

SELECT package demonstration

Paired-haul relative selectivity

Summary

This case study demonstrates:

- Reading data from separate gear-specific files and merging them into a single data frame.
- Handling sub-sampled data.
- Fitting splines to relative selectivity data using the `SplineSELECT` function.
- Using the bootstrap function `bootSELECT` to estimate the uncertainty in the catch share curve.
- Using the permutation function `permSELECT` to test for a difference in the size selectivity of the two gears.

Data source

The data are for school prawn relative selectivity in penaeid trawls with differing mesh types of the side panel, 43 mm diamond, or 35 mm square (Broadhurst et al., 2018. T45 side panels improve penaeid-trawl selection. Fisheries Research, 204: 8-15).

Load required packages

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2    3.5.2      v tibble     3.2.1
## v lubridate  1.9.4      v tidyr      1.3.1
## v purrr      1.0.4
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
require(mgcv)
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
##
```

```
## Attaching package: 'nlme'
##
## The following object is masked from 'package:dplyr':
##
## collapse
##
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
require(SELECT)
```

```
## Loading required package: SELECT
```

```
require(readxl) #This package is installed with tidyverse
```

```
## Loading required package: readxl
```

Read in data

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

```
Conv.df=read_excel("Exp 18 LenFreqs.xlsx", sheet = "Conventional trawl", na="NA")
T35.df=read_excel("Exp 18 LenFreqs.xlsx", sheet = "35 square trawl", na="NA")

Conv.df = Conv.df |> rename(Haul=Day, nConv=No.school, sfConv=Sf.school)
T35.df = T35.df |> rename(Haul=Day, n35=No.school, sf35=Sf.school)
Conv.df = Conv.df |> filter(Haul != "D6H4")
T35.df = T35.df |> filter(Haul != "D6H4")
```

Merge the separate dataframes to create a dataframe for the paired tows

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

```
#Merge conventional and 35mm side gears
Conv35.df = inner_join(Conv.df,T35.df,by=c("Haul","CL"))
#Convert scaling factors to sampling fractions
Conv35.df = Conv35.df |>
  transform(qConv=1/sfConv,q35=1/sf35) |> filter(CL>5 & CL<26)
```

Define variable names

```
var.names=c("CL","nConv","n35")
q.names=c("qConv","q35")
```

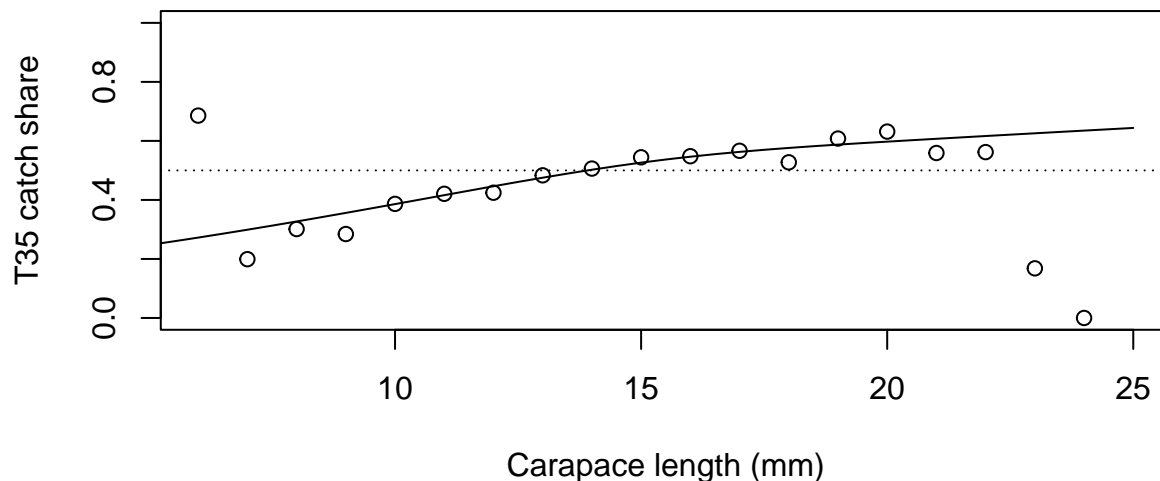
Define a prediction function to be used with the bootstrap

```

#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="cr",quasi=T,sumHauls=F,k=5,
    q.ODadjust = T)
  predict(SplineFit,newdata=data.frame(CL=CLseq),type="response") }
#Check it works
Tot35.df=Raw2Tots(Conv35.df,var.names,q.names) |>
  transform(lgth=CL, y=n35/(nConv+n35))
plot(y~CL,data=Tot35.df,ylim=c(0,1),xlab="Carapace length (mm)",
  ylab="T35 catch share",main="35 mm T45")
predn=Predn(Conv35.df,var.names)
points(CLseq,predn,type="l")
abline(h=0.5,lty=3)

```

35 mm T45



Do the bootstrap

```

nboot=100 #Increase to at least 1000 in practice
Boot35=bootSELECT(Conv35.df,var.names,Predn,haul="Haul",nsim=nboot,paired=T)

```

```

##
## Starting a 100 resamples bootstrap...
## |
##
## Bootstrap successfully completed

```

```

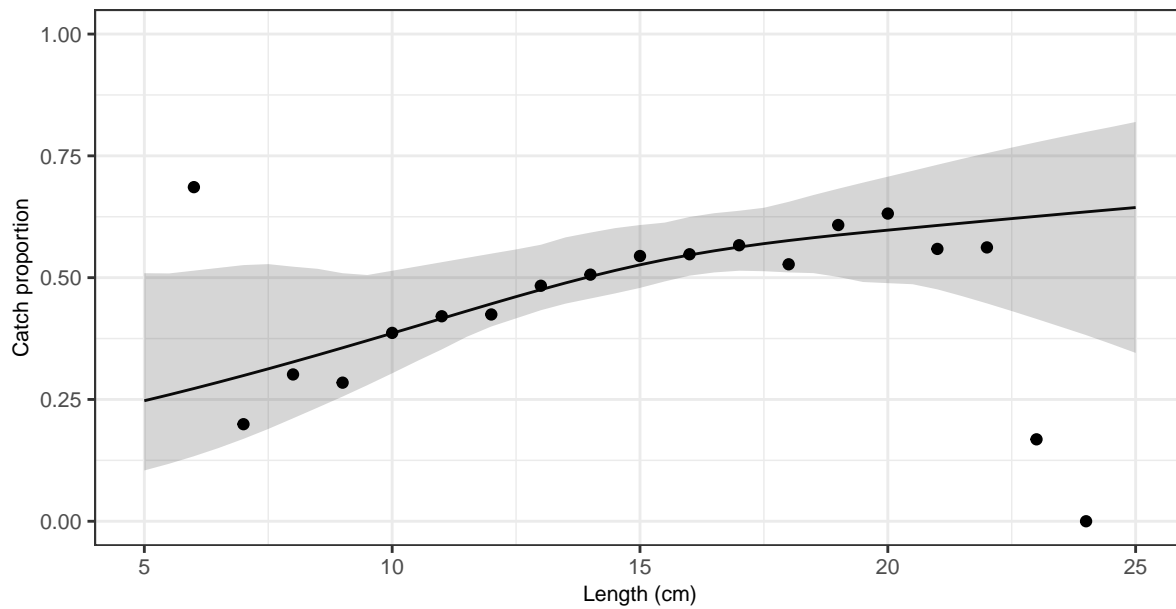
BootPlot(Boot35,CLseq,predn,Data=Tot35.df)

```

```

## Warning: Removed 1 row containing missing values or values outside the scale range
## ('geom_point()').

```



Define an r-squared function to be used with the permutation test

```
#Define the spline r-squared function
permRsqr=function(data,var.names) {
  SplineFit=SplineSELECT(data,var.names,q.names,bs="ts",quasi=T,sumHauls=F,k=5,
                           q.ODadjust = T)
  summary(SplineFit)$r.sq }
#Check it works
T35.rsqr=permRsqr(Conv35.df,var.names)
```

```
nperm=100 #Increase to at least 1000 in practice
Perm35=permSELECT(Conv35.df,var.names,permRsqr,haul="Haul",nsim=nperm,paired=T)
```

```
##
## Starting on 100 permutations...
## |
##
## Permutations successfully completed
```

```
#Proportion of permuted r-squared values greater than the observed
cat("The p-value for a CL effect is",mean(Perm35>T35.rsqr),"\n")
```

```
## The p-value for a CL effect is 0.01
```