Package 'augur'

October 23, 2017

Title augur is a seer for SEER-Medicare data

Version 0.0.0.9001

Date 2017-10-14

Description SEER-Medicare data arrive in an array of files of various formats. SAS Input files are provided with SEER-Medicare data; this package makes use of those SAS in-files to read corresponding SEER or Medicare fixed-width ASCII files.

Depends R (>= 3.3.3)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports readr, dplyr, purrr, rlang

Suggests testthat, knitr, rmarkdown

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Mustafa Ascha [aut, cre]

Maintainer Mustafa Ascha <mustafa.ascha@gmail.com>

R topics documented:

`format_medi` *Parse a SAS FWF format file*

Description

Parse a SAS FWF format file

Usage

```
format_medi(sas_input_file, formatfile.type = "default")
```

Arguments

`sas_input_file`	The SAS file describing a FWF claims file.
`formatfile.type`	The SEER-Medicare infile type being formatted.

`get_files_df` *Returns a dataframe listing all ASCII data file filepaths listed with their respective SAS infile filepaths, and the "type" of medicare file.*

Description

Returns a dataframe listing all ASCII data file filepaths listed with their respective SAS infile filepaths, and the "type" of medicare file.

Usage

```
get_files_df(infiles_directory, data_files_directory)
```

Arguments

`infiles_directory`	The directory containing SAS infiles provided by SEER-Medicare
`data_files_directory`	The directory containing ASCII FWF claims files.

get_filetype *Get the names of files from their filepaths, useful because medicare claims are presented with one SAS input file for each type of claims file.*

Description

Get the names of files from their filepaths, useful because medicare claims are presented with one SAS input file for each type of claims file.

Usage

```
get_filetype(filepath)
```

Arguments

`filepath`	This is the path to a SAS infile text document.

get_file_df *Returns a dataframe listing all ASCII data file filepaths of a specific type listed with its corresponding SAS infile filepaths, and the "type" of medicare file.*

Description

Returns a dataframe listing all ASCII data file filepaths of a specific type listed with its corresponding SAS infile filepaths, and the "type" of medicare file.

Usage

```
get_file_df(infile_path, data_files_directory)
```

Arguments

`infile_path`	The directory containing SAS infiles provided by SEER-Medicare
`data_files_directory`	The directory containing ASCII FWF claims files.

`grep_raw` *Return the matching regex*

Description

I suspect there's a better way to do this.

Usage

```
grep_raw(x, p)
```

Arguments

`x`	The thing to examine
`p`	The pattern to grep

`infile_names` *Make a names conversion key from an infile*

Description

Make a names conversion key from an infile

Usage

```
infile_names(to_read)
```

Arguments

`to_read`	The SAS infile provided by IMS.

`list_txt_files` *Retrieve file list and types from directory containing claims data*

Description

Retrieve file list and types from directory containing claims data

Usage

```
list_txt_files(files_path)
```

Arguments

`files_path`	The directory containing claims files of interest.

merge_scans *Merge data.frames of logical columns from scanning functions*

Description

This function will merge two of the output from ‘split_and_scan‘

Usage

```
merge_scans(scan_1, scan_2, ctp)
```

Arguments

`scan_1`	The first scan to merge
`scan_2`	The second scan to merge
`ctp`	Codes to pull, these are essential for names

pedsf_utils *PEDSF reading and recoding dataframes*

Description

This is a list of three dataframes that are useful for PEDSF data reading and recoding. The most important of these three dataframes is ‘pedsf_format‘, whose content was pulled from the SEER-Medicare PDF PEDSF documentation. The ‘pedsf_recodes‘ dataframe is, of course, useful in conjunction with the ‘recode_variable‘ or ‘recode_pedsf‘ functions. Finally, the ‘pedsf_names‘ dataframe is useful for converting between full versus short variable names.

Usage

```
pedsf_utils
```

Format

Three dataframes, each containing useful recoding or reading-relevant data.

pedsf_recodes 1090 rows by 3 columns, this is PEDSF recode data extracted from an NCI HDRP PDF.

- Var- The name of the PEDSF variable to be recoded
- Code- The coded representation of variables that are included as part of PEDSF.
- Meaning- The value that is encoded by ‘Code‘ in the PEDSF variable ‘Var‘.

pedsf_format A ‘data.frame‘ consisting of 245 rows of PEDSF file format data, produced using a SEER-Medicare PEDSF SAS infile. These are useful for reading PEDSF data.

- positions- In the PEDSF ASCII file, this is the starting character position for the variable shown in ‘vars‘.
- vars- The name of a variable that will be read using a starting position of ‘positions‘ and character width of ‘widths‘ and variable type ‘col_types‘.

pedsf_names 134 full NAACCR variable names alongside their corresponding SAS variable names, again from the i SEER-Medicare PEDSF SAS infile.

Source

https://healthcaredelivery.cancer.gov/seermedicare/program/inputs/pedsf.txt

https://healthcaredelivery.cancer.gov/seermedicare/aboutdata/pedsf_attachment_a.pdf

---

radio_chemo_codes *Dataframes to support recoding SEER and Medicare diagnosis and procedure data*

Description

Each dataframe contains the key-value pairs necessary to recode a variable so that it is interpretable, rather than its coded numeric value. These data come from several sources, the URL for each of which is listed here. For more information on each dataframe listed here, see its respective documentation.

Usage

```
radio_chemo_codes
```

Format

Four data frames of key-value pairs that can be used with ‘recode_variable‘ and appropriate claims files to translate codes into human-readable corresponding drug or procedure names.

chemo_drugs 8703 chemotherapy drugs over 9 recode variables

chemo_procs 569 chemo procedures over 8 recode variables

radio_procs 266 radiotherapy procedures over 6 variables

radio_diags 5 diagnoses described by 5 variables

Source

https://crn.cancer.gov/resources/ctcodes-drugs.csv

https://crn.cancer.gov/resources/ctcodes-procedures.csv

https://crn.cancer.gov/resources/RadiationTherapyCodes_Procedures.csv

https://crn.cancer.gov/resources/RadiationTherapyCodes_Diagnoses.csv

See Also

recode_variable

`read_medicare` *Read a medicare claims file*

Description

Read a medicare claims file

Usage

```
read_medicare(data_file, format_file, ...)
```

Arguments

`data_file` The claims file to be read

`format_file` The SAS infile provided by SEER-Medicare

`...` Arguments forwarded to ‘read_fwf‘ from ‘readr‘

`read_this_type` *Read all of a specific type of claims file*

Description

This function will look in ‘data_files_path‘ for a specific type of SEER-Medicare file, and read all files of that type that are contained therein.

Usage

```
read_this_type(infile_path, data_path, fileformat.type = "default",
  limit = 10)
```

Arguments

`infile_path` The SAS input file to be used for reading ASCII FWF files.

`data_path` The directory containing claims files of interest.

`fileformat.type`
The type of infile to be read, and the type of SEER-Medicare file that will be read from the ‘data_files_path‘. This must be one of "dme", "ccflag", "nch", "medpar", "outsaf", "pdesaf", "pedsf", "tract", or "zipcode". The default value uses ‘get_filetype‘ to regex the filetype out of its filename, though this is not a reliable approach and is discouraged.

`limit` The number of records to read from a file.

recode_variable *Recode PEDSF variables*

Description

This function joins variables by their value to a dataframe decoder

Recode PEDSF variables

Usage

```
recode_variable(df, which_variable, which_key, max.dist = 0,
  var_as_is = FALSE)

recode_pedsf(df, which_variable, ...)
```

Arguments

df The ‘data.frame‘ containing PEDSF data to recode.

which_variable The name of the variable to be recoded, unquoted.

which_key The dataframe containing key-value pairs for recoding.

max.dist Approximate-ness of the ‘df‘ variable name being matched to the PEDSF recode dataframe variable name

var_as_is A logicalswitch indicating whether to accept ‘which_variable‘ as-is and containing its own values versus doing an ‘enquo‘ on ‘which_variable‘

... One or more of the arguments passed to ‘recode_variable‘, among ‘max.dist‘ for variable names matching and ‘for_real‘ as a logical switch indicating whether to ‘enquo‘ ‘which_variable‘ or instead accept ‘which_variable‘ as-is.

See Also

recode_dfs

scan_nch_dx *For each code in a character array representing codes from a claims row, return TRUE/FALSE if that row contains a code of interest*

Description

For each code in a character array representing codes from a claims row, return TRUE/FALSE if that row contains a code of interest

Usage

```
scan_nch_dx(df, codes_to_pull)
```

Arguments

df An NCH dataframe made by ‘read_medicare‘

codes_to_pull ‘scan_nch‘ will test for the presence of these codes

Value

data.frame of TRUE/FALSE for each code of interest

`scan_op_dx` *Function to retrieve outpatient diagnosis code matches*

Description

Function to retrieve outpatient diagnosis code matches

Usage

```
scan_op_dx(df, codes_to_pull)
```

Arguments

`df`	An outpatient dataframe made by ‘read_medicare‘
`codes_to_pull`	‘scan_op_dx‘ will test for the presence of these codes

Value

‘data.frame‘ of TRUE/FALSE columns for each code of interest

`scan_pde_dx` *Function to retrieve outpatient diagnosis code matches*

Description

Function to retrieve outpatient diagnosis code matches

Usage

```
scan_pde_dx(df, codes_to_pull)
```

Arguments

`df`	An outpatient dataframe made by ‘read_medicare‘
`codes_to_pull`	‘scan_op_dx‘ will test for the presence of these codes

Value

‘data.frame‘ of TRUE/FALSE columns for each code of interest

split_and_scan *Returns TRUE if a diagnosis code is present, as a dataframe for each code*

Description

This is a version of ‘split_and_scan‘ based on ‘purrr‘ rather than ‘base‘

Usage

```
split_and_scan(data_codes, ctp)
```

Arguments

`data_codes`	Actual codes, whitespace between each code
`ctp`	Codes against which to check each diagnosis, codes to pull

Details

Data should be provided as it is returned by ‘read_medicare‘. That is, it should be a character array whose contents are of the form: "00000 00000 00000 00000 00000", in fixed-width format.

Value

data.frame of TRUE/FALSE for each code, where TRUE indicates that the code is present in an element of the data.

switch_names *Replace names with those from an infile*

Description

Replace names with those from an infile

Usage

```
switch_names(df, infile, to_short = FALSE)
```

Arguments

`df`	The dataframe whose names to change
`infile`	The infile to use for names conversion
`to_short`	Switch to use for converting back to short names

Index