JRC SCIENTIFIC INFORMATION SYSTEMS AND DATABASES REPORT

The JRC MEDITS R script

A tool to analyse MEDITS data during STECF EWGs

MANNINI ALESSANDRO

2020

This publication is a Technical report by the Joint Research Centre (JRC), the European Commission's science and knowledge service. It aims to provide evidence-based scientific support to the European policymaking process. The scientific output expressed does not imply a policy position of the European Commission. Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use that might be made of this publication. For information on the methodology and quality underlying the data used in this publication for which the source is neither Eurostat nor other Commission services, users should contact the referenced source. The designations employed and the presentation of material on the maps do not imply the expression of any opinion whatsoever on the part of the European Union concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries.

Contact information
Name: Alessandro Mannini
Address: EC Joint Research Centre (JRC) – Directorate D.02 Water and Marine Resources – Via Enrico Fermi 2749 (Bldg 5a) – 21027 Ispra (VA) Italy
Email: alessandro.mannini@ec.europa.eu
Tel.: +39 0332 78 5784

EU Science Hub
https://ec.europa.eu/jrc

JRC119776

PDF ISBN 978-92-76-10598-5 doi:10.2760/5799

Luxembourg: Publications Office of the European Union, 2020

How to cite this report: Mannini A. *The JRC MEDITS R script,* Publications Office of the European Union, Luxembourg, 2020, ISBN 978-92-76-10598-5, doi:10.2760/5799, JRC119176.

Contents

Acknowledgements

All the fisheries experts testing the script during the stock assessment STECF EWG meetings

Authors

Alessandro Mannini

Abstract

Main fishery stock assessment methods needs not only fishery dependent information but also fishery independet information which are, in almost all the cases, derived from scientific surveys at sea. In the Mediterranean Sea the main fishery independet information come from the Mediterranean International Trawl Survey (MEDITS) carried out since 1994. Primary Medits data such as, haul information, catches in weight and number by hauls and biological data of target species are routinely collected and stored in dedicated files. These file are made available to the STECF EWG stock assessment meetings through an official Data Call launched by DGMare. This tool developed as an R script gives the opportunity, starting from these files, to compute standardized biomass and density indexes and length frequency distributions, checks hauls positions and other main hauls characteristics and many other useful outputs.

1 Introduction

Main fishery stock assessment methods needs to tune commercial data fishery independet information which in almost all the cases are derived from scientific surveys at sea. In the Mediterranean Sea the main fishery independent information on demersal species exploited by MS come from the International bottom trawl survey in the Mediterranean) (MEDITS) carried out since 1994.

The MEDITS survey programme intends to produce basic information on benthic and demersal species in term of population distribution as well as demographic structure, on the continental shelves and along the upper slopes at a global scale in the Mediterranean Sea, through systematic bottom trawl surveys. For more informations and technical details of MEDITS surveys please visit *http://www.sibm.it/MEDITS%202011/principale%20project.htm.*

Primary Medits data such as haul information, catches in weight and number by hals and biological data for target species are routinely collected and stored in dedicated files. These file are made available to the STECF EWG stock assessment meetings (*https://stecf.jrc.ec.europa.eu/reports/medbs*) thorugh an official Data Call launched by DGMare yearly (for more information please visit *https://datacollection.jrc.ec.europa.eu/data-calls*.

JRC developed an ad-hoc R script given the opportunity, starting from these files, to compute standardized biomass and density indexes and length frequency distributions, species occurrence, sex ratio vector by length and many other useful outputs (quality checks, haul positions, survey period etc).

2 Data format

Standard formats are defined for the storage and to facilitate the exchange of the data produced by the MEDITS surveys. The exchange files are in .csv format, using semicolon as field separator.

Five file types are defined in order to store and exchange the data:

Type A: Characteristics of haul - this file includes the data on bottom temperature and stratification, formely included in TD and TT type files;

Type B: Catches by haul;

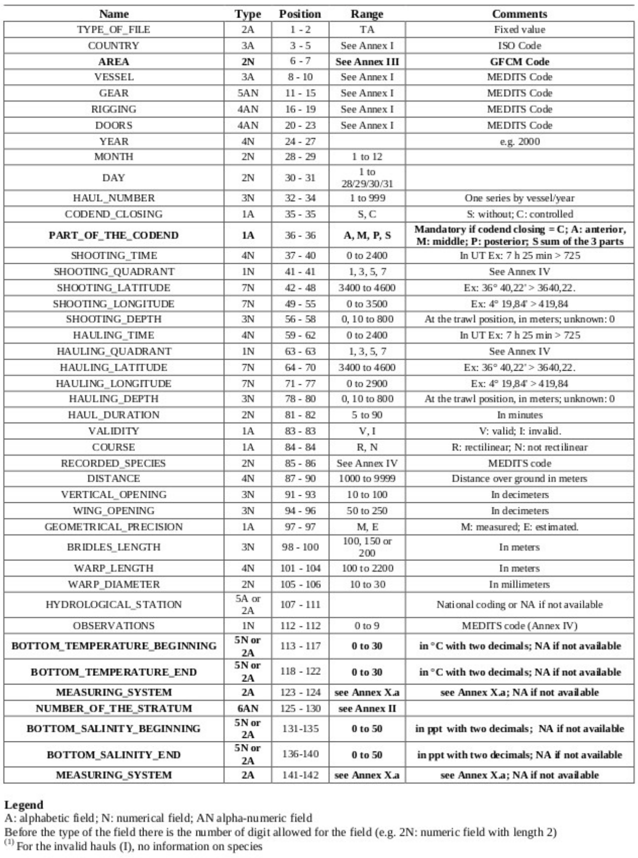
Type C: Length, sex, and maturity at aggregated level;

Type E: Age weight and maturity by length at individual level.

Type L: collection of marine litter data

Actually, only three files are called during the official Data Call (Type A, B and C). In the following figure the format of the three files are showed.

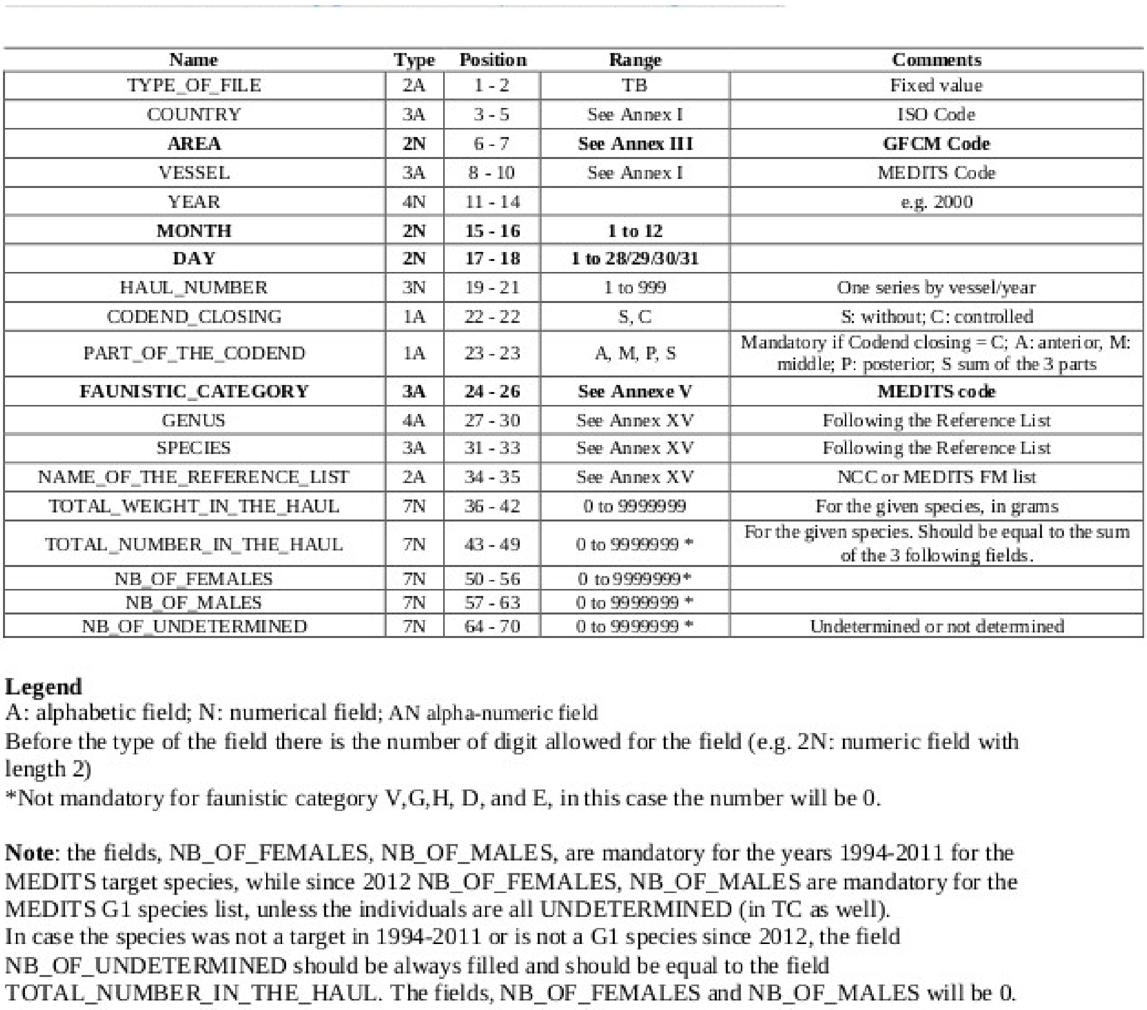
Figure 1. Format of the type A files (Data on the haul)



Name	Type	Position	Range	Comments
TYPE_OF_FILE	2A	1 - 2	TA	Fixed value
COUNTRY	3A	3 - 5	See Annex I	ISO Code
AREA	**2N**	6 - 7	**See Annex III**	**GFCM Code**
VESSEL	3A	8 - 10	See Annex I	MEDITS Code
GEAR	5AN	11 - 15	See Annex I	MEDITS Code
RIGGING	4AN	16 - 19	See Annex I	MEDITS Code
DOORS	4AN	20 - 23	See Annex I	MEDITS Code
YEAR	4N	24 - 27		e.g. 2000
MONTH	2N	28 - 29	1 to 12	
DAY	2N	30 - 31	1 to 28/29/30/31	
HAUL_NUMBER	3N	32 - 34	1 to 999	One series by vessel/year
CODEND_CLOSING	1A	35 - 35	S, C	S: without; C: controlled
PART_OF_THE_CODEND	**1A**	36 - 36	**A, M, P, S**	**Mandatory if codend closing = C; A: anterior, M: middle; P: posterior; S sum of the 3 parts**
SHOOTING_TIME	4N	37 - 40	0 to 2400	In UT Ex: 7 h 25 min > 725
SHOOTING_QUADRANT	1N	41 - 41	1, 3, 5, 7	See Annex IV
SHOOTING_LATITUDE	7N	42 - 48	3400 to 4600	Ex: 36° 40,22' > 3640,22.
SHOOTING_LONGITUDE	7N	49 - 55	0 to 3500	Ex: 4° 19,84' > 419,84
SHOOTING_DEPTH	3N	56 - 58	0, 10 to 800	At the trawl position, in meters; unknown: 0
HAULING_TIME	4N	59 - 62	0 to 2400	In UT Ex: 7 h 25 min > 725
HAULING_QUADRANT	1N	63 - 63	1, 3, 5, 7	See Annex IV
HAULING_LATITUDE	7N	64 - 70	3400 to 4600	Ex: 36° 40,22' > 3640,22.
HAULING_LONGITUDE	7N	71 - 77	0 to 2900	Ex: 4° 19,84' > 419,84
HAULING_DEPTH	3N	78 - 80	0, 10 to 800	At the trawl position, in meters; unknown: 0
HAUL_DURATION	2N	81 - 82	5 to 90	In minutes
VALIDITY	1A	83 - 83	V, I	V: valid; I: invalid.
COURSE	1A	84 - 84	R, N	R: rectilinear; N: not rectilinear
RECORDED_SPECIES	2N	85 - 86	See Annex IV	MEDITS code
DISTANCE	4N	87 - 90	1000 to 9999	Distance over ground in meters
VERTICAL_OPENING	3N	91 - 93	10 to 100	In decimeters
WING_OPENING	3N	94 - 96	50 to 250	In decimeters
GEOMETRICAL_PRECISION	1A	97 - 97	M, E	M: measured; E: estimated.
BRIDLES_LENGTH	3N	98 - 100	100, 150 or 200	In meters
WARP_LENGTH	4N	101 - 104	100 to 2200	In meters
WARP_DIAMETER	2N	105 - 106	10 to 30	In millimeters
HYDROLOGICAL_STATION	5A or 2A	107 - 111		National coding or NA if not available
OBSERVATIONS	1N	112 - 112	0 to 9	MEDITS code (Annex IV)
BOTTOM_TEMPERATURE_BEGINNING	**5N or 2A**	113 - 117	**0 to 30**	**in °C with two decimals; NA if not available**
BOTTOM_TEMPERATURE_END	**5N or 2A**	118 - 122	**0 to 30**	**in °C with two decimals; NA if not available**
MEASURING_SYSTEM	**2A**	123 - 124	**see Annex X.a**	**see Annex X.a; NA if not available**
NUMBER_OF_THE_STRATUM	**6AN**	125 - 130	**see Annex II**	
BOTTOM_SALINITY_BEGINNING	**5N or 2A**	131-135	**0 to 50**	**in ppt with two decimals; NA if not available**
BOTTOM_SALINITY_END	**5N or 2A**	136-140	**0 to 50**	**in ppt with two decimals; NA if not available**
MEASURING_SYSTEM	**2A**	141-142	**see Annex X.a**	**see Annex X.a; NA if not available**

Legend
A: alphabetic field; N: numerical field; AN alpha-numeric field
Before the type of the field there is the number of digit allowed for the field (e.g. 2N: numeric field with length 2)
(1) For the invalid hauls (I), no information on species

Figure 2. Format of the type B files (Catches by haul)

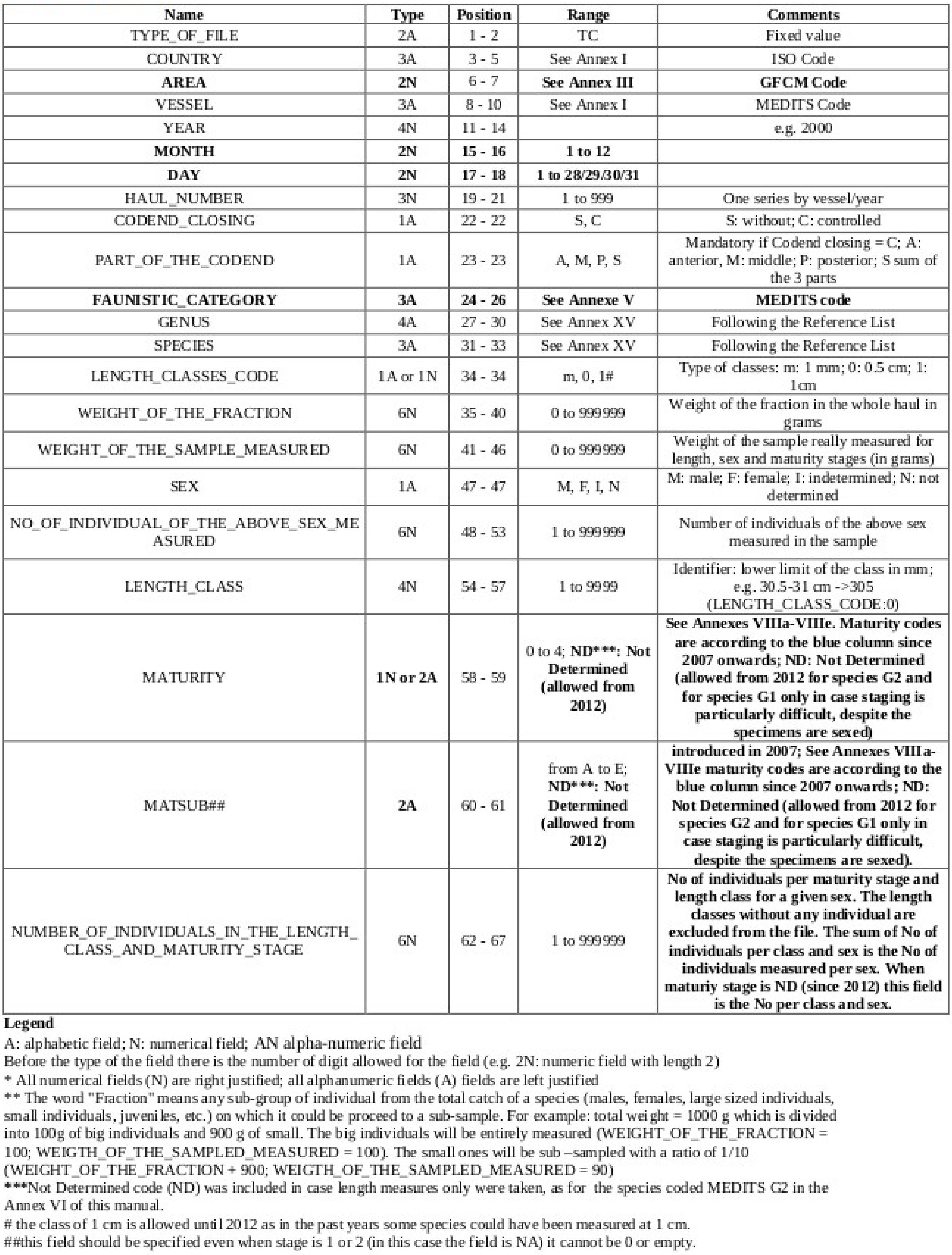


Name	Type	Position	Range	Comments
TYPE_OF_FILE	2A	1 - 2	TB	Fixed value
COUNTRY	3A	3 - 5	See Annex I	ISO Code
AREA	**2N**	**6 - 7**	**See Annex III**	**GFCM Code**
VESSEL	3A	8 - 10	See Annex I	MEDITS Code
YEAR	4N	11 - 14		e.g. 2000
MONTH	**2N**	**15 - 16**	**1 to 12**	
DAY	**2N**	**17 - 18**	**1 to 28/29/30/31**	
HAUL_NUMBER	3N	19 - 21	1 to 999	One series by vessel/year
CODEND_CLOSING	1A	22 - 22	S, C	S: without; C: controlled
PART_OF_THE_CODEND	1A	23 - 23	A, M, P, S	Mandatory if Codend closing = C; A: anterior, M: middle; P: posterior; S sum of the 3 parts
FAUNISTIC_CATEGORY	**3A**	**24 - 26**	**See Annexe V**	**MEDITS code**
GENUS	4A	27 - 30	See Annex XV	Following the Reference List
SPECIES	3A	31 - 33	See Annex XV	Following the Reference List
NAME_OF_THE_REFERENCE_LIST	2A	34 - 35	See Annex XV	NCC or MEDITS FM list
TOTAL_WEIGHT_IN_THE_HAUL	7N	36 - 42	0 to 9999999	For the given species, in grams
TOTAL_NUMBER_IN_THE_HAUL	7N	43 - 49	0 to 9999999 *	For the given species. Should be equal to the sum of the 3 following fields.
NB_OF_FEMALES	7N	50 - 56	0 to 9999999*	
NB_OF_MALES	7N	57 - 63	0 to 9999999 *	
NB_OF_UNDETERMINED	7N	64 - 70	0 to 9999999 *	Undetermined or not determined

Legend
A: alphabetic field; N: numerical field; AN alpha-numeric field
Before the type of the field there is the number of digit allowed for the field (e.g. 2N: numeric field with length 2)
*Not mandatory for faunistic category V,G,H, D, and E, in this case the number will be 0.

Note: the fields, NB_OF_FEMALES, NB_OF_MALES, are mandatory for the years 1994-2011 for the MEDITS target species, while since 2012 NB_OF_FEMALES, NB_OF_MALES are mandatory for the MEDITS G1 species list, unless the individuals are all UNDETERMINED (in TC as well).
In case the species was not a target in 1994-2011 or is not a G1 species since 2012, the field NB_OF_UNDETERMINED should be always filled and should be equal to the field TOTAL_NUMBER_IN_THE_HAUL. The fields, NB_OF_FEMALES and NB_OF_MALES will be 0.

Figure 3. *Format of type C files (length and aggregated biological parameters)*



Name	Type	Position	Range	Comments
TYPE_OF_FILE	2A	1 - 2	TC	Fixed value
COUNTRY	3A	3 - 5	See Annex I	ISO Code
AREA	**2N**	**6 - 7**	**See Annex III**	**GFCM Code**
VESSEL	3A	8 - 10	See Annex I	MEDITS Code
YEAR	4N	11 - 14		e.g. 2000
MONTH	**2N**	**15 - 16**	**1 to 12**	
DAY	**2N**	**17 - 18**	**1 to 28/29/30/31**	
HAUL_NUMBER	3N	19 - 21	1 to 999	One series by vessel/year
CODEND_CLOSING	1A	22 - 22	S, C	S: without; C: controlled
PART_OF_THE_CODEND	1A	23 - 23	A, M, P, S	Mandatory if Codend closing = C; A: anterior, M: middle; P: posterior; S sum of the 3 parts
FAUNISTIC_CATEGORY	**3A**	**24 - 26**	**See Annexe V**	**MEDITS code**
GENUS	4A	27 - 30	See Annex XV	Following the Reference List
SPECIES	3A	31 - 33	See Annex XV	Following the Reference List
LENGTH_CLASSES_CODE	1A or 1N	34 - 34	m, 0, 1#	Type of classes: m: 1 mm; 0: 0.5 cm; 1: 1cm
WEIGHT_OF_THE_FRACTION	6N	35 - 40	0 to 999999	Weight of the fraction in the whole haul in grams
WEIGHT_OF_THE_SAMPLE_MEASURED	6N	41 - 46	0 to 999999	Weight of the sample really measured for length, sex and maturity stages (in grams)
SEX	1A	47 - 47	M, F, I, N	M: male; F: female; I: indetermined; N: not determined
NO_OF_INDIVIDUAL_OF_THE_ABOVE_SEX_MEASURED	6N	48 - 53	1 to 999999	Number of individuals of the above sex measured in the sample
LENGTH_CLASS	4N	54 - 57	1 to 9999	Identifier: lower limit of the class in mm; e.g. 30.5-31 cm ->305 (LENGTH_CLASS_CODE:0)
MATURITY	**1N or 2A**	58 - 59	0 to 4; **ND***: Not Determined (allowed from 2012)**	**See Annexes VIIIa-VIIIe. Maturity codes are according to the blue column since 2007 onwards; ND: Not Determined (allowed from 2012 for species G2 and for species G1 only in case staging is particularly difficult, despite the specimens are sexed)**
MATSUB##	**2A**	60 - 61	from A to E; **ND***: Not Determined (allowed from 2012)**	**introduced in 2007; See Annexes VIIIa-VIIIe maturity codes are according to the blue column since 2007 onwards; ND: Not Determined (allowed from 2012 for species G2 and for species G1 only in case staging is particularly difficult, despite the specimens are sexed).**
NUMBER_OF_INDIVIDUALS_IN_THE_LENGTH_CLASS_AND_MATURITY_STAGE	6N	62 - 67	1 to 999999	**No of individuals per maturity stage and length class for a given sex. The length classes without any individual are excluded from the file. The sum of No of individuals per class and sex is the No of individuals measured per sex. When maturiy stage is ND (since 2012) this field is the No per class and sex.**

Legend

A: alphabetic field; N: numerical field; AN alpha-numeric field

Before the type of the field there is the number of digit allowed for the field (e.g. 2N: numeric field with length 2)

* All numerical fields (N) are right justified; all alphanumeric fields (A) fields are left justified

** The word "Fraction" means any sub-group of individual from the total catch of a species (males, females, large sized individuals, small individuals, juveniles, etc.) on which it could be proceed to a sub-sample. For example: total weight = 1000 g which is divided into 100g of big individuals and 900 g of small. The big individuals will be entirely measured (WEIGHT_OF_THE_FRACTION = 100; WEIGTH_OF_THE_SAMPLED_MEASURED = 100). The small ones will be sub –sampled with a ratio of 1/10 (WEIGHT_OF_THE_FRACTION + 900; WEIGTH_OF_THE_SAMPLED_MEASURED = 90)

*******Not Determined code (ND) was included in case length measures only were taken, as for the species coded MEDITS G2 in the Annex VI of this manual.

the class of 1 cm is allowed until 2012 as in the past years some species could have been measured at 1 cm.

##this field should be specified even when stage is 1 or 2 (in this case the field is NA) it cannot be 0 or empty.

According to the Annex I of the Official Mediterranean Data Call these data are requested in slight different format.

Type A. Medits haul data, in accordance to MEDITS instruction manual Version 7, 2013 (*http://www.sibm.it/MEDITS%202011/principaledownload.htm*).

Type B. Medits catch by haul data, all species , in accordance to MEDITS instruction manual Version 7, 2013(*http://www.sibm.it/MEDITS%202011/principaledownload.htm*).

Type C.Medits length and biological parameters by haul data, all species , in accordance to MEDITS instruction manual, Version 7, 2013(*http://www.sibm.it/MEDITS%202011/principaledownload.htm*).

Once files have been sent by Member States they are stored in the dedicated database in which other identifier fields are created (id and upload_id). The developed tool is able to manage both format: the original MEDITS format and the modified one extracted from the JRC database. The latter is the one commonly provided during STECF EWG stock assessment meetings.

3 Data analysis

Biomass and density indexes.

To estimate the mean, variance, standard deviation and coefficient of variation of the abundance indices in number and weight by square kilometer with a stratified random sampling, the following formulations are used (Cochran, 1977 and Souplet, 1996):

Average by strata

❑

$$\overline{x}_i = \frac{\sum_{j=1}^{n_i} x_{i,j}}{\sum_{j=1}^{n_i} A}$$

xi,j is the weight of individuals caught in the individual hauls of the stratum and Ai,j is the corresponding swept area . The variance is calculated by the following formulas::

$$S_{x_i^2} = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} A_{i,j} \left(\frac{x_{i,j}}{A_{i,j}} - \overline{x}_i \right)^2$$

Abundance index of the main strata (shelf, slope and total) is computed accoding to the following formula (cfr. Pennington e Brown, 1981):

$$I = \sum_{i=1}^{N} W_i \overline{x}_i$$

Wi is the weight of each individual stratum calculated as the ratio between the area of the stratum and the total area of the study area. The variance in this case is given by the formula:

$$\operatorname{var}(I) = \sum_{i=1}^{N} \frac{W_i^2 S_{x_i^2}}{\sum_{j=1}^{n_i} A_{i,j}} (1 - f_i)$$

as fi is the ratio between the swept area and the area of the stratum, i.e. the correction factor for finite populations (fpc).

Standard deviation is:

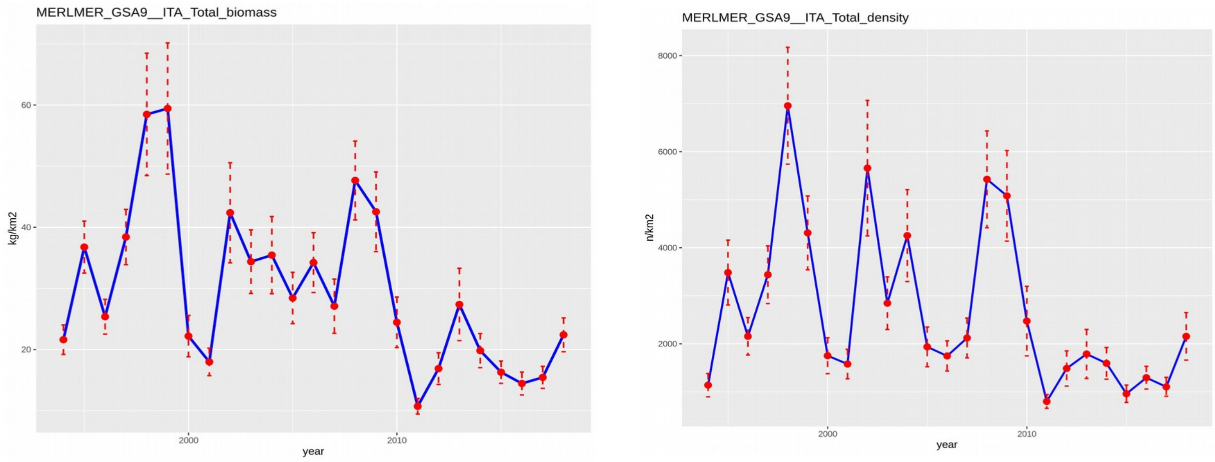
$$\text{s.d.} = \sqrt[2]{\operatorname{Var}(I)}$$

and Coefficient of Variation is:

CV% = (s.d./I) * 100

In the following figure biomass and density index of European Hake in the GSA9. Data needs to plot the indexes are also saved in a csv file.

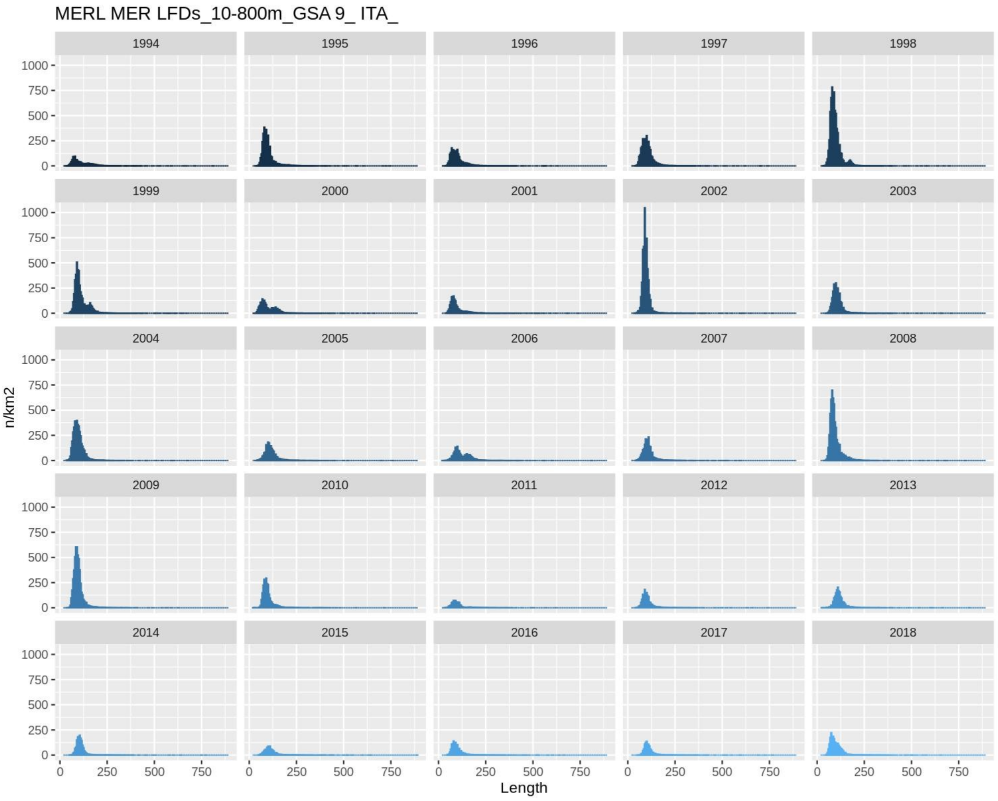
Figure 4. Biomass (left) and density (right) indexes by square kilometer



Length frequency distributions (LFDs)

Length distribution are standardized by square kilometer applying to each length classes the formulas used for abundance indexes. Length distributions are computed by sex (female, male, not sexed) and by total individuals. Since, almost all the individuals not sexed are smaller (juveniles) to create final female and male distributions the indeterminante distributon is splitted between sexes according to a ratio of 50% by each length classes. Female and male distribution computed according to sex assigment in TC file are saved as LFDFEM and LFDMALE while the ones created splitting not sexed individuals are saved as LFDFF and LFDMM. In the following figure an example of LFDs output. Data needs to plot the LFD are also saved in a csv file.

Figure 5. Length frequency distributions standardized by square kilometer



Occurence

Occurence is computed as ratio between total number of positive hauls (e.g. total number of hauls a species was caught) and total number of hauls carried out during the survey. The main output is a csv file.

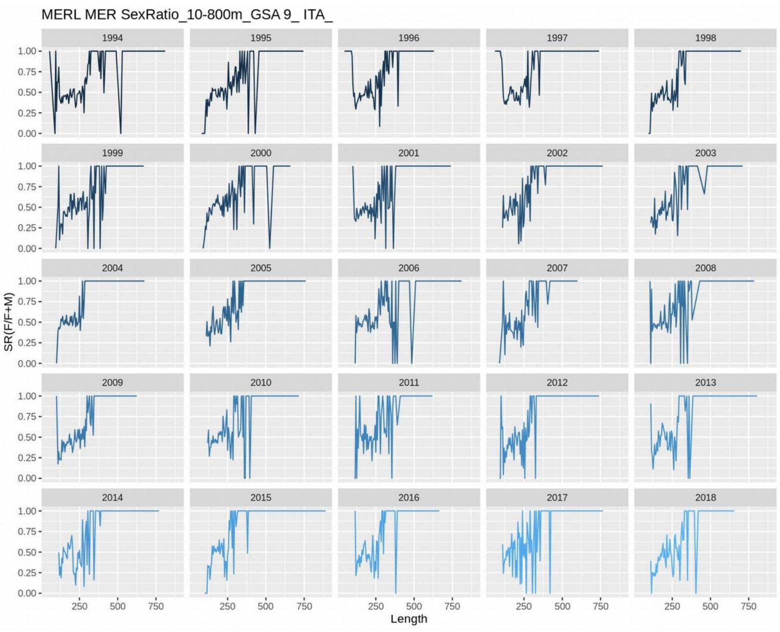
Sexratio

Having standardized length distribution by sex sex ratio vector by length classes and year is computed as ratio by each length classes between female and female plus male

SR=FF/(FF+MM)

In the following figure sex artio by year is showed. Data needs to plot the sex ratio vector by year are also saved in a csv file.

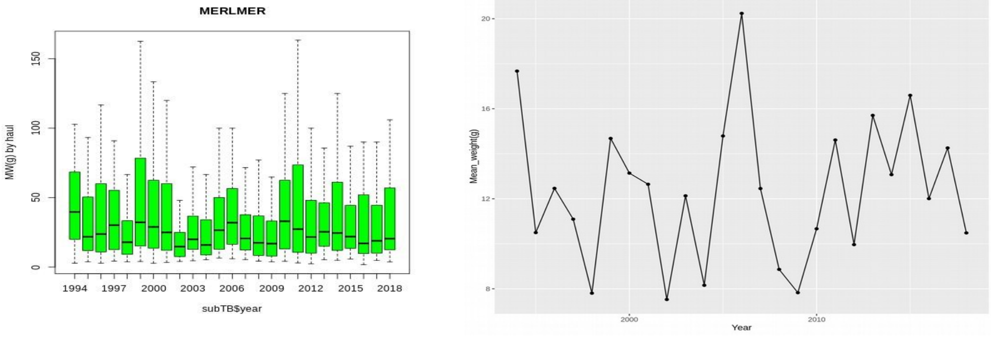
Figure 6. Sex ratio vector by length



Mean weight

Mean weight is computed by hauls and by year, In the first case is the ratio by biomass and density by hauls, in the latter case the same ratio is applied at yearly level. In the following figures results of the two computation are shown. Data needs to plot mean weight by year are also saved in a csv file.

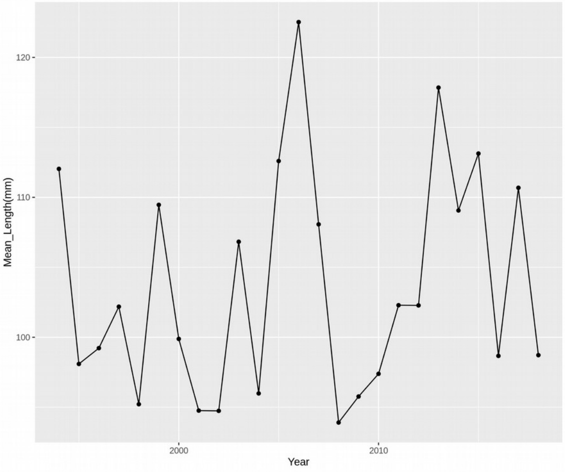
Figure 7. Boxplot of the mean weight by haul (left side) and yearly mean weight (right) as ratio between biomass and density indexes



Mean length

Mean length value is computed by year on the standardized length distribution calculated on the total area explored and with sex combined. Data needs to plot mean length by year are also saved in a csv file.

Figure 8. Yearly Mean length



Swept area

Swept area is computed according to a simple formula:

in the TA file are stored distance covered and net wing opening so it is only a matter to run

sqkm=TAn$wing_opening/10000000*TAn$distance

assignment of each haul to one of the five stratifcation MEDITS depth strata (A=0-50m, B=51-100m, C=101-200m, D=201-500m and E=501-800m) is done according to the haul mean depth

meandepth=(TAn$shooting_depth+TAn$hauling_depth)/2

Some plots saved according to the swept area, wing opening and depth

Figure 9. Swept area by hauls versus mean depth by five main MEDITS strata

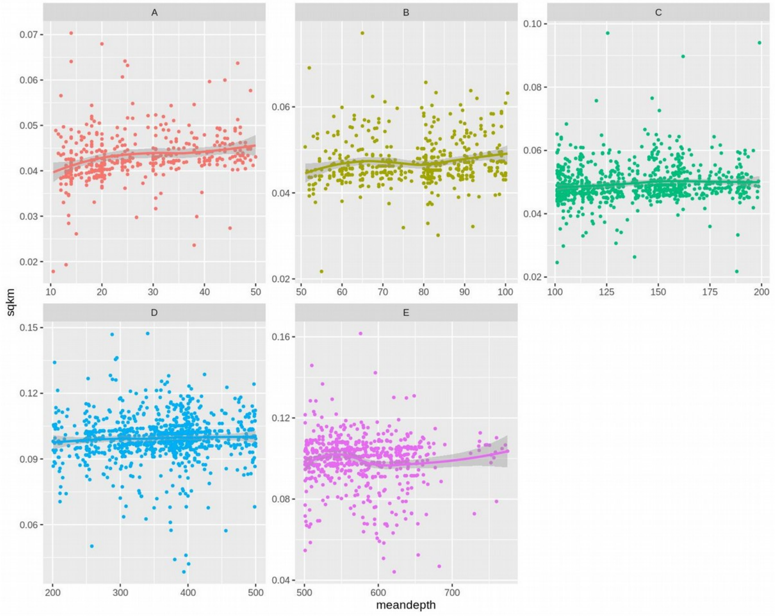


Figure 10. Wing opening by hauls versus mean depth by five main MEDITS strata

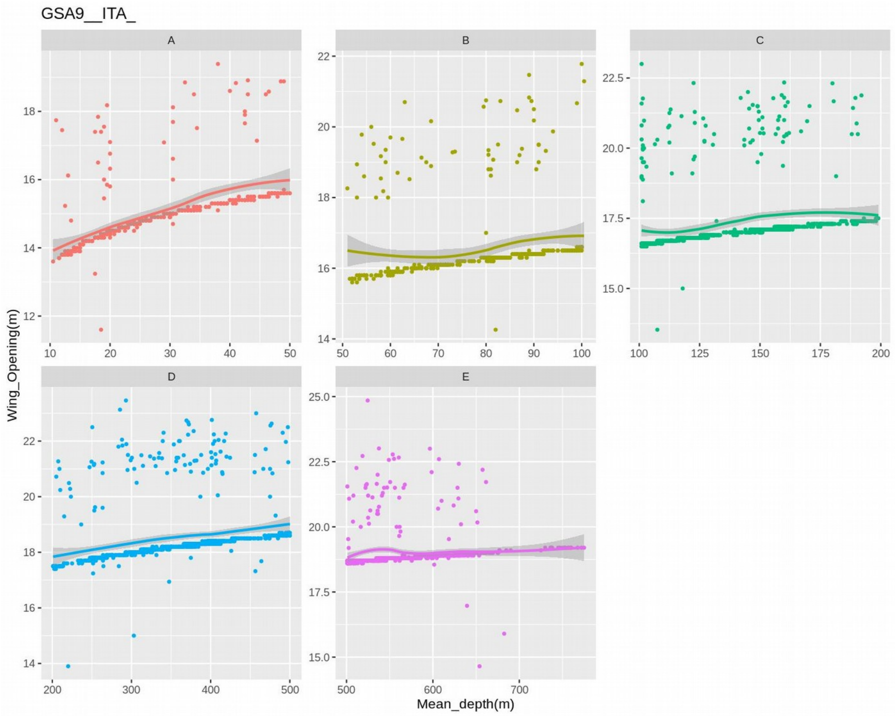
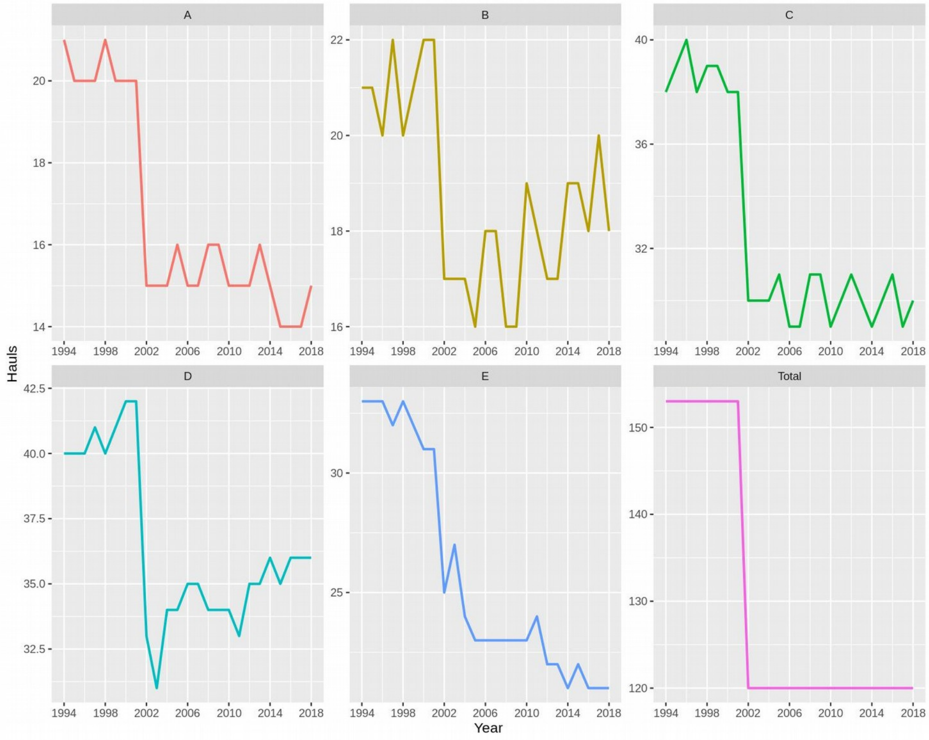


Figure 11. Yearly number of hauls by five main MEDITS strata

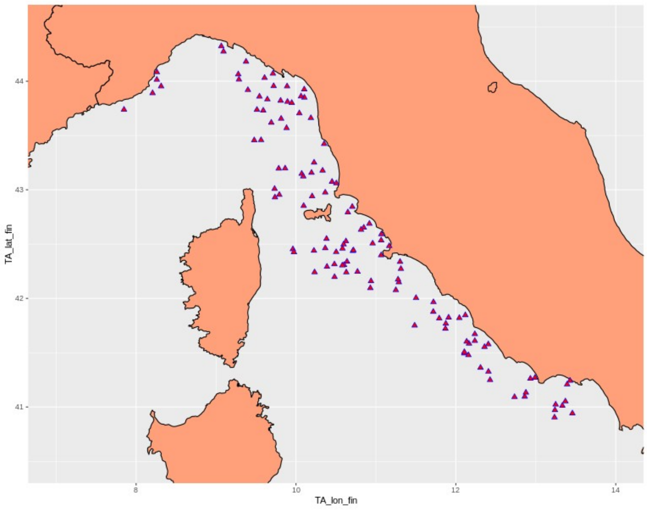


Data checks

Underneath are listed main quality checks:

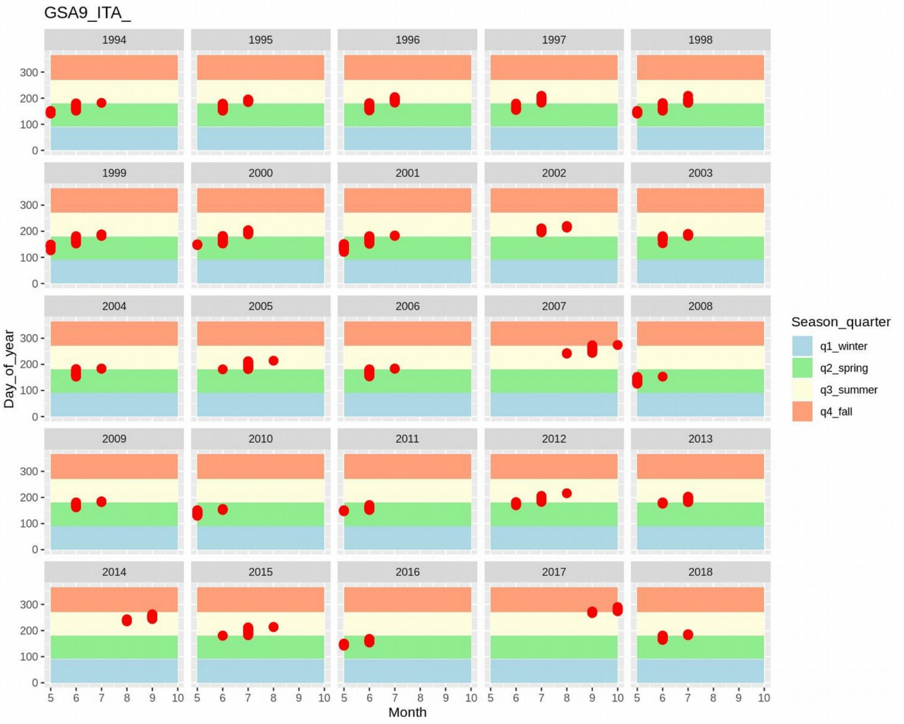
1) Check hauls position by year

Figure 12. Hauls positon map



2) Check survey period

Figure 13. Survey period according to the fourth quarter of the year



3) If there are hauls reported in TB file (catches) that are not recorded in TA file (hauls)

File: TBhaul_no_in_TAhaul.csv

4) If there are hauls for which total weight and number values are different between TB (catches) and TC (biology)

File: TBTCtoCheck_....csv

5) If there are TC hauls in which sample weight is higher than a prefixed treshold. According to MEDITS protocol catches subsamples are allowed. Usually a reference sample ratio shouldn't be too high so sub samples applied represent the whole haul catches.

File: TctoCheck......csv

This kind of check is important because raising factor applied to number of individuals in TC file is computed according to the ratio between weight of the sample measured and weight of the fraction caught. Misreporting weight could have affected on the raising procedure generating standardized length frequency distributions (unrealistic abundance by length).

6) Maximum length observed by year by sex and total.

File: maxLength.csv

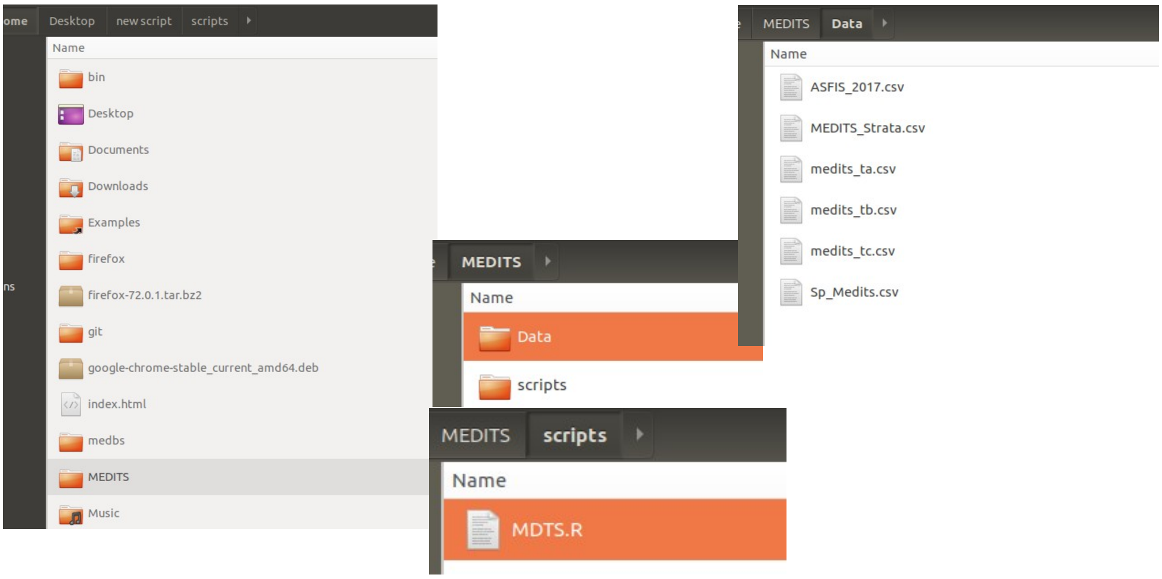
4 What we have to set before launch the script ?

Before to launch script the user has to set folder and main inputs.

Setting folder tree

The script (MDTS.R) have to be stored in a folder for example called “scripts” while data and support files in another one called “Data”. Both folder have to be place in the same main folder for example called “MEDITS”

Figure 14. Folder tree setting



Data file are the MEDITS file (TA, TB and TC) in one of the format: MEDITS Handbokk version 9 or DCF database output, while supporting file are MEDITS_strata, Sp_Medits and ASFIS_2017.

MEDITS_strata list strata information as reported in Annex II of the Medits handbook version 9 (Startification scheme (by stratum number)

Sp_MEDITS contains the Medits code to identify species and the corrisponding scientific name.

ASFIS_2017 contains the species three FAO alpha code and other information.

Setting script

After the first crunch of code regarding libraries there is a dedicated section which have to be set. This section is the only part of the code that user can deal with it.

```
# Specify gsa, gen, spec codes and recruits cutoff length (if needed!) ####
areacode=c ("9") # use c("1","2") for many GSA
countrycode=c("ITA") # Uppercase use c (“ITA”, “FRA”) for more countries
```

GSA number and country code must be coherent. You cant’n set GSA as 9 (Ligurian and Northern Tyrrhenian Sea) and Greece as country.

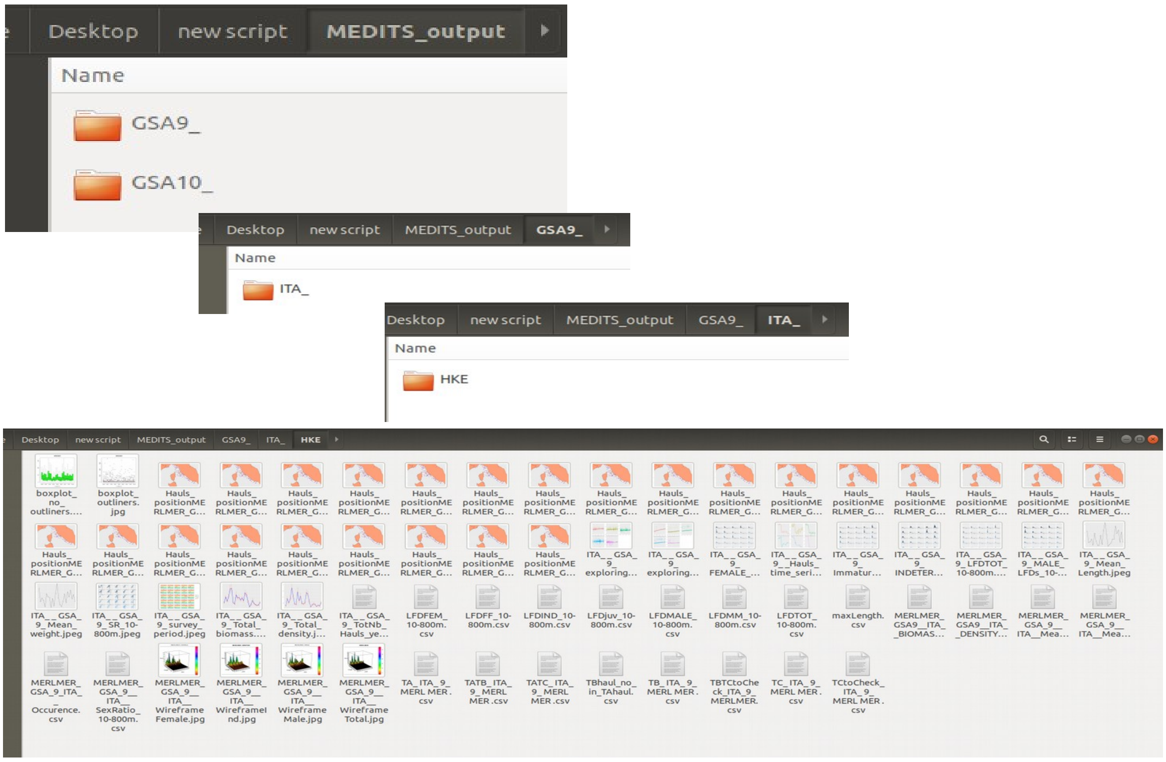
```
state="ITA_" #assign country code to output file name (use underscore separator for two or more countries e.g. ITA_HRV_ for 2)

gsa="9_" #assign gsa code to output file name (use underscore separator for two or more gsa i.e. 9_11_ for two GSAs)

format="MEDITS" # If format is the DCF one (provided in the EWG STECF set "DCF" otherwise if the files are in the MEDITS handbook format version 9 please set "MEDITS")

# Setting step to length classes (lfstep) and maxratiosampling to check sampling ratio apply in TC file####

lfstep=5 # set as 1 if you are working with crustaceans and 5 for fish and cephalopods

maxratiosampling=5 # Set value according to what do you think should be a reasonable max value sampling factor (e.g. maximum value of subsample apply to the total catch in a haul)

## Setting species ##

sspp="HKE" ## use the species FAO three alpha code
```

5 Where output are stored ?

Outputs are stored in a folder called “MEDITS_output” which will be create by the code as subfolder of the main one.

This folder contains many subfolders which are create to identify in unique way area, country and species analized. In the last one are stored the main outputs.

Figure 15. Output folders and main outcomes


Desktop
new script
MEDITS_output
Name
GSA9_
GSA10_
GSA9_
ITA_
HKE
boxplot_no_outliners....
boxplot_outliners.jpg
Hauls_positionMERLMER_G...
ITA__GSA_9_exploring...
ITA__GSA_9_FEMALE_...
ITA__GSA_9__Hauls_time_seri...
ITA__GSA_9_Immatur...
ITA__GSA_9_INDETER...
ITA__GSA_9_LFDTOT_10-800m....
ITA__GSA_9_MALE_LFDs_10-...
ITA__GSA_9_Mean_Length.jpeg
ITA__GSA_9_Mean_weight.jpeg
ITA__GSA_9_SR_10-800m.jpeg
ITA__GSA_9_survey_period.jpeg
ITA__GSA_9_Total_biomass....
ITA__GSA_9_Total_density.j...
ITA__GSA_9_TotNb_Hauls_ye...
LFDFEM_10-800m.csv
LFDFF_10-800m.csv
LFDIND_10-800m.csv
LFDjuv_10-800m.csv
LFDMALE_10-800m.csv
LFDMM_10-800m.csv
LFDTOT_10-800m.csv
maxLength.csv
MERLMER_GSA9__ITA_BIOMAS...
MERLMER_GSA9__ITA_DENSITY...
MERLMER_GSA_9__ITA__Mea...
MERLMER_GSA_9__ITA__Mea...
MERLMER_GSA_9_ITA__Occurence.csv
MERLMER_GSA_9__ITA__SexRatio_10-800m.csv
MERLMER_GSA_9__ITA__WireframeFemale.jpg
MERLMER_GSA_9__ITA__WireframeInd.jpg
MERLMER_GSA_9__ITA__WireframeMale.jpg
MERLMER_GSA_9__ITA__WireframeTotal.jpg
TA_ITA_9_MERL MER.csv
TATB_ITA_9_MERL MER.csv
TATC_ITA_9_MERL MER.csv
TBhaul_no_in_TAhaul.csv
TB_ITA_9_MERL MER.csv
TBTCtoCheck_ITA_9_MERLMER.csv
TC_ITA_9_MERL MER.csv
TCtoCheck_ITA_9_MERL MER.csv

6 The MEDITS JRC Rcode

```
# title: "Computing MEDITS indexes and LFD"
# author: "Alessandro Mannini"
# date: "January 24th, 2020"
# Copyright (C) <2014> <Alessandro Mannini>
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/
remove(list=ls())
library(doBy)
library(lattice)
library(gclus)
library(ggplot2)
library(reshape)
library(data.table)
library(ggmap)
library(mapdata)
library(maps)
library(stringr)
library(dplyr)

# Specify gsa,gen, spec codes and recruits cutoff length (if needed!) ####
areacode=c("9") # use c("1","2") for many GSA
countrycode=c("ITA")# Uppercase use c (“ITA”, “FRA”) for more countries
#GSA number and country code must be coherent. You can set 9 (Ligurian and Northern Tyrrhenian Sea) setting Greece as country.
state="ITA_" #assign country code to output file name (use underscore separator for two or more countries e.g. ITA_HRV_ for 2 )
gsa="9_" #assign gsa code to output file name (use underscore separator for two or more gsa i.e. 9_11_ for two GSAs)
```

```
format="MEDITS" # If format is the DCF one (provided in the EWG STECF set "DCF otherwise if the files are in the MEDITS handbook format version 9 please set "MEDITS")

# Setting step to length classes (lfstep) and maxratiosampling to check sampling ratio apply in TC file####

lfstep=5 # set as 1 if you are working with crustaceans and 5 for fish and cephalopods

maxratiosampling=5 # Set value according to what do you think should be a reasonable max value sampling factor (e.g. maximum value of subsample apply to the total catch in a haul)

## Setting species ##

sspp="HKE" ## use the species FAO three alpha code

# if you know Medits code

# gen<-"PAGE" # Uppercase genus MEDITS code

# spec<-"ERY"# Uppercase species MEDITS code

# mdts=fread("../Data/Sp_Medits.csv")

# asfis=fread("../Data/ASFIS_2017.csv")

# mdts=mdts[,c(2,3,7)]

# asfis=asfis[,c(3,4)]

# colnames(mdts)[2]="Scientific_name"

# sp_list=merge(mdts,asfis,by="Scientific_name",all = T)

# sp_list=subset(sp_list[!is.na(sp_list$code),])### NA values dismiss

# sp_list=subset(sp_list[!is.na(sp_list$`3A_CODE`),])### NA values dismiss

# sp_list$gen=str_sub(sp_list$code,start=1,end=4)

# sp_list$spec=str_sub(sp_list$code,start=5,end=7)

# code=paste0(gen,spec)

# alpha_code <- unique(sp_list$`3A_CODE`[which(sp_list$code==code)])


# if you prefer use 3alpha code

alpha_code=sspp

mdts=fread("../Data/Sp_Medits.csv")

asfis=fread("../Data/ASFIS_2017.csv")

mdts=mdts[,c(2,3,7)]

asfis=asfis[,c(3,4)]

colnames(mdts)[2]="Scientific_name"

sp_list=merge(mdts,asfis,by="Scientific_name",all = T)

sp_list=subset(sp_list[!is.na(sp_list$code),])### NA values dismiss

sp_list=subset(sp_list[!is.na(sp_list$`3A_CODE`),])### NA values dismiss
```

```r
sp_list$gen=str_sub(sp_list$code,start=1,end=4)
sp_list$spec=str_sub(sp_list$code,start=5,end=7)
gen=unique(sp_list$gen[which(sp_list$`3A_CODE`==alpha_code)])
spec=unique(sp_list$spec[which(sp_list$`3A_CODE`==alpha_code)])

# Extract data ####
if (format=="DCF"){
TAn <- fread("../Data/medits_ta.csv")
TAn <- as.data.frame(subset(TAn, (area %in% areacode) & (country %in% countrycode)))
TAn=subset(TAn[TAn$validity=="V",]) # subset only valid hauls
droplevels(TAn)
TBn <- fread("../Data/medits_tb.csv")
TBn <- as.data.frame(subset(TBn, (area %in% areacode) & (country %in% countrycode) & (genus %in% gen) & (species %in% spec)))
droplevels(TBn)
TCn <- fread("../Data/medits_tc.csv")
TCn <- as.data.frame(subset(TCn, (area %in% areacode) & (country %in% countrycode) & (genus %in% gen) & (species %in% spec)))
droplevels(TCn)
}else{
TAn <- fread("../Data/TA.csv")
TAn <- as.data.frame(subset(TAn, (AREA %in% areacode) & (COUNTRY %in% countrycode)))
TAn=subset(TAn[TAn$VALIDITY=="V",]) # subset only valid hauls
droplevels(TAn)
TAn$tf <- NA
TAn <- TAn[,c(1:12,14:36,44,37:40,13)]
TAn$upload_id <- NA
header_A <- c("id" , "country" ,"area" ,"vessel" ,
"gear" , "rigging" ,"door" ,"year" ,
"month" , "day" ,"haul_number" ,"codend_closing" ,
"shooting_time" , "shooting_quadrant" ,"shooting_latitude" ,"shooting_longitude" ,
"shooting_depth" , "hauling_time" ,"hauling_quadrant" ,"hauling_latitude" ,
"hauling_longitude" , "hauling_depth" ,"hauling_duration" ,"validity" ,
"course" , "recorded_species" ,"distance" ,"vertical_opening" ,
"wing_opening" , "geometrical_precision" ,"bridles_length" ,"warp_length" ,
"warp_diameter" , "hydrological_station" ,"observations" ,"tf" ,
"todeb" ,"tofin" ,"method" ,"nstrate" ,
"partit" ,"upload_id" )
colnames(TAn) <- header_A

TBn <- fread("../Data/TB.csv")
TBn <- as.data.frame(subset(TBn, (AREA %in% areacode) & (COUNTRY %in% countrycode) & (GENUS %in% gen) & (SPECIES %in% spec)))
droplevels(TBn)
TBn$tf <- NA
TBn <- TBn[,c(1:5,8:20,6,7)]
TBn$upload_id <- NA
header_B <- c("id" , "country" ,"area" ,"vessel" ,"year" ,"haul_number" ,
"codend_closing", "partit" ,"catfau" ,"genus" ,"species" ,"lref" ,
"ptot" ,"nbtot" ,"nbfem" ,"nbmal" ,"nbind" ,"tf" ,
"month" ,"day" ,"upload_id")
colnames(TBn) <- header_B
TCn <- fread("../Data/TC.csv")
TCn <- as.data.frame(subset(TCn, (AREA %in% areacode) & (COUNTRY %in% countrycode) & (GENUS %in% gen) & (SPECIES %in% spec)))
droplevels(TCn)
TCn$tf <- NA
TCn <- TCn[,c(1:5,8:10,12:20,22,21,23,6:7,11)]
TCn$upload_id <- NA
header_C <- c("id" , "country" ,"area" ,"vessel" ,"year" ,"haul_number",
"codend_closing" ,"partit" ,"genus" ,"species" ,"codlon" ,"pfrac" ,
"pechan" ,"sex" ,"nbsex" ,"length_class" ,"maturity" ,"nblon" ,
"matsub" ,"tf" ,"month" ,"day" ,"catfau" ,"upload_id" )
colnames(TCn) <- header_C
}

# read in stratification table of the survey Copyrigth Tristan Rouyer
stratification_scheme <- fread("../Data/MEDITS_Strata.csv")
stratum<-stratification_scheme[stratification_scheme$AREA %in% areacode,]#extraction based on GSA
stratum<-stratum[stratum$COUNTRY %in% countrycode,]# extraction based on country
```

```
stratum$AREASTRATA[which(stratum$AREASTRATA==0)] <- as.integer(1)

## prepare the TA file for the next elaborations##
meandepth=(TAn$shooting_depth+TAn$hauling_depth)/2
sqkm=TAn$wing_opening/10000000*TAn$distance
id2=paste(TAn$country,TAn$area,TAn$year,TAn$haul_number,sep="")
TAn["strata"]=NA
TAn=cbind(TAn,meandepth,sqkm,id2)
TAn$strata[]=TAn$nstrate[]

## Assigning strata code on meandepth value not on "codestrata"
for (i in 1:length(TAn$strata))
if(TAn$meandepth[i]>0 & TAn$meandepth[i] < 51){TAn$strata[i]="A"}else{
if(TAn$meandepth[i]>=51 & TAn$meandepth[i] < 101){TAn$strata[i]="B"}else{
if(TAn$meandepth[i]>=101 & TAn$meandepth[i] < 201){TAn$strata[i]="C"}else{
if(TAn$meandepth[i]>=201 & TAn$meandepth[i] <501){TAn$strata[i]="D"}else{
TAn$strata[i]="E"}}}}
unique(TAn$strata)
table(TAn$strata)

jones <- as.data.frame(table(TAn$strata))
moss <- data.frame(Var1=c("A","B","C","D","E"),Freq1=as.integer(c(0,0,0,0,0)))
cevert <- right_join(jones,moss)
cevert$FQ <- rowSums(cevert[,c("Freq", "Freq1")], na.rm=TRUE)
# prepare TB for next elaborations#
id2=paste(TBn$country,TBn$area,TBn$year,TBn$haul_number,sep="")
TBn=cbind(TBn,id2)
# create a new database merging TATB
TATBn=merge(TAn,TBn,by="id2",all=T)
TATBn$W_sqkm=TATBn$ptot/TATBn$sqkm/1000
TATBn$N_sqkm=TATBn$nbtot/TATBn$sqkm

WD=getwd()
dir_t=paste0("../MEDITS_output","/","GSA",gsa,sep="")
dir.create(file.path(dir_t,state,alpha_code), recursive = T)
```

```
getwd()
setwd(file.path(dir_t,state,alpha_code))
write.table(TAn,file=paste("TA_",state,gsa,gen,spec,".csv"),sep=";",row.names=F)
write.table(TBn,file=paste("TB_",state,gsa,gen,spec,".csv"),sep=";",row.names=F)
write.table(TCn,file=paste("TC_",state,gsa,gen,spec,".csv"),sep=";",row.names=F)
write.table(TATBn,file=paste("TATB_",state,gsa,gen,spec,".csv"),sep=";",row.names=F)


# Checking hauls positions ####
# TA_lon=formatC(TAn$shooting_longitude, width = 4, format = "d", flag = "0")
TA_lon=formatC(as.numeric(TAn$shooting_longitude),width=7,format='f',digits=2,flag='0')


TA_gr_lon=substr(TA_lon, 1, 2)
TA_gr_lon=as.integer(TA_gr_lon)
which(is.na(TA_gr_lon))
TA_gr_lon[is.na(TA_gr_lon)] <- 0


TA_mi_lon=substr(TA_lon, 3, 4)
TA_mi_lon=as.integer(TA_mi_lon)
which(is.na(TA_mi_lon))
TA_mi_lon[is.na(TA_mi_lon)] <- 0


TA_se_lon=substr(TA_lon, 6, 7)
TA_se_lon=as.integer(TA_se_lon)
which(is.na(TA_se_lon))
TA_se_lon[is.na(TA_se_lon)] <- 0


TA_lon_fin=TA_gr_lon+(TA_mi_lon/60)+(TA_se_lon/6000)# Attention are in decimal of minutes. Need to find conversion factor maybe 6000
which(is.na(TA_lon_fin))
TAn$TA_lon_fin=TA_lon_fin


TA_lat=formatC(as.numeric(TAn$shooting_latitude),width=7,format='f',digits=2,flag='0')


TA_gr_lat=substr(TA_lat, 1, 2)
```

```
TA_gr_lat=as.integer(TA_gr_lat)

which(is.na(TA_gr_lat))

TA_gr_lat[is.na(TA_gr_lat)] <- 0


TA_mi_lat=substr(TA_lat, 3, 4)

TA_mi_lat=as.integer(TA_mi_lat)

which(is.na(TA_mi_lat))

TA_mi_lat[is.na(TA_mi_lat)] <- 0


TA_se_lat=substr(TA_lat, 6, 7)

TA_se_lat=as.integer(TA_se_lat)

which(is.na(TA_se_lat))

TA_se_lat[is.na(TA_se_lat)] <- 0


TA_lat_fin=TA_gr_lat+(TA_mi_lat/60)+(TA_se_lat/6000)# Attention are in decimal of minutes. Need to find conversion factor maybe 6000

which(is.na(TA_lat_fin))

TAn$TA_lat_fin=TA_lat_fin

TAn$TA_lon_fin=TA_lon_fin


TAn$TA_lon_fin=ifelse(TAn$shooting_quadrant=="7",-TAn$TA_lon_fin,TAn$TA_lon_fin)

TA_lon_fin=TAn$TA_lon_fin


xlims <- range(pretty(TA_lon_fin))

ylims <- range(pretty(TA_lat_fin))


plot_list1 = list()

for (i in min(TAn$year):max(TAn$year))

{plot_list1[[i]]=ggplot(data=TAn[TAn$year==i,], aes(x =TA_lon_fin, y= TA_lat_fin))+

borders("worldHires",fill="lightsalmon",colour="black",xlim=xlims,ylim=ylims)+geom_point(pch = 24, cex=2, col="blue", bg="red", lwd=2)+

coord_quickmap(xlim = xlims,ylim=ylims)

ggtitle(paste0(state,"_","GSA",gsa," ",gen,spec," ",i))

}


# Save plots to tiff. Makes a separate file for each plot.
```

```
for (i in min(TAn$year):max(TAn$year)) {
file_name = paste0("Hauls_position",gen,spec,"_GSA_",gsa,"_",state,"_",i, ".tiff", sep="")
tiff(file_name, width = 800, height = 600, units = "px")
print(plot_list1[[i]])
dev.off()
}

# Exploring hauls time series ####
hauls_table=aggregate(TAn$haul_number,list(TAn$year,TAn$strata),length)
names(hauls_table)=c("Year","Stratum","Hauls")
hauls_table2=hauls_table
hauls_table_tot=aggregate(Hauls~Year, hauls_table, sum)
hauls_table_tot$Stratum=rep("Total",nrow(hauls_table_tot))
hauls_table_tot=hauls_table_tot[,c(1,3,2)]
hauls_table=rbind(hauls_table,hauls_table_tot)
Hauls_time_series=ggplot(hauls_table, aes(x=Year, y=Hauls,color=Stratum)) +
geom_line(lwd=0.9) + facet_wrap(~Stratum,scales ="free") +
theme(legend.position="none")+
scale_x_continuous(breaks = seq(min(TAn$year),max(TAn$year),by=4))
Hauls_time_series
ggsave(filename=paste(state,"_","GSA_",gsa,"_Hauls_time_series.jpeg"),width = 10, height = 8, dpi = 150, units = "in", plot=Hauls_time_series)
hauls_table3 <- cast(hauls_table,Year~Stratum,sum)
write.csv(hauls_table3,file=paste(state,"_","GSA_",gsa,"TotNb_Hauls_year.csv"),row.names = F)

# Exploring swept area ####
table(is.na(TAn$meandepth))
table(is.na(TAn$sqkm))
table(is.na(TAn$distance))
exploring_sweptarea=ggplot(TAn, aes(x=meandepth, y=sqkm,color=strata)) +
geom_point(cex=0.9) + facet_wrap(~strata,scales ="free") +
theme(legend.position="none") + geom_smooth()
exploring_sweptarea
ggsave(filename=paste(state,"_","GSA_",gsa,"exploring_sweptarea.jpeg"),width = 10, height = 8, dpi = 150, units = "in", plot=exploring_sweptarea)
```

```
exploring_sweptarea

# Exploring wing opening by meandepth area ####
table(is.na(TAn$wing_opening))
exploring_wingopening=ggplot(TAn, aes(x=meandepth,
y=wing_opening/10,color=strata)) +
geom_point(cex=0.9) + facet_wrap(~strata,scales ="free") +
theme(legend.position="none") + geom_smooth()+
ggtitle(paste0("GSA",gsa,"_",state)) +
xlab("Mean_depth(m)") + ylab("Wing_Opening(m)")
exploring_wingopening
ggsave(filename=paste(state,"_","GSA_",gsa,"exploring_wingopening.jpeg"),width =
10, height = 8, dpi = 150, units = "in", plot=exploring_wingopening)

# Exploring survey period ####
ggplot(TAn, aes(x=month, y=year))+
geom_point(size=2.5,colour=2)+
scale_y_continuous(breaks = seq(min(TAn$year),max(TAn$year),by=2))+
theme(axis.text.x = element_text(angle = 0, hjust = 1,size=9))
TA_date=TAn[,8:10]
TA_date$data=as.Date(paste0(TA_date$year,"-",TA_date$month,"-",TA_date$day))
invisible(TA_date[order(TA_date$data),])
TA_date$dayofyear=as.numeric(format(TA_date$data, "%j"))
TA_date=TA_date[!duplicated(TA_date$data), ]
ggplot(TA_date, aes(x=TA_date$month, y=TA_date$dayofyear))+
geom_point(size=2.5,colour=2)+
facet_grid(~TA_date$year)+
scale_x_continuous(breaks =
seq(min(TA_date$month),max(TA_date$month),by=1),minor_breaks = F)+
theme(axis.text.x = element_text(angle = 0, hjust = 1,size=9))+
ggtitle(paste0("GSA",gsa,state)) +
xlab("Month") + ylab("Day_of_year")
survey_period=ggplot(TA_date, aes(x=month, y=dayofyear))+
geom_rect(data=NULL,aes(xmin=min(month),xmax=max(month),ymin=0,ymax=90,
fill="q1_winter"))+
geom_rect(data=NULL,aes(xmin=min(month),xmax=max(month),ymin=91,ymax=1
80,
fill="q2_spring"))+
geom_rect(data=NULL,aes(xmin=min(month),xmax=max(month),ymin=181,ymax=
270,
fill="q3_summer"))+
geom_rect(data=NULL,aes(xmin=min(month),xmax=max(month),ymin=271,ymax=
365,
fill="q4_fall"))+
scale_fill_manual('Season_quarter',
values = c("lightblue","lightgreen","lightyellow","lightsalmon"))+
geom_point(size=3,colour="red")+
facet_wrap(~TA_date$year)+
ggtitle(paste0("GSA",gsa,state)) +
xlab("Month") + ylab("Day_of_year")

survey_period
ggsave(filename=paste(state,"_","GSA_",gsa,"survey_period.jpeg"),width = 10,
height = 8, dpi = 150, units = "in", plot=survey_period)

# Species occurence ########
tothaulyear=tapply(TATBn$haul_number.x,TATBn$year.x,length)
TATBn['pos']=1
TATBn['neg']=0
TATBn['occurence']=NA
TATBn$nbtot[is.na(TATBn$nbtot)]=0
occur=ifelse((TATBn$nbtot>0),
(TATBn$occurence=TATBn$pos),
(TATBn$occurence=TATBn$neg))
TATBn=cbind(TATBn,occur)
tothaulposyear=tapply(TATBn$occur,TATBn$year.x,sum)
occurence=tothaulposyear/tothaulyear
year=sort(unique(TAn$year))
occurence=as.data.frame(cbind(year,occurence))
names(occurence)=c("year","occurence")
occ=lm(occurence~year,data=occurence)
par(mar=c(4.1 ,2.1, 2.1, 2.1))
par(mfrow=(c(2,2)))
plot(occ,col=3,lty=2,lwd=2)
```

```
(spearmancor=cor(year,occurence,method="spearman"))

write.csv(occurence,file=paste(gen,spec,"_GSA_",gsa,state,"_","Occurence.csv",sep=""),row.names=FALSE)


# Abundance Indexes by strata####

strata=aggregate(TAn$sqkm,list(TAn$year,TAn$strata),sum)

names(strata)=c("year","stratum","sweptarea")

weigthbystrata=aggregate(TATBn$ptot,list(TATBn$year.y,TATBn$strata),sum)

names(weigthbystrata)=c("year","stratum","weigth")

numberbystrata=aggregate(TATBn$nbtot,list(TATBn$year.y,TATBn$strata),sum)

names(numberbystrata)=c("year","stratum","number")

biomass=merge(strata,weigthbystrata,by=c("year","stratum"))

biomass$kg_km2=((biomass$weigth/biomass$sweptarea)/1000)

density=merge(strata,numberbystrata,by=c("year","stratum"))

density$n_km2=((density$number/density$sweptarea))


# Area of each stratum####

area.shelf1=aggregate(stratum$AREASTRATA,by=list(stratum$CODESTRATA=="A"|stratum$CODESTRATA=="B"|stratum$CODESTRATA=="C"),FUN=sum)

area.str.A=aggregate(stratum$AREASTRATA,by=list(stratum$CODESTRATA=="A"),FUN=sum)

area.str.A1=area.str.A[2,2]##area strato A

area.str.B=aggregate(stratum$AREASTRATA,by=list(stratum$CODESTRATA=="B"),FUN=sum)

area.str.B1=area.str.B[2,2]##area strato B

area.str.C=aggregate(stratum$AREASTRATA,by=list(stratum$CODESTRATA=="C"),FUN=sum)

area.str.C1=area.str.C[2,2]##area strato C

area.str.D=aggregate(stratum$AREASTRATA,by=list(stratum$CODESTRATA=="D"),FUN=sum)

area.str.D1=area.str.D[2,2]##area strato D

area.str.E=aggregate(stratum$AREASTRATA,by=list(stratum$CODESTRATA=="E"),FUN=sum)

area.str.E1=area.str.E[2,2]##area strato E

area.shelf=area.shelf1[2,2]

area.slope=area.shelf1[1,2]

area.tot=area.shelf+area.slope
```

```
# Weigth coefficients by stratum####
W1sh=area.str.A1/area.shelf
W2sh=area.str.B1/area.shelf
W3sh=area.str.C1/area.shelf
W4sl=area.str.D1/area.slope
W5sl=area.str.E1/area.slope
W1tot=area.str.A1/area.tot
W2tot=area.str.B1/area.tot
W3tot=area.str.C1/area.tot
W4tot=area.str.D1/area.tot
W5tot=area.str.E1/area.tot

# Indexes by macrostrata (shelf, slope and total)####
# BIOMASS ####
#SHELF#
BIA=subset(biomass,stratum=="A")
BIA$W1sh=rep(W1sh,nrow(BIA))
BIA$biomA=BIA$kg_km2*BIA$W1sh
BIB=subset(biomass,stratum=="B")
BIB$W2sh=rep(W2sh,nrow(BIB))
BIB$biomB=BIB$kg_km2*BIB$W2sh
BIC=subset(biomass,stratum=="C")
BIC$W3sh=rep(W3sh,nrow(BIC))
BIC$biomC=BIC$kg_km2*BIC$W3sh
BIshelf=merge(BIA,BIB,by="year",all=T)
BIshelf=merge(BIshelf,BIC,by="year",all=T)
BIshelf$shelf_biomass <- rowSums(BIshelf[c(7,13,19)], na.rm=TRUE)
#SLOPE#
BID=subset(biomass,stratum=="D")
BID$W4sl=rep(W4sl,nrow(BID))
BID$biomD=BID$kg_km2*BID$W4sl
BIE=subset(biomass,stratum=="E")
BIE$W5sl=rep(W5sl,nrow(BIE))
BIE$biomE=BIE$kg_km2*BIE$W5sl
BIslope=merge(BID,BIE,by="year",all=T)
BIslope$slope_biomass <- rowSums(BIslope[c(7,13)], na.rm=TRUE)
```

```r
#Total#
BIA=subset(biomass,stratum=="A")
BIA$W1tot=rep(W1tot,nrow(BIA))
BIA$biomA=BIA$kg_km2*BIA$W1tot
BIB=subset(biomass,stratum=="B")
BIB$W2tot=rep(W2tot,nrow(BIB))
BIB$biomB=BIB$kg_km2*BIB$W2tot
BIC=subset(biomass,stratum=="C")
BIC$W3tot=rep(W3tot,nrow(BIC))
BIC$biomC=BIC$kg_km2*BIC$W3tot
BID=subset(biomass,stratum=="D")
BID$W4tot=rep(W4tot,nrow(BID))
BID$biomD=BID$kg_km2*BID$W4tot
BIE=subset(biomass,stratum=="E")
BIE$W5tot=rep(W5tot,nrow(BIE))
BIE$biomE=BIE$kg_km2*BIE$W5tot
BITot=merge(BIA,BIB,by="year",all=T)
BITot=merge(BITot,BIC,by="year",all=T)
BITot=merge(BITot,BID,by="year",all=T)
BITot=merge(BITot,BIE,by="year",all=T)
names(BITot)
BITot$total_biomass <- rowSums(BITot[c(7,13,19,25,31)], na.rm=TRUE)
## Final#
BIOMASS=merge(BIshelf,BIslope,by="year",all=T)
BIOMASS=merge(BIOMASS,BITot,by="year",all=T)
BIOMASS=BIOMASS[,-c(2:19,21:32,34:63)]
# BIOMASS[is.na(BIOMASS)]=0
#write.csv(BIOMASS,file=paste0(gen,spec,"_","GSA",gsa,"_",state,"_","BIOMASS.csv"))


#plotting some graphs
BMS=melt(BIOMASS,id="year")
ggplot(BMS,aes(x=year,y=value,color=variable))+
geom_line()+
facet_grid(~variable)+
ggtitle(paste0(gen,spec,"_","GSA",gsa,"_",state)) + xlab("Year") + ylab("Biomass (kg/km2)")+
theme(axis.text.x = element_text(angle=65, vjust=0.6))+
theme(legend.position="none")

# DENSITY ####
#SHELF#
DIA=subset(density,stratum=="A")
DIA$W1sh=rep(W1sh,nrow(DIA))
DIA$biomA=DIA$n_km2*DIA$W1sh
DIB=subset(density,stratum=="B")
DIB$W2sh=rep(W2sh,nrow(DIB))
DIB$biomB=DIB$n_km2*DIB$W2sh
DIC=subset(density,stratum=="C")
DIC$W3sh=rep(W3sh,nrow(DIC))
DIC$biomC=DIC$n_km2*DIC$W3sh
DIshelf=merge(DIA,DIB,by="year",all=T)
DIshelf=merge(DIshelf,DIC,by="year",all=T)
DIshelf$shelf_density <- rowSums(DIshelf[c(7,13,19)], na.rm=TRUE)
#SLOPE#
DID=subset(density,stratum=="D")
DID$W4sl=rep(W4sl,nrow(DID))
DID$biomD=DID$n_km2*DID$W4sl
DIE=subset(density,stratum=="E")
DIE$W5sl=rep(W5sl,nrow(DIE))
DIE$biomE=DIE$n_km2*DIE$W5sl
DIslope=merge(DID,DIE,by="year",all=T)
DIslope$slope_density <- rowSums(DIslope[c(7,13)], na.rm=TRUE)
#Total#
DIA=subset(density,stratum=="A")
DIA$W1tot=rep(W1tot,nrow(DIA))
DIA$biomA=DIA$n_km2*DIA$W1tot
DIB=subset(density,stratum=="B")
DIB$W2tot=rep(W2tot,nrow(DIB))
DIB$biomB=DIB$n_km2*DIB$W2tot
DIC=subset(density,stratum=="C")
DIC$W3tot=rep(W3tot,nrow(DIC))
DIC$biomC=DIC$n_km2*DIC$W3tot
```

```
DID=subset(density,stratum=="D")
DID$W4tot=rep(W4tot,nrow(DID))
DID$biomD=DID$n_km2*DID$W4tot
DIE=subset(density,stratum=="E")
DIE$W5tot=rep(W5tot,nrow(DIE))
DIE$biomE=DIE$n_km2*DIE$W5tot
DITot=merge(DIA,DIB,by="year",all=T)
DITot=merge(DITot,DIC,by="year",all=T)
DITot=merge(DITot,DID,by="year",all=T)
DITot=merge(DITot,DIE,by="year",all=T)
names(DITot)
DITot$total_density <- rowSums(DITot[c(7,13,19,25,31)], na.rm=TRUE)

DENSITY=merge(DIshelf,DIslope,by="year",all=T)
DENSITY=merge(DENSITY,DITot,by="year",all=T)
DENSITY=DENSITY[,-c(2:19,21:32,34:63)]

#plotting some graphs
DMS=melt(DENSITY,id="year")
ggplot(DMS,aes(x=year,y=value,color=variable))+
geom_line()+
facet_grid(~variable)+
ggtitle(paste0(gen,spec,"_","GSA",gsa,"_",state)) + xlab("Year") + ylab("DENSITY
(n/km2)")+
theme(axis.text.x = element_text(angle=65, vjust=0.6))+
theme(legend.position="none")

# VARIABILITY ####
# Hauls by year ####
hauls_table2A=subset(hauls_table,Stratum=="A")
hauls_table2B=subset(hauls_table,Stratum=="B")
hauls_table2C=subset(hauls_table,Stratum=="C")
hauls_table2D=subset(hauls_table,Stratum=="D")
hauls_table2E=subset(hauls_table,Stratum=="E")

# Swept area by stratum ####
```

```
if(cevert$FQ[1]>0){
swA=subset(strata,stratum=="A")
}else{swA=data.frame(year=as.integer(year),stratum="A",sweptarea=as.numeric(0.001))
}
if(cevert$FQ[2]>0){
swB=subset(strata,stratum=="B")
}else{swB=data.frame(year=as.integer(year),stratum="B",sweptarea=as.numeric(0.001))
}
if(cevert$FQ[3]>0){
swC=subset(strata,stratum=="C")
}else{swC=data.frame(year=as.integer(year),stratum="C",sweptarea=as.numeric(0.001))
}
if(cevert$FQ[4]>0){
swD=subset(strata,stratum=="D")
}else{swD=data.frame(year=as.integer(year),stratum="D",sweptarea=as.numeric(0.001))
}
if(cevert$FQ[5]>0){
swE=subset(strata,stratum=="E")
}else{swE=data.frame(year=as.integer(year),stratum="E",sweptarea=as.numeric(0.001))
}

# Total area ####
area.str.A<- sum(stratum[stratum$CODESTRATA %in% "A",7])
area.str.B<- sum(stratum[stratum$CODESTRATA %in% "B",7])
area.str.C<- sum(stratum[stratum$CODESTRATA %in% "C",7])
area.str.D<- sum(stratum[stratum$CODESTRATA %in% "D",7])
area.str.E<- sum(stratum[stratum$CODESTRATA %in% "E",7])

# Check GSA surface ####
tot_shelf=sum(stratum[stratum$CODESTRATA %in% c("A","B","C"),7])
tot_slope=sum(stratum[stratum$CODESTRATA %in% c("D","E"),7])
tot_area=sum(stratum[stratum$CODESTRATA %in% c("A","B","C","D","E"),7])
```

```
# Correction factor f ####
swA$f=swA$sweptarea/as.integer(area.str.A)
swB$f=swB$sweptarea/as.integer(area.str.B)
swC$f=swC$sweptarea/as.integer(area.str.C)
swD$f=swD$sweptarea/as.integer(area.str.D)
swE$f=swE$sweptarea/as.integer(area.str.E)
SW=rbind(swA,swB,swC,swD,swE)
SW$id3=paste0(SW$year,SW$stratum,sep="")

# Elaboration ####
TATBn$id3=paste0(TATBn$year.x,TATBn$strata,sep="")
noTAwithTB <- TATBn[is.na(TATBn$strata),]
write.csv(noTAwithTB,file="TBhaul_no_in_TAhaul.csv",row.names = FALSE)
TATBn <- TATBn[complete.cases(TATBn[ , 12]),]
biomass$id3=paste0(biomass$year,biomass$stratum,sep="")
density$id3=paste0(density$year,density$stratum,sep="")
hauls_table2$id3=paste0(hauls_table2$Year,hauls_table2$Stratum,sep="")
TATBBI=merge(TATBn,hauls_table2,by="id3",all=TRUE)
TATBBI=merge(TATBBI,biomass,by="id3",all.y=TRUE)
TATBDI=merge(TATBn,hauls_table2,by="id3",all=TRUE)
TATBDI=merge(TATBDI,density,by="id3",all.y=TRUE)

# BIOMASS ####
TATBBI$var1=TATBBI$sqkm*(TATBBI$W_sqkm-TATBBI$kg_km2)^2
var1=aggregate(TATBBI$var1,list(TATBBI$year,TATBBI$strata),sum,na.rm=TRUE)
var1$id3=paste0(var1$Group.1,var1$Group.2,sep="")

var2a=merge(var1,hauls_table2,by="id3",all=T)
var2a$variance=(1/(var2a$Hauls-1))*var2a$x
var2a$devst=sqrt(var2a$variance)
var2a=merge(var2a,SW,by="id3",all=T)

var2a$Stratum <- var2a$stratum
var2a$Group.2 <- var2a$stratum
var2a$Group.1 <- var2a$year
```

```
var2a$Year <- var2a$year
var2a[is.na(var2a)] <- 0

VarIA=subset(var2a,var2a$Stratum=="A")
VarIB=subset(var2a,var2a$Stratum=="B")
VarIC=subset(var2a,var2a$Stratum=="C")
VarID=subset(var2a,var2a$Stratum=="D")
VarIE=subset(var2a,var2a$Stratum=="E")

VarIA$W1sh=W1sh
VarIB$W2sh=W2sh
VarIC$W3sh=W3sh
VarID$W4sl=W4sl
VarIE$W5sl=W5sl
VarIA$W1tot=W1tot
VarIB$W2tot=W2tot
VarIC$W3tot=W3tot
VarID$W4tot=W4tot
VarIE$W5tot=W5tot

VarIA$Vsh=(VarIA$variance*VarIA$W1sh^2)/(VarIA$sweptarea*(1-VarIA$f))
VarIB$Vsh=(VarIB$variance*VarIB$W2sh^2)/(VarIB$sweptarea*(1-VarIB$f))
VarIC$Vsh=(VarIC$variance*VarIC$W3sh^2)/(VarIC$sweptarea*(1-VarIC$f))
VarID$Vsl=(VarID$variance*VarID$W4sl^2)/(VarID$sweptarea*(1-VarID$f))
VarIE$Vsl=(VarIE$variance*VarIE$W5sl^2)/(VarIE$sweptarea*(1-VarIE$f))

VarIA$Vt=(VarIA$variance*VarIA$W1tot^2)/(VarIA$sweptarea*(1-VarIA$f))
VarIB$Vt=(VarIB$variance*VarIB$W2tot^2)/(VarIB$sweptarea*(1-VarIB$f))
VarIC$Vt=(VarIC$variance*VarIC$W3tot^2)/(VarIC$sweptarea*(1-VarIC$f))
VarID$Vt=(VarID$variance*VarID$W4tot^2)/(VarID$sweptarea*(1-VarID$f))
VarIE$Vt=(VarIE$variance*VarIE$W5tot^2)/(VarIE$sweptarea*(1-VarIE$f))

names(VarIA)
VarIA2=VarIA[,c(10,16)]
VarIB2=VarIB[,c(10,16)]
VarIC2=VarIC[,c(10,16)]
```

```
VarID2=VarID[,c(10,16)]
VarIE2=VarIE[,c(10,16)]

prova=rbind(VarIA2,VarIB2,VarIC2)
prova1=aggregate(prova$V,list(prova$year),sum,na.rm=T)
prova1$devst=sqrt(prova1$x)
names(prova1)=c("year","var_shelf","stdev_shelf")
prova2=rbind(VarID2,VarIE2)
prova3=aggregate(prova2$V,list(prova2$year),sum,na.rm=T)
prova3$devst=sqrt(prova3$x)
names(prova3)=c("year","var_slope","stdev_slope")

VarIA3=VarIA[,c(10,17)]
VarIB3=VarIB[,c(10,17)]
VarIC3=VarIC[,c(10,17)]
VarID3=VarID[,c(10,17)]
VarIE3=VarIE[,c(10,17)]

prova4=rbind(VarIA3,VarIB3,VarIC3,VarID3,VarIE3)
prova5=aggregate(prova4$V,list(prova4$year),sum,na.rm=T)
prova5$devst=sqrt(prova5$x)
names(prova5)=c("year","var_tot","stdev_tot")

BIOMASS=merge(BIOMASS,prova1,by="year",all=T)
BIOMASS=merge(BIOMASS,prova3,by="year",all=T)
BIOMASS=merge(BIOMASS,prova5,by="year",all=T)
BIOMASS[is.na(BIOMASS)] <- 0
yr <- seq(min(year),max(year),1)
BIOMASS <- right_join(BIOMASS,setnames(as.data.frame(yr),"year"))
BIOMASS[is.na(BIOMASS)] <- 0
write.csv(BIOMASS,file=paste0(gen,spec,"_","GSA",gsa,"_",state,"_","BIOMASS.csv"))

ggplot(BIOMASS, aes(x=year, y=total_biomass)) +
geom_point(colour="red",cex=3)+
geom_line(linetype = "solid",size=1.25,colour="blue") +
geom_errorbar(aes(ymin=total_biomass-stdev_tot,  ymax=total_biomass+stdev_tot), width=.2,colour="red",
position=position_dodge(0.05),linetype = "dashed",size=0.75)+
ylab("kg/km2")+
ggtitle(paste0(gen,spec,"_","GSA",gsa,"_",state,"Total_biomass"))

ggsave(filename=paste(state,"_","GSA_",gsa,"Total_biomass.jpeg"), plot=ggplot(BIOMASS, aes(x=year, y=total_biomass)) +
geom_line(linetype = "solid",size=1.25,col="blue") +
geom_point(colour="red",cex=3)+
geom_errorbar(aes(ymin=total_biomass-stdev_tot,  ymax=total_biomass+stdev_tot), width=.2,colour="red",
position=position_dodge(0.05),linetype = "dashed",size=0.75)+
ylab("kg/km2")+
ggtitle(paste0(gen,spec,"_","GSA",gsa,"_",state,"Total_biomass")))

# DENSITY #####
TATBDI$var1=TATBDI$sqkm*(TATBDI$N_sqkm-TATBDI$n_km2)^2
var1a=aggregate(TATBDI$var1,list(TATBDI$year,TATBDI$strata),sum,na.rm=TRUE)
var1a$id3=paste0(var1a$Group.1,var1a$Group.2,sep="")

var2aa=merge(var1a,hauls_table2,by="id3",all=T)
var2aa$variance=(1/(var2aa$Hauls-1))*var2aa$x
var2aa$devst=sqrt(var2aa$variance)
var2aa=merge(var2aa,SW,by="id3",all=T)

var2aa$Stratum <- var2aa$stratum
var2aa$Group.2 <- var2aa$stratum
var2aa$Group.1 <- var2aa$year
var2aa$Year <- var2aa$year
var2aa[is.na(var2aa)] <- 0

VarIAa=subset(var2aa,var2aa$Stratum=="A")
VarIBa=subset(var2aa,var2aa$Stratum=="B")
VarICa=subset(var2aa,var2aa$Stratum=="C")
VarIDa=subset(var2aa,var2aa$Stratum=="D")
VarIEa=subset(var2aa,var2aa$Stratum=="E")
```

```
VarIAa$W1sh=W1sh
VarIBa$W2sh=W2sh
VarICa$W3sh=W3sh
VarIDa$W4sl=W4sl
VarIEa$W5sl=W5sl
VarIAa$W1tot=W1tot
VarIBa$W2tot=W2tot
VarICa$W3tot=W3tot
VarIDa$W4tot=W4tot
VarIEa$W5tot=W5tot

VarIAa$Vsh=(VarIAa$variance*VarIAa$W1sh^2)/(VarIAa$sweptarea*(1-VarIAa$f))
VarIBa$Vsh=(VarIBa$variance*VarIBa$W2sh^2)/(VarIBa$sweptarea*(1-VarIBa$f))
VarICa$Vsh=(VarICa$variance*VarICa$W3sh^2)/(VarICa$sweptarea*(1-VarICa$f))
VarIDa$Vsl=(VarIDa$variance*VarIDa$W4sl^2)/(VarIDa$sweptarea*(1-VarIDa$f))
VarIEa$Vsl=(VarIEa$variance*VarIEa$W5sl^2)/(VarIEa$sweptarea*(1-VarIEa$f))

VarIAa$Vt=(VarIAa$variance*VarIAa$W1tot^2)/(VarIAa$sweptarea*(1-VarIAa$f))
VarIBa$Vt=(VarIBa$variance*VarIBa$W2tot^2)/(VarIBa$sweptarea*(1-VarIBa$f))
VarICa$Vt=(VarICa$variance*VarICa$W3tot^2)/(VarICa$sweptarea*(1-VarICa$f))
VarIDa$Vt=(VarIDa$variance*VarIDa$W4tot^2)/(VarIDa$sweptarea*(1-VarIDa$f))
VarIEa$Vt=(VarIEa$variance*VarIEa$W5tot^2)/(VarIEa$sweptarea*(1-VarIEa$f))

VarIA2a=VarIAa[,c(10,16)]
VarIB2a=VarIBa[,c(10,16)]
VarIC2a=VarICa[,c(10,16)]
VarID2a=VarIDa[,c(10,16)]
VarIE2a=VarIEa[,c(10,16)]

provaa=rbind(VarIA2a,VarIB2a,VarIC2a)
prova1a=aggregate(provaa$V,list(provaa$year),sum,na.rm=T)
prova1a$devst=sqrt(prova1a$x)
names(prova1a)=c("year","var_shelf","stdev_shelf")
prova2a=rbind(VarID2a,VarIE2a)
prova3a=aggregate(prova2a$V,list(prova2a$year),sum,na.rm=T)
```

```
prova3a$devst=sqrt(prova3a$x)
names(prova3a)=c("year","var_slope","stdev_slope")

VarIA3a=VarIAa[,c(10,17)]
VarIB3a=VarIBa[,c(10,17)]
VarIC3a=VarICa[,c(10,17)]
VarID3a=VarIDa[,c(10,17)]
VarIE3a=VarIEa[,c(10,17)]

prova4a=rbind(VarIA3a,VarIB3a,VarIC3a,VarID3a,VarIE3a)
prova5a=aggregate(prova4a$V,list(prova4a$year),sum,na.rm=T)
prova5a$devst=sqrt(prova5a$x)
names(prova5a)=c("year","var_tot","stdev_tot")

DENSITY=merge(DENSITY,prova1a,by="year",all=T)
DENSITY=merge(DENSITY,prova3a,by="year",all=T)
DENSITY=merge(DENSITY,prova5a,by="year",all=T)
DENSITY[is.na(DENSITY)] <- 0
yr <- seq(min(year),max(year),1)
DENSITY <- right_join(DENSITY,setnames(as.data.frame(yr),"year"))
DENSITY[is.na(DENSITY)] <- 0

write.csv(DENSITY,file=paste0(gen,spec,"_","GSA",gsa,"_",state,"_","DENSITY.csv"))

ggplot(DENSITY, aes(x=year, y=total_density)) +
geom_point(colour="red",cex=3)+
geom_line(linetype = "solid",size=1,colour="blue") +
geom_errorbar(aes(ymin=total_density-stdev_tot,      ymax=total_density+stdev_tot),
width=.2,colour="red",
position=position_dodge(0.05),linetype = "dashed",size=0.75)+
ylab("n/km2")+
ggtitle(paste0(gen,spec,"_","GSA",gsa,"_",state,"Total_density"))

ggsave(filename=paste(state,"_","GSA_",gsa,"Total_density.jpeg"),
plot=ggplot(DENSITY, aes(x=year, y=total_density)) +
geom_line(linetype = "solid",size=1,colour="blue") +
geom_point(colour="red",cex=3)+
geom_errorbar(aes(ymin=total_density-stdev_tot, ymax=total_density+stdev_tot),
width=.2,colour="red",

position=position_dodge(0.05),linetype = "dashed",size=0.75)+

ylab("n/km2")+

ggtitle(paste0(gen,spec,"_","GSA",gsa,"_",state,"Total_density")))


# Mean weigth by year as ratio between biomass and density indexes #######

subTB=as.data.frame(cbind(TBn$year,TBn$haul_number,TBn$ptot,TBn$nbtot))

names(subTB)=c("year","haul","weigth","number")

head(subTB)

MW=subTB$weigth/subTB$number

subTB=cbind(subTB,MW)

par(mar=c(3.1 ,4.1, 2.1, 2.1))


jpeg('boxplot_outliners.jpg')

boxplot(subTB$MW~subTB$year,col="red",main=paste(gen,spec,sep=""),ylab="MW
(g) by haul")

dev.off()


jpeg('boxplot_no_outliners.jpg')

boxplot(subTB$MW~subTB$year,col="green",main=paste(gen,spec,sep=""),ylab="M
W(g) by haul",outline=F)

dev.off()


meanweight <- aggregate(subTB$weigth,by=list(subTB$year),
FUN=sum)/aggregate(subTB$number,by=list(subTB$year), FUN=sum)

Mweight=cbind(sort(unique(subTB$year)),meanweight)

Mweight=Mweight[,-2]

names(Mweight)=c("Year","Mean_weight(g)")

Mweight <- right_join(Mweight,setnames(as.data.frame(yr),"Year"))

write.csv(Mweight,file=paste0(gen,spec,"_GSA_",gsa,"_",state,"_","Mean_weight.csv",
sep=""))

Mean_weight=ggplot(data=Mweight,aes(x=Year,y=`Mean_weight(g)`))+geom_line()
+geom_point()

ggtitle(paste0(gen,spec,"_GSA_",gsa,"_",state))

Mean_weight

ggsave(filename=paste(state,"_","GSA_",gsa,"Mean_weight.jpeg"),
plot=Mean_weight)
```

```
# Standardized LFDs by km2 ####
raise=TCn$pfrac/TCn$pechan
nblonraise=TCn$nblon*raise
TCn=cbind(TCn,raise,nblonraise)

sum(TCn$nblonraise)
sum(TCn$nblon)
id2=paste(TCn$country,TCn$area,TCn$year,TCn$haul_number,sep="")
TCn=cbind(TCn,id2)
TATCn=merge(TAn,TCn,by=c("id2","year","haul_number"),all=T)

# Checking TC raising factor and compare TC and TB total weight and number ####
TCtoCheck <- TCn[which(TCn$raise > maxratiosampling),]
write.table(TCtoCheck,file=paste("TCtoCheck_",state,gsa,gen,spec,".csv"),sep=";",ro
w.names=F)

tempwgB <- TBn%>% group_by(country,area,year,haul_number)%>%
summarize(totwgB=sum(ptot))
tempnbB <- TBn%>% group_by(country,area,year,haul_number)%>%
summarize(totnbB=sum(nbtot))
tempTB <-
merge(tempwgB,tempnbB,by=c("country","area","year","haul_number"),all=T)

tempwgC <- TCn%>% group_by(country,area,year,haul_number)%>%
summarize(totwgC=mean(pfrac))
tempnbC <- TCn%>% group_by(country,area,year,haul_number)%>%
summarize(totnbC=sum(nblonraise))
tempTC <-
merge(tempwgC,tempnbC,by=c("country","area","year","haul_number"),all=T)

TBTCcheck <- merge(tempTB,tempTC,by=c("country", "area", "year",
"haul_number"))
TBTCcheck$wgratio <- TBTCcheck$totwgB/TBTCcheck$totwgC
TBTCcheck$nbratio <- TBTCcheck$totnbB/TBTCcheck$totnbC

TBTCcheck[which(TBTCcheck$wgratio != 1 | TBTCcheck$nbratio != 1),]
write.csv(TBTCcheck[which(TBTCcheck$wgratio != 1 | TBTCcheck$nbratio !=
1),],file=paste0("TBTCtoCheck_",state,gsa,gen,spec,".csv",sep=""),row.names = F)
```

```
# TOTAL ####
TATCn1=TATCn[,c("year","haul_number","strata","sqkm","length_class","nblonraise")]
st=TATCn1$nblonraise/TATCn1$sqkm
TATC1=cbind(TATCn1,st)
write.table(TATCn,file=paste("TATC_",state,gsa,gen,spec,".csv"),sep=";",row.names=F)

lf=data.frame(aggregate(TATC1$nblonraise,list(TATC1$length_class,TATC1$year,TATC1$strata),sum))
names(lf)=c("LC","Year","Stratum","Value")
lf$id3=paste0(lf$Year,lf$Stratum,sep="")
#SW
lfst=merge(lf,SW,by="id3",all=T)
lfst$st=lfst$Value/lfst$sweptarea
# size=sort(unique(TATC1$length_class))

##extract LFDs by strata from the dataframe##
lfA=subset(lfst,lfst$stratum=="A")
lfA$Wtot=W1tot
lfA[is.na(lfA)]=0
lfA$lfd=(lfA$Wtot*lfA$st)
lfB=subset(lfst,lfst$stratum=="B")
lfB$Wtot=W2tot
lfB[is.na(lfB)]=0
lfB$lfd=(lfB$Wtot*lfB$st)
lfC=subset(lfst,lfst$stratum=="C")
lfC$Wtot=W3tot
lfC[is.na(lfC)]=0
lfC$lfd=(lfC$Wtot*lfC$st)
lfD=subset(lfst,lfst$stratum=="D")
lfD$Wtot=W4tot
lfD[is.na(lfD)]=0
lfD$lfd=(lfD$Wtot*lfD$st)
lfE=subset(lfst,lfst$stratum=="E")
lfE$Wtot=W5tot
```

```
lfE[is.na(lfE)]=0

lfE$lfd=(lfE$Wtot*lfE$st)


LFD=rbind(lfA,lfB,lfC,lfD,lfE)

LFD=data.frame(aggregate(LFD$lfd,list(LFD$LC,LFD$year),sum))


names(LFD)=c("Length","Year","Frequency")

LFD=LFD[,c(2,1,3)]

LFD=LFD[!(LFD$Length==0),]


lclasses <- as.data.frame(rep(seq(min(LFD$Length),max(LFD$Length),lfstep),times=max(LFD$Year)-min(LFD$Year)+1))

names(lclasses) <- "Length"

lclasses$Year <- rep(min(LFD$Year):max(LFD$Year),each=length(unique(lclasses$Length)))


tempLFD <- merge(lclasses,LFD,by=c("Length","Year"),all=T)


tempLFD[is.na(tempLFD)] <- 0


tempLFD <- tempLFD[order(tempLFD$Year,tempLFD$Length),]


# LFDJ=subset(LFD,LFD$Length>160) if need to plot subset

ggplot(tempLFD, aes(y=Frequency, x=Length,col=Year))+ geom_bar(stat= "identity")+

facet_wrap(~Year)+

ggtitle(paste(gen,spec,"LFDs_10-800m_GSA",gsa,state))+xlab("Length")+ylab("n/km2")+

theme(legend.position="none")


ggsave(filename=paste(state,"_","GSA_",gsa,"LFDTOT_10-800m.jpeg"),width = 10, height = 8, dpi = 150, units = "in", plot=ggplot(tempLFD, aes(y=Frequency, x=Length,col=Year))+ geom_bar(stat= "identity")+

facet_wrap(~Year)+

ggtitle(paste(gen,spec,"LFDs_10-800m_GSA",gsa,state))+xlab("Length")+ylab("n/km2")+

theme(legend.position="none"))
```

```
jpeg(file=paste0(gen,spec,"_GSA_",gsa,"_",state,"_","WireframeTotal.jpg",sep=""),
width = 350, height = 350)

theseCol=rainbow(length(tempLFD$Length))##rainbow oppure heat.colors oppure
topo.colors

wireframe(Frequency~Year*Length, data =
tempLFD,drape=TRUE,col.regions=theseCol,zlab="n/km2", xlab="year",
ylab="length",scales = list(arrows = FALSE,tick.number = 10,col="black", font= 1,
tck=0.8,rot=90),

colorkey=T, ylim = rev(range(tempLFD$Length)),

screen = list(z = 28, x = -70, y = 3),

main=paste(gen,spec,sep=""))

dev.off()

write.csv(tempLFD,file=paste0("LFDTOT_10-800m.csv",sep=""),row.names = F)


# Mean Length estimation ####

meanlength <-
aggregate(tempLFD$Length*tempLFD$Frequency,by=list(tempLFD$Year),
FUN=sum)/aggregate(tempLFD$Frequency, by=list(tempLFD$Year), FUN=sum)

ML=cbind(unique(tempLFD$Year),meanlength)

ML=ML[,-2]

names(ML)=c("Year","Mean_Length(mm)")

write.csv(ML,file=paste0(gen,spec,"_GSA_",gsa,"_",state,"_","Mean_Length.csv",sep=
""))

Mean_Length=ggplot(data=ML,aes(x=Year,y=`Mean_Length(mm)`))+geom_line()
+geom_point()

ggtitle(paste0(gen,spec,"_GSA_",gsa,"_",state))

Mean_Length

ggsave(filename=paste(state,"_","GSA_",gsa,"Mean_Length.jpeg"),
plot=Mean_Length)


# FEMALE ####

TATCnF=subset(TATCn,TATCn$sex=="F")

droplevels(TATCnF)

TATCn0=TATCnF[,c("year","haul_number","strata","sqkm","length_class","nblonraise"
)]

st=TATCn0$nblonraise/TATCn0$sqkm

TATC0=cbind(TATCn0,st)

lff=data.frame(aggregate(TATC0$nblonraise,list(TATC0$length_class,TATC0$year,TA
TC0$strata),sum))

names(lff)=c("LC","Year","Stratum","Value")
```

```
lff$id3=paste0(lff$Year,lff$Stratum,sep="")
# SW
lfstf=merge(lff,SW,by="id3",all=T)
lfstf$st=lfstf$Value/lfstf$sweptarea

## extract LFDs by strata from the dataframe##
lfAf=subset(lfstf,lfstf$stratum=="A")
lfAf$Wtot=W1tot
lfAf[is.na(lfAf)]=0
lfAf$lfd=(lfAf$Wtot*lfAf$st)
lfBf=subset(lfstf,lfstf$stratum=="B")
lfBf$Wtot=W2tot
lfBf[is.na(lfBf)]=0
lfBf$lfd=(lfBf$Wtot*lfBf$st)
lfCf=subset(lfstf,lfstf$stratum=="C")
lfCf$Wtot=W3tot
lfCf[is.na(lfCf)]=0
lfCf$lfd=(lfCf$Wtot*lfCf$st)
lfDf=subset(lfstf,lfstf$stratum=="D")
lfDf$Wtot=W4tot
lfDf[is.na(lfDf)]=0
lfDf$lfd=(lfDf$Wtot*lfDf$st)
lfEf=subset(lfstf,lfstf$stratum=="E")
lfEf$Wtot=W5tot
lfEf[is.na(lfEf)]=0
lfEf$lfd=(lfEf$Wtot*lfEf$st)

LFDF=rbind(lfAf,lfBf,lfCf,lfDf,lfEf)
LFDF=data.frame(aggregate(LFDF$lfd,list(LFDF$LC,LFDF$year),sum))

names(LFDF)=c("Length","Year","Frequency")
LFDF=LFDF[,c(2,1,3)]
LFDF=LFDF[!(LFDF$Length==0),]

lclassesF <- as.data.frame(rep(seq(min(LFDF$Length),max(LFDF$Length),lfstep),times=max(LFDF$Year)-min(LFDF$Year)+1))
```

```
names(lclassesF) <- "Length"

lclassesF$Year <-
rep(min(LFDF$Year):max(LFDF$Year),each=length(unique(lclassesF$Length)))


tempLFDF <- merge(lclassesF,LFDF,by=c("Length","Year"),all=T)


tempLFDF[is.na(tempLFDF)] <- 0


tempLFDF <- tempLFDF[order(tempLFDF$Year,tempLFDF$Length),]


ggplot(tempLFDF, aes(y=Frequency, x=Length,col=Year))+ geom_bar(stat=
"identity")+

facet_wrap(~Year)+

ggtitle(paste(gen,spec,"FEMALE_LFDs_10-800m_GSA",gsa,state))+xlab("Length")
+ylab("n/km2")+

theme(legend.position="none")


ggsave(filename=paste(state,"_","GSA_",gsa,"FEMALE_LFDs_10-800m.jpeg"),width =
10, height = 8, dpi = 150, units = "in", plot=ggplot(tempLFDF, aes(y=Frequency,
x=Length,col=Year))+ geom_bar(stat= "identity")+

facet_wrap(~Year)+

ggtitle(paste(gen,spec,"FEMALE_LFDs_10-800m_GSA",gsa,state))+xlab("Length")
+ylab("n/km2")+

theme(legend.position="none"))


jpeg(file=paste0(gen,spec,"_GSA_",gsa,"_",state,"_","WireframeFemale.jpg",sep=""),
width = 350, height = 350)

theseCol=rainbow(length(tempLFDF$Length))##rainbow oppure heat.colors oppure
topo.colors

wireframe(Frequency~Year*Length, data =
tempLFDF,drape=TRUE,col.regions=theseCol,zlab="n/km2", xlab="year",
ylab="length",scales = list(arrows = FALSE,tick.number = 10,col="black", font= 1,
tck=0.8,rot=90),

colorkey=T, ylim = rev(range(tempLFDF$Length)),

screen = list(z = 28, x = -70, y = 3),

main=paste(gen,spec,"_FEMALE",sep=""))

dev.off()

write.csv(tempLFDF,file=paste0("LFDFEM_10-800m.csv",sep=""),row.names = F)
```

```
# MALE ####
TATCnM=subset(TATCn,TATCn$sex=="M")
droplevels(TATCnM)
TATCn2=TATCnM[,c("year","haul_number","strata","sqkm","length_class","nblonraise")]
st=TATCn2$nblonraise/TATCn2$sqkm
TATC2=cbind(TATCn2,st)
lfm=data.frame(aggregate(TATC2$nblonraise,list(TATC2$length_class,TATC2$year,TATC2$strata),sum))
names(lfm)=c("LC","Year","Stratum","Value")
lfm$id3=paste0(lfm$Year,lfm$Stratum,sep="")

# SW
lfstm=merge(lfm,SW,by="id3",all=T)
lfstm$st=lfstm$Value/lfstm$sweptarea

##extract LFDs by strata from the dataframe##
lfAm=subset(lfstm,lfstm$stratum=="A")
lfAm$Wtot=W1tot
lfAm[is.na(lfAm)]=0
lfAm$lfd=(lfAm$Wtot*lfAm$st)
lfBm=subset(lfstm,lfstm$stratum=="B")
lfBm$Wtot=W2tot
lfBm[is.na(lfBm)]=0
lfBm$lfd=(lfBm$Wtot*lfBm$st)
lfCm=subset(lfstm,lfstm$stratum=="C")
lfCm$Wtot=W3tot
lfCm[is.na(lfCm)]=0
lfCm$lfd=(lfCm$Wtot*lfCm$st)
lfDm=subset(lfstm,lfstm$stratum=="D")
lfDm$Wtot=W4tot
lfDm[is.na(lfDm)]=0
lfDm$lfd=(lfDm$Wtot*lfDm$st)
lfEm=subset(lfstm,lfstm$stratum=="E")
lfEm$Wtot=W5tot
lfEm[is.na(lfEm)]=0
lfEm$lfd=(lfEm$Wtot*lfEm$st)
```

```
LFDM=rbind(lfAm,lfBm,lfCm,lfDm,lfEm)

LFDM=data.frame(aggregate(LFDM$lfd,list(LFDM$LC,LFDM$year),sum))


names(LFDM)=c("Length","Year","Frequency")

LFDM=LFDM[,c(2,1,3)]

LFDM=LFDM[!(LFDM$Length==0),]


lclassesM                                                                   <-
as.data.frame(rep(seq(min(LFDM$Length),max(LFDM$Length),lfstep),times=max(LFD
M$Year)-min(LFDM$Year)+1))

names(lclassesM) <- "Length"

lclassesM$Year                                                              <-
rep(min(LFDM$Year):max(LFDM$Year),each=length(unique(lclassesM$Length)))


tempLFDM <- merge(lclassesM,LFDM,by=c("Length","Year"),all=T)


tempLFDM[is.na(tempLFDM)] <- 0


tempLFDM <- tempLFDM[order(tempLFDM$Year,tempLFDM$Length),]


ggplot(tempLFDM,    aes(y=Frequency,    x=Length,col=Year))+    geom_bar(stat=
"identity")+

facet_wrap(~Year)+

ggtitle(paste(gen,spec,"MALE_LFDs_10-800m_GSA",gsa,state))+xlab("Length")
+ylab("n/km2")+

theme(legend.position="none")


ggsave(filename=paste(state,"_","GSA_",gsa,"MALE_LFDs_10-800m.jpeg")

,width  =  10,  height  =  8,  dpi  =  150,  units  =  "in",  plot=ggplot(tempLFDM,
aes(y=Frequency, x=Length,col=Year))+ geom_bar(stat= "identity")+

facet_wrap(~Year)+

ggtitle(paste(gen,spec,"MALE_LFDs_10-800m_GSA",gsa,state))+xlab("Length")
+ylab("n/km2")+

theme(legend.position="none"))


jpeg(file=paste0(gen,spec,"_GSA_",gsa,"_",state,"_","WireframeMale.jpg",sep=""),
width = 350, height = 350)
```

```
theseCol=rainbow(length(tempLFDM$Length))##rainbow oppure heat.colors oppure topo.colors

wireframe(Frequency~Year*Length, data = tempLFDM,drape=TRUE,col.regions=theseCol,zlab="n/km2", xlab="year", ylab="length",scales = list(arrows = FALSE,tick.number = 10,col="black", font= 1, tck=0.8,rot=90),

colorkey=T, ylim = rev(range(tempLFDM$Length)),

screen = list(z = 28, x = -70, y = 3),

main=paste(gen,spec,"_MALE",sep=""))

dev.off()

write.csv(tempLFDM,file=paste0("LFDMALE_10-800m.csv",sep=""),row.names = F)


# INDETERMINATE ####

TATCnI=subset(TATCn,TATCn$sex=="I" | TATCn$sex=="N" )

droplevels(TATCnF)

TATCn4=TATCnI[,c("year","haul_number","strata","sqkm","length_class","nblonraise")]

st=TATCn4$nblonraise/TATCn4$sqkm

TATC4=cbind(TATCn4,st)

lfi=data.frame(aggregate(TATC4$nblonraise,list(TATC4$length_class,TATC4$year,TATC4$strata),sum))

names(lfi)=c("LC","Year","Stratum","Value")

lfi$id3=paste0(lfi$Year,lfi$Stratum,sep="")


# SW

lfsti=merge(lfi,SW,by="id3",all=T)

lfsti$st=lfsti$Value/lfsti$sweptarea


##extract LFDs by strata from the dataframe##

lfAi=subset(lfsti,lfsti$stratum=="A")

lfAi$Wtot=W1tot

lfAi[is.na(lfAi)]=0

lfAi$lfd=(lfAi$Wtot*lfAi$st)

lfBi=subset(lfsti,lfsti$stratum=="B")

lfBi$Wtot=W2tot

lfBi[is.na(lfBi)]=0

lfBi$lfd=(lfBi$Wtot*lfBi$st)

lfCi=subset(lfsti,lfsti$stratum=="C")
```

```
lfCi$Wtot=W3tot
lfCi[is.na(lfCi)]=0
lfCi$lfd=(lfCi$Wtot*lfCi$st)
lfDi=subset(lfsti,lfsti$stratum=="D")
lfDi$Wtot=W4tot
lfDi[is.na(lfDi)]=0
lfDi$lfd=(lfDi$Wtot*lfDi$st)
lfEi=subset(lfsti,lfsti$stratum=="E")
lfEi$Wtot=W5tot
lfEi[is.na(lfEi)]=0
lfEi$lfd=(lfEi$Wtot*lfEi$st)

LFDI=rbind(lfAi,lfBi,lfCi,lfDi,lfEi)
LFDI=data.frame(aggregate(LFDI$lfd,list(LFDI$LC,LFDI$year),sum))

names(LFDI)=c("Length","Year","Frequency")
LFDI=LFDI[,c(2,1,3)]
LFDI=LFDI[!(LFDI$Length==0),]

lclassesl <-
as.data.frame(rep(seq(min(LFDI$Length),max(LFDI$Length),lfstep),times=max(LFDI$
Year)-min(LFDI$Year)+1))
names(lclassesl) <- "Length"
lclassesl$Year <-
rep(min(LFDI$Year):max(LFDI$Year),each=length(unique(lclassesl$Length)))

tempLFDI <- merge(lclassesl,LFDI,by=c("Length","Year"),all=T)

tempLFDI[is.na(tempLFDI)] <- 0

tempLFDI <- tempLFDI[order(tempLFDI$Year,tempLFDI$Length),]

ggplot(tempLFDI, aes(y=Frequency, x=Length,col=Year))+ geom_bar(stat=
"identity")+
facet_wrap(~Year)+
ggtitle(paste(gen,spec,"INDETER_LFDs_10-800m_GSA",gsa,state))+
xlab("Length")+ylab("n/km2")+theme(legend.position="none")
```

```
ggsave(filename=paste(state,"_","GSA_",gsa,"INDETER_LFDs_10-800m.jpeg"),width = 10, height = 8, dpi = 150, units = "in", plot=ggplot(tempLFDI, aes(y=Frequency, x=Length,col=Year))+ geom_bar(stat= "identity")+

facet_wrap(~Year)+

ggtitle(paste(gen,spec,"INDETER_LFDs_10-800m_GSA",gsa,state))+

xlab("Length")+ylab("n/km2")+theme(legend.position="none"))


jpeg(file=paste0(gen,spec,"_GSA_",gsa,"_",state,"_","WireframeInd.jpg",sep=""), width = 350, height = 350)

theseCol=rainbow(length(tempLFDI$Length))##rainbow oppure heat.colors oppure topo.colors

wireframe(Frequency~Year*Length, data = tempLFDI,drape=TRUE,col.regions=theseCol,zlab="n/km2", xlab="year", ylab="length",scales = list(arrows = FALSE,tick.number = 10,col="black", font= 1, tck=0.8,rot=90),

colorkey=T, ylim = rev(range(tempLFDI$Length)),

screen = list(z = 28, x = -70, y = 3),

main=paste(gen,spec,"_INDETER",sep=""))

dev.off()

write.csv(tempLFDI,file=paste0("LFDIND_10-800m.csv",sep=""),row.names = F)


# Splitting Indeter by half and adding to LFD by sex ##

LFDFF <- left_join(tempLFDF,tempLFDI,by=c("Length","Year"))

LFDFF[is.na(LFDFF)] <- 0

LFDFF$Frequency <-LFDFF$Frequency.x+LFDFF$Frequency.y/2

LFDFF <- LFDFF[,-c(3,4)]

write.csv(LFDFF,file=paste0("LFDFF_10-800m.csv",sep=""),row.names = F)


LFDMM <- left_join(tempLFDM,tempLFDI,by=c("Length","Year"))

LFDMM[is.na(LFDMM)] <- 0

LFDMM$Frequency <-LFDMM$Frequency.x+LFDMM$Frequency.y/2

LFDMM <- LFDMM[,-c(3,4)]

write.csv(LFDMM,file=paste0("LFDMM_10-800m.csv",sep=""),row.names = F)


tempLFDF$check <-ifelse(tempLFDF$Length*tempLFDF$Frequency>0,1,0)

maxLFF <- as.data.frame(tempLFDF %>%

filter(tempLFDF$check == 1) %>%
group_by(Year) %>%
summarize(maxlengthF=max(Length)))

tempLFDM$check <-ifelse(tempLFDM$Length*tempLFDM$Frequency>0,1,0)
maxLMM <- as.data.frame(tempLFDM %>%
filter(tempLFDM$check == 1) %>%
group_by(Year) %>%
summarize(maxlengthM=max(Length)))

tempLFD$check <-ifelse(tempLFD$Length*tempLFD$Frequency>0,1,0)
maxLTT <- as.data.frame(tempLFD %>%
filter(tempLFD$check == 1) %>%
group_by(Year) %>%
summarize(maxlengthT=max(Length)))

tempLFDI$check <-ifelse(tempLFDI$Length*tempLFDI$Frequency>0,1,0)
maxLII <- as.data.frame(tempLFDI %>%
filter(tempLFDI$check == 1) %>%
group_by(Year) %>%
summarize(maxlengthI=max(Length)))

datalist <- list(maxLTT,maxLMM,maxLFF,maxLII)

library("tidyverse")

maxLength <- setNames(as.data.frame(datalist %>%
reduce(inner_join,by="Year")),c("Year","Total","Male","Female","Indeterminate"))

detach("package:tidyverse")
write.csv(maxLength,file=paste0("maxLength.csv",sep=""),row.names = F)

# RECRUITS ####
# Treshold based on length or maturity???

# setting recruits treshold ####
# len_juv=80 #length in mm
# TATCnJ=subset(TATCn,TATCn$length_class<=len_juv) # if you want use length treshold

TATCnJ=subset(TATCn,TATCn$maturity=="1" | TATCn$maturity=="0")# disactived this line code if you want to use length treshold

TATCnJ=subset(TATCnJ,TATCnJ$sex=="F" | TATCnJ$sex=="I" | TATCnJ$sex=="N")

TATCnj=TATCnJ[,c("year","haul_number","strata","sqkm","length_class","nblonraise")]

st=TATCnj$nblonraise/TATCnj$sqkm

TATCj=cbind(TATCnj,st)

lf=data.frame(aggregate(TATCj$nblonraise,list(TATCj$length_class,TATCj$year,TATCj$strata),sum))

names(lf)=c("LC","Year","Stratum","Value")

lf$id3=paste0(lf$Year,lf$Stratum,sep="")

# SW

lfst=merge(lf,SW,by="id3",all=T)

lfst$st=lfst$Value/lfst$sweptarea

# size=sort(unique(TATC1$length_class))

##extract LFDs by strata from the dataframe##

lfA=subset(lfst,lfst$stratum=="A")

lfA$Wtot=W1tot

lfA[is.na(lfA)]=0

lfA$lfd=(lfA$Wtot*lfA$st)

lfB=subset(lfst,lfst$stratum=="B")

lfB$Wtot=W2tot

lfB[is.na(lfB)]=0

lfB$lfd=(lfB$Wtot*lfB$st)

lfC=subset(lfst,lfst$stratum=="C")

lfC$Wtot=W3tot

lfC[is.na(lfC)]=0

lfC$lfd=(lfC$Wtot*lfC$st)

lfD=subset(lfst,lfst$stratum=="D")

lfD$Wtot=W4tot

lfD[is.na(lfD)]=0

lfD$lfd=(lfD$Wtot*lfD$st)

lfE=subset(lfst,lfst$stratum=="E")

lfE$Wtot=W5tot
```

```
lfE[is.na(lfE)]=0
lfE$lfd=(lfE$Wtot*lfE$st)

LFDj=rbind(lfA,lfB,lfC,lfD,lfE)
LFDj=data.frame(aggregate(LFDj$lfd,list(LFDj$LC,LFDj$year),sum))

names(LFDj)=c("Length","Year","Frequency")
LFDj=LFDj[,c(2,1,3)]
LFDj=LFDj[!(LFDj$Length==0),]

lclassesj <-
as.data.frame(rep(seq(min(LFDj$Length),max(LFDj$Length),lfstep),times=max(LFDj$
Year)-min(LFDj$Year)+1))
names(lclassesj) <- "Length"
lclassesj$Year <-
rep(min(LFDj$Year):max(LFDj$Year),each=length(unique(lclassesj$Length)))

tempLFDj <- merge(lclassesj,LFDj,by=c("Length","Year"),all=T)

tempLFDj[is.na(tempLFDj)] <- 0

tempLFDj <- tempLFDj[order(tempLFDj$Year,tempLFDj$Length),]

ggplot(tempLFDj, aes(y=Frequency, x=Length,col=Year))+ geom_bar(stat=
"identity")+
facet_wrap(~Year)+
ggtitle(paste(gen,spec,"Immature_LFDs_10-800m_GSA",gsa,state))+xlab("Length")
+ylab("n/km2")+
theme(legend.position="none")

ggsave(filename=paste(state,"_","GSA_",gsa,"Immature_LFDs_10-800m.jpeg"),width
= 10, height = 8, dpi = 150, units = "in", plot=ggplot(tempLFDj, aes(y=Frequency,
x=Length,col=Year))+ geom_bar(stat= "identity")+
facet_wrap(~Year)+
ggtitle(paste(gen,spec,"Immature_LFDs_10-800m_GSA",gsa,state))+xlab("Length")
+ylab("n/km2")+
theme(legend.position="none"))
```

```
write.csv(tempLFDj,file=paste0("LFDjuv_10-800m.csv",sep=""),row.names = F)


# Sex_ratio ####

LFDSR=merge(LFDF,LFDM,by=c("Year","Length"),all =T)

LFDSR[is.na(LFDSR)] <- 0

LFDSR$SR=LFDSR$Frequency.x/(LFDSR$Frequency.x+LFDSR$Frequency.y)

ggplot(LFDSR, aes(y=SR, x=Length,col=Year))+ geom_line()+

facet_wrap(~Year)+

ggtitle(paste(gen,spec,"SexRatio_10-800m_GSA",gsa,state))+xlab("Length")
+ylab("SR(F/F+M)")+

theme(legend.position="none")+geom_hline(yintercept = 0.50,col="red")


ggsave(filename=paste(state,"_","GSA_",gsa,"SR_10-800m.jpeg"),width = 10, height
= 8, dpi = 150, units = "in", plot=ggplot(LFDSR, aes(y=SR, x=Length,col=Year))+
geom_line()+

facet_wrap(~Year)+

ggtitle(paste(gen,spec,"SexRatio_10-800m_GSA",gsa,state))+xlab("Length")
+ylab("SR(F/F+M)")+

theme(legend.position="none"))


write.csv(LFDSR,file=paste0(gen,spec,"_GSA_",gsa,"_",state,"_","SexRatio_10-
800m.csv",sep=""),row.names = F)


setwd(WD)

# ######### END OF SCRIPT ###############
```

References

COCHRAN W.G. (1977). Sampling techniques (3rd edition). John Wiley & Sons, New York, USA, 428 pp.

FOGARTY M.J. 1985. Statistical considerations in the design of the trawl-survey. FAO Fish. Circ. 786, 1-21.

SOUPLET A. (1996) - Définition des estimateurs. In: Campagne internationale de chalutage démersal en Méditerranée (Medits 95). Vol. III. Indices de biomasse et distributions en tailles. Bertrand J. Coordonnateur général. Etude 94/047 IFREMER/CE, 94/011 IEO/CE, 94/057 SIBM/CE, 94/051 NCMR/CE.

List of figures

GETTING IN TOUCH WITH THE EU

In person

All over the European Union there are hundreds of Europe Direct information centres. You can find the address of the centre nearest you at: https://europa.eu/european-union/contact_en

On the phone or by email

Europe Direct is a service that answers your questions about the European Union. You can contact this service:

- by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),

- at the following standard number: +32 22999696, or

- by electronic mail via: https://europa.eu/european-union/contact_en

FINDING INFORMATION ABOUT THE EU

Online

Information about the European Union in all the official languages of the EU is available on the Europa website at: https://europa.eu/european-union/index_en

EU publications

You can download or order free and priced EU publications from EU Bookshop at: https://publications.europa.eu/en/publications. Multiple copies of free publications may be obtained by contacting Europe Direct or your local information centre (see https://europa.eu/european-union/contact_en).

KJ-03-20-048-EN-N

The European Commission's science and knowledge service

Joint Research Centre

JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.

EU Science Hub
ec.europa.eu/jrc

@EU_ScienceHub

EU Science Hub - Joint Research Centre

EU Science, Research and Innovation

EU Science Hub

Publications Office
of the European Union

doi:10.2760/5799

ISBN 978-92-76-10598-5