

# Package: LABTNSCPSS (via r-universe)

May 17, 2026

**Version** 1.0

**Title** Calculation of Comorbidity and Frailty Scores

**Description** Computes comorbidity indices and combined scores for different versions of ICD, including ICD-10-CA, ICD-10-CM, and ICD-11.

**URL** <https://github.com/bayaniazadeh/LabTNSCPSSPackage>

**License** GPL-3

**Depends** R (>= 4.3.0)

**Imports** dplyr, tibble

**Suggests** testthat, roxygen2

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Encoding** UTF-8

**LazyData** true

**Repository** <https://bayaniazadeh.r-universe.dev>

**Date/Publication** 2026-02-28 15:00:37 UTC

**RemoteUrl** <https://github.com/bayaniazadeh/labtnscpspackage>

**RemoteRef** HEAD

**RemoteSha** 0b23e69cea8cda51b493ec4cedfecab05302f61d

## Contents

australia10 . . . . .	2
available_algorithms . . . . .	3
Chronic_Pathologies.R . . . . .	3
comorbidity . . . . .	4
Comorbidity_calculation . . . . .	7
Comorbidity_Frailty_Calculation . . . . .	7
create_Data . . . . .	7
Frailty_Calculation . . . . .	8

Frailty_score . . . . .	9
icd10_2009 . . . . .	9
icd10_2011 . . . . .	10
icd10cm_2017 . . . . .	11
icd10cm_2018 . . . . .	11
icd10cm_2022 . . . . .	12
icd9_2015 . . . . .	12
LABTNSCPSS . . . . .	13
LabTNSCPSSPackage . . . . .	13
nhds2010 . . . . .	14
Pathologies_chronic.R . . . . .	15
sample_diag . . . . .	15
score . . . . .	16
<b>Index</b>	<b>19</b>

---

australia10	<i>Australian mortality data, 2010</i>
-------------	--

---

## 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](<https://raw.githubusercontent.com/ellessen/australia10/raw/make-data.R>).

---

`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()
```

---

`Chronic_Pathologies.R` *Propagating Chronic Pathologies*

---

**Description**

This script processes patient episode data, identifies ICD codes, assigns chronic categories, propagates relevant codes, and extracts basal codes. It follows a structured approach:

Load and Preprocess Data

1. Assign Chronic Categories to ICD Codes
3. Identify and Propagate Category 2 Codes
4. Identify and Propagate Category 1 Codes
5. Extract Basal Codes for Category 1 ICDs

**Note**

Ensure that:

Dates are automatically converted to YYYY-MM-DD format.

- Missing category assignments are labeled as None.

**Examples**

```
# To run the script:  
source('./R/Pathologie_chronic.R')
```

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'; 'tibble's 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', 'charlson_icd10_ca', 'charlson_icd10_ca' for Canadian version) and the Elixhauser score, again using either ICD-10 or ICD-9-CM ('elixhauser_icd10_quan', 'elixhauser_icd9_quan', 'elixhauser_icd10_ca' or Canadian version). 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 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" ('hycp') 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 characters 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 characters 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; \* 'hycp', 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")]
```

---

`Comorbidity_calculation`*Compute Charlson, Elixhauser and combined comorbidity scores*

---

**Description**

This script processes patient data containing ICD-10 codes to compute comorbidity scores using the Elixhauser and Charlson methods. The calculated scores are then used to create a coincidence matrix and visualize comorbidity relationships.

**Examples**

```
# Process and calculate frailty scores
source("R/Comorbidity_calculation.R")
```

---

`Comorbidity_Frailty_Calculation`*Compute Charlson, Elixhauser and Combined Comorbidity Scores, Including Frailty Score*

---

**Description**

This function processes patient data containing ICD-10CA codes to compute comorbidity scores using the Elixhauser and Charlson methods. The calculated scores are then used to create a coincidence matrix and visualize comorbidity relationships. The function also calculates frailty scores for patients.

**Examples**

```
# Process and calculate frailty scores
source("R/Comorbidity_Frailty_Calculation.R")
```

---

`create_Data`*Create Data*

---

**Description**

This script reads raw data from a CSV file, processes it by renaming columns based on a provided mapping, performs date transformations, cleans ICD codes, and saves the cleaned dataset to a new CSV file.

## Details

The script assumes that the input CSV file has specific columns that are mapped to standard names using the 'col\_mapping' list. After reading the data, it performs the following transformations:

- Renames columns based on the 'col\_mapping'. - Parses and converts the 'start\_date' column to a proper Date format. - Cleans up the 'ICD' codes by removing periods.

The resulting cleaned dataset is then written to 'LABTNSCPSS\_Data/input\_data\_cleaned.csv'.

## Note

This script is intended to be sourced and executed directly. It does not return a value but saves the processed data as a CSV file.

## Examples

```
# To run the script:  
source('./R/create_Data.R')
```

---

Frailty\_Calculation     *Calculating Frailty Scores*

---

## Description

This script processes patient episode data, assigns frailty categories to ICD codes, and calculates a frailty score based on categorized ICD codes.

The script follows these main steps:

1. Load and preprocess patient episode data from a CSV file.
2. Load the ICD-to-frailty mapping data.
3. Remove dots from ICD codes to standardize format.
4. Sort data by patient ID and start date.
5. Assign frailty categories based on ICD codes.
6. Prioritize exact matches, otherwise use prefix-based matching.
7. Calculate the frailty score as the sum of all relevant frailty categories per patient episode.
8. Export processed data to a CSV file.

## Examples

```
# Process and calculate frailty scores  
source("R/Frailty_score.R")
```

---

Frailty\_score

*Calculating Frailty Scores*


---

### Description

This script processes patient episode data, assigns frailty categories to ICD codes, and calculates a frailty score based on categorized ICD codes.

The script follows these main steps:

1. Load and preprocess patient episode data from a CSV file.
2. Load the ICD-to-frailty mapping data.
3. Remove dots from ICD codes to standardize format.
4. Sort data by patient ID and start date.
5. Assign frailty categories based on ICD codes.
6. Prioritize exact matches, otherwise use prefix-based matching.
7. Calculate the frailty score as the sum of all relevant frailty categories per patient episode.
8. Export processed data to a CSV file.

### Examples

```
# Process and calculate frailty scores
source("R/Frailty_score.R")
```

---

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](https://raw.githubusercontent.com/ropensci/make-data.R).

**Source**

CDC Website: <https://goo.gl/6e2mvp>

---

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](https://raw.githubusercontent.com/ropensci/make-data.R).

**Source**

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

---

`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]([https://raw.githubusercontent.com/rfordatascience/tidyr/master/data/icd10cm\\_2017/make-data.R](https://raw.githubusercontent.com/rfordatascience/tidyr/master/data/icd10cm_2017/make-data.R)).

---

`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]([https://raw.githubusercontent.com/rfordatascience/tidyr/master/data/icd10cm\\_2018/make-data.R](https://raw.githubusercontent.com/rfordatascience/tidyr/master/data/icd10cm_2018/make-data.R)).

---

`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](<https://raw.githubusercontent.com/.../raw/make-data.R>).

---

`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](<https://raw.githubusercontent.com/ellessenne/comorbidity/master/data-raw/make-data.R>).

**Source**

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

---

 LABTNSCPSS

---

*LABTNSCPSS: Comorbidity and Frailty Score Toolkit*


---

**Description**

This package provides tools for calculating comorbidity indices and frailty scores from different ICD coding systems (ICD-10-CA, ICD-10-CM, ICD-11).

---

 LabTNSCPSSPackage

---

*A Package for Chronic Pathology and Comorbidity Analysis*


---

**Description**

This package provides tools for preprocessing patient data, managing chronic pathologies, calculating frailty scores, and computing comorbidity indices (Charlson and Elixhauser). The package sources various scripts to automate these processes.

**Main Scripts**

- `setup_package.R`: Installs and loads required dependencies. - `source_scripts.R`: Sources external scripts required for data processing. - `create_Data.R`: Cleans and prepares input data. - `Pathologie_chronic.R`: Updates episodes with chronic pathology information. - `Fragility_comorbidity.R`: Computes frailty scores. - `comorbidity_ICD10CA_v2.R`: Calculates comorbidity indices.

**Usage**

The package should be used by sourcing the main functions in the order specified above.

---

`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](<https://raw.githubusercontent.com/ellessen/raw/make-data.R>).

---

 Pathologies\_chronic.R *Propagating Chronic Pathologies*


---

**Description**

This script processes patient episode data, identifies ICD codes, assigns chronic categories, propagates relevant codes, and extracts basal codes. It follows a structured approach:

Load and Preprocess Data

1. Assign Chronic Categories to ICD Codes
2. Identify and Propagate Category 2 Codes
3. Identify and Propagate Category 1 Codes
4. Extract Basal Codes for Category 1 ICDs

**Note**

Ensure that:

Dates are automatically converted to YYYY-MM-DD format.

- Missing category assignments are labeled as None.

**Examples**

```
# To run the script:
source('./R/Pathologie_chronic.R')
```

---

<code>sample_diag</code>	<i>Simulate ICD-10 and ICD-9 diagnostic codes</i>
--------------------------	---

---

**Description**

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

**Usage**

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

**Arguments**

<code>n</code>	Number of ICD codes to simulate.
<code>version</code>	The version of the ICD coding scheme to use. Possible choices are 'ICD10_2009', 'ICD10_2011', and 'ICD9_2015'; defaults to 'ICD10_2011'. See [comorbidity::icd10_2009], [comorbidity::icd10_2011], and [comorbidity::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 'comorbidity' 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" ('hycp') 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

## \* datasets

- australia10, [2](#)
- icd10\_2009, [9](#)
- icd10\_2011, [10](#)
- icd10cm\_2017, [11](#)
- icd10cm\_2018, [11](#)
- icd10cm\_2022, [12](#)
- icd9\_2015, [12](#)
- nhds2010, [14](#)

australia10, [2](#)

available\_algorithms, [3](#)

Chronic\_Pathologies.R, [3](#)

comorbidity, [4](#)

Comorbidity\_calculation, [7](#)

Comorbidity\_Frailty\_Calculation, [7](#)

create\_Data, [7](#)

Frailty\_Calculation, [8](#)

Frailty\_score, [9](#)

icd10\_2009, [9](#)

icd10\_2011, [10](#)

icd10cm\_2017, [11](#)

icd10cm\_2018, [11](#)

icd10cm\_2022, [12](#)

icd9\_2015, [12](#)

LABTNCPSS, [13](#)

LabTNCPSSPackage, [13](#)

nhds2010, [14](#)

Pathologies\_chronic.R, [15](#)

sample\_diag, [15](#)

score, [16](#)