Number at length and total catches estimates using Horwith-Thompson and ratio estimators

Laurent Dubroca

2021-06-07

Framework

In this report, after the simulation generated by `LD_01_simulation.Rmd` and the data conversion in `LD_03_simu2rdbes.Rmd`, estimation methods are tested and the results are compared with the population value.

Package and co

Library are loaded and code sourced, if any.

```
#library
library(dplyr)
library(ggplot2)
```

Data

Data are loaded in the R environment.

```
#load the data
datsim<-readRDS("../outputs/datasimu2.rds")
clrdb<-readRDS("../outputs/datclrdbsimpop.rds")
datrdbpop<-readRDS("../outputs/datrdbsimpop.rds")
datrdbsamp<-readRDS("../outputs/datrdbsimsamp.rds")
```

Exploratory data analyses

```
#a fct to join H1 hierachy
fct1<-function(dat){
    #dat<-datrdbpop
    pipo<-left_join(dat$FM,dat$SA)%>%
        left_join(dat$SS)%>%
        left_join(dat$FO)%>%
        left_join(dat$FT)%>%
```

```
        left_join(dat$VS)
    return(pipo)
}
#check data
tmp1<-datsim%>%filter(year==1)%>%mutate(vname=paste0("v",VDid),tname=paste0("t",TRid),ori="sim")
tmp2<-fct1(datrdbpop)%>%filter(FOendDate==1)%>%
    mutate(vname=VSencryptedVesselCode,
           tname=FTunitName,
           spp=SAspeciesCodeFAO,
           len=FMclass,
           n=FMnumberAtUnit,
           FOid=FOunitName,
           ori="rdbes")
```

```
## Joining, by = "SAid"
```

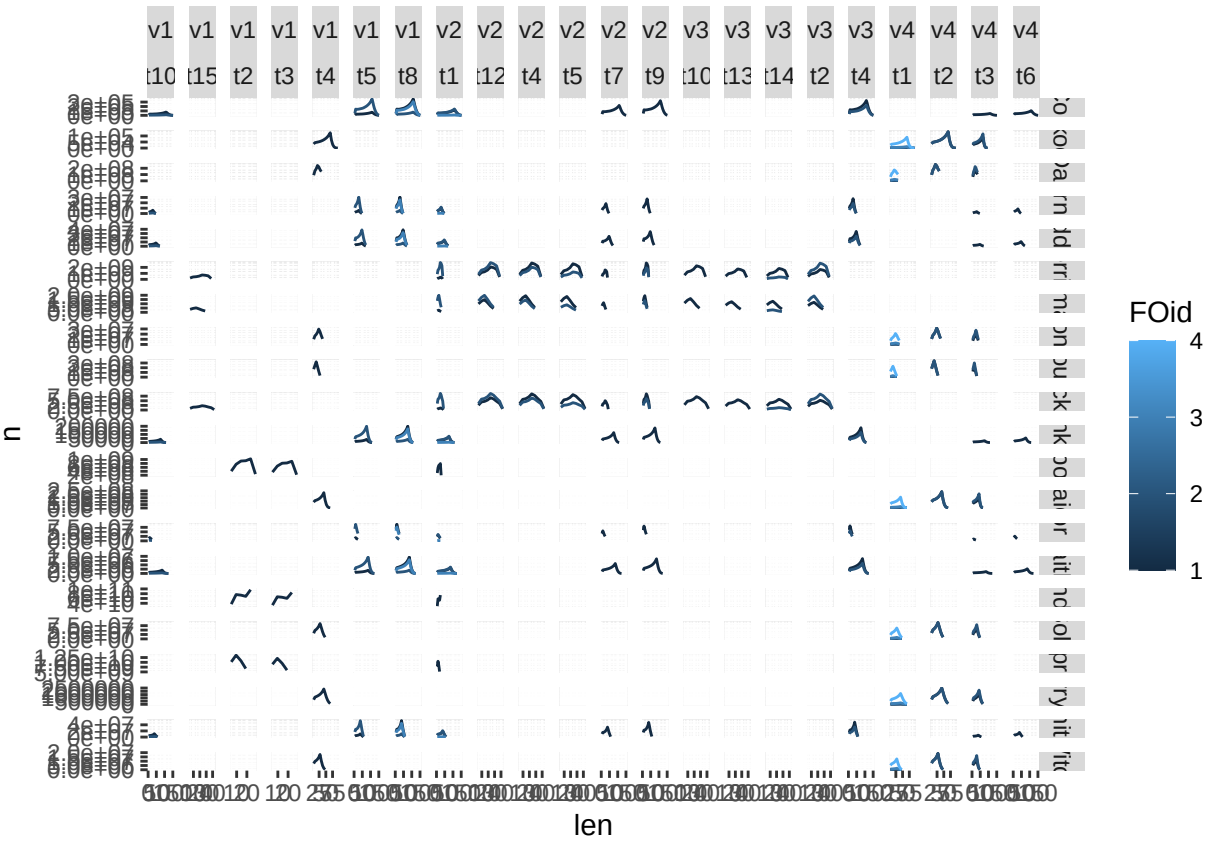
```
## Joining, by = "SSid"
```

```
## Joining, by = "FOid"
```

```
## Joining, by = "FTid"
```

```
## Joining, by = "VSid"
```

```
#some graph{{{
ggplot(tmp1, aes(x=len,y=n,group=FOid,color=FOid))+
    geom_path()+
    facet_grid(spp~vname+tname,scale="free")
```




```
#ggplot(tmp2, aes(x=len,y=n,group=FOid,color=FOid))+
#   geom_path()+
#   facet_grid(spp~vname+tname,scale="free")
#compute total by year gear spp
tmp1<-datsim%>%group_by(gear,spp,len)%>%summarise(n=sum(n),ori="ori")%>%ungroup()#filter(year==1)%>%mut
```

```
## `summarise()` has grouped output by 'gear', 'spp'. You can override using the `.groups` argument.
```

```
tmp2<-fct1(datrdbpop)%>%
    group_by(#year=FOendDate,
         gear=FOgear,
         spp=SAspeciesCodeFAO,
         len=FMclass)%>%summarise(n=sum(FMnumberAtUnit),ori="rdbes")%>%
    ungroup()
```

```
## Joining, by = "SAid"
```

```
## Joining, by = "SSid"
```

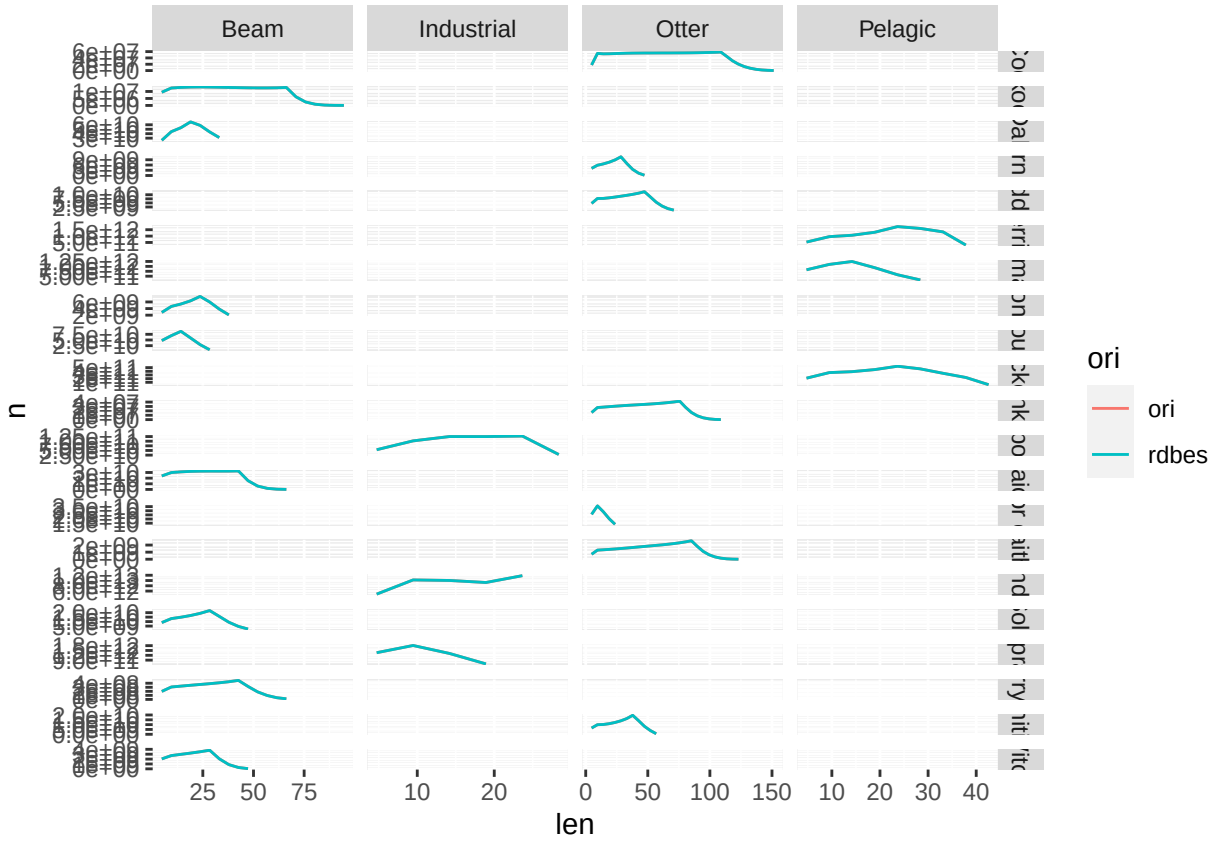
```
## Joining, by = "FOid"
```

```
## Joining, by = "FTid"
```

```
## Joining, by = "VSid"
```

```
## `summarise()` has grouped output by 'gear', 'spp'. You can override using the `.groups` argument.
```

```
ggplot(rbind(tmp1,tmp2), aes(x=len,y=n,group=ori,color=ori))+
    geom_path()+
    facet_grid(spp~gear,scale="free")
```




```
#end graph }}}
#seems fiiiiiiiiiiiine
```

Estimation

```
#test ratio estimators based on Liz code
#fct to calculte automatically incl prob whatever the hierarchy is
#input : rdbes data following david format
#hypothese of SRS
    #compute incp
    doincp<-function(b,a){
        #b<-datrdbpop$FO
        #a<-"FOinclusionProb"
        id<-(which(names(b)%in%a))
        if(length(id)==1){
            idtot<-which(substr(names(b),3,13)=="numberTotal"&nchar(names(b))<=13)
            idsamp<-which(substr(names(b),3,15)=="numberSampled"&nchar(names(b))<=15)
```

```
            b[,id]<-b[,idsamp]/b[,idtot]
            #print(id,idtot,idsamp)
        }
        return(b)
    }
    #inclusion proba for pop H1
    datrdbpop<-mapply(doincp,datrdbpop,
             paste0(names(datrdbpop),"inclusionProb"),
             SIMPLIFY=FALSE)
    #inclustion proba for samp H1
    datrdbsamp<-mapply(doincp,datrdbsamp,
             paste0(names(datrdbsamp),"inclusionProb"),
             SIMPLIFY=FALSE)
    #datrdbsamp$SA$SAinclusionProb<-datrdbsamp$SA$SAinclusionProb/10

# ------------------------------------------------------------------------
# a function to calculate inclusion probabilities for the units at the final stage
# of sampling given all the inclusion probabilities for the other stages
# ------------------------------------------------------------------------
getIncProb <- function(RDB,stages){#{{{
  nStages <- length(stages)
  if (any(stages %in% c("FM"))) {
    RDB[["FM"]][["FMinclusionProb"]] <- 1
  }
  RDB[[stages[[1]]]][["inclusionProb"]] <- RDB[[stages[[1]]]][[paste(stages[[1]],"inclusionProb",sep="")
  for (i in 2:(nStages)) {
    indx <- RDB[[stages[[i]]]][[paste(stages[[i-1]],"id",sep="")]]
    indxPrev <- RDB[[stages[[i-1]]]][[paste(stages[[i-1]],"id",sep="")]]
    RDB[[stages[[i]]]][["inclusionProbPrev"]] <- RDB[[stages[[i-1]]]][[paste("inclusionProb",sep="")]][n
    RDB[[stages[[i]]]][["inclusionProb"]] <- RDB[[stages[[i]]]][["inclusionProbPrev"]]*RDB[[stages[[i]]]
  }
  return(RDB)
}#}}}

#calcul incprob for H1 pop
stages<-list("VS","FT","FO","SS","SA","FM")
datrdbpop<-getIncProb(datrdbpop,stages)
#calcul incprob for H1 samp
stages<-list("VS","FT","FO","SS","SA","FM")
datrdbsamp<-getIncProb(datrdbsamp,stages)
#calcul incprob for H1 samp
stages<-list("FT","FO","SS","SA","FM")
datrdbsamp4ratio<-getIncProb(datrdbsamp,stages)


#add domain to FM and SA for pop data
#DOtime=temporal domain, DOtech=technical (metier), DOspp domain
#find the domain where they are: generic key
#DOtech and DOtime in FO, DOspp in SA
DO<-datrdbpop$FO%>%transmute(FOid,FTid,DOtech=FOgear,DOtime=FOendDate)%>%
    #add the key to SS
    left_join(datrdbpop$SS%>%transmute(SSid,FOid))%>%
    #add the key to SA
```

```
    left_join(datrdbpop$SA%>%transmute(SAid,SSid,DOspp=SAspeciesCodeFAO))%>%
    #add the key to FM
    left_join(datrdbpop$FM%>%transmute(FMid,SAid))
```

```
## Joining, by = "FOid"

## Joining, by = "SSid"

## Joining, by = "SAid"
```

```
#add domain to SA and FM
datrdbpop$FM<-left_join(datrdbpop$FM,DO)
```

```
## Joining, by = c("FMid", "SAid")
```

```
datrdbpop$SA<-left_join(datrdbpop$SA,DO%>%select(-FMid)%>%distinct())
```

```
## Joining, by = c("SAid", "SSid")
```

```
#same for samp data
DO<-datrdbsamp$FO%>%transmute(FOid,FTid,DOtech=FOgear,DOtime=FOendDate)%>%
    #add the key to SS
    left_join(datrdbsamp$SS%>%transmute(SSid,FOid))%>%
    #add the key to SA
    left_join(datrdbsamp$SA%>%transmute(SAid,SSid,DOspp=SAspeciesCodeFAO))%>%
    #add the key to FM
    left_join(datrdbsamp$FM%>%transmute(FMid,SAid))
```

```
## Joining, by = "FOid"

## Joining, by = "SSid"

## Joining, by = "SAid"
```

```
#add domain to SA and FM
datrdbsamp$FM<-left_join(datrdbsamp$FM,DO)
```

```
## Joining, by = c("FMid", "SAid")
```

```
datrdbsamp$SA<-left_join(datrdbsamp$SA,DO%>%select(-FMid)%>%distinct())
```

```
## Joining, by = c("SAid", "SSid")
```

```
#same for samp data4ratio
DO<-datrdbsamp4ratio$FO%>%transmute(FOid,FTid,DOtech=FOgear,DOtime=FOendDate)%>%
    #add the key to SS
    left_join(datrdbsamp4ratio$SS%>%transmute(SSid,FOid))%>%
    #add the key to SA
    left_join(datrdbsamp4ratio$SA%>%transmute(SAid,SSid,DOspp=SAspeciesCodeFAO))%>%
    #add the key to FM
    left_join(datrdbsamp4ratio$FM%>%transmute(FMid,SAid))
```

```
## Joining, by = "FOid"
```

```
## Joining, by = "SSid"
```

```
## Joining, by = "SAid"
```

```
#add domain to SA and FM
datrdbsamp4ratio$FM<-left_join(datrdbsamp4ratio$FM,DO)
```

```
## Joining, by = c("FMid", "SAid")
```

```
datrdbsamp4ratio$SA<-left_join(datrdbsamp4ratio$SA,DO%>%select(-FMid)%>%distinct())
```

```
## Joining, by = c("SAid", "SSid")
```

```
#n@len

# calculate a Horvitz Thompson estimate for total numbers at length by domain
# assuming srs within the domain - this is valid for the RDBshare data
#pop data
est4nlenpop <- datrdbpop$FM%>%group_by(len=FMclass,year=DOtime,gear=DOtech,spp=DOspp)%>%
    summarise(n=sum(FMnumberAtUnit/inclusionProb),type="est from pop")%>%
    ungroup()#%>%transmute
```

```
## `summarise()` has grouped output by 'len', 'year', 'gear'. You can override using the `.groups` argu
```

```
#samp data
est4nlensamp <- datrdbsamp$FM%>%group_by(len=FMclass,year=DOtime,gear=DOtech,spp=DOspp)%>%
    summarise(n=sum(FMnumberAtUnit/inclusionProb),type="est from samp")%>%
    ungroup()
```

```
## `summarise()` has grouped output by 'len', 'year', 'gear'. You can override using the `.groups` argu
```

```
#diag plot for ratio : plot n vs weight (and get the w samp for the ratio !)
pipo<-datrdbsamp4ratio
pipo1<-left_join(pipo$FM,
         pipo$SA%>%transmute(SAid,SSid,SAw=SAsampleWeightMeasured,SAp=inclusionProb))
```

```
## Joining, by = c("SAid", "SSid")
```

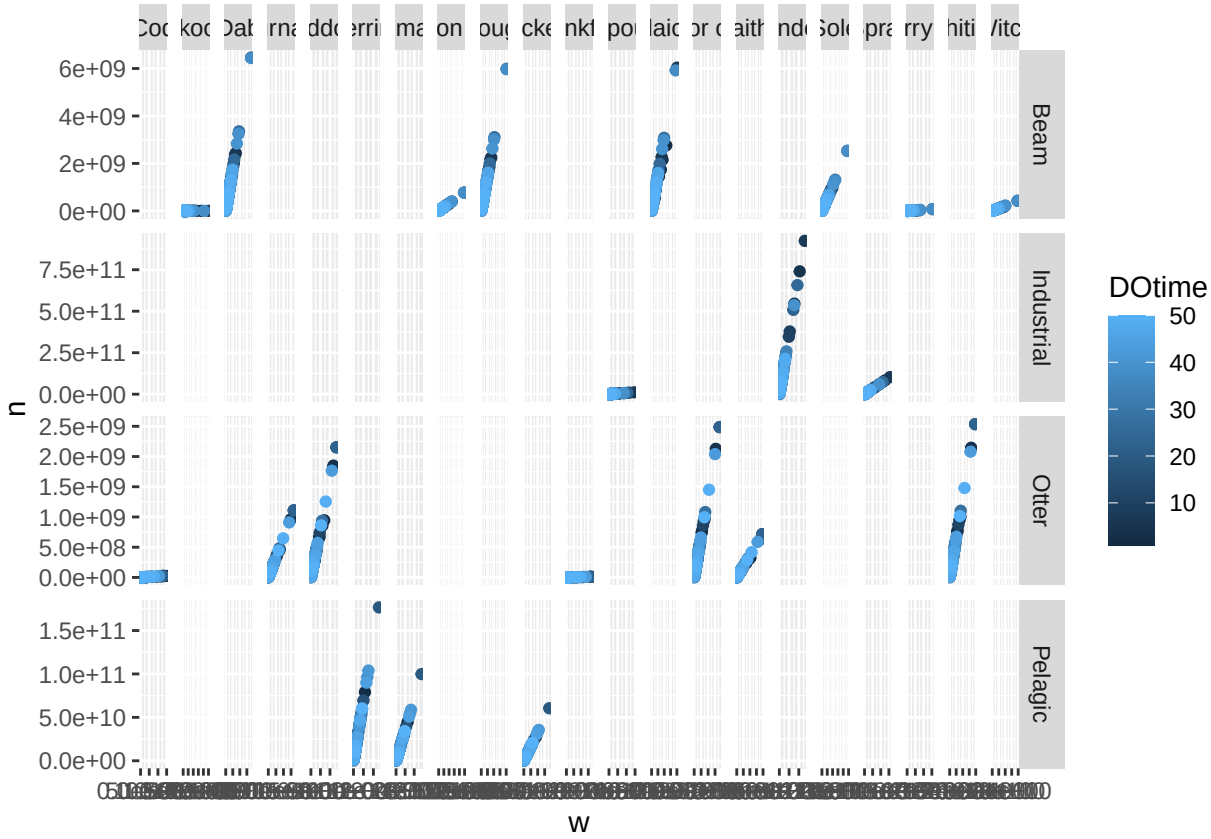
```
pipo2<-left_join(pipo1,pipo$SS%>%transmute(SSid,FOid,SSp=inclusionProb))
```

```
## Joining, by = c("FOid", "SSid")
```

```
uu1<-pipo2%>%
    group_by(FOid,DOspp,DOtime,DOtech)%>%summarise(w=unique(SAw),n=sum(FMnumberAtUnit))%>%
    ungroup()
```

```
## `summarise()` has grouped output by 'FOid', 'DOspp', 'DOtime'. You can override using the `.groups` a
```

```
ggplot(uu1,aes(x=w,y=n,color=DOtime,group=DOtime))+geom_point()+facet_grid(DOtech~DOspp,scale="free")
```




```
ggsave("rezcheckratio.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
#ratio estimation time
wsamp<-datrdbsamp4ratio$SA%>%
    group_by(year=DOtime,gear=DOtech,spp=DOspp)%>%
    summarise(w=sum(SAsampleWeightMeasured))%>%
    ungroup()
```

```
## `summarise()` has grouped output by 'year', 'gear'. You can override using the `.groups` argument.
```

```
wpop<-datsim%>%group_by(year,gear,spp)%>%
    summarise(wtot=sum(wspp))%>%ungroup()
```

```
## `summarise()` has grouped output by 'year', 'gear'. You can override using the `.groups` argument.
```

```
#samp data4ratio
est4nlensamp4ratio <- datrdbsamp4ratio$FM%>%
    #HT estimators from Trip to FM
    group_by(len=FMclass,year=DOtime,gear=DOtech,spp=DOspp)%>%
```

```
    #summarise(n=sum(FMnumberAtUnit/inclusionProb))%>%
    summarise(n=sum(FMnumberAtUnit))%>%
    #add sample weights and then pop w
    ungroup()%>%left_join(wsamp)%>%left_join(wpop)%>%
    #raise n@len using ratio estim
    mutate(n=n*wtot/w,type="ratio estim")%>%
    transmute(year,gear,spp,len,n,type)
```

```
## `summarise()` has grouped output by 'len', 'year', 'gear'. You can override using the `.groups` argu
```

```
## Joining, by = c("year", "gear", "spp")
## Joining, by = c("year", "gear", "spp")
```

```
#graph
tmp1<-datsim%>%group_by(year,gear,spp,len)%>%summarise(n=sum(n),type="pop")%>%ungroup()#filter(year==1)
```

```
## `summarise()` has grouped output by 'year', 'gear', 'spp'. You can override using the `.groups` argu
```

```
#numerical table
tmp1%>%select(-type)%>%left_join(est4nlenpop%>%transmute(year,gear,spp,len,npop=n))%>%
    left_join(est4nlensamp%>%transmute(year,gear,spp,len,nsamp=n))%>%
    left_join(est4nlensamp4ratio%>%transmute(year,gear,spp,len,nratio=n))%>%
    filter(year==1,gear=="Beam",spp=="Cuckoo ray")
```

```
## Joining, by = c("year", "gear", "spp", "len")
```

```
## Joining, by = c("year", "gear", "spp", "len")
## Joining, by = c("year", "gear", "spp", "len")
```

```
## # A tibble: 20 x 8
##     year gear  spp          len       n    npop   nsamp  nratio
##    <dbl> <chr> <chr>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1     1 Beam  Cuckoo ray  4.73 189070. 189070. 189228. 189070.
##  2     1 Beam  Cuckoo ray  9.46 250845. 250845. 251055. 250845.
##  3     1 Beam  Cuckoo ray 14.2  263071. 263071. 263291. 263071.
##  4     1 Beam  Cuckoo ray 18.9  275583. 275583. 275814. 275583.
##  5     1 Beam  Cuckoo ray 23.7  289443. 289443. 289685. 289443.
##  6     1 Beam  Cuckoo ray 28.4  306417. 306417. 306673. 306417.
##  7     1 Beam  Cuckoo ray 33.1  327681. 327681. 327955. 327681.
##  8     1 Beam  Cuckoo ray 37.8  353833. 353833. 354130. 353833.
##  9     1 Beam  Cuckoo ray 42.6  385561. 385561. 385884. 385561.
## 10     1 Beam  Cuckoo ray 47.3  424077. 424077. 424432. 424077.
## 11     1 Beam  Cuckoo ray 52.0  471402. 471402. 471796. 471402.
## 12     1 Beam  Cuckoo ray 56.8  530746. 530746. 531191. 530746.
## 13     1 Beam  Cuckoo ray 61.5  607272. 607272. 607780. 607272.
## 14     1 Beam  Cuckoo ray 66.2  709680. 709680. 710274. 709680.
## 15     1 Beam  Cuckoo ray 71.0  364887. 364887. 365193. 364887.
## 16     1 Beam  Cuckoo ray 75.7  156491. 156491. 156623. 156491.
## 17     1 Beam  Cuckoo ray 80.4   58580.  58580.  58629.  58580.
## 18     1 Beam  Cuckoo ray 85.1   18052.  18052.  18067.  18052.
## 19     1 Beam  Cuckoo ray 89.9    4083.   4083.   4086.   4083.
## 20     1 Beam  Cuckoo ray 94.6     460.    460.    461.    460.
```

```
#ggplot(rbind(tmp1,est4nlenpop), aes(x=len,y=n,color=year,group=year))+#group=year,color=type))+
p1<-ggplot(rbind(tmp1,est4nlenpop,est4nlensamp,est4nlensamp4ratio)%>%filter(year==1), aes(x=len,y=n,col
    geom_path()+
    facet_grid(spp~gear,scale="free")+
    ggtitle("year 1")
    theme_bw()
```

```
## List of 93
##  $ line                      :List of 6
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                      :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                     : NULL
##  $ aspect.ratio              : NULL
##  $ axis.title                : NULL
##  $ axis.title.x              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
```

```
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top          :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 2.75pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom       : NULL
##  $ axis.title.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 2.75pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 0pt 2.75pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
```

```
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 2.2pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 2.2pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
```

```
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 0pt 2.2pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                :List of 6
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : 'unit' num 2.75pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         :List of 5
##   ..$ fill         : NULL
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
```

```
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ legend.margin            : 'margin' num [1:4] 5.5pt 5.5pt 5.5pt 5.5pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ legend.spacing           : 'unit' num 11pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ legend.spacing.x         : NULL
##  $ legend.spacing.y         : NULL
##  $ legend.key               :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ legend.key.size          : 'unit' num 1.2lines
##   ..- attr(*, "valid.unit")= int 3
##   ..- attr(*, "unit")= chr "lines"
##  $ legend.key.height        : NULL
##  $ legend.key.width         : NULL
##  $ legend.text              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align        : NULL
##  $ legend.title             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align       : NULL
##  $ legend.position          : chr "right"
##  $ legend.direction         : NULL
##  $ legend.justification     : chr "center"
##  $ legend.box               : NULL
##  $ legend.box.just          : NULL
```

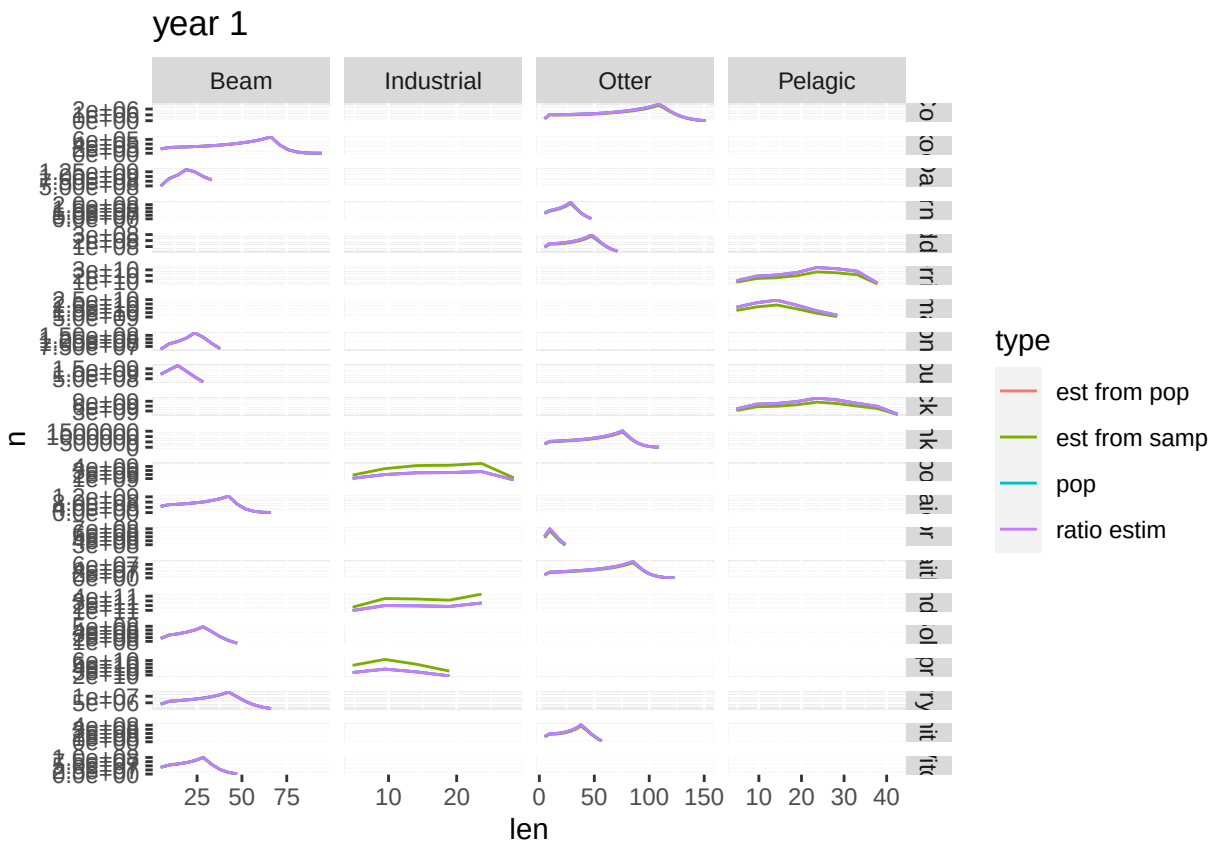
```
##  $ legend.box.margin         : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "valid.unit")= int 1
##   ..- attr(*, "unit")= chr "cm"
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : 'unit' num 11pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ panel.background          :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ panel.border              :List of 5
##   ..$ fill         : logi NA
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ panel.spacing             : 'unit' num 5.5pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           :List of 5
##   ..$ fill         : NULL
##   ..$ colour       : chr "white"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
```

```
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ plot.title               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 5.5pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 5.5pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 5.5pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
```

```
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : 'margin' num [1:4] 5.5pt 5.5pt 5.5pt 5.5pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ strip.background          :List of 5
##   ..$ fill         : chr "grey85"
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 4.4pt 4.4pt 4.4pt 4.4pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.switch.pad.grid     : 'unit' num 2.75pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
```

```
##  $ strip.switch.pad.wrap     : 'unit' num 2.75pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

```
print(p1)
```




```
ggsave("rezpopestimyear1.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
p1<-ggplot(rbind(tmp1,est4nlenpop,est4nlensamp,est4nlensamp4ratio)%>%filter(year==25), aes(x=len,y=n,co
    geom_path()+
    facet_grid(spp~gear,scale="free")+
    ggtitle("year 25")
    theme_bw()
```

```
## List of 93
##  $ line                      :List of 6
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                      :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                     : NULL
##  $ aspect.ratio              : NULL
##  $ axis.title                : NULL
##  $ axis.title.x              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
```

```
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top          :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 2.75pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom       : NULL
##  $ axis.title.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 2.75pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 0pt 2.75pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
```

```
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 2.2pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 2.2pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
```

```
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 0pt 2.2pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                :List of 6
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : 'unit' num 2.75pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         :List of 5
##   ..$ fill         : NULL
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
```

```
##  $ legend.margin            : 'margin' num [1:4] 5.5pt 5.5pt 5.5pt 5.5pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ legend.spacing           : 'unit' num 11pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ legend.spacing.x         : NULL
##  $ legend.spacing.y         : NULL
##  $ legend.key               :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ legend.key.size          : 'unit' num 1.2lines
##   ..- attr(*, "valid.unit")= int 3
##   ..- attr(*, "unit")= chr "lines"
##  $ legend.key.height        : NULL
##  $ legend.key.width         : NULL
##  $ legend.text              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align        : NULL
##  $ legend.title             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align       : NULL
##  $ legend.position          : chr "right"
##  $ legend.direction         : NULL
##  $ legend.justification     : chr "center"
##  $ legend.box               : NULL
##  $ legend.box.just          : NULL
##  $ legend.box.margin        : 'margin' num [1:4] 0cm 0cm 0cm 0cm
```

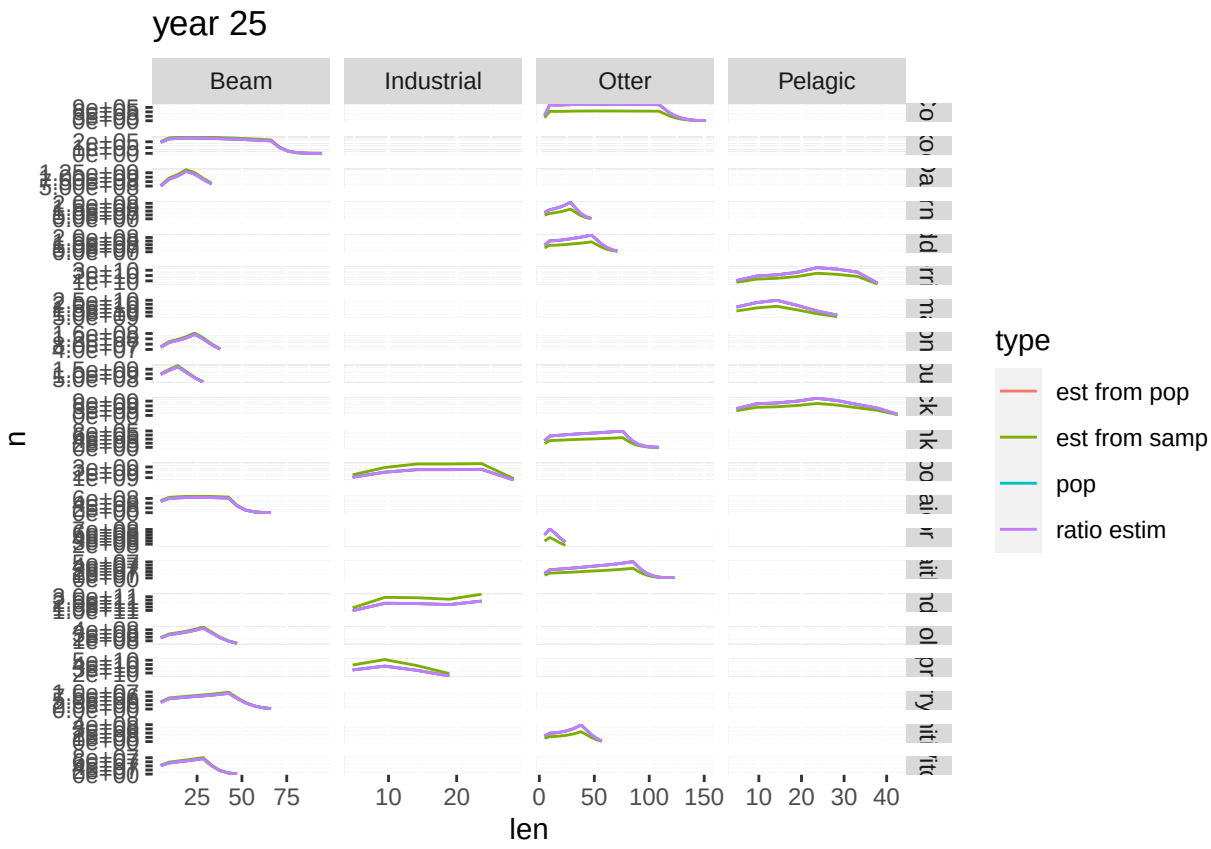
```
##   ..- attr(*, "valid.unit")= int 1
##   ..- attr(*, "unit")= chr "cm"
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : 'unit' num 11pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ panel.background          :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ panel.border              :List of 5
##   ..$ fill         : logi NA
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ panel.spacing             : 'unit' num 5.5pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           :List of 5
##   ..$ fill         : NULL
##   ..$ colour       : chr "white"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
```

```
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ plot.title               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 5.5pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 5.5pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 5.5pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0.5
```

```
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : 'margin' num [1:4] 5.5pt 5.5pt 5.5pt 5.5pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ strip.background          :List of 5
##   ..$ fill         : chr "grey85"
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 4.4pt 4.4pt 4.4pt 4.4pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.switch.pad.grid     : 'unit' num 2.75pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ strip.switch.pad.wrap     : 'unit' num 2.75pt
```

```
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

```
print(p1)
```



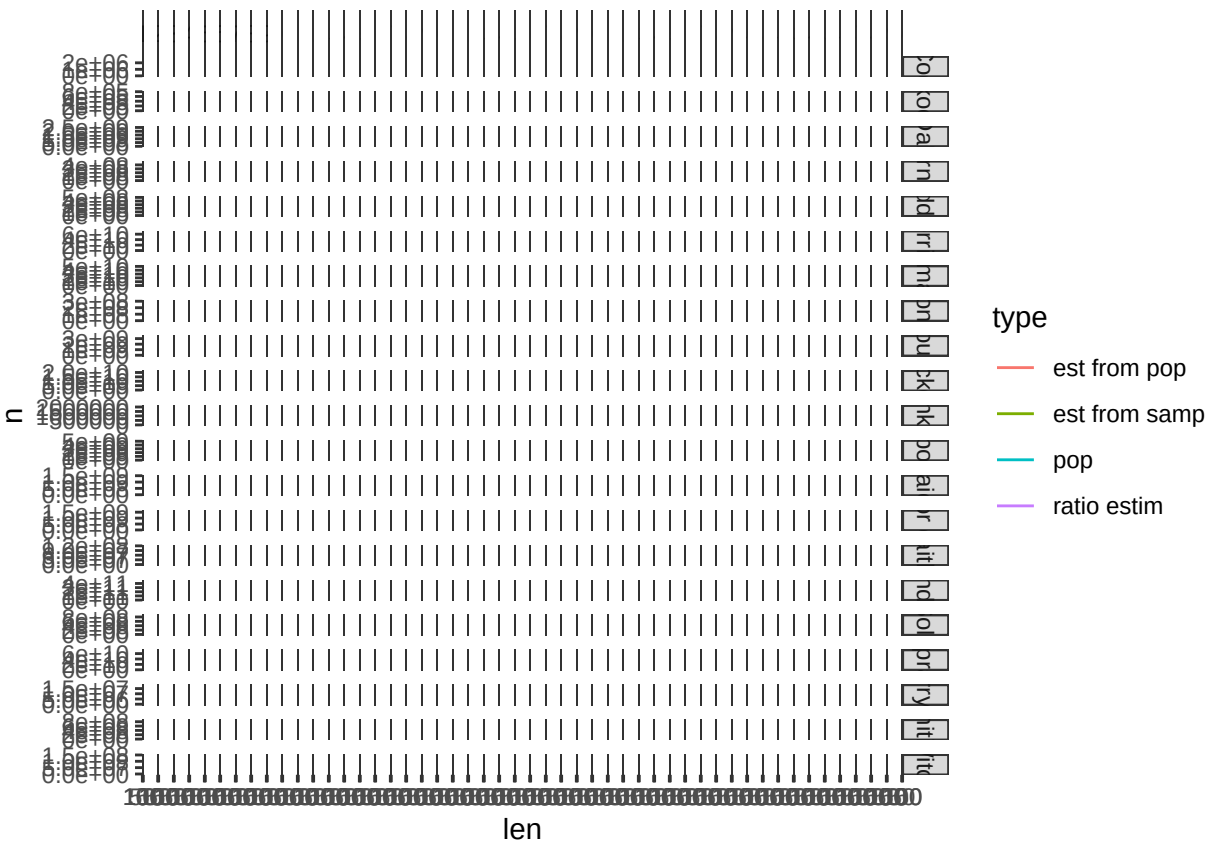

```
ggsave("rezpopestimyear25.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
pipo<-rbind(tmp1,est4nlenpop,est4nlensamp,est4nlensamp4ratio)%>%
    group_by(year,spp,type,len)%>%summarise(n=sum(n))%>%ungroup()
```

```
## `summarise()` has grouped output by 'year', 'spp', 'type'. You can override using the `.groups` argu
```

```
ggplot(pipo, aes(x=len,y=n,color=type,group=type))+#group=year,color=type))+
    geom_path()+
    facet_grid(spp~year,scale="free")+
    theme_bw()
```




```
#print(p1)
#ggsave("rezpopestim1.png")
```

R session information

```
sessioninfo::session_info()
```

```
## - Session info -------------------------------------------------------------------------------------
##  setting  value
##  version  R version 3.6.3 (2020-02-29)
##  os       Ubuntu 18.04.5 LTS
##  system   x86_64, linux-gnu
```

```
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       Europe/Paris
##  date     2021-06-07
##
## - Packages -------------------------------------------------------------------------------------
##  package     * version date       lib source
##  assertthat    0.2.1   2019-03-21 [1] CRAN (R 3.6.0)
##  cli           2.3.1   2021-02-23 [1] CRAN (R 3.6.3)
##  codetools     0.2-16  2018-12-24 [4] CRAN (R 3.6.3)
##  colorout    * 1.2-1   2019-04-30 [1] local
##  colorspace    2.0-0   2020-11-11 [1] CRAN (R 3.6.3)
##  crayon        1.4.1   2021-02-08 [1] CRAN (R 3.6.3)
##  DBI           1.1.0   2019-12-15 [1] CRAN (R 3.6.1)
##  debugme       1.1.0   2017-10-22 [1] CRAN (R 3.6.3)
##  digest        0.6.27  2020-10-24 [1] CRAN (R 3.6.3)
##  dplyr       * 1.0.5   2021-03-05 [1] CRAN (R 3.6.3)
##  drat          0.1.5   2019-03-28 [1] CRAN (R 3.6.0)
##  ellipsis      0.3.1   2020-05-15 [1] CRAN (R 3.6.3)
##  evaluate      0.14    2019-05-28 [1] CRAN (R 3.6.1)
##  fansi         0.4.2   2021-01-15 [1] CRAN (R 3.6.3)
##  farver        2.1.0   2021-02-28 [1] CRAN (R 3.6.3)
##  generics      0.1.0   2020-10-31 [1] CRAN (R 3.6.3)
##  ggplot2     * 3.3.3   2020-12-30 [1] CRAN (R 3.6.3)
##  glue          1.4.2   2020-08-27 [1] CRAN (R 3.6.3)
##  gtable        0.3.0   2019-03-25 [1] CRAN (R 3.6.0)
##  highr         0.8     2019-03-20 [1] CRAN (R 3.6.0)
##  htmltools     0.5.0   2020-06-16 [1] CRAN (R 3.6.3)
##  knitr         1.31    2021-01-27 [1] CRAN (R 3.6.3)
##  labeling      0.4.2   2020-10-20 [1] CRAN (R 3.6.3)
##  lifecycle     1.0.0   2021-02-15 [1] CRAN (R 3.6.3)
##  magrittr      2.0.1   2020-11-17 [1] CRAN (R 3.6.3)
##  munsell       0.5.0   2018-06-12 [1] CRAN (R 3.6.0)
##  nvimcom     * 0.9-75  2018-11-26 [1] local
##  pillar        1.5.1   2021-03-05 [1] CRAN (R 3.6.3)
##  pkgconfig     2.0.3   2019-09-22 [1] CRAN (R 3.6.1)
##  ps            1.4.0   2020-10-07 [1] CRAN (R 3.6.3)
##  purrr         0.3.4   2020-04-17 [1] CRAN (R 3.6.3)
##  R6            2.5.0   2020-10-28 [1] CRAN (R 3.6.3)
##  rlang         0.4.10  2020-12-30 [1] CRAN (R 3.6.3)
##  rmarkdown   * 2.7     2021-02-19 [1] CRAN (R 3.6.3)
##  rstudioapi    0.13    2020-11-12 [1] CRAN (R 3.6.3)
##  scales        1.1.1   2020-05-11 [1] CRAN (R 3.6.3)
##  sessioninfo   1.1.1   2018-11-05 [1] CRAN (R 3.6.3)
##  setwidth    * 1.0-4   2015-07-07 [1] Github (cran/setwidth@42b61c4)
##  stringi       1.5.3   2020-09-09 [1] CRAN (R 3.6.3)
##  stringr       1.4.0   2019-02-10 [1] CRAN (R 3.6.0)
##  tibble        3.1.0   2021-02-25 [1] CRAN (R 3.6.3)
##  tidyselect    1.1.0   2020-05-11 [1] CRAN (R 3.6.3)
##  utf8          1.2.1   2021-03-12 [1] CRAN (R 3.6.3)
##  vctrs         0.3.6   2020-12-17 [1] CRAN (R 3.6.3)
##  vimcom      * 1.3-1   2019-04-30 [1] local
```

```
##  withr         2.4.1   2021-01-26 [1] CRAN (R 3.6.3)
##  xfun          0.21    2021-02-10 [1] CRAN (R 3.6.3)
##  yaml          2.2.1   2020-02-01 [1] CRAN (R 3.6.3)
##
## [1] /home/moi/R/x86_64-pc-linux-gnu-library/3.6
## [2] /usr/local/lib/R/site-library
## [3] /usr/lib/R/site-library
## [4] /usr/lib/R/library
```

References

calculate a HT estimate for total landed weight using the sampled landed

weights

assuming srs etc as before

lansim<-datsim%>%group_by(year,gear,spp)%>% summarise(w=sum(wspp),type="sim")%>%ungroup()
lanpop<- datrdbpopSASA%>%group_by(year=DOtime,gear=DOtech,spp=DOspp)%>% summarise(w=sum(SAtotalWeightM
from samp")%>% ungroup()#%>%transmute lansampratio<- datrdbsamp4ratio$SA%>%group_by(year=DOtime,gear=DOte
summarise(w0=sum(SAtotalWeightMeasured/inclusionProb))%>% ungroup()%>%left_join(wsamp)%>%left_join(wpop)%>%
mutate(w=w0*wtot/w,type="ratio estim")%>% select(-wtot,-w0)%>% ungroup()#%>%transmute

lanall<-rbind(lansim,lanpop,lansamp,lansampratio)

p1<-ggplot(lanall, aes(x=year,y=w,color=type,group=type))+#group=year,color=type))+ geom_path()+
facet_grid(spp~gear,scale="free")+ theme_bw() print(p1) ggsave("rezlan1.png") p1<-ggplot(lanall%>%group_by(year,gear,ty
aes(x=year,y=w,color=type,group=type))+#group=year,color=type))+ geom_path()+ facet_wrap(~type)+#,scale="free")+
theme_bw() print(p1) ggsave("rezlan2.png")

calculate the ratio estimates for numbers at length

estLR <- estL*matrix(rep(popX/estX,dim(estL)[1]),byrow=T,ncol=dim(estL)[2])

#naive (and probably wrong) ratio estimator #pop data wtot<-clrdb%>% group_by(time=CLyear,metier=CLmetier6,space="
summarise(w=sum(CLoffWeight))%>% ungroup() #samp data FOinfo<-datrdbFOSS%>%transmute(FOid,SSid)%>%distinct
SAinfo<-datrdbSAFM%>%filter(FMtype=="l")%>% transmute(FMid,SAid,len=FMclass,n=FMnumberAtUnit)
#agg length tot wsamp<-SAinfo%>%left_join(SSinfo)%>%left_join(FOinfo)%>% group_by(time,metier,space,spp)%>%
summarise(wsamp=sum(wsamp),wsamptot=sum(wsamptot))%>% ungroup()

nsamp<-FMinfo%>%left_join(SAinfo)%>%left_join(SSinfo)%>%left_join(FOinfo)%>% ungroup()%>%
group_by(time,metier,space,spp,len)%>% summarise(n=sum(n))%>% ungroup() %>%left_join(wsamp)
#merge nsamp and wtot nsamp<-left_join(nsamp,wtot)%>%mutate(npop=n*w/wsamptot)

ggplot(nsamp,aes(x=len,y=npop))+geom_path()+facet_grid(metier~spp,scale="free") #load simulated data
uu<-readRDS("../outputs/datapop.rds") npg<-readRDS("../outputs/datapopn.rds")%>%filter(year==1,value>0)%>%transm
time=year, space="all", spp,len, n=value,type="ori") npgestim<-nsamp%>%transmute(metier,time,space,spp,len,n=npop,type
pipo<-rbind(npg,npgestim) ggplot(pipo,aes(x=len,y=n,color=type,group=type))+geom_path()+facet_wrap(metier~spp,scale=

#ggplot(pipo%>%filter(spp=="Dab"),aes(x=len,y=n,color=type,group=type))+geom_path()+facet_grid(metier$_{spp,scale="free}$

lanall<-rbind(lansim,lanpop,lansamp)

p1<-ggplot(lanall, aes(x=year,y=w,color=type,group=type))+#group=year,color=type))+ geom_path()+
facet_grid(spp~gear,scale="free")+ theme_bw() print(p1) ggsave("rezlan1.png") p1<-ggplot(lanall%>%group_by(year,gear,ty
aes(x=year,y=w,color=type,group=type))+#group=year,color=type))+ geom_path()+ facet_wrap(~type)+#,scale="free")+
theme_bw() print(p1) ggsave("rezlan2.png")

calculate the ratio estimates for numbers at length

estLR <- estL*matrix(rep(popX/estX,dim(estL)[1]),byrow=T,ncol=dim(estL)[2])

#naive (and probably wrong) ratio estimator #pop data wtot<-clrdb%>% group_by(time=CLyear,metier=CLmetier6,space="
summarise(w=sum(CLoffWeight))%>% ungroup() #samp data FOinfo<-datrdbFOSS%>%transmute(FOid,SSid)%>%distinct
SAinfo<-datrdbSAFM%>%filter(FMtype=="l")%>% transmute(FMid,SAid,len=FMclass,n=FMnumberAtUnit)
#agg length tot wsamp<-SAinfo%>%left_join(SSinfo)%>%left_join(FOinfo)%>% group_by(time,metier,space,spp)%>%
summarise(wsamp=sum(wsamp),wsamptot=sum(wsamptot))%>% ungroup()

nsamp<-FMinfo%>%left_join(SAinfo)%>%left_join(SSinfo)%>%left_join(FOinfo)%>% ungroup()%>%
group_by(time,metier,space,spp,len)%>% summarise(n=sum(n))%>% ungroup() %>%left_join(wsamp)
#merge nsamp and wtot nsamp<-left_join(nsamp,wtot)%>%mutate(npop=n*w/wsamptot)

ggplot(nsamp,aes(x=len,y=npop))+geom_path()+facet_grid(metier~spp,scale="free") #load simulated data
uu<-readRDS("../outputs/datapop.rds") npg<-readRDS("../outputs/datapopn.rds")%>%filter(year==1,value>0)%>%transm
time=year, space="all", spp,len, n=value,type="ori") npgestim<-nsamp%>%transmute(metier,time,space,spp,len,n=npop,typ
pipo<-rbind(npg,npgestim) ggplot(pipo,aes(x=len,y=n,color=type,group=type))+geom_path()+facet_wrap(metier~spp,scale

#ggplot(pipo%>%filter(spp=="Dab"),aes(x=len,y=n,color=type,group=type))+geom_path()+facet_grid(metier$_{spp,scale="fre}$