Act7 Series de tiempo C_IACD_Estadística

A01750164 | Paul Martín García Morfín

2022-11-15

Introducción a series de tiempo

Usa los datos de las ventas de gasolina en una estación de servicio para analizar modelos de pronósticos de la serie de tiempo:

Semana | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
Galones de gasolina (miles) | 17 | 21 | 19 | 23 | 18 | 16 | 20 | 18 | 22 | 20 | 15 | 22 |

```
t = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
y = c(17,   21, 19, 23, 18, 16, 20, 18, 22, 20, 15, 22)
n = 12
```

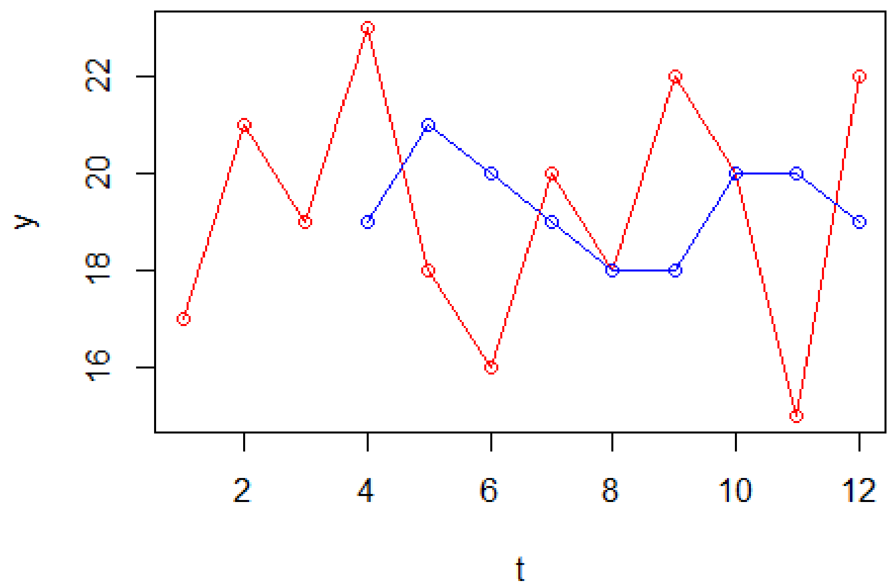
Utiliza los métodos de suavizamiento:

Promedios móviles

```
p = NA
e = NA
for(i in 1:(n-3)){
  p[i+3]=(y[i]+y[i+1]+y[i+2])/3; e[i+3] = p[i+3] - y[i+3]
}
T = data.frame(t, p, y, e^2)
CME = mean(e^2, na.rm=TRUE)
cat("CME = ", CME)

## CME =  10.22222

plot(t, y, type="o", col="red")
x = (3+1):n
lines(x,p[x],type="o", col="blue")
```

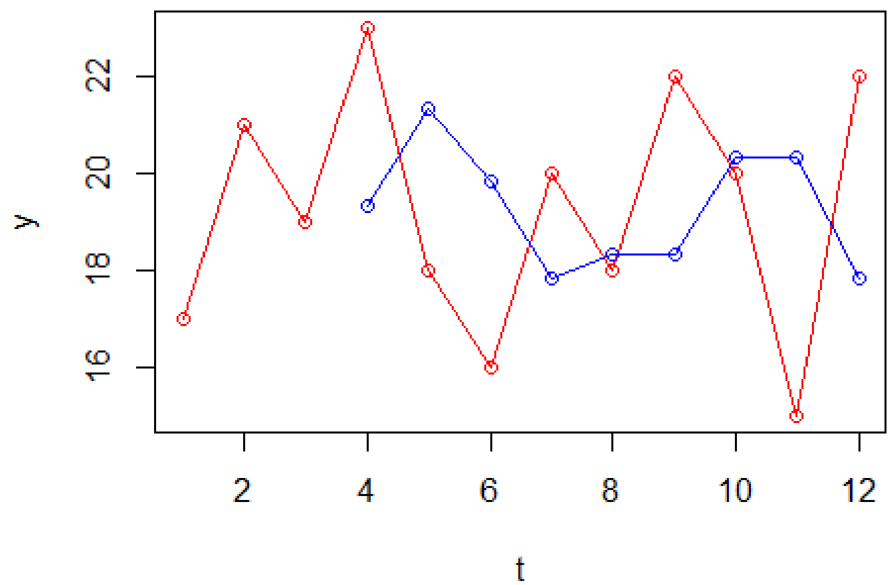


Promedios móviles ponderados

```
p2 = NA
e2 = NA
for(i in 1:(n-3)){
  p2[i+3] = (1/6)*y[i] + (2/6)*y[i+1] + (3/6)*y[i+2]
  e2[i+3] = p2[i+3] - y[i+3]
}
T2 = data.frame(t, p2, y, e2^2)
CME2 = mean(e2^2, na.rm=TRUE)
cat("CME = ", CME2)

## CME =  11.49074

plot(t, y, type="o", col="red")
x = (3+1):n
lines(x, p2[x], type="o", col="blue")
```

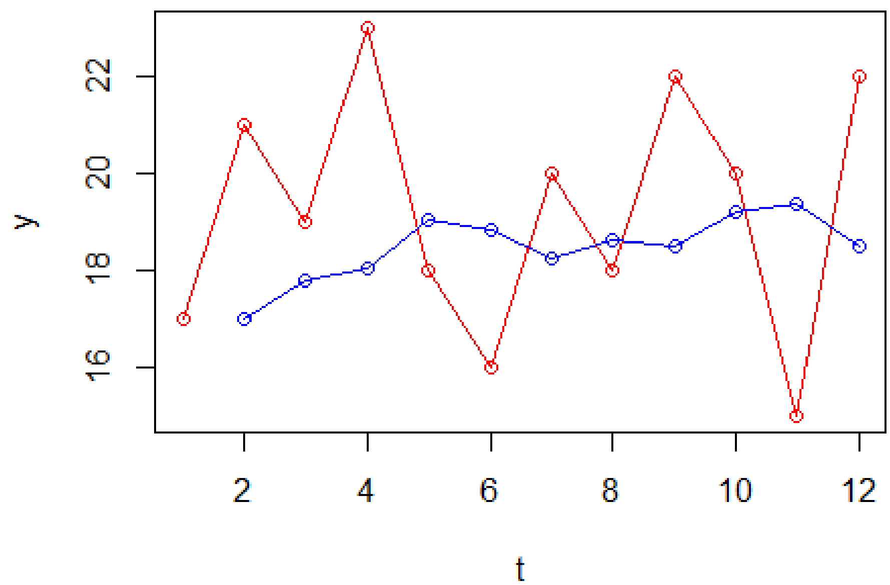


Método de suavizamiento exponencial

```
p3 = NA
e3 = NA
p3[1] = y[1]
p3[2] = y[1]
a = 0.20
for(i in 2:n){
  p3[i] = a*y[i-1]+(1-a)*p3[i-1]
  e3[i] = y[i]- p3[i]
}
T3 = data.frame(t, p3, y, e3^2)
CME3 = mean(e3^2, na.rm=TRUE)
cat("CME = ", CME3)

## CME =  8.982231

plot(t, y, type="o", col="red")
x = 2:n
lines(x, p3[x], type="o", col="blue")
```



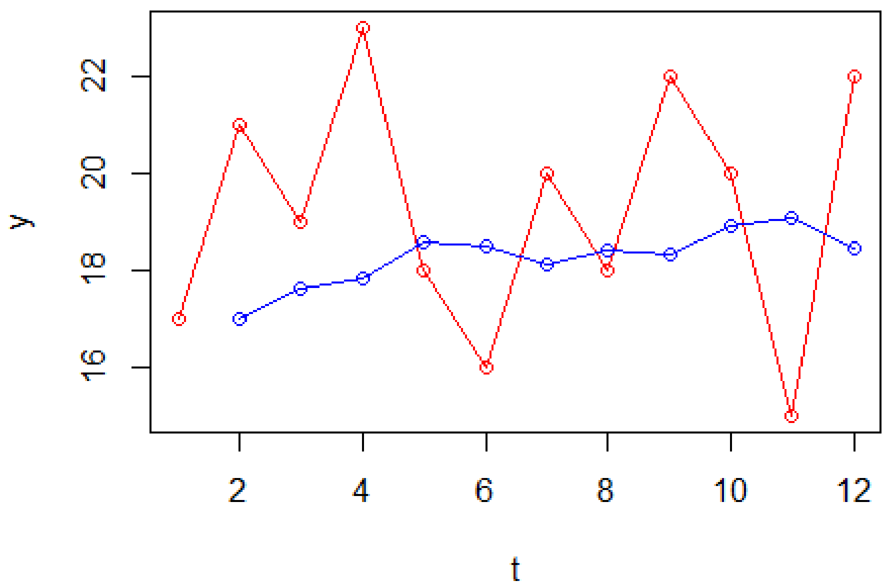
Utiliza varios valores de α en el método de suavizamiento hasta encontrar el valor de que minimice el CME.

- Con $\alpha = 0.15$

```
p3 = NA
e3 = NA
p3[1] = y[1]
p3[2] = y[1]
a = 0.15
for(i in 2:n){
  p3[i] = a*y[i-1]+(1-a)*p3[i-1]
  e3[i] = y[i]- p3[i]
}
T2 = data.frame(t, p3, y, e3^2)
CME3 = mean(e3^2, na.rm=TRUE)
cat("CME = ", CME3)

## CME =  8.984687

plot(t, y, type="o", col="red")
x = 2:n
lines(x, p3[x], type="o", col="blue")
```

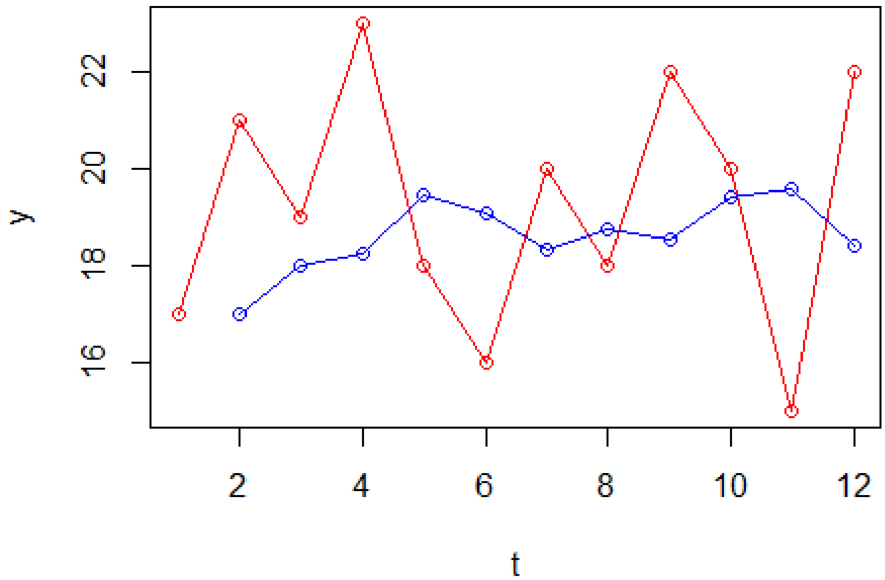


- Con $\alpha = 0.25$

```
p3 = NA
e3 = NA
p3[1] = y[1]
p3[2] = y[1]
a = 0.25
for(i in 2:n){
  p3[i] = a*y[i-1]+(1-a)*p3[i-1]
  e3[i] = y[i]- p3[i]
}
T2 = data.frame(t, p3, y, e3^2)
CME3 = mean(e3^2, na.rm=TRUE)
cat("CME = ", CME3)

## CME =  9.123907

plot(t, y, type="o", col="red")
x = 2:n
lines(x, p3[x], type="o", col="blue")
```

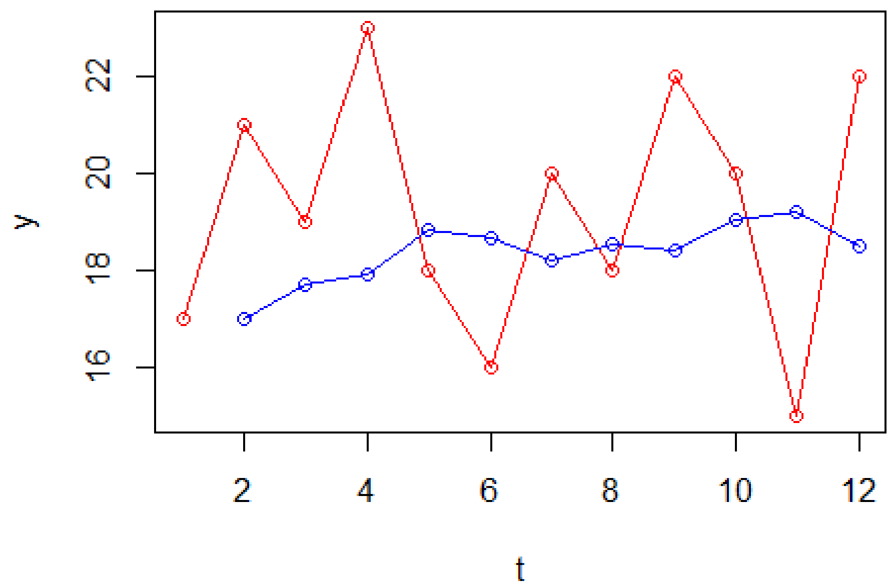


- Con $\alpha = 0.175$

```
p3 = NA
e3 = NA
p3[1] = y[1]
p3[2] = y[1]
a = 0.175
for(i in 2:n){
  p3[i] = a*y[i-1]+(1-a)*p3[i-1]
  e3[i] = y[i]- p3[i]
}
T3 = data.frame(t, p3, y, e3^2)
CME3 = mean(e3^2, na.rm=TRUE)
cat("CME = ", CME3)

## CME =  8.959901

plot(t, y, type="o", col="red")
x = 2:n
lines(x, p3[x], type="o", col="blue")
```



Concluye sobre cuál de los modelos usados es el mejor

Para este caso parece ser que el modelo de suavizamiento exponencial con $\alpha = 0.175$ es el que mejor se ajusta a nuestros datos y a su tendencia según la métrica del CME, pues tiene el valor más bajo de todos los métodos utilizados, siendo igual a 8.959901.

Predice cuáles son las ventas de gasolina esperadas para la semana 13

```
F13 = 0.175*22 + 0.825*18.48061
cat("Ventas de gasolina esperadas para la semana 13: $", F13)

## Ventas de gasolina esperadas para la semana 13: $ 19.0965
```