A preliminary model for Quad-Rig catch comparison

For discussion only

Methods for twin-rig catch comparison analysis are set out in Holst and Revill (2009). Here, this model is preliminarily extended to greater than 2 cod-ends, in particular we focus on the quad-rig with 4 cod-ends. All treatment of the data is included as in a tutorial, which can be used as a basis for capacity building in the analysis of gear technology trials.

1 Data

The data used for this example come from the July 2014 diamond cod-end mesh size trials conducted by BIM aboard MFV Celtic Warrior II on the Smalls grounds. The data are read into R and processed as follows:

```
library(gdata)

neph.dat <- read.xls("../data/Celtic Warrior Diamond mesh July 2014 Celtic Sea.xls",
                     sheet = "Nephrops Lengths",
                     stringsAsFactors = FALSE)

## remove Haul 22, as no recordings for 90mm
neph.dat <- subset(neph.dat, HAUL != 22)

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


##           Vessel       DATE HAUL COMPARTMENT Mesh.Size  SPECIES
## 1 Celtic Warrior 2014-07-19    1     Control      70mm Nephrops
## 2 Celtic Warrior 2014-07-19    1     Control      70mm Nephrops
##   Carapace.Length..mm.. COUNT SUBSRATIO
## 1                    16     1         1
## 2                    17    11         1


## Change the carapace length name
names(neph.dat)[names(neph.dat) == "Carapace.Length..mm.."] <- "Carapace.Length"

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

```
## make some factor variables used in the analyses
neph.dat$fHAUL <- factor(neph.dat$HAUL, levels = unique(neph.dat$HAUL))
neph.dat$fMesh.Size <- factor(neph.dat$Mesh.Size, levels = unique(neph.dat$Mesh.Size))

## remove observations above 99th and below 1th length percentile
## these can be highly influential on the fits
neph.dat <- subset(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("fMesh.Size", "Carapace.Length", "fHAUL", "COUNT")

## melt the data frame
neph.melt <- melt(neph.dat[, vars2keep],
                  id = c("fMesh.Size", "Carapace.Length", "fHAUL"))

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

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


##    Carapace.Length fHAUL 70mm_COUNT 80mm_COUNT 90mm_COUNT 100mm_COUNT
## 1               15    H1          0          2          1           0
## 24              16    H1          1          3          9           1


## format the subsampling ratio similarly
vars2keep <- c("fMesh.Size", "fHAUL", "SUBSRATIO")

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

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

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

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


##   fHAUL Carapace.Length 70mm_COUNT 80mm_COUNT 90mm_COUNT 100mm_COUNT
## 1    H1              15          0          2          1           0
## 2    H1              16          1          3          9           1
##   70mm_SUBSRATIO 80mm_SUBSRATIO 90mm_SUBSRATIO 100mm_SUBSRATIO
## 1              1              1              1               1
## 2              1              1              1               1
```

```
## Extract the matrix of counts
count.vars <- c("70mm_COUNT", "80mm_COUNT", "90mm_COUNT", "100mm_COUNT")

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

colnames(neph.count.mat) <- c("70mm_COUNT", "80mm_COUNT", "90mm_COUNT",
                              "100mm_COUNT")

## Extract the matrix of subsampling ratios
subsratio.vars <- c("70mm_SUBSRATIO", "80mm_SUBSRATIO", "90mm_SUBSRATIO",
                    "100mm_SUBSRATIO")

subsratio.mat <- as.matrix(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(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("70mm", "80mm", "90mm", "100mm"),
                    each = m), levels = c("70mm", "80mm", "90mm", "100mm")),
                  Carapace.Length = rep(neph.cast$Carapace.Length, 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")
```

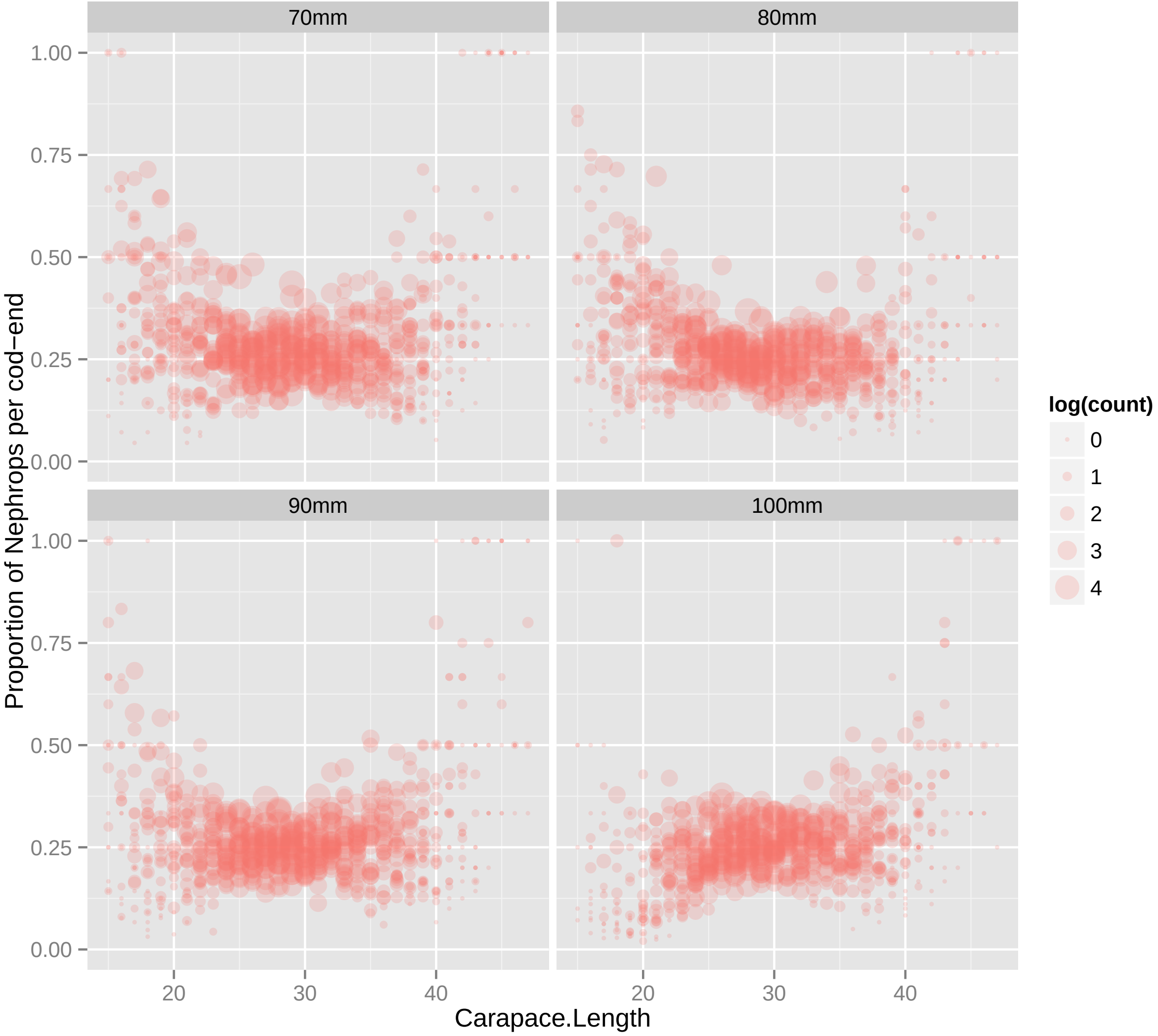

Figure 1: Proportion of Nephrops catch retained per haul. Each point represents the proportion of the Nephrops catch (in number) per haul and length class retained in a given cod-end (70mm, 80mm, 90mm, or 100mm). The size of the point is proportional to the log of the count.

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: 70mm, 80mm, 90mm, 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 + offset(offset.mat))


## # weights:  24 (3 variable)
## initial  value 65833.303791
## final  value 65796.538161
## converged


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

## Extend to third order polynomial (based on AIC and BIC)
neph.cast$Carapace.Length2 <- neph.cast$Carapace.Length^2
neph.cast$Carapace.Length3 <- neph.cast$Carapace.Length^3

##
mnom.length <- multinom(neph.count.mat ~
                        Carapace.Length + Carapace.Length2 + Carapace.Length3 +
                        offset(offset.mat), data = neph.cast)


## # weights:  36 (12 variable)
## initial  value 65833.303791
## iter  10 value 65574.013563
## iter  20 value 65540.974075
## final  value 65538.628411
## converged


AIC(mnom0, mnom.length)


##             df      AIC
## mnom0        3 131599.1
## mnom.length 12 131101.3
```

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

## model matrix
X <- cbind(1, pred.length, pred.length^2, pred.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)
  p80 <- exp(X %*% matrix(beta[1,]))/(1 + rowSums(exp(X %*% t(beta))))
  p90 <- exp(X %*% matrix(beta[2,]))/(1 + rowSums(exp(X %*% t(beta))))
  p100 <- exp(X %*% matrix(beta[3,]))/(1 + rowSums(exp(X %*% t(beta))))
  p70 <- 1 - p80 - p90 - p100
  pred.p <- cbind(p70, p80, p90, p100)
  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("70mm", "80mm", "90mm", "100mm"),
                 each = m), levels = c("70mm", "80mm", "90mm", "100mm")),
               Carapace.Length = rep(pred.length * max.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")
```

2.1 Including weight as a covariate

Make a row per observation and merge with the weight data

```
## get a row per length measurement (raise them also)
n <- nrow(neph.dat)

##neph.dat2 <- neph.dat[rep(1:n, times =
##                          round(neph.dat$COUNT/neph.dat$SUBSRATIO, 0)), ]

## Note: no raising here
neph.dat2 <- neph.dat[rep(1:n, times = neph.dat$COUNT), ]

weight.dat <- read.xls("../data/Celtic Warrior Diamond mesh July 2014 Celtic Sea.xls",
                       sheet = "Weights",
                       stringsAsFactors = FALSE)

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


##          Date Haul.. Compartment Mesh.Size Species Total.weight..kg.
## 1 2014-07-19      1       TEST1      90mm    Bulk             26.28
## 2 2014-07-19      1       TEST1      90mm Haddock              0.38
##   Sbsample.weight..kg.
## 1
## 2
```

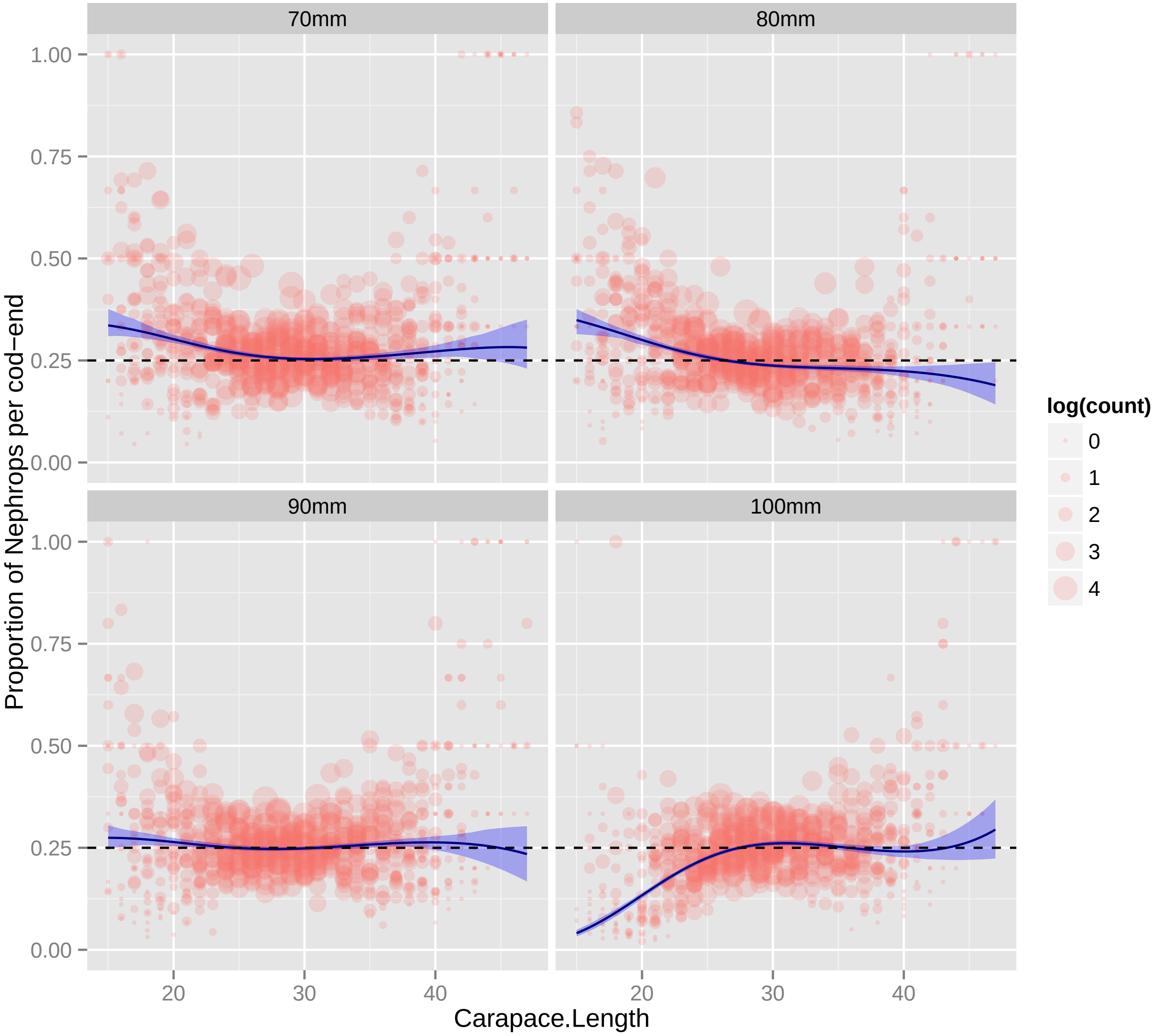

Figure 2: Proportion of Nephrops catch retained per haul with fitted multinomial model and associated re-sampled intervals. Null hypothesis of equal retention is displayed as the dashed line at 0.25.

```
## create a new "HAUL" variable for the merge
weight.dat$HAUL <- paste("H", weight.dat$Haul.., sep ="")

## re-name total weight column
names(weight.dat)[names(weight.dat) == "Total.weight..kg."] <- "Total.Weight"

## merge nephrops length and total bulk weight data
neph.dat3 <- merge(neph.dat2,
                   subset(weight.dat, Species == "Bulk")[, c("Mesh.Size", "HAUL",
                                                          "Total.Weight")],
                   by = c("Mesh.Size", "HAUL"))

max.length <- max(neph.dat3$Carapace.Length)
neph.dat3$Carapace.Length <- neph.dat3$Carapace.Length/max.length
neph.dat3$Carapace.Length2 <- neph.dat3$Carapace.Length^2
neph.dat3$Carapace.Length3 <- neph.dat3$Carapace.Length^3
```

Include weight in the fit

```
## compare two ways of writing same model
## in matrix format, as before except without offset for the moment
mnom.length.matrix <- multinom(neph.count.mat ~
                               Carapace.Length +
                               Carapace.Length2 +
                               Carapace.Length3, data = neph.cast)


## # weights:  20 (12 variable)
## initial  value 65846.209564
## iter  10 value 65594.036234
## iter  20 value 65560.638668
## final  value 65558.314009
## converged


## in long format (row per measurement)
mnom.length.long <- multinom(fMesh.Size ~
                             Carapace.Length +
                             Carapace.Length2 +
                             Carapace.Length3, data = neph.dat3)


## # weights:  20 (12 variable)
## initial  value 65846.209564
## iter  10 value 65594.036234
## iter  20 value 65560.638668
## final  value 65558.314009
## converged


AIC(mnom.length.matrix, mnom.length.long)
```

```
##                     df      AIC
## mnom.length.matrix 12 131140.6
## mnom.length.long   12 131140.6

## Same model, now include bulk weight in long format model
## in addition to carapace polynomials

mnom.length.bulk <- multinom(fMesh.Size ~
                             Carapace.Length +
                             Carapace.Length2 +
                             Carapace.Length3 + Total.Weight, data = neph.dat3)

## # weights:  24 (15 variable)
## initial  value 65846.209564
## iter  10 value 65467.670510
## iter  20 value 65392.143368
## iter  30 value 65389.022743
## iter  30 value 65389.022490
## iter  30 value 65389.022490
## final  value 65389.022490
## converged

AIC(mnom.length.long, mnom.length.bulk)

##                  df      AIC
## mnom.length.long 12 131140.6
## mnom.length.bulk 15 130808.0

## interaction between length and bulk
mnom.length.bulk.inter <- multinom(fMesh.Size ~
                                   Carapace.Length +
                                   Carapace.Length2 +
                                   Carapace.Length3 +
                                   Total.Weight +
                                   Carapace.Length:Total.Weight +
                                   Carapace.Length2:Total.Weight +
                                   Carapace.Length3:Total.Weight
                                   , data = neph.dat3)

## # weights:  36 (24 variable)
## initial  value 65846.209564
## iter  10 value 65582.220539
## iter  20 value 65400.493488
## iter  30 value 65364.125707
## iter  40 value 65361.240613
## final  value 65361.182171
## converged

AIC(mnom.length.bulk, mnom.length.bulk.inter)
```

```
##                         df      AIC
## mnom.length.bulk        15 130808.0
## mnom.length.bulk.inter 24 130770.4
```

Get predictions for low, medium and high catch weights

```
low.med.high.bulk <- quantile(weight.dat[weight.dat$Species == "Bulk",]$Total.Weight,
                              p = c(0.1, 0.5, 0.9))

low.med.high.bulk
```

```
##    10%    50%    90% 
## 26.132 60.500 93.280
```

```
## set high to 120kg
low.med.high.bulk[3] <- 120

pred.df <- expand.grid(Carapace.Length = pred.length,
                       Total.Weight = low.med.high.bulk)
pred.df$Carapace.Length2 <- pred.df$Carapace.Length^2
pred.df$Carapace.Length3 <- pred.df$Carapace.Length^3

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

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

mnom.pred.df <- data.frame(
                  Mesh.Size = factor(rep(c("70mm", "80mm", "90mm", "100mm"),
                    each = m), levels = c("70mm", "80mm", "90mm", "100mm")),
                  Carapace.Length = rep(pred.df$Carapace.Length, times = 4),
                  Total.Weight = rep(pred.df$Total.Weight, times = 4),
                  proportion = c(mnom.pred))

mnom.pred.df$Bulk.Weight <-
  ifelse(mnom.pred.df$Total.Weight == low.med.high.bulk[1], "Low (26kg)",
         ifelse(mnom.pred.df$Total.Weight == low.med.high.bulk[2], "Medium (60.5kg)", "High
         )

p + geom_line(data = mnom.pred.df,
              aes(x = Carapace.Length*max.length, y = proportion,
                  group = Bulk.Weight,
                  colour = Bulk.Weight)) +
  geom_hline(aes(yintercept = 0.25), linetype = "dashed") +
  geom_vline(aes(xintercept = 25), linetype = "dashed") +
  scale_colour_brewer(palette="Set1")
```

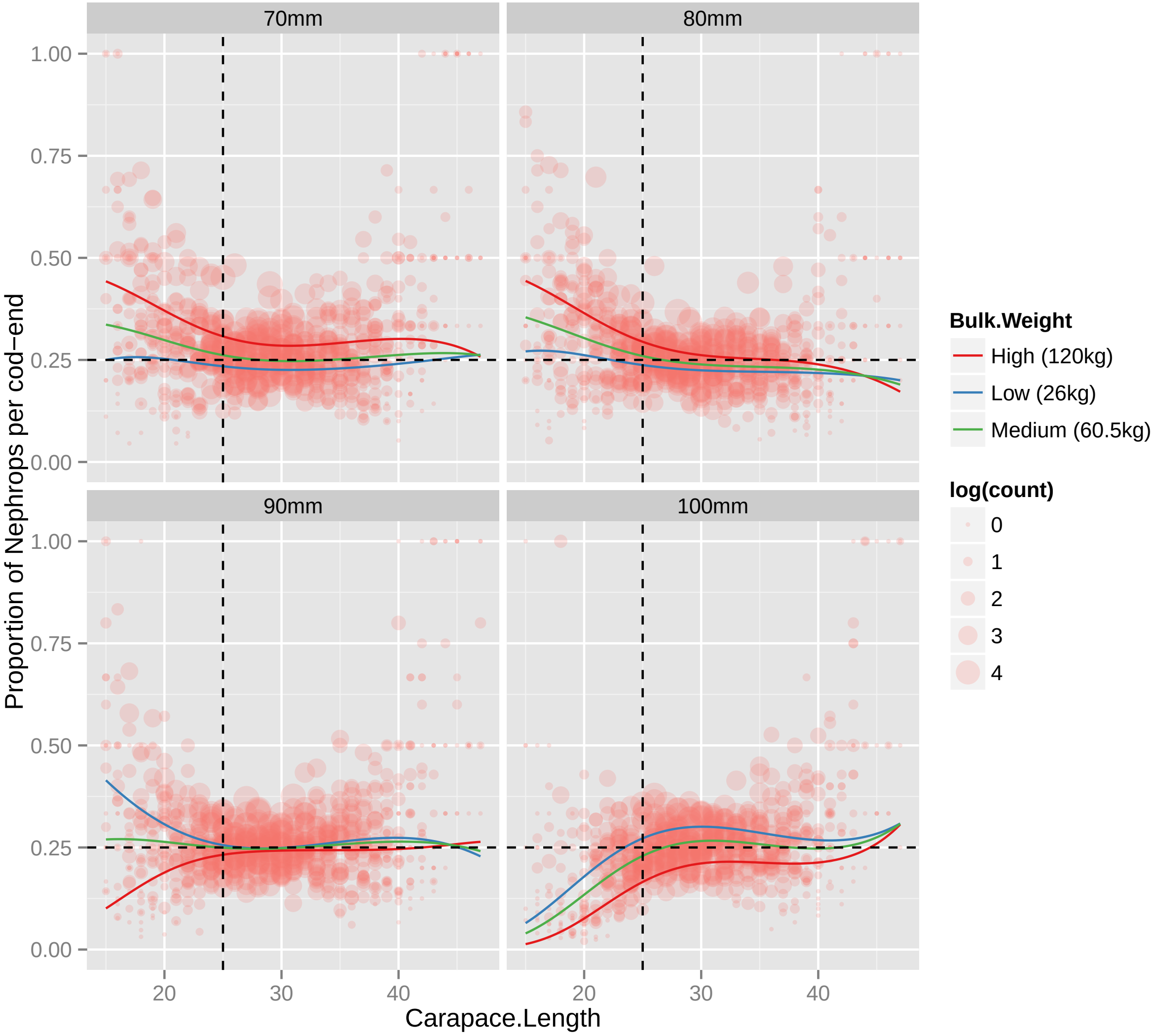

Figure 3: Predicted influence of the bulk catch weight (low: 26kg, medium: 61kg, high: 93kg) on the proportion of Nephrops catch retained per net. Null hypothesis of equal retention is displayed as the dashed line at 0.25. Minimum landing size of 25mm is shown as the vertical dashed line.

References

Holst, R. and Revill, A. 2009. A simple statistical method for catch comparison studies. *Fisheries Research* **95**(2–3): 254 – 259