HW11

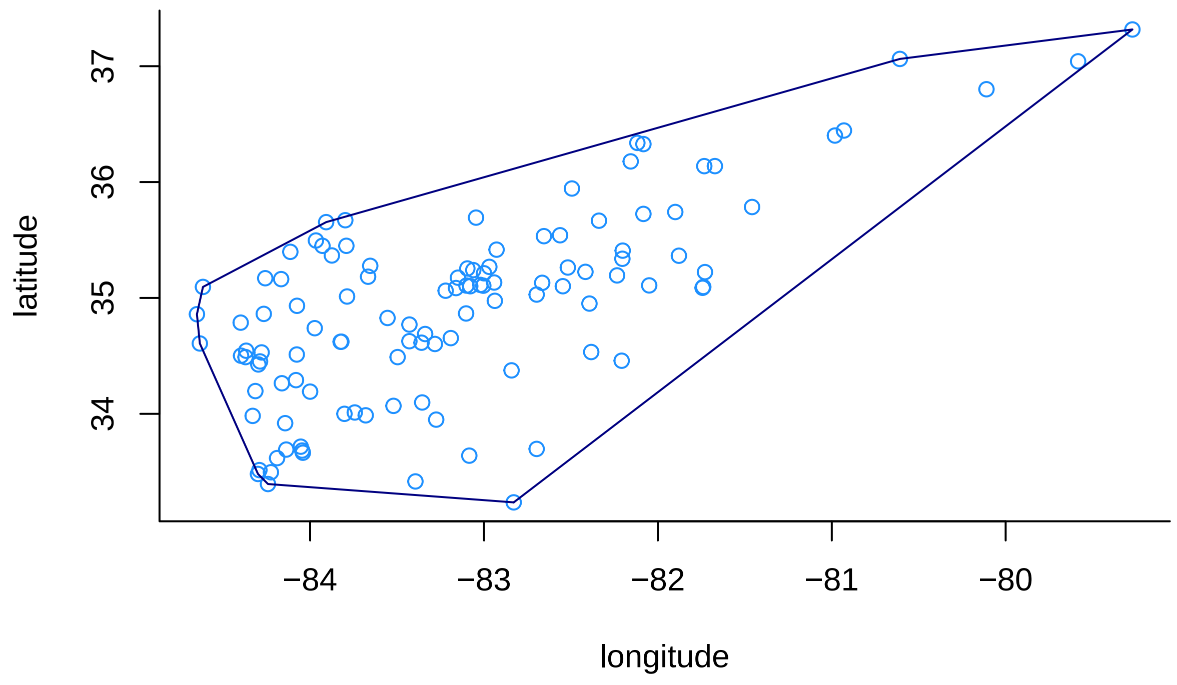
ASG

2023-05-03

5.4

a

```
library(gss) ; data(LakeAcidity)
x1 <- LakeAcidity$lon
x2 <- LakeAcidity$lat
chullInds <- chull(x1,x2)
chullInds <- c(chullInds,chullInds[1])
LakeAcidityConvexHullBdry <- cbind(x1,x2)[chullInds,]
plot(x1,x2,bty="l",xlab = "longitude",ylab = "latitude",col = "dodgerblue")
lines(LakeAcidityConvexHullBdry,col = "navy")
```



b

```r
#colnames(LakeAcidity)
library(mgcv)
```

```
## Loading required package: nlme

## This is mgcv 1.8-42. For overview type 'help("mgcv-package")'.
```

```r
fittp<-gam(log(cal)~s(x1,x2,bs='tp',k=60),data=LakeAcidity,method='REML',)
summary(fittp)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(cal) ~ s(x1, x2, bs = "tp", k = 60)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.68280    0.04933   13.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df     F p-value
## s(x1,x2) 27.69  36.82 4.278  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.59   Deviance explained = 69.2%
## -REML = 118.09  Scale est. = 0.27254   n = 112
```

```r
fitts<-gam(log(cal)~te(x1,x2),data=LakeAcidity,method='REML')
summary(fitts)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(cal) ~ te(x1, x2)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.68280    0.06054   11.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df   F  p-value
## te(x1,x2) 11.01  13.14 5.4 5.05e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.382   Deviance explained = 44.4%
## -REML = 116.79  Scale est. = 0.41046   n = 112
```

c

```
library(fields)
```

```
## Loading required package: spam

## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

```
library(HRW)
ngrid <- 201
lonGrid <- seq(min(x1),
           max(x1),length = ngrid)
latGrid <- seq(min(x2),max(x2),
               length = ngrid)
lonlatMesh <- expand.grid(lonGrid,latGrid)
names(lonlatMesh) <- c("longitude","latitude")
outInds <- (1:ngrid^2)[pointsInPoly(lonlatMesh,
                        LakeAcidityConvexHullBdry)==FALSE]


# Obtain the fitted surface over the mesh:

fitMeshtp <- matrix(predict(fittp,
                  newdata = lonlatMesh),ngrid,ngrid)
```

```
## Warning in predict.gam(fittp, newdata = lonlatMesh): not all required variables have been supplied i
## Warning: 'newdata' had 40401 rows but variables found have 112 rows
```

```
## Warning in matrix(predict(fittp, newdata = lonlatMesh), ngrid, ngrid): data
## length [112] is not a sub-multiple or multiple of the number of rows [201]
```

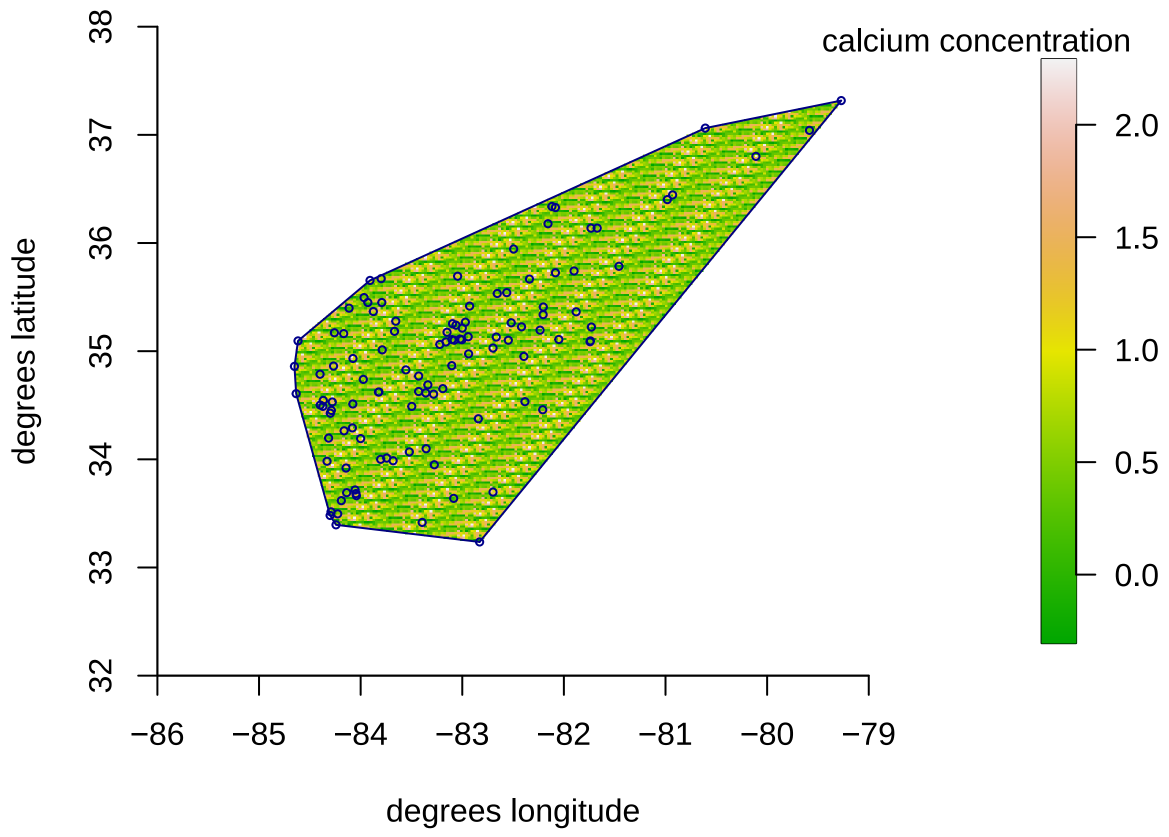
```
fitMeshts <- matrix(predict(fitts,
                  newdata = lonlatMesh),ngrid,ngrid)
```

```
## Warning in predict.gam(fitts, newdata = lonlatMesh): not all required variables have been supplied i
```

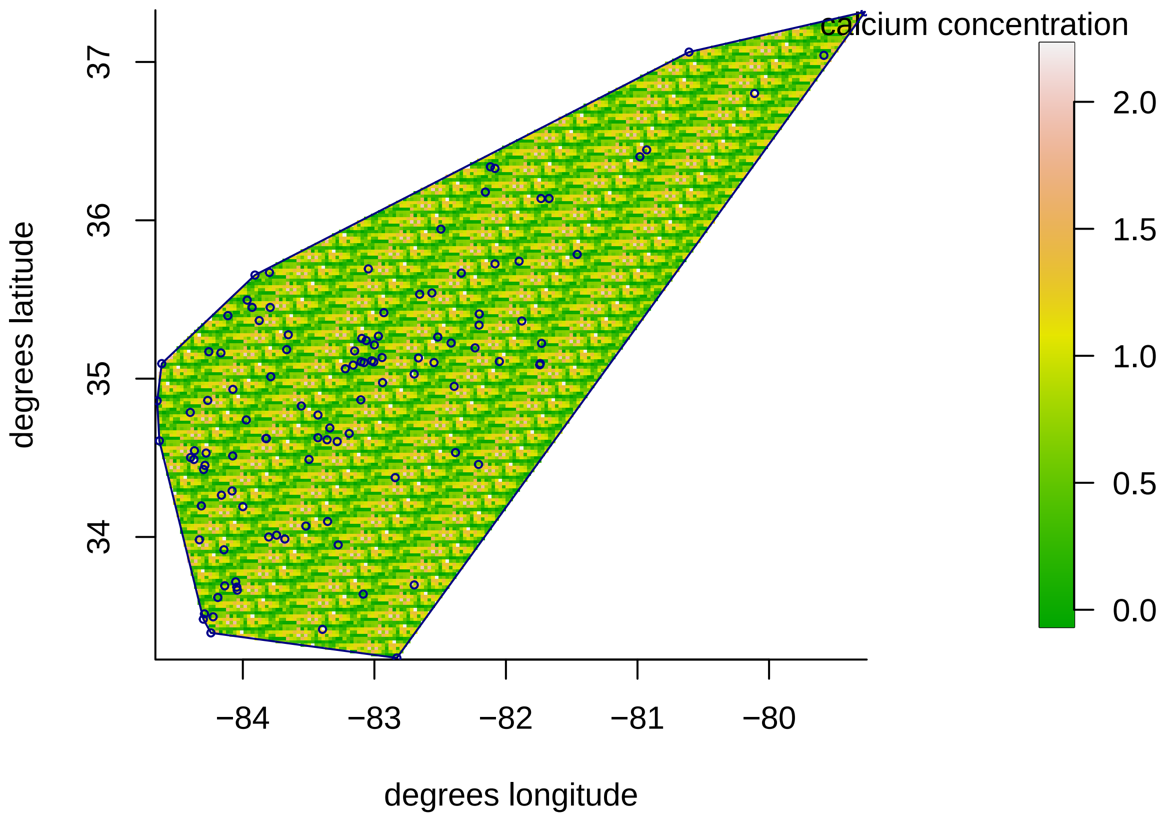
```
## Warning: 'newdata' had 40401 rows but variables found have 112 rows
```

```
## Warning in matrix(predict(fitts, newdata = lonlatMesh), ngrid, ngrid): data
## length [112] is not a sub-multiple or multiple of the number of rows [201]
```

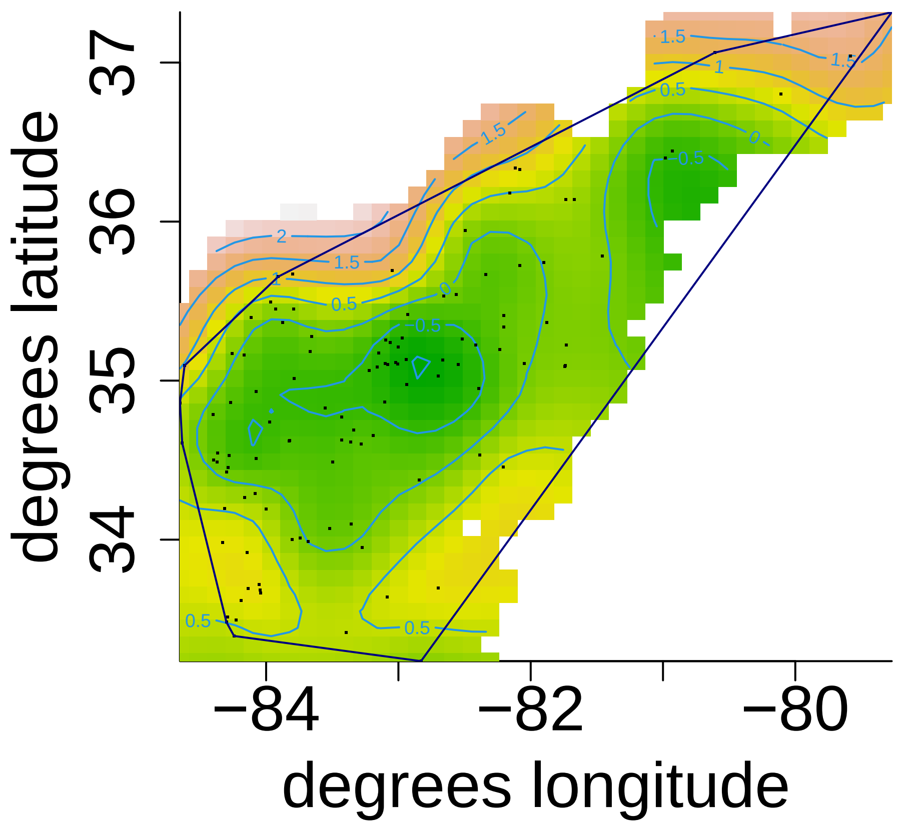
```
fitMeshtp[outInds] <- NA
fitMeshts[outInds] <- NA

par(mai = c(1.02,0.95,0.1,1.2))
library(fields)
image.plot(lonGrid,latGrid,fitMeshtp,col = terrain.colors(1000),
      xlab = "degrees longitude",ylab = "degrees latitude",
      legend.args = list(text = "calcium concentration",
      cex = 1,adj = 0.8),axis.args = list(cex.axis = 1),bty = "l",
      ylim=c(32,38),
      xlim=c(-86,-79),
      cex.lab = 1,cex.axis = 1)
lines(LakeAcidityConvexHullBdry,col = "navy")
points(x1,x2,
       col = "darkblue",cex = 0.5)
```



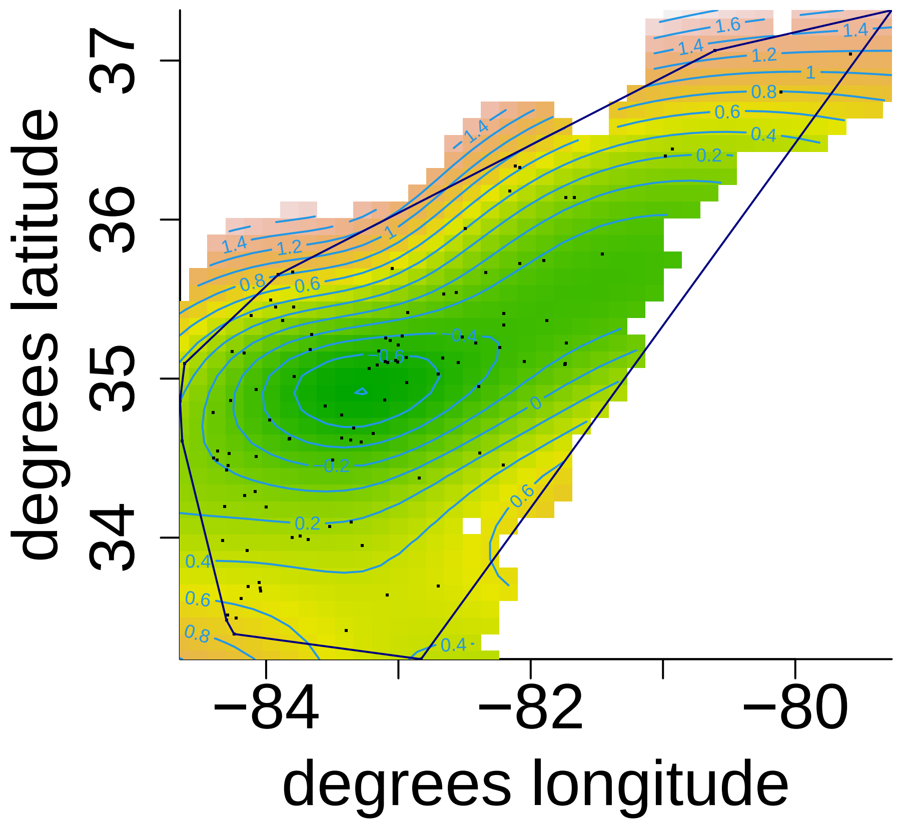
```
par(mai = c(1.02,0.95,0.1,1.2))
image.plot(lonGrid,latGrid,fitMeshts,col = terrain.colors(1000),
      xlab = "degrees longitude",ylab = "degrees latitude",
      legend.args = list(text = "calcium concentration",
      cex = 1,adj = 0.8),axis.args = list(cex.axis = 1),bty = "l",
      cex.lab = 1,cex.axis =1)
lines(LakeAcidityConvexHullBdry,col = "navy")
points(x1,x2,
       col = "darkblue",cex = 0.5)
```



```
plot(fittp,scheme = 2,hcolors = terrain.colors(1000),main="",bty="l", cex.lab = 2,cex.axis = 2,xlab = "d
lines(LakeAcidityConvexHullBdry,col = "navy")
```



```
plot(fitts,scheme = 2,hcolors = terrain.colors(1000),main="",bty="l", cex.lab = 2,cex.axis = 2,xlab = "d
lines(LakeAcidityConvexHullBdry,col = "navy")
```



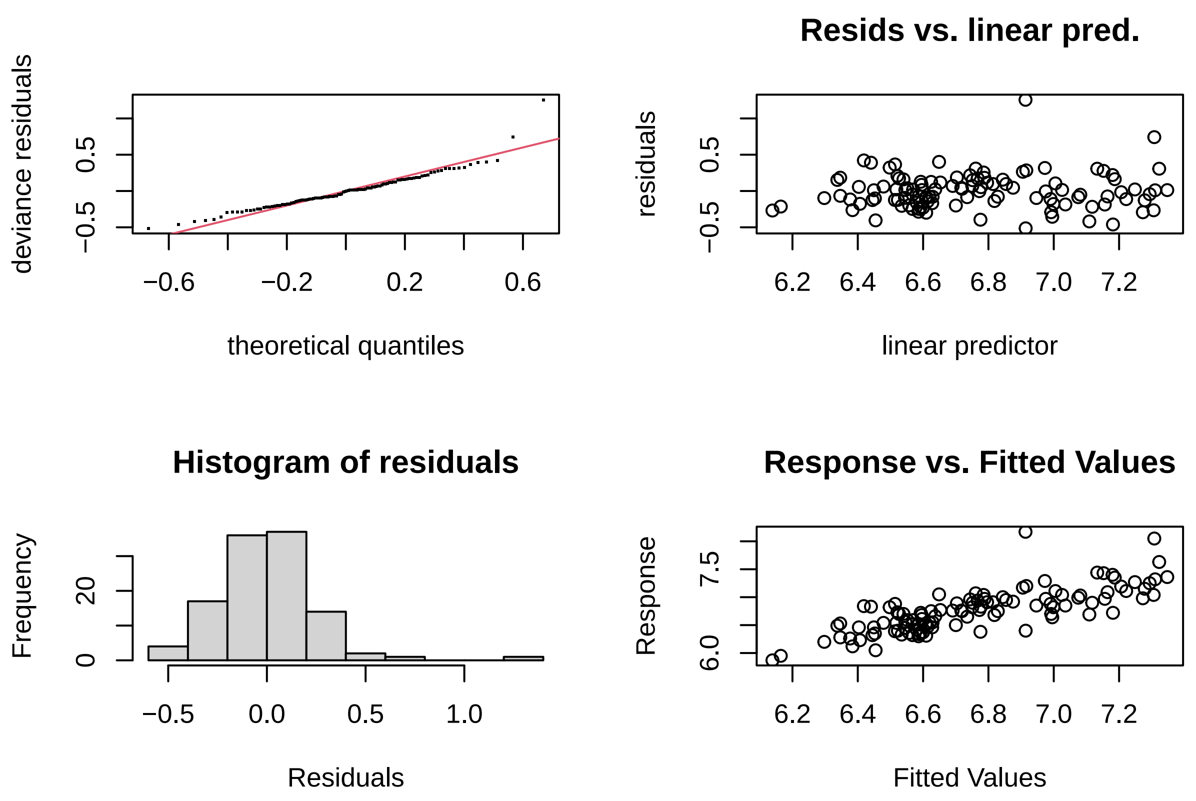
As from the geo plots , the highest mean concentration of log calcium concentrate is at the northern shore of the lake. The contours at that location is reddish hue which corresponds to the highest contour line of calcium concentrate i.e 1.4/2

d

the two bivariate penalized spline approach is using thin plate splines and using tensor product. If we look at the graph plot from the previous questions, ther tensor product fit is much more smoother/complicated as it generates more contours in the same area than thin plate splines. Although the highest level of log conc of calcium differs in both the fit with 2 and 1.4 respectively. From the summary we can see that the thin splate explains more variability in data with r square as 0.59 compared to the r square of 0.38 of tensor product

e

```
fitgeo<-gam(ph ~ s(x1,x2,bs='tp',k=60)+s(log(cal),k=15),data=LakeAcidity,method='REML')
gam.check(fitgeo)
```



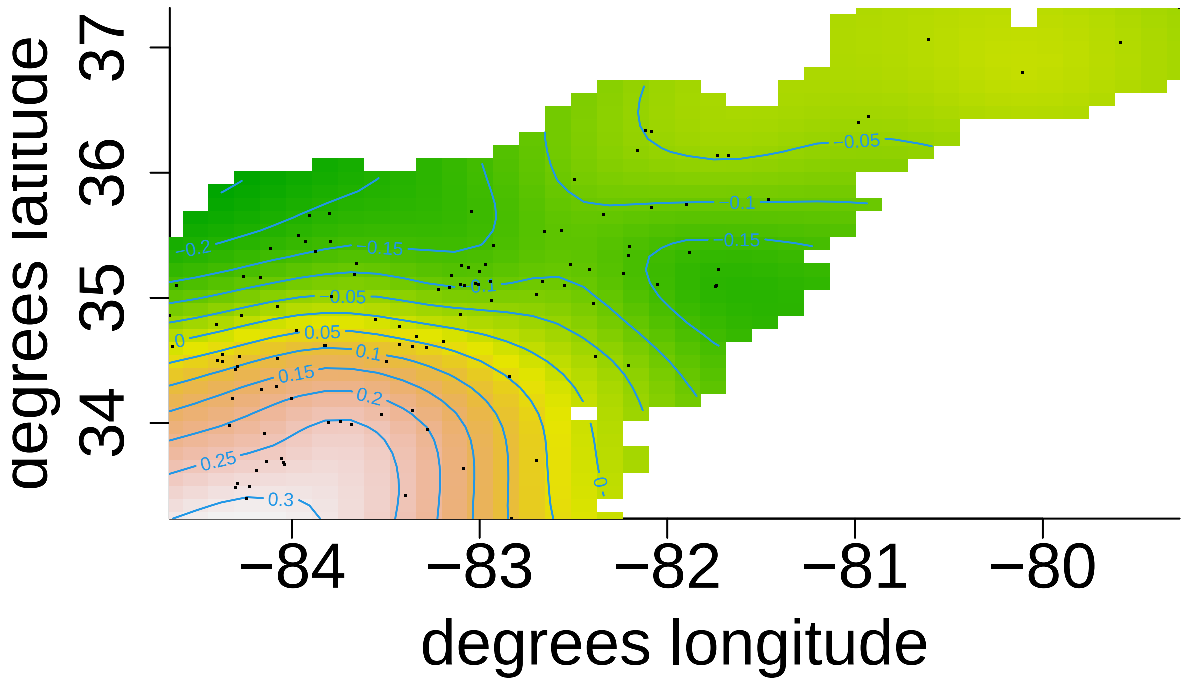
```
##
## Method: REML   Optimizer: outer newton
## full convergence after 9 iterations.
## Gradient range [-5.595976e-06,1.249086e-05]
## (score 20.54524 & scale 0.06543251).
## Hessian positive definite, eigenvalue range [5.596102e-06,54.22925].
## Model rank =  74 / 74
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                  k'   edf k-index p-value
## s(x1,x2)     59.00  8.95    1.12    0.85
## s(log(cal)) 14.00  1.00    1.00    0.45
```

```
summary(fitgeo)
```

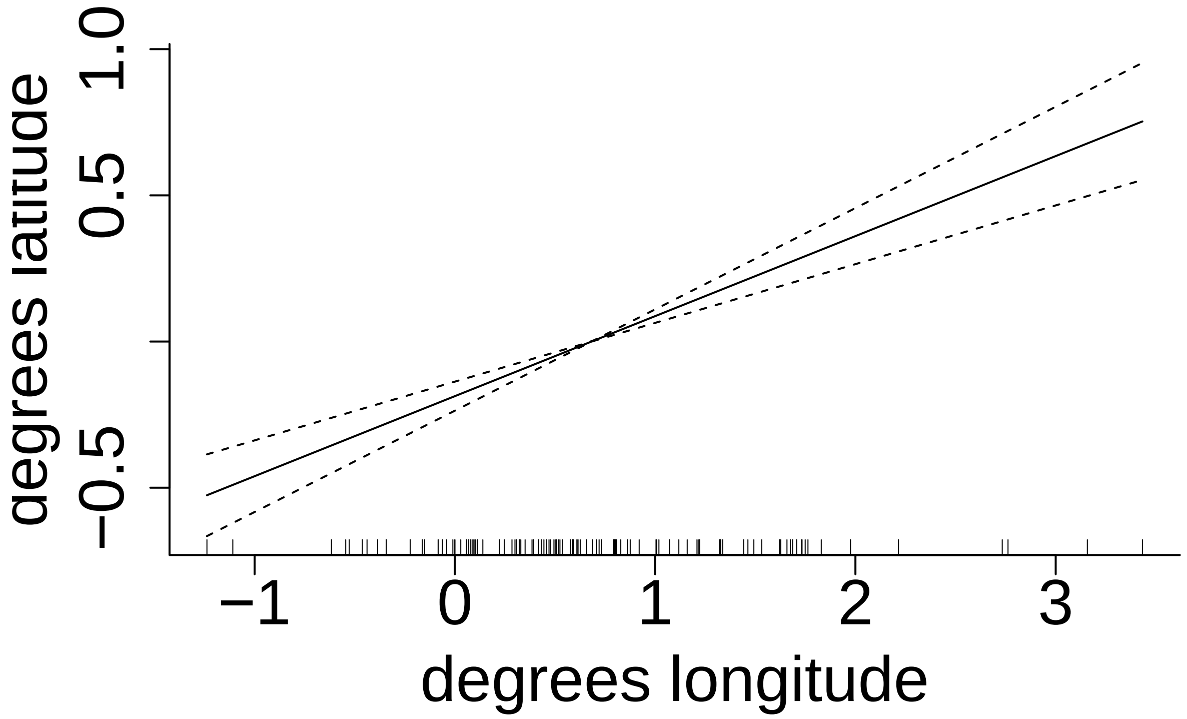
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## ph ~ s(x1, x2, bs = "tp", k = 60) + s(log(cal), k = 15)
##
## Parametric coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.76018    0.02417   279.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F  p-value
## s(x1,x2)      8.953  12.62  3.09 0.000804 ***
## s(log(cal)) 1.000   1.00 56.58  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.568   Deviance explained = 60.7%
## -REML = 20.545  Scale est. = 0.065433  n = 112
```

```
plot(fitgeo,scheme = 2,hcolors = terrain.colors(1000),main="",bty="l", cex.lab = 2,cex.axis = 2,xlab =
```



```
lines(LakeAcidityConvexHullBdry,col = "navy")
```

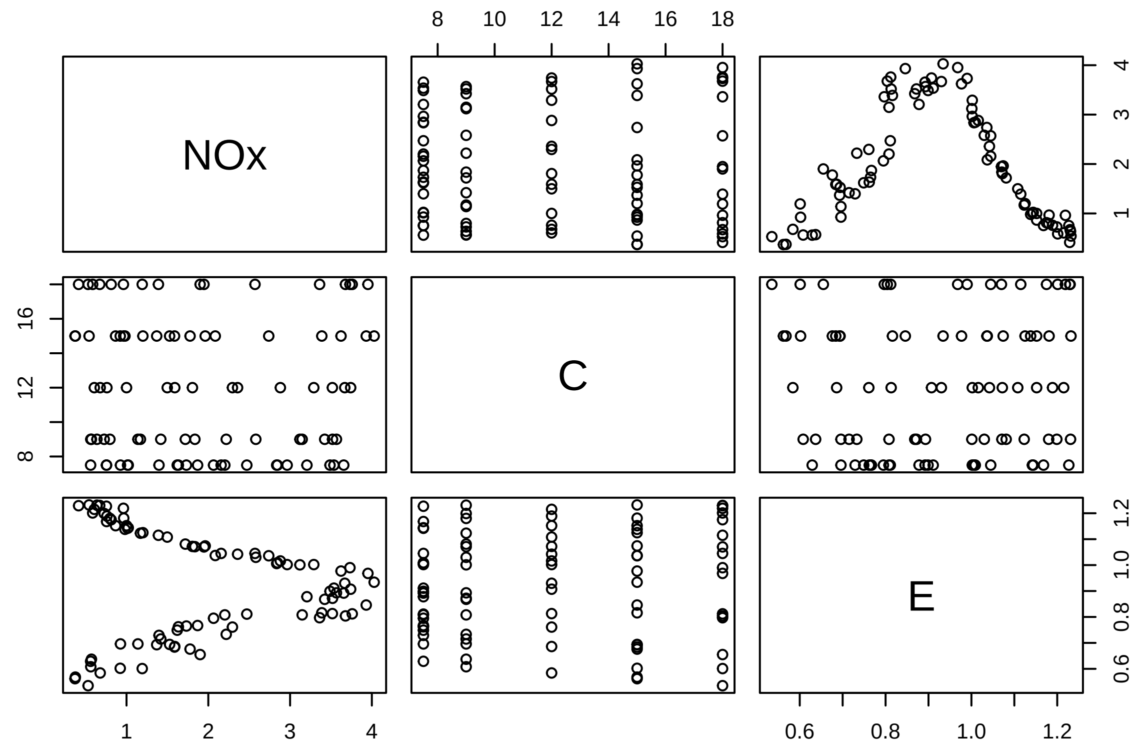


### f As from the summary of geoadditive fit, we can see that the log calcium concentration has edf of 1 in this fit , hence we can only assume that it has a linear and simple relationship with the ph of the lake. ALso since the F-test is significant , thus the log calcium is much more important. we can say that the as log calcium concentrtion increases so does the PH. ### g THe edf of the location from lat and long gives us as 8.9 which gives us a complicated relationship between the lat/long with the mean ph of the lake. Even this bivariate term is significant in this geoadditive model but the relationship is much more complicated

5.8

a

```
library(lattice) ; data(ethanol) ; pairs(ethanol)
```

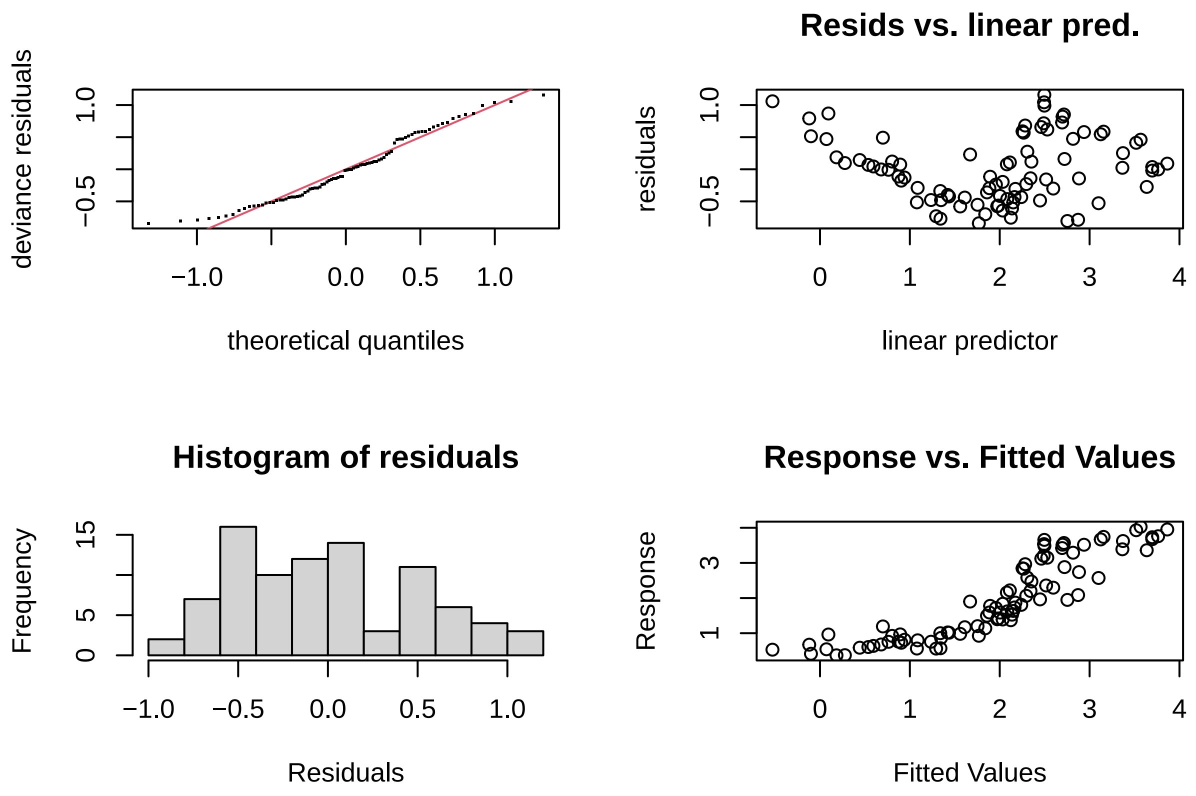


```
colnames(ethanol)
```

```
## [1] "NOx" "C"   "E"
```

b

```
fitethanol<-gam(NOx~s(C,k=3)+s(E,by=C,k=3),data=ethanol,method='REML')
gam.check(fitethanol)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 12 iterations.
## Gradient range [-2.981227e-05,4.887592e-05]
## (score 80.10966 & scale 0.2741214).
## Hessian positive definite, eigenvalue range [0.01414535,42.00611].
## Model rank =  5 / 6
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'  edf k-index p-value
## s(C)   2.00 1.17    1.17    0.93
## s(E):C 3.00 3.00    0.29  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

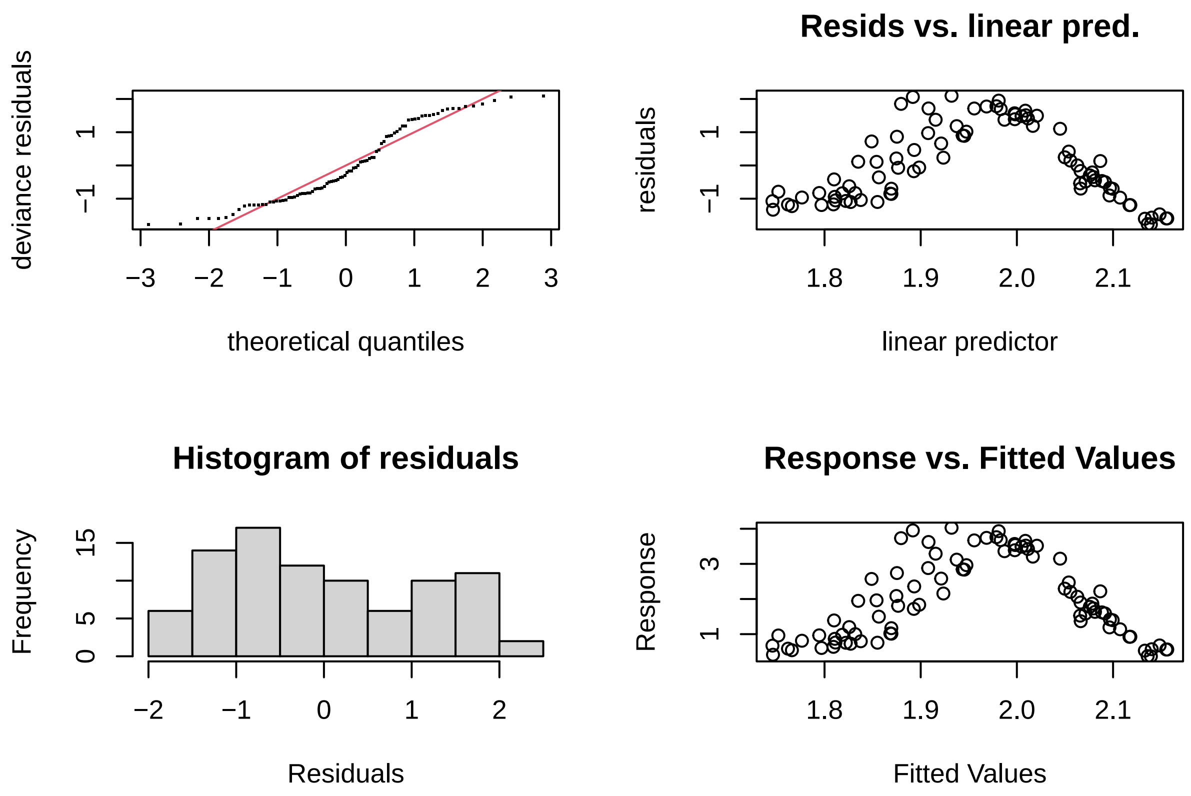
```
summary(fitethanol)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## NOx ~ s(C, k = 3) + s(E, by = C, k = 3)
```

```
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0          0     NaN      NaN
##
## Approximate significance of smooth terms:
##           edf Ref.df      F p-value
## s(C)    1.169   1.31  42.18  <2e-16 ***
## s(E):C 2.997   3.00 517.85  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 5/6
## R-sq.(adj) =  0.786   Deviance explained = 79.4%
## -REML =  80.11  Scale est. = 0.27412   n = 88
```

c

```
fitethanolbi<-gam(NOx~s(C,E,bs='tp',k=5),data=ethanol,method='REML')
gam.check(fitethanolbi)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 8 iterations.
## Gradient range [-5.095709e-05,6.542104e-07]
## (score 138.4362 & scale 1.298839).
## Hessian positive definite, eigenvalue range [5.095296e-05,42.5].
## Model rank =  5 / 5
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##        k' edf k-index p-value
## s(C,E)  4   2    0.21  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(fitethanolbi)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## NOx ~ s(C, E, bs = "tp", k = 5)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9574     0.1215   16.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##        edf Ref.df     F p-value
## s(C,E)   2      2 0.471   0.626
##
## R-sq.(adj) =  -0.0123   Deviance explained =  1.1%
## -REML = 138.44  Scale est. = 1.2988    n = 88
```

d

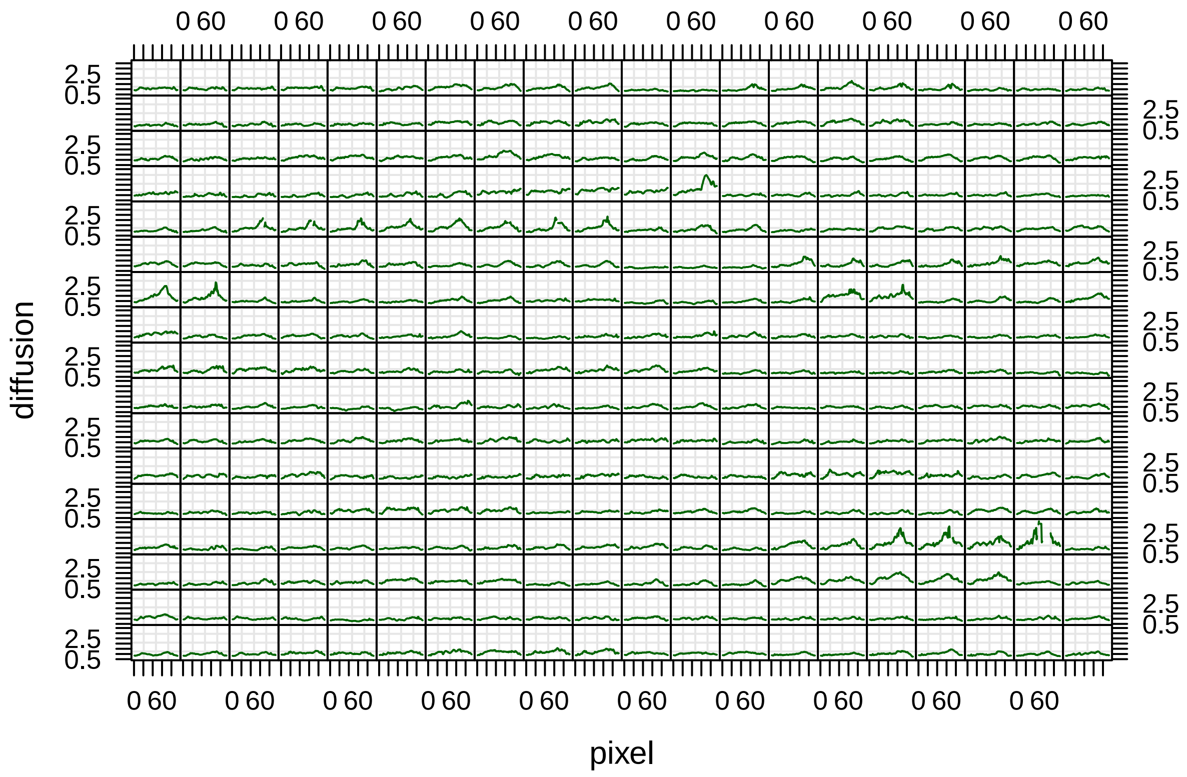
```
anova(fitethanol,fitethanolbi)
```

```
## Analysis of Deviance Table
##
## Model 1: NOx ~ s(C, k = 3) + s(E, by = C, k = 3)
## Model 2: NOx ~ s(C, E, bs = "tp", k = 5)
##   Resid. Df Resid. Dev     Df Deviance
## 1    83.547     22.981
## 2    85.000    110.401 -1.453   -87.42
```

5.10

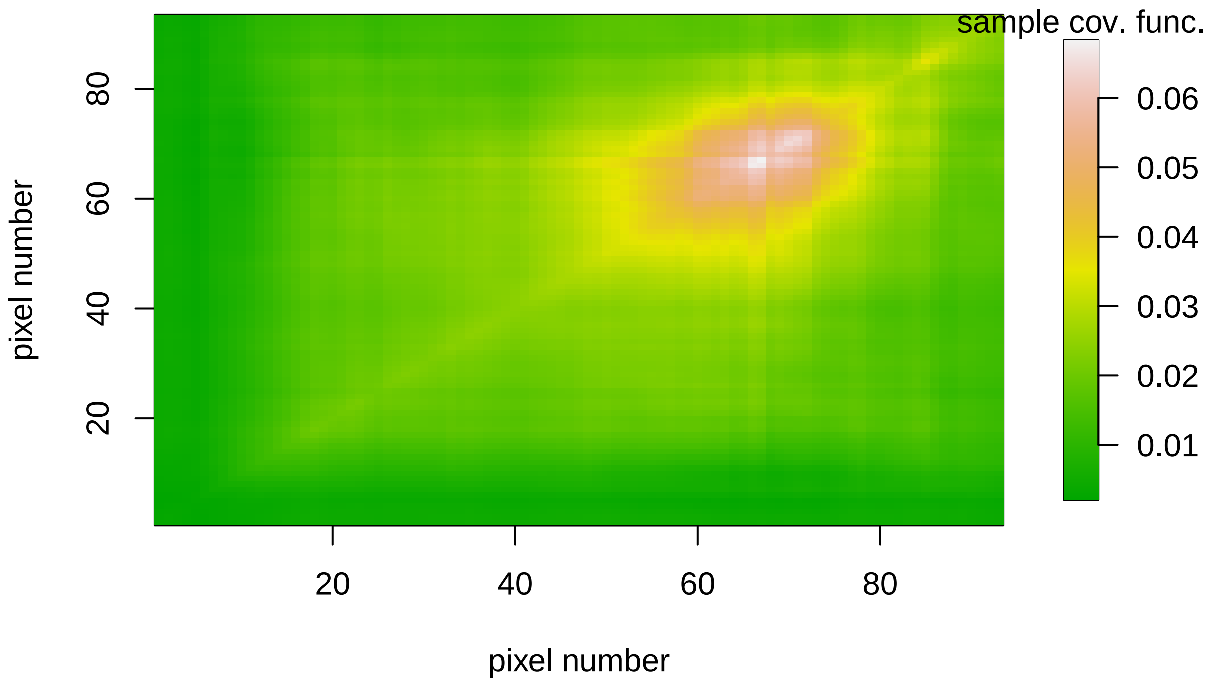
a

```
library(refund) ; data(DTI2) ; ccaLongitDF <- NULL
for (i in 1:340) ccaLongitDF <- rbind(ccaLongitDF,cbind(rep(i,93),1:93,DTI2$cca[i,]))
ccaLongitDF <- as.data.frame(ccaLongitDF)
names(ccaLongitDF) <- c("idnum","pixel", "diffusion")
library(lattice)
ccaLongitVis <- xyplot(diffusion ~ pixel|idnum,
  group = idnum,data = ccaLongitDF,
  strip = FALSE,layout = c(20,17),
  as.table = TRUE,
  panel = function(x,y,subscripts,groups)
    {
                  panel.grid()
                  panel.superpose(x,y,subscripts,groups,
                  col = "darkgreen",type = "l")
    })
print(ccaLongitVis)
```



b

```
sampCovMat <- var(DTI2$cca,na.rm = TRUE)
library(fields)
image.plot(1:93,1:93,sampCovMat,col = terrain.colors(1000),xlab = "pixel number",ylab = "pixel number",
```


sample cov. func.
0.06
0.05
0.04
0.03
0.02
0.01
pixel number
20
40
60
80
pixel number
20
40
60
80