

A7-Series de tiempo estacionarias

Francisco Castorena, A00827756

2023-11-03

Problema 1

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

Suavizamiento por promedios móviles

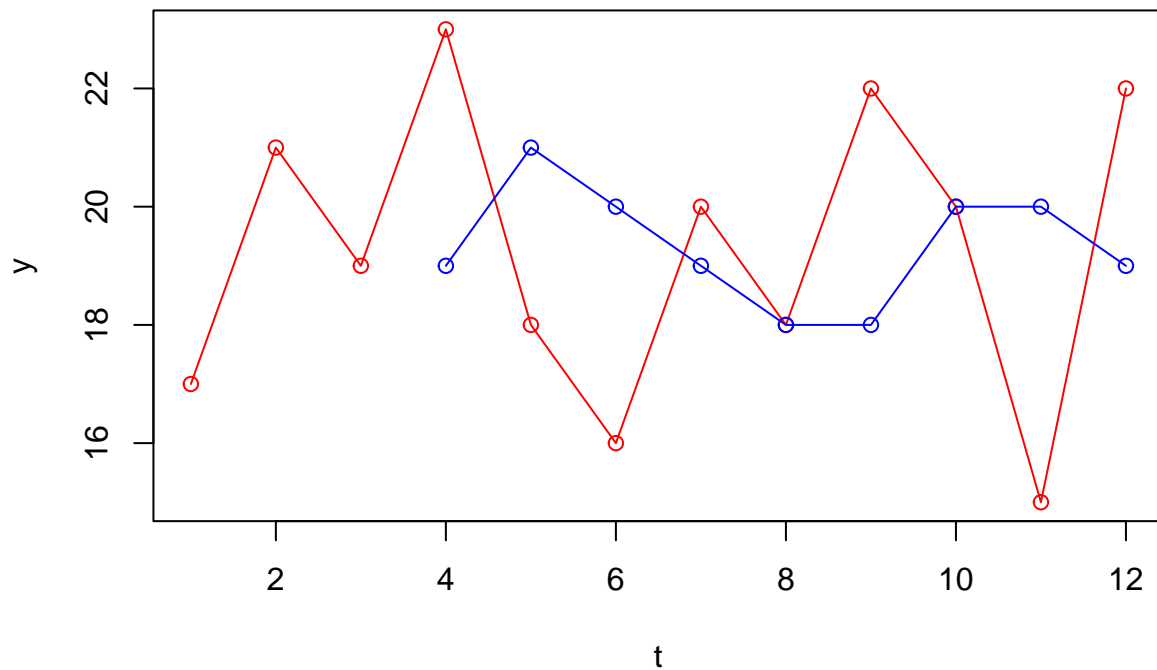
```
t = c(1:12)
y = c(17,21,19,23,18,16,20,18,22,20,15,22)
p = NA
e = NA
n = length(y)
k = 3 #Valor de k para promedios móviles
for(i in 1:(n-3)){p[i+3]=(y[i]+y[i+1]+y[i+2])/3; e[i+3] = p[i+3] - y[i+3]}
```

```
# Creamos dataframe para organizar los datos en una tabla
T=data.frame(t,y,p,e^2)

# Calculamos el cuadrado medio de los errores sin NA
CME = mean(e^2,na.rm=TRUE)
cat(CME)
```

```
## 10.22222
```

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



Suavizamiento por promedios móviles ponderados

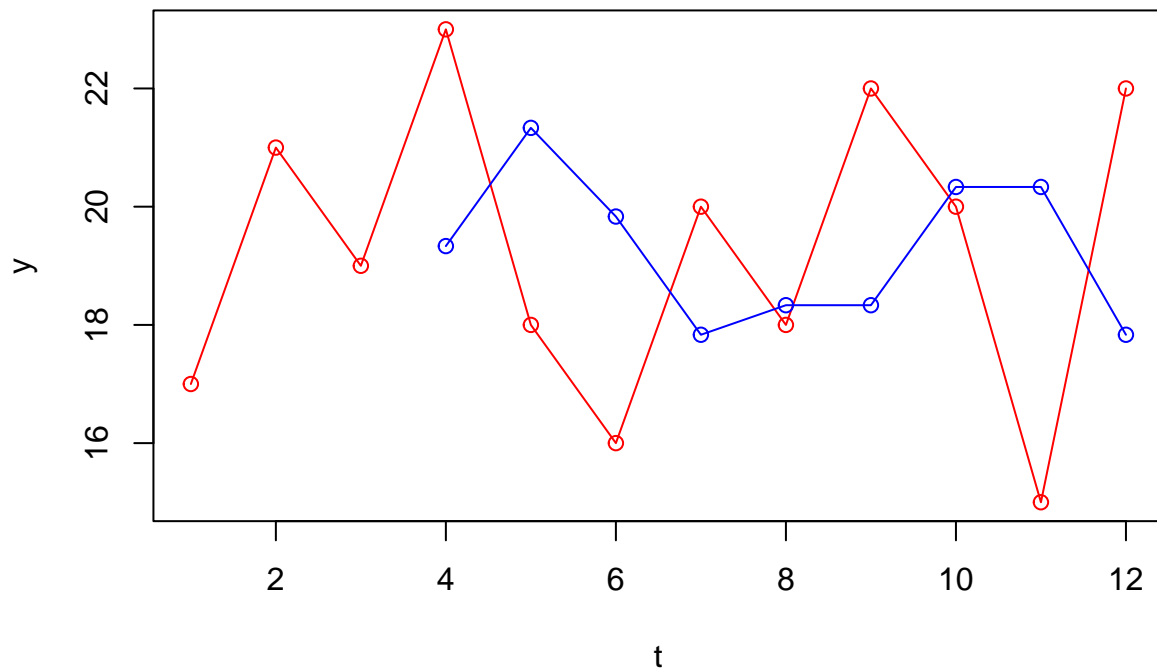
```
p2 = NA
e2 = NA
for(i in 1:(n-3)){p2[i+3]=(1/6)*y[i]+(2/6)*y[i+1]+(3/6)*y[i+2];
e2[i+3] = p2[i+3] - y[i+3]}
```

```
# Utiliza data.frame() para organizar una tabla:
T2=data.frame(t,y, p2,e2^2)

# Calcula el cuadrado medio de los errores sin NA:
CME2=mean(e2^2,na.rm=TRUE)
cat(CME2)
```

```
## 11.49074
```

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



Suavizamiento exponencial

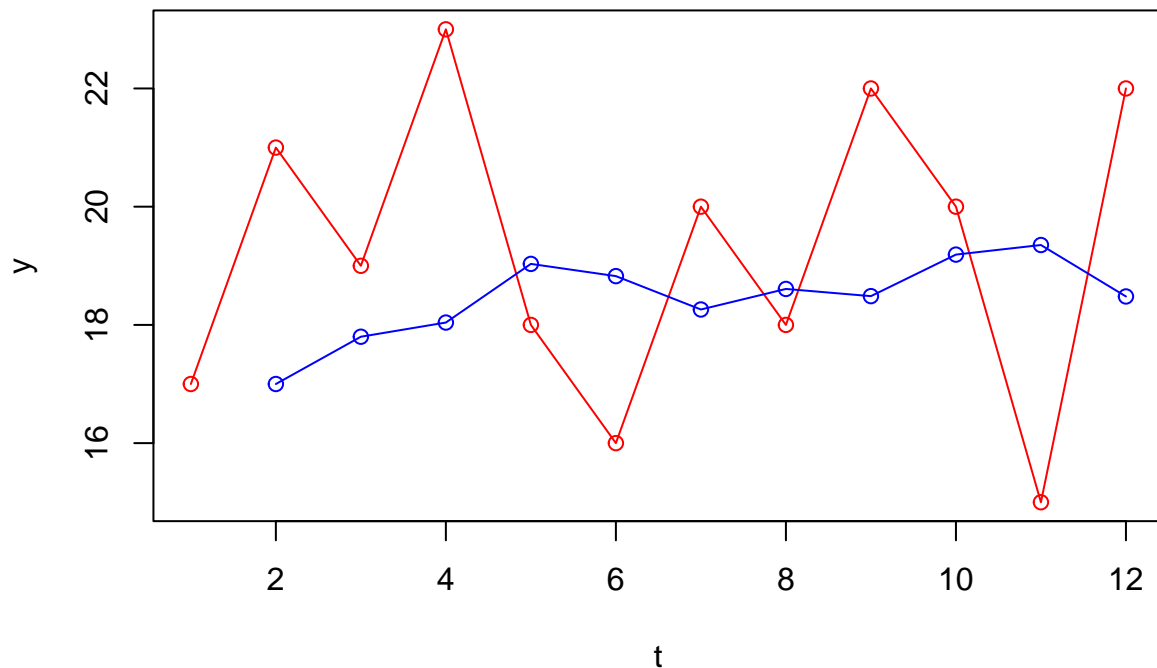
```
# Utiliza el comando for para el cálculo de los pronósticos y los errores:
p3 = NA
e3 = NA
p3[1]=y[1]
p3[2]=y[1]
a=0.20
for(i in 3:n){p3[i]=a*y[i-1]+(1-a)*p3[i-1];
e3[i] = y[i]- p3[i]}
```

```
# Utiliza data.frame() para organizar una tabla:
T3=data.frame(t,y, p3,e3^2)

# Calcula el cuadrado medio de los errores sin NA:
CME3=mean(e3^2,na.rm=TRUE)
cat(CME3)
```

```
## 8.280454
```

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



```
cat("Promedio de los cuadrados de los errores, medias moviles: ", CME,"\n")
```

```
## Promedio de los cuadrados de los errores, medias moviles: 10.22222
```

```
cat("Promedio de los cuadrados de los errores, medias moviles ponderadas: ", CME2,"\n")
```

```
## Promedio de los cuadrados de los errores, medias moviles ponderadas: 11.49074
```

```
cat("Promedio de los cuadrados de los errores, suavizamiento exponencial: ", CME3)
```

```
## Promedio de los cuadrados de los errores, suavizamiento exponencial: 8.280454
```

En base a estos valores se escoge el modelo con menor error, en este caso concluimos que el método de suavizamiento exponencial con α igual a 0.2 es el modelo que mejor pronostica los datos.

Suavizamiento exponencial, varios valores de α

Este código se creó con ayuda de chat-GPT

```
# Función para calcular el CME dado un valor de a
calculate_CME <- function(a) {
  p3 = rep(NA, n)
```

```

e3 = rep(NA, n)
p3[1] = y[1]
p3[2] = y[1]

for (i in 3:n) {
  p3[i] = a * y[i - 1] + (1 - a) * p3[i - 1]
  e3[i] = y[i] - p3[i]
}

CME = mean(e3^2, na.rm = TRUE)
return(CME)
}

# Valores de "a" a evaluar
a_values = seq(0.01, 0.99, by = 0.01)

# Inicializa el CME mínimo y el valor de "a" correspondiente
min_CME = Inf
best_a = NA

# Encuentra el valor de "a" que minimiza el CME
for (a in a_values) {
  current_CME = calculate_CME(a)
  if (current_CME < min_CME) {
    min_CME = current_CME
    best_a = a
  }
}

cat("El valor óptimo de 'a' que minimiza el CME es:", best_a, "\n")

```

```
## El valor óptimo de 'a' que minimiza el CME es: 0.17
```

```
cat("El CME mínimo es:", min_CME, "\n")
```

```
## El CME mínimo es: 8.256687
```

Predice cuáles son las ventas de gasolina esperadas para la semana 13 con el mejor método que hayas obtenido.

```

# Extiende la serie de tiempo hasta la semana 13
t = c(1:13)
y = c(17, 21, 19, 23, 18, 16, 20, 18, 22, 20, 15, 22, NA) # Agrega NA para la semana 13

n = length(y)

# Utiliza el comando for para el cálculo de los pronósticos y los errores:
p4 = rep(NA, n)
e4 = rep(NA, n)
p4[1] = y[1]

```

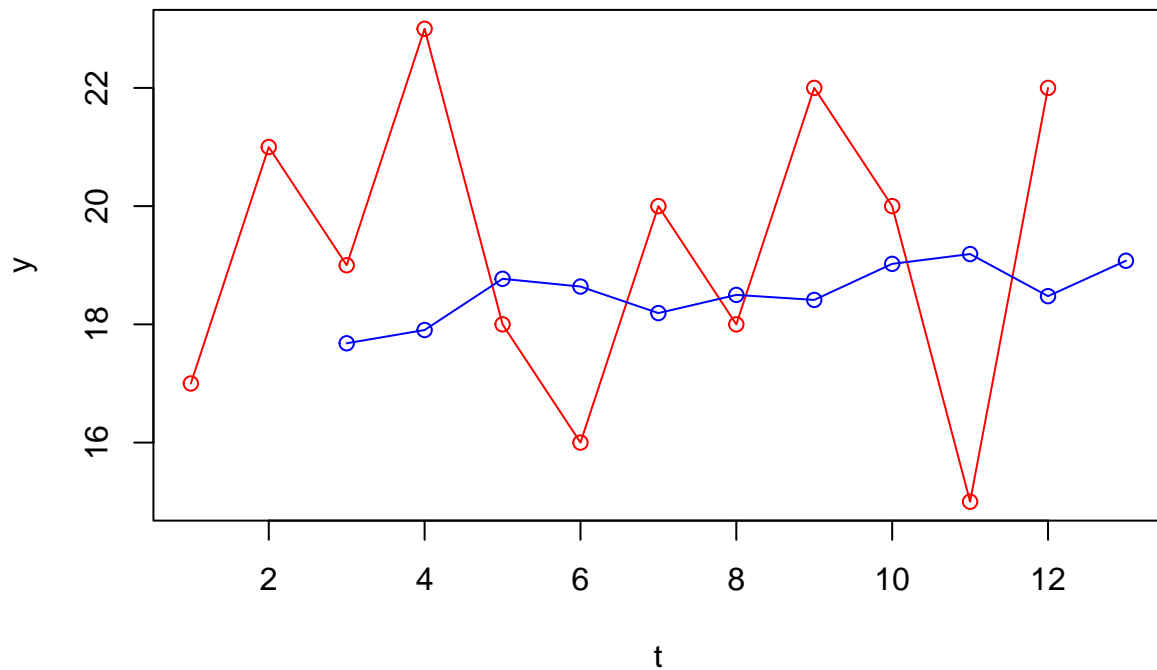
```

p4[2] = y[1]
a = 0.17

for (i in 3:n) {
  p4[i] = a * y[i - 1] + (1 - a) * p4[i - 1]
  e4[i] = y[i] - p4[i]
}

# Utiliza plot() para graficar, incluyendo la semana 13 en la línea azul
plot(t, y, type = "o", col = "red")
x = 2:n + 1
lines(x, p4[x], type = "o", col = "blue")

```



```

cat("Predicción de ventas para la semana 13: ", p4[13])

```

```

## Predicción de ventas para la semana 13: 19.07608

```

Problema 2

a. Promedios móviles de tres días para suavizar la serie de tiempo

```

t = c(1:18)
y = c(81.32,81.10,80.38,81.34,80.54,80.62,79.54,79.46,81.02,80.98,80.8,81.44,81.48,80.75,80.48,80.01,80.01)
p = rep(NA, n)
e = rep(NA,n)
n = length(y)
k = 3 #Valor de k para promedios moviles
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]}

```

```

# Creamos dataframe para organizar los datos en una tabla
T=data.frame(t,y,p,e^2)

# Calculamos el cuadrado medio de los errores sin NA
CME = mean(e^2,na.rm=TRUE)
cat(CME)

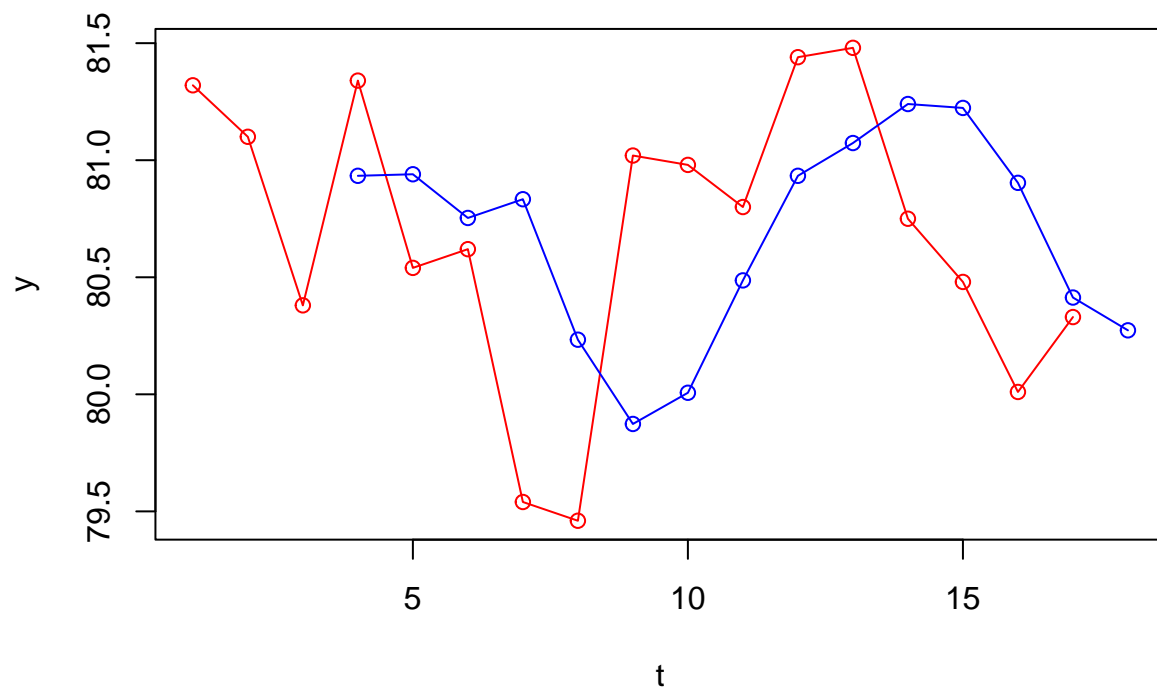
```

```
## 0.4995738
```

```

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

```



b. Suavizamiento exponencial

```

t = c(1:18)
y = c(81.32,81.10,80.38,81.34,80.54,80.62,79.54,79.46,81.02,80.98,80.8,81.44,81.48,80.75,80.48,80.01,80.32)

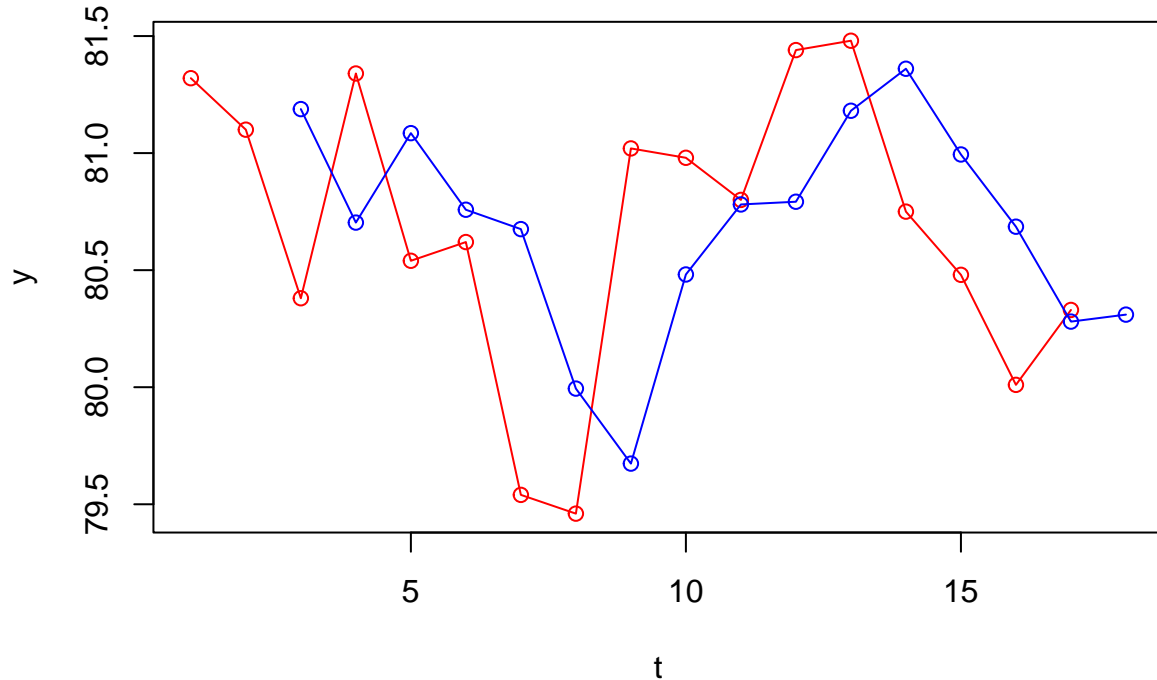
n = length(y)

# Utiliza el comando for para el cálculo de los pronósticos y los errores:
p = rep(NA, n)
e = rep(NA, n)
p[1] = y[1]
p[2] = y[1]
a = 0.6

for (i in 3:n) {
  p[i] = a * y[i - 1] + (1 - a) * p[i - 1]
  e[i] = y[i] - p[i]
}

# Utiliza plot() para graficar, incluyendo la semana 13 en la línea azul
plot(t, y, type = "o", col = "red")
x = 2:n + 1
lines(x, p[x], type = "o", col = "blue")

```



```

# Creamos dataframe para organizar los datos en una tabla
T=data.frame(t,y,p,e^2)

```



```
# Calculamos el cuadrado medio de los errores sin NA  
CME = mean(e^2, na.rm=TRUE)  
cat(CME)
```

```
## 0.4410245
```

c. Cual de los dos métodos se prefiere

EN este caso el CME obtenido por medio de la suavización exponencial es menor al realizado con medidas móviles, en este caso se escoge el modelo hecho con suavización exponencial.