Package: polypharmacy (via r-universe)

<div align="center">September 9, 2024</div>

Type Package

Version 1.0.0

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. Bjerrum, L., Rosholm, J. U., Hallas, J., & Kragstrup, J. (1997) <doi:10.1007/s002280050329>. Chan, D.-C., Hao, Y.-T., & Wu, S.-C. (2009a) <doi:10.1002/pds.1712>. Fincke, B. G., Snyder, K., Cantillon, C., Gaehde, S., Standring, P., Fiore, L., ... Gagnon, D.R. (2005) <doi:10.1002/pds.966>. Hovstadius, B., Astrand, B., & Petersson, G. (2009) <doi:10.1186/1472-6904-9-11>. Hovstadius, B., Astrand, B., & Petersson, G. (2010) <doi:10.1002/pds.1921>. Kennerfalk, A., Ruigómez, A., Wallander, M.-A., Wilhelmsen, L., & Johansson, S. (2002) <doi:10.1345/aph.1A226>. Masnoon, N., Shakib, S., Kalisch-Ellett, L., & Caughey, G. E. (2017) <doi:10.1186/s12877-017-0621-2>. Narayan, S. W., & Nishtala, P. S. (2015) <doi:10.1007/s40801-015-0020-y>. Nishtala, P. S., & Salahudeen, M. S. (2015) <doi:10.1159/000368191>. Park, H. Y., Ryu, H. N., Shim, M. K., Sohn, H. S., & Kwon, J. W. (2016) <doi:10.5414/cp202484>. Veehof, L., Stewart, R., Haaijer-Ruskamp, F., & Jong, B. M. (2000) <doi:10.1093/fampra/17.3.261>.

Maintainer Guillaume Boucher <guiboucher8@gmail.com>

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

License MIT + file LICENSE

Imports data.table, doParallel, foreach, itertools, lubridate, parallel, stringr

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Depends R (>= 3.5.0)

Config/testthat/edition 3

Repository https://guiboucher.r-universe.dev

RemoteUrl https://github.com/guiboucher/polypharmacy

RemoteRef HEAD

RemoteSha 0aa752a6184e213ffcc4493adf64f933a8a173d9

Contents

---

polypharmacy-package   *Calculate several polypharmacy indicators*

Description

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

Details

It is essential to understand the underlying 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 a 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) for every individual over the whole period of the study period. This process relies on several user-defined arguments such as the grace periods between renewals or the longest treatment duration that an individual may accumulate through successive early 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 wrapper 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 the delivery durations of some specified drugs with constant user-defined durations (cst_trt_dur()).

Author(s)

Maintainer: Guillaume Boucher <guiboucher8@gmail.com>

Authors:

- Bernard Candas <bernard.candas.1@gmail.com>
- Houssem Missaoui <missaoui_houssem@hotmail.fr>

Other contributors:

- Caroline Sirois <caroline.sirois@pha.ulaval.ca> [scientific advisor]
- Marc Simard <marc.simard@inspq.qc.ca> [scientific advisor]

See Also

Useful links:

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

---

cst_deliv_duration   *Constant delivery duration drugs*

Description

Overwrites the recorded delivery durations of specific drugs with constant durations as specified 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

```
# With matches
rx1 <- data.frame(id = c(1, 1, 2, 2, 2), code = c("A", "B", "B", "C", "D"),
                  duration = as.integer(c(30, 15, 15, 7, 90)))
cst1 <- data.frame(CODES = c("B", "D"), DURATION = as.integer(c(45, 60)))
cst_deliv_duration(
  Rx_deliv = rx1, Rx_drug_code = "code", Rx_deliv_dur = "duration",
  Cst_deliv_dur = cst1, Cst_drug_code = "CODES", Cst_duration = "DURATION"
)

# No matches
rx2 <- data.frame(id = c(1, 1, 2, 2, 2), code = c("A", "B", "B", "C", "D"),
                  duration = as.integer(c(30, 15, 15, 7, 90)))
cst2 <- data.frame(CODES = c("E", "F"), DURATION = as.integer(c(45, 60)))
cst_deliv_duration(
  Rx_deliv = rx2, Rx_drug_code = "code", Rx_deliv_dur = "duration",
  Cst_deliv_dur = cst2, Cst_drug_code = "CODES", Cst_duration = "DURATION"
)
```

---

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 *Details*.
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 *Details*).
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 *Details* 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 *Details*).
Hosp_discharge	Column name of Hosp_stays that contains the date of discharge from hospital (Date format, see *Details*).

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 *Details*).

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 [detectCores](#).

...

> Additional arguments. See *Details*. Should not be used.

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.

Grace period is always zero (0) for hospital stays.

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 grace_cst + (grace_fctr × 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

```
### Standard evaluation
data_process(
  Rx_deliv = sample_Rx_unprocessed, Rx_id = "id", Rx_drug_code = "code",
  Rx_drug_deliv = "start", Rx_deliv_dur = "duration",
  cores = 1L
)

### Hospitalisation stays
rx1 <- data.frame(
  id = c(1L, 3:8),
  code = LETTERS[c(1, 3:8)],
  date = as.Date(c("2001-01-15", "2003-03-15", "2004-04-15", "2005-05-15",
                   "2006-06-15", "2007-07-15", "2008-08-15")),
  duration = 10L
)
hosp1 <- data.frame(
  ID = 3:8,
  ADM = as.Date(c("2003-03-10", "2004-04-25", "2005-05-12",
                  "2006-06-20", "2007-07-26", "2008-08-01")),
  DEP = as.Date(c("2003-03-14", "2004-04-30", "2005-05-17",
                  "2006-06-30", "2007-07-30", "2008-08-13"))
)
data_process(
  Rx_deliv = rx1, Rx_id = "id", Rx_drug_code = "code",
  Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
  Hosp_stays = hosp1, Hosp_id = "ID", Hosp_admis = "ADM", Hosp_discharge = "DEP",
  study_start = "2001-01-01", study_end = "2008-12-31",
  cores = 1L
)
# Many drug codes
rx2 <- data.frame(
  id = 1L,
  code = c(111L, 222L, 222L, 333L, 444L),
 date = as.Date(c("2001-01-15", "2002-02-15", "2002-03-01", "2004-04-07", "2004-05-05")),
  duration = as.integer(c(10, 10, 10, 30, 10))
)
```

```
hosp2 <- data.frame(
  id = 1L,
  adm = as.Date(c("2000-01-01", "2000-01-15", "2001-01-01", "2002-02-23", "2004-04-15")),
  dep = as.Date(c("2000-01-31", "2000-01-31", "2001-01-10", "2002-02-28", "2004-05-15"))
)
data_process(
  Rx_deliv = rx2, Rx_id = "id", Rx_drug_code = "code",
  Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
  Hosp_stays = hosp2, Hosp_id = "id", Hosp_admis = "adm", Hosp_discharge = "dep",
  study_start = "2001-01-01", study_end = "2008-12-31",
  cores = 1L
)


### Study dates - start and end
rx3 <- data.frame(id = 1:3,
                  code = "A",
                  date = as.Date(c("2020-01-01", "2020-06-06", "2020-12-22")),
                  duration = 10L)
# NULLs
data_process(Rx_deliv = rx3, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             study_start = NULL, study_end = NULL,
             cores = 1)
# Not NULLs
data_process(Rx_deliv = rx3, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             study_start = "2020-06-10", study_end = NULL,
             cores = 1)
data_process(Rx_deliv = rx3, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             study_start = NULL, study_end = "2020-06-10",
             cores = 1)
data_process(Rx_deliv = rx3, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             study_start = "2020-01-05", study_end = "2020-12-25",
             cores = 1)


### Grace factor
rx4 <- data.frame(id = c(rep(1, 3), rep(2, 3)),
                  code = "A",
                  date = as.Date(c("2000-01-01", "2000-01-17", "2000-01-31",
                                   "2000-06-01", "2000-06-23", "2000-07-16")),
                  duration = as.integer(c(10, 10, 10, 15, 15, 15)))
# 50% of duration
data_process(Rx_deliv = rx4, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             grace_fctr = 0.5,
             cores = 1)
# 0% of duration
data_process(Rx_deliv = rx4, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             grace_fctr = 0,
             cores = 1)
```

```
### Grace constant
rx5 <- data.frame(id = 1,
                  code = "A",
                  date = as.Date(c("2000-01-01", "2000-01-14", "2000-01-25")),
                  duration = as.integer(c(10, 10, 6)))
# 2 days
data_process(Rx_deliv = rx5, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             grace_fctr = 0, grace_cst = 2,
             cores = 1)
# 3 days
data_process(Rx_deliv = rx5, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             grace_fctr = 0, grace_cst = 3,
             cores = 1)


### Max reserve
rx6 <- data.frame(id = as.integer(c(1, 1, 3, 3, 3, 5, 5)),
                  code = "A",
                  date = as.Date(c("2000-01-01", "2000-01-31",
                                   "2000-03-03", "2000-03-15", "2000-03-30",
                                   "2000-05-05", "2000-05-05")),
                  duration = as.integer(c(30, 30,
                                          30, 30, 30,
                                          90, 90)))
# 0 days
data_process(Rx_deliv = rx6, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             study_start = NULL, study_end = "2000-12-31",
             grace_fctr = 0, grace_cst = 0,
             max_reserve = 0,
             cores = 1)
# 60 days
data_process(Rx_deliv = rx6, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             study_start = NULL, study_end = "2000-12-31",
             grace_fctr = 0, grace_cst = 0,
             max_reserve = 60,
             cores = 1)
# Inf days
data_process(Rx_deliv = rx6, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             study_start = NULL, study_end = "2000-12-31",
             grace_fctr = 0, grace_cst = 0,
             max_reserve = NULL,
             cores = 1)


### Combine Hospital stays and Grace factor
rx7 <- data.frame(id = c(1L, 1L, 1L, 2L),
                  code = "A",
                  date = c("2000-01-01", "2000-02-20", "2000-04-11", "2002-02-02"),
                  duration = as.integer(c(30, 30, 30, 15)))
```

```
hosp7 <- data.frame(id = 1L,
                    adm = c("2000-01-11", "2000-02-21"),
                    dep = c("2000-01-15", "2000-02-25"))
data_process(Rx_deliv = rx7, Rx_id = "id", Rx_drug_code = "code",
             Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
             Hosp_stays = hosp7, Hosp_id = "id",
             Hosp_admis = "adm", Hosp_discharge = "dep",
             study_start = "2000-01-01", study_end = "2002-12-31",
             grace_fctr = 0.5, grace_cst = 0, max_reserve = NULL,
             cores = 1)
```

---

drug_bkdn   *Translate combination drug deliveries into several single active ingre-*
                                   dients

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

```
### With matches
rx1 <- data.frame(id = c(1L, 1L, 2L, 2L, 2L),
                   code = c(159L, 753L, 123L, 456L, 789L))
split1 <- data.frame(code = c(159L, 159L, 456L, 456L, 456L),
                      splitcode = c(1591L, 1592L, 4567L, 4568L, 4569L))
drug_bkdn(Rx_deliv = rx1, Rx_drug_code = "code",
          Combn_drugs = split1, Combn_drug_code = "code", Combn_act_code = "splitcode")

### No matches
rx2 <- data.frame(id = c(1L, 1L, 2L, 2L, 2L),
                   code = c(159L, 753L, 123L, 456L, 789L))
split2 <- data.frame(CODE = c(147L, 147L, 963L, 963L, 963L),
                      SPLITCODE = c(1471L, 1472L, 9637L, 9638L, 9639L))
drug_bkdn(Rx_deliv = rx2, Rx_drug_code = "code",
          Combn_drugs = split2, Combn_drug_code = "CODE", Combn_act_code = "SPLITCODE")
```

---

indicators	*Provide several polypharmacy indicators at once*

Description

Wrapper function to run sequentially various polypharmacy functions on a single set of data. Each function corresponds to a different definition of polypharmacy.

Usage

```
indicators(
  processed_tab,
 stats = c("mean", "sd", "min", "p5", "p10", "p25", "median", "p75", "p90", "p95",
    "max"),
  method = c("ind_simult", "ind_stdcumul", "ind_wcumul", "ind_stdcontinuous",
    "ind_ucontinuous"),
  stdconti_pdays = 90,
  simult_ind_stats = c("mean", "min", "median", "max"),
  simult_calendar = FALSE,
  stdcumul_nPeriod = c(1, 3),
  cores = parallel::detectCores()
)
```

Arguments

processed_tab	Name of the table of individual drug treatments to analyze. Created by the [data_process](#) function.
stats	Polypharmacy cohort descriptive statistics to calculate on every polypharmacy indicator requested. See *Details* for possible values.
method	Names of the functions corresponding to each of the polypharmacy indicators to be calculated.. See *Details* for possible values.

stdconti_pdays   pdays argument of the [ind_stdcontinuous](#) function. Can contain multiple
                 values. See *examples*.

simult_ind_stats

                 stats argument of the [ind_simult](#) function.

simult_calendar

                 TRUE or FALSE. calendar argument of the [ind_simult](#) function.

stdcumul_nPeriod

                 nPeriod argument of the [ind_stdcumul](#) function. Can contain multiple values.
                 See *examples*.

cores   The number of CPU cores to use when executing [ind_simult](#). See [detectCores](#).

Details

stats & simult_ind_stats: Possible values are

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

method: Possible values are

- 'ind_simult' to assess polypharmacy based on the daily simultaneous consumption of medication.
- 'ind_stdcumul to assess polypharmacy based on the cumulative number of distinct medications consumed over a given period of time (i.e. the standard definition).
- 'ind_wcumul' to assess polypharmacy based on the cumulative number of distinct medication consumed over a given period of time, weighted by the duration of consumption of each medication.
- 'ind_stdcontinuous' to assess polypharmacy based on the number of medications that are consumed both during the initial and the final period of the study period.
- 'ind_ucontinuous' to assess polypharmacy based on the uninterrupted consumption of distinct medications over the study period.

Value

list of the values returned by every function listed in the method argument.

Examples

```
dt_indic <- indicators(
  processed_tab = sample_Rx_processed,
 stats = c('mean', 'sd', 'min', 'p5', 'p10', 'p25', 'median', 'p75', 'p90', 'p95', 'max'),
 method = c('ind_simult', 'ind_stdcumul', 'ind_wcumul', 'ind_stdcontinuous', 'ind_ucontinuous'),
  stdconti_pdays = c(30, 90),
  simult_ind_stats = c('mean', 'min', 'median', 'max'),
  simult_calendar = TRUE,
  stdcumul_nPeriod = c(1, 3),
  cores = 1
)
```

---

ind_simult	*Assess polypharmacy based on the daily simultaneous consumption of medications*

Description

Calculates various metrics measuring the number of distinct medications consumed daily for every individual of the study cohort over the study period and provides cohort descriptive statistics on those metrics.

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()
)
```

Arguments

processed_tab   Table of individual drug treatments over the study period. Created by [data_process](#) function.

individual_stats

        Descriptive statistics of daily consumption over the study period to calculate for every individual. See *Details* for possible values.

stats   Cohort descriptive statistics to calculate on the polypharmacy indicator. See *Details* for possible values.

calendar   TRUE or FALSE. Create a table of the number of drugs consumed everyday by every individual (FALSE by default).

cores   The number of CPU cores to use. See [detectCores](#).

Details

individual_stats & stats: Possible values are

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

Value

list:

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

- stats_id: data.table indicating each individual_stats for each individuals (all cohort).

- min_conso: data.table indicating each stats for the number of days where an individual consume at least X drugs.

- calendar: If calendar=TRUE, data.table indicating the number of drugs consumed for each day (only for individuals who has at least 1 day with 1 drug consumption).

Examples

```
rx1 <- data.frame(id = c(1, 1, 2),
                  code = c("A", "B", "A"),
                  date = c("2000-01-01", "2000-01-04", "2000-01-08"),
                  duration = c(5, 7, 5))
cohort1 <- data.frame(id = as.numeric(1:3),
                      age = c(45, 12, 89),
                      sex = c("F", "F", "M"))
rx_proc1 <- data_process(Rx_deliv = rx1, Rx_id = "id", Rx_drug_code = "code",
                         Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
                         Cohort = cohort1, Cohort_id = "id",
                         study_start = "2000-01-01", study_end = "2000-01-15",
                         cores = 1)
dt_ind_simult <- ind_simult(rx_proc1, calendar = TRUE, cores = 1)
```

---

ind_stdcontinuous | *Assess polypharmacy based on the number of medications that is consumed both during the initial and the final period of the study period*

Description

Calculates the number of distinct medications that are consumed both during the initial and the final period of the overall study period by every individual of the study cohort and provides cohort descriptive statistics on this indicator.

Usage

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

Arguments

processed_tab   Table of individual drug treatments over the study period. Created by data_process function.

pdays   Duration (in days) of the initial and final periods of time . The initial period = [min; min+pdays] and the final period = [max-pdays; max], where *min* and *max* are the study_start and study_end arguments. See data_process.

stats   Cohort descriptive statistics to calculate on the polypharmacy indicator. See *Details* for possible values.

Details

stats: Possible values are

- 'mean', 'min', 'median', 'max', 'sd';

- 'pX' where *X* is an integer value in]0, 100];

- 'q1'='p25', 'q2'='p50'='median', q3='p75'.

Value

list:

- indic: data.table indicating each stats (columns).

- stats_id: data.table indicating the number of drugs use for each individual (all cohort).

Examples

```
rx1 <- data.frame(id = c(1, 1, 1, 2, 3),
                  code = c("A", "A", "B", "A", "A"),
              date = c("2000-01-01", "2000-01-22", "2000-01-10", "2000-01-01", "2000-01-20"),
                  duration = c(10, 10, 22, 31, 12))
cohort1 <- data.frame(id = as.numeric(1:4),
                      age = c(45, 12, 89, 31),
                      sex = c("F", "F", "M", "M"))
rx_proc1 <- data_process(Rx_deliv = rx1, Rx_id = "id", Rx_drug_code = "code",
                         Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
                         Cohort = cohort1, Cohort_id = "id",
                         study_start = "2000-01-01", study_end = "2000-01-31",
                         cores = 1)
dt_ind_stdcontinuous <- ind_stdcontinuous(processed_tab = rx_proc1, pdays = 10)
```

---

ind_stdcumul   *Assess polypharmacy based on the average number of distinct medi-*
                      cations consumed over successive periods of time of equal length

---

Description

Averages the number of distinct medications that are consumed by every individual during successive periods of time of equal length and provides cohort descriptive statistics on this indicator.

Usage

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

Arguments

processed_tab   Table of individual drug treatments over the study period. Created by data_process function.

nPeriod   Number of subperiods of equal time length in which the study period will be subdivided: 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 divided in nPeriod subperiods and the number of medications consumed in each subperiod is averaged over the number of subperiods.

stats   Cohort descriptive statistics to calculate on the polypharmacy indicator. See *Details* for possible values.

Details

stats: Possible values are

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

Value

list:

- indic: data.table indicating each stats (columns).
- stats_id: data.table. For each individual (all cohort), indicate the number of drug use per period (perX where X is a number between 1 and nPeriod) and the mean of the periods (nRx).

Examples

```
rx1 <- data.frame(id = c(1, 1, 1, 2),
                   code = c("A", "B", "C", "A"),
                   date = c("2000-01-01", "2000-01-01", "2000-01-26", "2000-01-17"),
                   duration = c(30, 5, 5, 10))
cohort1 <- data.frame(id = as.numeric(1:3),
                      age = c(45, 12, 89),
                      sex = c("F", "F", "M"))
rx_proc1 <- data_process(Rx_deliv = rx1, Rx_id = "id", Rx_drug_code = "code",
                         Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
                         Cohort = cohort1, Cohort_id = "id",
                         study_start = "2000-01-01", study_end = "2000-01-30",
                         cores = 1)
# 1 period
dt_ind_stdcumul_per1 <- ind_stdcumul(processed_tab = rx_proc1, nPeriod = 1)
# 3 periods
dt_ind_stdcumul_per3 <- ind_stdcumul(processed_tab = rx_proc1, nPeriod = 3)
```

---

ind_ucontinuous	*Assess polypharmacy based on the uninterrupted consumption of distinct medications over the study period*

Description

Calculates the number of distinct medications that are consumed everyday with no interruption over the study period by every individual and provides cohort descriptive statistics on this indicator.

Usage

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

Arguments

processed_tab	Table of individual drug treatments over the study period. Created by data_process function.
stats	Cohort descriptive statistics to calculate on the polypharmacy indicator. See *Details* for possible values.

Details

stats: Possible values are

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

Value

list:

- indic: data.table indicating each stats (columns).
- stats_id: data.table indicating the number of drugs use for each individual (all cohort).

Examples

```
rx1 <- data.frame(id = c(1, 1, 1, 2),
                   code = c("A", "B", "C", "A"),
                   date = c("2000-01-01", "2000-01-01", "2000-01-26", "2000-01-17"),
                   duration = c(30, 29, 5, 10))
cohort1 <- data.frame(id = as.numeric(1:3),
                       age = c(45, 12, 89),
                       sex = c("F", "F", "M"))
rx_proc1 <- data_process(Rx_deliv = rx1, Rx_id = "id", Rx_drug_code = "code",
                          Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
                          Cohort = cohort1, Cohort_id = "id",
                          study_start = "2000-01-01", study_end = "2000-01-30",
                          cores = 1)
dt_ind_ucontinuous <- ind_ucontinuous(processed_tab = rx_proc1)
```

---

ind_wcumul   *Assess polypharmacy based on the number of distinct medications*
                                   consumed weighted to their respective duration of consumption

---

Description

Calculates the number of distinct medications weighted by the duration of consumption that are consumed by every individual and provides cohort descriptive statistics on this indicator.

Usage

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

Arguments

processed_tab   Table of individual drug treatments over the study period. Created by [data_process](#) function.

stats   Cohort descriptive statistics to calculate on the polypharmacy indicator. See *Details* for possible values.

Details

stats: Possible values are

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

Value

list:

- indic: data.table indicating each stats (columns).
- stats_id: data.table indicating the number of drugs use for each individual (all cohort).

Examples

```
rx1 <- data.frame(id = c(1, 1, 1, 2),
                  code = c("A", "B", "C", "A"),
                  date = c("2000-01-01", "2000-01-01", "2000-01-26", "2000-01-17"),
                  duration = c(30, 5, 5, 10))
cohort1 <- data.frame(id = as.numeric(1:3),
                      age = c(45, 12, 89),
                      sex = c("F", "F", "M"))
rx_proc1 <- data_process(Rx_deliv = rx1, Rx_id = "id", Rx_drug_code = "code",
                         Rx_drug_deliv = "date", Rx_deliv_dur = "duration",
                         Cohort = cohort1, Cohort_id = "id",
                         study_start = "2000-01-01", study_end = "2000-01-30",
                         cores = 1)
dt_ind_wcumul <- ind_wcumul(processed_tab = rx_proc1)
```

---

sample_Rx_processed   *Table: Processed* unprocessed table

Description

This table is provided to users of this package for training purposes. It is created by using data_process function on sample_Rx_unprocessed data.

Usage

```
sample_Rx_processed
```

Format

A data.table with 6792 obs and 4 variables:

id Individual unique identifier.

code Medication unique identifier.

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

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

sample_Rx_unprocessed   *Table: Prescription drugs deliveries*

Description

A sample table of prescription drugs deliveries provided to users of this package for training purposes. It contains the raw information that leads to sample_Rx_processed when processes by the data_process function.

Usage

```
sample_Rx_unprocessed
```

Format

A `data.table` with 17060 obs and 4 variables:

id Individual unique identifier.

code Medication unique identifier.

start Date of the medication delivery.

duration Treatment duration of the delivery.

Index