Likelihood_v2.R

georges

2024-10-11

Contents

```
library(latticeExtra)
```

```
## Loading required package: lattice
```

```
library(spida2)
Green <- "#008800"
#
# Samples from the exponential distribution:
#
# e.g.
# - wait times where theta is mean waiting time
# - lifetimes of non-aging components with no lemons where theta is
#   mean lifetime
#
#   f(x,theta) = exp(-x/theta)/theta ;   x, theta > 0
```

```r
#

#
#  Exponential distribution with mean parametrization
#

{

  # Maximum likelihood estimator

  mle <- function(x) mean(x)

  # log-likelihood

  lx <- function(x,thetas) {
    # log-likelihood from sample x from exponential theta
    # evaluated for a fixed sample x over thetas
    n <- length(x)
    sx <- sum(x)
    -n*log(thetas) - sx / thetas
  }

  lxn <- function(x, thetas) {
    # normed log-likelihood
    lx(x, thetas) - lx(x, mle(x))
  }

  # Score: derivative of log-likelihood: gradient

  lx. <- function(x,thetas) {
```

```r
  # slope (first derivative) of log-likelihood
  n <- length(x)
  sx <- sum(x)
  -n/thetas + sx / thetas^2
}

# Negative of Observed Fisher Information: second derivative of log-likelihood: Hessian

lx.. <- function(x,thetas) {
  # curvature (second derivative) of log-likelihood
  n <- length(x)
  sx <- sum(x)
  n/thetas^2 - 2 * sx / thetas^3
}

qappx <- function(x, thetas) {
  lx0 <- lx(x, mle(x))
  lxq <- lx..(x, mle(x))
  lx0 + lxq * (thetas - mle(x))^2
}

# Generate random sample using mean parameter

rexp2 <- function(n, theta) {
  # random sample of size n
  # using waiting time as parameter, instead of rate
  x <- rexp(n, rate = 1/theta)
  attr(x,'theta') <- theta
  class(x) <- 'sample'
  print(as.vector(x))
```

```
    x
  }
}
graphics.off()

{
  # setup

  n <- 30
  theta_true <- 2.5

  low <- lx(rep(3*theta_true,n), 3 * theta_true)
  par(mar=c(0,0,0,0))
  plot(c(0,6),c(low,0), type = 'n')
  axis(1, pos = low + 1)
  ths <- seq(.2, 6, .01)

  col <- 'blue'
  abline(v = theta_true, col = 'red')
}
{
  n <- 3
  # log-likelihood
  x <- rexp(n, 1/theta_true )
  print(x)
  lines(ths, lx(x, ths))
  points(mle(x), lx(x, mle(x)), pch = 16, col = col)
  points(mle(x), low + 1, pch = 16, col = col)
  # lines(ths, qappx(x, ths), col = 'red')
}
```

```
## [1] 5.433962 1.699523 1.110298
```

```
{

  # log-likelihood

  col <- 'blue'
  theta_true <- 2.5
  ths <- seq(.02, 3 * theta_true, .01)
  plot(c(0, 3* theta_true), c(-20, 1), type = 'n')
  axis(1, pos = -9)
  abline(v = theta_true, col = 'red')

  n <- 10
  {
    # Score
    x <- rexp(n, 1/2.5 )
    print(x)
    lines(ths, lx(x, ths))
    points(mle(x), lx(x, mle(x)), pch = 16, col = col)
    points(mle(x), -9, pch = 16, col = col)

  }
}
```

```
##  [1] 0.1718229 1.6076075 0.4604620 7.6235563 1.5561371 3.3219473 1.6659668
##  [8] 2.0467795 2.8340127 1.2070569
```

```
{
  plotsetup.ex <- function(xlim = c(0,5)){
    graphics.off()
    plot(1, type = 'n',xlim = xlim, ylim = c(-10,4),
         xlab = expression(~theta), bty = 'n',
         ylab = expression(~l[x](theta)), axes = FALSE)
    lhat <- -1
    axis(1,pos = lhat)
    text(-.2,lhat, expression(~hat(theta)), xpd = TRUE)
    axis(1,pos = -10)
    inc <- .8
    lp0 <- .5
    lpp0 <- .5 + inc
    lptrue <- .5 + 2* inc
    lpptrue <- .5 + 3 * inc
    lpphat <- .5 + 4 * inc
    seg <- function(h) segments(xlim[1]+.2,h,xlim[2],h)

    text(0,lp0,expression(l*minute[x](theta[0])))
    seg(h = lp0)
    text(0,lpp0,expression(l*second[x](theta[0])))
    seg(h = lpp0)
    text(0,lptrue, expression(l*minute[x](theta[TRUE])))
    seg(h=lptrue)
    text(0,lpptrue, expression(l*second[x](theta[TRUE])))
    seg(h=lpptrue)
    text(0,lpphat, expression(l*second[x](hat(theta))))
    seg(h=lpphat)

    segments(mean(xlim)+c(0,1,-1),.5,mean(xlim)+c(0,1,-1),.3)
```

```
    text(mean(xlim)+c(0,1,-1),.3, c(0,1,-1), pos = 1)
    # text(0,.5+.2,expression(l*minute[x](theta[0])))
    # seg(h = .5)
    # text(0,0 +.2,expression(l*second[x](theta[0])))
    # seg(h = 0)
    # segments(mean(xlim)+c(0,1,-1),0,mean(xlim)+c(0,1,-1),-0.1)
    # text(mean(xlim)+c(0,1,-1),-0.1, c(0,1,-1), pos = 1)
    #
    parms <- list(lhat = lhat,lp0 =lp0,lpp0 = lpp0)
    parms
  }



  }


{

  plotsetup <- function(
    xlim=c(0,5), ylim = c(-10,5)    ){
    # graphics.off()
    par(mar=c(0,0,0,0))
    plot(xlim,ylim, type = 'n', bty = 'n', axes = FALSE)
    # axis(1,pos=ylim[1])
    # axis(2,pos=0)

  }
}
plotsetup()
```

```r
ths <- seq(.02,7,.01)
plot_lx <- function(xs, theta_true = NULL, theta_0 = NULL, ...){
  plotsetup()
  axy <- par('usr')[3]+1
  lapply(xs, function(x) {
    lines(ths, lx(x, ths))
    points(mle(x), lx(x, mle(x)), pch = 16, col = 'blue')
    points(mle(x), axy, pch = 16, col = 'blue')
  }
  )
  axis(1, pos = axy)
  if(!is.null(theta_true)) abline(v = theta_true, col = 'red')
  if(!is.null(theta_0)) abline(v = theta_0, col = Green)
}

theta_true <- 2.5
xs <- lapply(rep(3,3), rexp, 1/theta_true )
xs
```

```
## [[1]]
## [1] 0.16001358 0.02100876 3.53488134
##
## [[2]]
## [1] 3.02908723 0.04197061 0.63312979
##
## [[3]]
## [1] 4.286556 2.563531 1.819863
```

```r
plot_lx(xs[c(1,2,3)], 2.5, 2)

plot_lx. <- function(xs, theta_true = NULL, theta_0 = NULL, ...){
  plotsetup()
  plot_lx(xs, theta_true = theta_true, theta_0 = theta_0, ...)
  lapply(xs, function(x) {
    lines(ths, lx.(x,ths))
    points(mle(x), 0, pch = 16, col = 'blue')
  })
  axis(1, pos = 0)
}

plot_lx.(xs, theta_true = 2.5)

plot_lx.p <- function(xs, theta_true = NULL, theta_0 = NULL, ...){
  plotsetup()
  plot_lx(xs, theta_true = theta_true, theta_0 = theta_0, ...)
  lapply(xs, function(x) {
    lines(ths, lx.(x,ths))
    points(mle(x), 0, pch = 16, col = 'blue')
    if(!is.null(theta_true)) points(theta_true, lx.(x, theta_true), pch = 16, col = 'red')
    if(!is.null(theta_0)) points(theta_0, lx.(x, theta_0), pch = 16, col = Green)
  })
  axis(1, pos = 0)
}
plot_lx.p(xs, theta_true = 2.5, theta_0 = 2)


plot_lx.o <- function(xs, theta_true = NULL, theta_0 = NULL, ...){
  plotsetup(ylim = c(-3,3))
```

```
  lapply(xs, function(x) {
    lines(ths, lx.(x,ths))
    points(mle(x), 0, pch = 16, col = 'blue')
    if(!is.null(theta_true)) points(theta_true, lx.(x, theta_true), pch = 16, col = 'red')
    if(!is.null(theta_0)) points(theta_0, lx.(x, theta_0), pch = 16, col = Green)
  })
  axis(1, pos = 0)
  if(!is.null(theta_true)) abline(v = theta_true, col = 'red')
  if(!is.null(theta_0)) abline(v = theta_0, col = Green)
  lapply(xs, function(x) {
    lines(ths, lx..(x,ths), lty = 3)

    if(!is.null(theta_true)) points(theta_true, lx..(x, theta_true), pch = 0, col = 'red')
    # if(!is.null(theta_0)) points(theta_0, lx.(x, theta_0), pch = 16, col = Green)
  })
}
plot_lx.o(xs, theta_true = 2.5, theta_0 = 2)
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
```

```
xs <- lapply(rep(3,10), rexp, 1/theta_true)
```

```r
for(n in 1:10){
plot_lx.o(xs[1:n], theta_true = 2.5)
  # Sys.sleep(2)
}

xs <- lapply(rep(20,10), rexp, 1/theta_true)

for(n in 1:10){
  plot_lx.o(xs[1:n], theta_true = 2.5)
  # Sys.sleep(2)
}
xs <- lapply(rep(100,10), rexp, 1/theta_true)

for(n in 1:10){
  plot_lx.o(xs[1:n], theta_true = 2.5)
  # Sys.sleep(2)
}
```

+++