GAMs for Foraging Whales

Yvonne Barkley

10/4/2020

Load libraries

```
library(tidyverse)
library(mgcv)
library(corrplot)
library(geoR)
library(here)
```

Research question:

What environmental variables characterize sperm whale habitat?

Hypothesis: Sperm whales are found in deep, productive offshore waters.

Load universal variables

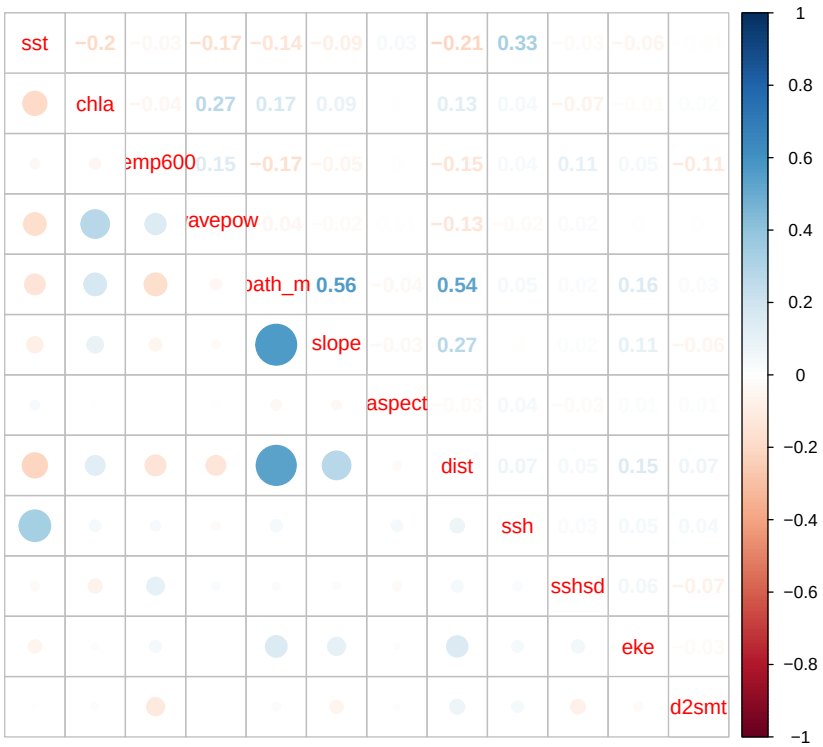
```
# Values used for file and directory names
survey = "AllSurveys"
gridsize = 25
loctype = "Combined"
loctype2 = "Comb"
```

Load data from 'models/data' folder

```
PmScaled <- readRDS(here::here(paste0("output/models/", loctype,
    "/data/", "CompletePm_", gridsize, "km_", loctype2, "_scaled.rda")))
# add column for log effort as offset #
PmScaled$log.effort = log(PmScaled$EffArea)
PmScaled <- subset(PmScaled, chla <= 9)  #some outliers in a handful of absences
PmScaled$distseamt.r = PmScaled$distseamt.r/1000
```

Check correlation of covariates

```
require(corrplot)
corrplot.mixed(cor(PmScaled[, 18:29]), upper = "number", lower = "circle")
```




```
# Are all correlation coefficients < |0.6|?
abs(cor(PmScaled[, 18:29])) <= 0.6
```

```
          sst  chla temp600 wavepow bath_m slope aspect  dist   ssh sshsd   eke
sst     FALSE  TRUE    TRUE    TRUE   TRUE  TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
chla     TRUE FALSE    TRUE    TRUE   TRUE  TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
temp600  TRUE  TRUE   FALSE    TRUE   TRUE  TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
wavepow  TRUE  TRUE    TRUE   FALSE   TRUE  TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
bath_m   TRUE  TRUE    TRUE    TRUE  FALSE  TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
slope    TRUE  TRUE    TRUE    TRUE   TRUE FALSE   TRUE  TRUE  TRUE  TRUE  TRUE
aspect   TRUE  TRUE    TRUE    TRUE   TRUE  TRUE  FALSE  TRUE  TRUE  TRUE  TRUE
dist     TRUE  TRUE    TRUE    TRUE   TRUE  TRUE   TRUE FALSE  TRUE  TRUE  TRUE
ssh      TRUE  TRUE    TRUE    TRUE   TRUE  TRUE   TRUE  TRUE FALSE  TRUE  TRUE
sshsd    TRUE  TRUE    TRUE    TRUE   TRUE  TRUE   TRUE  TRUE  TRUE FALSE  TRUE
eke      TRUE  TRUE    TRUE    TRUE   TRUE  TRUE   TRUE  TRUE  TRUE  TRUE FALSE
d2smt    TRUE  TRUE    TRUE    TRUE   TRUE  TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
         d2smt
sst       TRUE
chla      TRUE
temp600   TRUE
wavepow   TRUE
bath_m    TRUE
slope     TRUE
aspect    TRUE
dist      TRUE
ssh       TRUE
sshsd     TRUE
eke       TRUE
d2smt    FALSE
```

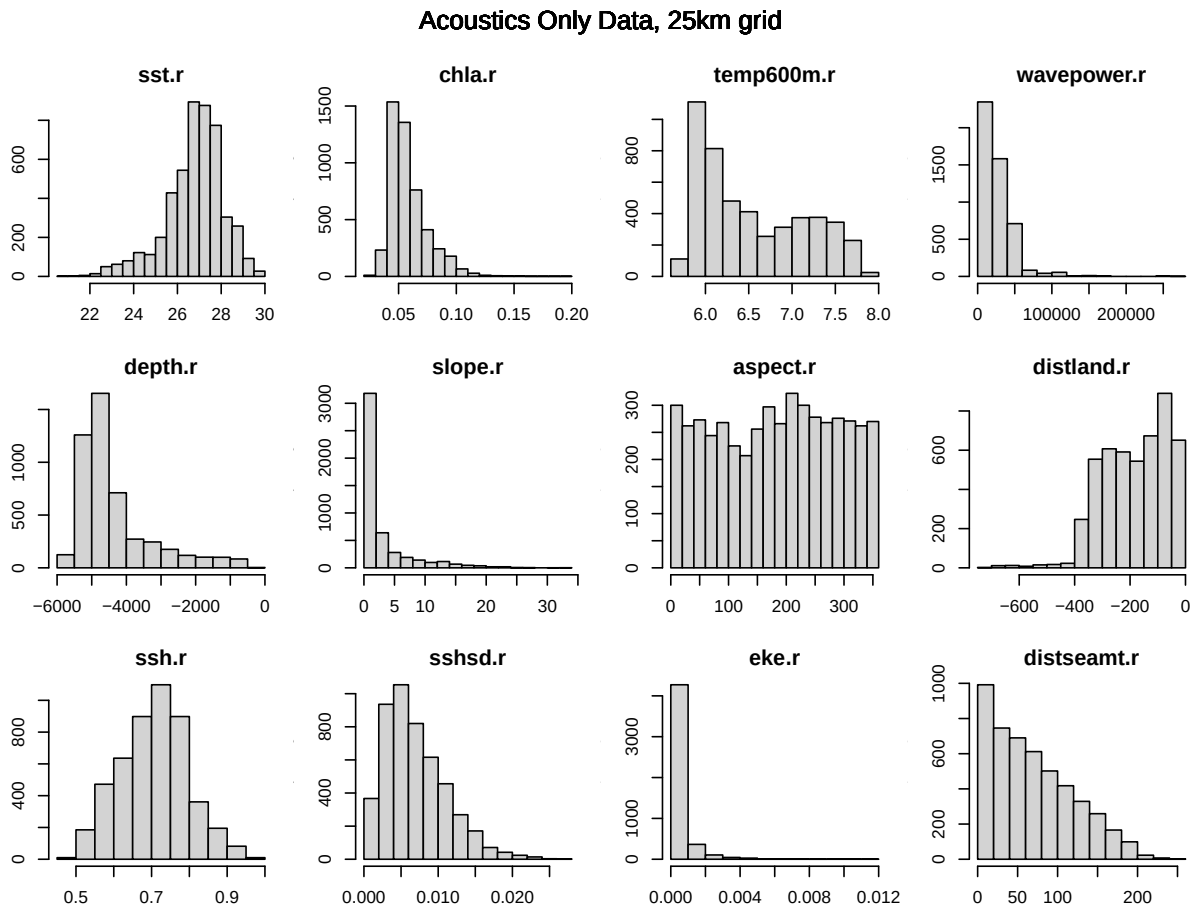
KS tests

I compared the distributions of environmental data between the whales and the absences. Plots are attached in separate powerpoint. In summary, temperature at 600 m, SSH, and chlorophyll were the only variables with significantly different distributions (p-value < 0.05). However, the D statistics were close to zero (D $\sim$ 0.1) for each, indicating that although the distributions were different, they were not that far apart. The plots also show how similar the general shape of the distributions are between where the whales were observed and where they were absent.

Data Visualization

Histograms showing the general distribution of each environmental predictor for the entire dataset.

```
par(mfrow = c(3, 4), mar = c(3, 3, 2, 1), oma = c(0, 0, 3, 1))

dataSet = PmScaled  #raw values

loopVec <- 30:41  #columns from PmScaled to plot

for (j in loopVec) {

    datPlot <- dataSet[, c(1, j)]

    hist(datPlot[, 2], main = colnames(datPlot)[2], ylab = "frequency",
        xlab = "")
    # plot(datPlot[,2], datPlot[,1], ylab = 'Whales', xlab =
    # colnames(datPlot)[2])
    mtext(paste0("Acoustics Only Data, ", gridsize, "km grid"),
        side = 3, line = 1, outer = TRUE, cex = 1, font = 1)

}
```




```
# dev.off()
```

Data Splitting

Split the data into train and test sets

```
require(dplyr)
splitdf <- function(dataframe, seed = NULL) {
    if (!is.null(seed))
        set.seed(seed)
    index <- 1:nrow(dataframe)
    trainindex <- sample(index, trunc(length(index) * 0.7))
    trainset <- dataframe[trainindex, ]
    testset <- dataframe[-trainindex, ]
    list(trainset = trainset, testset = testset)
}

trainForg = NULL
testForg = NULL
# preschk = NULL
seed = 3
```

```r
for (s in c(1641, 1303, 1604, 1705, 1706)) {

    trSub <- filter(PmForage, survey == s)

    # subset for presences and split 70/30
    pres1 <- filter(trSub, pa > 0 & forage == 1)  # & loc == 1) #include all presence data/acoustic enc
    listPres <- splitdf(pres1, seed)  #output is list for train and test
    # preschk = rbind(preschk, pres1) subset for absences and
    # split 70/30
    abs0 <- filter(trSub, pa == 0)
    listAbs <- splitdf(abs0, seed)  #output is list for train and test

    # combine train data for presence and absence
    trainAll <- rbind(listPres$trainset, listAbs$trainset)

    # combine test data for presence and absence
    testAll <- rbind(listPres$testset, listAbs$testset)

    trainForg = rbind(trainForg, trainAll)
    testForg = rbind(testForg, testAll)

    # trainAcOnly$log.effort <- log(trainAcOnly$EffArea)
    # testAcOnly$log.effort <- log(testAcOnly$EffArea)
}
saveRDS(trainForg, here::here(paste0("output/models/", loctype,
    "/data/Train_", gridsize, "km_", loctype2, "_Hunt.rda")))
saveRDS(testForg, here::here(paste0("output/models/", loctype,
    "/data/Test_", gridsize, "km_", loctype2, "_Hunt.rda")))

# nrow(dplyr::filter(trainAcOnly, trainAcOnly$pa >0))
# nrow(dplyr::filter(testAcOnly, testAcOnly$pa >0))
```

Generalized Additive Models

The data are treated as count data, number of sperm whale encounters per cell, and we used the negative binomial distribution to model the response variable for comparison with the Tweedie distribution. We used thin-plate regression splines (the default basis) for the smoothers of the environmental predictors. Each smoother was limited to 3 degrees of freedom (k=3) to reduce overfitting parameters per recommendations from other studies building similar types of cetaceans distribution models.The log of the effort was included as an offset to account for the variation in effort per cell.

25 km spatial scale

- NEGATIVE BINOMIAL DISTRIBUTION
- Knots contrained to k=3 according to literature on cetacean distribution models.
- Automatic term selection is uses an additional penalty term when determining the smoothness of the function ('select' argument = TRUE)..
- We excluded all non-significant variables (alpha=0.05) and refit the models until all variables were significant.
- REML is restricted maximum likelihood used to optimize the parameter estimates.

Load training and test data

```
# seed 1
trainHunt <- readRDS(here::here(paste0("output/models/", loctype,
    "/data/Train_", gridsize, "km_", loctype2, "_Hunt.rda")))
testHunt <- readRDS(here::here(paste0("output/models/", loctype,
    "/data/Test_", gridsize, "km_", loctype2, "_Hunt.rda")))
```

Model Selection

SET 1

Full Models

```
+ does not include spatial smoother
+ does not include slope or aspect due to the variation between left and rightes not include slope or as
```

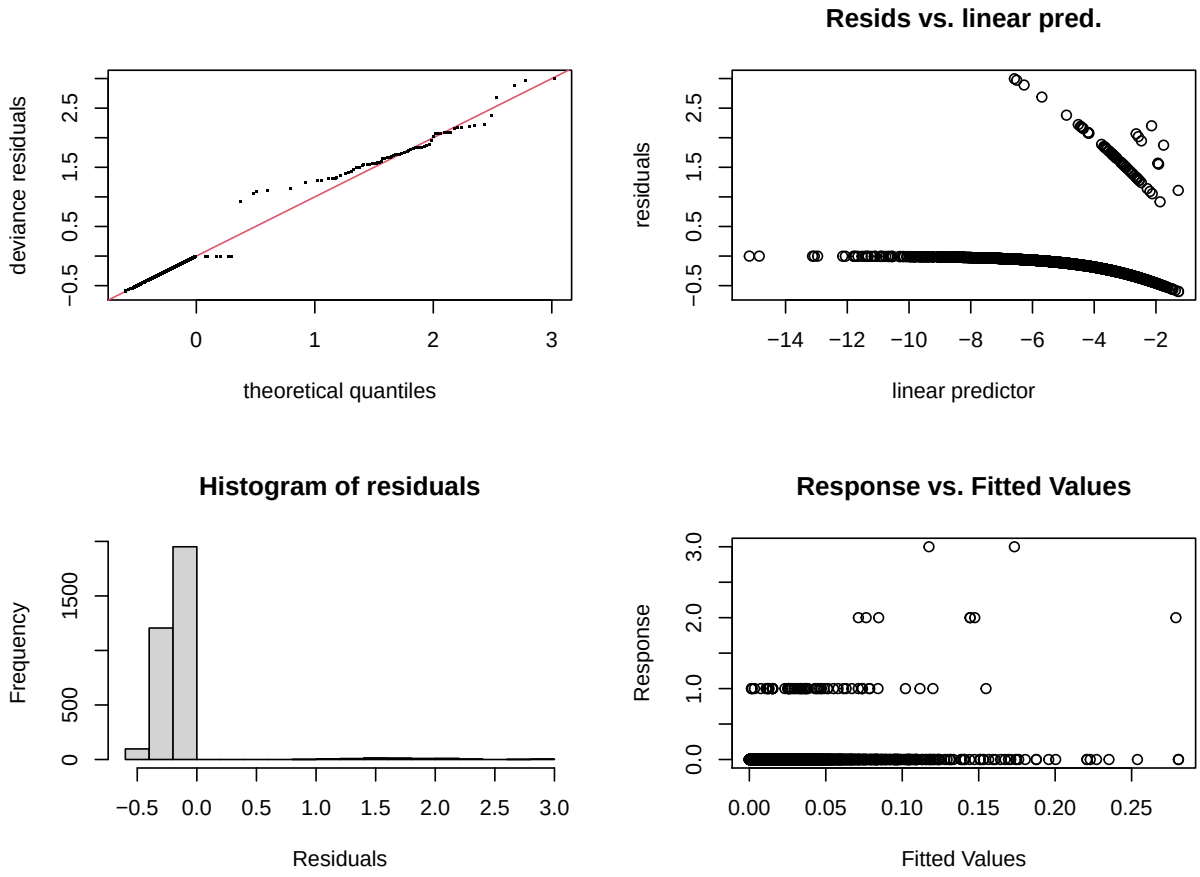
```
require(mgcv)
nbHunt <- gam(pa ~ s(bath_m, k = 3) + s(dist, k = 3) + s(d2smt,
    k = 3) + s(sst, k = 3) + s(chla, k = 3) + s(temp600, k = 3) +
    s(ssh, k = 3) + s(sshsd, k = 3) + s(eke, k = 3) + s(wavepow,
    k = 3) + offset(log.effort), data = trainHunt, family = nb,
    link = "log", select = TRUE, method = "REML")
summary(nbHunt)
```

```
Family: Negative Binomial(0.215)
Link function: log

Formula:
pa ~ s(bath_m, k = 3) + s(dist, k = 3) + s(d2smt, k = 3) + s(sst,
    k = 3) + s(chla, k = 3) + s(temp600, k = 3) + s(ssh, k = 3) +
    s(sshsd, k = 3) + s(eke, k = 3) + s(wavepow, k = 3) + offset(log.effort)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -22.9993     0.1387  -165.8   <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(bath_m)  1.890e-04      2  0.000 0.506523
s(dist)    7.229e-06      2  0.000 0.608563
s(d2smt)   4.459e-02      2  0.044 0.317263
s(sst)     1.577e+00      2 10.560 0.001159 **
s(chla)    9.139e-01      2  9.931 0.000813 ***
s(temp600) 1.634e+00      2 11.480 0.000884 ***
s(ssh)     1.245e-01      2  0.139 0.286352
s(sshsd)   7.662e-01      2  3.304 0.037689 *
s(eke)     6.343e-01      2  1.383 0.138333
s(wavepow) 6.744e-06      2  0.000 1.000000
---
```

```
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

R-sq.(adj) =  0.0335   Deviance explained = 9.91%
-REML = 349.66  Scale est. = 1         n = 3325
```

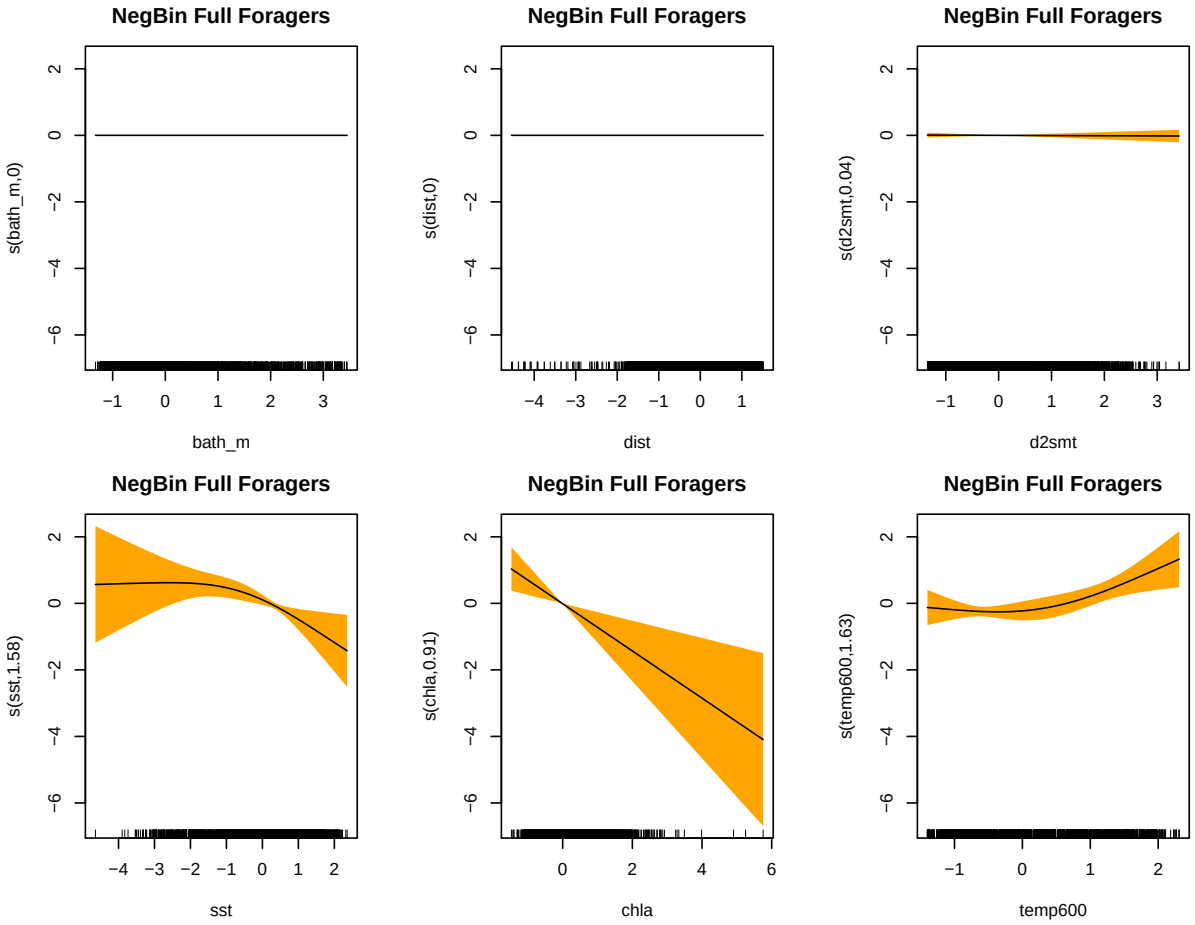
```
AIC(nbHunt)
```

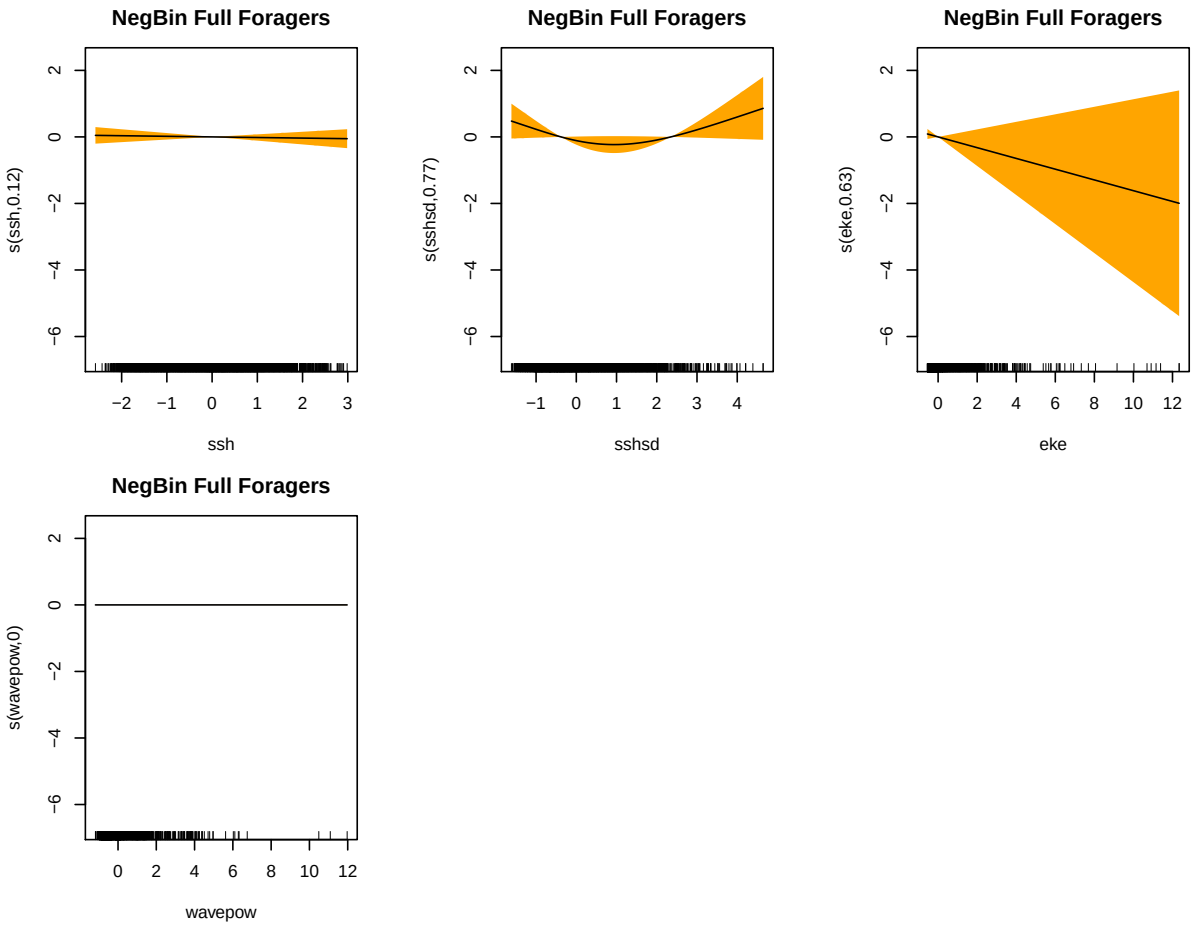
```
[1] 696.7931
```




```
Method: REML   Optimizer: outer newton
full convergence after 15 iterations.
Gradient range [-4.138678e-05,2.13395e-05]
(score 349.6551 & scale 1).
Hessian positive definite, eigenvalue range [8.288702e-07,5.446699].
Model rank =  21 / 21

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(bath_m)  2.00e+00 1.89e-04    0.78  <2e-16 ***
```

```
s(dist)    2.00e+00 7.23e-06    0.84   0.530
s(d2smt)   2.00e+00 4.46e-02    0.82   0.270
s(sst)     2.00e+00 1.58e+00    0.82   0.205
s(chla)    2.00e+00 9.14e-01    0.84   0.520
s(temp600) 2.00e+00 1.63e+00    0.77  <2e-16 ***
s(ssh)     2.00e+00 1.24e-01    0.82   0.105
s(sshsd)   2.00e+00 7.66e-01    0.81   0.055 .
s(eke)     2.00e+00 6.34e-01    0.84   0.440
s(wavepow) 2.00e+00 6.74e-06    0.78  <2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```





Trying the Tweedie for comparison

```
twHunt <- gam(pa ~ s(bath_m, k = 3) + s(dist, k = 3) + s(d2smt,
    k = 3) + s(sst, k = 3) + s(chla, k = 3) + s(temp600, k = 3) +
    s(ssh, k = 3) + s(sshsd, k = 3) + s(eke, k = 3) + s(wavepow,
    k = 3) + offset(log.effort), data = trainHunt, family = tw,
    link = "log", select = TRUE, method = "REML")
summary(twHunt)
```

```
Family: Tweedie(p=1.01)
Link function: log

Formula:
pa ~ s(bath_m, k = 3) + s(dist, k = 3) + s(d2smt, k = 3) + s(sst,
    k = 3) + s(chla, k = 3) + s(temp600, k = 3) + s(ssh, k = 3) +
    s(sshsd, k = 3) + s(eke, k = 3) + s(wavepow, k = 3) + offset(log.effort)

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -23.0296     0.1328  -173.4   <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(bath_m)  3.558e-05      2 0.000 0.648125
s(dist)    4.388e-05      2 0.000 0.455767
s(d2smt)   4.935e-01      2 0.473 0.164867
s(sst)     1.677e+00      2 6.836 0.000232 ***
s(chla)    9.264e-01      2 6.012 0.000265 ***
s(temp600) 1.720e+00      2 8.554 4.67e-05 ***
s(ssh)     4.187e-01      2 0.351 0.193042
s(sshsd)   8.708e-01      2 3.407 0.005133 **
s(eke)     6.688e-01      2 0.833 0.113263
s(wavepow) 1.863e-05      2 0.000 1.000000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =   0.04   Deviance explained = 9.12%
-REML = 265.07  Scale est. = 1.0427    n = 3325
```

```r
AIC(twHunt)
```

```
[1] 3297.434
```

Reduced Models

- Negative Binomial -> higher explained deviance
- Removed non-significant variables:
  - depth
  - distance to land
  - distance to seamount
  - SSH
  - eke
  - wave power

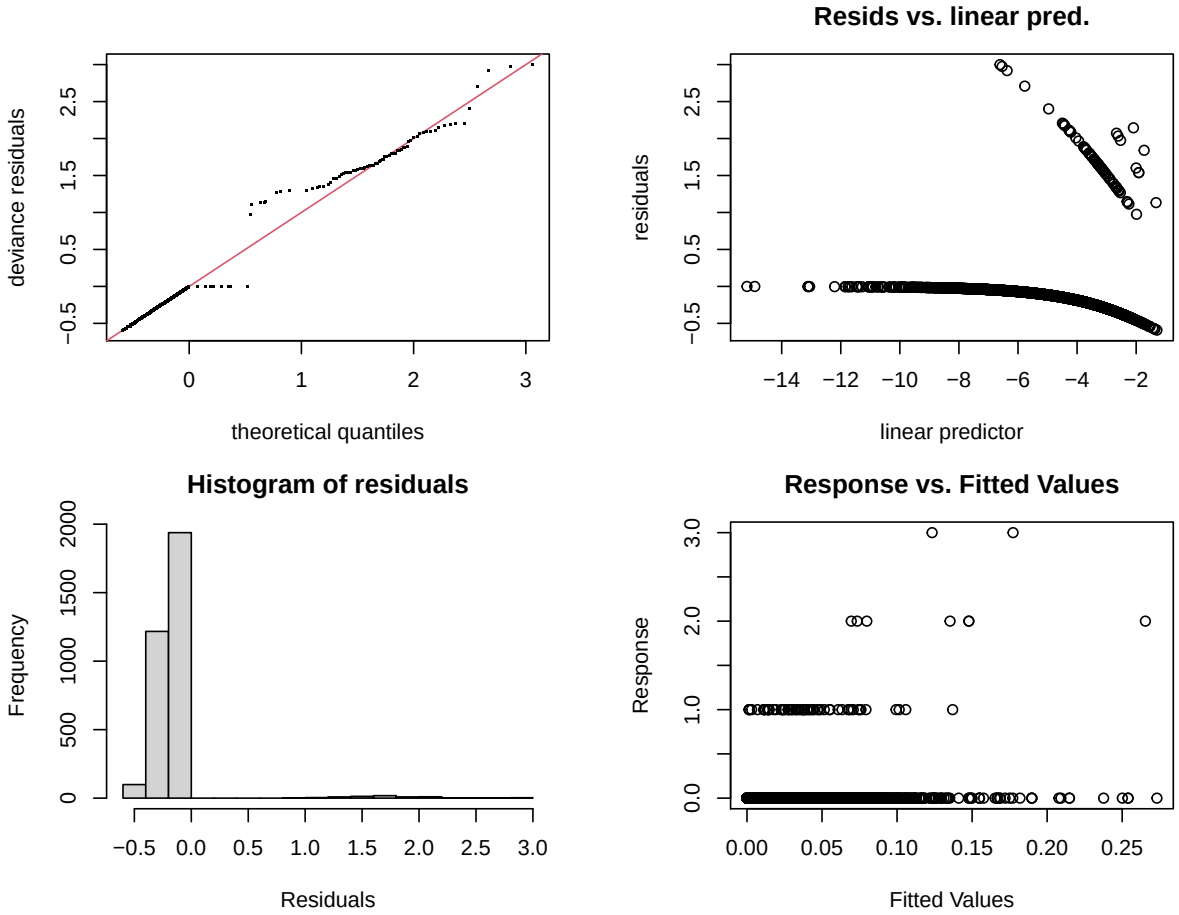
```r
# * Does NOT include sighted acoustic encounters

nbHuntb <- gam(pa ~ s(sst, k = 3) + s(chla, k = 3) + s(temp600,
    k = 3) + s(sshsd, k = 3) + offset(log.effort), data = trainHunt,
    family = nb, link = "log", select = TRUE, method = "REML")
summary(nbHuntb)
```

```
Family: Negative Binomial(0.209)
Link function: log

Formula:
pa ~ s(sst, k = 3) + s(chla, k = 3) + s(temp600, k = 3) + s(sshsd,
    k = 3) + offset(log.effort)

Parametric coefficients:
```

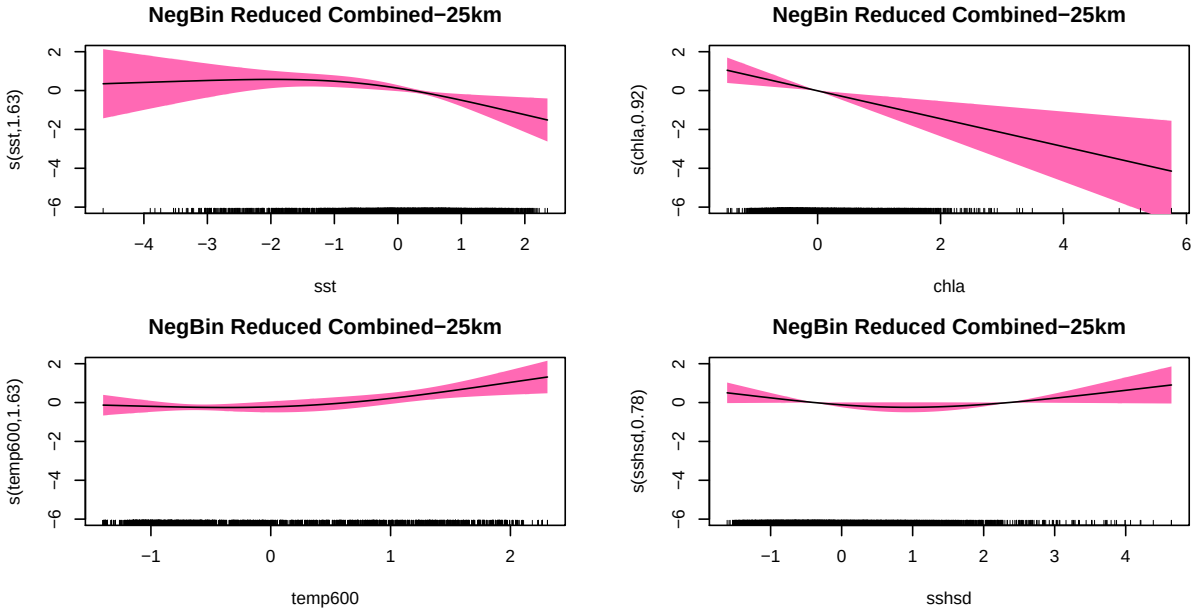
```
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -22.9910     0.1382  -166.4   <2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Approximate significance of smooth terms:
              edf Ref.df Chi.sq  p-value
s(sst)     1.6254      2 10.741 0.001321 **
s(chla)    0.9161      2 10.236 0.000697 ***
s(temp600) 1.6274      2 11.394 0.000923 ***
s(sshsd)   0.7813      2  3.599 0.031790 *
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

R-sq.(adj) =  0.032   Deviance explained =  9.4%
-REML = 349.93  Scale est. = 1         n = 3325
```

```
AIC(nbHuntb)  #slightly lower AIC
```

```
[1] 696.6479
```



```
Method: REML   Optimizer: outer newton
full convergence after 13 iterations.
Gradient range [-6.508201e-06,3.240972e-06]
(score 349.9333 & scale 1).
Hessian positive definite, eigenvalue range [5.12901e-06,5.535083].
Model rank =  9 / 9

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(sst)     2.000 1.625    0.82    0.24
s(chla)    2.000 0.916    0.84    0.54
s(temp600) 2.000 1.627    0.77  <2e-16 ***
s(sshsd)   2.000 0.781    0.81    0.06 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



SET 2

Full Models: Includes s(Longitude,Latitude)

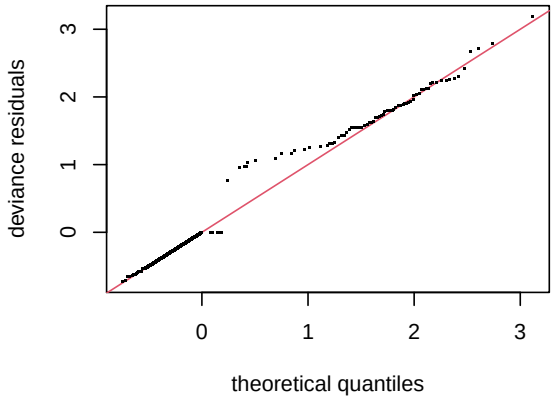
Includes 2D Lat-Lon smoother to account for spatial structure in the data and fit the spatial variation not explained by the other predictors

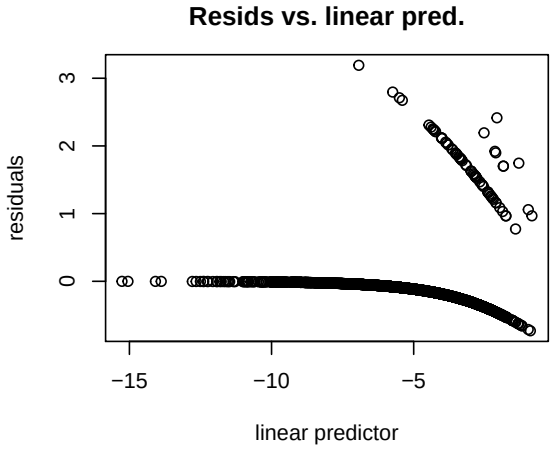
```
nbHuntLL <- gam(pa ~ s(Longitude, Latitude) + s(bath_m, k = 3) +
    s(dist, k = 3) + s(d2smt, k = 3) + s(sst, k = 3) + s(chla,
    k = 3) + s(temp600, k = 3) + s(ssh, k = 3) + s(sshsd, k = 3) +
    s(eke, k = 3) + s(wavepow, k = 3) + offset(log.effort), data = trainHunt,
    family = nb, link = "log", select = TRUE, method = "REML")
summary(nbHuntLL)
```

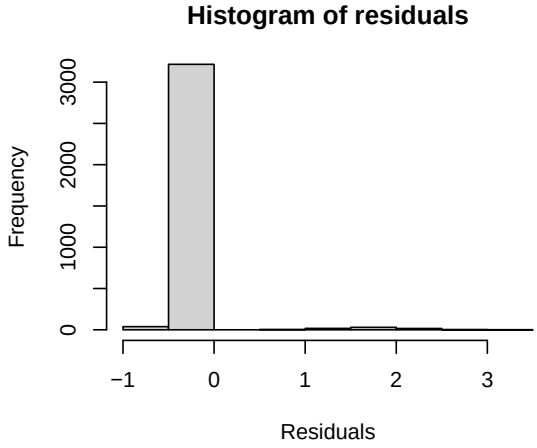
```
Family: Negative Binomial(0.336)
Link function: log

Formula:
pa ~ s(Longitude, Latitude) + s(bath_m, k = 3) + s(dist, k = 3) +
    s(d2smt, k = 3) + s(sst, k = 3) + s(chla, k = 3) + s(temp600,
    k = 3) + s(ssh, k = 3) + s(sshsd, k = 3) + s(eke, k = 3) +
    s(wavepow, k = 3) + offset(log.effort)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -23.1897     0.1538  -150.7   <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(Longitude,Latitude) 9.227e+00     29 40.180 9.67e-08 ***
s(bath_m)             6.055e-05      2  0.000  0.56098
s(dist)               1.069e-04      2  0.000  0.30197
s(d2smt)              9.713e-05      2  0.000  0.46633
s(sst)                9.454e-01      2  3.375  0.03437 *
s(chla)               8.916e-01      2  7.476  0.00279 **
s(temp600)            7.735e-01      2  3.325  0.02338 *
s(ssh)                6.952e-05      2  0.000  0.51109
s(sshsd)              8.420e-01      2  5.241  0.01207 *
s(eke)                4.675e-01      2  0.700  0.21651
s(wavepow)            3.730e-05      2  0.000  0.80305
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0624   Deviance explained = 18.3%
-REML = 341.55  Scale est. = 1         n = 3325
```

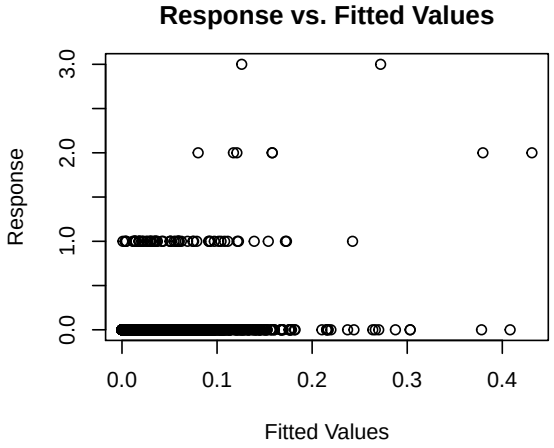
```
AIC(nbHuntLL)
```

```
[1] 680.0982
```



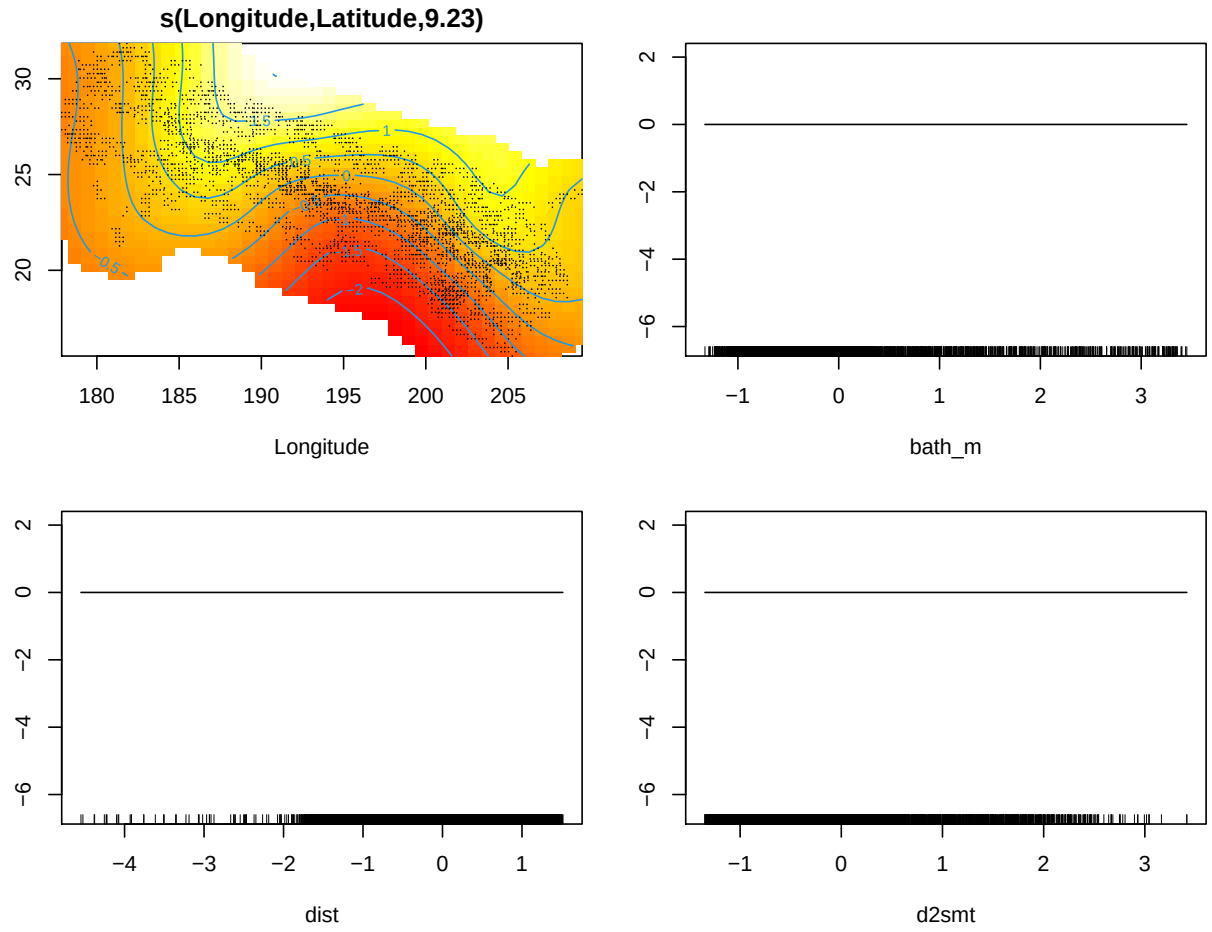


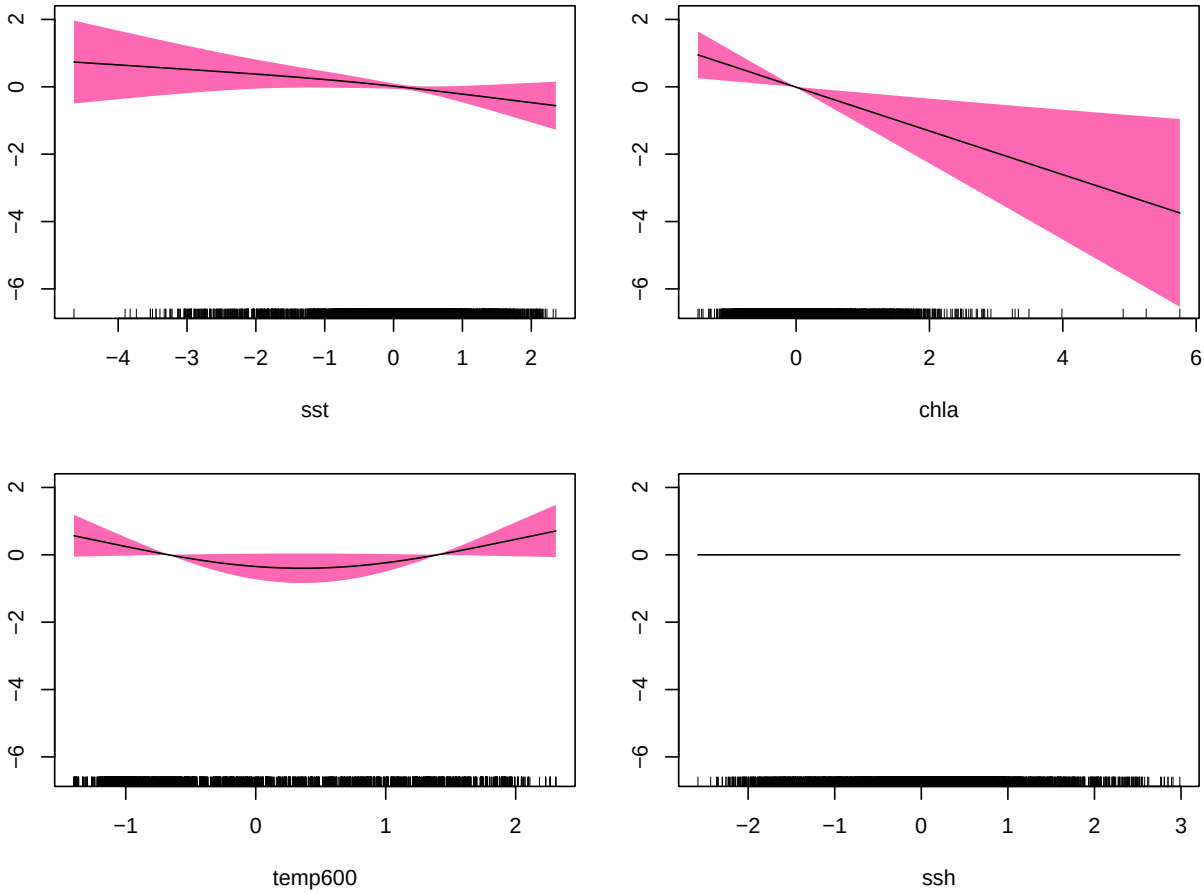


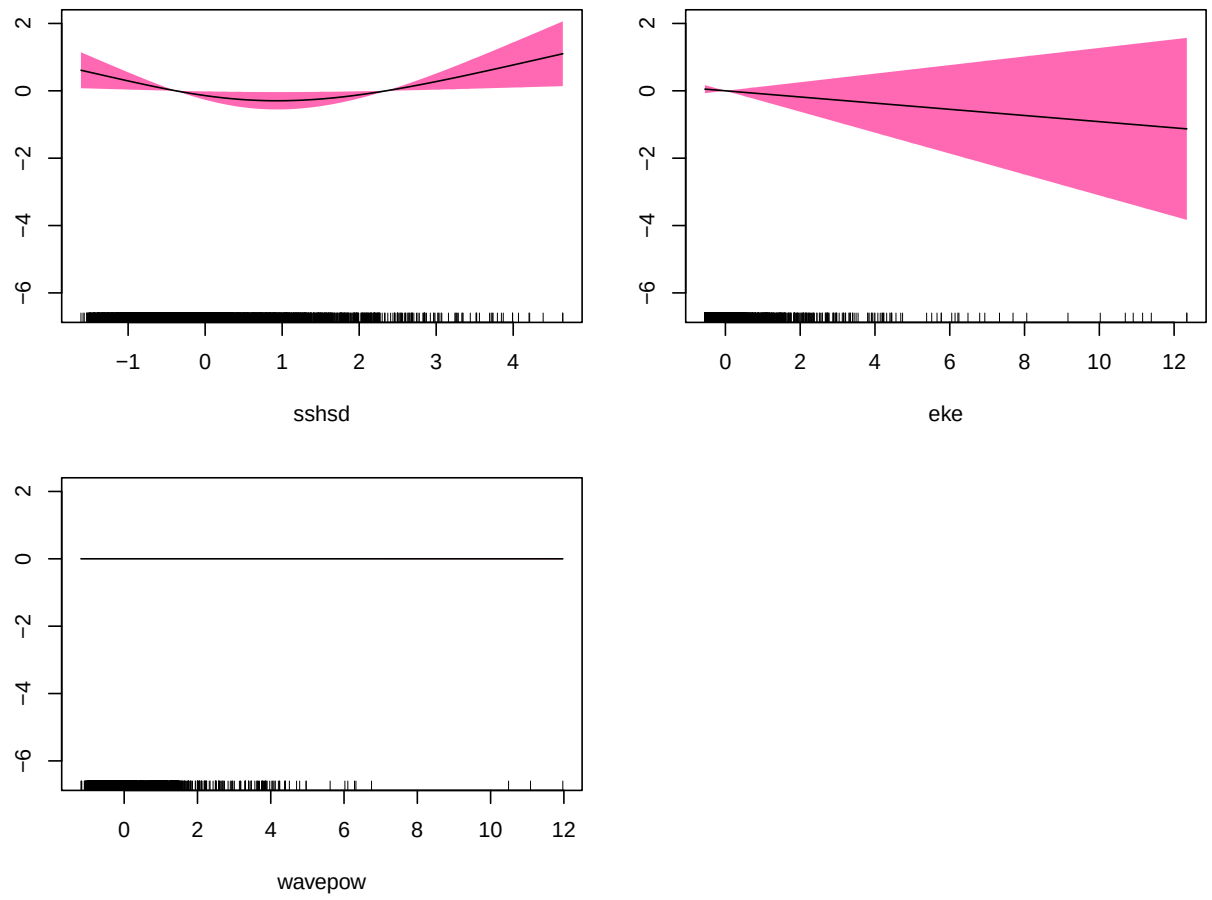



```
Method: REML   Optimizer: outer newton
full convergence after 12 iterations.
Gradient range [-0.0006049324,0.0001165966]
(score 341.5505 & scale 1).
Hessian positive definite, eigenvalue range [5.439635e-06,4.223997].
Model rank =  50 / 50

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(Longitude,Latitude) 2.90e+01 9.23e+00    0.82   0.065 .
s(bath_m)             2.00e+00 6.06e-05    0.80   0.010 **
s(dist)               2.00e+00 1.07e-04    0.86   0.650
s(d2smt)              2.00e+00 9.71e-05    0.85   0.300
s(sst)                2.00e+00 9.45e-01    0.84   0.190
s(chla)               2.00e+00 8.92e-01    0.85   0.390
s(temp600)            2.00e+00 7.74e-01    0.80  <2e-16 ***
s(ssh)                2.00e+00 6.95e-05    0.84   0.170
s(sshsd)              2.00e+00 8.42e-01    0.83   0.105
s(eke)                2.00e+00 4.68e-01    0.85   0.315
s(wavepow)            2.00e+00 3.73e-05    0.80  <2e-16 ***
---
```

```
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```




2
0
−2
−4
−6
−4
−3
−2
−1
0
1
2
sst
2
0
−2
−4
−6
0
2
4
6
chla
2
0
−2
−4
−6
−1
0
1
2
temp600
2
0
−2
−4
−6
−2
−1
0
1
2
3
ssh



Checking Tweedie for comparison

```
twHuntLL <- gam(pa ~ s(Longitude, Latitude) + s(bath_m, k = 3) +
    s(dist, k = 3) + s(d2smt, k = 3) + s(sst, k = 3) + s(chla,
    k = 3) + s(temp600, k = 3) + s(ssh, k = 3) + s(sshsd, k = 3) +
    s(eke, k = 3) + s(wavepow, k = 3) + offset(log.effort), data = trainHunt,
    family = tw, link = "log", select = TRUE, method = "REML")
summary(twHuntLL)
```

```
Family: Tweedie(p=1.01)
Link function: log

Formula:
pa ~ s(Longitude, Latitude) + s(bath_m, k = 3) + s(dist, k = 3) +
    s(d2smt, k = 3) + s(sst, k = 3) + s(chla, k = 3) + s(temp600,
    k = 3) + s(ssh, k = 3) + s(sshsd, k = 3) + s(eke, k = 3) +
    s(wavepow, k = 3) + offset(log.effort)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

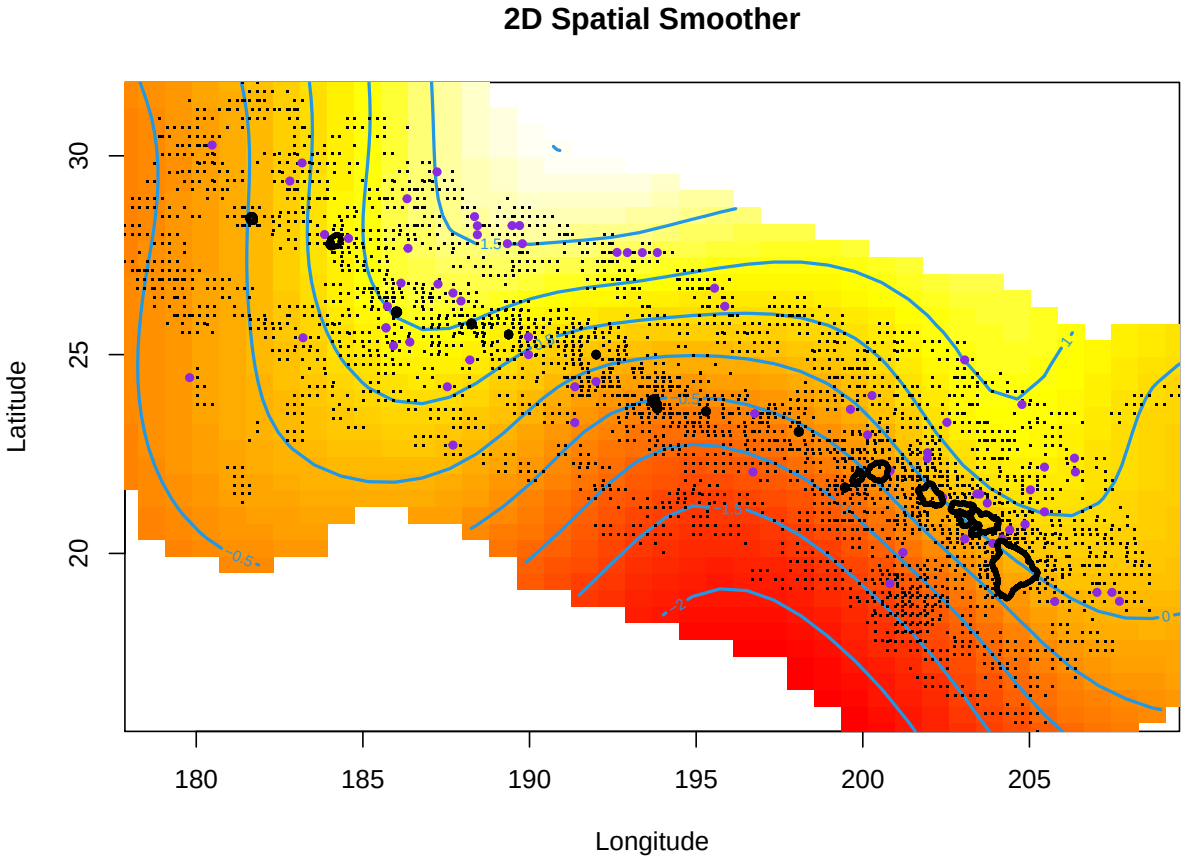


Figure 1: Purple dots represent acoustically detected encounters. Black dots are all data points(grid centroids)

```
(Intercept) -23.2218     0.1516  -153.2   <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(Longitude,Latitude) 1.022e+01     29 1.807 3.29e-10 ***
s(bath_m)             9.254e-05      2 0.000  0.55626
s(dist)               6.713e-05      2 0.000  0.31244
s(d2smt)              3.431e-01      2 0.262  0.19691
s(sst)                1.157e+00      2 2.074  0.02535 *
s(chla)               9.037e-01      2 4.457  0.00112 **
s(temp600)            8.432e-01      2 2.661  0.00581 **
s(ssh)                1.632e-04      2 0.000  0.54736
s(sshsd)              9.013e-01      2 4.633  0.00123 **
s(eke)                5.644e-01      2 0.535  0.16338
s(wavepow)            8.450e-05      2 0.000  0.51516
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0737   Deviance explained =   17%
-REML = 253.84  Scale est. = 1.0402    n = 3325
```

```
AIC(twHuntLL)
```

```
[1] 3456.021
```

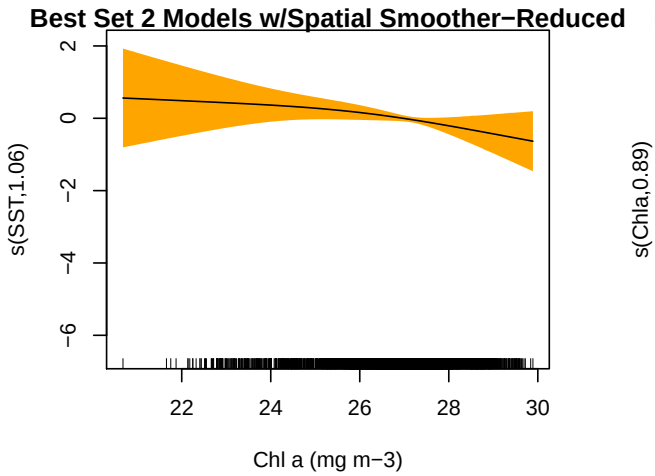
REDUCE MODEL PARAMETERS

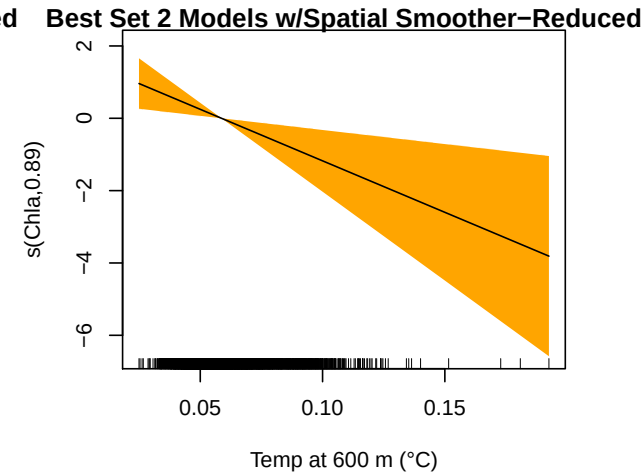
- Negative Binomial: higher explained deviance, lower AIC than Tweedie
- Keep:
  - Lon, Lat
  - SST
  - chlorophyll
  - temp at 600 m
  - SSHsd

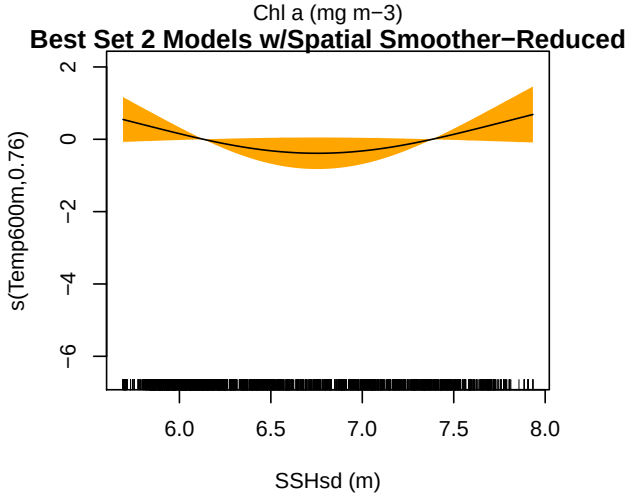
```
colnames(trainHunt)[30] <- "SST"
colnames(trainHunt)[31] <- "Chla"
colnames(trainHunt)[32] <- "Temp600m"
colnames(trainHunt)[39] <- "SSHsd"
nbHuntLLb2 <- gam(pa ~ s(Longitude, Latitude) + s(SST, k = 3) +
    s(Chla, k = 3) + s(Temp600m, k = 3) + s(SSHsd, k = 3) + offset(log.effort),
    data = trainHunt, family = nb, link = "log", select = TRUE,
    method = "REML")
summary(nbHuntLLb2)
```

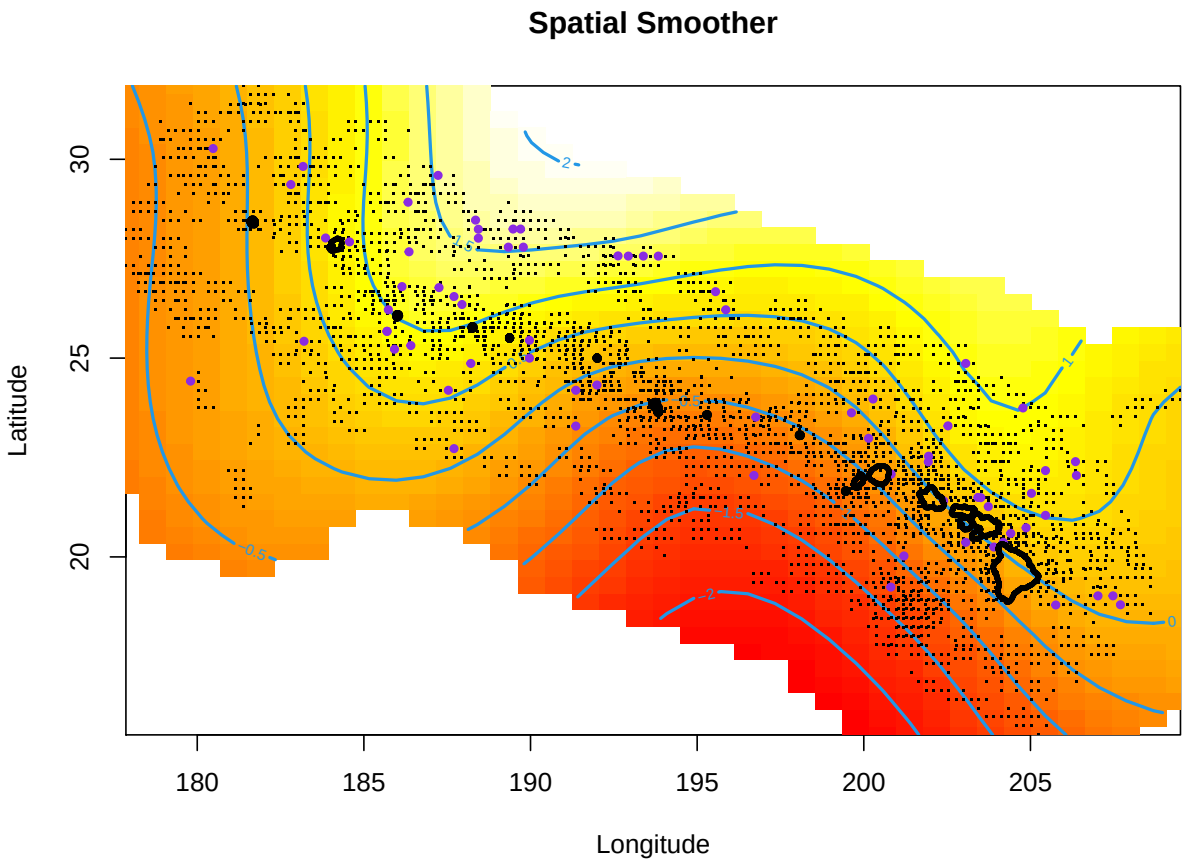
```
Family: Negative Binomial(0.332)
Link function: log

Formula:
```

```
pa ~ s(Longitude, Latitude) + s(SST, k = 3) + s(Chla, k = 3) +
    s(Temp600m, k = 3) + s(SSHsd, k = 3) + offset(log.effort)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -23.188      0.154  -150.6   <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(Longitude,Latitude) 9.1825     29 40.493 7.57e-08 ***
s(SST)                1.0574      2  3.322  0.04214 *
s(Chla)               0.8921      2  7.613  0.00247 **
s(Temp600m)           0.7631      2  3.143  0.02616 *
s(SSHsd)              0.8467      2  5.444  0.01076 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0614   Deviance explained = 18.1%
-REML = 341.61  Scale est. = 1         n = 3325
```









Conclusions

Predict Test Data

```
require(magrittr)
require(dplyr)

#### For twCombc, no spatial smoother ####
nbTrainFinal <- trainHunt %>% mutate(resid = resid(nbHuntb),
    predict = predict(nbHuntb))
predTrain <- predict.gam(nbHuntb, type = "response", se.fit = TRUE)  #calculate MSE for these to compar
nbTrainFinal$fit <- predTrain$fit
nbTrainFinal$se.fit <- predTrain$se.fit
# using scale of 0,1,2 makes this hard to interpret
nbMSEtrain <- mean((nbTrainFinal$pa - nbTrainFinal$fit)^2)  #MSE


# mean(abs((nbTrainFinal$pa - nbTrainFinal$fit))) #Mean
```

```r
# absolute error Calculate MSE AFTER transforming the
# predictions back to the same scale as the observed data
colnames(testHunt)[30] <- "SST"
colnames(testHunt)[31] <- "Chla"
colnames(testHunt)[32] <- "Temp600m"
colnames(testHunt)[39] <- "SSHsd"
nbPred <- predict.gam(nbHuntb, newdata = testHunt, type = "response",
    se.fit = TRUE)
nbTestFinal <- data.frame(testHunt, fit = nbPred$fit, se.fit = nbPred$se.fit)
nbMSEtest <- mean((nbTestFinal$pa - nbTestFinal$fit)^2)  #MSE

# mean(abs((testFinal$pa - testFinal$fit))) #Mean absolute
# error


#### For nbCombcLL, with spatial smoother #### pulling the
#### prediction and residual data from the model
nbTrainLL <- trainHunt %>% mutate(resid = resid(nbHuntLLb2),
    predict = predict(nbHuntLLb2))
predTrainLL <- predict.gam(nbHuntLLb2, type = "response")  #calculate MSE for these to compare with tes
nbTrainLL$fit <- predTrainLL

# using scale of 0,1,2 makes this hard to interpret
nbMSEtrainLL <- mean((nbTrainLL$pa - nbTrainLL$fit)^2)  #MSE
# mean(abs((nbTrainFinal$pa - nbTrainFinal$fit))) #Mean
# absolute error Calculate MSE AFTER transforming the
# predictions back to the same scale as the observed data

nbPredLL <- predict.gam(nbHuntLLb2, newdata = testHunt, type = "response",
    se.fit = TRUE)
nbTestLL <- data.frame(testHunt, fit = nbPredLL$fit, se.fit = nbPredLL$se.fit)
nbMSEtestLL <- mean((nbTestLL$pa - nbTestLL$fit)^2)  #MSE

# mean(abs((testFinal$pa - testFinal$fit))) #Mean absolute
# error

# AIC
nbAIC <- AIC(nbHuntb)
nbAICLL <- AIC(nbHuntLLb2)

# Explained Deviance
nbExpDev = round(((nbHuntb$null.deviance - nbHuntb$deviance)/nbHuntb$null.deviance) *
    100, 2)
nbExpDevLL = round(((nbHuntLLb2$null.deviance - nbHuntLLb2$deviance)/nbHuntLLb2$null.deviance) *
    100, 2)
```

```r
# make summary table of metrics

table = matrix(NA, nrow = 2, ncol = 5)
colnames(table) = c("Best Models", "ExpDev", "AIC", "MSEtrain",
    "MSEtest")

# enter info by row
```

```
table[1, ] <- c("nbCombb", paste0(nbExpDev, "%"), round(nbAIC,
    2), round(nbMSEtrain, 3), round(nbMSEtest, 3))

table[2, ] <- c("nbCombLLb2 (w/ s(Lon,Lat))", paste0(nbExpDevLL,
    "%"), round(nbAICLL, 2), round(nbMSEtrainLL, 3), round(nbMSEtestLL,
    3))
require(knitr)
kable(table, caption = "Negative Binomial Model Summary Metrics")
```

Table 1: Negative Binomial Model Summary Metrics

Best Models	ExpDev	AIC	MSEtrain	MSEtest
nbCombb	9.4%	696.65	0.031	0.029
nbCombLLb2 (w/ s(Lon,Lat))	18.14%	679.84	0.03	0.029