ETS, ARIMA and Fourier predictions

Guillermo Bustos-Pérez

9/8/2020

Cargar los datos y paquetes necesarios

Los datos fueron procesados en la Parte I. Constituyen la **suma mensual de ventas de diferentes tiendas de una empresa** de febrero de 2013 a diciembre de 2017. La serie exhibe tres patrones importantes a tener en cuenta a la hora de hacer predicciones:

- Hay una marcada estacionalidad en la serie con un primer máximo de ventas en el meses de agosto y mínimos en los meses de enero, febrero y marzo . El descenso de ventas a partir de septiembre se interrumpe con un repunte de incremento de ventas en diciembre.
- La tendencia general de la serie es ascendente.
- El nivel de las fluctuaciones se incrementa a medida que avanza la serie en el tiempo (heterocedasticos).

```
# Load packages
library(tidyverse); library(forecast); library(fpp2)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Warning: package 'forecast' was built under R version 4.0.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Warning: package 'fpp2' was built under R version 4.0.2
```

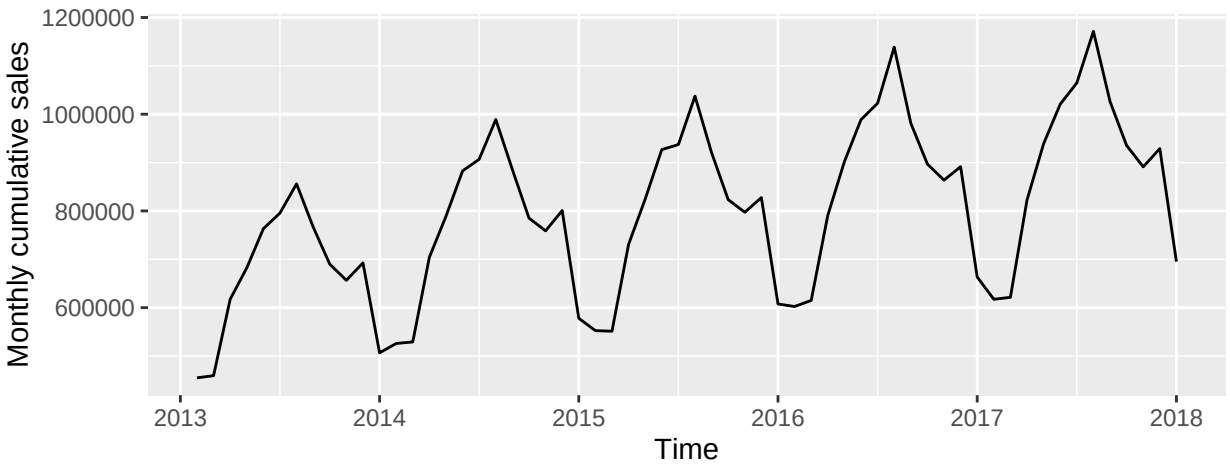
```
## Loading required package: fma
```

```
## Warning: package 'fma' was built under R version 4.0.2
```

```
## Loading required package: expsmooth
```

```
## Warning: package 'expsmooth' was built under R version 4.0.2
```

```r
# Make into a ts object
M_Sales <- ts(M_Sales,
              start = c(2013, 2),
              frequency = 12)

# autoplot
autoplot(M_Sales) + ylab("Monthly cumulative sales")
```



Prólogo a las predicción sobre series temporales

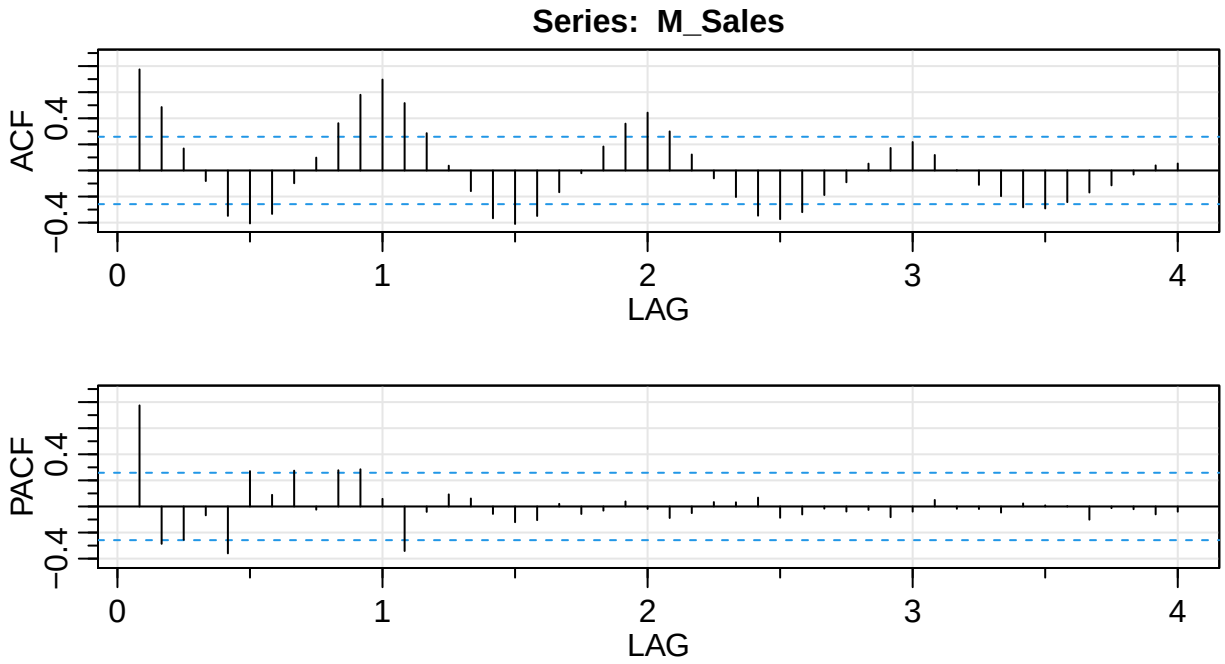
Por el carácter de la serie temporal queda claro que los modelos **ETS**, **ARIMA estacional** y **regresión armónica dinámica** son los que van a proporcionar mejores ajustes y resultados de predicción. Aún así es conveniente realizar la prueba Ljung-Box para asegurarnos de que no se trata de **White Noise** e inspeccionar los plots ACF y PACF.

En el caso de la prueba Ljung-Box nos indica que no se trata de ruido blanco (también podemos observarlo en el ACF). En el caso del ACF y el PACF nos están indicando la correlación estacional que disminuye con el tiempo.

```r
# Ljung-Box test
Box.test(M_Sales,
lag = 36,
fitdf = 0,
type = "Lj")
```

```
##
##  Box-Ljung test
##
## data:  M_Sales
## X-squared = 342.79, df = 36, p-value < 2.2e-16
```

```r
# Check ACF and PACF plots
acf_ts <- astsa::acf2(M_Sales)
```



Antes de realizar las predicciones recordamos la norma básica para Machine Learning o series temporales: es necesario probar los modelos sobre datos no empleados para entrenarlo (dividir entre training y test set). En este caso vamos a emplear el último año registrado (2017) como test set, mientras que el resto se emplearán como training set.

```
# Divide into train and test sets
train <- window(M_Sales,
                end = c(2016, 12))
test <- window(M_Sales,
               start = c(2017, 1))
```

Dado que los datos son heterocedasticos y hay una tendencia ascendente, es necesario aplicar una transformación previa a la realización de predicciones. La función **BoxCox.lambda()** devuelve el parámetro λ que se debe aplicar para estabilizar la serie temporal. En este caso el valor de λ devuelto es cercano a $1/3$, que es similar a una raíz cúbica.

```
# Transformation necessary to stabilize the time series
(BC_l <- BoxCox.lambda(train))
```

```
## [1] 0.312658
```

Predicción de ventas mensuales.

Entrenar y comprobar los modelos

El proceso para comparar los modelos es sencillo. Basta con comprobar la distribución de los residuals y las medidas de precisión de las estimaciones sobre el test set.

Recordamos: en el caso de los residuals queremos que su distribución se asemeje al White-Noise ($p > 0.05$). En el caso de las medidas de precisión generalmente se proporcionan la RMSE (que tiende a penalizar desviaciones más elevadas) y la MAE, aunque es adecuado proporcionar MAPE y la MASE (que proporcionan errores porcentuales y escalados).

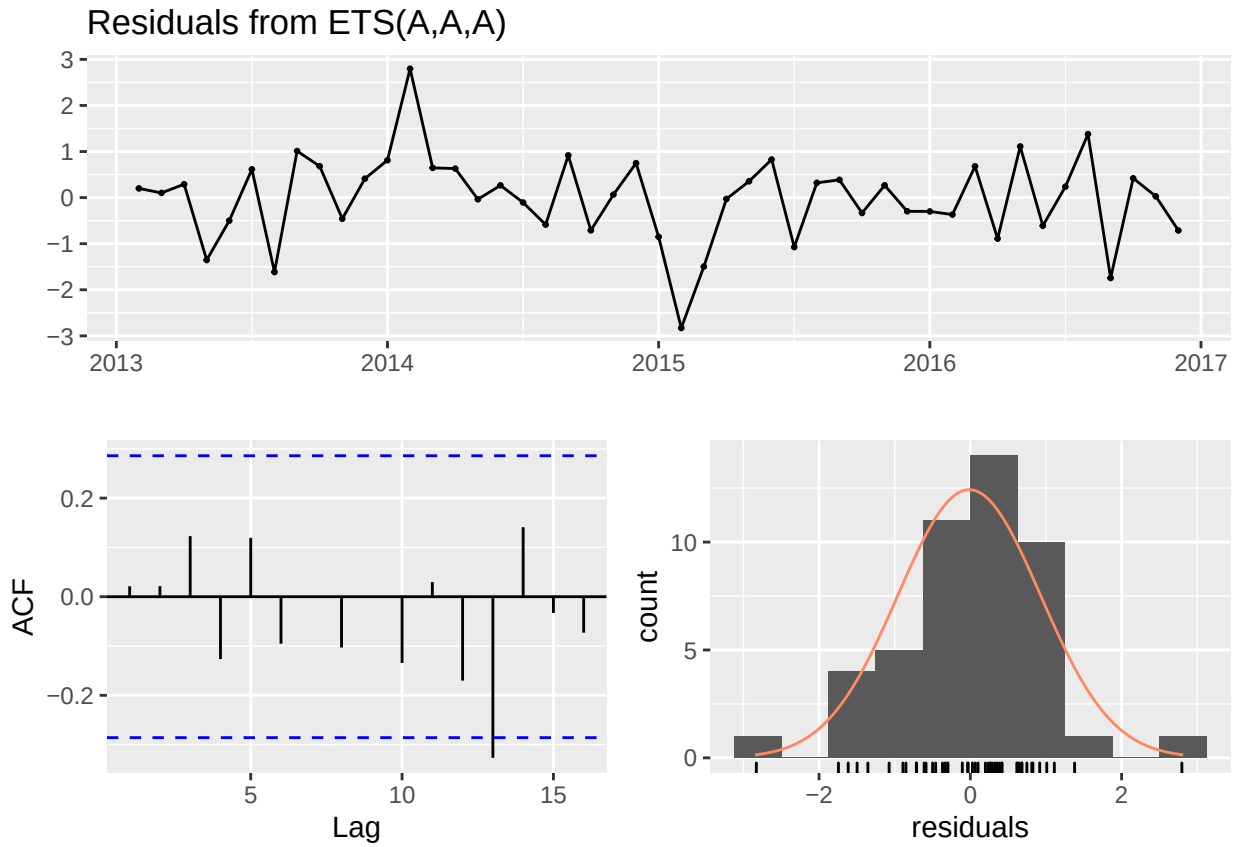
Empezamos entrenando un modelo **ets** en el que se proporciona a λ el valor proporcionado por la transformación BoxCox. En el caso del modelo de regresión armónica dinámica lo vamos a dejar para el final, ya que requiere de varias iteraciones para determinar el mejor valor de K.

```
# Train ETS model
(ETS_model <- ets(train, lambda = BC_l))
```

```
## ETS(A,A,A)
##
## Call:
##  ets(y = train, lambda = BC_l)
##
##   Box-Cox transformation: lambda= 0.3127
##
##   Smoothing parameters:
##     alpha = 0.7074
##     beta  = 1e-04
##     gamma = 0.0017
##
##   Initial states:
##     l = 204.358
##     b = 0.4992
##     s = -19.5922 1.5637 -0.9438 2.1842 10.2551 19.9778
##           13.5098 11.8837 4.6689 -2.8329 -20.318 -20.3562
##
##   sigma:  1.1489
##
##      AIC     AICc      BIC
## 208.4487 229.5521 239.9012
```

Al comprobar los residuals vemos que no pasan la prueba Ljung-Box para determinar si son White-Noise. Sin embargo, las distribuciones observadas en el ACF plot y el histograma de distribución indican que el modelo puede ser útil a pesar de no superar la prueba Ljung-Box.

```
checkresiduals(ETS_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,A)
## Q* = 16.633, df = 3, p-value = 0.0008408
##
## Model df: 16.   Total lags used: 19
```

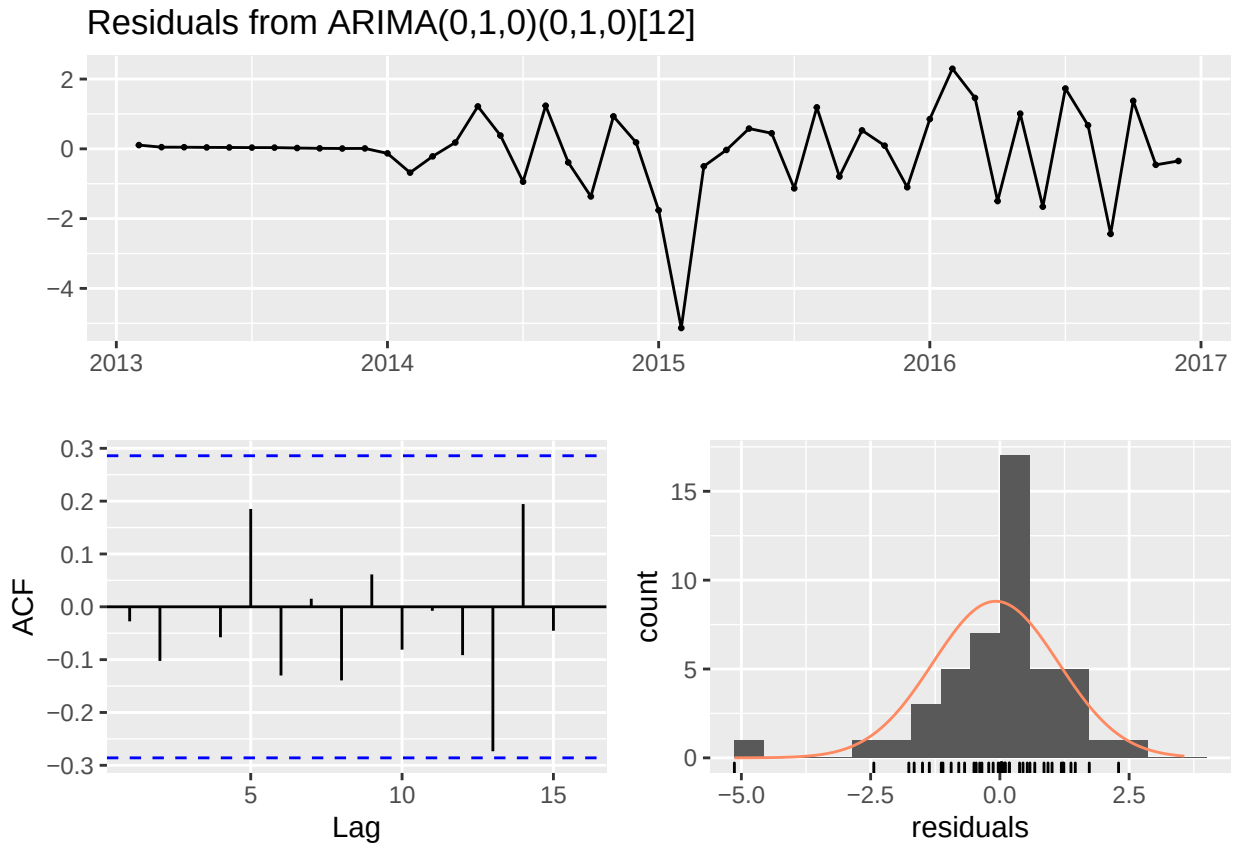
En el caso del modelo **ARIMA** es fácil su entrenamiento con la función **autoarima()**. En este caso podemos ver que la función ha seleccionado un ARIMA (0,1,0)(0,1,0) con una estacionalidad de 12.

También observamos que se han minimizado aún más los residuals y que en este caso sí superan la prueba Ljung-Box ($p > 0.05$). Esto también es observable en el gráfico ACF, estando el Lag 13 dentro de los límites de significación.

```
(ARIMA_model <- auto.arima(train, lambda = BC_l))
```

```
## Series: train
## ARIMA(0,1,0)(0,1,0)[12]
## Box Cox transformation: lambda= 0.312658
##
## sigma^2 estimated as 2.011:  log likelihood=-60
## AIC=121.99   AICc=122.12   BIC=123.52
```

```
checkresiduals(ARIMA_model)
```



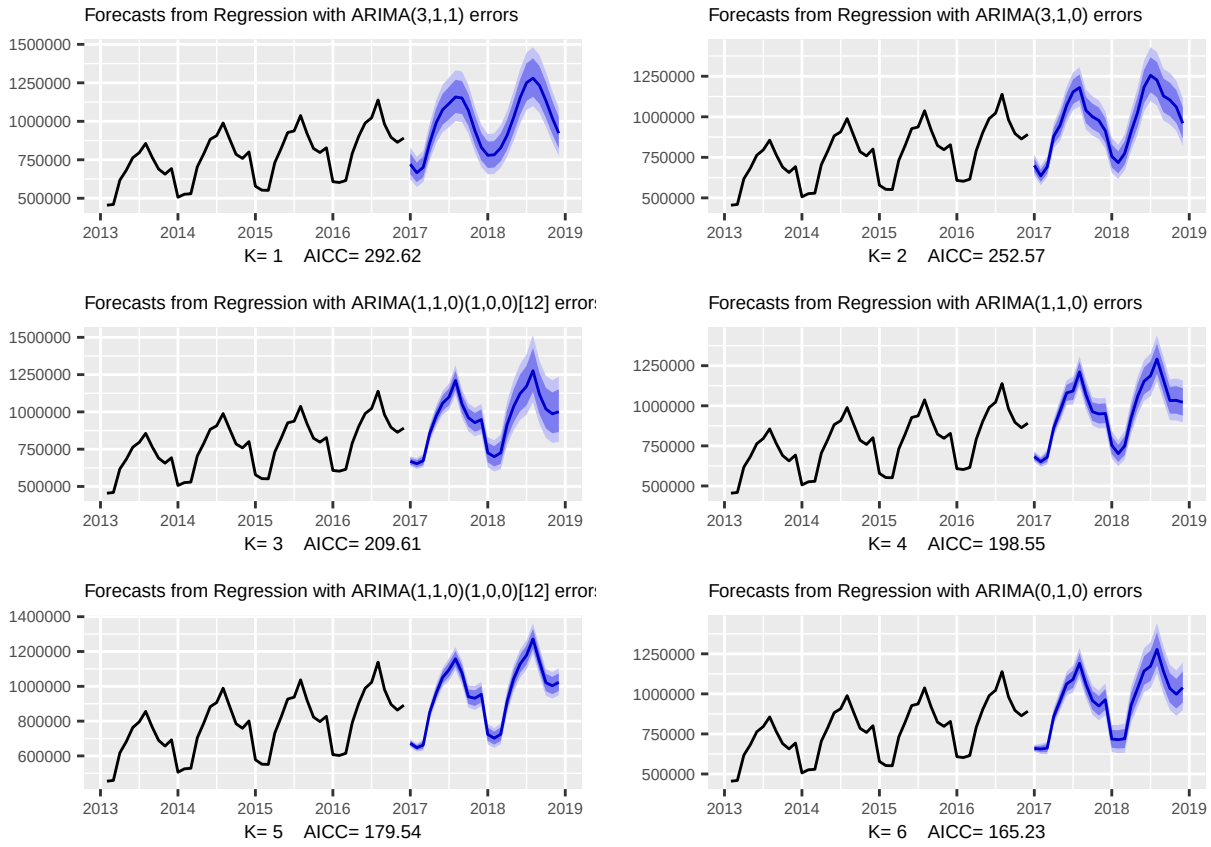

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0)(0,1,0)[12]
## Q* = 4.9683, df = 9, p-value = 0.8371
##
## Model df: 0.   Total lags used: 9
```

Ahora podemos entrenar un modelo de regresión armónica dinámica. En este caso, al ser una serie temporal con estacionalidad 12 limita a un máximo de 6 (mitad del ciclo estacional) los valores de K (número de senos y cosenos de Fourier).

El siguiente código (Hyndman & Athanasopoulos, 2018) nos permite evaluar el AICc de regresiones armónicas dinámicas con diferentes valores de K. AICc se minimiza para K = 6.

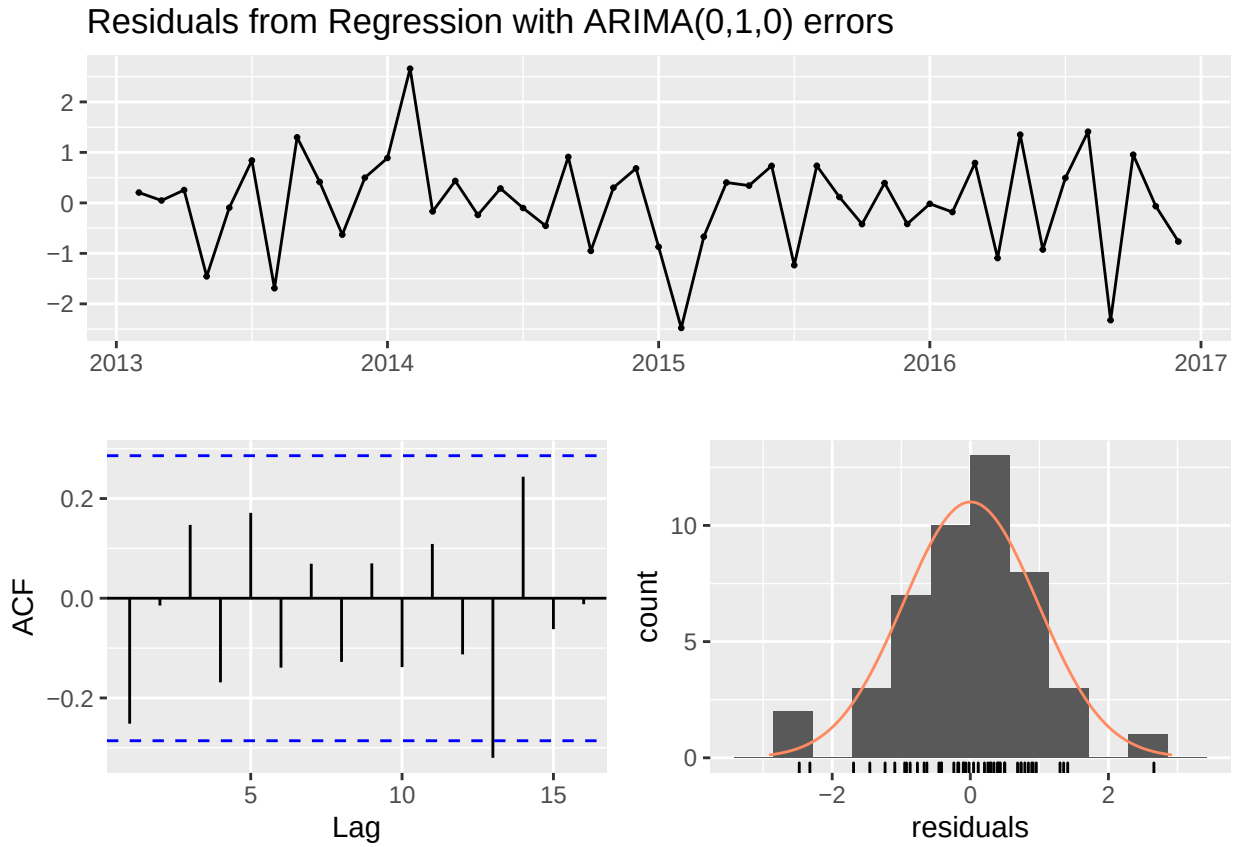
```
plots <- list()
for (i in seq(from = 1, to = 6, by = 1 )) {
  fit <- auto.arima(train, xreg = fourier(train, K = i),
                    seasonal = TRUE, lambda = BC_l)
  plots[[i]] <- autoplot(forecast(fit,
                                  xreg=fourier(train, K = i, h = 24))) +
    xlab(paste("K=",i,"   AICC=",round(fit[["aicc"]],2))) +
    ylab("") +
    theme(
```

```
      axis.text = element_text(size = 6),
      axis.title.x = element_text(size = 7),
      title = element_text(size = 6)
    )
}
gridExtra::grid.arrange(
  plots[[1]],plots[[2]],plots[[3]],
  plots[[4]],plots[[5]],plots[[6]], nrow=3)
```



Una vez determinado el valor de K, podemos entrenar el modelo y comprobar los residuals. Al igual que en modelo ETS, los residuals de la regresión armónica dinámica no superan la prueba Ljung-Box del W-N, aunque nuevamente hay que señalar que únicamente el Lag 13 es el que muestra una significación. Por consiguiente, podemos considerar el modelo como aprovechable a pesar de estos inconvenientes.

```
Four_model <- auto.arima(train, xreg = fourier(train, K = 6),
                  seasonal = TRUE, lambda = BC_l)

checkresiduals(Four_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,1,0) errors
## Q* = 24.215, df = 3, p-value = 2.252e-05
##
## Model df: 12.   Total lags used: 15
```

Elección del mejor modelo

Ahora basta con realizar las predicciones para el año siguiente y compararlas con los valores reales del test set.

```
# Predicciones del modelo ETS
ETS_forecast <- ETS_model %>%
  forecast(h = 12)
a1 <- accuracy(ETS_forecast, test)

# Predicciones del modelo ARIMA
ARIMA_forecast <- ARIMA_model %>%
  forecast(h = 12)
a2 <- accuracy(ARIMA_forecast, test)

# Predicciones de la AHR
Four_forecast <- Four_model %>%
  forecast(xreg = fourier(train, K = 6, h = 12))
a3 <- accuracy(Four_forecast, test)

# Cheeck accuracy of each model
a1[,c("RMSE","MAE", "MAPE","MASE")]
```

```
##                    RMSE       MAE     MAPE      MASE
## Training set  9919.683  7691.846 1.035925 0.1140702
## Test set     37036.438 34901.123 4.048188 0.5175843
```

```
a2[,c("RMSE","MAE", "MAPE","MASE")]
```

```
##                   RMSE      MAE     MAPE      MASE
## Training set 13051.70  8961.77 1.157055 0.1329032
## Test set     34113.01 32425.00 3.740908 0.4808634
```

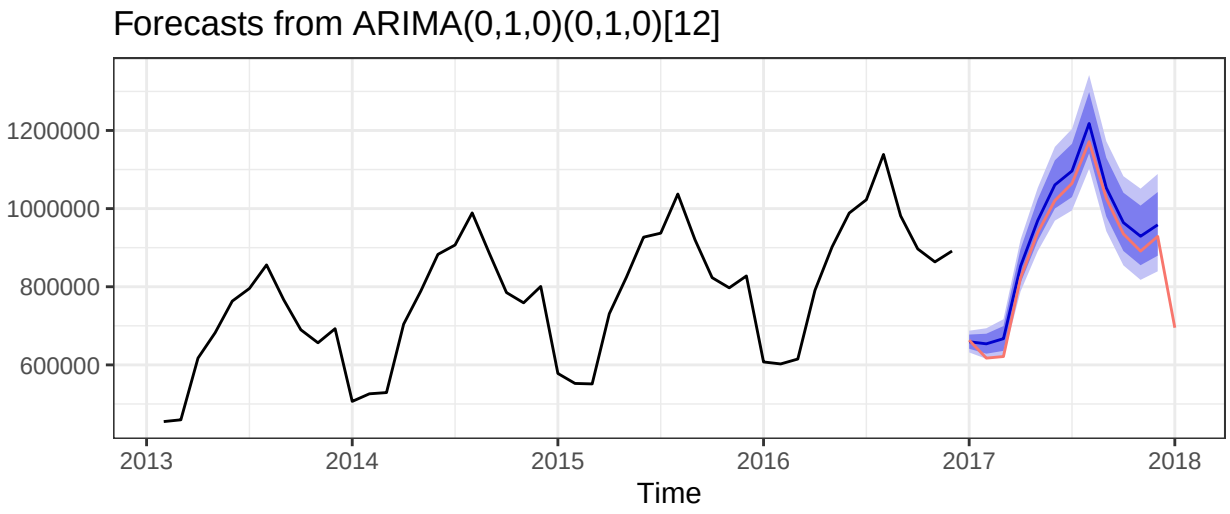
```
a3[,c("RMSE","MAE", "MAPE","MASE")]
```

```
##                   RMSE       MAE     MAPE      MASE
## Training set 10708.42  8262.236 1.072590 0.1225291
## Test set     30865.04 29006.332 3.412829 0.4301644
```

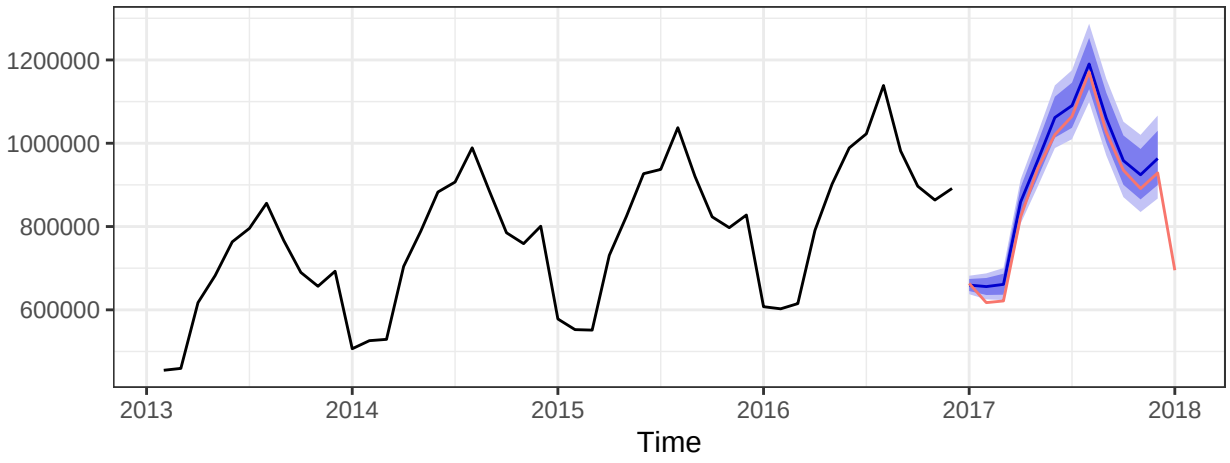
El modelo ETS no proporciona mejoras o diferencias sustanciales con respecto al ARIMA o la regresión armónica dinámica. Vamos a acompañar estas medidas con los gráficos de series temporales que comparan las predicciones según el modelo ARIMA o RAD y los datos reales del test set.

Para ser sinceros, podemos ver que no existe demasiada diferencia entre las predicciones del modelo ARIMA y la regresión armónica dinámica.

```
# Arima forecast and test set
autoplot(ARIMA_forecast) +
  autolayer(test) +
  theme_bw()  +
  theme(legend.position = "none",
        axis.title.y = element_blank())
```




```
# ADR forecast and test set
Four_model %>% forecast(xreg = fourier(train, K = 6, h = 12)) %>%
  autoplot() + autolayer(test) +
  theme_bw() +
  theme(legend.position = "none",
        axis.title.y = element_blank())
```



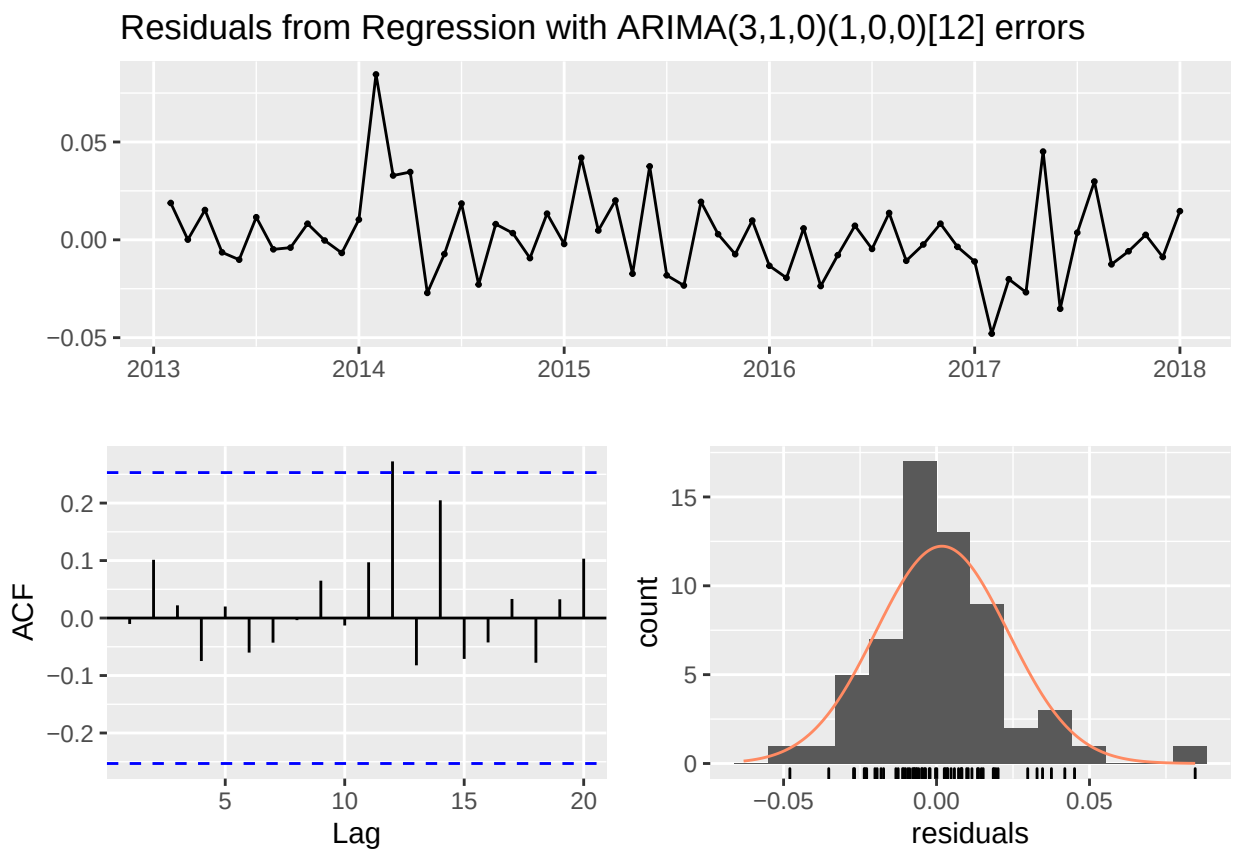
De acuerdo a los datos previamente vistos, la regresión armónica dinámica con K = 6 supone el mejor modelo al minimizar los valores de error (aunque, como ya hemos señalado, no parece haber gran diferencia con respecto al modelo ARIMA).

Entrenamos nuevamente el modelo, pero aprovechando todos los datos disponibles. Nuevamente comprobamos los residuals. Aunque no superan la prueba del WN hemos podido comprobar que sigue siendo un modelo válido para el desarrollo de predicciones.

```
(BC_l <- BoxCox.lambda(M_Sales))
```

```
## [1] 0.0489577
```

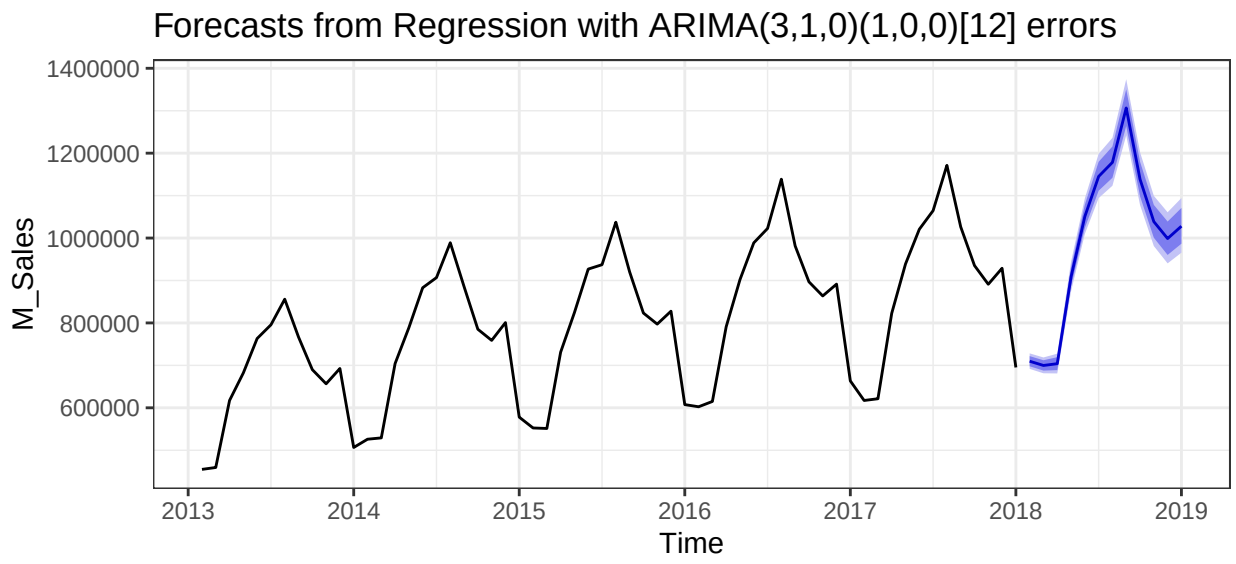
```
Four_model <- auto.arima(M_Sales, xreg = fourier(M_Sales, K = 6),
                  seasonal = TRUE, lambda = BC_l)

checkresiduals(Four_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,1,0)(1,0,0)[12] errors
## Q* = 13.487, df = 3, p-value = 0.003693
##
## Model df: 16.   Total lags used: 19
```

Obtenemos la representación de predicciones de ingresos mensuales que parece razonable conociendo el carácter estacional y la tendencia de la serie.

```
Four_model %>% forecast(xreg = fourier(train, K = 6, h = 12)) %>%
  autoplot()+
  theme_bw()
```



Por último podemos obtener las previsiones de ingresos mensuales para los próximos 12 meses. Estos proporcionan el punto de predicción junto con los intervalos superiores e inferiores a 80 y 95.

```
# Numeric forecast with arima dynamic harmonic regression
Four_model %>% forecast(xreg = fourier(train, K = 6, h = 12))
```

```
##          Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## Feb 2018       709778.3  697926.9  721821.0  691729.5  728274.4
## Mar 2018       699776.2  687708.1  712045.4  681400.1  718622.9
## Apr 2018       703915.1  688861.9  719280.7  681017.7  727543.4
## May 2018       908540.2  886099.8  931520.3  874435.1  943907.9
## Jun 2018      1050773.9 1023763.1 1078461.5 1009733.1 1093397.9
## Jul 2018      1145173.1 1111610.4 1179698.2 1094224.2 1198373.1
## Aug 2018      1178400.1 1142075.9 1215821.3 1123280.7 1236085.6
## Sep 2018      1306540.3 1264016.0 1350422.9 1242041.2 1374216.8
## Oct 2018      1138477.2 1098518.0 1179816.3 1077908.5 1202273.9
## Nov 2018      1038756.3 1000777.4 1078103.4  981210.9 1099502.2
## Dec 2018       998818.0  960233.3 1038874.5  940385.8 1060692.8
## Jan 2019      1028005.7  986749.8 1070898.6  965552.7 1094288.4
```

Ya que estamos, podemos obtener las predicciones numéricas empleando el modelo ARIMA. Con ello confirmamos que no difieren demasiado de las del modelo de regresión armónica dinámica.

```
# Numeric forecast with arima model
M_Sales %>% auto.arima(lambda = BC_l) %>%
  forecast(h = 12)
```

```
##          Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## Feb 2018       673599.7  658600.6  688923.3  650789.7  697168.7
## Mar 2018       686972.6  669543.1  704833.0  660487.6  714465.6
## Apr 2018       887947.2  861363.1  915310.5  847598.5  930118.4
```

```
## May 2018      1012724.8  979299.9 1047232.9  962033.3 1065950.2
## Jun 2018      1106947.8 1066930.9 1148389.4 1046308.0 1170920.4
## Jul 2018      1147396.8 1102721.8 1193789.7 1079747.0 1219065.2
## Aug 2018      1272857.1 1220119.5 1327758.1 1193049.9 1357725.1
## Sep 2018      1102715.2 1053946.1 1153625.7 1028966.7 1181473.2
## Oct 2018      1007316.3  960169.8 1056659.3  936069.2 1083701.8
## Nov 2018       967534.5  919974.3 1017427.2  895706.1 1044819.8
## Dec 2018      1001074.1  949789.3 1054985.5  923662.3 1084631.0
## Jan 2019       747245.9  706866.6  789812.8  686339.9  813270.0
```

Bibliografía

Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on <2020/08/05>

Shumway, R.H., & Stoffer, D.S. (2011) Time Series Analysis and Its Applications. With R Examples. 3rd edition. Springer Texts in Statistics. New York. 202 pgs.