

Package ‘polypharmacy’

February 12, 2021

Type Package

Version 0.0.0.9000

Title Calculate several polypharmacy indicators

Description Analyse prescription drug deliveries to calculate several indicators of polypharmacy corresponding to the various definitions found in the literature.

Maintainer Guillaume Boucher <guiboucher8@gmail.com>

BugReports <https://github.com/guiboucher/polypharmacy/issues>

License GPL-3 + file LICENSE

Imports data.table (>= 1.13.0),
doParallel (>= 1.0.16),
foreach (>= 1.5.1),
itertools (>= 0.1.3),
lubridate (>= 1.7.9),
parallel (>= 4.0.3),
stringr (>= 1.4.0)

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Encoding UTF-8

LazyData true

Suggests knitr,
rmarkdown,
testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 2.10)

Config/testthat/edition 3

R topics documented:

polypharmacy-package	2
cst_deliv_duration	3
data_process	4
drug_bkdn	7
indicators	7
ind_simult	8

ind_stdcontinuous	9
ind_stdcumul	10
ind_ucontinuous	11
ind_wcumul	11
Rx_processed	12
Rx_unprocessed	12
stat_quantile_prob	13

Index	14
--------------	-----------

polypharmacy-package	<i>Calculate several polypharmacy indicators</i>
----------------------	--

Description

This package analyse prescription drug deliveries to calculate several indicators of polypharmacy corresponding to the various definitions found in the literature.

Details

It is essential to know the concepts used to calculate the various polypharmacy indicators to adequately use this package.

The core of the package is the `data_process()` function that creates the `data.table` of drug treatments by restructuring the drug delivery records (usually extracted from a pharmacy or a health insurance information system) into continuous periods of drug availability (called drug treatments), applying user-defined arguments such as the grace periods between renewals or the longest treatment duration that an individual may accumulate through the successive renewals.

Then, each polypharmacy indicator can be computed using the corresponding function (`ind_simult()`, `ind_stdcumul()`, `ind_wcumul()`, `ind_stdcontinuous()`, `ind_ucontinuous()`) or using the overall function `indicators()` to select the desired indicator(s) to be calculated at once.

Prior to running `data_process()` the user may need to pre-process the table of original drug delivery records to break down combination drug into their individual components (`drugs_bkdn()`) and/or to overwrite some delivery durations of specified drugs with constant durations (`cst_trt_dur()`).

Author(s)

Maintainer: Guillaume Boucher <guiboucher8@gmail.com>

Authors:

- Bernard Candas <bernard.candas.1@gmail.com>
- Houssein Missaoui <missaoui_houssein@hotmail.fr>

See Also

Useful links:

- Report bugs at <https://github.com/guiboucher/polypharmacy/issues>

cst_deliv_duration	<i>Constant delivery duration drugs</i>
--------------------	---

Description

Overwrites the delivery durations with constant durations for each drug code listed in a user-provided table.

Usage

```
cst_deliv_duration(  
  Rx_deliv,  
  Rx_drug_code,  
  Rx_deliv_dur,  
  Cst_deliv_dur,  
  Cst_drug_code,  
  Cst_duration  
)
```

Arguments

Rx_deliv	Name of the table listing all prescription drugs delivered.
Rx_drug_code	Column name of Rx_deliv that contains the drug unique identifier.
Rx_deliv_dur	Column name of the constant treatment duration in the Rx_deliv table.
Cst_deliv_dur	Name of the table that contains the constant delivery durations that will overwrite that in the Rx_deliv table for the specified drug codes.
Cst_drug_code	Column name of Cst_deliv_dur that contains the drug unique identifier (same format as Rx_drug_code).
Cst_duration	Column name of the constant treatment duration in the Cst_deliv_dur table (same format as Rx_deliv_dur).

Value

data.table of the same structure as Rx_deliv.

Examples

```
Rx_dt <- data.frame(id = c(rep(1, 3), rep(2, 2)),  
  code = c("A", "B", "C", "B", "D"),  
  duration = c(rep(15, 3), 15, 90))  
cst_dt <- data.frame(codes = c("A", "C", "D"),  
  dur = c(50, 100, 45))  
Rx_cst <- cst_deliv_duration(Rx_deliv = Rx_dt,  
  Rx_drug_code = "code", Rx_deliv_dur = "duration",  
  Cst_deliv_dur = cst_dt,  
  Cst_drug_code = "codes", Cst_duration = "dur")
```

data_process	Create the table of the drug treatments
--------------	---

Description

Reads a table of successive drug delivery records (usually extracted from a pharmacy or a health insurance information system) and creates the table required for the calculation of the polypharmacy indicators by applying various user-defined arguments, incorporating hospital stays into the treatment periods and reconstruct continuous treatment periods by merging quasi continuous and/or overlapping drugs deliveries.

Usage

```
data_process(
  Rx_deliv,
  Rx_id,
  Rx_drug_code,
  Rx_drug_deliv,
  Rx_deliv_dur,
  Cohort = NULL,
  Cohort_id = NULL,
  Hosp_stays = NULL,
  Hosp_id = NULL,
  Hosp_admis = NULL,
  Hosp_discharge = NULL,
  study_start = NULL,
  study_end = NULL,
  grace_fctr = 0.5,
  grace_cst = 0,
  max_reserve = NULL,
  cores = parallel::detectCores(logical = FALSE),
  ...
)
```

Arguments

Rx_deliv	Name of the table listing all prescription drugs deliveries including the run-in period. See <i>Details</i> .
Rx_id	Column name of Rx_deliv containing individual unique identifier (any format).
Rx_drug_code	Column name of Rx_deliv that contains the drug unique identifier (any format).
Rx_drug_deliv	Column name of Rx_deliv that contains the dates of the drug delivery (Date format, see <i>Details</i>).
Rx_deliv_dur	Column name of Rx_deliv that contains the duration of the delivery (integer number).
Cohort	Name of the table providing the unique identifiers of the study cohort. Only the ids listed in both the Cohort and the Rx_deliv tables will be returned. if Cohort = NULL, all ids of the Rx_deliv table will be returned.
Cohort_id	Column name of Cohort containing individual's unique identifiers (same format as Rx_id). If Cohort is not NULL and Cohort_id is NULL, Cohort_id will take the same value as Rx_id.

Hosp_stays	Name of the table listing all hospital stays. (see <i>Details</i> for possible format).
Hosp_id	Column name of Hosp_stays containing individual's unique identifier (same format as Rx_id). If Hosp_stays is not NULL and Hosp_id is NULL, Hosp_id will take the same value as Rx_id.
Hosp_admis	Column name of Hosp_stays that contains the date of admission in hospital (Date format, see <i>Details</i>).
Hosp_discharge	Column name of Hosp_stays that contains the date of discharge from hospital (Date format, see <i>Details</i>).
study_start, study_end	Defines the first and last day of the study period for which the polypharmacy indicator(s) need to be calculated. All treatment periods prior to study_start and past study_end are not transcribed into the result table (Date format, see <i>Details</i>).
grace_fctr, grace_cst	Numbers ≥ 0 . Two types of grace periods can be applied. One is proportional to the treatment duration of the latest delivery (grace_fctr) and the other is a constant number of days (grace_cst).
max_reserve	An integer number ≥ 0 or NULL. Longest treatment duration, in days, that can be stored from successive overlapping deliveries. When max_reserve = NULL no limit is applied. When max_reserve = 0 no accumulation of extra treatment duration is accounted for.
cores	The number of cores to use when executing data_process(). See parallel::detectCores .

Details

Variables:

- Rx_id, Cohort_id and Hosp_id columns must be of the same class (integer, numeric, character, ...).
- Rx_drug_deliv, Hosp_admis and Hosp_discharge can be 1) as.Date('yyyy-mm-dd'), 2) as.character('yyyy-mm-dd') or 3) as.integer() where 0 is January 1st, 1970.

Arguments:

- study_start and study_end can be 1) as.Date('yyyy-mm-dd'), 2) as.character('yyyy-mm-dd') or 3) as.integer() where 0 is January 1st, 1970.

Hospital stays:

Drug availability is assumed to continue during the hospital stay as it is on the day prior admission. The patient is assumed to resume the consumption of the drugs delivered by community pharmacists (as recorded in Rx_deliv) the day after hosp_discharge.

Run-in period:

A run-in period is necessary to account for the medications that are available to the individuals on the day of study_start. It is recommended to include a run-in period of about 6 months (e.g. 7 months to account for possible delays) as some drugs are delivered for up to 6 months at once.

Grace period:

The grace period is used to determine if two successive deliveries can be considered as a continuous treatment even if there is a gap of several days for which no treatment is apparently available. Two successive deliveries of an identical drug are considered part of a single continuous treatment if the next delivery doesn't occur more than $\text{grace_cst} + (\text{grace_fctr} \times \text{Rx_deliv_dur})$ days after the end of the latest drug delivery. The availability of extra drugs accumulated over the successive deliveries is accounted for prior to evaluating the duration of the gap between deliveries.

Performance

For better performance, date columns are converted to integer numbers.

...

verif_cols=FALSE : For better performance, you can avoid columns class checking with `verif_cols=FALSE`.
Not recommended.

Value

data.table with four (4) variables:

- The individual unique identifier which name is defined by `Rx_id`.
- The drug unique identifier which name is defined by `Rx_drug_code`.
- `tx_start`: The date of initiation of the reconstructed continued treatment (format as date).
- `tx_end`: The date of the last day of the reconstructed continued treatment (format as date).

Examples

```
Rx_dt1 <- data.frame(id = 1, code = 'A',
  date = c('2020-01-01', '2020-01-09', '2020-01-21', '2020-02-05', '2020-02-21'),
  duration = 10)
Rx1 <- data_process(Rx_deliv = Rx_dt1,
  Rx_id = 'id', Rx_drug_code = 'code',
  Rx_drug_deliv = 'date', Rx_deliv_dur = 'duration')

## With a study cohort
Rx_dt2 <- data.frame(id = c(1, 1, 1, 2, 2), code = 'A',
  date = c('2020-01-01', '2020-01-09', '2020-01-21', '2020-02-05', '2020-02-21'),
  duration = 10)
Cohort_dt2 = data.frame(id = 1, age = 65, sex = 'F', x1 = 'ind8', x2 = 'ex1')
Rx2 <- data_process(Rx_deliv = Rx_dt2,
  Rx_id = 'id', Rx_drug_code = 'code',
  Rx_drug_deliv = 'date', Rx_deliv_dur = 'duration',
  Cohort = Cohort_dt2, Cohort_id = 'id')

## With hospital stays
Hosp_dt2 <- data.frame(id = 1,
  start = c('2019-01-01', '2019-12-25'),
  end = c('2019-05-20', '2019-12-31'))
Rx3 <- data_process(Rx_deliv = Rx_dt2,
  Rx_id = 'id', Rx_drug_code = 'code',
  Rx_drug_deliv = 'date', Rx_deliv_dur = 'duration',
  Cohort = Cohort_dt2, Cohort_id = 'id',
  Hosp_stays = Hosp_dt2, Hosp_id = 'id',
  Hosp_admis = 'start', Hosp_discharge = 'end')

## With study_start not NULL
Rx3_start <- data_process(Rx_deliv = Rx_dt2,
  Rx_id = 'id', Rx_drug_code = 'code',
  Rx_drug_deliv = 'date', Rx_deliv_dur = 'duration',
  Cohort = Cohort_dt2, Cohort_id = 'id',
  Hosp_stays = Hosp_dt2, Hosp_id = 'id',
  Hosp_admis = 'start', Hosp_discharge = 'end',
  study_start = '2019-12-29')
```

drug_bkdn	<i>Translate combination drug deliveries into single active ingredients</i>
-----------	---

Description

Replaces each combination drug into several deliveries of elementary active ingredients according to a user-provided correspondence table.

Usage

```
drug_bkdn(Rx_deliv, Rx_drug_code, Combn_drugs, Combn_drug_code, Combn_act_code)
```

Arguments

Rx_deliv	Name of the table listing all prescription drugs deliveries.
Rx_drug_code	Column name of Rx_deliv that contains the combination drug unique identifiers (any format).
Combn_drugs	Name of the correspondence table listing all elementary active ingredients that make up each combination drug.
Combn_drug_code	Column name of Combn_drugs that contains the combination drug unique identifiers (same format as Rx_drug_code).
Combn_act_code	Column name of elementary active ingredients that is present in Combn_drugs (same format as Rx_drug_code).

Value

data.table of the same structure as Rx_deliv.

Examples

```
Rx_dt <- data.frame(id = c(1, 1, 2, 2, 2),
                    codeDrug = c(159, 753, 123, 456, 789))
SplitCode <- data.frame(code = c(159, 159, 456, 456, 456),
                        split_code = c(1591, 1592, 4567, 4568, 4569))
Rx_split <- drug_bkdn(Rx_deliv = Rx_dt, Rx_drug_code = "codeDrug",
                     Combn_drugs = SplitCode, Combn_drug_code = "code",
                     Combn_act_code = "split_code")
```

indicators	<i>Title</i>
------------	--------------

Description

Description

Usage

```
indicators(arg)
```

Details

Detail

Value

Value

ind_simult	<i>Indicator: Simultaneous</i>
------------	--------------------------------

Description

Descriptive statistics on daily consumption.

Usage

```
ind_simult(  
  processed_tab,  
  individual_stats = c("mean", "min", "median", "max"),  
  stats = c("mean", "sd", "min", "p5", "p10", "p25", "median", "p75", "p90", "p95",  
    "max"),  
  calendar = FALSE,  
  cores = parallel::detectCores(logical = FALSE)  
)
```

Arguments

- processed_tab Table created by [data_process](#) function.
- individual_stats Statistics to calculate for each drug user. See *Details* for possible values.
- stats Statistics to calculate for each individual_stats. See *Details* for possible values.
- calendar TRUE or FALSE. Create a table indicating the number of drugs consumed for each day for each user (FALSE by default).
- cores The number of cores to use when executing ind_simult(). See [parallel::detectCores](#).

Details

individual_stats & stats: Possible values are

- 'mean', 'min', 'median', 'max', 'sd';
- 'pX' where X is a value in]0, 100];
- 'q1' = 'p25', 'q2' = 'p50' = 'median', q3 = 'p75'.

Value

if calendar is FALSE:

- data.table indicating each stats (columns) for each individual_stats (rows).

if calendar is TRUE, a list of two (2) elements:

- indicators: Table described above.
- calendar: Table indicating the number of drugs consumed for each day (only for those who has at least 1 day with 1 drug consumption).

Examples

```
dt_process <- data_process(
  Rx_deliv = data.frame(
    ID = c(1, 1, 1, 2, 2), Code = c('A', 'B', 'C', 'D', 'E'),
    Date = c('2020-01-01', '2020-01-05', '2020-01-10', '2020-01-15', '2020-01-26'),
    Duration = c(20, 15, 10, 5, 3)
  ), Rx_id = 'ID', Rx_drug_code = 'Code', Rx_drug_deliv = 'Date', Rx_deliv_dur = 'Duration',
  cores = 1
)
dt_simult <- ind_simult(dt_process, cores = 1)
dt_calendar <- ind_simult(dt_process, calendar = TRUE, cores = 1)
```

ind_stdcontinuous	<i>Title</i>
-------------------	--------------

Description

Description

Usage

```
ind_stdcontinuous(arg)
```

Details

Detail

Value

Value

ind_stdcumul	<i>Indicator: Cumulative (multiple medication)</i>
--------------	--

Description

Descriptive statistics: Sum of different drugs consumed over a given period time.

Cumulative polypharmacy, also known as multiple medication (Hovstadius et al.,2010a), is defined by the sum of different medications administered over a given period of time (Fincke et al., 2005).

Usage

```
ind_stdcumul(
  processed_tab,
  nPeriod = 1,
  stats = c("mean", "sd", "min", "p5", "p10", "p25", "median", "p75", "p90", "p95",
            "max")
)
```

Arguments

processed_tab	Table created by data_process function.
nPeriod	Integer value greater or equal to 1 and lesser or equal to the total number of days in the study period. If nPeriod is greater than 1, the study period is divide in nPeriod subperiod and the total number of drugs consumption would be the average of the periods.
stats	Statistics to calculate on the drug consumption. See <i>Details</i> for possible values.

Details

stats: Possible values are

- 'mean', 'min', 'median', 'max', 'sd';
- 'pX' where X is a value in]0, 100];
- 'q1' = 'p25', 'q2' = 'p50' = 'median', q3 = 'p75'.

The attribute nRx_period is a data.table indicating the number of drugs consumption per period and the average of these periods.

Value

data.table indicating each stats (columns).

ind_ucontinuous	<i>Title</i>
-----------------	--------------

Description

Description

Usage

```
ind_ucontinuous(arg)
```

Details

Detail

Value

Value

ind_wcumul	<i>Indicator: Weight Cumulative</i>
------------	-------------------------------------

Description

Description

Usage

```
ind_wcumul(
  processed_tab,
  stats = c("mean", "sd", "min", "p5", "p10", "p25", "median", "p75", "p90", "p95",
            "max")
)
```

Details

stats: Possible values are

- 'mean', 'min', 'median', 'max', 'sd';
- 'pX' where X is a value in]0, 100];
- 'q1' = 'p25', 'q2' = 'p50' = 'median', q3 = 'p75'.

Value

data.table indicating each stats (columns).

Rx_processed

*Table: Processed "unprocessed table"***Description**

Table required for the calculation of the polypharmacy indicators. This table is created by using `data_process()` function on Rx_unprocessed data.

Usage

Rx_processed

Format

A data.table with 6792 obs and 4 variables:

id Individual unique identifier.

code Drug unique identifier.

tx_start The date of initiation of the reconstructed continued treatment (format as date).

tx_end The date of the last day of the reconstructed continued treatment (format as date).

Rx_unprocessed

*Table: Prescription drugs deliveries***Description**

Table listing all prescription drugs deliveries.

Usage

Rx_unprocessed

Format

A data.table with 17060 obs and 4 variables:

id Individual unique identifier

code Drug unique identifier

start Date of the drug delivery

duration Duration of the delivery

stat_quantile_prob	<i>Statistic functions</i>
--------------------	----------------------------

Description

Determine the percentile from a qX value where X is a number from 1 to 3.

Usage

```
stat_quantile_prob(x)
```

Arguments

x	Character string, a quantile function, quarter values.
---	--

Value

Number 25, 50, 75

Index

* datasets

Rx_processed, [12](#)
Rx_unprocessed, [12](#)

cst_deliv_duration, [3](#)

data_process, [4](#), [8](#), [10](#)
drug_bkdn, [7](#)

ind_simult, [8](#)
ind_stdcontinuous, [9](#)
ind_stdcumul, [10](#)
ind_ucontinuous, [11](#)
ind_wcumul, [11](#)
indicators, [7](#)

parallel::detectCores, [5](#), [8](#)
polypharmacy (polypharmacy-package), [2](#)
polypharmacy-package, [2](#)

Rx_processed, [12](#)
Rx_unprocessed, [12](#)

stat_quantile_prob, [13](#)