Package 'multimorbidity'

October 13, 2022

Title Harmonizing Various Comorbidity, Multimorbidity, and Frailty Measures

Version 0.5.0

Description Identifying comorbidities, frailty, and multimorbidity in claims and administrative data is often a duplicative process.
The functions contained in this package are meant to first prepare the data to a format acceptable by all other packages, then provide a uniform and simple approach to generate comorbidity and multimorbidity metrics based on these claims data. The package is ever evolving to include new metrics, and is always looking for new measures to include.
The citations used in this package include the following publications:
Anne Elixhauser, Claudia Steiner, D. Robert Harris, Rosanna M. Coffey (1998) <doi:10.1097/00005650-199801000-00004>,
Brian J Moore, Susan White, Raynard Washington, et al. (2017) <doi:10.1097/MLR.0000000000000735>,
Mary E. Charlson, Peter Pompei, Kathy L. Ales, C. Ronald MacKenzie (1987) <doi:10.1016/0021-9681(87)90171-8>,
Richard A. Deyo, Daniel C. Cherkin, Marcia A. Ciol (1992) <doi:10.1016/0895-4356(92)90133-8>,
Hude Quan, Vijaya Sundararajan, Patricia Halfon, et al. (2005) <doi:10.1097/01.mlr.0000182534.19832.83>,
Dae Hyun Kim, Sebastian Schneeweiss, Robert J Glynn, et al. (2018) <doi:10.1093/gerona/glx229>,
Melissa Y Wei, David Ratz, Kenneth J Mukamal (2020) <doi:10.1111/jgs.16310>,
Kathryn Nicholson, Amanda L. Terry, Martin Fortin, et al. (2015) <doi:10.15256/joc.2015.5.61>,
Martin Fortin, José Almirall, and Kathryn Nicholson (2017)<doi:10.15256/joc.2017.7.122>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports tidyverse, dplyr, lubridate, magrittr, stringr, rlang, tidyselect, tidyr, devtools, sqldf, stats

URL https://github.com/WYATTBENSKEN/multimorbidity

BugReports https://github.com/WYATTBENSKEN/multimorbidity/issues

Suggests rmarkdown, knitr

VignetteBuilder knitr

Depends R (>= 3.5.0)

NeedsCompilation no

Author Wyatt Bensken [aut, cre] (<https://orcid.org/0000-0002-2597-9732>)

Maintainer Wyatt Bensken <wpb27@case.edu>

Repository CRAN

Date/Publication 2021-08-20 12:40:05 UTC

R topics documented:

`cfi` *Claims-based Frailty Index (CFI)*

Description

`cfi` returns a summary dataset containing the deficit-accumulation frailty index for each patient.

Usage

```
cfi(
  dat = NULL,
  id = NULL,
  dx = "dx",
  version = 19,
  version_var = NULL,
  hcpcs = "yes"
)
```

Arguments

dat	dataset which has been properly prepared using 'prepare_data()'
id	variable of the unique patient identifier
dx	the column with the diagnoses and procedures (defaults to 'dx')
version	which version(s) of ICD your data contain (ICD-9 only: 9, ICD-10 only: 10, Both: 19)
version_var	variable which denotes if the diagnoses on that row are ICD-9 (9) or ICD-10 (10)
hcpcs	whether or not HCPCS variables are included ("yes" or "no", where "yes" is the default)

Details

This function uses data which has been properly prepared to calculate the claims-based frailty index (CFI) developed by Kim et al. for each patient. As this algorithm was never developed to require two diagnosis codes, and is weighted, we have excluded that feature from this function. See full package documentation for additional details. This function is based largely on the code available via the Harvard Dataverse.

Value

dataframe with one row per patient, and a column for their patient id and a column with their frailty index.

Examples

```
cfi(dat = prepared_data, id = patient_id, dx = dx, version = 19, version_var = version)
```

---

cfi_dx10lookup *CFI ICD-10 Lookup.*

Description

A lookup dataset for CFI ICD-10.

Usage

```
data(cfi_dx10lookup)
```

Format

An object of class `data.frame` with 44807 rows and 2 columns.

Source

This was created by Kim et al.

cfi_dx9lookup *CFI ICD-9 Lookup.*

Description

A lookup dataset for CFI ICD-9.

Usage

```
data(cfi_dx9lookup)
```

Format

An object of class `data.frame` with 107 rows and 3 columns.

Source

This was created by Kim et al.

cfi_pxlookup *CFI Procedure Codes Lookup.*

Description

A lookup dataset for CFI Procedure Codes.

Usage

```
data(cfi_pxlookup)
```

Format

An object of class `data.frame` with 90 rows and 3 columns.

Source

This was created by Kim et al.

cfi_weightlookup *CFI Procedure Codes Lookup.*

Description

A lookup dataset for CFI weights.

Usage

```
data(cfi_weightlookup)
```

Format

An object of class `data.frame` with 93 rows and 2 columns.

Source

This was created by Kim et al.

charlson *Charlson Comorbidities*

Description

`charlson` returns a summary dataset containing the Charlson comorbidities for each patient.

Usage

```
charlson(
  dat = NULL,
  id = NULL,
  dx = "dx",
  version = 19,
  version_var = NULL,
  outpatient_two = "no"
)
```

Arguments

`dat`	dataset which has been properly prepared using 'prepare_data()'
`id`	variable of the unique patient identifier
`dx`	the column with the diagnoses (defaults to 'dx')
`version`	which version(s) of ICD your data contain (ICD-9 only: 9, ICD-10 only: 10, Both: 19)

version_var	variable which denotes if the diagnoses on that row are ICD-9 (9) or ICD-10 (10)
outpatient_two	whether or not it should be required for there to be two outpatient claims for a diagnosis for a patient to be positively coded with that diagnosis.

Details

This function uses data which has been properly prepared to identify and flag the Charlson comorbidities. See full package documentation for additional details.

Value

dataframe with one row per patient, and a column for their patient id, a column with each Charlson comorbidity, and a column with their Charlson score

Examples

```
charlson(dat = prepared_data, id = patient_id, dx = dx, version = 19,
version_var = version, outpatient_two = "yes")
```

---

comorbidity_window *Limit our comorbidities / multimorbidity measures to a specific time window.*

Description

comorbidity_window returns a dataset of claims which fall within a specific timeframe.

Usage

```
comorbidity_window(
  dat = NULL,
  id_dat = NULL,
  id = NULL,
  id_date = NULL,
  claims_date = NULL,
  time_pre = Inf,
  time_post = Inf
)
```

Arguments

dat	dataset
id_dat	dataset with our other identifying variables, this should be 1 row per person
id	ID variable which will be used to match and merge

id_date	name of the date of interest from the identification dataset, for example a date of diagnosis
claims_date	name for the variable in the claims data (dat) which is the date of the claim
time_pre	number to limit how many days, pre diagnosis, should be included. Default will be infinity (all claims)
time_post	similar to time_pre, but this will be after the date of interest

Details

This function takes prepared data, using the 'prepare_data' function, along with an identification dataset to limit the claims of interest to a specific time window.

Value

dataframe with which has limited the claims to a specific window

Examples

```
comorbidity_window(id_dat = id, dat = prepared_data, id = patient_id,
id_date = date_of_interest9, claims_date = claim_date, time_pre = 60)
```

---

elixhauser *Elixhauser Comorbidities*

Description

`elixhauser` returns a summary dataset containing the Elixhauser comorbidities for each patient.

Usage

```
elixhauser(
  dat = NULL,
  id = NULL,
  dx = "dx",
  version = 19,
  version_var = NULL,
  outpatient_two = "no"
)
```

Arguments

dat	dataset which has been properly prepared using 'prepare_data()'
id	variable of the unique patient identifier
dx	the column with the diagnoses (defaults to 'dx')

version	which version(s) of ICD your data contain (ICD-9 only: 9, ICD-10 only: 10, Both: 19)
version_var	variable which denotes if the diagnoses on that row are ICD-9 (9) or ICD-10 (10)
outpatient_two	whether or not it should be required for there to be two outpatient claims for a diagnosis for a patient to be positively coded with that diagnosis.

Details

This function uses data which has been properly prepared to identify and flag the Elixhauser comorbidities. See full package documentation for additional details.

Value

dataframe with one row per patient, and a column for their patient id, a column with each Elixhauser comorbidity, and a column with their Elixhauser index for readmission and death

Examples

```
elixhauser(dat = prepared_data, id = patient_id, dx = dx, version = 19,
version_var = version, outpatient_two = "yes")
```

---

i9_i10_comb *Example diagnosis data.*

Description

A dataset with fake patient data for 5 patients, with both inpatient and outpatient data, as well as HCPCS codes, and ICD9 and ICD10.

Usage

```
data(i9_i10_comb)
```

Format

A data frame with 58 rows and 11 variables:

patient_id patient_id

sex patient's sex (male or female)

date_of_serv the date of service for the fake claim

visit_type inpatient (ip) or outpatient(ot)

dx1 first diagnosis

dx2 second diagnosis

dx3 third diagnosis

dx4 fourth diagnosis

dx5 fifth diagnosis

hcpcs HCPCS code

icd_version Which version of ICD the row is. 9 = ICD-9, 0 = ICD-10

Source

This was created by the package author.

`id` *Example ID data.*

Description

A dataset with fake patient data, to match the diagnoses, that includes a date of interest to demonstrate how we can attach these dates and then subset the data to a specific time window around the date of interest.

Usage

```
data(id)
```

Format

A data frame with 5 rows and 3 variables:

patient_id patient_id

date_of_interest10 the date of interest, if you were to use only ICD-10 data

date_of_interest9 The date of interest, if you were to use only ICD-9 data

Source

This was created by the package author.

mwi *Multimorbidity Weighted Index (MWI)*

Description

`cfi` returns a summary dataset containing the multimorbidity weighted index for each patient.

Usage

```
mwi(dat = NULL, id = NULL, dx = "dx", version = 19, version_var = NULL)
```

Arguments

`dat`	dataset which has been properly prepared using 'prepare_data()'
`id`	variable of the unique patient identifier
`dx`	the column with the diagnoses and procedures (defaults to 'dx')
`version`	which version(s) of ICD your data contain (ICD-9 only: 9, ICD-10 only: 10, Both: 19)
`version_var`	variable which denotes if the diagnoses on that row are ICD-9 (9) or ICD-10 (10)

Details

This function uses data which has been properly prepared to calculate the multimorbidity weighted index developed by Wei et al. As this algorithm was never developed to require two diagnosis codes, and is weighted, we have excluded that feature from this function. See full package documentation for additional details.

Examples

```
mwi(dat = prepared_data, id = patient_id, dx = dx, version = 9, version_var = version)
```

nicholsonfortin *Nicholson and Fortin Conditions*

Description

`elixhauser` returns a summary dataset containing the Nicholson and Fortin Conditions for each patient.

Usage

```
nicholsonfortin(
  dat = NULL,
  id = NULL,
  dx = "dx",
  version = 19,
  version_var = NULL,
  outpatient_two = "no"
)
```

Arguments

dat	dataset which has been properly prepared using 'prepare_data()'
id	variable of the unique patient identifier
dx	the column with the diagnoses (defaults to 'dx')
version	which version(s) of ICD your data contain (ICD-9 only: 9, ICD-10 only: 10, Both: 19)
version_var	variable which denotes if the diagnoses on that row are ICD-9 (9) or ICD-10 (10)
outpatient_two	whether or not it should be required for there to be two outpatient claims for a diagnosis for a patient to be positively coded with that diagnosis.

Details

This function uses data which has been properly prepared to identify and flag the Nicholson and Fortin conditions See full package documentation for additional details.

Value

dataframe with one row per patient, and a column for their patient id, a column with each Nicholson/Fortin comorbidity

Examples

```
nicholsonfortin(dat = prepared_data, id = patient_id, dx = dx, version = 19,
version_var = version, outpatient_two = "yes")
```

prepared_data *Prepared dataset*

Description

A dataset which has been prepared using the prepare_data function in this package.

Usage

```
data(prepared_data)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 242 rows and 5 columns.

Source

Built using the packages in this code.

prepare_data *Prepare our claims data for analysis*

Description

`prepare_data` returns a dataset which has been transformed and prepared for subsequent functions in this package.

Usage

```
prepare_data(
  dat = NULL,
  style = "long",
  id = NULL,
  prefix_dx = "dx",
  hcpcs = "no",
  prefix_hcpcs,
  version_var,
  type_name,
  date
)
```

Arguments

dat	dataset
style	long, the default, is one diagnosis column per row whereas wide is multiple diagnosis columns
id	unique patient identifier variable name
prefix_dx	the variable prefix for the diagnosis columns (defaults to "dx"), in quotes
hcpcs	whether or not HCPCS variables are included ("yes" or "no", where "no" is the default)
prefix_hcpcs	if HCPCS are included, the variable prefix in quotes
version_var	variable which denotes if the diagnoses on that row are ICD-9 (9) or ICD-10 (10)
type_name	variable to denote if the claim is inpatient (ip) or outpatient (ot)
date	variable with the date of the claim

Details

This function takes our raw claims data, in a number of different forms, and prepares it in a way which allows the other functions in this package to easily work with it. It is recommended to run this package on all data regardless of setup.

Value

dataframe with multiple rows per patient, which has re-structured their claims

Examples

```
prepare_data(dat = i9_i10_comb, id = patient_id, style = "wide",
prefix_dx = "dx", hcpcs = "yes", prefix_hcpcs = "hcpcs", version_var = icd_version,
type_name = visit_type, date = date_of_serv)
```

Index