Funciones de variables aleatorias

J. Abellán

17 de noviembre de 2016

Funciones de variables aleatorias

Caso sencillo $Y = Y(X)$

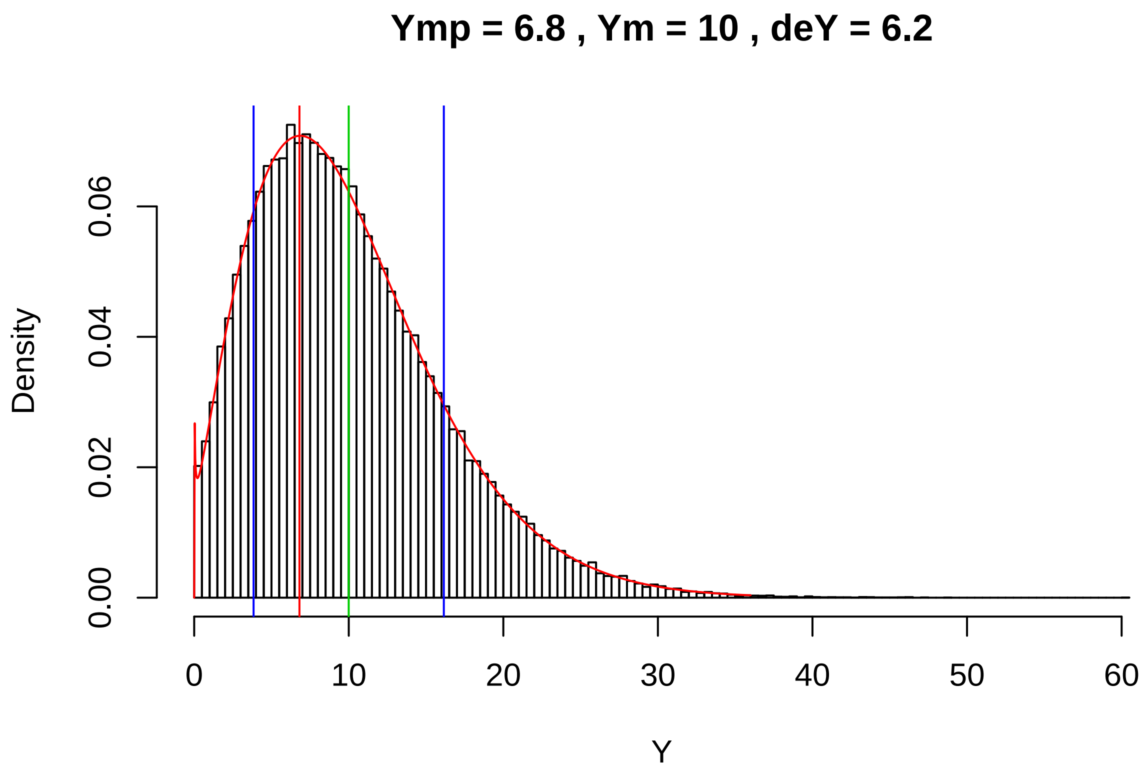
- Sea la variable aleatoria $X \sim N(\mu_X, \sigma_X)$
- Sea $Y = Y(X) = X^2$
- **¿Como será la función de distribución de $Y, f_Y(y)$?**

De acuerdo con el teorema:

$$f_Y(y) = f_X(x(y))|\frac{dx}{dy}(y)|$$

Calculamos primero la función de distribución de X (aunque no hace falta porque **R** ya la tiene definida). Y después aplicamos el teorema:

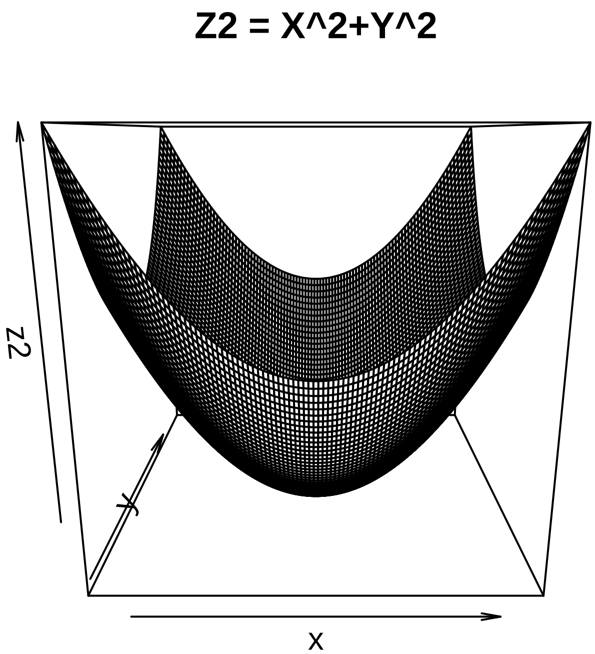
Consideremos un caso concreto $\mu_X = 3, \sigma_X = 1$

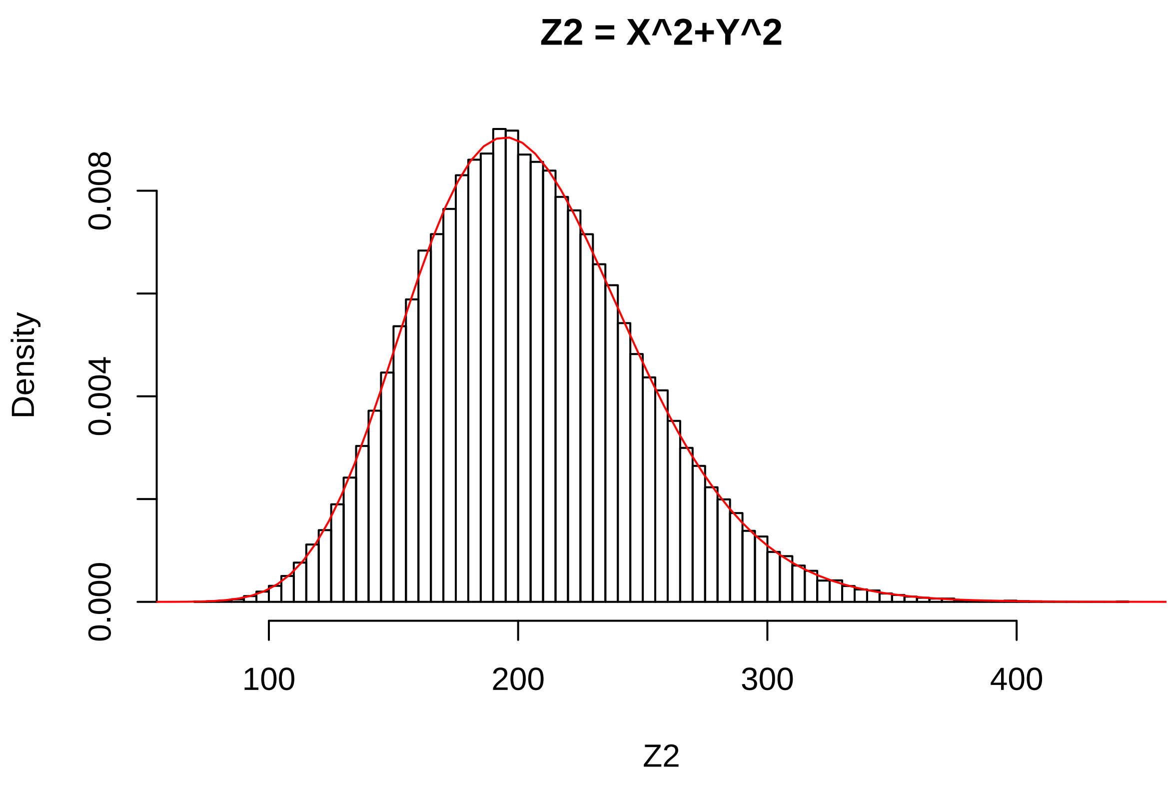


Puede observarse que la media de Y no coincide siempre con el valor más probable.

Ejemplo más complejo $Z = Z(X, Y) = X^2 + Y^2$

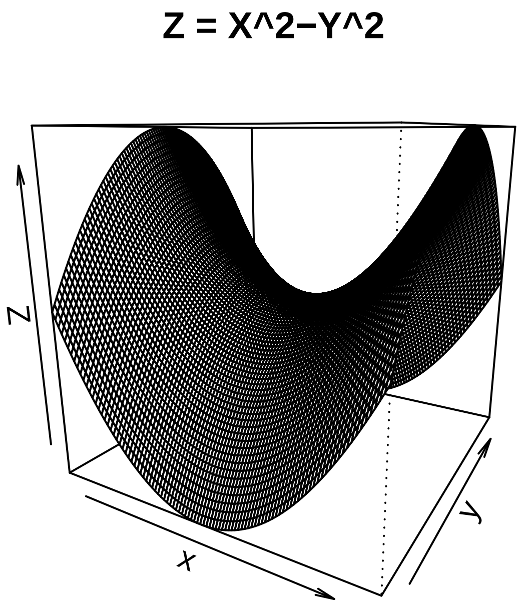
Supongamos ahora el caso de una variable aleatoria Z que es función de dos variables aleatorias X, Y. Primero vemos en tres dimensiones la forma de $Z(X, Y)$:



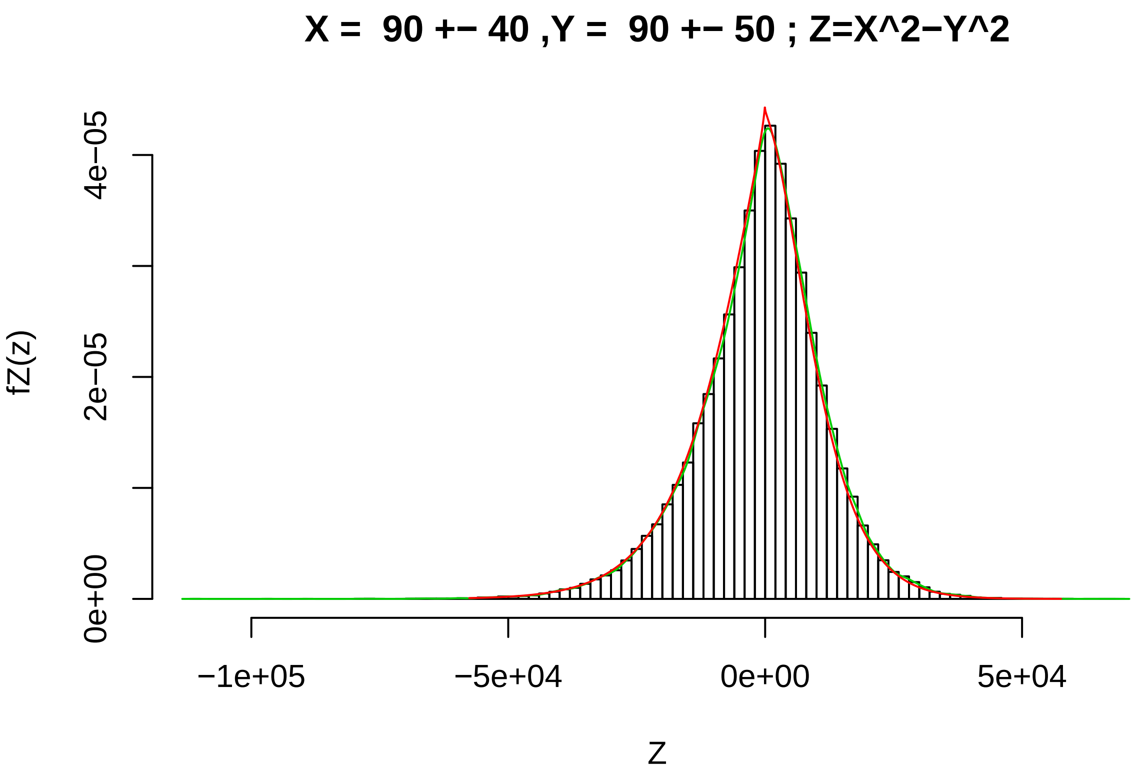


Otro ejemplo $Z = X^2 - Y^2$

El penúltimo ejemplo

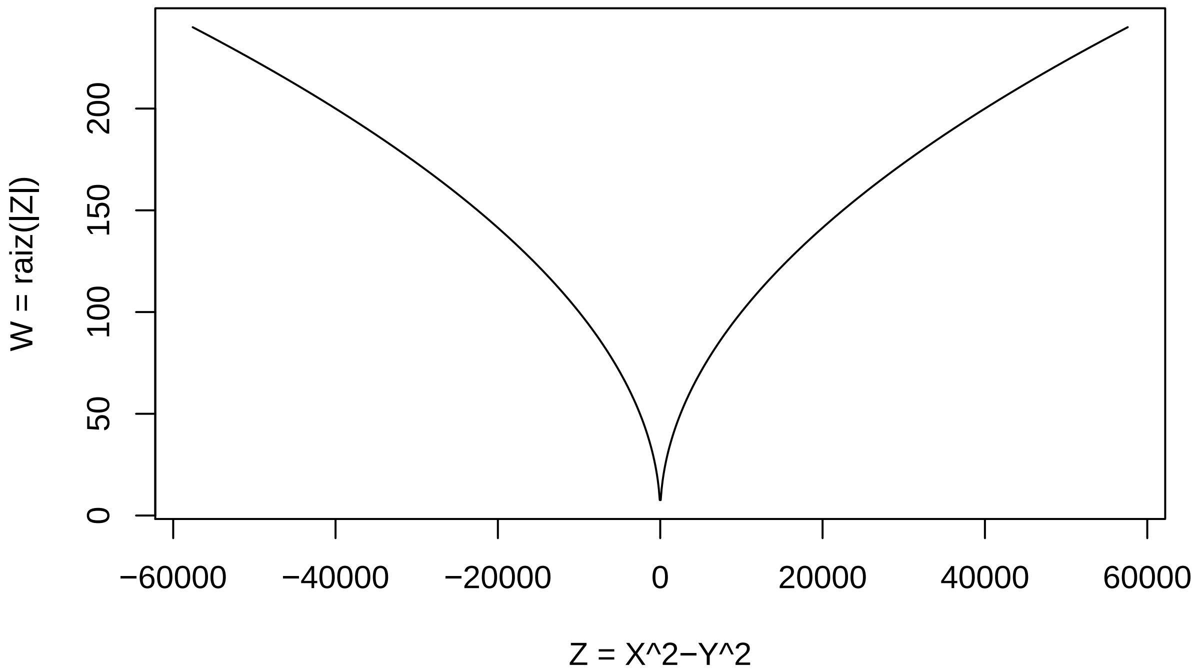


El último



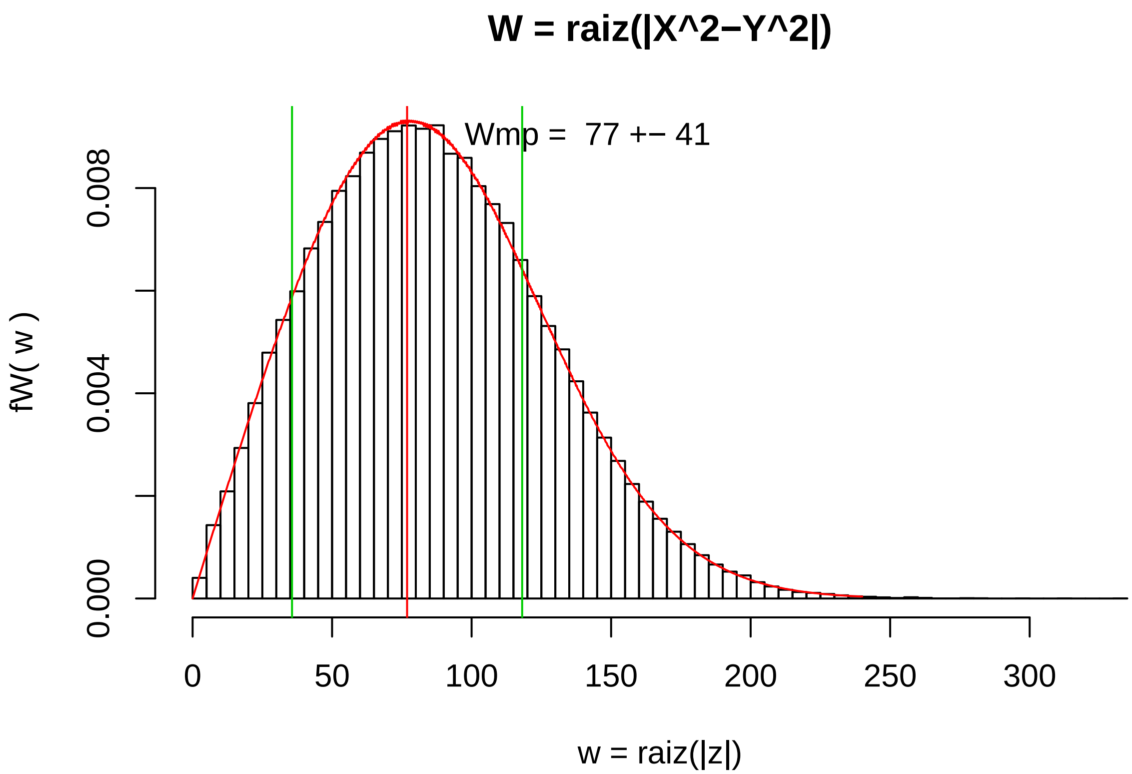
$W = \sqrt{|Z|}$

Por último, lo que nos interesa. Dibujamos primero la dependencia funcional entre WyZ:



Después dibujamos el histograma de W y comprobamos que coincide con lo previsto por la teoría:

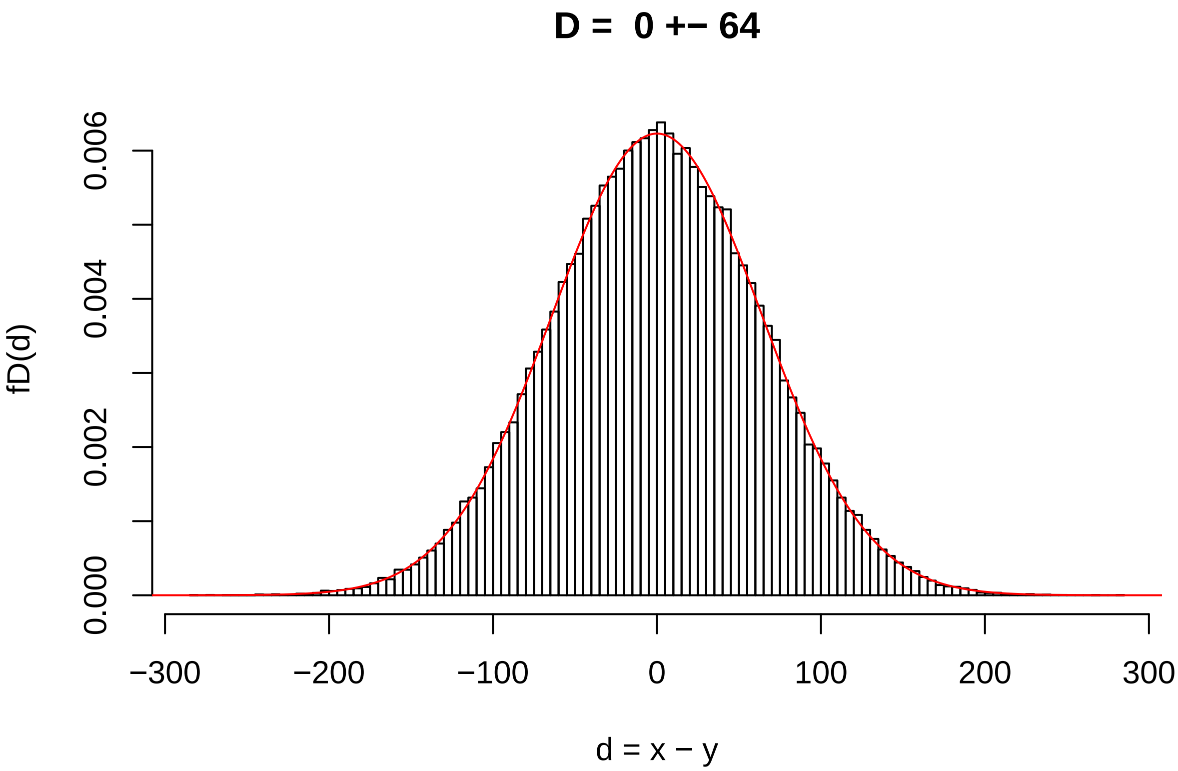
```
## [1] 0.9999987
```



Sencillo $D = X - Y$

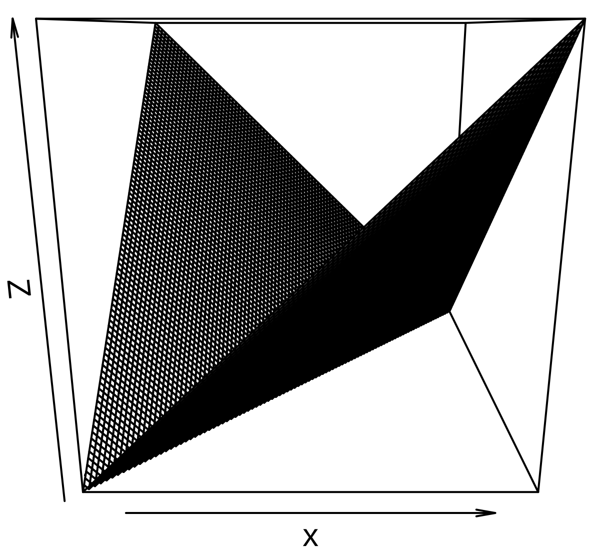
Veamos el ejemplo más sencillo. De acuerdo con la teoría $D \sim N(\mu_X - \mu_Y, \sqrt{\sigma_X^2 + \sigma_Y^2})$

```r
#Parámetros de X e Y
muX <- 90 ; deX <- 40

muY <- 90 ; deY <- 50

#Monte Carlo
N <- 1e5

X <- rnorm( N, muX, deX )

Y <- rnorm( N, muY, deY )

# la nueva función
D <- X - Y

#Comprobación teórica
xmax <- muX + 2 * deX ; ymax <- muY + 2 * deY

zmax <- 2 * max( xmax, ymax )

dxy <- seq( - zmax, zmax, length.out = 1000 )

muD <- muX - muY ; deD <- sqrt( deX^2 + deY^2 )

hist( D, 100, prob =T,

      xlab = "d = x - y",

      ylab = "fD(d)",

      main = paste( "D = ", round( muD ), "+-", round( deD ) )

    )

lines( dxy, dnorm( dxy, muD, deD ), col = 2 )
```



$Z = |X - Y|$

```
# función valor absoluto
adXY <- function( x, y ) abs( x - y )

# producto exterior
Z <- outer( x, y, FUN = adXY )

persp( x, y, Z )
```



```r
#Parámetros de X e Y
muX <- 90 ; deX <- 40

muY <- 90 ; deY <- 50

#Monte Carlo
N <- 1e6

X <- rnorm( N, muX, deX )

Y <- rnorm( N, muY, deY )

Z <- abs( X - Y )

Zm <- mean( Z ) ; deZ <- sd( Z )

#Teoría
xmax <- muX + 2 * deX ; ymax = muY + 2 * deY

# valores de las variables x e y
nx <- ny <- 100

x <- seq( - xmax, xmax, length.out = nx ) ; dx <- 2 * xmax / ( nx - 1 )

y <- seq( - ymax, ymax, length.out = ny )

nz <- 1000 ; z <- seq( 0, zmax, length.out = nz ) ; dz <- zmax / ( nz - 1 )

fZ <- rep( 0, n )

for ( i in 1 : n ) {

    #Hay dos ramas
    fZ1 <- sum( dx * dnorm( x, muX, deX ) * dnorm( x - z[ i ], muY, deY ) )

    fZ2 <- sum( dx * dnorm( x, muX, deX ) * dnorm( x + z[ i ], muY, deY ) )

    fZ[ i ] <- fZ1 + fZ2

}

#fZ=fZ/sum(fZ*dz)

hist( Z, 100,

      xlab = "|X-Y|",

      prob = T,

      main = paste( "Zm = ", round( Zm ), "+-", round( deZ ) )

    )
```

```r
lines( z, fZ, col = 2 )
```