Package ‘surveydata’

July 19, 2017

License GPL-2 | GPL-3
Title Tools to Work with Survey Data
LazyData true
LazyLoad true
Copyright Andrie de Vries
Description Data obtained from surveys contains information not only about the survey responses, but also the survey metadata, e.g. the original survey questions and the answer options. The surveydata package makes it easy to keep track of this metadata, and to easily extract columns with specific questions.
Version 0.2.0
URL https://github.com/andrie/surveydata
Date 2017-07-19
ByteCompile yes
Depends R (>= 2.15.0)
Imports stringr (>= 0.5), plyr, dplyr, magrittr, purrr
Suggests testthat, knitr, rmarkdown, withr, covr
RoxygenNote 6.0.1
VignetteBuilder knitr
NeedsCompilation no
Author Andrie de Vries [aut, cre, cph]
Maintainer Andrie de Vries <apdevries@gmail.com>
Repository CRAN
Date/Publication 2017-07-19 17:07:52 UTC

R topics documented:
surveydata-package .......................................................... 2
as.data.frame.surveydata .................................................. 5
### surveydata-package

Tools, classes and methods to manipulate survey data.

#### Description

Tools, classes and methods to manipulate survey data.

---

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>as.surveydata</td>
<td>6</td>
</tr>
<tr>
<td>cbind.surveydata</td>
<td>8</td>
</tr>
<tr>
<td>dropout</td>
<td>8</td>
</tr>
<tr>
<td>encToInt</td>
<td>9</td>
</tr>
<tr>
<td>Extract</td>
<td>9</td>
</tr>
<tr>
<td>fixCommonEncodingProblems</td>
<td>11</td>
</tr>
<tr>
<td>fixLevels01</td>
<td>11</td>
</tr>
<tr>
<td>fixLevels01R</td>
<td>12</td>
</tr>
<tr>
<td>fixLevels01SPSS</td>
<td>12</td>
</tr>
<tr>
<td>hasDK</td>
<td>13</td>
</tr>
<tr>
<td>intToEnc</td>
<td>13</td>
</tr>
<tr>
<td>is.surveydata</td>
<td>14</td>
</tr>
<tr>
<td>lapplyNames</td>
<td>14</td>
</tr>
<tr>
<td>leveltestR</td>
<td>15</td>
</tr>
<tr>
<td>leveltestSPSS</td>
<td>15</td>
</tr>
<tr>
<td>membersurvey</td>
<td>16</td>
</tr>
<tr>
<td>merge</td>
<td>16</td>
</tr>
<tr>
<td>names&lt;-</td>
<td>17</td>
</tr>
<tr>
<td>pattern</td>
<td>17</td>
</tr>
<tr>
<td>qOrder</td>
<td>18</td>
</tr>
<tr>
<td>qText</td>
<td>19</td>
</tr>
<tr>
<td>qTextCommon</td>
<td>20</td>
</tr>
<tr>
<td>qTextUnique</td>
<td>21</td>
</tr>
<tr>
<td>questions</td>
<td>22</td>
</tr>
<tr>
<td>removeAllDK</td>
<td>23</td>
</tr>
<tr>
<td>removeDK</td>
<td>24</td>
</tr>
<tr>
<td>rm.attrs</td>
<td>24</td>
</tr>
<tr>
<td>rm.pattern</td>
<td>25</td>
</tr>
<tr>
<td>splitCommonUnique</td>
<td>25</td>
</tr>
<tr>
<td>strCommonUnique</td>
<td>26</td>
</tr>
<tr>
<td>varlabels</td>
<td>26</td>
</tr>
<tr>
<td>which.q</td>
<td>27</td>
</tr>
</tbody>
</table>

---

**Index**

29
surveydata-package

Details
Surveydata objects have been designed to function with SPSS export data, i.e. the result of an SPSS import, foreign::read.spss(). This type of data is contained in a data.frame, with information about the questionnaire text in the variable labels attribute. Surveydata objects keep track of the variable labels, by offering methods for renaming, subsetting, etc.

Coercion functions:
- `as.surveydata()`
- `is.surveydata()`
- `as.data.frame.surveydata()`

To access and modify attributes:
- `pattern()`
- `varlabels()`

To subset or merge surveydata objects:
- `surveydata::merge()`
- `surveydata::Extract()`
- `cbind.surveydata()`

To extract question text from varlabels:
- `qText()`
- `qTextCommon()`
- `qTextUnique()`

To fix common encoding problems:
- `encToInt()`
- `intToEnc()`
- `fixCommonEncodingProblems()`

To clean data:
- `removeDK()` to remove "Don’t know" responses
- `removeAllDK()` to remove "Don’t know" responses from all questions
- `fixLevels01()` to fix level formatting of all question with Yes/No type answers

Miscellaneous tools:
- `dropout()` to determine questions where respondents drop out

Author(s)
Andrie de Vries <apdevries@gmail.com>
Examples

# Create surveydata object

sdat <- data.frame(
  id = 1:4,
  Q1 = c("Yes", "No", "Yes", "Yes"),
  Q4_1 = c(1, 2, 1, 2),
  Q4_2 = c(3, 4, 4, 3),
  Q4_3 = c(5, 5, 6, 6),
  Q10 = factor(c("Male", "Female", "Female", "Male")),
  crossbreak = c("A", "A", "B", "B"),
  weight = c(0.9, 1.1, 0.8, 1.2)
)

varlabels(sdat) <- c(
  "RespID",
  "Question 1",
  "Question 4: red", "Question 4: green", "Question 4: blue",
  "Question 10",
  "crossbreak",
  "weight"
)

sv <- as.surveydata(sdat, renameVarlabels=TRUE)

# Extract specific questions

sv[, "Q1"]
sv[, "Q4"]

# Query attributes

varlabels(sv)
pattern(sv)

# Find unique questions

questions(sv)
which.q(sv, "Q1")
which.q(sv, "Q4")

# Find question text

qText(sv, "Q1")
qText(sv, "Q4")

qTextCommon(sv, "Q4")
qTextUnique(sv, "Q4")

# Basic operations on a surveydata object, illustrated with the example dataset membersurvey

class(membersurvey)

questions(membersurvey)
as.data.frame.surveydata

Coerces surveydata object to data.frame.

Description

Coerces surveydata object to data.frame.

Usage

```r
## S3 method for class 'surveydata'
as.data.frame(x, ..., rm.pattern = FALSE)
```

Arguments

- `x` Surveydata object to coerce to class data.frame
- `...` ignored
- `rm.pattern` If TRUE removes `pattern()` attributes from x

See Also

- `surveydata-package`
as.surveydata

Coercion from and to surveydata.

Description

Methods for creating surveydata objects, testing for class, and coercion from other objects.

Usage

```r
as.surveydata(x, sep = "_", exclude = "other", ptn = pattern(x),
  defaultPtnt = list(sep = sep, exclude = exclude), renameVarlabels = FALSE)
```

Arguments

- `x`: Object to coerce to surveydata
- `sep`: Separator between question and subquestion names
- `exclude`: Excludes from pattern search
- `ptn`: A list with two elements, `sep` and `exclude`. See `pattern()` and `which.q()` for more detail.
- `defaultPtnt`: The default for `ptn`, if it doesn’t exist in the object that is being coerced.
- `renameVarlabels`: If TRUE, turns variable.labels attribute into a named vector, using `names(x)` as names.

See Also

- `surveydata-package`, `is.surveydata()`

Examples

```r
# Create surveydata object

sdat <- data.frame(
  id = 1:4,
  Q1  = c("Yes", "No", "Yes", "Yes"),
  Q4_1 = c(1, 2, 1, 2),
  Q4_2 = c(3, 4, 4, 3),
  Q4_3 = c(5, 5, 6, 6),
  Q10 = factor(c("Male", "Female", "Female", "Male")),
  crossbreak = c("A", "A", "B", "B"),
  weight  = c(0.9, 1.1, 0.8, 1.2)
)

varlabels(sdat) <- c(
  "RespID",
  "Question 1",
  "Question 4: red", "Question 4: green", "Question 4: blue",
  ...)
"Question 10",
"crossbreak",
"weight"
)
sv <- as.surveydata(sdat, renameVarlabels=TRUE)

# Extract specific questions
sv[, "Q1"]
sv[, "Q4"]

# Query attributes
varlabels(sv)
pattern(sv)

# Find unique questions
questions(sv)
which.q(sv, "Q1")
which.q(sv, "Q4")

# Find question text
qText(sv, "Q1")
qText(sv, "Q4")
qTextCommon(sv, "Q4")
qTextUnique(sv, "Q4")

# Basic operations on a surveydata object, illustrated with the example dataset membersurvey
class(membersurvey)

questions(membersurvey)

which.q(membersurvey, "Q1")
which.q(membersurvey, "Q3")
which.q(membersurvey, c("Q1", "Q3"))

qText(membersurvey, "Q3")
qTextUnique(membersurvey, "Q3")
qTextCommon(membersurvey, "Q3")

# Extracting columns from a surveydata object
head(membersurvey[, "Q1"])
head(membersurvey[, "Q1"])
head(membersurvey[, "Q3"])
head(membersurvey[, c("Q1", "Q3")])

# Note that the result is always a surveydata object, even if only one column is extracted
head(membersurvey[, "id"])

dropout

\[ \text{str(membersurvey[, "id"])} \]

cbind.surveydata \textit{Combines surveydata object by columns.}

\section*{Description}
Combines surveydata object by columns.

\section*{Usage}
\#
\textbackslash S3 method for class 'surveydata'
cbind(..., deparse.level = 1)

\section*{Arguments}
\begin{itemize}
\item \textit{...} surveydata objects
\item \textit{deparse.level} ignored
\end{itemize}

dropout \textit{Calculates at which questions respondents drop out.}

\section*{Description}
The number of respondents for each question is calculated as the length of the vector, after omitting NA values.

\section*{Usage}
dropout(x, summary = TRUE)

\section*{Arguments}
\begin{itemize}
\item \textit{x} surveydata object, list or data.frame
\item \textit{summary} If TRUE, returns a shortened vector that contains only the points where respondents drop out. Otherwise, returns the number of respondents for each question.
\end{itemize}

\section*{Value}
Named numeric vector of respondent counts

\section*{Examples}
dropout(membersurvey[\-(127:128)])
**encToInt**  

*Converts a character vector to an integer vector*

**Description**

Conversion of character vector to integer vector. The encoding of the character vector can be specified but defaults to the current locale.

**Usage**

```
encToInt(x, encoding = localeToCharset())
```

**Arguments**

- `x`  
  Character vector

- `encoding`  
  A character string describing the encoding of `x`. Defaults to the current locale. See also `iconvlist()`

**Value**

An integer vector

**See Also**

`iconv()`

Other Functions to clean data: `fixCommonEncodingProblems, fixLevels01R, fixLevels01SPSS, fixLevels01.hasOK, intToEnc, leveltestR, leveltestSPSS, removeAllDK, removeDK`

**Examples**

```
encToInt("\xfa")
```

---

**Extract**

*Extract or replace subsets of surveydata, ensuring that the varlabels stay in synch.*

**Description**

The surveydata package makes it easy to extract specific questions from a surveydata object. Because survey data typically has question names like Q1_a, Q1_b, Q1_c the extract method for a surveydata object makes it easy to extract all columns by simply specifying "Q1" as the argument to the column index.
Usage

```r
## S3 method for class 'surveydata'
x[i, j, drop = FALSE]

## S3 replacement method for class 'surveydata'
x[i, j] <- value

## S3 replacement method for class 'surveydata'
x$name <- value
```

Arguments

- `x` surveydata object
- `i` row index
- `j` column index
- `drop` logical. Passed to `.data.frame`. Note that the default is `FALSE`.
- `value` New value
- `name` Names of columns
- `...` Other arguments passed to `.data.frame`

Details

Extraction is similar to data frames, with three important exceptions:

- The column argument `j` is evaluated using `which.q()` and will return all columns where the column names match the `pattern()`.
- The drop argument is `FALSE`. Thus the result will always be a surveydata object, even if only a single column is returned.
- All extraction methods retain the `pattern` and `varlabels` arguments.

See Also

`surveydata-package`, `varlabels`

Examples

```r
names(membersurvey)
head(membersurvey["Q1"])
head(membersurvey[c("Q1", "Q2")])
head(membersurvey[membersurvey$Q2=="2009", c("Q1", "Q2")])

# The pattern is used to extract columns

pattern(membersurvey)

grep("Q20", names(membersurvey), value=TRUE)
head(membersurvey["Q20"])
```
**Fix common encoding problems when working with web imported data.**

Description

This function tries to resolve typical encoding problems when importing web data on Windows. Typical problems occur with pound and hyphen (-), especially when these originated in MS-Word.

Usage

```r
fixCommonEncodingProblems(x, encoding = localeToCharset())
```

Arguments

- `x`: A character vector
- `encoding`: A character string describing the encoding of `x`. Defaults to the current locale. See also `iconvlist()`

See Also

Other Functions to clean data: `enctoint`, `fixLevels01R`, `fixLevels01SPSS`, `fixLevels01`, `hasDK`, `intToEnc`, `leveltestR`, `leveltestSPSS`, `removeAllDK`, `removeDK`
fixLevels01R  
*Fix level formatting of all question with Yes/No type answers.*

**Description**

Fix level formatting of all question with Yes/No type answers.

**Usage**

`fixLevels01R(dat)`

**Arguments**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>surveydata object</td>
</tr>
</tbody>
</table>

**See Also**

Other Functions to clean data: `encToInt, fixCommonEncodingProblems, fixLevels01SPSS, fixLevels01, hasDK, intToEnc, leveltestR, leveltestSPSS, removeAllDK, removeDK`  

---

fixLevels01SPSS  
*Fix level formatting of all question with Yes/No type answers.*

**Description**

Fix level formatting of all question with Yes/No type answers.

**Usage**

`fixLevels01SPSS(dat)`

**Arguments**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>Surveyor data object</td>
</tr>
</tbody>
</table>

**See Also**

Other Functions to clean data: `encToInt, fixCommonEncodingProblems, fixLevels01R, fixLevels01, hasDK, intToEnc, leveltestR, leveltestSPSS, removeAllDK, removeDK`
hasDK  

Tests whether levels contain "Don't know".

Description

Returns TRUE if x contains any instances of dk

Usage

hasDK(x, dk = "Don't Know")

Arguments

x  
Character vector or factor

dk  
Character vector, containing search terms, e.g. c("Don't know", "Don't Know")

Value

TRUE or FALSE

See Also

Other Functions to clean data: encToInt, fixCommonEncodingProblems, fixLevels01R, fixLevels01SPSS, fixLevels01, intToEnc, leveltestR, leveltestSPSS, removeAllDK, removeDK

intToEnc  

Converts an integer vector to a character vector.

Description

Conversion of integer vector to character vector. The encoding of the character vector can be specified but defaults to the current locale.

Usage

intToEnc(x, encoding = localeToCharset())

Arguments

x  
Integer vector

encoding  
A character string describing the encoding of x. Defaults to the current locale. See also iconvlist()

Value

A character vector
See Also

iconv()

Other Functions to clean data: encToInt, fixCommonEncodingProblems, fixLevels01R, fixLevels01SPSS, fixLevels01, hasDK, leveltestR, leveltestSPSS, removeAllDK, removeDK

Examples

intToEnc(8212)

is.surveydata Tests whether an object is of class surveydata.

Description

Tests whether an object is of class surveydata.

Usage

is.surveydata(x)

Arguments

x Object to check for being of class surveydata

See Also

surveydata-package

lapplyNames Applies function only to named elements of a list.

Description

This is useful to clean only some columns in a list (or data.frame or surveydata object). This is a simple wrapper around lapply() where only the named elements are changed.

Usage

lapplyNames(x, names, FUN, ...)

Arguments

x list
names character vector identifying which elements of the list to apply FUN
FUN function to apply.
... additional arguments passed to FUN
leveltestR

See Also
Other Tools: qOrder

| leveltestR | Fix level formatting of all question with Yes/No type answers. |

Description
Fix level formatting of all question with Yes/No type answers.

Usage
leveltestR(x)

Arguments
x surveydata object

See Also
Other Functions to clean data: encToInt, fixCommonEncodingProblems, fixLevels01R, fixLevels01SPSS, fixLevels01, hasDK, intToEnc, leveltestSPSS, removeAllDK, removeDK

| leveltestSPSS | Fix level formatting of all question with Yes/No type answers. |

Description
Fix level formatting of all question with Yes/No type answers.

Usage
leveltestSPSS(x)

Arguments
x Surveyor data object

See Also
Other Functions to clean data: encToInt, fixCommonEncodingProblems, fixLevels01R, fixLevels01SPSS, fixLevels01, hasDK, intToEnc, leveltestR, removeAllDK, removeDK
membersurvey | Data frame with survey data of member satisfaction survey.

Description
Data frame with survey data of member satisfaction survey.

Usage
membersurvey

Format
data frame

merge | Merge surveydata objects.

Description
The base R merge will merge data but not all of the attributes. This function also merges the variable.labels attribute.

Usage
## S3 method for class 'surveydata'
merge(x, y, ...)

Arguments
x       surveydata object
y       surveydata object
...     Other parameters passed to merge()
names<-  

Updates names and variable.labels attribute of surveydata.

Description

Updates names and variable.labels attribute of surveydata.

Usage

\method{names}{surveydata}(x) <- value

Arguments

x  
surveydata object

value

New names

See Also

surveydata-package(), is.surveydata()

pattern  

Returns and updates pattern attribute.

Description

The pattern attribute contains information about the separator character used to name subquestions in the data. Survey software typically makes use of underscores to distinguish subquestions in a grid of questions, e.g. Q4_1, Q4_2, Q4_3, Q4_other. The function pattern() returns the pattern attribute, and pattern<- updates the attribute.

Usage

pattern(x)

pattern(x) <- value

Arguments

x  
surveydata object

value

New value

See Also

as.surveydata(), which.q()

Other Attribute functions: varlabels
Examples

# Extract the pattern from membersurvey

oldptn <- pattern(membersurvey)
oldptn

# The pattern is used to extract columns

names(membersurvey)
grep("Q20", names(membersurvey), value=TRUE)

head(membersurvey["Q20"])
head(membersurvey["Q20_other"])

# Define a new pattern

pattern(membersurvey) <- list(sep=" ", exclude="")
head(membersurvey["Q20"])  

# Reset original pattern

pattern(membersurvey) <- oldptn
rm(oldptn)

qOrder(x) Changes vector to ordered factor, adding NA levels if applicable.

Description

Changes vector to ordered factor, adding NA levels if applicable.

Usage

qOrder(x)

Arguments

x character vector

See Also

Other Tools: `lapplyNames`
qText

Returns question text.

Description

Given a question id, e.g. "Q4", returns question text for this question. Note that this returns. The functions \texttt{qTextUnique()} and \texttt{qTextCommon()} returns the unique and common components of the question text.

Usage

\texttt{qText(x, Q)}

Arguments

\begin{itemize}
\item \texttt{x} \hspace{1cm} A surveydata object
\item \texttt{Q} \hspace{1cm} The question id, e.g. "Q4"
\end{itemize}

Value

character vector

See Also

Other Question functions: \texttt{qTextCommon}, \texttt{qTextUnique}, \texttt{questions}, \texttt{splitCommonUnique}, \texttt{which.q}

Examples

\begin{verbatim}
# Basic operations on a surveydata object, illustrated with the example dataset membersurvey
class(membersurvey)
questions(membersurvey)

which.q(membersurvey, "Q1")
which.q(membersurvey, "Q3")
which.q(membersurvey, c("Q1", "Q3"))

qText(membersurvey, "Q3")
qTextUnique(membersurvey, "Q3")
qTextCommon(membersurvey, "Q3")

# Extracting columns from a surveydata object
head(membersurvey[, "Q1"])
head(membersurvey["Q1"])
head(membersurvey[, "Q3"])
head(membersurvey[, c("Q1", "Q3")])
\end{verbatim}
# Note that the result is always a surveydata object, even if only one column is extracted

head(membersurvey[, "id"])
str(membersurvey[, "id"])

qTextCommon

Returns common element of question text.

Description

Given a question id, e.g. "Q4", finds all subquestions, e.g. Q4_1, Q4_2, etc, and returns the question text that is common to each.

Usage

qTextCommon(x, Q)

Arguments

x
A surveydata object
Q
The question id, e.g. "Q4"

Value

character vector

See Also

Other Question functions: qTextUnique, qText, questions, splitCommonUnique, which.q

Examples

# Basic operations on a surveydata object, illustrated with the example dataset membersurvey

class(membersurvey)

questions(membersurvey)

which.q(membersurvey, "Q1")
which.q(membersurvey, "Q3")
which.q(membersurvey, c("Q1", "Q3"))

qText(membersurvey, "Q3")
qTextUnique(membersurvey, "Q3")
qTextCommon(membersurvey, "Q3")

# Extracting columns from a surveydata object
qTextUnique

```
head(membersurvey[, "Q1"])
head(membersurvey["Q1"])
head(membersurvey[, "Q3"])
head(membersurvey[, c("Q1", "Q3")])

# Note that the result is always a surveydata object, even if only one column is extracted
head(membersurvey[, "id"])
str(membersurvey[, "id"])
```

---

**qTextUnique**

*Returns unique elements of question text.*

**Description**

Given a question id, e.g. "Q4", finds all subquestions, e.g. Q4_1, Q4_2, etc, and returns the question text that is unique to each

**Usage**

`qTextUnique(x, Q)`

**Arguments**

- **x** A surveydata object
- **Q** The question id, e.g. "Q4"

**Value**

character vector

**See Also**

Other Question functions: `qTextCommon, qText, questions, splitCommonUnique, which.q`

**Examples**

```r
# Basic operations on a surveydata object, illustrated with the example dataset membersurvey
class(membersurvey)
questions(membersurvey)

which.q(membersurvey, "Q1")
which.q(membersurvey, "Q3")
which.q(membersurvey, c("Q1", "Q3"))
qText(membersurvey, "Q3")
```
# Extracting columns from a surveydata object

head(membersurvey[, "Q1"])
head(membersurvey["Q1"])
head(membersurvey[, "Q3"])
head(membersurvey[, c("Q1", "Q3")])

# Note that the result is always a surveydata object, even if only one column is extracted

head(membersurvey[, "id"])
str(membersurvey[, "id"])

---

**questions**

*Returns a list of all the unique questions in the surveydata object.*

**Description**

In many survey systems, subquestions take the form Q1_a, Q1_b, with the main question and sub-question separated by an underscore. This function conveniently returns all of the main questions in a `surveydata()` object. It does this by using the `pattern()` attribute of the surveydata object.

**Usage**

`questions(x, ptn = pattern(x))`

**Arguments**

- **x**: Object to coerce to surveydata.
- **ptn**: A list with two elements, sep and exclude. See `pattern()` and `which.q()` for more detail.

**Value**

numeric vector

**See Also**

- `which.q`

Other Question functions: `qTextCommon`, `qTextUnique`, `qText`, `splitCommonUnique`, `which.q`
Examples

# Basic operations on a surveydata object, illustrated with the example dataset membersurvey

class(membersurvey)

questions(membersurvey)

which.q(membersurvey, "Q1")
which.q(membersurvey, "Q3")
which.q(membersurvey, c("Q1", "Q3"))

qText(membersurvey, "Q3")
qTextUnique(membersurvey, "Q3")
qTextCommon(membersurvey, "Q3")

# Extracting columns from a surveydata object

head(membersurvey[, "Q1"])
head(membersurvey[, "Q1"])
head(membersurvey[, "Q3"])
head(membersurvey[, c("Q1", "Q3")])

# Note that the result is always a surveydata object, even if only one column is extracted

head(membersurvey[, "id"])
str(membersurvey[, "id"])

removeAllDK

Removes "Do not know" and other similar words from factor levels in data frame.

Description

Removes "Do not know" and other similar words from factor levels in data frame

Usage

removeAllDK(x, dk = NULL)

Arguments

x List or data frame

dk Character vector, containing search terms, e.g. c("Do not know", "DK"). These terms will be replaced by NA. If NULL, defaults to c("I don't know", "Don't Know", "Don't know")

Value

A data frame
See Also

- `hasDK()` and `removeDK()`

Other Functions to clean data: `encToInt`, `fixCommonEncodingProblems`, `fixLevels01R`, `fixLevels01SPSS`, `fixLevels01`, `hasDK`, `intToEnc`, `leveltestR`, `leveltestSPSS`, `removeDK`

---

**removeDK**

*Removes "Don’t know" from levels and replaces with NA.*

**Description**

Tests the levels of x contain any instances of "Don’t know". If so, replaces these levels with NA.

**Usage**

```r
removeDK(x, dk = "Don't Know")
```

**Arguments**

- `x`: Character vector or factor
- `dk`: Character vector, containing search terms, e.g. `c("Don’t know", "Don’t Know")`

**Value**

A factor with "Dont know" removed

**See Also**

Other Functions to clean data: `encToInt`, `fixCommonEncodingProblems`, `fixLevels01R`, `fixLevels01SPSS`, `fixLevels01`, `hasDK`, `intToEnc`, `leveltestR`, `leveltestSPSS`, `removeAllDK`

---

**rm.attrs**

*Removes pattern and variable.labels from attributes list.*

**Description**

Removes pattern and variable.labels from attributes list.

**Usage**

```r
rm.attrs(x)
```

**Arguments**

- `x`: Surveydata object
rm.pattern

Removes pattern from attributes list.

Description
Removes pattern from attributes list.

Usage
rm.pattern(x)

Arguments
x Surveydata object

splitCommonUnique

Get common and unique text in question based on regex pattern identification

Description
Get common and unique text in question based on regex pattern identification

Usage
splitCommonUnique(x, ptn = NULL)

Arguments
x A character vector
ptn A regex() pattern that defines how the string should be split into common and unique elements

See Also
Other Question functions: qTextCommon, qTextUnique, qText, questions, which.q
strCommonUnique  
*Finds the common and unique elements in a character vector.*

**Description**
Function takes a character string as input and find the common and unique elements. Assumes that the common element is at start of string.

**Usage**
```r
strCommonUnique(string)
```

**Arguments**
- `string` Character vector

**Value**
list of common and unique strings

**Examples**
```r
test <- c("Q_1", "Q_2", "Q_3")
strCommonUnique(test)$common
strCommonUnique(test)$unique
```

---

varlabels  
*Returns and updates variable.labels attribute of surveydata object.*

**Description**
In a surveydata object, the `variable.labels` attribute store metadata about the original question text (see `foreign::read.spss()` for details). The function `varlabels()` returns the `variable.labels` attribute of data, and `varlabels(x) <- value` updates this attribute.

**Usage**
```r
varlabels(x)
"varlabels(x) <- value"
```

**Arguments**
- `x` surveydata object
- `value` New value
Details

In a surveydata object, the varlabels attribute is a named character vector, where the names correspond to the names of the the columns in

See Also

surveydata-package
Other Attribute functions: pattern
Other Attribute functions: pattern

Examples

# Extract the variable labels from membersurvey
ms <- membersurvey[, c("id", "Q1", "Q2")]

str(ms)
varlabels(ms)
varlabels(ms)["Q2"]

# Assign a new value to the text of question 2
varlabels(ms)["Q2"] <- "When did you join?"
varlabels(ms)
str(ms["Q2"])

which.q(x, Q, ptn = pattern(x))

Arguments

x Object to coerce to surveydata
Q Character string with question number, e.g. "Q2"
ptn A list with two elements, sep and exclude. See pattern() and which.q() for more detail.
See Also

questions() to return all questions matching the pattern()

Other Question functions: qTextCommon, qTextUnique, qText, questions, splitCommonUnique

Examples

# Basic operations on a surveydata object, illustrated with the example dataset membersurvey
class(membersurvey)
questions(membersurvey)

which.q(membersurvey, "Q1")
which.q(membersurvey, "Q3")
which.q(membersurvey, c("Q1", "Q3"))

qText(membersurvey, "Q3")
qTextUnique(membersurvey, "Q3")
qTextCommon(membersurvey, "Q3")

# Extracting columns from a surveydata object
head(membersurvey[, "Q1"])
head(membersurvey["Q1"])  # Note that the result is always a surveydata object, even if only one column is extracted
head(membersurvey[, "Q3"])
head(membersurvey[, c("Q1", "Q3")])

str(membersurvey[, "id"])
Index

*Topic **Internal**
  - rm.attrs, 24
  - rm.pattern, 25

*Topic **Questions**
  - qtext, 19
  - qtextCommon, 20
  - qtextUnique, 21
  - questions, 22
  - splitCommonUnique, 25
  - which.q, 27

*Topic **clean**
  - fixLevels01, 11
  - fixLevels01R, 12
  - fixLevels01SPSS, 12
  - hasDK, 13
  - leveltestR, 15
  - leveltestSPSS, 15
  - removeAllDK, 23
  - removeDK, 24

*Topic **datasets**
  - membersurvey, 16

*Topic **encoding**
  - encToInt, 9
  - fixCommonEncodingProblems, 11
  - intToEnc, 13

*Topic **package**
  - surveydata-package, 2

*Topic **string**
  - strCommonUnique, 26
  - [ (Extract), 9
  - [<- . surveydata (Extract), 9
  - $<-$ (Extract), 9

as.data.frame
  - (as.data.frame.surveydata), 5
as.data.frame.surveydata, 5
as.data.frame.surveydata(), 3
as.surveydata, 6
as.surveydata(), 3, 17

cbind.surveydata, 8
cbind.surveydata(), 3
dropout, 8
dropout(), 3
encToInt, 9, 11–15, 24
encToInt(), 3
Extract, 9
fixCommonEncodingProblems, 9, 11, 11–15, 12–15, 24
fixCommonEncodingProblems(), 3
fixLevels01, 9, 11, 11, 12–15, 24
fixLevels01(), 3
fixLevels01R, 9, 11, 12, 12, 13–15, 24
fixLevels01SPSS, 9, 11, 12, 12, 13–15, 24
foreign::read.spss(), 3, 26
hasDK, 9, 11, 12, 13, 14, 15, 24
hasDK(), 24
iconv(), 9, 14
iconvlist(), 9, 11, 13
intToEnc, 9, 11–13, 13, 15, 24
intToEnc(), 3
is.surveydata, 14
is.surveydata(), 3, 6, 17
lapply(), 14
lapplyNames, 14, 18
leveltestR, 9, 11–15, 15, 24
leveltestSPSS, 9, 11–15, 15, 24
membersurvey, 16
merge, 16
merge(), 16
names<-, 17
pattern, 17, 27
pattern(), 3, 5, 6, 10, 17, 22, 27, 28
pattern<=, 17
pattern<=< (pattern), 17

qOrder, 15, 18
qText, 19, 20–22, 25, 28
qText(), 3
qTextCommon, 19, 20, 21, 22, 25, 28
qTextCommon(), 3, 19
qTextUnique, 19, 20, 21, 22, 25, 28
qTextUnique(), 3, 19
questions, 19–21, 22, 25, 28
questions(), 28

regex(), 25
removeAllDK, 9, 11–15, 23, 24
removeAllDK(), 3
removeDK, 9, 11–15, 24, 24
removeDK(), 3, 24
rm.atrs, 24
rm.pattern, 25

splitCommonUnique, 19–22, 25, 28
strCommonUnique, 26
surveydata, 27
surveydata (surveydata-package), 2
surveydata(), 22
surveydata-package, 2, 5, 6, 10, 14, 27
surveydata::Extract(), 3
surveydata::merge(), 3

varlabels, 10, 17, 26
varlabels(), 3, 26
varlabels<=< (varlabels), 26

which.q, 19–22, 25, 27
which.q(), 6, 10, 17, 22, 27