fecR: Checking the fishing trip input data

Finlay Scott[1], Nuno Prista[2] and Thomas Reilly[3]
1. European Commission, DG Joint Research Centre, Directorate D - Sustainable Resources, Unit D.02 Water and Marine Resources, via Enrico Fermi 2749, 21027 Ispra (VA), Italy
2. SLU, Swedish University of Agricultural Sciences, Havsfiskelaboratoriet, Turistgatan 5, 453 30 LYSEKIL
3. Marine Scotland Science, PO Box 101, 375 Victoria Road, Aberdeen, AB11 9DB, Scotland, UK

2017-09-08

Contents

1 Introduction

Using the effort calculations in *fecR* requires the input data to be in a particular format. This vignette introduces the *check_format()* function in *fecR* that can be used to check the structure of your input data before the effort is calculated.

The function checks:

- The type of the columns (e.g. the *gt* column should be numeric);
- The data columns (*depdate*, *retdate* and *fishdate*) are in the correct format;
- The time columns (*deptime* and *rettime*) are in the correct format;
- The gear code is included in the Master Data Register;
- The fishing area in included in the DCF level 3 (level 4 for Baltic);
- The ICES rectangle exists;
- The trip identifier is unique.

It currently does not check:

- That the date and times are sensible;

Some basic automatic corrections are offered (see the details below). However, any changes made to the data should be confirmed by the user after the function has executed.

The function returns the data set. If automatic corrections have been asked for, the returned data set will have the corrections. Changes are noted to the screen by the function.

2 Data structure

This table is adapted from the Nicosia report Annex. Each row in the table is a fishing operation. Each fishing operation is part of a fishing trip. Each fishing trip has the same vessel identifier, departure and return dates and times and trip ID.

Column name	Description	Format	Notes	Example
eunr_id	Vessel identifier, anonymous	Character string		"MyVessel1234"
loa	Vessel length in cm	Numeric		3654
gt	Gross tonnage	Numeric		355
kw	Engine power	Numeric		1251
trip_id	Unique identifier for fishing trip	Character string		"MyTrip1234"
depdate	Date of trip departure	Character string	8 numeric characters: YYYYMMDD	"20140214"
deptime	Time of trip departure	Character string	4 numeric characters: HHMM. HH and MM can be separated by a colon: HH:MM	"0745" or "0745"
retdate	Date of trip return	Character string	8 numeric characters: YYYYMMDD	"20140214"
rettime	Time of trip return	Character string	4 numeric characters: HHMM. HH and MM can be separated by a colon: HH:MM	"0745" or "0745"
fishdate	Date of fishing operation	Character string	8 numeric characters: YYYYMMDD	"20140214"
gear	Gear used for specific fishing operation	Character string	Gear must be listed in the Master Data Register	"OTB"

Column name	Description	Format	Notes	Example
gear_mesh_size	Mesh size in mm	Integer	Every mm will be considered as a different gear. For example, a gear with a mesh size of 81 will be considered as a different gear to one with a mesh size of 80. The data needs to be encoded so that size ranges have the same integer. For example, set all sizes in the range 80-89 as 80. A gear without a mesh, e.g. a long line, will have a mesh size of 0. Missing values are not allowed.	80
fishing_area	Area where the fishing operation took place. DCF level 3 (level 4 for Baltic)	Character string	Must be upper case. Missing values are not allowed.	"27.4.B"
economic_zone	Economic zone where the fishing operation took place	Character string	Must be one of "EU", "NOR" or "UNKNOWN"	"EU"
rectangle	Rectangle where fishing operation took place	Character string	For example, ICES rectangle or GSA + statistical area. No symbols are allowed, e.g. no ' to separate characters. Must be upper case. Note: GSAs not yet added to list	"39F8"

3 Workflow

The *calc_fishing_effort()* function in *fecR* will only work if the input data is correct. The data can be be prepared using a spreadsheet and then saved as a CSV file. This can then be read into R to used by *fecR*.

The simplest way of confirming if the input data is correct is to use the *check_format()* function in *fecR*. If the function executes with a positive message and no warnings, the data is OK and can be used with *calc_fishing_effort()*. If the data is not OK warnings are produced and informative messages written to the screen. The user should then make changes to the data (either in R or in the original CSV file) and try again.

As mentioned above, it is possible to call *check_format()* and ask for some basic automatic corrections. If any corrections are made messages are written to screen describing them. It is not possible to automatically correct for everything. The returned data set should be passed into *check_format()* again to see if it passes the checks.

4 Examples

In this section we show some examples of running *check_format()* with data and how the automatic correction be used.

First we load the library:

```
library(fecR)
```

4.1 Perfect data

In this test we invent some data that passes the checks without corrections. The data conists of two trips and four fishing activities.

```
okdata <- data.frame(
    eunr_id = "my_boat", loa = 2000, gt = 70, kw = 400,
    trip_id = c("trip1","trip1","trip2","trip2"),
    depdate = rep(c("20140718", "20141023"), each=2),
    deptime = rep(c("0615", "0730"), each=2),
    retdate = rep(c("20140719", "20141024"), each=2),
    rettime = rep(c("1830", "1615"), each=2),
    fishdate = c("20140718", "20140719", "20141023", "20141024"),
    gear = c("OTB","OTB","GN","GN"), gear_mesh_size = 80, fishing_area = "27.4.B",
    economic_zone = "EU", rectangle = "39F0",
    stringsAsFactors = FALSE
)
okdata
```

```
##   eunr_id  loa gt  kw trip_id  depdate deptime  retdate rettime fishdate
## 1 my_boat 2000 70 400   trip1 20140718    0615 20140719    1830 20140718
## 2 my_boat 2000 70 400   trip1 20140718    0615 20140719    1830 20140719
## 3 my_boat 2000 70 400   trip2 20141023    0730 20141024    1615 20141023
## 4 my_boat 2000 70 400   trip2 20141023    0730 20141024    1615 20141024
##   gear gear_mesh_size fishing_area economic_zone rectangle
## 1  OTB             80       27.4.B            EU      39F0
## 2  OTB             80       27.4.B            EU      39F0
## 3   GN             80       27.4.B            EU      39F0
## 4   GN             80       27.4.B            EU      39F0
```

We can check the data by calling *check_format()* without correction (the default setting):

```
test <- check_format(okdata)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "=========================="
## [1] "The returned data passes the check."
## [1] "=========================="
```

You can see that the function checks each of the columns before giving an output message saying that everything is OK. As there were no warnings we can now use this data in *calc_fishing_effort()* if we want to.

4.2 Data with an extra column

The input data has a strict number of columns and the names need to follow the example above. In this example, an extra column is added to the data. Without asking for corrections, *check_format()* will complain.

```
extra_col <- cbind(okdata, new_col = runif(nrow(okdata)))
test <- check_format(extra_col)
```

```
## [1] "=========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "=========================="
## [1] "Checking column names..."
## [1] "ATT: You have 1 or more unrecognised columns"
```

```
## Warning in check_format(extra_col): You have 1 or more unrecognised columns
```

```
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "=========================="
## [1] "Attention: There are problems with this data set."
## [1] "=========================="
```

You can see that a warning is produced and the output message indicates that there is a problem with the data.

We can run *check_format()* with the automatic corrections turned on:

```
test <- check_format(extra_col, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "ATT: You have 1 or more unrecognised columns"
## [1] "Attempting to fix this problem by removing the unrecognised columns"
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

You can see that there is a message about the extra column and that it will be removed. The returned data set has been corrected by removing the extra column. This means that if we call the check function on the returned, corrected data, it should pass the check without problem.

```
test2 <- check_format(test)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "============================"
## [1] "The returned data passes the check."
## [1] "============================"
```

4.3 Wrong column names

If one or more of your columns is named incorrectly, the data check complains. No automatic correction is possible for this problem. You will have to rename the columns yourself.

Note that the column names are case-sensitive.

```
wrong_col <- okdata
colnames(wrong_col)[3] <- "something"
test <- check_format(wrong_col)
```

```
## [1] "============================"
## [1] " STECF Transversal2 checks on formats"
## [1] "============================"
## [1] "Checking column names..."

## Warning in check_format(wrong_col): You are missing the columns: gt

## Attention: There are problems with this data set.
## You are missing one or more column. Exiting.
```

The warning message tells you which columns are missing.

4.4 Checking the *eunr_id* column

The *eunr_id* column is the vessel identifier. It should be a character string. If the column is not a character string it is possible to use the automatic correction to force it to be a character.

```
wrong_eunr_id <- okdata
# Force them to be numeric instead of character
wrong_eunr_id[,"eunr_id"] <- 1234
test <- check_format(wrong_eunr_id)
```

```
## [1] "============================"
## [1] " STECF Transversal2 checks on formats"
## [1] "============================"
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "ATT: Error in eunr_id column. Column must be a character string."

## Warning in character_column_check(x$eunr_id, "eunr_id", character_correct =
## correct, : ATT: Error in eunr_id column. Column must be a character string.

## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```
# With the automatic check
test <- check_format(wrong_eunr_id, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "ATT: Error in eunr_id column. Column must be a character string."
## [1] "Attempting to correct by forcing to character string"
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

If an entry is missing in the column (e.g. it is *NA* or empty) then check complains. It is not possible to automatically correct for missing data.

```
wrong_eunr_id <- okdata
# Set to be missing
wrong_eunr_id[1,"eunr_id"] <- as.character(NA)
test <- check_format(wrong_eunr_id)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "ATT: Missing code in the eunr_id column"
## [1] "Problem rows: 1"
```

```
## Warning in character_column_check(x$eunr_id, "eunr_id", character_correct =
## correct, : ATT: Missing code in the eunr_id column
```

```
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "ATT: trip_id does not uniquely identify fishing_trip:"
## [1] "e.g.:"
##   eunr_id trip_id  depdate deptime  retdate rettime
## 1    <NA>   trip1 20140718    0615 20140719    1830
## 2 my_boat   trip1 20140718    0615 20140719    1830
```

```
## Warning in check_format(wrong_eunr_id): Trips must have unique combination
## of eunr_id, depdate, deptime, retdate and rettime
```

```
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

4.5 Checking the *loa*, *gt* and *kw* columns

The *loa*, *gt* and *kw* columns store the vessel length in cm, the gross tonnage and the engine power respectively. These columns must be numeric, i.e. no units or characters. If they are not numeric check complains.

Here we demonstrate with the *loa* column.

```
wrong_loa <- okdata
# Turn to a character string
```

```r
wrong_loa[c(2,3),"loa"] <- "90m"
test <- check_format(wrong_loa)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "ATT: loa column must be numeric"
```

```
## Warning in check_format(wrong_loa): Error in loa column
```

```
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

If automatic correction is turned on, the columns are stripped of non-numeric characters and forced to be numeric. This may be enough to pass check. However, this correction is not a guarantee and all automatic corrections should be verified by the user.

```r
test <- check_format(wrong_loa, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "ATT: loa column must be numeric"
## [1] "Attempting to correct error by removing non-numeric characters"
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

If there are no numerics in the columns automatic correction is not possible and and check complains.

```
wrong_loa <- okdata
# Change to some entries to be alphabetical with no numerics
wrong_loa[c(2,3),"loa"] <- "notnumeric"
test <- check_format(wrong_loa, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "ATT: loa column must be numeric"
## [1] "Attempting to correct error by removing non-numeric characters"
## [1] "Unable to force to numeric. There is a problem."
```

```
## Warning in check_format(wrong_loa, correct = TRUE): Error in loa column
```

```
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

This error will need to be fixed by hand.

4.6 Checking date columns

The date columns *depdate*, *retdate* and *fishdate* must be characters and each entry must have 8 numeric characters of the format: *YYYYMMDD*, e.g. “20161023”.

It is possible to automatically correct for the column not being a character string, e.g. if an 8 character numeric is entered. However, it is not possible to correct for the the format, e.g. if there are too few characters.

Here the data is numeric when it should be a character string. Automatic correction is possible in this case.

```r
wrong_date <- okdata
# Needs to be character string, not numeric even if format is OK
wrong_date[, "retdate"] <- as.numeric(wrong_date[, "retdate"])
test <- check_format(wrong_date)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "ATT: Error in retdate column. Column must be a character string."

## Warning in character_column_check(x[, datecol], datecol, character_correct
## = correct, : ATT: Error in retdate column. Column must be a character
## string.

## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```r
# We can correct
test <- check_format(wrong_date, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "ATT: Error in retdate column. Column must be a character string."
## [1] "Attempting to correct by forcing to character string"
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

If the format is wrong then we cannot automatically correct and check complains.

```
wrong_date <- okdata
# Wrong format - year is too short
wrong_date[c(3,4), "retdate"] <- "141024"
test <- check_format(wrong_date)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "ATT: Error in retdate column. Character string of format: yyyymmdd needed."
## [1] "Problem rows: 3, 4"
```

```
## Warning in check_format(wrong_date): ATT: Error in retdate column.
## Character string of format: yyyymmdd needed.
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```
# Wrong format again - month must be a numeric character
wrong_date[c(3,4), "retdate"] <- "October14"
test <- check_format(wrong_date)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "ATT: Error in retdate column. Character string of format: yyyymmdd needed."
## [1] "Problem rows: 3, 4"
```

```
## Warning in check_format(wrong_date): ATT: Error in retdate column.
## Character string of format: yyyymmdd needed.
```

```
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

Missing data is not allowed and check complains. It is not possible to correct for missing data.

```
wrong_date <- okdata
# Missing data
wrong_date[c(3,4), "retdate"] <- as.character(NA)
test <- check_format(wrong_date)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "ATT: Missing code in the retdate column"
## [1] "Problem rows: 3, 4"
```

```
## Warning in character_column_check(x[, datecol], datecol, character_correct
## = correct, : ATT: Missing code in the retdate column
```

```
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

4.7 Checking time columns

The time columns *deptime* and *rettime* must be character strings of 4 numeric characters with the format *HHMM*, e.g. "0615". An additional : is allowed to seperate the *HH* and *MM*, e.g. "06:15".

Note that the times use the 24 hour clock.

Here the data is numeric when it should be a character string. It is possible to automatically correct for this.

```
wrong_time <- okdata
# Needs to be character string, not numeric even if format is OK
wrong_time[, "rettime"] <- as.numeric(wrong_time[, "rettime"])
test <- check_format(wrong_time)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "ATT: Error in rettime column. Column must be a character string."
```

```
## Warning in character_column_check(x[, timecol], timecol, character_correct
## = correct, : ATT: Error in rettime column. Column must be a character
## string.
```

```
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```r
# We can correct by forcing to character
test <- check_format(wrong_time, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "ATT: Error in rettime column. Column must be a character string."
## [1] "Attempting to correct by forcing to character string"
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

If only 2 characters are provided and automatic correction is TRUE, the characters are assumed to be hours (*HH*) and minutes of “00” are appended to the string, e.g. “16” becomes “1600”.

```r
wrong_time <- okdata
wrong_time[c(3,4), "rettime"] <- "16"
test <- check_format(wrong_time)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "ATT: rettime needs 4 numeric characters: HHMM."
## [1] "4 numeric characters in rettime column are needed in rows: 3, 4"

## Warning in check_format(wrong_time): ATT: rettime needs 4 numeric
## characters: HHMM.

## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```r
test <- check_format(wrong_time, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Only 2 characters in rettime column in rows: 3, 4"
## [1] "Assuming these 2 are HH. Adding 00 as MM"
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

This automatic correction only works if the 2 characters are numeric

```
wrong_time <- okdata
wrong_time[c(3,4), "rettime"] <- "TT"
test <- check_format(wrong_time, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "ATT: rettime needs 4 numeric characters: HHMM."
## [1] "4 numeric characters in rettime column are needed in rows: 3, 4"
```

```
## Warning in check_format(wrong_time, correct = TRUE): ATT: rettime needs 4
## numeric characters: HHMM.
```

```
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

Two or four characters are needed. For example, “0130” is OK whereas “130” is not

```r
wrong_time <- okdata
wrong_time[c(3,4), "rettime"] <- "130"
test <- check_format(wrong_time)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "ATT: rettime needs 4 numeric characters: HHMM."
## [1] "4 numeric characters in rettime column are needed in rows: 3, 4"
```

```
## Warning in check_format(wrong_time): ATT: rettime needs 4 numeric
## characters: HHMM.
```

```
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

A : separator is accecptable (*HH:MM*) but is removed from the returned data.

```r
wrong_time <- okdata
wrong_time[c(3,4), "rettime"] <- "16:15"
test <- check_format(wrong_time)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "=========================="
## [1] "The returned data passes the check."
## [1] "=========================="
```

```
test
```

```
##   eunr_id  loa gt  kw trip_id  depdate deptime  retdate rettime fishdate
## 1 my_boat 2000 70 400   trip1 20140718    0615 20140719    1830 20140718
## 2 my_boat 2000 70 400   trip1 20140718    0615 20140719    1830 20140719
## 3 my_boat 2000 70 400   trip2 20141023    0730 20141024    1615 20141023
## 4 my_boat 2000 70 400   trip2 20141023    0730 20141024    1615 20141024
##   gear gear_mesh_size fishing_area economic_zone rectangle
## 1  OTB             80       27.4.B            EU      39F0
## 2  OTB             80       27.4.B            EU      39F0
## 3   GN             80       27.4.B            EU      39F0
## 4   GN             80       27.4.B            EU      39F0
```

*NA*s and missing values are not acceptable and cannot be automatically corrected.

```
wrong_time <- okdata
wrong_time[c(3,4), "rettime"] <- as.character(NA)
test <- check_format(wrong_time)
```

```
## [1] "=========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "=========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "ATT: Missing code in the rettime column"
## [1] "Problem rows: 3, 4"
```

```
## Warning in character_column_check(x[, timecol], timecol, character_correct
## = correct, : ATT: Missing code in the rettime column
```

```
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

4.8 Checking gear codes

The *gear* code column must be a character string and the code must be found in the Master Data Register. The check is case sensitive, e.g. "otb" is not valid whereas "OTB" is.

Whitespace is not allowed. It can be automatically removed if wanted.

```r
wrong_gear <- okdata
# Gear code is OK but whitespace
wrong_gear[1,"gear"] <- " OTB"
# Fails
test <- check_format(wrong_gear)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "ATT: Whitespace found in gear column"
## [1] "Problem rows: 1"
## [1] "ATT: Unknown code in gear column"
## [1] "Problem rows: 1"
```

```
## Warning in check_format(wrong_gear): ATT: Unknown code in gear column
```

```
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```r
# Correct removes whitespace
test <- check_format(wrong_gear, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "ATT: Whitespace found in gear column"
## [1] "Problem rows: 1"
## [1] "Attempting to correct by removing whitespace from gears"
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

If the gear code is not found in the MDR list then no automatic correction is possible. Unknown gear codes are not allowed and not corrected for. Here the gear is unknown because it is lower case. All gear codes must be upper case.

```
wrong_gear[1,"gear"] <- "otb"
test <- check_format(wrong_gear)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "ATT: Unknown code in gear column"
## [1] "Problem rows: 1"
```

```
## Warning in check_format(wrong_gear): ATT: Unknown code in gear column
```

```
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

4.9 Checking gear mesh size

The *gear_mesh_size* column must be an integer. It holds the mesh size in mm. Every mm is considered as a different gear, e.g. a gear with a mesh size of 80 is considered to be a different gear to that with a mesh size of 81. This means that gear meshes in the range 80-89 mm should all be given the same gear mesh size of 80.

Entries that are not integer will make check complain. There is no option to autocorrect this.

```
wrong_ms <- okdata
# Text in the entry - must be integer
wrong_ms[4,"gear_mesh_size"] <- "80mm"
test <- check_format(wrong_ms)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "ATT: Error in gear_mesh_size column. Only integers allowed."
## [1] "Problem rows: 4"
```

```
## Warning in check_format(wrong_ms): Error in gear_mesh_size column
```

```
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "============================"
## [1] "Attention: There are problems with this data set."
## [1] "============================"
```

```
# Not an integer
wrong_ms[4,"gear_mesh_size"] <- 80.8
test <- check_format(wrong_ms)
```

```
## [1] "============================"
## [1] " STECF Transversal2 checks on formats"
## [1] "============================"
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "ATT: Error in gear_mesh_size column. Only integers allowed."
## [1] "Problem rows: 4"

## Warning in check_format(wrong_ms): Error in gear_mesh_size column

## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "============================"
## [1] "Attention: There are problems with this data set."
## [1] "============================"
```

If an entry is missing (e.g. it is *NA*) then check will complain. It is possible to automatically correct this in which case the missing entry has a mesh size of 0. The returned data will pass check but may not be what you want.

```
wrong_ms <- okdata
wrong_ms[4,"gear_mesh_size"] <- NA
test <- check_format(wrong_ms)
```

```
## [1] "============================"
## [1] " STECF Transversal2 checks on formats"
## [1] "============================"
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "ATT: Missing gear_mesh_size. If no mesh, set to 0"
```

```
## Warning in check_format(wrong_ms): Missing gear_mesh_size
```

```
## [1] "Problem rows: 4"
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```r
test <- check_format(wrong_ms, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "ATT: Attempting to correct missing gear mesh sizes by setting mesh size to 0"
## [1] "Problem rows: 4"
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

```
test
```

```
##   eunr_id  loa gt  kw trip_id  depdate deptime  retdate rettime fishdate
## 1 my_boat 2000 70 400   trip1 20140718    0615 20140719    1830 20140718
## 2 my_boat 2000 70 400   trip1 20140718    0615 20140719    1830 20140719
## 3 my_boat 2000 70 400   trip2 20141023    0730 20141024    1615 20141023
## 4 my_boat 2000 70 400   trip2 20141023    0730 20141024    1615 20141024
##   gear gear_mesh_size fishing_area economic_zone rectangle
## 1  OTB             80       27.4.B            EU      39F0
## 2  OTB             80       27.4.B            EU      39F0
## 3   GN             80       27.4.B            EU      39F0
## 4   GN              0       27.4.B            EU      39F0
```

4.10 Checking the *fishing_area* column

The *fishing_area* column must be a character string that stores the DCF level 3 code (or DCF level 4 if in the Baltic). Whitespace is not allowed. However, it is possible to automatically correct for whitespace. Similarly, if points (.) are found at the beginning or end of an entry, it is possible to automatically correct for them. Note that the check is case sensitive and all the entries must be in upper case. It is possible to automatically correct for lower case.

```
wrong_fish_area <- okdata
# Point at end
wrong_fish_area[c(1,2),"fishing_area"] <- "27.4.A."
test <- check_format(wrong_fish_area)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "There is whitespace, leading or ending points, or lowercase characters in some of the data."

## Warning in check_format(wrong_fish_area): There is whitespace, leading or
## ending points, or lowercase characters in some of the data.

## [1] "ATT: Unknown areas in fishing_area column"
## [1] "Unknown fishing_area rows: 1, 2"

## Warning in check_format(wrong_fish_area): Problem in fishing_area column

## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```
test <- check_format(wrong_fish_area, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "There is whitespace, leading or ending points, or lowercase characters in some of the data."
## [1] "Attempting to correct these issues"
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

```
# Lowercase
wrong_fish_area[c(1,2),"fishing_area"] <- "27.4.a"
test <- check_format(wrong_fish_area)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "There is whitespace, leading or ending points, or lowercase characters in some of the data."

## Warning in check_format(wrong_fish_area): There is whitespace, leading or
## ending points, or lowercase characters in some of the data.

## [1] "ATT: Unknown areas in fishing_area column"
## [1] "Unknown fishing_area rows: 1, 2"

## Warning in check_format(wrong_fish_area): Problem in fishing_area column

## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```r
test <- check_format(wrong_fish_area, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "There is whitespace, leading or ending points, or lowercase characters in some of the data."
## [1] "Attempting to correct these issues"
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

```
# White space
wrong_fish_area[c(1,2),"fishing_area"] <- "27.4.A "
test <- check_format(wrong_fish_area)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "There is whitespace, leading or ending points, or lowercase characters in some of the data."
```

```
## Warning in check_format(wrong_fish_area): There is whitespace, leading or
## ending points, or lowercase characters in some of the data.
```

```
## [1] "ATT: Unknown areas in fishing_area column"
## [1] "Unknown fishing_area rows: 1, 2"
```

```
## Warning in check_format(wrong_fish_area): Problem in fishing_area column
```

```
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```
test <- check_format(wrong_fish_area, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "There is whitespace, leading or ending points, or lowercase characters in some of the data."
## [1] "Attempting to correct these issues"
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

Missing values are not allowed and cannot be automatically corrected.

```r
wrong_fish_area <- okdata
wrong_fish_area[c(1,2),"fishing_area"] <- as.character(NA)
test <- check_format(wrong_fish_area)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "ATT: Missing code in the fishing_area column"
## [1] "Problem rows: 1, 2"
```

```
## Warning in character_column_check(x$fishing_area, "fishing_area",
## character_correct = FALSE, : ATT: Missing code in the fishing_area column
```

```
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "============================"
```

4.11 Checking the *economic_zone* columm

The *economic_zone* column must be a character string and must be one of "EU", "NOR" or "UNKNOWN". The check is case sensitive. If the entries do not match these strings then check complains.

It is not possible to automatically correct any errors.

```
wrong_econ <- okdata
wrong_econ[3,"economic_zone"] <- "USA"
test <- check_format(wrong_econ)
```

```
## [1] "============================"
## [1] " STECF Transversal2 checks on formats"
## [1] "============================"
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "ATT: Unknown code in economic_zone column"
## [1] "Unknown economic_zone code rows: 3"
```

```
## Warning in check_format(wrong_econ): Problem in economic_zone column
```

```
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "============================"
## [1] "Attention: There are problems with this data set."
## [1] "============================"
```

Missing values are not allowed and cannot be automatically corrected.

```
wrong_econ <- okdata
wrong_econ[3,"economic_zone"] <- ""
test <- check_format(wrong_econ)
```

```
## [1] "============================"
## [1] " STECF Transversal2 checks on formats"
## [1] "============================"
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "ATT: Missing code in the economic_zone column"
## [1] "Problem rows: 3"
```

```
## Warning in character_column_check(x$economic_zone, "economic_zone",
## character_correct = FALSE, : ATT: Missing code in the economic_zone column
```

```
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "============================"
## [1] "Attention: There are problems with this data set."
## [1] "============================"
```

4.12 Checking ICES rectangles

The *rectangle* column must be a character string and each entry must be a valid ICES rectangle. If non alpha-numeric characters are found in the data it is possible to automatically correct for them by removing them. Similarly, the check is case sensitive but it is possible to automatically correct the case.

```r
wrong_rect <- okdata
# with extra punctuation
wrong_rect[3,"rectangle"] <- "39F0'"
test <- check_format(wrong_rect)
```

```
## [1] "============================"
## [1] " STECF Transversal2 checks on formats"
## [1] "============================"
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "ATT: Non alphanumeric or lowercase characters in ices rectangle column ."
## Warning in check_format(wrong_rect): ATT: Non alphanumeric or lowercase
## characters in ices rectangle column .
```

```
## [1] "ATT: Unknown code in rectangle column"
## [1] "Unknown rectangle code in rows: 3"
```

```
## Warning in check_format(wrong_rect): Problem in rectangle column
```

```
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```
test <- check_format(wrong_rect, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "ATT: Non alphanumeric or lowercase characters in ices rectangle column ."
## [1] "Attempting to correct this by removing them."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```

```
test
```

```
##   eunr_id  loa gt  kw trip_id  depdate deptime  retdate rettime fishdate
## 1 my_boat 2000 70 400   trip1 20140718    0615 20140719    1830 20140718
## 2 my_boat 2000 70 400   trip1 20140718    0615 20140719    1830 20140719
## 3 my_boat 2000 70 400   trip2 20141023    0730 20141024    1615 20141023
## 4 my_boat 2000 70 400   trip2 20141023    0730 20141024    1615 20141024
##   gear gear_mesh_size fishing_area economic_zone rectangle
## 1  OTB             80       27.4.B            EU      39F0
## 2  OTB             80       27.4.B            EU      39F0
## 3   GN             80       27.4.B            EU      39F0
## 4   GN             80       27.4.B            EU      39F0
```

Missing values are not allowed and cannot be automatically corrected.

```
wrong_rect <- okdata
wrong_rect[3,"rectangle"] <- ""
test <- check_format(wrong_rect)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "ATT: Missing code in the rectangle column"
## [1] "Problem rows: 3"
```

```
## Warning in character_column_check(x$rectangle, "rectangle",
## character_correct = FALSE, : ATT: Missing code in the rectangle column
```

```
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

4.13 Checking that the trip identifier is unique

Each trip is defined by the vessel identifier, start and return dates and times and has a unique trip identifier. A trip entry with the same trip ID cannot have different vessel IDs, dates and times.

For example, here we change the departure time of an entry for *trip2* so that the trip has different departure times.

```
# one trip, two days, different departure time, same identifier
wrong_unique <- okdata
wrong_unique[4,"deptime"] <- "0731"
test <- check_format(wrong_unique)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "ATT: trip_id does not uniquely identify fishing_trip:"
## [1] "e.g.:"
##   eunr_id trip_id  depdate deptime  retdate rettime
## 3 my_boat   trip2 20141023    0730 20141024    1615
## 4 my_boat   trip2 20141023    0731 20141024    1615
```

```
## Warning in check_format(wrong_unique): Trips must have unique combination
## of eunr_id, depdate, deptime, retdate and rettime
```

```
## [1] "Checking duplicates..."
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

4.14 Checking for duplicate entries

There should be no duplicate entries in the data set. If duplicate entries are detected, it is possible to automatically correct by removing them.

```r
# Duplicates
wrong_dup <- okdata
# Add a duplicate row
wrong_dup <- rbind(wrong_dup, wrong_dup[1,])
test <- check_format(wrong_dup)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "ATT: duplicate row(s) in final dataset: 1"
```

```
## Warning in check_format(wrong_dup): Duplicate rows detected
```

```
## [1] "==========================="
## [1] "Attention: There are problems with this data set."
## [1] "==========================="
```

```r
test <- check_format(wrong_dup, correct=TRUE)
```

```
## [1] "==========================="
## [1] " STECF Transversal2 checks on formats"
## [1] "==========================="
## [1] "==============================="
## [1] "WARNING WARNING WARNING WARNING"
## [1] "correct=TRUE selected => changes other than simple formatting will be made to the dataset"
## [1] "The changes may not be what you want and implemented only for testing purposes"
## [1] "The corrected data is returned as the output"
## [1] "Correct input data should run with correct=FALSE"
## [1] "==============================="
## [1] "Checking column names..."
## [1] "Checking eunr_id..."
## [1] "Checking loa..."
## [1] "Checking gt..."
## [1] "Checking kw..."
## [1] "Checking depdate..."
## [1] "Checking retdate..."
## [1] "Checking fishdate..."
## [1] "Checking deptime..."
## [1] "Checking rettime..."
## [1] "Checking gear..."
## [1] "Checking gear_mesh_size..."
## [1] "Checking fishing_area..."
## [1] "Checking economic_zone..."
## [1] "Checking rectangle..."
## [1] "Checking trip_id"
## [1] "Checking uniqueness of trip_id..."
## [1] "Checking duplicates..."
## [1] "ATT: 1 duplicated entries removed"
## [1] "==========================="
## [1] "The returned data passes the check."
## [1] "==========================="
```