

Preliminary multinomial analysis of *Our Lass* data

For discussion

1 Data

```
library(gdata)

our.lass.neph.dat <- read.xls("../data//2015 BIM Nephrops quad rig trials/Our Lass 2 70_80_90",
                              sheet = "All hauls",
                              stringsAsFactors = FALSE)

## Each mesh size has a unique codend number
with(our.lass.neph.dat, table(Mesh.Size, Codend.No))

## position switched
with(our.lass.neph.dat, table(Mesh.Size, Net.position))

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

## Make the "HAUL" variable character
our.lass.neph.dat$HAUL <- paste("H", our.lass.neph.dat$Haul.No, sep = "")

## make some factor variables used in the analyses
our.lass.neph.dat$fHAUL <- factor(our.lass.neph.dat$HAUL, levels = unique(our.lass.neph.dat$HAUL))
our.lass.neph.dat$Mesh.Size <- factor(paste("C", our.lass.neph.dat$Mesh.Size, sep = ""))

## remove observations above 99th and below 1th length percentile
## these can be highly influential on the fits
our.lass.neph.dat <- subset(our.lass.neph.dat, Carapace.length < quantile(Carapace.length, 0.99) &
                           Carapace.length > quantile(Carapace.length, 0.01))
```

Prepare the data for a multinomial fit.

```
## 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("Mesh.Size", "Carapace.length", "fHAUL", "Count")

## melt the data frame
our.lass.neph.melt <- melt(our.lass.neph.dat[, vars2keep],
                          id = c("Mesh.Size", "Carapace.length", "fHAUL"))

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

## show the first few rows
head(our.lass.neph.cast, 2)

##      Carapace.length fHAUL c100mm_Count c70mm_Count c80mm_Count c90mm_Count
## 10                22    H1             1           1           0           1
## 23                23    H1             1           4           2           1

## format the subsampling ratio similarly

## unique raising factors per haul
rf.count <- with(our.lass.neph.dat, table(fHAUL, Overall.raising.factor, Mesh.Size))

apply(rf.count, 1, FUN = function(x){sum(x>0)})

##   H1  H2  H3  H4  H5  H6  H7  H8  H9 H10 H11 H12 H14
##   4   4   4   4   4   4   4   4   4   4   4   4   4

## convert to sub-sampling ratio as in Celtic Warrior
our.lass.neph.dat$SUBSRATIO <- 1/our.lass.neph.dat$Overall.raising.factor
vars2keep <- c("Mesh.Size", "fHAUL", "SUBSRATIO")
subs.melt <- melt(unique(our.lass.neph.dat[, vars2keep]), id = c("Mesh.Size", "fHAUL"))
subs.cast <- cast(subs.melt, fHAUL ~ Mesh.Size + variable)

## get net position of each
vars2keep <- c("Mesh.Size", "fHAUL", "Net.position")
netpos.melt <- melt(unique(our.lass.neph.dat[, vars2keep]), id = c("Mesh.Size", "fHAUL"))
netpos.cast <- cast(netpos.melt, fHAUL ~ Mesh.Size + variable)

## merge counts and subsampling ratio back together
our.lass.neph.cast0 <- merge(our.lass.neph.cast, subs.cast, by = "fHAUL", all.x = TRUE)

our.lass.neph.cast <- merge(our.lass.neph.cast0, netpos.cast, by = "fHAUL", all.x = TRUE)

## show first few lines
head(our.lass.neph.cast, 2)

##      fHAUL Carapace.length c100mm_Count c70mm_Count c80mm_Count c90mm_Count
## 1      H1                22             1           1           0           1

```

```
## 2      H1          23          1          4          2          1
##      c100mm_SUBSRATIO c70mm_SUBSRATIO c80mm_SUBSRATIO c90mm_SUBSRATIO
## 1          0.02904138      0.03969299      0.03840203      0.06179486
## 2          0.02904138      0.03969299      0.03840203      0.06179486
##      c100mm_Net.position c70mm_Net.position c80mm_Net.position
## 1          4          1          3
## 2          4          1          3
##      c90mm_Net.position
## 1          2
## 2          2

## Extract the matrix of counts
count.vars <- c("c100mm_Count", "c70mm_Count", "c80mm_Count", "c90mm_Count")

neph.count.mat <- as.matrix(our.lass.neph.cast[, count.vars])

## Extract the matrix of subsampling ratios
subsratio.vars <- c("c100mm_SUBSRATIO", "c70mm_SUBSRATIO", "c80mm_SUBSRATIO", "c90mm_SUBSRATIO")

subsratio.mat <- as.matrix(our.lass.neph.cast[, subsratio.vars])

## Create the offset (NEED TO CHECK THIS)
offset.mat <- log(apply(subsratio.mat, 2, FUN =
                        function(zz){zz/subsratio.mat[,1]}))
```

Plot the data

```
library(ggplot2)

## Get the proportions
count.mesh <- as.matrix(our.lass.neph.cast[, count.vars])

prop.mesh <- prop.table(count.mesh, margin = 1)

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

## make a dataframe of the proportions for ggplot
netpos.vars <- c("c100mm_Net.position", "c70mm_Net.position", "c80mm_Net.position", "c90mm_Net.position")

prop.mesh.df <- data.frame(
  Mesh.Size = factor(rep(count.vars, each = m)),
  Carapace.length = rep(our.lass.neph.cast$Carapace.length, times = 4),
  fHAUL = rep(our.lass.neph.cast$fHAUL, times = 4),
  Net.position = unlist(our.lass.neph.cast[, netpos.vars]),
  proportion = c(prop.mesh),
  count = c(count.mesh))

ggplot(prop.mesh.df, aes(x = Carapace.length, y = proportion)) +
  geom_point(colour = "#F8766D", alpha = 0.2, aes(size = log(count))) +
  facet_wrap(~ Mesh.Size) + ylab("Proportion of Nephrops per cod-end")
```

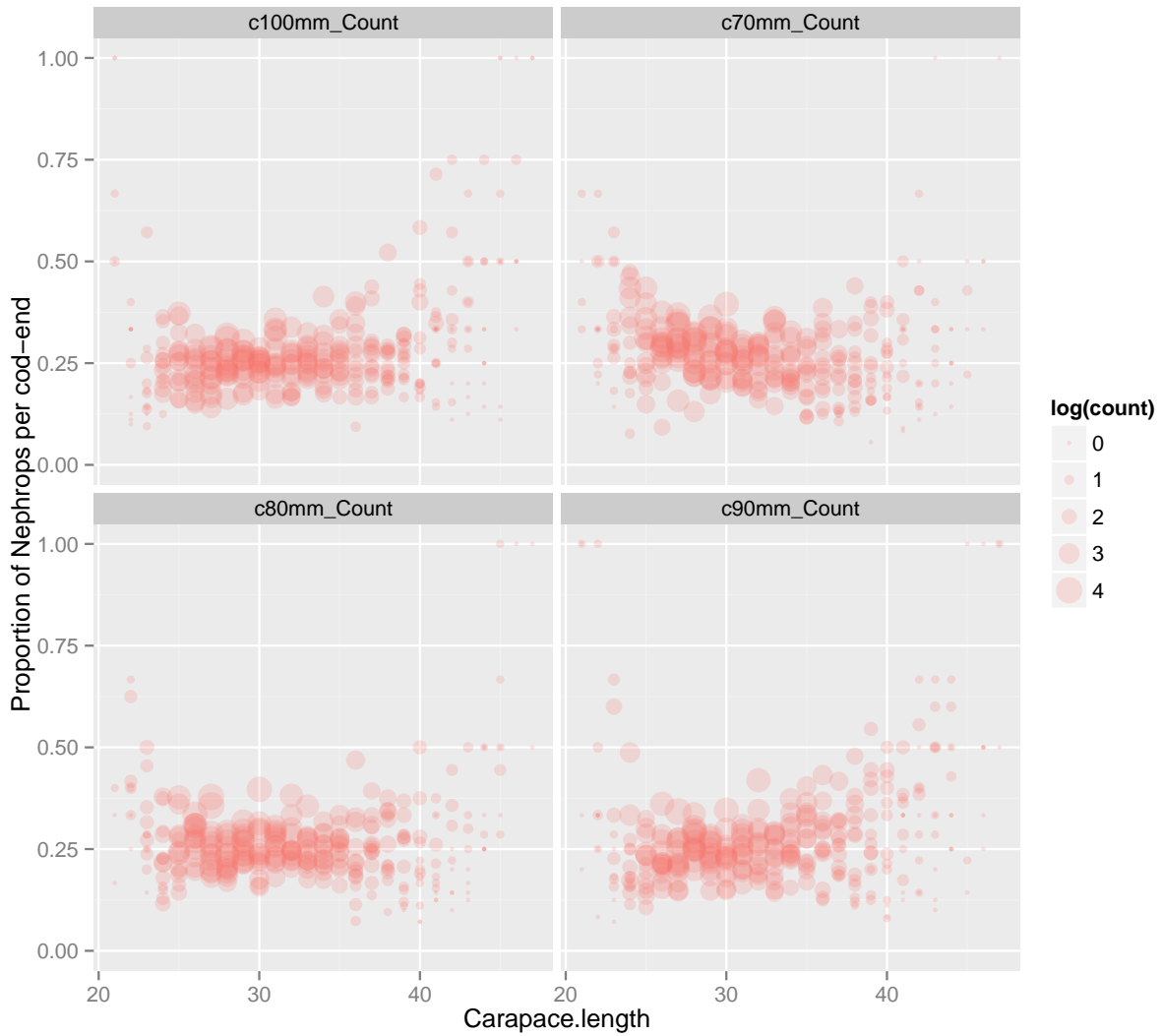


Figure 1: Proportion of Nephrops catch retained per haul. Each point represents the proportion of the Nephrops catch (in number) per haul and length class retained in a given cod-end (70mm, 80mm, 90mm, or 100mm). The size of the point is proportional to the log of the count.

```
##scale_colour_gradientn(colours=rainbow(4))
```

2 Model

The model we focus on is the multinomial, which is a generalization of the binomial to cases with more than two categories (here 4 categories: 70, 80, 90 and 100mm). Under the assumption that each net fishes the same, we would expect 25% of the catch to be retained in each net. We can test that hypothesis.

```
library(nnet)

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

mnom0 <- multinom(neph.count.mat ~ 1)

## # weights:  8 (3 variable)
## initial  value 23374.309223
## final    value 23367.514160
## converged

## include carapace length
## first scale it to range between zero and one
max.length <- max(our.lass.neph.cast$Carapace.length)
our.lass.neph.cast$prop.Carapace.length <- our.lass.neph.cast$Carapace.length/max.length

## Extend to third order polynomial (based on AIC and BIC)
our.lass.neph.cast$prop.Carapace.length2 <- our.lass.neph.cast$prop.Carapace.length^2
our.lass.neph.cast$prop.Carapace.length3 <- our.lass.neph.cast$prop.Carapace.length^3

##
mnom.length <- multinom(neph.count.mat ~
                        prop.Carapace.length + prop.Carapace.length2 + prop.Carapace.length3)

## # weights:  20 (12 variable)
## initial  value 23374.309223
## iter   10 value 23336.622678
## iter   20 value 23336.444786
## final   value 23336.392881
## converged

AIC(mnom0, mnom.length)

##           df          AIC
## mnom0         3 46741.03
## mnom.length   12 46696.79
```

```

## try a VGAM fit for comparison
library(VGAM)

df.vec <- seq(1,10)
aic.vec <- rep(NA, 10)

for(i in 1:10){
  mnom.length.vgam <- vgam(neph.count.mat ~ s(prop.Carapace.length, df = df.vec[i]), family = r
  aic.vec[i] <- AIC(mnom.length.vgam)
}

df.best <- df.vec[which.min(aic.vec)]
mnom.length.vgam <- vgam(neph.count.mat ~ s(prop.Carapace.length, df = df.best), family = r

pred.prop.length <- seq(min(our.lass.neph.cast$prop.Carapace.length),
                        max(our.lass.neph.cast$prop.Carapace.length), length = 100)

pred.length <- seq(min(our.lass.neph.cast$Carapace.length),
                  max(our.lass.neph.cast$Carapace.length), length = 100)

pred.vgam <- predict(mnom.length.vgam, newdata = data.frame(prop.Carapace.length = pred.pro

m <- length(pred.length)

pred.vgam.df <- data.frame(
  Mesh.Size = factor(rep(count.vars, each = m)),
  Carapace.length = rep(pred.length, times = 4),
  proportion = c(pred.vgam) )

```

Get predictions for the fitted model (note this is long-winded here but will be better coded for more than the preliminary example).

```

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

## fit coefficients
beta.mu <- c(t(coef(mnom.length)))

## fit coefficient variance covariance matrix
Sigma <- vcov(mnom.length)

## number of lengths to predict for
nlength <- 100
pred.prop.length <- seq(min(our.lass.neph.cast$prop.Carapace.length),
                        max(our.lass.neph.cast$prop.Carapace.length), length = 100)

pred.length <- seq(min(our.lass.neph.cast$Carapace.length),
                  max(our.lass.neph.cast$Carapace.length), length = 100)

## model matrix

```

```

X <- cbind(1, pred.prop.length, pred.prop.length^2, pred.prop.length^3)

## number of times to resample predictions to get CIs
nresamp <- 100
pred.array <- array(NA, dim = c(nlength, 4, nresamp))

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

for(i in 1:nresamp){
  ## print(i)
  beta <- matrix(rmvnorm(1, mean = beta.mu, sigma = Sigma),
                 nrow = 3, byrow = TRUE)
  p70mmPort <- exp(X %*% matrix(beta[1,])) / (1 + rowSums(exp(X %*% t(beta))))
  p100mmStarboard <- exp(X %*% matrix(beta[2,])) / (1 + rowSums(exp(X %*% t(beta))))
  p70mmStarboard <- exp(X %*% matrix(beta[3,])) / (1 + rowSums(exp(X %*% t(beta))))
  p100mmPort <- 1 - p70mmPort - p100mmStarboard - p70mmStarboard
  pred.p <- cbind(p100mmPort, p70mmPort, p100mmStarboard, p70mmStarboard)
  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(
  Mesh.Size = factor(rep(count.vars, 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 <- ggplot(prop.mesh.df, aes(x = Carapace.length, y = proportion)) +
  geom_point(colour = "#F8766D", alpha = 0.2, aes(size = log(count))) +
  facet_wrap(~ Mesh.Size) + ylab("Proportion of Nephrops per cod-end")

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") +
geom_line(data = pred.vgam.df, aes(x = Carapace.length, y = proportion),
          col = "green", size = 0.5)
```

2.1 Including net position as a covariate

```
## create factor variables
our.lass.neph.cast$f100mm_Net.position <- paste("pos", our.lass.neph.cast$c100mm_Net.position)
our.lass.neph.cast$f90mm_Net.position <- paste("pos", our.lass.neph.cast$c90mm_Net.position)
our.lass.neph.cast$f80mm_Net.position <- paste("pos", our.lass.neph.cast$c80mm_Net.position)
our.lass.neph.cast$f70mm_Net.position <- paste("pos", our.lass.neph.cast$c70mm_Net.position)

mnom.length.netpos <- multinom(neph.count.mat ~
                                prop.Carapace.length +
                                prop.Carapace.length2 +
                                prop.Carapace.length3 +
                                f100mm_Net.position +
                                f70mm_Net.position +
                                f80mm_Net.position +
                                f90mm_Net.position,
                                data = our.lass.neph.cast)

## # weights: 68 (48 variable)
## initial value 23374.309223
## iter 10 value 23326.650060
## iter 20 value 23319.754756
## iter 30 value 23318.476676
## final value 23318.304129
## converged

AIC(mnom.length, mnom.length.netpos)

##              df      AIC
## mnom.length    12 46696.79
## mnom.length.netpos 24 46684.61
```

Plot the net position predictions

```
## get predictions by net position

pred.prop.length <- seq(min(our.lass.neph.cast$prop.Carapace.length),
                        max(our.lass.neph.cast$prop.Carapace.length), length = 100)

pred.length <- seq(min(our.lass.neph.cast$Carapace.length),
                   max(our.lass.neph.cast$Carapace.length), length = 100)
```

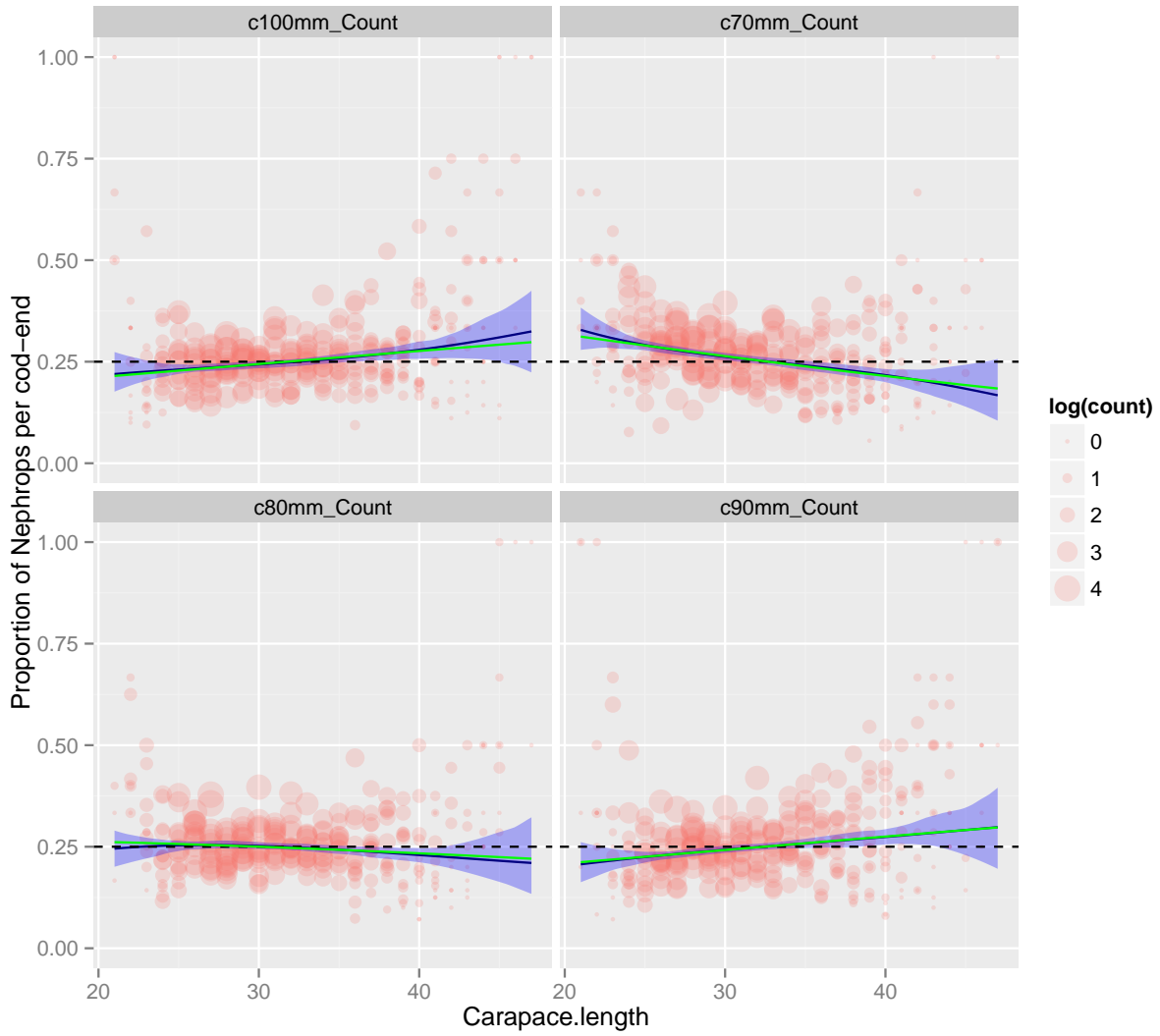



Figure 2: Proportion of Nephrops catch retained per haul with fitted multinomial model and associated re-sampled intervals. Null hypothesis of equal retention is displayed as the dashed line at 0.25.

```

pos.vec <- paste("pos", 1:4, sep = "")

## removing fifth row for now
netpos.unique <- as.data.frame(apply(unique(netpos.cast[,2:5])[-5, ], 2, FUN = function(x) {

names(netpos.unique) <- names(netpos.cast[,2:5])

names(netpos.unique) <- gsub("c", "f", names(netpos.unique))

m <- length(pred.prop.length)

pred.df <- netpos.unique[rep(1:4, each = m), ]
pred.df$prop.Carapace.length <- rep(pred.prop.length, times = 4)
pred.df$prop.Carapace.length2 <- pred.df$prop.Carapace.length^2
pred.df$prop.Carapace.length3 <- pred.df$prop.Carapace.length^3
pred.df$Carapace.length <- rep(pred.length, times = 4)

pred.netpos <- predict(mnom.length.netpos, newdata = pred.df, type = "prob")

m <- dim(pred.df)[1]

pred.netpos.df <- data.frame(
  Mesh.Size = factor(rep(count.vars, each = m)),
  Carapace.length = rep(pred.df$Carapace.length, times = 4),
  Net.position = unlist(pred.df[, 1:4]),
  proportion = c(pred.netpos))

ggplot(prop.mesh.df2, aes(x = Carapace.length, y = proportion, colour = Net.position)) +
  geom_point(alpha = 0.2, aes(size = log(count))) +
  facet_wrap(~ Mesh.Size) + ylab("Proportion of Nephrops per cod-end") +
  ##scale_colour_manual(values=rainbow(4)) +
  geom_line(data = pred.netpos.df, aes(x = Carapace.length, y = proportion))

```

