Package ‘surveydata’

January 23, 2019

Version    0.2.3
Date        2019-01-23
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.

URL         http://andrie.github.io/surveydata/index.html
BugReports   https://github.com/andrie/surveydata
ByteCompile  yes
Depends      R (>= 3.0.0)
Imports      stringr (>= 0.5), plyr, dplyr, rlang, magrittr, purrr, ggplot2, scales, tidyr, DT, assertthat
Suggests     testthat, knitr, rmarkdown, withr, covr, rprojroot
RoxygenNote  6.1.1
VignetteBuilder knitr
Encoding     UTF-8
NeedsCompilation no
Author       Andrie de Vries [aut, cre, cph]
Maintainer   Andrie de Vries <apdevries@gmail.com>
Repository   CRAN
Date/Publication 2019-01-23 12:50:04 UTC
R topics documented:

surveydata-package .................................................. 3
as.data.frame.surveydata ........................................... 5
as.surveydata .......................................................... 6
as_opentext_datatable ................................................ 8
cbind.surveydata ..................................................... 9
dropout ................................................................. 9
encToInt ............................................................... 10
Extract ................................................................. 10
fix_common_encoding_problems ...................................... 12
fix_levels_01_spss .................................................... 12
has_dont_know .......................................................... 13
infToEnc ............................................................... 13
is.surveydata .......................................................... 14
lapply_names ........................................................... 14
leveltest ............................................................... 15
membersurvey .......................................................... 15
merge ................................................................. 16
names<-.surveydata ................................................. 16
pattern ............................................................... 17
print_opentext ........................................................ 18
qText ................................................................. 18
questions ............................................................. 19
question_order ......................................................... 21
question_text ........................................................ 21
question_text_common ............................................... 22
question_text_unique ............................................... 23
remove_all_dont_know ............................................... 24
remove_dont_know ..................................................... 25
rm.attrs .............................................................. 26
rm.pattern ........................................................... 26
split_common_unique ................................................. 26
strCommonUnique ..................................................... 27
survey_plot_question ............................................... 28
survey_plot_satisfaction ............................................ 28
survey_plot_title .................................................... 29
survey_plot_yes_no .................................................. 29
varlabels ............................................................. 30
which.q ............................................................. 31

Index .... 33
Description

Tools, classes and methods to manipulate survey data.

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:

- `question_text()`
- `question_text_common()`
- `question_text_unique()`

To fix common encoding problems:

- `encToInt()`
- `intToEnc()`
- `fix_common_encoding_problems()`

To clean data:

- `remove_dont_know()` to remove "Don’t know" responses
- `remove_all_dont_know()` to remove "Don’t know" responses from all questions
- `fix_levels_01()` to fix level formatting of all question with Yes/No type answers

Miscellaneous tools:

- `dropout()` to determine questions where respondents drop out
library(surveydata)

# 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
question_text(sv, "Q1")
question_text(sv, "Q4")

question_text_common(sv, "Q4")
question_text_unique(sv, "Q4")
# Basic operations on a surveydata object, illustrated with the example dataset membersurvey

```r
class(membersurvey)

questions(membersurvey)

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

question_text(membersurvey, "Q3")
question_text_unique(membersurvey, "Q3")
question_text_common(membersurvey, "Q3")
```

# Extracting columns from a surveydata object

```r
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

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

---

**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`
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),
  defaultPtn = list(sep = sep, exclude = exclude),
  renameVarlabels = FALSE)

un_surveydata(x)
```

Arguments

- `x` Object to coerce to surveydata
- `sep` Separator between question and sub-question names
- `exclude` Excludes from pattern search
- `ptn` A list with two elements, `sep` and `exclude`. See `pattern()` and `which.q()` for more detail.
- `defaultPtn` 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.

Details

The function `un_surveydata()` removes the surveydata class from the object, leaving intact the other classes, e.g. data.frame or tibble.

See Also

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

Examples

```r
library(surveydata)

# 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
question_text(sv, "Q1")
question_text(sv, "Q4")

question_text_common(sv, "Q4")
question_text_unique(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"))

question_text(membersurvey, "Q3")
question_text_unique(membersurvey, "Q3")
question_text_common(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"])

---

**as_opentext_datatable**  
Converts free format question text to datatable using the DT package.

## Description

Converts free format question text to datatable using the DT package.

## Usage

```r
as_opentext_datatable(data, q)
```

## Arguments

- **data**  
surveydata object
- **q**  
Question

## See Also

Other open text functions: `print_opentext`

## Examples

```r
as_opentext_datatable(membersurvey, "Q33")
```
**cbind.surveydata**  
*Combines surveydata object by columns.*

**Description**
Combines surveydata object by columns.

**Usage**
```r
## S3 method for class 'surveydata'
cbind(..., deparse.level = 1)
```

**Arguments**
- `...`  
surveydata objects
- `deparse.level`  
ignored

**dropout**  
*Calculates at which questions respondents drop out.*

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

**Usage**
```r
dropout(x, summary = TRUE)
```

**Arguments**
- `x`  
surveydata object, list or data.frame
- `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.

**Value**
Named numeric vector of respondent counts

**Examples**
```r
dropout(membersurvey[(-27:28)])
```
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

\[ \text{encToInt}(x, \text{encoding} = \text{localeToCharset}()) \]

Arguments

- **x**: Character vector
- **encoding**: A character string describing the encoding of \( x \). Defaults to the current locale. See also \text{iconvlist()}\.

Value

An integer vector

See Also

\text{iconv()}

Other Functions to clean data: \text{fix_common_encoding_problems, fix_levels_01_spss, has_dont_know, intToEnc, leveltest, remove_all_dont_know, remove_dont_know}

Examples

\[ \text{encToInt}("\text{xfa}") \]

Extract

Extract or replace subsets of surveydata, ensuring that the varlabels stay synchronized.

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]
```

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

```r
## 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"])```
Description

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

Usage

```r
fix_common_encoding_problems(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`, `fix_levels_01_spss`, `has_dont_know`, `intToEnc`, `leveltest`, `remove_all_dont_know`, `remove_dont_know`

Description

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

Usage

```r
fix_levels_01_spss(dat)
```

```r
fix_levels_01_r(dat)
```

```r
fix_levels_01(dat, origin = c("R", "SPSS"))
```

Arguments

- `dat`: surveydata object
- `origin`: Either R or SPSS
has_dont_known

See Also

Other Functions to clean data: encToInt, fix_common_encoding_problems, has_dont_known, intToEnc, leveltest, remove_all_dont_known, remove_dont_known

Description

Tests whether levels contain "Don’t know".

Usage

has_dont_known(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, fix_common_encoding_problems, fix_levels_01_spss, intToEnc, leveltest, remove_all_dont_known, remove_dont_known

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
dk 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, fix_common_encoding_problems, fix_levels_01_spss, has_dont_know, leveltest, remove_all_dont_know, remove_dont_know

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

lapply_names 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

lapply_names(x, names, FUN, ...)
leveltest

Arguments

\begin{itemize}
\item [x] list
\item [names] character vector identifying which elements of the list to apply FUN
\item [FUN] function to apply.
\item [... \ldots] additional arguments passed to FUN
\end{itemize}

See Also

Other Tools: question_order

\begin{itemize}
\item [leveltest] \textit{Fix level formatting of all question with Yes/No type answers.}
\end{itemize}

Description

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

Usage

leveltest\_spss(x)

leveltest\_r(x)

Arguments

\begin{itemize}
\item [x] surveydata object
\end{itemize}

See Also

Other Functions to clean data: encToInt, fix\_common\_encoding\_problems, fix\_levels\_01\_spss, has\_dont\_know, intToEnc, remove\_all\_dont\_know, remove\_dont\_know

\begin{itemize}
\item [membersurvey] Data frame with survey data of member satisfaction survey.
\end{itemize}

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

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

Arguments

- `x`: surveydata object
- `y`: surveydata object
- `...`: Other parameters passed to `merge()`

names<-.surveydata

Updates names and variable.labels attribute of surveydata.

Description

Updates names and variable.labels attribute of surveydata.

Usage

```r
### S3 replacement method for class 'surveydata'
names(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 sub-questions in the data. Survey software typically makes use of underscores to distinguish sub-questions 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

```r
pattern(x)
pattern(x) <- value
```

Arguments

- `x` surveydata object
- `value` New value

See Also

- `as.surveydata()`, `which.q()`
- Other Attribute functions: `varlabels`

Examples

```r
# 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)

print_opentext  Print open text

Description
Print open text

Usage
print_opentext(data, q, cat = TRUE)

Arguments
data  data
q  Question number
cat  If TRUE, prints results using cat()

See Also
Other open text functions: as_opentext_datatable

Examples
print_opentext(membersurvey, "Q33")

qText  Deprecated functions.

Description
These functions have all been superseded with functions using snake_case function names.

- hasDK: has_dont_know()
- removeDK: remove_dont_know()
- removeAllDK: remove_all_dont_know()
- leveltestSPSS: leveltest_spss()
- leveltestR: leveltest_r()
- fixLevels01SPSS: fix_levels_01_spss()
- fixLevels01R: fix_levels_01_r()
- fixLevels01: fix_levels_01()
- qOrder: question_order()
- lapplyNames: lapply_names()
- fixCommonEncodingProblems: fix_common_encoding_problems()
**Usage**

```
qText(...)  
qTextUnique(...)  
qTextCommon(...)  
hasDK(...)  
removeDK(...)  
removeAllDK(...)  
leveltestSPSS(...)  
leveltestR(...)  
fixLevels01SPSS(...)  
fixLevels01R(...)  
fixLevels01(...)  
qOrder(...)  
lapplyNames(...)  
fixCommonEncodingProblems(...)  
```

**Arguments**

```
... passed to replacement function
```

---

**Description**

In many survey systems, sub-questions 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: `question_text_common`, `question_text_unique`, `question_text`, `split_common_unique`, `which.q`

Examples

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

```r
class(membersurvey)

questions(membersurvey)

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

question_text(membersurvey, "Q3")
question_text_unique(membersurvey, "Q3")
question_text_common(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"])
```
**question_order**

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

**Description**

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

**Usage**

```r
question_order(x)
```

**Arguments**

- `x` character vector

**See Also**

Other Tools: `lapply_names`

---

**question_text**

Returns question text.

**Description**

Given a question id, e.g. "Q4", returns question text for this question. Note that this returns. The functions `question_text_unique()` and `question_text_common()` returns the unique and common components of the question text.

**Usage**

```r
question_text(x, Q)
```

**Arguments**

- `x` A surveydata object
- `Q` The question id, e.g. "Q4". If not supplied, returns the text for all questions.

**Value**

character vector

**See Also**

Other Question functions: `question_text_common`, `question_text_unique`, `questions`, `split_common_unique`, `which.q`
question_text_common

Examples

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

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

question_text(membersurvey, "Q3")
question_text_unique(membersurvey, "Q3")
question_text_common(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"])

question_text_common Returns common element of question text.

Description

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

Usage

question_text_common(x, Q)

Arguments

x A surveydata object
Q The question id, e.g. "Q4". If not supplied, returns the text for all questions.

Value

character vector
question_text_unique

See Also

Other Question functions: question_text_unique, question_text, questions, split_common_unique, 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"))

question_text(membersurvey, "Q3")
question_text_unique(membersurvey, "Q3")
question_text_common(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"])

question_text_unique  Returns unique elements of question text.

Description

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

Usage

question_text_unique(x, Q)

Arguments

x  A surveydata object
Q  The question id, e.g. "Q4". If not supplied, returns the text for all questions.
remove_all_dont_know

Value
character vector

See Also
Other Question functions: question_text_common, question_text, questions, split_common_unique, 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"))

question_text(membersurvey, "Q3")
question_text_unique(membersurvey, "Q3")
question_text_common(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"])

remove_all_dont_know  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
remove_all_dont_know(x, dk = NULL, message = TRUE)
remove_dont_know

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"). message If TRUE, displays message with the number of instances that were removed.

Value

A data frame

See Also

hasDK() and removeDK()

Other Functions to clean data: encToInt, fix_common_encoding_problems, fix_levels_01_spss, has_dont_know, intToEnc, leveltest, remove_dont_know

remove_dont_know    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

remove_dont_know(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, fix_common_encoding_problems, fix_levels_01_spss, has_dont_know, intToEnc, leveltest, remove_all_dont_know
### rm.attrs

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

**Usage**  
\[ \text{rm.attrs}(x) \]

**Arguments**

- \( x \)  
  Surveydata object

### rm.pattern

**Description**  
Removes pattern from attributes list.

**Usage**  
\[ \text{rm.pattern}(x) \]

**Arguments**

- \( x \)  
  Surveydata object

### split_common_unique

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

**Usage**  
\[ \text{split_common_unique}(x, \text{ptn} = \text{NULL}) \]
strCommonUnique

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: question_text_common, question_text_unique, question_text, 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

strCommonUnique(string)

Arguments

- string: Character vector

Value

list of common and unique strings

Examples

test <- c("Q_1", "Q_2", "Q_3")
strCommonUnique(test)$common
strCommonUnique(test)$unique
survey_plot_question  
Plots single and as multi-response questions.

Description
Plots single and as multi-response questions.

Usage
survey_plot_question(data, q)

Arguments
- data: surveydata object
- q: Question

See Also
Other survey plotting functions: survey_plot_satisfaction, survey_plot_yes_no

Examples
question_text(membersurvey)
survey_plot_question(membersurvey, "Q2")
survey_plot_yes_no(membersurvey, "Q2")
survey_plot_satisfaction(membersurvey, "Q14")

survey_plot_satisfaction  
Plot satisfaction

Description
Plot satisfaction

Usage
survey_plot_satisfaction(data, q, fun = c("net", "top3", "top2"))

Arguments
- data: surveydata object
- q: Question
- fun: Aggregation function, one of net (compute net satisfaction score), top3 (compute top 3 box score) and top2 (compute top 2 box score)
survey_plot_title

See Also

Other survey plotting functions: survey_plot_question, survey_plot_yes_no

Examples

question_text(membersurvey)

survey_plot_question(membersurvey, "Q2")
survey_plot_yes_no(membersurvey, "Q2")
survey_plot_satisfaction(membersurvey, "Q14")

---

survey_plot_title  Construct plot title from the question text, wrapping at the desired width.

---

Description

This creates a plot title using \scriptsize{ggplot2::ggtitle()}. The main title is string wrapped, and the subtitle is the number of observations in the data.

Usage

survey_plot_title(data, q, width = 50)

Arguments

data surveydata object
q Question
width Passed to \scriptsize{strwrap()}

---

survey_plot_yes_no  Plot data in yes/no format.

---

Description

Plot data in yes/no format.

Usage

survey_plot_yes_no(data, q)

Arguments

data surveydata object
q Question
See Also

Other survey plotting functions: `survey_plot_question`, `survey_plot_satisfaction`

Examples

```r
question_text(membersurvey)

survey_plot_question(membersurvey, "Q2")
survey_plot_yes_no(membersurvey, "Q2")
survey_plot_satisfaction(membersurvey, "Q14")
```

<table>
<thead>
<tr>
<th>varlabels</th>
<th>Returns and updates variable.labels attribute of surveydata object.</th>
</tr>
</thead>
</table>

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

```
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 columns in

See Also

- `surveydata-package`
- Other Attribute functions: `pattern`
- Other Attribute functions: `pattern`
**which.q**

**Examples**

```r
# 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**  
Identifies the columns indices corresponding to a specific question.

**Description**

In many survey systems, sub-questions take the form "Q1_a", "Q1_b", with the main question and sub-question separated by an underscore. This function conveniently returns column index of matches found for a question id in a surveydata object. It does this by using the pattern attribute of the surveydata object.

**Usage**

```
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: `question_text_common, question_text_unique, question_text, questions, split_common_unique`
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"))

question_text(membersurvey, "Q3")
question_text_unique(membersurvey, "Q3")
question_text_common(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")

str(membersurvey[, "id"])
Index

*Topic **Internal**
  rm.attrs, 26
  rm.pattern, 26

*Topic **Questions**
  question_text, 21
  question_text_common, 22
  question_text_unique, 23
  questions, 19
  split_common_unique, 26
  which.q, 31

*Topic **clean**
  fix_levels_01_spss, 12
  has_dont_know, 13
  leveltest, 15
  remove_all_dont_know, 24
  remove_dont_know, 25

*Topic **datasets**
  membersurvey, 15

*Topic **encoding**
  encToInt, 10
  fix_common_encoding_problems, 12
  intToEnc, 13

*Topic **package**
  surveydata-package, 3

*Topic **string**
  strCommonUnique, 27

\[
(\text{Extract}), 10
\]
\[
<-\text{surveydata}(\text{Extract}), 10
\]
\[
<-\text{(Extract)}, 10
\]

as.data.frame
  \(\text{as.data.frame.surveydata}, 5\)
  \(\text{as.data.frame.surveydata}, 5\)
  \(\text{as.data.frame.surveydata()}, 3\)
  \(\text{as.surveydata}, 6\)
  \(\text{as.surveydata()}, 3, 17\)
  \(\text{as_opentext_datatable}, 8, 18\)

\[
\text{cbind.surveydata}, 9
\]
\[
\text{cbind.surveydata()}, 3
\]

dropout, 9
dropout(), 3
encToInt, 10, 12–15, 25
encToInt(), 3
Extract, 10

fix_common_encoding_problems, 10, 12, 13–15, 25
fix_common_encoding_problems(), 3, 18
fix_levels_01_spss, 12
fix_levels_01(), 3, 18
fix_levels_01_r, 18
fix_levels_01_spss, 10, 12, 13–15, 25
fix_levels_01_spss(), 18
fixCommonEncodingProblems(qText), 18
fixLevels01, 18
fixLevels01R(qText), 18
fixLevels01SPSS(qText), 18
foreign::read.spss(), 3, 30

has_dont_know, 10, 12, 13, 14, 15, 25
has_dont_know(), 18
hasDK, 18
hasDK(), 25

iconv(), 10, 14
iconvlist(), 10, 12, 13
intToEnc, 10, 12, 13, 15, 25
intToEnc(), 3
is.surveydata, 14
is.surveydata(), 3, 6, 16

lapply(), 14
lapply_names, 14, 21
lapply_names(), 18
lapplyNames(qText), 18
leveltest, 10, 12–14, 15, 25
leveltest_r(leveltest), 15
leveltest_r(), 18
leveltest_spss (leveltest), 15
leveltest_spss(), 18
leveltestR (qtext), 18
leveltestSPSS (qtext), 18
membersurvey, 15
merge, 16
merge(), 16
names<- .surveydata, 16
names<- (names<- .surveydata), 16
pattern, 17, 30, 31
pattern(), 3, 5, 6, 11, 17, 19, 20, 31
pattern<- , 17
pattern<- (pattern), 17
print_opentext, 8, 18
qOrder (qtext), 18
qText, 18
qTextCommon (qtext), 18
qTextUnique (qtext), 18
question_order, 15, 21
question_order(), 18
question_text, 20, 21, 23, 24, 27, 31
question_text(), 3
question_text_common, 20, 21, 22, 24, 27, 31
question_text_common(), 3, 21
question_text_unique, 20, 21, 23, 27, 31
question_text_unique(), 3, 21
questions, 19, 21, 23, 24, 27, 31
questions(), 31
regex(), 27
remove_all_dont_know, 10, 12–15, 24, 25
remove_all_dont_know(), 3, 18
remove_dont_know, 10, 12–15, 25, 25
remove_dont_know(), 3, 18
removeAllDK (qtext), 18
removeDK (qtext), 18
removeDK(), 25
rm.attrs, 26
rm.pattern, 26
split_common_unique, 20, 21, 23, 24, 26, 31
strCommonUnique, 27
strwrap(), 29
survey_plot_question, 28, 29, 30
survey_plot_satisfaction, 28, 28, 30
survey_plot_title, 29
survey_plot_yes_no, 28, 29, 29
surveydata, 31
surveydata (surveydata-package), 3
surveydata(), 19
surveydata-package, 3, 5, 6, 11, 14, 30
surveydata::Extract(), 3
surveydata::merge(), 3
un_surveydata (as.surveydata), 6
varlabels, 11, 17, 30
varlabels(), 3, 30
varlabels<- (varlabels), 30
which<- q, 20, 21, 23, 24, 27, 31
which.q(), 6, 11, 17, 20, 31