Package 'icd9'

September 29, 2015

Title Tools for Working with ICD-9 Codes, and Finding Comorbidities

Description Calculate comorbidities, Charlson scores, perform fast and accurate validation, conversion, manipulation, filtering and comparison of ICD-9-CM (clinical modification) codes. ICD-9 codes appear numeric but leading and trailing zeroes, and both decimal and non-decimal ``short'' format codes exist. The package enables a work flow from raw lists of ICD-9 codes from hospital billing databases to comorbidities. ICD-9 to comorbidity mappings from Quan (Deyo and Elixhauser versions), Elixhauser and AHRQ included. Any other mapping of codes, such as ICD-10, to comorbidities can be used.

Version 1.3

Date 2015-09-28

Maintainer Jack O. Wasey <jack@jackwasey.com>

URL https://github.com/jackwasey/icd9

Depends R (>= 3.1.0)

Imports Rcpp (>= 0.12.0), checkmate (>= 1.5.1), stats, utils, fastmatch

Suggests testthat, devtools, knitr, microbenchmark, magrittr, XML, memoise, profr, ggplot2, digest, xtable, rmarkdown

LazyData true

LazyDataCompression xz

BugReports https://github.com/jackwasey/icd9/issues

License GPL-3

Copyright See file (inst/)COPYRIGHTS

VignetteBuilder knitr

LinkingTo Rcpp

NeedsCompilation yes

Author Jack O. Wasey [aut, cre, cph], William Murphy [ctb], R Core Team [ctb, cph]

Repository CRAN

Date/Publication 2015-09-29 17:46:47

R topics documented:

`icd9-package` *tools for working with ICD-9 codes, and finding comorbidities*

Description

Calculate comorbidities, and perform fast and accurate validation, conversion, manipulation, filtering and comparison of ICD-9-CM (clinical modification) codes. ICD-9 codes appear numeric but leading and trailing zeroes, and both decimal and non-decimal "short" format codes exist. The package enables a work flow from raw lists of ICD-9 codes from hospital billing databases to comorbidities. ICD-9 to comorbidity mappings from Quan (Deyo and Elixhauser versions), Elixhauser and AHRQ included.

Comorbidities `icd9Comorbid` determines co-morbidities for a set of patients with one or more ICD-9 codes each. `icd9Charlson` calculates Charlson score (Comorbidity Index).

- AHRQ comorbidity mapping is provided, and a function to read the raw SAS code from AHRQ into R data structures. The pre-processed data is available by lazy-loading in `ahrqComorbid`. AHRQ releases new mappings annually.
- Quan revised both Deyo/Charlson and Elixhauser ICD-9 to comorbidity mappings. These are presented as: `link{quanDeyoComorbid}` (which is also derived from the original SAS code used in his publication, referenced in the data documentation), and `quanElixComorbid` which was transcribed directly from the same paper.
- The original Elixhauser mapping is provided, with codes transcribed from the original publication. See `elixComorbid`.

Validation `icd9IsValid` checks whether ICD-9 codes are syntactically valid (although not necessarily genuine ICD-9 diagnoses). In contrast, `icd9IsReal` checks whether ICD-9 codes correspond to diagnoses in the current ICD-9-CM definition from CMS.

Conversion There are many functions to convert ICD-9 codes or their components between different formats and structures. The most commonly used are: `icd9DecimalToShort`, `icd9ShortToDecimal` to convert, e.g., 002.3 to 0023 and back again. See convert for other options.

Manipulation You can find children of a higher-level ICD-9 code with `icd9Children` and find a common parent to a set of children (or arbitrary list of ICD-9 codes) with `icd9Condense`. `icd9Sort` sorts in hierarchical, then numerical order, so 100.0 comes before 100.00, for example. `icd9WideToLong` and `icd9LongToWide` convert the two most common data structures containing patient disease data. This is more sophisticated than standard R or Hadleyverse reshaping.

Explanation, or decoding Use `icd9Explain` to convert a list of codes into human-readable descriptions. This function can optionally reduce the codes to a their top-level groups if all the child members of a group are present. `icd9DiffComorbid` allows summary of the differences between comorbidity mappings, e.g. to find what has changed from year-to-year or between revisions by different authors. `icd9Hierarchy` is a `data.frame` containing the full ICD-9 classification for each diagnosis. `icd9Chapters` contains definitions of chapters, sub-chapters and three-digit groups.

Author(s)

Jack O. Wasey <jack@jackwasey.com>

References

http://www.hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity.jsp

See Also

rClinicalCodes comorbidities

`ahrqComorbid` *AHRQ comorbidities*

Description

This mapping of comorbidities to ICD-9 codes is dervied directly from SAS code provided by AHRQ, and translated into this R data structure. This is a revision of the Elixhauser system, notably exluding cardiac arrythmia.

Format

list of character vectors

Source

http://www.hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity.jsp

---

`ahrqComorbidAll` *AHRQ comorbidities, with HTN, CHF and renal failure subgroups*

Description

This mapping of comorbidities to ICD-9 codes is dervied directly from SAS code provided by AHRQ, and translated into this R data structure. Beyond ahrqComorbid, this includes all the HTN, CHF and renal subgroups, not rolled into their parent categories. This resolution is not needed in typical usage: `ahrqComorbid` is probably what you want.

Format

list of character vectors, each named by co-morbidity

Source

http://www.hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity.jsp

elixComorbid *Elixhauser comorbidities*

Description

The original mapping of Elixhauser's ICD-9-CM to 30 comorbidities. According to Sharabiani et al, this mapping provides the best long-term mortality prediction. The weaknesses of this mapping are that it is based on slightly out-dated ICD-9 codes. I have not yet verified what changes to the ICD-9-CM specification between 1998 and now would impact this mapping.

Format

list of character vectors, each named by co-morbidity

References

Sharabiani, Mansour T. A., Paul Aylin, and Alex Bottle. "Systematic Review of Comorbidity Indices for Administrative Data." Medical Care December 2012 50, no. 12 (2012): 1109-18. doi:10.1097/MLR.0b013e31825f64d0. http://www.ncbi.nlm.nih.gov/pubmed/22929993

Elixhauser, Anne, Claudia Steiner, D. Robert Harris, and Rosanna M. Coffey. "Comorbidity Measures for Use with Administrative Data." Medical Care January 1998 36, no. 1 (1998): 8-27.

elixComorbidNames *Comorbidity names*

Description

These lists provide correctly sorted names of the comorbidities and their particular permutations in both full and abbreviated forms.

In the Elixhauser derived mappings, uncomplicated and complicated hypertension are listed separately, but are always combined in the final analyses. Uncomplicated and complicated hypertension are list separately and as "Hypertension, combined." Abbrev suffix indicates a very short space-free description. Quan's version of Elixhauser is identical. AHRQ's update drops the arrythmia field. The naming convention is a root, e.g. `elixComorbid`, with neither/either/both suffixes `Htn` and `Abbrev`. The Charlson derived mappings do not include hypertension. Abbreviated comorbidity names are helpful for interactive work, whereas the full names might be prefered for plotting.

Format

list, with character/numeric code. 'Hypertension, uncomplicated' and 'Hypertension, complicated' are labelled '6a' and '6b'. Diabetes, cancer, and metastasis are counted independently, as in the original paper, giving the original 30 groups. "01" to "30"

icd9Billable *list of annual versions of billable leaf nodes of ICD-9-CM*

Description

These are derived from the CMS published updates, with versions 23 to 32 currently available going back to 2004/5. The source files back to version 27 have short and long descriptions. The short descriptions are in ASCII with no special characters, whereas the long descriptions contain accented characters which seem to be interpretable as unicode, latin-1 or cp1252. This all done during package creation, but can be repeated by package users, including pulling the data from the web pages directly. Despite my best efforts, current locale can give different results, but this packaged data is correct, with some UTF-8 encoded strings.

Format

list of data frames. Each list item is named by the version as a string, e.g. "32". The constituent data frames have columns `icd9`, `shortDesc`, and `longDesc`.

Source

http://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes.html

icd9Chapters *ICD-9-CM chapters*

Description

`icd9Chapters`, `icd9ChaptersSub` and `icd9ChaptersMajor` contain mappings from the higher level descriptions of ICD-9 codes to the ranges of ICD-9 codes they describe. Helpful in summarizing codes or grouping for human-readable output. These can easily be converted to a co-morbidity mapping, as shown in the vignette.

- 001-139 Infectious And Parasitic Diseases
- 140-239 Neoplasms
- 240-279 Endocrine, Nutritional And Metabolic Diseases, And Immunity Disorders
- 280-289 Diseases Of The Blood And Blood-Forming Organs
- 290-319 Mental Disorders
- 320-389 Diseases Of The Nervous System And Sense Organs
- 390-459 Diseases Of The Circulatory System
- 460-519 Diseases Of The Respiratory System
- 520-579 Diseases Of The Digestive System
- 580-629 Diseases Of The Genitourinary System

- 630-679 Complications Of Pregnancy, Childbirth, And The Puerperium
- 680-709 Diseases Of The Skin And Subcutaneous Tissue
- 710-739 Diseases Of The Musculoskeletal System And Connective Tissue
- 740-759 Congenital Anomalies
- 760-779 Certain Conditions Originating In The Perinatal Period
- 780-799 Symptoms, Signs, And Ill-Defined Conditions
- 800-999 Injury And Poisoning
- V01-V91 Supplementary Classification Of Factors Influencing Health Status And Contact With Health Services
- E000-E999 Supplementary Classification Of External Causes Of Injury And Poisoning

Format

list with chapter/usb-chapter or major names stored in list names, each with two element named character vector with start and end codes.

Source

http://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes.html

---

icd9Charlson *Calculate Charlson Comorbidity Index (Charlson Score)*

Description

Charlson score is calculated in the basis of the Quan revision of Deyo's ICD-9 mapping. (Peptic Ulcer disease no longer warrants a point.) Quan published an updated set of scores, but it seems most people use the original scores for easier comaprison between studies, even though Quan's were more predictive.

Usage

```
icd9Charlson(x, visitId = NULL, scoringSystem = c("original", "charlson",
  "quan"), return.df = FALSE,
  stringsAsFactors = getOption("stringsAsFactors"), ...)

## S3 method for class 'data.frame'
icd9Charlson(x, visitId = NULL,
  scoringSystem = c("original", "charlson", "quan"), return.df = FALSE,
  stringsAsFactors = getOption("stringsAsFactors"), ...)

icd9CharlsonComorbid(x, visitId = NULL, applyHierarchy = FALSE,
  scoringSystem = c("original", "charlson", "quan"))
```

Arguments

x	data frame containing a column of visit or patient identifiers, and a column of ICD-9 codes. It may have other columns which will be ignored. By default, the first column is the patient identifier and is not counted. If visitId is not specified, the first column is used.
visitId	The name of the column in the data frame which contains the patient or visit identifier. Typically this is the visit identifier, since patients come leave and enter hospital with different ICD-9 codes. It is a character vector of length one. If left empty, or NULL, then an attempt is made to guess which field has the ID for the patient encounter (not a patient ID, although this can of course be specified directly). The guesses proceed until a single match is made. Data frames may be wide with many matching fields, so to avoid false positives, anything but a single match is rejected. If there are no successful guesses, and visitId was not specified, then the first column of the data frame is used.
scoringSystem	One of original, charlson, or quan. The first two will give the original Charlson weights for each comorbidity, whereas quan uses the updated weights from Quan 2001.
return.df	single logical value, if true, a two column data frame will be returned, with the first column named as in input data frame (i.e. visitId), containing all the visits, and the second column containing the Charlson Comorbidity Index.
stringsAsFactors	single logical, passed on when constructing data.frame if return.df is TRUE. If the input data frame x has a factor for the visitId, this is not changed, but a non-factor visitId may be converted or not converted according to your system default or this setting.
...	further arguments to pass on to icd9ComorbidQuanDeyo, e.g. icd9Field
applyHierarchy	single logical value, default is FALSE. If TRUE, will drop DM if DMcx is present, etc.

Details

When used, hierarchy is applied per Quan, "The following comorbid conditions were mutually exclusive: diabetes with chronic complications and diabetes without chronic complications; mild liver disease and moderate or severe liver disease; and any malignancy and metastatic solid tumor." The "quan" scoring weights come from the 2011 paper (dx.doi.org/10.1093/aje/kwq433). The comorbidity weights were recalculated using updated discharge data, and some changes, such as Myocardial Infarcation decreasing from 1 to 0, may reflect improved outcomes due to advances in treatment since the original weights were determined in 1984.

Methods (by class)

- data.frame: Charlson scores from data frame of visits and ICD-9 codes

Examples

```
mydf <- data.frame(visitId = c("a", "b", "c"),
                   icd9 = c("441", "412.93", "044.9"))
```

```
cmb <- icd9ComorbidQuanDeyo(mydf, isShort = FALSE, applyHierarchy = TRUE)
cmb
icd9Charlson(mydf, isShort = FALSE)
icd9Charlson(mydf, isShort = FALSE, return.df = TRUE)
icd9CharlsonComorbid(cmb)
```

`icd9Children` *Expand ICD-9 codes to all possible sub-codes*

Description

Expand ICD-9 codes to all possible sub-codes

Usage

```
icd9Children(icd9, isShort = icd9GuessIsShort(icd9), onlyReal = TRUE,
  onlyBillable = FALSE)

icd9ChildrenShort(icd9Short, onlyReal = TRUE, onlyBillable = FALSE)

icd9ChildrenDecimal(icd9Decimal, onlyReal = TRUE, onlyBillable = FALSE)
```

Arguments

`icd9`	is a character vector or factor of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010".
`isShort`	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.
`onlyReal`	single logical value, if TRUE, will limit the search to those codes which appear in the master list, not just syntactically valid codes. Since nearly valid, outdated or new codes may be missed, not limiting to 'real' values will be useful. Ultimately, there will need to be annual (and all-time) master lists of codes and the ability to test against a given master list given the year of the ICD-9 coding.
`onlyBillable`	single logical value, if `TRUE`, describes the input data, stating that it only contains billable codes. Usually, the function will try to guess this, but if you know in advance what they should be, the functions can optionally warn if this is incorrect, and save some computation time. The billable codes are derived from the CMS list. The most recent version is used by default.
`icd9Short`	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
`icd9Decimal`	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

See Also

Other ICD-9 ranges: %i9d%, %i9da%, %i9mj%, %i9s%, %i9sa%, icd9ExpandRange, icd9ExpandRangeDecimal, icd9ExpandRangeMajor, icd9ExpandRangeShort; icd9Condense, icd9CondenseDecimal, icd9CondenseShort; icd9ExpandMinor

Examples

```
library(magrittr)
icd9ChildrenShort("10201", FALSE) # no children other than self
icd9Children("0032", FALSE) # guess it was a short, not decimal code
icd9ChildrenShort("10201", TRUE) # empty because 102.01 is not meaningful
icd9ChildrenShort("003", TRUE) %>% icd9ExplainShort(doCondense = FALSE)
icd9ChildrenDecimal("100.0")
icd9ChildrenDecimal("100.00")
icd9ChildrenDecimal("2.34")
```

---

icd9ComorbidDfToMat *convert matrix of comorbidities into data frame, preserving visitId information*

Description

convert matrix of comorbidities into data frame, preserving visitId information

Usage

```
icd9ComorbidDfToMat(x, visitId = NULL,
  stringsAsFactors = getOption("stringsAsFactors"))
```

Arguments

x	data frame, with a visitId column (not necessarily first), and other columns with flags for comorbidities, as such column names are required.
visitId	The name of the column in the data frame which contains the patient or visit identifier. Typically this is the visit identifier, since patients come leave and enter hospital with different ICD-9 codes. It is a character vector of length one. If left empty, or NULL, then an attempt is made to guess which field has the ID for the patient encounter (not a patient ID, although this can of course be specified directly). The guesses proceed until a single match is made. Data frames may be wide with many matching fields, so to avoid false positives, anything but a single match is rejected. If there are no successful guesses, and visitId was not specified, then the first column of the data frame is used.
stringsAsFactors	whether the resulting data frame should have strings, i.e. visitId converted to factor. Default is to follow the current session option.

Examples

```
longdf <- data.frame(visitId = c("a", "b", "b", "c"),
    icd9 = c("441", "4424", "443", "441"))
cmbdf <- icd9ComorbidElix(longdf, return.df = TRUE)
class(cmbdf)
rownames(cmbdf)
mat.out <- icd9ComorbidDfToMat(cmbdf)
stopifnot(is.matrix(mat.out))
mat.out[, 1:4]
```

`icd9ComorbidMatToDf` *convert matrix of comorbidities into data frame, preserving visitId information*

Description

convert matrix of comorbidities into data frame, preserving visitId information

Usage

```
icd9ComorbidMatToDf(x, visitId = "visitId",
  stringsAsFactors = getOption("stringsAsFactors"))
```

Arguments

`x`	Matrix of comorbidities, with row and columns names defined
`visitId`	Single character string with name for new column in output data frame. Everywhere else, `visitId` describes the input data, but here it is for output data.
`stringsAsFactors`	whether the resulting data frame should have strings, i.e. visitId converted to factor. Default is to follow the current session option.

Examples

```
longdf <- data.frame(visitId = c("a", "b", "b", "c"),
    icd9 = c("441", "4424", "443", "441"))
mat <- icd9ComorbidElix(longdf)
class(mat)
typeof(mat)
rownames(mat)
df.out <- icd9ComorbidMatToDf(mat)
stopifnot(is.data.frame(df.out))
# output data frame has a factor for the visitId column
stopifnot(identical(rownames(mat), as.character(df.out$visitId)))
df.out[, 1:4]
```

icd9Condense *Condense ICD-9 code by replacing complete families with parent codes*

Description

This can be thought of as the inverse operation to `icd9Children`.

Usage

```
icd9Condense(icd9, isShort = icd9GuessIsShort(icd9), onlyReal = NULL,
  warn = TRUE)

icd9CondenseDecimal(icd9Decimal, onlyReal = NULL, warn = TRUE)

icd9CondenseShort(icd9Short, onlyReal = NULL, warn = TRUE,
  keepFactorLevels = FALSE)
```

Arguments

`icd9`	is a character vector or factor of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010".
`isShort`	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.
`onlyReal`	single logical value, if TRUE, will limit the search to those codes which appear in the master list, not just syntactically valid codes. Since nearly valid, outdated or new codes may be missed, not limiting to 'real' values will be useful. Ultimately, there will need to be annual (and all-time) master lists of codes and the ability to test against a given master list given the year of the ICD-9 coding.
`warn`	single logical value, if `TRUE`, give warnings when there is discrepancy between `onlyReal` being `TRUE` yet data containing undefined codes.
`icd9Decimal`	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
`icd9Short`	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
`keepFactorLevels`	single logical value, default `FALSE`. If `TRUE`, will reuse the factor levels from the input data for the output data. This only applies if a factor is given for the input codes.

See Also

Other ICD-9 ranges: %i9d%, %i9da%, %i9mj%, %i9s%, %i9sa%, icd9ExpandRange, icd9ExpandRangeDecimal, icd9ExpandRangeMajor, icd9ExpandRangeShort; icd9Children, icd9Children, icd9ChildrenDecimal, icd9ChildrenShort; icd9ExpandMinor

---

`icd9Count` *count ICD codes or comorbidities for each patient*

Description

`icd9Count` takes a data frame with a column for `visitId` and another for ICD-9 code, and returns the number of distinct codes for each patient.

The visitId field is typically the first column. If there is no column called `visitId` and `visitId` is not specified, the first column is used.

`icd9CountComorbidBin` differs from the other counting functions in that it counts _comorbidities_, not individual diagnoses. It accepts any data frame with either logicals or zero/non-zero contents, with a single column for visitId. No checks are made to see whether visitId is duplicated.

For `icd9Count`, it is assumed that all the columns apart from `visitId` represent actual or possible ICD-9 codes. Duplicate `visitId`s are repeated as given and aggregated.

Usage

```
icd9Count(x, visitId = NULL, return.df = FALSE)

icd9CountComorbidBin(x, visitId = NULL, return.df = FALSE)

icd9CountWide(x, visitId = NULL, return.df = FALSE, aggregate = FALSE)
```

Arguments

`x`	data frame with one row per patient, and a true/false or 1/0 flag for each column. By default, the first column is the patient identifier and is not counted. If `visitId` is not specified, the first column is used.
`visitId`	The name of the column in the data frame which contains the patient or visit identifier. Typically this is the visit identifier, since patients come leave and enter hospital with different ICD-9 codes. It is a character vector of length one. If left empty, or `NULL`, then an attempt is made to guess which field has the ID for the patient encounter (not a patient ID, although this can of course be specified directly). The guesses proceed until a single match is made. Data frames may be wide with many matching fields, so to avoid false positives, anything but a single match is rejected. If there are no successful guesses, and `visitId` was not specified, then the first column of the data frame is used.
`return.df`	single logical, if `TRUE`, return the result as a data frame with the first column being the `visitId`, and the second being the count. If `visitId` was a factor or named differently in the input, this is preserved.

aggregate, single logical, default is FALSE. If TRUE, the length (or rows) of the output will no longer match the input, but duplicate visitIds will be counted together.

Value

vector of the count of comorbidities for each patient. This is sometimes used as a metric of comorbidity load, instead of, or inaddition to metrics like the Charlson Comorbidity Index (aka Charlson Score)

Examples

```
mydf <- data.frame(visitId = c("r", "r", "s"),
                   icd9 = c("441", "412.93", "044.9"))
  icd9Count(mydf, return.df = TRUE)
  icd9Count(mydf)

  cmb <- icd9ComorbidQuanDeyo(mydf, isShort = FALSE, return.df = TRUE)
  icd9CountComorbidBin(cmb)

  wide <- data.frame(visitId = c("r", "s", "t"),
                   icd9_1 = c("0011", "441", "456"),
                   icd9_2 = c(NA, "442", NA),
                   icd9_3 = c(NA, NA, "510"))
  icd9CountWide(wide)
  # or:
  library(magrittr)
  wide %>% icd9WideToLong %>% icd9Count
```

---

icd9DiffComorbid *show the difference between two comorbidity mappings*

Description

Compares two comorbidity:icd9 code mappings. The results are returned invisibly as a list. Only those comorbidities with (case sensitive) overlapping names are compared.

Usage

```
icd9DiffComorbid(x, y, names = NULL, x.names = NULL, y.names = NULL,
  show = TRUE, explain = TRUE)
```

Arguments

x	list of character vectors
y	list of character vectors
names	character vector of the comorbidity names
x.names	character vector of the comorbidity names from x to compare
y.names	character vector of the comorbidity names from y to compare

show	single logical value. The default is TRUE which causes a report to be printed.
explain	single logical value. The default is TRUE which means the differing codes are attempted to be reduced to their parent codes, in order to give a more succinct summary.

Value

A list, each item of which is another list containing the intersections and both asymmetric differences.

Examples

```
icd9DiffComorbid(elixComorbid, ahrqComorbid, "CHF")
## Not run:
# give full report on all comorbidities for these mappings
icd9DiffComorbid(elixComorbid, ahrqComorbid)

## End(Not run)
```

---

icd9ExpandRange *take two ICD-9 codes and expand range to include all child codes*

Description

this is cumbersome code, covering a whole load of edge cases relating to the fact that icd9 codes are **not** in numeric order. An alternative strategy would be to list all the ICD9 codes, then a range would just pick out start and finish positions, and return subset of the list. Not all ICD-9 codes are valid, including some parent codes which have valid children. However, I expect at least some of these have been used in some billing databases.

As with link{icd9ExpandRangeShort} great care is taken not to include codes which have children not in the range. E.g. "100.9" to "101.1" would _not_ include code "101".

onlyReal default is TRUE (a change from previous versions) since this is far more likely to be useful to the end user.

Usage

```
icd9ExpandRange(start, end, isShort = icd9GuessIsShort(c(start, end)),
  onlyReal = TRUE, excludeAmbiguousStart = TRUE,
  excludeAmbiguousEnd = TRUE)

icd9ExpandRangeShort(start, end, onlyReal = TRUE,
  excludeAmbiguousStart = TRUE, excludeAmbiguousEnd = TRUE)

icd9ExpandRangeMajor(start, end, onlyReal = TRUE)

icd9ExpandRangeDecimal(start, end, onlyReal = TRUE,
  excludeAmbiguousStart = TRUE, excludeAmbiguousEnd = TRUE)
```

```
start %i9da% end

start %i9sa% end

start %i9d% end

start %i9mj% end

start %i9s% end
```

Arguments

start,end	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.
onlyReal	single logical value, if TRUE, will limit the search to those codes which appear in the master list, not just syntactically valid codes. Since nearly valid, outdated or new codes may be missed, not limiting to 'real' values will be useful. Ultimately, there will need to be annual (and all-time) master lists of codes and the ability to test against a given master list given the year of the ICD-9 coding.
excludeAmbiguousStart	single logical value, if TRUE the range returned will not include codes which are explicitly listed in the range, but would imply a broader range than specified. E.g. V10 V1009 would by default (FALSE) include V10 even though V10 itself is parent to everything up to V11.
excludeAmbiguousEnd	single logical, same as excludeAmbiguousStart but affects codes at the end of the range. E.g. 99.99 to 101.01 would by default exclude 101 and 101.0

See Also

Other ICD-9 ranges: icd9Children, icd9Children, icd9ChildrenDecimal, icd9ChildrenShort; icd9Condense, icd9CondenseDecimal, icd9CondenseShort; icd9ExpandMinor

Examples

```
"4280 " %i9s% "4289 "
"4280 " %i9s% "42821"
"42799 " %i9sa% "42802" # doesn't include 428 or 4280
"427.99 " %i9da% "428.02"
"V80 " %i9s% " V810 "
```

icd9Explain *explain ICD9 codes*

Description

convert 'decimal' format (123.45 style) ICD9 codes into the name and description for human review there are official ICD9-CM data tables, not with conversion to decimal notation, but to the textual format.

Usage

```
icd9Explain(icd9, isShort = icd9GuessIsShort(icd9), doCondense = TRUE,
  brief = FALSE, warn = TRUE)

icd9ExplainShort(icd9Short, doCondense = TRUE, brief = FALSE, warn = TRUE)

icd9ExplainDecimal(icd9Decimal, doCondense = TRUE, brief = FALSE,
  warn = TRUE)

## S3 method for class 'list'
icd9Explain(icd9, isShort = icd9GuessIsShort(icd9),
  doCondense = TRUE, brief = FALSE, warn = TRUE)

## S3 method for class 'factor'
icd9Explain(icd9, isShort = icd9GuessIsShort(icd9),
  doCondense = TRUE, brief = FALSE, warn = TRUE)

## S3 method for class 'character'
icd9Explain(icd9, isShort = icd9GuessIsShort(icd9),
  doCondense = TRUE, brief = FALSE, warn = TRUE)

## S3 method for class 'numeric'
icd9Explain(icd9, isShort = icd9GuessIsShort(icd9),
  doCondense = TRUE, brief = FALSE, warn = FALSE)
```

Arguments

icd9	is a character vector or factor of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010".
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.
doCondense	single logical value which indicates whether to condense the given set of ICD-9 codes by replacing subsets of codes with 'parent' codes which exactly encompass certain subsets. E.g. If all cholera diagnoses are provided, only '001 - Cholera' needs to be displayed, not all subtypes.

brief	single logical value, default is FALSE. If TRUE, the short description from the canonical CMS descriptions (included in extdata) will be used, otherwise the long description is used.
warn	single logical value, default is TRUE, meaning that codes which do not correspond to diagnoses, or to three-digit codes, will trigger a warning.
icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

Value

data frame, or list of data frames, with fields for ICD9 code, name and description, derived from datamart lookup table

Methods (by class)

- list: explain all ICD-9 codes in a list of vectors
- factor: explain factor of ICD-9 codes
- character: explain character vector of ICD-9 codes
- numeric: explain numeric vector of ICD-9 codes, with warning. In general, this is not allowed because of the possible ambiguity of numeric decimal codes, but for convenience, this is allowed in this case to avoid typing many quotes.

See Also

package comorbidities

Examples

```
icd9ExplainShort(ahrqComorbid[[1]][1:3])
icd9Explain(ahrqComorbid[[1]][1:3], brief = TRUE)
```

---

icd9FilterInvalid *Filter ICD-9 codes by invalidity.*

Description

Filters a data.frame of patients for valid or invalid ICD-9 codes

Usage

```
icd9FilterInvalid(icd9df, icd9Field = NULL, isShort = NULL,
  invert = FALSE)
```

Arguments

icd9df	data frame containing columns for visitId (which is the feault name), icd9 (default for the icd9 code), and maybe also a POA flag.
icd9Field	The column in the data frame which contains the ICD codes. This is a character vector of length one. If it is NULL, icd9 will attempt to guess the column name, looking for progressively less likely possibilities until it matche a single column. Failing this, it will take the first column in the data frame. Specifying the column using this argument avoids the guesswork.
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.
invert	single logical value, if TRUE will return valid instead of invalid rows.

---

icd9FilterPoa *Filters data frame based on present-on-arrival flag*

Description

Present On Arrival (POA) is not a simple flag, since many codes are exempt, unspecified, or unknown. Therefore, two options are given: get all the comorbidities where the POA flag was definitely -ve, coded as "N" or definitely +ve and coded as "Y". Negating one set won't give the other set unless all codes were either Y or N. #describeIn icd9Comorbid

Usage

```
icd9FilterPoa(icd9df, poaField = "poa", poa = icd9PoaChoices)

icd9FilterPoaYes(icd9df, poaField = "poa")

icd9FilterPoaNo(icd9df, poaField = "poa")

icd9FilterPoaNotNo(icd9df, poaField = "poa")

icd9FilterPoaNotYes(icd9df, poaField = "poa")
```

Arguments

icd9df	data frame containing columns for visitId (which is the feault name), icd9 (default for the icd9 code), and maybe also a POA flag.
poaField	The name of column in the data frame which contains the Present On Arrival flag. The flag itself is a single character, typically one of "Y", "N", "E", "X", "U" or empty. The poaField is a character vector of length one.
poa	single character value, being one of poaChoices whether to account for comorbidities flagged as present-on-arrival. This is not a simple binary, since many codes are exempt, unspecified, or unknown. poaField gives the choices: yes, not no, no, not yes. The intermediate codes, such as "exempt", "unknown" and NA mean that "yes" is not the same as "not no."

Functions

- `icd9FilterPoaYes`: Select rows where Present-on-Arrival flag is explicitly "Yes."
- `icd9FilterPoaNo`: Select rows where Present-on-Arrival flag is explicitly "No."
- `icd9FilterPoaNotNo`: Select rows where Present-on-Arrival flag is anything but "No." This includes unknown, exempt, other codes, and of course all those marked "Yes."
- `icd9FilterPoaNotYes`: Select rows where Present-on-Arrival flag is anything byt "Yes." This would group exempt, unknown and other codes under "Not POA" which is unlikely to be a good choice, since exempt codes, of which there are a quite large number, tend to describe chronic or out-of-hospital characteristics.

Examples

```
## Not run:
# using magrittr is beautiful:
library("magrittr", quietly = TRUE, warn.conflicts = FALSE)
myData <- data.frame(
  visitId = c("v1", "v2", "v3", "v4"),
  diag = c("39891", "39790", "41791", "4401"),
  poa = c("Y", "N", NA, "Y"),
  stringsAsFactors = FALSE
)
myData %>% icd9FilterPoaNotNo() %>% icd9ComorbidAhrq
# can fill out named fields also:
myData %>% icd9FilterPoaYes(poaField="poa") %>%
  icd9ComorbidAhrq(icd9Field = "diag", visitId = "visitId", isShort = TRUE)
# can call the core icd9Comorbid function with an arbitrary mapping
myData %>%
icd9FilterPoaYes() %>%
icd9Comorbid(icd9Field = "diag", visitId = "visitId",
  icd9Mapping = quanElixComorbid,
 isShortMapping = TRUE)

## End(Not run)
```

---

icd9FilterValid *Filter ICD-9 codes by validity.*

Description

Filters a data.frame of patients for valid or invalid ICD-9 codes

Usage

```
icd9FilterValid(icd9df, icd9Field = NULL, isShort = NULL, invert = FALSE)
```

Arguments

icd9df	data frame containing columns for visitId (which is the feault name), icd9 (default for the icd9 code), and maybe also a POA flag.
icd9Field	The column in the data frame which contains the ICD codes. This is a character vector of length one. If it is NULL, icd9 will attempt to guess the column name, looking for progressively less likely possibilities until it matche a single column. Failing this, it will take the first column in the data frame. Specifying the column using this argument avoids the guesswork.
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.
invert	single logical value, if TRUE will return invalid instead of valid rows.

---

icd9GetValid *invalid subset of decimal or short ICD-9 codes*

Description

Given vector of short or decimal ICD-9 codes, return (in the same format) those codes which are valid or invalid. Useful for generating error messages with the faulty codes if validation fails.

Usage

```
icd9GetValid(icd9, isShort = icd9GuessIsShort(icd9))

icd9GetValidDecimal(icd9Decimal)

icd9GetValidShort(icd9Short)

icd9GetInvalid(icd9, isShort = icd9GuessIsShort(icd9))

icd9GetInvalidDecimal(icd9Decimal)

icd9GetInvalidShort(icd9Short)
```

Arguments

icd9	is a character vector or factor of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010".
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.
icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

`icd9Short` is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

Functions

- `icd9GetValidDecimal`: Returns subset of codes which are in valid decimal format, e.g. "100" or "V01.10"
- `icd9GetValidShort`: Returns subset of codes which are in valid short format, e.g. "E800" or "41001"
- `icd9GetInvalid`: Returns subset of codes which are not in valid short or decimal format.
- `icd9GetInvalidDecimal`: Returns subset of codes which are not in valid decimal format.
- `icd9GetInvalidShort`: Returns subset of codes which are not in valid short format.

`icd9Hierarchy` *ICD9-CM diagnosis code lookup*

Description

short-form ICD-9 codes with short and long descriptions, and description of each hierarchy level containing each code.

Format

data frame

Source

http://wonder.cdc.gov/wonder/sci_data/codes/icd9/type_txt/icd9cm.asp

Rich text descriptions here: http://www.cdc.gov/nchs/icd/icd9cm.htm http://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes.html This page has versions 23 to 32 (2005 to 2014). At present, only the 2014 data is included in this package.

http://wonder.cdc.gov/wonder/sci_data/codes/icd9/type_txt/icd9abb.asp

http://wonder.cdc.gov/wonder/sci_data/codes/icd9/type_txt/icd9cm.asp

http://wonder.cdc.gov/wonder/sci_data/codes/icd9/type_txt/icdcm.asp

http://wonder.cdc.gov/wonder/sci_data/codes/icd9/type_txt/icd9abb.asp

`icd9IsBillable` *Determine whether codes are billable leaf-nodes*

Description

Codes provided are compared to the most recent version of the CMS list of billable codes, or another version if specified.

Usage

```
icd9IsBillable(icd9, isShort = icd9GuessIsShort(icd9),
  version = getLatestBillableVersion())

icd9IsBillableShort(icd9Short, version = getLatestBillableVersion())

icd9IsBillableDecimal(icd9Decimal, version = getLatestBillableVersion())

icd9GetBillable(icd9, isShort = icd9GuessIsShort(icd9), invert = FALSE,
  version = getLatestBillableVersion())

icd9GetBillableShort(icd9Short, version = getLatestBillableVersion())

icd9GetBillableDecimal(icd9Decimal, version = getLatestBillableVersion())

icd9GetNonBillableShort(icd9Short, version = getLatestBillableVersion())

icd9GetNonBillableDecimal(icd9Decimal, version = getLatestBillableVersion())

icd9GetNonBillable(icd9, isShort = icd9GuessIsShort(icd9),
  version = getLatestBillableVersion())
```

Arguments

`icd9`	is a character vector or factor of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010".
`isShort`	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.
`version`	single character string, default is "32" which is the latest release from CMS. Currently anything from "23" to "32" is accepted. Not numeric because there are possible cases with non-numeric names, e.g. revisions within one year, although none currently implemented.
`icd9Short`	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
invert	single logical value, if TRUE, then the non-billable codes are returned. For functions with logical result, just negate with !. Default is FALSE.

Value

logical vector of same length as input

Functions

- icd9IsBillableShort: Are the given short-form codes leaf (billable) codes in the hierarchy?
- icd9IsBillableDecimal: Are the given decimal-form codes leaf (billable) codes in the hierarchy?
- icd9GetBillable: Return only those codes which are leaf (billable) codes in the hierarchy.
- icd9GetBillableShort: Return only those short-form codes which are leaf (billable) codes in the hierarchy.
- icd9GetBillableDecimal: Return only those decimal-form codes which are leaf (billable) codes in the hierarchy.
- icd9GetNonBillableShort: Return only those short-form codes which are not leaf (billable) codes in the hierarchy. This would include invalid and heading codes.
- icd9GetNonBillableDecimal: Return only those decimal-form codes which are not leaf (billable) codes in the hierarchy. This would include invalid and heading codes.
- icd9GetNonBillable: Return only those codes which are not leaf (billable) codes in the hierarchy. This would include invalid and heading codes. Codes are specified (or guessed) to be all decimal- or short-form.

---

icd9IsN *do codes belong to numeric, V or E classes?*

Description

For each code, return TRUE if numric or FALSE if a V or E code.

Usage

```
icd9IsN(icd9)

icd9IsV(icd9)

icd9IsE(icd9)
```

Arguments

icd9	is a character vector or factor of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010".

Value

logical vector

Functions

- icd9IsV: are the given codes V type?
- icd9IsE: are the given codes E type?

---

icd9IsReal *Check whether ICD-9 codes exist*

Description

This is different from syntactic validity: it looks it up in the canonical list of ICD-9 codes published by the CMS, and which are included in this package under extdata. Checking syntactic validity using link{icd9IsValid} etc. is still useful, with a changing list of icd-9 codes over time, and possible imperfections in the master lists derived from CMS.

Usage

```
icd9IsReal(icd9, isShort = icd9GuessIsShort(icd9), onlyBillable = FALSE)

icd9IsRealShort(icd9Short, onlyBillable = FALSE)

icd9IsRealDecimal(icd9Decimal, onlyBillable = FALSE)

icd9GetReal(icd9, isShort = icd9GuessIsShort(icd9), onlyBillable = FALSE)

icd9GetRealShort(icd9Short, onlyBillable = FALSE)

icd9GetRealDecimal(icd9Decimal, onlyBillable = FALSE)
```

Arguments

icd9	is a character vector or factor of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010".
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.

onlyBillable	single logical value (default FALSE), if TRUE will divert to test whether the codes are in the billable list instead of seeing if they are any leaf or branch node.
icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

Value

logical vector

Functions

- icd9IsRealShort: Are the given short-form codes defined at heading or leaf (billable) level?
- icd9IsRealDecimal: Are the given decimal-form codes defined at heading or leaf (billable) level?
- icd9GetReal: Return only those codes which are heading or leaf (billable), specifying whether codes are all short-form or all decimal-form
- icd9GetRealShort: Return only those short-form codes which are heading or leaf (billable)
- icd9GetRealDecimal: Return only those decimal-form codes which are heading or leaf (billable)

---

icd9IsValid *check whether ICD-9 codes are syntactically valid*

Description

This does not check whether the code coresponds to a real ICD-9-CM billing code, or parent grouping. For that, see icd9IsReal.

Factors are accepted, and since the validation is done with grepl these are handled correctly.

Currently, there is a limitation on NA values. Calling with NA (which is a logical vector of length one by default) fails, because it is not a string. This is rarely of significance in real life, since the NA will be part of a character vector of codes, and will therefore be cast already to NA_character

NA values result in a return value of FALSE.

Usage

```
icd9IsValid(icd9, isShort)

icd9Valid(icd9, isShort)

icd9IsValidDecimal(icd9Decimal)
```

```
icd9ValidDecimal(icd9)

icd9IsValidShort(icd9Short)

icd9ValidShort(icd9)

icd9IsValidShortV(icd9Short)

icd9IsValidShortE(icd9Short)

icd9IsValidShortN(icd9Short)

icd9IsValidMajor(major)
```

Arguments

icd9	is a character vector or factor of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010".
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.
icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
major	character vector of 'major' part of ICD-9 codes, i.e. that part which falls before the decimal point, in decimal notation. (In 5 digit notation, the 'major' part is be the first three characters (with leading zeroes), and includes V or E prefix. xyz

Details

Leading zeroes in the decimal form are not ambiguous. Although integer ICD-9 codes could be intended by the user, there is a difference between 100, 100.0, 100.00. Therefore a warning is given if a numeric value is provided

Value

logical vector with TRUE or FALSE for each icd9 code provided according to its validity

Three-digit validation

isValidMajor validates just the 'major' three-digit part of an ICD-9 code. This can in fact be provided as a numeric, since there is no ambiguity. Numeric-only codes should be one to three digitis, V codes are followed by one or two digits, and E codes always by three digits between 800 and 999.

See Also

icd9IsValidDecimal, icd9IsValidShort, http://www.stata.com/users/wgould/icd9/icd9.hlp urlhttp://www.sascommunity.org/wiki/Validate_the_format_of_ICD-9_codes

Other ICD9 validation: icd9GetInvalidMappingDecimal, icd9GetInvalidMappingShort, icd9IsValidMapping, icd9IsValidMappingDecimal, icd9IsValidMappingShort

Examples

```
icd9IsValidShort(c("", "1", "22", "333", "4444", "123.45", "V",
                   "V2", "V34", "V567", "E", "E1", "E70", "E"))
  icd9IsValidMajor(c("", "1", "22", "333", "4444", "123.45", "V",
                   "V2", "V34", "V567", "E", "E1", "E70", "E"))
```

---

icd9IsValidMapping *validate an icd9 mapping to comorbidities*

Description

takes each item in each vector of the list of vectors and checks validity, or returns those items which are valid for each comorbidity.

Usage

```
icd9IsValidMapping(icd9Mapping, isShort)

icd9IsValidMappingShort(icd9Mapping)

icd9IsValidMappingDecimal(icd9Mapping)

icd9GetInvalidMappingShort(icd9Mapping)

icd9GetInvalidMappingDecimal(icd9Mapping)
```

Arguments

icd9Mapping	named list containing vectors of icd9 codes. E.g. the AHRQ comorbidities, contains list(OBESE = c("2780", "27800", "27801","27803", "V8554", "79391", "64910", "6491 c("3004", "30112", "3090", "3091", "311")) amongst other longer groups.
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.

See Also

Other ICD9 validation: icd9IsValid, icd9IsValidDecimal, icd9IsValidMajor, icd9IsValidShort, icd9IsValidShortE, icd9IsValidShortN, icd9IsValidShortV, icd9Valid, icd9ValidDecimal, icd9ValidShort

icd9LongToWide *convert ICD data from long to wide format*

Description

This is more complicated than reshape or reshape2::dcast allows. This is a reasonably simple solution using built-in functions.

Usage

```
icd9LongToWide(icd9df, visitId = NULL, icd9Field = NULL, prefix = "icd_",
  min.width = 0, aggregate = TRUE, return.df = FALSE)
```

Arguments

icd9df	data.frame of long-form data, one column for visitId and one for ICD code
visitId	The name of the column in the data frame which contains the patient or visit identifier. Typically this is the visit identifier, since patients come leave and enter hospital with different ICD-9 codes. It is a character vector of length one. If left empty, or NULL, then an attempt is made to guess which field has the ID for the patient encounter (not a patient ID, although this can of course be specified directly). The guesses proceed until a single match is made. Data frames may be wide with many matching fields, so to avoid false positives, anything but a single match is rejected. If there are no successful guesses, and visitId was not specified, then the first column of the data frame is used.
icd9Field	The column in the data frame which contains the ICD codes. This is a character vector of length one. If it is NULL, icd9 will attempt to guess the column name, looking for progressively less likely possibilities until it matche a single column. Failing this, it will take the first column in the data frame. Specifying the column using this argument avoids the guesswork.
prefix	character, default "icd_" to prefix new columns
min.width,	single integer, if specified, writes out this many columns even if no patients have that many codes. Must be greater than or equal to the maximum number of codes per patient.
aggregate	single logical value, if TRUE (the default) will take more time to find out-of-order visitIds, and combine all the codes for each unique visitId. If FALSE, then out-of-order visitIds will result in a row in the output data per contiguous block of identical visitIds.
return.df	single logical value, if TRUE, return a data frame with a field for the visitId. This may be more convenient, but the default of FALSE gives the more natural return data of a matrix with rownames being the visitIds.

See Also

Other ICD-9 convert: convert, icd9DecimalToParts, icd9DecimalToPartsCpp, icd9DecimalToShort, icd9PartsToDecimal, icd9PartsToShort, icd9ShortToParts, icd9ShortToPartsCpp; icd9ChaptersToMap; icd9DropLeadingZeroes, icd9DropLeadingZeroesDecimal, icd9DropLeadingZeroesMajor, icd9DropLeadingZeroes
icd9WideToLong

Examples

```
longdf <- data.frame(visitId = c("a", "b", "b", "c"),
    icd9 = c("441", "4424", "443", "441"))
  icd9LongToWide(longdf)
  icd9LongToWide(longdf, prefix = "ICD10_")
```

---

icd9PoaChoices *present-on-admission flags*

Description

Present-on-admission (POA) is not simply true or false. It can be one of a number of indeterminate values, including NA, or "Y" or "N". "Present-on-arrival" in this context will mean a positive "Y" flag and nothing else. Other interpretations are to include all ICD-9 codes not flagged 'N': but this would include many unknowns. Conversely, when looking for definite new diagnoses, we should only find 'N' flagged codes, and ignore anything marked "Y" or indeterminate. This gives four options: poa == "Y" , poa == "N", poa != "N" , poa != "Y".

Usage

```
icd9PoaChoices
```

Format

```
chr [1:4] "yes" "no" "notYes" "notNo"
```

---

icd9ShortToDecimal *Convert ICD-9 codes between short and decimal forms*

Description

Convert ICD-9 codes between short and decimal forms

Usage

```
icd9ShortToDecimal(icd9Short)

icd9DecimalToShortOld(icd9Decimal)
```

Arguments

icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

---

icd9Sort *sort short-form icd9 codes*

Description

Sorts lists of numeric only, V or E codes. Note that a simple numeric sort does not work for ICD-9 codes, since "162" > "1620", and also V codes precede E codes.

Usage

```
icd9Sort(icd9, isShort = icd9GuessIsShort(icd9))

icd9SortShort(icd9Short)

icd9SortDecimal(icd9Decimal)
```

Arguments

icd9	is a character vector or factor of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010".
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form. Where reasonable, this is guessed from the input data.
icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

Details

Implementation used fast built-in sort, then shuffles the E codes to the end.

Value

sorted vector of ICD-9 codes. Numeric, then E codes, then V codes.

`icd9VanWalraven` *Calculate van Walraven Elixhauser Score*

Description

van Walraven Elixhauser score is calculated from the Quan revision of Elixhauser's ICD-9 mapping. This function allows for the hierarchical exlusion of less severe versions of comorbidities when their more severe version is also present via the applyHeirarchy argument. For the Elixhauser comorbidities, this is diabetes v. complex diabetes and solid tumor v. metastatic tumor

Usage

```
icd9VanWalraven(x, visitId = NULL, return.df = FALSE,
  stringsAsFactors = getOption("stringsAsFactors"), ...)

## S3 method for class 'data.frame'
icd9VanWalraven(x, visitId = NULL, return.df = FALSE,
  stringsAsFactors = getOption("stringsAsFactors"), ...)

icd9VanWalravenComorbid(x, visitId = NULL, applyHierarchy = FALSE)
```

Arguments

`x`	data frame containing a column of visit or patient identifiers, and a column of ICD-9 codes. It may have other columns which will be ignored. By default, the first column is the patient identifier and is not counted. If `visitId` is not specified, the first column is used.
`visitId`	The name of the column in the data frame which contains the patient or visit identifier. Typically this is the visit identifier, since patients come leave and enter hospital with different ICD-9 codes. It is a character vector of length one. If left empty, or `NULL`, then an attempt is made to guess which field has the ID for the patient encounter (not a patient ID, although this can of course be specified directly). The guesses proceed until a single match is made. Data frames may be wide with many matching fields, so to avoid false positives, anything but a single match is rejected. If there are no successful guesses, and `visitId` was not specified, then the first column of the data frame is used.
`return.df`	single logical value, if true, a two column data frame will be returned, with the first column named as in input data frame (i.e. `visitId`), containing all the visits, and the second column containing the Charlson Comorbidity Index.
`stringsAsFactors`	single logical, passed on when constructing data.frame if `return.df` is `TRUE`. If the input data frame `x` has a factor for the visitId, this is not changed, but a non-factor visitId may be converted or not converted according to your system default or this setting.
`...`	further arguments to pass on to `icd9ComorbidQuanElix`, e.g. `icd9Field`, `applyHeirarchy`
`applyHierarchy`	single logical value, default is `FALSE`. If `TRUE`, will drop DM if DMcx is present, etc.

Methods (by class)

- `data.frame`: van Walraven scores from data frame of visits and ICD-9 codes

Author(s)

wmurphyrd

References

van Walraven C, Austin PC, Jennings A, Quan H, Forster AJ. A Modification to the Elixhauser Comorbidity Measures Into a Point System for Hospital Death Using Administrative Data. Med Care. 2009; 47(6):626-633. http://www.ncbi.nlm.nih.gov/pubmed/19433995

Examples

```
mydf <- data.frame(visitId = c("a", "b", "c"),
                   icd9 = c("412.93", "441", "044.9"))

print(
  cmb <- icd9ComorbidQuanElix(mydf, isShort = FALSE, applyHierarchy = TRUE, return.df=TRUE)
)
icd9VanWalravenComorbid(cmb)

icd9VanWalraven(mydf)
icd9VanWalraven(mydf, return.df = TRUE)
```

`icd9WideToLong` *convert ICD data from wide to long format*

Description

This is different enough to `dcast` in `reshape2` that it needs writing again specifically for ICD codes. This function packages the core `reshape` function. Empty strings and NA values will be dropped, and everything else kept. No validation of the ICD codes is done.

Usage

```
icd9WideToLong(x, visitId = NULL, icdLabels = NULL, icdName = "icdCode",
  icdRegex = c("icd", "diag", "dx_", "dx"), verbose = FALSE)
```

Arguments

`x`	`data.frame` in wide format, i.e. one row per patient, and multiple columns containing ICD codes, empty strings or NA.

visitId	The name of the column in the data frame which contains the patient or visit identifier. Typically this is the visit identifier, since patients come leave and enter hospital with different ICD-9 codes. It is a character vector of length one. If left empty, or NULL, then an attempt is made to guess which field has the ID for the patient encounter (not a patient ID, although this can of course be specified directly). The guesses proceed until a single match is made. Data frames may be wide with many matching fields, so to avoid false positives, anything but a single match is rejected. If there are no successful guesses, and visitId was not specified, then the first column of the data frame is used.
icdLabels	vector of column names in which codes are found. If NULL, all columns matching icd or ICD will be included.
icdName	character vector length one containing the new column name for the ICD codes, defaults to "icdCode"
icdRegex	vector of character strings containg a regex to identify ICD-9 diagnosis columns to try (case-insensitive) in order. Default is c("icd", "diag", "dx_", "dx")
verbose	single logical value, defaults to FALSE in most functions.

Value

data frame with visitId column named the same as input, and a column named by icd.name containing all the non-NA and non-empty codes found in the wide input data.

See Also

Other ICD-9 convert: convert, icd9DecimalToParts, icd9DecimalToPartsCpp, icd9DecimalToShort, icd9PartsToDecimal, icd9PartsToShort, icd9ShortToParts, icd9ShortToPartsCpp; icd9ChaptersToMap; icd9DropLeadingZeroes, icd9DropLeadingZeroesDecimal, icd9DropLeadingZeroesMajor, icd9DropLeadingZeroes
icd9LongToWide

Examples

```
widedf <- data.frame(visitId = c("a", "b", "c"),
    icd9_01 = c("441", "4424", "441"),
    icd9_02 = c(NA, "443", NA))
  icd9WideToLong(widedf)
```

---

quanDeyoComorbid *Quan adaptation of Deyo/Charlson comorbidities*

Description

Derived programmatically from the SAS code used in the original publication. According to the referenced study, this provides the best presdictor of in-patient to <30d mortality. Of note, Deyo drops the distinction between leukemia, lymphoma and non-metastatic cancer. As far as I have looked into this, in the rare cases where someone had two or three of leukemia, lymphoma and non-metastatic cancer, the Quan adaptation would give a lower Charlson score than the original scheme. The Deyo original Charlson to ICD-9-CM groups does include distinct categories for these things.

Format

list of character vectors, each named by co-morbidity

References

Quan, Hude, Vijaya Sundararajan, Patricia Halfon, Andrew Fong, Bernard Burnand, Jean-Christophe Luthi, L. Duncan Saunders, Cynthia A. Beck, Thomas E. Feasby, and William A. Ghali. "Coding Algorithms for Defining Comorbidities in ICD-9-CM and ICD-10 Administrative Data." Medical Care 43, no. 11 (November 1, 2005): 1130-39. http://www.ncbi.nlm.nih.gov/pubmed/16224307 http://web.archive.org/web/20110225042437/http://www.chaps.ucalgary.ca/sas

---

quanElixComorbid *Quan adaptation of Elixhauser comorbidities*

Description

These were transcribed directly from the Quan paper referenced.

Format

list of character vectors, each named by co-morbidity

References

Quan, Hude, Vijaya Sundararajan, Patricia Halfon, Andrew Fong, Bernard Burnand, Jean-Christophe Luthi, L. Duncan Saunders, Cynthia A. Beck, Thomas E. Feasby, and William A. Ghali. "Coding Algorithms for Defining Comorbidities in ICD-9-CM and ICD-10 Administrative Data." Medical Care 43, no. 11 (November 1, 2005): 1130-39. http://www.ncbi.nlm.nih.gov/pubmed/16224307 http://web.archive.org/web/20110225042437/http://www.chaps.ucalgary.ca/sas

---

vermont_dx *de-identified data from public Vermont source for 2013*

Description

de-identified data from public Vermont source for 2013

Usage

```
.vermont()
```

Format

CSV original, minimally processed into R data.

Details

Conditions of Release Release of public use data is subject to the following conditions, which the requestor agrees to upon accepting copies of the data:

1. The data may not be used in any manner that attempts to or does identify, directly or indirectly, any individual patient or physician.

2. The requestor agrees to incorporate the following, or a substantially similar, disclaimer in all reports or publications that include public use data: "Hospital discharge data for use in this study were supplied by the Vermont Association of Hospitals and Health Systems-Network Services Organization (VAHHS-NSO) and the Vermont Department of Banking, Insurance, Securities and Health Care Administration (BISHCA). All analyses, interpretations or conclusions based on these data are solely that of [the requestor]. VAHHS-NSO and BISHCA disclaim responsibility for any such analyses, interpretations or conclusions. In addition, as the data have been edited and processed by VAHHS-NSO, BISHCA assumes no responsibility for errors in the data due to coding or processing"

Author(s)

Vermont Division of Health Care Administration

Source

http://healthvermont.gov/research/hospital-utilization/RECENT_PU_FILES.aspx

Index