

# Package ‘DrugUtilisation’

July 29, 2024

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

**Version** 0.7.0

**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 ( $\geq 2$ )

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Suggests** covr, testthat ( $\geq 3.1.5$ ), odbc, CohortSurvival, here, RPostgres, duckdb, knitr, rmarkdown, ggplot2, tibble, gt, flextable, ggtext, scales, tictoc, DBI

**Config/testthat/edition** 3

**Imports** CDMConnector ( $\geq 1.4.0$ ), CodelistGenerator ( $\geq 3.1.0$ ), checkmate, dbplyr, dplyr, glue, clock, tidyr, rlang, cli, PatientProfiles ( $\geq 1.0.0$ ), purrr, omopgenerics ( $\geq 0.2.3$ ), visOmopResults, lifecycle, stringr

**Depends** R ( $\geq 2.10$ )

**LazyData** true

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

**Config/testthat/parallel** true

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Martí Català [aut, cre] (<<https://orcid.org/0000-0003-3308-9905>>),  
Mike Du [ctb] (<<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 [aut] (<<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>>),

Nuria Mercade-Besora [aut] (<<https://orcid.org/0009-0006-7948-3747>>),  
 Xihang Chen [aut] (<<https://orcid.org/0009-0001-8112-8959>>)

**Maintainer** Martí Català <[marti.catalasabate@dorms.ox.ac.uk](mailto:marti.catalasabate@dorms.ox.ac.uk)>

**Repository** CRAN

**Date/Publication** 2024-07-29 18:20:02 UTC

## Contents

addCumulativeDose . . . . .	3
addCumulativeQuantity . . . . .	4
addDailyDose . . . . .	5
addDrugUse . . . . .	6
addDrugUtilisation . . . . .	8
addExposedTime . . . . .	10
addIndication . . . . .	11
addInitialDailyDose . . . . .	12
addInitialQuantity . . . . .	14
addNumberEras . . . . .	15
addNumberExposures . . . . .	16
addRoute . . . . .	17
addTimeToExposure . . . . .	18
benchmarkDrugUtilisation . . . . .	19
cohortGapEra . . . . .	20
dailyDoseCoverage . . . . .	21
defaultTableOptions . . . . .	21
generateAtcCohortSet . . . . .	22
generateDrugUtilisationCohortSet . . . . .	23
generateIngredientCohortSet . . . . .	25
mockDrugUtilisation . . . . .	27
patternsWithFormula . . . . .	28
patternTable . . . . .	28
plotDrugRestart . . . . .	29
plotIndication . . . . .	30
plotProportionOfPatientsCovered . . . . .	31
plotTreatment . . . . .	32
readConceptList . . . . .	33
requireDrugInDateRange . . . . .	33
requireIsFirstDrugEntry . . . . .	34
requireObservationBeforeDrug . . . . .	35
requirePriorDrugWashout . . . . .	36
stratifyByUnit . . . . .	37
summariseDoseCoverage . . . . .	38
summariseDrugRestart . . . . .	39
summariseDrugUse . . . . .	40
summariseDrugUtilisation . . . . .	42
summariseIndication . . . . .	43
summariseProportionOfPatientsCovered . . . . .	45

*addCumulativeDose* 3

summariseTreatment . . . . .	46
tableDoseCoverage . . . . .	47
tableDrugRestart . . . . .	48
tableDrugUtilisation . . . . .	49
tableIndication . . . . .	51
tableProportionOfPatientsCovered . . . . .	52
tableTreatment . . . . .	53

**Index** 55

---

*addCumulativeDose*      *To add a new column with the cumulative dose. To add multiple columns use addDrugUtilisation() for efficiency.*

---

**Description**

To add a new column with the cumulative dose. To add multiple columns use addDrugUtilisation() for efficiency.

**Usage**

```
addCumulativeDose(  
  cohort,  
  ingredientConceptId,  
  conceptSet = NULL,  
  indexDate = "cohort_start_date",  
  censorDate = "cohort_end_date",  
  restrictIncident = TRUE,  
  nameStyle = "cumulative_dose_{concept_name}_{ingredient}",  
  name = NULL  
)
```

**Arguments**

<code>cohort</code>	Cohort in the cdm.
<code>ingredientConceptId</code>	Ingredient OMOP concept that we are interested for the study. It is a compulsory input, no default value is provided.
<code>conceptSet</code>	List of concepts to be included. If NULL all the descendants of ingredient concept id will be used.
<code>indexDate</code>	Name of a column that indicates the date to start the analysis.
<code>censorDate</code>	Name of a column that indicates the date to stop the analysis, if NULL end of individuals observation is used.
<code>restrictIncident</code>	Whether to include only incident prescriptions in the analysis. If FALSE all prescriptions that overlap with the study period will be included.
<code>nameStyle</code>	Character string to specify the nameStyle of the new columns.
<code>name</code>	Name of the new computed cohort table, if NULL a temporary tables is created.

**Value**

The same cohort with the added column.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

cdm$cohort1 |>
  addCumulativeDose(ingredientConceptId = 1125315)
```

---

`addCumulativeQuantity` *To add a new column with the cumulative quantity. To add multiple columns use `addDrugUtilisation()` for efficiency.*

---

**Description**

To add a new column with the cumulative quantity. To add multiple columns use `addDrugUtilisation()` for efficiency.

**Usage**

```
addCumulativeQuantity(
  cohort,
  conceptSet,
  indexDate = "cohort_start_date",
  censorDate = "cohort_end_date",
  restrictIncident = TRUE,
  nameStyle = "cumulative_quantity_{concept_name}",
  name = NULL
)
```

**Arguments**

<code>cohort</code>	Cohort in the cdm.
<code>conceptSet</code>	List of concepts to be included.
<code>indexDate</code>	Name of a column that indicates the date to start the analysis.
<code>censorDate</code>	Name of a column that indicates the date to stop the analysis, if NULL end of individuals observation is used.
<code>restrictIncident</code>	Whether to include only incident prescriptions in the analysis. If FALSE all prescriptions that overlap with the study period will be included.
<code>nameStyle</code>	Character string to specify the nameStyle of the new columns.
<code>name</code>	Name of the new computed cohort table, if NULL a temporary tables is created.

**Value**

The same cohort with the added column.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(
  cdm,
  name = "acetaminophen"
)
cdm <- generateDrugUtilisationCohortSet(
  cdm = cdm, name = "dus_cohort", conceptSet = codelist
)

cdm$dus_cohort |>
  addCumulativeQuantity(conceptSet = codelist)
```

---

 addDailyDose

*add daily dose information to a drug\_exposure table*


---

**Description**

**[Deprecated]**

**Usage**

```
addDailyDose(drugExposure, ingredientConceptId, name = NULL)
```

**Arguments**

**drugExposure** drugExposure it must contain drug\_concept\_id, quantity, drug\_exposure\_start\_date and drug\_exposure\_end\_date as columns

**ingredientConceptId** ingredientConceptId for which to filter the drugs of interest

**name** Name of the computed table, if NULL a temporary table will be generated.

**Value**

same input table

## Examples

```
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

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

---

addDrugUse

*Add new columns with drug use related information*

---

## Description

**[Deprecated]**

## Usage

```
addDrugUse(
  cohort,
  cdm = lifecycle::deprecated(),
  ingredientConceptId,
  conceptSet = NULL,
  duration = TRUE,
  quantity = TRUE,
  dose = TRUE,
  gapEra = 0,
  eraJoinMode = "zero",
  overlapMode = "sum",
  sameIndexMode = "sum",
  imputeDuration = "none",
  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.
imputeDuration	Whether/how the duration should be imputed "none", "median", "mean", "mode" or a number
imputeDailyDose	Whether/how the daily_dose should be imputed "none", "median", "mean", "mode" or a number
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 must not be NULL if imputeDuration is not "none". If NULL no restrictions are applied.
dailyDoseRange	Range between the daily_dose 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 must not be NULL if imputeDailyDose is not "none". If NULL no restrictions are applied.

**Value**

The same cohort with the added columns.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(
  cdm,
  name = "acetaminophen"
)

cdm <- generateDrugUtilisationCohortSet(cdm, "dus_cohort", codelist)
cdm[["dus_cohort"]] |>
  addDrugUse(ingredientConceptId = 1125315)
```

---

addDrugUtilisation      *Add new columns with drug use related information*

---

**Description**

Add new columns with drug use related information

**Usage**

```
addDrugUtilisation(
  cohort,
  gapEra,
  conceptSet = NULL,
  ingredientConceptId = NULL,
  indexDate = "cohort_start_date",
  censorDate = "cohort_end_date",
  restrictIncident = TRUE,
  numberExposures = TRUE,
  numberEras = TRUE,
  exposedTime = TRUE,
  timeToExposure = TRUE,
  initialQuantity = TRUE,
  cumulativeQuantity = TRUE,
  initialDailyDose = TRUE,
  cumulativeDose = TRUE,
  nameStyle = "{value}_{concept_name}_{ingredient}",
  name = NULL
)
```



**Arguments**

cohort	Cohort in the cdm
gapEra	Number of days between two continuous exposures to be considered in the same era.
conceptSet	List of concepts to be included. If NULL all the descendants of ingredient concept id will be used.
ingredientConceptId	Ingredient OMOP concept that we are interested for the study. It is a compulsory input, no default value is provided.
indexDate	Name of a column that indicates the date to start the analysis.
sensorDate	Name of a column that indicates the date to stop the analysis, if NULL end of individuals observation is used.
restrictIncident	Whether to include only incident prescriptions in the analysis. If FALSE all prescriptions that overlap with the study period will be included.
numberExposures	Whether to add a column with the number of exposures.
numberEras	Whether to add a column with the number of eras.
exposedTime	Whether to add a column with the number of exposed days.
timeToExposure	Whether to add a column with the number of days between indexDate and start of the first exposure.
initialQuantity	Whether to add a column with the initial quantity.
cumulativeQuantity	Whether to add a column with the cumulative quantity of the identified prescription.
initialDailyDose	Whether to add a column with the initial daily dose.
cumulativeDose	Whether to add a column with the cumulative dose.
nameStyle	Character string to specify the nameStyle of the new columns.
name	Name of the new computed cohort table, if NULL a temporary tables is created.

**Value**

The same cohort with the added columns.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(
  cdm,
  name = "acetaminophen"
)
```

```

cdm <- generateDrugUtilisationCohortSet(cdm, "dus_cohort", codelist)
cdm[["dus_cohort"]] |>
  addDrugUtilisation(ingredientConceptId = 1125315, gapEra = 30)

```

---

addExposedTime	<i>To add a new column with the exposed time. To add multiple columns use addDrugUtilisation() for efficiency.</i>
----------------	--

---

### Description

To add a new column with the exposed time. To add multiple columns use addDrugUtilisation() for efficiency.

### Usage

```

addExposedTime(
  cohort,
  conceptSet,
  gapEra,
  indexDate = "cohort_start_date",
  censorDate = "cohort_end_date",
  restrictIncident = TRUE,
  nameStyle = "exposed_time_{concept_name}",
  name = NULL
)

```

### Arguments

cohort	Cohort in the cdm.
conceptSet	List of concepts to be included.
gapEra	Number of days between two continuous exposures to be considered in the same era.
indexDate	Name of a column that indicates the date to start the analysis.
censorDate	Name of a column that indicates the date to stop the analysis, if NULL end of individuals observation is used.
restrictIncident	Whether to include only incident prescriptions in the analysis. If FALSE all prescriptions that overlap with the study period will be included.
nameStyle	Character string to specify the nameStyle of the new columns.
name	Name of the new computed cohort table, if NULL a temporary tables is created.

### Value

The same cohort with the added column.

**Examples**

```

library(DrugUtilisation)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(
  cdm,
  name = "acetaminophen"
)
cdm <- generateDrugUtilisationCohortSet(
  cdm = cdm, name = "dus_cohort", conceptSet = codelist
)

cdm$dus_cohort |>
  addExposedTime(conceptSet = codelist, gapEra = 1)

```

---

addIndication

*Add a variable indicating individuals indications*


---

**Description**

Add a variable to a drug cohort indicating their presence in an indication cohort in a specified time window. If an individual is not in one of the indication cohorts, they will be considered to have an unknown indication if they are present in one of the specified OMOP CDM clinical tables. If they are neither in an indication cohort or a clinical table they will be considered as having no observed indication.

**Usage**

```

addIndication(
  cohort,
  indicationCohortName,
  indicationCohortId = NULL,
  indicationWindow = list(c(0, 0)),
  unknownIndicationTable = NULL,
  indexDate = "cohort_start_date",
  censorDate = NULL,
  name = NULL
)

```

**Arguments**

cohort	A cohort table in the cdm.
indicationCohortName	Name of indication cohort table
indicationCohortId	target cohort Id to add indication

indicationWindow	time window of interests
unknownIndicationTable	Tables to search unknown indications
indexDate	Date respect to indication will be calculated.
sensorDate	After that day no indication will be considered.
name	name of permanant table

**Value**

The original table with a variable added that summarises the individual's indications.

**Examples**

```
library(DrugUtilisation)
library(CDMConnector)
library(dplyr)

cdm <- mockDrugUtilisation()

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

cdm <- generateIngredientCohortSet(
  cdm = cdm, name = "drug_cohort",
  ingredient = "acetaminophen"
)

cdm$drug_cohort |>
  addIndication("indication_cohorts", indicationWindow = list(c(0, 0))) |>
  glimpse()
```

---

addInitialDailyDose    *To add a new column with the initial daily dose. To add multiple columns use addDrugUtilisation() for efficiency.*

---

**Description**

To add a new column with the initial daily dose. To add multiple columns use addDrugUtilisation() for efficiency.

**Usage**

```
addInitialDailyDose(
  cohort,
  ingredientConceptId,
  conceptSet = NULL,
  indexDate = "cohort_start_date",
  censorDate = "cohort_end_date",
  restrictIncident = TRUE,
  nameStyle = "initial_daily_dose_{concept_name}_{ingredient}",
  name = NULL
)
```

**Arguments**

cohort	Cohort in the cdm.
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.
indexDate	Name of a column that indicates the date to start the analysis.
censorDate	Name of a column that indicates the date to stop the analysis, if NULL end of individuals observation is used.
restrictIncident	Whether to include only incident prescriptions in the analysis. If FALSE all prescriptions that overlap with the study period will be included.
nameStyle	Character string to specify the nameStyle of the new columns.
name	Name of the new computed cohort table, if NULL a temporary tables is created.

**Value**

The same cohort with the added column.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

cdm$cohort1 |>
  addInitialDailyDose(ingredientConceptId = 1125315)
```

---

addInitialQuantity      *To add a new column with the initial quantity. To add multiple columns use addDrugUtilisation() for efficiency.*

---

### Description

To add a new column with the initial quantity. To add multiple columns use addDrugUtilisation() for efficiency.

### Usage

```
addInitialQuantity(
  cohort,
  conceptSet,
  indexDate = "cohort_start_date",
  censorDate = "cohort_end_date",
  restrictIncident = TRUE,
  nameStyle = "initial_quantity_{concept_name}",
  name = NULL
)
```

### Arguments

cohort	Cohort in the cdm.
conceptSet	List of concepts to be included.
indexDate	Name of a column that indicates the date to start the analysis.
censorDate	Name of a column that indicates the date to stop the analysis, if NULL end of individuals observation is used.
restrictIncident	Whether to include only incident prescriptions in the analysis. If FALSE all prescriptions that overlap with the study period will be included.
nameStyle	Character string to specify the nameStyle of the new columns.
name	Name of the new computed cohort table, if NULL a temporary tables is created.

### Value

The same cohort with the added column.

### Examples

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(
  cdm,
  name = "acetaminophen"
```

```

)
cdm <- generateDrugUtilisationCohortSet(
  cdm = cdm, name = "dus_cohort", conceptSet = codelist
)

cdm$dus_cohort |>
  addInitialQuantity(conceptSet = codelist)

```

---

addNumberEras	<i>To add a new column with the number of eras. To add multiple columns use addDrugUtilisation() for efficiency.</i>
---------------	--

---

### Description

To add a new column with the number of eras. To add multiple columns use addDrugUtilisation() for efficiency.

### Usage

```

addNumberEras(
  cohort,
  conceptSet,
  gapEra,
  indexDate = "cohort_start_date",
  censorDate = "cohort_end_date",
  restrictIncident = TRUE,
  nameStyle = "number_eras_{concept_name}",
  name = NULL
)

```

### Arguments

cohort	Cohort in the cdm.
conceptSet	List of concepts to be included.
gapEra	Number of days between two continuous exposures to be considered in the same era.
indexDate	Name of a column that indicates the date to start the analysis.
censorDate	Name of a column that indicates the date to stop the analysis, if NULL end of individuals observation is used.
restrictIncident	Whether to include only incident prescriptions in the analysis. If FALSE all prescriptions that overlap with the study period will be included.
nameStyle	Character string to specify the nameStyle of the new columns.
name	Name of the new computed cohort table, if NULL a temporary tables is created.

**Value**

The same cohort with the added column.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(
  cdm,
  name = "acetaminophen"
)
cdm <- generateDrugUtilisationCohortSet(
  cdm = cdm, name = "dus_cohort", conceptSet = codelist
)

cdm$dus_cohort |>
  addNumberEras(conceptSet = codelist, gapEra = 1)
```

---

addNumberExposures	<i>To add a new column with the number of exposures. To add multiple columns use addDrugUtilisation() for efficiency.</i>
--------------------	---

---

**Description**

To add a new column with the number of exposures. To add multiple columns use addDrugUtilisation() for efficiency.

**Usage**

```
addNumberExposures(
  cohort,
  conceptSet,
  indexDate = "cohort_start_date",
  censorDate = "cohort_end_date",
  restrictIncident = TRUE,
  nameStyle = "number_exposures_{concept_name}",
  name = NULL
)
```

**Arguments**

cohort	Cohort in the cdm
conceptSet	List of concepts to be included.
indexDate	Name of a column that indicates the date to start the analysis.



sensorDate	Name of a column that indicates the date to stop the analysis, if NULL end of individuals observation is used.
restrictIncident	Whether to include only incident prescriptions in the analysis. If FALSE all prescriptions that overlap with the study period will be included.
nameStyle	Character string to specify the nameStyle of the new columns.
name	Name of the new computed cohort table, if NULL a temporary tables is created.

**Value**

The same cohort with the added columns.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(
  cdm,
  name = "acetaminophen"
)
cdm <- generateDrugUtilisationCohortSet(
  cdm = cdm, name = "dus_cohort", conceptSet = codelist
)

cdm$dus_cohort |>
  addNumberExposures(conceptSet = codelist)
```

---

addRoute	<i>add route column to a table containing drug_exposure information</i>
----------	---

---

**Description**

**[Deprecated]**

**Usage**

```
addRoute(drugTable)
```

**Arguments**

drugTable      Table in the cdm that must contain drug\_concept\_id

**Value**

It adds route to the current table

**Examples**

```
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

cdm[["drug_exposure"]] |>
  addRoute()
```

---

addTimeToExposure	<i>To add a new column with the time to exposure. To add multiple columns use addDrugUtilisation() for efficiency.</i>
-------------------	--

---

**Description**

To add a new column with the time to exposure. To add multiple columns use addDrugUtilisation() for efficiency.

**Usage**

```
addTimeToExposure(
  cohort,
  conceptSet,
  indexDate = "cohort_start_date",
  censorDate = "cohort_end_date",
  restrictIncident = TRUE,
  nameStyle = "time_to_exposure_{concept_name}",
  name = NULL
)
```

**Arguments**

cohort	Cohort in the cdm.
conceptSet	List of concepts to be included.
indexDate	Name of a column that indicates the date to start the analysis.
censorDate	Name of a column that indicates the date to stop the analysis, if NULL end of individuals observation is used.
restrictIncident	Whether to include only incident prescriptions in the analysis. If FALSE all prescriptions that overlap with the study period will be included.
nameStyle	Character string to specify the nameStyle of the new columns.
name	Name of the new computed cohort table, if NULL a temporary tables is created.

**Value**

The same cohort with the added column.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(
  cdm,
  name = "acetaminophen"
)
cdm <- generateDrugUtilisationCohortSet(
  cdm = cdm, name = "dus_cohort", conceptSet = codelist
)

cdm$dus_cohort |>
  addTimeToExposure(conceptSet = codelist)
```

---

benchmarkDrugUtilisation

*Run benchmark of drug utilisation cohort generation*

---

**Description**

Run benchmark of drug utilisation cohort generation

**Usage**

```
benchmarkDrugUtilisation(
  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

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

timings <- benchmarkDrugUtilisation(cdm)

timings
```

---

cohortGapEra	<i>Get the gapEra used to create a cohort</i>
--------------	---

---

### Description

Get the gapEra used to create a cohort

### Usage

```
cohortGapEra(cohort, cohortId = NULL)
```

### Arguments

cohort	A cohort_table object.
cohortId	Integer vector referring to cohortIds from cohort. If NULL all cohort definition ids in settings will be used.

### Value

gapEra values for the specific cohortIds

### Examples

```
library(CDMConnector)
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

druglist <- CodelistGenerator::getDrugIngredientCodes(
```

```

    cdm, c("acetaminophen", "metformin")
  )

  cdm <- generateDrugUtilisationCohortSet(
    cdm = cdm,
    name = "drug_cohorts",
    conceptSet = druglist,
    gapEra = 100
  )

  cohortGapEra(cdm$drug_cohorts)

```

---

dailyDoseCoverage	<i>Check coverage of daily dose computation in a sample of the cdm for selected concept sets and ingredient</i>
-------------------	---

---

## Description

**[Deprecated]**

## 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

---

defaultTableOptions	<i>Additional arguments for the table functions.</i>
---------------------	--

---

## Description

It provides a list of allowed inputs for .option argument in table functions, and their given default values.

**Usage**

```
defaultTableOptions()
```

**Value**

The default .options named list.

**Examples**

```
{
  defaultTableOptions()
}
```

---

generateAtcCohortSet    *Generate a set of drug cohorts based on ATC classification*

---

**Description**

Adds a new cohort table to the cdm reference with individuals who have drug exposure records that belong to the specified Anatomical Therapeutic Chemical (ATC) classification. Cohort start and end dates will be based on drug record start and end dates, respectively. Records that overlap or have fewer days between them than the specified gap era will be concatenated into a single cohort entry.

**Usage**

```
generateAtcCohortSet(
  cdm,
  name,
  atcName = NULL,
  level = c("ATC 1st"),
  doseForm = NULL,
  gapEra = 1,
  durationRange = lifecycle::deprecated(),
  imputeDuration = lifecycle::deprecated(),
  priorUseWashout = lifecycle::deprecated(),
  priorObservation = lifecycle::deprecated(),
  cohortDateRange = lifecycle::deprecated(),
  limit = lifecycle::deprecated()
)
```

**Arguments**

cdm	A cdm reference.
name	The name of the new cohort table to add to the cdm reference.
atcName	Names of ATC classification of interest.

level	ATC level. Can be one or more of "ATC 1st", "ATC 2nd", "ATC 3rd", "ATC 4th", and "ATC 5th"
doseForm	Only descendants codes with the specified dose form will be returned. If NULL, descendant codes will be returned regardless of dose form.
gapEra	Number of days between two continuous exposures to be considered in the same era. Records that have fewer days between them than this gap will be concatenated into the same cohort record.
durationRange	Deprecated.
imputeDuration	Deprecated.
priorUseWashout	Deprecated
priorObservation	Deprecated.
cohortDateRange	Deprecated.
limit	Deprecated.

**Value**

The function returns the cdm reference provided with the addition of the new cohort table.

**Examples**

```
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

cdm <- generateAtcCohortSet(
  cdm = cdm,
  atcName = "alimentary tract and metabolism",
  name = "drugs"
)

cdm$drugs |>
  glimpse()
```

---

generateDrugUtilisationCohortSet

*Generate a set of drug cohorts based on given concepts*

---

**Description**

Adds a new cohort table to the cdm reference with individuals who have drug exposure records with the specified concepts. Cohort start and end dates will be based on drug record start and end dates, respectively. Records that overlap or have fewer days between them than the specified gap era will be concatenated into a single cohort entry.

**Usage**

```
generateDrugUtilisationCohortSet(
  cdm,
  name,
  conceptSet,
  gapEra = 1,
  durationRange = lifecycle::deprecated(),
  imputeDuration = lifecycle::deprecated(),
  priorUseWashout = lifecycle::deprecated(),
  priorObservation = lifecycle::deprecated(),
  cohortDateRange = lifecycle::deprecated(),
  limit = lifecycle::deprecated()
)
```

**Arguments**

cdm	A cdm reference.
name	The name of the new cohort table to add to the cdm reference.
conceptSet	The concepts used to create the cohort, provide as a codelist or concept set expression.
gapEra	Number of days between two continuous exposures to be considered in the same era. Records that have fewer days between them than this gap will be concatenated into the same cohort record.
durationRange	Deprecated.
imputeDuration	Deprecated.
priorUseWashout	Deprecated.
priorObservation	Deprecated.
cohortDateRange	Deprecated.
limit	Deprecated.

**Value**

The function returns the cdm reference provided with the addition of the new cohort table.

**Examples**

```
library(CDMConnector)
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

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



```
)  
  
cdm <- generateDrugUtilisationCohortSet(  
  cdm = cdm,  
  name = "drug_cohorts",  
  conceptSet = druglist  
)  
  
cdm$drug_cohorts |>  
  glimpse()
```

---

generateIngredientCohortSet

*Generate a set of drug cohorts based on drug ingredients*

---

### Description

Adds a new cohort table to the cdm reference with individuals who have drug exposure records with the specified drug ingredient. Cohort start and end dates will be based on drug record start and end dates, respectively. Records that overlap or have fewer days between them than the specified gap era will be concatenated into a single cohort entry.

### Usage

```
generateIngredientCohortSet(  
  cdm,  
  name,  
  ingredient = NULL,  
  doseForm = NULL,  
  doseUnit = NULL,  
  routeCategory = NULL,  
  ingredientRange = c(1, Inf),  
  gapEra = 1,  
  durationRange = lifecycle::deprecated(),  
  imputeDuration = lifecycle::deprecated(),  
  priorUseWashout = lifecycle::deprecated(),  
  priorObservation = lifecycle::deprecated(),  
  cohortDateRange = lifecycle::deprecated(),  
  limit = lifecycle::deprecated()  
)
```

### Arguments

cdm	A cdm reference.
name	The name of the new cohort table to add to the cdm reference.

ingredient	Accepts both vectors and named lists of ingredient names. For a vector input, e.g., <code>c("acetaminophen", "codeine")</code> , it generates a cohort table with descendant concept codes for each ingredient, assigning unique <code>cohort_definition_id</code> . For a named list input, e.g., <code>list("test_1" = c("simvastatin", "acetaminophen"), "test_2" = "metformin")</code> , it produces a cohort table based on the structure of the input, where each name leads to a combined set of descendant concept codes for the specified ingredients, creating distinct <code>cohort_definition_id</code> for each named group.
doseForm	Only descendants codes with the specified dose form will be returned. If NULL, descendant codes will be returned regardless of dose form.
doseUnit	Only descendants codes with the specified dose unit will be returned. If NULL, descendant codes will be returned regardless of dose unit
routeCategory	Only descendants codes with the specified route will be returned. If NULL, descendant codes will be returned regardless of route category.
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 <code>c(2, 2)</code> would restrict to only concepts associated with two ingredients.
gapEra	Number of days between two continuous exposures to be considered in the same era. Records that have fewer days between them than this gap will be concatenated into the same cohort record.
durationRange	Deprecated.
imputeDuration	Deprecated.
priorUseWashout	Deprecated
priorObservation	Deprecated.
cohortDateRange	Deprecated.
limit	Deprecated.

### Value

The function returns the cdm reference provided with the addition of the new cohort table.

### Examples

```
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

cdm <- generateIngredientCohortSet(
  cdm = cdm,
  ingredient = "acetaminophen",
  name = "acetaminophen"
```

```
)  
  
cdm$acetaminophen |>  
  glimpse()
```

---

mockDrugUtilisation *It creates a mock database for testing DrugUtilisation package*

---

### Description

It creates a mock database for testing DrugUtilisation package

### Usage

```
mockDrugUtilisation(  
  con = NULL,  
  writeSchema = NULL,  
  numberIndividuals = 10,  
  seed = NULL,  
  ...  
)
```

### Arguments

con	A DBIConnection object to a database. If NULL a new duckdb connection will be used.
writeSchema	A schema with writing permissions to copy there the cdm tables.
numberIndividuals	Number of individuals in the mock cdm.
seed	Seed for the random numbers. If NULL no seed is used.
...	Tables to use as basis to create the mock. If some tables are provided they will be used to construct the cdm object.

### Value

A cdm reference with the mock tables

### Examples

```
library(DrugUtilisation)  
  
cdm <- mockDrugUtilisation()  
  
cdm
```

---

patternsWithFormula	<i>Patterns valid to compute daily dose with the associated formula.</i>
---------------------	--

---

**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	<i>Function to create a tibble with the patterns from current drug strength table</i>
--------------	---

---

**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 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**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

patternTable(cdm)
```

---

plotDrugRestart	<i>Generate a custom ggplot2 from a summarised_result object generated with summariseDrugRestart() function.</i>
-----------------	--

---

### Description

Generate a custom ggplot2 from a summarised\_result object generated with summariseDrugRestart() function.

### Usage

```
plotDrugRestart(  
  result,  
  facetX = "variable_name",  
  facetY = c("cdm_name", "cohort_name", "strata"),  
  colour = "variable_level",  
  splitStrata = TRUE  
)
```

### Arguments

result	A summarised_result object with results from summariseDrugRestart().
facetX	Vector of variables to facet by horizontally. Allowed options are: "cdm_name", "cohort_name", "strata", "variable_name"
facetY	Vector of variables to facet by vertically. Allowed options are: "cdm_name", "cohort_name", "strata", "variable_name".
colour	Vector of variables to distinct by colour. Allowed options are: "cdm_name", "cohort_name", "strata", "variable_name", and "variable_level".
splitStrata	Whether to split strata columns.

### Value

A ggplot2 object.

### Examples

```
## Not run:  
library(DrugUtilisation)  
  
cdm <- mockDrugUtilisation()  
  
conceptlist <- list("a" = 1125360, "b" = c(1503297, 1503327))  
cdm <- generateDrugUtilisationCohortSet(  
  cdm = cdm,  
  name = "switch_cohort",  
  conceptSet = conceptlist  
)
```

```
result <- cdm$cohort1 |>
  summariseDrugRestart(switchCohortTable = "switch_cohort")

plotDrugRestart(result)

CDMConnector::cdmDisconnect(cdm = cdm)

## End(Not run)
```

---

plotIndication	<i>Generate a plot visualisation (ggplot2) from the output of summariseIndication</i>
----------------	---

---

## Description

Generate a plot visualisation (ggplot2) from the output of summariseIndication

## Usage

```
plotIndication(
  result,
  x = "window",
  facet = c("cdm_name", "cohort_name", "strata"),
  color = c("indication"),
  splitStrata = TRUE
)
```

## Arguments

result	A summarised_result object.
x	Variables to be used in the x axis.
facet	Variables to be used to facet the plot.
color	Variables to be used to color the plot.
splitStrata	Whether to split strata.

## Value

A ggplot2 object

## Examples

```
library(DrugUtilisation)
library(CDMConnector)
library(dplyr)

cdm <- mockDrugUtilisation()
```

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

cdm <- generateIngredientCohortSet(
  cdm = cdm, name = "drug_cohort", ingredient = "acetaminophen"
)

result <- cdm$drug_cohort |>
  summariseIndication(
    indicationCohortName = "indication_cohorts",
    unknownIndicationTable = "condition_occurrence",
    indicationWindow = list(c(-Inf, 0), c(-365, 0))
  )

plotIndication(result)
```

---

plotProportionOfPatientsCovered

*Plot proportion of patients covered*

---

## Description

Plot proportion of patients covered

## Usage

```
plotProportionOfPatientsCovered(
  result,
  ylim = c(0, NA),
  facet = NULL,
  colour = NULL,
  colour_name = NULL
)
```

## Arguments

result	Output of summariseProportionOfPatientsCovered
ylim	Limits for the Y axis
facet	Variables to use for facets
colour	Variables to use for colours
colour_name	Colour legend name

## Value

Plot of proportion Of patients covered over time

---

plotTreatment	<i>Generate a custom ggplot2 from a summarised_result object generated with summariseTreatment function.</i>
---------------	--

---

### Description

Generate a custom ggplot2 from a summarised\_result object generated with summariseTreatment function.

### Usage

```
plotTreatment(  
  result,  
  facetX = "window_name",  
  facetY = c("cdm_name", "cohort_name", "strata"),  
  splitStrata = TRUE,  
  colour = "treatment"  
)
```

### Arguments

result	A summarised_result object with results from summariseDrugRestart().
facetX	Vector of variables to facet by horizontally. Allowed options are: "cdm_name", "cohort_name", "strata", "variable_name"
facetY	Vector of variables to facet by vertically. Allowed options are: "cdm_name", "cohort_name", "strata", "variable_name".
splitStrata	Whether to split strata columns.
colour	Vector of variables to distinct by colour. Allowed options are: "cdm_name", "cohort_name", "strata", "variable_name", and "variable_level".

### Value

A ggplot2 object.

### Examples

```
## Not run:  
library(DrugUtilisation)  
  
cdm <- mockDrugUtilisation()  
result <- cdm$cohort1 |>  
  summariseTreatment(  
    treatmentCohortName = "cohort2",  
    window = list(c(0, 30), c(31, 365))  
  )  
  
plotTreatment(result)
```



```
## End(Not run)
```

---

readConceptList      *Get concept ids from a provided path to json files*

---

### Description

**[Deprecated]**

### 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
)
```

---

requireDrugInDateRange

*Restrict cohort to only cohort records within a certain date range*

---

### Description

Filter the cohort table keeping only the cohort records for which the specified index date is within a specified date range.

**Usage**

```
requireDrugInDateRange(
  cohort,
  dateRange,
  indexDate = "cohort_start_date",
  cohortId = NULL,
  name = omopgenerics::tableName(cohort)
)
```

**Arguments**

cohort	A cohort table in a cdm reference.
dateRange	Date interval to consider. Any records with the index date outside of this range will be dropped.
indexDate	The column containing the date that will be checked against the date range.
cohortId	IDs of the cohorts to modify. The default is NULL meaning all cohorts will be used; otherwise, only the specified cohorts will be modified, and the rest will remain unchanged.
name	Name of the table with the filtered cohort records. The default name is the original cohort name, where the original table will be overwritten.

**Value**

The cohort table having applied the date requirement.

**Examples**

```
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

cdm$cohort1 <- cdm$cohort1 |>
  requireDrugInDateRange(
    dateRange = as.Date(c("2020-01-01", NA))
  )

attrition(cdm$cohort1) |> glimpse()
```

---

requireIsFirstDrugEntry

*Restrict cohort to only the first cohort record per subject*

---

**Description**

Filter the cohort table keeping only the first cohort record per subject.

**Usage**

```
requireIsFirstDrugEntry(
  cohort,
  cohortId = NULL,
  name = omopgenerics::tableName(cohort)
)
```

**Arguments**

cohort	A cohort table in a cdm reference.
cohortId	IDs of the cohorts to modify. The default is NULL meaning all cohorts will be used; otherwise, only the specified cohorts will be modified, and the rest will remain unchanged.
name	Name of the table with the filtered cohort records. The default name is the original cohort name, where the original table will be overwritten.

**Value**

The cohort table having applied the first entry requirement.

**Examples**

```
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

cdm$cohort1 <- cdm$cohort1 |>
  requireIsFirstDrugEntry()

attrition(cdm$cohort1) |> glimpse()
```

---

requireObservationBeforeDrug

*Restrict cohort to only cohort records with the given amount of prior observation time in the database*

---

**Description**

Filter the cohort table keeping only the cohort records for which the individual has the required observation time in the database prior to their cohort start date.

**Usage**

```
requireObservationBeforeDrug(
  cohort,
  days,
  cohortId = NULL,
  name = omopgenerics::tableName(cohort)
)
```

**Arguments**

cohort	A cohort table in a cdm reference.
days	Number of days of prior observation required before cohort start date. Any records with fewer days will be dropped.
cohortId	IDs of the cohorts to modify. The default is NULL meaning all cohorts will be used; otherwise, only the specified cohorts will be modified, and the rest will remain unchanged.
name	Name of the table with the filtered cohort records. The default name is the original cohort name, where the original table will be overwritten.

**Value**

The cohort table having applied the prior observation requirement.

**Examples**

```
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

cdm$cohort1 <- cdm$cohort1 |>
  requireObservationBeforeDrug(days = 365)

attrition(cdm$cohort1) |> glimpse()
```

---

requirePriorDrugWashout

*Restrict cohort to only cohort records with a given amount of time since the last cohort record ended*

---

**Description**

Filter the cohort table keeping only the cohort records for which the required amount of time has passed since the last cohort entry ended for that individual.

**Usage**

```
requirePriorDrugWashout(
  cohort,
  days,
  cohortId = NULL,
  name = omopgenerics::tableName(cohort)
)
```

**Arguments**

cohort	A cohort table in a cdm reference.
days	The number of days required to have passed since the last cohort record finished. Any records with fewer days than this will be dropped. Note that setting days to Inf will lead to the same result as that from using the requireIsFirstDrugEntry function (with only an individual's first cohort record kept).
cohortId	IDs of the cohorts to modify. The default is NULL meaning all cohorts will be used; otherwise, only the specified cohorts will be modified, and the rest will remain unchanged.
name	Name of the table with the filtered cohort records. The default name is the original cohort name, where the original table will be overwritten.

**Value**

The cohort table having applied the washout requirement.

**Examples**

```
library(DrugUtilisation)
library(dplyr)

cdm <- mockDrugUtilisation()

cdm$cohort1 <- cdm$cohort1 |>
  requirePriorDrugWashout(days = 90)

attrition(cdm$cohort1) |> glimpse()
```

---

stratifyByUnit

*Function to stratify a conceptSet by unit*


---

**Description**

**[Deprecated]**

**Usage**

```
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)
cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(cdm, "acetaminophen")
codelistStratified <- stratifyByUnit(codelist, cdm, 1125315)
codelistStratified
```

---

`summariseDoseCoverage` *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**

```
summariseDoseCoverage(  
  cdm,  
  ingredientConceptId,  
  estimates = c("count_missing", "percentage_missing", "mean", "sd", "q25", "median",  
               "q75"),  
  sampleSize = NULL  
)
```

**Arguments**

cdm	A cdm reference created using CDMConnector.
ingredientConceptId	Code indicating the ingredient of interest.
estimates	Estimates to obtain.
sampleSize	Maximum number of records of an ingredient to estimate dose coverage. If an ingredient has more, a random sample equal to sampleSize will be considered. If NULL, all records will be used.

**Value**

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

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

summariseDoseCoverage(cdm, 1125315)
```

---

summariseDrugRestart *Summarise the drug restart per window.*

---

**Description**

**[Experimental]**

**Usage**

```
summariseDrugRestart(  
  cohort,  
  switchCohortTable,  
  switchCohortId = NULL,  
  strata = list(),  
  followUpDays = Inf,  
  censorDate = NULL,  
  restrictToFirstDiscontinuation = TRUE  
)
```

**Arguments**

cohort	A cohort table.
switchCohortTable	A cohort table in the cdm that contains possible alternative treatments.
switchCohortId	The cohort ids to be used from switchCohortTable. If NULL all cohort definition ids are used.
strata	List with column names or vectors of column names groups to stratify results by.
followUpDays	A vector of number of days to follow up. It can be multiple values.
censorDate	Date of censoring. Individuals are always censored at the end of observation.
restrictToFirstDiscontinuation	Whether to consider only the first discontinuation episode or all of them.

**Value**

A summarised\_result object with the percentages of restart, switch and not exposed per window.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

conceptlist <- list("a" = 1125360, "b" = c(1503297, 1503327))
cdm <- generateDrugUtilisationCohortSet(
  cdm = cdm,
  name = "switch_cohort",
  conceptSet = conceptlist
)

result <- cdm$cohort1 |>
  summariseDrugRestart(switchCohortTable = "switch_cohort")

tableDrugRestart(result)

CDMConnector::cdmDisconnect(cdm = cdm)
```

---

summariseDrugUse	<i>This function is used to summarise the dose table over multiple cohorts.</i>
------------------	---

---

**Description**

**[Deprecated]**



**Usage**

```

summariseDrugUse(
  cohort,
  cdm = lifecycle::deprecated(),
  strata = list(),
  estimates = c("min", "q05", "q25", "median", "q75", "q95", "max", "mean", "sd",
    "count_missing", "percentage_missing"),
  minCellCount = lifecycle::deprecated()
)

```

**Arguments**

cohort	Cohort with drug use variables and strata.
cdm	Deprecated.
strata	Stratification list.
estimates	Estimates that we want for the columns.
minCellCount	Deprecated.

**Value**

A summary of the drug use stratified by cohort\_name and strata\_name

**Examples**

```

library(DrugUtilisation)
library(PatientProfiles)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(cdm, "acetaminophen")
cdm <- generateDrugUtilisationCohortSet(
  cdm, "dus_cohort", codelist
)
cdm[["dus_cohort"]] <- cdm[["dus_cohort"]] |>
  addDrugUse(ingredientConceptId = 1125315)
result <- summariseDrugUse(cdm[["dus_cohort"]])
print(result)

cdm[["dus_cohort"]] <- cdm[["dus_cohort"]] |>
  addSex() |>
  addAge(ageGroup = list("<40" = c(0, 39), ">=40" = c(40, 150)))

cdm[["dus_cohort"]] |>
  summariseDrugUse(strata = list("age_group", "sex", c("age_group", "sex")))

```

---

summariseDrugUtilisation

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

---

## Description

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

## Usage

```
summariseDrugUtilisation(
  cohort,
  strata = list(),
  estimates = c("q25", "median", "q75", "mean", "sd", "count_missing",
    "percentage_missing"),
  ingredientConceptId = NULL,
  conceptSet = NULL,
  indexDate = "cohort_start_date",
  censorDate = "cohort_end_date",
  restrictIncident = TRUE,
  gapEra = 1,
  numberExposures = TRUE,
  numberEras = TRUE,
  exposedTime = TRUE,
  timeToExposure = TRUE,
  initialQuantity = TRUE,
  cumulativeQuantity = TRUE,
  initialDailyDose = TRUE,
  cumulativeDose = TRUE
)
```

## Arguments

cohort	Cohort with drug use variables and strata.
strata	Stratification list.
estimates	Estimates that we want for the columns.
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.
indexDate	Name of a column that indicates the date to start the analysis.
censorDate	Name of a column that indicates the date to stop the analysis, if NULL end of individuals observation is used.

restrictIncident	Whether to include only incident prescriptions in the analysis. If FALSE all prescriptions that overlap with the study period will be included.
gapEra	Number of days between two continuous exposures to be considered in the same era.
numberExposures	Whether to add a column with the number of exposures.
numberEras	Whether to add a column with the number of eras.
exposedTime	Whether to add a column with the number of exposed days.
timeToExposure	Whether to add a column with the number of days between indexDate and start of the first exposure.
initialQuantity	Whether to add a column with the initial quantity.
cumulativeQuantity	Whether to add a column with the cumulative quantity of the identified prescription.
initialDailyDose	Whether to add a column with the initial daily dose.
cumulativeDose	Whether to add a column with the cumulative dose.

**Value**

A summary of drug utilisation stratified by cohort\_name and strata\_name

**Examples**

```
library(DrugUtilisation)
library(CodelistGenerator)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(cdm, "acetaminophen")
cdm <- generateDrugUtilisationCohortSet(
  cdm, "dus_cohort", codelist
)
cdm[["dus_cohort"]] %>%
  summariseDrugUtilisation(ingredientConceptId = 1125315)
```

---

summariseIndication     *Summarise the indications of individuals in a drug cohort*

---

**Description**

Summarise the observed indications of patients in a drug cohort based on their presence in an indication cohort in a specified time window. If an individual is not in one of the indication cohorts, they will be considered to have an unknown indication if they are present in one of the specified OMOP CDM clinical tables. Otherwise, if they are neither in an indication cohort or a clinical table they will be considered as having no observed indication.

**Usage**

```
summariseIndication(
  cohort,
  strata = list(),
  indicationCohortName,
  indicationCohortId = NULL,
  indicationWindow = list(c(0, 0)),
  unknownIndicationTable = NULL,
  indexDate = "cohort_start_date",
  censorDate = NULL
)
```

**Arguments**

cohort	A cohort table in a cdm reference.
strata	List of variables to stratify results by. These variables must be present in the cohort table.
indicationCohortName	Name of the cohort table with potential indications.
indicationCohortId	The target cohort ID to add indication. If NULL all cohorts will be considered.
indicationWindow	The time window over which to identify indications.
unknownIndicationTable	Tables in the OMOP CDM to search for unknown indications.
indexDate	A date variable in the cohort table for which indications will be found relative to.
censorDate	After that day no indication will be considered.

**Value**

A summarised result

**Examples**

```
library(DrugUtilisation)
library(CDMConnector)
library(dplyr)

cdm <- mockDrugUtilisation()

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

cdm <- generateIngredientCohortSet(
  cdm = cdm, name = "drug_cohort",
  ingredient = "acetaminophen"
)
```

```
cdm$drug_cohort |>
  summariseIndication(
    indicationCohortName = "indication_cohorts",
    unknownIndicationTable = "condition_occurrence",
    indicationWindow = list(c(-Inf, 0))
  ) |>
  glimpse()
```

---

summariseProportionOfPatientsCovered

*Summarise proportion of patients covered*

---

### Description

Gives the proportion of patients still in observation who are in the cohort on any given day following their first cohort entry. This is known as the “proportion of patients covered” (PPC) method for assessing treatment persistence.

### Usage

```
summariseProportionOfPatientsCovered(
  cohort,
  cohortId = NULL,
  strata = list(),
  followUpDays = NULL
)
```

### Arguments

cohort	A cohort table
cohortId	Cohort definition ID of interest. If NULL, results for all cohorts will be returned.
strata	List of variables to stratify by.
followUpDays	Number of days to follow up individuals for. If NULL the maximum amount of days from an individuals first cohort start date to their last cohort end date will be used

### Value

A summarised result

---

summariseTreatment      *This function is used to summarise treatments received*

---

### Description

This function is used to summarise treatments received

### Usage

```
summariseTreatment(
  cohort,
  window,
  treatmentCohortName,
  treatmentCohortId = NULL,
  strata = list(),
  indexDate = "cohort_start_date",
  censorDate = NULL,
  minCellCount = lifecycle::deprecated()
)
```

### Arguments

cohort	A cohort table in a cdm reference.
window	Time window over which to summarise the treatments.
treatmentCohortName	Name of a cohort in the cdm that contains the treatments of interest.
treatmentCohortId	Cohort definition id of interest from treatmentCohortName.
strata	List with column names or vectors of column names groups to stratify results by.
indexDate	Variable in x that contains the date to compute the intersection.
censorDate	Whether to censor overlap events at a specific date or a column date of x. If NULL, end of observation will be used.
minCellCount	<code>\ifelse{html}{\href{https://lifecycle.r-lib.org/articles/stages.html#deprecated}}{\fi}</code>

### Value

A summary of treatments stratified by cohort\_name and strata\_name

### Examples

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()
cdm$cohort1 |>
  summariseTreatment(
```

```

    treatmentCohortName = "cohort2",
    window = list(c(0, 30), c(31, 365))
  )

```

---

tableDoseCoverage	<i>Format a dose_coverage object into a visual table.</i>
-------------------	---

---

## Description

[Experimental]

## Usage

```

tableDoseCoverage(
  result,
  header = c("variable", "estimate"),
  splitStrata = TRUE,
  ingradientName = TRUE,
  cdmName = TRUE,
  groupColumn = NULL,
  type = "gt",
  formatEstimateName = c(`N (%)` = "<count_missing> (<percentage_missing> %)", N =
    "<count>", `Mean (SD)` = "<mean> (<sd>)", `Median (Q25 - Q75)` =
    "<median> (<q25> - <q75>)" ),
  .options = list()
)

```

## Arguments

result	A summarised_result object with results from summariseDoseCoverage().
header	A vector containing which elements should go into the header in order. Allowed are: cdm_name, group, strata, variable, and estimate.
splitStrata	If TRUE strata columns will be split.
ingradientName	If TRUE cohort names will be displayed.
cdmName	If TRUE database names will be displayed.
groupColumn	Column to use as group labels.
type	Type of desired formatted table, possibilities: "gt", "flextable", "tibble".
formatEstimateName	Named list of estimate name's to join, sorted by computation order. Indicate estimate_name's between <...>.
.options	Named list with additional formatting options. DrugUtilisation::defaultTableOptions() shows allowed arguments and their default values.

**Value**

A table with a formatted version of summariseDrugCoverage() results.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()

result <- summariseDoseCoverage(cdm, 1125315)

tableDoseCoverage(result)
```

---

tableDrugRestart	<i>Format a drug_restart object into a visual table.</i>
------------------	--

---

**Description**

**[Experimental]**

**Usage**

```
tableDrugRestart(
  result,
  header = c("strata"),
  splitStrata = TRUE,
  cohortName = TRUE,
  cdmName = TRUE,
  groupColumn = c("cdm_name", "cohort_name"),
  type = "gt",
  formatEstimateName = c(`N (%)` = "<count> (<percentage> %)",
  .options = list()
)
```

**Arguments**

result	A summarised_result object with results from summariseDrugRestart().
header	A vector containing which elements should go into the header in order. Allowed values: cdm_name, cohort_name, strata, variable, estimate.
splitStrata	If TRUE strata columns will be split.
cohortName	If TRUE cohort names will be displayed.
cdmName	If TRUE database names will be displayed.
groupColumn	Column to use as group labels. Allowed values: cdm_name, cohort_name, strata, variable_name, variable_level, estimate_name.



**type**                   Type of desired formatted table, possibilities: "gt", "flextable", "tibble".  
**formatEstimateName**       Named list of estimate name's to join, sorted by computation order. Indicate estimate\_name's between <...>.  
**.options**                Named list with additional formatting options. DrugUtilisation::defaultTableOptions() shows allowed arguments and their default values.

## Value

A table with a formatted version of summariseDrugRestart() results.

## Examples

```

library(DrugUtilisation)

cdm <- mockDrugUtilisation()

conceptlist <- list("a" = 1125360, "b" = c(1503297, 1503327))
cdm <- generateDrugUtilisationCohortSet(
  cdm = cdm,
  name = "switch_cohort",
  conceptSet = conceptlist
)

result <- cdm$cohort1 |>
  summariseDrugRestart(switchCohortTable = "switch_cohort")

tableDrugRestart(result)

CDMConnector::cdmDisconnect(cdm = cdm)
  
```

---

tableDrugUtilisation   *Format a drug\_utilisation object into a visual table.*

---

## Description

**[Experimental]**

## Usage

```

tableDrugUtilisation(
  result,
  header = c("group", "strata"),
  splitStrata = TRUE,
  cohortName = TRUE,
  cdmName = TRUE,
)
  
```

```

conceptSet = TRUE,
ingredient = TRUE,
groupColumn = NULL,
type = "gt",
formatEstimateName = c(`N (%)` = "<count_missing> (<percentage_missing> %)", N =
  "<count>", `Mean (SD)` = "<mean> (<sd>)", `Median (Q25 - Q75)` =
  "<median> (<q25> - <q75>)",
.options = list()
)

```

### Arguments

result	A summarised_result object with results from summariseDrugUtilisation().
header	A vector containing which elements should go into the header in order. Allowed are: cdm_name, group, strata, variable, estimate.
splitStrata	If TRUE strata columns will be split.
cohortName	If TRUE cohort names will be displayed.
cdmName	If TRUE database names will be displayed.
conceptSet	If TRUE concept sets name will be displayed.
ingredient	If TRUE ingredients names will be displayed for dose calculation.
groupColumn	Column to use as group labels, these can be: "cdm_name", "cohort_name", "concept_set", "variable_name", and/or "ingredient". If strata is split, any of the levels can be used, otherwise "strata_name" and "strata_level" can be used for table group format.
type	Type of desired formatted table, possibilities: "gt", "flextable", "tibble".
formatEstimateName	Named list of estimate name's to join, sorted by computation order. Indicate estimate_name's between <...>.
.options	Named list with additional formatting options. DrugUtilisation::defaultTableOptions() shows allowed arguments and their default values.

### Value

A table with a formatted version of summariseIndication() results.

### Examples

```

library(DrugUtilisation)
library(CodelistGenerator)

cdm <- mockDrugUtilisation()
codelist <- CodelistGenerator::getDrugIngredientCodes(cdm, "acetaminophen")
cdm <- generateDrugUtilisationCohortSet(cdm, "dus_cohort", codelist)
cdm[["dus_cohort"]] %>%
  summariseDrugUtilisation(ingredientConceptId = 1125315) |>
  tableDrugUtilisation()

```

---

tableIndication	<i>Create a table showing indication results</i>
-----------------	--

---

**Description****[Experimental]****Usage**

```
tableIndication(
  result,
  header = c("group", "strata"),
  splitStrata = TRUE,
  cohortName = TRUE,
  cdmName = TRUE,
  groupColumn = "variable_name",
  type = "gt",
  .options = list()
)
```

**Arguments**

result	A summarised_result created by summariseIndication().
header	A vector containing which elements should go into the header in order. Allowed are: cdm_name, group, strata, variable.
splitStrata	If TRUE strata columns will be split.
cohortName	If TRUE cohort names will be displayed.
cdmName	If TRUE database names will be displayed.
groupColumn	Column to use as group labels.
type	Type of desired formatted table, possibilities: "gt", "flextable", "tibble".
.options	Named list with additional formatting options. DrugUtilisation::defaultTableOptions() shows allowed arguments and their default values.

**Value**

A table with a formatted version of summariseIndication() results.

**Examples**

```

library(DrugUtilisation)

cdm <- mockDrugUtilisation()

result <- cdm$cohort1 |>
  summariseIndication(
    indicationCohortName = "cohort2",
    indicationWindow = list(c(-30, 0)),
    unknownIndicationTable = "condition_occurrence"
  )

tableIndication(result, type = "tibble")

```

---

tableProportionOfPatientsCovered

*Create a table with proportion of patients covered results*

---

**Description****[Experimental]****Usage**

```

tableProportionOfPatientsCovered(
  result,
  times = NULL,
  header = c("group", "strata"),
  splitStrata = TRUE,
  cohortName = TRUE,
  cdmName = TRUE,
  groupColumn = "variable_name",
  type = "gt",
  .options = list()
)

```

**Arguments**

result	A summarised_result object with results from summariseProportionOfPatientsCovered().
times	Days to include in the table. If NULL all days will be included.
header	A vector containing which elements should go into the header in order. Allowed are: cdm_name, group, strata, variable.
splitStrata	If TRUE strata columns will be split.

cohortName	If TRUE cohort names will be displayed.
cdmName	If TRUE database names will be displayed.
groupColumn	Column to use as group labels.
type	Type of desired formatted table, possibilities: "gt", "flextable", "tibble".
.options	Named list with additional formatting options. DrugUtilisation::defaultTableOptions() shows allowed arguments and their default values.

### Value

A table with a formatted version of summariseProportionOfPatientsCovered() results.

---

tableTreatment	<i>Format a summarised_treatment result into a visual table.</i>
----------------	--

---

### Description

**[Experimental]**

### Usage

```
tableTreatment(
  result,
  header = c("window_name"),
  splitStrata = TRUE,
  cdmName = TRUE,
  groupColumn = c("cdm_name", "cohort_name"),
  type = "gt",
  formatEstimateName = c(`N (%)` = "<count> (<percentage> %)",
  .options = list()
)
```

### Arguments

result	A summarised_result object with results from summariseTreatmentFromCohort() or summariseTreatmentFromConceptSet().
header	A vector containing which elements should go into the header in order. Allowed values: cdm_name, cohort_name, strata, variable, estimate and window_name.
splitStrata	If TRUE strata columns will be split.
cdmName	If TRUE database names will be displayed.
groupColumn	Column to use as group labels. Allowed values: cdm_name, cohort_name, strata, variable, estimate and window_name.
type	Type of desired formatted table, possibilities: "gt", "flextable", "tibble".
formatEstimateName	Named list of estimate name's to join, sorted by computation order. Indicate estimate_name's between <...>.
.options	Named list with additional formatting options. DrugUtilisation::defaultTableOptions() shows allowed arguments and their default values.

**Value**

A table with a formatted version of summariseTreatment() results.

**Examples**

```
library(DrugUtilisation)

cdm <- mockDrugUtilisation()
result <- cdm$cohort1 |>
  summariseTreatment(
    treatmentCohortName = "cohort2",
    window = list(c(0, 30), c(31, 365))
  )

tableTreatment(result)
```

# Index

- \* **datasets**
  - patternsWithFormula, 28
- addCumulativeDose, 3
- addCumulativeQuantity, 4
- addDailyDose, 5
- addDrugUse, 6
- addDrugUtilisation, 8
- addExposedTime, 10
- addIndication, 11
- addInitialDailyDose, 12
- addInitialQuantity, 14
- addNumberEras, 15
- addNumberExposures, 16
- addRoute, 17
- addTimeToExposure, 18

benchmarkDrugUtilisation, 19

cohortGapEra, 20

dailyDoseCoverage, 21

defaultTableOptions, 21

generateAtcCohortSet, 22

generateDrugUtilisationCohortSet, 23

generateIngredientCohortSet, 25

mockDrugUtilisation, 27

patternsWithFormula, 28

patternTable, 28

plotDrugRestart, 29

plotIndication, 30

plotProportionOfPatientsCovered, 31

plotTreatment, 32

readConceptList, 33

requireDrugInDateRange, 33

requireIsFirstDrugEntry, 34

requireObservationBeforeDrug, 35

requirePriorDrugWashout, 36

stratifyByUnit, 37

summariseDoseCoverage, 38

summariseDrugRestart, 39

summariseDrugUse, 40

summariseDrugUtilisation, 42

summariseIndication, 43

summariseProportionOfPatientsCovered, 45

summariseTreatment, 46

tableDoseCoverage, 47

tableDrugRestart, 48

tableDrugUtilisation, 49

tableIndication, 51

tableProportionOfPatientsCovered, 52

tableTreatment, 53