

Package ‘OmopSketch’

August 27, 2026

Title Characterise Tables of an OMOP Common Data Model Instance

Version 1.1.1

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 7.3.3

Suggests CDMConnector (>= 1.3.0), CodelistGenerator, CohortCharacteristics, DBI, duckdb, DT, flextable, gt, here, knitr, lubridate, odbc, OmopViewer (>= 0.5.0), sortable, reactable, remotes, rmarkdown, RPostgres, shinyWidgets, testthat (>= 3.0.0), withr, omock (>= 0.4.0), covr, ggplot2, visOmopResults (>= 1.4.2), devtools, usethis, plotly

Config/testthat/edition 3

Config/testthat/parallel true

Imports cli, clock, CohortConstructor (>= 0.3.1), dplyr, lifecycle, omopgenerics (>= 1.3.1), PatientProfiles (>= 1.4.3), purrr, rlang, stringr, tibble, tidyr

Depends R (>= 4.2.0)

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 2026-08-27 11:10:02 UTC

Contents

clinicalTables	3
databaseCharacteristics	3
mockOmopSketch	5
plotConceptSetCounts	6
plotInObservation	7
plotObservationPeriod	8
plotPerson	9
plotRecordCount	10
plotTrend	11
shinyCharacteristics	12
summariseClinicalRecords	14
summariseConceptCounts	16
summariseConceptIdCounts	17
summariseConceptSetCounts	18
summariseInObservation	19
summariseMissingData	20
summariseObservationPeriod	22
summariseOmopSnapshot	23
summarisePerson	24
summariseRecordCount	25
summariseTrend	26
tableClinicalRecords	28
tableConceptIdCounts	29
tableInObservation	30
tableMissingData	31
tableObservationPeriod	32
tableOmopSnapshot	34
tablePerson	35
tableRecordCount	37
tableTopConceptCounts	38
tableTrend	39

Index	41
--------------	-----------

clinicalTables	<i>Tables in the cdm_reference that contain clinical information</i>
----------------	--

Description

This function provides a list of allowed inputs for the omopTableName argument in summariseClinicalRecords().

Usage

```
clinicalTables()
```

Value

A character vector with table names.

Examples

```
library(OmopSketch)

clinicalTables()
```

databaseCharacteristics	<i>Summarise Database Characteristics for OMOP CDM</i>
-------------------------	--

Description

Summarise Database Characteristics for OMOP CDM

Usage

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

Arguments

<code>cdm</code>	A <code>cdm_reference</code> object. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
<code>omopTableName</code>	A character vector of the names of the tables to summarise in the <code>cdm</code> object. Run <code>clinicalTables()</code> to check the available options.
<code>sample</code>	Either an integer or a character string. • If an integer ($n > 0$), the function will first sample n distinct <code>person_ids</code> from the <code>person</code> table and then subset the input tables to those subjects. • If a character string, it must be the name of a cohort in the <code>cdm</code>; in this case, the input tables are subset to subjects (<code>subject_id</code>) belonging to that cohort. • Use <code>NULL</code> to disable subsetting (default value).
<code>sex</code>	Logical; whether to stratify results by sex (<code>TRUE</code>) or not (<code>FALSE</code>).
<code>ageGroup</code>	A list of age groups to stratify the results by. Each element represents a specific age range. You can give them specific names, e.g. <code>ageGroup = list(children = c(0, 17), adult = c(18, Inf))</code> .
<code>dateRange</code>	A vector of two dates defining the desired study period. Only the <code>start_date</code> column of the OMOP table is checked to ensure it falls within this range. If <code>dateRange</code> is <code>NULL</code> , no restriction is applied.
<code>interval</code>	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
<code>conceptIdCounts</code>	Logical; whether to summarise concept ID counts (<code>TRUE</code>) or not (<code>FALSE</code>).
<code>...</code>	additional arguments passed to the <code>OmopSketch</code> functions that are used internally.

Value

A `summarised_result` object with the results.

Examples

```
## Not run:
library(OmopSketch)
library(omock)
library(dplyr)
library(here)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

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

```

      conceptIdCounts = FALSE
    )

    result |>
      glimpse()

    shinyCharacteristics(result = result, directory = here())

    cdmDisconnect(cdm = cdm)

    ## End(Not run)

```

mockOmopSketch	<i>Creates a mock database to test OmopSketch package</i>
----------------	---

Description

[Deprecated]

Usage

```

mockOmopSketch(
  numberIndividuals = 100,
  con = lifecycle::deprecated(),
  writeSchema = lifecycle::deprecated(),
  seed = lifecycle::deprecated()
)

```

Arguments

numberIndividuals	Number of individuals to create in the cdm reference object.
con	deprecated.
writeSchema	deprecated.
seed	deprecated.

Value

A mock cdm_reference object.

plotConceptSetCounts *Plot the concept counts of a summariseConceptSetCounts output*

Description

[Deprecated]

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: visOmapResults::tidyColumns().
colour	Columns to colour by. See possible columns to colour by with: visOmapResults::tidyColumns().

Value

A plot visualisation.

Examples

```
library(dplyr)
library(OmapSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

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"
  )

cdmDisconnect(cdm = cdm)
```

plotInObservation	Create a ggplot2 plot from the output of summariseInObservation()
-------------------	---

Description

[Deprecated]

Usage

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

Arguments

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

Value

A plot visualisation.

Examples

```
library(dplyr)
library(OmopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

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 == "Person-days") |>
  plotInObservation(facet = "sex", colour = "age_group")

cdmDisconnect(cdm = 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,
  style = NULL,
  type = NULL
)
```

Arguments

result	A summarised_result object (output of summariseObservationPeriod()).
variableName	The variable to plot it can be: "Number subjects", "Records per person", "Duration in days" or "Days to next observation period".
plotType	The plot type, it can be: "barplot", "boxplot", "densityplot" or "cumulativeplot".
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().
style	Visual theme to apply. Character, or NULL. If a character, this may be either the name of a built-in style (see visOmopResults::plotStyle()), or a path to a .yaml file that defines a custom style. If NULL, the function will use the explicit default style, unless a global style option is set (see visOmopResults::setGlobalPlotOptions()) or a _brand.yaml file is present (in that order).
type	Character string indicating the output plot format. See visOmopResults::plotType() for the list of supported plot types. If type = NULL, the function will use the global setting defined via visOmopResults::setGlobalPlotOptions() (if available); otherwise, a standard ggplot2 plot is produced by default.

Value

A plot visualisation.

Examples

```

library(omopSketch)
library(dplyr, warn.conflicts = FALSE)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summariseObservationPeriod(cdm = cdm)

tableObservationPeriod(result = result)

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

cdmDisconnect(cdm = cdm)

```

plotPerson

*Visualise the output of summarisePerson()***Description**

Visualise the output of summarisePerson()

Usage

```
plotPerson(result, variableName = NULL, style = NULL, type = NULL)
```

Arguments

result	A summarised_result object (output of summarisePerson()).
variableName	The variable to plot, a choice between unique(result\$variable_name). If NULL it will only work if only one variable is present in the result object.
style	Visual theme to apply. Character, or NULL. If a character, this may be either the name of a built-in style (see visOmopResults::plotStyle()), or a path to a .yaml file that defines a custom style. If NULL, the function will use the explicit default style, unless a global style option is set (see visOmopResults::setGlobalPlotOptions()) or a _brand.yaml file is present (in that order).
type	Character string indicating the output plot format. See visOmopResults::plotType() for the list of supported plot types. If type = NULL, the function will use the global setting defined via visOmopResults::setGlobalPlotOptions() (if available); otherwise, a standard ggplot2 plot is produced by default.

Value

A plot visualisation.

Examples

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

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summarisePerson(cdm = cdm)

tablePerson(result = result)

cdmDisconnect(cdm = cdm)
```

plotRecordCount	<i>Create a ggplot of the records' count trend</i>
-----------------	--

Description

[Deprecated]

Usage

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

Arguments

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

Value

A plot visualisation.

Examples

```
library(omock)
library(omopSketch)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")
```

```

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 ~ .
)

cdmDisconnect(cdm = cdm)

```

plotTrend

Create a ggplot2 plot from the output of summariseTrend()

Description

Create a ggplot2 plot from the output of summariseTrend()

Usage

```

plotTrend(
  result,
  output = NULL,
  facet = "type",
  colour = NULL,
  style = NULL,
  type = NULL
)

```

Arguments

result	A summarised_result object (output of summariseTrend()).
output	The output to plot. Accepted values are: "record", "person", "person-days", "age", "sex" If not specified, the function will default to: - the only available output if there is just one in the results, or "record" if multiple outputs are present.
facet	Columns to facet by. Formula format can be provided. See possible columns to facet by with: visOmoResults::tidyColumns().
colour	Columns to colour by. See possible columns to colour by with: visOmoResults::tidyColumns().

style	Visual theme to apply. Character, or NULL. If a character, this may be either the name of a built-in style (see <code>visOmapResults::plotStyle()</code>), or a path to a .yml file that defines a custom style. If NULL, the function will use the explicit default style, unless a global style option is set (see <code>visOmapResults::setGlobalPlotOptions()</code>) or a <code>_brand.yml</code> file is present (in that order).
type	Character string indicating the output plot format. See <code>visOmapResults::plotType()</code> for the list of supported plot types. If <code>type = NULL</code> , the function will use the global setting defined via <code>visOmapResults::setGlobalPlotOptions()</code> (if available); otherwise, a standard <code>ggplot2</code> plot is produced by default.

Value

A plot visualisation.

Examples

```
library(dplyr)
library(OmapSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summariseTrend(cdm,
  episode = "observation_period",
  output = c("person-days", "record"),
  interval = "years",
  ageGroup = list("<=40" = c(0, 40), ">40" = c(41, Inf)),
  sex = TRUE
)

plotTrend(
  result = result,
  output = "record",
  colour = "sex",
  facet = "age_group"
)

cdmDisconnect(cdm = cdm)
```

shinyCharacteristics	<i>Generate an interactive Shiny application that visualises the results obtained from the <code>databaseCharacteristics()</code> function</i>
----------------------	--

Description

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

Usage

```
shinyCharacteristics(
  result,
  directory,
  background = TRUE,
  title = "Database characterisation",
  logo = "ohdsi",
  theme = NULL
)
```

Arguments

<code>result</code>	A summarised_result object (output of databaseCharacteristics()).
<code>directory</code>	A character string specifying the directory where the application will be saved.
<code>background</code>	Background panel for the Shiny app. • If set to TRUE (default), a standard background panel with a general description will be included. • If set to FALSE, no background panel will be displayed. • If it is a path (e.g., "path/to/file.md") the file will be used as background panel of your shiny App.
<code>title</code>	Title of the shiny. Default is "Characterisation".
<code>logo</code>	Name of a logo or path to a logo. If NULL no logo is included. Only svg format allowed for the moment.
<code>theme</code>	A character string specifying the theme for the Shiny application. It can be any of the OmopViewer supported themes.

Value

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

Examples

```
## Not run:
library(OmopSketch)
library(omock)
library(here)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

res <- databaseCharacteristics(cdm = cdm)

shinyCharacteristics(result = res, directory = here())

cdmDisconnect(cdm = cdm)

## End(Not run)
```

summariseClinicalRecords

Summarise an omop table from a cdm object

Description

You will obtain information related to the number of records, number of subjects, whether the records are in observation, number of present domains, number of present concepts, missing data and inconsistencies in start date and end date.

Usage

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

Arguments

cdm	A cdm_reference object. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
omopTableName	A character vector of the names of the tables to summarise in the cdm object. Run <i>clinicalTables()</i> to check the available options.
recordsPerPerson	Generates summary statistics for the number of records per person. Set to NULL if no summary statistics are required.
conceptSummary	Logical. If TRUE, includes summaries of concept-level information, including: • Domain ID of standard concepts. • Type concept ID. • Standard vs non-standard concepts. • Source vocabulary usage.
missingData	Logical. If TRUE, includes a summary of missing data for relevant fields.
quality	Logical. If TRUE, performs basic data quality checks, including:

	• Percentage of records within the observation period. • Number of records with end date before start date. • Number of records with start date before the person's birth date.
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. You can give them specific names, e.g. <code>ageGroup = list(children = c(0, 17), adult = c(18, Inf))</code> .
dateRange	A vector of two dates defining the desired study period. Only the <code>start_date</code> column of the OMOP table is checked to ensure it falls within this range. If <code>dateRange</code> is NULL, no restriction is applied.
inObservation	Deprecated. Use <code>quality = TRUE</code> instead.
standardConcept	Deprecated. Use <code>conceptSummary = TRUE</code> instead.
sourceVocabulary	Deprecated. Use <code>conceptSummary = TRUE</code> instead.
domainId	Deprecated. Use <code>conceptSummary = TRUE</code> instead.
typeConcept	Deprecated. Use <code>conceptSummary = TRUE</code> instead.

Value

A summarised_result object with the results.

Examples

```
library(OmopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summariseClinicalRecords(
  cdm = cdm,
  omopTableName = "condition_occurrence",
  recordsPerPerson = c("mean", "sd"),
  quality = TRUE,
  conceptSummary = TRUE,
  missingData = TRUE
)

tableClinicalRecords(result = result)

cdmDisconnect(cdm = cdm)
```

summariseConceptCounts

Summarise concept counts in patient-level data

Description

Only concepts recorded during observation period are counted.

Usage

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

Arguments

cdm	A cdm_reference object. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
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	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. You can give them specific names, e.g. <code>ageGroup = list(children = c(0, 17), adult = c(18, Inf))</code> .
dateRange	A vector of two dates defining the desired study period. Only the <code>start_date</code> column of the OMOP table is checked to ensure it falls within this range. If <code>dateRange</code> is NULL, no restriction is applied.

Details

[Deprecated]

Value

A summarised_result object with the results.

summariseConceptIdCounts

Summarise concept use in patient-level data

Description

Only concepts recorded during observation period are counted.

Usage

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

Arguments

cdm	A cdm_reference object. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
omopTableName	A character vector of the names of the tables to summarise in the cdm object. Run <code>clinicalTables()</code> to check the available options.
countBy	Either "record" for record-level counts or "person" for person-level counts.
interval	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
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. You can give them specific names, e.g. <code>ageGroup = list(children = c(0, 17), adult = c(18, Inf))</code> .
inObservation	Logical. If TRUE, the results are stratified to indicate whether each record occurs within an observation period.
sample	Either an integer or a character string. - If an integer ($n > 0$), the function will first sample n distinct <code>person_ids</code> from the <code>person</code> table and then subset the input tables to those subjects. - If a character string, it must be the name of a cohort in the cdm; in this case, the input tables are subset to subjects (<code>subject_id</code>) belonging to that cohort.

	• Use NULL to disable subsetting (default value).
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.
year	deprecated.

Value

A summarised_result object with the results.

Examples

```
library(OmopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summariseConceptIdCounts(
  cdm = cdm,
  omopTableName = "condition_occurrence",
  countBy = c("record", "person"),
  sex = TRUE
)

tableConceptIdCounts(result = result)

cdmDisconnect(cdm = cdm)
```

summariseConceptSetCounts

Summarise concept counts in patient-level data

Description

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_reference object. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
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	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. You can give them specific names, e.g. <code>ageGroup = list(children = c(0, 17), adult = c(18, Inf))</code> .
dateRange	A vector of two dates defining the desired study period. Only the <code>start_date</code> column of the OMOP table is checked to ensure it falls within this range. If <code>dateRange</code> is NULL, no restriction is applied.

Details**[Deprecated]****Value**

A summarised_result object with the results.

summariseInObservation

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

Description**[Deprecated]****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 the results by. Each element represents a specific age range. You can give them specific names, e.g. ageGroup = list(children = c(0, 17), adult = c(18, Inf)).
sex	Logical; whether to stratify results 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 results.

summariseMissingData	<i>Summarise missing data in omop tables</i>
----------------------	--

Description

Summarise missing data in omop tables

Usage

```
summariseMissingData(
  cdm,
  omopTableName,
  col = NULL,
  sex = FALSE,
  interval = "overall",
  ageGroup = NULL,
  sample = 1e+05,
  dateRange = NULL,
  year = lifecycle::deprecated()
)
```

Arguments

<code>cdm</code>	A <code>cdm_reference</code> object. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
<code>omopTableName</code>	A character vector of the names of the tables to summarise in the <code>cdm</code> object. Run <code>clinicalTables()</code> to check the available options.
<code>col</code>	A character vector of column names to check for missing values. If <code>NULL</code> , all columns in the specified tables are checked. Default is <code>NULL</code> .
<code>sex</code>	Logical; whether to stratify results by sex (<code>TRUE</code>) or not (<code>FALSE</code>).
<code>interval</code>	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
<code>ageGroup</code>	A list of age groups to stratify the results by. Each element represents a specific age range. You can give them specific names, e.g. <code>ageGroup = list(children = c(0, 17), adult = c(18, Inf))</code> .
<code>sample</code>	Either an integer or a character string. • If an integer ($n > 0$), the function will first sample n distinct <code>person_ids</code> from the <code>person</code> table and then subset the input tables to those subjects. • If a character string, it must be the name of a cohort in the <code>cdm</code>; in this case, the input tables are subset to subjects (<code>subject_id</code>) belonging to that cohort. • Use <code>NULL</code> to disable subsetting (default value).
<code>dateRange</code>	A vector of two dates defining the desired study period. Only the <code>start_date</code> column of the OMOP table is checked to ensure it falls within this range. If <code>dateRange</code> is <code>NULL</code> , no restriction is applied.
<code>year</code>	deprecated

Value

A summarised_result object with the results.

Examples

```
library(OmopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

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

tableMissingData(result = result)

cdmDisconnect(cdm = 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(
  cdm,
  estimates = c("mean", "sd", "min", "q05", "q25", "median", "q75", "q95", "max",
    "density"),
  missingData = TRUE,
  quality = TRUE,
  byOrdinal = TRUE,
  ageGroup = NULL,
  sex = FALSE,
  dateRange = NULL,
  observationPeriod = lifecycle::deprecated()
)
```

Arguments

cdm	A cdm_reference object. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
estimates	Estimates to summarise the variables of interest (Records per person, Duration in days and Days to next observation period).
missingData	Logical. If TRUE, includes a summary of missing data for relevant fields.
quality	Logical. If TRUE, performs basic data quality checks, including: • Number of subjects not included in person table. • Number of records with end date before start date. • Number of records with start date before the person's birth date.
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 the results by. Each element represents a specific age range. You can give them specific names, e.g. <code>ageGroup = list(children = c(0, 17), adult = c(18, Inf))</code> .
sex	Logical; whether to stratify results by sex (TRUE) or not (FALSE).
dateRange	A vector of two dates defining the desired study period. Only the <code>start_date</code> column of the OMOP table is checked to ensure it falls within this range. If <code>dateRange</code> is NULL, no restriction is applied.
observationPeriod	deprecated.

Value

A summarised_result object with the results.

Examples

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

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summariseObservationPeriod(cdm = cdm)

tableObservationPeriod(result = result)

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

cdmDisconnect(cdm = cdm)
```

summariseOmopSnapshot	<i>Summarise a cdm_reference object creating a snapshot with the meta-data of the cdm_reference object</i>
-----------------------	--

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. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
-----	--

Value

A summarised_result object with the results.

Examples

```
library(OmopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summariseOmopSnapshot(cdm = cdm)

tableOmopSnapshot(result = result)

cdmDisconnect(cdm = cdm)
```

summarisePerson	<i>Summarise person table</i>
-----------------	-------------------------------

Description

Summarise person table

Usage

```
summarisePerson(cdm)
```

Arguments

cdm	A cdm_reference object. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
-----	--

Value

A summarised_result object with the results.

Examples

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

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summarisePerson(cdm = cdm)

tablePerson(result = result)

cdmDisconnect(cdm = cdm)
```

summariseRecordCount	<i>Summarise record counts of an omop_table using a specific time interval</i>
----------------------	--

Description

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. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
omopTableName	A character vector of the names of the tables to summarise in the cdm object. Run <code>clinicalTables()</code> to check the available options.
interval	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
ageGroup	A list of age groups to stratify the results by. Each element represents a specific age range. You can give them specific names, e.g. <code>ageGroup = list(children = c(0, 17), adult = c(18, Inf))</code> .
sex	Logical; whether to stratify results by sex (TRUE) or not (FALSE).
sample	Either an integer or a character string. - If an integer ($n > 0$), the function will first sample n distinct <code>person_ids</code> from the <code>person</code> table and then subset the input tables to those subjects. - If a character string, it must be the name of a cohort in the <code>cdm</code>; in this case, the input tables are subset to subjects (<code>subject_id</code>) belonging to that cohort. - Use <code>NULL</code> to disable subsetting (default value).
dateRange	A vector of two dates defining the desired study period. Only the <code>start_date</code> column of the OMOP table is checked to ensure it falls within this range. If <code>dateRange</code> is <code>NULL</code> , no restriction is applied.

Details

[Deprecated]

Value

A summarised_result object with the results.

Examples

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

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- 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 = result)

cdmDisconnect(cdm = cdm)
```

summariseTrend

Summarise temporal trends in OMOP tables

Description

This function summarises temporal trends from OMOP CDM tables, considering only data within the observation period. It supports both event and episode tables and can report trends such as number of records, number of subjects, person-days, median age, and number of females.

Usage

```
summariseTrend(
  cdm,
  event = NULL,
  episode = NULL,
  output = "record",
  interval = "overall",
  ageGroup = NULL,
  sex = FALSE,
  inObservation = FALSE,
  dateRange = NULL
)
```

Arguments

cdm	A cdm_reference object. Use <i>CDMConnector</i> to create a reference to a database or <i>omock</i> to create a reference to synthetic data.
event	A character vector of OMOP table names to treat as event tables (uses only start date).
episode	A character vector of OMOP table names to treat as episode tables (uses start and end date).
output	A character vector indicating what to summarise. Options include "record" (default), "person", "person-days", "age", "sex". If included, the number of person-days and end dates are computed only for episode tables.
interval	Time interval to stratify by. It can either be "years", "quarters", "months" or "overall".
ageGroup	A list of age groups to stratify the results by. Each element represents a specific age range. You can give them specific names, e.g. <code>ageGroup = list(children = c(0, 17), adult = c(18, Inf))</code> .
sex	Logical; whether to stratify results by sex (TRUE) or not (FALSE).
inObservation	Logical. If TRUE, the results are stratified to indicate whether each record occurs within an observation period.
dateRange	A vector of two dates defining the desired study period. If <code>dateRange</code> is NULL, no restriction is applied.

Details

- **Event tables:** Records are included if their **start date** falls within the study period. Each record contributes to the time interval containing the start date.
- **Episode tables:** Records are included if their **start or end date** overlaps with the study period. Records are **trimmed** to the date range, and contribute to **all** overlapping time intervals between start and end dates.

Value

A summarised_result object with the results.

Examples

```
library(omopSketch)
library(omock)
library(dplyr)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summariseTrend(
  cdm = cdm,
  event = c("condition_occurrence", "drug_exposure"),
  episode = "observation_period",
  output = "person",
  interval = "years",
```

```

ageGroup = list("<=20" = c(0, 20), ">20" = c(21, Inf)),
sex = TRUE,
dateRange = as.Date(c("1950-01-01", "2010-12-31"))
)

plotTrend(result = result, facet = sex ~ omop_table, colour = c("age_group"))

cdmDisconnect(cdm = cdm)

```

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

Description

Create a visual table from a summariseClinicalRecord() output

Usage

```

tableClinicalRecords(
  result,
  header = "cdm_name",
  hide = c(omopgenerics::settingsColumns(result),
    omopgenerics::additionalColumns(result)),
  groupColumn = c("omop_table", omopgenerics::strataColumns(result)),
  type = NULL,
  style = NULL
)

```

Arguments

result	A summarised_result object (output of summariseClinicalRecords()).
header	A vector specifying the elements to include in the header. The order of elements matters, with the first being the topmost header.
hide	Columns to drop from the output table.
groupColumn	Columns to use as group labels, to see options use visOmopResults::tableColumns(result).
type	Character string specifying the desired output table format. See visOmopResults::tableType() for supported table types. If type = NULL, global options (set via visOmopResults::setGlobalTableOpt will be used if available; otherwise, a default 'gt' table is created.
style	Defines the visual formatting of the table. This argument can be provided in one of the following ways: Pre-defined style: Use the name of a built-in style (e.g., "darwin"). See visOmopResults::tableStyle() for available options. YAML file path: Provide the path to an existing .yaml file defining a new style.

3. **List of custom R code:** Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type.

If `style = NULL`, the function will use global options (see `visOmopResults::setGlobalTableOptions()`) or a `_brand.yml` file (if found); otherwise, the default style is applied.

Value

A formatted table visualisation.

Examples

```
library(OmopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

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()

cdmDisconnect(cdm = 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 (output of summariseConceptIdCounts()).
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 formatted table visualisation.

Examples

```
library(OmopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

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

cdmDisconnect(cdm = cdm)
```

tableInObservation	<i>Create a visual table from a summariseInObservation() result</i>
--------------------	---

Description

[Deprecated]

Usage

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

Arguments

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

Value

A formatted table visualisation.

Examples

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

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

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

result |>
  tableInObservation()

cdmDisconnect(cdm = cdm)
```

tableMissingData	<i>Create a visual table from a summariseMissingData() result</i>
------------------	---

Description

Create a visual table from a summariseMissingData() result

Usage

```
tableMissingData(
  result,
  header = "cdm_name",
  hide = c("variable_name", omopgenerics::settingsColumns(result)),
  groupColumn = c("omop_table", omopgenerics::strataColumns(result)),
  type = NULL,
  style = NULL
)
```

Arguments

result	A summarised_result object (output of summariseMissingData()).
header	A vector specifying the elements to include in the header. The order of elements matters, with the first being the topmost header.

hide	Columns to drop from the output table.
groupColumn	Columns to use as group labels, to see options use <code>visOmopResults::tableColumns(result)</code> .
type	Character string specifying the desired output table format. See <code>visOmopResults::tableType()</code> for supported table types. If <code>type = NULL</code> , global options (set via <code>visOmopResults::setGlobalTableOptions()</code>) will be used if available; otherwise, a default 'gt' table is created.
style	Defines the visual formatting of the table. This argument can be provided in one of the following ways: 1. Pre-defined style: Use the name of a built-in style (e.g., "darwin"). See <code>visOmopResults::tableStyle()</code> for available options. 2. YAML file path: Provide the path to an existing .yaml file defining a new style. 3. List of custom R code: Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type. If <code>style = NULL</code>, the function will use global options (see <code>visOmopResults::setGlobalTableOptions()</code>) or a <code>_brand.yaml</code> file (if found); otherwise, the default style is applied.

Value

A formatted table visualisation.

Examples

```
library(OmopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

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

tableMissingData(result = result)

cdmDisconnect(cdm = cdm)
```

tableObservationPeriod

Create a visual table from a summariseObservationPeriod() result

Description

Create a visual table from a `summariseObservationPeriod()` result

Usage

```
tableObservationPeriod(
  result,
  header = "cdm_name",
  hide = omopgenerics::settingsColumns(result),
  groupColumn = omopgenerics::strataColumns(result),
  type = NULL,
  style = NULL
)
```

Arguments

result	A summarised_result object (output of summariseObservationPeriod()).
header	A vector specifying the elements to include in the header. The order of elements matters, with the first being the topmost header.
hide	Columns to drop from the output table.
groupColumn	Columns to use as group labels, to see options use visOmopResults::tableColumns(result).
type	Character string specifying the desired output table format. See visOmopResults::tableType() for supported table types. If type = NULL, global options (set via visOmopResults::setGlobalTableOptions()) will be used if available; otherwise, a default 'gt' table is created.
style	Defines the visual formatting of the table. This argument can be provided in one of the following ways: 1. Pre-defined style: Use the name of a built-in style (e.g., "darwin"). See visOmopResults::tableStyle() for available options. 2. YAML file path: Provide the path to an existing .yaml file defining a new style. 3. List of custom R code: Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type. If style = NULL, the function will use global options (see visOmopResults::setGlobalTableOptions()) or a _brand.yaml file (if found); otherwise, the default style is applied.

Value

A formatted table visualisation.

Examples

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

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summariseObservationPeriod(cdm = cdm)

tableObservationPeriod(result = result)
```

```

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

cdmDisconnect(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,
  header = "cdm_name",
  hide = "variable_level",
  groupColumn = "variable_name",
  type = NULL,
  style = NULL
)

```

Arguments

result	A summarised_result object (output of summariseOmopSnapshot()).
header	A vector specifying the elements to include in the header. The order of elements matters, with the first being the topmost header.
hide	Columns to drop from the output table.
groupColumn	Columns to use as group labels, to see options use visOmopResults::tableColumns(result).
type	Character string specifying the desired output table format. See visOmopResults::tableType() for supported table types. If type = NULL, global options (set via visOmopResults::setGlobalTableOpt will be used if available; otherwise, a default 'gt' table is created.
style	Defines the visual formatting of the table. This argument can be provided in one of the following ways: Pre-defined style: Use the name of a built-in style (e.g., "darwin"). See visOmopResults::tableStyle() for available options. YAML file path: Provide the path to an existing .yaml file defining a new style.

- 3. **List of custome R code:** Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type.

If style = NULL, the function will use global options (seevisO mopResults::setGlobalTableOptions()) or a _brand.yml file (if found); otherwise, the default style is applied.

Value

A formatted table visualisation.

Examples

```
library(O mopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summariseO mopSnapshot(cdm = cdm)

tableO mopSnapshot(result = result)

cdmDisconnect(cdm = cdm)
```

tablePerson	<i>Visualise the results of summarisePerson() into a table</i>
-------------	--

Description

Visualise the results of summarisePerson() into a table
Visualise the output of summarisePerson()

Usage

```
tablePerson(
  result,
  header = "cdm_name",
  hide = omopgenerics::settingsColumns(result),
  groupColumn = character(),
  type = NULL,
  style = NULL
)

tablePerson(
  result,
  header = "cdm_name",
  hide = omopgenerics::settingsColumns(result),
```

```

    groupColumn = character(),
    type = NULL,
    style = NULL
  )

```

Arguments

result	A summarised_result object (output of summarisePerson()).
header	A vector specifying the elements to include in the header. The order of elements matters, with the first being the topmost header.
hide	Columns to drop from the output table.
groupColumn	Columns to use as group labels, to see options use visOmopResults::tableColumns(result).
type	Character string specifying the desired output table format. See visOmopResults::tableType() for supported table types. If type = NULL, global options (set via visOmopResults::setGlobalTableOptions()) will be used if available; otherwise, a default 'gt' table is created.
style	Defines the visual formatting of the table. This argument can be provided in one of the following ways: 1. Pre-defined style: Use the name of a built-in style (e.g., "darwin"). See visOmopResults::tableStyle() for available options. 2. YAML file path: Provide the path to an existing .yaml file defining a new style. 3. List of custom R code: Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type. If style = NULL, the function will use global options (see visOmopResults::setGlobalTableOptions()) or a _brand.yaml file (if found); otherwise, the default style is applied.

Value

A formatted table visualisation.

A formatted table visualisation.

Examples

```

library(OmopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summarisePerson(cdm = cdm)

tablePerson(result = result)

library(OmopSketch)
library(omock)

```

```

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summarisePerson(cdm = cdm)

tablePerson(result = result)

cdmDisconnect(cdm = cdm)

```

tableRecordCount	<i>Create a visual table from a summariseRecordCount() result</i>
------------------	---

Description

[Deprecated]

Usage

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

Arguments

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

Value

A formatted table visualisation.

Examples

```

library(OmopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

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)

cdmDisconnect(cdm = cdm)

```

`tableTopConceptCounts` *Create a visual table of the most common concepts from summariseConceptIdCounts() output*

Description

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 = NULL,
  style = NULL
)
```

Arguments

- | | |
|----------------------|---|
| <code>result</code> | A <code>summarised_result</code> object (output of <code>summariseConceptIdCounts()</code>). |
| <code>top</code> | Integer. The number of top concepts to display. Defaults to 10. |
| <code>countBy</code> | Either 'person' or 'record'. If NULL whatever is in the data is used. |
| <code>type</code> | Character string specifying the desired output table format. See <code>visOmapResults::tableType()</code> for supported table types. If <code>type = NULL</code> , global options (set via <code>visOmapResults::setGlobalTableOptions()</code>) will be used if available; otherwise, a default 'gt' table is created. |
| <code>style</code> | <p>Defines the visual formatting of the table. This argument can be provided in one of the following ways:</p> <ol style="list-style-type: none"> 1. Pre-defined style: Use the name of a built-in style (e.g., "darwin"). See <code>visOmapResults::tableStyle()</code> for available options. 2. YAML file path: Provide the path to an existing .yaml file defining a new style. 3. List of custom R code: Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type. <p>If <code>style = NULL</code>, the function will use global options (see <code>visOmapResults::setGlobalTableOptions()</code>) or a <code>_brand.yaml</code> file (if found); otherwise, the default style is applied.</p> |

Value

A formatted table visualisation.

Examples

```
library(omopSketch)
library(omock)

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

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

tableTopConceptCounts(result = result, top = 5)

cdmDisconnect(cdm = cdm)
```

tableTrend	Create a visual table from a summariseTrend() result
------------	--

Description

Create a visual table from a summariseTrend() result

Usage

```
tableTrend(
  result,
  header = "cdm_name",
  hide = "variable_level",
  groupColumn = c("type", "omop_table"),
  type = NULL,
  style = NULL
)
```

Arguments

result	A summarised_result object (output of summariseTrend()).
header	A vector specifying the elements to include in the header. The order of elements matters, with the first being the topmost header.
hide	Columns to drop from the output table.
groupColumn	Columns to use as group labels, to see options use visOmopResults::tableColumns(result).
type	Type of formatting output table between gt, datatable and reactable. Default is "gt".
style	Defines the visual formatting of the table. This argument can be provided in one of the following ways: Pre-defined style: Use the name of a built-in style (e.g., "darwin"). See visOmopResults::tableStyle() for available options.

2. **YAML file path:** Provide the path to an existing .yaml file defining a new style.
3. **List of custom R code:** Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type.

If style = NULL, the function will use global options (see `visOmopResults::setGlobalTableOptions()`) or a `_brand.yaml` file (if found); otherwise, the default style is applied.

Value

A formatted table visualisation.

Examples

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

cdm <- mockCdmFromDataset(datasetName = "GiBleed", source = "duckdb")

result <- summariseTrend(
  cdm = cdm,
  episode = "observation_period",
  event = c("drug_exposure", "condition_occurrence"),
  interval = "years",
  ageGroup = list("<=20" = c(0, 20), ">20" = c(21, Inf)),
  sex = TRUE
)

tableTrend(result = result)

cdmDisconnect(cdm = cdm)
```

Index

`clinicalTables`, [3](#)

`databaseCharacteristics`, [3](#)

`mockOmopSketch`, [5](#)

`plotConceptSetCounts`, [6](#)

`plotInObservation`, [7](#)

`plotObservationPeriod`, [8](#)

`plotPerson`, [9](#)

`plotRecordCount`, [10](#)

`plotTrend`, [11](#)

`shinyCharacteristics`, [12](#)

`summariseClinicalRecords`, [14](#)

`summariseConceptCounts`, [16](#)

`summariseConceptIdCounts`, [17](#)

`summariseConceptSetCounts`, [18](#)

`summariseInObservation`, [19](#)

`summariseMissingData`, [20](#)

`summariseObservationPeriod`, [22](#)

`summariseOmopSnapshot`, [23](#)

`summarisePerson`, [24](#)

`summariseRecordCount`, [25](#)

`summariseTrend`, [26](#)

`tableClinicalRecords`, [28](#)

`tableConceptIdCounts`, [29](#)

`tableInObservation`, [30](#)

`tableMissingData`, [31](#)

`tableObservationPeriod`, [32](#)

`tableOmopSnapshot`, [34](#)

`tablePerson`, [35](#)

`tableRecordCount`, [37](#)

`tableTopConceptCounts`, [38](#)

`tableTrend`, [39](#)