

# Package ‘RDBqc’

October 5, 2022

**Title** Quality check functions for RDBFIS

**Version** 0.0.11

**Date** 2022-10-05

**Maintainer** Isabella Bitetto <bitetto@coispa.eu>

**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** 7.2.1

**Depends** R (>= 4.1.0)

**Imports** dplyr, ggplot2, rworldmap, sp, rworldxtra, pander, data.table, grDevices, magrittr, tictoc, tidyverse, fishmethods, tidyr, gridExtra, scales

**LazyData** true

**Suggests** rmarkdown,  
knitr,  
markdown

**VignetteBuilder** knitr

## R topics documented:

|                                  |    |
|----------------------------------|----|
| ALK_tab_example . . . . .        | 4  |
| Catch_tab_example . . . . .      | 4  |
| catfau_check . . . . .           | 5  |
| check_EF_FDI_A . . . . .         | 5  |
| check_EF_FDI_G . . . . .         | 6  |
| check_EF_FDI_H . . . . .         | 6  |
| check_EF_FDI_I . . . . .         | 7  |
| check_EF_FDI_J . . . . .         | 7  |
| check_EF_taskII2 . . . . .       | 8  |
| check_EF_taskIII . . . . .       | 8  |
| check_EF_taskVII2 . . . . .      | 9  |
| check_EF_TaskVII31 . . . . .     | 9  |
| check_EF_TaskVII32 . . . . .     | 10 |
| check_lmat_TaskVII.3.2 . . . . . | 10 |

|  |    |
|--|----|
| check_lw_TaskVII.2 . . . . .               | 11 |
| check_minmax150_TaskVII.3.1 . . . . .      | 11 |
| check_minmax1_TaskVII.2 . . . . .          | 12 |
| check_presence_taskII2 . . . . .           | 13 |
| check_RD_FDI_A . . . . .                   | 13 |
| check_RD_FDI_G . . . . .                   | 14 |
| check_RD_FDI_H . . . . .                   | 14 |
| check_RD_FDI_I . . . . .                   | 15 |
| check_RD_FDI_J . . . . .                   | 15 |
| check_RD_taskII2 . . . . .                 | 16 |
| check_RD_taskIII . . . . .                 | 16 |
| check_RD_taskVII2 . . . . .                | 17 |
| check_RD_taskVII31 . . . . .               | 17 |
| check_RD_TaskVII32 . . . . .               | 18 |
| check_species_catfau_TaskVII.3.2 . . . . . | 18 |
| circabc . . . . .                          | 19 |
| combination_taskII2 . . . . .              | 19 |
| data_ex . . . . .                          | 20 |
| data_exampleCL . . . . .                   | 20 |
| Discard_tab_example . . . . .              | 21 |
| fdi_a_catch . . . . .                      | 21 |
| FDI_checks_spatial_HI . . . . .            | 22 |
| FDI_check_coord . . . . .                  | 22 |
| FDI_coverage . . . . .                     | 23 |
| FDI_cov_tableA . . . . .                   | 24 |
| FDI_cov_tableG . . . . .                   | 25 |
| FDI_cov_tableJ . . . . .                   | 26 |
| FDI_cross_checks_AG . . . . .              | 27 |
| FDI_cross_checks_AH . . . . .              | 27 |
| FDI_cross_checks_IG . . . . .              | 28 |
| FDI_cross_checks_JG . . . . .              | 28 |
| FDI_disc_coverage . . . . .                | 29 |
| FDI_fishdays_cov . . . . .                 | 29 |
| fdi_g_effort . . . . .                     | 30 |
| fdi_h_spatial_land . . . . .               | 30 |
| fdi_h_spatial_landings . . . . .           | 31 |
| fdi_i_spatial_effort . . . . .             | 31 |
| fdi_j_capacity . . . . .                   | 32 |
| FDI_landweight_cov . . . . .               | 32 |
| FDI_prices_cov . . . . .                   | 33 |
| FDI_prices_not_null . . . . .              | 33 |
| FDI_vessel_lenth . . . . .                 | 34 |
| FDI_vessel_numbers . . . . .               | 35 |
| GP_tab_example . . . . .                   | 35 |
| Landing_tab_example . . . . .              | 36 |
| MA_tab_example . . . . .                   | 36 |
| MEDBS_ALK . . . . .                        | 37 |
| MEDBS_Catch_coverage . . . . .             | 37 |
| MEDBS_check_duplicates . . . . .           | 38 |
| MEDBS_comp_disc_YQ . . . . .               | 39 |
| MEDBS_comp_disc_YQ_fishery . . . . .       | 39 |
| MEDBS_comp_land_Q_VL . . . . .             | 40 |

|  |    |
|--|----|
| MEDBS_comp_land_Q_VL_fishery . . . . . | 41 |
| MEDBS_comp_land_YQ . . . . .           | 42 |
| MEDBS_comp_land_YQ_fishery . . . . .   | 42 |
| MEDBS_discard_coverage . . . . .       | 43 |
| MEDBS_disc_mean_weight . . . . .       | 44 |
| MEDBS_GP_check . . . . .               | 44 |
| MEDBS_ks . . . . .                     | 45 |
| MEDBS_Landing_coverage . . . . .       | 46 |
| MEDBS_land_mean_weight . . . . .       | 46 |
| MEDBS_lengthclass_0 . . . . .          | 47 |
| MEDBS_length_ind . . . . .             | 48 |
| MEDBS_LW_check . . . . .               | 49 |
| MEDBS_MA_check . . . . .               | 50 |
| MEDBS_ML_check . . . . .               | 50 |
| MEDBS_plot_discard_ts . . . . .        | 51 |
| MEDBS_plot_disc_vol . . . . .          | 52 |
| MEDBS_plot_landing_ts . . . . .        | 52 |
| MEDBS_plot_land_vol . . . . .          | 53 |
| MEDBS_SA_check . . . . .               | 54 |
| MEDBS_SL_check . . . . .               | 55 |
| MEDBS_SOP . . . . .                    | 55 |
| MEDBS_weight_0 . . . . .               | 56 |
| MEDBS_weight_minus1 . . . . .          | 57 |
| MEDBS_yr_missing_length . . . . .      | 58 |
| minmaxLtaskVII2 . . . . .              | 59 |
| minmaxLtaskVII31 . . . . .             | 59 |
| ML_tab_example . . . . .               | 60 |
| RCG_check_AL . . . . .                 | 60 |
| RCG_check_CL . . . . .                 | 61 |
| RCG_check_LFD . . . . .                | 61 |
| RCG_check_LFD_comm_cat . . . . .       | 62 |
| RCG_check_loc . . . . .                | 63 |
| RCG_check_lw . . . . .                 | 63 |
| RCG_check_mat . . . . .                | 64 |
| RCG_check_mat_ogive . . . . .          | 64 |
| RCG_summarize_ind_meas . . . . .       | 65 |
| RCG_summarize_trips . . . . .          | 66 |
| SA_tab_example . . . . .               | 67 |
| sex_mat . . . . .                      | 67 |
| SL_tab_example . . . . .               | 68 |
| task_ii2 . . . . .                     | 68 |
| task_iii . . . . .                     | 69 |
| task_vii2 . . . . .                    | 69 |
| task_vii31 . . . . .                   | 70 |
| task_vii32 . . . . .                   | 70 |

---

|                 |  |
|-----------------|--|
| 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 16 rows and 4 columns.

**Author(s)**

Isabella Bitetto <bitetto@coispa.it>

---

|                |  |
|----------------|--|
| 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    | GFCM 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.

**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    | GFCM 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.

### Examples

```
check_EF_FDI_G(fdi_g_effort)
```

---

|                |  |
|----------------|--|
| check_EF_FDI_H | <i>Check empty fields in FDI H table</i> |
|----------------|--|

---

### 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    | GFCM 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.

### Examples

```
check_EF_FDI_H(fdi_h_spatial_landings)
```

---

`check_EF_FDI_I`*Check empty fields in FDI I table*

---

**Description**

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

**Usage**

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

**Arguments**

|                      |  |
|----------------------|--|
| <code>data</code>    | GFCM Task I table                      |
| <code>verbose</code> | 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.

**Examples**

```
check_EF_FDI_I(fdi_i_spatial_effort)
```

---

`check_EF_FDI_J`*Check empty fields in FDI J table*

---

**Description**

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

**Usage**

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

**Arguments**

|                      |  |
|----------------------|--|
| <code>data</code>    | GFCM Task J table                      |
| <code>verbose</code> | 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.

**Examples**

```
check_EF_FDI_J(fdi_j_capacity)
```

---

|                  |   |
|------------------|---|
| check_EF_taskII2 | <i>Check empty fields in GFCM Task II.2 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_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.

**Examples**

```
check_EF_taskII2(task_ii2)
```

---

|                  |  |
|------------------|--|
| check_EF_taskIII | <i>Check empty fields in GFCM Task III 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_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.

**Examples**

```
check_EF_taskIII(task_iii)
```



---

|                   |  |
|-------------------|--|
| check_EF_taskVII2 | <i>Check empty fields in GFCM Task VII.2 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_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.

**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.

**Examples**

```
check_EF_TaskVII31(task_vii31)
```

---

|                    |  |
|--------------------|--|
| check_EF_TaskVII32 | <i>Check empty fields in GFCM Task VII.3.2 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_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.

**Examples**

```
check_EF_TaskVII32(task_vii32)
```

---

|                        |  |
|------------------------|--|
| check_lmat_TaskVII.3.2 | <i>Plot of the maturity stages per length for each sex and species</i> |
|------------------------|--|

---

**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)
```

**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 |

**Value**

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

**Examples**

```
check_lmat_TaskVII.3.2(task_vii32)
```

---

|                    |  |
|--------------------|--|
| check_lw_TaskVII.2 | <i>Plot of the relationship length weight for each species</i> |
|--------------------|--|

---

**Description**

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

**Usage**

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

**Arguments**

|         |  |
|---------|--|
| data    | GFCM Task VII.2 table  |
| 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 return a plot of the length weight relationship per species.

**Examples**

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

---

|                             |   |
|-----------------------------|---|
| check_minmaxl50_TaskVII.3.1 | <i>Comparison between min/max L50 observed for each species and sex with theoretical values</i> |
|-----------------------------|---|

---

**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_minmaxl50_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.

**Examples**

```
check_minmaxl50_TaskVII.3.1(task_vii31, minmaxLtaskVII31, MS = "ITA", GSA = "19")
```

---

```
check_minmaxl_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_minmaxl_TaskVII.2(data, tab_length, MS, GSA)
```

**Arguments**

|            |   |
|------------|---|
| data       | GFCM Task II.2 table                                  |
| tab_length | Theoretical values of min/max length for each species |
| MS         | member state code                                     |
| GSA        | GSA code  |

**Value**

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

**Examples**

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

---

 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.

### Examples

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

---

 check\_RD\_FDI\_A

*Check duplicated records in FDI A table*


---

### 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    | GFCM 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.

**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    | GFCM 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.

**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    | GFCM 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.

**Examples**

```
check_RD_FDI_H(fdi_h_spatial_landings)
```

---

|                |  |
|----------------|--|
| check_RD_FDI_I | <i>Check duplicated records in FDI I 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_I(data, verbose = TRUE)
```

**Arguments**

|         |  |
|---------|--|
| data    | GFCM 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.

**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    | GFCM 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.

**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.

**Examples**

```
check_RD_taskII2(task_ii2)
```

---

|                  |  |
|------------------|--|
| check_RD_taskIII | <i>Check duplicated records in GFCM Task III 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_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.



**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.

**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.

**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.

**Examples**

```
check_RD_TaskVII32(task_vii32)
```

---

|                                  |  |
|----------------------------------|--|
| check_species_catfau_TaskVII.3.2 | <i>Check mismatching species/Catfau and Sex per maturity stages for Task VII.3.2 table</i> |
|----------------------------------|--|

---

**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)
```

**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   |

**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.

**Examples**

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

---

|         |  |
|---------|--|
| circabc | <i>Ports coordinates according to codification CIRCABC</i> |
|---------|--|

---

**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 <<https://circabc.europa.eu>>

---

|                     |  |
|---------------------|--|
| combination_taskII2 | <i>combination_taskII2 in GFCM datacall format</i> |
|---------------------|--|

---

**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`*RCG CS example*

---

**Description**

RCG CS example

**Usage**`data_ex`**Format**An object of class `data.frame` with 9099 rows and 40 columns.**Author(s)**

Isabella Bitetto &lt;bitetto@coispa.it&gt;

---

`data_exampleCL`*RCG CL example*

---

**Description**

RCG CL example

**Usage**`data_exampleCL`**Format**An object of class `data.frame` with 35 rows and 22 columns.**Author(s)**

Isabella Bitetto &lt;bitetto@coispa.it&gt;

---

|                     |  |
|---------------------|--|
| Discard_tab_example | <i>Discard table in MED&amp;BS datacall format</i> |
|---------------------|--|

---

**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_a_catch | <i>fdi_a_catch in FDI DGMAREMED&amp;BS datacall format</i> |
|-------------|--|

---

**Description**

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

**Usage**

fdi\_a\_catch

**Format**

An object of class `data.frame` with 2450 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.

### 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                      *Compatibility of the geographical coordinates with rectangle type*

---

### 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.

**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.

**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 | <i>Check number of record in FDI A table</i> |
|----------------|--|

---

### 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)).

### 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 number of record in FDI G table***Description**

The function checks and count the numbers of records 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

### 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 checks and count the numbers of records 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;

### 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 | <i>Cross check between FDI tables A and G</i> |
|---------------------|---|

---

**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 miss matches between landings and effort are shown.

**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 | <i>Cross check between FDI tables A and H</i> |
|---------------------|---|

---

**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 miss matches 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.

**Examples**

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

---

|                     |   |
|---------------------|---|
| FDI_cross_checks_IG | <i>Cross check between FDI tables I and G</i> |
|---------------------|---|

---

**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.

**Examples**

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

---

|                     |   |
|---------------------|---|
| FDI_cross_checks_JG | <i>Cross check between FDI tables J and G</i> |
|---------------------|---|

---

**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.

**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, verbose = TRUE)
```

**Arguments**

|         |  |
|---------|--|
| data    | data frame of the FDI table A          |
| MS      | Country                                |
| 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

**Examples**

```
FDI_disc_coverage(fdi_a_catch, MS = "PSP", 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

**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>

---

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

---

**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 | <i>fdi_j_capacity in FDI DGMAREMED&amp;BS datacall format</i> |
|----------------|---|

---

**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 | <i>Coverage of weight of landings in FDI table A and H</i> |
|--------------------|--|

---

**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.

**Examples**

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



---

|                |  |
|----------------|--|
| FDI_prices_cov | <i>Check prices trend in FDI A table</i> |
|----------------|--|

---

**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)).

**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 | <i>Check of species value</i> |
|---------------------|-------------------------------|

---

**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, SP = NA, verbose = FALSE)
```

**Arguments**

|         |  |
|---------|--|
| data    | data frame of FDA table A catch.                                   |
| MS      | country 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.

**Examples**

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

---

FDI\_vessel\_lenth

---

*Check of vessel lenght in FDI table J*


---

**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.

**Examples**

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

---

|                    |   |
|--------------------|---|
| FDI_vessel_numbers | <i>Check number of vessels in FDI table J and G</i> |
|--------------------|---|

---

**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.

**Examples**

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

---

|                |   |
|----------------|---|
| GP_tab_example | <i>GP table in MED&amp;BS datacall format</i> |
|----------------|---|

---

**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>

---

|                     |  |
|---------------------|--|
| Landing_tab_example | <i>Landing table in MED&amp;BS datacall format</i> |
|---------------------|--|

---

**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>

---

|           |                                |
|-----------|--------------------------------|
| MEDBS_ALK | <i>Plot of Age-Length Keys</i> |
|-----------|--------------------------------|

---

**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 (three alpha code)   |
| MS      | Country  |
| GSA     | GSA (Geographical sub-area (GFCM sensu))                               |
| verbose | boolean value to obtain further explanation messages from the function |

**Value**

a list of ALK plots by sex is returned

**Examples**

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

---

|                      |   |
|----------------------|---|
| MEDBS_Catch_coverage | <i>Catch_cov: function to check the coverage in Catch table</i> |
|----------------------|---|

---

**Description**

The function allows to check the coverage in 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 (three alpha code)   |
| MS      | Country  |
| GSA     | GSA (Geographical sub-area (GFCM sensu))                               |
| verbose | boolean value to obtain further explanation messages from the function |

**Value**

summary table and plots

**Examples**

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

---

MEDBS\_check\_duplicates

*Check for duplicated data rows*

---

**Description**

The function checks the presence of duplicated rows in both landings and discards data.

**Usage**

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

**Arguments**

|         |   |
|---------|---|
| data    | data frame containing landing data  |
| type    | string vector indicating the type of table to be checked. "l" for landing; "d" for discards; "c" for catch table. |
| SP      | vector of the species reference codes in the three alpha code format  |
| MS      | member state code   |
| GSA     | vector of the GSA codes to be included in the check   |
| verbose | Boolean value to obtain further explanation messages from the function  |

**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)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@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_comp_disc_YQ | <i>Comparison between discards in weight by quarter and -1</i> |
|--------------------|--|

---

### Description

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

### Usage

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

### Arguments

|      |  |
|------|--|
| data | data frame containing discards data                      |
| MS   | member state code as it is reported in the discards data |
| GSA  | GSA code   |
| SP   | species reference code in the three alpha code format    |

### Value

The function returns a data frame for 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)

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

### Examples

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

---

|                            |  |
|----------------------------|--|
| MEDBS_comp_disc_YQ_fishery | <i>Comparison between discards in weight by quarter, quarter -1 and by fishery</i> |
|----------------------------|--|

---

### 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, MS, GSA, SP, verbose = TRUE)
```

**Arguments**

|         |  |
|---------|--|
| data    | data frame containing discards data                                    |
| MS      | member state code as it is reported in the discards data               |
| 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 data frame for 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)

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

**Examples**

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

---

|                      |  |
|----------------------|--|
| MEDBS_comp_land_Q_VL | <i>Comparison between landings in weight by quarter accounting for vessel length</i> |
|----------------------|--|

---

**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 reference code in the three alpha code format                  |
| MS      | member state code  |
| GSA     | GSA code   |
| verbose | boolean value to obtain further explanation messages from the function |

**Value**

The function returns a dataframe 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)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@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 allows to perform 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 reference code in the three alpha code format                  |
| MS      | member state code  |
| GSA     | GSA code   |
| verbose | boolean value to obtain further explanation messages from the function |

**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)

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

**Examples**

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

---

|                    |  |
|--------------------|--|
| MEDBS_comp_land_YQ | <i>Comparison between landings in weight by quarter and -1</i> |
|--------------------|--|

---

### 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 reference code in the three alpha code format                  |
| MS      | member state code  |
| GSA     | GSA code   |
| verbose | boolean value to obtain further explanation messages from the function |

### 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)

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

### Examples

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

---

|                            |  |
|----------------------------|--|
| MEDBS_comp_land_YQ_fishery | <i>Comparison between landings in weight by quarter; quarter -1 and by fishery</i> |
|----------------------------|--|

---

### 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 reference code in the three alpha code format                  |
| MS      | member state code  |
| GSA     | GSA code   |
| verbose | boolean value to obtain further explanation messages from the function |

**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)

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

**Examples**

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

---

MEDBS\_discard\_coverage

*Discard\_cov: function to check the coverage in discard table*

---

**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 (three alpha code)   |
| MS          | Country  |
| GSA         | GSA (Geographical sub-area (GFCM sensu))                               |
| verbose     | boolean value to obtain further explanation messages from the function |

**Value**

summary table and plots of discard time series by year and gear

**Examples**

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

---

MEDBS\_disc\_mean\_weight

*Mean weight by year, gear and fishery aggregation*


---

### 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 reference code in the three alpha code format                  |
| MS      | member state code as it is reported in the discards data               |
| GSA     | GSA code   |
| verbose | boolean value to obtain further explanation messages from the function |

### 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)

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

### Examples

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

---

MEDBS\_GP\_check

*GP\_tab (growth params) table check*


---

### Description

The function allows to check the growth parameters by sex and year for a selected species

### Usage

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

**Arguments**

|      |   |
|------|---|
| data | growth params table in MED&BS datacall format |
| SP   | species (three alpha code)                    |
| MS   | Country                                       |
| GSA  | GSA (Geographical sub-area (GFCM sensu))      |

**Value**

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

**Examples**

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

---

|          |                                |
|----------|--------------------------------|
| MEDBS_ks | <i>Kolmogorov-Smirnov test</i> |
|----------|--------------------------------|

---

**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 landings or discards data  |
| type    | type of data frame. "l" for landing and "d" for discard                                |
| SP      | species reference code in the three alpha code format                                  |
| MS      | member state code  |
| GSA     | GSA code   |
| Rt      | ratio to be applied to subsample data to reduce the risk of rejection of H0 Hypothesis |
| verbose | boolean value to obtain further explanation messages from the function                 |

**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)

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

**Examples**

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

---

```
MEDBS_Landing_coverage
```

*Landing\_cov: function to check the coverage in 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 (three alpha code)   |
| MS      | Country  |
| GSA     | GSA (Geographical sub-area (GFCM sensu))                               |
| verbose | boolean value to obtain further explanation messages from the function |

**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 reference code in the three alpha code format                  |
| MS      | member state code  |
| GSA     | GSA code   |
| verbose | boolean value to obtain further explanation messages from the function |

**Value**

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

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

---

MEDBS\_lengthclass\_0     *Length classes number 0 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 data   |
| type    | string vector indicating the type of table to be checked. "l" for landing; "d" for discards. |
| SP      | species reference code in the three alpha code format  |
| MS      | member state code  |
| GSA     | GSA code   |
| verbose | Boolean value to obtain further explanation messages from the function                       |

**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)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@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 landings or discards data                                |
| type       | type of data frame. "l" for landing and "d" for discard                |
| SP         | species reference code in the three alpha code format                  |
| MS         | member state code  |
| GSA        | GSA code   |
| splines    | spline values assignment to fit cumulative distributions               |
| Xtresholds | threshold value  |
| verbose    | boolean value to obtain further explanation messages from the function |

**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)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@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\_LW\_check

*LW params in GP\_tab in table check*


---

**Description**

The function allows to check the length-weight parameters included in the GP table for a selected species

**Usage**

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

**Arguments**

|         |  |
|---------|--|
| data    | growth params table in MED&BS datacall format                          |
| SP      | species (three alpha code)   |
| MS      | Country  |
| GSA     | GSA (Geographical sub-area (GFCM sensu))                               |
| verbose | boolean value to obtain further explanation messages from the function |

**Value**

a summary table and plots of the LW params

**Examples**

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

---

|                |   |
|----------------|---|
| MEDBS_MA_check | <i>MA_tab (maturity at age) table check</i> |
|----------------|---|

---

### Description

The function allows to check 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 (three alpha code)   |
| MS      | Country  |
| GSA     | GSA (Geographical sub-area (GFCM sensu))                               |
| verbose | boolean value to obtain further explanation messages from the function |

### Value

a summary table and plots

### Examples

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

---

|                |  |
|----------------|--|
| MEDBS_ML_check | <i>ML_tab (maturity at length) table check</i> |
|----------------|--|

---

### Description

The function allows to check 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 (three alpha code)   |
| MS      | Country  |
| GSA     | GSA (Geographical sub-area (GFCM sensu))                               |
| verbose | boolean value to obtain further explanation messages from the function |

**Value**

a summary table and plots

**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 reference code in the three alpha code format  |
| MS   | member state code as it is reported in the discard data  |
| GSA  | GSA code   |
| 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)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@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 reference code in the three alpha code format   |
| MS   | member state code as it is reported in the discard data |
| GSA  | GSA code  |

### 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)

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

---

### 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 reference code in the three alpha code format  |
| MS      | member state code  |
| GSA     | GSA code   |
| by      | string defining the temporal aggregation level of landing data to be plotted. Allowed values are: "year" and "quarter" |
| verbose | boolean value to obtain further explanation messages from the function   |

**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 allows to visual check 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 reference code in the three alpha code format                  |
| MS      | member state code  |
| GSA     | GSA code   |
| verbose | boolean value to obtain further explanation messages from the function |

**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 | <i>SA_tab (sex ratio at age) table check</i> |
|----------------|--|

---

**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 (three alpha code)   |
| MS      | Country  |
| GSA     | GSA (Geographical sub-area (GFCM sensu))                               |
| verbose | boolean value to obtain further explanation messages from the function |

**Value**

a summary table and plots

**Examples**

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

---

|                |   |
|----------------|---|
| MEDBS_SL_check | <i>SL_tab (sex ratio at length) table check</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 datacall format                    |
| SP      | species (three alpha code)   |
| MS      | Country  |
| GSA     | GSA (Geographical sub-area (GFCM sensu))                               |
| verbose | boolean value to obtain further explanation messages from the function |

**Value**

a summary table and plots

**Examples**

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

---

|           |                       |
|-----------|-----------------------|
| MEDBS_SOP | <i>um of products</i> |
|-----------|-----------------------|

---

**Description**

um of products

**Usage**

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

**Arguments**

|           |   |
|-----------|---|
| data      | Catch table in MED&BS datacall format   |
| SP        | species (three alpha code)  |
| MS        | Country   |
| GSA       | GSA (Geographical sub-area (GFCM sensu))  |
| 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 value to obtain further explanation messages from the function  |

**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.

**Examples**

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

---

|                |  |
|----------------|--|
| MEDBS_weight_0 | <i>weight 0 in landings and discards</i> |
|----------------|--|

---

**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 reference code in the three alpha code format                  |
| MS      | member state code as it is reported in both landing and discard data   |
| GSA     | GSA code   |
| verbose | boolean value to obtain further explanation messages from the function |

**Value**

The function returns the number 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)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@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 | <i>weight -1 in landings</i> |
|---------------------|------------------------------|

---

## 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 reference code in the three alpha code format                  |
| MS      | member state code  |
| GSA     | GSA code   |
| verbose | Boolean value to obtain further explanation messages from the function |

## Value

The function returns the number 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)

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

## Arguments

|                   |   |
|-------------------|---|
| <code>data</code> | data frame of landings or discards data                 |
| <code>type</code> | type of data frame. "l" for landing and "d" for discard |
| <code>SP</code>   | species reference code in the three alpha code format   |
| <code>MS</code>   | member state code                                       |
| <code>GSA</code>  | GSA code  |

## 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)

Isabella Bitetto [bitetto@coispa.it](mailto:bitetto@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`*minmaxLtaskVII2 in GFCM datacall format*

---

**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`*minmaxLtaskVII31 in GFCM datacall format*

---

**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**

Check consistency of age-length relationship

**Usage**

```
RCG_check_AL(data, MS, GSA, SP, min_age = 0, max_age = 30, verbose = TRUE)
```

**Arguments**

|         |   |
|---------|---|
| data    | table of detailed data in RCG format                          |
| MS      | member state code   |
| GSA     | GSA code  |
| SP      | reference species for the analysis                            |
| min_age | minimum age expected  |
| max_age | maximum age expected  |
| verbose | boolean. If it is TRUE messages are reported with the outputs |

**Value**

summary table length-age and error (if any)

**Examples**

```
RCG_check_AL(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus", min_age = 0, max_age = 30)
```

---

|              |                                       |
|--------------|---------------------------------------|
| RCG_check_CL | <i>Quality checks on CL RCG table</i> |
|--------------|---------------------------------------|

---

### Description

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, foCatEu5, foCatEu6;
6. Sum of landing value by Year, Species, foCatEu5, foCatEu6.

### 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  |
| SP      | reference species for the analysis                            |
| verbose | boolean. If it is TRUE messages are reported with the outputs |

### Value

Checks\_CL list of tables for temporal, spatial, species and metier coverage

### Examples

```
RCG_check_CL(data_exampleCL, MS = "COUNTRY1", GSA = "GSA99", SP = "Parapenaeus longirostris")
```

---

|               |                      |
|---------------|----------------------|
| RCG_check_LFD | <i>RCG_check LFD</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, MS, GSA, SP, min_len = 1, max_len = 1000, verbose = TRUE)
```

**Arguments**

|         |   |
|---------|---|
| data    | RCG CS table  |
| MS      | member state code   |
| GSA     | GSA code  |
| SP      | reference species for the analysis                            |
| min_len | minimum length  |
| max_len | maximum length  |
| verbose | boolean. If it is TRUE messages are reported with the outputs |

**Value**

comparison plot of LFDs among the years and check of length range of the data using allowed range

**Examples**

```
RCG_check_LFD(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus", min_len = 1, max_len = 35)
```

---

```
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 generating a multi-frame plot. The function also returns a data frame with the length range by year and commercial size category.

**Usage**

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

**Arguments**

|         |   |
|---------|---|
| data    | RCG CS table  |
| 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**

plot and a summary table with ranges by year and commercial category

**Examples**

```
RCG_check_LFD_comm_cat(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus")
```

---

|               |                            |
|---------------|----------------------------|
| RCG_check_loc | <i>Check trip location</i> |
|---------------|----------------------------|

---

**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.

If Initial and/or Final coordinates are included in the data, maps of them are produced. If not the locations of the harbours are mapped.

**Usage**

```
RCG_check_loc(data, ports = circabc)
```

**Arguments**

|       |                                |
|-------|--------------------------------|
| data  | detailed data in RCG CS format |
| ports | ports codification file        |

**Value**

map of trip locations

**Examples**

```
RCG_check_loc(data_ex)
```

---

|              |  |
|--------------|--|
| RCG_check_lw | <i>Consistency of length-weight relationship and consistency with allowed ranges</i> |
|--------------|--|

---

**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).

**Usage**

```
RCG_check_lw(data, MS, GSA, SP, Min = 0, Max = 1000, verbose = TRUE)
```

**Arguments**

|         |   |
|---------|---|
| data    | table of detailed data in RCG format                          |
| MS      | member state code   |
| GSA     | GSA code  |
| SP      | reference species for the analysis                            |
| 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

**Examples**

```
RCG_check_lw(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus", Min = 0, Max = 1000)
```

---

|               |   |
|---------------|---|
| RCG_check_mat | <i>Check consistency sex and maturity stage</i> |
|---------------|---|

---

**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

**Examples**

```
RCG_check_mat(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus")
```

---

|                     |                               |
|---------------------|-------------------------------|
| RCG_check_mat_ogive | <i>Maturity ogives by sex</i> |
|---------------------|-------------------------------|

---

**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  |
| SP              | reference species for the analysis  |
| sex             | defines the sex of the individuals selected for the analysis ('F' for females, 'M' for males) |
| immature_stages | maturity stages considered immature   |
| verbose         | boolean. If it is TRUE messages are reported with the outputs                                 |

**Value**

Plot ogive by sex

**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**

a list containing a summary of measurements by trip for each biological variable

**Examples**

```
RCG_summarize_ind_meas(data = data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus")
```

---

|                     |  |
|---------------------|--|
| RCG_summarize_trips | <i>summarizing the number of trips/hauls monitored by year by port, metier, sampling method;</i> |
|---------------------|--|

---

**Description**

summarizing the number of trips/hauls monitored by year by port, metier, sampling method;

**Usage**

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

**Arguments**

|         |  |
|---------|--|
| data    | RCG CS table   |
| 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**

Number of trips by area, year, port, metier and sampling method

**Examples**

```
RCG_summarize_trips(data_ex, MS = "ITA", GSA = "GSA99", SP = "Mullus barbatus")
```

---

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

---

**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>

---

sex\_mat*sex\_mat in GFCM datacall format*

---

**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`*SL table in MED&BS datacall format*

---

**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>

---

`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 5 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 | <i>task_vii31 in GFCM datacall format</i> |
|------------|---|

---

**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 | <i>task_vii32 in GFCM DCRF datacall format</i> |
|------------|--|

---

**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, [19](#)
- \* **CL,**
  - data\_exampleCL, [20](#)
- \* **CS**
  - data\_ex, [20](#)
- \* **DCRF**
  - task\_ii2, [68](#)
  - task\_iii, [69](#)
  - task\_vii2, [69](#)
  - task\_vii32, [70](#)
- \* **DGMAREMED&BS**
  - fdi\_a\_catch, [21](#)
  - fdi\_g\_effort, [30](#)
  - fdi\_h\_spatial\_land, [30](#)
  - fdi\_h\_spatial\_landings, [31](#)
  - fdi\_i\_spatial\_effort, [31](#)
  - fdi\_j\_capacity, [32](#)
- \* **FDI**
  - fdi\_a\_catch, [21](#)
  - fdi\_g\_effort, [30](#)
  - fdi\_h\_spatial\_land, [30](#)
  - fdi\_h\_spatial\_landings, [31](#)
  - fdi\_i\_spatial\_effort, [31](#)
  - fdi\_j\_capacity, [32](#)
- \* **GFCM**
  - catfau\_check, [5](#)
  - combination\_taskII2, [19](#)
  - minmaxLtaskVII2, [59](#)
  - minmaxLtaskVII31, [59](#)
  - sex\_mat, [67](#)
  - task\_ii2, [68](#)
  - task\_iii, [69](#)
  - task\_vii2, [69](#)
  - task\_vii31, [70](#)
  - task\_vii32, [70](#)
- \* **Harbour**
  - circabc, [19](#)
- \* **MED&BS**
  - ALK\_tab\_example, [4](#)
  - Catch\_tab\_example, [4](#)
  - Discard\_tab\_example, [21](#)
  - GP\_tab\_example, [35](#)
  - Landing\_tab\_example, [36](#)
  - MA\_tab\_example, [36](#)
  - ML\_tab\_example, [60](#)
  - SA\_tab\_example, [67](#)
  - SL\_tab\_example, [68](#)
- \* **RCG,**
  - data\_exampleCL, [20](#)
- \* **RCG**
  - data\_ex, [20](#)
- \* **datacall**
  - ALK\_tab\_example, [4](#)
  - Catch\_tab\_example, [4](#)
  - catfau\_check, [5](#)
  - combination\_taskII2, [19](#)
  - Discard\_tab\_example, [21](#)
  - fdi\_a\_catch, [21](#)
  - fdi\_g\_effort, [30](#)
  - fdi\_h\_spatial\_land, [30](#)
  - fdi\_h\_spatial\_landings, [31](#)
  - fdi\_i\_spatial\_effort, [31](#)
  - fdi\_j\_capacity, [32](#)
  - GP\_tab\_example, [35](#)
  - Landing\_tab\_example, [36](#)
  - MA\_tab\_example, [36](#)
  - minmaxLtaskVII2, [59](#)
  - minmaxLtaskVII31, [59](#)
  - ML\_tab\_example, [60](#)
  - SA\_tab\_example, [67](#)
  - sex\_mat, [67](#)
  - SL\_tab\_example, [68](#)
  - task\_ii2, [68](#)
  - task\_iii, [69](#)
  - task\_vii2, [69](#)
  - task\_vii31, [70](#)
  - task\_vii32, [70](#)
- \* **data**
  - data\_exampleCL, [20](#)
- \* **landing**
  - data\_exampleCL, [20](#)
- \* **samplingData**
  - data\_ex, [20](#)
- ALK\_tab\_example, [4](#)

- Catch\_tab\_example, 4
- catfau\_check, 5
- check\_EF\_FDI\_A, 5
- check\_EF\_FDI\_G, 6
- check\_EF\_FDI\_H, 6
- check\_EF\_FDI\_I, 7
- check\_EF\_FDI\_J, 7
- check\_EF\_taskII2, 8
- check\_EF\_taskIII, 8
- check\_EF\_taskVII2, 9
- check\_EF\_TaskVII31, 9
- check\_EF\_TaskVII32, 10
- check\_lmat\_TaskVII.3.2, 10
- check\_lw\_TaskVII.2, 11
- check\_minmaxl50\_TaskVII.3.1, 11
- check\_minmaxl\_TaskVII.2, 12
- check\_presence\_taskII2, 13
- check\_RD\_FDI\_A, 13
- check\_RD\_FDI\_G, 14
- check\_RD\_FDI\_H, 14
- check\_RD\_FDI\_I, 15
- check\_RD\_FDI\_J, 15
- check\_RD\_taskII2, 16
- check\_RD\_taskIII, 16
- check\_RD\_taskVII2, 17
- check\_RD\_taskVII31, 17
- check\_RD\_TaskVII32, 18
- check\_species\_catfau\_TaskVII.3.2, 18
- circabc, 19
- combination\_taskII2, 19
- data\_ex, 20
- data\_exampleCL, 20
- Discard\_tab\_example, 21
- fdi\_a\_catch, 21
- FDI\_check\_coord, 22
- FDI\_checks\_spatial\_HI, 22
- FDI\_cov\_tableA, 24
- FDI\_cov\_tableG, 25
- FDI\_cov\_tableJ, 26
- FDI\_coverage, 23
- FDI\_cross\_checks\_AG, 27
- FDI\_cross\_checks\_AH, 27
- FDI\_cross\_checks\_IG, 28
- FDI\_cross\_checks\_JG, 28
- FDI\_disc\_coverage, 29
- FDI\_fishdays\_cov, 29
- fdi\_g\_effort, 30
- fdi\_h\_spatial\_land, 30
- fdi\_h\_spatial\_landings, 31
- fdi\_i\_spatial\_effort, 31
- fdi\_j\_capacity, 32
- FDI\_landweight\_cov, 32
- FDI\_prices\_cov, 33
- FDI\_prices\_not\_null, 33
- FDI\_vessel\_lenth, 34
- FDI\_vessel\_numbers, 35
- GP\_tab\_example, 35
- Landing\_tab\_example, 36
- MA\_tab\_example, 36
- MEDBS\_ALK, 37
- MEDBS\_Catch\_coverage, 37
- MEDBS\_check\_duplicates, 38
- MEDBS\_comp\_disc\_YQ, 39
- MEDBS\_comp\_disc\_YQ\_fishery, 39
- MEDBS\_comp\_land\_Q\_VL, 40
- MEDBS\_comp\_land\_Q\_VL\_fishery, 41
- MEDBS\_comp\_land\_YQ, 42
- MEDBS\_comp\_land\_YQ\_fishery, 42
- MEDBS\_disc\_mean\_weight, 44
- MEDBS\_discard\_coverage, 43
- MEDBS\_GP\_check, 44
- MEDBS\_ks, 45
- MEDBS\_land\_mean\_weight, 46
- MEDBS\_Landing\_coverage, 46
- MEDBS\_length\_ind, 48
- MEDBS\_lengthclass\_0, 47
- MEDBS\_LW\_check, 49
- MEDBS\_MA\_check, 50
- MEDBS\_ML\_check, 50
- MEDBS\_plot\_disc\_vol, 52
- MEDBS\_plot\_discard\_ts, 51
- MEDBS\_plot\_land\_vol, 53
- MEDBS\_plot\_landing\_ts, 52
- MEDBS\_SA\_check, 54
- MEDBS\_SL\_check, 55
- MEDBS\_SOP, 55
- MEDBS\_weight\_0, 56
- MEDBS\_weight\_minus1, 57
- MEDBS\_yr\_missing\_length, 58
- minmaxLtaskVII2, 59
- minmaxLtaskVII31, 59
- ML\_tab\_example, 60
- RCG\_check\_AL, 60
- RCG\_check\_CL, 61
- RCG\_check\_LFD, 61
- RCG\_check\_LFD\_comm\_cat, 62
- RCG\_check\_loc, 63
- RCG\_check\_lw, 63
- RCG\_check\_mat, 64
- RCG\_check\_mat\_ogive, 64



RCG\_summarize\_ind\_meas, [65](#)

RCG\_summarize\_trips, [66](#)

SA\_tab\_example, [67](#)

sex\_mat, [67](#)

SL\_tab\_example, [68](#)

task\_ii2, [68](#)

task\_iii, [69](#)

task\_vii2, [69](#)

task\_vii31, [70](#)

task\_vii32, [70](#)