PREVISIÓN VENTAS APPLE

Sergio Cañón

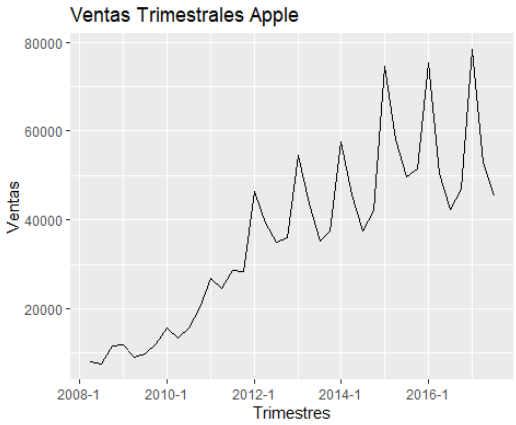
16/11/2020

1. INTRODUCCIÓN

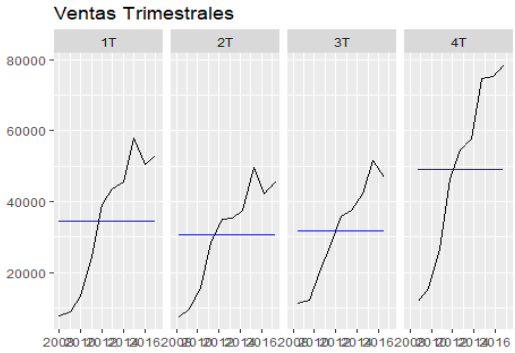
El objetivo del presente trabajo es el de analizar las ventas trimestrales de la empresa Apple desde el 2008 hasta el 2016 creando dos modelos: Arima y otro ETS, Con el fin de poder hace una previsión para los próximos años seleccionando los modelos con una validación cruzada.

2. VISUALIZACIÓN

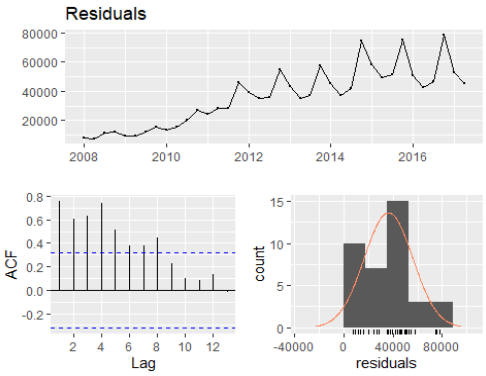
Lo primero que llama la atención es la gran tendencia que ha tenido en los últimos años la empresa y que tiene picos si bajadas cada cierto tiempo lo que hace pensar en una posible tendencia estacionaria.



El siguiente gráfico se puede ver como a los 4 trimestres eso son muy parecidos en cuanto al comportamiento en los primeros años subieron muchísimo las ventas en los finales es tan caro así que puedo hacer seguir pensando en esta finalidad.



En el gráfico inferior se ve claramente en los residuos no hay ruido blanco en los

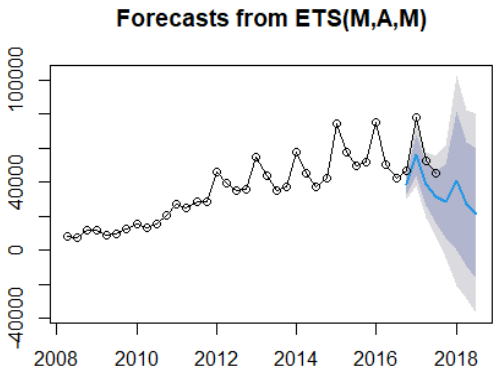


residuos y que la serie no es estacionaria ya que el valor de la función de autocorrelación no decae de manera exponencial a medida que aumentan los rezagos en el tiempo.

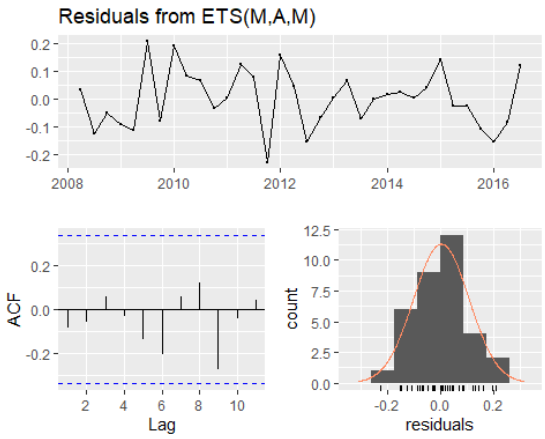
3. ETS

Los modelos ETS son una familia de modelos de series de tiempo con un modelo de espacio de estados subyacente que consta de un componente de nivel, un componente de tendencia (T), un componente estacional (S) y un término de error (E).

El modelo ETS obtenido es el (M, A, M) lo que indica: Error multiplicativo, tendencia aditiva y estacionalidad multiplicativa.



So modelo hemos intentado coger datos anteriores al 2016 para predecir los siguientes cómo se ve como del oeste eso suele predecir menor precio que lo observado.



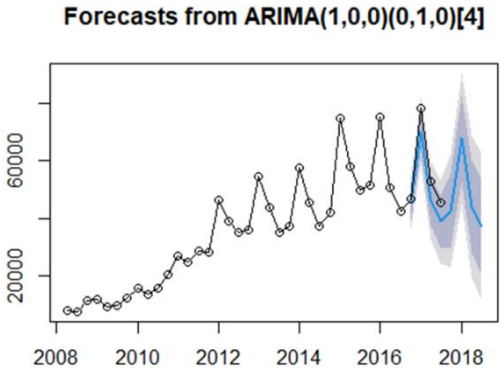
Cabe decir que este modelo ha eliminado la estacionalidad.

4. ARIMA

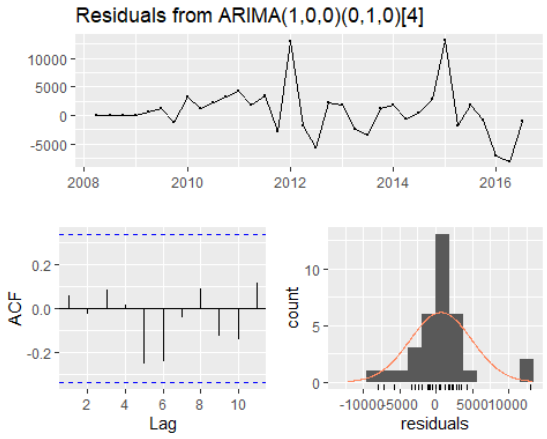
Es una generalización de una autorregresivo de media móvil modelo (ARMA). Ambos modelos se ajustan a datos de series de tiempo para comprender mejor los datos o para predecir puntos futuros de la serie (pronóstico). Los modelos ARIMA se aplican en algunos casos donde los datos muestran evidencia de no estacionariedad, donde un paso de diferenciación inicial (correspondiente a la parte "integrada" del modelo) se puede aplicar una o más veces para eliminar la no estacionariedad.

Tras eliminar la estacionalidad de la serie y aplicar Arima, hemos obtenido el siguiente modelo:

`ARIMA(1,0,0)(0,1,0)[4]`



Este modelo parece más preciso que la anterior puesto que los dos primeros cuatrimestres los predice perfectamente y los dos siguientes se aproximan bastante y además, a diferencia del anterior las previsiones son más optimistas.



Cabe decir que este modelo ha eliminado la estacionalidad.

5. CROSS VALIDATION

El RMSE del ETS es de 6485 y el de Arima es 5707, así que, como pensábamos, el modelo más preciso es el Arima

6. ANEXO CÓDIGO

```
library(readr)
library(forecast)

library(ggplot2)
library(xts)

library(ggfortify)

library(ggplot2)
```

```
rawData <- read_delim("IngresosApple.csv",
                      ";", escape_double = FALSE, trim_ws = TRUE)

## Parsed with column specification:
## cols(
##   Trimestre = col_character(),
##   Ingresos = col_double()
## )

rawVentas <- rawData$Ingresos
rawDate <- seq(as.Date("2008/04/01"),
               as.Date("2017/07/01"), by = "quarter")

xVentas <- xts(rawVentas, order.by = rawDate)
xVentas <- to.quarterly(xVentas)
zVentas <- as.zoo(xVentas$xVentas.Close)

tsVentas <- ts(coredata(zVentas), start = c(2008, 1), frequency = 4)
```

```
autoplot(zVentas)+
  ggtitle("Ventas Trimestrales Apple")+
  xlab("Trimestres")+
  ylab("Ventas")
```

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

## Warning: `group_by_()` is deprecated as of dplyr 0.7.0.
## Please use `group_by()` instead.
## See vignette('programming') for more help
```

```
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generat
ed.

## Warning: `summarise_()` is deprecated as of dplyr 0.7.0.
## Please use `summarise()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generat
ed.
```

```
checkresiduals(tsVentas)

## Warning in modeldf.default(object): Could not find appropriate degrees
of
## freedom for this model.
```

```
# TRAIN
nObs <- length(zVentas)
cOmit <- 4
oVentas <- window(zVentas,start=index(zVentas[1]),end=index(zVentas[nObs-c
Omit]))
```

```
ets <- ets(oVentas)

fore_ets <- forecast(ets)

summary(fore_ets) #Multiplicative Holt-Winters' method with multiplicative
 errors

##
## Forecast method: ETS(M,A,M)
##
## Model Information:
## ETS(M,A,M)
##
## Call:
##  ets(y = oVentas)
##
##   Smoothing parameters:
##     alpha = 0.4652
##     beta  = 0.4652
##     gamma = 0.5311
##
##   Initial states:
##     l = 7122.371
##     b = 1492.6432
##     s = 1.1496 1.1214 0.8327 0.8962
##
```

```
##   sigma:  0.1175
##
##      AIC     AICc      BIC
## 679.8611 687.3611 693.5983
##
## Error measures:
##                     ME    RMSE     MAE        MPE     MAPE      MASE
   ACF1
## Training set -322.6123 4042.51 2733.75 -0.8774628 8.428508 0.3901659 0.
1319957
##
## Forecasts:
##         Point Forecast       Lo 80    Hi 80      Lo 95     Hi 95
## 2016 Q4       38518.08  32716.5807 44319.59  29645.450  47390.72
## 2017 Q1       56113.05  44132.4513 68093.64  37790.308  74435.78
## 2017 Q2       38594.10  26026.2941 51161.90  19373.301  57814.89
## 2017 Q3       31722.60  16354.0087 47091.19   8218.367  55226.84
## 2017 Q4       28760.23   7397.6189 50122.85  -3911.066  61431.53
## 2018 Q1       40797.81    184.7347 81410.88 -21314.526 102910.14
## 2018 Q2       27185.53  -9001.0390 63372.10 -28157.051  82528.11
## 2018 Q3       21504.22 -16699.6423 59708.07 -36923.543  79931.98
```

```
# ERROR = ULTIPLICATIVO, TENENDCIA= ADITIVA, ESTACIONALIAD= NONE


plot(fore_ets)
lines(window(zVentas),type="o")
```

```
checkresiduals(ets)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,A,M)
## Q* = 7.9837, df = 3, p-value = 0.04635
##
## Model df: 8.   Total lags used: 11
```

```
ets_cv <- tsCV(zVentas,fore_ets,drift=TRUE, h=1)
ets_cv
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2008         NA   NA   NA
## 2009   NA   NA   NA   NA
## 2010   NA   NA   NA   NA
## 2011   NA   NA   NA   NA
## 2012   NA   NA   NA   NA
## 2013   NA   NA   NA   NA
## 2014   NA   NA   NA   NA
## 2015   NA   NA   NA   NA
```

```
## 2016   NA   NA   NA   NA
## 2017   NA   NA   NA
```

```
#ARIMA
```

```
arima <- auto.arima(oVentas)
summary(arima)
```

```
## Series: oVentas
## ARIMA(1,0,0)(0,1,0)[4]
##
## Coefficients:
##          ar1
##       0.8364
## s.e.  0.0894
##
## sigma^2 estimated as 21100218:  log likelihood=-295.63
## AIC=595.27   AICc=595.71   BIC=598.07
##
## Training set error measures:
##                     ME     RMSE     MAE      MPE     MAPE     MASE
 ACF1
## Training set 677.8963 4242.316 2839.06 2.930918 8.179109 0.405196 0.056
27805
```

```
fore_arima <- forecast(arima)
summary(fore_arima)
```

```
##
## Forecast method: ARIMA(1,0,0)(0,1,0)[4]
##
## Model Information:
## Series: oVentas
## ARIMA(1,0,0)(0,1,0)[4]
##
## Coefficients:
##          ar1
##       0.8364
## s.e.  0.0894
##
## sigma^2 estimated as 21100218:  log likelihood=-295.63
## AIC=595.27   AICc=595.71   BIC=598.07
##
## Error measures:
##                     ME     RMSE     MAE      MPE     MAPE     MASE
 ACF1
## Training set 677.8963 4242.316 2839.06 2.930918 8.179109 0.405196 0.056
27805
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2016 Q4       45439.27 39552.47 51326.07 36436.18 54442.36
## 2017 Q1       70253.69 62579.04 77928.33 58516.33 81991.05
## 2017 Q2       46315.95 37605.98 55025.92 32995.20 59636.70
## 2017 Q3       38810.59 29444.06 48177.12 24485.72 53135.46
```

```
## 2017 Q4       42472.05 29641.75 55302.35 22849.79 62094.31
## 2018 Q1       67771.76 52993.11 82550.42 45169.76 90373.77
## 2018 Q2       44239.96 28238.59 60241.32 19767.98 68711.94
## 2018 Q3       37074.13 20270.12 53878.15 11374.61 62773.65

plot(fore_arima)
lines(window(zVentas),type="o")
```

```
checkresiduals(fore_arima)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(0,1,0)[4]
## Q* = 5.8201, df = 6, p-value = 0.4436
##
## Model df: 1.   Total lags used: 7

# CV
fore_ets_time <- function(x, h) {
  forecast(ets(x), h = h)
}
fore_arima_time <- function(x, h) {
  forecast(auto.arima(x), h = h)
}


cv_ets <- tsCV(tsVentas, fore_ets_time, h=1)
cv_arima <- tsCV(tsVentas, fore_arima_time, h=1)

plot(cv_ets)
```

```
sqrt(mean(cv_ets^2, na.rm = T))

## [1] 6485.73

sqrt(mean(cv_arima^2, na.rm = T))

## [1] 5707.563
```