Grid trial multinomial fits to the *Nephrops* data

For discussion only

January 15, 2016

1 Data

```
library(gdata)

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

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

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

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

## NOT REMOVING HERE
## remove observations above 99th and below 1th length percentile
## these can be highly influential on the fits
##grid.neph.dat <- subset(grid.neph.dat, Carapace.length < quantile(Carapace.length, 0.99)
##                   Carapace.length > quantile(Carapace.length, 0.01)
##                   )

## change a few names
names(grid.neph.dat)[names(grid.neph.dat) == "CARAPACE.LENGTH..mm."] <- "Carapace.length"
names(grid.neph.dat)[names(grid.neph.dat) == "BULK.CATCH...kg."] <- "Bulk.weight"

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

##   HAUL COMPARTMENT Carapace.length COUNT RAISED.COUNT Bulk.weight
## 1   H1          SG              13     2           NA         132
## 2   H1          SG              15     3           NA         132
##   BULK.CATCH.SUBSAMPLE.WEIGHT..kg. BULK.CATCH.SUBSAMPLE.RATIO
## 1                              132                          1
## 2                              132                          1
##   NEPHROPS.WEIGHT.IN.SUBSAMPLE..kg.
## 1                              89.6
## 2                              89.6
##   SUBSAMPLE.OF.NEPHROPS.FOR.MEASUREMENT..kg. NEPHROPS.SUBSAMPLE.RATIO
```

```
## 1                                          5.2                 17.23077
## 2                                          5.2                 17.23077
##   OVERALL.RAISING.FACTOR OVERALL.SUBS.RATIO fHAUL
## 1               17.23077         0.05803571    H1
## 2               17.23077         0.05803571    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$HAUL <- paste("H", rotation.dat$Haul.., sep = "")
rotation.dat$fHAUL <- factor(rotation.dat$HAUL, levels = unique(rotation.dat$HAUL))
```

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", "Carapace.length", "fHAUL", "COUNT")

## melt the data frame
grid.neph.melt <- melt(grid.neph.dat[, vars2keep],
```

```
                  id = c("COMPARTMENT", "Carapace.length", "fHAUL"))

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

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

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

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

grid.neph.cast <- merge(grid.neph.cast, bulk.weight.cast)

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

```
##   fHAUL Carapace.length CTRL_COUNT NSG1_COUNT NSG2_COUNT SG_COUNT
## 1    H1              13          0          0          0        2
## 2    H1              15          0          1          0        3
##           netconfig CTRL_Bulk.weight NSG1_Bulk.weight NSG2_Bulk.weight
## 1 NSG1:SG:CTRL:NSG2           375.75            467.5             294.7
## 2 NSG1:SG:CTRL:NSG2           375.75            467.5             294.7
##   SG_Bulk.weight
## 1            132
## 2            132
```

```
## format the subsampling ratio similarly

## double-check that there are unique raising factors per haul
rf.count <- with(grid.neph.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   4   4   4   4
```

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

## merge counts and subsampling ratio back together
grid.neph.cast <- merge(grid.neph.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.neph.cast, 2)
```

```
##   fHAUL Carapace.length CTRL_COUNT NSG1_COUNT NSG2_COUNT SG_COUNT
## 1    H1              13          0          0          0        2
## 2    H1              15          0          1          0        3
##           netconfig CTRL_Bulk.weight NSG1_Bulk.weight NSG2_Bulk.weight
## 1 NSG1:SG:CTRL:NSG2           375.75            467.5             294.7
## 2 NSG1:SG:CTRL:NSG2           375.75            467.5             294.7
##   SG_Bulk.weight CTRL_SUBSRATIO NSG1_SUBSRATIO NSG2_SUBSRATIO SG_SUBSRATIO
## 1            132     0.05051344      0.0614082       0.140078   0.05803571
## 2            132     0.05051344      0.0614082       0.140078   0.05803571
```

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")

neph.count.mat <- as.matrix(grid.neph.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.neph.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 <- neph.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),
                         Carapace.length = rep(grid.neph.cast$Carapace.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 = Carapace.length, y = proportion)) +
  geom_point(colour = "#F8766D", alpha = 0.2, aes(size = 2*log(count))) +
facet_wrap(~ COMPARTMENT) + ylab("Proportion of Nephrops per compartment") +
  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(neph.count.mat ~ 1 + offset(offset.mat))
```

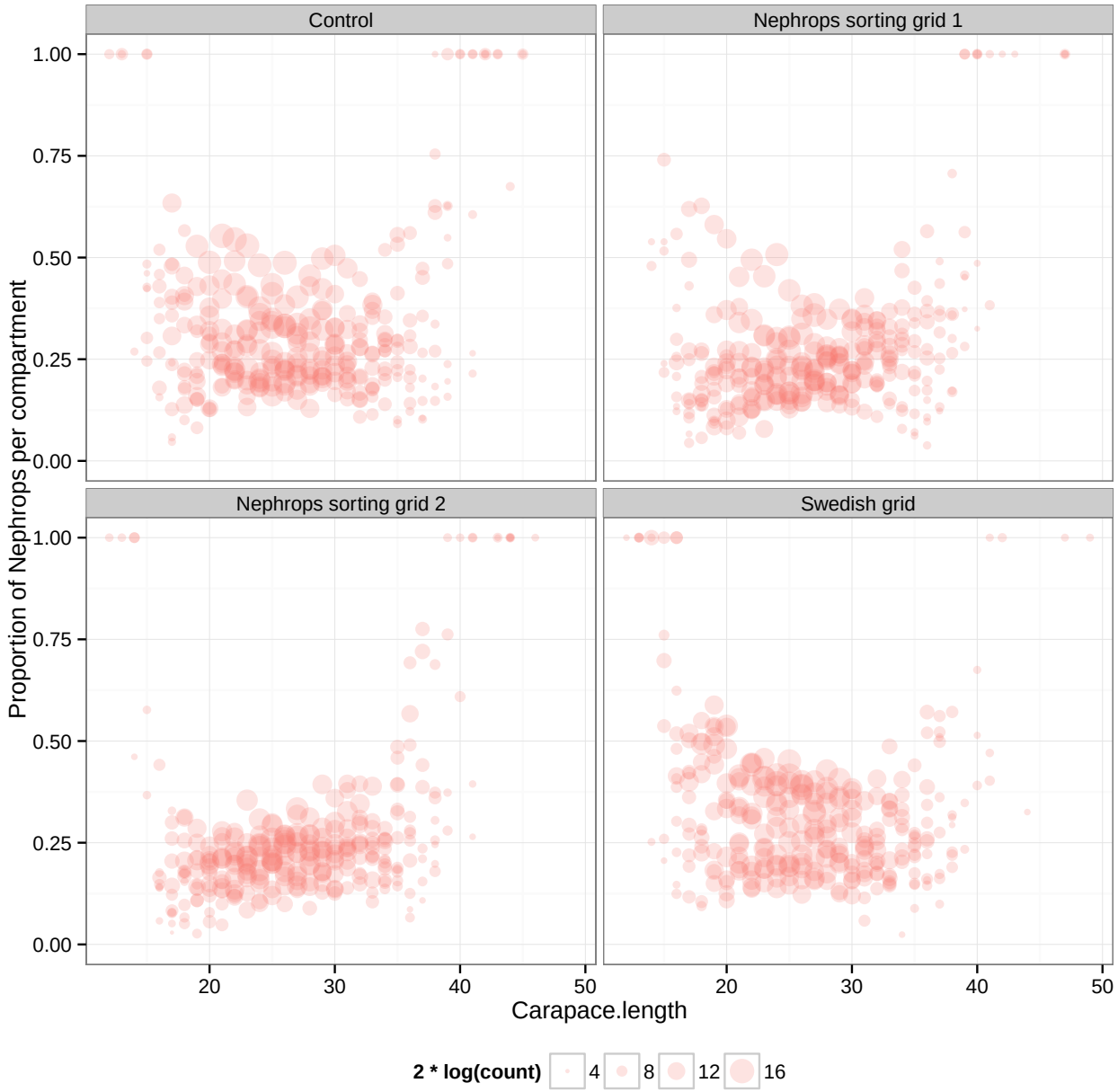

Figure 1: Proportion of Nephrops catch retained per haul. Each point represents the proportion of raised Nephrops catch per haul and length class retained in a given cod-end (Control: CTRL, Nephrops Sorting grid 1: NSG1, Nephrops Sorting grid 2: NSG2, or the Swedish Grid). The size of the point is proportional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

```
## # weights:  24 (3 variable)
## initial  value 28451.789893
## final  value 28311.405094
## converged

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

## # weights:  36 (12 variable)
## initial  value 28451.789893
## iter  10 value 27887.473369
## final  value 27749.753260
## converged

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

## # weights:  40 (15 variable)
## initial  value 28451.789893
## iter  10 value 27869.288971
## final  value 27696.063407
## converged

mnom2 <- update(mnom0.1, . ~ . + poly(Carapace.length, 2),
                data = grid.neph.cast)

## # weights:  44 (18 variable)
## initial  value 28451.789893
## iter  10 value 27861.252166
## iter  20 value 27691.533797
## iter  20 value 27691.533734
## iter  20 value 27691.533734
## final  value 27691.533734
## converged

mnom3 <- update(mnom0.1, . ~ . + poly(Carapace.length, 3), data = grid.neph.cast)

## # weights:  48 (21 variable)
## initial  value 28451.789893
## iter  10 value 27855.790341
## iter  20 value 27690.817351
## final  value 27690.682490
## converged

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

##         df      AIC
## mnom0    3 56628.81
## mnom0.1 12 55523.51
## mnom1   15 55422.13
## mnom2   18 55419.07
## mnom3   21 55423.36

## 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).

```r
## 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.neph.cast$Carapace.length),
                   max(grid.neph.cast$Carapace.length), length = nlength)

## get the polynomial of lengths
polyfun <- poly(grid.neph.cast$Carapace.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(rmvnorm(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),
                Carapace.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 = "blue") +
  geom_line(data = pred.ci.df, aes(x = Carapace.length, y = proportion),
            col = "navy", size = 0.5) +
  geom_hline(aes(yintercept = 0.25), linetype = "dashed")
```

2.1 Including weight as a covariate

Here we use the ADMB code to fit weight as a compartment-specific variable. Please note we are working on ways to better integrate this into R.

```
Yobs <- neph.count.mat
n <- nrow(Yobs)
m <- ncol(Yobs)
X <- model.matrix(best.model)
## here is where the bulk weights enter
Xcond <- grid.neph.cast[, c("CTRL_Bulk.weight", "NSG1_Bulk.weight",
                            "NSG2_Bulk.weight", "SG_Bulk.weight")]
## hauls
gps <- as.numeric(grid.neph.cast$fHAUL)
ngp <- length(unique(gps))

## predicted values of weight
## get median bulk weight per net

median.weights <- with(bulk.weight.melt, tapply(value, COMPARTMENT, median))
median.weights

##   CTRL   NSG1   NSG2     SG
## 418.65 346.85 391.00 184.35

## construct the "conditional" prediction matrix
npred <- nrow(Xpred)
Xcondpred <- matrix(median.weights, nrow = 1)[rep(1, npred), ]

## Output the data to ADMB
## write the data out
datfile <- "../admbre/multinomialme.dat"
cat("# number of observations n \n", n, "\n", file = datfile)
cat("# number of categories m \n", m, "\n", file = datfile, append = TRUE)
cat("# dimension of parameter vector p \n", ncol(X), "\n",
    file = datfile, append = TRUE)
cat("# dimension of conditional variables q \n", ncol(Xcond), "\n",
    file = datfile, append = TRUE)
cat("# number of groups ngp \n", ngp, "\n", file = datfile, append = TRUE)
cat("# response counts Y \n", file = datfile, append = TRUE)
write.table(Yobs, file = datfile, append = TRUE,
            col.names = FALSE, row.names = FALSE)
cat("# model/design matrix X \n", file = datfile, append = TRUE)
write.table(X, file = datfile, append = TRUE,
            col.names = FALSE, row.names = FALSE)
cat("# conditional matrix Xcond \n", file = datfile, append = TRUE)
write.table(Xcond, file = datfile, append = TRUE,
            col.names = FALSE, row.names = FALSE)
cat("# groups \n", gps, "\n", file = datfile, append = TRUE)
cat("# Offset (log(q[i]/q[1])) \n", file = datfile, append = TRUE)
```

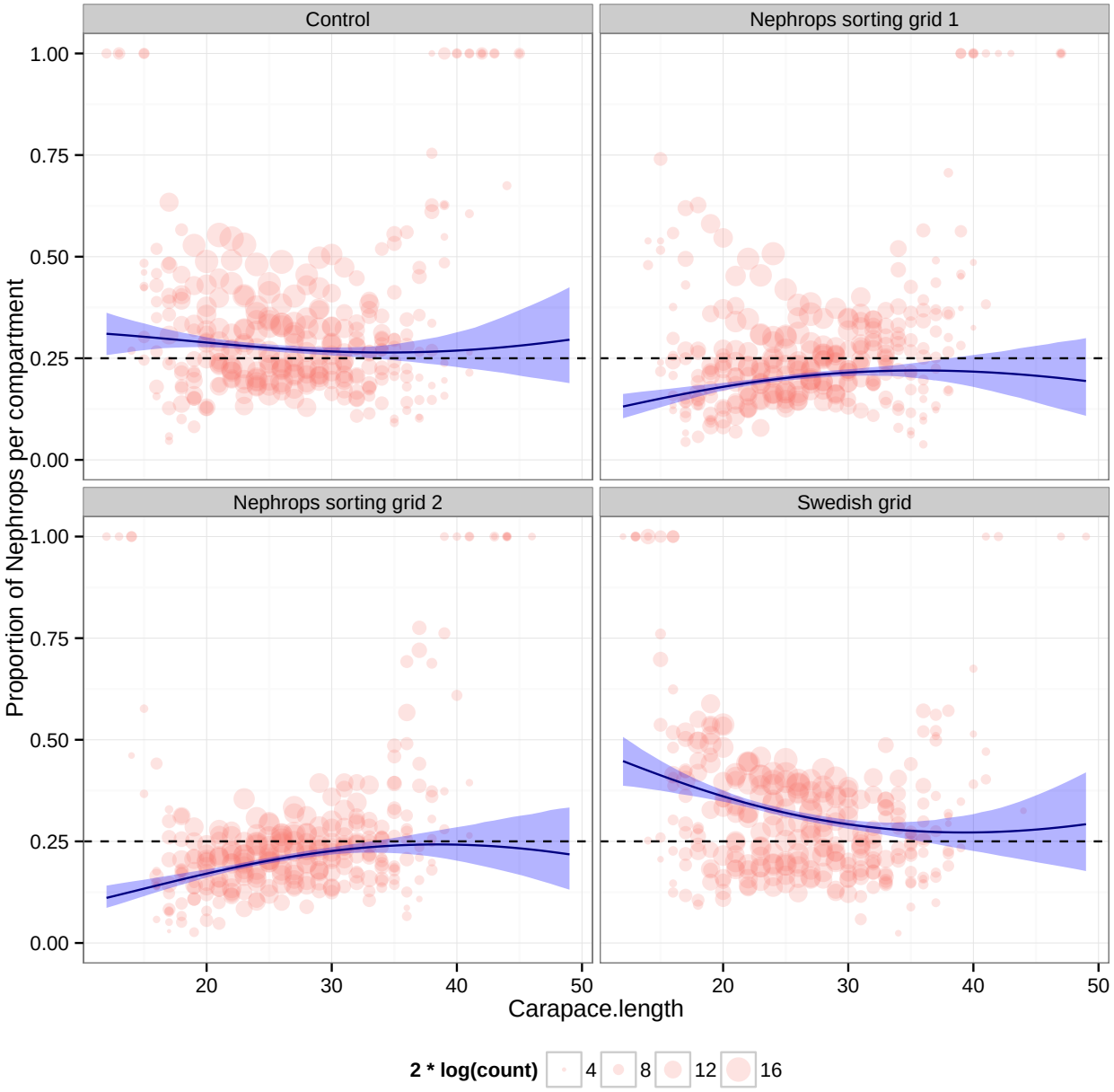

Figure 2: Proportion of Nephrops catch retained per haul with fitted multinomial model (without a weight effect or random effects) and associated re-sampled intervals. Null hypothesis of equal retention is displayed as the dashed line at 0.25.

```
write.table(offset.mat, file = datfile, append = TRUE,
            col.names = FALSE, row.names = FALSE)

## predictions
cat("# number of prediction rows npred \n", npred, "\n",
    file = datfile, append = TRUE)
cat("# prediction matrix Xpred \n", file = datfile, append = TRUE)
write.table(Xpred, file = datfile, append = TRUE,
            col.names = FALSE, row.names = FALSE)
cat("# conditional prediction matrix Xcondpred \n",
    file = datfile, append = TRUE)
write.table(Xcondpred, file = datfile, append = TRUE,
            col.names = FALSE, row.names = FALSE)

## starting values pinfile
beta0.start <- matrix(0, nrow = ncol(X), ncol = ncol(Yobs) - 1)
u0.start <- matrix(0, nrow = ngp, ncol = m - 1)
nchol <- (m-1)*(m)/2
L.start <- diag(m-1)
L.start[upper.tri(L.start)] <- NA
a.start <- as.numeric(na.omit(c(t(L.start))))
##a.start <- rep(0.1, 3)

pinfile <- "../admbre/multinomialme.pin"
cat("# beta0 \n", file = pinfile)
write.table(beta0.start, file = pinfile, append = TRUE,
            col.names = FALSE, row.names = FALSE)
cat("# betacond \n", 0, "\n", file = pinfile, append = TRUE)
##set.seed(10)
cat("# a \n", a.start, "\n", file = pinfile, append = TRUE)
##cat("# a \n", rep(0.1, m-1), "\n", file = pinfile, append = TRUE)
cat("# u0 \n", file = pinfile, append = TRUE)
write.table(u0.start, file = pinfile, append = TRUE,
            col.names = FALSE, row.names = FALSE)

## RUN THE MODEL IN ADMB
```

Plot the model fits

```
coef.admb <- read.table("../admbre/multinomialme.std", header = TRUE)

##
p0 <- dim(X)[2]
beta.hat <- cbind(0, matrix(subset(coef.admb, name == "beta0")$value,
                            nrow = p0, byrow = TRUE))

betacond.hat <- subset(coef.admb, name == "betacond")$value

## predictions from admb
eta.hat.admb <- matrix(subset(coef.admb, name == "etapred")$value,
                       ncol = 4, byrow = TRUE)
P.hat.admb <- exp(eta.hat.admb) / rowSums(exp(eta.hat.admb))
eta.se.admb <- matrix(subset(coef.admb, name == "etapred")$std.dev,
                      ncol = 4, byrow = TRUE)

eta.lwr.admb <- eta.hat.admb + qnorm(0.025) * eta.se.admb
eta.upr.admb <- eta.hat.admb + qnorm(0.975) * eta.se.admb

P.admb.lwr <- exp(eta.lwr.admb) / rowSums(exp(eta.lwr.admb))
P.admb.upr <- exp(eta.upr.admb) / rowSums(exp(eta.upr.admb))
```

```r
plot.pred.df.admb <- data.frame(
                       COMPARTMENT = rep(c("Control", "Nephrops sorting grid 1",
                           "Nephrops sorting grid 2", "Swedish grid"),
                         each = nrow(P.hat.admb)),
                       Carapace.length = rep(pred.length, times = 4),
                       proportion = c(P.hat.admb),
                       lwr = c(P.admb.lwr),
                       upr = c(P.admb.upr))

p + geom_ribbon(data = plot.pred.df.admb, aes(ymin = lwr, ymax = upr),
                  alpha = 0.3, fill = "blue") +
  geom_line(data = plot.pred.df.admb, aes(x = Carapace.length, y = proportion),
            col = "navy", size = 0.5) +
  geom_hline(aes(yintercept = 0.25), linetype = "dashed")
```

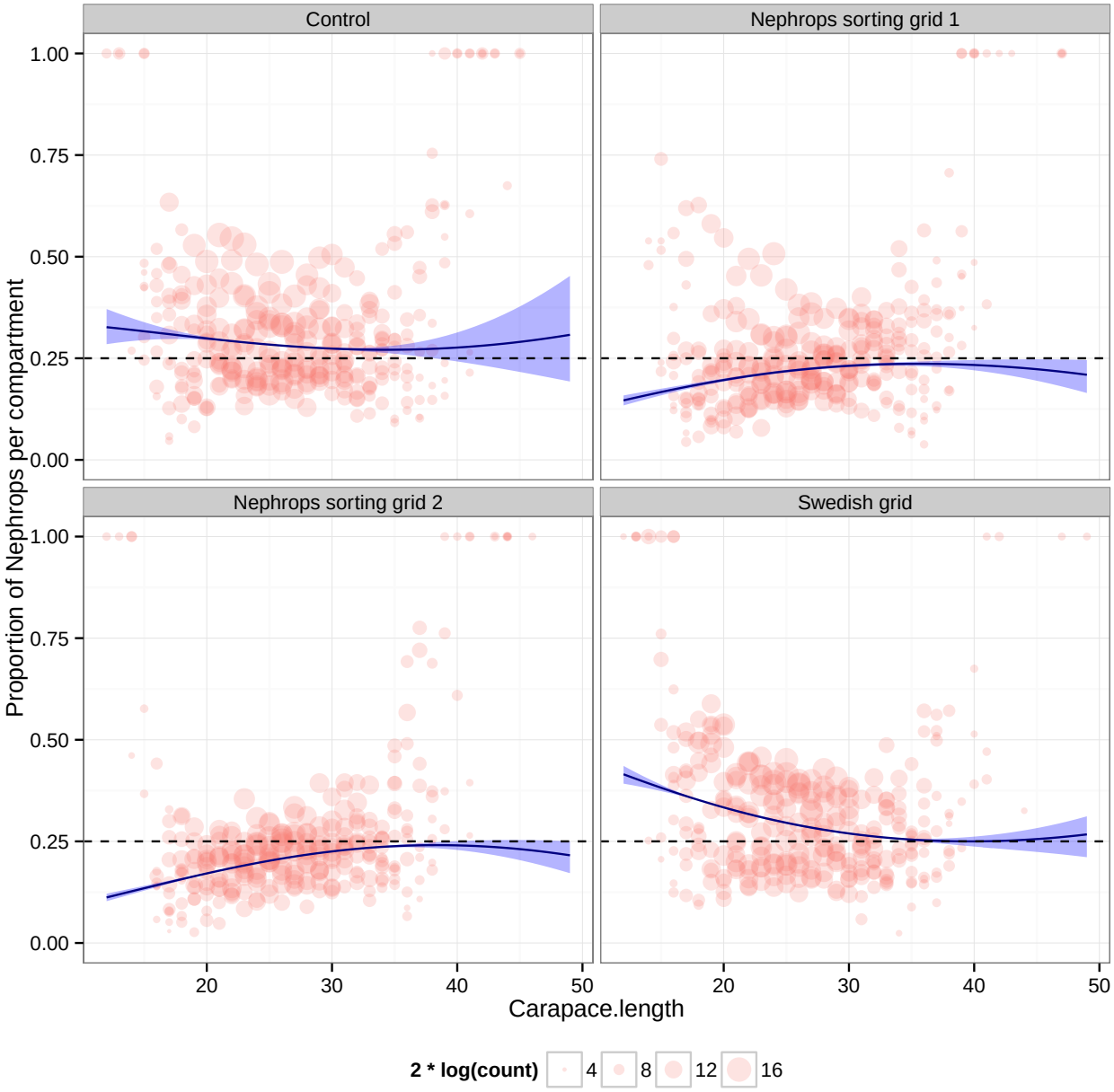

Figure 3: Proportion of Nephrops catch retained per haul with fitted ADMB multinomial model (with bulk weights set to the mean bulk per compartment). No random effects were estimated here due to convergence issues which will be investigated.