Package ‘DrugUtilisation’

June 19, 2023

Title Summarise Patient-Level Drug Utilisation in Data Mapped to the OMOP Common Data Model

Version 0.2.0

Description Summarise patient-level drug utilisation cohorts using data mapped to the Observational Medical Outcomes Partnership (OMOP) common data model. New users and prevalent users cohorts can be generated and their characteristics, indication and drug use summarised.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.2.3

Suggests covr, testthat (>= 3.1.5), odbc, here, RPostgres, duckdb, CodelistGenerator

Config/testthat/edition 3

Imports CDMConnector (>= 1.0.0), checkmate, DBI, dbplyr, dplyr, glue, lubridate, tidyr, rlang, RJSONIO, cli, PatientProfiles, magrittr, purrr

Depends R (>= 2.10)

LazyData true

URL https://darwin-eu-dev.github.io/DrugUtilisation/

NeedsCompilation no

Author Marti Catala [aut, cre] (https://orcid.org/0000-0003-3308-9905), Mike Du [aut] (https://orcid.org/0000-0002-9517-8834), Yuchen Guo [aut] (https://orcid.org/0000-0002-0847-4855), Edward Burn [ctb] (https://orcid.org/0000-0002-9286-1128), Xintong Li [ctb] (https://orcid.org/0000-0002-6872-5804)

Maintainer Marti Catala <marti.catalasabate@ndorms.ox.ac.uk>

Repository CRAN

Date/Publication 2023-06-19 15:00:05 UTC
R topics documented:

addDailyDose                           2
addDrugUse                             3
addIndication                          5
addPattern                             6
computeCohortAttrition                 7
computeCohortCount                     8
dailyDoseCoverage                      9
drugUseColumns                         10
generateConceptCohortSet               11
generateDrugUtilisationCohortSet       13
indicationColumns                      14
indicationToStrata                     15
mockDrugUtilisation                    16
patternTable                           17
readConceptList                        18
reportGenerator                        19
stratifyByUnit                         19
summariseCharacteristicsFromCodelist   20
summariseDrugUse                       21
summariseIndication                    22
summariseLargeScaleCharacteristics    24
summariseTableOne                      25

Index

addDailyDose    add daily dose information to a drug_exposure table

Description

add daily dose information to a drug_exposure table

Usage

addDailyDose(drugExposure, cdm, ingredientConceptId)

Arguments

drugExposure drugExposure it must contain drug_concept_id, quantity, drug_exposure_start_date
and drug_exposure_end_date as columns

cdm cdm

ingredientConceptId

ingredientConceptId for which to filter the drugs of interest

Value

same input table
addDrugUse

Examples

library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

cdm$drug_exposure %>%
  filter(drug_concept_id == 2905077) %>%
  addDailyDose(cdm, 1125315)

Description

Add new columns with dose related information to the cohort

Usage

addDrugUse(
  cohort,
  cdm,
  ingredientConceptId,
  conceptSetList = NULL,
  initialDailyDose = TRUE,
  numberExposures = TRUE,
  duration = TRUE,
  cumulativeDose = TRUE,
  numberEras = TRUE,
  supplementary = FALSE,
  gapEra = 30,
  eraJoinMode = "Zero",
  overlapMode = "Sum",
  sameIndexMode = "Sum",
  imputeDuration = "eliminate",
  imputeDailyDose = "eliminate",
  durationRange = c(1, Inf),
  dailyDoseRange = c(0, Inf)
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cohort</td>
<td>Cohort in the cdm</td>
</tr>
<tr>
<td>cdm</td>
<td>cdm_reference created with CDMConnector::cdmFromCon</td>
</tr>
</tbody>
</table>
ingredientConceptId
Ingredient OMOP concept that we are interested for the study. It is a compulsory input, no default value is provided.

conceptSetList
List of concepts to be included. If NULL all the descendants of ingredient concept id will be used. By default: NULL.

initialDailyDose
Whether initial dose should be displayed

numberExposures
Whether number exposures should be displayed

duration
Whether duration should be displayed

cumulativeDose
Whether cumulative dose should be displayed

numberEras
Whether the number of eras should be displayed

supplementary
Whether all supplementary variables should be displayed

gapEra
Number of days between two continuous exposures to be considered in the same era. By default: 180.

eraJoinMode
How two different continuous exposures are joined in an era. There are four options: "Zero" the exposures are joined considering that the period between both continuous exposures the subject is treated with a daily dose of zero. The time between both exposures contributes to the total exposed time. "Join" the exposures are joined considering that the period between both continuous exposures the subject is treated with a daily dose of zero. The time between both exposures does not contribute to the total exposed time. "Previous" the exposures are joined considering that the period between both continuous exposures the subject is treated with the daily dose of the previous subexposure. The time between both exposures contributes to the total exposed time. "Subsequent" the exposures are joined considering that the period between both continuous exposures the subject is treated with the daily dose of the subsequent subexposure. The time between both exposures contributes to the total exposed time. By default: "Previous".

overlapMode
How the overlapping between two exposures that do not start on the same day is solved inside a subexposure. There are five possible options: "Previous" the considered daily_dose is the one of the earliest exposure. "Subsequent" the considered daily_dose is the one of the new exposure that starts in that subexposure. "Minimum" the considered daily_dose is the minimum of all of the exposures in the subexposure. "Maximum" the considered daily_dose is the maximum of all of the exposures in the subexposure. "Sum" the considered daily_dose is the sum of all the exposures present in the subexposure. By default: "Previous".

sameIndexMode
How the overlapping between two exposures that start on the same day is solved inside a subexposure. There are five possible options: "Minimum" the considered daily_dose is the minimum of all of the exposures in the subexposure. "Maximum" the considered daily_dose is the maximum of all of the exposures in the subexposure. "Sum" the considered daily_dose is the sum of all the exposures present in the subexposure. By default: "Sum".

imputeDuration
Whether/how the duration should be imputed "eliminate", "median", "mean", "quantile25", "quantile75" . By default: eliminate
addIndication

Get indication for a target cohort

Description

Get indication for a target cohort

Usage

addIndication(
  x,
  cdm,
  indicationCohortName,
  indicationGap = 0,
  unknownIndicationTable = NULL,
  indicationDate = "cohort_start_date"
)
**Arguments**

- `x` Table in the cdm
- `cdm` A cdm reference created using CDMConnector
- `indicationCohortName` Name of indication cohort table
- `indicationGap` Gap between the event and the indication
- `unknownIndicationTable` Tables to search unknown indications
- `indicationDate` Date of the indication

**Value**

Same cohort adding the indications

**Examples**

```r
library(DrugUtilisation)
library(CodelistGenerator)

cdm <- mockDrugUtilisation()

indications <- list("headache" = 378253, "asthma" = 317009)
cdm <- generateConceptCohortSet(cdm, "indication_cohorts", indications)

acetaminophen <- getDrugIngredientCodes(cdm, "acetaminophen")
cdm <- generateDrugUtilisationCohortSet(cdm, "drug_cohort", acetaminophen)

cdm$drug_cohort %>%
  addIndication(cdm, "indication_cohorts", indicationGap = c(0, 30, 365))
```

---

**addPattern**

*add pattern info to a table containing drug_strength information*

**Description**

add pattern info to a table containing drug_strength information

**Usage**

`addPattern(drugList, cdm, ingredientConceptId)`
computeCohortAttrition

Arguments

   drugList   Table in the cdm that has contain drug_concept_id
   cdm        cdm_reference
   ingredientConceptId
               ingredientConceptId

Value

   It adds pattern_id and unit to the current table

Examples

library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

cdm$drug_exposure %>%
   addPattern(cdm, 1125315)

cdm$concept %>%
   filter(domain_id == "Drug") %>%
   select(drug_concept_id = concept_id) %>%
   addPattern(cdm, 1125315)

computeCohortAttrition

   Add a line in the attrition table. If the table does not exist it is created

Description

   Add a line in the attrition table. If the table does not exist it is created

Usage

   computeCohortAttrition(
     x,
     cdm,
     attrition = NULL,
     reason = "Qualifying initial records"
   )
computeCohortCount

Arguments

- **x**: A table in the cdm with at lest: 'cohort_definition_id' and subject_id'
- **cdm**: A cdm reference created using CDMConnector
- **attrition**: An attrition table. If NULL a new attrition table is created.
- **reason**: A character with the name of the reason.

Value

Reference to a table with the cohort attrition

Examples

```r
library(DrugUtilisation)
library(dplyr)
library(PatientProfiles)

cdm <- mockDrugUtilisation()

cdm$new_cohort <- cdm$observation_period %>%
  mutate(cohort_definition_id = 1) %>%
  select(cohort_definition_id, subject_id = person_id, cohort_start_date = observation_period_start_date, cohort_end_date = observation_period_end_date)

compute()

attrition <- computeCohortAttrition(cdm$new_cohort, cdm)

cdm$new_cohort <- cdm$new_cohort %>%
  addSex(cdm = cdm) %>%
  filter(sex == "Female") %>%
  select(-"sex") %>%
  compute()

attrition <- computeCohortAttrition(cdm$new_cohort, cdm)

print(attrition)
```

computeCohortCount  Computes the cohortCount attribute for a certain table
**dailyDoseCoverage**

**Description**

Computes the cohortCount attribute for a certain table

**Usage**

```
computeCohortCount(x, cdm)
```

**Arguments**

- `x` A table in the cdm with at lest: 'cohort_definition_id' and subject_id'
- `cdm` A cdm_reference object

**Value**

A reference to a table in the database with the cohortCount

**Examples**

```r
library(DrugUtilisation)
library(dplyr)
library(PatientProfiles)

cdm <- mockDrugUtilisation()

cdm$new_cohort <- cdm$observation_period %>%
mutate(cohort_definition_id = 1) %>%
addSex(cdm = cdm) %>%
filter(sex == "Female") %>%
select(
    cohort_definition_id, subject_id = person_id,
    cohort_start_date = observation_period_start_date,
    cohort_end_date = observation_period_end_date
) %>%
compute()

cohortCountRef <- computeCohortCount(cdm$new_cohort, cdm)

print(cohortCountRef)
```

**dailyDoseCoverage**

*Check coverage of daily dose computation in a sample of the cdm for selected concept sets and ingredient*

**Description**

Check coverage of daily dose computation in a sample of the cdm for selected concept sets and ingredient
Usage

dailyDoseCoverage(
    cdm,
    ingredientConceptId,
    sample = NULL,
    conceptSetList = NULL,
    stratifyByConcept = TRUE,
    seed = 1
)

Arguments

cdm            A cdm reference created using CDMConnector
ingredientConceptId
               Code indicating the ingredient of interest
sample         A number indicating the size of the random sample to take from the 'person'
               table of the cdm
conceptSetList A concept list that we want to test
stratifyByConcept
               Whether to stratify the result by drug_concept_id
seed           Seed for the random sample

Value

The function returns information of the coverage of computeDailyDose.R for the selected ingredients and concept sets

Examples

library(DrugUtilisation)

cdm <- mockDrugUtilisation()
dailyDoseCoverage(cdm, 1125315)

drugUseColumns Obtain automatically the drug use columns

Description

Obtain automatically the drug use columns

Usage

drugUseColumns(cohort)
generateConceptCohortSet

Arguments

cohort A cohort

Value

Name of the drug use columns

Examples

library(DrugUtilisation)
library(CodelistGenerator)

cdm <- mockDrugUtilisation()
acetaminophen <- getDrugIngredientCodes(cdm, "acetaminophen")
cdm <- generateDrugUtilisationCohortSet(
  cdm, "dus", acetaminophen
)
cdm$dus <- cdm$dus %>%
  addDrugUse(cdm, 1125315, acetaminophen)
drugUseColumns(cdm$dus)

generateConceptCohortSet

Generates a cohort for a certain list of concepts.

Description

Generates a cohort for a certain list of concepts.

Usage

generateConceptCohortSet(
  cdm,
  name,
  conceptSetList,
  daysPriorHistory = 0,
  gap = 0,
  washout = 0,
  offset = 0,
  cohortDateRange = as.Date(c(NA, NA))
)
generateConceptCohortSet

Arguments

cdm
'cdm' object created with CDMConnector::cdm_from_con().

name Name of the GeneratedCohortSet.

conceptSetList Named list of concept sets.

daysPriorHistory Minimum number of days of prior history required for the incident events. If NULL, it is not required that the veent is within the observation period.

gap Number of days between two events to be joined.

washout Prior days of washout without a previous event.

offset Number of days of offset after the cohort_end_date.

cohortDateRange Range for cohort_start_date and cohort_end_date

Value

The function returns the 'cdm' object with the created cohort.

Examples

library(DrugUtilisation)
library(CDMConnector)

cdm <- mockDrugUtilisation()

conditions <- list("headache" = 378253, "asthma" = 317009)

cdm <- generateConceptCohortSet(
  cdm = cdm,
  name = "covariates",
  conceptSetList = conditions,
  daysPriorHistory = 365
)

cdm$covariates

cohortSet(cdm$covariates)

cohortCount(cdm$covariates)

cohortAttrition(cdm$covariates)
**generateDrugUtilisationCohortSet**

Generates a cohort of the drug use of a certain list of concepts.

**Description**

Generates a cohort of the drug use of a certain list of concepts.

**Usage**

```r
generateDrugUtilisationCohortSet(
  cdm,
  name,
  conceptSetList,
  summariseMode = "AllEras",
  fixedTime = NULL,
  daysPriorHistory = 0,
  gapEra = 30,
  priorUseWashout = 0,
  cohortDateRange = as.Date(c(NA, NA)),
  imputeDuration = "eliminate",
  durationRange = c(1, Inf)
)
```

**Arguments**

- `cdm` A cdm_reference object.
- `name` Name of the GeneratedCohortSet
- `conceptSetList` Named list of concept sets.
- `summariseMode` Choice on how to summarize the exposures. There are three options: "Fixed-Time" each individual is followed the exact same number of days specified in 'fixedTime' argument. "AllEras" we summarize the output will be a summary of the exposed eras of each individual. Each individual can contribute multiple times. "FirstEra" we only consider the first observable era of each individual. In this case each individual can not contribute with multiple rows. By default: "AllEras".
- `fixedTime` Time period after first exposure where we summarize the ingredient of interest. Argument only considered if 'summariseMode' = "FixedTime". No default value is provided.
- `daysPriorHistory` Minimum number of days of prior history (observation time) required for the incident eras to be considered. By default: 0, meaning it has to be in observation_period table. When Null, we do not check if in observation_period table.
- `gapEra` Number of days between two continuous exposures to be considered in the same era. By default: 180.
priorUseWashout

Prior days without exposure. By default: NULL.

cohortDateRange

Range for cohort_start_date and cohort_end_date

imputeDuration

Whether/how the duration should be imputed "eliminate", "median", "mean", "quantile25", "quantile75".

durationRange

Range between the duration must be comprised. It should be a numeric vector of length two, with no NAs and the first value should be equal or smaller than the second one. It is only required if imputeDuration = TRUE. If NULL no restrictions are applied. By default: NULL.

Value

The function returns the 'cdm' object with the created tables as references of the object.

Examples

```r
# library(DrugUtilisation)
library(CodelistGenerator)
library(CDMConnector)

cdm <- mockDrugUtilisation()

druglist <- getDrugIngredientCodes(cdm, c("acetaminophen", "metformin"))

cdm <- generateDrugUtilisationCohortSet(
  cdm = cdm,
  name = "drug_cohorts",
  conceptSetList = druglist,
  daysPriorHistory = 365)

cdm$drug_cohorts

cohortSet(cdm$drug_cohorts)

cohortCount(cdm$drug_cohorts)

cohortAttrition(cdm$drug_cohorts)
```

__indicationColumns__

Obtain automatically the indication columns

Description

Obtain automatically the indication columns
**indicationToStrata**

**Usage**

\[ \text{indicationColumns}(x) \]

**Arguments**

- \( x \) - Tibble

**Value**

Name of the indication columns

**Examples**

```r
library(DrugUtilisation)
library(CodelistGenerator)

cdm <- mockDrugUtilisation()
indications <- list("headache" = 378253, "asthma" = 4214676)
cdm <- generateConceptCohortSet(cdm, "indication_cohorts", indications)
acetaminophen <- getDrugIngredientCodes(cdm, "acetaminophen")
cdm <- generateDrugUtilisationCohortSet(cdm, "drug_cohort", acetaminophen)
cdm$drug_cohort <- cdm$drug_cohort %>%
  addIndication(cdm, "indication_cohorts", indicationGap = c(0, 30, 365))

indicationColumns(cdm$drug_cohort)
```

---

**indicationToStrata** - Create new variables summarising the data of indication that can be used as stratification columns

**Description**

Create new variables summarising the data of indication that can be used as stratification columns

**Usage**

```r
indicationToStrata(
  cohort,
  indicationVariables = indicationColumns(cohort),
  keep = FALSE
)
```
Arguments

- cohort: A cohort in the cdm
- indicationVariables: Indication variables that we want to join
- keep: Whether to keep the prior indication variables or not

Value

description: The cohort with the new variable

Examples

```r
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

cdm$cohort1 <- cdm$cohort1 %>%
  addIndication(cdm, "cohort2") %>%
  indicationToStrata()
```

Description

It creates a mock database for testing drugutilisation package

Usage

```r
mockDrugUtilisation(
  connectionDetails = list(con = DBI::dbConnect(duckdb::duckdb(), ":memory:"),
    writeSchema = "main", mockPrefix = NULL),
  numberIndividuals = 10,
  seed = 1,
  concept = NULL,
  concept_ancestor = NULL,
  drug_strength = NULL,
  person = NULL,
  observation_period = NULL,
  drug_exposure = NULL,
  condition_occurrence = NULL,
  observation = NULL,
  extraTables = list(),
  ...
)
```
patternTable

Arguments

  connectionDetails  Details of the connection
  numberIndividuals  Number of individuals in the mock cdm
  seed              Seed for the random numbers
  concept           A concept tibble, if NULL a mock one is created
  concept_ancestor   A concept_ancestor tibble, if NULL a mock one is created
  drug_strength     A drug_strength tibble, if NULL a mock one is created
  person            A person tibble, if NULL a mock one is created
  observation_period A observation_period tibble, if NULL a mock one is created
  drug_exposure     A drug_exposure tibble, if NULL a mock one is created
  condition_occurrence A condition_occurrence tibble, if NULL a mock one is created
  observation       A observation tibble, if NULL a mock one is created
  extraTables       Extra tibbles to be instantiated that are not cohorts or cdm tables
  ...               Cohorts can be added to the cdm reference, cohort1 and cohort2 will be created if not provided

Value

  A cdm reference with the mock tables

Examples

library(DrugUtilisation)

cdm <- mockDrugUtilisation()

cdm

patternTable  

  Function to create a tibble with the patterns from current drug strength table

Description

  Function to create a tibble with the patterns from current drug strength table
readConceptList

Get concept ids from a provided path to json files

Description

Get concept ids from a provided path to json files

Usage

readConceptList(path, cdm)

Arguments

path path to a file or folder containing jsons to be read
cdm A cdm reference created with CDMConnector

Value

list of concept_ids and respective concept_ids of interest
Examples

```r
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

codelist <- readConceptList(
  path = system.file("concepts", package="DrugUtilisation"), cdm = cdm
)
```

---

**reportGenerator**

*Information that connects DrugUtilisation with ReportGenerator*

---

**Description**

Information that connects DrugUtilisation with ReportGenerator

**Usage**

```r
reportGenerator
```

**Format**

List of exported elements

**Examples**

```r
reportGenerator
```

---

**stratifyByUnit**

*Function to stratify a conceptSetList by unit*

---

**Description**

Function to stratify a conceptSetList by unit

**Usage**

```r
stratifyByUnit(conceptSetList, cdm, ingredientConceptId)
```

**Arguments**

- **conceptSetList** List of concept sets
- **cdm** cdm reference
- **ingredientConceptId** ConceptId that refers to an ingredient
summariseCharacteristicsFromCodelist

Value
The conceptSetList stratified by unit

Examples

```r
library(DrugUtilisation)
library(CodelistGenerator)

cdm <- mockDrugUtilisation()

codelist <- getDrugIngredientCodes(cdm, "acetaminophen")

codelistStratified <- stratifyByUnit(codelist, cdm, 1125315)

codelistStratified
```

summariseCharacteristicsFromCodelist

`Summarise a cohort from multiple codelist and windows`

Description

Summarise a cohort from multiple codelist and windows

Usage

```r
summariseCharacteristicsFromCodelist(
  cohort,
  cdm,
  conceptSetList,
  strata = list(),
  window = list(c(-Inf, -366), c(-365, -91), c(-365, -31), c(-90, -1), c(-30, -1), c(0, 0), c(1, 30), c(1, 90), c(31, 365), c(91, 365), c(366, Inf)),
  overlap = TRUE,
  minCellCount = 5
)
```

Arguments

- **cohort**: Cohort to summarise
- **cdm**: cdm_reference
- **conceptSetList**: A list of concept sets
- **strata**: Stratification list
- **window**: Windows to characterize
summariseDrugUse

- overlap: Whether we consider episodes (overlap = TRUE) or incident (overlap = FALSE)
- minCellCount: Minimum cell counts

Value

A SummarisedResults object that contains the characterization

Examples

```r
library(DrugUtilisation)
library(CodelistGenerator)

cdm <- mockDrugUtilisation()

conceptSetList <- list(
  "acetaminophen" = c(1125315, 1125360, 2905077, 43135274),
  "group A" = c(
    3665501, 378253, 317009, 761948, 1539403, 1503327, 1516980, 4141052,
    4313306
  )
)

summariseCharacteristicsFromCodelist(
  cdm$cohort1, cdm, conceptSetList,
  window = list(c(-365, -1), c(0, 0), c(1, 365))
)
```

summariseDrugUse

This function is used to summarise the dose table over multiple cohorts.

Description

This function is used to summarise the dose table over multiple cohorts.

Usage

```r
summariseDrugUse(
  cohort,
  cdm,
  strata = list(),
  drugUseVariables = drugUseColumns(cohort),
  drugUseEstimates = c("mean", "q25", "q75"),
  minCellCount = 5
)
```
Arguments

- `cohort` Cohort with drug use variables and strata
- `cdm` cdm_reference generated by CDMConnector
- `strata` Stratification list
- `drugUseVariables` Name of columns with drug use variables
- `drugUseEstimates` Estimates that we want for the columns
- `minCellCount` Below this number counts will be supressed

Value

A summary of the drug use stratified by cohort_name and strata_name

Examples

```r
library(DrugUtilisation)
library(PatientProfiles)
library(CodelistGenerator)

cdm <- mockDrugUtilisation()
cdm <- generateDrugUtilisationCohortSet(
  cdm, "dus_cohort", getDrugIngredientCodes(cdm, "acetaminophen")
)
cdm$dus_cohort <- cdm$dus_cohort %>%
  addDrugUse(cdm, 1125315)
result <- summariseDrugUse(cdm$dus_cohort, cdm)
print(result)

cdm$dus_cohort <- cdm$dus_cohort %>%
  addSex(cdm) %>%
  addAge(cdm, ageGroup = list("<40" = c(0, 30), ">40" = c(40, 150)))
result <- summariseDrugUse(
  cdm$dus_cohort, cdm, strata = list(
    "Age" = "age_group", "Sex" = "sex", "Age & sex" = c("age_group", "sex")
  )
)
print(result)
```

---

`summariseIndication`  
This function is used to summarise the indication table over multiple cohorts.

Description

This function is used to summarise the indication table over multiple cohorts.
Usage

```r
summariseIndication(
    cohort,
    cdm,
    strata = list(),
    indicationVariables = indicationColumns(cohort),
    minCellCount = 1
)
```

Arguments

- `cohort` Cohort with indications and strata
- `cdm` cdm_reference created by CDMConnector
- `strata` Stratification list
- `indicationVariables` Variables that point to an indication column
- `minCellCount` Minimum counts that a group can have. Cohorts with less counts than this value are obscured. By default: 5.

Value

A Tibble with 4 columns: cohort_definition_id, variable, estimate and value. There will be one row for each cohort, variable and cohort combination.

Examples

```r
library(DrugUtilisation)
library(PatientProfiles)
library(CodelistGenerator)

cdm <- mockDrugUtilisation()
indications <- list("headache" = 378253, "asthma" = 317009)
cdm <- generateConceptCohortSet(cdm, "indication_cohorts", indications)
acetaminophen <- getDrugIngredientCodes(cdm, "acetaminophen")
cdm <- generateDrugUtilisationCohortSet(cdm, "drug_cohort", acetaminophen)

# Add indication

md <- cdm$drug_cohort %>%
  addIndication(cdm, "indication_cohorts", indicationGap = c(0, 30, 365))

summariseIndication(cdm$drug_cohort, cdm)

# Add age

md <- cdm$drug_cohort %>%
  addAge(cdm, ageGroup = list("<40" = c(0, 39), ">=40" = c(40, 150))) %>%
  addSex(cdm)

summariseIndication(cdm$drug_cohort, cdm, strata = list(
  "Age" = "age_group", "Age & Sex" = c("age_group", "sex")
))
```
summariseLargeScaleCharacteristics

This function is used to summarise the large scale characteristics of a cohort table

Description

This function is used to summarise the large scale characteristics of a cohort table

Usage

summariseLargeScaleCharacteristics(
  cohort,
  cdm,
  window = list(c(-Inf, -366), c(-365, -91), c(-90, -1), c(-30, -1), c(0, 0), c(1, 30), c(1, 90), c(31, 365), c(91, 365), c(366, Inf)),
  tablesToCharacterize = c("condition_occurrence", "drug_era", "procedure_occurrence", "measurement"),
  overlap = TRUE,
  minCellCount = 5
)

Arguments

cohort cohort

'cdm' object created with CDMConnector::cdm_from_con(). It must contain the 'targetCohort' table and all the tables that we want to characterize. It is a compulsory input, no default value is provided.

window Temporal windows that we want to characterize. It must be a list of numeric vectors of length two. The tables will be characterized between the first element and the second element respect to the cohort_start_date of each individual. To refer to any time prior set NA the first element of the vector. To refer to any time after the index date set NA the second element of the vector. By default: list(c(NA, -366), c(-365, -90), c(-90, -1), c(-30, -1), c(0, 0), c(1, 30), c(1, 90), c(31, 365), c(91, 365), c(366, NA)).

tablesToCharacterize Name of the tables in the cdm that we want to summarize. The available tables to characterize are: "visit_occurrence", "condition_occurrence", "drug_exposure", "procedure_occurrence", "device_exposure", "measurement", "observation", "drug_era", "condition_era" and "specimen". By default: c("condition_occurrence", "drug_era", "procedure_occurrence", "measurement").

overlap Whether you want to consider overlapping events (overlap = TRUE) or only incident ones (overlap = FALSE).

minCellCount All counts lower than minimumCellCount will be obscured changing its value by NA. 'obscured' column of characterization tibble is TRUE when a count has been obscured. Otherwise it is FALSE.
Value

The output of this function is a 3 elements list. First ("Characterization") is a reference to a temporal table in the database. It contains the characterization of the desired cohorts of interest. The cohorts of interest are specified using 'targetCohortId' and 'targetCohortName'. The characterized tables are the ones specified in 'tablesToChacaterize'. Second ("temporalWindow") contains the windows used to do the characterization. Finally "overlap" is also included in the list.

Examples

```r
library(DrugUtilisation)
cdm <- mockDrugUtilisation()
summariseLargeScaleCharacteristics(
  cohort = cdm$cohort1, cdm = cdm,
  tablesToCharacterize= c("drug_exposure", "condition_occurrence")
)
```

Description

This function is used to summarise the dose and/or indication over multiple cohorts.

Usage

```r
summariseTableOne(
  cohort,
  cdm,
  strata = list(),
  ageGroup = NULL,
  windowVisitOcurrence = NULL,
  covariates = list(),
  minCellCount = 5
)
```

Arguments

- **cohort**: A cohort in the cdm
- **cdm**: A cdm_reference created by CDMConnector
- **strata**: Stratification list
- **ageGroup**: A list of age groups.
windowVisitOccurrence
Window to count visit occurrences.
covariates
Named list of windows to check covariates. The name must point to a cohort-TableName in the cdm.
minCellCount
minimum counts due to obscure

Value
A summary of the characteristics of the individuals

Examples

library(DrugUtilisation)
cdm <- mockDrugUtilisation(numberIndividuals = 100)
summariseTableOne(
  cdm$cohort1,
  cdm,
  ageGroup = list(c(0, 19), c(20, 39), c(40, 59), c(60, 79), c(80, 150)),
  windowVisitOccurrence = c(-180, 0)
)
Index

* datasets
  reportGenerator, 19

addDailyDose, 2
addDrugUse, 3
addIndication, 5
addPattern, 6

computeCohortAttrition, 7
computeCohortCount, 8

dailyDoseCoverage, 9
drugUseColumns, 10

generateConceptCohortSet, 11
generateDrugUtilisationCohortSet, 13

indicationColumns, 14
indicationToStrata, 15

mockDrugUtilisation, 16

patternTable, 17

readConceptList, 18
reportGenerator, 19

stratifyByUnit, 19
summariseCharacteristicsFromCodelist, 20
summariseDrugUse, 21
summariseIndication, 22
summariseLargeScaleCharacteristics, 24
summariseTableOne, 25