Package 'comorbidity'

July 16, 2024

Version 1.1.0

Title Computing Comorbidity Scores

Description Computing comorbidity indices and scores such as the weighted Charlson
score (Charlson, 1987 <doi:10.1016/0021-9681(87)90171-8>) and the Elixhauser
comorbidity score (Elixhauser, 1998 <doi:10.1097/00005650-199801000-00004>)
using ICD-9-CM or ICD-10 codes (Quan, 2005 <doi:10.1097/01.mlr.0000182534.19832.83>).
Australian and Swedish modifications of the Charlson Comorbidity Index are available
as well (Sundararajan, 2004 <doi:10.1016/j.jclinepi.2004.03.012> and Ludvigsson, 2021
<doi:10.2147/CLEP.S282475>), together with different weighting algorithms for both
the Charlson and Elixhauser comorbidity scores.

URL https://ellessenne.github.io/comorbidity/,
https://github.com/ellessenne/comorbidity

BugReports https://github.com/ellessenne/comorbidity/issues

License GPL (>= 3)

Depends R (>= 2.10)

Imports checkmate, data.table, stats, stringi, utils

Suggests covr, knitr, rmarkdown, testthat

RoxygenNote 7.3.2

Encoding UTF-8

LazyData true

ByteCompile true

VignetteBuilder knitr

Language en-GB

NeedsCompilation no

Author Alessandro Gasparini [aut, cre]
(<https://orcid.org/0000-0002-8319-7624>),
Hojjat Salmasian [ctb] (ICD-9-CM scores),
Jonathan Williman [ctb] (<https://orcid.org/0000-0001-5080-4435>),
Sing Yi Chia [ctb] (<https://orcid.org/0000-0002-7591-3141>),
Edmund Teo [ctb] (<https://orcid.org/0000-0003-3936-4082>),
Desi Quintans [ctb]

Maintainer Alessandro Gasparini <alessandro@ellessenne.xyz>

Repository CRAN

Date/Publication 2024-07-16 06:40:02 UTC

Contents

---

australia10	*Australian mortality data, 2010*

Description

A dataset containing Australian mortality data, obtained from Stata 17.

Usage

```
australia10
```

Format

A data frame with 3,322 rows and 3 variables:

cause ICD-10 code representing cause of death

sex Gender

deaths Number of deaths

Note

The R code used to download and process the dataset from Stata is available here.

---

available_algorithms   *Display Currently Supported Algorithms*

Description

This function prints all (currently) supported and implemented comorbidity mapping, and for each one of those, each supported scoring and weighting algorithm.

Usage

```
available_algorithms()
```

Examples

```
available_algorithms()
```

---

comorbidity   *Comorbidity mapping.*

Description

Maps comorbidity conditions using algorithms from the Charlson and the Elixhauser comorbidity scores.

Usage

```
comorbidity(x, id, code, map, assign0, labelled = TRUE, tidy.codes = TRUE)
```

Arguments

x	A tidy `data.frame` (or a `data.table`; `tibbles` are supported too) with one column containing an individual ID and a column containing all diagnostic codes. Extra columns other than ID and codes are discarded. Column names must be syntactically valid names, otherwise they are forced to be so by calling the `make.names()` function.
id	String denoting the name of a column of x containing the individual ID.
code	String denoting the name of a column of x containing diagnostic codes. Codes must be in upper case with no punctuation in order to be properly recognised.
map	String denoting the mapping algorithm to be used (values are case-insensitive). Possible values are the Charlson score with either ICD-10 or ICD-9-CM codes (`charlson_icd10_quan`, `charlson_icd9_quan`) and the Elixhauser score, again using either ICD-10 or ICD-9-CM (`elixhauser_icd10_quan`, `elixhauser_icd9_quan`). These mapping are based on the paper by Quan et al. (2011). It is also possible to obtain a Swedish (`charlson_icd10_se`) or Australian (`charlson_icd10_am`) modification of the Charlson score using ICD-10 codes.

assign0   Logical value denoting whether to apply a hierarchy of comorbidities: should a
                   comorbidity be present in a patient with different degrees of severity, then the
                   milder form will be assigned a value of 0. By doing this, a type of comorbidity
                   is not counted more than once in each patient. If assign0 = TRUE, the comor-
                   bidities that are affected by this argument are:

- "Mild liver disease" (mld) and "Moderate/severe liver disease" (msld) for
  the Charlson score;
- "Diabetes" (diab) and "Diabetes with complications" (diabwc) for the Charl-
  son score;
- "Cancer" (canc) and "Metastatic solid tumour" (metacanc) for the Charl-
  son score;
- "Hypertension, uncomplicated" (hypunc) and "Hypertension, complicated"
  (hypc) for the Elixhauser score;
- "Diabetes, uncomplicated" (diabunc) and "Diabetes, complicated" (diabc)
  for the Elixhauser score;
- "Solid tumour" (solidtum) and "Metastatic cancer" (metacanc) for the
  Elixhauser score.

labelled   Logical value denoting whether to attach labels to each comorbidity, which are
                   compatible with the RStudio viewer via the utils::View() function. Defaults
                   to TRUE.

tidy.codes   Logical value, defaulting to TRUE, denoting whether ICD codes are to be tidied.
                   If TRUE, all codes are converted to upper case and all non-alphanumeric charac-
                   ters are removed using the regular expression [^[:alnum:]]. It can be set to
                   FALSE to speed up computations, but please be aware that in that case codes are
                   assumed to be formatted as above. If codes are incorrectly formatted, this may
                   lead to wrong results: use at your own risk!

Details

The ICD-10 and ICD-9-CM coding for the Charlson and Elixhauser scores is based on work by
Quan *et al.* (2005). ICD-10 and ICD-9 codes must be in upper case and with alphanumeric char-
acters only in order to be properly recognised; set tidy.codes = TRUE to properly tidy the codes
automatically (this is the default behaviour). A message is printed to the R console when non-
alphanumeric characters are found.

Value

A data frame with id and columns relative to each comorbidity domain, with one row per individual.

For the Charlson score, the following variables are included in the dataset:

- The id variable as defined by the user;
- mi, for myocardial infarction;
- chf, for congestive heart failure;
- pvd, for peripheral vascular disease;
- cevd, for cerebrovascular disease;
- dementia, for dementia;

- `cpd`, for chronic pulmonary disease;
- `rheumd`, for rheumatoid disease;
- `pud`, for peptic ulcer disease;
- `mld`, for mild liver disease;
- `diab`, for diabetes without complications;
- `diabwc`, for diabetes with complications;
- `hp`, for hemiplegia or paraplegia;
- `rend`, for renal disease;
- `canc`, for cancer (any malignancy);
- `msld`, for moderate or severe liver disease;
- `metacanc`, for metastatic solid tumour;
- `aids`, for AIDS/HIV. Please note that we combine "chronic obstructive pulmonary disease" and "chronic other pulmonary disease" for the Swedish version of the Charlson index, for comparability (and compatibility) with other definitions/implementations.

Conversely, for the Elixhauser score the dataset contains the following variables:

- The `id` variable as defined by the user;
- `chf`, for congestive heart failure;
- `carit`, for cardiac arrhythmias;
- `valv`, for valvular disease;
- `pcd`, for pulmonary circulation disorders;
- `pvd`, for peripheral vascular disorders;
- `hypunc`, for hypertension, uncomplicated;
- `hypc`, for hypertension, complicated;
- `para`, for paralysis;
- `ond`, for other neurological disorders;
- `cpd`, for chronic pulmonary disease;
- `diabunc`, for diabetes, uncomplicated;
- `diabc`, for diabetes, complicated;
- `hypothy`, for hypothyroidism;
- `rf`, for renal failure;
- `ld`, for liver disease;
- `pud`, for peptic ulcer disease, excluding bleeding;
- `aids`, for AIDS/HIV;
- `lymph`, for lymphoma;
- `metacanc`, for metastatic cancer;
- `solidtum`, for solid tumour, without metastasis;
- `rheumd`, for rheumatoid arthritis/collaged vascular disease;

- coag, for coagulopathy;
- obes, for obesity;
- wloss, for weight loss;
- fed, for fluid and electrolyte disorders;
- blane, for blood loss anaemia;
- dane, for deficiency anaemia;
- alcohol, for alcohol abuse;
- drug, for drug abuse;
- psycho, for psychoses;
- depre, for depression;

Labels are presented to the user when using the RStudio viewer (e.g. via the utils::View() function) for convenience, if labelled = TRUE.

References

Quan H, Sundararajan V, Halfon P, Fong A, Burnand B, Luthi JC, et al. *Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data*. Medical Care 2005; 43(11):1130-1139.

Charlson ME, Pompei P, Ales KL, et al. *A new method of classifying prognostic comorbidity in longitudinal studies: development and validation*. Journal of Chronic Diseases 1987; 40:373-383.

Ludvigsson JF, Appelros P, Askling J et al. *Adaptation of the Charlson Comorbidity Index for register-based research in Sweden*. Clinical Epidemiology 2021; 13:21-41.

Sundararajan V, Henderson T, Perry C, Muggivan A, Quan H, Ghali WA. *New ICD-10 version of the Charlson comorbidity index predicted in-hospital mortality*. Journal of Clinical Epidemiology 2004; 57(12):1288-1294.

Examples

```
set.seed(1)
x <- data.frame(
  id = sample(1:15, size = 200, replace = TRUE),
  code = sample_diag(200),
  stringsAsFactors = FALSE
)

# Charlson score based on ICD-10 diagnostic codes:
comorbidity(x = x, id = "id", code = "code", map = "charlson_icd10_quan", assign0 = FALSE)

# Elixhauser score based on ICD-10 diagnostic codes:
comorbidity(x = x, id = "id", code = "code", map = "elixhauser_icd10_quan", assign0 = FALSE)

# The following example describes how the `assign0` argument works.
# We create a dataset for a single patient with two codes, one for
# uncomplicated diabetes ("E100") and one for complicated diabetes
# ("E102"):
x2 <- data.frame(
```

```
  id = 1,
  code = c("E100", "E102"),
  stringsAsFactors = FALSE
)
# Then, we calculate the Quan-ICD10 Charlson score:
ccF <- comorbidity(x = x2, id = "id", code = "code", map = "charlson_icd10_quan", assign0 = FALSE)
# With `assign0 = FALSE`, both diabetes comorbidities are counted:
ccF[, c("diab", "diabwc")]
# Conversely, with `assign0 = TRUE`, only the more severe diabetes with
# complications is counted:
ccT <- comorbidity(x = x2, id = "id", code = "code", map = "charlson_icd10_quan", assign0 = TRUE)
ccT[, c("diab", "diabwc")]
```

---

icd10cm_2017 *ICD-10-CM Diagnostic Codes, 2017 Version*

Description

A dataset containing the 2017 version of the ICD10-CM coding system.

Usage

```
icd10cm_2017
```

Format

A data frame with 71,486 rows and 2 variables:

Code ICD-10-CM diagnostic code

Description Description of each code

Note

The R code used to download and process the dataset from the CDC website is available here.

---

icd10cm_2018 *ICD-10-CM Diagnostic Codes, 2018 Version*

Description

A dataset containing the 2018 version of the ICD10-CM coding system.

Usage

```
icd10cm_2018
```

Format

A data frame with 71,704 rows and 2 variables:

Code ICD-10-CM diagnostic code

Description Description of each code

Note

The R code used to download and process the dataset from the CDC website is available here.

---

icd10cm_2022   *ICD-10-CM Diagnostic Codes, 2022 Version*

Description

A dataset containing the 2022 version of the ICD10-CM coding system.

Usage

```
icd10cm_2022
```

Format

A data frame with 72,750 rows and 2 variables:

Code ICD-10-CM diagnostic code

Description Description of each code

Note

The R code used to download and process the dataset from the CDC website is available here.

---

icd10_2009   *ICD-10 Diagnostic Codes, 2009 Version*

Description

A dataset containing the 2009 version of the ICD-10 codes.

Usage

```
icd10_2009
```

Format

A data frame with 10,817 rows and 4 variables:

Code ICD-10 diagnostic code

Code.clean ICD-10 diagnostic code, removing all punctuation

ICD.title Code description, in plain English.

Status Additional information, if available.

Note

The R code used to download and process the dataset from the CDC website is available here.

Source

CDC Website: https://goo.gl/6e2mvb

---

icd10_2011   *ICD-10 Diagnostic Codes, 2011 Version*

Description

A dataset containing the 2011 version of the ICD-10 codes.

Usage

```
icd10_2011
```

Format

A data frame with 10,856 rows and 4 variables:

Code ICD-10 diagnostic code

Code.clean ICD-10 diagnostic code, removing all punctuation

ICD.title Code description, in plain English.

Status Additional information, if available.

Note

The R code used to download and process the dataset from the CDC website is available here.

Source

CDC Website: https://goo.gl/rcTJJ2

---

icd9_2015 *ICD-9 Diagnostic Codes, 2015 Version (v32)*

Description

A dataset containing the version of the ICD-9 codes effective October 1, 2014.

Usage

```
icd9_2015
```

Format

A data frame with 14,567 rows and 3 variables:

Code ICD-9 diagnostic code

Long_description Long description of each code

Short_description Short description of each code

Note

The R code used to download and process the dataset from the CMS.gov website is available here.

Source

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

---

nhds2010 *Adult same-day discharges, 2010*

Description

A dataset containing adult same-day discharges from 2010, obtained from Stata 17.

Usage

```
nhds2010
```

Format

A data frame with 2,210 rows and 15 variables:

ageu Units for age

age Age

sex Sex

race Race

month Discharge month

status Discharge status

region Region

atype Type of admission

dx1 Diagnosis 1, ICD9-CM

dx2 Diagnosis 2, ICD9-CM

dx3 Diagnosis 3, ICD9-CM, imported incorrectly

dx3corr Diagnosis 3, ICD9-CM, corrected

pr1 Procedure 1

wgt Frequency weight

recid Order of record (raw data)

Note

The R code used to download and process the dataset from Stata is available here.

---

sample_diag *Simulate ICD-10 and ICD-9 diagnostic codes*

Description

A simple function to simulate ICD-10 and ICD-9 diagnostic codes at random.

Usage

```
sample_diag(n = 1, version = "ICD10_2011")
```

Arguments

n Number of ICD codes to simulate.

version The version of the ICD coding scheme to use. Possible choices are ICD10_2009,
ICD10_2011, and ICD9_2015; defaults to ICD10_2011. See icd10_2009, icd10_2011,
and icd9_2015 for further information on the different schemes.

Value

A vector of n ICD diagnostic codes.

Examples

```
# Simulate 10 ICD-10 codes
sample_diag(10)

# Simulate a tidy dataset with 15 individuals and 200 rows
set.seed(1)
x <- data.frame(
  id = sample(1:15, size = 200, replace = TRUE),
  code = sample_diag(n = 200),
  stringsAsFactors = FALSE
)
head(x)
```

---

score   *Compute (weighted) comorbidity scores*

Description

Compute (weighted) comorbidity scores

Usage

```
score(x, weights = NULL, assign0)
```

Arguments

x	An object of class comorbidty returned by a call to the comorbidity() function.
weights	A string denoting the weighting system to be used, which will depend on the mapping algorithm. Possible values for the Charlson index are: • charlson, for the original weights by Charlson et al. (1987); • quan, for the revised weights by Quan et al. (2011). Possible values for the Elixhauser score are: • vw, for the weights by van Walraven et al. (2009); • swiss, for the Swiss Elixhauser weights by Sharma et al. (2021). Defaults to NULL, in which case an unweighted score will be used.
assign0	A logical value denoting whether to apply a hierarchy of comorbidities: should a comorbidity be present in a patient with different degrees of severity, then the milder form will be assigned a value of 0 when calculating the score. By doing this, a type of comorbidity is not counted more than once in each patient. If assign0 = TRUE, the comorbidities that are affected by this argument are:

- "Mild liver disease" (`mld`) and "Moderate/severe liver disease" (`msld`) for the Charlson score;
- "Diabetes" (`diab`) and "Diabetes with complications" (`diabwc`) for the Charlson score;
- "Cancer" (`canc`) and "Metastatic solid tumour" (`metacanc`) for the Charlson score;
- "Hypertension, uncomplicated" (`hypunc`) and "Hypertension, complicated" (`hypc`) for the Elixhauser score;
- "Diabetes, uncomplicated" (`diabunc`) and "Diabetes, complicated" (`diabc`) for the Elixhauser score;
- "Solid tumour" (`solidtum`) and "Metastatic cancer" (`metacanc`) for the Elixhauser score.

Value

A numeric vector with the (possibly weighted) comorbidity score for each subject from the input dataset.

References

Charlson ME, Pompei P, Ales KL, et al. *A new method of classifying prognostic comorbidity in longitudinal studies: development and validation*. Journal of Chronic Diseases 1987; 40:373-383.

Quan H, Li B, Couris CM, et al. *Updating and validating the Charlson Comorbidity Index and Score for risk adjustment in hospital discharge abstracts using data from 6 countries*. American Journal of Epidemiology 2011; 173(6):676-682.

van Walraven C, Austin PC, Jennings A, Quan H and Forster AJ. *A modification of the Elixhauser comorbidity measures into a point system for hospital death using administrative data*. Medical Care 2009; 47(6):626-633.

Sharma N, Schwendimann R, Endrich O, et al. *Comparing Charlson and Elixhauser comorbidity indices with different weightings to predict in-hospital mortality: an analysis of national inpatient data*. BMC Health Services Research 2021; 21(13).

Examples

```
set.seed(1)
x <- data.frame(
  id = sample(1:15, size = 200, replace = TRUE),
  code = sample_diag(200),
  stringsAsFactors = FALSE
)

# Charlson score based on ICD-10 diagnostic codes:
x1 <- comorbidity(x = x, id = "id", code = "code", map = "charlson_icd10_quan", assign0 = FALSE)
score(x = x1, weights = "charlson", assign0 = FALSE)

# Elixhauser score based on ICD-10 diagnostic codes:
x2 <- comorbidity(x = x, id = "id", code = "code", map = "elixhauser_icd10_quan", assign0 = FALSE)
score(x = x2, weights = "vw", assign0 = FALSE)
```

```
# Checking the `assign0` argument.
# Please make sure to check the example in the documentation of the
# `comorbidity()` function first, with ?comorbidity().
# We use the same dataset for a single subject with two codes, for
# complicated and uncomplicated diabetes:
x3 <- data.frame(
  id = 1,
  code = c("E100", "E102"),
  stringsAsFactors = FALSE
)
# Then, we calculate the Quan-ICD10 Charlson score:
ccF <- comorbidity(x = x3, id = "id", code = "code", map = "charlson_icd10_quan", assign0 = FALSE)
ccF[, c("diab", "diabwc")]
# If we calculate the unweighted score with `assign0 = FALSE`, both diabetes
# conditions are counted:
score(x = ccF, assign0 = FALSE)
# Conversely, with `assign0 = TRUE`, only the most severe is considered:
score(x = ccF, assign0 = TRUE)
```

Index