The following supplement accompanies the article:

Environmental factors determining the distribution and abundance of a diving marine bird: conservation implications for the common loon *Gavia immer*

Kristopher J. Winiarski*[1,2], David L. Miller[1], Peter W. C. Paton[1], Scott R. McWilliams[1]

[1]Department of Natural Resources Science, 1 Greenhouse Road, University of Rhode Island, Kingston, RI 02881 USA
[2]Present Address: Department of Environmental Conservation, 160 Holdsworth Way, University of Massachusetts, Amherst, MA 01003, USA

Email: withakri@gmail.com

Marine Ecology Progress Series 492:273-283 (2013)

---

1. Calculation of FCPI

Frequency peaks index of chlorophyll *a* (FCPI) for the Ocean Special Area Management Plan (OSAMP) was calculated as in Suryan et al. (2012). The procedure consisted of three steps. Given that we have measurements of chlorophyll *a* at locations i, and times t, g_{it}

1. Standardise the (base 10) logarithm of chlorophyll over time: $g_{i*} = \frac{\log_{10}(g_{it}) - \overline{[\log_{10}(g_{it})]_t}}{SD(\log_{10}(g_{it}))}$
2. Fit the linear regression model: $g_t = \beta_0 + \beta_1 \sin(2\pi f_1 t) + \beta_2 \cos(2\pi f_1 t) + \beta_3 \sin(2\pi f_2 t) + \beta_4 \cos(2\pi f_2 t) + \beta_5 t$ where $\beta_0, \ldots, \beta_5$ are regression coefficients, f_1 and f_2 are the frequency of annual and semi-annual cycles (12 and 6 months, respectively).
3. Calculate the proportion of time g_{i*} is >1 standard deviation higher than the fitted value given by the above regression.

The Fig. S1 shows the fitted model (red) and the data (black).
Output from the linear regression performed in `R` is:

```
Call:
lm(formula = mchl ~ sinf1 + cosf1 + sinf2 + cosf2 + t, data = mdat)

Residuals:
     Min       1Q   Median       3Q      Max
-0.27435 -0.06457 -0.01016  0.05616  0.38961

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.2506956  0.0199918  12.540  < 2e-16 ***
sinf1       -0.0383449  0.0140471  -2.730 0.007345 **
cosf1       -0.1077635  0.0140092  -7.692 5.61e-12 ***
sinf2       -0.0510859  0.0140150  -3.645 0.000404 ***
cosf2        0.0014290  0.0140092   0.102 0.918932
t            0.0007287  0.0002870   2.539 0.012472 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

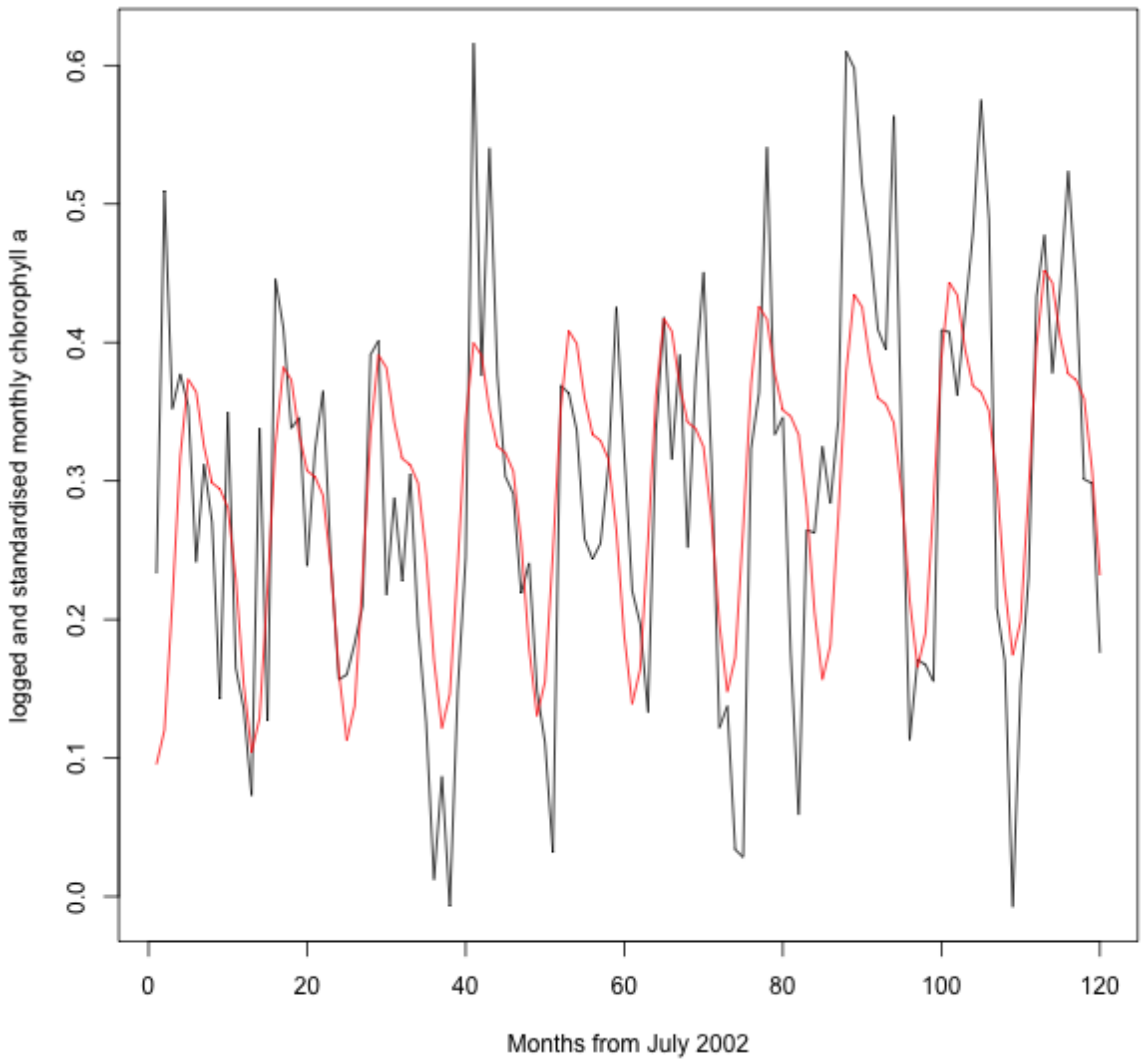

Figure S1: Chlorophyll *a* (once standardised and logarithms taken base 10) plotted over time (black) with the model predictions overlaid in red.

```
Residual standard error: 0.1085 on 114 degrees of freedom
Multiple R-squared: 0.4341, Adjusted R-squared: 0.4092
F-statistic: 17.49 on 5 and 114 DF,  p-value: 8.003e-13
```

The columns `sinf1`, `cosf1`, `sinf2` and `cosf2` were calulcated as $\sin(2\pi f_1 t)$ $\cos(2\pi f_1 t)$, $\sin(2\pi f_2 t)$ and $\cos(2\pi f_2 t)$, respectively. Here $f_1 = 12$, $f_2 = 6$ and t ranged from 1 to 120.

2. R code for FCPI calculation

We list below some R code for the calculation of the FCPI.

```
## FCPI calculation
# assume that data is in a data.frame called dat
# this has columns N+2 columns: first 2 columns to specify location (x,y or lat, long)
#  and N columns giving the N times that chlorophyll was sampled.

# separate the locations and chlorophyll measurements
chl.dat <- dat[,2:N]
loc.dat <- dat[,1:2]
# note the locations aren't used in this code, but can be useful for plotting a
# heatmap of FCPI.

## 1. log the chlorophyll readings and standardize
chl.dat <- log10(chl.dat)
zdat <- (dat-colMeans(chl.dat,na.rm=TRUE))/apply(chl.dat,2,sd,na.rm=TRUE)

## 2. fit the linear regression model
# set f1, f2 and t
f1 <- 1/12
f2 <- 1/6
t <- 1:120

# build a new data frame with the evaluations of the periodic functions
mdat <- data.frame(mchl  = colMeans(chl.dat,na.rm=TRUE),
                   t     = t,
                   sinf1 = sin(2*pi*f1*t),
                   cosf1 = cos(2*pi*f1*t),
                   sinf2 = sin(2*pi*f2*t),
                   cosf2 = cos(2*pi*f2*t))

# fit the linear model
m.lm <- lm(mchl~sinf1+cosf1+sinf2+cosf2+t,data=mdat)

# extract the fitted values
m.lm.pred <- predict(m.lm)

## 3. find the proportion of time that a pixel is >1 SD above the monthly average
fcpi <- rowSums(zdat-m.lm.pred) >
               apply(zdat,2,sd,na.rm=TRUE),na.rm=TRUE)/120
```

```
# make the plot
plot(sort(t),mdat$mchl[order(t)],type="l",
     ylab="logged and standardised monthly chlorophyll a",
     main="Monthly Chlorophyll a\n(data=black, model=red)",
     xlab="Months from July 2002")
lines(sort(t),m.lm.pred[order(t)],col="red")
```

3. Analysis of common loon data

This section records the modeling for the common loon data from the URI survey of the OSAMP area off the coast of Rhode Island. It includes minimal interpretation of results, which can be found in the accompanying paper.

This document has been created using `knitr` (Xei 2013). The file `MEPS-analysis.Rmd` includes all the code necessary to run the models described.

3.1. Preamble

Load the data and `dsm` package (Miller 2012).

```
load("loon-data.RData")
suppressPackageStartupMessages(library(dsm))
```

3.2. Exploratory data analysis

We begin by plotting the raw data: the observed distances (Fig. S2), raw observations both unaggregated (Fig. S3) and split according to survey season (Fig. S4).

First plotting the histogram of observed distances:

```
hist(obs.loons$distance, breaks = sort(unique(c(obs.loons$distbegin, obs.loons$distend))),
    main = "", xlab = "Distance (m)", axes = FALSE)
axis(2)
axis(1, at = c(44, 164, 433, 1000))
box()


p <- ggplot(obs.loons)
p <- p + geom_point(aes(x = x, y = y, size = size), alpha = 0.5)
p <- p + geom_path(aes(x = x, y = y, group = group), data = coast)
p <- p + p.opts.geo
p <- p + coord_equal(xlim = xlims, ylim = ylims)
leg.breaks <- unique(quantile(obs.loons$size))
leg.breaks <- round(seq(leg.breaks[1], leg.breaks[2], len = 5), 0)
leg.breaks <- round(leg.breaks, 0)
p <- p + scale_size(breaks = leg.breaks)
p <- p + labs(x = "km east", y = "km north")
print(p)


p <- p + facet_wrap(~SeasonYear)
print(p)
```

These raw plots clearly show higher observed abundances in the area between Block Island and Long Island Sound.

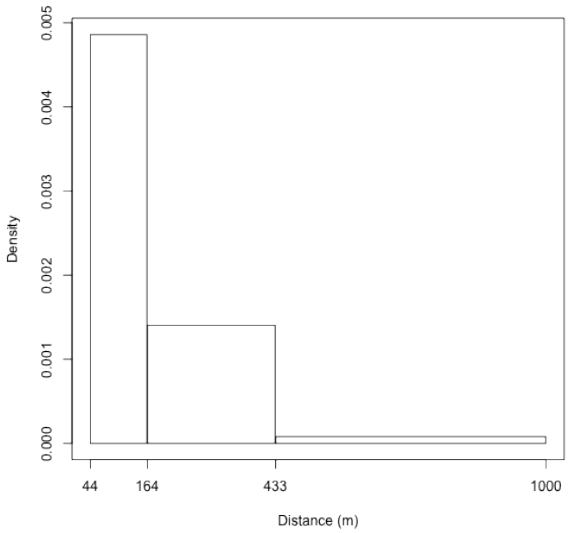

Figure S2: Histogram of observed distances to flocks of common loons

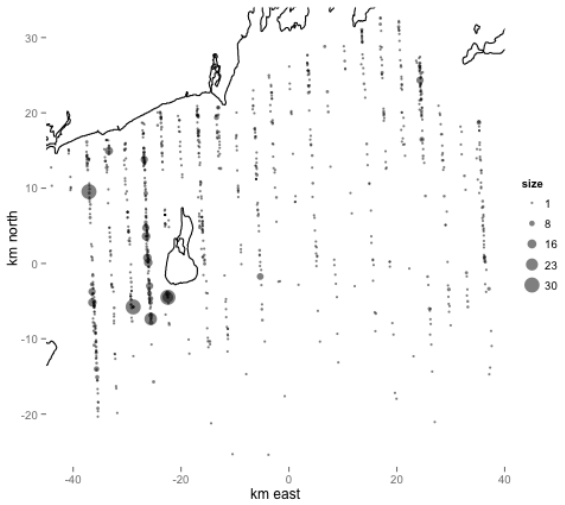

Figure S3: Raw observations of common loons. The size of the circle relates to the size of the observed flock.

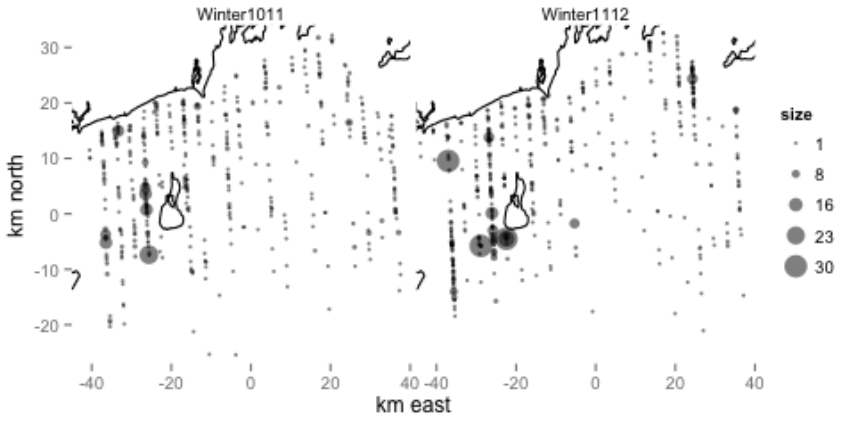

Figure S4: Raw observations of common loons aggregated to the survey season. The size of the circle relates to the size of the observed flock.

3.3. Detection function analysis

Stage one of the density surface modeling approach is to adjust the counts to account for detectability. We begin by fitting a detection function to the distance data using the `R` package `Distance` (Miller 2013).

Fit a detection function with half-normal, hazard rate and uniform key functions with no covariates and select adjustment terms by AIC. We enforce monotonicity on the resulting models to ensure that probability of detection doesn't increase with distance.

```
hn.df <- ds(obs.loons, truncation = list(right = 1000, left = 44), monotonicity = "strict")
```

```
## Warning: No cutpoints specified but distbegin and distend are columns in
## data. Performing a binned analysis...
```

```
## Starting AIC adjustment term selection. Fitting half-normal key function
## AIC= 1603.281 Fitting half-normal key function with cosine(2) adjustments
## AIC= 1605.19
##
## half-normal key function selected! No survey area information supplied,
## only estimating detection function.
```

```
hr.df <- ds(obs.loons, truncation = list(right = 1000, left = 44), key = "hr",
    monotonicity = "strict")
```

```
## Warning: No cutpoints specified but distbegin and distend are columns in
## data. Performing a binned analysis...
```

```
## Starting AIC adjustment term selection. Fitting hazard-rate key function
## AIC= 1580.559 Fitting hazard-rate key function with cosine(2) adjustments
```

```
## First partial hessian is singular; using second-partial hessian
```

```
## AIC= 1599.192
##
## hazard-rate key function selected! No survey area information supplied,
## only estimating detection function.
```

```
hn.herm.df <- ds(obs.loons, adjustment = "herm", truncation = list(right = 1000,
    left = 44), monotonicity = "strict")
```

```
## Warning: No cutpoints specified but distbegin and distend are columns in
## data. Performing a binned analysis...
```

```
## Starting AIC adjustment term selection. Fitting half-normal key function
## AIC= 1603.281 Fitting half-normal key function with Hermite(4) adjustments
## AIC= 1605.283
##
## half-normal key function selected! No survey area information supplied,
## only estimating detection function.
```

```
hr.poly.df <- ds(obs.loons, adjustment = "poly", truncation = list(right = 1000,
    left = 44), key = "hr", monotonicity = "strict")
```

```
## Warning: No cutpoints specified but distbegin and distend are columns in
## data. Performing a binned analysis...
```

```
## Starting AIC adjustment term selection. Fitting hazard-rate key function
## AIC= 1580.559 Fitting hazard-rate key function with simple polynomial(2)
## adjustments AIC= 1582.19
##
## hazard-rate key function selected! No survey area information supplied,
## only estimating detection function.
```

We can also see if covariates have an effect (though we cannot then ensure that the resulting functions are monotonic), looking at flock size and observer:

```
hn.df.size <- ds(obs.loons, formula = ~size, adjustment = NULL, truncation = list(right = 1000,
    left = 44))
```

```
## Warning: No cutpoints specified but distbegin and distend are columns in
## data. Performing a binned analysis...

## Fitting half-normal key function AIC= 1602.062 No survey area information
## supplied, only estimating detection function.
```

```
hr.df.size <- ds(obs.loons, formula = ~size, adjustment = NULL, truncation = list(right = 1000,
    left = 44), key = "hr")
```

```
## Warning: No cutpoints specified but distbegin and distend are columns in
## data. Performing a binned analysis...

## Fitting hazard-rate key function AIC= 1580.124 No survey area information
## supplied, only estimating detection function.
```

```
hn.df.obs <- ds(obs.loons, formula = ~as.factor(Observer), adjustment = NULL,
    truncation = list(right = 1000, left = 44))
```

```
## Warning: No cutpoints specified but distbegin and distend are columns in
## data. Performing a binned analysis...

## Fitting half-normal key function AIC= 1601.629 No survey area information
## supplied, only estimating detection function.
```

```
hr.df.obs <- ds(obs.loons, formula = ~as.factor(Observer), adjustment = NULL,
    truncation = list(right = 1000, left = 44), key = "hr")
```

```
## Warning: No cutpoints specified but distbegin and distend are columns in
## data. Performing a binned analysis...

## Fitting hazard-rate key function AIC= 1579.468 No survey area information
## supplied, only estimating detection function.
```

Also a model with both covariates included:

```
hn.df.size.obs <- ds(obs.loons, formula = ~size + as.factor(Observer), adjustment = NULL,
    truncation = list(right = 1000, left = 44))
```

```
## Warning: No cutpoints specified but distbegin and distend are columns in
## data. Performing a binned analysis...
```

```
## Fitting half-normal key function AIC= 1599.763 No survey area information
## supplied, only estimating detection function.
```

```
hr.df.size.obs <- ds(obs.loons, formula = ~size + as.factor(Observer), adjustment = NULL,
    truncation = list(right = 1000, left = 44), key = "hr")
```

```
## Warning: No cutpoints specified but distbegin and distend are columns in
## data. Performing a binned analysis...
```

```
## Fitting hazard-rate key function AIC= 1578.736 No survey area information
## supplied, only estimating detection function.
```

We first note that out of the 10 models fitted, the AIC score of the top 5 are within 2 points of each other (column Δ AIC). Models 4 and 5 are identical, as no adjustments were selected by AIC for either hazard-rate model. Fig. S5 shows plots of the top 5 models.

Discarding model 5, the rest of the top 4 models will be used in the spatial models fitted below and compared. We note however that the top model requires 5 parameters to obtain an improvement of less than 1 AIC point over the next ranked model (and less than 2 different from the 4th ranked model).

```
par(mfrow = c(2, 3))
plot(hr.df.size.obs, pl.den = 0, main = "Hazard-rate (size+obs)")
plot(hn.df, pl.den = 0, main = "Half-normal+cos(2)")
plot(hr.df.obs, pl.den = 0, main = "Hazard-rate (obs)")
plot(hr.df.size, pl.den = 0, main = "Hazard-rate (size)")
plot(hr.df, pl.den = 0, main = "Hazard-rate (No covars)")
```

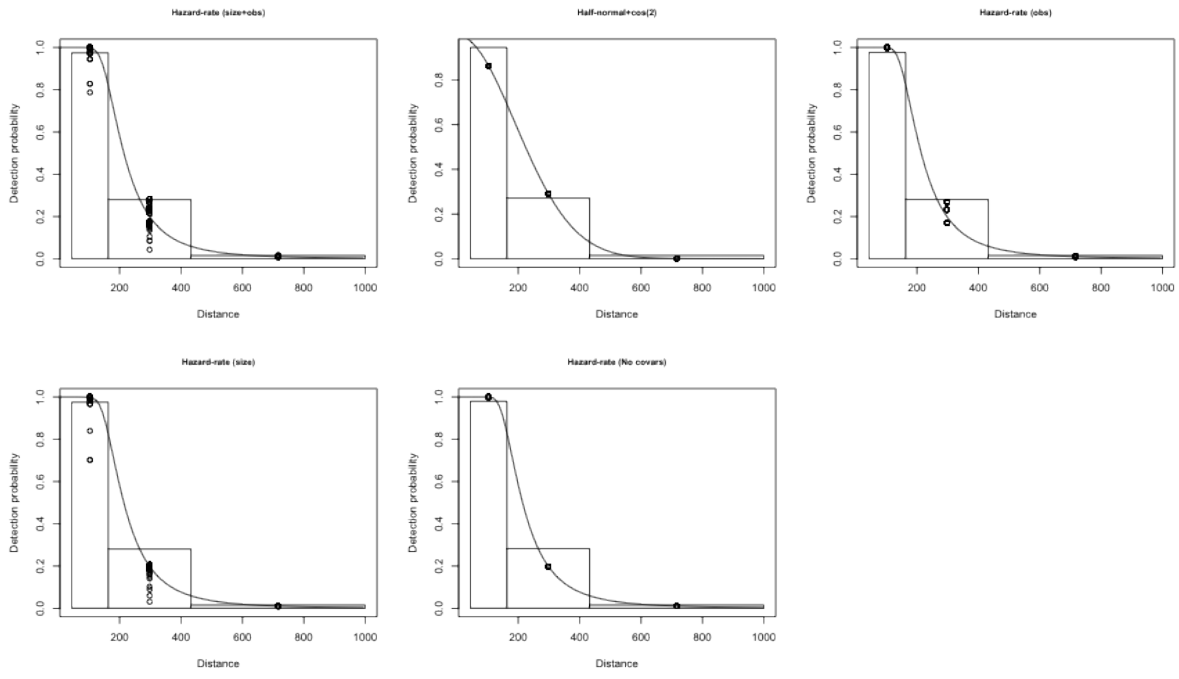

Figure S5: Fitted detection functions.

Table S1: Table of the detection function results ordered by AIC. Note that in none of the models considered were adjustment terms selected.

Detection function	Adjustments	Covariates	AIC	Δ AIC	# pars	p	CV(p)	C-vM p
1	hr	size+Observer	1578.736	0	5	0.201	0.037	0.02
2	hr	Observer	1579.468	0.732	4	0.201	0.037	0.023
3	hr	size	1580.124	1.388	3	0.201	0.037	0.03
4	hr	cos(0)	1580.559	1.823	2	0.201	0.036	0.034
5	hr	simple polynomial(0)	1580.559	1.823	2	0.201	0.036	0.034
6	hn	size+Observer	1599.763	21.027	4	0.192	0.028	0.016
7	hn	Observer	1601.629	22.893	3	0.193	0.027	0.019
8	hn	size	1602.062	23.326	2	0.193	0.027	0.025
9	hn	cos(0)	1603.281	24.545	1	0.194	0.027	0.03
10	hn	hermite polynomial(0)	1603.281	24.545	1	0.194	0.027	0.03

3.4. Spatial modeling

We now fit a model using the three measures of chlorophyll *a* described in the article. Before proceeding, we give a few comments about modeling strategy.

We can include all spatially-referenced covariates in the model and use the `select=TRUE` option, this allows the smooth terms to be shrunk to zero (flat linear effects), which we can then remove from the model and refit (see Wood 2006 Section 4.1.6 and Wood 2011).

From Winiarski et al (2014), we see that the parameter of the negative binomial is around 0.18. Here we specify a relatively wide range (which we can widen if it appears that the parameter is hitting the bounds) and estimate it along with the other parameters.

All models below show the "final" model with the terms removed commented out. Those wishing to see how model development progressed can uncomment these terms and follow through the term selection process, re-fitting these models.

In terms of model checking, we use the plots provided by `gam.check` (in particular Q-Q plots, see Figs. S6-S9). It is likely that neighbouring segments have similar counts, since we would expect that loons cluster near, for example, prey agglomerations. To check that there is not unmodelled correlation in the data, we calculate the correlations between the per-segment residuals at different "lags". This can be achieved using the `dsm.cor` function in the `dsm` package.

First setting the basis sizes for unidimensional and bivariate smooth terms:

```
k1 <- 10
k2 <- 18
```

We then proceed with model fitting.

3.4.1. DSM – hazard-rate (observer and group size)

```
loon.model.obs.size <- dsm(Nhat~#s(gchl_winter,k=k1)+
                           s(gchl_long,k=k1)+
                           #s(fcpi,k=k1)+
                           #s(roughness,k=k1)+
                           #s(phimedian,k=k1)+
                           #distancelandkm+
                           #s(distancelandkm,k=k1)+
                           s(depthm,k=k1)+
                           #s(x,k=k1)+
                           s(y,k=k1),#+
                           #s(x,y,k=k2),
                     hr.df.size.obs, seg, obs.loons,
                     family=negbin(theta=c(0.1,0.2)), availability=0.7,
                     #family=negbin(theta=c(0.1,0.12)), availability=0.7,
                     select=TRUE, method="REML")

summary(loon.model.obs.size)

##
## Family: Negative Binomial(0.105)
## Link function: log
##
## Formula:
## Nhat ~ s(gchl_long, k = k1) + s(depthm, k = k1) + s(y, k = k1) +
##     offset(off.set)
```

```
## <environment: 0x10c255b20>
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.6094     0.0748    -195   <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(gchl_long) 4.27      9   66.4 5.3e-16 ***
## s(depthm)    2.85      9   27.1 6.2e-08 ***
## s(y)         3.47      9   42.6 4.0e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0857   Deviance explained = 29.1%
## REML score = 3185.7  Scale est. = 1         n = 2067
```

```
gam.check(loon.model.obs.size)
```

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 1 iteration.
## Gradient range [-0.0007213,0.0001573]
## (score 3186 & scale 1).
## eigenvalue range [-0.0001571,1.405].
##
## 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(gchl_long) 9.000 4.270   0.581    0.02
## s(depthm)    9.000 2.850   0.587    0.08
## s(y)         9.000 3.471   0.587    0.04
```

```
summary(dsm.var.gam(loon.model.obs.size, pred, pred$cellaream))
```

```
## Summary of uncertainty in a density surface model calculated
##  analytically for GAM, with delta method
##
## Approximate asymptotic confidence interval:
##   5% Mean  95%
## 4454 5707 7312
## (Using delta method)
##
## Point estimate                : 5707
## Standard error                : 692.6
## CV of detection function      : 0.03728
## CV from GAM                   : 0.1214
## Total coefficient of variation : 0.127
```

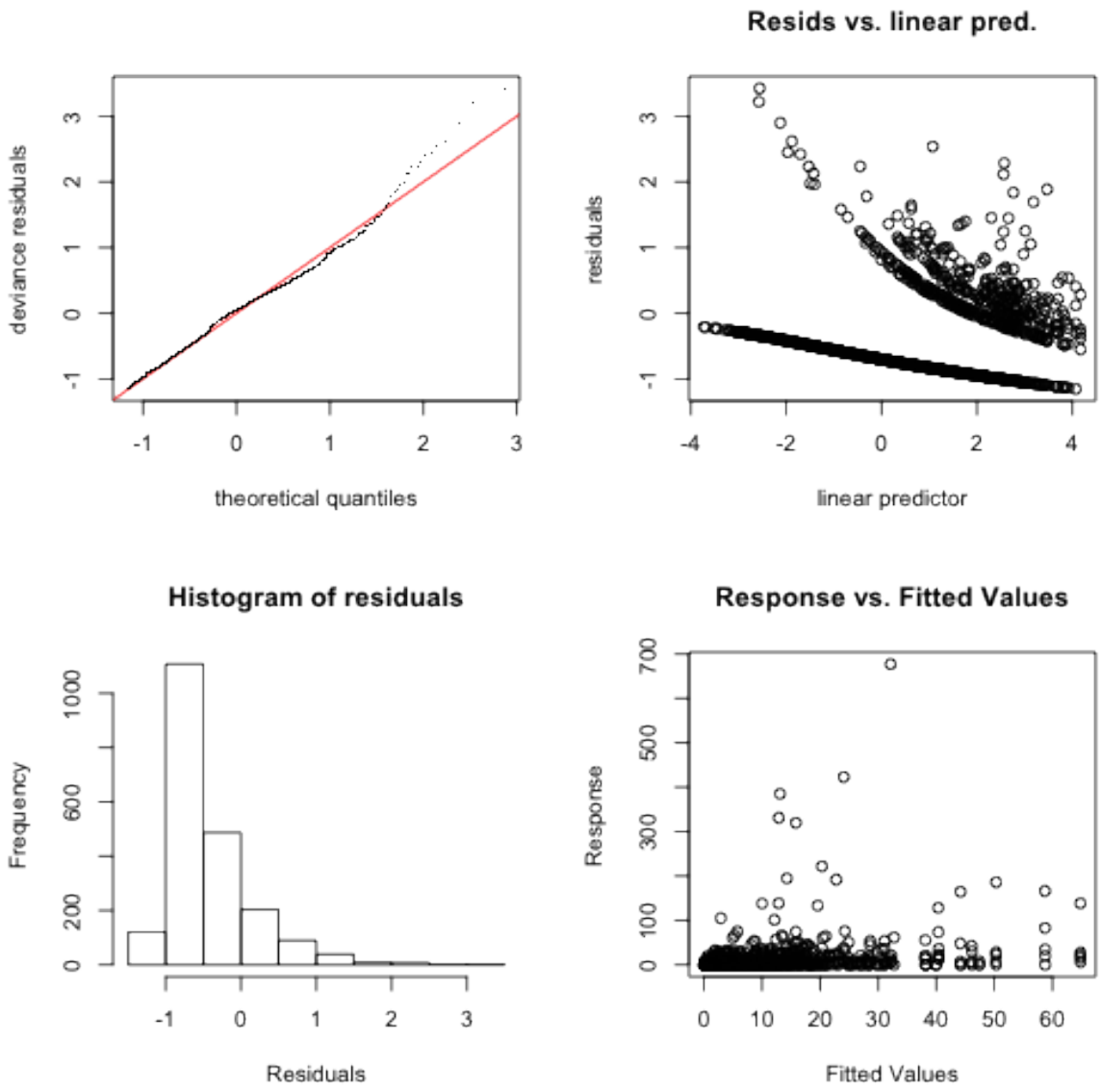

Figure S6: GAM check plots for model `loon.model.obs.size`.

3.4.2. DSM – hazard-rate (observer)

```
loon.model.obs <- dsm(Nhat~#s(gchl_winter,k=k1)+
                          s(gchl_long,k=k1)+
                          #s(fcpi,k=k1)+
                          #s(roughness,k=k1)+
                          #s(phimedian,k=k1)+
                          #distancelandkm+
                          #s(distancelandkm,k=k1)+
                          s(depthm,k=k1)+
                          #s(x,k=k1)+
                          s(y,k=k1),#+
                          #s(x,y,k=k2),
                      hr.df.obs, seg, obs.loons,
                      family=negbin(theta=c(0.1,0.2)), availability=0.7,
                      select=TRUE, method="REML")

summary(loon.model.obs)

##
## Family: Negative Binomial(0.108)
## Link function: log
##
## Formula:
## Nhat ~ s(gchl_long, k = k1) + s(depthm, k = k1) + s(y, k = k1) +
##     offset(off.set)
## <environment: 0x109fa2eb0>
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.6334     0.0736    -199   <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(gchl_long) 4.02      9   57.1 6.7e-14 ***
## s(depthm)    2.86      9   27.5 4.6e-08 ***
## s(y)         3.35      9   38.8 3.4e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.127   Deviance explained = 27.9%
## REML score = 3177.7  Scale est. = 1         n = 2067

gam.check(loon.model.obs)


##
## Method: REML   Optimizer: outer newton
## full convergence after 1 iteration.
## Gradient range [-0.0003685,0.0008255]
```

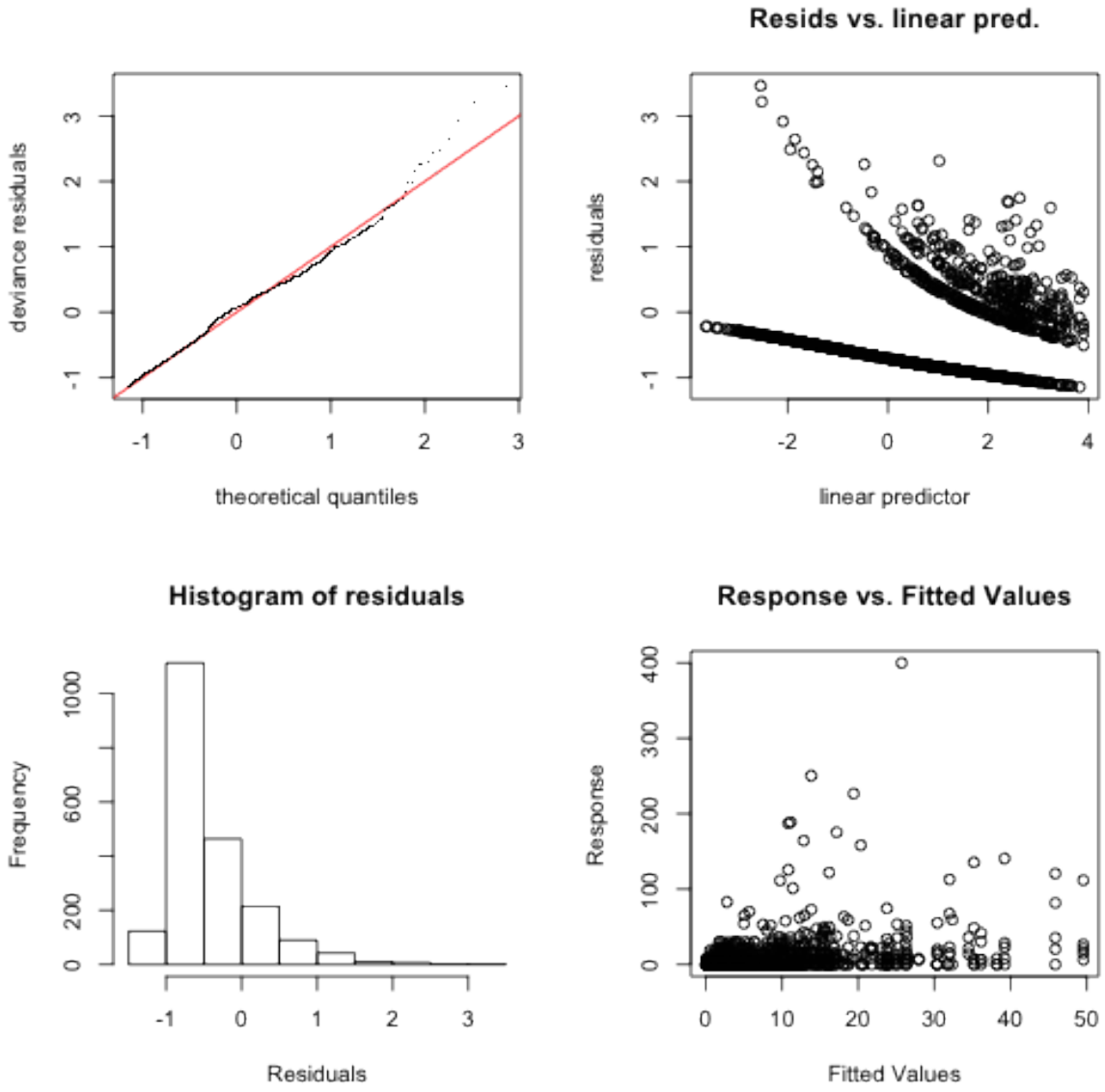

Figure S7: GAM check plots for model `loon.model.obs`.

```
## (score 3178 & scale 1).
## eigenvalue range [-0.0008221,1.392].
##
## 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(gchl_long) 9.000 4.017   0.587    0.01
## s(depthm)    9.000 2.863   0.593    0.04
## s(y)         9.000 3.349   0.593    0.04
```

```
summary(dsm.var.gam(loon.model.obs, pred, pred$cellaream))
```

```
## Summary of uncertainty in a density surface model calculated
##  analytically for GAM, with delta method
##
## Approximate asymptotic confidence interval:
##   5% Mean  95%
## 4056 5127 6481
## (Using delta method)
##
## Point estimate                 : 5127
## Standard error                 : 585.6
## CV of detection function       : 0.03671
## CV from GAM                    : 0.1142
## Total coefficient of variation : 0.12
```

3.4.3. DSM – hazard-rate (group size)

```
loon.model.size <- dsm(Nhat~#s(gchl_winter,k=k1)+
                           s(gchl_long,k=k1)+
                           #s(fcpi,k=k1)+
                           #s(roughness,k=k1)+
                           #s(phimedian,k=k1)+
                           #distancelandkm+
                           #s(distancelandkm,k=k1)+
                           s(depthm,k=k1)+
                           #s(x,k=k1)+
                           s(y,k=k1),#+
                           #s(x,y,k=k2),
                  hr.df.size, seg, obs.loons,
                  family=negbin(theta=c(0.1,0.2)), availability=0.7,
                  select=TRUE, method="REML")
```

```
summary(loon.model.size)
```

```
##
## Family: Negative Binomial(0.105)
## Link function: log
##
## Formula:
## Nhat ~ s(gchl_long, k = k1) + s(depthm, k = k1) + s(y, k = k1) +
```

```
##     offset(off.set)
## <environment: 0x112b65720>
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.6105     0.0747    -196   <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(gchl_long) 4.34      9   69.0 < 2e-16 ***
## s(depthm)    2.83      9   26.6 8.2e-08 ***
## s(y)         3.46      9   42.8 3.4e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0774   Deviance explained = 29.3%
## REML score = 3186.2  Scale est. = 1         n = 2067
```

```
gam.check(loon.model.size)
```

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 1 iteration.
## Gradient range [-0.000657,0.0001847]
## (score 3186 & scale 1).
## eigenvalue range [-0.0001845,1.407].
##
## 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(gchl_long) 9.000 4.336   0.582    0.03
## s(depthm)    9.000 2.829   0.588    0.08
## s(y)         9.000 3.461   0.587    0.04
```

```
summary(dsm.var.gam(loon.model.size, pred, pred$cellaream))
```

```
## Summary of uncertainty in a density surface model calculated
##  analytically for GAM, with delta method
##
## Approximate asymptotic confidence interval:
##   5% Mean  95%
## 4486 5760 7396
## (Using delta method)
##
## Point estimate                : 5760
## Standard error                : 706
## CV of detection function      : 0.0371
## CV from GAM                   : 0.1226
## Total coefficient of variation : 0.1281
```

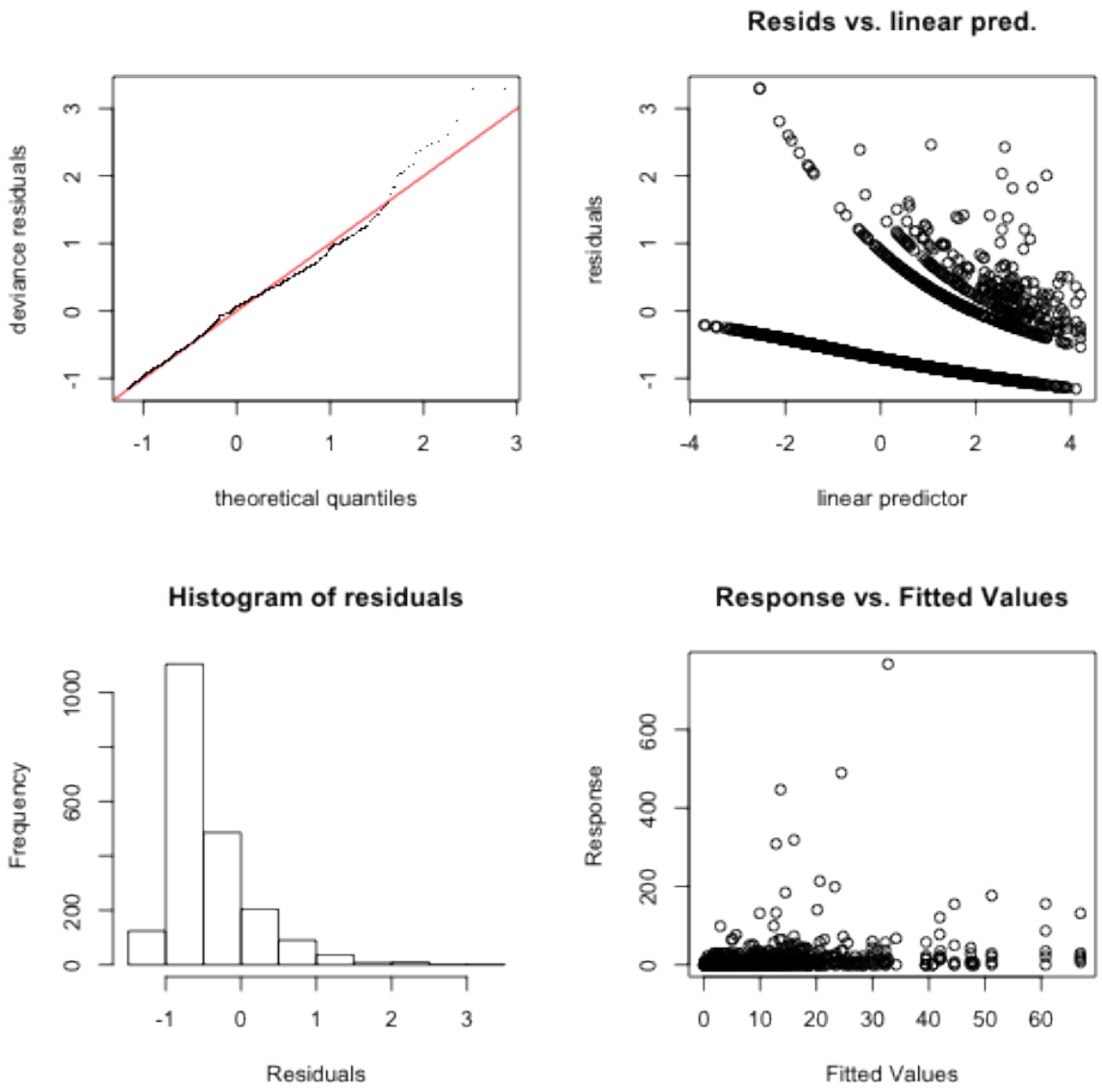

Figure S8: GAM check plot for model `loon.model.size`.

3.4.4. DSM – hazard-rate (no covariates)

```
loon.model.hr <- dsm(N~s(gchl_long,k=k1)+
#                      s(gchl_winter,k=k1)+
#                      s(fcpi,k=k1)+
#                      s(roughness,k=k1)+
#                      s(phimedian,k=k1)+
#                      s(distancelandkm,k=k1)+
                     s(depthm,k=k1)+
#                      s(x,k=k1)+
                     s(y,k=k1),#+
#                      s(x,y,k=k2),
                   hr.df, seg, obs.loons,
                   family=negbin(theta=c(0.1,0.2)), availability=0.7,
                   select=TRUE, method="REML")

summary(loon.model.hr)
```

```
##
## Family: Negative Binomial(0.199)
## Link function: log
##
## Formula:
## N ~ s(gchl_long, k = k1) + s(depthm, k = k1) + s(y, k = k1) +
##     offset(off.set)
## <environment: 0x10af7f468>
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.6640     0.0837    -175   <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(gchl_long) 4.71      9   97.2 < 2e-16 ***
## s(depthm)    2.37      9   27.7 1.7e-08 ***
## s(y)         3.48      9   40.1 2.2e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.129   Deviance explained = 37.7%
## REML score = 2066.9  Scale est. = 1         n = 2067
```

The Q-Q plot for this model looks rather good!

```
gam.check(loon.model.hr)
```

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 1 iteration.
```

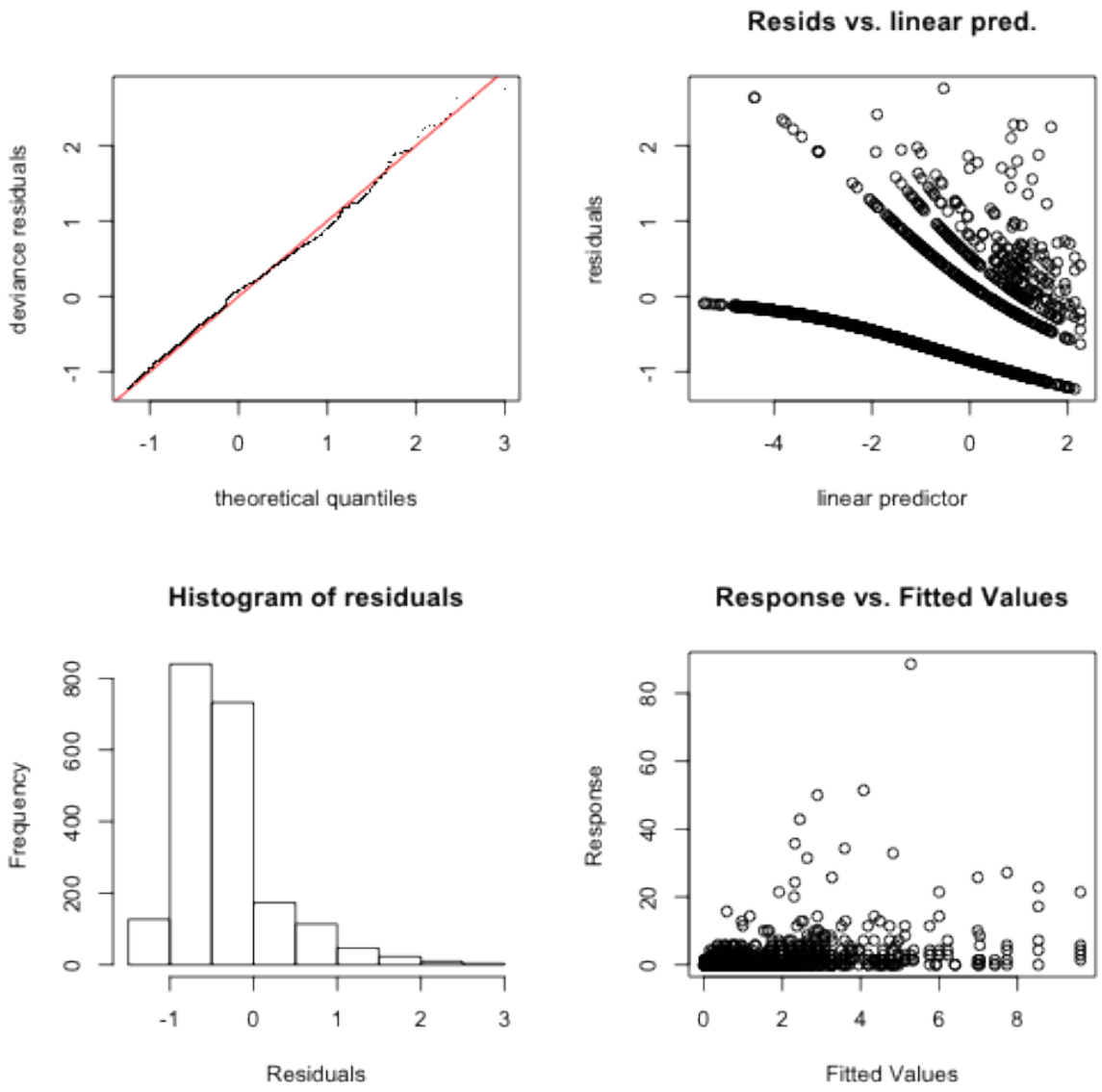

Figure S9: GAM check plot for model `loon.model.hr`.

```
## Gradient range [-8.916e-05,7.864e-06]
## (score 2067 & scale 1).
## Hessian positive definite, eigenvalue range [2.877e-05,1.321].
##
## 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(gchl_long) 9.000 4.711   0.712    0.00
## s(depthm)    9.000 2.374   0.719    0.02
## s(y)         9.000 3.485   0.718    0.02
```

Predicting the abundance over the OSAMP area and the corresponding confidence interval using the method of Williams et al (2011).

```
summary(dsm.var.prop(loon.model.hr, pred, pred$cellaream))
```

```
## Summary of uncertainty in a density surface model calculated
##  by variance propagation.
##
## Quantiles of differences between fitted model and variance model
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -5.77e-04 -5.10e-06 -3.00e-07 -1.60e-06  3.60e-06  1.41e-03
##
## Approximate asymptotic confidence interval:
##   5% Mean  95%
## 3993 5047 6379
## (Using delta method)
##
## Point estimate                : 5047
## Standard error                : 605.4
## Coefficient of variation      : 0.12
```

3.5. Comparison of DSMs

We now compare the models fitted above. Table S2 summarises the results from model fitting, as well as predictions over the OSAMP area. Also in the table are the EDFs of the smooth terms in each model (which are all fairly similar).

3.5.1. Comparison of smooth terms

We note that all of the models selected the same covariates (`gchl_long`, `depth` and `y`), which is encouraging in the sense that there appears to be stability in covariate selction, invariant to the choice of detection function. As Fig. S10 shows that there are minimal differences in the smooths per-covariate. We also see that the confidence bands around the smooths overlap almost all of the time.

3.5.2. Predictive plots and uncertainty plots

Comparing predictive abundance maps over the OSAMP area and corresponding uncertainty maps, we see similar results between models (Fig. S11 and Fig. S12).

Table S2: Table comparing DSM results.

Model	Geometric mean of chlorophyll EDF	Depth EDF	Northing (y) EDF	Abundance estimate	CV	Variance propagation	% deviance explained	adj-R^2
loon.model.size	4.34	2.83	3.46	5760	0.128	No	29.299	0.077
loon.model.obs.size	4.27	2.85	3.47	5707	0.127	No	29.081	0.086
loon.model.obs	4.02	2.86	3.35	5127	0.12	No	27.863	0.127
loon.model.hr	4.71	2.37	3.48	5047	0.12	Yes	37.739	0.129

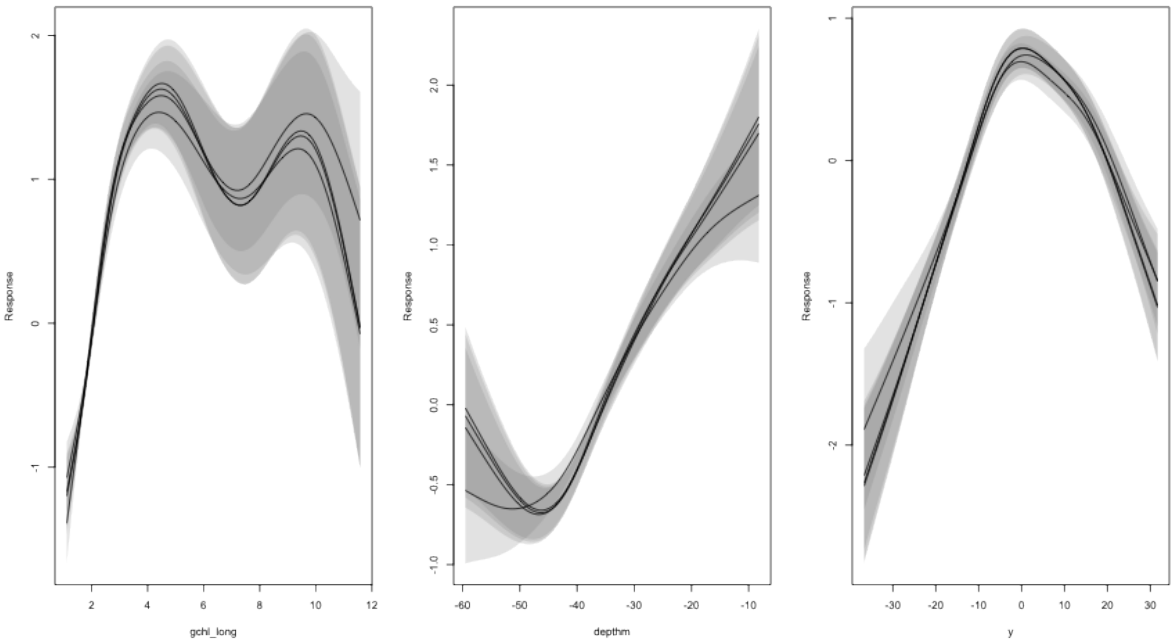

Figure S10: Overlaid model plots of per-covariate smooths with confidence intervals.

3.5.3. Comparison of point estimates and confidence intervals

In addition to the plots above, we can also look at the abundance point estimates (i.e. the sums of the predicted abundance maps, above), along with their confidence intervals...

```
y.vals <- 1:4
plot(p.ci[, 2], y.vals, pch = 19, axes = FALSE, xlim = range(p.ci) - c(600,
    0), ylab = "", xlab = "Abundance")
axis(1)
axis(2, at = y.vals, labels = nm, las = 2, lwd = 0, hadj = 0)
segments(x0 = p.ci[, 1], x1 = p.ci[, 3], y0 = y.vals)
```

Drawing some conclusions from the above plots and tables, we see that the best model in predictive power terms (adjusted-R^2 and percentage deviance explained) is the hazard-rate model with no covariates.

3.5.4. Final model selection

Given the relatively small differences observed between the models, there is not a huge body of evidence swaying the investigator to one over the other. However, we choose the hazard-rate detection function with no covariates for the following reasons:

- The DSM has a higher adjusted-R^2 and percentage deviance explained than the other models. The abundance predicted from the model had the lowest coefficient of variation.
- The hazard-rate detection function requires less parameters and differs by less than two AIC points from the more complex models.
- Using a model without covariates allows us to use the variance propagation method of Williams et al. (2011), giving us a more reliable estimate of the variance in the predicted abundance.

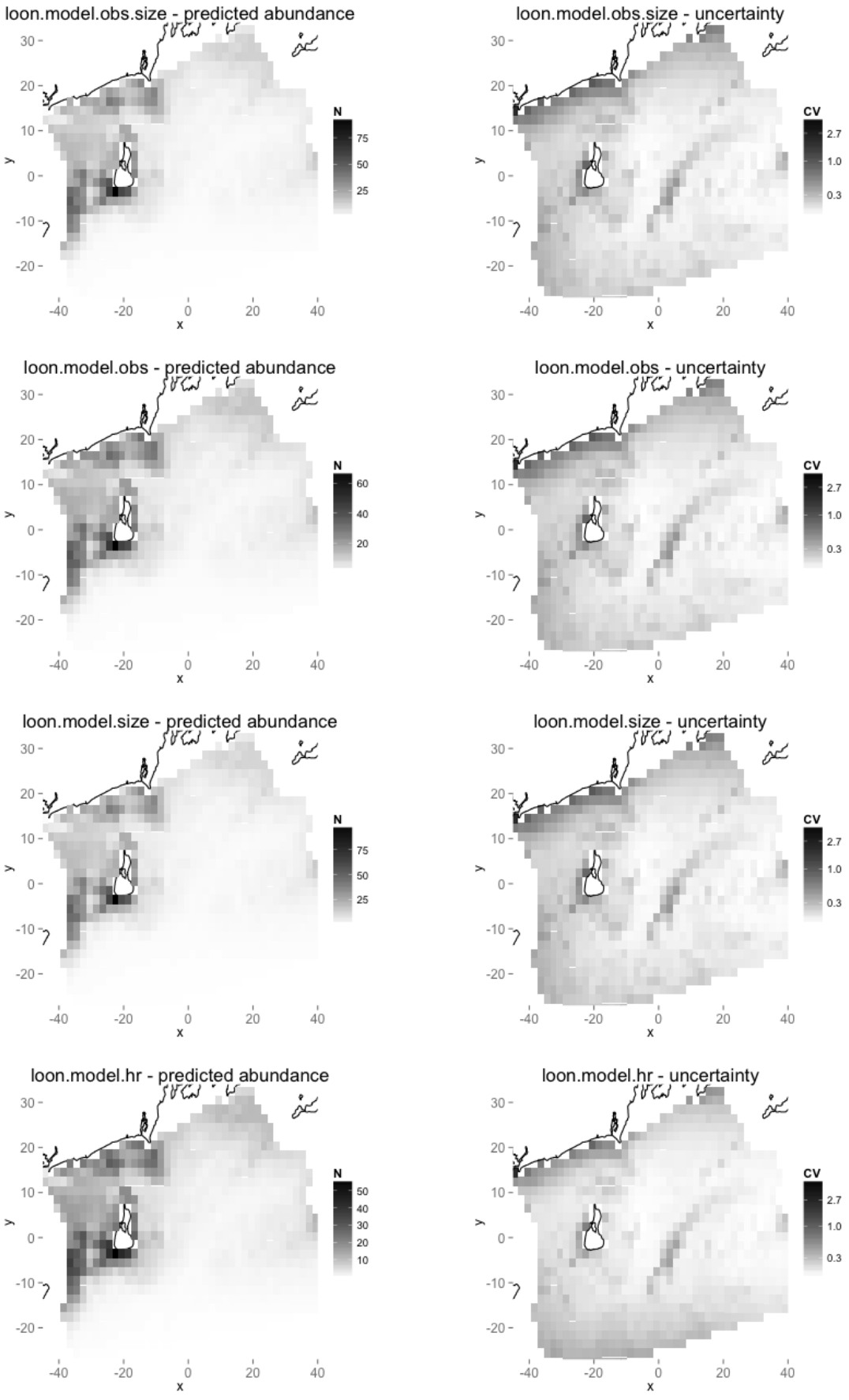

Figure S11: Predicted abundance and uncertainty plots for all models.

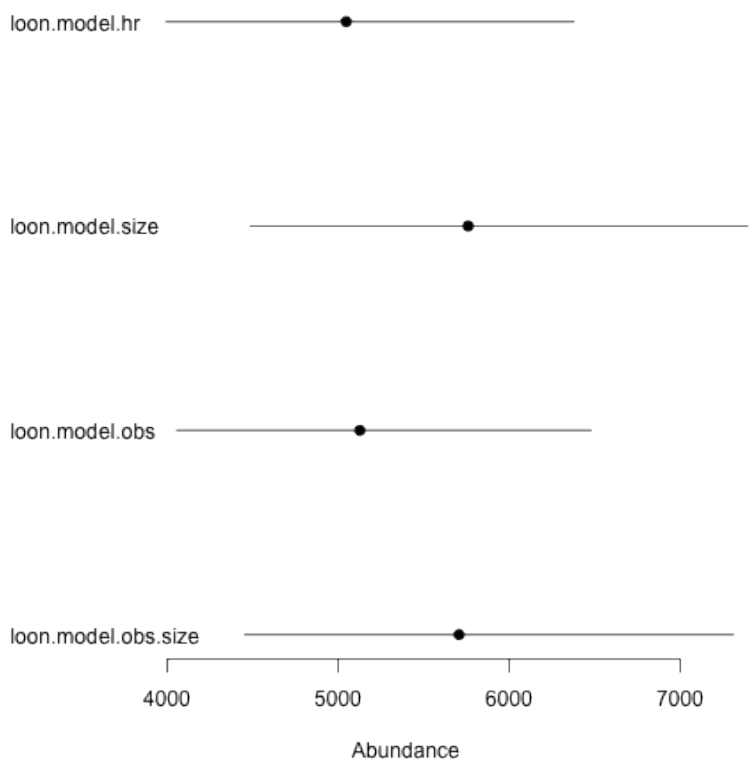

Figure S12: Plot of predicted abundances and confidence intervals for the DSMs.

- Predicted abundances, maps of coefficients of variation, smooth curves and average detection probabilities were very similar between models.

3.6. Sensitivity analysis

Sensitivity – DSM without `gchl_long`. To check that the other measures of chlorophyll *a* were not simply removed due to high correlation with `gchl_long`, we refit the model without `gchl_long`. After covariate selection we have:

```
loon.model.nogchl_long <- dsm(N~#s(gchl_winter,k=k1)+
                               #s(fcpi,k=k1)+
                               #s(roughness,k=k1)+
                               #s(phimedian,k=k1)+
                               s(distancelandkm,k=k1)+
                               #s(depthm,k=k1)+
                               depthm+
                               #s(x,k=k1)+
                               s(y,k=k1)+
                               s(x,y,k=k2),
                             hr.df, seg, obs.loons,
                             family=negbin(theta=c(0.1,0.4)), availability=0.7,
                             select=TRUE, method="REML")

summary(loon.model.nogchl_long)
```

```
##
## Family: Negative Binomial(0.302)
## Link function: log
##
## Formula:
## N ~ s(distancelandkm, k = k1) + depthm + s(y, k = k1) + s(x,
##     y, k = k2) + offset(off.set)
## <environment: 0x11894b6a8>
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -13.0248     0.3982  -32.71  < 2e-16 ***
## depthm        0.0448     0.0106    4.23  2.4e-05 ***
## ---
## 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(distancelandkm) 1.78      9   9.49 3.8e-05 ***
## s(y)              4.25      9  15.69 1.4e-06 ***
## s(x,y)            6.88     16  44.32 3.9e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.134   Deviance explained = 39.8%
## REML score = 2050.9  Scale est. = 1         n = 2067
```

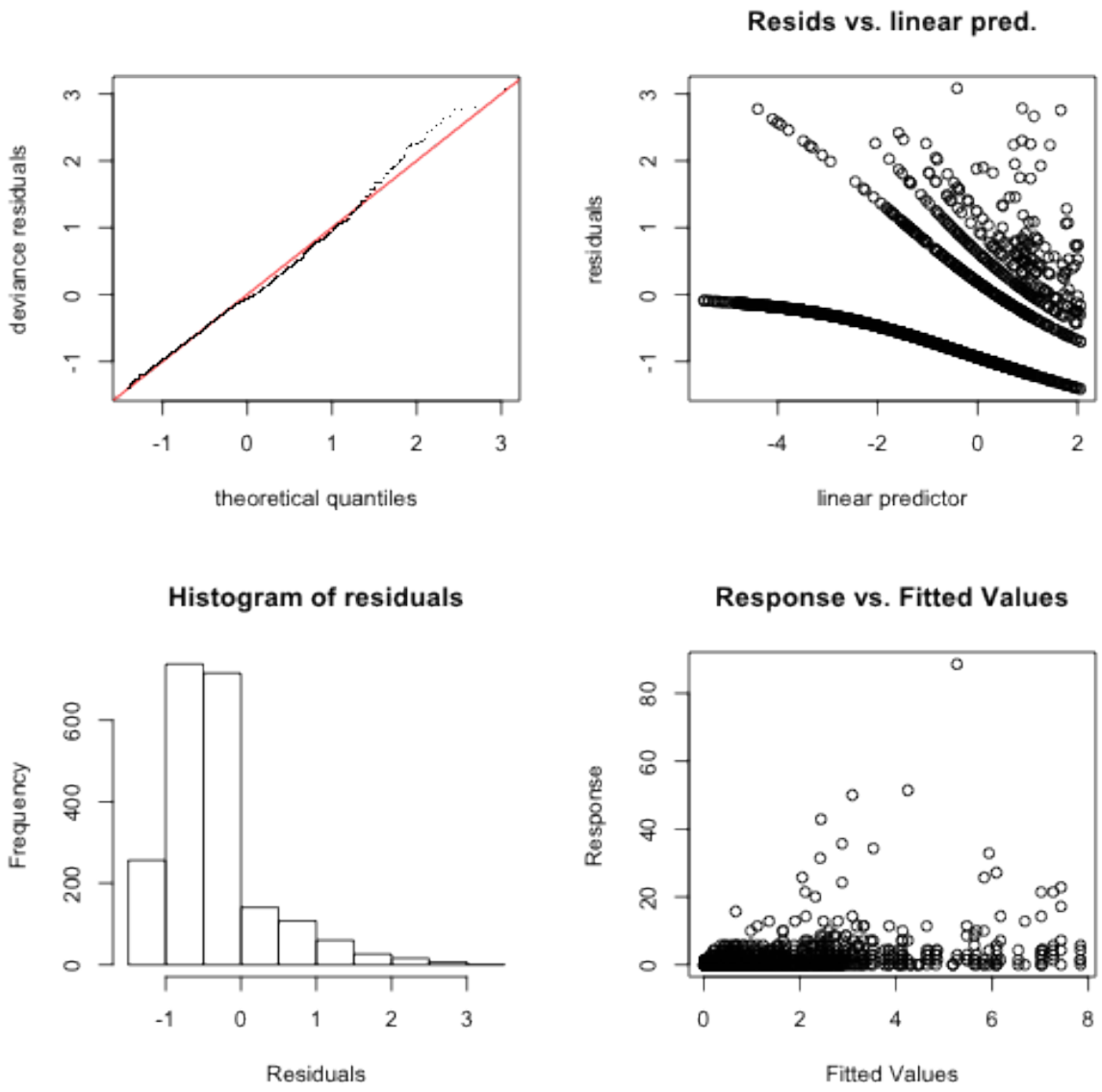

Figure S13: Model checking plot for sensitivity analysis model without long term chlorophyll *a* (no `gchl_long`).

```
gam.check(loon.model.nogchl_long)
```

A check for this model is shown in Fig. S13.

```
##
## Method: REML   Optimizer: outer newton
## step failed after 3 iterations.
## Gradient range [-0.006266,-8.08e-05]
## (score 2051 & scale 1).
## Hessian positive definite, eigenvalue range [8.078e-05,1.702].
##
## 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(distancelandkm)  9.000  1.778   0.739    0.01
## s(y)               9.000  4.250   0.750    0.05
## s(x,y)            16.000  6.879   0.725    0.00
```

```
summary(dsm.var.prop(loon.model.nogchl_long, pred, pred$cellaream))
```

```
## Summary of uncertainty in a density surface model calculated
##  by variance propagation.
##
## Quantiles of differences between fitted model and variance model
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.47e-03 -3.75e-05  7.00e-07 -1.69e-05  4.95e-05  5.60e-04
##
## Approximate asymptotic confidence interval:
##   5% Mean  95%
## 1471 2474 4160
## (Using delta method)
##
## Point estimate                 : 2474
## Standard error                 : 667.6
## Coefficient of variation       : 0.2698
```

The terms selected here are all clearly highly concurve: we can write all of the smooth terms as smooth functions of each other. This may well explain why the corresponding predicted abundance is so far from the above model and that given in other literature.

Sensitivity – DSM without `gchl_long` *or bivariate* `x` *and* `y` *smooth.* The above model includes both a univariate smooth of `y` and a bivariate smooth of `x` and `y`, which seems rather redundant. Removing the bivariate term from the outset yields:

```
loon.model.nogchl_longxy <- dsm(N~#s(gchl_winter,k=k1)+
                                 #s(fcpi,k=k1)+
                                 s(roughness,k=k1)+
                                 #s(phimedian,k=k1)+
                                 #s(distancelandkm,k=k1)+
                                 #s(depthm,k=k1)+
```

```
                                depthm+
                                s(x,k=k1)+
                                s(y,k=k1),
                            hr.df, seg, obs.loons,
                            family=negbin(theta=c(0.1,0.4)), availability=0.7,
                            select=TRUE, method="REML")

summary(loon.model.nogchl_longxy)

##
## Family: Negative Binomial(0.31)
## Link function: log
##
## Formula:
## N ~ s(roughness, k = k1) + depthm + s(x, k = k1) + s(y, k = k1) +
##     offset(off.set)
## <environment: 0x11a303028>
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.6979     0.3791  -33.50  < 2e-16 ***
## depthm        0.0542     0.0101    5.35  8.8e-08 ***
## ---
## 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(roughness) 3.50      9   17.1 0.00043 ***
## s(x)         5.73      9  116.0 < 2e-16 ***
## s(y)         4.73      9   97.4 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.117   Deviance explained = 40.4%
## REML score = 2052.4  Scale est. = 1         n = 2067

gam.check(loon.model.nogchl_longxy)
```

A check for this model is shown in Fig. S14.

```
##
## Method: REML   Optimizer: outer newton
## step failed after 3 iterations.
## Gradient range [-0.002874,0.002815]
## (score 2052 & scale 1).
## eigenvalue range [-2.43e-05,1.597].
##
## 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
```

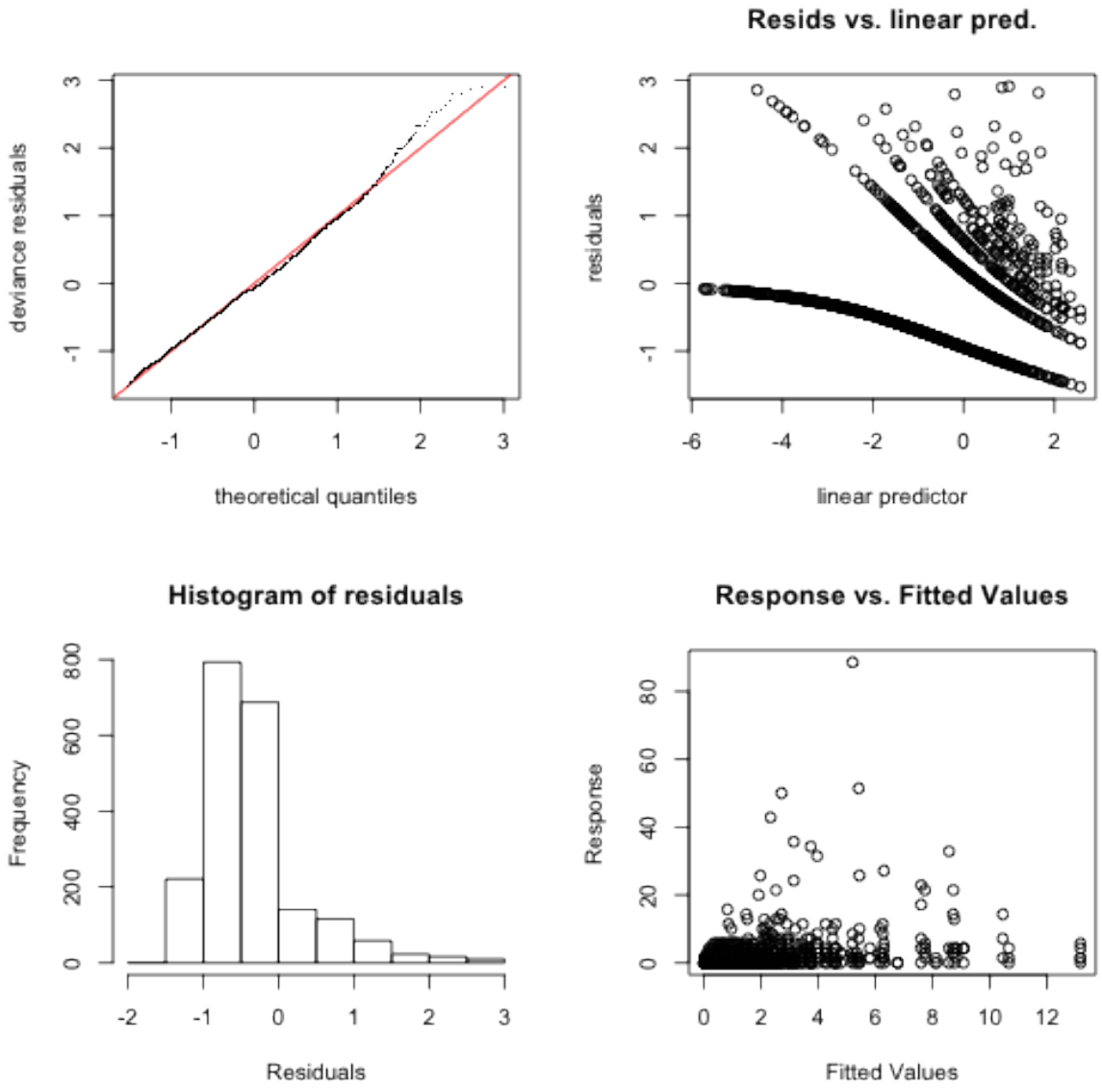

Figure S14: Model checking plot for sensitivity analysis model without long term chlorophyll a or bivariate smooth of location.

```
## s(roughness) 9.000 3.501   0.756    0.06
## s(x)         9.000 5.730   0.759    0.10
## s(y)         9.000 4.729   0.756    0.06
```

This seems to give an abundance estimate more in line with the model that included `gchl_long`.

```
summary(dsm.var.prop(loon.model.nogchl_longxy, pred, pred$cellaream))
```

```
## Summary of uncertainty in a density surface model calculated
##  by variance propagation.
##
## Quantiles of differences between fitted model and variance model
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.49e-03 -7.90e-06  3.00e-07 -6.00e-07  1.03e-05  1.19e-03
##
## Approximate asymptotic confidence interval:
##   5% Mean  95%
## 4182 5123 6277
## (Using delta method)
##
## Point estimate                : 5123
## Standard error                : 532.3
## Coefficient of variation      : 0.1039
```

Neither of the above models includes the other measures of chlorohyll *a*, so it is safe to conclude that there are not issues with confounding between `gchl_long` and `gchl_winter`/`fcpi`.

3.6.1. Checking the response distribution

We can also try other response distributions to check that the negative binomial is appropriate. Again going through the same steps of covariate selection to ensure the model is "optimal" (in some sense).

Quasi-Poisson.

```
loon.model.qp <- dsm(N~s(gchl_long,k=k1)+
                       #s(gchl_winter,k=k1)+
                       #s(fcpi,k=k1)+
                       s(roughness,k=k1)+
                       #s(phimedian,k=k1)+
                       s(distancelandkm,k=k1)+
                       #s(depthm,k=k1)+
                       depthm+
                       s(x,k=k1)+
                       s(y,k=k1),#+
                       #s(x,y,k=k2),
                     hr.df, seg, obs.loons,
                     family=quasipoisson(), availability=0.7,
                     select=TRUE, method="REML")

summary(loon.model.qp)
```

```
##
## Family: quasipoisson
## Link function: log
##
## Formula:
## N ~ s(gchl_long, k = k1) + s(roughness, k = k1) + s(distancelandkm,
##     k = k1) + depthm + s(x, k = k1) + s(y, k = k1) + offset(off.set)
## <environment: 0x110c87ce8>
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -13.17376    0.26610  -49.51  < 2e-16 ***
## depthm        0.04274    0.00688    6.21  6.4e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                    edf Ref.df     F p-value
## s(gchl_long)      6.24      9  5.57 9.6e-11 ***
## s(roughness)      5.68      9  4.48 3.2e-08 ***
## s(distancelandkm) 2.51      9  1.99 5.9e-06 ***
## s(x)              7.21      9 12.92 < 2e-16 ***
## s(y)              3.77      9  4.31 4.2e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.19   Deviance explained = 42.6%
## REML score = 1993.2  Scale est. = 2.4041    n = 2067
```

The Q-Q plot for this model is not nice! (Fig. S15.)

```
gam.check(loon.model.qp)
```

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 11 iterations.
## Gradient range [-0.0007386,0.0001518]
## (score 1993 & scale 2.404).
## eigenvalue range [-6.257e-05,1033].
##
## 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(gchl_long)      9.000 6.244   0.950    0.70
## s(roughness)      9.000 5.677   0.964    0.86
## s(distancelandkm) 9.000 2.507   0.942    0.50
## s(x)              9.000 7.214   0.956    0.78
## s(y)              9.000 3.766   0.956    0.78
```

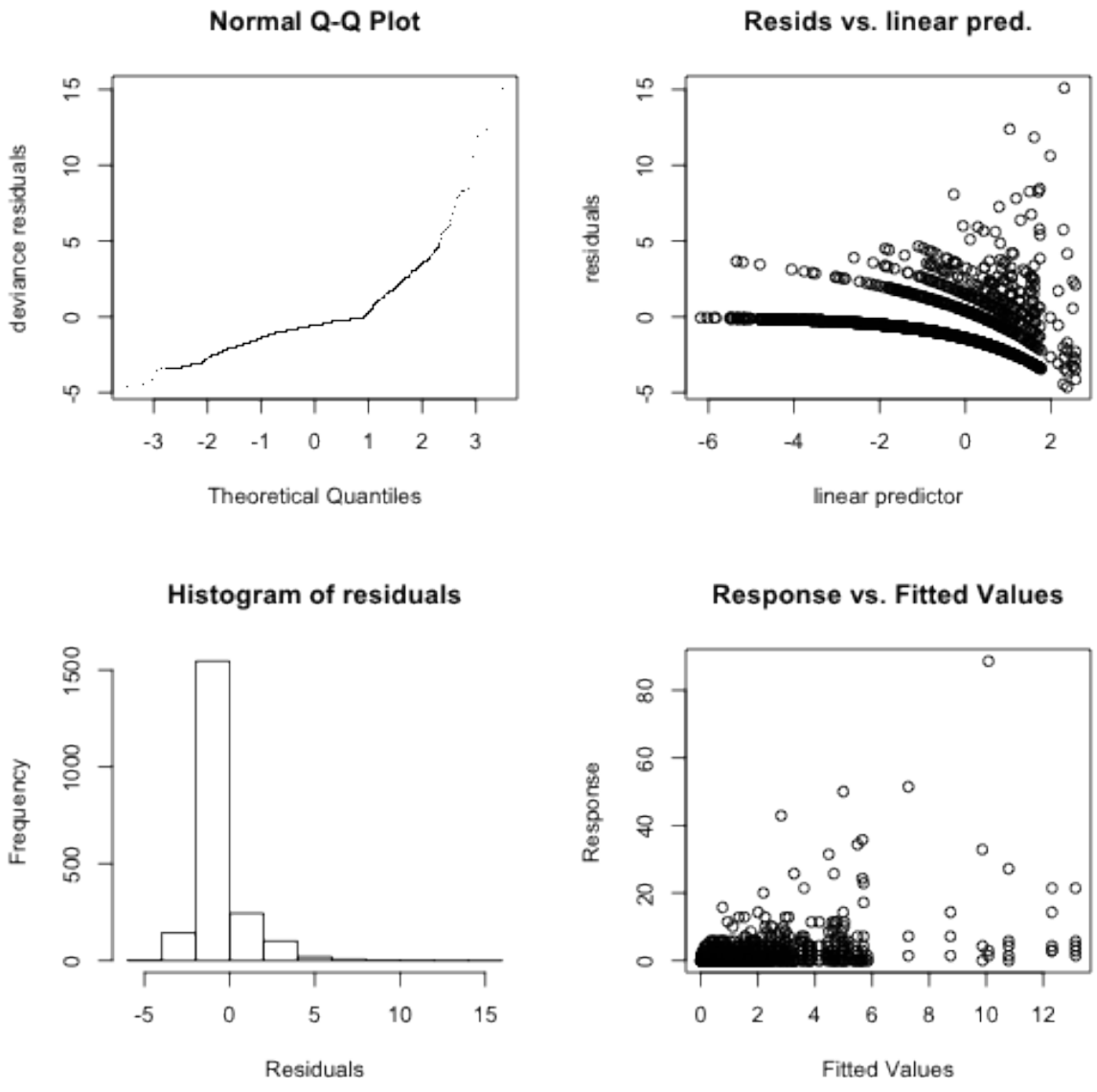

Figure S15: Model checking plot for model using quasi-Poisson response distribution

The quasi-Poisson model also gives an extremely large abundance estimate and confidence interval, it seems safe to discard this model.

```
summary(dsm.var.prop(loon.model.qp, pred, pred$cellaream))
```

```
## Summary of uncertainty in a density surface model calculated
##  by variance propagation.
##
## Quantiles of differences between fitted model and variance model
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -7.16e-13 -1.85e-14 -6.00e-16 -1.25e-14  3.40e-15  3.34e-13
##
## Approximate asymptotic confidence interval:
##     5%   Mean    95%
##   1936  17789 163427
## (Using delta method)
##
## Point estimate                : 17789
## Standard error                : 28672
## Coefficient of variation      : 1.612
```

Tweedie.

```
loon.model.tw <- dsm(N~s(gchl_long,k=k1)+
                       #s(gchl_winter,k=k1)+
                       #s(fcpi,k=k1)+
                       s(roughness,k=k1)+
                       #s(phimedian,k=k1)+
                       s(distancelandkm,k=k1)+
                       #s(depthm,k=k1)+
                       depthm+
                       s(x,k=k1)+
                       s(y,k=k1),#+
                       #s(x,y,k=k2),
                     hr.df, seg, obs.loons,
                     family=Tweedie(p=1.1), availability=0.7,
                     select=TRUE, method="REML")
```

```
summary(loon.model.tw)
```

```
##
## Family: Tweedie(1.1)
## Link function: log
##
## Formula:
## N ~ s(gchl_long, k = k1) + s(roughness, k = k1) + s(distancelandkm,
##     k = k1) + depthm + s(x, k = k1) + s(y, k = k1) + offset(off.set)
## <environment: 0x1110853c8>
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -13.11388    0.26637  -49.23   <2e-16 ***
## depthm        0.04372    0.00699    6.25    5e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                     edf Ref.df     F p-value
## s(gchl_long)       6.08      9  5.22 3.8e-10 ***
## s(roughness)       5.77      9  4.43 4.7e-08 ***
## s(distancelandkm) 2.37      9  1.58 5.6e-05 ***
## s(x)               7.12      9 11.70 < 2e-16 ***
## s(y)               3.78      9  5.02 8.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.186   Deviance explained = 42.9%
## REML score = 2150.5  Scale est. = 2.3214    n = 2067
```

To find the p parameter for the Tweedie distribution we need to manually search over values of p, which range from 1.1 to 1.9 and are relatively invariant to changes beyond the first decimal place... We can use AIC to decide on a best value.

```
for (p in seq(1.1, 1.9, by = 0.1)) {
    loon.model.tw.test <- dsm(N ~ s(gchl_long, k = k1) + s(roughness, k = k1) +
        s(distancelandkm, k = k1) + depthm + s(x, k = k1) + s(y, k = k1), hr.df,
        seg, obs.loons, family = Tweedie(p = p), availability = 0.7, select = TRUE,
        method = "REML")
    cat("p=", p, "AIC=", AIC(loon.model.tw.test), "\n")
}
```

```
## p= 1.1 AIC= 4239
## p= 1.2 AIC= 4239
## p= 1.3 AIC= 4333
## p= 1.4 AIC= 4454
## p= 1.5 AIC= 4611
## p= 1.6 AIC= 4804
## p= 1.7 AIC= 5064
## p= 1.8 AIC= 5438
## p= 1.9 AIC= 6073
```

The smallest AIC is given by a value of p=1.1, as used above.

Again, the Q-Q plot for this model is not as good as the Q-Q plot for the negative binomial model above (Fig. S16), showing significant divergance from the theoretical residuals.

```
gam.check(loon.model.tw)
```

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 12 iterations.
## Gradient range [-0.0009476,0.0001564]
```

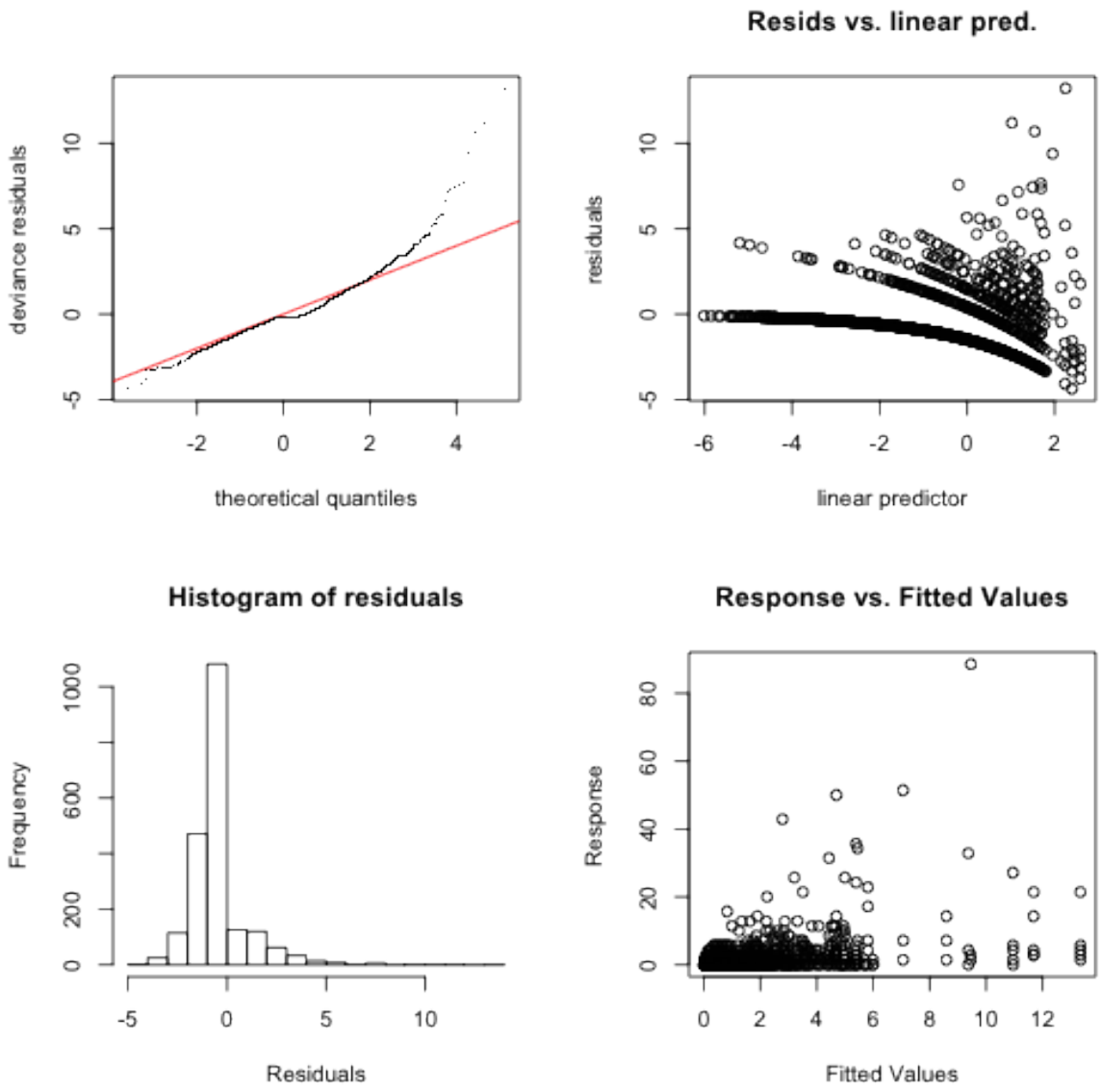

Figure S16: Model checking plot for model with Tweedie response distribution.

```
## (score 2150 & scale 2.321).
## eigenvalue range [-0.0001056,2840].
##
## 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(gchl_long)      9.000 6.077   0.923    0.65
## s(roughness)      9.000 5.771   0.934    0.75
## s(distancelandkm) 9.000 2.370   0.914    0.44
## s(x)              9.000 7.121   0.928    0.71
## s(y)              9.000 3.776   0.927    0.66
```

Predicted abundance is also rather large in comparison to the negative binomial model.

```
summary(dsm.var.prop(loon.model.tw, pred, pred$cellaream))
```

```
## Summary of uncertainty in a density surface model calculated
##  by variance propagation.
##
## Quantiles of differences between fitted model and variance model
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -2.40e-11 -5.21e-13  0.00e+00 -1.90e-14  7.54e-13  1.97e-11
##
## Approximate asymptotic confidence interval:
##    5%  Mean   95%
##  1593  9474 56340
## (Using delta method)
##
## Point estimate                : 9474
## Standard error                : 10750
## Coefficient of variation      : 1.135
```

3.6.2. Sensitivity to availability correction

Looking at the negative binomial model, again we can test the sensitivity of the model to values of the availability correction factor.

```
avail.range <- seq(0.5,1,by=0.05)

N.ci.res <- c()

for(this.avail in avail.range){
  this.loon.model <- dsm(N~s(gchl_long,k=k1)+
                           s(depthm,k=k1)+
                           s(y,k=k1),#+
                         hr.df, seg, obs.loons,
                         family=negbin(theta=c(0.1,0.4)),availability=this.avail,
                         select=TRUE, method="REML")
  var.est <- summary(dsm.var.prop(this.loon.model,pred,pred$cellaream))

  cv2 <- var.est$cv^2
```

```
  asymp.ci.c.term <- exp(1.96*sqrt(log(1+cv2)))
  asymp.tot <- c(sum(var.est$pred.est) / asymp.ci.c.term,
                 sum(var.est$pred.est),
                 sum(var.est$pred.est) * asymp.ci.c.term)
  names(asymp.tot) <- c("5%","Mean","95%")
  N.ci.res<-rbind(N.ci.res,asymp.tot)
}
```

Fig. S17 shows that there is a clear relationship between abundance and the availability bias correction factor. This can be seen from examining the model equation for the spatial model, where the correction factor effects only the offset of the model.

```
plot(avail.range, N.ci.res[, 2], pch = 19, type = "n", xlim = c(0.45, 1), ylim = c(min(N.ci.res),
    max(N.ci.res)), xlab = "Availability bias correction", ylab = "Abundance")
segments(x0 = avail.range, x1 = avail.range, y0 = N.ci.res[, 1], y1 = N.ci.res[,
    3])

polygon(x = c(0.3, 1.1, 1.1, 0.3, 0.3), y = N.ci.res[5, c(3, 3, 1, 1, 3)], col = rgb(166/255,
    189/255, 219/255, 0.2), border = NA)

points(avail.range, N.ci.res[, 2], pch = 19)
```

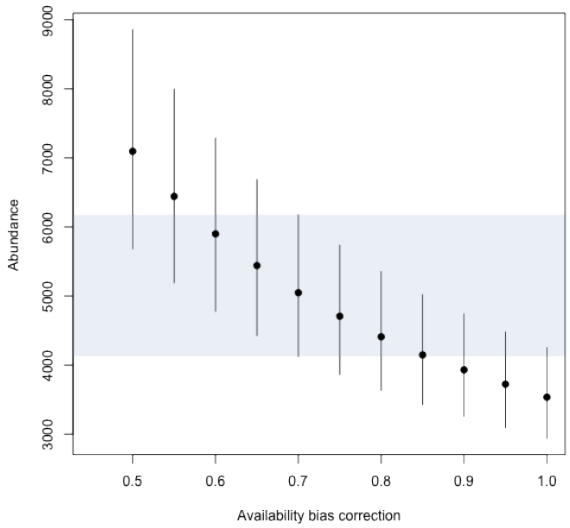

Figure S17: Plot of predicted abundances and corresponding confidence intervals when the availability bias is varied between 0.5 and 1.

3.7. Conclusion

Model selection and sensitivity testing shows that including the long term average of chlorophyll *a* (`gchl_long`), using the negative binomial distribution as response gives robust estimates of the abundance and distribution of common loons in the Rhode Island OSAMP region.

3.8. Save results

```
save.image("MEPS-loons-models.RData")
```

4. References

Miller DL (2012) Distance: A simple way to fit detection functions to distance sampling data and calculate abundance/density for biological populations. R package version 0.7.1. http://CRAN.R-project.org/package=Distance

Miller DL (2013) dsm: Density surface modelling of distance sampling data. R package version 2.0.1. http://CRAN.R-project.org/package=dsm

Suryan RM, Santora JA, Sydeman WJ (2012) New approach for using remotely sensed chlorophyll a to identify seabird hotspots. Mar Ecol Prog Ser 451:213?225

Williams R, Hedley SL, Branch TA, Bravington MV, Zerbini AN, Findlay KP (2011) Chilean blue whales as a case study to illustrate methods to estimate abundance and evaluate conservation status of rare species. Conserv Bio 25:526–535

Winiarski KJ, Burt LM, Rexstad EA, Miller DL, Trocki CL, Paton PWC, McWilliams SR (2014) Integrating aerial and ship surveys of marine birds into a combined density surface model: a case study of wintering Common Loons. Condor (in press)

Wood SN (2006) Generalized Additive Models: An Introduction with R. Chapman and Hall/CRC Press, Boca Raton, FL.

Wood SN (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. J R Stat Soc B 73:3-36

Xie Y (2013) knitr: A general-purpose package for dynamic report generation in R. R package version 1.0. http://CRAN.R-project.org/package=knitr