VA-FuncionNoLineal

J. Abellán

20 de septiembre de 2016

Función de variable aleatoria

El caso $Y = Y(X) = X^2$

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

De acuerdo con el teorema:

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

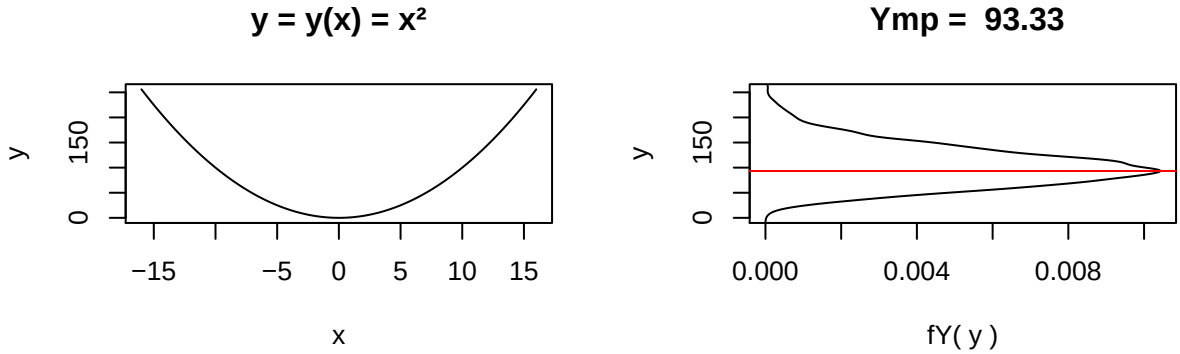
Comprobaremos el teorema de la forma habitual: generando al azar un número grande de valores de la variable normal X y transformándolos de acuerdo con la función $Y = Y(X)$. A continuación haremos el histograma de los valores de Y.

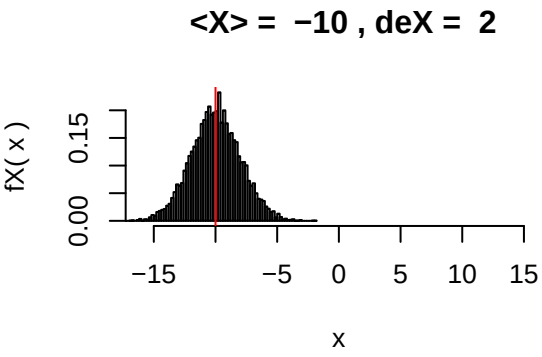
Queremos ver como se modifica $f_Y(y)$ al variar el parámetro μ_X

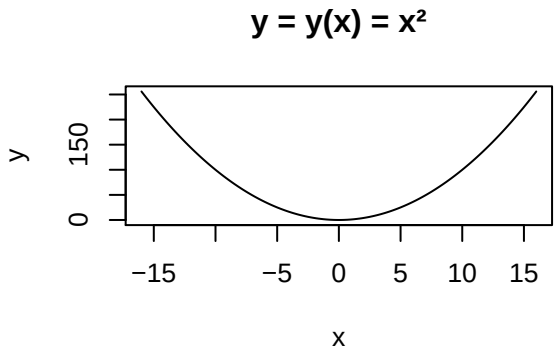
```
#library("latex2exp", lib.loc="~/R/i686-pc-linux-gnu-library/3.2")

#la nueva función
ny <- function( x ) x^2

#Parámetros de la variable normal X
deX <- 2

MUX <- c(-10, -5, -2 , 0, 2, 5, 10 )

n <- length( MUX )

#Valores que toma la variable X
#xmin=muX-4*deX;xmax=muX+4*deX
xmin <- - 16 ; xmax <- 16

x <- seq( xmin, xmax, len = 1000 )

#Por tanto, los valores de y serán
y <- ny( x )

#Límites de dibujo
ymin <- 0 ; ymax <- max( y )

for (i in 1 : n ) {

    muX <- MUX[ i ]

    #Para cuatro gráficas
    matriz <- matrix( 1 : 4, 2, 2 )
```

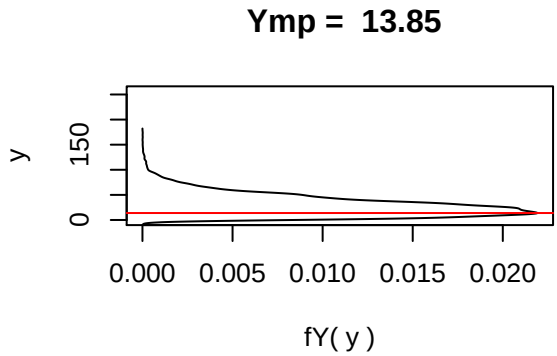
```
layout( matriz )

#Primera gráfica fila=1, columna=1
plot( x, y,

      type = "l",

      # main = latex2exp("$ y = y(x) = \\x^2 $")
      main = "y = y(x) = x²"

    )

#Generamos los valores X y los transformamos
N <- 10000

X <- rnorm( N, muX, deX )

Y <- ny( X )

#Segunda gráfica: fila=2, columna=1
hist( X, 100,

      xlab = " x ",

      ylab = " fX( x ) ",

      xlim = c( xmin, xmax ),

      prob = T,

      main = paste( " <X> = ", muX, ", deX = ", deX )

    )

abline( v = muX, col = 2 )

#Tercera gráfica: fila=1, columna=2
#Calculamos el histograma pero no lo dibujamos
hY <- hist( Y, 100, plot = FALSE )

#Extraemos la información del objeto hY
fYe <- density( Y )$y  #hY$density

ye <- density( Y )$x #hY$mids

#Valor más probable
ymp <- ye[ which.max( fYe ) ]

#Dibujamos pero girando la gráfica
plot( fYe, ye,

      xlab =" fY( y ) ",
```

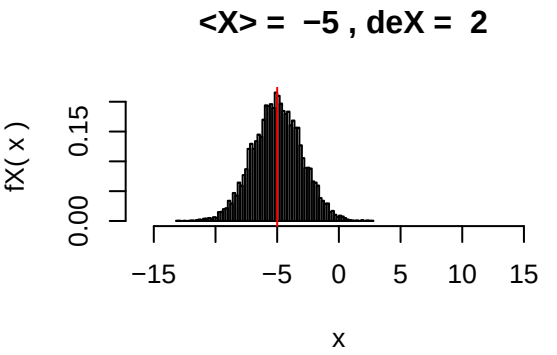
```
          ylab = " y ",
          ylim = c( ymin, ymax ),
          type = "l",
          main = paste( " Ymp = ", round( ymp, 2 ) )
        )
    abline( h = ymp, col = 2 )
}
```

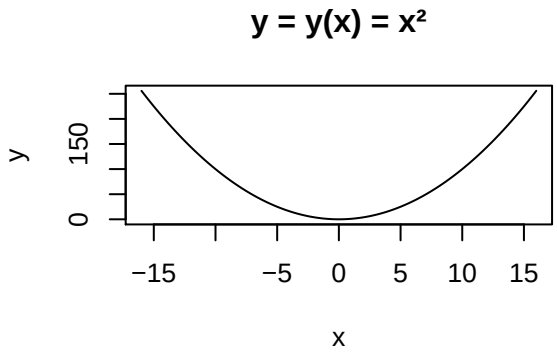


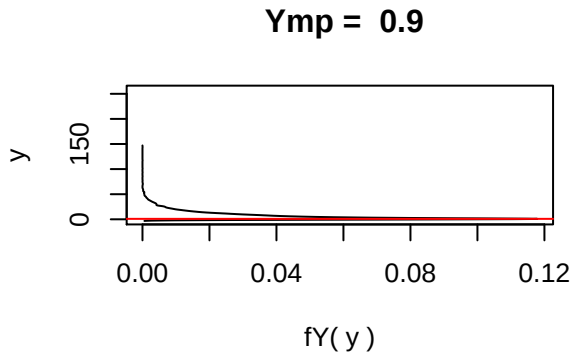


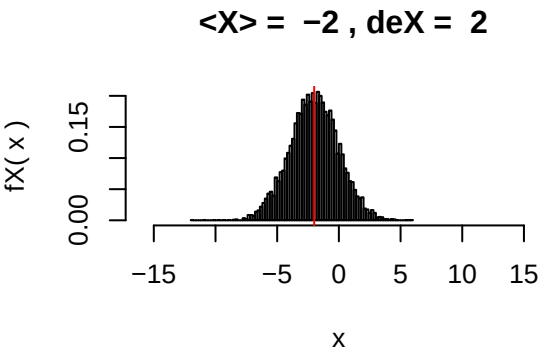

y = y(x) = x²
y
0
150
−15
−5
0
5
10
15
x

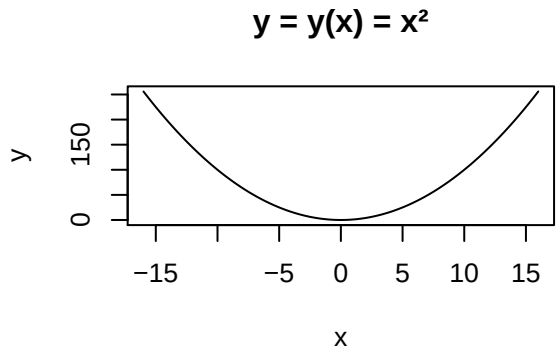

Ymp = 13.85
y
0
150
0.000
0.005
0.010
0.015
0.020
fY(y)

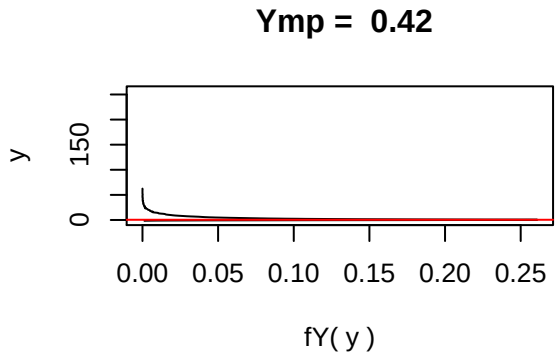

<X> = −5 , deX = 2
fX(x)
0.00
0.15
−15
−5
0
5
10
15
x

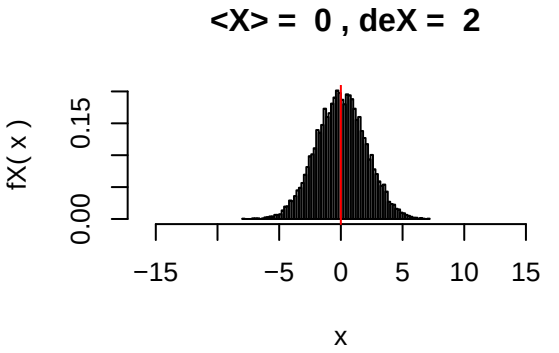

y = y(x) = x²
y
0
150
−15
−5
0
5
10
15
x

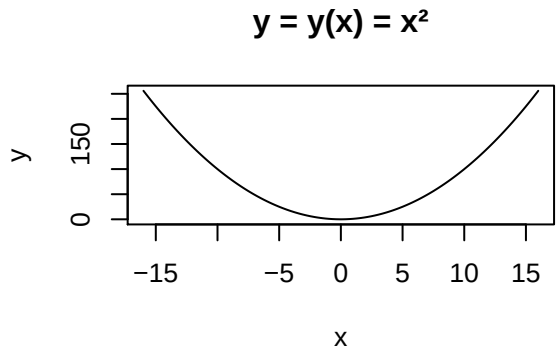

Ymp = 0.9
y
0
150
0.00
0.04
0.08
0.12
fY(y)

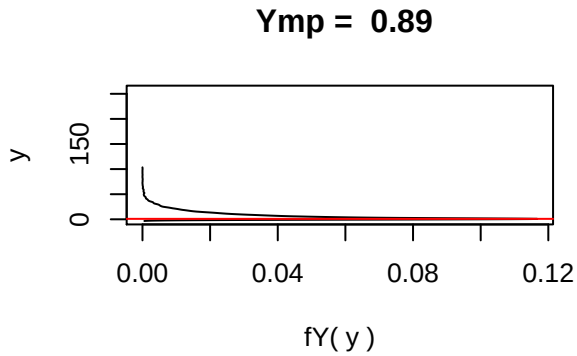

<X> = −2 , deX = 2
fX(x)
0.00
0.15
−15
−5
0
5
10
15
x

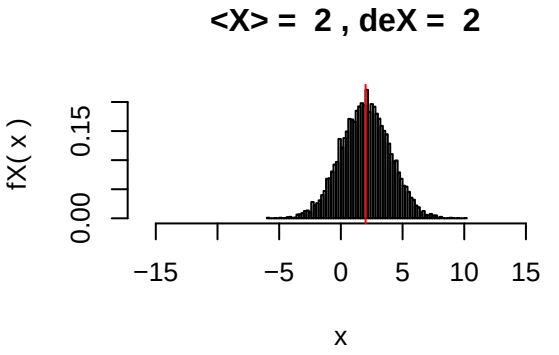

y = y(x) = x²
y
0
150
−15
−5
0
5
10
15
x

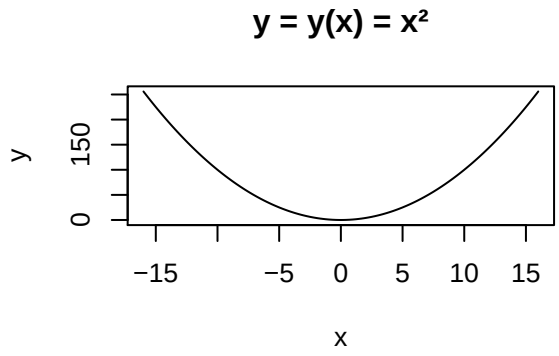

Ymp = 0.42
y
0
150
0.00
0.05
0.10
0.15
0.20
0.25
fY(y)

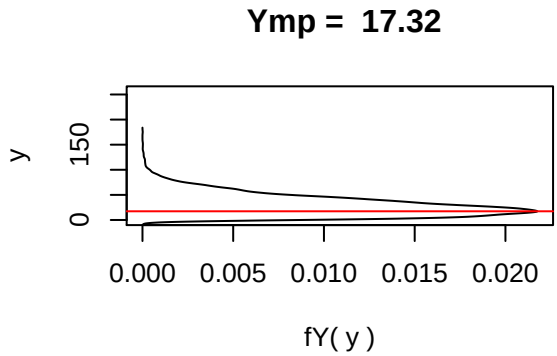

<X> = 0 , deX = 2
fX(x)
0.00
0.15
−15
−5
0
5
10
15
x

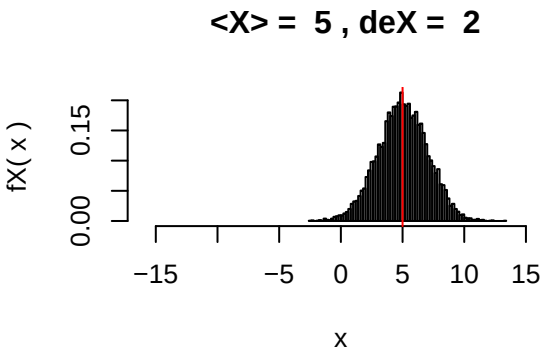

y = y(x) = x²
y
0
150
−15
−5
0
5
10
15
x

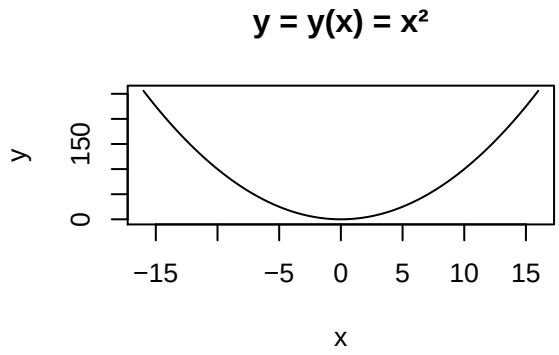

Ymp = 0.89
y
0
150
0.00
0.04
0.08
0.12
fY(y)

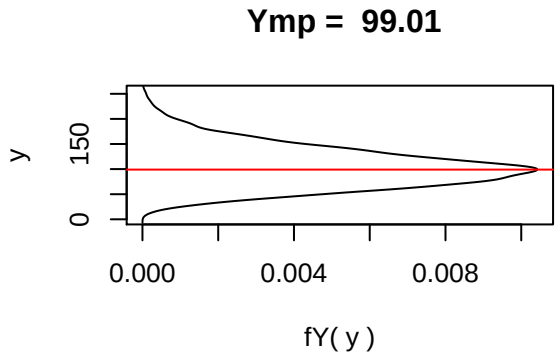

<X> = 2 , deX = 2
fX(x)
0.00
0.15
−15
−5
0
5
10
15
x

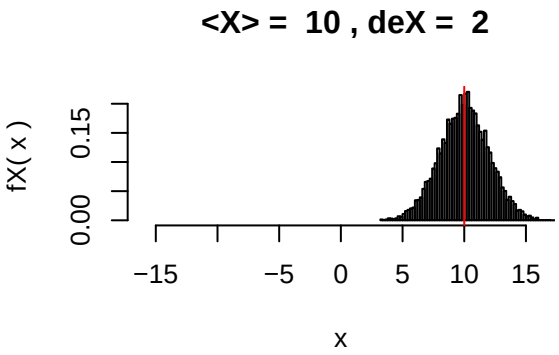

y = y(x) = x²
y
0
150
−15
−5
0
5
10
15
x


Ymp = 17.32
y
0
150
0.000
0.005
0.010
0.015
0.020
fY(y)


<X> = 5 , deX = 2
fX(x)
0.00
0.15
−15
−5
0
5
10
15
x


y = y(x) = x²
y
0
150
−15
−5
0
5
10
15
x


Ymp = 99.01
y
0
150
0.000
0.004
0.008
fY(y)


<X> = 10 , deX = 2
fX(x)
0.00
0.15
−15
−5
0
5
10
15
x