

# *Our Lass* 70mm versus 80mm analysis

For discussion

## 1 Data

Read the data in and make a few factor variables

```
library(gdata)
library(reshape)

neph.dat <- read.xls("../data/2015 BIM Nephrops quad rig trials/Our Lass 2 70_80_90_100mm d
                      sheet = "All hauls",
                      stringsAsFactors = FALSE)

## subset containing the 70mm and 80mm data
neph.7080 <- subset(neph.dat, Mesh.Size %in% c("70mm", "80mm"))

## make factor haul variable
neph.7080$fHAUL <- factor(paste("H", neph.7080$Haul.No, sep = ""))

## make a factor mesh size variable
neph.7080$fMesh.Size <- factor(paste("mesh", neph.7080$Mesh.Size, sep = ""))

## remove trailing dot from raised count column name
names(neph.7080)[names(neph.7080) == "Raised.count."] <- "Raised.count"
```

Re-shape the data to wide format (columns for 70mm, 80mm variables).

```
## 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("fHAUL", "fMesh.Size", "Net.position", "Carapace.length",
              "Count", "Raised.count", "Total.catch", "Overall.Sampling.Ratio")

## melt the data frame
neph.7080.melt <- melt(neph.7080[, vars2keep],
                      id = c("fHAUL", "fMesh.Size", "Carapace.length"))

## re-form the dataframe in required format
```

```
neph.7080.cast <- cast(neph.7080.melt, Carapace.length + fHAUL ~ fMesh.Size + variable)
```

```
## first couple of lines
```

```
head(neph.7080.cast, 2)
```

```
##      Carapace.length fHAUL mesh70mm_Net.position mesh70mm_Count
## 1              17    H14                      4              1
## 2              17     H3                      NA              NA
##      mesh70mm_Raised.count mesh70mm_Total.catch
## 1              32.1              467.7
## 2              NA              NA
##      mesh70mm_Overall.Sampling.Ratio mesh80mm_Net.position mesh80mm_Count
## 1              0.03112036                      NA              NA
## 2              NA                      3              1
##      mesh80mm_Raised.count mesh80mm_Total.catch
## 1              NA              NA
## 2              32.1              420.05
##      mesh80mm_Overall.Sampling.Ratio
## 1              NA
## 2              0.03118295
```

```
summary(neph.7080.cast) ## note lots of NAs
```

```
##      Carapace.length      fHAUL      mesh70mm_Net.position mesh70mm_Count
## Min.      :17.00    H11      : 31    Min.      :1.000      Min.      : 1.00
## 1st Qu.:27.00    H14      : 28    1st Qu.:2.000      1st Qu.: 3.00
## Median :33.00    H4       : 27    Median :3.000      Median :10.00
## Mean   :33.18    H7       : 27    Mean   :2.657      Mean   :14.76
## 3rd Qu.:40.00    H6       : 26    3rd Qu.:4.000      3rd Qu.:24.00
## Max.   :54.00    H8       : 26    Max.   :4.000      Max.   :66.00
##              (Other):172    NA's    :37      NA's    :37
##      mesh70mm_Raised.count mesh70mm_Total.catch
## Min.      : 10.8      Min.      :203.1
## 1st Qu.: 71.6      1st Qu.:306.0
## Median : 205.8      Median :411.3
## Mean   : 374.9      Mean   :416.3
## 3rd Qu.: 580.4      3rd Qu.:490.1
## Max.   :2251.3      Max.   :618.0
## NA's    :37      NA's    :37
##      mesh70mm_Overall.Sampling.Ratio mesh80mm_Net.position mesh80mm_Count
## Min.      :0.02310      Min.      :1.000      Min.      : 1.00
## 1st Qu.:0.03112      1st Qu.:2.000      1st Qu.: 2.25
## Median :0.04072      Median :2.000      Median :10.00
## Mean   :0.04926      Mean   :2.441      Mean   :13.79
## 3rd Qu.:0.05830      3rd Qu.:3.000      3rd Qu.:21.75
## Max.   :0.09226      Max.   :4.000      Max.   :61.00
## NA's    :37      NA's    :31      NA's    :31
##      mesh80mm_Raised.count mesh80mm_Total.catch
## Min.      : 9.4      Min.      :165.8
## 1st Qu.: 52.5      1st Qu.:264.8
```

```

## Median : 193.4          Median :407.2
## Mean   : 305.8          Mean    :388.1
## 3rd Qu.: 441.5          3rd Qu.:458.8
## Max.    :1621.2         Max.     :635.3
## NA's     :31           NA's      :31
## mesh80mm_Overall.Sampling.Ratio
## Min.     :0.02309
## 1st Qu.:0.03163
## Median   :0.04821
## Mean     :0.05499
## 3rd Qu.:0.06868
## Max.     :0.10591
## NA's     :31

## fill in missing values
## these occur if there is a count for e.g. 20mm CL in 70mm but not in 80mm
neph.7080.cast$mesh70mm_Count[is.na(neph.7080.cast$mesh70mm_Count)] <- 0
neph.7080.cast$mesh70mm_Raised.count[is.na(neph.7080.cast$mesh70mm_Raised.count)] <- 0
neph.7080.cast$mesh80mm_Count[is.na(neph.7080.cast$mesh80mm_Count)] <- 0
neph.7080.cast$mesh80mm_Raised.count[is.na(neph.7080.cast$mesh80mm_Raised.count)] <- 0

for(i in 1:dim(neph.7080.cast)[1]){
  haul.dat <- subset(neph.7080.cast, fHAUL == neph.7080.cast$fHAUL[i])
  ## 70mm net position
  if(is.na(neph.7080.cast$mesh70mm_Net.position[i])){
    neph.7080.cast$mesh70mm_Net.position[i] <-
      unique(na.omit(haul.dat$mesh70mm_Net.position))
  }
  ## 80mm net position
  if(is.na(neph.7080.cast$mesh80mm_Net.position[i])){
    neph.7080.cast$mesh80mm_Net.position[i] <-
      unique(na.omit(haul.dat$mesh80mm_Net.position))
  }
  ## 70mm total catch
  if(is.na(neph.7080.cast$mesh70mm_Total.catch[i])){
    neph.7080.cast$mesh70mm_Total.catch[i] <-
      unique(na.omit(haul.dat$mesh70mm_Total.catch))
  }
  ## 80mm total catch
  if(is.na(neph.7080.cast$mesh80mm_Total.catch[i])){
    neph.7080.cast$mesh80mm_Total.catch[i] <-
      unique(na.omit(haul.dat$mesh80mm_Total.catch))
  }
  ## Sampling ratio
  ## 70mm total catch
  if(is.na(neph.7080.cast$mesh70mm_Overall.Sampling.Ratio[i])){
    neph.7080.cast$mesh70mm_Overall.Sampling.Ratio[i] <-
      unique(na.omit(haul.dat$mesh70mm_Overall.Sampling.Ratio))
  }
  ## 80mm total catch
  if(is.na(neph.7080.cast$mesh80mm_Overall.Sampling.Ratio[i])){

```

```

neph.7080.cast$mesh80mm_Overall.Sampling.Ratio[i] <-
  unique(na.omit(haul.dat$mesh80mm_Overall.Sampling.Ratio))
}
}

summary(neph.7080.cast) ## no missing

## Carapace.length      fHAUL      mesh70mm_Net.position mesh70mm_Count
## Min.      :17.00    H11      : 31    Min.      :1.000      Min.      : 0.00
## 1st Qu.:27.00    H14      : 28    1st Qu.:2.000      1st Qu.: 2.00
## Median :33.00    H4       : 27    Median :3.000      Median : 8.00
## Mean   :33.18    H7       : 27    Mean   :2.656      Mean   :13.14
## 3rd Qu.:40.00    H6       : 26    3rd Qu.:4.000      3rd Qu.:23.00
## Max.   :54.00    H8       : 26    Max.   :4.000      Max.   :66.00
##                (Other):172
## mesh70mm_Raised.count mesh70mm_Total.catch
## Min.      : 0.0      Min.      :203.1
## 1st Qu.: 34.3      1st Qu.:306.0
## Median : 160.7      Median :411.3
## Mean   : 333.7      Mean   :416.9
## 3rd Qu.: 520.6      3rd Qu.:490.1
## Max.   :2251.3      Max.   :618.0
##
## mesh70mm_Overall.Sampling.Ratio mesh80mm_Net.position mesh80mm_Count
## Min.      :0.02310      Min.      :1.000      Min.      : 0.00
## 1st Qu.:0.03112      1st Qu.:2.000      1st Qu.: 2.00
## Median :0.04072      Median :2.000      Median : 7.00
## Mean   :0.04960      Mean   :2.442      Mean   :12.52
## 3rd Qu.:0.05830      3rd Qu.:3.000      3rd Qu.:20.00
## Max.   :0.09226      Max.   :4.000      Max.   :61.00
##
## mesh80mm_Raised.count mesh80mm_Total.catch
## Min.      : 0.0      Min.      :165.8
## 1st Qu.: 32.1      1st Qu.:264.8
## Median : 158.1      Median :407.2
## Mean   : 277.7      Mean   :387.6
## 3rd Qu.: 406.3      3rd Qu.:458.8
## Max.   :1621.2      Max.   :635.3
##
## mesh80mm_Overall.Sampling.Ratio
## Min.      :0.02309
## 1st Qu.:0.03163
## Median :0.04821
## Mean   :0.05452
## 3rd Qu.:0.06868
## Max.   :0.10591
##

```

Get the empirical proportion  $80/(70 + 80)$  at length. Note that the length-specific CIs do not reflect the non-independence of the observations across lengths at the haul level are therefore not plotted.

```
## vector of carapace lengths
cl.vec <- unique(neph.7080.cast$Carapace.length)
cl.vec <- cl.vec[order(cl.vec)]

## including number of observations contributing to count here
count.df <- data.frame(Carapace.length = cl.vec, N = NA, prop.80 = NA)

for(i in 1:dim(count.df)[1]){
  sub.dat <- subset(neph.7080.cast, Carapace.length == count.df$Carapace.length[i])
  ##
  if(dim(sub.dat)[1] > 1){
    ## raised number
    count.df$N[i] <- with(sub.dat,
                          round(sum(mesh80mm_Raised.count)) + round(sum(mesh70mm_Raised.co

    ##
    count.df$prop.80[i] <- with(sub.dat, round(sum(mesh80mm_Raised.count)) /
                                (round(sum(mesh80mm_Raised.count)) + round(sum(mesh70mm_Ra

    rm(list = c("sub.dat"))
  }
}
```

Plot the data (Figure 1)

```
with(count.df, plot(Carapace.length, prop.80, ylim = c(0, 1), pch = 19,
                    xlab = "Carapace length (mm)",
                    ylab = "Proportion (N80mm/(N70mm + N80mm))",
                    bty = "L",
                    cex = 1/5 * log(N)))
abline(h = 0.5, lty = 2)
```

## 2 Models

A catch comparison binomial Generalized Additive/Linear Mixed Model is suitable choice for these count data where we are interested in estimating how the proportion changes with carapace length. We first try a model with only carapace length as an explanatory variable with haul random effects.

```
library(mgcv)

neph.7080.cast$dum <- 1

## no length effect
gamm.null <- gam(cbind(mesh80mm_Count, mesh70mm_Count) ~ 1 +
                 s(fHAUL, bs="re", by = dum),
                 offset =
                 log(mesh80mm_Overall.Sampling.Ratio /
```

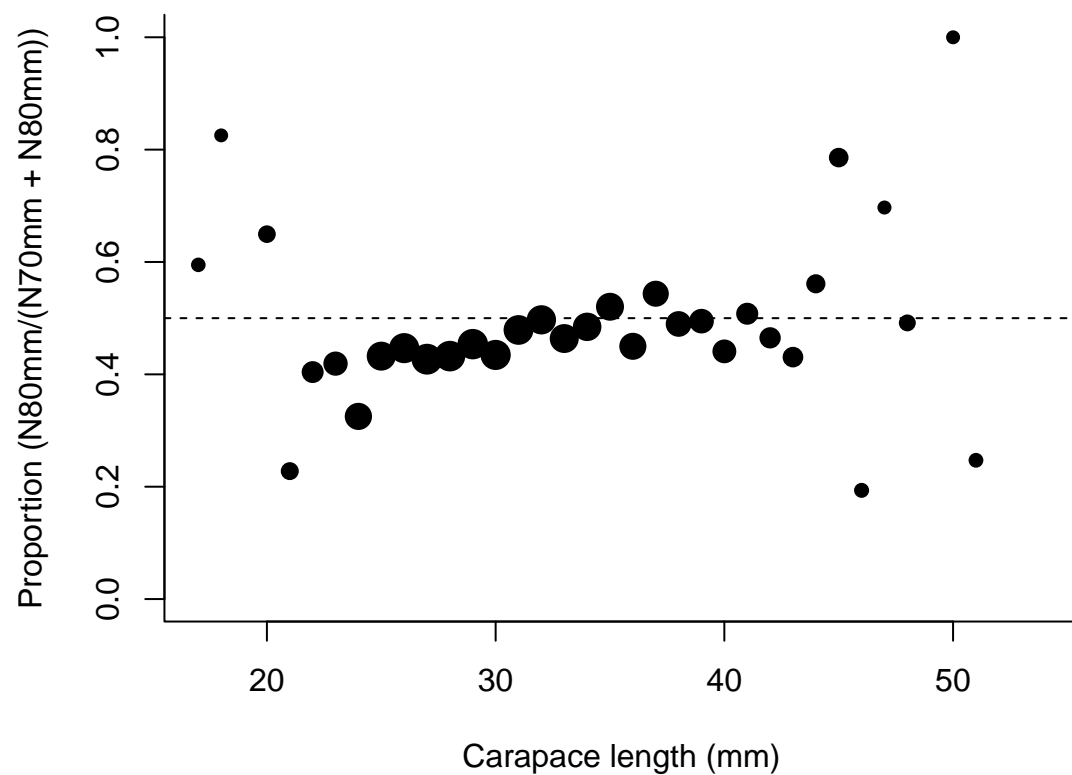


Figure 1: Proportion of Nephrops raised numbers retained in the 80mm over the sum of the 80mm and 70mm meshes.

```

        mesh70mm_Overall.Sampling.Ratio),
        family = binomial,
        data = neph.7080.cast)

gamm.alt <- gam(cbind(mesh80mm_Count, mesh70mm_Count) ~
  s(Carapace.length, k = 5) +
  s(fHAUL, bs="re", by = dum),
  offset =
  log(mesh80mm_Overall.Sampling.Ratio /
    mesh70mm_Overall.Sampling.Ratio),
  family = binomial,
  data = neph.7080.cast)

## likelihood ratio test for the significance of carapace length
anova(gamm.null, gamm.alt, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: cbind(mesh80mm_Count, mesh70mm_Count) ~ 1 + s(fHAUL, bs = "re",
##   by = dum)
## Model 2: cbind(mesh80mm_Count, mesh70mm_Count) ~ s(Carapace.length, k = 5) +
##   s(fHAUL, bs = "re", by = dum)
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1      324.11      438.09
## 2      323.11      429.68 0.99849    8.4149 0.003711 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Plot the predictions from this model.

```

mean.catch <- mean(c(unique(neph.7080.cast$mesh70mm_Total.catch),
  unique(neph.7080.cast$mesh80mm_Total.catch)))

## data frame to predictfor
pred.df <- data.frame(Carapace.length = cl.vec,
  fHAUL = "H1",
  dum = 0,
  mesh80mm_Overall.Sampling.Ratio = 1,
  mesh70mm_Overall.Sampling.Ratio = 1,
  mesh70mm_Total.catch = mean.catch,
  mesh80mm_Total.catch = mean.catch
)

pred.gamm.alt <- predict(gamm.alt, newdata = pred.df, se.fit = TRUE)

## predicted proportions and confidence intervals
pred.df$pred.prop <- plogis(pred.gamm.alt$fit)
pred.df$lwr.prop <- plogis(pred.gamm.alt$fit - qnorm(0.975) * pred.gamm.alt$se.fit)
pred.df$upr.prop <- plogis(pred.gamm.alt$fit + qnorm(0.975) * pred.gamm.alt$se.fit)

```

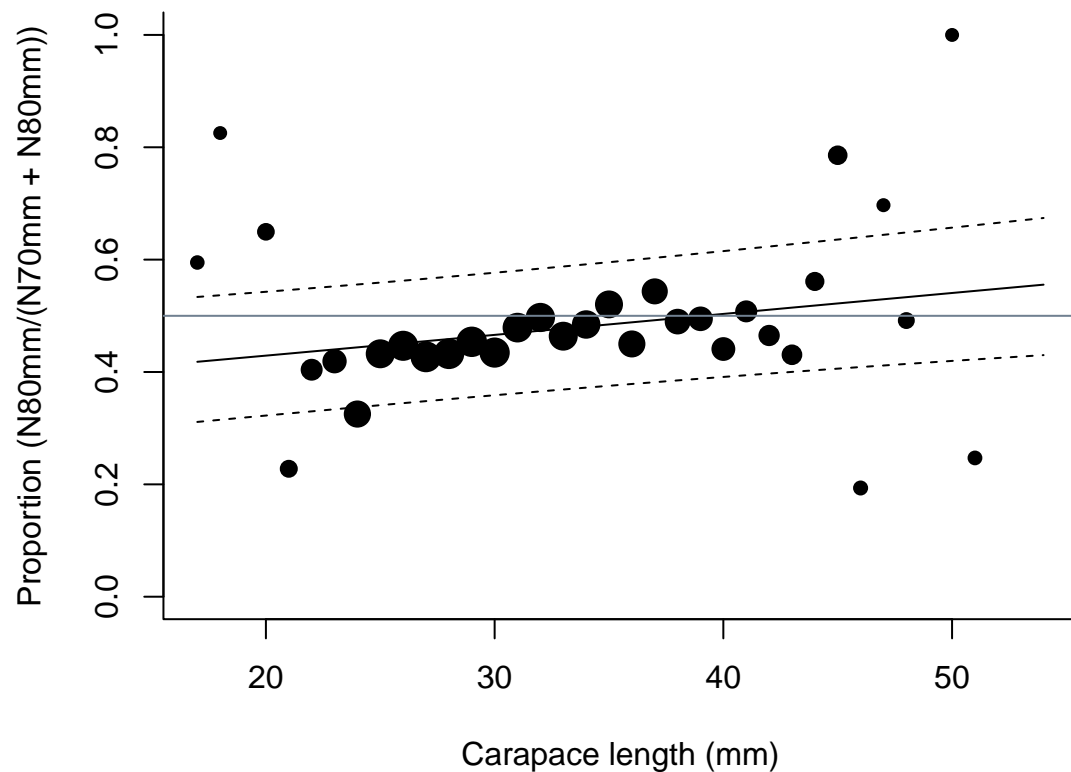


Figure 2: Predicted proportion from binomial GLMM without covariates.

```
with(count.df, plot(Carapace.length, prop.80, ylim = c(0, 1), pch = 19,
  xlab = "Carapace length (mm)",
  ylab = "Proportion (N80mm/(N70mm + N80mm))",
  bty = "L",
  cex = 1/5 * log(N) ))

with(pred.df, lines(Carapace.length, pred.prop))
with(pred.df, lines(Carapace.length, lwr.prop, lty = 2))
with(pred.df, lines(Carapace.length, upr.prop, lty = 2))
abline(h = 0.5, col = "slategrey")
```

The cause of the wide confidence intervals (Figure 2) is the large amount of inter-haul variability in the proportion retained in the 80mm (Figure 3)



```

## Get the proportion at length by haul
haul.count.df <- expand.grid(Carapace.length = cl.vec,
                             fHAUL = unique(neph.7080.cast$fHAUL))

## order the levels
haul.count.df$fHAUL <- factor(as.character(haul.count.df$fHAUL),
                              levels = c(paste("H", 1:12, sep = ""), "H14"))

haul.count.df$prop.80 <- NA
haul.count.df$N <- NA
haul.count.df$lwr <- NA
haul.count.df$upr <- NA

for(i in 1:dim(haul.count.df)[1]){
  sub.dat <- subset(neph.7080.cast,
                    Carapace.length == haul.count.df$Carapace.length[i] &
                    fHAUL == haul.count.df$fHAUL[i])

  ##
  ##if((sub.dat$mesh80mm_Raised.count + sub.dat$mesh70mm_Raised.count) > 0){
  if(dim(sub.dat)[1] > 0){
    btest <- with(sub.dat,
                  binom.test(x = round(mesh80mm_Raised.count),
                             n = round(mesh80mm_Raised.count + mesh70mm_Raised.count)))

    ##
    haul.count.df$N[i] <- with(sub.dat,
                              round(mesh80mm_Raised.count + mesh70mm_Raised.count))
    haul.count.df$prop.80[i] <- btest$estimate
    haul.count.df$lwr[i] <- btest$conf.int[1]
    haul.count.df$upr[i] <- btest$conf.int[2]
    ##
    rm(list = c("sub.dat", "btest"))
  }
}

## get predictions at the HAUL level from model
haul.count.df$dum <- 1
haul.count.df$pred.prop <- plogis(predict(gamm.alt, newdata = haul.count.df))

library(ggplot2)

blue2red <- colorRampPalette(c("darkblue", "white", "red"))

ggplot(haul.count.df, aes(x = Carapace.length, y = prop.80)) +
  geom_point(aes(colour = fHAUL, size = 1/5 * log(N))) +
  geom_line(data = haul.count.df, aes(x = Carapace.length, y = pred.prop, colour = fHAUL)) +
  scale_colour_manual(values = blue2red(13)) + ylab("Proportion in 80mm")

## Warning: Removed 118 rows containing missing values (geom_point).

```

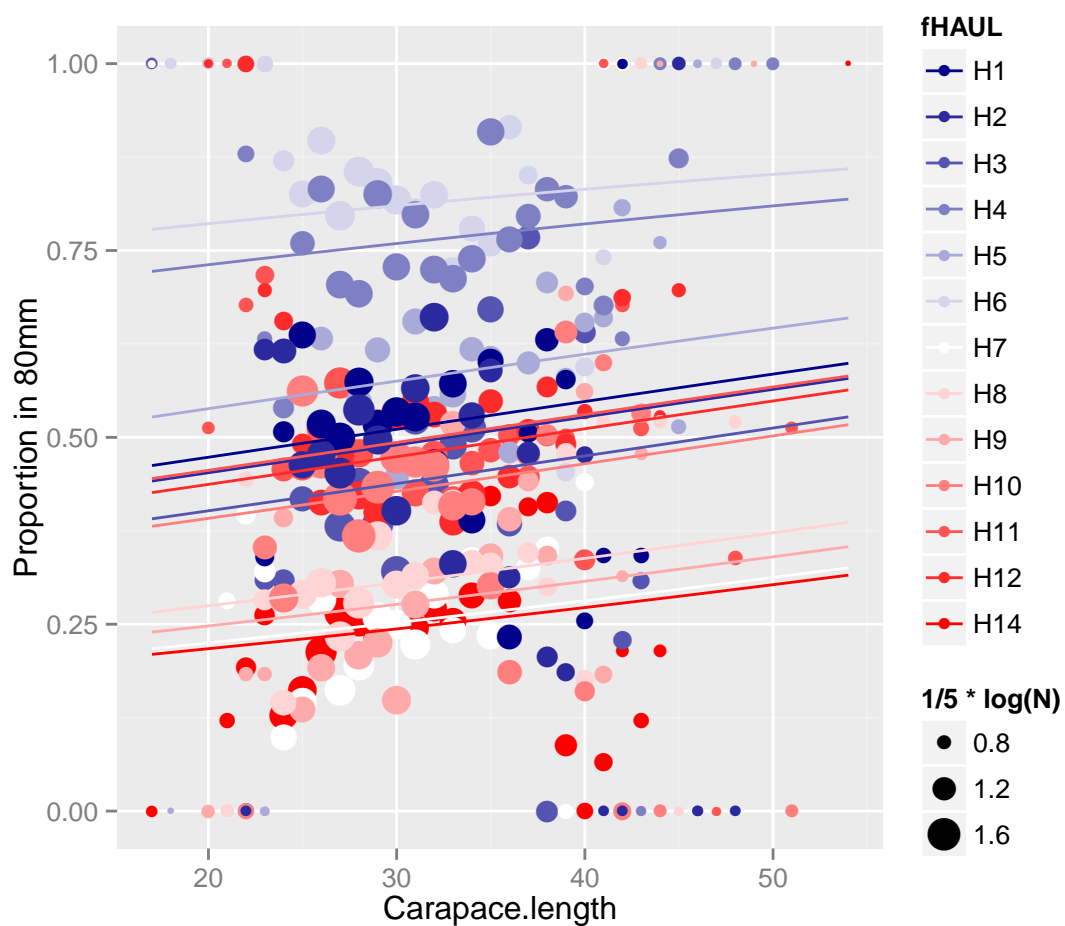


Figure 3: Observed proportion in the 80mm by haul. Note the wide variability of the proportion with some having much higher or lower proportions. Fitted lines come from the GLMM with carapace length only.

We can take a look at additional measured covariates to see if these relate to the haul-level variability (random effects in the model above) (Figure 4). Firstly, read in the haul duration data

```
gear.details <- read.xls("../data/2015 BIM Nephrops quad rig trials/Our Lass 2 70_80_90_100")

## merge with the neph data
gear.details$fHAUL <- factor(paste("H", gear.details$Tow., sep = ""))

tmp <- strsplit(as.character(gear.details$Tow.duration), split = ":")

hr <- as.numeric(unlist(lapply(tmp, "[", 1)))
min <- as.numeric(unlist(lapply(tmp, "[", 2)))

gear.details$dec.hr <- hr + min / 60
```

```
##
ranef.df <- data.frame(fHAUL = levels(neph.7080.cast$fHAUL),
                      ranef = coef(gamm.alt)[-c(1:5)])

##
covar.names <- c("fHAUL", "mesh70mm_Net.position", "mesh70mm_Total.catch",
                "mesh80mm_Net.position", "mesh80mm_Total.catch")

##
covar.df <- unique(neph.7080.cast[, covar.names])
## include haul duration
covar.df <- merge(covar.df, gear.details[, c("fHAUL", "dec.hr")])
##
ranef.df <- merge(ranef.df, covar.df)
## convert to long format for plotting
ranef.df <- melt(ranef.df, id = c("fHAUL", "ranef"))

##
ggplot(ranef.df, aes(x = value, y = ranef)) +
  geom_point() +
  facet_wrap(~ variable, scales = "free") +
  xlab("Covariate value") +
  ylab("Random effect")
```

There are some strong relationships between the random effects and measured covariates (Figure 4). It is best to include these measured variables in the model as fixed effects.

```
library(lme4)
library(effects)

## including additional covariates
## check identifiability
neph.7080.cast$fmesh80mm_Net.position <-
  factor(paste("pos",
               neph.7080.cast$mesh80mm_Net.position, sep = ""))
```

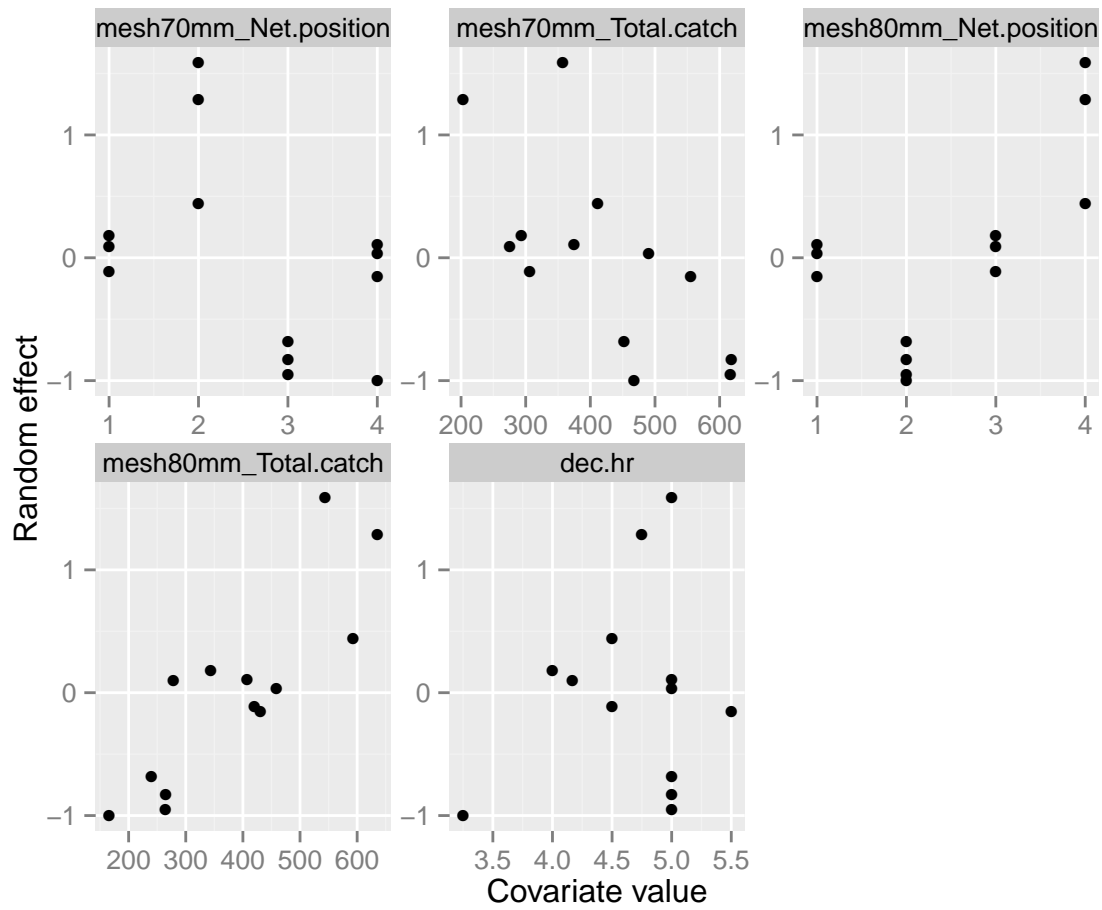


Figure 4: Relationship between the random effects of the carapace length only model and measured covariates .

```

neph.7080.cast <- merge(neph.7080.cast, gear.details[, c("fHAUL", "dec.hr")])

## using log of total catch weights - return to this
neph.7080.cast$log.mesh80mm_Total.catch <- log(neph.7080.cast$mesh80mm_Total.catch)
neph.7080.cast$log.mesh70mm_Total.catch <- log(neph.7080.cast$mesh70mm_Total.catch)

max.Carapace.length <- max(neph.7080.cast$Carapace.length)
neph.7080.cast$prop.Carapace.length <- neph.7080.cast$Carapace.length / max.Carapace.length

## Note should return to this warning later
## fits okay in gam but glmm used for effects package
glmm.alt.covar <- glmer(cbind(mesh80mm_Count, mesh70mm_Count) ~
  ##I(log(mesh80mm_Total.catch / mesh70mm_Total.catch)) * Carapace.l
  log.mesh80mm_Total.catch + log.mesh70mm_Total.catch +
  prop.Carapace.length +
  fmesh80mm_Net.position +
  (1 | fHAUL),
  offset =
  log(mesh80mm_Overall.Sampling.Ratio /
    mesh70mm_Overall.Sampling.Ratio),
  family = binomial,
  data = neph.7080.cast,
  control=glmerControl(optimizer="bobyqa"))

glmm.alt.covar.catchdur <- glmer(cbind(mesh80mm_Count, mesh70mm_Count) ~
  ##I(log(mesh80mm_Total.catch / mesh70mm_Total.catch)) * C
  log.mesh80mm_Total.catch + log.mesh70mm_Total.catch +
  prop.Carapace.length +
  fmesh80mm_Net.position +
  poly(dec.hr, 1) +
  (1 | fHAUL),
  offset =
  log(mesh80mm_Overall.Sampling.Ratio /
    mesh70mm_Overall.Sampling.Ratio),
  family = binomial,
  data = neph.7080.cast,
  control=glmerControl(optimizer="bobyqa"))

glmm.alt.covar.catchdur.nobulk <- glmer(cbind(mesh80mm_Count, mesh70mm_Count) ~
  ##I(log(mesh80mm_Total.catch / mesh70mm_Total.catch)
  ##log.mesh80mm_Total.catch + log.mesh70mm_Total.ca
  prop.Carapace.length +
  fmesh80mm_Net.position +
  poly(dec.hr, 1) +
  (1 | fHAUL),
  offset =
  log(mesh80mm_Overall.Sampling.Ratio /
    mesh70mm_Overall.Sampling.Ratio),
  family = binomial,
  data = neph.7080.cast,
  control=glmerControl(optimizer="bobyqa"))

```

```

## include squared length term
glmm.alt.covar2 <- glmer(cbind(mesh80mm_Count, mesh70mm_Count) ~
  ##I(log(mesh80mm_Total.catch / mesh70mm_Total.catch)) * Carapace.length
  log.mesh80mm_Total.catch + log.mesh70mm_Total.catch +
  poly(prop.Carapace.length, 2) +
  poly(dec.hr, 1) +
  fmesh80mm_Net.position +
  (1 | fHAUL),
  offset =
  log(mesh80mm_Overall.Sampling.Ratio /
    mesh70mm_Overall.Sampling.Ratio),
  family = binomial,
  data = neph.7080.cast,
  control=glmerControl(optimizer="bobyqa"))

## in GAM
gamm.alt.covar2 <- gam(cbind(mesh80mm_Count, mesh70mm_Count) ~
  ##I(log(mesh80mm_Total.catch / mesh70mm_Total.catch)) * Carapace.length
  log.mesh80mm_Total.catch + log.mesh70mm_Total.catch +
  s(prop.Carapace.length, k = 5, fx = TRUE) +
  poly(dec.hr, 1) +
  fmesh80mm_Net.position +
  s(fHAUL, bs="re", by = dum),
  offset =
  log(mesh80mm_Overall.Sampling.Ratio /
    mesh70mm_Overall.Sampling.Ratio),
  family = binomial,
  data = neph.7080.cast)

## use effects package to get prediction for model with net position
## set predictor variables
xlevels <- list(prop.Carapace.length = cl.vec/max.Carapace.length)

## if we wanted to set the proportions of net positions equivalent
## otherwise set to the proportion observed in the data
##given.values <- c("fmesh80mm_Net.positionpos2" = 1/4,
##                  "fmesh80mm_Net.positionpos3" = 1/4,
##                  "fmesh80mm_Net.positionpos4" = 1/4
##                  )

##cl.effect <- effect("Carapace.length", glmm.alt.covar, xlevels = xlevels, offset = 0, given.values = given.values)
cl.effect <- effect("prop.Carapace.length", glmm.alt.covar.catchdur, xlevels = xlevels, offset = 0, given.values = given.values)

```

Plot the effect of carapace length with the other variables set to their mean in the data (Figure 5).

```

with(count.df, plot(Carapace.length, prop.80, ylim = c(0, 1), pch = 19,
  xlab = "Carapace length (mm)",
  ylab = "Proportion (N80mm/(N70mm + N80mm))",
  bty = "L",
  cex = 1/5 * log(N)))

```

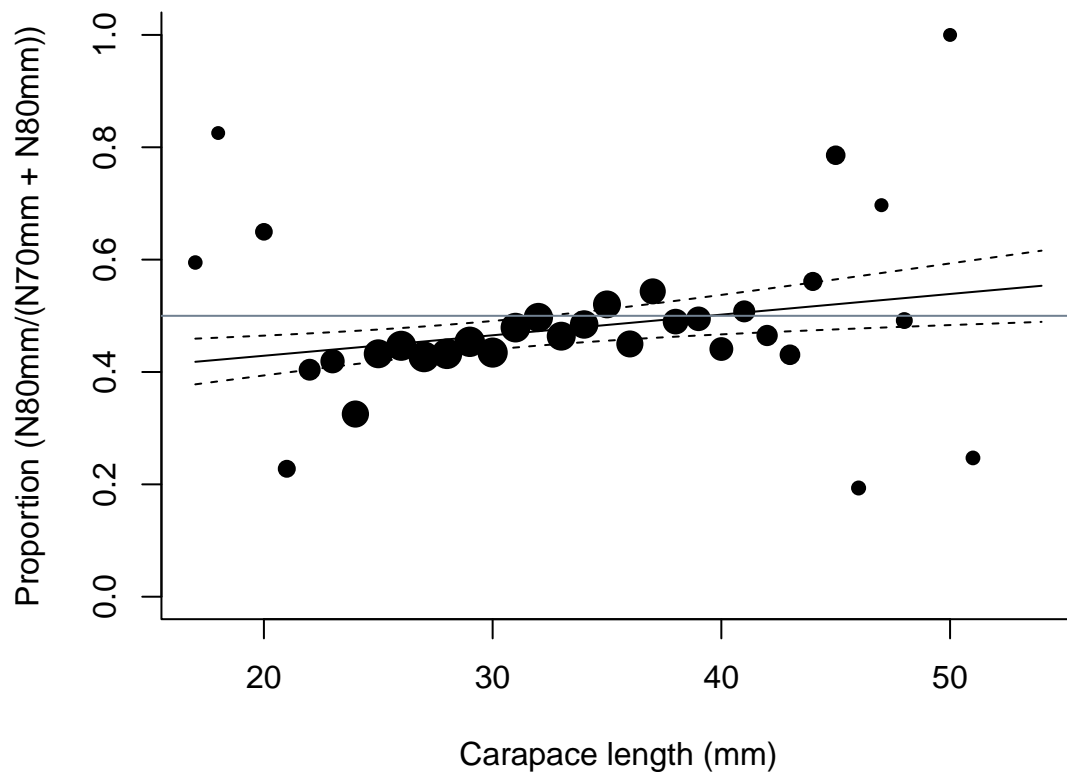


Figure 5: Predicted proportion from binomial GLMM with covariates. Note in the predictions the bulk weights are set to their mean and the net positions to their proportional occurrence in the data.

```
## effects prediction
lines(cl.vec, plogis(cl.effect$fit[, 1]))
lines(cl.vec, plogis(cl.effect$lower[, 1]), lty = 2)
lines(cl.vec, plogis(cl.effect$upper[, 1]), lty = 2)
abline(h = 0.5, col = "slategrey")
```

```
## Note the wide confidence intervals
```

Finally test length effect in covariate model

```
glmm.alt.covar.catchdur.nolength <- glmer(cbind(mesh80mm_Count, mesh70mm_Count) ~
  log.mesh80mm_Total.catch + log.mesh70mm_Total.catch +
  fmesh80mm_Net.position +
  poly(dec.hr, 1) +
```

```

(1 | fHAUL),
offset =
log(mesh80mm_Overall.Sampling.Ratio /
    mesh70mm_Overall.Sampling.Ratio),
family = binomial,
data = neph.7080.cast,
control=glmerControl(optimizer="bobyqa"))

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,
: Model failed to converge with max|grad| = 0.0073401 (tol = 0.001, component 2)

## likelihood ratio test
anova(glmm.alt.covar.catchdur.nolength, glmm.alt.covar.catchdur)

## Data: neph.7080.cast
## Models:
## glmm.alt.covar.catchdur.nolength: cbind(mesh80mm_Count, mesh70mm_Count) ~ log.mesh80mm_7
## glmm.alt.covar.catchdur.nolength:      log.mesh70mm_Total.catch + fmesh80mm_Net.position
## glmm.alt.covar.catchdur.nolength:      1) + (1 | fHAUL)
## glmm.alt.covar.catchdur: cbind(mesh80mm_Count, mesh70mm_Count) ~ log.mesh80mm_Total.cat
## glmm.alt.covar.catchdur:      log.mesh70mm_Total.catch + prop.Carapace.length + fmesh80mm
## glmm.alt.covar.catchdur:      poly(dec.hr, 1) + (1 | fHAUL)
##
##              Df      AIC      BIC  logLik deviance  Chisq
## glmm.alt.covar.catchdur.nolength  8 1417.9 1448.5 -700.97   1401.9
## glmm.alt.covar.catchdur           9 1411.6 1446.0 -696.79   1393.6 8.3654
##
##              Chi Df Pr(>Chisq)
## glmm.alt.covar.catchdur.nolength
## glmm.alt.covar.catchdur           1    0.003824 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## significant effect of carapace length

```

### 3 Economic section

```

curve(0.00032 * x^3.21, from = 0, to = 60, xlab = "Carapace length (mm)",
      ylab = "Total weight (g)")

npkilo.breaks <- c(1, 11, 16, 21, 31, 41, 50)
wt.breaks <- (1/ npkilo.breaks) * 1e3
abline(h = wt.breaks, col = "grey")

## corresponding length cut-offs
lt.breaks = exp(log(wt.breaks / 0.00032)/3.21)
abline(v = lt.breaks, col = "grey")

```



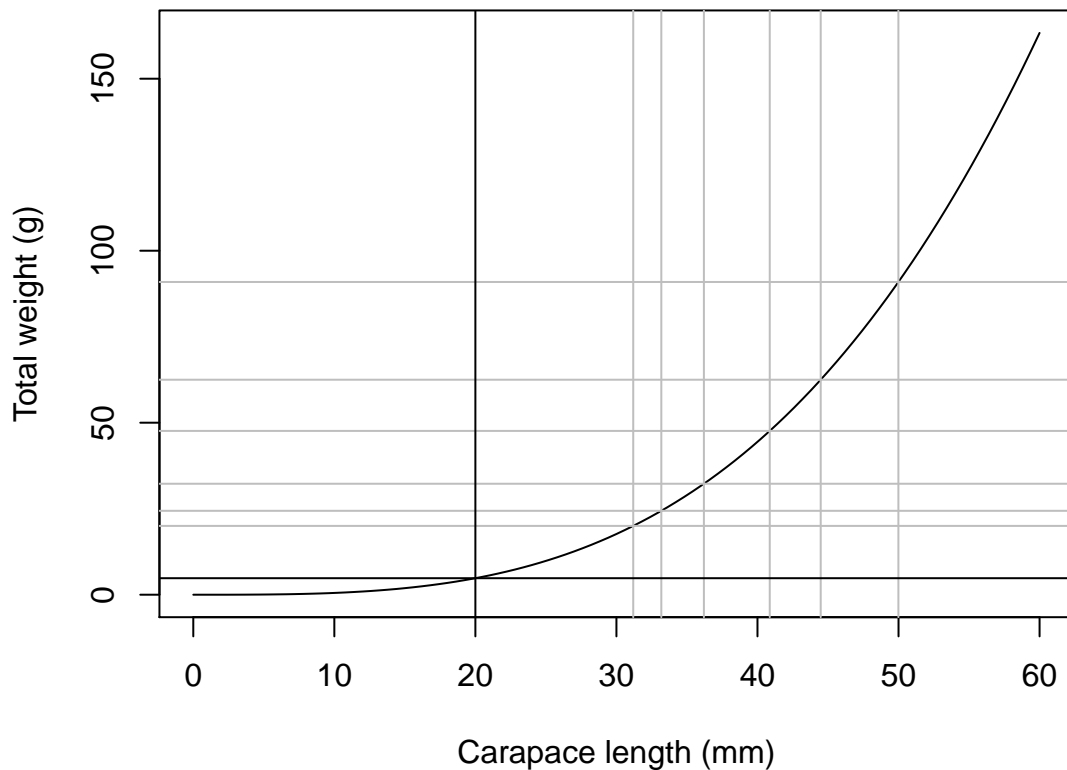


Figure 6: Length - weight relationship with horizontal price bands converted to length-based price bands. Price band in black denotes below Minimum Conservation Reference Size prawns.

```
## below MCRS band
abline(v = 20)
abline(h = 0.00032*20^3.21)
```

Simulate distributions of retained catches with different means

```
sim.means <- c(24, 27, 33)

## get the vector of raised counts for the whole Our Lass trip
neph.lengths <- with(subset(neph.dat, Mesh.Size%in%c("70mm", "80mm")),
  rep(Carapace.length, times = round(Raised.count.)))

length(neph.lengths) ## raised number of prawns caught

## [1] 206009
```

```

(mean.cl <- mean(neph.lengths)) ## mean carapace length below

## [1] 30.09305

## differences in the mean observed length and simulated means
## plus 0.5 to account for nearest mm below
sim.diff <- mean.cl - (sim.means + 0.5)

## plot the simulated distributions of retained catches
hist.orig <- hist(neph.lengths, breaks = seq(0.5, 60, by = 1), plot = FALSE)
##hist.orig <- hist(neph.lengths, breaks = seq(1, 60, by = 1), plot = FALSE)
cl.mids <- hist.orig$mids
count.sim <- matrix(hist.orig$counts, nrow = 1)

plot(hist.orig$mids, hist.orig$counts, type = "l", xlim = c(10, 50),
      xlab = "Carapace length (mm)", ylab = "Count per 1 mm bin")
##
for(i in 1:length(sim.means)){
  hist.sim <- hist(neph.lengths - sim.diff[i],
                  breaks = seq(0.5, 60, by = 1), plot = FALSE)
  ##breaks = seq(1, 60, by = 1), plot = FALSE)
  lines(hist.sim$mids, hist.sim$counts, lty = 1 + i, col = 1+i)
  count.sim <- rbind(count.sim, hist.sim$counts)
}

legend("topright", legend =
      c("Original (30.09mm)", "24mm mean",
        "27mm mean", "33mm mean"),
      lty = c(1:4), col = 1:4, bty = "n")

##
rownames(count.sim) <- c("original", "mm.24", "mm.27", "mm.33")

```

## Per-haul variables

```

## count per haul (13 hauls)
count.phaul.sim <- count.sim / (13) ## note this is the sum for the 70 and 80

## get predicted prawn weight per length bin in kgs
wt.mids <- (0.00032 * cl.mids^3.21) / 1e3

## get total weight per haul
wt.phaul.sim <- t(apply(count.phaul.sim, 1, FUN = function(z){z * wt.mids}))

## predicted price per length class
## note 20mm CL included here
wt.cuts <- cut(wt.mids, breaks = c(wt.breaks, 0.00032*20^3.21, 0) / 1e3)

```

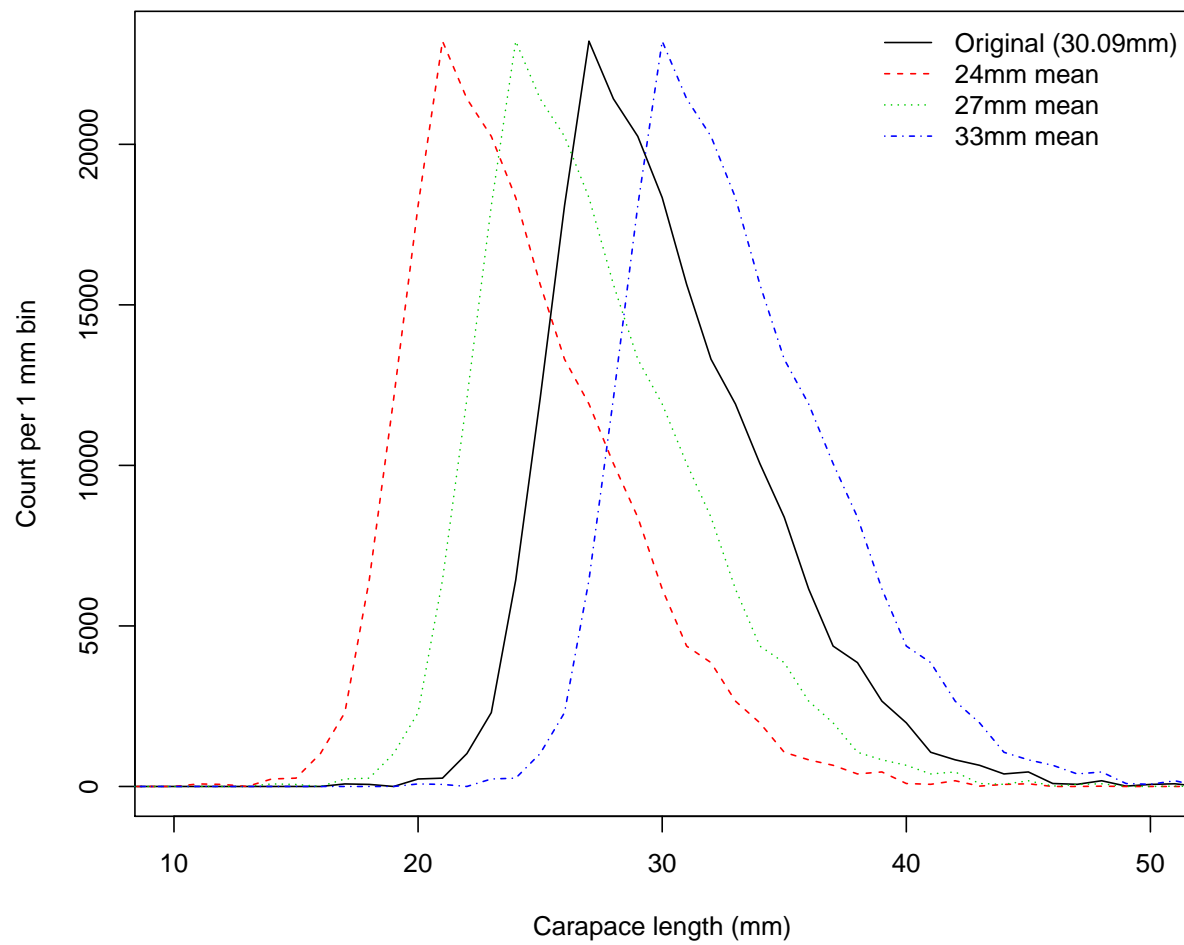


Figure 7: Simulated distributions of retained catches.

```

price.df <- data.frame(wt.bin = unique(wt.cuts),
                      price = c(-0.2, 1.90, 4.75, 5.35, 7.75, 10.75, 13, 13))

price.df

##           wt.bin price
## 1      (0,0.0048] -0.20
## 2    (0.0048,0.02]  1.90
## 3    (0.02,0.0244]  4.75
## 4 (0.0244,0.0323]  5.35
## 5 (0.0323,0.0476]  7.75
## 6 (0.0476,0.0625] 10.75
## 7 (0.0625,0.0909] 13.00
## 8      (0.0909,1] 13.00

## prices per length class bin
price.bin.df <- merge(data.frame(wt.bin = wt.cuts), price.df)
price.bin.df$Carapace.length <- cl.mids

## value per haul
value.phaul.sim <- t(apply(wt.phaul.sim, 1, FUN = function(z){z * price.bin.df$price}))

```

Finally, split the variables at length between 70 and 80mm.

```

## get predicted proportions in 80mm for given carapace length mid-points
xlevels <- list(prop.Carapace.length = cl.mids/max.Carapace.length)

## set the proportions of net positions equivalent
given.values <- c("fmesh80mm_Net.positionpos2" = 1/4,
                  "fmesh80mm_Net.positionpos3" = 1/4,
                  "fmesh80mm_Net.positionpos4" = 1/4
                  )

cl.effect <- effect("prop.Carapace.length", glmm.alt.covar.catchdur, xlevels = xlevels, of)

p80 <- plogis(cl.effect$fit[, 1])

## plot(cl.mids, p80, ylim = c(0, 1))

## split out 70 and 80
## count
count.phaul.sim.80 <- t(apply(count.phaul.sim, 1, FUN = function(z){z * p80}))
count.phaul.sim.70 <- t(apply(count.phaul.sim, 1, FUN = function(z){z * (1 - p80)}))

## weight
wt.phaul.sim.80 <- t(apply(wt.phaul.sim, 1, FUN = function(z){z * p80}))
wt.phaul.sim.70 <- t(apply(wt.phaul.sim, 1, FUN = function(z){z * (1 - p80)}))

## value

```

```
value.phaul.sim.80 <- t(apply(value.phaul.sim, 1, FUN = function(z){z * p80}))
value.phaul.sim.70 <- t(apply(value.phaul.sim, 1, FUN = function(z){z * (1 - p80)}))
```

Plot the counts, weights and value per length bin split by 70 and 80mm (Figure 8).

```
par(mfrow = c(2, 2), mar = c(2, 3, 1, 1), oma = c(2, 2, 1, 1))
## Count
matplot(cl.mids, t(count.phaul.sim.80),
        type = "l", col = "darkblue",
        xlim = c(10, 50), ylim = c(0, 1e3))
matlines(cl.mids, t(count.phaul.sim.70), type = "l", col = "red1")
mtext(side = 2, line = 2, text = "Count per 1mm bin")
## to demonstrate same retention across scenarios
## use xlim = c(15, 40) and abline(v = c(23.3, 26.3, 29.3))
## Weight
matplot(cl.mids, t(wt.phaul.sim.80),
        type = "l", col = "darkblue",
        xlim = c(10, 50), ylim = c(0, 20))
matlines(cl.mids, t(wt.phaul.sim.70), type = "l", col = "red1")
mtext(side = 2, line = 2, text = "Weight (kg) per 1mm bin")
## Value
matplot(cl.mids, t(value.phaul.sim.80),
        type = "l", col = "darkblue",
        xlim = c(10, 50), ylim = c(0, 110))
matlines(cl.mids, t(value.phaul.sim.70), type = "l", col = "red1")
mtext(side = 2, line = 2, text = "Value (euro) per 1mm bin")
##
plot.new()
legend("center", legend = c("70mm", "80mm", NA, "Original (30.09mm)", "24mm mean",
                           "27mm mean", "33mm mean"),
      lty = c(1,1,NA, 1:4),
      col = c("red1", "darkblue", NA, rep("darkblue", 4)),
      bty = "n"
    )
```

Summary table (as in BIM report Table 3)

```
## calculate the resulting mean sizes per length class
##apply(count.phaul.sim, 1, FUN = function(z){sum(cl.mids * z) / sum(z)})
mean.cl.70 <- apply(count.phaul.sim.70, 1, FUN = function(z){sum(cl.mids * z) / sum(z)})
mean.cl.80 <- apply(count.phaul.sim.80, 1, FUN = function(z){sum(cl.mids * z) / sum(z)})

## data frame for predictions
na.vec <- rep(NA, 8)
sim.order <- c("original", "mm.33", "mm.27", "mm.24")

pred.df <- data.frame(Mesh.Size = rep(c(70, 80), each = 4),
                     Mean.CL = rep(sim.order, 2),
                     Mean.CL.mesh = c(mean.cl.70[sim.order], mean.cl.80[sim.order]),
                     c.less.mcrcs = na.vec,
```

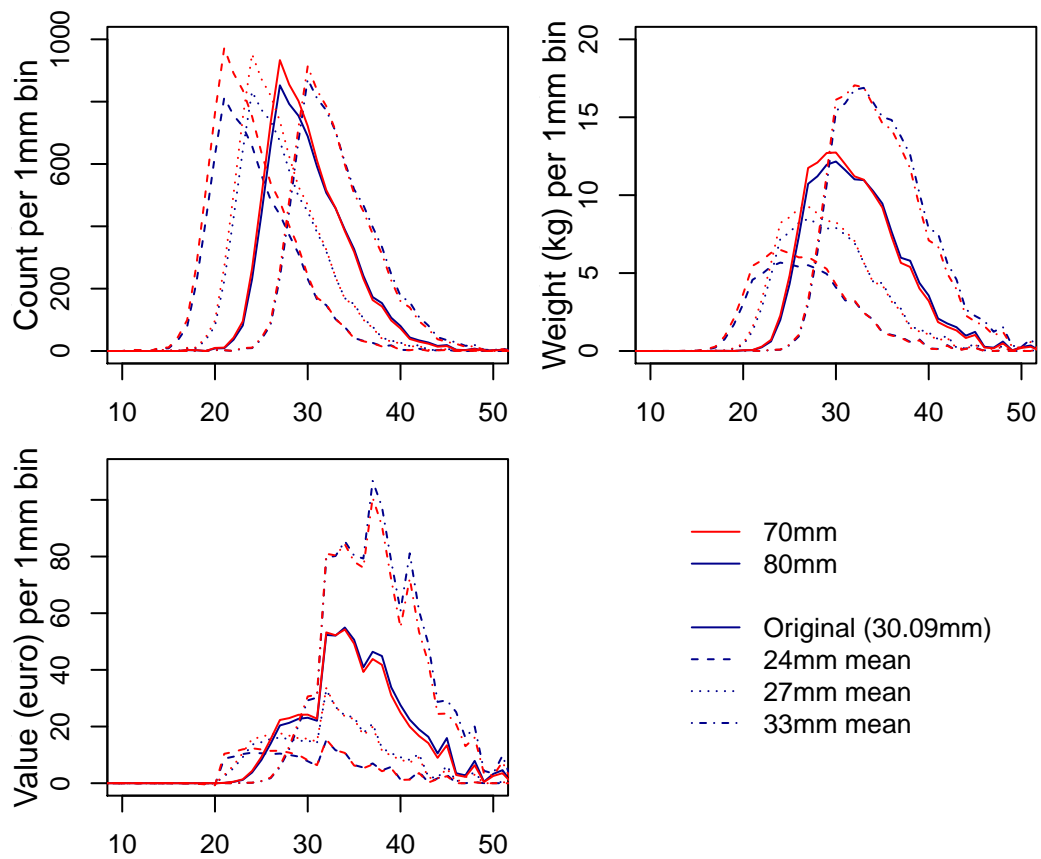


Figure 8: Simulated counts, weight and value by: length class, mesh size and simulated scenario (24mm mean catch, 27mm mean catch, 30mm mean catch (original), 33mm mean catch).

```

        c.greater.mcrcs = na.vec,
        total.catch = na.vec,
        v.less.mcrcs = na.vec,
        v.greater.mcrcs = na.vec,
        total.value = na.vec,
        stringsAsFactors = FALSE)

##

for(i in 1:dim(pred.df)[1]){
  print(i)
  wt <- get(paste("wt.phaul.sim.", pred.df$Mesh.Size[i], sep = ""))
  value <- get(paste("value.phaul.sim.", pred.df$Mesh.Size[i], sep = ""))
  ## catch
  pred.df$c.less.mcrcs[i] <- sum(wt[pred.df$Mean.CL[i], cl.mids < 20])
  pred.df$c.less.mcrcs[i] <- round(pred.df$c.less.mcrcs[i], 2)
  pred.df$c.greater.mcrcs[i] <- sum(wt[pred.df$Mean.CL[i], cl.mids >= 20])
  pred.df$c.greater.mcrcs[i] <- round(pred.df$c.greater.mcrcs[i], 2)
  pred.df$total.catch[i] <- sum(wt[pred.df$Mean.CL[i], ])
  pred.df$total.catch[i] <- round(pred.df$total.catch[i], 2)
  ## value
  pred.df$v.less.mcrcs[i] <- sum(value[pred.df$Mean.CL[i], cl.mids < 20])
  pred.df$v.less.mcrcs[i] <- round(pred.df$v.less.mcrcs[i], 2)
  pred.df$v.greater.mcrcs[i] <- sum(value[pred.df$Mean.CL[i], cl.mids >= 20])
  pred.df$v.greater.mcrcs[i] <- round(pred.df$v.greater.mcrcs[i], 2)
  pred.df$total.value[i] <- sum(value[pred.df$Mean.CL[i], ])
  pred.df$total.value[i] <- round(pred.df$total.value[i], 2)
}

## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8

```

```
library(xtable)
print(xtable(pred.df, digits = 2, align = "lcccccccc"), include.rownames = FALSE)
```

Mesh.Size	Mean.CL	Mean.CL.mesh	c.less.mcrrs	c.greater.mcrrs	total.catch	v.less.mcrrs	v.greater.mcrrs	total.value
70.00	original	29.96	0.02	152.80	152.82	-0.00	628.79	628.79
70.00	mm.33	32.96	0.00	200.60	200.60	0.00	1113.07	1113.07
70.00	mm.27	26.97	0.25	112.89	113.14	-0.05	353.73	353.68
70.00	mm.24	23.97	3.46	77.50	80.96	-0.69	190.95	190.26
80.00	original	30.23	0.02	150.65	150.67	-0.00	648.12	648.12
80.00	mm.33	33.23	0.00	206.10	206.10	0.00	1187.36	1187.36
80.00	mm.27	27.23	0.20	106.90	107.11	-0.04	350.37	350.33
80.00	mm.24	24.24	2.78	70.86	73.65	-0.56	182.10	181.55