Práctico 3 - Ejercicio 7

Series Cronológicas 2024

Abril 2024

1. Ejercicio

Cree una función en R que haga las proyecciones de un modelo ARMA(1,1) genérico. Los argumentos de la función deben ser:

- Los parámetros del modelo $(\phi; \theta; \mu)$, siendo μ la media del proceso y **no** su constante.
- El horizonte hasta el que se desea predecir (s).
- La serie que se desea utilizar (Y_t).

Para corroborar su función, puede simular una serie AR(1) con $(\phi; \theta; \mu) = (\phi_1, 0, 0)$ y contrastar las predicciones de su función con su valor teórico para este modelo.

2. Predicciones de un ARMA(1,1)

Un proceso ARMA(1,1) puede escribirse como un modelo sin constante:

$$X_t = \phi X_{t-1} + \varepsilon_t - \theta \varepsilon_{t-1}$$

Donde $X_t = Y_t - \mu$.

De esta forma, las predicciones variarán dependiendo de cuál sea el horizonte de predicción:

- Predicción a un paso $(s = 1)$:

$$X_{t+1} = \phi X_t + \varepsilon_{t+1} - \theta \varepsilon_t \Rightarrow \hat{X}_{t+1|t} = E\left(\phi X_t + \varepsilon_{t+1} - \theta \varepsilon_t \middle| X_t, X_{t-1}, \ldots; \varepsilon_t, \varepsilon_{t-1}, \ldots\right) = \phi X_t - \theta \varepsilon_t$$

- Predicción a dos pasos $(s = 2)$:

$$\begin{aligned} X_{t+2} = \phi X_{t+1} + \varepsilon_{t+2} - \theta \varepsilon_{t+1} \Rightarrow \hat{X}_{t+2|t} &= E\left(\phi X_{t+1} + \varepsilon_{t+2} - \theta \varepsilon_{t+1} | X_t, X_{t-1}, \ldots; \varepsilon_t, \varepsilon_{t-1}, \ldots\right) \\ &= \phi E\left(X_{t+1} | X_t, X_{t-1}, \ldots; \varepsilon_t, \varepsilon_{t-1}, \ldots\right) = \phi \hat{X}_{t+1|t} \end{aligned}$$

- Predicción a $s > 1$ pasos:

$$\begin{aligned} X_{t+s} = \phi X_{t+s-1} + \varepsilon_{t+s} - \theta \varepsilon_{t+s-1} \Rightarrow \hat{X}_{t+s|t} &= E\left(\phi X_{t+s-1} + \varepsilon_{t+s} - \theta \varepsilon_{t+s-1} | X_t, X_{t-1}, \ldots; \varepsilon_t, \varepsilon_{t-1}, \ldots\right) \\ &= \phi E\left(X_{t+s-1} | X_t, X_{t-1}, \ldots; \varepsilon_t, \varepsilon_{t-1}, \ldots\right) = \phi \hat{X}_{t+s-1|t} \end{aligned}$$

Alternativamente, puede trabajarse con el modelo con constante:

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + \varepsilon_t - \theta\varepsilon_{t-1} \Rightarrow Y_t = \mu + \phi Y_{t-1} - \phi\mu + \varepsilon_t - \theta\varepsilon_{t-1} \Rightarrow$$
$$Y_t = \mu(1-\phi) + \phi Y_{t-1} + \varepsilon_t - \theta\varepsilon_{t-1} \Rightarrow Y_t = \delta + \phi Y_{t-1} + \varepsilon_t - \theta\varepsilon_{t-1}$$

Donde $\delta = \mu(1-\phi)$.

En este caso, las predicciones serán de la forma:

- Predicción a un paso ($s = 1$):

$$Y_{t+1} = \delta + \phi Y_t + \varepsilon_{t+1} - \theta\varepsilon_t \Rightarrow \hat{Y}_{t+1|t} = E\left(\delta + \phi Y_t + \varepsilon_{t+1} - \theta\varepsilon_t \middle| Y_t, Y_{t-1}, \ldots; \varepsilon_t, \varepsilon_{t-1}, \ldots\right) = \delta + \phi Y_t - \theta\varepsilon_t$$

- Predicción a dos pasos ($s = 2$):

$$\begin{aligned} Y_{t+2} = \delta + \phi Y_{t+1} + \varepsilon_{t+2} - \theta\varepsilon_{t+1} &\Rightarrow \hat{Y}_{t+2|t} = E\left(\delta + \phi Y_{t+1} + \varepsilon_{t+2} - \theta\varepsilon_{t+1} | Y_t, Y_{t-1}, \ldots; \varepsilon_t, \varepsilon_{t-1}, \ldots\right) \\ &= \delta + \phi E\left(Y_{t+1} | Y_t, Y_{t-1}, \ldots; \varepsilon_t, \varepsilon_{t-1}, \ldots\right) = \delta + \phi\hat{Y}_{t+1|t} \end{aligned}$$

- Predicción a $s > 1$ pasos:

$$\begin{aligned} Y_{t+s} = \delta + \phi Y_{t+s-1} + \varepsilon_{t+s} - \theta\varepsilon_{t+s-1} &\Rightarrow \hat{Y}_{t+s|t} = E\left(\delta + \phi Y_{t+s-1} + \varepsilon_{t+s} - \theta\varepsilon_{t+s-1} | Y_t, Y_{t-1}, \ldots; \varepsilon_t, \varepsilon_{t-1}, \ldots\right) \\ &= \delta + \phi E\left(Y_{t+s-1} | Y_t, Y_{t-1}, \ldots; \varepsilon_t, \varepsilon_{t-1}, \ldots\right) = \delta + \phi\hat{Y}_{t+s-1|t} \end{aligned}$$

A partir de las fórmulas anteriores, queda claro que es necesario contar con una estimación de los shocks pasados para poder calcular las predicciones. Para ello, se utiliza los residuos, los cuales se calculan mediante los errores de predicción a un paso dentro de la muestra:

$$\hat{\varepsilon}_t = Y_t - \hat{Y}_{t|t-1}$$

3. Función

Para construir una función que calcule las predicciones de un ARMA(1,1) genérico, se siguen los siguientes pasos:

1. Obtener los valores ajustados de la serie ($\hat{Y}_{t|t-1}$).
2. Calcular los residuos del modelo ($\hat{\varepsilon}_t$).
3. Obtener la predicción a un paso ($\hat{Y}_{t+1|t}$).
4. Calcular las predicciones hasta s pasos ($\hat{Y}_{t+s|t}$).

```r
predicciones_ARMA_1_1 <- function(phi, theta, mu, s, y){

  # Paso 0: inicializar variables
  n <- length(y)
  y_gorro <- rep(0, n)
  res <- rep(0, n)
  pred <- rep(0, s)
  delta <- mu*(1 - phi)

  # Pasos 1 y 2: obtener valores ajustados y residuos
  y_gorro[1] <- mu # La predicción sin información pasada es igual a la media incondicional
  res[1] <- y[1] - y_gorro[1]
  for (i in 2:n) {
```

```
    y_gorro[i] <- delta + phi*y[i-1] + theta*res[i-1] # Utilizamos notación de R para parte MA
    res[i] <- y[i] - y_gorro[i]
  }

  # Paso 3: obtener la predicción a un paso
  pred[1] <- delta + phi*y[n] + theta*res[n]

  # Paso 4: calcular las predicciones a s pasos
  for (i in 2:s) {
    pred[i] <- delta + phi*pred[i-1]
  }

  return(pred)

}
```

4. Aplicaciones

4.1. Serie simulada: proceso AR(1)

Para probar que la función funciona, se simularon 1000 observaciones de un proceso AR(1) con media igual a cero y coeficiente $\phi_1 = 0,2$.

```
# Simulamos un proceso AR(1) con phi = 0,2
set.seed(1234)
ar_1 <- arima.sim(n = 1000, list(order = c(1, 0, 0), ar = 0.2), mean = 0)
```

```
# Aplicamos la función predicciones_ARMA_1_1() para calcular las predicciones a 5 pasos
pred_funcion <- predicciones_ARMA_1_1(phi = 0.2,
                                      theta= 0,
                                      mu = 0,
                                      s = 5,
                                      y = ar_1)
pred_funcion
```

```
## [1] 0.1537713439 0.0307542688 0.0061508538 0.0012301708 0.0002460342
```

```
# Estimamos un modelo para la serie simulada
modelo <- Arima(y = ar_1,
                order = c(1, 0, 0),
                include.mean = FALSE,
                lambda = NULL)
summary(modelo)
```

```
## Series: ar_1
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##          ar1
##       0.2231
```

```
## s.e.  0.0308
##
## sigma^2 = 0.9912:  log likelihood = -1414.06
## AIC=2832.11   AICc=2832.12   BIC=2841.93
##
## Training set error measures:
##                       ME      RMSE       MAE      MPE     MAPE      MASE
## Training set -0.02517977 0.9951038 0.7837872 131.7476 168.7819 0.7675972
##                     ACF1
## Training set 0.006255908
```

```r
# Calculamos las predicciones mediante la función predict()
pred_predict <- predict(modelo, n.ahead = 5)
pred_predict <- pred_predict$pred
pred_predict
```

```
## Time Series:
## Start = 1001
## End = 1005
## Frequency = 1
## [1] 0.1715594533 0.0382810545 0.0085418734 0.0019059977 0.0004252963
```

```r
# Calculamos las predicciones mediante la función forecast()
pred_forecast <- forecast(modelo, h = 5)
pred_forecast <- pred_forecast$mean
pred_forecast
```

```
## Time Series:
## Start = 1001
## End = 1005
## Frequency = 1
## [1] 0.1715594533 0.0382810545 0.0085418734 0.0019059977 0.0004252963
```

```r
# Ordenamos las distintas predicciones para poder graficarlas
predicciones <- data.frame(paso = 1:5,
                           "predicciones_ARMA_1_1" = pred_funcion,
                           "predict" = pred_predict,
                           "forecast" = pred_forecast)
predicciones <- predicciones %>%
  pivot_longer(cols = c("predicciones_ARMA_1_1",
                        "predict",
                        "forecast"),
               values_to = "valores")

# Graficamos las FAC teórica y estimada
predicciones_grafico <- ggplot(predicciones) +
  geom_line(aes(x = paso, y = valores, color = name)) +
  labs(x = "Paso",
       y = "Predicción",
       color = "Función")
predicciones_grafico
```

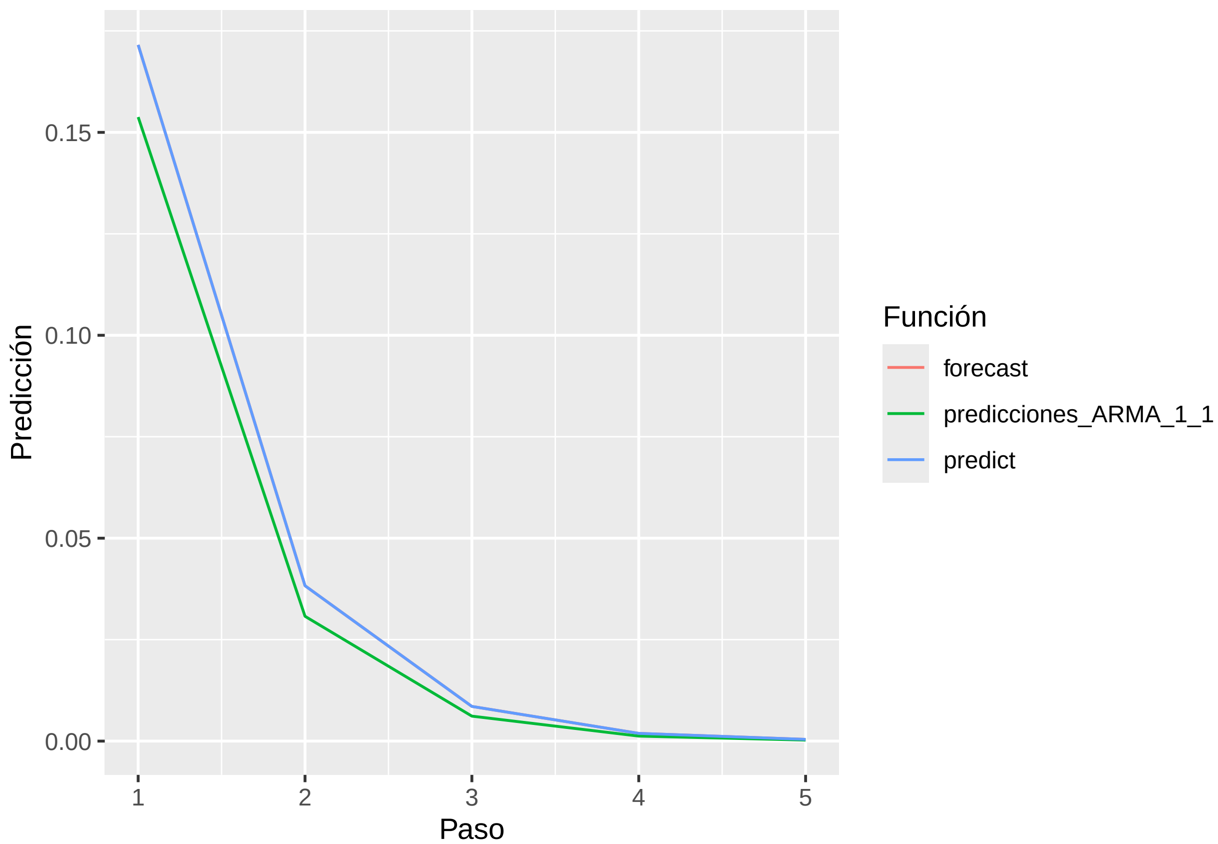

Figura 1: Predicciones a 5 pasos de un proceso AR(1).

4.2. Serie real: *bluebird*

```
# Cargamos la serie "bluebird" de la librería TSA
# La serie mide el logaritmo de la cantidad y el precio de las papas fritas "Bluebird"
# de Nueva Zelanda a lo largo de 104 semanas
data(bluebird)
head(bluebird)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
##   log.sales price
## 1  11.49276  1.71
## 2  11.53681  1.76
## 3  11.82208  1.64
## 4  11.89337  1.64
## 5  11.33383  1.73
## 6  11.93430  1.59
```

```r
# Graficamos la serie bluebird (ventas)
autoplot(bluebird[,1]) +
  labs(x = "Semana",
       y = "Ventas")
```

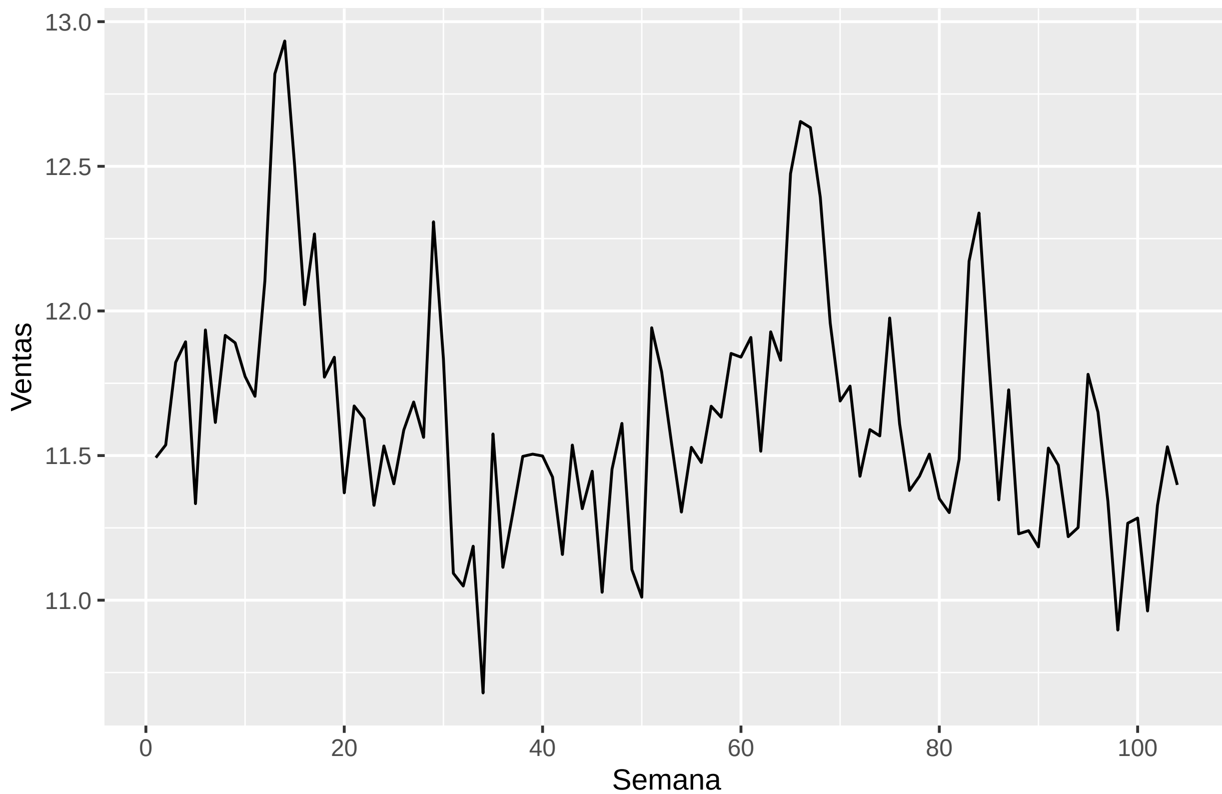

Figura 2: Unidades vendidas (en logaritmos) de las papas fritas Bluebird a lo largo de 104 semanas.

```r
# Ajustamos un modelo ARMA(1,1) a la serie
modelo_bluebird <- Arima(y = bluebird[,1],
                         order = c(1, 0, 1),
                         include.mean = TRUE,
                         lambda = NULL)
summary(modelo_bluebird)
```

```
## Series: bluebird[, 1]
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ma1     mean
##       0.7055  -0.0959  11.6138
## s.e.  0.1077   0.1570   0.0908
##
## sigma^2 = 0.09761:  log likelihood = -25.33
## AIC=58.67   AICc=59.07   BIC=69.24
```

```
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE     MAPE      MASE
## Training set 0.001162347 0.3078823 0.2376595 -0.05943344 2.038052 0.8837386
##                     ACF1
## Training set 0.007564352
```

```
coeficientes <- modelo_bluebird$coef
coeficientes # El coeficiente intercept nos da la media estimada (NO la constante)
```

```
##         ar1         ma1   intercept
##  0.70554455 -0.09587068 11.61378534
```

```
# Aplicamos la función predicciones_ARMA_1_1() para calcular las predicciones a 5 pasos
pred_funcion <- predicciones_ARMA_1_1(phi = coeficientes[1],
                                      theta = coeficientes[2], # Notación de R
                                      mu = coeficientes[3],
                                      s = 5,
                                      y = bluebird[,1])
pred_funcion
```

```
## [1] 11.47574 11.51639 11.54507 11.56530 11.57958
```

```
# Calculamos las predicciones mediante la función predict()
pred_predict <- predict(modelo_bluebird, n.ahead = 5)
pred_predict <- pred_predict$pred
pred_predict
```

```
## Time Series:
## Start = 105
## End = 109
## Frequency = 1
## [1] 11.47574 11.51639 11.54507 11.56530 11.57958
```

```
# Calculamos las predicciones mediante la función forecast()
pred_forecast <- forecast(modelo_bluebird, h = 5)
pred_forecast <- pred_forecast$mean
pred_forecast
```

```
## Time Series:
## Start = 105
## End = 109
## Frequency = 1
## [1] 11.47574 11.51639 11.54507 11.56530 11.57958
```

```
# Ordenamos las distintas predicciones para poder graficarlas
predicciones <- data.frame(paso = 1:5,
                           "predicciones_ARMA_1_1" = pred_funcion,
                           "predict" = pred_predict,
                           "forecast" = pred_forecast)
```

```r
predicciones <- predicciones %>%
  pivot_longer(cols = c("predicciones_ARMA_1_1",
                        "predict",
                        "forecast"), values_to = "valores")

# Graficamos las FAC teórica y estimada
predicciones_grafico <- ggplot(predicciones) +
  geom_line(aes(x = paso, y = valores, color = name)) +
  labs(x = "Paso",
       y = "Predicción",
       color = "Función")
predicciones_grafico
```

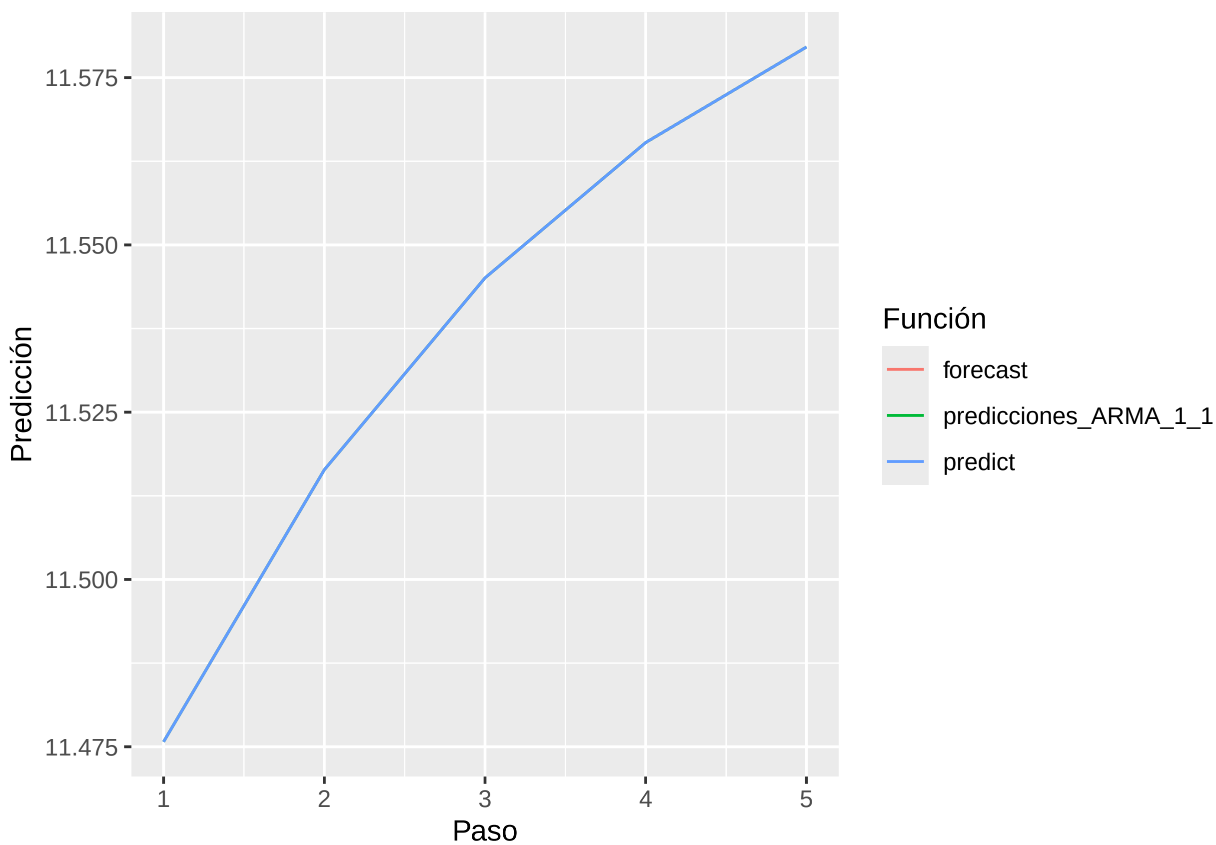

Figura 3: Predicciones a 5 pasos para la serie Bluebird.

```r
pred_forecast <- forecast(modelo_bluebird, h = 5, fan = TRUE)
pred_funcion <- ts(c(as.vector(bluebird[,1]), pred_funcion))
autoplot(pred_forecast) +
  autolayer(pred_funcion, color = "red") +
  autolayer(bluebird[,1], color = "black") +
  labs(x = "Semana",
       y = "Ventas",
       title = "")
```

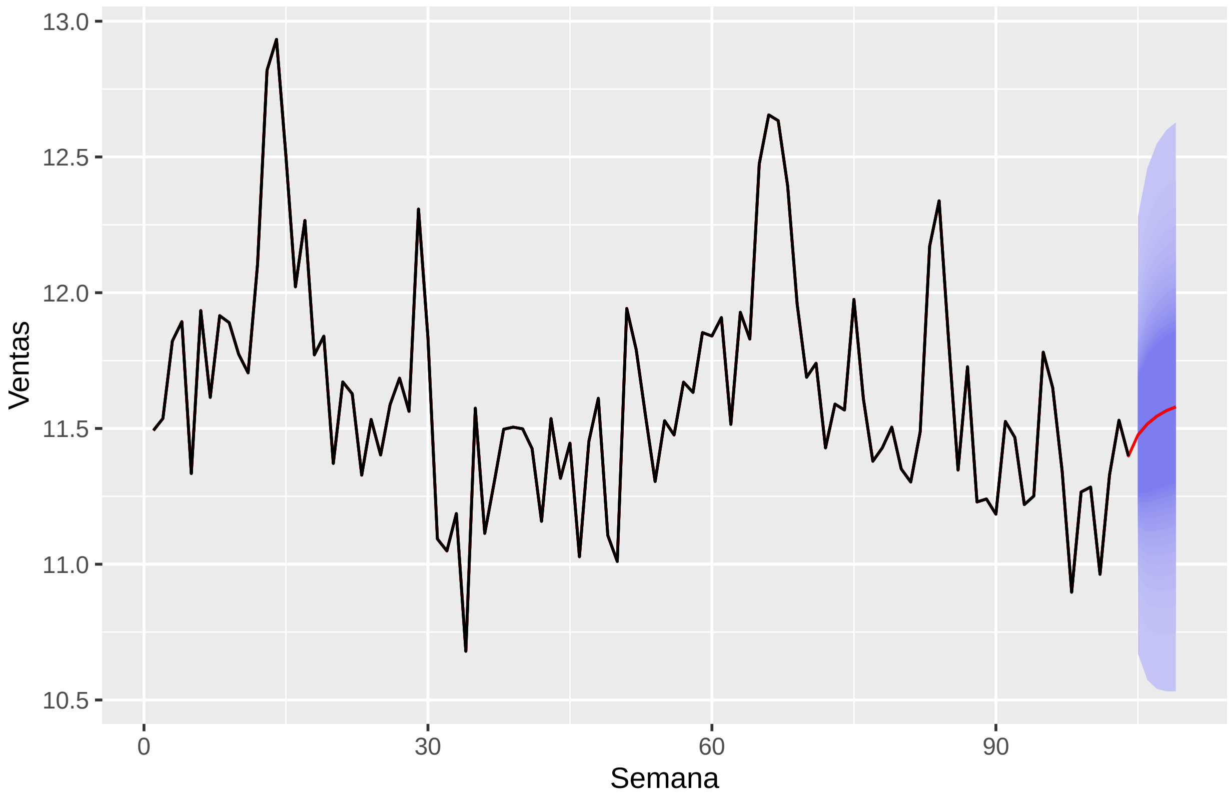

Figura 4: Predicción a 5 pasos de unidades vendidas (en logaritmos) de las papas fritas Bluebird. La línea roja corresponde a la predicción de la función creada y la azul a la predicción de la función forecast.