

Package ‘COSTcore’

July 31, 2009

Title Core package of COST.

Version 1.3-5

Date 2009

Author COST Team and various contributors.

Description COSTcore contains the core classes and methods for COST, a framework for analysis of fisheries sampling data.

Depends R(>= 2.8.1), methods

Maintainer Ernesto Jardim <ernesto@ipimar.pt>

License GPL 2 or above

Collate utils.R check-methods.R check.fields.R ceData-class.R ceDataVal-class.R
ceDataCons-class.R clData-class.R clDataVal-class.R clDataCons-class.R csData-class.R
csDataVal-class.R csDataCons-class.R FLtest.R zzz.R

LazyLoad Yes

R topics documented:

accessor-methods	1
ceData-class	2
ceDataCons-class	3
ceDataVal-class	4
check-methods	6
check.fields	6
checkDataIntegrity-methods	7
clData-class	7
clDataCons-class	9
clDataVal-class	10
code.list	11
coerceCons-methods	12
coerceDataFrameColumns-methods	12
csData-class	12
csDataCons-class	16
csDataVal-class	19
numeric.list	22

rbind2-methods	23
sole.ce	23
sole.cl	24
sole.cs	24
strIni-class	25
subset-methods	26
subsetSpp	27
Tests	27
utils-methods	29
variable.list	29

accessor-methods

Accessor methods in COSTcore

Description

These accessor methods can be used to obtain specific slots from a COST object and are designed to implement "encapsulation" within S4 classes. Users are strongly advised to apply these functions when accessing their data to ensure compatibility with any future changes to the class definition.

Methods

tr signature(object = csData or csDataVal or csDataCons): tr slot accessor.
hh signature(object = csData or csDataVal or csDataCons): hh slot accessor.
sl signature(object = csData or csDataVal or csDataCons): sl slot accessor.
hl signature(object = csData or csDataVal or csDataCons): hl slot accessor.
ca signature(object = csData or csDataVal or csDataCons): ca slot accessor.
ce signature(object = ceData or ceDataVal or ceDataCons): ce slot accessor.
cl signature(object = clData or clDataVal or clDataCons): cl slot accessor.
desc signature(object = csData or csDataVal or csDataCons): description slot accessor.

Examples

```
obj <- csData()
tr(obj)
hh(obj)
sl(obj)
hl(obj)
ca(obj)
obj <- ceData()
ce(obj)
obj <- clData()
cl(obj)
```

ceData-class	Class "ceData"
--------------	----------------

Description

The ceData class stores commercial fisheries effort data and maps the format of the CE tables in Fish Frame version 5.

Objects from the Class

The creator function "ceData" can be called to create objects from this class.

Slots

slot	columns	class	description
desc		character	object description
ce		data.frame	effort information
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	quarter	numeric	quarter
	month	numeric	month
	area	character	area
	rect	character	statistical rectangle
	subRect	character	statistical subrectangle
	foCatNat	character	fishing operations category, national
	foCatEu5	character	fishing operations category, european level 5
	foCatEu6	character	fishing operations category, european level 6
	harbour	character	landing harbour
	vslLenCat	character	vessel length categories
	trpNum	numeric	number of trips
	foNum	numeric	number of fishing operations
	foDur	numeric	duration of fishing operations
	effKwDays	numeric	effort in kw days
	effGtDays	numeric	effort in T days
	daysAtSea	numeric	effort in days at sea

Methods

ce signature(object = "ceData"): ce slot accessor.

ceDataVal signature(object = "ceData"): creator of Validated data class.

desc signature(object = "ceData"): desc slot accessor.

dim signature(x = "ceData"): dimensions of ce slot (see dim).

head signature(x = "ceData"): first rows of the ce slot (see head).

rbind2 signature(x = "ceData", y = "ceData"): bind two ceData objects by adding rows (see rbind2).

subset signature(x = "ceData"): subset a ceData object (see subset).

summary signature(object = "ceData"): summary a ceData object (see summary).

tail signature(x = "ceData"): last rows of the ce slot (see tail).

Author(s)

Ernesto Jardim <ernesto@ipimar.pt>

Examples

```
showClass("ceData")
```

ceDataCons-class *Class "ceDataCons"*

Description

The ceDataCons-class is similar to ceDataVal but stores consolidated commercial effort data, i.e. that which is aggregated over spatial, temporal and technical (e.g. gear type) strata. The expectation is that this post-stratified data will be the basis for the application of statistical methods.

Objects from the Class

The creator function "ceDataCons" can be called to create objects from this class.

Slots

slot	columns	class	description
desc		character	object description
ce		data.frame	effort information
	vslFlgCtry	factor	vessel flag country
	time	factor	time strata
	space	factor	space strata
	technical	factor	technical strata
	trpNum	numeric	number of trips
	foNum	numeric	number of fishing operations
	foDur	numeric	duration of fishing operations
	effKwDays	numeric	effort in kw days
	effGtDays	numeric	effort in T days
	daysAtSea	numeric	effort in days at sea

Methods

ce signature(object = "ceDataCons"): ce slot accessor.

desc signature(object = "ceDataCons"): desc slot accessor.

dim signature(x = "ceDataCons"): dimensions of ce slot (see dim).

head signature(x = "ceDataCons"): first rows of the ce slot (see head).

rbind2 signature(x = "ceDataCons", y = "ceDataCons"): bind two ceDataCons objects by adding rows (see rbind2).

subset signature(x = "ceDataCons"): subset a ceDataCons object (see subset).

summary signature(object = "ceDataCons"): summary a ceDataCons object (see summary).

tail signature(x = "ceData"): last rows of the ce slot (see tail).

Author(s)

Ernesto Jardim <ernesto@ipimar.pt>

Examples

```
showClass("ceDataCons")
```

ceDataVal-class	<i>Class "ceDataVal"</i>
-----------------	--------------------------

Description

The ceDataVal-class is equivalent in structure to ceData but stores a validated form of the commercial effort data contained in the ceDataclass. The validation process consists of checking the compatibility of entries of variables of type character against the permissible code lists, the checking of numeric variable types, the checking of numeric variable ranges, and the identification of outliers. Data of "ceDataVal" is considered suitable for analysis and is additionally a prerequisite to aggregation over strata to form data of class *ceDataCons*.

Objects from the Class

The creator function "ceDataVal" can be called to create objects from this class.

Slots

slot	columns	class	description
desc		character	object description
ce		data.frame	effort information
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	quarter	numeric	quarter
	month	numeric	month
	area	character	area
	rect	character	statistical rectangle
	subRect	character	statistical subrectangle
	foCatNat	character	fishing operations category, national
	foCatEu5	character	fishing operations category, european level 5
	foCatEu6	character	fishing operations category, european level 6
	harbour	character	landing harbour
	vslLenCat	character	vessel length categories
	trpNum	numeric	number of trips
	foNum	numeric	number of fishing operations
	foDur	numeric	duration of fishing operations
	effKwDays	numeric	effort in kw days
	effGtDays	numeric	effort in T days
	daysAtSea	numeric	effort in days at sea

Methods

ce signature(object = "ceDataVal"): ce slot accessor.

ceDataCons signature(object = "ceDataVal"): creator of Consolidated data class.

desc signature(object = "ceDataVal"): desc slot accessor.

dim signature(x = "ceDataVal"): dimensions of ce slot (see dim).

head signature(x = "ceDataVal"): first rows of the ce slot (see head).

rbind2 signature(x = "ceDataVal", y = "ceDataVal"): bind two ceDataVal objects by adding rows (see rbind2).

subset signature(x = "ceDataVal"): subset a ceDataVal object (see subset).

summary signature(object = "ceDataVal"): summary a ceDataVal object (see summary).

tail signature(x = "ceDataVal"): last rows of the ce slot (see tail).

Author(s)

Ernesto Jardim <ernesto@ipimar.pt>

Examples

```
showClass("ceDataVal")
```

check-methods

Check methods in COSTcore

Description

Check methods are used to validate S4 classes. These functions are for internal usage only and are not intended to be called directly by users.

Methods

checkNms signature(object, names): Checks the names in a data.frame against a character vector.

checkTys signature(object, tys): Checks the column types in a data.frame against a list of class names.

checkTRnms signature(object): Wrapper for TR checkNms.

checkHHnms signature(object): Wrapper for HH checkNms.

checkSLnms signature(object): Wrapper for SL checkNms.

checkHLnms signature(object): Wrapper for HL checkNms.

checkCANms signature(object): Wrapper for CA checkNms.

checkCLnms signature(object): Wrapper for CL checkNms.

checkCEnms signature(object): Wrapper for CE checkNms.

checkTRpk signature(object): Checks the TR primary key.

checkHHpk signature(object): Checks the HH primary key.

checkSLpk signature(object): Checks the SL primary key.

checkHLpk signature(object): Checks the HL primary key.

checkCLpk signature(object): Checks the CL primary key.

checkCEpk signature(object): Checks the CE primary key.

check.fields

Checks fields of csData clData and ceData COST objects

Description

This function is designed to perform variable checks on the fields of csData, ceData and clData COST objects. It compares character variables against those specified in *code.list* and the range of numeric values against those specified in *numeric.list*. Error messages specifying any discrepancies are printed to the screen and optionally to a log file.

Usage

```
check.fields(costobj,logfile=FALSE)
```

Arguments

costobj	An object of class csData, ceData or clData
logfile	Logical, if TRUE creates a logfile in the current working directory, the default is FALSE.

Value

Prints to the screen a summary of the entries in each field within the component data frames. Optionally prints a log file to the current working directory.

Author(s)

Alastair Pout <a.pout@marlab.ac.uk>

See Also

code.list for the expected character variables, *numeric.list* for the ranges of numeric variables, and *variable.list* for details of variable names, types and a brief description of each.

Examples

```
data(sole)
check.fields(sole.cs)
check.fields(sole.cs,logfile=TRUE)
```

checkDataIntegrity-methods

Methods for Function checkDataIntegrity in Package 'COST-core'

Description

Methods to check that all data in the "current" data frame have related records in the "target" data frame. These functions are for data integrity checking when validating csData objects and are not intended to be called directly by users.

Methods

checkDataIntegrity signature(target = "data.frame", current = "data.frame"):
Checks that data in the "current" data frame have related records in the "target" data frame.

clData-class

Class "clData"

Description

The clData class stores commercial fisheries landings data and maps the format of the CE tables in Fish Frame version 5.

Objects from the Class

The creator function "clData" can be called to create objects from this class.

Slots

slot	columns	class	description
desc		character	object description
cl		data.frame	landings information
	landCtry	character	landing country
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	quarter	numeric	quarter
	month	numeric	month
	area	character	area
	rect	character	statistical rectangle
	subRect	character	statistical subrectangle
	taxon	character	scientific name
	landCat	character	landing category
	commCatScl	character	commercial category scale
	commCat	character	commercial category
	foCatNat	character	fishing operations category, national
	foCatEu5	character	fishing operations category, european level 5
	foCatEu6	character	fishing operations category, european level 6
	harbour	character	landing harbour

vslLenCat	character	vessel length categories
unallocCatchWt	numeric	unallocated catch weight
misRepCatchWt	numeric	area misreported catch weight
landWt	numeric	official landings weight
landMult	numeric	landings multiplier
landValue	numeric	official landings value

Methods

cl signature(object = "clData"): cl slot accessor.

clDataVal signature(object = "clData"): creator of Validated data class.

desc signature(object = "clData"): desc slot accessor.

dim signature(x = "clData"): dimensions of cl slot (see **dim**).

head signature(x = "clData"): first rows of the cl slot (see **head**).

rbind2 signature(x = "clData", y = "clData"): bind two clData objects by adding rows (see **rbind2**).

subset signature(x = "clData"): subset a clData object (see **subset**).

summary signature(object = "clData"): summary a clData object (see **summary**).

tail signature(x = "clData"): last rows of the cl slot (see **tail**).

Author(s)

Ernesto Jardim <ernesto@ipimar.pt>

Examples

```
showClass("clData")
```

clDataCons-class	<i>Class "clDataCons"</i>
------------------	---------------------------

Description

The clDataCons-class is similar to clDataVal but stores consolidated commercial landings data, i.e. that which is aggregated over spatial, temporal and technical (e.g. gear type) strata. The expectation is that this post-stratified data will be the basis for the application of statistical methods.

Objects from the Class

The creator function "clDataCons" can be called to create objects from this class.

Slots

slot	columns	class	description
desc		character	object description
c1		data.frame	landings information
	landCtry	factor	landing country
	vslFlgCtry	factor	vessel flag country
	time	numeric	time strata
	space	numeric	space strata
	technocal	numeric	technical strata
	spp	factor	species
	landCat	factor	landing category
	commCatSel	factor	commercial category scale
	commCat	factor	commercial category
	unallocCatchWt	numeric	unallocated catch weight
	misRepCatchWt	numeric	area misreported catch weight
	landWt	numeric	official landings weight
	landMult	numeric	landings multiplier
	landValue	numeric	official landings value

Methods

cl signature(object = "clData"): cl slot accessor.

desc signature(object = "clDataCons"): desc slot accessor.

dim signature(x = "clDataCons"): dimensions of c1 slot (see dim).

head signature(x = "clDataCons"): first rows of the c1 slot (see head).

rbind2 signature(x = "clDataCons", y = "clDataCons"): bind two clDataCons objects by adding rows (see rbind2).

subset signature(x = "clDataCons"): subset a clDataCons object (see subset).

summary signature(object = "clDataCons"): summary a clDataCons object (see summary).

tail signature(x = "clDataCons"): last rows of the c1 slot (see tail).

Author(s)

Ernesto Jardim <ernesto@ipimar.pt>

Examples

```
showClass("clDataCons")
```

clDataVal-class *Class "clDataVal"*

Description

The clDataVal-class is equivalent in structure to clData but stores a validated form of the commercial landings data contained in the clDataclass. The validation process consists of checking the compatibility of entries of variables of type character against the permissible code lists, the checking of numeric variable types, the checking of numeric variable ranges, and the identification of outliers. Data of "clDataVal" is considered suitable for analysis and is additionally a prerequisite to aggregation over strata to form data of class "clDataCons".

Objects from the Class

The creator function "clDataVal" can be called to create objects from this class.

Slots

slot	columns	class	description
desc		character	object description
cl		data.frame	landings information
	landCtry	character	landing country
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	quarter	numeric	quarter
	month	numeric	month
	area	character	area
	rect	character	statistical rectangle
	subRect	character	statistical subrectangle
	taxon	character	scientific name
	landCat	character	landing category
	commCatScl	character	commercial category scale
	commCat	character	commercial category
	foCatNat	character	fishing operations category, national
	foCatEu5	character	fishing operations category, european level 5
	foCatEu6	character	fishing operations category, european level 6
	harbour	character	landing harbour
	vslLenCat	character	vessel length categories
	unallocCatchWt	numeric	unallocated catch weight
	misRepCatchWt	numeric	area misreported catch weight
	landWt	numeric	official landings weight
	landMult	numeric	landings multiplier
	landValue	numeric	official landings value

Methods

cl signature(object = "clData"): cl slot accessor.

clDataCons signature(object = "clDataVal"): creator of Consolidated data class.

desc signature(object = "clDataVal"): desc slot accessor.
dim signature(x = "clDataVal"): dimensions of cl slot (see dim).
head signature(x = "clDataVal"): first rows of the cl slot (see head).
rbind2 signature(x = "clDataVal", y = "clDataVal"): bind two clDataVal objects by adding rows (see rbind2).
subset signature(x = "clDataVal"): subset a clDataVal object (see subset).
summary signature(object = "clDataVal"): summary a clDataVal object (see summary).
tail signature(x = "clDataVal"): last rows of the cl slot (see tail).

Author(s)

Ernesto Jardim <ernesto@ipimar.pt>

Examples

```
showClass("clDataVal")
```

code.list

Code lists of expected values for COST variables

Description

Code lists of expected values for COST character string variables.

Usage

```
data(code.list)
```

Format

A list of 28 variables each a data frame with (usually) \$code and \$description and possibly other information.

Source

FishFrame 5.0 Exchange Format Specification 25th January 2008.

Examples

```
data(code.list)
# cost variables for which there are code lists i.e. all the character fields
names(code.list)
# the (start of the) expected species list
head(code.list$spp)
# (Start of the) list of ICES statistical rectangles
head(code.list$rect)
```

coerceCons-methods *Coerces df columns into specific classes.*

Description

Specific 'consolidated' version of 'coerceDataframeColumns' method. These functions are for internal usage only and are not intended to be called directly by users.

Methods

coerceCons signature(object = "data.frame", refObject = "data.frame"): Checks the class of refObject columns and coerces the object columns in agreement.

coerceDataframeColumns-methods

Coerces data.frame columns into specific classes.

Description

Methods for the function *coerceDataframeColumns*. These functions are for internal usage only and are not intended to be called directly by users.

Methods

coerceDataframeColumns signature(object = "data.frame", refObject = "data.frame"): Checks the class of refObject columns and coerces the object columns in agreement.

csData-class

Class "csData"

Description

The csData class stores commercial fisheries sampling data and maps the CS tables in Fish Frame version 5.

Objects from the Class

The creator function "csData" can be called to create objects from this class.

Slots

slot	columns	class	description
desc		character	object description
tr		data.frame	trip information
	sampType	character	sampling type
	landCtry	character	landing country
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	proj	character	project
	trpCode	character	trip code
	vslLen	numeric	vessel length-over-all
	vslPwr	numeric	vessel power (kw)
	vslSize	numeric	vessel size (GRT)
	vslType	character	vessel type
	harbour	character	landing harbour
	foNum	numeric	number of fishing operations
	daysAtSea	numeric	effort in days at sea
	vslId	numeric	vessel identifier
	sampCtry	character	sampling country
	sampMeth	character	sampling method
hh		data.frame	haul information
	sampType	character	sampling type
	landCtry	character	landing country
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	proj	character	project
	trpCode	character	trip code
	staNum	numeric	station number
	foVal	character	fishing operation validity
	aggLev	character	aggregation level
	catReg	character	parts (landings/discards) of the catch registered
	sppReg	character	species of the catch registered
	date	character	date
	time	character	time
	foDur	numeric	fishing operation duration
	latIni	numeric	latitude of shooting position in decimal degrees
	lonIni	numeric	longitude of shooting position in decimal degrees
	latFin	numeric	latitude of hauling position in decimal degrees
	lonFin	numeric	longitude of hauling position in decimal degrees
	area	character	area
	rect	character	statistical rectangle
	subRect	character	statistical subrectangle
	foDep	numeric	fishing operation depth
	waterDep	numeric	water depth
	foCatNat	character	fishing operations category, national
	foCatEu5	character	fishing operations category, european level 5
	foCatEu6	character	fishing operations category, european level 6
	meshSize	numeric	mesh size
	selDev	character	selectivity device
	meshSizeSelDev	numeric	mesh size in selectivity device

s1		data.frame	species list
	sampType	character	sampling type
	landCtry	character	landing country
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	proj	character	project
	trpCode	character	trip code
	staNum	numeric	station number
	spp	character	species
	catchCat	character	catch category
	landCat	character	landing category
	commCatSel	character	commercial category scale
	commCat	character	commercial category
	subSampCat	character	subsampling category
	sex	character	gender
	wt	numeric	weight
	subSampWt	numeric	subsample weight
	lenCode	character	length code
h1		data.frame	length frequencies
	sampType	character	sampling type
	landCtry	character	landing country
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	proj	character	project
	trpCode	character	trip code
	staNum	numeric	station number
	spp	character	species
	catchCat	character	catch category
	landCat	character	landing category
	commCatSel	character	commercial category scale
	commCat	character	commercial category
	subSampCat	character	subsampling category
	sex	character	gender
	lenCls	numeric	length class
	lenNum	numeric	frequency at length
ca		data.frame	growth and maturity
	sampType	character	sampling type
	landCtry	character	landing country
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	proj	character	project
	trpCode	character	trip code
	staNum	numeric	station number
	quarter	numeric	quarter
	month	numeric	month
	spp	character	species
	sex	character	gender
	catchCat	character	catch category
	landCat	character	landing category

commCatScl	character	commercial category scale
commCat	character	commercial category
stock	character	stock
area	character	area
rect	character	statistical rectangle
subRect	character	statistical subrectangle
lenCls	numeric	length class
age	numeric	age
fishId	numeric	individual identification
lenCode	character	length code
ageMeth	character	methodology for estimating the age
plusGrp	character	plus group
otoWt	numeric	otolith weight
otoSide	character	otolith side
indWt	numeric	individual weight
matMeth	character	methodology for estimating the maturity stage
matScale	character	maturity scale
matStage	character	maturity stage

Methods

ca signature(object = "csData"): ca slot accessor.

csDataVal signature(object = "csData"): creator of Validated data class.

desc signature(object = "csData"): desc slot accessor.

dim signature(x = "csData"): dimensions of slots (see **dim**).

head signature(x = "csData"): first rows of the each slot (see **head**).

hh signature(object = "csData"): hh slot accessor.

hl signature(object = "csData"): hl slot accessor.

rbind2 signature(x = "csData", y = "csData"): bind two csData objects by adding rows (see **rbind2**).

sl signature(object = "csData"): sl slot accessor.

subset signature(x = "csData"): subset a ceData object (see **subset**).

summary signature(object = "csData"): summary a ceData object (see **summary**).

tail signature(x = "csData"): last rows of each slot (see **tail**).

tr signature(object = "csData"): tr slot accessor.

Author(s)

Ernesto Jardim <ernesto@ipimar.pt>

Examples

```
showClass("csData")
```

csDataCons-class *Class "csDataCons"*

Description

The csDataCons-class is similar to csDataVal but stores consolidated commercial sampling data, i.e. that which is aggregated over spatial, temporal and technical (e.g. gear type) strata. The expectation is that this post-stratified data will be the basis for the application of statistical methods.

Objects from the Class

The creator function "csDataCons" can be called to create objects from this class.

Slots

slot	columns	class	description
desc		character	object description
tr		data.frame	trip information
	PSUid	numeric	primary sampling unit id
	time	factor	time strata
	space	factor	space strata
	technical	factor	technical strata
	sampType	factor	sampling type
	landCtry	factor	landing country
	vslFlgCtry	factor	vessel flag country
	proj	factor	project
	trpCode	factor	trip code
	foNum	numeric	number of fishing operations
	daysAtSea	numeric	effort in days at sea
	vslId	numeric	vessel identifier
	sampCtry	factor	sampling country
	sampMeth	factor	sampling method
hh		data.frame	haul information
	PSUid	numeric	primary sampling unit id
	SSUid	numeric	secondary sampling unit id
	time	factor	time strata
	space	factor	space strata
	technical	factor	technical strata
	sampType	factor	sampling type
	landCtry	factor	landing country
	vslFlgCtry	factor	vessel flag country
	proj	factor	project
	trpCode	factor	trip code
	staNum	numeric	station number
	foVal	factor	fishing operation validity
	catReg	character	parts (landings/discards) of the catch registered
	sppReg	character	species of the catch registered
	aggLev	factor	aggregation level

	foDur	numeric	fishing operation duration
	latIni	numeric	latitude of shooting position in decimal degrees
	lonIni	numeric	longitude of shooting position in decimal degrees
	latFin	numeric	latitude of hauling position in decimal degrees
	lonFin	numeric	longitude of hauling position in decimal degrees
	foDep	numeric	fishing operation depth
s1		data.frame	species list
	PSUId	numeric	primary sampling unit id
	SSUId	numeric	secondary sampling unit id
	TSUId	numeric	terciary sampling unit id
	time	factor	time strata
	space	factor	space strata
	technical	factor	technical strata
	sort	factor	sorting strata
	sampType	factor	sampling type
	landCtry	factor	landing country
	vslFlgCtry	factor	vessel flag country
	proj	factor	project
	trpCode	factor	trip code
	staNum	numeric	station number
	spp	factor	species
	valCode	factor	validity code
	wt	numeric	weight
	subSampWt	numeric	subsample weight
	lenCode	factor	length code
hl		data.frame	length frequencies
	PSUId	numeric	primary sampling unit id
	SSUId	numeric	secondary sampling unit id
	TSUId	numeric	terciary sampling unit id
	time	factor	time strata
	space	factor	space strata
	technical	factor	technical strata
	sort	factor	sorting strata
	sampType	factor	sampling type
	landCtry	factor	landing country
	vslFlgCtry	factor	vessel flag country
	proj	factor	project
	trpCode	factor	trip code
	staNum	numeric	station number
	spp	factor	species
	sex	factor	gender
	lenCls	numeric	length class
	lenNum	numeric	frequency at length
ca		data.frame	growth and maturity
	PSUId	numeric	primary sampling unit id
	SSUId	numeric	secondary sampling unit id
	time	factor	time strata
	space	factor	space strata
	technical	factor	technical strata

sort	factor	sorting strata
sampType	factor	sampling type
landCtry	factor	landing country
vslFlgCtry	factor	vessel flag country
proj	factor	project
trpCode	factor	trip code
staNum	numeric	station number
spp	factor	species
sex	factor	gender
stock	factor	stock
lenCls	numeric	length class
age	numeric	age
fishId	numeric	individual identification
lenCode	factor	length code
plusGrp	factor	plus group
otoWt	numeric	otolith weight
otoSide	factor	otolith side
indWt	numeric	individual weight
matScale	factor	maturity scale
matStage	factor	maturity stage

Methods

ca signature(object = "csDataCons"): ca slot accessor.

desc signature(object = "csDataCons"): desc slot accessor.

dim signature(x = "csDataCons"): dimensions of slots (see **dim**).

head signature(x = "csDataCons"): first rows of the each slot (see **head**).

hh signature(object = "csDataCons"): hh slot accessor.

hl signature(object = "csDataCons"): hl slot accessor.

rbind2 signature(x = "csDataCons", y = "csDataCons"): bind two csDataCons objects by adding rows (see **rbind2**).

sl signature(object = "csDataCons"): sl slot accessor.

subset signature(x = "csDataCons"): subset a ceData object (see **subset**).

summary signature(object = "csDataCons"): summary a ceData object (see **summary**).

tail signature(x = "csDataCons"): last rows of each slot (see **tail**).

tr signature(object = "csDataCons"): tr slot accessor.

Author(s)

Ernesto Jardim <ernesto@ipimar.pt>

Examples

```
showClass("csDataCons")
```

csDataVal-class *Class "csDataVal"*

Description

The csDataVal-class is equivalent in structure to csData but stores a validated form of the commercial landings data contained in the csDataclass. The validation process consists of checking the compatibility of entries of variables of type character against the permissible code lists, the checking of numeric variable types, the checking of numeric variable ranges, and the identification of outliers. Data of "csDataVal" is considered suitable for analysis and is additionally a prerequisite to aggregation over strata to form data of class "csDataCons".

Objects from the Class

The creator function "csDataVal" can be called to create objects from this class.

Slots

slot	columns	class	description
desc		character	object description
tr		data.frame	trip information
	sampType	character	sampling type
	landCtry	character	landing country
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	proj	character	project
	trpCode	character	trip code
	vslLen	numeric	vessel length-over-all
	vslPwr	numeric	vessel power (kw)
	vslSize	numeric	vessel size (GRT)
	vslType	character	vessel type
	harbour	character	landing harbour
	foNum	numeric	number of fishing operations
	daysAtSea	numeric	effort in days at sea
	vslId	numeric	vessel identifier
	sampCtry	character	sampling country
	sampMeth	character	sampling method
hh		data.frame	haul information
	sampType	character	sampling type
	landCtry	character	landing country
	vslFlgCtry	character	vessel flag country
	year	numeric	year
	proj	character	project
	trpCode	character	trip code
	staNum	numeric	station number
	foVal	character	fishing operation validity
	catReg	character	parts (landings/discards) of the catch registered
	sppReg	character	species of the catch registered
	aggLev	character	aggregation level

	date	character	date
	time	character	time
	foDur	numeric	fishing operation duration
	latIni	numeric	latitude of shooting position in decimal degrees
	lonIni	numeric	longitude of shooting position in decimal degrees
	latFin	numeric	latitude of hauling position in decimal degrees
	lonFin	numeric	longitude of hauling position in decimal degrees
	area	character	area
	rect	character	statistical rectangle
	subRect	character	statistical subrectangle
	foDep	numeric	fishing operation depth
	waterDep	numeric	water depth
	foCatNat	character	fishing operations category, national
	foCatEu5	character	fishing operations category, european level 5
	foCatEu6	character	fishing operations category, european level 6
	meshSize	numeric	mesh size
	selDev	character	selectivity device
	meshSizeSelDev	numeric	mesh size in selectivity device
s1		data.frame	species list
	sampType	character	sampling type
	landCtry	character	landing country
	vsFlgCtry	character	vessel flag country
	year	numeric	year
	proj	character	project
	trpCode	character	trip code
	staNum	numeric	station number
	spp	character	species
	catchCat	character	catch category
	landCat	character	landing category
	commCatSel	character	commercial category scale
	commCat	character	commercial category
	subSampCat	character	subsampling category
	sex	character	gender
	wt	numeric	weight
	subSampWt	numeric	subsample weight
	lenCode	character	length code
h1		data.frame	length frequencies
	sampType	character	sampling type
	landCtry	character	landing country
	vsFlgCtry	character	vessel flag country
	year	numeric	year
	proj	character	project
	trpCode	character	trip code
	staNum	numeric	station number
	spp	character	species
	catchCat	character	catch category
	landCat	character	landing category
	commCatSel	character	commercial category scale
	commCat	character	commercial category
	subSampCat	character	subsampling category

	sex	character	gender
	lenCls	numeric	length class
	lenNum	numeric	frequency at length
ca		data.frame	growth and maturity
	sampType	character	sampling type
	landCtry	character	landing country
	vsFlgCtry	character	vessel flag country
	year	numeric	year
	proj	character	project
	trpCode	character	trip code
	staNum	numeric	station number
	quarter	numeric	quarter
	month	numeric	month
	spp	character	species
	sex	character	gender
	catchCat	character	catch category
	landCat	character	landing category
	commCatScl	character	commercial category scale
	commCat	character	commercial category
	stock	character	stock
	area	character	area
	rect	character	statistical rectangle
	subRect	character	statistical subrectangle
	lenCls	numeric	length class
	age	numeric	age
	fishId	numeric	individual identification
	lenCode	character	length code
	ageMeth	character	methodology for estimating the age
	plusGrp	character	plus group
	otoWt	numeric	otolith weight
	otoSide	character	otolith side
	indWt	numeric	individual weight
	matMeth	character	methodology for estimating the maturity stage
	matScale	character	maturity scale
	matStage	character	maturity stage

Methods

ca signature(object = "csDataVal"): ca slot accessor.

csDataCons signature(object = "csDataVal"): creator of Consolidated data class.

desc signature(object = "csDataVal"): desc slot accessor.

dim signature(x = "csDataVal"): dimensions of slots (see **dim**).

head signature(x = "csDataVal"): first rows of the each slot (see **head**).

hh signature(object = "csDataVal"): hh slot accessor.

hl signature(object = "csDataVal"): hl slot accessor.

rbind2 signature(x = "csDataVal", y = "csDataVal"): bind two **csDataVal** objects by adding rows (see **rbind2**).

`sl` signature(object = "csDataVal"): sl slot accessor.
`subset` signature(x = "csDataVal"): subset a `ceData` object (see `subset`).
`summary` signature(object = "csDataVal"): summary a `ceData` object (see `summary`).
`tail` signature(x = "csDataVal"): last rows of each slot (see `tail`).
`tr` signature(object = "csDataVal"): tr slot accessor.

Author(s)

Ernesto Jardim <ernesto@ipimar.pt>

Examples

```
showClass("csDataVal")
```

numeric.list

Ranges of numeric COST variables

Description

The minimum and maximum expected values of COST numeric variables.

Usage

```
data(numeric.list)
```

Format

A list of 40 variables each a numeric vector of length 2 with min and max expected value of the specified variable.

Source

FishFrame 5.0 Exchange Format Specification 25th January 2008.

Examples

```
data(numeric.list)
# the numeric variables
names(numeric.list)
# the expected age range of a fish
numeric.list$age
```

<code>rbind2-methods</code>	<i>rbind2 for 'COSTcore' classes</i>
-----------------------------	--------------------------------------

Description

This method implements `rbind2` for the classes provided by `COSTcore`.

Methods

```
signature(x = "ceData", y = "ceData")
signature(x = "ceDataCons", y = "ceDataCons")
signature(x = "clData", y = "clData")
signature(x = "clDataCons", y = "clDataCons")
signature(x = "csData", y = "csData")
signature(x = "csDataCons", y = "csDataCons")
```

<code>sole.ce</code>	<i>Sole commercial effort data.</i>
----------------------	-------------------------------------

Description

A `ceData` object containing sole data to be used for testing and development purposes.

Usage

```
sole.ce
```

Format

Formal class 'ceData' [package "COSTcore"] with 2 slots: `desc` and `ce`.

Details

To Do

WARNING

This data is provided is for testing only and should not be used for any other purpose.

Source

To Do

Examples

```
data(sole.ce)
```

`sole.cl`*Sole commercial landings data.*

Description

A clData object containing sole data to be used for testing and development purposes.

Usage

```
sole.cl
```

Format

Formal class 'clData' [package "COSTcore"] with 2 slots: `desc` and `cl`.

Details

To Do

WARNING

This data is provided is for testing only and should not be used for any other purpose.

Source

To Do

Examples

```
data(sole.cl)
```

`sole.cs`*Sole commercial sampling data.*

Description

A csData object containing sole data to be used for testing and development purposes.

Usage

```
sole.cs
```

Format

Formal class 'csData' [package "COSTcore"] with 6 slots: `desc`, `tr`, `hh`, `s1`, `h1` and `ca`.

Details

To Do

WARNING

This data is provided is for testing only and should not be used for any other purpose.

Source

To Do

Examples

```
data(sole.cs)
```

strIni-class	<i>Class "strIni"</i>
--------------	-----------------------

Description

The strIni class stores stratification and recoding information that is required for consolidated objects creation process.

Objects from the Class

The creator function "strIni" can be called to create objects from this class. Default values for slots are NAs, that means no stratification or no recoding)

Slots

slot	columns	class	description
timeStrata		character	time stratification (e.g "quarter", "month",...)
spaceStrata		character	space stratification (e.g "area",...)
techStrata		character	technical stratification (e.g "commCat", "foCatEu5",...)
tpRec		list	list for time strata recoding (e.g list(from="1",to="2"))
spRec		list	list for space strata recoding (e.g list(from=c("7D", "7D1"),to=c("27.7.d", "27.7.d")))
tcRec		list	list for technical strata recoding (e.g list(from=c("OTB-DEF", "OTB-MOL"),to=c("OTB", "OTB")))

Methods

csDataCons signature(object="csDataVal",objStrat="strIni"):csDataCons class creator.

clDataCons signature(object="clDataVal",objStrat="strIni"):clDataCons class creator.

ceDataCons signature(object="ceDataVal",objStrat="strIni"):ceDataCons class creator.

Author(s)

Mathieu Merzereaud <Mathieu.Merzereaud@ifremer.fr>

Examples

```
showClass("strIni")
```

subset-methods	subset for 'COSTcore' classes
----------------	-------------------------------

Description

This method implements subsetting for the classes provided by COSTcore.

Details

Subset methods for *csData* and *csDataVal* objects are quite specific. An additional "table" parameter (with default value `table="tr"`) allows to specify which table will be subset. According to the subset table, the impact on other tables will be various. A subset on *tr* will impact on all the datasets. A subset on *hh*, *sl* or *hl* will potentially affect all the tables except *ca*. Finally, a subset *ca* will impact on *tr*, and then incidentally on the other datasets. Another subsetting method proceeding on *sl* table with more restrictive effects is also available (see `subsetSpp`).

Methods

```
signature(x = "ceData")
signature(x = "ceDataVal")
signature(x = "ceDataCons")
signature(x = "clData")
signature(x = "clDataVal")
signature(x = "clDataCons")
signature(x = "csData")
signature(x = "csDataVal")
signature(x = "csDataCons")
```

See Also

`subsetSpp`

<code>subsetSpp</code>	<i>Specific subsetting function applying on SL table from COST objects</i>
------------------------	--

Description

This method implements subsetting for the raw, the validated and the consolidated CS objects provided by COSTcore, proceeding specifically on SL table. This subset only impacts on HL table, and preserves the other tables.

Usage

```
subsetSpp(x, subset, link=TRUE, ...)
```

Arguments

<code>x</code>	A <i>csData</i> , <i>csDataVal</i> or <i>csDataCons</i> object.
<code>subset</code>	Logical expression specifying the subsetting to be applied on sl table.
<code>link</code>	Logical. If TRUE, subset is also applied on ca table.
<code>...</code>	Further arguments.

See Also

`subset`, `csData-method`

Tests	<i>Functions to be used in tests.</i>
-------	---------------------------------------

Description

These functions are used to set up unit tests that are similar to those implemented by the package **RUnit**, the aim being to implement methods that follow sQuoteExtreme Programming best practices. The objectives of these functions are to test the functionality of the software and establish consistent debugging routines throughout the development process. For more about sQuoteExtreme Programming check Wikipedia's page http://en.wikipedia.org/wiki/Extreme_Programming.

Usage

```
setCon(con="zz")
startTest(file="testReport.txt")
tagTest(tag="My tag !", con=getOption("con"))
checkTrue(x, con = getOption("con"))
checkFalse(x, con = getOption("con"))
checkIdentical(x, y, con = getOption("con"))
checkRun(x, con = getOption("con"))
checkFail(x, con = getOption("con"))
checkEqual(x, y, con = getOption("con"))
finishTest(con=getOption("con"))
```

Arguments

x	R object to be used in test. It maybe an expression for <code>checkTrue</code> , <code>checkFalse</code> , <code>checkRun</code> or <code>checkFail</code> or an object to compared to <code>y</code> with <code>checkIdentical</code> or <code>checkEqual</code> .
y	R object to be compared to <code>x</code> in tests <code>checkIdentical</code> or <code>checkEqual</code> .
con	Name of communication.
file	Report file name.
tag	A tag.

Details

setCon Sets the name of connection to a file.

startTest Starts test by creating the report file and opening the connection to the file as defined by `sQuotecon`.

tagTest Tags the test for future reference. Automatically includes date and time.

checkTrue Checks 'x' expression is `TRUE` and reports to the file opened by the connection defined in 'con'.

checkFalse Checks 'x' expression is `FALSE` and reports to the file opened by the connection defined in 'con'.

checkIdentical Checks 'x' and 'y' are `identical` and reports to the file opened by the connection defined in 'con'.

checkRun Checks 'x' expression runs successfully and reports to the file opened by the connection defined in 'con'.

checkFail Checks 'x' expression fails to run and reports to the file opened by the connection defined in 'con'.

checkEqual Checks 'x' and 'y' are `equal` and reports to the file opened by the connection defined in 'con'.

finishTest Closes the connection defined by 'con'.

Value

A report with the results of the tests.

Note

A simple way to implement these tests is to set the result to `TRUE`, so that any `FALSE` results are immediately apparent.

The test must be set up in a single file that is to be used during R CMD `check` and should be placed on directory 'tests'. The procedure to set up a test is:

- 1 Set up the connection with `setCon()`. The name of the connection can be changed but has to be consistent in all functions.
- 2 Start the test, for example: `zz <- startTest("clDataTest.txt")`
- 3 Tag the test for later reference: `tagTest("1, 2, testing ...")`
- 4 Test expressions and comparisons with the `check` functions, *e.g.* `checkRun(mm <- rnorm(100))`
- 5 To finish test close the connection to the report file with `finishTest()`

Author(s)

FLCore Team and Ernesto Jardim (ernesto@ipimar.pt)

References

These functions were originally developed for FLR (<http://flr-project.org>).

<code>utils-methods</code>	<i>Utils methods in COSTcore</i>
----------------------------	----------------------------------

Description

Utilities to be used on `sQuoteCOST` classes that return meta information about the object's contents.

Methods

`dim signature(object)`: dimensions.
`head signature(object)`: first ten rows of data.
`summary signature(object)`: summary statistics.
`tail signature(object)`: last ten rows of data.

See Also

`dim`, `head`, `summary`, `tail`

<code>variable.list</code>	<i>Variables used within the COST data exchange format</i>
----------------------------	--

Description

A data frame of all the variables used within the COST data exchange format. Culled almost verbatim from the Fish Frame exchange format document of 25 January 2008.

Usage

```
data(variable.list)
```

Format

A data frame with 69 observations of 13 variables.

`variable` The abbreviated variable name.

`description` The full variable name.

`type` The type of the `variable`: character string "s", integer "i" or real "r".

`ce` Indicator variable for presence of `variable.list$variable` in the ce table. 1 if present 0 otherwise.

`c1` As above for the c1 table.

tr As above for the tr table.

hh As above for the hh table.

sl As above for the sl table.

hl As above for the hl table.

ca As above for the ca table.

requirement Required, either mandatory M or optional O

basic.checks Either numeric ranges or code.list

comments More about the variable

Source

FishFrame 5.0 Exchange Format Specification 25th January 2008.

Examples

```
data(variable.list)
```

Index

- *Topic **attribute**
 - check.fields, 6
- *Topic **classes**
 - ceData-class, 2
 - ceDataCons-class, 3
 - ceDataVal-class, 4
 - clData-class, 7
 - clDataCons-class, 9
 - clDataVal-class, 10
 - csData-class, 12
 - csDataCons-class, 16
 - csDataVal-class, 19
 - strIni-class, 25
- *Topic **datasets**
 - code.list, 11
 - numeric.list, 22
 - sole.ce, 23
 - sole.cl, 24
 - sole.cs, 24
 - variable.list, 29
- *Topic **debugging**
 - Tests, 27
- *Topic **error**
 - Tests, 27
- *Topic **methods**
 - accessor-methods, 1
 - check-methods, 6
 - checkDataIntegrity-methods, 7
 - coerceCons-methods, 12
 - coerceDataFrameColumns-methods, 12
 - rbind2-methods, 23
 - subset-methods, 26
 - subsetSpp, 27
 - utils-methods, 29
- *Topic **programming**
 - Tests, 27
- *Topic **utilities**
 - Tests, 27
- accessor-methods, 1
- ca (*accessor-methods*), 1
- ca,csData-method (*accessor-methods*), 1
- ca,csDataCons-method (*accessor-methods*), 1
- ca-methods (*accessor-methods*), 1
- ce (*accessor-methods*), 1
- ce,ceData-method (*accessor-methods*), 1
- ce,ceDataCons-method (*accessor-methods*), 1
- ce-methods (*accessor-methods*), 1
- ceData, 4
- ceData (*ceData-class*), 2
- ceData,character-method (*ceData-class*), 2
- ceData,data.frame-method (*ceData-class*), 2
- ceData,matrix-method (*ceData-class*), 2
- ceData,missing-method (*ceData-class*), 2
- ceData-class, **2**
- ceData-methods (*ceData-class*), 2
- ceDataCons (*ceDataCons-class*), 3
- ceDataCons,ceDataVal,missing-method (*ceDataCons-class*), 3
- ceDataCons,ceDataVal,strIni-method (*ceDataCons-class*), 3
- ceDataCons,ceDataVal-method (*ceDataCons-class*), 3
- ceDataCons,missing,missing-method (*ceDataCons-class*), 3
- ceDataCons,missing-method (*ceDataCons-class*), 3
- ceDataCons-class, **3**
- ceDataCons-methods (*ceDataCons-class*), 3
- ceDataVal, 3
- ceDataVal (*ceDataVal-class*), 4
- ceDataVal,ceData-method (*ceDataVal-class*), 4
- ceDataVal,missing-method (*ceDataVal-class*), 4

- ceDataVal-class, 4
- ceDataVal-methods (*ceDataVal-class*), 4
- check-methods, 6
- check.fields, 6
- checkCANms (*check-methods*), 6
- checkCANms, data.frame-method (*check-methods*), 6
- checkCANms-methods (*check-methods*), 6
- checkCApk (*check-methods*), 6
- checkCApk, data.frame-method (*check-methods*), 6
- checkCApk-methods (*check-methods*), 6
- checkCENms (*check-methods*), 6
- checkCENms, data.frame-method (*check-methods*), 6
- checkCENms-methods (*check-methods*), 6
- checkCEpk (*check-methods*), 6
- checkCEpk, data.frame-method (*check-methods*), 6
- checkCEpk-methods (*check-methods*), 6
- checkCLnms (*check-methods*), 6
- checkCLnms, data.frame-method (*check-methods*), 6
- checkCLnms-methods (*check-methods*), 6
- checkCLpk (*check-methods*), 6
- checkCLpk, data.frame-method (*check-methods*), 6
- checkCLpk-methods (*check-methods*), 6
- checkDataIntegrity (*checkDataIntegrity-methods*), 7
- checkDataIntegrity, data.frame, data.frame-method (*checkDataIntegrity-methods*), 7
- checkDataIntegrity-methods, 7
- checkEqual (*Tests*), 27
- checkFail (*Tests*), 27
- checkFalse (*Tests*), 27
- checkHHnms (*check-methods*), 6
- checkHHnms, data.frame-method (*check-methods*), 6
- checkHHnms-methods (*check-methods*), 6
- checkHHpk (*check-methods*), 6
- checkHHpk, data.frame-method (*check-methods*), 6
- checkHHpk-methods (*check-methods*), 6
- checkHLnms (*check-methods*), 6
- checkHLnms, data.frame-method (*check-methods*), 6
- checkHLnms-methods (*check-methods*), 6
- checkHLpk (*check-methods*), 6
- checkHLpk, data.frame-method (*check-methods*), 6
- checkHLpk-methods (*check-methods*), 6
- checkRun (*Tests*), 27
- checkSLnms (*check-methods*), 6
- checkSLnms, data.frame-method (*check-methods*), 6
- checkSLnms-methods (*check-methods*), 6
- checkSLpk (*check-methods*), 6
- checkSLpk, data.frame-method (*check-methods*), 6
- checkSLpk-methods (*check-methods*), 6
- checkTRnms (*check-methods*), 6
- checkTRnms, data.frame-method (*check-methods*), 6
- checkTRnms-methods (*check-methods*), 6
- checkTRpk (*check-methods*), 6
- checkTRpk, data.frame-method (*check-methods*), 6
- checkTRpk-methods (*check-methods*), 6
- checkTrue (*Tests*), 27
- checkTys (*check-methods*), 6
- checkTys, data.frame, list-method (*check-methods*), 6
- checkTys-methods (*check-methods*), 6
- cl (*accessor-methods*), 1
- cl, clData-method (*accessor-methods*), 1
- cl, clDataCons-method (*accessor-methods*), 1
- cl-methods (*accessor-methods*), 1
- clData, 10
- clData (*clData-class*), 7
- clData, character-method (*clData-class*), 7
- clData, data.frame-method (*clData-class*), 7
- clData, matrix-method (*clData-class*), 7
- clData, missing-method (*clData-class*), 7
- clData-class, 7
- clData-methods (*clData-class*), 7
- clDataCons (*clDataCons-class*), 9
- clDataCons, clDataVal, missing-method (*clDataCons-class*), 9
- clDataCons, clDataVal, strIni-method

- dim, clData-method (*utils-methods*), 29
- dim, clDataCons-method
(*utils-methods*), 29
- dim, csData-method (*utils-methods*), 29
- dim, csDataCons-method
(*utils-methods*), 29
- finishTest (*Tests*), 27
- head, 3–5, 8, 9, 11, 15, 18, 21, 29
- head, ceData-method (*utils-methods*),
29
- head, ceDataCons-method
(*utils-methods*), 29
- head, clData-method (*utils-methods*),
29
- head, clDataCons-method
(*utils-methods*), 29
- head, csData-method (*utils-methods*),
29
- head, csDataCons-method
(*utils-methods*), 29
- hh (*accessor-methods*), 1
- hh, csData-method (*accessor-methods*),
1
- hh, csDataCons-method
(*accessor-methods*), 1
- hh-methods (*accessor-methods*), 1
- hl (*accessor-methods*), 1
- hl, csData-method (*accessor-methods*),
1
- hl, csDataCons-method
(*accessor-methods*), 1
- hl-methods (*accessor-methods*), 1
- numeric.list, 7, 22
- rbind2, 3–5, 8, 9, 11, 15, 18, 21
- rbind2, ceData, ceData-method
(*rbind2-methods*), 23
- rbind2, ceDataCons, ceDataCons-method
(*rbind2-methods*), 23
- rbind2, clData, clData-method
(*rbind2-methods*), 23
- rbind2, clDataCons, clDataCons-method
(*rbind2-methods*), 23
- rbind2, csData, csData-method
(*rbind2-methods*), 23
- rbind2, csDataCons, csDataCons-method
(*rbind2-methods*), 23
- rbind2-methods, 23
- setCon (*Tests*), 27
- sl (*accessor-methods*), 1
- sl, csData-method (*accessor-methods*),
1
- sl, csDataCons-method
(*accessor-methods*), 1
- sl-methods (*accessor-methods*), 1
- sole.ce, 23
- sole.cl, 24
- sole.cs, 24
- startTest (*Tests*), 27
- strIni (*strIni-class*), 25
- strIni-class, 25
- strIni-methods (*strIni-class*), 25
- subset, 3–5, 8, 9, 11, 15, 18, 22
- subset, ceData-method
(*subset-methods*), 26
- subset, ceDataCons-method
(*subset-methods*), 26
- subset, ceDataVal-method
(*subset-methods*), 26
- subset, clData-method
(*subset-methods*), 26
- subset, clDataCons-method
(*subset-methods*), 26
- subset, clDataVal-method
(*subset-methods*), 26
- subset, csData-method, 27
- subset, csData-method
(*subset-methods*), 26
- subset, csDataCons-method
(*subset-methods*), 26
- subset, csDataVal-method
(*subset-methods*), 26
- subset-methods, 26
- subsetSpp, 26, 27
- subsetSpp, csData-method (*subsetSpp*),
27
- subsetSpp, csDataCons-method
(*subsetSpp*), 27
- subsetSpp, csDataVal-method
(*subsetSpp*), 27
- summary, 3–5, 8, 9, 11, 15, 18, 22, 29
- summary, ceData-method
(*utils-methods*), 29
- summary, ceDataCons-method
(*utils-methods*), 29
- summary, clData-method
(*utils-methods*), 29
- summary, clDataCons-method
(*utils-methods*), 29
- summary, csData-method
(*utils-methods*), 29

summary, csDataCons-method
 (*utils-methods*), 29

tagTest (*Tests*), 27

tail, 3-5, 8, 9, 11, 15, 18, 22, 29

tail, ceData-method (*utils-methods*),
 29

tail, ceDataCons-method
 (*utils-methods*), 29

tail, clData-method (*utils-methods*),
 29

tail, clDataCons-method
 (*utils-methods*), 29

tail, csData-method (*utils-methods*),
 29

tail, csDataCons-method
 (*utils-methods*), 29

Tests, **27**

tr (*accessor-methods*), 1

tr, csData-method (*accessor-methods*),
 1

tr, csDataCons-method
 (*accessor-methods*), 1

tr-methods (*accessor-methods*), 1

utils-methods, **29**

variable.list, 7, **29**