Package ‘naaccr’

December 17, 2019

Type Package

Title Read Cancer Records in the NAACCR Format

Version 1.0.0

Author Nathan Werth [aut, cre],
Pennsylvania Department of Health [cph],
North American Association of Cancer Registries [cph],
World Health Organization [cph],
United States Centers for Disease Control and Prevention [ctb],
United States Bureau of the Census [ctb],
United States National Program of Cancer Registries [ctb]

Maintainer Nathan Werth <nwerth@pa.gov>

Description Functions for reading cancer record files which follow a format
defined by the North American Association of Central Cancer Registries
(NAACCR).

URL https://github.com/WerthPADOH/naaccr

BugReports https://github.com/WerthPADOH/naaccr/issues

Imports data.table, stringi, utils

Suggests devtools, httr, jsonlite, magrittr, testthat, ISOcodes, xml2,
markdown, roxygen2, rvest

License MIT + file LICENSE

Copyright file COPYRIGHTS

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-12-17 12:20:05 UTC
### R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>as.naaccr_record</td>
<td>2</td>
</tr>
<tr>
<td>clean_address_city</td>
<td>3</td>
</tr>
<tr>
<td>clean_address_number_and_street</td>
<td>4</td>
</tr>
<tr>
<td>clean_age</td>
<td>4</td>
</tr>
<tr>
<td>clean_census_block</td>
<td>5</td>
</tr>
<tr>
<td>clean_census_tract</td>
<td>5</td>
</tr>
<tr>
<td>clean_count</td>
<td>6</td>
</tr>
<tr>
<td>clean_county_fips</td>
<td>6</td>
</tr>
<tr>
<td>clean_facility_id</td>
<td>7</td>
</tr>
<tr>
<td>clean_icd_9_cm</td>
<td>7</td>
</tr>
<tr>
<td>clean_icd_code</td>
<td>8</td>
</tr>
<tr>
<td>clean_physician_id</td>
<td>8</td>
</tr>
<tr>
<td>clean_postal</td>
<td>9</td>
</tr>
<tr>
<td>clean_ssn</td>
<td>9</td>
</tr>
<tr>
<td>clean_telephone</td>
<td>10</td>
</tr>
<tr>
<td>clean_text</td>
<td>10</td>
</tr>
<tr>
<td>field_levels</td>
<td>11</td>
</tr>
<tr>
<td>naaccr_boolean</td>
<td>11</td>
</tr>
<tr>
<td>naaccr_date</td>
<td>12</td>
</tr>
<tr>
<td>naaccr_datetime</td>
<td>12</td>
</tr>
<tr>
<td>naaccr_encode</td>
<td>13</td>
</tr>
<tr>
<td>naaccr_factor</td>
<td>14</td>
</tr>
<tr>
<td>naaccr_format_12</td>
<td>14</td>
</tr>
<tr>
<td>naaccr_override</td>
<td>15</td>
</tr>
<tr>
<td>naaccr_record</td>
<td>16</td>
</tr>
<tr>
<td>read_naaccr_plain</td>
<td>16</td>
</tr>
<tr>
<td>record_format</td>
<td>18</td>
</tr>
<tr>
<td>split_sentineled</td>
<td>20</td>
</tr>
<tr>
<td>split_sequence_number</td>
<td>21</td>
</tr>
<tr>
<td>unknown_to_na</td>
<td>22</td>
</tr>
<tr>
<td>write_naaccr</td>
<td>23</td>
</tr>
</tbody>
</table>

---

**Description**

Coerce to a `naaccr_record` dataset. Convert objects into `naaccr_record` objects, if a method exists.
**clean_address_city**

**Usage**

```r
as.naaccr_record(x, keep_unknown = FALSE, version = NULL, format = NULL, ...)
```

```r
## S3 method for class 'list'
as.naaccr_record(x, keep_unknown = FALSE, version = NULL, format = NULL, ...)
```

```r
## S3 method for class 'data.frame'
as.naaccr_record(x, keep_unknown = FALSE, version = NULL, format = NULL, ...)
```

**Arguments**

- `x` An R object.
- `keep_unknown` Logical indicating whether values of "unknown" should be a level in the factor or NA.
- `version` An integer specifying the NAACCR format version for parsing the records. Use this or `format`, not both. If both `version` and `format` are NULL (default), the most recent NAACCR format will be used.
- `format` A `record_format` object for parsing the records.
- `...` Additional arguments passed to or from methods.

**Value**

An object of class `naaccr_record`

**See Also**

- `naaccr_record`

---

**clean_address_city**  
*Clean city names*

**Description**

Clean city names

**Usage**

```r
clean_address_city(city, keep_unknown = FALSE)
```

**Arguments**

- `city` A character vector of city names.
- `keep_unknown` Replace values for "unknown" with NA?
clean_age

Value

A character vector with leading and trailing whitespace removed. If keep_unknown is FALSE, blanks and "UNKNOWN" are replaced with NA.

clean_address_number_and_street

Clean house number and street values

Description

Clean house number and street values

Usage

clean_address_number_and_street(location, keep_unknown = FALSE)

Arguments

location A character vector of house numbers and street names.
keep_unknown Replace values for "unknown" with NA?

Value

A character vector with leading and trailing whitespace removed. If keep_unknown is FALSE, blanks and "UNKNOWN" are replaced with NA.

clean_age

Clean patient ages

Description

Clean patient ages

Usage

clean_age(age, keep_unknown = FALSE)

Arguments

age Age_at_Diagnosis values.
keep_unknown Replace values for "unknown" with NA?

Value

An integer vector of ages. If keep_unknown is FALSE, values representing unknown ages are replaced with NA.
**clean_census_block**  
*Clean Census block group codes*

**Description**

Clean Census block group codes

**Usage**

```r
clean_census_block(block, keep_unknown = FALSE)
```

**Arguments**

- `block` A character vector of Census block group codes.
- `keep_unknown` Replace values for "unknown" with NA?

**Value**

A character vector with leading and trailing whitespace removed. If keep_unknown is FALSE, blanks and values representing unknown block groups are replaced with NA.

---

**clean_census_tract**  
*Clean Census tract group codes*

**Description**

Clean Census tract group codes

**Usage**

```r
clean_census_tract(tract, keep_unknown = FALSE)
```

**Arguments**

- `tract` A character vector of Census tract group codes.
- `keep_unknown` Replace values for "unknown" with NA?

**Value**

A character vector with leading and trailing whitespace removed. If keep_unknown is FALSE, blanks and values representing unknown Census Tracts are replaced with NA.
### clean_count

**Clean counts**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replaces any values of all 9’s with <code>NA</code> (if <code>keep_unknown</code> is <code>TRUE</code>) and converts the rest to integers.</td>
</tr>
</tbody>
</table>

**Usage**

```r
clean_count(count, width, keep_unknown = FALSE)
```

**Arguments**

- **count**: A character vector of counts (integer characters only).
- **width**: Integer giving the character width of the field.
- **keep_unknown**: Replace values for "unknown" with `NA`?

**Value**

Integer vector of `count`. If `keep_unknown` is `FALSE`, values representing unknown counts are replaced with `NA`.

### clean_county_fips

**Clean county FIPS codes**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean county FIPS codes</td>
</tr>
</tbody>
</table>

**Usage**

```r
clean_county_fips(county, keep_unknown = FALSE)
```

**Arguments**

- **county**: A character vector of county FIPS codes.
- **keep_unknown**: Replace values for "unknown" with `NA`?

**Value**

A character vector with leading and trailing whitespace removed. If `keep_unknown` is `FALSE`, blanks and values representing unknown counties are replaced with `NA`. 
clean_facility_id

**Description**

Clean facility identification numbers

**Usage**

`clean_facility_id(fin, keep_unknown = FALSE)`

**Arguments**

- `fin` A character vector of facility identification numbers (FIN).
- `keep_unknown` Replace values for "unknown" with NA?

**Value**

A character vector with leading and trailing whitespace removed. If `keep_unknown` is FALSE, blanks and values representing unknown facilities are replaced with NA.

---

clean_icd_9_cm

**Description**

Clean ICD-9-CM codes

**Usage**

`clean_icd_9_cm(code, keep_unknown = FALSE)`

**Arguments**

- `code` A character vector of ICD-9-CM codes.
- `keep_unknown` Replace values for "unknown" with NA?

**Value**

A character vector with leading and trailing whitespace removed. If `keep_unknown` is FALSE, blanks and the ICD-9-CM code for "unknown" ("00000") are replaced with NA.
clean_icd_code  
*Clean cause of death codes*

**Description**

Clean cause of death codes

**Usage**

```r
clean_icd_code(code, keep_unknown = FALSE)
```

**Arguments**

- `code`: A character vector of ICD-7, ICD-8, ICD-9, and/or ICD-10 codes.
- `keep_unknown`: Replace values for "unknown" with NA?

**Value**

A character vector with leading and trailing whitespace removed. If `keep_unknown` is FALSE, blanks and the ICD codes for "unknown" ("0000", "7777" and "7797") are replaced with NA.

---

clean_physician_id  
*Clean physician identification numbers*

**Description**

Clean physician identification numbers

**Usage**

```r
clean_physician_id(physician, keep_unknown = FALSE)
```

**Arguments**

- `physician`: A character vector of medical license number or facility-generated codes.
- `keep_unknown`: Replace values for "unknown" with NA?

**Value**

A character vector with leading and trailing whitespace removed. If `keep_unknown` is FALSE, blanks and values representing unknown physicians or non-applicable are replaced with NA.
clean_postal

Clean postal codes

**Description**

Clean postal codes

**Usage**

```r
clean_postal(postal, keep_unknown = FALSE)
```

**Arguments**

- `postal`: A character vector of postal codes.
- `keep_unknown`: Replace values for "unknown" with NA?

**Value**

A character vector with leading and trailing whitespace removed. If `keep_unknown` is FALSE, blanks and values representing uncertain postal codes are replaced with NA.

---

clean_ssn

Clean Social Security ID numbers

**Description**

Clean Social Security ID numbers

**Usage**

```r
clean_ssn(number, keep_unknown = FALSE)
```

**Arguments**

- `number`: A character vector of Social Security identification numbers. No spaces or punctuation, only numbers.
- `keep_unknown`: Replace values for "unknown" with NA?

**Value**

A character vector with leading and trailing whitespace removed. If `keep_unknown` is FALSE, blanks and values representing unknown Social Security ID numbers are replaced with NA.
clean_telephone | *Clean telephone numbers*

**Description**

Clean telephone numbers

**Usage**

```
clean_telephone(number, keep_unknown = FALSE)
```

**Arguments**

- `number` | A character vector of telephone numbers. No spaces or punctuation, only numbers.
- `keep_unknown` | Replace values for "unknown" with NA?

**Value**

A character vector with leading and trailing whitespace removed. If `keep_unknown` is FALSE, blanks and values representing unknown numbers or patients without a number are replaced with NA.

---

clean_text | *Clean free-form text*

**Description**

Clean free-form text

**Usage**

```
clean_text(text, keep_unknown = FALSE)
```

**Arguments**

- `text` | A character vector of free text values.
- `keep_unknown` | Replace values for "unknown" with NA?

**Value**

A character vector with leading and trailing whitespace removed. If `keep_unknown` is FALSE, blank values are replaced with NA.
**field_levels**  
*List of possible values for a field*

**Description**

These lists gives the levels for each categorical or flag field from the NAACCR formats. It is intended to help researchers

**Usage**

```r
field_levels
field_levels_all
```

**Format**

A named list, where the names are for categorical fields or sentinel flags, and the values are the possible levels for each field.

**Details**

`field_levels` does not include levels representing "unknown." `field_levels_all` does include the "unknown" levels.

**naaccr_boolean**  
*Interpret NAACCR-style booleans*

**Description**

Interpret NAACCR-style booleans

**Usage**

```r
naaccr_boolean(flag, false_value = c("0", "1"))
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>flag</code></td>
<td>Character vector of flags.</td>
</tr>
<tr>
<td><code>false_value</code></td>
<td>The flag value to interpret as FALSE. If &quot;0&quot; (default), then &quot;1&quot; is interpreted as TRUE. If &quot;1&quot;, then &quot;2&quot; is interpreted as TRUE.</td>
</tr>
</tbody>
</table>

**Value**

A logical vector with the interpreted values of `flag`. Any original values not seen as TRUE or FALSE are converted to NA.
Examples

```r
x <- c("0", "1", "2", "9", NA)
naaccr_boolean(x)
naaccr_boolean(x, false_value = "1")
```

---

### naaccr_date

**Parse NAACCR-formatted dates**

**Description**

Parse NAACCR-formatted dates

**Usage**

```r
naaccr_date(date)
```

**Arguments**

- `date`  
  Character vector of dates in NAACCR format (`"YYYYMMDD"`).

**Value**

A Date vector. Any incomplete or invalid dates are converted to NA. The original strings can be retrieved with the `naaccr_encode` function.

**Examples**

```r
input <- c("20151031", "201408 ", "99999999")
d <- naaccr_date(input)
d
naaccr_encode(d, "dateOfDiagnosis")
```

---

### naaccr_datetime

**Parse NAACCR-formatted datetimes**

**Description**

Parse NAACCR-formatted datetimes

**Usage**

```r
naaccr_datetime(datetime)
```

**Arguments**

- `datetime`  
  Character vector of datetimes in NAACCR format (`"YYYYMMDDHHMMSS"`).

---
Value
A POSIXct vector. Any incomplete or invalid datetimes are converted to NA. The original strings
 can be retrieved with the `naaccr_encode` function.

Examples

```r
input <- c("20151031100856", "20140822 ", "99999999")
d <- naaccr_datetime(input)
d
naaccr_encode(d, "pathDateSpecCollect1")
```

---

**naaccr_encode**

**Format a value as a string according to the NAACCR format**

**Description**
Format a value as a string according to the NAACCR format

**Usage**

```r
naaccr_encode(x, field, flag = NULL, version = NULL, format = NULL)
```

**Arguments**

- `x` Vector of values.
- `field` Character string naming the field.
- `flag` Character vector of flags for the field. Only needed if the field contains sentinel
  values.
- `version` An integer specifying the NAACCR format version for parsing the records. Use
  this or `format`, not both. If both `version` and `format` are `NULL` (the default), the
  most recent version is used.
- `format` A `record_format` object for writing the records.

**Value**
Character vector of the values as they would be encoded in a NAACCR-formatted text file.

**See Also**

`split_sentineled`

**Examples**

```r
r <- naaccr_record(
    ageAtDiagnosis = c("089", "000", "200"),
    dateOfDiagnosis = c("20070402", "201709 ", " ")
)
r
mapply(FUN = naaccr_encode, x = r, field = names(r))
```
naaccr_factor

Replace NAACCR codes with understandable factors

Description

Replace NAACCR codes with understandable factors

Usage

naaccr_factor(x, field, keep_unknown = FALSE, ...)

Arguments

x
Vector (usually character) of codes.

field
String giving the XML name of the NAACCR field to code.

keep_unknown
Logical indicating whether values of "unknown" should be a level in the factor or NA.

... Additional arguments passed onto factor.

Value

A factor vector version of x. The levels are short descriptions instead of the basic NAACCR codes. Codes which stood for "unknown" with no further information are replaced with NA.

If field names a text or site-specific field, x will be returned unchanged with a warning.

Examples

naaccr_factor(c("20", "43", "99"), "radRegionalRxModality")
naaccr_factor(c("USA", "GER", "XEN"), "addrAtDxCountry")
# Default: NA for unknowns,
naaccr_factor(c("1", "8", "9"), "tumorGrowthPattern")
naaccr_factor(c("1", "8", "9"), "tumorGrowthPattern", keep_unknown = TRUE)

naaccr_format_12

Field definitions from all NAACCR format versions

Description

See record_format.
Usage

naaccr_format_12
naaccr_format_13
naaccr_format_14
naaccr_format_15
naaccr_format_16
naaccr_format_18

Format

An object of class record_format (inherits from data.table, data.frame) with 509 rows and 8 columns.

Details

A data.table object defining the fields for each version of NAACCR’s fixed-width record file format.

<table>
<thead>
<tr>
<th>naaccr_override</th>
<th>Interpret basic over-ride flags</th>
</tr>
</thead>
</table>

Description

Interpret basic over-ride flags

Usage

naaccr_override(flag)

Arguments

flag Character vector of over-ride flags. Its values should only include ”” (blank), ”1”, and possibly NA.

Value

A logical vector with the interpreted values of flag. The interpretation follows these rules: ”1” goes to TRUE (reviewed and confirmed as reported), ”” (blank) goes to FALSE (not reviewed or reviewed and corrected), and all other values go to NA.

Examples

naaccr_override(c("", "1", NA, "9"))
naaccr_record  
**Analysis-ready NAACCR records**

**Description**

Subclass of `data.frame` for doing analysis with NAACCR records.

**Usage**

```r
naaccr_record(..., keep_unknown = FALSE, version = NULL, format = NULL)
```

**Arguments**

- `...`: Arguments of the form `tag = value`, where `tag` is a valid NAACCR data item name and `value` is the vector of the item's values from the NAACCR format.
- `keep_unknown`: Logical indicating whether values of "unknown" should be a level in the factor or `NA`.
- `version`: An integer specifying the NAACCR format version for parsing the records. Use this or `format`, not both. If both `version` and `format` are `NULL` (default), the most recent NAACCR format will be used.
- `format`: A `record_format` object for parsing the records.

**Details**

`naaccr_record` creates a `data.frame` of cancer incidence records ready for analysis: columns are of appropriate classes, coded values are replaced with factors, and unknowns are replaced with `NA`.

**Value**

A `data.frame` with columns named using the NAACCR XML scheme.

---

read_naaccr_plain  
**Read NAACCR records**

**Description**

Read and parse cancer incidence records according to a NAACCR format. `read_naaccr` returns a data set suited for analysis in R, and `read_naaccr_plain` returns a data set with the unchanged record values.
read_naaccr_plain

**Usage**

```r
read_naaccr_plain(input, version = NULL, format = NULL,
                   keep_fields = NULL, skip = 0, nrows = Inf, buffersize = 10000,
                   encoding = getOption("encoding"))
```

```r
read_naaccr(input, version = NULL, format = NULL, keep_fields = NULL,
             keep_unknown = FALSE, skip = 0, nrows = Inf, buffersize = 10000,
             encoding = getOption("encoding"), ...)```

**Arguments**

- **input**: Either a string with a file name (containing no \n character), a `connection` object, or the text records themselves as a character vector.
- **version**: An integer specifying the NAACCR format version for parsing the records. Use this or `format`, not both. If both `version` and `format` are `NULL` (default), the most recent NAACCR format will be used.
- **format**: A `record_format` object for parsing the records.
- **keep_fields**: Character vector of XML field names to keep in the dataset. If `NULL` (default), all columns are kept.
- **skip**: An integer specifying the number of lines of the data file to skip before beginning to read data.
- **nrows**: A number specifying the maximum number of records to read. `Inf` (the default) means "all records."
- **buffersize**: Maximum number of lines to read at one time.
- **encoding**: String giving the input’s encoding. See the 'Encoding’ section of `file` in the `base` package.
- **keep_unknown**: Logical indicating whether values of "unknown" should be a level in the factor or NA.
- **...**: Additional arguments passed onto `as.naaccr_record`.

**Details**

Anyone who wants to analyze the records in R should use `read_naaccr`. In the returned `data.frame`, columns are of appropriate classes, coded values are replaced with factors, and unknowns are replaced with NA.

`read_naaccr_plain` is a "format strict" way to read incidence records. All values returned are the literal character values from the records. The only processing done is that leading and trailing whitespace is trimmed. This is useful if the values will be passed to other software that expects the plain NAACCR values.

**Value**

For `read_naaccr`, a `data.frame` of the records. The columns included depend on the NAACCR record format version. Columns are atomic vectors; there are too many to describe them all.

For `read_naaccr_plain`, a `data.frame` with the columns specified by `start_cols`, `end_cols`, and `col_names`. All columns are character vectors.
record_format

Define custom fields for NAACCR records

Description

Create a record_format object, which is used to read NAACCR records.

Usage

record_format(name, item, start_col, end_col, type, alignment = "left", padding = " ", name_literal = NULL)

as.record_format(x, ...)

Arguments

name Item name appropriate for a data.frame column name.
item NAACCR item number.
start_col First column of the field in a fixed-width record.
end_col Last column of the field in a fixed-width record.
type Name of the column class.
alignment Alignment of the field in fixed-width files. Either "left" (default) or "right".
padding Single-character strings to use for padding in fixed-width files.
nname_literal (Optional) Item name in plain language.
x Object to be coerced to a record_format, usually a data.frame or list.
... Other arguments passed to record_format.
**record_format**

**Details**

To define registry-specific fields in addition to the standard fields, create a `record_format` object for the registry-specific fields and combine it with one of the formats provided with the package using `rbind`.

**Value**

An object of class "record_format" which has the following columns:

- **name** (character) XML field name.
- **item** (integer) Field item number.
- **start_col** (integer) First column of the field in a fixed-width text file.
- **end_col** (integer) Last column of the field in a fixed-width text file.
- **type** (factor) R class for the column vector.
- **alignment** (factor) Alignment of the field’s values in a fixed-width text file.
- **padding** (character) String used for padding field values in a fixed-width text file.
- **name_literal** (character) Field name in plain language.

**Format Types**

The levels `type` can take, along with the functions used to process them when reading a file:

- **address** (`clean_address_number_and_street`) Street number and street name parts of an address.
- **age** (`clean_age`) Age in years.
- **boolean01** (`naaccr_boolean`, with `false_value = "0"`) True/false, where "0" means false and "1" means true.
- **boolean12** (`naaccr_boolean`, with `false_value = "1"`) True/false, where "1" means false and "2" means true.
- **census_block** (`clean_census_block`) Census Block ID number.
- **census_tract** (`clean_census_tract`) Census Tract ID number.
- **character** (`clean_text`) Miscellaneous text.
- **city** (`clean_address_city`) City name.
- **count** (`clean_count`) Integer count.
- **county** (`clean_county_fips`) County FIPS code.
- **Date** (`as.Date`, with `format = "%Y%m%d"`) NAACCR-formatted date (YYYYMMDD).
- **datetime** (`as.POSIXct`, with `format = "%Y%m%d%H%M%S"`) NAACCR-formatted datetime (YYYYMMDDHHMMSS).
- **facility** (`clean_facility_id`) Facility ID number.
- **icd_9** (`clean_icd_9_cm`) ICD-9-CM code.
- **icd_code** (`clean_icd_code`) ICD-9 or ICD-10 code.
- **integer** (`as.integer`) Miscellaneous whole number.
**numeric** (*as.numeric*) Miscellaneous decimal number.

**override** (*naaccr_override*) Field describing why another field’s value was over-ridden.

**physician** (*clean_physician_id*) Physician ID number.

**postal** (*clean_postal*) Postal code for an address (a.k.a. ZIP code in the United States).

**ssn** (*clean_ssn*) Social Security Number.

**telephone** (*clean_telephone*) 10-digit telephone number.

**Examples**

```r
my_fields <- record_format(
  name = c("foo", "bar"),
  item = c(2163, 1180),
  start_col = c(975, 1381),
  end_col = c(975, 1435),
  type = c("numeric", "facility")
)

my_format <- rbind(naaccr_format_16, my_fields)
```

---

**split_sentinled** 
*Separate a field’s continuous and sentinel values*

**Description**

Separate a sentineled field’s values into two vectors: one with the continuous data and one with the sentinel values.

**Usage**

```r
split_sentinled(x, field)
```

**Arguments**

- **x** Vector (usually character) of codes.
- **field** String giving the XML name of the NAACCR field to code.

**Value**

If `field` is a sentineled field, a `data.frame` with two columns. The first is a numeric version of the continuous values from `x`. Its name is the value of `field`. The second is a factor with levels representing the sentinel values. For all non-missing values in the numeric vector, the respective value in the factor is `NA`. If a value of `x` was not valid, the respective row will be `NA` for the continuous and flag values.

If `field` is not a sentineled field, a `data.frame` with just `x` is returned with a warning.
Examples

```r
node_codes <- c("10", "20", "90", "95", "99", NA)
s <- split_sentineled(node_codes, "regionalNodesPositive")
print(s)
s[is.na(s[["regionalNodesPositive"]]), "regionalNodesPositiveFlag"]
```

---

**split_sequence_number**  
Unpack tumor sequence number data

**Description**

Separate the multiple types of information in `sequenceNumberCentral` and `sequenceNumberHospital` into multiple columns.

**Usage**

```r
split_sequence_number(x)
```

**Arguments**

- `x`  
  Vector (usually character) of sequence number codes.

**Value**

A `data.frame` with three columns:

- `sequenceNumber` (integer) The number of the tumor in chronological sequence for the patient.
- `reportable` (logical) If TRUE, then the tumor is required to be reported by SEER/NPCR standards. If FALSE, it is either non-malignant or defined as reportable by the registry.
- `onlyTumor` (logical) If TRUE, this is the only known SEER/NPCR-reportable or the only known non-SEER/NPCR-reportable tumor for the patient.
- `sequenceFlag` (factor) Special flags, such as unknowns or changes in reporting requirements. Created using `split_sentineled`.

**See Also**

- `split_sentineled`
unknown_to_na

Replace labels for unknown with NA

Description

Replace labels for unknown with NA

Usage

unknown_to_na(x, ...)

## S3 method for class 'naaccr_record'
unknown_to_na(x, ...)

## S3 method for class 'factor'
unknown_to_na(x, field, ...)

Arguments

x

Either a factor created with naaccr_factor, or a naaccr_record object.

...

Further arguments passed to or from other methods.

field

String giving the XML name of the NAACCR field to code.

Value

If x was a factor, then the result is a vector with the values of x, except all levels which effectively mean "unknown" are replaced with NA. The returned factor won't have those in its levels, either.

If x is a naaccr_record object, then the result is the naaccr_record created by applying this function to all columns of x.

Examples

r <- naaccr_record(
  sex = c("1", "2", "9"),
  kras = c("8", "9", "3"),
  keep_unknown = TRUE
)
r
unknown_to_na(r["sex"], field = "sex")
unknown_to_na(r)
write_naaccr

---

**write_naaccr**

* Arrange the format for writing Write records in NAACCR format

---

**Description**

Write records from a `naaccr_record` object to a connection in fixed-width format, according to a specific version of the NAACCR format.

**Usage**

```r
write_naaccr(records, con, version = NULL, format = NULL)
```

**Arguments**

- `records` A `naaccr_record` object.
- `con` Either a character string naming a file or a `connection` open for writing.
- `version` An integer specifying the NAACCR format version for parsing the records. Use this or `format`, not both. If both `version` and `format` are `NULL` (the default), the most recent version is used.
- `format` A `record_format` object for writing the records.
Index

*Topic datasets
  field_levels, 11
  naaccr_format_12, 14

  as.Date, 19
  as.integer, 19
  as.naaccr_record, 2, 17
  as.numeric, 20
  as.POSIXct, 19
  as.record_format (record_format), 18
  clean_address_city, 3, 19
  clean_address_number_and_street, 4, 19
  clean_age, 4, 19
  clean_census_block, 5, 19
  clean_census_tract, 5, 19
  clean_count, 6, 19
  clean_county_fips, 6, 19
  clean_facility_id, 7, 19
  clean_icd_9_cm, 7, 19
  clean_icd_code, 8, 19
  clean_physician_id, 8, 20
  clean_postal, 9, 20
  clean_ssn, 9, 20
  clean_telephone, 10, 20
  clean_text, 10, 19
  connection, 17, 23

  factor, 14
  field_levels, 11
  field_levels_all (field_levels), 11
  file, 17

  naaccr_boolean, 11, 19
  naaccr_date, 12
  naaccr_datetime, 12
  naaccr_encode, 12, 13, 13
  naaccr_factor, 14, 22
  naaccr_format_12, 14
  naaccr_format_13 (naaccr_format_12), 14
  naaccr_format_14 (naaccr_format_12), 14
  naaccr_format_15 (naaccr_format_12), 14
  naaccr_format_16 (naaccr_format_12), 14
  naaccr_format_18 (naaccr_format_12), 14
  naaccr_override, 15, 20
  naaccr_record, 3, 16, 18, 22, 23

  read.fwf, 18
  read.table, 18
  read_naaccr (read_naaccr_plain), 16
  read_naaccr_plain, 16
  record_format, 3, 13, 14, 16, 17, 18, 23

  split_sentineled, 13, 20, 21
  split_sequence_number, 21

  unknown_to_na, 22

  write_naaccr, 23