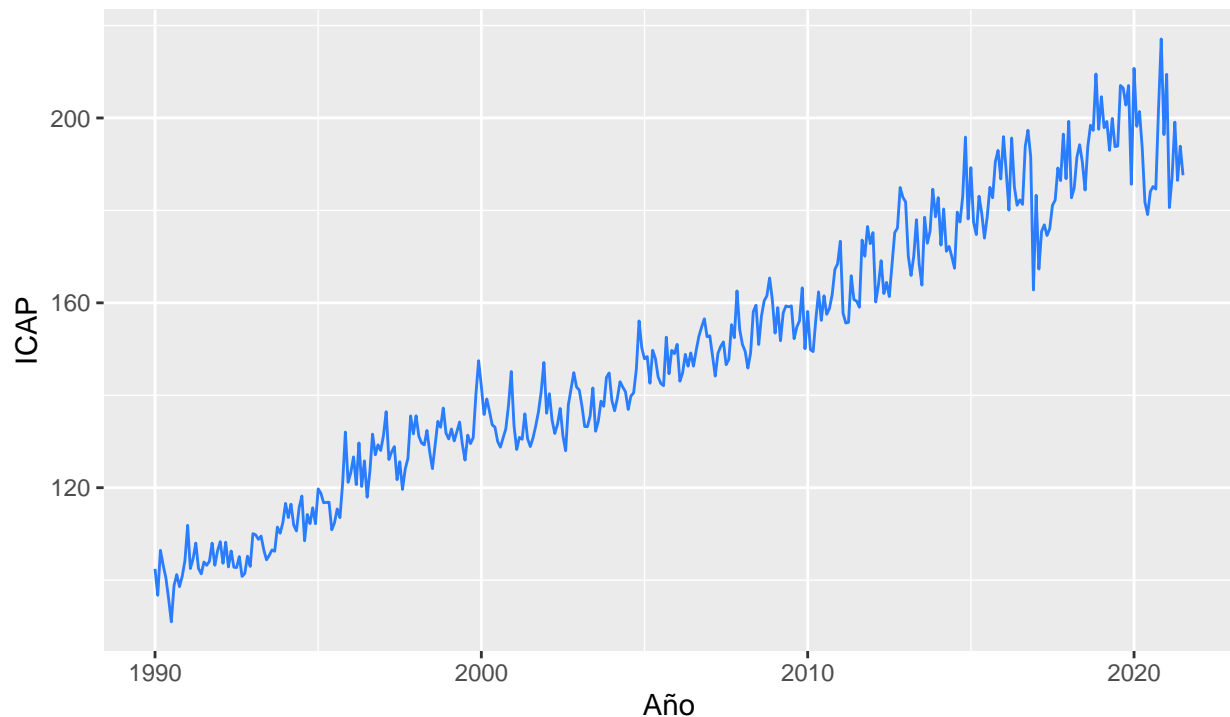
3 Gráfico

```
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")
```

Indice mensual de consumo de agua potable en Bolivia: enero 1990 a julio

(año base 1990=100)



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.

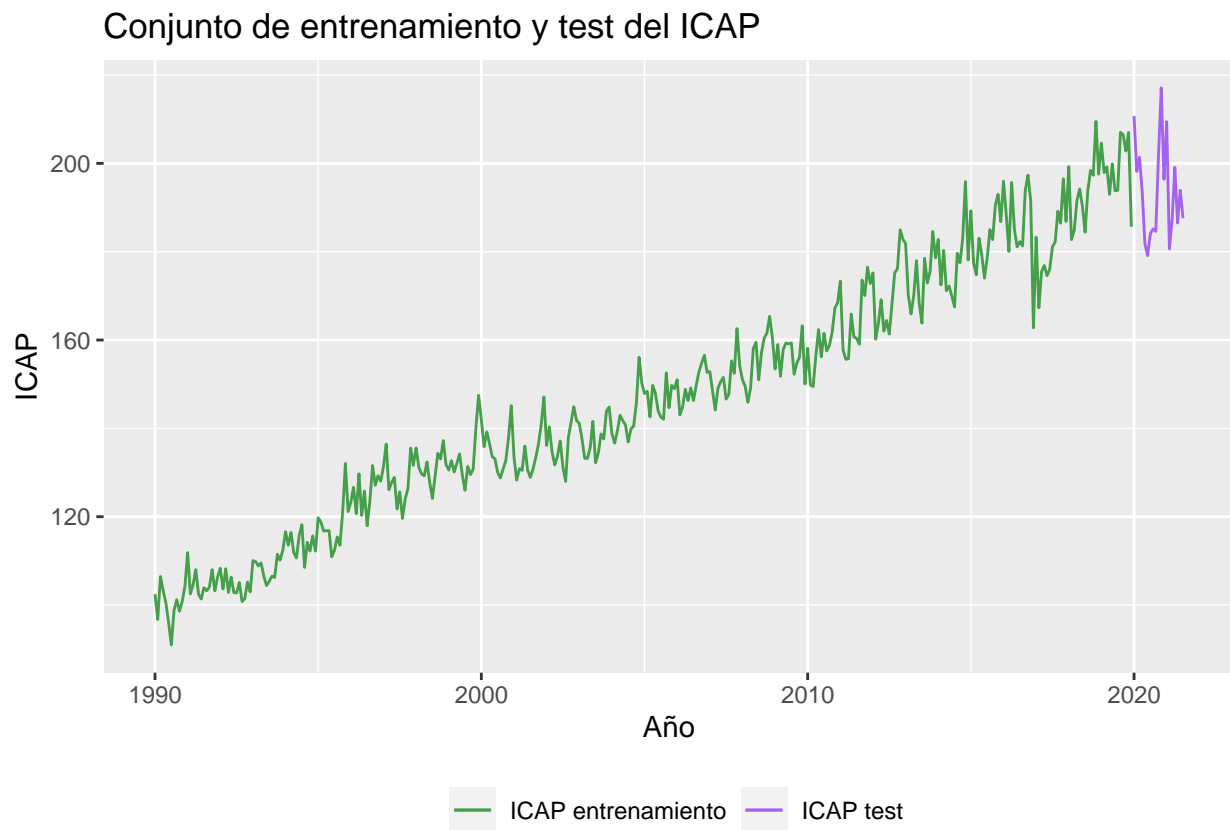
- Conjunto de datos de test.

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 Espacio Estado

En busca del mejor modelo para la serie de tiempo se estimará los siguientes tres modelos:

- Local Linear Trend Model
- Local Level Model with Seasonal

- Local Linear Trend Model with Seasonal

Se utilizará en base a los datos de la serie de entrenamiento aplicando logaritmo y posteriormente se verifica que sea un modelo de espacio estado.

Mencionar que el criterio de selección será el Criterio de Información de Akaike AIC, denotado como:

$$AIC = \frac{1}{n}[-2n(\log L_d) + 2(q + w)]$$

donde n es el número de observaciones en la serie temporal, $\log L_d$ es el valor de la función de verosimilitud logarítmica difusa que se maximiza en el estado, q es el número de valores iniciales difusos en el estado, y w es el número total de varianzas de perturbación estimadas en el análisis. Cuando se comparan diferentes modelos con el AIC se cumple la siguiente regla los valores más pequeños denotan modelos que se ajustan mejor que los más grandes.

5.1 Local Linear Trend Model

5.1.1 Ajuste del modelo

```
log_serie_ent<-log(serie_ent)
mod1 <- SSMModel(log_serie_ent ~SSMtrend(degree=2, Q=list(NA,NA),a1=c(0,0)),H=NA)
is.SSMModel(mod1)
```

```
## [1] TRUE
```

```
mod1$H
```

```
## , , 1
##
##      [,1]
## [1,]    NA
```

```
mod1$Q
```

```
## , , 1
##
##      [,1] [,2]
## [1,]    NA    0
## [2,]    0    NA
```

```
modle<-fitSSM(mod1,init=c(log(var(serie_ent)),log(var(serie_ent)),log(var(serie_ent))), method="BFGS")
modle
```

```
## $optim.out
## $optim.out$par
## [1] -8.255980 -23.290728 -7.203253
##
## $optim.out$value
## [1] -673.2368
```

```
##
## $optim.out$counts
## function gradient
##      81      41
##
## $optim.out$convergence
## [1] 0
##
## $optim.out$message
## NULL
##
##
## $model
## Call:
## SSMModel(formula = log_serie_ent ~ SSMtrend(degree = 2, Q = list(NA,
##      NA), a1 = c(0, 0)), H = NA)
##
## State space model object of class SSMModel
##
## Dimensions:
## [1] Number of time points: 360
## [1] Number of time series: 1
## [1] Number of disturbances: 2
## [1] Number of states: 2
## Names of the states:
## [1] level slope
## Distributions of the time series:
## [1] gaussian
##
## Object is a valid object of class SSMModel.
```

La varianza de ε es:

```
mod1e$model["H"]
```

```
## , , 1
##
##      [,1]
## [1,] 0.0007441613
```

La varianza de η es:

```
mod1e$model["Q"]
```

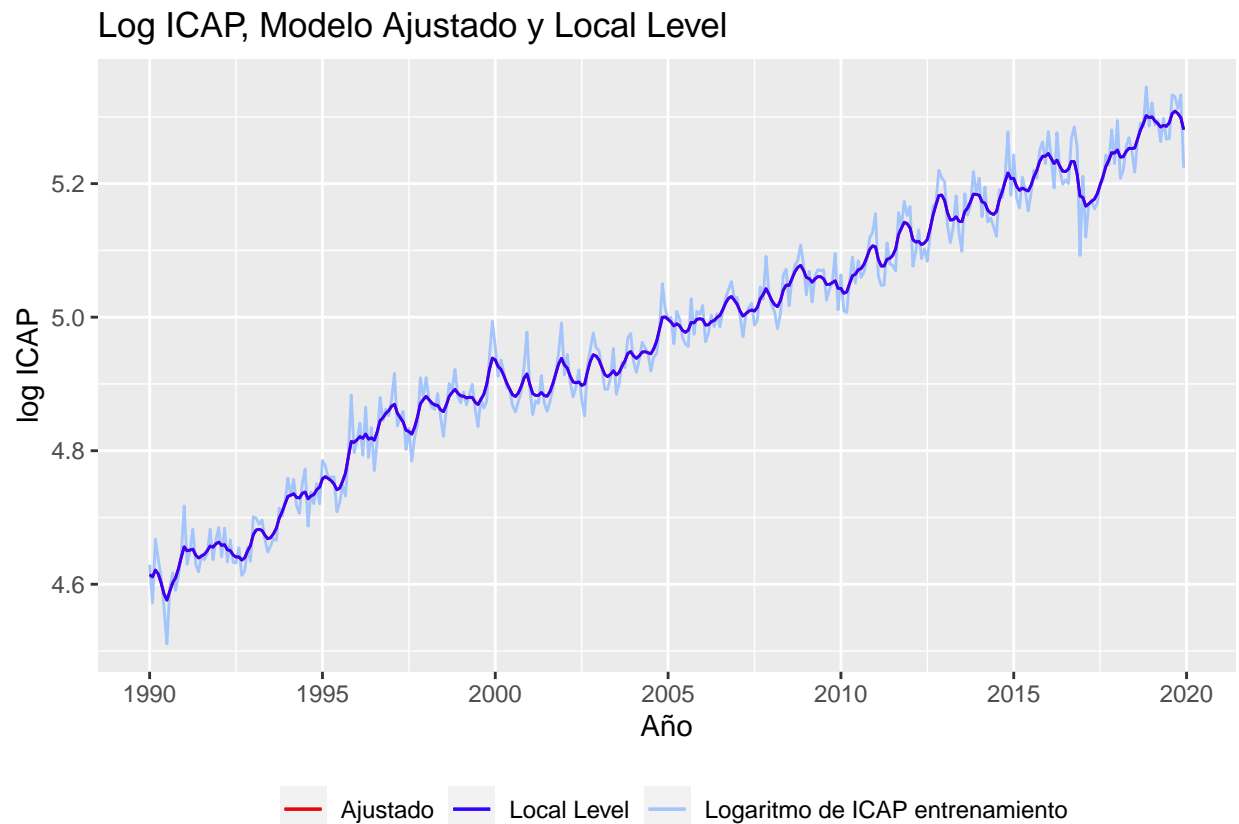
```
## , , 1
##
##      [,1]      [,2]
## [1,] 0.0002597008 0.000000e+00
## [2,] 0.00000000000 7.673006e-11
```

Se estima los parámetros del modelo.

```
out1 <- KFS(mod1$model, filtering = "state", smoothing = "state")
att1 <- out1$att#filtraje
alphahat1 <- out1$alphahat#suavizado
```

5.1.2 Gráfico de la serie estimada

```
autoplot(log_serie_ent,series="Logaritmo de ICAP entrenamiento")+
  autolayer(fitted(mod1$model),series = "Ajustado")+
  autolayer(alphahat1[,1],series = "Local Level")+
  ggtitle("Log ICAP, Modelo Ajustado y Local Level")+
  xlab("Año")+ylab("log ICAP")+
  scale_color_manual(values=c("#E80808","#3301FF","#A4C4FC"))+
  theme(legend.position = "bottom",legend.title = element_blank() )
```



5.1.3 AIC

```
aic_mod1<-(1/length(log_serie_ent))*((-2*length(log_serie_ent)*out1$logLik)+(2*(3+2)))
aic_mod1
```

```
## [1] -1346.446
```

5.2 Local Level Model with Seasonal

5.2.1 Ajuste del modelo

```
mod2 <- SSMModel(log_serie_ent ~ SSMtrend(1, Q = list(NA))+  
                  SSMseasonal(period = 12, sea.type = "trigonometric", Q = NA), H = NA)  
is.SSMModel(mod2)
```

```
## [1] TRUE
```

```
# varianza de Epsilon  
mod2["H"]
```

```
## , , 1  
##  
##      [,1]  
## [1,]   NA
```

```
#varianza de eta  
mod2["Q"]
```

```
## , , 1  
##  
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]  
## [1,]   NA    0    0    0    0    0    0    0    0    0    0    0  
## [2,]    0   NA    0    0    0    0    0    0    0    0    0    0  
## [3,]    0    0   NA    0    0    0    0    0    0    0    0    0  
## [4,]    0    0    0   NA    0    0    0    0    0    0    0    0  
## [5,]    0    0    0    0   NA    0    0    0    0    0    0    0  
## [6,]    0    0    0    0    0   NA    0    0    0    0    0    0  
## [7,]    0    0    0    0    0    0   NA    0    0    0    0    0  
## [8,]    0    0    0    0    0    0    0   NA    0    0    0    0  
## [9,]    0    0    0    0    0    0    0    0   NA    0    0    0  
## [10,]   0    0    0    0    0    0    0    0    0   NA    0    0  
## [11,]   0    0    0    0    0    0    0    0    0    0   NA    0  
## [12,]   0    0    0    0    0    0    0    0    0    0    0   NA
```

```
ownupdatefn <- function(pars, model){  
  model$H[] <- exp(pars[1])  
  diag(model$Q[, , 1]) <- exp(c(pars[2], rep(pars[3], 11)))  
  model  
}
```

```
mod2e <- fitSSM(mod2, log(c(var(serie_ent), var(serie_ent), var(serie_ent))), ownupdatefn, method = "BFGS")  
mod2e
```

```
## $optim.out  
## $optim.out$par  
## [1] -7.413520 -8.876086 -156.474196  
##
```



```
## $optim.out$value
## [1] -682.5464
##
## $optim.out$counts
## function gradient
##      31      12
##
## $optim.out$convergence
## [1] 0
##
## $optim.out$message
## NULL
##
##
## $model
## Call:
## SSMModel(formula = log_serie_ent ~ SSMtrend(1, Q = list(NA)) +
##      SSMseasonal(period = 12, sea.type = "trigonometric", Q = NA),
##      H = NA)
##
## State space model object of class SSMModel
##
## Dimensions:
## [1] Number of time points: 360
## [1] Number of time series: 1
## [1] Number of disturbances: 12
## [1] Number of states: 12
## Names of the states:
## [1] level      sea_trig1    sea_trig*1  sea_trig2   sea_trig*2  sea_trig3
## [7] sea_trig*3  sea_trig4   sea_trig*4  sea_trig5   sea_trig*5  sea_trig6
## Distributions of the time series:
## [1] gaussian
##
## Object is a valid object of class SSMModel.
```

La varianza de ε es:

```
mod2e$model["H"]
```

```
## , , 1
##
##      [,1]
## [1,] 0.0006030439
```

La varianza de η es:

```
mod2e$model$Q
```

```
## , , 1
##
##      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.0001396899 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [2,] 0.00000000000 1.10693e-68 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
```

```
## [3,] 0.0000000000 0.00000e+00 1.10693e-68 0.00000e+00 0.00000e+00 0.00000e+00
## [4,] 0.0000000000 0.00000e+00 0.00000e+00 1.10693e-68 0.00000e+00 0.00000e+00
## [5,] 0.0000000000 0.00000e+00 0.00000e+00 0.00000e+00 1.10693e-68 0.00000e+00
## [6,] 0.0000000000 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 1.10693e-68
## [7,] 0.0000000000 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [8,] 0.0000000000 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [9,] 0.0000000000 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [10,] 0.0000000000 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [11,] 0.0000000000 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [12,] 0.0000000000 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
##      [,7]      [,8]      [,9]      [,10]      [,11]      [,12]
## [1,] 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [2,] 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [3,] 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [4,] 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [5,] 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [6,] 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [7,] 1.10693e-68 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [8,] 0.00000e+00 1.10693e-68 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
## [9,] 0.00000e+00 0.00000e+00 1.10693e-68 0.00000e+00 0.00000e+00 0.00000e+00
## [10,] 0.00000e+00 0.00000e+00 0.00000e+00 1.10693e-68 0.00000e+00 0.00000e+00
## [11,] 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 1.10693e-68 0.00000e+00
## [12,] 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 1.10693e-68
```

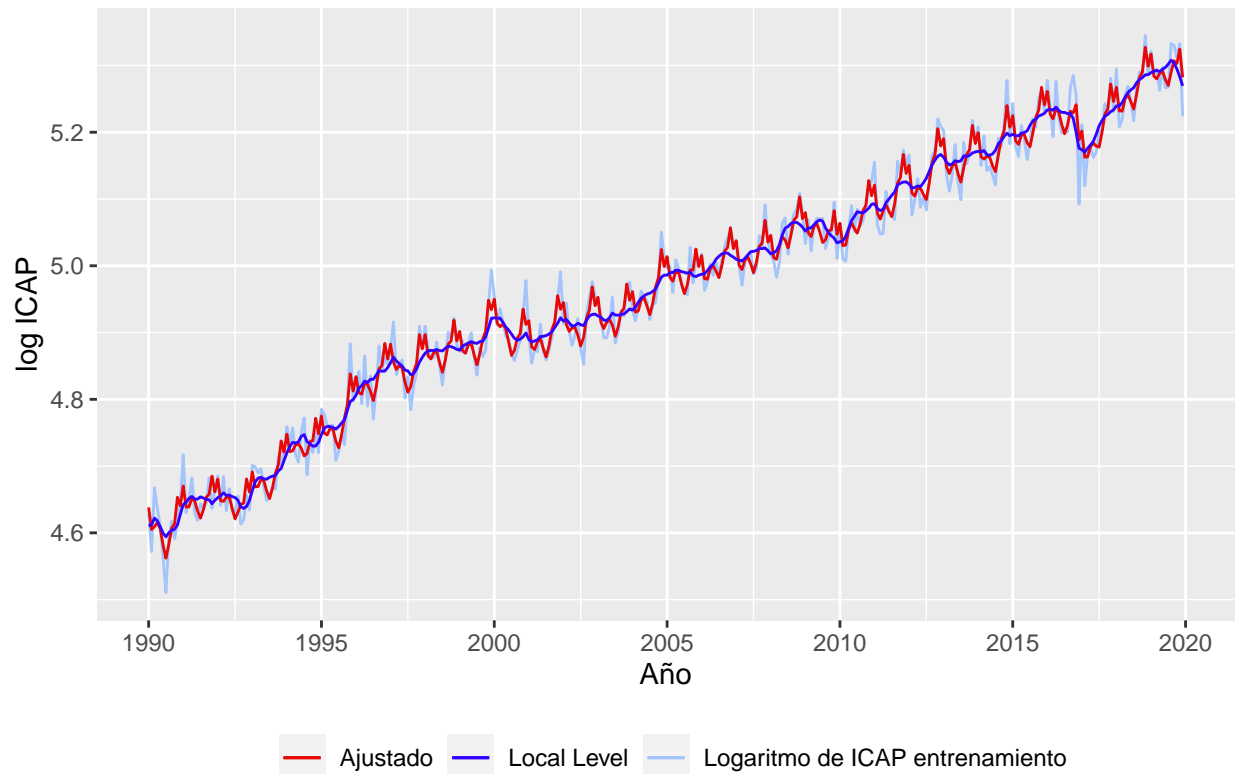
Se estima los parámetros del modelo.

```
out2 <- KFS(mod2e$model, filtering = "state", smoothing = "state")
att2 <- out2$att#filtraje
alphahat2 <- out2$alphahat#suavizado
```

5.2.2 Gráfico de la serie estimada

```
autoplot(log_serie_ent,series="Logaritmo de ICAP entrenamiento")+
  autolayer(fitted(mod2e$model),series = "Ajustado")+
  autolayer(alphahat2[,1],series = "Local Level")+
  ggtitle("Log ICAP, Modelo Ajustado y Local Level")+
  xlab("Año")+ylab("log ICAP")+
  scale_color_manual(values=c("#E80808","#3301FF","#A4C4FC"))+
  theme(legend.position = "bottom",legend.title = element_blank() )
```

Log ICAP, Modelo Ajustado y Local Level



5.2.3 AIC

```
aic_mod2 <- (1/length(log_serie_ent)) * ( -2*length(log_serie_ent)*out2$logLik) + (2*(3+12))
aic_mod2
```

```
## [1] -1365.009
```

5.3 Local Linear Trend Model with Seasonal

5.3.1 Ajuste del modelo

```
mod3 <- SSMModel(log_serie_ent ~ SSMtrend(2, Q = list(NA, NA)) +
  SSMseasonal(period = 12, sea.type = "trigonometric", Q = NA), H = NA)

is.SSMModel(mod3)
```

```
## [1] TRUE
```

```
# varianza de Epsilon
mod3["H"]
```

```
## , , 1
##
##      [,1]
## [1,]    NA
```

```
#varianza de eta
mod3["Q"]
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    NA     0     0     0     0     0     0     0     0     0     0     0     0
## [2,]     0    NA     0     0     0     0     0     0     0     0     0     0     0
## [3,]     0     0    NA     0     0     0     0     0     0     0     0     0     0
## [4,]     0     0     0    NA     0     0     0     0     0     0     0     0     0
## [5,]     0     0     0     0    NA     0     0     0     0     0     0     0     0
## [6,]     0     0     0     0     0    NA     0     0     0     0     0     0     0
## [7,]     0     0     0     0     0     0    NA     0     0     0     0     0     0
## [8,]     0     0     0     0     0     0     0    NA     0     0     0     0     0
## [9,]     0     0     0     0     0     0     0     0    NA     0     0     0     0
## [10,]    0     0     0     0     0     0     0     0     0    NA     0     0     0
## [11,]    0     0     0     0     0     0     0     0     0     0    NA     0     0
## [12,]    0     0     0     0     0     0     0     0     0     0     0    NA     0
## [13,]    0     0     0     0     0     0     0     0     0     0     0     0    NA
```

```
ownupdatefn <- function(pars, model){
  model$H[] <- exp(pars[1])
  diag(model$Q[, , 1]) <- exp(c(pars[2],pars[3], rep(pars[4], 11)))
  model
}

mod3e <- fitSSM(mod3,log(c(var(serie_ent),var(serie_ent), var(serie_ent),var(serie_ent))),
  ownupdatefn, method = "BFGS")
mod3e
```

```
## $optim.out
## $optim.out$par
## [1] -7.312585 -9.420756 -25.520802 -149.527076
##
## $optim.out$value
## [1] -681.516
##
## $optim.out$counts
## function gradient
##      55      31
##
## $optim.out$convergence
## [1] 0
##
## $optim.out$message
## NULL
##
##
```

```
## $model
## Call:
## SSMModel(formula = log_serie_ent ~ SSMtrend(2, Q = list(NA, NA)) +
##       SSMseasonal(period = 12, sea.type = "trigonometric", Q = NA),
##       H = NA)
##
## State space model object of class SSMModel
##
## Dimensions:
## [1] Number of time points: 360
## [1] Number of time series: 1
## [1] Number of disturbances: 13
## [1] Number of states: 13
## Names of the states:
## [1] level      slope      sea_trig1  sea_trig*1  sea_trig2  sea_trig*2
## [7] sea_trig3  sea_trig*3  sea_trig4  sea_trig*4  sea_trig5  sea_trig*5
## [13] sea_trig6
## Distributions of the time series:
## [1] gaussian
##
## Object is a valid object of class SSMModel.
```

La varianza de ε es:

```
mod3e$model["H"]
```

```
## , , 1
##
##           [,1]
## [1,] 0.0006670905
```

La varianza de η es:

```
mod3e$model$Q
```

```
## , , 1
##
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 8.102471e-05 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [2,] 0.000000e+00 8.250049e-12 0.000000e+00 0.000000e+00 0.000000e+00
## [3,] 0.000000e+00 0.000000e+00 1.151373e-65 0.000000e+00 0.000000e+00
## [4,] 0.000000e+00 0.000000e+00 0.000000e+00 1.151373e-65 0.000000e+00
## [5,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.151373e-65
## [6,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [7,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [8,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [9,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [10,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [11,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [12,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [13,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##           [,6]      [,7]      [,8]      [,9]      [,10]
## [1,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
```

```
## [2,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [3,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [4,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [5,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [6,] 1.151373e-65 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [7,] 0.000000e+00 1.151373e-65 0.000000e+00 0.000000e+00 0.000000e+00
## [8,] 0.000000e+00 0.000000e+00 1.151373e-65 0.000000e+00 0.000000e+00
## [9,] 0.000000e+00 0.000000e+00 0.000000e+00 1.151373e-65 0.000000e+00
## [10,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.151373e-65
## [11,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [12,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [13,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##      [,11]      [,12]      [,13]
## [1,] 0.000000e+00 0.000000e+00 0.000000e+00
## [2,] 0.000000e+00 0.000000e+00 0.000000e+00
## [3,] 0.000000e+00 0.000000e+00 0.000000e+00
## [4,] 0.000000e+00 0.000000e+00 0.000000e+00
## [5,] 0.000000e+00 0.000000e+00 0.000000e+00
## [6,] 0.000000e+00 0.000000e+00 0.000000e+00
## [7,] 0.000000e+00 0.000000e+00 0.000000e+00
## [8,] 0.000000e+00 0.000000e+00 0.000000e+00
## [9,] 0.000000e+00 0.000000e+00 0.000000e+00
## [10,] 0.000000e+00 0.000000e+00 0.000000e+00
## [11,] 1.151373e-65 0.000000e+00 0.000000e+00
## [12,] 0.000000e+00 1.151373e-65 0.000000e+00
## [13,] 0.000000e+00 0.000000e+00 1.151373e-65
```

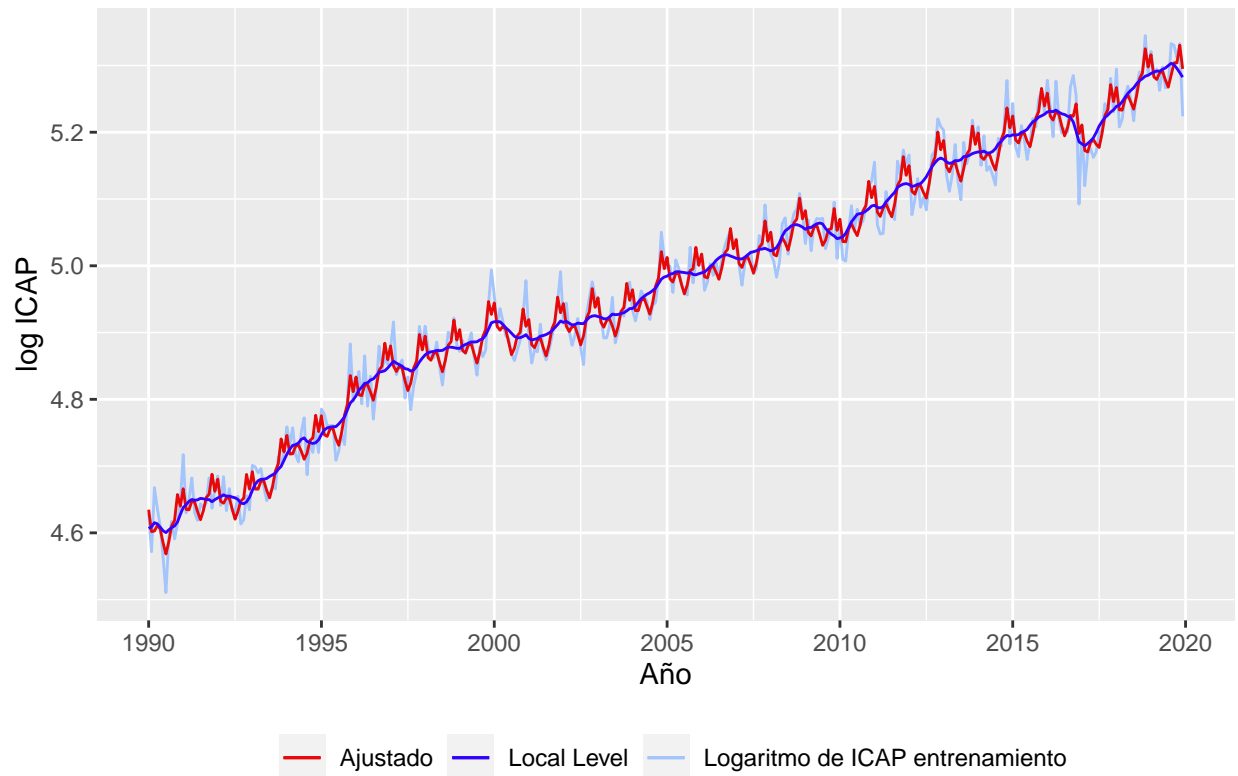
Se estima los parámetros del modelo.

```
out3 <- KFS(mod3e$model, filtering = "state", smoothing = "state")
att3 <- out3$att
alphahat3 <- out3$alphahat
```

5.3.2 Gráfico de la serie estimada

```
autoplot(log_serie_ent, series="Logaritmo de ICAP entrenamiento")+
  autolayer(fitted(mod3e$model), series = "Ajustado")+
  autolayer(alphahat3[,1], series = "Local Level")+
  ggtitle("Log ICAP, Modelo Ajustado y Local Level")+
  xlab("Año")+ylab("log ICAP")+
  scale_color_manual(values=c("#E80808", "#3301FF", "#A4C4FC"))+
  theme(legend.position = "bottom", legend.title = element_blank() )
```

Log ICAP, Modelo Ajustado y Local Level



5.3.3 AIC

```
aic_mod3<-(1/length(log_serie_ent))*((-2*length(log_serie_ent)*out3$logLik)+(2*(4+13)))
aic_mod3
```

```
## [1] -1362.937
```

5.4 Decisión

Si se compara los tres modelos de espacio estado en sus valores de AIC se tiene:

- Local Linear Trend Model $AIC = -1346.4458123$
- Local Level Model with Seasonal $AIC = -1365.0094917$
- Local Linear Trend Model with Seasonal $AIC = -1362.9374857$

Se puede ver que los menores valores de AIC son del modelo *Local Level Model with Seasonal* y *Local Linear Trend Model with Seasonal* teniendo una diferencia de 2.066 puntos.

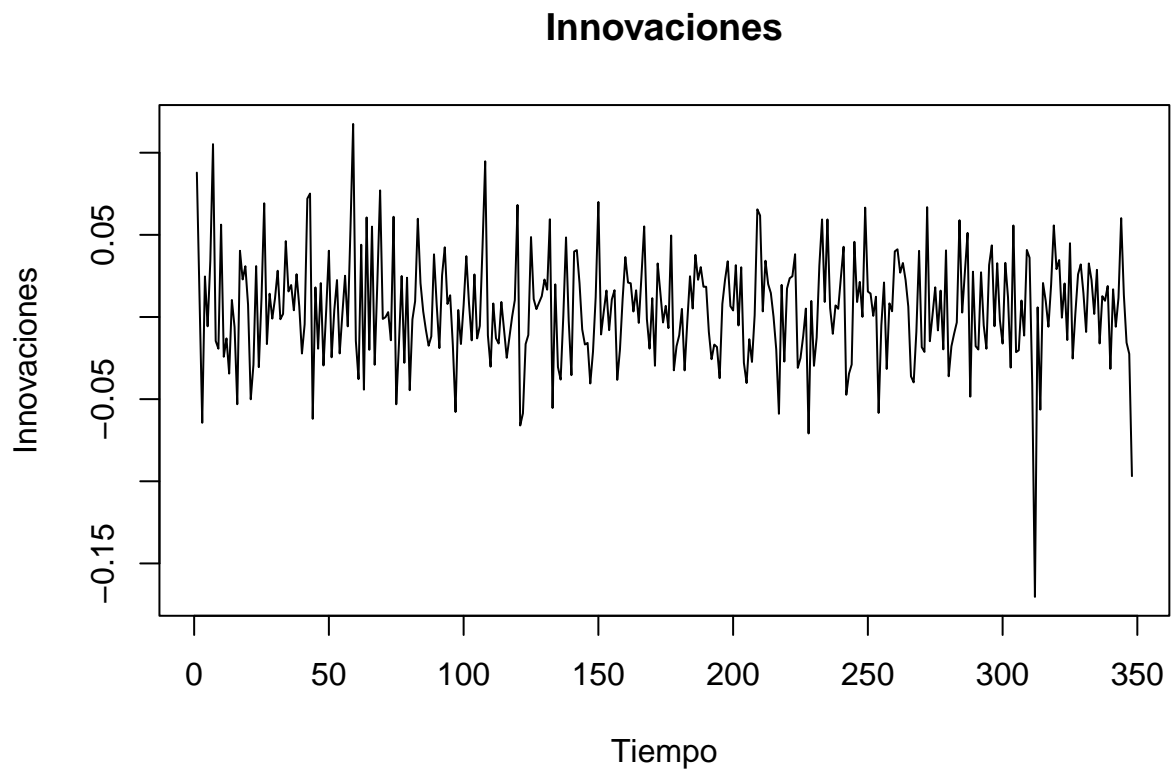
Por tal motivo se elegirá el modelo *Local Level model with Seasonal* ya que tiene el menor AIC y al estimar menos parámetros es más parsimonioso.

5.5 Validación de supuestos

5.5.1 Análisis de residuos

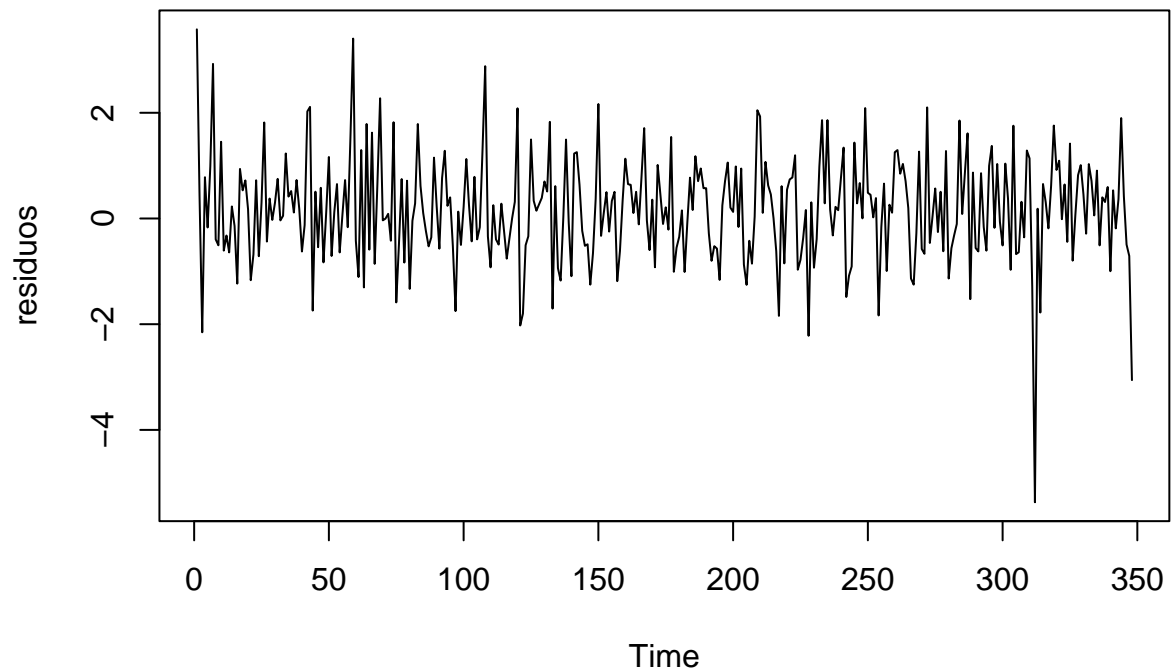
Se gráfica las innovaciones.

```
innovaciones <- out2$v  
innovaciones <- innovaciones[13:length(innovaciones)]  
plot.ts(innovaciones ,main="Innovaciones",ylab="Innovaciones",xlab="Tiempo")
```



Se gráfica los residuos.

```
F1 <- as.numeric(out2$F[1:length(innovaciones)])  
residuos <- innovaciones/sqrt(F1)  
plot.ts(residuos)
```

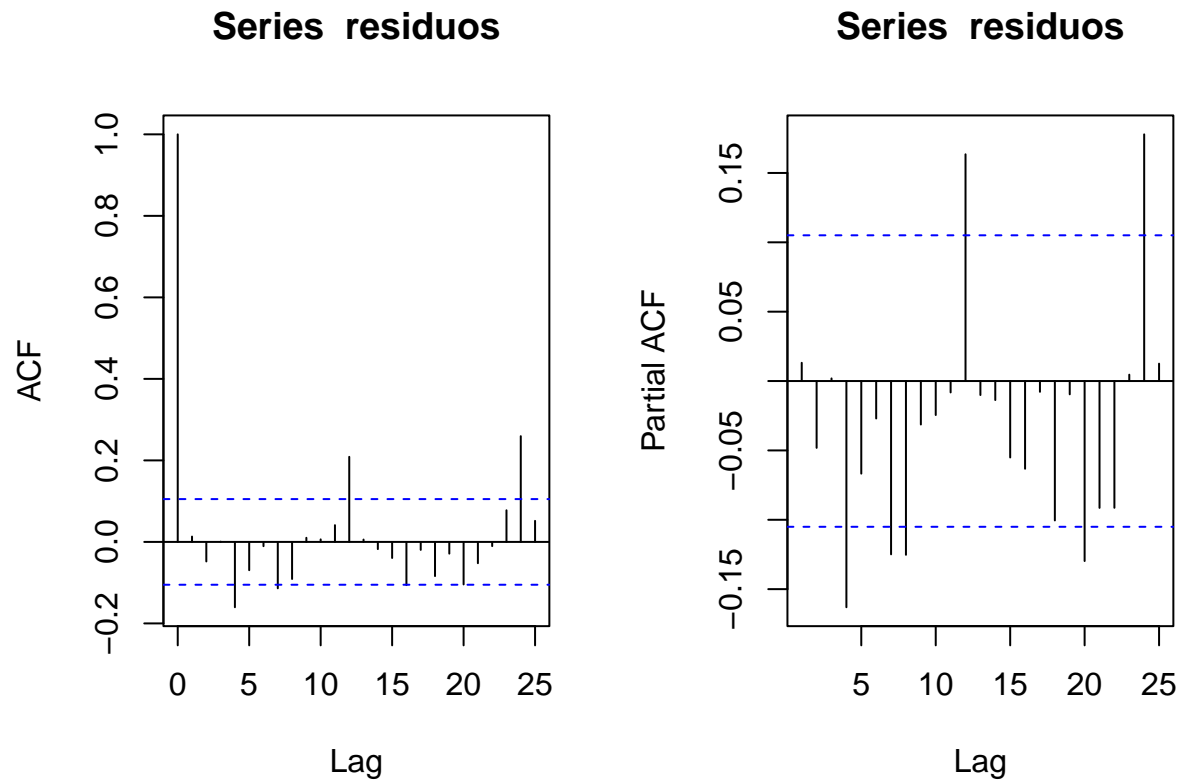



5.5.2 Independencia de Residuos

- **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 no todos los coeficientes estimados de la FAC y FACP de los residuos son aproximadamente cero, no teniendo un posible 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:

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

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

```
##
## Box-Pierce test
##
## data:  residuos
## X-squared = 0.060173, df = 1, p-value = 0.8062
```

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

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

```
##
## Box-Ljung test
##
## data:  residuos
## X-squared = 0.060693, df = 1, p-value = 0.8054
```

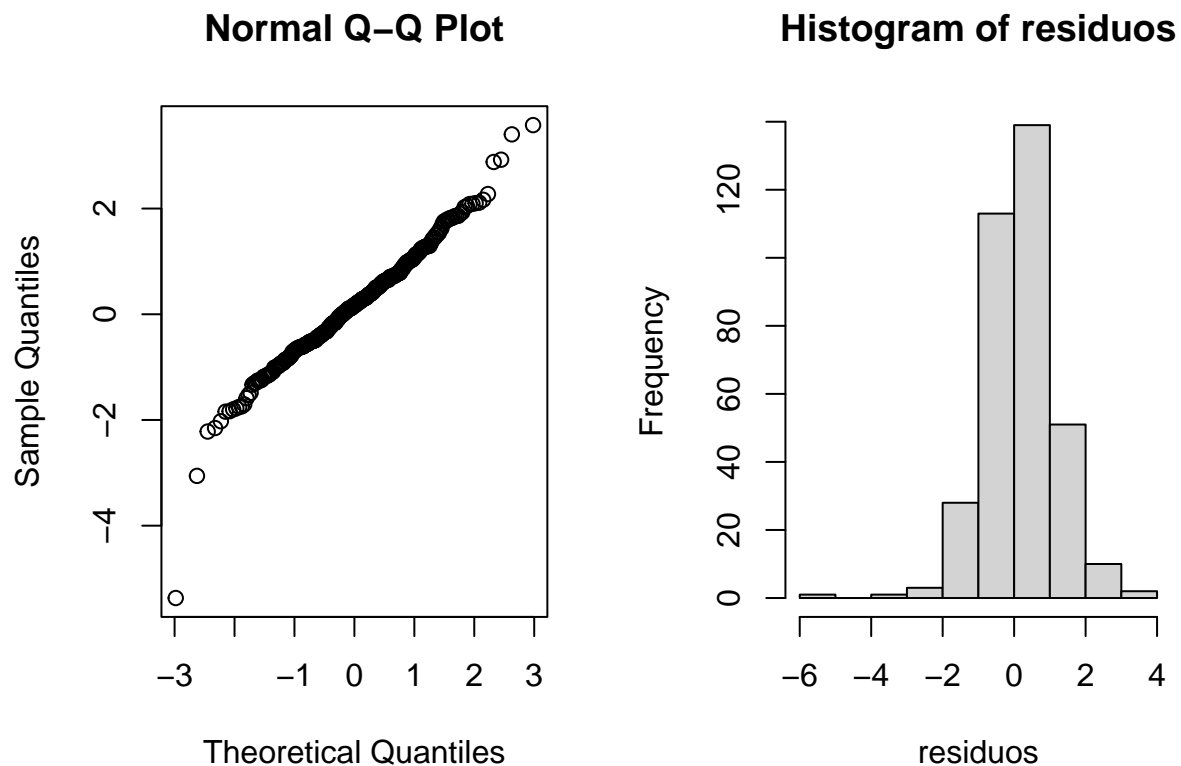
Para la prueba de Ljung-Box no se rechaza H_0 teniendo así independencia en los residuos.

Conclusión: Existe evidencia estadística para decir que los residuos son independientes.

5.5.3 Normalidad de Residuos

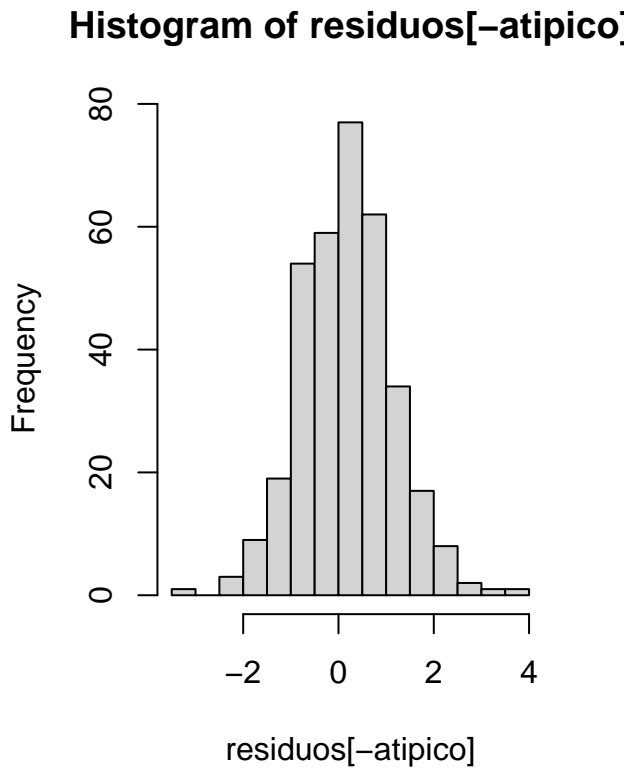
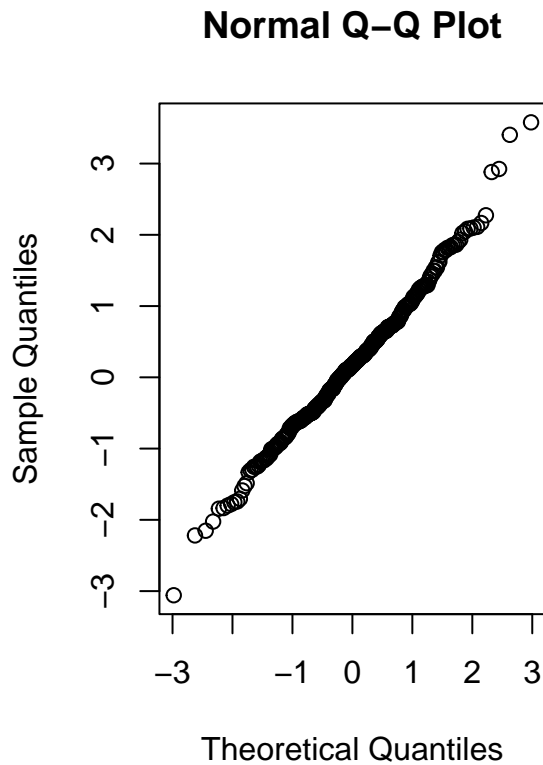
Se verifica en los residuos QQ-normal y el histograma.

```
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 d́cimas de normalidad con este valor atípico y sin ́l.

Se realizará las pruebas de normalidad.

- **D́cima de Jarque-Bera**

La d́cima de Jarque-Bera tiene la siguiente hiṕtesis nula

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

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

```
##
##  Jarque-Bera test for normality
##
## data:  residuos
## JB = 111.15, p-value < 2.2e-16
```

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

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

```
##
## Jarque-Bera test for normality
##
## data:  residuos[-atipico]
## JB = 7.8144, p-value = 0.016
```

Sin contar con este dato se puede observar que aún se rechaza la hipótesis nula, no teniendo normalidad en los residuos, pero el p-valor esta mas cercano a 0.05

- **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 = 0.97534, p-value = 1.148e-05
```

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 = 0.99331, p-value = 0.1261
```

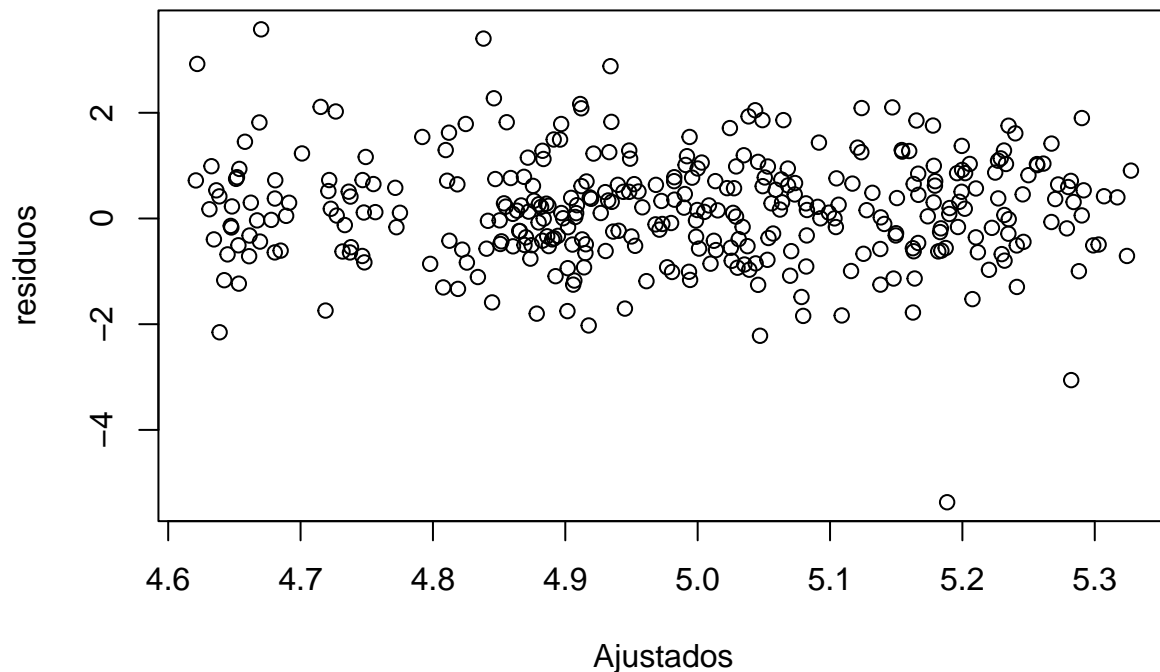
No se rechaza la hipótesis nula donde se concluye que los residuos cumplen normalidad.

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 la dócima de Jarque-Bera aún determina no normalidad mientras que la dócima de Shapiro Wilk determina que los residuos son normales, cumpliendo así este supuesto.

5.5.4 Homocedasticidad de Residuos

Se hará primero un análisis gráfico.

```
ajustados<-fitted(mod2e$model)
plot(ajustados[13:length(ajustados)],residuos,xlab="Ajustados")
```



Se puede evidenciar que existe un valor atípico el cual corresponde a diciembre del 2016, pero según el gráfico existiría homocedasticidad.

Ahora se calculará el estadístico $H(h)$, donde h es el entero más próximo a $(n - q)/3$ y q es el número de inicializaciones difusas y se compara con una $F(h, h)$.

```
H <- function(innovations){
  n <- length(innovations)
  d <- 1
  h <- round((n-d)/3)
  H <- sum(innovations[(n-h+1):n]^2)/sum(innovations[(d+1):(d+h)]^2)
  cat("El valor del estadístico es H=", H, "y el valor crítico es", qf(0.95,h,h))
}
H(residuos)
```

```
## El valor del estadístico es H= 1.164754 y el valor crítico es 1.358929
```

Como el valor H es menor al valor crítico se dice que hay homocedasticidad.

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

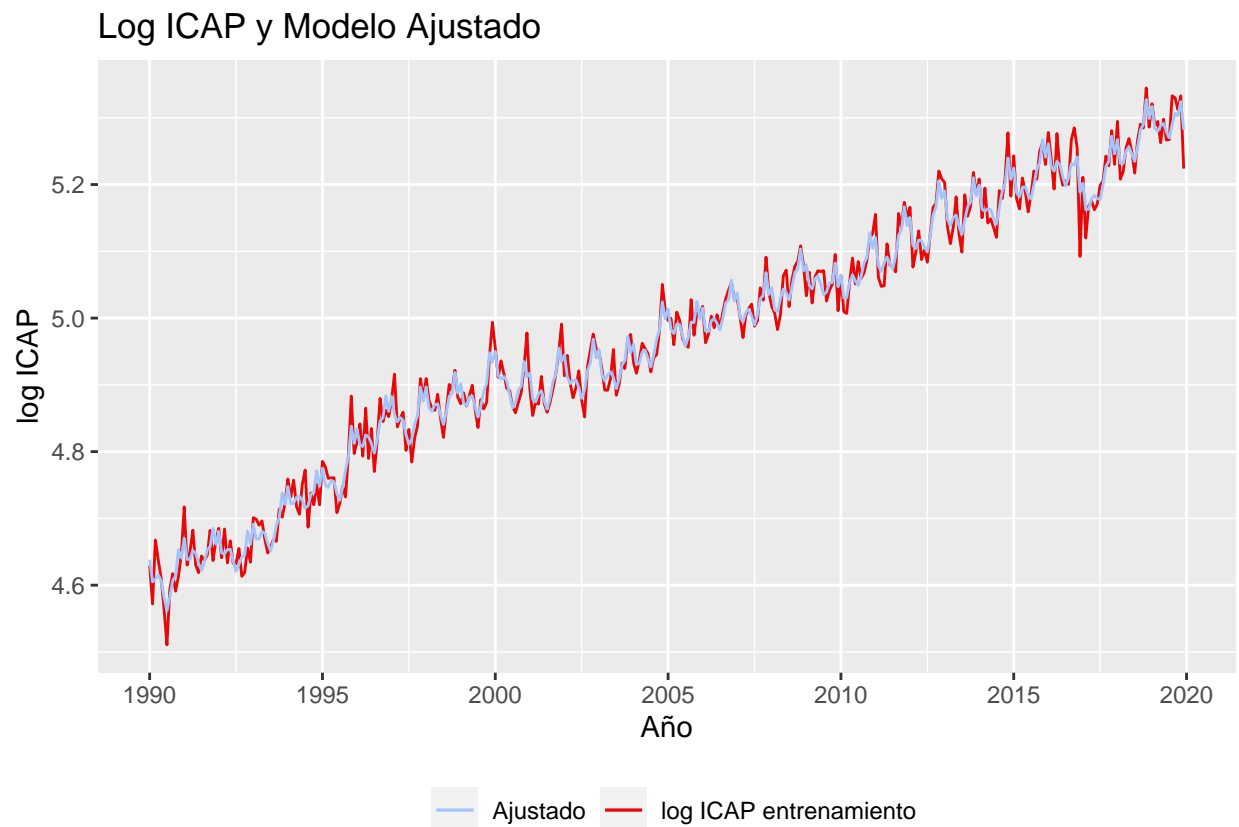
5.6 Conclusión ajuste del modelo

Se puede concluir entonces que el modelo elegido *Local Level Model with Seasonal* tiene el menor AIC y cumple los supuestos de independencia y homocedasticidad, en cuanto a la normalidad sin contar con el valor atípico del 2016 se puede asumir este supuesto con la prueba de Shapiro-Wilk.

6 Gráfico serie original y el ajustado

Primero graficamos la serie del ICAP aplicado a logaritmo y el ajuste del modelo *Local Level Model with Seasonal*

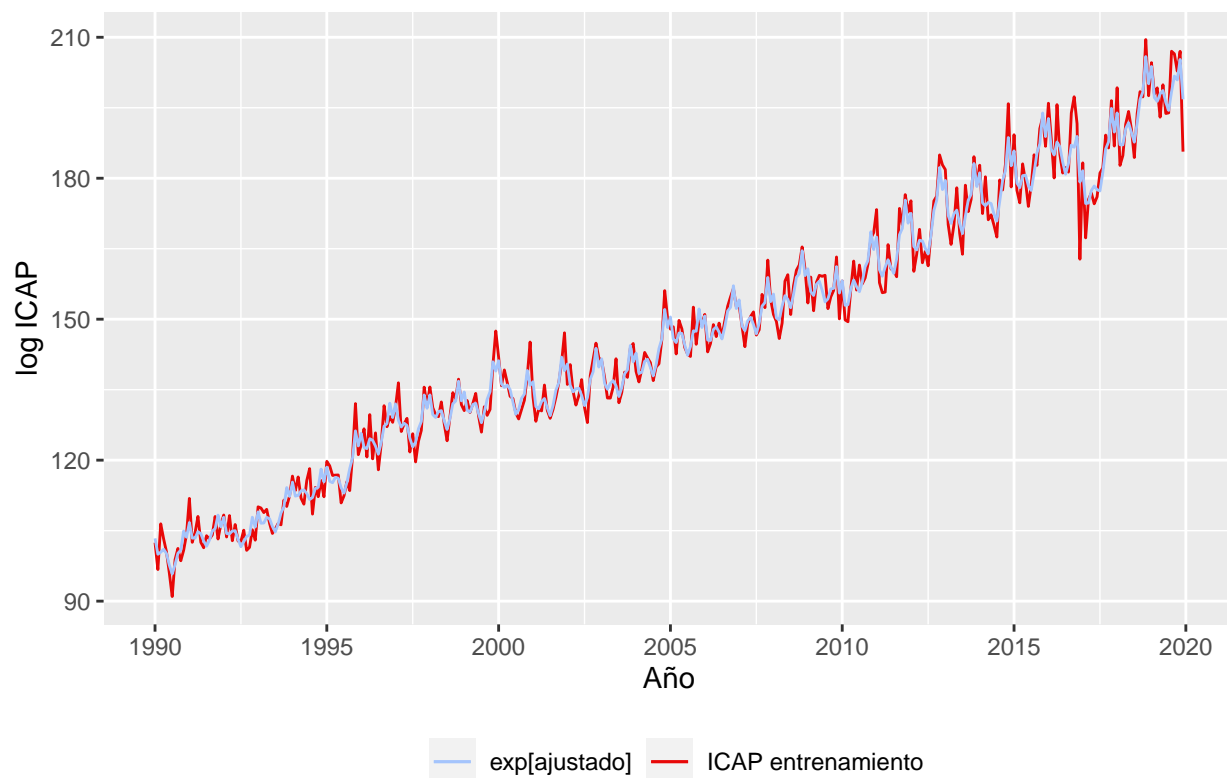
```
autoplot(log_serie_ent,series="log ICAP entrenamiento")+  
  autolayer(ajustados,series = "Ajustado")+  
  ggtitle("Log ICAP y Modelo Ajustado")+  
  xlab("Año")+ylab("log ICAP")+  
  scale_color_manual(values=c("#A4C4FC","#E80808"))+  
  theme(legend.position = "bottom",legend.title = element_blank()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



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

```
autoplot(serie_ent,series="ICAP entrenamiento")+  
  autolayer(exp(ajustados),series = "exp[ajustado]")+  
  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() )
```

ICAP y Exponencial del Modelo Ajustado



7 Predicción

Se realizará la predicción para el conjunto test del ICAP.

```
mod2_pred <- SSMModel(log_serie_ent ~ SSMtrend(
  2,Q = list(mod2e$model$Q[,1][1,1],mod2e$model$Q[,1][2,2]))+
  SSMseasonal(period = 12,
    sea.type = "trigonometric", Q = mod2e$model$Q[,1][3,3]), H =mod2e$model$H)

pred <- predict(mod2_pred,n.ahead=19,interval="prediction",level=0.95)
pred
```

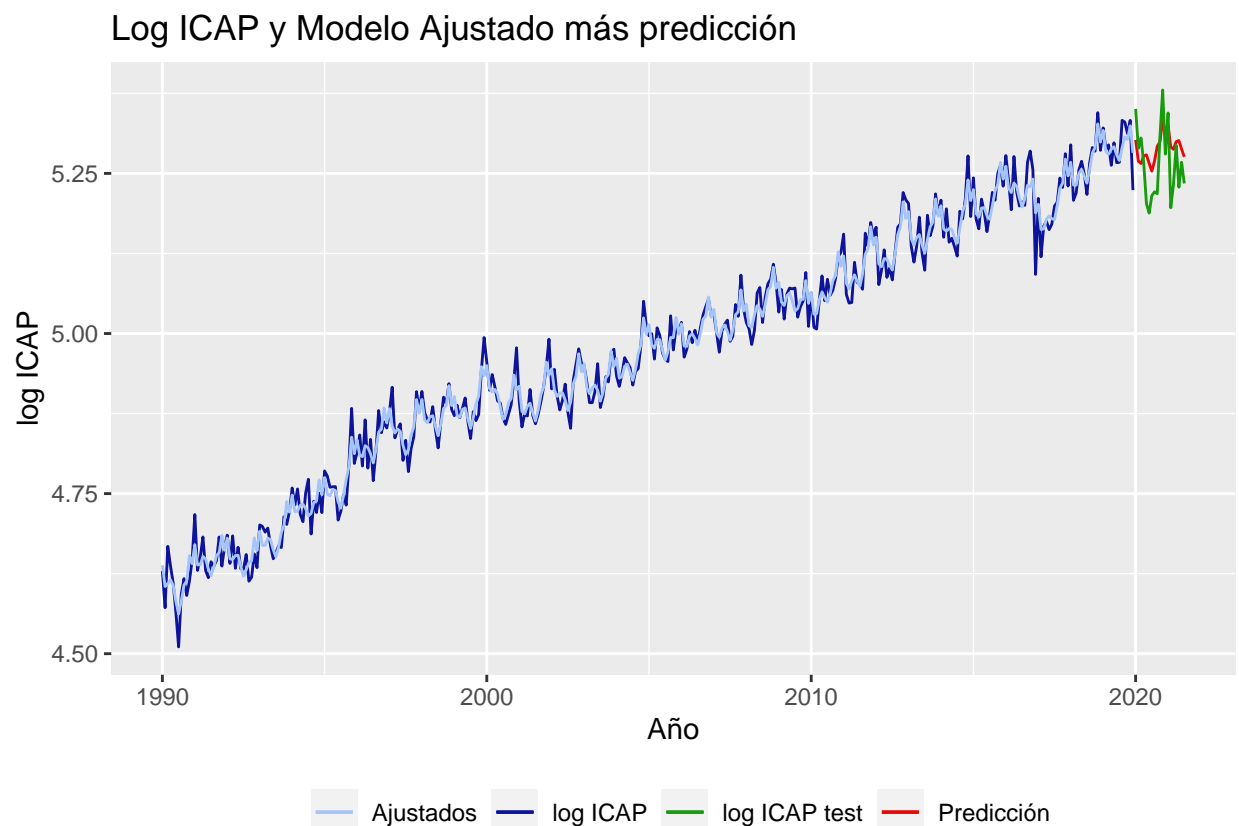
```
##           fit      lwr      upr
## Jan 2020 5.302438 5.240320 5.364557
## Feb 2020 5.269044 5.202612 5.335476
## Mar 2020 5.265448 5.194966 5.335930
## Apr 2020 5.277418 5.203107 5.351730
## May 2020 5.279042 5.201089 5.356995
## Jun 2020 5.265782 5.184350 5.347214
## Jul 2020 5.253571 5.168803 5.338340
## Aug 2020 5.269707 5.181729 5.357686
## Sep 2020 5.292285 5.201210 5.383361
## Oct 2020 5.299439 5.205369 5.393509
```



```
## Nov 2020 5.334249 5.237276 5.431221
## Dec 2020 5.307446 5.207656 5.407237
## Jan 2021 5.324681 5.221652 5.427710
## Feb 2021 5.291287 5.185432 5.397142
## Mar 2021 5.287691 5.179083 5.396298
## Apr 2021 5.299661 5.188370 5.410952
## May 2021 5.301285 5.187372 5.415197
## Jun 2021 5.288025 5.171550 5.404499
## Jul 2021 5.275814 5.156833 5.394796
```

Primero graficamos la serie del ICAP tanto de entrenamiento como de test aplicado a logaritmo y el ajuste del modelo *Local Level Model with Seasonal* más su predicción.

```
autoplot(log_serie_ent,series="log ICAP")+
  autolayer(ajustados,series = "Ajustados")+
  autolayer(pred[,1],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() )
```

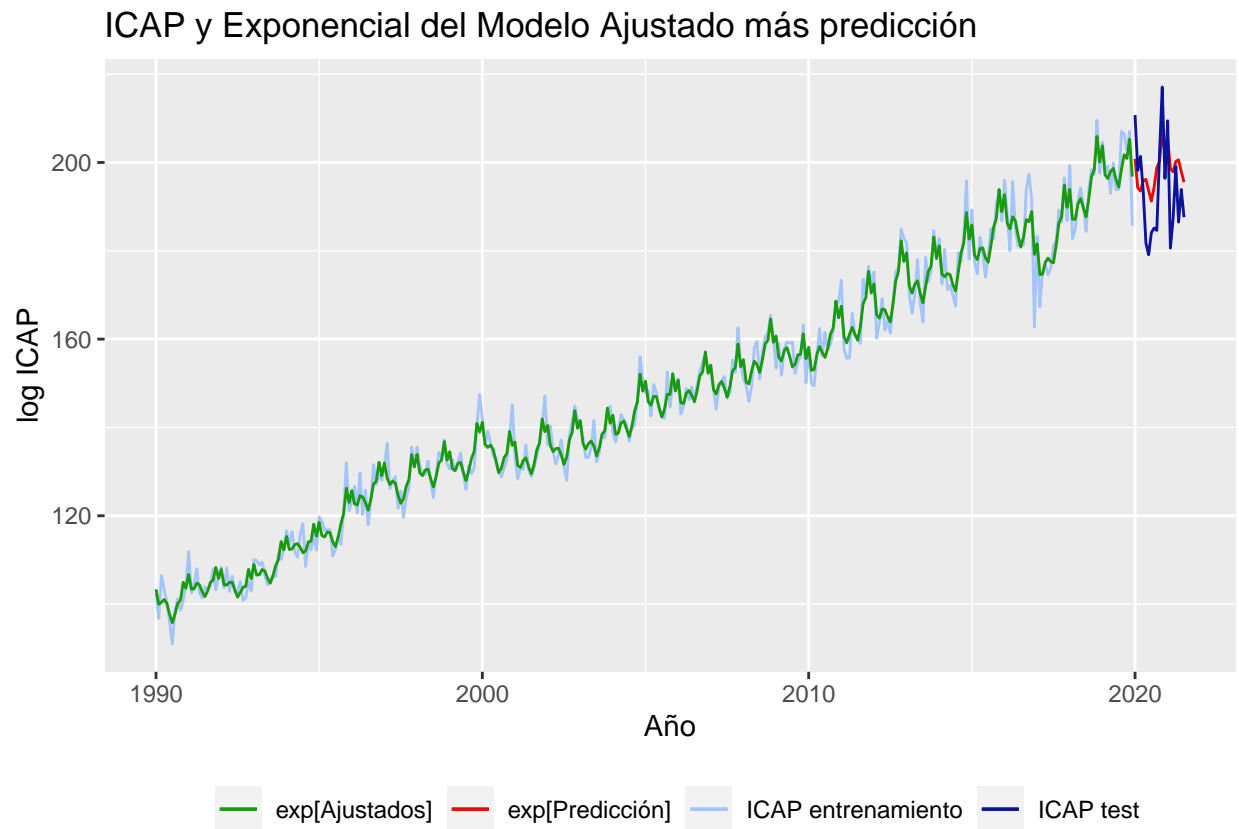


Ahora graficamos la serie original del ICAP tanto de entrenamiento como de test y el inverso del logaritmo (exponencial) al modelo ajustado más su predicción.

```

autoplot(serie_ent,series="ICAP entrenamiento")+
  autolayer(exp(ajustados),series = "exp[Ajustados]")+
  autolayer(exp(pred[,1]),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 aplicados a logaritmo.

```

mape<-function(y,f){
  pe<-((y-f)/y)
  mape1<-(sum(abs(pe))/length(y))*100
  return(mape1)
}
mape(log(serie_test),pred[,1])

```

```
## [1] 0.8299066
```

Se calcula el MAPE para la predicción de los datos del conjunto del test originales y aplicando exponente a los predichos.

```
mape_test<-mape(serie_test,exp(pred[,1]))  
mape_test
```

```
## [1] 4.4467
```

El MAPE es igual a 4.4466998, esto aplicado a la exponencial de las predicciones.

9 Comparación de la predicción

El MAPE obtenido en el primer trabajo por el método de suavizamiento exponencial específicamente al modelo Holt-Winters aplicado a la serie de test fue de 4.5956761, en el segundo trabajo el MAPE obtenido por el modelo $SARIMA(0, 1, 1)(1, 0, 1)_{12}$ de Box y Jenkins fue 4.6803368 y por último el MAPE obtenido por el modelo *Local Level Model with Seasonal* de espacio estado es 4.4466998.

Por tal motivo se concluye que el mejor modelo que ajusta los datos del ICAP de Bolivia es el modelo de espacio estado específicamente el *Local Level Model with Seasonal*.

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.

Se transformó la serie con logaritmo para estabilizar varianza y se ajustó tres modelos de espacio estado siendo el mejor el *Local Level Model with Seasonal* presentando el menor AIC de todos los 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 con la dócima de Shapiro Wilk.

Al pronosticar los datos con el modelo *Local Level Model with Seasonal* y obtener el MAPE del mismo, al comparar con el de Holt Winters y $SARIMA(0, 1, 1)(1, 0, 1)_{12}$ realizado en anteriores trabajos, se pudo evidenciar que el menor MAPE lo tiene el de Espacio Estado específicamente *Local Level Model with Seasonal*, siendo así este el modelo óptimo de las tres metodologías propuestas para realizar pronósticos a la serie del ICAP de Bolivia.