Preliminary multinomial analysis of *Arkh Angell* data

For discussion

1 Data

```
library(gdata)

arkh.neph.dat <- read.xls("../data/2015 BIM Nephrops quad rig trials/Arkh Angell 70 V 100m
                     sheet = "Nephrops length",
                     stringsAsFactors = FALSE)

## remove Haul 11 and 12 currently as these compare 80mm
arkh.neph.dat <- subset(arkh.neph.dat, !Haul %in% c(11, 12))

## replace spaces in Compartment names
arkh.neph.dat$Compartment <- gsub(" ", "", arkh.neph.dat$Compartment)

## remove any redundant levels
arkh.neph.dat <- droplevels(arkh.neph.dat)

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

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

## make some factor variables used in the analyses
arkh.neph.dat$fHAUL <- factor(arkh.neph.dat$HAUL, levels = unique(arkh.neph.dat$HAUL))
arkh.neph.dat$Compartment <- factor(paste("c", arkh.neph.dat$Compartment, sep = ""))

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

Prepare the data for a multinomial fit.

```r
## 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
arkh.neph.melt <- melt(arkh.neph.dat[, vars2keep],
                  id = c("Compartment", "Carapace.length", "fHAUL"))

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

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

## format the subsampling ratio similarly

## multiple raising factors per haul
rf.count <- with(arkh.neph.dat, table(fHAUL, Raising.factor, Compartment))

apply(rf.count, 1, FUN = function(x){sum(x>0)})
## hauls 4 and 9
## rf.count["H4",,] ## 6.8(28obs) and 6.9(2obs) for 100mm Port
## rf.count["H9",,] ## 15.4(27obs) and 15.5(2obs) for 100mm Starboard
## re-set
arkh.neph.dat$Raising.factor[arkh.neph.dat$fHAUL == "H4" &
                             arkh.neph.dat$Compartment == "c100mmPort"] <- 6.8
arkh.neph.dat$Raising.factor[arkh.neph.dat$fHAUL == "H9" &
                             arkh.neph.dat$Compartment == "c100mmStarboard"] <- 15.4

## convert to sub-sampling ratio as in Celtic Warrior
arkh.neph.dat$SUBSRATIO <- 1/arkh.neph.dat$Raising.factor

vars2keep <- c("Compartment", "fHAUL", "SUBSRATIO")

subs.melt <- melt(unique(arkh.neph.dat[, vars2keep]), id = c("Compartment", "fHAUL"))

subs.cast <- cast(subs.melt, fHAUL  ~ Compartment + variable)

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

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

## Extract the matrix of counts
```

```r
count.vars <- c("c100mmPort_Count", "c70mmPort_Count",
                "c100mmStarboard_Count", "c70mmStarboard_Count")

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

colnames(neph.count.mat) <- c("c100mmPort_Count", "c70mmPort_Count",
                              "c100mmStarboard_Count", "c70mmStarboard_Count")

## Extract the matrix of subsampling ratios
subsratio.vars <- c("c100mmPort_SUBSRATIO", "c70mmPort_SUBSRATIO",
                    "c100mmStarboard_SUBSRATIO", "c70mmStarboard_SUBSRATIO")

subsratio.mat <- as.matrix(arkh.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

```r
library(ggplot2)

## Get the proportions
count.mesh <- as.matrix(arkh.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
prop.mesh.df <- data.frame(
                  Mesh.Size = factor(rep(
                    c("100mmPort", "70mmPort", "100mmStarboard", "70mmStarboard"),
                    each = m),
                    levels = c("100mmPort", "70mmPort", "100mmStarboard", "70mmStarboard"))
                  Carapace.length = rep(arkh.neph.cast$Carapace.length, times = 4),
                  fHAUL = rep(arkh.neph.cast$fHAUL, times = 4),
                  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")
```

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: 100mm Port, 70mm Port, 100mm Starboard, 70mm

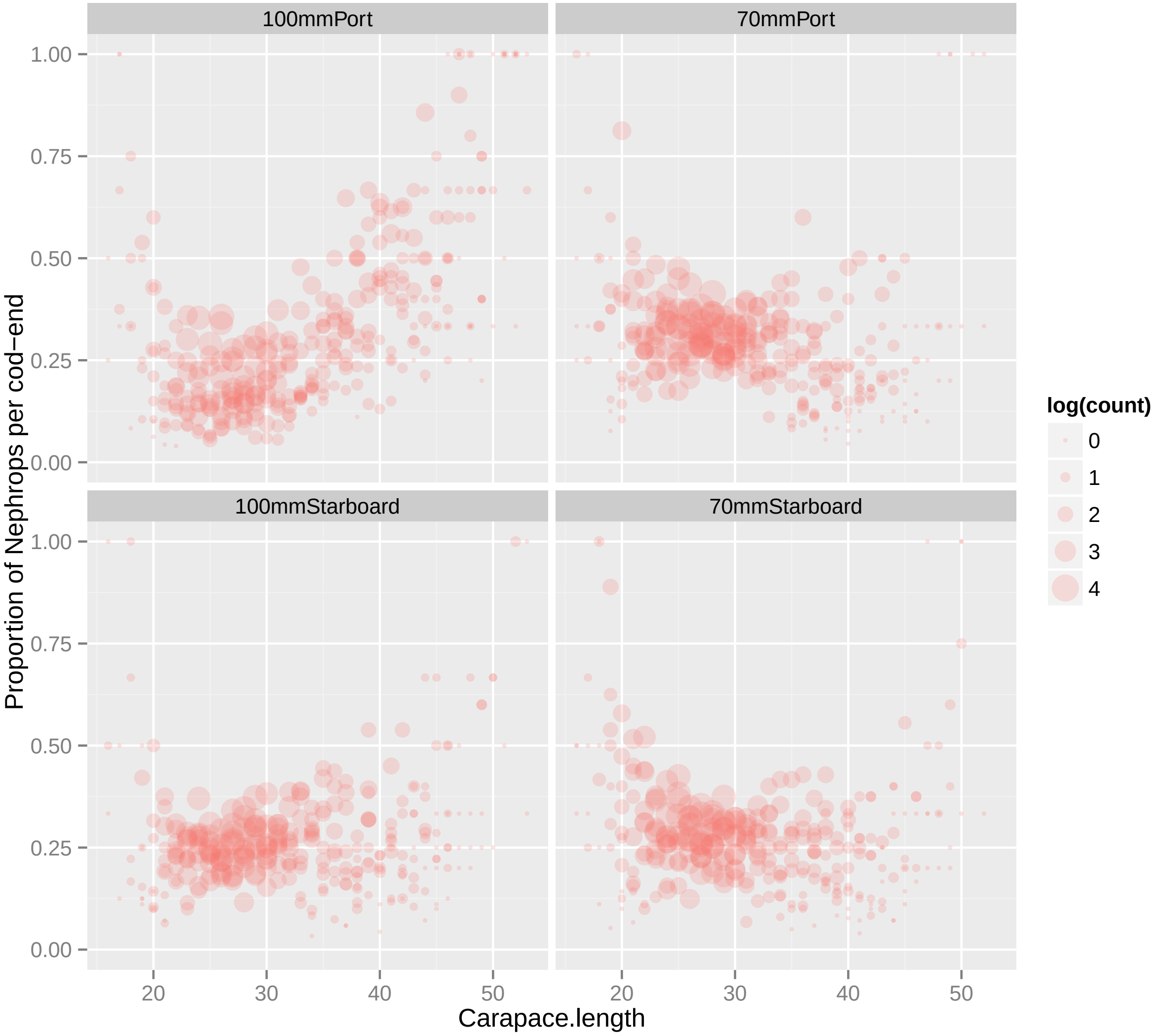

Figure 1: Arkh Angell quad-rig trial. Proportion of Nephrops catch retained per haul. Each point represents the proportion of the Nephrops catch (in number) per haul and length class retained in a given cod-end (70mm, 80mm, 90mm, or 100mm). The size of the point is proportional to the log of the count.

Starboard). 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.

```r
library(nnet)

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

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

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

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

##
mnom.length <- multinom(neph.count.mat ~
                        prop.Carapace.length + prop.Carapace.length2 +
                        prop.Carapace.length3, data = arkh.neph.cast)

AIC(mnom0, mnom.length)

## try a VGAM fit for comparison
## particularly for 100mm Port small lengths
library(VGAM)

## find out best degrees of freedom
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 = multinomial, data = arkh.neph.cast)
  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 = multinomial, data = arkh.neph.cast)

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

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

```
pred.vgam <- predict(mnom.length.vgam,
                     newdata = data.frame(prop.Carapace.length = pred.prop.length),
                     type = "response")

pred.vgam.df <- data.frame(
                  Mesh.Size =
                  factor(rep(c("100mmPort", "70mmPort",
                               "100mmStarboard", "70mmStarboard"),
                    each = m),
                         levels = c("100mmPort", "70mmPort",
                           "100mmStarboard", "70mmStarboard")),
                  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(arkh.neph.cast$prop.Carapace.length),
                        max(arkh.neph.cast$prop.Carapace.length), length = 100)

pred.length <- seq(min(arkh.neph.cast$Carapace.length),
                        max(arkh.neph.cast$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(c("100mmPort", "70mmPort",
                  "100mmStarboard", "70mmStarboard"),
                 each = m), levels = c("100mmPort", "70mmPort",
                              "100mmStarboard", "70mmStarboard")),
                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)
```

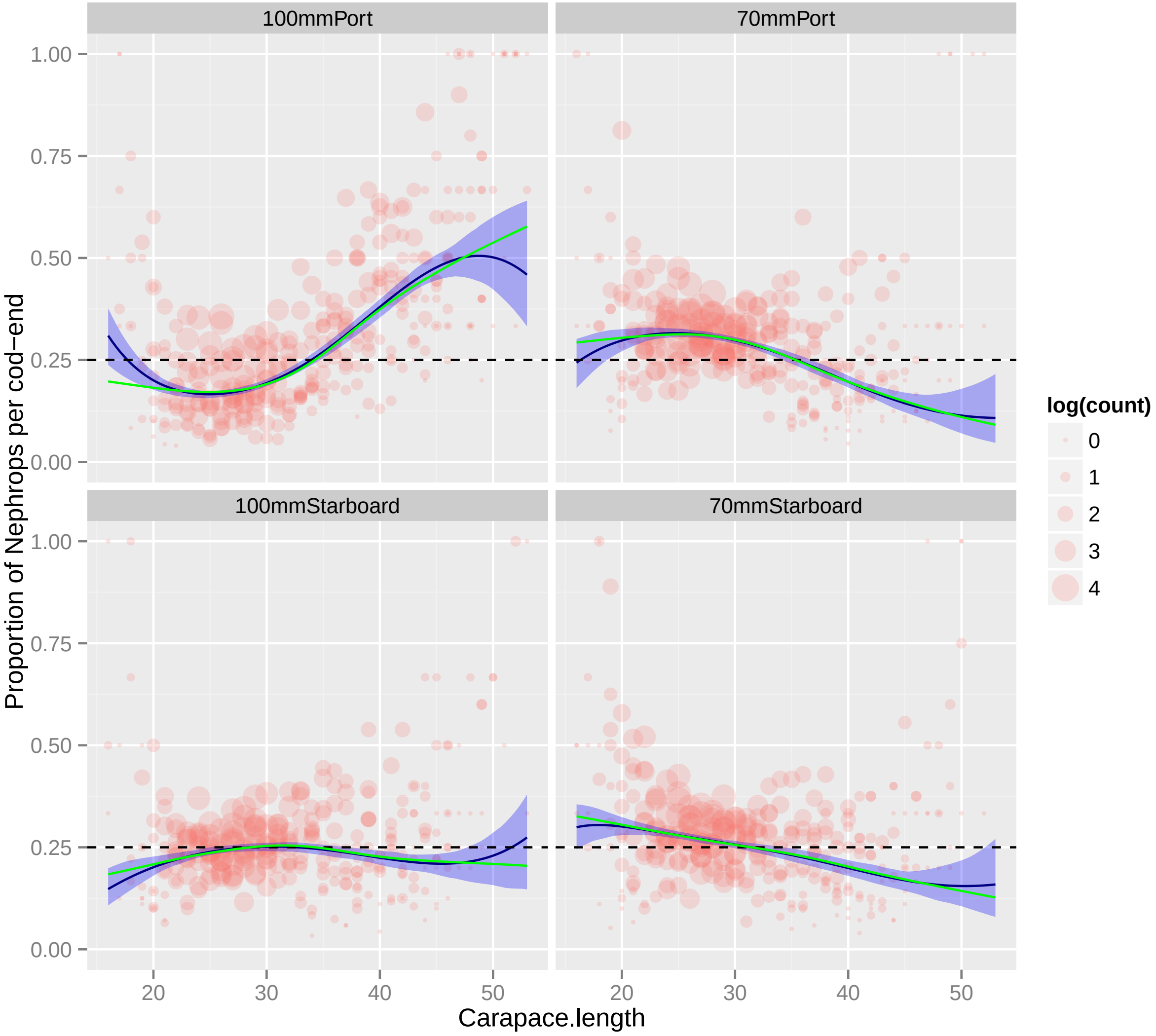

Figure 2: Arkh Angell quad-rig trials. Proportion of Nephrops catch retained per haul with fitted multinomial model and associated re-sampled intervals (blue). Null hypothesis of equal retention is displayed as the dashed line at 0.25. A multinomial additive model is shown in green for comparison.