Call MARE/2020/08

Agreement reference: SI2.839444
European Maritime and Fisheries Fund (EMFF)

Development of the Regional Database for the Mediterranean & Black Seas

Work-package 6 - Deliverable 6.1

The entire scope of product,
production testing,
user manuals

Stefanos Kavadas, Maria Teresa Spedicato

Partners involved:

HCMR, COISPA, CIBM, CNR, IFREMER, NISEA

Table of Contents

Acronyms

DCF	Data Collection Framework
DCRF	Data Collection Reference Framework
EC	European Comission
ERD	Entity–Relationship Diagram
EU	European Union
FDI	Fisheries Dependent Information
GFCM	General Fisheries Commission for the Mediterranean
GIS	Geographic Information System
GSA	Geographical Sub-Area
ICES	International Council for the Exploration of the Sea
MCDA	Multi Criteria Decision Analysis
Med&BS	Mediterranean and Black Sea
MEDIAS	Mediterranean Acoustic Survey
MEDITS	MEDIterranean Trawl Survey
MS	Member State
PETS	Protected, endangered and threatened species
PLSQL	Procedural Language extensions to the Structured Query Language
RCG	Regional Coordination Group
RDB	Regional Data Base
RDBES	Regional Data Base - Estimation System
RDBFIS	Regional Data Base - Fisheries Information System
SC	Steering Committee
STECF	Scientific, Technical and Economic Committee for Fisheries

Executive summary

In the Production Phase, we mainly focused on the implementation of project and optimization including the reduced cost and risk factors of RDBFIS. It also involved various testing for efficient deployment of the project.

In the first phase, a relational DB scheme was designed based on the requirements of the reporting obligations at the EU and international levels, taking into consideration existing scheme format (Phase 1).

In the second phase were implemented: GUI design, Security subsystem, script development and improvements, data validation/error checking, data processing-estimates, reporting, data import/submission, data queries/export.

Data were requested from DGMARE and GFCM (as submitted by MSs for the datacalls obligations) in order to update the RDB, test the functionality of the application in the input, quality control and output processes as well as to be used for the Workshops (testing phase, WP5).

RDBFIS performance and ability to meet the end users' needs have been tested at a round of communication with the Steering Committee, Member States' representatives and other relevant stakeholders.

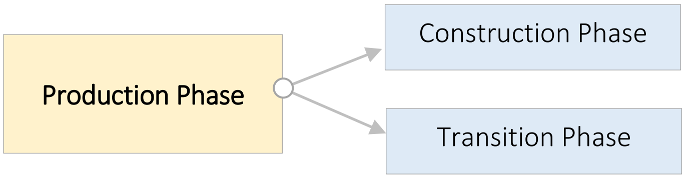
Familiarizing the end users with the application was realized in the course of the 'end-users training' tasks. Two workshops had brought together developers and users.

A manual of the system functionalities is an integral part of the deliverable as well as the Service Level Agreement.

Supplementary spatial tools (standalone version) were developed to support the spatial distribution of fishing effort for small-scale fisheries and the analysis of vessel monitoring system data.

1. Introduction

In the Production Phase, we mainly focused on the implementation of project and optimization including the reduced cost and risk factors of RDBFIS. It also involved various testing for efficient deployment of the project. It was broadly divided into 2 Phases: Construction Phase and Transition Phase.



i. **Construction Phase** – In the construction phase, we performed the implementation and testing of RDBFIS. In this phase, the RDB was constructed as well as GUI, security system, data quality checks, processing, reporting, data mining and spatial components were integrated into RDBFIS. Strict testing and process optimization were performed.

ii. **Transition Phase** – In the Transition phase, we performed strict tests, mainly beta testing and deployment of software. After receiving the feedback from the users, the software was improved (R routines, GUI, integration) making the RDBFIS more supportable and user friendly.

Besides user friendliness, the application is in a position to conduct in real-time a series of quality-checking and error handling routines, accompanied with a relevant report (error log). The RDBFIS environment focused on being as interactive and self-explanatory as possible. The goal is to keep RDBFIS system useful and productive after this has been deployed to the end users focused on the key principle “keep the system running and help users to use it”. To this end, an operational staff will run, monitor and improve the system so that it meets and supports the specific needs of DCF.

2. Production Phase 1 – Design and feedback

The first step was to conduct a detailed System Analysis of the DB. A relational DB scheme was designed based on the requirements of the reporting obligations at the EU and international levels, taking into consideration existing scheme format. This involved the definition of data tables with corresponding fields and possible range of values, as well as the hierarchical structure (parent-child) linking them. The schema has been provided to the Steering Committee, Member States’ representatives and other relevant stakeholders for feedback. Identified comments, suggestions and gaps were taken into account.

A detailed description of the RDB structure and relationships is given in **ANNEX-IV** “Entity–Relationship Diagram (ERD), tables and relationships”.

3. Production Phase 2 – Intermediate

Based on the first round of communication with the users, and having finalized the DB schema, the next development steps to build RDBFIS are given below:

1. GUI design (Task4.2)
2. Security subsystem (Task4.2)
3. Script development & improvements (Tasks 4.2, 4.3, 4.4, 4.5, Spatial tools)
4. Data validation/error checking (Task 4.3)

5. Data processing, estimates (Task4.4)
6. Reporting (Task4.5)
7. Data import/Submission (Tasks 4.2, 4.3, 4.4, 4.5)
8. Data queries/export (Task 4.2)

The R packages (for quality checks, processing and reporting – see **D4.3.1**, **D4.4.1**, **D4.5.1**) as well as additional scripts supporting spatial tools, have been integrated into RDBFIS (see **D4.2.1**).

Specific effort was spent to ensure that each stage in the software development process, whether its design, development, testing, or release, was completed successfully.

Data were requested from DGMARE and GFCM (as submitted by MSs for the datacalls obligations) in order to update the RDB, test the functionality of the application in the input, quality control and output processes as well as to be used for the Workshops (testing phase, WP5) (see APPENDIX III – data request)

RDBFIS performance and ability to meet the end users' needs have been tested at a round of communication with the Steering Committee, Member States' representatives and other relevant stakeholders.

Familiarizing the end users with the application was realized in the course of the 'end-users training' tasks. Two workshops had brought together developers and users. Besides educating people on the application, this also had served as an ideal opportunity to test system performance with real data at nearly real-world situations. During this "pilot data submission/query testing" all member states and MARE had called to participate (**D5.1**).

4. Production Phase 3 – Final

A fully functional system to store, query, estimate, extract and report is delivered at this phase. The product is free of bugs and errors and meets the needs of the end users as explicitly stated in the Service Level Agreement (see APPENDIX I - Service Level Agreement).

A manual of the system functionalities is an integral part of the deliverable. It is available both in a downloadable electronic format document, as well as an on-line capability of the user interface (**ANNEX XVIII (RDBFIS UserGuide)**).

5. Spatial information module

GIS functionalities that render interactive maps from map tiles and vector data are integrated in RDBFIS. From a technological point of view, its integration with the RDBFIS system has been fully implemented. To date, limited information has been added and can be visualized, in the future it is planned to be supplemented with spatial information coming from DCF actions, the European fishing fleet and FDI datacall.

The Spatial Information module is constructed with the Opelnayers javascript library (https://openlayers.org) which is combined with the Angular framework in the frontend. The layers are served by the Geoserver (https://geoserver.org) which is installed on the application server. On the Spatial Information part of the application the user can view information on the species occurrence per GFCM GSA, ports location, fishing grounds, fishing effort, landings and MEDITS stations (see **ANNEX XVIII (RDBFIS UserGuide)**).

6. Supplementary spatial tools (standalone version)

Spatial distribution of fishing effort for small-scale fisheries

Professional small-scale fisheries (SSF) comprise a high amount of the Mediterranean fishing fleet. However, the majority of vessels has total length < 12 m and are not equipped (the vessels) with monitoring devices. A methodological approach aiming to estimate the spatial distribution of fishing effort was developed to fill this gap and provide SSF spatial effort information to FDI datacall (Table I. Effort by rectangle). The `fprmcda` R package provides a tool for estimating and visualizing fishing pressure for small and medium-scale coastal fisheries. `fprmcda` combines the multi-criteria decision analysis (MCDA) and geospatial techniques to quantify the synergistic effect of influential components (e.g., bathymetry, distance from coast, Chl-a, marine traffic activity, vessel capacity of ports, etc.) on defining fishing footprint. (Kavadas et al., 2015 on *https://ejournals.epublishing.ekt.gr/index.php/hcmr-med-mar-sc/article/viewFile/12919/12483.pdf*).

The `fprmcda` R package can be downloaded from *https://cloudfs.hcmr.gr/index.php/s/MuUCkionc5sIuNP*

A brief description of the MCDA methodological approach and `fprmcda` can be found in APPENDIX-II SSF fishing effort estimation based on Multi-Criteria Decision Analysis-MCDA.

Analysis of Vessel Monitoring System data

Routines written in ORACLE PLSQL are developed to analyze Vessel Monitoring System (VMS) data. Due to the huge number of records to be used during the analysis process and the complexity of the operations performed (to estimate fishing effort), it was decided not to integrate these routines into RDBFIS. Otherwise, there would be a risk that a large part of the system's resources would be tied up, thus reducing its performance. So routines can be executed locally, and then the result sent to system administrator to store on the centralize system. A brief description of the method to analyze VMS data as well as the PLSQL code is given in APPENDIX-II Analysis of Vessel Monitoring System data.

Due to the complexity of performing SSF fishing effort and VMS analysis, workshops are planned to be organize within the first semester of the RDBFIS-II execution.

APPENDIX I - Service Level Agreement

Service Level Agreement
between
the Hellenic Centre for Marine Research
and DGMARE

1. Definitions

Service Level Agreement (SLA): the agreement between the supplier (the Hellenic Centre of Marine Research – HCMR on behalf of Med&BS RDBFIS consortium) and DGMARE, comprising terms, conditions and technical appendices.

2. Duration

The service duration inside the framework of the Med&BS RDBFIS project (contract number: SI2.839444) is defined to a period of approximately 24 months, starting on 01/04/2023 and ending on 31/03/2025

3. Technical description of the service

RDBFIS will contain detailed biological data of demersal and small pelagic species and aggregated transversal data (i.e. landings and effort). Included are also a number of other areas (such as bycatch, PETS, recreational fishery, alien species, environmental data etc.)and a series of advanced functionalities. RDBFIS will service thereporting specific to the data calls of the main end users of the region (eg. DG MARE, GFCM, RCG Med&BS). A suite of data validation and quality checkswill be ready to be applied both on detailed and aggregated data. Additional user-oriented tools will support uploading and downloading of data, exchange format(s), data processing tools for statistical analysis and automatic reporting linked to DCF processes. Compatibility with RDBESis also considered and depends on the outcomes of ongoing discussions at the RCG Med & BS level;

Regarding governance, RDBFIS has already provided a data policy document and a proposal for guidelines in the Med&BS RDB Steering Committeeto the National Correspondents. These draft documents have been discussed in relevant fora and agreement is expected at the RCG Med&BS September meeting;

A security system has been developed and integrated into the RDBFIS, ensuring data protection at MS level. In line with the DCF Regulation, detailed and aggregated data hosted in the RDB will be accessible only by the National Correspondents, who will be able (and responsible) to import, edit and export data (in data call format, tabulation form, graphical representation or maps). Through the RDBFIS application, public data (such as EU fleet register, fishing ports, ITIS species list) can be accessible by different stakeholders: COM, MS of the RCG Med & BS, the main end users of the region (STECF, GFCM, ICCAT). The RCG Med & BS will have access to the RDB and will be able to use the data. Moreover, a broad range of information security controls to protect databases against compromises of their confidentiality, integrity and availability have been included in the RDBFIS. These include technical, administrative and physical types of controls.

The launching and maintenance of the RDB Med&BS, but also its fine-tuning and further improvement are the key points of the service. This includes solving any issues that may arise during the first phase of implementation and any adjustments and modifications deemed necessary for the seamless functioning of the RDB Med&BS. Modifications needed as a result of changes in the structure and format of the main recurrent data calls and reporting obligations could also be tackled during this time.

4. Commitments from the supplier

- Real-time monitoring and notification of RDBFIS services health and availability
- Weekly rootkit and vulnerability scan of the server environment
- Provision and support of hosting environment and server
- Weekly backups (data, configurations, source code)
- Monthly updates for server environment
- Mail support as requested / required.
- Semi-annual availability reports

5. Service Response Times

Ticket	Definition	Maximum Response Time (Expected Response Time)	Measurement Window for Response Time	Ticket Submission Method
High Priority	Issue renders the Application unusable andhave noworkaround, or cause loss/corruption ofstored data. (Lost userinput, e.g. a failed formsubmission, is notthesame thing as datalossand in most cases isMedium).	4 hours (2 hour)	Business Hours (Mon-Fri, 9:00am MST – 5:00pm MST, not including statutory holidays)	Email to support email or as defined in the communication chapter
Medium Priority	Not High Priority, butdohave significantimpact orare important as definedbyClient.	8 hours (4 hours)	Business Hours (Mon-Fri, 9:00am MST – 5:00pm MST, not including statutory holidays)	Email to support email or as defined in the communication chapter
Low Priority	Have isolated impactandmay haveworkarounds.Most issues areconsideredlow.	12 hours (6 hours)	Business Hours (Mon-Fri, 9:00am MST – 5:00pm MST, not including statutory holidays)	Email to support email or as defined in the communication chapter

6. Uptime guarantee

Subject to the exclusions below, service availability must be at least 99.2%.

Service Provider shall have no liability under this SLA with respect to any failure orservice availability deficiency caused by or in conjunction with:

- Circumstances beyond Service Provider's reasonable control, including, without limitation, acts of any governmental body, war, terrorism, insurrection, sabotage, armed conflict, embargo, fire, flood, strike, weather, earthquake, or other labor disturbance, interruption of or delay in transportation, unavailability of or interruption or delay in telecommunications or third party services, virus attacks or hackers, failure of third party software (including, without limitation, eCommerce software, payment gateways, chat, statistics or free scripts) or inability to obtain raw materials, supplies, or power used in or equipment needed for provision of this SLA;
- Scheduled maintenance and emergency maintenance and upgrades;
- DNS (Domain Name System) issues outside the direct control of Service Provider;
- Erroneous SLA outages reported as a result of outages or errors of any Service Provider measurement system;
- E-mail or webmail delivery and transmission;
- DNS propagation;
- Outages elsewhere on the Internet that hinder access to your account or services. Service Provider is not responsible for browser or DNS caching that may make your Application appears inaccessible when others can still access it. Service Provider will guarantee only those areas considered under the control of Service Provider.

7. Communication

The notifications shall be in written form or e-mail and sent to:
For the supplier side: stefanos@hcmr.gr
For the user side: MARE-2020-08@ec.europa.eu

STATEMENT

We, the undersigned, certify that the information indicated in the present agreement is correct, appraise the general terms and conditions, and acknowledge the technical description of the service.

Date : ___/___/___ **DGMARE**	Date : ___/___/___ **Hellenic Centre for Marine Research**

APPENDIX II - Supplementary spatial tools

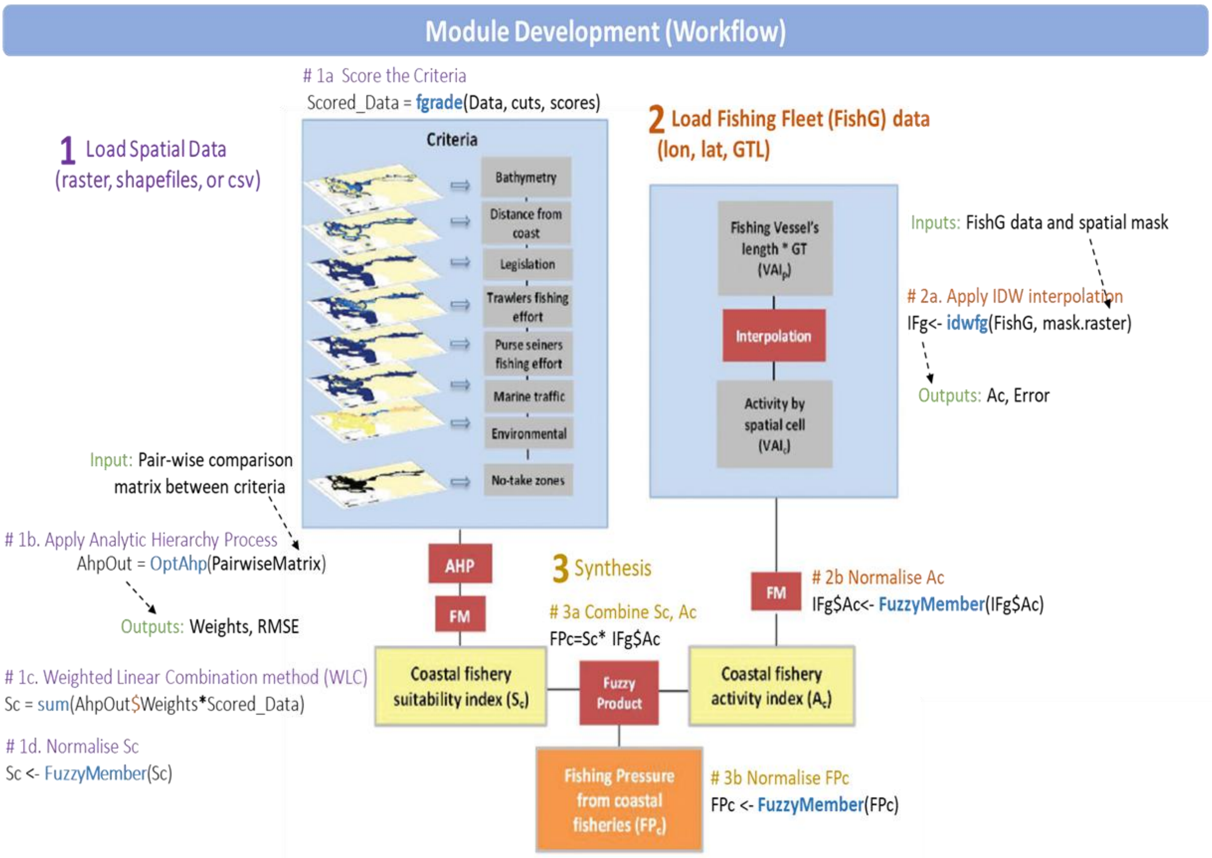
SSF fishing effort estimation based on Multi-Criteria Decision Analysis- MCDA

The approach is based on a Multi-Criteria Decision Analysis-MCDA, while further details for the spatial analytical methods used in the MCDA are provided in Kavadas et al. 2015. The MCDA produces a potential fishing pressure (spatial) index (FPI), by taking into consideration several interactions with anthropogenic and environmental factors. Spatial analytical methods such as the Analytic Hierarchy Process (AHP) and the Fuzzy logic technique were applied in an effort to estimate FPI for small scale fisheries. FPI was estimated as the fuzzy product of the following two indices: a) the fishery Suitability index (Sc) and b) the Activity index (Ac), based on the spatial distribution of registered fishing vessels, according to the formula:

$$\mathrm{FPI} = S_c \times A_c$$

The MCDA has been also applied to estimate FPI for medium scale fisheries such as bottom trawl (Hidalgo et al. 2020; Mérigot et al. 2020) and in the framework of the Recommendation GFCM/44/2021/6 for sustainable trawl fisheries targeting giant red shrimp and blue and red shrimp in the Levant Sea (FAO paper under preparation). Related application of MCDA on SSF has been performed for the entire Mediterranean Sea (Spedicato et al., 2021). Moreover, the outcomes derived by MCDA on SSF have been used as inputs to decision support tools oriented to fisheries management or/and marine spatial planning (e.g. Bordersen et al., 2017; Bastardie et al., 2017; Sini et al., 2019; Maina et al., 2021; Markantonatou et al., 2021).

Furthermore, the main components of the MCDA (Kavadas et al., 2015), were incorporated in 'fprmcda', an R package, which provides a tool for visualizing fishing pressure for small and medium-scale fisheries. 'fprmcda' composes a suit of geospatial techniques to quantify the synergistic effect of influential components (e.g., bathymetry, distance from coast, Chl-a concentration, MSF fishing effort, marine traffic activity, vessel capacity of ports, etc.) on defining fishing pressure. A general overview of the MCDA method and the main functions of the R package is given in the following flowchart.


Module Development (Workflow)
1 Load Spatial Data
(raster, shapefiles, or csv)
# 1a Score the Criteria
Scored_Data = fgrade(Data, cuts, scores)
Criteria
Bathymetry
Distance from coast
Legislation
Trawlers fishing effort
Purse seiners fishing effort
Marine traffic
Environmental
No-take zones
Input: Pair-wise comparison matrix between criteria
# 1b. Apply Analytic Hierarchy Process
AhpOut = OptAhp(PairwiseMatrix)
Outputs: Weights, RMSE
# 1c. Weighted Linear Combination method (WLC)
Sc = sum(AhpOut$Weights*Scored_Data)
# 1d. Normalise Sc
Sc <- FuzzyMember(Sc)
AHP
FM
Coastal fishery suitability index (Sc)
2 Load Fishing Fleet (FishG) data
(lon, lat, GTL)
Fishing Vessel's length * GT (VAIp)
Interpolation
Activity by spatial cell (VAIc)
Inputs: FishG data and spatial mask
# 2a. Apply IDW interpolation
IFg<- idwfg(FishG, mask.raster)
Outputs: Ac, Error
FM
# 2b Normalise Ac
IFg$Ac<- FuzzyMember(IFg$Ac)
Coastal fishery activity index (Ac)
3 Synthesis
# 3a Combine Sc, Ac
FPc=Sc* IFg$Ac
Fuzzy Product
Fishing Pressure from coastal fisheries (FPc)
# 3b Normalise FPc
FPc <- FuzzyMember(FPc)

Flowchart of steps and methods of the Multi-Criteria Decision Analysis (AHP: Analytic Hierarchy Process, FM: Fuzzy Membership) followed in the fprmcda R package.

Additionally, the method has been further expanded in order to include fishing effort estimations from EU Fisheries Dependent Information datasets (FDI; *https://stecf.jrc.ec.europa.eu/reports/fdi*, Table I. Effort by rectangle) and spatiotemporal environmental factors (i.e. weather conditions and primary productivity). The merit of including the above information to the MCDA method is that maps of fishing pressure can be estimated in days at sea and by quarter in finer spatial scales. A second release of the 'fprmcda' R package including improvements on the main functions and the expanded version of MCDA approach is planned to be integrated in RDBFIS-II (Maintaner/creator: Irida Maina in preparation). Latest version 0.1.5 can be downloaded from *https://cloudfs.hcmr.gr/index.php/s/MuUCkionc5sIuNP*.

'fprmcda' R package (Release 0.1.5)

```
Package: fprmcda
Type: Package
Title: Mapping of Fishing Pressure using Multi-Criteria Decision Analysis
Version: 0.1.5
Copyright: Hellenic Center for Marine Research
Description: A package for Visualizing Small and Medium-Scale Fisheries footprints
            using Multi-Criteria Decision Analysis.
Depends: R (>= 3.0.0)
Imports: raster, FuzzyAHP, phylin, methods, stats, sp, spatstat,deldir, maptools, shiny
Suggests: knitr, rgdal, rasterVis, RColorBrewer, maps, tmap, tmaptools, rmarkdown,
mapdata, latticeExtra
License: GPL (>= 2)
NeedsCompilation: no
VignetteBuilder: knitr
Encoding: UTF-8
LazyData: Yes
RoxygenNote: 6.1.1
Author: Dimitris Politikos [aut, cre],
  Irida Maina [aut, cre],
  Stefanos Kavadas [aut]
```

R Code (Tutorial)

~~~
title: |
  | Visualizing Small and Medium-Scale Fisheries footprints: the ***fprmcda*** R-Package
resource_files:
  - functions.png
  - workflow.png
output: rmarkdown::pdf_document
vignette: >
  %\VignetteIndexEntry{fprmcda package for R: Tutorial}
  %\VignetteEngine{knitr::rmarkdown}
  %\usepackage[utf8]{inputenc}

---
```{r setup, global_options, include = FALSE}
library(fprmcda)
library(knitr)
knitr::opts_chunk$set(collapse = T, comment = "#>")
knitr::opts_chunk$set(fig.pos = 'ht')
knitr::opts_chunk$set(fig.path="fig/")
~~~

\setcounter{secnumdepth}{5}

# Introduction

The `fprmcda` package provides a tool for visualizing fishing pressure for small and
medium-scale coastal fisheries in **R**. `fprmcda` combines the multi-criteria decision
analysis (MCDA) and geospatial techniques to quantify the synergistic effect of
influential components (e.g., bathymetry, distance from coast, Chl-a concentration,
fishing effort, marine traffic activity, vessel capacity of ports, etc.) on defining
fishing footprint. For more information about the methodology associated with the
`fprmcda` package, please see [(Kavadas et al.,
2015)](https://ejournals.epublishing.ekt.gr/index.php/hcmr-med-mar-
sc/article/viewFile/12919/12483.pdf)

The general workflow of `fprmcda` can be seen in Fig. 1. The functions used in `fprmcda`
are summarized in Fig. 2.
```{r, echo = FALSE, fig.cap = "Flowchart of steps and methods (AHP: Analytic Hierarchy
Process, FM: Fuzzy Membership) followed in fprmcda. More details about the methodolgy can
be found in Kavadas et al. (2015).", fig.align="default", out.width = "100%", dpi = 300,
fig.asp=0.10}
include_graphics("workflow.png")
```

This vignette describes in a stepwise way how to apply `fprmcda` for mapping small-scale
fisheries footprint in the central Aegean Sea (E. Mediterranean Sea).

To start using `fprmcda`, first install the package by typing (within R)
install.packages("fprmcda") and then load the package with
```{r help, eval = FALSE}
library(fprmcda)
help(fprmcda)
```

```{r, echo = FALSE, fig.cap = "Flowchart of steps and methods followed (AHP: Analytic
Hierarchy Process, FM: Fuzzy Membership), Kavadas et al. (2015).", fig.align="default",
out.width = "100%", fig.asp=0.1, dpi = 300}
include_graphics("functions.png")
```

# Tutorial
```
```

~~~~
The aim of this tutorial to visualize fishing footprint in the central Aegean Sea (CAS).
To achieve this, we use *FishData.grd* raster file, which includes the components of
bathymetry, distance from the coast and chlorophyll from the CAS and the
*Fishing_Ports.csv* csv file, which includes the total capacity of fishing vessels from
the main ports in CAS. Both data files are pre-installed in `fprmcda`.

To access the source code of the tutorial, type

```{r, eval = FALSE}
file.edit(system.file("demo", "demo.R", package = "fprmcda"))
```

&nbsp;
&nbsp;

To run the source code of the tutorial, found in the demo folder of the `fprmcda`
package, type
```{r, eval = FALSE}
demo("demo1", package = "fprmcda")
```


## Step-1: Libraries and Data import

First, we load the libraries,
```{r}
library(sp)            # To handle spatial data
library(raster)        # To handle raster objects
library(RColorBrewer) # To define color pallette
library(rasterVis)     # To plot raster objects
library(maps)          # To display maps
library(mapdata)       # Map databases
library(maptools)      # To read and handle spatial objects
library(latticeExtra) # Graphical Utilities Based on Lattice
library(FuzzyAHP)      # To perform AHP process
library(phylin)        # To perform IDW interpolation
```

Load the raster data into the current working directory as follows,

```{r}
# Load the raster data
Data <- brick(system.file("external/demo1", "FishData.grd", package = "fprmcda"))
# Load the csv file for fishing gear
fg <- read.csv(system.file("external/demo1", "Fishing_Ports.csv", package = "fprmcda"),
sep=",")
```

The contents of the variables `Data` and `fg` can be explored by,
```{r}
class(Data)
Data
head(fg)
```

The function `brick` loads the data from the `"FishData.grd"` file and formats it into a
raster object with layers, which is then assigned to `Data` via the
assignment operator `<-`. Each layer in FishData.grd file represents a variable, i.e.,
bathymetry, the distance from the coast and chlorophyll.

The data object `fg` is a dataframe that illustrates for each fishing port of the case
area, its longitude and latitude (first and second columns in the dataframe), and the
total capacity of fishing vessels (third column), represented as the total sum of $L*Gt$,
where $L$ is Vessel Length and $GT$ is the Gross tonnage.

Plot a layer (e.g., bathymetry) of the `Data` object with the `levelplot` function of the
`RasterVis` package as follows:
```{r, fig.cap="Bathymetry in meters of the case study area.", fig.align="center",
out.width = "80%"}
# Plot raster
~~~~

~~~
levelplot(Data$Bathymetry,
          margin = F,
          main = "Bathymetry",
          scales = list(cex = 1.2),
          xlab = list("Longitude",cex = 1.2),
          ylab = list("Latitude",cex = 1.2)) +
      # Add map on the levelplot
      latticeExtra::layer({
      ext <- as.vector(extent(Data$Bathymetry))
      boundaries <- map('worldHires', fill = TRUE,
                        xlim = ext[1:2], ylim = ext[3:4],
                        plot = FALSE)
      IDs <- sapply(strsplit(boundaries$names, ":"), function(x) x[1])
      bPols <- map2SpatialPolygons(boundaries, IDs = IDs,
                                   proj4string = CRS(projection(Data$Bathymetry)))
      sp.polygons(bPols, fill = 'grey80', data = list(bPols = bPols))
                         })
```

Plot the contents of the `fg` variable as follows,
```{r, fig.cap="Vessel capacity of fishing ports.", fig.align="center", out.width =
"60%"}
cexwei = 3*fg[,3]/max(fg[,3])
plot(fg[,1], fg[,2], xlab = "Longitude", ylab = "Latitude", cex = cexwei)
```

## Step-2: Calculation of coastal fishery suitability index (Sc)

For the calculation of the Fish Suitability Index (Sc), we need to:

2.1 Apply grades on `Data` using specific knowledge-based criteria.

2.2 Perform the Analytic Hierarchy Process to evaluate the relative importance of one
component over the others.

**2.1 Grading of components**

We grade each components in the `Data` raster object with values between 0 and 5 using
the following criteria,
```{r}
# Define criteria for bathymetry
bath_cuts <- c( -Inf, 0, 50, 100, 150, 200, 500 ) # in meters
bath_grad <- c(   NA, 1,  2,   3,   4,   5,   0 )
# Define criteria for distance from coast
coastdist_cuts <- c( -Inf, 0, 1.5, 3, 6) # in nautical miles
coastdist_grad <- c(   NA, 4,   3, 2, 1)
# Define grades for sea surface Chrollophyl
chl_cuts <- c( -Inf, 0, 1, 2, 2.5, 3) # in mg/m3
chl_grad <- c(    0, 1, 2, 3,   4, 5)
```
The higher the grading, the most `favorable` this area is for coastal fishing activities.

The function `fgrade.r` of the *fprmcda* package implements the grading of `Data` object,
```{r}
# Define the ranked raster graded_data having the same structure with Data RasterLayer
graded_data = Data
# Update the values of the graded_data RasterLayer using the criteria
values(graded_data$Bathymetry)  <- fgrade(values(Data$Bathymetry), bath_cuts, bath_grad)
values(graded_data$CoastDist)   <- fgrade(values(Data$CoastDist), coastdist_cuts,
coastdist_grad)
values(graded_data$Chlorophyll) <- fgrade(values(Data$Chlorophyll), chl_cuts, chl_grad)
```

The `fgrade` function is a step function that performs the scoring of the components. It
has two arguments: 1) the raster object and 2) the ranking criteria and returns an
updated raster with ranked integer values between 0-5.

We can plot the ranked raster object using the `leveplot` function of the *rasterVis*
package as,
```{r, fig.cap="Bathymetric map after grading.", fig.align="center", out.width = "80%"}
~~~

~~~
  # Create a color palette
  buls <- brewer.pal(n = 9,"Blues")
  # Plot the graded raster
  levelplot(ratify(graded_data$Bathymetry),
            cuts = max(values(graded_data$Bathymetry), na.rm = TRUE),
            margin = F,
            col.regions = buls,
            main = "Bathymetry (Graded)",
            att = "ID",
            scales = list(cex = 1.2),
            xlab = list("Longitude",cex = 1.2),
            ylab = list("Latitude",cex = 1.2)) +
      # Add map on the leveplot
      latticeExtra::layer({
      ext <- as.vector(extent(graded_data$Bathymetry))
      boundaries <- map('worldHires', fill = TRUE,
                        xlim = ext[1:2], ylim = ext[3:4],
                        plot = FALSE)
      IDs <- sapply(strsplit(boundaries$names, ":"), function(x) x[1])
      bPols <- map2SpatialPolygons(boundaries, IDs = IDs,
                                   proj4string = CRS(projection(graded_data$Bathymetry)))
      sp.polygons(bPols, fill = 'grey80', data = list(bPols = bPols))
                          })
```

**2.2 Apply Analytic Hierarchy Process (AHP)**

During this step, pair-wise comparisons between the components are organized in a matrix
form, in which the components on the rows are compared against those in the columns. Nine
levels of preference (1-9) are adopted to evaluate the relative importance of one
component over the others. The higher the level of preference of a component the higher
the importance of this component. The levels of preference derived from the comparisons
formed the elements of the upper hemi-matrix;the lower hemi-matrix was then is produced
under the assumption of reciprocity,that is, if the preference for component $i$ compare
to $j$ was $a_{ij}$, then the preference for component $j$ compare to $i$ should be
$a_{ji}=1/a_{ij}$.

*! Important note:* The `pairwise_matrix` should be reciprocal, i.e., the diagonal
elements should be 1 (i.e., matrix[i,i]=1) and the other elements follow the rule
matrix[j,i]=1/matrix[i,j].

The pairwise matrix used in this tutorial is,
```{r, echo=FALSE, results = 'asis'}
comparisonMatrixValues1 <- read.table(system.file("external/demo1", "AhpMatrix.txt",
package = "fprmcda"), header=T,sep="\t")
kable(comparisonMatrixValues1, align = rep('c', 4))
```

We insert the pairwise matrix in R,
```{r}
# Pairwise matrix
pairwise_matrix <- c( 1, 5,   4,
                    1/5, 1, 1/3,
                    1/4, 3,   1)
pairwise_matrix = matrix(pairwise_matrix,
                         nrow = sqrt(length(pairwise_matrix)),
                         ncol = sqrt(length(pairwise_matrix)),
                         byrow = TRUE)
```


The Analytic Hierarchy Process is applied with the `getConsistentAHP` function
of the *fprmcda* package using the `PairwiseMatrix` as input,

```{r}
# Apply AHP to pairwise matrix defined in config.R to calculate
# weights, constistency ratio, etc.
ahp_out = getConsistentAHP(pairwise_matrix)
```
~~~

~~~
The `getConsistentAHP` function accepts as input an initial pair-wise reciprocal matrix
and does the following: (1) Applies the Analytic Hierarchy Process based on Saaty (2001)
(`ahp` function from the *FuzzyAHP* package); (2) adjusts the matrix following the
iteration scheme of Xu (2004) and reapplies `ahp` function till consistency ratio
criterion is achieved, i.e., CR<0.1, (3) Returns the **weights** that describe the
relative contribution of each criterion, the Consistency Ratio (**CR**), the adjusted
pairwise matrix (**PairwiseMatAdj**), the error defined as the maximum absolute
difference between the initial and adjusted matrix (**delta**), and the standard
deviation of their difference (**sig**).

*!Good practice*: delta should be <2 and sig ~1.

The main outputs of the `getConsistentAHP` are:
```{r}
ahp_out
```
The `ahp_out$Weights` indicate the importance of each component (i.e., bathymetry,
distance from coast, chlorophyll).

**2.3 Apply Weighted Linear Combination method (WLC)**

In this step, the Weighted Linear Combination method (WLC) performs a multiplication, in
which each weight from `ahp_out$weights` is multiplied with the corresponding component
from the `graded_data` raster object. The coastal fishery suitability index (Sc) is,
then, calculated by summing the products over all components, i.e, $S_c=\sum_{i=1}^{3}
ahp\_out\$weights_i * graded\_data_i$,
```{r}
# Apply of Weighted Linear Combination to calculate Sc=Sum(Weights*Data)
Sc = sum(ahp_out$weights * graded_data)
# Normalise Sc
Sc <- fuzzyMember(Sc, "raster")
Sc
```
The `fuzzyMember` function of the `fprmcda` package rescales the values of the data to
[0-1]. To see how `fuzzyMember` function works, type
```{r, eval=FALSE}
help(fuzzyMember)
```

The plot of Sc is shown by,
```{r, fig.cap="Coastal fish suitability index (Sc).", fig.align="center", out.width =
"85%"}
# Plot the Sc raster
levelplot(Sc,
          margin = F,
          main = "Coastal fish suitability index (Sc)",
          scales = list(cex = 1.2),
          xlab = list("Longitude",cex = 1.2),
          ylab = list("Latitude",cex = 1.2)) +
      # Add map on the leveplot
      latticeExtra::layer({
      ext <- as.vector(extent(Sc))
      boundaries <- map('worldHires', fill = TRUE,
                        xlim = ext[1:2], ylim = ext[3:4],
                        plot = FALSE)
      IDs <- sapply(strsplit(boundaries$names, ":"), function(x) x[1])
      bPols <- map2SpatialPolygons(boundaries, IDs = IDs,
                                   proj4string = CRS(projection(Sc)))
      sp.polygons(bPols, fill = 'grey80', data = list(bPols = bPols))
                          })
```

## Step-3: Calculation of fishing vessel capability (Ac)

The `idwfg` function of the `fprmcda` package is applied on the `fg` dataframe to perform
Inverse Distance Weighted (IDW) interpolation (`idw` function from the `phylin` package)
in order to map in space the capability of fishing vessels. The `idwfg` function does the
following: 1) creates an empty raster with the same extent as in `Data` Raster object, 2)
performs the IDW interpolation, (3) Returns a raster object with the interpolated values
and the RMSE error.
~~~

~~~~
```{r}
# Apply IDW interpolation to fishing gear data
fg_intp <- idwfg(fg, Data$Bathymetry)
fg_intp
```

Use of `fuzzyMember` function of the `fprmcda` package to normalize the `Fg$Ac` and
make the plot
```{r, fig.cap="Coastal fishing vessel capability (Ac). Blue dots indicate the fishing
port locations. The size of the dots is proportional to the magnitude of total capacity
of fish vessels.", fig.align="default", out.width = "80%"}
# Spatialize fishing Lon, Lat data points
xy = fg[,1:2]
poi <- SpatialPoints(xy)
proj4string(poi) = CRS(projection(fg_intp$raster))
cexwei = 3 * fg[,3] / max(fg[,3])
# Normalise
fg_intp$raster = fuzzyMember(fg_intp$raster, "raster")
Ac = fg_intp$raster
# Plot raster
levelplot(Ac,
          margin = F,
          main = "Interpolated fishing vessel capability (Ac)",
          scales = list(cex=1.2),
          xlab = list("Longitude",cex = 1.25),
          ylab = list("Latitude",cex = 1.25)) +
      # Add map on the leveplot
      latticeExtra::layer({
      ext <- as.vector(extent(fg_intp$raster))
      boundaries <- map('worldHires', fill = TRUE,
                        xlim = ext[1:2], ylim = ext[3:4],
                        plot = FALSE)
      IDs <- sapply(strsplit(boundaries$names, ":"), function(x) x[1])
      bPols <- map2SpatialPolygons(boundaries, IDs = IDs,
                                   proj4string = CRS(projection(fg_intp$raster)))
      sp.polygons(bPols, fill = 'grey80', data = list(bPols = bPols))
                          })+
      latticeExtra::layer(sp.points(poi, pch = 20, col = "blue", cex = cexwei))
```

To see the different options of applying `idwfg` function, type
```{r, eval=FALSE}
help(idwfg)
```

## Step-4: Calculation of Fishing Pressure (FPc)

In this step, we combine the outputs of Steps 3 and 4. The Fishing Pressure (FPc) is
calculated as the product of Sc and Ac raster objects, $FPc = Sc * Ac$, as
```{r, fig.cap="Fishing pressure index from the small scale coastal fishery (FPc).",
fig.align="center", out.width = "80%"}
# Calculate FPc
FPc = Sc * Ac
# Normalise FPc
FPc <- fuzzyMember(FPc, "raster")
# Plot the raster
levelplot(FPc,
          margin = F,
          main = "Fishing pressure - FPc",
          scales = list(cex=1.2),
          xlab = list("Longitude",cex = 1.2),
          ylab = list("Latitude",cex = 1.2)) +
      # Add map on the leveplot
      latticeExtra::layer({
      ext <- as.vector(extent(FPc))
      boundaries <- map('worldHires', fill = TRUE,
                        xlim = ext[1:2], ylim = ext[3:4],
                        plot=FALSE)
      IDs <- sapply(strsplit(boundaries$names, ":"), function(x) x[1])
~~~~

~~~
      bPols <- map2SpatialPolygons(boundaries, IDs = IDs,
                                   proj4string = CRS(projection(FPc)))
      sp.polygons(bPols, fill = 'grey80', data = list(bPols = bPols))
                          })
```

&nbsp;
&nbsp;
&nbsp;

# Notes

In case the data are in `csv` form, then they need to be converted in raster form. For
instance, load the `csv` from the *fprmcda* package,
```{r}
df <- read.csv(system.file("extdata", "MyData.csv", package = "fprmcda"), header = TRUE,
sep = ",")
# Read the dataframe
head(df)
```

Load the libraries,

```{r}
library(deldir, quietly = TRUE)
library(rgdal)
```
Then, apply the `csv2raster` function of the *fprmcda* package to convert the dataframe
`df` to RasterLayer object `r`. For instance, the column of dataframe referring to
bathymetry can is converted to raster with
```{r}
#Assign a projection
map.proj <- CRS("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0")
# Create a raster for Bathymetry from the df dataframe
r <- csv2raster(df, df$Bathymetry, map.proj)
names(r) <- "Bathymetry"
r
```

Then, `r` can be plotted by
```{r, fig.cap="Bathymetry plot derived from the .csv file.", fig.align="center",
out.width = "80%"}
# Plot the raster
levelplot(r,
          margin = F,
          main = "Bathymetry (m)",
          scales = list(cex = 1.2),
          xlab = list("Longitude",cex = 1.2),
          ylab = list("Latitude",cex = 1.2)) +
      # Add map on the leveplot
      latticeExtra::layer({
      ext <- as.vector(extent(FPc))
      boundaries <- map('worldHires', fill = TRUE,
                        xlim = ext[1:2], ylim = ext[3:4],
                        plot = FALSE)
      IDs <- sapply(strsplit(boundaries$names, ":"), function(x) x[1])
      bPols <- map2SpatialPolygons(boundaries, IDs = IDs,
                                   proj4string = CRS(projection(FPc)))
      sp.polygons(bPols, fill = 'grey80', data = list(bPols = bPols))
                          })
```

&nbsp;
&nbsp;
&nbsp;
~~~

References

Bastardie, F., Angelini, S.,Bolognini, L.,Fuga, F.,Manfredi, C.,Martinelli, M.,Nielsen, J.R.,Santojanni, A.,Scarcella, G.,GratiF., 2017.Spatial planning for fisheries in the Northern Adriatic: working toward viable and sustainable fishing. Ecosphere, 8 (2), Article e01696, 10.1002/ecs2.1696

Brodersen, M.M., Pantazi, M., Kokkali, A., Panayotidis, P., Gerakaris, V., Maina, I., Kavadas, S., Kaberi, H., Vassilopoulou, V., 2017. Cumulative impacts from multiple human activities on seagrass meadows in eastern Mediterranean waters: the case of Saronikos Gulf (Aegean Sea, Greece). Environmental Science and Pollution Research.doi:10.1007/s11356-017-0848-7

Hidalgo, M., Ligas, A., Bellido, J., Bitetto, I., Carbonara, P., Carlucci, R., Guijarro, B., Jadaud, A., Lembo, G., Manfredi, C., Esteban, A., Garofalo, G., Ikica, Z., García, C., Gil de Sola, L., Kavadas, S., Maina, I., Sion, L., Vittori, S., & Vrgoc, N. 2019. Size-dependent survival of European hake juveniles in the Mediterranean Sea. Scientia Marina, 0. doi:http://dx.doi.org/10.3989/scimar.04857.16A

Kavadas, S., I. Maina, D. Damalas, I. Dokos, M. Pantazi, and V. Vassilopoulou, (2015). Multi-Criteria Decision Analysis as a tool to extract fishing footprints and estimate fishing pressure: application to small scale coastal fisheries and implications for management in the context of the Maritime Spatial Planning Directive. Mediterranean Marine Science 16: 294-304.http://dx.doi.org/10.12681/mms.1087

Maina, I., Kavadas, S., Vassilopoulou, V., Bastardie F., 2021. Fishery spatial plans and effort displacement in the eastern Ionian Sea: A bioeconomic modelling. Ocean and Coastal Management.203, 105456, https://doi.org/10.1016/j.ocecoaman.2020.105456.

Markantonatou, V., Giakoumi, S., Koukourouvli, N., Maina, I., Gonzalez-Mirelis, G., Sini, M., Maistrelis, K., Stithou, M., Gadolou, E., Petza, D., et al. 2021. Marine spatial plans focusing on biodiversity conservation: The case of the Aegean Sea. Conserv. Mar. Freshw. Ecosyst. 1– 15. https://doi.org/10.1002/aqc.3610

Mérigot, B., Gaertner, JC., Brind'Amour, A., Carbonara, P., Esteban, A., Garcia-Ruiz, C., Gristina, M., Imzilen, T., Jadaud, A., Joksimovic, A., Kavadas, S., Kolitari, J., Maina, I., Maiorano, P., Manfredi, C., Mifsud, R., Ordines., F., Peristeraki, P., Relini, G., Sbrana, M., Spedicato, MT., Thasitis, I., Vittori, S., Vrgoc, N., 2019. Stability of the relationships among demersal fish assemblages and environmental trawling drivers at large spatio-temporal scales in the northern Mediterranean Sea. Scientia Marina doi: http://dx.doi.org/10.3989/scimar.04954.30A.

Saaty, T.L. (2001). Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World, New Edition 2001 (3 Revised). Pittsburgh, PA: RWS Publications, ISBN 978-0962031786.

Sini M, Vatikiotis K, Thanopoulou Z, Katsoupis C, Maina I, Kavadas S, Karachle PK, Katsanevakis S. 2019. Small-Scale Coastal Fishing Shapes the Structure of Shallow Rocky Reef Fish in the Aegean Sea. Frontiers in Marine Science.doi: 10.3389/fmars.2019.00599

Spedicato M.T., Cannas R., Mahe Kelig, Morales B., Tsigenopoulos C., Zane L., Kavadas S., Maina I., Scarcella G., Sartor P., Bandelj V., Russo T., Fiorentino F. (2021). Study on advancing fisheries assessment and management advice in the Mediterranean by aligning biological and management units of priority species. MED_UNITs.Final Report.Publications Office of the European Union.ISBN 978-92-95225-33-6. https://doi.org/10.2926/909535

Xu, Z. (2004). A practical method for improving consistency of judgement matrix in the ahp.Journal of Systems Science and Complexity, 17(2).

Analysis of Vessel Monitoring System (VMS) data

Primary analysis of VMS dataset

Data quality control is the first key step in VMS data analysis. Common errors found in VMS datasets are;

- vessel positions on land
- implausibly high speeds (over than 20 knots)
- headings outside a compass range (0 to 360)
- duplicated records.

In addition, vessel positions lying either in harbours or very close to them were identified. After filtering out the common errors, points in harbour were removed using a 300 meters buffer from coastline. Vessels with the same fisheries strategy (trawlers or purse seiners) identified by the type of vessel's license which provided by the fleet register. For the classification of mobile polyvalent licenses, as trawlers or purse seiners, temporal and spatial restrictions for each fisheries strategy were used. For example fishing vessels with mobile polyvalent licenses which usually operate in the period from June to September that trawling is prohibited, were classified as purse seines.

It was assumed that a "fishing day" for trawlers is a 24 hours period and for purse seiners is the period from 20:00pm until next day's 4:00am which differentiated by one hour during winter months (19:00pm to 4:00am). Classification of fishing activity as "fishing", "steaming" or "mooring" consist one of the most important processes in VMS data analysis. Speed thresholds for trawlers and purse seiners were used in order to define the "fishing" activity. These speed thresholds were selected based on the depth stratum, experimental sampling and onboard observations. Trawlers were classified as "fishing" at speed intervals lower or equal to 4 knots and purse seiners at speed intervals lower or equal to 1 knot. Moreover, fisheries legislation for both gears was integrated in order to define fishing areas that fishing activity is permitted.

Fishing Effort Estimations

Analysis of VMS data (primary and fishing effort estimations) was based on PLSQL stored procedures which provides advantages to speed up calculations. In particular, fishing effort from trawlers and purse seiners was estimated for signals that classified as "fishing". The distribution of fishing effort expressed in days at sea, in days multiplied by Gross tonnage (GT) and also in days multiplied by Kilowatt (KW) (Kavadas et al., 2014; Maina et al., 2016). It was assumed that every vessel visits a fishing rectangle (e.g. 5x5 km) more than once at a "fishing day"; thereby a weighted value is given in each rectangle depends on the number of visits it receives within 24 hours. Annual and monthly distribution maps of fishing effort estimated by the following procedures (Table 1).

Table 1: Methodology to estimate the fishing effort by grid cell (a) Each group of signals spatially joined to the related grid cell (5x5 km)	
(b) Calculate the number of signals (p) per vessel (v) and day (d)	Tv,d=count(pv,d)
(c) Calculate the number of signals per grid cell (c), vessel and day	Tc,v,d=count(pc,v,d)
(d) Allocate weight in each grid cell per vessel and day. The resulting value is multiplied by tonnage (GT) and power engine (KW)	Ec,v,d= Tv,d,c / Tv,d (≤1) Σ Ec,v,d=1 (one "fishing day") Gc,v,d= Ec,v,d*GTv Wc,v,d= Ec,v,d*KWv

(e) Estimation of fishing effort by 5x5 grid cell	DSc= ΣvΣd Ec,v,d
	GTDc=ΣvΣd Gc,v,d
	KWDc=ΣvΣd Wc,v,d

PLSQL code

```
create or replace procedure WG84toGGRS87(lat1 in number, lng1 in number, East out number, North out number) is
begin
  DECLARE
      pi number :=3.14159265358979;
      lat number := lat1 * pi / 180;
      lng number := lng1 * pi / 180;
      a number := 6378137;
      b number := 6356752.31424518;
      e2 number := (power(a,2) - power(b,2)) / power(a,2);
      v number := a / sqrt(1 - e2 * (sin(lat) * sin(lat)));
      X number := v * Cos(lat) * Cos(lng);
      Y number := v * Cos(lat) * Sin(lng);
      Z number := (v * (1 - e2)) * Sin(lat);
      px number := 199.723;
      py number := -74.03;
      pz number := -246.018;
      rw number := 0;
      rf number := 0;
      rk number := 0;
      ks number := 1;
      c1 number := Cos(rw);
      c2 number := Cos(rf);
      c3 number := Cos(rk);
      s1 number := Sin(rw);
      s2 number := Sin(rf);
      s3 number := Sin(rk);
      D11 number := c2 * c3;
      D21 number := -c2 * s3;
      D31 number := Sin(rf);
      D12 number := s1 * s2 * c3 + c1 * s3;
      D22 number := -s1 * s2 * s3 + c1 * c3;
      D32 number := -s1 * c2;
      D13 number := -c1 * s2 * c3 + s1 * s3;
      D23 number := c1 * s2 * s3 + s1 * c3;
      D33 number := c1 * c2;
      X1 number := px + ks * (D11 * X + D12 * Y + D13 * Z);
      Y1 number := py + ks * (D21 * X + D22 * Y + D23 * Z);
      Z1 number := pz + ks * (D31 * X + D32 * Y + D33 * Z);
      lat0 number;
      No number;
      h number;
      m0 number;
      es2 number;
      eta number;
      Bf number;
      Vf number;
      etaf number;
      n number;
      r1 number;
      r2 number;
      r3 number;
      r4 number;
      r5 number;
      Northing number;
      Easting number;
      ys number;
  BEGIN
      lng := Atan2(Y1, X1);
      lat0 := Atan2(Z1, Sqrt(X1 * X1 + Y1 * Y1));
      b := 6356752.31414036;
      e2 := (power(a,2) - power(b,2)) / power(a,2);
      while (Abs(lat - lat0) > 0.0000000001)
        loop
              No := a / Sqrt(1 - e2 * Sin(lat0) * Sin(lat0));
              h := Sqrt(X1 * X1 + Y1 * Y1) / Cos(lat0) - No;
              lat := lat0;
              lat0 := Atan(Z1 / Sqrt(X1 * X1 + Y1 * Y1) * (1 / (1 - e2 * No / (No+ h))));
        end loop;
          lng := lng - 24 * PI / 180;
          m0 := 0.9996;
          es2 := (Power(a,2) - Power(b,2)) / (Power(b,2));
          V := Sqrt(1 + es2 * Cos(lat) * Cos(lat));
          eta := Sqrt(es2 * Cos(lat) * Cos(lat));
          Bf := Atan(Tan(lat) / Cos(V * lng) * (1 + eta * eta / 6 * (1 - 3 * Sin(lat) * Sin(lat)) * lng * lng * lng * lng));
          Vf := Sqrt(1 + es2 * Cos(Bf) * Cos(Bf));
          etaf := Sqrt(es2 * Cos(Bf) * Cos(Bf));
          n := (a - b) / (a + b);
          r1 := (1 + n * n / 4 + n * n * n * n / 64) * Bf;
          r2 := 3.0 / 2.0 * n * (1 - n * n / 8) * Sin(2 * Bf);
          r3 := 15.0 / 16.0 * n * n * (1 - n * n / 4) * Sin(4 * Bf);
          r4 := 35.0 / 48.0 * n * n * n * Sin(6 * Bf);
          r5 := 315.0 / 512.0 * n * n * n * n * Sin(8 * Bf);
          Northing := (a / ((1 + n))) * (r1 - r2 + r3 - r4 + r5) * m0 - 0.001; --+ 4202812 - 4207988.1206046063;
          ys := Tan(lng) * Cos(Bf) / Vf * (1 + etaf * etaf * lng * lng * Cos(Bf) * Cos(Bf) * (etaf * etaf / 6 + lng * lng / 10));
          ys := Ln(ys + Sqrt(ys * ys + 1));
          Easting := m0 * Power(a,2) / b * ys + 500000;
          --- correction due to diff between procedure and ArcGIS
          Northing := Northing + 287;
          Easting := Easting + 148.76;
          East:= Easting;
          North:=Northing;
  END;
end WG84toGGRS87;
/*
Projected Coordinate System:  Greek_Grid
```

```
Projection:  Transverse_Mercator
False_Easting:  500000.00000000
False_Northing:  0.00000000
Central_Meridian:  24.00000000
Scale_Factor:  0.99960000
Latitude_Of_Origin:  0.00000000
Linear Unit:   Meter*/

create or replace procedure vms2021_put_rectid is
begin
 declare
    X1  NUMBER(7);
    Y1  NUMBER(7);
    X2  NUMBER(7);
    Y2  NUMBER(7);
    Xstep  NUMBER(7);
    Ystep  NUMBER(7);
    RowC number;
    ColC number;
    m_lat number(7,4);
    m_long number(7,4);
    m_rect number(10);
    east number;
    north number;
    grid_rows number :=146;
    grid_columns number :=187;
    xDist number;
    yDist number;
    nCols number;
    nRows number;
    rectID number(10);

    CURSOR MasterTC is SELECT vms_latitude, vms_longitude FROM fv_vms_2021
    where to_char(vms_date,'yyyy')='2021' for update;
    begin
      X1:=  79393;
      Y1:=  4552687;
      X2:=  1014393;
      Y2:=  3822687;
      open MasterTC;
      LOOP
        fetch MasterTC into m_lat, m_long;
        EXIT WHEN MasterTC%NOTFOUND;
        wg84toggrs87(m_lat, m_long, east, north);
        update fv_vms_2021 set X=east where current of MasterTC;
        update fv_vms_2021 set Y=north where current of MasterTC;
        -- ERS 30x30 nmil
        grid_rows:=13;
        grid_columns:=23;
        Xstep:=55560;
        Ystep:=55560;

        xDist:=east - X1;
        yDist:=Y1-north;
        nCols:=xDist/Xstep;
        nRows:=yDist/Xstep;
        rectID:=(trunc(nRows,0)*grid_columns)+trunc(nCols,0)+1;
        update fv_vms_2021 set rect_ers=rectID where current of MasterTC;

        -- 5x5 km
        grid_rows:=146;
        grid_columns:=187;
        Xstep:=5000;
        Ystep:=5000;
        xDist:=east - X1;
        yDist:=Y1-north;
        nCols:=xDist/Xstep;
        nRows:=yDist/Xstep;
        rectID:=(trunc(nRows,0)*grid_columns)+trunc(nCols,0)+1;
        update fv_vms_2021 set rect_5km=rectID where current of MasterTC;

        -- 3x3 km
        grid_rows:=243;
        grid_columns:=312;
        Xstep:=3000;
        Ystep:=3000;
        --wg84toggrs87(m_lat, m_long, east, north);
        xDist:=east - X1;
        yDist:=Y1-north;
        nCols:=xDist/Xstep;
        nRows:=yDist/Xstep;
        rectID:=(trunc(nRows,0)*grid_columns)+trunc(nCols,0)+1;
        update fv_vms_2021 set rect_3km=rectID where current of MasterTC;
        -- 2x2 km
        grid_rows:=365;
        grid_columns:=468;
        Xstep:=2000;
        Ystep:=2000;
        xDist:=east - X1;
        yDist:=Y1-north;
        nCols:=xDist/Xstep;
        nRows:=yDist/Xstep;
        rectID:=(trunc(nRows,0)*grid_columns)+trunc(nCols,0)+1;
        update fv_vms_2021 set rect_2km=rectID where current of MasterTC;
        -- 1x1 km
        grid_rows:=730;
        grid_columns:=935;
        Xstep:=1000;
        Ystep:=1000;
        xDist:=east - X1;
        yDist:=Y1-north;
        nCols:=xDist/Xstep;
        nRows:=yDist/Xstep;
        rectID:=(trunc(nRows,0)*grid_columns)+trunc(nCols,0)+1;
        update fv_vms_2021 set rect_1km=rectID where current of MasterTC;
        -- 250x250 m
        grid_rows:=2920;
        grid_columns:=3740;
        Xstep:=250;
        Ystep:=250;
        xDist:=east - X1;
        yDist:=Y1-north;
        nCols:=xDist/Xstep;
        nRows:=yDist/Xstep;
        rectID:=(trunc(nRows,0)*grid_columns)+trunc(nCols,0)+1;
        update fv_vms_2021 set rect_250m=rectID where current of MasterTC;
      END LOOP;
      close MasterTC;
      COMMIT;
  end;
  end vms2021_put_rectid;
```

```
create or replace procedure VMS_DUPL2021 is
begin
 declare
 mVMS_AMAS  VARCHAR2(9);
 currAMAS  VARCHAR2(9);
 mVMS_DATE date;
 currDATE  date;
 mVMS_LATITUDE number(7,4);
 currLAT  number(7,4);
 mVMS_LONGITUDE  number(7,4);
 currLON number(7,4);
 mDUPLICATE char(1);

  CURSOR c1 IS SELECT VMS_AMAS,VMS_DATE,VMS_LATITUDE,VMS_LONGITUDE, duplicate
       FROM FV_VMS_2021
       order by VMS_AMAS,VMS_DATE,VMS_LATITUDE,VMS_LONGITUDE for update;

  begin
   OPEN c1;
   FETCH c1 INTO mVMS_AMAS,mVMS_DATE,mVMS_LATITUDE,mVMS_LONGITUDE, mDUPLICATE ;
   currAMAS:=null;
   currDATE:=null;
   currLAT:=null;
   currLON:=null;
   LOOP
       IF (currAMAS=mVMS_AMAS) and (currDATE=mVMS_DATE) and (currLAT=mVMS_LATITUDE) and
(currLON=mVMS_LONGITUDE) THEN
        update FV_VMS_2021 set duplicate='Y' where current of c1;
      END IF;
       currAMAS:=mVMS_AMAS;
       currDATE:=mVMS_DATE;
       currLAT:=mVMS_LATITUDE;
       currLON:=mVMS_LONGITUDE;
      FETCH c1 INTO mVMS_AMAS,mVMS_DATE,mVMS_LATITUDE,mVMS_LONGITUDE, mDUPLICATE ;
       EXIT WHEN c1%NOTFOUND;

 END LOOP;
 commit;
 close c1;
 END;
end VMS_DUPL2021;

begin
 vms2021_put_rectid;
end;

begin
  vms_dupl2021;
end;

select t.vesselid, count(*) from fv_vms_2021 t
     where t.duplicate='Y'
     group by t.vesselid ORDER BY 2 desc;

select t.vms_amas, count(*) from fv_vms_2021 t
     where t.duplicate='Y'
     group by t.vms_amas ORDER BY 2 desc;
```

```
update fv_vms_2021 v set v.gear=v.vms_vessel_gear1;
commit;

drop table FV_OTB_VMS2021_ANALYSIS_S1;
create table FV_OTB_VMS2021_ANALYSIS_S1 as
 select t.vms_amas, --t.vms_vessel_gear1,
      t.rect_2km,
      --t.dcf_area,
      to_char(t.vms_date,'YYYYMMDD') as dd,
      count(case when t.vms_speed between 0 and 4 then 1 end) as fishing_pings
 from fv_vms_2021 t
 where t.duplicate is null
 and t.vms_speed between 0 and 20
 and t.gear='OTB'
 group by t.vms_amas, --t.vms_vessel_gear1,
      t.rect_2km,
      --t.dcf_area,
      to_char(t.vms_date,'YYYYMMDD')
 order by 1, 3, 2 ;

drop table FV_VMS_ANALYSIS_work77a_tm;
create table FV_VMS_ANALYSIS_work77a_tm as
select t.vms_amas, --t.vms_vessel_gear1,
      --t.dcf_area,
      to_char(t.vms_date,'YYYYMMDD') as dd,
      --max(t.vms_date) as maxd, min(t.vms_date) as mind,
      24*(max(t.vms_date)-min(t.vms_date)) as fish_hours
 from  fv_vms_2021 t, p_fishing_rect_2x2km r
 where r.rect_2x2km=t.rect_2km
 and r.mooring is null
 --and t.vms_speed between 0 and 4
 and t.gear='OTB'
 group by t.vms_amas, to_char(t.vms_date,'YYYYMMDD')
 order by 1, 3, 2;

-- Create/Recreate primary, unique and foreign key constraints
alter table FV_VMS_ANALYSIS_WORK77a_TM
  add constraint tm77a_pri primary key (VMS_AMAS, DD);

drop table FV_OTB_VMS2021_ANALYSIS_S2;
create table FV_OTB_VMS2021_ANALYSIS_S2 as
select t.vms_amas, --t.vms_vessel_gear1,
      t.rect_2km,
      --t.dcf_area,
      t.dd,
      t.fishing_pings
 from  FV_OTB_VMS2021_ANALYSIS_S1 t, p_fishing_rect_2x2km r
 where r.rect_2x2km=t.rect_2km and r.mooring is null--not '2x2kmWGS84'
 --and t.vms_vessel_gear1='OTB'
 order by 1, 3, 2 ;

-- Create/Recreate primary, unique and foreign key constraints
alter table FV_OTB_VMS2021_ANALYSIS_S2
  add constraint otb2021_s2_pri primary key (VMS_AMAS, RECT_2KM, DD);


drop table FV_OTB_VMS2021_ANALYSIS_S3;
create table FV_OTB_VMS2021_ANALYSIS_S3 as
select t.vms_amas, --t.vms_vessel_gear1,
      --t.rect_2km,
      --t.dcf_area,
      t.dd,
      sum(t.fishing_pings) as day_signals
 from  FV_OTB_VMS2021_ANALYSIS_S2 t
 group by t.vms_amas, t.dd
 --and t.vms_vessel_gear1='OTB'
 order by 1, 2;

alter table FV_OTB_VMS2021_ANALYSIS_S3
 add constraint otb2021_s3_pri primary key (VMS_AMAS, DD);

-- Add/modify columns
alter table FV_OTB_VMS2021_ANALYSIS_S2 add day_pings number;
alter table FV_OTB_VMS2021_ANALYSIS_S2 add day_fish_hours number;
alter table FV_OTB_VMS2021_ANALYSIS_S2 add vessel_day_wg number(5,2);
alter table FV_OTB_VMS2021_ANALYSIS_S2 add LEN_CAT VARCHAR2(7);
alter table FV_OTB_VMS2021_ANALYSIS_S2 add GT number(10,2);
alter table FV_OTB_VMS2021_ANALYSIS_S2 add KW number(10,2);

update FV_OTB_VMS2021_ANALYSIS_S2 s2 set s2.day_pings=
      (select s3.day_signals from FV_OTB_VMS2021_ANALYSIS_S3 s3
       where s3.vms_amas=s2.vms_amas
       and s3.dd=s2.dd);
commit;

update FV_OTB_VMS2021_ANALYSIS_S2 s2
      set s2.vessel_day_wg=s2.fishing_pings/s2.day_pings
      where s2.day_pings>0;
commit;

update FV_OTB_VMS2021_ANALYSIS_S2 s2 set s2.day_fish_hours=
      (select tm.fish_hours from FV_VMS_ANALYSIS_WORK77a_TM tm
       where tm.vms_amas=s2.vms_amas
       and tm.dd=s2.dd);
commit;

alter table fv_vms_vesselid_all add len_cat_l4 varchar2(6);
update fv_vms_vesselid_all a set a.len_cat_l4=
      case when a.loa<6 then 'VL0006'
       else case when a.loa<12 then 'VL0612'
        else case when a.loa<18 then 'VL1218'
         else case when a.loa<24 then 'VL1824'
          else case when a.loa<40 then 'VL2440'
           else case when a.loa>=40 then 'VL40XX' end end end end end end;
commit;

--- set LEN_CAT

update FV_OTB_VMS2021_ANALYSIS_S2 s2 set s2.len_cat=
      (select a.len_cat_l4 from fv_vms_vesselid_all a
       where a.amas=s2.vms_amas);
commit;

update FV_OTB_VMS2021_ANALYSIS_S2 s2 set s2.gt=
      (select a.gt from fv_vms_vesselid_all a
       where a.amas=s2.vms_amas);

update FV_OTB_VMS2021_ANALYSIS_S2 s2 set s2.kw=
      (select a.kw from fv_vms_vesselid_all a
       where a.amas=s2.vms_amas);


drop table FV_OTB_VMS2021_ANALYSIS_S4;
create table FV_OTB_VMS2021_ANALYSIS_S4 as
select  substr(s2.dd,1,6) as mm,
       s2.rect_2km,
       r.dcf_area,
       r.gsa,
       s2.len_cat,
       sum(s2.vessel_day_wg) as rect_wg,
       sum(s2.vessel_day_wg*s2.gt) as rect_wg_gt,
       sum(s2.vessel_day_wg*s2.kw) as rect_wg_kw,
       avg(tm.fish_hours) as day_hours
       from FV_OTB_VMS2021_ANALYSIS_S2 s2, p_fishing_rect_2x2km r, FV_VMS_ANALYSIS_WORK77a_TM tm
       where s2.rect_2km=r.rect_2x2km
       and tm.vms_amas=s2.vms_amas
       and tm.dd=s2.dd
       group by substr(s2.dd,1,6), s2.rect_2km, r.dcf_area, r.gsa, s2.len_cat
       order by 4, 1, 2, 3;

-- Add/modify columns
alter table FV_OTB_VMS2021_ANALYSIS_S4 add rect_hour number;

update FV_OTB_VMS2021_ANALYSIS_S4 s4
      set s4.rect_hour=s4.rect_wg*s4.day_hours;
commit;

update FV_OTB_VMS2021_ANALYSIS_S4 s4 set s4.gsa='GSA-22' where s4.gsa='GSA-20' and s4.dcf_area='ARGSAR';
update FV_OTB_VMS2021_ANALYSIS_S4 s4 set s4.gsa='GSA-23' where s4.gsa='GSA-20' and s4.dcf_area='CRETE';
update FV_OTB_VMS2021_ANALYSIS_S4 s4 set s4.gsa='GSA-23' where s4.gsa='GSA-22' and s4.dcf_area='CRETE';
update FV_OTB_VMS2021_ANALYSIS_S4 s4 set s4.gsa='GSA-20' where s4.gsa='GSA-22' and s4.dcf_area='N-ION';
update FV_OTB_VMS2021_ANALYSIS_S4 s4 set s4.gsa='GSA-20' where s4.gsa='GSA-22' and s4.dcf_area='S-ION';
update FV_OTB_VMS2021_ANALYSIS_S4 s4 set s4.gsa='GSA-22' where s4.gsa='GSA-23' and s4.dcf_area='CYCL';
update FV_OTB_VMS2021_ANALYSIS_S4 s4 set s4.gsa='GSA-22' where s4.gsa='GSA-23' and s4.dcf_area='DODEC';
COMMIT;
--- output v1 'OTB'
select s4.gsa,
      s4.dcf_area,
      s4.len_cat,
      substr(s4.mm,1,4) as yy,
      substr(s4.mm,5,2) as mm,
      cast(sum(s4.rect_wg) as number(14,2)) as days_fe,
      cast(sum(s4.rect_hour) as number(14,2)) as fishing_hours
      from FV_OTB_VMS2021_ANALYSIS_S4 s4
    group by s4.gsa, s4.dcf_area, s4.len_cat, substr(s4.mm,1,4), substr(s4.mm,5,2)
    order by s4.gsa, s4.dcf_area, s4.len_cat, substr(s4.mm,1,4), substr(s4.mm,5,2);

--- output v2 'OTB'
select s4.gsa,
      s4.dcf_area,
      s4.len_cat,
      substr(s4.mm,1,4) as yy,
      --substr(s4.mm,5,2) as mm,
      cast(sum(s4.rect_wg) as number(14,2)) as days_fe,
      cast(sum(s4.rect_hour) as number(14,2)) as fishing_hours
      from FV_OTB_VMS2021_ANALYSIS_S4 s4
    group by s4.gsa, s4.dcf_area, s4.len_cat, substr(s4.mm,1,4)
    order by s4.gsa, s4.dcf_area, s4.len_cat, substr(s4.mm,1,4);

--- output v3 'OTB'
select s4.gsa,
      --s4.dcf_area,
      s4.len_cat,
      substr(s4.mm,1,4) as yy,
      --substr(s4.mm,5,2) as mm,
      cast(sum(s4.rect_wg) as number(14,2)) as days_fe,
      cast(sum(s4.rect_hour) as number(14,2)) as fishing_hours
      from FV_OTB_VMS2021_ANALYSIS_S4 s4
    group by s4.gsa, s4.len_cat, substr(s4.mm,1,4)
    order by s4.gsa, s4.len_cat, substr(s4.mm,1,4);

--- output v4 'OTB'
select --s4.gsa,
      --s4.dcf_area,
      substr(s4.mm,1,4) as yy,
      --substr(s4.mm,5,2) as mm,
      cast(sum(s4.rect_wg) as number(14,2)) as days_fe,
      cast(sum(s4.rect_hour) as number(14,2)) as fishing_hours
      from FV_OTB_VMS2021_ANALYSIS_S4 s4
    group by substr(s4.mm,1,4)
    order by substr(s4.mm,1,4);

---- final step: insert estimated OTB fishing effort into l_est_fe

select sum(e1.rect_wg) from FV_OTB_VMS2021_ANALYSIS_S4 e1 where e1.dcf_area is null;
delete FV_OTB_VMS2021_ANALYSIS_S4 e1 where e1.dcf_area is null;
delete FV_OTB_VMS2021_ANALYSIS_S4 e1 where e1.rect_wg is null;
delete FV_OTB_VMS2021_ANALYSIS_S4 e1 where e1.rect_wg=0;
commit;

drop table fv_vms2021_OTB_tmp1;
create table fv_vms2021_OTB_tmp1 as
select 'OTB' as gear,
      to_number(substr(e1.mm,1,4),'9999') as yy,
      to_number(substr(e1.mm,5,2),'99') as mm,
      e1.dcf_area,
      e1.gsa,
      e1.len_cat,
      sum(e1.rect_wg) as fe_days,
      sum(e1.RECT_WG_GT) as fe_days_gt, --sum(e1.days_gt) as fe_days_gt,
      sum(e1.RECT_WG_KW) as fe_days_kw --sum(e1.days_kw) as fe_days_kw
   from FV_OTB_VMS2021_ANALYSIS_S4 e1
   group by to_number(substr(e1.mm,1,4),'9999'), to_number(substr(e1.mm,5,2),'99'), e1.dcf_area, e1.gsa,e1.len_cat
   order by 1, 2, 3, 4;

delete l_est_fe e where e.gear='OTB' and e.yy=2021;
delete l_est_fe e where e.dcf_area is null;
commit;


INSERT INTO l_est_fe (GEAR, YY, MM, DCF_AREA, LEN_CAT, FE_DAYS, FE_DAYS_GT, FE_DAYS_KW, GSA, quarter)
    SELECT t.gear, t.yy, t.mm, t.dcf_area, t.len_cat, sum(t.fe_days), sum(t.fe_days_gt), sum(t.fe_days_kw), t.gsa,
    case when t.mm in (1,2,3) then 1
      else case when t.mm in (4,5,6) then 2
        else case when t.mm in (7,8,9) then 3
          else case when t.mm in (10,11,12) then 4 end end end end
    FROM fv_vms2021_OTB_tmp1 t
    where t.dcf_area is not null
    and t.yy=2021
    group by t.gear, t.yy, t.mm, t.dcf_area, t.len_cat, t.gsa;
COMMIT;

delete l_est_fe fe where fe.fe_days is null or fe.fe_days=0;
COMMIT;

select e.gear,
      e.yy,
      sum(e.fe_days)
  from l_est_fe e
  where e.gear='OTB'
  group by e.gear, e.yy
  order by e.gear, e.yy;

delete fv_vms_analysis_s4 s where substr(s.mm,1,4)='2021';

insert into fv_vms_analysis_s4 (MM, RECT_2KM, DCF_AREA, GSA, LEN_CAT, RECT_WG, RECT_WG_GT, RECT_WG_KW, DAY_HOURS,
RECT_HOUR, GEAR)
      select MM, RECT_2KM, DCF_AREA, GSA, LEN_CAT, RECT_WG, RECT_WG_GT, RECT_WG_KW, DAY_HOURS, RECT_HOUR, 'OTB'
      from FV_OTB_VMS2021_ANALYSIS_S4;
```

```
commit;

--output to GIS
select t.rect_2km,
     t.dcf_area,
     t.gsa,
     cast(sum(t.rect_wg) as number(10,2)) as days_sea,
     cast(sum(t.rect_wg_gt) as number(10,2)) as days_gt,
     cast(sum(t.rect_wg_kw) as number(10,2)) as days_kw
     from FV_OTB_VMS2021_ANALYSIS_S4 t
     where t.gsa in ('GSA-20','GSA-22','GSA-23')
     and (t.rect_wg_gt>0)
     group by t.rect_2km, t.dcf_area, t.gsa
     order by 4 desc;
```

References

Kavadas, S., Carmen, B., Andrea, B., Piera, C., Stefano, C., C Camilla, C., Lorenzo, D-A., Dokos, J., Maina, I., Martinelli, M., Massutí , E., Moranta, J., Parisi , A., Quetglas, A., Russo, T., Santojanni, A., Vasilopoulou V., 2014. Common methodological procedures for analysis of VMS data, including web-based GIS applications related to the spatial extent and intensity of fishing effort. PERSEUS Project. ISBN no: 978-960-9798-14-3

Maina, I., Kavadas, S., Katsanevakis, S., Somarakis, S., Tserpes, G., Georgakarakos S. 2016. A methodological approach to identify fishing grounds: A case study on Greek trawlers. Fisheries Research. 183, 326-339, http://dx.doi.org/10.1016/j.fishres.2016.06.021

APPENDIX III – data request

Med&BS and FDI datacalls

Purpose & requested data sets

During the implementation of the Med&BS RDBFIS, data submitted for the Med&BS and FDI datacalls are needed to be inserted in the Regional Data Base (RDB) to test the functionality of the application in the input, quality control and output processes.

The requested reference period is:

a. 2018-2019 for all FDI tables listed bellow and Med&BS tables catch, discards_length and landings_length.
b. whole time series available at JRC for Med&BS biological data (tables alk, gp, ma, ml, sra, srl)

The data sets are requested from the Med&BS MSs.

Tables with the specifications of the requested data.

Data call	Table name	Specifications
FDI	Table A. Catch summary	*Data to be provided for all landings, both those from métiers selected for biological sampling and otherwise (Fisheries Dependent Information data call 2021 - Annex 1)*
	Table B. Refusal rate	*Member States should only submit data to this table if their sampling design can be considered a probability based vessel selection design (Fisheries Dependent Information data call 2021 - Annex 1)*
	Table G. Effort summary	*Data to be provided for all effort, both that from métiers selected for biological sampling and otherwise (Fisheries Dependent Information data call 2021 - Annex 1)*
	Table H. Landings by rectangle	*Data to be provided for all landings, both those from métiers selected for biological sampling and otherwise6. If it is not possible to submit data at a finer spatial resolution to that required for Table A please do not submit data to table H (Fisheries Dependent Information data call 2021 - Annex 1)*
	Table I. Effort by rectangle	*Data to be provided for all effort, both that from métiers selected for biological sampling and otherwise7. If it is not possible to submit data at a finer spatial resolution to that required for Table G please do not submit data to table I (Fisheries Dependent Information data call 2021 - Annex 1)*
	Table J. Capacity and fleet segment effort	*Capacity and fleet segment effort (Fisheries Dependent Information data call 2021 - Annex 1)*
Med&BS	alk	*Table ALK. Age length key aggregated by age class, sex, species, start-end year, area and country where fish were caught (ANNEX 1, Data call for the STECF EWGs on Mediterranean and Black Sea)*
	catch	*Table A. Catch data fully aggregated (sum) by ID except for mean weight and length (arithmetic mean) in landings and discards at age. (ANNEX 1, Data call for the STECF EWGs on Mediterranean and Black Sea).*
	discards_length	*Table C. Discards data fully aggregated (sum) by ID (ANNEX 1, Data call for the STECF EWGs on Mediterranean and Black Sea).*
	gp	*Table GP. Growth parameters (von Bertalanffy growth parameters & Length-Weight relationship parameters) aggregated by sex, species, start-end year, area and country where fish were caught (ANNEX 1, Data call for the STECF EWGs on Mediterranean and Black Sea).*
	landings_length	*Table B. Landings data fully aggregated (sum) by ID (ANNEX 1, Data call for the STECF EWGs on Mediterranean and Black Sea).*
	ma	*Table MA. Maturity ogives at Age - aggregated by age class, sex, species, start-end year, area and country where fish were caught (ANNEX 1, Data call for the STECF EWGs on Mediterranean and Black Sea).*
	ml	*Table ML. Maturity ogives at Length - aggregated by length class, sex, species, start-end year, area and country where fish were caught (ANNEX 1, Data call for the*

		STECF EWGs on Mediterranean and Black Sea).
	sra	*Table SRA. Sex ratio at age - aggregated by age class, species, start-end year, area and country where fish were caught (ANNEX 1, Data call for the STECF EWGs on Mediterranean and Black Sea).*
	srl	*Table SRL. Sex ratio at length - aggregated by length class, species, start-end year, area and country where fish were caught (ANNEX 1, Data call for the STECF EWGs on Mediterranean and Black Sea).*

GFCM datacalls

Purpose & requested data sets

During the implementation of the Med&BS RDBFIS, data submitted for the GFCM/DCRF datacall are needed to be inserted in the Regional Data Base (RDB) to test the functionality of the application in the input, quality control and output processes.
The requested reference period is 2018-2019 for 11 tables listed bellow:

- GFCM/DCRF II.1 - Landing data (DCRF manual: table II.1)
- GFCM/DCRF II.2 - Catch data per species (DCRF manual: table II.2)
- GFCM/DCRF III - Incidental catch of vulnerable species (DCRF manual: table III.1)
- GFCM/DCRF IV.1 - Description of the vessels ≤ 15 metres length overall (DCRF manual: tables IV.1 and IV.4)
- GFCM/DCRF IV.2 - Description of the vessels over 15 metres length overall (DCRF manual: tables IV.1, IV.2 and IV.4)
- GFCM/DCRF V.1 - Effort data per fleet segment (DCRF manual: table V.1)
- GFCM/DCRF V.2 - Effort data per fishing gear (DCRF manual: table V.2)
- GFCM/DCRF TASK V - FISHING EFFORT V.3 - Effort data CPUE
- GFCM/DCRF VII.2 - Length data: mandatory data for species Group 1, Group 2 and Group 3 (DCRF manual: table VII.2)
- GFCM/DCRF VII.3.1 - Size at first maturity: mandatory data for species Group 1 (DCRF manual: table VII.3.1)
- VII.3.2 - Maturity data: mandatory data for species Group 1 (DCRF manual: table VII.3.2)