Gamma MLE by nlm() and by Fisher Scoring

```
> gamma.mlen
function (xx, shape0, rate0)
{
    negLL <- function(p, x) ifelse(p[1] <= 0 | p[2] <= 0, 1e+50,
        -sum(log(dgamma(x, p[1], p[2]))))
    nlm(negLL, p = c(shape0, rate0), hessian = T, x = xx)
}

> gamma.mles
function (xx, shape0, rate0)
{
    n <- length(xx)
    xbar <- mean(xx)
    logxbar <- mean(log(xx))
    theta <- c(shape0, rate0)
    repeat {
        theta0 <- theta
        shape <- theta0[1]
        rate <- theta0[2]
        S <- n * matrix(c(log(rate) - digamma(shape) + logxbar,
            shape/rate - xbar), ncol = 1)
        I <- n * matrix(c(trigamma(shape), -1/rate, -1/rate,
            shape/rate^2), ncol = 2)
        theta <- theta0 + solve(I) %*% S
        if (max(abs(theta - theta0)) < 1e-08)
            break
    }
    list(estimates = theta, infmat = I)
}

> gamma.mles1
function (xx, shape0, rate0)
{
    n <- length(xx)
    xbar <- mean(xx)
    logxbar <- mean(log(xx))
    theta <- c(shape0, rate0)
    repeat {
        theta0 <- theta
        shape <- theta0[1]
        rate <- theta0[2]
        S <- n * matrix(c(log(rate) - digamma(shape) + logxbar,
            shape/rate - xbar), ncol = 1)
        I <- n * matrix(c(trigamma(shape), -1/rate, -1/rate,
            shape/rate^2), ncol = 2)
        theta <- theta0 + solve(I) %*% S
        repeat {
            if (min(theta) > 0)
                break
            theta <- (theta0 + theta)/2
        }
        if (max(abs(theta - theta0)) < 1e-08)
            break
    }
    list(estimates = theta, infmat = I)
}
```