# Urna de Ehrenfest

*J. Abellán*

*7/11/2019*

## Urna de Ehrenfest

### Vamos a simular la famosa urna de Ehrenfest

Supongamos una urna inicialmente con un número $B$ de bolas blancas y $N - B$ bolas negras.

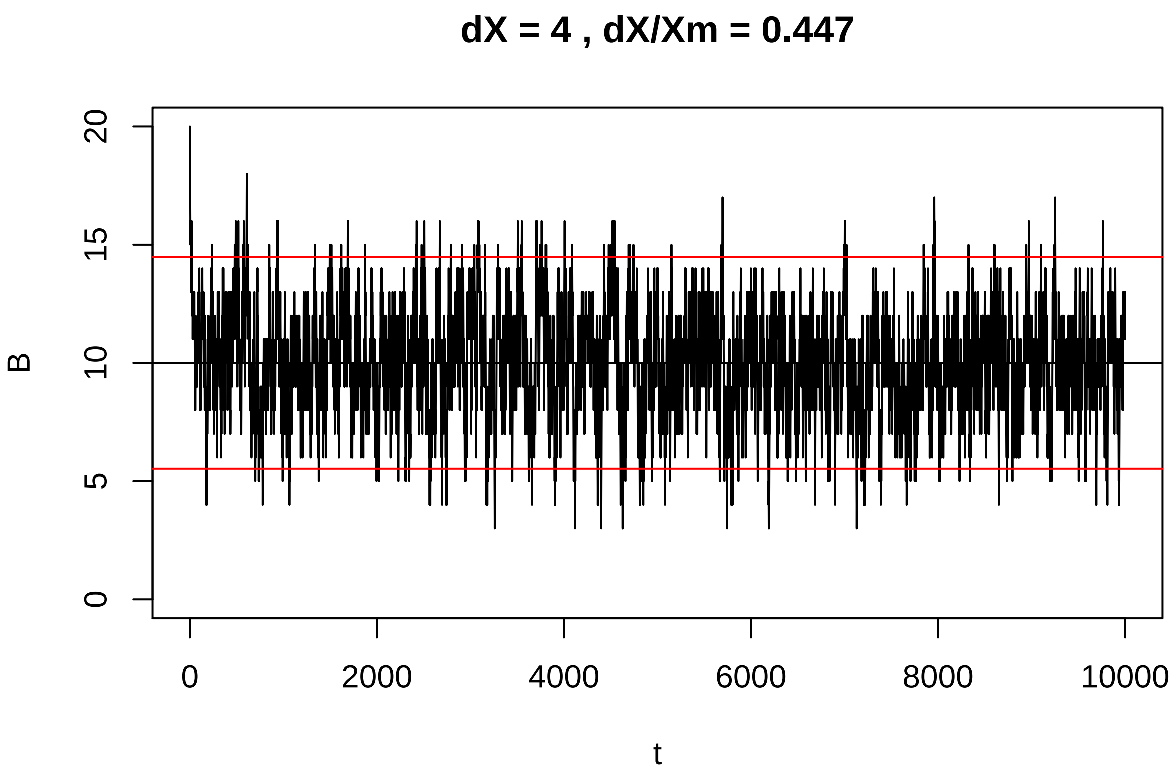
- Introducimos la mano y extraemos una bola al azar.
- Si la bola extraida es blanca introducimos una negra y si es negra introducimos una blanca. Dicho de otro modo, **castigamos** el color extraido, porque no devolvemos a la urna la bola extraida, y **favorecemos** el opuesto. El número total de bolas permanece constante.
- Repetimos las extracciones un número grande de veces.

*¿Cuál será el contenido de la urna al final del juego?*

El sentido común nos dice que al final habrá aproximadamente $N/2$ bolas blancas y $N/2$ bolas negras.

```r
#Número de bolas
N <- 20

#Teoría: media, desviación estándar y desviación relativa
Xm <- N / 2

dX <- sqrt( N )

frX <- dX / Xm

#Extracciones o tiempo
tmax <- 10000

t <- 1:tmax

#Número inicial de bolas blancas en la urna: todas
B <- rep( 0, tmax )

B[ 1 ] <- N

for ( i in 1:( tmax - 1 ) ) {

  #Recuerda: p(B)=B/N. Castigamos la que sale
  pB <- B[ i ] / N

  dB <- sample( c( -1, 1 ), 1, prob = c( pB, 1 - pB ) )

  B[ i + 1 ] <- B[ i ] + dB

}
```

```r
plot( t, B,

      type = "l",

      ylim = c( 0, N ),

      main = paste( "dX =",round( dX ),", dX/Xm =", round( frX, 3 ) )

    )

abline( h = Xm )

abline( h = c( Xm - dX, Xm + dX ), col = 2 )
```

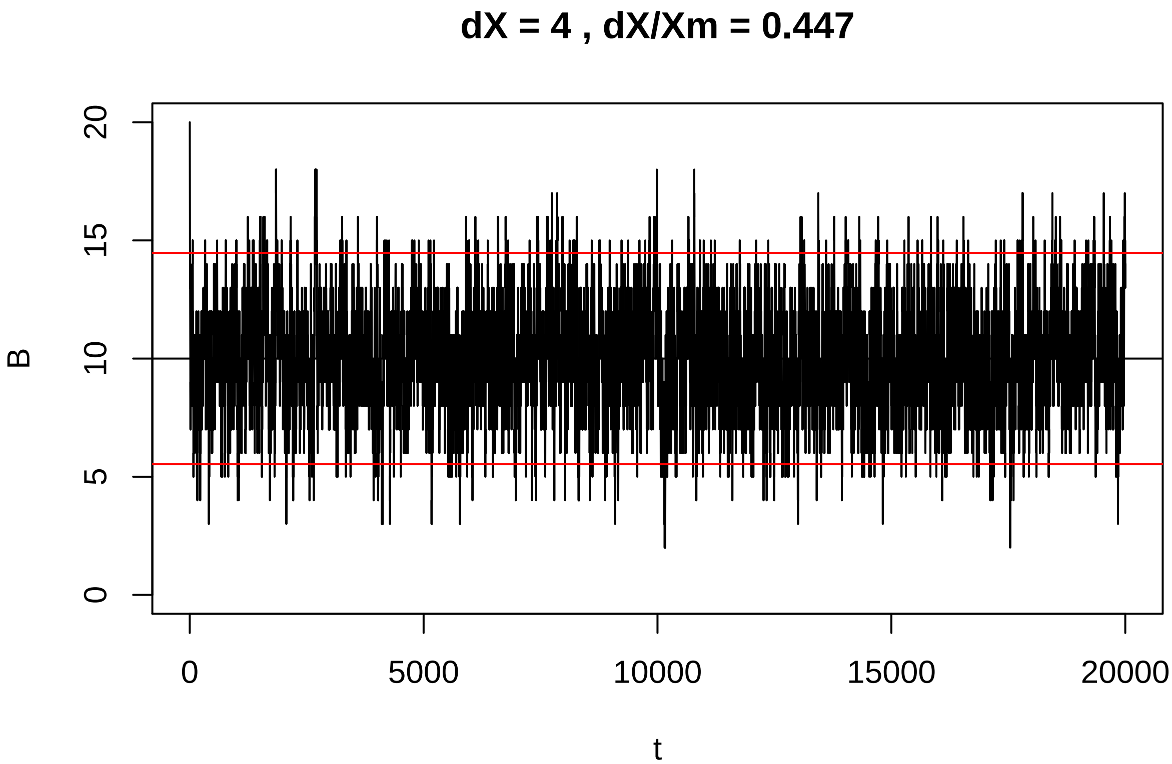


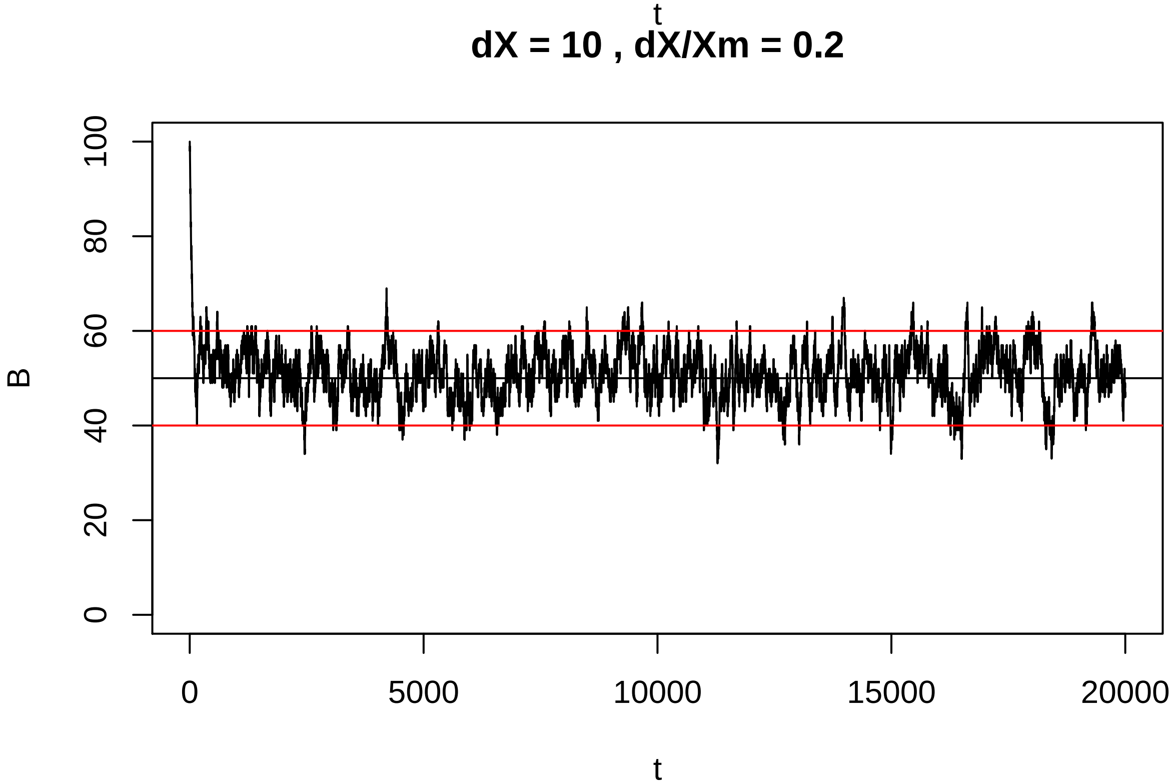

### Tamaño de las fluctuaciones

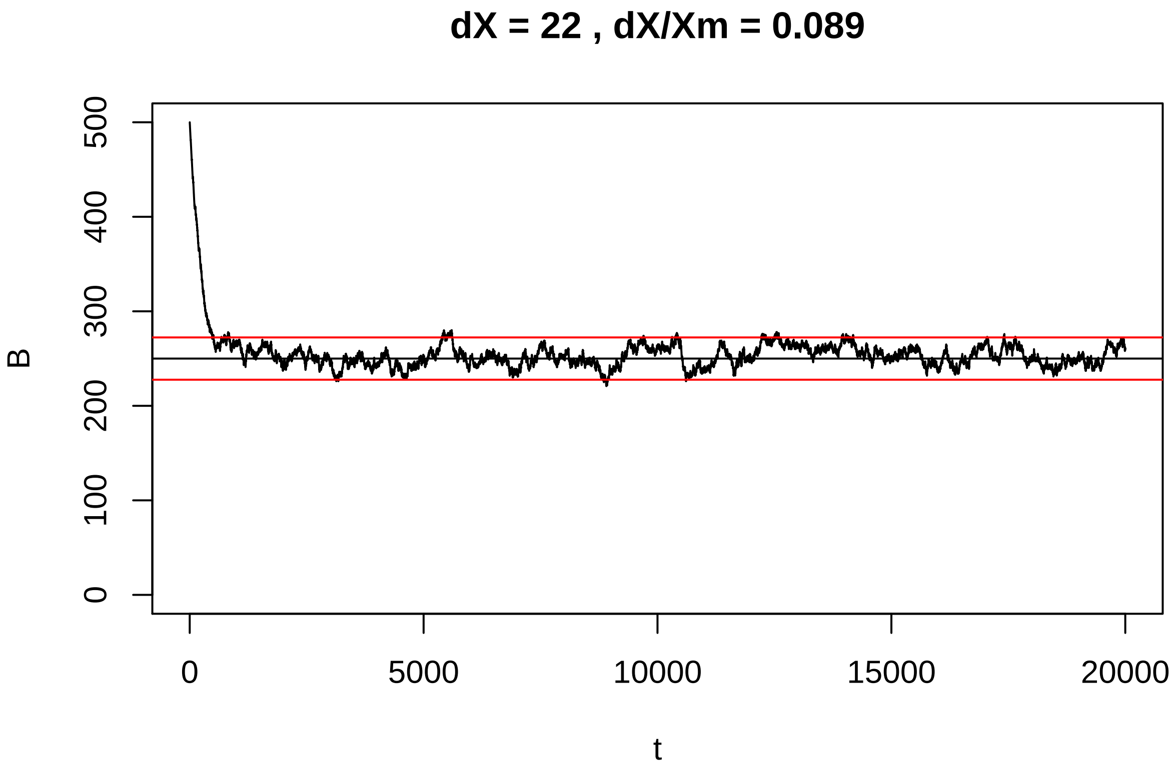
Si aumentamos el número total de bolas en la urna, las fluctuaciones relativas serán cada vez más pequeñas. Repetimos la simulación variando **N**

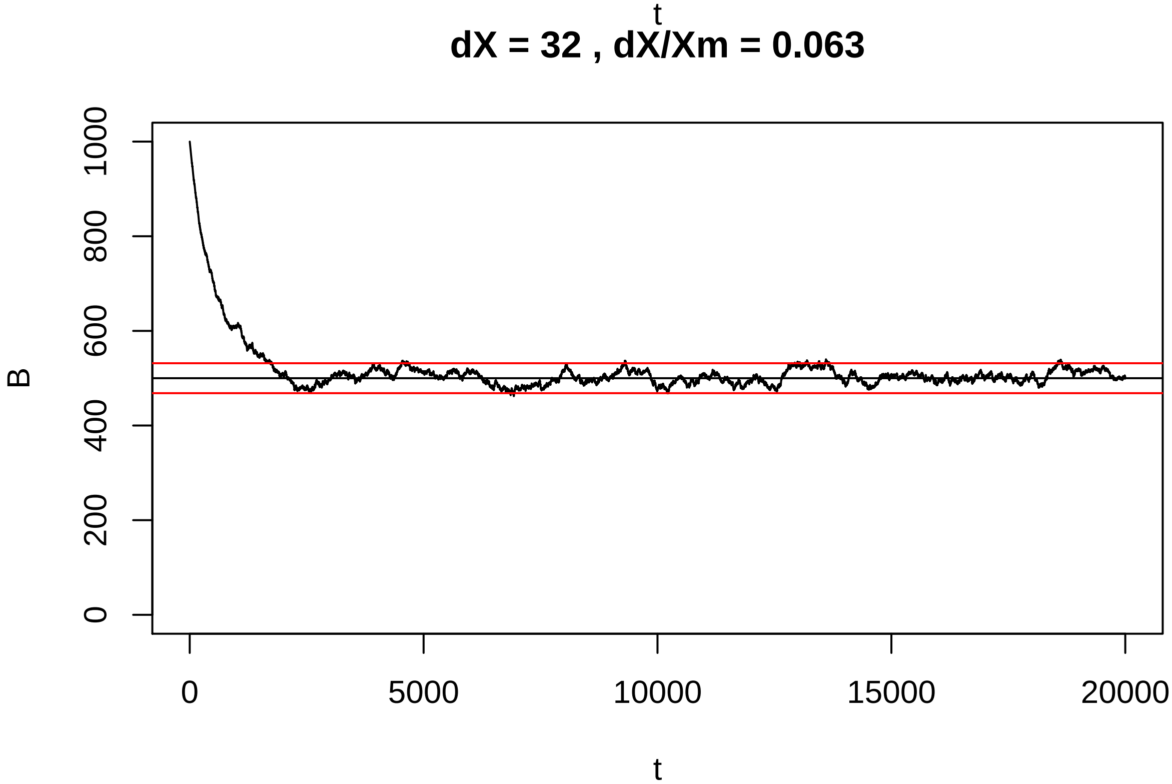
```r
#Diferentes número de bolas
N <- c( 20, 100, 500, 1000, 10000 )

nb <- length( N )

#Guardo memoria de la fluctuación relativa
frX <- rep( 0, nb )

#Extracciones o tiempo
tmax <- 20000
```

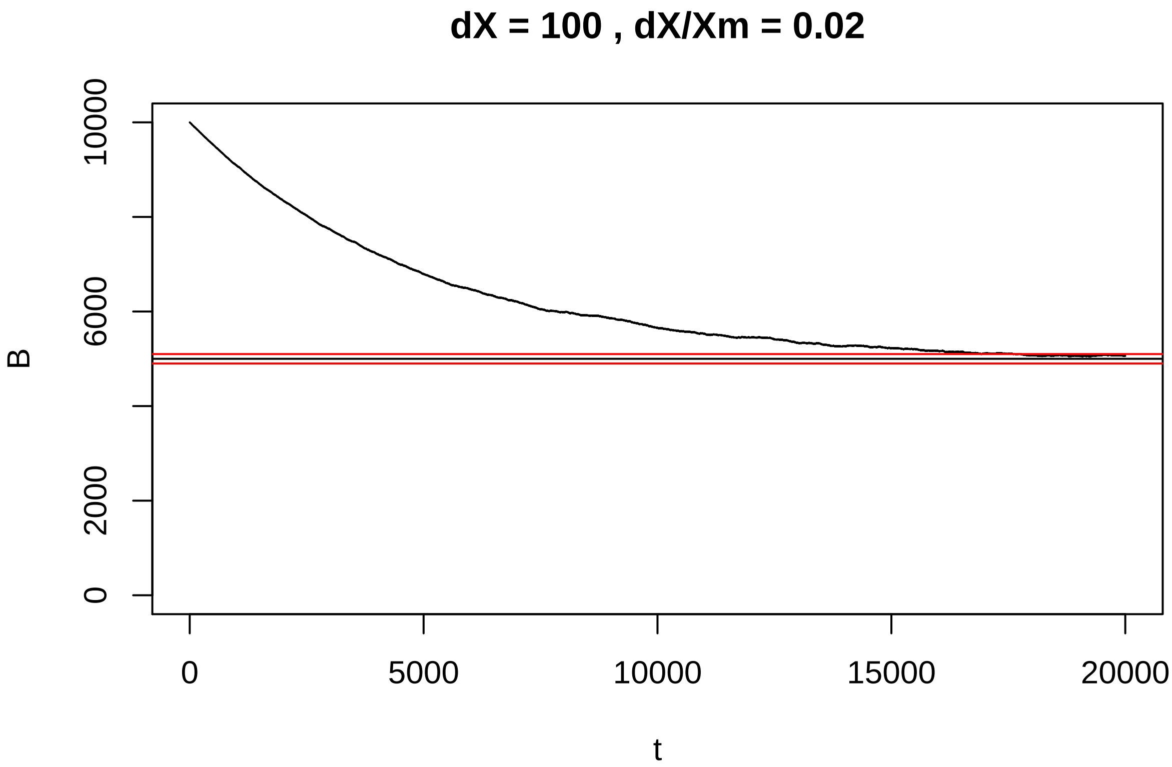
```r
t <- 1:tmax

for( j in 1:nb ) {

  #Número inicial de bolas blancas en la urna: todas
  B <- rep( 0, tmax )

  B[ 1 ] <- N[ j ]

  for ( i in 1:( tmax - 1 ) ) {

     #RecuerdE: p(B) = B/N.
     pB <- B[ i ] / N[ j ]

     dB <- sample( c( -1, 1 ), 1, prob = c( pB, 1 - pB ) )

     B[ i + 1 ] <- B[ i ] + dB

  }

  #Teoría: media, desviación estándar y desviación relativa
  Xm <- N[ j ] / 2

  dX <- sqrt( N[ j ] )

  frX[ j ] <- dX / Xm

  plot( t, B,

        type = "l",

        ylim = c( 0, N[ j ] ),

        main = paste("dX =", round( dX ),

                     ", dX/Xm =", round( frX[ j ], 3 ) )

    )

  abline( h = Xm )

  abline( h = c( Xm - dX, Xm + dX ), col = 2 )

}
```


dX = 4 , dX/Xm = 0.447
B
t
0
5
10
15
20
0
5000
10000
15000
20000


dX = 10 , dX/Xm = 0.2
B
t
0
20
40
60
80
100
0
5000
10000
15000
20000


dX = 22 , dX/Xm = 0.089
B
0
100
200
300
400
500
0
5000
10000
15000
20000
t

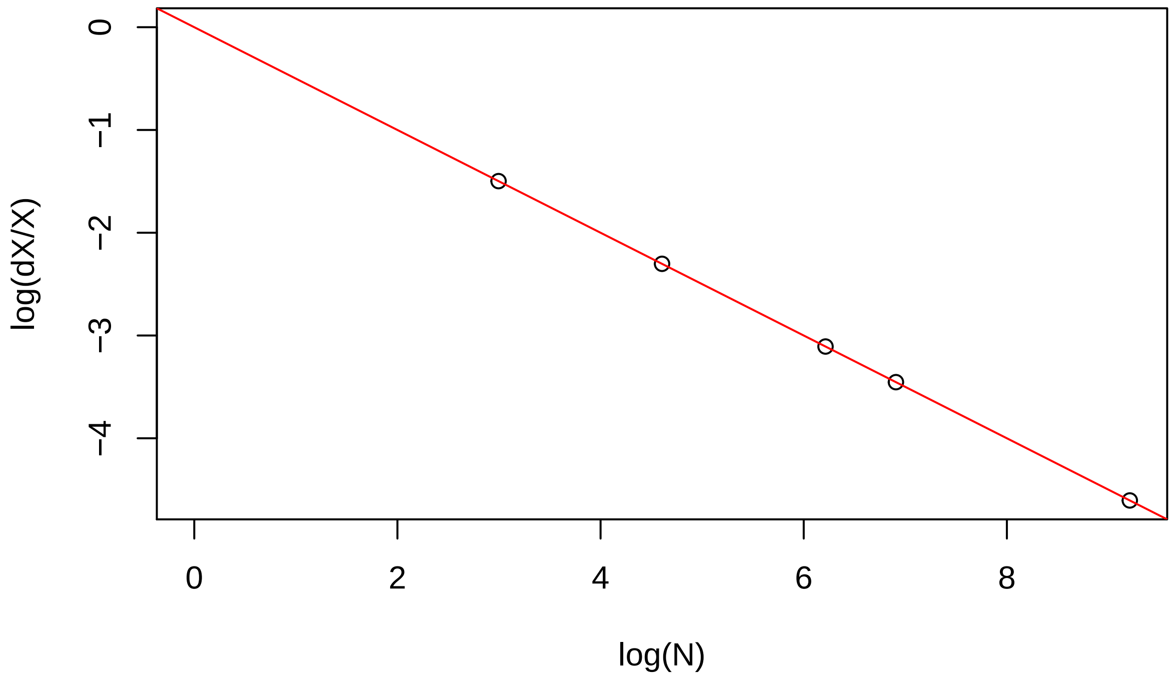

dX = 32 , dX/Xm = 0.063
B
0
200
400
600
800
1000
0
5000
10000
15000
20000
t




### Comprobación teórica

Para comprobar la dependencia de las fluctuaciones relativas con el número de bolas, dibujamos el logaritmo de la fluctuación frente al logaritmo del número de bolas en la siguiente gráfica. De acuerdo con la teoría la pendiente debe ser $-1/2$:

```
x <- log( N )

y <- log( frX ) - log( 2 )

xm <- log( B[ nb ] )

ym <- -xm / 2

plot( x, y,

      xlab = "log(N)",

      ylab = "log(dX/X)",

      xlim = c( 0, xm ),

      ylim = c( ym, 0 )

    )

ajuste = lm( y ~ x )

abline( ajuste, col = 2 )
```




```
ajuste$coefficients
```

```
##   (Intercept)             x
##  1.986027e-15 -5.000000e-01
```

**Es ésta una manera sencilla de entender la estabilidad del equilibrio termodinámico**.

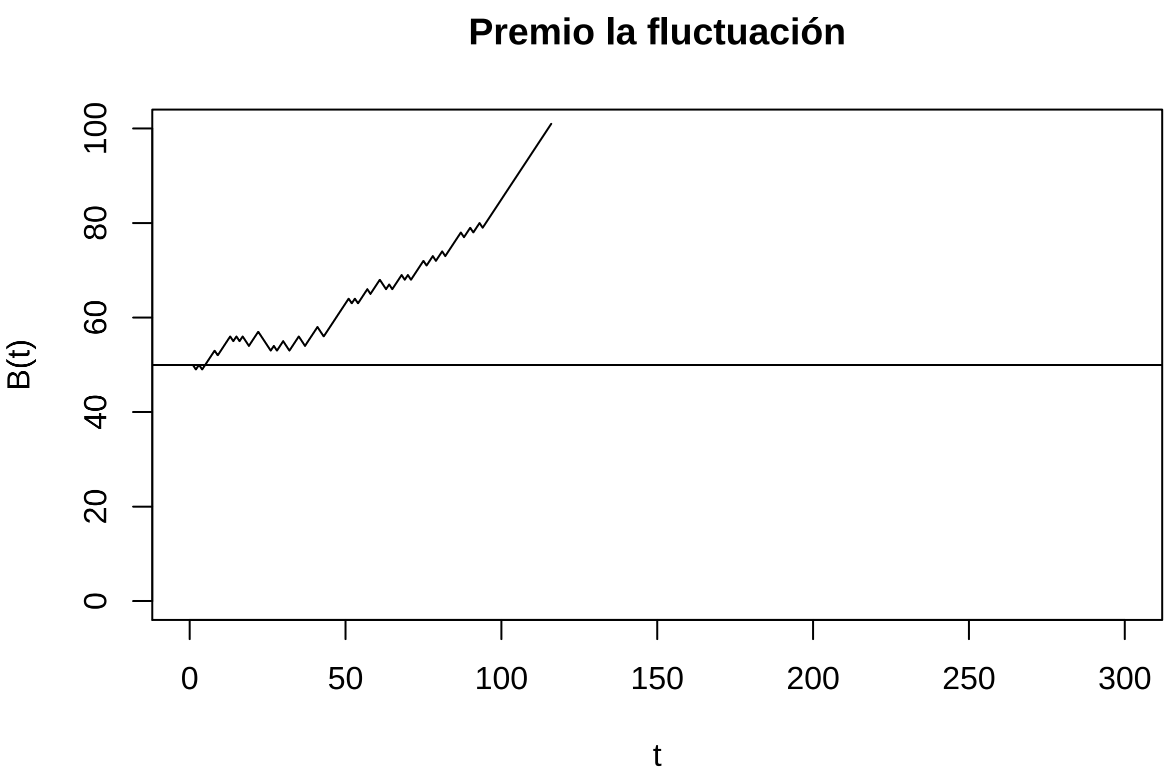
## Fluctuaciones 'premiadas'

¿Qué ocurre si cambiamos las reglas del juego y "premiamos" el color de la extracción?

Repetimos la simulación cambiando simplemente el *signo* del cambio en el número de bolas $dB$

```
#Número de bolas
N <- 100

#Extracciones o tiempo
tmax <- 300

t <- 1:tmax

#Número inicial de bolas blancas en la urna: la mitad
B <- rep( 0, tmax )

i <- 1

B[ i ] <- N / 2

for ( i in 1:(tmax - 1 ) ) {

  #Recuerda: pB = B/N. Si pB<0 o pB>1 terminamos!
  pB <- B[ i ] / N; if ( pB < 0 || pB > 1 ) break()
```

```
  B[ i + 1 ] <- B[ i ] + sample( c( 1, -1 ), 1, prob = c( pB, 1 - pB ) )

}

# Valor de i con el que terminamos
ifinal <- i

plot( t[ 1:ifinal ], B[ 1:ifinal ],

      type = "l",

      ylim = c( 0, N ),

      xlim = c( 0, tmax),

      xlab = "t",

      ylab = "B(t)",

      main = "Premio la fluctuación"

    )

abline( h = N / 2 )
```



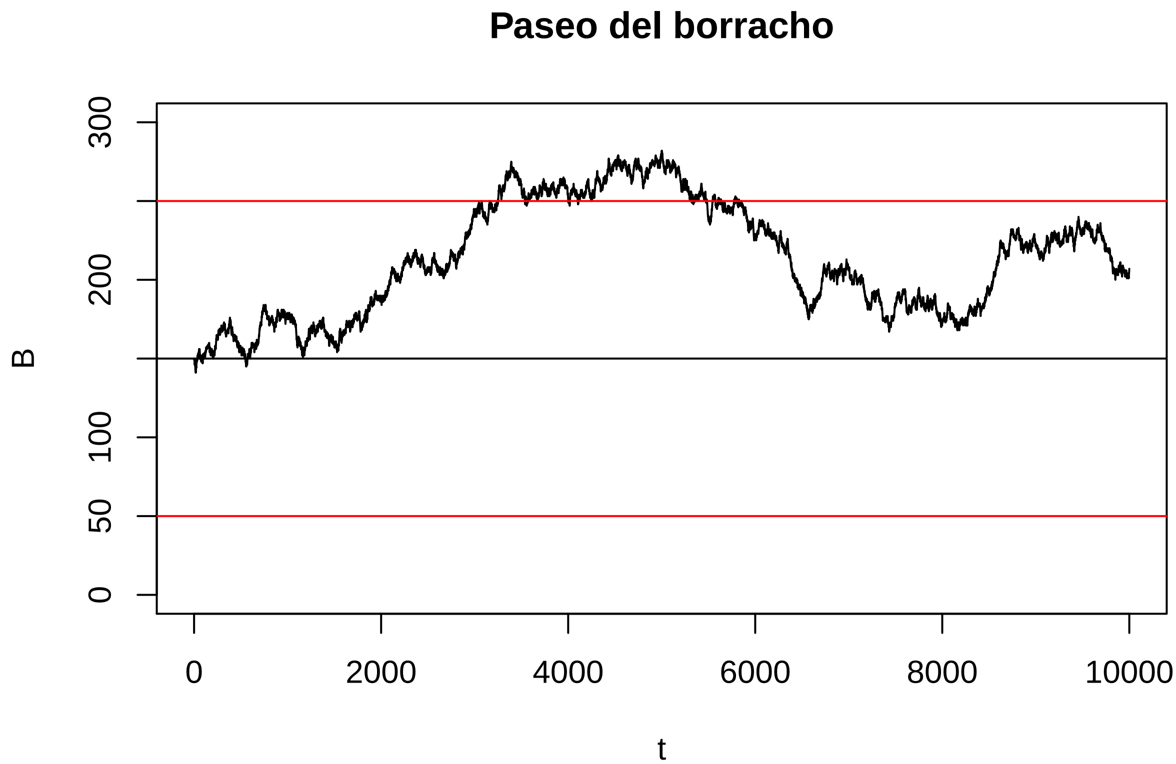

El resultado del juego es claro: gana uno de los colores pero **¡no podemos saber cual!**

## Paseo del borracho

Finalmente ¿qué ocurrirá si el premio no depende del número de bolas y es al azar, con igual probabilidad el que decide?

La respuesta es el paseo del borracho: si se dan $N$ pasos, se avanza $\sqrt{N}$

```r
#Número de bolas
N <- 300

#Extracciones o tiempo
tmax <- 10000

t <- 1:tmax

#Número inicial de bolas blancas en la urna: la mitad
B <- rep( 0, tmax )

B[ 1 ] <- N / 2

for ( i in 2:tmax ) {

  B[ i ] <- B[ i - 1 ] + sample( c( -1, 1 ), 1 )

  # terminamos el juego si ...
  if ( B[ i ] == 0 || B[ i ] == 1 ) break()

}

plot( t, B,

      type = "l",

      ylim = c( 0, N ),

      main = "Paseo del borracho"

    )

abline( h = N / 2 )

abline( h = c( N / 2 + sqrt( tmax ), N / 2 - sqrt( tmax ) ), col = 2 )
```



Paseo del borracho
B
t
0
50
100
200
300
0
2000
4000
6000
8000
10000