Package: RDBEScore (via r-universe)

August 17, 2024

Type Package

Title Functions for the ICES Regional Database and Estimation System (RDBES)

Version 0.3.0

Author c(person(given = ``David", family = ``Currie", role = c(``aut"), comment = c(ORCID = ``0000-0002-3523-6895")), person(given = ``Richard", family = ``Meitern", role = c(``aut"), email = ``richard.meitern@ut.ee", comment = c(ORCID = ``0000-0002-2600-3002")), person(given = ``Nuno", family = ``Prista", role = c(``aut"), email = ``nuno.prista@slu.se", comment = c(ORCID = ``0000-0002-5145-7241")), person(given = ``Nicholas", family = ``Carey", role = c(``aut"), email = ``nicholas.carey@gov.scot"), person(given = ``Petri", family = ``Sarvamaa", role = c(``aut"), email = ``Petri.Sarvamaa@luke.fi"), person(given = ``Kirsten", family = ``Birch Håkansson", role = c(``aut"), email = ``kih@aqua.dtu.dk"), person(given = ``Karolina", family = ``Molla Gazi", role = c(``aut"), email = ``karolina.mollagazi@wur.nl"), person(given = ``Julia", family = ``Wischnewski", role = c(``aut"), email = ``julia.wischnewski@thuenen.de"), person(given = ``Ana Cláudia", family = ``Fernandes", role = c(``aut"), email = ``acfernandes@ipma.pt"), person(given = ``Katarzyna", family = ``Krakówka", role = c(``aut"), email = ``kkrakowka@mir.gdynia.pl"), person(given = ``Marta", family = ``Szymańska", role = c(``aut"), email = ``msuska@mir.gdynia.pl"), person(given = ``Nicolas", family = ``Goñi", role = c(``aut"), email = ``nicolas.goni@luke.fi"), person(given = ``Annica", family = ``de Groote", role = c(``ctb"), email = ``annica.isaksson.de.groote@slu.se"), person(given = ``Jonathan", family = ``Ball", role = c(``ctb"), email = ``johnathan.ball@cefas.co.uk"), person(given = ``Jonathan", family = ``Rault", role = c(``ctb"), email = ``Jonathan.Rault@ifremer.fr"), person(given = ``Antti", family = ``Sykkö", role = c(``ctb"), email = ``antti.sykko@luke.fi"), person(given = ``Liz", family = ``Clarke", role = c(``ctb"), email

= ``Liz.Clarke@gov.scot"), person(given = ``Chun", family = ``Chen", role = c(``ctb"), email = ``chun.chen@wur.nl"), person(given = ``Hongru", family = ``Zhai", role = c(``ctb"), email = ``hongru.zhai@slu.se"), person(given = ``Eros", family = ``Quesada", role = c(``ctb"), email = ``eros.quesada@slu.se"), person(given = ``Jonathan", family = ``Stounberg", role = c(``ctb"), email = ``jostou@aqua.dtu.dk"), person(given = ``Ana", family = ``Ribeiro Santos", role = c(``ctb"), email = ``ana.ribeirosantos@cefas.co.uk"), person(given = ``Jose", family = ``Castro", role = c(``ctb"), email = ``jose.castro@ieo.es"), person(given = ``Jessica", family = ``Craig", role = c(``ctb"), email = ``Jessica.Craig@gov.scot"))

Maintainer Colin Millar <colin.millar@ices.dk>

Description The RDBEScore package provides functions to import and work with fisheries data downloaded from the ICES RDBES database. It also contains functions to perform estimation analysis using the resulting objects.

URL https://github.com/ices-tools-dev/RDBEScore, https://rdbes.ices.dk

License GPL-3 + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

Suggests knitr, rmarkdown, testthat

Depends R (>= 4.1)

Imports dplyr (>= 1.1.3), data.table

VignetteBuilder knitr

Repository https://ices-tools-prod.r-universe.dev

RemoteUrl https://github.com/ices-tools-dev/RDBEScore

RemoteRef HEAD

RemoteSha fc7683c9744c733d5282ab12d79fd53fbf58626b

Contents

applyGenerateProbs *Generate probabilities missing from RDBES Data*

Description

Wrapper to generate probabilities. The wrapper calls runChecksOnSelectionAndProbs which main tests need to be passed before probabilities can be calculated. The it calls generateProbs for each sample in each sampling level of the hierarchy.

Usage

```
applyGenerateProbs(
  x,
  probType,
  overwrite,
  runInitialProbChecks = TRUE,
  strict = TRUE
)
```

Arguments

`x`
- RDBES data object

`probType`
- string. Can be set to "selection" (only selection probabilities are calculated), "inclusion" (only inclusion probabilities are calculated) or "both" (both types of probabilities are calculated)

`overwrite`
- if TRUE will overwrite probabilities already existing for SRSWR and SRSWOR

`runInitialProbChecks`
- if TRUE runs runChecksOnSelectionAndProbs

`strict` (Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Value

a list of all the RDBES data tables with probabilites calculated

See Also

runChecksOnSelectionAndProbs generateProbs

Examples

```
# To be added
```

---

check_key_column   *Check if a column exists in a data table and has unique values*

Description

This function checks if a specified column exists in a given data table and has unique values. If the column does not exist or has non-unique values, an error is thrown.

Usage

```
check_key_column(dt, col)
```

Arguments

dt	A data table to check
col	A character string specifying the name of the column to check

Value

nothing if the column exists and has unique values, otherwise an error is thrown

Examples

```
## Not run:
  RDBEScore:::check_key_column(H1Example$DE, "DEid")

## End(Not run)
```

combineRDBESDataObjects

Combine Two RDBES Raw Objects combines 2 RDBESDataObjects into a single RDBESDataObject by merging individual tables one by one

Description

Combine Two RDBES Raw Objects combines 2 RDBESDataObjects into a single RDBESDataObject by merging individual tables one by one

Usage

```
combineRDBESDataObjects(RDBESDataObject1, RDBESDataObject2, strict = TRUE)
```

Arguments

`RDBESDataObject1`
The first object to combine

`RDBESDataObject2`
The second object to combine

`strict` (Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Value

the combination of `RDBESDataObject1` and `RDBESDataObject2`

See Also

rbindlist

Examples

```
## Not run:

myH1RawObject <-
    importRDBESDataCSV(rdbesExtractPath = "tests\\testthat\\h1_v_1_19")
myH5RawObject <-
    importRDBESDataCSV(rdbesExtractPath = "tests\\testthat\\h5_v_1_19")

myCombinedRawObject <- combineRDBESDataObjects(RDBESDataObject1=myH1RawObject,
                                             RDBESDataObject2=myH5RawObject)

## End(Not run)
```

createDBEPrepObj *Load raw object and create prepared object Function relies on the data being correctly named following established hierarchy*

Description

Load raw object and create prepared object Function relies on the data being correctly named following established hierarchy

Usage

```
createDBEPrepObj(input, output)
```

Arguments

`input`	a string pointing towards the input folder
`output`	a string pointing towards the output folder

Value

.Rdata files

Examples

```
## Not run:
input <- "WKRDB-EST2/testData/output/DBErawObj/"
output <- "WKRDB-EST2/subGroup1/personal/John/PreparedOutputs/"

createDBEPrepObj(input = input, output = output)

## End(Not run)
```

createRDBESDataObject *Create an RDBES Data Object*

Description

This function lets you create an RDBES Data object in your current R environment.

Usage

```
createRDBESDataObject(
  input = NULL,
  listOfFileNames = NULL,
  castToCorrectDataTypes = TRUE,
  ...
)
```

Arguments

`input` Strings or `list` object. The path to the zip file downloaded from RDBES (or multiple zip files - see details), or path to a folder of `csv` files, or a `list` object in the current environment containing data frames of each table. If `NULL` an empty `RDBESDataObject` is created.

`listOfFileNames`
`list` of Strings, Optional. For use with `csv` inputs only, and only required if the csv file names are *not* the default file names used by RDBES when downloading data (for instance if you created the files yourself). The actual file names should be a `list` of the two-letter code for the relevant table e.g. `list("DE" = "DE.csv", "SD" = "SD.csv", etc.)`. If not used then it is assumed the files have the default file names used by the RDBES data download ("Design.csv" etc).

`castToCorrectDataTypes`
Logical. If `TRUE` then the function will attempt to cast the required columns to the correct data type. If `FALSE` then the column data types will be determined by how the csv files are read in. Default is `TRUE`.

`...` parameters passed to validateRDBESDataObject if input is list of data frames e.g.`strict=FALSE`

Details

The `input` should be either:

- A `zip` file downloaded from RDBES (or multiple zip files if you want to include or overwrite tables, for example CL and CE data)
- A folder containing `csv` files downloaded from RDBES (e.g. the unzipped file), or any set of csv files of the RDBES tables.
- A `list` of data frames in the current environment representing different tables in the hierarchy.
- A `NULL` input will return and empty RDBES data object

ZIP file inputs This `input` should be a path to a zip file downloaded from RDBES. Multiple zip files can be entered if you want to include additional tables, for example CL and CE. E.g. ‘input = c("path/to/H1.zip", "path/to/CL.zip"). If any tables in the first input are overwritten by other inputs a warning is given. You should not input different hierarchy files; this function will not combine them.

CSV file inputs This `input` should be a path to a folder of `csv` files. These can be the `csv` files downloaded from RDBES (e.g. an unzipped hierarchy), or *any* set of csv files containing RDBES tables. If the files do not have the default RDBES name (e.g. 'Design.csv') the `listOfFileNames` input can by used to specify the file names e.g. `list("DE" = "DE.csv", "SD" = "SD.csv", etc.)`.

List of data frames inputs This `input` should be a `list` object containing data frames (or data.tables) for each table in your hierarchy. They should be named with the appropriate 2-letter code (`DE`, `SD`, etc.). Columns within these tables will be renamed to the RDBES model documentation 'R name'. Note if you choose to create an `RDBESDAtaObject` from local data frames these may have not passed the data integrity checks performed when you upload to RDBES!

NULL inputs This `input` produces an empty `RDBESDataObject`, i.e. all tables with correct data classes but the tables will be empty.

Value

A RDBESDataObject

Examples

```
myEmptyRDBESObject <- createRDBESDataObject(input = NULL)
```

`createRDBESEstObject` *Creates an rdbesEStObject from prepared RDBES data*

Description

Creates an rdbesEStObject from prepared RDBES data

Usage

```
createRDBESEstObject(
  rdbesPrepObject,
  hierarchyToUse = NULL,
  stopTable = NULL,
  verbose = FALSE,
  strict = TRUE
)
```

Arguments

`rdbesPrepObject`
The prepared RDBES object that should be used to create an estimation object

`hierarchyToUse` The upper RDBES hiearchy to use

`stopTable` (Optional) The table to stop at in the RDBES hierarchy. If specified, only tables up to and including this table will be included in the resulting RDBESEstObject. The default is NULL, which means all tables in the hierarchy will be included.

`verbose` (Optional) Set to TRUE if you want informative text printed out, or FALSE if you don't. The default is FALSE.

`strict` (Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Value

An object of class RDBESEstObject ready for use in design based estimation

An object of class RDBESEstObject ready for use in design based estimation

Examples

```
#Creates an rdbesEStObject from prepared RDBES data
myH1EstObj <- createRDBESEstObject(H1Example, 1, "SA")


myH1EstObj <- createRDBESEstObject(H1Example, 1, "SA")
```

`createTableOfRDBESIds` *Create a table of RDBES Ids*

Description

examples for now see https://github.com/ices-eg/WK_RDBES/tree/master/WKRDB-EST2/chairs/Nuno

Usage

```
createTableOfRDBESIds(x, addSAseqNums = TRUE)
```

Arguments

`x`	RDBESdataObject
`addSAseqNums`	should SAseqNum be included? Default value is TRUE

Value

data frame of Ids of all tables in sampling hierarchy

Examples

```
## Not run:

myH1RawObject <-
  importRDBESDataCSV(rdbesExtractPath = "tests\\testthat\\h1_v_1_19_13")

myTableOfIds<- createTableOfRDBESIds(myH1RawObject)

## End(Not run)
```

designVariables *A dataset containing the RDBES "design variable" names*

Description

A dataset containing the RDBES "design variable" names

Usage

```
designVariables
```

Format

A vector containing the short R names of the RDBES design variables (without any 2 letter table prefixes) R field name:

designVariables The design variable names

Source

https://sboxrdbes.ices.dk

doDBEestimantionObjUpp *Generates the DBE estimation object for the upper hierarchy tables*

Description

Generates the DBE estimation object for the upper hierarchy tables

Usage

```
doDBEestimantionObjUpp(inputList)
```

Arguments

`inputList` All the data tables in a named list. Name should be equal to the short table names e.g. DE, SD, TE, FO.

Value

The upper hierarchy tables in the DBE estimation object (DBEestimantionObjUpp)

Examples

```
## Not run:
H1 <-
readRDS("./WKRDB-EST2/testData/output/DBErawObj/DBErawObj_DK_1966_H1.rds")
H1out <- doDBEestimantionObjUpp(H1)

## End(Not run)
```

---

doDBestimation *Create design-based point and variance estimates from RDBES estimation object (rdbesEstimObj)*

Description

Create design-based point and variance estimates from RDBES estimation object (rdbesEstimObj)

Usage

```
doDBestimation(
  x = rdbesEstimObj,
  estimateType = "total",
  pointEstimator = "Unbiased",
  varEstimator = "WRonPSUviaPik",
  stage = 0,
  domainOfinterest = NULL
)
```

Arguments

`x` a data.frame (or data.table) in rdbesEstimObj format with value of target variable in column targetValue

`estimateType` a string with type of estimate. As of now only "total" is defined

`pointEstimator` a string with type of point estimator. As of now only "Unbiased" is defined

`varEstimator` a string with type of variance estimator. As of now only "WRonPSUviaPik" is defined

`stage` a natural number (0,1,..) with sampling stage of estimate. 0 corresponds to DE level.

`domainOfinterest` list ofdomains of interest (e.g., SAarea). As of now only NULL (=no domain estimate) is defined

Value

a list of values for pointEstimate, varEstimate and estimation options

Examples

```
## Not run:
data(shrimps)
doDBestimation (x = shrimps, estimateType = "total",
pointEstimator = "Unbiased", varEstimator = "WRonPSUviaPsi",stage = 0,
domainOfinterest = NULL )

## End(Not run)
```

`doEstimationForAllStrata`

Estimate totals and means, and try to generate samples variances for all strata in an RDBESEstObject

---

Description

Estimate totals and means, and try to generate samples variances for all strata in an RDBESEstObject

Usage

```
doEstimationForAllStrata(RDBESEstObjectForEstim, targetValue, verbose = FALSE)
```

Arguments

`RDBESEstObjectForEstim`	The RDBESEstObject to generate estimates for
`targetValue`	The field to estimate for, for example "SAsampWtLive"
`verbose`	(Optional) If set to TRUE more detailed text will be printed out by the function. Default is FALSE

Value

A data frame containing estimates for all strata

Examples

```
## Not run:

myH1RawObject <-
  importRDBESDataCSV(rdbesExtractPath = "tests\\testthat\\h1_v_1_19")

# Update our test data with some random sample measurements
myH1RawObject[["SA"]]$SAsampWtLive <-
  round(runif(n = nrow(myH1RawObject[["SA"]]), min = 1, max = 100))

myH1EstObj <- createRDBESEstObject(myH1RawObject, 1)
```

```
myStrataEst <- doEstimationForAllStrata(
  RDBESDataObjectForEstim = myH1EstObj,
  targetValue = 'SAsampWtLive'
)

## End(Not run)
```

estim *Generic function for estimation of population total and variance*

Description

Generic function for estimation of population total and variance

Usage

```
estim(y, enk, enkl, method = "SRSWOR", estFunction, varFunction)
```

Arguments

y	numeric variable to be estimated
enk	expected value of k
enkl	expected value of k, given l
method	character selection method code e.g SRSWOR
estFunction	the function to use to estimate total given parameters y and enk
varFunction	the function to use to estimate variance given parameters y,enk and enkl

Value

list of 7 elements including the population mean, total (and their variance), the algorithm name used and the I order inclusion probabilities

Examples

```
estimMC(c(3, 4, 4, 5), c(4, 4, 4, 4), c(8, 8, 8, 8))
```

estimMC *Multiple Count Estimator for Population Total and Variance*

Description

Multiple Count Estimator for Population Total and Variance

Usage

```
estimMC(y, sampled, total, method = "SRSWOR", selProb = NULL, incProb = NULL)
```

Arguments

y	numeric variable to be estimated
sampled	numeric total number of units sampled
total	numeric total number of units int the population
method	character selection method code e.g SRSWOR
selProb	the selection probabilities (if known)
incProb	the inclusion probabilities (if known)

Value

list of 7 elements including the population mean, total (and their variance), the algorithm name used and the I order inclusion probabilities

Examples

```
estimMC(c(3, 4, 4, 5), c(4, 4, 4, 4), c(8, 8, 8, 8))
```

filterAndTidyRDBESDataObject *Filter and remove orphan records in an RDBESDataObject*

Description

This function filters an RDBESDataObject based on specified fields and values, and can optionally remove any orphan records. The returned object will include all rows which either: a) do not include any of the field names in `fieldsToFilter`, or b) do include the field names and have one of the allowed values in `valuesToFilter`. If `killOrphans` is set to `TRUE`, the function will remove orphaned rows. The default is `FALSE`.

Usage

```
filterAndTidyRDBESDataObject(
  RDBESDataObjectToFilterAndTidy,
  fieldsToFilter,
  valuesToFilter,
  killOrphans = FALSE,
  verboseOrphans = FALSE,
  verboseBrokenVesselLinks = FALSE
)
```

Arguments

`RDBESDataObjectToFilterAndTidy`	The RDBESDataObject to filter.
`fieldsToFilter`	A vector of the field names you wish to check.
`valuesToFilter`	A vector of the field values you wish to filter for.
`killOrphans`	Controls if orphan rows are removed. Default is `FALSE`.
`verboseOrphans`	Controls if verbose output for orphan rows is printed. Default is `FALSE`.
`verboseBrokenVesselLinks`	Controls if verbose output for broken vessel links is printed. Default is `FALSE`.

Value

The filtered input object of the same class as `RDBESDataObjectToFilterAndTidy`.

Examples

```
## Not run:

myH1RawObject <- createRDBESDataObject(rdbesExtractPath = "tests\\testthat\\h1_v_1_19_13")

# To check how removeBrokenVesselLinks() works
myH1RawObject$VD$VDlenCat[which(myH1RawObject$VD$VDencrVessCode=="VDcode_10")] <- "VL40XX"

myFields <- c("VSencrVessCode", "VDlenCat")
myValues <- c("VDcode_1","VDcode_2", "VDcode_10","VL1518","VL2440")

myFilteredObject <- filterAndTidyRDBESDataObject(myH1RawObject,
  fieldsToFilter = myFields,
  valuesToFilter = myValues,
  killOrphans = TRUE, verboseBrokenVesselLinks = TRUE
)

## End(Not run)
```

filterRDBESDataObject *Filter an RDBESDataObject*

Description

The returned object will include all rows which either: a) do not included any of the field names in `fieldsToFilter`, or b) do include the field names and have one of the allowed values in `valuesToFilter`. If you want to filter for a id field like `DEid`, `FTid` etc, the filtering works only on the table where the id field is its key. For example, if you try to filter on `FOid` it does not look `FOid` in other tables like `FT`, although the field `FOid` exists in `FT` table.

Usage

```
filterRDBESDataObject(
  RDBESDataObjectToFilter,
  fieldsToFilter,
  valuesToFilter,
  killOrphans = FALSE,
  verbose = FALSE,
  strict = TRUE
)
```

Arguments

`RDBESDataObjectToFilter`
The `RDBESDataObject` to filter

`fieldsToFilter` A vector of the field names you wish to check

`valuesToFilter` A vector of the field values you wish to filter for

`killOrphans` Controls if orphan rows are removed. Default is `FALSE`.

`verbose` (Optional) Set to TRUE if you want informative text printed out, or FALSE if you don’t. The default is FALSE.

`strict` (Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Details

`killOrphans` allows you to remove orphaned rows if set to `TRUE`. The default is `FALSE`.

Value

the filtered input object of the same class as `RDBESDataObjectToFilter`

Examples

```
## Not run:

myH1RawObject <-
  importRDBESDataCSV(rdbesExtractPath = "tests\\testthat\\h1_v_1_19")

myFields <- c("SDctry", "VDctry", "VDflgCtry", "FTarvLoc")
myValues <- c("ZW", "ZWBZH", "ZWVFA")

myFilteredObject <- filterRDBESDataObject(myH1RawObject,
  fieldsToFilter = myFields,
  valuesToFilter = myValues
)

## End(Not run)
```

`findAndKillOrphans`

This function finds and removed any orphan records in an RDBESDataObject. Normally data that has been downloaded from the RDBES will not contain orphan records - however if the data is subsequently filtered it is possible to introduce orphan records.

Description

This function finds and removed any orphan records in an RDBESDataObject. Normally data that has been downloaded from the RDBES will not contain orphan records - however if the data is subsequently filtered it is possible to introduce orphan records.

Usage

```
findAndKillOrphans(objectToCheck, verbose = FALSE, strict = TRUE)
```

Arguments

`objectToCheck`	an RDBESDataObject.
`verbose`	(Optional) If set to TRUE more detailed text will be printed out by the function. Default is TRUE.
`strict`	(Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Value

an RDBESDataObject with any orphan records removed

Examples

```
## Not run:

myH1RawObject <-
importRDBESDataCSV(rdbesExtractPath = "tests\\testthat\\h1_v_1_19")
myFields <- c("SDctry","VDctry","VDflgCtry","FTarvLoc")
myValues <- c("ZW","ZWBZH","ZWVFA" )
myFilteredObject <- filterRDBESDataObject(myH1RawObject,
                                         fieldsToFilter = myFields,
                                         valuesToFilter = myValues )
myObjectNoOrphans <- findAndKillOrphans(objectToCheck = myFilteredObject,
                                        verbose = FALSE)

## End(Not run)
```

findOrphansByTable *Internal function to identify orphan records in a given RDBESDataObject table*

Description

Internal function to identify orphan records in a given RDBESDataObject table

Usage

```
findOrphansByTable(tableToCheck, objectToCheck, foreignKeyIds, verbose = FALSE)
```

Arguments

`tableToCheck`	The two letter code for the table to check
`objectToCheck`	An RDBESDataObject
`foreignKeyIds`	A vetor of the foreign key field names to check
`verbose`	(Optional) If set to TRUE more detailed text will be printed out by the function. Default is TRUE.

Value

A data frame with the primary keys of the table checked, the two letter table identifier, and their orphan status.

`fixSLids` *Fixes SLid in SL table (facilitating SS-SL joins).*

Description

Fixes SLid in SL table (facilitating SS-SL joins).

Usage

```
fixSLids(RDBESDataObject, verbose = FALSE, strict = TRUE)
```

Arguments

`RDBESDataObject`	A valid RDBESDataObject
`verbose`	(Optional) Set to TRUE if you want informative text printed out, or FALSE if you don't. The default is FALSE.
`strict`	(Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Details

RDBES SL can be seen as a join of two tables - one that identifies the species list in terms of *SLcou * SLinst * SLspeclistName * SLyear * SLcatchFrac* and one that specifies the taxa (*SLcommTaxon * SLsppCode*) in the list. In SS, SLid remits to the 1st taxa in a species list and not - as it would be expected - to the species list itself. This function fixes this by creating a new SLtaxaId variable in SL and assigning all taxa in a species to a single SSid.

Value

an RDBESDataObject with SL ids reworked

Examples

```
# To add
```

`generateMissingSSRows` *Generate any missing SS rows. When FOcatchReg=="All" it is expected that SScatchFraction is either "Catch" OR "Lan"+"Dis". In the latter case, if one is missing the other is to be assumed 0. This function generates SS rows for any missing catch fractions.*

Description

Generate any missing SS rows. When FOcatchReg=="All" it is expected that SScatchFraction is either "Catch" OR "Lan"+"Dis". In the latter case, if one is missing the other is to be assumed 0. This function generates SS rows for any missing catch fractions.

Usage

```
generateMissingSSRows(
  RDBESDataObject,
  speciesListName,
  verbose = FALSE,
  strict = TRUE
)
```

Arguments

`RDBESDataObject`
A valid RDBESDataObject

`speciesListName`
The name of the Species List you want to use for any SS rows that are created.

`verbose`
(Optional) Set to TRUE if you want informative text printed out, or FALSE if you don't. The default is FALSE.

`strict`
(Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Value

A data table of SS data with any missing rows added

Examples

```
# To follow
```

generateNAsUsingSL *Generate NAs in samples using Species List information*

Description

Generate NAs in samples using Species List information

Usage

```
generateNAsUsingSL(
  RDBESDataObject,
  targetAphiaId,
  overwriteSampled = TRUE,
  validate = TRUE,
  verbose = FALSE,
  strict = TRUE
)
```

Arguments

`RDBESDataObject`
An RDBESDataObject.

`targetAphiaId` a vector of aphiaId.

`overwriteSampled`
(Optional) should SAtotalWtMes and SAsampWtMes be set to 0 if spp recorded but absent from SL? The default is TRUE.

`validate` (Optional) Set to TRUE if you want validation to be carried out. The default if TRUE.

`verbose` (Optional) Set to TRUE if you want informative text on validation printed out, or FALSE if you don't. The default is FALSE.

`strict` (Optional) This function can validate its input data - should the validation be strict? The default is TRUE.

Value

RDBES data object where SA was complemented with NAs for species not looked for (sensu in SL)

Examples

```
# To be added
```

---

generateProbs *Generate vector of selection or inclusion probabilities*

Description

Generate vector of selection or inclusion probabilities

Usage

```
generateProbs(x, probType)
```

Arguments

`x` RDBES data object

`probType` "selection" or "inclusion" for selection and inclusion probabilities respectively

Details

When the selection method is SRSWR selection probabilities are calculated as $1/N$ and inclusion probabilities as $1-(1-1/N)^n$. When the selection method is SRSWOR selection probabilities are not currently implemented. Inclusion probabilities are calculated as n/N. When the selection method is CENSUS both types of probabilities are set to 1. Probabilities for selection methods UPSWR and UPSWOR are not calculated (they need to be supplied by the user). The same happens with regards to non-probabilistic methods

Value

A vector or probabilities

Examples

```
## Not run:
generateProbs(x = Pckg_SDAResources_agstrat_H1[["VS"]], probType = ("inclusion"))
# population size
a<-generateProbs(x = Pckg_SDAResources_agstrat_H1[["VS"]], probType = ("inclusion"))
sum(1/a$VSincProb)
# returns error
generateProbs(x = Pckg_SDAResources_agstrat_H1[["VS"]], probType = ("selection"))


## End(Not run)
```

`generateSSRows` *Private function to generate SS rows*

Description

Private function to generate SS rows

Usage

```
generateSSRows(FOids, speciesListName, catchFra)
```

Arguments

`FOids`	Vector of FOids
`speciesListName`	Name of the species list
`catchFra`	The catch fraction to create

Value

SS data frame

generateTestTbls *Generate a List of Related Data Tables*

Description

Generates a named list of data tables that follow the structure of RDBESDataObject. The tables only have columns required for testing The generate tables

Usage

```
generateTestTbls(tblNames, prevTbls = list(), ...)
```

Arguments

`tblNames`	character vector of table names to be created
`prevTbls`	list of data.tables upstream of the generated table. Defaults to empty list
`...`	Arguments passed on to `makeTbl` `tblName` Name of the table `rows` numeric number of rows per parent record. Defaults to 4. `propSamp` numeric proportion of how many of total are sampled. This is ignored for "CENSUS". Defaults to 0.5 `selMeth` character selection method used. Defaults to "CENSUS". Others like SRSWR or SRSSWOR can be used as well `stratums` character vector of the stratum names to be created. Defaults to c("U"), meaning not stratified. `mean` numeric the expected mean of the target variable. The variable is created using `rnorm` and saved under column ending with "y". Defaults to 5.

Value

a list of named data.table's

Examples

```
## Not run:
generateTestTbls(c("A", "B", "C"), selMeth = "SRSWOR")
generateTestTbls(LETTERS[1:5]) # makes 5 tables with method CENSUS

## End(Not run)
```

generateZerosUsingSL *Generate zeros in samples using Species List information*

Description

examples for now see https://github.com/ices-eg/WK_RDBES/tree/master/WKRDB-EST2/chairs/Nuno

Usage

```
generateZerosUsingSL(x, verbose = FALSE, strict = TRUE)
```

Arguments

x	RDBES data frame
verbose	(Optional) Set to TRUE if you want informative text printed out, or FALSE if you don't. The default is FALSE.
strict	(Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Value

RDBES data frame where SA was complemented with species looked for (sensu in sampling objectives) but not registered in sample

getEstimForStratum *Private function used by doEstimationForAllStrata to get the estimates*

Description

Private function used by doEstimationForAllStrata to get the estimates

Usage

```
getEstimForStratum(x)
```

Arguments

x	The input

Value

Data frame with estimated values

getMissingSSCatchFraction *Private function to find which FO rows are not matching SS*

Description

Private function to find which FO rows are not matching SS

Usage

```
getMissingSSCatchFraction(FOdata, SSdata, catchFra, verbose)
```

Arguments

`FOdata`	The FOdata
`SSdata`	The SSdata
`catchFra`	The catchfra
`verbose`	verbose or not?

Value

Vector of FOids that aren't matching SS rows

getSubSampleLevel *Private function to get sub-sample level and top-level SAid for SA data*

Description

Private function to get sub-sample level and top-level SAid for SA data

Usage

```
getSubSampleLevel(SAdata, SAidToCheck, subSampleLevel = 1)
```

Arguments

`SAdata`	The SA data to check
`SAidToCheck`	The SAid to check
`subSampleLevel`	The currrent level of sampling

Value

Whoever revises this function please specify what it returns here

getTablesInRDBESHierarchy

Returns the tables for a given hierarchy

Description

Returns the tables for a given hierarchy

Usage

```
getTablesInRDBESHierarchy(
  hierarchy,
  includeOptTables = TRUE,
  includeLowHierTables = TRUE,
  includeTablesNotInSampHier = TRUE
)
```

Arguments

`hierarchy`	Integer value between 1 and 13 inclusive
`includeOptTables`	Include any optional tables? Default value is TRUE
`includeLowHierTables`	Include the lower hierarchy tables? Default value is TRUE
`includeTablesNotInSampHier`	Include tables that aren't sampling units in that hierarcy? Default value is TRUE

Value

A vector containing the 2-letter names of the tables in the requested hierarchy

Examples

```
getTablesInRDBESHierarchy(5)
```

H1Example

A dataset containing test RDBES data for H1 in the RDBESDataObject structure

Description

A dataset containing test RDBES data for H1 in the RDBESDataObject structure

Usage

```
H1Example
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table

SD the Sampling Details data table

VS the Vessel Selection data table

FT the Fishing Trip data table

FO the Fishing Operation data table

SS the Species Selection data table

SA the Sample data table

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table

`H5Example` *A dataset containing test RDBES data for H5 in the RDBESDataObject structure*

Description

A dataset containing test RDBES data for H5 in the RDBESDataObject structure

Usage

```
H5Example
```

Format

A list containing entries required for H5 RDBES data:

DE the Design data table

SD the Sampling Details data table

FT the Fishing Trip data table

OS the Onshore Event data table

LE the Landing Event data table

SS the Species Selection data table

SA the Sample data table

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table

H8ExampleEE1 *A dataset containing test RDBES data for H8 in the RDBESDataObject structure*

Description

This dataset does not have passed the RDBES upload checks, hence the object might be somewhat invalid, however it resembles real data from the Estonian Baltic Trawling fleet for 2022 sprat total landings and commercial sampling

Usage

```
H8ExampleEE1
```

Format

A list containing entries required for H8 RDBES data:

DE the Design data table

SD the Sampling Details data table

TE the Temporal Event data table

VS the Vessel Selection data table

LE the Landing Event data table

SS the Species Selection data table

SA the Sample data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table

CL the Commertial Landing data table

CE the Commertial Effort data table

#' @source Richard Meitern @ Estonian Marine Institute, 2023 `icesSpecWoRMS` *A dataset containing a copy of the icesSpecWoRMS code list. The latest code list data can be downloaded from https://vocab.ices.dk/*

Description

A dataset containing a copy of the icesSpecWoRMS code list. The latest code list data can be downloaded from https://vocab.ices.dk/

Usage

```
icesSpecWoRMS
```

Format

A data frame

GUID Globally unique identifier assigned by ICES

Key AphiaID

Description Scientific name

LongDescription Ignore

Modified Date when the code was last updated

Deprecated IS this still a valid code. If FALSE the code is no longer valid within ICES.

DateDownloaded E.g. "2023-10-18" ...

Source

https://vocab.ices.dk/

`killOrphans` *Internal function to remove orphan records from an RDBESDataObject*

Description

Internal function to remove orphan records from an RDBESDataObject

Usage

```
killOrphans(objectToCheck, orphansToRemove)
```

Arguments

`objectToCheck`	an RDBESDataObject
`orphansToRemove`	The output from the findOrphansByTable function (A data frame with the primary keys of the table checked, the two letter table identifier, and their orphan status.)

Value

RDBESDataObject with orphan records removed

`MadeUpData_for_SL_SA_tests_v1` *A made-up dataset for testing manipulations of SA based on SL*

Description

This data set is created for testing the idea of manipulating Sample data (SA) based on Species List (SL). It represents the simplest case for testing this idea. The data set contains two species in SL for the same SLcountry, SLinstitute, SLspeciesListName, SLyear, SLcatchFraction, SLcommercialTaxon, SLspeciesCode & SLcommercialTaxon == SLspeciesCode. There is one species in SA - one row in SS with keys equal to the SL keys.

Usage

```
MadeUpData_for_SL_SA_tests_v1
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table

SD the Sampling Details data table

VS the Vessel Selection data table

FT the Fishing Trip data table

FO the Fishing Operation data table

SS the Species Selection data table. Contains one row with keys equal to the SL keys

SA the Sample data table. Contains one species

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table. Contains two species for the same SLcountry, SLinstitute, SLspeciesListName, SLyear, SLcatchFraction, SLcommercialTaxon, SLspeciesCode & SLcommercialTaxon == SLspeciesCode

makeTbl *Generate a Data Table*

Description

Generate a Data Table

Usage

```
makeTbl(
  tblName,
  prevTbls = list(),
  rows = 4,
  propSamp = 0.5,
  selMeth = "CENSUS",
  stratums = c("U"),
  mean = 5
)
```

Arguments

`tblName`	Name of the table
`prevTbls`	list of data.tables upstream of the generated table. Defaults to empty list
`rows`	numeric number of rows per parent record. Defaults to 4.
`propSamp`	numeric proportion of how many of total are sampled. This is ignored for "CENSUS". Defaults to 0.5
`selMeth`	character selection method used. Defaults to "CENSUS". Others like SRSWR or SRSSWOR can be used as well
`stratums`	character vector of the stratum names to be created. Defaults to c("U"), meaning not stratified.
`mean`	numeric the expected mean of the target variable. The variable is created using `rnorm` and saved under column ending with "y". Defaults to 5.

Value

a data.table

mapColNamesFieldR *A dataset containing the mapping from database column names to R field names*

Description

A dataset containing the mapping from database column names to R field names

Usage

```
mapColNamesFieldR
```

Format

A data frame containing database field names and their equivalent R field name:

Table.Prefix The two letter prefix of the relevent RDBES table

Field.Name The database field names

R.Name The equivalent R field name

RDataType The equivalent R data type (e.g. "integer", "character" etc)

Type The Data type in the RDBES documentation (e.g. "Decimal", etc)

EssentialForEst Is this column considered essential? ...

Source

https://sboxrdbes.ices.dk

newRDBESDataObject *Constructor for RDBESDataObject class*

Description

Constructor for RDBESDataObject class

Usage

```
newRDBESDataObject(
  DE = NULL,
  SD = NULL,
  VS = NULL,
  FT = NULL,
  FO = NULL,
  TE = NULL,
  LO = NULL,
  OS = NULL,
  LE = NULL,
  SS = NULL,
  SA = NULL,
  FM = NULL,
  BV = NULL,
  VD = NULL,
  SL = NULL,
  CL = NULL,
  CE = NULL
)
```

Arguments

DE	Data table of RDBES DE data or null
SD	Data table of RDBES DE data or null
VS	Data table of RDBES DE data or null
FT	Data table of RDBES DE data or null
FO	Data table of RDBES DE data or null
TE	Data table of RDBES DE data or null
LO	Data table of RDBES DE data or null
OS	Data table of RDBES DE data or null
LE	Data table of RDBES DE data or null
SS	Data table of RDBES DE data or null
SA	Data table of RDBES DE data or null
FM	Data table of RDBES DE data or null
BV	Data table of RDBES DE data or null
VD	Data table of RDBES DE data or null
SL	Data table of RDBES DE data or null
CL	Data table of RDBES DE data or null
CE	Data table of RDBES DE data or null

Value

a named list

Pckg_SDAResources_agsrs_H1

A RDBESDataObject converted from package SDAResources dataset agsrs

Description

This data set is derived from the data(agsrs) used in Lohr examples 2.6, 2.7 and 2.11 of SDA book. Information required for example 4.8 (domain estimation) is also added to SA (farmcat <=> SAarea). VSnumberSampled and VSnumberTotal set according to agsrs and book pop values. VSunitName is set to a combination of original agsrs$county, agsrs$state, agsrs$region and row numbers. Table SA contains the variable measured agsrs$acres92 in SAtotalWeightMeasured, SAsampleWeightMeasured and SAconversionFactorMeasLive set to 1. Table SA also contains the domain information, coded in SAarea. Table DE, SD, FT and FO are for the most dummy tables inserted to meet RDBES model requirements to be aggregated during estimation tests. Values of mandatory fields have dummy values with exception of Design-Variables in VS that match the book. BV, FM, CL, and CE are not provided. SL and VD are subset to the essential rows.

Usage

```
Pckg_SDAResources_agsrs_H1
```

Format

A list containing entries required for H1 RDBES data:

- **DE** the Design data table. Contains dummy values with exception of Design-Variables in VS that match the book
- **SD** the Sampling Details data table. Contains dummy values
- **VS** the Vessel Selection data table. Contains core information of data(agsrs), VSnumberSampled and VSnumberTotal set according to agsrs and book pop values, VSunitName is set to a combination of original agsrs$county, agsrs$state, agsrs$region and row numbers
- **FT** the Fishing Trip data table. Contains dummy values
- **FO** the Fishing Operation data table. Contains dummy values
- **SS** the Species Selection data table. Contains dummy values
- **SA** the Sample data table. Contains the variable measured agsrs$acres92 in SAtotalWeightMeasured, SAsampleWeightMeasured and SAconversionFactorMeasLive set to 1, and the domain information, coded in SAarea
- **FM** the Frequency Measure data table. Not provided
- **BV** the Biological Variable data table. Not provided
- **VD** the Vessel Details data table. Subset to the essential rows
- **SL** the Species List data table. Subset to the essential rows

Source

https://CRAN.R-project.org/package=SDAResources

Pckg_SDAResources_agstrat_H1 *A RDBESDataObject converted from package SDAResources dataset agstrat*

Description

This data set is derived from the data(agstrat) used in Lohr examples 3.2 and 3.6. Table VS is stratified with VSstratumName set to agstrat$region, and VSnumberSampled and VSnumberTotal set according to agstrat. VSunitName is set to a combination of original agstrat$county, agstrat$state, agstrat$region and agstrat$agstrat row numbers. Table SA contains the variable measured agstrat$acres92 in SAtotalWeightMeasured, SAsampleWeightMeasured and SAconversionFactorMeasLive set to 1. Table DE, SD, FT and FO are for the most dummy tables inserted to meet RDBES model requirements to be aggregated during estimation tests. Values of mandatory fields have dummy values taken from an onboard programme, with exception of selectionMethod that is set to CENSUS. BV, FM, CL, and CE are not provided. SL and VD are subset to the essential rows.

Usage

```
Pckg_SDAResources_agstrat_H1
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table. Contains dummy values with exception of selectionMethod that is set to CENSUS

SD the Sampling Details data table. Contains dummy values

VS the Vessel Selection data table. Contains core information of data(agstrat), VSstratumName set to agstrat$region, and VSnumberSampled and VSnumberTotal set according to agstrat, VSunitName is set to a combination of original agstrat$county, agstrat$state, agstrat$region and agstrat$agstrat row numbers

FT the Fishing Trip data table. Contains dummy values

FO the Fishing Operation data table. Contains dummy values

SS the Species Selection data table. Contains dummy values

SA the Sample data table. Contains the variable measured agstrat$acres92 in SAtotalWeightMeasured, SAsampleWeightMeasured and SAconversionFactorMeasLive set to 1

FM the Frequency Measure data table. Not provided

BV the Biological Variable data table. Not provided

VD the Vessel Details data table. Subset to the essential rows

SL the Species List data table. Subset to the essential rows

Source

https://CRAN.R-project.org/package=SDAResources

`Pckg_SDAResources_algebra_H1`

A RDBESDataObject converted from package SDAResources dataset algebra

Description

This data set is derived from a fictional data for an SRS of 12 algebra classes in a city, from a population of 187 classes. The design is 1-stage cluster sampling with clusters of unequal sizes. Clusters are classes (class). Clusters (psu) are unequal sized (Mi). In each cluster, all students are selected (ssus, nrows). The total number of psus is known (187). The target variable is score.

Usage

```
Pckg_SDAResources_algebra_H1
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table. Contains 1 DE row with DEstratumName == "Pckg_SDAResources_algebra_H1"

SD the Sampling Details data table. Contains 1 child SD row

VS the Vessel Selection data table. Contains 299 child rows (the 299 students observed), each associated to its cluster (class), VSnumberTotalClusters is 187, VSnumberSampledClusters is 12, VSnumberTotal is Missing

FT the Fishing Trip data table. Just 1:1 links to the final data (in SA)

FO the Fishing Operation data table. Just 1:1 links to the final data (in SA)

SS the Species Selection data table. Just 1:1 links to the final data (in SA)

SA the Sample data table. Each score is a SAsampleWeightMeasured

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table

Source

https://CRAN.R-project.org/package=SDAResources

Pckg_SDAResources_coots_H1

A RDBESDataObject converted from package SDAResources dataset coots

Description

This data set is derived from the data(coots). The design is 2-stage cluster sampling with clusters of unequal sizes and Npsu not known. Clusters are clutches of eggs (nests) with at least 2 eggs. In each cluster, the volume of two eggs is measured. Clusters (psu) are unequal sized. In each cluster, 2 eggs are selected (ssus) and measured. The total number of psus is not known (a drawback in this example). It is assumed very large (fpc negligible). The target variable is volume (others are available).

Usage

```
Pckg_SDAResources_coots_H1
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table. Contains 1 DE row with DEstratumName == "Pckg_SDAResources_coots_H1"

SD the Sampling Details data table. Contains 1 child SD row

VS the Vessel Selection data table. Contains 368 child rows (the 368 eggs/psus observed), each associated to its cluster (clutch), VSnumberTotalClusters is not known, VSnumberTotal is csize

FT the Fishing Trip data table. Just 1:1 links to the final data (in SA)

FO the Fishing Operation data table. Just 1:1 links to the final data (in SA)

SS the Species Selection data table. Just 1:1 links to the final data (in SA)

SA the Sample data table. Each volume is a SAsampleWeightMeasured. ATT volumes are *100000000 to meet type requirement (integer)

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table

Source

https://CRAN.R-project.org/package=SDAResources

`Pckg_SDAResources_coots_multistage_H1` *A Multi-stage RDBESDataObject from package SDAResources dataset coots*

Description

This data set is derived from the data(coots). The design is 2-stage cluster sampling with clusters of unequal sizes and Npsu not known. Clusters are clutches of eggs (nests) with at least 2 eggs. In each cluster, the volume of two eggs is measured. Clusters (psu) are unequal sized. In each cluster, 2 eggs are selected (ssus) and measured. The total number of psus is not known (a drawback in this example). It is assumed very large (fpc negligible). The target variable is volume.

Usage

```
Pckg_SDAResources_coots_multistage_H1
```

Format

A list containing entries required for H1 RDBES data:

- **DE** the Design data table. Contains 1 DE row with DEstratumName == "Pckg_SDAResources_coots_H1"
- **SD** the Sampling Details data table. Contains 1 child SD row
- **VS** the Vessel Selection data table. Contains 184 child rows (the 184 clutches/psus observed), each associated to its cluster (clutch), VSnumberTotal is not known, VSnumberSampled is 184
- **FT** the Fishing Trip data table. Contains 368 child rows (the 368 eggs/ssus measured), each associated to its vessel (clutch), FTnumbersampled is 2, FTnumberTotal is csize
- **FO** the Fishing Operation data table. Just 1:1 links to the final data (in SA)
- **SS** the Species Selection data table. Just 1:1 links to the final data (in SA)
- **SA** the Sample data table. Each volume is a SAsampleWeightMeasured. ATT volumes are *100000000 to meet type requirement (integer)
- **FM** the Frequency Measure data table
- **BV** the Biological Variable data table
- **VD** the Vessel Details data table
- **SL** the Species List data table

Source

https://CRAN.R-project.org/package=SDAResources

`Pckg_SDAResources_gpa_H1`

A RDBESDataObject converted from package SDAResources dataset gpa

Description

This data set is derived from the data(gpa). The design is 1-stage cluster sampling with clusters of equal sizes. Each cluster (suite) has 4 elements with the same weight. The target variable is gpa.

Usage

```
Pckg_SDAResources_gpa_H1
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table. Contains 1 DE row with DEstratumName == "Pckg_SDAResources_gpa_H1"

SD the Sampling Details data table. Contains 1 child SD row

VS the Vessel Selection data table. Contains 20 child rows (the 20 observations), each associated to its cluster (suite), VSnumberTotalClusters is 100, VSnumberTotal is 4 because all elements in cluster are sampled

FT the Fishing Trip data table. Just 1:1 links to the final data (in SA)

FO the Fishing Operation data table. Just 1:1 links to the final data (in SA)

SS the Species Selection data table. Just 1:1 links to the final data (in SA)

SA the Sample data table. Each gpa score is a SAsampleWeightMeasured. ATT gpa scores are *100 to meet type requirement (integer)

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table

Source

https://CRAN.R-project.org/package=SDAResources

Pckg_SDAResources_schools_H1

A RDBESDataObject converted from package SDAResources dataset schools

Description

This data set is derived from the data(schools). The design is 2-stage cluster sampling with clusters of unequal sizes and Npsu not known. Clusters are schools (schoolid). Clusters (psu) are unequal sized (Mi). In each cluster, 20 students are selected (ssus) and measured (nrows). The total number of psus is known (75). The target variable is mathlevel.

Usage

```
Pckg_SDAResources_schools_H1
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table. Contains 1 DE row with DEstratumName == "Pckg_SDAResources_schools_H1"

SD the Sampling Details data table. Contains 1 child SD row

VS the Vessel Selection data table. Contains 200 child rows (the 200 students observed), each associated to its cluster (schoolid), VSnumberTotalClusters is 100, VSnumberTotal is Mi

FT the Fishing Trip data table. Just 1:1 links to the final data (in SA)

FO the Fishing Operation data table. Just 1:1 links to the final data (in SA)

SS the Species Selection data table. Just 1:1 links to the final data (in SA)

SA the Sample data table. Each volume is a SAsampleWeightMeasured

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table

Source

https://CRAN.R-project.org/package=SDAResources

`Pckg_survey_apiclus1_v2_H1`

A RDBESDataObject converted from package survey dataset apiclus1

Description

This data set is derived from the Academic Performance Index computed for all California schools based on standardized testing of students. The original data sets contain information for all schools with at least 100 students and for various probability samples of the data. The design is 1-stage cluster sampling with clusters of unequal sizes. An SRS of 15 districts is selected (psus) from the 757 districts in the population. All schools within district are selected (ssus). The weights (pw) do not match 757/15 probably because they have been calibrated. The target variable is enroll.

Usage

```
Pckg_survey_apiclus1_v2_H1
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table. Contains 1 DE row with DEstratumName == "Pckg_SDAResources_apiclus1_v2_H1"

SD the Sampling Details data table. Contains 1 child SD row

VS the Vessel Selection data table. Contains 183 child rows (the 186 schools finally observed), each associated to its cluster (dname), VSnumberTotalClusters is 757, VSnumberTotal is the number of schools in the cluster (census), calibrated weights are provided as 1/pw in VSinclusionProbCluster

FT the Fishing Trip data table. Just 1:1 links to the final data (in SA)

FO the Fishing Operation data table. Just 1:1 links to the final data (in SA)

SS the Species Selection data table. Just 1:1 links to the final data (in SA)

SA the Sample data table. SAsampleWeightMeasured is enroll

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table

Source

https://CRAN.R-project.org/package=survey

Pckg_survey_apiclus2_H1 *A Multi-Stage RDBESDataObject converted from package survey dataset apiclus2*

Description

This data set is derived from the Academic Performance Index computed for all California schools based on standardized testing of students. The original data sets contain information for all schools with at least 100 students and for various probability samples of the data. The design is 2-stage cluster sampling with clusters of unequal sizes. An SRS of 40 districts is selected (psus) from the 757 districts in the population and then up to 5 schools (min

1. were selected from each district (ssus).

Usage

```
Pckg_survey_apiclus2_H1
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table. Contains 1 DE row with DEstratumName == "Pckg_SDAResources_apiclus2_H1"

SD the Sampling Details data table. Contains 1 child SD row

VS the Vessel Selection data table. Contains 40 child rows (the 40 districts), VSnumberTotal is 757, VSnumberSampled is 40

FT the Fishing Trip data table. Contains 126 child rows (the 126 schools finally observed), each associated to its cluster (dname), FTnumberTotal is the number of schools in district, FTnumberSAmpled is 1...5 schools sampled

FO the Fishing Operation data table. Just 1:1 links to the final data (in SA)

SS the Species Selection data table. Just 1:1 links to the final data (in SA)

SA the Sample data table. SAsampleWeightMeasured is enroll (NB! there are 4 NAs)

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table

Source

`https://CRAN.R-project.org/package=survey`

Pckg_survey_apiclus2_v2_H1 *A Clustered RDBESDataObject converted from package survey dataset apiclus2*

Description

This data set is derived from the Academic Performance Index computed for all California schools based on standardized testing of students. The original data sets contain information for all schools with at least 100 students and for various probability samples of the data. The design is 2-stage cluster sampling with clusters of unequal sizes. An SRS of 40 districts is selected (psus) from the 757 districts in the population and then up to 5 schools (min

1. were selected from each district (ssus). The target variable is enroll - note that it contains 4 NA values.

Usage

```
Pckg_survey_apiclus2_v2_H1
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table. Contains 1 DE row with DEstratumName == "Pckg_SDAResources_apiclus2_v2_H1"

SD the Sampling Details data table. Contains 1 child SD row

VS the Vessel Selection data table. Contains 126 child rows (the 126 schools finally observed), each associated to its cluster (dname), VSnumberTotalClusters is 757, VSnumberTotal is 1...5 schools sampled

FT the Fishing Trip data table. Just 1:1 links to the final data (in SA)

FO the Fishing Operation data table. Just 1:1 links to the final data (in SA)

SS the Species Selection data table. Just 1:1 links to the final data (in SA)

SA the Sample data table. SAsampleWeightMeasured is enroll (note the 4 NAs)

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table

SL the Species List data table

Source

https://CRAN.R-project.org/package=survey

Pckg_survey_apistrat_H1 *A RDBESDataObject converted from package survey dataset apistrat*

Description

This data set is a stratified version of the previous "apiclus2" data. It is derived from the Academic Performance Index computed for all California schools based on standardized testing of students. The original data sets contain information for all schools with at least 100 students and for various probability samples of the data. The design is 1-stage cluster sampling with clusters of unequal sizes. An SRS of 200 districts is selected (psus) from the 755 districts in the population. All schools within district are selected (ssus).

Usage

```
Pckg_survey_apistrat_H1
```

Format

A list containing entries required for H1 RDBES data:

DE the Design data table. Contains 1 DE row

SD the Sampling Details data table. Contains 1 child SD row

VS the Vessel Selection data table. Contains 200 child rows (the 200 schools finally observed), each associated to its cluster (dname), VSnumberTotalClusters is 755, VSnumberTotal is 50-100 schools sampled

FT the Fishing Trip data table. Contains 200 child rows (the 200 schools finally observed), each associated to its cluster (dname), FTnumberTotal is the number of schools in the cluster (census)

FO the Fishing Operation data table. Just 1:1 links to the final data (in SA)

SS the Species Selection data table. Just 1:1 links to the final data (in SA)

SA the Sample data table. SAsampleWeightMeasured is enroll

FM the Frequency Measure data table

BV the Biological Variable data table

VD the Vessel Details data table. Contains 311 child rows

SL the Species List data table. Contains 1 child row

Source

https://CRAN.R-project.org/package=survey

print.RDBESDataObject *Print method for RDBESDataObject*

Description

This method prints the hierarchy of the DE data.table (if it exists), and the number of rows for each data.table in the RDBESDataObject that is not NULL. It also provides the sampling method and number sampled and number total for tables where it is applicable. If the RDBESDataObject has a mixed hierarchy, a warning message is printed.

This method sorts the RDBESDataObject based on the hierarchy.

This method returns a list containing the hierarchy of the DE data.table, the number of rows for each data.table in the RDBESDataObject that is not NULL, and a logical value indicating if the hierarchy is not NULL.

Usage

```
## S3 method for class 'RDBESDataObject'
print(x, ...)

## S3 method for class 'RDBESDataObject'
sort(x, decreasing = TRUE, ...)

## S3 method for class 'RDBESDataObject'
summary(object, ...)
```

Arguments

`x`	An object of class RDBESDataObject.
`...`	parameters to underling functions (not used currently)
`decreasing`	should hierarchy tables be the first ones
`object`	An object of class RDBESDataObject.

Value

None.

The sorted RDBESDataObject by hierarchy.

A list with three elements:

- hierarchy: The hierarchy of the DE data.table in the RDBESDataObject.
- rows: A named list where the names are the names of the data.tables in the RDBESDataObject and the values are the number of rows in each data.table. NULL values are excluded.
- CS: A logical value indicating if the hierarchy is not NULL.

Examples

```
# Print the package data object
print(H1Example)
# Sort the package data
sort(H8ExampleEE1)
# Get summary of the package data
summary(H1Example)
```

`procRDBESEstObjLowHier` *Private function to process the lower hierarchies when creating the RDBESEstObject*

Description

Private function to process the lower hierarchies when creating the RDBESEstObject

Usage

```
procRDBESEstObjLowHier(rdbesPrepObject, verbose = FALSE)
```

Arguments

`rdbesPrepObject`
A prepared RDBESRawObj

`verbose` logical. Output messages to console.

Value

allLower - the FM and BV tables combined

`procRDBESEstObjUppHier` *Private function to process the upper hierarchies when creating the RDBESEstObject*

Description

Private function to process the upper hierarchies when creating the RDBESEstObject

Usage

```
procRDBESEstObjUppHier(
  myRDBESEstObj = NULL,
  rdbesPrepObject,
  hierarchyToUse,
  i = 1,
  targetTables,
  verbose = FALSE
)
```

Arguments

`myRDBESEstObj`	An RDBESEstObj to add data to
`rdbesPrepObject`	A prepared RDBESRawObj
`hierarchyToUse`	The hierarchy we are using
`i`	Integer to keep track of where we are in the list of tables
`targetTables`	The RDBES tables we are interested in
`verbose`	logical. Output messages to console.

Value

Whoever revises this function please specify what it returns here

---

removeBrokenSpeciesListLinks

Remove rows which are not pointing to a valid SpecliestListDetails (SL) records i.e.those rows which have a value of SpeciesListName that does not exist in the SL table.

Description

Remove rows which are not pointing to a valid SpecliestListDetails (SL) records i.e.those rows which have a value of SpeciesListName that does not exist in the SL table.

Usage

```
removeBrokenSpeciesListLinks(objectToCheck, verbose = FALSE, strict = TRUE)
```

Arguments

`objectToCheck`	an RDBESDataObject.
`verbose`	(Optional) If set to TRUE more detailed text will be printed out by the function. Default is TRUE.
`strict`	(Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Value

an RDBESDataObject with any records with an invalid SpeciesListName rows removed

Examples

```
## Not run:

myH1RawObject <-
  importRDBESDataCSV(rdbesExtractPath = "tests\\testthat\\h1_v_1_19")
myFields <- c("SLspeclistName")
myValues <- c("WGRDBES-EST TEST 5 - sprat data")
myFilteredObject <- filterRDBESDataObject(myH1RawObject,
  fieldsToFilter = myFields,
  valuesToFilter = myValues
)
myObjectValidSpeciesListLinks <- removeBrokenSpeciesListLinks(
  objectToCheck = myFilteredObject,
  verbose = FALSE
)

## End(Not run)
```

removeBrokenVesselLinks

Remove rows which are not pointing to a valid VesselDetails (VD) records i.e. those rows which have a value of VDid that does not exist in the VD table.

Description

Remove rows which are not pointing to a valid VesselDetails (VD) records i.e. those rows which have a value of VDid that does not exist in the VD table.

Usage

```
removeBrokenVesselLinks(objectToCheck, verbose = FALSE, strict = TRUE)
```

Arguments

`objectToCheck`	an RDBESDataObject.
`verbose`	(Optional) If set to TRUE more detailed text will be printed out by the function. Default is TRUE.
`strict`	(Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Value

an RDBESDataObject with any records with an invalid VDid removed

Examples

```
## Not run:

myH1RawObject <-
  importRDBESDataCSV(rdbesExtractPath = "tests\\testthat\\h1_v_1_19")
myFields <- c("VDlenCat")
myValues <- c("18-<24")
myFilteredObject <- filterRDBESDataObject(myH1RawObject,
  fieldsToFilter = myFields,
  valuesToFilter = myValues
)
myObjectValidVesselLinks <- removeBrokenVesselLinks(
  objectToCheck = myFilteredObject,
  verbose = FALSE
)

## End(Not run)
```

removePrefixFromVarNames

Remove table prefix from variable names

Description

Remove table prefix from variable names

Usage

```
removePrefixFromVarNames(x)
```

Arguments

x	RDBES raw object

Value

updated RDBES raw object where table prefix has been removed from all variables names except ids

Examples

```
## Not run:
myH1RawObject <-
  importRDBESDataCSV(rdbesExtractPath = "tests\\testthat\\h1_v_1_19")
cleanPrefixFromVarNames(x = myH1RawObject)

## End(Not run)
```

`runChecksOnSelectionAndProbs`
Run basic checks on selection methods and probabilities

Description

This function runs some basic checks on selection methods and and probabilities of the different sampling tables of a hierarchy. It should be run ahead of generateProbs to secure its correct execution and for that reason it is included in the wrapper applyGenerateProbs.

Usage

```
runChecksOnSelectionAndProbs(x, verbose = FALSE, strict = TRUE)
```

Arguments

`x`	• RDBES raw object
`verbose`	• If TRUE prints the issue behind the stop
`strict`	(Optional) This function validates its input data - should the validation be strict? The default is TRUE.

Value

nothing

See Also

`applyGenerateProbs` `generateProbs`

examples for now see https://github.com/ices-eg/WK_RDBES/tree/master/WKRDB-EST2/chairs/Nuno

---

`setRDBESDataObjectDataTypes`
For a given RDBESDataObject convert the required columns to the correct data types. (This function can cause an error if we have data in the columns that can't be cast to the desired data type.)

Description

For a given RDBESDataObject convert the required columns to the correct data types. (This function can cause an error if we have data in the columns that can't be cast to the desired data type.)

Usage

```
setRDBESDataObjectDataTypes(RDBESDataObjectToConvert)
```

Arguments

`RDBESDataObjectToConvert`
list - the raw item for conversion

Value

An RDBESDataObject with the correct date types for the required fields

---

`shrimps` *One quarter of sample data from swedish shrimp landings of the SWE_OTB_CRU_32-69_0_0 fishery*

Description

A dataset of rdbesEstimObj type containing simplified haul-level samples (rows) of shrimp landings (targetValue, in kg) observed onboard using H1 of RDBES with UPWOR on vessels. Data is provided for developing/testing purposes only.

Usage

```
shrimps
```

Format

A data frame with 10 rows and 95 variables:

- DEsamplingScheme - Sampling Scheme
- DEyear - Year of data collection
- DEstratumName - Fishery code
- DEhierarchyCorrect - Design Variable of RDBES. More details in RDBES documentation
- DEhierarchy - Design Variable of RDBES. More details in RDBES documentation
- DEsampled - Design Variable of RDBES. More details in RDBES documentation
- DEreasonNotSampled - Design Variable of RDBES. More details in RDBES documentation
- SDcountry - Country that collected the data
- SDinstitution - Institution that collected the data
- su1, su2, su3, su4, su5 - sampling units of RDBES. More details in RDBES documentation
- XXXnumberSampled, ... - Design Variables of RDBES. More details in RDBES documentation
- targetValue - estimate of weight landed in each haul (in kg)
- plus XX other columns

Source

Nuno Prista @ SLU Aqua, 2022

`shrimpsStrat` *One quarter of sample data from swedish shrimp catches of the SWE_OTB_CRU_32-69_0_0 fishery*

Description

A dataset of rdbesEstimObj type containing simplified haul-level samples (rows) of shrimp catches (targetValue, in kg) observed onboard using H1 of RDBES with UPWOR on vessels. Catches are divided into three strata (91, 92, 93_94) that correspond to sorting sieves used onboard. Data is provided for developing/testing purposes only.

Usage

```
shrimpsStrat
```

Format

A data frame with 10 rows and 95 variables:

- DEsamplingScheme - Sampling Scheme
- DEyear - Year of data collection
- DEstratumName - Fishery code
- DEhierarchyCorrect - Design Variable of RDBES. More details in RDBES documentation
- DEhierarchy - Design Variable of RDBES. More details in RDBES documentation
- DEsampled - Design Variable of RDBES. More details in RDBES documentation
- DEreasonNotSampled - Design Variable of RDBES. More details in RDBES documentation
- SDcountry - Country that collected the data
- SDinstitution - Institution that collected the data
- su1, su2, su3, su4, su5 - sampling units of RDBES. More details in RDBES documentation
- XXXnumberSampled, ... - Design Variables of RDBES. More details in RDBES documentation
- su5stratumName - sieve fraction
- targetValue - estimate of weight fraction in each haul (in kg)
- plus XX other columns

Source

Nuno Prista @ SLU Aqua, 2022

tablesInRDBESHierarchies

The tables required for each RDBES hierarchy.

Description

A data frame containing the tables required for each RDBES hierachy

Usage

```
tablesInRDBESHierarchies
```

Format

A data frame containing the tables required for each RDBES hierachy.

hierarchy the hierachy this applies to H1 to H13

table the 2-letter table name

lowerHierarchy is this a lower hierarchy table?

optional is this table optional within the hierarchy?

samplingUnit is this table a sampling unit within the hierarchy?

sortOrder the table sort order within the hiaerarchy

Source

https://github.com/davidcurrie2001/MI_RDBES_ExchangeFiles

validateRDBESDataObject

Check Whether an `RDBESDataObject` *is in a Valid Format*

Description

Perform basic checks on a object.

Usage

```
validateRDBESDataObject(
  objectToCheck,
  checkDataTypes = FALSE,
  verbose = FALSE,
  strict = TRUE
)
```

```
checkRDBESDataObject(
  objectToCheck,
  checkDataTypes = FALSE,
  verbose = FALSE,
  strict = TRUE
)
```

Arguments

`objectToCheck` RDBESDataObject i.e. a list of data.tables

`checkDataTypes` (Optional) Set to TRUE if you want to check that the data types of the required columns are correct, or FALSE if you don't care. Default value is FALSE.

`verbose` (Optional) Set to TRUE if you want informative text printed out, or FALSE if you don't. The default is FALSE.

`strict` (Optional) Set to TRUE if you want to be sure all columns are present in the data, set to FALSE if you only want to check that essential columns are present. The default is TRUE.

Details

Checks if 'objectToCheck' parameter is valid. Returns the parameter if it is valid and otherwise stops on error. It checks the `RDBESDataObject` if:

- Is this an object of class RDBESDataObject
- Tables don't have column names that aren't allowed
- Tables have all the required column names

It does not check if the data is valid. The RDBES upload system performs an extensive set of checks on the uploaded data.

Value

Returns objectToCheck

Examples

```
## Not run:
myH1RawObject <-
importRDBESDataCSV(rdbesExtractPath = ”tests/testthat/h1_v_1_19”)
validateRDBESDataObject(myH1RawObject)
## End(Not run)
```

validateRDBESDataObjectDataTypes

Checks the data types of the columns in an RDBESDataObject against an expected list of data types. Any differences are returned

Description

Checks the data types of the columns in an RDBESDataObject against an expected list of data types. Any differences are returned

Usage

```
validateRDBESDataObjectDataTypes(objectToCheck)
```

Arguments

`objectToCheck`	An RDBESDataObject to check

Value

A data frame containing any data type differences (an empty data frame if there are no differences)

validateRDBESDataObjectDuplicates

check RDBES Raw Object Content Private function to do some basic checks on the content of the RDBESDataObject (e.g. all required field names are present). Function is only used by checkRDBESDataObject and should only be passed a list of non-null objects

Description

check RDBES Raw Object Content Private function to do some basic checks on the content of the RDBESDataObject (e.g. all required field names are present). Function is only used by checkRDBESDataObject and should only be passed a list of non-null objects

Usage

```
validateRDBESDataObjectDuplicates(
  objectToCheck,
  verbose = FALSE,
  strict = TRUE
)
```

Arguments

`objectToCheck`
- RDBESDataObject i.e. a list of data.tables

`verbose` (Optional) Set to TRUE if you want informative text printed out, or FALSE if you don't. The default is FALSE.

`strict` (Optional) Set to TRUE if you want to be sure all columns are present in the data, set to FALSE if you only want to check that essential columns are present. The default is TRUE.

Value

list with first element as the object and the second the warnings

---

`validateRDBESDataObjectFieldNames`

check RDBES Data Object field names Private function to do some checks on the columns of an RDBESDataObject -

1. *are all required fields present? 2) are there any extra fields present? It is used by validateRDBESDataObject() and should only be passed a list of non-null objects*

Description

check RDBES Data Object field names Private function to do some checks on the columns of an RDBESDataObject -

1. are all required fields present? 2) are there any extra fields present? It is used by validateRDBESDataObject() and should only be passed a list of non-null objects

Usage

```
validateRDBESDataObjectFieldNames(
  objectToCheck,
  verbose = FALSE,
  strict = TRUE
)
```

Arguments

`objectToCheck`
- RDBESDataObject i.e. a list of data.tables

`verbose` (Optional) Set to TRUE if you want informative text printed out, or FALSE if you don't. The default is FALSE.

`strict` (Optional) Set to TRUE if you want to be sure all columns are present in the data, set to FALSE if you only want to check that essential columns are present. The default is TRUE.

Value

list with first element as a boolean indicating validity and the second element contains any warnings

validateRDBESEstObject

Check whether an object is a valid RDBESEstObject

Description

Check whether an object is a valid RDBESEstObject

Usage

```
validateRDBESEstObject(objectToCheck, verbose = FALSE)
```

Arguments

objectToCheck	The object to check
verbose	(Optional) Set to TRUE if you want informative text printed out, or FALSE if you don't. The default is FALSE.

Value

Whoever revises this function please specify what it returns here

Examples

```
## Not run:
myH1RawObject <-
importRDBESDataCSV(rdbesExtractPath = "tests/testthat/h1_v_1_19")
myEStObj <- createRDBESEstObject(myH1RawObject,1)
validateRDBESEstObject(myEStObj)
## End(Not run)
```

wormsAphiaRecord

A dataset containing aphia records for species found in icesSpecWoRMS

Description

A dataset containing aphia records for species found in icesSpecWoRMS

Usage

```
wormsAphiaRecord
```

Format

A data frame

AphiaID E.g. 100684

url E.g. "https://www.marinespecies.org/aphia.php?p=taxdetails&id=100684"

scientificname E.g. "Cerianthidae"

authority E.g. "Milne Edwards & Haime, 1851"

status E.g. "accepted"

unacceptreason E.g. NA

taxonRankID E.g. 140

rank E.g. "Family" "Genus" "Species" "Species"

valid_AphiaID E.g. 100684

valid_name E.g. "Cerianthidae"

valid_authority E.g. "Milne Edwards & Haime, 1851"

parentNameUsageID E.g. 151646

kingdom E.g. "Animalia"

phylum E.g. "Cnidaria"

class E.g. "Anthozoa"

order E.g. "Spirularia"

family E.g. "Cerianthidae"

genus E.g. NA "Cerianthus"

citation E.g. "Molodtsova, T. (2023). World List of Ceriantharia. Cerianthidae Milne Edwards & Haime, 1851. Accessed through: "...

lsid internal database identifier

isMarine E.g. 1

isBrackish E.g. 1

isFreshwater E.g. 0

isTerrestrial E.g. 0

isExtinct E.g. NA

match_type E.g. "exact"

modified E.g. "2018-01-22T17:48:34.063Z"

DateDownloaded E.g. "2023-10-18" ...

Source

`https://www.marinespecies.org/`

Index