Package 'touch'

October 14, 2022

Title Tools of Utilization and Cost in Healthcare

Version 0.1-6

Description R implementation of the software tools developed in the H-CUP (Healthcare Cost and Utilization Project) <https://www.hcup-us.ahrq.gov> and AHRQ (Agency for Healthcare Research and Quality) <https://www.ahrq.gov>. It currently contains functions for mapping ICD-9 codes to the AHRQ comorbidity measures and translating ICD-9 (resp. ICD-10) codes to ICD-10 (resp. ICD-9) codes based on GEM (General Equivalence Mappings) from CMS (Centers for Medicare and Medicaid Services).

Depends R (>= 3.2.0)

LinkingTo Rcpp

Imports Rcpp

License GPL (>= 3)

LazyData yes

SysDataCompression xz

SystemRequirements C++11

URL https://github.com/wenjie2wang/touch

BugReports https://github.com/wenjie2wang/touch/issues

Encoding UTF-8

NeedsCompilation yes

Author Wenjie Wang [aut, cre] (<https://orcid.org/0000-0003-0363-3180>),
Yan Li [aut],
Jun Yan [aut] (<https://orcid.org/0000-0003-4401-7296>)

Maintainer Wenjie Wang <wang@wwenjie.org>

Repository CRAN

Date/Publication 2022-07-08 08:50:02 UTC

R topics documented:

cmbd 2
dxDat 3
find_billable 3
icd9Clean 5
icd_map 6
insert_dot 9

Index **11**

---

cmbd *Comorbidity measures from AHRQ HCUP*

Description

This function maps a matrix of ICD9 codes to AHRQ comorbidity measures.

Usage

```
cmbd(icd, drg = NULL, needClean = TRUE, needPrep = TRUE)
```

Arguments

`icd`	a character matrix of icd9 codes, with rows representing patients.
`drg`	a numeric vector of drg codes, with length the same as `nrow(icd)`.
`needClean`	logical, TRUE means cleaning is needed (string trimming, zero supplementation)
`needPrep`	logical, TRUE means preparation is needed (convert char to numeric)

Value

a matrix with the same number of rows as the input and with the comorbidity measures in columns

Author(s)

Jun Yan

References

Elixhauser et. al. (1998)

Examples

```
data(dxDat)
drg <- dxDat$drg
icd <- dxDat[, -1L]
output <- cmbd(icd, drg=drg)
```

dxDat *Sample Data of Diagnosis Code*

Description

A data frame of diagnosis code with variates named drg and dx1, dx2, ..., dx10, where

- drg: drg code for comorbidity;
- dx1-dx10: icd-9 code for 10 diagnosises.

Usage

```
data(dxDat)
```

Format

A data frame with 1000 rows and 11 variables.

Examples

```
data(dxDat)
drg <- dxDat$drg
icd <- dxDat[, - 1L]
```

find_billable *Find Billable ICD Codes from CMS GEMs*

Description

This function tries to find all the billable ICD codes that can be translated by CMS GEMs for each of the input diagnosis codes representing a major category.

Usage

```
find_billable(dx, version = 10, year = 2018,
              match_all = TRUE, decimal = FALSE,
              output = c("character", "list", "tidy-data"), ...)
```

Arguments

dx	A character vector representing diagnosis codes. Each element of the vector can either represent individual diagnosis code or a set of diagnosis codes that are concartenated by commas in between.
version	A numeric value specifying the version of the diagnosis codes that should be either 9 for ICD-9 codes or 10 for ICD-10 codes.

year	A numeric value specifying the year of the CMS GEMs. The currently available options are 2017 and 2018. By default, 2018 CMS GEMs is used.
match_all	A logical value specifying the strategy for finding billable codes based on the input diagnosis category. If TRUE (the default), the function will add the regular expression "[[[[:alnum:]]{1,4}]]" to the tail of diagnosis category so that all the billable diagnosis codes under the given category will be matched. If FALSE, the function will add the regular experssion "[[:alnum:]]" repeatedly at most four times until any set of billable codes are matched.
decimal	A logical value. If TRUE, the diagnosis codes would be returned with decimal points. The default is FALSE.
output	A character value specifying the format of the output. The avaiable options are "character", "list", and "tidy-data". By default, option "character" is used and results in a character vector that consists of element-wise concatenation by commas of all the translated diagnosis codes from the original codes. If "list" is specified, all the translated codes will not be concartenated and a list of character vectors will be returned by the function. Similarly, if "tidy-data" is specified, a data frame in a tidy format will be returned. The first column of the data frame consists of the original diagnosis codes; the second column consists of the translated diagnosis codes.
...	Other arguments for future usage. A warning will be thrown out if any argument goes into ... accidentally.

Details

It is designed to be used with the function icd_map for translating the diagnosis codes that are not billable but representing major categories. Notice that only the character vector output can be directly passed to the function icd_map for translation.

Value

A character vector of the same length with the input vector will be returned by default or if output = "charactor". A list of character vectors will be returned if output = "list"; A data frame in tidy-format will be returned if output = "tidy-data". See argument output for details.

Author(s)

Wenjie Wang <wenjie.2.wang@uconn.edu>

See Also

icd_map

Examples

```
library(touch)

### for ICD-9 codes
icd9_major <- c("001", "316", "808", NA, "not_a_dx")
```

```
## find all billable codes under the major category
find_billable(icd9_major, version = 9)

## find the billable codes right under the major category
(icd9_billable <- find_billable(icd9_major, version = 9,
                                match_all = FALSE))

## compare the translation results
icd_map(icd9_major, nomatch = NA)
icd_map(icd9_billable, nomatch = NA)

### for ICD-10 codes
icd10_major <- c("T36.0X2", "T36.3X2", "T38.6X2")

## find all billable codes and output in different formats
find_billable(icd10_major, version = 10)
find_billable(icd10_major, version = 10, output = "list")
find_billable(icd10_major, version = 10, output = "tidy-data")

## add decimal if wanted
(icd10_billable <- find_billable(icd10_major, version = 10, decimal = TRUE))

## compare the translation results
icd_map(icd10_major, from = 10, to = 9, nomatch = NA)
icd_map(icd10_billable, from = 10, to = 9)
```

`icd9Clean` *Reformat Comorbidity Measures*

Description

This function processes the character matrix of ICD9 codes by converting them to character codes of length 5. For SAS procedure from HCUP, it trims all character string to be of length 5, adds the missing trailing white space, and capitalizes the first character in ICD9 codes.

Usage

```
icd9Clean(input, style = c("touch", "hcup"))
```

Arguments

`input`	a character matrix of ICD9 codes, with rows representing patients.
`style`	a character vector of length one indicating the reformatting style to follow. The possible option are "touch" and "hcup". The former does the cleaning for this package; The latter does the reformatting for the SAS script provided by HCUP.

Value

a matrix of cleaned ICD9 codes.

Author(s)

Jun Yan and Wenjie Wang

Examples

```
data(dxDat)
icd <- dxDat[, -1L]
output <- icd9Clean(icd)
```

`icd_map` *Translatation of ICD Codes by General Equivalence Mappings (GEMs)*

Description

An open-source implementation in R similar to the Mapping tool developed by the Agency for Healthcare Research and Quality (AHRQ).

Usage

```
icd_map(dx, from = 9, to = 10, year = 2018,
        method = c("gem", "reverse-gem", "both", "multi-stage"),
        decimal = FALSE, nomatch = c('""', NA),
        output = c("character", "list", "tidy-data"), cache = TRUE, ...)
```

Arguments

`dx`	A character vector representing diagnosis codes. Each element of the vector can either represent individual diagnosis code or a set of diagnosis codes that are concartenated by commas in between.
`from`	A integer value specifying the original code version. Currently, the available options are `9` or `10`.
`to`	A integer value specifying the original code version. Currently, the available options are `9` or `10`. If the input `from` and `to` are the same, the function will skip all the translation and return the input `dx` with a warning.
`year`	A numeric value specifying the year of the CMS GEMs. The currently available options are `2017` and `2018`. By default, 2018 CMS GEMs is used.
`method`	A character string specifying the translateion method. The available options are `"gem"` for CMS GEM, `"reverse-gem"` for the reverse of CMS GEM, `"both"` for both GEM and reverse GEM, `"multi-stage"` for multiple stage procedure. See Section Details for more detailed description of the procedure.
`decimal`	A logical value. If `TRUE`, the diagnosis codes would be returned with decimal points. The default is `FALSE`.

nomatch — A character string indicating no translation result can be found through the specified mapping. By default, empty strings, "", will be used. Another available option is NA (or more specific NA_character_). In that case, the code will be translated to NA_character_ if no translataion result can be found.

output — A character value specifying the format of the output. The avaiable options are "character", "list", and "tidy-data". By default, option "character" is used and results in a character vector that consists of element-wise concatenatation by commas of all the translated diagnosis codes from the original codes. If "list" is specified, all the translated codes will not be concartenated and a list of character vectors will be returned by the function. Similarly, if "tidy-data" is specified, a data frame in a tidy format will be returned. The first column of the data frame consists of the original diagnosis codes; the second column consists of the translated diagnosis codes.

cache — A logical value specifying whether to cache all the mappings for method = "both" (both CMS GEM and its reverse mapping), and method = "multi-stage" (the multiple stage procedure). If TRUE by default, the specified mapping will be generated, cached and, applied to the translation. If FALSE, the CMS GEM and its reverse mapping will be used for translatation every time without cache. It is recommended to set cache = TRUE for translation from ICD-9 to ICD-10. For translation from ICD-10 to ICD-9, the caching process only takes noticeable time (usually several minutes at most) for the multi-stage procedure.

... — Other arguments for future usage. A warning will be thrown out if any argument goes into ... accidentally.

Details

This function aims to efficiently translates the ICD diagnosis codes to the a different version by the General Equivalence Mappings (GEMs) developed by the National Center for Health Statistics, Centers for Medicare and Medicaid Services (CMS), AHIMA, the American Hospital Association, and 3M Health Information Systems. The CMS GEMs currently consist of the forward mapping from ICD-9 codes to ICD-10 codes and the backward mapping from ICD-10 codes to ICD-9 codes. In addition to these two mappings, the Agency for Healthcare Research and Quality (AHRQ) also proposed translation by using the reverse mappings and multi-stage procedure.

Taking the translation from ICD-9 codes to ICD-10 codes as an example, the procedure is elaborated as follows: In stage one, the input ICD-9 codes are mapped to ICD-10 codes using the ICD-9 to ICD-10 forward map as well as the reverse of the ICD-10 to ICD-9 backward map. If multiStage = FALSE, the procedure will return the translation results from stage one (and skip the following stages). Otherwise, the procedure will continue and become a multiple stage process. In stage two, the ICD-10 codes output from the stage one are mapped back to ICD-9 codes using the backward map as well as the reverse of the forward map; In stage three, it applies the forward map and reverse-backward map used in stage one again to the ICD-9 codes from the stage two and return the resulting ICD-10 codes.

The flags of the GEMs are not exported from this function. For codes with positive combination flags, the combination of the converted ICD-10 codes is indicated by the plus sign "+". For example, the ICD-9 code "24951" can be translated by 2018 GEMs to ICD-10 code, "E0839", "E0939", or one of the codes from ("E08311","E08319", "E0836", "E09311" "E09319", "E0936") with "E0865". The plus sign in the output, such as "E08311+E0865", is used to indicate the combination of "E08311" and "E0865".

Value

A character vector of the same length with the input vector will be returned by default or if `output = "charactor"`. A list of character vectors will be returned if `output = "list"`; A data frame in tidy-format will be returned if `output = "tidy-data"`. See argument `output` for details.

Author(s)

Wenjie Wang <wenjie.2.wang@uconn.edu>

References

2017-ICD-10-CM-and-GEMs. The U.S. Centers for Medicare & Medicaid Services. 22 August, 2016. https://www.cms.gov/medicare/coding/icd10/2017-icd-10-cm-and-gems. Accessed 5 July, 2022.

2018-ICD-10-CM-and-GEMs. The U.S. Centers for Medicare & Medicaid Services. 11 August, 2017. https://www.cms.gov/medicare/coding/icd10/2018-icd-10-cm-and-gems. Accessed 5 July, 2022.

The AHRQ MapIT Automated In-house Stand-alone Mapping Tool. Agency for Healthcare Research and Quality. 26 March, 2018. https://qualityindicators.ahrq.gov/resources/toolkits. Accessed 5 July, 2022.

See Also

find_billable

Examples

```
library(touch)

### some random ICD-9 and ICD-10 codes
icd9codes <- c("0011", "001.1", "316", "29383", "E9808", "V90")
icd10codes <- c("F0390", "F0630", "F54", "F30.13", "A010", "M61019")

### forward mapping from ICD-9 to ICD-10
icd_map(icd9codes)
icd_map(icd9codes, decimal = TRUE, nomatch = NA)

### backward mapping from ICD-10 to ICD-9
icd_map(icd10codes, from = 10, to = 9)
icd_map(icd10codes, from = 10, to = 9, nomatch = NA, output = "list")
icd_map(icd10codes, from = 10, to = 9,
        decimal = TRUE, nomatch = NA, output = "tidy")

### reverse-backward mapping from ICD-9 to ICD-10
icd_map(icd9codes, method = "reverse-gem")
icd_map(icd9codes, method = "reverse", decimal = TRUE, nomatch = NA)

### reverse-forward mapping from ICD-10 to ICD-9
icd_map(icd10codes, from = 10, to = 9, method = "reverse-gem")
icd_map(icd10codes, from = 10, to = 9, method = "reverse",
```

```
        decimal = TRUE, nomatch = NA)

### forward and reverse-backward mapping from ICD-9 to ICD-10
icd_map(icd9codes, method = "both")
icd_map(icd9codes, method = "both", decimal = TRUE, nomatch = NA)

### backward and reverse-forward mapping from ICD-10 to ICD-9
icd_map(icd10codes, from = 10, to = 9, method = "both")
icd_map(icd10codes, from = 10, to = 9, method = "both",
        decimal = TRUE, nomatch = NA)

### multi-stage process mapping ICD-9 to ICD-10
icd_map(icd9codes, method = "multi-stage")
icd_map(icd9codes, method = "multi-stage", decimal = TRUE, nomatch = NA)

### multi-stage process mapping ICD-10 to ICD-9
icd_map(icd10codes, from = 10, to = 9,
        method = "multi-stage", cache = FALSE)
icd_map(icd10codes, from = 10, to = 9, method = "multi-stage",
        decimal = TRUE, nomatch = NA, cache = FALSE)

### For codes with positive combination flags
icd_map("24951", output = "list")
## where the "+" signs indicate the code combinations
```

`insert_dot` *Insert Dot to ICD-9 and ICD-10 Diagnosis Codes*

Description

This function adds dot to diagnosis codes of the given ICD version.

Usage

```
insert_dot(dx, version = c(10, 9))
```

Arguments

`dx`	A character vector for the diagnosis codes
`version`	The version of the diagnosis codes. Two available options are `10` and `9`. The default version is `10` for ICD-10. This argument can be either a numerical value (nummerical vector of length one) or a character string (character vector of length one).

Value

A character vector representing the diagnosis codes in decimal format.

Examples

```
library(touch)

## for ICD-9 codes
icd9codes <- c("0011", "001.1", "316", "E950", "E9808", "V90", "v100")
insert_dot(icd9codes, 9)

## for ICD-10 codes
icd10codes <- c("A010", "M61019", "p52", "p528")
insert_dot(icd10codes)
```

Index

∗ **datasets**
dxDat, 3
∗ **manipulation**
cmbd, 2

cmbd, 2

dxDat, 3

find_billable, 3, *8*

icd9Clean, 5
icd_map, *4*, 6
insert_dot, 9