Seguros\_ETS\_ARIMA

Val Huerta

11/11/2019

#Librerias

library(readr)  
require(forecast)

require(xts)

require(ggplot2)

library(ggfortify)

library(dplyr)

**#Lectura de datos**

Primas\_mapfre <- read.csv("~/Documents/Master Data Science.CUNEF/Predicción/Práctica 5/Práctica 5. Seguros/data/Primas\_mapfre.csv", header = TRUE, sep = ";", dec = ",")  
  
# View(Primas\_mapfre)

Procedo a la suma de las primas de vida y no vida para así posteriorme realizar la predicción.

**#Suma de las primas**

Primas\_mapfre$Suma <- (Primas\_mapfre$Primas\_vida + Primas\_mapfre$Primas\_no\_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")

Podemos observar como la tendencia es creciente positiva, también podemos observar que existe estacionalidad.

#Manipulación de datos

#Create a XTS object  
#Convert data to XTS  
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")

## Don't know how to automatically pick scale for object of type yearqtr. Defaulting to continuous.

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

Eliminamos los datos de 2017 para realizar posteriormente con ETS la predicción de 2017 y 2018

#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)

## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):  
## running command ''/usr/bin/otool' -L '/Library/Frameworks/R.framework/  
## Resources/modules/R\_de.so'' had status 1

# 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

Obtengo como resultado la predicción “Point Forecast”, así como los valores en los intervalos de confianza al 80 y al 95%

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

Nos da como resultado un modelo ETS, que se corresponde en este orden (error,tendencia y estacionalidad) Error = A: Tendencia lineal Tendencia = Ad: Tendencia lineal amortiguada Estacionalidad = A: Tendencia lineal

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

## [,1] [,2]  
## [1,] 6360.067 6674.6  
## [2,] 5930.371 6398.6  
## [3,] 5490.845 4913.4  
## [4,] 5528.540 5507.2

Obtenemos la matriz de el valor medio de mi predicción [1] y el valor actual[2]

**#MODELO ARIMA**

Como vemos la serie temporal no es estacionaria, por lo que no es constante en varianza y media. Debemos realizar una transformación logarítmica primero de la varianza y observamos los resultados, después, si no apreciamos estacionariedad aun asi, procedemos a realizar el logaritmo de la tasa de variación.

#Log transformation  
  
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=value))+ylab("Primas")+ggtitle("Modelo ARIMA")+xlab("Años")

## Don't know how to automatically pick scale for object of type yearqtr. Defaulting to continuous.

#Estacionaria en varianza   
ggtsdisplay(zPrimas\_log)

#Los ACF y PACF son los gráficos que muestran los residuos.

CREO: En la funcion de autocorrelación ACF, obtenemos como resultado que las correlaciones que se encuentran dentro del intervalo son estadisticamente 0, por lo que lo que ocurrió hasta el periodo 5 no nos interesa, pero si del 6 en adelante.

#Estacionaria en media   
ggtsdisplay(diff(zPrimas\_log))

# Modelo ARIMA

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

#residual analysis  
ggtsdisplay(fit1$residuals)

#El arima lo que me está generando es el modelo sin los residuos (en ACF y PACF) de manera   
#automática que estaba obteniendo al tranformar en logaritmo y al realizar la diferencia para   
#hacer estacionaria la serie.

La hipótesis nula de este test las autocorrelaciones de hasta un desfase k son iguales a 0.Es decir, que los valores de los datos son aleatorios e independientes hasta un cierto número de desfases. En nuestro caso son 12 desfases porque son 12 retardos.

#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

fprimas.arima=forecast(fit1)  
  
#Se acepta la hipotesis nula porque nuestro nivel de significación es superior a 0.05. Es decir, a un nivel de confianza de 95% se acepta la hipótesis nula.

**#Representación gráfica ARIMA**

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=value))+ geom\_forecast(fprimas.arima,alpha=0.4)+ggtitle("ARIMA: Predicción Primas")

fprimas.arima

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2017 Q1 6912.247 6198.153 7708.611 5850.501 8166.677  
## 2017 Q2 6652.463 5701.808 7761.619 5254.856 8421.783  
## 2017 Q3 5814.438 4813.798 7023.079 4355.813 7761.510  
## 2017 Q4 6085.474 4893.130 7568.365 4359.649 8494.490  
## 2018 Q1 7374.785 5657.687 9613.019 4917.033 11061.030  
## 2018 Q2 7097.617 5232.525 9627.508 4452.688 11313.655  
## 2018 Q3 6203.515 4415.261 8716.041 3687.896 10435.112  
## 2018 Q4 6492.689 4476.124 9417.747 3676.185 11467.053