Get and summarize AFSC survey data

Kirstin Holsman, Alaska Fisheries Science Center

#### [**AFSC Survey CPUE data: github.com/kholsman/AFSC\_CPUE**](https://github.com/kholsman/AFSC_CPUE)

Repo maintained by:  
Kirstin Holsman  
Alaska Fisheries Science Center  
NOAA Fisheries, Seattle WA  
[**kirstin.holsman@noaa.gov**](kirstin.holsman@noaa.gov)  
*Last updated: Jul 13, 2023*

# Overview

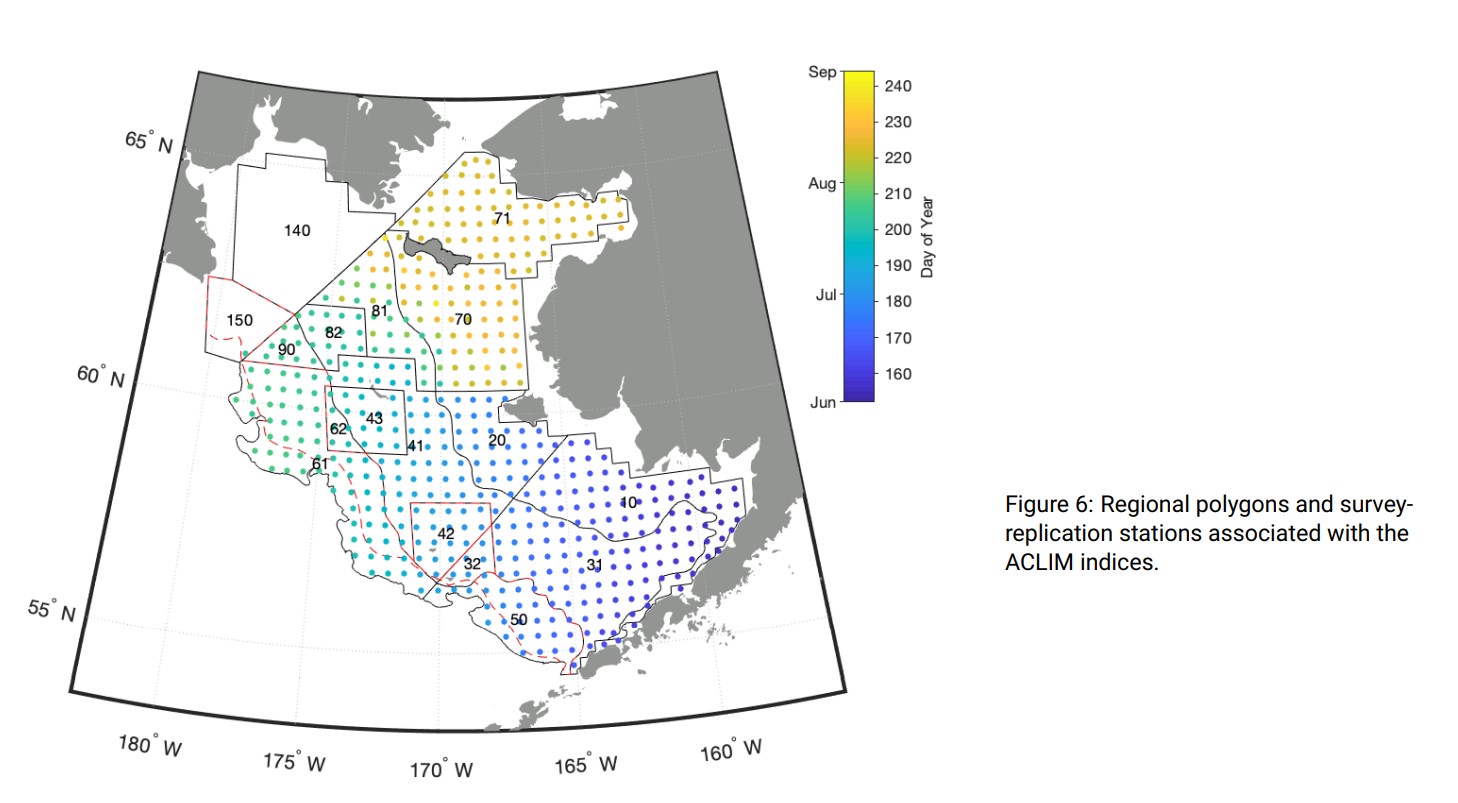
The below scripts return a list object cpue\_data saved as a compressed Rdata file with the naming ‘reg.srvy#.spp.cpue\_data.Rdata’ such as “ebs.srvy98.plk.cpue\_data.Rdata”. Each cpue\_data list contains 8 data.frames:

load(paste0("data/out/",qrydate,"/cpue/ebs/ebs.srvy98.plk.cpue\_data.Rdata"))  
  
names(cpue\_data)

There is a folder for each region “ebs”, “goa”, “ai”. For the “ebs” (Bering Sea) there are two sets of cpue\_data, one that is NEBS+SEBS combined (‘ebs.srvy98.[sp].cpue\_data.Rdata’) and one that is just SEBS survey areas (‘sebs.srvy98.[sp].cpue\_data.Rdata’). For both the Gulf of Alaska (“goa”) and the Bering Sea, mean CPUE (Kg per km2 or Number per km2) for each size bin at each strata was calculated and then multiplied by the STRATA area to get total Biomass and abundance. **Note:Since strata area estimates where not available for the Aleutian Island (“ai”) or slope surveys (“slope”) these AREA was set equal to 1 and the Total Biomass and abundance is actually the sum of mean biomass.**

The data.frames within each cpue\_data object are:

1. **totalB\_N**: Total biomass (kg) or abundance (# of fish) for the species in each year
2. **mnCPUE\_strata\_yr** : Average survey CPUE (kg per Km2) or abundance (# per Km2) for the species in each strata and year
3. **total\_bin\_B\_N**: Total biomass (kg) or abundance (# of fish) for each bin (10 mm) for the species in each year
4. **mnCPUE\_strata\_bin\_yr** : Average survey CPUE (kg per Km2) or abundance (# per Km2) for each size bin for the species in each strata and year
5. **CPUE\_station\_bin\_yr**: Station specific survey CPUE (kg per Km2) or abundance (# per Km2) for each size bin for the species in each year
6. **CPUE\_station\_yr**: Station specific survey CPUE (kg per Km2) or abundance (# per Km2) for the species in each year
7. **propByBin**: proportion of biomass in each size bin per species per year
8. **propByStrata**: proportion of biomass in each strata per species per year
9. **propByStrataBin**: proportion of biomass in each bin and strata per species per year

 These are calculated from the RACEBASE data tables for survey results where total CPUE was recorded for the species (location\_catch) at each haul, expanded to include stations where CPUE=0 (location) and expanded to each size bin using the proportional subset of frequency of fish of given length (mm), binned into 10 mm bins () and predicted weight () for each size bin at each station :

where is the area of the strata in and is the strata specific average CPUE (kg per or number per ) of all stations in strata :

where is the station specific CPUE (saved as the object cpue\_data$CPUE\_station\_yr).

To obtain population level estimates of the biomass or abundance of fish by size bin , we used a length weight regression to estimate the weight of each size fish measured () to calculate the proportion by weight or frequency at each station where

where and are fit (updated each year) to all available length and weight data from surveys from all years across the EBS, GOA, and AI (rather than each basin separately) and

and

This was then multiplied by the CPUE at each station () to obtain a station estimate of CPUE by size bin :

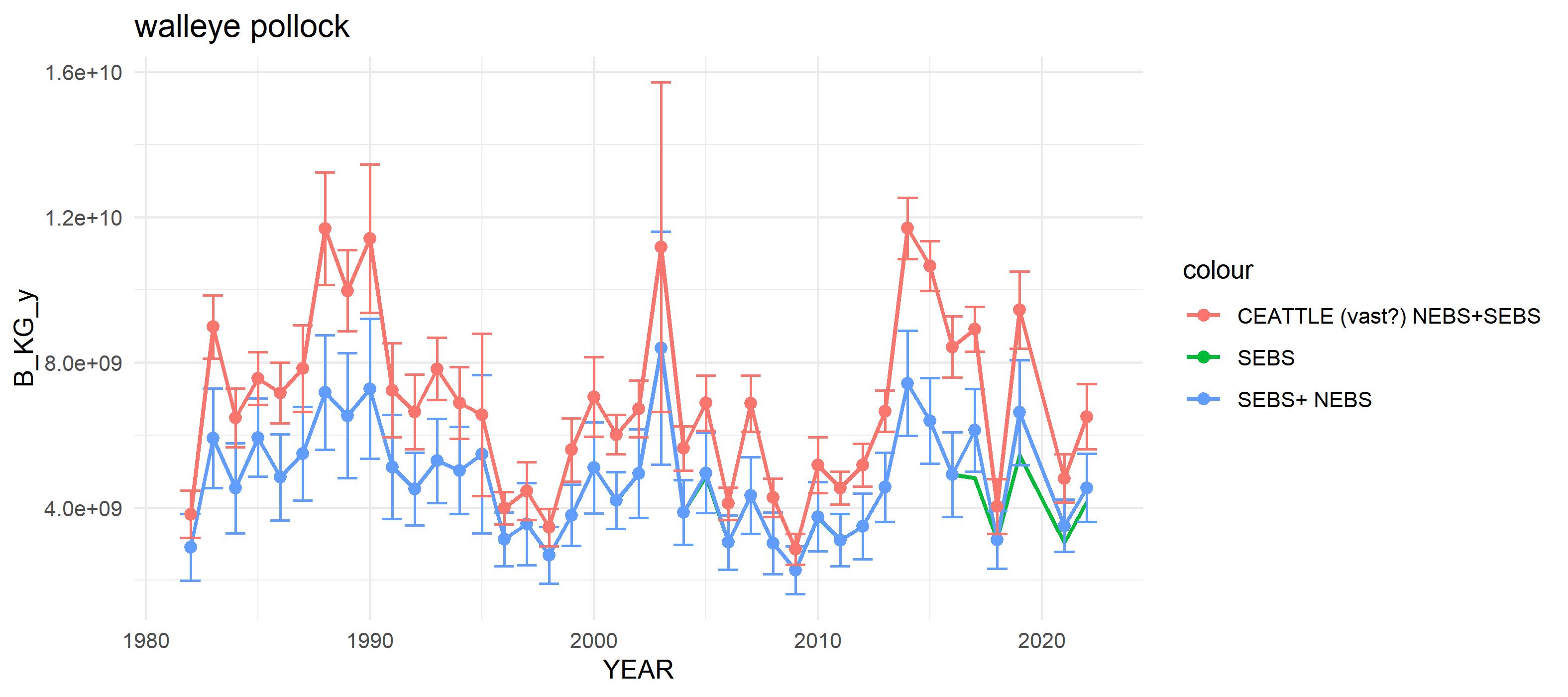
Finally, the average strata CPUE () was calculated as:

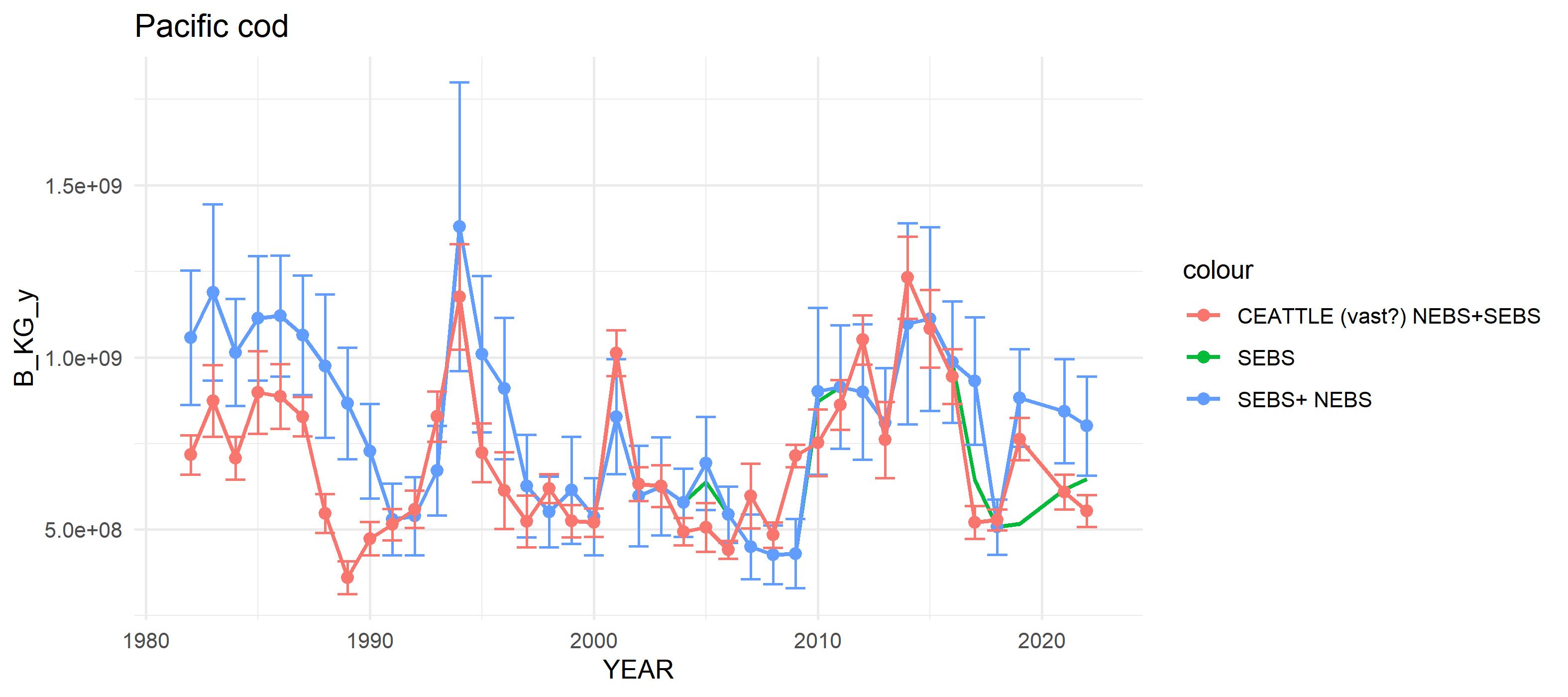
and strata- and bin-specific biomass (Kg) was calculated the product of mean CPUE (Kg per Km^2) and strata area (Km^2):

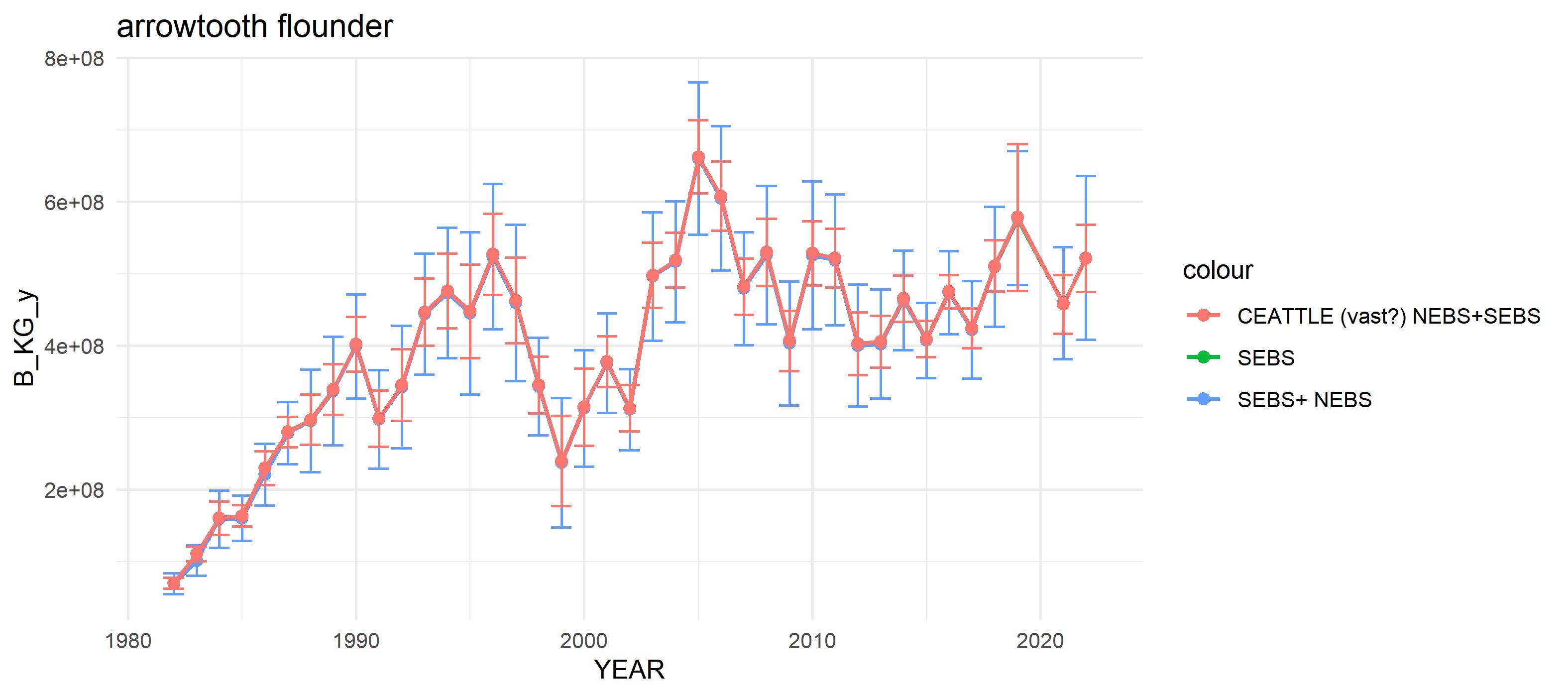
and total annual biomass by bin is the sum of strata-specific biomass for each species:

and total annual biomass is:

# Comparison of Bering Sea survey values







“arrowtooth”

# Code

# ## Step 0: Set up the R workspace  
#   
# The first step is to set up the switches for what files to update and create in the file `R/setup.R`. The code below then loads these settings as well as base data, functions, and packages.   
#   
# ## Step 1: Update SQL queries  
# This step must be run on a computer that has access to RACEBASE. The code below will generate the base files for steps 2 and 3 below,and will save them in the folder `r data.path ` under subfolders for each region in `srvys$reg` and each species in `splist` (see `R/setup.R` to change these settings).  
  
# \*\*IMPORTANT:\*\*   
#   
# \* \*\*This step must be connected to the RACEBASE SQL database\*\*  
#   
# \* \*\*To change R studio from the default 64 bit to 32 bit go to Tools>Global options and select the 32 bit version of R.\*\*   
#   
# \* \*\*The code will connect to the SQL database using your password and username. Remember to update the `username\_path` in the first line of the `R/setup.R ` file and corresponding `username` and `password` under `username\_password.R`. A template is available under `R/`.\*\*  
#   
# <!-- ![Header of `setup.R` where `username\_path` can be adjusted. This file also is where species, regions, and bins are specified.](figs/setup.jpg){width=80%} -->  
#   
   
   
 # get everything set up:  
 #----------------------------------------  
 # rm(list=ls())  
 # this uses the password saved in R/password.R  
 suppressMessages(source("R/make.R"))  
  
 # update the SQL queries  
 #---------------------------------------------   
  
 source(file.path(code.path,"R/sub\_scripts/runRACE\_qrys.R"))  
  
 # combine sebs and nebs into one region: ebs  
 if(dir.exists(file.path(data.path,"ebs")))  
 system(paste("rm -r",file.path(data.path,"ebs")))  
 dir.create(file.path(data.path,"ebs"))  
   
 # combine files and rename survey area to all of EBS  
 for(sp in names(splist)){  
 if(dir.exists(file.path(data.path,"ebs",sp)))  
 system(paste("rm -r",file.path(data.path,"ebs",sp)))  
 dir.create(file.path(data.path,"ebs",sp))  
 #"length.Rdata"   
 load(file.path(data.path,"nebs",sp,"length.Rdata"))  
 length\_nebs <- length;rm(length)  
 load(file.path(data.path,"sebs",sp,"length.Rdata"))  
 length\_sebs <- length;rm(length)  
 length<- rbind(length\_nebs%>%  
 mutate(SURVEY\_DEFINITION\_ID\_aka =SURVEY\_DEFINITION\_ID,SURVEY\_DEFINITION\_ID =98),  
 length\_sebs%>%  
 mutate(SURVEY\_DEFINITION\_ID\_aka =SURVEY\_DEFINITION\_ID,SURVEY\_DEFINITION\_ID =98))  
   
 save(length,file = file.path(data.path,"ebs",sp,"length.Rdata"))  
 rm(length)  
   
 #"location.Rdata"   
 load(file.path(data.path,"nebs",sp,"location.Rdata"))  
 location\_nebs <- location;rm(location)  
 load(file.path(data.path,"sebs",sp,"location.Rdata"))  
 location\_sebs <- location;rm(location)  
 location<- rbind(location\_nebs, location\_sebs)  
   
 save(location,file = file.path(data.path,"ebs",sp,"location.Rdata"))  
   
   
 #"location\_catch.Rdata"  
 load(file.path(data.path,"nebs",sp,"location\_catch.Rdata"))  
 location\_catch\_nebs <- location\_catch;rm(location\_catch)  
 load(file.path(data.path,"sebs",sp,"location\_catch.Rdata"))  
 location\_catch\_sebs <- location\_catch;rm(location\_catch)  
 location\_catch <- rbind(location\_catch\_nebs%>%  
 mutate(SURVEY\_DEFINITION\_ID\_aka =SURVEY\_DEFINITION\_ID,SURVEY\_DEFINITION\_ID =98),  
 location\_catch\_sebs%>%  
 mutate(SURVEY\_DEFINITION\_ID\_aka =SURVEY\_DEFINITION\_ID,SURVEY\_DEFINITION\_ID =98))  
   
 save(location\_catch,file = file.path(data.path,"ebs",sp,"location\_catch.Rdata"))  
   
 }  
  
   
#```  
  
  
## Step 2: Update the LWA regressions  
  
# The default code for RACEBASE uses set LW relationships, however we prefer to update the LW regressions using glms. Depending on how many observations exist the LW relationships can be region specific or use data across all regions.The default below is all regions combined. This code generates two outputs in `r data.out`, `r LWname` and `LW\_SmryTable.Rdata`. It also updates the `species\_lkup$LW\_a` and species\_lkup$LW\_b` parms used in Step 3.  
#   
# #```{r updateLWglms, echo=TRUE, eval=FALSE}   
  
 # update the LW regressions   
 #---------------------------------------------   
  
 if(update\_LWdata){   
 source(file.path(code.path,"R/sub\_scripts/updateLW.R"))  
 # reload with updated data:  
 source(file.path(code.path,"R/load\_data.R"))  
 }  
 species\_lkup  
  
#```  
  
  
## Step 3: Get CPUE data from the surveys  
  
#This code is the core script for generating the CPUE\_NUMKM2 and CPUE\_BIOMKM2 values by size bin, region, and species.   
  
#```{r updateCPUE, echo=TRUE, eval=FALSE}   
   
 STRATA\_AREA%>%filter(REGION=="BS")%>%  
 group\_by(YEAR)%>%summarise(mnAREA = mean(AREA, na.rm=T),  
 sumAREA = sum(AREA, na.rm=T),  
 cnt = length(unique(STRATUM)))  
   
 STRATA\_AREA%>%filter(REGION=="GOA")%>%  
 group\_by(YEAR)%>%summarise(mnAREA = mean(AREA, na.rm=T),  
 sumAREA = sum(AREA, na.rm=T),  
 cnt = length(unique(STRATUM)))  
   
 STRATA\_AREA%>%filter(REGION=="BS",YEAR==2022)%>%select(STRATUM)  
  
 # overwrite the NEBS frame from setup for the next set of code (ebs = sebs+nebs now forward)  
 srvys <- data.frame(reg=c("ebs","goa","ai","slope"),RGN = c("BS","GOA","AI","SLOPE"), num=c(98,47,52,78) )  
 # srvys <- data.frame(reg=c("ebs","goa","ai"),RGN = c("BS","GOA","AI"), num=c(98,47,52) )  
 nreg <- length(srvys$reg)  
 nspp <- length(species\_lkup$sp)  
   
 for (r in 1:nreg){  
 for(s in 1:nspp){  
   
 if(srvys[r,]$reg =="ebs"){  
 # first SEBS only:  
 # -------------------------------  
 STRATA\_AREAUSE <- STRATA\_AREA%>%filter(REGION==srvys$RGN[r])  
 maxyr <- max(STRATA\_AREAUSE$YEAR)  
 STRATA\_AREAUSE <- STRATA\_AREAUSE%>%  
 filter(YEAR==2022)%>%  
 group\_by(REGION,STRATUM)%>%  
 summarize(AREA = mean(AREA, na.rm=T))%>%ungroup()  
   
 flnm <- paste0("s",srvys[r,]$reg,".srvy",  
 srvys[r,]$num,".",  
 species\_lkup[s,]$sp)  
 cat("now getting data for: ",flnm,"\n")  
 cpue\_data <- suppressMessages(  
 get\_CPUE\_DATA(  
 datapath = data.path,  
 out\_dir = file.path(data.out),  
 STRATA\_AREAIN = STRATA\_AREAUSE,  
 flnm = flnm,  
 species = species\_lkup[s,]$SPECIES\_CODE,  
 survey = srvys[r,]$num,  
 includeNBS = FALSE,  
 NEBSStrataIN = NEBS\_strata ,  
 saveit = T,  
 bins = sp\_bins[[ species\_lkup[s,]$sp ]]))  
   
 rm(cpue\_data)  
   
 # Now NESB + SEBS  
 # -------------------------------  
 flnm <- paste0(srvys[r,]$reg,".srvy",  
 srvys[r,]$num,".",  
 species\_lkup[s,]$sp)  
 cat("now getting data for: ",flnm,"\n")  
 cpue\_data <- suppressMessages(  
 get\_CPUE\_DATA(  
 datapath = data.path,  
 out\_dir = file.path(data.out),  
 STRATA\_AREAIN = STRATA\_AREAUSE,  
 flnm = flnm,  
 species = species\_lkup[s,]$SPECIES\_CODE,  
 survey = srvys[r,]$num,  
 includeNBS = TRUE,  
 NEBSStrataIN = NEBS\_strata ,  
 saveit = T,  
 bins = sp\_bins[[ species\_lkup[s,]$sp ]]))  
   
 }  
 if(srvys[r,]$reg =="goa"){  
 STRATA\_AREAUSE <- STRATA\_AREA%>%filter(REGION==srvys$RGN[r])  
 maxyr <- max(STRATA\_AREAUSE$YEAR)  
 STRATA\_AREAUSE <- STRATA\_AREAUSE%>%  
 filter(YEAR==1993)%>%  
 group\_by(REGION,STRATUM)%>%  
 summarize(AREA = mean(AREA, na.rm=T))%>%ungroup()  
   
 flnm <- paste0(srvys[r,]$reg,".srvy",  
 srvys[r,]$num,".",  
 species\_lkup[s,]$sp)  
 cat("now getting data for: ",flnm,"\n")  
 cpue\_data <- suppressMessages(  
 get\_CPUE\_DATA(  
 datapath = data.path,  
 out\_dir = file.path(data.out),  
 STRATA\_AREAIN = STRATA\_AREAUSE,  
 flnm = flnm,  
 species = species\_lkup[s,]$SPECIES\_CODE,  
 survey = srvys[r,]$num,  
 includeNBS = FALSE,  
 NEBSStrataIN = NEBS\_strata ,  
 saveit = T,  
 bins = sp\_bins[[ species\_lkup[s,]$sp ]]))  
   
 }  
 if(!srvys[r,]$reg%in%c("ebs","goa")){  
 flnm <- paste0(srvys[r,]$reg,".srvy",  
 srvys[r,]$num,".",  
 species\_lkup[s,]$sp)  
 cat("now getting data for: ",flnm,"\n")  
 cpue\_data <- suppressMessages(  
 get\_CPUE\_DATA(  
 datapath = data.path,  
 out\_dir = file.path(data.out),  
 STRATA\_AREAIN = NULL,  
 flnm = flnm,  
 species = species\_lkup[s,]$SPECIES\_CODE,  
 survey = srvys[r,]$num,  
 includeNBS = FALSE,  
 NEBSStrataIN = NEBS\_strata ,  
 saveit = T,  
 bins = sp\_bins[[ species\_lkup[s,]$sp ]]))  
  
 }  
   
 # # check the data :  
 if(1==10){  
 tt <- cpue\_data%>%  
 group\_by(YEAR,REGION,STATIONID,SN)%>%  
 filter(BIN ==400)%>%  
 summarize(cnt =length(STATIONID))  
 max(tt$cnt) #Should be 1  
 #this looks to be a duplicate sampling...  
 #mis-entry or code error ?  
 cpue\_data%>%filter(YEAR==1988,STATIONID=="J-13")  
 }  
 rm(cpue\_data)  
  
 }  
 }  
#   
# ```  
#   
# \*The cpue files are now saved in the directory `r file.path(data.out,"../")`\*  
#   
# ```{r viewcpue\_data, echo=TRUE, eval=FALSE}   
 # this uses the password saved in R/password.R  
 # suppressMessages(source("R/make.R"))  
load(file.path(data.out,"cpue/ebs/ebs.srvy98.pcod.cpue\_data.Rdata"))  
  
names(cpue\_data)  
  
library(dplyr)  
  
checkit <-function(x){  
 if(round(max(x ),1)!=1) {  
 warning("ERROR! propB > 1 ")  
 print(x)}  
   
}  
  
 #double check the results  
 cnt\_ByStrataBin <- cpue\_data$propByStrataBin%>%  
 select("REGION","YEAR","STRATUM",BIN,BIN\_mm,  
 SPECIES\_CODE,CN,SN,sp,num,  
 "propB\_ykl","propN\_ykl")%>%  
 group\_by(YEAR,REGION,SN,CN)%>%  
 summarise(sum\_propB\_ykl=sum(propB\_ykl,na.rm=T),  
 sum\_propN\_ykl=sum(propN\_ykl,na.rm=T))  
   
 cnt\_ByStrata <- cpue\_data$propByStrata%>%  
 select("REGION","YEAR","STRATUM",  
 SPECIES\_CODE,CN,SN,sp,num,  
 "propB\_yk","propN\_yk")%>%  
 group\_by(YEAR,REGION,SN,CN)%>%  
 summarise(sum\_propB\_yk=sum(propB\_yk,na.rm=T),  
 sum\_propN\_yk=sum(propN\_yk,na.rm=T))  
   
 cnt\_ByBin <- cpue\_data$propByBin%>%  
 select("REGION","YEAR",BIN,BIN\_mm,num,  
 SPECIES\_CODE,CN,SN,sp,  
 "propB\_yl","propN\_yl")%>%  
 group\_by(YEAR,REGION,SN,CN)%>%  
 summarise(sum\_propB\_yl=sum(propB\_yl ,na.rm=T),  
 sum\_propN\_yl=sum(propN\_yl ,na.rm=T))  
   
 checkit(cnt\_ByStrataBin$sum\_propB\_ykl)  
 checkit(cnt\_ByStrata$sum\_propB\_yk)  
 checkit(cnt\_ByBin$sum\_propB\_yl)  
   
#

## Appendix 1: R/setup.Rprimary setup script