# Kriging Acoustic Fish Density

Allison White

January 15, 2020

# Kriging Acoustic Fish Density

This is an example application of kriging to account for the autocorrelation between transect nodes in a flower survey design. Fish density (fish/m^3) estimates were derived from cone model echo-counting without any filtering to remove multiple echoes. Data was converted into an Sv variable and integrated by 1x1m cells for export.

In the .csv exported from Echoview, the variable “NASC” in the furthest right column is fish density in fish/m^3. This is calculated by converting the log-scale “Sv”" variable (Echoview does not export linear variables) to linear fish density using the equation

density = 10^(Sv/10)

Open all required packages

```
library(geoR)
```

```
## --------------------------------------------------------------
##  Analysis of Geostatistical Data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.7-5.2.1 (built on 2016-05-02) is now loaded
## --------------------------------------------------------------
```

```
library(sp)
library(RColorBrewer)
library(classInt)
library(gstat)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

First, import your data and take a look at the first few lines

```
Site<-read.csv(file.choose())
head(Site)
```

```
##   Process_ID Interval Layer Sv_mean NASCfurrealstheotheroneisdensity
## 1       2155      471     6    -999                                0
## 2       2155      471     7    -999                                0
## 3       2155      471     8    -999                                0
## 4       2155      471     9    -999                                0
## 5       2155      471    10    -999                                0
## 6       2155      471    11    -999                                0
##   Height_mean Depth_mean Layer_depth_min Layer_depth_max Ping_S Ping_E   Dist_M
## 1    0.989382   5.515807               5               6   1393   1398 470.5967
## 2    0.989382   6.505189               6               7   1393   1398 470.5967
## 3    0.989382   7.494572               7               8   1393   1398 470.5967
## 4    0.989382   8.483954               8               9   1393   1398 470.5967
## 5    1.038852   9.498071               9              10   1393   1398 470.5967
## 6    0.989382  10.512188              10              11   1393   1398 470.5967
##     Date_M         Time_M    Lat_M     Lon_M Noise_Sv_1m
## 1 20170809  03:10:34.2080 26.97833 -80.02456        -999
## 2 20170809  03:10:34.2080 26.97833 -80.02456        -999
## 3 20170809  03:10:34.2080 26.97833 -80.02456        -999
## 4 20170809  03:10:34.2080 26.97833 -80.02456        -999
## 5 20170809  03:10:34.2080 26.97833 -80.02456        -999
## 6 20170809  03:10:34.2080 26.97833 -80.02456        -999
##   Minimum_Sv_threshold_applied Maximum_Sv_threshold_applied Standard_deviation
## 1                            0                            0                  0
## 2                            0                            0                  0
## 3                            0                            0                  0
## 4                            0                            0                  0
## 5                            0                            0                  0
## 6                            0                            0                  0
##   Thickness_mean Range_mean Exclude_below_line_range_mean
## 1       0.989382   5.515807                         -9999
## 2       0.989382   6.505189                         -9999
## 3       0.989382   7.494572                         -9999
## 4       0.989382   8.483954                         -9999
## 5       1.038852   9.498071                         -9999
## 6       0.989382  10.512188                         -9999
##   Exclude_above_line_range_mean        NASC
## 1                         -9999 1.2589e-100
## 2                         -9999 1.2589e-100
## 3                         -9999 1.2589e-100
## 4                         -9999 1.2589e-100
## 5                         -9999 1.2589e-100
## 6                         -9999 1.2589e-100
```

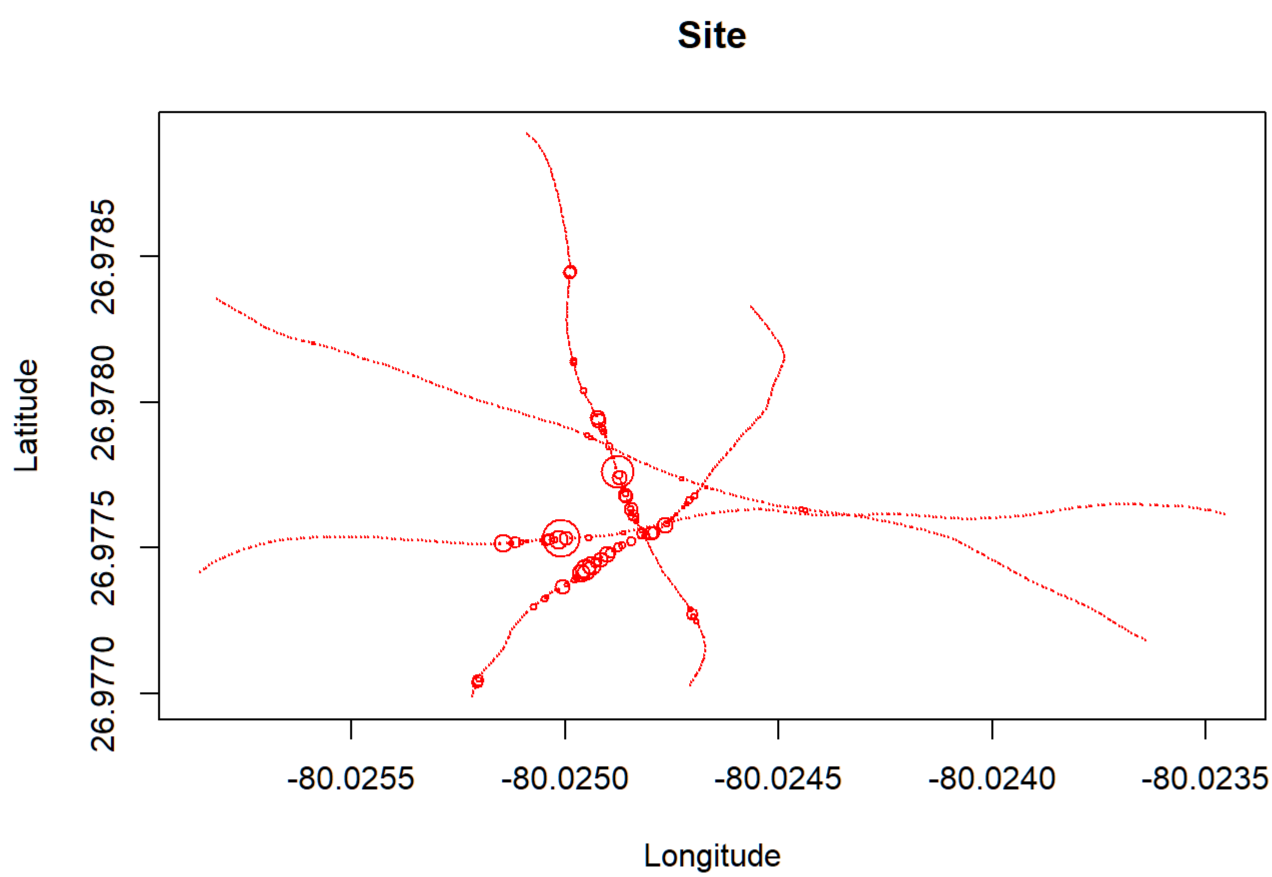
Now, take the mean density of each cell, log-transform it, and create a data frame containing Latitude, Longitude, Density, and log(Density+1)

```
fLon_M = tapply(Site$Lon_M,Site$Interval,mean)
fLat_M = tapply(Site$Lat_M,Site$Interval,mean)
fNASC = tapply(Site$NASC,Site$Interval,mean)
flog_NASC = log(fNASC+1)
Data = data.frame(fLon_M,fLat_M,fNASC,flog_NASC)
head(Data)
```

```
##        fLon_M   fLat_M       fNASC flog_NASC
## 471 -80.02456 26.97833 1.2589e-100         0
## 472 -80.02456 26.97832 1.2589e-100         0
## 473 -80.02455 26.97831 1.2589e-100         0
## 474 -80.02455 26.97830 1.2589e-100         0
## 475 -80.02455 26.97829 1.2589e-100         0
## 476 -80.02454 26.97829 1.2589e-100         0
```

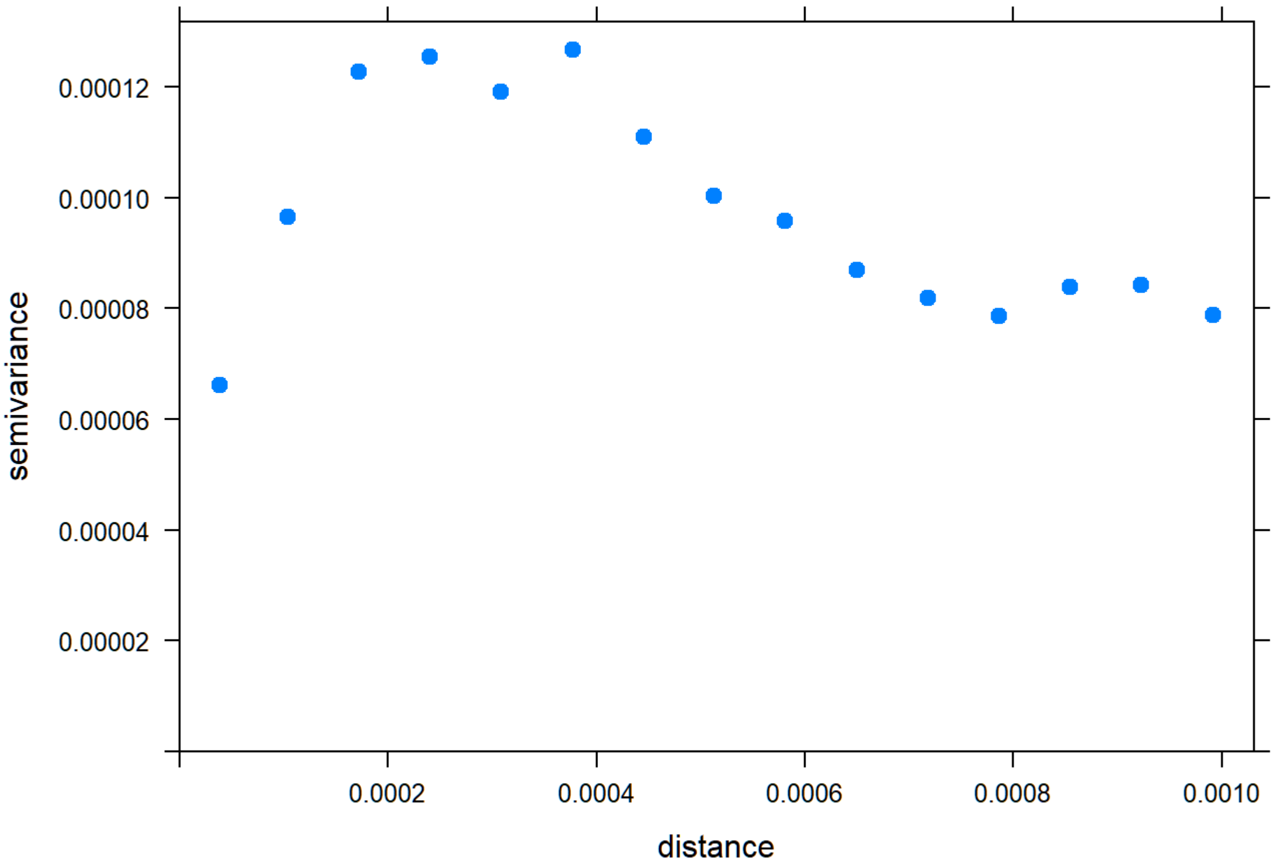
You can create a bubble plot showing your log-transformed density values per sampling location

```
plot(fLat_M~fLon_M,data=Data,type='n',xlab="Longitude",ylab="Latitude")
title("Site")
with(Data,symbols(fLon_M,fLat_M,circles=flog_NASC,inches=0.1,add=T,fg="red"))
```



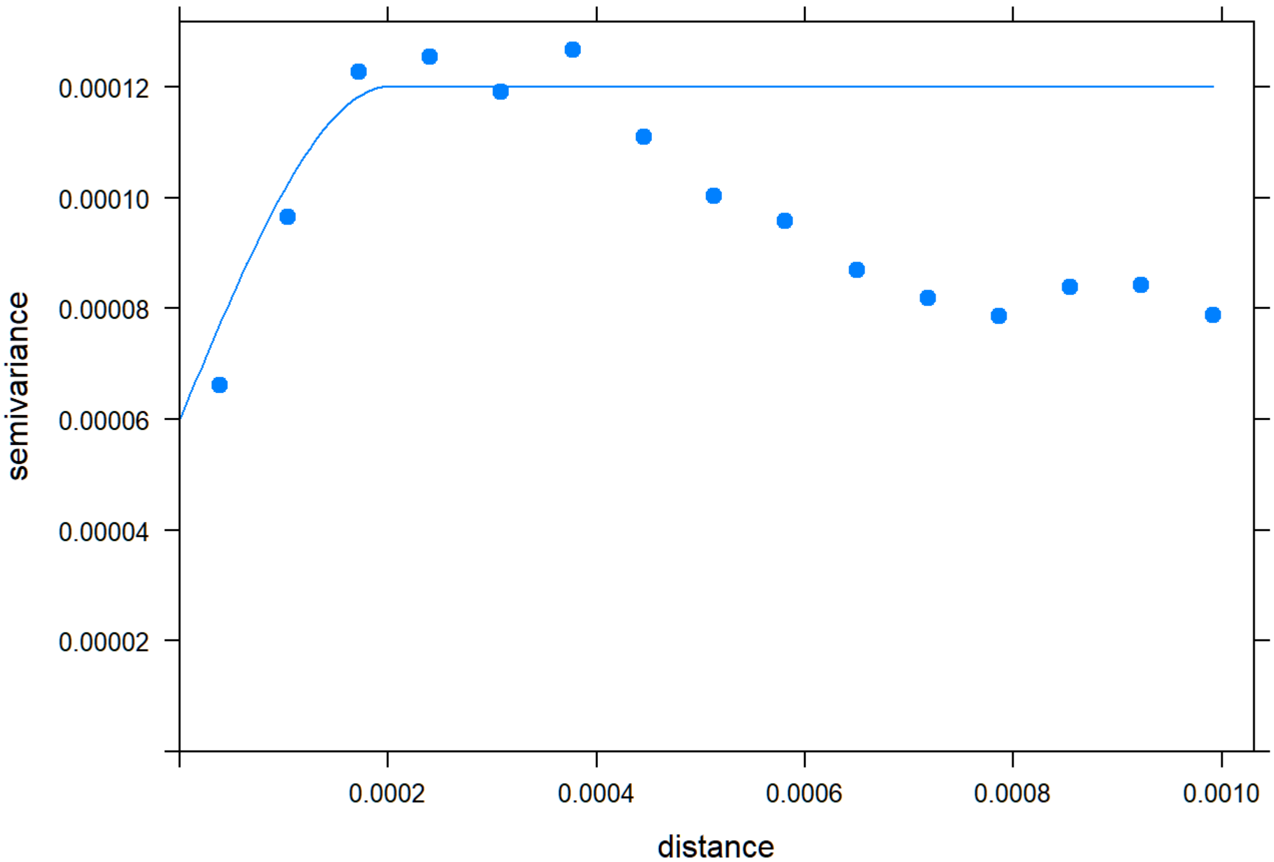

Now we'll begin kriging the data. We'll start by calculating and plotting the empirical variogram

```
Data$fXloc = Data$fLon_M
Data$fYloc = Data$fLat_M
coordinates(Data)=c("fXloc","fYloc")

fsite.vario = variogram(flog_NASC~1,Data)
plot(fsite.vario,pch=20,cex=1.5)
```




Fit the variogram model by eye and plot the results

```
fmy.psill=0.00006
fmy.range=0.0002
fmy.nugget=0.00006

fsite.eye = vgm(model="Sph",psill=fmy.psill,range=fmy.range,nugget=fmy.nugget)
plot(fsite.vario,fsite.eye,pch=20,cex=1.5)
```

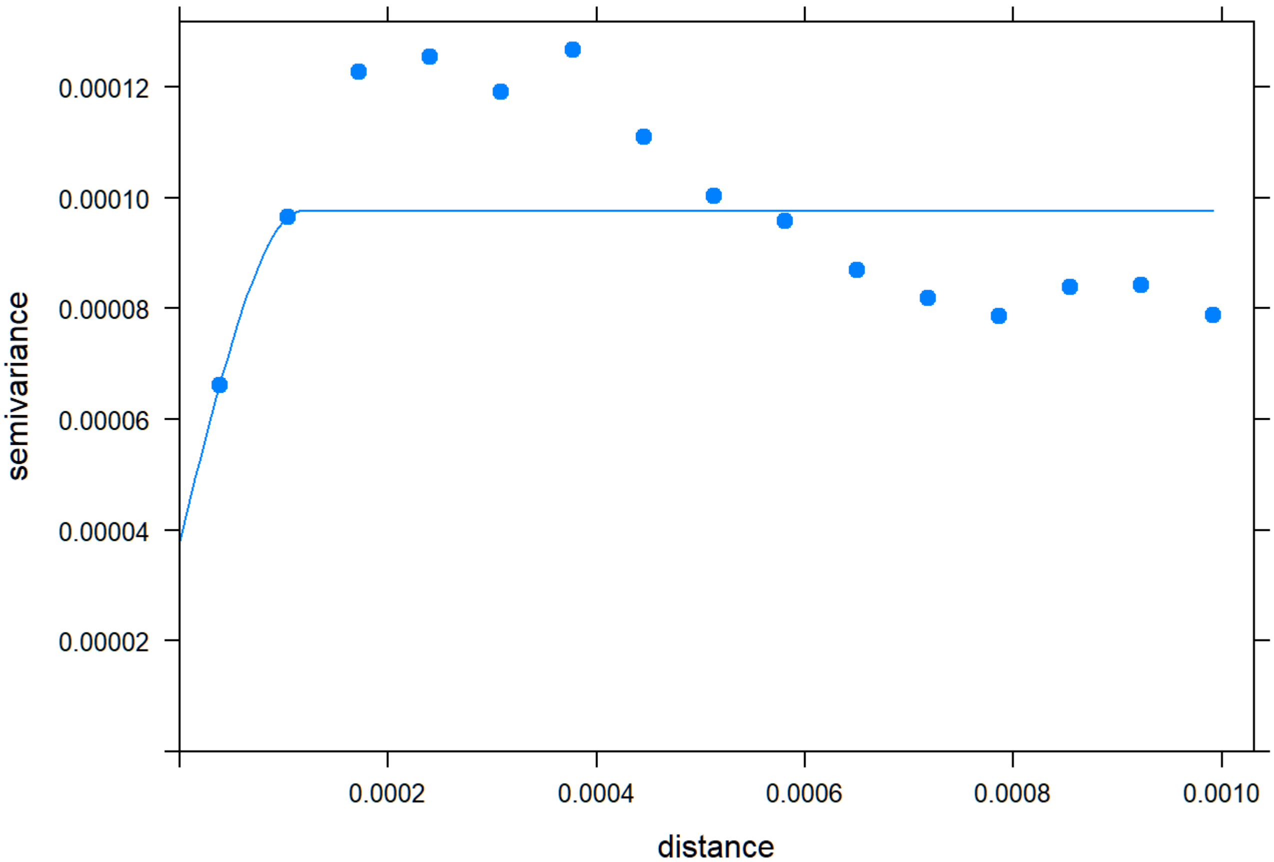



Once you've found reasonable starting parameters, fit the variogram model to your data. View the model-selected parameters and plot the results over your data

```
fsite.fit=fit.variogram(fsite.vario,
     vgm(model="Sph",psill=fmy.psill,range=fmy.range,nugget=fmy.nugget),
                         fit.method=1)

fsite.fit
```

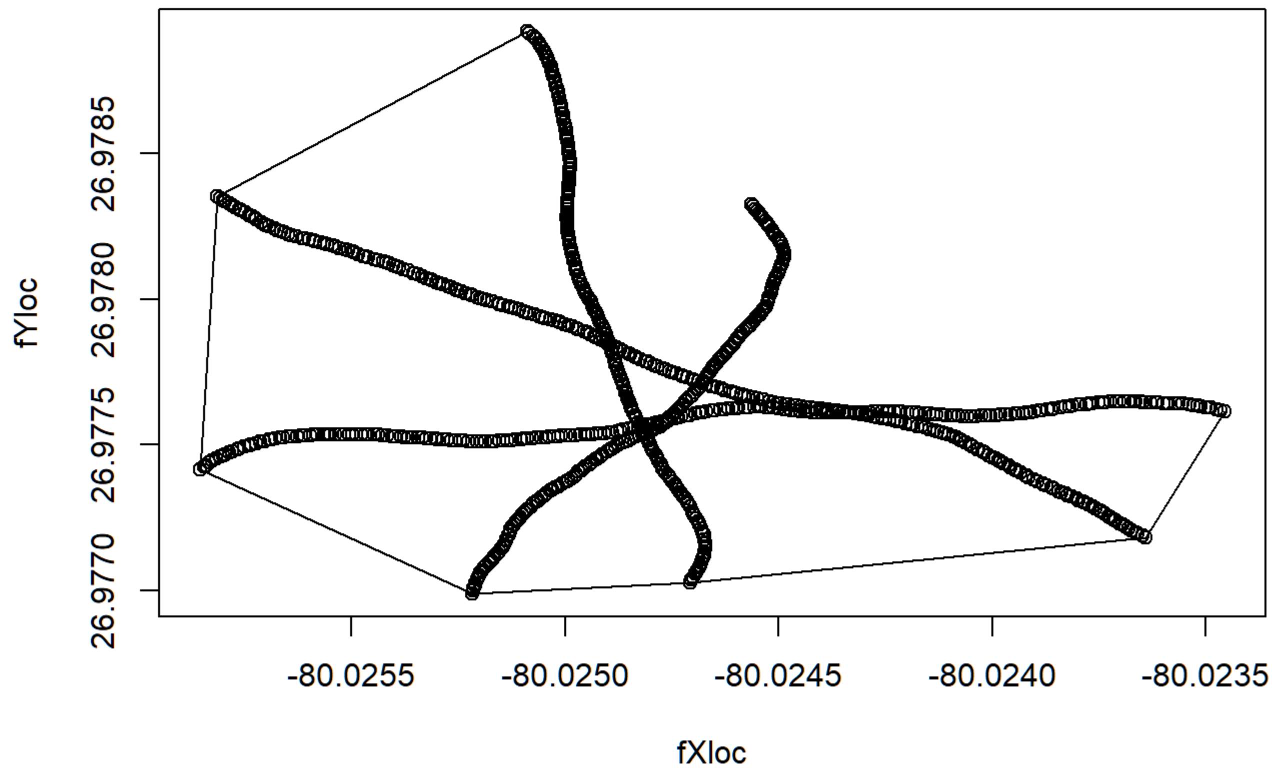
```
##   model        psill        range
## 1   Nug 3.802863e-05 0.0000000000
## 2   Sph 5.961992e-05 0.0001174397
```

```
fsite.psill=fsite.fit$psill[2]
fsite.range=fsite.fit$range[2]
fsite.nugget=fsite.fit$psill[1]

plot(fsite.vario,fsite.fit,pch=20,cex=1.5)
```




Find the convex hull surrounding your data and plot the resulting polygon

```
fXloc = data.frame(Data)$fXloc
fYloc = data.frame(Data)$fYloc
fsite.chull = chull(fXloc,fYloc)

plot(fXloc,fYloc)
lines(fXloc[fsite.chull],fYloc[fsite.chull])
```

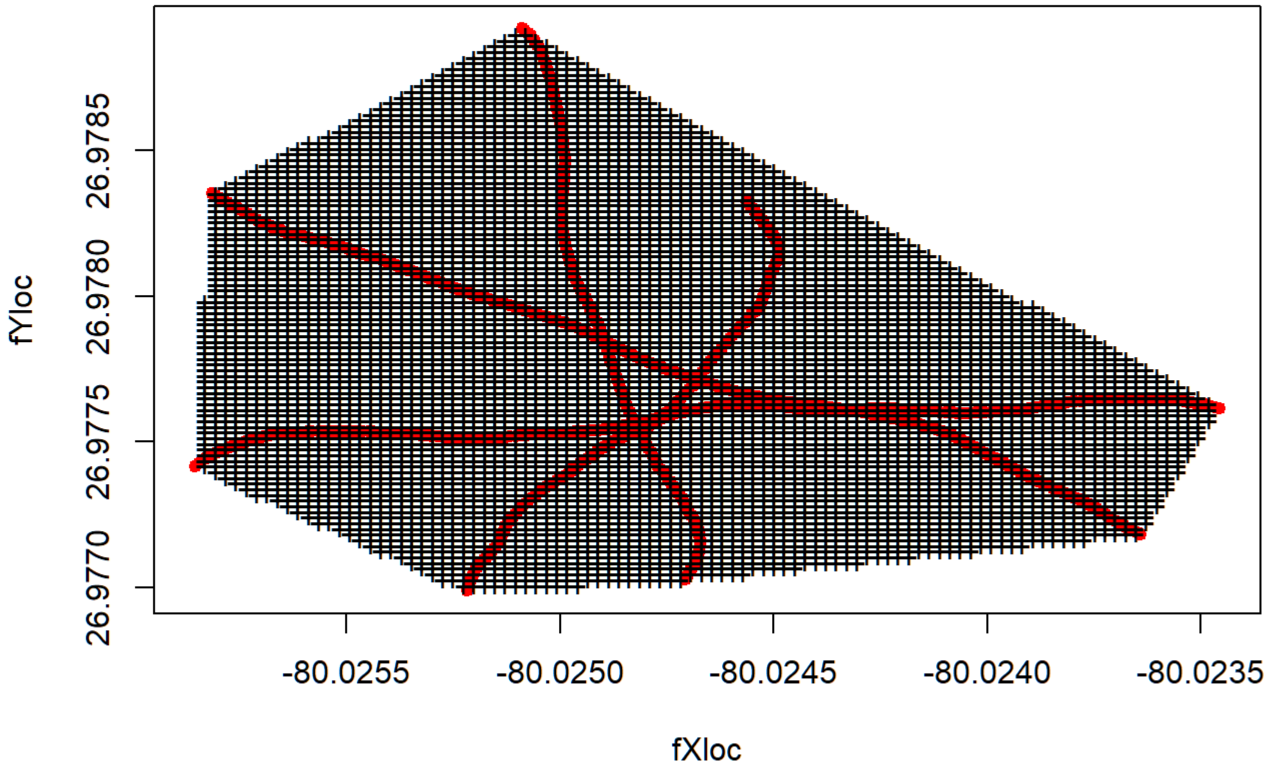



Create a grid of points to predict over

```
fsite.grid = polygrid(
  xgrid=seq(min(fXloc),max(fXloc),length=100),
  ygrid=seq(min(fYloc),max(fYloc),length=100),
  cbind(
    fXloc[fsite.chull],
    fYloc[fsite.chull]))
names(fsite.grid)=c("fXloc","fYloc")
coordinates(fsite.grid)=c("fXloc","fYloc")
fsite.grid = as(fsite.grid, "SpatialPixels")
```

Overlay this grid over your plotted data

```
plot(fYloc~fXloc,cex=1.2,pch=20,col=2)
points(data.frame(fsite.grid)$fXloc,data.frame(fsite.grid)$fYloc,pch="+")
```




Now we are going to use ordinary kriging to predict the values at all points in this domain

```
fsite.ok = krige(flog_NASC~1, Data, fsite.grid, fsite.fit)
```

```
## [using ordinary kriging]
```

Finally, we plot our predicted means and variance

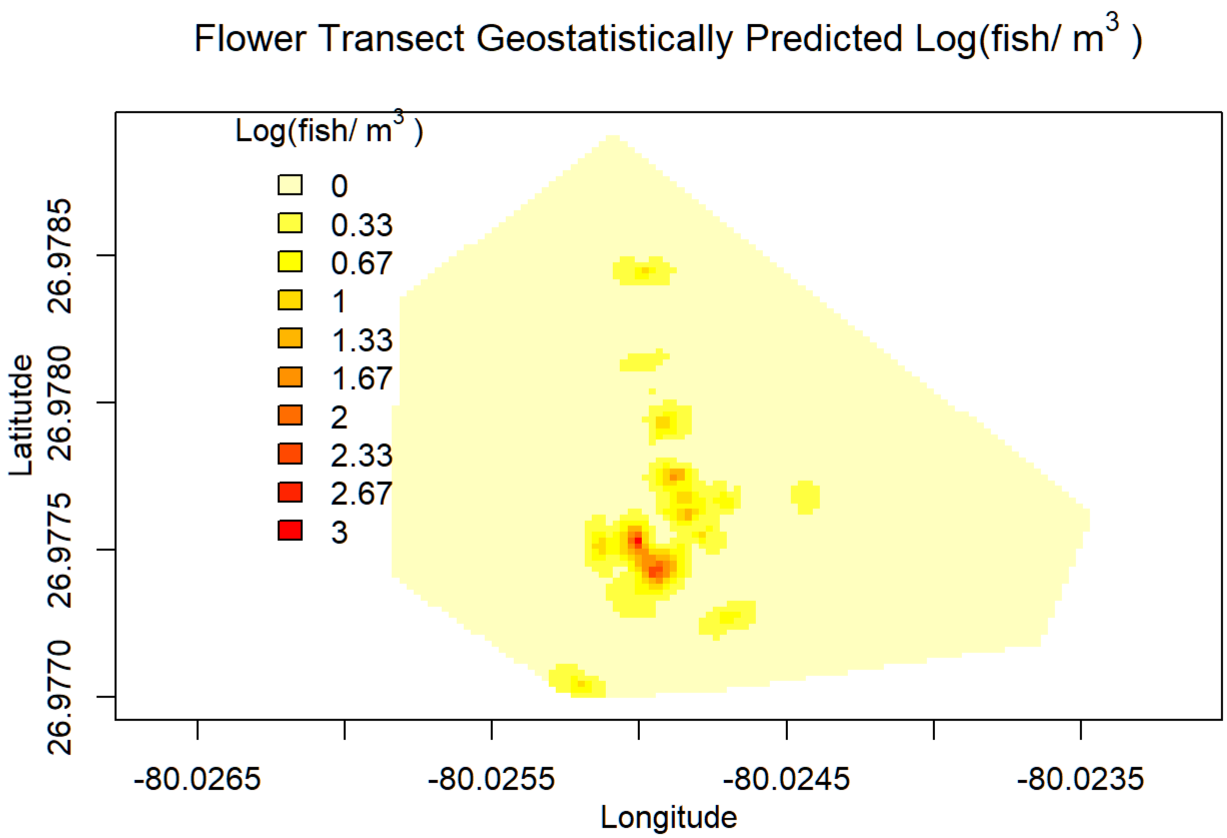
```
mean(fsite.ok$var1.pred)
```

```
## [1] 0.002471503
```

```
range(fsite.ok$var1.pred,na.rm=T)
```

```
## [1] -0.002217027  0.043299413
```

```
image(fsite.ok["var1.pred"],col=rev(heat.colors(10)),axes=T,xlim=c(-80.0263,-80.0235))
mtext(side=1,line=2,"Longitude")
mtext(side=2,line=2,"Latitutde")
title(as.expression(bquote("Flower Transect Geostatistically Predicted Log(fish/"~m^3~")")))
par(xpd=TRUE)
legend(-80.0264,26.979,legend=round(seq(0,3,length=10),2),fill=rev(heat.colors(10)),
       bty="n",title=as.expression(bquote("Log(fish/"~m^3~")")),cex=1)
```




```
sqrt(mean(fsite.ok$var1.var)/length(fsite.ok$var1.var))
```

```
## [1] 0.000119705
```

```
range(fsite.ok$var1.var)
```

```
## [1] 4.668350e-05 9.838078e-05
```

```
image(fsite.ok["var1.var"],col=rev(heat.colors(4)),axes=T,xlim=c(-80.0263,-80.0235))
mtext(side=1,line=2,"Longitude")
mtext(side=2,line=2,"Latitute")
par(xpd=TRUE)
legend(-80.0264,26.979,legend=round(seq(.3,.5,length=10),2),fill=rev(heat.colors(10)),
       bty="n",title="Variance",cex=1)
title(as.expression(bquote("Flower Transect Geostatistical Predicted Var(Log(fish/"~m^3~"))")))
```