Package ‘DrugUtilisation’

March 6, 2024

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

Version 0.5.2

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 7.3.1

Suggests covr, testthat (>= 3.1.5), odbc, here, RPostgres, duckdb, CodelistGenerator, knitr, PaRe, rmarkdown, DT, magick, DiagrammeRsvg, ggplot2, cowplot, plotly, tibble

Config/testthat/edition 3

Imports CDMConnector (>= 1.3.0), checkmate, DBI, dbplyr, dplyr, glue, lubridate, tidyr, rlang, RJSONIO, cli, PatientProfiles (>= 0.6.0), magrittr, purrr, tictoc, omopgenerics (>= 0.0.2), visOmopResults, lifecycle

Depends R (>= 2.10)

LazyData true

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

VignetteBuilder knitr

Config/testthat/parallel true

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>), Kim Lopez-Guell [aut] (<https://orcid.org/0000-0002-8462-8668>), Edward Burn [ctb] (<https://orcid.org/0000-0002-9286-1128>), Xintong Li [ctb] (<https://orcid.org/0000-0002-6872-5804>), Marta Alcalde-Herraiz [ctb] (<https://orcid.org/0009-0002-4405-1814>)
addDailyDose

add daily dose information to a drug_exposure table

Description

add daily dose information to a drug_exposure table

Usage

addDailyDose(
  drugExposure,
  cdm = attr(drugExposure, "cdm_reference"),
  ingredientConceptId
)
**addDrugUse**

**Arguments**

- `drugExposure` it must contain `drug_concept_id`, `quantity`, `drug_exposure_start_date` and `drug_exposure_end_date` as columns
- `cdm` A cdm reference
- `ingredientConceptId` ingredientConceptId for which to filter the drugs of interest

**Value**

same input table

**Examples**

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

cdm <- mockDrugUtilisation()

cdm[["drug_exposure"]]
  %>% filter(drug_concept_id == 2905077)
  %>% addDailyDose(ingredientConceptId = 1125315)
```

---

**Description**

Add new columns with drug use related information

**Usage**

```r
addDrugUse(
  cohort,
  cdm = lifecycle::deprecated(),
  ingredientConceptId,
  conceptSet = NULL,
  duration = TRUE,
  quantity = TRUE,
  dose = TRUE,
  gapEra = 0,
  eraJoinMode = "zero",
  overlapMode = "sum",
  sameIndexMode = "sum",
  imputeDuration = "none",
)```
addDrugUse

imputeDailyDose = "none",
durationRange = c(1, Inf),
dailyDoseRange = c(0, Inf)
)

Arguments

cohort Cohort in the cdm
cdm deprecated
ingredientConceptId Ingredient OMOP concept that we are interested for the study. It is a compulsory input, no default value is provided.
conceptSet List of concepts to be included. If NULL all the descendants of ingredient concept id will be used.
duration Whether to add duration related columns.
quantity Whether to add quantity related columns.
dose Whether to add dose related columns.
gapEra Number of days between two continuous exposures to be considered in the same era.
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.
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.
sameIndexMode How the overlapping between two exposures that start on the same day is solved inside a subexposure. There are three 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.
addIndication

get indication for a target cohort

**Description**

Get indication for a target cohort

**Usage**

```r
addIndication(
  x,
  cdm = lifecycle::deprecated(),
  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, indications, "indication_cohorts"
)

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


```

```
addRoute(drugTable, cdm = attr(drugTable, "cdm_reference"))
```

addRoute  

**addRoute**  

*add route column to a table containing drug_exposure information*

Description

add route column to a table containing drug_exposure information

Usage

```
addRoute(drugTable, cdm = attr(drugTable, "cdm_reference"))
```
**benchmarkDUS**

**Arguments**

- **drugTable**
  - Table in the cdm that must contain drug_concept_id
- **cdm**
  - 'cdm' object created with CDMConnector::cdm_from_con(). It must contain 'concept_relationship' table.

**Value**

- It adds route to the current table

**Examples**

```
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

cdm[['drug_exposure']] %>%
  addRoute()
```

---

**benchmarkDUS**

*Run benchmark of drug utilisation cohort generation*

**Description**

Run benchmark of drug utilisation cohort generation

**Usage**

```
benchmarkDUS(
  cdm,  
  numberOfCohort = 1:4,  
  indicationCohortName = "cohort1",  
  ingredientId = 1125315,  
  drugExposureName = "drug_exposure"
)
```

**Arguments**

- **cdm**
  - A CDM reference object
- **numberOfCohort**
  - Number of cohort to generate for benchmarking. An integer or a vector of integers
- **indicationCohortName**
  - Name of indication cohort table
ingredientId  Ingredient OMOP concept that we are interested for the study. It is a compulsory input, no default value is provided.

drugExposureName  Name of drug_exposure table in cdm, the table must contain drug_concept_id, quantity, drug_exposure_start_date and drug_exposure_end_date as columns

Value  
a tibble with time taken for different analyses

Examples

cdm <- DrugUtilisation::mockDrugUtilisation()
timings <- DrugUtilisation::benchmarkDUS(cdm)

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)

Arguments

cdm A cdm reference created using CDMConnector

ingredientConceptId  Code indicating the ingredient of interest

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

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

**Examples**

```r
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

dailyDoseCoverage(cdm, 1125315)
```

**Description**

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

**Usage**

```r
generateAtcCohortSet(
  cdm,
  name,
  atcName = NULL,
  durationRange = c(1, Inf),
  imputeDuration = "none",
  gapEra = 0,
  priorUseWashout = 0,
  priorObservation = 0,
  cohortDateRange = as.Date(c(NA, NA)),
  limit = "all",
  level = c("ATC 1st"),
  doseForm = NULL
)
```

**Arguments**

- **cdm**: A cdm_reference object.
- **name**: Name of the GeneratedCohortSet
- **atcName**: Names of ATC of interest.
- **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.
- **imputeDuration**: Whether/how the duration should be imputed "none", "median", "mean", "mode", or it can be a count
generateDrugUtilisationCohortSet

**Description**

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

**Value**

The function returns the `cdm` object with the created cohorts as references of the object.

**Examples**

```r
library(DrugUtilisation)
cdm <- mockDrugUtilisation()
cdm <- generateAtcCohortSet(cdm, name = "test")
cdm
cdm$test
settings(cdm$test)
```
Usage

generateDrugUtilisationCohortSet(
  cdm,
  name,
  conceptSet,
  durationRange = c(1, Inf),
  imputeDuration = "none",
  gapEra = 0,
  priorUseWashout = 0,
  priorObservation = 0,
  cohortDateRange = as.Date(c(NA, NA)),
  limit = "all"
)

Arguments

cdm A cdm_reference object.
name Name of the GeneratedCohortSet
conceptSet Named list of concept sets.
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.
imputeDuration Whether/how the duration should be imputed "none", "median", "mean", "mode", or it can be a count
gapEra Number of days between two continuous exposures to be considered in the same era.
priorUseWashout Prior days without exposure.
priorObservation Minimum number of days of prior observation required for the incident eras to be considered.
cohortDateRange Range for cohort_start_date and cohort_end_date
limit Choice on how to summarise the exposures. There are two options: "all" we summarise the output will be a summary of the exposed eras of each individual. Each individual can contribute multiple times. "first" we only consider the first observable era of each individual that fulfills the criteria provided in previous parameters. In this case each individual can not contribute with multiple rows.

Value

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

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

cdm <- mockDrugUtilisation()

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

cdm <- generateDrugUtilisationCohortSet(
  cdm = cdm,
  name = "drug_cohorts",
  conceptSet = druglist,
  priorObservation = 365
)

cdm[["drug_cohorts"]]

settings(cdm[["drug_cohorts"]])

cohortCount(cdm[["drug_cohorts"]])

attrition(cdm[["drug_cohorts"]])
```

---

**generateIngredientCohortSet**

*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
generateIngredientCohortSet(
  cdm, name, ingredient = NULL,
  durationRange = c(1, Inf), imputeDuration = "none",
  gapEra = 0, priorUseWashout = 0, priorObservation = 0,
  cohortDateRange = as.Date(c(NA, NA)),
  limit = "all",
)```
```r
doseForm = NULL,
ingredientRange = c(1, Inf)
)

Arguments

- **cdm**
  A cdm_reference object.

- **name**
  Name of the GeneratedCohortSet

- **ingredient**
  Names of ingredients of interest. For example, c("acetaminophen", "codeine"), would result in a list of length two with the descendant concepts for these two particular drug ingredients.

- **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.

- **imputeDuration**
  Whether/how the duration should be imputed "none", "median", "mean", "mode", or it can be a count

- **gapEra**
  Number of days between two continuous exposures to be considered in the same era.

- **priorUseWashout**
  Prior days without exposure.

- **priorObservation**
  Minimum number of days of prior observation required for the incident eras to be considered.

- **cohortDateRange**
  Range for cohort_start_date and cohort_end_date

- **limit**
  Choice on how to summarise the exposures. There are two options: "all" we summarise the output will be a summary of the exposed eras of each individual. Each individual can contribute multiple times. "first" we only consider the first observable era of each individual that fulfills the criteria provided in previous parameters. In this case each individual can not contribute with multiple rows.

- **doseForm**
  Only descendants codes with the specified dose form will be returned. If NULL, descendant codes will be returned regardless of dose form.

- **ingredientRange**
  Used to restrict descendant codes to those associated with a specific number of ingredients. Must be a vector of length two with the first element the minimum number of ingredients allowed and the second the maximum. A value of c(2, 2) would restrict to only concepts associated with two ingredients.

Value

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

library(DrugUtilisation)
cdm <- mockDrugUtilisation()
cdm <- generateIngredientCohortSet(
  cdm = cdm,
  ingredient = "acetaminophen",
  name = "test"
)
cdm

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

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

library(DrugUtilisation)
cdm <- mockDrugUtilisation()
cdm["cohort1"] <- cdm["cohort1"] %>%
  addIndication(cdm, "cohort2") %>%
It creates a mock database for testing drugutilisation package

Usage

```r
mockDrugUtilisation(
  connectionDetails = list(con = DBI::dbConnect(duckdb::duckdb(), "::memory:"),
    writeSchema = "main"),
  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,
  concept_relationship = NULL,
  extraTables = list(),
  ...
)
```

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

## Description

It creates a mock database for testing drugutilisation package
patternsWithFormula

Patterns valid to compute daily dose with the associated formula.

Description

Patterns valid to compute daily dose with the associated formula.

Usage

patternsWithFormula

Format

A data frame with eight variables: pattern_id, amount, amount_unit, numerator, numerator_unit, denominator, denominator_unit, formula_name and formula.
**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

**Usage**

`patternTable(cdm)`

**Arguments**

- `cdm` 'cdm' object created with CDMConnector::cdm_from_con(). It must must contain 'drug_strength' and 'concept' tables.

**Value**

The function creates a tibble with the different patterns found in the table, plus a column of potentially valid and invalid combinations.

**Examples**

```r
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

patternTable(cdm)
```

**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

library(DrugUtilisation)

cdm <- mockDrugUtilisation()

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

stratifyByUnit(conceptSet, cdm, ingredientConceptId)

Arguments

  conceptSet   List of concept sets
  cdm          cdm reference
  ingredientConceptId  ConceptId that refers to an ingredient

Value

  The conceptSet stratified by unit

Examples

library(DrugUtilisation)
library(CodelistGenerator)

cdm <- mockDrugUtilisation()

codelist <- getDrugIngredientCodes(cdm, "acetaminophen")
**summariseCharacteristicsFromCodelist**

```r
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 = lifecycle::deprecated(),
  conceptSet,
  strata = list(),
  window = list(c(-Inf, -366), c(-365, -31), c(-30, -1), c(0, 0), c(1, 30), c(31, 365),
                c(366, Inf)),
  overlap = TRUE,
  minCellCount = lifecycle::deprecated()
)
```

**Arguments**

- `cohort`  
  Cohort to summarise
- `cdm`  
  `cdm_reference`
- `conceptSet`  
  A list of concept sets
- `strata`  
  Stratification list
- `window`  
  Windows to characterize
- `overlap`  
  Whether we consider episodes (overlap = TRUE) or incident (overlap = FALSE)
- `minCellCount`  
  Minimum cell counts

**Value**

A `SummarisedResults` object that contains the characterization
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 = lifecycle::deprecated(),
  strata = list(),
  drugUseEstimates = c("min", "q05", "q25", "median", "q75", "q95", "max", "mean", "sd", "missing"),
  minCellCount = lifecycle::deprecated()
)
```

Arguments

- **cohort**: Cohort with drug use variables and strata
- **cdm**: cdm_reference generated by CDMConnector
- **strata**: Stratification list
- **drugUseEstimates**: Estimates that we want for the columns
- **minCellCount**: Below this number counts will be suppressed

Value

A summary of the drug use stratified by cohort_name and strata_name

Examples

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

cdm <- mockDrugUtilisation()

# Generate drug utilisation cohort set
cdm <- generateDrugUtilisationCohortSet(cdm, "dus_cohort", getDrugIngredientCodes(cdm, "acetaminophen"))

# Add drug use
cdm["dus_cohort"] <- cdm["dus_cohort"] %>%
  addDrugUse(cdm, 1125315)

result <- summariseDrugUse(cdm["dus_cohort"], cdm)
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 = lifecycle::deprecated(),
  strata = list(),
  minCellCount = lifecycle::deprecated()
)
```

**Arguments**

- **cohort**
  - Cohort with indications and strata
- **cdm**
  - cdm_reference created by CDMConnector
- **strata**
  - Stratification list
- **minCellCount**
  - Minimum counts that a group can have. Cohorts with less counts than this value are obscured.

**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.
summariseTreatmentFromCohort

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

summariseTreatmentFromCohort(cohort, strata = list(), window, treatmentCohortName, treatmentCohortId = NULL, combination = FALSE, minCellCount = 5)
summariseTreatmentFromCohort

Arguments

- **cohort**: Cohort with drug use variables and strata.
- **strata**: Stratification list.
- **window**: Window where to summarise the treatments.
- **treatmentCohortName**: Name of a cohort in the cdm that contains the interest treatments.
- **treatmentCohortId**: Cohort definition id of interest from treatmentCohortName.
- **combination**: Whether to include combination treatments.
- **minCellCount**: Below this number counts will be suppressed.

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)))

summariseDrugUse(
  cdm["dus_cohort"], cdm, strata = list(
    "age_group" = "age_group", "sex" = "sex",
    "age_group and sex" = c("age_group", "sex")
  )
)
```
summariseTreatmentFromConceptSet

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
summariseTreatmentFromConceptSet(
  cohort, 
  strata = list(),
  window, 
  treatmentConceptSet, 
  combination = FALSE, 
  minCellCount = 5
)
```

Arguments

- `cohort`: Cohort with drug use variables and strata.
- `strata`: Stratification list.
- `window`: Window where to summarise the treatments.
- `treatmentConceptSet`: Concept set list to summarise.
- `combination`: Whether to include combination treatments.
- `minCellCount`: Below this number counts will be suppressed.

Value

A summary of the drug use stratified by cohort_name and strata_name
Index

* datasets
  patternsWithFormula, 16

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

benchmarkDUS, 7
dailyDoseCoverage, 8
generateAtcCohortSet, 9
generateDrugUtilisationCohortSet, 10
generateIngredientCohortSet, 12

indicationToStrata, 14
mockDrugUtilisation, 15

patternsWithFormula, 16
patternTable, 17

readConceptList, 17

stratifyByUnit, 18
summariseCharacteristicsFromCodelist, 19
summariseDrugUse, 20
summariseIndication, 21
summariseTreatmentFromCohort, 22
summariseTreatmentFromConceptSet, 24