

# Package ‘RDBqc’

January 31, 2025

**Title** Quality check functions for RDBFIS

**Version** 0.0.17.25

**Date** 2024-06-25

**Author** Walter Zupa [aut, cre] <zupa@fondazionecoispa.org>,  
Isabella Bitetto [aut,cre] <bitetto@fondazionecoispa.org>,  
Alessandro Mannini [aut] <alessandro.mannini@irbim.cnr.it>,  
Loredana Casciaro [cbt] <casciaro@coispa.eu>,  
Sebastien Alfonso [cbt] <salfonso@coispa.eu>,  
Vasiliki Sgardeli [aut] <vsgard@hcmr.gr>,  
Andrea Pierucci [cbt] <pierucci@coispa.eu>

**Maintainer** Walter Zupa [aut, cre] <zupa@fondazionecoispa.org>

**Description** The package contains functions used to perform a priori and a posteriori quality checks on RDBFIS data.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Depends** R (>= 4.1.0)

**Imports** dplyr, ggplot2, rworldmap, sp, rworldxtra, data.table, grDevices, magrittr, tidyverse, fish-methods, tidyr, gridExtra, scales, outliers, sf, methods, reshape2, cowplot, plyr

**LazyData** true

**Suggests** rmarkdown, knitr, markdown

**VignetteBuilder** knitr

## R topics documented:

aer_catch . . . . .	4
ALK_tab_example . . . . .	5
Catch_tab_example . . . . .	5
catfau_check . . . . .	6
check_age_MEDBS_AR . . . . .	6
check_cs_header . . . . .	7
check_EF_FDI_A . . . . .	8
check_EF_FDI_G . . . . .	8
check_EF_FDI_H . . . . .	9

check_EF_FDI_I . . . . .	10
check_EF_FDI_J . . . . .	10
check_EF_taskII2 . . . . .	11
check_EF_taskIII . . . . .	12
check_EF_taskVII2 . . . . .	12
check_EF_TaskVII31 . . . . .	13
check_EF_TaskVII32 . . . . .	14
check_gfcm_header . . . . .	14
check_150_TaskVII.3.1 . . . . .	15
check_ldf_TaskVII.2 . . . . .	16
check_lengths_MEDBS_AR . . . . .	16
check_lmat_TaskVII.3.2 . . . . .	17
check_lw_TaskVII.2 . . . . .	18
check_lw_TaskVII.3.2 . . . . .	19
check_maturity_MEDBS_AR . . . . .	20
check_minmax150_TaskVII.3.1 . . . . .	21
check_minmax1_TaskVII.2 . . . . .	22
check_n_trips_MEDBS_AR . . . . .	23
check_presence_taskII2 . . . . .	23
check_RD_FDI_A . . . . .	24
check_RD_FDI_G . . . . .	25
check_RD_FDI_H . . . . .	25
check_RD_FDI_I . . . . .	26
check_RD_FDI_J . . . . .	27
check_RD_taskII2 . . . . .	27
check_RD_taskIII . . . . .	28
check_RD_taskVII2 . . . . .	29
check_RD_taskVII31 . . . . .	29
check_RD_TaskVII32 . . . . .	30
check_species_catfau_TaskVII.3.2 . . . . .	31
Check_Tot_Disc . . . . .	32
Check_Tot_Disc_gear . . . . .	33
Check_Tot_Disc_gear_Q . . . . .	34
Check_Tot_Disc_metier . . . . .	35
Check_Tot_Land . . . . .	36
Check_Tot_Land2 . . . . .	37
Check_Tot_Land3 . . . . .	38
Check_Tot_Land_gear . . . . .	39
Check_Tot_Land_gear_Q . . . . .	40
Check_Tot_Land_metier . . . . .	41
check_weights_MEDBS_AR . . . . .	42
circabc . . . . .	43
combination_taskII2 . . . . .	43
data_ex . . . . .	44
data_exampleCL . . . . .	44
Discard_tab_example . . . . .	45
FDI_AER_land_landvalue . . . . .	45
fdi_a_catch . . . . .	46
FDI_checks_spatial_HI . . . . .	47
FDI_check_coord . . . . .	48
FDI_coverage . . . . .	48
FDI_cov_tableA . . . . .	49

FDI_cov_tableG . . . . .	50
FDI_cov_tableJ . . . . .	51
FDI_cross_checks_AG . . . . .	52
FDI_cross_checks_AH . . . . .	53
FDI_cross_checks_IG . . . . .	54
FDI_cross_checks_JG . . . . .	54
FDI_disc_coverage . . . . .	55
FDI_fishdays_cov . . . . .	56
fdi_g_effort . . . . .	56
fdi_h_spatial_land . . . . .	57
fdi_h_spatial_landings . . . . .	57
fdi_i_spatial_effort . . . . .	58
fdi_j_capacity . . . . .	58
FDI_landweight_cov . . . . .	59
FDI_prices_cov . . . . .	59
FDI_prices_not_null . . . . .	60
FDI_vessel_lenth . . . . .	61
FDI_vessel_numbers . . . . .	62
GFCM_check_headers . . . . .	62
GFCM_cov_II2 . . . . .	63
GFCM_cov_task_iii . . . . .	64
GP_tab_example . . . . .	64
GSAlist . . . . .	65
Landing_tab_example . . . . .	65
MA_tab_example . . . . .	66
MEDBSSP . . . . .	66
MEDBS_ALK . . . . .	67
MEDBS_Catch_coverage . . . . .	67
MEDBS_check_disaggregated . . . . .	68
MEDBS_check_duplicates . . . . .	69
MEDBS_check_missing_years . . . . .	70
MEDBS_comp_disc_YQ . . . . .	71
MEDBS_comp_disc_YQ_fishery . . . . .	72
MEDBS_comp_land_Q_VL . . . . .	72
MEDBS_comp_land_Q_VL_fishery . . . . .	73
MEDBS_comp_land_YQ . . . . .	74
MEDBS_comp_land_YQ_fishery . . . . .	75
MEDBS_discard_coverage . . . . .	75
MEDBS_disc_mean_weight . . . . .	76
MEDBS_GP_check . . . . .	77
MEDBS_ks . . . . .	78
MEDBS_Landing_coverage . . . . .	79
MEDBS_land_mean_weight . . . . .	79
MEDBS_lengthclass_0 . . . . .	80
MEDBS_length_ind . . . . .	81
MEDBS_LFD . . . . .	82
MEDBS_LW_check . . . . .	83
MEDBS_MA_check . . . . .	83
MEDBS_ML_check . . . . .	84
MEDBS_plot_discard_ts . . . . .	85
MEDBS_plot_disc_vol . . . . .	86
MEDBS_plot_landing_ts . . . . .	86

MEDBS_plot_land_vol . . . . .	87
MEDBS_SA_check . . . . .	88
MEDBS_SL_check . . . . .	89
MEDBS_SOP . . . . .	89
MEDBS_weight_0 . . . . .	90
MEDBS_weight_minus1 . . . . .	91
MEDBS_yr_missing_length . . . . .	92
minmaxLtaskVII2 . . . . .	93
minmaxLtaskVII31 . . . . .	93
ML_tab_example . . . . .	94
RCG_check_AL . . . . .	94
RCG_check_CL . . . . .	95
RCG_check_LFD . . . . .	96
RCG_check_LFD_comm_cat . . . . .	97
RCG_check_loc . . . . .	97
RCG_check_lw . . . . .	98
RCG_check_mat . . . . .	99
RCG_check_mat_ogive . . . . .	100
RCG_summarize_ind_meas . . . . .	101
RCG_summarize_trips . . . . .	101
SA_tab_example . . . . .	102
sex_mat . . . . .	103
SL_tab_example . . . . .	103
SSPP . . . . .	104
task_ii2 . . . . .	104
task_iii . . . . .	105
task_vii2 . . . . .	105
task_vii31 . . . . .	106
task_vii32 . . . . .	106

**Index** **107**

---

aer_catch	<i>AER catch table</i>
-----------	------------------------

---

**Description**

AER catch table

**Usage**

aer\_catch

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 104 rows and 18 columns.

**Author(s)**

Vasiliki Sgardeli <vsgard@hcmr.gr>

---

ALK_tab_example	<i>ALK table in MED&amp;BS datacall format</i>
-----------------	--

---

**Description**

ALK table in MED&BS datacall format

**Usage**

ALK\_tab\_example

**Format**

An object of class data.frame with 62 rows and 114 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

Catch_tab_example	<i>Catch table in MED&amp;BS datacall format</i>
-------------------	--

---

**Description**

Catch table in MED&BS datacall format

**Usage**

Catch\_tab\_example

**Format**

An object of class data.frame with 57 rows and 171 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

catfau_check	<i>catfau_check in GFCM datacall format</i>
--------------	---

---

**Description**

catfau\_check in GFCM datacall format

**Usage**

catfau\_check

**Format**

An object of class `data.frame` with 355 rows and 5 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

check_age_MEDBS_AR	<i>Cross check of number of age measurements between MED&amp;BS ALK table and AR Table 2.2</i>
--------------------	--

---

**Description**

The function compares the number of age measurements reported in MED & BS data call ALK table with the information reported in the table 2.2 ("Biol variables") of the Annual Report. The user have to define the two source tables as data frames in the ALK and AR parameters. The analysis should be constrained at the selected country level (MS), geographical sub-area GSA and year.

**Usage**

```
check_age_MEDBS_AR(
  ALK,
  AR,
  MS,
  GSA,
  SP,
  year,
  species_list = RDBqc::SSPP,
  OUT = FALSE,
  verbose = TRUE
)
```

**Arguments**

ALK	data frame containing MED&BS ALK table
AR	data frame containing Annual Report data table 2.2
MS	member state code
GSA	Geographical Subarea
SP	Species 3-alpha code
year	Reference year for the analysis
species_list	table of species 3-alpha codes, reporting the MED & BS mandatory species
OUT	Default is FALSE. If set as TRUE tables in csv will be saved in the OUTPUT folder created in the working directory
verbose	boolean. If TRUE a message is printed.

**Value**

A list of three data frames is returned. The first list element contains all the species matching between the two tables; the second list element reports the species reported in Annual Report but not in the MED&BS ALK table; the third list element contains the species reported in MED&BS ALK table but not in the Annual Report table 2.2. Warnings are reported for data of species not included in the MED&BS data call, and for species expected in the Work Plan and not reported in the Annual Report. Errors are reported for species expected in both tables but not present at least in one of the two tables.

**Author(s)**

Walter Zupa [zupa@fondazionecoispa.org](mailto:zupa@fondazionecoispa.org)

---

check_cs_header	<i>Headers check for CS table</i>
-----------------	-----------------------------------

---

**Description**

Headers check for CS table

**Usage**

```
check_cs_header(cs, verbose = FALSE)
```

**Arguments**

cs	RCG sampling table (CS)
verbose	boolean. If TRUE messages are returned

**Value**

The data frame of CS data is returned with the expected format used by the QC functions

**Examples**

```
check_cs_header(data_ex)
```

---

check_EF_FDI_A	<i>Check empty fields in FDI A table</i>
----------------	--

---

**Description**

The function checks the presence of not allowed empty data in the given table.

**Usage**

```
check_EF_FDI_A(data, verbose = TRUE)
```

**Arguments**

data	FDI Task A table
verbose	boolean. If TRUE a message is printed.

**Value**

Two lists are returned by the function. The first list returns the number of NA for each reference column. The second list provides the index of each NA in the reference column.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_EF_FDI_A(fdi_a_catch)
```

---

check_EF_FDI_G	<i>Check empty fields in FDI G table</i>
----------------	--

---

**Description**

The function checks the presence of not allowed empty data in the given table.

**Usage**

```
check_EF_FDI_G(data, verbose = TRUE)
```

**Arguments**

data	FDI Task G table
verbose	boolean. If TRUE a message is printed.



**Value**

Two lists are returned by the function. The first list returns the number of NA for each reference column. The second list provides the index of each NA in the reference column.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_EF_FDI_G(fdi_g_effort)
```

---

check\_EF\_FDI\_H

*Check empty fields in FDI H table*

---

**Description**

The function checks the presence of not allowed empty data in the given table.

**Usage**

```
check_EF_FDI_H(data, verbose = TRUE)
```

**Arguments**

data            FDI Task H table

verbose        boolean. If TRUE a message is printed.

**Value**

Two lists are returned by the function. The first list gives the number of NA for each reference column. The second list gives the index of each NA in the reference column.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_EF_FDI_H(fdi_h_spatial_landings)
```

---

check_EF_FDI_I	<i>Check empty fields in FDI I table</i>
----------------	--

---

**Description**

The function checks the presence of not allowed empty data in the given table.

**Usage**

```
check_EF_FDI_I(data, verbose = TRUE)
```

**Arguments**

data	FDI Task I table
verbose	boolean. If TRUE a message is printed.

**Value**

Two lists are returned by the function. The first list gives the number of NA for each reference column. The second list gives the index of each NA in the reference column.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_EF_FDI_I(fdi_i_spatial_effort)
```

---

check_EF_FDI_J	<i>Check empty fields in FDI J table</i>
----------------	--

---

**Description**

The function checks the presence of not allowed empty data in the given table.

**Usage**

```
check_EF_FDI_J(data, verbose = TRUE)
```

**Arguments**

data	FDI Task J table
verbose	boolean. If TRUE a message is printed.

**Value**

Two lists are returned by the function. The first list gives the number of NA for each reference column. The second list gives the index of each NA in the reference column.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_EF_FDI_J(fdi_j_capacity)
```

---

check\_EF\_taskII2

*Check empty fields in GFCM Task II.2 table*

---

**Description**

The function checks the presence of not allowed empty data in the given table, according to the [GFCM, 2018. GFCM Data Collection Reference Framework \(DCRF\). Version: 20.1](#)

**Usage**

```
check_EF_taskII2(data, verbose = TRUE)
```

**Arguments**

data                   GFCM Task II.2 table

verbose                boolean. If TRUE a message is printed.

**Value**

Two lists are returned by the function. The first list gives the number of NA for each reference column. The second list gives the index of each NA in the reference column.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_EF_taskII2(task_ii2)
```

---

check\_EF\_taskIII      *Check empty fields in GFCM Task III table*

---

### Description

The function checks the presence of not allowed empty data in the given table, according to the [GFCM, 2018. GFCM Data Collection Reference Framework \(DCRF\). Version: 20.1](#)

### Usage

```
check_EF_taskIII(data, verbose = TRUE)
```

### Arguments

data                    GFCM Task III table  
 verbose                boolean. If TRUE a message is printed.

### Value

Two lists are returned by the function. The first list gives the number of NA for each reference column. The second list gives the index of each NA in the reference column.

### Author(s)

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)  
 Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)  
 Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
check_EF_taskIII(task_iii)
```

---

check\_EF\_taskVII2      *Check empty fields in GFCM Task VII.2 table*

---

### Description

The function checks the presence of not allowed empty data in the given table, according to the [GFCM, 2018. GFCM Data Collection Reference Framework \(DCRF\). Version: 20.1](#)

### Usage

```
check_EF_taskVII2(data, verbose = TRUE)
```

### Arguments

data                    GFCM Task VII.2 table  
 verbose                boolean. If TRUE a message is printed.

**Value**

Two lists are returned by the function. The first list gives the number of NA for each reference column. The second list gives the index of each NA in the reference column.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_EF_taskVII2(task_vii2)
```

---

check_EF_TaskVII31	<i>Check empty fields in GFCM Task VII.3.1 table</i>
--------------------	--

---

**Description**

The function checks the presence of not allowed empty data in the given table, according to the [GFCM, 2018. GFCM Data Collection Reference Framework \(DCRF\). Version: 20.1](#)

**Usage**

```
check_EF_TaskVII31(data, verbose = TRUE)
```

**Arguments**

data	GFCM Task VII.3.1 table
verbose	boolean. If TRUE a message is printed.

**Value**

Two lists are returned by the function. The first list gives the number of NA for each reference column. The second list gives the index of each NA in the reference column.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_EF_TaskVII31(task_vii31)
```

---

check\_EF\_TaskVII32      *Check empty fields in GFCM Task VII.3.2 table*

---

### Description

The function checks the presence of not allowed empty data in the given table, according to the [GFCM, 2018. GFCM Data Collection Reference Framework \(DCRF\). Version: 20.1](#)

### Usage

```
check_EF_TaskVII32(data, verbose = TRUE)
```

### Arguments

data	GFCM Task VII.3.2 table
verbose	boolean. If TRUE a message is printed.

### Value

Two lists are returned by the function. The first list gives the number of NA for each reference column. The second list gives the index of each NA in the reference column.

### Author(s)

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)  
 Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)  
 Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
check_EF_TaskVII32(task_vii32)
```

---

check\_gfcm\_header      *Headers check for GFCM tables*

---

### Description

Headers check for GFCM tables

### Usage

```
check_gfcm_header(data, task, verbose = FALSE)
```

### Arguments

data	GFCM table data frame
task	character value reporting the specific GFCM table (task) of the data. Allowed values: "II.2", "III", "VII.2", "VII.3.1", "VII.3.2"
verbose	boolean. If TRUE messages are returned

**Value**

The data frame of the selected GFCM task is returned with the expected format used by the QC functions

**Examples**

```
check_gfcm_header(task_ii2, "TASK_II.2", verbose = FALSE)
```

---

check\_150\_TaskVII.3.1 *Consistency of L50 values in Task VII.3.1 table*

---

**Description**

Function to verify the consistency of L50 reported in the TaskVII.3.1 table by mean of plot by GSA, species and sex

**Usage**

```
check_150_TaskVII.3.1(data, MS, GSA, SP = NA, verbose = FALSE)
```

**Arguments**

data	GFCM Task VII.3.1 table
MS	member state code
GSA	GSA code
SP	species reference code in the three alpha code format
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a plot to visualize the L50 time series by species and sex for the selected MS and GSA

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_150_TaskVII.3.1(task_vii31, MS = "ITA", GSA = "19", SP = "HKE")
```

---

check\_ldf\_TaskVII.2     *Consistency of length frequency distributions*

---

### Description

Function to verify the consistency of the length frequency distributions (LFD) reported in the TaskVII.2 table. The functions generates plots of the LFD by species, source, segment and year.

### Usage

```
check_ldf_TaskVII.2(data, MS, GSA, SP, verbose = TRUE)
```

### Arguments

data	GFCM Task II.2 table
MS	member state code
GSA	GSA code
SP	species reference code in the three alpha code format
verbose	boolean. If TRUE a message is printed.

### Value

The function returns a table with the comparison between min/max lengths observed for each species with theoretical values.

### Author(s)

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
check_ldf_TaskVII.2(task_vii2, MS = "ITA", GSA = "18", SP = "HKE")
```

---

check\_lengths\_MEDBS\_AR

*Cross check of number of lengths between MED&BS catch table and AR Table 2.1*

---

### Description

The function compares the number of length measurements reported in MED & BS data call catch table with the information reported in the table 2.1 ("Stocks") of the Annual Report. The user have to define the two source tables as data frames in the MEDBS and AR parameters. The analysis should be constrained at the selected country level (MS) and at the selected year.



**Usage**

```

check_lengths_MEDBS_AR(
  MEDBS,
  AR,
  MS,
  GSA,
  SP,
  year,
  species_list = RDBqc::SSPP,
  OUT = FALSE,
  verbose = FALSE
)

```

**Arguments**

MEDBS	data frame containing MED&BS catch table
AR	data frame containing Annual Report data
MS	member state code
GSA	Geographical Subarea
SP	Species 3-alpha code
year	Reference year for the analysis
species_list	table of species 3-alpha codes, reporting the MED & BS mandatory species
OUT	Default is FALSE. If set as TRUE tables in csv will be saved in the OUTPUT folder created in the working directory
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a data frame reporting the total number of length measurements related to catches, landings and discards conducted in the selected country and year. MED & BS data will be reported as sum of the measurements by year and sum of the trips reported by quarter in the year.

**Author(s)**

Walter Zupa [zupa@fondazionecoispa.org](mailto:zupa@fondazionecoispa.org)

---

check\_lmat\_TaskVII.3.2

*Plot of the maturity stages per length for each sex and species*

---

**Description**

Function to plot the lengths at maturity stages by species and sex to easily identify outliers.

**Usage**

```

check_lmat_TaskVII.3.2(data, MS, GSA, SP, verbose = TRUE)

```

**Arguments**

data	GFCM Task VII.3.2 table
MS	member state code
GSA	GSA code
SP	species reference code in the three alpha code format
verbose	boolean. If TRUE a message is printed.

**Value**

The function return a plot of the maturity stages per length and sex per species.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_lmat_TaskVII.3.2(task_vii32, MS = "ITA", GSA = "18", SP = "CTC")
```

---

check\_lw\_TaskVII.2     *Plot of the length-weight relationship*

---

**Description**

Function to check the consistency of length-weight relationship in the GFCM Task VII.2 table for the selected species.

**Usage**

```
check_lw_TaskVII.2(data, MS, GSA, SP, verbose = TRUE)
```

**Arguments**

data	GFCM Task VII.2 table
MS	country code
GSA	GSA code
SP	species reference code in the three alpha code format
verbose	boolean value to obtain further explanation messages from the function

**Value**

The function return a plot of the length weight relationship for the selected species.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_lw_TaskVII.2(task_vii2, MS = "ITA", GSA = "18", SP = "BOG")
```

---

check\_lw\_TaskVII.3.2 *Plot of the length-weight relationship by sex*

---

**Description**

Function to check the consistency of length-weight relationship in the GFCM Task VII.3.2 table by sex for the selected species.

**Usage**

```
check_lw_TaskVII.3.2(data, MS, GSA, SP, verbose = TRUE)
```

**Arguments**

data	GFCM Task VII.3.2 table
MS	country code
GSA	GSA code
SP	species reference code in the three alpha code format
verbose	boolean value to obtain further explanation messages from the function

**Value**

The function return a plot of the length-weight relationship by sex for the selected species.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_lw_TaskVII.3.2(task_vii32, MS = "ITA", GSA = "18", SP = "CTC")
```

---

check\_maturity\_MEDBS\_AR

*Cross check of number of maturity measurements between MED&BS ML table and AR Table 2.2*

---

### Description

The function compares the number of maturity measurements reported in MED & BS data call ML table with the information reported in the table 2.2 ("Biol variables") of the Annual Report. The user have to define the two source tables as data frames in the ML and AR parameters. The analysis should be constrained at the selected country level (MS), geographical sub-area GSA and year. When GSA and SP parameter are NA, the analysis is conducted on all the GSAs and species included in the data frame for the selected country.

### Usage

```
check_maturity_MEDBS_AR(
  ML,
  AR,
  MS,
  GSA,
  SP,
  year,
  species_list = RDBqc::SSPP,
  OUT = FALSE,
  verbose = TRUE
)
```

### Arguments

ML	data frame containing MED&BS ML table
AR	data frame containing Annual Report data table 2.2
MS	member state code
GSA	Geographical Subarea
SP	Species 3-alpha code
year	Reference year for the analysis
species_list	table of species 3-alpha codes, reporting the MED & BS mandatory species
OUT	Default is FALSE. If set as TRUE tables in csv will be saved in the OUTPUT folder created in the working directory
verbose	boolean. If TRUE a message is printed.

### Value

A list of three data frames is returned. The first list element contains all the species matching between the two tables; the second list element reports the species reported in Annual Report but not in the MED&BS ML table; the third list element contains the species reported in MED&BS ML table but not in the Annual Report table 2.2. Warnings are reported for data of species not included in the MED&BS data call, and for species expected in the Work Plan and not reported in the Annual Report. Errors are reported for species expected in both tables but not present at least in one of the two tables.

**Author(s)**

Walter Zupa [zupa@fondazionecoispa.org](mailto:zupa@fondazionecoispa.org)

---

check\_minmax150\_TaskVII.3.1

*Comparison between min/max L50 observed for each species and sex with theoretical values*

---

**Description**

Function to verify the consistency of L50 reported in the TaskVII.3.1 table with the theoretical values reported in the minmaxLtaskVII31 table. The function allows to identify the records in which the observed L50 are greater or lower than the expected ones.

**Usage**

```
check_minmax150_TaskVII.3.1(data, tab_L50, MS, GSA)
```

**Arguments**

data	GFCM Task VII.3.1 table
tab_L50	Theoretical values of min/max L50 for each species and sex
MS	member state code
GSA	GSA code

**Value**

The function returns a table with the comparison between min/max L50 observed for each species and sex with theoretical values.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_minmax150_TaskVII.3.1(task_vii31, minmaxLtaskVII31, MS = "ITA", GSA = "19")
```

---

check\_minmax1\_TaskVII.2

*Comparison between min/max observed for each species with theoretical values*

---

### Description

Function to verify the consistency of the lengths reported in the TaskVII.2 table with the theoretical values reported in the minmaxLtaskVII2 table. The function allows to identify the records in which the observed lengths are greater or lower than the expected ones.

### Usage

```
check_minmax1_TaskVII.2(data, MS, GSA, SP, verbose = TRUE)
```

### Arguments

data	GFCM Task II.2 table
MS	member state code
GSA	GSA code
SP	species reference code in the three alpha code format
verbose	boolean. If TRUE a message is printed.

### Value

The function returns a table with the comparison between min/max lengths observed for each species with theoretical values.

### Author(s)

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
check_minmax1_TaskVII.2(task_vii2, minmaxLtaskVII2, MS = "ITA", GSA = "18")
```

---

 check\_n\_trips\_MEDBS\_AR

*Cross check of number of trips between MED&BS catch table and AR Table 2.5*

---

### Description

The function compares the number of trips reported in MED & BS data call catch table with the information reported in the table 2.5 ("Sampling plan biol") of the Annual Report. The user have to define the two source tables as data frames in the MEDBS and AR parameters. The analysis should be constrained at the selected country level (MS) and at the selected year.

### Usage

```
check_n_trips_MEDBS_AR(MEDBS, AR, MS, year, OUT = FALSE, verbose = FALSE)
```

### Arguments

MEDBS	data frame containing MED&BS catch table
AR	data frame containing Annual Report data
MS	member state code
year	Reference year for the analysis
OUT	Default is FALSE. If set as TRUE tables in csv will be saved in the OUTPUT folder created in the working directory
verbose	boolean. If TRUE a message is printed.

### Value

The function returns a data frame reporting the total number of trips related to catches, landings and discards conducted in the selected country and year. MED & BS data will be reported both as sum of the trips by year and as sum of the trips reported by quarter in the year.

### Author(s)

Walter Zupa [zupa@fondazionecoispa.org](mailto:zupa@fondazionecoispa.org)

---

check\_presence\_taskII2

*Check of missing combination GSA/Fleet segment per year*

---

### Description

Function to verify the completeness of the GSA/Fleet segments in Task II.2 table, as reported in the combination\_taskII2 table.

### Usage

```
check_presence_taskII2(data1, data2, MS, GSA)
```

**Arguments**

data1	GFCM Task II.2 table
data2	List of combination of the expected GSA/Fleet segments per year for Task II.2 table
MS	member state code
GSA	GSA code

**Value**

The function returns a list of missing combinations GSA/Fleet segment per year.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_presence_taskII2(task_ii2, combination_taskII2, MS = "ITA", GSA = "18")
```

---

check_RD_FDI_A	<i>Check duplicated records in FDI A table</i>
----------------	--

---

**Description**

The function check the presence of duplicated records. In particular, it checks whether the combination of the first 19 columns generates duplicate records.

**Usage**

```
check_RD_FDI_A(data, verbose = TRUE)
```

**Arguments**

data	FDI Task A table
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns the indices of the duplicated rows, checking the unique combinations of the first 19 columns of the FDI A table.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_RD_FDI_A(fdi_a_catch)
```



---

check_RD_FDI_G	<i>Check duplicated records in FDI G table</i>
----------------	--

---

**Description**

The function check the presence of duplicated records. In particular, it checks whether the combination of the first 15 columns generates duplicate records.

**Usage**

```
check_RD_FDI_G(data, verbose = TRUE)
```

**Arguments**

data	FDI Task G table
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns the indices of the duplicated rows, checking the unique combinations of the first 15 columns of the FDI G table.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_RD_FDI_G(fdi_g_effort)
```

---

check_RD_FDI_H	<i>Check duplicated records in FDI H table</i>
----------------	--

---

**Description**

The function check the presence of duplicated records. In particular, it checks whether the combination of the first 15 columns generates duplicate records.

**Usage**

```
check_RD_FDI_H(data, verbose = TRUE)
```

**Arguments**

data	FDI Task H table
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns the indices of the duplicated rows, checking the unique combinations of the first 20 columns of the FDI G table.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_RD_FDI_H(fdi_h_spatial_landings)
```

---

check\_RD\_FDI\_I

*Check duplicated records in FDI I table*

---

**Description**

The function check the presence of duplicated records. In particular, it checks whether the combination of the first 15 columns generates duplicate records.

**Usage**

```
check_RD_FDI_I(data, verbose = TRUE)
```

**Arguments**

data            FDI Task I table

verbose        boolean. If TRUE a message is printed.

**Value**

The function returns the indices of the duplicated rows, checking the unique combinations of the first 19 columns of the FDI G table.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_RD_FDI_I(fdi_i_spatial_effort)
```

---

check_RD_FDI_J	<i>Check duplicated records in FDI J table</i>
----------------	--

---

**Description**

The function check the presence of duplicated records. In particular, it checks whether the combination of the first 15 columns generates duplicate records.

**Usage**

```
check_RD_FDI_J(data, verbose = TRUE)
```

**Arguments**

data	FDI Task J table
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns the indices of the duplicated rows, checking the unique combinations of the first 7 columns of the FDI G table.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_RD_FDI_J(fdi_j_capacity)
```

---

check_RD_taskII2	<i>Check duplicated records in GFCM Task II.2 table</i>
------------------	---

---

**Description**

The function check the presence of duplicated records. In particular, it checks whether the combination of the first 5 columns generates duplicate records.

**Usage**

```
check_RD_taskII2(data, verbose = TRUE)
```

**Arguments**

data	GFCM Task II.2 table
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns the indices of the duplicated rows, checking the unique combinations of the first 5 columns of the Task II.2 table.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_RD_taskII2(task_ii2)
```

---

check\_RD\_taskIII

*Check duplicated records in GFCM Task III table*

---

**Description**

The function check the presence of duplicated records. In particular, it checks whether the combination of the first 10 columns generates duplicate records.

**Usage**

```
check_RD_taskIII(data, verbose = TRUE)
```

**Arguments**

data                   GFCM Task III table

verbose                boolean. If TRUE a message is printed.

**Value**

The function returns the indices of the duplicated rows, checking the unique combinations of the first 10 columns of the Task Task III table.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_RD_taskIII(task_iii)
```

---

check_RD_taskVII2	<i>Check duplicated records in GFCM Task VII.2 table</i>
-------------------	--

---

**Description**

The function check the presence of duplicated records. In particular, it checks whether the combination of the first 9 columns generates duplicate records.

**Usage**

```
check_RD_taskVII2(data, verbose = TRUE)
```

**Arguments**

data	GFCM Task VII.2 table
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns the indices of the duplicated rows, checking the unique combinations of the first 9 columns of the Task Task VII.2 table.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_RD_taskVII2(task_vii2)
```

---

check_RD_taskVII31	<i>Check duplicated records in GFCM Task VII.3.1 table</i>
--------------------	--

---

**Description**

The function check the presence of duplicated records. In particular, it checks whether the combination of the first 5 columns generates duplicate records.

**Usage**

```
check_RD_taskVII31(data, verbose = TRUE)
```

**Arguments**

data	GFCM Task VII.3.1 table
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns the indices of the duplicated rows, checking the unique combinations of the first 5 columns of the Task Task VII.3.1 table.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_RD_taskVII31(task_vii31)
```

---

check_RD_TaskVII32	<i>Check duplicated records in GFCM Task VII.3.2 table</i>
--------------------	--

---

**Description**

The function check the presence of duplicated records. In particular, it checks whether the combination of the first 10 columns generates duplicate records.

**Usage**

```
check_RD_TaskVII32(data, verbose = TRUE)
```

**Arguments**

data	GFCM Task VII.3.2 table
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns the indices of the duplicated rows, checking the unique combinations of the first 10 columns of the Task Task VII.3.2 table.

**Author(s)**

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
check_RD_TaskVII32(task_vii32)
```

---

 check\_species\_catfau\_TaskVII.3.2

*Check mismatching species/Catfau and Sex per maturity stages for Task VII.3.2 table*

---

### Description

Function to check the correct codification of faunistic category according to species and sex in Task VII.3.2 table.

### Usage

```
check_species_catfau_TaskVII.3.2(
  data,
  species,
  matsex,
  MS,
  GSA,
  SP,
  verbose = FALSE
)
```

### Arguments

data	GFCM Task VII.3.2 table
species	List of combination of species/faunistic category for Task VII.3.2 table
matsex	List of combination of sex/maturity stages for Task VII.3.2 table
MS	member state code
GSA	GSA code
SP	species code
verbose	boolean. If TRUE messages are returned

### Value

Two vectors are returned by the function. The first provides the list of mismatching combination of species/faunistic categories. The second vector provides the list of mismatching combination of sex/maturity stages. Furthermore, a plot of the length distribution by sex and maturity is returned for the selected species.

### Author(s)

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

Sebastien Alfonso [salfonso@coispa.eu](mailto:salfonso@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
check_species_catfau_TaskVII.3.2(task_vii32, catfau_check, sex_mat,
  MS = "ITA", GSA = "18", SP = "HKE"
)
```

---

Check_Tot_Disc	<i>Comparing total discards in weight between MEDBS and FDI EU Data Calls</i>
----------------	---

---

### Description

The function compares the discards in weight values reported in the MEDBS discards by length table with the one reported in the FDI Table A catch table. The analysis is done at MS, GSA and species level

### Usage

```
Check_Tot_Disc(data, data1, MS, GSA, SP, MEDBSSP, verbose = FALSE, OUT = FALSE)
```

### Arguments

data	data frame containing MEDBS discards by length table
data1	data frame containing FDI Table A Catch data
MS	member state 3 alpha code
GSA	FAO Geographical Sub Area code (GSA9, GSA29, etc)
SP	ASFIS FAO 3alpha code of the choosen species
MEDBSSP	look up table in which requested MEDBS Data Call species (see MEDBS data call Annex I) are listed
verbose	boolean. If TRUE a message is printed
OUT	Default is FALSE. If set as TRUE plots and tables in csv will be saved in the OUTPUT folder created in the working directory

### Value

The function returns a plot comparison by year of the discards in weight provided through the MEDBS and FDI Data Calls. Moreover a csv file with the actual values will be created. Plot, csv and rds output files will be saved in the dedicated folders in OUTPUT folder

### Author(s)

Alessandro Mannini [alessandro.mannini@irbim.cnr.it](mailto:alessandro.mannini@irbim.cnr.it)

### Examples

```
# Check_Tot_Disc(MEDBS,FDI,"ITA","GSA10","HKE",MEDBSSP)
# The function works by one country, subarea and species each.
# It is not possible assign more country, subarea or species.
```



---

Check\_Tot\_Disc\_gear      *Comparing total discards in weight between MEDBS and FDI EU Data Calls by Country, GSA, Species and Gear*

---

### Description

The function compares the discards in weight values reported in the MEDBS discards by length table with the one reported in the FDI Table A catch table. The analysis is done at MS, GSA, species and gear level

### Usage

```
Check_Tot_Disc_gear(
  data,
  data1,
  MS,
  GSA,
  SP,
  MEDBSSP,
  verbose = TRUE,
  OUT = FALSE
)
```

### Arguments

data	data frame containing MEDBS discards by length table
data1	data frame containing FDI Table A Catch data
MS	member state 3 alpha code
GSA	FAO Geographical Sub Area code (GSA9, GSA29, etc)
SP	ASFIS FAO 3alpha code of the choosen species
MEDBSSP	look up table in which requested MEDBS Data Call species (see MEDBS data call Annex I) are listed
verbose	boolean. If TRUE a message is printed
OUT	Default is FALSE. If set as TRUE plots and tables in csv will be saved in the OUTPUT folder created in the working directory

### Value

The function returns plots comparison by year of the discards in weight provided through the MEDBS and FDI Data Calls. Moreover a csv file with the actual values will be created. Plot, csv and rds output files will be saved in the dedicated folders in OUTPUT folder

### Author(s)

Alessandro Mannini [alessandro.mannini@irbim.cnr.it](mailto:alessandro.mannini@irbim.cnr.it)

### Examples

```
# Check_Tot_Disc_gear(MEDBS,FDI,"ITA","GSA10","HKE",MEDBSSP)
# The function works by one country, subarea and species each.
# It is not possible assign more country, subarea or species.
```

---

Check\_Tot\_Disc\_gear\_Q *Comparing total discards in weight between MEDBS and FDI EU Data Calls by Country, GSA, Species and Gear*

---

### Description

The function compares the discards in weight values reported in the MEDBS discards by length table with the one reported in the FDI Table A catch table. The analysis is done at MS, GSA, species and gear level

### Usage

```
Check_Tot_Disc_gear_Q(
  data,
  data1,
  MS,
  GSA,
  SP,
  MEDBSSP,
  verbose = TRUE,
  OUT = FALSE
)
```

### Arguments

data	data frame containing MEDBS discards by length table
data1	data frame containing FDI Table A Catch data
MS	member state 3 alpha code
GSA	FAO Geographical Sub Area code (GSA9, GSA29, etc)
SP	ASFIS FAO 3alpha code of the choosen species
MEDBSSP	look up table in which requested MEDBS Data Call species (see MEDBS data call Annex I) are listed
verbose	boolean. If TRUE a message is printed
OUT	Default is FALSE. If set as TRUE plots and tables in csv will be saved in the OUTPUT folder created in the working directory

### Value

The function returns plots comparison by year of the discards in weight provided through the MEDBS and FDI Data Calls. Moreover a csv file with the actual values will be created. Plot, csv and rds output files will be saved in the dedicated folders in OUTPUT folder

### Author(s)

Alessandro Mannini [alessandro.mannini@irbim.cnr.it](mailto:alessandro.mannini@irbim.cnr.it)

### Examples

```
# Check_Tot_Disc_gear_Q (MEDBS,FDI,"ITA","GSA10","HKE",MEDBSSP,verbose=TRUE,OUT=FALSE)
# The function works by one country, subarea and species each.
# It is not possible assign more country, subarea or species.
```

---

Check\_Tot\_Disc\_metier *Comparing total discards in weight between MEDBS and FDI EU Data Calls by Country, GSA, Species and Metier*

---

### Description

The function compares the discards in weight values reported in the MEDBS discards by length table with the one reported in the FDI Table A catch table. The analysis is done at MS, GSA, species and metier level

### Usage

```
Check_Tot_Disc_metier(
  data,
  data1,
  MS,
  GSA,
  SP,
  MEDBSSP,
  verbose = TRUE,
  OUT = FALSE
)
```

### Arguments

data	data frame containing MEDBS discards by length table
data1	data frame containing FDI Table A Catch data
MS	member state 3 alpha code
GSA	FAO Geographical Sub Area code (GSA9, GSA29, etc)
SP	ASFIS FAO 3alpha code of the choosen species
MEDBSSP	look up table in which requested MEDBS Data Call species (see MEDBS data call Annex I) are listed
verbose	boolean. If TRUE a message is printed
OUT	Default is FALSE. If set as TRUE plots and tables in csv will be saved in the OUTPUT folder created in the working directory

### Value

The function returns plots comparison by year of the discards in weight provided through the MEDBS and FDI Data Calls. Moreover a csv file with the actual values will be created. Plot, csv and rds output files will be saved in the dedicated folders in OUTPUT folder

### Author(s)

Alessandro Mannini [alessandro.mannini@irbim.cnr.it](mailto:alessandro.mannini@irbim.cnr.it)

### Examples

```
# Check_Tot_Disc_metier(MEDBS,FDI,"ITA","GSA10","HKE",MEDBSSP,verbose=TRUE,OUT=FALSE)
# The function works by one country, subarea and species each.
# It is not possible assign more country, subarea or species.
```

---

Check_Tot_Land	<i>Comparing total landings in weight between MEDBS and FDI EU Data Calls</i>
----------------	---

---

### Description

The function compares the landings in weight values reported in the MEDBS landings by length table with the one reported in the FDI Table A catch table. The analysis is done at MS, GSA and species level

### Usage

```
Check_Tot_Land(data, data1, MS, GSA, SP, MEDBSSP, verbose = FALSE, OUT = FALSE)
```

### Arguments

data	data frame containing MEDBS landings by length table
data1	data frame containing FDI Table A Catch data
MS	member state 3 alpha code
GSA	FAO Geographical Sub Area code (GSA9, GSA29, etc)
SP	ASFIS FAO 3alpha code of the choosen species
MEDBSSP	look up table in which requested MEDBS Data Call species (see MEDBS data call Annex I) are listed
verbose	boolean. If TRUE a message is printed
OUT	Default is FALSE. If set as TRUE plots and tables in csv will be saved in the OUTPUT folder created in the working directory

### Value

The function returns a plot comparison by year of the landings in weight provided through the MEDBS and FDI Data Calls. Moreover a csv file with the actual values will be created. Plot, csv and rds output files will be saved in the dedicated folders in OUTPUT folder

### Author(s)

Alessandro Mannini [alessandro.mannini@irbim.cnr.it](mailto:alessandro.mannini@irbim.cnr.it)

### Examples

```
# Check_Tot_Land(MEDBS,FDI,"ITA","GSA10","HKE",MEDBSSP,verbose=TRUE,OUT=FALSE)

# The function works by one country, subarea and species each.
# It is not possible assign more country, subare or species.
```

---

Check_Tot_Land2	<i>Comparing total landings in weight between MEDBS, FDI and AER EU Data Calls</i>
-----------------	--

---

### Description

The function compares the landings in weight values reported in the MEDBS landings by length table with the ones reported in the FDI Table A catch table and AER data (<https://stecf.jrc.ec.europa.eu/reports/economic>). The analysis is done at MS, GSA and species level

### Usage

```
Check_Tot_Land2(
  data,
  data1,
  data2,
  MS,
  GSA,
  SP,
  MEDBSSP,
  verbose = TRUE,
  OUT = FALSE
)
```

### Arguments

data	data frame containing MEDBS landings by length table
data1	data frame containing FDI Table A Catch data
data2	data frame containing AER landings data
MS	member state 3 alpha code
GSA	FAO Geographical Sub Area code (GSA9, GSA29, etc)
SP	ASFIS FAO 3alpha code of the choosen species
MEDBSSP	look up table in which requested MEDBS Data Call species (see MEDBS data call Annex I) are listed
verbose	boolean. If TRUE a message is printed
OUT	Default is FALSE. If set as TRUE plots and tables in csv will be saved in the OUTPUT folder created in the working directory

### Value

The function returns a plot comparison by year of the landings in weight provided through the MEDBS, FDI and AER Data Calls. Moreover a csv file with the actual values will be created. Plot, csv and rds output files will be saved in the dedicated folders in OUTPUT folder

### Author(s)

Alessandro Mannini [alessandro.mannini@irbim.cnr.it](mailto:alessandro.mannini@irbim.cnr.it)

**Examples**

```
# Check_Tot_Land2(MEDBS,FDI,AER,"ITA","GSA10","HKE",MEDBSSP,verbose=TRUE,OUT=FALSE)

# The function works by one country, subarea and species each.
# It is not possible assign more country, subarea or species.
```

---

Check_Tot_Land3	<i>Comparing total landings in weight between MEDBS, FDI and AER EU Data Calls</i>
-----------------	--

---

**Description**

The function compares the landings in weight values reported in the MEDBS landings by length table with the ones reported in the FDI Table A catch table and AER input data (map\_fsfao.xlsx) . The analysis is done at MS, GSA and species level

**Usage**

```
Check_Tot_Land3(
  data,
  data1,
  data2,
  MS,
  GSA,
  SP,
  MEDBSSP,
  verbose = TRUE,
  OUT = FALSE
)
```

**Arguments**

data	data frame containing MEDBS landings by length table
data1	data frame containing FDI Table A Catch data
data2	data frame containing AER landings data
MS	member state 3 alpha code
GSA	FAO Geographical Sub Area code (GSA9, GSA29, etc)
SP	ASFIS FAO 3alpha code of the choosen species
MEDBSSP	look up table in which requested MEDBS Data Call species (see MEDBS data call Annex I) are listed
verbose	boolean. If TRUE a message is printed
OUT	Default is FALSE. If set as TRUE plots and tables in csv will be saved in the OUTPUT folder created in the working directory

**Value**

The function returns a plot comparison by year of the landings in weight provided through the MEDBS,FDI and AER Data Calls. Moreover a csv file with the actual values will be created. Plot, csv and rds output files will be saved in the dedicated folders in OUTPUT folder

**Author(s)**

Alessandro Mannini [alessandro.mannini@irbim.cnr.it](mailto:alessandro.mannini@irbim.cnr.it)

**Examples**

```
# Check_Tot_Land3(MEDBS, FDI, AER, "ITA", "GSA10", "HKE", MEDBSSP, verbose = TRUE, OUT = FALSE)

# The function works by one country, subarea and species each.
# It is not possible assign more country, subare or species.
```

---

Check\_Tot\_Land\_gear     *Comparing total landings in weight between MEDBS and FDI EU Data Calls by Country, GSA, Species and Gear*

---

**Description**

The function compares the landings in weight values reported in the MEDBS landings by length table with the one reported in the FDI Table A catch table. The analysis is done at MS, GSA, species and gear level

**Usage**

```
Check_Tot_Land_gear(
  data,
  data1,
  MS,
  GSA,
  SP,
  MEDBSSP,
  verbose = TRUE,
  OUT = FALSE
)
```

**Arguments**

data	data frame containing MEDBS landings by length table
data1	data frame containing FDI Table A Catch data
MS	member state 3 alpha code
GSA	FAO Geographical Sub Area code (GSA9, GSA29, etc)
SP	ASFIS FAO 3alpha code of the choosen species
MEDBSSP	look up table in which requested MEDBS Data Call species (see MEDBS data call Annex I) are listed
verbose	boolean. If TRUE a message is printed
OUT	Default is FALSE. If set as TRUE plots and tables in csv will be saved in the OUTPUT folder created in the working directory

**Value**

The function returns plots comparison by year of the landings in weight provided through the MEDBS and FDI Data Calls. Moreover a csv file with the actual values will be created. Plot, csv and rds output files will be saved in the dedicated folders in OUTPUT folder

**Author(s)**

Alessandro Mannini [alessandro.mannini@irbim.cnr.it](mailto:alessandro.mannini@irbim.cnr.it)

**Examples**

```
# Check_Tot_Land_gear(MEDBS,FDI,"ITA","GSA10","HKE",MEDBSSP)

# The function works by one country, subarea and species each.
# It is not possible assign more country, subarea or species.
```

---

Check\_Tot\_Land\_gear\_Q *Comparing total landings in weight between MEDBS and FDI EU Data Calls by Country, GSA, Species and Gear*

---

**Description**

The function compares the landings in weight values reported in the MEDBS landings by length table with the one reported in the FDI Table A catch table. The analysis is done at MS, GSA, species and gear level

**Usage**

```
Check_Tot_Land_gear_Q(
  data,
  data1,
  MS,
  GSA,
  SP,
  MEDBSSP,
  verbose = TRUE,
  OUT = FALSE
)
```

**Arguments**

data	data frame containing MEDBS landings by length table
data1	data frame containing FDI Table A Catch data
MS	member state 3 alpha code
GSA	FAO Geographical Sub Area code (GSA9, GSA29, etc)
SP	ASFIS FAO 3alpha code of the choosen species
MEDBSSP	look up table in which requested MEDBS Data Call species (see MEDBS data call Annex I) are listed
verbose	boolean. If TRUE a message is printed
OUT	Default is FALSE. If set as TRUE plots and tables in csv will be saved in the OUTPUT folder created in the working directory

**Value**

The function returns plots comparison by year of the landings in weight provided through the MEDBS and FDI Data Calls. Moreover a csv file with the actual values will be created. Plot, csv and rds output files will be saved in the dedicated folders in OUTPUT folder



**Author(s)**

Alessandro Mannini [alessandro.mannini@irbim.cnr.it](mailto:alessandro.mannini@irbim.cnr.it)

**Examples**

```
# Check_Tot_Land_gear_Q (MEDBS,FDI,"ITA","GSA10","HKE",MEDBSSP)

# The function works by one country, subarea and species each.
# It is not possible assign more country, subarea or species.
```

---

Check\_Tot\_Land\_metier *Comparing total landings in weight between MEDBS and FDI EU Data Calls by Country, GSA, Species and Metier*

---

**Description**

The function compares the landings in weight values reported in the MEDBS landings by length table with the one reported in the FDI Table A catch table. The analysis is done at MS, GSA, species and metier level

**Usage**

```
Check_Tot_Land_metier(
  data,
  data1,
  MS,
  GSA,
  SP,
  MEDBSSP,
  verbose = TRUE,
  OUT = FALSE
)
```

**Arguments**

data	data frame containing MEDBS landings by length table
data1	data frame containing FDI Table A Catch data
MS	member state 3 alpha code
GSA	FAO Geographical Sub Area code (GSA9, GSA29, etc)
SP	ASFIS FAO 3alpha code of the choosen species
MEDBSSP	look up table in which requested MEDBS Data Call species (see MEDBS data call Annex I) are listed
verbose	boolean. If TRUE a message is printed
OUT	Default is FALSE. If set as TRUE plots and tables in csv will be saved in the OUTPUT folder created in the working directory

**Value**

The function returns plots comparison by year of the landings in weight provided through the MEDBS and FDI Data Calls. Moreover a csv file with the actual values will be created. Plot, csv and rds output files will be saved in the dedicated folders in OUTPUT folder

**Author(s)**

Alessandro Mannini [alessandro.mannini@irbim.cnr.it](mailto:alessandro.mannini@irbim.cnr.it)

**Examples**

```
# Check_Tot_Land_metier(MEDBS, FDI, "ITA", "GSA10", "HKE", MEDBSSP, verbose = TRUE, OUT = FALSE)

# The function works by one country, subarea and species each.
# It is not possible assign more country, subarea or species.
```

---

check\_weights\_MEDBS\_AR

*Cross check of number of weight measurements between MED&BS GP table and AR Table 2.2*

---

**Description**

The function compares the number of weight measurements reported in GP table of MED & BS data call with the information reported in the table 2.2 ("Biol variables") of the Annual Report. The user have to define the two source tables as data frames in the GP and AR parameters. The analysis should be constrained at the selected country level (MS), geographical sub-area GSA and year.

**Usage**

```
check_weights_MEDBS_AR(
  GP,
  AR,
  MS,
  GSA,
  SP,
  year,
  species_list = RDBqc::SSPP,
  OUT = FALSE,
  verbose = TRUE
)
```

**Arguments**

GP	data frame containing MED&BS GP table
AR	data frame containing Annual Report data table 2.2
MS	member state code
GSA	Geographical Subarea
SP	Species 3-alpha code
year	Reference year for the analysis
species_list	table of species 3-alpha codes, reporting the MED & BS mandatory species
OUT	Default is FALSE. If set as TRUE tables in csv will be saved in the OUTPUT folder created in the working directory
verbose	boolean. If TRUE a message is printed.

**Value**

A list of three data frames is returned. The first list element contains all the species matching between the two tables; the second list element reports the species reported in Annual Report but not in the MED&BS GP table; the third list element contains the species reported in MED&BS GP table but not in the Annual Report table 2.2. Warnings are reported for data of species not included in the MED&BS data call, and for species expected in the Work Plan and not reported in the Annual Report. Errors are reported for species expected in both tables but not present at least in one of the two tables.

**Author(s)**

Walter Zupa [zupa@fondazionecoispa.org](mailto:zupa@fondazionecoispa.org)

---

circabc

*Ports coordinates according to codification CIRCABC*

---

**Description**

Ports coordinates according to codification CIRCABC

**Usage**

circabc

**Format**

An object of class `data.frame` with 24335 rows and 5 columns.

**Author(s)**

CIRCABC

---

combination\_taskII2

*combination\_taskII2 in GFCM datacall format*

---

**Description**

combination\_taskII2 in GFCM datacall format

**Usage**

combination\_taskII2

**Format**

An object of class `data.frame` with 60 rows and 4 columns.

**Author(s)**

Isabella Bitetto <[bitetto@coispa.it](mailto:bitetto@coispa.it)>

---

data_ex	<i>RCG CS example</i>
---------	-----------------------

---

**Description**

RCG CS example

**Usage**

data\_ex

**Format**

An object of class `data.frame` with 9099 rows and 40 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

data_exampleCL	<i>RCG CL example</i>
----------------	-----------------------

---

**Description**

RCG CL example

**Usage**

data\_exampleCL

**Format**

An object of class `data.frame` with 35 rows and 11 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

Discard\_tab\_example     *Discard table in MED&BS datacall format*

---

**Description**

Discard table in MED&BS datacall format

**Usage**

Discard\_tab\_example

**Format**

An object of class data.frame with 21 rows and 114 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

FDI\_AER\_land\_landvalue

*Cross check of landings or landings' value between the FDI and AER databases*

---

**Description**

The function compares landings (var='landings') or landings' value (var='value') by SP and YEAR between the FDI and AER databases for the given MS. By default, the comparison is done by GSA, GEAR and METIER. Optionally, the user can select a higher level of aggregation (e.g. 'MS', 'GSA', etc.) (see level variable)

**Usage**

```
FDI_AER_land_landvalue(  
  FDI,  
  AER,  
  var = "landings",  
  MS,  
  level = "GSA",  
  YEAR = NA,  
  GSA = NA,  
  SP = NA,  
  OUT = FALSE,  
  verbose = FALSE  
)
```

**Arguments**

FDI	(mandatory) data frame containing FDI catch data (i.e. TABLE_A_CATCH)
AER	(mandatory) data frame containing AER catch data (i.e. map_fsfa0)
var	(mandatory) string to select the variable of interest. Use either 'landings' for landings or 'value' for landings' value.
MS	(mandatory) The MS code (e.g. 'GRC'). Only one MS code is allowed.
level	(optional) character to select the level of aggregation of variable 'var', i.e. 'MS', 'GSA', 'GEAR' or 'METIER'. E.g. if level 'GEAR' is selected, the comparison will be performed by 'MS', 'GSA' and 'GEAR'. Default level is 'METIER'.
YEAR	(optional) vector of years to perform the check on. Default is NA, which will produce a check for all the years found in both data frames
GSA	(optional) vector string of GSAs for the given MS to produce the check on (e.g. GSA = c('GSA20', 'GSA22')). Default is NA, which will produce the check for all GSAs existing in the data frames.
SP	(optional) vector string of species FAO 3-alpha codes to produce the checks on (e.g. SPECIES = c('DPS', 'TGS')). Default is NA, which will produce the check for all SP existing in the data frames.
OUT	If set as TRUE a table in csv will be saved in the OUTPUT folder created in the working directory. Default is FALSE
verbose	(optional) boolean, if TRUE messages are printed. Default is FALSE.

**Value**

The function returns a data frame containing all entries found in FDI, in AER and in both dataframes, with the last column reporting the ratio FDI/AER for the selected var. This ratio equals 1 if the two entries are the same, differs from 1 if the two entries differ or is NA if the entry is missing in one of the two dataframes. This output table is also saved in .csv files if OUT is TRUE.

**Author(s)**

Vasiliki Sgardeli [vsgard@hcmr.gr](mailto:vsgard@hcmr.gr)

**Examples**

```
FDI_AER_land_landvalue(FDI=fdi_a_catch, AER=aer_catch, var='landings',
  level='GEAR', MS='PSP', YEAR=NA, GSA=c('GSA97', 'GSA98'), SP=NA, OUT = TRUE, verbose = TRUE)
```

---

fdi\_a\_catch

*fdi\_a\_catch in FDI DGMAREMED&BS datacall format*

---

**Description**

fdi\_a\_catch in FDI DGMAREMED&BS datacall format

**Usage**

fdi\_a\_catch

**Format**

An object of class `data.frame` with 2600 rows and 23 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

FDI\_checks\_spatial\_HI *check NA values in spatial columns of both table H and I*

---

**Description**

The function checks the incorrect combination of NA in the spatial columns in both table H (Landings by rectangle) and table I (Effort by rectangle). The following check are included:

1. presence of NA values in 'c\_square' field when 'rectangle\_type', 'rectangle\_lat', 'rectangle\_lon' are all NA;
2. the presence any data in 'rectangle\_type', 'rectangle\_lat', 'rectangle\_lon' when 'c\_square' is reported;
3. the presence of any data in 'c-squares' when 'rectangle\_type', 'rectangle\_lat', 'rectangle\_lon' are filled in.

Furthermore the function identifies the records without any sub-region assignment.

**Usage**

```
FDI_checks_spatial_HI(data, MS, verbose = FALSE)
```

**Arguments**

data	data frame of one between FDI table H (Landings by rectangle) and table I (Effort by rectangle)
MS	Country code
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list of 2 tables. The first (NA\_inconsistencies) reports the records with the inconsistencies detected in spatial of the selected table, while the second one reports the record without any sub\_region assignment.

**Author(s)**

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_checks_spatial_HI(data = fdi_h_spatial_landings, MS = "PSP", verbose = TRUE)
FDI_checks_spatial_HI(data = fdi_i_spatial_effort, MS = "PSP", verbose = TRUE)
```

---

FDI_check_coord	<i>Compatibility of the geographical coordinates with rectangle type</i>
-----------------	--

---

**Description**

The function checks the compatibility of the geographical coordinates (latitude and longitude) with the value provided for the rectangle type.

**Usage**

```
FDI_check_coord(data, MS, verbose = FALSE)
```

**Arguments**

data	data frame of one FDI table between H (Landings by rectangle) and table I (Effort by rectangle)
MS	Country code
verbose	boolean. If TRUE a message is printed.

**Value**

The input data frame is filtered and is returned retaining the only records in which at least one among latitude and longitude is not compatible with the rectangle type. Two more columns ('lat.check' and 'lon.check') are added to the data frame structure to report the results of the checks respectively for latitude and longitude.

**Author(s)**

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_check_coord(data = fdi_i_spatial_effort, MS = "PSP", verbose = TRUE)
```

---

FDI_coverage	<i>Coverage of data by GSA and year (reporting the number of records)</i>
--------------	---

---

**Description**

Coverage of data by GSA and year (reporting the number of records)

**Usage**

```
FDI_coverage(data, MS, verbose = TRUE)
```

**Arguments**

data	data frame. One of the allowed FDI table among A, G, H, I, J
MS	Country
verbose	boolean. If TRUE a message is printed.



**Value**

the function returns a data frame reporting the coverage of the selected table in terms of number of records by country, GSA and year.

**Author(s)**

Vasiliki Sgardeli [vsgard@hcmr.gr](mailto:vsgard@hcmr.gr)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_coverage(data = fdi_a_catch, MS = "PSP", verbose = FALSE)
FDI_coverage(data = fdi_h_spatial_landings, MS = "PSP", verbose = FALSE)
FDI_coverage(data = fdi_g_effort, MS = "PSP", verbose = FALSE)
```

---

FDI\_cov\_tableA

*Check number of record in FDI A table*

---

**Description**

The function checks and count the numbers of records data in the given table A grouped by year, GSA, MS, species, vessels length, and fishing techniques for three variables (Total live weight landed (ton), total value of landings (euro), and total discards (ton)). If SP, Vessel length, and fishing technique are not specified by the user the function combines those by default.

**Usage**

```
FDI_cov_tableA(
  data,
  MS,
  SP = "COMBINED",
  vessel_len = "COMBINED",
  fishtech = "COMBINED",
  GSA,
  verbose = TRUE
)
```

**Arguments**

data	FDI table A catch
MS	member state code
SP	species reference code in the three alpha code format ("COMBINED" values perform the analysis for all species present in data)
vessel_len	vessels length code ("COMBINED" values perform the analysis for all vessels length present in data)
fishtech	selected fishing techniques ("COMBINED" values perform the analysis for all fishing techniques present in data)
GSA	GSA code
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list. The first element gives the summary table of records number. From the second to the fourth element gives 3 plots for each variables among: of total live weight landed, total value of landings (euro), and total discards (ton)).

**Author(s)**

Andrea Pierucci [pierucci@coispa.eu](mailto:pierucci@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_cov_tableA(data = fdi_a_catch, SP = "MUT", MS = "PSP", GSA = "GSA99")
FDI_cov_tableA(
  data = fdi_a_catch, SP = "MUT", MS = "PSP",
  fishtech = unique(fdi_a_catch$fishing_tech), GSA = "GSA99"
)
FDI_cov_tableA(data = fdi_a_catch, SP = "MUT", MS = "PSP", GSA = "GSA99")
```

---

FDI\_cov\_tableG

*Check FDI G table coverage*

---

**Description**

The function counts the numbers of records and summarise data in the given table grouped by year, GSA, MS, vessels length, fishing techniques, and metier for the following 8 variables: Total days at sea; Total Fishing Days; Total kW days at Sea; total GT days at sea; Total kW fishing days; totgtfishdays; Hours at Sea; kW hours at sea; according to the [Fisheries Dependent Information data call 2021 - Annex 1](#) If Vessel length, fishing technique, and metier are not specified by the user the function combines those by default.

**Usage**

```
FDI_cov_tableG(
  data,
  MS,
  GSA,
  vessel_len = "COMBINED",
  fishtech = "COMBINED",
  met = "COMBINED",
  verbose = TRUE
)
```

**Arguments**

data	FDI table G effort
MS	member state code
GSA	GSA code
vessel_len	vessels length code ("COMBINED" values perform the analysis for all vessels length present in data)

fishtech	selected fishing techniques ("COMBINED" values perform the analysis for all fishing techniques present in data)
met	selected metiers ("COMBINED" values perform the analysis for all metiers present in data)
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list. The first element gives the summary table of records number. From the second to the ninth element gives 8 plots for each variables among:

- totseadays,
- totfishdays,
- totkwdaysatsea,
- totgtdaysatsea,
- totkwfishdays,
- totgtfishdays,
- hrsea,
- kwhrsea

**Author(s)**

Andrea Pierucci [pierucci@coispa.eu](mailto:pierucci@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_cov_tableG(data = fdi_g_effort, MS = "PSP", GSA = "GSA99")
FDI_cov_tableG(fdi_g_effort, MS = "PSP", GSA = "GSA99", fishtech = "DTS", met = "OTB_MDD_>=40_0_0")
```

---

FDI_cov_tableJ	<i>Check number of record in FDI J table</i>
----------------	--

---

**Description**

The function provides counts the numbers of records and generate summary of data in the given table grouped by year, GSA, MS, vessels length, and fishing techniques for the following 4 variables: total trips; total kW; total GT; total vessels.

**Usage**

```
FDI_cov_tableJ(
  data,
  MS,
  GSA,
  vessel_len = "COMBINED",
  fishtech = "COMBINED",
  verbose = TRUE
)
```

**Arguments**

data	FDI table J capacity
MS	member state code
GSA	GSA code
vessel_len	vessels length code ("COMBINED" values perform the analysis for all vessels length present in data)
fishtech	selected fishing techniques ("COMBINED" values perform the analysis for all fishing techniques present in data)
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list. The first element gives the summary table of records number. From the second to the fifth element gives 4 plots for each variables among:

- total trips;
- total kW;
- total GT;
- total vessels;

**Author(s)**

Andrea Pierucci [pierucci@coispa.eu](mailto:pierucci@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_cov_tableJ(data = fdi_j_capacity, MS = "PSP", GSA = "GSA99")
FDI_cov_tableJ(data = fdi_j_capacity, MS = "PSP", GSA = "GSA99", fishtech = c("DTS", "PGP"))
```

---

FDI\_cross\_checks\_AG     *Cross check between FDI tables A and G*

---

**Description**

The function checks the possible data inconsistency between landings and effort.

**Usage**

```
FDI_cross_checks_AG(data1, data2, verbose = FALSE)
```

**Arguments**

data1	FDI table A catch
data2	FDI table G effort
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a table where all the mismatches between landings and effort are shown.

**Author(s)**

Andrea Pierucci [pierucci@coispa.eu](mailto:pierucci@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_cross_checks_AG(data1 = fdi_a_catch, data2 = fdi_g_effort)
FDI_cross_checks_AG(fdi_a_catch, fdi_g_effort)
```

---

FDI\_cross\_checks\_AH     *Cross check between FDI tables A and H*

---

**Description**

The function checks the possible data inconsistency between landings in table A and spatial landings in table H.

**Usage**

```
FDI_cross_checks_AH(data1, data2, verbose = FALSE)
```

**Arguments**

data1	FDI catch table A
data2	FDI spatial landings table H
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list with two tables. In the first table all the mismatches between landings in table A and spatial landings in table H are shown, in the second table the comparison between total landings of table A and total spatial landings in table H is shown.

**Author(s)**

Andrea Pierucci [pierucci@coispa.eu](mailto:pierucci@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_cross_checks_AH(data1 = fdi_a_catch, data2 = fdi_h_spatial_landings)
```

---

FDI\_cross\_checks\_IG    *Cross check between FDI tables I and G*

---

**Description**

The function checks the possible data inconsistency between spatial effort in table I and effort in table G.

**Usage**

```
FDI_cross_checks_IG(data1, data2, verbose = FALSE)
```

**Arguments**

data1	FDI spatial effort in table I
data2	FDI effort table G
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list with two tables. In the first table all the mismatches between spatial effort in table I and effort in table G are shown, in the second table the comparison between total spatial effort of table I and total effort in table G is shown.

**Author(s)**

Andrea Pierucci [pierucci@coispa.eu](mailto:pierucci@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_cross_checks_IG(data1 = fdi_i_spatial_effort, data2 = fdi_g_effort)
```

---

FDI\_cross\_checks\_JG    *Cross check between FDI tables J and G*

---

**Description**

The function checks the possible data inconsistency between the amount of vessels in table J capacity and the amount of vessels in table G.

**Usage**

```
FDI_cross_checks_JG(data1, data2, verbose = FALSE)
```

**Arguments**

data1	FDI capacity in table J
data2	FDI effort table G
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list with all the mismatches between number of vessels in table J and G.

**Author(s)**

Andrea Pierucci [pierucci@coispa.eu](mailto:pierucci@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_cross_checks_JG(data1 = fdi_j_capacity, data2 = fdi_g_effort, verbose = TRUE)
FDI_cross_checks_JG(fdi_j_capacity, fdi_g_effort)
```

---

FDI_disc_coverage	<i>Coverage of FDI discard data</i>
-------------------	-------------------------------------

---

**Description**

The functions checks the discard coverage in table A for the selected MS by GSAs.

**Usage**

```
FDI_disc_coverage(data, MS, GSA, SP, verbose = TRUE)
```

**Arguments**

data	data frame of the FDI table A
MS	Country
GSA	GSA code
SP	species reference code in the three alpha code format.
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list of data frames by GSA reporting the landing volumes (with discard >0, =0 and =NK and total landing) by year

**Author(s)**

Vasiliki Sgardeli [vsgard@hcmr.gr](mailto:vsgard@hcmr.gr)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_disc_coverage(fdi_a_catch, MS = "PSP", GSA = "GSA99", SP = "HKE", verbose = TRUE)
```

---

FDI_fishdays_cov	<i>Coverage comparison of totfishdays between FDI tables G and I</i>
------------------	--

---

**Description**

The function checks the comparison of totfishdays in FDI tables G and I

**Usage**

```
FDI_fishdays_cov(dataG, dataI, MS, verbose = TRUE)
```

**Arguments**

dataG	data frame of FDA table G
dataI	data frame of FDA table I
MS	country code
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a data frame of the comparison of totfishdays between FDI tables G and I

**Author(s)**

Vasiliki Sgardeli [vsgard@hcmr.gr](mailto:vsgard@hcmr.gr)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_fishdays_cov(dataG = fdi_g_effort, dataI = fdi_i_spatial_effort, MS = "PSP", verbose = TRUE)
```

---

fdi_g_effort	<i>fdi_g_effort in FDI DGMAREMED&amp;BS datacall format</i>
--------------	---

---

**Description**

fdi\_g\_effort in FDI DGMAREMED&BS datacall format

**Usage**

```
fdi_g_effort
```

**Format**

An object of class `data.frame` with 2450 rows and 26 columns.

**Author(s)**

Isabella Bitetto <[bitetto@coispa.it](mailto:bitetto@coispa.it)>



---

`fdi_h_spatial_land`     *fdi\_h\_spatial\_land in FDI DGMAREMED&BS datacall format*

---

**Description**

`fdi_h_spatial_land` in FDI DGMAREMED&BS datacall format

**Usage**

`fdi_h_spatial_land`

**Format**

An object of class `data.frame` with 5 rows and 23 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

`fdi_h_spatial_landings`  
*fdi\_h\_spatial\_landings in FDI DGMAREMED&BS datacall format*

---

**Description**

`fdi_h_spatial_landings` in FDI DGMAREMED&BS datacall format

**Usage**

`fdi_h_spatial_landings`

**Format**

An object of class `data.frame` with 2450 rows and 23 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

fdi\_i\_spatial\_effort    *fdi\_i\_spatial\_effort in FDI DGMAREMED&BS datacall format*

---

**Description**

fdi\_i\_spatial\_effort in FDI DGMAREMED&BS datacall format

**Usage**

fdi\_i\_spatial\_effort

**Format**

An object of class data.frame with 2450 rows and 21 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

fdi\_j\_capacity    *fdi\_j\_capacity in FDI DGMAREMED&BS datacall format*

---

**Description**

fdi\_j\_capacity in FDI DGMAREMED&BS datacall format

**Usage**

fdi\_j\_capacity

**Format**

An object of class data.frame with 2450 rows and 14 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

FDI\_landweight\_cov      *Coverage of weight of landings in FDI table A and H*

---

**Description**

The functions checks the coverage of weight of landings comparing data reported in table A and H.

**Usage**

```
FDI_landweight_cov(dataA, dataH, MS, verbose = FALSE)
```

**Arguments**

dataA	data frame of the FDI table A
dataH	data frame of the FDI table H
MS	Country code
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a data frame reporting the weight of landings by GSA and by year.

**Author(s)**

Vasiliki Sgardeli [vsgard@hcmr.gr](mailto:vsgard@hcmr.gr)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_landweight_cov(dataA = fdi_a_catch, dataH = fdi_h_spatial_landings, MS = "PSP", verbose = TRUE)
```

---

FDI\_prices\_cov      *Check prices trend in FDI A table*

---

**Description**

The function checks the trend prices in the given table grouped by year, GSA, MS, and species. If SP are not specified by the user the function combines those by default.

**Usage**

```
FDI_prices_cov(data, MS, SP = "COMBINED", GSA = "COMBINED", verbose = TRUE)
```

**Arguments**

data	FDI table A catch
MS	member state code
SP	species reference code in the three alpha code format ("COMBINED" values perform the analysis for all species present in data)
GSA	GSA code ("COMBINED" values perform the analysis for all GSAs present in data).
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list. The first element gives the summary table of records number. From the second to the fourth element gives 3 plots for each variables among: of total live weight landed, total value of landings (euro), and total discards (ton)).

**Author(s)**

Andrea Pierucci [pierucci@coispa.eu](mailto:pierucci@coispa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_prices_cov(data = fdi_a_catch, SP = c("MUT", "HKE"), MS = "PSP", GSA = "GSA99")
FDI_prices_cov(data = fdi_a_catch, SP = "MUT", MS = "PSP", GSA = "GSA99")
```

---

FDI\_prices\_not\_null    *Check of species value*

---

**Description**

The function estimates from the FDI table A an average price per species and year and compares it with average price calculated per country (by species). Furthermore, the function performs comparisons between total weight landings and total value landings. In particular it identifies the cases with total landings > 0 but landings value = 0. In case SP parameter is not specified, the analysis is conducted over all the species in the provided data frame.

**Usage**

```
FDI_prices_not_null(data, MS, GSA, SP = NA, verbose = FALSE)
```

**Arguments**

data	data frame of FDA table A catch.
MS	country code
GSA	GSA code
SP	vector of the species code for which the check should be performed
verbose	boolean. If TRUE a message is printed.

**Value**

the function returns a list of two data frames. The first one reports the prices comparison by species, while the second one reports the cases in which total landings > 0 but landings value = 0.

**Author(s)**

Vasiliki Sgardeli [vsgard@hcmr.gr](mailto:vsgard@hcmr.gr)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_prices_not_null(
  data = fdi_a_catch, MS = "PSP", GSA = "GSA99",
  SP = c("ARA", "BOG", "HKE"), verbose = TRUE
)
```

---

FDI_vessel_lenth	<i>Check of vessel lenth in FDI table J</i>
------------------	---

---

**Description**

The function checks the average length vessels with the vessel length category (table J)

**Usage**

```
FDI_vessel_lenth(data, MS, verbose = TRUE)
```

**Arguments**

data	data frame of table J
MS	country code
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list of two data frames. The first reports the records with NAs in either vessel length or vessel category or both, while the second table reports the cases in which vessel length does not match vessel length category.

**Author(s)**

Vasiliki Sgardeli [vsgard@hcmr.gr](mailto:vsgard@hcmr.gr)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
FDI_vessel_lenth(data = fdi_j_capacity, MS = "PSP", verbose = TRUE)
```

---

FDI\_vessel\_numbers      *Check number of vessels in FDI table J and G*

---

### Description

The function cross-checks the number of vessels in table J in comparison with the number reported in table G.

### Usage

```
FDI_vessel_numbers(dataJ, dataG, MS, verbose = TRUE)
```

### Arguments

dataJ	data frame of FDA table J catch.
dataG	data frame of FDA table G catch.
MS	country code
verbose	boolean. If TRUE a message is printed.

### Value

The function returns a list of data frames. The first element reports the number of vessel in table J in comparison with table G and the relative difference percentage, while the second one reports the vessels not present in table G.

### Author(s)

Vasiliki Sgardeli [vsgard@hcmr.gr](mailto:vsgard@hcmr.gr)  
 Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
FDI_vessel_numbers(dataJ = fdi_j_capacity, dataG = fdi_g_effort, MS = "PSP", verbose = TRUE)
```

---

GFCM\_check\_headers      *Check headers of GFCM tables*

---

### Description

Check headers of GFCM tables

### Usage

```
GFCM_check_headers(data, table)
```

### Arguments

data	data frame of data for the reference table
table	character, vector specifying the reference table. Allowed values: "task2.2", "task3", "task7.2", "task7.3.1", "task7.3.2"

**Value**

In case the data frame reports has the headers in the format of the RDBFIS database, it is returned back with the headers in the GFCM format adopted also by RDBqc package

---

GFCM_cov_II2	<i>Check coverage of GFCM task II.2 table</i>
--------------	---

---

**Description**

The function checks and count the numbers of records data in the given task II.2 table grouped by year, GSA, MS, species, and segment for two variables (Total landing (ton) and total discards (ton)). If SP and segment are not specified by the user the function combines those by default.

**Usage**

```
GFCM_cov_II2(
  data,
  MS,
  SP = "COMBINED",
  segment = "COMBINED",
  GSA,
  verbose = TRUE
)
```

**Arguments**

data	GFCM task II.2 table
MS	member state code
SP	species reference code in the three alpha code format ("COMBINED" values perform the analysis for all species present in data)
segment	segment code ("COMBINED" values perform the analysis for all segments present in data)
GSA	GSA code
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a list. The first element gives the summary table of records number. From the second to the third element gives 2 plots for total live landing and total discards (ton)).

**Author(s)**

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
GFCM_cov_II2(data = task_ii2, MS = "ITA", GSA = "18", SP = "HKE", segment = "COMBINED")
```

---

GFCM\_cov\_task\_iii      *Check number of individuals in GFCM Task III table*

---

### Description

The function checks the consistencies of the total number of individual caught in comparison with the sum of the individuals alive, dead and released reported in Task III table.

### Usage

```
GFCM_cov_task_iii(data, MS, GSA, SP = "COMBINED", verbose = TRUE)
```

### Arguments

data	Task III table
MS	member state code
GSA	GSA code
SP	species reference code in the three alpha code format ("COMBINED" values perform the analysis for all species present in data)
verbose	boolean. If TRUE a message is printed.

### Value

The function returns a list. The first element is a table reporting the inconsistencies of the total individual caught in comparison to the sum of the individuals alive, dead and released. The second element of the list is a plot of the time series of the individuals caught by year, source, and gear.

### Author(s)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
GFCM_cov_task_iii(
  data = task_iii, SP = "Dasyatis pastinaca",
  MS = "ITA", GSA = "18"
)
```

---

GP\_tab\_example      *GP table in MED&BS datacall format*

---

### Description

GP table in MED&BS datacall format

### Usage

```
GP_tab_example
```



**Format**

An object of class `data.frame` with 12 rows and 22 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

GSAlist

*GSAlist table*

---

**Description**

GSAlist table

**Usage**

GSAlist

**Format**

An object of class `data.frame` with 22 rows and 2 columns.

**Author(s)**

Walter Zupa <zupa@fondazionecoispa.org>

---

Landing\_tab\_example

*Landing table in MED&BS datacall format*

---

**Description**

Landing table in MED&BS datacall format

**Usage**

Landing\_tab\_example

**Format**

An object of class `data.frame` with 57 rows and 114 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

MA_tab_example	<i>MA table in MED&amp;BS datacall format</i>
----------------	---

---

**Description**

MA table in MED&BS datacall format

**Usage**

MA\_tab\_example

**Format**

An object of class `data.frame` with 71 rows and 11 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

MEDBSSP	<i>MEDBSSP table</i>
---------	----------------------

---

**Description**

MEDBSSP table

**Usage**

MEDBSSP

**Format**

An object of class `data.frame` with 63 rows and 1 columns.

**Author(s)**

Alessandro Mannini <alessandro.mannini@irbim.cnr.it>

---

MEDBS\_ALK                      *Plot of Age-Length Keys*

---

### Description

The function allows to check the data in the ALK table providing plots by sex for a selected species

### Usage

```
MEDBS_ALK(data, SP, MS, GSA, verbose = TRUE)
```

### Arguments

data	ALK table in MED&BS datacall format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned.

### Value

a list of ALK plots by sex is return.

### Author(s)

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
MEDBS_ALK(data = ALK_tab_example, SP = "MUT", MS = "ITA", GSA = "GSA 99")
```

---

MEDBS\_Catch\_coverage    *Check the coverage of Catch table*

---

### Description

The function allows to check the coverage of Catch table by mean of summary tables summarizing both landing and discard volumes and producing relative plots for the selected species.

### Usage

```
MEDBS_Catch_coverage(data, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	Catch table in MEDBS format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns two summary tables: one for landing coverage and the other for discard coverage. Furthermore, plots of landings and discards by gear are also returned

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_Catch_coverage(Catch_tab_example, "DPS", "ITA", "GSA 9")
```

---

MEDBS\_check\_disaggregated

*Check for disaggregated data rows in landings, discards and catch tables*

---

**Description**

The function checks the presence of disaggregated data (same stratum but with different volumes of landings and discards) rows in MED & BS landings, discards and catch tables.

**Usage**

```
MEDBS_check_disaggregated(data, type = "l", SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	data frame containing either landings, discards or catch data
type	string vector indicating the type of table to be checked. "l" for landing; "d" for discards; "c" for catch table.
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a data frame containing the disaggregated rows to be likely aggregated in the data.

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

Loredana Casciaro [casciaro@coispa.eu](mailto:casciaro@coispa.eu)

**Examples**

```
MEDBS_check_disaggregated(
  data = Discard_tab_example, type = "d", SP = "DPS",
  MS = "ITA", GSA = "GSA 9", verbose = TRUE
)
MEDBS_check_disaggregated(
  data = Landing_tab_example, type = "l", SP = "DPS",
  MS = "ITA", GSA = "GSA 9", verbose = TRUE
)
MEDBS_check_disaggregated(
  data = Catch_tab_example, type = "c", SP = "DPS",
  MS = "ITA", GSA = "GSA 9", verbose = TRUE
)
```

---

MEDBS\_check\_duplicates

*Check for duplicated data rows in MED & BS tables*

---

**Description**

The function checks the presence of duplicated rows in MED & BS tables data.

**Usage**

```
MEDBS_check_duplicates(data, type = "l", SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	data frame containing either landings, discards or catch data
type	string vector indicating the type of table to be checked. "l" for landing; "d" for discards; "c" for catch table; "gp" for growth parameters table; "alk" for age-length keys table; "ma" for maturity at age table; "ml" for maturity at length table; "sra" for sex ratio at age table; "srl" for maturity at length table..
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a data frame containing the duplicated rows to be likely deleted from the data.

**Author(s)**Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)**Examples**

```

MEDBS_check_duplicates(
  data = Discard_tab_example, type = "d", SP = "DPS",
  MS = "ITA", GSA = "GSA 9", verbose = TRUE
)
MEDBS_check_duplicates(
  data = Landing_tab_example, type = "l", SP = "DPS",
  MS = "ITA", GSA = "GSA 9", verbose = TRUE
)
MEDBS_check_duplicates(
  data = Catch_tab_example, type = "c", SP = "DPS",
  MS = "ITA", GSA = "GSA 9", verbose = TRUE
)

```

---

MEDBS\_check\_missing\_years

*Check for missing years in MED & BS tables*


---

**Description**

The function checks the presence of missing years in MED & BS tables data.

**Usage**

```

MEDBS_check_missing_years(
  data,
  end_year,
  type = "l",
  SP,
  MS,
  GSA,
  verbose = TRUE
)

```

**Arguments**

data	data frame containing either landings, discards or catch data
end_year	numeric value reporting the final year of the time series for the given country
type	string vector indicating the type of table to be checked. "l" for landing; "d" for discards; "c" for catch table.
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a vector containing the missing years in data. NULL is returned in case no data were available for the selected combination of country, GSA and species. A vector of length 0 is returned in case there are no missing years in the data.

**Author(s)**

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
df <- Discard_tab_example[-which(Discard_tab_example$year==2011),]
MEDBS_check_missing_years(
  data = df, end_year=2002, type = "d", SP = "DPS",
  MS = "ITA", GSA = "GSA 9", verbose = TRUE
)
```

---

MEDBS\_comp\_disc\_YQ      *Comparison between discards in weight by quarter and -1*

---

**Description**

The function compares the discards weights aggregated by quarter and by year for a selected species at gear level.

**Usage**

```
MEDBS_comp_disc_YQ(data, MS, GSA, SP, verbose = FALSE)
```

**Arguments**

data	data frame containing discards data
MS	member state code
GSA	GSA code (Geographical sub-area)
SP	species code
verbose	boolean. If TRUE a message is printed.

**Value**

The function returns a data frame with the comparison of discards aggregated by quarters and by year

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_comp_disc_YQ(data = Discard_tab_example, MS = "ITA", GSA = "GSA 9", SP = "DPS")
```

---

MEDBS\_comp\_disc\_YQ\_fishery

*Comparison between discards in weight by quarter, quarter -1 and by fishery*

---

### Description

The function allow to estimates the discards in weight for a selected species by quarter and fishery

### Usage

```
MEDBS_comp_disc_YQ_fishery(data, SP, MS, GSA, verbose = TRUE)
```

### Arguments

data	data frame containing discards data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

### Value

The function returns a data frame with the comparison of discards aggregated by quarters and by year and fishery

### Author(s)

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
MEDBS_comp_disc_YQ_fishery(data = Discard_tab_example, SP = "DPS", MS = "ITA", GSA = "GSA 9")
```

---

MEDBS\_comp\_land\_Q\_VL

*Comparison between landings in weight by quarter accounting for vessel length*

---

### Description

The function allows to perform the comparison of landings of a selected species aggregated by quarters, accounting for the presence of vessel length

### Usage

```
MEDBS_comp_land_Q_VL(data, SP, MS, GSA, verbose = TRUE)
```



**Arguments**

data	data frame containing landing data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a data frame for the comparison of landings aggregated by quarters, accounting for the presence of vessel length information.

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_comp_land_Q_VL(data = Landing_tab_example, MS = "ITA", GSA = "GSA 9", SP = "DPS")
```

---

MEDBS\_comp\_land\_Q\_VL\_fishery

*Comparison between landings in weight by quarter and fishery, accounting for vessel length*

---

**Description**

The function performs the comparison of landings of a selected species aggregated by quarters and fishery, accounting for the presence of vessel length

**Usage**

```
MEDBS_comp_land_Q_VL_fishery(data, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	data frame containing landing data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a data frame for the comparison of landings aggregated by quarters and fishery accounting for the presence of vessel length information.

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_comp_land_Q_VL_fishery(data = Landing_tab_example, SP = "DPS", MS = "ITA", GSA = "GSA 9")
```

---

MEDBS\_comp\_land\_YQ      *Comparison between landings in weight by quarter and -1*

---

**Description**

The function allows to perform the comparison of landings of a selected species aggregated by quarters and by year

**Usage**

```
MEDBS_comp_land_YQ(data, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	data frame containing landing data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a data frame for the comparison of landings aggregated by quarters and by year

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_comp_land_YQ(data = Landing_tab_example, SP = "DPS", MS = "ITA", GSA = "GSA 9")
```

---

MEDBS\_comp\_land\_YQ\_fishery

*Comparison between landings in weight by quarter, quarter -1 and by fishery*

---

### Description

The function allows to perform the comparison of landings of a selected species aggregated by quarters, and by year and fishery

### Usage

```
MEDBS_comp_land_YQ_fishery(data, SP, MS, GSA, verbose = TRUE)
```

### Arguments

data	data frame containing landing data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

### Value

The function returns a data frame for the comparison of landings aggregated by quarters, and by year and fishery

### Author(s)

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
MEDBS_comp_land_YQ_fishery(data = Landing_tab_example, SP = "DPS", MS = "ITA", GSA = "GSA 9")
```

---

MEDBS\_discard\_coverage

*Check the coverage of discard data*

---

### Description

The function allows to check the coverage of the time series in discard table for a selected species.

### Usage

```
MEDBS_discard_coverage(Discard_tab, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

Discard_tab	Discard table in MED&BS format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

A summary table and a plot of discard time series by year and gear are returned.

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_discard_coverage(Discard_tab_example, "DPS", "ITA", "GSA 9")
```

---

MEDBS\_disc\_mean\_weight

*Check of mean discard weight aggregations by year, gear and fishery*

---

**Description**

The function allows to check consistency of mean discard of a selected species plotting the discards' weight by year, gear and fishery

**Usage**

```
MEDBS_disc_mean_weight(data, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	data frame containing discards data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a plot of the mean discards weight by year, gear and fishery aggregation

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_disc_mean_weight(data = Discard_tab_example, SP = "DPS", MS = "ITA", GSA = "GSA 9")
```

---

MEDBS\_GP\_check

*Check of growth parameters table*

---

**Description**

The function checks the growth parameters by sex and year for a selected species

**Usage**

```
MEDBS_GP_check(data, SP, MS, GSA, verbose = FALSE)
```

**Arguments**

data	growth parameters table in MED&BS data call format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

A list of objects containing a summary table and different plots of the growth curves by sex and year is returned by the function.

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_GP_check(GP_tab_example, "MUT", "ITA", "GSA 18")
```

---

MEDBS\_ks

*Kolmogorov-Smirnov test*

---

### Description

The function allows to perform the Kolmogorov-Smirnov test on both landings and discards for a selected species providing cumulative length distribution plots by fishery and year. The function performs Kolmogorov-Smirnov tests on couples of years to assess if they belong to the same population.

### Usage

```
MEDBS_ks(data, type, SP, MS, GSA, Rt = 1, verbose = TRUE)
```

### Arguments

data	data frame of either landings or discards data
type	type of data frame. "l" for landing and "d" for discard
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
Rt	ratio to be applied to subsample data to reduce the risk of rejection of H0 Hypothesis
verbose	boolean. If TRUE messages are returned

### Value

the function returns a list of data frames and cumulative distribution plots

### Author(s)

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
MEDBS_ks(data = Landing_tab_example, type = "l", SP = "DPS", MS = "ITA", GSA = "GSA 9", Rt = 1)
```

---

 MEDBS\_Landing\_coverage

*Check the coverage of Landing table*


---

### Description

The function allows to check the coverage in landing table providing a summary table and a plot of landing.

### Usage

```
MEDBS_Landing_coverage(data, SP, MS, GSA, verbose = TRUE)
```

### Arguments

data	Landing table in MED&BS format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

### Value

A list containing a summary table and coverage plot is provided.

### Examples

```
MEDBS_Landing_coverage(Landing_tab_example, "DPS", "ITA", "GSA 9")
```

---

 MEDBS\_land\_mean\_weight

*Mean weight by year, gear and fishery aggregation*


---

### Description

The function allows to check consistency of mean landing of a selected species plotting the landings' weight by year, gear and fishery

### Usage

```
MEDBS_land_mean_weight(data, SP, MS, GSA, verbose = TRUE)
```

### Arguments

data	data frame containing landing data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a plot of the mean landing weight by year, gear and fishery aggregation, and returns the data frame as well.

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_land_mean_weight(data = Landing_tab_example, SP = "DPS", MS = "ITA", GSA = "GSA 9")
```

---

MEDBS\_lengthclass\_0 *Checks length classes numbers with zeros in landings and discards*

---

**Description**

The function checks landings and discards for the presence of length class filled in having weight > 0.

**Usage**

```
MEDBS_lengthclass_0(data, type = "l", SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	data frame containing landing or discards data
type	string vector indicating the type of table to be checked. "l" for landing; "d" for discards.
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a data frame with the rows with 0 values length class having weight > 0.

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)



**Examples**

```

MEDBS_lengthclass_0(
  data = Landing_tab_example, type = "l",
  SP = "DPS", MS = "ITA", GSA = "GSA 9", verbose = TRUE
)
MEDBS_lengthclass_0(
  data = Discard_tab_example, type = "d", SP = "DPS",
  MS = "ITA", GSA = "GSA 9", verbose = TRUE
)

```

---

MEDBS_length_ind	<i>Main length size indicators</i>
------------------	------------------------------------

---

**Description**

The function allows to check the consistency of length data for a selected species on both landings and discards: Main length size indicators

**Usage**

```

MEDBS_length_ind(
  data,
  type,
  SP,
  MS,
  GSA,
  splines = c(0.2, 0.4, 0.6, 0.8),
  Xtresholds = c(0.25, 0.5, 0.75),
  verbose = TRUE
)

```

**Arguments**

data	data frame of either landings or discards data
type	type of data frame. "l" for landing and "d" for discard
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
splines	spline values assignment to fit cumulative distributions
Xtresholds	threshold value
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a plot of the Main length size indicators time series by fishery

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```

MEDBS_length_ind(Landing_tab_example,
  type = "l", SP = "DPS", MS = c("ITA"),
  GSA = c("GSA 9"), splines = c(0.2, 0.4, 0.6, 0.8),
  Xtresholds = c(0.25, 0.5, 0.75)
)
MEDBS_length_ind(Discard_tab_example,
  type = "d", SP = "DPS", MS = c("ITA"),
  GSA = c("GSA 9"), splines = c(0.2, 0.4, 0.6, 0.8),
  Xtresholds = c(0.25, 0.5, 0.75)
)

```

MEDBS\_LFD

*LFD of MED & BS landing and discard data***Description**

The function allows to report the observed Length Frequency Distributions of a selected species.

**Usage**

```
MEDBS_LFD(data, data2, type, SP, MS, GSA, OUT = FALSE, verbose = FALSE)
```

**Arguments**

data	data frame of either landings or discards data
data2	data frame of discards data, only in case type="b"
type	type of data frame. "l" for landing, "d" for discard and "b" for both the data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
OUT	boolean. If TRUE outputs are saved in the OUTPUT folder
verbose	boolean. If TRUE messages are returned

**Value**

The function returns different data frame and plots of LFD of Landing, Discards and Catches according to the source data used in the analysis.

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```

MEDBS_LFD(data=RDBqc::Landing_tab_example,
  data2=RDBqc::Discard_tab_example, type="b",
  SP="DPS", MS="ITA", GSA="GSA 9", OUT=TRUE, verbose = TRUE)

```

---

MEDBS\_LW\_check      *Check LW parameters in GP table*

---

### Description

The function checks the length-weight parameters included in the GP table for a selected species.

### Usage

```
MEDBS_LW_check(data, SP, MS, GSA, verbose = FALSE)
```

### Arguments

data	growth params table in MED&BS datacall format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

### Value

A summary table and plots of the LW parameters are returned by the function.

### Author(s)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
MEDBS_LW_check(GP_tab_example, "MUT", "ITA", "GSA 18")
```

---

MEDBS\_MA\_check      *Check of MA\_tab (maturity at age) table*

---

### Description

The function checks the maturity at age (MA) table providing a summary table of the data coverage and plots for the selected species of the proportion of matures for age class by sex and year.

### Usage

```
MEDBS_MA_check(data, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	maturity at AGE table in MED&BS datacall format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

A summary table and plots are returned by the function.

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_MA_check(MA_tab_example, "DPS", "ITA", "GSA 99")
```

---

MEDBS\_ML\_check

*Check of ML\_tab (maturity at length) table*

---

**Description**

The function checks the maturity at length (ML) table providing a summary table of the data coverage and plots for the selected species of the proportion of matures for age class by sex and year.

**Usage**

```
MEDBS_ML_check(data, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	maturity at length table in MED&BS datacall format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

a summary table and plots

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

## Examples

```
MEDBS_ML_check(ML_tab_example, "DPS", "ITA", "GSA 99")
```

---

MEDBS\_plot\_discard\_ts *Plot of total discards*

---

## Description

The function estimates the total discard time series by both year and quarters for a selected combination of member state, GSA and species.

## Usage

```
MEDBS_plot_discard_ts(data, SP, MS, GSA, by = "year")
```

## Arguments

data	data frame containing discard data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
by	string defining the temporal aggregation level of discard data to be plotted. Allowed values are: "year" and "quarter"

## Value

The function returns a plot of the total discard time series by year or by quarters. The plot by year also reports the landing by gear.

## Author(s)

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

## Examples

```
MEDBS_plot_discard_ts(  
  data = Discard_tab_example, SP = "DPS",  
  MS = "ITA", GSA = "GSA 9", by = "quarter"  
)  
MEDBS_plot_discard_ts(  
  data = Discard_tab_example, SP = "DPS",  
  MS = "ITA", GSA = "GSA 9", by = "year"  
)
```

MEDBS\_plot\_disc\_vol *Plot of total discards by gear and fishery*

---

### Description

The function allows to visual check the time series of discard volumes by fishery of a selected species

### Usage

```
MEDBS_plot_disc_vol(data, SP, MS, GSA)
```

### Arguments

data	data frame containing discard data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)

### Value

The function returns a plot of the total discards time series by fishery and gear

### Author(s)

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
MEDBS_plot_disc_vol(data = Discard_tab_example, SP = "DPS", MS = "ITA", GSA = "GSA 9")
```

---

MEDBS\_plot\_landing\_ts *Plot of total landing time series*

---

### Description

The function estimates the total landings time series by both year and quarters for a selected combination of member state, GSA and species.

### Usage

```
MEDBS_plot_landing_ts(data, SP, MS, GSA, by = "year", verbose = TRUE)
```

**Arguments**

data	data frame containing landing data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
by	string defining the temporal aggregation level of landing data to be plotted. Allowed values are: "year" and "quarter"
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a plot of the total landing time series by year or by quarters. The plot by year also reports the landing by gear.

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

**Examples**

```
MEDBS_plot_landing_ts(
  data = Landing_tab_example, SP = "DPS",
  MS = "ITA", GSA = "GSA 9", by = "quarter"
)
MEDBS_plot_landing_ts(
  data = Landing_tab_example, SP = "DPS", MS = "ITA",
  GSA = "GSA 9", by = "year"
)
```

---

MEDBS\_plot\_land\_vol *Plot of total landing by gear and fishery*

---

**Description**

The function checks the time series of landing volumes by fishery of a selected species

**Usage**

```
MEDBS_plot_land_vol(data, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	data frame containing landing data
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a plot of the total landing time series by fishery and gear

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

**Examples**

```
MEDBS_plot_land_vol(data = Landing_tab_example, SP = "DPS", MS = "ITA", GSA = "GSA 9")
```

---

MEDBS\_SA\_check

*Check of SA\_tab (sex ratio at age) table*

---

**Description**

The function allows to check the sex ratio at age (SA) table providing a summary table of the data coverage and plots for the selected species of the proportion of sex ratio for age class by year.

**Usage**

```
MEDBS_SA_check(data, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	sex ratio at age table in MED&BS datacall format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

a summary table and plots

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_SA_check(SA_tab_example, "DPS", "ITA", "GSA 99")
```



---

MEDBS_SL_check	<i>Check of SL_tab (sex ratio at length) table</i>
----------------	--

---

### Description

The function allows to check the sex ratio at length (SL) table providing a summary table of the data coverage and plots for the selected species of the proportion of sex ratio for length class by year.

### Usage

```
MEDBS_SL_check(data, SP, MS, GSA, verbose = TRUE)
```

### Arguments

data	sex ratio at length table in MED&BS data call format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

### Value

a summary table and plots

### Author(s)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
MEDBS_SL_check(SL_tab_example, "DPS", "ITA", "GSA 99")
```

---

MEDBS_SOP	<i>check of the sum of products</i>
-----------	-------------------------------------

---

### Description

check of the sum of products

### Usage

```
MEDBS_SOP(data, SP, MS, GSA, threshold = 5, verbose = TRUE)
```

**Arguments**

data	Catch table in MED&BS datacall format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
threshold	threshold value in percentage to check the percentage difference between SOP and both volumes of landing and discard. Default value is 5%
verbose	boolean. If TRUE messages are returned

**Value**

the function returns a data frame of the record with values of percentage difference between SOP and both volumes of landing and discard greater then threshold value.

**Author(s)**

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_SOP(data = Catch_tab_example, SP = "DPS", MS = "ITA", GSA = "GSA 9", threshold = 5)
```

---

MEDBS\_weight\_0                      *Check of weights 0 in landings and discards*

---

**Description**

The function checks landings or discards in weight equal to 0 having length classes filled in

**Usage**

```
MEDBS_weight_0(data, type = "l", SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	data.table object containing landing or discard data
type	type of table: "l" for landings; "d" for discards
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a table of rows with 0 values in weights having length classes filled in.

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
MEDBS_weight_0(  
  data = Landing_tab_example, type = "l",  
  SP = "DPS", MS = "ITA", GSA = "GSA 9", verbose = TRUE  
)  
MEDBS_weight_0(  
  data = Discard_tab_example, type = "d", SP = "DPS",  
  MS = "ITA", GSA = "GSA 9", verbose = TRUE  
)
```

---

MEDBS\_weight\_minus1    *Check weights -1 in landings*

---

**Description**

The function checks landings in weight equal to -1 having length class filled in

**Usage**

```
MEDBS_weight_minus1(data, type = "l", SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	data.table object containing landing or discards data
type	type of table: "l" for landings; "d" for discards
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned.

**Value**

The function returns a table of rows with -1 values in landing weights having length class filled in.

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```

MEDBS_weight_minus1(
  data = Landing_tab_example, type = "l",
  SP = "DPS", MS = "ITA", GSA = "GSA 9", verbose = TRUE
)
MEDBS_weight_minus1(
  data = Discard_tab_example, type = "d", SP = "DPS",
  MS = "ITA", GSA = "GSA 9", verbose = TRUE
)

```

---

MEDBS\_yr\_missing\_length

*Years with missing length distributions*

---

**Description**

The function checks the presence of years with missing length distributions in both landings and discards for a selected species.

**Usage**

```
MEDBS_yr_missing_length(data, type, SP, MS, GSA, verbose = FALSE)
```

**Arguments**

data	data frame of landings or discards data
type	type of data frame. "l" for landing and "d" for discard
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a data frame containing the reference combination of year, gear and fishery missing length distributions.

**Author(s)**

Alessandro Mannini [alessandro.mannini@ec.europa.eu](mailto:alessandro.mannini@ec.europa.eu)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```

MEDBS_yr_missing_length(
  data = Discard_tab_example, type = "d",
  SP = "DPS", MS = "ITA", GSA = "GSA 9"
)
MEDBS_yr_missing_length(
  data = Landing_tab_example, type = "l", SP = "DPS",
  MS = "ITA", GSA = "GSA 9"
)

```

---

minmaxLtaskVII2	<i>minmaxLtaskVII2 in GFCM datacall format</i>
-----------------	--

---

**Description**

minmaxLtaskVII2 in GFCM datacall format

**Usage**

minmaxLtaskVII2

**Format**

An object of class data.frame with 26 rows and 3 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

minmaxLtaskVII31	<i>minmaxLtaskVII31 in GFCM datacall format</i>
------------------	---

---

**Description**

minmaxLtaskVII31 in GFCM datacall format

**Usage**

minmaxLtaskVII31

**Format**

An object of class data.frame with 36 rows and 4 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

ML_tab_example	<i>ML table in MED&amp;BS datacall format</i>
----------------	---

---

**Description**

ML table in MED&BS datacall format

**Usage**

ML\_tab\_example

**Format**

An object of class `data.frame` with 437 rows and 12 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

RCG_check_AL	<i>Check consistency of age-length relationship</i>
--------------	---

---

**Description**

The function checks the consistency of the age-length data included in the table of RCG detailed data. Furthermore, the function identifies the age data outside a reference range of values provided by the user for the species. In case the `min_age` and `max_age` parameters are not provided, the function tests the lowest and the higher values of Age as outliers by mean of the Grubbs' test (from package outliers)

**Usage**

```
RCG_check_AL(data, MS, GSA, SP, min_age = NA, max_age = NA, verbose = TRUE)
```

**Arguments**

<code>data</code>	table of detailed data in RCG format
<code>MS</code>	member state code
<code>GSA</code>	GSA code (Geographical sub-area)
<code>SP</code>	species code
<code>min_age</code>	minimum age expected
<code>max_age</code>	maximum age expected
<code>verbose</code>	boolean. If TRUE messages are returned

**Value**

The function returns a list of object containing a summary table, an age-length plot and a table of the outliers, if any.

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**References**

Grubbs, F.E. (1950). Sample Criteria for testing outlying observations. *Ann. Math. Stat.* 21, 1, 27-58

**Examples**

```
RCG_check_AL(
  data = data_ex, MS = "ITA", GSA = "GSA99",
  SP = "Mullus barbatus", min_age = 0, max_age = 9
)
RCG_check_AL(data = data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus")
```

---

 RCG\_check\_CL

*Quality checks on CL RCG table*


---

**Description**

The function checks CL tables for temporal, spatial, species and metier coverage

**Usage**

```
RCG_check_CL(data, MS, GSA, SP, verbose = TRUE)
```

**Arguments**

data	Landing table in RCG CL format
MS	member state code
GSA	GSA code (Geographical sub-area)
SP	species code
verbose	boolean. If TRUE messages are returned

**Value**

The output is a list of 6 data frames:

1. Sum of Landings by year, quarter and month;
2. Sum of Landing value by year, quarter and month;
3. Sum of landings by LandCtry, VslFlgCtry, Area, Rect, SubRect, Harbour;
4. Sum of landing value by LandCtry, VslFlgCtry, Area, Rect, SubRect, Harbour;
5. Sum of landings by Year, Species, fishing\_activity\_category\_national, fishing\_activity\_category\_eu\_l6;
6. Sum of landing value by Year, Species, fishing\_activity\_category\_national, fishing\_activity\_category\_eu\_l6.

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
RCG_check_CL(data_exampleCL, MS = "COUNTRY1", GSA = "GSA99", SP = "Parapenaeus longirostris")
```

---

RCG_check_LFD	<i>Consistency check of LFDs</i>
---------------	----------------------------------

---

**Description**

The function allows to check the consistency of LFDs (length frequency distributions) by year on a given species generating a multi-frame plot. The function also returns the records in which the length classes are greater or lower than the expected ones (`min_len` and `max_len` parameters).

**Usage**

```
RCG_check_LFD(data, SP, MS, GSA, min_len = NA, max_len = NA, verbose = TRUE)
```

**Arguments**

data	RCG CS table
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
min_len	minimum length
max_len	maximum length
verbose	boolean. If TRUE messages are returned

**Value**

The function returns a comparison plot of LFDs among the years and a table reporting the length classes out of the expected range, according to the reference length provided.

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
RCG_check_LFD(
  data = data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus",
  min_len = 30, max_len = 300
)
RCG_check_LFD(
  data = data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus",
  min_len = NA, max_len = NA
)
```



---

`RCG_check_LFD_comm_cat`*Check consistency of LFD by year and commercial category*

---

**Description**

The function allows to check the consistency of LFDs (length frequency distributions) by year and commercial size category on a given species.

**Usage**

```
RCG_check_LFD_comm_cat(data, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

<code>data</code>	RCG CS table
<code>SP</code>	species code
<code>MS</code>	member state code
<code>GSA</code>	GSA code (Geographical sub-area)
<code>verbose</code>	boolean. If TRUE messages are returned

**Value**

The function returns a multi-frame plot of LFDs (length frequency distributions) by year and commercial size category for the selected species. The function also returns a data frame with the length range by year and commercial size category.

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
RCG_check_LFD_comm_cat(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus")
```

---

`RCG_check_loc`*Check trip location*

---

**Description**

The function allows to check the spatial distribution of data using the initial and final coordinates, where available, and the ports position included in the data in case coordinates are not available.

**Usage**

```
RCG_check_loc(data, MS = NA, GSA = NA, ports = circabc)
```

**Arguments**

data	detailed data in RCG CS format
MS	member state code
GSA	GSA code
ports	ports codification file

**Value**

A map of trip locations is generated.

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
RCG_check_loc(data_ex)
```

---

RCG\_check\_lw

*Consistency check of length-weight relationship*

---

**Description**

The function allows to check the consistency of length-weight relationship by sex and year on a given species generating a multi-frame plot. The function also returns the records in which the individual weights are greater or lower than the expected ones (Min and Max parameters). In case the Min and Max parameters are not provided, the function tests the lowest and the higher values of individual weights as outliers by mean of the Grubbs' test (from package outliers)

**Usage**

```
RCG_check_lw(data, SP, MS, GSA, Min = NA, Max = NA, verbose = TRUE)
```

**Arguments**

data	table of detailed data in RCG format
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
Min	min weight expected in the data
Max	max weight expected in the data
verbose	boolean. If it is TRUE messages are reported with the outputs

**Value**

Plot and error message

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**References**

Grubbs, F.E. (1950). Sample Criteria for testing outlying observations. *Ann. Math. Stat.* 21, 1, 27-58

**Examples**

```
RCG_check_lw(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus", Min = 0, Max = 1000)
RCG_check_lw(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus")
```

---

RCG\_check\_mat

*Check consistency sex and maturity stage*

---

**Description**

The function allows to perform a visual check of the maturity stage composition by length class, sex and year.

**Usage**

```
RCG_check_mat(data, MS, GSA, SP, verbose = TRUE)
```

**Arguments**

data	table of detailed data in RCG format
MS	member state code
GSA	GSA code
SP	reference species for the analysis
verbose	boolean. If it is TRUE messages are reported with the outputs

**Value**

The function the returns the plot of the maturity stages by length class

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
RCG_check_mat(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus")
```

---

RCG\_check\_mat\_ogive    *check\_mat\_ogive*

---

### Description

The function allows to check the maturity stages composition estimating the maturity ogives by sex for the selected species, using a selected pool of stages to classify the immature stages in the sample.

### Usage

```
RCG_check_mat_ogive(  
  data,  
  MS,  
  GSA,  
  SP,  
  sex,  
  immature_stages = c("0", "1", "2a"),  
  verbose = TRUE  
)
```

### Arguments

data	table of detailed data in RCG format
MS	member state code
GSA	GSA code (Geographical sub-area)
SP	species code
sex	defines the sex of the individuals selected for the analysis ('F' for females, 'M' for males)
immature_stages	vector of maturity stages considered immature
verbose	boolean. If TRUE messages are returned

### Value

Plot ogive by sex

### Author(s)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
RCG_check_mat_ogive(data_ex,  
  MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus",  
  sex = "F", immature_stages = c("0", "1", "2a")  
)
```

---

 RCG\_summarize\_ind\_meas

*Number of individual by trip for which biological data have been collected (length, sex, maturity, weight and age)*

---

### Description

Number of individual by trip for which biological data have been collected (length, sex, maturity, weight and age)

### Usage

```
RCG_summarize_ind_meas(data, MS, GSA, SP, verbose = TRUE)
```

### Arguments

data	detailed data in RCG CS format
MS	member state code
GSA	GSA code
SP	species reference code in the three alpha code format
verbose	boolean value to obtain further explanation messages from the function

### Value

The function returns a table containing a summary of measurements by trip for each biological variable

### Author(s)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

### Examples

```
RCG_summarize_ind_meas(data = data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus")
```

---

 RCG\_summarize\_trips *summarizing the number of trips/hauls monitored by year by port, metier, sampling method;*


---

### Description

The function allows to summarise monitored by year by port, metier, sampling method

### Usage

```
RCG_summarize_trips(data, SP, MS, GSA, verbose = TRUE)
```

**Arguments**

data	RCG CS table
SP	species code
MS	member state code
GSA	GSA code (Geographical sub-area)
verbose	boolean. If TRUE messages are returned

**Value**

Number of trips by area, year, port, metier and sampling method

**Author(s)**

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@coispa.it)

Walter Zupa [zupa@coispa.it](mailto:zupa@coispa.it)

**Examples**

```
RCG_summarize_trips(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus")
```

---

SA_tab_example	<i>SA table in MED&amp;BS datacall format</i>
----------------	---

---

**Description**

SA table in MED&BS datacall format

**Usage**

```
SA_tab_example
```

**Format**

An object of class `data.frame` with 74 rows and 9 columns.

**Author(s)**

Isabella Bitetto <[bitetto@coispa.it](mailto:bitetto@coispa.it)>

---

sex_mat	<i>sex_mat in GFCM datacall format</i>
---------	--

---

**Description**

sex\_mat in GFCM datacall format

**Usage**

sex\_mat

**Format**

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 42 rows and 5 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

SL_tab_example	<i>SL table in MED&amp;BS datacall format</i>
----------------	---

---

**Description**

SL table in MED&BS datacall format

**Usage**

SL\_tab\_example

**Format**

An object of class `data.frame` with 454 rows and 10 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

SSPP

*SSPP table*

---

**Description**

SSPP table

**Usage**

SSPP

**Format**

An object of class `data.frame` with 13417 rows and 8 columns.

**Author(s)**

Walter Zupa <zupa@fondazionecoispa.org>

---

task\_ii2

*task\_ii2 in GFCM DCRF datacall format*

---

**Description**

task\_ii2 in GFCM DCRF datacall format

**Usage**

task\_ii2

**Format**

An object of class `data.frame` with 5 rows and 9 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>



---

`task_iii`*task\_iii in GFCM DCRF datacall format*

---

**Description**

`task_iii` in GFCM DCRF datacall format

**Usage**

```
task_iii
```

**Format**

An object of class `data.frame` with 43 rows and 16 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

`task_vii2`*task\_vii2 in GFCM DCRF datacall format*

---

**Description**

`task_vii2` in GFCM DCRF datacall format

`task_vii31` in GFCM DCRF datacall format

**Usage**

```
task_vii2
```

```
task_vii2
```

**Format**

An object of class `data.frame` with 20 rows and 13 columns.

An object of class `data.frame` with 20 rows and 13 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

`task_vii31`*task\_vii31 in GFCM datacall format*

---

**Description**

`task_vii31` in GFCM datacall format

**Usage**

`task_vii31`

**Format**

An object of class `data.frame` with 5 rows and 8 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

`task_vii32`*task\_vii32 in GFCM DCRF datacall format*

---

**Description**

`task_vii32` in GFCM DCRF datacall format

**Usage**

`task_vii32`

**Format**

An object of class `data.frame` with 10 rows and 15 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

# Index

- \* **CIRCABC**
  - circabc, [43](#)
- \* **CL**,
  - data\_exampleCL, [44](#)
- \* **CS**
  - data\_ex, [44](#)
- \* **DCRF**
  - task\_ii2, [104](#)
  - task\_iii, [105](#)
  - task\_vii2, [105](#)
  - task\_vii32, [106](#)
- \* **DGMAREMED&BS**
  - fdi\_a\_catch, [46](#)
  - fdi\_g\_effort, [56](#)
  - fdi\_h\_spatial\_land, [57](#)
  - fdi\_h\_spatial\_landings, [57](#)
  - fdi\_i\_spatial\_effort, [58](#)
  - fdi\_j\_capacity, [58](#)
- \* **FDI**
  - fdi\_a\_catch, [46](#)
  - fdi\_g\_effort, [56](#)
  - fdi\_h\_spatial\_land, [57](#)
  - fdi\_h\_spatial\_landings, [57](#)
  - fdi\_i\_spatial\_effort, [58](#)
  - fdi\_j\_capacity, [58](#)
- \* **GFCM**
  - catfau\_check, [6](#)
  - combination\_taskII2, [43](#)
  - minmaxLtaskVII2, [93](#)
  - minmaxLtaskVII31, [93](#)
  - sex\_mat, [103](#)
  - task\_ii2, [104](#)
  - task\_iii, [105](#)
  - task\_vii2, [105](#)
  - task\_vii31, [106](#)
  - task\_vii32, [106](#)
- \* **GSAlist**
  - GSAlist, [65](#)
- \* **Harbour**
  - circabc, [43](#)
- \* **MEDBSSP**
  - MEDBSSP, [66](#)
- \* **MED&BS**
  - ALK\_tab\_example, [5](#)
  - Catch\_tab\_example, [5](#)
  - Discard\_tab\_example, [45](#)
  - GP\_tab\_example, [64](#)
  - Landing\_tab\_example, [65](#)
  - MA\_tab\_example, [66](#)
  - ML\_tab\_example, [94](#)
  - SA\_tab\_example, [102](#)
  - SL\_tab\_example, [103](#)
- \* **RCG**,
  - data\_exampleCL, [44](#)
- \* **RCG**
  - data\_ex, [44](#)
- \* **SSPP**
  - SSPP, [104](#)
- \* **aer\_catch**
  - aer\_catch, [4](#)
- \* **datacall**
  - ALK\_tab\_example, [5](#)
  - Catch\_tab\_example, [5](#)
  - catfau\_check, [6](#)
  - combination\_taskII2, [43](#)
  - Discard\_tab\_example, [45](#)
  - fdi\_a\_catch, [46](#)
  - fdi\_g\_effort, [56](#)
  - fdi\_h\_spatial\_land, [57](#)
  - fdi\_h\_spatial\_landings, [57](#)
  - fdi\_i\_spatial\_effort, [58](#)
  - fdi\_j\_capacity, [58](#)
  - GP\_tab\_example, [64](#)
  - Landing\_tab\_example, [65](#)
  - MA\_tab\_example, [66](#)
  - minmaxLtaskVII2, [93](#)
  - minmaxLtaskVII31, [93](#)
  - ML\_tab\_example, [94](#)
  - SA\_tab\_example, [102](#)
  - sex\_mat, [103](#)
  - SL\_tab\_example, [103](#)
  - task\_ii2, [104](#)
  - task\_iii, [105](#)
  - task\_vii2, [105](#)
  - task\_vii31, [106](#)
  - task\_vii32, [106](#)

- \* **data**
  - data\_exampleCL, 44
- \* **landing**
  - data\_exampleCL, 44
- \* **samplingData**
  - data\_ex, 44
- aer\_catch, 4
- ALK\_tab\_example, 5
- Catch\_tab\_example, 5
- catfau\_check, 6
- check\_age\_MEDBS\_AR, 6
- check\_cs\_header, 7
- check\_EF\_FDI\_A, 8
- check\_EF\_FDI\_G, 8
- check\_EF\_FDI\_H, 9
- check\_EF\_FDI\_I, 10
- check\_EF\_FDI\_J, 10
- check\_EF\_taskII2, 11
- check\_EF\_taskIII, 12
- check\_EF\_taskVII2, 12
- check\_EF\_TaskVII31, 13
- check\_EF\_TaskVII32, 14
- check\_gfcm\_header, 14
- check\_l50\_TaskVII.3.1, 15
- check\_ldf\_TaskVII.2, 16
- check\_lengths\_MEDBS\_AR, 16
- check\_lmat\_TaskVII.3.2, 17
- check\_lw\_TaskVII.2, 18
- check\_lw\_TaskVII.3.2, 19
- check\_maturity\_MEDBS\_AR, 20
- check\_minmaxl50\_TaskVII.3.1, 21
- check\_minmaxl\_TaskVII.2, 22
- check\_n\_trips\_MEDBS\_AR, 23
- check\_presence\_taskII2, 23
- check\_RD\_FDI\_A, 24
- check\_RD\_FDI\_G, 25
- check\_RD\_FDI\_H, 25
- check\_RD\_FDI\_I, 26
- check\_RD\_FDI\_J, 27
- check\_RD\_taskII2, 27
- check\_RD\_taskIII, 28
- check\_RD\_taskVII2, 29
- check\_RD\_taskVII31, 29
- check\_RD\_TaskVII32, 30
- check\_species\_catfau\_TaskVII.3.2, 31
- Check\_Tot\_Disc, 32
- Check\_Tot\_Disc\_gear, 33
- Check\_Tot\_Disc\_gear\_Q, 34
- Check\_Tot\_Disc\_metier, 35
- Check\_Tot\_Land, 36
- Check\_Tot\_Land2, 37
- Check\_Tot\_Land3, 38
- Check\_Tot\_Land\_gear, 39
- Check\_Tot\_Land\_gear\_Q, 40
- Check\_Tot\_Land\_metier, 41
- check\_weights\_MEDBS\_AR, 42
- circabc, 43
- combination\_taskII2, 43
- data\_ex, 44
- data\_exampleCL, 44
- Discard\_tab\_example, 45
- fdi\_a\_catch, 46
- FDI\_AER\_land\_landvalue, 45
- FDI\_check\_coord, 48
- FDI\_checks\_spatial\_HI, 47
- FDI\_cov\_tableA, 49
- FDI\_cov\_tableG, 50
- FDI\_cov\_tableJ, 51
- FDI\_coverage, 48
- FDI\_cross\_checks\_AG, 52
- FDI\_cross\_checks\_AH, 53
- FDI\_cross\_checks\_IG, 54
- FDI\_cross\_checks\_JG, 54
- FDI\_disc\_coverage, 55
- FDI\_fishdays\_cov, 56
- fdi\_g\_effort, 56
- fdi\_h\_spatial\_land, 57
- fdi\_h\_spatial\_landings, 57
- fdi\_i\_spatial\_effort, 58
- fdi\_j\_capacity, 58
- FDI\_landweight\_cov, 59
- FDI\_prices\_cov, 59
- FDI\_prices\_not\_null, 60
- FDI\_vessel\_lenth, 61
- FDI\_vessel\_numbers, 62
- GFCM\_check\_headers, 62
- GFCM\_cov\_II2, 63
- GFCM\_cov\_task\_iii, 64
- GP\_tab\_example, 64
- GSAlist, 65
- Landing\_tab\_example, 65
- MA\_tab\_example, 66
- MEDBS\_ALK, 67
- MEDBS\_Catch\_coverage, 67
- MEDBS\_check\_disaggregated, 68
- MEDBS\_check\_duplicates, 69
- MEDBS\_check\_missing\_years, 70
- MEDBS\_comp\_disc\_YQ, 71
- MEDBS\_comp\_disc\_YQ\_fishery, 72

MEDBS\_comp\_land\_Q\_VL, [72](#)  
MEDBS\_comp\_land\_Q\_VL\_fishery, [73](#)  
MEDBS\_comp\_land\_YQ, [74](#)  
MEDBS\_comp\_land\_YQ\_fishery, [75](#)  
MEDBS\_disc\_mean\_weight, [76](#)  
MEDBS\_discard\_coverage, [75](#)  
MEDBS\_GP\_check, [77](#)  
MEDBS\_ks, [78](#)  
MEDBS\_land\_mean\_weight, [79](#)  
MEDBS\_Landing\_coverage, [79](#)  
MEDBS\_length\_ind, [81](#)  
MEDBS\_lengthclass\_0, [80](#)  
MEDBS\_LFD, [82](#)  
MEDBS\_LW\_check, [83](#)  
MEDBS\_MA\_check, [83](#)  
MEDBS\_ML\_check, [84](#)  
MEDBS\_plot\_disc\_vol, [86](#)  
MEDBS\_plot\_discard\_ts, [85](#)  
MEDBS\_plot\_land\_vol, [87](#)  
MEDBS\_plot\_landing\_ts, [86](#)  
MEDBS\_SA\_check, [88](#)  
MEDBS\_SL\_check, [89](#)  
MEDBS\_SOP, [89](#)  
MEDBS\_weight\_0, [90](#)  
MEDBS\_weight\_minus1, [91](#)  
MEDBS\_yr\_missing\_length, [92](#)  
MEDBSSP, [66](#)  
minmaxLtaskVII2, [93](#)  
minmaxLtaskVII31, [93](#)  
ML\_tab\_example, [94](#)

RCG\_check\_AL, [94](#)  
RCG\_check\_CL, [95](#)  
RCG\_check\_LFD, [96](#)  
RCG\_check\_LFD\_comm\_cat, [97](#)  
RCG\_check\_loc, [97](#)  
RCG\_check\_lw, [98](#)  
RCG\_check\_mat, [99](#)  
RCG\_check\_mat\_ogive, [100](#)  
RCG\_summarize\_ind\_meas, [101](#)  
RCG\_summarize\_trips, [101](#)

SA\_tab\_example, [102](#)  
sex\_mat, [103](#)  
SL\_tab\_example, [103](#)  
SSPP, [104](#)

task\_ii2, [104](#)  
task\_iii, [105](#)  
task\_vii2, [105](#)  
task\_vii31, [106](#)  
task\_vii32, [106](#)