Zooplankton estimates for 2025 ERP Benchmark

M. Celestino, D. Chagaris and A. Buchheister

SEDAR102-RW-06

25 July 2025

Please cite this document as:

Celestino, M., D. Chagaris and A. Buchheister. 2025. Revised Estimates of Natural Mortality for Atlantic Menhaden. SEDAR102-RW-06. SEDAR, North Charleston, SC. 35 pp.

SEDAR 102 WP-06: Zooplankton estimates for 2025 ERP Benchmark

M Celestino, D Chagaris, A Buchheister

2025-05-16

Zooplankton data background & early discussions

For the present assessment we sought to develop a time series of zooplankton abundance to which the NWACS models could be fit. A subgroup of the ERP-WG formed to review existing ecological indicator datasets (e.g., State of the Ecosystem reports, regional surveys, etc.) to determine whether indicators could be useful to this end. The outcome of the subgroup's work suggested that indicator data from existing sources could be fed directly into one or more of our ERP models. Review of NMFS (2023) by the subgroup suggested that the NMFS's EcoMon program could be a rich source of data for the ERP assessment.

While NMFS (2023) does generate specific zooplankton indices of abundance, early discussions with NMFS (H Walsh; harvey.walsh@noaa.gov) indicated they routinely generate unique custom zooplankton indices of abundance for stock assessments and that they would be happy to do the same for us. Indices of abundance can be provided as either number per 100 m^3 or number per 10 m_2, for any number of zooplankton categories we would like (only number per 10 m^2 were of interest to us). H Walsh sent us a spreadsheet with the species composition of their current categories and indicated that we can, for example, rearrange species membership, or change the number of categories from however many they have to just the ones in NWACS etc. Indices can be generated by ecological production unit (EPU) or coastwide (or both).

H Walsh sent us a spreadsheet with nearly all (see next bullet) taxa collected in the EcoMon survey:

- All of the taxa in the spreadsheet account for ~95% of all plankton NMFS captures in the EcoMon survey (i.e., omits the rarest 5% of taxa).
- Dave, Andre, and I met via conference call a number of times to discuss the data needs for NWACS-Full and NWACS-MICE. From Link et al. (2006), and used in SEDAR (2020), the 5 zooplankton groups we were interested in:
  - Large copepods: the V and VI copepodites stages of Calanus finmarchicus, Metridia lucens, and Centropages typicus. a,b
  - Small copeopods: stage I-IV copepodites of the large copepod species (i.e., stage I-IV copepodites of Calanus finmarchicus, Metridia lucens, and Centropages typicus) and the I-VI copepodites stages of Centropages hamatus, Pseudocalanus spp., Temora longicornis, Paracalanus parvus, Nannocalanus minor, and Clausocalanus arcuicornis. a,b

- Gelatenous zooplankton: Cnidaria (both the medusae and hydrozoans); the Ctenophores (comb jellies); the colonial Siphonophores; and the colonial Salpidae. c
- Micronekton: the largest size animals taken in plankton nets, typically having body lengths of 5-10mm or more. For this study, the micronekton group is considered inclusive of the crustacean groups: amphipoda, euphausiacea, mysidacea, and similar decapoda captured in plankton nets. Chaetognatha are also included in this group. d
- Microzooplankton = 13% of phytoplankton biomass estimate.

a Methods described in Link et al (2006): Plankton samples were collected seasonally on two types of cruises: 1) broad scale surveys dedicated to plankton where sampling was done at standard or randomly selected stations spaced approximately 8-35 km apart; and 2) trawl and dredge surveys where plankton stations were selected from a stratified random plan at locations uniformly distributed over the region. Samples were all collected with a 61 cm bongo frame fitted with a 0.333 mm mesh net towed obliquely to a maximum depth of 200 m or 5 m from the bottom and back to the surface. A digital flowmeter was suspended in the center of the bongo frame to measure volume of water filtered during the tow. Samples were then reduced to approximately 500 organisms by subsampling with a modified box splitter.

b Copepod abundance by size group was converted to biomass using the length to wet weight (W) equation given by Pearre (1980): W=0.08810L^2.8514 (where L is from Pearre 1980).

c Gelatinous zooplankton biomass was estimated from 60 cm bongo tows with 333 mm mesh nets taken on NEFSC monitoring cruises. Mean abundance per m^3 for each station are the calculated mean of the abundance for each stratum sampled. Mean station abundance was multiplied by the sampling depth to calculate number per m^2. Individual group biomasses were calculated using the following relationship (Reeve and Walter 1976): Log DW=2.65 Log L Where DW = dry weight (g) and L is length (mm). This relationship was established for ctenophores and is assumed to be similar enough for all other gelatinous zooplankton groups such that we use it for all these zooplankton taxa. A mean length of 1.3 mm was assumed for this calculation. Total biomass for all groups was then integrated into an annual average, summed across all gelatinous zooplankton taxa, and then converted to g wet weight per m^2. Conversion to wet weight from dry weight was approximated from Pages (1997), with DW = 4.48% of WW.

d The mean abundance (no / 10 m^2) of micronekton were calculated distinctly for each of the groups listed above. These estimates of mean abundance were then converted to biomass based upon established size-biomass relationships. Here we assume that most micronekton were roughly equivalent to a common micronekton taxa, the amphipod Gamarrus sp. Mean abundance was converted to dry weight from the relationship established in Avery et al (1996), using a mean length of 6 mm which produced a dry weight estimate of 1.2 mg. This average weight was then multiplied by the abundance estimates to obtain a total biomass for each group. However, for the chaetognaths group, we used a dry weight of 0.026 mg. After conversions to biomass were done for each micronekton group, the values were then converted to biomass and integrated into an annual estimate. This was then converted to g per weight per m^2.

(1) Pearre, S Jr. 1980. The copepod width-weight relation and its utility in food chain research. Can J Zool. 58:1884-1891.

Into the weeds:

As part of our (Dave, Andre, and I) discussions, we agreed to, rather than classifying copepods as large or small based on observed size (or stage), as a simplifying measure, to instead classify copepods as large or small based on an average size of the adult animal, reasoning that diet habits are likely similar among similarly sized copepods (e.g., small copepods likely eat bacteria, while larger copepods likely eat larger algae (link; e.g., herbivorous copepods vs predatory copepods). A series of Google and ChatGPT searches aided in assigning an average size to the various copepod species collected in the EcoMon survey; additionally, the spreadsheet from H Walsh had some initial large or small copepod designations, which were also used as a guide.

As was done in Link et al. (2006), chaetognaths were grouped as a separate category within the micronekton zooplankton group – they are included in the NWACS 'micronekton' category but have such a different assumed average weight than the other micronekton (0.026 mg vs 1.2 mg) that we thought it was appropriate to break them out (as was done for EMAX), then recombine for total micronekton.

We assigned all taxa in ZooTaxaUsed_NWACS_submitted.xlsx to one of our various NWACS categories (see above, Appendix 2 of SEDAR 2020, and Buchheister et al. 2017). There were a number of taxa that were binned into an 'omit' category - that either didn't quite fit into the descriptions of NWACS categories or are classified as something other than zooplankton in our categories. To get a handle on the magnitude of this 'omit' category, we requested an index of abundance on the 'omit' category (i.e., all 'omit' taxa combined).

We talked about attempting to convert zooplankton biomass per m^2 to biovolume, and did some Google searches on conversions to that end, but in the end, decided not to use this approach (at least partly due to uncertainties in conversion factors). We also considered obtaining biovolume for each of our NWACS zooplankton groups, but learned that this is not physically possible; biovolume = laboratory displacement volume = all plankton + preservative in graduated cylinder poured through mesh cone into second cylinder. Displacement volume = the difference in volume between 1st and 2nd cylinder. That is, NMFS biovolume is calculated for all zooplankton all at once.

For the MICE model, we agreed that seasonal resolution of indices of abundance would be helpful for exploration, but this level of resolution was not necessary for NWACS-Full. Additionally, since the MICE model pools all zooplankton species into a single group (vs NWACS-Full that keeps all zooplankton groups separate), we also agreed to request biovolume of all zooplankton (seasonally, and by EPU) from NMFS.

Pteropods have been identified as an increasingly important component of the zooplankton (NMFS 2023). Pteropods were not mentioned in the EMAX (Link et al. 2006) or NWACS model documentation (Buchheister et al. 2017) and so we believe they were omitted / not included, and so we classified them as a separate zooplankton group that could be treated separately if need be. We also suspected that the energetic value of pteropods could differ from other plankton and so keeping as a separate group seemed sensible. With some Google searches we were able to find some average size and weight information for pteropods. We thought that micronekton could reasonably be grouped with micronketon or omitted, and this is a decision point we discussed with the full ERP-WG at their Nov 2004 methods workshop and March 2025 model workshop.

Deeper into the weeds - pteropods:

From Googling around, we initially assumed pteropods that EcoMon collects hover around 5-6mm (see <outputs (2025) Feb 6th 2025> in ERP_zooplankton_data_1977to2022.xlsx); we found a citation for the average size of a pteropod in the Scotia Sea = 2.74 mm with dry weight = 0.62 mg; and we found a length weight relationship for Scotia Sea pteropods - if the relationship holds in the north western Atlantic, then a ~5.5mm pteropod might weigh around ~1.8 mg, which is about what we are assuming for micronekton (minus chaetognaths; see table below). We considered bundling pteropds with micronekton. But also wondered whether the assumptions might be a little too shaky (average size of a pteropod in NWACS, applying an Antarctic pteropod len-wt relationship to NWACS), bolstered by the omission in EMAX, as to support their continued omission from NWACS? We also wondered whether if by including pteropods, while EMAX didn't, would we need to re-configure some of the EwE inputs (P:B, Q:B, etc)? We also learned later that most plankton in EcoMon are < 1.5 mm.

Pteropod information.

Species or group	Assumed average length (mm)	Assumed average dry weight (mg)	Source of assumption
Micronekton (without chateognaths; no pteropds)	6.00	1.200	EMAX
Micronekton (chateognaths only)	4.00	0.026	EMAX
Limacina antartica (Scotia Sea pteropod)	2.74	0.620	Literature(a)
Generic NWACS pteropod	5.50	1.800	Literature(b)

Footnotes

Footnote	Source
(a)	Bednarsek et al. 2017
(b)	Bednarsek et al. 2012 (len-wt rel) & Google (mean size)

Size of EPUs:

Units = m^2; values from Brandon Beltz, brandon.beltz@noaa.gov, pers. comm. 28 October 2024.

EPU	area
SS	30,209,438,176
GOM	68,311,878,130
GB	57,307,696,043
MAB	126,862,117,858

Final 2025 benchmark methods

Small copepods:

We converted numeric abundance of small copepods to weight (mg to g) via: (sm copepod index) x ((0.0881*2.0^2.8514)/1000), assuming mean size = 2.0 mm to get final densities in the ballpark of the SEDAR (2020) estimates. Finally, we converted to g per m^2 via ÷ 10. Length-wt conversion from Table 4 of Pearre (1980) (wet weight predicted from prosome length).

Large copepods:

We converted numeric abundance of large copepods to weight (mg to g) via: (lg copepod index) x ((0.0881*2.2^2.8514)/1000), assuming mean size = 2.2 mm to get final densities in the ballpark of the SEDAR (2020) estimates. Finally, we converted to g per m^2 via ÷ 10. Length-wt conversion from Table 4 of Pearre (1980) (wet weight predicted from prosome length).

Gelatenous zooplankton

We converted numeric abundance of gelatinous zooplankton to weight (mg to g) via: (gel zoop index) x ((exp(-1.54+2.65*log(0.55))/0.0448)/1000), assuming mean size = 0.55 mm to get final densities in the ballpark of the SEDAR (2020) estimates. Finally, we converted to g per m^2 via ÷ 10. The conversion from length to dry weight was from Reeve and Walter (1976), and conversion to WW (÷ 0.0448) was from Link et al. (2006).

Chateognaths

We converted numeric abundance of chateognaths to weight (mg to g) via: (chaeto index) x ((0.00097*4^2.365)/1000), assuming mean size = 4 mm to get final densities in the ballpark of the SEDAR (2020) estimates. We needed to convert from DW to WW via ÷ 0.083 (Table 7.1 in Link et al. 2006), then finally, converted to g per m^2 via ÷ 10. As a reminder, chaetognaths are micronekton, but literature suggests that they have a very different average weight than the other micronekton, so total mass was calculated separately, and subsequently combined with other micronekton.

Micronekton

We converted numeric abundance of micronekton (excluding pteropods) to weight (mg to g) via: (micronekton index) x (0.28/1000); avg weight came from the literature (0.28/1e3), converted to WW via DW*9.30 (Table 7.1 in Link et al. 2006), and finally, converted to g per m^2 via ÷ 10.

Pteropods (not included in 2025 benchmark)

Though not used, we are documenting for future explorations. We converted numeric abundance of pteropods via: (ptero index) x (0.137 * 1.5^1.5005)/1000), assuming mean size = 1.5 mm. Given the available length weight equation, this size approximated the average weight of 'other micronekton.' Finally, we converted to weight per m^2 via ÷ 10. Note: converted DW to WW assuming the same relationship as with other micronekton (i.e., DW*9.30).

Microzooplankton

We made the same assumptions as reported in Link et al. (2006):

- "…assumed that the microzooplankton were primarily composed of protozoans…"
- "Unfortunately traditional zooplankton sampling nets destroy the fragile protozoans, so we lack a monitoring database…"
- "Since we didn't have any independent data on protozoan biomass and rates on the Northeast Continental Shelf, we decided to relate the MZ biomass (in carbon units) to that of phytoplankton (in carbon units) based on Figure 3 in Caron et al. (1990) which showed a relationship (log-log) between ciliate and phytoplankton biomass. *We assumed that MZ (microzooplankton) biomass was 0.13 of the phytoplankton biomass* (emphasis added), similar to values for unfertilized North Sea mesocosms (Baretta-Bekker, 1994) and Narragansett Bay (Monaco, 1997). Given the boom and bust life history strategy of protozoans, we assumed that their annual biomass would be a relatively small fraction of the annual phytoplankton biomass."

Consequently, we are unaware of direct data sources for microzooplankton. Therefore, to estimate the biomass of microzooplankton, we looked to SEDAR (2020) and found that a density of 7 g/m^2 was assumed. Discussions with A Buchheister suggested that the value likely came from Link et al. (2006) (which summed to 3 g/m^2); SEDAR (2020) provides some additional information on how the EMAX value was modified for SEDAR (2020):

- "Section 3.3.1 Parameters for lower and higher trophic levels that leverage the EMAX models… We relied heavily on ecosystem models developed by the NEFSC for the Energy Modeling and Analysis exercise (EMAX) project (Link et al. 2006, 2008)… We preferentially used the original EMAX input values as our starting point, but some values were changed during the model balancing process."
- "Section 3.4. Description of Ecopath balancing procedure… Link et al (2006) had recognized that the EMAX model values for unassimilated fractions were low for several low-trophic-level groups (e.g., bacteria, *microzooplankton* (emphasis added), copepods, and micronekton); therefore, we increased the unassimilated fractions

for those groups to balance the detritus group and to bring the values more in line with general recommendations (Link et al. 2006; Christensen et al. 2008)." [from pdf page 516 of 560]

So as a starting point, combined with the details in Link et al. (2006), we used the ratio of microzooplankton to phytoplankton from SEDAR (2020; Appendix 2 parameters for the balanced NWACS ecosystem model): $\frac{7}{30} = 0.23$. In the absence of any direct data sources, the percentage of phytoplankton approach was retained.[1] Further, we used this ratio, multiplied by estimates of phytoplankton from the GLORYS data set (provided by D Chagaris) to develop the final microzooplankton time series.

Recollections from Andre via email: As best we can recall, the percentage approach wasn't used in SEDAR (2020), but rather the EMAX value would have been used, tweaked as part of balancing and calibration, and finalized for use in SEDAR (2020). Since this value equated with 23% of phytoplankton, we are using that as a starting value for the present benchmark.

A note on the area sampled

The EcoMon samples are collected over a spatial area of 282,691.1 km^2, but the NWACS spatial domain is 441,000 km^2, which is calculated from spatial grids used in Ecospace. The 2025 NWACS spatial domain is a larger area than the survey domain used in SEDAR (2020). Since the 2025 model domain is larger than the area sampled by EcoMon, rather than use absolute biomass estimates from EcoMon as inputs to NWACS, we used the **density** estimates as the inputs to NWACS-MICE and NWACS-Full (g/m^2 = mt/km^2). We felt comfortable applying the EcoMon densities to the full updated NWACS model domain since Ecomon samples across a broad representative spatial domain.

Research recommendations

Things that we weren't able to accomplish for the 2025 benchmark include the items below and might be explored as part of the next benchmark:

- Estimates of uncertainty: communication with H Walsh suggested that a new pc and some associated lost files conspired to complicate generation of estimates of uncertainty.
- Pteropods: NMFS was able to generate a time series of pteropod abundance indices, but we felt that additional vetting and decision points could be explored. For example, are the energetics and biology (e.g., P:B, Q:B) of pteropods similar enough to one of our existing zooplankton groups (e.g., micronekton) to combine, or are they different enough to warrant a separate group. Additionally, more localized information on mean size and weight would be helpful (current weight is estimated

[1] We also discussed the possibility of having EwE estimate the micronekton biomass values, but due to time limitations we do not believe this was explored, but could be explored as part of the next benchmark.

using an asumed mean size and a length-weight equation from the Scotian Sea (Bednarsek et al. 2012)).

Literature cited

Bednarsek, N, GA Tarling, S Fielding, and DCE Bakker. 2012. Population dynamics and biogeochemical significance of Limacina helicinia antarctica in the Scotia Sea (Southern Ocean). Deep-Sea Research II 5-60, 105-116.

Buchheister, A., Miller, T. J., Houde, E. D., and D. A. Loewensteiner. 2017. Technical Documentation of the Northwest Atlantic Continental Shelf (NWACS) Ecosystem Model. Report to the Lenfest Ocean Program, Washington, D.C. University of Maryland Center for Environmental Sciences Report TS-694-17. Available at: http://hjort.cbl.umces.edu/NWACS/TS_694_17_NWACS_Model_Documentation.pdf

Link JS, Griswold CA, Methratta ET, Gunnard J, Editors. 2006. Documentation for the Energy Modeling and Analysis eXercise (EMAX). US Dep. Commer., Northeast Fish. Sci. Cent. Ref. Doc. 06-15; 166 p. Available from: National Marine Fisheries Service, 166 Water Street, Woods Hole, MA 02543-1026.

NMFS. 2023. 2023 State of the ecosystem, Mid-Atlantic. Report Technical Documentation

Pearre, S. 1980. The copepod width-weight relation and its utility in food chain research. Can J. Zool. 58:1884-1891.

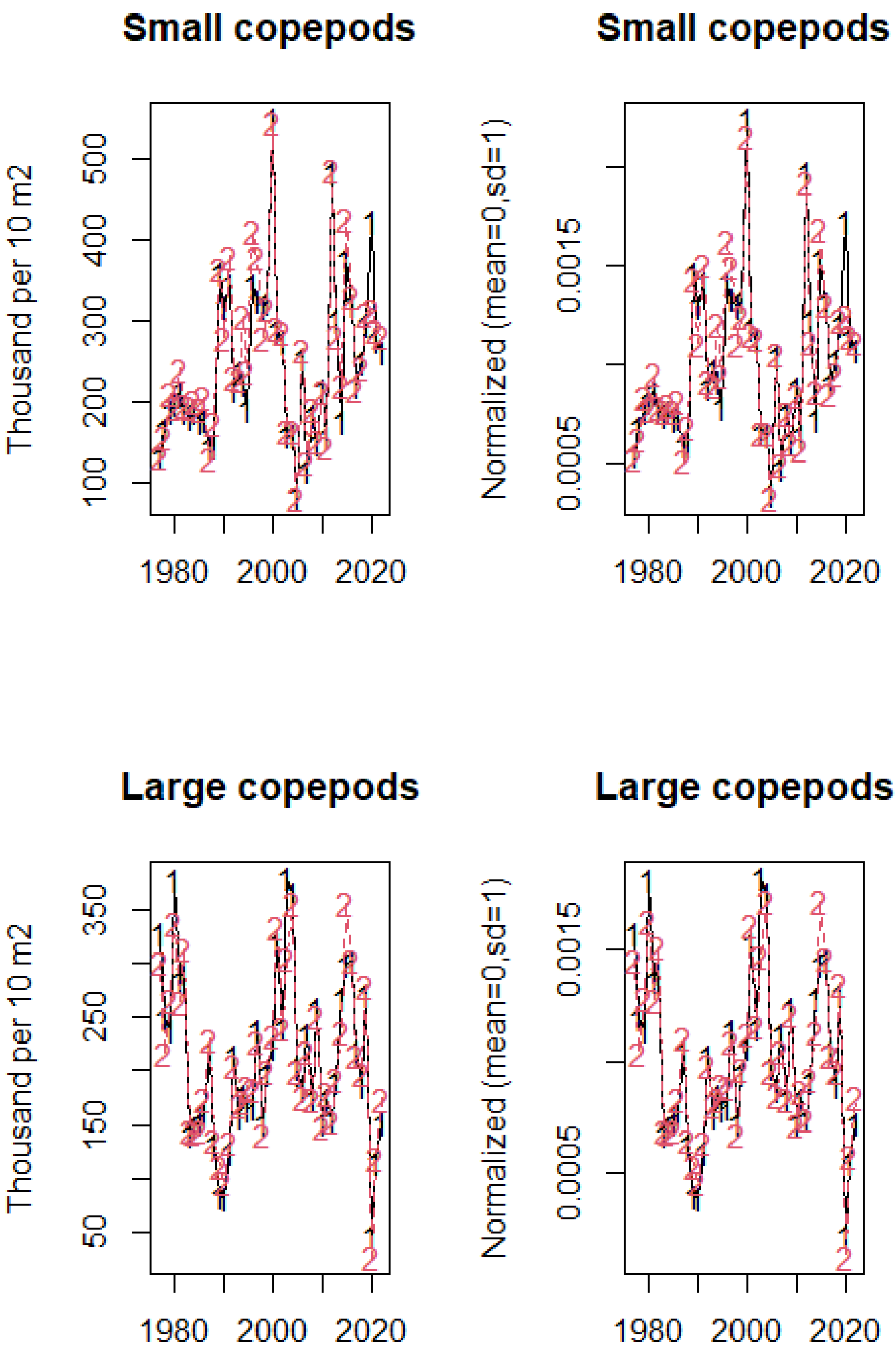
Reeve R and M Walter. 1976.A large scale experiment on the growth and predation potential of ctenophore populations. In: Mackie, G., ed. Coelenterate ecology and behavior. New York, NY: Plenum Pubcorp; pp 187-199.

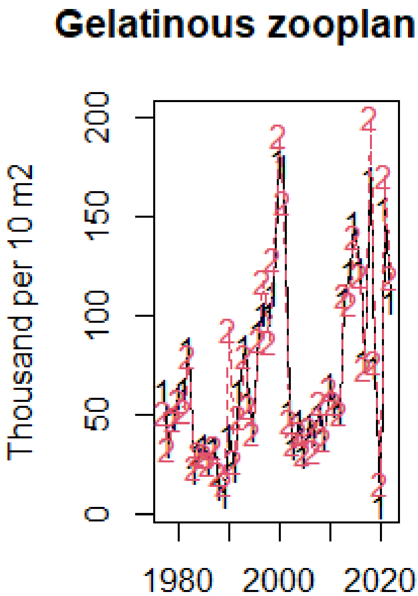
SEDAR. 2020. SEDAR 69 – Atlantic Menhaden Ecological Reference Points Stock Assessment Report. SEDAR, North Charleston SC. 560 pp. available online at: http://sedarweb.org/sedar-69

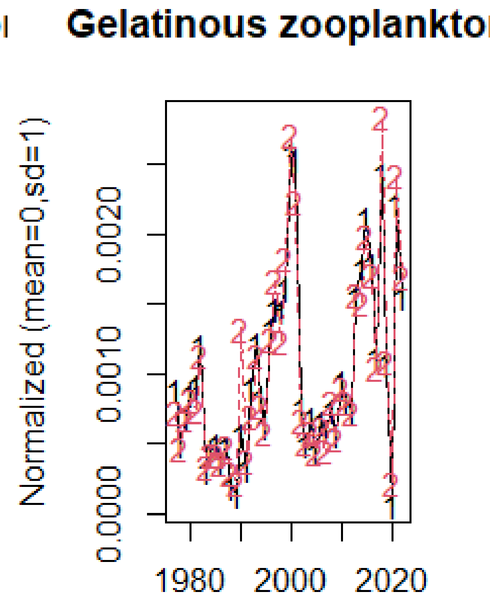
Appendix (code)

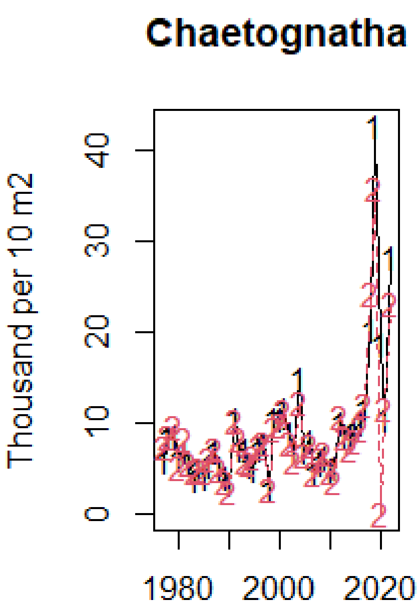
```
# Pared down from zoop_summary_03c-Feb6.r
# a = all rows, all columns
require(readxl)
require(reshape)

## Loading required package: reshape

require(plotrix)

## Loading required package: plotrix

require(latticeExtra)

## Loading required package: latticeExtra

## Loading required package: lattice

##
## Attaching package: 'latticeExtra'

## The following object is masked from 'package:ggplot2':
##
##     layer

require(plyr)

## Loading required package: plyr

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:reshape':
##
##     rename, round_any

#newO <- options(scipen=2)


a <- as.data.frame(read_xlsx("X:\\zooplankton\\zoopIndices\\ERP_zooplankton_data_1977to20
23_5Feb2025_v1b.xlsx",
   sheet="Data",na="NaN")) # Winter = Jan, Feb, March

b <- as.data.frame(read_xlsx("X:\\zooplankton\\zoopIndices\\ERP_zooplankton_data_1977to20
23_5Feb2025_v2b.xlsx",
   sheet="Data",na="NaN")) # winter = Dec, Jan, Feb


# compare results with winter defined as months 1:3 vs 12:2 (generally pretty similar):
cbind(
   cbind(
   meanJan=round(colMeans(a[a$Year<2023,-c(1:4)],na.rm=TRUE)),
   meanDec=round(colMeans(b[b$Year<2023,-c(1:4)],na.rm=TRUE))),

   cbind(
   minJan=round(apply(a[a$Year<2023,-c(1:4)],2,min,na.rm=TRUE)),
   minDec=round(apply(b[b$Year<2023,-c(1:4)],2,min,na.rm=TRUE))),
```

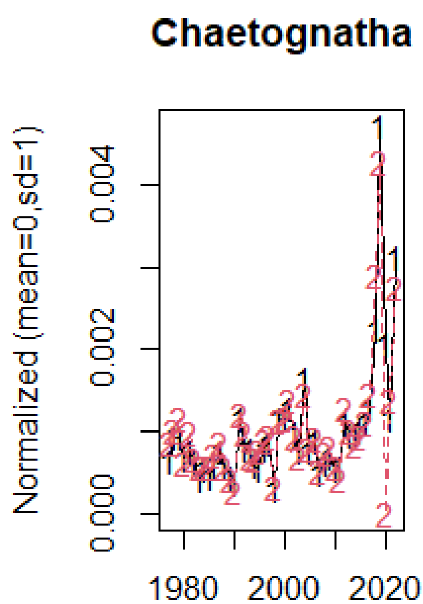
```
  cbind(
  maxJan=round(apply(a[a$Year<2023,-c(1:4)],2,max,na.rm=TRUE)),
  maxDec=round(apply(b[b$Year<2023,-c(1:4)],2,max,na.rm=TRUE)))
)

##                        meanJan meanDec minJan minDec  maxJan  maxDec
## Small copepods          236492  246568      0      0 2647596 2647596
## Large copepods          214203  211140      0      0 1587002 1587002
## Gelatinous zooplankton   68662   67595      0      0  894248  731699
## Chaetognatha              8529    8172      0      0  297363  194073
## Micronekton              29371   27809      0      0 4166377 4166377
## Pteropod                 11862    9825      0      0  388950  163670
## Omit                     74285   68771      0      0 2020761 1935041

norm <- function(x) {
  x/mean(x)
}

# Plot indicies over time, comparing index that defines Winter starting in Jan vs Dec
# (Very similar)
op <- par(mfrow=c(1,2))
for(i in c("Small copepods","Large copepods","Gelatinous zooplankton","Chaetognatha","Mic
ronekton","Pteropod","Omit")) {

  matplot(1977:2022,
  data.frame(winterJan= tapply(a[a$Year<2023,i],a[a$Year<2023,"Year"],mean,na.rm=TRUE),
     winterDec=tapply(b[b$Year<2023,i],b[b$Year<2023,"Year"],mean,na.rm=TRUE))/1e3,type=
"o",ylab="Thousand per 10 m2",xlab="",main=i)

  matplot(1977:2022,
  data.frame(winterJan= norm(tapply(a[a$Year<2023,i],a[a$Year<2023,"Year"],mean,na.rm=TR
UE)),
     winterDec=norm(tapply(b[b$Year<2023,i],b[b$Year<2023,"Year"],mean,na.rm=TRUE))))/1e3
,type="o",ylab="Normalized (mean=0,sd=1)",xlab="",main=i)
}
```

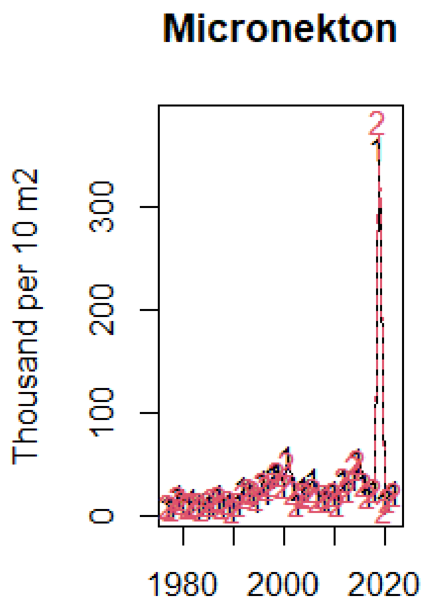
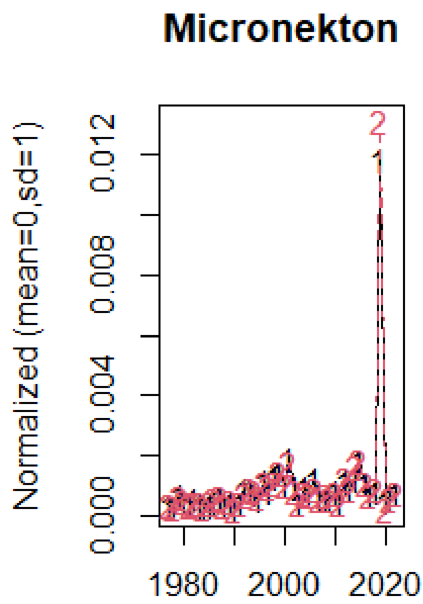
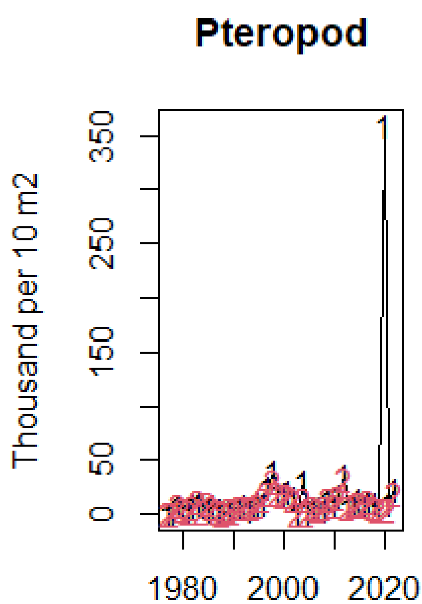
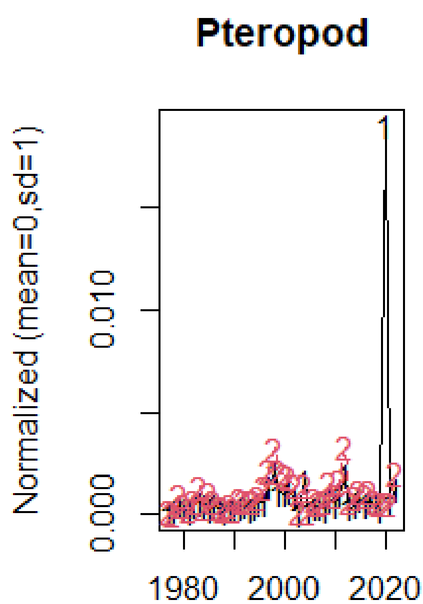

Small copepods
Small copepods
Thousand per 10 m2
Normalized (mean=0,sd=1)
100
200
300
400
500
0.0005
0.0015
1980
2000
2020
1980
2000
2020
Large copepods
Large copepods
Thousand per 10 m2
Normalized (mean=0,sd=1)
50
150
250
350
0.0005
0.0015
1980
2000
2020
1980
2000
2020

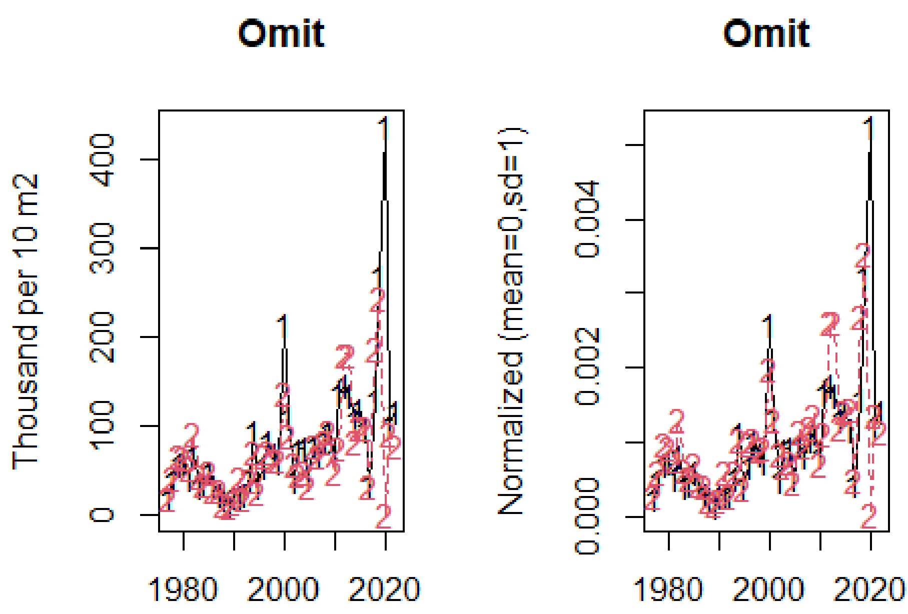

Gelatinous zooplanktor
Thousand per 10 m2
0
50
100
150
200
1980
2000
2020


Gelatinous zooplanktor
Normalized (mean=0,sd=1)
0.0000
0.0010
0.0020
1980
2000
2020


Chaetognatha
Thousand per 10 m2
0
10
20
30
40
1980
2000
2020


Chaetognatha
Normalized (mean=0,sd=1)
0.000
0.002
0.004
1980
2000
2020


Micronekton
Thousand per 10 m2
0
100
200
300
1980
2000
2020

Micronekton
Normalized (mean=0,sd=1)
0.000
0.004
0.008
0.012
1980
2000
2020

Pteropod
Thousand per 10 m2
0
50
150
250
350
1980
2000
2020

Pteropod
Normalized (mean=0,sd=1)
0.000
0.010
1980
2000
2020




```r
# Look at correlation in mean abundance as a function of how winter is defined:
# (again, very similar)
for(i in c("Small copepods","Large copepods","Gelatinous zooplankton","Chaetognatha","Mic
ronekton","Pteropod","Omit")) {
   cat("\n\n\n")

   print(i)

   print(round(cor(data.frame(winterJan= tapply(a[a$Year<2023,i],a[a$Year<2023,"Year"],me
an,na.rm=TRUE),
      winterDec=tapply(b[b$Year<2023,i],b[b$Year<2023,"Year"],mean,na.rm=TRUE)),method="s
pearman"),2))
}

##
##
##
## [1] "Small copepods"
##           winterJan winterDec
## winterJan      1.00      0.96
## winterDec      0.96      1.00
##
##
##
## [1] "Large copepods"
##           winterJan winterDec
## winterJan      1.00      0.98
## winterDec      0.98      1.00
##
##
##
## [1] "Gelatinous zooplankton"
```

```
##           winterJan winterDec
## winterJan      1.00      0.96
## winterDec      0.96      1.00
##
##
##
## [1] "Chaetognatha"
##           winterJan winterDec
## winterJan      1.00      0.85
## winterDec      0.85      1.00
##
##
##
## [1] "Micronekton"
##           winterJan winterDec
## winterJan      1.00      0.96
## winterDec      0.96      1.00
##
##
##
## [1] "Pteropod"
##           winterJan winterDec
## winterJan      1.00      0.84
## winterDec      0.84      1.00
##
##
##
## [1] "Omit"
##           winterJan winterDec
## winterJan       1.0       0.8
## winterDec       0.8       1.0
```

```
# Punchline: defining winter as 1:3, or 12:2 has little impact, but we went with
#  Winter = Jan-Mar for consistency with other data source
rm(b) # delete this before it gets in the way...

# Interpolate missing samples? No!

################################################################################
# Small copepods
# Multiply by avg wt of sm copepod (originally assumed mean size of sm copepod = 1.2 mm (
EMAX), but after some tweaks to get closer to SEDAR(2020) increased to 2.0mm; and note th
at when I went to the primary literature, I was able to confirm the conversion is for wet
weight! (not dry wt as reported in EMAX)):
a$`Small copepods` <- a$`Small copepods` * ((0.0881*2.0^2.8514)/1000)

# See how close we can get to EMAX:
tmp <- ddply(a[is.element(a$Year,1996:2000),],.(EPU,Season),summarize, .mean=mean(`Small
copepods`,na.rm=TRUE)) #

a$`Small copepods` <- a$`Small copepods` * 1 # already WW so no need to convert
mean(a$`Small copepods`,na.rm=TRUE)
```

```
## [1] 151.4111
```

```
# Now convert to wt per m2
a$`Small copepods` <- a$`Small copepods`/10
mean(a$`Small copepods`,na.rm=TRUE)

## [1] 15.14111

#####################################################################################
# Large copepods
# Multiply by avg wt of lg copepod (originally assumed avg size of lg copepod to be 2.5mm
, but after tuning to SEDAR (2020) using 2.2 mm; and as above, the primary literature con
firms the equation below converts length to wet weight!
a$`Large copepods` <- a$`Large copepods` * ((0.0881*2.2^2.8514)/1000)

# See how close we can get to EMAX:
tmp <- ddply(a[is.element(a$Year,1996:2000),c("EPU","Season","Large copepods")],.(EPU,Sea
son),summarize, .mean=mean(`Large copepods`,na.rm=TRUE))

a$`Large copepods` <- a$`Large copepods` * 1 # Already wet wt (WW), so no need to convert
mean(a$`Large copepods`,na.rm=TRUE)

## [1] 180.173

# Now convert to wt per m2
a$`Large copepods` <- a$`Large copepods`/10
mean(a$`Large copepods`,na.rm=TRUE)

## [1] 18.0173

#####################################################################################
# Gel zooplankton
# See how close we are to EMAX (numbers per 10m2; Table 6.1):
ddply(a[is.element(a$Year,1996:2000),], .(EPU), summarize, .mean=mean(`Gelatinous zooplan
kton`,na.rm=TRUE)) # We're in the right ballpark with abundance!

##   EPU    .mean
## 1  GB  93973.4
## 2 GOM 132282.7
## 3 MAB 108197.7
## 4  SS 132206.1

# Final
a$`Gelatinous zooplankton` <- a$`Gelatinous zooplankton` * ((exp(-1.54+2.65*log(0.55))/0.
0448)/1000)# EMAX w/ typo corrected, and have tweaked the mean size of gelzoop to tune to
SEDAR (2019), and converted to WW
# Then convert to abundance per m2
a$`Gelatinous zooplankton` <- a$`Gelatinous zooplankton`/10
mean(a$`Gelatinous zooplankton`,na.rm=TRUE) # 6.7 g kg per m2

## [1] 6.709856

#####################################################################################
# Chaetognatha
# Multiply by avg wt of Chaetognath (g): DW

a$`Chaetognatha` <- a$`Chaetognatha` * ((0.00097*4^2.365)/1e3) # DW g/10m2

# How close to EMAX can we get:
```

```
tmp <- ddply(a[is.element(a$Year,1996:2000),], .(Season,EPU), summarize, .mean=mean(`Chaetognatha`,na.rm=TRUE)*1000) # convert back to mg for comparison with EMAX tables
apply(reshape(tmp,direction="wide",idvar="Season",timevar="EPU")[,-1],2,mean) # Avg DW (mg) Pretty good!
```

```
##  .mean.GB .mean.GOM .mean.MAB  .mean.SS 
## 225.19257 103.57818 389.55422  56.49015
```

```
((apply(reshape(tmp,direction="wide",idvar="Season",timevar="EPU")[,-1],2,mean)/1000)/0.083)/10 # WW g/m2; very good!
```

```
##   .mean.GB  .mean.GOM  .mean.MAB   .mean.SS 
## 0.27131635 0.12479299 0.46934244 0.06806042
```

```
# Convert to wet weight:
a$`Chaetognatha` <- a$`Chaetognatha`/0.083 # DW = 8.3% (+/- 0.2%) of the wet weight

# Convert per 10m2 to per m2
a$`Chaetognatha` <- a$`Chaetognatha`/10
mean(a$`Chaetognatha`,na.rm=TRUE) # mean g/m2; very close to EMAX Table 7.1
```

```
## [1] 0.2662079
```

```
################################################################################
# Other micronekton
# How close can we get to EMAX:
tmp <- ddply(a[is.element(a$Year,1996:2000),], .(Season,EPU), summarize, .mean=mean(`Micronekton`,na.rm=TRUE))
apply(reshape(tmp,direction="wide",idvar="Season",timevar="EPU")[,-1],2,mean) # quite a bit higher than EMAX abudnance per 10 m2
```

```
##  .mean.GB .mean.GOM .mean.MAB  .mean.SS 
##  33046.69  14592.73  76284.59  32307.15
```

```
# Multiply by avg wt of a micronekton (g): DW
a$`Micronekton` <- a$`Micronekton` * (0.28/1e3) # DW g/10m2                      # Assumed dry wt = 1.2 mg, or 0.0012 g each

# Convert to wet weight:
a$`Micronekton` <- (9.299855693452*a$`Micronekton`)                             # From EMAX Table 7.1, assume DW = 0.1075286 x WW; therefore 1/0.1075286

# How close to EMAX did we get:
ddply(a, .(EPU), summarize, .mean=mean(`Micronekton`,na.rm=TRUE))
```

```
##   EPU     .mean
## 1  GB  58.05299
## 2 GOM  38.79491
## 3 MAB 113.95131
## 4  SS  99.00890
```

```
# Convert per 10m2 to per m2
a$`Micronekton` <- a$`Micronekton`/10
mean(a$`Micronekton`,na.rm=TRUE)
```

```
## [1] 7.769656
```

```
##################################################################################
# Pteropods
tmp <- ddply(a, .(Season,EPU), summarize, .mean=mean(`Pteropod`,na.rm=TRUE))
apply(
   reshape(tmp,direction="wide",idvar="Season",timevar="EPU")[,-1],
   2,mean)/10 # number per m2

##  .mean.GB .mean.GOM .mean.MAB  .mean.SS
## 1475.2357  606.4824 2133.6823  534.7312

# Multiply by avg wt of a Pteropod (g): DW
a$`Pteropod` <- a$`Pteropod` * ((0.137*1.5^1.5005)/1e3) # DW g/10m2  (assume avg size of
Pteropod is 1.5mm; see spreadsheet for details, but essentially, given the len-wt equatio
n I have, this is the size that approxiamtes the avg wt of 'Other micronekton' and makes
sense (Harvey noted that most things they see in nets are < about 1.5mm).

# Convert to wet weight (assuming same relationship as Other micronekton...):
a$`Pteropod` <- (9.299856*a$`Pteropod`)

# Convert per 10m2 to per m2
a$`Pteropod` <- a$`Pteropod`/10
mean(a$`Pteropod`,na.rm=TRUE)

## [1] 2.863368

##################################################################################
# Area of each EPU:
epu <- data.frame("EPU"=c("MAB","GOM","SS","GB"), "areaKm2"=c(126862117858, 68311878130,
30209438176, 57307696043)/1e6)
                                                                # units = square km; note that
by having units = km2, multiplying grams per m2 by km2, results in MT!

A <- merge(a,epu,by.x="EPU",by.y="EPU",all.x=TRUE)
B <- A[A$Year>1984,]                                                                  # Note th
at for final run, including Scotian Shelf; in reality, should probably only use ~1/2 of S
S, but to keep me sane, going with the entire area.
B <- B[order(B$Year,B$EPU,B$Season),]

out <- ddply(B, .(Year,EPU,Season), summarize,
   smCopepods=`Small copepods`*areaKm2,
   lgCopepods=`Large copepods`*areaKm2,
   gelZoop=`Gelatinous zooplankton`*areaKm2,
   chateo=`Chaetognatha`*areaKm2,
   micronek=`Micronekton`*areaKm2,
   pteropod=`Pteropod`*areaKm2
   )

                                                                                      # Note th
at for final run, excluding Pteropods (not clear which, if any, group they should be comb
ined with, plus I had to make a number of assumptions about which I am not super clear on
(e.g., assumed length etc; see code above for all assumptions).
out2 <- data.frame(out[,c("Year","EPU","Season","smCopepods","lgCopepods","gelZoop")],"mi
cronek"=rowSums(out[,c("chateo","micronek")],na.rm=TRUE))#,pteropod=out[,"pteropod"])


# Average over seasons within a year (and EPU)                                        # Skip do
```

```
wn to line approx XXXX for calcs with calcs by season (rather than avg over seasons)
out3 <- ddply(out2, .(Year,EPU), summarize,                                    # I think
regular means (not weighted) are ok here, because I have already expanded the
   smCopepods=mean(smCopepods,na.rm=TRUE),                                     #   bioma
ss by the area of the EPU above!
   lgCopepods=mean(lgCopepods,na.rm=TRUE),
   gelZoop=mean(gelZoop,na.rm=TRUE),
   micronek=mean(micronek,na.rm=TRUE)#,
#   pteropod=mean(pteropod,na.rm=TRUE)
   )

G <- reshape::melt(out3,id=c("Year","EPU"))
colnames(G)[3:4] <- c("spp","totalWt")

H <- reshape(G,direction="wide", idvar=c("Year","spp"),timevar="EPU")

J <- data.frame(H[,c(1:2)],"totalWt"=rowSums(H[,-c(1,2)],na.rm=TRUE))

K <- reshape(J,direction="wide",idvar="Year",timevar="spp")                    # These w
ere the main outputs for NWACS-FULL (non-seasonal estimates) in total biomass
print(K)

##     Year totalWt.smCopepods totalWt.lgCopepods totalWt.gelZoop totalWt.micronek
## 1   1985            3431554            3425055      1268670.55        1112159.9
## 5   1986            3440344            3639841       800741.25        1370655.8
## 9   1987            2644889            4744488      1033159.61        2045327.5
## 13  1988            3588997            2888617       412271.91         776389.9
## 17  1989            7195068            1806078       274026.57         421091.0
## 21  1990            6789591            1405306      1075466.97         542611.8
## 25  1991            6417473            2577277       602044.93         905297.4
## 29  1992            4542449            4547651      1886828.09        2991831.8
## 33  1993            4349079            3720062      2503117.60        2363433.1
## 37  1994            4057677            3900582      1537559.39        2134298.4
## 41  1995            3548378            3543290      1248087.65        1386493.7
## 45  1996            6258443            3428739      2749291.72        3543381.0
## 49  1997            6126498            4336470      2760482.99        1809456.5
## 53  1998            6258336            3541391      3264596.80        3592821.6
## 57  1999            5644321            3943443      2367019.98        2445039.3
## 61  2000            8814913            5068131      4873471.86        2462884.7
## 65  2001            5307195            6967732      5461644.81        4754488.9
## 69  2002            5208322            5179505      1880889.14        3153318.9
## 73  2003            3146642            9658619      1175894.95        1137838.2
## 77  2004            3190861            7758209      1581424.72        3057422.3
## 81  2005            1677783            4762070      1026346.53        2826247.1
## 85  2006            4556767            4079822      1638105.96        4068321.2
## 89  2007            2564104            4951487      1541922.26        2002032.9
## 93  2008            3830498            3789274      1780934.58        1449205.0
## 97  2009            2944705            5356916      1375016.81        1549235.4
## 101 2010            3732773            3319225      1785077.23        1956917.6
## 105 2011            3137070            3553507      1577343.09        1346850.0
## 109 2012            7665340            3179961      1246596.34        2778440.6
## 113 2013            5504799            4176236      2766185.10        2665814.0
## 117 2014            3412276            5361754      3201763.47        3040033.3
## 121 2015            6770999            7423748      4309090.36        3672034.8
## 125 2016            5004780            6365594      3457567.62        2104169.9
```

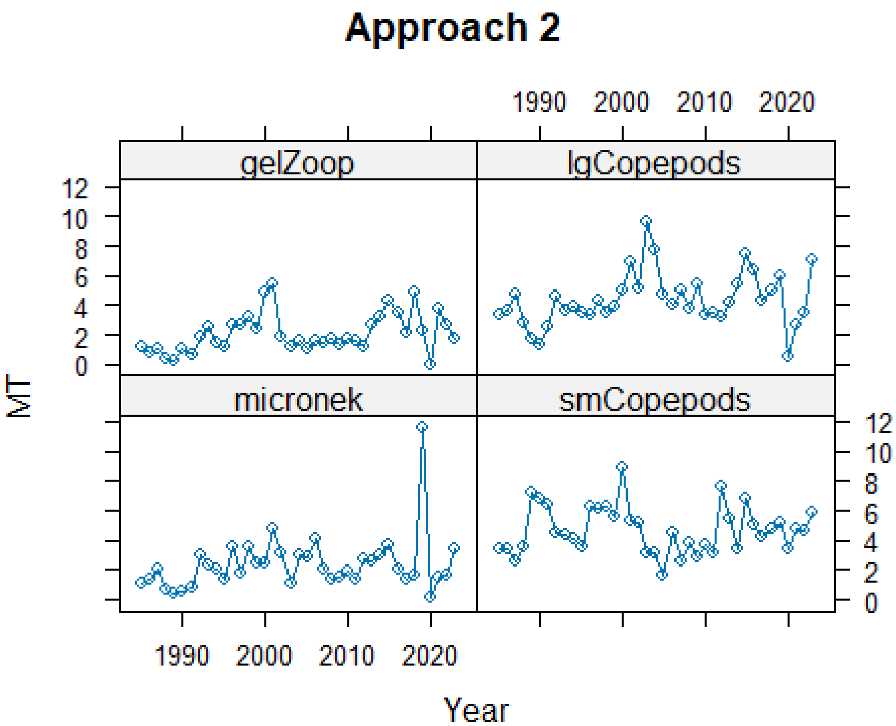
```
## 129 2017           4174560           4289158     2206050.91      1454232.5
## 133 2018           4716700           4953572     4884686.04      1659904.4
## 137 2019           5160204           6022847     2360542.80     11621742.7
## 141 2020           3412234            493223       48392.79       140113.0
## 145 2021           4793319           2732574     3783866.95      1496381.3
## 149 2022           4690930           3560822     2700313.60      1642634.7
## 153 2023           5863249           7071742     1765079.98      3411208.5
```

```
# Look at g/m2 over time across (and mean)                                       # Here ar
e the density estimates that were actually input into NWACS (note that for final input, 2
020 est was omitted):
K[,-1]/sum(epu$areaKm2) #[epu$EPU!="SS","areaKm2"])
```

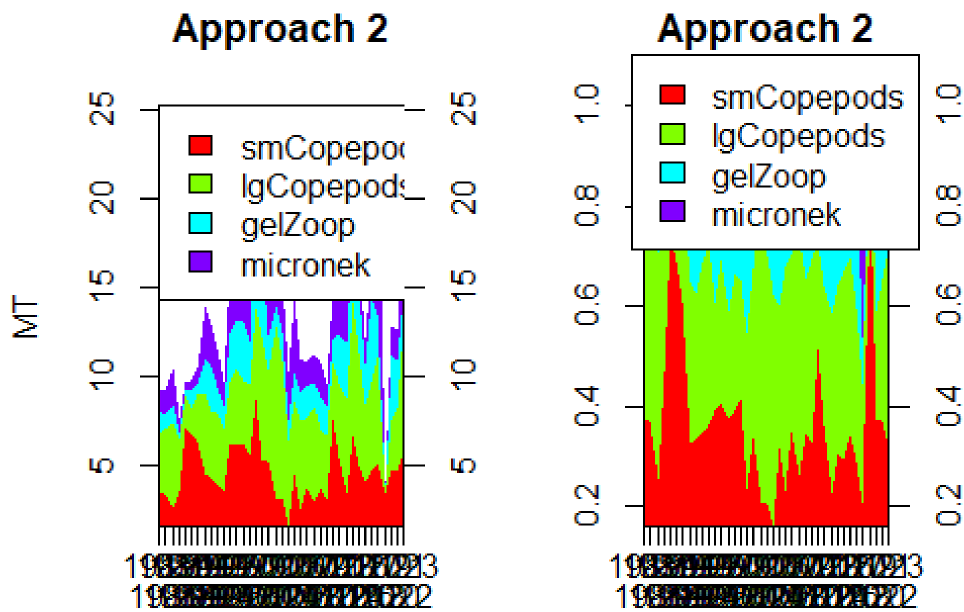
```
##     totalWt.smCopepods totalWt.lgCopepods totalWt.gelZoop totalWt.micronek
## 1            12.138881          12.115890       4.4878329        3.9341876
## 5            12.169975          12.875682       2.8325659        4.8485984
## 9             9.356109          16.783292       3.6547295        7.2352024
## 13           12.695825          10.218281       1.4583829        2.7464247
## 17           25.452046           6.388873       0.9693497        1.4895798
## 21           24.017702           4.971171       3.8043888        1.9194511
## 25           22.701359           9.116937       2.1296916        3.2024259
## 29           16.068592          16.086997       6.6745217       10.5833948
## 33           15.384560          13.159457       8.8546025        8.3604783
## 37           14.353746          13.798035       5.4390083        7.5499307
## 41           12.552137          12.534138       4.4150223        4.9046240
## 45           22.138802          12.128924       9.7254262       12.5344611
## 49           21.672057          15.339957       9.7650145        6.4008253
## 53           22.138424          12.527420      11.5482817       12.7093537
## 57           19.966388          13.949651       8.3731668        8.6491547
## 61           31.182135          17.928157      17.2395641        8.7122814
## 65           18.773828          24.647862      19.3201846       16.8186701
## 69           18.424073          18.322135       6.6535131       11.1546440
## 73           11.131024          34.166685       4.1596457        4.0250227
## 77           11.287446          27.444118       5.5941788       10.8154164
## 81            5.935038          16.845488       3.6306287        9.9976504
## 85           16.119243          14.432085       5.7946847       14.3914003
## 89            9.070339          17.515537       5.4544416        7.0820505
## 93           13.550118          13.404292       6.2999309        5.1264608
## 97           10.416686          18.949713       4.8640253        5.4803114
## 101          13.204420          11.741526       6.3145852        6.9224583
## 105          11.097166          12.570281       5.5797403        4.7643872
## 109          27.115601          11.248887       4.4097469        9.8285384
## 113          19.472840          14.773141       9.7851853        9.4301295
## 117          12.070687          18.966829      11.3260132       10.7539039
## 121          23.951934          26.260988      15.2431042       12.9895650
## 125          17.704056          22.517840      12.2309024        7.4433529
## 129          14.767212          15.172594       7.8037500        5.1442452
## 133          16.684994          17.522913      17.2792335        5.8717950
## 137          18.253858          21.305398       8.3502542       41.1110977
## 141          12.070537           1.744742       0.1711861        0.4956398
## 145          16.956029           9.666290      13.3851633        5.2933436
## 149          16.593836          12.596157       9.5521695        5.8107047
## 153          20.740831          25.015790       6.2438463       12.0669101
```

```
apply(K[,-1]/sum(epu$areaKm2),2,mean)   #(sum(epu[epu$EPU!="SS","areaKm2"])),2,mean)
```

```
## totalWt.smCopepods totalWt.lgCopepods    totalWt.gelZoop   totalWt.micronek
##           16.650783          15.557799           7.456863           8.425592

xyplot((value/1e6)~Year | as.factor(gsub("totalWt.","",variable)), data=melt(K,id="Year")
,as.table=TRUE,type="o",ylab="MT",main="Approach 2")
```




```
stackpoly(x=K[,1], y=K[,-1]/1e6, staxx=TRUE,stack=TRUE,col=rainbow(4),main="Approach 2",y
lab="MT")
legend("topleft",legend=gsub("totalWt.","",c(colnames(K)[-1])),fill=rainbow(4))


stackpoly(x=K[,1], y=K[,-1]/rowSums(K[,-1]), staxx=TRUE,stack=TRUE,col=rainbow(4),main="A
pproach 2")
op <- par(xpd=TRUE)
legend(x=1983,y=1.1,legend=gsub("totalWt.","",c(colnames(K)[-1])),fill=rainbow(4),bg="whi
te")
```




```
# Any relationships among copepods and pteropods? Not run since we have lopped pteropods
#plot(K[-36,c("totalWt.smCopepods","totalWt.pteropod")])
#plot(K[-36,c("totalWt.lgCopepods","totalWt.pteropod")])
#plot(data.frame("copepods"=rowSums(K[-36,c("totalWt.smCopepods","totalWt.lgCopepods")]),
"pteropds"=K[-36,"totalWt.pteropod"]))









# Do *NOT* average over season:                                                   # Februar
y 2025

if(FALSE) {
out3 <-
out2 |>
#   filter(Year==2022) |>  # do a test year to make sure things are working the way I thi
nk they should
   group_by(Year,Season) |>
   summarize(smCopepods=sum(smCopepods,na.rm=TRUE),
   lgCopepods=sum(lgCopepods,na.rm=TRUE),
   gelZoop=sum(gelZoop,na.rm=TRUE),
   micronek=sum(micronek,na.rm=TRUE)) |>
   print(n=Inf,width=Inf)
}

# Equivalent, and much more straightforward than code immediately above:
out3 <-
ddply(out2, .(Year,Season), summarize,                                            # summing
```

```
over EPUs
   smCopepods=sum(smCopepods,na.rm=TRUE),
   lgCopepods=sum(lgCopepods,na.rm=TRUE),
   gelZoop=sum(gelZoop,na.rm=TRUE),
   micronek=sum(micronek,na.rm=TRUE)#,
#   pteropod=sum(pteropod,na.rm=TRUE)
   )

reshape(ddply(out3, .(Year,Season), summarize,
   sumZoop=sum(smCopepods,lgCopepods,gelZoop,micronek)),
   direction="wide",idvar="Year",timevar="Season")                        # Note th
at 0s should actually be NAs

##      Year sumZoop.Fall sumZoop.Spring sumZoop.Summer sumZoop.Winter
## 1    1985     11866997       10583094       10830011        3669655
## 5    1986      7072820       11167990       13981895        4783624
## 9    1987      9334159       16436986       12817552        3282758
## 13   1988      9263438        2339411       10212986        5162853
## 17   1989     12703981              0              0        7530727
## 21   1990     13823029        3316883        7070015        7342413
## 25   1991     15427760              0       13022019        3961795
## 29   1992     11415652       22037082       16922278        5500028
## 33   1993     14252968       19410488        8675370        4910128
## 37   1994     13643757       18013068        3439447        4738713
## 41   1995      8921034       12194187       12709408        5080367
## 45   1996     13924069       29349385       16207159        3321683
## 49   1997     16218188        9908675       23972698        5246246
## 53   1998     16363006       27770950        7994368        8077239
## 57   1999     17636641       11116963       17800414        7802163
## 61   2000     21125320       20838111       24911313        6876127
## 65   2001     12160602       46150646       16497493        7032286
## 69   2002     12205263       22928892       16386076       10167909
## 73   2003     11840605       26138413       15470687        7026271
## 77   2004     10016974       21947399       19250947       11136346
## 81   2005      5787924       16182318       11572917        7626626
## 85   2006     13753976       18670034       17394599        7553457
## 89   2007      9048197       16898982       10624785        7666222
## 93   2008     11197535        7639341       16783860        7778912
## 97   2009      8930440       15168377       15746996        5057678
## 101  2010      9041848       13863149       15664961        4606013
## 105  2011     12322879       13101685        3209166        4382965
## 109  2012     12679288       11810192       28189457        6802414
## 113  2013     14175211       23036242       16010903        5655751
## 117  2014     12501512       28063529        4837925        2684549
## 121  2015     19404099       29098999        5260569        7141123
## 125  2016     12206475       24948716       15745312              0
## 129  2017     13197754       19483093              0        5145389
## 133  2018     13337477       19219627       10569852        4454481
## 137  2019     55576794       10211181       22871891        2832654
## 141  2020            0              0              0        4514301
## 145  2021     15641733       10056105       14929678        2171441
## 149  2022     14261808       15312413        3032964        1755834
## 153  2023      8524460       20006440       29214146              0

                                                                           # Example
: sum(c(NA,NA,NA),na.rm=TRUE) sum is actually NA, not 0!
```

```r
# fin!
```

```r
reshape(out3, direction="wide",idvar=c("Year"),timevar="Season")
```

```
##     Year smCopepods.Fall lgCopepods.Fall gelZoop.Fall micronek.Fall
## 1   1985         5621503         4376674     455355.3     1413464.2
## 5   1986         3301310         2452584     358975.5      959950.1
## 9   1987         3595523         2494079     103732.2     3140825.7
## 13  1988         5376774         2595919     101248.5     1189496.3
## 17  1989         8976746         2162722     319553.3     1244960.0
## 21  1990        11528095         1083204     420287.2      791442.1
## 25  1991        10629353         1929672     763738.8     2104996.3
## 29  1992         6395641         2235428     395506.7     2389075.1
## 33  1993         9878515         1980666     595296.4     1798490.5
## 37  1994         8884102         2502210     621298.1     1636145.9
## 41  1995         4589340         2358530     755111.3     1218052.7
## 45  1996         9842861         1989404    1015246.2     1076557.5
## 49  1997         9267882         2481839    1796712.1     2671754.2
## 53  1998         9959304         2943445    1510750.7     1949507.4
## 57  1999         8820127         3592861    1040202.6     4183450.0
## 61  2000        14349206         2097658    2150981.0     2527475.4
## 65  2001         3936137         5373462     752182.2     2098820.9
## 69  2002         6711774         3028381     563399.7     1901707.7
## 73  2003         3994491         4843060     912567.4     2090486.9
## 77  2004         3167717         3714475     355883.2     2778897.9
## 81  2005         1321187         2446586     346818.0     1673332.6
## 85  2006         6364806         3181383    1527853.9     2679933.4
## 89  2007         2656704         3873597     920312.0     1597583.1
## 93  2008         4473059         3435091     890056.1     2399329.4
## 97  2009         3872856         2948010     419687.0     1689886.6
## 101 2010         4609861         1762801     528706.9     2140478.5
## 105 2011         7282673         2411735     495579.9     2132891.1
## 109 2012         6322967         1457182    1599668.5     3299470.6
## 113 2013         9092327         1895519    1240664.1     1946700.6
## 117 2014         6325173         2051316     927106.8     3197916.8
## 121 2015        11746887         2492056    2647050.3     2518106.3
## 125 2016         4970044         2114298    2622446.0     2499687.3
## 129 2017         7861816         1867047    1367351.8     2101538.8
## 133 2018         5758145         2337609    2170168.6     3071554.5
## 137 2019         6695993         4573240    3817228.6    40490332.1
## 141 2020               0               0          0.0           0.0
## 145 2021         8363790         1126134    4227001.4     1924808.7
## 149 2022         8670437         1495086    1768366.0     2327918.1
## 153 2023         3594486         3224390     596598.4     1108986.2
##     smCopepods.Spring lgCopepods.Spring gelZoop.Spring micronek.Spring
## 1          1541976.87           4662643      3142056.6      1236417.78
## 5          1300087.61           5875113      2023519.4      1969270.01
## 9           717818.47           9865187      2915015.5      2938965.06
## 13           53466.69           1668514       582171.2        35259.06
## 17               0.00                 0            0.0            0.00
## 21           67376.75           2274950       974556.2            0.00
## 25               0.00                 0            0.0            0.00
```

```
## 29         323263.47          8334016     6133949.2       7245852.82
## 33         479016.10          8241651     4913743.5       5776077.50
## 37         456116.87          8683843     3455742.6       5417365.24
## 41         975400.14          5741663     3030407.9       2446715.89
## 45         756765.26          9367878     7378321.2      11846421.31
## 49          85611.52          5133252     3216757.1       1473053.75
## 53        2955177.27          6267363     6752357.6      11796052.31
## 57         179184.47          5798914     4245462.4        893401.98
## 61        1901646.06          9245905     5207867.6       4482692.54
## 65        4549640.78          9976344    18292041.6      13332619.59
## 69        1248308.71          9143868     5067795.2       7468919.43
## 73         653234.68         23151976     2169522.8        163678.87
## 77         658732.43         11522486     4218758.7       5547421.78
## 81          69527.66          7737234     1214846.9       7160709.64
## 85         245265.63          6157693     1789232.0      10477843.97
## 89        1515370.48          8292069     3530016.3       3561526.20
## 93         458269.07          3744001     2969605.7        467465.20
## 97         294155.91         10322958     2632619.5       1918643.39
## 101         87019.65          6427073     4193349.3       3155706.17
## 105        370142.40          7169647     3558455.3       2003440.18
## 109       1630409.80          6931089     2013936.9       1234756.05
## 113       1363657.25          9046559     6005879.9       6620145.47
## 117       2030980.54         12386230     6542267.8       7104050.82
## 121       1098212.45         13070394     5970654.0       8959738.96
## 125       2356420.78         12088228     6117317.7       4386749.31
## 129       1466454.72          9780896     4954424.0       3281318.69
## 133       1443448.50          8765503     6956546.1       2054128.94
## 137        368744.55          5865879     2056461.3       1920096.36
## 141             0.00                0           0.0             0.00
## 145       1834812.92          5379498     1605314.8       1236479.19
## 149       1600206.14          6681764     4199220.0       2831222.67
## 153       2193581.05          9684622     3650999.7       4477237.03
##     smCopepods.Summer lgCopepods.Summer gelZoop.Summer micronek.Summer
## 1           4770690.0         3575147.0      1105641.3       1378532.5
## 5           6859422.7         4857373.6       773376.1       1491722.2
## 9           4476390.4         5611511.7       974621.0       1755029.2
## 13          4234480.9         4071853.5       456415.9       1450236.0
## 17                0.0               0.0            0.0             0.0
## 21          5223545.7          130224.4       730566.3        985678.4
## 25          5375246.1         5333215.1       960073.2       1353484.6
## 29          7651072.2         6605512.4       579219.9       2086473.1
## 33          3320963.3          928036.2      3028339.5       1398030.9
## 37          1090095.4          482441.1       889869.6        977040.6
## 41          5376662.4         4922571.9       994805.9       1415367.5
## 45         11512403.6         1454158.3      2089910.8       1150686.6
## 49          8761497.4         8533924.2      4139568.6       2537708.2
## 53          4289270.5         1767346.1      1484272.7        453478.6
## 57          6205203.9         4021180.5      3362821.0       4211208.6
## 61          9347155.1         4347447.1      9216593.2       2000117.7
## 65          5854317.5         6353505.9      1422580.6       2867089.5
## 69          6709206.7         5365103.6      1757796.6       2553969.3
## 73          3609244.6         8158309.6      1576329.8       2126802.8
## 77          4816812.5         9691842.6      1383457.4       3358834.4
## 81          2897362.8         5384763.2      1023103.4       2267687.8
## 85          7664489.8         4395608.8      2815595.6       2518905.0
```

```
## 89         2687738.8         4845914.6       1041929.4        2049201.9
## 93         5910713.3         5265246.0       3147579.0        2460321.4
## 97         4703213.8         6732203.0       2289519.6        2022059.2
## 101        8532359.8         3316669.0       1941624.1        1874308.3
## 105         613654.1          692455.9        859179.4        1043876.5
## 109       17896283.1         3248194.1       1257496.7        5787482.9
## 113        7093095.4         4218738.3       3074069.6        1624999.8
## 117        1058762.5          430911.2       1598845.5        1749405.5
## 121        1774333.5          381969.7       1658126.2        1446139.9
## 125        7687874.8         4894254.9       1632939.2        1530242.8
## 129              0.0               0.0             0.0              0.0
## 133        3476359.7         1079127.5       4603758.3        1410606.3
## 137        8329318.5         8046493.0       2704177.8        3791901.6
## 141              0.0               0.0             0.0              0.0
## 145        4173017.4         2291133.6       6506260.2        1959266.6
## 149         886535.5          202388.7        977117.3         966922.7
## 153       11801680.0         8306213.8       1047641.9        8058610.6
##     smCopepods.Winter lgCopepods.Winter gelZoop.Winter micronek.Winter
## 1             1792046        1085754.98     371628.974       420225.28
## 5             2300555        1374293.46      47093.963      1061680.68
## 9             1789825        1007173.18     139269.784       346490.27
## 13            2250624        2189827.40     291832.667       430568.20
## 17            5413389        1449433.40     228499.892       439404.05
## 21            4685648        1027765.26    1235672.669       393326.68
## 25            3247819         468944.55      82322.839       162708.72
## 29            3799817        1015648.61     438636.539       245926.22
## 33            2346655        1300720.99     781619.045       481133.33
## 37            3307254         674105.77     250711.903       506642.00
## 41            3252108        1150394.65     212025.578       465838.73
## 45            2705113         318386.37     198324.482        99858.49
## 49            3479846         767274.36     443815.767       555309.96
## 53            4516016        1853713.60    1535261.110       172247.91
## 57            5127682        1744333.77     438050.861       492096.63
## 61            5149091         729772.34     156009.708       841253.13
## 65            5238857         740540.17     333463.460       719425.41
## 69            6163999        3180666.54     134565.081       688679.16
## 73            4329597        2481129.83      45159.749       170384.31
## 77            4120181        6104030.45     367599.551       544535.04
## 81            2423053        3479696.56    1520617.880       203258.32
## 85            3952507        2584605.08     419742.355       596602.47
## 89            3396604        2794366.69     675431.339       799820.31
## 93            4479951        2712759.41     116497.576       469703.98
## 97            2908593        1424491.91     158241.169       566352.59
## 101           1701850        1770357.63     476628.635       657177.30
## 105           2720924         978918.04     475931.112       207192.18
## 109           4811700        1083377.91     115283.175       792053.03
## 113           3930439        1023197.48     230704.765       471410.10
## 117            821913        1216806.02     537070.377       108759.84
## 121           3282847         385338.24    1708783.782      1764154.07
## 125                 0              0.00          0.000            0.00
## 129           3195409        1219529.93     296376.939       434072.47
## 133           4062941         283742.53       4469.333       103327.69
## 137           1448152        1048431.63      51430.116       284640.63
## 141           3412234         493222.97      48392.792       560451.91
## 145           1130863         126920.76      48686.357       864970.77
```

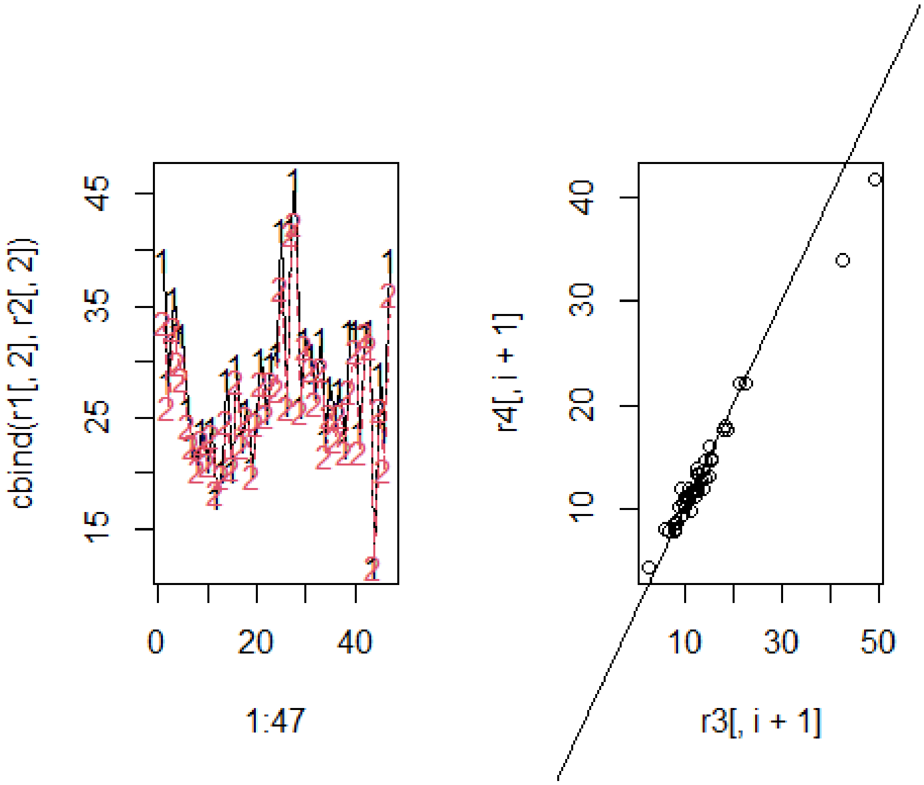
```
## 149          1203892          63270.92      44196.003       444475.30
## 153                0              0.00          0.000            0.00
```

```r
H <- reshape(G,direction="wide", idvar=c("Year","spp"),timevar="EPU")

J <- data.frame(H[,c(1:2)],"totalWt"=rowSums(H[,-c(1,2)],na.rm=TRUE))

K <- reshape(J,direction="wide",idvar="Year",timevar="spp")

# Look at g/m2 over time across (and mean)
K[,-1]/sum(epu$areaKm2)#[epu$EPU!="SS","areaKm2"])
```

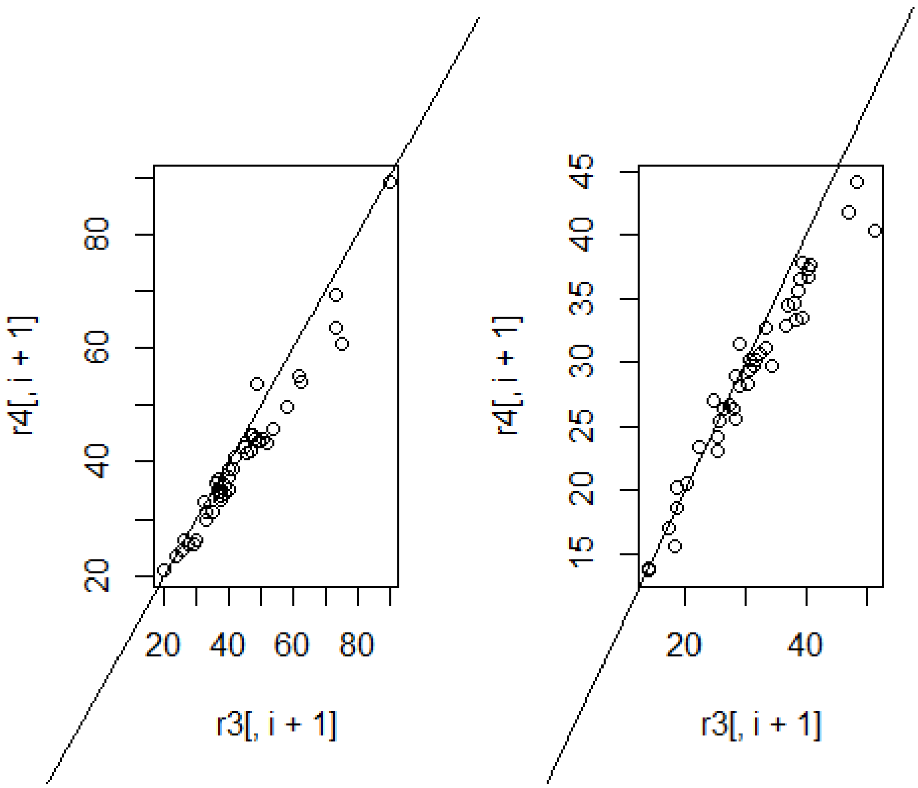
```
##     totalWt.smCopepods totalWt.lgCopepods totalWt.gelZoop totalWt.micronek
## 1            12.138881          12.115890       4.4878329        3.9341876
## 5            12.169975          12.875682       2.8325659        4.8485984
## 9             9.356109          16.783292       3.6547295        7.2352024
## 13           12.695825          10.218281       1.4583829        2.7464247
## 17           25.452046           6.388873       0.9693497        1.4895798
## 21           24.017702           4.971171       3.8043888        1.9194511
## 25           22.701359           9.116937       2.1296916        3.2024259
## 29           16.068592          16.086997       6.6745217       10.5833948
## 33           15.384560          13.159457       8.8546025        8.3604783
## 37           14.353746          13.798035       5.4390083        7.5499307
## 41           12.552137          12.534138       4.4150223        4.9046240
## 45           22.138802          12.128924       9.7254262       12.5344611
## 49           21.672057          15.339957       9.7650145        6.4008253
## 53           22.138424          12.527420      11.5482817       12.7093537
## 57           19.966388          13.949651       8.3731668        8.6491547
## 61           31.182135          17.928157      17.2395641        8.7122814
## 65           18.773828          24.647862      19.3201846       16.8186701
## 69           18.424073          18.322135       6.6535131       11.1546440
## 73           11.131024          34.166685       4.1596457        4.0250227
## 77           11.287446          27.444118       5.5941788       10.8154164
## 81            5.935038          16.845488       3.6306287        9.9976504
## 85           16.119243          14.432085       5.7946847       14.3914003
## 89            9.070339          17.515537       5.4544416        7.0820505
## 93           13.550118          13.404292       6.2999309        5.1264608
## 97           10.416686          18.949713       4.8640253        5.4803114
## 101          13.204420          11.741526       6.3145852        6.9224583
## 105          11.097166          12.570281       5.5797403        4.7643872
## 109          27.115601          11.248887       4.4097469        9.8285384
## 113          19.472840          14.773141       9.7851853        9.4301295
## 117          12.070687          18.966829      11.3260132       10.7539039
## 121          23.951934          26.260988      15.2431042       12.9895650
## 125          17.704056          22.517840      12.2309024        7.4433529
## 129          14.767212          15.172594       7.8037500        5.1442452
## 133          16.684994          17.522913      17.2792335        5.8717950
## 137          18.253858          21.305398       8.3502542       41.1110977
## 141          12.070537           1.744742       0.1711861        0.4956398
## 145          16.956029           9.666290      13.3851633        5.2933436
## 149          16.593836          12.596157       9.5521695        5.8107047
## 153          20.740831          25.015790       6.2438463       12.0669101
```

```r
apply(K[,-1]/(sum(epu$areaKm2)),2,mean)#[epu$EPU!="SS","areaKm2"])),2,mean)
```

```
## totalWt.smCopepods totalWt.lgCopepods    totalWt.gelZoop   totalWt.micronek
##          16.650783          15.557799           7.456863           8.425592

################################################################################
# Biovolume:
.seasons <- data.frame("Season"=c("Winter","Spring","Summer","Fall"),"Qs"=1:4)
aa <- merge(a,.seasons,all.x=TRUE)

bv <- merge(aa[,c("Year","Season","Qs","EPU","Biovolume")],epu,all.x=TRUE)

# annual biovolume, averaged over all seasons and regions
r1 <- ddply(bv, .(Year), summarize,
   meanBv=mean(Biovolume,na.rm=TRUE))

# mean weighted by size of EPU:
r2 <- ddply(bv, .(Year), summarize,
   meanBv=weighted.mean(Biovolume,areaKm2,na.rm=TRUE))

matplot(1:47,cbind(r1[,2],r2[,2]),type="o")
cor(cbind(r1[,2],r2[,2]),method="spearman")

##          [,1]     [,2]
## [1,] 1.000000 0.970629
## [2,] 0.970629 1.000000

sbv <- ddply(bv, .(Year,Qs), summarize,                                    # seasona
l biovolume, weighted by EPU area
   meanBv=weighted.mean(Biovolume,na.rm=TRUE))

sbv.w <- ddply(bv, .(Year,Qs), summarize,                                  # seasona
l biovolume, weighted by EPU area
   meanBv=weighted.mean(Biovolume,areaKm2,na.rm=TRUE))

r3 <- reshape(sbv,direction="wide",idvar="Year",timevar="Qs")
r4 <- reshape(sbv.w,direction="wide",idvar="Year",timevar="Qs")

op <- par(ask=FALSE)
for(i in 1:4) {
   plot(r3[,i+1],r4[,i+1]); abline(lm(y ~ x, data = data.frame(x = 1:10, y = 1:10)))
   print(cor(r3[,i+1],r4[,i+1]))
}
```

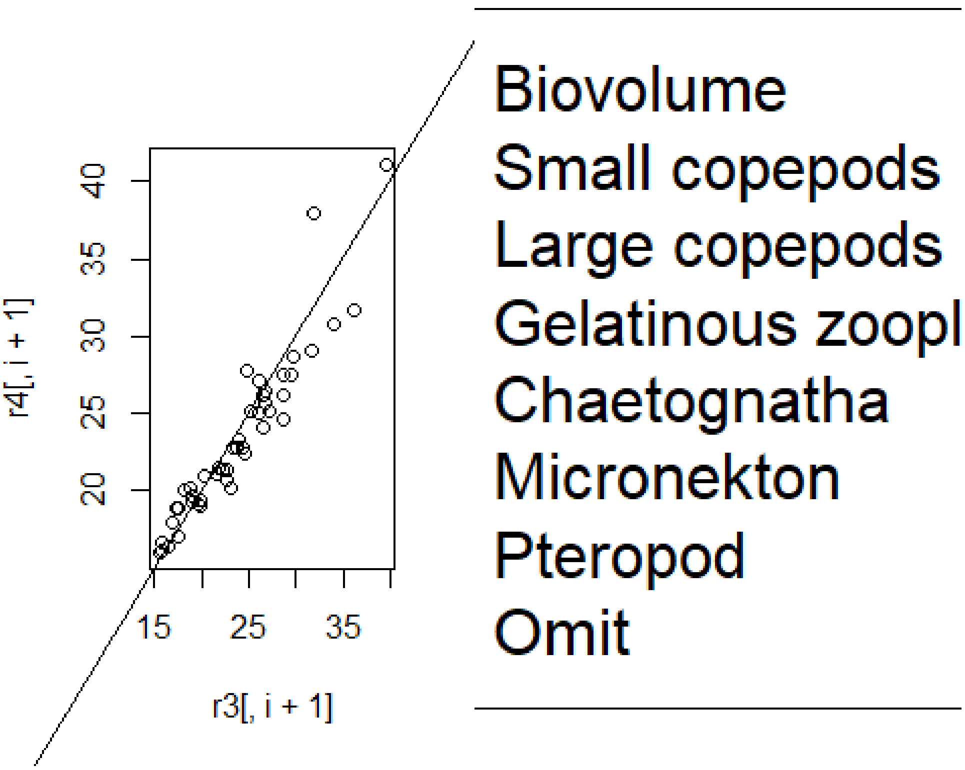



```
## [1] NA
## [1] NA
```

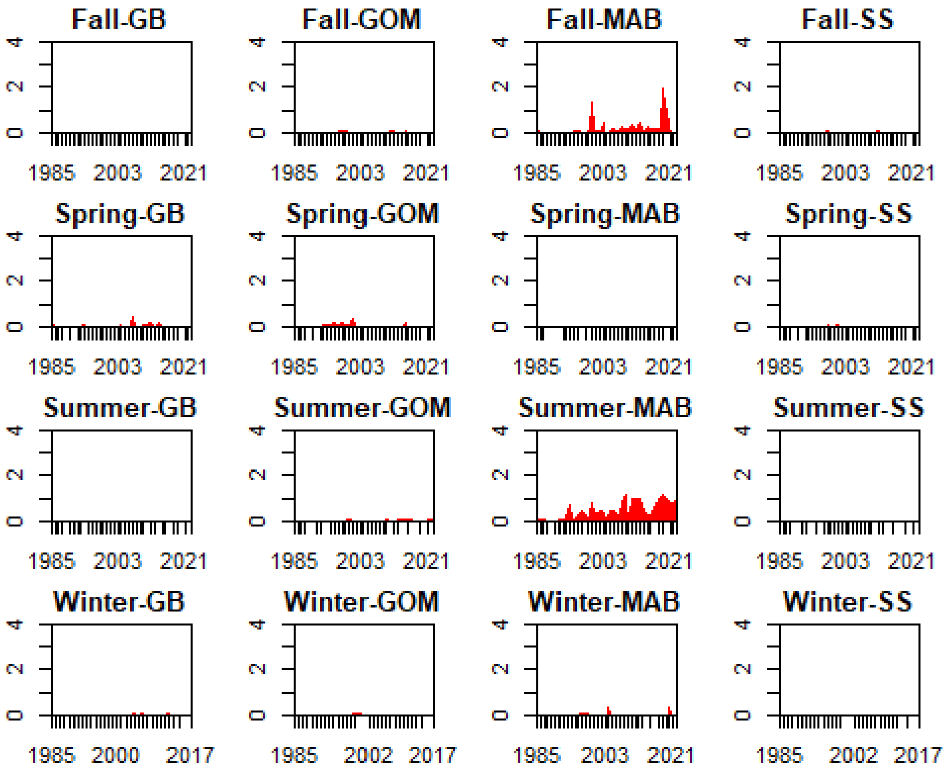



```
## [1] NA
## [1] NA
### Miscellany:
```

```
plot(1,1,type="n",xlab="",ylab="",axes=FALSE)
legend("center",legend=colnames(a)[-c(1:3)],fill=rainbow(7),cex=2)
```

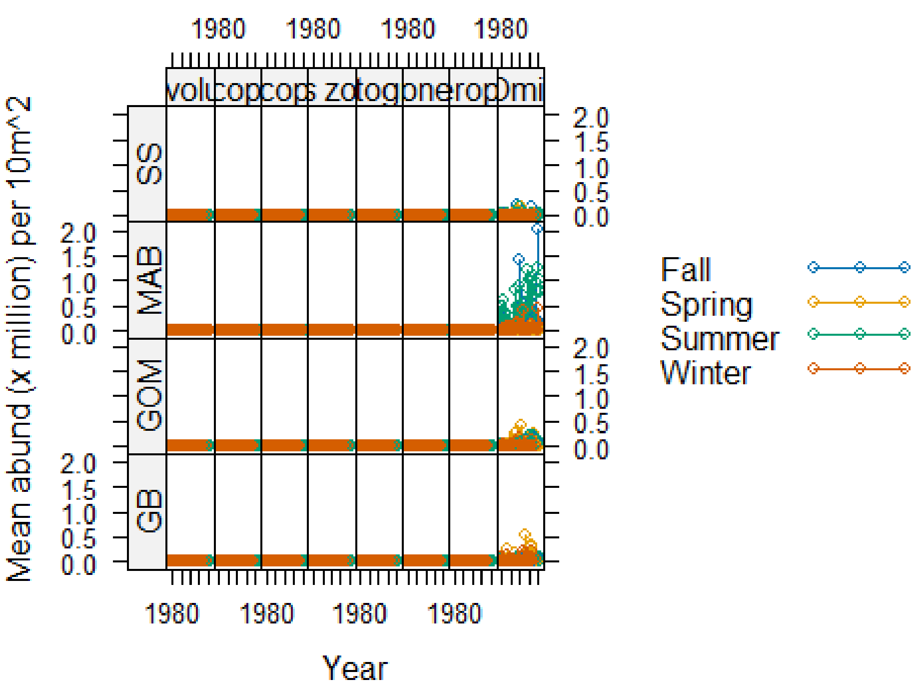



```
op <- par(mfrow=c(4,4),mar=c(2,2,2,2))
lapply(split(a[a$Year>1984,], paste(a[a$Year>1984,"Season"],a[a$Year>1984,"EPU"],sep="-")
), function(x) {
#   stackpoly(x$Year,x[,-c(1:3)]/rowSums(x[,-c(1:3)]))
   x <- x[-which(is.na(x$`Small copepods`)),]
   stackpoly(x$Year,(x[,-c(1:3)]/1e6),stack=TRUE,col=rainbow(7),
      main=paste(x$Season[1],x$EPU[1],sep="-"),
#      ylab="Abund (x million) per 10^m2",
      staxx=FALSE,axis4=FALSE,ylim=c(0,4))
#   legend("topright",legend=colnames(a)[-c(1:3)],fill=rainbow(7))
   }
)
```

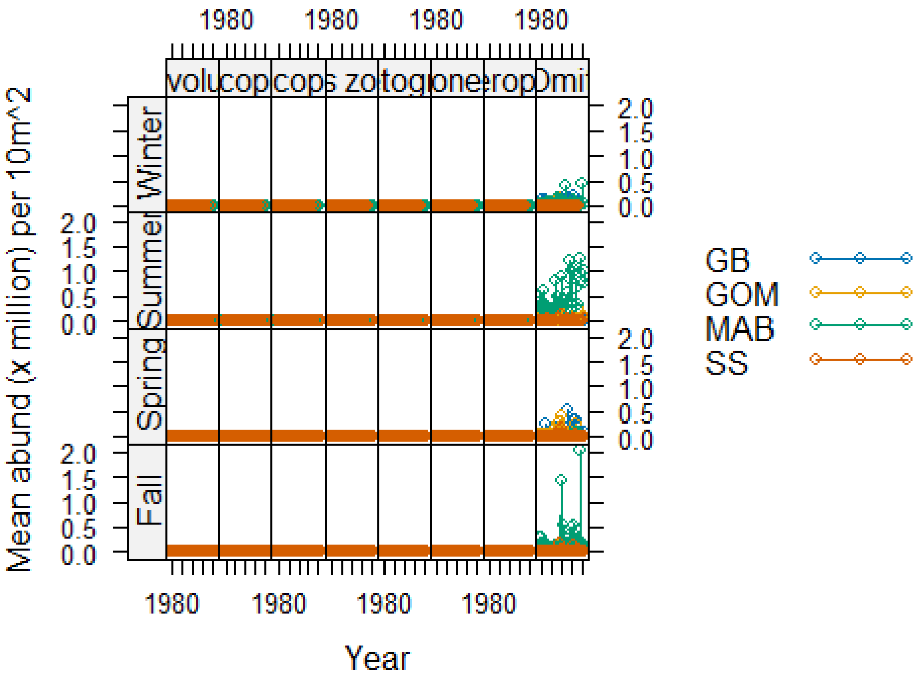



```
## $`Fall-GB`
## NULL
##
## $`Fall-GOM`
## NULL
##
## $`Fall-MAB`
## NULL
##
## $`Fall-SS`
## NULL
##
## $`Spring-GB`
## NULL
##
## $`Spring-GOM`
## NULL
##
## $`Spring-MAB`
## NULL
##
## $`Spring-SS`
## NULL
##
## $`Summer-GB`
## NULL
##
## $`Summer-GOM`
## NULL
##
## $`Summer-MAB`
## NULL
```

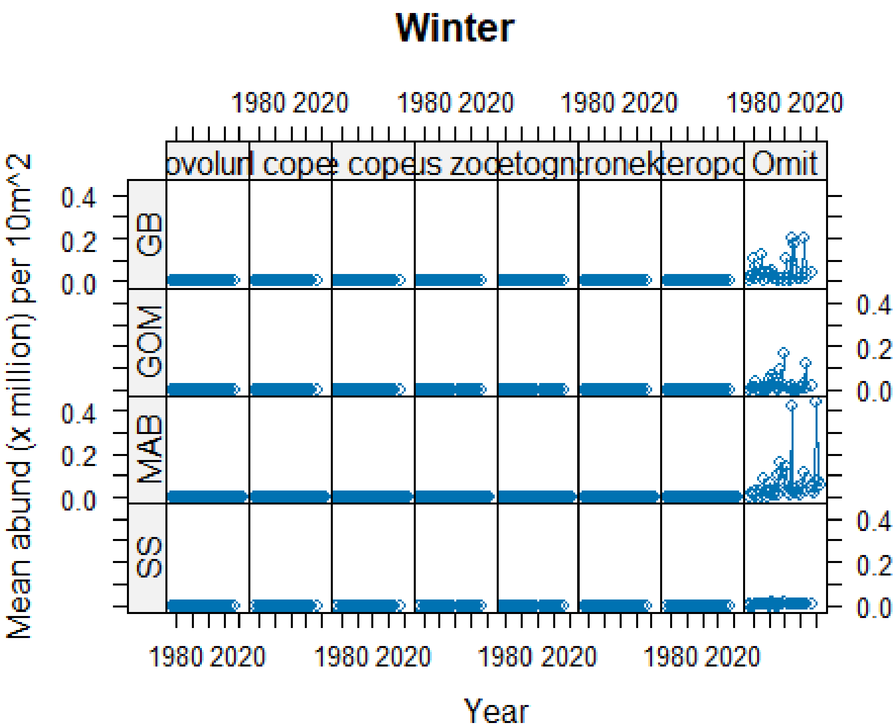
```
##
## $`Summer-SS`
## NULL
##
## $`Winter-GB`
## NULL
##
## $`Winter-GOM`
## NULL
##
## $`Winter-MAB`
## NULL
##
## $`Winter-SS`
## NULL

par(op)

A <- melt(a,id=c("Year","Season","EPU"))

useOuterStrips(xyplot((value/1e6)~Year | variable*EPU, groups=Season, data=A,type="o",aut
o.key=TRUE,ylab="Mean abund (x million) per 10m^2"))
```

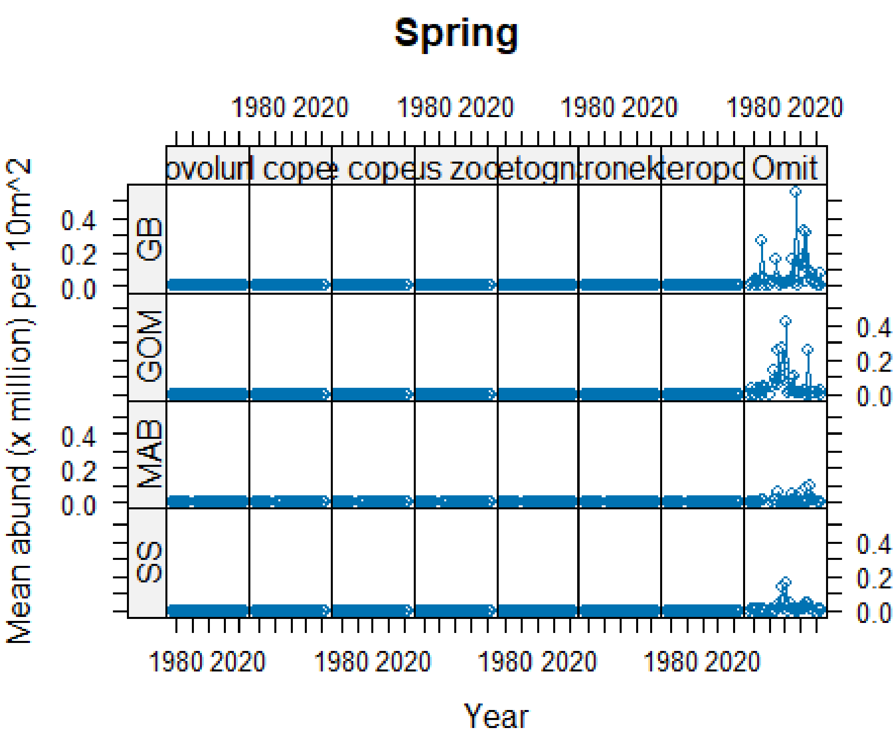


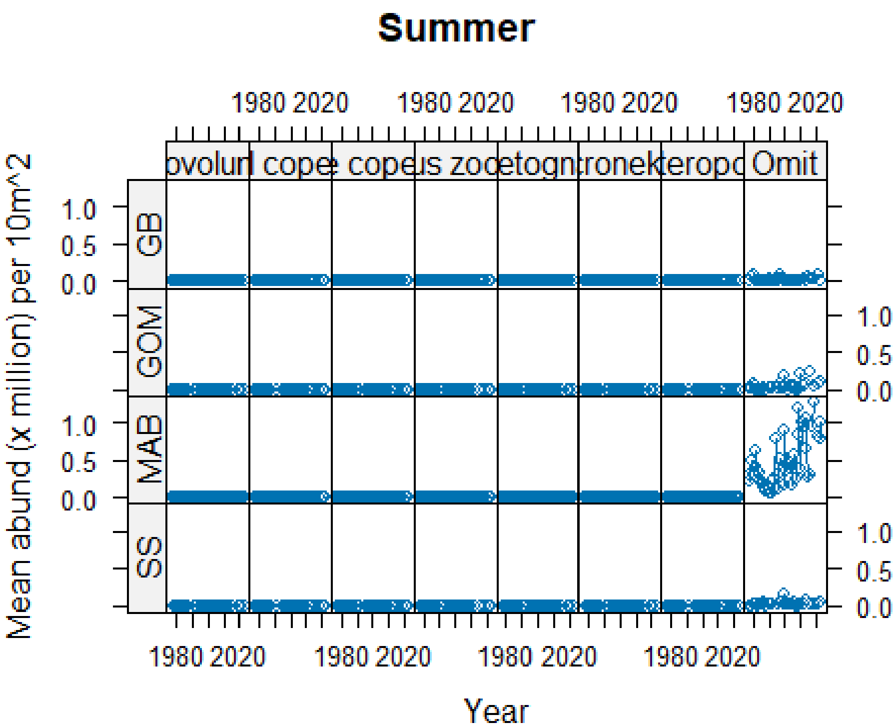

```
                                                                      #A[A$variable!
="Omit",]
useOuterStrips(xyplot((value/1e6)~Year | variable*Season, groups=EPU, data=A,type="o",aut
o.key=TRUE,ylab="Mean abund (x million) per 10m^2"))
```

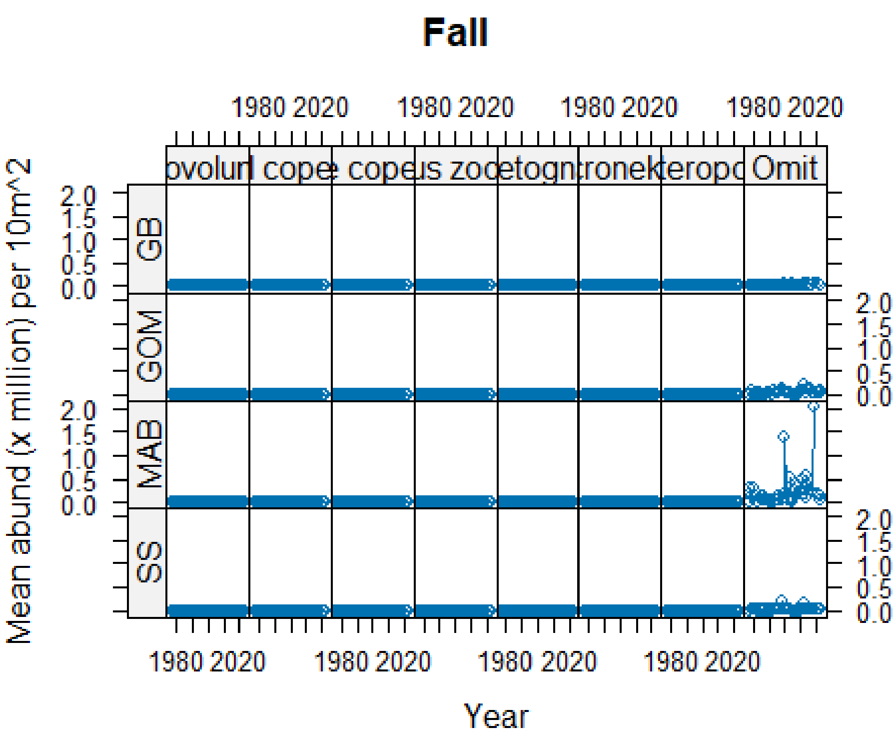



```
for(i in c("Winter","Spring","Summer","Fall")) {
   print(useOuterStrips(xyplot((value/1e6)~Year | variable*EPU, data=A[A$Season==i,],
      type="o",auto.key=TRUE,as.table=TRUE,main=i,ylab="Mean abund (x million) per 10m^2"
)))
}
```


Winter
Mean abund (x million) per 10m^2
Year
GB
GOM
MAB
SS
Omit
1980 2020


Spring
Mean abund (x million) per 10m^2
Year
GB
GOM
MAB
SS
Omit
1980 2020


Summer
1980 2020
ovolun
l cope
e cope
us zoo
etogn
ronek
eropo
Omit
GB
GOM
MAB
SS
1.0
0.5
0.0
Mean abund (x million) per 10m^2
Year


Fall
1980 2020
ovolun
l cope
e cope
us zoo
etogn
ronek
eropo
Omit
GB
GOM
MAB
SS
2.0
1.5
1.0
0.5
0.0
Mean abund (x million) per 10m^2
Year