Recreational cod catch estimation R-Script

# Introduction

This is the main R-script to compute the German recreational cod removals (harvest + dead releases) in the Baltic Sea as input data for the western Baltic cod stock assessment. The script utilizes data from two data sources: (1) effort data from an off-site population-wide telephone-diary survey and (2) catch data from an annual on-site intercept survey.

On-site data are used to calculate CPUE values of the anglers for different strata (federal\_state, SD, year, half\_year, survey\_category). Off-site data are used to calculate fishing effort for different strata (federal\_state, half\_year, survey\_category). Catch in numbers (CANUM) is calculated by multiplying the CPUE values with the fishing effort(total fishing days). CANUM is then divided into individual length classes based on the recreational length distribution from onboard length samples. With a length-weight relationship from commercial catch sampling CANUM is transferred to catch in tons (CATON) and then transferred to the final CATON (catch in tons at length and age).

At the end of the script is an overview sections of the results. Here you can see the results in different resolutions.

#data work  
library("readr")  
#library("plyr")  
library("dplyr")  
library("tidyr")  
library("reshape2")  
#library("Rmisc")  
  
#plots  
library("ggplot2")  
library("ggrepel")  
#library("cowplot")  
#library("ggpubr")

# Importing data

Before the national recreational catch data can be imported, a copy is made of the "\_original" data frames. This data frames are generated in the “get\_the\_data” script and include all necessary data, they should be saved in the working environment.

dat <- onsite\_ohneNullbepro\_original  
  
dat\_length <- dat\_length\_original  
   
fishing\_days <- fishing\_days\_original  
  
mass\_coefficient <- mass\_coefficient\_own\_original  
  
SnglIOR <- SnglIOR\_original  
  
all\_strata <- all\_strata\_original

# CPUE

In a first step CPUE values are calculated for the different strata for retained and released cod from the on-site data (dat data frame).

In the dat data frame every angler is one row (Unit effort = on fishing day). The mean CPUE is calculated for all anglers in a stratum summing the number of retained and released cod and dividing by the total numbers of anglers respectively.

dat <- na.omit(dat) %>%  
 group\_by(federal\_state, SD, year, half\_year, survey\_category)%>%  
 mutate(n.angsamples = length(ID),  
 n.ret = sum(retained),  
 n.rel = sum(released),  
 mean.CPUE = mean(retained),  
 mean.CPUE.rel = mean(released))

Here an overview table of the CPUE´s per strata is created.

dat.strata <- dat %>%  
 group\_by(federal\_state, SD, year, half\_year, survey\_category, n.angsamples, n.ret, n.rel,  
 mean.CPUE, mean.CPUE.rel) %>%  
 summarise()

# Catch in numbers

The next step is to calculate catch in numbers. Therefore, CPUE values are multiplied with the effort (fishing days) per strata.

### Ratio between SD22 and SD24

In contrast to the CPUEs, effort is not available per subdivision but for federal states. In one federal state (Schleswig-Holstein, SH) subdivision and federal state overlap in another state (Mecklenburg Western Pomerania. MV) the coastline spans two subdivisions and therefore effort needs to be recalculated. Therefore, effort is split between the MV strata in SD 22 and 24. Here we calculate the ratio between encountered anglers during the on-site survey in SD22 to SD24. In SH the ratio is always 1 (only SD 22). In MV the ratio for the two subdivisions is used to split the effort.

dat <- dat %>%  
 group\_by(federal\_state, year, half\_year, survey\_category)%>%  
 mutate(n.ang.fs = length(ID))   
  
dat <- dat %>%  
 group\_by(federal\_state, SD, year, half\_year, survey\_category)%>%  
 mutate(n.ang.SD = length(ID))  
  
dat$SD.ratio <- dat$n.ang.SD/dat$n.ang.fs

### Combining data frames.

In order to combine data frames the CPUE data tables (dat data frame) are joined with the effort data (fishing\_days data frame). The data frames should be joined over “survey\_category”, “federal\_state”, “year” and “half\_year”.

dat <- left\_join(x = dat, y = fishing\_days)

## Joining, by = c("year", "survey\_category", "federal\_state", "half\_year")

### Catch and release numbers

With the CPUE´s, the effort and the effort ratio split in SD`s in on data frame, the numbers of retained (n.retained) and released (n.released) cod per strata are calculated. From the off-site telephone survey in 2015, we have upper and lower values of average fishing effort per angler, which are used to calculate upper and lower total fishing days. In the end the attributes are sumed , whereby every row represents on strata and not an individual angler.

dat <- mutate(.data = dat, n.retained = mean.CPUE \* SD.ratio \* fishing\_days,  
 n.retained\_lower = mean.CPUE \* SD.ratio \* fishing\_days\_lower,  
 n.retained\_upper = mean.CPUE \* SD.ratio \* fishing\_days\_upper)  
  
dat <- mutate(.data = dat, n.released = mean.CPUE.rel \* SD.ratio \* fishing\_days,  
 n.released\_lower = mean.CPUE.rel \* SD.ratio \* fishing\_days\_lower,  
 n.released\_upper = mean.CPUE.rel \* SD.ratio \* fishing\_days\_upper)  
  
  
  
  
CANUM <- dat %>%  
 group\_by(federal\_state, SD, year, half\_year, based, survey\_category,   
 n.angsamples = length(ID), n.ret = sum(retained), n.rel = sum(released),  
 mean.CPUE, mean.CPUE.rel,   
 SD.ratio, number\_angler,   
 fishing\_days, fishing\_days\_lower, fishing\_days\_upper,  
 n.retained, n.retained\_lower, n.retained\_upper,   
 n.released, n.released\_lower, n.released\_upper)%>%  
 summarise()

# CANUM

CANUM refers to all recreational removals (harvest + dead releases). Accordingly, the release component is adjusted with the respective release mortality. Dead releases are calculated separately for sea and land-based fishing methods. The sea-based value for post-release survival is 89,9% based on Weltersbach and Strehlow (2013). Due to lack of data, post-release survival for land-based fishing is assumed to 0%. The inverse mortality rate (death rate) is used for further calculations. After multiplying the released numbers with the mortality rate, harvested and dead released cod are summed to estimate CANUM.

CANUM$death.rate <- 1  
CANUM$death.rate[CANUM$based == "Sea"] <- 0.112  
  
CANUM <- mutate(.data = CANUM, n.death.released = n.released \* death.rate,  
 n.death.released\_lower = n.released\_lower \* death.rate,  
 n.death.released\_upper = n.released\_upper \* death.rate)  
  
CANUM <- mutate(.data = CANUM, CANUM = n.retained + n.death.released,  
 CANUM\_lower = n.retained\_lower + n.death.released\_lower,  
 CANUM\_upper = n.retained\_upper + n.death.released\_upper)

Here an overview table of the CANUM per year is created.

CANUM.year <- CANUM %>%  
 group\_by(year) %>%  
 summarise(CANUM = sum(CANUM),   
 CANUM\_lower = sum(CANUM\_lower),   
 CANUM\_upper = sum(CANUM\_upper),  
 retained = sum(n.retained),   
 retained\_lower = sum(n.retained\_lower),   
 retained\_upper = sum(n.retained\_upper),  
 released = sum(n.death.released),   
 released\_lower = sum(n.death.released\_lower),   
 released\_upper = sum(n.death.released\_upper))

Here an overview table of the CANUM per strata is created.

CANUM.strata <- CANUM %>%  
 group\_by(federal\_state, SD, year, half\_year, survey\_category,) %>%  
 summarise(CANUM = sum(CANUM),   
 CANUM\_lower = sum(CANUM\_lower),   
 CANUM\_upper = sum(CANUM\_upper),  
 retained = sum(n.retained),   
 retained\_lower = sum(n.retained\_lower),   
 retained\_upper = sum(n.retained\_upper),  
 released = sum(n.death.released),   
 released\_lower = sum(n.death.released\_lower),   
 released\_upper = sum(n.death.released\_upper))

# CANUM at Length

The next step is to divide the CANUM into length categories. Therefore, a length distribution from the on-site length sampling data (dat\_length) is created.

### Create length distribution

First, the sum of all measured fish per strata (n.length.total) is calculated. Since we want to create length distributions with a minimum amount of samples, land-based samples and sea-based strata under 500 samples (500 is a self-introduced limit and a not tested assumption) are removed. Then the sum of all measured fish per length class for each stratum (n.length.lc) is calculated. With these two numbers we can calculate the percentage of each length class per strata (perc.length.lc). We summarise the dat\_length data frame, so that every length class per strata is represented by on row and not each individual measured fish.

The table output is a control if the percentages per strata sum up to 1. When n.length.total = 0 the sum.perc should be INF.

dat\_length <- dat\_length %>%  
 group\_by(SD, year, based, retained) %>%  
 mutate(n.length.total = length(length\_class))  
  
  
dat\_length$n.length.total[dat\_length$based == "Land"] <- 0  
dat\_length$n.length.total[dat\_length$n.length.total < 500] <- 0  
  
  
dat\_length <- dat\_length %>%  
 group\_by(SD, year, based, retained, length\_class) %>%  
 mutate(n.length.lc = length(length\_class))  
  
  
dat\_length$perc.length.lc <- dat\_length$n.length.lc/dat\_length$n.length.total  
  
  
dat\_length <- dat\_length %>%  
 group\_by(SD, year, based, retained, length\_class,   
 n.length.total, n.length.lc, perc.length.lc) %>%  
 summarise()  
  
  
#control if percentages are 1 per strata  
dat\_length %>%  
 group\_by(SD, year, based, retained, n.length.total) %>%  
 summarise( sum.perc = sum(perc.length.lc))

## # A tibble: 69 x 6  
## # Groups: SD, year, based, retained [69]  
## SD year based retained n.length.total sum.perc  
## <fct> <dbl> <fct> <int> <dbl> <dbl>  
## 1 22 2005 Sea 1 0 Inf  
## 2 22 2005 Land 1 0 Inf  
## 3 22 2006 Sea 1 0 Inf  
## 4 22 2006 Land 1 0 Inf  
## 5 22 2007 Sea 0 0 Inf  
## 6 22 2007 Sea 1 0 Inf  
## 7 22 2007 Land 1 0 Inf  
## 8 22 2008 Sea 0 0 Inf  
## 9 22 2008 Sea 1 0 Inf  
## 10 22 2008 Land 0 0 Inf  
## # ... with 59 more rows

### Fill the length distribution gaps

For some sea-based strata and for land-based fishing methods we do not have enough samples to build a trustworthy length-distribution. To cover these strata summed length distributions are used combining data over several years.

##### Creating length distributions over several years

First, a new column is created with the time spans and the data frame divided into sea- and land-based length distributions, since the calculations differ.

~~~~~~ Check from year to year the labels of the time spans ~~~~~~~~~~

dat\_length$year\_span <- NA  
dat\_length$year\_span[dat\_length$year > 2013] <- "2014-19"  
dat\_length$year\_span[dat\_length$year < 2014] <- "2005-13"  
  
  
dat\_length\_spansea <- filter(.data = dat\_length, based == "Sea")  
dat\_length\_spanland <- filter(.data = dat\_length, based == "Land")

For sea-based fishing methods:

First the sum of all measured fish per year span is calculated (n.length.total). Then the sum of all measured fish per length class in each year span is claulated (n.length.lc). Out of these two numbers the percentage of each length class to the total per year span (perc.length.lc) is calculated. The data frame is summed, so that every length class per year span is represented by on row and not each year.

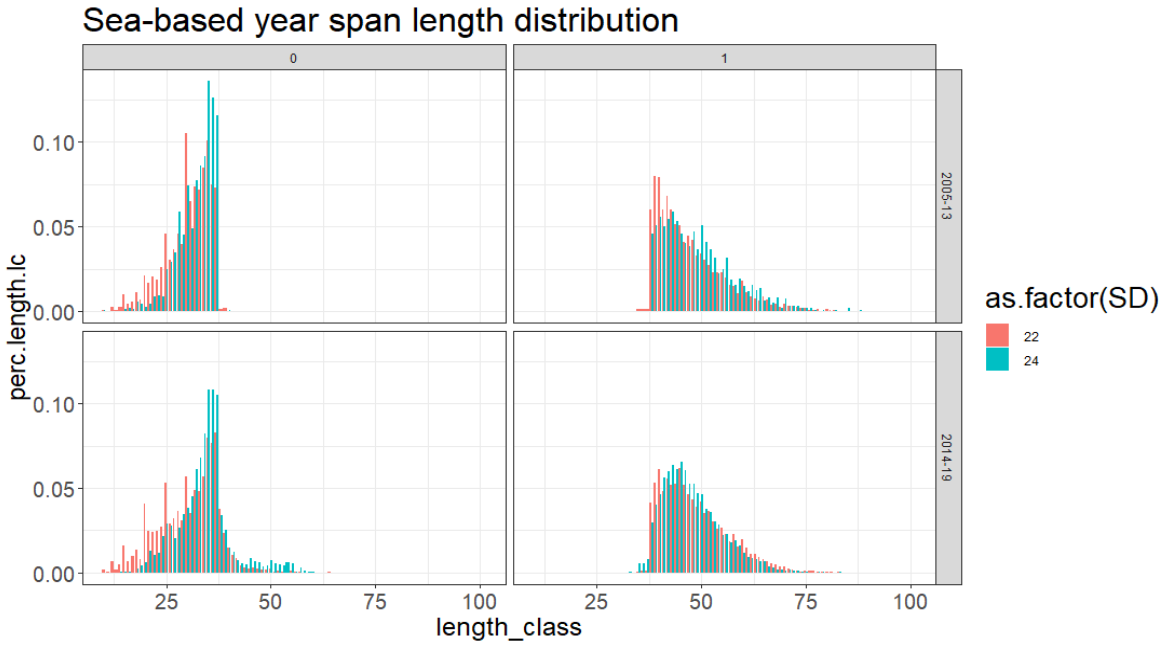
The table output is a control, if the percentages per strata sum up to 1.

dat\_length\_spansea <- dat\_length\_spansea %>%  
 group\_by(based, SD, retained, year\_span) %>%  
 mutate(n.length.total = sum(n.length.lc))  
  
  
dat\_length\_spansea <- dat\_length\_spansea %>%  
 group\_by(based,SD, year\_span, retained, length\_class) %>%  
 mutate(n.length.lc = sum(n.length.lc))  
  
dat\_length\_spansea$perc.length.lc <- dat\_length\_spansea$n.length.lc/dat\_length\_spansea$n.length.total  
  
  
dat\_length\_spansea <- dat\_length\_spansea %>%  
 group\_by(based,SD, year\_span, retained, length\_class,   
 n.length.total, n.length.lc, perc.length.lc) %>%  
 summarise()  
  
  
  
#control if percentages are 1 per strata  
dat\_length\_spansea %>%  
 group\_by(based,SD, year\_span, retained, n.length.total) %>%  
 summarise( sum.perc = sum(perc.length.lc))

## # A tibble: 8 x 6  
## # Groups: based, SD, year\_span, retained [8]  
## based SD year\_span retained n.length.total sum.perc  
## <fct> <fct> <chr> <int> <int> <dbl>  
## 1 Sea 22 2005-13 0 3900 1  
## 2 Sea 22 2005-13 1 7995 1  
## 3 Sea 22 2014-19 0 4590 1  
## 4 Sea 22 2014-19 1 8065 1  
## 5 Sea 24 2005-13 0 1571 1  
## 6 Sea 24 2005-13 1 4895 1  
## 7 Sea 24 2014-19 0 2624 1  
## 8 Sea 24 2014-19 1 6296 1

The plot shows the sea-based length distributions for each year span and SD. The plot is separated for retained and released cod.

ggplot(data = dat\_length\_spansea, aes(x = length\_class, y = perc.length.lc, fill = as.factor(SD)))+  
 geom\_col(position = "dodge")+  
 facet\_grid(year\_span ~ retained)+  
 labs(title = "Sea-based year span length distribution")+  
 theme\_bw()+  
 theme(axis.text = element\_text(size = 15),  
 axis.title = element\_text(size = 18),  
 title = element\_text(size = 20))



For land-based fishing methods:

First the sum of all measured fish is calculated (n.length.total). Then, the sum of all measured fish per length class is calculated (n.length.lc). From these two numbers the percentage of each length class to the total (perc.length.lc) is calculated. The data frame is summed, so that every length class is represented by on row.

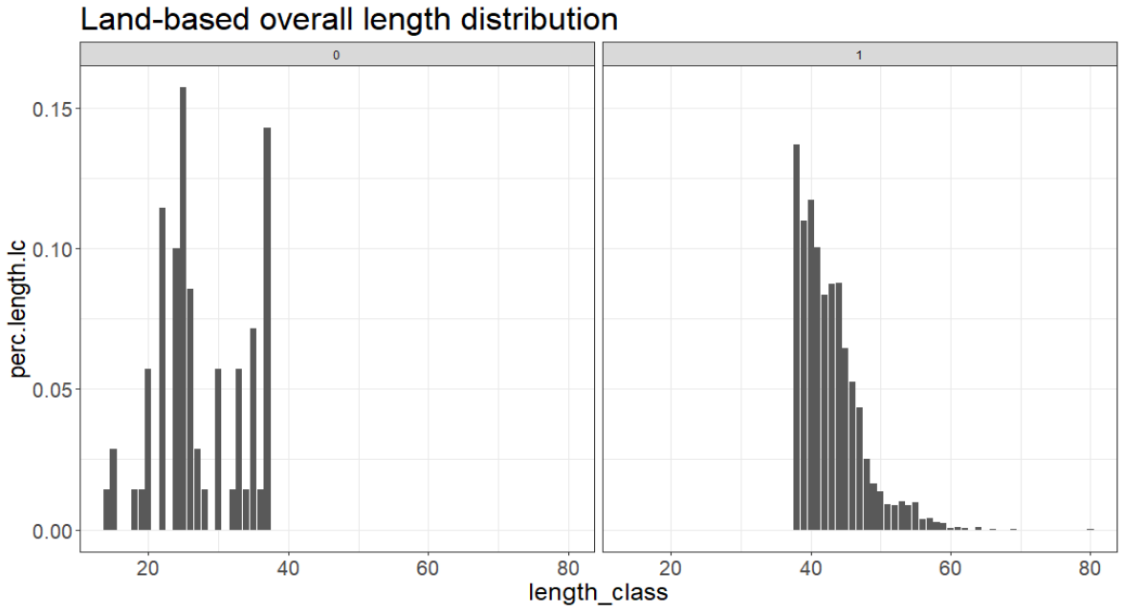
The table output is a control if the percentages per strata sum up to 1.

dat\_length\_spanland <- dat\_length\_spanland %>%  
 group\_by(based, retained)%>%  
 mutate(n.length.total = sum(n.length.lc))  
  
  
dat\_length\_spanland <- dat\_length\_spanland %>%  
 group\_by(based, retained, length\_class) %>%  
 mutate(n.length.lc = sum(n.length.lc))  
  
dat\_length\_spanland$perc.length.lc <- dat\_length\_spanland$n.length.lc/dat\_length\_spanland$n.length.total  
  
  
dat\_length\_spanland <- dat\_length\_spanland %>%  
 group\_by(based, retained, length\_class, n.length.total, n.length.lc, perc.length.lc) %>%  
 summarise()  
  
  
  
#control if percentages are 1 per strata  
dat\_length\_spanland %>%  
 group\_by(based, retained, n.length.total) %>%  
 summarise( sum.perc = sum(perc.length.lc))

## # A tibble: 2 x 4  
## # Groups: based, retained [2]  
## based retained n.length.total sum.perc  
## <fct> <int> <int> <dbl>  
## 1 Land 0 70 1  
## 2 Land 1 3530 1

The plot shows the land-based length distributions separated for retained and released cod.

ggplot(data = dat\_length\_spanland, aes(x = length\_class, y = perc.length.lc))+  
 geom\_col(position = "dodge")+  
 facet\_grid(.~ retained)+  
 labs(title = "Land-based overall length distribution")+  
 theme\_bw()+  
 theme(axis.text = element\_text(size = 15),  
 axis.title = element\_text(size = 18),  
 title = element\_text(size = 20))



##### Fill the strata without samples with the distribution from the year span

To fill strata with insufficient numbers of samples a with the combined length distribution over several years a data frame which includes only strata that had not enough samples or no samples at all is created.

The all\_strata data frame (Reminder that this data frame needs to be updated with Excel once a year!) is joined with the factors sea- & land-based and retained/released.

all\_strata\_sd <- all\_strata %>%  
 group\_by(SD, year, based)%>%  
 summarise()  
  
re <- data.frame(based = c("Sea", "Sea", "Land", "Land"),  
 retained = c(1,0,1,0))  
  
all\_strata\_sd <- left\_join(x = re, y = all\_strata\_sd)

## Joining, by = "based"

In a next step, the resulting all strata data frame is joined with dat\_length. Due to the joining procedure strata that are not in the dat\_length data frame are filled with NA´s. NA´s are changed to 0. Now all the strata that are not sampled at all and the strata with insufficient numbers of samples have a ‘0’ at the column n.length.total. Since we don´t want the strata, that have enough samples for a single length distribution, to be overwritten we only filter for the strata with n.length.total = 0.

dat\_length\_gaps <- right\_join(x = dat\_length, y = all\_strata\_sd)

## Joining, by = c("SD", "year", "based", "retained")

dat\_length\_gaps$n.length.total[is.na(dat\_length\_gaps$n.length.total)] <- 0  
  
  
dat\_length\_gaps <- filter(.data = dat\_length\_gaps, n.length.total == 0)

To fill the gaps with the combined length distributions over several years, only the columns that describe the strata (SD, year, land- & sea-based and retained) are kept and the year span they should be filled with.

Since we have two different data frames for the sea- & land-based length distributions over several years, we also split the data frame into sea- & land-based gaps. Then the sea gap strata is joined over “based”, “SD” , “retained” & “year span” with the year span length distribution and the land gap strata over “based” & “retained” with the combined length distribution over several years. Last, sea- and land-based data frames are brought together again.

dat\_length\_gaps <- dat\_length\_gaps%>%  
 group\_by(SD, year, based, retained, year\_span)%>%  
 summarise()  
  
dat\_length\_gaps$year\_span <- NA  
dat\_length\_gaps$year\_span[dat\_length\_gaps$year > 2013] <- "2014-19"  
dat\_length\_gaps$year\_span[dat\_length\_gaps$year < 2014] <- "2005-13"  
  
  
dat\_length\_gapssea <- filter(.data = dat\_length\_gaps, based == "Sea")  
dat\_length\_gapsland <- filter(.data = dat\_length\_gaps, based == "Land")  
  
  
dat\_length\_gapsfilledsea <- left\_join(x = dat\_length\_gapssea, y = dat\_length\_spansea)

## Joining, by = c("SD", "based", "retained", "year\_span")

dat\_length\_gapsfilledland <- left\_join(x = dat\_length\_gapsland, y = dat\_length\_spanland)

## Joining, by = c("based", "retained")

dat\_length\_gapsfilled <- rbind(dat\_length\_gapsfilledsea, dat\_length\_gapsfilledland)

##### Combine length distributions from sufficient strata and the filled length distributions

We now have a data frame with all length distributions (dat\_length, which include strata with insufficient numbers of samples) and a data frame with the filled length distributions (dat\_length\_gapsfilled). To combine the two sets the strata with insufficient numbers of samples need to be removed from dat\_length. A new column is created to see, if the distribution is filled or from one stratum. Last, the two data frames are combined to on final data frame containing all length distributions.

dat\_length <- filter(.data = dat\_length, n.length.total != 0)  
  
dat\_length$lc.origin <- "one\_strata"  
dat\_length\_gapsfilled$lc.origin <- "filled"  
  
dat\_length <- rbind(dat\_length, dat\_length\_gapsfilled)

### Combine CANUM and length distributions

To distribute CANUM into all length distributions the two data frames are joined together over “SD”, “year” and “based”.

CANUM\_length <- right\_join(x = dat\_length, y = CANUM)

## Joining, by = c("SD", "year", "based")

### Calculate the CANUM per Length class

Since the length distributions are separated for retained and released cod CANUM needs to be separated too and retained cod has to be divided with the retained length distribution and the dead released cod with the released length distribution. As a result, number of removed cod are shown for each length class (lc) separated for harvest and dead releases.

CANUM\_length <- mutate(.data = CANUM\_length, CANUM.lc = if\_else(retained == 1,   
 n.retained \* perc.length.lc,   
 n.death.released \* perc.length.lc))  
  
  
CANUM\_length <- mutate(.data = CANUM\_length, CANUM.lc\_lower = if\_else(retained == 1,   
 n.retained\_lower \* perc.length.lc,   
 n.death.released\_lower \*   
 perc.length.lc))  
  
  
CANUM\_length <- mutate(.data = CANUM\_length, CANUM.lc\_upper = if\_else(retained == 1,   
 n.retained\_upper \* perc.length.lc,   
 n.death.released\_upper \*   
 perc.length.lc))

# CATON

To estimate catch in weight (tons) the mean weight of cod per length class (lc) is calculated using the length-weight relationship from commercial samples.

First, the CANUM per length data frame is joined with the mass coefficient data frame, so every “SD”, “year”, “half\_year” combination will get the parameters of the length-weight relationship.

CATON\_length <- left\_join(x = CANUM\_length, y = mass\_coefficient)

## Joining, by = c("SD", "year", "half\_year")

Second, mean weight [g] is calculated for each length class with the formal: weight = a \* length ^b = exp(ln(a) + b \* ln(length[mm]))

Third, mean weight per length class is multiplied with CANUM of each length class to get the total weight of the length class.

CATON\_length <- mutate(.data = CATON\_length, mean.weight.lc = a \* (length\_class ^ b))  
  
CATON\_length <- mutate(.data = CATON\_length, CATON.g.lc = (CANUM.lc \* mean.weight.lc),  
 CATON.g.lc\_lower = (CANUM.lc\_lower \* mean.weight.lc),  
 CATON.g.lc\_upper = (CANUM.lc\_upper \* mean.weight.lc),  
   
 CATON.kg.lc = (CANUM.lc \* mean.weight.lc)/1000,  
 CATON.kg.lc\_lower = (CANUM.lc\_lower \* mean.weight.lc)/1000,  
 CATON.kg.lc\_upper = (CANUM.lc\_upper \* mean.weight.lc)/1000,  
   
 CATON.t.lc = (CANUM.lc \* mean.weight.lc)/1000000,  
 CATON.t.lc\_lower = (CANUM.lc\_lower \* mean.weight.lc)/1000000,  
 CATON.t.lc\_upper = (CANUM.lc\_upper \* mean.weight.lc)/1000000)

Here an overview table of the CATON per strata is created.

CATON.t.strata <- CATON\_length %>%  
 group\_by(federal\_state, SD, year, half\_year, survey\_category) %>%  
 summarise(CATON = sum(CATON.t.lc),  
 CATON\_lower = sum(CATON.t.lc\_lower),  
 CATON\_upper = sum(CATON.t.lc\_upper))

Here an overview table of the CATON per year is created.

CATON.t.year <- CATON\_length %>%  
 group\_by(year) %>%  
 summarise(CATON = sum(CATON.t.lc),  
 CATON\_lower = sum(CATON.t.lc\_lower),  
 CATON\_upper = sum(CATON.t.lc\_upper))

### CATON at age

Finally, CATON per length class is converted to CATON per age class. Therefore, biological length and age data from the German commercial fisheries catch sampling and/or the German Baltic International Trawl Survey (BITS) is used to divied the recreational CATON per lengthclass with commcial/BITS age classes to obtain CATON per length and age classes. This data is stored in the institute data base SnglIOR.

For further steps a forever age distribution SnglIOR\_ever is created.

SnglIOR\_ever <- SnglIOR

First, age distributions per length\_class, haly\_year and year are created to calculate the age proportion within a length class (note: not every cod with a given length has the same age) The sum of all aged fish per length class for each stratum (n.length.age) and the sum of all aged fish per age class and length class for each stratum (n.age) is calculated. From these two numbers the percentage of each age class to the total per strata is calculated (perc.age). For control purpose the percentages per length class need to sum up to 1.

SnglIOR <- SnglIOR %>%  
 group\_by(year, half\_year, length\_class) %>%  
 mutate(n.length.age = length(length\_class))  
  
  
SnglIOR <- SnglIOR %>%  
 group\_by(year, half\_year, length\_class, n.length.age, age) %>%  
 summarise(n.age = length(age))  
  
  
SnglIOR <- mutate(.data = SnglIOR, perc.age = n.age/n.length.age)  
  
  
#control if percentages are 1 per strata  
SnglIOR %>%  
 group\_by(year, half\_year, length\_class) %>%  
 summarise(sum.perc = sum(perc.age))

## # A tibble: 5,171 x 4  
## # Groups: year, half\_year [71]  
## year half\_year length\_class sum.perc  
## <int> <chr> <dbl> <dbl>  
## 1 1977 2 9 1  
## 2 1977 2 10 1  
## 3 1977 2 11 1  
## 4 1977 2 12 1  
## 5 1977 2 13 1  
## 6 1977 2 14 1  
## 7 1977 2 15 1  
## 8 1977 2 16 1  
## 9 1977 2 17 1  
## 10 1977 2 18 1  
## # ... with 5,161 more rows

In a next step, the CATON per length data frame is joined with the above calculated age distribution over “year”, “half\_year” and “length\_class”.

Due to the joining procedure every length class is now matched with an age. In those cases were several ages relate to the same length class the length class get duplicated. For length classes where no age can be joined, an NA is introduced.

CATON\_age <- left\_join(x = CATON\_length, y = SnglIOR)

## Joining, by = c("year", "length\_class", "half\_year")

The resulting data frame is split into all length class with an age and all length classes where no age could be joined.

CATON\_age\_gaps <- filter(.data = CATON\_age, is.na(age))  
CATON\_age\_gaps <- subset(x = CATON\_age\_gaps, select = -c(n.length.age, age, n.age, perc.age))  
  
  
CATON\_age <- filter(.data = CATON\_age, age != "NA")

Then the length class where no age could be joined are filled with the age distribution from a forever age distribution. For this purpose a forever age distribution is created.

At first the sum of all aged fish per length class is calculated (n.length.age). Then the sum of all aged fish per age- and length class is calculated (n.age). From these two numbers the percentage of each age class to the total per strata is calculated (perc.age). And we control if the percentages per length class sum up to 1.

SnglIOR\_ever <- SnglIOR\_ever %>%  
 group\_by(length\_class) %>%  
 mutate(n.length.age = length(length\_class))  
  
  
SnglIOR\_ever <- SnglIOR\_ever %>%  
 group\_by(length\_class, n.length.age, age) %>%  
 summarise(n.age = length(age))  
  
  
SnglIOR\_ever <- mutate(.data = SnglIOR\_ever, perc.age = n.age/n.length.age)  
  
  
#control if percentages are 1 per strata  
SnglIOR\_ever %>%  
 group\_by(length\_class) %>%  
 summarise(sum.perc = sum(perc.age))

## # A tibble: 116 x 2  
## length\_class sum.perc  
## <dbl> <dbl>  
## 1 4 1  
## 2 5 1  
## 3 6 1  
## 4 7 1  
## 5 8 1  
## 6 9 1  
## 7 10 1  
## 8 11 1  
## 9 12 1  
## 10 13 1  
## # ... with 106 more rows

Now the forever age distribution is joined with the data frame which contains the length classes without an age over “length\_class” and leave a filled dataset without NA (anyNA = FALSE)

CATON\_age\_gapsfilled <- left\_join(x = CATON\_age\_gaps, y = SnglIOR\_ever)

## Joining, by = "length\_class"

anyNA(x = CATON\_age\_gapsfilled$age)

## [1] FALSE

We now have a data frame where the age distributions comes from one year (CATON\_age) and a data frame with the forever age distributions (CATON\_age\_gapsfilled). In both data frames a new column is created to determine the origin of the distribution(forever or from a certain year). These data frames are then combined to one data frame with all age distributions.

CATON\_age$age.origin <- "one\_year"  
CATON\_age\_gapsfilled$age.origin <- "forever"  
  
  
CATON\_age <- rbind(CATON\_age, CATON\_age\_gapsfilled)

Now we can calculate CATON per length- and age class. Therefore, CATON per length class is multiplied with the percentages of the age distribution.

CATON\_age <- mutate(.data = CATON\_age, CANUM.lc.age = CANUM.lc \* perc.age,  
 CANUM.lc.age\_lower = CANUM.lc\_lower \* perc.age,  
 CANUM.lc.age\_upper = CANUM.lc\_upper \* perc.age,  
 CATON.t.lc.age = CATON.t.lc \* perc.age,  
 CATON.t.lc.age\_lower = CATON.t.lc\_lower \* perc.age,  
 CATON.t.lc.age\_upper = CATON.t.lc\_upper \* perc.age)

# Overview of the results

The last steps create a final data frame with all values and smaller data frames for specific strata.

Final data frame: Creating the data frame with all values per strata (federal\_state, SD, year, half\_year, based, survey\_category, retained) including: - the CPUE values, - the fishing effort, - the total CANUM, - the CANUM per length including the length distribution, - the CATON per length - the CATON per length and age including the age distribution

values\_all <- subset(x = CATON\_age,   
 select = c(federal\_state, SD, year, half\_year, based, survey\_category, retained,   
 n.angsamples, n.ret, n.rel,  
 mean.CPUE, mean.CPUE.rel, SD.ratio, number\_angler,   
 fishing\_days, fishing\_days\_lower, fishing\_days\_upper,   
 n.retained, n.retained\_lower, n.retained\_upper,  
 n.released, n.released\_lower, n.released\_upper,  
 death.rate,   
 n.death.released, n.death.released\_lower, n.death.released\_upper,  
 CANUM, CANUM\_lower, CANUM\_upper,  
   
 lc.origin, year\_span, length\_class, n.length.total, n.length.lc, perc.length.lc,   
 CANUM.lc, CANUM.lc\_lower, CANUM.lc\_upper,  
   
 a, b, mean.weight.lc,   
 CATON.g.lc, CATON.g.lc\_lower, CATON.g.lc\_upper,   
 CATON.kg.lc, CATON.kg.lc\_lower, CATON.kg.lc\_upper,   
 CATON.t.lc, CATON.t.lc\_lower, CATON.t.lc\_upper,  
   
 age.origin, age, n.length.age, n.age, perc.age,   
 CANUM.lc.age, CANUM.lc.age\_lower, CANUM.lc.age\_upper,   
 CATON.t.lc.age, CATON.t.lc.age\_lower, CATON.t.lc.age\_upper))

Specific data frame: This data frame shows the CANUM and CATON per length class and age for each stratum.

values\_length\_age <- values\_all %>%  
 group\_by(federal\_state, SD, year, half\_year, based, survey\_category, retained,   
 length\_class, age,   
 CANUM.lc.age, CANUM.lc.age\_lower, CANUM.lc.age\_upper,  
 CATON.t.lc.age, CATON.t.lc.age\_lower, CATON.t.lc.age\_upper)%>%  
 summarise()

This data frame shows the total CANUM and CATON for each stratum.

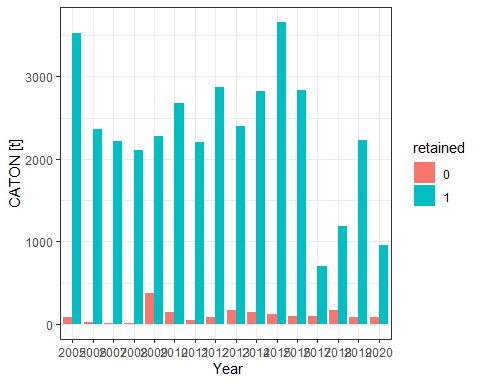
values\_strata <- values\_all %>%  
 group\_by(federal\_state, SD, year, half\_year, based, survey\_category, retained)%>%  
 summarise(CANUM = sum(CANUM.lc.age),  
 CANUM\_lower = sum(CANUM.lc.age\_lower),  
 CANUM\_upper = sum(CANUM.lc.age\_upper),  
 CATON = sum(CATON.t.lc.age),  
 CATON\_lower = sum(CATON.t.lc.age\_lower),  
 CATON\_upper = sum(CATON.t.lc.age\_upper))

This data frame shows the total CANUM and CATON for each year.

values\_year <- values\_all %>%  
 group\_by(year, retained)%>%  
 summarise(CANUM = sum(CANUM.lc.age),  
 CANUM\_lower = sum(CANUM.lc.age\_lower),  
 CANUM\_upper = sum(CANUM.lc.age\_upper),  
 CATON = sum(CATON.t.lc.age),  
 CATON\_lower = sum(CATON.t.lc.age\_lower),  
 CATON\_upper = sum(CATON.t.lc.age\_upper))

This plot visualizes the total CATON per year of retained and dead released cod.

ggplot(data = values\_year, aes(x = as.factor(year), y = CATON, fill = as.factor(retained)))+  
 geom\_col(position = "dodge")+  
 labs(x = " Year", y = "CATON [t]", fill = "retained")+  
 theme\_bw()



# safe the results

#write.table(values\_all, "results/values\_all.csv", sep = ";", dec = ",", row.names = FALSE)  
#write.table(values\_length\_age, "results/values\_length\_age.csv", sep = ";", dec = ",", row.names = FALSE)  
#write.table(values\_strata, "results/values\_strata.csv", sep = ";", dec = ",", row.names = FALSE)  
#write.table(values\_year, "results/values\_year.csv", sep = ";", dec = ",", row.names = FALSE)