Función Q2

J. Abellán

27 de enero de 2016

Cálculo de la función Q y de la función densidad de probabilidad cuando son **DOS** los parámetros de interés.

Dos parámetros: Q(g, lo)

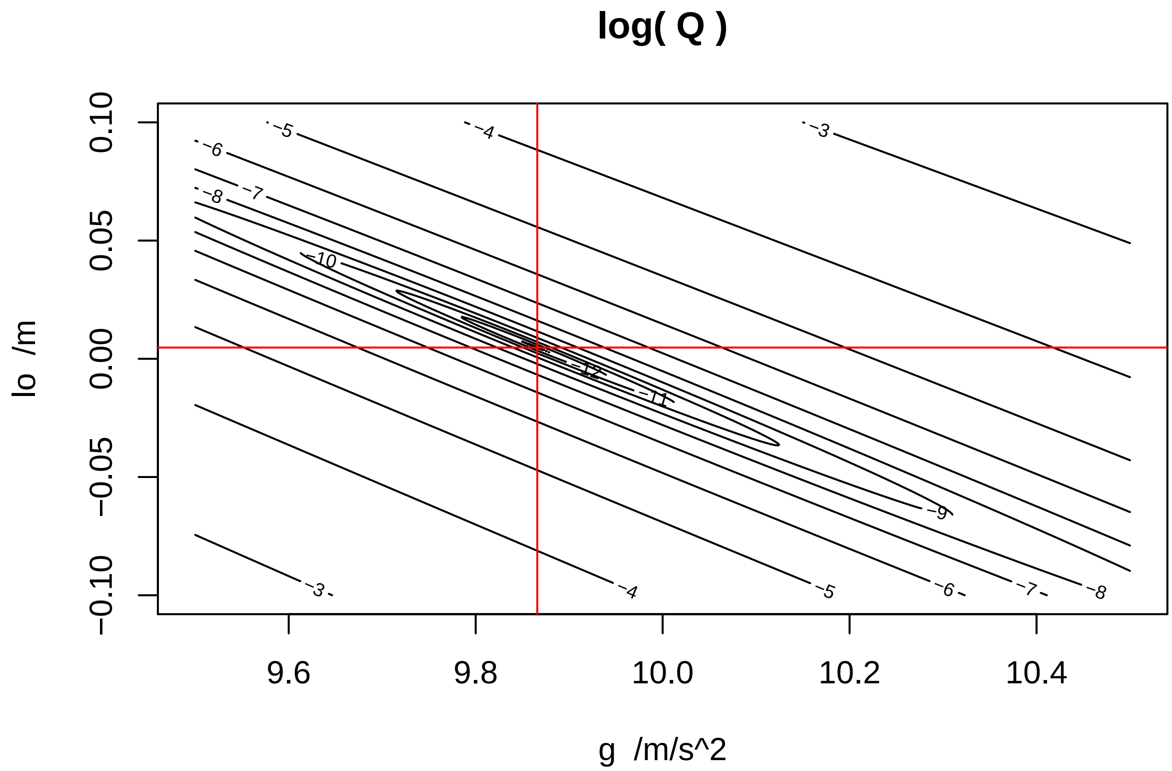
Como ejemplo utilizaremos el péndulo simple considerando, en este caso, que la longitud del mismo viene dada con una incertidumbre fija l_0:

En este caso la teoría depende de dos parámetros desconocidos g, l_0:

$$T = T(g, l_0) = 2\pi\sqrt{\frac{l - l_0}{g}}$$

```r
#Función teórica.
#Péndulo simple con error sistematico en las longitudes
y.teo <- function( l, g, l0 ) { 2 * pi * sqrt( ( l - l0 ) / g ) }

# Los datos
# variable de control: longitud del péndulo (cm)
lexpcm <- c( 180.6, 164.5, 154.3, 137.0 )

# número de datos
Ndatos <- length( lexpcm )

# longitud del péndulo (en metros)
X.exp <- lexpcm / 100

#Variable respuesta: periodo (en segundos)
Y.exp <- c( 2.685, 2.561, 2.482, 2.337 )

#Función Q, calidad del ajuste

#Rango de los parámetros g, l0
#divisones del rango
n <- m <- 400

#Rango de los parámetros
gmin <- 9.5; gmax <- 10.5

l0min <- - .1; l0max <- 0.1

G <- seq( gmin, gmax, len = n )

dG <- ( gmax - gmin ) / ( n - 1 )

L0 <- seq( l0min, l0max, len = m )
```

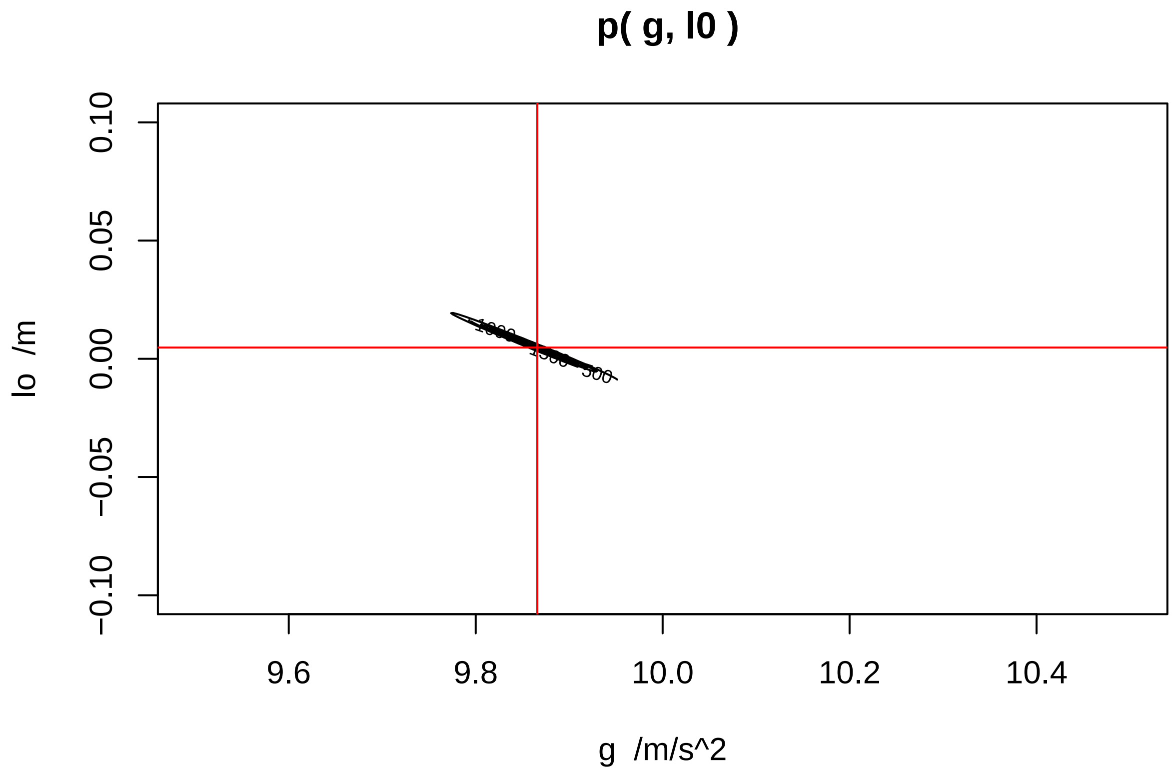
```r
dL0 <-  ( l0max - l0min ) / ( m - 1 )

# Declaración de la matriz Q = Q(g,l0)
Q <- matrix( 0, nrow = n, ncol = m )

# Cálculo de los elementos de Q
for (i in seq( along = G ) ) {

  for (j in seq( along = L0 ) ) {

    Q[ i, j ] <- sum( ( Y.exp - y.teo( X.exp, G[ i ], L0[ j ] ) )^2 )

  }

}

# Cálculo del mínimo de Q
Qmin <- min( Q )

# índice que hace mínimo Q
indice <- which.min( Q )

# fila del mínimo (operación módulo)
imin <- indice %% n

gmin <- G[ imin ]

# columna del mínimo (división entera)
jmin <- ( indice - imin )  %/% m + 1

l0min <- L0[ jmin ]

# Dibujamos las curvas de nivel
contour( G, L0, log( Q ),

         xlab = " g  /m/s^2 ",

         ylab = " lo  /m ",

         main = " log( Q ) "

        )

abline( v = gmin, h = l0min, col = 2 )
```



Bayes

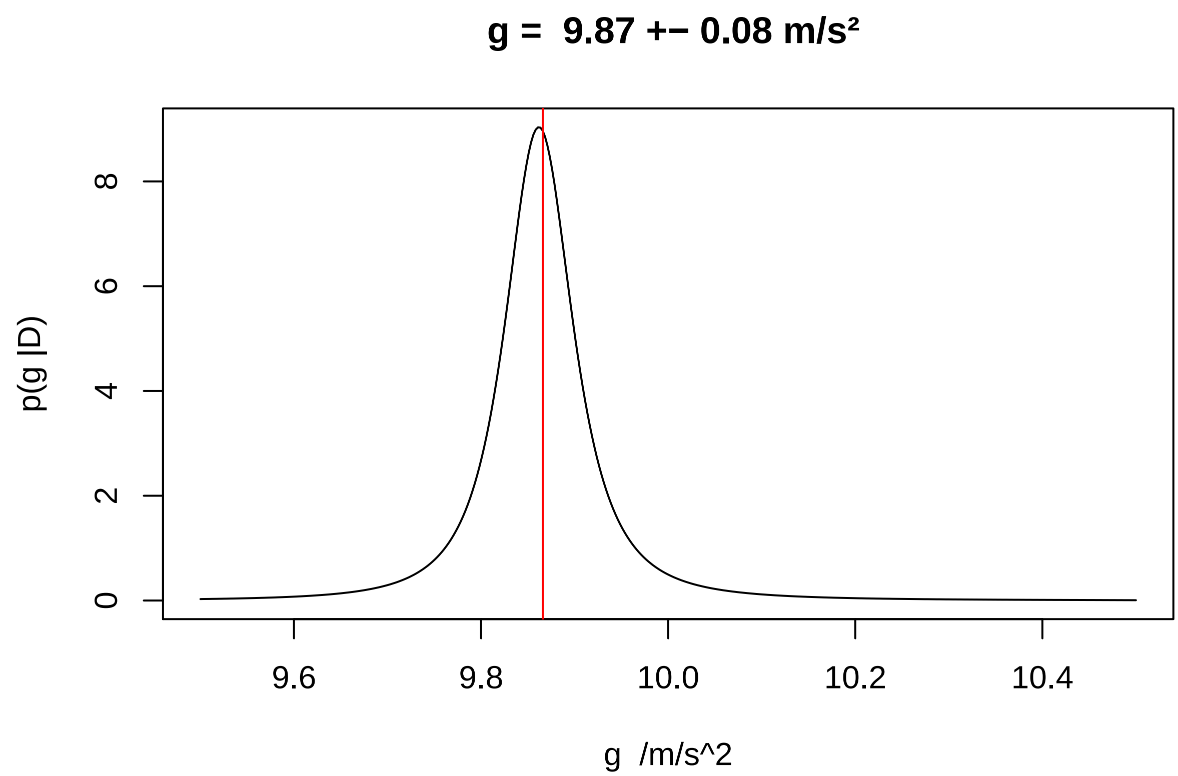
De acuerdo con el teorema de Bayes

```
#Teorema de Bayes
pGL0 <- Q^( - Ndatos / 2 )

#Normalización
pGL0 <- pGL0 / sum( pGL0 * dG * dL0 )

#curvas de nivel
contour( G, L0, pGL0,

         xlab = "g  /m/s^2",

         ylab = "lo  /m",

         main = " p( g, l0 )"

        )

abline( v = gmin, h = l0min, col = 2 )
```

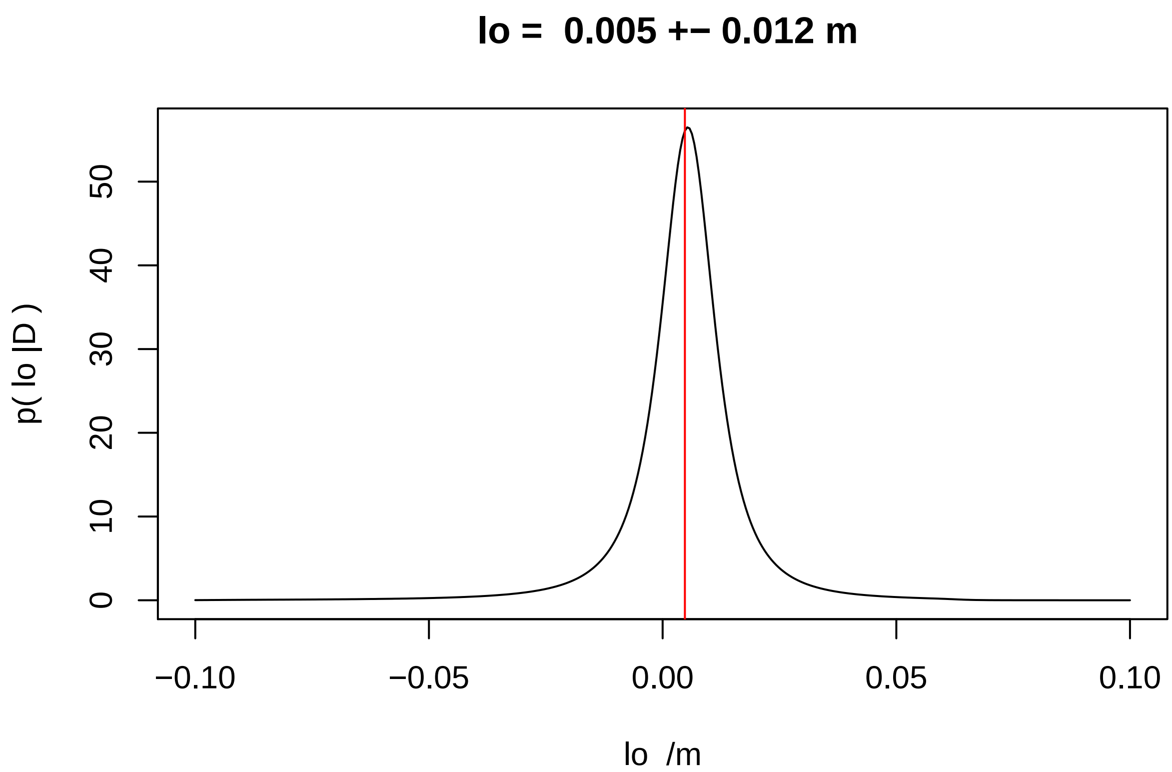



```r
#marginalización y normalización
# aplico por filas la operación suma
pG <- apply( pGL0, 1, sum )

# aplico por columnas la operación suma
pL0 <- apply( pGL0, 2, sum )

# normalización
pG <- pG / sum( pG * dG )

pL0 <- pL0 / sum( pL0 * dL0 )

# estadísticos
# medias
Gm <- sum( G * pG * dG )

L0m <- sum( L0 * pL0 * dL0 )

# varianzas
vG <- sum( ( G - Gm )^2 * pG * dG )

vL0 <- sum( ( L0 - L0m )^2 * pL0 * dL0 )

# desviaciones estándar o errores (incertidumbres)
eG <- sqrt( vG )

eL0 <- sqrt( vL0 )

# dibujo funciones de distribución de G y L0
```

```r
plot( G, pG,
      type = "l",
      xlab = "g  /m/s^2",
      ylab = "p(g |D)",
      main = paste( " g = ", round( Gm, 2), "+-", round( eG, 2 ), "m/s²" )
    )

abline( v = gmin, col = 2 )
```




```r
plot( L0, pL0,
      type = "l",
      xlab = "lo  /m",
      ylab = "p( lo |D )",
      main = paste( " lo = ", round( l0min, 3),"+-", round( eL0, 3 ), "m" )
    )
abline( v = l0min, col = 2 )
```



El ajuste

```
# El ajuste
# Rango de la variable de control
l1 <- 0.95 * min( X.exp )

l2 <- 1.1 * max( X.exp )

n <- 1000

l <- seq( l1, l2, len = n )

# La curva teórica más probable
T.teo <- y.teo( l, Gm, L0m  )

# Superposición de los datos y la curva teórica
plot( X.exp, Y.exp,

      xlab = expression( sqrt( l - lo ) / sqrt( m ) ),

      ylab = "T  /s",

      main = paste("g = ", round( Gm, 3 ),

                   "+-", round( eG, 3 ), "m/s^2" )

    )
```

```r
#el mejor ajuste teórico
lines( l, T.teo, col = 2 )
```