

Package ‘RstoxData’

January 13, 2021

Version 1.0.23

Date 2021-01-13

Title Tools to Read and Manipulate Fisheries Data

Depends R (>= 3.6)

Description Set of tools to read and manipulate various data formats for fisheries. Mainly catered towards scientific trawl survey sampling ('biotic') data, acoustic trawl data, and commercial fishing catch ('landings') data. Among the supported data formats are the data products from the Norwegian Institute Marine Research ('IMR') and the International Council for the Exploration of the Sea (ICES).

URL <https://github.com/StoXProject/RstoxData>

BugReports <https://github.com/StoXProject/RstoxData/issues>

License LGPL-3

LazyData true

Encoding UTF-8

Imports data.table (>= 1.12.6), Rcpp (>= 1.0.0), xml2 (>= 1.2.2)

Suggests readr (>= 1.3.1), testthat

LinkingTo Rcpp

RoxygenNote 7.1.1

NeedsCompilation yes

Author Ibrahim Umar [cre, aut],
Sindre Vatnehol [aut],
Arne Johannes Holmin [aut],
Edvin Fuglebakk [aut],
Espen Johnsen [aut],
Norwegian Institute of Marine Research [cph, fnd]

Maintainer Ibrahim Umar <ibrahim.umar@hi.no>

Repository CRAN

Date/Publication 2021-01-13 18:40:13 UTC

R topics documented:

AcousticData	3
AddToStoxBiotic	4
backwardCompatibility	4
BioticData	5
ConvertStoxBiotic	5
DataTypes	6
DefineStoxBioticTranslation	6
FilterAcoustic	7
FilterBiotic	8
filterData	8
FilterStoxAcoustic	9
FilterStoxBiotic	10
general_arguments	10
getRstoxDataDefinitions	11
getStoxKeys	11
ICESAcoustic	12
ICESAcousticData	12
ICESBiotic	13
ICESBioticData	14
ICESDatras	14
ICESDatrasData	15
is.LandingData	15
is.StoxLandingData	16
LandingData	16
lapplyOnCores	17
mapplyOnCores	17
mergeByIntersect	18
mergeByStoxKeys	18
mergeDataTables	19
MergeStoxAcoustic	19
MergeStoxAcousticData	20
MergeStoxBiotic	20
MergeStoxBioticData	21
ModelData	21
parseInterCatch	22
ProcessData	23
processPropertyFormats	23
ReadAcoustic	23
ReadBiotic	24
readErsFile	25
readLssFile	25
readXmlFile	26
RedefineStoxBiotic	27
ReportICESAcoustic	28
ReportICESAcousticData	28
ReportICESBiotic	29

<i>AcousticData</i>	3
ReportICESBioticData	29
ReportICESDatras	30
ReportICESDatrasData	30
RstoxData	31
setRstoxPrecisionLevel	32
StoxAcoustic	32
StoxAcousticData	33
StoxBiotic	33
StoxBioticData	34
stoxBioticObject	34
StoxBioticTranslation	34
stoxFunctionAttributes	35
StoxLanding	35
StoxLandingData	36
TranslateStoxBiotic	37
xsdObjects	38
Index	39

AcousticData	<i>StoX data type AcousticData</i>
--------------	------------------------------------

Description

Biotic data read from biotic xml files.

Details

This StoX data type is produced by [ReadAcoustic](#), and contains one list per input acoustic file holding the tables read from each file, added a table named "metadata" holding the input file path and format. Currently supported are NMDEchosounder1 (<https://www.imr.no/formats/nmdechosounder/v1/>), and ICESAcoustic (<https://ices.dk/data/data-portals/Pages/acoustic.aspx>, click on "Acoustic data format" to download the format description).

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

AddToStoxBiotic *Add variables to StoxBioticData from BioticData*

Description

Add variables to StoxBioticData from BioticData

Usage

```
AddToStoxBiotic(StoxBioticData, BioticData, VariableNames = character())
```

Arguments

StoxBioticData [StoxBioticData](#).

BioticData [BioticData](#).

VariableNames A character vector with names of the variables to add from the BioticData.

Value

An object of StoX data type [StoxBioticData](#).

backwardCompatibility *Backward compabitibility actions:*

Description

Backward compabitibility actions:

Usage

```
backwardCompatibility
```

Format

An object of class list of length 2.

BioticData	<i>StoX data type BioticData</i>
------------	----------------------------------

Description

Biotic data read from biotic xml files.

Details

This StoX data type is produced by [ReadBiotic](#), and contains one list per input biotic file holding the tables read from each file, added a table named "metadata" holding the input file path and format. Currently supported are NMDBiotic1.4 (<https://www.imr.no/formats/nmdbiotic/v1.4/>), NMDBiotic3.0 (<https://www.imr.no/formats/nmdbiotic/v3/>), and ICESBiotic (<https://ices.dk/data/data-portals/Pages/acoustic.aspx>, click on "Acoustic data format" to download the format description).

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

ConvertStoxBiotic	<i>Convert StoxBioticData</i>
-------------------	-------------------------------

Description

This function converts one or more columns of [StoxBioticData](#) by the function given by `ConversionFunction`.

Usage

```
ConvertStoxBiotic(
  StoxBioticData,
  ConversionFunction = c("Constant", "Addition", "Scaling", "AdditionAndScaling"),
  GruopingVariables = character(),
  Conversion = data.table::data.table()
)
```

Arguments

`StoxBioticData` An input of [ModelData](#) object

`ConversionFunction`

Character: The function to convert by, one of "Constant", for replacing the specified columns by a constant value; "Addition", for adding to the columns; "Scaling", for multiplying by a factor; and "AdditionAndScaling", for both adding and multiplying.

GroupingVariables

A vector of variables to specify in the Conversion. The parameters specified in the table are valid for the combination of the GroupingVariables in the data.

Conversion

A table of the GroupingVariables and the columns "TargetVariable", "SourceVariable" and the parameters of the ConversionFunction (see details).

The parameters of the ConversionFunction are "Constant" for ConversionFunction "Constant", "Addition" for ConversionFunction "Addition", "Scaling" for ConversionFunction "Scaling", and "Addition" and "Scaling" for ConversionFunction "AdditionAndScaling".

Value

A [StoxBioticData](#) object.

DataTypes

StoX data types of the RstoxData package

Description

StoX data types are the data types used to transfer data and information between processes in a StoX estimation model. The data types are divided into two types, the [ModelData](#) and [ProcessData](#).

DefineStoxBioticTranslation

Define StoxBioticData variables translation

Description

This function defines the translation table used as input to [TranslateStoxBiotic](#) to translate values of one or more columns of [StoxBioticData](#) to new values given by a table or read from a CSV file.

Usage

```
DefineStoxBioticTranslation(
  processData,
  UseProcessData = FALSE,
  DefinitionMethod = c("Table", "ResourceFile"),
  Translation = data.table::data.table(),
  FileName
)
```

Arguments

processData	The current data produced by a previous instance of the function.
UseProcessData	Logical: If TRUE use the existing function output in the process.
DefinitionMethod	Character: A string naming the method to use, one of "Table" for defining the Translation, and "ResourceFile" for reading the table from the file given by FileName.
Translation	A table of the columns "VariableName", representing the variable to translate; "Value", giving the values to translate; and "NewValue", giving the values to translate to.
FileName	The csv file holding a table with the three variables listed for Translation.

Value

A [StoxBioticTranslation](#) object.

FilterAcoustic	<i>Filter (raw) Acoustic data</i>
----------------	-----------------------------------

Description

Filters [AcousticData](#).

Usage

```
FilterAcoustic(AcousticData, FilterExpression, FilterUpwards = FALSE)
```

Arguments

AcousticData	Input AcousticData data.
FilterExpression	Filter expression in list of strings. The name of the list and structures should be identical to the names of the input data list.
FilterUpwards	Whether the filter action will propagate in the upwards direction. Default to FALSE.

Value

An object of filtered data in the same format as the input data.

FilterBiotic	<i>Filter (raw) Biotic data</i>
--------------	---------------------------------

Description

Filters [BioticData](#).

Usage

```
FilterBiotic(BioticData, FilterExpression, FilterUpwards = FALSE)
```

Arguments

BioticData Input [BioticData](#) data.

FilterExpression

Filter expression given as a list of strings. The name of the list and structures should be identical to the names of the input data list. To extract or exclude missing values (NAs) use the operator `%in%` or the special operator `%notin%`, which is defined in [RstoxData](#).

FilterUpwards Whether the filter action will propagate in the upwards direction. Default to FALSE.

Value

An object of filtered data in the same format as the input data.

filterData	<i>Run filter on any StoX related data source</i>
------------	---

Description

Run filter on any StoX related data source

Usage

```
filterData(
  inputData,
  filterExpression,
  propagateDownwards = TRUE,
  propagateUpwards = FALSE
)
```


Arguments

inputData	An input data. Can be a list of biotic data (StoX data type BioticData), list of acoustic data, StoxBiotic data, or StoxAcoustic data.
filterExpression	Filter expression in list of strings. The name of the list and structures should be identical to the names of the input data list.
propagateDownwards	Whether the filter action will propagate in the downwards direction. Default to TRUE.
propagateUpwards	Whether the filter action will propagate in the upwards direction. Default to FALSE.

Value

An object of filtered data in the same format as the input data.

FilterStoxAcoustic *Filter StoxAcoustic data*

Description

Filters [StoxAcousticData](#).

Usage

```
FilterStoxAcoustic(StoxAcousticData, FilterExpression, FilterUpwards = FALSE)
```

Arguments

StoxAcousticData	Input StoxAcousticData data.
FilterExpression	Filter expression in list of strings. The name of the list and structures should be identical to the names of the input data list.
FilterUpwards	Whether the filter action will propagate in the upwards direction. Default to FALSE.

Value

An object of filtered data in the same format as the input data.

FilterStoxBiotic *Filter StoxBiotic data*

Description

Filters [StoxBioticData](#).

Usage

```
FilterStoxBiotic(StoxBioticData, FilterExpression, FilterUpwards = FALSE)
```

Arguments

StoxBioticData Input [StoxBioticData](#) data.

FilterExpression

Filter expression in list of strings. The name of the list and structures should be identical to the names of the input data list.

FilterUpwards Whether the filter action will propagate in the upwards direction. Default to FALSE.

Value

An object of filtered data in the same format as the input data.

general_arguments *General parameters of RstoxData.*

Description

All functions referring to a project, a model, a process or an output table use the same parameters, listed here.

Arguments

processData The current data produced by a previous instance of the function.

UseProcessData Logical: If TRUE use the existing function output in the process.

NumberOfCores The number of cores to use (defaulted to 1), truncated to the number of available cores.

```
getRstoxDataDefinitions
```

Get RstoxData definitions

Description

This function gets vital definitions from the RstoxData environment.

Usage

```
getRstoxDataDefinitions(name = NULL, ...)
```

Arguments

name	An optional string vector denoting which definitions to extract.
...	values overriding the values of definitions.

Value

A list of definitions.

Examples

```
getRstoxDataDefinitions()
```

```
getStoxKeys
```

Get the keys of a StoX format

Description

Get the keys of a StoX format

Usage

```
getStoxKeys(  
  StoxDataType = c("StoxBiotic", "StoxAcoustic"),  
  level = NULL,  
  keys.out = c("all", "only.present", "all.but.present")  
)
```

Arguments

StoxDataType	The name of the StoX format (only StoxBiotic implemented yet).
level	The name of the level/table to get keys for.
keys.out	Specification of what to return. One of "all", to return all keys of the level; "only.present", to return only the key of the level; and "all.but.present", to return all keys except the present key.

ICESAcoustic	<i>Converts AcousticData to ICESAcousticData</i>
--------------	--

Description

Note that this function only supports AcousticData object that is created from reading an ICES acoustic XML file.

Usage

```
ICESAcoustic(AcousticData)
```

Arguments

AcousticData	A AcousticData object from an ICES acoustic XML format file.
--------------	--

Value

List of data.table objects in the ICES acoustic CSV format.

ICESAcousticData	<i>StoX data type ICESAcousticData</i>
------------------	--

Description

Acoustic data stored in the ICESAcoustic (CSV) format.

Details

This StoX data type is produced by [ICESAcoustic](#), and contains one list per input biotic file read to produce the input to [ICESAcoustic](#), each holding the tables Instrument, Calibration, DataAcquisition, DataProcessing, Cruise and Data (here Data is a table merged from Log, Sample and Data of the ICESAcoustic xml format). Each file read to produce the input to [ICESAcoustic](#)

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

ICESBiotic	<i>Write ICES biotic CSV format file</i>
------------	--

Description

Given an BioticData object, this function will write an ICES biotic CSV file. Note that this function only supports BioticData object that is created from reading an NMD biotic version 3 XML file.

Usage

```
ICESBiotic(  
  BioticData,  
  SurveyName = "NONE",  
  Country = character(),  
  Organisation = integer(),  
  AllowRemoveSpecies = TRUE  
)
```

Arguments

- | | |
|--------------------|--|
| BioticData | a BioticData object from an XML file with NMD biotic version 3 format. |
| SurveyName | A string naming the survey. Must be one of the names listed on https://vocab.ices.dk/?ref=1453 or NONE (the default). |
| Country | The ISO_3166 code of the country running the cruise. See http://vocab.ices.dk/?ref=337 . |
| Organisation | An integer code representing the organization running the cruise. See https://vocab.ices.dk/?ref=1398 for a list of possible codes (e.g., Institute of Marine Research: 612). |
| AllowRemoveSpecies | ICES submission will not allow the resulting CSV file to be uploaded if the file contains species not listed in https://acoustic.ices.dk/Services/Schema/XML/SpecWoRMS.xml . Setting this parameter to TRUE will remove the unlisted species records. |

Value

List of data.table objects in the ICES acoustic CSV format.

ICESBioticData	<i>StoX data type ICESBioticData</i>
----------------	--------------------------------------

Description

Biotic data stored in the ICESBiotic (CSV) format.

Details

This StoX data type is produced by [ICESBiotic](#), and contains one list per input biotic file read to produce the input to [ICESBiotic](#), each holding the tables Cruise, Haul, Catch and Biology, in that hierarchical order.

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

ICESDatras	<i>Write ICES DATRAS (NS-IBTS) format file</i>
------------	--

Description

Given an BioticData object, this function will write an ICES DATRAS (NS-IBTS) file. Note that this function only supports BioticData object that is created from reading an NMD biotic version 3 XML file.

Usage

```
ICESDatras(BioticData)
```

Arguments

BioticData a BioticData object from an XML file with NMD biotic version 3 format.

Value

List of data.table objects in the ICES DATRAS CSV format.

ICESDatrasData	<i>StoX data type ICESDatrasData</i>
----------------	--------------------------------------

Description

Biotic data stored in the ICESDatras (CSV) format.

Details

This StoX data type is produced by [ICESDatras](#), and contains one list per input biotic file read to produce the input to [ICESDatras](#), each holding the tables HH, HL and CA, in that hierarchical order.

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

is.LandingData	<i>Check if argument is LandingData</i>
----------------	---

Description

Checks if argument conforms to specification for [LandingData](#)

Usage

```
is.LandingData(LandingData)
```

Arguments

LandingData argument to be checked for data conformity

Value

logical, TRUE if argument conformed to specification for [LandingData](#)

`is.StoxLandingData` *Check if argument is StoxLandingData*

Description

Checks if argument conforms to specification for [StoxLandingData](#)

Usage

```
is.StoxLandingData(StoxLandingData)
```

Arguments

`StoxLandingData`
argument to be checked for data conformity

Value

logical, TRUE if argument conformed to specification for [StoxLandingData](#)

`LandingData` *LandingData*

Description

`LandingData`

Data

One entry 'Seddellinje' is one line of a sales-note or landing-note. These are issued as fish is landed, and a complete set of these for a period can be considered a census of all first hand sale of fish sold from Norwegian vessels.

Format

`list()` of [data.table](#) representing the different complexTypes in namespace <http://www.imr.no/formats/landinger/v2>
For ease of merging: all top level attributes are repeated for all tables. And all line-identifying variables are included as top-level attributes.

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

lapplyOnCores	<i>Run a function on all elements of x on one or more cores</i>
---------------	---

Description

Run a function on all elements of x on one or more cores

Usage

```
lapplyOnCores(x, FUN, NumberOfCores = 1L, ...)
```

Arguments

x	An object to apply FUN to.
FUN	The function to apply.
NumberOfCores	The number of cores to use (defaulted to 1), truncated to the number of available cores.
...	Additional arguments to FUN.

Value

A list of outputs from FUN.

mapplyOnCores	<i>Run a function on all elements of x on one or more cores</i>
---------------	---

Description

Run a function on all elements of x on one or more cores

Usage

```
mapplyOnCores(
  FUN,
  NumberOfCores = integer(),
  ...,
  MoreArgs = NULL,
  SIMPLIFY = FALSE
)
```

Arguments

FUN	The function to apply.
NumberOfCores	The number of cores to use (defaulted to 1), truncated to the number of available cores.
..., MoreArgs, SIMPLIFY	See mapply .

Value

A list of outputs from FUN.

mergeByIntersect	<i>Merge two data tables by the intersect of the names</i>
------------------	--

Description

Merge two data tables by the intersect of the names

Usage

```
mergeByIntersect(x, y, ..., msg = FALSE)
```

Arguments

x, y	Data tables of class data.table .
...	Various overrides.
msg	Verbose message switch, default to FALSE.

Value

A merged data table.

mergeByStoxKeys	<i>Merge two data tables by StoX keys</i>
-----------------	---

Description

Merge two data tables by StoX keys

Usage

```
mergeByStoxKeys(x, y, StoxDataType, toMergeFromY = NULL, replace = FALSE, ...)
```

Arguments

x, y	Data tables of class data.table .
StoxDataType	Input data type. Text string of <code>StoxBiotic</code> or <code>StoxAcoustic</code> .
toMergeFromY	Specify key columns from y. NULL means all similarly named columns from x and y will be merged. Default to NULL.
replace	Whether to replace the variables in the target. Default to FALSE.
...	Extra parameters that will be passed into merge .

Value

A merged data table.

mergeDataTables	<i>Merge list of data tables recursively</i>
-----------------	--

Description

Merge list of data tables recursively

Usage

```
mergeDataTables(data, tableNames = NULL, output.only.last = FALSE, ...)
```

Arguments

data	A list of data tables.
tableNames	A character vector holding the names of the tables to merge.
output.only.last	Only returns last merged table.
...	Extra parameters that will be passed into merge .

Value

A merged data table.

MergeStoxAcoustic	<i>Merge StoxAcousticData</i>
-------------------	-------------------------------

Description

Merge StoxAcousticData

Usage

```
MergeStoxAcoustic(StoxAcousticData, TargetTable = "NASC")
```

Arguments

StoxAcousticData	A list of StoX acoustic data (StoX data type StoxAcousticData).
TargetTable	The name of the table up until which to merge (the default "NASC" implies merging all tables)

Value

An object of StoX data type [MergeStoxAcousticData](#).

MergeStoxAcousticData *StoX data type MergeStoxAcousticData*

Description

Merged [StoxAcousticData](#).

Details

This StoX data type is produced by [MergeStoxAcoustic](#), and contains one merged table of [StoxAcousticData](#).

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

MergeStoxBiotic *Merge StoxBioticData*

Description

Merge StoxBioticData

Usage

```
MergeStoxBiotic(StoxBioticData, TargetTable = "Individual")
```

Arguments

StoxBioticData A list of StoX biotic data (StoX data type [StoxBioticData](#)).

TargetTable The name of the table up until which to merge (the default "Individual" implies merging all tables)

Value

An object of StoX data type [MergeStoxBioticData](#).

MergeStoxBioticData *StoX data type MergeStoxBioticData*

Description

Merged [StoxBioticData](#).

Details

This StoX data type is produced by [MergeStoxBiotic](#), and contains one merged table of [StoxBioticData](#).

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

ModelData *StoX data types of the RstoxData package*

Description

StoX data types are the data types used to transfer data and information between processes in a StoX estimation model.

Arguments

BioticData [BioticData](#).

StoxBioticData [StoxBioticData](#).

AcousticData [AcousticData](#).

StoxAcousticData
 [StoxAcousticData](#).

Details

This RstoxData package produces the following StoX data types:

- [BioticData](#)
- [StoxBioticData](#)
- [MergeStoxBioticData](#)
- [AcousticData](#)
- [StoxAcousticData](#)
- [MergeStoxAcousticData](#)
- [LandingData](#)
- [StoxLandingData](#)

- [ICESAcousticData](#)
- [ICESBioticData](#)
- [ICESDatrasData](#)
- [ReportICESAcousticData](#)
- [ReportICESBioticData](#)
- [ReportICESDatrasData](#)

See Also

[RstoxBase](#) and [RstoxFDA](#) for a list of all StoX data types produced by the other official StoX function packages.

parseInterCatch	<i>Parses InterCatch</i>
-----------------	--------------------------

Description

Parses the InterCatch exchange format v 1.0 for Commercial Catch and Sample Data.

Usage

```
parseInterCatch(file, encoding = "UTF-8")
```

Arguments

file	path to file containing intercatch formatted data
encoding	encoding of 'file'

Details

The InterCatch exchange format is a jagged comma-separated format, where the number of fields on a line is determined by a record-type identifier in position 1. Three record types are defined, "HI" (header information), "SI" (species information), and "SD" (species data). The format is specified on <https://ices.dk/data/Documents/Intercatch/IC-ExchangeFormat1-0.pdf>.

Value

named list with three members:

HI [data.table](#) with HI records

SI [data.table](#) with SI records

SD [data.table](#) with SD records

ProcessData	<i>Process data used in estimation models in StoX</i>
-------------	---

Description

The process data of the RstoxData package.

Details

- [StoxBioticTranslation](#)

See Also

[ModelData](#) for model data types and [DataTypes](#) for all data types produced by [RstoxData](#).

processPropertyFormats	<i>Define the process property formats:</i>
------------------------	---

Description

Define the process property formats:

Usage

```
processPropertyFormats
```

Format

An object of class list of length 8.

ReadAcoustic	<i>Read acoustic XML files</i>
--------------	--------------------------------

Description

This function reads multiple acoustic file to a list with a list of tables for each file.

Usage

```
ReadAcoustic(FileNames)
```

Arguments

FileNames The paths of the acoustic files.

Details

This function is awesome and does excellent stuff.

Value

An object of StoX data type AcousticData: A list of a list of data.tables of the different levels of the input acoustic files.

See Also

[readXmlFile](#).

Examples

```
exampleFile <- system.file(
  "testresources", "libas_ListUserFile20__L40.0-2259.9_small.xml", package="RstoxData")
bioticData <- ReadBiotic(exampleFile)
```

ReadBiotic

Read biotic XML files

Description

This function reads multiple biotic file to a list with a list of tables for each file.

Usage

```
ReadBiotic(FileNames)
```

Arguments

FileNames The paths of the biotic files.

Details

This function is awesome and does excellent stuff.

Value

An object of StoX data type BioticData: A list of a list of data.tables of the different levels of the input biotic files.

See Also

[readXmlFile](#).

Examples

```
exampleFile <- system.file("testresources","biotic3.1_example.xml", package="RstoxData")
bioticData <- ReadBiotic(exampleFile)
```

readErsFile *Parses logbooks (ERS)*

Description

Parses electronic logbooks (ERS) from tabular format delivered by Directorate of Fisheries (FDIR)

Usage

```
readErsFile(file, encoding = "latin1")
```

Arguments

file	path to file
encoding	encoding for 'file'

Details

The format is a pipe-separated format encoding aggregated ERS records (logbooks). It is provided to IMR on a regular basis from FDIR. Column headers are in Norwegian.

Value

data.table() with logbooks

readLssFile *Parses landings (sales notes)*

Description

Parses sales notes data from the Norwegian Directorate of Fisheries (FDIR) on the LSS format

Usage

```
readLssFile(file, encoding = "latin1", guessMax = 1e+05, strict = T)
```

Arguments

file	path to file with LSS landings
encoding	encoding for 'file'
guessMax	passed to read_delim , unless 'strict' is true
strict	enforce strict adherence to data format.

Details

The LSS format is a pipe-separated format encoding landings (sales-notes). It is provided to IMR on a regular basis from FDIR. Column headers are in Norwegian.

Historically, columns in the landings provided from FDIR has been adapted for each data delivery. Lately data deliveries has become standardized, but in order to support variants adherence to the standardization is not enforced by this function, unless option 'strict' is selected. If column names does not match specification, but data is otherwise parse-able, a warning will be issued.

If the parameter 'strict' is not TRUE, data types may be inferred from data. The parameter 'guess-Max' limits how many lines are inspected for data type inference (passed to [read_delim](#))

Value

data.table with LSS landings

readXmlFile

Read fisheries XML data format file

Description

Read fisheries XML data format file. Currently supports IMR Biotic version 1 until 3, IMR Echosounder version 1, and IMR Landing version 2 formats at the moment. Streaming XML pull parser can be used to avoid loading the whole XML into memory and it supports ZIP file reading. Please note that the XML file inside the zip file should be using the same name as the zip file itself (e.g. test.xml inside test.zip).

Usage

```
readXmlFile(xmlFilePath, stream = TRUE, useXsd = NULL, verbose = FALSE)
```

Arguments

xmlFilePath	full path to the XML file to be read.
stream	a streaming XML pull parser is used if this is set to TRUE. An XML DOM parser is used if this is set to FALSE. Default to TRUE.
useXsd	Specify an xsd object to use. Default to NULL.
verbose	Show verbose output. Default to FALSE.

Value

List of data.table objects containing the "flattened" XML data.

Examples

```
## Not run:  
# Reading test.xml using XML pull parser  
one <- readXmlFile("./test.xml")  
# Reading test.xml using XML DOM parser  
two <- readXmlFile("./test.xml", stream = FALSE)  
# Reading test.xml inside test.zip file  
three <- readXmlFile("./test.zip")  
  
## End(Not run)
```

RedefineStoxBiotic *Redefine StoxBioticData variables by data from BioticData*

Description

This function redefines one or more columns of [StoxBioticData](#) by columns of [BioticData](#).

Usage

```
RedefineStoxBiotic(  
  StoxBioticData,  
  BioticData,  
  Redefinition = data.table::data.table()  
)
```

Arguments

StoxBioticData An input of [ModelData](#) object

BioticData An input of [ModelData](#) object

Redefinition A table of the columns "VariableName", representing the variable to redefine; and "RedefineBy", representing the variable from [BioticData](#) to replace by.

Value

A [StoxBioticData](#) object.

ReportICESAcoustic *Reports ICESAcousticData to a csv file for each input acoustic file used to create the ICESAcousticData*

Description

Reports ICESAcousticData to a csv file for each input acoustic file used to create the ICESAcousticData

Usage

```
ReportICESAcoustic(ICESAcousticData)
```

Arguments

ICESAcousticData
A ICESAcousticData object obtained from an ICES acoustic XML format file.

Value

List of string matrices in the ICES acoustic CSV format.

ReportICESAcousticData
Rbind ICESAcousticData to a string matrix.

Description

The output of this function is suited for submission to <https://acoustic.ices.dk/>.

Details

The ICESAcoustic CSV format is one string matrix containing all tables of ICESAcousticData, where column names are included as header rows.

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

ReportICESBiotic	<i>Reports ICESBioticData to a csv file for each input acoustic file used to create the ICESBioticData</i>
------------------	--

Description

Reports [ICESBioticData](#) to a csv file for each input acoustic file used to create the [ICESBioticData](#)

Usage

```
ReportICESBiotic(ICESBioticData)
```

Arguments

`ICESBioticData` A [ICESBioticData](#) object obtained from an ICES acoustic XML format file.

Value

List of string matrices in the ICES acoustic CSV format.

ReportICESBioticData	<i>Rbind ICESBioticData to a string matrix.</i>
----------------------	---

Description

The output of this function is suited for submission to <https://acoustic.ices.dk/>.

Details

The ICESBiotic CSV format is one string matrix containing all tables of [ICESBioticData](#), where column names are included as header rows.

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

ReportICESDatras	<i>Reports ICESDatrasData to a csv file for each input acoustic file used to create the ICESDatras</i>
------------------	--

Description

Reports [ICESDatrasData](#) to a csv file for each input acoustic file used to create the [ICESDatras](#)

Usage

```
ReportICESDatras(ICESDatrasData)
```

Arguments

ICESDatrasData A [ICESDatrasData](#) object returned from [ICESDatras](#).

Value

List of string matrices in the ICES Datras CSV format.

ReportICESDatrasData	<i>Rbind ICESDatrasData to a string matrix.</i>
----------------------	---

Description

The output of this function is suited for submission to <https://www.ices.dk/data/data-portals/Pages/DATRAS.aspx>.

Details

The ICESDatras CSV format is one string matrix containing all tables of [ICESDatrasData](#), where column names are included as header rows.

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

Description

Set of tools to read and manipulate various data formats for fisheries. Mainly catered towards scientific trawl survey sampling ('biotic') data, acoustic trawl data, and commercial fishing catch ('landings') data. Among the supported data formats are the data products from the Norwegian Institute Marine Research ('IMR') and the International Council for the Exploration of the Sea (ICES).

Details

The RstoxData package contains functions for reading, filtering and writing biotic, acoustic and landing data as XML files. Filtering can be done by R syntax such as `longitude > 10`, or by pre defined functions such as `inside()`. On computers that return errors when trying to run the Rtools through RStudio (most institutional Windows machines), install the binary directly from <https://github.com/StoXProject/RstoxData/releases>. Download the newest RstoxData zip file, click the "Packages" tab -> "Install" -> "Install from:" "Package Archive File" -> "Install". If the installer does not complain, the package is installed correctly.

Author(s)

Maintainer: Ibrahim Umar <ibrahim.umar@hi.no>

Authors:

- Sindre Vatnehol
- Arne Johannes Holmin
- Edvin Fuglebakk
- Espen Johnsen

Other contributors:

- Norwegian Institute of Marine Research [copyright holder, funder]

See Also

Useful links:

- <https://github.com/StoXProject/RstoxData>
- Report bugs at <https://github.com/StoXProject/RstoxData/issues>

setRstoxPrecisionLevel

Round off to number of digits

Description

Round off to number of digits

Usage

setRstoxPrecisionLevel(x)

Arguments

x A list of data . tables or a single data . table object.

Value

A transformed object.

StoxAcoustic

Convert AcousticData to StoxAcousticData

Description

Convert AcousticData to StoxAcousticData

Usage

StoxAcoustic(AcousticData)

Arguments

AcousticData [AcousticData](#).

Value

An object of StoX data type [StoxAcousticData](#).

StoxAcousticData	<i>StoX data type StoxAcousticData</i>
------------------	--

Description

Acoustic data stored in the StoxAcoustic format, which contains the variables needed for most estimation models used by StoX.

Details

This StoX data type is produced by [StoxAcoustic](#), and contains the tables Cruise, Log, Beam, AcousticCategory, ChannelReference and NASC in that hierarchical order.

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

StoxBiotic	<i>Convert BioticData to StoxBioticData</i>
------------	---

Description

Convert BioticData to StoxBioticData

Usage

```
StoxBiotic(BioticData)
```

Arguments

BioticData [BioticData](#).

Value

An object of StoX data type [StoxBioticData](#).

StoxBioticData	<i>StoX data type StoxBioticData</i>
----------------	--------------------------------------

Description

Biotic data stored in the StoxBiotic format, which contains the variables needed for most estimation models used by StoX.

Details

This StoX data type is produced by [StoxBiotic](#), and contains the tables Cruise, Station, Haul, SpeciesCategory, Sample and Individual in that hierarchical order.

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

stoxBioticObject	<i>stoxBioticObject</i>
------------------	-------------------------

Description

Pre-processed objects for raw XML data to StoXBiotic format

Usage

```
stoxBioticObject
```

Format

An object of class list of length 9.

StoxBioticTranslation	<i>Translation definition (from file) for StoxBioticData.</i>
-----------------------	---

Description

Translation definition (from file) for [StoxBioticData](#).

Details

This StoX data type is produced by [DefineStoxBioticTranslation](#), and contains the columns VariableName, Value and NewValue.

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

 stoxFunctionAttributes

Function specification for inclusion in StoX projects

Description

Function specification for inclusion in StoX projects

Usage

stoxFunctionAttributes

Format

An object of class list of length 22.

 StoxLanding

Convert landing data

Description

StoX function Convert landing data to the aggregated format [StoxLandingData](#)

Usage

```
StoxLanding(
  LandingData,
  appendColumns = character(),
  appendColumnsNames = appendColumns
)
```

Arguments

LandingData Sales-notes data. See [LandingData](#)

appendColumns character() vector that identifies additional columns in [LandingData](#) to append to [StoxLandingData](#).

appendColumnsNames
 character() vector that defines the names of the columns in 'appendColumns' in the output.

Details

All columns that are not the ones aggregated (weight), will be used as aggregation variables. This includes any columns added with 'appendColumns' and may not make much sense for continuous variables.

If 'LandingData' does not contain columns identified by 'appendColumns'. NA columns will be added.

Correspondences indicate which field a value is derived from, not necessarily verbatim copied.

Correspondence to LandingData (<http://www.imr.no/formats/landinger/v2>):

Species Art_kode

Year Fangstår

CatchDate SisteFangstdato

Gear Redskap_kode

Area Hovedområde_kode

SubArea Lokasjon_kode

Coastal KystHav_kode

N62Code NordSørFor62GraderNord

VesselLengthGroup Lengdegruppe_kode

CountryVessel Fartøynasjonalitet_kode

LandingSite Mottaksstasjon

CountryLanding Landingsnasjon_kode

Usage HovedgruppeAnvendelse_kode

RoundWeightKilogram Rundvekt

Value

[StoxLandingData](#), aggregated landings data.

StoxLandingData

StoxLandingData

Description

Table ([data.table](#)) with aggregated weight of landings from landing records.

Column definitions

- Species** character() code for species category (species identified by market or regulation standards. Several codes may code the same species or stock, and some catch may be recorded only at higher taxonomic classifications)
- Year** integer() Year of catch
- CatchDate** POSIXct() Date of catch (last catch on trip) in UTC
- Gear** character() Code for gear used for catch (dominant gear for trip)
- Area** character() Area code for the position where the catch was caught (dominant area for trip)
- SubArea** character() Subdivision of area code for the position where the catch was caught (dominant area for trip)
- Coastal** character() Code indicating whether catch was taken within coastal delimitation line (dominant side for trip)
- N62Code** character() Code indicating whether catch was taken north or south of 62 deg. Lat. (dominant side for trip)
- VesselLengthGroup** character() Length group for vessel
- CountryVessel** character() Country of the vessel that caught the catch
- LandingSite** character() Code identifying landing site (buyer of catch)
- CountryLanding** character() Country where catch was landed
- Usage** character() Code for market usage of catch.
- RoundWeightKilogram** numeric() Weight of round catch in kg.

See Also

[DataTypes](#) for a list of all StoX data types produced by [RstoxData](#)

TranslateStoxBiotic *Translate StoxBioticData*

Description

This function translates one or more columns of [StoxBioticData](#) to new values given by the table Translation or by the input StoxBioticTranslation.

Usage

```
TranslateStoxBiotic(
  StoxBioticData,
  TranslationDefinition = c("FunctionParameter", "FunctionInput"),
  Translation = data.table::data.table(),
  StoxBioticTranslation
)
```

Arguments

StoxBioticData	An input of ModelData object
TranslationDefinition	Character: A string naming the method to use for the translation, one of "FunctionParameter" for defining the Translation, and "FunctionInput" for using the table produced by the process given by the function input StoxBioticTranslation.
Translation	A table of the columns "VariableName", representing the variable to translate; "Value", giving the values to translate; and "NewValue", giving the values to translate to.
StoxBioticTranslation	The process from which to get the StoxBioticTranslation definition.

Value

A [StoxBioticData](#) object.

xsd0bjects

xsdObjects

Description

Pre-processed XSD file objects

Usage

xsd0bjects

Format

A list with 4 elements

landingerv2.xsd List Landing Format v2

nmdbioticv1.xsd List NMD Biotic Format v1

nmdbioticv1.1.xsd List NMD Biotic Format v1.1

nmdbioticv1.2.xsd List NMD Biotic Format v1.2

nmdbioticv1.3.xsd List NMD Biotic Format v1.3

nmdbioticv1.4.xsd List NMD Biotic Format v1.4

nmdbioticv3.xsd List NMD Biotic Format v3

nmdbioticv3.1.xsd List NMD Biotic Format v3.1

nmdechounder1.xsd List NMD Echounder Format v1

Source

<https://www.imr.no/formats>

Index

* datasets

- backwardCompatibility, 4
 - processPropertyFormats, 23
 - stoxBioticObject, 34
 - stoxFunctionAttributes, 35
 - xsdObjects, 38
- AcousticData, 3, 7, 21, 32
- AddToStoxBiotic, 4
- backwardCompatibility, 4
- BioticData, 4, 5, 8, 9, 21, 27, 33
- ConvertStoxBiotic, 5
- data.table, 16, 18, 22, 36
- DataTypes, 3, 5, 6, 12, 14–16, 20, 21, 23, 28–30, 33, 34, 37
- DefineStoxBioticTranslation, 6, 34
- FilterAcoustic, 7
- FilterBiotic, 8
- filterData, 8
- FilterStoxAcoustic, 9
- FilterStoxBiotic, 10
- general_arguments, 10
- getRstoxDataDefinitions, 11
- getStoxKeys, 11
- ICESAcoustic, 12, 12
- ICESAcousticData, 12, 22, 28
- ICESBiotic, 13, 14
- ICESBioticData, 14, 22, 29
- ICESDatras, 14, 15, 30
- ICESDatrasData, 15, 22, 30
- is.LandingData, 15
- is.StoxLandingData, 16
- LandingData, 15, 16, 21, 35
- lapplyOnCores, 17
- mapply, 17
- mapplyOnCores, 17
- merge, 18, 19
- mergeByIntersect, 18
- mergeByStoxKeys, 18
- mergeDataTables, 19
- MergeStoxAcoustic, 19, 20
- MergeStoxAcousticData, 19, 20, 21
- MergeStoxBiotic, 20, 21
- MergeStoxBioticData, 20, 21, 21
- ModelData, 5, 6, 21, 23, 27, 38
- parseInterCatch, 22
- ProcessData, 6, 23
- processPropertyFormats, 23
- read_delim, 25, 26
- ReadAcoustic, 3, 23
- ReadBiotic, 5, 24
- readErsFile, 25
- readLssFile, 25
- readXmlFile, 24, 26
- RedefineStoxBiotic, 27
- ReportICESAcoustic, 28
- ReportICESAcousticData, 22, 28
- ReportICESBiotic, 29
- ReportICESBioticData, 22, 29
- ReportICESDatras, 30
- ReportICESDatrasData, 22, 30
- RstoxData, 3, 5, 12, 14–16, 20, 21, 23, 28–30, 31, 33, 34, 37
- RstoxData-package (RstoxData), 31
- setRstoxPrecisionLevel, 32
- StoxAcoustic, 32, 33
- StoxAcousticData, 9, 19–21, 32, 33
- StoxBiotic, 33, 34
- StoxBioticData, 4–6, 10, 20, 21, 27, 33, 34, 34, 37, 38
- stoxBioticObject, 34

StoxBioticTranslation, [7](#), [23](#), [34](#), [38](#)

stoxFunctionAttributes, [35](#)

StoxLanding, [35](#)

StoxLandingData, [16](#), [21](#), [35](#), [36](#), [36](#)

TranslateStoxBiotic, [6](#), [37](#)

xsdObjects, [38](#)