

Package: crosswalkr (via r-universe)

November 6, 2024

Title Rename and Encode Data Frames Using External Crosswalk Files

Version 0.3.0

Description A pair of functions for renaming and encoding data frames using external crosswalk files. It is especially useful when constructing master data sets from multiple smaller data sets that do not name or encode variables consistently across files. Based on similar commands in 'Stata'.

URL <https://github.com/btskinner/crosswalkr>

BugReports <https://github.com/btskinner/crosswalkr/issues>

Depends R (>= 3.3.0)

Imports haven, labelled, methods, readr, readxl, tibble

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Suggests testthat, knitr, rmarkdown, dplyr

VignetteBuilder knitr

Config/pak/sysreqs make libicu-dev libx11-dev zlib1g-dev

Repository <https://btskinner.r-universe.dev>

RemoteUrl <https://github.com/btskinner/crosswalkr>

RemoteRef HEAD

RemoteSha c447932ae38323ee679148c8f8f204aa2eda9a17

Contents

encodefrom	2
renamefrom	3
stcrosswalk	5
sttercrosswalk	6
Index	7

encodefrom	<i>Encode data frame column using external crosswalk file.</i>
------------	--

Description

Encode data frame column using external crosswalk file.

Usage

```
encodefrom(
  .data,
  var,
  cw_file,
  raw,
  clean,
  label,
  delimiter = NULL,
  sheet = NULL,
  case_ignore = TRUE,
  ignore_tibble = FALSE
)
```

```
encodefrom_(
  .data,
  var,
  cw_file,
  raw,
  clean,
  label,
  delimiter = NULL,
  sheet = NULL,
  case_ignore = TRUE,
  ignore_tibble = FALSE
)
```

Arguments

<code>.data</code>	Data frame or <code>tbl_df</code>
<code>var</code>	Column name of vector to be encoded
<code>cw_file</code>	Either data frame object or string with path to external crosswalk file, including path, which has columns representing <code>raw</code> (current) vector values, <code>clean</code> (new) vector values, and <code>label</code> s for values. Values in <code>raw</code> and <code>clean</code> columns must be unique (1:1 match) or an error will be thrown. Acceptable file types include: delimited (<code>.csv</code> , <code>.tsv</code> , or other), R (<code>.rda</code> , <code>.rdata</code> , <code>.rds</code>), or Stata (<code>.dta</code>).
<code>raw</code>	Name of column in <code>cw_file</code> that contains values in current vector.
<code>clean</code>	Name of column in <code>cw_file</code> that contains new values for vector.

label	Name of column in cw_file with labels for new values.
delimiter	String delimiter used to parse cw_file. Only necessary if using a delimited file that isn't a comma-separated or tab-separated file (guessed by function based on file ending).
sheet	Specify sheet if cw_file is an Excel file and required sheet isn't the first one.
case_ignore	Ignore case when matching current (raw) vector name with new (clean) column name.
ignore_tibble	Ignore .data status as tbl_df and return vector as a factor rather than labelled vector.

Value

Vector that is either a factor or labelled, depending on data input and options

Functions

- `encodefrom_()`: Standard evaluation version of `encodefrom` (var, raw, clean, and label must be strings when using this version)

Examples

```
df <- data.frame(state = c('Kentucky', 'Tennessee', 'Virginia'),
                 stfips = c(21, 47, 51),
                 cenregnm = c('South', 'South', 'South'))

df_tbl <- tibble::as_tibble(df)

cw <- get(data(stcrosswalk))

df$state2 <- encodefrom(df, state, cw, stname, stfips, stabbr)
df_tbl$state2 <- encodefrom(df_tbl, state, cw, stname, stfips, stabbr)
df_tbl$state3 <- encodefrom(df_tbl, state, cw, stname, stfips, stabbr,
                           ignore_tibble = TRUE)

haven::as_factor(df_tbl)
haven::zap_labels(df_tbl)
```

renamefrom

Rename data frame columns using external crosswalk file.

Description

Rename data frame columns using external crosswalk file.

Usage

```

renamefrom(
  .data,
  cw_file,
  raw,
  clean,
  label = NULL,
  delimiter = NULL,
  sheet = NULL,
  drop_extra = TRUE,
  case_ignore = TRUE,
  keep_label = FALSE,
  name_label = FALSE
)

```

```

renamefrom_(
  .data,
  cw_file,
  raw,
  clean,
  label = NULL,
  delimiter = NULL,
  sheet = NULL,
  drop_extra = TRUE,
  case_ignore = TRUE,
  keep_label = FALSE,
  name_label = FALSE
)

```

Arguments

<code>.data</code>	Data frame or <code>tbl_df</code>
<code>cw_file</code>	Either data frame object or string with path to external crosswalk file, which has columns representing <code>raw</code> (current) column names, <code>clean</code> (new) column names, and labels (optional). Values in <code>raw</code> and <code>clean</code> columns must be unique (1:1 match) or an error will be thrown. Acceptable file types include: delimited (.csv, .tsv, or other), R (.rda, .rdata, .rds), or Stata (.dta).
<code>raw</code>	Name of column in <code>cw_file</code> that contains column names of current data frame.
<code>clean</code>	Name of column in <code>cw_file</code> that contains new column names.
<code>label</code>	Name of column in <code>cw_file</code> with labels for columns.
<code>delimiter</code>	String delimiter used to parse <code>cw_file</code> . Only necessary if using a delimited file that isn't a comma-separated or tab-separated file (guessed by function based on file ending).
<code>sheet</code>	Specify sheet if <code>cw_file</code> is an Excel file and required sheet isn't the first one.
<code>drop_extra</code>	Drop extra columns in current data frame if they are not matched in <code>cw_file</code> .

case_ignore	Ignore case when matching current (raw) column names with new (clean) column names.
keep_label	Keep current label, if any, on data frame columns that aren't matched in cw_file. Default FALSE means that unmatched columns have any existing labels set to NULL.
name_label	Use old (raw) column name as new (clean) column name label. Cannot be used if label option is set.

Value

Data frame or tbl_df with new column names and labels.

Functions

- `renamefrom_()`: Standard evaluation version of `renamefrom` (raw, clean, and label must be strings when using this version)

Examples

```
df <- data.frame(state = c('Kentucky', 'Tennessee', 'Virginia'),
                 fips = c(21, 47, 51),
                 region = c('South', 'South', 'South'))

cw <- data.frame(old_name = c('state', 'fips'),
                 new_name = c('stname', 'stfips'),
                 label = c('Full state name', 'FIPS code'))

df1 <- renamefrom(df, cw, old_name, new_name, label)
df2 <- renamefrom(df, cw, old_name, new_name, name_label = TRUE)
df3 <- renamefrom(df, cw, old_name, new_name, drop_extra = FALSE)
```

stcrosswalk

State crosswalk data set.

Description

An example state crosswalk. Includes information for all states plus the District of Columbia.

Usage

```
stcrosswalk
```

Format

A data frame with 51 rows and 7 variables:

stfips Two-digit state FIPS codes
stabbr Two-letter state abbreviation
stname Full state name
cenreg Census region number
cenregnm Census region name
cendiv Census division number
cendivnm Census division name

sttercrosswalk	<i>State and territory crosswalk data set.</i>
----------------	--

Description

An example state and territory crosswalk. Includes information for all states plus the District of Columbia plus territories.

Usage

```
sttercrosswalk
```

Format

A data frame with 69 rows and 10 variables:

stfips Two-digit FIPS codes
stabbr Two-letter abbreviation
stname Full name
cenreg Census region number
cenregnm Census region name
cendiv Census division number
cendivnm Census division name
is_state Indicator for status as state
is_state_dc Indicator for status as state or DC
status 1 := Under U.S. sovereignty; 2 := Minor Outlying Islands; 3 := Independent nation under Compact of Free Association with U.S.; 4 := Individual Minor Outlying Islands (within status 2)

Index

* datasets

stcrosswalk, 5

sttercrosswalk, 6

encodefrom, 2, 3

encodefrom_ (encodefrom), 2

renamefrom, 3, 5

renamefrom_ (renamefrom), 3

stcrosswalk, 5

sttercrosswalk, 6