Package 'libbib'

February 9, 2018

Type Package

Title Verification and Normalization of Various Bibliographic Codes

Version 0.1.0

Author Tony Fischetti

Maintainer Tony Fischetti <tony.fischetti@gmail.com>

Description Provides functions for validating and normalizing bibliographic codes such as ISBN, ISSN, LCCN, and OCLC.

License MIT + file LICENSE

LazyData TRUE

Imports stringr

Suggests assertr, testthat, knitr, magrittr

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 6.0.1

R topics documented:

check_isbn_10_check_digit

Check the check digit of an ISBN 10

Description

Takes a string representation of an ISBN 10 and verifies that check digit checks out

Usage

```
check_isbn_10_check_digit(x, error.is.false = FALSE)
```

Arguments

x	A string of 10 digits or 9 digits with terminal "X"
error.is.false	return false if error instead of throwing error

Value

Returns TRUE if check passes, FALSE if not, and NA if NA

Examples

```
check_isbn_10_check_digit("012491540X")  # TRUE

# vectorized
check_isbn_10_check_digit(c("012491540X", "9004037812"))  # TRUE FALSE
```

check_issn_check_digit

Check the check digit of an ISSN

Description

Takes a string representation of an ISSN and verifies that check digit checks out

Usage

```
check_issn_check_digit(x, allow.hyphens = FALSE, error.is.false = FALSE)
```

Arguments

x	A string of 8 digits or 7 digits with terminal "X"
allow.hyphens	A logical indicating whether the hyphen separator should be allowed
error.is.false	return false if error instead of throwing error

Value

Returns TRUE if check passes, FALSE if not, and NA if NA

Examples

```
check_issn_check_digit("2434561X")  # TRUE

# vectorized
check_issn_check_digit(c("03785955", "2434561X", NA)  # TRUE TRUE NA
check_issn_check_digit(c("03785955", "2434-561X", NA),
                       allow.hyphens=TRUE)  # TRUE TRUE NA
```

---

get_isbn_10_check_digit

Get ISBN 10 check digit

Description

Takes a string representation of an ISBN 10 and returns the check digit that satisfies the necessary condition. It can take a 10 digit string (and ignore the already extant check digit) or a 9 digit string (without the last digit)

Usage

```
get_isbn_10_check_digit(x)
```

Arguments

x	A string of nine or 10 digits

Value

Returns the character check digit that satifies the mod 11 condition. Returns "X" if 10. Returns NA if input is NA

Examples

```
get_isbn_10_check_digit("012491540X")

# nine digit string
get_isbn_10_check_digit("900403781")

# vectorized
get_isbn_10_check_digit(c("012491540X", "9004037810", "900403781"))
```

get_issn_check_digit *Get ISSN check digit*

Description

Takes a string representation of an ISSN and returns the check digit that satisfies the necessary condition. It can take a 8 digit string (and ignore the already extant check digit) or a 7 digit string (without the last digit)

Usage

```
get_issn_check_digit(x, allow.hyphens = FALSE)
```

Arguments

x	A string of 7 or 8 digits
allow.hyphens	A logical indicating whether the hyphen separator should be allowed

Value

Returns the character check digit that satifies the mod 11 condition. Returns "X" if 10. Returns NA if input is NA

Examples

```
get_issn_check_digit("03785955")

get_issn_check_digit("2434-561X", allow.hyphens=TRUE)

# nine digit string
get_issn_check_digit("0378595")

# vectorized
get_issn_check_digit(c("0378595", "2434561X", NA))
```

is_valid_isbn_10 *Return TRUE if valid ISBN 10*

Description

Takes a string representation of an ISBN 10 verifies that it is valid. An ISBN 10 is valid if it is a 10 digit string or a 9 digit string with a terminal "X" AND the check digit matches

Usage

```
is_valid_isbn_10(x, lower.x.allowed = TRUE)
```

Arguments

x	A string of or 10 digits or nine digits with terminal "X"
lower.x.allowed	A logical indicating whether ISBN 10s with a check digit with a lower-case "x" should be treated as valid

Value

Returns TRUE if checks pass, FALSE if not, and NA if NA

Examples

```
is_valid_isbn_10("012491540X")  # TRUE

# vectorized
is_valid_isbn_10(c("012491540X", "9004037812"))  # TRUE FALSE
is_valid_isbn_10(c("012491540X", "hubo un tiempo"))  # TRUE FALSE
```

---

normalize_isbn_10 *Attempt to enforce validity and canonical form to ISBN 10*

Description

Takes a string representation of an ISBN 10. Strips all non-digit or "X" characters and checks if it is valid (whether the check digit works out, etc). User can specify whether "aggresive" measures should be taken to salvage the malformed ISBN 10 string.

Usage

```
normalize_isbn_10(x, aggresive = TRUE)
```

Arguments

x	A string of or 10 digits or nine digits with terminal "X"
aggresive	A logical indicating whether aggresive measures should be taken to try to get the "ISBN 10" into a valid form. See "Details" for more info

Details

If aggresive is TRUE, aggresive measures are taken to try to salvage the malformed ISBN 10 string. If the ISBN 10, for example, is 9 digits, and either adding an "X" to the end, or leading "0"s fix it, this function will return the salvaged ISBN 10

Value

Returns TRUE if checks pass, FALSE if not, and NA if NA

Examples

```
is_valid_isbn_10("012491540X")  # TRUE

# vectorized
is_valid_isbn_10(c("012491540X", "9004037812"))  # TRUE FALSE
is_valid_isbn_10(c("012491540X", "hubo un tiempo"))  # TRUE FALSE
```

Index