Package ‘OmopSketch’

June 19, 2025

Title Characterise Tables of an OMOP Common Data Model Instance

Version 0.5.1

Maintainer Cecilia Campanile <`cecilia.campanile@ndorms.ox.ac.uk`>

Description Summarises key information in data mapped to the Observational Medical Outcomes Partnership (OMOP) common data model. Assess suitability to perform specific epidemiological studies and explore the different domains to obtain feasibility counts and trends.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Suggests CodelistGenerator, CohortCharacteristics, DBI, duckdb, DT, flextable, gt, here, knitr, lubridate, odbc, OmopViewer (>= 0.3.0), sortable, reactable, remotes, rmarkdown, RPostgres, shinyWidgets, testthat (>= 3.0.0), withr, omock (>= 0.3.0), covr, ggplot2, visOmopResults (>= 0.5.0), devtools

Config/testthat/edition 3

Config/testthat/parallel true

Imports CDMConnector (>= 1.3.0), cli, clock, CohortConstructor (>= 0.3.1), dplyr, glue, lifecycle, omopgenerics (>= 0.4.1), PatientProfiles (>= 1.3.1), purrr, rlang, stringr, tibble, tidyr

Depends R (>= 4.1)

URL `https://OHDSI.github.io/OmopSketch/`

BugReports `https://github.com/OHDSI/OmopSketch/issues`

VignetteBuilder knitr

NeedsCompilation no

Author Marta Alcalde-Herraiz [aut] (ORCID: <`https://orcid.org/0009-0002-4405-1814`>),
Kim Lopez-Guell [aut] (ORCID: <`https://orcid.org/0000-0002-8462-8668`>),
Elin Rowlands [aut] (ORCID: <`https://orcid.org/0009-0005-5166-0417`>),

Cecilia Campanile [aut, cre] (ORCID:
<https://orcid.org/0009-0007-6629-4661>),
Edward Burn [aut] (ORCID: <https://orcid.org/0000-0002-9286-1128>),
Martí Català [aut] (ORCID: <https://orcid.org/0000-0003-3308-9905>)

Repository CRAN

Date/Publication 2025-06-19 19:50:06 UTC

Contents

`databaseCharacteristics` *Summarise Database Characteristics for OMOP CDM*

Description

Summarise Database Characteristics for OMOP CDM

Usage

```
databaseCharacteristics(
  cdm,
  omopTableName = c("person", "observation_period", "visit_occurrence",
   "condition_occurrence", "drug_exposure", "procedure_occurrence", "device_exposure",
    "measurement", "observation", "death"),
  sex = FALSE,
  ageGroup = NULL,
  dateRange = NULL,
  interval = "overall",
  conceptIdCounts = FALSE,
  ...
)
```

Arguments

`cdm`	A `cdm_reference` object representing the Common Data Model (CDM) reference.
`omopTableName`	A character vector specifying the OMOP tables from the CDM to include in the analysis. If "person" is present, it will only be used for missing value summarisation.
`sex`	Logical; whether to stratify results by sex (`TRUE`) or not (`FALSE`).
`ageGroup`	A list of age groups to stratify the results by. Each element represents a specific age range.
`dateRange`	A vector of two dates defining the desired study period. Only the `start_date` column of the OMOP table is checked to ensure it falls within this range. If `dateRange` is `NULL`, no restriction is applied.
`interval`	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
`conceptIdCounts`	Logical; whether to summarise concept ID counts (`TRUE`) or not (`FALSE`).
`...`	additional arguments passed to the OmopSketch functions that are used internally.

Value

A `summarised_result` object containing the results of the characterisation.

Examples

```
cdm <- mockOmopSketch(numberIndividuals = 100)

result <- databaseCharacteristics(cdm = cdm,
omopTableNam = c("drug_exposure", "condition_occurrence"),
sex = TRUE, ageGroup = list(c(0,50), c(51,100)), interval = "years", conceptIdCounts = FALSE)
```

```
PatientProfiles::mockDisconnect(cdm)
```

`mockOmopSketch` *Creates a mock database to test OmopSketch package.*

Description

Creates a mock database to test OmopSketch package.

Usage

```
mockOmopSketch(
  con = NULL,
  writeSchema = NULL,
  numberIndividuals = 100,
  seed = NULL
)
```

Arguments

`con`	A DBI connection to create the cdm mock object. By default, the connection would be a 'duckdb' one.
`writeSchema`	Name of an schema of the DBI connection with writing permissions.
`numberIndividuals`	Number of individuals to create in the cdm reference object.
`seed`	An optional integer used to set the seed for random number generation, ensuring reproducibility of the generated data. If provided, this seed allows the function to produce consistent results each time it is run with the same inputs. If 'NULL', the seed is not set, which can lead to different outputs on each run.

Value

A mock cdm_reference object.

Examples

```
mockOmopSketch(numberIndividuals = 100)
```

`plotConceptSetCounts` *Plot the concept counts of a summariseConceptSetCounts output.*

Description

Plot the concept counts of a summariseConceptSetCounts output.

Usage

```
plotConceptSetCounts(result, facet = NULL, colour = NULL)
```

Arguments

`result`	A summarised_result object (output of summariseConceptSetCounts).
`facet`	Columns to face by. Formula format can be provided. See possible columns to face by with: `visOmopResults::tidyColumns()`.
`colour`	Columns to colour by. See possible columns to colour by with: `visOmopResults::tidyColumns()`.

Value

A ggplot2 object showing the concept counts.

Examples

```
library(dplyr)

cdm <- mockOmopSketch()

result <- summariseConceptSetCounts(
  cdm = cdm,
  conceptSet = list(
    "asthma" = c(4051466, 317009) ,
    "rhinitis" = c(4280726, 4048171, 40486433)
  )
)

result |>
  filter(variable_name == "Number subjects") |>
  plotConceptSetCounts(facet = "codelist_name", colour = "standard_concept_name")

PatientProfiles::mockDisconnect(cdm)
```

`plotInObservation` *Create a ggplot2 plot from the output of summariseInObservation().*

Description

Create a ggplot2 plot from the output of summariseInObservation().

Usage

```
plotInObservation(result, facet = NULL, colour = NULL)
```

Arguments

`result`	A summarised_result object (output of summariseInObservation).
`facet`	Columns to face by. Formula format can be provided. See possible columns to face by with: `visOmopResults::tidyColumns()`.
`colour`	Columns to colour by. See possible columns to colour by with: `visOmopResults::tidyColumns()`.

Value

A ggplot showing the table counts

Examples

```
library(dplyr)

cdm <- mockOmopSketch()

result <- summariseInObservation(
  observationPeriod = cdm$observation_period,
  output = c("person-days","record"),
  ageGroup = list("<=40" = c(0, 40), ">40" = c(41, Inf)),
  sex = TRUE
)

result |>
  filter(variable_name == "Number person-days") |>
  plotInObservation(facet = "sex", colour = "age_group")

PatientProfiles::mockDisconnect(cdm)
```

`plotObservationPeriod` *Create a plot from the output of summariseObservationPeriod().*

Description

Create a plot from the output of summariseObservationPeriod().

Usage

```
plotObservationPeriod(
  result,
  variableName = "Number subjects",
  plotType = "barplot",
  facet = NULL,
  colour = NULL
)
```

Arguments

`result`	A summarised_result object.
`variableName`	The variable to plot it can be: "number subjects", "records per person", "duration" or "days to next observation period".
`plotType`	The plot type, it can be: "barplot", "boxplot" or "densityplot".
`facet`	Columns to colour by. See possible columns to colour by with: `visOmopResults::tidyColumns()`.
`colour`	Columns to colour by. See possible columns to colour by with: `visOmopResults::tidyColumns()`.

Value

A ggplot2 object.

Examples

```
cdm <- mockOmopSketch(numberIndividuals = 100)

result <- summariseObservationPeriod(observationPeriod = cdm$observation_period)

plotObservationPeriod(result = result,
    variableName = "Duration in days",
    plotType = "boxplot"
  )

PatientProfiles::mockDisconnect(cdm)
```

`plotRecordCount` *Create a ggplot of the records' count trend.*

Description

Create a ggplot of the records' count trend.

Usage

```
plotRecordCount(result, facet = NULL, colour = NULL)
```

Arguments

`result`	Output from summariseRecordCount().
`facet`	Columns to face by. Formula format can be provided. See possible columns to face by with: `visOmopResults::tidyColumns()`.
`colour`	Columns to colour by. See possible columns to colour by with: `visOmopResults::tidyColumns()`.

Value

A ggplot showing the table counts

Examples

```
cdm <- mockOmopSketch()

summarisedResult <- summariseRecordCount(
  cdm = cdm,
  omopTableName = "condition_occurrence",
  ageGroup = list("<=20" = c(0, 20), ">20" = c(21, Inf)),
  sex = TRUE
)

plotRecordCount(result = summarisedResult, colour = "age_group", facet = sex ~ .)

PatientProfiles::mockDisconnect(cdm = cdm)
```

`shinyCharacteristics` *Generate an interactive Shiny application that visualises the results obtained from the* `databaseCharacteristics()` *function.*

Description

Generate an interactive Shiny application that visualises the results obtained from the `databaseCharacteristics()` function.

Usage

```
shinyCharacteristics(
  result,
  directory,
  title = "Database characterisation",
  logo = "ohdsi",
  theme = "bslib::bs_theme(bootswatch = 'flatly')"
)
```

Arguments

`result`	A summarised_result object containing the results from the `databaseCharacteristics()` function. This object should include summaries of various OMOP CDM tables, such as population characteristics, clinical records, missing data, and more
`directory`	A character string specifying the directory where the application will be saved.
`title`	Title of the shiny. Default is "Characterisation"
`logo`	Name of a logo or path to a logo. If NULL no logo is included. Only svg format allowed for the moment.
`theme`	A character string specifying the theme for the Shiny application. Default is `"bslib::bs_theme(bootswatch = 'flatly')"` to use the Flatly theme from the Bootswatch collection. You can customise this to use other themes.

Value

This function invisibly returns NULL and generates a static Shiny app in the specified directory.

Examples

```
## Not run:

library(OmopSketch)
cdm <- mockOmopSketch()
res <- databaseCharacteristics(cdm = cdm)
shinyCharacteristics(result = res, directory = here::here())

## End(Not run)
```

`summariseClinicalRecords`

Summarise an omop table from a cdm object. You will obtain information related to the number of records, number of subjects, whether the records are in observation, number of present domains and number of present concepts.

Description

Summarise an omop table from a cdm object. You will obtain information related to the number of records, number of subjects, whether the records are in observation, number of present domains and number of present concepts.

Usage

```
summariseClinicalRecords(
  cdm,
  omopTableName,
  recordsPerPerson = c("mean", "sd", "median", "q25", "q75", "min", "max"),
  inObservation = TRUE,
  standardConcept = TRUE,
  sourceVocabulary = TRUE,
  domainId = TRUE,
  typeConcept = TRUE,
  sex = FALSE,
  ageGroup = NULL,
  dateRange = NULL
)
```

Arguments

`cdm`	A cdm_reference object.
`omopTableName`	A character vector of the names of the tables to summarise in the cdm object.
`recordsPerPerson`	Generates summary statistics for the number of records per person. Set to NULL if no summary statistics are required.
`inObservation`	Boolean variable. Whether to include the percentage of records in observation.
`standardConcept`	Boolean variable. Whether to summarise standard concept information.
`sourceVocabulary`	Boolean variable. Whether to summarise source vocabulary information.
`domainId`	Boolean variable. Whether to summarise domain id of standard concept id information.
`typeConcept`	Boolean variable. Whether to summarise type concept id field information.
`sex`	Boolean variable. Whether to stratify by sex (TRUE) or not (FALSE).
`ageGroup`	A list of age groups to stratify results by.
`dateRange`	A vector of two dates defining the desired study period. Only the `start_date` column of the OMOP table is checked to ensure it falls within this range. If `dateRange` is `NULL`, no restriction is applied.

Value

A summarised_result object.

Examples

```
cdm <- mockOmopSketch()

summarisedResult <- summariseClinicalRecords(
  cdm = cdm,
  omopTableName = "condition_occurrence",
  recordsPerPerson = c("mean", "sd"),
  inObservation = TRUE,
  standardConcept = TRUE,
  sourceVocabulary = TRUE,
  domainId = TRUE,
  typeConcept = TRUE
)

summarisedResult

PatientProfiles::mockDisconnect(cdm = cdm)
```

`summariseConceptCounts` *Summarise concept counts in patient-level data. Only concepts recorded during observation period are counted.*

Description

[Deprecated]

Usage

```
summariseConceptCounts(
  cdm,
  conceptId,
  countBy = c("record", "person"),
  concept = TRUE,
  interval = "overall",
  sex = FALSE,
  ageGroup = NULL,
  dateRange = NULL
)
```

Arguments

`cdm`	A cdm object
`conceptId`	List of concept IDs to summarise.
`countBy`	Either "record" for record-level counts or "person" for person-level counts
`concept`	TRUE or FALSE. If TRUE code use will be summarised by concept.

`interval`	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
`sex`	TRUE or FALSE. If TRUE code use will be summarised by sex.
`ageGroup`	A list of ageGroup vectors of length two. Code use will be thus summarised by age groups.
`dateRange`	A vector of two dates defining the desired study period. Only the `start_date` column of the OMOP table is checked to ensure it falls within this range. If `dateRange` is `NULL`, no restriction is applied.

Value

A summarised_result object with results overall and, if specified, by strata.

`summariseConceptIdCounts` *Summarise concept use in patient-level data. Only concepts recorded during observation period are counted.*

Description

Summarise concept use in patient-level data. Only concepts recorded during observation period are counted.

Usage

```
summariseConceptIdCounts(
  cdm,
  omopTableName,
  countBy = "record",
  year = lifecycle::deprecated(),
  interval = "overall",
  sex = FALSE,
  ageGroup = NULL,
  sample = NULL,
  dateRange = NULL
)
```

Arguments

`cdm`	A cdm object
`omopTableName`	A character vector of the names of the tables to summarise in the cdm object.
`countBy`	Either "record" for record-level counts or "person" for person-level counts
`year`	deprecated
`interval`	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".

`sex`	TRUE or FALSE. If TRUE code use will be summarised by sex.
`ageGroup`	A list of ageGroup vectors of length two. Code use will be thus summarised by age groups.
`sample`	An integer to sample the tables to only that number of records. If NULL no sample is done.
`dateRange`	A vector of two dates defining the desired study period. Only the `start_date` column of the OMOP table is checked to ensure it falls within this range. If `dateRange` is `NULL`, no restriction is applied.

Value

A summarised_result object with results overall and, if specified, by strata.

Examples

```
library(OmopSketch)
library(CDMConnector)
library(duckdb)

requireEunomia()
con <- dbConnect(duckdb(), eunomiaDir())
cdm <- cdmFromCon(con = con, cdmSchema = "main", writeSchema = "main")

summariseConceptIdCounts(cdm = cdm, omopTableName = "condition_occurrence",
countBy = c("record", "person"), sex = TRUE)
```

`summariseConceptSetCounts`

Summarise concept counts in patient-level data. Only concepts recorded during observation period are counted.

Description

Summarise concept counts in patient-level data. Only concepts recorded during observation period are counted.

Usage

```
summariseConceptSetCounts(
  cdm,
  conceptSet,
  countBy = c("record", "person"),
  concept = TRUE,
  interval = "overall",
  sex = FALSE,
```

```
  ageGroup = NULL,
  dateRange = NULL
)
```

Arguments

`cdm`	A cdm object
`conceptSet`	List of concept IDs to summarise.
`countBy`	Either "record" for record-level counts or "person" for person-level counts
`concept`	TRUE or FALSE. If TRUE code use will be summarised by concept.
`interval`	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
`sex`	TRUE or FALSE. If TRUE code use will be summarised by sex.
`ageGroup`	A list of ageGroup vectors of length two. Code use will be thus summarised by age groups.
`dateRange`	A vector of two dates defining the desired study period. Only the `start_date` column of the OMOP table is checked to ensure it falls within this range. If `dateRange` is `NULL`, no restriction is applied.

Value

A summarised_result object with results overall and, if specified, by strata.

Examples

```
library(OmopSketch)

cdm <- mockOmopSketch()

cs <- list(sinusitis = c(4283893, 257012, 40481087, 4294548))

results <- summariseConceptSetCounts(cdm, conceptSet = cs)

results

PatientProfiles::mockDisconnect(cdm)
```

`summariseInObservation`

Summarise the number of people in observation during a specific interval of time.

Description

Summarise the number of people in observation during a specific interval of time.

Usage

```
summariseInObservation(
  observationPeriod,
  interval = "overall",
  output = "record",
  ageGroup = NULL,
  sex = FALSE,
  dateRange = NULL
)
```

Arguments

`observationPeriod`	An observation_period omop table. It must be part of a cdm_reference object.
`interval`	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
`output`	Output format. It can be either the number of records ("record") that are in observation in the specific interval of time, the number of person-days ("person-days"), the number of subjects ("person"), the number of females ("sex") or the median age of population in observation ("age").
`ageGroup`	A list of age groups to stratify results by.
`sex`	Boolean variable. Whether to stratify by sex (TRUE) or not (FALSE). For output = "sex" this stratification is not applied.
`dateRange`	A vector of two dates defining the desired study period. Only the `start_date` column of the OMOP table is checked to ensure it falls within this range. If `dateRange` is `NULL`, no restriction is applied.

Value

A summarised_result object.

Examples

```
library(dplyr, warn.conflicts = FALSE)

cdm <- mockOmopSketch()

result <- summariseInObservation(
  observationPeriod = cdm$observation_period,
  interval = "months",
  output = c("person-days", "record"),
  ageGroup = list("<=60" = c(0, 60), ">60" = c(61, Inf)),
  sex = TRUE
)

result |>
  glimpse()
```

```
PatientProfiles::mockDisconnect(cdm)
```

`summariseMissingData` *Summarise missing data in omop tables*

Description

Summarise missing data in omop tables

Usage

```
summariseMissingData(
  cdm,
  omopTableName,
  col = NULL,
  sex = FALSE,
  year = lifecycle::deprecated(),
  interval = "overall",
  ageGroup = NULL,
  sample = 1e+06,
  dateRange = NULL
)
```

Arguments

`cdm`	A cdm object
`omopTableName`	A character vector of the names of the tables to summarise in the cdm object.
`col`	A character vector of column names to check for missing values. If `NULL`, all columns in the specified tables are checked. Default is `NULL`.
`sex`	TRUE or FALSE. If TRUE code use will be summarised by sex.
`year`	deprecated
`interval`	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
`ageGroup`	A list of ageGroup vectors of length two. Code use will be thus summarised by age groups.
`sample`	An integer to sample the table to only that number of records. If NULL no sample is done.
`dateRange`	A vector of two dates defining the desired study period. Only the `start_date` column of the OMOP table is checked to ensure it falls within this range. If `dateRange` is `NULL`, no restriction is applied.

Value

A summarised_result object with results overall and, if specified, by strata.

Examples

```
cdm <- mockOmopSketch(numberIndividuals = 100)

result <- summariseMissingData (cdm = cdm,
omopTableName = c("condition_occurrence", "visit_occurrence"),
sample = 10000)

PatientProfiles::mockDisconnect(cdm)
```

`summariseObservationPeriod` *Summarise the observation period table getting some overall statistics in a summarised_result object.*

Description

Summarise the observation period table getting some overall statistics in a summarised_result object.

Usage

```
summariseObservationPeriod(
  observationPeriod,
  estimates = c("mean", "sd", "min", "q05", "q25", "median", "q75", "q95", "max",
    "density"),
  byOrdinal = TRUE,
  ageGroup = NULL,
  sex = FALSE,
  dateRange = NULL
)
```

Arguments

`observationPeriod`	observation_period omop table.
`estimates`	Estimates to summarise the variables of interest (`records per person`, `duration in days` and `days to next observation period`).
`byOrdinal`	Boolean variable. Whether to stratify by the ordinal observation period (e.g., 1st, 2nd, etc.) (TRUE) or simply analyze overall data (FALSE)
`ageGroup`	A list of age groups to stratify results by.
`sex`	Boolean variable. Whether to stratify by sex (TRUE) or not (FALSE).
`dateRange`	A vector of two dates defining the desired study period. Only the `start_date` column of the OMOP table is checked to ensure it falls within this range. If `dateRange` is `NULL`, no restriction is applied.

Value

A summarised_result object with the summarised data.

Examples

```
library(dplyr, warn.conflicts = FALSE)

cdm <- mockOmopSketch(numberIndividuals = 100)

result <- summariseObservationPeriod(observationPeriod = cdm$observation_period)

result |>
  glimpse()

PatientProfiles::mockDisconnect(cdm)
```

`summariseOmopSnapshot` *Summarise a cdm_reference object creating a snapshot with the metadata of the cdm_reference object.*

Description

Summarise a cdm_reference object creating a snapshot with the metadata of the cdm_reference object.

Usage

```
summariseOmopSnapshot(cdm)
```

Arguments

`cdm`	A cdm_reference object.

Value

A summarised_result object.

Examples

```
cdm <- mockOmopSketch(numberIndividuals = 10)

summariseOmopSnapshot(cdm = cdm)
```

`summariseRecordCount` *Summarise record counts of an omop_table using a specific time interval. Only records that fall within the observation period are considered.*

Description

Summarise record counts of an omop_table using a specific time interval. Only records that fall within the observation period are considered.

Usage

```
summariseRecordCount(
  cdm,
  omopTableName,
  interval = "overall",
  ageGroup = NULL,
  sex = FALSE,
  sample = NULL,
  dateRange = NULL
)
```

Arguments

`cdm`	A cdm_reference object.
`omopTableName`	A character vector of omop tables from the cdm.
`interval`	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
`ageGroup`	A list of age groups to stratify results by.
`sex`	Whether to stratify by sex (TRUE) or not (FALSE).
`sample`	An integer to sample the tables to only that number of records. If NULL no sample is done.
`dateRange`	A vector of two dates defining the desired study period. Only the `start_date` column of the OMOP table is checked to ensure it falls within this range. If `dateRange` is `NULL`, no restriction is applied.

Value

A summarised_result object.

Examples

```
library(dplyr, warn.conflicts = FALSE)

cdm <- mockOmopSketch()
```

```
summarisedResult <- summariseRecordCount(
  cdm = cdm,
  omopTableName = c("condition_occurrence", "drug_exposure"),
  interval = "years",
  ageGroup = list("<=20" = c(0, 20), ">20" = c(21, Inf)),
  sex = TRUE
)

summarisedResult |>
  glimpse()

PatientProfiles::mockDisconnect(cdm = cdm)
```

`tableClinicalRecords` *Create a visual table from a summariseClinicalRecord() output.*

Description

Create a visual table from a summariseClinicalRecord() output.

Usage

```
tableClinicalRecords(result, type = "gt")
```

Arguments

`result`	Output from summariseClinicalRecords().
`type`	Type of formatting output table. See `visOmopResults::tableType()` for allowed options.

Value

A formatted table object with the summarised data.

Examples

```
cdm <- mockOmopSketch()

summarisedResult <- summariseClinicalRecords(
  cdm = cdm,
  omopTableName = c("condition_occurrence", "drug_exposure"),
  recordsPerPerson = c("mean", "sd"),
  inObservation = TRUE,
  standardConcept = TRUE,
  sourceVocabulary = TRUE,
  domainId = TRUE,
  typeConcept = TRUE
)
```

```
summarisedResult |>
  suppress(minCellCount = 5) |>
  tableClinicalRecords()

PatientProfiles::mockDisconnect(cdm)
```

`tableConceptIdCounts` *Create a visual table from a summariseConceptIdCounts() result.*

Description

Create a visual table from a summariseConceptIdCounts() result.

Usage

```
tableConceptIdCounts(result, display = "overall", type = "reactable")
```

Arguments

`result`	A summarised_result object.
`display`	A character string indicating which subset of the data to display. Options are: • `"overall"`: Show all source and standard concepts. • `"standard"`: Show only standard concepts. • `"source"`: Show only source codes. • `"missing standard"`: Show only source codes that are missing a mapped standard concept.
`type`	Type of formatting output table, either "reactable" or "datatable".

Value

A reactable or datatable object with the summarised data.

Examples

```
library(OmopSketch)
library(CDMConnector)
library(duckdb)

requireEunomia()
con <- dbConnect(duckdb(), eunomiaDir())
cdm <- cdmFromCon(con = con, cdmSchema = "main", writeSchema = "main")

result <- summariseConceptIdCounts(cdm = cdm, omopTableName = "condition_occurrence")
tableConceptIdCounts(result = result, display = "standard")
```

`tableInObservation` *Create a visual table from a summariseInObservation() result.*

Description

Create a visual table from a summariseInObservation() result.

Usage

```
tableInObservation(result, type = "gt")
```

Arguments

`result`	A summarised_result object.
`type`	Type of formatting output table. See `visOmopResults::tableType()` for allowed options. Default is `"gt"`

Value

A formatted table object with the summarised data.

Examples

```
library(dplyr, warn.conflicts = FALSE)

cdm <- mockOmopSketch()

result <- summariseInObservation(
  observationPeriod = cdm$observation_period,
  interval = "months",
  output = c("person-days", "record"),
  ageGroup = list("<=60" = c(0, 60), ">60" = c(61, Inf)),
  sex = TRUE
)

result |>
  tableInObservation()

PatientProfiles::mockDisconnect(cdm)
```

`tableMissingData` *Create a visual table from a summariseMissingData() result.*

Description

Create a visual table from a summariseMissingData() result.

Usage

```
tableMissingData(result, type = "gt")
```

Arguments

`result`	A summarised_result object.
`type`	Type of formatting output table. See `visOmopResults::tableType()` for allowed options. Default is `"gt"`.

Value

A formatted table object with the summarised data.

Examples

```
cdm <- mockOmopSketch(numberIndividuals = 100)

result <- summariseMissingData(cdm = cdm,
omopTableName = c("condition_occurrence", "visit_occurrence"))

tableMissingData(result = result)

PatientProfiles::mockDisconnect(cdm = cdm)
```

`tableObservationPeriod`

Create a visual table from a summariseObservationPeriod() result.

Description

Create a visual table from a summariseObservationPeriod() result.

Usage

```
tableObservationPeriod(result, type = "gt")
```

Arguments

`result`	A summarised_result object.
`type`	Type of formatting output table. See `visOmopResults::tableType()` for allowed options. Default is `"gt"`.

Value

A formatted table object with the summarised data.

Examples

```
cdm <- mockOmopSketch(numberIndividuals = 100)

result <- summariseObservationPeriod(observationPeriod = cdm$observation_period)

tableObservationPeriod(result = result)

PatientProfiles::mockDisconnect(cdm = cdm)
```

`tableOmopSnapshot` *Create a visual table from a summarise_omop_snapshot result.*

Description

Create a visual table from a summarise_omop_snapshot result.

Usage

```
tableOmopSnapshot(result, type = "gt")
```

Arguments

`result`	Output from summariseOmopSnapshot().
`type`	Type of formatting output table. See `visOmopResults::tableType()` for allowed options. Default is `"gt"`.

Value

A formatted table object with the summarised data.

Examples

```
cdm <- mockOmopSketch(numberIndividuals = 10)

result <- summariseOmopSnapshot(cdm = cdm)

tableOmopSnapshot(result = result)

PatientProfiles::mockDisconnect(cdm)
```

`tableRecordCount` *Create a visual table from a summariseRecordCount() result.*

Description

Create a visual table from a summariseRecordCount() result.

Usage

```
tableRecordCount(result, type = "gt")
```

Arguments

`result`	A summarised_result object.
`type`	Type of formatting output table. See `visOmopResults::tableType()` for allowed options. Default is `"gt"`.

Value

A formatted table object with the summarised data.

Examples

```
library(dplyr, warn.conflicts = FALSE)

cdm <- mockOmopSketch()

summarisedResult <- summariseRecordCount(
  cdm = cdm,
  omopTableName = c("condition_occurrence", "drug_exposure"),
  interval = "years",
  ageGroup = list("<=20" = c(0, 20), ">20" = c(21, Inf)),
  sex = TRUE
)

tableRecordCount(result = summarisedResult)

PatientProfiles::mockDisconnect(cdm = cdm)
```

`tableTopConceptCounts` *Create a visual table of the most common concepts from* `summariseConceptIdCounts()` *output. This function takes a* `summarised_result` *object and generates a formatted table highlighting the most frequent concepts.*

Description

Create a visual table of the most common concepts from `summariseConceptIdCounts()` output. This function takes a `summarised_result` object and generates a formatted table highlighting the most frequent concepts.

Usage

```
tableTopConceptCounts(result, top = 10, countBy = NULL, type = "gt")
```

Arguments

`result`	A `summarised_result` object, typically returned by `summariseConceptIdCounts()`.
`top`	Integer. The number of top concepts to display. Defaults to `10`.
`countBy`	Either 'person' or 'record'. If NULL whatever is in the data is used.
`type`	Character. The output table format. Defaults to `"gt"`. Use `visOmopResults::tableType()` to see all supported formats.

Value

A formatted table object displaying the top concepts from the summarised data.

Examples

```
library(OmopSketch)
library(CDMConnector)
library(duckdb)

requireEunomia()
con <- dbConnect(drv = duckdb(dbdir = eunomiaDir()))
cdm <- cdmFromCon(con = con, cdmSchema = "main", writeSchema = "main")

result <- summariseConceptIdCounts(cdm = cdm, omopTableName = "condition_occurrence")

tableTopConceptCounts(result = result, top = 5)
```

Index