Ingresos_Apple

Sergio Casares

16/11/2020

Contents

1. Introducción al trabajo

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.

Una vez seleccionado el modelo se estimara el modelo con todos los datos y se harán las predicciones del año 2017 y 2018.

2.Importación de librerías y Dataset

Importamos las librerías necesarias para la realización del trabajo.

```r
library(readr)
library(forecast)
library(xts)
library(ggplot2)
library(ggfortify)

#Importamos los datos

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

2.1 Transformación del dataset

Eliminamos el índice del dataset para poder facilitar la manipulación de los datos.

```r
ventas <- data.frame(ventas[,-1], row.names =  ventas[,1])

colnames(ventas)[1] <- 'Ingresos'

head(ventas)
```

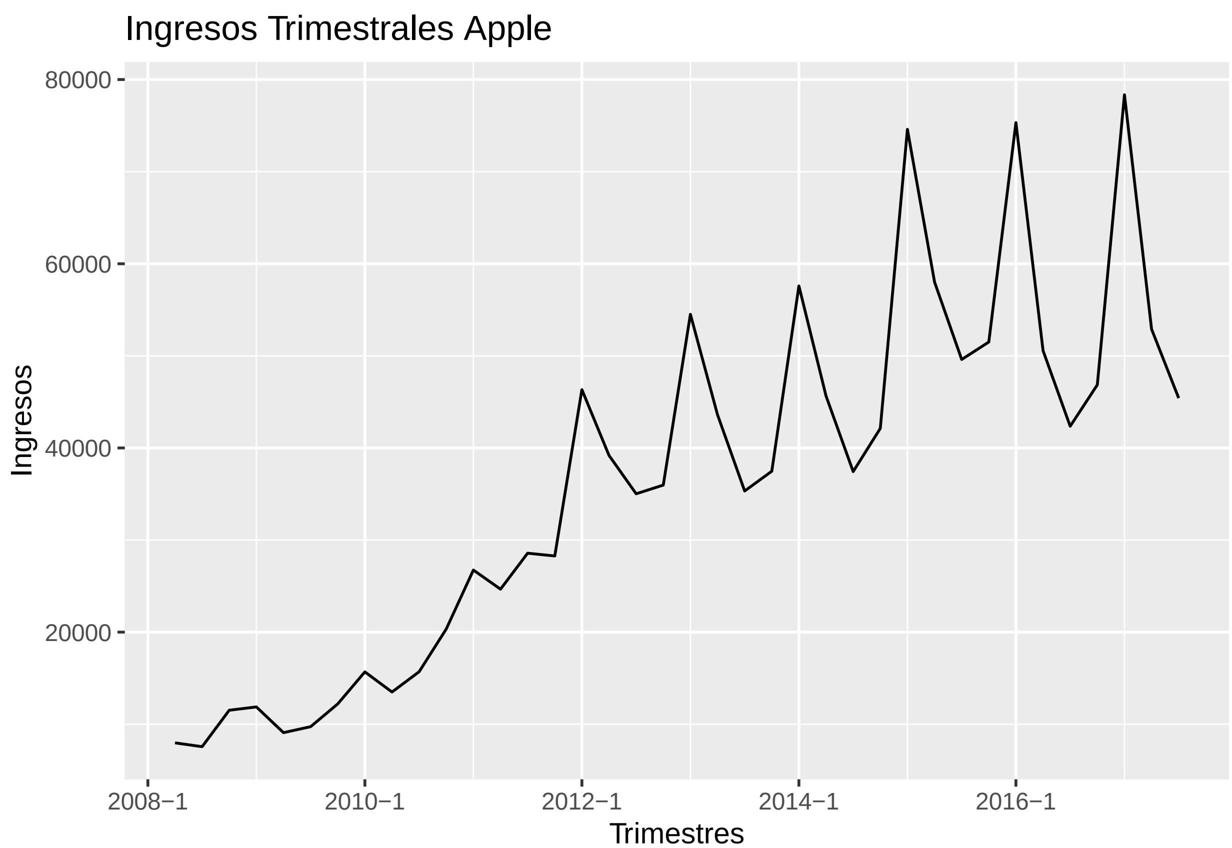
```
##         Ingresos
## Q2 2008     7980
## Q3 2008     7561
## Q4 2008    11520
## Q1 2009    11880
## Q2 2009     9084
## Q3 2009     9734
```

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

xVentas <- xts(ventas$Ingresos, order.by = rawDate)
xVentas <- to.quarterly(xVentas)

zVentas <- as.zoo(xVentas$xVentas.Open)
names(zVentas) = "Ingresos"

zlVentas = log(zVentas)

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

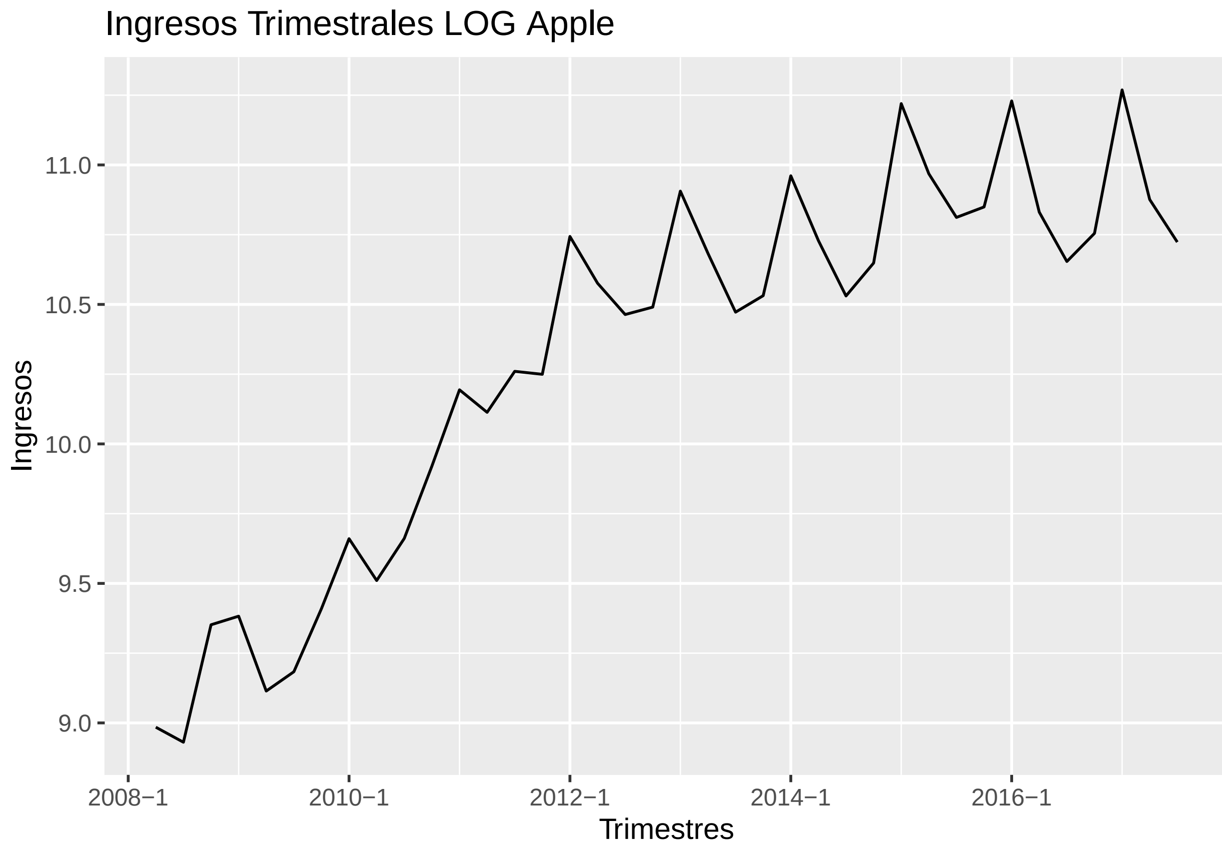
2.2 Análisis Exploratorio de Datos (EDA)

En este apartado, realizaremos una visualización de los ingresos de Apple de manera general y posteriormente lo diferenciaremos por trimestres.

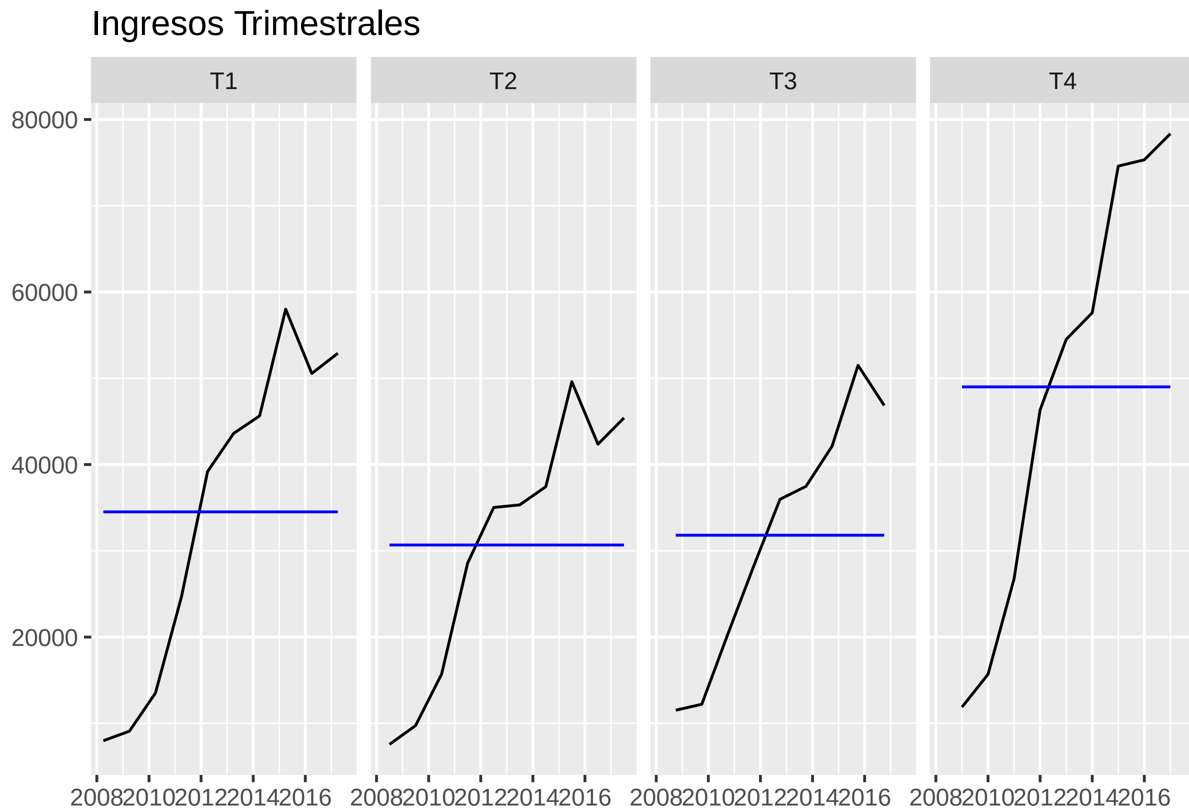
```r
#Visualizamos los datos de manera general

zventas <- as.zoo(xVentas$xVentas.Open)
names(zVentas) = "Ingresos"


autoplot(zventas)+ggtitle('Ingresos Trimestrales Apple')+
  xlab('Trimestres')+ylab('Ingresos')
```




```r
zlventas = log(zventas)

autoplot(zlventas)+ggtitle('Ingresos Trimestrales LOG Apple')+
  xlab('Trimestres')+ylab('Ingresos')
```




```r
#Dividimos los datos por trimestres

tsventas <- ts(coredata(ventas), start = c(2008,2), frequency = 4)

ggfreqplot(tsventas, freq = 4, nrow = 1, facet.labeller = c('T1','T2','T3','T4')) +
  ggtitle('Ingresos Trimestrales')
```



En la primera gráfica se puede observar una tendencia alcista de las ventas, lo que nos indica que tanto la media como la varianza no son estacionarias, este tipo de series se caracterizan por:

- Pueden mostrar cambios de varianza.
- Pueden mostrar una tendencia, es decir que la media crece o baja a lo largo del tiempo.
- Además, se puede observar que el comportamiento de la serie es parecido en ciertos tiempos periódicos en el tiempo.

En la segunda gráfica se puede observar una clara similitud de los primeros, segundos y terceros trimestres durante todos los años, en cambio, el cuarto trimestre se diferencia de los demás por una clara y mayor tendencia alcista de las ventas. Todo esto nos puede indicar cierta estacionalidad de los datos.

3. Predicción

Para poder predecir, debemos dividir la muestra en training y test tanto de la muestra con los ingresos de manera logarítmica como sin ella.

Con la distribución y representación actual de los datos, se puede observar que la serie temporal es el tipo Aditive Damped, Multiplicative.

```
cOmit = 3

nObs = length(zventas)
```

```r
nObs_log =length(zlventas)
#sub sample
oVentas = window(zventas, start = index(zventas[1]), end = index(zventas[nObs - cOmit]))

#sub sample log
olVentas = window(zlventas, start = index(zlventas[1]), end = index(zlventas[nObs_log - cOmit]))
```

3.1. Seasonal, trend & remainder

Antes de comenzar con la predicción, realizamos un análisis de la estacionalidad, la tendencia y el remainder.

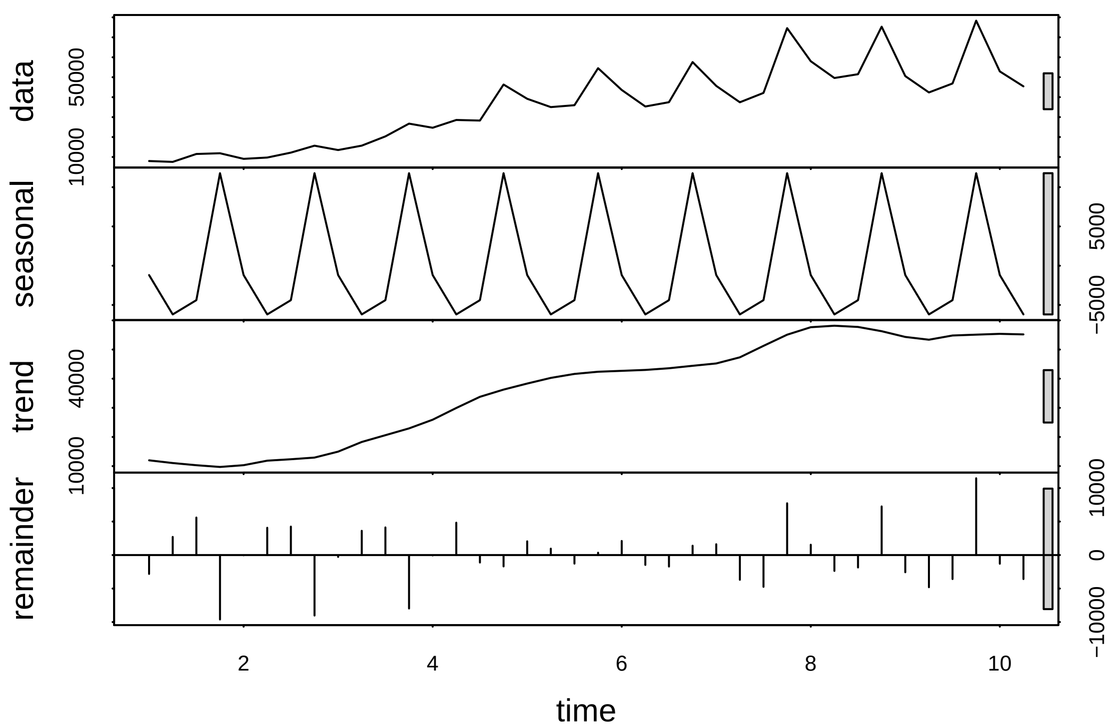
- Data: se observa la tendencia alcista
- Seasonal: se puede observar una clara estacionalidad de los datos
- Tendencia: se observa, al igual que data, una clara tendencia alcista
- Remainder: es el residuo de la estacionalidad (seasonal) más el ajuste de la tendencia (trend), no se observan valores anormales.

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

```
##  Call:
##  stl(x = xVentas[, 1], s.window = "periodic")
##
## Components
##        seasonal     trend   remainder
##  1 Q1 -1191.515 11987.713 -2816.19786
##  1 Q2 -6202.795 11054.647  2709.14756
##  1 Q3 -4388.396 10306.954  5601.44193
##  1 Q4 11782.720  9712.795 -9615.51515
##  2 Q1 -1191.515 10313.378   -37.86329
##  2 Q2 -6202.795 11865.051  4071.74373
##  2 Q3 -4388.396 12348.763  4246.63276
##  2 Q4 11782.720 12927.130 -9026.85026
##  3 Q1 -1191.515 14968.422  -277.90712
##  3 Q2 -6202.795 18282.820  3619.97453
##  3 Q3 -4388.396 20608.629  4122.76624
##  3 Q4 11782.720 22936.090 -7977.81022
##  4 Q1 -1191.515 25917.203   -58.68777
##  4 Q2 -6202.795 29940.698  4833.09640
##  4 Q3 -4388.396 33777.770 -1119.37412
##  4 Q4 11782.720 36249.530 -1699.24976
##  5 Q1 -1191.515 38326.357  2051.15766
##  5 Q2 -6202.795 40267.415   958.37982
##  5 Q3 -4388.396 41633.253 -1278.85770
##  5 Q4 11782.720 42379.047   350.23228
##  6 Q1 -1191.515 42689.157  2105.35767
##  6 Q2 -6202.795 42998.839 -1473.04409
##  6 Q3 -4388.396 43579.104 -1718.70816
##  6 Q4 11782.720 44414.890  1396.39005
##  7 Q1 -1191.515 45222.672  1614.84282
```

```
##  7 Q2 -6202.795 47330.414 -3695.61937
##  7 Q3 -4388.396 51256.536 -4745.14062
##  7 Q4 11782.720 55088.745  7727.53443
##  8 Q1 -1191.515 57657.107  1544.40748
##  8 Q2 -6202.795 58173.837 -2366.04237
##  8 Q3 -4388.396 57747.384 -1857.98841
##  8 Q4 11782.720 56283.309  7257.97053
##  9 Q1 -1191.515 54318.201 -2569.68577
##  9 Q2 -6202.795 53365.704 -4804.90925
##  9 Q3 -4388.396 54818.937 -3578.54150
##  9 Q4 11782.720 55104.114 11464.16602
## 10 Q1 -1191.515 55382.281 -1294.76586
## 10 Q2 -6202.795 55190.924 -3580.12898
```

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



3.3. Modelo ETS

Utilizando la función ETS podemos seleccionar el modelo más adecuado para nuestra serie temporal.

La función de ETS nos indica que la serie temporal es 'Multiplicative Holt-Winters' method with multiplicative errors, este modelo calcula valores expoencialmente suavizados para los distintos niveles, tendencia y ajustes estacionales de la predicción. Este método multiplica las tendencias predecidas por la estacionalidad.

Este método es el más adecuado para datos con tendencia y estacionalidad que incrementa a lo largo del tiempo. Esto resulta en una curva que reproduce los cambios en los datos.

3.3.A Seleccionar ETS automático (no logarítmica)

```r
#select automatic ETS

## Select automatic ETS
etsfit <- ets(oVentas)
etsfit
```

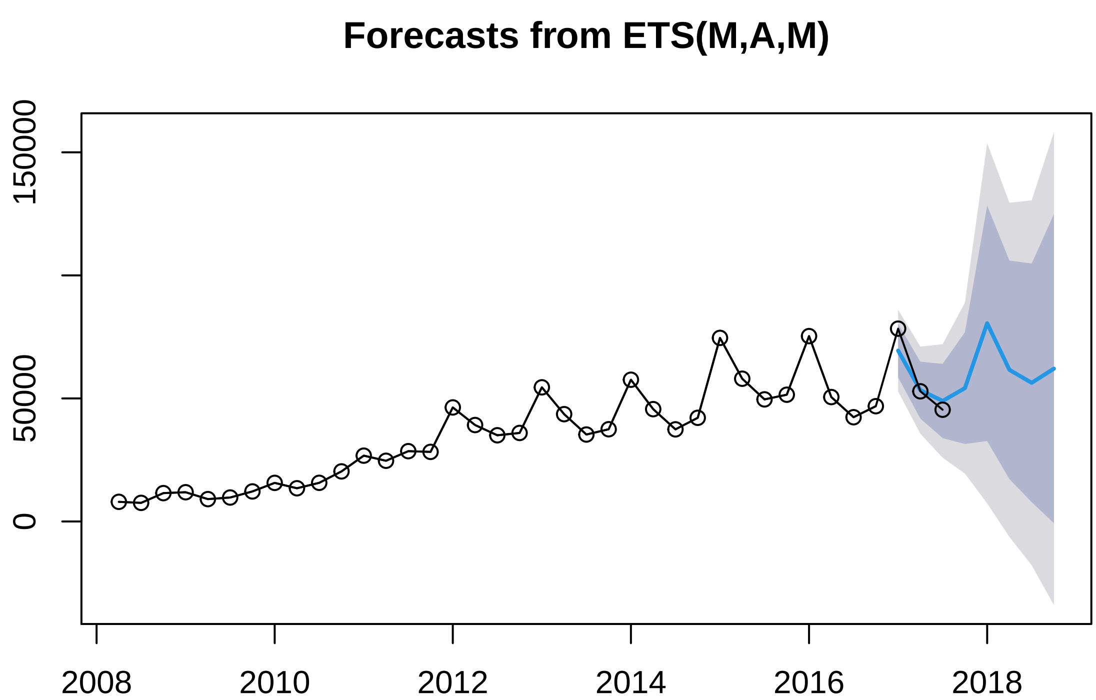
```
## ETS(M,A,M)
##
## Call:
##  ets(y = oVentas)
##
##   Smoothing parameters:
##     alpha = 0.493
##     beta  = 0.493
##     gamma = 0.507
##
##   Initial states:
##     l = 7125.3462
##     b = 1485.7975
##     s = 1.1511 1.1163 0.8322 0.9004
##
##   sigma:  0.1222
##
##      AIC     AICc      BIC
## 703.9538 711.1538 717.9519
```

```r
#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 = oVentas)
##
##   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
```

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



Predicción y Accuracy Realizamos el cálculo de la diferencia entre los valores predecidos y los reales.

Uno de los estimadores que más indican la precisión del modelo es el Error Medio, que en este caso es de 1631,71, que para valores de entre 45000 y 80000 millones, representa una diferencia de entre el 2% y el 5% de error, por lo que la predicción de nuestro modelo se ajusta bien a la realidad.

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

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

```
etsfit<-ets(window(tsventas,end=2016+3/4))

fventas.ets=forecast(etsfit,h=cOmit)


forecast:::testaccuracy(fventas.ets$mean,window(tsventas,start=2008),test = NULL, d = NULL, D = NULL)
```

```
##            ME          RMSE           MAE           MPE          MAPE
## 1631.71485476 5547.24116365 4309.06585239    0.88999690    6.69226842
##          ACF1     Theil's U
##   -0.05148699    0.19082258
```

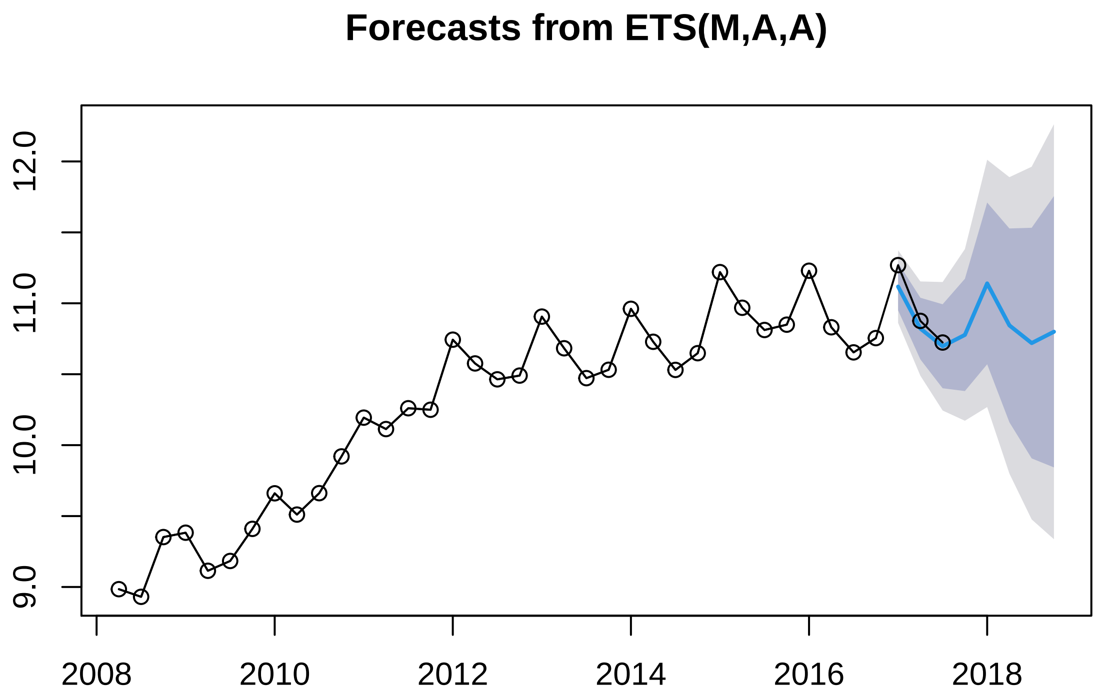
3.3.B. Seleccionar ETS automático (logarítmica)

```
# Selección automática ETS
etsfit_log <- ets(olVentas)

# Forecast model
fventas.ets_log = forecast(etsfit_log)

#Resultado
summary(fventas.ets_log)
```

```
##
## Forecast method: ETS(M,A,A)
##
## Model Information:
## ETS(M,A,A)
##
## Call:
##  ets(y = olVentas)
##
##   Smoothing parameters:
##     alpha = 0.4924
##     beta  = 0.3676
##     gamma = 0.5076
##
##   Initial states:
##     l = 9.0376
```

```
##     b = 0.0906
##     s = 0.1881 0.0807 -0.2048 -0.064
##
##   sigma:  0.0117
##
##         AIC        AICc         BIC
## -14.9459243  -7.7459243  -0.9477917
##
## Error measures:
##                        ME      RMSE        MAE         MPE      MAPE      MASE
## Training set -0.006606452 0.1059325 0.08553984 -0.06269475 0.8397395 0.3489455
##                     ACF1
## Training set 0.007074111
##
## Forecasts:
##         Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## 2017 Q1       11.11778 10.950534 11.28502 10.862000 11.37356
## 2017 Q2       10.82187 10.604626 11.03912 10.489623 11.15412
## 2017 Q3       10.69711 10.401017 10.99319 10.244277 11.14993
## 2017 Q4       10.77713 10.381404 11.17286 10.171919 11.38234
## 2018 Q1       11.14016 10.570250 11.71007 10.268556 12.01177
## 2018 Q2       10.84425 10.161066 11.52744  9.799408 11.88910
## 2018 Q3       10.71949  9.906558 11.53242  9.476220 11.96276
## 2018 Q4       10.79951  9.843115 11.75591  9.336827 12.26220
```

```r
plot(fventas.ets_log)
lines(window(zlVentas), type = "o")
```




```
matrix(c(fventas.ets_log$mean[1:cOmit], zlVentas[(nObs_log - cOmit + 1):nObs_log]), ncol = 2)
```

Predicción y Accuracy - ETS

```
##          [,1]     [,2]
## [1,] 11.11778 11.26895
## [2,] 10.82187 10.87608
## [3,] 10.69711 10.72344
```

```
# Predicciones y Precisión

tsVentas_log <- log(tsventas)

etsfit_log <- ets(window(tsVentas_log, end = 2016 + 3/4))
fventas.ets_log = forecast(etsfit_log , h = cOmit)

forecast::accuracy(fventas.ets_log$mean, window(tsVentas_log, start = 2017), test = NULL, d = NULL, D =
```

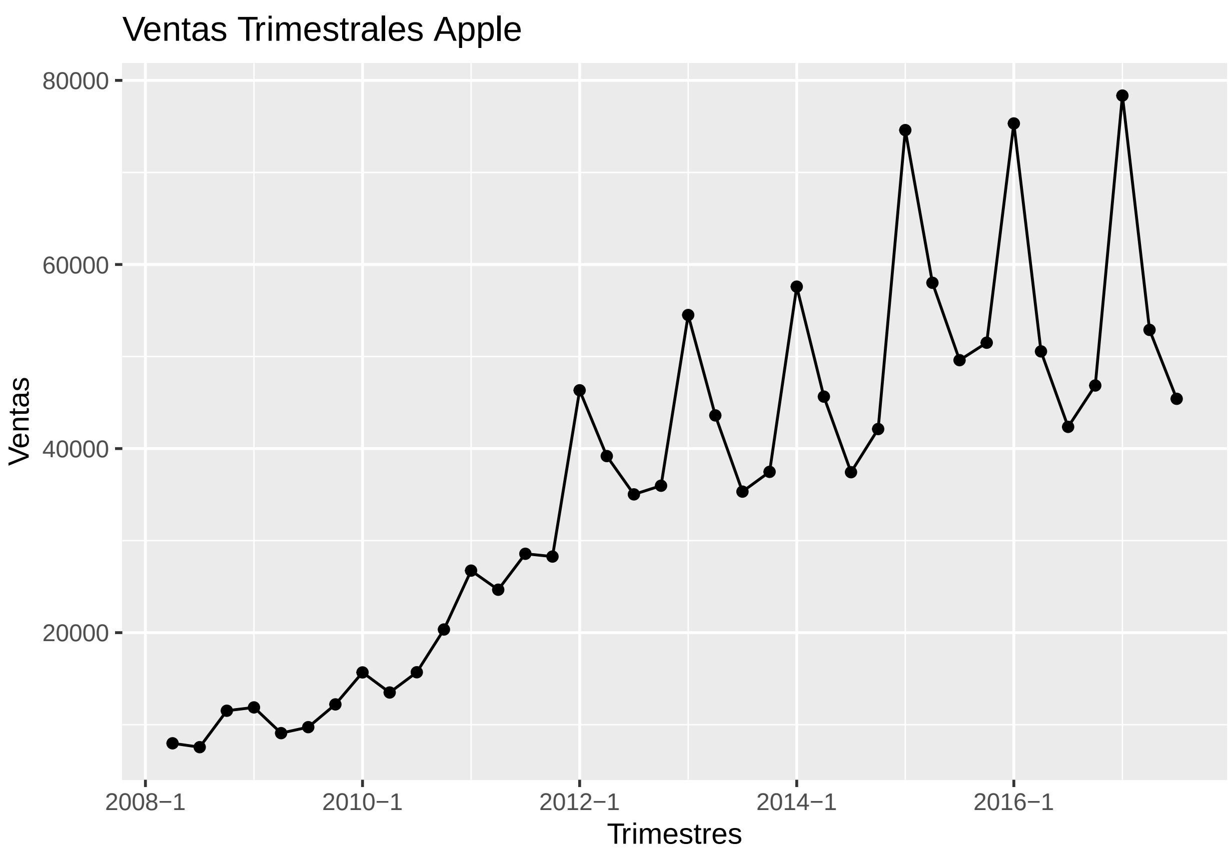
```
##                   ME       RMSE       MAE       MPE      MAPE        ACF1
## Test set 0.0772417 0.09396171 0.0772417 0.6951939 0.6951939 -0.06175923
##          Theil's U
## Test set 0.1433107
```

3.4. Modelo ARIMA

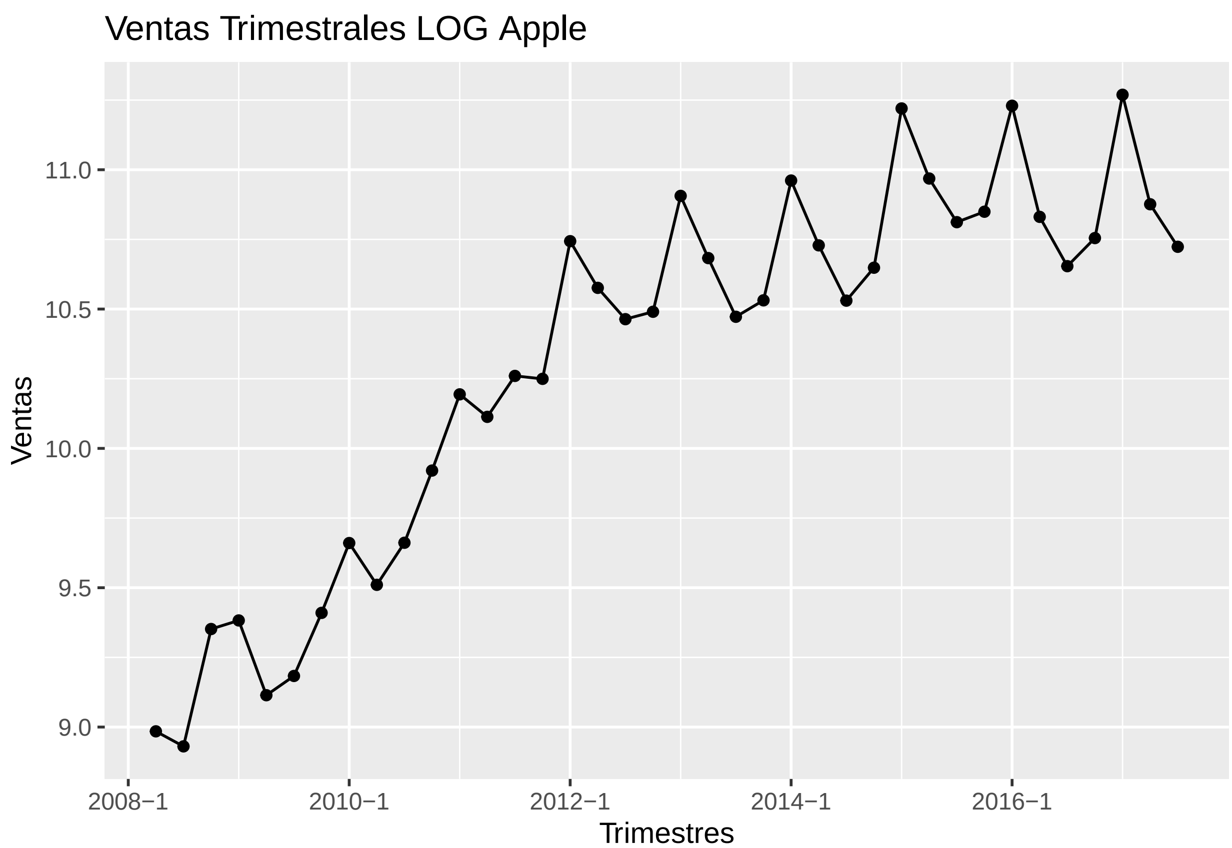
Es un modelo estadístico que utiliza variaciones y regresiones de datos estadísticos con el fin de encontrar patrones para una predicción hacia el futuro. Se trata de un modelo dinámico de series temporales, es decir, las estimaciones futuras vienen explicadas por los datos del pasado y no por variables independientes.

Transformación de los datos Creamos dos dataframes, uno con los ingresos con los valores normales y otro dataframe con los logaritmos de los valores para evitar la estacionalidad de la varianza, lo cuál nos ayuda a poder predecir mejor.

```
zlVentas=log(zventas)

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

ggplot(df_new)+geom_point(aes(x=time,y=value))+geom_line(aes(x=time,y=value))+ylab("Ventas")+ggtitle("V
```




```
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=value))+ylab("Ventas")+ggtitle("
```



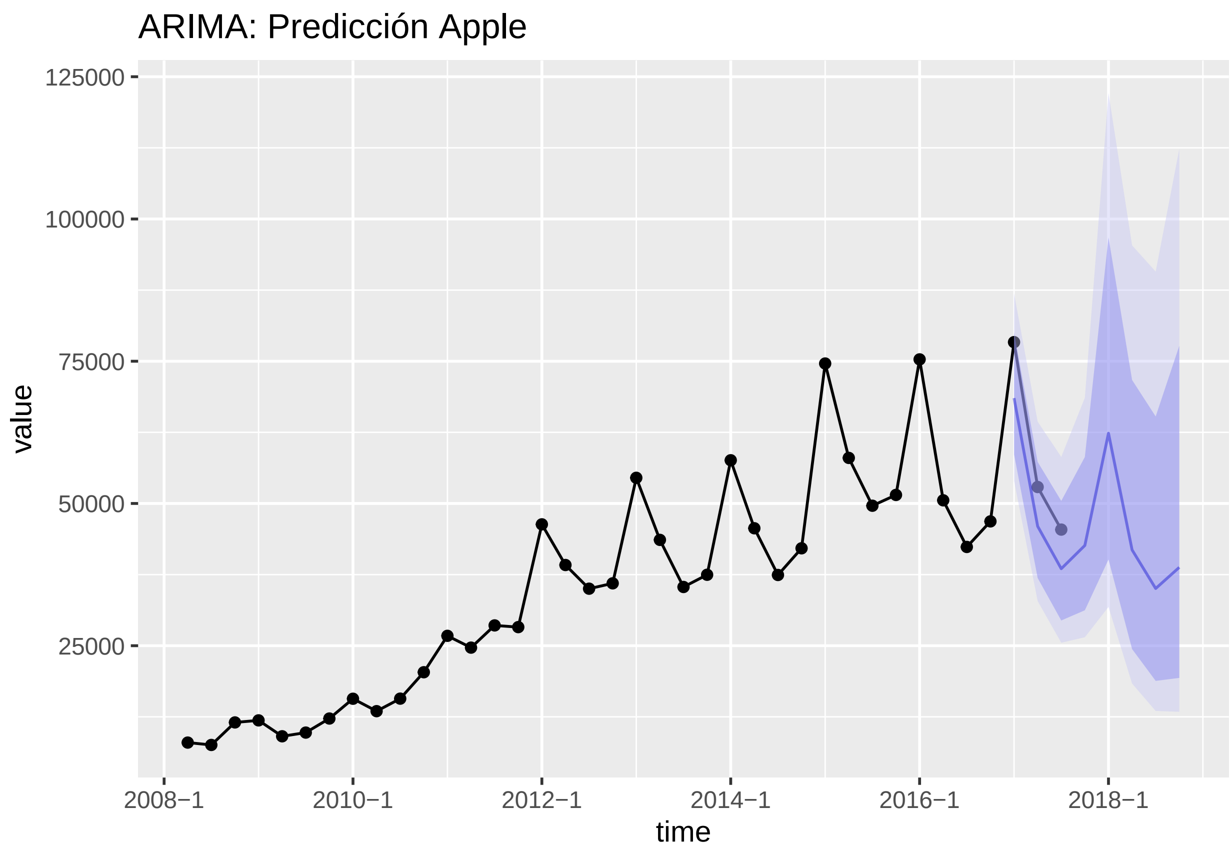
3.4.A Seleccionar ARIMA automático (no logarítmica)

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

```
## Series: oVentas
## 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
```

```
fit_arima = auto.arima(oVentas, lambda = 0)

fventas.arima = forecast(fit_arima)

ggplot(df_new) +
  geom_point(aes(x = time, y = value)) +
  geom_line(aes(x = time, y = value)) +
```

```r
  geom_forecast(fventas.arima, alpha = 0.4) +
  ggtitle("ARIMA: Predicción Apple")
```




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

Predicciones y Accuracy ARIMA

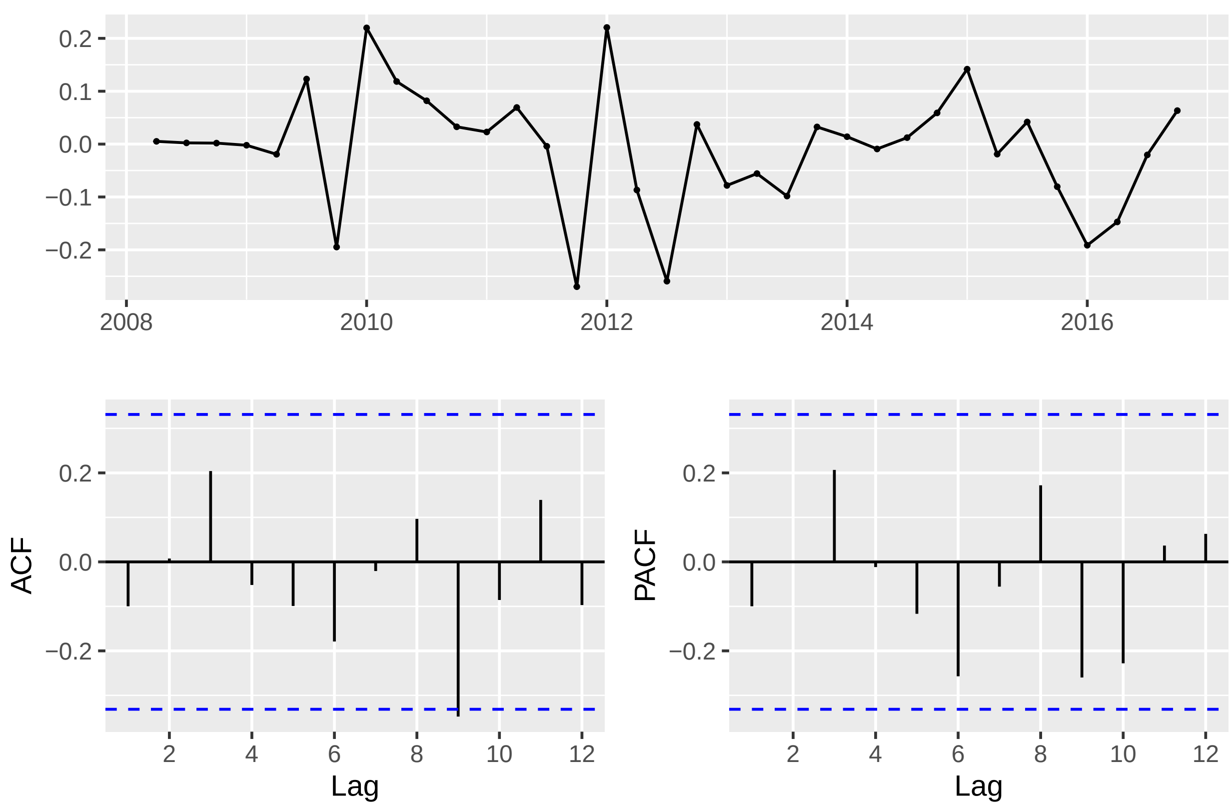
```
##          [,1]  [,2]
## [1,] 68524.50 78351
## [2,] 45993.21 52896
## [3,] 38534.34 45408
```

```r
arimafit <- ets(window(tsventas, end = 2016 + 3/4))
fventas.arima = forecast(arimafit , h = cOmit)

forecast::accuracy(fventas.arima$mean, window(tsventas, start = 2017), test = NULL, d = NULL, D = NULL)
```

```
##                ME     RMSE      MAE       MPE     MAPE        ACF1 Theil's U
## Test set 1631.715 5547.241 4309.066 0.8899969 6.692268 -0.05148699 0.1908226
```

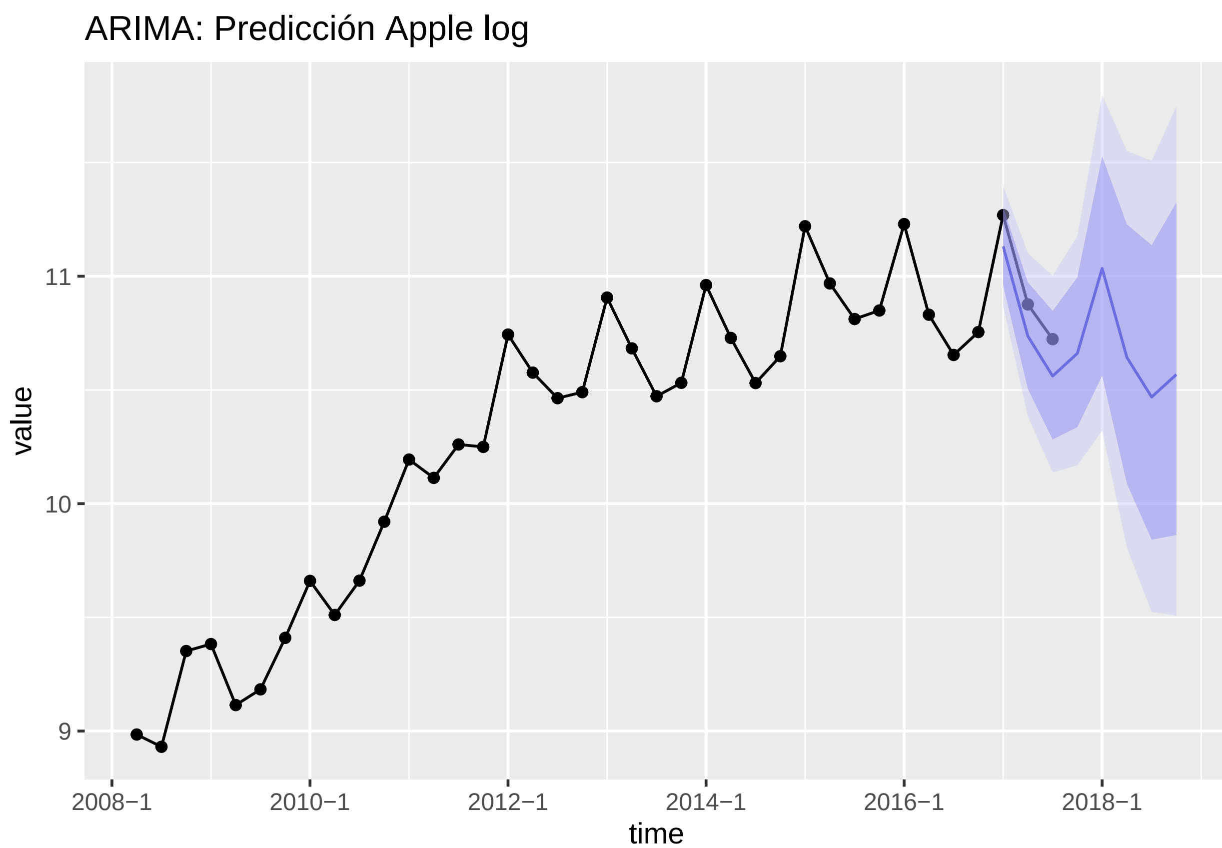
Análisis de residuos .

```r
ggtsdisplay(fit_arima$residuals)
```



3.4.B Seleccionar ARIMA automático (logarítmica)

```r
fit_arima_log = auto.arima(olVentas, lambda = 0)
fventas.arima_log = forecast(fit_arima_log)


ggplot(df_newl) +
  geom_point(aes(x = time, y = value)) +
  geom_line(aes(x = time, y = value)) +
  geom_forecast(fventas.arima_log, alpha = 0.4) +
  ggtitle("ARIMA: Predicción Apple log")
```




```
matrix(c(fventas.arima_log$mean[1:cOmit], zlVentas[(nObs_log - cOmit + 1):nObs_log]), ncol = 2)
```

Predicciones y Accuracy ARIMA

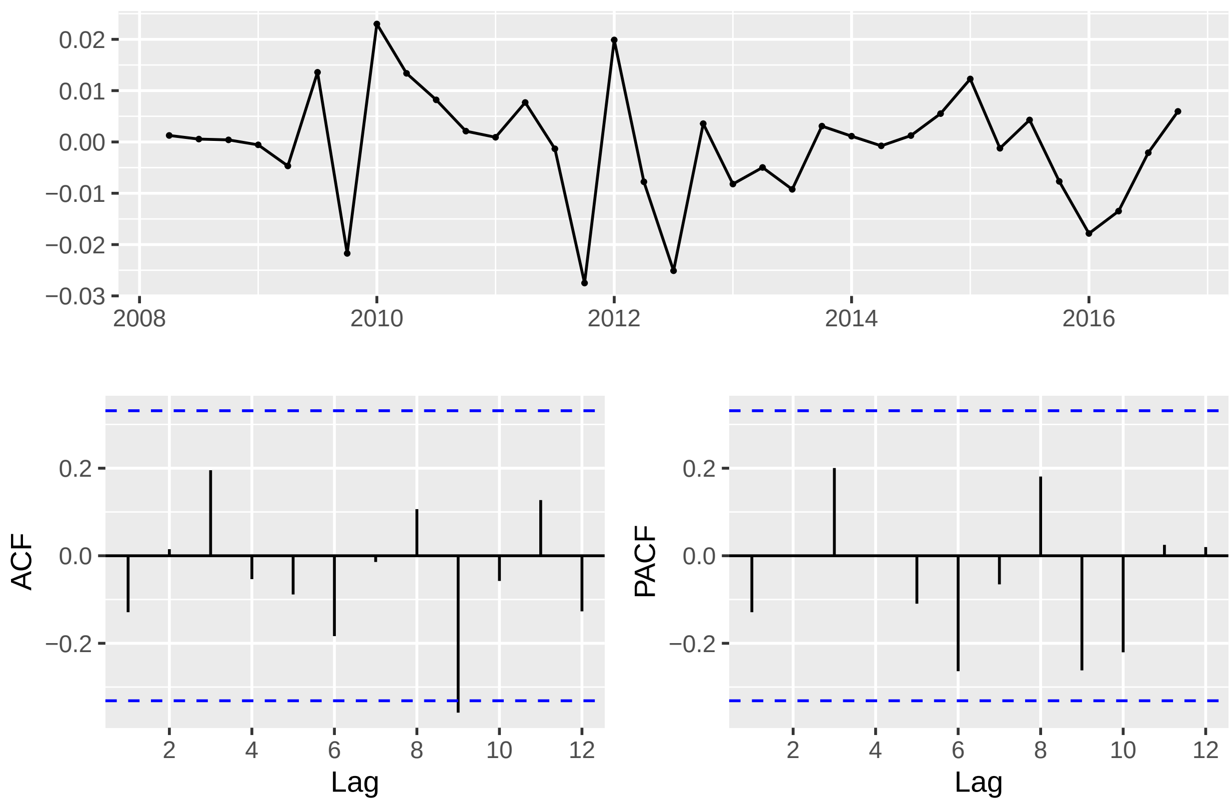
```
##          [,1]     [,2]
## [1,] 11.13163 11.26895
## [2,] 10.73641 10.87608
## [3,] 10.56101 10.72344
```

```
arimafit_log <- ets(window(tsVentas_log, end = 2016 + 3/4))
fventas.arima_log = forecast(arimafit_log , h = cOmit)

forecast::accuracy(fventas.arima_log$mean, window(tsVentas_log, start = 2017), test = NULL, d = NULL, D
```

```
##                 ME       RMSE       MAE       MPE      MAPE        ACF1
## Test set 0.0772417 0.09396171 0.0772417 0.6951939 0.6951939 -0.06175923
##          Theil's U
## Test set 0.1433107
```

Análisis de residuos .

```r
ggtsdisplay(fit_arima_log$residuals)
```



4. Conclusiones

A la hora de examinar las conclusiones, si analizamos los resultados de manera gráfica, se puede observar sin duda alguna que el modelo ETS ha sido capaz de predecir mejor los flujos futuros de ingresos, en cambio, el modelo ARIMA (tanto tomando valores logaritmicos como sin ellos) presenta una predicción, que aún siendo capaz de analizar la tendencia de los ingresos, no llega a ajustarse tanto como el modelo ETS.

Por otra parte, hemos podido observar, tanto en el modelo ETS como en el ARIMA, que realizar el logaritmo de los ingresos con el fin de evitar la estacionalidad de la varianza ha conseguido ajustar aún más los modelos haciendolos más precisos.

Por lo tanto, podemos asumir, que tanto el mejor modelo ETS como ARIMA, son los logarítmicos.

5. Bibliografía

Apuntes proporcionados por el profesor

ETS Models:

- https://robjhyndman.com/talks/RevolutionR/6-ETS.pdf
- https://otexts.com/fpp2/ets-forecasting.html

ARIMA Models:

- http://www.fuenterrebollo.com/Master-Econometria/ECONOMETRIA/SERIES-TEMPORALES/modelo-arima.pdf
- https://es.wikipedia.org/wiki/Modelo_autorregresivo_integrado_de_media_m%C3%B3vil