SELECT package demonstration  
unpaired-haul relative selectivity

## Summary

Here we re-use paired haul data but without the paired haul structure.

With unpaired data, a data storage strategy is used whereby the data are put into an artificial paired format that creates a paired-haul structure by pairing each haul with a ghost zero-catch haul with the other gear. Consequently, the SELECT curve fitting functions can be used without knowing whether the data are paired or unpaired. The paired, or unpaired, nature of the data becomes relevant when using the bootstrap and permutation functions, which have a paired argument (default=TRUE).

**This case study demonstrates:**

* Reading data from separate gear-specific files and stacking them into a single data frame.
* Working with sub-sampled data.
* Converting unpaired data into the paired SELECT format using the SELECT.FORMAT function.
* Fitting catch share (relative selectivity) curves using splines via the SplineSELECT function.
* Using the bootstrap function bootSELECT to estimate the uncertainty in the catch share curve.
* Using the permutation function permSELECT to quantify the evidence for a length effect on the catch share.

### Data source

The data are for school prawn relative selectivity in penaeid trawls from the experiments conducted by Broadhurst et al., (2018, T45 side panels improve penaeid-trawl selection. Fisheries Research, 204: 8-15).

Here, any actual paired haul structure is ignored, and the hauls with the 32 m and 35 mm square side panels are treated as unpaired.

### Load required packages

require(tidyverse)  
require(mgcv)  
require(SELECT)  
require(readxl) #This package is installed with tidyverse  
nsim=1000 #Number of bootstrap or permutation simulations. Should be >=1000 in practice.

### Read in the data

NOTE: Paired haul D6H4 with conventional/T35 pairing has NAs for the penaeid counts in the conventional gear, and so must be removed

GearA.df=read\_excel("SchoolPrawnLenFreqs.xlsx", sheet = "32 square trawl")  
GearB.df=read\_excel("SchoolPrawnLenFreqs.xlsx", sheet = "35 square trawl")  
CommonHauls=intersect(unique(GearA.df$Day),unique(GearB.df$Day))  
  
#Choice of hauls to use  
#filter(!Haul %in% CommonHauls) |>  
  
GearA.df = GearA.df |>   
 rename(Haul=Day, n=No.school) |>   
 filter(Haul %in% CommonHauls & Haul!="D6H4") |>   
 mutate(Haul=paste0(Haul,".A"),q=1/Sf.school,Gear="A") |>   
 select(-Sf.school)  
GearB.df = GearB.df |>   
 rename(Haul=Day, n=No.school) |>   
 filter(Haul %in% CommonHauls & Haul!="D6H4") |>   
 mutate(Haul=paste0(Haul,".B"),q=1/Sf.school,Gear="B") |>  
 select(-Sf.school)  
  
head(GearA.df)

## # A tibble: 6 × 5  
## Haul CL n q Gear   
## <chr> <dbl> <dbl> <dbl> <chr>  
## 1 D1H2.A 1 0 0.0839 A   
## 2 D1H2.A 2 0 0.0839 A   
## 3 D1H2.A 3 0 0.0839 A   
## 4 D1H2.A 4 0 0.0839 A   
## 5 D1H2.A 5 0 0.0839 A   
## 6 D1H2.A 6 0 0.0839 A

head(GearB.df)

## # A tibble: 6 × 5  
## Haul CL n q Gear   
## <chr> <dbl> <dbl> <dbl> <chr>  
## 1 D1H2.B 1 0 0.116 B   
## 2 D1H2.B 2 0 0.116 B   
## 3 D1H2.B 3 0 0.116 B   
## 4 D1H2.B 4 0 0.116 B   
## 5 D1H2.B 5 0 0.116 B   
## 6 D1H2.B 6 0 0.116 B

### Stack the separate dataframes

Note the conversion from sub-sampling scaling factors to sampling fractions and removal of CLs that are outside of the range of measured data.

Df=rbind(GearA.df,GearB.df) #Stack the two dataframes  
Df = Df |> filter(CL>=5 & CL<=25)  
Df |> group\_by(Haul) |> summarise(n=sum(n/q)) #Check the haul totals

## # A tibble: 24 × 2  
## Haul n  
## <chr> <dbl>  
## 1 D1H2.A 2228.  
## 2 D1H2.B 1800.  
## 3 D1H6.A 2326.  
## 4 D1H6.B 1872.  
## 5 D2H4.A 1565.  
## 6 D2H4.B 1273.  
## 7 D2H5.A 1400.  
## 8 D2H5.B 1216.  
## 9 D3H1.A 910.  
## 10 D3H1.B 846.  
## # ℹ 14 more rows

### Put into SELECT format

The SELECT.FORMAT function converts the unpaired data into a paired format. Note that it includes a variable (column) indicating the actual gear for the haul (since the other element of the pair is a ghost zero-catch haul).

Gears.df=SELECT.FORMAT(Df,by=c("Haul","CL"),gear="Gear",freq="n",  
 q.name="q",paired=F)  
head(Gears.df)

## # A tibble: 6 × 7  
## Haul CL nA nB qA qB Gear   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>  
## 1 D1H2.A 5 0 0 0.0839 0 A   
## 2 D1H2.A 6 0 0 0.0839 0 A   
## 3 D1H2.A 7 0 0 0.0839 0 A   
## 4 D1H2.A 8 1 0 0.0839 0 A   
## 5 D1H2.A 9 0 0 0.0839 0 A   
## 6 D1H2.A 10 1 0 0.0839 0 A

### Define variable names

names(Gears.df)

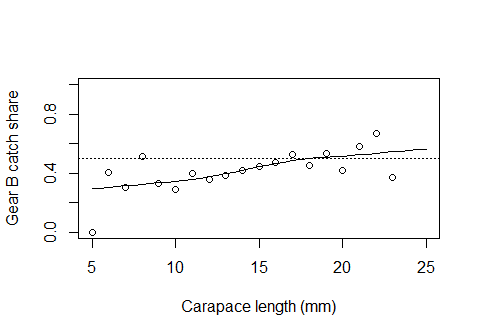
## [1] "Haul" "CL" "nA" "nB" "qA" "qB" "Gear"

var.names=c("CL","nA","nB")  
q.names=c("qA","qB")

### Define a prediction function to be used with the bootstrap

This fit used the SplineSELECT defaults, but more generally one may want to try other values of k, say 5 and 10.

#Define the bootstrap prediction function  
CLseq=seq(5,25,0.5) #Carapace lengths to use for predn  
Predn=function(data,var.names) {  
 SplineFit=SplineSELECT(data,var.names,q.names,bs="tp",  
 quasi=T,sumHauls=T,k=7,q.ODadjust = T)  
 predict(SplineFit,newdata=data.frame(CL=CLseq),type="response") }  
#Check that it works  
predn=Predn(Gears.df,var.names)  
  
#Plot predictions against observed proportions  
Tots.df=Raw2Tots(Gears.df,var.names,q.names) |>   
 transform(lgth=CL, y=nB/(nA+nB))  
plot(y~CL,data=Tots.df,ylim=c(0,1),xlab="Carapace length (mm)",  
 ylab="Gear B catch share")  
points(CLseq,predn,type="l")  
abline(h=0.5,lty=3)



### Do the bootstrap of the catch share curve

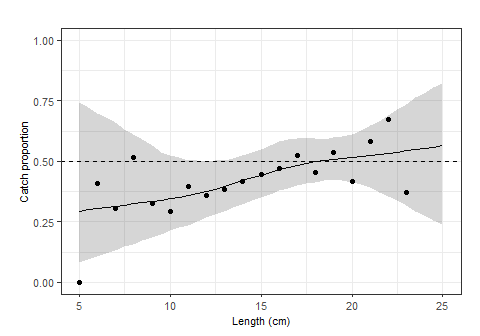
Note that it is necessary to provide the gear argument to SELECT when analyzing unpaired data.

BootPreds=bootSELECT(Gears.df,var.names,Predn,haul="Haul",nsim=nsim,  
 paired=F,gear="Gear",verbose=F)

##   
## Bootstrap successfully completed

BootPlot(BootPreds,CLseq,predn,Data=Tots.df) +  
 geom\_hline(yintercept=0.5,linetype="dashed")

## Warning: Removed 2 rows containing missing values or values outside the scale range  
## (`geom\_point()`).



### Define a goodness of fit function to be used with the permutation test

#Define the deviance explained function  
DevExplained=function(data,var.names) {  
 SplineFit=SplineSELECT(data,var.names,q.names,bs="tp",  
 quasi=T,sumHauls=T,k=7,q.ODadjust = T)   
 summary(SplineFit)$dev.expl }  
#Check that it works  
ObsDev=DevExplained(Gears.df,var.names)  
cat("Proportion of deviance explained is",ObsDev,"\n")

## Proportion of deviance explained is 0.7180213

PermDev=permSELECT(Gears.df,var.names,DevExplained,haul="Haul",nsim=nsim,  
 paired=F,gear="Gear",verbose=F)

##   
## Permutations successfully completed

#Proportion of permuted gof values greater than the observed  
cat("The p-value for a CL effect is",mean(PermDev>ObsDev))

## The p-value for a CL effect is 0.342