Package: medicalrisk (via r-universe)

November 15, 2026

Type Package

Title Medical Risk and Comorbidity Tools for ICD-9-CM Data

Version 1.3.9001

Description Generates risk estimates and comorbidity flags from ICD-9-CM codes available in administrative medical datasets. The package supports the Charlson Comorbidity Index, the Elixhauser Comorbidity classification, the Revised Cardiac Risk Index, and the Risk Stratification Index. Methods are table-based, fast, and use the 'plyr' package, so parallelization is possible for large jobs. Also includes a sample of real ICD-9 data for 100 patients from a publicly available dataset. Compatibility updates by Renova.

Depends R (>= 3.0.0)

Imports plyr (>= 1.5), reshape2, hash

Suggests testthat, knitr, ggplot2, gridExtra

License GPL-3 | file LICENSE

LazyData true

VignetteBuilder knitr

RoxygenNote 7.0.2

Encoding UTF-8

BugReports https://github.com/patrickmdnet/medicalrisk/issues

URL https://github.com/patrickmdnet/medicalrisk

NeedsCompilation no

Author Patrick McCormick [aut, cre], Thomas Joseph [aut]

Maintainer Patrick McCormick <patrick.mccormick@alum.mit.edu>

Config/pak/sysreqs libicu-dev

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

Date/Publication 2026-11-15 20:26:19 UTC

RemoteUrl https://github.com/renova-r/medicalrisk

RemoteRef renova-1.3.9001

RemoteSha bde34970c85bda52bb5d434357d4635041d9292b

Contents

charlson_list . . . 2
charlson_weights . . . 3
charlson_weights_orig . . . 4
elixhauser_list . . . 5
generate_charlson_index_df . . . 5
generate_comorbidity_df . . . 6
icd9cm_charlson_deyo . . . 7
icd9cm_charlson_quan . . . 8
icd9cm_charlson_romano . . . 10
icd9cm_elixhauser_ahrq37 . . . 11
icd9cm_elixhauser_quan . . . 12
icd9cm_list . . . 13
icd9cm_rcri . . . 14
icd9cm_sessler_rsi . . . 15
melt_icd9list . . . 16
merge_icd9_dx_and_procs . . . 17
rsi_beta_1yrpod . . . 18
rsi_beta_30dlos . . . 18
rsi_beta_30dpod . . . 19
rsi_beta_inhosp . . . 19
rsi_sample_data . . . 20
rsi_sample_results . . . 20
sessler_get_single_beta . . . 21
verify_sessler_rsi . . . 22
vt_inp_sample . . . 22

Index **24**

`charlson_list` *List of Charlson comorbidities*

Description

List of Charlson comorbidities

Usage

```
charlson_list
```

Format

A list, with one column for each comorbidity; value is a textual description

Examples

```
# List the strings used to refer to Charlson comorbidities
names(charlson_list)

# List descriptions of comorbidities
charlson_list
```

`charlson_weights` *Map of Charlson comorbidity categories to revised weights*

Description

List that links the Charlson comorbidity categories to revised weights as calculated by Schneeweiss in Table 4 of his paper.

Usage

```
charlson_weights
```

Format

A list, with Charlson comorbidities as names and weight as value

Details

Revised Schneeweiss weights:

0 = Connective tissue dz, Ulcer

1 = MI, PVD, CVD, Diabetes, Hemiplegia

2 = CHF, Chronic pulm dz, Mild liver dz, Diabetes with end organ damage, Any tumor, Leukemia, Lymphoma

3 = Dementia, Mod or severe renal dz

4 = Moderate or severe liver dz, AIDS

6 = Metastatic solid tumor

References

1. Schneeweiss S, Wang PS, Avorn J, Glynn RJ: Improved comorbidity adjustment for predicting mortality in Medicare populations. Health services research 2003; 38:1103 http://www.ncbi.nlm.nih.gov/pubmed/12968819

See Also

`charlson_weights_orig`, `icd9cm_charlson_deyo`, `icd9cm_charlson_romano`, `icd9cm_charlson_quan`, `melt_icd9list`

Examples

```
charlson_weights["dementia"]
```

`charlson_weights_orig` *Map of Charlson comorbidity categories to weights*

Description

List that links the Charlson comorbidity categories to the original weights (specified in the original Charlson paper, Table 3)

Usage

```
charlson_weights_orig
```

Format

A list, with Charlson comorbidities as names and weight as value

Details

Original Weights:

1 = MI, CHF, PVD, CVD, Dementia, Chronic pulm dz, Connective tissue dz, Ulcer, Mild liver dz, Diabetes

2 = Hemiplegia, Mod or severe renal dz, Diabetes with end organ damage, Any tumor, Leukemia, Lymphoma

3 = Moderate or severe liver dz

6 = Metastatic solid tumor, AIDS

References

1. Charlson ME, Pompei P, Ales KL, MacKenzie CR: A new method of classifying prognostic comorbidity in longitudinal studies: development and validation. Journal of chronic diseases 1987; 40:373-83 http://www.ncbi.nlm.nih.gov/pubmed/3558716

See Also

`charlson_weights`, `icd9cm_charlson_deyo`, `icd9cm_charlson_romano`, `icd9cm_charlson_quan`, `melt_icd9list`

Examples

```
charlson_weights_orig["aids"]
```

elixhauser_list *List of Elixhauser comorbidities*

Description

List of Elixhauser comorbidities

Usage

```
elixhauser_list
```

Format

A list, with one column for each comorbidity; value is a textual description

Examples

```
# List the strings used to refer to Elixhauser comorbidities
names(elixhauser_list)

# List descriptions of comorbidities
elixhauser_list
```

generate_charlson_index_df *Calculate the Charlson Comorbidity Index*

Description

`generate_charlson_index_df` merges a data frame of Charlson comorbidities with `charlson_weights` and sums the results per patient.

Usage

```
generate_charlson_index_df(
  df,
  idvar = "id",
  weights = medicalrisk::charlson_weights
)
```

Arguments

`df`	a data frame with ID column `idvar` and logical columns for each comorbidity, such as that generated by `generate_comorbidity_df`
`idvar`	string with name of ID variable within `df`
`weights`	defaults to `charlson_weights`

Value

a dataframe with two columns, `idvar` and `"index"`

See Also

`generate_comorbidity_df`, `charlson_weights`, `charlson_weights_orig`

Examples

```
# calculate Charlson Comorbidity Index for all patients in the \code{\link{vt_inp_sample}}
data(vt_inp_sample)
generate_charlson_index_df(generate_comorbidity_df(vt_inp_sample))
```

`generate_comorbidity_df`

Generate a comorbidity dataframe

Description

Merges a given DF of IDs and ICD-9-CM codes to one of the ICD9CM maps, removes redundant comorbidities, and returns a dataframe.

Usage

```
generate_comorbidity_df(
  df,
  idvar = "id",
  icd9var = "icd9cm",
  icd9mapfn = icd9cm_charlson_quan,
  .progress = "none",
  .parallel = FALSE,
  .paropts = NULL
)
```

Arguments

`df`	a data frame with at least two columns, specified as `idvar` and `icd9var`.
`idvar`	string with name of ID variable within `df` (defaults to "id")
`icd9var`	string with name of ICD code variable within `df` (defaults to `icd9cm`)
`icd9mapfn`	Function to generate comorbidity data frame from ICD-9 codes (defaults to `icd9cm_charlson_quan`)
`.progress`	passed to `ddply`
`.parallel`	passed to `ddply`
`.paropts`	passed to `ddply`

Details

Redundancy rules: * If "tumor" and "mets", only "mets" will be returned. * If "htn" and "htncx", only "htncx" will be returned. * If "dm" and "dmcx", only "dmcx" will be returned. * If "liver" and "modliver", only "modliver" will be returned.

Van Walraven has a modification adopted here where the following "dmcx" codes are downgraded to "dm" if the specific DM complication is separately coded: * D2(49|50)4x is DM w renal * D2(49|50)6x is DM w neuro * D2(49|50)7x is DM w PVD

Cases without any comorbidities will not appear in the returned data frame.

Value

a dataframe with column `idvar` and a logical column for each comorbidity

Examples

```
cases <- data.frame(id=c(1,1,1,2,2,2,2,2),
         icd9cm=c("D20206","D24220","D4439","D5064","DE8788","D40403","D1960","D1958"),
         stringsAsFactors=TRUE)
generate_comorbidity_df(cases)
# generate categories for patients in the \code{\link{vt_inp_sample}}
generate_comorbidity_df(vt_inp_sample)
# in this example, D25071 is reduced to "dm" from "dmcx" because D4439 already codes perivasc
# also, D20206 "tumor" and D1970 "mets" lead to just "mets"
# D25001 and D25040 are just "dmcx"
# D45621 and D570 are just "modliver"
cases <- data.frame(id=c(1,1,1,1,2,2,2,2),
  icd9cm=c("D1970","D20206","D25071","D4439","D25001","D25040","D45621","D570"),
  stringsAsFactors=TRUE)
generate_comorbidity_df(cases)
```

`icd9cm_charlson_deyo` *Create Deyo map of ICD-9-CM to Charlson comorbidities*

Description

Function that generates a data frame linking ICD-9-CM codes to the Charlson comorbidity categories using the Deyo mapping.

Usage

```
icd9cm_charlson_deyo(icd9)
```

Arguments

`icd9`	a unique character vector of ICD-9-CM codes

Details

NOTE: The input vector of ICD-9-CM codes must be unique, because the output dataframe uses the ICD-9-CM code as row.name.

Uses regular expressions created from the paper by Deyo in 1992.

ICD-9-CM codes must have periods removed. Diagnostic codes are prefixed with 'D' while procedure codes are prefixed with 'P'. So, diagnostic code `404.03` should be `"D40403"`.

Value

A data frame, with ICD9 codes as row names and one logical column for each comorbidity in `charlson_list`

References

1. Deyo RA, Cherkin DC, Ciol MA: Adapting a clinical comorbidity index for use with ICD-9-CM administrative databases. Journal of clinical epidemiology 1992; 45:613-9 http://www.ncbi.nlm.nih.gov/pubmed/1607900

See Also

`icd9cm_charlson_quan`, `icd9cm_charlson_romano`, `icd9cm_elixhauser_quan`, `icd9cm_elixhauser_ahrq37`, `charlson_weights`,

Examples

```
# Identify Charlson categories in ICD-9-CM listing
cases <- data.frame(id=c(1,1,1,2,2,2),
  icd9cm=c("D20206","D24220","D4439","D5064","DE8788","D40403"),
  stringsAsFactors=TRUE)
cases_with_cm <- merge(cases, icd9cm_charlson_deyo(levels(cases$icd9cm)),
  by.x="icd9cm", by.y="row.names", all.x=TRUE)

# generate crude comorbidity summary for each patient
library(plyr)
ddply(cases_with_cm, .(id),
  function(x) { data.frame(lapply(x[,3:ncol(x)], any)) })
```

---

icd9cm_charlson_quan *Create Quan map of ICD-9-CM to Charlson comorbidities*

Description

Function that creates a dataframe that links ICD-9-CM codes to the Charlson comorbidity categories using Quan's method.

Usage

```
icd9cm_charlson_quan(icd9)
```

Arguments

icd9	a unique character vector of ICD-9-CM codes

Details

NOTE: The input vector of ICD-9-CM codes must be unique, because the output dataframe uses the ICD-9-CM code as row.name.

Uses regular expressions created from the paper by Quan in 2005.

ICD-9-CM codes must have periods removed. Diagnostic codes are prefixed with 'D' while procedure codes are prefixed with 'P'. So, diagnostic code `404.03` should be `"D40403"`.

Value

A data frame, with ICD9 codes as row names and one logical column for each comorbidity in `charlson_list`

References

1. Quan H, Sundararajan V, Halfon P, Fong A, Burnand B, Luthi J-C, Saunders LD, Beck CA, Feasby TE, Ghali WA: Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data. Medical care 2005; 43:1130-9 http://www.ncbi.nlm.nih.gov/pubmed/16224307

See Also

`icd9cm_charlson_deyo`, `icd9cm_charlson_romano`, `icd9cm_charlson_quan`, `icd9cm_elixhauser_quan`

Examples

```
# Identify Charlson categories in ICD-9-CM listing
cases <- data.frame(id=c(1,1,1,2,2,2),
  icd9cm=c("D20206","D24220","D4439","D5064","DE8788","D40403"),
  stringsAsFactors=TRUE)
cases_with_cm <- merge(cases, icd9cm_charlson_quan(levels(cases$icd9cm)),
  by.x="icd9cm", by.y="row.names", all.x=TRUE)

# generate crude comorbidity summary for each patient
library(plyr)
ddply(cases_with_cm, .(id),
  function(x) { data.frame(lapply(x[,3:ncol(x)], any)) })
```

`icd9cm_charlson_romano` *Create Romano map of ICD-9-CM to Charlson comorbidities*

Description

Function that creates a dataframe which links ICD-9-CM codes to the Charlson comorbidity categories using the Romano mapping.

Usage

```
icd9cm_charlson_romano(icd9)
```

Arguments

`icd9`	a unique character vector of ICD-9-CM codes

Details

NOTE: The input vector of ICD-9-CM codes must be unique, because the output dataframe uses the ICD-9-CM code as row.name.

Uses regular expressions created from the paper by Romano in 1993.

ICD-9-CM codes must have periods removed. Diagnostic codes are prefixed with 'D' while procedure codes are prefixed with 'P'. So, diagnostic code `404.03` should be `"D40403"`.

Value

A data frame, with ICD9 codes as row names and one logical column for each comorbidity in `charlson_list`

References

1. Romano PS, Roos LL, Jollis JG: Adapting a clinical comorbidity index for use with ICD-9-CM administrative data: differing perspectives. Journal of clinical epidemiology 1993; 46:1075-9; discussion 1081-90 http://www.ncbi.nlm.nih.gov/pubmed/8410092

See Also

`icd9cm_charlson_quan`, `icd9cm_charlson_deyo`, `icd9cm_elixhauser_quan`, `icd9cm_elixhauser_ahrq37`, `charlson_weights`,

Examples

```
# Identify Charlson categories in ICD-9-CM listing
cases <- data.frame(id=c(1,1,1,2,2,2),
  icd9cm=c("D20206","D24220","D4439","D5064","DE8788","D40403"),
  stringsAsFactors=TRUE)
cases_with_cm <- merge(cases, icd9cm_charlson_romano(levels(cases$icd9cm)),
  by.x="icd9cm", by.y="row.names", all.x=TRUE)

# generate crude comorbidity summary for each patient
library(plyr)
ddply(cases_with_cm, .(id),
  function(x) { data.frame(lapply(x[,3:ncol(x)], any)) })
```

`icd9cm_elixhauser_ahrq37`

Create AHRQ v3.7 map of ICD-9-CM to Elixhauser comorbidities

Description

Function makes a dataframe that links ICD-9-CM codes to the Elixhauser comorbidity categories using the AHRQ v3.7 mapping.

Usage

```
icd9cm_elixhauser_ahrq37(icd9)
```

Arguments

`icd9`	a unique character vector of ICD-9-CM codes

Details

Uses regular expressions based on the file "comformat2012-2013.txt" from AHRQ.

The Agency for Healthcare Research and Quality (AHRQ) has developed Comorbidity Software as part of the Healthcare Cost and Utilization Project (HCUP). The software was developed to report on the comorbidity measures reported by Elixhauser (1998).

The AHRQ software has two parts, one that classifies ICD-9-CM codes by comorbidity, and another that performs heuristics to eliminate duplicate comorbidities and ignore comorbidities which are the primary reason for the hospital visit, as per the DRG.

This table is a translation of the first part of the software, the classifier, as implemented in the SAS file `Comformat2012-2013.txt`.

ICD-9-CM codes must have periods removed. Diagnostic codes are prefixed with 'D' while procedure codes are prefixed with 'P'. So, diagnostic code `404.03` should be `"D40403"`.

Value

A data frame, with ICD9 codes as row names and one logical column for each comorbidity in `elixhauser_list`

References

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

See Also

icd9cm_charlson_deyo, icd9cm_charlson_romano, icd9cm_charlson_quan, icd9cm_elixhauser_quan

Examples

```
# Identify Elixhauser categories
cases <- data.frame(id=c(1,1,1,2,2,2),
  icd9cm=c("D20206","D24220","D4439","D5064","DE8788","D40403"),
  stringsAsFactors=TRUE)
cases_with_cm <- merge(cases, icd9cm_elixhauser_ahrq37(levels(cases$icd9cm)),
  by.x="icd9cm", by.y="row.names", all.x=TRUE)

# generate crude comorbidity summary for each patient
library(plyr)
ddply(cases_with_cm, .(id),
  function(x) { data.frame(lapply(x[,3:ncol(x)], any)) })
```

`icd9cm_elixhauser_quan` *Create Quan map of ICD-9-CM to Elixhauser comorbidities*

Description

Function to make a dataframe that links ICD-9-CM codes to the Elixhauser comorbidity categories using the Quan mapping.

Usage

```
icd9cm_elixhauser_quan(icd9)
```

Arguments

`icd9`	a unique character vector of ICD-9-CM codes

Details

Uses regular expressions created from the Quan paper from 2005.

ICD-9-CM codes must have periods removed. Diagnostic codes are prefixed with 'D' while procedure codes are prefixed with 'P'. So, diagnostic code `404.03` should be `"D40403"`.

Some ICD-9-CM codes will correspond to more than one category. For example, `404.03` (Hypertensive heart and chronic kidney disease ... stage V) is in both `chf` and `renlfail` categories.

Value

A data frame, with ICD9 codes as row names and one logical column for each comorbidity in elixhauser_list

References

1. Quan H, Sundararajan V, Halfon P, Fong A, Burnand B, Luthi J-C, Saunders LD, Beck CA, Feasby TE, Ghali WA: Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data. Medical care 2005; 43:1130-9 http://www.ncbi.nlm.nih.gov/pubmed/16224307

See Also

icd9cm_charlson_deyo, icd9cm_charlson_romano, icd9cm_charlson_quan, icd9cm_elixhauser_ahrq37

Examples

```
# Identify Elixhauser categories
cases <- data.frame(id=c(1,1,1,2,2,2),
  icd9cm=c("D20206","D24220","D4439","D5064","DE8788","D40403"),
  stringsAsFactors=TRUE)
cases_with_cm <- merge(cases, icd9cm_elixhauser_quan(levels(cases$icd9cm)),
  by.x="icd9cm", by.y="row.names", all.x=TRUE)

# generate crude comorbidity summary for each patient
library(plyr)
ddply(cases_with_cm, .(id),
  function(x) { data.frame(lapply(x[,3:ncol(x)], any)) })
```

---

icd9cm_list *List of ICD-9-CM diagnostic and procedural codes*

Description

ICD-9-CM codes have the periods removed. Diagnostic codes are prefixed with 'D' while procedure codes are prefixed with 'P'. So, diagnostic code `404.03` appears as `"D40403"`.

Usage

```
icd9cm_list
```

Format

A string vector

Details

Obsolete codes not active in 2012 are not present, and may cause this dataset to miss certain classifications when applied to older datasets. For example, codes 043 and 044 (both obsolete AIDS codes) are not included.

References

1. https://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes.html

Examples

```
# Count procedural codes
length(icd9cm_list[grep('^P',icd9cm_list)])
```

`icd9cm_rcri` *Create Map of ICD-9-CM to Revised Cardiac Risk Index classes*

Description

Function to generate data frame that links ICD-9-CM codes to the RCRI comorbidity categories.

Usage

```
icd9cm_rcri(icd9)
```

Arguments

`icd9`	a unique character vector of ICD-9-CM codes

Details

Lee et al in 1999 published a "Revised Cardiac Risk Index" based on the work on Goldman in 1997. The RCRI is used to determine the major cardiac complication risk for a patient about to undergo major noncardiac surgery. The six predictors that make up the RCRI are: 1. high-risk surgery 2. history of ischemic heart disease 3. history of congestive heart failure 4. history of cerebrovascular disease 5. preoperative treatment with insulin 6. preoperative serum creatinine with Cr > 2 mg/dL.

In 2005 Boersma et al demonstrated that the Lee indexed can be adapted to use administrative data to predict cardiovascular mortality. They used the following for each point above: 1. retroperitoneal, intrathoracic, or suprainguinal vascular procedure; 2. Ischemia: ICD-9 codes 410.*, 411.*, 412.*, 413.*, 414.*; 3. CVA: ICD-9 428.*; 4. CHF: ICD-9 943.0; 5. DM: ICD-9 425.0; 6. Renal: ICD-9 958.0.

This function merges the ICD-9 guidelines used by Boersma with some of the other ICD-9 classifiers in this package. This data set uses the following for each aspect of the RCRI: 1. procedure is left to you 2. 'ischemia' as defined in Boersma 3. 'cvd' as defined by Quan in `icd9cm_charlson_quan` 4. 'chf' as defined by AHRQ in `icd9cm_elixhauser_ahrq37` 5. 'dm' as defined by AHRQ (both 'dm' and 'dmcx') 6. renlfail' as defined by AHRQ.

Value

A data frame, with ICD9 codes as row names and logical columns for `chf`, `cvd`, `dm`, `ischemia`, and `renlfail`.

References

1. Lee TH, Marcantonio ER, Mangione CM, Thomas EJ, Polanczyk CA, Cook EF, Sugarbaker DJ, Donaldson MC, Poss R, Ho KK, Ludwig LE, Pedan A, Goldman L: Derivation and prospective validation of a simple index for prediction of cardiac risk of major noncardiac surgery. Circulation 1999; 100:1043-9 http://www.ncbi.nlm.nih.gov/pubmed/10477528

2. Boersma E, Kertai MD, Schouten O, Bax JJ, Noordzij P, Steyerberg EW, Schinkel AFL, Santen M van, Simoons ML, Thomson IR, Klein J, Urk H van, Poldermans D: Perioperative cardiovascular mortality in noncardiac surgery: validation of the Lee cardiac risk index. The American journal of medicine 2005; 118:1134-41 http://www.ncbi.nlm.nih.gov/pubmed/16194645

See Also

icd9cm_charlson_quan, icd9cm_elixhauser_quan, icd9cm_elixhauser_ahrq37

`icd9cm_sessler_rsi` *Returns composite Sessler risk stratification index, given a list of ICD-9-CM codes.*

Description

ICD-9-CM codes must have periods removed. Diagnostic codes are prefixed with 'D' while procedure codes are prefixed with 'P'. So, diagnostic code `404.03` should be `"D40403"`.

Usage

```
icd9cm_sessler_rsi(icd9)
```

Arguments

`icd9`	a unique character vector of ICD-9-CM codes

Details

Note: A subsequent publication (Sigakis, 2013) found the following: "Calibration "in-the-large" for RSI in-hospitalmortality illustrated a discrepancy between actual (1.5 identified a regression constant (-2.198) in the published RSI "all-covariates.xls" file that was not used in the published SPSS implementation macro."

Value

The risk stratification index score

Author(s)

Tom Joseph <thomas.joseph@mountsinai.org>, Patrick McCormick <patrick.mccormick@mountsinai.org>

References

1. Sessler DI, Sigl JC, Manberg PJ, Kelley SD, Schubert A, Chamoun NG. Broadly applicable risk stratification system for predicting duration of hospitalization and mortality. Anesthesiology. 2010 Nov;113(5):1026-37. doi: 10.1097/ALN.0b013e3181f79a8d.

2. Sigakis MJG, Bittner EA, Wanderer JP: Validation of a risk stratification index and risk quantification index for predicting patient outcomes: in-hospital mortality, 30-day mortality, 1-year mortality, and length-of-stay. Anesthesiology 2013; 119:525-40

Examples

```
# Calculate RSI for each patient ("id") in dataframe
cases <- data.frame(id=c(1,1,1,2,2,2),
  icd9cm=c("D20206","D24220","D4439","D5064","DE8788","D40403"),
  stringsAsFactors=TRUE)
library(plyr)
ddply(cases, .(id), function(x) { icd9cm_sessler_rsi(x$icd9cm) } )
```

`melt_icd9list` *Convert ICD-9-CM code list to dataframe*

Description

`melt_icd9list` uses `ddply` to melt a column of comma-separated ICD-9-CM codes into a series of rows, one for each code.

Usage

```
melt_icd9list(
  df,
  idvar = "id",
  icd9var = "icd9cm",
  .progress = "none",
  .parallel = FALSE,
  .paropts = NULL
)
```

Arguments

`df`	a data frame with at least two columns, specified as `idvar` and `icd9var`.
`idvar`	string with name of ID variable within `df` (defaults to "id")
`icd9var`	string with name of ICD code variable within `df` (defaults to "icd9cm")
`.progress`	passed to `ddply`
`.parallel`	passed to `ddply`
`.paropts`	passed to `ddply`

Value

a dataframe with two columns, `idvar` and `"icd9cm"`

Examples

```
cases <- data.frame(id=c(1,2),
    icd9list=c('162.4,070.30,155.0,401.9','996.52,E878.8,V45.86'),
    stringsAsFactors=TRUE)
melt_icd9list(cases, "id", "icd9list")
```

`merge_icd9_dx_and_procs` *Merge ICD-9-CM diagnostic and procedure codes*

Description

Merges a dataframe containing ICD-9-CM diagostic codes with a dataframe containing ICD-9 procedure codes Diagnostic codes are prefixed with 'D', while procedure codes are prefixed with 'P'

Usage

```
merge_icd9_dx_and_procs(
  dx_df,
  proc_df,
  icd9dxvar = "icd9cm",
  icd9pvar = "icd9cm"
)
```

Arguments

`dx_df`	a data frame with at least two columns, specified as `idvar` and `icd9dxvar`, where the values are ICD-9 diagnostic codes
`proc_df`	a data frame with at least two columns, specified as `idvar` and `icd9pvar`, where the values are ICD-9 procedure codes
`icd9dxvar`	name of icd9 diagnostic code column, default "icd9cm"
`icd9pvar`	name of icd9 procedure code column, default "icd9cm"

Value

a merged dataframe with common columns and `"icd9cm"`

Examples

```
cases <- data.frame(id=c(1,2),
                    icd9dxlist=c('162.4,070.30,155.0,401.9','996.52,E878.8,V45.86'),
                    icd9plist=c('38.16','38.42'),
                    stringsAsFactors=TRUE)
dx_df <- melt_icd9list(cases, "id", "icd9dxlist")
proc_df <- melt_icd9list(cases, "id", "icd9plist")
merge_icd9_dx_and_procs(dx_df, proc_df)
```

---

rsi_beta_1yrpod *Values for calculating RSI for 1 year mortality*

Description

Values for calculating RSI for 1 year mortality

Usage

```
rsi_beta_1yrpod
```

Format

A hash (see package "hash"), where key is icd9cm code, and value is beta. Special key "popbeta" has the population beta for the entire table.

References

1. Sessler DI, Sigl JC, Manberg PJ, Kelley SD, Schubert A, Chamoun NG. Broadly applicable risk stratification system for predicting duration of hospitalization and mortality. Anesthesiology. 2010 Nov;113(5):1026-37. doi: 10.1097/ALN.0b013e3181f79a8d. http://my.clevelandclinic.org/anesthesiology/outcomes-research/risk-stratification-index.aspx

---

rsi_beta_30dlos *Values for calculating RSI for 30-day length of stay*

Description

Values for calculating RSI for 30-day length of stay

Usage

```
rsi_beta_30dlos
```

Format

A hash (see package "hash"), where key is icd9cm code, and value is beta. Special key "popbeta" has the population beta for the entire table.

References

1. Sessler DI, Sigl JC, Manberg PJ, Kelley SD, Schubert A, Chamoun NG. Broadly applicable risk stratification system for predicting duration of hospitalization and mortality. Anesthesiology. 2010 Nov;113(5):1026-37. doi: 10.1097/ALN.0b013e3181f79a8d. http://my.clevelandclinic.org/anesthesiology/outcomes-research/risk-stratification-index.aspx

---

`rsi_beta_30dpod` *Values for calculating RSI for 30-day mortality*

Description

Values for calculating RSI for 30-day mortality

Usage

```
rsi_beta_30dpod
```

Format

A hash (see package "hash"), where key is icd9cm code, and value is beta. Special key "popbeta" has the population beta for the entire table.

References

1. Sessler DI, Sigl JC, Manberg PJ, Kelley SD, Schubert A, Chamoun NG. Broadly applicable risk stratification system for predicting duration of hospitalization and mortality. Anesthesiology. 2010 Nov;113(5):1026-37. doi: 10.1097/ALN.0b013e3181f79a8d. http://my.clevelandclinic.org/anesthesiology/outcomes-research/risk-stratification-index.aspx

---

`rsi_beta_inhosp` *Values for calculating RSI for in-hospital mortality.*

Description

Values for calculating RSI for in-hospital mortality.

Usage

```
rsi_beta_inhosp
```

Format

A hash (see package "hash"), where key is icd9cm code, and value is beta. Special key "popbeta" has the population beta for the entire table.

References

1. Sessler DI, Sigl JC, Manberg PJ, Kelley SD, Schubert A, Chamoun NG. Broadly applicable risk stratification system for predicting duration of hospitalization and mortality. Anesthesiology. 2010 Nov;113(5):1026-37. doi: 10.1097/ALN.0b013e3181f79a8d. http://my.clevelandclinic.org/anesthesiology/outcomes-research/risk-stratification-index.aspx

---

rsi_sample_data *Sample data for validating RSI*

Description

Sample data for validating RSI

Usage

```
rsi_sample_data
```

Format

A data table with a patient ID and several columns with ICD-9-CM codes.

References

1. Sessler DI, Sigl JC, Manberg PJ, Kelley SD, Schubert A, Chamoun NG. Broadly applicable risk stratification system for predicting duration of hospitalization and mortality. Anesthesiology. 2010 Nov;113(5):1026-37. doi: 10.1097/ALN.0b013e3181f79a8d. http://my.clevelandclinic.org/anesthesiology/outcomes-research/risk-stratification-index.aspx

See Also

rsi_sample_results, verify_sessler_rsi

---

rsi_sample_results *Sample results for validating RSI*

Description

Sample results for validating RSI

Usage

```
rsi_sample_results
```

Format

A data table with a patient ID, principal diagnosis, principal procedure, and RSI result columns.

References

1. Sessler DI, Sigl JC, Manberg PJ, Kelley SD, Schubert A, Chamoun NG. Broadly applicable risk stratification system for predicting duration of hospitalization and mortality. Anesthesiology. 2010 Nov;113(5):1026-37. doi: 10.1097/ALN.0b013e3181f79a8d. http://my.clevelandclinic.org/anesthesiology/outcomes-research/risk-stratification-index.aspx

See Also

rsi_sample_data, verify_sessler_rsi

---

sessler_get_single_beta

Returns the covariate coefficient for a particular diagnosis or procedure code, along with the actual code that was found in the internal database of coefficients. If a child code is supplied but its parent is in the database, the coefficient for the parent will be returned, along with that parent code. For example, if D1231 is supplied but only D123 is available, D123 will be used. This is apparently how the SPSS sample code works.

Description

Returns the covariate coefficient for a particular diagnosis or procedure code, along with the actual code that was found in the internal database of coefficients. If a child code is supplied but its parent is in the database, the coefficient for the parent will be returned, along with that parent code. For example, if D1231 is supplied but only D123 is available, D123 will be used. This is apparently how the SPSS sample code works.

Usage

```
sessler_get_single_beta(code, betalist)
```

Arguments

code	A single ICD-9-CM code
betalist	One of the rsi_beta_* datasets (supplied with this package)

Value

Covariate coefficient. You must sum all of these for a given patient and then subtract the appropriate population beta (e.g. rsi_beta_1yrpod$popbeta)

Examples

```
# get coefficient for hypercholesterolemia
sessler_get_single_beta('D2720', rsi_beta_inhosp)
# Also works with extra 0 on the end
sessler_get_single_beta('D27200', rsi_beta_inhosp)
```

verify_sessler_rsi *Validates this Sessler RSI implementation against reference data*

Description

Requires that "sample data rev2.csv" and "sample results rev2.csv" be available in datasrc directory

Usage

```
verify_sessler_rsi()
```

Value

Table of patients with scores >0.001 difference.

Author(s)

Patrick McCormick <patrick.mccormick@mountsinai.org>

References

1. Sessler DI, Sigl JC, Manberg PJ, Kelley SD, Schubert A, Chamoun NG. Broadly applicable risk stratification system for predicting duration of hospitalization and mortality. Anesthesiology. 2010 Nov;113(5):1026-37. doi: 10.1097/ALN.0b013e3181f79a8d.

See Also

rsi_sample_data, rsi_sample_results

Examples

```
## Not run:
verify_sessler_rsi()

## End(Not run)
```

vt_inp_sample *First 100 patients and their ICD-9-CM codes from the Vermont Uniform Hospital Discharge Data Set for 2011, Inpatient.*

Description

Diagnostic ICD-9 codes are prefixed with 'D', while procedural ICD-9 codes are prefixed with 'P'.

Usage

```
vt_inp_sample
```

Format

A data frame, with column "id" (numeric), some descriptive columns, "dx" (factor), and "icd9cm" (factor)

References

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

See Also

icd9cm_charlson_deyo, icd9cm_charlson_quan, icd9cm_charlson_romano, icd9cm_elixhauser_quan, icd9cm_elixhauser_ahrq37

Examples

```
max(vt_inp_sample$scu_days)
```

Index

∗ **datasets**
    charlson_list, 2
    charlson_weights, 3
    charlson_weights_orig, 4
    elixhauser_list, 5
    icd9cm_list, 13
    vt_inp_sample, 22

charlson_list, 2, *8–10*
charlson_weights, 3, *4–6*, *8*, *10*
charlson_weights_orig, *3*, 4, *6*

ddply, *6*, *16*

elixhauser_list, 5, *11*, *13*

generate_charlson_index_df, 5
generate_comorbidity_df, *5*, 6, *6*

icd9cm_charlson_deyo, *3, 4*, 7, *9, 10*, *12, 13*, *23*
icd9cm_charlson_quan, *3, 4*, *6*, 8, *8–10*, *12–15*, *23*
icd9cm_charlson_romano, *3, 4*, *8, 9*, 10, *12, 13*, *23*
icd9cm_elixhauser_ahrq37, *8*, *10*, 11, *13–15*, *23*
icd9cm_elixhauser_quan, *8–10*, 12, *12*, *15*, *23*
icd9cm_list, 13
icd9cm_rcri, 14
icd9cm_sessler_rsi, 15

melt_icd9list, *3, 4*, 16
merge_icd9_dx_and_procs, 17

rsi_beta_1yrpod, 18
rsi_beta_30dlos, 18
rsi_beta_30dpod, 19
rsi_beta_inhosp, 19
rsi_sample_data, 20, *21*
rsi_sample_results, 20, *20*

sessler_get_single_beta, 21

verify_sessler_rsi, *20, 21*, 22
vt_inp_sample, 22