Predicción: Series Temporales

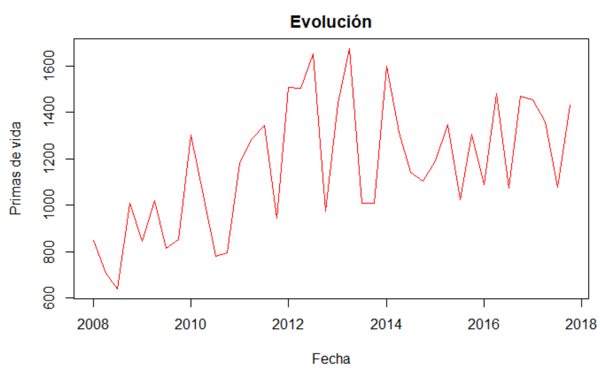
Carlota Echevarría

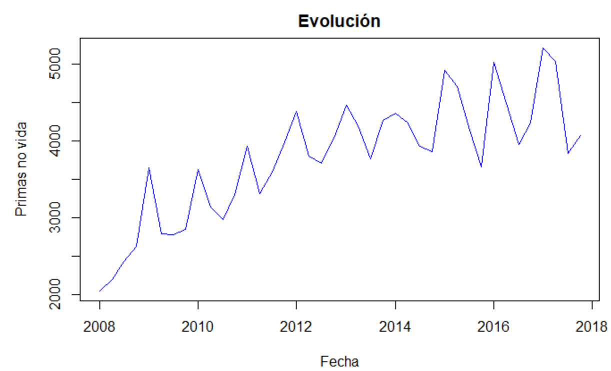
13/11/2019

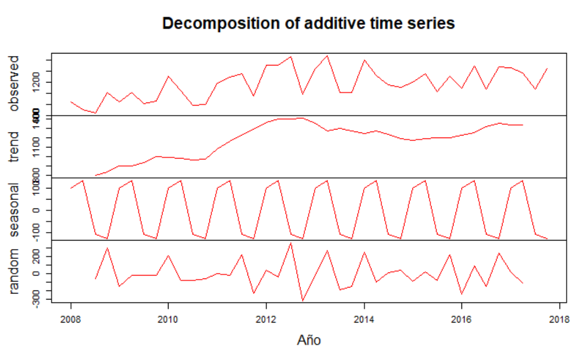
El objetivo de este trabajo es predecir la prima de los seguros de Mapfre para el año 2018, mediante la estimación de los modelos ETS y ARIMA y comparar ambos según los resultados.

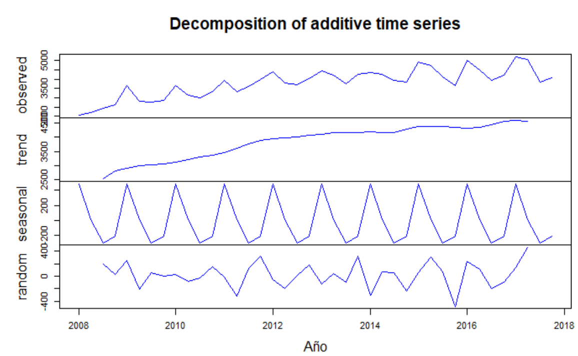
Nuestra base de datos contiene 4 variables de 40 observaciones y distribuidas en cuatrimestres desde 2008 hasta 2018, nuestras observaciones a predecir son las primas de seguro de vida y las de no vida. Se realizará la predicción de 2017 y 2018.

Ambas series no son estacionarias en media ni varianza, la tendencia es creciente y presentan estacionalidad.



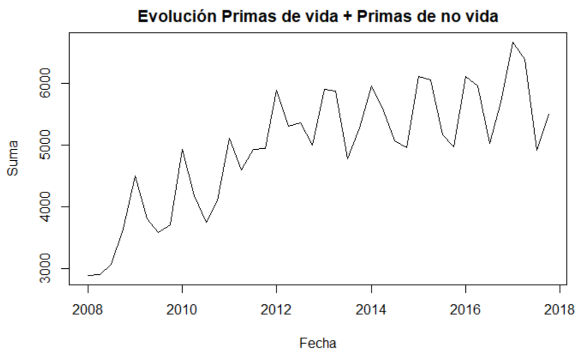


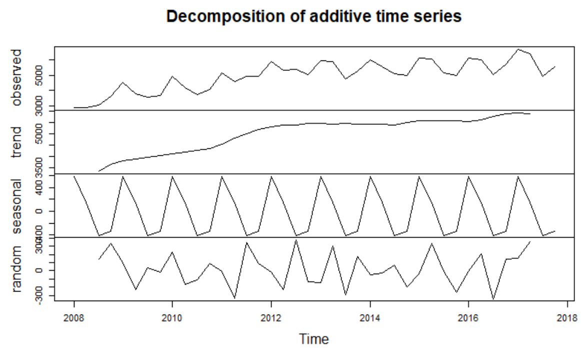




Se ha realizado la suma de las primas de vida y no vida, para realizar la estimación de los modelos y posteriormente la predicción.

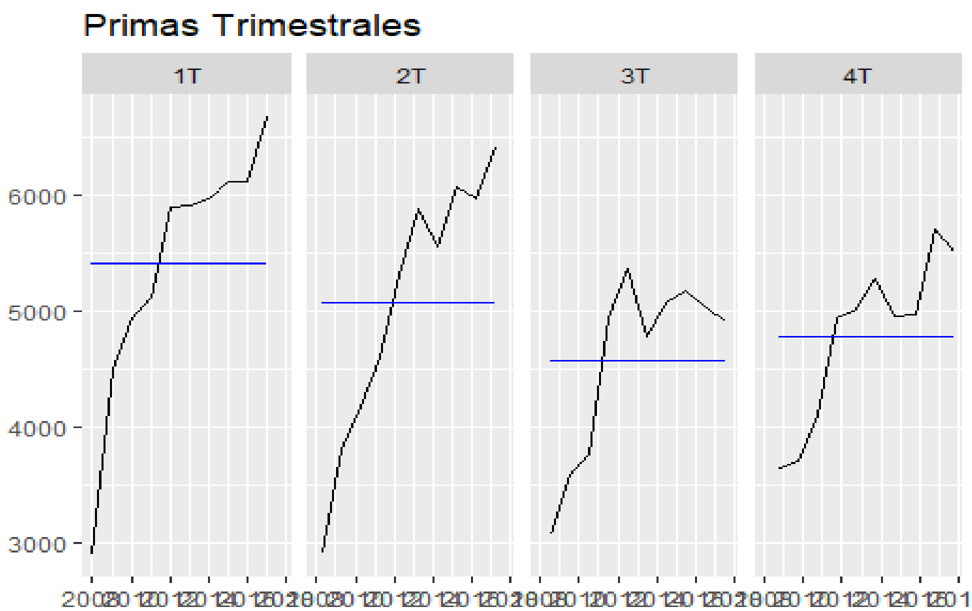
Se explicará primero el modelo ETS , seguido del modelo ARIMA.





Modelo ETS:

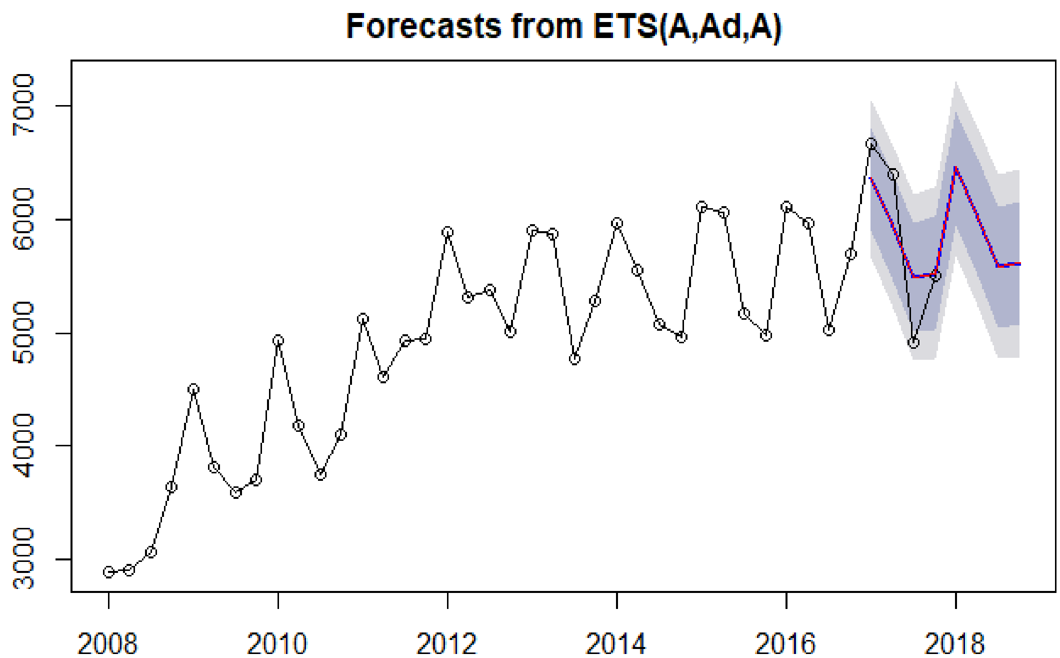
En el primer y segundo trimestre del año las primas de seguros son más elevadas, y el tercer trimestre más reducida.



Se ha estimado el modelo ETS, se trata de un modelo aditivo amortiguado y se ha realizado una predicción para las primas de seguros de los cuatrimestres de 2017 y 2018.

Respecto a la predicción de 2017, comparamos la observación la predicción y se observa que la predicción de los dos primeros cuatrimestres esta por debajo de la realidad, respecto al tercer y cuarto trimestre por encima.

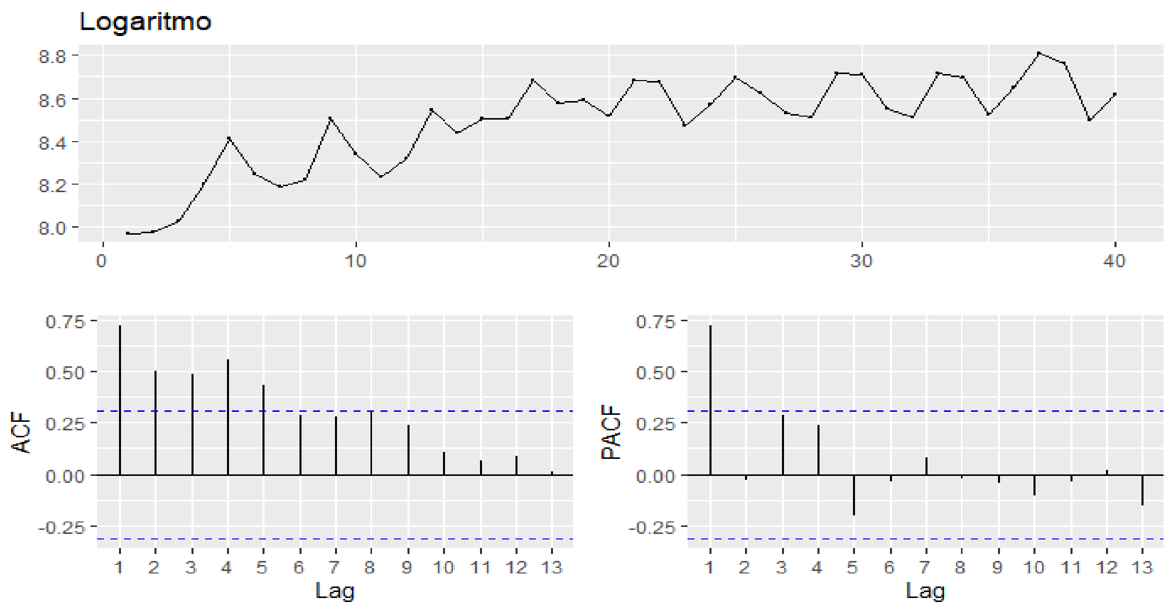
Es el resultado de que las primas son más elevadas en el primer y segundo cuatrimestre, y en el tercer y cuarto disminuyen notablemente. El resultado más favorable de la predicción es el del cuarto, puesto que es en el que las primas se estabilizan.

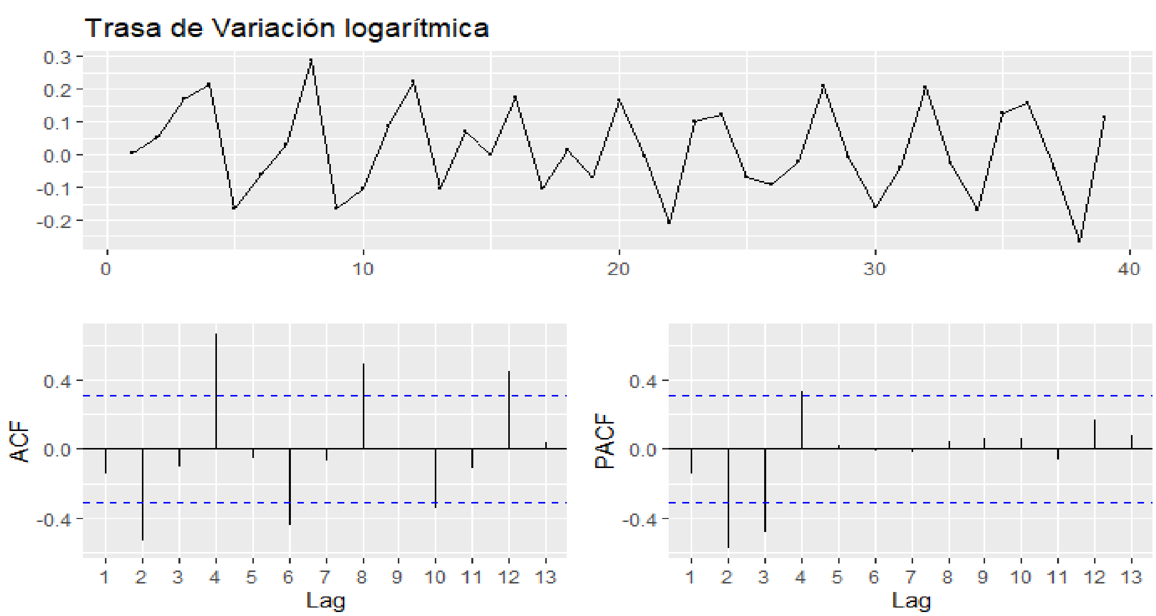


	Predicción VS Observación		
	Predicción	**Observación**	**Error**
2017 Q1	6360.1	6674.6	314.5
2017 Q2	5930.4	6398.6	468.2
2017 Q3	5490.8	4913.4	-577.44
2017 Q4	5528.5	5507.2	-21.3

Modelo ARIMA:

Se ha observado anteriormente que la serie no es estacionaria ni en media ni varianza, por lo que se ha realizado una tasa de variación logarítmica, convirtiendo la serie en estacionaria tanto en varianza como en media.

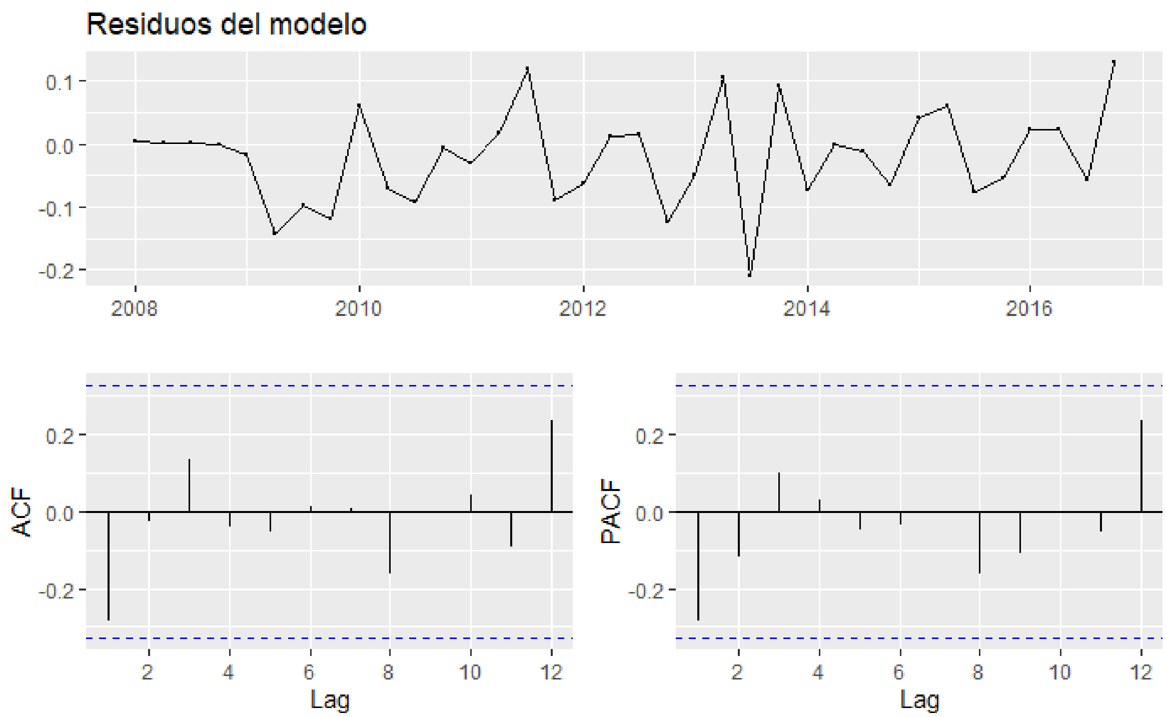




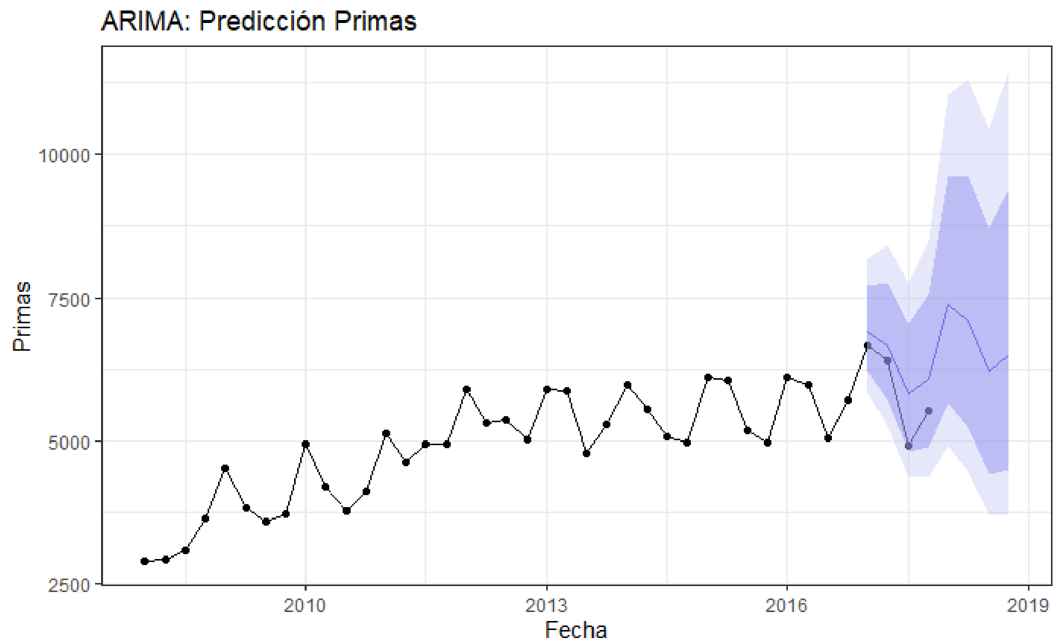
Tras convertir la serie en estacionaria, con la función de autocorrelación simple intuimos que hay un componente estacional en nuestra serie.

Procedemos a calcular el modelo y se obtiene el modelo: ARIMA(0,1,0) * $(0,1,1)_4$.

Se observa que las autocorrelaciones (tanto simples como parciales) se encuentran dentro de las bandas de confianza, lo que indica que los residuos están incorrelados. Adicionalmente se realiza el contraste Ljung-Box, cuyo p-valor es mayor a un nivel de significatividad del 5%, por tanto, a través del contraste, podemos decir que las autocorrelaciones son iguales a cero.



Se observa que la predicción de los cuatrimestres de 2017, las primas son más elevadas que en los valores observados, y que el mayor error se encuentra en el tercer trimestre.



	Predicción VS Observación		
	Predicción	**Observación**	**Error**
2017 Q1	6912.2	6674.6	-237.6
2017 Q2	6652.4	6398.6	-253.8
2017 Q3	5814.4	4913.4	-901
2017 Q4	6085.5	5507.2	-578.3

Conclusión:

Comparando ambos modelos, el error obtenido en el primer y segundo cuatrimestre es menor con el modelo ARIMA, pero el error del modelo ETS en el tercer y cuarto cuatrimestre es menor.

Si comparamos el error cuadrático medio de ambos modelos, concluimos que el **mejor modelo de estimación es el ETS.**

- RMSE (ETS): 308.6
- RMSE (ARIMA): 388.9

	Comparación del ERROR de ambos modelos	
	Error ETS	**Error ARIMA**
2017 Q1	314.5	-237.6
2017 Q2	468.2	-253.8
2017 Q3	-577.44	-901
2017 Q4	-21.3	-578.3

Bibliografía:

Hyndman, R. J., A. B. Koehler, J. K. Ord and R. D. Snyder (2008). Forecasting with exponential smoothing: the state space approach. Berlin: Springer-Verlag.

Anexo- Código:

```
#Library
library(readr)
require(forecast)
require(xts)
require(ggplot2)
library(ggfortify)
library(dplyr)
```

```
#Datos

Primas_mapfre <- read.csv("data/Primas_mapfre.csv", header = TRUE,
    sep = ";", dec = "," )
View(Primas_mapfre)
```

```
#Suma de primas

Primas_mapfre$Suma <- (Primas_mapfre$Primas_vida + Primas_mapfre$Primas_n
o_vida)



#Análisis Exploratorio

primas_ts <- ts(Primas_mapfre$Suma, start = c(2008,1), frequency = 4)
primas_desc <- decompose(primas_ts)
plot(primas_desc, col = "blue")

#Conversión de datos

xPrimas = xts(Primas_mapfre$Suma, order.by = as.Date(Primas_mapfre$Fecha,
"%m/%d/%Y"),frequency=4)

#Generate quarterly data
xPrimas = to.quarterly(xPrimas)

#Transform to zoo data (forecast package)
zPrimas = as.zoo(xPrimas$xPrimas.Close)
names(zPrimas)="Primas"

##Plot Serie
autoplot(zPrimas)+ggtitle("Primas trimestrales")+xlab("Trimestres")+ylab(
"Primas")

#Seasonal Plot
ggfreqplot(as.ts(zPrimas),freq=4,nrow=1,facet.labeller=c("1T","2T","3T","
4T"))+ggtitle("Primas Trimestrales")
```

```
#Select number of observation to compare forecast
cOmit=4

#Data Size
nObs=length(zPrimas)

#sub_sample
#oVentas=zVentas[1:(nObs-cOmit),]
oPrimas <- window(zPrimas,start=index(zPrimas[1]),end=index(zPrimas[nObs-
cOmit]))
View(oPrimas)
```

Modelo ETS

```
## Select automatic ETS
etsfit<-ets(oPrimas,damped=TRUE)


#forecast model
fprimas.ets=forecast(etsfit)


#Results
summary(fprimas.ets)
```

```
 Forecast method: ETS(A,Ad,A)

 Model Information:
 ETS(A,Ad,A)

 Call:
  ets(y = oPrimas, damped = TRUE)

   Smoothing parameters:
     alpha = 0.2424
     beta  = 1e-04
     gamma = 1e-04
     phi   = 0.9505

   Initial states:
     l = 2931.1472
     b = 178.7185
     s = -335.3013 -349.5291 114.6065 570.2239

   sigma:  356.3793
```

```
     AIC     AICc      BIC 
561.7218 570.5218 577.5570 

Error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE
Training set -5.403611 308.6335 249.7589 -0.4948873 5.595178 0.6033052
                  ACF1
Training set 0.1276507

Forecasts:
        Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2017 Q1       6360.067 5903.349 6816.786 5661.577 7058.558
2017 Q2       5930.371 5460.419 6400.324 5211.641 6649.102
2017 Q3       5490.845 5008.011 5973.679 4752.414 6229.275
2017 Q4       5528.540 5033.151 6023.929 4770.908 6286.172
2018 Q1       6456.184 5948.533 6963.836 5679.798 7232.570
2018 Q2       6021.727 5502.104 6541.349 5227.033 6816.421
2018 Q3       5577.675 5046.343 6109.006 4765.074 6390.276
2018 Q4       5611.068 5068.274 6153.862 4780.937 6441.200
```

```
#Plot
plot(fprimas.ets)
lines(window(zPrimas),type="o")
```

```
#Actual and Forecast
matrix(c(fprimas.ets$mean[1:cOmit],zPrimas[(nObs-cOmit+1):nObs]),ncol=2)
```

Modelo ARIMA

```
zPrimas_log = log(zPrimas)
df_newl <- data.frame(value = as.vector(zPrimas_log),
                     time = time(zPrimas_log))
ggplot(df_newl)+geom_point(aes(x=time,y=value))+geom_line(aes(x=time,y=va
lue))+ylab("Primas")+ggtitle("Modelo ARIMA")+xlab("Años")
```

```
#Estacionaria en varianza
ggtsdisplay(zPrimas_log)
```

```r
#Estacionaria en media
ggtsdisplay(diff(zPrimas_log))
```

```r
fit1=auto.arima(oPrimas,lambda=0)
summary(fit1)
```

```
 Series: oPrimas
 ARIMA(0,1,0)(0,1,1)[4]
 Box Cox transformation: lambda= 0

 Coefficients:
          sma1
       -0.6185
 s.e.   0.1830

 sigma^2 estimated as 0.007238:  log likelihood=31.97
 AIC=-59.94   AICc=-59.51   BIC=-57.07

 Training set error measures:
                     ME      RMSE       MAE       MPE     MAPE      MASE
 Training set -93.54831 388.8768 301.9985 -2.36378 6.164447 0.7294924
                    ACF1
 Training set -0.3575378
```

```r
#residual analysis
ggtsdisplay(fit1$residuals)
```

```r
#box-Ljung Test
Box.test(fit1$residuals,lag=6, fitdf=3, type="Lj")
  Box-Ljung test

 data:  fit1$residuals
 X-squared = 4.1928, df = 3, p-value = 0.2414

Box.test(fit1$residuals,lag=8, fitdf=3, type="Lj")

  Box-Ljung test

 data:  fit1$residuals
 X-squared = 5.4651, df = 5, p-value = 0.3618

Box.test(fit1$residuals,lag=12, fitdf=3, type="Lj")
  Box-Ljung test
```

```
 data:  fit1$residuals
 X-squared = 9.1663, df = 9, p-value = 0.4221
```

```
Box.test(fit1$residuals,lag=24, fitdf=3, type="Lj")
  Box-Ljung test

 data:  fit1$residuals
 X-squared = 20.534, df = 21, p-value = 0.4877
```

```
df_new <- data.frame(value = as.vector(zPrimas), time = time(zPrimas))

ggplot(df_new)+geom_point(aes(x=time,y=value))+geom_line(aes(x=time,y=val
ue))+ geom_forecast(fprimas.arima,alpha=0.4)+xlab("Fecha")+ylab("Primas")
+ggtitle("ARIMA: Predicción Primas") + theme_bw()

fprimas.arima
```