

Filtrado Moving Average

Dr. Marcelo Risk

Data Mining de Series Temporales, Maestría en Explotación de Datos y
Descubrimiento de Conocimientos, FCEyN UBA

2020

Filtros de media móvil

Los filtros de media móvil (MA) calculan para cada muestra el promedio de una cantidad determinada de muestras a su alrededor, la fórmula de un filtro de media móvil de 3 muestras, denominado *MA3*:

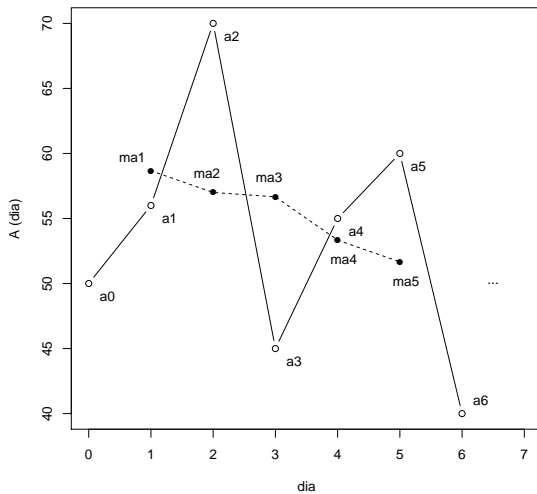
$$MA3(k) = \frac{a(k-1) + a(k) + a(k+1)}{3} \quad (1)$$

Como lo muestra la ecuación ??, la salida del filtro para cada muestra k , que va desde 1 hasta $N - 1$, se calcula como el valor medio o promedio entre el valor de la serie a filtrar A para dicha muestra k , la muestra anterior $k - 1$ y la posterior $k + 1$.

Representación en una tabla

dia	$A(dia)$	$MA3(dia)$
0	a_0	
1	a_1	$(a_0 + a_1 + a_2)/3$
2	a_2	$(a_1 + a_2 + a_3)/3$
3	a_3	$(a_2 + a_3 + a_4)/3$
4	a_4	$(a_3 + a_4 + a_5)/3$
5	a_5	$(a_4 + a_5 + a_6)/3$
...

Detalle del filtrado MA3



Otros filtros de media móvil

$$MA5(k) = \frac{a(k-2) + a(k-1) + a(k) + a(k+1) + a(k+2)}{5} \quad (2)$$

$$MA7(k) = \frac{a(k-3) + a(k-2) + a(k-1) + a(k) + a(k+1) + a(k+2) + a(k+3)}{7} \quad (3)$$

Generación de una línea de tiempo y una serie temporal + ruido uniforme

$N = 256$

tiempo = 0:(N-1)

ciclos = 2

amplitud = 1

seno2 = amplitud*sin(ciclos*2*pi*tiempo/N)

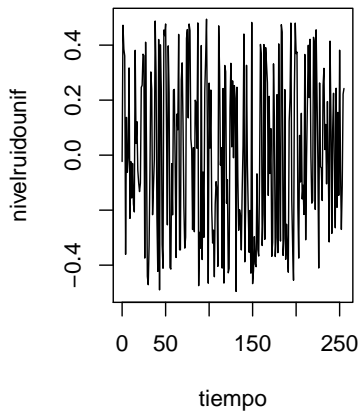
nivelruidounif = runif(N,min=-0.5,max=0.5)

op <- par(mfrow = c(1, 2))

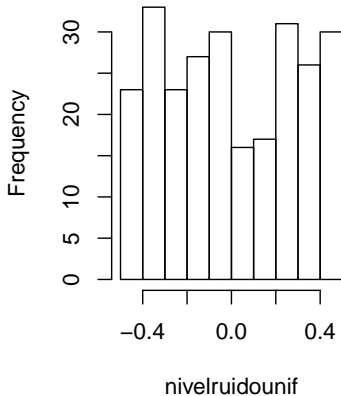
plot(tiempo, nivelruidounif ,type='l')

hist(nivelruidounif)

Generación de ruido uniforme



Histogram of nivelruidounif

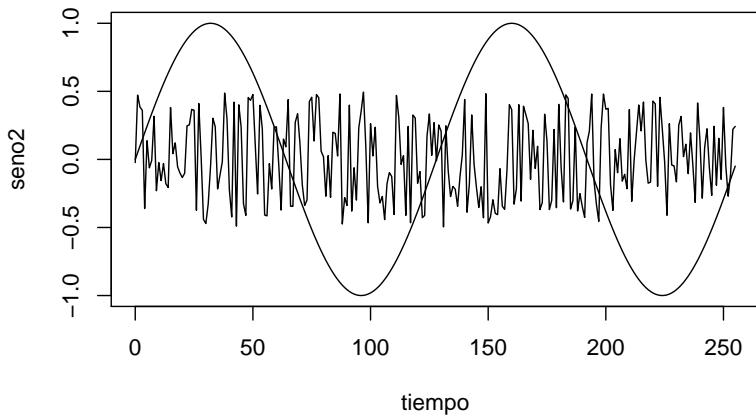


Generación de una línea de tiempo y una serie temporal + ruido uniforme

```
op <- par(mfrow = c(1, 1))
```

```
plot(tiempo,seno2,type='l')  
lines(tiempo, nivelruidounif )
```

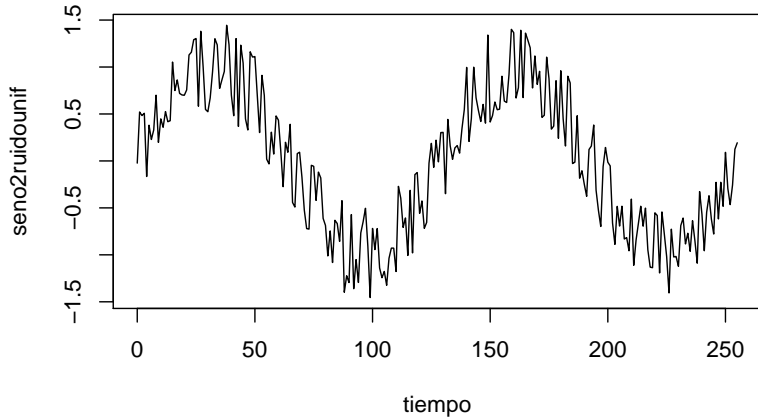
Generación de una línea de tiempo y una serie temporal + ruido uniforme



Generación de una línea de tiempo y una serie temporal + ruido uniforme

```
seno2ruidounif = seno2 + nivelruidounif  
plot(tiempo, seno2ruidounif , type='l')
```

Generación de una línea de tiempo y una serie temporal + ruido uniforme



Filtrado MA de 3 a 25 taps

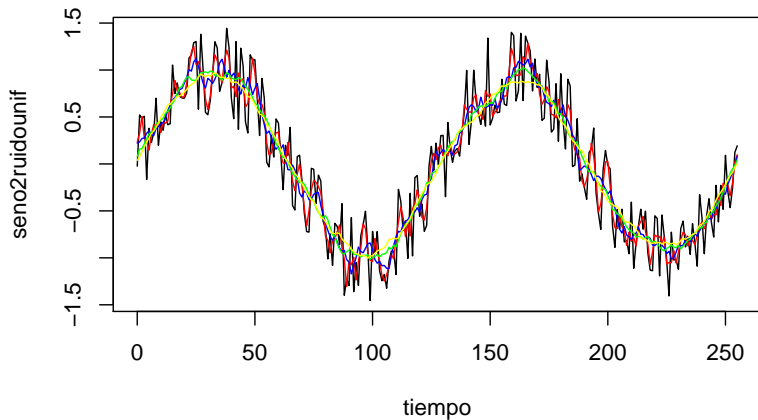
```
seno2ruidounif.MA3 =  
    filter( seno2ruidounif , rep(1/3,3) , circular =TRUE)  
lines ( tiempo, seno2ruidounif.MA3,col='red')
```

```
seno2ruidounif.MA7 =  
    filter( seno2ruidounif , rep(1/7,7) , circular =TRUE)  
lines ( tiempo, seno2ruidounif.MA7,col='blue')
```

```
seno2ruidounif.MA15 =  
    filter( seno2ruidounif , rep(1/15,15) , circular =TRUE)  
lines ( tiempo, seno2ruidounif.MA15,col='green')
```

```
seno2ruidounif.MA25 =  
    filter( seno2ruidounif , rep(1/25,25) , circular =TRUE)  
lines ( tiempo, seno2ruidounif.MA25,col='yellow')
```

Generación de una línea de tiempo y una serie temporal + ruido uniforme



Filtrado MA de 3 a 25 taps en el dominio de la frecuencia

```
fft . seno2ruidounif = fft( seno2ruidounif )
```

```
fft . seno2ruidounif . MA3 = fft(seno2ruidounif.MA3)
```

```
fft . seno2ruidounif . MA7 = fft(seno2ruidounif.MA7)
```

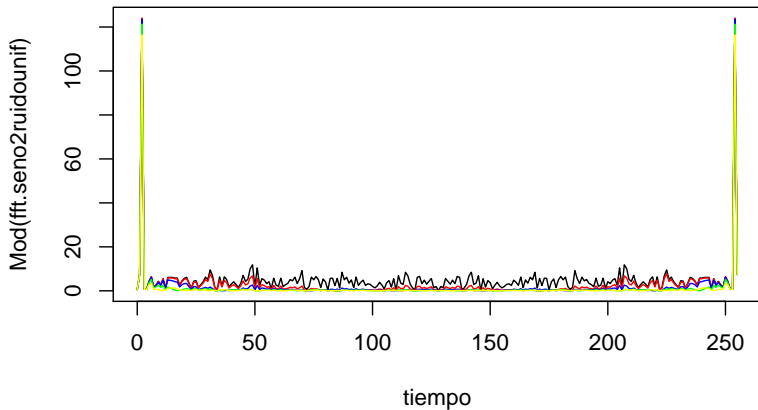
```
fft . seno2ruidounif . MA15 = fft(seno2ruidounif.MA15)
```

```
fft . seno2ruidounif . MA25 = fft(seno2ruidounif.MA25)
```

Filtrado MA de 3 a 25 taps en el dominio de la frecuencia

```
plot (tiempo, Mod(fft. seno2ruidounif ), type='l' )  
lines (tiempo, Mod(fft. seno2ruidounif . MA3), col='red')  
lines (tiempo, Mod(fft. seno2ruidounif . MA7), col='blue')  
lines (tiempo, Mod(fft. seno2ruidounif . MA15), col='green')  
lines (tiempo, Mod(fft. seno2ruidounif . MA25), col='yellow')
```

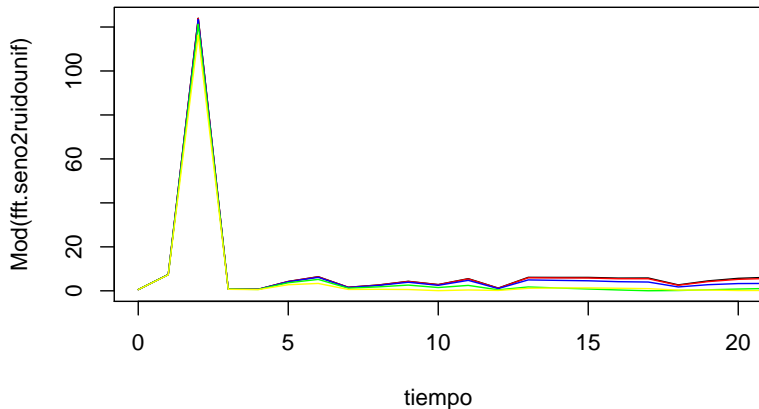
Filtrado MA de 3 a 25 taps en el dominio de la frecuencia



Filtrado MA de 3 a 25 taps en el dominio de la frecuencia
(zoom de las primeras armónicas)

```
plot (tiempo,Mod(fftw.seno2ruidounif),type='l',xlim=c(0,20))  
lines (tiempo,Mod(fftw.seno2ruidounif.MA3),col='red')  
lines (tiempo,Mod(fftw.seno2ruidounif.MA7),col='blue')  
lines (tiempo,Mod(fftw.seno2ruidounif.MA15),col='green')  
lines (tiempo,Mod(fftw.seno2ruidounif.MA25),col='yellow')
```

Filtrado MA de 3 a 25 taps en el dominio de la frecuencia



Respuesta en frecuencia de los MA

$\text{ma3} = \text{rep}(0, N)$

$\text{ma3}[1:3] = 1/3$

$\text{ma7} = \text{rep}(0, N)$

$\text{ma7}[1:7] = 1/7$

$\text{ma15} = \text{rep}(0, N)$

$\text{ma15}[1:15] = 1/15$

$\text{ma25} = \text{rep}(0, N)$

$\text{ma25}[1:25] = 1/25$

Respuesta en frecuencia de los MA

```
fft.ma3 = fft(ma3)
```

```
fft.ma7 = fft(ma7)
```

```
fft.ma15 = fft(ma15)
```

```
fft.ma25 = fft(ma25)
```

```
plot(tiempo, Mod(fft.ma3), type='l', col='red', xlim=c(0, N/2))
```

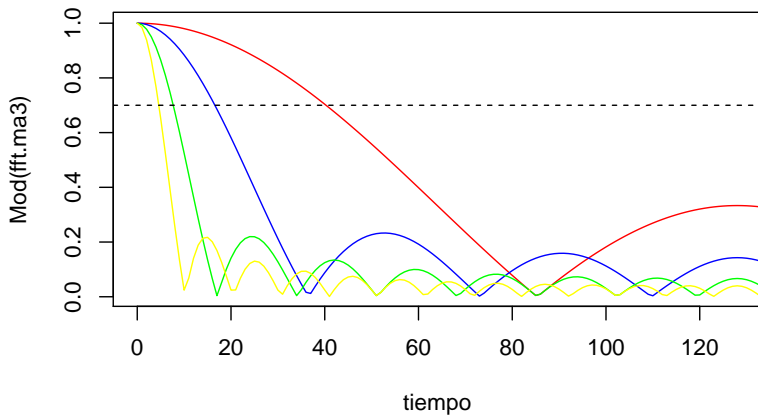
```
lines(tiempo, Mod(fft.ma7), col='blue')
```

```
lines(tiempo, Mod(fft.ma15), col='green')
```

```
lines(tiempo, Mod(fft.ma25), col='yellow')
```

```
abline(h=0.7, lty=2)
```

Respuesta en frecuencia de los MA



Descomposición de una Serie Temporal con filtros de MA

La descomposición de una Serie Temporal de una variable con un intervalo de muestreo de 1 día, durante un año, se puede descomponer de forma de estimar las contribuciones con periodicidad semanal y mensual, de acuerdo al siguiente algoritmo:

1. Filtre la señal original con un MA_7 , grafique el resultado.
2. Reste a la serie original la salida del paso anterior (MA_7), grafique el resultado.
3. Filtre la señal original con un MA_{21} , grafique el resultado.
4. Reste a la salida del MA_7 la salida del paso anterior (MA_{21}), grafique el resultado.

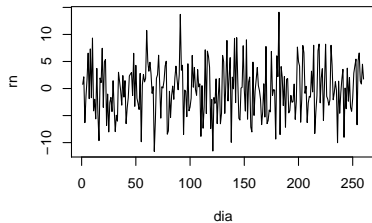
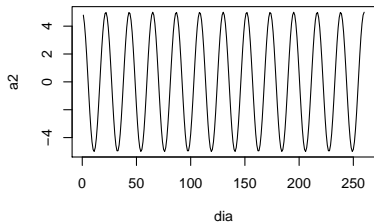
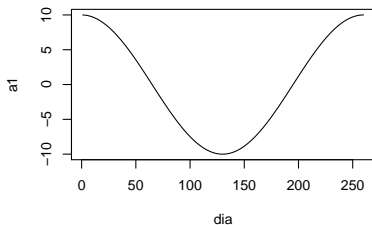
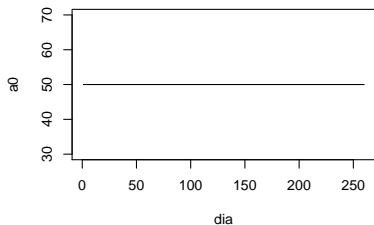
Descomposición de una Serie Temporal con filtros de MA

```
dir = '/Users/marcelorisk/Dropbox/MateriaDataMining/Clases/'
dato = read.csv(paste(dir, 'TS_1.csv', sep=""))
print(names(dato))
# [1] "X" "t" "a0" "a1" "a2" "rn" "a"

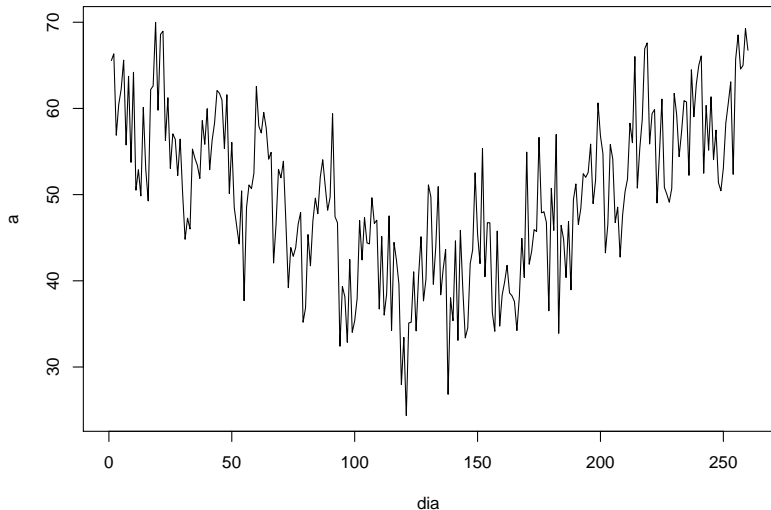
op <- par(mfrow = c(2, 2))
plot(dato$t, dato$a0, type='l', xlim=c(0,260), xlab='dia', ylab='a0')
plot(dato$t, dato$a1, type='l', xlim=c(0,260), xlab='dia', ylab='a1')
plot(dato$t, dato$a2, type='l', xlim=c(0,260), xlab='dia', ylab='a2')
plot(dato$t, dato$rn, type='l', xlim=c(0,260), xlab='dia', ylab='rn')

op <- par(mfrow = c(1, 1))
plot(dato$t, dato$a, type='l', xlim=c(0,260), xlab='dia', ylab='a')
```

Descomposición de una Serie Temporal con filtros de MA



Descomposición de una Serie Temporal con filtros de MA

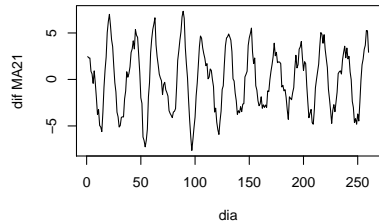
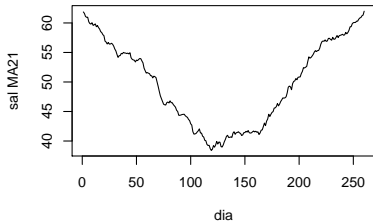
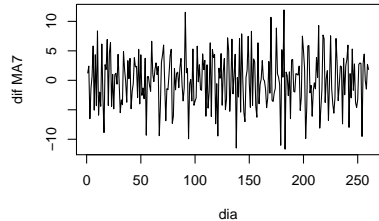
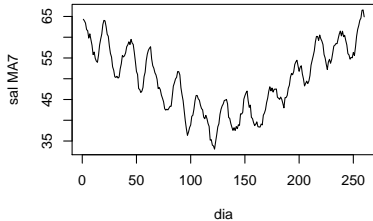


Descomposición de una Serie Temporal con filtros de MA

```
outMA7 <- filter(dato$a,rep(1/7,7), circular =TRUE)
difMA7 <- dato$a - outMA7
outMA21 <- filter(dato$a,rep(1/21,21), circular =TRUE)
difMA21 <- outMA7 - outMA21

op <- par(mfrow = c(2, 2))
plot(dato$t,outMA7,type='l',xlab='dia', ylab='sal MA7')
plot(dato$t,difMA7,type='l',xlab='dia', ylab='dif MA7')
plot(dato$t,outMA21,type='l',xlab='dia', ylab='sal MA21')
plot(dato$t,difMA21,type='l',xlab='dia', ylab='dif MA21')
```

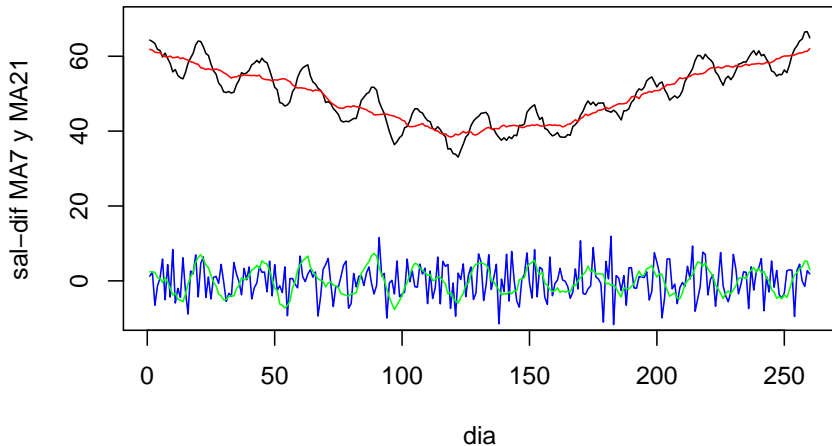
Descomposición de una Serie Temporal con filtros de MA



Descomposición de una Serie Temporal con filtros de MA

```
op <- par(mfrow = c(1, 1))
plot(dato$t, outMA7, type='l', ylim=c(-10,70), xlab='dia', ylab='sal - dif
      MA7 y MA21', col='black')
lines(dato$t, difMA7, col='blue')
lines(dato$t, outMA21, col='red')
lines(dato$t, difMA21, col='green')
```

Descomposición de una Serie Temporal con filtros de MA



Descomposición de una Serie Temporal con filtros de MA

```
fft.a = Mod(fft(dato$a))  
fft.outMA7 = Mod(fft(outMA7))  
fft.difMA7 = Mod(fft(difMA7))  
fft.outMA21 = Mod(fft(outMA21))  
fft.difMA21 = Mod(fft(difMA21))
```

```
op <- par(mfrow = c(1, 1))  
plot(fft.a  
      , type='l' , col='black' , ylim=c(0,800),xlim=c(0,128), lty=2)  
lines(fft.outMA7,col='black')  
lines(fft.difMA7,col='blue')  
lines(fft.outMA21,col='red')  
lines(fft.difMA21,col='green')
```

Descomposición de una Serie Temporal con filtros de MA

