Package ‘unstruwwel’

January 19, 2021

Title  Detect and Parse Historic Dates
Version  0.1.0
Maintainer  Stefanie Schneider <stefanie.schneider@itg.uni-muenchen.de>
Description  Automatically converts language-specific verbal information, e.g., "1st half of the 19th century," to its standardized numerical counterparts, e.g., "1801-01-01/1850-12-31." It follows the recommendations of the 'MIDAS' ('Marburger Informations-, Dokumentations- und Administrations-System'), see <doi:10.11588/artdok.00003770>.
License  GPL-3
Encoding  UTF-8
LazyData  true
RoxygenNote  7.1.1
URL  https://github.com/stefanieschneider/unstruwwel
BugReports  https://github.com/stefanieschneider/unstruwwel/issues
Suggests  testthat, roxygen2
Imports  R6, assertthat, lubridate, magrittr, stringr, tibble, tidyr, purrr, dplyr, rlang
Depends  R (>= 2.10)
NeedsCompilation  no
Author  Stefanie Schneider [cre, aut] (<https://orcid.org/0000-0003-4915-6949>)
Repository  CRAN
Date/Publication  2021-01-19 10:10:02 UTC

R topics documented:

   Century .......................................................... 2
   Decade .................................................................. 3
   languages ............................................................. 4
   midas .................................................................... 4
   Periods ................................................................. 5
   schemes .................................................................. 6
   unstruwwel ............................................................. 7
   Year ..................................................................... 8
Description

Set a Century and Get its Time Interval
Set a Century and Get its Time Interval

Details

An Object of \texttt{R6Class} with methods to set common time periods and specifications for centuries.

Super class

\texttt{unstruwel::Periods} \rightarrow \texttt{Century}

Methods

\textbf{Public methods:}

- \texttt{Century$new()}
- \texttt{Century$clone()}

\textbf{Method} \texttt{new():} Helper function to specify the beginning of a century.
Helper function to specify the middle of a century.
Helper function to specify the end of a century.
Create a century.
\textit{Usage:}
\texttt{Century$new(value)}
\textit{Arguments:}
value A numerical scalar.
\textit{Returns:} Object of \texttt{R6Class} with methods to set common time periods and specifications for centuries.

\textbf{Method} \texttt{clone():} The objects of this class are cloneable with this method.
\textit{Usage:}
\texttt{Century$clone(deep = FALSE)}
\textit{Arguments:}
deep Whether to make a deep clone.

Examples

```r
if (interactive()) {
  x <- Century$new(15)
  x$take(2, type = "third")
}
```
Decade

Set a Decade and Get its Time Interval

Description

Set a Decade and Get its Time Interval
Set a Decade and Get its Time Interval

Details

An Object of \texttt{R6Class} with methods to set common time periods and specifications for decades.

Super class

\texttt{unstruwel::Periods} -> Decade

Methods

Public methods:

- \texttt{Decade$new()}
- \texttt{Decade$clone()}

Method \texttt{new()}:
Helper function to specify the beginning of a decade.
Helper function to specify the middle of a decade.
Helper function to specify the end of a decade.
Create a decade.

Usage:
\texttt{Decade$new(value, official\_def = FALSE)}

Arguments:

value A numerical scalar.
official\_def If ‘TRUE’, the official definition that begins with the year 1 is used.

Returns: Object of \texttt{R6Class} with methods to set common time periods and specifications for decades.

Method \texttt{clone()}:
The objects of this class are cloneable with this method.

Usage:
\texttt{Decade$clone(deep = FALSE)}

Arguments:

depth Whether to make a deep clone.

Examples

if (interactive()) {
  x <- Decade$new(1520)
  x$take(1, type = "half")
}

languages

<table>
<thead>
<tr>
<th>Language Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>A dataset containing the names, date orders, stop words, simplifications, and replacements of 4 languages.</td>
</tr>
<tr>
<td>Usage</td>
</tr>
<tr>
<td>data(languages)</td>
</tr>
<tr>
<td>Format</td>
</tr>
<tr>
<td>A tibble with 4 rows and 5 variables.</td>
</tr>
</tbody>
</table>

midas

<table>
<thead>
<tr>
<th>MIDAS Standardization Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>A dataset containing eight thousand standardization examples of the MIDAS (Marburger Informations-, Dokumentations- und Administrations-System).</td>
</tr>
<tr>
<td>Usage</td>
</tr>
<tr>
<td>data(midas)</td>
</tr>
<tr>
<td>Format</td>
</tr>
<tr>
<td>A vector of length 8115.</td>
</tr>
</tbody>
</table>
Description

Set a Period and Get its Time Interval
Set a Period and Get its Time Interval

Details

An Object of \texttt{R6Class} with methods to set common time periods and specifications for time periods.

Public fields

\begin{itemize}
\item \texttt{.interval} Stores a time interval.
\item \texttt{fuzzy} Either '-1' (approximate) or '1' (uncertain).
\item \texttt{express} Either '-1' (before) or '1' (after).
\end{itemize}

Active bindings

\begin{itemize}
\item \texttt{.interval} Stores a time interval.
\item \texttt{interval} Convert and return a POSIXt time interval.
\item \texttt{time_span} Convert and return a time span in years.
\item \texttt{iso_format} Convert and return a date in ISO 8601.
\end{itemize}

Methods

Public methods:

\begin{itemize}
\item \texttt{Periods$new()}
\item \texttt{Periods$set_additions()}
\item \texttt{Periods$take()}
\item \texttt{Periods$clone()}
\end{itemize}

Method \texttt{new()}: Helper function to specify a time period.

Create a time period.

\textit{Usage:}

\texttt{Periods$new(...)}

\textit{Arguments:}

\texttt{...} Intervals, numerical scalars, or objects of class \texttt{Period}.
\texttt{x} A numerical scalar. The range of valid values depends on type. If type is "early", "mid", or "late", \texttt{x} is ignored.
\texttt{type} A character scalar. The following values are supported: "early", "mid", "late", "quarter", "third", and "half". If type is 'NULL', \texttt{x} defines a year or decade.
**Method** `set_additions()`: Set additions for a time period.

*Usage:*

```r
Periods$set_additions(x)
```

*Arguments:*

- `x` A character vector.

**Method** `take()`: Specify a period.

*Usage:*

```r
Periods$take(x = NA, type = NA, ignore_errors = FALSE)
```

*Arguments:*

- `x` A numerical scalar. The range of valid values depends on `type`. If `type` is "early", "mid", or "late", `x` is ignored.
- `type` A character scalar. The following values are supported: "early", "mid", "late", "quarter", "third", and "half". If `type` is 'NULL', `x` defines a year or decade.
- `ignore_errors` If 'TRUE', error messages are ignored.

*Returns:* Object of `R6Class` with methods to set common time periods and specifications for time periods.

**Method** `clone()`: The objects of this class are cloneable with this method.

*Usage:*

```r
Periods$clone(deep = FALSE)
```

*Arguments:*

- `deep` Whether to make a deep clone.

---

### Language-Specific Scheme Variants

**Description**

A dataset containing the values, schemes, and languages for over three thousand language-specific scheme variants.

**Usage**

```r
data(schemes)
```

**Format**

A tibble with 3583 rows and 3 variables.
unstruwwel  Detect and Parse Historic Dates

**Description**


**Usage**

```r
unstruwwel(
  x,
  language = NULL,
  verbose = TRUE,
  scheme = "time-span",
  fuzzify = c(0, 0)
)
```

**Arguments**

- `x` Input vector. Either a character vector, or something coercible to one.
- `language` Language code of the input vector as defined in ISO 639-1. If NULL, language is detected automatically.
- `verbose` If TRUE, additional diagnostics are printed.
- `scheme` Scheme code of the output list. Either time-span, iso-format, or object.
- `fuzzify` A numerical vector of length 2 to extend the interval of approximate or uncertain time periods. This is only applied if scheme == "time-span".

**Value**

A named list of vectors or objects of R6Class.

**Note**

Although multiple languages can be detected, only dominant ones are ultimately set.

**Examples**

```r
if (interactive()) {
  unstruwwel("1. Hälfte 19. Jahrhundert", language = "de")
  unstruwwel("circa between 1901 and 1905", language = "en")
}
Set a Year and Get its Time Interval

Description
Set a Year and Get its Time Interval
Set a Year and Get its Time Interval

Details
An Object of \texttt{R6Class} with methods to set common time periods and specifications for years.

Super class
\texttt{unstruwel::Periods} -> Year

Methods

Public methods:
- \texttt{Year$new()}
- \texttt{Year$take()}
- \texttt{Year$clone()}

Method \texttt{new()}: Helper function to specify a time period.
Helper function to specify a season.
Helper function to specify a month.
Create a year.

Usage:
\texttt{Year$new(value)}

Arguments:
value A numerical scalar.

Returns: Object of \texttt{R6Class} with methods to set common time periods and specifications for years.

Method \texttt{take()}: Specify a year.

Usage:
\texttt{Year$take(x = NA, type = NA, ignore_errors = FALSE)}

Arguments:
x A numerical scalar. The range of valid values depends on type. If type is "spring", "summer", "autumn", or "winter", \texttt{x} is ignored.
type A character scalar. The following values are supported: "spring", "summer", "autumn", "winter", and all English-language months.
ignore_errors If 'TRUE', error messages are ignored.
Returns: Object of `R6Class` with methods to set common time periods and specifications for years.

Method `clone()`: The objects of this class are cloneable with this method.

Usage:
`Year$clone(deep = FALSE)`

Arguments:
deep Whether to make a deep clone.

Examples
```r
if (interactive()) {
  x <- Year$new(1520)
x$take(15, type = "june")
}
```
Index

* datasets
  languages, 4
  midas, 4
  schemes, 6

Century, 2
Decade, 3
languages, 4
midas, 4
Periods, 5
R6Class, 2, 3, 5–9
schemes, 6
unstruwwel, 7
unstruwwel::Periods, 2, 3, 8
Year, 8