Prepare stomach content data

AUTHOR
Max Lindmark

PUBLISHED
September 12, 2023

Load libraries

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(tidylog)
library(janitor)
library(devtools)
library(sdmTMB)
library(readr)

# Import some plotting functions
# Source code for map plots
# You need: # devtools::install_github("seananderson/ggsidekick") # not on CRAN; library(
devtools::source_url("https://raw.githubusercontent.com/maxlindmark/pred-prey-overlap/mai
options(ggplot2.continuous.colour = "viridis")

# Set path
home <- here::here()
```

Read data

I downloaded stomach data from the Baltic, all countries, from year 1993-now, on the morning of Sep 12th. I see lots of new and old data missing. Here's a test script to prepare those data. Text in red are warnings/questions from me.

In our case study, we want to create a response variable that is the weight ratio of saduria, herring and sprat (separately) to cod weight. Hence we need, per predator, the total weight of these species, and if they are absent, weight should be 0. We also need coordinates for these stomachs. To get there, we need to do the follow:

1. Join all data sets
2. Filter predators with the above prey present and calculate the total weight of these for each predator
3. Filter predators where these prey are not present.
4. Bind rows, and replace NA with 0

```
fi <- read.csv(paste0(home, "/data/stomach/StomachContent_0912160542/File_information.csv
hi <- read.csv(paste0(home, "/data/stomach/StomachContent_0912160542/HaulInformation.csv"
```

```
pred <- read.csv(paste0(home, "/data/stomach/StomachContent_0912160542/PredatorInformatio
prey <- read.csv(paste0(home, "/data/stomach/StomachContent_0912160542/PreyInformation.cs
```

Have a look at the data... The description of the data can be found here: http://datsu.ices.dk/web/selRep.aspx?Dataset=157

```
names(fi)
```

```
[1] "tblUploadID"           "Country"               "Reporting_organisation"
[4] "CruiseID"
```

```
names(hi)
```

```
 [1] "tblUploadID"   "tblHaulID"     "Ship"          "Gear"
 [5] "HaulNo"        "StationNumber" "Year"          "Month"
 [9] "Day"           "Time"          "ShootLat"      "ShootLong"
[13] "HaulLat"       "HaulLong"      "ICESrectangle" "Depth"
[17] "Survey"        "ICESDatabase"  "Notes"
```

```
names(pred)
```

```
 [1] "tblUploadID"              "tblHaulID"
 [3] "tblPredatorInformationID" "Ship"
 [5] "Gear"                     "HaulNo"
 [7] "StationNumber"            "Year"
 [9] "Month"                    "Day"
[11] "Time"                     "FishID"
[13] "AphiaIDPredator"          "IndWgt"
[15] "Number"                   "MeasurementIncrement"
[17] "Length"                   "Code"
[19] "Age"                      "Sex"
[21] "MaturityScale"            "MaturityStage"
[23] "PreservationMethod"       "Regurgitated"
[25] "StomachFullness"          "FullStomWgt"
[27] "EmptyStomWgt"             "StomachEmpty"
[29] "GenSamp"                  "Notes"
```

```
names(prey)
```

```
 [1] "tblUploadID"              "tblHaulID"
 [3] "tblPredatorInformationID" "tblPreyInformationID"
 [5] "Ship"                     "Gear"
 [7] "HaulNo"                   "StationNumber"
 [9] "Year"                     "Month"
[11] "Day"                      "Time"
[13] "FishID"                   "AphiaIDPredator"
[15] "AphiaIDPrey"              "IdentMet"
[17] "DigestionStage"           "GravMethod"
[19] "SubFactor"                "PreySequence"
```

```
[21] "Count"                    "UnitWgt"
[23] "Weight"                   "UnitLngt"
[25] "Length"                   "OtherItems"
[27] "OtherCount"               "OtherWgt"
[29] "AnalysingOrg"             "Notes"
```

Join all data files

We do this specific order: fi -> hi -> pred -> prey.

For some joins, there are multiple column names shared in addition to the key. I suppose I could remove them and keep only the ID key and the non-shared columns, but in I could also keep them. First I need to ensure they are the same, and not only have the same name though. Will also check if both datasets have the same amount of NA before choosing which column to carry from which dataset.

```
hi <- left_join(hi, fi, by = "tblUploadID")
```

```
left_join: added 3 columns (Country, Reporting_organisation, CruiseID)

           > rows only in x     0

           > rows only in y  (  0)

           > matched rows     251

           >                 =====

           > rows total       251
```

```
comcol_hi_pred <- intersect(colnames(pred), colnames(hi))

# Check if any of the two datasets have NA in the common columns
unique(is.na(hi |> dplyr::select(all_of(comcol_hi_pred))))
```

```
     tblUploadID tblHaulID  Ship  Gear HaulNo StationNumber  Year Month   Day
[1,]       FALSE     FALSE FALSE FALSE  FALSE         FALSE FALSE FALSE FALSE
      Time Notes
[1,] FALSE FALSE
```

```
unique(is.na(pred |> dplyr::select(all_of(comcol_hi_pred))))
```

```
     tblUploadID tblHaulID  Ship  Gear HaulNo StationNumber  Year Month   Day
[1,]       FALSE     FALSE FALSE FALSE  FALSE         FALSE FALSE FALSE FALSE
      Time Notes
[1,] FALSE FALSE
```

```
# Nope, but the column Notes does have different meanings so we will remove that before j
pred <- left_join(pred |> dplyr::select(-Notes),
                  hi |> dplyr::select(-Notes),
                  by = comcol_hi_pred[!comcol_hi_pred == "Notes"])
```

```
left_join: added 11 columns (ShootLat, ShootLong, HaulLat, HaulLong, ICESrectangle, …)
           > rows only in x       0
           > rows only in y  (   29)
           > matched rows     3,295
           >                =======
           > rows total       3,295
```

Interesting, here we see that there are 29 rows that only exist in the haul data, not in the predator data… I wonder why they are included here in the download file if they do not have a match in the predator data. Anyway, the other way around would have been worse. Not too concerned because at least all predators have their haul information now.

Now let's join predator data to prey data following the same procedure.

```
intersect(colnames(pred), colnames(prey))
```

```
 [1] "tblUploadID"              "tblHaulID"
 [3] "tblPredatorInformationID" "Ship"
 [5] "Gear"                     "HaulNo"
 [7] "StationNumber"            "Year"
 [9] "Month"                    "Day"
[11] "Time"                     "FishID"
[13] "AphiaIDPredator"          "Length"
```

```
# Length is a common column, but it corresponds to predator or prey. Rename!
pred <- pred |> rename(pred_length = Length)
```

```
rename: renamed one variable (pred_length)
```

```
prey <- prey |> rename(prey_length = Length)
```

```
rename: renamed one variable (prey_length)
```

```
comcol_prey_pred <- intersect(colnames(pred), colnames(prey))
comcol_prey_pred
```

```
 [1] "tblUploadID"              "tblHaulID"
 [3] "tblPredatorInformationID" "Ship"
```

```
 [5] "Gear"                    "HaulNo"
 [7] "StationNumber"           "Year"
 [9] "Month"                   "Day"
[11] "Time"                    "FishID"
[13] "AphiaIDPredator"
```

```
# Check if any of the two datasets have NA in the common columns
unique(is.na(pred |> dplyr::select(all_of(comcol_prey_pred))))
```

```
     tblUploadID tblHaulID tblPredatorInformationID  Ship  Gear HaulNo
[1,]       FALSE     FALSE                    FALSE FALSE FALSE  FALSE
     StationNumber  Year Month   Day  Time FishID AphiaIDPredator
[1,]         FALSE FALSE FALSE FALSE FALSE  FALSE           FALSE
```

```
unique(is.na(prey |> dplyr::select(all_of(comcol_prey_pred))))
```

```
     tblUploadID tblHaulID tblPredatorInformationID  Ship  Gear HaulNo
[1,]       FALSE     FALSE                    FALSE FALSE FALSE  FALSE
     StationNumber  Year Month   Day  Time FishID AphiaIDPredator
[1,]         FALSE FALSE FALSE FALSE FALSE  FALSE           FALSE
```

```
# Remove "Notes" from the prey data to avoid confusion as to which dataset it belongs.
d <- left_join(prey |> dplyr::select(-Notes),
               pred, by = comcol_prey_pred)
```

```
left_join: added 27 columns (IndWgt, Number, MeasurementIncrement, pred_length, Code, …)

           > rows only in x     249

           > rows only in y  (  706)

           > matched rows     7,418

           >                =======

           > rows total       7,667
```

```
# How many unique predators are there of the number of rows in the prey data that are not
length(unique(filter(prey, !tblPredatorInformationID %in% unique(pred$tblPredatorInformat
```

```
filter: removed 7,418 rows (97%), 249 rows remaining

[1] 220
```

```
# Total number of occuring predator ID's in both datasets
length(unique(c(pred$tblPredatorInformationID), unique(prey$tblPredatorInformationID)))
```

```
[1] 3295
```

Here we see that there are 249 rows that only exist in the prey data, and 284 that only exists in the predator data. The latter equals 284 predators, because 1 row = 1 predator. For the former this corresponds to 220 predators. This seems like a fairly substantial mismatch, given that the total amount of predator ID's is 3295. I.e., 8.6% of predators are not present in the prey data, and 7.5% of predators in the prey data are not present in the predator data.

Here is some more information about the predators missing in the prey data.

```
# Now the predators missing in the prey data
pred_not_in_prey <- pred |> filter(!tblPredatorInformationID %in% unique(prey$tblPredator
```

```
filter: removed 2,589 rows (79%), 706 rows remaining
```

```
unique(pred_not_in_prey$tblPredatorInformationID)
```

```
  [1] 13767 14036 14037 14041 14046 14050 14052 14060 14063 14082 14085 14086
 [13] 14090 14092 14093 14098 14101 14106 14110 14117 14118 14119 14120 14121
 [25] 14125 14132 14140 14142 14146 14166 14167 14175 14176 14183 14184 14193
 [37] 14198 14200 14207 14208 14211 14215 14216 14217 14220 14222 14226 14227
 [49] 14230 14232 14237 14243 14244 14249 14258 14259 14263 14264 14266 14267
 [61] 14268 14275 14276 14280 14281 14283 14286 14287 14288 14289 14290 14291
 [73] 14292 14298 14303 14305 14306 14308 14310 14311 14312 14314 14316 14317
 [85] 14324 14325 14326 14328 14329 14330 14332 14333 14334 14339 14340 14341
 [97] 14344 14354 14359 14366 14367 14372 14374 14378 14379 14385 14390 14391
[109] 14392 14398 14399 14401 14407 14409 14410 14423 14429 14432 14435 14440
[121] 14443 14458 14460 14465 14467 14473 14476 14478 14479 14482 14488 14499
[133] 14503 14510 14515 14517 14519 14525 14526 14535 14540 14541 14545 14548
[145] 14550 14553 14561 14565 14567 14568 14577 14580 14581 14583 14586 14605
[157] 14606 14615 14628 14630 14661 15894 15919 15924 15925 15927 15931 15938
[169] 15943 15951 15972 15973 16002 16004 16007 16013 16015 16016 16019 16096
[181] 16098 16103 16118 16125 16128 16129 16130 16131 16133 16134 16137 16144
[193] 16145 16147 16153 16159 16160 16164 16167 16168 16169 16170 16174 16179
[205] 16181 16182 16199 16200 16202 16206 16211 16212 16213 16214 16215 16229
[217] 16230 16234 16244 16246 16249 16252 16254 16257 16258 16259 16262 16263
[229] 16270 16275 16276 16283 16296 16300 16302 16304 16324 16331 16339 16362
[241] 16371 16378 16381 16395 16402 16428 16429 16438 16442 16443 16448 16451
[253] 16452 16454 16456 16461 16462 16466 16468 16470 16472 16496 16497 16505
[265] 16506 16507 16508 16509 16512 16515 16516 16517 16520 16521 16543 16547
[277] 16564 16582 16585 16589 16590 16603 16771 16779 28290 28293 28297 28298
[289] 28299 28300 28301 28303 28304 28305 28307 28314 28315 28318 28319 28320
[301] 28324 28333 28336 28340 28345 28347 28349 28356 28357 28360 28364 28365
[313] 28368 28376 28377 28378 28380 28383 28384 28385 28387 28389 28393 28395
[325] 28396 28399 28401 28407 28411 28412 28424 28426 28429 28430 28432 28433
[337] 28435 28437 28438 28444 28445 28448 28458 28459 28460 28461 28463 28465
[349] 28466 28473 28479 28483 28486 28489 28511 28520 28521 28532 28549 28550
[361] 28551 28556 28557 28558 28559 28560 28562 28565 28566 28574 28578 28579
[373] 28580 28583 28585 28586 28588 28589 28590 28594 28595 28596 28599 28600
[385] 28601 28603 28606 28607 28608 28609 28611 28612 28613 28615 28616 28617
[397] 28618 28622 28625 28628 28633 28634 28635 28636 28639 28640 28650 28651
```

```
[409] 28652 28654 28655 28657 28658 28659 28661 28663 28664 28667 28669 28671
[421] 28674 28675 28676 28677 28678 28679 28683 28684 28685 28687 28690 28691
[433] 28695 28697 28702 28706 28707 28708 28710 28735 28737 28738 28739 28742
[445] 28747 28748 28756 28759 28761 28764 28767 28769 28771 28772 28783 28787
[457] 28788 28798 28804 28813 28815 28818 28819 28820 28828 28840 28841 28845
[469] 28846 28866 28867 28869 28870 28884 28888 28891 28895 28897 28898 28899
[481] 28900 28902 28904 28907 28908 28911 28914 28926 28932 28945 28946 28952
[493] 28953 28954 28956 28960 28977 28978 28983 28984 28991 28993 28995 29007
[505] 29021 29023 29025 29026 29031 29035 29037 29043 29049 29052 29053 29054
[517] 29059 29060 29063 29064 29067 29069 29071 29072 29073 29074 29075 29077
[529] 29078 29080 29092 29094 29099 29105 29114 29118 29124 29129 29130 29131
[541] 29132 29134 29137 29138 29139 29141 29155 29169 29178 29191 29193 29194
[553] 29201 29202 29206 29208 29216 29224 29228 29231 29241 29242 29249 29251
[565] 29253 29259 29265 29268 29271 29272 29276 29284 29288 29293 29295 29300
[577] 29306 29309 29311 29314 29317 29319 29322 29326 29328 29331 29332 29336
[589] 29340 29341 29342 29344 29350 29354 29357 29362 29370 29372 29380 29381
[601] 29386 29388 29391 29392 29393 29402 29408 29409 29410 29416 29424 29425
[613] 29428 29432 29435 29438 29439 29442 29443 29446 29447 29449 29450 29451
[625] 29452 29455 29457 29470 29474 29475 29476 29486 29487 29488 29489 29490
[637] 29491 29492 29497 29502 29503 29506 29513 29517 29520 29532 29533 29535
[649] 29536 29539 29542 29544 29545 29547 29551 29552 29557 29562 29572 29573
[661] 29574 29576 29577 29581 29582 29584 29585 29587 29589 29590 29591 29592
[673] 29593 29595 29596 29598 29599 29600 29601 29603 29604 29605 29607 29608
[685] 29609 29610 29613 29616 29617 29619 29620 29623 29625 29627 29628 29629
[697] 29631 29632 29633 29639 29642 29645 29648 29651 29652 29654
```

```
unique(pred_not_in_prey$tblUploadID)
```

```
 [1] 8164 8167 8168 8169 8175 8177 8184 8185 8186 8187 8188
```

```
unique(pred_not_in_prey$Reporting_organisation)
```

```
[1] 3140  194 4484
```

```
# 3140  Institute of Food Safety, Animal Health and Environment (BIOR)  EDMO    False   2
# 194   National Marine Fisheries Research Institute (NMFRI)    EDMO    False   2015-02-0
# 4484  Swedish University of Agricultural Sciences, Department of Aquatic Resources (SLU
```

Here is some more information about the predators missing in the predator data but are in the prey data.

```
# Now the predators missing in the predator data
pred_not_in_pred <- prey |> filter(!tblPredatorInformationID %in% unique(pred$tblPredator
```

```
filter: removed 7,418 rows (97%), 249 rows remaining
```

```
unique(pred_not_in_pred$tblPredatorInformationID)
```

```
 [1] 13769 13770 13772 13773 13774 13775 13776 13777 13778 13779 13781 13782
[13] 13783 13784 13785 13786 13787 13788 13789 13790 13791 13792 13793 13794
[25] 13795 13796 13797 13798 13799 13800 13801 13802 13803 13804 13805 13806
[37] 13807 13808 13809 13810 13811 13812 13813 13814 13815 13816 13819 13820
[49] 13821 13824 13826 13829 13831 13832 13833 13837 13838 13840 13844 13845
[61] 13847 13848 13849 13851 13853 13854 13855 13856 13857 13858 13861 13863
[73] 13864 13867 13868 13869 13870 13871 13872 13873 13874 13875 13876 13877
[85] 13879 13880 13881 13882 13883 13885 13886 13887 13888 13889 13890 13891
[97] 13892 13893 13894 13895 13896 13897 13898 13899 13900 13901 13902 13903
[109] 13904 13906 13908 13909 13910 13911 13912 13913 13914 13915 13916 13917
[121] 13918 13919 13920 13921 13922 13923 13924 13925 13929 13931 13933 13934
[133] 13935 13936 13938 13939 13940 13941 13943 13944 13945 13947 13948 13949
[145] 13950 13953 13955 13956 13959 13960 13963 13964 13965 13967 13968 13969
[157] 13970 13971 13972 13973 13974 13976 13977 13979 13980 13981 13982 13984
[169] 13985 13987 13988 13989 13991 13993 13995 14000 14001 14004 14005 14006
[181] 14007 14009 14012 14014 14015 14018 14019 14020 14021 14022 14023 14024
[193] 14025 14026 14027 14028 14030 14031 14032 14033 14034 14043 14047 14048
[205] 14053 14054 14055 14056 14064 14065 14068 14071 14072 14073 14074 14075
[217] 14076 14077 14078 14079
```

```
unique(pred_not_in_pred$tblUploadID)
```

```
[1] 8164 8165 8166 8167
```

```
unique(pred_not_in_pred$AnalysingOrg)
```

```
[1] 3140
```

```
# 3140  Institute of Food Safety, Animal Health and Environment (BIOR)  EDMO    False   2

# Can't print country or anything like that because it comes from the predator data, so I
unique(pred_not_in_pred$tblPredatorInformationID)
```

```
 [1] 13769 13770 13772 13773 13774 13775 13776 13777 13778 13779 13781 13782
[13] 13783 13784 13785 13786 13787 13788 13789 13790 13791 13792 13793 13794
[25] 13795 13796 13797 13798 13799 13800 13801 13802 13803 13804 13805 13806
[37] 13807 13808 13809 13810 13811 13812 13813 13814 13815 13816 13819 13820
[49] 13821 13824 13826 13829 13831 13832 13833 13837 13838 13840 13844 13845
[61] 13847 13848 13849 13851 13853 13854 13855 13856 13857 13858 13861 13863
[73] 13864 13867 13868 13869 13870 13871 13872 13873 13874 13875 13876 13877
[85] 13879 13880 13881 13882 13883 13885 13886 13887 13888 13889 13890 13891
[97] 13892 13893 13894 13895 13896 13897 13898 13899 13900 13901 13902 13903
[109] 13904 13906 13908 13909 13910 13911 13912 13913 13914 13915 13916 13917
[121] 13918 13919 13920 13921 13922 13923 13924 13925 13929 13931 13933 13934
[133] 13935 13936 13938 13939 13940 13941 13943 13944 13945 13947 13948 13949
[145] 13950 13953 13955 13956 13959 13960 13963 13964 13965 13967 13968 13969
[157] 13970 13971 13972 13973 13974 13976 13977 13979 13980 13981 13982 13984
[169] 13985 13987 13988 13989 13991 13993 13995 14000 14001 14004 14005 14006
[181] 14007 14009 14012 14014 14015 14018 14019 14020 14021 14022 14023 14024
```

```
[193] 14025 14026 14027 14028 14030 14031 14032 14033 14034 14043 14047 14048
[205] 14053 14054 14055 14056 14064 14065 14068 14071 14072 14073 14074 14075
[217] 14076 14077 14078 14079
```

Inspect data

Check to see how many predators I have by country and year, to see how much is missing. Because we can't match them perfectly, I will drop rows that couldn't be matched in the predator data, using the predator-specific `pred_length` column

```
# By year and country from years => 1993
d |>
  drop_na(pred_length) |>
  distinct(tblPredatorInformationID, .keep_all = TRUE) |>
  group_by(Country, Year) |>
  summarise(n = n()) |>
  arrange(Year, Country)
```

```
drop_na: removed 249 rows (3%), 7,418 rows remaining

distinct: removed 4,829 rows (65%), 2,589 rows remaining

group_by: 2 grouping variables (Country, Year)

summarise: now 10 rows and 3 columns, one group variable remaining (Country)

# A tibble: 10 × 3
# Groups:   Country [3]
   Country  Year     n
   <chr>   <int> <int>
 1 LV       1993   213
 2 LV       1994   143
 3 LV       1995   232
 4 LV       1996   222
 5 LV       1997   135
 6 LV       2001   115
 7 LV       2002   158
 8 LV       2003   168
 9 PL       2021   582
10 SE       2021   621
```

```
# By year and country from years => 2017
d |>
  drop_na(pred_length) |>
  distinct(tblPredatorInformationID, .keep_all = TRUE) |>
  filter(Year > 2017) |>
  group_by(Country, Year) |>
```

```
  summarise(n = n()) |>
  arrange(Year, Country)
```

```
drop_na: removed 249 rows (3%), 7,418 rows remaining

distinct: removed 4,829 rows (65%), 2,589 rows remaining

filter: removed 1,386 rows (54%), 1,203 rows remaining

group_by: 2 grouping variables (Country, Year)

summarise: now 2 rows and 3 columns, one group variable remaining (Country)

# A tibble: 2 × 3
# Groups:   Country [2]
  Country  Year     n
  <chr>   <int> <int>
1 PL       2021   582
2 SE       2021   621
```

```
# Total sample size from years => 1993
d |>
  drop_na(pred_length) |>
  distinct(tblPredatorInformationID, .keep_all = TRUE) |>
  summarise(n = n())
```

```
drop_na: removed 249 rows (3%), 7,418 rows remaining

distinct: removed 4,829 rows (65%), 2,589 rows remaining

summarise: now one row and one column, ungrouped

     n
1 2589
```

```
# Total sample size from years => 2017
d |>
  drop_na(pred_length) |>
  distinct(tblPredatorInformationID, .keep_all = TRUE) |>
  filter(Year > 2017) |>
  summarise(n = n())
```

```
drop_na: removed 249 rows (3%), 7,418 rows remaining

distinct: removed 4,829 rows (65%), 2,589 rows remaining

filter: removed 1,386 rows (54%), 1,203 rows remaining

summarise: now one row and one column, ungrouped
```

```
     n
1 1203
```

We are supposed to have 6128 samples from 2017 and onwards, according to Table A3 in the interim report. We have 1203, which is only 20% of the promised new data. Moreover, we have only and 442 individuals from the older times (earlier than 2017). I do not recall exactly how many we have from 1993-2016, but I suspect there should be thousands based on memory.

Now let's have a look at stomach contents.

Calculating total weight of specific prey species by unique predator ID

Because of the mismatch in predator ID's, we will filter the combined dataframe to only have the IDs that are present in both dataframes

```
d <- d |> drop_na(pred_length) # this removes predators not in the predator data
```

```
drop_na: removed 249 rows (3%), 7,418 rows remaining
```

```
pred <- pred |> filter(tblPredatorInformationID %in% unique(prey$tblPredatorInformationID
```

```
filter: removed 706 rows (21%), 2,589 rows remaining
```

Next we need to summarize our prey weights by predator and prey group. First filter stomachs where these prey are present. Create a new common name column to make life easier...

```
selected_prey_present <- d |>
  mutate(common_prey_name = NA,
         common_prey_name = ifelse(AphiaIDPrey == 126425, "sprat", common_prey_name),
         common_prey_name = ifelse(AphiaIDPrey == 126417, "herring", common_prey_name),
         common_prey_name = ifelse(AphiaIDPrey == 119034, "saduria", common_prey_name)) |
  filter(common_prey_name %in% c("sprat", "herring", "saduria"))
```

Check how many predators have these prey in stomachs

```
length(unique(selected_prey_present$tblPredatorInformationID))
```

```
[1] 1178
```

```
length(unique(d$tblPredatorInformationID))
```

```
[1] 2589
```

```
length(unique(selected_prey_present$tblPredatorInformationID)) / length(unique(d$tblPreda
```

```
[1] 0.4550019
```

Only 45% of cod have *any* of these prey species? That sounds fairly low, given that these are the most common prey and likely make up >80% of a cod's diet around 30 cm. Update! This percentage is higher now that I have old data as well. In the newer data this number is around 23%.

Next we want to group be predator and prey, and summarize the total weight. Before though, we need to make sure there are no `NA`s in `Weight`

```
unique(selected_prey_present$Weight)
```

```
 [1]   0   1   3   5   2   6  12  10   4   8   7  25  11  20   9  27  15  34  18
[20]  21  26  17  14  13  23  22  42  30  50  31  47  16  37  55  33  NA  24  19
[39]  58  28  41  80  32  29  66  89  36 122  88  67 165  72  43  35  40  76  48
[58]  57  52  83  38  44  79 327 190  51  69  54 189 142 102  49
```

```
selected_prey_present <- selected_prey_present |> drop_na(Weight)
```

```
drop_na: removed 18 rows (1%), 2,228 rows remaining
```

Only 18 row with `Weight == NA`. We can remove that. But there are also weights that are 0. How can a prey weigh nothing? If it can't be measured I would have assumed it to be NA. Here are the analyzing organisations and the predator ID's.

```
zero_weights <- selected_prey_present |> filter(Weight == 0)
```

```
filter: removed 1,928 rows (87%), 300 rows remaining
```

```
unique(zero_weights$AnalysingOrg)
```

```
[1] 3140  194
```

```
unique(zero_weights$tblPredatorInformationID)
```

```
  [1] 14114 14152 14158 14159 14161 14174 14177 14178 14185 14186 14189 14196
 [13] 14197 14214 14225 14634 14649 14654 14655 14656 14657 14662 15908 15979
 [25] 15966 15911 15926 15932 15977 15989 15982 16012 16036 16037 16035 16059
 [37] 16061 16079 16092 16078 16142 16261 16319 16361 16363 16369 16482 16523
 [49] 16529 16554 16558 16601 16604 16605 16614 16616 16650 16682 16731 16798
 [61] 16872 16874 16875 16876 16894 16900 16904 16910 16911 16916 16923 16980
 [73] 16985 16989 17003 17028 17029 17037 17039 17081 17085 17095 17099 17117
 [85] 17127 17183 17204 17207 17217 17221 28306 28362 28374 28382 28366 28372
 [97] 28414 28449 28510 28518 28531 28541 28544 28547 28561 28563 28569 28680
[109] 28610 28619 28620 28623 28626 28602 28638 28649 28653 28656 28665 28668
[121] 28670 28688 28689 28693 28696 28704 28713 28725 28751 28755 28757 28762
[133] 28774 28792 28795 28803 28805 28806 28809 28811 28814 28779 28782 28825
[145] 28835 28847 28864 28878 28882 28886 28894 28950 28963 28973 28979 28940
[157] 28992 29008 29011 29001 29041 29044 29045 29051 29055 29098 29106 29110
```

```
[169] 29127 29135 29136 29145 29154 29157 29146 29148 29150 29153 29161 29173
[181] 29174 29177 29179 29166 29167 29168 29190 29181 29185 29186 29204 29207
[193] 29209 29210 29212 29214 29215 29218 29195 29220 29229 29230 29233 29200
[205] 29248 29256 29239 29257 29260 29263 29267 29269 29273 29274 29281 29286
[217] 29244 29245 29297 29298 29299 29302 29305 29307 29321 29291 29337 29345
[229] 29294 29353 29371 29373 29375 29394 29389 29390 29399 29400 29401 29418
[241] 29420 29421 29411 29414 29426 29437 29427 29453 29433 29468 29456 29477
[253] 29460 29518 29519 29500 29504 29526 29555 29559 29586 29588 29579 29612
[265] 29636
```

```
# 3140  Institute of Food Safety, Animal Health and Environment (BIOR)  EDMO   False  2
# 194   National Marine Fisheries Research Institute (NMFRI)            EDMO   False  2015-02-0
```

13% of rows where these prey are present do not have a weight.... Do they have a length? Can we estimate Weight? If length is not `NA` and Weight is 0, estimate weight based on length and multiply with the count of prey. Else give weight `NA` and drop it.

```
t <- selected_prey_present |>
  mutate(weight_source = ifelse(Weight == 0 & !is.na(prey_length) & !is.na(Count), "estim
  mutate(Weight = ifelse(weight_source == "estimated", (0.01*prey_length^3)*Count, Weight
  filter(Weight > 0)
```

```
mutate: new variable 'weight_source' (character) with 2 unique values and 0% NA

mutate: converted 'Weight' from integer to double (0 new NA)

filter: removed 230 rows (10%), 1,998 rows remaining
```

Even if we estimate the prey weight based on the length of the prey and the number of the prey, we still have 10% of rows in this data with saduria, herring and sprat that does not have any weight information, even though they were clearly present in the data. We can see that here:

```
selected_prey_present |>
  filter(Weight == 0) |>
  head(5)
```

```
filter: removed 1,928 rows (87%), 300 rows remaining

  tblUploadID tblHaulID tblPredatorInformationID tblPreyInformationID Ship Gear
1        8167      2414                    14114                17420 LA99  LBT
2        8167      2418                    14152                17465 LA99  LBT
3        8167      2419                    14158                17473 LA99  LBT
4        8167      2419                    14159                17474 LA99  LBT
5        8167      2419                    14161                17480 LA99  LBT
  HaulNo StationNumber Year Month Day Time FishID AphiaIDPredator AphiaIDPrey
1      2             2 2001    11  25  700     17          126436      119034
2      6             2 2001    11  25 1010      7          126436      119034
3      7             2 2001    11  25 1055      6          126436      119034
4      7             2 2001    11  25 1055      7          126436      119034
```

```
5      7             2 2001    11  25 1055      9          126436      119034
  IdentMet DigestionStage GravMethod SubFactor PreySequence Count UnitWgt
1     VISO            NA      WETWT        NA           1     3       g
2     VISO            NA      WETWT        NA           1     2       g
3     VISO            NA      WETWT        NA           3    NA       g
4     VISO            NA      WETWT        NA           1     1       g
5     VISO            NA      WETWT        NA           2    NA       g
  Weight UnitLngt prey_length OtherItems OtherCount OtherWgt AnalysingOrg
1      0                  NA                    NA       NA         3140
2      0                  NA                    NA       NA         3140
3      0                  NA                    NA       NA         3140
4      0                  NA                    NA       NA         3140
5      0                  NA                    NA       NA         3140
  IndWgt Number MeasurementIncrement pred_length Code Age Sex MaturityScale
1   21.6      1                   1          13 <NA>  NA   U
2   25.5      1                   1          14 <NA>  NA   U
3   18.4      1                   1          13 <NA>  NA   U
4   24.4      1                   1          14 <NA>  NA   U
5   25.1      1                   1          14 <NA>  NA   F            M6
  MaturityStage PreservationMethod Regurgitated StomachFullness FullStomWgt
1            NA                NBF            0              NA          NA
2            NA                NBF            0              NA          NA
3            NA                NBF            0              NA          NA
4            NA                NBF            0              NA          NA
5            65                NBF            0              NA          NA
  EmptyStomWgt StomachEmpty GenSamp ShootLat ShootLong HaulLat HaulLong
1           NA            0       N  56.6167   20.6499      NA       NA
2           NA            0       N  56.6833   20.7500      NA       NA
3           NA            0       N  56.6499   20.6832      NA       NA
4           NA            0       N  56.6499   20.6832      NA       NA
5           NA            0       N  56.6499   20.6832      NA       NA
  ICESrectangle Depth Survey ICESDatabase Country Reporting_organisation
1          42H0    45                  Y      LV                   3140
2          42H0    47                  Y      LV                   3140
3          42H0    51                  Y      LV                   3140
4          42H0    51                  Y      LV                   3140
5          42H0    51                  Y      LV                   3140
       CruiseID common_prey_name
1 LV3140LA992001          saduria
2 LV3140LA992001          saduria
3 LV3140LA992001          saduria
4 LV3140LA992001          saduria
5 LV3140LA992001          saduria
```

I ***guess*** this is because they couldn't be measured or weighed, only counted. Which seems fair. For our purpose, it would probably be less wrong to give them the average weight rather than treating them as 0s, so that's what I'll do here.

```
prey_avg_ind_weight <- selected_prey_present |>
  filter(!is.na(Count) & Weight > 0) |>
  group_by(common_prey_name) |>
```

```
  mutate(ind_weight = Weight / Count) |>
  summarise(avg_weight = mean(ind_weight))
```

```
filter: removed 420 rows (19%), 1,808 rows remaining

group_by: one grouping variable (common_prey_name)

mutate (grouped): new variable 'ind_weight' (double) with 107 unique values and 0% NA

summarise: now 3 rows and 2 columns, ungrouped
```

```
prey_avg_ind_weight
```

```
# A tibble: 3 × 2
  common_prey_name avg_weight
  <chr>                 <dbl>
1 herring               32.4
2 saduria                3.65
3 sprat                  5.44
```

This should be the average prey weight which we can use to calculate the weight of these prey if we have the counts. Left join to data.

```
selected_prey_present_clean <- selected_prey_present |>
  left_join(prey_avg_ind_weight) |>
  mutate(weight_source = ifelse(Weight == 0 & !is.na(prey_length) & !is.na(Count), "estim
  mutate(Weight = ifelse(weight_source == "estimated", (0.01*prey_length^3)*Count, Weight
  group_by(common_prey_name) |>
  mutate(Weight = ifelse(Weight == 0 & !is.na(Count), Count * avg_weight, Weight)) |>
  filter(Weight > 0)
```

```
Joining with `by = join_by(common_prey_name)`
left_join: added one column (avg_weight)
> rows only in x 0
> rows only in y ( 0)
> matched rows 2,228
> =======
> rows total 2,228
mutate: new variable 'weight_source' (character) with 2 unique values and 0% NA
mutate: converted 'Weight' from integer to double (0 new NA)
group_by: one grouping variable (common_prey_name)
mutate (grouped): changed 178 values (8%) of 'Weight' (0 new NA)
filter (grouped): removed 52 rows (2%), 2,176 rows remaining
```

With these estimates of weight based on either length or worst case, average weight of that prey, we only need to drop 52 rows in the presence data (around 2%). Now calculate the total weight of these prey per individual predator stomach, and then pivot wider.

```
selected_prey_present_summed <- selected_prey_present_clean |>
  group_by(tblPredatorInformationID, common_prey_name) |>
```

```
  summarise(tot_weight = sum(Weight))
```

```
group_by: 2 grouping variables (tblPredatorInformationID, common_prey_name)

summarise: now 1,292 rows and 3 columns, one group variable remaining
(tblPredatorInformationID)
```

```
selected_prey_present_summed <- selected_prey_present_summed |>
  pivot_wider(names_from = "common_prey_name", values_from = "tot_weight", values_fill =
```

```
pivot_wider: reorganized (common_prey_name, tot_weight) into (saduria, sprat, herring)
[was 1292x3, now 1135x4]
```

Here I need to set `values_fill = 0`. Because they are not `NA` but 0. This is just because I work with three species at the same time. Next I will `left_join` in the remaining predator information, and after that `bind_rows` "empty stomachs" (with respect to these 3 prey species). Since the IDs are not overlapping, it doesn't matter that I already have some 0's here for some species

```
selected_prey_present_summed <- selected_prey_present_summed |>
  left_join(pred, by = "tblPredatorInformationID")
```

```
left_join: added 39 columns (tblUploadID, tblHaulID, Ship, Gear, HaulNo, …)

           > rows only in x       0

           > rows only in y  (1,454)

           > matched rows     1,135

           >                 =======

           > rows total       1,135
```

Now add in the "empty stomachs" using bind_rows. When I `bind_rows`, the columns that are not matching get `NA`. The only column not matching should be the average weight columns. They will get `NA`, and I'll change it to 0.

```
empty <- pred |> filter(!tblPredatorInformationID %in% c(selected_prey_present_summed$tbl
```

```
filter: removed 1,135 rows (44%), 1,454 rows remaining
```

```
dd <- bind_rows(selected_prey_present_summed, empty)

# Yes, works as intended, see the added NAs in the selected prey weights. Make them 0!
unique(is.na(dd))
```

```
     tblPredatorInformationID saduria sprat herring tblUploadID tblHaulID  Ship
[1,]                    FALSE   FALSE FALSE   FALSE       FALSE     FALSE FALSE
[2,]                    FALSE   FALSE FALSE   FALSE       FALSE     FALSE FALSE
```

```
[3,]                   FALSE   FALSE FALSE   FALSE       FALSE     FALSE FALSE
[4,]                   FALSE   FALSE FALSE   FALSE       FALSE     FALSE FALSE
[5,]                   FALSE    TRUE  TRUE    TRUE       FALSE     FALSE FALSE
[6,]                   FALSE    TRUE  TRUE    TRUE       FALSE     FALSE FALSE
[7,]                   FALSE    TRUE  TRUE    TRUE       FALSE     FALSE FALSE
[8,]                   FALSE    TRUE  TRUE    TRUE       FALSE     FALSE FALSE
      Gear HaulNo StationNumber  Year Month   Day  Time FishID AphiaIDPredator
[1,] FALSE  FALSE         FALSE FALSE FALSE FALSE FALSE  FALSE           FALSE
[2,] FALSE  FALSE         FALSE FALSE FALSE FALSE FALSE  FALSE           FALSE
[3,] FALSE  FALSE         FALSE FALSE FALSE FALSE FALSE  FALSE           FALSE
[4,] FALSE  FALSE         FALSE FALSE FALSE FALSE FALSE  FALSE           FALSE
[5,] FALSE  FALSE         FALSE FALSE FALSE FALSE FALSE  FALSE           FALSE
[6,] FALSE  FALSE         FALSE FALSE FALSE FALSE FALSE  FALSE           FALSE
[7,] FALSE  FALSE         FALSE FALSE FALSE FALSE FALSE  FALSE           FALSE
[8,] FALSE  FALSE         FALSE FALSE FALSE FALSE FALSE  FALSE           FALSE
     IndWgt Number MeasurementIncrement pred_length  Code   Age   Sex
[1,]  FALSE  FALSE                FALSE       FALSE  TRUE  TRUE FALSE
[2,]  FALSE  FALSE                FALSE       FALSE  TRUE  TRUE FALSE
[3,]  FALSE  FALSE                FALSE       FALSE FALSE FALSE FALSE
[4,]  FALSE  FALSE                FALSE       FALSE FALSE FALSE FALSE
[5,]  FALSE  FALSE                FALSE       FALSE  TRUE  TRUE FALSE
[6,]  FALSE  FALSE                FALSE       FALSE  TRUE  TRUE FALSE
[7,]  FALSE  FALSE                FALSE       FALSE FALSE FALSE FALSE
[8,]  FALSE  FALSE                FALSE       FALSE FALSE FALSE FALSE
     MaturityScale MaturityStage PreservationMethod Regurgitated
[1,]         FALSE          TRUE              FALSE        FALSE
[2,]         FALSE         FALSE              FALSE        FALSE
[3,]         FALSE         FALSE              FALSE         TRUE
[4,]         FALSE         FALSE              FALSE        FALSE
[5,]         FALSE         FALSE              FALSE        FALSE
[6,]         FALSE          TRUE              FALSE        FALSE
[7,]         FALSE         FALSE              FALSE         TRUE
[8,]         FALSE         FALSE              FALSE        FALSE
     StomachFullness FullStomWgt EmptyStomWgt StomachEmpty GenSamp ShootLat
[1,]            TRUE        TRUE         TRUE        FALSE   FALSE    FALSE
[2,]            TRUE        TRUE         TRUE        FALSE   FALSE    FALSE
[3,]            TRUE        TRUE         TRUE         TRUE   FALSE    FALSE
[4,]            TRUE        TRUE         TRUE         TRUE   FALSE    FALSE
[5,]            TRUE        TRUE         TRUE        FALSE   FALSE    FALSE
[6,]            TRUE        TRUE         TRUE        FALSE   FALSE    FALSE
[7,]            TRUE        TRUE         TRUE         TRUE   FALSE    FALSE
[8,]            TRUE        TRUE         TRUE         TRUE   FALSE    FALSE
     ShootLong HaulLat HaulLong ICESrectangle Depth Survey ICESDatabase Country
[1,]     FALSE    TRUE     TRUE         FALSE FALSE  FALSE        FALSE   FALSE
[2,]     FALSE    TRUE     TRUE         FALSE FALSE  FALSE        FALSE   FALSE
[3,]     FALSE   FALSE    FALSE         FALSE FALSE  FALSE        FALSE   FALSE
[4,]     FALSE   FALSE    FALSE         FALSE FALSE  FALSE        FALSE   FALSE
[5,]     FALSE    TRUE     TRUE         FALSE FALSE  FALSE        FALSE   FALSE
[6,]     FALSE    TRUE     TRUE         FALSE FALSE  FALSE        FALSE   FALSE
[7,]     FALSE   FALSE    FALSE         FALSE FALSE  FALSE        FALSE   FALSE
[8,]     FALSE   FALSE    FALSE         FALSE FALSE  FALSE        FALSE   FALSE
```

```
     Reporting_organisation CruiseID
[1,]                  FALSE    FALSE
[2,]                  FALSE    FALSE
[3,]                  FALSE    FALSE
[4,]                  FALSE    FALSE
[5,]                  FALSE    FALSE
[6,]                  FALSE    FALSE
[7,]                  FALSE    FALSE
[8,]                  FALSE    FALSE
```

```
dd <- dd |>
  mutate(saduria = ifelse(is.na(saduria), 0, saduria),
         herring = ifelse(is.na(herring), 0, herring),
         sprat   = ifelse(is.na(sprat), 0, sprat))
```

```
mutate (grouped): changed 1,454 values (56%) of 'saduria' (1454 fewer NA)

                  changed 1,454 values (56%) of 'sprat' (1454 fewer NA)

                  changed 1,454 values (56%) of 'herring' (1454 fewer NA)
```

```
# Now trim the dataset a bit...

dd <- dd |>
  dplyr::select(tblPredatorInformationID, tblHaulID, Year, Month, Day, Time, AphiaIDPreda
                IndWgt, pred_length, Age, Sex, saduria, sprat, herring, ShootLat,
                ShootLong, ICESrectangle, Depth, Survey) |>
  rename(lon = ShootLong,
         lat = ShootLat) |>
  janitor::clean_names()
```

```
rename: renamed 2 variables (lat, lon)
```

```
glimpse(dd)
```

```
Rows: 2,589
Columns: 19
Groups: tbl_predator_information_id [2,589]
$ tbl_predator_information_id <int> 14114, 14152, 14159, 14174, 14177, 14178, …
$ tbl_haul_id                 <int> 2414, 2418, 2419, 2420, 2421, 2421, 2421, …
$ year                        <int> 2001, 2001, 2001, 2001, 2001, 2001, 2001, …
$ month                       <int> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11…
$ day                         <int> 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25…
$ time                        <int> 700, 1010, 1055, 1140, 1225, 1225, 1225, 1…
$ aphia_id_predator           <int> 126436, 126436, 126436, 126436, 126436, 12…
$ ind_wgt                     <dbl> 21.6, 25.5, 24.4, 31.5, 35.5, 30.8, 25.1, …
$ pred_length                 <int> 13, 14, 14, 15, 15, 14, 14, 14, 14, 13, 14…
$ age                         <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
$ sex                         <chr> "U", "U", "U", "M", "F", "M", "F", "M", "M…
```

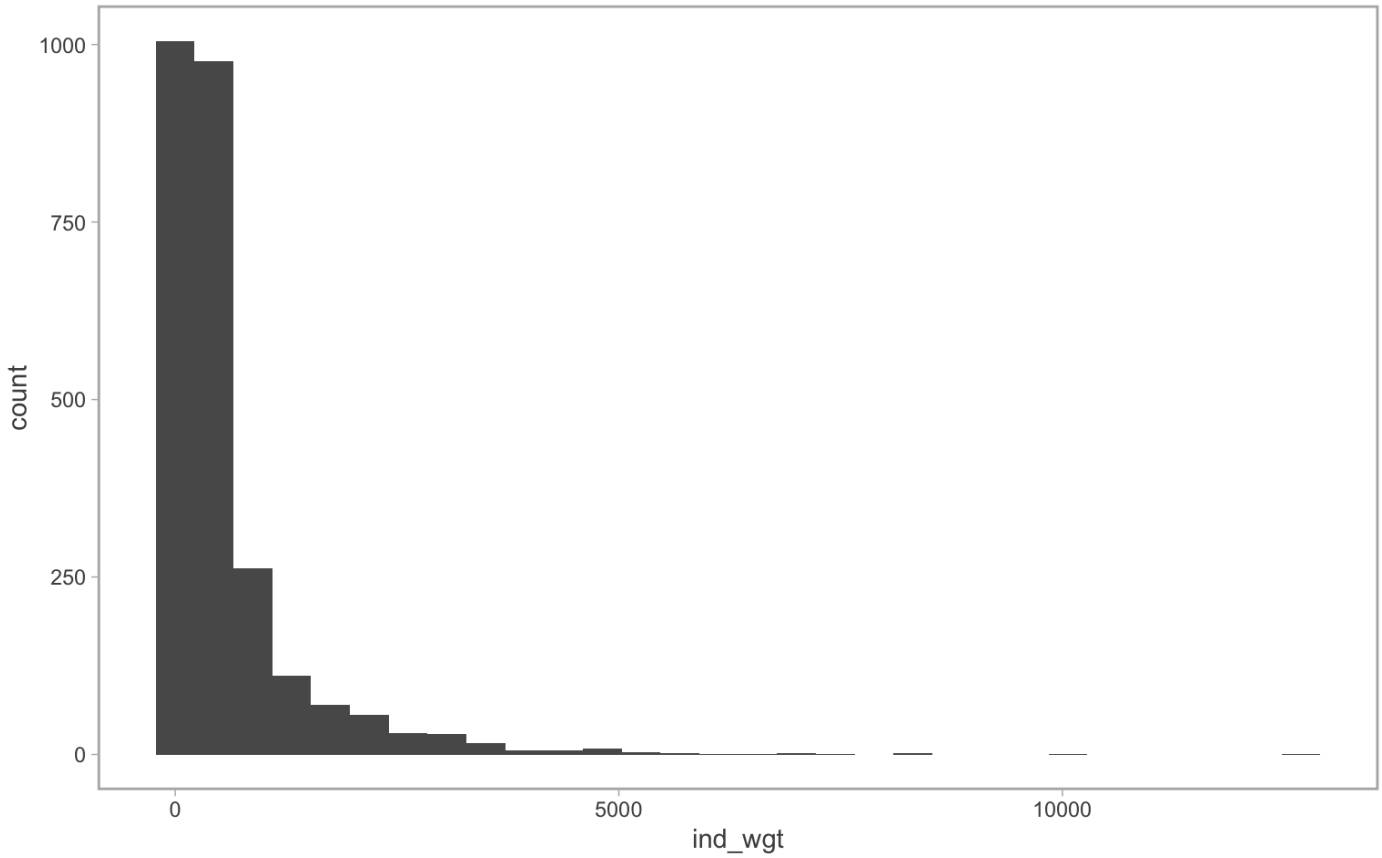
```
$ saduria                    <dbl> 10.957875, 7.305250, 3.652625, 10.957875, …
$ sprat                      <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.…
$ herring                    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
$ lat                        <dbl> 56.6167, 56.6833, 56.6499, 56.6499, 56.633…
$ lon                        <dbl> 20.6499, 20.7500, 20.6832, 20.6332, 20.616…
$ ice_srectangle             <chr> "42H0", "42H0", "42H0", "42H0", "42H0", "4…
$ depth                      <int> 45, 47, 51, 59, 60, 60, 60, 60, 60, 60, 51…
$ survey                     <chr> "", "", "", "", "", "", "", "", "", "", ""…
```

```r
# Ok, now summarise and plot these data. First calculate the feeding ratio, which is the
unique(is.na(dd$ind_wgt))
```

```
[1] FALSE
```

```r
ggplot(dd, aes(ind_wgt)) +
  geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
# Good, no NAs and all positive weight values. Calculate feeding ratios
dd <- dd |>
  mutate(FR_sad = saduria / ind_wgt,
         FR_spr = sprat / ind_wgt,
         FR_her = herring / ind_wgt)
```

```
mutate (grouped): new variable 'FR_sad' (double) with 264 unique values and 0% NA
                  new variable 'FR_spr' (double) with 602 unique values and 0% NA
                  new variable 'FR_her' (double) with 270 unique values and 0% NA
```

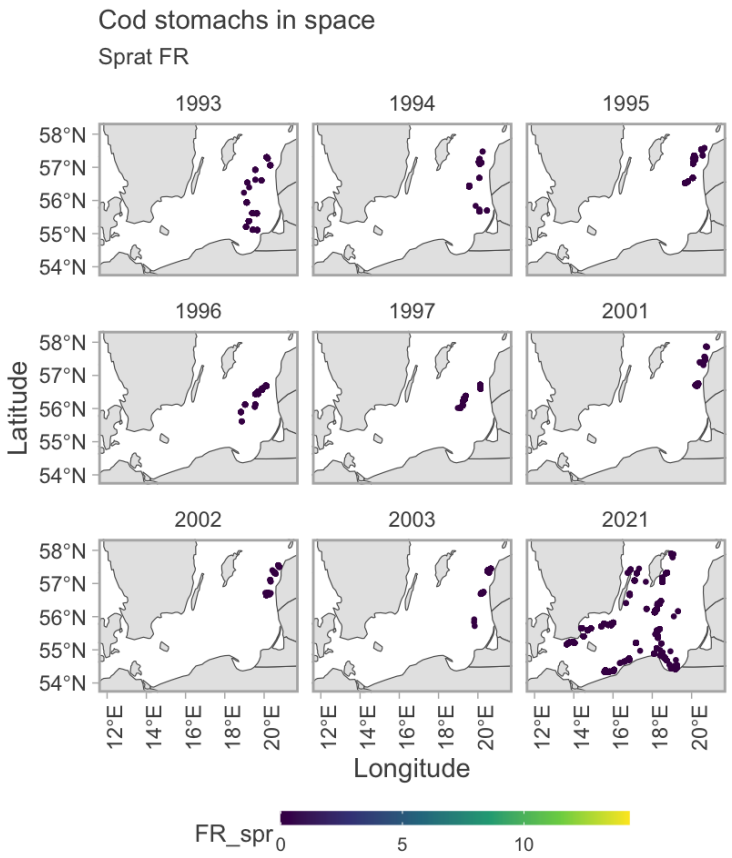
Check the proportion of stomachs without these prey. TODO: too late to figure out how to do this but overall it seems low...

```
dd %>%
  pivot_longer(c("herring", "saduria", "sprat")) |>
  group_by(name) |>
  summarise(prop_empty = sum(value == 0)/n(),
            prop_not_empty = sum(value != 0)/n())
```

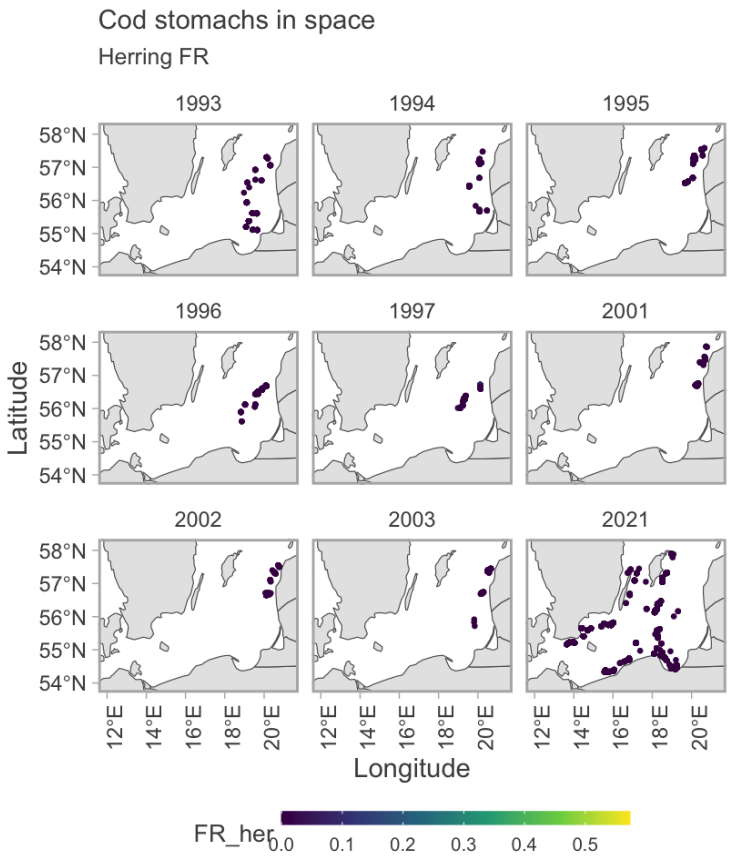
```
pivot_longer: reorganized (saduria, sprat, herring) into (name, value) [was 2589x22, now
7767x21]

group_by: one grouping variable (name)

summarise: now 3 rows and 3 columns, ungrouped

# A tibble: 3 × 3
  name    prop_empty prop_not_empty
  <chr>        <dbl>          <dbl>
1 herring      0.893          0.107
2 saduria      0.886          0.114
3 sprat        0.722          0.278
```

Seems like fairly high proportions of cod without these key species in stomachs, but again, this is just from memory.

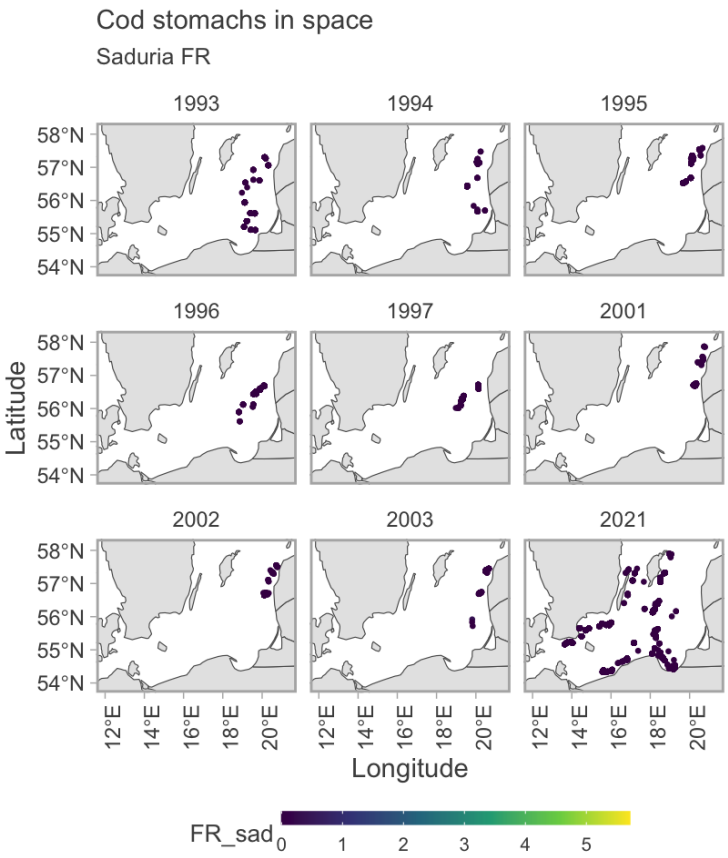
```
# Inspect spatiotemporal resolution of feeding ratio
dd <- dd |> add_utm_columns(ll_names = c("lon", "lat"), utm_crs = 32633)

# This contains stuff from the loaded map plot script, I have a thing there for continues
ggplot(swe_coast_proj) +
  xlim(xmin2, xmax2) +
  ylim(ymin2, ymax2) +
  labs(x = "Longitude", y = "Latitude") +
  geom_sf(size = 0.3) +
  theme_facet_map() +
  geom_point(data = dd, aes(X*1000, Y*1000, color = FR_spr), size = 0.5) +
  facet_wrap(~ year) +
  theme(legend.position = "bottom") +
  labs(title = "Cod stomachs in space", subtitle = "Sprat FR")
```




```
ggplot(swe_coast_proj) +
  xlim(xmin2, xmax2) +
  ylim(ymin2, ymax2) +
  labs(x = "Longitude", y = "Latitude") +
  geom_sf(size = 0.3) +
  theme_facet_map() +
  geom_point(data = dd, aes(X*1000, Y*1000, color = FR_her), size = 0.5) +
  facet_wrap(~ year) +
  theme(legend.position = "bottom") +
  labs(title = "Cod stomachs in space", subtitle = "Herring FR")
```




```
ggplot(swe_coast_proj) +
  xlim(xmin2, xmax2) +
  ylim(ymin2, ymax2) +
  labs(x = "Longitude", y = "Latitude") +
  geom_sf(size = 0.3) +
  theme_facet_map() +
  geom_point(data = dd, aes(X*1000, Y*1000, color = FR_sad), size = 0.5) +
  facet_wrap(~ year) +
  theme(legend.position = "bottom") +
  labs(title = "Cod stomachs in space", subtitle = "Saduria FR")
```



In summary:

1) When are the rest of the data uploaded?

2) What do we do with the predator ID's that are not matched across predator and prey files (both ways?).

3) How should we interpret weight 0? Treat it as NA weight?.

4) Do the proportions of cod stomachs without sprat, saduria or herring look reasonable?.

```
# Save new data

write_csv(dd, paste0(home, "/data/clean/new_stomachs.csv"))
```