

Segundo Informe Artículo: Estimación robusta del modelo loggamma
generalizado: El paquete `robustloggamma` de R
Seminario de Estadística

Andres Felipe Franco Valero
Luisa Fernanda Guantiva Vargas
Juan Andres Valero Sierra

Presentado a:
Profesor Ramon Giraldo Henao



UNIVERSIDAD NACIONAL DE COLOMBIA

Departamento de Estadística
Universidad Nacional de Colombia
Bogotá, Colombia
Junio de 2022

1. Resumen

El artículo empleado para la elaboración de este informe fue publicado en mayo del 2016 en la Revista de software estadístico por Claudio Agostinelli, Alfio Marazzi, Víctor J. Yohai y Alex Randriamiharisoa [Ago+16]. En el artículo se hace uso de la distribución loggamma generalizada para modelar datos altamente sesgados positivamente como los que provienen de distribuciones logexponencial, logWeibull y loggamma. Además, también se puede usar para datos que provengan de una distribución normal.

La distribución loggamma generalizada es una familia de distribuciones de 3 parámetros denotada por $LG(\mu, \lambda, \sigma)$ donde $\mu, \lambda \in \mathbb{R}$ y $\sigma > 0$. Una variable y que se distribuye $LG(\mu, \lambda, \sigma)$ se obtiene mediante una transformación de localización y de escala $y = \mu + \sigma u$ donde u es una variable aleatoria con función de densidad dada por:

$$f_{\lambda}(u) = \begin{cases} \frac{|\lambda|}{\Gamma(\lambda^{-2})} (\lambda^{-2})^{\lambda^{-2}} \exp((\lambda^{-2})(\lambda u - e^{\lambda u})) & \lambda \neq 0 \\ \frac{1}{\sqrt{2\pi}} \exp(-u^2/2) & \lambda = 0 \end{cases}$$

Por lo tanto, la función de densidad de y es $f_{\theta}(y) = f_{\lambda}((y - \mu)/\sigma)/\sigma$ donde $\theta = (\mu, \sigma, \lambda)$. La distribución normal se obtiene haciendo $\lambda = 0$, la distribución Weibull haciendo $\lambda = 1$, la distribución logexponencial haciendo $\lambda = \sigma = 1$ y la distribución loggamma haciendo $\sigma = \lambda$.

El paquete **robustloggamma** permite calcular la función de densidad (**dloggamma**), la función de distribución (**ploggamma**), los cuantiles (**qloggamma**) y valores aleatorios de la distribución loggamma (**rloggamma**).

Con el objetivo de ajustar un modelo loggamma generalizado a un conjunto específico de datos se estiman los parámetros $\theta = (\mu, \sigma, \lambda)$, para este fin el paquete **robustloggamma** implementa dos estimadores.

2. Metodología

Inicialmente se plantea el uso del estimador $Q\tau$ para obtener las estimaciones de los 3 parámetros y posteriormente se propone el estimador $Q\tau$ ponderado el cual es una mejora del estimador $Q\tau$. Ambos estimadores tienen un buen desempeño cuando el modelo es el correcto y no tienen problemas con la presencia de valores atípicos. Lamentablemente, los dos estimadores mencionados anteriormente no son asintóticamente normales. Sin embargo, sus tasas de convergencia son de orden $n^{1/2}$, lo cual los hace un buen punto de inicio para un procedimiento de verosimilitud ponderada (WL) que es asintóticamente normal y eficiente en el modelo. El paquete **robustloggamma** implementa dos estimadores WL los cuales son la verosimilitud ponderada totalmente iterada (FIWL) y la verosimilitud ponderada de un paso (1SWL).

3. Resultados

El paquete **robustloggamma** ajusta un modelo loggamma generalizado para datos altamente sesgados positivamente. Para la estimación de los parámetros el paquete da la posibilidad de usar 5 métodos.

La función principal del paquete es **loggamma** se puede llamar con la siguiente sintaxis:

```
1 loggamma(x, start=NULL, weights = rep(1, length(x)),  
2 method=c("oneWL", "WQTau", "WL", "QTau", "ML"),...)
```

La descripción de los argumentos es:

- x: Un vector numérico con el conjunto de datos.
- start: Un vector de longitud 3 usado cuando el método es WL, oneWL y ML, de lo contrario, los valores iniciales se obtienen de WQTau en los primeros dos métodos y QTau en el último.
- weights: Un vector de pesos usado en el método QTau.

- method: El método usado. Por defecto está oneWL, el cual es un estimador de verosimilitud ponderada a un paso iniciando desde WQTau.

Como resultado obtenemos los 3 parámetros estimados.

4. Descripción de los datos

Los datos se refieren a 70323 estancias que se observaron en el año 2000 en un grupo de hospitales suizos dentro de un estudio piloto destinado a la implementación de un sistema de agrupación por diagnóstico (GRD). Los sistemas DRG se utilizan en la gestión hospitalaria moderna para clasificar cada estancia individual en un grupo según las características del paciente. Las reglas de clasificación se definen para que los grupos sean lo más homogéneos posible en cuanto a criterios clínicos (diagnósticos y procedimientos) y al consumo de recursos. Normalmente se estima anualmente un coste medio de cada grupo con la ayuda de los datos disponibles sobre las estancias observadas a nivel nacional. Luego, este costo se asigna a cada estadía en el mismo grupo y se utiliza para el reembolso y la elaboración de presupuestos.

Los datos contienen las siguientes variables:

- LOS: Duración de la hospitalización.
- Cost: Costo de las hospitalizaciones en francos suizos.
- MDC: Categoría de diagnóstico principal.
- APDRG: Todos los grupos relacionados con el diagnóstico de pacientes.

Los datos están disponibles en el paquete `robustloggamma`.

5. Código en

```

1 library("xtable")
2 library("lattice")
3 data("drg2000", package = "robustloggamma")
4 APDRG <- c(185, 222, 237, 360)
5 index <- unlist(sapply(APDRG, function(x) which(drg2000$APDRG == x)))
6 print(densityplot(~I(log(Cost)) | factor(APDRG), data = DRG, plot.
7       points = "rug", ref = TRUE))
8 lapply(split(DRG$Cost, DRG$APDRG), summary)
9 lapply(split(DRG$Cost, DRG$APDRG), function(x) c(sd(x), mad(x)))
10 Cost185 <- sort(DRG$Cost[DRG$APDRG == 185])
11 est185 <- loggammarob(log(Cost185))
12 est185
13 summary(est185, p = c(0.9, 0.95, 0.99))
14 which(est185$weights < 0.1)
15 loggammarob.test(est185, lambda = 0)
16 loggammarob.test(est185, mu = 0, sigma = 1)
17 qtau185 <- summary(loggammarob(log(Cost185), method = "QTau"))
18 wqtau185 <- summary(loggammarob(log(Cost185), method = "WQTau"))
19 fiwl185 <- summary(loggammarob(log(Cost185), method = "WL"))
20 oswl185 <- summary(loggammarob(log(Cost185), method = "oneWL"))
21 ml <- summary(loggammarob(log(Cost185), method = "ML"))
22 results <- sapply(APDRG, function(x) analysis(APDRG = x, data = DRG),
23                 simplify = FALSE)
24 weights <- function(x, results) {
25   os <- results[[x]]$os
26   wl <- results[[x]]$wl
27   ans <- t(cbind(os$weights, wl$weights, wl$data, x))
28   return(ans)

```

```

27 }
28 w <- as.data.frame(matrix(unlist(sapply(1:4, function(x) weights(x,
29   results = results))), ncol = 4, byrow = TRUE))
30 colnames(w) <- c("OSWL", "FIWL", "data", "drg")
31 w$drg <- factor(w$drg, labels = APDRG)
32 lattice.theme <- trellis.par.get()
33 col <- lattice.theme$superpose.symbol$col[1:2]
34 print(xyplot(OSWL+FIWL~data | drg, data = w, type = "b", col = col,
35   pch = 21, key = list(text = list(c("1SWL", "FIWL"))),
36   lines = list(col = col)), xlab = "log(Cost)", ylab = "weights"))
37 wilks <- extractwilks(results)
38 wilks <- cbind(c("185", rep(" ", 3), "222", rep(" ", 3), "237", rep(" ",
39   3), "360", rep(" ", 3)), rep(c("statistic", "p$ value", "$\\mu_0
40   $", "$\\sigma_0$"), 4), wilks)
41 xwilks <- xtable(wilks)
42 quant <- function(x, method, results) {
43   res <- results[[x]][[method]]
44   n <- length(res$data)
45   q <- qloggamma(p = ppoints(n), mu = res$mu, sigma = res$sigma,
46     lambda = res$lambda)
47   qconf <- summary(res, p = ppoints(n), conf.level = 0.90)$qconf.int
48   ans <- t(cbind(q, qconf, res$data, res$weights, x, method))
49   return(ans)
50 }
51 q1 <- matrix(unlist(sapply(1:4, function(x) quant(x, method = 1,
52   results = results))), ncol = 7, byrow = TRUE)
53 q2 <- matrix(unlist(sapply(1:4, function(x) quant(x, method = 2,
54   results = results))), ncol = 7, byrow = TRUE)
55 q3 <- matrix(unlist(sapply(1:4, function(x) quant(x, method = 3,
56   results = results))), ncol = 7, byrow = TRUE)
57 q <- as.data.frame(rbind(q1, q2, q3))
58 colnames(q) <- c("q", "qlower", "qupper", "Cost", "weights", "drg", "
59   method")
60 q$drg <- factor(q$drg, labels = APDRG)
61 q$method <- factor(q$method, labels = c("ML", "1SWL", "FIWL"))
62 print(xyplot(Cost ~ q | drg + method, data = q,
63   xlab = "Theoretical Quantiles", ylab = "Empirical Quantiles",
64   fill.color = grey(q$weights), q = q,
65   panel = function(x, y, fill.color, ..., subscripts, q) {
66     fill <- fill.color[subscripts]
67     q <- q[subscripts, ]
68     panel.xyplot(x, y, pch = 21, fill = fill, col = "black", ...)
69     panel.xyplot(x, y = q$qupper, type = "l", col = "grey75")
70     panel.xyplot(x, y = q$qlower, type = "l", col = "grey75")
71   })))

```

Referencias

- [Ago+16] Claudio Agostinelli y col. “Robust Estimation of the Generalized Loggamma Model: The R Package robustloggamma”. En: *Journal of Statistical Software* 70.7 (2016), págs. 1-21. DOI: 10.18637/jss.v070.i07. URL: <https://www.jstatsoft.org/index.php/jss/article/view/v070i07>.