Práctica3 - Predicción Ventas Apple

Manuel Carmona Cabello de Alba

25 de noviembre de 2017

Cargamos las librerias que vamos a utilizar:

```
library(quantmod) #Package to download financials historical data

## Warning: package 'quantmod' was built under R version 3.4.2

## Loading required package: xts

## Warning: package 'xts' was built under R version 3.4.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.4.2

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR

## Warning: package 'TTR' was built under R version 3.4.2

## Version 0.4-0 included new data defaults. See ?getSymbols.

library(moments) # Calculate Kurstosis and skewness
library (ggplot2) #Plot library

## Warning: package 'ggplot2' was built under R version 3.4.2

library(ggfortify) #Plot Monthplot
library(plyr) #Using ddply with data.frames

## Warning: package 'plyr' was built under R version 3.4.2

library(zoo)
require(forecast)

## Loading required package: forecast

## Warning: package 'forecast' was built under R version 3.4.2

##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:ggfortify':
##
##     gglagplot

require(xts)
library(base)
library(tseries)

## Warning: package 'tseries' was built under R version 3.4.2

library(forcats)

## Warning: package 'forcats' was built under R version 3.4.2

library(timsac)
library(TSA)

## Warning: package 'TSA' was built under R version 3.4.2

## Loading required package: leaps

## Warning: package 'leaps' was built under R version 3.4.2

## Loading required package: locfit

## Warning: package 'locfit' was built under R version 3.4.2

## locfit 1.5-9.1    2013-03-22

## Loading required package: mgcv

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:forecast':
##
##     getResponse

## This is mgcv 1.8-17. For overview type 'help("mgcv-package")'.

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:moments':
##
##     kurtosis, skewness

## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.4.2
```

```
##
## Attaching package: 'e1071'
```

```
## The following objects are masked from 'package:TSA':
##
##     kurtosis, skewness
```

```
## The following objects are masked from 'package:moments':
##
##     kurtosis, moment, skewness
```

```r
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 3.4.2
```

```r
library(reader)
```

```
## Warning: package 'reader' was built under R version 3.4.2
```

```
## Loading required package: NCmisc
```

```
## Warning: package 'NCmisc' was built under R version 3.4.2
```

```
##
## Attaching package: 'NCmisc'
```

```
## The following object is masked from 'package:nlme':
##
##     Dim
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.4.2
```

Cargamos los datos y los visualizamos

```r
datos <- read.csv("C:/Users/Manuel/Desktop/CUNEF/Prediccion/Clase05/Datos
/apple.csv", header=T, sep=",")
datos<-datos[-1,]
datos[is.na(datos)]<-0
datos$totalsales<-datos$iPhone+datos$iPad+datos$iPod+datos$Mac
datos
```

```
##     Time Period iPhone   iPad   iPod   Mac totalsales
## 2  Q1/99      2  0.000  0.000  0.000 0.827      0.827
## 3  Q2/99      3  0.000  0.000  0.000 0.905      0.905
```

```
## 4  Q3/99      4  0.000  0.000  0.000 0.772      0.772
## 5  Q4/99      5  0.000  0.000  0.000 1.377      1.377
## 6  Q1/00      6  0.000  0.000  0.000 1.043      1.043
## 7  Q2/00      7  0.000  0.000  0.000 1.016      1.016
## 8  Q3/00      8  0.000  0.000  0.000 1.122      1.122
## 9  Q4/00      9  0.000  0.000  0.000 0.659      0.659
## 10 Q1/01     10  0.000  0.000  0.000 0.751      0.751
## 11 Q2/01     11  0.000  0.000  0.000 0.827      0.827
## 12 Q3/01     12  0.000  0.000  0.000 0.850      0.850
## 13 Q4/01     13  0.000  0.000  0.125 0.746      0.871
## 14 Q1/02     14  0.000  0.000  0.062 0.813      0.875
## 15 Q2/02     15  0.000  0.000  0.054 0.808      0.862
## 16 Q3/02     16  0.000  0.000  0.140 0.734      0.874
## 17 Q4/02     17  0.000  0.000  0.219 0.743      0.962
## 18 Q1/03     18  0.000  0.000  0.080 0.711      0.791
## 19 Q2/03     19  0.000  0.000  0.304 0.771      1.075
## 20 Q3/03     20  0.000  0.000  0.336 0.787      1.123
## 21 Q4/03     21  0.000  0.000  0.733 0.829      1.562
## 22 Q1/04     22  0.000  0.000  0.807 0.749      1.556
## 23 Q2/04     23  0.000  0.000  0.860 0.876      1.736
## 24 Q3/04     24  0.000  0.000  2.016 0.836      2.852
## 25 Q4/04     25  0.000  0.000  4.580 1.046      5.626
## 26 Q1/05     26  0.000  0.000  5.311 1.070      6.381
## 27 Q2/05     27  0.000  0.000  6.155 1.182      7.337
## 28 Q3/05     28  0.000  0.000  6.451 1.236      7.687
## 29 Q4/05     29  0.000  0.000 14.043 1.254     15.297
## 30 Q1/06     30  0.000  0.000  8.526 1.112      9.638
## 31 Q2/06     31  0.000  0.000  8.111 1.327      9.438
## 32 Q3/06     32  0.000  0.000  8.729 1.610     10.339
## 33 Q4/06     33  0.000  0.000 21.066 1.606     22.672
## 34 Q1/07     34  0.000  0.000 10.549 1.517     12.066
## 35 Q2/07     35  0.270  0.000  9.815 1.764     11.849
## 36 Q3/07     36  1.119  0.000 10.200 2.164     13.483
## 37 Q4/07     37  2.315  0.000 22.121 2.319     26.755
## 38 Q1/08     38  1.703  0.000 10.644 2.289     14.636
## 39 Q2/08     39  0.717  0.000 11.011 2.496     14.224
## 40 Q3/08     40  6.892  0.000 11.052 2.611     20.555
## 41 Q4/08     41  4.363  0.000 22.727 2.524     29.614
## 42 Q1/09     42  3.793  0.000 11.013 2.216     17.022
## 43 Q2/09     43  5.208  0.000 10.215 2.603     18.026
## 44 Q3/09     44  7.367  0.000 10.177 3.053     20.597
## 45 Q4/09     45  8.737  0.000 20.970 3.362     33.069
## 46 Q1/10     46  8.752  0.000 10.885 2.943     22.580
## 47 Q2/10     47  8.398  3.270  9.406 3.472     24.546
## 48 Q3/10     48 14.102  4.188  9.051 3.885     31.226
## 49 Q4/10     49 16.235  7.331 19.446 4.134     47.146
## 50 Q1/11     50 18.647  4.694  9.017 3.760     36.118
## 51 Q2/11     51 20.338  9.246  7.535 3.947     41.066
## 52 Q3/11     52 17.073 11.123  6.622 4.894     39.712
## 53 Q4/11     53 37.044 15.434 15.397 5.198     73.073
```

```
## 54 Q1/12     54 35.064 11.798  7.673 4.017     58.552
## 55 Q2/12     55 26.028 17.042  6.751 4.020     53.841
## 56 Q3/12     56 26.910 14.036  5.344 4.923     51.213
## 57 Q4/12     57 47.789 22.860 12.679 4.061     87.389
## 58 Q1/13     58 37.430 19.477  5.633 3.952     66.492
## 59 Q2/13     59 31.241 14.617  4.569 3.754     54.181
## 60 Q3/13     60 33.797 14.079  3.498 4.574     55.948
## 61 Q4/13     61 51.025 26.035  6.049 4.837     87.946
## 62 Q1/14     62 43.719 16.350  2.761 4.136     66.966
## 63 Q2/14     63 35.203 13.276  2.926 4.413     55.818
## 64 Q3/14     64 39.272 12.316  2.641 5.520     59.749
## 65 Q4/14     65 74.468 21.419  0.000 5.519    101.406
## 66 Q1/15     66 61.170 12.623  0.000 4.563     78.356
## 67 Q2/15     67 47.534 10.931  0.000 4.796     63.261
## 68 Q3/15     68 48.050  9.880  0.000 5.710     63.640
## 69 Q4/15     69 74.780 16.120  0.000 5.310     96.210
## 70 Q1/16     70 51.200 10.250  0.000 4.030     65.480
```

Cambiamos el formato para poder trabajar con los datos como serie temporal

```r
library(zoo)
datos$Time<-as.Date(as.yearqtr(datos$Time, format="Q%q/%y"))
str(datos$Time)
```

```
##  Date[1:69], format: "1999-01-01" "1999-04-01" "1999-07-01" "1999-10-0
1" "2000-01-01" ...
```

```r
xVentas=xts((datos$totalsales),order.by=datos$Time) #La y mayuscula porqu
e esta el a?o completo

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

#Transform to zoo data (forecast package)
zVentas=as.zoo(xVentas$xVentas.Close)
```

Pasamos el nombre a Ventas totales

```r
names(zVentas)="Ventas_totales"
zVentas
```

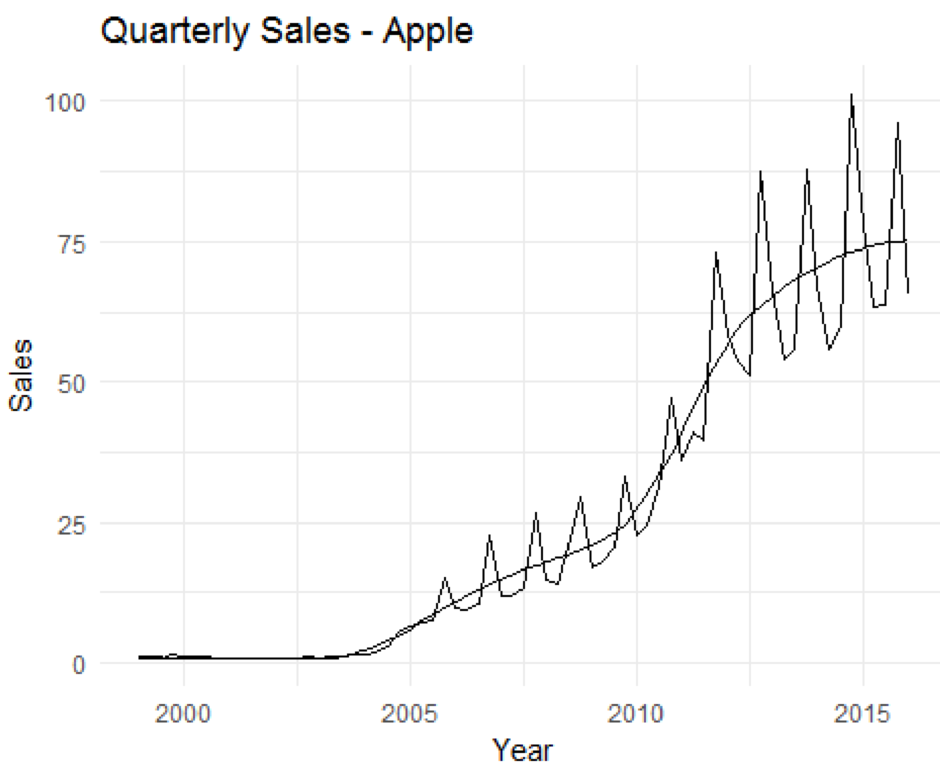
```
##         Ventas_totales
## 1999 Q1          0.827
## 1999 Q2          0.905
## 1999 Q3          0.772
## 1999 Q4          1.377
## 2000 Q1          1.043
## 2000 Q2          1.016
## 2000 Q3          1.122
## 2000 Q4          0.659
## 2001 Q1          0.751
```

```
## 2001 Q2           0.827
## 2001 Q3           0.850
## 2001 Q4           0.871
## 2002 Q1           0.875
## 2002 Q2           0.862
## 2002 Q3           0.874
## 2002 Q4           0.962
## 2003 Q1           0.791
## 2003 Q2           1.075
## 2003 Q3           1.123
## 2003 Q4           1.562
## 2004 Q1           1.556
## 2004 Q2           1.736
## 2004 Q3           2.852
## 2004 Q4           5.626
## 2005 Q1           6.381
## 2005 Q2           7.337
## 2005 Q3           7.687
## 2005 Q4          15.297
## 2006 Q1           9.638
## 2006 Q2           9.438
## 2006 Q3          10.339
## 2006 Q4          22.672
## 2007 Q1          12.066
## 2007 Q2          11.849
## 2007 Q3          13.483
## 2007 Q4          26.755
## 2008 Q1          14.636
## 2008 Q2          14.224
## 2008 Q3          20.555
## 2008 Q4          29.614
## 2009 Q1          17.022
## 2009 Q2          18.026
## 2009 Q3          20.597
## 2009 Q4          33.069
## 2010 Q1          22.580
## 2010 Q2          24.546
## 2010 Q3          31.226
## 2010 Q4          47.146
## 2011 Q1          36.118
## 2011 Q2          41.066
## 2011 Q3          39.712
## 2011 Q4          73.073
## 2012 Q1          58.552
## 2012 Q2          53.841
## 2012 Q3          51.213
## 2012 Q4          87.389
## 2013 Q1          66.492
## 2013 Q2          54.181
## 2013 Q3          55.948
```

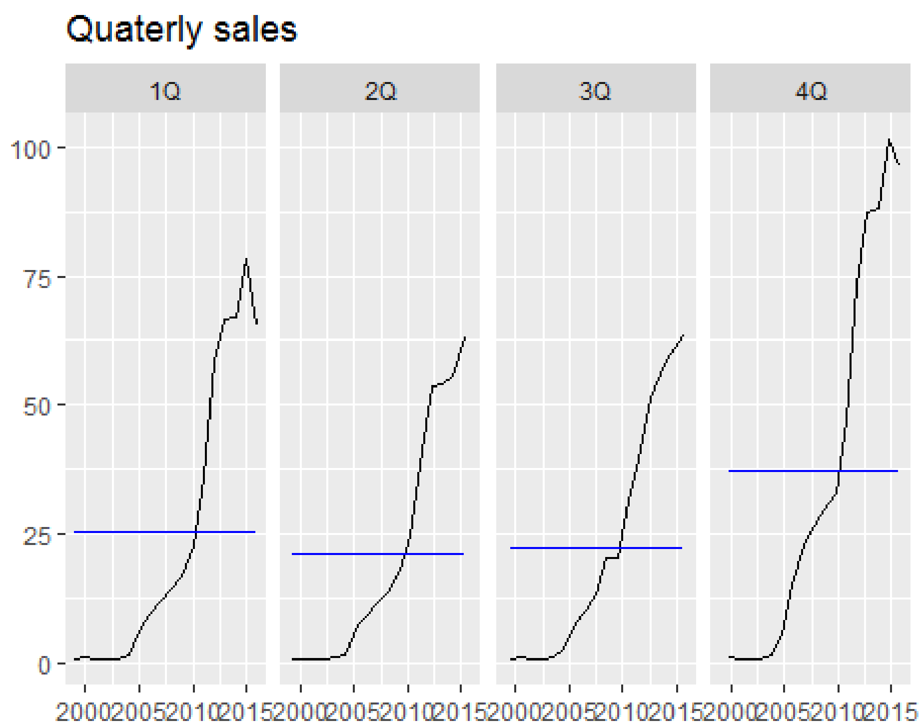
```
## 2013 Q4         87.946
## 2014 Q1         66.966
## 2014 Q2         55.818
## 2014 Q3         59.749
## 2014 Q4        101.406
## 2015 Q1         78.356
## 2015 Q2         63.261
## 2015 Q3         63.640
## 2015 Q4         96.210
## 2016 Q1         65.480
```

Vamos a ver la serie

```
apple.plot <- autoplot(as.ts(zVentas), ts.colour="dodgerblue3")+
  labs(y="Sales",x="Year")+
  ggtitle("Quarterly Sales - Apple")+
  theme_minimal()+
  geom_smooth(aes(y=(zVentas)),span=0.35,se=F,size=0.4,col="black")
apple.plot

## `geom_smooth()` using method = 'loess'
```

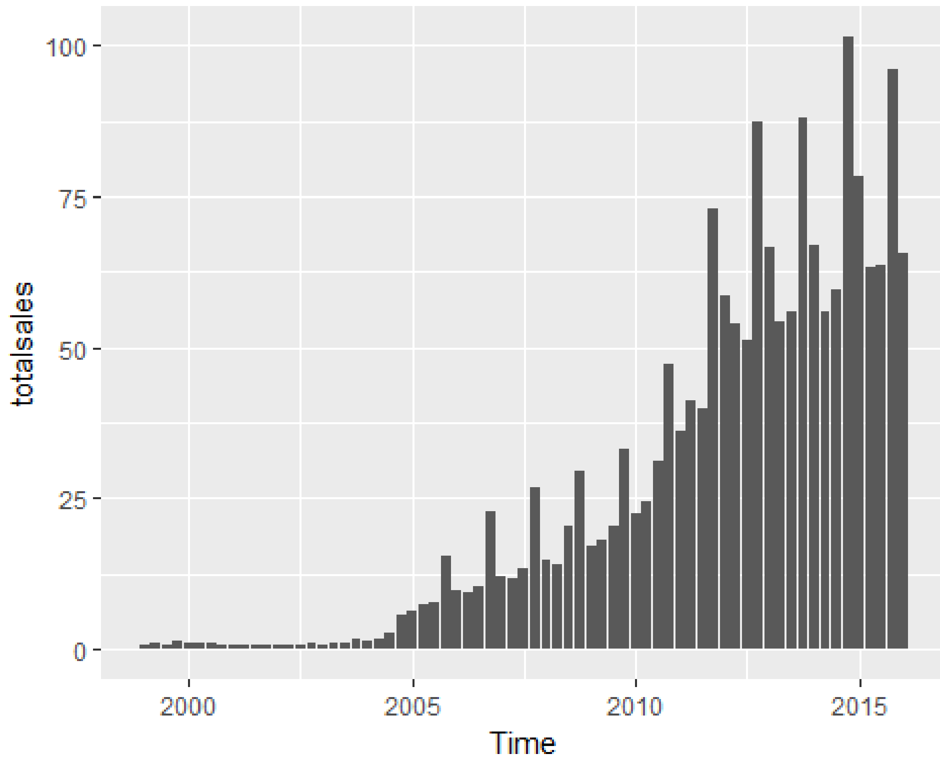



```
ggfreqplot(as.ts(zVentas),freq=4,nrow=1,facet.labeller=c("1Q","2Q","3Q","
4Q"))+ggtitle("Quaterly sales")
```



Podemos obsevar que el cuarto trimestre del a?o es el que lidera las ventas, coincidiendo con la politica de lanzamiento de productos de Apple.

```
library(ggplot2)
ggplot(datos, aes(x=Time, y=totalsales)) + geom_col()
```



En este gr?fico corroboramos la anterior afirmaci?n

Descripcion de la serie: Tendencia: tiene tendencia positiva, no queda claro si es lineal o exponencial (componente aditivo o multiplicativo) y no tiene ciclo. Estacionalidad: Claramente existe componente estacional multiplicativo, como puede observarse en el primer trimestre de cada a?o. La serie es no estacionaria en varianza y EN MEDIA????

Modelos ETS

Los modelos ETS solo asumen datos no estacionarios. As?, no tenemos que transformar la serie para estimar los modelos. Dentro de los modelos ETS, los que incluyen el componente estacional son los llamados Holt-Winters. Vamos a estimar 4 modelos Holt-Winter y seleccionaremos uno en funcion de los criterios AIC, BIC y HQ.

Periodo entrenamiento y periodo test

Vamos a coger un periodo de entrenamiento y uno de test para probar el modelo

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

Modelo Holt-Winter (A,M)

```r
fitHWAM <- holt(oVentas, exponential=FALSE, damped=FALSE, seasonal="multi
plicative")
fitHWAM$model
```

```
## Holt's method
##
## Call:
##  holt(y = oVentas, damped = FALSE, exponential = FALSE, seasonal = "mu
ltiplicative")
##
##   Smoothing parameters:
##     alpha = 0.1959
##     beta  = 0.0258
##
##   Initial states:
##     l = 0.4765
##     b = -0.0383
##
##   sigma:  9.1891
##
##      AIC     AICc      BIC
## 569.6772 570.6941 580.5491
```

Modelo Holt-Winter (Ad,M)

```r
fitHWAdM <- holt(oVentas,exponential=FALSE, damped=TRUE, seasonal="multip
licative")
fitHWAdM$model
```

```
## Damped Holt's method
##
## Call:
##  holt(y = oVentas, damped = TRUE, exponential = FALSE, seasonal = "mul
tiplicative")
##
##   Smoothing parameters:
##     alpha = 0.1171
##     beta  = 0.117
##     phi   = 0.8634
##
##   Initial states:
##     l = 1.212
##     b = 0.0469
##
##   sigma:  9.2475
##
##      AIC     AICc      BIC
## 572.5013 573.9496 585.5476
```

Modelo Holt-Winter (M,M)

```
fitHWMM <- holt(oVentas,exponential=TRUE, damped=FALSE, seasonal="multipl
icative")
fitHWMM$model
```

```
## Holt's method with exponential trend
##
## Call:
##  holt(y = oVentas, damped = FALSE, exponential = TRUE, seasonal = "mul
tiplicative")
##
##   Smoothing parameters:
##     alpha = 0.2217
##     beta  = 1e-04
##
##   Initial states:
##     l = 1.8787
##     b = 1.0471
##
##   sigma:  0.5055
##
##      AIC     AICc      BIC
## 472.4415 473.4585 483.3134
```

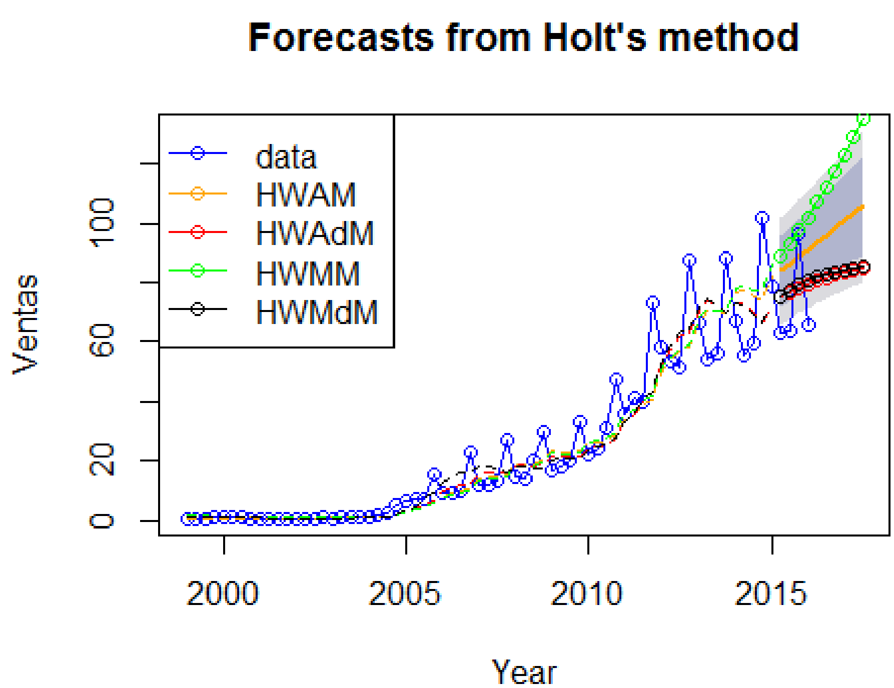
Modelo Holt-Winter(Md,M)

```
fitHWMdM <- holt(oVentas,exponential=TRUE, damped=TRUE, seasonal="multipl
icative")
fitHWMdM$model
```

```
## Damped Holt's method with exponential trend
##
## Call:
##  holt(y = oVentas, damped = TRUE, exponential = TRUE, seasonal = "mult
iplicative")
##
##   Smoothing parameters:
##     alpha = 0.1192
##     beta  = 0.1192
##     phi   = 0.8
##
##   Initial states:
##     l = 1.1602
##     b = 1.2634
##
##   sigma:  0.412
##
##      AIC     AICc      BIC
## 442.3112 443.7595 455.3576
```

Como podemos observar, según los criterios de informacion los modelos aditivos funcionaron peor. Vamos a graficar los modelos para comprobarlo.

```
plot(fitHWAM,ylab="Ventas",
     plot.conf=FALSE, type="o", fcol="orange", xlab="Year")

## Warning in plot.window(xlim, ylim, log, ...): "plot.conf" is not a
## graphical parameter

## Warning in title(main = main, xlab = xlab, ylab = ylab, ...): "plot.co
nf"
## is not a graphical parameter

## Warning in axis(1, ...): "plot.conf" is not a graphical parameter

## Warning in axis(2, ...): "plot.conf" is not a graphical parameter

## Warning in box(...): "plot.conf" is not a graphical parameter

lines(window(zVentas),type="o",col="blue")
lines(fitted(fitHWAM), col="orange", lty=2)
lines(fitted(fitHWAdM), col="red", lty=2)
lines(fitted(fitHWMM), col="green", lty=2)
lines(fitted(fitHWMdM), col="black", lty=2)
lines(fitHWAdM$mean, type="o", col="red")
lines(fitHWMM$mean, type="o", col="green")
lines(fitHWMdM$mean, type="o", col="black")
legend("topleft",lty=1, pch=1, col=c("blue","orange", "red", "green", "bl
ack"),
       c("data","HWAM", "HWAdM","HWMM", "HWMdM"))
```



Ahora vamos a calcular el modelo de forma automatica:

```
etsfit<-ets(oVentas)
```

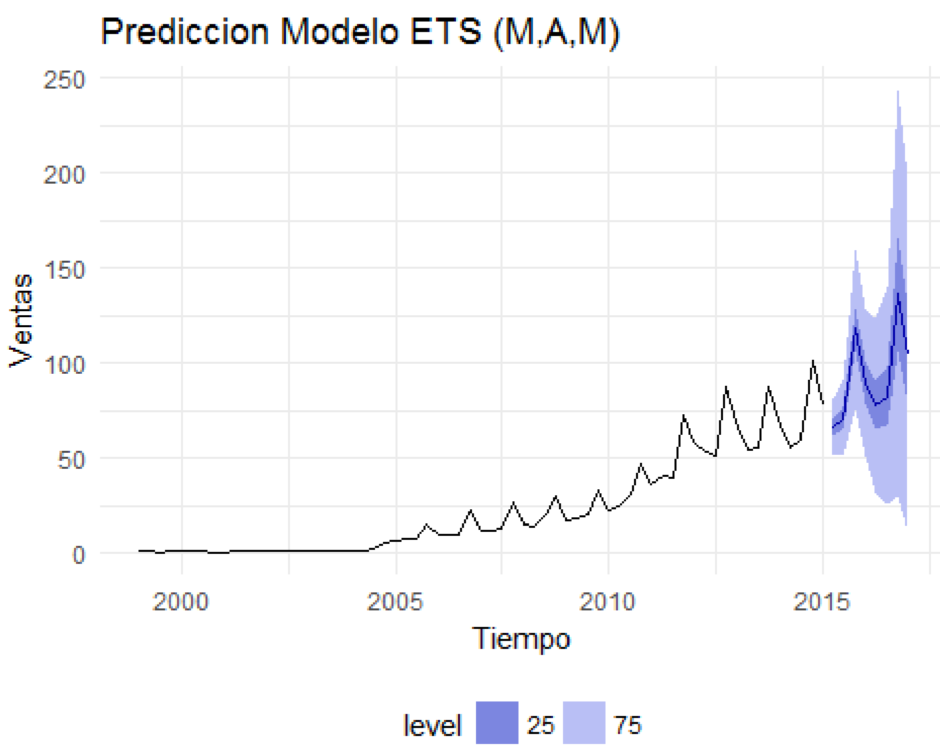
Vemos los coeficientes, la puntuaci?n en los criterios de informaci?n y las predicciones.

```
coef(etsfit) #esto returns all fitted parameters.

##      alpha       beta      gamma          l          b         s0
## 0.52328766 0.22725586 0.47671233 0.77769849 0.03445515 1.23396977
##         s1         s2
## 0.82714353 0.97243785

#forecast model
#forecast model
fventas.ets=forecast(etsfit, level=(c(75,25)))
#Results
summary(fventas.ets)

##
## Forecast method: ETS(M,A,M)
##
## Model Information:
## ETS(M,A,M)
##
## Call:
##  ets(y = oVentas)
##
##   Smoothing parameters:
##     alpha = 0.5233
##     beta  = 0.2273
##     gamma = 0.4767
##
##   Initial states:
##     l = 0.7777
##     b = 0.0345
##     s=1.234 0.8271 0.9724 0.9664
##
##   sigma:  0.197
##
##      AIC     AICc      BIC
## 353.4342 356.7070 373.0037
##
## Error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.08502138 3.165721 2.054496 0.2131736 14.01946 0.4254655
##                     ACF1
## Training set -0.08122899
##
## Forecasts:
```

```
##         Point Forecast     Lo 25     Hi 25    Lo 75     Hi 75
## 2015 Q2       66.57793  62.39823  70.75764 51.48839  81.66747
## 2015 Q3       70.95723  65.41958  76.49488 50.96526  90.94920
## 2015 Q4      117.89441 106.37482 129.41400 76.30646 159.48236
## 2016 Q1       90.02745  79.16621 100.88869 50.81628 129.23862
## 2016 Q2       77.89727  65.10224  90.69231 31.70474 124.08980
## 2016 Q3       82.66843  67.00899  98.32786 26.13484 139.20201
## 2016 Q4      136.80562 107.10449 166.50676 29.57882 244.03243
## 2017 Q1      104.07810  78.36769 129.78852 11.25857 196.89764
```

Obtenemos un modelo con un AIC mucho menor que los anteriores. El modelo es de tipo M,A,M, es decir, de error multiplicativo, tendencia aditiva y componente estacional multiplicativo.

```
#plot(fventas.ets)
#lines(window(zVentas),type="o")

plotets <- autoplot(fventas.ets)+
  labs(y="Ventas",x="Tiempo")+
  ggtitle("Prediccion Modelo ETS (M,A,M)")+
  theme_minimal()+
  scale_colour_economist()+
  theme(legend.position = "bottom")
plotets
```



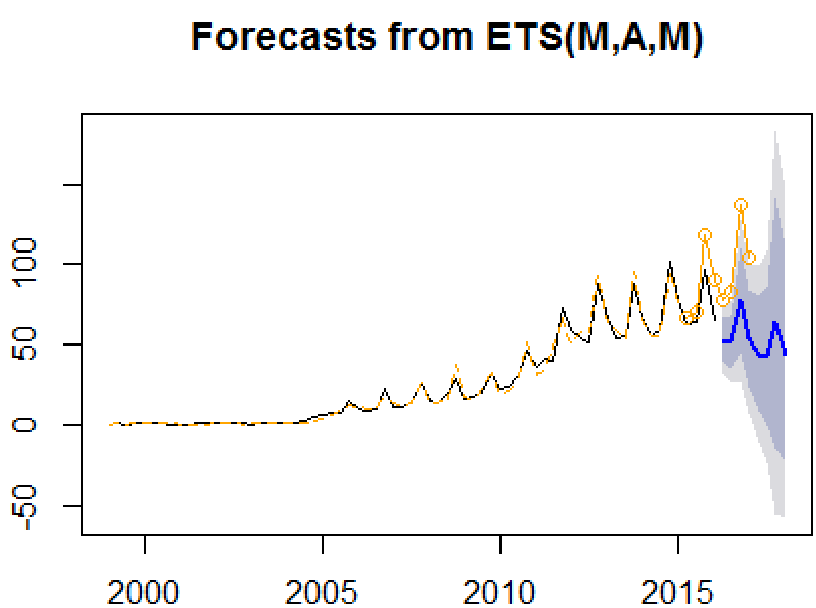
Como vemos, se ajusta mucho mejor que los modelos anteriormente calculados.

Ahora vamos a calcular los trimestres y los comparas con lo real, vemos que en el 4T es donde mas me alejo.

```
comparativa<-as.data.frame(matrix(c(fventas.ets$mean[1:cOmit],zVentas[(nObs-cOmit+1):nObs]),ncol=2))
names(comparativa)<-c("predicted","real")
comparativa

##   predicted   real
## 1  66.57793 63.261
## 2  70.95723 63.640
## 3 117.89441 96.210
## 4  90.02745 65.480
```

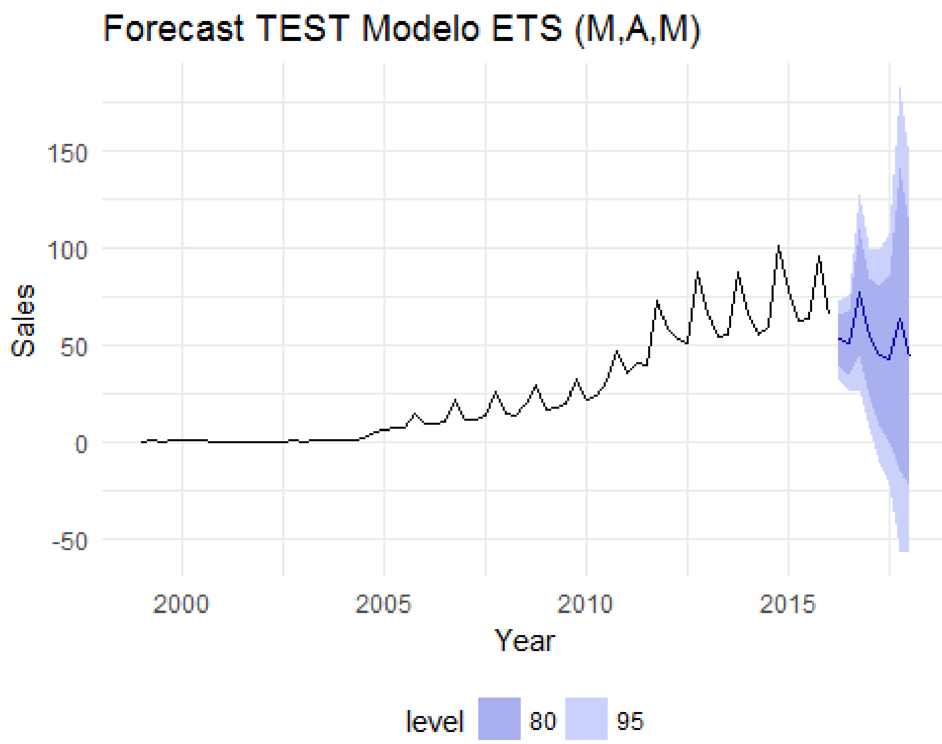
Ahora predecimos con todo para ver en el plot que es lo que hace el modelo y demostrar el caracter cortoplacista.

```
estfit2=ets(zVentas, damped=FALSE)
f.estfit2=forecast(estfit2)
plot(f.estfit2)
lines(fitted(fventas.ets), col="orange", lty=2)
lines(fventas.ets$mean, type="o", col="orange")
```




```
plotets <- autoplot(f.estfit2)+
  labs(y="Sales",x="Year")+
  ggtitle("Forecast TEST Modelo ETS (M,A,M)")+
  theme_minimal()+
  scale_colour_economist()+
```

```
  theme(legend.position = "bottom")
plotets
```



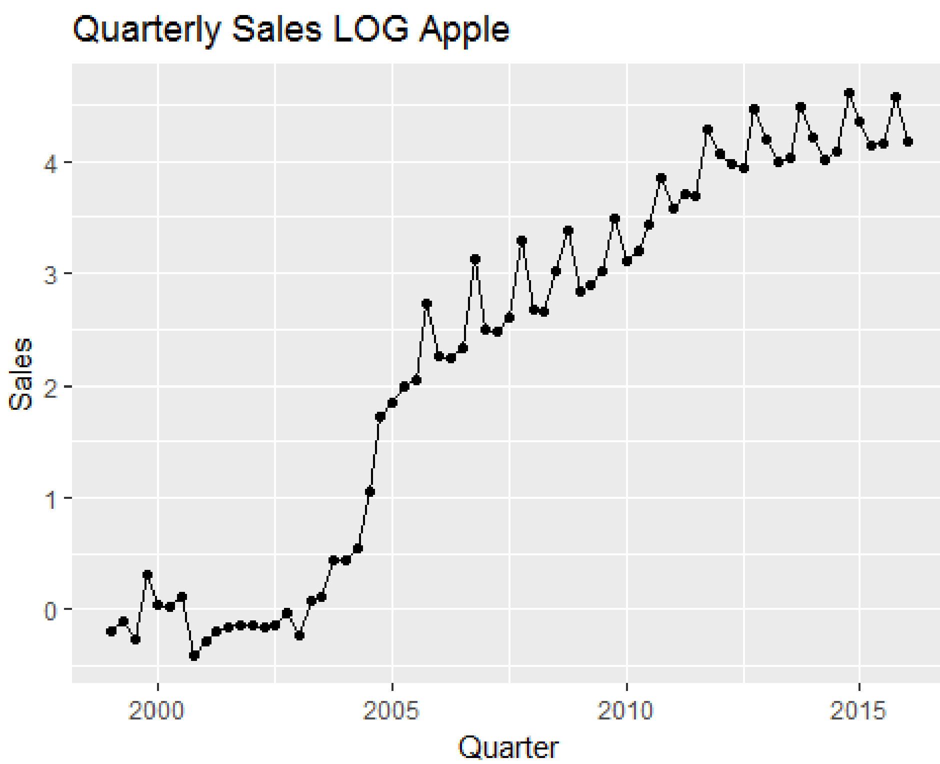
Modelos ARIMA

Como sabemos la serie es no estacionaria. Para hacerla estacionaria y poder utilizar los modelos ARIMA hemos de tranformarla en estacionaria.

Análisis de la serie

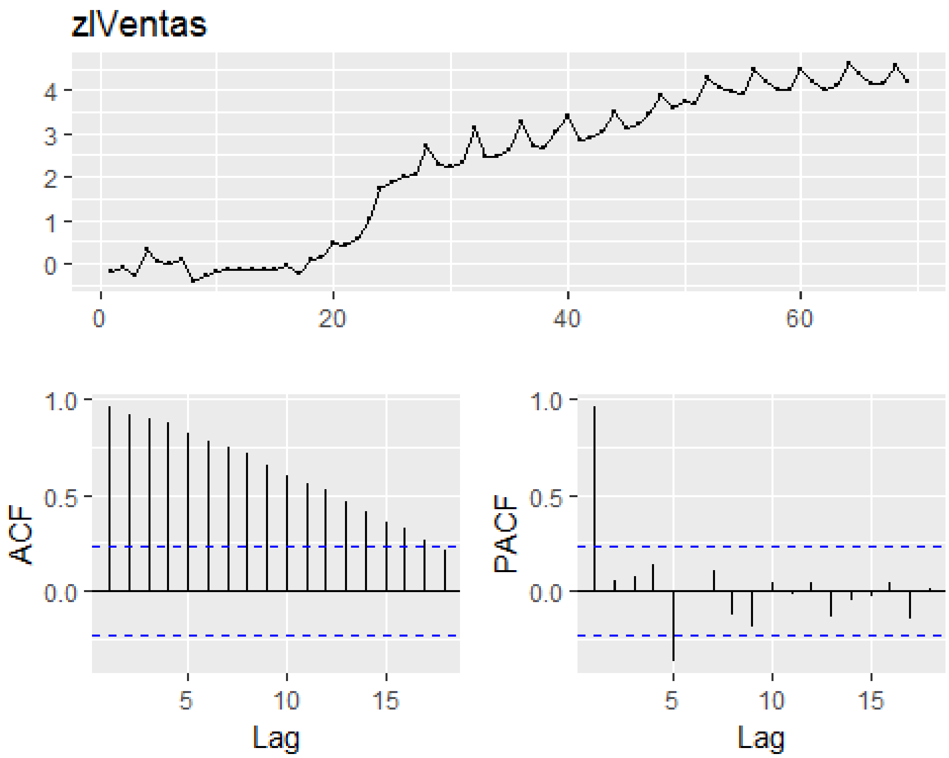
El primer paso es hacerla estacionaria en varianza:

```
zlVentas=log(zVentas)
df_newl <- data.frame(value = as.vector(zlVentas),
                      time = time(zlVentas))
ggplot(df_newl)+geom_point(aes(x=time,y=value))+geom_line(aes(x=time,y=va
lue))+ylab("Sales")+ggtitle("Quarterly Sales LOG Apple")+xlab("Quarter")

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



Ahora como vemos la serie es estacionaria en varianza tras aplicar la transformacion mas habitual, la logaritmica (logaritmo neperiano).

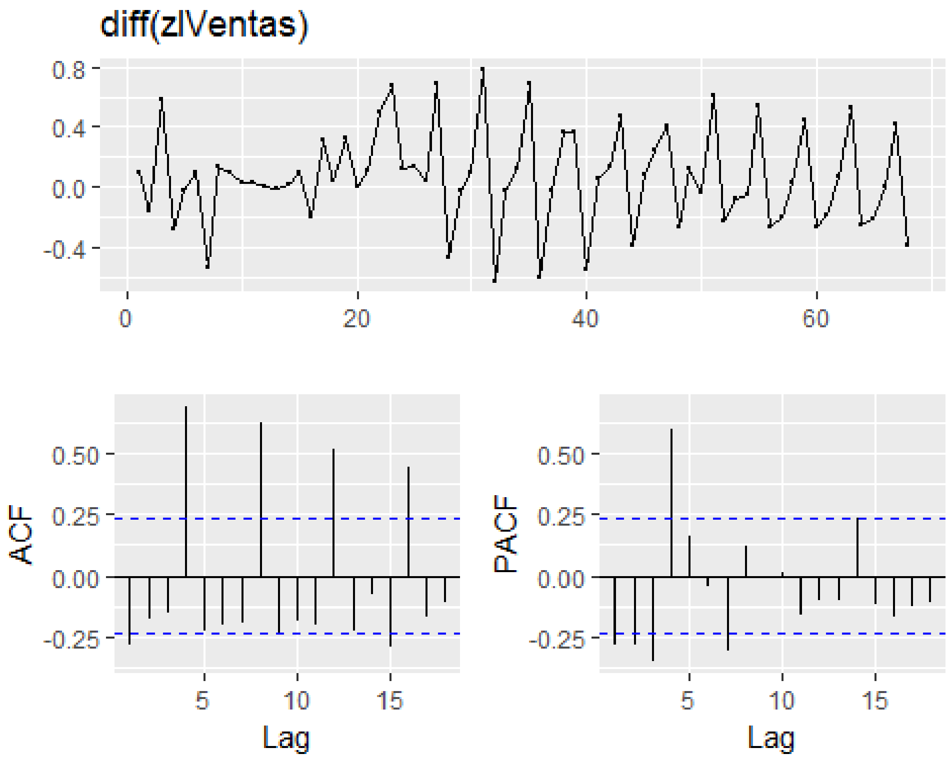
```
ggtsdisplay(zlVentas)
```



Claramente la serie es no estacionaria. Los valores de hoy dependen de los de hace 15 periodos.

Para transformar la serie a estacionaria en media, tenemos que aplicar el operador diferencias. Vamos a realizar la diferencia de primer orden, que consiste en restar a la serie original la misma serie pero retardar un periodo:

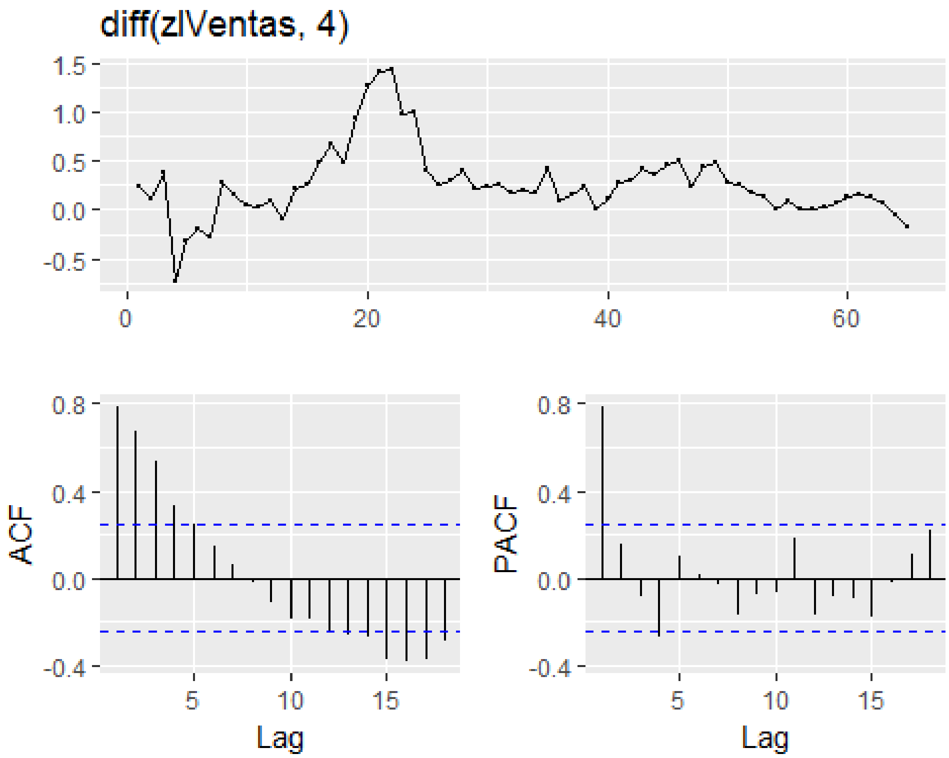
```
ggtsdisplay(diff(zlVentas)) #tasa intertrimestral, aunque parece estacion
aria, si vemos la correlacion estacional en el grafico de abajo a la izq,
```



Con la diferencia obtenemos la tasa intertrimestral. Aunque parece estacionaria, si vemos la correlacion estacional en el grafico de abajo a la izquierda vemos que en la parte esacional no cae (barritas para arriba parecidas), por eso no es estacionaria.

Vamos a hacer la diferencia de orden 4 (anual)

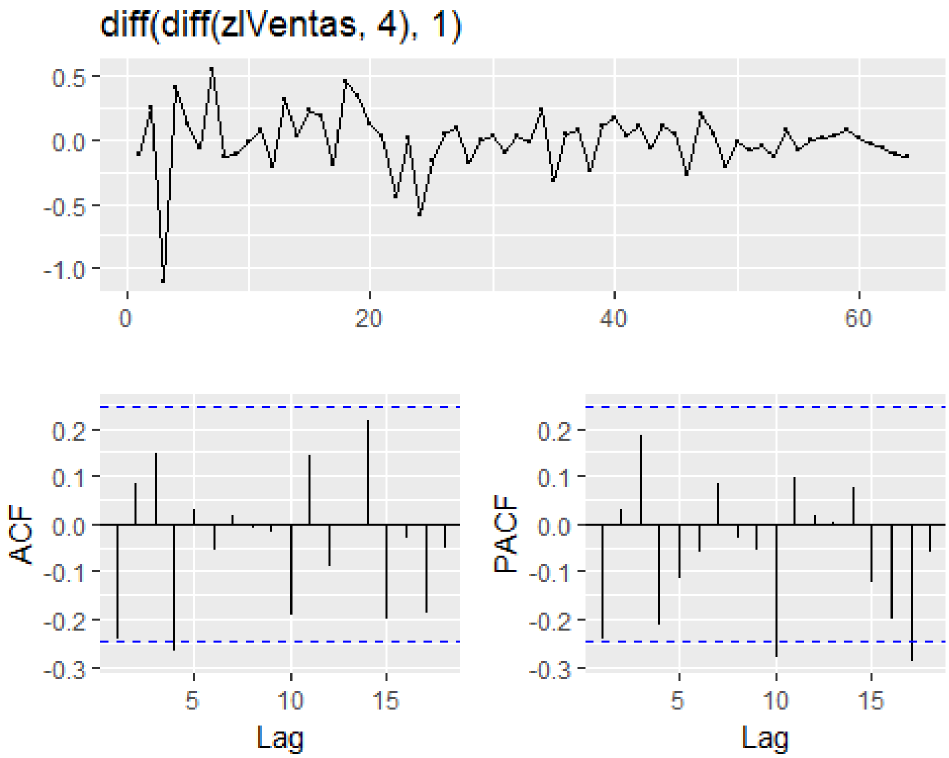
```
ggtsdisplay(diff(zlVentas,4))
```



Aquí si observamos un ciclo y se ve que es estacionario.

Ahora haremos la segunda diferencia para ver como crece la tasa de variación anual a modo ilustrativo

```
ggtsdisplay(diff(diff(zlVentas,4),1))
```



Para ver como evoluciona la tasa de variación annual, hago una diferencia.

Estimacion del modelo ARIMA

Estimacion del modelo

Estimamos el modelo ARIMA con la funcion autoarima y marcando lambda=0 ya que queremos transformar la serie a estacionaria con la transformacion mas habitual,la logaritmica (logaritmo neperiano), que es un caso particular de la transformacion Box-Cox cuando el parametro lambda es cero.

```
fitARIMA=auto.arima(oVentas,lambda=0)
summary(fitARIMA)
```

```
## Series: oVentas
## ARIMA(1,0,0)(1,1,0)[4]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1     sar1
##       0.9145  -0.3764
## s.e.  0.0478   0.1541
##
## sigma^2 estimated as 0.05184:  log likelihood=3.75
## AIC=-1.51   AICc=-1.09   BIC=4.83
```

```
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE     MASE
## Training set 0.2063695 3.880316 2.350587 0.546513 14.85802 0.486783
##                    ACF1
## Training set -0.1767947
```

Obtenemos un modelo SARIMA, ARIMA con componente estacional ARIMA(1,0,0)(1,1,0)[4]. El componente principal tiene componente autorregresivo de 1 retardo. El componente estacional igual, y adem?s requiere de una diferencia para hacer la serie estacionaria en media. Ninguno de los componentes depende de los errores que tuvieron lugar anteriormente. El [4] nos indica anualidad (4 trimestres) tal y como habiamos observado anteriormente aplicando las diferencias de forma manual para entender la serie.

Obtenemos un AIC mucho menor al que obteniamos con los modelos ETS. Esto tiene sentido puesto que los modelos ETS tienen un caracter cortoplacista muy marcado.

Vamos a probar estimandolo sin las funciones de aproximacion y stepwise de auto.arima, de forma que se exploren todos los modelos posibles.

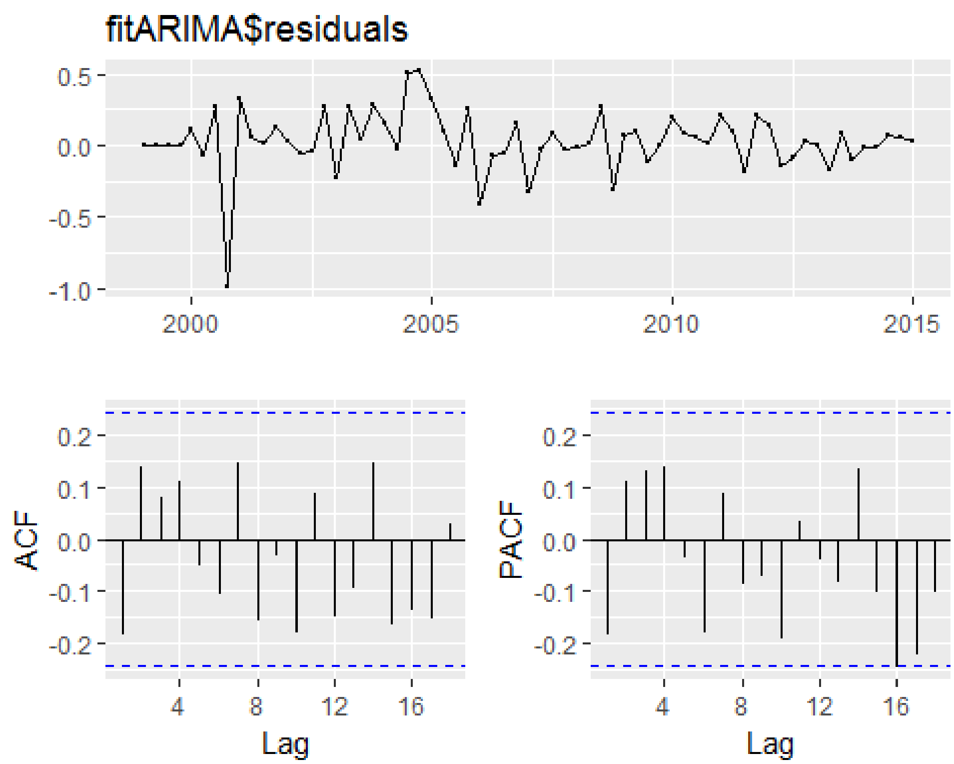
```
fitARIMA2=auto.arima(oVentas,lambda=0, approximation = F, stepwise = F)
summary(fitARIMA2)
```

```
## Series: oVentas
## ARIMA(1,0,3)(0,1,1)[4] with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1     ma1     ma2     ma3     sma1   drift
##       0.5535  0.1087  0.4203  0.7347  -0.1648  0.0741
## s.e.  0.1738  0.1573  0.1242  0.2120   0.2318  0.0254
##
## sigma^2 estimated as 0.0413:  log likelihood=11.26
## AIC=-8.52   AICc=-6.4   BIC=6.26
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.6010531 3.333167 2.089742 -2.192735 13.37792 0.4327646
##                     ACF1
## Training set -0.01694753
```

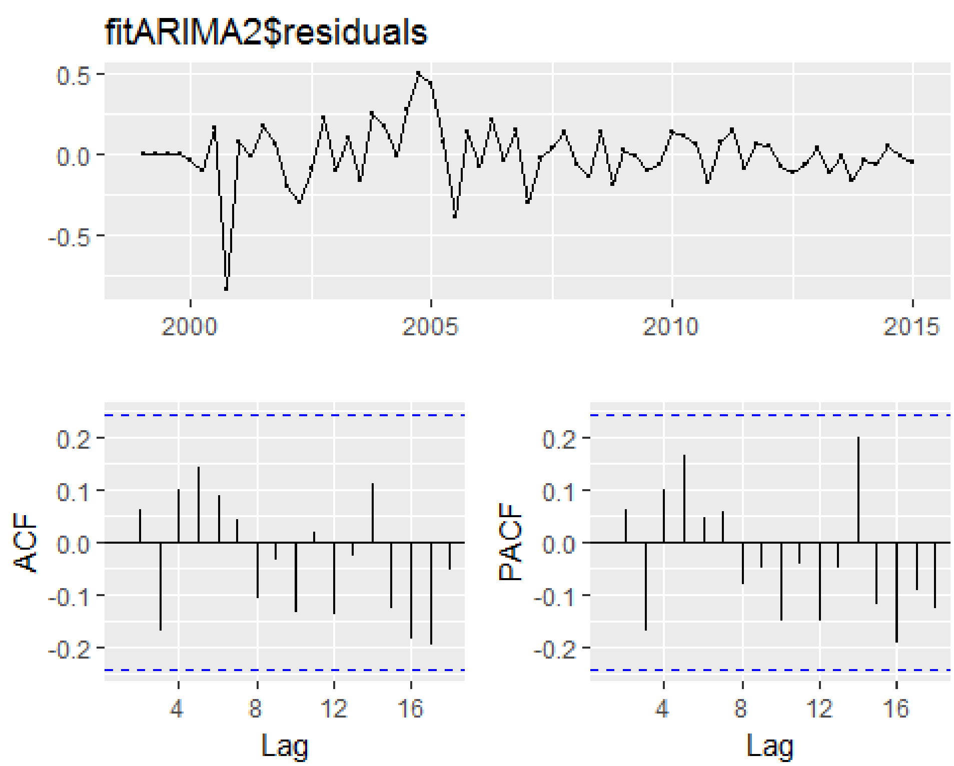
El modelo que se obtiene tiene mejor AIC pero peor BIC. Ante la duda, conviene comprobar como predicen ambos modelos.

Analisis de resiudos

```
ggtsdisplay(fitARIMA$residuals)
```




```
ggtsdisplay(fitARIMA2$residuals)
```



Prestando atención a las bandas vemos que los residuos se pueden calificar de ruido blanco. Aun as?, vamos a realizar un test para comprobar que, efectivamente, los residuos son ruido blanco.

Realizaremos el test de Box-Ljung, que consiste en contrastar si los retardos de la acf son cero a la vez:

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

##
##  Box-Ljung test
##
## data:  fitARIMA$residuals
## X-squared = 5.0026, df = 1, p-value = 0.02531

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

##
##  Box-Ljung test
##
## data:  fitARIMA$residuals
## X-squared = 9.5764, df = 5, p-value = 0.08817

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

##
##  Box-Ljung test
##
## data:  fitARIMA$residuals
## X-squared = 14.735, df = 9, p-value = 0.09847

Box.test(fitARIMA2$residuals,lag=4, fitdf=3, type="Lj")

##
##  Box-Ljung test
##
## data:  fitARIMA2$residuals
## X-squared = 2.9448, df = 1, p-value = 0.08615

Box.test(fitARIMA2$residuals,lag=8, fitdf=3, type="Lj")

##
##  Box-Ljung test
##
## data:  fitARIMA2$residuals
## X-squared = 5.9693, df = 5, p-value = 0.3092

Box.test(fitARIMA2$residuals,lag=12, fitdf=3, type="Lj")

##
##  Box-Ljung test
##
```

```
## data:  fitARIMA2$residuals
## X-squared = 9.0738, df = 9, p-value = 0.4305
```

H0: ruido blanco, H1:no es ruido blanco. En ambos modelos podemos considerar que los residuos son ruido blanco. fitdf es el número de parámetros que hemos estimado para obtener los residuos.

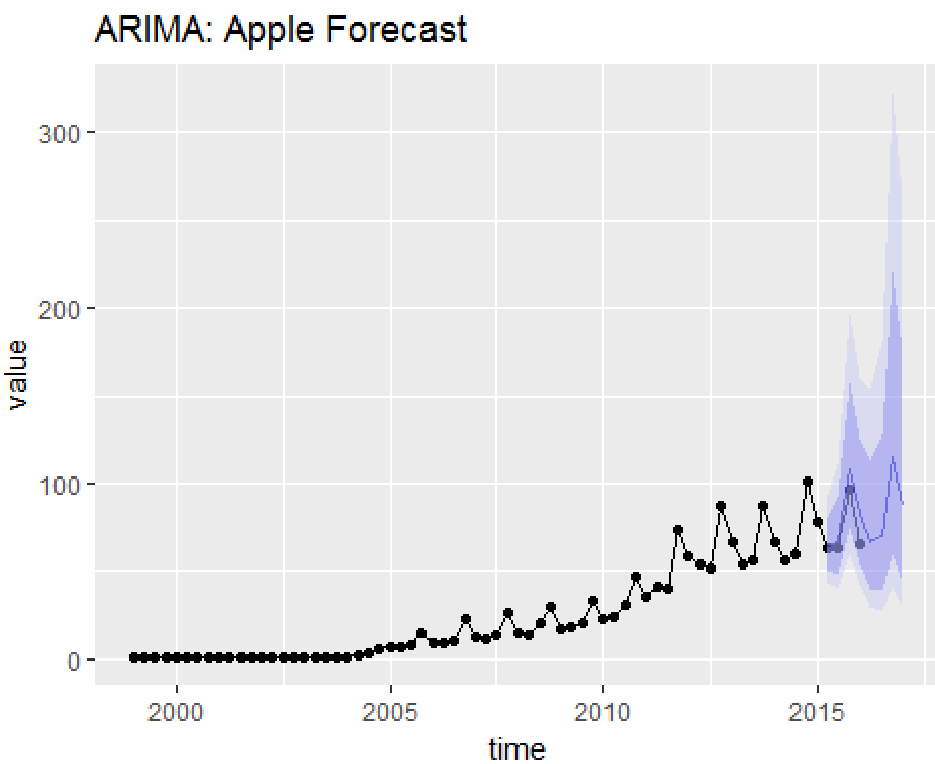
Vamos a ver como predicen ambos modelos.

Primero estimamos las predicciones

```
fventas.arima=forecast(fitARIMA)
fventas.arima=forecast(fitARIMA, level=c(70,90)) #podemos cambiar el inte
rvalo


fventas.arima2=forecast(fitARIMA2, level=c(70,90))
```

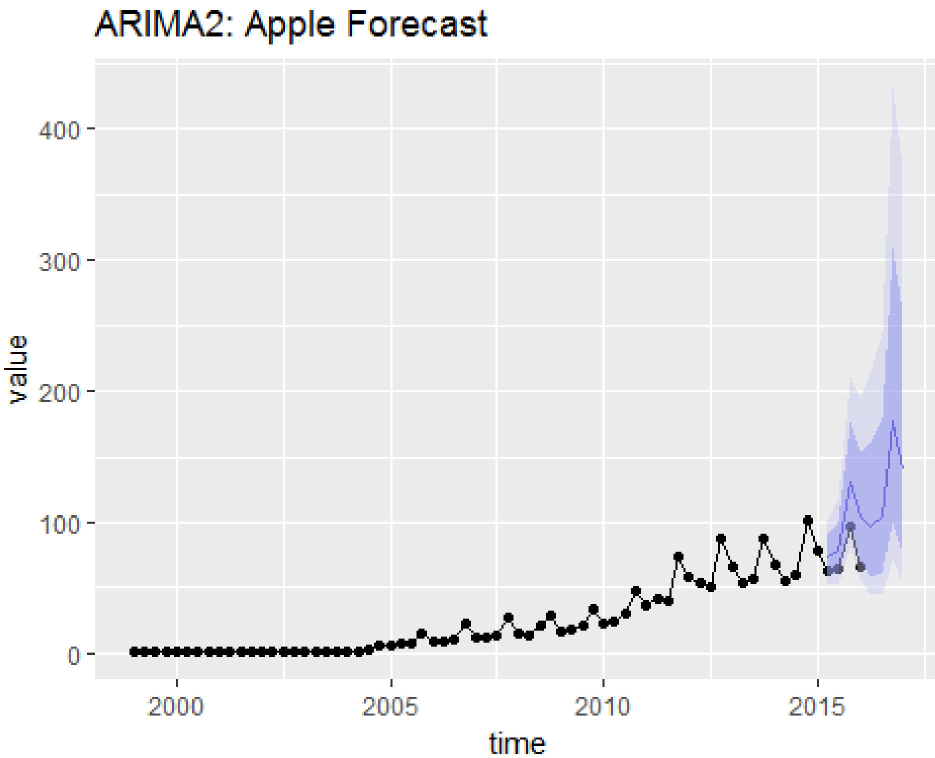
Creamos un dataframe con la serie (zVentas) para poder representar el comportamiento del modelo y Graficamos:

```
df_newl<- data.frame(value = as.vector(zVentas),
                     time = time(zVentas))




ggplot(df_newl)+geom_point(aes(x=time,y=value))+geom_line(aes(x=time,y=va
lue))+ geom_forecast(fventas.arima,alpha=0.4)+ggtitle("ARIMA: Apple Forec
ast")

## Warning in geom_forecast(fventas.arima, alpha = 0.4): Use autolayer in
stead
## of geom_forecast to add a forecast layer to your ggplot object.

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



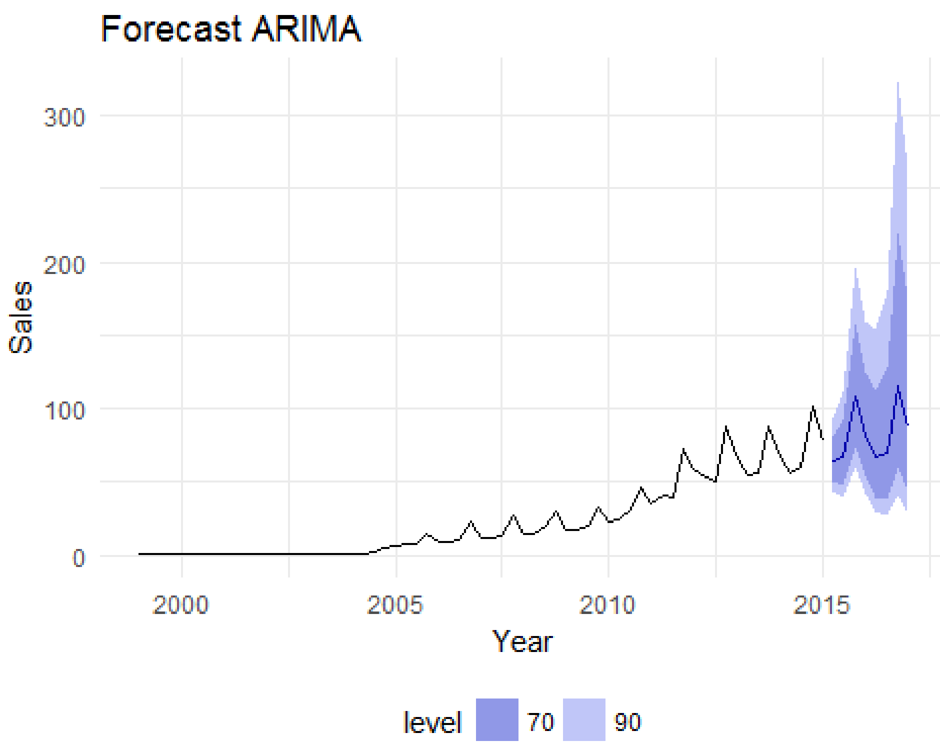

```
ggplot(df_newl)+geom_point(aes(x=time,y=value))+geom_line(aes(x=time,y=va
lue))+ geom_forecast(fventas.arima2,alpha=0.4)+ggtitle("ARIMA2: Apple For
ecast")
```

```
## Warning in geom_forecast(fventas.arima2, alpha = 0.4): Use autolayer
## instead of geom_forecast to add a forecast layer to your ggplot object
.

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

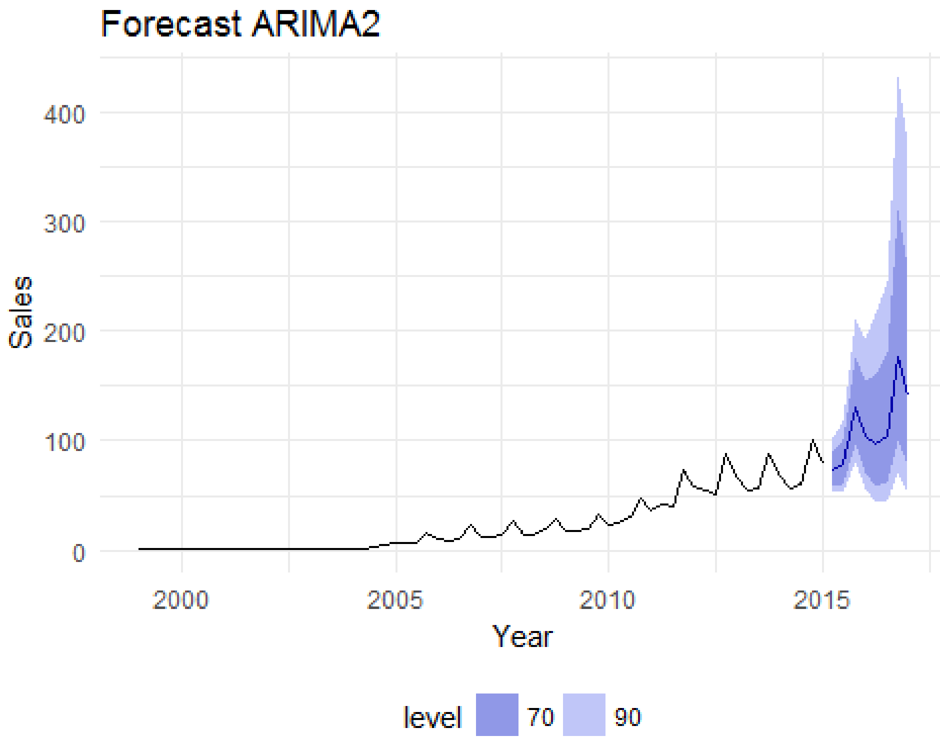


Con autoplot:

```
plotarima <- autoplot(fventas.arima)+
        labs(y="Sales",x="Year")+
        ggtitle("Forecast ARIMA")+
        theme_minimal()+
        scale_color_economist()+
        theme(legend.position = "bottom")
plotarima
```




```r
plotarima2 <- autoplot(fventas.arima2)+
  labs(y="Sales",x="Year")+
  ggtitle("Forecast ARIMA2")+
  theme_minimal()+
  scale_color_economist()+
  theme(legend.position = "bottom")
plotarima2
```



La prediccion es la linea azul. Vemos la clara diferencia entre ambos modelos, siendo el modelo ARIMA (1,0,0)(1,1,0)[4] un poco mejor que el modelo ARIMA2 (1,0,3)(0,1,1)[4].

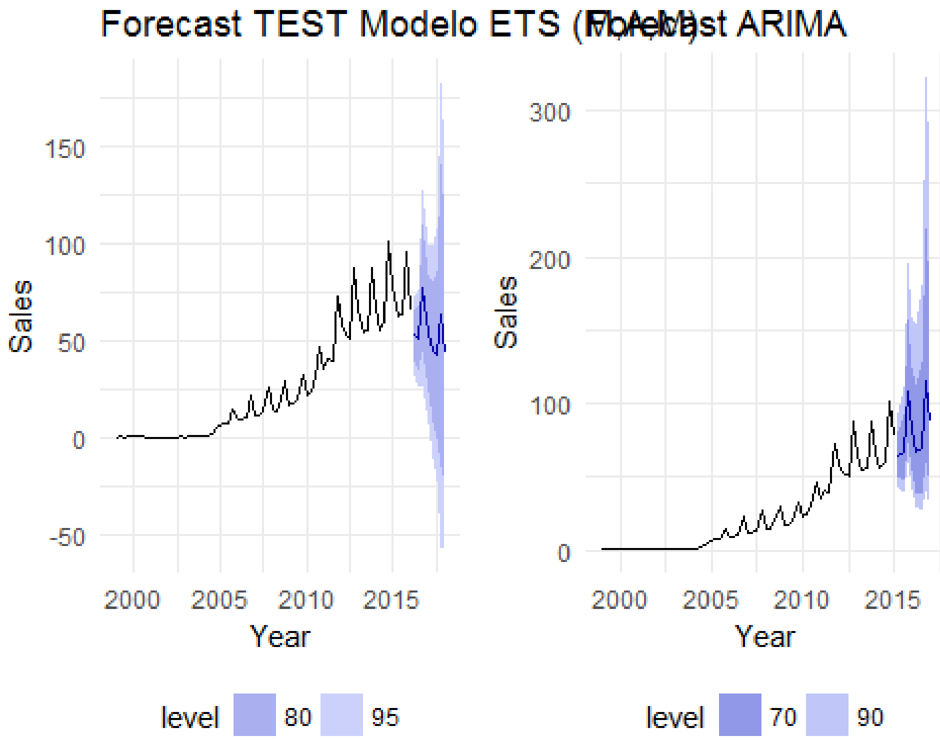
Si prestamos atencion a las predicciones del modelo ARIMA, observamos como en los Q4 el modelo estima unas ventas notablemente mayores a las de los otros trimestres.

```
fventas.arima

##         Point Forecast    Lo 70     Hi 70    Lo 90     Hi 90
## 2015 Q2       63.87911 50.45181  80.87998 43.92546  92.89694
## 2015 Q3       66.62150 48.38795  91.72582 40.10623 110.66672
## 2015 Q4      108.60618 74.58567 158.14435 59.82079 197.17730
## 2016 Q1       82.59056 54.43531 125.30838 42.61854 160.05242
## 2016 Q2       67.25047 39.94798 113.21288 29.42440 153.70321
## 2016 Q3       70.21197 38.76557 127.16753 27.35346 180.22297
## 2016 Q4      115.28329 60.25182 220.57817 41.16676 322.83906
## 2017 Q1       87.55462 43.85305 174.80683 29.22308 262.32045
```

Comparacion ETS y ARIMA

Vamos a comparar los graficos de los dos modelos elegidos:

```
grid.arrange(plotets,plotarima,ncol=2)
```



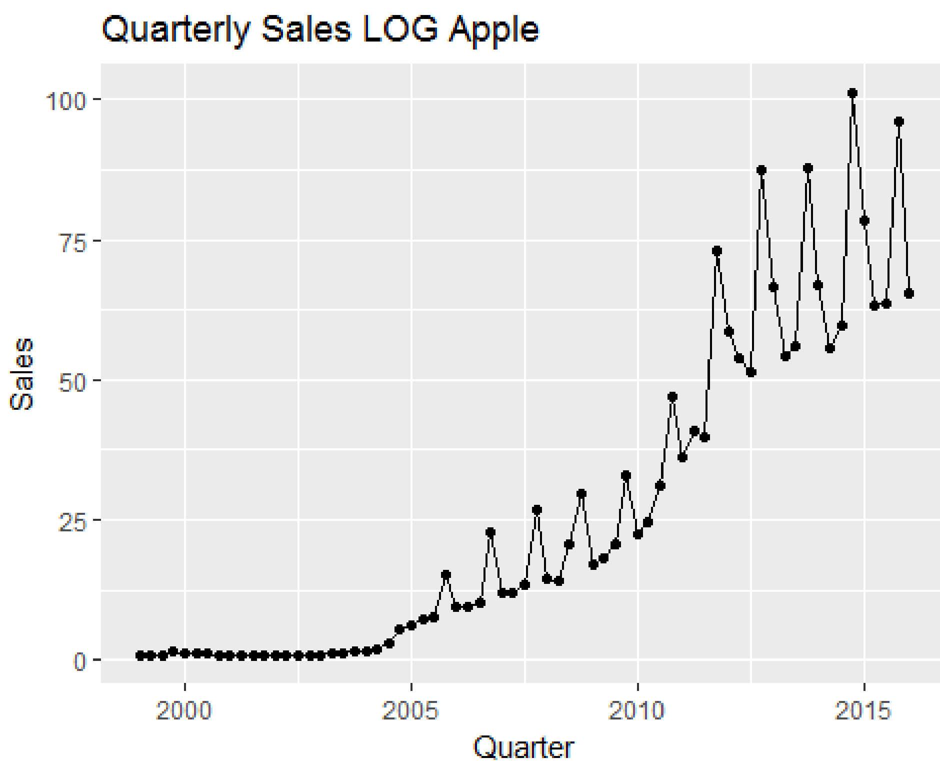
ANALISIS DE INTERVENCION

vamos a comprobar si podemos mejorar el modelo utilizando un modelo ARIMAX.

Para ello tenemos ver si hay que practicar analisis de intervencion.

Lo primero es observar el grafico de crecimiento (graficamos zlVentas)

```
ggplot(df_newl)+geom_point(aes(x=time,y=value))+geom_line(aes(x=time,y=va
lue))+ylab("Sales")+ggtitle("Quarterly Sales LOG Apple")+xlab("Quarter")

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

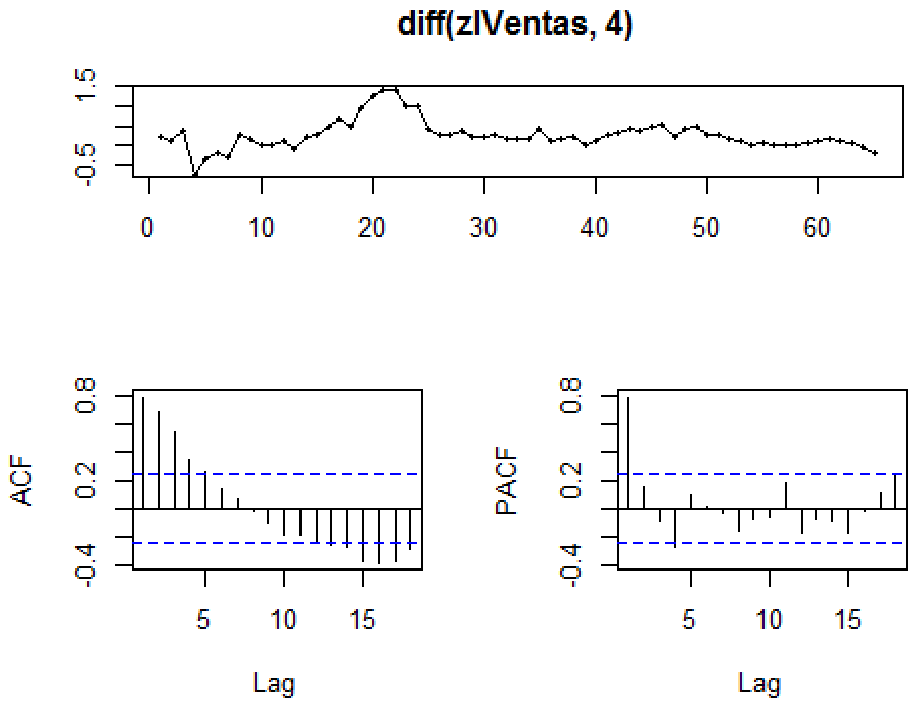


Se puede observar un incremento de las ventas notable en el Q3 de 2004. Esto coincide con el lanzamiento de la 4º generacion de iPods. Podemos ver el escalon que tiene lugar en 2004 y que se mantiene en el tiempo. https://www.cnet.com/pictures/the-complete-history-of-apples-ipod/

En el Q4 de 2010 comienza el verdadero crecimiento exponencial de la compañía como consecuencia del lanzamiento de iPhone 4, que marco un antes y un despues en la telefonia movil y en la trayectoria de Appel. Podriamos considerar aquí otro escalón, sin embargo, teniendo en cuenta que después del lanzamiento del iPhone 4 cada Q4 Apple obtiene un crecimiento de las ventas considerable con cada nuevo lanzamiento, considero que la suma de las intervenciones de Apple forman parte intrínseca de la tendencia en sus ventas y, por tanto, las asumo como hechos normales.

Consideraré únicamente el escalón que provoca el lanzamiento de la cuarta generación de iPods a la hora de estimar el modelo ARIMAX, ya que se trata de la intervención que da lugar al germen de crecimiento de las ventas de Apple.

```
tsdisplay(diff(zlVentas,4))
```



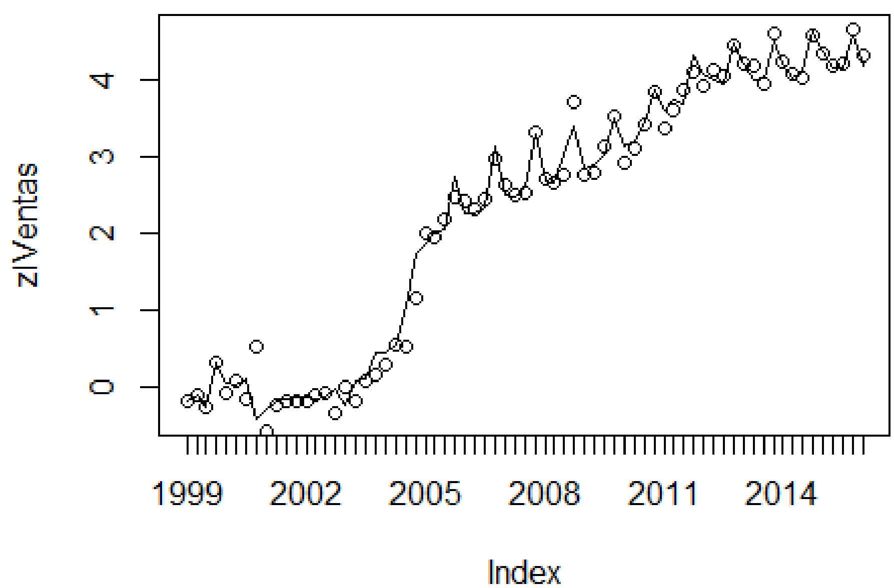
ARIMAX

```
fitARIMAX=arimax(log(zVentas),order=c(1,0,0),
              seasonal=list(order=c(1,1,0),period=4),
              xtransf=data.frame(iPod2004=1*(seq(zVentas)==25)),
              transfer=list(c(1,0)),
              method='ML')

fitARIMAX

##
## Call:
## arimax(x = log(zVentas), order = c(1, 0, 0), seasonal = list(order = c
(1, 1,
##     0), period = 4), method = "ML", xtransf = data.frame(iPod2004 = 1
* (seq(zVentas) ==
##     25)), transfer = list(c(1, 0)))
##
## Coefficients:
##          ar1     sar1  iPod2004-AR1  iPod2004-MA0
##       0.8952  -0.4468        1.0118        0.5173
## s.e.  0.0585   0.1513        0.0344        0.1849
##
## sigma^2 estimated as 0.04209:  log likelihood = 9.72,  aic = -11.45
```

Obtenemos un modelo con menos AIC que el anterior ARIMA, aunque no es del todo correcto comparar ARIMA y ARIMAX en criterios de información.

Graficamos el modelo

```
plot(log(zVentas), ylab="zlVentas")
points(fitted(fitARIMAX))

## Warning in fitted.Arima(fitARIMAX): Métodos incompatibles ("Ops.zoo",
## "Ops.ts") para "-"
```



Outliers

Aditivos

Tiene valores atipicos aditivos el modelo?

```
detectAO(fitARIMAX)

##              [,1]      [,2]      [,3]
## ind     7.000000  8.000000 24.000000
## lambda2 4.620476 -5.067049  3.603647
```

El modelo detecta 2 aditivos. Tendremos que comprobar si efectivamente se trata de outliers o si se trata de intervenciones. En este caso no se trata de intervenciones

Innovativos

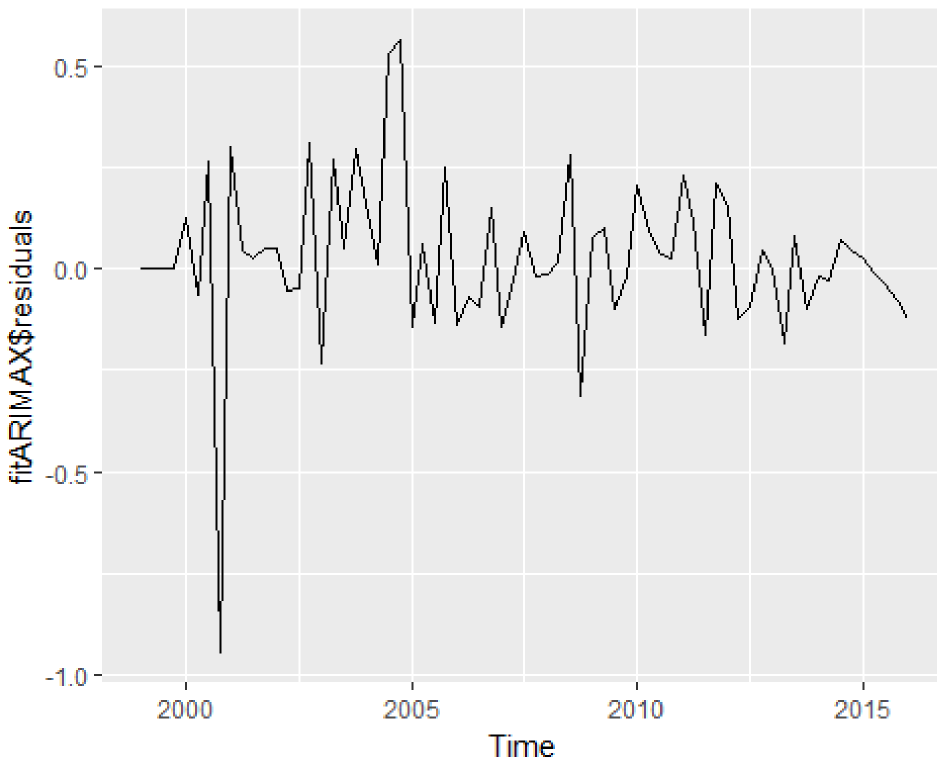
Tiene valores atipicos innovativos el modelo?

```
detectIO(fitARIMAX)
```

```
##                [,1]      [,2]
## ind       8.000000 24.000000
## lambda1 -5.818846  3.464391
```

Existe un valor atipico innovativo. Los valores atipicos innovativos afectan al error y empeoran los modelos.

```
autoplot(fitARIMAX$residuals)
```



La caída que hay al final del a?o 2000, en el Q4, contrasta mucho con el resto de la tendencia. Hemos visto que el Q4 es el trimestre con mas ventas año tras año y que sirve de propulsor para el crecimiento en la facturacion de Apple.

Vamos a estimar el modelo ARIMAX para ver si mejora definiendo el outlier innovativo

```
fitARIMAX2=arimax(log(zVentas),order=c(1,0,0),
              seasonal=list(order=c(1,1,0),period=4),
              xtransf=data.frame(iPod2004=1*(seq(zVentas)==25)),
              transfer=list(c(1,0)),
              io=c(8,24,23),
              method='ML')

fitARIMAX2

##
## Call:
## arimax(x = log(zVentas), order = c(1, 0, 0), seasonal = list(order = c
```

```
(1, 1,
##     0), period = 4), method = "ML", io = c(8, 24, 23), xtransf = data.
frame(iPod2004 = 1 *
##     (seq(zVentas) == 25)), transfer = list(c(1, 0)))
##
## Coefficients:
##          ar1     sar1     IO-8   IO-24   IO-23  iPod2004-AR1  iPod2004
-MA0
##       0.8354  -0.1989  -0.5619  0.7548  0.5117        1.0133        1.
2165
## s.e.  0.0789   0.1525   0.0919  0.1900  0.1315        0.0084        0.
2432
##
## sigma^2 estimated as 0.02341:  log likelihood = 29.21,  aic = -44.42
```

El AIC del modelo disminuye. El criterio de información del modelo mejora mucho AIC=-44.2

Aditivos

Tiene valores atipicos aditivos el modelo?

```
detectAO(fitARIMAX2)

##              [,1]
## ind     7.000000
## lambda2 3.712198
```

El modelo detecta 2 aditivos. Tendremos que comprobar si efectivamente se trata de outliers o si se trata de intervenciones. En este caso no se trata de intervenciones

Innovativos

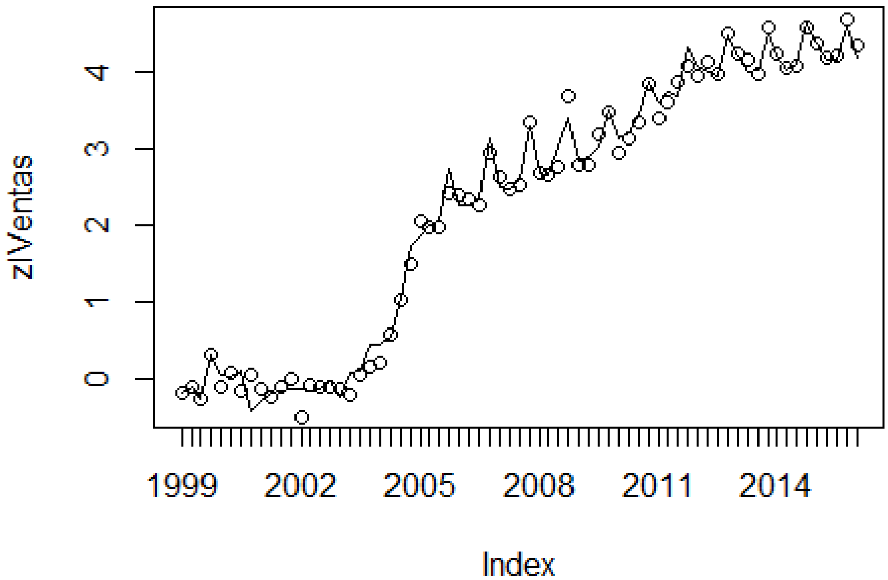
Tiene valores atipicos innovativos el modelo?

```
detectIO(fitARIMAX2)

## [1] "No IO detected"
```

Se eliminan los valores atípicos innovativos completamente.

Graficamos el modelo

```
plot(log(zVentas), ylab="zlVentas")
points(fitted(fitARIMAX2))

## Warning in fitted.Arima(fitARIMAX2): Métodos incompatibles ("Ops.zoo",
## "Ops.ts") para "-"
```



Conclusiones

A lo largo de la practica hemos comprobado que los tres modelos, ETS, ARIMA y ARIMAX2, se ajustan bastante bien aunque con diferencias en función del peso que le están dando a la información. En funcion del objetivo que tengamos dentro del negocio, utilizaremos un modelo u otro.

El ETS tiene mucho mas en cuenta la información mas reciente. En este sentido, es posible que convenga utilizar el modelo ETS para predecir a muy corto plazo, a 1 o 2 trimestres: dado que el ETS tiene en cuenta que las ventas del iphone 7 no fueron muy buenas y que las ventas cayeron (última información disponible en la serie), si no se espera el lanzamiento de un producto innovador por parte de Apple parece que convendría utilizar este modelo para hacer la predicción.

A la hora de predecir periodos mas lejanos, convendra utilizar el modelo ARIMAX2, puesto que ha mejorado el ARIMA que teníamos.

Lo ideal, dado el objeto de análisis y la incertidumbre inherente a cualquier prediccion, seria combinar ambos modelos para fundamentar mejor las decisiones.