Title  Convert Categorical Representations of Logicals to Actual Logicals

Version  0.1.0

Date  2015-10-13

Author  Oliver Keyes [aut, cre], Ruben C. Arslan [ctb], Christopher Akiki [ctb], Mine Cetinkaya-Rundel [ctb], Peter Meissner [ctb], Ilaria Prosdocimi [ctb], Thomas Leeper [ctb], Amy Lee [ctb], Adolfo Álvarez [ctb]

Maintainer  Oliver Keyes <ironholds@gmail.com>

Description  Survey systems and other third-party data sources commonly use non-standard representations of logical values when it comes to qualitative data - ``Yes'', ``No'' and ``N/A'', say. batman is a package designed to seamlessly convert these into logicals. It is highly localised, and contains equivalents to boolean values in languages including German, French, Spanish, Italian, Turkish, Chinese and Polish.

License  MIT + file LICENSE

Encoding  UTF-8

LazyData  true

URL  https://github.com/ironholds/batman

BugReports  https://github.com/ironholds/batman/issues

Suggests  testthat

LinkingTo  Rcpp

Imports  Rcpp

NeedsCompilation  yes

Repository  CRAN

Date/Publication  2015-10-26 17:07:24
**R topics documented:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>batman</td>
<td>Convert categorical representations of logicals to actual logicals</td>
</tr>
<tr>
<td>categorical_booleans</td>
<td>TRUE/FALSE equivalents in categorical data for various languages</td>
</tr>
<tr>
<td>get_languages</td>
<td></td>
</tr>
<tr>
<td>to_logical</td>
<td></td>
</tr>
</tbody>
</table>

### Description

Survey systems and other third-party data sources commonly use non-standard representations of logical values when it comes to qualitative data - "Yes", "No" and "N/A", say. batman is a package designed to seamlessly convert these into actual logical values.

### See Also

- to_logical

### categorical_booleans

**TRUE/FALSE equivalents in categorical data for various languages**

**Description**

A dataset containing the equivalents of TRUE or FALSE in categorical or user-submitted data, localised to various languages.

**Usage**

```r
categorical_booleans
```

**Format**

A list of named lists, each one containing two columns:

- **true** a character vector of equivalents to TRUE
- **false** a character vector of equivalents to FALSE

**See Also**

- to_logical, which uses this dataset, and get_languages to see what languages are available.
**get_languages**

Get language codes for batman-supported languages

**Description**

retrieves a list of language codes for languages supported by the language parameter in to_logical.

**Usage**

```r
get_languages()
```

**See Also**

categorical_booleans, the underlying dataset, or to_logical, which uses that dataset.

**Examples**

```r
get_languages()
# [1] "en"
```

**to_logical**

Convert categorical representations of true/false values to a logical

**Description**

to_logical is designed for the situation where you have categorical data (perhaps from a survey, or direct user input) that represents TRUE/FALSE values as something other than TRUE/FALSE - "Yes", "No", "None", "Y" or "False", say. With to_logical you can easily convert a vector of these values into an actual, logical vector, using either a predefined set of accepted TRUE or FALSE equivalents, or a set you specify yourself.

**Usage**

```r
to_logical(x, language = "en", custom_true = character(),
          custom_false = character())
```

**Arguments**

- `x` a vector of categorical TRUE/FALSE/NA values.
- `language` the language to use. See get_languages for the list of supported languages. If your language is not supported, you can use custom_true and custom_false to provide values.
- `custom_true` a vector of values to consider, in addition to the ones to_logical already recognises, TRUE. Empty by default. Note that the comparison code is case-insensitive, so there’s no need to include (for example) both "ja" and "Ja".
- `custom_false` a vector of values to consider, in addition to the ones to_logical already recognises, FALSE. Empty by default; see the notes on case sensitivity above.
Examples

# A very simple example using the pre-known true and false equivalents
categorical_values <- c("true","t","y","yes","f","no","1")
to_logical(categorical_values)

# Use a custom specifier
categorical_values <- c("NA","NA","NA","NA","NA","NA","NA","NA","Batman")
to_logical(categorical_values, custom_true = c("Batman"))
Index

*Topic datasets
  categorical_booleans, 2

batman, 2
batman-package (batman), 2

categorical_booleans, 2, 3
get_languages, 2, 3
to_logical, 2, 3, 3