

Package: rimu (via r-universe)

September 17, 2024

Title Responses in Multiplex

Version 0.7

Description Tools for manipulating, exploring, and visualising multiple-response data, including scored or ranked responses. Conversions to and from factors, lists, strings, matrices; reordering, lumping, flattening; set operations; tables; frequency and co-occurrence plots.

Imports graphics, stats, UpSetR, ggplot2

Depends R (>= 3.6.0)

Suggests knitr, rmarkdown, vctrs, pillar, dplyr, tidyr, tidyselect

VignetteBuilder knitr

Encoding UTF-8

License GPL-3

Maintainer Thomas Lumley <t.lumley@auckland.ac.nz>

Repository <https://tslumley.r-universe.dev>

RemoteUrl <https://github.com/tslumley/rimu>

RemoteRef HEAD

RemoteSha cb0aeea1d19239c876038a5a43d522ecb671304a

Contents

as.mr	2
as.ms	4
as.vmr	5
birds	6
ethnicity	7
mrequal	7
mr_count	8
mr_flatten	9
mr_inorder	10
mr_lump	11

mr_recode	12
mr_stack	13
mr_union	14
ms_flatten	15
mtable	16
nzbirds	17
plot.mr	17
rstudiosurvey	18
usethnicity	22
vmr_stack	23

Index 24

as.mr	<i>Construct multiple-response objects</i>
-------	--

Description

Constructs mr objects representing multiple-choice questions where more than one choice is allowed.

Usage

```
as.mr(x, ...)
## S3 method for class 'logical'
as.mr(x, name=deparse(substitute(x)), ...)
## S3 method for class 'list'
as.mr(x, sort.levels=TRUE, ..., levels=NULL)
## S3 method for class 'factor'
as.mr(x, sort.levels=FALSE, ...)
## S3 method for class 'data.frame'
as.mr(x, sort.levels=FALSE, ..., na.rm=TRUE)
## S3 method for class 'character'
as.mr(x, sep=", ", sort.levels=TRUE, ..., levels=NULL)
## Default S3 method:
as.mr(x, sort.levels=TRUE, levels=unique(x), ...)
## S3 method for class 'ms'
as.mr(x, ...)
```

Arguments

x	Object to be converted to class mr
...	for compatibility; not used
sort.levels	put the levels of the mr object in increasing sort order. Defaults to TRUE except when there already is a natural order
levels	optional character vector of the permitted levels

name	level name (for a vector) or vector of level names to replace the column names (for a matrix)
na.rm	If TRUE, replace NA in the input by FALSE
sep	Regular expression for splitting the string

Details

The internal representation of `mr` objects is as a logical matrix with the levels as column names.

The method for logical `x` coerces a single vector to a one-column matrix, and then applies the `name` argument as the column name. Given a matrix, the `name` argument is optional and replaces the existing column names; the default is not used.

The method for list `x` takes a list of character vectors that represent the levels present for one observation. The method for strings splits the string at the supplied separator and then uses the list method.

The method for factor `x` produces an `mr` object with the factor levels as levels. Each observation will have only one value.

The `data.frame` object works for logical or numeric columns of a data frame. Zero or negative values are treated as 'not present', positive values as 'present'. Optionally, NA values are coded as 'not present', which is useful when the data frame was created by `reshape` or `dplyr::spread`.

The method for `ms` objects simply drops the score/rank information

Value

Object of class `mr`

Examples

```
nzbirds_list<-list(c("kea","tui"), c("kea","ruru","kaki"), c("ruru"),
c("tui","ruru"), c("tui","kea","ruru"), c("tauhou","kea"))
nzbirds_list
as.mr(nzbirds_list)
as.mr(c("kea, tui","kea, ruru, kaki","ruru","tui, ruru"))

data(nzbirds)
nzbirds
as.mr(nzbirds)

data(ethnicity)
ethnicity
as.logical(ethnicity)
as.mr(as.logical(ethnicity))
```

as.ms

*Construct scored or ranked multiple-response objects***Description**

The internal representation is as a numeric matrix with 0 when a level is not present and the non-zero rank or score when it is present. The data.frame and matrix methods uses the numeric values of x, and by default set NA values to 'not present'. The list method takes a list with a character vector for each observation and uses the position in the list as the rank/score. The character method splits the string at the separators to give a list and uses the list method. The mr method uses a score of 1 whenever the level is present.

Usage

```
as.ms(x, ...)
## S3 method for class 'list'
as.ms(x,...,levels=NULL)
## S3 method for class 'data.frame'
as.ms(x,...,na.rm=TRUE)
## S3 method for class 'matrix'
as.ms(x,...,na.rm=TRUE)
## S3 method for class 'mr'
as.ms(x,...)
## S3 method for class 'character'
as.ms(x,sep=" ", ...,levels=NULL)
```

Arguments

x	object to be converted
...	for compatibility; not used.
levels	Optional character vector giving the permitted levels
na.rm	Convert NA values to 'not present', ie, 0
sep	Regular expression for splitting the character string

Value

Object of class ms

Examples

```
nzbirds_list<-list(c("kea","tui"), c("kea","ruru","kaki"), c("ruru"),
c("tui","ruru"), c("tui","kea","ruru"), c("tauhou","kea"))
nzbirds_list
(msbirds<-as.ms(nzbirds_list))

(bird_mat <- unclass(msbirds))
as.ms(bird_mat)
```

`as.vmr`*Tidyversatile multiple-response objects*

Description

The `vmr` class wraps the `mr` class using the `vctrs` package, for compatibility with tidyverse `tbl_df` objects (tibbles).

Usage

```
as.vmr(x, ...)  
new_vmr(x, levels = unique(do.call(c, x)))
```

Arguments

<code>x</code>	For <code>as.vmr</code> , an <code>mr</code> object or anything coercible to one. For <code>new_vmr</code> , a list of character vectors
<code>...</code>	not used
<code>levels</code>	the permitted levels for the object

Details

These objects need the `vctrs` and `pillar` packages to work, and need the `tibble` package to be useful.

Value

An object of class `vmr`

See Also

[as.mr](#)

The `internals` vignette for internal structure

Examples

```
if (requireNamespace("vctrs", quietly=TRUE)){  
  data(nzbirds)  
  nzbirds  
  tidybirds<-as.vmr(nzbirds, na.rm=TRUE)  
  tidybirds  
}
```

birds

Subset of the Great Backyard Bird survey

Description

Counts of observations for 12 bird species by US county and Canadian province in the Great Backyard Bird survey. These birds were randomly sampled from the much larger number in the full data set. See the vignette for more details.

Usage

```
data("birds")
```

Format

A data frame with 3046 observations on the following 13 variables.

‘Phalaenoptilus nuttallii’ a numeric vector
‘Fregata magnificens’ a numeric vector
‘Melanerpes lewis’ a numeric vector
‘Melospiza georgiana’ a numeric vector
‘Rallus limicola’ a numeric vector
‘Myioborus pictus’ a numeric vector
‘Poecile gambeli’ a numeric vector
‘Aythya collaris’ a numeric vector
‘Xanthocephalus xanthocephalus’ a numeric vector
‘Gracula religiosa’ a numeric vector
‘Icterus parisorum’ a numeric vector
‘Coccyzus erythrophthalmus’ a numeric vector
location a character vector

Examples

```
data(birds)  
birds<-as.ms(birds[,1:12],na.rm=TRUE)  
mtable(as.mr(birds))
```

`ethnicity`*Toy example using New Zealand level 1 ethnicity values*

Description

The statistical standard for collecting ethnicity data requires that respondents can mark all that are applicable. The level 1 values are "Māori", "Pacific Peoples" (ie, Pacific Island ethnicities), "Asian", "European", and "MELAA" (Middle Eastern, Latin American, and African). This is artificial data

Usage

```
data("ethnicity")
```

Format

An object of class `mr`

Examples

```
data(ethnicity)
ethnicity
```

`mrequal`*Check if a level or levels is present*

Description

Returns vector of TRUE or FALSE according to whether `y` is one of the levels present for that row or is the only level present for that row.

Usage

```
x %has% y
x %hasonly% y
x %hasall% ys
x %hasany% ys
```

Arguments

<code>x</code>	<code>mr</code> object
<code>y</code>	character vector specifying a level
<code>ys</code>	character vector specifying one or more levels

Value

Logical vector

Examples

```
data(ethnicity)

ethnicity
ethnicity %has% "Maori"
ethnicity %hasonly% "Maori"

data(nzbirds)
as.mr(nzbirds)
as.mr(nzbirds)
```

 mr_count

Utility functions for multiple-response objects

Description

These perform diverse useful tasks. `mr_count` counts the number of levels present for each individual. `mr_na` sets NA values to something else, `ms_na` sets them to 0 (ie, not present), `mr_drop` and `ms_drop` drop some levels from the object.

Usage

```
mr_count(x, na.rm = TRUE)
mr_drop(x, levels, ...)
ms_drop(x, levels)
mr_na(x, na=TRUE)
ms_na(x)
```

Arguments

<code>x</code>	mr, or ms object
<code>na.rm</code>	Remove NA first?
<code>levels</code>	character vector of levels to remove
<code>na</code>	Value (TRUE/FALSE) to set NA to
<code>...</code>	not used

Value

An integer vector for `mr_count`, an object of class `mr`, or `ms` for the other two functions

Examples

```

data(usetnicity)
race<-as.mr(strsplit(as.character(usetnicity$Q5), ""))
mtable(race)
race<-mr_drop(race,c(" ", "F", "G", "H"))
mtable(race)

## to keep just specified levels use [
mtable(race[,c("A", "D")])

## How many do people identify with
table(mr_count(race))

data(nzbirds)
seenbirds<-as.mr(nzbirds>0)
countbirds<-mr_count(seenbirds)

## How many types of birds were seen
table(countbirds)

data(ethnicity)
ethnicity
mr_na(ethnicity, FALSE)

```

mr_flatten

Flatten a multiple-response object into a factor

Description

Convert a multiple-response object into a factor using a supplied ordering. Each observation is assigned its first level in the ordering. That is, an observation that has `priorities[1]` as one of its levels is assigned that value. An observation that does not have `priorities[1]` as one of its levels, but does have `priorities[2]` is assigned `priorities[2]`.

Usage

```
mr_flatten(x, priorities, sort=FALSE)
```

Arguments

<code>x</code>	mr object
<code>priorities</code>	Character vector of levels.
<code>sort</code>	if TRUE, the levels of the output will be in the order of priorities, if FALSE they will be in the original order

Value

A factor

Examples

```

data(ethnicity)
ethnicity

## NZ 'prioritised ethnicity'
priority<-c("Maori", "Pacific", "Asian", "European/Other")
eth <- mr_na(mr_recode(ethnicity, `European/Other`="European", `European/Other` = "MELAA"), FALSE)

mr_flatten(eth, priority)
mr_flatten(eth, priority, sort=TRUE)

```

`mr_inorder`*Reorder levels of multiple-response objects*

Description

`mr_inorder` and `ms_inorder` use the order in which the levels first appear in the data (which is invariant to locale), `mr_inseq` and `ms_inseq` sort alphabetically (for the current locale). `mr_infreq` sorts by frequency, and `ms_inscore` applies a function to the values in each level – one such function is `mean0`, which takes the mean of non-zero values. Finally, `ms_reorder` and `mr_reorder` use some function of a second variable computed on the observations where each level is present.

Usage

```

mr_inorder(x,...)
ms_inorder(x)
mr_inseq(x,...)
ms_inseq(x)
mr_infreq(x,na.rm=TRUE,...)
ms_infreq(x)
ms_inscore(x, fun=mean0)
mean0(y)
mr_reorder(x, v, fun=median,...)
ms_reorder(x, v, fun=median)

```

Arguments

<code>x</code>	<code>mr</code>
<code>na.rm</code>	Remove NA values before computing frequencies (NA sorts last)
<code>v, fun</code>	Sort levels of <code>x</code> according to the values of <code>fun(v)</code> for observations having each level
<code>y</code>	numeric vector
<code>...</code>	not used

Value

Object of class mr

References

These are based on the reordering functions for factors in the forcats package.

Examples

```
data(ethnicity)
mr_infreq(ethnicity)
mr_inseq(ethnicity)

data(nzbirds)
mtable(nzbirds)
mtable(ms_inorder(nzbirds))
mtable(ms_inseq(nzbirds))
mtable(ms_inscore(nzbirds, mean0))
```

 mr_lump

Collapse common or rare levels

Description

Combine the least common or most common levels of a mr object into an "other" level.

Usage

```
mr_lump(x, n, prop, other_level = "Other",
        ties.method = c("min", "average", "first", "last", "random", "max"),...)
```

Arguments

x	Object of class mr
n	Positive integer to keep the most common n values, negative integer to keep the least common values.
prop	Positive prop preserves values that appear at least prop of the time. Negative prop preserves values that appear at most -prop of the time.
other_level	Label for the lumped levels
ties.method	How to handle ties. Passed to rank
...	not used

Value

An object of class mr

References

Based on `fct_lump` from the `forcats` package.

Examples

```
data(ethnicity)
mtable(ethnicity)
mtable(mr_lump(ethnicity,2))
mtable(mr_lump(ethnicity,-2))

data(rstudiosurvey)
## Other software being used
other_software<- as.mr(rstudiosurvey[[40]])
mtable(other_software)
## The top 20 responses
common<-mr_lump(other_software, n=20)
mtable(common)
## 'None' isn't really another package
mtable(mr_drop(common,"None"))

## Packages with at least 20% use
mtable(mr_lump(other_software, prop=0.2))
```

mr_recode

Relabel levels of multiple-response objects

Description

Relabel some or all of the levels of a multiple-response object. Two levels that are recoded to the same value will be combined.

Usage

```
mr_recode(x, ...)
```

Arguments

x	Object of class <code>mr</code> or <code>ms</code>
...	new names in the form <code>new_name="old name"</code> .

Value

New object of class `mr`, `ms`

Examples

```

data(nzbirds)
nzbirds<-as.mr(nzbirds)
nzbirds
## recode to English names
mr_recode(nzbirds,morepork="ruru",stilt="kaki",waxeye="tauhou")

data(usethnicity)
race<-as.mr(usethnicity$Q5,"")
race<-mr_drop(race,c(" ", "F", "G", "H"))
race <- mr_recode(race, AmIndian="A",Asian="B", Black="C", Pacific="D", White="E")
mtable(race)

```

mr_stack

Pivot a multiple-response object to long form

Description

Creates a data frame where every observation has as many rows as it has levels present, plus an id column to specify which rows go together.

Usage

```

mr_stack(x, ..., na.rm = FALSE)
ms_stack(x, ..., na.rm = FALSE)

```

Arguments

x	multiple response object
...	other multiple response objects
na.rm	drop NA values?

Value

A data frame with columns values and id, plus a column scores if x is a ms object. When more than one object is supplied, the result is an outer join of the two individual results, so it contains a row for every combination of an observed value from each object.

See Also

[vmr_stack](#)

Examples

```

data(ethnicity)
ethnicity
mr_stack(ethnicity)

data(nzbirds)
nzbirds
ms_stack(nzbirds)

## not actually a sensible use
d <- mr_stack(ethnicity, nzbirds)
head(d)
with(d, table(ethnicity, nzbirds))
## equivalent, but more efficient
mtable(mr_na(ethnicity), mr_na(nzbirds))

```

mr_union

Set operations on multiple-response objects

Description

These functions take union, intersection, and difference of two multiple-response objects. An observation has a level in the union if it has that level in either input. It has the level in the intersection if it has the level in both inputs. It has the level in the difference if it has the level in x and not in y

Usage

```

mr_union(x, y, ...)
mr_intersect(x, y, ...)
mr_diff(x, y, ...)

```

Arguments

x, y	Objects of class mr
...	not used

Value

Object of class mr

Examples

```

data(usethnicity)
race<-as.mr(usethnicity$Q5,"")
race<-mr_drop(race,c(" ", "F", "G", "H"))
race <- mr_recode(race, AmIndian="A", Asian="B", Black="C", Pacific="D", White="E")
mtable(race)

```

```

hispanic<-as.mr(usetnicity$Q4==1, "Hispanic")
ethnicity<-mr_union(race, hispanic)
mtable(ethnicity)
ethnicity[101:120]

```

ms_flatten

Flatten a scored multiple-response object into a factor

Description

Convert a multiple-response object into a named numeric vector using a supplied ordering.

Usage

```
ms_flatten(x, priorities, fun, start=0)
```

Arguments

x	mr object
priorities	Character vector of levels.
fun	Function for reducing two values to one.
start	starting value for fun

Details

Each observation is initially assigned the value start. Starting with the lowest-priority level, the current value is combined with the new value as fun(new, current). Using fun=function(x,y) x would return the value for the highest-priority level present; using fun=pmax would return the highest score for any level present; using fun="+" would return the sum of the scores.

Value

A factor

Examples

```

data(ethnicity)
ethnicity

## NZ 'prioritised ethnicity'
eth <- mr_recode(ethnicity, `European/Other`="European", `European/Other` = "MELAA")
mr_flatten(ethnicity, c("Maori","Pacific","Asian","European/Other"))

data(nzbirds)
## hardest to see first

```

```

ms_flatten(nzbirds, c("kaki", "ruru", "kea", "tui", "tauhou"), "+")
ms_flatten(nzbirds, c("kaki", "ruru", "kea", "tui", "tauhou"),
  fun=function(x,y) x)
ms_flatten(nzbirds, c("kaki", "ruru", "kea", "tui", "tauhou"), pmin, start=Inf)

```

mtable

Tables involving multiple-response objects

Description

Creates one-way and two-way tables using every level of a multiple response object. Use `table(as.character(x))` to tabulate combinations of levels

Usage

```
mtable(x, y, na.rm = TRUE)
```

Arguments

x	mr object or a factor
y	mr object or a factor
na.rm	remove missing values?

Value

A 1-d or 2-d array with names giving the levels

Examples

```

data(ethnicity)
mtable(ethnicity)
table(as.character(ethnicity))

data(nzbirds)
nzbirds<-as.mr(nzbirds)

## co-occurrence table
mtable(nzbirds, nzbirds)

## table by a factor
v<-rep(c("A", "B"),3)
mtable(nzbirds,v)

data(nzbirds)
mtable(nzbirds>0)

```

`nzbirds`*Toy example using New Zealand birds*

Description

A small artificial dataset that could be produced by asking people to name New Zealand birds. Each observation has scores from 1 (first bird named) to at most 4 (fourth bird named).

Usage

```
data("nzbirds")
```

Format

A `ms` object with 6 observations on the following 5 variables.

`kea` a numeric vector

`ruru` a numeric vector

`tui` a numeric vector

`tauhou` a numeric vector

`kaki` a numeric vector

Examples

```
data(nzbirds)
nzbirds
as.mr(nzbirds)
```

`plot.mr`*Plot multiple-response objects*

Description

The `plot` method for `mr` objects is an UpSet plot, showing co-occurrences of the various categories. The `image` method is a heatmap of the variable plotted against itself with [mtable](#).

Usage

```
## S3 method for class 'mr'
plot(x, ...)
## S3 method for class 'mr'
image(x, type = c("overlap", "conditional", "association",
"raw"), ...)
## S3 method for class 'mr'
barplot(height,...)
```

Arguments

x	mr object
type	"overlap" is a plot of counts, "conditional" is of column proportions, "association" has rows and columns scaled to give unit diagonals. "raw" just plots as <code>.logical(x)</code> .
height	mr object; the barplot will be of <code>mtable(height)</code>
...	Passed to <code>barplot</code> but otherwise not used

Value

Used for its side effect

See Also

[as.mr](#), [mtable](#)

Examples

```
data(rstudiosurvey)
other_software<- as.mr(rstudiosurvey[[40]])
## only those with at least 20 responses
common<-mr_lump(other_software, n=20)
common<-mr_drop(common, "None")

## UpSet plot
plot(common)

## images

image(common, type="conditional")
image(common, type="association")
```

rstudiosurvey

Subset of RStudio 2019 Community Survey

Description

The ‘rstudiosurvey’ data set contains 1838 rows of responses from the 2019 RStudio Community Survey, where columns are the 51 questions and a column for the timestamp. The variable names are the full questions. Multiple responses are separated by a comma and space. Non-ASCII characters have been converted with the "ASCII//TRANSLIT" option of [iconv](#).

Usage

```
data("rstudiosurvey")
```

Format

A data frame with 1838 observations on the following 52 variables.

Timestamp a character vector

‘How would you rate your level of experience using R?’ a character vector

‘Compared with other technical topics you’ve learned in school and on the job, on a scale of 1 to 5, how difficult was it to learn R?’ a numeric vector

‘From what you know about R, how long do you expect that it will take for you to learn enough to use R productively?’ a character vector

‘How do you think you would go about the process of learning R?’ a character vector

‘Which statement most closely reflects the primary reason why you are interested in learning R?’ a character vector

‘If you were to learn R, what would do you think you would use it for? (check all that apply)’ a character vector

‘Which analytical tools do you use today for the functions that you might learn R for? (please check all that apply)’ a character vector

‘What do you think is the biggest obstacle you must overcome in trying to learn R? The choices below are only a few.’ a character vector

‘What year did you first start learning R?’ a numeric vector

‘How did you learn R? If you used multiple methods, please select the one you used the most.’ a character vector

‘Compared with other technical topics you’ve learned in school and on the job, on a scale of 1 to 5, how difficult was it to learn R?’ a numeric vector

‘Roughly how long did it take you to achieve proficiency in R?’ a character vector

‘Which statement most closely reflects the primary reason why you learned R?’ a character vector

‘What do you think was the biggest obstacle you had to overcome in learning R? The choices below are only a few.’ a character vector

‘How often do you use R today, either for professional or personal projects?’ a character vector

‘What applications do you use R for most? (check all that apply)’ a character vector

‘Please rate how much you enjoy using R on a scale of 1 to 5, where 1 is you don’t enjoy it at all, and 5 is that you love it.’ a numeric vector

‘How likely are you to recommend R to a colleague, friend, or family member?’ a numeric vector

‘Which tools do you use with your R applications? (please check all that apply)’ a character vector

‘Did you use tidyverse packages such as ggplot2 or dplyr to learn R?’ a character vector

‘Do you use tidyverse packages when you use R now?’ a character vector

‘What do you like best about using R?’ a character vector

‘What do you like least about using R?’ a character vector

- ‘When you have problems in R, where do you go for help?’ a character vector
- ‘How do you discover new packages or packages that are unfamiliar to you?’ a character vector
- ‘How do you share the results that you create in R? Check all that apply.’ a character vector
- ‘Looking ahead, how do you expect your use of R to change in 2020?’ a character vector
- ‘To help us ensure that you are not a robot, please enter the number of characters in the word “analysis” in t
a character vector
- ‘Do you currently use R Markdown? Choose the statement that most closely matches your use.’
a character vector
- ‘What applications do you use R Markdown for? Check all that apply.’ a character vector
- ‘Looking forward, how do you expect your use of R Markdown to change in 2020?’ a character vector
- ‘How often do you currently use Shiny? Choose the statement that most closely matches your use.’
a character vector
- ‘Looking forward, how do you expect your use of Shiny to change in 2020?’ a character vector
- ‘Do you currently use Python? Choose the statement that most closely matches your use.’
a character vector
- ‘What applications do you use Python for most? (check all that apply)’ a character vector
- ‘Please rate how much you enjoy using Python on a scale of 1 to 5, where 1 is you don’t enjoy it at all, and 5 is
a numeric vector
- ‘How likely are you to recommend Python to a colleague, friend, or family member?’ a numeric vector
- ‘Looking forward, how do you expect your use of Python to change in 2020?’ a character vector
- ‘What computer tools and/or languages have you used besides R?’ a character vector
- ‘What was the FIRST computer language or tool that you learned?’ a character vector
- ‘What year were you born?’ a numeric vector
- ‘What gender do you identify with?’ a character vector
- ‘I identify my ethnicity as (select all that apply):’ a character vector
- ‘What is the highest degree or level of school you have completed? If currently enrolled, please use the high
a character vector
- ‘In what country do you currently reside?’ a character vector
- ‘What industry do you work or participate in?’ a character vector
- ‘What is your job title, if any?’ a character vector
- ‘Which category best describes the work you do?’ a character vector
- ‘How many people in your organization or work group do you feel that you can ask for help or support when work
a numeric vector
- ‘Which of the following events have you attended, if any? Check all that apply.’ a character vector
- ‘How did you hear about this survey?’ a character vector

Source

<https://github.com/rstudio/r-community-survey/tree/master/2019>

Examples

```
data(rstudiosurvey)
names(rstudiosurvey)[40]
## Other software being used
other_software<- as.mr(rstudiosurvey[[40]])
mtable(other_software)
## top 20 responses
common<-mr_lump(other_software, n=20)
mtable(common)
## 'None' isn't really another package
common<-mr_drop(common, "None")
mtable(common)

## UpSet plot
plot(common)

## Excel users filled in the survey later
timestamp<-as.Date(rstudiosurvey[[1]],format="%m/%d/%y")
boxplot(timestamp~I(common %has% "Excel"))

## names in order of popularity
t<-mtable(common)
popular<-colnames(t)[order(t,decreasing=TRUE)]
## most popular package for each user
cuml_users <- mr_flatten(common, popular, sort=TRUE)
class(cuml_users)
table(cuml_users)

## two-way tables
## people who also use Stata or Julia are less happy with R than those who don't
names(rstudiosurvey)[18]
happy<-factor(rstudiosurvey[[18]])
mtable(happy, common)
round(prop.table(mtable(happy,common),2),2)

## mr objects can be dataframe columns, or expanded to individual levels
df<-data.frame(timestamp, happy, common)
dim(df)
head(df)
df_raw<-data.frame(timestamp, happy, as.matrix(common))
dim(df_raw)
head(df_raw)
```

`usethnicity`*Data from Youth Risk Behaviour Survey*

Description

This data set contains variables on race and ethnic identification from the 2017 Youth Risk Behaviour Survey, together with two variables on smoking behaviour. The YRBS is a multistage cluster-sampled survey, so valid inference about associations requires using survey design information. This subset is useful only for demonstration purposes.

Usage

```
data("usethnicity")
```

Format

A data frame with 14765 observations on the following 4 variables.

Q4 1 is "Hispanic or Latino"

Q5 Character string with zero or more of: A. American Indian or Alaska Native, B. Asian, C. Black or African American, D. Native Hawaiian or Other Pacific Islander, E. White

QN30 1 is "smoked cigarettes on one or more of the past 30 days"

QN31 1 is "smoked more than 10 cigarettes per day on the days they smoked during the past 30 days", those who did not smoke at all are NA

Source

<https://www.cdc.gov/healthyouth/data/yrbs/data.htm>

Examples

```
data(usethnicity)
race<-as.mr(strsplit(as.character(usethnicity$Q5), ""))
race<-mr_drop(race, " ")
mtable(race)
hispanic<-as.mr(usethnicity$Q4==1, "Hispanic")
ethnicity<-mr_union(race, hispanic)
ethnicity[101:120]
```

`vnr_stack`*Stack a multiple response column in a tibble*

Description

Similar to [mr_stack](#), but it works on whole data frames rather than individual vectors, expanding a multiple-response variable to a factor with a record for each observed response. The algorithm is to convert the object to a logical matrix, `pivot_longer`, then `filter` on the logical value.

Usage

```
vnr_stack(data, col, names_to = "level")
```

Arguments

<code>data</code>	a tibble
<code>col</code>	the name of a multiple-response column in <code>data</code>
<code>names_to</code>	the desired output name of the variable with the stacked levels

Value

An expanded data frame

See Also

[mr_stack](#)

Examples

```
data(ethnicity)

if(require("tidyr",quietly=TRUE)){

  t<-tibble(a=LETTERS[1:6], e=as.vnr(ethnicity,na.rm=TRUE))
  t
  t |> vnr_stack(e,names_to="ethnicity")

}
```

Index

- * **classes**
 - as.mr, 2
- * **datasets**
 - birds, 6
 - ethnicity, 7
 - nzbirds, 17
 - rstudiosurvey, 18
 - usethnicity, 22
- * **manip**
 - as.mr, 2
 - mr_count, 8
 - mr_flatten, 9
 - mr_inorder, 10
 - mr_lump, 11
 - mr_recode, 12
 - mr_stack, 13
 - mr_union, 14
 - mrequal, 7
 - vmr_stack, 23
- %has%(mrequal), 7
- %hasall%(mrequal), 7
- %hasany%(mrequal), 7
- %hasonly%(mrequal), 7

- as.mr, 2, 5, 18
- as.ms, 4
- as.vmr, 5

- barplot, 18
- barplot.mr (plot.mr), 17
- birds, 6

- ethnicity, 7

- iconv, 18
- image.mr (plot.mr), 17

- mean0 (mr_inorder), 10
- mr_count, 8
- mr_diff (mr_union), 14
- mr_drop (mr_count), 8
- mr_flatten, 9
- mr_infreq (mr_inorder), 10
- mr_inorder, 10
- mr_inseq (mr_inorder), 10
- mr_intersect (mr_union), 14
- mr_lump, 11
- mr_na (mr_count), 8
- mr_recode, 12
- mr_reorder (mr_inorder), 10
- mr_stack, 13, 23
- mr_union, 14
- mrequal, 7
- ms_drop (mr_count), 8
- ms_flatten, 15
- ms_infreq (mr_inorder), 10
- ms_inorder (mr_inorder), 10
- ms_inscore (mr_inorder), 10
- ms_inseq (mr_inorder), 10
- ms_na (mr_count), 8
- ms_recode (mr_recode), 12
- ms_reorder (mr_inorder), 10
- ms_stack (mr_stack), 13
- mtable, 16, 17, 18

- new_vmr (as.vmr), 5
- nzbirds, 17

- plot.mr, 17

- rank, 11
- rstudiosurvey, 18

- usethnicity, 22

- vmr_stack, 13, 23