Actividad 7 - Introducción a Series Tiempo

Frida Cano Falcón - A01752953

2023-10-31

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:

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

Promedios Móviles

```r
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(y)
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,y,p,e^2)
T
```

```
##     t  y  p e.2
## 1   1 17 NA  NA
## 2   2 21 NA  NA
## 3   3 19 NA  NA
## 4   4 23 19  16
## 5   5 18 21   9
## 6   6 16 20  16
## 7   7 20 19   1
## 8   8 18 18   0
## 9   9 22 18  16
## 10 10 20 20   0
## 11 11 15 20  25
## 12 12 22 19   9
```

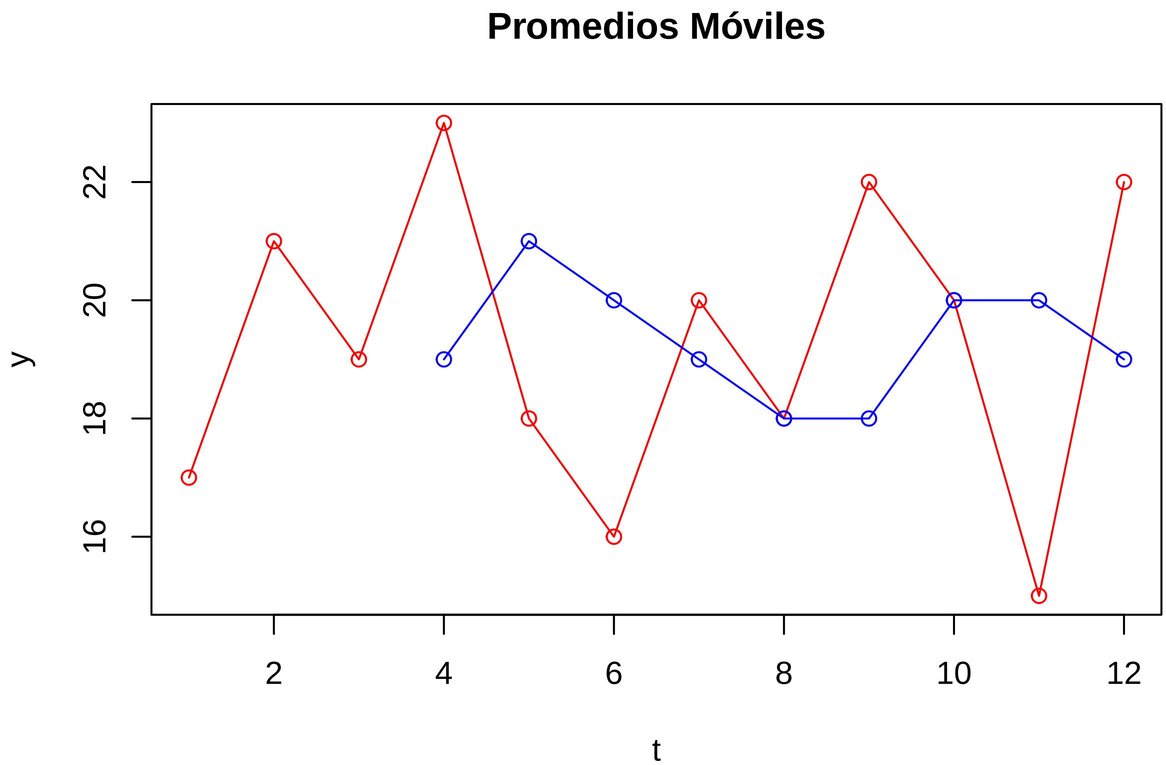
Calculo del cuadrado medio de los errores sin NA

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

```
## [1] 10.22222
```

Gráfica

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



##

Promedios Móviles Ponderados

```r
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,y,p2,e2^2)
T2
```

```
##    t  y       p2       e2.2
## 1  1 17       NA         NA
## 2  2 21       NA         NA
## 3  3 19       NA         NA
## 4  4 23 19.33333 13.4444444
## 5  5 18 21.33333 11.1111111
## 6  6 16 19.83333 14.6944444
## 7  7 20 17.83333  4.6944444
```

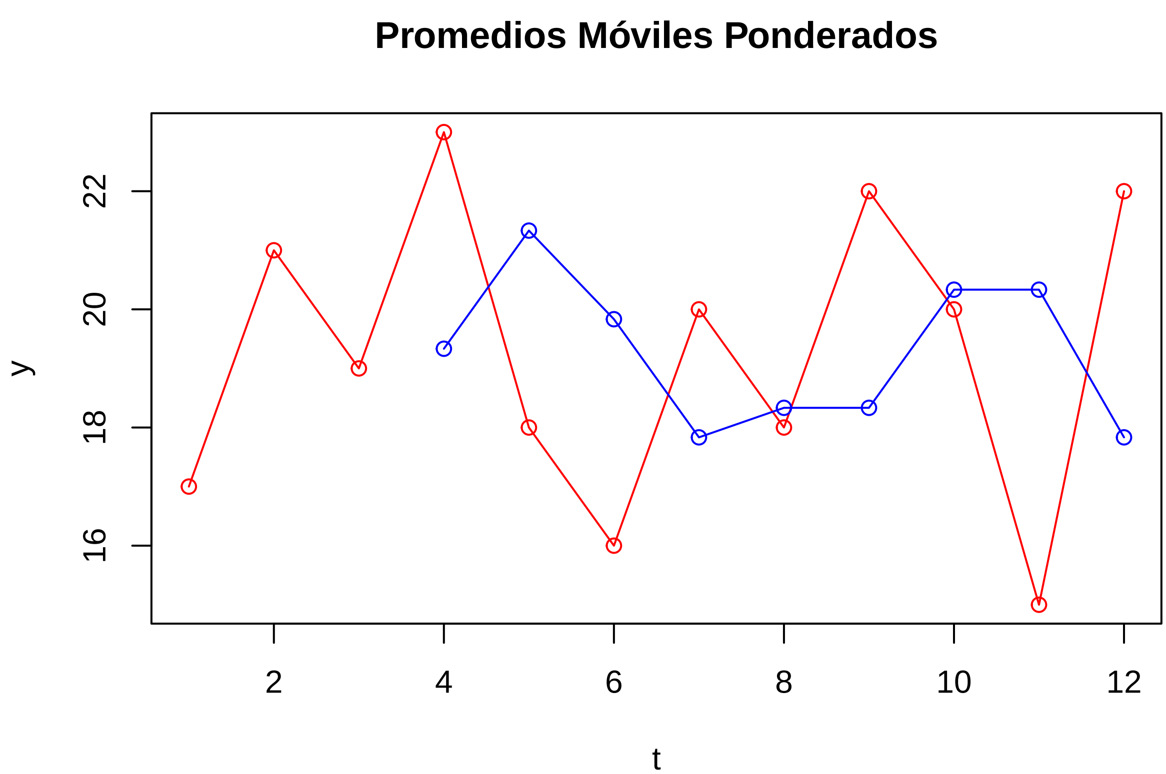
```
## 8    8 18 18.33333  0.1111111
## 9    9 22 18.33333 13.4444444
## 10  10 20 20.33333  0.1111111
## 11  11 15 20.33333 28.4444444
## 12  12 22 17.83333 17.3611111
```

```
CME2 = mean(e2^2, na.rm=TRUE)
CME2
```

```
## [1] 11.49074
```

Gráfica

```
plot(t,y,type="o", col="red")

title("Promedios Móviles Ponderados")
x=(3+1):n
lines(x,p2[x],type="o",col="blue")
```



##

Suavizamiento exponencial

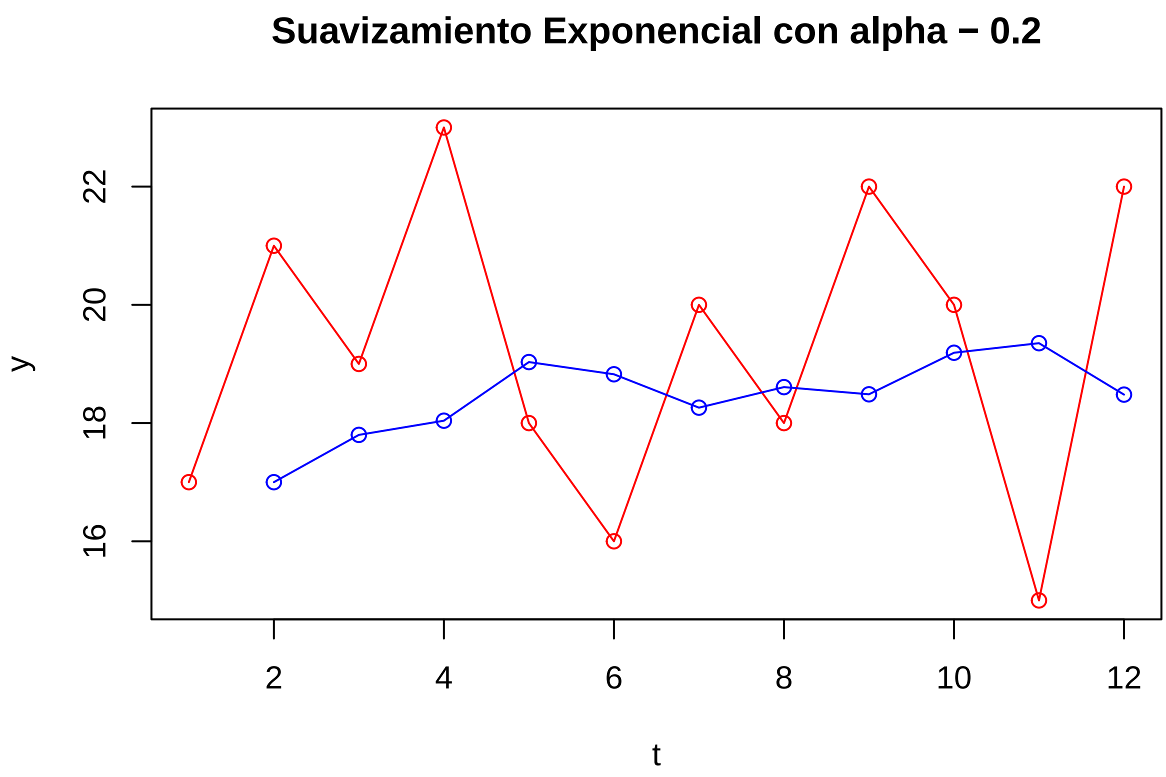
```
p3 = NA
e3 = NA
p3[1]=y[1]
p3[2]=y[1]
a=0.20
for(i in 3:n){
  p3[i]=a*y[i-1]+(1-a)*p3[i-1];
  e3[i]=y[i]-p3[i]
}
Te = data.frame(t,y,p3,e3^2)
Te
```

```
##     t  y       p3       e3.2
## 1   1 17 17.00000         NA
## 2   2 21 17.00000         NA
## 3   3 19 17.80000  1.4400000
## 4   4 23 18.04000 24.6016000
## 5   5 18 19.03200  1.0650240
## 6   6 16 18.82560  7.9840154
## 7   7 20 18.26048  3.0259298
## 8   8 18 18.60838  0.3701311
## 9   9 22 18.48671 12.3432263
## 10 10 20 19.18937  0.6571279
## 11 11 15 19.35149 18.9354879
## 12 12 22 18.48119 12.3819951
```

```r
CMEe = mean(e3, na.rm=TRUE)
CMEe
```

```
## [1] 0.6924776
```

```r
plot(t,y,type="o", col="red")
title("Suavizamiento Exponencial con alpha - 0.2")
x=2:n
lines(x,p3[x],type="o",col="blue")
```



Función que permite evaluar varios valores de α en el método de suavizamiento exponencial hasta encontrar el valor de α que minimice el CME.

```r
alphas <- seq(0.01, 1, by = 0.01)  # Valores de alfa a probar
min_cme <- Inf  # Inicializamos el valor mínimo del CME con un valor grande
best_alpha <- NA
best_p <- numeric(n)
best_e <- numeric(n)
```

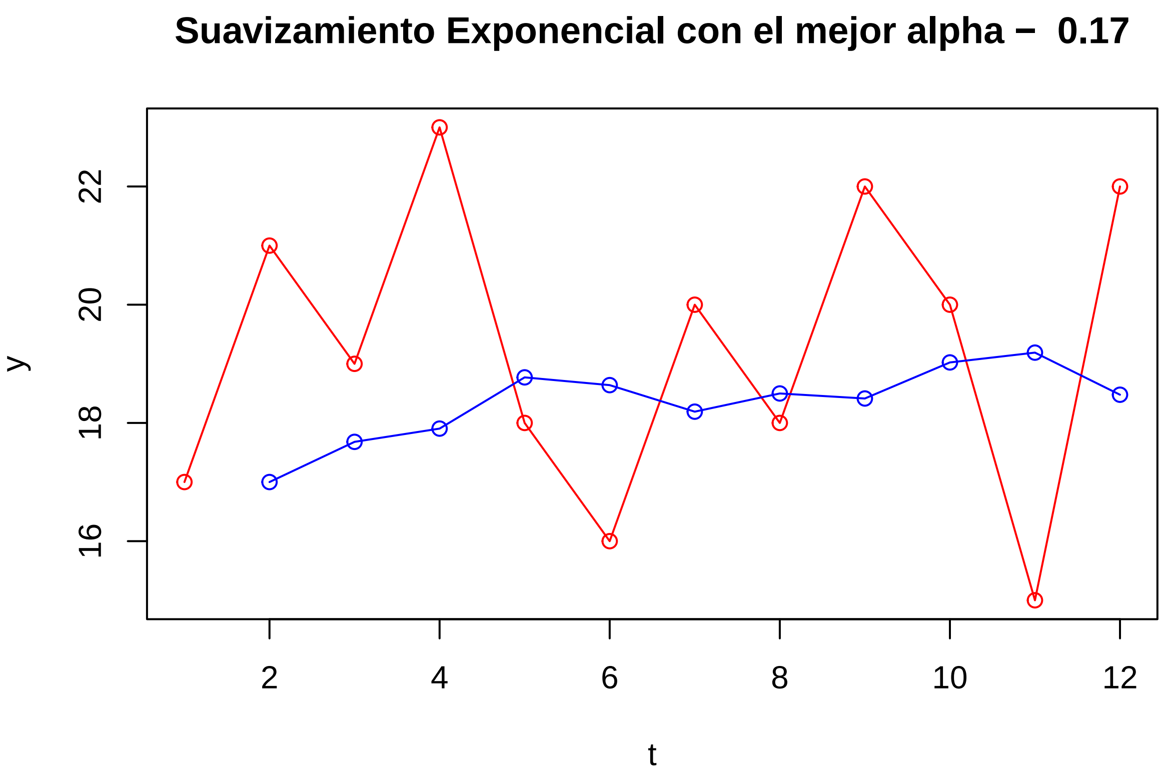
```r
for (a in alphas) {
  pE <- numeric(n)
  eE <- numeric(n)
  pE[1] <- y[1]
  pE[2] <- y[1]

  for (i in 3:n) {
    pE[i] <- a * y[i - 1] + (1 - a) * pE[i - 1]
    eE[i] <- y[i] - pE[i]
  }

  cme <- sum(eE^2) / n  # Calculamos el CME para el valor actual de alfa

  if (cme < min_cme) {
    min_cme <- cme
    best_alpha <- a
    best_p <- pE
    best_e <- eE
  }
}

Te <- data.frame(t, y, best_p, best_e^2)
cat("Mejor valor de alfa:", best_alpha, "\n")
```

```
## Mejor valor de alfa: 0.17
```

```r
Te
```

```
##     t  y   best_p   best_e.2
## 1   1 17 17.00000  0.0000000
## 2   2 21 17.00000  0.0000000
## 3   3 19 17.68000  1.7424000
## 4   4 23 17.90440 25.9651394
## 5   5 18 18.77065  0.5939045
## 6   6 16 18.63964  6.9677055
## 7   7 20 18.19090  3.2728350
## 8   8 18 18.49845  0.2484512
## 9   9 22 18.41371 12.8614580
## 10 10 20 19.02338  0.9537839
## 11 11 15 19.18941 17.5511272
## 12 12 22 18.47721 12.4100675
```

```r
plot(t,y,type="o", col="red")
title("Suavizamiento Exponencial con el mejor alpha -  0.17")
x=2:n
lines(x,best_p[x],type="o",col="blue")
```



###

Conclusiones

Gracias a las gráficas de comportamiento de las predicciones y los valores reales a través del tiempo para cada modelo, es evidente que el valor que se aproxima al comportamiento de una función es el suavizamiento exponencial, ya que sus predicciones resultan ser más apegadas a la realidad que con los otros.

Es por ello que para realizar la predicción de la semana 13 utilizaremos el método de predicción de suavizamiento exponencial.

```
pred_19 = p3[12] + (a * (y[12]-p3[12]))
cat("Predicción: ", pred_19)
```

```
## Predicción:  22
```

Problema 2

Se registró el precio de las acciones de una compañía al cierre de cada día hábil del 24 de agosto al 16 de septiembre.

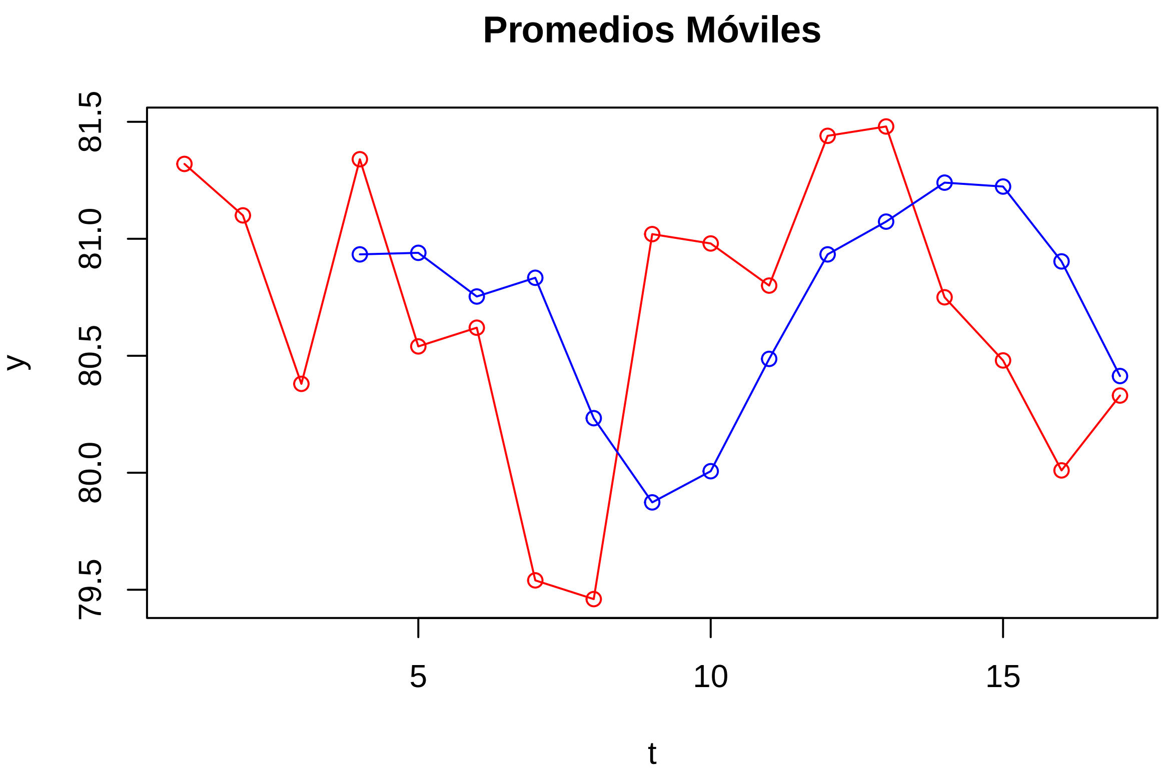
Promedio Móvil

```
t <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17)
y <- c(81.32,81.10,80.38,81.34,80.54,80.62,79.54,79.46,81.02,80.98,80.80,81.44,81.48,80.75,80.48,80.01,8
n <- length(y)

# Promedio Móvil
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]
```

```r
}
T = data.frame(t,y,p,e^2)
T
```

```
##     t     y        p         e.2
## 1   1 81.32       NA          NA
## 2   2 81.10       NA          NA
## 3   3 80.38       NA          NA
## 4   4 81.34 80.93333 0.165377778
## 5   5 80.54 80.94000 0.160000000
## 6   6 80.62 80.75333 0.017777778
## 7   7 79.54 80.83333 1.672711111
## 8   8 79.46 80.23333 0.598044444
## 9   9 81.02 79.87333 1.314844444
## 10 10 80.98 80.00667 0.947377778
## 11 11 80.80 80.48667 0.098177778
## 12 12 81.44 80.93333 0.256711111
## 13 13 81.48 81.07333 0.165377778
## 14 14 80.75 81.24000 0.240100000
## 15 15 80.48 81.22333 0.552544444
## 16 16 80.01 80.90333 0.798044444
## 17 17 80.33 80.41333 0.006944444
```

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



###

Predicción del precio del 19 de septiembre del 2005

```r
pred_19 = (p[15]+p[16]+p[17])/3
cat("Predicción: ", pred_19)
```

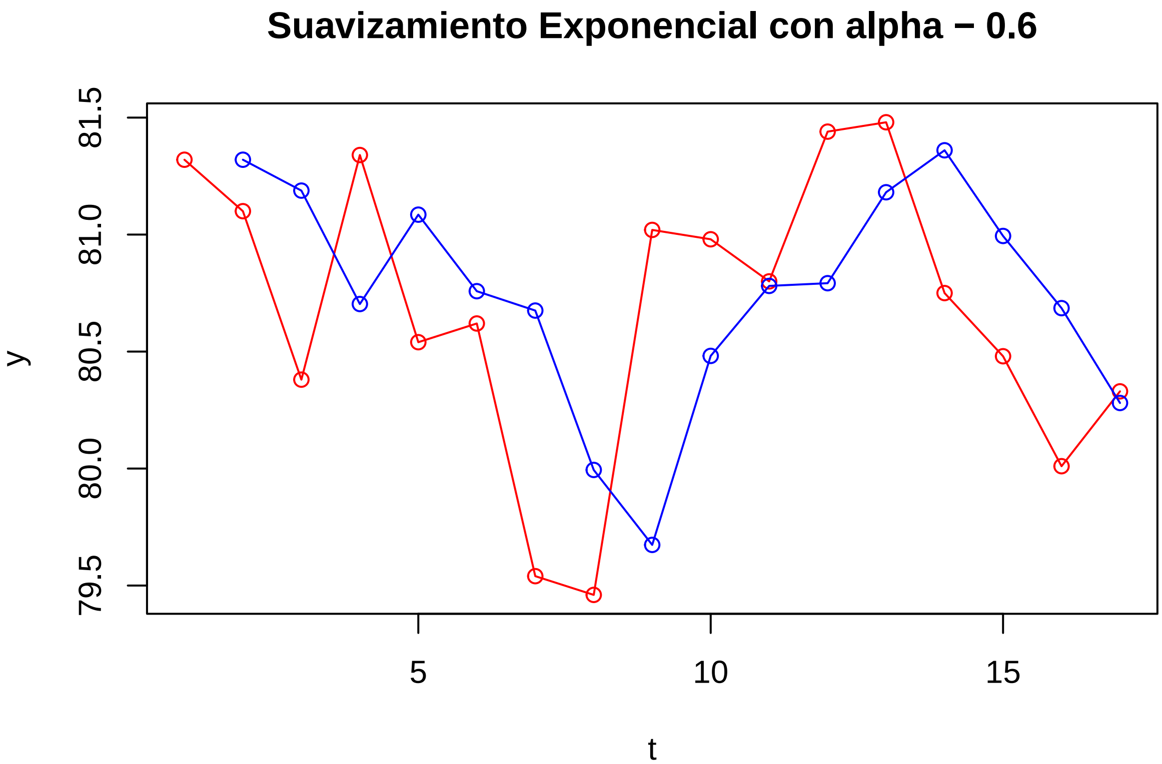
```
## Predicción:  80.84667
```

Suavizamiento exponencial

```r
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]
}
Te = data.frame(t,y,p3,e3^2)
Te
```

```
##     t     y       p3         e3.2
## 1   1 81.32 81.32000           NA
## 2   2 81.10 81.32000           NA
## 3   3 80.38 81.18800 0.6528640000
## 4   4 81.34 80.70320 0.4055142400
## 5   5 80.54 81.08528 0.2973302784
## 6   6 80.62 80.75811 0.0190749245
## 7   7 79.54 80.67524 1.2887807559
## 8   8 79.46 79.99410 0.2852605881
## 9   9 81.02 79.67364 1.8126874899
## 10 10 80.98 80.48146 0.2485464518
## 11 11 80.80 80.78058 0.0003770484
## 12 12 81.44 80.79223 0.4196022071
## 13 13 81.48 81.18089 0.0894649001
## 14 14 80.75 81.36036 0.3725359910
## 15 15 80.48 80.99414 0.2643429278
## 16 16 80.01 80.68566 0.4565126011
## 17 17 80.33 80.28026 0.0024737826
```

```r
plot(t,y,type="o", col="red")
title("Suavizamiento Exponencial con alpha - 0.6")
x=2:n
lines(x,p3[x],type="o",col="blue")
```



###

Predicción del precio del 19 de septiembre del 2005

```
pred_19 = p3[17] + (a * (y[17]-p3[17]))
cat("Predicción: ", pred_19)
```

```
## Predicción:  80.31011
```

Conclusiones

Vemmos que el método que realiza las mejores predicciones es el exponencial, al presentarnos un comrportamiento muy similar a través del tiempo con los datos reales.