S-PLUS code for Estimating fish concentrations using trawl data

Sue Paul G A Vignaux

Abstract

This documents S-PLUS code that was used in the studies reported in Paul and Vignaux (1999). Code to generate random trawls and to estimate the location of rectangular or circular 'schools' of fish covered by the trawls are documented. Functions to plot the trawl locations and to make contour plotes of posterior probability are also listed.

This is *Revision* : 1.9.

Contents

1 Introduction

This S-PLUS code is used in the investigation of techniques for the analysis of the parameter method of estimating the location and size of schools of fish using random trawl data.

2 Generating Random Trawls

A set of trawls is generated randomly and can be pictured or used in the development of a posterior probability countour plot or a maximisation. A trawl set is given a unique identifier.

2.1 Uniformly distributed length trawls

maketrawls generates a set of n trawls with start and end-points uniformly distributed between a and b. It returns an $n \times 2$ matrix. Column 1 containes the starting points.

$\langle$*maketrawls*$\rangle \equiv$

```
maketrawls<-function(a = -5, b = 5, n = 50){
### generates a set of random trawl start and end positions,
### at random points in the range a,b
        s <- runif(n, a, b)
        t <- runif(n, a, b)
        s1 <- ifelse(s < t, s, t) #starting point must be less
        t1 <- ifelse(s < t, t, s) #than the finish point.
        cbind(s1, t1)
}
```

2.2 Constant length trawls

maketrawls2 develops a set of n trawls of constant length, *size*, with starting points randomly positioned between a and b. Returns a $n \times 2$ matrix.

⟨*maketrawls2*⟩≡

```
maketrawls2<-function(a, b, n, size){
  s2 <- runif(n, a, b)
  t2 <- s2 + rep(size, length(s2))
  cbind(s2, t2)
}
```

2.3 Exponential trawls

maketrawls3 develops a set with n trawls of exponentially distributed lengths, mean $1/\theta$, with starting points randomly positioned between a and b. It returns a $n \times 2$ matrix of start and end points for the trawls.

⟨*exponential trawls*⟩≡

```
maketrawls3<-function(a, b, n, theta){
        s3 <- runif(n, a, b)
        trawl.length <- rexp(n, theta)
        t3 <- s3 + trawl.length
        cbind(s3, t3)
}
```

2.4 Uniform trawls

maketrawls4 generates a set of trawls uniformly distributed in length. It returns an $n \times 3$ matrix with the trawl starting and finishing positions and the length.

⟨*uniform trawls*⟩≡

```
maketrawls4<-function(a, b, n, R = 1, k = 2){
  start <- vector(length = n)
  finish <- vector(length = n)
  halflength <- vector(length = n)
        halfdist <- vector(length = n)
        length <- vector(length = n)
        anew <- vector(length = n)
        bnew <- vector(length = n)
        midpoints <- runif(n, a, b)
        for(i in 1:n) {
                anew[i] <- midpoints[i] - k * R
                bnew[i] <- midpoints[i] + k * R
                halflength[i] <- runif(1, anew[i], bnew[i])
                halfdist[i] <- abs(halflength[i] - midpoints[i])
                start[i] <- midpoints[i] - halfdist[i]
                finish[i] <- midpoints[i] + halfdist[i]
                length[i] <- finish[i] - start[i]
        }
        cbind(start, finish, length)
}
```

2.5 Horizontal trawls

hortrawls generates horizontal trawls with uniform start and finish positions. It also generates uniformly a vertical (y) position for the trawls. The starting and finishing y values are the same. This routine is used in the two-dimensional case.

⟨*horizontal*⟩≡

```
hortrawls<-function(xmin, xmax, ymin, ymax, n, size)
{
        startx <- runif(n, xmin, xmax)  #generate n starting x coordinates
        finishx <- startx + size
        starty <- runif(n, ymin, ymax)  #finishy<-starty
        cbind(startx, starty, finishx)
}
```

2.6 Combined trawl generation routines

Here all the routines for generating trawls are collected together into a file `generate`.

$\langle$*generate*$\rangle\equiv$
  $\langle$*do not edit note*$\rangle$
  $\langle$*maketrawls*$\rangle$
  $\langle$*maketrawls2*$\rangle$
  $\langle$*exponential trawls*$\rangle$
  $\langle$*uniform trawls*$\rangle$
  $\langle$*horizontal*$\rangle$
  $\langle$*trawls.q end*$\rangle$

3 The one-dimensional Model

This section deals with the one-dimensional case. The overlap for a set of trawls is calculated by the S-PLUS function `ovrlap` for a matrix of values for x and R. This is done using S-PLUS array calculations on the $x \times R$ array. The key idea is to use the *outer* function with $<$ and $>$ operators to construct a *MATCH* logical array for each special case of overlap. For example, the special cases where some trawls do not overlap the school at all. The logical arrays are then progressively used to insert the overlap calculations into the final overlap matrix, *ov*.

⟨*routines*⟩≡

```
⟨do not edit note⟩
ovrlap<-function(x,R,ss,tt){
    l1<-length(x)
      l2<-length(R)
      l3<-length(ss)
      zero<-rep(0,l3)
      y<-outer(x,R,"-")
      z<-outer(x,R,"+")
      ov<-array(0,c(l1,l2,l3))
      dimnames(ov)<-list(x,format(R),ss)

    ov<-outer((z-y),zero,"+")

    MATCH1<-outer(y,tt,">")|outer(z,ss,"<")
    ov[MATCH1]<-0

    MATCH3<-outer(y,tt,"<")&outer(y,ss,">")
    ov[MATCH3]<- -outer(y,tt,"-")[MATCH3]

    MATCH4<-outer(z,ss,">")&outer(z,tt,"<")
    ov[MATCH4]<- outer(z,ss,"-")[MATCH4]

    MATCH5<-outer(y,ss,"<")&outer(z,tt,">")
    ov[MATCH5]<- outer(y-y,tt-ss,"+")[MATCH5]

    MATCH6<-outer(y,ss,">")&outer(z,tt,"<")
    ov[MATCH6]<- outer(z-y,zero,"+")[MATCH6]
    ov
  }
⟨genobs⟩
⟨Alogpost⟩
⟨genobs2⟩
```

⟨*trawls.q end*⟩

3.1 Generate a sample of data

Generate a sample of observed catches given a set of trawls and a location for the school.

⟨*genobs*⟩≡
```
#2.  Get observed catches (sample)
genobs<-function(x,R,ss,tt,sig){
    n<-length(ss)
    E<-rnorm(n,0,sig)
    catch<-as.vector(ovrlap(x,R,ss,tt))  #(x and R are scalars here)
    obs<-catch + E
    obs
  }
```

3.2 Calculate a log-probability matrix

`Alogpost` constructs the the log-posterior probability matrix for a range of m values of both x and R. The routine uses the `ovrlap` function to obtain a matrix of overlaps and then converts these to a $m \times m$ log-probability matrix which is returned with the actual vectors giving the list of x and R that have been used.

⟨*Alogpost*⟩≡
```
Alogpost<-function(minx,maxx,minR,maxR,obs,datmat,m,sig=0.1){
  ss<-datmat[,1]
  tt<-datmat[,2]
  x<-seq(from=minx, to=maxx, length =m)
  R<-seq(from=minR, to=maxR, length =m)
  theor<-ovrlap(x,R,ss,tt)
  n<-length(obs)
  const<-( - n/2) * log(2 * pi * sig^2)
  xR0<-array(0,dim(theor)[c(1,2)])
  Aobs<-outer(xR0,obs,"+")
  d2<--0.5*((Aobs-theor)/sig)^2
 chi<- apply(d2,c(1,2),sum)
 likmat<-chi+const
 likmat<-likmat-max(likmat)
  list(x=x,R=R,likmat=likmat)
}
```

3.3 Calculate the overlap when there are two schools

If there are two schools the overlap for a trawl may contain components from both schools if it is long enough. The total overlap is the sumof the overlaps from each of the schools. This routine is used in generating test data.

⟨*genobs2*⟩≡

```
#One dimensional school:  two blobs
genobs2<-function(x1,R1,x2,R2,ss,tt,sig){
    n<-length(ss)
    E<-rnorm(n,0,sig)
    o1<-as.vector(ovrlap(x1,R1,ss,tt))
    o2<-as.vector(ovrlap(x2,R2,ss,tt))
    totalcatch<-o1+o2
    obs<-totalcatch+E
    obs
  }
```

4 The two-dimensional case: a rectangular School of fish

A more compicated overlap function has to be used for this case because the trawls not only have start and end points but also displacements in the y direction.

For the moment we continue to use only horizontal trawls.

⟨*routines3*⟩≡

⟨*do not edit note*⟩

```
overlap5<-function(x,y,Rx,Ry,xstart,ystart,xend) {
### calculates overlaps of trawls onto a rectangle.
### x,y is the location of centre of rectangle
### Rx,Ry the half-lengths in each direction
### xstart, xend (v) are starts and ends of horiz trawls
### ystart(v) their vertical position.

  l1<-length(x)
  l2<-length(y)
  l3<-length(Rx)
  l4<-length(Ry)
  l5<-length(xstart)
  zero<-rep(0,l5)
  yRy0<-array(0,c(l2,l4))             # l2 l4
  xRx0<-array(0,c(l1,l3))             # l1 l3
  xRx1<-outer(x,Rx,"-")               # l1 l3
  xRx1<-outer(xRx1,yRy0,"+")          # l1 l3 l2 l4
  xRx2<-outer(x,Rx,"+")
  xRx2<-outer(xRx2,yRy0,"+")          # l1 l3 l2 l4
  yRy1<-outer(y,Ry,"+")
  yRy1<-outer(xRx0,yRy1,"+")          # l1 l3 l2 l4
  yRy2<-outer(y,Ry,"-")
  yRy2<-outer(xRx0,yRy2,"+")          # l1 l3 l2 l4
  ov<-array(9999,c(l1,l3,l2,l4,l5))  # l1 l3 l2 l4 l5

 ##zero overlaps (1) and (2)
  MATCH<-outer(yRy2,ystart,">")|outer(yRy1,ystart,"<=")|
  outer(xRx1,xend,">")|outer(xRx2,xstart,"<=")

  ov[MATCH]<-0

  ##Partial Overlaps (3)
  MATCH<-outer(yRy2,ystart,"<=")&outer(yRy1,ystart,">")&
```

```
  outer(xRx1,xstart,">")&outer(xRx1,xend,"<=")&outer(xRx2,xend,">")
  ov[MATCH]<- - outer(xRx1,xend,"-")[MATCH]

  ## partial (4)
  MATCH<-outer(yRy2,ystart,"<=")&outer(yRy1,ystart,">")&
  outer(xRx2,xend,"<=")&outer(xRx1,xstart,"<=")&outer(xRx2,xstart,">")
  ov[MATCH]<- outer(xRx2,xstart,"-")[MATCH]

  ##Overlaps contained entirely within the square
  MATCH<-outer(yRy2,ystart,"<=")&outer(yRy1,ystart,">")&
  outer(xRx1,xstart,"<=")&outer(xRx2,xend,">")
  ov[MATCH]<-outer(xRx1-xRx1,xend-xstart,"+")[MATCH]

  ##Overlaps which go right through the square
  MATCH<-outer(yRy2,ystart,"<=")&outer(yRy1,ystart,">")&
  outer(xRx1,xstart,">")&outer(xRx2,xend,"<=")
  ov[MATCH]<-outer(xRx2-xRx1,zero,"+")[MATCH]

  ov
}
#2. Getting the log posterior probability
```
⟨*Alogpost2*⟩
⟨*trawls.q end*⟩

4.1 Calculating the log posterior probability

This returns a list with the values of x, y,, R_x, R_y and an array with the log-posterior probability values for different combinations of the arguments.

⟨*Alogpost2*⟩≡

```
Alogpost2<-function(minx,maxx,miny,maxy,minRx,maxRx,
                    minRy,maxRy,obs,datmat,m,sig=0.1){
    xs<-datmat[,1]
    ys<-datmat[,2]
    xe<-datmat[,3]
    x<-seq(from=minx, to=maxx, length=m)
    y<-seq(from=miny, to=maxy, length=m)
    Rx<-seq(from=minRx, to=maxRx, length=m)
    Ry<-seq(from=minRy, to=maxRy, length=m)

    theor<-overlap5(x,y,Rx,Ry,xs,ys,xe)
    n<-length(obs)
    const <- ( - n/2) * log(2 * pi * sig^2)
    xR0 <- array(0, dim(theor)[c(1,2,3,4)])
    Aobs <- outer(xR0, obs, "+")
    d2 <- -0.5 * ((Aobs - theor)/sig)^2
    chi <- apply(d2, c(1,2,3,4), sum)
    likmat <- chi + const
    #likmat <- likmat - max(likmat)
    list(x = x, y = y, Rx = Rx, Ry = Ry, likmat = likmat)
  }


###3.  Generating catches
genobs2<-function(x,y,Rx,Ry,xstart,ystart,xend,sig=0.1)
  {
    n<-length(xstart)
    E<-rnorm(n,0,sig)
    catch<-as.vector(overlap5(x,y,Rx,Ry,xstart,ystart,xend))
    obs<-catch + E
    obs
  }

### two rectangular schools
```

⟨*twocatch*⟩

4.2 Two Rectangular Schools

`twocatch` returns the artificial catches for trawls with two rectangular schools centred at points $x1, y1$ and $x2, y2$.

$\langle$*twocatch*$\rangle\equiv$

```
twocatch<-function(x1,y1,Rx1,Ry1,x2,y2,Rx2,Ry2,
                   xstart,ystart,xend,sig=0.1){
    n<-length(xstart)
    E<-rnorm(n,0,sig)
    catch1<-as.vector(overlap5(x1,y1,Rx1,Ry1,xstart,ystart,xend))
    catch2<-as.vector(overlap5(x2,y2,Rx2,Ry2,xstart,ystart,xend))
    totalcatch<-catch1+catch2
    obscatch<-totalcatch+E

    obscatch
  }
```

5 Pictures for the two-dimensional case

These routines produce figures for the report.

5.1 One-dimensional, one school

This draws a picture of a single rectangle with an overlapping set of trawls.

$\langle$*oned*$\rangle\equiv$

```
oned<-function(x,R,trawls,xlim=c(-5.5,5.5),
               ylim=c(-6.5,6.5),number="trl001"){
    x1_x-R
    x2_x+R
    y1<-1
    y2<--1
    plot(c(x1,x2,x2,x1,x1),c(y1,y1,y2,y2,y1),axes=F,
         type="l",xlim=xlim,ylim=ylim,xlab="",ylab="")
    lines(c(x1,xlim[1]),c(0,0))
    lines(c(x2,xlim[2]),c(0,0))
    text(x,-1.5,"x",cex=1)
    text(x+R,-1.5,"x+R",cex=1)
    text(x-R,-1.5,"x-R",cex=1)
    text(xlim[1],ylim[1],number,cex=0.5)
    ss<-trawls[,1]
    tt<-trawls[,2]
    yc <- seq(from = -6, to = 6, length = length(ss))
    for(i in 1:length(ss))  {
      lines(c(ss[i],tt[i]),c(yc[i], yc[i]))
    }
  }
pictrawls3<-maketrawls2(-5,3,30,2)
postscript(file="figs/trl001",width=6.0, height=6.0)
oned(0,1,pictrawls3)
dev.off()
```

5.2 One-dimensional, 2 Schools

This constructs a figure for the two-school, one-dimensional case.

⟨*oned2*⟩≡

```
oned2<-function(x1,R1,x2,R2,trawls,xlim=c(-4.5,7.5),
                ylim=c(-6.5,6.5),number="trl002"){
    BELOW<--1.5
    x1a_x1-R1
    x2a_x1+R1
    y1a_1
    y2a_-1
    x1b_x2-R2
    x2b_x2+R2
    y1b_1
    y2b_-1
    plot(c(x1a,x2a,x2a,x1a,x1a),c(y1a,y1a,y2a,y2a,y1a),
         axes=F,type="l",xlim=xlim+c(-1,+1),ylim=ylim,
         xlab="",ylab="")
    lines(c(x1b,x2b,x2b,x1b,x1b),c(y1b,y1b,y2b,y2b,y1b))
    lines(c(x1a,xlim[1]),c(0,0))
    lines(c(x2a,x1b),c(0,0))
    lines(c(x2b,xlim[2]+1),c(0,0))
    ## text(x1,BELOW,"x1",cex=1)
    text(x1+R1,BELOW,"x1+R1",cex=1)
    text(x1-R1,BELOW,"x1-R1",cex=1)
    ##  text(x2,BELOW,"x2",cex=1)
    text(x2+R2,BELOW,"x2+R2",cex=1)
    text(x2-R2,BELOW,"x2-R2",cex=1)
    text(xlim[1],ylim[1],number,cex=0.5)
    ss<-trawls[,1]
    tt<-trawls[,2]
    yc <- seq(from = -6, to = 6, length = length(ss))
    for(i in 1:length(ss))  {
      lines(c(ss[i],tt[i]),c(yc[i], yc[i]))
    }
  }
### execution code follows
pictrawls4<-maketrawls2(-4,7,30,2)
postscript(file="figs/trl002.ps",width=6.0, height=6.0)
oned2(0,1,5,1,pictrawls4)
dev.off()
```

5.3 Circles

These figures illustrate the two-dimensional case with one or two circles. in the first case, one circle is drawn with center a, b and radius, r. The set of *trawls* is a matrix as generated by `hortrawls`.

⟨*cirschool*⟩≡

```
#Given the centre and the radius draw the circular school and the trawls
#Let (a, b) denote the centre and r the radius.

circschool<-function(a,b,r,trawls) {
    #Draw the school
    x<-seq(-r-4,r+4,0.01)

    ypos<-b+sqrt((r^2)-((x-a)^2))
    yneg<-b-sqrt((r^2)-((x-a)^2))

    plot(x,ypos,type="l",ylim=c(-r-9,r+9),xlim=c(-r-9,r+9),
         xlab="x",ylab="y",main="",cex=0.8)
    lines(x,yneg)
      #Draw the trawls
        xs <- trawls[, 1]
        ys <- trawls[, 2]
        xf <- trawls[, 3]
        yf <- trawls[, 2]
        for(i in 1:length(xs)) {
                lines(c(xs[i], xf[i]), c(ys[i], yf[i]),lty=2)
                #text(locator(1),paste("trawl   ",  i),cex=1)
        }
  }
```

cirschool2 draws a couple of circles and the trawls.

⟨*cirschool2*⟩≡

```
circschool2<-function(a1,b1,r1,a2,b2,r2,trawls)  {
   #Draw the school
   x<-seq(-max(r1,r2)-4,max(r1,r2)+4,0.01)

   y1pos<-b1+sqrt((r1^2)-((x-a1)^2))
   y1neg<-b1-sqrt((r1^2)-((x-a1)^2))

   y2pos<-b2+sqrt((r2^2)-((x-a2)^2))
   y2neg<-b2-sqrt((r2^2)-((x-a2)^2))


   plot(x,y1pos,type="l",ylim=c(-15,15),xlim=c(-15,15),
        xlab="x",ylab="y",main="",cex=0.8)
   lines(x,y1neg)
   lines(x,y2pos)
   lines(x,y2neg)


    #Draw the trawls

       xs <- trawls[, 1]
       ys <- trawls[, 2]
       xf <- trawls[, 3]
       yf <- trawls[, 2]
       for(i in 1:length(xs)) {
               lines(c(xs[i], xf[i]), c(ys[i], yf[i]),lty=2)
             # text(locator(1),paste("trawl   ",  i),cex=1)
       }
 }
### some execution code
pictrawls1<-hortrawls(-6,6,-6,6,50,4)
circschool(0,0,3,pictrawls1)
circschool2(0,0,2,3,4,2,pictrawls1)
```

5.4 Rectangles

`drawpic` draws a rectangle centered at x, y with half-widths R_x, R_y and a set of trawls on top of it.

⟨*drawpic*⟩≡

```
drawpic<-function(x,y,Rx,Ry,trawls)  {
     x1<-x-Rx
     y1<-y+Ry
     x2<-x+Rx
     y2<-y-Ry

     #Draw the rectangle
     plot(c(x1,x2,x2,x1,x1),c(y1,y1,y2,y2,y1),type="l",
          xlim=c(-8,8),ylim=c(-7,7),main="",
          xlab="x",ylab="y",cex=0.8)

     #Draw the trawls
     xs<-trawls[,1]
     ys<-trawls[,2]
     xf<-trawls[,3]
     yf<-trawls[,2]
     for(i in 1:length(xs)) {
         lines(c(xs[i],xf[i]),c(ys[i],yf[i]))
       }
   }
### execution code
pictrawls2<-hortrawls(-3,3,-4,4,40,2)
drawpic(0,0,1,2,pictrawls2)
```

5.5 Two Rectangles

This function produces a picture of two rectangles and a set of trawls that cover them.

⟨*drawpic2*⟩≡

```
###Two rectangular schools
drawpic2<-function(x1,y1,Rx1,Ry1,x2,y2,Rx2,Ry2,trawls)  {
     x1a<-x1-Rx1
     y1a<-y1+Ry1
     x2a<-x1+Rx1
     y2a<-y1-Ry1

     x1b<-x2-Rx2
     y1b<-y2+Ry2
     x2b<-x2+Rx2
     y2b<-y2-Ry2

     #Draw the square
     plot(c(x1a,x2a,x2a,x1a,x1a),c(y1a,y1a,y2a,y2a,y1a),
           type="l",xlim=c(-8,8),ylim=c(-7,7),
           main="",xlab="x",ylab="y",cex=0.8)
     lines(c(x1b,x2b,x2b,x1b,x1b),c(y1b,y1b,y2b,y2b,y1b))

     #Draw the trawls
     xs<-trawls[,1]
     ys<-trawls[,2]
     xf<-trawls[,3]
     yf<-trawls[,2]

     for(i in 1:length(xs)) {
         lines(c(xs[i],xf[i]),c(ys[i],yf[i]))
       }
   }
pictrawls2a<-hortrawls(-5,5,-6,6,40,2)
drawpic2(0,0,1,2,3,3,1,2,pictrawls2a)
```

5.6 Picture generation routines collected together

⟨*pictures*⟩≡
  ⟨*do not edit note*⟩
  ⟨*cirschool*⟩
  ⟨*cirschool2*⟩
  ⟨*oned*⟩
  ⟨*oned2*⟩
  ⟨*drawpic*⟩
  ⟨*drawpic2*⟩
  ⟨*trawls.q end*⟩

A Additional code to be inserted

⟨*do not edit note*⟩≡
```
### -----------------------------------------
### NOTE: DO NOT edit this file!
###       This program is generated
###       automatically by NOWEB from trawls.nw
###       $Author: vignaux $
###       $Revision: 1.9 $
###       $Date: 1999/05/26 03:36:25 $
### ----------------------------------------
```

⟨*heading material*⟩≡
```
### ----------------------------------------
### trawls.q
### gav $Revision: 1.9 $
### ---------------------------------------
```

⟨*trawls.q end*⟩≡
```
### Local variables:
### mode: S
### mode: outline-minor
### outline-regexp: "\^L\\|###\\*\\|###\\*\\*\\|###\\*\\*\\*"
### End:
```

References

Paul, S. and Vignaux, G. A. (1999). Estimating fish concentration and location using trawl data. Technical Report 99a, MCS, Victoria University, Wellington, NZ.