

Package ‘crunch’

November 23, 2016

Type Package

Title Crunch.io Data Tools

Description The Crunch.io service <<http://crunch.io/>> provides a cloud-based data store and analytic engine, as well as an intuitive web interface. Using this package, analysts can interact with and manipulate Crunch datasets from within R. Importantly, this allows technical researchers to collaborate naturally with team members, managers, and clients who prefer a point-and-click interface.

Version 1.14.2

Date 2016-11-22

Author Neal Richardson [aut, cre]

Maintainer Neal Richardson <neal@crunch.io>

URL <https://github.com/Crunch-io/rcrunch>

BugReports <https://github.com/Crunch-io/rcrunch/issues>

License LGPL (>= 3)

Depends R (>= 3.0.0)

Imports httr (>= 1.0.0), httpcache (>= 0.1.4), jsonlite (>= 0.9.15), methods, curl

Suggests knitr, testthat

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-11-23 01:35:29

R topics documented:

addSubvariable	4
addVariables	5

appendDataset	5
as.environment,CrunchDataset-method	6
as.vector,CrunchExpr-method	6
batches	7
c-categories	8
catalog-extract	8
catalog-length	12
catalogToDataFrame	12
Categories-class	13
category-extract	14
cleanseBatches	15
combine	15
compareDatasets	16
consent	17
ContextManager	18
copyVariable	18
crtabs	19
crunch	20
crunch-uni	20
crunchBox	21
CrunchDataset-class	22
CrunchVariable-class	22
cube-computing	22
cube-methods	23
dataset-extract	24
dataset-owner	25
dataset-to-R	26
dataset-update	26
dataset-variables	28
DatasetOrder-class	28
datasets	29
dedupeOrder	30
delete	30
deleteDataset	31
deleteSubvariables	32
deleteVariables	33
describe	33
describe-category	35
dichotomize	36
dim-dataset	37
dropRows	38
duplicated	39
embedCrunchBox	39
entity,CrunchProject-method	40
exclusion	42
exportDataset	43
expressions	43
filter-catalog	45

filter-methods	46
flattenOrder	46
flipArrays	47
forkDataset	47
getAccountUserCatalog	48
getTeams	48
grouped	49
hiddenVariables	49
hide	50
hideVariables	50
http-methods	51
index	52
is-na-categories	52
is.archived,DatasetCatalog-method	53
is.dataset	54
is.editor	55
joinDatasets	56
listDatasets	57
loadDataset	58
locateEntity	58
lock	59
login	59
logout	60
makeArray	60
me	61
members<- ,CrunchTeam,MemberCatalog-method	61
mergeFork	62
moveToGroup	63
na-omit-categories	64
names,BatchCatalog-method	64
newDataset	66
newDatasetByColumn	67
newDatasetByCSV	67
newDatasetFromFile	68
ordering	68
owners	69
permissions	70
preCrunchBoxCheck	70
project-icon	71
projects	71
refresh	72
removeEmptyGroups	72
restoreVersion	73
saveVersion	74
self	74
session	75
session-extract	75
settings	76

share	77
ShojiObject-class	77
ShojiOrder-extract	78
ShojiOrder-length	80
ShojiOrder-slots	81
show-crunch	82
Subvariables-class	83
subvars-extract	84
table	86
temp.options	87
tojson-crunch	87
toVariable	88
type	89
unbind	90
unshare	90
updateDatasetList	91
var-categories	91
variable-update	92
VariableCatalog-class	94
VariableDefinition	94
variableMetadata	95
VariableOrder-class	95
versions	96
weight	96
which	97
with-context-manager	97
[,CrunchExpr,CrunchLogicalExpr,ANY-method	98

Index **99**

addSubvariable	<i>Add subvariable to an array</i>
----------------	------------------------------------

Description

Add subvariable to an array

Usage

addSubvariable(variable, subvariable)

addSubvariables(variable, subvariable)

Arguments

variable the array variable to modify

subvariable the subvariable to add, or a list of those to add, or a dataset subset

Value

variable with the indicated subvariables added.

Examples

```
## Not run:
ds$allpets <- addSubvariable(ds$allpets, ds$allpets_4)
ds$petloc <- addSubvariables(ds$petloc, ds[c("petloc_school", "petloc_daycare")])

## End(Not run)
```

addVariables	<i>Add multiple variables to a dataset</i>
--------------	--

Description

This function lets you add more than one variable at a time to a dataset. If you have multiple variables to add, this function will be faster than doing `ds$var <- value` assignment because it doesn't refresh the dataset's state in between variable POST requests.

Usage

```
addVariables(dataset, ...)
```

Arguments

dataset a CrunchDataset
 ... [VariableDefinitions](#) or a list of VariableDefinitions.

Value

dataset with the new variables added (invisibly)

appendDataset	<i>Append one Crunch dataset to another</i>
---------------	---

Description

Append one Crunch dataset to another

Usage

```
appendDataset(dataset1, dataset2, autorollback = TRUE,
  cleanup = autorollback)
```

Arguments

dataset1	a CrunchDataset
dataset2	another CrunchDataset, or possibly a data.frame. If dataset2 is not a Crunch dataset, it will be uploaded as a new dataset before appending.
autorollback	logical: If the append fails, revert the dataset back to its state before attempting to append? Default is TRUE, and you probably won't want to change that.
cleanup	Deprecated in favor of autorollback. See also cleanseBatches .

Value

A CrunchDataset with dataset2 appended to dataset1

as.environment, CrunchDataset-method
as.environment method for CrunchDataset

Description

This method allows you to eval within a Dataset.

Usage

```
## S4 method for signature 'CrunchDataset'
as.environment(x)
```

Arguments

x	CrunchDataset
---	---------------

Value

an environment in which named objects are (promises that return) CrunchVariables.

as.vector, CrunchExpr-method
Convert Variables to local R objects

Description

Convert Variables to local R objects

Usage

```
## S4 method for signature 'CrunchExpr'
as.vector(x, mode = "any")

## S4 method for signature 'CrunchVariable'
as.vector(x, mode = "any")
```

Arguments

x	a CrunchVariable subclass
mode	for Categorical variables, one of either "factor" (default, which returns the values as factor); "numeric" (which returns the numeric values); or "id" (which returns the category ids). If "id", values corresponding to missing categories will return as the underlying integer codes; i.e., the R representation will not have any NAs. Otherwise, missing categories will all be returned NA. For non-Categorical variables, the mode argument is ignored.

Value

an R vector of the type corresponding to the Variable. E.g. CategoricalVariable yields type factor by default, NumericVariable yields numeric, etc.

batches	<i>See the appended batches of this dataset</i>
---------	---

Description

See the appended batches of this dataset

Usage

```
batches(x)
```

Arguments

x	a CrunchDataset
---	-----------------

Value

a BatchCatalog

c-categories

S3 method to concatenate Categories and Category objects

Description

S3 method to concatenate Categories and Category objects

Usage

```
## S3 method for class 'Categories'  
c(...)  
  
## S3 method for class 'Category'  
c(...)
```

Arguments

... see [c](#)

Value

An object of class [Categories](#)

Examples

```
cat.a <- Category(name="First", id=1, numeric_value=1, missing=FALSE)  
cat.b <- Category(name="Second", id=2)  
cat.c <- Category(name="Third", id=3, missing=TRUE)  
cats.1 <- Categories(cat.a, cat.b)  
identical(cats.1, c(cat.a, cat.b))  
identical(c(cats.1, cat.c), Categories(cat.a, cat.b, cat.c))
```

catalog-extract*Extract and modify subsets of Catalog-type objects*

Description

Extract and modify subsets of Catalog-type objects

Usage

```
## S4 method for signature 'DatasetCatalog,character'
x[[i, j, ...]]

## S4 method for signature 'DatasetCatalog,ANY'
x[[i, j, ...]]

## S4 replacement method for signature 'DatasetCatalog,character,missing,DatasetTuple'
x[[i, j]] <- value

## S4 method for signature 'FilterCatalog,character'
x[[i, j, ...]]

## S4 method for signature 'FilterCatalog,numeric'
x[[i, j, ...]]

## S4 replacement method for signature 'FilterCatalog,character,missing,CrunchLogicalExpr'
x[[i, j]] <-
  value

## S4 replacement method for signature 'FilterCatalog,numeric,missing,CrunchLogicalExpr'
x[[i, j]] <- value

## S4 replacement method for signature 'FilterCatalog,character,missing,CrunchFilter'
x[[i, j]] <- value

## S4 replacement method for signature 'FilterCatalog,numeric,missing,CrunchFilter'
x[[i, j]] <- value

## S4 method for signature 'MemberCatalog,character'
x[[i, j, ...]]

## S4 method for signature 'MemberCatalog,character,ANY'
x[i, j, ..., drop = TRUE]

## S4 replacement method for signature 'MemberCatalog,ANY,missing,ANY'
x[[i, j]] <- value

## S4 replacement method for signature 'MemberCatalog,character,missing,`NULL`'
x[[i, j]] <- value

## S4 method for signature 'PermissionCatalog,character'
x[[i, j, ...]]

## S4 method for signature 'PermissionCatalog,character,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'ProjectCatalog,character'
```

```
x[[i, j, ...]]

## S4 method for signature 'ProjectCatalog,ANY'
x[[i, j, ...]]

## S4 replacement method for signature 'ProjectCatalog,character,missing,list'
x[[i, j]] <- value

## S4 replacement method for signature 'ProjectCatalog,character,missing,CrunchProject'
x[[i, j]] <- value

## S4 method for signature 'ShojiCatalog,character,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'ShojiCatalog,numeric,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'ShojiCatalog,logical,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'ShojiCatalog,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'ShojiCatalog,ANY'
x[[i, j, ...]]

## S4 method for signature 'ShojiCatalog,character'
x[[i, j, ...]]

## S4 method for signature 'ShojiCatalog'
x$name

## S4 replacement method for signature 'ShojiCatalog'
x$name <- value

## S4 replacement method for signature 'ShojiCatalog,ANY,missing,ShojiCatalog'
x[i, j] <- value

## S4 method for signature 'TeamCatalog,character'
x[[i, j, ...]]

## S4 method for signature 'TeamCatalog,numeric'
x[[i, j, ...]]

## S4 replacement method for signature 'TeamCatalog,character,missing,list'
x[[i, j]] <- value

## S4 replacement method for signature 'TeamCatalog,character,missing,CrunchTeam'
```

```

x[[i, j]] <- value

## S4 method for signature 'VariableCatalog,character'
x[[i, j, ...]]

## S4 method for signature 'VariableCatalog,ANY'
x[[i, j, ...]]

## S4 replacement method for signature 'VariableCatalog,character,missing,VariableTuple'
x[[i, j]] <- value

## S4 replacement method for signature 'VariableCatalog,character,missing,CrunchVariable'
x[[i, j]] <-
  value

## S4 method for signature 'VariableCatalog,VariableOrder,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'VariableCatalog,VariableGroup,ANY'
x[i, j, ..., drop = TRUE]

## S4 replacement method for signature
## 'VariableCatalog,VariableOrder,missing,VariableCatalog'
x[i, j] <-
  value

## S4 replacement method for signature
## 'VariableCatalog,VariableGroup,missing,VariableCatalog'
x[i, j] <-
  value

```

Arguments

x	a Catalog object
i	which catalog elements to extract
j	Invalid
...	additional arguments
value	For updating, an object of the appropriate class and size to insert
drop	Invalid
name	for \$, the same as i for []

Value

A subset of x if extracting, otherwise x duly modified

catalog-length	<i>Length of Catalog</i>
----------------	--------------------------

Description

Length of Catalog

Usage

```
## S4 method for signature 'ShojiCatalog'
length(x)
```

Arguments

x a Catalog

Value

Integer: the number of elements in the index list

catalogToDataFrame	<i>Utility to get a more human-readable view of a Shoji Catalog</i>
--------------------	---

Description

Utility to get a more human-readable view of a Shoji Catalog

Usage

```
catalogToDataFrame(x, keys = TRUE, rownames, ...)
```

Arguments

x	ShojiCatalog or subclass
keys	character vector of attribute names from each catalog tuple to include in the result. Default is TRUE, which means all.
rownames	See data.frame , the row.names argument, to which this is passed in data.frame. The difference here is that if rownames is explicitly set as NULL, the resulting object will not have row names set. By default, row names will be the URLs of the catalog tuples.
...	additional arguments passed to data.frame

Value

a data.frame view of the catalog

Categories-class *Categories in CategoricalVariables*

Description

CategoricalVariables, as well as the array types composed from Categoricals, contain Categories. Categories are a subclass of list that contains only Category objects. Category objects themselves subclass list and contain the following fields: "name", "id", "numeric_value", "missing", and optionally "selected".

Usage

```

Categories(..., data = NULL)

Category(..., data = NULL)

## S4 method for signature 'Categories,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'Categories,numeric,ANY'
x[i, j, ..., drop = TRUE]

## S4 replacement method for signature 'Categories,ANY,ANY,ANY'
x[i, j, ...] <- value

## S4 method for signature 'Categories'
names(x)

## S4 method for signature 'Categories'
values(x)

## S4 method for signature 'Categories'
ids(x)

## S4 replacement method for signature 'Categories'
names(x) <- value

## S4 replacement method for signature 'Categories'
values(x) <- value

## S4 replacement method for signature 'Categories'
ids(x) <- value

```

Arguments

... additional arguments to [, ignored

data	For the constructor functions <code>Category</code> and <code>Categories</code> , you can either pass in attributes via <code>...</code> or you can create the objects with a fully defined list representation of the objects via the <code>data</code> argument. See the examples.
x	For the attribute getters and setters, an object of class <code>Category</code> or <code>Categories</code>
i	For the <code>[]</code> methods, just as with list extract methods
j	Invalid argument to <code>[]</code> , but in the generic's signature
drop	Invalid argument to <code>[]</code> , but in the generic's signature
value	For <code>[-</code> , the replacement <code>Category</code> to insert

Examples

```
cat.a <- Category(name="First", id=1, numeric_value=1, missing=FALSE)
cat.b <- Category(data=list(name="First", id=1, numeric_value=1, missing=FALSE))
identical(cat.a, cat.b)
cat.c <- Category(name="Second", id=2)
cats.1 <- Categories(cat.a, cat.c)
cats.2 <- Categories(data=list(cat.a, cat.c))
identical(cats.1, cats.2)
```

category-extract	<i>Access Category fields directly</i>
------------------	--

Description

Don't do this. Instead, use the category setters.

Usage

```
## S4 method for signature 'Category'
x$name

## S4 replacement method for signature 'Category'
x$name <- value
```

Arguments

x	a <code>Category</code>
name	a field within x
value	a value for that field to update

Value

`$` returns the value of the desired field. Setter returns x duly modified.

See Also

[describe-category](#)

cleanseBatches	<i>Remove batches from a dataset</i>
----------------	--------------------------------------

Description

Sometimes append operations do not succeed, whether due to conflicts between the two datasets or other server-side issues. Failed appends can leave behind "error" status batch records, which can cause confusion. This function lets you delete batches that don't match the status or statuses you want to keep.

Usage

```
cleanseBatches(dataset, keep = c("imported", "appended"))
```

Arguments

dataset	CrunchDataset
keep	character batch status(es) you want to keep. By default, batches that don't have either "imported" or "appended" status will be deleted.

Value

dataset with the undesired batches removed.

combine	<i>Combine categories or responses</i>
---------	--

Description

Combine categories or responses

Usage

```
combine(variable, combinations = list(), ...)
```

Arguments

variable	Categorical, Categorical Array, or Multiple Response variable
combinations	list of named lists containing (1) "categories": category ids or names for categorical types, or for multiple response, "responses": subvariable names, aliases, or positional indices; (2) a "name" for the new category or response; and (3) optionally, other category ("missing", "numeric_value") or subvariable ("alias", "description") attributes. If combinations is omitted, the resulting variable will essentially be a copy (but see link{copy} for a more natural way to do that, if desired).
...	Additional variable metadata for the new derived variable

Value

A `VariableDefinition` that will create the new comined-category or -response derived variable. Categories/responses not referenced in combinations will be appended to the end of the combinations.

Examples

```
## Not run:
ds$fav_pet2 <- combine(ds$fav_pet, name="Pets (combined)",
  combinations=list(list(name="Mammals", categories=c("Cat", "Dog")),
    list(name="Reptiles", categories=c("Snake", "Lizard"))))
ds$pets_owned2 <- combine(ds$allpets, name="Pets owned (collapsed)",
  combinations=list(list(name="Mammals", responses=c("Cat", "Dog"))))

## End(Not run)
```

 compareDatasets

Compare two datasets to see how they will append

Description

When one dataset is appended to another, variables and subvariables are matched on their aliases, and then categories for variables that have them are matched on category name. This function lines up the metadata between two datasets as the append operation will so that you can inspect how well the datasets will align before you do the append.

Usage

```
compareDatasets(A, B)
```

Arguments

A	CrunchDataset
B	CrunchDataset

Details

Calling `summary` on the return of this function will print an overview of places where the matching on variable alias and category name may lead to undesired outcomes, enabling you to alter one or both datasets to result in better alignment.

Value

An object of class 'compareDatasets', a list of three elements: (1) 'variables', a data.frame of variable metadata joined on alias; (2) 'categories', a list of data.frames of category metadata joined on category name, one for each variable with categories; and (3) 'subvariables', a list of data.frames of subvariable metadata joined on alias, one for each array variable.

Summary output reports on (1) variables that, when matched across datasets by alias, have different types; (2) variables that have the same name but don't match on alias; (3) for variables that match and have categories, any categories that have the same id but don't match on name; (4) for array variables that match, any subvariables that have the same name but don't match on alias; and (5) array variables that, after assembling the union of their subvariables, point to subvariables that belong to other arrays.

Examples

```
## Not run:  
  comp <- compareDataset(ds1, ds2)  
  summary(comp)  
  
## End(Not run)
```

consent

Give consent to do things that require permission

Description

Give consent to do things that require permission

Usage

```
consent()
```

Value

an S3 class "contextManager" object

See Also

[with-context-manager](#) [ContextManager](#)

ContextManager *Context managers*

Description

Context managers

Usage

```
ContextManager(enter = function() { }, exit = function() { },
               error = NULL, as = NULL)
```

Arguments

enter	function to run before doing things
exit	function to run after doing things
error	optional function to run if an error is thrown
as	character optional way to specify a default name for assinging the return of the enter function.

Value

an S3 class "contextManager" object

See Also

[with-context-manager](#)

copyVariable *Copy a variable*

Description

Makes a copy of a Crunch variable on the server.

Usage

```
copyVariable(x, deep = FALSE, ...)
```

```
copy(x, deep = FALSE, ...)
```

Arguments

x	a CrunchVariable to copy
deep	logical: should this be a deep copy, in which there is no dependence on the original variable, or a shallow one, in which the copy is more of a symbolic link? Default is FALSE, meaning symlink.
...	Additional metadata to give to the new variable. If not given, the new variable will have a name that is the same as the original but with " (copy)" appended, and its alias will be the old alias with "_copy" appended.

Details

Copies can be shallow (linked) or deep. Shallow copying is faster and should be preferred unless a true hard copy is required, though keep in mind the implications of shallow copying. When you append data to the original variable or otherwise alter its values, the values in the copy automatically update. This linking may be desirable, but it comes with some limitations. First, you cannot edit the values of the copy independently of the original. Second, some attributes of the copy are immutable: of note, properties of categories cannot be altered independently in the copy. Subvariable names and ordering within arrays, however, can.

Value

a VariableDefinition for the copy expression. Assign into a Dataset to make the copy happen.

crtabs	<i>Crunch xtabs: Crosstab and otherwise aggregate variables in a Crunch Dataset</i>
--------	---

Description

Create a contingency table or other aggregation from cross-classifying variables in a CrunchDataset.

Usage

```
crtabs(formula, data, weight = crunch::weight(data), useNA = c("no",
  "ifany", "always"))
```

Arguments

formula	an object of class 'formula' object with the cross-classifying variables, separated by '+', on the right hand side. Compare to xtabs .
data	an object of class CrunchDataset
weight	a CrunchVariable that has been designated as a potential weight variable for data, or NULL for unweighted results. Default is the currently applied weight, weight (data).
useNA	whether to include missing values in tabular results. See table .

Value

an object of class CrunchCube

crunch	<i>Crunch.io: instant, visual, collaborative data analysis</i>
--------	--

Description

Crunch.io provides a cloud-based data store and analytic engine. It has a **web client** for interactive data exploration and visualization. The crunch package for R allows analysts to interact with and manipulate Crunch datasets from within R. Importantly, this allows technical researchers to collaborate naturally with team members, managers, and clients who prefer a point-and-click interface: because all connect to the same dataset in the cloud, there is no need to email files back and forth continually to share results.

See Also

To learn more about using the package, see `vignette("getting-started", package="crunch")`. To sign up for a Crunch.io account, visit <https://app.crunch.io/>.

crunch-uni	<i>Univariate statistics on Crunch objects</i>
------------	--

Description

Univariate statistics on Crunch objects

Usage

```
## S4 method for signature 'CrunchVariable'
mean(x, ...)

## S4 method for signature 'NumericVariable'
mean(x, ...)

## S4 method for signature 'CrunchVariable'
sd(x, na.rm = FALSE)

## S4 method for signature 'NumericVariable'
sd(x, na.rm = FALSE)

## S4 method for signature 'CrunchVariable'
median(x, na.rm = FALSE)

## S4 method for signature 'NumericVariable'
```

```

median(x, na.rm = FALSE)

## S4 method for signature 'CrunchVariable'
min(x, na.rm)

## S4 method for signature 'NumericVariable'
min(x, na.rm = FALSE)

## S4 method for signature 'DatetimeVariable'
min(x, na.rm = FALSE)

## S4 method for signature 'CrunchVariable'
max(x, na.rm)

## S4 method for signature 'NumericVariable'
max(x, na.rm = FALSE)

## S4 method for signature 'DatetimeVariable'
max(x, na.rm = FALSE)

```

Arguments

x	a NumericVariable, or for min and max, possibly a DatetimeVariable
...	additional arguments to mean
na.rm	logical: exclude missings?

See Also

[mean](#) [sd](#) [median](#) [min](#) [max](#)

crunchBox

Make a CrunchBox

Description

CrunchBoxes allow you to publish results to the world.

Usage

```
crunchBox(dataset, filters = crunch::filters(dataset), ...)
```

Arguments

dataset	CrunchDataset
filters	FilterCatalog, or NULL for no filters. Default all filters in your catalog, <code>filters(dataset)</code> .
...	additional metadata for the box, such as "title", "header", etc.

Value

The URL to the newly created box.

See Also

[preCrunchBoxCheck](#) to provide guidance on what you're including in the CrunchBox

CrunchDataset-class *Crunch Datasets*

Description

Crunch Datasets

CrunchVariable-class *Variables in Crunch*

Description

Variables are S4 objects. All inherit from the base class `CrunchVariable`. be persisted on the server? Default is FALSE

Slots

`filter` either NULL or `CrunchLogicalExpr`

cube-computing *Work with CrunchCubes*

Description

Crunch.io supports more complex data types than base R does, such as multiple response and array types. If you want to compute margin or proportion tables on an aggregation of these variable types, special methods are required. These functions provide an interface like [margin.table](#) and [prop.table](#) for the `CrunchCube` object, handling those special data types.

Usage

```
## S4 method for signature 'CrunchCube'
margin.table(x, margin = NULL)
```

```
## S4 method for signature 'CrunchCube'
prop.table(x, margin = NULL)
```

```
## S4 method for signature 'CrunchCube'
round(x, digits = 0)
```

Arguments

x	a CrunchCube
margin	index, or vector of indices to generate margin for. See prop.table
digits	see round

Value

The appropriate `margin.table` or `prop.table`.

See Also

[margin.table](#) [prop.table](#)

cube-methods

Methods on Cube objects

Description

These methods provide an array-like interface to the `CrunchCube` object.

Usage

```
## S4 method for signature 'CubeDims'  
dimnames(x)
```

```
## S4 method for signature 'CubeDims'  
dim(x)
```

```
## S4 method for signature 'CubeDims'  
is.na(x)
```

Arguments

x	a <code>CrunchCube</code> or its <code>CubeDims</code> component.
---	---

Value

Generally, the same shape of result that each of these functions return when applied to an array object.

See Also

[cube-computing](#)

dataset-extract *Subset datasets and extract variables*

Description

Subset datasets and extract variables

Usage

```
## S4 method for signature 'CrunchDataset,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CrunchDataset,logical,missing'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'CrunchDataset,character,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CrunchDataset,VariableGroup,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CrunchDataset,VariableOrder,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CrunchDataset,missing,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'CrunchDataset,CrunchLogicalExpr,missing'
x[i, j, ...,
  drop = FALSE]

## S4 method for signature 'CrunchDataset,CrunchLogicalExpr,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'CrunchDataset'
subset(x, ...)

## S4 method for signature 'CrunchDataset,ANY'
x[[i, ..., drop = FALSE]]

## S4 method for signature 'CrunchDataset,character'
x[[i, ..., drop = FALSE]]

## S4 method for signature 'CrunchDataset'
x$name
```


Arguments

x	a CrunchDataset
i	As with a <code>data.frame</code> , there are two cases: (1) if no other arguments are supplied (i.e <code>x[i]</code>), <code>i</code> provides for <code>as.list</code> extraction: columns of the dataset rather than rows. If character, identifies variables to extract based on their aliases (by default: <code>set options(crunch.namekey.dataset="name")</code> to use variable names); if numeric or logical, extracts variables accordingly. Alternatively, (2) if <code>j</code> is specified (as either <code>x[i, j]</code> or <code>x[i,]</code>), <code>i</code> is an object of class <code>CrunchLogicalExpr</code> that will define a subset of rows.
j	columnar extraction, as described above
...	additional arguments
drop	logical: automatically simplify a 1-column Dataset to a Variable? Default is FALSE, and the TRUE option is in fact not implemented.
name	columnar extraction for \$

Value

`[]` yields a Dataset; `[[` and `$` return a Variable

dataset-owner	<i>Change the owner of a dataset</i>
---------------	--------------------------------------

Description

Change the owner of a dataset

Usage

```
## S4 method for signature 'CrunchDataset'
owner(x)

## S4 replacement method for signature 'CrunchDataset'
owner(x) <- value
```

Arguments

x	CrunchDataset
value	For the setter, either a URL (character) or a Crunch object with a <code>self</code> method. Users and Projects are valid objects to assign as dataset owners.

Value

The dataset.

dataset-to-R

as.data.frame method for CrunchDataset

Description

This method is defined principally so that you can use a `CrunchDataset` as a `data` argument to other R functions (such as `lm`). Unless you give it the `force==TRUE` argument, this function does not in fact return a `data.frame`: it returns an object with an interface like a `data.frame`, such that you get R vectors when you access its columns (unlike a `CrunchDataset`, which returns `CrunchVariable` objects). This allows modeling functions that require select columns of a dataset to retrieve only those variables from the remote server, rather than pulling the entire dataset into local memory.

Usage

```
## S3 method for class 'CrunchDataset'
as.data.frame(x, row.names = NULL, optional = FALSE,
             force = FALSE, ...)

## S3 method for class 'CrunchDataFrame'
as.data.frame(x, row.names = NULL,
             optional = FALSE, ...)
```

Arguments

<code>x</code>	a <code>CrunchDataset</code>
<code>row.names</code>	part of <code>as.data.frame</code> signature. Ignored.
<code>optional</code>	part of <code>as.data.frame</code> signature. Ignored.
<code>force</code>	logical: actually coerce the dataset to <code>data.frame</code> , or leave the columns as unevaluated promises. Default is <code>FALSE</code> .
<code>...</code>	additional arguments passed to <code>as.data.frame.default</code>

Value

an object of class `CrunchDataFrame` unless `force`, in which case the return is a `data.frame`.

dataset-update

Update a variable or variables in a dataset

Description

Update a variable or variables in a dataset

Usage

```
## S4 replacement method for signature 'CrunchDataset,character,missing,CrunchVariable'
x[[i]] <- value

## S4 replacement method for signature 'CrunchDataset,ANY,missing,CrunchVariable'
x[[i]] <- value

## S4 replacement method for signature 'CrunchDataset,character,missing,ANY'
x[[i]] <- value

## S4 replacement method for signature 'CrunchDataset,character,missing,CrunchLogicalExpr'
x[[i]] <- value

## S4 replacement method for signature 'CrunchDataset,ANY,ANY,ANY'
x[[i]] <- value

## S4 replacement method for signature 'CrunchDataset,character,missing,`NULL`'
x[[i]] <- value

## S4 replacement method for signature 'CrunchDataset,ANY,missing,`NULL`'
x[[i]] <- value

## S4 replacement method for signature 'CrunchDataset'
x$name <- value

## S4 replacement method for signature 'CrunchDataset,ANY,missing,list'
x[i, j] <- value

## S4 replacement method for signature 'CrunchDataset,CrunchExpr,ANY,ANY'
x[i, j] <- value
```

Arguments

x	a CrunchDataset
i	For [, a CrunchLogicalExpr, numeric, or logical vector defining a subset of the rows of x. For [[, see j for the as.list column subsetting.
value	replacement values to insert. These can be crunchExprs or R vectors of the corresponding type
name	like j but for \$
j	if character, identifies variables to extract based on their aliases (by default: set options(crunch.namekey.dataset="name") to use variable names); if numeric or logical, extracts variables accordingly. Note that this is the as.list extraction, columns of the dataset rather than rows.

Value

x, modified.

dataset-variables *Access a Dataset's Variables Catalog*

Description

Datasets contain collections of variables. For a few purposes, such as editing variables' metadata, it is helpful to access these variable catalogs more directly.

Usage

```
## S4 method for signature 'CrunchDataset'
variables(x)

## S4 replacement method for signature 'CrunchDataset,VariableCatalog'
variables(x) <- value

## S4 method for signature 'CrunchDataset'
allVariables(x)

## S4 replacement method for signature 'CrunchDataset,VariableCatalog'
allVariables(x) <- value
```

Arguments

x a Dataset
value For the setters, a VariableCatalog to assign.

Details

variables gives just the active variables in the dataset, while allVariables, as the name suggests, yields all variables, including hidden variables.

Value

Getters return VariableCatalog; setters return x duly modified.

DatasetOrder-class *Organize Datasets*

Description

A DatasetOrder object is a subclass of list that contains DatasetGroups. DatasetGroup objects contain a group name and an set of "entities", which can be dataset references or other nested DatasetGroups.

Slots

`group` character, the name of the DatasetGroup. In the constructor and more generally, this field can be referenced as "name" as well.

`entities` a character vector of dataset URLs, or a list containing a combination of dataset URLs and DatasetGroup objects.

datasets	<i>Get the dataset catalog</i>
----------	--------------------------------

Description

Get the dataset catalog

Usage

```
datasets(x = getAPIRoot())
```

```
datasets(x) <- value
```

Arguments

`x` a ShojiObject, such as a CrunchProject. If omitted, the default value for `x` means that you will load the user's primary dataset catalog.

`value` CrunchDataset for the setter

Value

An object of class DatasetCatalog. The setter returns the project (or other object that contains a dataset catalog with the given dataset added to it (via changing its owner to be the specified object, `x`).

Examples

```
## Not run:
# Get the primary dataset catalog
mydatasets <- datasets()
# Can load a dataset from that
ds <- loadDataset(mydatasets[["Dataset name"]])
# Can use the same function to get the dataset catalog for a project
proj <- projects()[["Project name"]]
projdatasets <- datasets(proj)
# The assignment method lets you move a dataset to a project
datasets(proj) <- ds

## End(Not run)
```

dedupeOrder	<i>Remove duplicated entites from an order/group</i>
-------------	--

Description

This function recurses through a ShojiOrder/OrderGroup and removes any duplicate entities. As with the default of `duplicated`, the first appearance of an entity is kept, and subsequent occurrences are marked as duplicated and removed. (Unlike `duplicated`, however, there is no option to reverse that order.) "First" occurrence of an entity is determined by the function's recursion: within each group, nested groups are processed first, in order, and recursively their nested groups are processed. See the test suite, in `test-variable-order.R`, for an example that illustrates which entities are dropped as duplicate.

Usage

```
dedupeOrder(x)
```

Arguments

`x` VariableOrder, DatasetOrder, VariableGroup, or DatasetGroup

Value

`x` with duplicate entities removed.

See Also

`duplicates`, which when set to `FALSE` also calls this function.

delete	<i>Delete a Crunch object from the server</i>
--------	---

Description

These methods delete entities, notably Datasets and Variables within them, from the server. This action is permanent and cannot be undone, so it should not be done lightly. Consider instead using `archive` for datasets and `hide` for variables

Usage

```
## S4 method for signature 'CrunchDataset'
delete(x, confirm = requireConsent(), ...)
```

```
## S4 method for signature 'CrunchProject'
delete(x, confirm = requireConsent(), ...)
```

```
## S4 method for signature 'ShojiObject'
delete(x, ...)

## S4 method for signature 'ANY'
delete(x, ...)

## S4 method for signature 'CrunchTeam'
delete(x, confirm = requireConsent(), ...)

## S4 method for signature 'CrunchVariable'
delete(x, confirm = requireConsent(), ...)

## S4 method for signature 'VariableTuple'
delete(x, confirm = requireConsent(), ...)
```

Arguments

x	a Crunch object
confirm	logical: should the user be asked to confirm deletion. Option available for datasets and teams only. Default is TRUE if in an interactive session. You can avoid the confirmation prompt if you delete with(consent).
...	additional arguments, in the generic

See Also

[hide deleteDataset](#)

deleteDataset	<i>Delete a dataset from the dataset list</i>
---------------	---

Description

This function lets you delete a dataset without first loading it. If you have a dataset that somehow is corrupted and won't load, you can delete it this way.

Usage

```
deleteDataset(x, ...)
```

Arguments

x	The name (character) of a dataset, its (numeric) position in the return of listDatasets , or an object of class <code>CrunchDataset</code> . x can only be of length 1—this function is not vectorized (for your protection).
...	additional parameters (such as <code>confirm</code>) passed to <code>delete</code>

Details

The function also works on CrunchDataset objects, just like [delete](#), which may be useful if you have loaded another package that masks the delete method.

Value

(Invisibly) the API response from deleting the dataset

See Also

[delete](#)

deleteSubvariables *Delete subvariables from an array*

Description

This function conceals the dirty work in making this happen. The array gets unbound, the subvariables deleted, and then the remaining subvariable are rebound into a new array.

Usage

```
deleteSubvariables(variable, to.delete, confirm = requireConsent())
```

```
deleteSubvariable(variable, to.delete, confirm = requireConsent())
```

Arguments

variable	the array variable
to.delete	aliases (following <code>crunch.namekey.dataset</code>) or indices of variables to delete.
confirm	logical: should the user be asked to confirm deletion. Default is TRUE if in an interactive session. You can avoid the confirmation prompt if you delete with(consent).

Value

a new version of variable without the indicated subvariables

deleteVariables	<i>Delete Variables Within a Dataset</i>
-----------------	--

Description

Delete Variables Within a Dataset

Usage

```
deleteVariables(dataset, variables, confirm = requireConsent())
```

```
deleteVariable(dataset, variables, confirm = requireConsent())
```

Arguments

dataset	the Dataset to modify
variables	aliases (following <code>crunch.namekey.dataset</code>) or indices of variables to delete.
confirm	logical: should the user be asked to confirm deletion. Default is TRUE if in an interactive session. You can avoid the confirmation prompt if you delete with(consent).

Value

(invisibly) dataset with the specified variables deleted

See Also

[hide](#)

describe	<i>Name, alias, and description for Crunch objects</i>
----------	--

Description

Name, alias, and description for Crunch objects

Usage

```
## S4 method for signature 'CrunchDataset'
name(x)
```

```
## S4 replacement method for signature 'CrunchDataset'
name(x) <- value
```

```
## S4 method for signature 'CrunchDataset'
```

```
description(x)

## S4 replacement method for signature 'CrunchDataset'
description(x) <- value

## S4 method for signature 'CrunchDataset'
startDate(x)

## S4 replacement method for signature 'CrunchDataset'
startDate(x) <- value

## S4 method for signature 'CrunchDataset'
endDate(x)

## S4 replacement method for signature 'CrunchDataset'
endDate(x) <- value

## S4 method for signature 'CrunchDataset'
id(x)

## S4 method for signature 'CrunchDataset'
notes(x)

## S4 replacement method for signature 'CrunchDataset'
notes(x) <- value

## S4 method for signature 'ShojiObject'
name(x)

## S4 method for signature 'CrunchVariable'
name(x)

## S4 replacement method for signature 'CrunchVariable'
name(x) <- value

## S4 method for signature 'CrunchVariable'
description(x)

## S4 replacement method for signature 'CrunchVariable'
description(x) <- value

## S4 method for signature 'CrunchVariable'
alias(object)

## S4 replacement method for signature 'CrunchVariable'
alias(x) <- value

## S4 method for signature 'CrunchVariable'
```

```

notes(x)

## S4 replacement method for signature 'CrunchVariable'
notes(x) <- value

```

Arguments

x	a Dataset or Variable.
value	For the setters, a length-1 character vector to assign
object	Same as x but for the alias method, in order to match the generic from another package. Note that alias is only defined for Variables.

Value

Getters return the character object in the specified slot; setters return x duly modified.

See Also

[Categories describe-catalog](#)

describe-category	<i>Category attributes</i>
-------------------	----------------------------

Description

Category attributes

Usage

```

## S4 method for signature 'Category'
name(x)

## S4 replacement method for signature 'Category'
name(x) <- value

## S4 replacement method for signature ``NULL``
name(x) <- value

## S4 method for signature 'Category'
value(x)

## S4 replacement method for signature 'Category'
value(x) <- value

## S4 method for signature 'Category'
id(x)

## S4 method for signature 'Category'
is.selected(x)

```

Arguments

x a Category
 value For the setters, an appropriate value to set

Value

name returns character; value and id return numeric; value but not id may be NA; is.selected returns logical indicating whether this Category is a "selected" dichotomy. Setters return x duly modified.

See Also

[Categories dichotomize](#)

dichotomize	<i>Indicate how categories represent a dichotomized value</i>
-------------	---

Description

Multiple Response variables are essentially Categorical Arrays that have had a category or categories indicated as the "selected" value. These methods let you set that state.

Usage

```
## S4 method for signature 'Categories'
is.dichotomized(x)

## S4 method for signature 'Categories,numeric'
dichotomize(x, i)

## S4 method for signature 'Categories,logical'
dichotomize(x, i)

## S4 method for signature 'Categories,character'
dichotomize(x, i)

## S4 method for signature 'Categories'
undichotomize(x)

## S4 method for signature 'CategoricalVariable,ANY'
dichotomize(x, i)

## S4 method for signature 'CategoricalArrayVariable,ANY'
dichotomize(x, i)

## S4 method for signature 'CategoricalVariable'
```

```
undichotomize(x)

## S4 method for signature 'CategoricalArrayVariable'
undichotomize(x)
```

Arguments

x Categories or a Variable subclass that has Categories

i For the dichotomize methods, the numeric or logical indices of the categories to mark as "selected", or if character, the Category "names". Note that unlike some other categorical variable methods, numeric indices are positional, not with reference to category ids.

Details

dichotomize lets you specify which categories are "selected", while undichotomize strips that selection information. Dichotomize converts a Categorical Array to a Multiple Response, and undichotomize converts back.

Value

Categories or the Variable, (un)dichotomized accordingly

See Also

[describe-category](#)

dim-dataset

Dataset dimensions

Description

Dataset dimensions

Usage

```
## S4 method for signature 'CrunchDataset'
dim(x)
```

```
## S4 method for signature 'CrunchDataset'
ncol(x)
```

Arguments

x a Dataset

Value

integer vector of length 2, indicating the number of rows and non-hidden variables in the dataset. Array subvariables are excluded from the column count.

See Also

[dim](#)

dropRows	<i>Permanently delete rows from a dataset</i>
----------	---

Description

Permanently delete rows from a dataset

Usage

```
dropRows(dataset, expr)
```

Arguments

dataset	a CrunchDataset
expr	a CrunchLogicalExpr

Value

dataset without the rows indicated by expr

See Also

[exclusion](#) for a non-destructive way to suppress rows

Examples

```
## Not run:  
ds <- dropRows(ds, ds$gender == "Male")  
  
## End(Not run)
```

duplicated	<i>"duplicated" method for Crunch objects</i>
------------	---

Description

"duplicated" method for Crunch objects

Usage

```
## S4 method for signature 'CrunchVariable'
duplicated(x, incomparables = FALSE, ...)
```

```
## S4 method for signature 'CrunchExpr'
duplicated(x, incomparables = FALSE, ...)
```

Arguments

x	CrunchVariable or CrunchExpr
incomparables	Ignored
...	Ignored

Value

A CrunchLogicalExpr that evaluates TRUE for all repeated entries after the first occurrence of a value.

See Also

[duplicated](#)

embedCrunchBox	<i>Get HTML for embedding a CrunchBox</i>
----------------	---

Description

[crunchBox](#) returns a URL to the box data that it generates, but in order to view it in a CrunchBox or to embed it on a website, you'll need to translate that to the Box's public URL and wrap it in some HTML.

Usage

```
embedCrunchBox(box, title = NULL, logo = NULL, ...)
```

Arguments

box	character URL of the box data, as returned by crunchBox
title	character title for the Box, to appear above the iframe. Default is NULL, meaning no title shown
logo	character URL of a logo to show instead of a title. Default is NULL, meaning no logo shown. If both logo and title are provided, only logo will be shown. Note also that logo must be a URL of an image hosted somewhere—it cannot be a path to a local file.
...	Additional arguments, not currently used.

Value

Prints the HTML markup to the screen and also returns it invisibly.

See Also

[crunchBox](#)

Examples

```
## Not run:
box <- crunchBox(ds)
embedCrunchBox(box, logo="//myco.example/img/logo_200px.png")

## End(Not run)
```

entity, CrunchProject-method
Methods for ShojiTuples

Description

ShojiTuples are objects extracted from ShojiCatalogs. They are internally used.

Usage

```
## S4 method for signature 'CrunchProject'
entity(x)

## S4 method for signature 'ShojiEntity'
x$name

## S4 replacement method for signature 'ShojiEntity'
x$name <- value

## S4 method for signature 'ShojiEntity, ANY'
```



```
x[[i]]

## S4 replacement method for signature 'ShojiEntity,ANY,ANY,ANY'
x[[i]] <- value

## S4 method for signature 'ShojiTuple'
refresh(x)

## S4 method for signature 'ShojiTuple'
x$name

## S4 replacement method for signature 'ShojiTuple'
x$name <- value

## S4 method for signature 'ShojiTuple,ANY'
x[[i]]

## S4 replacement method for signature 'ShojiTuple,ANY,ANY,ANY'
x[[i]] <- value

## S4 method for signature 'ShojiTuple'
self(x)

## S4 method for signature 'VariableTuple'
entity(x)

## S4 method for signature 'CrunchVariable'
entity(x)

## S4 method for signature 'DatasetTuple'
entity(x)

## S4 method for signature 'ShojiTuple'
delete(x, ...)

## S4 method for signature 'DatasetTuple'
delete(x, confirm = requireConsent(), ...)

## S4 method for signature 'ShojiTuple'
name(x)

## S4 replacement method for signature 'ShojiTuple'
name(x) <- value

## S4 method for signature 'ShojiTuple'
type(x)
```

Arguments

x	a Tuple
name	a Tuple slot to get or set
value	What to set in a given slot
i	In [], a Tuple slot to get
...	additional arguments to [], ignored
confirm	For delete, whether confirmation is required. See delete .

exclusion	<i>View and set exclusion filters</i>
-----------	---------------------------------------

Description

Exclusion filters express logic that defines a set of rows that should be dropped from the dataset. The rows aren't permanently deleted—you can recover them at any time by removing the exclusion filter—but they are omitted from all views and calculations, as if they had been deleted.

Usage

```
exclusion(x)

exclusion(x) <- value
```

Arguments

x	a Dataset
value	an object of class <code>CrunchLogicalExpr</code> , or <code>NULL</code>

Details

Note that exclusion filters work opposite from how "normal" filters work. That is, a regular filter expression defines the subset of rows to operate on: it says "keep these rows." An exclusion filter defines which rows to omit. Applying a filter expression as a query filter will have the opposite effect if applied as an exclusion. Indeed, applying it as both query filter and exclusion at the same time will result in 0 rows.

Value

`exclusion` returns a `CrunchFilter` if there is one, else `NULL`. The setter returns `x` with the filter set.

exportDataset	<i>Export a dataset to a file</i>
---------------	-----------------------------------

Description

Export a dataset to a file

Usage

```
exportDataset(dataset, file, format = c("csv", "spss"),
  categorical = c("name", "id"), ...)
```

```
## S4 method for signature 'CrunchDataset'
write.csv(...)
```

Arguments

dataset	CrunchDataset
file	character local filename to write to
format	character export format: currently supported values are "csv" and "spss".
categorical	character: export categorical values to CSV as category "name" (default) or "id". Ignored by the SPSS exporter.
...	additional arguments, currently ignored

Value

Invisibly, file.

expressions	<i>Construct Crunch Expressions</i>
-------------	-------------------------------------

Description

Crunch Expressions, i.e. `CrunchExpr` and `CrunchLogicalExpr`, encapsulate derivations of Crunch variables, which are only evaluated when passed to a function like `as.vector`. They allow you to compose functional expressions of variables and evaluate them against the server only when appropriate.

Usage

```
## S4 method for signature 'CrunchExpr'  
!x  
  
## S4 method for signature 'CategoricalVariable,character'  
x %in% table  
  
## S4 method for signature 'CategoricalVariable,factor'  
x %in% table  
  
## S4 method for signature 'TextVariable,character'  
x %in% table  
  
## S4 method for signature 'NumericVariable,numeric'  
x %in% table  
  
## S4 method for signature 'DatetimeVariable,Date'  
x %in% table  
  
## S4 method for signature 'DatetimeVariable,POSIXt'  
x %in% table  
  
## S4 method for signature 'DatetimeVariable,character'  
x %in% table  
  
## S4 method for signature 'CategoricalVariable,numeric'  
x %in% table  
  
## S4 method for signature 'CategoricalVariable,character'  
e1 == e2  
  
## S4 method for signature 'CategoricalVariable,factor'  
e1 == e2  
  
## S4 method for signature 'CategoricalVariable,character'  
e1 != e2  
  
## S4 method for signature 'CategoricalVariable,factor'  
e1 != e2  
  
## S4 method for signature 'CrunchVariable'  
is.na(x)  
  
bin(x)  
  
rollup(x, resolution = rollupResolution(x))
```

Arguments

x	an input
table	For %in%. See match
e1	an input
e2	an input
resolution	For rollup. Either NULL or a character in c("Y", "Q", "M", "W", "D", "h", "m", "s", "ms") indicating the unit of time at which a Datetime variable should be aggregated. If NULL, the server will determine an appropriate resolution based on the range of the data.

Value

Most functions return a CrunchExpr or CrunchLogicalExpr. as .vector returns an R vector.

filter-catalog	<i>Filter entities for a dataset</i>
----------------	--------------------------------------

Description

Filter entities for a dataset

Usage

```
## S4 method for signature 'CrunchDataset'
filters(x)

## S4 replacement method for signature 'CrunchDataset'
filters(x) <- value
```

Arguments

x	a CrunchDataset
value	for the assignment method, a FilterCatalog

Value

an object of class FilterCatalog containing references to Filter entities usable in the web application. (Setter returns the Dataset.)

filter-methods	<i>View and modify Filter entity attributes</i>
----------------	---

Description

View and modify Filter entity attributes

Usage

```
## S4 method for signature 'CrunchFilter'
is.public(x)

## S4 replacement method for signature 'CrunchFilter'
is.public(x) <- value
```

Arguments

x	a CrunchFilter
value	an attribute to set

Value

For `is.public`, a logical value for whether the filter is flagged as shared with all dataset viewers. (Its setter thus takes a logical value as well.)

flattenOrder	<i>Remove nesting of groups within an order/group</i>
--------------	---

Description

This function reduces a potentially nested order to its flattened representation, containing no nested groups. Entities are ordered in the result by their first appearance in the order object passed as input.

Usage

```
flattenOrder(x)
```

Arguments

x	VariableOrder, DatasetOrder, VariableGroup, or DatasetGroup; or a Crunch-Dataset or catalog that has an <code>ordering</code> property.
---	---

Value

x, or its order resource, flattened.

 flipArrays

Rearrange array subvariables into other configurations

Description

Sometimes it is useful to group subvariables across arrays in order to compare them more easily. This function generates a set of derived views of common subvariables across arrays. Because they are derived, they share data with the underlying array variables, and they are thus automatically updated when new data is appended.

Usage

```
flipArrays(variables, suffix = "", flipped)
```

Arguments

variables	List of variables, variable catalog, or dataset subset containing the categorical array or multiple response variables you want to rearrange.
suffix	character string to append to the new variable names. Make it "" if you don't want it to append anything.

Value

A list of derived VariableDefinitions, one per unique subvariable name across all variables. Each variable (in variables) that contains this subvariable will appear as a subvariable in these new derived array definitions. Use addVariables to add these to your dataset.

Examples

```
## Not run:
ds <- addVariables(ds, flipArrays(ds[c("petloc", "petloc2")], suffix=" ", rearranged))

## End(Not run)
```

 forkDataset

Create a fork of a dataset

Description

As with many other version control systems, in Crunch you can fork a dataset's revision history, effectively making a copy on which you can work independently of the original dataset. You can then merge those change back to the original dataset or keep working independently.

Usage

```
forkDataset(dataset, name = defaultForkName(dataset), draft = FALSE, ...)
```

Arguments

dataset	The CrunchDataset to fork
name	character name to give the fork. If omitted, one will be provided for you
draft	logical: Should the dataset be a draft, available only to editors? Default is FALSE.
...	Additional dataset metadata

Value

The new fork, a CrunchDataset.

getAccountUserCatalog *Find all users on your account*

Description

Find all users on your account

Usage

```
getAccountUserCatalog(x = shojiURL(getAccount(), "catalogs", "users"))
```

Arguments

x	URL of the user catalog. Default is the right thing; you shouldn't specify one
---	--

Value

a UserCatalog

getTeams *Retrieve all teams you're a member of*

Description

Retrieve all teams you're a member of

Usage

```
getTeams()
```

Value

A TeamCatalog. Extract an individual team by name. Create a team by assigning in with a new name.

See Also

[teams](#)

grouped	<i>Get un(grouped) OrderGroups</i>
---------	------------------------------------

Description

"ungrouped" is a magic OrderGroup that contains all entities not found in groups at a given level of nesting.

Usage

```
grouped(order.obj)
```

```
ungrouped(order.obj)
```

Arguments

order.obj an subclass of ShojiOrder or OrderGroup

Value

For grouped(), an Order/Group, respectively, with "ungrouped" omitted. For ungrouped(), an OrderGroup subclass.

See Also

[VariableOrder](#)

hiddenVariables	<i>Show the names of hidden variables within the dataset</i>
-----------------	--

Description

Show the names of hidden variables within the dataset

Usage

```
hiddenVariables(dataset, key = namekey(dataset))
```

Arguments

dataset the Dataset

key the Variable attribute to return. Default is "alias", following `getOption("crunch.namekey.dataset")`.

Value

a vector of the names of Variables marked as hidden.

hide	<i>Hide and Unhide Variables</i>
------	----------------------------------

Description

Hide and Unhide Variables

Usage

```
## S4 method for signature 'CrunchVariable'  
hide(x)
```

```
## S4 method for signature 'VariableCatalog'  
hide(x)
```

```
## S4 method for signature 'CrunchVariable'  
unhide(x)
```

```
## S4 method for signature 'VariableCatalog'  
unhide(x)
```

Arguments

x a Variable or subset of a VariableCatalog to hide or unhide

Value

(invisibly) the Variable or VariableCatalog, hidden or unhidden

hideVariables	<i>Hide and Unhide Variables Within a Dataset</i>
---------------	---

Description

Hide and Unhide Variables Within a Dataset

Usage

```
hideVariables(dataset, variables)
```

```
hiddenVariables(x) <- value
```

```
unhideVariables(dataset, variables)
```

Arguments

dataset	the Dataset to modify
variables	names or indices of variables to (un)hide
x	same as dataset, for 'hiddenVariables<-'
value	same as variables, for 'hiddenVariables<-'

Value

(invisibly) dataset with the specified variables (un)hidden

See Also

[hide](#)

http-methods

HTTP methods for communicating with the Crunch API

Description

HTTP methods for communicating with the Crunch API

Usage

crGET(...)

crPUT(...)

crPATCH(...)

crPOST(...)

crDELETE(...)

Arguments

... see [crunchAPI](#) for details. url is the first named argument and is required; body is also required for PUT, PATCH, and POST.

Value

Depends on the response status of the HTTP request and any custom handlers.

index	<i>Get the body of a Catalog</i>
-------	----------------------------------

Description

The core of Catalog data is in its "index". These methods get and set that slot.

Usage

```
## S4 method for signature 'ShojiCatalog'
index(x)

## S4 replacement method for signature 'ShojiCatalog'
index(x) <- value
```

Arguments

x	a Catalog (VariableCatalog, Subvariables, or similar object)
value	For the setters, an appropriate-length list to assign

Value

Getters return the list object in the "index" slot; setters return x duly modified.

is-na-categories	<i>is.na for Categories</i>
------------------	-----------------------------

Description

is.na for Categories

Usage

```
## S4 method for signature 'Categories'
is.na(x)

## S4 replacement method for signature 'Categories,character'
is.na(x) <- value

## S4 replacement method for signature 'Categories,logical'
is.na(x) <- value

## S4 method for signature 'Category'
is.na(x)

## S4 replacement method for signature 'Category,logical'
is.na(x) <- value
```

Arguments

x	Categories or a single Category
value	To change the missingness of categories, supply either (1) a logical vector of equal length of the categories (or length 1 for the Category method), or (2) the names of the categories to mark as missing. If supplying the latter, any categories already indicated as missing will remain missing.

Value

Getters return logical, a named vector in the case of the Categories method; setters return x duly modified.

is.archived, DatasetCatalog-method

Get and set "archived" and "published" status of a dataset

Description

"Archived" datasets are excluded from some views. "Draft" datasets are visible only to editors. "Published" is the inverse of "Draft", i.e. `is.draft(x)` entails `!is.published(x)`. These properties are accessed and set with the "is" methods. The verb functions `archive` and `publish` are alternate versions of the setters (at least in the TRUE direction).

Usage

```
## S4 method for signature 'DatasetCatalog'
is.archived(x)

## S4 method for signature 'DatasetCatalog'
is.draft(x)

## S4 method for signature 'DatasetCatalog'
is.published(x)

## S4 replacement method for signature 'DatasetCatalog,logical'
is.archived(x) <- value

## S4 replacement method for signature 'DatasetCatalog,logical'
is.draft(x) <- value

## S4 replacement method for signature 'DatasetCatalog,logical'
is.published(x) <- value

## S4 method for signature 'CrunchDataset'
is.archived(x)
```

```

## S4 method for signature 'CrunchDataset'
is.draft(x)

## S4 method for signature 'CrunchDataset'
is.published(x)

## S4 replacement method for signature 'CrunchDataset,logical'
is.archived(x) <- value

archive(x)

## S4 replacement method for signature 'CrunchDataset,logical'
is.draft(x) <- value

## S4 replacement method for signature 'CrunchDataset,logical'
is.published(x) <- value

publish(x)

```

Arguments

x	CrunchDataset
value	logical

Value

For the getters, the logical value of whether the dataset is archived, in draft mode, or published, where draft and published are inverses. The setters return the dataset.

is.dataset	<i>Is it?</i>
------------	---------------

Description

Is it?

Usage

```

is.dataset(x)

is.shoji(x)

is.variable(x)

is.Numeric(x)

is.Categorical(x)

```

```
is.Text(x)
is.Datetime(x)
is.Multiple(x)
is.MR(x)
is.MultipleResponse(x)
is.CA(x)
is.Array(x)
is.CategoricalArray(x)
```

Arguments

x an object

Value

logical

is.editor *Read and set edit privileges*

Description

Read and set edit privileges

Usage

```
## S4 method for signature 'MemberCatalog'
is.editor(x)

## S4 replacement method for signature 'MemberCatalog,logical'
is.editor(x) <- value

## S4 method for signature 'PermissionCatalog'
is.editor(x)

## S4 method for signature 'PermissionTuple'
is.editor(x)
```

Arguments

x	PermissionCatalog or MemberCatalog
value	For the setter, logical: should the indicated users be allowed to edit the associated object?

Value

is.editor returns a logical vector corresponding to whether the users in the catalog can edit or not.
is.editor<- returns the catalog, modified.

joinDatasets	<i>Add columns from one dataset to another, joining on a key</i>
--------------	--

Description

As [merge](#) does for data.frames, this function takes two datasets, matches rows based on a specified key variable, and adds columns from one to the other.

Usage

```
joinDatasets(x, y, by = intersect(names(x), names(y)), by.x = by,
  by.y = by, all = FALSE, all.x = TRUE, all.y = FALSE, copy = TRUE)

extendDataset(x, y, by = intersect(names(x), names(y)), by.x = by,
  by.y = by, all = FALSE, all.x = TRUE, all.y = FALSE, ...)

## S3 method for class 'CrunchDataset'
merge(x, y, by = intersect(names(x), names(y)),
  by.x = by, by.y = by, all = FALSE, all.x = TRUE, all.y = FALSE, ...)
```

Arguments

x	CrunchDataset to add data to
y	CrunchDataset to copy data from. May be filtered by rows and/or columns.
by	character, optional shortcut for specifying by.x and by.y by alias if the key variables have the same alias in both datasets.
by.x	CrunchVariable in x on which to join, or the alias (following crunch.namekey.dataset) of a variable. Must be type numeric or text and have all unique, non-missing values.
by.y	CrunchVariable in y on which to join, or the alias (following crunch.namekey.dataset) of a variable. Must be type numeric or text and have all unique, non-missing values.
all	logical: should all rows in x and y be kept, i.e. a "full outer" join? Only FALSE is currently supported.

all.x	logical: should all rows in x be kept, i.e. a "left outer" join? Only TRUE is currently supported.
all.y	logical: should all rows in y be kept, i.e. a "right outer" join? Only FALSE is currently supported.
copy	logical: make a virtual or materialized join. Default is TRUE, which means materialized. Virtual joins are experimental and not advised.
...	additional arguments, ignored

Value

x extended by the columns of y, matched on the "by" variables.

listDatasets	<i>Show the names of all Crunch datasets</i>
--------------	--

Description

Show the names of all Crunch datasets

Usage

```
listDatasets(kind = c("active", "all", "archived"), project = NULL,
             refresh = FALSE)
```

Arguments

kind	character specifying whether to look in active, archived, or all datasets. Default is "active", i.e. non-archived.
project	CrunchProject entity, character name of a project, or NULL, the default. If a Project entity or reference is supplied, the function will display datasets from that Project's datasets. If NULL, the primary dataset catalog for the user will be used.
refresh	logical: should the function check the Crunch API for new datasets? Default is FALSE.

Value

Character vector of dataset names, each of which would be a valid input for [loadDataset](#)

loadDataset	<i>Load a Crunch Dataset</i>
-------------	------------------------------

Description

Load a Crunch Dataset

Usage

```
loadDataset(dataset, kind = c("active", "all", "archived"), project = NULL,
            refresh = FALSE)
```

Arguments

dataset	character, the name of a Crunch dataset you have access to. Or, a DatasetTuple.
kind	character specifying whether to look in active, archived, or all datasets. Default is "active", i.e. non-archived.
project	CrunchProject entity, character name of a project, or NULL, the default. If a Project entity or reference is supplied, the function will display datasets from that Project's datasets. If NULL, the primary dataset catalog for the user will be used.
refresh	logical: should the function check the Crunch API for new datasets? Default is FALSE.

Value

An object of class CrunchDataset

locateEntity	<i>Find an entity in an order object</i>
--------------	--

Description

Find an entity in an order object

Usage

```
locateEntity(x, ord)
```

Arguments

x	Variable or Dataset, depending on the type of order, or URL for it
ord	ShojiOrder (VariableOrder or DatasetOrder)

Value

If `x` is found in `ord`, a character vector of group names that provide the "path" to the entity. The length of the vector corresponds to the depth of nesting. If not found, NA is returned

lock	<i>Lock and unlock a dataset for editing</i>
------	--

Description

Crunch allows a single active editor. If you have edit privileges but are not currently editing the dataset, you must unlock the dataset before making changes. You may then lock the dataset when you're done editing.

Usage

```
lock(dataset)

unlock(dataset)
```

Arguments

dataset a CrunchDataset

Value

dataset, invisibly, after having set the current editor.

login	<i>Authenticate with the Crunch API</i>
-------	---

Description

Note that you can store your Crunch account info in your .Rprofile under "crunch.email" and "crunch.pw" for convenience. If you do so, you can simply `login()` to authenticate. For running batch jobs, this could be particularly useful. However, be warned that storing your password in a plain text file such as .Rprofile is a security risk (though perhaps less so than in every .R script you write), and we cannot officially recommend that you do so.

Usage

```
login(email = getOption("crunch.email"), password = getOption("crunch.pw"),
      ...)
```

Arguments

email	the email address associated with the user's Crunch account
password	the password associated with the user's Crunch account
...	additional parameters passed in the authentication. Not currently supported by the Crunch API.

Details

If a password is not supplied (or, if no arguments are supplied and only the `crunch.email` is specified in `.Rprofile`), and you are in an interactive session, you will be prompted to enter your password. At present, this is the most secure practice as your password is not stored locally.

logout	<i>Kill the active Crunch session</i>
--------	---------------------------------------

Description

Kill the active Crunch session

Usage

```
logout()
```

makeArray	<i>Make a Categorical Array or Multiple Response variable</i>
-----------	---

Description

Make a Categorical Array or Multiple Response variable

Usage

```
makeArray(subvariables, name, ...)
```

```
makeMR(subvariables, name, selections, ...)
```

```
deriveArray(subvariables, name, ...)
```

Arguments

subvariables	a list of Variable objects to bind together, or a Dataset object containing only the Variables to bind (as in <code>from</code> subsetting a Dataset)
name	character, the name that the new Categorical Array variable should have. Required.
...	Optional additional attributes to set on the new variable.
selections	character, for <code>makeMR</code> , the names of the categories to mark as the dichotomous selections. Required for <code>makeMR</code> ; ignored in <code>makeArray</code> .

Value

A VariableDefinition that when added to a Dataset will create the categorical-array or multiple-response variable. `deriveArray` will make a derived array expression, while `makeArray` and `makeMR` return an expression that "binds" variables together, removing them from independent existence.

me	<i>My user entity</i>
----	-----------------------

Description

My user entity

Usage

`me()`

Value

A UserEntity that corresponds to you, the authenticated user

<code>members<- ,CrunchTeam,MemberCatalog-method</code>	<i>Teams</i>
--	--------------

Description

Teams contain users and datasets. You can share a dataset with a group of users by sharing the dataset with a team. You can also share a bunch of datasets with a user all at once by adding them to a team that has those datasets.

Usage

```
## S4 replacement method for signature 'CrunchTeam,MemberCatalog'
members(x) <- value
```

```
## S4 replacement method for signature 'CrunchTeam,character'
members(x) <- value
```

```
## S4 method for signature 'CrunchProject'
members(x)
```

```
## S4 replacement method for signature 'CrunchProject,MemberCatalog'
members(x) <- value
```

```
## S4 replacement method for signature 'CrunchProject,character'
members(x) <- value
```

```
## S4 method for signature 'CrunchTeam'
members(x)
```

Arguments

x a CrunchTeam
value for members<-, a character vector of emails or URLs of users to add to the team.

Details

These methods allow you to work with teams. Find your teams with the [getTeams](#) function, which returns your TeamCatalog. Extract an individual team by name. Create a team by assigning in with a new name, with the assignment value a list, either empty (to just create a team with that name), or with a "members" element, containing emails or URLs of users to add to the team. Users can be added later with the members<- method.

Value

members returns a MemberCatalog, which has references to the users that are members of the team.
members<- returns x with the given users added to the members catalog.

See Also

[getTeams](#)

mergeFork

Merge changes to a dataset from a fork

Description

Merge changes to a dataset from a fork

Usage

```
mergeFork(dataset, fork, autorollback = TRUE, force = FALSE)
```

Arguments

dataset The CrunchDataset to merge to
fork The CrunchDataset, perhaps forked from dataset, that is to be merged in.
autorollback logical If the merge fails, should dataset be restored to its state prior to the merge, or should it be left in its partially merged state for debugging and manual fixing? Default is TRUE, i.e. the former.

force logical Attempt to push through merge conflicts by dropping all changes to dataset since fork diverged from and take only the changes from fork? Default is FALSE, and it is recommended only to use force=TRUE after first attempting and failing to merge without forcing.

Value

dataset with changes from fork merged to it.

moveToGroup	<i>Move entities to a group</i>
-------------	---------------------------------

Description

The function has two versions: a regular function and a setter. They do the same thing, but the setter probably results in less verbose code for you.

Usage

```
moveToGroup(x, value)
moveToGroup(x) <- value
```

Arguments

x	VariableGroup
value	Variable, VariableCatalog subset, or Dataset subset

Value

x with the entities in value appended to it. If the containing order object has duplicates=FALSE, the entities will be "moved" to this group. Otherwise, their references will be copied to the group.

Examples

```
## Not run:
moveToGroup(ordering(ds)[["Demographics"]]) <- ds[c("gender", "age")]

## End(Not run)
```

na.omit-categories	<i>Omit missing categories</i>
--------------------	--------------------------------

Description

Omit missing categories

Usage

```
## S4 method for signature 'Categories'
na.omit(object, ...)
```

Arguments

object	Categories
...	additional arguments, ignored

Value

object with any categories that have missing: TRUE excluded

names, BatchCatalog-method

Get and set names, aliases on Catalog-type objects

Description

These methods let you get and set names and aliases for variables in a Dataset's catalog, or within [Subvariables](#) in an array variable. They work like the base R names methods.

Usage

```
## S4 method for signature 'BatchCatalog'
names(x)

## S4 method for signature 'CrunchDataset'
names(x)

## S4 method for signature 'ShojiCatalog'
names(x)

## S4 replacement method for signature 'ShojiCatalog'
names(x) <- value

## S4 method for signature 'ShojiCatalog'
```



```
emails(x)

## S4 method for signature 'Subvariables'
aliases(x)

## S4 replacement method for signature 'Subvariables'
aliases(x) <- value

## S4 method for signature 'CategoricalArrayVariable'
names(x)

## S4 method for signature 'VariableCatalog'
aliases(x)

## S4 replacement method for signature 'VariableCatalog'
aliases(x) <- value

## S4 method for signature 'VariableCatalog'
notes(x)

## S4 replacement method for signature 'VariableCatalog'
notes(x) <- value

## S4 method for signature 'VariableCatalog'
descriptions(x)

## S4 replacement method for signature 'VariableCatalog'
descriptions(x) <- value

## S4 method for signature 'VariableCatalog'
types(x)

## S4 method for signature 'VariableCatalog'
ids(x)

## S4 method for signature 'VersionCatalog'
names(x)

## S4 method for signature 'VersionCatalog'
descriptions(x)

## S4 method for signature 'VersionCatalog'
timestamps(x)
```

Arguments

x	a VariableCatalog, Subvariables, or similar object
value	For the setters, an appropriate-length character vector to assign

Details

Note that the `names` method on a `Dataset` returns the aliases of its variables by default. This is controlled by `getOption("crunch.namekey.dataset")`, which is "alias" by default. Set `options(crunch.namekey.dataset=)` if you wish to use variable names. See the vignette on variables for more information.

Value

Getters return the character object in the specified slot; setters return `x` duly modified.

See Also

[Subvariables Categories names](#) `vignette("variables", package="crunch")`

newDataset

Upload a data.frame to Crunch to make a new dataset

Description

Upload a `data.frame` to Crunch to make a new dataset

Usage

```
newDataset(x, name = deparse(substitute(x))[1], ...)
```

Arguments

<code>x</code>	a <code>data.frame</code> or other rectangular R object
<code>name</code>	character, the name to give the new Crunch dataset. Default is the name of the R object passed in <code>x</code>
<code>...</code>	additional arguments passed to createDataset

Value

If successful, an object of class `CrunchDataset`.

newDatasetByColumn *Upload a data.frame column-by-column to make a new dataset*

Description

Use this version if you have lots of variables, under 1M rows, perhaps backed by ff or other memory-mapped files, and time to kill.

Usage

```
newDatasetByColumn(x, name = deparse(substitute(x))[1], ...)
```

Arguments

x a data.frame or other rectangular R object
name character, the name to give the new Crunch dataset. Default is the name of the R object passed in x
... additional arguments passed to [createDataset](#)

Value

If successful, an object of class CrunchDataset.

See Also

[newDataset](#) [newDatasetByCSV](#)

newDatasetByCSV *Upload a data.frame to Crunch to make a new dataset*

Description

This function uses the CSV+JSON import format, which is faster and more effective for certain dataset sizes and shapes than [newDatasetByColumn](#).

Usage

```
newDatasetByCSV(x, name = deparse(substitute(x))[1], ...)
```

Arguments

x a data.frame or other rectangular R object
name character, the name to give the new Crunch dataset. Default is the name of the R object passed in x
... additional arguments passed to [createDataset](#)

Value

If successful, an object of class `CrunchDataset`.

See Also

[newDataset](#) [newDatasetByColumn](#)

`newDatasetFromFile` *Upload a file to Crunch to make a new dataset*

Description

Use this import method if you have an SPSS data file. Reading such a file into R as a `data.frame` will result in lost metadata. You can just send it directly to Crunch and let the server process it.

Usage

```
newDatasetFromFile(file, name = basename(file), ...)
```

Arguments

<code>file</code>	character, the path to a file to upload. This should either be a <code>.csv</code> or <code>.sav</code> (SPSS) file.
<code>name</code>	character, the name to give the new Crunch dataset. Default is the file name
<code>...</code>	additional arguments passed to createDataset

Value

On success, an object of class `CrunchDataset`.

`ordering` *Get and set VariableOrder*

Description

The ordering methods allow you to get and set a [VariableOrder](#) on a [CrunchDataset](#) or on the [VariableCatalog](#) that the dataset contains.

Usage

```
## S4 method for signature 'CrunchDataset'
ordering(x)

## S4 replacement method for signature 'CrunchDataset'
ordering(x) <- value

## S4 method for signature 'VariableCatalog'
ordering(x)

## S4 replacement method for signature 'VariableCatalog'
ordering(x) <- value

## S4 method for signature 'DatasetCatalog'
ordering(x)

## S4 replacement method for signature 'DatasetCatalog'
ordering(x) <- value
```

Arguments

x a VariableCatalog or CrunchDataset
value a valid VariableOrder object

Value

ordering returns a VariableOrder object, while ordering<- sets the VariableOrder in value on x

owners	<i>See who owns these datasets</i>
--------	------------------------------------

Description

See who owns these datasets

Usage

```
owners(x)

ownerNames(x)
```

Arguments

x DatasetCatalog

Value

For owners, the URLs of the users or projects that own these datasets. For ownerNames, their names.

permissions	<i>See who has access to this dataset</i>
-------------	---

Description

See who has access to this dataset

Usage

```
## S4 method for signature 'CrunchDataset'
permissions(x)
```

Arguments

x	CrunchDataset
---	---------------

Value

A PermissionCatalog containing information on the users and teams that have access to this dataset.

preCrunchBoxCheck	<i>Summarize any characteristics of a dataset that could make for an awkward CrunchBox</i>
-------------------	--

Description

Summarize any characteristics of a dataset that could make for an awkward CrunchBox

Usage

```
preCrunchBoxCheck(dataset)
```

Arguments

dataset	CrunchDataset, potentially subsetted on variables
---------	---

Value

Invisibly, the dataset. Called for side-effect of printing things.

See Also

[crunchBox](#)

project-icon	<i>A project's icon</i>
--------------	-------------------------

Description

A project's icon

Usage

```
icon(x)
```

```
icon(x) <- value
```

Arguments

x	a CrunchProject
value	character file path of the icon image file to set

Value

The URL of the project's icon. The setter returns the project after having uploaded the specified file as the new icon.

projects	<i>Get the project catalog</i>
----------	--------------------------------

Description

Get the project catalog

Usage

```
projects(x = getAPIRoot())
```

Arguments

x	a ShojiObject that has a project catalog associated. If omitted, the default value for x means that you will load the user's primary project catalog. (Currently, there are no other project catalogs to load.)
---	---

Value

An object of class ProjectCatalog.

Examples

```
## Not run:
myprojects <- projects()
proj <- myprojects[["Project name"]]

## End(Not run)
```

refresh	<i>Get a fresh copy from the server</i>
---------	---

Description

Crunch objects usually keep themselves in sync with the server when you manipulate them, but sometimes they can drift. Maybe someone else has modified the dataset you're working on, or maybe you have modified a variable outside of the context of its dataset. `refresh()` allows you to get back in sync.

Usage

```
## S4 method for signature 'CrunchDataset'
refresh(x)

## S4 method for signature 'ShojiObject'
refresh(x)

## S4 method for signature 'CrunchVariable'
refresh(x)
```

Arguments

x pretty much any Crunch object

Value

a new version of x

removeEmptyGroups	<i>Remove OrderGroups with no entities</i>
-------------------	--

Description

This function recurses through a `ShojiOrder/OrderGroup` and removes any groups that contain no entities.

Usage

```
removeEmptyGroups(x)
```

Arguments

x VariableOrder, DatasetOrder, VariableGroup, or DatasetGroup

Value

x with empty groups removed.

restoreVersion	<i>Restore a dataset to a previously saved version</i>
----------------	--

Description

Restore a dataset to a previously saved version

Usage

```
restoreVersion(dataset, version)
```

Arguments

dataset a CrunchDataset

version either the name ("description") of the version to restore to or the integer index of the version, as given by versions(dataset)

Value

dataset, rolled back to version.

See Also

[versions saveVersion](#)

saveVersion	<i>Create a new saved version</i>
-------------	-----------------------------------

Description

Create a new saved version

Usage

```
saveVersion(dataset, description = paste("Version", length(versions(dataset))
+ 1))
```

Arguments

dataset	a CrunchDataset
description	character name to give the saved version, as in a commit message. You are encouraged, though not strictly required, to give versions unique descriptions.

Value

invisibly, the URL of the newly created version

See Also

[versions](#) [restoreVersion](#)

self	<i>Get the URL of this object</i>
------	-----------------------------------

Description

Get the URL of this object

Usage

```
## S4 method for signature 'ShojiObject'
self(x)

## S4 method for signature 'CrunchVariable'
self(x)
```

Arguments

x	a Crunch object
---	-----------------

Value

the URL for x

session	<i>Get various catalogs for your Crunch session</i>
---------	---

Description

Get various catalogs for your Crunch session

Usage

```
session()
```

Value

A Session object. Access dataset and project catalogs from it.

Examples

```
## Not run:  
cr <- session()  
cr$databases  
cr$projects  
  
## End(Not run)
```

session-extract	<i>Extract catalogs from a Session object</i>
-----------------	---

Description

Extract catalogs from a Session object

Usage

```
## S4 method for signature 'Session,ANY'  
x[[i, ..., drop = FALSE]]  
  
## S4 method for signature 'Session'  
x$name  
  
## S4 replacement method for signature 'Session,ANY,ANY,ANY'  
x[[i]] <- value  
  
## S4 replacement method for signature 'Session'  
x$name <- value
```

Arguments

x	a Session object
i	which catalog to load. Supported values are "datasets" and "projects"
...	additional arguments, ignored.
drop	Invalid
name	for \$, the same as i for []
value	For updating, an object of the appropriate class and size to insert. In practice value is ignored; Session objects hold no state and it is assumed that any state modification on the server happens in other methods.

Value

The requested catalog when extracting; a Session object if assigning.

See Also

[session](#)

settings

View and modify dataset-level settings

Description

These methods allow access and control over dataset settings. Currently supported settings include 'viewers_can_export' and 'viewers_can_change_weight', which govern specific authorizations for users with view-only access to this dataset; and 'weight', which is the default weight variable for the dataset, the one that will be set for newly shared users and the one that viewers will have always on if they are not authorized to change weights. Additional settings will be added in the future. See <http://docs.crunch.io/#fragments>, under 'Settings', for an up-to-date list of settings supported throughout the Crunch system. Clients may also provide and use custom settings if they choose.

Usage

```
settings(x)
```

```
settings(x) <- value
```

Arguments

x	CrunchDataset
value	A settings object (ShojiEntity), for the setter

Value

The getter returns a settings object (ShojiEntity). The setter returns the dataset (x).

Examples

```
## Not run:
settings(ds)
settings(ds)$viewers_can_export <- TRUE

## End(Not run)
```

share	<i>Share a dataset</i>
-------	------------------------

Description

Share a dataset

Usage

```
share(dataset, users, edit = FALSE, notify = TRUE)
```

Arguments

dataset	a CrunchDataset
users	character: email address(es) or URLs of the users or teams with whom to share the dataset. If there is no Crunch user associated with an email, an invitation will be sent.
edit	logical: should the specified user(s) be given edit privileges on the dataset? Default is FALSE. edit can be a single value or, if inviting multiple users, a vector of logical values of equal length of the number of emails given.
notify	logical: should users who are getting new privileges on this dataset be sent an email informing them of this fact? Default is TRUE.

Value

Invisibly, the dataset.

See Also

[unshare](#)

ShojiObject-class	<i>Mix-in class for multiple inheritance of variables and datasets.</i>
-------------------	---

Description

Exists for common methods in interacting with Crunch API only. Has no Extract methods declared so as not to conflict with the vector/list/data.frame methods jointly inherited in CrunchVariable and CrunchDataset.

ShojiOrder-extract *Extract and update in VariableOrder and VariableGroup*

Description

Extract and update in VariableOrder and VariableGroup

Usage

```
## S4 method for signature 'ShojiOrder,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'ShojiOrder,character,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'ShojiOrder,ANY'
x[[i, j, ...]]

## S4 method for signature 'ShojiOrder,character'
x[[i, j, ...]]

## S4 method for signature 'ShojiOrder'
x$name

## S4 replacement method for signature 'ShojiOrder,character,missing,ShojiOrder'
x[i, j] <- value

## S4 replacement method for signature 'ShojiOrder,ANY,missing,ShojiOrder'
x[i, j] <- value

## S4 replacement method for signature 'ShojiOrder,character,missing,list'
x[[i, j]] <- value

## S4 replacement method for signature 'ShojiOrder,character,missing,character'
x[[i, j]] <- value

## S4 replacement method for signature 'ShojiOrder,character,missing,OrderGroup'
x[[i, j]] <- value

## S4 replacement method for signature 'ShojiOrder,ANY,missing,OrderGroup'
x[[i, j]] <- value

## S4 replacement method for signature 'ShojiOrder,ANY,missing,ANY'
x[[i, j]] <- value

## S4 replacement method for signature 'ShojiOrder,ANY,missing,`NULL`'
x[[i, j]] <- value
```

```
## S4 replacement method for signature 'ShojiOrder,character,missing,`NULL`'  
x[[i, j]] <- value  
  
## S4 replacement method for signature 'ShojiOrder,character,missing,ShojiOrder'  
x[[i, j]] <- value  
  
## S4 replacement method for signature 'ShojiOrder'  
x$name <- value  
  
## S4 method for signature 'OrderGroup,ANY,ANY'  
x[i, j, ..., drop = TRUE]  
  
## S4 method for signature 'OrderGroup,character,ANY'  
x[i, j, ..., drop = TRUE]  
  
## S4 method for signature 'OrderGroup,character'  
x[[i, j, ...]]  
  
## S4 method for signature 'OrderGroup,ANY'  
x[[i, j, ...]]  
  
## S4 method for signature 'OrderGroup'  
x$name  
  
## S4 replacement method for signature 'OrderGroup,character,missing,list'  
x[[i, j]] <- value  
  
## S4 replacement method for signature 'OrderGroup,character,missing,character'  
x[[i, j]] <- value  
  
## S4 replacement method for signature 'OrderGroup,character,missing,ShojiOrder'  
x[[i, j]] <- value  
  
## S4 replacement method for signature 'OrderGroup,character,missing,OrderGroup'  
x[[i, j]] <- value  
  
## S4 replacement method for signature 'OrderGroup,ANY,missing,OrderGroup'  
x[[i, j]] <- value  
  
## S4 replacement method for signature 'OrderGroup,numeric,missing,`NULL`'  
x[[i, j]] <- value  
  
## S4 replacement method for signature 'OrderGroup,character,missing,`NULL`'  
x[[i, j]] <- value  
  
## S4 replacement method for signature 'OrderGroup'  
x$name <- value
```

```
## S4 replacement method for signature 'VariableOrder,character,missing,CrunchDataset'
x[[i, j]] <- value

## S4 replacement method for signature 'VariableGroup,character,missing,CrunchDataset'
x[[i, j]] <- value
```

Arguments

x	a VariableOrder or VariableGroup
i	an index. Numeric and logical indexing supported for both classes; character indexing supported for VariableOrder, matching on VariableGroup names
j	Invalid
...	additional arguments
drop	Ignored
name	Same as i but for \$
value	For update methods, an object equivalent in class to what is being updated

Value

[[and \$ on a VariableOrder return the VariableGroup. [[on VariableGroup returns the entity within, either a character (URL) or nested VariableGroup. [and assignment methods return objects of the same class as x

ShojiOrder-length	<i>Length of an Order</i>
-------------------	---------------------------

Description

Length of an Order

Usage

```
## S4 method for signature 'ShojiOrder'
length(x)

## S4 method for signature 'OrderGroup'
length(x)
```

Arguments

x	a ShojiOrder
---	--------------

Value

Integer: the number of elements in the Order

ShojiOrder-slots *Manipulate VariableGroup and VariableOrder*

Description

Manipulate VariableGroup and VariableOrder

Usage

```
## S4 method for signature 'OrderGroup'  
entities(x, simplify = FALSE)  
  
## S4 method for signature 'ShojiOrder'  
entities(x, simplify = FALSE)  
  
## S4 method for signature 'list'  
entities(x, simplify = FALSE)  
  
## S4 replacement method for signature 'OrderGroup'  
entities(x) <- value  
  
## S4 replacement method for signature 'ShojiOrder'  
entities(x) <- value  
  
## S4 method for signature 'OrderGroup'  
name(x)  
  
## S4 replacement method for signature 'OrderGroup'  
name(x) <- value  
  
## S4 method for signature 'ShojiOrder'  
names(x)  
  
## S4 method for signature 'OrderGroup'  
names(x)  
  
## S4 replacement method for signature 'ShojiOrder'  
names(x) <- value  
  
## S4 method for signature 'ShojiOrder'  
duplicates(x)  
  
## S4 method for signature 'OrderGroup'  
duplicates(x)  
  
## S4 method for signature 'VariableCatalog'  
duplicates(x)
```

```
## S4 replacement method for signature 'ShojiOrder,logical'
duplicates(x) <- value

## S4 replacement method for signature 'OrderGroup,logical'
duplicates(x) <- value

## S4 replacement method for signature 'VariableCatalog,logical'
duplicates(x) <- value
```

Arguments

x	a VariableGroup or VariableOrder
simplify	logical: should variable URLs inside of groups be flattened or preserved in their nested lists? Default is FALSE.
value	(1) For name, a character (length-1 vector); for names, a character vector of equal length to the number of VariableGroups being modified; for entities, either a character vector of variable URLs or a list containing a combination of variable URLs and VariableGroups. Note that group names must be unique, should be greater than 0 characters long, and "ungrouped" is a reserved group name. (2) For duplicates, logical for whether duplicate variable entries should be allowed in the VariableOrder.

Value

entities returns Variable references and VariableGroups; names returns group names; duplicates returns logical for whether duplicate variable entries should be allowed

See Also

[VariableOrder](#)
[grouped](#)

show-crunch

Show methods for Crunch objects

Description

Show methods for Crunch objects

Usage

```
## S4 method for signature 'ShojiObject'
show(object)

## S4 method for signature 'CrunchVariable'
show(object)
```

```
## S4 method for signature 'Category'  
show(object)  
  
## S4 method for signature 'Categories'  
show(object)  
  
## S4 method for signature 'CrunchExpr'  
show(object)  
  
## S4 method for signature 'CrunchLogicalExpr'  
show(object)  
  
## S4 method for signature 'CrunchCube'  
show(object)  
  
## S4 method for signature 'OrderGroup'  
show(object)
```

Arguments

object the object

Value

invisibly

See Also

[show](#)

Subvariables-class *Subvariables in Array Variables*

Description

Multiple-response and categorical-array variables contain a set of subvariables within them. The Subvariables class encapsulates them.

Usage

```
## S4 method for signature 'CategoricalArrayVariable'  
subvariables(x)  
  
## S4 method for signature 'VariableTuple'  
subvariables(x)  
  
## S4 replacement method for signature 'CategoricalArrayVariable,ANY'
```

```

subvariables(x) <- value

## S4 replacement method for signature 'CategoricalArrayVariable,Subvariables'
subvariables(x) <- value

```

Arguments

x	A Variable or Subvariables object
value	For the setters, the appropriate values to set

Details

Subvariables can be accessed from array variables (including multiple response) with the `subvariables` method. They can be assigned back with the `subvariables<-` setter, but there are limitations to what is supported. Specifically, you can reorder subvariables, but you cannot add or remove subvariables by `subvariables<-` assignment. See [deleteSubvariable](#) to remove subvariables from an array.

Subvariables have a `names` attribute that can be accessed, showing the display names of the subvariables. These can be set with the `names<-` method.

Finally, subvariables can be accessed as regular (categorical) variables with the `$` and `[[` extract methods.

See the vignette on array variables for further details and examples.

See Also

[subvars-extract describe-catalog deleteSubvariable vignette\("array-variables", package="crunch"\)](#)

subvars-extract	<i>Extract and modify subsets of subvariables</i>
-----------------	---

Description

Extract and modify subsets of subvariables

Usage

```

## S4 method for signature 'Subvariables,character'
x[[i, j, ...]]

## S4 method for signature 'Subvariables,ANY'
x[[i, j, ...]]

## S4 method for signature 'Subvariables,character,ANY'
x[i, j, ..., drop = TRUE]

## S4 replacement method for signature 'Subvariables,character,missing,CrunchVariable'

```

```

x[[i]] <- value

## S4 replacement method for signature 'Subvariables,ANY,missing,CrunchVariable'
x[[i]] <- value

## S4 replacement method for signature 'Subvariables,ANY,missing,`NULL`'
x[[i]] <- value

## S4 replacement method for signature 'Subvariables,ANY,missing,ANY'
x[[i]] <- value

## S4 replacement method for signature 'Subvariables,character,missing,Subvariables'
x[i] <- value

## S4 replacement method for signature 'Subvariables,ANY,missing,Subvariables'
x[i] <- value

## S4 replacement method for signature 'Subvariables,ANY,missing,ANY'
x[i] <- value

## S4 method for signature 'CategoricalArrayVariable,character,ANY'
x[i, j, ...,
  drop = TRUE]

## S4 method for signature 'CategoricalArrayVariable,ANY'
x[[i, j, ...]]

## S4 method for signature 'CategoricalArrayVariable,character'
x[[i, j, ...]]

## S4 method for signature 'CategoricalArrayVariable'
x$name

## S4 replacement method for signature 'CategoricalArrayVariable,ANY,missing,ANY'
x[[i]] <- value

## S4 replacement method for signature 'CategoricalArrayVariable,character,missing,ANY'
x[[i]] <- value

## S4 replacement method for signature 'CategoricalArrayVariable'
x$name <- value

```

Arguments

x	Subvariables or an array Variable (which contains subvariables)
i	which subvariables to extract
j	Invalid
...	additional arguments

drop	Invalid
value	For updating, a CrunchExpr
name	For \$, the name (not alias) of the subvariable to extract

Value

A subset of x if extracting, otherwise x duly modified

table	<i>Table function for Crunch objects</i>
-------	--

Description

Table function for Crunch objects

Usage

```
table(..., exclude, useNA = c("no", "ifany", "always"), dnn, deparse.level)
```

Arguments

...	things to tabulate
exclude	see table
useNA	see table
dnn	see table
deparse.level	see table

Value

a table object

See Also

[table](#)

temp.options	<i>Set some global options temporarily</i>
--------------	--

Description

Set some global options temporarily

Usage

```
temp.options(...)
```

```
temp.option(...)
```

Arguments

... named options to set

Value

an S3 class "contextManager" object

See Also

[with-context-manager](#) [ContextManager](#)

tojson-crunch	<i>toJSON methods for Crunch objects</i>
---------------	--

Description

crunch uses the `jsonlite` package for (de)serialization of JSON. Unlike `RJSONIO`'s `toJSON`, `toJSON` does not allow for defining S4 methods for other object types. So, `crunch::toJSON` wraps `jsonprep`, which exists to translate objects to base R objects, which `jsonlite::toJSON` can handle. `jsonprep` is defined as an S4 generic, and it is exported (unlike `codejsonlite::asJSON`), so you can define methods for it if you have other objects that you want to successfully serialize to JSON.

Usage

```
jsonprep(x, ...)
```

```
## S4 method for signature 'Categories'
```

```
jsonprep(x, ...)
```

```
## S4 method for signature 'list'
```

```
jsonprep(x, ...)
```

```
## S4 method for signature 'ANY'
jsonprep(x, ...)

## S4 method for signature 'ShojiOrder'
jsonprep(x, ...)

## S4 method for signature 'OrderGroup'
jsonprep(x, ...)

toJSON(x, ...)
```

Arguments

x	the object
...	additional arguments

Value

jsonprep returns a base R object that jsonlite::toJSON can handle. toJSON returns the JSON-serialized character object.

See Also

[toJSON](#)

toVariable	<i>Generic method for converting objects to Crunch representations</i>
------------	--

Description

If you have other object types you wish to convert to Crunch variables, you can declare methods for toVariable

Usage

```
toVariable(x, ...)

## S4 method for signature 'character'
toVariable(x, ...)

## S4 method for signature 'numeric'
toVariable(x, ...)

## S4 method for signature 'factor'
toVariable(x, ...)

## S4 method for signature 'Date'
```



```

toVariable(x, ...)

## S4 method for signature 'POSIXt'
toVariable(x, ...)

## S4 method for signature 'VariableDefinition'
toVariable(x, ...)

## S4 method for signature 'logical'
toVariable(x, ...)

## S4 method for signature 'CrunchExpr'
toVariable(x, ...)

```

Arguments

x	the object
...	additional arguments

Value

a list object suitable for POSTing to the Crunch API. See the API documentation for specifications.

type	<i>Change the type of Crunch variables</i>
------	--

Description

Numeric, text, and categorical variables can be cast to one another by assigning them a new "type". This modifies the storage of the data on the server and should only be done in narrow circumstances, as in when importing data from a different file format has resulted in incorrect types being specified.

Usage

```

## S4 method for signature 'CrunchVariable'
type(x)

## S4 replacement method for signature 'CrunchVariable'
type(x) <- value

```

Arguments

x	a Variable
value	For the setter, a character value in c("numeric", "text", "categorical")

Value

Getter returns character; setter returns x duly modified.

unbind *Split an array or multiple-response variable into its CategoricalVariables*

Description

Split an array or multiple-response variable into its CategoricalVariables

Usage

```
unbind(x)
```

Arguments

x a CategoricalArrayVariable or MultipleResponseVariable

Value

invisibly, the API response from DELETEing the array variable definition. If you [refresh](#) the corresponding dataset after unbinding, you should see the array variable removed and its subvariables promoted to regular variables.

unshare *Revoke a user's access to a dataset*

Description

Revoke a user's access to a dataset

Usage

```
unshare(dataset, users)
```

Arguments

dataset a CrunchDataset
users character: email address(es) or URLs of the users or teams to unshare with.

Value

Invisibly, the dataset.

See Also

[share](#)

updateDatasetList *Refresh the local list of Crunch datasets*

Description

Refresh the local list of Crunch datasets

Usage

```
updateDatasetList()
```

Value

Nothing. Called for its side effects of setting local environment variables.

var-categories *Get and set Categories on Variables*

Description

Get and set Categories on Variables

Usage

```
## S4 method for signature 'CrunchVariable'  
categories(x)  
  
## S4 method for signature 'CategoricalVariable'  
categories(x)  
  
## S4 method for signature 'CategoricalArrayVariable'  
categories(x)  
  
## S4 method for signature 'VariableEntity'  
categories(x)  
  
## S4 replacement method for signature 'CategoricalVariable,Categories'  
categories(x) <- value  
  
## S4 replacement method for signature 'CategoricalArrayVariable,Categories'  
categories(x) <- value  
  
## S4 replacement method for signature 'CategoricalVariable,numeric'  
categories(x) <- value
```

```
## S4 replacement method for signature 'CategoricalVariable,character'
categories(x) <- value

## S4 replacement method for signature 'CategoricalVariable,ANY'
categories(x) <- value

## S4 replacement method for signature 'CategoricalArrayVariable,numeric'
categories(x) <- value

## S4 replacement method for signature 'CategoricalArrayVariable,character'
categories(x) <- value

## S4 replacement method for signature 'CategoricalArrayVariable,ANY'
categories(x) <- value

## S4 replacement method for signature 'CrunchVariable,ANY'
categories(x) <- value
```

Arguments

x a Variable
value for the setters, an object of class Categories to set.

Value

Getters return Categories; setters return x duly modified.

variable-update	<i>Updating variables with expressions or values</i>
-----------------	--

Description

Updating variables with expressions or values

Usage

```
## S4 replacement method for signature 'CrunchVariable,ANY,missing,ANY'
x[i, j] <- value

## S4 replacement method for signature 'CrunchVariable,ANY,missing,`NULL`'
x[i, j] <- value

## S4 replacement method for signature 'TextVariable,ANY,missing,character'
x[i, j] <- value

## S4 replacement method for signature 'NumericVariable,ANY,missing,numeric'
x[i, j] <- value
```

```

## S4 replacement method for signature 'DatetimeVariable,ANY,missing,Date'
x[i, j] <- value

## S4 replacement method for signature 'DatetimeVariable,ANY,missing,POSIXt'
x[i, j] <- value

## S4 replacement method for signature 'CrunchVariable,ANY,missing,CrunchExpr'
x[i, j] <- value

## S4 replacement method for signature 'CrunchVariable,CrunchExpr,missing,CrunchExpr'
x[i, j] <- value

## S4 replacement method for signature 'CategoricalVariable,ANY,missing,numeric'
x[i, j] <- value

## S4 replacement method for signature 'CategoricalVariable,ANY,missing,character'
x[i, j] <- value

## S4 replacement method for signature 'CategoricalVariable,ANY,missing,factor'
x[i, j] <- value

## S4 replacement method for signature 'CategoricalArrayVariable,ANY,missing,numeric'
x[i, j] <- value

## S4 replacement method for signature 'CategoricalArrayVariable,ANY,missing,character'
x[i, j] <- value

## S4 replacement method for signature 'CategoricalArrayVariable,ANY,missing,factor'
x[i, j] <- value

## S4 replacement method for signature 'CrunchVariable,ANY,missing,logical'
x[i, j] <- value

## S4 replacement method for signature 'CrunchVariable,ANY'
is.na(x) <- value

```

Arguments

x	a Variable
i	a CrunchLogicalExpr or R index, optionally
j	Invalid
value	an R vector or a CrunchExpr with which to update

Value

x duly modified

VariableCatalog-class *Collection of Variables within a Dataset*

Description

A VariableCatalog contains references to all variables in a dataset, plus some descriptive metadata about each. VariableCatalogs also contain a [VariableOrder](#) that governs how variables within it are organized.

VariableDefinition *Construct a variable definition with (optional) additional metadata*

Description

Construct a variable definition with (optional) additional metadata

Usage

```
VariableDefinition(data, ...)
```

```
VarDef(data, ...)
```

Arguments

data	an R vector of data to convert to the Crunch payload format. See code to Variable for how R types are converted. If data is not supplied, you may instead supply values, which will not be converted in any way, nor will extra type information be supplied. Only send values if you know what you're doing. You may also omit both data and values to create an empty variable on the server (all values will be system missing "No Data").
...	additional metadata attributes to send.

Value

a VariableDefinition object, ready to POST to Crunch.

See Also

[toVariable](#)

Examples

```
VariableDefinition(rnorm(5), name="Some numbers",
  description="Generated pseudorandomly from the normal distribution")
VarDef(name="Integers", values=1:5, type="numeric",
  description="When creating variable definitions with 'values', you must
  specify 'type', and categorical variables will require 'categories'.")
```

variableMetadata	<i>Get all variable metadata for a dataset</i>
------------------	--

Description

Get all variable metadata for a dataset

Usage

```
variableMetadata(dataset)
```

Arguments

dataset	CrunchDataset
---------	---------------

Value

A VariableCatalog that has things like categories embedded in each categorical variable, and all subvariables are represented

VariableOrder-class	<i>Organize Variables within a Dataset</i>
---------------------	--

Description

Variables in the Crunch web application can be viewed in an ordered, hierarchical list. These objects and methods allow you to modify that order from R.

Details

A VariableOrder object is a subclass of list that contains VariableGroups. VariableGroup objects contain a group name and an set of "entities", which can be variable references or other nested VariableGroups.

Slots

group character, the name of the VariableGroup. In the constructor and more generally, this field can be referenced as "name" as well.

entities a character vector of variable URLs, or a list containing a combination of variable URLs and VariableGroup objects.

duplicates logical: should duplicate variable references be allowed in this object? Default is FALSE.

vars either NULL or a [VariableCatalog](#). If not NULL, it will be used to look up variable names from the URLs.

versions	<i>Access the saved versions of a dataset</i>
----------	---

Description

Access the saved versions of a dataset

Usage

```
versions(x)
```

Arguments

x a CrunchDataset

Value

an object of class `VersionCatalog`. Supported methods on the catalog include "names" and "timestamps".

See Also

[saveVersion](#) [restoreVersion](#)

weight	<i>Dataset weights</i>
--------	------------------------

Description

Dataset weights

Usage

```
weight(x)
```

```
weight(x) <- value
```

Arguments

x a Dataset
value a Variable to set as weight, or NULL to remove the existing weight

Value

For the getter, a Variable if there is a weight, else NULL. For the setter, x, modified accordingly

which	<i>"which" method for CrunchLogicalExpr</i>
-------	---

Description

NOTE: this isn't correct. Don't use it yet.

Usage

```
## S4 method for signature 'CrunchLogicalExpr'
which(x, arr.ind = FALSE, useNames = TRUE)
```

Arguments

x	CrunchLogicalExpr
arr.ind	Ignored
useNames	Ignored

Value

Integer row indices where x is true. Note that this does not return a Crunch expression. Use this when you need to translate to R values. For filtering a Crunch expression by x, don't use which.

with-context-manager	<i>Context manager's "with" method</i>
----------------------	--

Description

Context manager's "with" method

Usage

```
## S3 method for class 'contextManager'
with(data, expr, ...)
```

Arguments

data	<code>contextManager</code>
expr	code to evaluate within that context
...	additional arguments. One additional supported argument is "as", which lets you assign the return of your "enter" function to an object you can access.

Value

Nothing.

See Also[ContextManager](#)

[,CrunchExpr,CrunchLogicalExpr,ANY-method
"Subset" a Variable

Description

These methods subset variables by creating Expressions, which can be composed and evaluated as needed.

Usage

```
## S4 method for signature 'CrunchExpr,CrunchLogicalExpr,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'CrunchExpr,logical,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CrunchExpr,numeric,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CrunchVariable,CrunchExpr,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'CrunchVariable,numeric,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CrunchVariable,logical,ANY'
x[i, j, ..., drop = TRUE]
```

Arguments

x	a Variable
i	a CrunchExpr, logical, or numeric
j	Invalid
...	additional arguments, ignored
drop	Invalid

Value

a CrunchExpr containing references to the variable x and the filter logic contained in i

Index

- !,CrunchExpr-method (expressions), 43
- !=,CategoricalVariable,character-method (expressions), 43
- !=,CategoricalVariable,factor-method (expressions), 43
- ==,CategoricalVariable,character-method (expressions), 43
- ==,CategoricalVariable,factor-method (expressions), 43
- [,CategoricalArrayVariable,character,ANY-method (subvars-extract), 84
- [,Categories,ANY,ANY-method (Categories-class), 13
- [,Categories,numeric,ANY-method (Categories-class), 13
- [,CrunchDataset,ANY,ANY-method (dataset-extract), 24
- [,CrunchDataset,CrunchLogicalExpr,ANY-method (dataset-extract), 24
- [,CrunchDataset,CrunchLogicalExpr,missing-method (dataset-extract), 24
- [,CrunchDataset,VariableGroup,ANY-method (dataset-extract), 24
- [,CrunchDataset,VariableOrder,ANY-method (dataset-extract), 24
- [,CrunchDataset,character,ANY-method (dataset-extract), 24
- [,CrunchDataset,logical,missing-method (dataset-extract), 24
- [,CrunchDataset,missing,ANY-method (dataset-extract), 24
- [,CrunchExpr,CrunchLogicalExpr,ANY-method, 98
- [,CrunchExpr,logical,ANY-method ([,CrunchExpr,CrunchLogicalExpr,ANY-method), 98
- [,CrunchExpr,numeric,ANY-method ([,CrunchExpr,CrunchLogicalExpr,ANY-method), 98
- [,CrunchVariable,CrunchExpr,ANY-method ([,CrunchExpr,CrunchLogicalExpr,ANY-method), 98
- [,CrunchVariable,logical,ANY-method ([,CrunchExpr,CrunchLogicalExpr,ANY-method), 98
- [,CrunchVariable,numeric,ANY-method ([,CrunchExpr,CrunchLogicalExpr,ANY-method), 98
- MemberCatalog,character,ANY-method (catalog-extract), 8
- [,OrderGroup,ANY,ANY-method (ShojiOrder-extract), 78
- [,OrderGroup,character,ANY-method (ShojiOrder-extract), 78
- [,PermissionCatalog,character,ANY-method (catalog-extract), 8
- [,ShojiCatalog,ANY,ANY-method (catalog-extract), 8
- [,ShojiCatalog,character,ANY-method (catalog-extract), 8
- [,ShojiCatalog,logical,ANY-method (catalog-extract), 8
- [,ShojiCatalog,numeric,ANY-method (catalog-extract), 8
- [,ShojiOrder,ANY,ANY-method (ShojiOrder-extract), 78
- [,ShojiOrder,character,ANY-method (ShojiOrder-extract), 78
- [,Subvariables,character,ANY-method (subvars-extract), 84
- [,VariableCatalog,VariableGroup,ANY-method (catalog-extract), 8
- [,VariableCatalog,VariableOrder,ANY-method (catalog-extract), 8
- [<-,CategoricalArrayVariable,ANY,missing,character-method (variable-update), 92
- [<-,CategoricalArrayVariable,ANY,missing,factor-method (variable-update), 92

- [<- , CategoricalArrayVariable, ANY, missing, numeric-method (variable-update), 92
- [<- , CategoricalArrayVariable, ANY-method (subvars-extract), 84
- [<- , CategoricalVariable, ANY, missing, character-method (variable-update), 92
- [<- , CategoricalVariable, ANY, missing, character-method (subvars-extract), 84
- [<- , CategoricalVariable, ANY, missing, factor-method (variable-update), 92
- [<- , CrunchDataset, ANY-method (dataset-extract), 24
- [<- , CrunchDataset, character-method (dataset-extract), 24
- [<- , Categories, ANY, ANY, ANY-method (Categories-class), 13
- [<- , DatasetCatalog, ANY-method (catalog-extract), 8
- [<- , CrunchDataset, ANY, missing, list-method (dataset-update), 26
- [<- , DatasetCatalog, character-method (catalog-extract), 8
- [<- , CrunchDataset, CrunchExpr, ANY, ANY-method (dataset-update), 26
- [<- , FilterCatalog, character-method (catalog-extract), 8
- [<- , CrunchVariable, ANY, missing, ANY-method (variable-update), 92
- [<- , FilterCatalog, numeric-method (catalog-extract), 8
- [<- , CrunchVariable, ANY, missing, CrunchExpr-method (variable-update), 92
- [<- , MemberCatalog, character-method (catalog-extract), 8
- [<- , CrunchVariable, ANY, missing, NULL-method (variable-update), 92
- [<- , OrderGroup, ANY-method (ShojiOrder-extract), 78
- [<- , CrunchVariable, ANY, missing, logical-method (variable-update), 92
- [<- , OrderGroup, character-method (ShojiOrder-extract), 78
- [<- , CrunchVariable, CrunchExpr, missing, CrunchExpr-method (variable-update), 92
- [<- , PermissionCatalog, character-method (catalog-extract), 8
- [<- , DatetimeVariable, ANY, missing, Date-method (variable-update), 92
- [<- , ProjectCatalog, ANY-method (catalog-extract), 8
- [<- , DatetimeVariable, ANY, missing, POSIXt-method (variable-update), 92
- [<- , ProjectCatalog, character-method (catalog-extract), 8
- [<- , NumericVariable, ANY, missing, numeric-method (variable-update), 92
- [<- , Session, ANY-method (session-extract), 75
- [<- , ShojiCatalog, ANY, missing, ShojiCatalog-method (catalog-extract), 8
- [<- , ShojiCatalog, ANY-method (catalog-extract), 8
- [<- , ShojiOrder, ANY, missing, ShojiOrder-method (ShojiOrder-extract), 78
- [<- , ShojiCatalog, character-method (catalog-extract), 8
- [<- , ShojiOrder, character, missing, ShojiOrder-method (ShojiOrder-extract), 78
- [<- , ShojiEntity, ANY-method (entity, CrunchProject-method), 40
- [<- , Subvariables, ANY, missing, ANY-method (subvars-extract), 84
- [<- , ShojiOrder, ANY-method (ShojiOrder-extract), 78
- [<- , Subvariables, ANY, missing, Subvariables-method (subvars-extract), 84
- [<- , ShojiOrder, character-method (ShojiOrder-extract), 78
- [<- , Subvariables, character, missing, Subvariables-method (subvars-extract), 84
- [<- , ShojiTuple, ANY-method (entity, CrunchProject-method), 40
- [<- , TextVariable, ANY, missing, character-method (variable-update), 92
- [<- , VariableCatalog, ANY-method (subvars-extract), 84
- [<- , VariableCatalog, VariableGroup, missing, VariableCatalog-method (catalog-extract), 8
- [<- , VariableCatalog, VariableOrder, missing, VariableCatalog-method (catalog-extract), 8
- [<- , VariableCatalog, ANY-method (subvars-extract), 84

- [[,TeamCatalog,character-method
(catalog-extract), 8
- [[,TeamCatalog,numeric-method
(catalog-extract), 8
- [[,VariableCatalog,ANY-method
(catalog-extract), 8
- [[,VariableCatalog,character-method
(catalog-extract), 8
- [[<-,CategoricalArrayVariable,ANY,missing,ANY-method
(subvars-extract), 84
- [[<-,CategoricalArrayVariable,character,missing,ANY-method
(subvars-extract), 84
- [[<-,CrunchDataset,ANY,ANY,ANY-method
(dataset-update), 26
- [[<-,CrunchDataset,ANY,missing,CrunchVariable-method
(dataset-update), 26
- [[<-,CrunchDataset,ANY,missing,NULL-method
(dataset-update), 26
- [[<-,CrunchDataset,character,missing,ANY-method
(dataset-update), 26
- [[<-,CrunchDataset,character,missing,CrunchLogicalExpr-method
(dataset-update), 26
- [[<-,CrunchDataset,character,missing,CrunchVariable-method
(dataset-update), 26
- [[<-,CrunchDataset,character,missing,NULL-method
(dataset-update), 26
- [[<-,DatasetCatalog,character,missing,DataSetTuple-method
(catalog-extract), 8
- [[<-,FilterCatalog,character,missing,CrunchFilter-method
(catalog-extract), 8
- [[<-,FilterCatalog,character,missing,CrunchLogicalExpr-method
(catalog-extract), 8
- [[<-,FilterCatalog,numeric,missing,CrunchFilter-method
(catalog-extract), 8
- [[<-,FilterCatalog,numeric,missing,CrunchLogicalExpr-method
(catalog-extract), 8
- [[<-,MemberCatalog,ANY,missing,ANY-method
(catalog-extract), 8
- [[<-,MemberCatalog,character,missing,NULL-method
(catalog-extract), 8
- [[<-,OrderGroup,ANY,missing,OrderGroup-method
(ShojiOrder-extract), 78
- [[<-,OrderGroup,character,missing,NULL-method
(ShojiOrder-extract), 78
- [[<-,OrderGroup,character,missing,OrderGroup-method
(ShojiOrder-extract), 78
- [[<-,OrderGroup,character,missing,ShojiOrder-method
(ShojiOrder-extract), 78
- [[<-,OrderGroup,character,missing,character-method
(ShojiOrder-extract), 78
- [[<-,OrderGroup,character,missing,list-method
(ShojiOrder-extract), 78
- [[<-,OrderGroup,numeric,missing,NULL-method
(ShojiOrder-extract), 78
- [[<-,ProjectCatalog,character,missing,CrunchProject-method
(catalog-extract), 8
- [[<-,ProjectCatalog,character,missing,list-method
(catalog-extract), 8
- [[<-,SessionCatalog,ANY,ANY,ANY-method
(session-extract), 75
- [[<-,ShojiEntity,ANY,ANY,ANY-method
(entity,CrunchProject-method), 40
- [[<-,ShojiOrder,ANY,missing,ANY-method
(ShojiOrder-extract), 78
- [[<-,ShojiOrder,ANY,missing,NULL-method
(ShojiOrder-extract), 78
- [[<-,ShojiOrder,ANY,missing,OrderGroup-method
(ShojiOrder-extract), 78
- [[<-,ShojiOrder,character,missing,NULL-method
(ShojiOrder-extract), 78
- [[<-,ShojiOrder,character,missing,OrderGroup-method
(ShojiOrder-extract), 78
- [[<-,ShojiOrder,character,missing,ShojiOrder-method
(ShojiOrder-extract), 78
- [[<-,ShojiOrder,character,missing,character-method
(ShojiOrder-extract), 78
- [[<-,ShojiOrder,character,missing,list-method
(ShojiOrder-extract), 78
- [[<-,ShojiTuple,ANY,ANY,ANY-method
(entity,CrunchProject-method), 40
- [[<-,Subcatalogs,ANY,missing,ANY-method
(subvars-extract), 84
- [[<-,Subvariables,ANY,missing,CrunchVariable-method
(subvars-extract), 84
- [[<-,Subvariables,ANY,missing,NULL-method
(subvars-extract), 84
- [[<-,Subvariables,character,missing,CrunchVariable-method
(subvars-extract), 84
- [[<-,TeamCatalog,character,missing,CrunchTeam-method
(catalog-extract), 8
- [[<-,TeamCatalog,character,missing,list-method
(catalog-extract), 8
- [[<-,VariableCatalog,character,missing,CrunchVariable-method
(catalog-extract), 8

- [[<- , VariableCatalog, character, missing, VariableTuple-method (expressions), 43
 - (catalog-extract), 8
 - %in%, DatetimeVariable, Date-method
- [[<- , VariableGroup, character, missing, CrunchDataset-method (expressions), 43
 - (ShojiOrder-extract), 78
 - %in%, DatetimeVariable, POSIXt-method
- [[<- , VariableOrder, character, missing, CrunchDataset-method (expressions), 43
 - (ShojiOrder-extract), 78
 - %in%, DatetimeVariable, character-method
- \$, CategoricalArrayVariable-method
 - (subvars-extract), 84
 - %in%, NumericVariable, numeric-method
- \$, Category-method (category-extract), 14
 - (expressions), 43
- \$, CrunchDataset-method
 - (dataset-extract), 24
 - %in%, TextVariable, character-method
 - (expressions), 43
- \$, OrderGroup-method
 - (ShojiOrder-extract), 78
- \$, Session-method (session-extract), 75
- \$, ShojiCatalog-method
 - (catalog-extract), 8
- \$, ShojiEntity-method
 - (entity, CrunchProject-method), 40
- \$, ShojiOrder-method
 - (ShojiOrder-extract), 78
- \$, ShojiTuple-method
 - (entity, CrunchProject-method), 40
- \$<- , CategoricalArrayVariable-method
 - (subvars-extract), 84
- \$<- , Category-method (category-extract), 14
- \$<- , CrunchDataset-method
 - (dataset-update), 26
- \$<- , OrderGroup-method
 - (ShojiOrder-extract), 78
- \$<- , Session-method (session-extract), 75
- \$<- , ShojiCatalog-method
 - (catalog-extract), 8
- \$<- , ShojiEntity-method
 - (entity, CrunchProject-method), 40
- \$<- , ShojiOrder-method
 - (ShojiOrder-extract), 78
- \$<- , ShojiTuple-method
 - (entity, CrunchProject-method), 40
- %in%, CategoricalVariable, character-method
 - (expressions), 43
- %in%, CategoricalVariable, factor-method
 - (expressions), 43
- %in%, CategoricalVariable, numeric-method
 - addSubvariable, 4
 - addSubvariables (addSubvariable), 4
 - addVariables, 5
 - alias, CrunchVariable-method (describe), 33
 - alias<- (describe), 33
 - alias<- , CrunchVariable-method (describe), 33
 - aliases (names, BatchCatalog-method), 64
 - aliases, Subvariables-method (names, BatchCatalog-method), 64
 - aliases, VariableCatalog-method (names, BatchCatalog-method), 64
 - aliases<- (names, BatchCatalog-method), 64
 - aliases<- , Subvariables-method (names, BatchCatalog-method), 64
 - aliases<- , VariableCatalog-method (names, BatchCatalog-method), 64
 - allVariables (dataset-variables), 28
 - allVariables, CrunchDataset-method (dataset-variables), 28
 - allVariables<- (dataset-variables), 28
 - allVariables<- , CrunchDataset, VariableCatalog-method (dataset-variables), 28
 - appendDataset, 5
 - archive
 - (is.archived, DatasetCatalog-method), 53
 - archive-and-publish
 - (is.archived, DatasetCatalog-method), 53
 - as.data.frame.CrunchDataFrame
 - (dataset-to-R), 26
 - as.data.frame.CrunchDataset
 - (dataset-to-R), 26
 - as.environment, CrunchDataset-method, 6

- as.vector, CrunchExpr-method, 6
- as.vector, CrunchVariable-method
 - (as.vector, CrunchExpr-method), 6
- batches, 7
- bin (expressions), 43
- c, 8
- c-categories, 8
- c.Categories (c-categories), 8
- c.Category (c-categories), 8
- catalog-extract, 8
- catalog-length, 12
- catalogToDataFrame, 12
- CategoricalArrayVariable
 - (CrunchVariable-class), 22
- CategoricalArrayVariable-class
 - (CrunchVariable-class), 22
- CategoricalVariable
 - (CrunchVariable-class), 22
- CategoricalVariable-class
 - (CrunchVariable-class), 22
- Categories, 8, 35, 36, 66
- Categories (Categories-class), 13
- categories (var-categories), 91
- categories, CategoricalArrayVariable-method
 - (var-categories), 91
- categories, CategoricalVariable-method
 - (var-categories), 91
- categories, CrunchVariable-method
 - (var-categories), 91
- categories, VariableEntity-method
 - (var-categories), 91
- Categories-class, 13
- categories<- (var-categories), 91
- categories<-, CategoricalArrayVariable, ANY-method
 - (var-categories), 91
- categories<-, CategoricalArrayVariable, CategoricalArrayVariable-method
 - (var-categories), 91
- categories<-, CategoricalArrayVariable, character-method
 - (var-categories), 91
- categories<-, CategoricalArrayVariable, numeric-method
 - (var-categories), 91
- categories<-, CategoricalVariable, ANY-method
 - (var-categories), 91
- categories<-, CategoricalVariable, Categories-class-method
 - (var-categories), 91
- categories<-, CategoricalVariable, character-method
 - (var-categories), 91
- categories<-, CategoricalVariable, numeric-method
 - (var-categories), 91
- categories<-, CrunchVariable, ANY-method
 - (var-categories), 91
- Category (Categories-class), 13
- Category-class (Categories-class), 13
- category-extract, 14
- cleanseBatches, 6, 15
- combine, 15
- compareDatasets, 16
- consent, 17, 31–33
- ContextManager, 17, 18, 87, 98
- contextManager, 97
- contextManager (ContextManager), 18
- copy (copyVariable), 18
- copyVariable, 18
- crDELETE (http-methods), 51
- createDataset, 66–68
- crGET (http-methods), 51
- crPATCH (http-methods), 51
- crPOST (http-methods), 51
- crPUT (http-methods), 51
- crtabs, 19
- crunch, 20
- crunch-package (crunch), 20
- crunch-uni, 20
- crunchAPI, 51
- crunchBox, 21, 39, 40, 70
- CrunchDataset, 68
- CrunchDataset (CrunchDataset-class), 22
- CrunchDataset-class, 22
- CrunchVariable-class, 22
- cube-computing, 22
- cube-methods, 23
- data.frame, 12
- dataset-extract, 24
- dataset-method, 25
- dataset-to-R, 26
- dataset-update, 26
- dataset-variables, 28
- DatasetGroup (DatasetOrder-class), 28
- DatasetGroup-class
 - (DatasetOrder-class), 28
- DatasetOrder (DatasetOrder-class), 28
- DatasetOrder-class, 28
- datasets, 29

- datasets<- (datasets), 29
- DatetimeVariable
 - (CrunchVariable-class), 22
- DatetimeVariable-class
 - (CrunchVariable-class), 22
- dedupeOrder, 30
- delete, 30, 32, 42
- delete, ANY-method (delete), 30
- delete, CrunchDataset-method (delete), 30
- delete, CrunchProject-method (delete), 30
- delete, CrunchTeam-method (delete), 30
- delete, CrunchVariable-method (delete), 30
- delete, DatasetTuple-method
 - (entity, CrunchProject-method), 40
- delete, ShojiObject-method (delete), 30
- delete, ShojiTuple-method
 - (entity, CrunchProject-method), 40
- delete, VariableTuple-method (delete), 30
- deleteDataset, 31, 31
- deleteSubvariable, 84
- deleteSubvariable (deleteSubvariables), 32
- deleteSubvariables, 32
- deleteVariable (deleteVariables), 33
- deleteVariables, 33
- deriveArray (makeArray), 60
- describe, 33
- describe-catalog
 - (names, BatchCatalog-method), 64
- describe-category, 35
- description (describe), 33
- description, CrunchDataset-method
 - (describe), 33
- description, CrunchVariable-method
 - (describe), 33
- description<- (describe), 33
- description<-, CrunchDataset-method
 - (describe), 33
- description<-, CrunchVariable-method
 - (describe), 33
- descriptions
 - (names, BatchCatalog-method), 64
- descriptions, VariableCatalog-method
 - (names, BatchCatalog-method), 64
- descriptions, VersionCatalog-method
 - (names, BatchCatalog-method), 64
- descriptions<-
 - (names, BatchCatalog-method), 64
- descriptions<-, VariableCatalog-method
 - (names, BatchCatalog-method), 64
- dichotomize, 36, 36
- dichotomize, CategoricalArrayVariable, ANY-method
 - (dichotomize), 36
- dichotomize, CategoricalVariable, ANY-method
 - (dichotomize), 36
- dichotomize, Categories, character-method
 - (dichotomize), 36
- dichotomize, Categories, logical-method
 - (dichotomize), 36
- dichotomize, Categories, numeric-method
 - (dichotomize), 36
- dim, 38
- dim, CrunchDataset-method (dim-dataset), 37
- dim, CubeDims-method (cube-methods), 23
- dim-dataset, 37
- dimnames, CubeDims-method
 - (cube-methods), 23
- dropRows, 38
- duplicate, 30, 39, 39
- duplicate, CrunchExpr-method
 - (duplicate), 39
- duplicate, CrunchVariable-method
 - (duplicate), 39
- duplicates, 30
- duplicates (ShojiOrder-slots), 81
- duplicates, OrderGroup-method
 - (ShojiOrder-slots), 81
- duplicates, ShojiOrder-method
 - (ShojiOrder-slots), 81
- duplicates, VariableCatalog-method
 - (ShojiOrder-slots), 81
- duplicates<- (ShojiOrder-slots), 81
- duplicates<-, OrderGroup, logical-method
 - (ShojiOrder-slots), 81
- duplicates<-, ShojiOrder, logical-method
 - (ShojiOrder-slots), 81
- duplicates<-, VariableCatalog, logical-method
 - (ShojiOrder-slots), 81
- emails (names, BatchCatalog-method), 64
- emails, ShojiCatalog-method
 - (names, BatchCatalog-method), 64
- embedCrunchBox, 39

- endDate (describe), 33
- endDate, CrunchDataset-method
 - (describe), 33
- endDate<- (describe), 33
- endDate<- , CrunchDataset-method
 - (describe), 33
- entities (ShojiOrder-slots), 81
- entities, list-method
 - (ShojiOrder-slots), 81
- entities, OrderGroup-method
 - (ShojiOrder-slots), 81
- entities, ShojiOrder-method
 - (ShojiOrder-slots), 81
- entities<- (ShojiOrder-slots), 81
- entities<- , OrderGroup-method
 - (ShojiOrder-slots), 81
- entities<- , ShojiOrder-method
 - (ShojiOrder-slots), 81
- entity (entity, CrunchProject-method), 40
- entity, CrunchProject-method, 40
- entity, CrunchVariable-method
 - (entity, CrunchProject-method), 40
- entity, DatasetTuple-method
 - (entity, CrunchProject-method), 40
- entity, VariableTuple-method
 - (entity, CrunchProject-method), 40
- exclusion, 38, 42
- exclusion<- (exclusion), 42
- exportDataset, 43
- expressions, 43
- extendDataset (joinDatasets), 56

- filter-catalog, 45
- filter-methods, 46
- filters (filter-catalog), 45
- filters, CrunchDataset-method
 - (filter-catalog), 45
- filters<- (filter-catalog), 45
- filters<- , CrunchDataset-method
 - (filter-catalog), 45
- flattenOrder, 46
- flipArrays, 47
- forkDataset, 47

- getAccountUserCatalog, 48
- getTeams, 48, 62

- grouped, 49, 82

- hiddenVariables, 49
- hiddenVariables<- (hideVariables), 50
- hide, 30, 31, 33, 50, 51
- hide, CrunchVariable-method (hide), 50
- hide, VariableCatalog-method (hide), 50
- hideVariables, 50
- http-methods, 51

- icon (project-icon), 71
- icon<- (project-icon), 71
- id (describe-category), 35
- id, Category-method (describe-category), 35
- id, CrunchDataset-method (describe), 33
- ids (Categories-class), 13
- ids, Categories-method
 - (Categories-class), 13
- ids, VariableCatalog-method
 - (names, BatchCatalog-method), 64
- ids<- (Categories-class), 13
- ids<- , Categories-method
 - (Categories-class), 13
- index, 52
- index, ShojiCatalog-method (index), 52
- index<- (index), 52
- index<- , ShojiCatalog-method (index), 52
- is-na-categories, 52
- is.archived
 - (is.archived, DatasetCatalog-method), 53
- is.archived, CrunchDataset-method
 - (is.archived, DatasetCatalog-method), 53
- is.archived, DatasetCatalog-method, 53
- is.archived<-
 - (is.archived, DatasetCatalog-method), 53
- is.archived<- , CrunchDataset, logical-method
 - (is.archived, DatasetCatalog-method), 53
- is.archived<- , DatasetCatalog, logical-method
 - (is.archived, DatasetCatalog-method), 53
- is.Array (is.dataset), 54
- is.CA (is.dataset), 54
- is.Categorical (is.dataset), 54
- is.CategoricalArray (is.dataset), 54

- is.dataset, 54
- is.Datetime (is.dataset), 54
- is.dichotomized (dichotomize), 36
- is.dichotomized, Categories-method (dichotomize), 36
- is.draft
 - (is.archived, DatasetCatalog-method), 53
- is.draft, CrunchDataset-method (is.archived, DatasetCatalog-method), 53
- is.draft, DatasetCatalog-method (is.archived, DatasetCatalog-method), 53
- is.draft<-
 - (is.archived, DatasetCatalog-method), 53
- is.draft<-, CrunchDataset, logical-method (is.archived, DatasetCatalog-method), 53
- is.draft<-, DatasetCatalog, logical-method (is.archived, DatasetCatalog-method), 53
- is.editor, 55
- is.editor, MemberCatalog-method (is.editor), 55
- is.editor, PermissionCatalog-method (is.editor), 55
- is.editor, PermissionTuple-method (is.editor), 55
- is.editor<- (is.editor), 55
- is.editor<-, MemberCatalog, logical-method (is.editor), 55
- is.MR (is.dataset), 54
- is.Multiple (is.dataset), 54
- is.MultipleResponse (is.dataset), 54
- is.na, Categories-method (is-na-categories), 52
- is.na, Category-method (is-na-categories), 52
- is.na, CrunchVariable-method (expressions), 43
- is.na, CubeDims-method (cube-methods), 23
- is.na<-, Categories, character-method (is-na-categories), 52
- is.na<-, Categories, logical-method (is-na-categories), 52
- is.na<-, Category, logical-method (is-na-categories), 52
- is.na<-, CrunchVariable, ANY-method (variable-update), 92
- is.Numeric (is.dataset), 54
- is.public (filter-methods), 46
- is.public, CrunchFilter-method (filter-methods), 46
- is.public<- (filter-methods), 46
- is.public<-, CrunchFilter-method (filter-methods), 46
- is.published
 - (is.archived, DatasetCatalog-method), 53
- is.published, CrunchDataset-method (is.archived, DatasetCatalog-method), 53
- is.published, DatasetCatalog-method (is.archived, DatasetCatalog-method), 53
- is.published<-
 - (is.archived, DatasetCatalog-method), 53
- is.published<-, CrunchDataset, logical-method (is.archived, DatasetCatalog-method), 53
- is.published<-, DatasetCatalog, logical-method (is.archived, DatasetCatalog-method), 53
- is.selected (describe-category), 35
- is.selected, Category-method (describe-category), 35
- is.shoji (is.dataset), 54
- is.Text (is.dataset), 54
- is.variable (is.dataset), 54
- joinDatasets, 56
- jsonprep (tojson-crunch), 87
- jsonprep, ANY-method (tojson-crunch), 87
- jsonprep, Categories-method (tojson-crunch), 87
- jsonprep, list-method (tojson-crunch), 87
- jsonprep, OrderGroup-method (tojson-crunch), 87
- jsonprep, ShojiOrder-method (tojson-crunch), 87
- length, OrderGroup-method (ShojiOrder-length), 80

- length, ShojiCatalog-method
(catalog-length), 12
- length, ShojiOrder-method
(ShojiOrder-length), 80
- listDatasets, 31, 57
- lm, 26
- loadDataset, 57, 58
- locateEntity, 58
- lock, 59
- login, 59
- logout, 60

- makeArray, 60
- makeMR (makeArray), 60
- margin.table, 22, 23
- margin.table (cube-computing), 22
- margin.table, CrunchCube-method
(cube-computing), 22
- match, 45
- max, 21
- max (crunch-uni), 20
- max, CrunchVariable-method (crunch-uni),
20
- max, DatetimeVariable-method
(crunch-uni), 20
- max, NumericVariable-method
(crunch-uni), 20
- me, 61
- mean, 21
- mean (crunch-uni), 20
- mean, CrunchVariable-method
(crunch-uni), 20
- mean, NumericVariable-method
(crunch-uni), 20
- median, 21
- median (crunch-uni), 20
- median, CrunchVariable-method
(crunch-uni), 20
- median, NumericVariable-method
(crunch-uni), 20
- members
(members<-, CrunchTeam, MemberCatalog-method),
61
- members, CrunchProject-method
(members<-, CrunchTeam, MemberCatalog-method),
61
- members, CrunchTeam-method
(members<-, CrunchTeam, MemberCatalog-method),
61
- members<-
(members<-, CrunchTeam, MemberCatalog-method),
61
- members<-, CrunchProject, character-method
(members<-, CrunchTeam, MemberCatalog-method),
61
- members<-, CrunchProject, MemberCatalog-method
(members<-, CrunchTeam, MemberCatalog-method),
61
- members<-, CrunchTeam, character-method
(members<-, CrunchTeam, MemberCatalog-method),
61
- members<-, CrunchTeam, MemberCatalog-method,
61
- merge, 56
- merge.CrunchDataset (joinDatasets), 56
- mergeFork, 62
- min, 21
- min (crunch-uni), 20
- min, CrunchVariable-method (crunch-uni),
20
- min, DatetimeVariable-method
(crunch-uni), 20
- min, NumericVariable-method
(crunch-uni), 20
- moveToGroup, 63
- moveToGroup<- (moveToGroup), 63
- MultipleResponseVariable
(CrunchVariable-class), 22
- MultipleResponseVariable-class
(CrunchVariable-class), 22

- na-omit-categories, 64
- na.omit, Categories-method
(na-omit-categories), 64
- name (describe), 33
- name, Category-method
(describe-category), 35
- name, CrunchDataset-method (describe), 33
- name, CrunchVariable-method (describe),
33
- name, OrderGroup-method
(ShojiOrder-slots), 81
- name, ShojiObject-method (describe), 33
- name, ShojiTuple-method
(entity, CrunchProject-method),
33
- name<- (describe), 33

- name<- ,Category-method
(describe-category), 35
- name<- ,CrunchDataset-method (describe),
33
- name<- ,CrunchVariable-method
(describe), 33
- name<- ,NULL-method (describe-category),
35
- name<- ,OrderGroup-method
(ShojiOrder-slots), 81
- name<- ,ShojiTuple-method
(entity,CrunchProject-method),
40
- names, 66
- names,BatchCatalog-method, 64
- names,CategoricalArrayVariable-method
(names,BatchCatalog-method), 64
- names,Categories-method
(Categories-class), 13
- names,CrunchDataset-method
(names,BatchCatalog-method), 64
- names,OrderGroup-method
(ShojiOrder-slots), 81
- names,ShojiCatalog-method
(names,BatchCatalog-method), 64
- names,ShojiOrder-method
(ShojiOrder-slots), 81
- names,VersionCatalog-method
(names,BatchCatalog-method), 64
- names<- ,Categories-method
(Categories-class), 13
- names<- ,ShojiCatalog-method
(names,BatchCatalog-method), 64
- names<- ,ShojiOrder-method
(ShojiOrder-slots), 81
- ncol,CrunchDataset-method
(dim-dataset), 37
- newDataset, 66, 67, 68
- newDatasetByColumn, 67, 67, 68
- newDatasetByCSV, 67, 67
- newDatasetFromFile, 68
- notes (describe), 33
- notes,CrunchDataset-method (describe),
33
- notes,CrunchVariable-method (describe),
33
- notes,VariableCatalog-method
(names,BatchCatalog-method), 64
- notes<- (describe), 33
- notes<- ,CrunchDataset-method
(describe), 33
- notes<- ,CrunchVariable-method
(describe), 33
- notes<- ,VariableCatalog-method
(names,BatchCatalog-method), 64
- NumericVariable (CrunchVariable-class),
22
- NumericVariable-class
(CrunchVariable-class), 22
- ordering, 68
- ordering,CrunchDataset-method
(ordering), 68
- ordering,DatasetCatalog-method
(ordering), 68
- ordering,VariableCatalog-method
(ordering), 68
- ordering<- (ordering), 68
- ordering<- ,CrunchDataset-method
(ordering), 68
- ordering<- ,DatasetCatalog-method
(ordering), 68
- ordering<- ,VariableCatalog-method
(ordering), 68
- owner (dataset-owner), 25
- owner,CrunchDataset-method
(dataset-owner), 25
- owner<- (dataset-owner), 25
- owner<- ,CrunchDataset-method
(dataset-owner), 25
- ownerNames (owners), 69
- owners, 69
- permissions, 70
- permissions,CrunchDataset-method
(permissions), 70
- preCrunchBoxCheck, 22, 70
- project-icon, 71
- projects, 71
- prop.table, 22, 23
- prop.table (cube-computing), 22
- prop.table,CrunchCube-method
(cube-computing), 22
- publish
(is.archived,DatasetCatalog-method),
53

- refresh, [72](#), [90](#)
- refresh, CrunchDataset-method (refresh), [72](#)
- refresh, CrunchVariable-method (refresh), [72](#)
- refresh, ShojiObject-method (refresh), [72](#)
- refresh, ShojiTuple-method (entity, CrunchProject-method), [40](#)
- removeEmptyGroups, [72](#)
- restoreVersion, [73](#), [74](#), [96](#)
- rollup (expressions), [43](#)
- round, [23](#)
- round, CrunchCube-method (cube-computing), [22](#)

- saveVersion, [73](#), [74](#), [96](#)
- sd, [21](#)
- sd (crunch-uni), [20](#)
- sd, CrunchVariable-method (crunch-uni), [20](#)
- sd, NumericVariable-method (crunch-uni), [20](#)
- self, [74](#)
- self, CrunchVariable-method (self), [74](#)
- self, ShojiObject-method (self), [74](#)
- self, ShojiTuple-method (entity, CrunchProject-method), [40](#)
- session, [75](#), [76](#)
- session-extract, [75](#)
- settings, [76](#)
- settings<- (settings), [76](#)
- share, [77](#), [90](#)
- ShojiObject (ShojiObject-class), [77](#)
- ShojiObject-class, [77](#)
- ShojiOrder-extract, [78](#)
- ShojiOrder-length, [80](#)
- ShojiOrder-slots, [81](#)
- show, [83](#)
- show, Categories-method (show-crunch), [82](#)
- show, Category-method (show-crunch), [82](#)
- show, CrunchCube-method (show-crunch), [82](#)
- show, CrunchExpr-method (show-crunch), [82](#)
- show, CrunchLogicalExpr-method (show-crunch), [82](#)
- show, CrunchVariable-method (show-crunch), [82](#)
- show, OrderGroup-method (show-crunch), [82](#)
- show, ShojiObject-method (show-crunch), [82](#)
- show-crunch, [82](#)
- startDate (describe), [33](#)
- startDate, CrunchDataset-method (describe), [33](#)
- startDate<- (describe), [33](#)
- startDate<- , CrunchDataset-method (describe), [33](#)
- subset, CrunchDataset-method (dataset-extract), [24](#)
- Subvariables, [64](#), [66](#)
- Subvariables (Subvariables-class), [83](#)
- subvariables (Subvariables-class), [83](#)
- subvariables, CategoricalArrayVariable-method (Subvariables-class), [83](#)
- subvariables, VariableTuple-method (Subvariables-class), [83](#)
- Subvariables-class, [83](#)
- subvariables<- (Subvariables-class), [83](#)
- subvariables<- , CategoricalArrayVariable, ANY-method (Subvariables-class), [83](#)
- subvariables<- , CategoricalArrayVariable, Subvariables-method (Subvariables-class), [83](#)
- subvars-extract, [84](#)

- table, [19](#), [86](#), [86](#)
- teams, [48](#)
- teams (members<- , CrunchTeam, MemberCatalog-method), [61](#)
- temp.option (temp.options), [87](#)
- temp.options, [87](#)
- TextVariable (CrunchVariable-class), [22](#)
- TextVariable-class (CrunchVariable-class), [22](#)
- timestamps (names, BatchCatalog-method), [64](#)
- timestamps, VersionCatalog-method (names, BatchCatalog-method), [64](#)
- toJSON, [87](#), [88](#)
- toJSON (tojson-crunch), [87](#)
- tojson-crunch, [87](#)
- toVariable, [88](#), [94](#)
- toVariable, character-method (toVariable), [88](#)
- toVariable, CrunchExpr-method (toVariable), [88](#)
- toVariable, Date-method (toVariable), [88](#)

- toVariable, factor-method (toVariable), 88
- toVariable, logical-method (toVariable), 88
- toVariable, numeric-method (toVariable), 88
- toVariable, POSIXt-method (toVariable), 88
- toVariable, VariableDefinition-method (toVariable), 88
- tuple-methods
 - (entity, CrunchProject-method), 40
- type, 89
- type, CrunchVariable-method (type), 89
- type, ShojiTuple-method
 - (entity, CrunchProject-method), 40
- type<- (type), 89
- type<-, CrunchVariable-method (type), 89
- types (names, BatchCatalog-method), 64
- types, VariableCatalog-method
 - (names, BatchCatalog-method), 64

- unbind, 90
- undichotomize (dichotomize), 36
- undichotomize, CategoricalArrayVariable-method
 - (dichotomize), 36
- undichotomize, CategoricalVariable-method
 - (dichotomize), 36
- undichotomize, Categories-method
 - (dichotomize), 36
- ungrouped (grouped), 49
- unhide (hide), 50
- unhide, CrunchVariable-method (hide), 50
- unhide, VariableCatalog-method (hide), 50
- unhideVariables (hideVariables), 50
- unlock (lock), 59
- unshare, 77, 90
- updateDatasetList, 91

- value (describe-category), 35
- value, Category-method
 - (describe-category), 35
- value<- (describe-category), 35
- value<-, Category-method
 - (describe-category), 35
- values (Categories-class), 13
- values, Categories-method
 - (Categories-class), 13
- values<- (Categories-class), 13
- values<-, Categories-method
 - (Categories-class), 13
- var-categories, 91
- VarDef (VariableDefinition), 94
- variable-extract
 - ([, CrunchExpr, CrunchLogicalExpr, ANY-method), 98
- variable-to-R
 - (as.vector, CrunchExpr-method), 6
- variable-update, 92
- VariableCatalog, 68, 95
- VariableCatalog
 - (VariableCatalog-class), 94
- VariableCatalog-class, 94
- VariableDefinition, 5, 16, 94
- VariableGroup (VariableOrder-class), 95
- VariableGroup-class
 - (VariableOrder-class), 95
- variableMetadata, 95
- VariableOrder, 49, 68, 82, 94
- VariableOrder (VariableOrder-class), 95
- VariableOrder-class, 95
- variables (dataset-variables), 28
- variables, CrunchDataset-method
 - (dataset-variables), 28
- variables<- (dataset-variables), 28
- variables<-, CrunchDataset, VariableCatalog-method
 - (dataset-variables), 28
- versions, 73, 74, 96

- weight, 19, 96
- weight<- (weight), 96
- which, 97
- which, CrunchLogicalExpr-method (which), 97
- with-context-manager, 17, 18, 87, 97
- with.contextManager
 - (with-context-manager), 97
- write.csv, CrunchDataset-method
 - (exportDataset), 43

- xtabs, 19