

Package ‘LABTNSCPSS’

March 26, 2026

Type Package

Version 1.0.2

Title Calculation of Comorbidity and Frailty Scores

Description Computes comorbidity indices and combined frailty scores for multiple ICD coding systems, including ICD-10-CA, ICD-10-CM, and ICD-11. The package provides tools to preprocess episode data, map diagnosis codes to chronic categories, propagate conditions across episodes, and generate comorbidity and frailty measures. The methods implemented are original to this package and were developed by the authors for research applications; a manuscript describing the methodology is currently in preparation.

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

BugReports <https://github.com/bayaniazadeh/LabTNSCPSSPackage/issues>

License GPL-3

Depends R (>= 4.3.0)

Imports dplyr, magrittr, tibble, checkmate, tidyselect, data.table, stringi, lubridate, purrr, tidyr, glue

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

NeedsCompilation no

Author Azadeh Bayani [aut, cre] (ORCID:
<<https://orcid.org/0009-0002-7707-9602>>),
Jean Noel Nikiema [ctb],
Michèle Bally [ctb]

Maintainer Azadeh Bayani <azadeh.bayani@umontreal.ca>

Repository CRAN

Date/Publication 2026-03-26 16:10:02 UTC

Contents

australia10	2
Basal_Codes	3
chronic_pathologies	3
Code_Categories	4
comorbidity	4
Create_data	5
Frailty_Comorbidty_Mapping	6
Frailty_ICD10CM	6
Frailty_ICD11	7
Frailty_mapping	7
get_coding_system	8
ICD10CA_categorisation	8
icd10cm_2017	9
icd10cm_2018	9
icd10cm_2022	10
ICD10CM_Basal_Codes	10
ICD10CM_categorisation	11
icd10_2009	11
icd10_2011	12
ICD11_Basal_Codes	12
ICD11_categorisation	13
icd9_2015	13
LABTNSCPSS	14
nhds2010	14
run_pipeline	15
score	16
Index	18

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

 Basal_Codes

Basal_Codes

Description

Basal codes used for frailty/comorbidity mapping.

Usage

```
Basal_Codes
```

Format

A data frame.

Source

Internal data.

 chronic_pathologies

Carry-forward chronic ICD codes and build updated episode-level ICD history

Description

This function assigns chronic categories to ICD codes, propagates chronic conditions forward in time (category 2 indefinitely; category 1 within 1 year), adds basal codes, and writes an updated episode file with 'updated_icd_codes'.

Usage

```
chronic_pathologies(
  file_path_main,
  updated_path = NULL,
  coding_system = c("ICD-10-CA", "ICD-10-CM", "ICD-11")
)
```

Arguments

`file_path_main` Path to cleaned episode-level CSV (must include patient_id, episode_id, start_date, ICD).

`updated_path` Output CSV path. If NULL, a temp file is created.

`coding_system` One of "ICD-10-CA", "ICD-10-CM", "ICD-11".

Value

The output path ('updated_path').

Code_Categories	<i>Code_Categories</i>
-----------------	------------------------

Description

Category definitions used in the comorbidity/frailty algorithms.

Usage

```
Code_Categories
```

Format

A data frame.

Source

Internal data.

comorbidity	<i>Comorbidity mapping</i>
-------------	----------------------------

Description

Maps ICD diagnostic codes to comorbidity categories using supported algorithms.

Usage

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

Arguments

x	A data.frame (or data.table/tibble) containing at least an ID column and a code column.
id	Name of the patient identifier column in 'x'.
code	Name of the diagnostic code column in 'x'.
map	Name of the mapping algorithm to use.
assign0	Logical; if TRUE, applies hierarchy rules to avoid double-counting.
labelled	Logical; if TRUE, adds labels for easier viewing.
tidy.codes	Logical; if TRUE, cleans codes before mapping.

Value

A data.frame with comorbidity indicator columns and class "comorbidity".

Examples

```
x <- data.frame(
  id = c(1, 1, 2, 2),
  code = c("I10", "E119", "F200", "E110"),
  stringsAsFactors = FALSE
)

comorbidity(x = x, id = "id", code = "code",
            map = "charlson_icd10_quan", assign0 = FALSE)

comorbidity(x = x, id = "id", code = "code",
            map = "elixhauser_icd10_quan", assign0 = FALSE)
```

Create_data

Create and clean episode data

Description

Reads a CSV file, renames columns using a mapping, parses dates, cleans ICD codes, and writes a cleaned CSV to 'cleaned_path'.

Usage

```
Create_data(input_file, col_mapping, cleaned_path = NULL)
```

Arguments

input_file	Path to the input CSV.
col_mapping	Named character vector or list mapping standard names to user column names. Example: list(patient_id="Patient_id", ICD="diag_code", start_date="date_start", end_date="date_end", episode_id="episode_id")
cleaned_path	Path to write the cleaned CSV. If NULL, a default path in 'tempdir()' is used.

Value

The 'cleaned_path' (character string). Also invisibly returns the cleaned data.frame.

Frailty_Comorbidity_Mapping
Frailty_Comorbidity_Mapping

Description

Mapping table between frailty items and comorbidity categories.

Usage

Frailty_Comorbidity_Mapping

Format

A data frame.

Source

Internal data.

Frailty_ICD10CM *Frailty_ICD10CM*

Description

ICD-10-CM code lists used to compute frailty scores.

Usage

Frailty_ICD10CM

Format

A data frame.

Source

Internal data.

`Frailty_ICD11`*Frailty_ICD11*

Description

ICD-11 code lists used to compute frailty scores.

Usage

`Frailty_ICD11`

Format

A data frame.

Source

Internal data.

`Frailty_mapping`*Frailty_mapping*

Description

Mapping table used to compute frailty scores.

Usage

`Frailty_mapping`

Format

A data frame.

Source

Internal data.

get_coding_system *Choose a coding system*

Description

Choose a coding system

Usage

```
get_coding_system(
  coding_system = NULL,
  valid_systems = c("ICD-10-CA", "ICD-10-CM", "ICD-11"),
  verbose = interactive()
)
```

Arguments

coding_system Character string. If NULL and interactive(), the user is prompted.
 valid_systems Character vector of allowed coding systems.
 verbose Logical; if TRUE, prints guidance using message().

Value

A single character string indicating the selected coding system.

ICD10CA_categorisation
 ICD10CA_categorisation

Description

ICD-10-CA categorisation table used by the package.

Usage

```
ICD10CA_categorisation
```

Format

A data frame.

Source

Internal data.

`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/rockwellr/icd10cm_2017/master/raw/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/rockwellr/icd10cm_2018/master/raw/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/ICD10CM_Basal_Codes/raw/make-data.R).

`ICD10CM_Basal_Codes`*ICD10CM_Basal_Codes*

Description

Basal codes for ICD-10-CM used by the package.

Usage`ICD10CM_Basal_Codes`**Format**

A data frame.

Source

Internal data.

ICD10CM_categorisation
ICD10CM_categorisation

Description

ICD-10-CM categorisation table used by the package.

Usage

```
ICD10CM_categorisation
```

Format

A data frame.

Source

Internal data.

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/rford/ICD10CM_categorisation/master/raw/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/rfordatascience/veryinteresting/master/data/icd10_2011/make-data.R).

Source

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

`ICD11_Basal_Codes`*ICD11_Basal_Codes*

Description

Basal codes for ICD-11 used by the package.

Usage`ICD11_Basal_Codes`**Format**

A data frame.

Source

Internal data.

ICD11_categorisation *ICD11_categorisation*

Description

ICD-11 categorisation table used by the package.

Usage

ICD11_categorisation

Format

A data frame.

Source

Internal data.

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: Tools for comorbidity and frailty scoring

Description

This package provides tools for preprocessing patient data, managing chronic pathologies, calculating frailty scores, and computing comorbidity indices (Charlson and Elixhauser).

Tools for calculating comorbidity indices and frailty scores from ICD coding systems (ICD-10-CA, ICD-10-CM, ICD-11).

Author(s)

Maintainer: Azadeh Bayani <azadeh.bayani@umontreal.ca> ([ORCID](#))

Other contributors:

- Jean Noel Nikiema <jean.nikiema@umontreal.ca> [contributor]
- Michèle Bally <michele.bally.chum@ssss.gouv.qc.ca> [contributor]

See Also

Useful links:

- <https://github.com/bayaniazadeh/LabTNSCPSSPackage>
- Report bugs at <https://github.com/bayaniazadeh/LabTNSCPSSPackage/issues>

Useful links:

- <https://github.com/bayaniazadeh/LabTNSCPSSPackage>
- Report bugs at <https://github.com/bayaniazadeh/LabTNSCPSSPackage/issues>

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>).

run_pipeline

Run the LABTNSCPSS Comorbidity-Frailty Pipeline

Description

Computes Charlson, Elixhauser, Combined Comorbidity, Frailty, and Morbidity-Frailty scores from episode-level data.

Usage

```
run_pipeline(  
  input_file,  
  col_mapping,  
  coding_system = c("ICD-10-CA", "ICD-10-CM", "ICD-11"),  
  out_dir = tempdir()  
)
```

Arguments

<code>input_file</code>	Path to input CSV file.
<code>col_mapping</code>	Named list mapping required fields to column names in <code>'input_file'</code> . Must include: <code>'patient_id'</code> , <code>'ICD'</code> , <code>'start_date'</code> , <code>'end_date'</code> , <code>'episode_id'</code> .
<code>coding_system</code>	One of: "ICD-10-CA", "ICD-10-CM", "ICD-11".
<code>out_dir</code>	Directory where output CSV files will be written (default: <code>'tempdir()'</code>).

Value

A named list containing:

scores_final Final combined scores (comorbidity + frailty).

final_data_charlson Charlson scores by episode.

final_data_elixhauser Elixhauser scores by episode.

final_data_combined Combined comorbidity scores by episode.

score_pop_Elixh Weighted Elixhauser score (ICD-10-CA only; may be NULL).

Examples

```
library(LABTNCPSS)
f <- system.file("extdata", "testpackage.csv", package = "LABTNCPSS")

col_mapping <- list(
  patient_id = "trajectoire_id",
  ICD       = "diagnostic_code",
  start_date = "date_debut",
  end_date   = "date_fin",
  episode_id = "episode_id"
)

res <- run_pipeline(
  input_file = f,
  col_mapping = col_mapping,
  coding_system = "ICD-10-CA",
  out_dir = tempdir()
)
names(res)
```

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 <code>comorbidity()</code> .
weights	A string denoting the weighting system to be used (depends 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 indicating whether to apply a hierarchy of comorbidities (so milder forms are set to 0).

Value

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

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

```
# IMPORTANT: these examples must be runnable during R CMD check.
# If `x` is not defined in examples, create a tiny reproducible dataset here.
x <- data.frame(id = c(1, 1, 2), code = c("I10", "E119", "J449"), stringsAsFactors = FALSE)

# Checking the `assign0` argument:
x3 <- data.frame(id = 1, code = c("E100", "E102"), stringsAsFactors = FALSE)
ccF <- comorbidity(x = x3, id = "id", code = "code", map = "charlson_icd10_quan", assign0 = FALSE)
score(x = ccF, assign0 = FALSE)
score(x = ccF, assign0 = TRUE)
```

Index

* datasets

- australia10, [2](#)
 - Basal_Codes, [3](#)
 - Code_Categories, [4](#)
 - Frailty_Comorbidity_Mapping, [6](#)
 - Frailty_ICD10CM, [6](#)
 - Frailty_ICD11, [7](#)
 - Frailty_mapping, [7](#)
 - icd10_2009, [11](#)
 - icd10_2011, [12](#)
 - ICD10CA_categorisation, [8](#)
 - icd10cm_2017, [9](#)
 - icd10cm_2018, [9](#)
 - icd10cm_2022, [10](#)
 - ICD10CM_Basal_Codes, [10](#)
 - ICD10CM_categorisation, [11](#)
 - ICD11_Basal_Codes, [12](#)
 - ICD11_categorisation, [13](#)
 - icd9_2015, [13](#)
 - nhds2010, [14](#)
 - icd10cm_2018, [9](#)
 - icd10cm_2022, [10](#)
 - ICD10CM_Basal_Codes, [10](#)
 - ICD10CM_categorisation, [11](#)
 - ICD11_Basal_Codes, [12](#)
 - ICD11_categorisation, [13](#)
 - icd9_2015, [13](#)
 - nhds2010, [14](#)
 - LABTNSCPSS, [14](#)
 - nhds2010, [14](#)
 - run_pipeline, [15](#)
 - score, [16](#)
- australia10, [2](#)
- Basal_Codes, [3](#)
- chronic_pathologies, [3](#)
- Code_Categories, [4](#)
- comorbidity, [4](#), [17](#)
- Create_data, [5](#)
- Frailty_Comorbidity_Mapping, [6](#)
- Frailty_ICD10CM, [6](#)
- Frailty_ICD11, [7](#)
- Frailty_mapping, [7](#)
- get_coding_system, [8](#)
- icd10_2009, [11](#)
- icd10_2011, [12](#)
- ICD10CA_categorisation, [8](#)
- icd10cm_2017, [9](#)