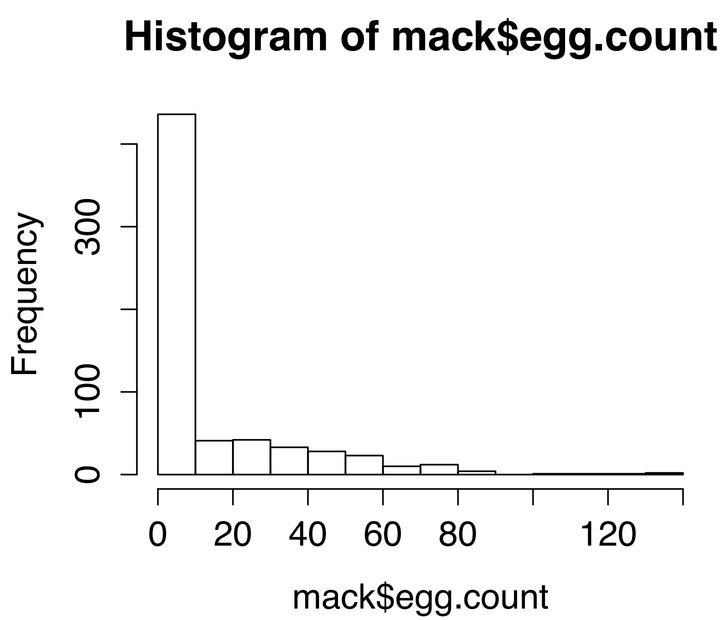
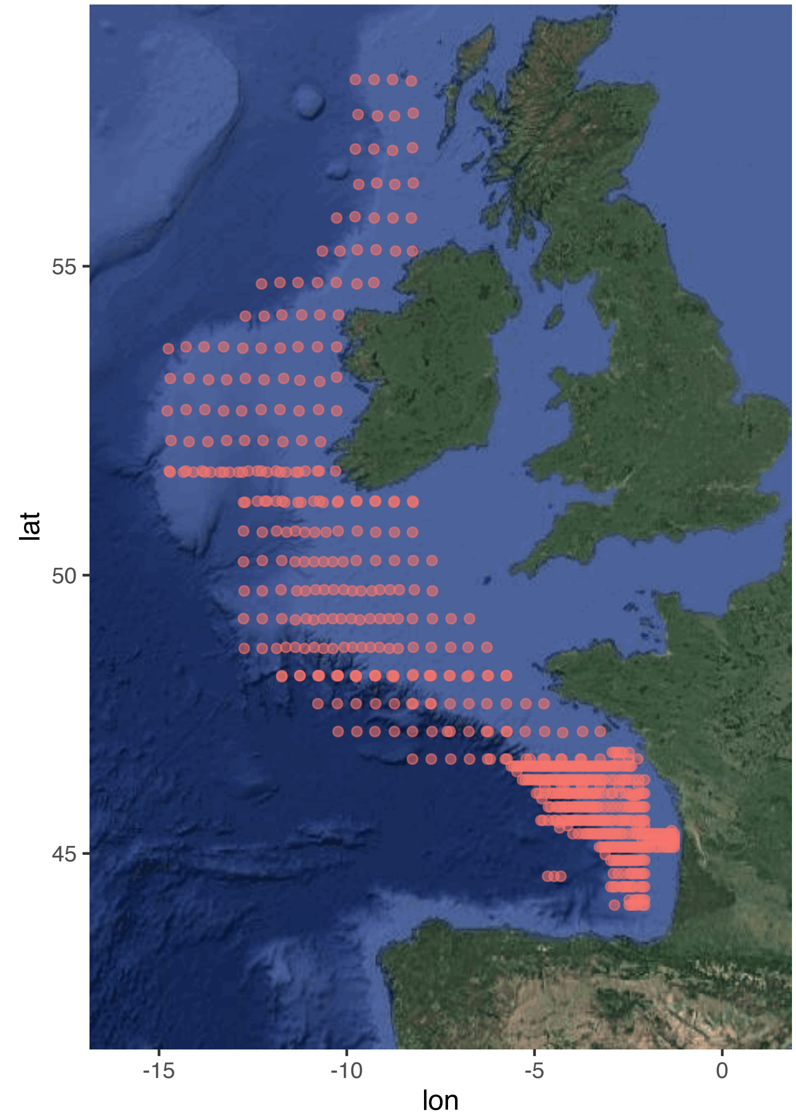
Example: GAMs for spatially distributed responses

Survey of mackerel egg density

- Useful to back-calculate adult biomass, but need good estimates of egg density as a function of location
- Also would be nice to see how environment predicts egg densities
- Many egg counts by four different countries
- Also measure temperature (surface and 20m), seabed depth, distance from 200m contour (shelf edge), time of day





Example: GAMs for spatially distributed responses

Reasons to use GAMs to fit a 2D spatial smoother

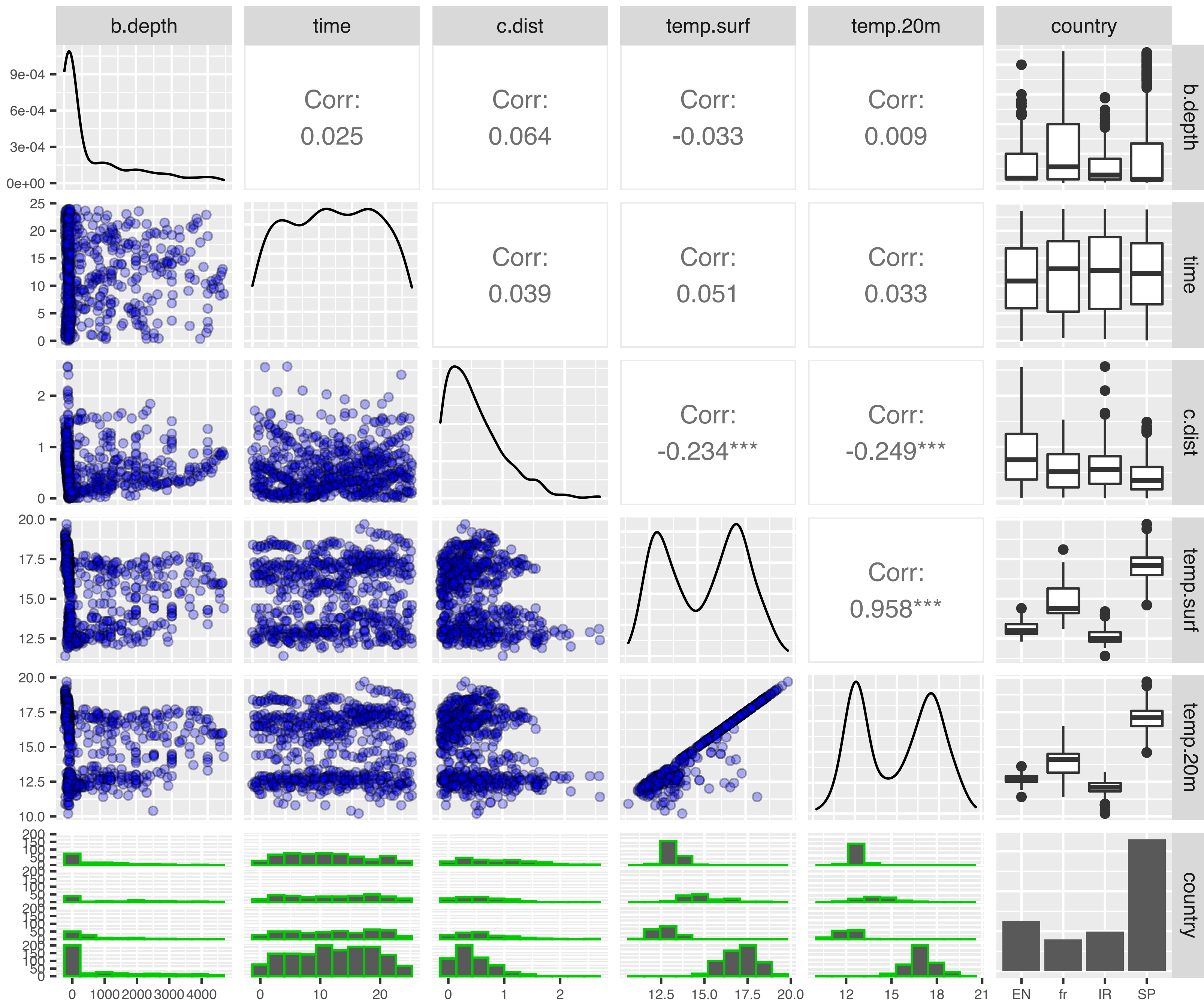
- The authors want good estimates of egg density, which is variable between samples
- When doing a regression with spatial data, need to consider **spatial autocorrelation**
- Samples near each other may be more similar, due to unmeasured effects
- This can be strong violation of independence assumption
- A GAM is one solution: will fit the smooth spatial structure

Example: GAMs for spatially distributed responses

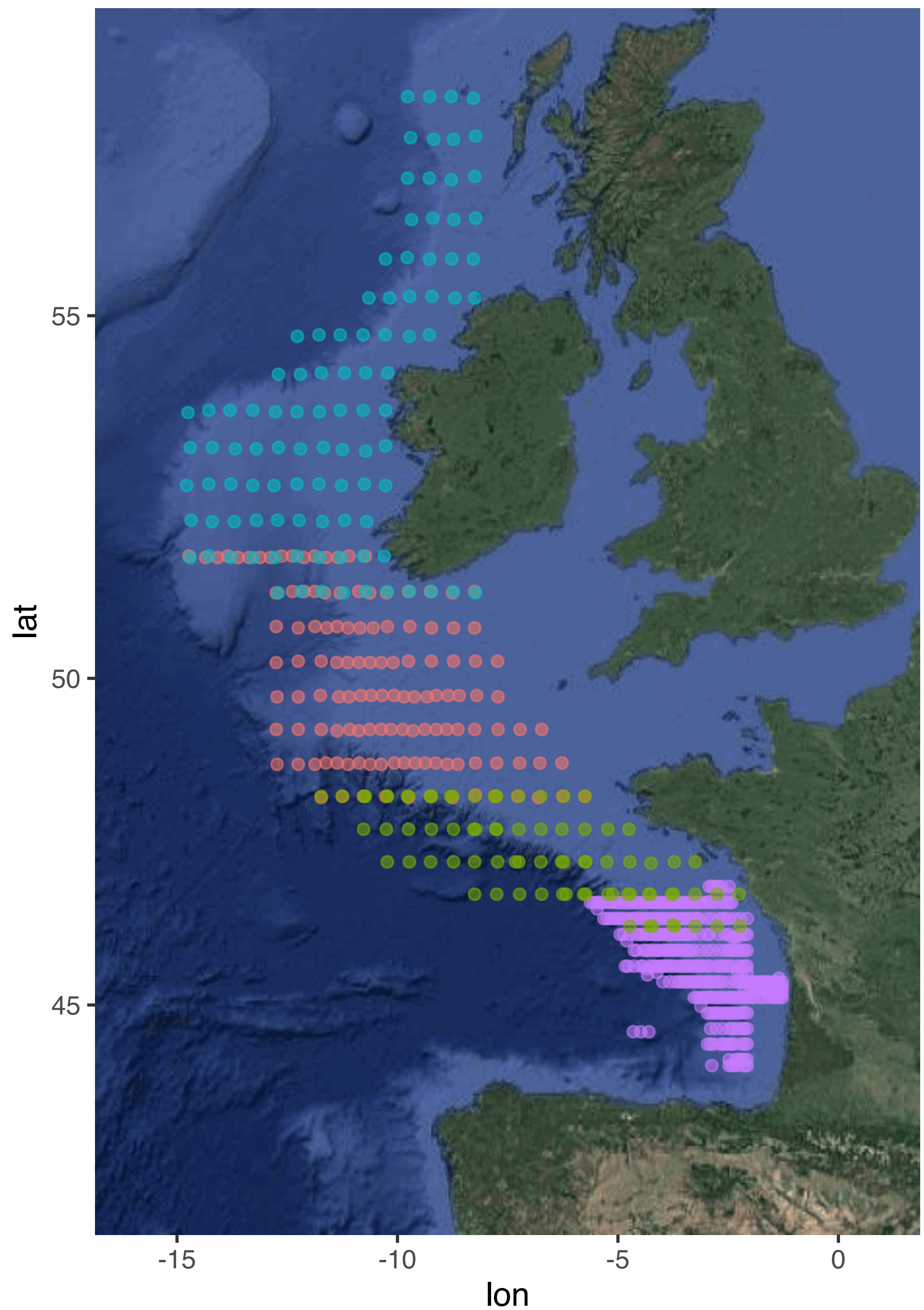
Model strategy:

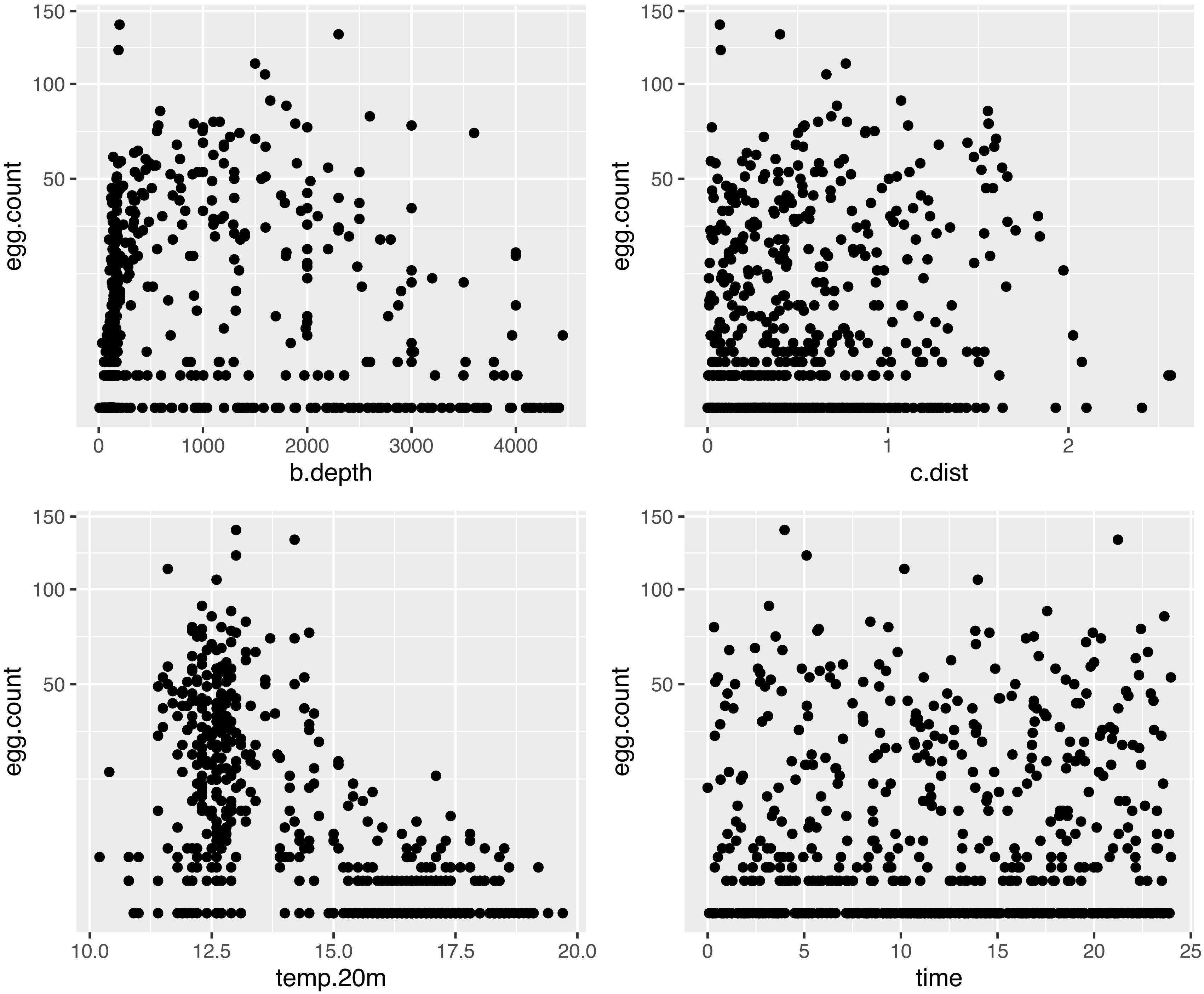
- Want a 2D spatial smoother
- Also want to look at the effects of the predictors
- Have a lot of data (>600 rows), so just fit one big model

Look at predictor correlation



Look at predictor correlation





egg.count
b.depth
c.dist
temp.20m
time

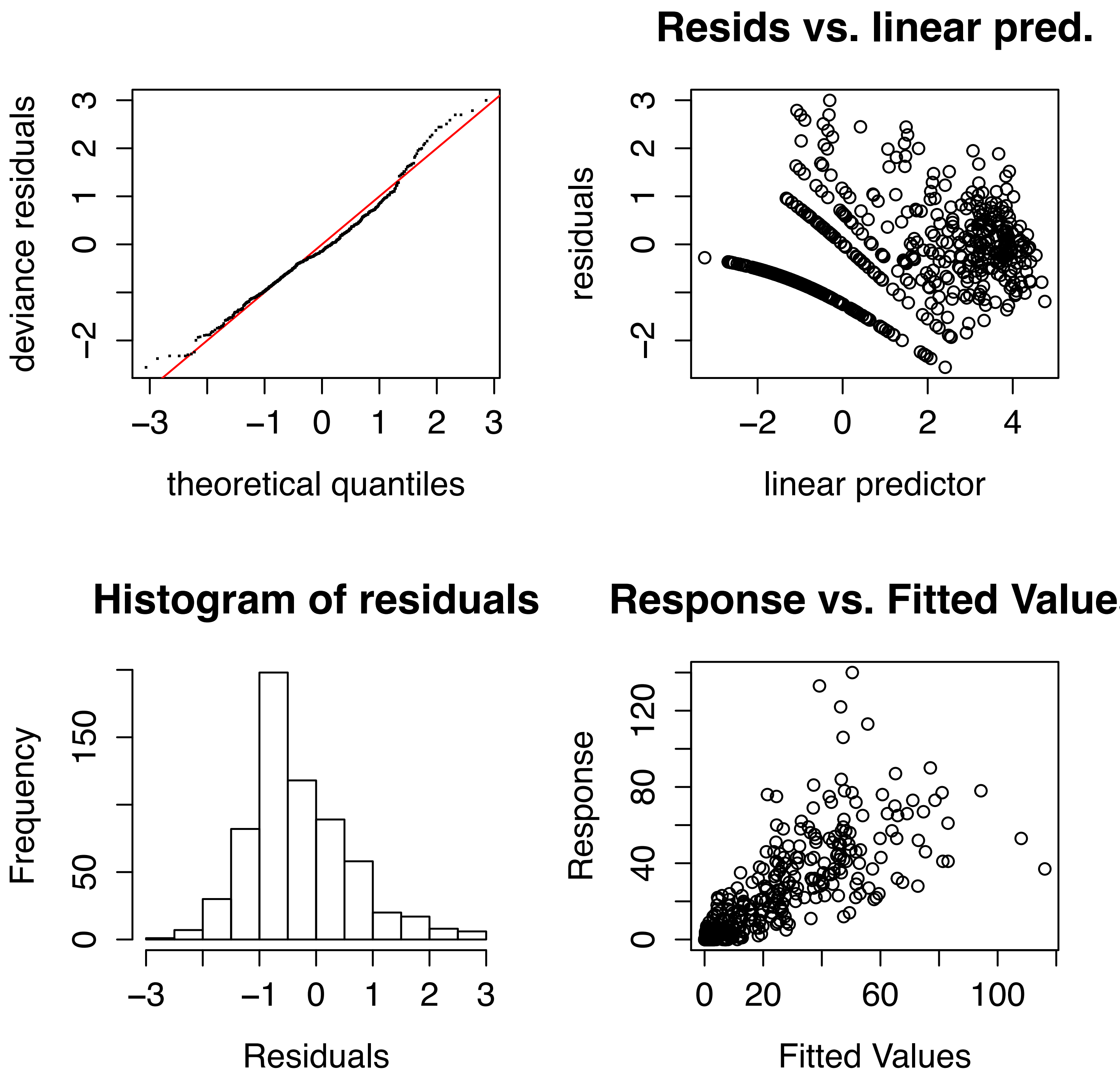
Negative binomial model with offset for net area

```
gm = gam(egg.count ~ s(lon,lat) + s(b.depth) + s(c.dist) + s(temp.20m) + s(tim
e) +offset(log.net.area), data= mack, family = nb)


summary(gm)

##
## Family: Negative Binomial(1.25)
## Link function: log
##
## Formula:
## egg.count ~ s(lon, lat) + s(b.depth) + s(c.dist) + s(temp.20m) +
##     s(time) + offset(log.net.area)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.6118     0.0561    46.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df Chi.sq p-value
## s(lon,lat)  21.40  25.51  160.9 < 2e-16 ***
## s(b.depth)   3.94   4.83   29.1 2.2e-05 ***
## s(c.dist)    1.00   1.01   17.2 3.4e-05 ***
## s(temp.20m)  6.29   7.44   44.0 4.0e-07 ***
## s(time)      4.98   6.07   22.8 0.00092 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.572   Deviance explained = 74.8%
## -REML = 1600.1  Scale est. = 1         n = 634
```

```r
gam.check(gm)

## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-0.0006894,0.0001601]
## (score 1600 & scale 1).
## Hessian positive definite, eigenvalue range [0.000689,103.3].
## Model rank =  66 / 66
##
## 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(lon,lat)  29.000 21.403   0.743    0.00
## s(b.depth)   9.000  3.942   0.875    0.42
## s(c.dist)    9.000  1.003   0.859    0.18
## s(temp.20m)  9.000  6.293   0.790    0.00
## s(time)      9.000  4.979   0.885    0.45
```

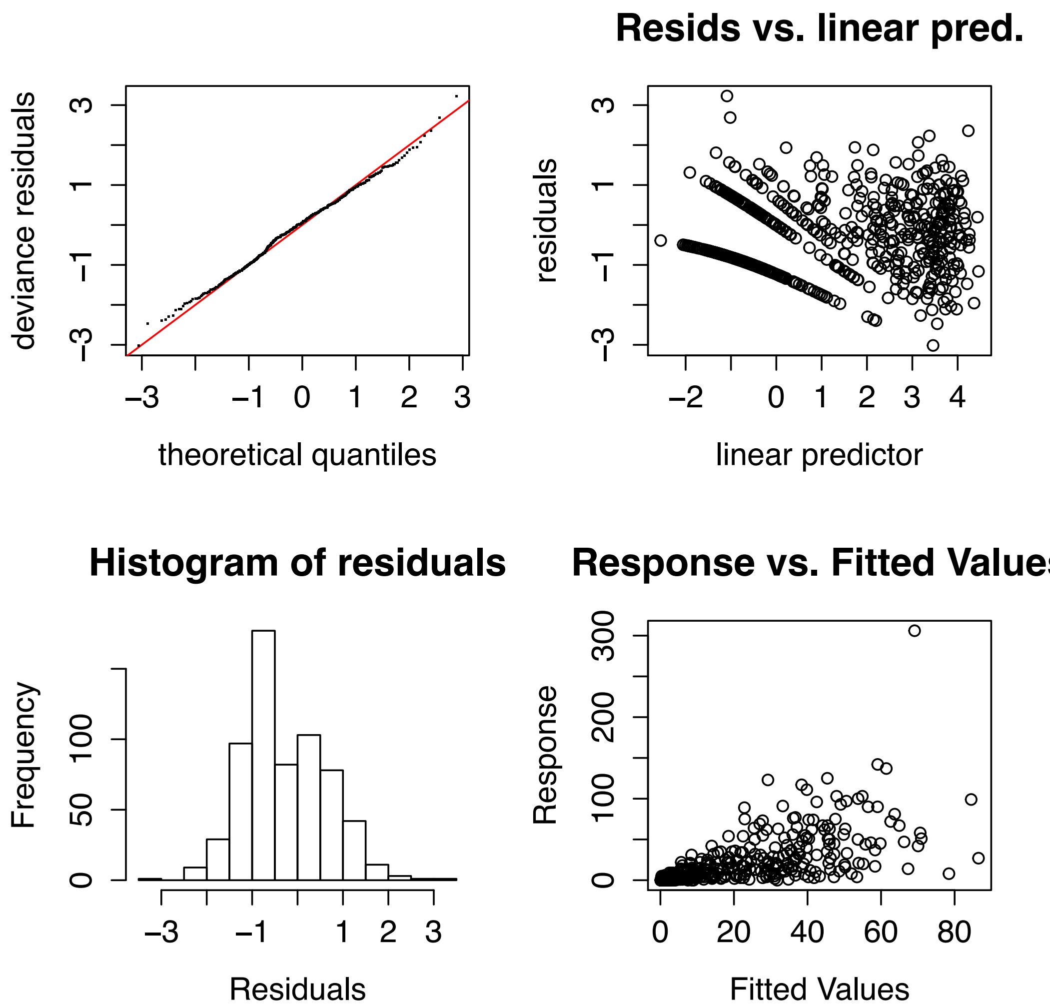
```r
gm = gam(egg.count ~ s(lon,lat,k=100) + s(b.depth) + s(c.dist) + s(temp
20m) + s(time) +offset(log.net.area), data= mack, family = nb)
```

```r
gam.check(gm)
## Method: REML   Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-0.000808,0.0003011]
## (score 1592 & scale 1).
## Hessian positive definite, eigenvalue range [0.0008069,80.39].
## Model rank =  136 / 136
##
## 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(lon,lat)  99.000 51.323   0.834    0.04
## s(b.depth)   9.000  3.914   0.894    0.42
## s(c.dist)    9.000  1.002   0.868    0.29
## s(temp.20m)  9.000  6.277   0.802    0.00
## s(time)      9.000  4.682   0.894    0.50
```

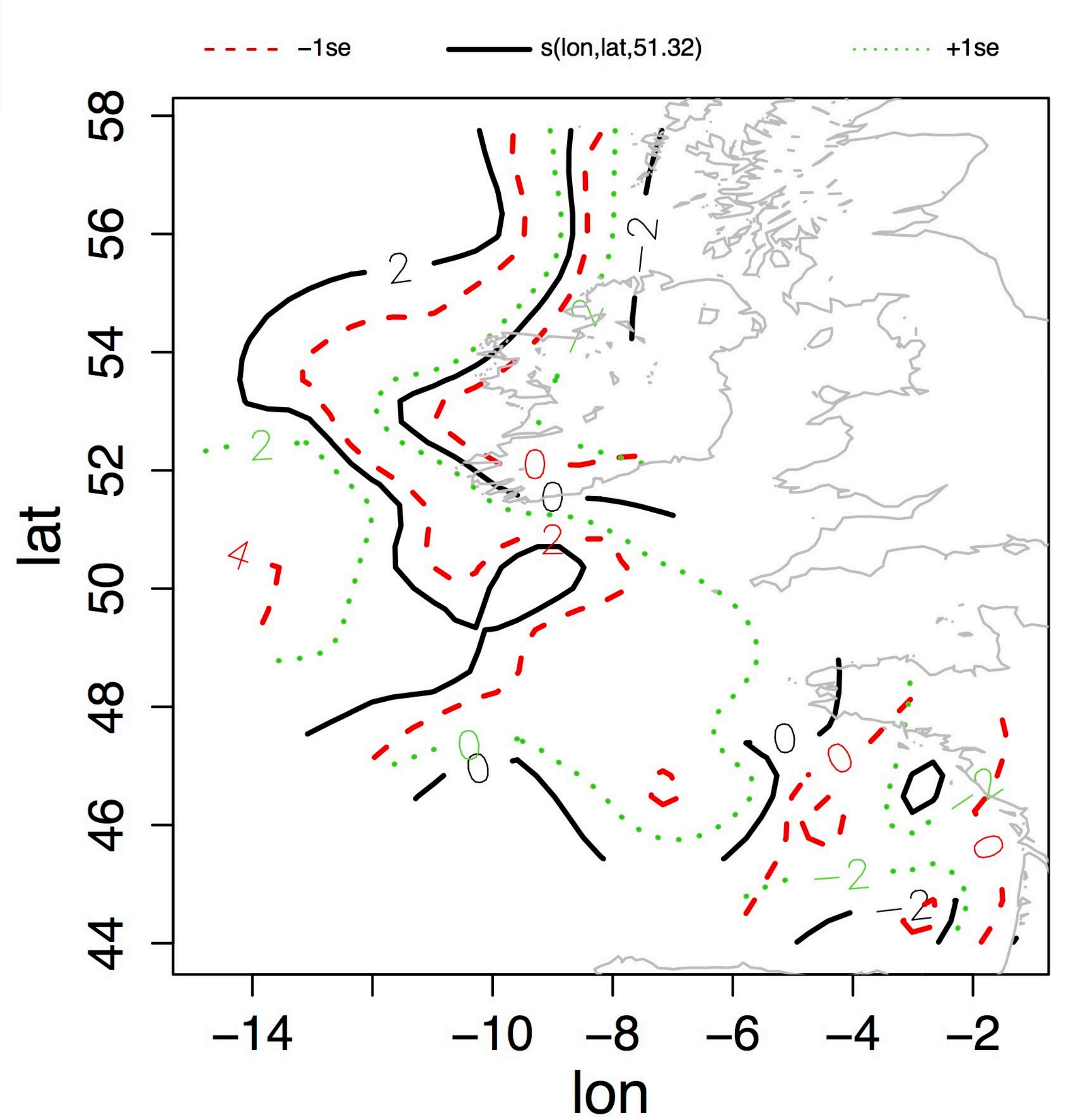


Resids vs. linear pred.
deviance residuals
theoretical quantiles
residuals
linear predictor
Histogram of residuals
Response vs. Fitted Values
Frequency
Residuals
Response
Fitted Values

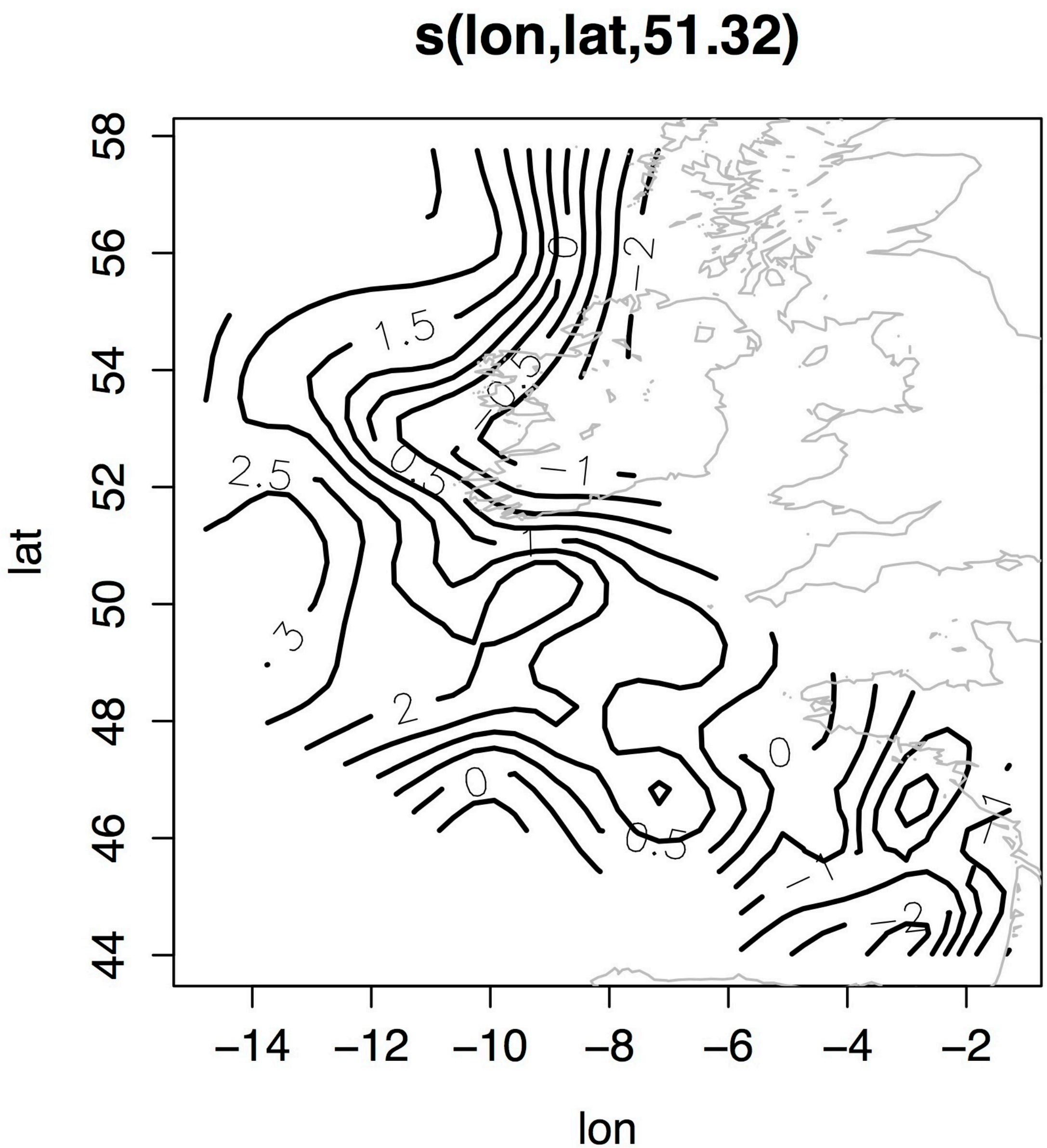
```r
#simulate your own values
simulated = rnbinom(length(fitted(gm)), mu = fitted(gm), size = gm$family$getT
heta(TRUE))

gm.sim = gam(simulated ~s(lon,lat,k=100) + s(b.depth) + s(c.dist) + s(temp.20
m) + s(time) +offset(log.net.area), data= mack, family = nb)
gam.check(gm.sim)
```



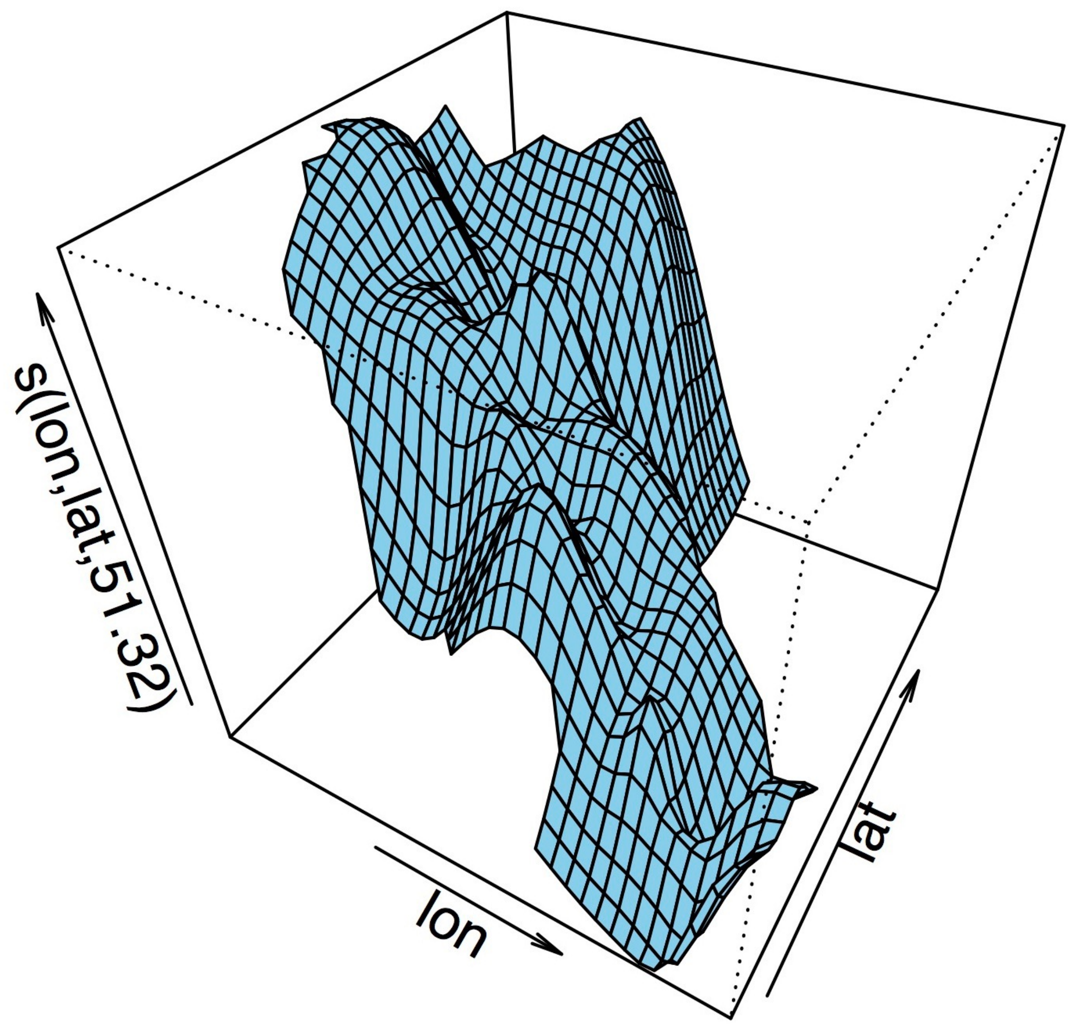
```r
plot(gm, select = 1, lwd = 2, scheme = 0, rug = F, se = T)
lines(coast$lon,coast$lat, col = 'grey')
```



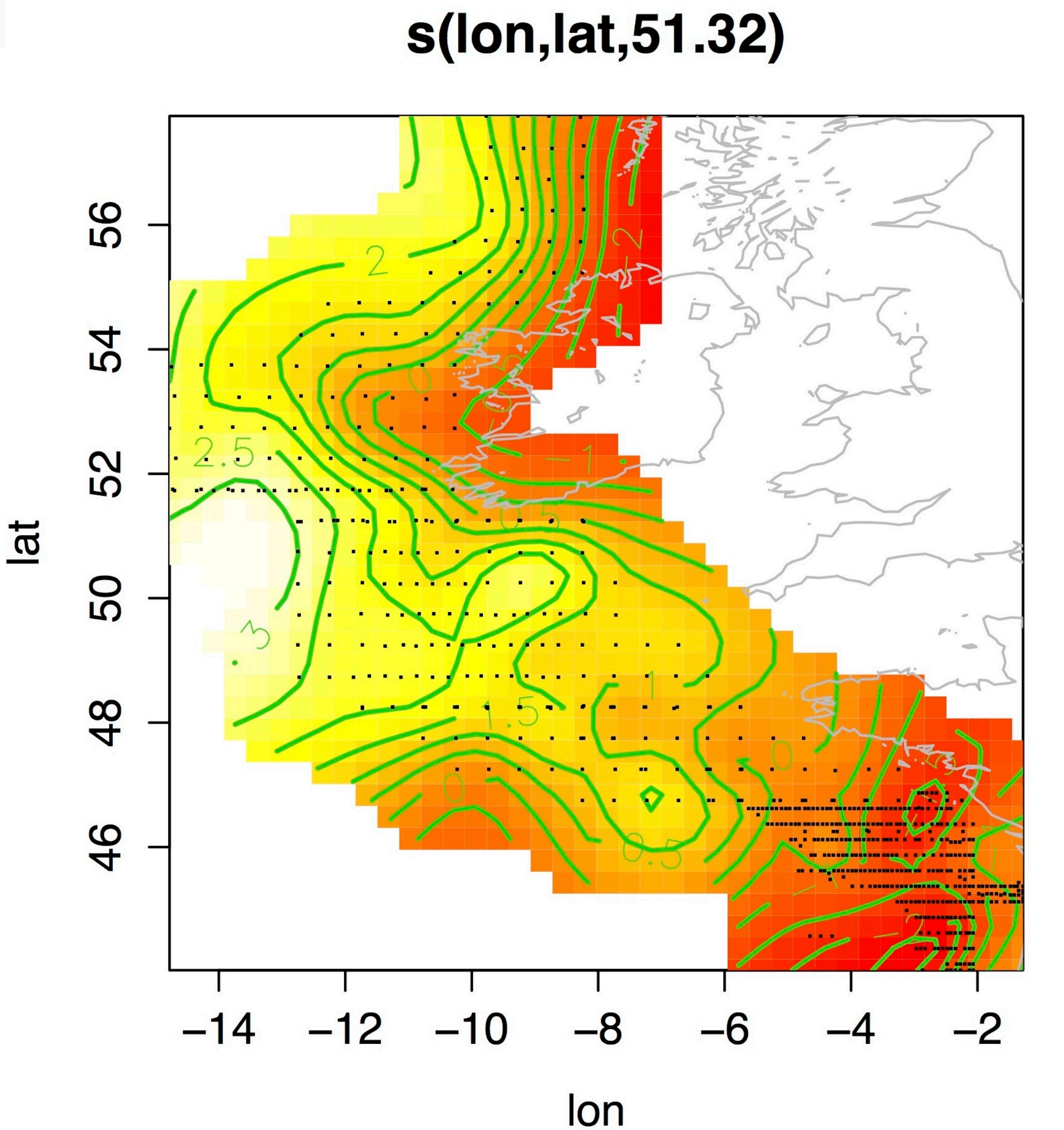
```r
plot(gm, select = 1, lwd = 2, scheme = 0, rug = F, se = F)
lines(coast$lon,coast$lat, col = 'grey')
```



```r
plot(gm, select = 1, scheme = 1, theta = 30, col = 'skyblue', phi = 40)
```



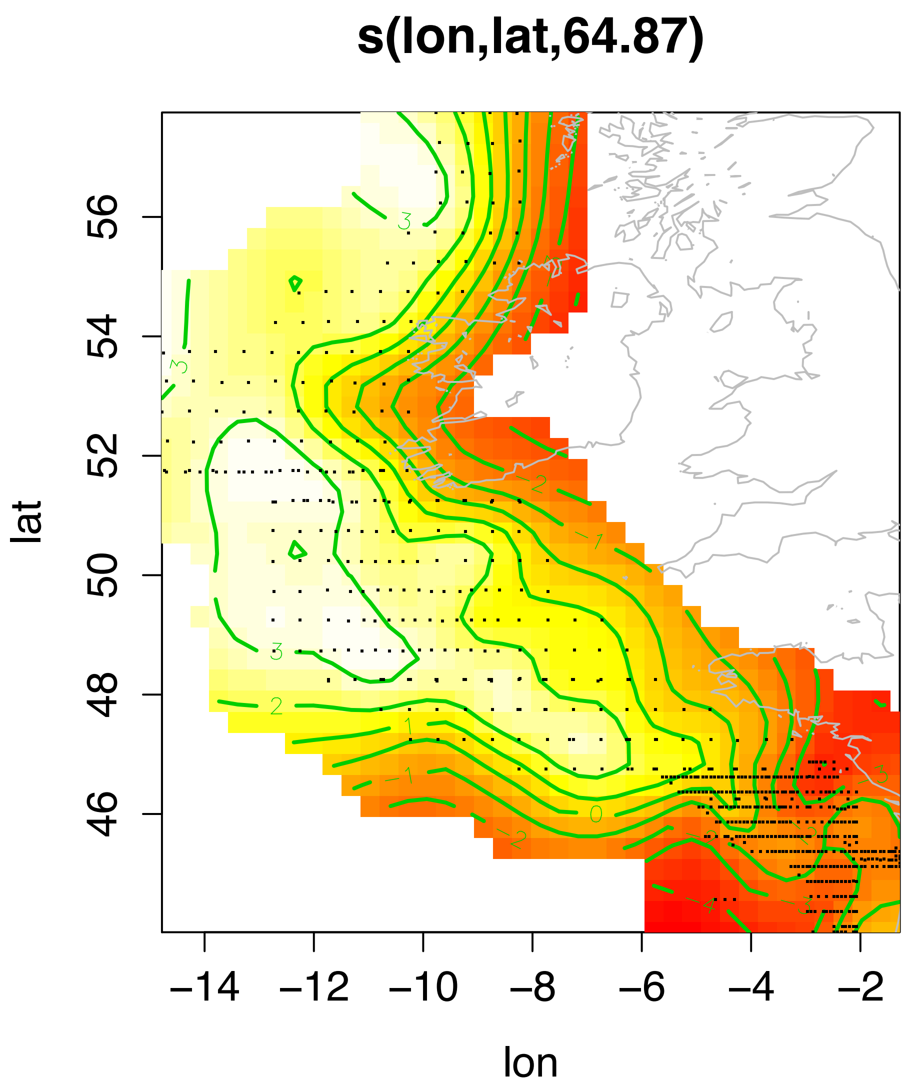
```r
plot(gm, select = 1, scheme = 2, lwd = 2)

lines(coast$lon,coast$lat, col = 'grey')
```



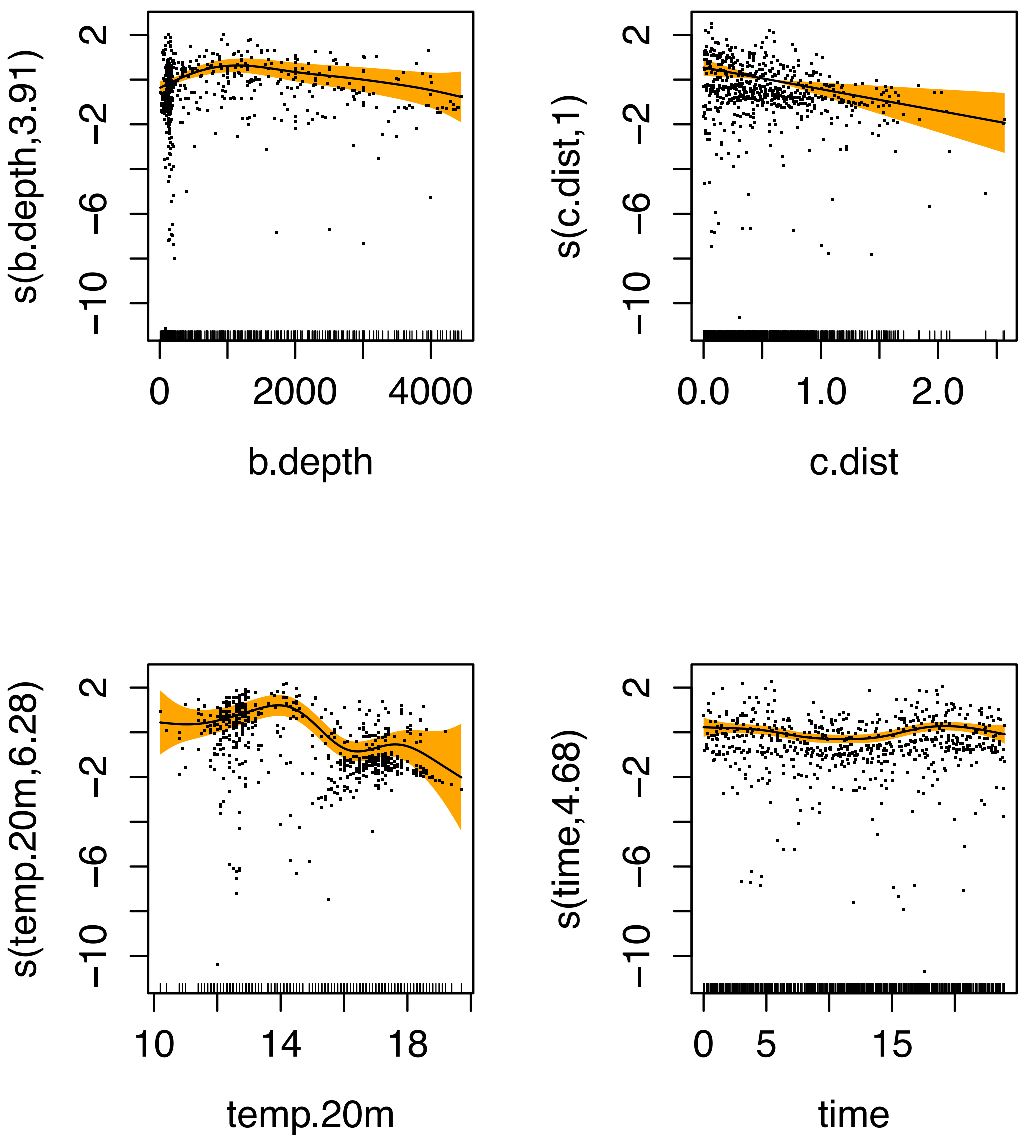
The smoother finds substantial spatial variation (50-fold)

But this is variation not explained by the other predictors

Can compare to model with only the 2D smoother

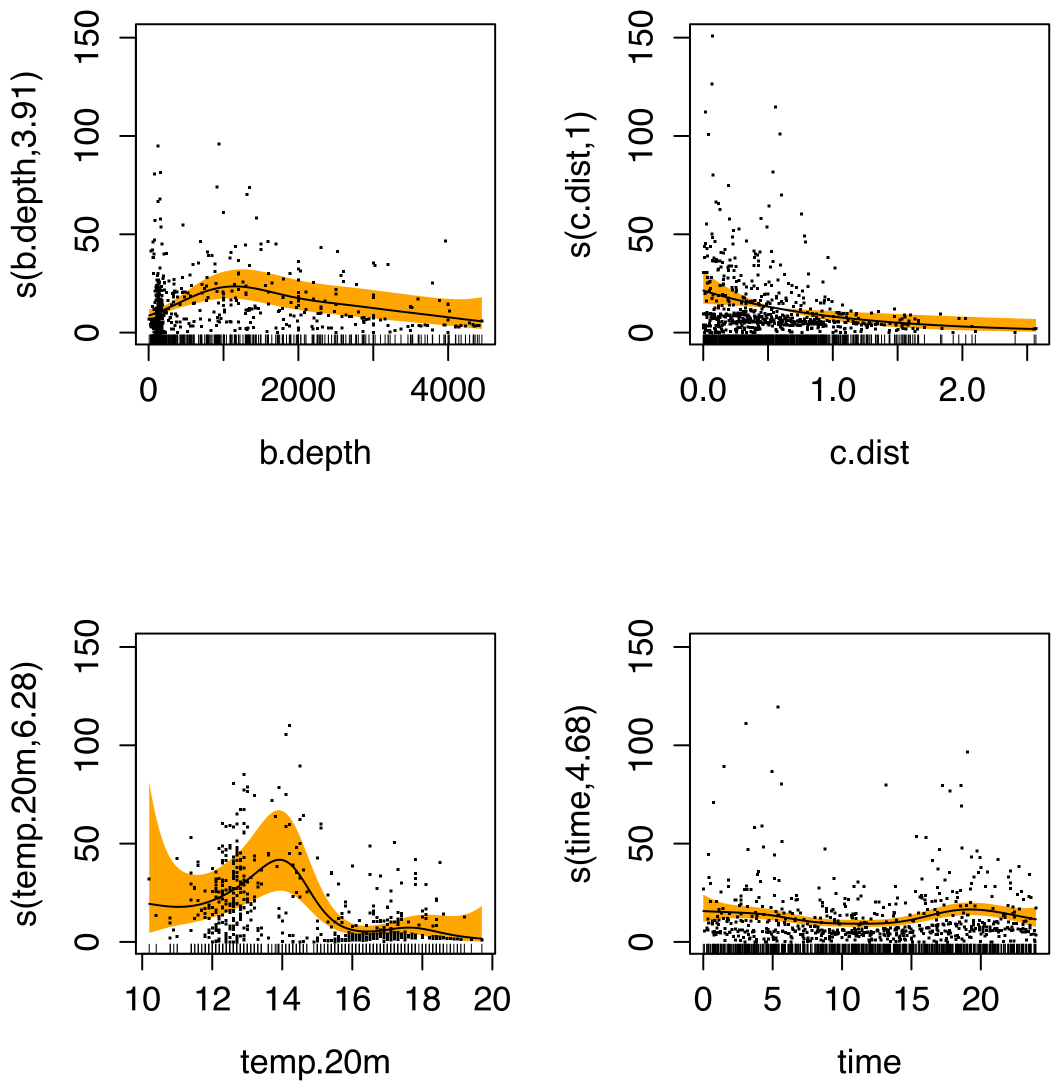


```r
par(mfrow = c(2,2))
plot(gm, select = 2, residuals = T, shade.col = 'orange', shade = T, col = 'black')
plot(gm, select = 3, residuals = T, shade.col = 'orange', shade = T, col = 'black')
plot(gm, select = 4, residuals = T, shade.col = 'orange', shade = T, col = 'black')
plot(gm, select = 5, residuals = T, shade.col = 'orange', shade = T, col = 'black')
```



Check out
s(time, bs = ‘cc’)
For cyclic
smoother

```
par(mfrow = c(2,2))
plot(gm, select = 2, residuals = T, shade.col = 'orange', shade = T, col = 'black',
shift = coef(gm)[1], trans = function(x) exp(x))
plot(gm, select = 3, residuals = T, shade.col = 'orange', shade = T, col = 'black',
shift = coef(gm)[1], trans = function(x) exp(x)))
plot(gm, select = 4, residuals = T, shade.col = 'orange', shade = T, col = 'black',
shift = coef(gm)[1], trans = function(x) exp(x)))
plot(gm, select = 5, residuals = T, shade.col = 'orange', shade = T, col = 'black',
shift = coef(gm)[1], trans = function(x) exp(x)))
```



Smoother significance

- Can get from summary() or anova(), either gives approximate marginal tests by default (note this is a little different from lm and glm)

```
anova(gm)

##
## Family: Negative Binomial(1.56)
## Link function: log
##
## Formula:
## egg.count ~ s(lon, lat, k = 100) + s(b.depth) + s(c.dist) + s(temp.20m) +
##     s(time) + offset(log.net.area)
##
## Approximate significance of smooth terms:
##                edf Ref.df Chi.sq p-value
## s(lon,lat)  51.32  67.26 245.58 < 2e-16
## s(b.depth)   3.91   4.75  24.85 0.00013
## s(c.dist)    1.00   1.00   8.35 0.00390
## s(temp.20m)  6.28   7.39  41.47 1.1e-06
## s(time)      4.68   5.71  18.14 0.00499
```

Some thoughts on the model results

- Both b.depth and c.dist are capturing how egg density tends to increase and then decrease, moving away from the coast

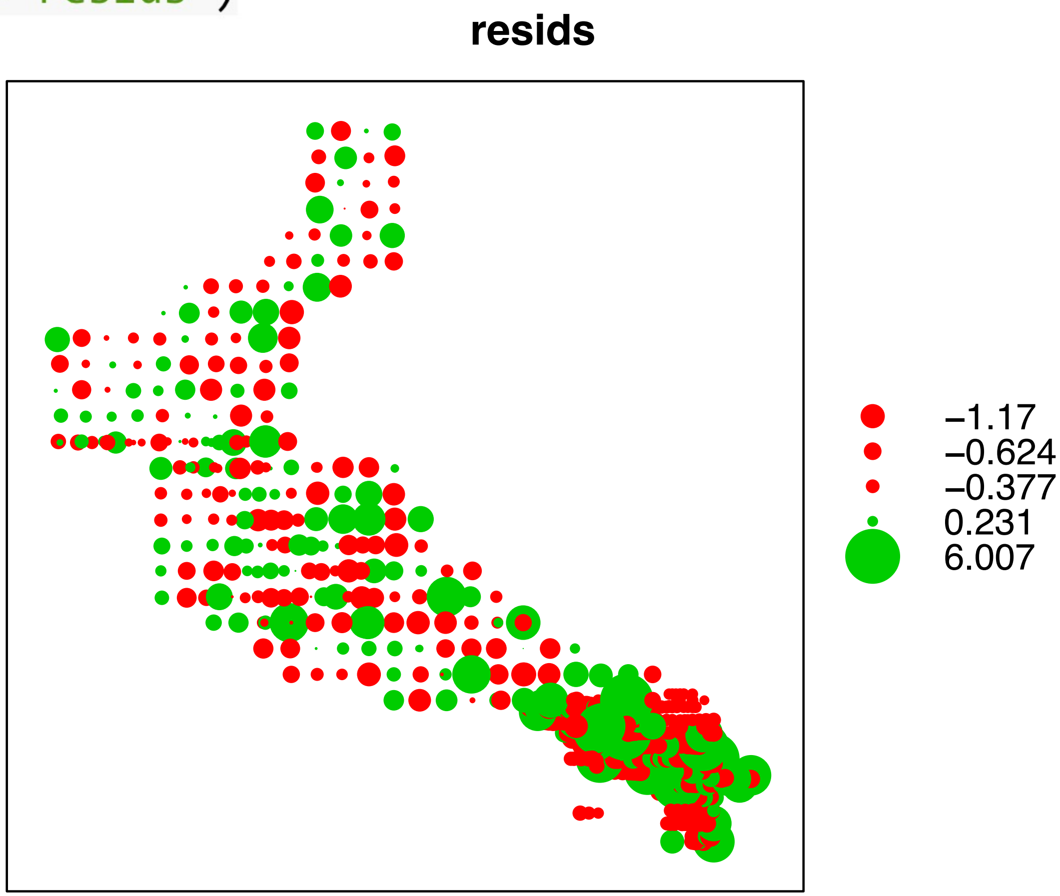
- Maybe 200 m is not the best reference

- But depth can’t capture most of the spatial variation anyways

- Temperature captures some of the north-south

We skipped a diagnostic: spatial correlation in residuals

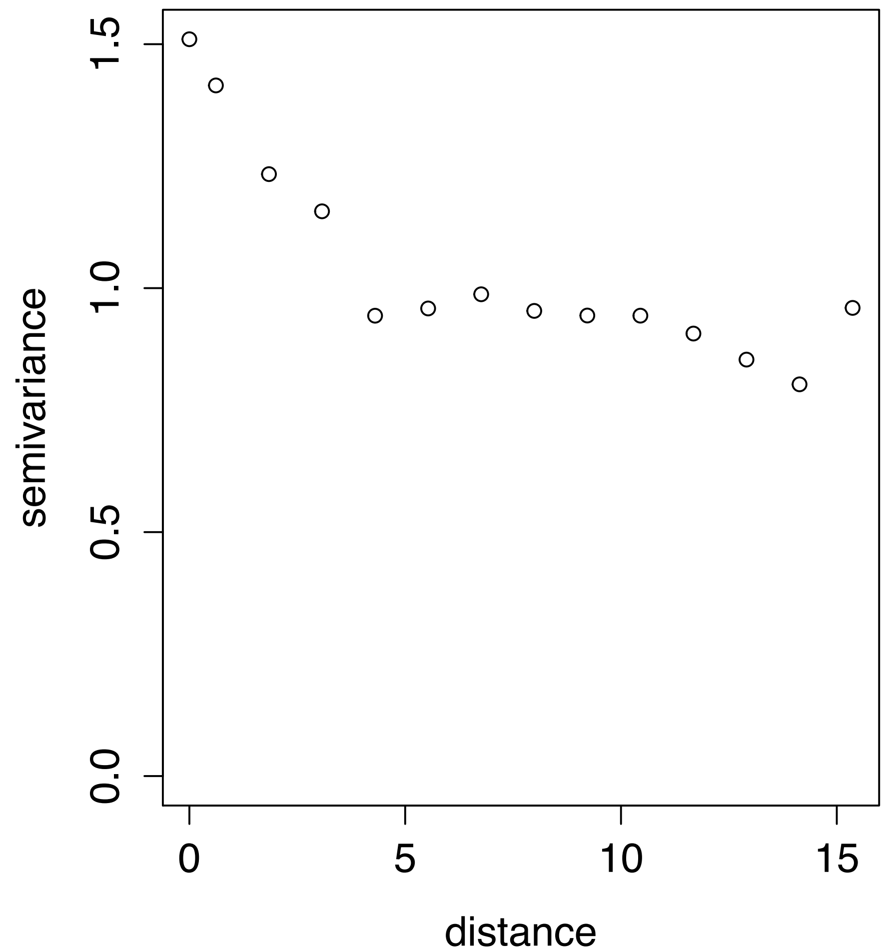
```r
library(sp)

dat<-data.frame(lon = mack$lon, lat = mack$lat, resids=resid(gm, type =
'pearson'))
coordinates(dat)<-c('lon','lat')
bubble(dat,zcol='resids')
```



To directly look for spatial auto-correlation: semivariogram

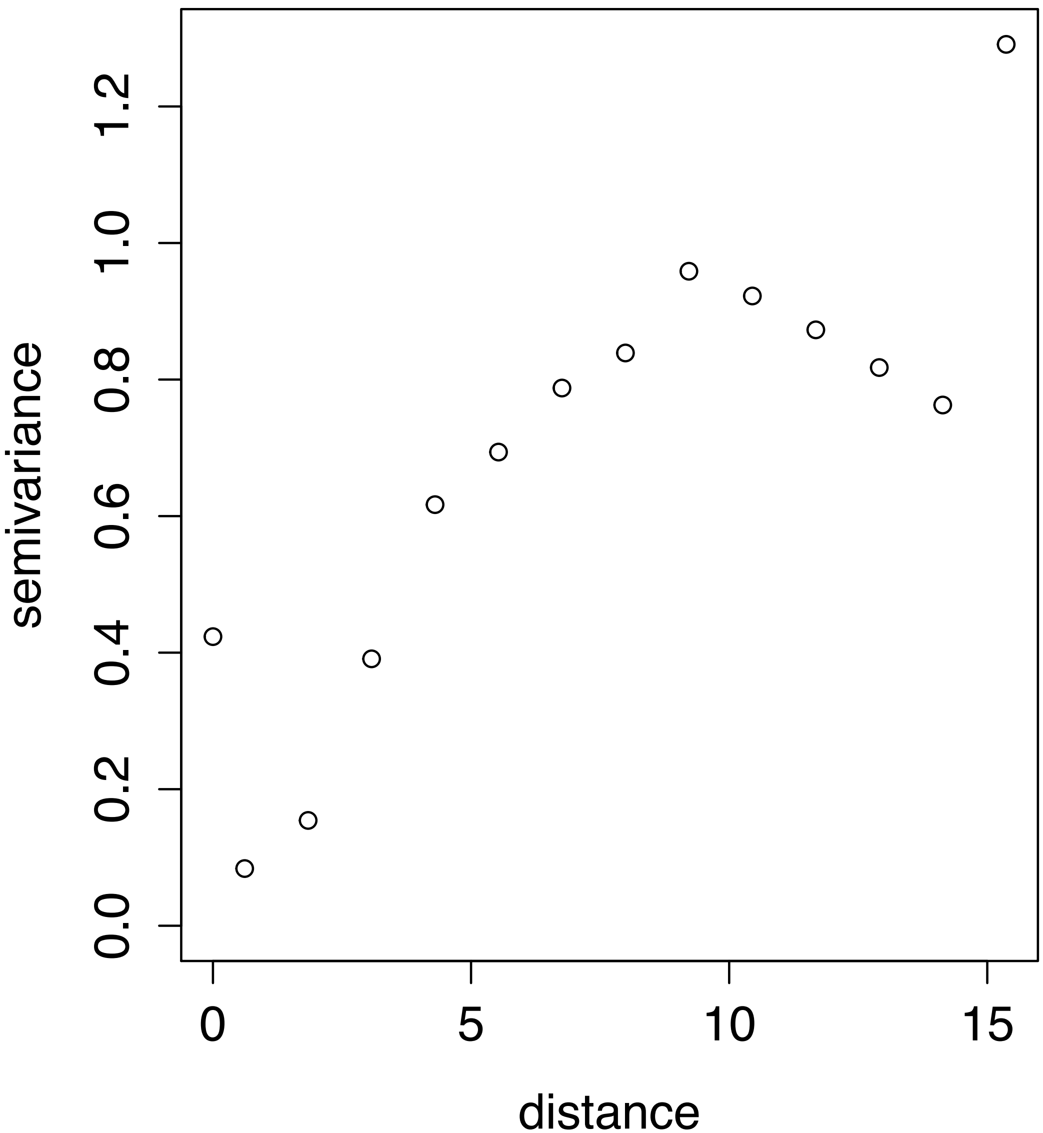
```r
library(geoR)

v1 <- variog(coords = mack[,c('lat','lon')], data = residuals(gm, type
= 'pearson'))

plot(v1)
```

- Looking at squared difference of residuals
- $(e_i - e_k)^2$
- As a function of distance in space



To directly look for spatial auto-correlation: semivariogram

Can compare to a model with no predictors



Would be nice to plot model predictions in space

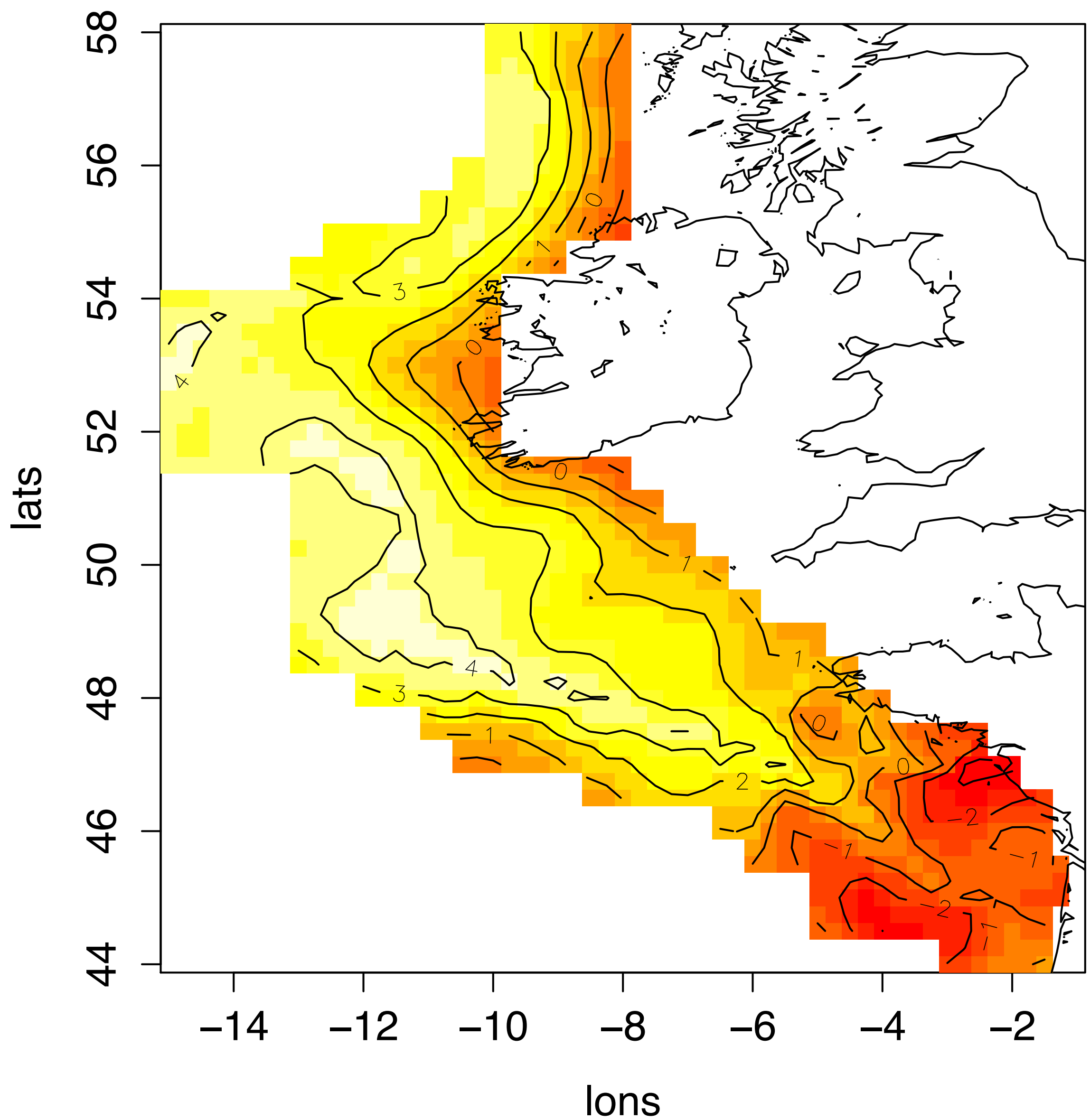
- Easiest way is to make a grid with average values of the predictors
- Then use the gridded predictors to make gridded model predictions
- Let's try it with b.depth, c.dist, temp.20m
- But need values for all predictors in the model

```
data(mackp)

mackp$log.net.area = median(mack$log.net.area)
mackp$time = 12

mackp$predicted = predict(gm, newdata = mackp)

lons<-seq(-15,-1,1/4)
lats<-seq(44,58,1/4)
zz<-matrix(NA, nrow = 57, ncol = 57)
zz[mackp$area.index]<-mackp$predicted
```

Would be nice to plot model predictions in space

```r
#use image to make a heat map of the predictions
image(lons,lats,zz)

#make a contour plot on top of the heat map
contour(lons,lats,zz, add = T)

#add the coastline
lines(coast$lon,coast$lat,col=1)
```

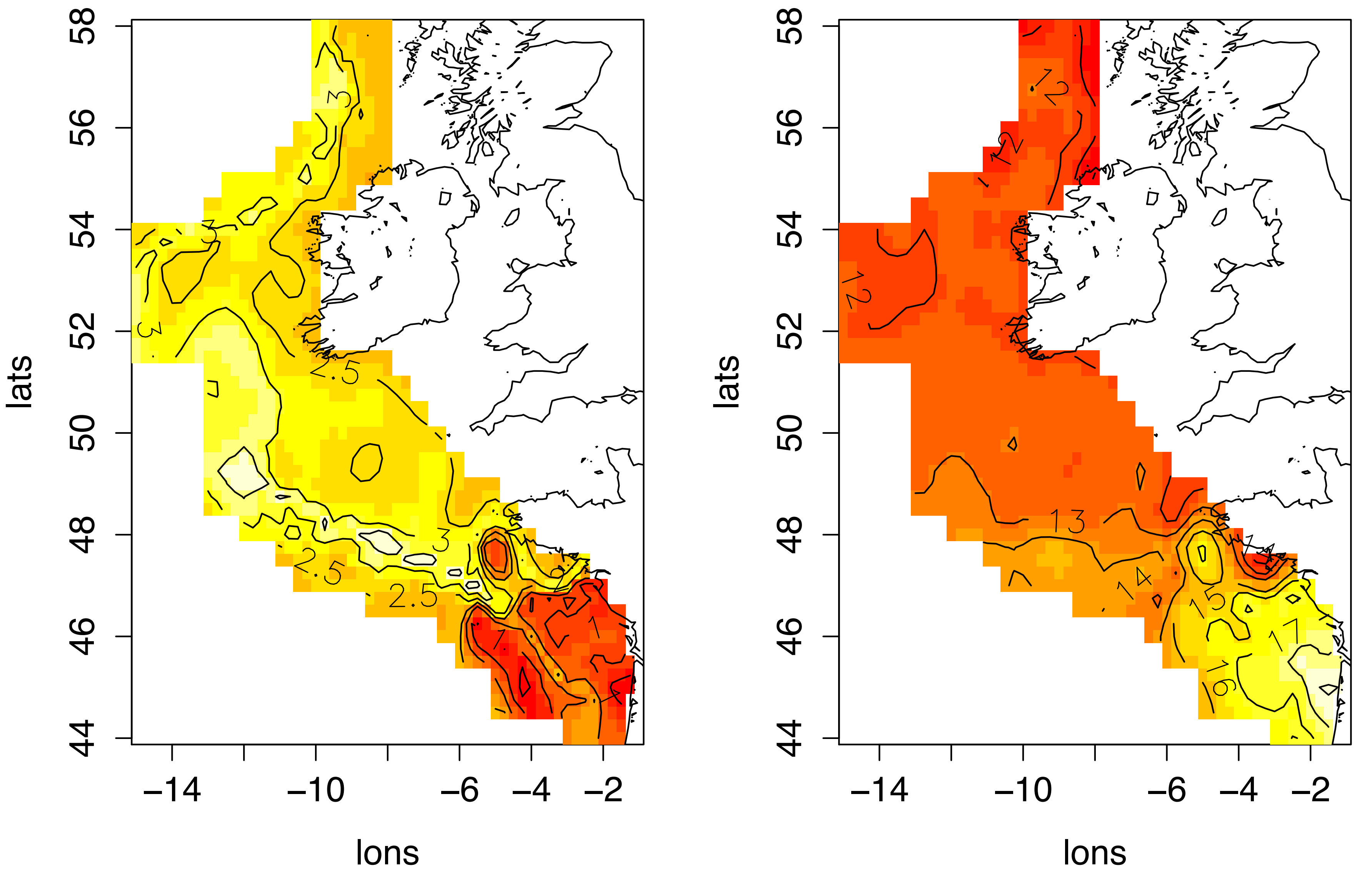
Would be nice to plot model predictions in space



Can look at individual predictors, by holding the other predictors constant

```
#effect of temp
data(mackp)
mackp$log.net.area = median(mack$log.net.area)
mackp$time = 12
mackp$lat = 50
mackp$lon = -8
mackp$c.dist = mean(mackp$c.dist)

#make predictions on the grid
mackp$predicted = predict(gm, newdata = mackp)
```

Can look at individual predictors, by holding the other predictors constant



Other notes

Can use smoothers for time, e.g. s(Day) or s(Year)

Can use 2D smoothers for non-parametric interactions, e.g. s(light, nitrate)