

Whale Wave: Habitat Modeling

February 27, 2016

Contents

1	Dataset	2
2	Data Exploration	4
2.1	Outliers	4
2.1.1	Observed variable: Humpbacks	4
2.1.2	Explanatory variables: backscatter	5
2.2	Collinearity	5
3	Data modification	8
3.1	Remove outliers and collinear predictors	8
3.2	Standardize explanatory variables	8
3.2.1	Standardize explanatory variables	8
3.2.2	Prepare Offset variable	10
4	Model design	11
4.1	Distribution family	11
4.2	Game plan	12
4.3	Pure Prey Following	16
4.3.1	Relative Variable Importance using Akaike Weights	16
4.3.2	95% confidence set using Akaike Weights	16
4.4	Pure Habit	16
4.5	Environmental model	17
4.6	Foraging strategy	17
4.6.1	95% confidence set using Akaike Weights	18
4.7	Full habitat use	18
5	Best-fit models	20
5.1	Model validation	21
6	Smoothing functions figure	23
7	Leg by Leg Comparison	24
7.1	Setup variable sets	24
7.2	Setup monthly datasets	25
7.3	Model fitting	25
7.4	Relative Variable Importance	26
7.5	Prediction Error	28
7.5.1	Prediction Error figure	29
7.5.2	Best-fit models	29

1 Dataset

```

> data <- read.csv("trn-bins-2015-5.csv",stringsAsFactors=FALSE)
> # Exclude CMF and SQF
> data <- data[!data$block%in%c("CMF","SQF"),]
> # Exclude Leg 3!
> data <- data[!data$leg==3,]
> # Prepare factors
> data$leg <- as.factor(data$leg)
> data$block <- as.character(data$block)
> data$block <- as.factor(data$block)
> # Column key
> preycol <- 16:23
> habitcol <- c(4,5,6,7,8,13)
> envcol <- c(24:28)
> poscol <- c(4,5,6,7,8,11,13:15)
> # Remove NAs and 0s
> data <- data[data$eff!=0 &
+             !is.na(data$hw) &
+             !is.na(data$eff) &
+             !is.na(data$X) &
+             !is.na(data$Y) &
+             !is.na(data$OD) &
+             !is.na(data$I.hi) &
+             !is.na(data$I.lo) &
+             !is.na(data$M.hi) &
+             !is.na(data$M.lo) &
+             !is.na(data$D.hi) &
+             !is.na(data$D.lo) &
+             !is.na(data$sst) &
+             !is.na(data$sss) &
+             !is.na(data$chl) &
+             !is.na(data$secchi) &
+             !is.na(data$inlet) ,]
> setwd(curdir)

```

	jul	leg	block	X	Y	OD	hw	eff	I.hi	M.hi
1	151	1	CAA	-129.4118	52.98154	91.8399	0	6.434970	12.2247	248.9394
2	151	1	CAA	-129.3943	52.94628	84.6729	0	6.419487	1354.3191	154.1501
3	151	1	CAA	-129.3566	52.92390	83.0088	0	6.400050	1837.2020	142.3119
4	151	1	CAA	-129.3032	52.90745	82.1605	0	5.158003	583.1664	134.4320
5	151	1	CAA	-129.3080	52.85992	87.2552	0	2.477410	1800.1434	136.5063
6	151	1	CAA	-129.2610	52.84556	87.8737	0	5.155597	2165.0730	137.7188
	C.hi	D.hi	I.lo	M.lo	C.lo	D.lo	sst	sss		
1	34.676012	0.8553884	193.73103	216.4176	3.532878	2.061151	10.91985	29.78565		
2	2.098564	8.4715780	279.47792	172.0914	2.299209	32.547263	11.08053	29.61390		
3	1.505858	10.9476568	80.15758	165.0073	3.972373	26.129538	11.27107	29.15403		
4	2.672079	15.1210623	108.04499	176.5945	1.581366	40.345801	11.86306	28.03992		
5	1.777071	11.7423510	257.19162	179.3361	1.340929	31.904652	11.85052	28.63781		
6	1.116974	9.2703684	175.71355	164.6961	1.874366	37.966058	11.85132	28.72181		
	inlet	chla	secchi							
1	20.5831	15.16125	12.04529							
2	17.7133	14.26091	12.24549							
3	14.4584	13.07419	12.56048							

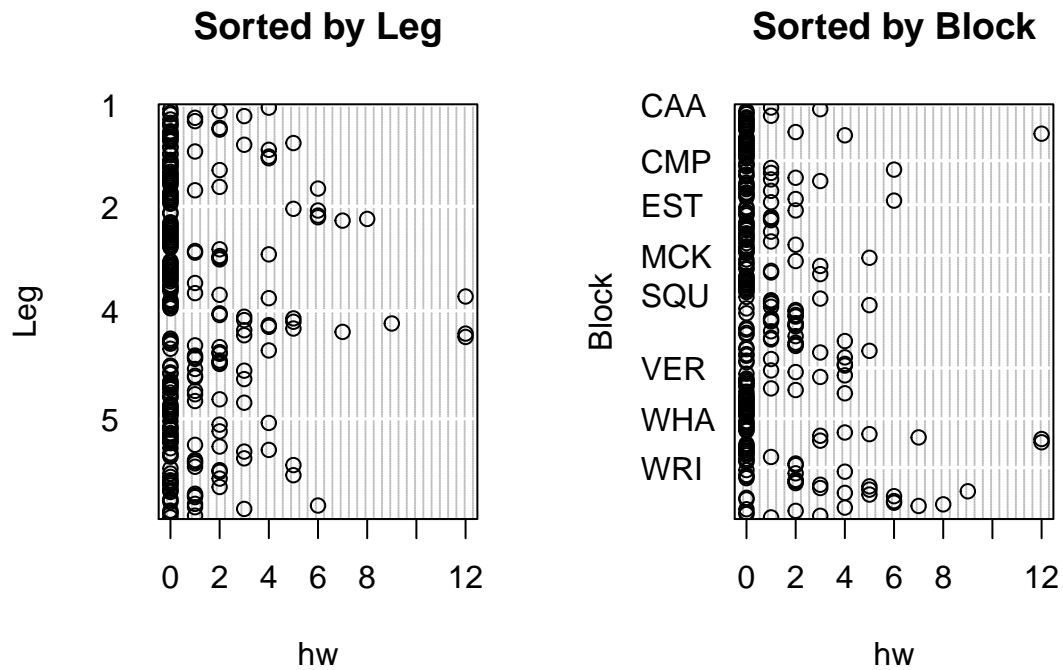
4	10.4926	11.04515	13.04556
5	10.7853	12.03973	12.74519
6	8.4221	12.07881	12.82413

2 Data Exploration

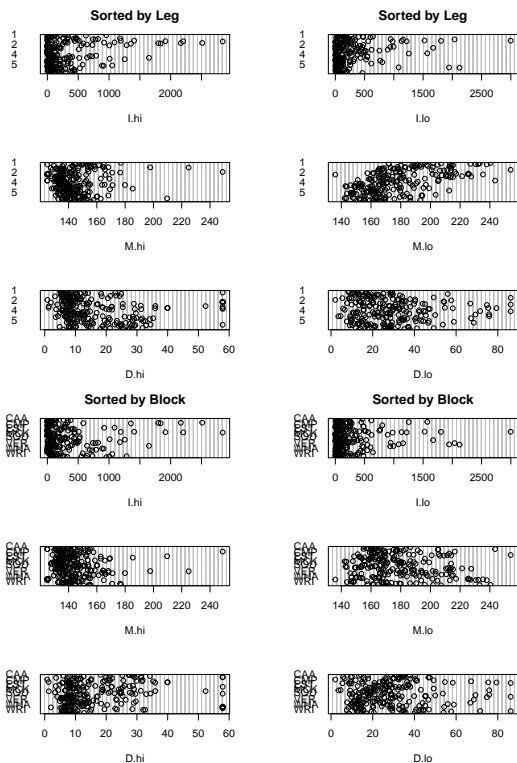
2.1 Outliers

2.1.1 Observed variable: Humpbacks

Larger numbers of humpbacks are observed in legs 2 and 4 (Each leg has an outlier than is a candidate for removal. The leg 2 outlier is the bubble net group in Caamano. The leg 4 outlier is that crazy whale pile up in Whale Channel). There's much less variance in legs 1 and 5.



2.1.2 Explanatory variables: backscatter



2.2 Collinearity

From Zuur et al. (p.473): “Booth et al. (1994) suggest that correlations between pairs of variables with magnitudes greater than plus or minus 0.5 indicate high collinearity, and we use this rough rule-of-thumb here.” So I am not going to use the patchiness metric for this analysis.

```
> cor(data[,c(1,4:ncol(data))], method = "spearman")
```

	jul	X	Y	OD	hw
jul	1.000000000	-0.04976459	-0.006807152	0.028204365	0.09320464
X	-0.049764587	1.000000000	0.478449163	-0.626803395	-0.09182879
Y	-0.006807152	0.47844916	1.000000000	-0.941790222	0.13278786
OD	0.028204365	-0.62680340	-0.941790222	1.000000000	-0.15109854
hw	0.093204644	-0.09182879	0.132787861	-0.151098537	1.000000000
eff	0.116660818	-0.15298830	-0.131570175	0.078985188	0.21337599
I.hi	-0.129627781	-0.20614729	0.017575242	-0.005339156	0.20556257
M.hi	0.053540792	-0.11971187	0.068278649	-0.043669191	0.01593914
C.hi	0.157857988	0.16740992	0.070957135	-0.062814812	-0.07261269
D.hi	0.280647864	0.02050671	-0.175304187	0.130694445	-0.09193726
I.lo	-0.285715551	-0.10068147	-0.041820086	0.076401113	-0.23008147
M.lo	-0.537486639	0.23894463	0.114371791	-0.154232235	-0.11605732
C.lo	0.356321100	0.04866927	0.008063328	-0.018888314	0.20926201
D.lo	0.102799883	-0.22453570	-0.120367877	0.137183516	0.04998518
ssst	-0.199194882	0.20555054	0.295595467	-0.292271645	0.15324617
sss	0.167583645	-0.54802542	-0.806774314	0.840827030	-0.13192128
inlet	0.043078559	-0.71568250	0.094341873	0.072305581	0.21949608
chla	-0.768820686	0.09602070	0.266623910	-0.245565882	0.05848421

secchi	0.395815495	-0.15420289	-0.279929715	0.254668448	-0.05998266
	eff	I.hi	M.hi	C.hi	D.hi
jul	0.1166608184	-0.1296277814	0.05354079	0.15785799	0.28064786
X	-0.1529882989	-0.2061472857	-0.11971187	0.16740992	0.02050671
Y	-0.1315701748	0.0175752416	0.06827865	0.07095713	-0.17530419
OD	0.0789851876	-0.0053391561	-0.04366919	-0.06281481	0.13069445
hw	0.2133759860	0.2055625725	0.01593914	-0.07261269	-0.09193726
eff	1.0000000000	0.0006649466	0.08800511	-0.01301641	-0.00012151
I.hi	0.0006649466	1.0000000000	0.33066717	-0.64649534	-0.31149021
M.hi	0.0880051149	0.3306671699	1.00000000	-0.14404920	-0.13797719
C.hi	-0.0130164115	-0.6464953397	-0.14404920	1.00000000	0.09780301
D.hi	-0.0001215100	-0.3114902077	-0.13797719	0.09780301	1.00000000
I.lo	-0.1034293067	-0.0032953578	0.18883125	-0.04182654	0.05413514
M.lo	-0.0250106928	-0.1405093562	-0.10211955	0.07606578	-0.13546049
C.lo	0.1622977881	0.0347668258	-0.05634556	0.11415319	0.01203483
D.lo	0.0794561421	0.0124078966	-0.14816115	-0.01996059	0.09461185
sst	0.0237265226	-0.2349467691	-0.20627529	0.11633201	-0.07965414
sss	0.0365259999	-0.1238903786	-0.12397552	0.07371936	0.21062398
inlet	0.0594644298	0.2609857500	0.12567861	-0.10379793	-0.12458067
chla	-0.2013757995	0.1989365306	0.06314508	-0.16322433	-0.30430150
secchi	0.1595534883	-0.3340488705	-0.23322993	0.19299762	0.23353106
	I.lo	M.lo	C.lo	D.lo	sst
jul	-0.285715551	-0.53748664	0.356321100	0.10279988	-0.19919488
X	-0.100681469	0.23894463	0.048669275	-0.22453570	0.20555054
Y	-0.041820086	0.11437179	0.008063328	-0.12036788	0.29559547
OD	0.076401113	-0.15423223	-0.018888314	0.13718352	-0.29227164
hw	-0.230081465	-0.11605732	0.209262010	0.04998518	0.15324617
eff	-0.103429307	-0.02501069	0.162297788	0.07945614	0.02372652
I.hi	-0.003295358	-0.14050936	0.034766826	0.01240790	-0.23494677
M.hi	0.188831253	-0.10211955	-0.056345563	-0.14816115	-0.20627529
C.hi	-0.041826544	0.07606578	0.114153194	-0.01996059	0.11633201
D.hi	0.054135145	-0.13546049	0.012034829	0.09461185	-0.07965414
I.lo	1.000000000	0.14421906	-0.711060739	-0.07099180	0.00298334
M.lo	0.144219056	1.00000000	-0.056389888	-0.19933489	0.23172048
C.lo	-0.711060739	-0.05638989	1.000000000	0.10689471	0.00386548
D.lo	-0.070991799	-0.19933489	0.106894715	1.00000000	0.10928568
sst	0.002983340	0.23172048	0.003865480	0.10928568	1.00000000
sss	0.057897629	-0.21766411	0.037098367	0.12930074	-0.33370127
inlet	0.033454555	-0.20492212	0.016978075	0.16791294	-0.01696316
chla	0.227269066	0.45502103	-0.279632408	-0.15014307	0.10535779
secchi	-0.053783560	-0.21286928	0.068765497	0.21020884	0.19883475
	sss	inlet	chla	secchi	
jul	0.16758365	0.04307856	-0.76882069	0.39581549	
X	-0.54802542	-0.71568250	0.09602070	-0.15420289	
Y	-0.80677431	0.09434187	0.26662391	-0.27992971	
OD	0.84082703	0.07230558	-0.24556588	0.25466845	
hw	-0.13192128	0.21949608	0.05848421	-0.05998266	
eff	0.03652600	0.05946443	-0.20137580	0.15955349	
I.hi	-0.12389038	0.26098575	0.19893653	-0.33404887	
M.hi	-0.12397552	0.12567861	0.06314508	-0.23322993	
C.hi	0.07371936	-0.10379793	-0.16322433	0.19299762	
D.hi	0.21062398	-0.12458067	-0.30430150	0.23353106	
I.lo	0.05789763	0.03345456	0.22726907	-0.05378356	
M.lo	-0.21766411	-0.20492212	0.45502103	-0.21286928	

C.lo	0.03709837	0.01697807	-0.27963241	0.06876550
D.lo	0.12930074	0.16791294	-0.15014307	0.21020884
sst	-0.33370127	-0.01696316	0.10535779	0.19883475
sss	1.00000000	0.10050825	-0.33892088	0.35566417
inlet	0.10050825	1.00000000	0.06941708	-0.02000544
chla	-0.33892088	0.06941708	1.00000000	-0.72662216
secchi	0.35566417	-0.02000544	-0.72662216	1.00000000

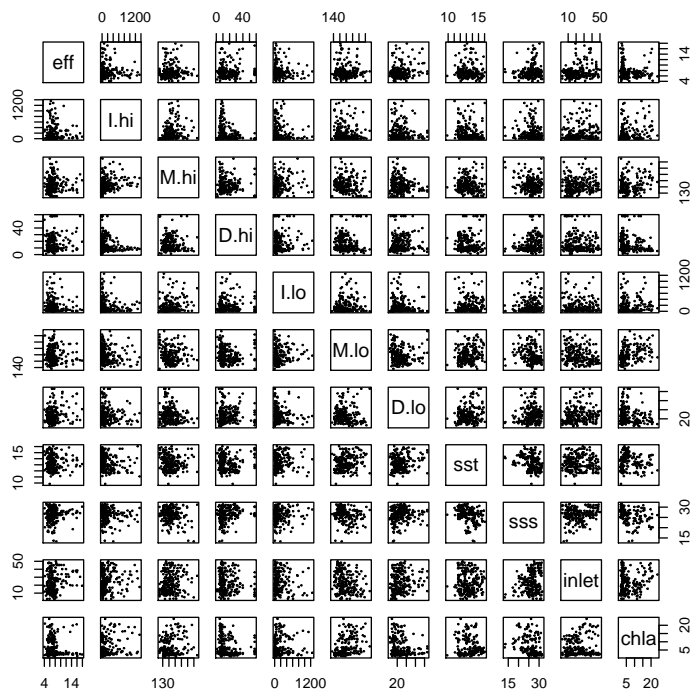
Correlation issues:

- Most of the position variables (X,Y,OD) - obviously. Will only use one kind in each model.
- SSS and OD/y - will have to keep these separate in models.
- M.lo and julian day: -0.52 (would be corrected by in-leg standardization)
- Chla and julian day: -0.71 (same)
- Patchiness metrics are correlated with integrated backscatter. Shouldn't be but they are. Removing Patchiness terms for now.
- Chla and secchi are anticorrelated. Using Chla only.

3 Data modification

3.1 Remove outliers and collinear predictors

```
> # Remove outliers
> newdata <- data[data$hw <= 10 &
+               data$I.hi <= 1500 &
+               data$I.lo <= 1500 &
+               data$M.hi <= 200,]
> # Remove collinear predictors
> newdata$C.hi <- newdata$C.lo <- newdata$secchi <- NULL
```



```
> pairs(newdata[,8:ncol(newdata)],cex=.2)
```

3.2 Standardize explanatory variables

Zuur et al says "Before fitting each of these models to the data, the explanatory variables should be standardised so that they each have a mean of zero and standard deviation of one. This helps to improve convergence of the fitting algorithm and puts the estimated coefficients on the same scale, allowing effect sizes to be more easily compared." (p. 485)

Standardize preserves the spread of the data, but places the variables on similar (though not exactly equal) scales.

3.2.1 Standardize explanatory variables

Now, I could standardize them across the entire season, OR I could standardize explanatory variables WITHIN each leg. Choosing the latter may be a good idea, because...

1. Doing so would reflect the idea that humpbacks are aggregating around the best prey conditions AVAILABLE at any given time, not holding to some season-wide standard.
2. It would clarify interpretation of variables such as time of year and geographic block. If I do not standardize within legs, then variables the best explain season trends in the variables would appear most important, when what we are looking for is insight into trends in the RELATIONSHIP between humpbacks and their prey/habitat. We dont want a model that explains seasonal patterns in the variables, we want a model that elucidates patterns in habitat use strategy.

I decided to standardize backscatter variables within each leg, but standardize environmental variables across the season (sst, sss, chla).

WELL - I ran the process both ways: full-season standardization and by-leg standardization. There was no significant difference in results. In each step, the same models cropped to the top of the 95 set, the top habitat use models were the same, and deviance explained and AIC was about the same throughout. The only differences: RVI for prey: I.lo was more important by about 0.1 in the full-season stdzn, and D.hi was less significant Final best model: chla term was not significant in full-season stdzn, but was (barely) in the leg-by-leg run.

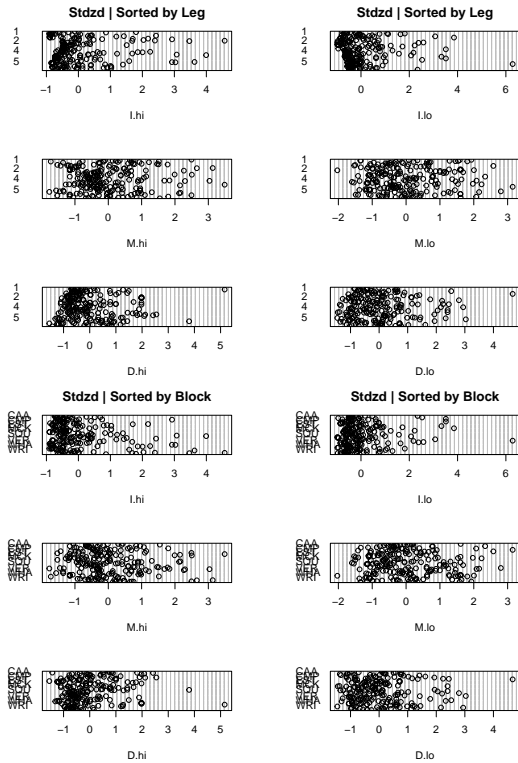
I think I am still going to stick with the Leg by Leg.

```
> #####
> # Full season standardization for sst, sss, OD, X, Y, inletD chla
> newdata$juls <- newdata$jul
> newdata$Xs <- newdata$X
> newdata$Ys <- newdata$Y
> newdata$ODs <- newdata$OD
> cols <- 15:22
> for(i in cols){
+   newdata[,i] <- standardize(newdata[,i])
+ }
> #####
> # Leg-by-leg standardization for backscatter
> cols <- 9:14
> for(i in cols){
+   #newdata[,i] <- standardize(newdata[,i]) # full season standardization
+   newdata[newdata$leg==1,i] <- standardize(newdata[newdata$leg==1,i]) # leg-by-leg standardization
+   newdata[newdata$leg==2,i] <- standardize(newdata[newdata$leg==2,i])
+   newdata[newdata$leg==3,i] <- standardize(newdata[newdata$leg==3,i])
+   newdata[newdata$leg==4,i] <- standardize(newdata[newdata$leg==4,i])
+   newdata[newdata$leg==5,i] <- standardize(newdata[newdata$leg==5,i])
+ }
> #####
> head(newdata)
```

	jul	leg	block	X	Y	OD	hw	eff	I.hi
2	151	1	CAA	-129.3943	52.94628	84.6729	0	6.419487	2.92760221
4	151	1	CAA	-129.3032	52.90745	82.1605	0	5.158003	0.74351202
8	151	1	CAA	-129.1909	52.87676	85.5232	0	5.040933	0.02313504
11	162	1	CMP	-129.2300	53.07911	64.1841	1	6.120328	1.69973122
12	162	1	CMP	-129.2905	53.04965	69.4263	6	8.018912	2.17902122
13	162	1	CMP	-129.2466	53.02769	69.1515	2	13.528481	-0.48479596
				M.hi	D.hi	I.lo	M.lo	D.lo	ssst
2				0.64359735	-0.33084288	0.30170672	-1.3378082	0.8548631	-1.8400816
4				-0.95811395	0.40746991	-0.41164375	-1.1224779	1.4099693	-1.1967393
8				-1.01757784	0.02427807	-0.76883762	-0.8678068	-0.1916187	-0.7072347
11				-0.24734642	0.03748623	-0.38406502	0.3032612	-0.8304472	-1.3514627

12	0.02157703	-0.43331156	0.08239343	0.4329290	-0.3881265	-1.3463113
13	-0.88947455	1.00680864	-0.67261039	0.6634101	-0.7651487	-1.7002837
	sss	inlet	chla	juls	Xs	Ys
2	1.0403438	-0.47103122	1.4538649	-1.467172	-0.73812333	-1.4470419
4	0.5634917	-1.05488018	0.8490723	-1.467172	-0.25027122	-1.6678465
8	0.2619048	-1.68645943	0.9412198	-1.467172	0.35110959	-1.8423635
11	0.6590419	-0.20916557	0.9943820	-1.178094	0.14150098	-0.6916927
12	0.6302530	0.08040191	1.6639934	-1.178094	-0.18226110	-0.8592343
13	0.7419899	-0.28340100	1.7825536	-1.178094	0.05282899	-0.9841086
						ODs

The result of standardization can be seen in this updated dot chart:



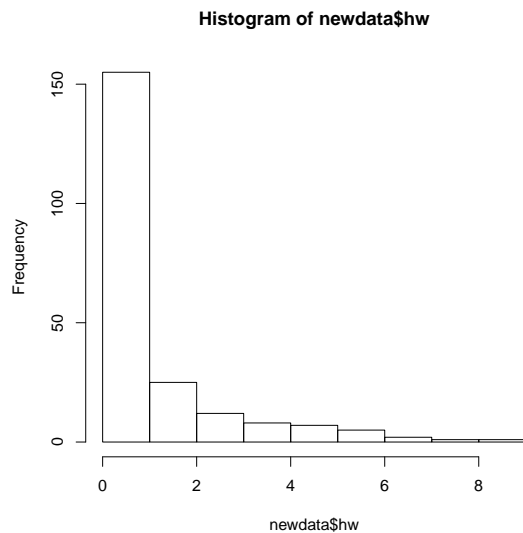
3.2.2 Prepare Offset variable

```
> newdata$off <- log(newdata$eff)
```

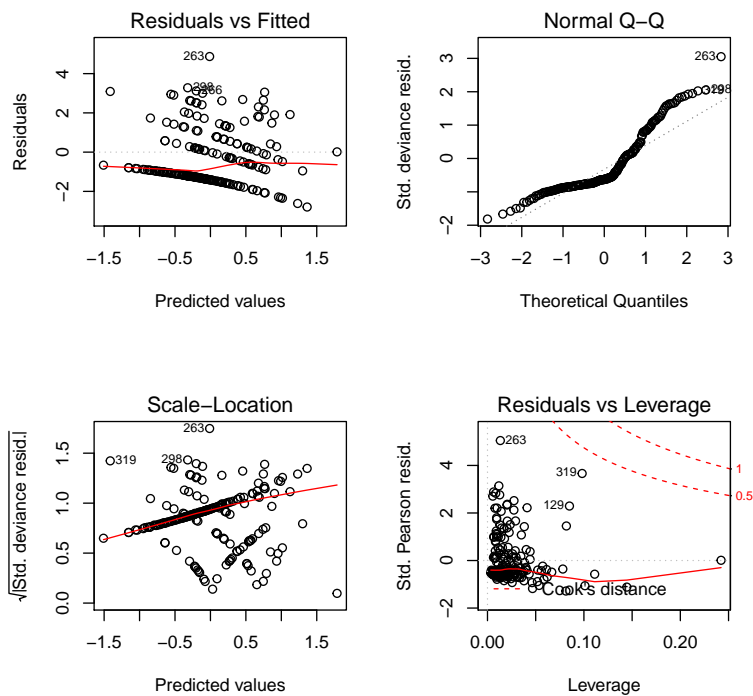
4 Model design

4.1 Distribution family

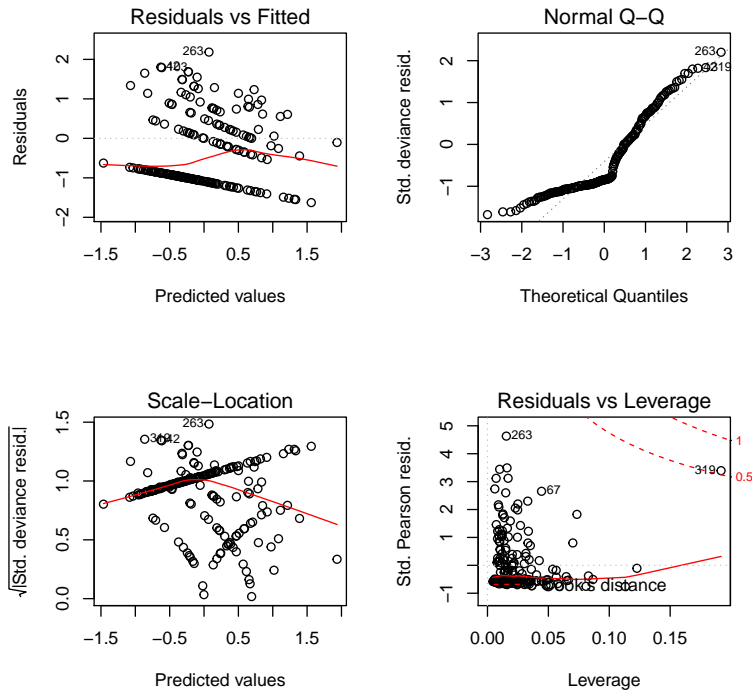
Histogram of humpback counts:



I looked at two family distribution fits using a GLM with all explanatory variables. First, quasi-poisson:



Then, negative binomial:



Negative binomial is a better fit.

4.2 Game plan

If whales are simply following their food, then prey should predict humpbacks with consistent power throughout the summer; i.e., the effect of prey on HWs distribution is fixed, and factors like time of year and geographic block should not matter. This would be a fixed effects model.

However, there's probably more going on. Humpback distribution could be the result of a foraging habit (simply a function of time of year). Or, there could be other environmental variables at work.

Build competing models and compare their predictive power. Four types of models would be built: Pure prey following, pure habit (descriptive variables like X,Y,jul,OD), a combo of the two (Foragin strategy), then a multivariate habitat use model (add sst, sss, and inlet, then re-select all possible explanatory variables from a balanced set of candidates, so that Akaike weights are possible).

The best model of each type will be determined by selection on just the candidate models **WITHIN** each type. So there is no confusion about protocols or comparing across different effects structures.

Then, prediction error for each best-fit model will be calculated with L-O-O Cross validation on the season dataset. Error will be calculated as Anscombe residuals.

```

> # Check balance
> var.balance <- function(fixedf,minchar=10){
+   combos <- substr(fixedf,20,nchar(fixedf))
+   splits <- strsplit(combos," + ",fixed=TRUE)
+   # Only count up variables not in interaction formulae
+   havecommas <- vector()
+   for(i in 1:length(splits)){
+     havecommasi <- which(nchar(splits[[i]]) > minchar)
+     if(length(havecommasi)>0){havecommas <- c(havecommas,i)}
+   }
+   vars2use <- 1:length(splits)
+   if(length(havecommas)!=0){vars2use <- vars2use[-havecommas]}
+   vars <- vector()
+   for(i in vars2use){
+     varsi <- as.character(splits[[i]])
+     vars <- c(vars,varsi)
+   }
+   vars <- gsub(" ", "",vars)
+   uvars <- unique(vars)
+   scores <- rep(0,times=length(uvars))
+   # Tally up occurrences of each option
+   for(i in 1:length(uvars)){
+     for(j in 1:length(vars)){
+       if(nchar(vars[j])<minchar & uvars[i]==vars[j]){scores[i] <- scores[i] + 1}
+     } }
+   return(cbind(uvars,scores) )
+ }

> gam2glm <- function(formulae){
+   # Prep GLM formulae
+   forms <- gsub(" ", "",formulae)
+   forms <- strsplit(forms,"+",fixed=TRUE)
+   glmforms <- vector()
+   for(j in 1:length(forms)){
+     formj <- forms[[j]]
+     newj <- formj[1]
+     termsj <- formj[2:length(formj)]
+     termsj <- substr(termsj,3,(nchar(termsj)-1))
+     termsj <- paste(termsj,collapse="+")
+     newj <- paste0(newj,"+",termsj)
+     glmforms <- c(glmforms,newj)
+   }
+   return(glmforms)
+ }

> gamloop <- function(formulae,newdata,name,progress=TRUE,weight.rank=FALSE,
+   cross.validate=FALSE){
+   library(mgcv)
+   library(MuMIn)
+   +
+   results <- data.frame()
+   for(i in 1:length(formulae)){
+     fi <- formula(formulae[i])
+     gami <- gam(formula= fi, data=newdata, family=nb,gamma=1.4)
+     #AIC <- round(gami$aic[1],3)

```

```

+ AIC <- round(AICc(gami),3)
+ DevExpl <- round(summary(gami)$dev.expl,3)
+ n <- summary(gami)$n
+ form <- as.character(fi)
+ form <- paste(form[2],form[1],form[3],collapse=" ")
+ Cat <- name
+ iresults <- data.frame(Cat,form,n,DevExpl,AIC,row.names=NULL)
+
+ # Cross Validation
+ if(cross.validate){
+   glmforms <- gam2glm(formulae)
+
+   # Leave-one-out cross-validation
+   preds <- thtas <- vector()
+   for(m in 1:nrow(newdata)){
+     test <- newdata[m,]
+     training <- newdata[-m,]
+     training.model <- gam(formula=fi, data=training, family=nb,gamma=1.4)
+
+     # Predict Left-Out-Value
+     if(!is.na(training.model[1])){
+       pred <- predict.gam(training.model,test)
+       ob <- test$hw
+     }else{
+       pred <- ob <- NA
+     }
+
+     # Gather THETA for Anscombes
+     glmfi <- formula(glmforms[i])
+     thta <- glm.nb(formula=glmfi,data=training)
+     thta <- thta$theta
+
+     # Store residuals etc
+     preds <- c(preds,pred)
+     thtas <- c(thtas,thta)
+   }
+
+   # Calculate prediction error
+   obs <- newdata$hw
+
+   # Simple Raw
+   Raw <- preds - obs
+   MSRaw <- mean((Raw)^2,na.rm=TRUE)
+
+   par(mfrow=c(1,1)) ; par(mar=c(5,5,1,1)) ; plot(preds~obs,xlim=c(0,10),ylim=c(0,10))
+   # Anscombe's residuals
+   AR1 <- AR2 <- vector()
+   for(m in 1:length(obs)){
+     AR1i <- anscombe(thta=thtas[m],obs=obs[m],pred=preds[m],method=1)
+     AR2i <- anscombe(thta=thtas[m],obs=obs[m],pred=preds[m],method=2)
+     AR1 <- c(AR1,AR1i)
+     AR2 <- c(AR2,AR2i)
+   }
+   par(mfrow=c(3,1)) ; par(mar=c(5,5,3,3)) ; hist(Raw) ; hist(AR1) ; hist(AR2)

```

```

+
+   # Calculate measn sq error
+   MSAR1 <- MSAR(AR1)
+   MSAR2 <- MSAR(AR2)
+
+   # Mean HW in leg
+   MuHW <- mean(obs,na.rm=TRUE)
+   # Max HW in leg
+   MxHW <- max(obs,na.rm=TRUE)
+
+   # Store results
+   iresults$MSRaw <- MSRaw
+   iresults$MSAR1 <- MSAR1
+   iresults$MSAR2 <- MSAR2
+   iresults$MuHW <- MuHW
+   iresults$MxHW <- MxHW
+ }
+
+
+   if(progress){print(iresults)}
+   results <- rbind(results,iresults)
+ }
+
+
+ weight <- Weights(as.numeric(results$AIC))
+ results$weight <- round(weight,5)
+ if(weight.rank){results <- results[rev(order(results$weight)),]}
+
+
+ return(results)
+ }

> # Only use on balanced model sets!
> RVI <- function(subprey,maxscale=FALSE){
+   fixedf <- as.character(subprey$form)
+   combos <- substr(fixedf,20,nchar(fixedf))
+   splits <- strsplit(combos," + ",fixed=TRUE)
+   #####
+   vars <- vector()
+   for(i in 1:length(splits)){
+     varsi <- as.character(splits[[i]])
+     vars <- c(vars,varsi)
+   }
+   vars <- gsub(" ","",vars)
+   uvars <- unique(vars)
+   #####
+   totweight <- rep(0,times=length(uvars))
+   # Total up weights for each variable
+   for(i in 1:length(uvars)){
+     for(j in 1:length(splits)){
+       splitj <- as.character(splits[[j]])
+       weightj <- subprey$weight[j]
+       if(any(uvars[i]==splitj)){totweight[i] <- totweight[i] + weightj}
+     }
+   }
+   VAR <- substr(uvars,3,6)

```

```

+ RVI <- as.numeric(totweight)
+ result <- data.frame(VAR,RVI)
+ result <- result[rev(order(result$RVI)),]
+ if(maxscale){
+   maxRVI <- max(result$RVI)
+   result$RVI <- result$RVI / maxRVI
+ }
+ result$RVI <- round(result$RVI,digits=2)
+ return(result)
+ }

```

4.3 Pure Prey Following

Build as many possible combinations of backscatter-only models.

4.3.1 Relative Variable Importance using Akaike Weights

Of the 95 models fit, 51 did not include interaction terms. This subset of models was balanced (in total, they contained an equal number of all candidate variables; 27 occurrences of each). Because this model set is balanced, I can use it to assess the relative importance of candidate variables (Burnham and Anderson 2002, Zuur et al. Mixed Effects) using Akaike weights.

```

> # Recalculate Akaike weights with this subset
> subprey <- Prey[1:51,]
> subprey$weight <- Weights(as.numeric(subprey$AIC))
> RVI(subprey,maxscale=TRUE)

```

4.3.2 95% confidence set using Akaike Weights

Now find a 95% confidence set for all 95 candidate prey models using Akaike Weights. This involves interactions and the variables are no longer balanced. RVI cannot be directly measured.

The more models in the 95% confidence set, the more model uncertainty. Confirmed if all the AIC values are similar.

Calculate evidence ratios (weight1 / weight 2)

Zuur et al. p. 484

For my information:

According to Hilbe (p. 70), AIC values must be at least 2.5 AIC units apart in order to be indicative of better fit. For small sample sizes, this different must increase. For instance, when $60 < n < 256$, AIC values must be at least 6 units apart to be compared.

Now rank and subset results to a 95% confidence set of models:

```

> Prey <- Prey[rev(order(Prey$weight)),]
> Prey$cumulativeAIC <- cumsum(Prey$weight)
> Prey95 <- Prey[Prey$cumulativeAIC <= .95,]
> Prey95$form <- substr(as.character(Prey95$form),20,nchar(as.character(Prey95$form)))
> print(Prey95)

```

4.4 Pure Habit

I include models that treat time of year as a continuous explanatory variable (Julian day), as well as models that treat time of year as a fixed-effect factor (Legs, levels 1-5). Likewise, I try treating geographic

block as a continuous variable (calculated as swimming distance, variable "OD", from the inlandmost point in the study area) or as a fixed-effect factor (geographic block, levels 1-8). I also try using GPS coordinates.

If time and location are significant effects, they could either reflect a foraging "habit" (whales are adhering to a behavioral heuristic, not responding to actual environmental changes), or they could be serving as proxies for some other environmental process.

```
> nhabit <- c("s(ODs)", "s(juls)", "s(ODs) + s(juls)", "s(ODs,juls)",
+           "s(Xs) + s(Ys)", "s(Xs,Ys)", "s(Ys,juls)", "s(Xs,Ys) + s(juls)")
> fhabit <- c("leg", "block", "leg + block")
> combhabit <- c("leg + s(ODs)", "leg + s(Xs) + s(Ys)", "leg + s(Xs,Ys)", "leg + s(Ys)", "block + s(juls)")
> habit <- c(nhabit, fhabit, combhabit)
> fixedf <- vector()
> for(i in 1:length(habit)){fi <- paste0("hw ~ offset(off) + ", habit[i]) ; fixedf <- c(fixedf, fi) }
> Habit <- gamloop(fixedf, newdata, name="Habit", weight.rank=TRUE)

> Habit$cumulativeAIC <- cumsum(Habit$weight)
> Habit95 <- Habit[1:3,]
> Habit95$form <- substr(as.character(Habit95$form), 20, nchar(as.character(Habit95$form)))
> print(Habit95)

> summary(gam(formula= hw ~ offset(off) + s(Xs,Ys) + s(juls), data=newdata, family=nb, gamma=1.4))
> summary(gam(formula= hw ~ offset(off) + block + s(juls), data=newdata, family=nb, gamma=1.4))
> summary(gam(formula= hw ~ offset(off) + s(ODs,juls), data=newdata, family=nb, gamma=1.4))
```

4.5 Environmental model

```
> subenv <- Env[1:15,]
> subenv$weight <- Weights(as.numeric(subenv$AIC))
> RVI(subenv, maxscale=TRUE)

> Env <- Env[rev(order(Env$weight)),]
> Env$cumulativeAIC <- cumsum(Env$weight)
> Env95 <- Env[Env$cumulativeAIC <= 1,]
> Env95$form <- substr(as.character(Env95$form), 20, nchar(as.character(Env95$form)))
> print(Env95)

> summary(gam(formula= hw ~ offset(off) + s(sss,sst), data=newdata, family=nb, gamma=1.4))
```

4.6 Foraging strategy

```
> # Move forward with top 3 prey models
> topprey <- Prey95$form[1:5]
> # And top 4 environmental models:
> topenv <- Env95$form[1:4]
> newcombos <- c("s(sss,I.hi)",
+               "s(sss,I.hi) + s(chla)",
+               "s(sss,I.hi) + s(inlet)",
+               "s(chla,I.hi)",
+               "s(chla,I.hi) + s(sst,sss)",
+               "s(chla,I.hi) + s(inlet)",
+               "s(inlet,I.hi)",
+               "s(inlet,I.hi) + s(sss,sst)",
+               "s(inlet,I.hi) + s(sss,sst) + s(chla)",
+               "s(inlet,I.hi) + s(chla)")
```

```

> fixedf <- vector()
> for(i in 1:length(topenv)){
+   for(j in 1:length(topprey)){
+     fixedf <- c(fixedf,paste(topenv[i],topprey[j],sep=" + "))
+   }
+ }
> fixedf <- c(fixedf,newcombos)
> fixedf <- paste0("hw ~ offset(off) + ",fixedf)
> fordff <- fixedf
> ForStrat <- gamloop(fordff,newdata,name="ForStrat",weight.rank=FALSE)

```

4.6.1 95% confidence set using Akaike Weights

```

> ForStrat <- ForStrat[order(ForStrat$AIC),]
> ForStrat$cumulativeAIC <- cumsum(ForStrat$weight)
> ForStrat95 <- ForStrat[ForStrat$cumulativeAIC <= 1,]
> ForStrat95$form <- substr(as.character(ForStrat95$form),20,nchar(as.character(ForStrat95$form)))
> print(ForStrat95)

```

4.7 Full habitat use

```

> # Best Habit models
> #habits <- c("s(Xs, Ys) + s(juls)")
> # Combine with foraging strategy models
> #fixedf <- c(paste0(fixedf," + ",habits))
> # Then add combination so that vector:
> fixedf <- c("hw ~ offset(off) + s(sss, sst) + s(inlet) + s(I.hi, I.lo) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss, sst) + s(inlet) + s(I.hi, D.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss, sst) + s(inlet) + s(I.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss, sst) + s(I.hi, I.lo) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss, sst) + s(I.hi, D.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss, sst) + s(I.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss, sst) + s(chla) + s(I.hi, I.lo) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss, sst) + s(chla) + s(I.hi, D.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss, sst) + s(chla) + s(I.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(chla, inlet) + s(sss, sst) + s(I.hi, I.lo) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(chla, inlet) + s(sss, sst) + s(I.hi, D.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(chla, inlet) + s(sss, sst) + s(I.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss,I.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss,I.hi) + s(chla) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss,I.hi) + s(inlet) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(chla,I.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(chla,I.hi) + s(sst,sss) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(chla,I.hi) + s(inlet) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet,I.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet,I.hi) + s(sss,sss) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet,I.hi) + s(sss,sss) + s(chla) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet,I.hi) + s(chla) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet) + s(I.hi, I.lo) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet) + s(I.hi, D.hi) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet) + s(I.hi) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(I.hi, I.lo) + s(Xs, sss) + s(juls)" ,

```

```

+ "hw ~ offset(off) + s(I.hi, D.hi) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(I.hi) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(chla) + s(I.hi, I.lo) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(chla) + s(I.hi, D.hi) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(chla) + s(I.hi) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(chla, inlet) + s(I.hi, I.lo) + s(Xs, sss) + s(juls)",
+ "hw ~ offset(off) + s(chla, inlet) + s(I.hi, D.hi) + s(Xs, sss) + s(juls)",
+ "hw ~ offset(off) + s(chla, inlet) + s(I.hi) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(sss,I.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss,I.hi) + s(chla) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(sss,I.hi) + s(inlet) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(chla,I.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(chla,I.hi) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(chla,I.hi) + s(sst,sss) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(chla,I.hi) + s(inlet) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(chla,I.hi) + s(inlet) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet,I.hi) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet,I.hi) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet,I.hi) + s(sss,sst) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet,I.hi) + s(sss,sst) + s(chla) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet,I.hi) + s(chla) + s(Xs, Ys) + s(juls)" ,
+ "hw ~ offset(off) + s(inlet,I.hi) + s(chla) + s(Xs, sss) + s(juls)" ,
+ "hw ~ offset(off) + s(sss,juls)",
+ "hw ~ offset(off) + s(sss,juls) + s(I.hi)",
+ "hw ~ offset(off) + s(sss,juls) + s(I.hi,D.hi)",
+ "hw ~ offset(off) + s(sss,juls) + s(I.hi,I.lo)",
+ "hw ~ offset(off) + s(sss,juls) + s(I.hi,inlet)",
+ "hw ~ offset(off) + s(sss,juls) + s(I.hi,chla)",
+ "hw ~ offset(off) + s(I.hi,juls)",
+ "hw ~ offset(off) + s(I.hi,juls) + s(Xs,Ys)",
+ "hw ~ offset(off) + s(I.hi,juls) + s(chla)",
+ "hw ~ offset(off) + s(I.hi,juls) + s(inlet)",
+ "hw ~ offset(off) + s(I.hi,juls) + s(sss,sst)",
+ "hw ~ offset(off) + s(I.hi,sss) + s(juls)",
+ "hw ~ offset(off) + s(I.hi,sss) + s(chla)",
+ "hw ~ offset(off) + s(I.hi,sss) + s(inlet)",
+ "hw ~ offset(off) + s(I.hi,sss) + s(juls) + s(Xs,Ys)",
+ "hw ~ offset(off) + s(I.hi,sss) + s(chla) + s(Xs,Ys)",
+ "hw ~ offset(off) + s(I.hi,sss) + s(inlet) + s(Xs,Ys)"
> habusedf <- unique(fixedf)
> HabUse <- gamloop(habusedf,newdata,name="HabUse",weight.rank=FALSE)
>
> # Check these for insignificant terms, and update model vector
> #forms <- as.character(HabUse$form)
> #i <- 65
> #summary(gam(formula= formula(forms[i]), data=newdata, family=nb,gamma=1.4))
> HabUse <- HabUse[order(HabUse$AIC),]
> HabUse$cumulativeAIC <- cumsum(HabUse$weight)
> HabUse95 <- HabUse[HabUse$cumulativeAIC <= .985,]
> HabUse95$form <- substr(as.character(HabUse95$form),20,nchar(as.character(HabUse95$form)))
> print(HabUse95)

```

5 Best-fit models

```
> # Best
> bestform <- formula(hw ~ offset(off) + s(chla, I.hi) + s(sst,sss) + s(Xs,Ys) + s(juls) )
> bestgam <- gam(formula= bestform, data=newdata, family=nb,gamma=1.4)
> summary(bestgam)
```

```
Family: Negative Binomial(3.926)
Link function: log
```

Formula:

```
hw ~ offset(off) + s(chla, I.hi) + s(sst, sss) + s(Xs, Ys) +
      s(juls)
```

Parametric coefficients:

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.5736      0.1167  -22.05   <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(chla,I.hi)   6.100   8.277  18.47 0.021071 *
s(sst,sss)    14.825  19.085  42.30 0.001686 **
s(Xs,Ys)       12.538  16.578  36.20 0.003613 **
s(juls)         4.901   5.650  26.04 0.000181 ***
---
```

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

```
R-sq.(adj) = 0.601   Deviance explained = 62.4%
-REML = 280.05   Scale est. = 1           n = 216
```

```
> # Best with no I.hi interactions (2nd place, equiv AIC)
> form2 <- formula(hw ~ offset(off) + s(sss, sst) + s(chla) + s(I.hi) + s(Xs, Ys) + s(juls))
> bestgam2 <- gam(formula= form2, data=newdata, family=nb,gamma=1.4)
> summary(bestgam2)
```

```
Family: Negative Binomial(3.507)
Link function: log
```

Formula:

```
hw ~ offset(off) + s(sss, sst) + s(chla) + s(I.hi) + s(Xs, Ys) +
      s(juls)
```

Parametric coefficients:

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.5629      0.1168  -21.95   <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(sss,sst)  14.896  19.235  41.998 0.001990 **
s(chla)       1.088   1.156   5.099 0.030704 *
s(I.hi)        2.750   3.373  11.297 0.014798 *
```

```

s(Xs,Ys)    12.034 16.029 34.420 0.004836 **
s(juls)      4.854  5.624 26.863 0.000125 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.582   Deviance explained = 60.6%
-REML =    280.4   Scale est. = 1           n = 216

```

5.1 Model validation

Model validate the best-fit Habitat Use gam:

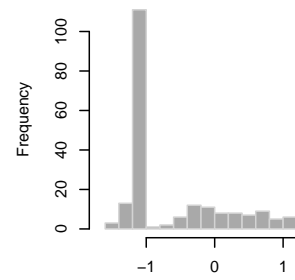
```

Method: REML   Optimizer: outer newton
full convergence after 6 iterations.
Gradient range [-9.982715e-06,2.827818e-06]
(score 280.0461 & scale 1).
Hessian positive definite, eigenvalue range [0.6010659,2.987353].
Model rank =  97 / 97

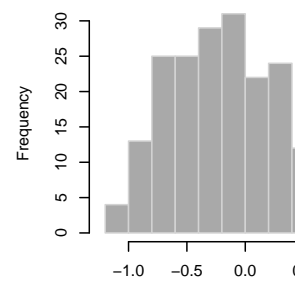
```

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'.

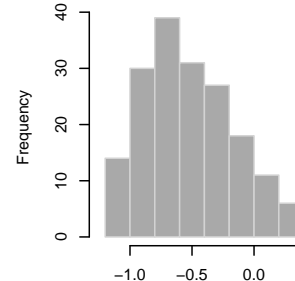
	k'	edf	k-index	p-value
s(chla,I.hi)	29.000	6.100	0.947	0.90
s(sst,sss)	29.000	14.825	0.921	0.75
s(Xs,Ys)	29.000	12.538	0.934	0.80
s(juls)	9.000	4.901	0.829	0.16



GAM

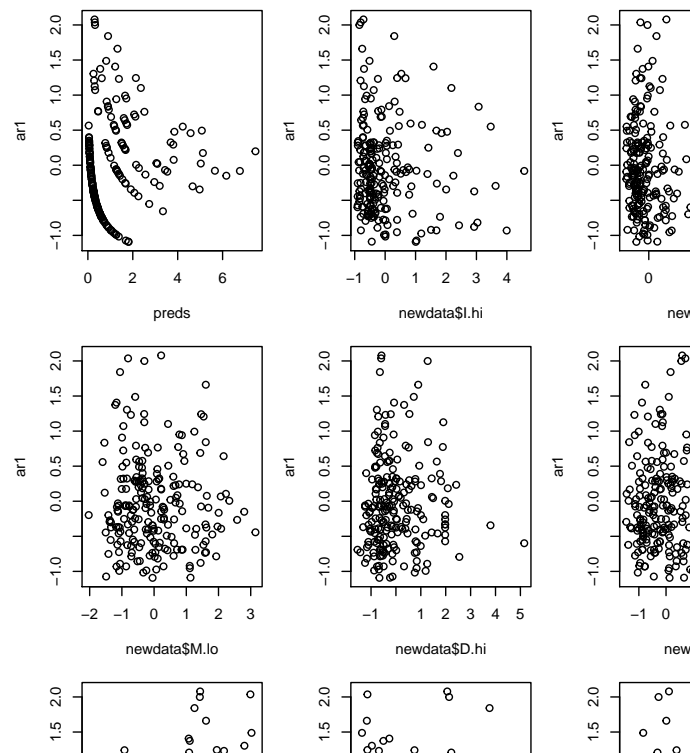


Choi et al.

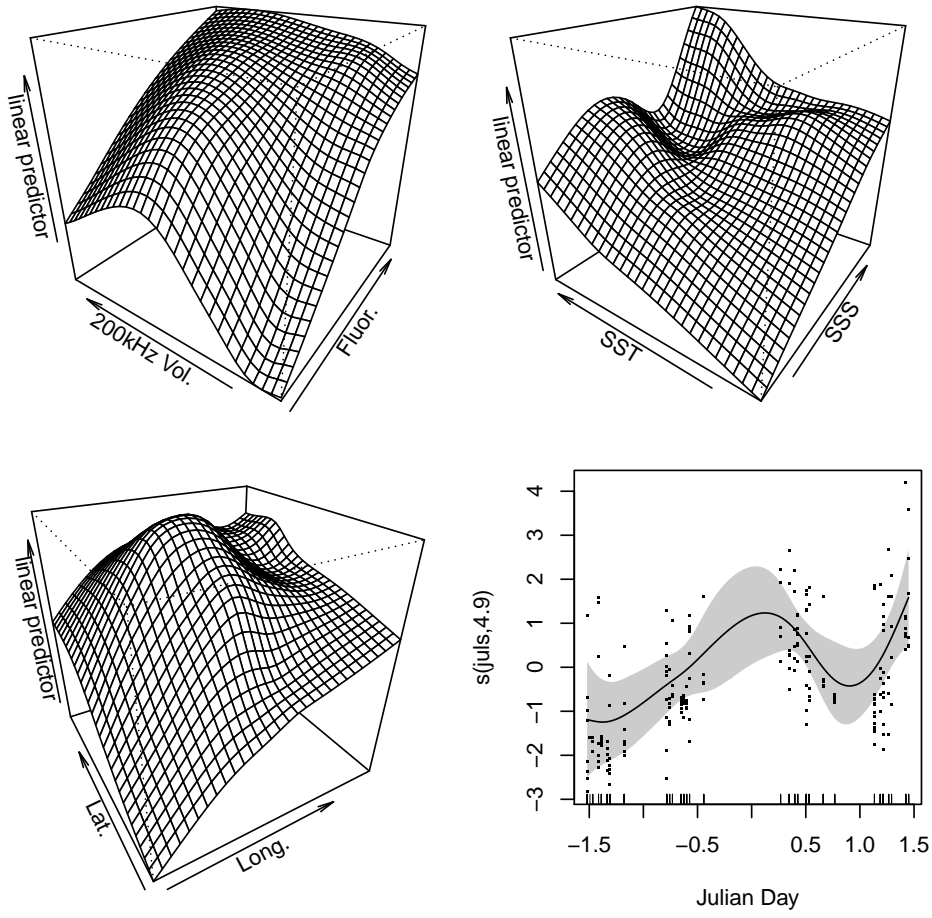


McCullagh

Should I use the raw residuals, or calculate Anscombe residuals? What difference would it make?



6 Smoothing functions figure



7 Leg by Leg Comparison

7.1 Setup variable sets

```
> lprey <- c("s(I.hi)",
+ "s(I.lo)",
+ "s(D.hi)",
+ "s(M.lo)",
+ "s(I.hi) + s(I.lo)",
+ "s(I.hi) + s(D.hi)",
+ "s(I.hi) + s(M.lo)",
+ "s(I.lo) + s(D.hi)",
+ "s(I.lo) + s(M.lo)",
+ "s(D.hi) + s(M.lo)",
+ "s(I.hi) + s(I.lo) + s(D.hi)",
+ "s(I.hi) + s(I.lo) + s(M.lo)",
+ "s(I.lo) + s(D.hi) + s(M.lo)",
+ "s(D.hi) + s(M.lo) + s(I.hi)",
+ "s(I.hi) + s(I.lo) + s(D.hi) + s(M.lo)")
> lprey <- paste0("hw ~ offset(off) + ",lprey)
> var.balance(lprey)
> lprox <- c("s(sst)",
+ "s(sss)",
+ "s(chla)",
+ "s(inlet)",
+ "s(sst) + s(sss)",
+ "s(sst) + s(chla)",
+ "s(sst) + s(inlet)",
+ "s(sss) + s(chla)",
+ "s(sss) + s(inlet)",
+ "s(chla) + s(inlet)",
+ "s(sst) + s(sss) + s(chla)",
+ "s(sst) + s(sss) + s(inlet)",
+ "s(sss) + s(chla) + s(inlet)",
+ "s(chla) + s(inlet) + s(sst)",
+ "s(sst) + s(sss) + s(chla) + s(inlet)")
> lprox <- paste0("hw ~ offset(off) + ",lprox)
> var.balance(lprox)
> # Sampling variables
> lsamp <- c(lprey,lprox)
> # Habit Variables
> lhabit <- c("s(Xs)", "s(Ys)", "s(ODs)", "s(Xs) + s(Ys)")
> # Habitat Variables
> lhabitat <- vector()
> for(i in 1:length(lhabit)){
+   for(j in 1:length(lsamp)){
+     lhabitat <- c(lhabitat,paste0(lsamp[j]," + ",lhabit[i]))
+   }
+ }
> # Finish Habit Variables
> lhabit <- paste0("hw ~ offset(off) + ",lhabit)
```

7.2 Setup monthly datasets

```
> # Remove outliers
> legdata <- data[data$hw <= 10      &
+               data$I.hi <= 1500 &
+               data$I.lo <= 1500 ,]
> # Remove collinear predictors
> legdata$C.hi <- legdata$C.lo <- legdata$secchi <- NULL
> # Offset
> legdata$off <- log(legdata$eff)
> # Standardize explanatory variables
> legdata$juls <- legdata$jul
> legdata$Xs <- legdata$X
> legdata$Ys <- legdata$Y
> legdata$ODs <- legdata$OD
> cols <- c(9:18,20:23)
> for(i in cols){
+   legdata[legdata$leg==1,i] <- standardize(legdata[legdata$leg==1,i]) # leg-by-leg standardization
+   legdata[legdata$leg==2,i] <- standardize(legdata[legdata$leg==2,i])
+   legdata[legdata$leg==3,i] <- standardize(legdata[legdata$leg==3,i])
+   legdata[legdata$leg==4,i] <- standardize(legdata[legdata$leg==4,i])
+   legdata[legdata$leg==5,i] <- standardize(legdata[legdata$leg==5,i])
+ }
> # Subset to leg
> leg1 <- legdata[legdata$leg==1,]
> leg2 <- legdata[legdata$leg==2,]
> leg3 <- legdata[legdata$leg==4,]
> leg4 <- legdata[legdata$leg==5,]
>
```

7.3 Model fitting

Fit Prey models:

```
> Prey1 <- gamloop(formulae=lprey,name="Prey",newdata=leg1,cross.validate=TRUE)
> Prey2 <- gamloop(formulae=lprey,name="Prey",newdata=leg2,cross.validate=TRUE)
> Prey3 <- gamloop(formulae=lprey,name="Prey",newdata=leg3,cross.validate=TRUE)
> Prey4 <- gamloop(formulae=lprey,name="Prey",newdata=leg4,cross.validate=TRUE)
> Prey1$leg <- rep(1,times=nrow(Prey1))
> Prey2$leg <- rep(2,times=nrow(Prey2))
> Prey3$leg <- rep(3,times=nrow(Prey3))
> Prey4$leg <- rep(4,times=nrow(Prey4))
```

Fit Proxy Models:

```
> Prox1 <- gamloop(formulae=lprox,name="Prox",newdata=leg1,cross.validate=TRUE)
> Prox2 <- gamloop(formulae=lprox,name="Prox",newdata=leg2,cross.validate=TRUE)
> Prox3 <- gamloop(formulae=lprox,name="Prox",newdata=leg3,cross.validate=TRUE)
> Prox4 <- gamloop(formulae=lprox,name="Prox",newdata=leg4,cross.validate=TRUE)
> Prox1$leg <- rep(1,times=nrow(Prox1))
> Prox2$leg <- rep(2,times=nrow(Prox2))
> Prox3$leg <- rep(3,times=nrow(Prox3))
> Prox4$leg <- rep(4,times=nrow(Prox4))
```

Fit Prey+Proxy Models:

```

> Samp1 <- gamloop(formulae=lsamp,name="Samp",newdata=leg1,cross.validate=TRUE)
> Samp2 <- gamloop(formulae=lsamp,name="Samp",newdata=leg2,cross.validate=TRUE)
> Samp3 <- gamloop(formulae=lsamp,name="Samp",newdata=leg3,cross.validate=TRUE)
> Samp4 <- gamloop(formulae=lsamp,name="Samp",newdata=leg4,cross.validate=TRUE)
> Samp1$leg <- rep(1,times=nrow(Samp1))
> Samp2$leg <- rep(2,times=nrow(Samp2))
> Samp3$leg <- rep(3,times=nrow(Samp3))
> Samp4$leg <- rep(4,times=nrow(Samp4))

```

Fit Habit models:

```

> Habit1 <- gamloop(formulae=lhabit,name="Habit",newdata=leg1,cross.validate=TRUE)
> Habit2 <- gamloop(formulae=lhabit,name="Habit",newdata=leg2,cross.validate=TRUE)
> Habit3 <- gamloop(formulae=lhabit,name="Habit",newdata=leg3,cross.validate=TRUE)
> Habit4 <- gamloop(formulae=lhabit,name="Habit",newdata=leg4,cross.validate=TRUE)
> Habit1$leg <- rep(1,times=nrow(Habit1))
> Habit2$leg <- rep(2,times=nrow(Habit2))
> Habit3$leg <- rep(3,times=nrow(Habit3))
> Habit4$leg <- rep(4,times=nrow(Habit4))

```

Fit full Prey+Proxy+Habit:

```

> lhabitatred <- lhabitat[-which(nchar(lhabitat)>55)]
> Habitat1 <- gamloop(formulae=lhabitatred,name="Habitat",newdata=leg1,cross.validate=TRUE)
> Habitat2 <- gamloop(formulae=lhabitat,name="Habitat",newdata=leg2,cross.validate=TRUE)
> Habitat3 <- gamloop(formulae=lhabitat,name="Habitat",newdata=leg3,cross.validate=TRUE)
> Habitat4 <- gamloop(formulae=lhabitatred,name="Habitat",newdata=leg4,cross.validate=TRUE)
> Habitat1$leg <- rep(1,times=nrow(Habitat1))
> Habitat2$leg <- rep(2,times=nrow(Habitat2))
> Habitat3$leg <- rep(3,times=nrow(Habitat3))
> Habitat4$leg <- rep(4,times=nrow(Habitat4))

```

7.4 Relative Variable Importance

```

> RVI(Prey1,maxscale=TRUE)

```

```

VAR  RVI
1 I.hi 1.00
4 M.lo 0.92
3 D.hi 0.55
2 I.lo 0.48

```

```

> RVI(Prey2,maxscale=TRUE)

```

```

VAR  RVI
1 I.hi 1.00
3 D.hi 0.42
2 I.lo 0.30
4 M.lo 0.23

```

```

> RVI(Prey3,maxscale=TRUE)

```

```

VAR  RVI
3 D.hi 1.00
2 I.lo 0.73
4 M.lo 0.52
1 I.hi 0.47

```

```

> RVI(Prey4,maxscale=TRUE)

  VAR  RVI
3 D.hi 1.00
1 I.hi 0.89
4 M.lo 0.87
2 I.lo 0.60

> RVI(Prox1,maxscale=TRUE)

  VAR  RVI
3 chla 1.00
2 sss) 0.31
1 sst) 0.26
4 inle 0.24

> RVI(Prox2,maxscale=TRUE)

  VAR  RVI
4 inle 1.00
1 sst) 0.27
3 chla 0.12
2 sss) 0.08

> RVI(Prox3,maxscale=TRUE)

  VAR  RVI
2 sss) 1.00
4 inle 0.85
3 chla 0.22
1 sst) 0.19

> RVI(Prox4,maxscale=TRUE)

  VAR  RVI
3 chla 1.00
1 sst) 0.63
4 inle 0.55
2 sss) 0.53

> RVI(Samp1,maxscale=TRUE)

  VAR  RVI
7 chla 1.00
6 sss) 0.31
5 sst) 0.26
8 inle 0.24
1 I.hi 0.03
4 M.lo 0.03
3 D.hi 0.02
2 I.lo 0.01

> RVI(Samp2,maxscale=TRUE)

  VAR  RVI
1 I.hi 1.00
8 inle 0.46
3 D.hi 0.42
2 I.lo 0.30
4 M.lo 0.23

```

```

5 sst) 0.12
7 chla 0.05
6 sss) 0.04

> RVI(Samp3,maxscale=TRUE)

VAR RVI
6 sss) 1.00
8 inle 0.85
7 chla 0.22
5 sst) 0.19
4 M.lo 0.00
3 D.hi 0.00
2 I.lo 0.00
1 I.hi 0.00

> RVI(Samp4,maxscale=TRUE)

VAR RVI
7 chla 1.00
5 sst) 0.63
8 inle 0.55
6 sss) 0.53
3 D.hi 0.48
1 I.hi 0.43
4 M.lo 0.42
2 I.lo 0.29

```

7.5 Prediction Error

Prepare residuals datasets:

```

> min.resid <- function(results){
+   MSRaw <- results[which.min(results$MSRaw),]
+   MSRaw$MSAR1 <- MSRaw$MSAR2 <- NA
+
+   MSAR1 <- results[which.min(results$MSAR1),]
+   MSAR1$MSRaw <- MSAR1$MSAR2 <- NA
+
+   MSAR2 <- results[which.min(results$MSAR2),]
+   MSAR2$MSRaw <- MSAR2$MSAR1 <- NA
+
+   summresults <- rbind(MSRaw,MSAR1,MSAR2)
+   return(summresults)
+ }
> PreyR <- rbind(min.resid(Prey1),min.resid(Prey2),min.resid(Prey3),min.resid(Prey4))
> ProxR <- rbind(min.resid(Prox1),min.resid(Prox2),min.resid(Prox3),min.resid(Prox4))
> SampR <- rbind(min.resid(Samp1),min.resid(Samp2),min.resid(Samp3),min.resid(Samp4))
> HabitR <- rbind(min.resid(Habit1),min.resid(Habit2),min.resid(Habit3),min.resid(Habit4))
> HabitatR <- rbind(min.resid(Habitat1),min.resid(Habitat2),min.resid(Habitat3),min.resid(Habitat4))
>

```

Scale prediction error:

```

> PreyR$SMSAR1 <- PreyR$MSAR1/PreyR$MuHW
> PreyR$SMSAR2 <- PreyR$MSAR2/PreyR$MuHW
> ProxR$SMSAR1 <- ProxR$MSAR1/ProxR$MuHW

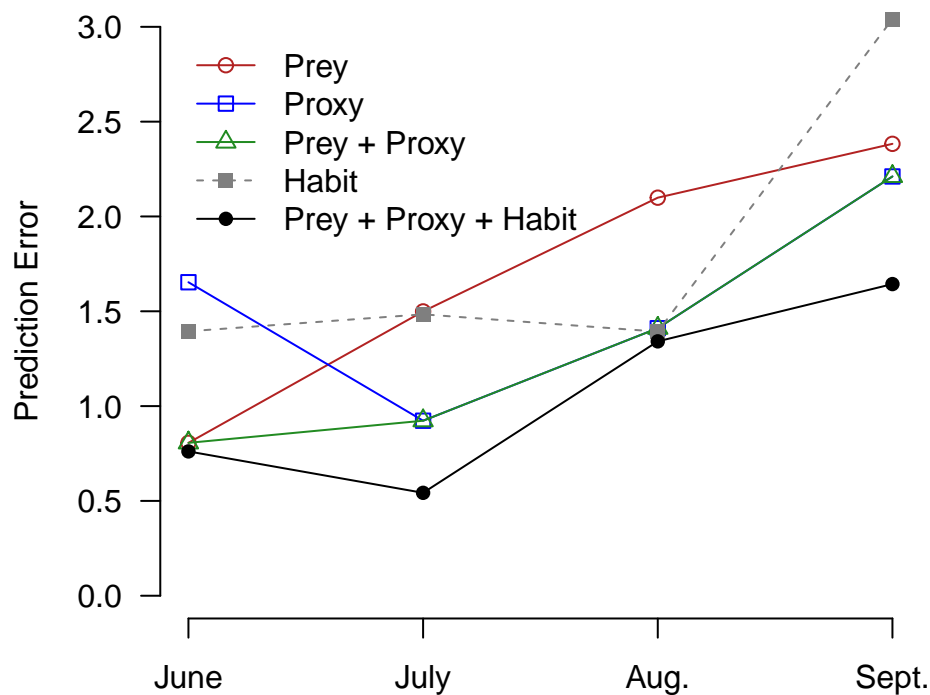
```

```

> ProxR$SMSAR2 <- ProxR$MSAR2/ProxR$MuHW
> SampR$SMSAR1 <- SampR$MSAR1/SampR$MuHW
> SampR$SMSAR2 <- SampR$MSAR2/SampR$MuHW
> HabitR$SMSAR1 <- HabitR$MSAR1/HabitR$MuHW
> HabitR$SMSAR2 <- HabitR$MSAR2/HabitR$MuHW
> HabitatR$SMSAR1 <- HabitatR$MSAR1/HabitatR$MuHW
> HabitatR$SMSAR2 <- HabitatR$MSAR2/HabitatR$MuHW

```

7.5.1 Prediction Error figure



7.5.2 Best-fit models

	Cat	form	n	DevExpl
13	Prey	hw ~ offset(off) + s(I.lo) + s(D.hi) + s(M.lo)	47	0.161
132	Prey	hw ~ offset(off) + s(I.lo) + s(D.hi) + s(M.lo)	57	0.380
8	Prey	hw ~ offset(off) + s(I.lo) + s(D.hi)	59	0.187
15	Prey	hw ~ offset(off) + s(I.hi) + s(I.lo) + s(D.hi) + s(M.lo)	56	0.167
10	Prox	hw ~ offset(off) + s(chla) + s(inlet)	47	0.393
131	Prox	hw ~ offset(off) + s(sss) + s(chla) + s(inlet)	57	0.559
9	Prox	hw ~ offset(off) + s(sss) + s(inlet)	59	0.680
14	Prox	hw ~ offset(off) + s(chla) + s(inlet) + s(sst)	56	0.161
133	Samp	hw ~ offset(off) + s(I.lo) + s(D.hi) + s(M.lo)	47	0.161
28	Samp	hw ~ offset(off) + s(sss) + s(chla) + s(inlet)	57	0.559
24	Samp	hw ~ offset(off) + s(sss) + s(inlet)	59	0.680
29	Samp	hw ~ offset(off) + s(chla) + s(inlet) + s(sst)	56	0.161
3	Habit	hw ~ offset(off) + s(ODs)	47	0.410
2	Habit	hw ~ offset(off) + s(Xs)	57	0.365

	Habit		hw ~ offset(off) + s(Xs) + s(Ys)	59	0.694		
21	Habit		hw ~ offset(off) + s(Ys)	56	0.119		
78	Habitat	hw ~ offset(off) + s(sss) + s(chla) + s(ODs)	47	0.515			
47	Habitat	hw ~ offset(off) + s(sss) + s(Ys)	57	0.478			
107	Habitat	hw ~ offset(off) + s(sss) + s(Xs) + s(Ys)	59	0.688			
48	Habitat	hw ~ offset(off) + s(sst) + s(chla) + s(Ys)	56	0.184			
	AIC	MSAR1	MuHW	weight	leg	SMSAR1	SMSAR2
13	121.724	0.703	0.87	0.01273	1	0.806	NA
132	149.134	1.578	1.05	0.00028	2	1.499	NA
8	194.148	3.131	1.49	0.23683	3	2.099	NA
15	169.936	2.468	1.04	0.00269	4	2.383	NA
10	111.916	1.442	0.87	0.08552	1	1.653	NA
131	142.533	0.971	1.05	0.01629	2	0.923	NA
9	160.436	2.104	1.49	0.55206	3	1.411	NA
14	165.000	2.290	1.04	0.01979	4	2.211	NA
133	121.724	0.703	0.87	0.00060	1	0.806	NA
28	142.533	0.971	1.05	0.00514	2	0.923	NA
24	160.436	2.104	1.49	0.55206	3	1.411	NA
29	165.000	2.290	1.04	0.01219	4	2.211	NA
3	112.577	1.217	0.87	0.06490	1	1.395	NA
2	138.940	1.562	1.05	0.02573	2	1.484	NA
4	162.469	2.077	1.49	0.93149	3	1.393	NA
21	158.312	3.150	1.04	0.31667	4	3.041	NA
78	110.180	0.664	0.87	0.01080	1	0.761	NA
47	145.064	0.571	1.05	0.00003	2	0.543	NA
107	159.545	2.001	1.49	0.03142	3	1.341	NA
48	165.760	1.702	1.04	0.00219	4	1.643	NA

Explore details of best fits for each month-model:

```
> # Prey
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Prey" & bestleg$leg==1])), data=leg1, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Prey" & bestleg$leg==2])), data=leg2, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Prey" & bestleg$leg==3])), data=leg3, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Prey" & bestleg$leg==4])), data=leg4, gamma=1))
> # Proxy
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Prox" & bestleg$leg==1])), data=leg1, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Prox" & bestleg$leg==2])), data=leg2, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Prox" & bestleg$leg==3])), data=leg3, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Prox" & bestleg$leg==4])), data=leg4, gamma=1))
> # Sampling
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Samp" & bestleg$leg==1])), data=leg1, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Samp" & bestleg$leg==2])), data=leg2, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Samp" & bestleg$leg==3])), data=leg3, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Samp" & bestleg$leg==4])), data=leg4, gamma=1))
> # Habit
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Habit" & bestleg$leg==1])), data=leg1, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Habit" & bestleg$leg==2])), data=leg2, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Habit" & bestleg$leg==3])), data=leg3, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Habit" & bestleg$leg==4])), data=leg4, gamma=1))
> # Habitat
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Habitat" & bestleg$leg==1])), data=leg1, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Habitat" & bestleg$leg==2])), data=leg2, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Habitat" & bestleg$leg==3])), data=leg3, gamma=1))
> summary(gam(formula(as.character(bestleg$form[bestleg$Cat=="Habitat" & bestleg$leg==4])), data=leg4, gamma=1))
```