

# Package ‘crosswalkr’

November 23, 2018

**Title** Rename and Encode Data Frames Using External Crosswalk Files

**Version** 0.1.1

**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** dplyr, haven, labelled, methods, readr, readxl, tibble

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Benjamin Skinner [aut, cre]

**Maintainer** Benjamin Skinner <btskinner@virginia.edu>

**Repository** CRAN

**Date/Publication** 2017-10-17 10:45:19 UTC

## R topics documented:

crosswalkr . . . . .	2
encodefrom . . . . .	2
renamefrom . . . . .	3
stcrosswalk . . . . .	4

Index . . . . .	6
-----------------	---

---

crosswalkr	<i>crosswalkr: Rename and Encode Data Frames Using External Crosswalk Files</i>
------------	---

---

### Description

A pair of functions for renaming and encoding data frames using external crosswalk files. Especially useful when constructing master data sets from multiple smaller data sets that do not name or encode variables consistently across files. Based on `-renamefrom-` and `-encodefrom-` Stata commands.

---

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>	String of external crosswalk file, including path, which has columns representing <code>raw</code> (current) vector values, <code>clean</code> (new) vector values, and <code>labels</code> 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 new values for vector.
<code>label</code>	Name of column in <code>cw_file</code> with labels for new values.
<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>case_ignore</code>	Ignore case when matching current ( <code>raw</code> ) vector name with new ( <code>clean</code> ) column name.
<code>ignore_tibble</code>	Ignore <code>.data</code> status as <code>tbl_df</code> 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_data_frame(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	<i>Rename data frame columns using external crosswalk file.</i>
------------	---

---

## 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>	String of external crosswalk file, including path, 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> .
<code>case_ignore</code>	Ignore case when matching current ( <code>raw</code> ) column names with new ( <code>clean</code> ) column names.
<code>keep_label</code>	Keep current label, if any, on data frame columns that aren't matched in <code>cw_file</code> . Default <code>FALSE</code> means that unmatched columns have any existing labels set to <code>NULL</code> .
<code>name_label</code>	Use old ( <code>raw</code> ) column name as new ( <code>clean</code> ) column name label. Cannot be used if <code>label</code> 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

# Index

## \*Topic **datasets**

stcrosswalk, [4](#)

crosswalkr, [2](#)

crosswalkr-package (*crosswalkr*), [2](#)

encodefrom, [2](#), [3](#)

encodefrom\_ (*encodefrom*), [2](#)

renamefrom, [3](#), [4](#)

renamefrom\_ (*renamefrom*), [3](#)

stcrosswalk, [4](#)