

Package ‘mudata’

February 27, 2017

Title Interchange Tools for Multi-Parameter Spatiotemporal Data

Version 0.1

Author Authors@R: person(“Dewey”, “Dunnington”, role = c(“aut”, “cre”),
email = “dewey@fishandwhistle.net”)

Maintainer Dewey Dunnington <dewey@fishandwhistle.net>

Description Formatting and structuring multi-parameter spatiotemporal data
is often a time-consuming task. This package offers functions and data structures
designed to easily organize and visualize these data for applications in geology,
paleolimnology, dendrochronology, and paleoclimate.

Imports reshape2, ggplot2, plyr, dplyr, jsonlite (>= 1.2), methods

Depends R (>= 3.2.2)

Suggests testthat

License GPL-2

URL <https://github.com/paleolimbot/mudata>

BugReports <https://github.com/paleolimbot/mudata/issues>

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-27 08:15:36

R topics documented:

mudata-package	2
aggregate.qtag.long	3
as.qtag	4
autoplot.mudata	5
biplot.mudata	6
condense.tags	6
expand.tags	7

group	8
id.vars	9
is.summarised	9
kentvillegreenwood	10
long	11
longbiplot	12
measure.vars	13
mudata	13
parallel.melt	15
pocmaj	16
pocmajsum	16
qualifierdata	16
qualifierplot	17
rbind.mudata	18
rbind.qtag.long	19
rename.cols	19
rename.datasets	20
rename.values	21
subset.mudata	22
summary.mudata	23
tag.vars	23
tagdata	24
validate.mudata	25
valuedata	25
wide	26
write.mudata	27
Index	29

mudata-package	<i>A (Mostly) Universal Data Format for Multi-Parameter, Spatiotemporal Data</i>
----------------	--

Description

The 'mudata' package for R is a set of tools to create, manipulate, and visualize multi-parameter, spatiotemporal data. Data of this type includes all data where multiple parameters (e.g. wind speed, precipitation, temperature) are measured along a common axis (e.g. time, depth) at discrete locations (e.g. climate stations). These data include long-term climate data collected from climate stations, paleolimnological data, ice core data, and ocean core data among many others. Data of this type is often voluminous and difficult to organize given its multi-dimensional nature. The (mostly) universal data (mudata) format is an attempt to organize these data in a common way to facilitate their documentation and comparison.

Details

The (mostly) universal data format is a collection of five (or more) tables, one of which contains the data in a molten form (see `reshape2::melt`). The easiest way to visualize a `mudata` object is to inspect the `kentvillegreenwood` dataset within the package. This object is a collection of daily observations from Kentville, Nova Scotia, and Greenwood, Nova Scotia from July and August, 1999.

See Also

[mudata](#)

Examples

```
# inspect the example dataset
data(kentvillegreenwood)
plot(kentvillegreenwood)

# create a mudata object from a wide data.frame
library(reshape2)
data("pocmajsum")
pocmajwide <- pocmajsum[c("core", "depth", "Ca", "V", "Ti")]
pocmajwide <- rename.cols(pocmajwide, "core"="location", "depth"="x")
pocmajlong <- melt(pocmajwide, id.vars=c("location", "x"), variable.name = "param",
                  value.name="value")
md <- mudata(pocmajlong)
plot(md, yvar="x")
```

aggregate.qtag.long *Aggregate/Summarise a qualifier/tag structure*

Description

Summarises a [qtag](#) object such that one value exists for every unique qualifier combination. This is useful for summarising replicates.

Usage

```
## S3 method for class 'qtag.long'
aggregate(x, ..., force = TRUE)

## S3 method for class 'qtag.wide'
aggregate(x, ..., force = TRUE)
```

Arguments

x	A <code>qtag</code> object
...	A parameter including at least one unnamed parameter for summarising values. Additional parameters may be used for aggregating a long format.
force	Use <code>force=FALSE</code> to only aggregate if the object is not already summarised.

Value

A (possibly unchanged) `qtag` object

Examples

```
data(pocmaj)
pocmajqt <- as.qtag(pocmaj, id.vars=c("core", "depth"))
aggregate(pocmajqt)
aggregate(pocmajqt, mean)
aggregate(long(pocmajqt), mean, sd, length)
```

as.qtag

Convert an object to a qualifier/tag structure

Description

A qualifer/tag structure is an experimental version of a data frame that stores `id.vars` and `measure.vars` with the data structure, so it can be handled in a more automatic way. This functionality is experimental.

Usage

```
as.qtag(df, id.vars = NA, measure.vars = NA, tag.vars = NA,
        quiet = FALSE)
```

```
qtag(df, ...)
```

Arguments

df	A <code>data.frame</code> or similar object
id.vars	Column names of qualifying <code>measure.vars</code> (NA to guess)
measure.vars	Column names containing the values of interest (NA to guess)
tag.vars	Column names of tag values
quiet	Use <code>quiet=TRUE</code> to suppress error messages
...	Passed to/from methods

Value

An object of type `qtag`, which is essentially the unchanged input with `id.vars`, `measure.vars`, and `tag.vars` information attached.

Examples

```
data("pocmaj")
pocmaj <- as.qtag(pocmaj, id.vars = c("core", "depth"))
long(pocmaj)
aggregate(pocmaj)
aggregate(long(pocmaj))
```

autoplot.mudata	<i>Autoplot a mudata object</i>
-----------------	---------------------------------

Description

Produces a quick graphical summary of a `mudata` object using the `ggplot` framework. The `subset` argument is quite powerful for filtering, but does not affect the order of appearance. For this, use [subset.mudata](#), which by default orders the datasets, locations, and parameters passed into the function.

Usage

```
## S3 method for class 'mudata'
autoplot(x, ...)

## S3 method for class 'mudata'
plot(x, ...)
```

Arguments

<code>x</code>	A mudata object
<code>...</code>	Passed on to qualifierplot

Value

A `ggplot` object

Examples

```
data(kentvillegreenwood)
plot(kentvillegreenwood)
library(ggplot2)
autoplot(kentvillegreenwood)
```

biplot.mudata	<i>Biplot a mudata object</i>
---------------	-------------------------------

Description

Uses the ggplot framework and facet_grid to produce a biplot of a mudata object.

Usage

```
## S3 method for class 'mudata'
biplot(x, ..., namecolumn = "param")
```

Arguments

x	A mudata object
...	passed to longbiplot
namecolumn	The column that contains the names for biplotting

Value

a ggplot object

Examples

```
data(kentvillegreenwood)
biplot(kentvillegreenwood, c("meantemp", "maxtemp", "mintemp"), col="location")
```

condense.tags	<i>Condense multiple columns to a single JSON column</i>
---------------	--

Description

Performs the opposite of [expand.tags](#): collapses the data in tagcolumns to a single column (called tagcolumn) that contains a JSON representation of the data that was previously in the tag columns at that row.

Usage

```
condense.tags(x, ...)

## S3 method for class 'data.frame'
condense.tags(x, tagcolumns, tagcolumn = "tags", ...)

## S3 method for class 'mudata'
condense.tags(x, ...)
```

Arguments

x	a data frame with tag columns
...	passed to/from methods
tagcolumns	column names to be condensed to JSON
tagcolumn	the column in which to store JSON

Value

A modified data.frame

Examples

```
data(pocmaj)
condense.tags(pocmaj, tagcolumns = c("Ca", "Ti", "V"))
```

expand.tags	<i>Expand JSON to multiple columns</i>
-------------	--

Description

Expands a character vector of JSON values or a data.frame with a column tagcolumn to a data.frame with columns for each key in the vector of JSON objects. If x is a mudata object, the operation will be performed on the entire object.

Usage

```
expand.tags(x, ...)

## S3 method for class 'data.frame'
expand.tags(x, tagcolumn = "tags", lazy = FALSE, ...)

## S3 method for class 'mudata'
expand.tags(x, ...)
```

Arguments

x	a vector of JSON values or a data.frame with a 'tags' column
...	Passed to/from methods
tagcolumn	the column containing json
lazy	Don't preform expansion if tagcolumn exists in x

Value

A data.frame with columns added

Examples

```
data(pocmaj)
condensed <- condense.tags(pocmaj, tagcolumns = c("Ca", "Ti", "V"))
expand.tags(condensed)
```

group	<i>Group a qualifier/tag structure</i>
-------	--

Description

Essentially a shortcut for grouping a [qtag](#) object by its id.vars

Usage

```
group(qtag, quiet = FALSE)
```

Arguments

qtag	A qualifier/tag structure
quiet	Pass TRUE to suppress warnings on coercion to a qualifier tag structure.

Value

A dplyr grouped data frame

Examples

```
data(pocmaj)
library(dplyr)
pocmajqt <- as.qtag(pocmaj, id.vars = c("core", "depth"))
pocmajqt %>% group() %>% summarise(mean(Ca))
# equivalent to
pocmaj %>% group_by(core, depth) %>% summarise(mean(Ca))
```

id.vars	<i>Extract value qualifier names from a qualifier/tag structure</i>
---------	---

Description

Extract value qualifier names from a qualifier/tag structure

Usage

```
id.vars(x, ...)  
  
## S3 method for class 'qtag'  
id.vars(x, ...)  
  
## S3 method for class 'grouped_df'  
id.vars(x, ...)  
  
## S3 method for class 'data.frame'  
id.vars(x, ..., quiet = FALSE)
```

Arguments

x	A qtag object
...	Passed to/from methods
quiet	Supress error messages on coercion to a qualifier/tag structure

Value

A vector of qualifier column names

Examples

```
data(pocmaj)  
pocmaj <- as.qtag(pocmaj, id.vars=c("core", "depth"))  
id.vars(pocmaj)
```

is.summarised	<i>Extract if the argument is already summarised</i>
---------------	--

Description

Checks the `nrow()` of sub-data frames produced by `group_by()`. Returns `all(lengths==1)`.

Usage

```
is.summarised(x, quiet = FALSE)
```

Arguments

x	The object
quiet	Pass TRUE to suppress warnings on coercion to a qualifier/tag structure.

Value

TRUE if the argument is summarised, FALSE otherwise

Examples

```
data(pocmaj)
pocmaj <- as.qtag(pocmaj, c("core", "depth"))
is.summarised(pocmaj)
is.summarised(aggregate(pocmaj))
```

kentvillegreenwood *Kentville/Greenwood Climate Data*

Description

Climate data for Kentville and Greenwood (Nova Scotia) for July and August of 1999.

Usage

```
kentvillegreenwood
```

Format

A `mudata` object (list with elements data, locations, params, datasets, columns)

Source

Environment Canada via the 'rclimateca' package. <http://climate.weather.gc.ca/>

long	<i>Convert data to long format</i>
------	------------------------------------

Description

Convert data to long format

Usage

```
long(x, ...)  
  
## Default S3 method:  
long(x, ...)  
  
## S3 method for class 'qtag.long'  
long(x, ...)  
  
## S3 method for class 'qtag.wide'  
long(x, varname = "param", quiet = FALSE, ...)
```

Arguments

x	A qtag object
...	Passed to other methods
varname	The column name in which column names will be stored.
quiet	Supress warning messages on coersion to a qualifier/tag structure.

Value

A (possibly unchanged) `qtag.long` data.frame

Examples

```
data(pocmaj)  
pocmaj <- as.qtag(pocmaj, id.vars = c("core", "depth"))  
long(pocmaj)
```

longbiplot

Biplot a molten data frame using facets

Description

Uses the ggplot framework and `facet_grid` to produce biplots of a molten data frame.

Usage

```
longbiplot(x, id.vars, measure.var, namesx = NULL, namesy = NULL,
  namecolumn = NULL, errors = NULL, labeller = ggplot2::label_value,
  validate = TRUE, ...)
```

```
## S3 method for class 'qtag.long'
biplot(x, ...)
```

```
## S3 method for class 'qtag.wide'
biplot(x, ...)
```

Arguments

<code>x</code>	the object to biplot
<code>id.vars</code>	the columns that identify a single value
<code>measure.var</code>	The column containing the values to plot
<code>namesx</code>	The names to be included in the x axes, or all the names to be included
<code>namesy</code>	The names to be included on the y axes, or NULL for all possible combinations of <code>namesx</code> .
<code>namecolumn</code>	The column where <code>namesx</code> and <code>namesy</code> are to be found
<code>errors</code>	The column containing the errors. Use NULL for default ("err" if column exists or none otherwise), or NA to suppress.
<code>labeller</code>	The labeller to use to label facets (may want to use <code>label_parsed</code> to use plotmath-style labels)
<code>validate</code>	Ensure <code>id.vars</code> identify unique values
<code>...</code>	passed to <code>aes_string()</code>

Examples

```
data(pocmaj)
qt <- as.qtag(pocmaj)
biplot(qt, color="core")
```

measure.vars	<i>Extract value column names from a qualifier/tag structure</i>
--------------	--

Description

Extract value column names from a qualifier/tag structure

Usage

```
measure.vars(x, quiet = FALSE)
```

Arguments

x	A qtag object
quiet	Suppress error messages on coercion to a qualifier/tag structure

Value

A vector of value column names

Examples

```
data(pocmaj)
pocmaj <- as.qtag(pocmaj, id.vars=c("core", "depth"))
measure.vars(pocmaj)
```

mudata	<i>Create a MuData object</i>
--------	-------------------------------

Description

Create an object describing multi-parameter spatiotemporal data in the (mostly) universal data format. This format is a collection of tables as described below. For an example of data already in this format, see the [kentvillegreenwood](#) dataset.

Usage

```
mudata(data, locations = NULL, params = NULL, datasets = NULL,
        columns = NULL, dataset.id = "default", location.id = "default",
        defactorize = TRUE, validate = TRUE, expand.tags = TRUE,
        retype = FALSE)
```

Arguments

<code>data</code>	The data table, which is a molten data frame containing the columns (at least) 'dataset', 'location', 'x', 'param', and 'value'. The 'dataset' column can be omitted if there is only one dataset contained in the object (its name can be specified by passing the parameter <code>dataset.id</code>). The 'location' column can be omitted if there is only data for one dataset and one location (its name can be specified by passing the parameter <code>location.id</code>).
<code>locations</code>	The locations table, which is a data frame containing the columns (at least) 'dataset', and 'location'. If omitted, it will be created automatically using all unique dataset/location combinations.
<code>params</code>	The params table, which is a data frame containing the columns (at least) 'dataset', and 'param'. If omitted, it will be created automatically using all unique dataset/param combinations.
<code>datasets</code>	The datasets table, which is a data frame containing the column (at least) 'dataset'. If omitted, it will be generated automatically using all unique datasets.
<code>columns</code>	The columns table, which is a data frame containing the columns (at least) 'dataset', 'table', and 'column'. If omitted, it will be created automatically using all dataset/table/column combinations.
<code>dataset.id</code>	The dataset id to use if the datasets table is omitted.
<code>location.id</code>	The location id if the locations table is omitted.
<code>defactorize</code>	Pass FALSE to suppress coercion of 'dataset', 'location', and 'param' columns to type 'character'.
<code>validate</code>	Pass FALSE to skip validation of input tables.
<code>expand.tags</code>	Pass FALSE to collapse non-required columns to a single column (called 'tags'), with key/value pairs in JSON format. See expand.tags .
<code>retype</code>	Pass TRUE to retype columns based on the 'type' column of the 'columns' table. This is useful when reading data from disk, where date/time columns may be stored as text.

Value

A mudata object

Examples

```
library(reshape2)
library(dplyr)
data(pocmaj)

# melt data and summarise replicates
datatable <- pocmaj %>%
  melt(id.vars=c("core", "depth"), variable.name="param") %>%
  group_by(core, param, depth) %>%
  summarise(sd=mean(value), value=mean(value)) %>%
  rename.cols("depth"="x", "core"="location")
```

```
# create mdata object
md <- mdata(datatable)
summary(md)
plot(md, yvar="x", geom=c("path", "point"))
```

parallel.melt

Melt multiple sets of columns in parallel

Description

Essentially this is a wrapper around `reshape2::melt.data.frame` that is able to `cbind` several melt operations. This is useful when a wide data frame contains uncertainty or flag information in paired columns.

Usage

```
parallel.melt(x, id.vars, ..., variable.name = "param",
             factorsAsStrings = TRUE)
```

Arguments

<code>x</code>	A <code>data.frame</code>
<code>id.vars</code>	vector of ID variable names
<code>...</code>	Named arguments specifying the <code>measure.vars</code> to be stored to the column name specified.
<code>variable.name</code>	Column name to use to store variables
<code>factorsAsStrings</code>	Control whether factors are converted to character when melted as measure variables.

Value

A `qtag.long` object

Examples

```
data(pocmajsum)
parallel.melt(pocmajsum,
             id.vars=c("core", "depth"),
             values=c("Ca", "Ti", "V"),
             err=c("Ca_sd", "Ti_sd", "V_sd"))
```

pocmaj

Pockwock Lake/Lake Major Elemental Sample Data

Description

A small example data.frame used to test structure methods.

Usage

```
pocmaj
```

Format

A data.frame containing multi-qualifier concentration data

pocmajsum

Pre-summarised Sample Data

Description

A small example data.frame of pre-summarised data; a summarised version of the [pocmaj](#) dataset.

Usage

```
pocmajsum
```

Format

A data.frame containing multi-qualifier data

qualifierdata

Extract qualifier column data from a qualifier/tag structure

Description

Extract qualifier column data from a qualifier/tag structure

Usage

```
qualifierdata(x)
```

Arguments

x A [qtag](#) object

Value

A [data.frame](#) of qualifier column data

Examples

```
data(pocmaj)
pocmaj <- as.qtag(pocmaj, id.vars=c("core", "depth"))
qualifierdata(pocmaj)
```

qualifierplot

Smart plotting of tidy data frames

Description

This function uses `ggplot` to plot a 'long' data frame with a few `id.vars`, or variables that identify the values in the value column. The function is optimised to plot multi-parameter spatiotemporal data either horizontally (time on the x axis) or vertically (time on the y axis). Facets are intended to be by parameter, which is guessed based on the right-most variable named in `id.vars`. In the case of a `qtag` object, many of these values are guessed. This is intended to produce a quick visual of an object to examine its contents.

Usage

```
qualifierplot(x, id.vars, measure.var, subset, xvar, yvar, facets,
  geom = "path", errors = "err", ...)
```

```
## S3 method for class 'qtag.long'
autoplot(x, ...)
```

```
## S3 method for class 'qtag.wide'
autoplot(x, ...)
```

```
## S3 method for class 'qtag.long'
plot(x, ...)
```

```
## S3 method for class 'qtag.wide'
plot(x, ...)
```

Arguments

<code>x</code>	A data.frame
<code>id.vars</code>	Columns that identify unique values
<code>measure.var</code>	Column that contains values to be plotted
<code>subset</code>	Subset to plot
<code>xvar</code>	Column to be used on the x-axis

yvar	Column to be used on the y-axis
facets	Column to be used as facetting variable
geom	GGPlot geometries to be used. Can be any combination of point, path, or line.
errors	The colum that contains uncertainty information
...	Passed on to aes_string()

Value

A ggplot object

Examples

```
data(pocmaj)
qualifierplot(pocmaj, c("core", "depth"), "Ca")

pocmajqt <- as.qtag(pocmaj, id.vars=c("core", "depth"))
plot(pocmajqt, geom=c("path", "point"))
plot(pocmajqt, subset=core=="MAJ-1" & param %in% c("Ca", "Ti"))
plot(pocmajqt, shape="core", geom=c("path", "point"))
plot(long(pocmajqt))

library(ggplot2)
autoplot(pocmajqt, col="core")
```

rbind.mudata	<i>Combine mudata objects</i>
--------------	-------------------------------

Description

Combine mudata objects

Usage

```
rbind.mudata(..., validate = TRUE)
```

Arguments

...	Mudata objects to combine
validate	Flag to validate final object

Value

A Mudata object

rbind.qtag.long	<i>Combine qualifier/tag structures</i>
-----------------	---

Description

Combine qualifier/tag structures

Usage

```
rbind.qtag.long(...)
```

```
rbind.qtag(...)
```

Arguments

... Objects to combine

Value

A qtag.long object

Examples

```
data(pocmaj)
pocmaj <- as.qtag(pocmaj, id.vars=c("core", "depth"))
newrow <- data.frame(core="POC-2", depth=6, Ca=2100, Ti=4100, V=45)
rbind(pocmaj, newrow)
```

rename.cols	<i>Replace/rename a column in an object</i>
-------------	---

Description

Essentially a thin convenience wrapper around `plyr::rename(x, list(...))`, except `qtag` objects have their `id.vars/tag.vars/measure.vars` attributes properly modified.

Usage

```
rename.cols(.data, ..., warn_missing = TRUE, warn_duplicated = TRUE)
```

```
## Default S3 method:
rename.cols(.data, ..., warn_missing = TRUE,
  warn_duplicated = TRUE)
```

```
## S3 method for class 'qtag'
rename.cols(.data, ..., warn_missing = TRUE,
  warn_duplicated = TRUE)
```


Arguments

md	A mudata object
...	Key/value pairs in the form "oldvalue"="newvalue"
apply_to	The tables which the rename operation should consider
warn_missing	Print a message if any old names are not actually present
.data	A mudata object
warn_duplicated	Print a message if any name appears more than once in x after the operation.

Value

A modified [mudata](#) object.

Examples

```
data(kentvillegreenwood)
md2 <- rename.datasets(kentvillegreenwood, ecclimate="avalley")
validate.mudata(md2)
md2 <- rename.locations(kentvillegreenwood, "GREENWOOD A"="Greenwood")
validate.mudata(md2)
md2 <- rename.params(kentvillegreenwood, maxtemp="Maximum Temperature")
validate.mudata(md2)
md2 <- rename.cols(kentvillegreenwood, latitude="lat", longitude="lon")
validate.mudata(md2)
```

rename.values	<i>Replace/rename values in a vector</i>
---------------	--

Description

This function replaces character values with new character values, which is useful when performing rename operations when values are held in character vectors.

Usage

```
rename.values(x, ..., defaultValue = x, warn_missing = TRUE)
```

Arguments

x	Vector of values to replace
...	Key/value pairs in the form oldvalue="newvalue"
defaultValue	A vector of values to use as the default should the value not be found in ...
warn_missing	Print a message if any old names are not actually present in x

Value

A vector with values replaced

Examples

```
x <- c("fish", "fish", "fish", "whistle")
rename.values(x, fish="newfish")
rename.values(x, whistle="newwhistle")
rename.values(x, fish="newfish", defaultValue="not a fish")
```

subset.mudata	<i>Subset a MuData object</i>
---------------	-------------------------------

Description

Subset a MuData object

Usage

```
## S3 method for class 'mudata'
subset(x, datasets = NULL, params = NULL,
       locations = NULL, validate = TRUE, defactorize = FALSE, ...)
```

Arguments

x	The object to subset
datasets	Vector of datasets to include
params	Vector of parameters to include
locations	Vector of locations to include
validate	Flag to validate output
defactorize	Pass FALSE to keep input columns as factors (may cause errors).
...	Arguments to/from methods

Value

A subsetted MuData object

summary.mudata	<i>Object summary for a mudata object</i>
----------------	---

Description

Returns a data.frame containing summary (by dataset, location, and param) statistics.

Usage

```
## S3 method for class 'mudata'
summary(object, ..., digits = NA)
```

```
## S3 method for class 'mudata'
print(x, ..., digits = 4)
```

Arguments

object	A mudata object
...	Unused
digits	The number of digits to be used for rounding, or NA to suppress rounding.
x	A mudata object

Value

A data.frame containing the summary

Examples

```
data(kentvillegreenwood)
summary(kentvillegreenwood, digits=2)
summary(kentvillegreenwood)
print(kentvillegreenwood, digits=2)
```

tag.vars	<i>Extract tag column names from a qualifier/tag structure</i>
----------	--

Description

Extract tag column names from a qualifier/tag structure

Usage

```
tag.vars(x)
```

Arguments

x A [qtag](#) object

Value

A vector of tag column names

Examples

```
data(pocmaj)
pocmaj <- as.qtag(pocmaj, id.vars=c("core", "depth"))
tag.vars(pocmaj)
```

tagdata

Extract tag column data from a qualifier/tag structure

Description

Extract tag column data from a qualifier/tag structure

Usage

```
tagdata(x)
```

Arguments

x A [qtag](#) object

Value

A [data.frame](#) of tag column data

Examples

```
data(pocmaj)
pocmaj <- as.qtag(pocmaj, id.vars=c("core", "depth"))
tagdata(pocmaj)
```

validate.mudata	<i>Validate a MUData object</i>
-----------------	---------------------------------

Description

Validates a MUData object by calling stop when an error is found.

Usage

```
validate.mudata(md)
```

Arguments

md An object of class 'mudata'

Value

TRUE, invisibly

Examples

```
data(kentvillegreenwood)
validate.mudata(kentvillegreenwood)
```

valuedata	<i>Extract value column data from a qualifier/tag structure</i>
-----------	---

Description

Extract value column data from a qualifier/tag structure

Usage

```
valuedata(x)
```

Arguments

x A [qtag](#) object

Value

A [data.frame](#) of value column data

Examples

```
data(pocmaj)
pocmaj <- as.qtag(pocmaj, id.vars=c("core", "depth"))
valuedata(pocmaj)
```

wide	<i>Convert data to wide format</i>
------	------------------------------------

Description

Convert data to wide format

Usage

```
wide(x, ...)
```

Default S3 method:

```
wide(x, ...)
```

S3 method for class 'qtag.wide'

```
wide(x, ...)
```

S3 method for class 'qtag.long'

```
wide(x, colvar, fun.aggregate, quiet = FALSE, ...)
```

Arguments

x	A qtag object or one that can be coerced to one.
...	Passed to other methods
colvar	The column that contains the names of the to-be columns
fun.aggregate	The aggregation function to be used if id.vars other than colvar identify more than one row each.
quiet	Supress warning messages on coercion to a qualifier/tag structure.

Value

A (possibly unchanged) `qtag.wide` data.frame

Examples

```
data(pocmaj)
pocmaj <- as.qtag(pocmaj, id.vars = c("core", "depth"))
pocmajlong <- long(pocmaj)
wide(pocmajlong)
wide(pocmaj)
```

write.mudata	<i>Read/Write mudata objects</i>
--------------	----------------------------------

Description

Read/Write mudata objects

Usage

```
write.mudata(md, filename, ...)
```

```
read.mudata(filename, ...)
```

```
write.mudata.zip(md, filename, overwrite = FALSE, expand.tags = TRUE,  
  validate = TRUE, ...)
```

```
read.mudata.zip(filename, validate = TRUE, expand.tags = TRUE,  
  retype = TRUE, load = c("data", "locations", "params", "datasets",  
  "columns"), ...)
```

```
write.mudata.json(md, filename, overwrite = FALSE, expand.tags = TRUE,  
  validate = TRUE, ...)
```

```
read.mudata.json(filename, validate = TRUE, expand.tags = TRUE,  
  retype = TRUE, load = c("data", "locations", "params", "datasets",  
  "columns"), ...)
```

Arguments

md	a mudata object
filename	file to read/write (can also be a directory)
...	passed to read/write.csv
overwrite	Pass TRUE to overwrite if zipfile already exists.
expand.tags	flag to expand tags to columns
validate	flag to validate mudata object upon read or before write
retype	Pass TRUE to retype columns based on the 'type' column of the 'columns' table.
load	a list of csv files (without the .csv extension) to load from the source.

Examples

```
data(kentvillegreenwood)  
# read/write to zip  
outfile <- tempfile(fileext=".zip")  
write.mudata(kentvillegreenwood, outfile)  
md <- read.mudata(outfile)
```

```
md <- read.mudata(outfile, retype=TRUE)
plot(subset(md, params=c("meantemp", "maxtemp")))
unlink(outfile)

# read/write to JSON
outfile <- tempfile(fileext=".json")
write.mudata(kentvillegreenwood, outfile)
md <- read.mudata(outfile)
md <- read.mudata(outfile, retype=TRUE)
plot(subset(md, params=c("meantemp", "maxtemp")))
unlink(outfile)
```

Index

*Topic **datasets**

- kentvillegreenwood, 10
- pocmaj, 16
- pocmajsum, 16
- aggregate.qtag.long, 3
- aggregate.qtag.wide
 - (aggregate.qtag.long), 3
- as.qtag, 4
- autoplot.mudata, 5
- autoplot.qtag.long (qualifierplot), 17
- autoplot.qtag.wide (qualifierplot), 17
- biplot.mudata, 6
- biplot.qtag.long (longbiplot), 12
- biplot.qtag.wide (longbiplot), 12
- condense.tags, 6
- data.frame, 17, 24, 25
- expand.tags, 6, 7, 14
- group, 8
- id.vars, 9
- is.summarised, 9
- kentvillegreenwood, 10, 13
- long, 11
- longbiplot, 6, 12
- measure.vars, 13
- mudata, 3, 5, 10, 13, 21, 23
- mudata-package, 2
- parallel.melt, 15
- plot.mudata (autoplot.mudata), 5
- plot.qtag.long (qualifierplot), 17
- plot.qtag.wide (qualifierplot), 17
- pocmaj, 16, 16
- pocmajsum, 16
- print.mudata (summary.mudata), 23
- qtag, 3, 4, 8, 9, 11, 13, 16, 19, 24–26
- qtag (as.qtag), 4
- qualifierdata, 16
- qualifierplot, 5, 17
- rbind.mudata, 18
- rbind.qtag (rbind.qtag.long), 19
- rbind.qtag.long, 19
- read.mudata (write.mudata), 27
- rename.cols, 19
- rename.cols.mudata (rename.datasets), 20
- rename.datasets, 20
- rename.locations (rename.datasets), 20
- rename.params (rename.datasets), 20
- rename.values, 21
- subset.mudata, 5, 22
- summary.mudata, 23
- tag.vars, 23
- tagdata, 24
- validate.mudata, 25
- valuedata, 25
- wide, 26
- write.mudata, 27