

GUÍA TEÓRICO PRÁCTICA 8

Método de **medianas repetidas** para la estimación de la pendiente y ordenada al origen de una recta en un diagrama de dispersión (Ver clase teórica).

a1) Genere un conjunto de datos y obtenga su diagrama de dispersión

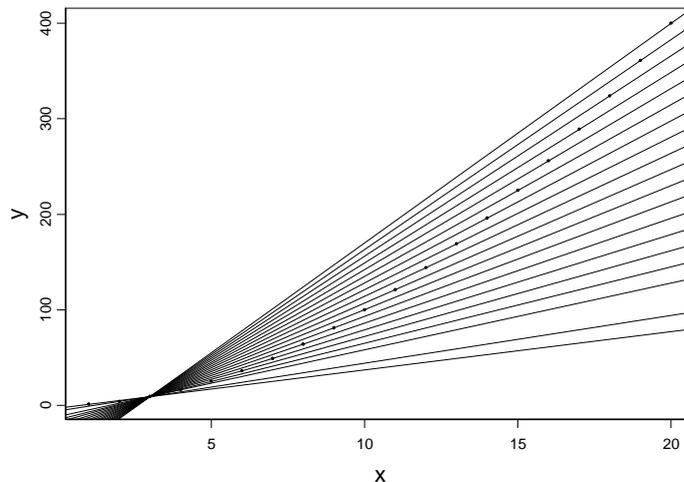
Por ejemplo

```
x<- seq(1,20)
y <- x*x
plot(x,y)
```

a2) Considere la recta que pasa por dos puntos

(por ejemplo, (x[3], y[3]) ; (x[18],y[18]))

Halle la **pendiente** *primero* y *luego* la **ordenada al origen** de la recta que tiene esa pendiente y pasa por uno de los puntos anteriores.



```
pendientes<- vector(length=length(y))
for(i in (1:length(x))) {
  pendientes[i]<-(y[3]-y[i])/(x[3]-x[i])
}
```

```
vectora<- y[3]-pendientes*x[3]
for(i in (1:length(x))[-3]) {
  abline(vectora[i],pendientes[i])
}
```

b) analice detalladamente que obtiene con las siguientes expresiones vectoriales

b1)

```
> (y[1]-y)/(x[1] - x)
> (y[2]-y)/(x[2] - x)
-----
> (y[20]-y)/(x[20] - x)
```

Cada una de las instrucciones anteriores podría obtenerse mediante un ciclo.
¿Cuál?

b2)

```
> median((y[1]-y)/(x[1] - x),na.rm=T)
> median((y[2]-y)/(x[2] - x),na.rm=T)
-----
```

c) Para cada punto $(x[i], y[i])$: $i:1, \dots, 20$ y otro fijo que usted elija halle la pendiente de la recta que los une, halle la ordenada al origen de la recta que tiene esa pendiente.

d) Agregue un outlier y repita a), b) y c)

```
x <-c(x,20)
y <-c(y,1)
```

e) Analice detalladamente que realiza la siguiente función

```
reppedians <- function(x,y) {
  med.pendientes<- vector(length=length(y))
  #mediana de las pendientes para cada punto fijo i
  for(i in 1:length(x)){
    med.pendientes[i]<-median((y[i]-y)/(x[i] -
x),na.rm=T)
  }
  bRM =median(med.pendientes)
  vectora<- y - bRM*x
  aRM <- median(vectora)
  list(ord.origen=aRM,pendiente=bRM)
}
```

f) Si fun es una función de 2 argumentos, la función `outer(v1,v2, FUN)` aplica la función FUN a todas las combinaciones de pares posibles formados con un valor del vector v1 y otro valor del vector v2. Mire el help y genere sus propios ejemplos.

g) Obtenga un vector con las medianas de las pendientes de las rectas que pasan por cada punto

```
med.pendientes<- vector(length=length(y))
  for(i in 1:length(x)){
    med.pendientes[i]<-median((y[i]-y)/(x[i] -
      x),na.rm=T)
  }
med.pendientes
```

y compare con lo obtenido con las siguientes instrucciones

```
mediana <- function(x)median(x,na.rm=T) #para poder calcular con
  datos faltantes
apply( outer(y,-y, "+")/outer(x,-x, "+") ,1,mediana )
```

h) Analice la siguiente función vectorial

```
repmedianas <- function(x,y){
  bRM <- median(apply( outer(y,-y, "+")/outer(x,-x, "+")
    ,1,mediana ))
  vectora<- y - bRM*x
    aRM <- median(vectora)
    list(ord.origen=aRM,pendiente=bRM)
  }
}
```

i) Compare:

```
repmedians(x,y) #con ciclos semivectorial
repmedianas(x,y) #cálculo matricial
```

j) Con los datos originales y con los datos con un outlier obtenga las rectas de medianas repetidas y la de cuadrados mínimos

```
plot(x,y)
abline(unlist(repmedianas(x,y)),col="red")
abline(lm(y~x),col="blue")
```

k) Repita el paso j) para los siguientes conjuntos de datos:

1) x =10 8 13 9 11 14 6 4 12 7 5
y= 8.04 6.95 7.58 8.81 8.33 9.96 7.24 4.26 10.84 4.82 5.68

2) x= 10 8 13 9 11 14 6 4 12 7 5
y= 7.46 6.77 12.74 7.11 7.81 8.84 6.08 5.39 8.15 6.42 5.73

¿Cuál de las dos rectas le parece que ajusta mejor en cada caso?