Package ‘naaccr’

July 22, 2025

Type Package

Title Read Cancer Records in the NAACCR Format

Version 3.1.1

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

Depends R (>= 2.10)

Imports data.table, stringi, utils, XML

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

License MIT + file LICENSE

Copyright file COPYRIGHTS

Encoding UTF-8

LazyData true

NeedsCompilation no

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 [cph],
United States Bureau of the Census [cph],
United States National Program of Cancer Registries [cph]

Repository CRAN

Date/Publication 2024-09-20 14:20:05 UTC

Contents

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

Description

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

Usage

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

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

## 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

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

Arguments

`city`	A character vector of city 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_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

```
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

```
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*

Description

Replaces any values of all 9's with NA (if keep_unknown is TRUE) and converts the rest to integers.

Usage

```
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*

Description

Clean county FIPS codes

Usage

```
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` *Clean facility identification numbers*

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` *Clean ICD-9-CM codes*

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

```
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

```
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

```
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

```
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

An 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

```
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.

An object of class `list` of length 340.

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

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

Arguments

`flag`	Character vector of flags.
`false_value`	The flag value to interpret as `FALSE`. If `"0"` (default), then `"1"` is interpreted as `TRUE`. If `"1"`, then `"2"` is interpreted as `TRUE`.

Value

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

Examples

```
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

```
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

```
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

```
naaccr_datetime(datetime, tz = "")
```

Arguments

`datetime` Character vector of datetimes in HL7 OBR-7 format (`"YYYYMMDDHHMMSS"`) or the ISO 8601 format for datetimes accurate to the second (`YYYY-MM-DDThh:mm:ss+zz:zz`). Values containing a hyphen (`"-"`) will be assumed to follow ISO 8601, and other values will be assumed to follow HL7 OBR-7.

`tz` time zone specification to be used for the conversion, *if one is required*. System-specific (see time zones), but `""` is the current time zone, and `"GMT"` is UTC (Universal Time, Coordinated). Invalid values are most commonly treated as UTC, on some platforms with a warning.

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

```
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

```
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 <- 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_formats *Field definitions from all NAACCR format versions*

Description

See `record_format`.

Usage

```
naaccr_formats

naaccr_format_12

naaccr_format_13

naaccr_format_14

naaccr_format_15

naaccr_format_16

naaccr_format_18

naaccr_format_21

naaccr_format_22

naaccr_format_23

naaccr_format_24

naaccr_format_25
```

Format

An object of class `list` of length 22.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 509 rows and 12 columns.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 529 rows and 12 columns.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 548 rows and 12 columns.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 555 rows and 12 columns.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 587 rows and 12 columns.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 791 rows and 12 columns.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 800 rows and 12 columns.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 810 rows and 12 columns.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 782 rows and 12 columns.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 783 rows and 12 columns.

An object of class `record_format` (inherits from `data.table`, `data.frame`) with 780 rows and 12 columns.

Details

Each `naaccr_format_XX` object is a `data.table` defining the fields for each version of NAACCR's record file format. `naaccr_formats` is a list of these record formats, with each name being the two- or three-digit code for the format.

`naaccr_override` *Interpret basic over-ride flags*

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

```
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 `naaccr_record` with columns named using the NAACCR XML scheme. It inherits from `data.frame`.

`parse_geocoding_quality_codes` *Parse the 14 values from the* `geocodingQualityCodeDetail` *field*

Description

Parse the 14 values from the `geocodingQualityCodeDetail` field

Usage

```
parse_geocoding_quality_codes(value)
```

Arguments

value Character vector of values from the `geocodingQualityCodeDetail` field. Each value should be 14 characters long or `NA`.

Value

A `data.frame` with the following columns:

`geocodingQualityInputType` (`factor`) Type of address given to geocoder. Has the following levels: `"full address"`, `"street only"`, `"number, no street"`, `"city only"`, `"zip and city"`, `"zip only"`, `"error"`

`geocodingQualityStreetType` (`factor`) Type of street for address. Has the following levels: `"street"`, `"PO box"`, `"rural route"`, `"highway contract route"`, `"star route"`, `"error"`

`geocodingQualityStreet` (`factor`) Quality of match for the street name. Has the following levels: `"100% match"`, `"soundex match"`, `"street name different"`, `"missing street name"`, `"error"`

`geocodingQualityZip` (`factor`) Quality of match for the ZIP code. Has the following levels: `"100% match"`, `"5th digit different"`, `"4th digit different"`, `"3rd digit different"`, `"2nd digit different"`, `"1st digit different"`, `"more than one digit different"`, `"invalid ZIP"`, `"error"`

`geocodingQualityCity` (`factor`) Quality of match for the city name. Has the following levels: `"100% match"`, `"alias match"`, `"soundex match"`, `"no match"`, `"error"`

`geocodingQualityCityRefs` (`factor`) Number of city reference data sets that don't match the geocoding result. Has the following levels: `"all match"`, `"1 reference unmatched"`, `"2 to 4 references unmatched"`, `"5 or more references unmatched"`, `"no references matched"`, `"error"`

`geocodingQualityDirectionals` (`factor`) Whether the street directionals are present in the input and feature data sets. Has the following levels: `"all match"`, `"missing feature pre and post directionals"`, `"missing input pre and post directionals"`, `"both pre and post directionals do not match"`, `"feature missing post directional"`, `"input missing post directional"`, `"post directionals do not match"`, `"missing feature pre directional"`, `"missing feature pre directional and input post directional"`, `"missing feature pre directional and post directionals do not match"`, `"missing input pre directional"`, `"missing input pre directional and missing feature post directional"`, `"missing input pre directional and post directionals do not match"`, `"pre directionals do not match"`, `"pre directionals do not match and missing feature post directional"`, `"pre directionals do not match and missing input post directional"`

`geocodingQualityQualifiers` (`factor`) Whether the address qualifiers are present in the input and feature data sets. Has the following levels: `"all match"`, `"missing feature pre and post qualifiers"`, `"missing input pre and post qualifiers"`, `"both pre and post qualifiers do not match"`, `"feature missing post qualifier"`, `"input missing post qualifier"`, `"post qualifiers do not match"`, `"missing feature pre qualifier"`, `"missing feature pre qualifier and input post qualifier"`, `"missing feature pre qualifier and post qualifiers do not match"`, `"missing input pre qualifier"`, `"missing input pre qualifier and missing feature post qualifier"`, `"missing input pre qualifier and post qualifiers do not match"`, `"pre qualifiers do not match"`, `"pre qualifiers do not match and missing feature post qualifier"`, `"pre qualifiers do not match and missing input post qualifier"` geocodingQualityDistance (factor) Average distance between the possible matched parcels and their respective possible matched streets. Has the following levels: "< 10m", "10m-100m", "100m-500m", "500m-1km", "1km-5km", "> 5km", "error"

geocodingQualityOutliers (factor) Distribution of distances between the possible matched parcels and their respective possible matched streets. Has the following levels: "100% within 10m", "60% within 10m and 40% within 100m", "60% within 10m and 40% within 500m", "60% within 10m and 40% within 1km", "60% within 10m and 40% within 5km", "60% within 10m and at least 1 over 5km exists", "30% within 10m and 70% within 100m", "30% within 10m and 70% within 500m", "30% within 10m and 70% within 1km", "30% within 10m and 70% within 5km", "30% within 10m and at least 1 over 5km exists", "error"

geocodingQualityCensusBlockGroups (factor) Consistency of geocoded result against Census Block Group references. Has the following levels: "all match", "at least one reference different", "no Census data", "error"

geocodingQualityCensusTracts (factor) Consistency of geocoded result against Census Tract references. Has the following levels: "all match", "at least one reference different", "no Census data", "error"

geocodingQualityCensusCounties (factor) Consistency of geocoded result against Census County references. Has the following levels: "all match", "at least one reference different", "no Census data", "error"

geocodingQualityRefMatchCount (integer) Number of reference data sets matched by geocoding result.

read_naaccr_plain *Read NAACCR records from a file*

Description

Read and parse cancer incidence records according to a NAACCR format from either fixed-width files (read_naaccr and read_naaccr_plain) or XML documents (read_naaccr_xml and read_naaccr_xml_plain).

Usage

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

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

read_naaccr_xml_plain(
  input,
  version = NULL,
  format = NULL,
  keep_fields = NULL,
  as_text = FALSE,
  encoding = getOption("encoding")
)

read_naaccr_xml(
  input,
  version = NULL,
  format = NULL,
  keep_fields = NULL,
  keep_unknown = FALSE,
  as_text = FALSE,
  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. For `read_naaccr_xml` and `read_naaccr_xml_plain`, this is a *backup* encoding. If the XML document includes an encoding specification, that will be used. Otherwise, `encoding` will be used.
`keep_unknown`	Logical indicating whether values of "unknown" should be a level in the factor or `NA`.
`...`	Additional arguments passed onto `as.naaccr_record`.
`as_text`	Logical indicating (if `TRUE`) that `input` is a character string containing XML or (if `FALSE`) it is the path to a file with XML content.

Details

`read_naaccr` and `read_naaccr_xml` return data sets suited for analysis in R. `read_naaccr_plain` and `read_naaccr_xml_plain` return data sets with the unchanged record values.

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

`read_naaccr_plain` and `read_naaccr_xml_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.

For `read_naaccr_plain` and `read_naaccr`, if the `version` and `format` arguments are left `NULL`, the default format is version 18. This was the last format to be used for fixed-width files.

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` based on the `record_format` specified by either the `version` or `format` argument. The names of the columns will be those in the format's `name` column. All columns are character vectors.

Note

Some of the parameter text was shamelessly copied from the `read.table` and `read.fwf` help pages.

References

North American Association of Central Cancer Registries (October 2018). Standards for Cancer Registries Volume II: Data Standards and Data Dictionary. Twenty first edition. https://apps.naaccr.org/data-dictionary/.

North American Association of Central Cancer Registries (April 2019). NAACCR XML Data Exchange Standard. Version 1.4. https://www.naaccr.org/xml-data-exchange-standard/.

See Also

`naaccr_record`

Examples

```
# This file has synthetic abstract records
incfile <- system.file(
  "extdata", "synthetic-naaccr-18-abstract.txt",
  package = "naaccr"
)
fields <- c("ageAtDiagnosis", "sex", "sequenceNumberCentral")
read_naaccr(incfile, version = 18, keep_fields = fields)
recs <- read_naaccr_plain(incfile, version = 18, keep_fields = fields)
recs
# Note sequenceNumberCentral has been split in two: a number and a flag
summary(recs[["sequenceNumberCentral"]])
summary(recs[["sequenceNumberCentralFlag"]])
```

---

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 = NA_integer_,
  end_col = NA_integer_,
  type = "character",
  alignment = "left",
  padding = " ",
  parent = "Tumor",
  cleaner = list(NULL),
  unknown_finder = list(NULL),
  name_literal = NA_character_,
  width = NA_integer_
)

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	*Deprecated: Use the width parameter instead.* 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.

`parent` Name of the parent node to include this field under when writing to an XML file. Values can be `"NaaccrData"`, `"Patient"`, `"Tumor"`, or `NA` (default). Fields with `NA` for parent won't be included in an XML file.

`cleaner` (Optional) List of functions to handle special cases of cleaning field data (e.g., convert all values to uppercase). Values of `NULL` (the default) mean the default cleaning function for the `type` is used. The value can also be the name of a function to retrieve with `getFunction`. See Details.

`unknown_finder` (Optional) List of functions to detect when codes mean the actual values are unknown or not applicable. Values of `NULL` (the default) mean the default unknown finding function for the `type` is used. The value can also be the name of a function to retrieve with `getFunction`. See Details.

`name_literal` (Optional) Item name in plain language.

`width` (Optional) Item width in characters.

`x` Object to be coerced to a `record_format`, usually a `data.frame` or `list`.

`...` Other arguments passed to `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. If `NA`, the field will not be read from or written to fixed-width files. They will included in XML files.

`end_col (integer)` (*Deprecated: Use `width` instead.*) Last column of the field in a fixed-width text file. If `NA`, the field will not be read from or written to fixed-width files. This is the norm for fields only found in XML formats.

`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.

`parent (factor)` Parent XML node for the field. One of `"NaaccrData"`, `"Patient"`, or `"Tumor"`.

`cleaner (list` of `function` objects) Function to prepare the field's values for analysis. Values of `NULL` will use the standard cleaner functions for the `type` (see below).

`unknown_finder (list` of `function` objects) Function to detect codes meaning the actual values are missing or unknown for the field.

name_literal (character) Field name in plain language.

width (integer) Character width of the field values. Mostly meant for reading and writing flat files.

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

```
my_fields <- record_format(
  name      = c("foo", "bar", "baz"),
  item      = c(2163, 1180, 1181),
  start_col = c(975, 1381, NA),
  width     = c(1, 55, 4),
  type      = c("numeric", "facility", "character"),
  parent    = c("Patient", "Tumor", "Tumor"),
  cleaner   = list(NULL, NULL, trimws)
)
my_format <- rbind(naaccr_format_16, my_fields)
```

split_sentineled *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

```
split_sentineled(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

```
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

```
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 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

```
write_naaccr(records, con, version = NULL, format = NULL, encoding = "UTF-8")
```

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.
encoding	String specifying the character encoding for the output file.

write_naaccr_xml *Write records to a NAACCR-formatted XML file*

Description

Write records to a NAACCR-formatted XML file

Usage

```
write_naaccr_xml(
  records,
  con,
  version = NULL,
  format = NULL,
  base_dictionary = NULL,
  user_dictionary = NULL,
  encoding = "UTF-8"
)
```

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.

base_dictionary
: URI for the dictionary defining the NAACCR data items. If this is NULL and either version is not NULL or format is one of the standard NAACCR formats, then the URI from NAACCR's website for that version's dictionary will be used.

user_dictionary
: URI for the dictionary defining the user-specified data items. If NULL (default), it won't be included in the XML.

encoding
: String specifying the character encoding for the output file.

Index