GAMs - Acoustics Only - 25 km

Yvonne Barkley

9/26/2020

Technical coding questions:

- what's the easiest way to make the partial residual plots using the raw data when I'm running the models with scaled data?
- I need help with the predict() function for predicting the test data. I thought it was straight-forward but I'm misunderstanding how it works.

Load libraries

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

Research question:

What environmental variables characterize sperm whale habitat?

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

Include more details about what to expect in this document

Load universal variables

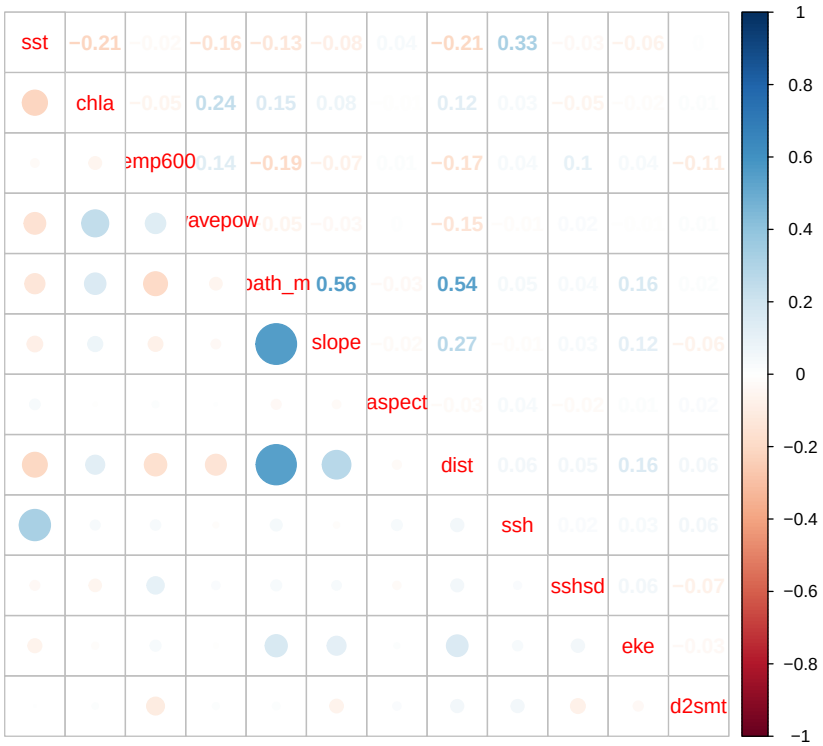
```
# Values used for file and directory names
survey = "AllSurveys"
gridsize = 25
loctype = "AcOnly"
loctype2 = "Ac"
```

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 <= 10)  #some outliers in a handful of absences
```

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
```

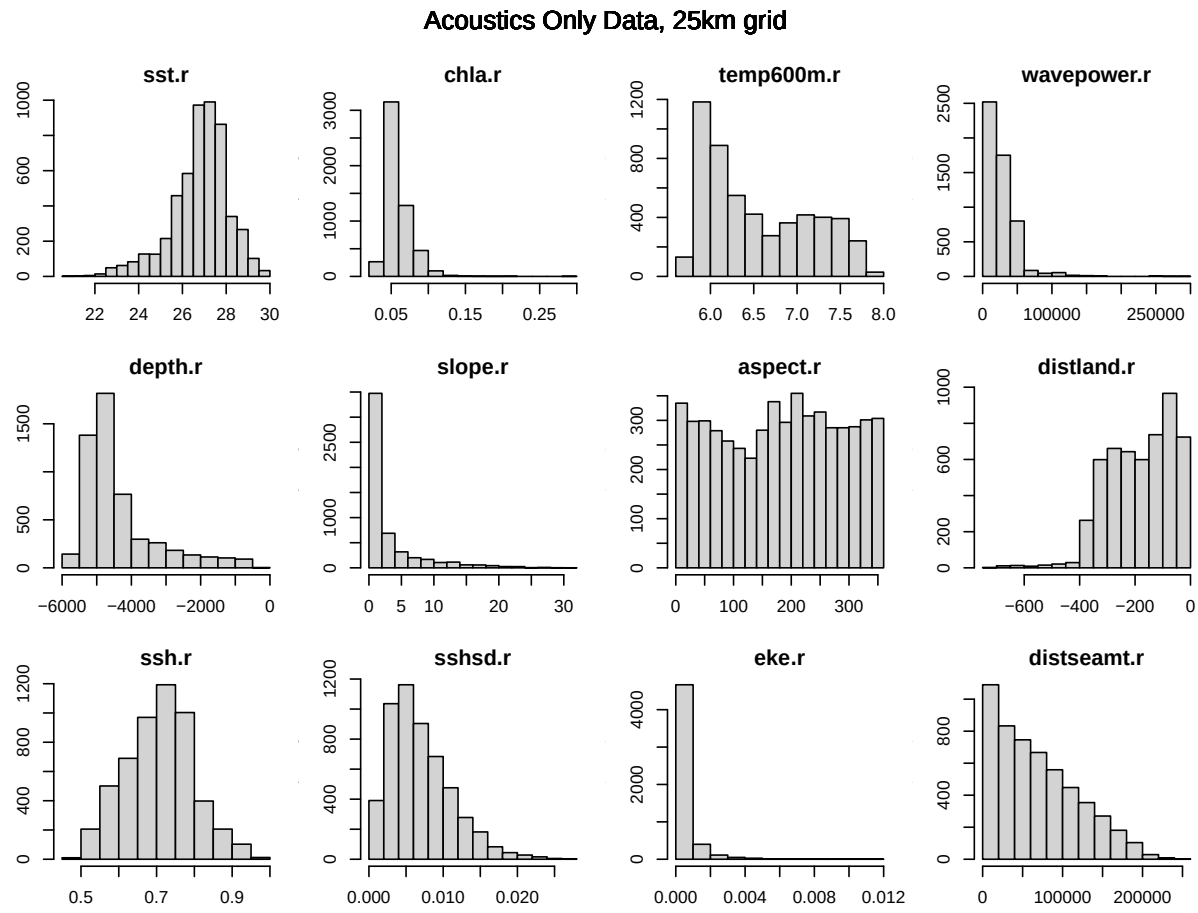
KS tests

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()
```

```
# Let's take the log of the more skewed variables * chla *
# eke * wave power take the log of some variables that are
# more skewed
PmScaled$chla.log <- log(PmScaled$chla.r)
PmScaled$eke.log <- log(PmScaled$eke.r)
PmScaled$wavepow.log <- log(PmScaled$wavepower.r)


# plot them
dataSet = PmScaled  #raw values

loopVec <- 57:59  #columns from PmScaled to plot

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

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, ", surveynum, ", ", gridsize,
        "km grid"), side = 3, line = 1, outer = TRUE, cex = 1,
        font = 1)

}
```

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)
}

trainAcOnly = NULL
testAcOnly = NULL
seed = 1

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

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

    # subset for presences and split 70/30
    pres1 <- filter(trSub, pa > 0 & sid == 999)  # & loc == 1) #for S999 versions
    listPres <- splitdf(pres1, seed)  #output is list for train and test

    # 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)

    trainAcOnly = rbind(trainAcOnly, trainAll)
    testAcOnly = rbind(testAcOnly, testAll)

    # trainAcOnly$log.effort <- log(trainAcOnly$EffArea)
    # testAcOnly$log.effort <- log(testAcOnly$EffArea)
}
saveRDS(trainAcOnly, here::here(paste0("output/models/", loctype,
    "/data/Train_", gridsize, "km_", loctype2, "_S999b.rda")))
```

```
saveRDS(testAcOnly, here::here(paste0("output/models/", loctype,
    "/data/Test_", gridsize, "km_", loctype2, "_S999b.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 Tweedie distribution since it has been shown to work well when fewer positive responses exist within the data. 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

- 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
trainS999 <- readRDS(here::here(paste0("output/models/", loctype,
    "/data/Train_", gridsize, "km_", loctype2, "_S999b.rda")))
testS999 <- readRDS(here::here(paste0("output/models/", loctype,
    "/data/Test_", gridsize, "km_", loctype2, "_S999b.rda")))
```

Model Selection

FULL MODEL

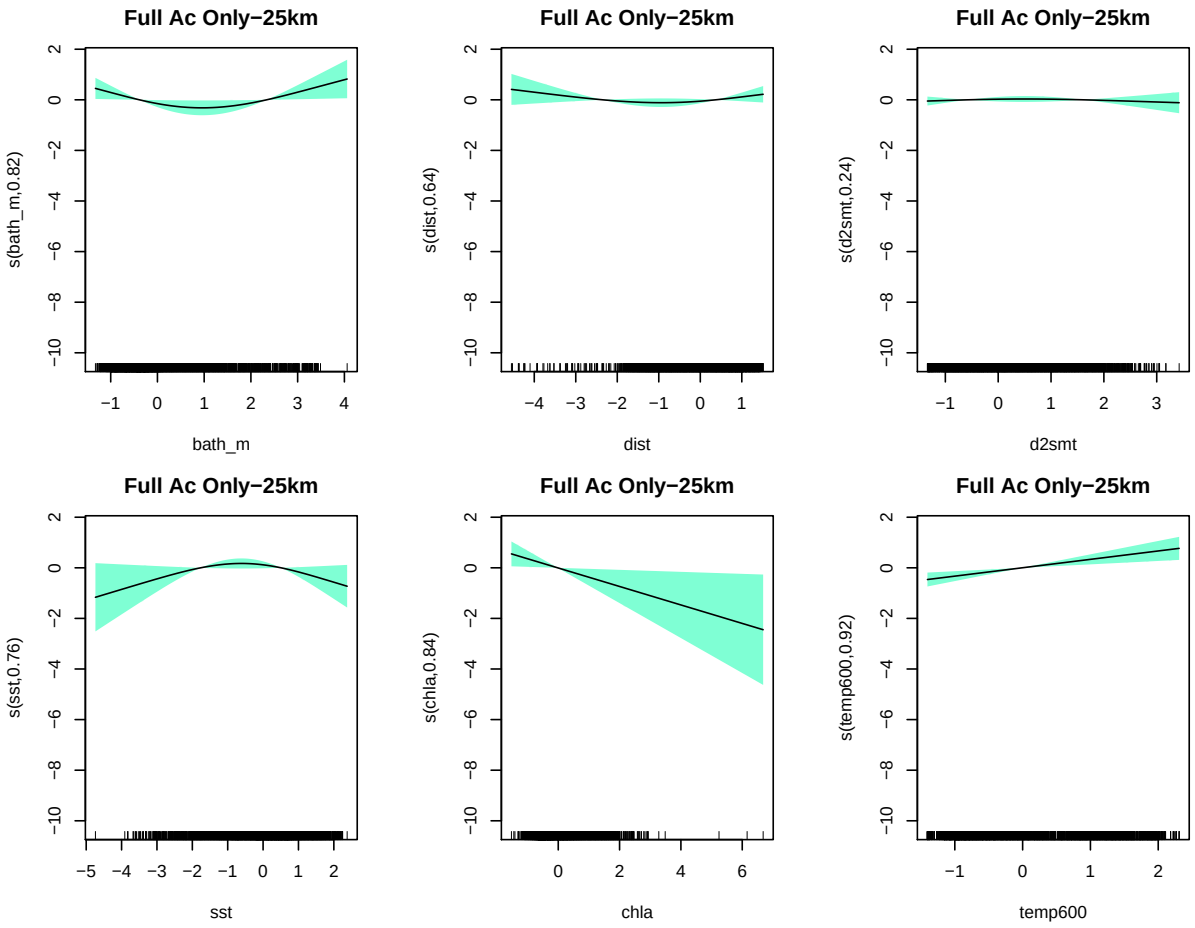
Estimate the smoothing parameters for each predictor variable using restricted maximum likelihood (method = 'REML') + does not include spatial smoother + does not include slope or aspect due to the variation between left and right

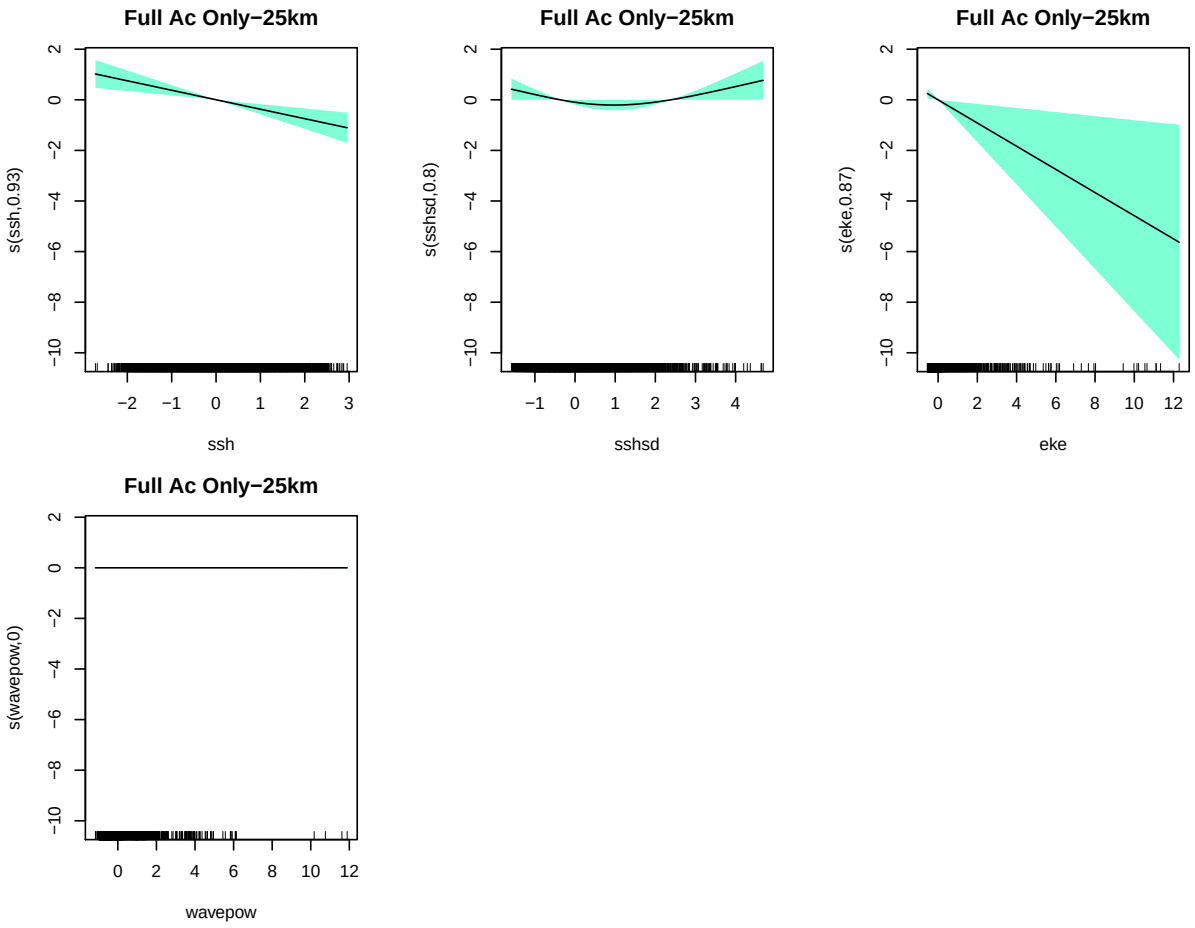
Using the scaled values

```
# * Does NOT include sighted acoustic encounters OR spatial
# smoother
require(mgcv)
twS999 <- 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 = trainS999, family = tw,
    link = "log", select = TRUE, method = "REML")
summary(twS999)
```

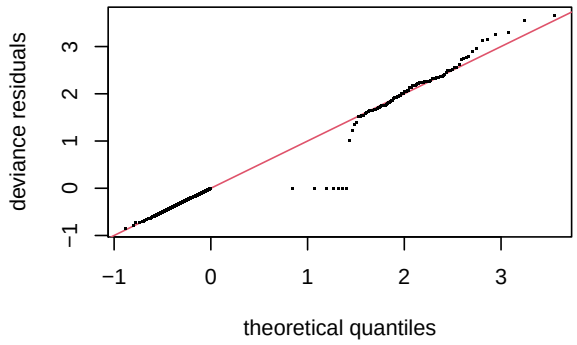
```
Family: Tweedie(p=1.01)
Link function: log

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) -22.9376     0.1157  -198.3   <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)  8.239e-01      2 2.342 0.016406 *
s(dist)    6.391e-01      2 0.907 0.089417 .
s(d2smt)   2.366e-01      2 0.152 0.254782
s(sst)     7.598e-01      2 1.492 0.045248 *
s(chla)    8.409e-01      2 2.518 0.013254 *
s(temp600) 9.193e-01      2 5.684 0.000324 ***
s(ssh)     9.328e-01      2 6.870 0.000122 ***
s(sshsd)   7.977e-01      2 2.014 0.024321 *
s(eke)     8.695e-01      2 2.945 0.008894 **
s(wavepow) 9.707e-05      2 0.000 0.634825
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0434   Deviance explained = 9.45%
-REML = 303.69  Scale est. = 1.0402    n = 3659
```

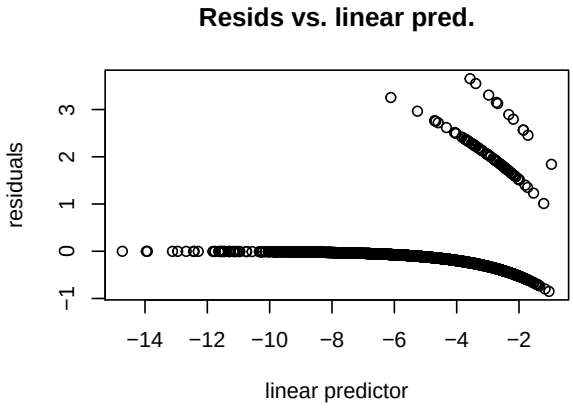
```
# + s(slope, k=3) + s(aspect, k=3) # removed 9/27
```

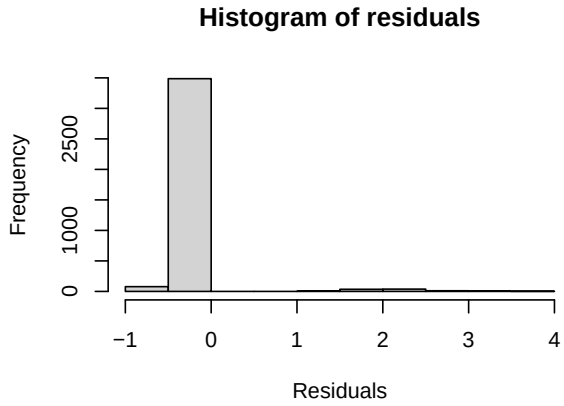

Full Ac Only−25km
s(bath_m,0.82)
bath_m
Full Ac Only−25km
s(dist,0.64)
dist
Full Ac Only−25km
s(d2smt,0.24)
d2smt
Full Ac Only−25km
s(sst,0.76)
sst
Full Ac Only−25km
s(chla,0.84)
chla
Full Ac Only−25km
s(temp600,0.92)
temp600

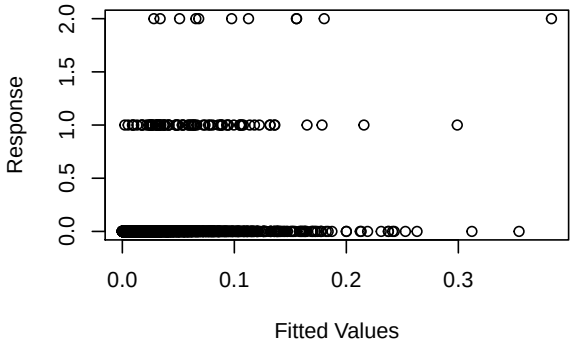


MODEL DIAGNOSTICS The stripe at the bottom left of the residuals vs. fitted values (linear predictor) corresponds to the zeros.









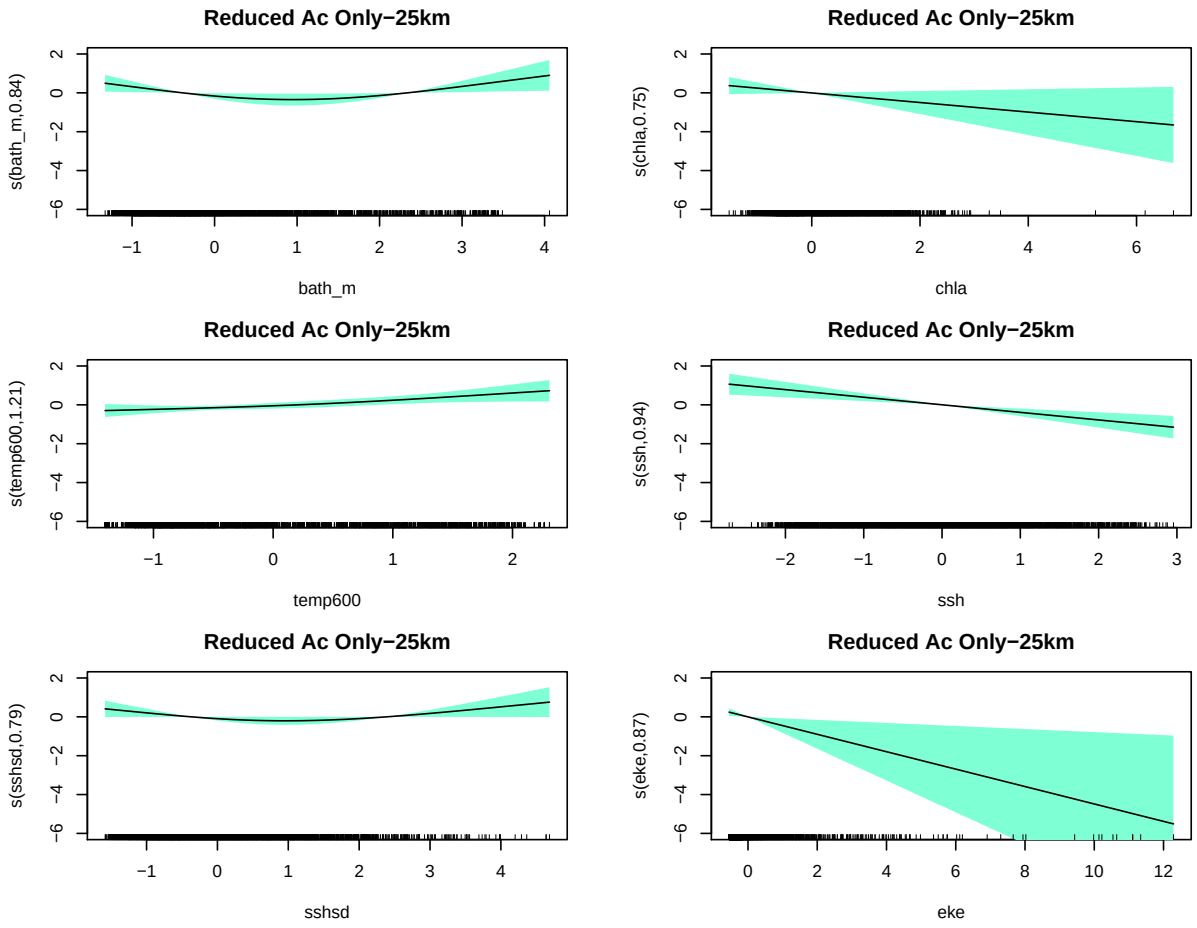

```
Method: REML   Optimizer: outer newton
full convergence after 26 iterations.
Gradient range [-0.0006124275,8.028359e-05]
(score 303.6892 & scale 1.040183).
Hessian positive definite, eigenvalue range [1.189187e-05,10499.48].
Model rank =  21 / 21

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 8.24e-01    0.86   0.050 *
s(dist)    2.00e+00 6.39e-01    0.89   0.585
s(d2smt)   2.00e+00 2.37e-01    0.89   0.625
s(sst)     2.00e+00 7.60e-01    0.85   0.010 **
s(chla)    2.00e+00 8.41e-01    0.88   0.280
s(temp600) 2.00e+00 9.19e-01    0.76  <2e-16 ***
s(ssh)     2.00e+00 9.33e-01    0.87   0.185
s(sshsd)   2.00e+00 7.98e-01    0.86   0.015 *
s(eke)     2.00e+00 8.70e-01    0.87   0.140
s(wavepow) 2.00e+00 9.71e-05    0.81  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

REDUCE MODEL PARAMETERS

- Removed non-significant variables:
  - distance to land
  - distance to seamount
  - sst
  - wave power

```
# * Does NOT include sighted acoustic encounters

twS999b <- gam(pa ~ s(bath_m, k = 3) + s(chla, k = 3) + s(temp600,
    k = 3) + s(ssh, k = 3) + s(sshsd, k = 3) + s(eke, k = 3) +
    offset(log.effort), data = trainS999, family = tw, link = "log",
    select = TRUE, method = "REML")
summary(twS999b)
```

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

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

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -22.9035     0.1129  -202.8   <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)  0.8395      2 2.615  0.01201 *
s(chla)    0.7504      2 1.422  0.05080 .
s(temp600) 1.2070      2 4.280  0.00229 **
s(ssh)     0.9405      2 7.826 4.51e-05 ***
s(sshsd)   0.7922      2 1.943  0.02654 *
s(eke)     0.8698      2 2.939  0.00933 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0393   Deviance explained = 8.49%
-REML =  304.8  Scale est. = 1.0404    n = 3659
```

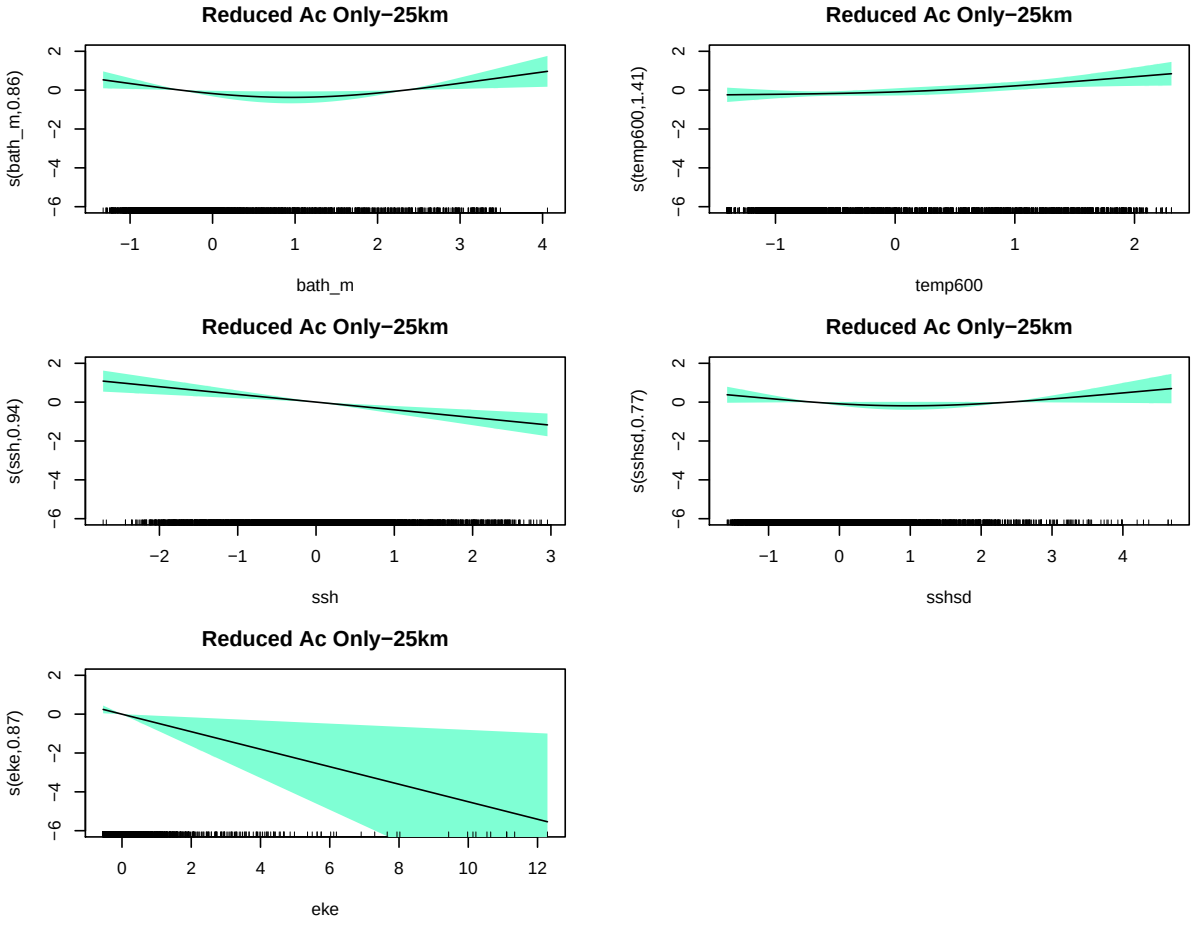


- Remove chlorophyll

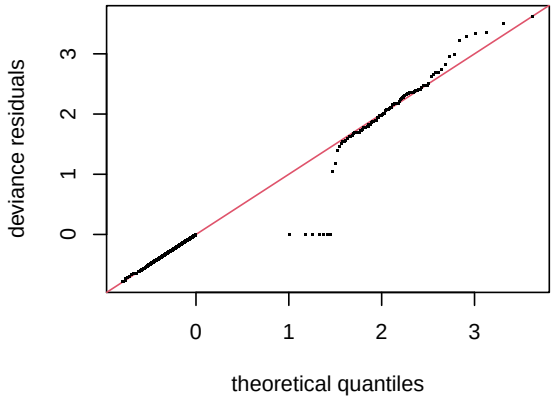
```
# * Does NOT include sighted acoustic encounters

twS999c <- gam(pa ~ s(bath_m, k = 3) + s(temp600, k = 3) + s(ssh,
    k = 3) + s(sshsd, k = 3) + s(eke, k = 3) + offset(log.effort),
    data = trainS999, family = tw, link = "log", select = TRUE,
    method = "REML")
summary(twS999c)
```

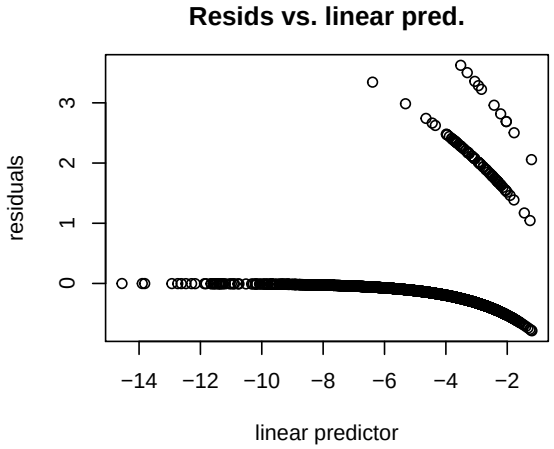
```
Family: Tweedie(p=1.01)
Link function: log

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

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -22.881      0.111  -206.2   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

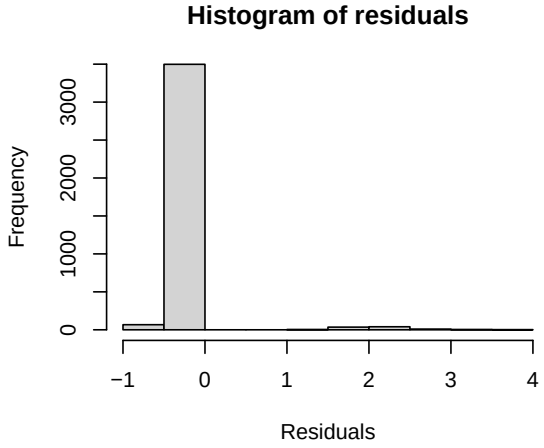
```
Approximate significance of smooth terms:
              edf Ref.df     F  p-value
s(bath_m)  0.8560      2 2.982  0.00794 **
s(temp600) 1.4065      2 4.771  0.00168 **
s(ssh)     0.9414      2 7.991 3.79e-05 ***
s(sshsd)   0.7690      2 1.701  0.03525 *
s(eke)     0.8710      2 2.976  0.00893 **
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

R-sq.(adj) =  0.0383   Deviance explained = 8.06%
-REML = 305.58  Scale est. = 1.0406    n = 3659
```

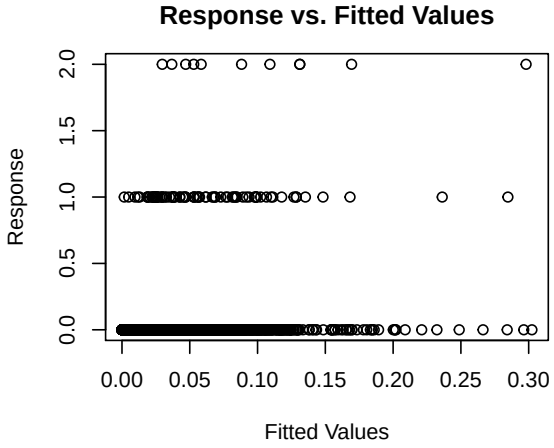


MODEL DIAGNOSTICS The stripe at the bottom left of the residuals vs. fitted values (linear predictor) corresponds to the zeros.










```
Method: REML   Optimizer: outer newton
full convergence after 22 iterations.
Gradient range [-0.001214774,0.0001396882]
(score 304.8044 & scale 1.040446).
Hessian positive definite, eigenvalue range [1.218905e-05,10499.47].
Model rank =  13 / 13

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)  2.000 0.840    0.85    0.02 *
s(chla)    2.000 0.750    0.88    0.29
s(temp600) 2.000 1.207    0.76  <2e-16 ***
s(ssh)     2.000 0.940    0.87    0.17
s(sshsd)   2.000 0.792    0.85    0.03 *
s(eke)     2.000 0.870    0.87    0.17
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

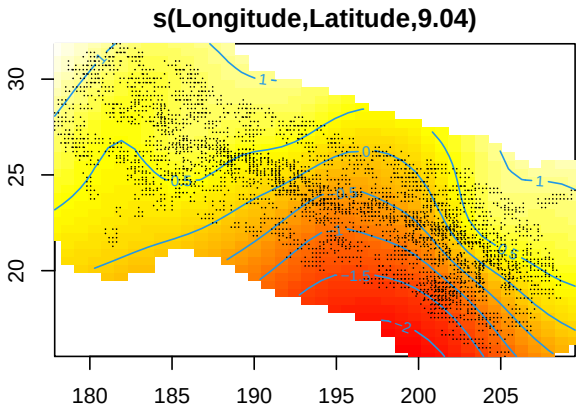
Include 2D Lat-Lon Smoother

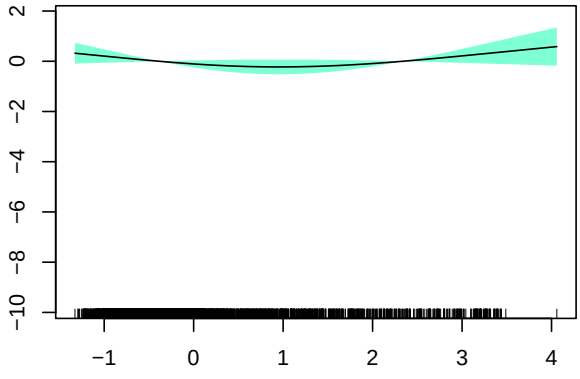
The 2D Lat-Lon smoother accounts for spatial autocorrelation in the data and fit the spatial variation not explained by the other predictors
* Notice that the temperature at 600m is no longer significant compared to the previous models
* Chlorophyll and SSHsd remain significant
+ Does this indicate that they aren't spatially structured and are independent of location?

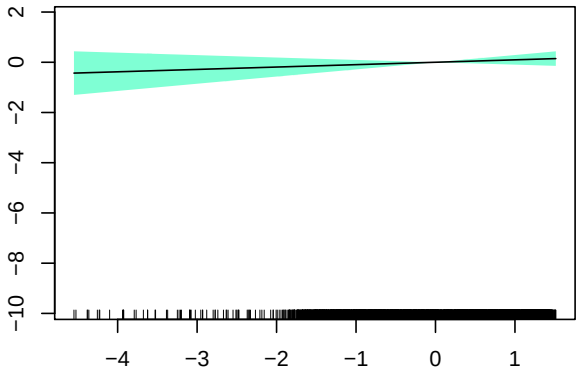
```
twS999LL <- gam(pa ~ s(Longitude, Latitude) + s(bath_m, k = 3) +
    s(dist, k = 3) + s(d2smt, k = 3) + s(sst, k = 3) + s(chla,
    k = 3) + s(temp600, k = 3) + s(ssh, k = 3) + s(sshsd, k = 3) +
    s(eke, k = 3) + s(wavepow, k = 3) + offset(log.effort), data = trainS999,
    family = tw, link = "log", select = TRUE, method = "REML")
summary(twS999LL)
```

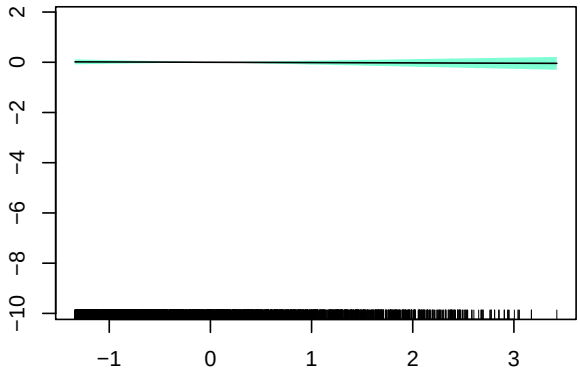
```
Family: Tweedie(p=1.01)
Link function: log

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|)
(Intercept) -23.0393     0.1255  -183.5   <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) 9.041e+00     29 1.200 1.62e-06 ***
s(bath_m)             7.025e-01      2 1.182  0.06115 .
s(dist)               5.511e-01      2 0.510  0.10537
s(d2smt)              1.023e-01      2 0.058  0.26724
s(sst)                6.318e-01      2 0.796  0.10296
s(chla)               7.613e-01      2 1.476  0.04146 *
s(temp600)            9.725e-05      2 0.000  0.62060
s(ssh)                8.220e-01      2 2.298  0.01453 *
s(sshsd)              8.495e-01      2 2.882  0.00865 **
s(eke)                8.548e-01      2 2.584  0.01319 *
s(wavepow)            2.367e-04      2 0.000  0.42850
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0577   Deviance explained = 14.4%
-REML = 297.21  Scale est. = 1.0388    n = 3659
```

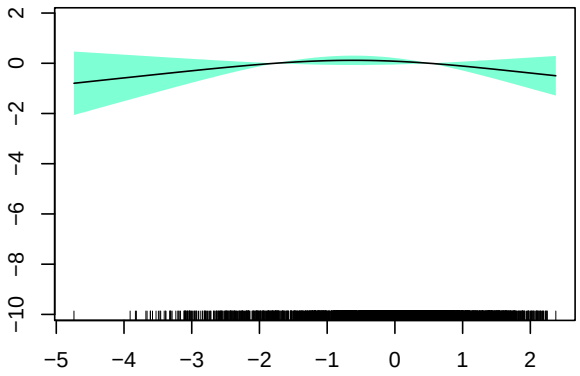
Full Acoustics Only Model w/ Spatial Smoother

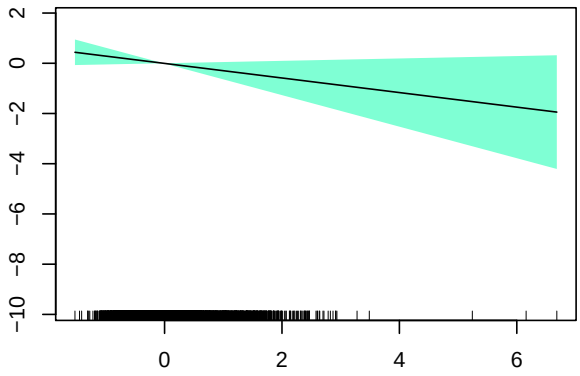

s(Longitude,Latitude,9.04)
30
25
20
180
185
190
195
200
205
Longitude

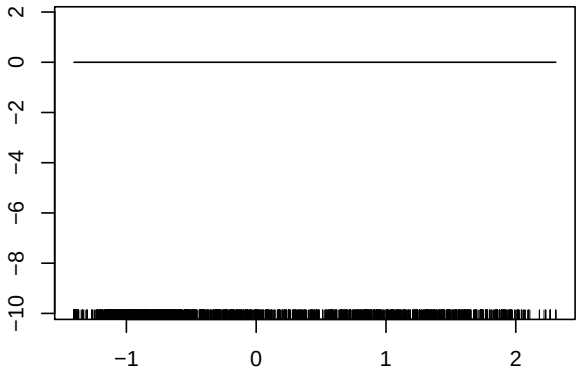

2
0
−2
−4
−6
−8
−10
−1
0
1
2
3
4
bath_m

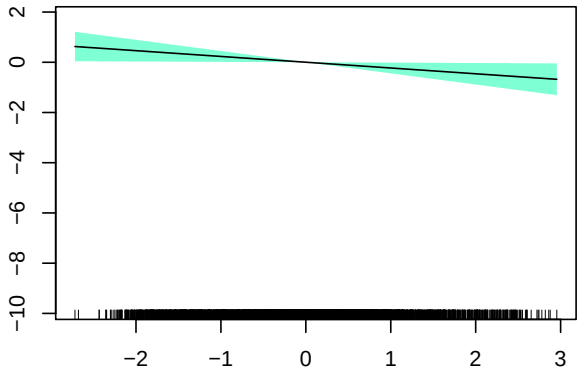

2
0
−2
−4
−6
−8
−10
−4
−3
−2
−1
0
1
dist

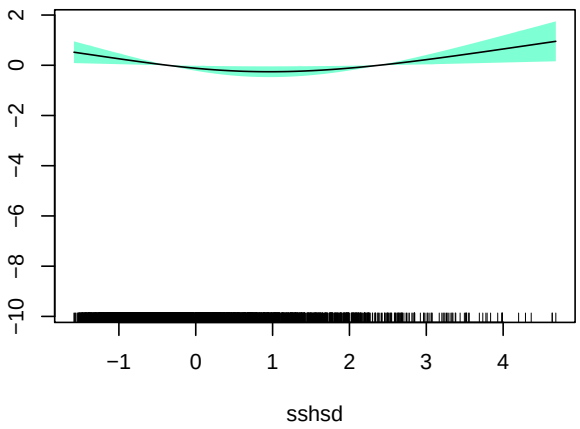

2
0
−2
−4
−6
−8
−10
−1
0
1
2
3
d2smt

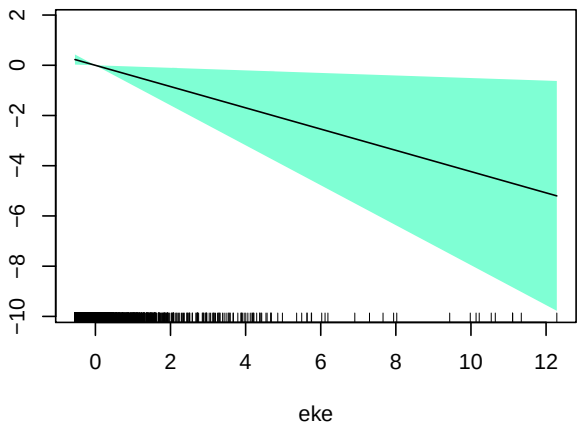

2
0
−2
−4
−6
−8
−10
−5
−4
−3
−2
−1
0
1
2
sst

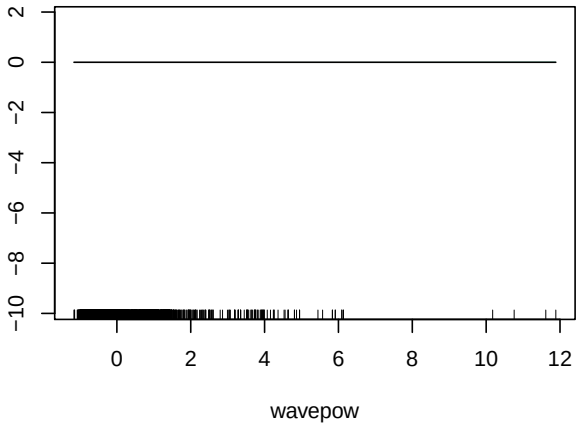

2
0
−2
−4
−6
−8
−10
0
2
4
6
chla


2
0
−2
−4
−6
−8
−10
−1
0
1
2
temp600


2
0
−2
−4
−6
−8
−10
−2
−1
0
1
2
3
ssh








```
# model diagnostics
par(mar = c(4, 4, 3, 3), mfrow = c(2, 2))
gam.check(twS999LL)
```

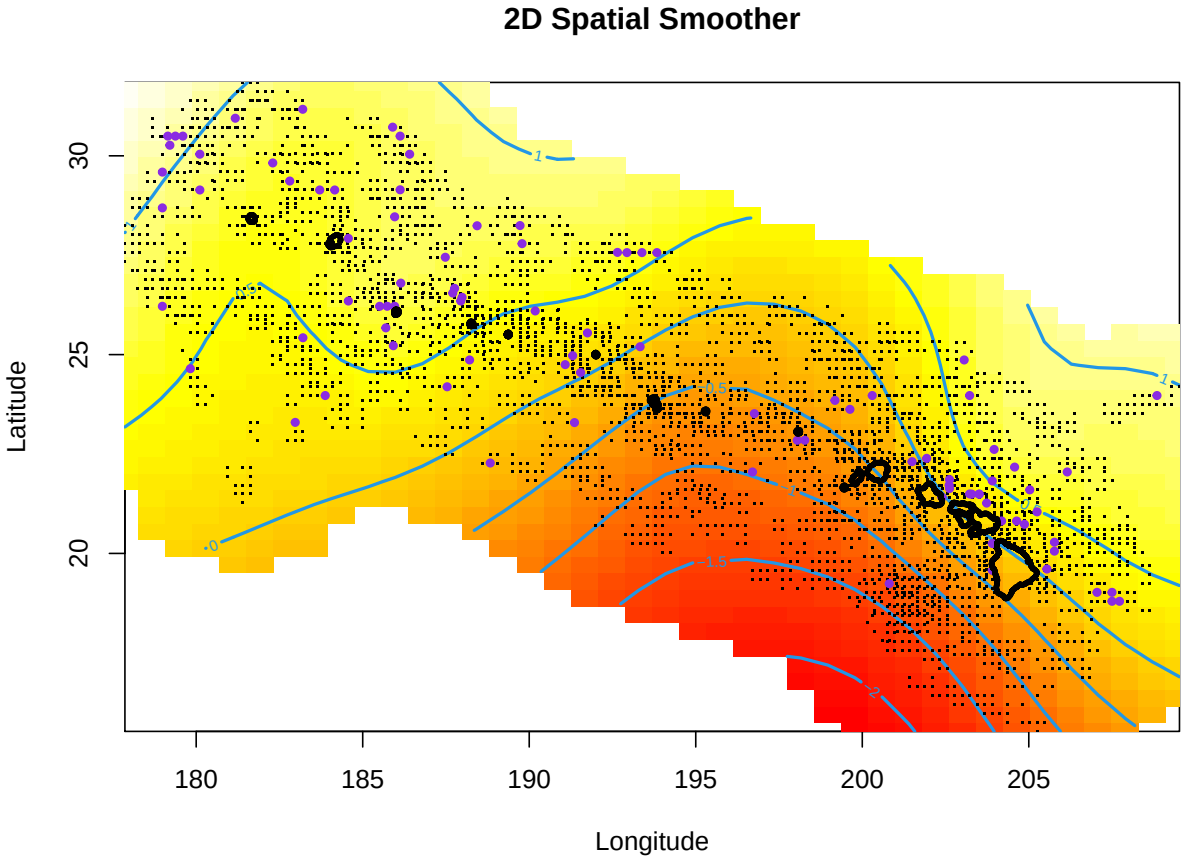
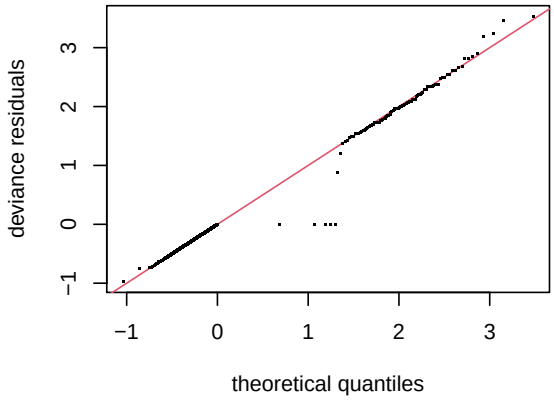
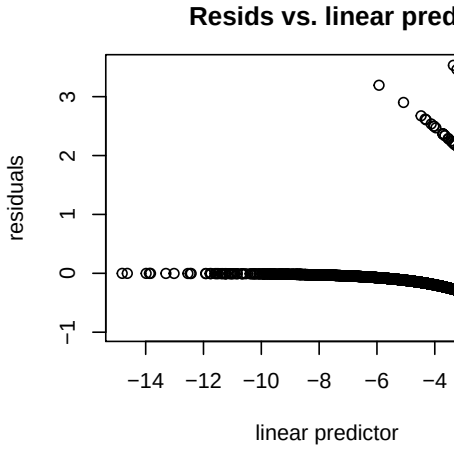
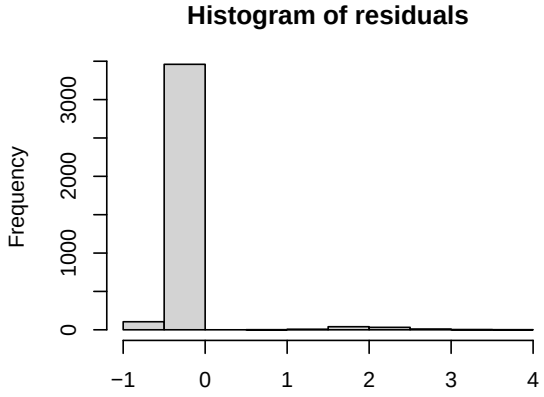
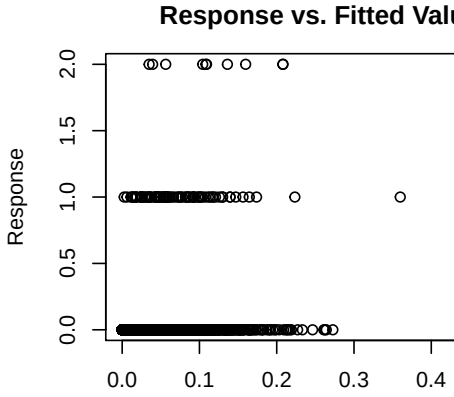


Figure 1: Purple dots represent acoustically detected encounters. Black dots are all data points(grid centroids)









MODEL DIAGNOSTICS

```
Method: REML   Optimizer: outer newton
full convergence after 28 iterations.
Gradient range [-0.0005930141,0.001301463]
(score 297.2075 & scale 1.03883).
Hessian positive definite, eigenvalue range [1.163763e-05,10499.5].
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 9.04e+00    0.83  <2e-16 ***
s(bath_m)             2.00e+00 7.02e-01    0.86   0.070 .
s(dist)               2.00e+00 5.51e-01    0.89   0.605
s(d2smt)              2.00e+00 1.02e-01    0.89   0.700
s(sst)                2.00e+00 6.32e-01    0.85   0.010 **
s(chla)               2.00e+00 7.61e-01    0.88   0.270
s(temp600)            2.00e+00 9.73e-05    0.77  <2e-16 ***
s(ssh)                2.00e+00 8.22e-01    0.88   0.225
s(sshsd)              2.00e+00 8.50e-01    0.85   0.005 **
s(eke)                2.00e+00 8.55e-01    0.88   0.265
s(wavepow)            2.00e+00 2.37e-04    0.81  <2e-16 ***
---
```

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

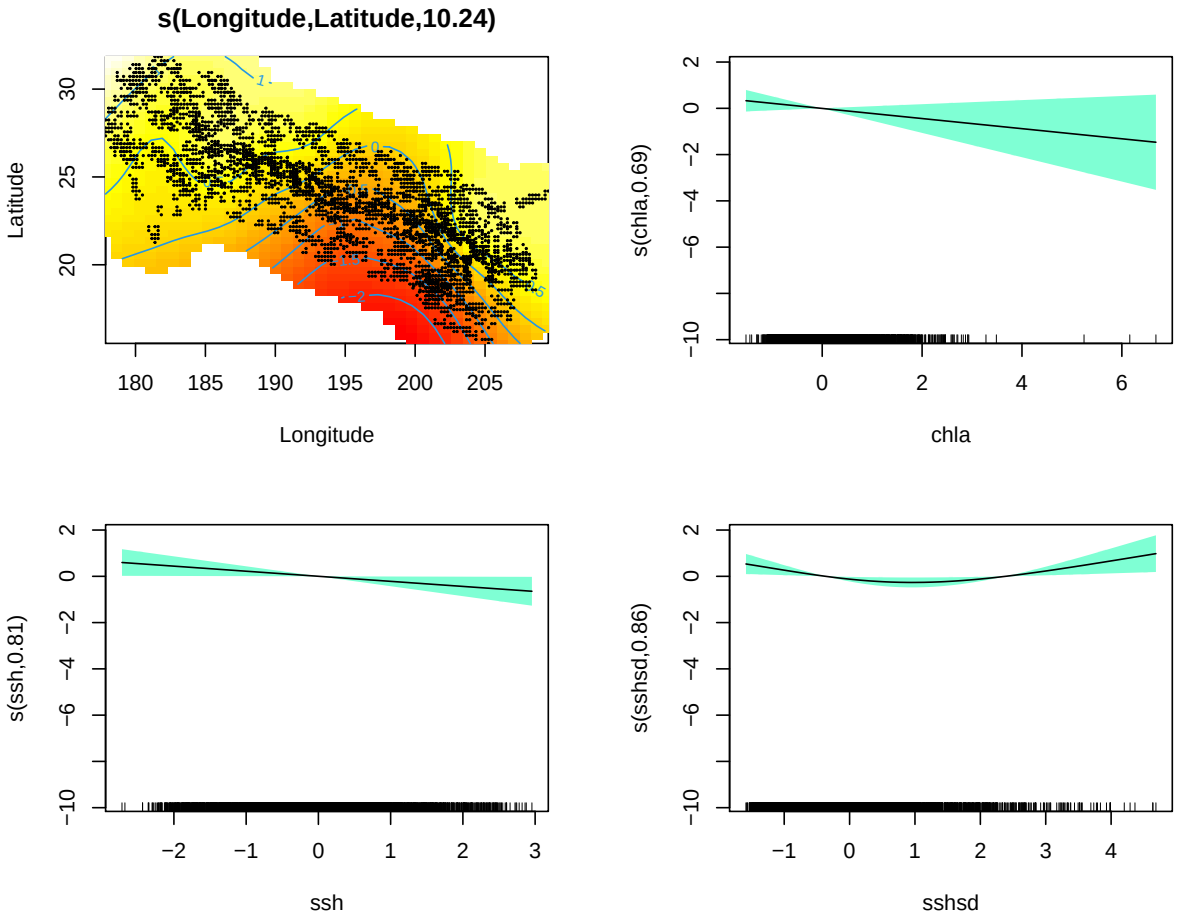
REDUCE MODEL PARAMETERS

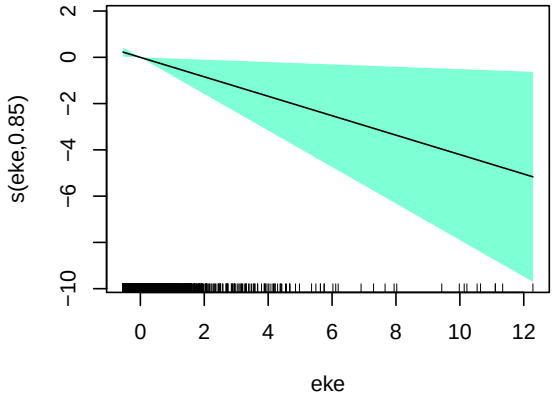
- Removed non-significant variables:
  - depth
  - distance to land
  - distance to seamount
  - sst
  - temp at 600 m
  - wave power

```
twS999LLb <- gam(pa ~ s(Longitude, Latitude) + s(chla, k = 3) +
    s(ssh, k = 3) + s(sshsd, k = 3) + s(eke, k = 3) + offset(log.effort),
    data = trainS999, family = tw, link = "log", select = TRUE,
    method = "REML")
summary(twS999LLb)
```

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

Formula:
pa ~ s(Longitude, Latitude) + s(chla, k = 3) + s(ssh, k = 3) +
    s(sshsd, k = 3) + s(eke, k = 3) + offset(log.effort)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -23.0255     0.1244  -185.1   <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) 10.2368     29 1.409 3.22e-07 ***
s(chla)                0.6877      2 1.016   0.0790 .
s(ssh)                 0.8114      2 2.151   0.0175 *
s(sshsd)               0.8571      2 3.057   0.0071 **
s(eke)                 0.8541      2 2.593   0.0131 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0543   Deviance explained = 13.9%
-REML = 298.07  Scale est. = 1.039     n = 3659
```

Reduced Acoustics Only Model w/ Spatial Smoother


s(Longitude,Latitude,10.24)
Latitude
Longitude
180
185
190
195
200
205
20
25
30
s(chla,0.69)
chla
0
2
4
6
−10
−6
−4
−2
0
2
s(ssh,0.81)
ssh
−2
−1
0
1
2
3
s(sshsd,0.86)
sshsd
−1
0
1
2
3
4



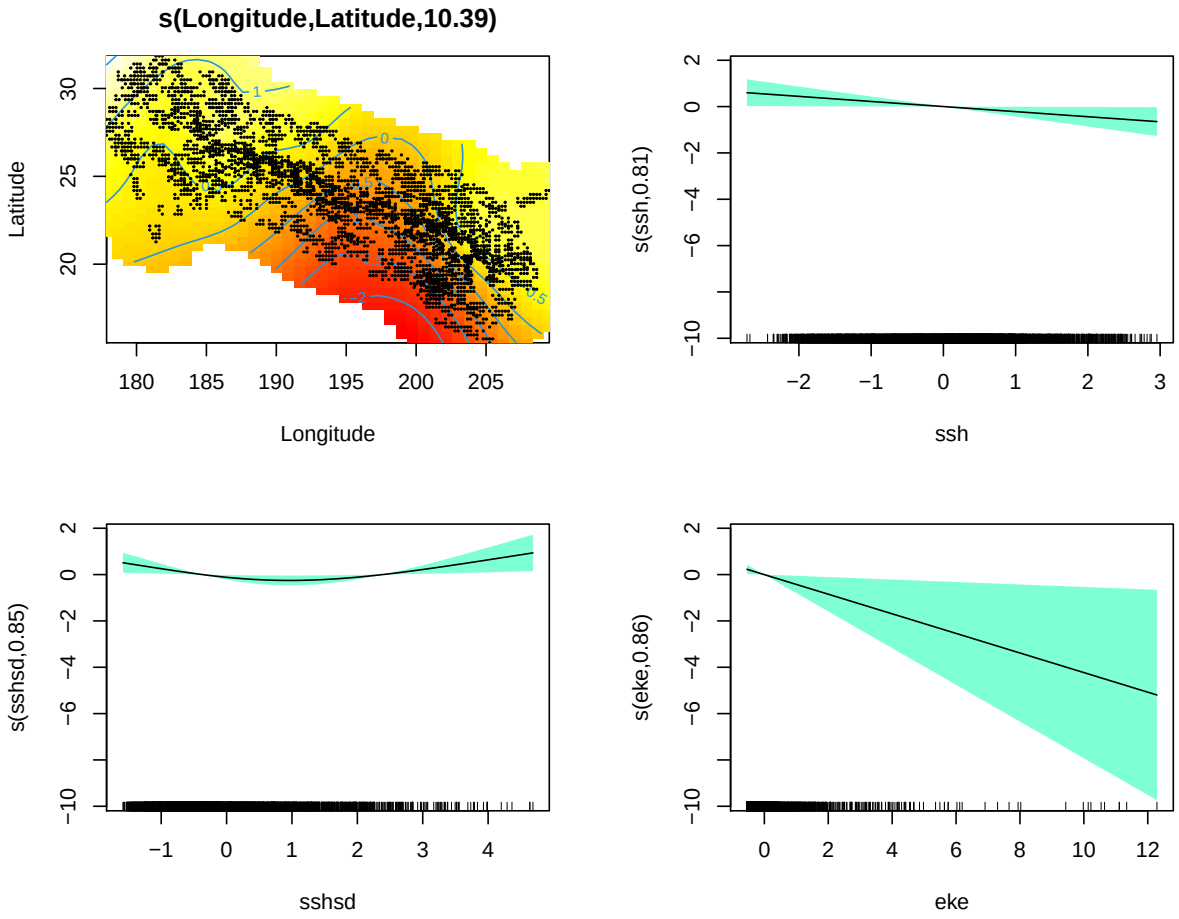
- Remove chlorophyll

```
twS999LLc <- gam(pa ~ s(Longitude, Latitude) + s(ssh, k = 3) +
    s(sshsd, k = 3) + s(eke, k = 3) + offset(log.effort), data = trainS999,
    family = tw, link = "log", select = TRUE, method = "REML")
summary(twS999LLc)
```

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

Formula:
pa ~ s(Longitude, Latitude) + s(ssh, k = 3) + s(sshsd, k = 3) +
    s(eke, k = 3) + offset(log.effort)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -23.0113     0.1234  -186.6   <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) 10.3931     29 1.510 8.9e-08 ***
s(ssh)                 0.8116      2 2.169  0.0171 *
s(sshsd)               0.8474      2 2.841  0.0091 **
s(eke)                 0.8552      2 2.620  0.0126 *
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

R-sq.(adj) =  0.0526   Deviance explained = 13.5%
-REML = 298.57  Scale est. = 1.0391    n = 3659
```

Reduced Acoustics Only Model w/ Spatial Smoother



Conclusions

The first reduced Acoustics Only model included more dynamic variables for the significant smooth terms compared to both models that included a 2D spatial smoother. The former addressed my hypothesis more clearly given that depth, chl a, and SSH/SSHsd were included as important variables. The introduction of the 2D smoother reduces all dynamic variables except for chla and SSHsd. Acoustic encounters of sperm whales indicate sperm whale occurrence is related to productivity in some way whether or not a 2D smoother is included. If the 2D smoother is left out of the model, I should acknowledge how certain important variables may also be spatially autocorrelated. Some papers state that spatial autocorrelation exists, calculate Moran’s I to determine the magnitude, and don’t do anything more about it, such as Forney et al. 2015.

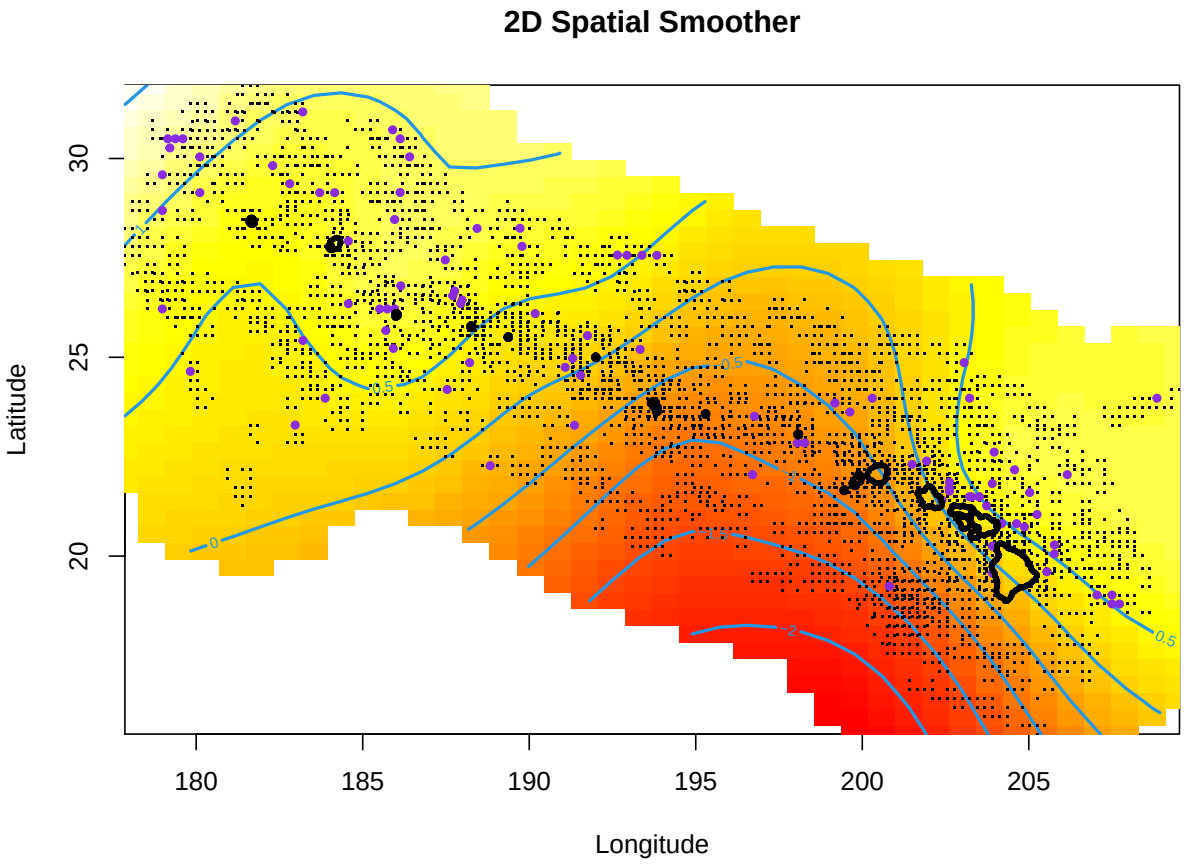


Figure 2: Purple dots represent acoustically detected encounters. Black dots are all data points.

NEED HELP HERE: Trying to sort out how to use the test data set for prediction purposes and how to use those results to evaluate model performance.

Predict Test Data

```
# https://gist.github.com/aperium/9fc737ea311a758328eadf27c2426e47

require(magrittr)
require(dplyr)

#### For twS999c, no spatial smoother ####
twTrainFinal <- trainS999 %>% mutate(resid = resid(twS999c),
    predict = predict(twS999c))
predTrain <- predict.gam(twS999c, type = "response")  #calculate MSE for these to compare with test set
twTrainFinal$fit <- predTrain

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

twPred <- predict.gam(twS999c, newdata = testS999, type = "response",
    se.fit = TRUE)
twTestFinal <- data.frame(testS999, fit = twPred$fit, se.fit = twPred$se.fit)
twMSEtest <- mean((twTestFinal$pa - twTestFinal$fit)^2)  #MSE

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


#### For twS999cLL, with spatial smoother #### pulling the
#### prediction and residual data from the model
twTrainLL <- trainS999 %>% mutate(resid = resid(twS999LLc), predict = predict(twS999LLc))
predTrainLL <- predict.gam(twS999LLc, type = "response")  #calculate MSE for these to compare with test
twTrainFinal$fit <- predTrainLL

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

twPredLL <- predict.gam(twS999LLc, newdata = testS999, type = "response",
    se.fit = TRUE)
twTestLL <- data.frame(testS999, fit = twPredLL$fit, se.fit = twPredLL$se.fit)
twMSEtestLL <- mean((twTestLL$pa - twTestLL$fit)^2)  #MSE

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

# AIC
```

```
twAIC <- AIC(twS999c)
twAICLL <- AIC(twS999LLc)

# Explained Deviance
twExpDev = round(((twS999c$null.deviance - twS999c$deviance)/twS999c$null.deviance) *
    100, 2)
twExpDevLL = round(((twS999LLc$null.deviance - twS999LLc$deviance)/twS999LLc$null.deviance) *
    100, 2)
```

```
# make summary table of metrics

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

# enter info by row

table[1, ] <- c("twS999c", paste0(twExpDev, "%"), round(twAIC,
    2), round(twMSEtrain, 2), round(twMSEtest, 2))

table[2, ] <- c("twS999LLc", paste0(twExpDevLL, "%"), round(twAICLL,
    2), round(twMSEtrainLL, 2), round(twMSEtestLL, 2))
require(knitr)
kable(table, caption = "Model Summary Metrics")
```

Table 1: Model Summary Metrics

Model	ExpDev	AIC	MSEtrain	MSEtest
twS999c	8.06%	3845.17	0.03	0.03
twS999LLc	13.52%	3642.05	NaN	0.03

Spatial Autocorrelation in data

Not model residuals

```
# https://stats.idre.ucla.edu/r/faq/how-can-i-calculate-morans-i-in-r/

# Create distance matrix, then take the inverse of the matrix
# values and replace with zeros on diagonal
sw.dists <- as.matrix((dist(cbind(trainS999$Longitude, trainS999$Latitude))))
sw.dists.inv <- 1/sw.dists
diag(sw.dists.inv) <- 0
sw.dists.inv[is.infinite(sw.dists.inv)] <- 0  #turn infinite values to 0

require(ape)
Moran.I(trainS999$pa, sw.dists.inv)
```