7-intro-a-series-de-tiempo

October 31, 2023

1 7. Introducción a series de tiempo

Francisco Mestizo Hernández A01731549

1.1 Problema 1: Ventas de gasolina

Primero definimos los datos que se utilizaran para el problema en los vectores t y y

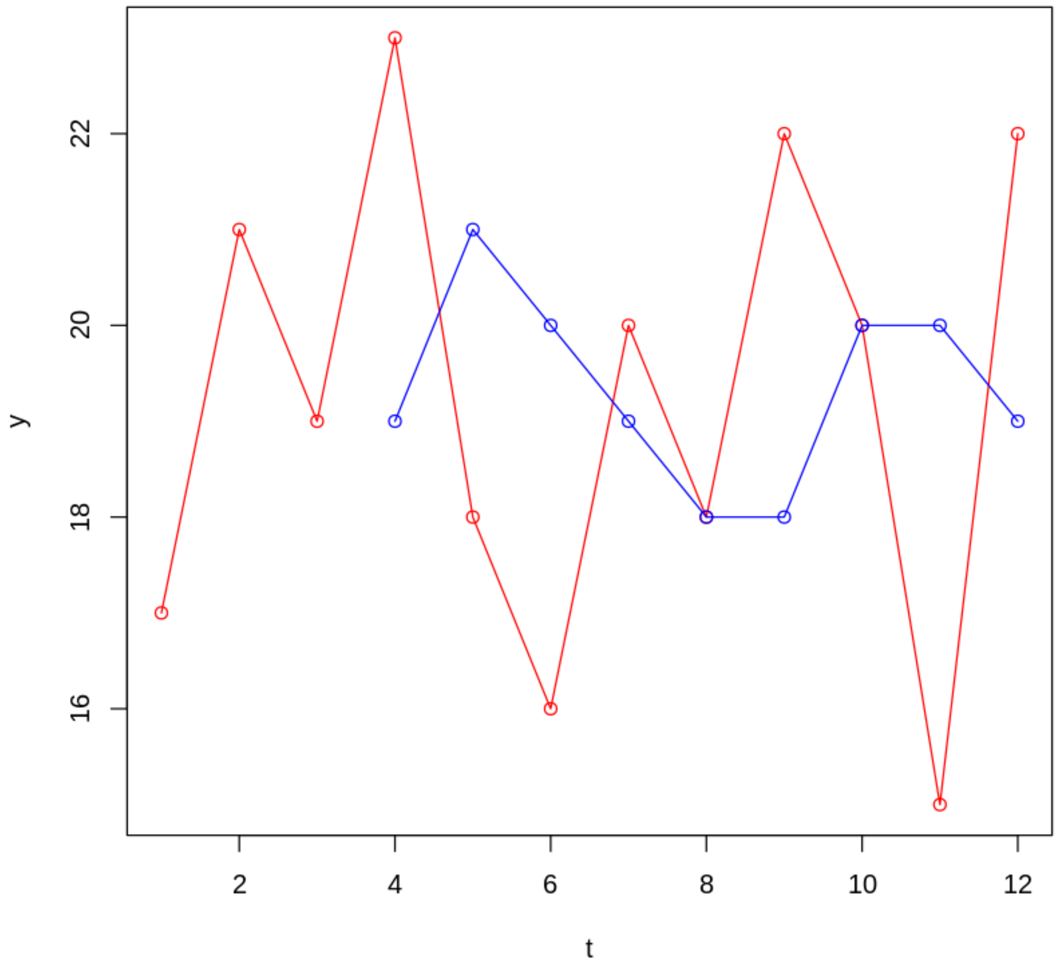
[85]:
```
t <- c(1,2,3,4,5,6,7,8,9,10,11,12)
y <- c(17, 21, 19, 23, 18, 16, 20, 18, 22, 20, 15, 22)
n <- length(t)
```

1.1.1 Promedios moviles

Primero, se hará una suavizacion por promedios moviles. Esta suavización nos da un CME de **10.22**

[86]:
```
# Suavizamiento por 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]}

# Utiliza data.frame() para organizar una tabla:
T=data.frame(t,y,p,e^2)

#Calcula el cuadrado medio de los errores sin NA:

CME=mean(e^2,na.rm=TRUE)

#Utiliza plot() para graficar:

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

CME
```

10.2222222222222



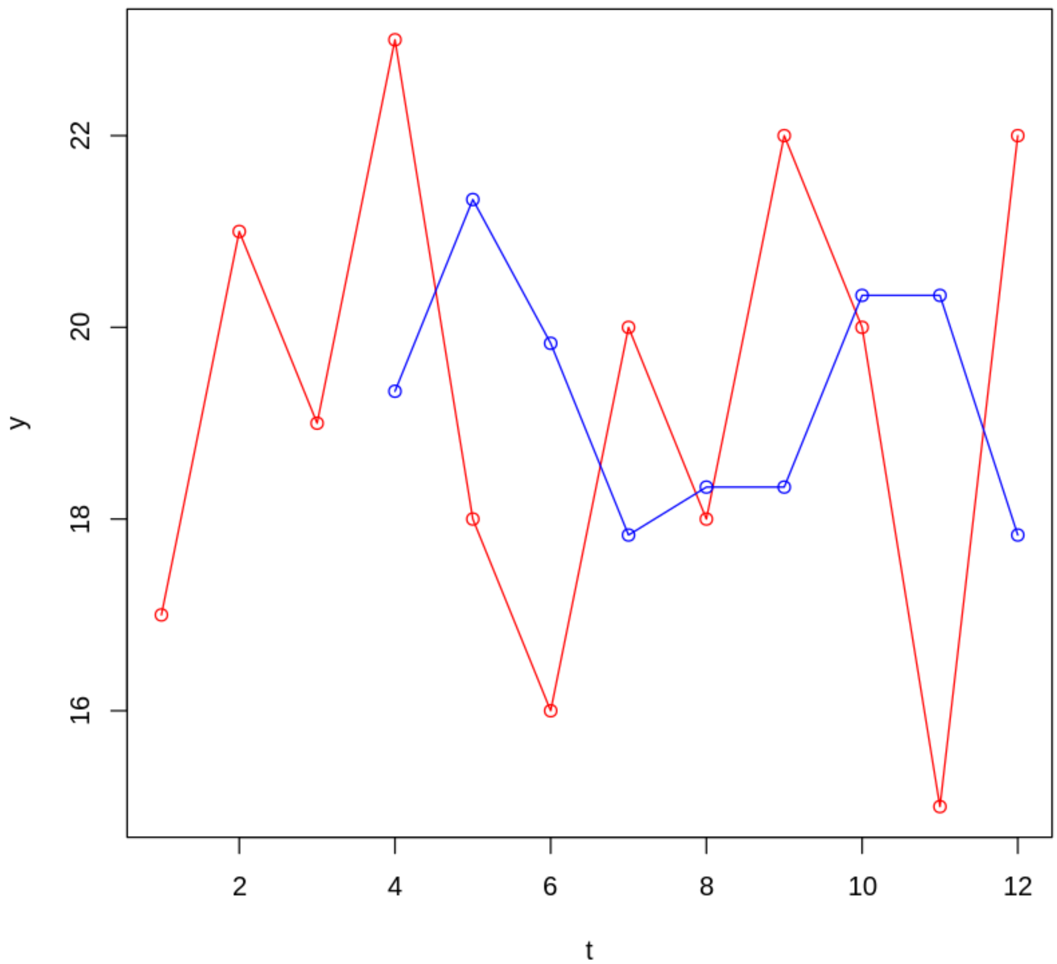
1.1.2 Promedios móviles ponderados

Como segundo metodo de suavizacion se hará por promedios moviles ponderados. El CME de este metodo nos da **11.50**

[87]:
```
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)
#Utiliza plot() para graficar:
plot(t, y, type="o", col="red")
x = (3+1):n
lines(x,p2[x],type="o",col="blue")

CME2
```

11.4907407407407



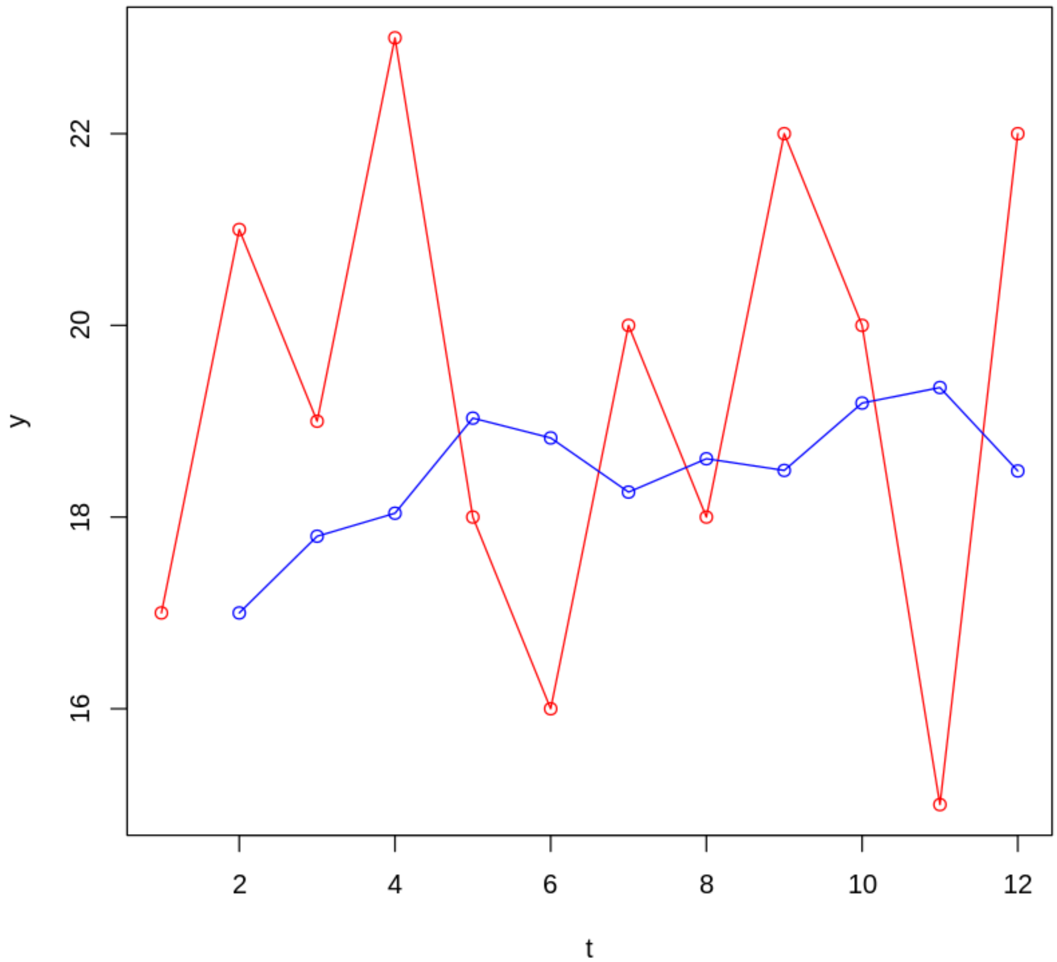
1.1.3 Suavizamiento exponencial

Finalmente haremos una suavizacion exponencial con un valor de alpha de 0.2. El CME de este metodo nos da **8.28**

[88]:

```
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]}
T3=data.frame(t,y, p3,e3^2)
#Calcula el cuadrado medio de los errores sin NA:
CME3=mean(e3^2,na.rm=TRUE)
#Utiliza plot() para graficar:
plot(t, y, type="o", col="red")
x = 2:n
lines(x,p3[x],type="o",col="blue")

CME3
```

8.28045374265132



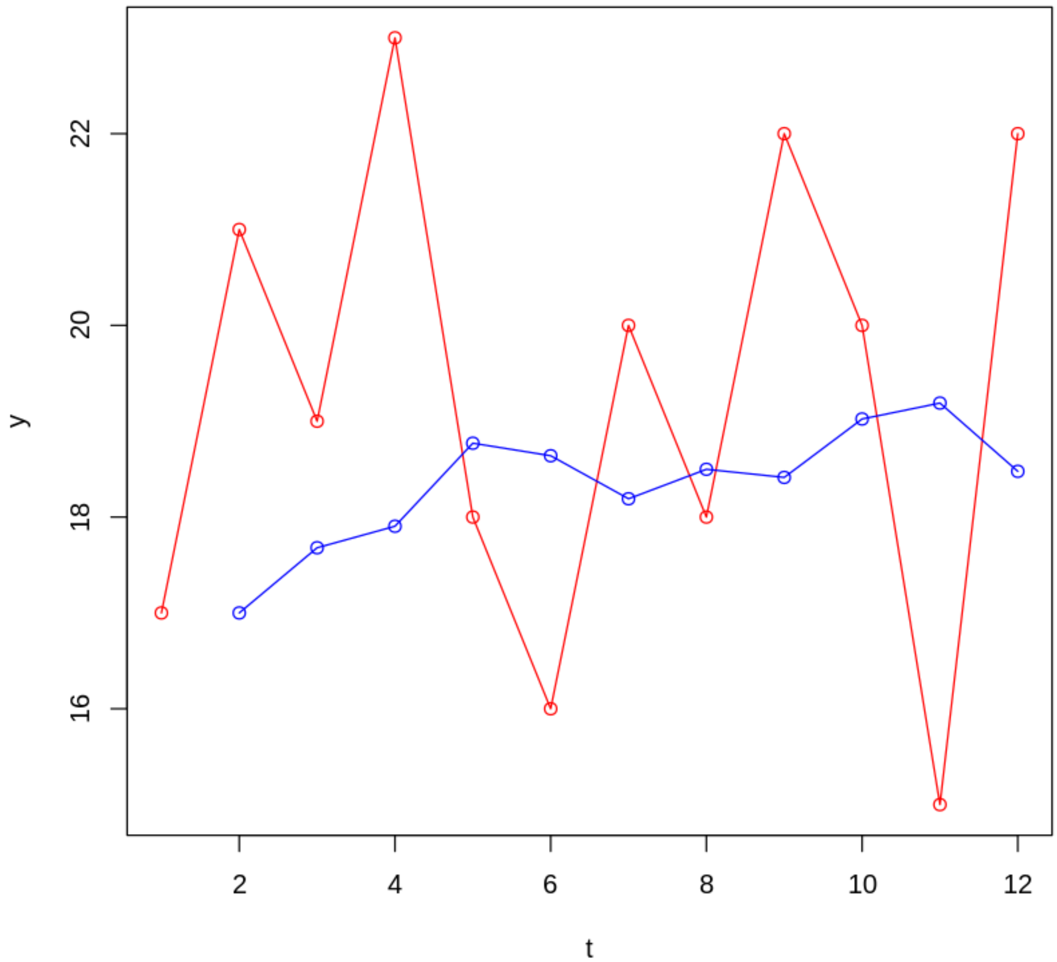
1.1.4 Suavizamiento exponencial con mejor alpha

Debido a que este metodo utiliza un alpha para hacer las predicciones, podemos buscar el mejor alpha que nos de las predicciones con menor error. Como se puede ver, se obtiene que el mejor alpha es de **0.17**, dando un CME de **8.26**

[89]:
```
minCME = Inf
minp3 = NA
minAlpha = NA
for(j in seq(0, 1, by = 0.01)){
  p3 = NA
  e3 = NA
  p3[1]=y[1]
```

```
  p3[2]=y[1]
  a=j
  for(i in 3:n){p3[i]=a*y[i-1]+(1-a)*p3[i-1];
  e3[i] = y[i]- p3[i]}
  T3=data.frame(t,y, p3,e3^2)
  #Calcula el cuadrado medio de los errores sin NA:
  CME3=mean(e3^2,na.rm=TRUE)
  if(CME3 < minCME){
    minCME = CME3
    T3=data.frame(t,y, p3,e3^2)
    minp3 = p3
    minAlpha = a
  }
  #cat("Alpha:", a, " CME: ", CME3, "\n")
}


cat("\n Min CME was ", minCME, " with alpha ", minAlpha, ", displaying plot...")
#Utiliza plot() para graficar:
plot(t, y, type="o", col="red")
x = 2:n
lines(x,minp3[x],type="o",col="blue")
```

```
 Min CME was  8.256687  with alpha  0.17 , displaying plot…
```



1.1.5 Predicción

Finalmente, podemos decir que el mejor modelo será el suavizamiento exponencial usando un alpha de 0.17. Por lo tanto, este es el elegido para hacer las predicciones de los siguientes valores. En este caso haremos una predicción para el 13 y el resultado que nos da es **19.08**

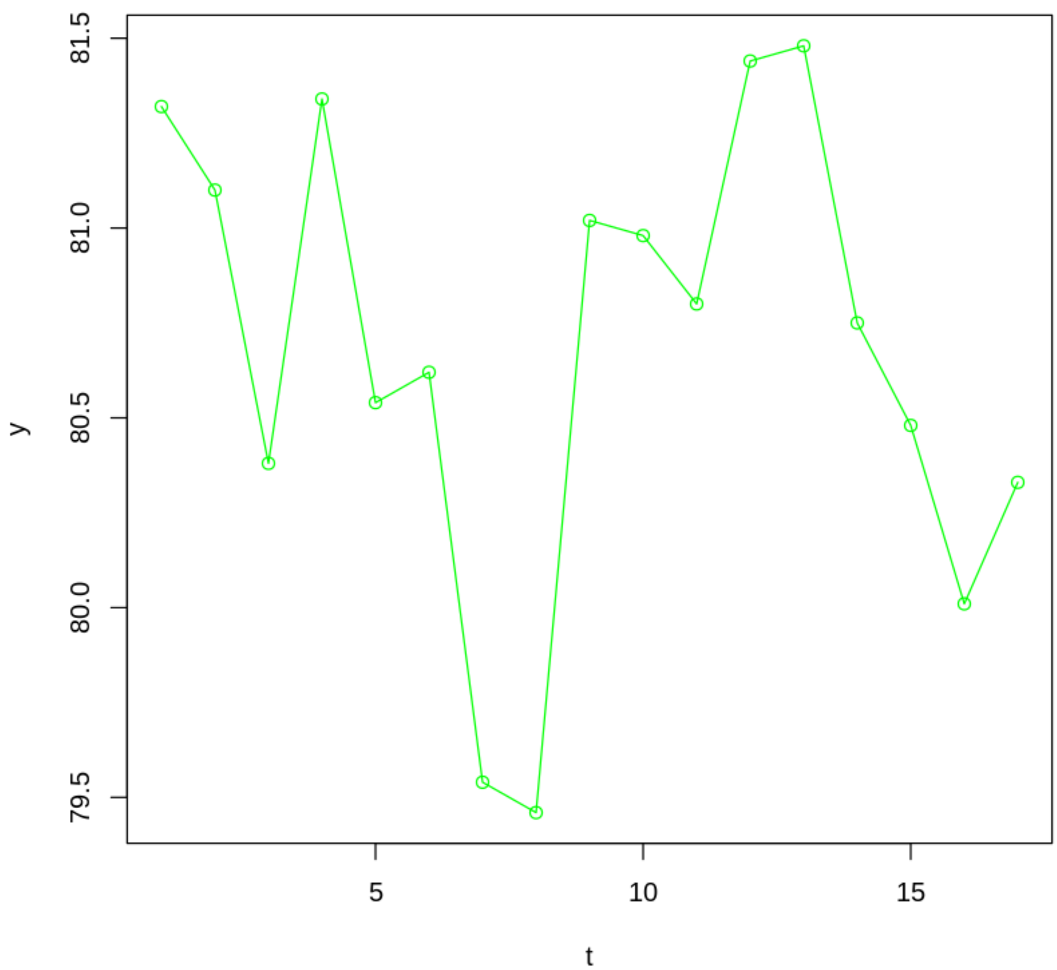
[90]:
```
a = minAlpha
i = 13
pred=a*y[i-1]+(1-a)*minp3[i-1]

pred
```

19.0760821662448

1.2 Problema 2: Acciones

Primero definimos los datos que se utilizaran para el problema en los vectores t y y. Además, verificamos que los datos pertenezcan a una serie estacionaria. Por lo que se puede ver en la gráfica, sí parece ser que es una serie estacionaria ya que los datos no parecen estar haciendo ningun ciclo.

[91]:
```
t <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17)
y <- c(81.32, 81.10, 80.38, 81.34, 80.54, 80.62, 79.54, 79.46, 81.02, 80.98, 80.
  ↪80, 81.44, 81.48, 80.75, 80.48, 80.01, 80.33)
n <- length(t)
plot(t, y, type="o", col="green")
```

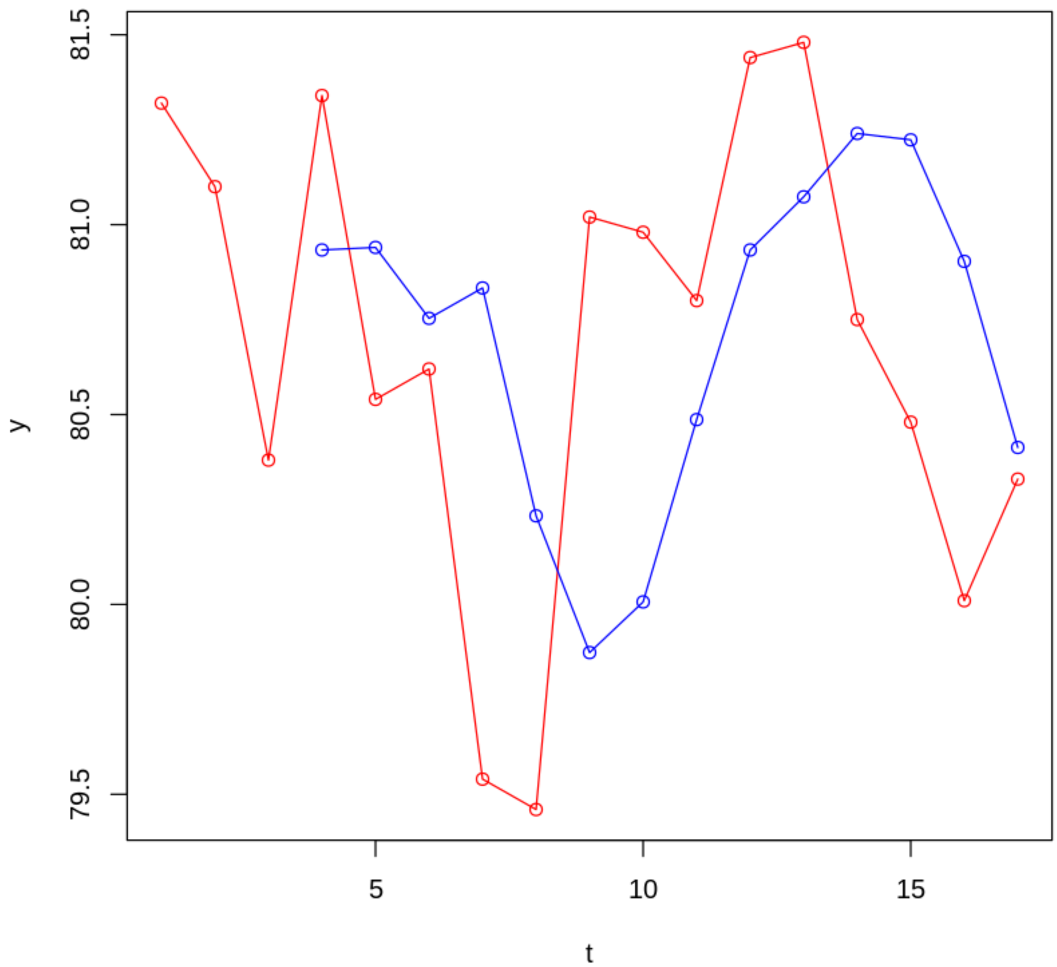


1.2.1 Promedios moviles

Primero haremos un suavizado con promedios moviles a tres dias. Podemos ver que con este método se tiene un error CME de **0.50**

[92]:
```
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]}

# Suavizamiento por promedios móviles
# Utiliza data.frame() para organizar una tabla:
T=data.frame(t,y,p,e^2)

#Calcula el cuadrado medio de los errores sin NA:

CME=mean(e^2,na.rm=TRUE)

#Utiliza plot() para graficar:

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

CME
```

0.499573809523813



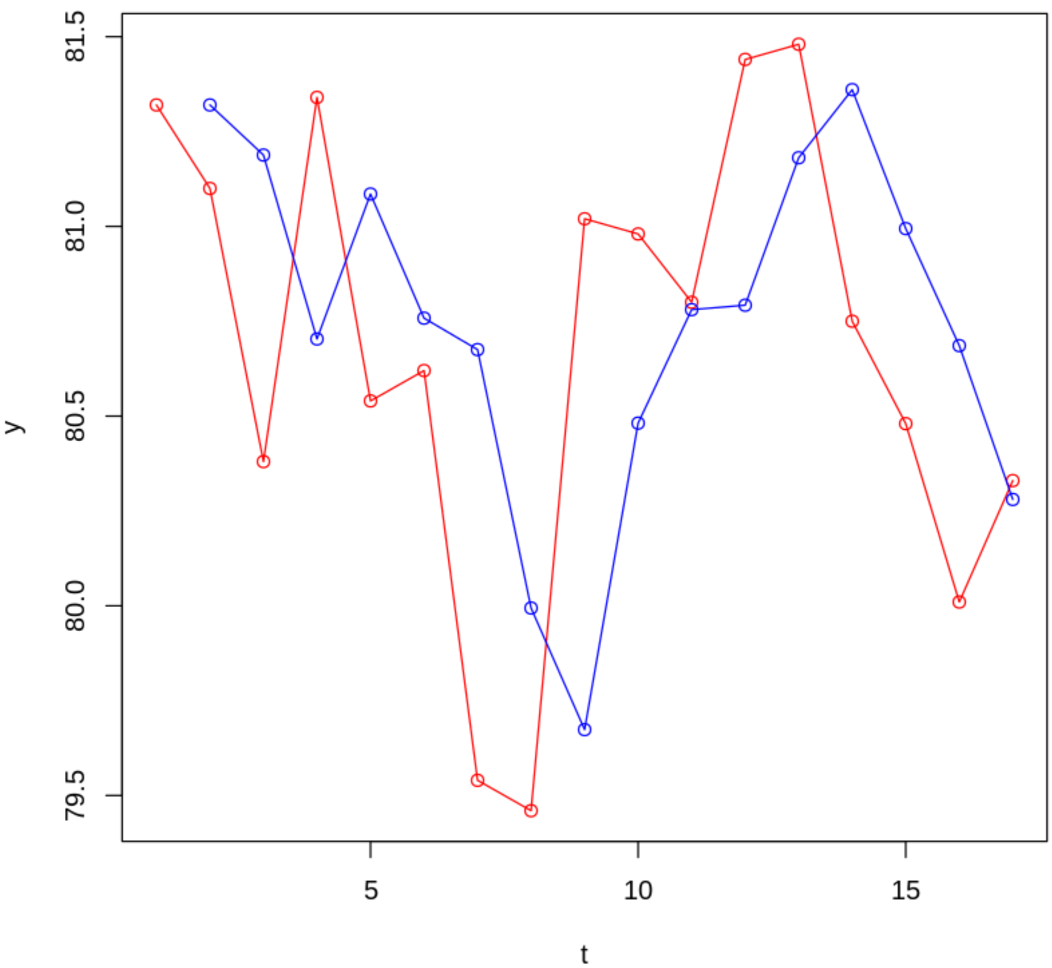
1.2.2 Suavizado exponencial

También probaremos el método de suavizado exponencial. Con este, obtenemos un CME **0.44**.

[93]:
```
p3 = NA
e3 = NA
p3[1]=y[1]
p3[2]=y[1]
a=0.6
for(i in 3:n){p3[i]=a*y[i-1]+(1-a)*p3[i-1];
e3[i] = y[i]- p3[i]}
T3=data.frame(t,y, p3,e3^2)
#Calcula el cuadrado medio de los errores sin NA:
```

```
CME3=mean(e3^2,na.rm=TRUE)
#Utiliza plot() para graficar:
plot(t, y, type="o", col="red")
x = 2:n
lines(x,p3[x],type="o",col="blue")

CME3
```

0.441024545792299



1.2.3 Predicción

Ahora debemos elegir alguno de los dos modelos para hacer alguna predicción. La selecció la haremos basandonos en el CME, ya que el que tenga el menor error nos dará una mejor predicción.

Por esto, usaremos el segundo (suavizado exponencial) ya que tiene un error de 0.44 en comparacion con el 0.50 de error del suavizado por promedios móviles.

La prediccion para el siguiente dia es de **80.31**

[94]:
```
i=18
a=0.6
pred=a*y[i-1]+(1-a)*p3[i-1]
pred
```

80.3101051459861