

Grid trial multinomial fits to the whiting data

For discussion only

January 18, 2016

1 Data

```
library(gdata)

## Whiting length data
grid.whg.dat <-
  read.xls(
    "../data/Do_Not_Overwrite_OUR_LASSII_Grid_Trial_FU15_September_2015.xlsx",
    sheet = "Lengths",
    stringsAsFactors = FALSE)

grid.whg.dat <- subset(grid.whg.dat, SPECIES == "Whiting")

## remove any unused factor levels (housekeeping step)
grid.whg.dat <- droplevels(grid.whg.dat)

## change haul name
names(grid.whg.dat)[names(grid.whg.dat) == "HAUL."] <- "HAUL"

## Remove haul number 4
grid.whg.dat <- subset(grid.whg.dat, HAUL != 4)

## Make the "HAUL" variable character
grid.whg.dat$HAUL <- paste("H", grid.whg.dat$HAUL, sep = "")

## make some factor variables used in the analyses
grid.whg.dat$fHAUL <- factor(grid.whg.dat$HAUL, levels = unique(grid.whg.dat$HAUL))
grid.whg.dat$COMPARTMENT <- factor(grid.whg.dat$COMPARTMENT)

## remove observations above 99th and below 1th length percentile
## these can be highly influential on the fits
grid.whg.dat <- subset(grid.whg.dat, TOTAL.LENGTH < quantile(TOTAL.LENGTH, 0.99) &
  TOTAL.LENGTH > quantile(TOTAL.LENGTH, 0.01)
)

## Show the first 2 rows
head(grid.whg.dat, 2)

##           ID HAUL      DATE COMPARTMENT POSITION SPECIES TOTAL.LENGTH
## 1 1NSG1Whiting   H1 2015-09-21      NSG1      SO Whiting          16
## 2 1NSG1Whiting   H1 2015-09-21      NSG1      SO Whiting          17
##      COUNT OVERALL.RAISING.FACTOR RAISED.COUNT fHAUL
## 1         3                     1             3   H1
## 2         1                     1             1   H1
```

```

## Bring in rotation data (used further down)
rotation.dat <-
  read.xls(
    "../data/Do_Not_Overwrite_OUR_LASSII_Grid_Trial_FU15_September_2015.xlsx",
    sheet = "Rotations",
    stringsAsFactors = FALSE, nrows = 13)

## need to create a rotation variable
## use short codes for rotation names
## starboard outside
rotation.dat$SO <- NA
rotation.dat$SO[rotation.dat$Starboard.Outside == "Nephrops sorting grid 1"] <- "NSG1"
rotation.dat$SO[rotation.dat$Starboard.Outside == "Nephrops sorting grid 2"] <- "NSG2"
rotation.dat$SO[rotation.dat$Starboard.Outside == "Swedish grid"] <- "SG"
rotation.dat$SO[rotation.dat$Starboard.Outside == "Control 70mm"] <- "CTRL"

## starboard inside
rotation.dat$SI <- NA
rotation.dat$SI[rotation.dat$Starboard.Inside == "Nephrops sorting grid 1"] <- "NSG1"
rotation.dat$SI[rotation.dat$Starboard.Inside == "Nephrops sorting grid 2"] <- "NSG2"
rotation.dat$SI[rotation.dat$Starboard.Inside == "Swedish grid"] <- "SG"
rotation.dat$SI[rotation.dat$Starboard.Inside == "Control 70mm"] <- "CTRL"

## Port outside
rotation.dat$PO <- NA
rotation.dat$PO[rotation.dat$Port.Outside == "Nephrops sorting grid 1"] <- "NSG1"
rotation.dat$PO[rotation.dat$Port.Outside == "Nephrops sorting grid 2"] <- "NSG2"
rotation.dat$PO[rotation.dat$Port.Outside == "Swedish grid"] <- "SG"
rotation.dat$PO[rotation.dat$Port.Outside == "Control 70mm"] <- "CTRL"

## Port inside
rotation.dat$PI <- NA
rotation.dat$PI[rotation.dat$Port.Inside == "Nephrops sorting grid 1"] <- "NSG1"
rotation.dat$PI[rotation.dat$Port.Inside == "Nephrops sorting grid 2"] <- "NSG2"
rotation.dat$PI[rotation.dat$Port.Inside == "Swedish grid"] <- "SG"
rotation.dat$PI[rotation.dat$Port.Inside == "Control 70mm"] <- "CTRL"

## get a unique net configuration variable
rotation.dat$netconfig <- with(rotation.dat, paste(SO, SI, PO, PI, sep = ":"))

rotation.dat$SHAUL <- paste("H", rotation.dat$Haul., sep = "")
rotation.dat$fHAUL <- factor(rotation.dat$SHAUL, levels = unique(rotation.dat$SHAUL))

```

Data pre-processing to format needed for model fits

```

## get count per length bin per haul by mesh size
## using the reshape package (makes it easier to process data)
library(reshape)

## variables to keep
vars2keep <- c("COMPARTMENT", "TOTAL.LENGTH", "fHAUL", "COUNT")

## melt the data frame
grid.whg.melt <- melt(grid.whg.dat[, vars2keep],
  id = c("COMPARTMENT", "TOTAL.LENGTH", "fHAUL"))

## re-form the dataframe in required format
grid.whg.cast <- cast(grid.whg.melt, TOTAL.LENGTH + fHAUL ~ COMPARTMENT + variable)
grid.whg.cast <- grid.whg.cast[order(grid.whg.cast$fHAUL, grid.whg.cast$TOTAL.LENGTH), ]
grid.whg.cast[is.na(grid.whg.cast)] <- 0

```

```

## merge in the net position
grid.whg.cast <- merge(grid.whg.cast, rotation.dat[, c("fHAUL", "netconfig")])

## merge in the bulk weights
##bulk.weight.melt <- melt(unique(grid.whg.dat[, c("fHAUL", "COMPARTMENT", "Bulk.weight")
##                                id = c("COMPARTMENT", "fHAUL"))

##bulk.weight.cast <- cast(bulk.weight.melt, fHAUL ~ COMPARTMENT + variable)

##grid.whg.cast <- merge(grid.whg.cast, bulk.weight.cast)

## show the first few rows
head(grid.whg.cast, 2)

##      fHAUL TOTAL.LENGTH CTRL_COUNT NSG1_COUNT NSG2_COUNT SG_COUNT
## 1      H1           12           0           0           1           0
## 2      H1           13           0           0           1           0
##      netconfig
## 1 NSG1:SG:CTRL:NSG2
## 2 NSG1:SG:CTRL:NSG2

## format the subsampling ratio similarly
grid.whg.dat$OVERALL.SUBS.RATIO <- 1 / grid.whg.dat$OVERALL.RAISING.FACTOR

## double-check that there are unique raising factors per haul
rf.count <- with(grid.whg.dat, table(fHAUL, OVERALL.SUBS.RATIO, COMPARTMENT))
apply(rf.count, 1, FUN = function(x){sum(x>0)}) ## yes

##      H1  H2  H3  H5  H6  H7  H8  H9 H10 H11 H12 H13
##      4   4   4   4   4   4   4   4   3   4   4   4

## convert to sub-sampling ratio as in Celtic Warrior
names(grid.whg.dat)[names(grid.whg.dat) == "OVERALL.SUBS.RATIO"] <- "SUBSRATIO"
vars2keep <- c("COMPARTMENT", "fHAUL", "SUBSRATIO")
subs.melt <- melt(unique(grid.whg.dat[, vars2keep]), id = c("COMPARTMENT", "fHAUL"))
subs.cast <- cast(subs.melt, fHAUL ~ COMPARTMENT + variable)

## no value for NSG2 SUBSRATIO - assigning to a value of 1 for the moment
subs.cast$NSG2_SUBSRATIO[subs.cast$fHAUL == "H10"] <- 1

## merge counts and subsampling ratio back together
grid.whg.cast <- merge(grid.whg.cast, subs.cast, by = "fHAUL", all.x = TRUE)

## show first few lines
## Note that this is how the data look just prior to analysis
head(grid.whg.cast, 2)

##      fHAUL TOTAL.LENGTH CTRL_COUNT NSG1_COUNT NSG2_COUNT SG_COUNT
## 1      H1           12           0           0           1           0
## 2      H1           13           0           0           1           0
##      netconfig CTRL_SUBSRATIO NSG1_SUBSRATIO NSG2_SUBSRATIO
## 1 NSG1:SG:CTRL:NSG2           1           1           1
## 2 NSG1:SG:CTRL:NSG2           1           1           1
##      SG_SUBSRATIO
## 1           1
## 2           1

```

Extract the data in exact format for a multinomial fit.

```
## Extract the matrix of counts
count.vars <- c("CTRL_COUNT", "NSG1_COUNT", "NSG2_COUNT", "SG_COUNT")

whg.count.mat <- as.matrix(grid.whg.cast[, count.vars])

## Extract the matrix of subsampling ratios
subsratio.vars <- c("CTRL_SUBSRATIO", "NSG1_SUBSRATIO", "NSG2_SUBSRATIO", "SG_SUBSRATIO")

subsratio.mat <- as.matrix(grid.whg.cast[, subsratio.vars])

## Create the offset
offset.mat <- log(subsratio.mat / subsratio.mat[,1])
```

Plot the data

```
library(ggplot2)

## N.B. to plot the proportions correctly, use the raised counts
## raised count per compartment and haul and carapace length
raised.count.compartment <- whg.count.mat / subsratio.mat

## Get the proportions
prop.compartment <- prop.table(raised.count.compartment, margin = 1)

m <- dim(prop.compartment)[1]

## make a dataframe of the proportions for ggplot
prop.compartment.df <- data.frame(
  proportion = c(prop.compartment),
  count = c(raised.count.compartment),
  TOTAL.LENGTH = rep(grid.whg.cast$TOTAL.LENGTH, times = 4),
  COMPARTMENT = rep(c("Control", "Nephrops sorting grid 1",
    "Nephrops sorting grid 2", "Swedish grid"), each = m)
)

theme_set(theme_bw())

p <- ggplot(prop.compartment.df, aes(x = TOTAL.LENGTH, y = proportion)) +
  geom_point(colour = "darkgreen", alpha = 0.2, aes(size = 2*log(count))) +
  facet_wrap(~ COMPARTMENT) + ylab("Proportion of whiting per compartment") +
  xlab("Total length (cm)") +
  theme(legend.position = "bottom")

p
```

2 Model

The first model we can use is a multinomial for the proportion retained in the four compartments.

```
library(nnet)

## First fit is constant proportions
## not accounting for length

mnom0 <- multinom(whg.count.mat ~ 1 + offset(offset.mat))

## # weights: 24 (3 variable)
```

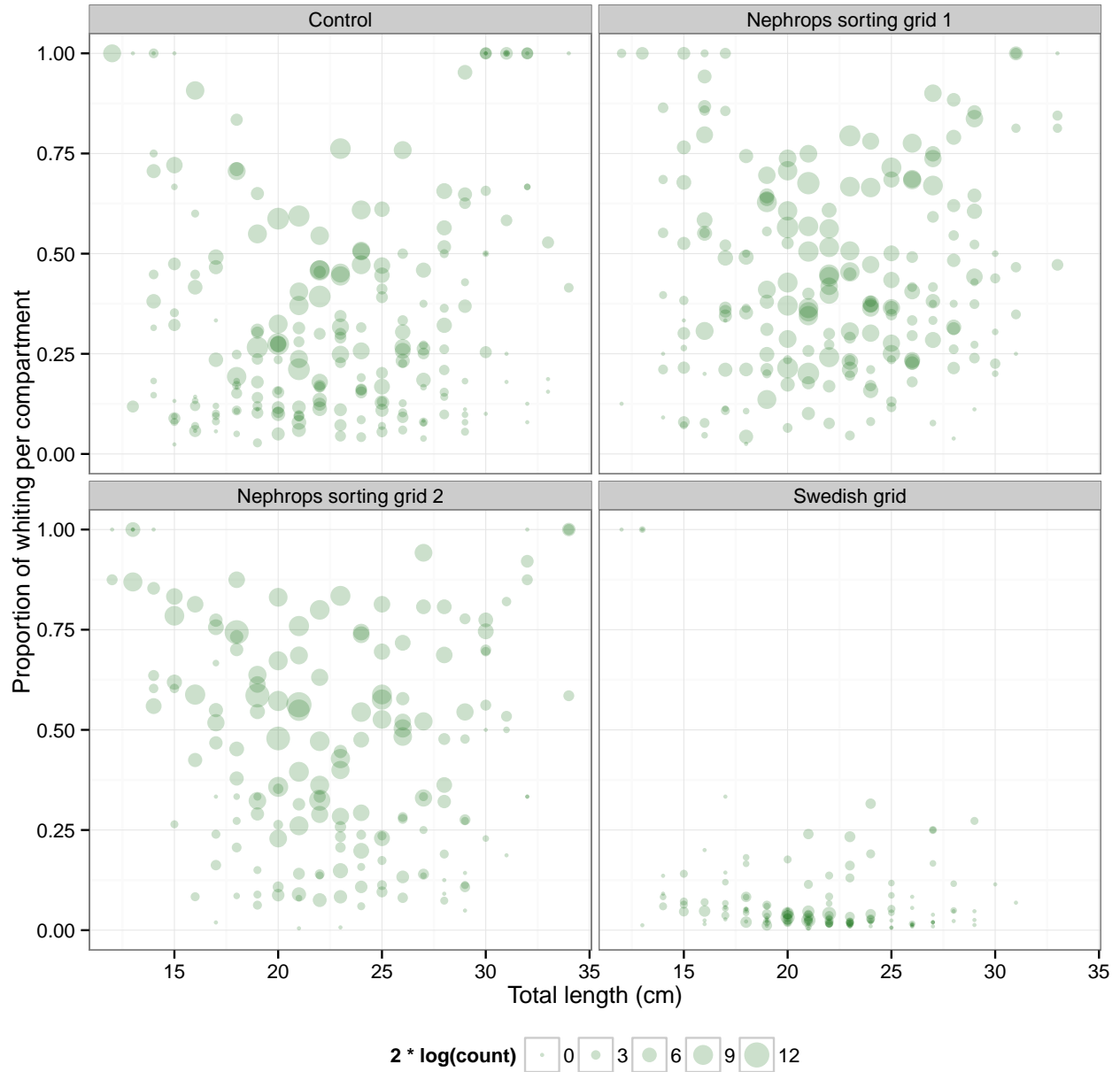


Figure 1: Proportion of whiting catch retained per haul. Each point represents the proportion of raised whiting catch per haul and length class retained in a given cod-end (Control: CTRL, Nephrops sorting grid 1: NSG1, Nephrops sorting grid 2: NSG2, or the Swedish Grid). The size of the point is proportional to the log of the count.

```

## initial value 6484.787263
## final value 5402.678685
## converged

## second fit include net configuration/rotations
mnom0.1 <- multinom(whg.count.mat ~ netconfig +
                    offset(offset.mat), data = grid.whg.cast)

## # weights: 36 (12 variable)
## initial value 6484.787263
## iter 10 value 4974.370768
## final value 4833.915629
## converged

## include carapace length polynomials of different complexity
mnom1 <- update(mnom0.1, . ~ . + poly(TOTAL.LENGTH, 1),
               data = grid.whg.cast)

## # weights: 40 (15 variable)
## initial value 6484.787263
## iter 10 value 4972.140159
## iter 20 value 4822.312835
## final value 4822.310892
## converged

mnom2 <- update(mnom0.1, . ~ . + poly(TOTAL.LENGTH, 2),
               data = grid.whg.cast)

## # weights: 44 (18 variable)
## initial value 6484.787263
## iter 10 value 4967.527498
## iter 20 value 4820.324322
## final value 4820.314123
## converged

mnom3 <- update(mnom0.1, . ~ . + poly(TOTAL.LENGTH, 3), data = grid.whg.cast)

## # weights: 48 (21 variable)
## initial value 6484.787263
## iter 10 value 4966.280119
## iter 20 value 4820.170772
## final value 4819.869672
## converged

mnom4 <- update(mnom0.1, . ~ . + poly(TOTAL.LENGTH, 4), data = grid.whg.cast)

## # weights: 52 (24 variable)
## initial value 6484.787263
## iter 10 value 4964.317977
## iter 20 value 4814.728213
## iter 30 value 4814.192615
## iter 30 value 4814.192596
## iter 30 value 4814.192596
## final value 4814.192596
## converged

AIC(mnom0, mnom0.1, mnom1, mnom2, mnom3, mnom4)

##          df          AIC
## mnom0      3 10811.357
## mnom0.1    12  9691.831
## mnom1      15  9674.622
## mnom2      18  9676.628
## mnom3      21  9681.739
## mnom4      24  9676.385

## looks like a quadratic carapace length effect fits best

```

Get predictions for the fitted model (note that this is run in a cleaner fashion in the ADMB fit below).

```
## get predictions manually
## CIs not defined in multinomial context but let's try

best.model <- mnom2

## fit coefficients
beta.mu <- c(t(coef(best.model)))

## fit coefficient variance covariance matrix
Sigma <- vcov(best.model)

## number of lengths to predict for
nlength <- 50

pred.length <- seq(min(grid.whg.cast$TOTAL.LENGTH),
                   max(grid.whg.cast$TOTAL.LENGTH), length = nlength)

## get the polynomial of lengths
polyfun <- poly(grid.whg.cast$TOTAL.LENGTH, 2)

## model matrix
Xpred <- cbind(1, 1/3, 1/3, 1/3, predict(polyfun, pred.length))

## number of times to resample predictions to get CIs
nresamp <- 1e3
pred.array <- array(NA, dim = c(nlength, 4, nresamp))

## package to draw from multivariate normal
library(mvtnorm)

for(i in 1:nresamp){
  ##print(i)
  beta0 <- matrix(rmnorm(1, mean = beta.mu, sigma = Sigma),
                 nrow = 3, byrow = TRUE)
  beta <- cbind(0, t(beta0))
  eta <- Xpred %*% beta
  pred.p <- exp(eta) / rowSums(exp(eta))
  pred.array[, , i] <- pred.p
  rm(pred.p)
}

## mean across samples
pred.mu <- apply(pred.array, c(1, 2), mean)

## upper across samples
pred.upper <- apply(pred.array, c(1, 2), quantile, p = 0.975)

## lower across samples
pred.lower <- apply(pred.array, c(1, 2), quantile, p = 0.025)

## bring all together in a data frame for ggplot
m <- dim(pred.mu)[1]

pred.ci.df <- data.frame(
  COMPARTMENT = rep(c("Control", "Nephrops sorting grid 1",
                     "Nephrops sorting grid 2", "Swedish grid"), each = m),
  TOTAL.LENGTH = rep(pred.length, times = 4),
```

```
proportion = c(pred.mu),  
lower = c(pred.lower),  
upper = c(pred.upper))
```

Finally overlay the fit on the sample proportions

```
p + geom_ribbon(data = pred.ci.df, aes(ymin = lower, ymax = upper),  
              alpha = 0.3, fill = "purple") +  
  geom_line(data = pred.ci.df, aes(x = TOTAL.LENGTH, y = proportion),  
            col = "purple", size = 0.5) +  
  geom_hline(aes(yintercept = 0.25), linetype = "dashed")
```

Overlay the model predictions

```
ggplot(pred.ci.df, aes(x = TOTAL.LENGTH, y = proportion, group = COMPARTMENT)) + geom_line
```

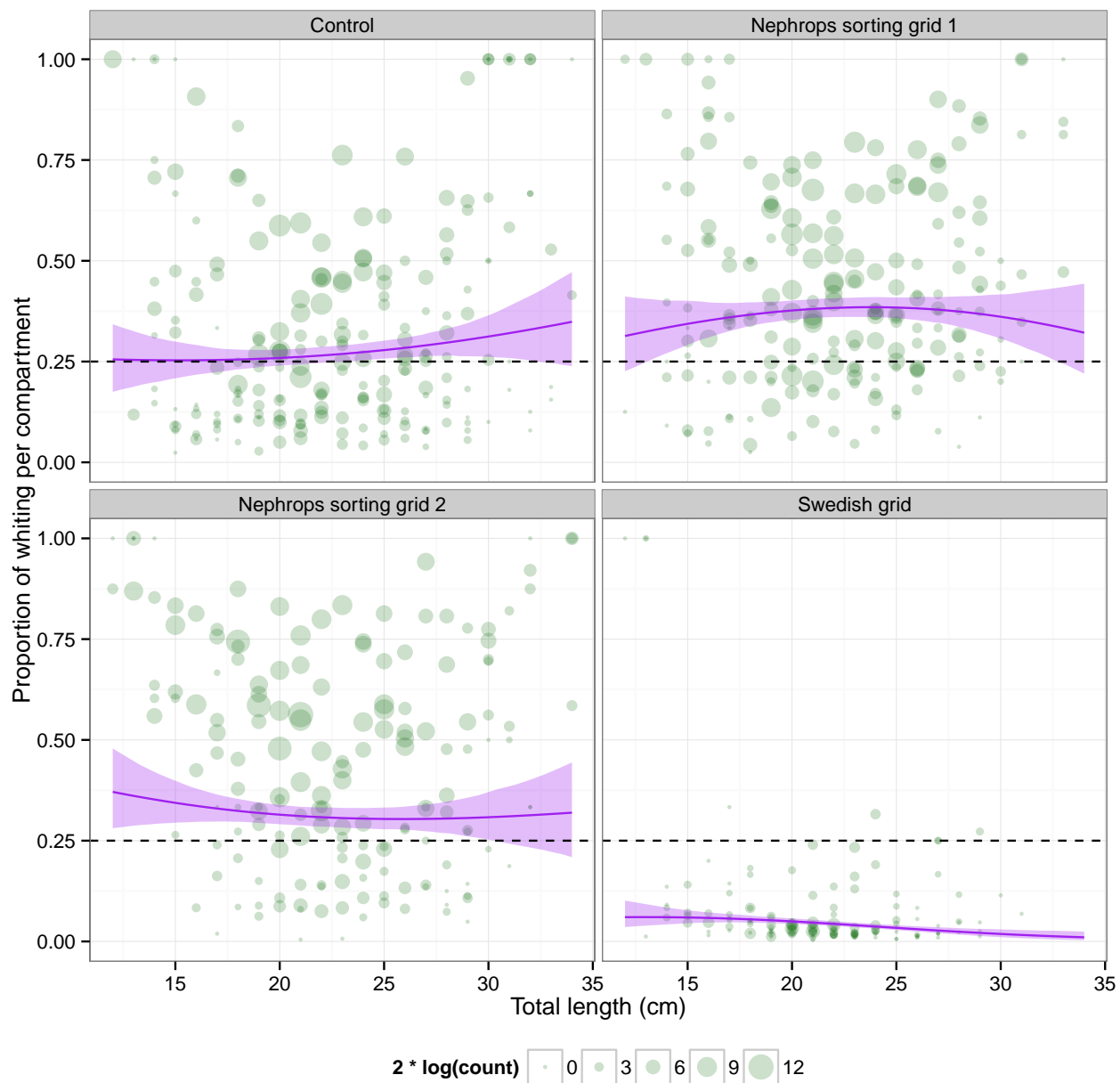
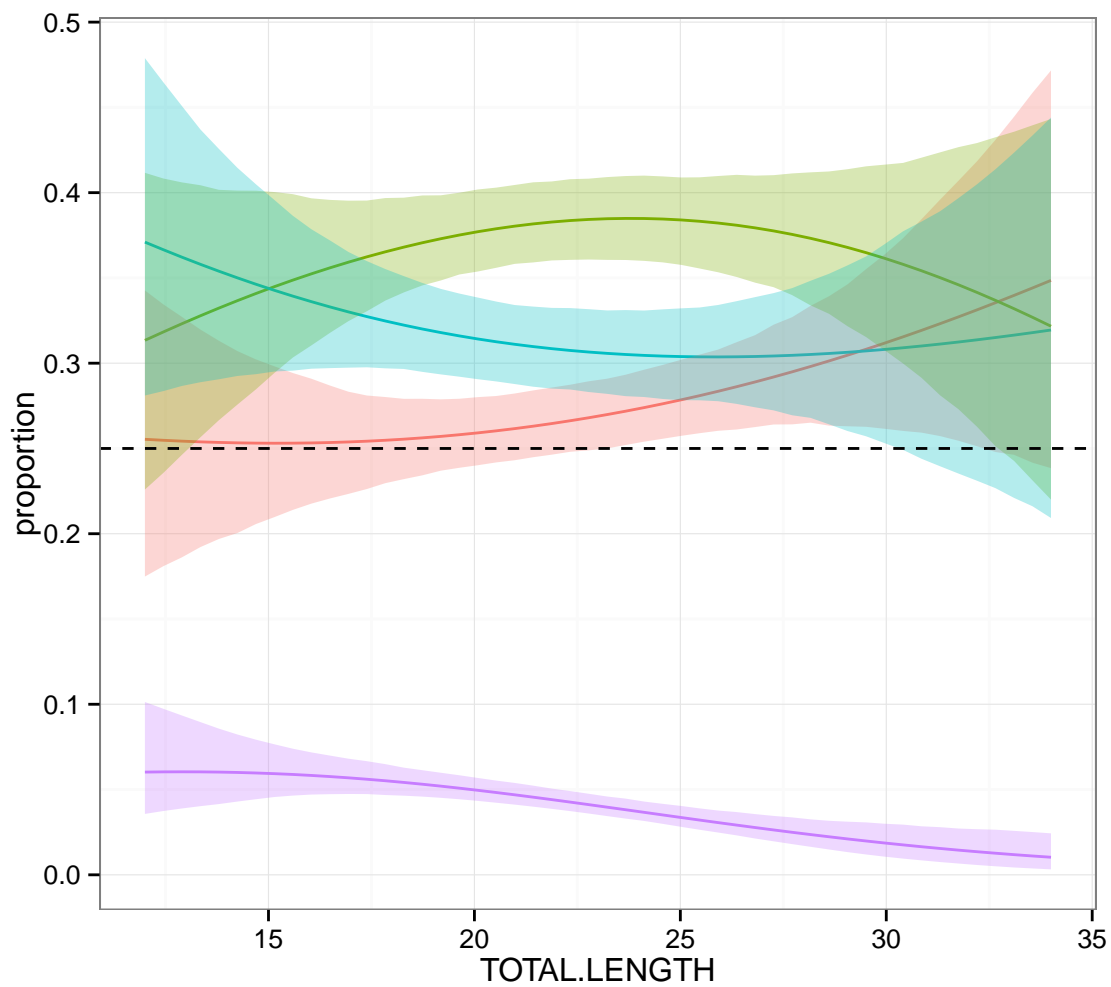



Figure 2: Proportion of whiting catch retained per haul with fitted multinomial model (without a weight effect or random effects) and associated re-sampled intervals. Null hypothesis of equal retention is displayed as the dashed line at 0.25.



COMPARTMENT Control Nephrops sorting grid 1 Nephrops sorting grid 2 Swedi

Figure 3: Multinomial model predicted proportion of whiting catch retained by compartment. No weight effects or random effects are included in this model. Null hypothesis of equal retention is displayed as the dashed line at 0.25.