Ventas de Apple

Antonio Pascual Hernández

16/11/2020

Introducción

Se debe elegir el modelo ETS y el modelo ARIMA que mejor predice las ventas, habiendo dejado fuera de la estimación los trimestres del 2017.

Cargamos los datos:

```
apple <- read.csv("IngresosApple.csv", sep  = ";")
head(apple)
```

```
##   Trimestre Ingresos
## 1   Q2 2008     7980
## 2   Q3 2008     7561
## 3   Q4 2008    11520
## 4   Q1 2009    11880
## 5   Q2 2009     9084
## 6   Q3 2009     9734
```

Cargamos las librerias que vamos a necesitar:

```
require(forecast)
require(xts)
require(ggplot2)
library(ggfortify) #Plot Monthplot
library(dplyr)
```

Como las fechas están representadas por trimestres, debemos escribirlas en formato fecha para poder continuar con el análisis:

```
fechas <- seq(as.Date("2008-04-01"), as.Date("2017-09-30"), by = "quarter")
fechas
```

```
##  [1] "2008-04-01" "2008-07-01" "2008-10-01" "2009-01-01" "2009-04-01"
##  [6] "2009-07-01" "2009-10-01" "2010-01-01" "2010-04-01" "2010-07-01"
## [11] "2010-10-01" "2011-01-01" "2011-04-01" "2011-07-01" "2011-10-01"
## [16] "2012-01-01" "2012-04-01" "2012-07-01" "2012-10-01" "2013-01-01"
## [21] "2013-04-01" "2013-07-01" "2013-10-01" "2014-01-01" "2014-04-01"
## [26] "2014-07-01" "2014-10-01" "2015-01-01" "2015-04-01" "2015-07-01"
## [31] "2015-10-01" "2016-01-01" "2016-04-01" "2016-07-01" "2016-10-01"
## [36] "2017-01-01" "2017-04-01" "2017-07-01"
```

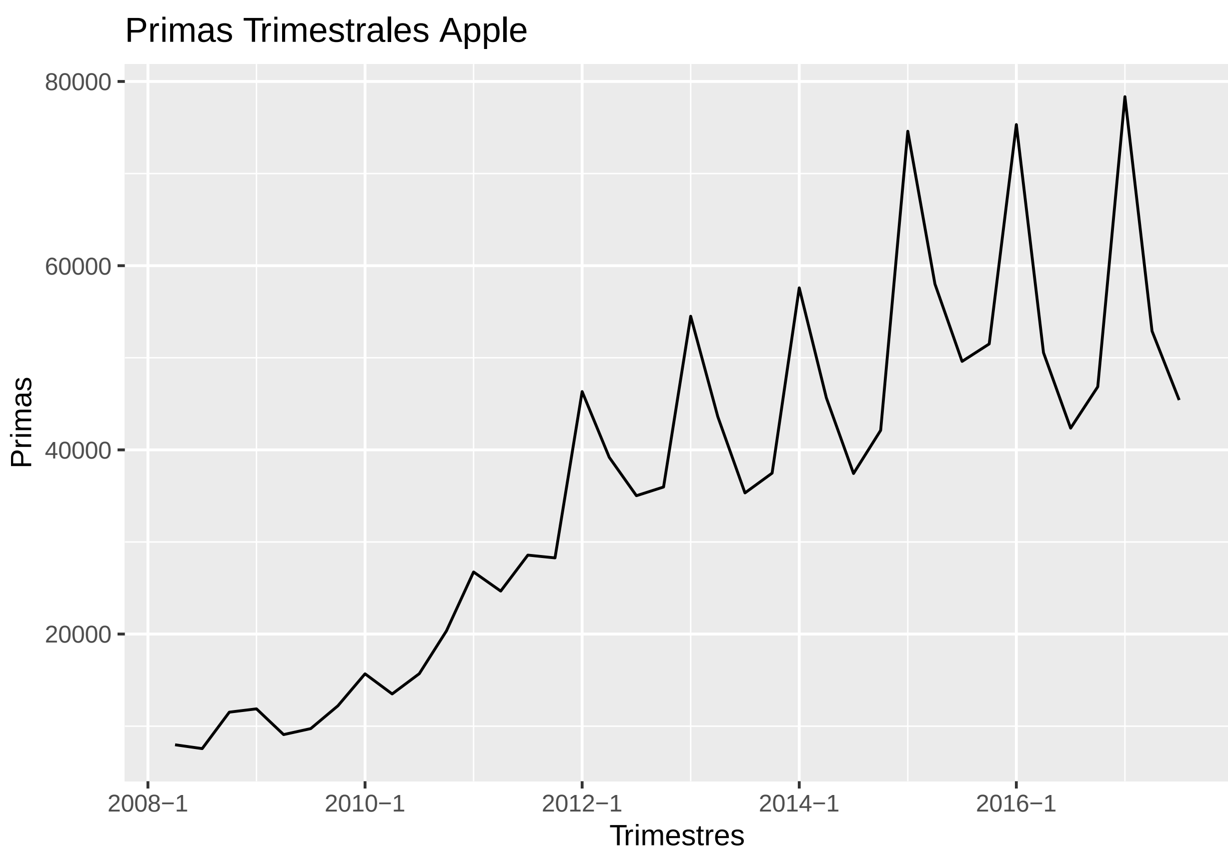
```
apple <- mutate(apple, Date =fechas)
```

Comprobamos que las fechas se han cambiado correctamente:

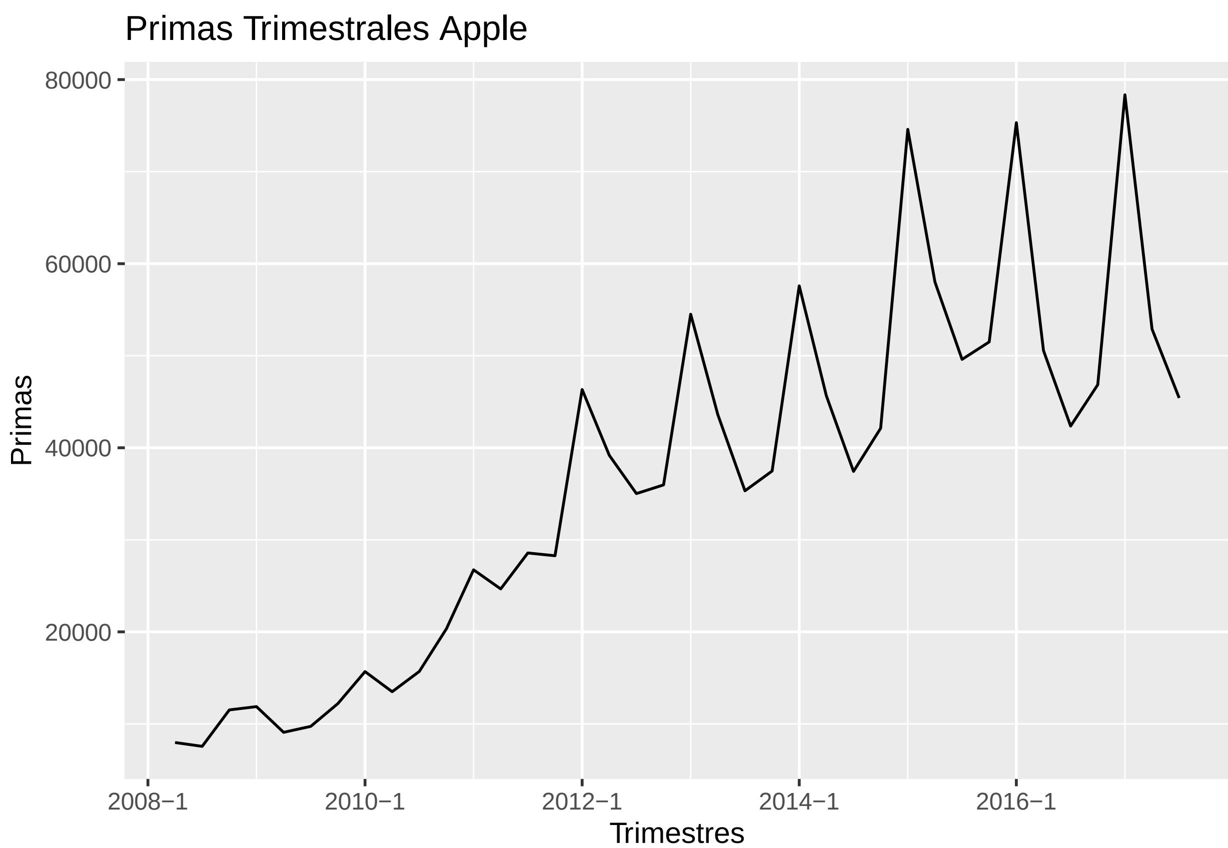
```
head(apple)
```

```
##   Trimestre Ingresos       Date
## 1   Q2 2008     7980 2008-04-01
## 2   Q3 2008     7561 2008-07-01
## 3   Q4 2008    11520 2008-10-01
## 4   Q1 2009    11880 2009-01-01
## 5   Q2 2009     9084 2009-04-01
## 6   Q3 2009     9734 2009-07-01
```

```
#Convert data to XTS
xapple=xts(apple$Ingresos, order.by = as.Date(apple$Date),frequency=4)
xapple=to.quarterly(xapple)
zapple=as.zoo(xapple$xapple.Close)
autoplot(zapple)+ggtitle("Primas Trimestrales Apple")+xlab("Trimestres")+ylab("Primas")
```

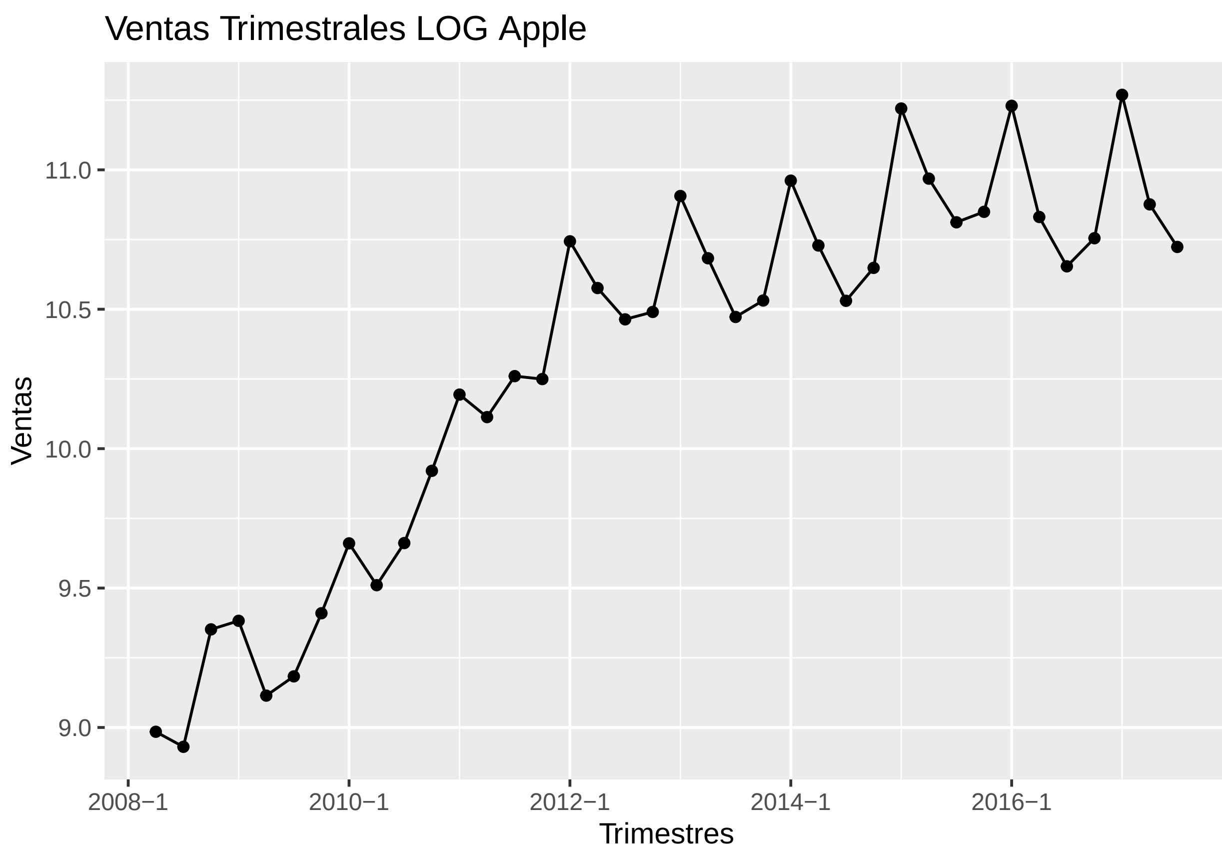



```
#Convert data to XTS
xapple=xts(apple$Ingresos, order.by = as.Date(apple$Date),frequency=4)
xapple=to.quarterly(xapple)
zapple=as.zoo(xapple$xapple.Close)
autoplot(zapple)+ggtitle("Primas Trimestrales Apple")+xlab("Trimestres")+ylab("Primas")
```

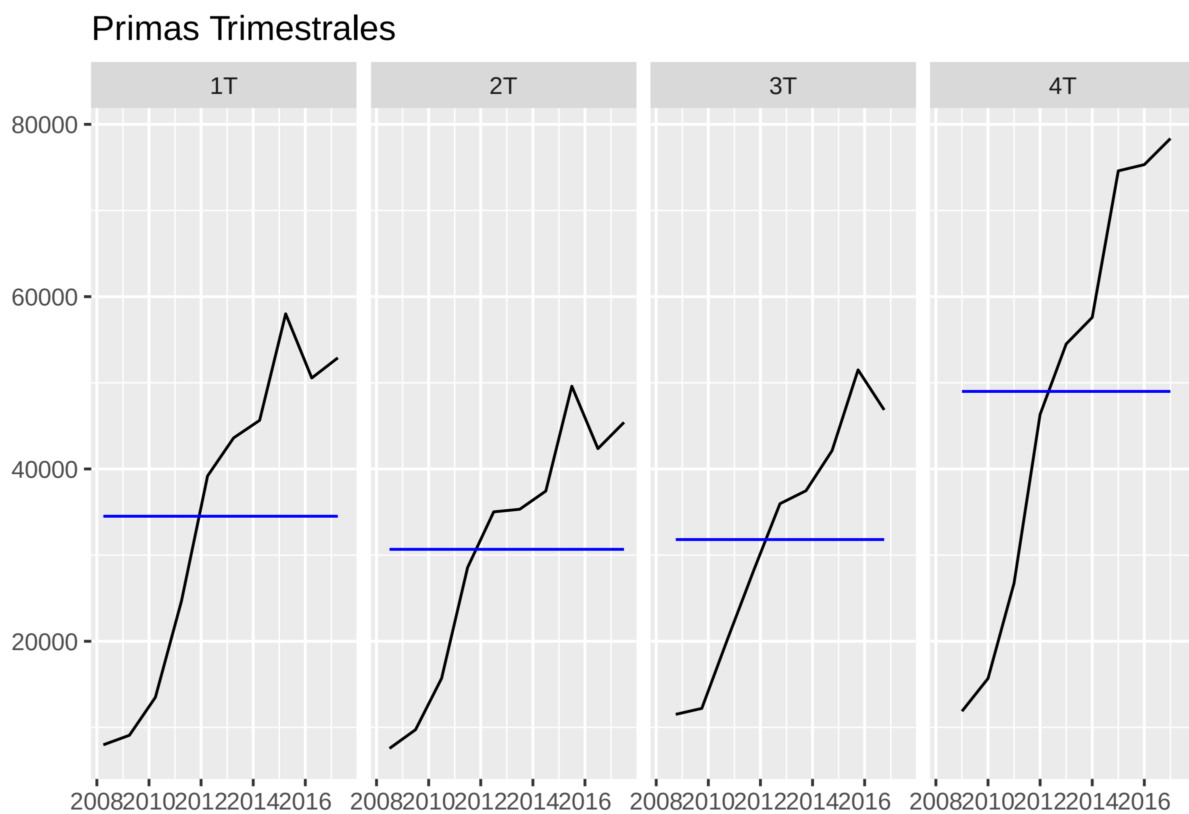


Transformacion Logarítmica

```
zlapple=log(zapple)
df_newl <- data.frame(value = as.vector(zlapple),
                      time = time(zlapple))
ggplot(df_newl)+geom_point(aes(x=time,y=value))+geom_line(aes(x=time,y=value))+ylab("Ventas")+ggtitle("V
```




```r
#Transform to ts data
tsapple=ts(coredata(zapple), start = c(2008, 2), frequency = 4)
#Seasonal Plot
ggfreqplot(tsapple,freq=4,nrow=1,facet.labeller=c("1T","2T","3T","4T"))+ggtitle("Primas Trimestrales")
```



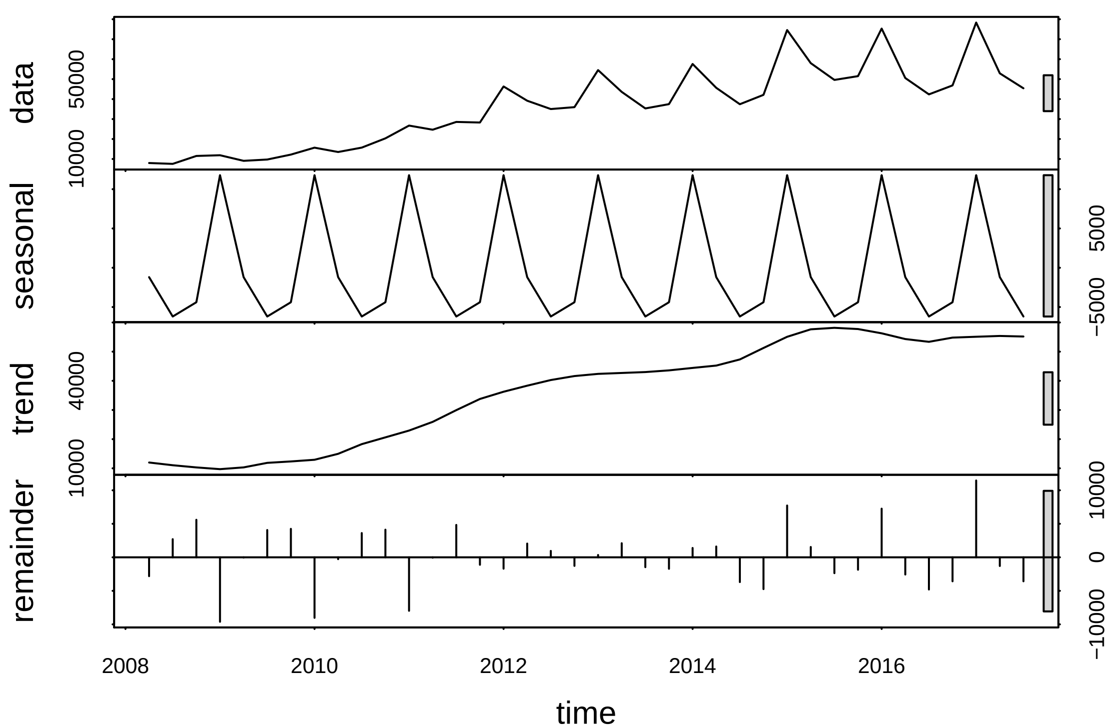
Descomposicion

```r
stl(tsapple[, 1], s.window = "periodic")
```

```
##  Call:
##  stl(x = tsapple[, 1], s.window = "periodic")
##
## Components
##           seasonal     trend   remainder
## 2008 Q2 -1191.515 11987.713 -2816.19786
## 2008 Q3 -6202.795 11054.647  2709.14756
## 2008 Q4 -4388.396 10306.954  5601.44193
## 2009 Q1 11782.720  9712.795 -9615.51515
## 2009 Q2 -1191.515 10313.378   -37.86329
## 2009 Q3 -6202.795 11865.051  4071.74373
## 2009 Q4 -4388.396 12348.763  4246.63276
## 2010 Q1 11782.720 12927.130 -9026.85026
## 2010 Q2 -1191.515 14968.422  -277.90712
## 2010 Q3 -6202.795 18282.820  3619.97453
## 2010 Q4 -4388.396 20608.629  4122.76624
## 2011 Q1 11782.720 22936.090 -7977.81022
## 2011 Q2 -1191.515 25917.203   -58.68777
## 2011 Q3 -6202.795 29940.698  4833.09640
```

```
## 2011 Q4 -4388.396 33777.770 -1119.37412
## 2012 Q1 11782.720 36249.530 -1699.24976
## 2012 Q2 -1191.515 38326.357  2051.15766
## 2012 Q3 -6202.795 40267.415   958.37982
## 2012 Q4 -4388.396 41633.253 -1278.85770
## 2013 Q1 11782.720 42379.047   350.23228
## 2013 Q2 -1191.515 42689.157  2105.35767
## 2013 Q3 -6202.795 42998.839 -1473.04409
## 2013 Q4 -4388.396 43579.104 -1718.70816
## 2014 Q1 11782.720 44414.890  1396.39005
## 2014 Q2 -1191.515 45222.672  1614.84282
## 2014 Q3 -6202.795 47330.414 -3695.61937
## 2014 Q4 -4388.396 51256.536 -4745.14062
## 2015 Q1 11782.720 55088.745  7727.53443
## 2015 Q2 -1191.515 57657.107  1544.40748
## 2015 Q3 -6202.795 58173.837 -2366.04237
## 2015 Q4 -4388.396 57747.384 -1857.98841
## 2016 Q1 11782.720 56283.309  7257.97053
## 2016 Q2 -1191.515 54318.201 -2569.68577
## 2016 Q3 -6202.795 53365.704 -4804.90925
## 2016 Q4 -4388.396 54818.937 -3578.54150
## 2017 Q1 11782.720 55104.114 11464.16602
## 2017 Q2 -1191.515 55382.281 -1294.76586
## 2017 Q3 -6202.795 55190.924 -3580.12898
```

```
plot(stl(tsapple[, 1], s.window = "periodic"))
```



Modelos ETS

Eliminamos los últimos 3 trimestres. Estimamos y predecimos con modelo no estacionales.

Como debemos dejar fuera de la estimación los trimestres de 2017 y en nuestro dataset encontramos 3 trimestres de 2017, nuestro c0mit será igual a 3.

```
#Select number of observation to compare forecast
cOmit=3

#Data Size
nObs=length(zapple)

#sub_sample
oapple <- window(zapple,start=index(zapple[1]),end=index(zapple[nObs-cOmit]))

#Fit Simple Exponential Smoothing
fit1 <- ses(oapple)

#Fit Holt
fit2 <- holt(oapple)

#Fit Holt- exponential
fit3 <- holt(oapple ,exponential=TRUE,initial="simple")

#Fit Holt - damped
fit4 <- holt(oapple,damped=TRUE)

#Fit Holt - (exponential+damped)
fit5 <- holt(oapple,exponential=TRUE,damped=TRUE)
```

Resultados de los modelos:

```
fit1$model
```

```
## Simple exponential smoothing
##
## Call:
##  ses(y = oapple)
##
##   Smoothing parameters:
##     alpha = 0.4288
##
##   Initial states:
##     l = 9016.5881
##
##   sigma:  10303.87
##
##      AIC     AICc      BIC
## 775.1970 775.9712 779.8631
```

```
fit2$model
```

```
## Holt's method
##
## Call:
##  holt(y = oapple)
##
##   Smoothing parameters:
##     alpha = 1e-04
##     beta  = 1e-04
##
##   Initial states:
##     l = 7035.2209
##     b = 1512.4299
##
##   sigma:  9428.668
##
##      AIC     AICc      BIC
## 770.7953 772.8642 778.5720
```

```
fit3$model
```

```
## Holt's method with exponential trend
##
## Call:
##  holt(y = oapple, initial = "simple", exponential = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.3725
##     beta  = 0.2346
##
##   Initial states:
##     l = 7980
##     b = 0.9475
##
##   sigma:  0.2622
```

```
fit4$model
```

```
## Damped Holt's method
##
## Call:
##  holt(y = oapple, damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.2198
##     beta  = 1e-04
##     phi   = 0.98
##
##   Initial states:
##     l = 7035.3965
##     b = 1512.5793
##
##   sigma:  10020.06
##
```

```
##      AIC     AICc      BIC
## 775.9060 778.9060 785.2381
```

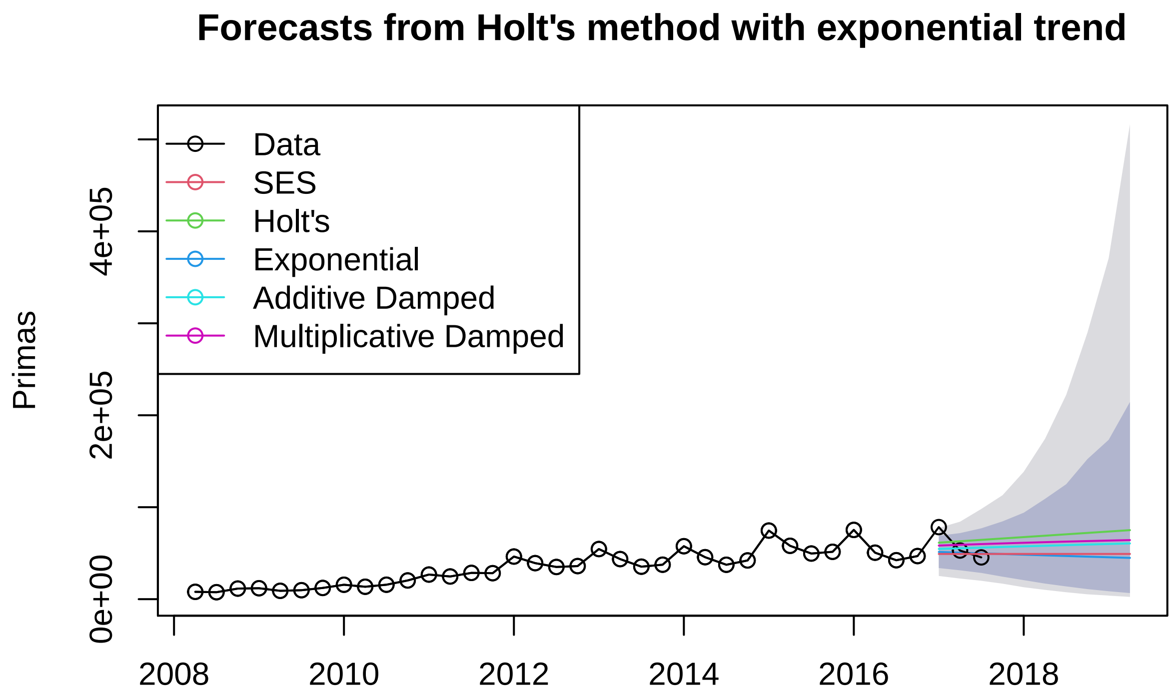
```r
fit5$model
```

```
## Damped Holt's method with exponential trend
##
## Call:
##  holt(y = oapple, damped = TRUE, exponential = TRUE)
##
##   Smoothing parameters:
##     alpha = 1e-04
##     beta  = 1e-04
##     phi   = 0.9357
##
##   Initial states:
##     l = 7821.9792
##     b = 1.1643
##
##   sigma:  0.2466
##
##      AIC     AICc      BIC
## 757.2782 760.2782 766.6103
```

De acuerso con el AIC, el modelo 5 (exponential+damped) sería la mejor opción

Representación gráfica de los modelos

```r
plot(fit3, type="o", ylab="Primas",  flwd=1, plot.conf=FALSE)
lines(window(zapple),type="o")
lines(fit1$mean,col=2)
lines(fit2$mean,col=3)
lines(fit4$mean,col=5)
lines(fit5$mean,col=6)
legend("topleft", lty=1, pch=1, col=1:6,
       c("Data","SES","Holt's","Exponential",
         "Additive Damped","Multiplicative Damped"))
```

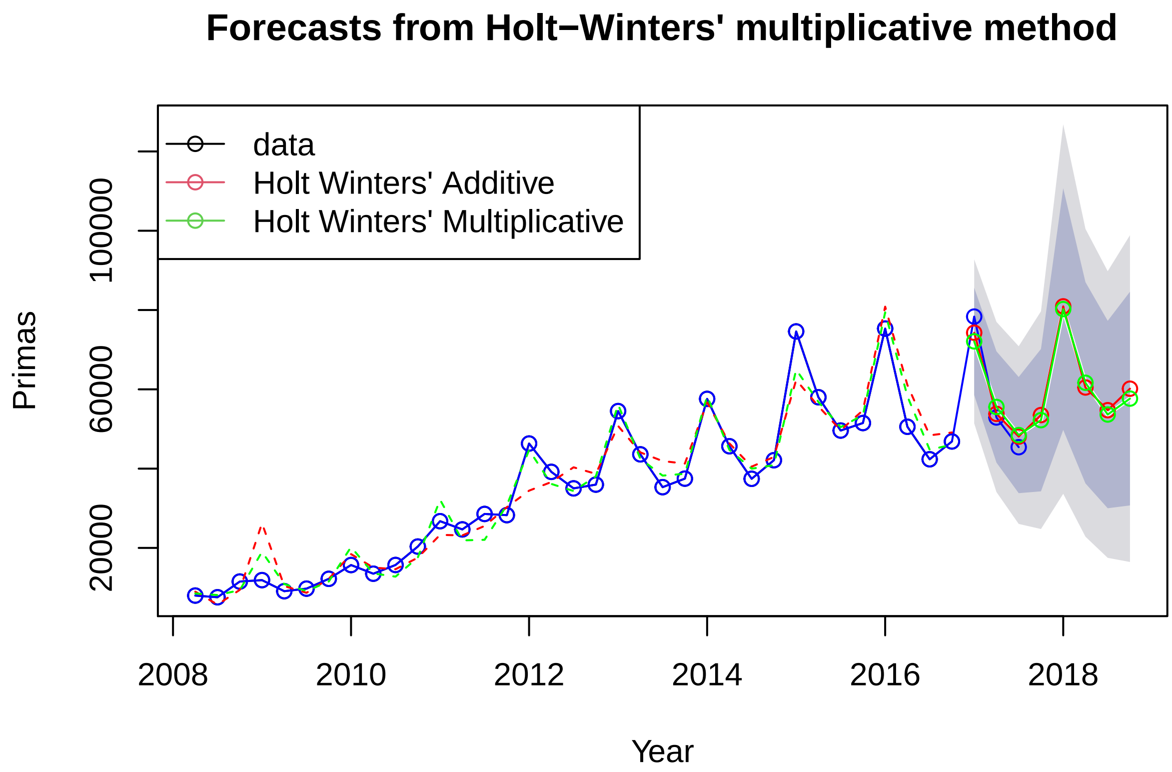


A continuación estimamos modelos no estacionales:

```
#seasonal model Holt-winters
fit6 <- hw(oapple,seasonal="additive")
fit7 <- hw(oapple,seasonal="multiplicative")



#Plot models
plot(fit7,ylab="Primas",
     plot.conf=FALSE, type="o", fcol="white", xlab="Year")
lines(window(zapple),type="o",col="blue")
lines(fitted(fit6), col="red", lty=2)
lines(fitted(fit7), col="green", lty=2)
lines(fit6$mean, type="o", col="red")
lines(fit7$mean, type="o", col="green")
legend("topleft",lty=1, pch=1, col=1:3,
       c("data","Holt Winters' Additive","Holt Winters' Multiplicative"))
```



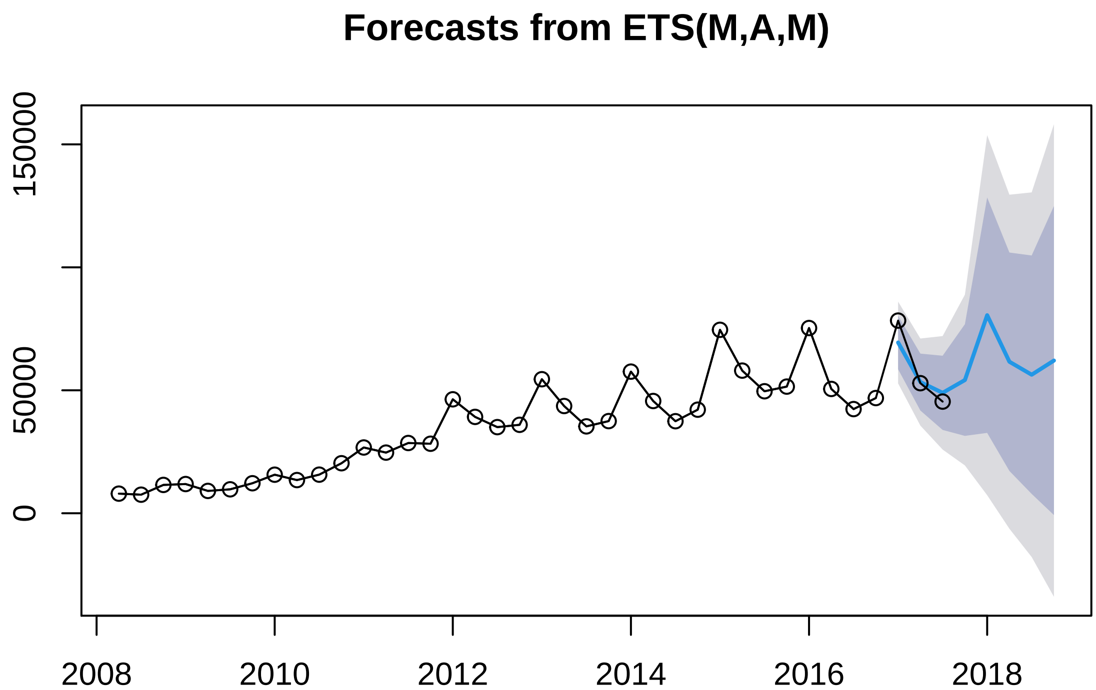
Seleccionamos de forma automática el modelo ETS

```
## Select automatic ETS
etsfit<-ets(oapple)
#forecast model
fventas.ets=forecast(etsfit)
#Results
summary(fventas.ets)
```

```
##
## Forecast method: ETS(M,A,M)
##
## Model Information:
## ETS(M,A,M)
##
## Call:
##  ets(y = oapple)
##
##   Smoothing parameters:
##     alpha = 0.493
##     beta  = 0.493
##     gamma = 0.507
##
##   Initial states:
```

```
##     l = 7125.3462
##     b = 1485.7975
##     s = 1.1511 1.1163 0.8322 0.9004
##
##   sigma:  0.1222
##
##      AIC     AICc      BIC
## 703.9538 711.1538 717.9519
##
## Error measures:
##                   ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set -41.934 4120.155 2883.262 -0.297759 8.677434 0.4160202 0.1438481
##
## Forecasts:
##         Point Forecast     Lo 80     Hi 80      Lo 95     Hi 95
## 2017 Q1       69439.83 58568.387  80311.27  52813.394  86066.26
## 2017 Q2       53347.98 41773.016  64922.95  35645.598  71050.37
## 2017 Q3       48972.04 33884.613  64059.47  25897.811  72046.27
## 2017 Q4       54176.09 31475.035  76877.14  19457.824  88894.35
## 2018 Q1       80540.07 32680.293 128399.85   7344.857 153735.28
## 2018 Q2       61619.03 17211.212 106026.85  -6296.866 129534.93
## 2018 Q3       56344.41  7869.773 104819.05 -17791.151 130479.97
## 2018 Q4       62103.80  -718.363 124925.97 -33974.410 158182.02
```

```r
#Plot
plot(fventas.ets)
lines(window(zapple),type="o")
```



Comparación entre los valores actuales y los valores predichos:

```
matrix(c(fventas.ets$mean[1:cOmit],zapple[(nObs-cOmit+1):nObs]),ncol=2)
```

```
##          [,1]  [,2]
## [1,] 69439.83 78351
## [2,] 53347.98 52896
## [3,] 48972.04 45408
```

Predicciones y Precisión

```
etsfit<-ets(window(tsapple,end=2016+4/4))
fventas.ets=forecast(etsfit,h=cOmit)
forecast:::testaccuracy(fventas.ets$mean,window(tsapple,start=2017),test = NULL, d = NULL, D = NULL)
```

```
##           ME         RMSE          MAE          MPE         MAPE         ACF1
## -9937.360471 10259.914274  9937.360471   -20.733460    20.733460    -0.500000
##    Theil's U
##     1.667969
```

Modelos ARIMA

```
#Select number of observation to compare forecast
cOmit=3

#Data Size
nObs=length(zapple)

#sub_sample
oapple <- window(zapple,start=index(zapple[1]),end=index(zapple[nObs-cOmit]))

#out sample (real data to forecast performance)
papple <- window(zapple,start=index(zapple[nObs-cOmit+1]),end=index(zapple[nObs]))
```

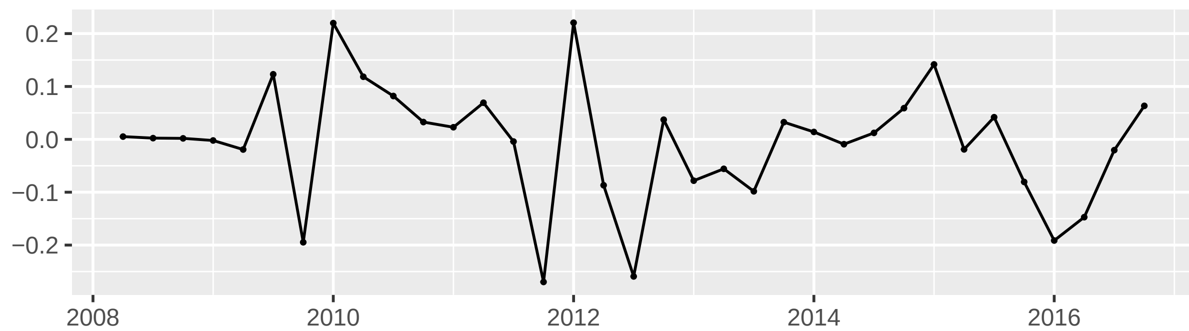
Creamos un modelo ARIMA

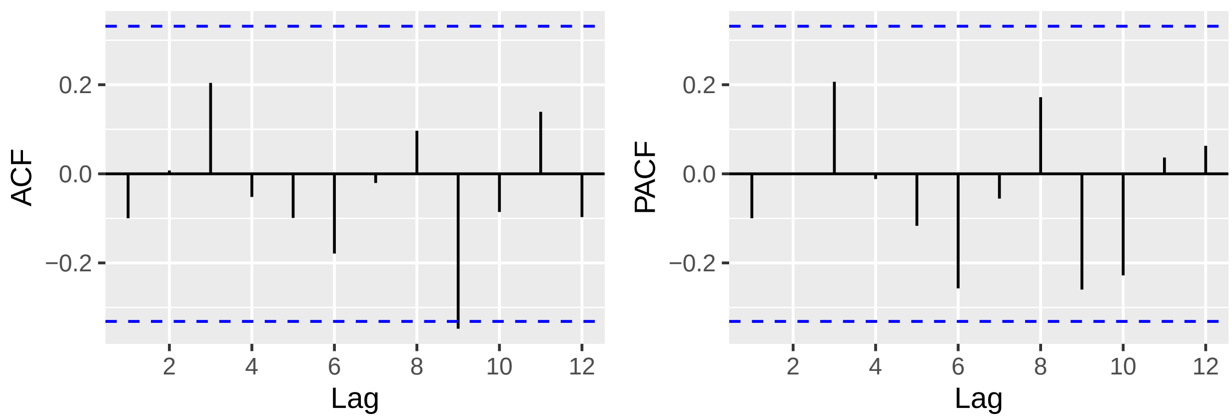
```
fit1=auto.arima(oapple,lambda=0)
summary(fit1)
```

```
## Series: oapple
## ARIMA(0,1,0)(0,1,0)[4]
## Box Cox transformation: lambda= 0
##
## sigma^2 estimated as 0.01472:  log likelihood=20.72
## AIC=-39.45   AICc=-39.3   BIC=-38.04
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set -764.5058 4786.405 3054.054 -1.321616 8.284962 0.4406634 0.1269135
```

Análisis de los Residuos:

```
ggtsdisplay(fit1$residuals)
```





El gráfico del ACF no muestra correlación entre los residuos. Se observa en el gráfico del PACF que ningún lag es significativo.

Box-Ljung Test

```r
Box.test(fit1$residuals,lag=4, fitdf=3, type="Lj")
```

```
##
##  Box-Ljung test
##
## data:  fit1$residuals
## X-squared = 2.1794, df = 1, p-value = 0.1399
```

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

```
##
##  Box-Ljung test
##
## data:  fit1$residuals
## X-squared = 4.501, df = 5, p-value = 0.4797
```

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

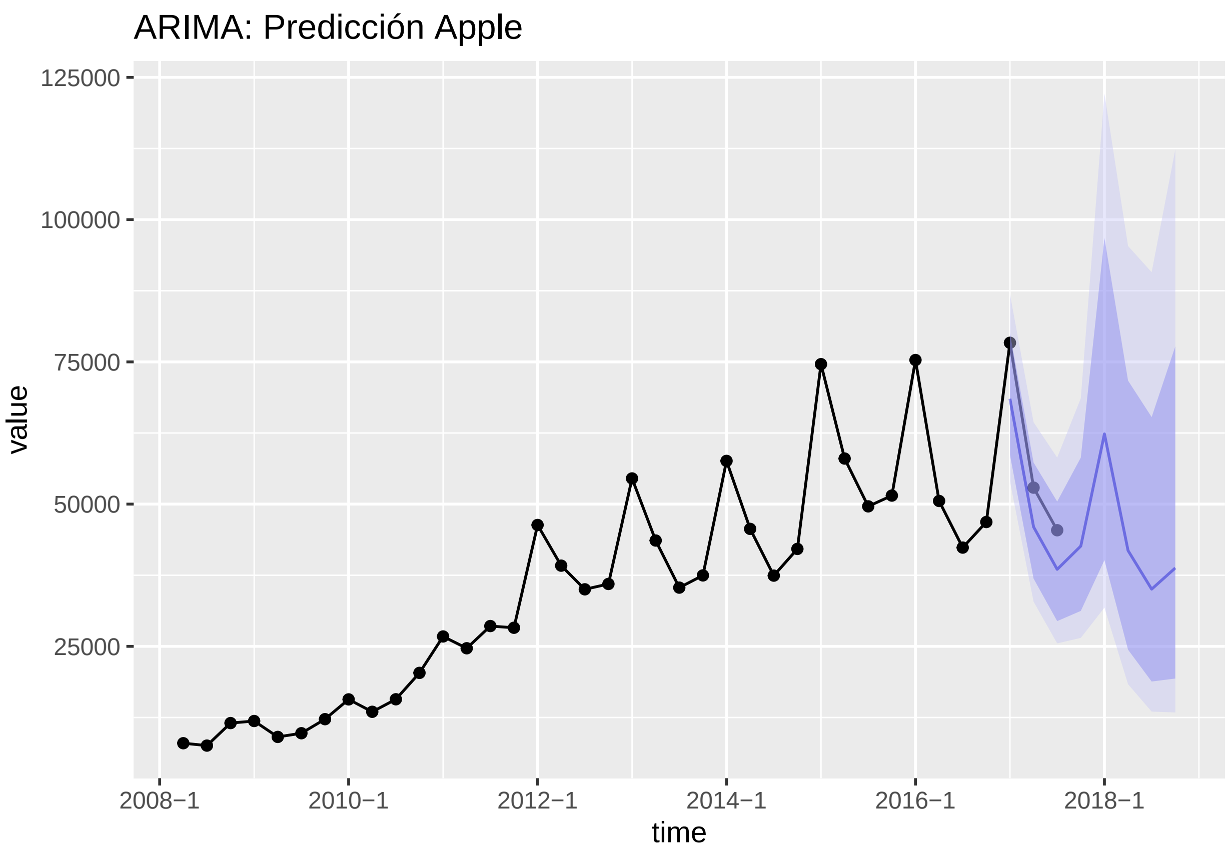
```
##
##  Box-Ljung test
##
## data:  fit1$residuals
## X-squared = 12.477, df = 9, p-value = 0.1877
```

La hipótesis nula del Test Box-Ljung implica que los residuos son ruido blanco, y como p-valor > 0.05, se acepta la hipótesis nula.

```
fventas.arima=forecast(fit1)

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


ggplot(df_new)+geom_point(aes(x=time,y=value))+geom_line(aes(x=time,y=value))+ geom_forecast(fventas.ar
```




```
fventas.arima
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
## 2017 Q1       68524.50 58656.57 80052.53 54021.77  86920.63
## 2017 Q2       45993.21 36914.17 57305.26 32857.77  64379.78
## 2017 Q3       38534.34 29436.35 50444.28 25525.07  58174.00
## 2017 Q4       42622.67 31230.74 58169.99 26490.28  68579.55
## 2018 Q1       62338.78 40156.60 96774.23 31816.09 122143.34
```

```
## 2018 Q2       41841.40 24416.21 71702.49 18358.80  95360.44
## 2018 Q3       35055.84 18821.04 65294.59 13541.05  90754.54
## 2018 Q4       38775.11 19344.31 77723.59 13387.03 112310.89
```

Comparación entre ARIMA y ETS

Si comparamos los MAPE de los modelos ETS y ARIMAS, escogeriamos ARIMAS ya que el MAPE DE ARIMAS (8.284962) es menor que el MAPE de ETS (8.677434).