

ejemplo de xreg

Xavi Barber

28/5/2018

ARIMA y xreg

Nuestros datos

	Customer_Visit	Weekday	Christmas	Day
1	2947	1	0	1
2	3003	2	0	2
3	3006	3	0	3
4	2997	4	0	4
5	2983	5	0	5
6	2970	6	0	6
7	3009	7	0	7
8	2987	1	0	8
9	3062	2	0	9
10	2992	3	0	10
11	2949	4	0	11
12	2915	5	0	12
13	2937	6	0	13
14	2974	7	0	14
15
40	3034	5	0	40
41	2952	6	1	41
42	2943	7	0	42
43	3027	1	0	43
44	3069	2	0	44
45	3036	3	0	45
46	2890	4	0	46
47	3012	5	0	47
48	2967	6	0	48
49	2914	7	0	49

Como podeis ver, hay una variable que es “Navidad”, otra variable que es día de la semana pero en números de 1 a 7.

Por lo que habrá que decirle al comando que tenga en cuenta que estas variables son factores.

¿cómo se hace esto?

Nosotros queremos ayudar el siguiente modelo

$$Y_t = \beta_0 + \beta_1 \text{Weekday} + \beta_2 \text{Day} + \beta_3 \text{Christmas} + \text{ARMA}(p, q)$$

model.matrix

Debemos pasar nuestras variables factor a variables “dummies”. Una variable dummie es una variable que se crea a partir de una variable nominal y que identifica sólo a una de sus categorías. Es decir, tenemos que crear diferentes variables dummie para identificar los 7 días de la semana.

```
diaSemana<-model.matrix(~as.factor(modelfitsample$Weekday))
diaSemana<-diaSemana[,-1]
colnames(diaSemana) <- c("Mon","Tue","Wed","Thu","Fri","Sat")
kable(head(diaSemana, n=10))
```

Como pueden ver, hemos generado 6 variables donde antes sólo había una. Falta el Domingo, pero el sistema toma como Domingo cuando no es ninguno de los otros días, es decir, cuando Mon ... Sat valen cero.

Ya podemos ajustar el modelo:

```
# Creamos las variables del modelo lineal
Variables.xreg <- cbind(Weekday=diaSemana,
                       Day=modelfitsample$Day,
                       Christmas=modelfitsample$Christmas)

# Le decimos a R cual es nuestra variable de "Time Series"
visits <- ts(modelfitsample$Customer_Visit, frequency=7)

# Ajustamos el modelo ARIMAX (ARMA(2,0) a modo de ejemplo)
modArima <- Arima(visits, xreg=Variables.xreg, order=c(2,0,0))
```

Los resultados del ajuste (parámetros y error):

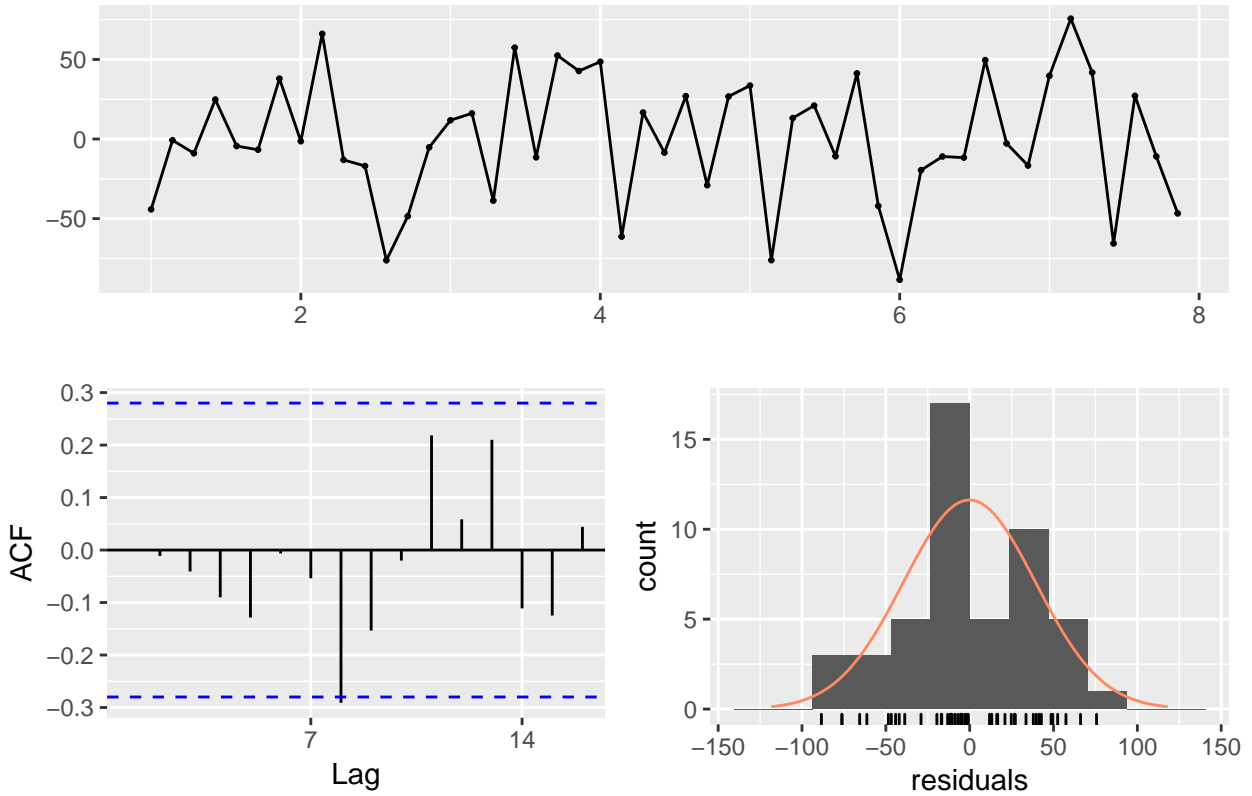
```
summary(modArima)

## Series: visits
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##      ar1      ar2  intercept      Mon      Tue      Wed      Thu
##    -0.0753 -0.1026 2991.6783   9.3461  19.3159 -18.9895 -2.0801
## s.e.   0.1472   0.1450  16.6667  21.7712  22.0738  20.8362  20.8734
##      Fri      Sat      Day  Christmas
##    -11.9253 -20.8399 -0.1526  -15.9929
## s.e.   22.7538  21.7508   0.3491   43.1579
##
## sigma^2 estimated as 1962:  log likelihood=-249.06
## AIC=522.12  AICc=530.79  BIC=544.83
##
## Training set error measures:
##              ME  RMSE  MAE  MPE  MAPE  MASE
## Training set -0.1359757 39.0052 31.61679 -0.02176012 1.06166 0.6567285
##              ACF1
## Training set -0.001764875
```

Comprobaremos si el modelo se ha ajustado bien:

```
checkresiduals(modArima)
```

Residuals from Regression with ARIMA(2,0,0) errors



```
##  
## Ljung-Box test  
##  
## data: Residuals from Regression with ARIMA(2,0,0) errors  
## Q* = 15.632, df = 3, p-value = 0.001349  
##  
## Model df: 11. Total lags used: 14
```

El modelo es mejorable, pero esto es simplemente un ejemplo.

y como resultado, tambien podemos predecir

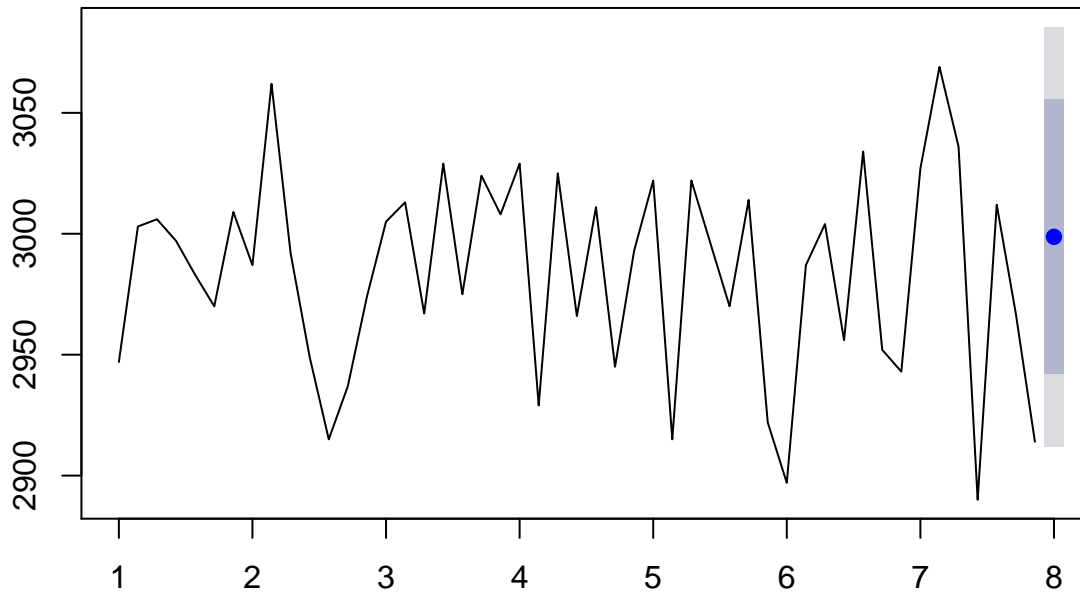
```
## hay que darle valores "X" para que haga la prediccion  
  
#voy a predecir un valor para el lunes sin ser Navidad del dia 43  
# por que tengo Mon=1 Tue=0 Wed Thu Fri Sat=0 Day=53 Xmas=0  
variables.prediccion<-data.frame(1,0,0,0,0,0,43,0)  
  
prediccion<-forecast(modArima, xreg=variables.prediccion )
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
8	2998.74	2941.977	3055.503	2911.928	3085.551
Y dib	ujamos la predic	ión			

Opción 1

```
plot(prediccion)
```

Forecasts from Regression with ARIMA(2,0,0) errors

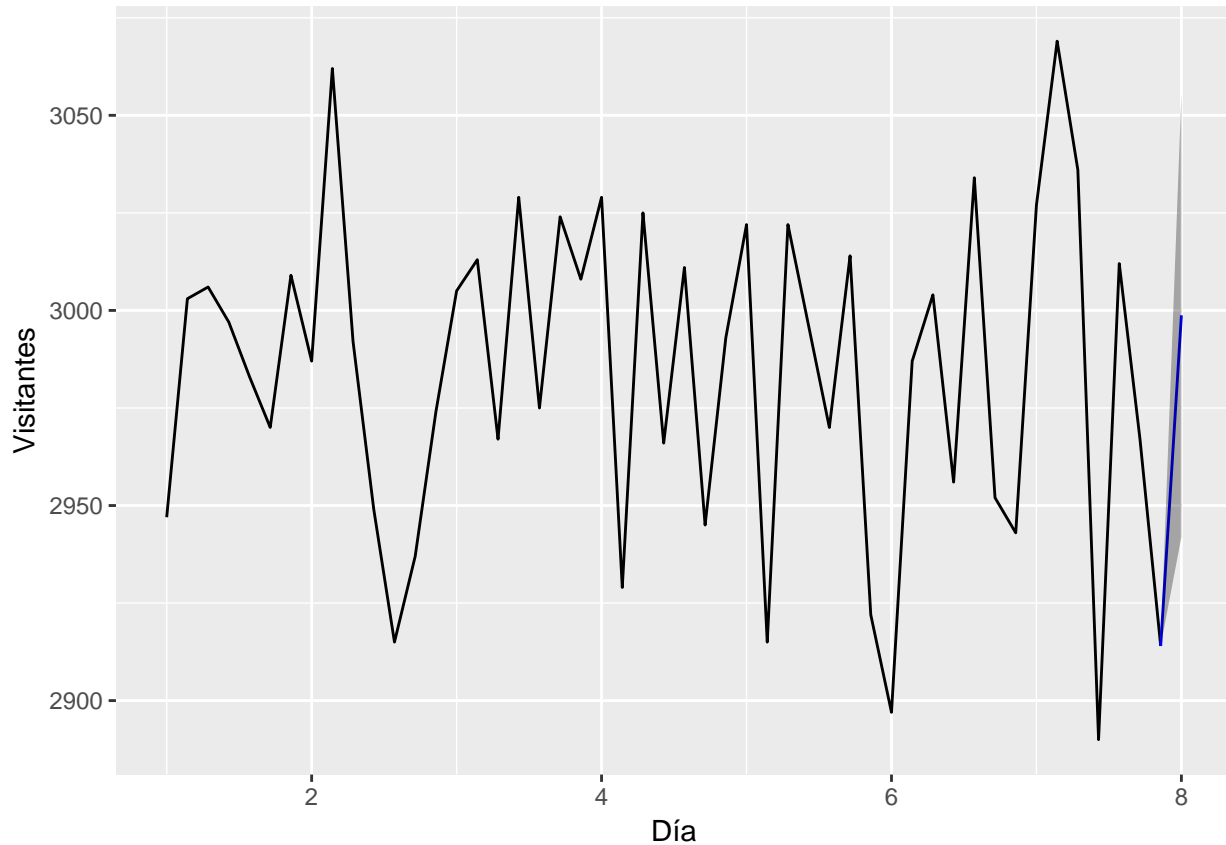


Opción 2

```
library(ggfortify)
```

```
## Loading required package: ggplot2
```

```
autoplot(prediccion) + xlab("Día") + ylab("Visitantes")
```



Más predicciones

Vamos a pedirle más predicciones, según los valores de "X"

```
## hay que darle valores "X" para que haga la predicción

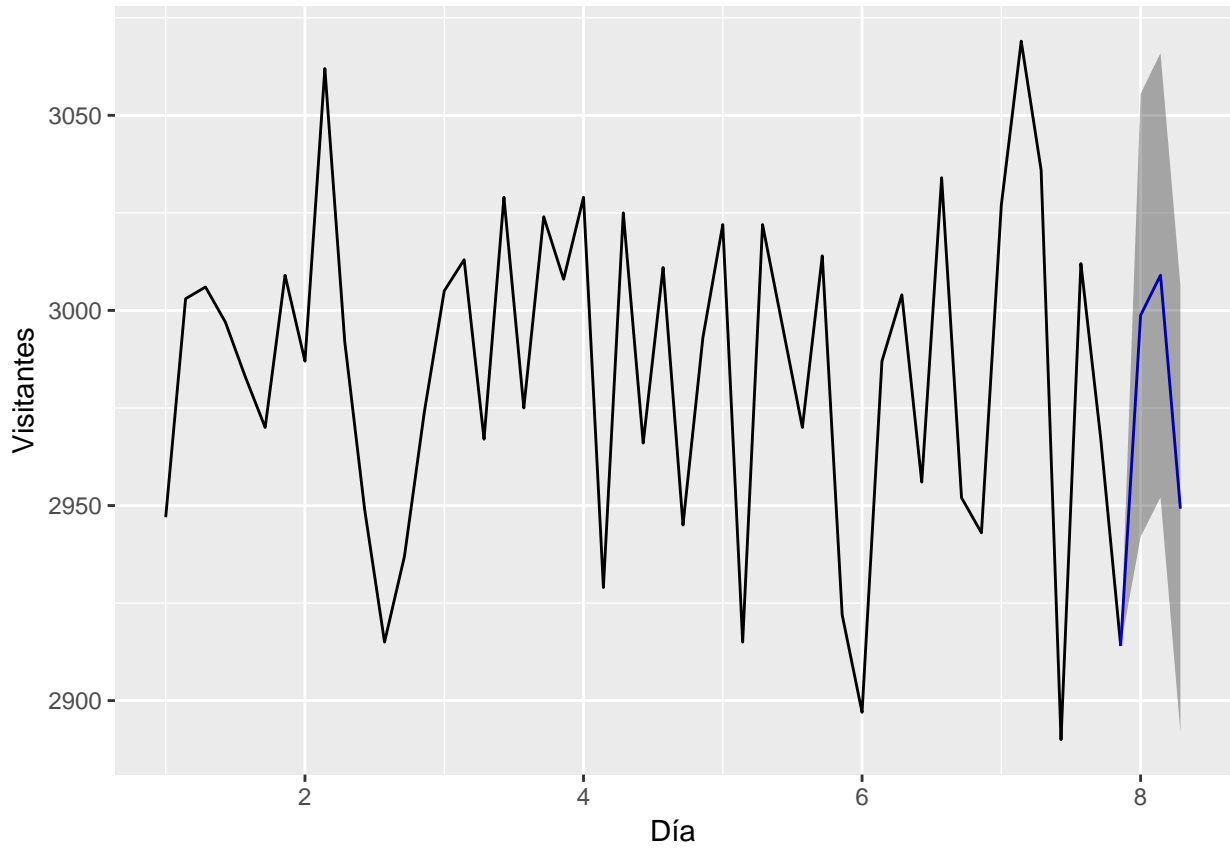
#voy a predecir un valor para el lunes sin ser Navidad del día 43
# por que tengo Mon=1 Tue=0 Wed Thu Fri Sat=0 Day=53 Xmas=0
dia1<-c(1,0,0,0,0,0,43,0)
dia2<-c(0,1,0,0,0,0,44,0)
dia3<-c(0,0,1,0,0,0,44,1) #Navidad

variables.prediccion<-data.frame(rbind(dia1,dia2,dia3))

prediccion<-forecast(modArima, xreg=variables.prediccion )
```

Y dibujamos

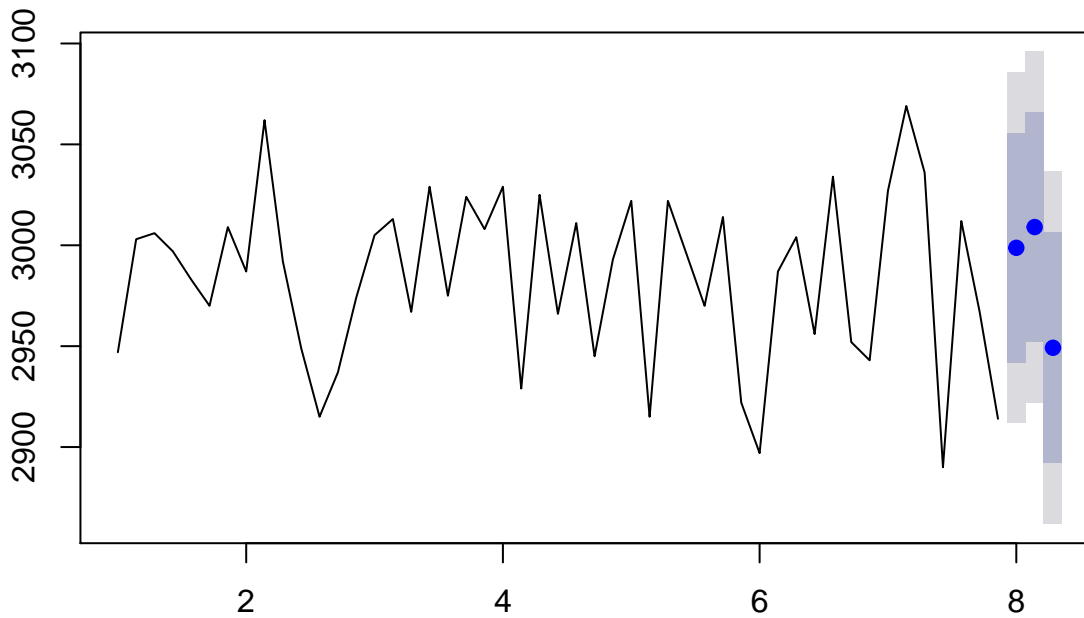
```
autoplot(prediccion) + xlab("Día")+ ylab("Visitantes")
```



o la otra opción

```
plot(prediccion)
```

Forecasts from Regression with ARIMA(2,0,0) errors



Escenarios

Una vez visto esto, podemos estar interesados en generar escenarios Pesimistas, neutros y Optimistas. Para ello lo que debemos de hacer es darle valores la **covariables** según deseemos crear esos escenarios.