流行病學與生物統計計算Homework 11

學號：b07401048 系級：醫學五 姓名：賴柏瑞

**# Homework7**

**#EX 22-1**

**# n = 30**

*# n = 30*

n <- 30

norep <- 1000

beta <- **c**(-6, 1, 0.005)

y <- **c**()

mle30 <- **matrix**(NA, norep, 3)

for (i in 1 : norep) {

**set.seed**(i)

gpa <- **rnorm**(n = n, mean = 3.1, sd = 0.3)

gre <- **rnorm**(n = n, mean = 580, sd = 80)

linear <- beta[1] + beta[2] \* gpa + beta[3] \* gre

pii <- **exp**(linear) / (1 + **exp**(linear))

x <- **cbind**(**rep**(1, n), gpa, gre)

for (k in 1 : n) {

y[k] <- **sample**(**c**(0, 1), 1, **c**(1 - pii[k], pii[k]), replace = FALSE)

}

**ftn** <- function(betacoef) {

pi1 <- **exp**(x %\*% betacoef) / (1 + **exp**(x %\*% betacoef))

gradient <- **t**(x) %\*% (y - pi1)

hessian <- -**t**(x) %\*% **diag**(**c**(pi1 \* (1 - pi1)), n) %\*% x

return(list(gradient, hessian))

}

**highnew** <- function(ftn, x0, tol, maxiter) {

x <- x0

fx <- ftn(x)

iter <- 0

while ((**max**(**abs**(fx[[1]])) > tol) && (iter < maxiter)) {

x <- x - (**solve**(fx[[2]]) %\*% fx[[1]])

fx <- ftn(x)

iter <- iter + 1

}

if (**max**(**abs**(fx[[1]])) > tol) {

**cat**("algorithm failed to converge\n")

return(NA)

} else {

**cat**("algorithm converges\n")

return(x)

}

}

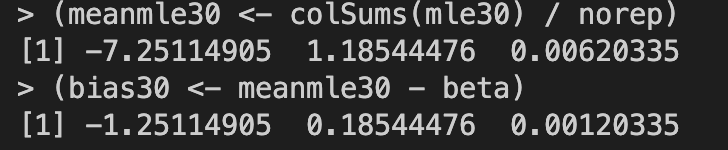
mle30[i, ] <- highnew(ftn, x0 = **c**(0, 0, 0), tol = 1e-9, maxiter = 100)

}

mle30

meanmle30 <- **colSums**(mle30) / norep

(bias30 <- meanmle30 - beta)



MLE of regression coefficient

Bias of MLE of regression coefficient

**# n = 230**

*# n = 230*

n <- 230

norep <- 1000

beta <- **c**(-6, 1, 0.005)

y <- **c**()

mle230 <- **matrix**(NA, norep, 3)

for (i in 1 : norep) {

**set.seed**(i)

gpa <- **rnorm**(n = n, mean = 3.1, sd = 0.3)

gre <- **rnorm**(n = n, mean = 580, sd = 80)

linear <- beta[1] + beta[2] \* gpa + beta[3] \* gre

pii <- **exp**(linear) / (1 + **exp**(linear))

x <- **cbind**(**rep**(1, n), gpa, gre)

for (k in 1 : n) {

y[k] <- **sample**(**c**(0, 1), 1, **c**(1 - pii[k], pii[k]), replace = FALSE)

}

**ftn** <- function(betacoef) {

pi1 <- **exp**(x %\*% betacoef) / (1 + **exp**(x %\*% betacoef))

gradient <- **t**(x) %\*% (y - pi1)

hessian <- -**t**(x) %\*% **diag**(**c**(pi1 \* (1 - pi1)), n) %\*% x

return(list(gradient, hessian))

}

**highnew** <- function(ftn, x0, tol, maxiter) {

x <- x0

fx <- ftn(x)

iter <- 0

while ((**max**(**abs**(fx[[1]])) > tol) && (iter < maxiter)) {

x <- x - (**solve**(fx[[2]]) %\*% fx[[1]])

fx <- ftn(x)

iter <- iter + 1

}

if (**max**(**abs**(fx[[1]])) > tol) {

**cat**("algorithm failed to converge\n")

return(NA)

} else {

**cat**("algorithm converges\n")

return(x)

}

}

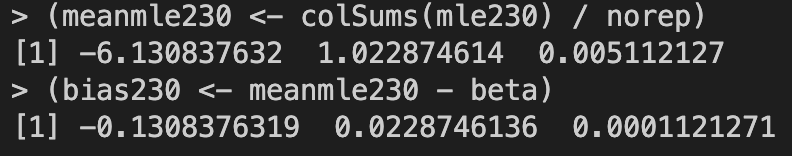
mle230[i, ] <- highnew(ftn, x0 = **c**(0, 0, 0), tol = 1e-9, maxiter = 100)

}

mle230

meanmle230 <- **colSums**(mle230) / norep

(bias230 <- meanmle230 - beta)



MLE of regression coefficient

Bias of MLE of regression coefficient

**# n = 430**

*# n = 430*

n <- 430

norep <- 1000

beta <- **c**(-6, 1, 0.005)

y <- **c**()

mle430 <- **matrix**(NA, norep, 3)

for (i in 1 : norep) {

**set.seed**(i)

gpa <- **rnorm**(n = n, mean = 3.1, sd = 0.3)

gre <- **rnorm**(n = n, mean = 580, sd = 80)

linear <- beta[1] + beta[2] \* gpa + beta[3] \* gre

pii <- **exp**(linear) / (1 + **exp**(linear))

x <- **cbind**(**rep**(1, n), gpa, gre)

for (k in 1 : n) {

y[k] <- **sample**(**c**(0, 1), 1, **c**(1 - pii[k], pii[k]), replace = FALSE)

}

**ftn** <- function(betacoef) {

pi1 <- **exp**(x %\*% betacoef) / (1 + **exp**(x %\*% betacoef))

gradient <- **t**(x) %\*% (y - pi1)

hessian <- -**t**(x) %\*% **diag**(**c**(pi1 \* (1 - pi1)), n) %\*% x

return(list(gradient, hessian))

}

**highnew** <- function(ftn, x0, tol, maxiter) {

x <- x0

fx <- ftn(x)

iter <- 0

while ((**max**(**abs**(fx[[1]])) > tol) && (iter < maxiter)) {

x <- x - (**solve**(fx[[2]]) %\*% fx[[1]])

fx <- ftn(x)

iter <- iter + 1

}

if (**max**(**abs**(fx[[1]])) > tol) {

**cat**("algorithm failed to converge\n")

return(NA)

} else {

**cat**("algorithm converges\n")

return(x)

}

}

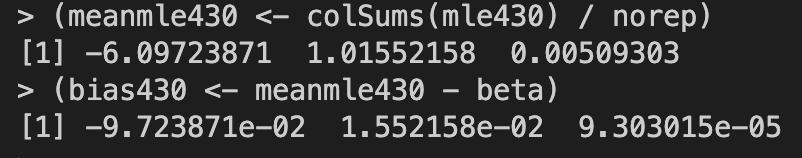
mle430[i, ] <- highnew(ftn, x0 = **c**(0, 0, 0), tol = 1e-9, maxiter = 100)

}

mle430

meanmle430 <- **colSums**(mle430) / norep

(bias430 <- meanmle430 - beta)



MLE of regression coefficient

Bias of MLE of regression coefficient

**# n = 630**

*# n = 630*

n <- 630

norep <- 1000

beta <- **c**(-6, 1, 0.005)

y <- **c**()

mle630 <- **matrix**(NA, norep, 3)

for (i in 1 : norep) {

**set.seed**(i)

gpa <- **rnorm**(n = n, mean = 3.1, sd = 0.3)

gre <- **rnorm**(n = n, mean = 580, sd = 80)

linear <- beta[1] + beta[2] \* gpa + beta[3] \* gre

pii <- **exp**(linear) / (1 + **exp**(linear))

x <- **cbind**(**rep**(1, n), gpa, gre)

for (k in 1 : n) {

y[k] <- **sample**(**c**(0, 1), 1, **c**(1 - pii[k], pii[k]), replace = FALSE)

}

**ftn** <- function(betacoef) {

pi1 <- **exp**(x %\*% betacoef) / (1 + **exp**(x %\*% betacoef))

gradient <- **t**(x) %\*% (y - pi1)

hessian <- -**t**(x) %\*% **diag**(**c**(pi1 \* (1 - pi1)), n) %\*% x

return(list(gradient, hessian))

}

**highnew** <- function(ftn, x0, tol, maxiter) {

x <- x0

fx <- ftn(x)

iter <- 0

while ((**max**(**abs**(fx[[1]])) > tol) && (iter < maxiter)) {

x <- x - (**solve**(fx[[2]]) %\*% fx[[1]])

fx <- ftn(x)

iter <- iter + 1

}

if (**max**(**abs**(fx[[1]])) > tol) {

**cat**("algorithm failed to converge\n")

return(NA)

} else {

**cat**("algorithm converges\n")

return(x)

}

}

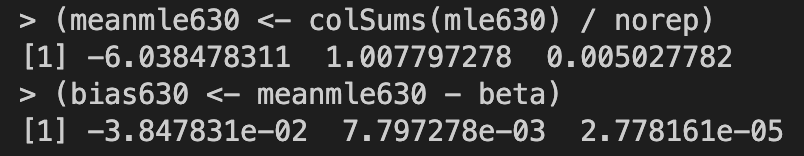
mle630[i, ] <- highnew(ftn, x0 = **c**(0, 0, 0), tol = 1e-9, maxiter = 100)

}

mle630

meanmle630 <- **colSums**(mle630) / norep

(bias630 <- meanmle630 - beta)



MLE of regression coefficient

Bias of MLE of regression coefficient

**# plotting**

**setwd**("/Users/raymond/Desktop/")

**png**(filename = "logistic1.png", width = 2000, height = 6000, res = 500)

**par**(mfrow = **c**(3, 1))

**boxplot**(mle30[, 1], mle230[, 1], mle430[, 1], mle630[, 1], main = "beta0")

**abline**(h = beta[1], col = 2)

**boxplot**(mle30[, 2], mle230[, 2], mle430[, 2], mle630[, 2], main = "beta1")

**abline**(h = beta[2], col = 2)

**boxplot**(mle30[, 3], mle230[, 3], mle430[, 3], mle630[, 3], main = "beta2")

**abline**(h = beta[3], col = 2)

**dev.off**()

**png**(filename = "logistic2.png", width = 6000, height = 6000, res = 500, main = "beta0")

**par**(mfrow = **c**(1, 1))

**boxplot**(mle30[, 1], mle230[, 1], mle430[, 1], mle630[, 1])

**abline**(h = beta[1], col = 2)

**dev.off**()

**png**(filename = "logistic3.png", width = 6000, height = 6000, res = 500, main = "beta1")

**boxplot**(mle30[, 2], mle230[, 2], mle430[, 2], mle630[, 2])

**abline**(h = beta[2], col = 2)

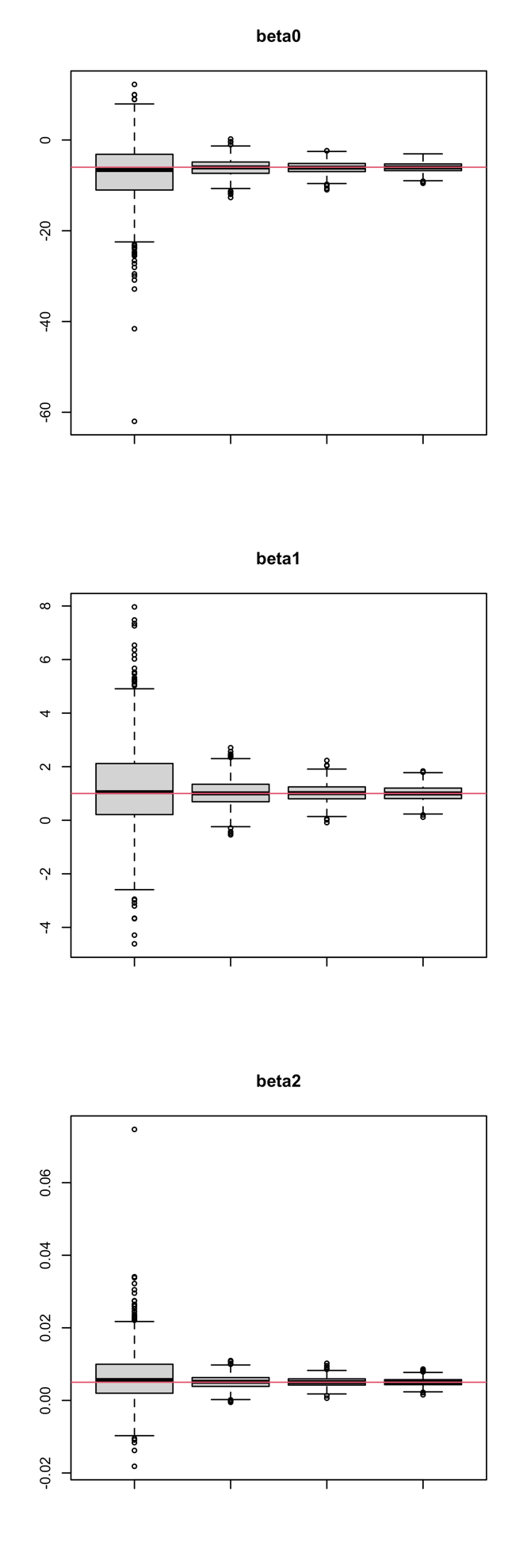
**dev.off**()

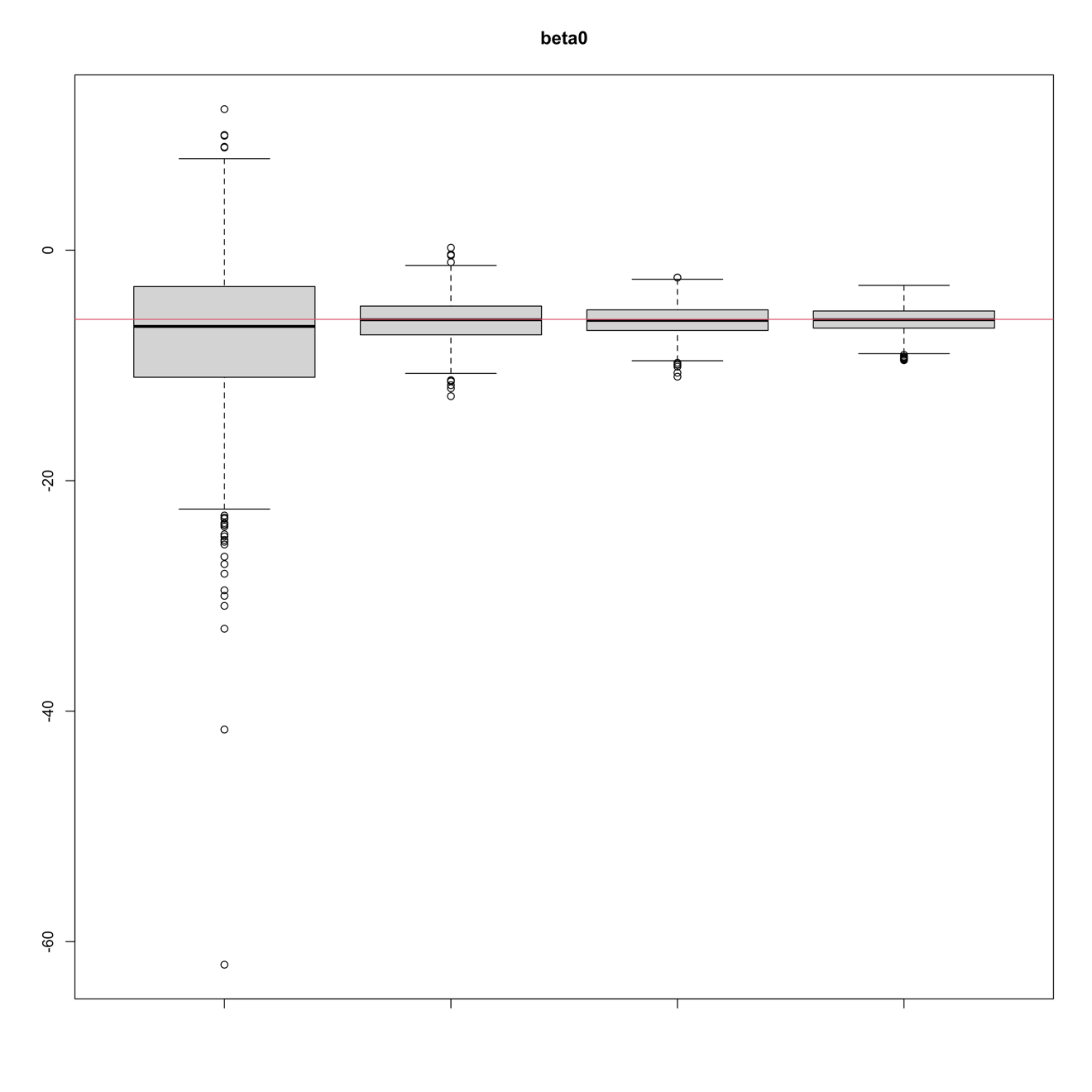
**png**(filename = "logistic4.png", width = 6000, height = 6000, res = 500, main = "beta2")

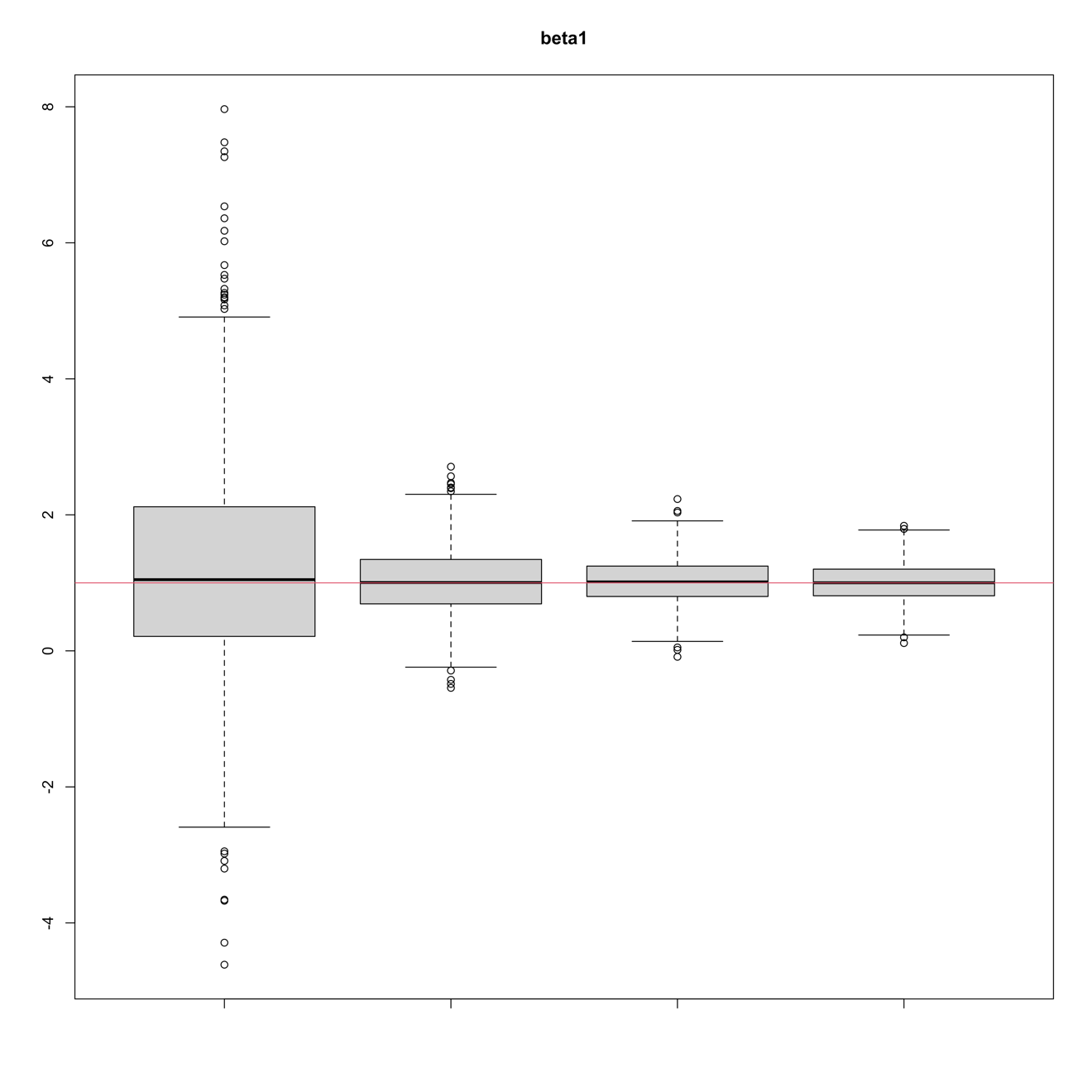
**boxplot**(mle30[, 3], mle230[, 3], mle430[, 3], mle630[, 3])

**abline**(h = beta[3], col = 2)

**dev.off**()

****

****

****

