Estimación de Demanda Usando R

Objetos xts

Importar datos

Archivo `bev.csv`

- Datos: Estimación Producción de cerveza en EEUU
- Inicio de la serie: 19 de enero de 2014
- Frecuencia: Semanal
- Cantidad de observaciones: 176
- ¿Cómo lo llevo a un formato adecuado?

Importar datos

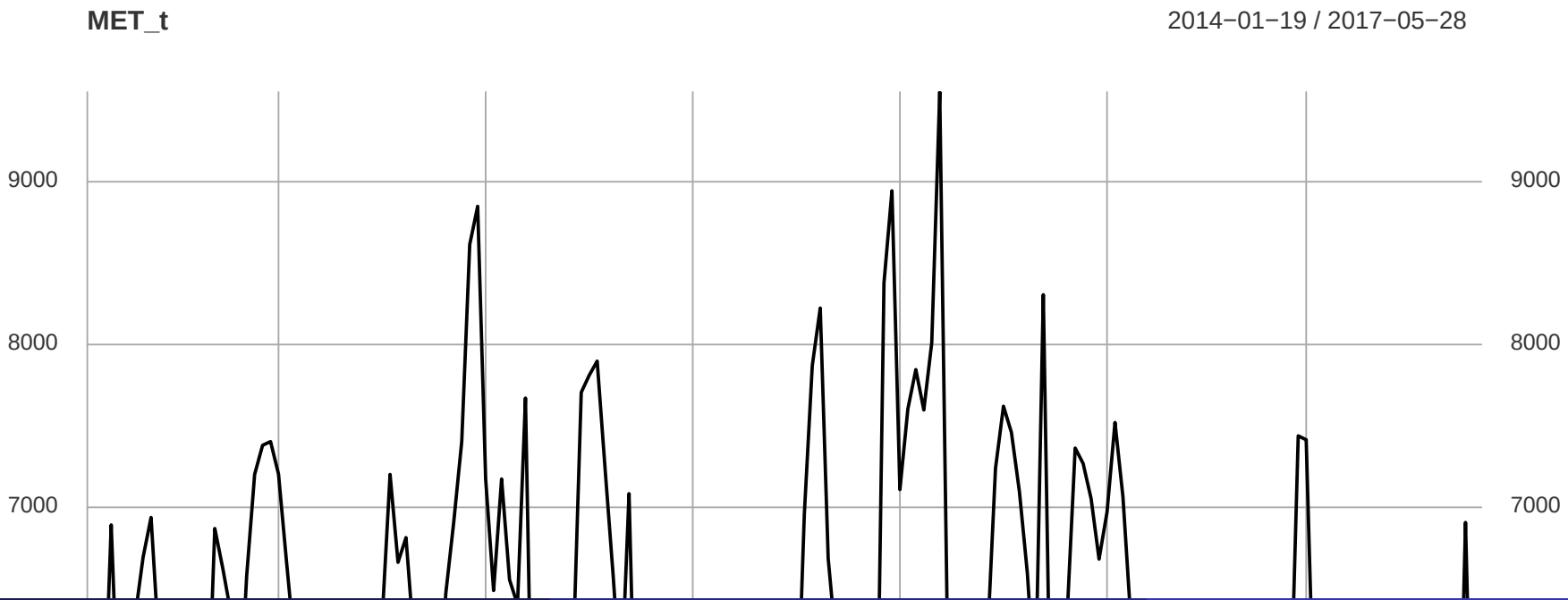
```
library(readr)
bev <- read_csv("bev.csv")

library(xts)
dates <- seq(as.Date("2014-01-19"),
             length = 176, by = "weeks")
bev_xts <- xts(bev, order.by = dates)
```

Visualizar datos

- En los datos MET se refiere a área metropolitana
- High, end y sp (especialidad) se refieren a las líneas de producto
- ¿Qué ocurre con las ventas en el área metropolitana?

Visualizar datos

```
MET_hi <- bev_xts[,"MET.hi"]
MET_lo <- bev_xts[,"MET.lo"]
MET_sp <- bev_xts[,"MET.sp"]

MET_t <- MET_hi + MET_lo + MET_sp

plot(MET_t)
```



auto.arima()

- Proporciona un modelo de partida
- Sobre los datos puedo definir un conjunto de validación (e.g. 2017)

auto.arima()

```
library(forecast)

MET_hi <- bev_xts[,"MET.hi"]
MET_lo <- bev_xts[,"MET.lo"]
MET_sp <- bev_xts[,"MET.sp"]

MET_t <- MET_hi + MET_lo + MET_sp

MET_t <- xts(MET_t$MET.hi, order.by = dates)

MET_t_train <- MET_t[index(MET_t) < "2017-01-01"]
MET_t_valid <- MET_t[index(MET_t) >= "2017-01-01"]

auto.arima(MET_t_train)
```

```
## Series: MET_t_train
## ARIMA(1,0,0) with non-zero mean
```

Interpretación de `auto.arima()`

- El modelo entrega información respecto del comportamiento de los datos
- En este caso nos dice que un proceso AR 1 describe mejor la serie

Interpretación de auto.arima()

```
MET_t_model <- auto.arima(MET_t_train)
MET_t_model
```

```
## Series: MET_t_train
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1       mean
##       0.6706  6252.8350
## s.e.  0.0594   181.7486
##
## sigma^2 estimated as 574014:  log likelihood=-1238.86
## AIC=2483.72   AICc=2483.88   BIC=2492.83
```

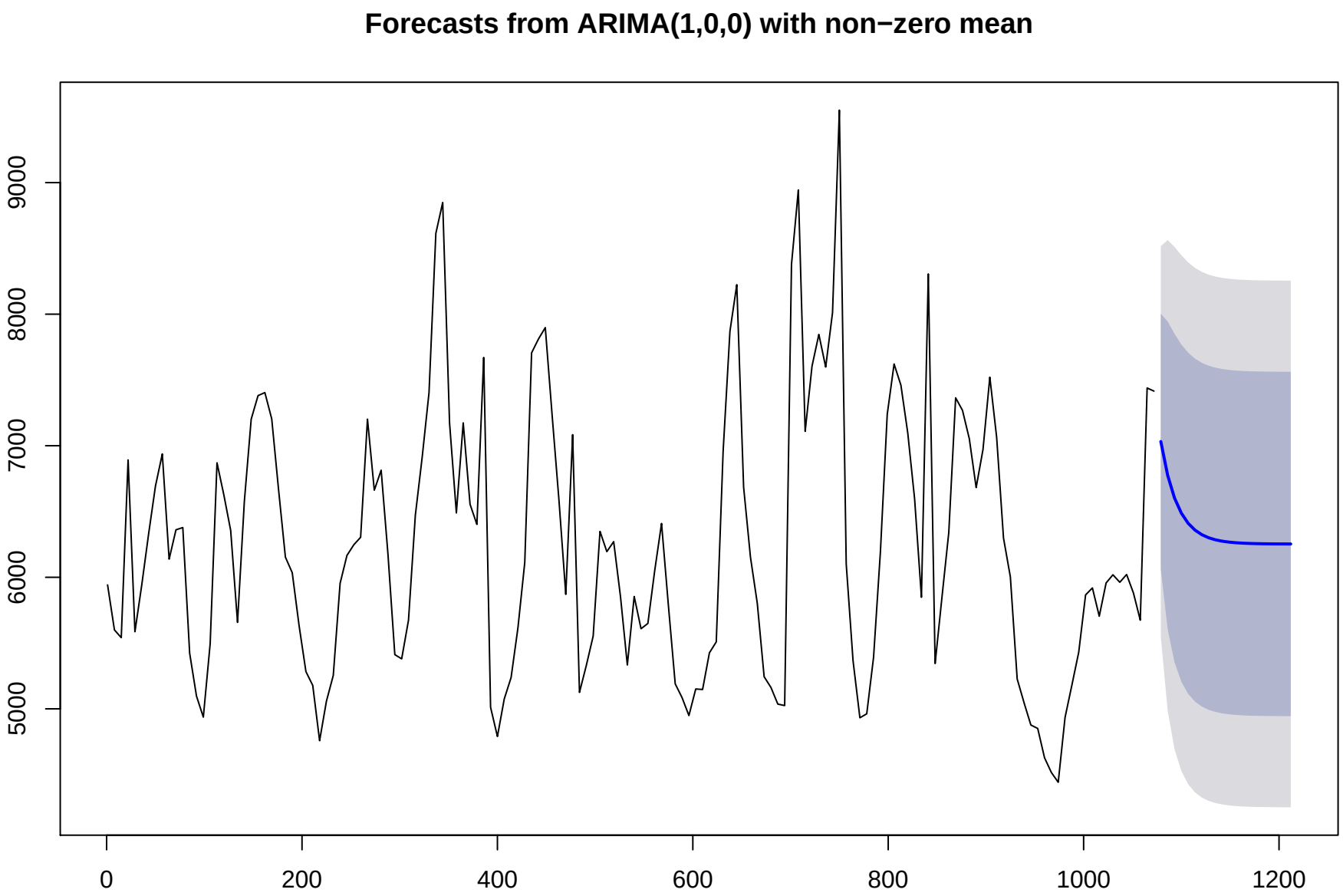
forecast()

- Con lo anterior puedo hacer una proyección
- Con los datos de entrenamiento puedo hacer una estimación a cinco meses (20 semanas) y contrastar

forecast()

```
forecast_MET_t <- forecast(MET_t_model, h = 20)
plot(forecast_MET_t)
```

forecast()


Forecasts from ARIMA(1,0,0) with non−zero mean
9000
8000
7000
6000
5000
0
200
400
600
800
1000
1200

MAPE y MAE

Error absoluto con respecto a la media (MAE)

$$\frac{1}{T}\sum_{t=1}^{T}|Y_t - \hat{Y}_t|$$

Porcentaje de error absoluto con respecto a la media (MAPE)

$$\frac{100}{T}\sum_{t=1}^{T}\left|\frac{Y_t - \hat{Y}_t}{Y_y}\right|$$

MAPE y MAE

```
for_MET_t <- as.numeric(forecast_MET_t$mean)
v_MET_t <- as.numeric(MET_t_valid)

MAE <- mean(abs(for_MET_t - v_MET_t))

MAPE <- 100*mean(abs((for_MET_t - v_MET_t) /v_MET_t))

print(MAE)
```

```
## [1] 957.9956
```

```
print(MAPE)
```

```
## [1] 18.304
```

Visualización de datos (nuevamente)

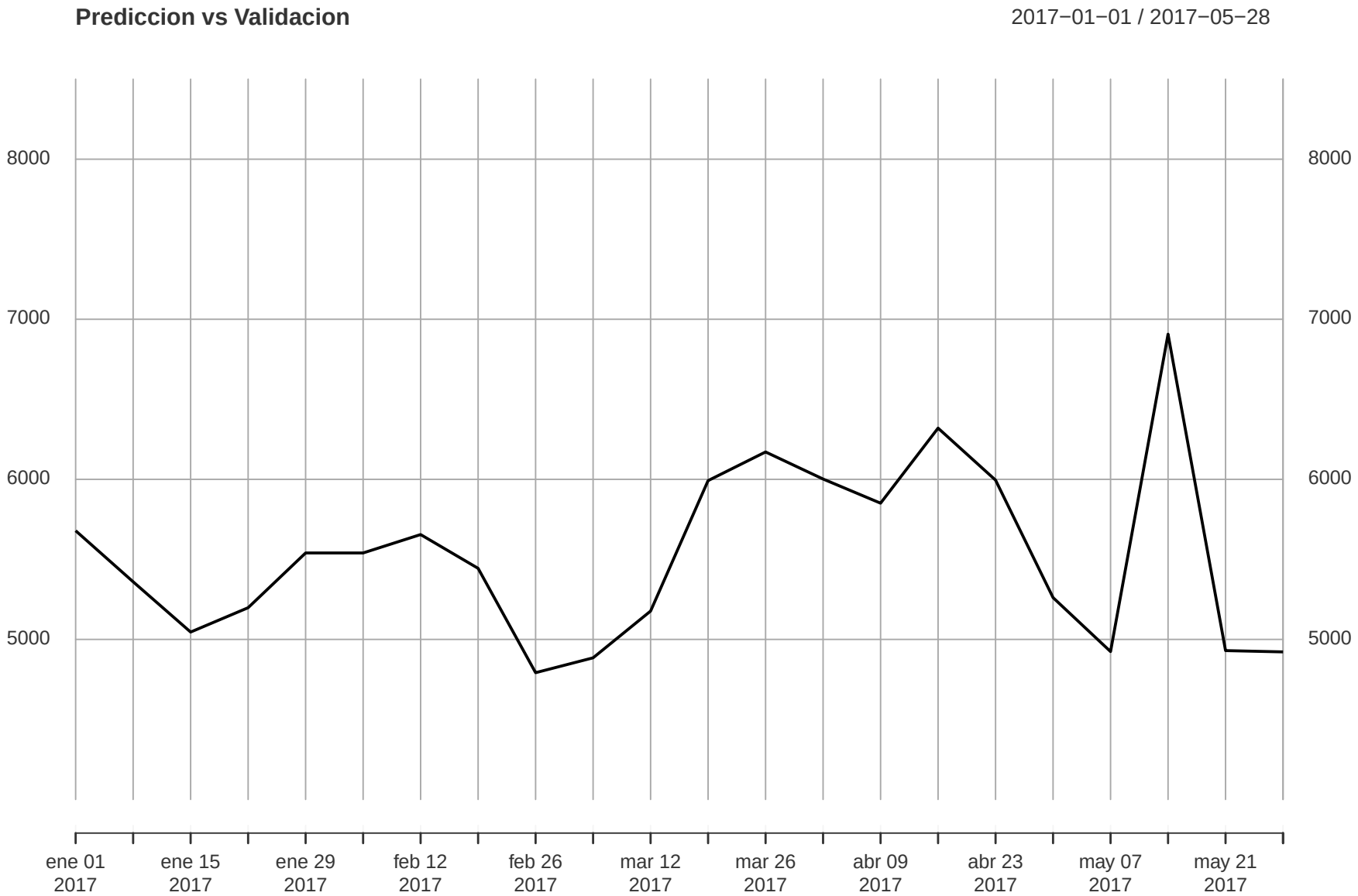
Your forecast seemed to be off by over 18% on average. Let's visually compare your forecast with the validation data set to see if we can see why. Your workspace has your forecast object forecast_MET_t and validation data set MET_t_valid loaded for you.

- La predicción falla en torno a un 20%
- ¿Qué tanto difiere de los datos de validación?

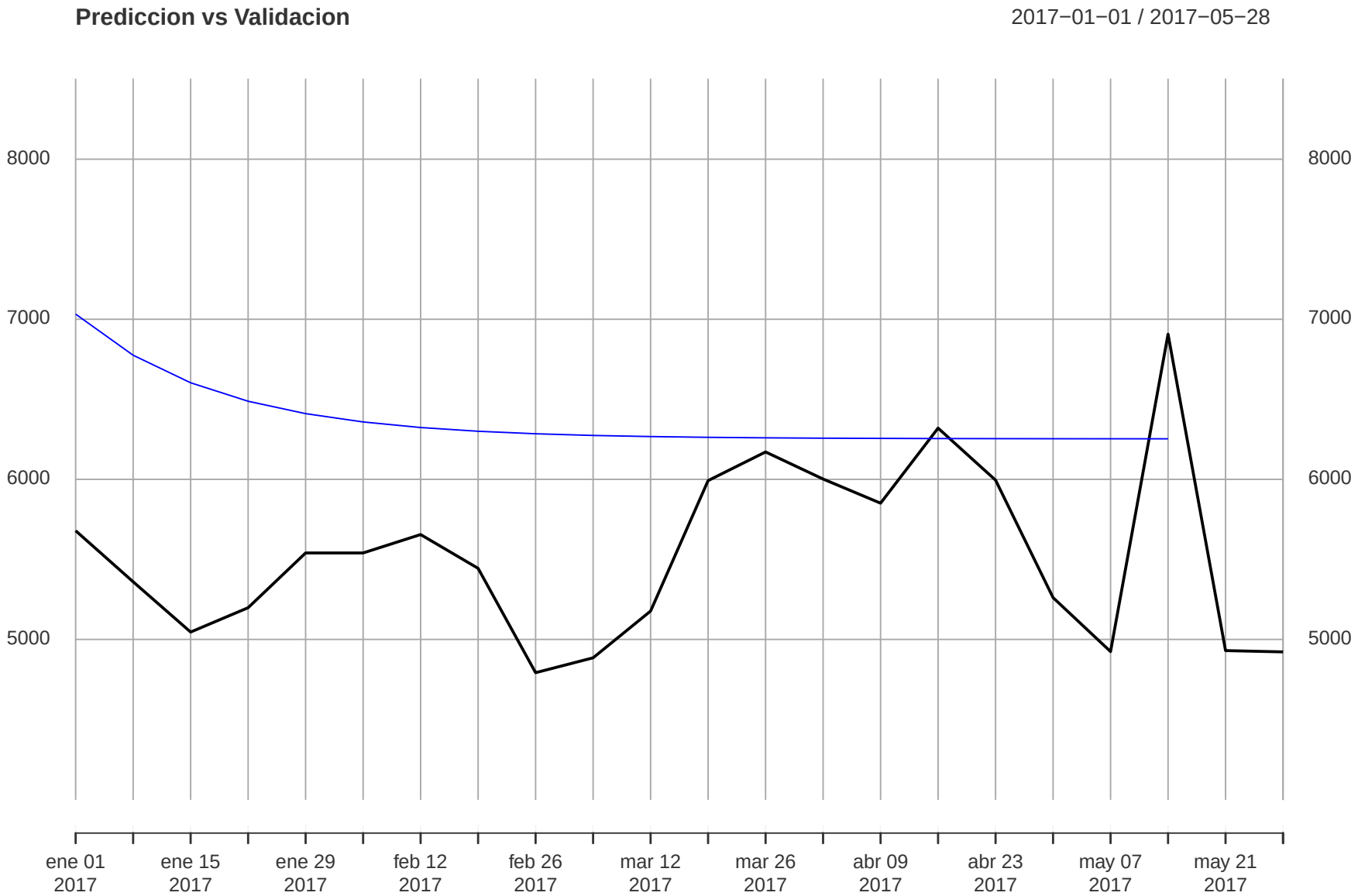
Visualización de datos (nuevamente)

```
for_dates <- seq(as.Date("2017-01-01"), length = 20,
                 by = "weeks")
for_MET_t_xts <- xts(forecast_MET_t$mean,
                     order.by = for_dates)

plot(MET_t_valid, main = 'Prediccion vs Validacion',
     ylim = c(4000, 8500))

lines(for_MET_t_xts, col = "blue")
```

Visualización de datos (nuevamente)



Visualización de datos (nuevamente)



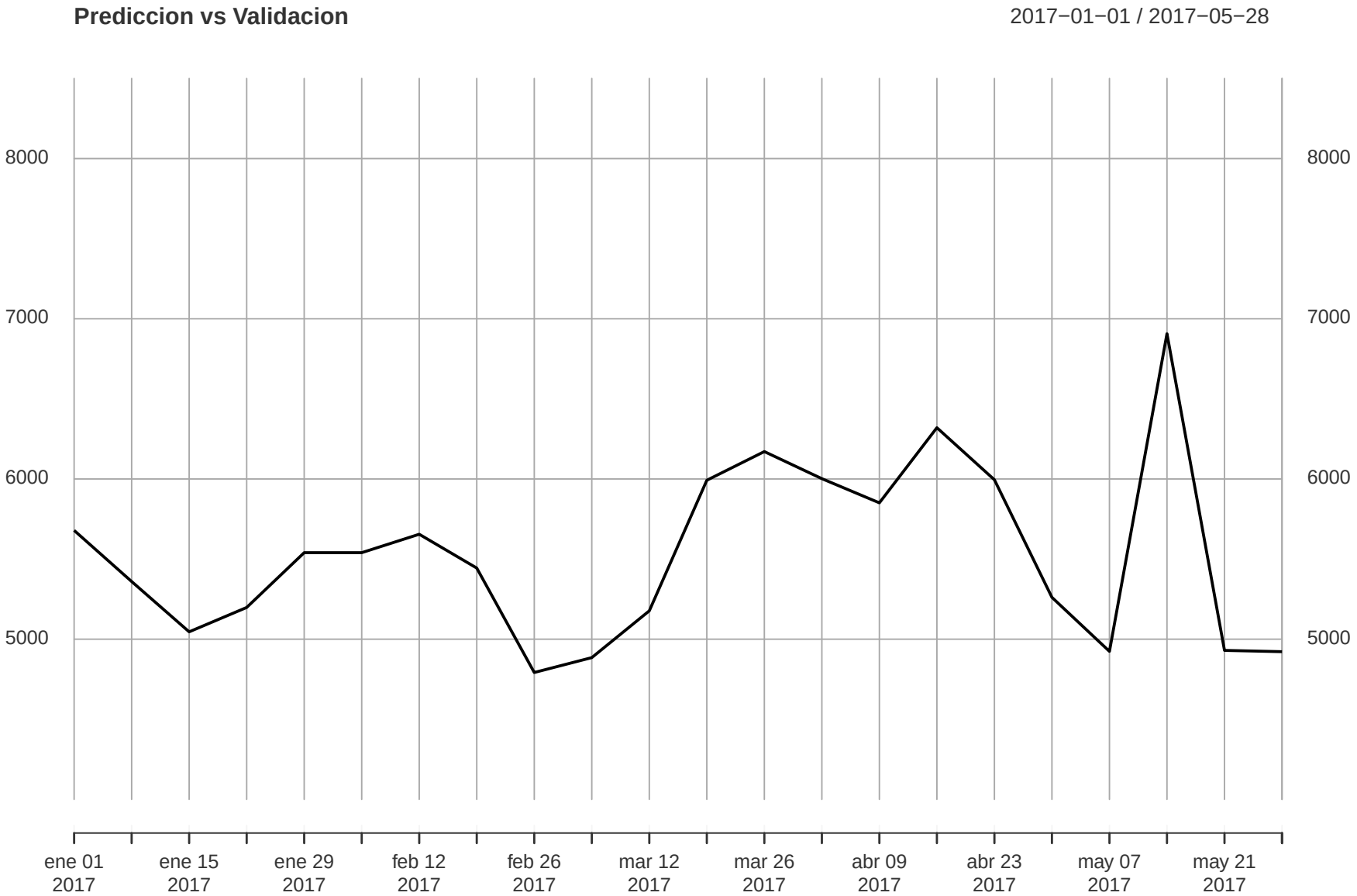
Intervalo de confianza

- La estimación automáticamente proporciona los intervalos de confianza al 80% y 95%
- Se pueden extraer estos elementos de igual modo que seleccionando columnas

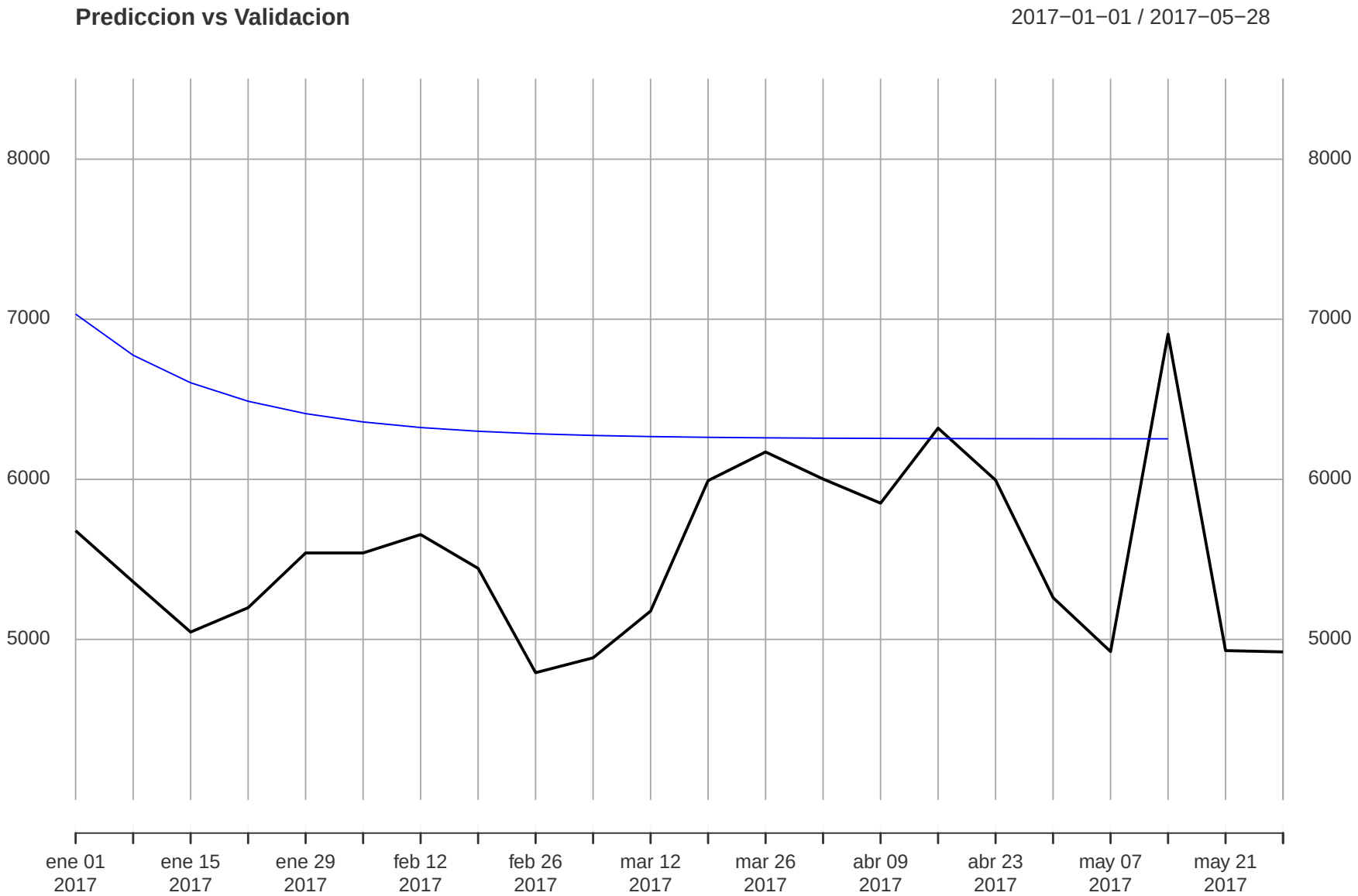
Intervalo de confianza

```
plot(MET_t_valid, main = 'Prediccion vs Validacion',
     ylim = c(4000, 8500))

lines(for_MET_t_xts, col = "blue")

lower <- xts(forecast_MET_t$lower[,"95%"],
             order.by = for_dates)
upper <- xts(forecast_MET_t$upper[,"95%"],
             order.by = for_dates)

lines(lower, col = "blue", lty = "dashed")
lines(upper, col = "blue", lty = "dashed")
```

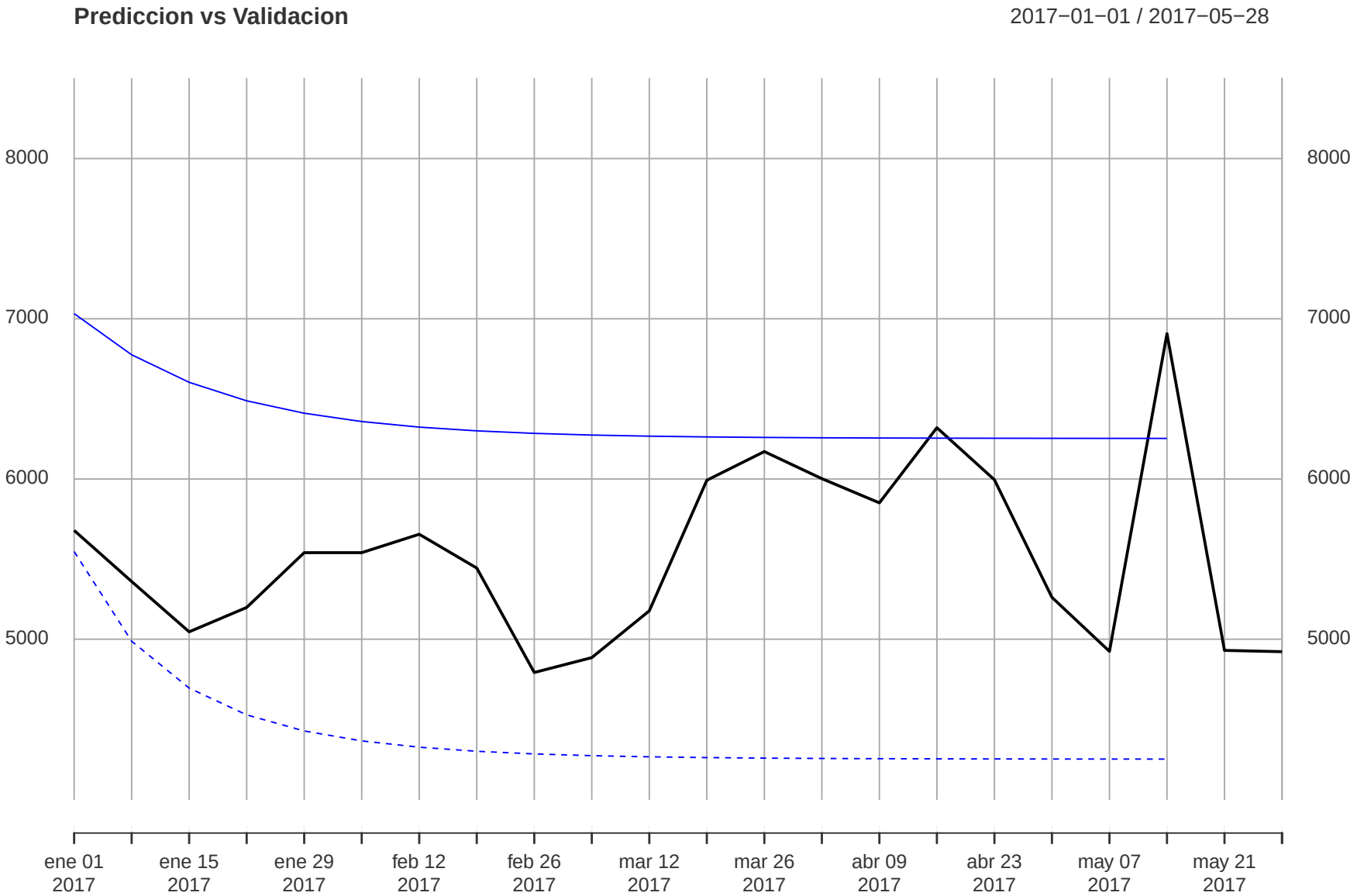
Intervalo de confianza



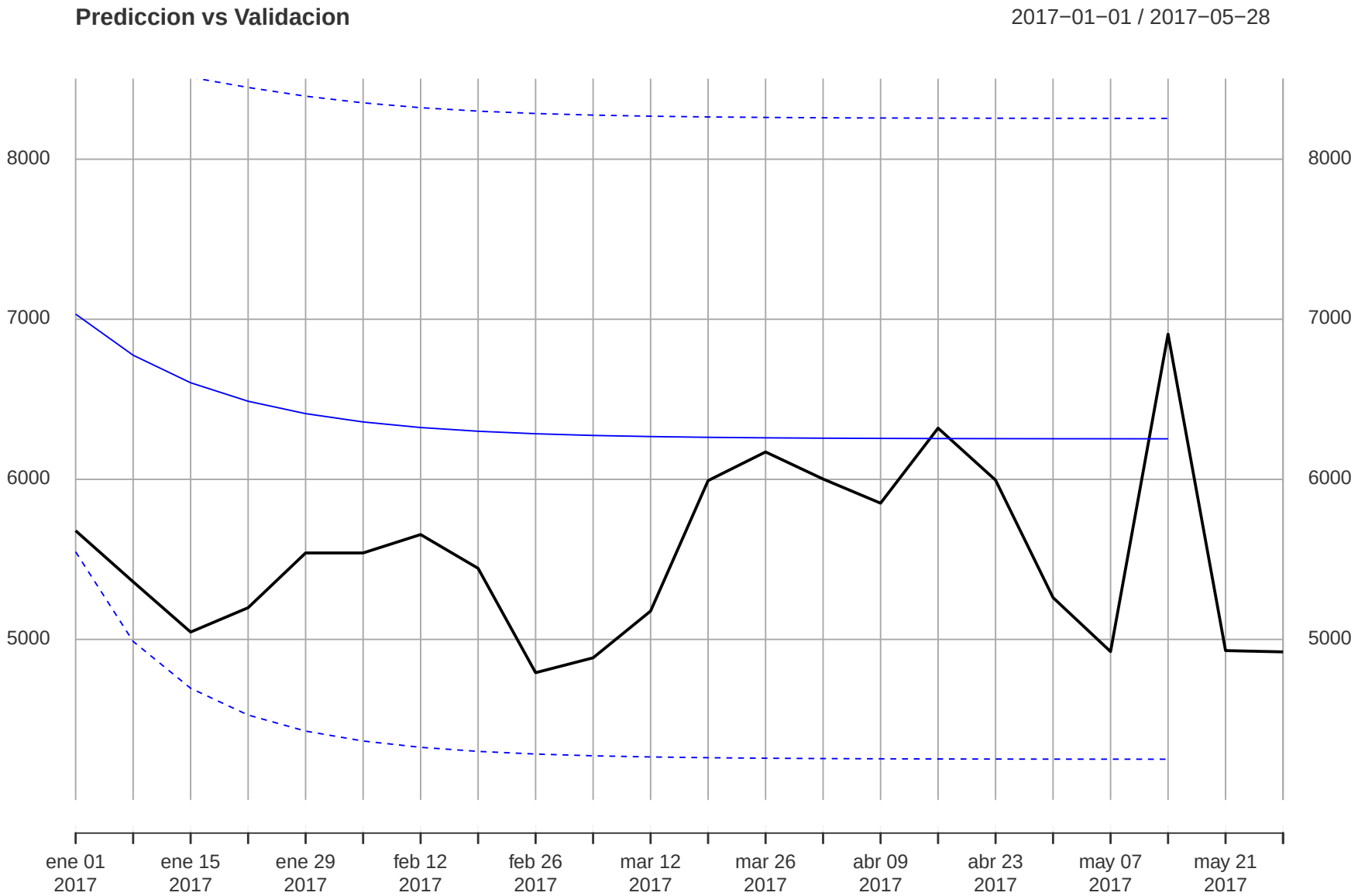
Intervalo de confianza



Intervalo de confianza



Intervalo de confianza



Elasticidad de la demanda

- Tomando las variables (unidimensionales) precio y cantidad demandada, supongamos que la cantidad se puede escribir en función del precio $q = f(p)$.
- La elasticidad de q respecto de p corresponde a

$$e_{q,p} = \frac{'partialq}{\partial p} \cdot \frac{p}{q}$$

Elasticidad de la demanda

Lo anterior es extraño, pero si p aumenta en un 1% la variación porcentual en $q(p)$ es

$$\frac{q(1,01p) - q(p)}{q(p)} \cdot 100$$

Si f es diferenciable se puede aplicar el polinomio de Taylor de grado uno y tomar $p_0 \approx 0$ para obtener

$$\frac{q(1,01p) - q(p)}{q(p)} \cdot 100 \approx 0.01p \frac{\partial q(p)/\partial p}{q(p)} \times 100$$

Elasticidad de la demanda

Para efectos de la calculabilidad de la elasticidad se puede definir $g(p) = \ln(q(p))$ y entonces

$$e_{q,p} = \frac{\Delta \ln(q)}{\Delta \ln(p)}$$

Elasticidad de la demanda

- El último resultado es extremadamente útil si uso un software como R
- En el caso de un modelo log-log, los coeficientes de regresión corresponden a la elasticidad
- Si estimo los betas del modelo $\ln(q_i) = \beta_0 + \beta_1 \ln(p_i) + \varepsilon_i$, β_1 es exactamente la elasticidad precio de la demanda
- ¿Qué se puede decir respecto de la elasticidad precio de la gama alta en la zona metropolitana?

Elasticidad de la demanda

```
bev_xts_train <- bev_xts[index(bev_xts) < "2017-01-01"]
bev_xts_valid <- bev_xts[index(bev_xts) >= "2017-01-01"]

l_MET_hi_p <- as.vector(log(bev_xts_train[,"MET.hi.p"]))
MET_hi_train <- MET_hi[1:length(l_MET_hi_p)]

MET_hi_train <- data.frame(as.vector(log(MET_hi_train)),
                           l_MET_hi_p)
colnames(MET_hi_train) <- c("log_sales", "log_price")

model_MET_hi <- lm(log_sales ~ log_price,
                   data = MET_hi_train)
model_MET_hi
```

Elasticidad de la demanda

```
##
## Call:
## lm(formula = log_sales ~ log_price, data = MET_hi_train)
##
## Coefficients:
## (Intercept)    log_price
##      13.977       -1.517
```