

Convolución y correlación de series temporales

Dr. Marcelo Risk

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

2020

Convolución de dos series de tiempo

- ▶ El procesamiento una serie temporal con un sistema lineal, con un filtro por ejemplo, si se produce la operación en el DT se denomina **convolución**, resultando en una nueva serie temporal.
- ▶ Por otro lado si representamos dicha operación en el DF, se transforma en un producto algebraico simple de componente a componente.
- ▶ entonces en el DT:

$$x(t) * h(t) = y(t)$$

- ▶ se convierte en el DF:

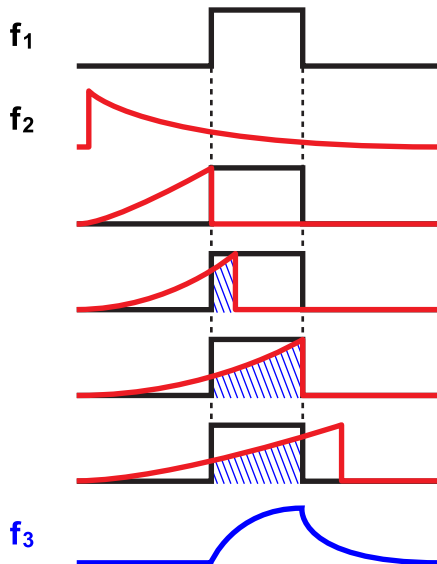
$$X(f)H(f) = Y(f)$$

Convolución en el DT

- La expresión de la **convolución** en el DT:

$$y(n) = \sum_{k=0}^{N-1} x(n-k)h(k)$$

Convolución en el DT



Convolución en el DT

`N = 256`

`tiempo = 0:(N-1)`

`# Convolucion de dos series de tiempo en el DT`

`# f1 pulso`

`f1 = rep(0,N)`

`f1[100:200] = 1`

`# f2 exponencial decreciente`

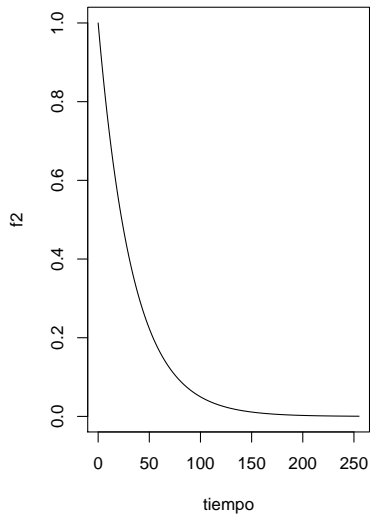
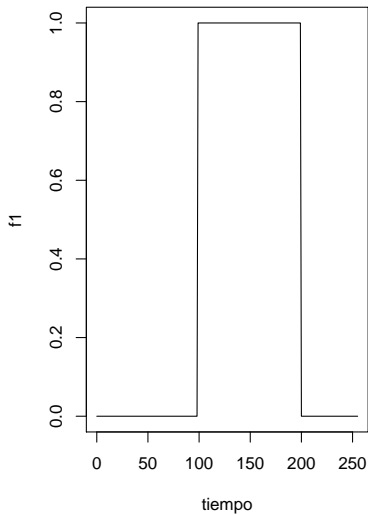
`f2 = 1*exp(-0.03*tiempo)`

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

`plot(tiempo,f1,type='l')`

`plot(tiempo,f2,type='l')`

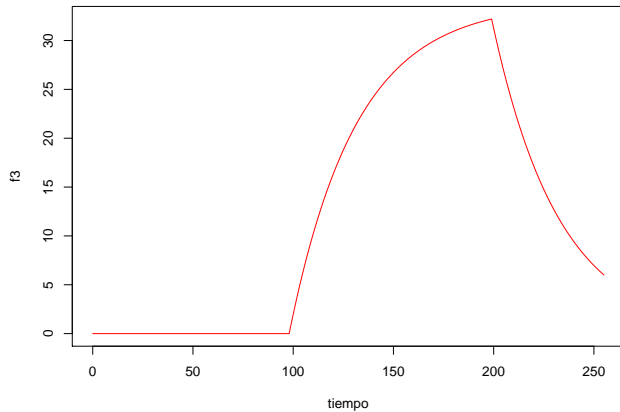
Convolución en el DT



Convolución en el DT

```
# algoritmo de convolucion en el dominio del tiempo
f3 = rep(NA,N)
for(i in 1:N)
{
  suma = 0
  j = 1
  while(j<=i)
  {
    suma = suma + f1[j]*f2[i-j+1]
    j = j+1
  }
  f3[i] = suma
  print(i)
}
plot(tiempo,f3,type='l',col='red')
```

Convolución en el DT



Convolución en el DF

```
# convolucion en el DF
```

```
fft.f1 = fft(f1)
```

```
fft.f2 = fft(f2)
```

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

```
plot(Mod(fft.f1), type='l')
```

```
plot(Mod(fft.f2), type='l')
```

```
fft.f3 = fft.f1 * fft.f2
```

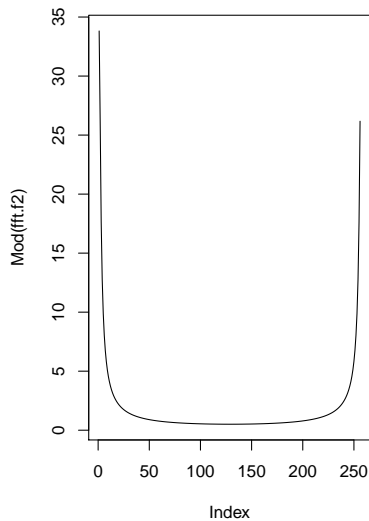
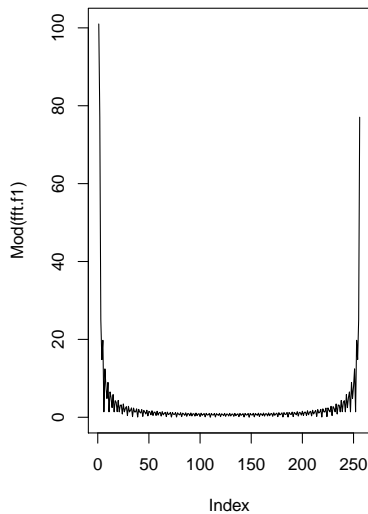
```
plot(Mod(fft.f3), type='l')
```

```
f3b = Re(fft(fft.f3, inverse=TRUE)/N)
```

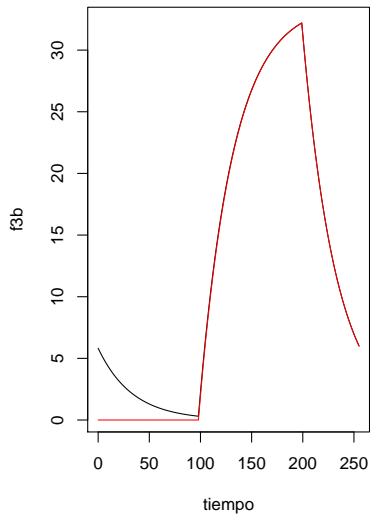
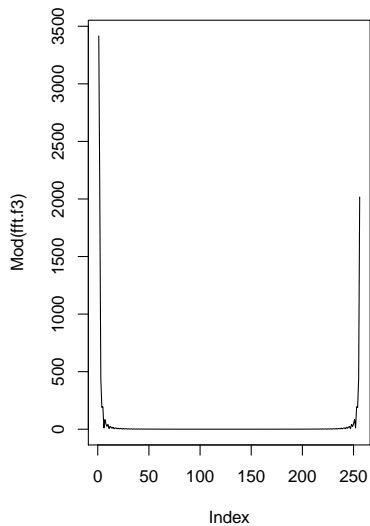
```
plot(tiempo, f3b, type='l')
```

```
lines(tiempo, f3, col='red')
```

Convolución en el DF



Convolución en el DF

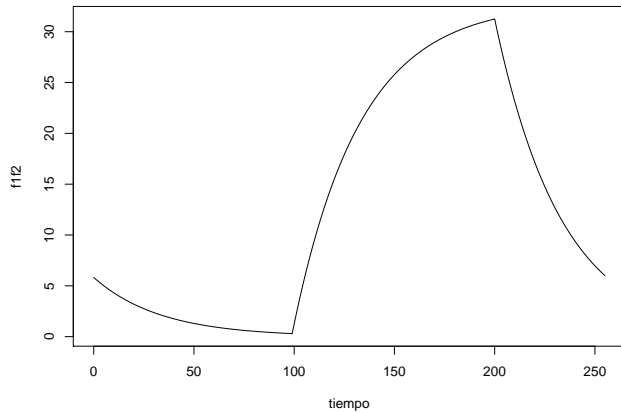


Convolución con **convolve**

```
# convolucion con convolve
```

```
f1f2 = convolve(f1, f2, conj = FALSE)  
plot(tiempo, f1f2, type='l')
```

Convolución con **convolve**



Correlación de dos series de tiempo

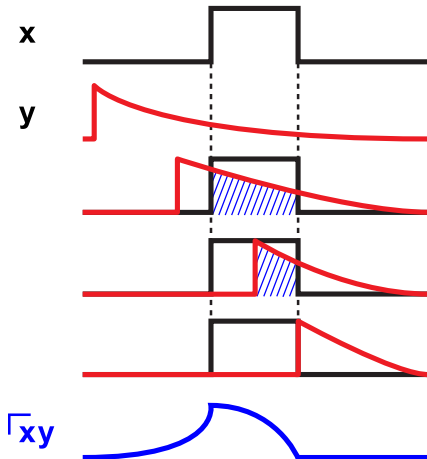
- ▶ La correlación de dos series temporales, también denominada correlación cruzada, es una tercera serie temporal, no como en el caso de la correlación entre dos conjuntos de números que se reporta como un sólo número.
- ▶ La tercera serie temporal representa la *similitud* entre las dos primeras series, cuando una de ellas se desplaza a lo largo de la línea de tiempo.
- ▶ La expresión de la correlación cruzada en el DT:

$$r_{xy}(n) = \sum_{k=0}^{N-1} x(n)y(n+k)$$

- ▶ se convierte en el DF:

$$r_{XY}(f) = X^*(f)Y(f)$$

Correlación de dos series de tiempo en el DT

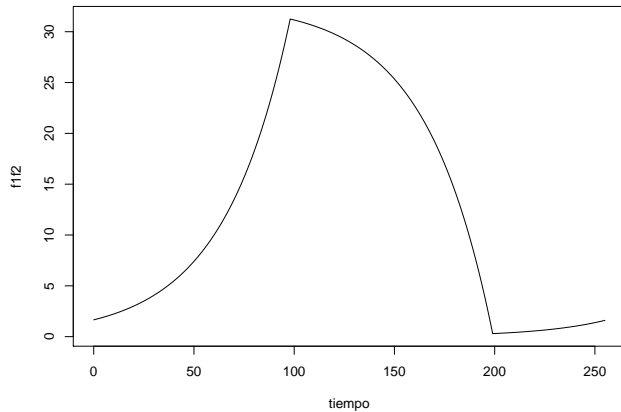


Correlación cruzada con **convolve**

```
# correlacion cruzada con convolve
```

```
f1f2 = convolve(f1,f2,conj = TRUE)  
plot(tiempo,f1f2,type='l')
```

Correlación cruzada con **convolve**



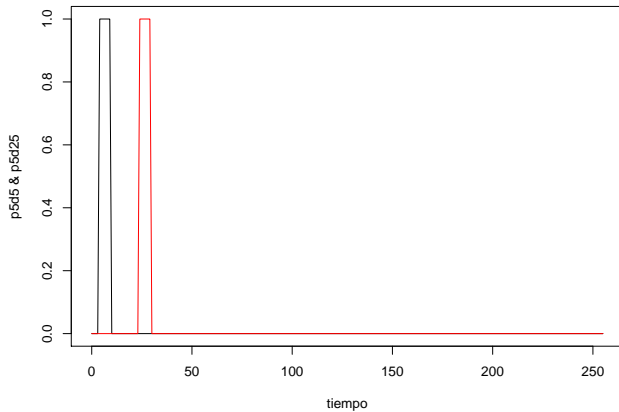
Correlación cruzada de dos pulsos

```
p5d5 = rep(0,N)
p5d5[5:10] = 1
```

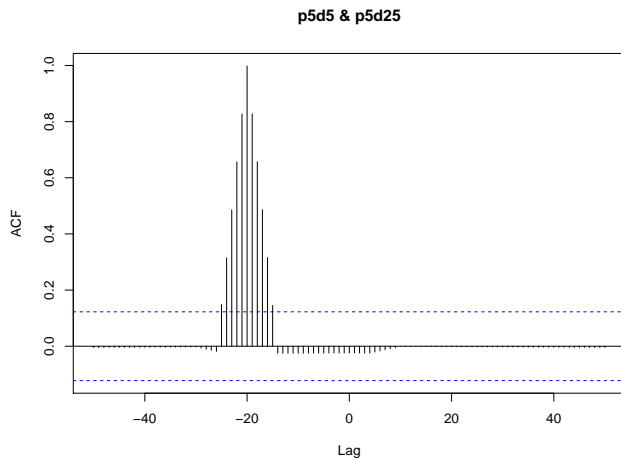
```
p5d25 = rep(0,N)
p5d25[25:30] = 1
plot(tiempo,p5d5,type='l' , ylab='p5d5 & p5d25')
lines (tiempo,p5d25,col='red')
```

```
ccf(p5d5,p5d25,lag.max = 50)
```

Correlación cruzada de dos pulsos



Correlación cruzada de dos pulsos con **ccf**

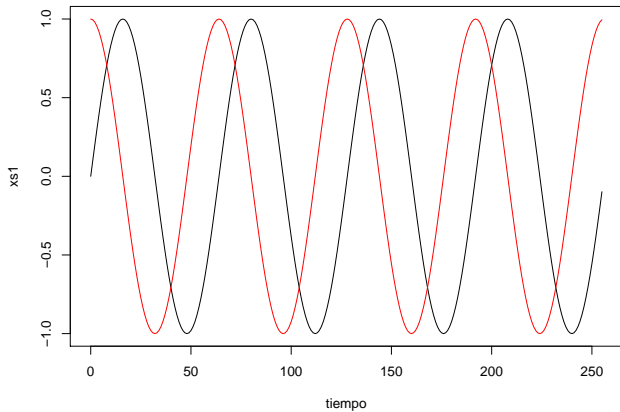


Correlación cruzada de dos senoides

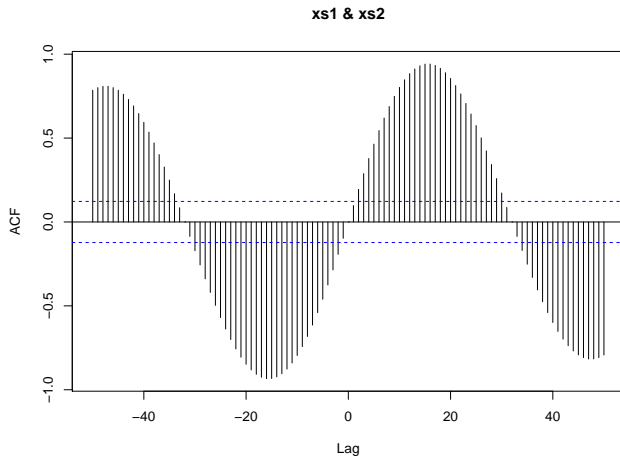
```
xs1 = sin(4*2*pi*tiempo/N)
xs2 = sin(4*2*pi*tiempo/N+pi/2)
# xs2 esta defasada en pi/2 = 16 muestras
plot(tiempo,xs1,type='l')
lines(tiempo,xs2,col='red')

ccf(xs1,xs2,lag.max = 50)
```

Correlación cruzada de dos senoides



Correlación cruzada de dos senoides

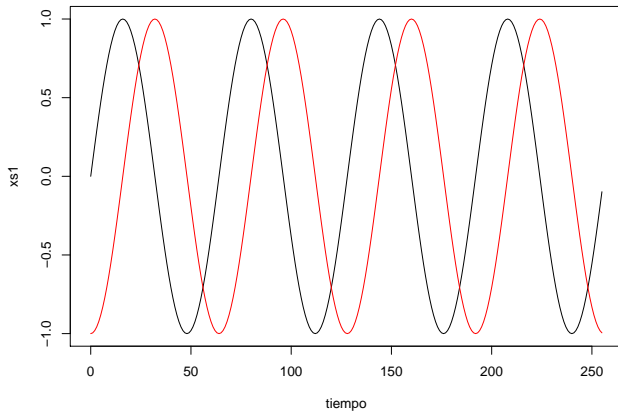


Correlación cruzada de dos senoides

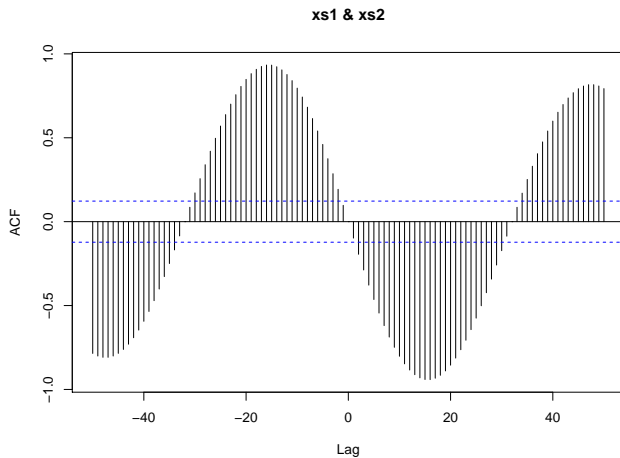
```
xs1 = sin(4*2*pi*tiempo/N)
xs2 = sin(4*2*pi*tiempo/N-pi/2)
# xs2 esta defasada en  $-\pi/2 = -16$  muestras
plot(tiempo,xs1,type='l')
lines(tiempo,xs2,col='red')

ccf(xs1,xs2,lag.max = 50)
```

Correlación cruzada de dos senoides



Correlación cruzada de dos senoides



Autocorrelación de una serie de tiempo

- ▶ La autocorrelación calcula la similaridad de una serie de tiempo consigo misma, pudiendo ser máxima para desplazamiento cero, pero distinta para otros desplazamientos.
- ▶ La expresión de la autocorrelación en el DT:

$$r_{xx}(n) = \sum_{k=0}^{N-1} x(n)x(n+k)$$

- ▶ se convierte en el DF:

$$r_{XX}(f) = X^*(f)X(f)$$

Autocorrelación de ruido uniforme

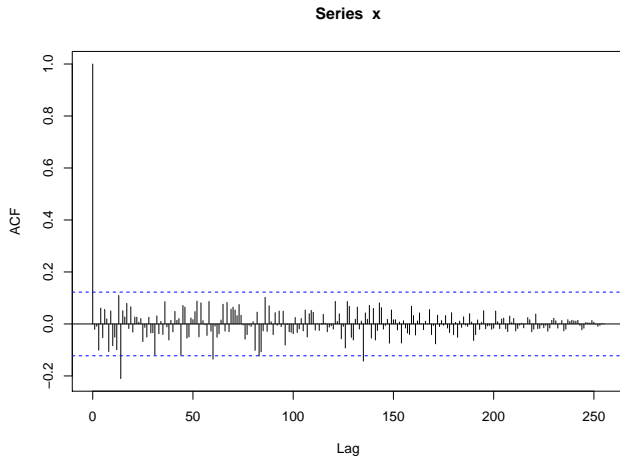
$N = 256$

tiempo = $0:(N-1)$

$x = \text{runif}(N, \text{min} = -2, \text{max} = 2)$

$\text{acf}(x, \text{lag}.\text{max} = N)$

Autocorrelación de ruido uniforme



Autocorrelación de una senoide con/sin ruido

autocorrelacion de una senoide

xs1 = sin(4*2*pi*tiempo/N)

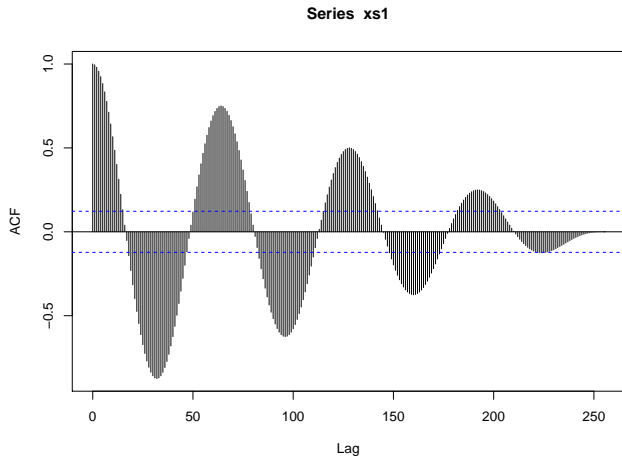
acf(xs1, lag.max = N)

autocorrelacion de una senoide + ruido

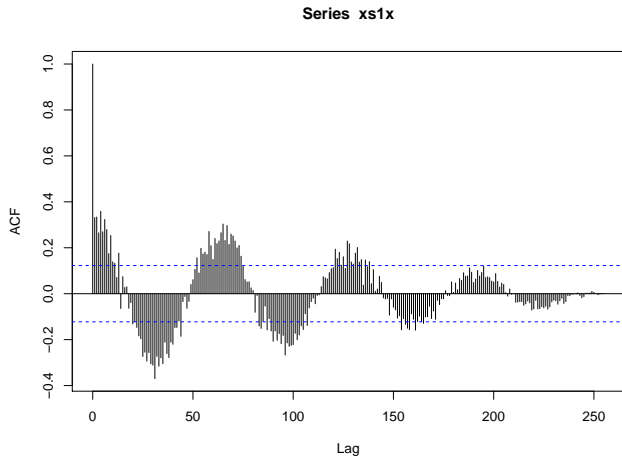
xs1x = xs1 + x

acf(xs1x, lag.max = N)

Autocorrelación de una senoide



Autocorrelación de una senoide con ruido



Ejercicio: analizar las ventas de dos locales

- ▶ Una empresa tiene dos locales de venta, en base a reportes de comerciales se cree que las ventas de un local siguen a las del otro.
- ▶ Los registros de las ventas de cada local para el año anterior muestran estacionalidades anual y mensual, así como variaciones día a día.
- ▶ La inspección visual y gráfica de los datos no sugiere nada definitivo.
- ▶ Analizaremos la relación de las ventas entre los dos locales con la correlación cruzada.

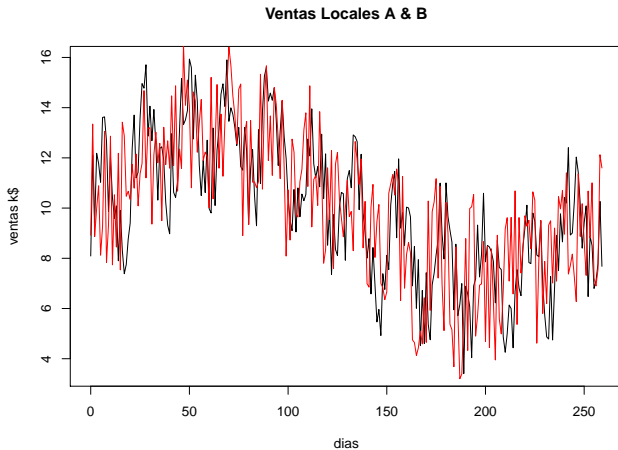
Ejercicio: analizar las ventas de dos locales

```
dos. locales = read.csv('DosLocales.csv')
print(names(dos. locales))
# [1] "X"      "t"      "localA" "localB"

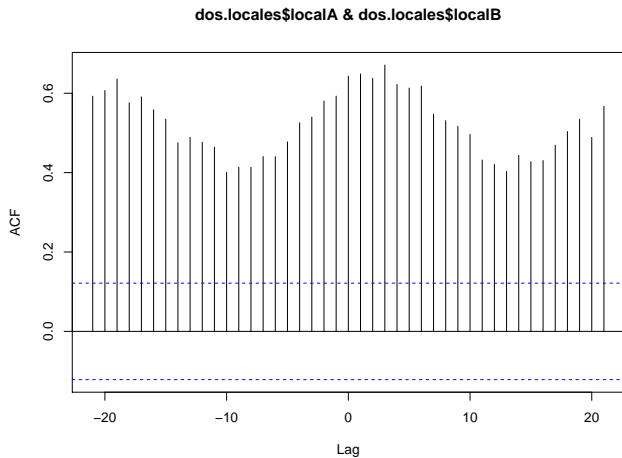
plot(dos. locales$t, dos. locales$localA, type =
      'l', main='Ventas Locales A & B', xlab='dias', ylab='ventas
      k$')
lines(dos. locales$t, dos. locales$localB, col = 'red')

ccf(dos. locales$localA, dos. locales$localB)
```

Ejercicio: ventas de los dos locales



Ejercicio: correlación cruzada de las ventas de los dos locales



Ejercicio: simulación de las ventas de los dos locales

$N = 260$

tiempo = 0:(N-1)

xs1 = 2*sin(12*2*pi*tiempo/N)

xs2 = 1*sin(12*2*pi*tiempo/N+pi/3)

xs3 = 3*sin(2*pi*tiempo/N)

x1 = runif(N,min = -2,max = 2)

x2 = runif(N,min = -3,max = 3)

xsa = xs1 + xs3 + x1 + 10

xsb = xs2 + xs3 + x2 + 10

dir = '/Users/marcelorisk/Dropbox/MateriaDataMining/Clases/'

res = data.frame(t=tiempo,localA=xsa,localB=xsb)

write.csv(res, paste(dir, 'DosLocales.csv', sep=''))

Ejercicio: radio AM, multiplicación DT => convolución DF

ejemplo AM (amplitud modulada)

N = 16*256

tiempo = 0:(N-1)

cm = 0.5 # coef de modulacion

portadora = sin(256*2*pi*tiempo/N)

moduladora = sin(2*2*pi*tiempo/N)

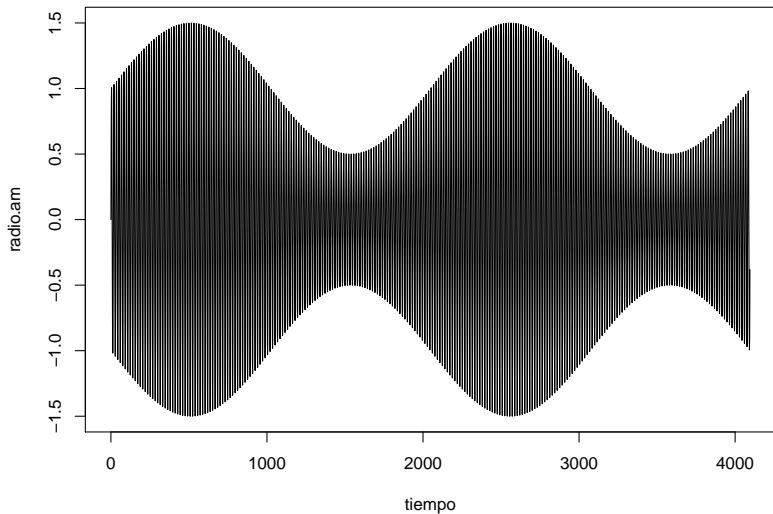
radio.am = (1+cm*moduladora)*portadora

plot(tiempo,radio.am,type='l')

fft.radio.am = Mod(fft(radio.am))

plot(fft.radio.am,type='l',xlim=c(200,300))

Ejercicio: radio AM, multiplicación DT \Rightarrow convolución DF



Ejercicio: radio AM, multiplicación DT \Rightarrow convolución DF

