Package: medicare (via r-universe)

September 15, 2024

Type Package

Title Tools for Obtaining and Cleaning Medicare Public Use Files

Version 0.2.1

Description Publicly available data from Medicare frequently requires
extensive initial effort to extract desired variables and merge
them; this package formalizes the techniques I've found work
best. More information on the Medicare program, as well as
guidance for the publicly available data this package targets,
can be found on CMS's website covering publicly available data.
See
<https://www.cms.gov/Research-Statistics-Data-and-Systems/
Research-Statistics-Data-and-Systems.html>.

License MIT + file LICENSE

LazyData TRUE

Depends R (>= 2.10)

URL http://www.github.com/robertgambrel/medicare

BugReports http://www.github.com/robertgambrel/medicare/issues

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, dplyr, ggplot2, maps, magrittr, testthat

VignetteBuilder knitr

Repository https://robertgambrel.r-universe.dev

RemoteUrl https://github.com/robertgambrel/medicare

RemoteRef HEAD

RemoteSha f633bea82b785b1fefde1e282105522a5d0b0c0e

Contents

---

cr_alpha_names   *Provide names for Cost Report "Alpha Table" data*

Description

Provide names for Cost Report "Alpha Table" data

Usage

```
cr_alpha_names()
```

Value

A list of names for the cost report Alpha Table

Examples

```
# get the list
cr_nmrc_names()
```

---

cr_extract   *Extract a variable from a Cost Report*

Description

This function takes a 5-column alpha-numeric dataset or numeric dataset from the Medicare cost reports, which are stored in a long format, and subsets them based on the worksheet number, line number, and column number provided. If desired, it will rename the resulting variable to whatever the user chooses.

Usage

```
cr_extract(dataset, worksheet, row, column, newname = "newvar")
```

Arguments

dataset	The name of a cost report alpha or numeric dataset
worksheet	The name of the workheet, converted to 7-character format
row	The row number of the data, as it appears in the Medicare workbook or documentation (i.e. at least 3 digits. Row 5 must be entered as 500, row 5.1 as 501, etc.)
column	The column number of the data, as it appears in the Medicare workbook (same general rule as for rows)
newname	The name given to the variable that appears as a result of this extraction

Details

It does not automatically adjust for the same variable having different rows / columns in Medicare data formatted for the 1996 vs 2010 form. The user may have to use this function twice, once on each source of data, to extract one variable over time.

It does automatically recode rows and columns into all possible permutations (ie '500', '0500', '00500', 500) when subsetting, since different cost reports use different schema.

This function *does not* throw an error if the parameters yield an empty dataset at any point. It only gives warnings. This is because oftentimes the parameters are valid but the data is missing in the source material, due to CMS scrubbing of what data gets published.

Value

A 2-column dataset: one with the cost report `rpt_rec_number`, used to merge data, and a column of the data requested, which is renamed if desired.

Examples

```
alpha_data <- hospiceALPHA
hospice_name <- cr_extract(alpha_data, "S100000", 100, 100, "name")
```

---

cr_nmrc_names	*Provide names for Cost Report "Numeric Table" data*

Description

Provide names for Cost Report "Numeric Table" data

Usage

```
cr_nmrc_names()
```

Value

A list of names for the cost report Numeric Table

Examples

```
# get the list
cr_nmrc_names()
```

---

cr_rpt_names	*Provide names for Cost Report "Report Table" data*

Description

Provide names for Cost Report "Report Table" data

Usage

```
cr_rpt_names()
```

Value

A list of names for the cost report Report Table

Examples

```
# get the list
cr_rpt_names()
```

---

hospiceALPHA	*Sample Medicare Hospice Cost Report 2014 data*

Description

A dataset containing the alpha data for the first 500 hospices in the Hospice 2014 cost reports. This is raw data, similar to what you'd get on your own with `read.csv("hospc_2014_ALPHA.csv", stringsAsFactors = FALSE)`.

Usage

```
hospiceALPHA
```

Format

A data frame with 61820 rows and 5 variables:

- V1The `rpt_rec_num`, used to link a hospices dataset across the 3 yearly files.
- V2The `wksht_cd`, indicating which worksheet the variable comes from.
- V3The `line_num`, indicating the line on the worksheet where the variable is found.
- V4The `clmn_num`, indicating the column on the worksheet where the variable is found.
- V5The `itm_alphanmrc_itm_txt`, indicating the variable's value.

Source

https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/
Cost-Reports/Hospice.html

---

hospiceNMRC *Sample Medicare Hospice Cost Report 2014 data*

Description

A dataset containing the numeric data for the first 500 hospices in the Hospice 2014 cost reports. This is raw data, similar to what you'd get on your own with `read.csv("hospc_2014_NMRC.csv", stringsAsFactors = FALSE)`.

Usage

    hospiceNMRC

Format

A data frame with 200,202 rows and 5 variables:

- V1The `rpt_rec_num`, used to link a hospices dataset across the 3 yearly files.
- V2The `wksht_cd`, indicating which worksheet the variable comes from.
- V3The `line_num`, indicating the line on the worksheet where the variable is found.
- V4The `clmn_num`, indicating the column on the worksheet where the variable is found.
- V5The `itm_val_num`, indicating the variable's value.

Source

https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/
Cost-Reports/Hospice.html

---

hospiceRPT   *Sample Medicare Hospice Cost Report 2014 data*

Description

A dataset containing the report data for the first 500 hospices in the Hospice 2014 cost reports. This is raw data, similar to what you'd get on your own with `read.csv("hospc_2014_RPT.csv", stringsAsFactors = FALSE)`.

Usage

```
hospiceRPT
```

Format

A data frame with 500 rows and 5 variables:

- V1The `rpt_rec_num`, used to link a hospices dataset across the 3 yearly files.
- V2The `prvdr_ctrl_type_cd`, indicating the ownership structure of the facility.
- V3The `prvdr_num`, a 6-character unique ID used to link the facility's information across time and with other Medicare data.
- V4The `npi`, a unique provider number assigned under HIPAA, which can also be used to link to other data sources.
- V5The `rpt_stus_cd`, the status of the report (initial submission, audited and settled, settled w/o audit, reopened). Facility filings can be revised, so even older years' data might be updated if data is accessed multiple times.
- V6The `fy_bgn_dt`, the start date for the fiscal year of filing. Most facilities submit cost reports soon after close of their fiscal year. They can also have multiple entries in a calendar year if they change their fiscal year start and end dates.
- V7The `fy_end_dt`, fiscal year end date. Usually 365 days after the start, unless the facility is re-basing its fiscal year system.
- V8The `proc_dt`, process date, when the report was processed by CMS.
- V9The `initl_rpt_sw`, initial report indicator, not currently actively used.
- V10The `last_rpt_sw`, last report indicator, not currently used.
- V11The `trnsmtl_num`, the current transmittal number when the report was generated.
- V12The `fi_num`, fiscal intermediary number, which denotes which fiscal intermediary processes the facility's filings.
- V13The `adr_vndr_cd`, automated desk reviewer vendor code, indicating the vendor for the fiscal intermediary.
- V14The `fi_creat_dt`, when the fiscal intermediary processed the submitted report.
- V15The `util_cd`, indicating the level of medicare utilization by the facility.
- V16The `npr_dt`, the date of notice of program reimbursement.
- V17The `spec_ind`, a CMS internal special purposes code.
- V18The `fi_rcpt_dt`, the date the cost report was received by the fiscal intermediary.

Source

https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/
Cost-Reports/Hospice.html

---

medicare   *medicare: tools for simplifying Medicare data analysis*

Description

The medicare package contains useful functions for manipulating raw Medicare public use files. These sometimes come with SAS read-in code provided, but more frequently require the analyst to manually recode and rename variables based on thorough review of the data documentation. This package focuses mostly on Cost Reports and Provider of Service files, but more support will be added for other sources in the future.

See Also

For more information on Medicare the the data available, see:

- CMS's website on publicly available data
- CMS's documentation and download links for Cost Report data
- CMS's documentation and download links for Provider of Services data

---

pos2010   *Sample Medicare Provider of Service 2010 data for hospices*

Description

A sample of Provider of Services data for select hospices in 2010. This is raw data, similar to what you'd get on your own when reading in a csv.

Usage

pos2010

Format

A data frame with 402 rows and 530 variables

Source

https://www.nber.org/data/provider-of-services.html

---

pos_names	*Provide pre-extracted names for Provider of Service file, years 2000-2010*

Description

Provide pre-extracted names for Provider of Service file, years 2000-2010

Usage

```
pos_names(year)
```

Arguments

year   A year in the range 2000-2010

Value

A list of names for the POS dataset in the year specified, in the order that the raw data lists them

This function returns the results of running `pos_names_extract` on the layout and raw data files for the chosen year. For years 2000-2010, the raw data had unhelpful, generic, sequentially numbered variable names. These can be calculated fresh by calling `pos_names_extract` on the imported dataframe and its corresponding layout .txt file. Alternately, this function returns the names compiled by that function for years 2000-2010, saving the user a step.

Examples

```
# get the list
pos_names(2005)
pos_names(2010)
```

---

pos_names_extract	*Process a Provider of Services Record Layour file to extract variable names*

Description

This function takes a Provider of Services Record Layout file (in .txt form) and parses it to extract the descriptive variable names instead of generic ones. For example, the 2006 file variable PROV0085, which is the name of the variable in the raw dataset downloaded from CMS, has a more descriptive name in the layout file: CATEGORY-SUBTYPE-IND.

Usage

```
pos_names_extract(layout_file, data_file)
```

Arguments

`layout_file`	The file location of the layout file
`data_file`	The year's data file

Details

This uses regular expressions to find variable names. It works with years 2000-2010. Later years seem to have descriptive names already, though they aren't necessarily identical across years (nor do they match the names produced here). This code can be run to produce variable names fresh, but pre-computed variable names can also be accessed by `names_pos_20XX()` functions also in this package.

Value

A vector of names, ordered to match the corresponding year's data file

Examples

```
## Not run:
pos_names_extract("pos_2006_layout.txt", pos_2006_data)

## End(Not run)
```

---

price_deflate	*Deflate prices within a sector, relative to a base period.*

Description

CMS publishes yearly final rules that detail annual price increases across various sectors of healthcare spending. In order to analyze spending increases due to utilization changes, it is frequently useful to "deflate" spending based on a reference period, so that observed changes are not due to inflation.

Usage

```
price_deflate(current_value, sector, current_year, reference_year = 2007)
```

Arguments

`current_value`	The current value that is being deflated to reference-period-equivalent dollars
`sector`	What sector is being adjusted. Currently supports: ip, op, phys, snf, hh, hospice, part_b_drugs, part_d_drugs, dme, and other
`current_year`	The current year (2007 - 2014)
`reference_year`	The base period to standardize to (2007 - 2014).

Details

Most sectors come from annual Federal Register Final Rules. Example: Physician Fee Change Rules.

Exceptions are `other`, which uses the general CPI deflator, and `part_d_drugs`, which use the CPI-Pharmaceutical deflator.'

Value

A float value, current_value / (current year index / reference year index)

Examples

```
# convert $100 in current inpatient spending to year 2007 dollars
price_deflate(100, "ip", 2014, 2007)
```

---

subset_column	*Subset the desired dataset, based on a column code.*

Description

This function is not standalone - it is called from the cr_extract function.

Usage

```
subset_column(dataset, column)
```

Arguments

dataset	The name of a cost report alpha or numeric dataset
column	The column of the workheet

Value

A subset of the provided dataset, subset to only having columns of the correct value

---

| subset_row | *Subset the desired dataset, based on a row code.* |

Description

This function is not standalone - it is called from the cr_extract function.

Usage

```
subset_row(dataset, row)
```

Arguments

dataset	The name of a cost report alpha or numeric dataset
row	The number of the row

Value

A subset of the provided dataset, subset to only having rows of the correct value

---

| subset_worksheet | *Subset the desired dataset, based on a worksheet code.* |

Description

This function is not standalone - it is called from the cr_extract function.

Usage

```
subset_worksheet(dataset, worksheet)
```

Arguments

dataset	The name of a cost report alpha or numeric dataset
worksheet	The name of the workheet, converted to 7-character format

Value

A subset of the provided dataset, subset to only having worksheets of the correct value

Index