Package ‘CohortCharacteristics’

July 30, 2024

Type Package
Title Summarise and Visualise Characteristics of Patients in the OMOP CDM
Version 0.2.2
Maintainer Marti Catala <marti.catalasabate@ndorms.ox.ac.uk>
Description Summarise and visualise the characteristics of patients in data mapped to the Observational Medical Outcomes Partnership (OMOP) common data model (CDM).
License Apache License (>= 2)
Encoding UTF-8
RoxygenNote 7.3.2
Suggests covr, duckdb (>= 0.9.0), testthat (>= 3.1.5), knitr, CodelistGenerator, rmarkdown, glue, odbc, ggplot2, spelling, RPostgres, dbplyr, PaRe, here, magick, plotly, ggraph, DT, cowplot, DiagrammeRsvg, DBI, gt, flextable, tictoc, withr, scales, ggrepur, DiagrammeR
Imports CDMConnector (>= 1.3.2), dplyr, tidyr, checkmate, rlang, cli, stringr, omopgenerics (>= 0.2.0), visOmopResults (>= 0.3.0), PatientProfiles (>= 1.0.0), magrittr, snakecase
URL https://darwin-eu-dev.github.io/CohortCharacteristics/
BugReports https://github.com/darwin-eu-dev/CohortCharacteristics/issues
Language en-US
Depends R (>= 2.10)
Config/testthat/edition 3
Config/testthat/parallel true
VignetteBuilder knitr
NeedsCompilation no
Author Marti Catala [aut, cre] (<https://orcid.org/0000-0003-3308-9905>), Yuchen Guo [aut] (<https://orcid.org/0000-0002-0847-4855>), Mike Du [aut] (<https://orcid.org/0000-0002-9517-8834>),
mockCohortCharacteristics

It creates a mock database for testing CohortCharacteristics package

Description

It creates a mock database for testing CohortCharacteristics package
Usage

mockCohortCharacteristics(
  con = NULL,
  writeSchema = NULL,
  numberIndividuals = 10,
  ..., 
  seed = NULL
)

Arguments

con         A DBI connection to create the cdm mock object.
writeSchema Name of an schema on the same connection with writing permission.
numberIndividuals Number of individuals to create in the cdm reference.
...          User self defined tables to put in cdm, it can input as many as the user want.
seed        A number to set the seed. If NULL seed is not used.

Value

A mock cdm_reference object created following user’s specifications.

Examples

library(CohortCharacteristics)
library(CDMConnector)

cdm <- mockCohortCharacteristics()
mockDisconnect(cdm = cdm)

optionsTableCharacteristics

Additional arguments for the function tableCharacteristics.

Description

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

Usage

optionsTableCharacteristics()
Value

The default .options named list.

Examples

{  
  optionsTableCharacteristics()
}

optionsTableCohortOverlap

Additional arguments for the function tableCohortOverlap.

Description

It provides a list of allowed inputs for .option argument in tableCohortOverlap and their given default value.

Usage

optionsTableCohortOverlap()

Value

The default .options named list.

Examples

{  
  optionsTableCohortOverlap()
}

optionsTableCohortTiming

Additional arguments for the function tableCohortTiming.

Description

It provides a list of allowed inputs for .option argument in tableCohortTiming and their given default value.

Usage

optionsTableCohortTiming()
plotCharacteristics

Value

The default .options named list.

Examples

```
{
  optionsTableCohortTiming()
}
```

Description

Create a ggplot from the output of summariseCharacteristics. ‘r lifecycle::badge("experimental")’

Usage

```
plotCharacteristics(
  data,
  x = "variable_name",
  plotStyle = "barplot",
  facet = NULL,
  colour = NULL,
  colourName = NULL,
  .options = list()
)
```

Arguments

- **data**: output of summariseCharacteristics.
- **x**: what to plot on x axis, default as variable_name column. Has to be a column in data.
- **plotStyle**: Now allows boxplot or barplot only.
- **facet**: Variables to facet by
- **colour**: column in data to color by.
- **colourName**: Colour legend name
- **.options**: Additional plotting options.

Value

A ggplot.
Examples

```r
library(CohortCharacteristics)
library(dplyr)

cdm <- mockCohortCharacteristics()

results <- summariseCharacteristics(
  cohort = cdm$cohort1,
  ageGroup = list(c(0, 19), c(20, 39), c(40, 59), c(60, 79), c(80, 150)),
  tableIntersectCount = list(
    tableName = "visit_occurrence", window = c(-365, -1)
  ),
  cohortIntersectFlag = list(
    targetCohortTable = "cohort2", window = c(-365, -1)
  )
)

results |>
  filter(
    variable_name == "Cohort2 flag -365 to -1",
    estimate_name == "percentage"
  ) |>
  plotCharacteristics(
    plotStyle = "barplot",
    colour = "variable_level",
    x = "variable_level",
    facet = c(
      "cdm_name",
      "group_level",
      "strata_level"
    )
  )

mockDisconnect(cdm = cdm)
```

**plotCohortAttrition**

create a ggplot from the output of summariseLargeScaleCharacteristics.

Description

`r lifecycle::badge("experimental")`

Usage

```r
plotCohortAttrition(x, cohortId = NULL)
```
Arguments

- **x**: attrition table
- **cohortId**: target cohort_definition_id

Value

A dgr_graph

Examples

```r
library(omopgenerics)
library(dplyr)
library(DiagrammeR)

cdm <- mockCohortCharacteristics(numberIndividuals = 1000)

cdm[["cohort1"]]<-
  cdm[["cohort1"]]
  |> filter(year(cohort_start_date) >= 2000) |
  |> recordCohortAttrition("Restrict to cohort_start_date >= 2000") |
  |> filter(year(cohort_end_date) < 2020) |
  |> recordCohortAttrition("Restrict to cohort_end_date < 2020") |
  |> compute(temporary = FALSE, name = "cohort1")

cdm$cohort1 |
  summariseCohortAttrition() |
  plotCohortAttrition(cohortId = 2)
```

Description

'r lifecycle::badge("experimental")'

Usage

```r
plotCohortOverlap(
  result,
  facet = NULL,
  uniqueCombinations = TRUE,
  .options = list()
)
```
plotCohortTiming

Arguments

result A summariseCohortOverlap result.
facet Variables to facet by.
uniqueCombinations If TRUE, only unique combinations of reference and comparator plots will be plotted.
.options Additional plotting options

Value

A ggplot.

Examples

library(CohortCharacteristics)
cdm <- CohortCharacteristics::mockCohortCharacteristics()
overlap <- summariseCohortOverlap(cdm$cohort2)
plotCohortOverlap(overlap)

plotCohortTiming

Plot summariseCohortTiming results.

Description

'\texttt{lifecycle::badge("experimental")}'

Usage

plotCohortTiming(
  result,
  plotType = "boxplot",
  timeScale = "days",
  facet = NULL,
  colour = NULL,
  colourName = NULL,
  uniqueCombinations = TRUE,
  .options = list()
)

Arguments

result A summariseCohortTiming result.
plotType Type of desired formatted table, possibilities are "boxplot" and "density".
timeScale Time scale to plot results. Can be days or years.
plotComparedLargeScaleCharacteristics

facets variables to facet by
colour Variables to use for colours
colourName Colour legend name
uniqueCombinations If TRUE, only unique combinations of reference and comparator plots will be plotted.
.options Additional plotting options

Value
A ggplot.

plotComparedLargeScaleCharacteristics create a ggplot from the output of summariseLargeScaleCharacteristics.

Description
‘r lifecycle::badge("experimental")’

Usage
plotComparedLargeScaleCharacteristics(
  data,
  referenceGroupLevel = NULL,
  referenceStrataLevel = NULL,
  referenceVariableLevel = NULL,
  referenceCdmName = NULL,
  splitStrata = FALSE,
  facet = NULL,
  colorVars = NULL,
  missings = 0
)

Arguments
data output of summariseLargeScaleCharacteristics().
referenceGroupLevel group_level value to be used as the reference.
referenceStrataLevel strata_level value to be used as the reference.
referenceVariableLevel variable_level value to be used as the reference.
referenceCdmName cdm_name value to be used as the reference.
plotLargeScaleCharacteristics

create a ggplot from the output of summariseLargeScaleCharacteristics.

Description

'\texttt{r lifecycle::badge("experimental")}'

Usage

\begin{verbatim}
plotLargeScaleCharacteristics(
  data, 
  position = "horizontal", 
  splitStrata = FALSE, 
  facet = NULL, 
  colorVars = "variable_level"
)
\end{verbatim}

Arguments

- **data**: output of summariseLargeScaleCharacteristics().
- **position**: if set to "horizontal" the horizontal axis will plot "variable_name" column and the vertical axis "estimate_value" column. If "vertical", axis will be the other way around.
- **splitStrata**: boolean variable (TRUE/FALSE)
**facet**
columns in data to facet. If the facet position wants to be specified, use the formula class for the input (e.g., strata + table_name ~ group_level + cdm_name). Variables before "~" will be facet by on horizontal axis, whereas those after "~" on vertical axis. Character format is also allowed (e.g., c("strata","table_name","group_level","cdm_name"). Only the following columns are allowed to be facet by: c("cdm_name", "group_level", "strata_level", "variable_level", "strata", "table_name"). If splitStrata = TRUE, strata levels are also allowed.

**colorVars**
column in data to color by. Only the following columns are allowed to be used: c("cdm_name", "group_level", "strata_level", "variable_level", "strata", "table_name"). If splitStrata = TRUE, strata levels are also allowed.

**Value**
A ggplot.

---

**summariseCharacteristics**

*Summarise characteristics of cohorts in a cohort table*

**Description**
Summarise characteristics of cohorts in a cohort table

**Usage**

```r
summariseCharacteristics(
  cohort,
  cohortId = NULL,
  strata = list(),
  counts = TRUE,
  demographics = TRUE,
  ageGroup = NULL,
  tableIntersectFlag = list(),
  tableIntersectCount = list(),
  tableIntersectDate = list(),
  tableIntersectDays = list(),
  cohortIntersectFlag = list(),
  cohortIntersectCount = list(),
  cohortIntersectDate = list(),
  cohortIntersectDays = list(),
  conceptIntersectFlag = list(),
  conceptIntersectCount = list(),
  conceptIntersectDate = list(),
  conceptIntersectDays = list(),
  otherVariables = character(),
  otherVariablesEstimates = c("min", "q25", "median", "q75", "max", "count", "percentage")
)
```
Arguments

- **cohort**: A cohort table in the cdm.
- **cohortId**: Vector of cohort definition ids to include. If NULL all cohort will be selected.
- **strata**: A list of variables to stratify results. These variables must have been added as additional columns in the cohort table.
- **counts**: TRUE or FALSE. If TRUE, record and person counts will be produced.
- **demographics**: TRUE or FALSE. If TRUE, patient demographics (cohort start date, cohort end date, age, sex, prior observation, and future observation will be summarised).
- **ageGroup**: A list of age groups to stratify results by.
- **tableIntersectFlag**: A list of arguments that uses PatientProfiles::addTableIntersectFlag() to add variables to summarise.
- **tableIntersectCount**: A list of arguments that uses PatientProfiles::addTableIntersectCount() to add variables to summarise.
- **tableIntersectDate**: A list of arguments that uses PatientProfiles::addTableIntersectDate() to add variables to summarise.
- **tableIntersectDays**: A list of arguments that uses PatientProfiles::addTableIntersectDays() to add variables to summarise.
- **cohortIntersectFlag**: A list of arguments that uses PatientProfiles::addCohortIntersectFlag() to add variables to summarise.
- **cohortIntersectCount**: A list of arguments that uses PatientProfiles::addCohortIntersectCount() to add variables to summarise.
- **cohortIntersectDate**: A list of arguments that uses PatientProfiles::addCohortIntersectDate() to add variables to summarise.
- **cohortIntersectDays**: A list of arguments that uses PatientProfiles::addCohortIntersectDays() to add variables to summarise.
- **conceptIntersectFlag**: A list of arguments that uses PatientProfiles::addConceptIntersectFlag() to add variables to summarise.
- **conceptIntersectCount**: A list of arguments that uses PatientProfiles::addConceptIntersectCount() to add variables to summarise.
- **conceptIntersectDate**: A list of arguments that uses PatientProfiles::addConceptIntersectDate() to add variables to summarise.
- **conceptIntersectDays**: A list of arguments that uses PatientProfiles::addConceptIntersectDays() to add variables to summarise.
- **otherVariables**: Other variables contained in cohort that you want to be summarised.
- **otherVariablesEstimates**: Name of the estimates for the otherVariables columns.
**Value**

A summary of the characteristics of the cohorts in the cohort table.

**Examples**

```r
library(dplyr)
library(CohortCharacteristics)
library(PatientProfiles)

cdm <- mockCohortCharacteristics()
cdm$cohort1 |>
  addSex() |>
  addAge(
    ageGroup = list(c(0, 40), c(41, 150))
  ) |>
  summariseCharacteristics(
    strata = list("sex", "age_group"),
    cohortIntersectFlag = list(
      "Cohort 2 Flag" = list(
        targetCohortTable = "cohort2",
        window = c(-365, 0)
      )
    ),
    cohortIntersectCount = list (
      "Cohort 2 Count" = list(
        targetCohortTable = "cohort2",
        window = c(-365, 0)
      )
    )
  ) |>
  glimpse()

mockDisconnect(cdm = cdm)
```

---

**summariseCohortAttrition**

*Summarise attrition associated with cohorts in a cohort table*

**Description**

Summarise attrition associated with cohorts in a cohort table

**Usage**

```r
summariseCohortAttrition(cohort, cohortId = NULL)
```
summariseCohortCount

Arguments

cohort A cohort table in the cdm.
cohortId A cohort definition id to restrict by. If NULL, all cohorts will be included.

Value

A summary of the attrition for the cohorts in the cohort table.

Examples

library(CohortCharacteristics)
cdm <- mockCohortCharacteristics()
summariseCohortAttrition(cohort = cdm$cohort1) |> dplyr::glimpse()
mockDisconnect(cdm = cdm)

summariseCohortCount Summarise counts for cohorts in a cohort table

Description

Summarise counts for cohorts in a cohort table

Usage

summariseCohortCount(cohort, cohortId = NULL, strata = list())

Arguments

cohort A cohort table in the cdm.
cohortId A cohort definition id to restrict by. If NULL, all cohorts will be included.
strata A list of variables to stratify results. These variables must have been added as additional columns in the cohort table.

Value

A summary of counts of the cohorts in the cohort table.

Examples

library(CohortCharacteristics)
cdm <- mockCohortCharacteristics()
summariseCohortCount(cohort = cdm$cohort1) |> dplyr::glimpse()
mockDisconnect(cdm = cdm)
summariseCohortOverlap

Summarise overlap between cohorts in a cohort table

Description

Summarise overlap between cohorts in a cohort table

Usage

summariseCohortOverlap(cohort, cohortId = NULL, strata = list())

Arguments

cohort A cohort table in the cdm.
cohortId A cohort definition id to restrict by. If NULL, all cohorts will be included.
strata A list of variables to stratify results. These variables must have been added as additional columns in the cohort table.

Value

A summary of overlap between cohorts in the cohort table.

Examples

library(CohortCharacteristics)
cdm <- CohortCharacteristics::mockCohortCharacteristics()
summariseCohortOverlap(cdm$cohort2) |> dplyr::glimpse()
mockDisconnect(cdm = cdm)

summariseCohortTiming

Summarise timing between entries into cohorts in a cohort table

Description

Summarise timing between entries into cohorts in a cohort table

Usage

summariseCohortTiming(
  cohort,
  cohortId = NULL,
  strata = list(),
  restrictToFirstEntry = TRUE,
  estimates = c("min", "q25", "median", "q75", "max"),
  density = FALSE
)


Arguments

- **cohort**: A cohort table in a cdm reference.
- **cohortId**: A cohort definition id to restrict by. If NULL, all cohorts will be included.
- **strata**: A list of variables to stratify results. These variables must have been added as additional columns in the cohort table.
- **restrictToFirstEntry**: If TRUE only an individual’s first entry per cohort will be considered. If FALSE all entries per individual will be considered.
- **estimates**: Summary statistics to use when summarising timing.
- **density**: TRUE or FALSE. If TRUE, estimates for a density plot will also be computed.

Value

A summary of timing between entries into cohorts in the cohort table.

Examples

```r
library(CohortCharacteristics)
cdm <- mockCohortCharacteristics(numberIndividuals = 100)
summariseCohortTiming(cdm$cohort2) |> dplyr::glimpse()
mockDisconnect(cdm = cdm)
```

Description

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

Usage

```r
summariseLargeScaleCharacteristics(
  cohort,
  strata = list(),
  window = list(c(-Inf, -366), c(-365, -31), c(-30, -1), c(0, 0), c(1, 30), c(31, 365), c(366, Inf)),
  eventInWindow = NULL,
  episodeInWindow = NULL,
  indexDate = "cohort_start_date",
  censorDate = NULL,
  includeSource = FALSE,
  minimumFrequency = 0.005,
  excludedCodes = c(0)
)
```
summariseLargeScaleCharacteristics

Arguments

- cohort: The cohort to characterise.
- strata: Stratification list.
- window: Temporal windows that we want to characterize.
- eventInWindow: Tables to characterise the events in the window. eventInWindow must be provided if episodeInWindow is not specified.
- episodeInWindow: Tables to characterise the episodes in the window. episodeInWindow must be provided if eventInWindow is not specified.
- 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
- includeSource: Whether to include source concepts.
- minimumFrequency: Minimum frequency covariates to report.
- excludedCodes: Codes excluded.

Value

The output of this function is a ‘ResultSummary’ containing the relevant information.

Examples

```r
library(CohortCharacteristics)
cdm <- CohortCharacteristics::mockCohortCharacteristics()

concept <- dplyr::tibble(
  concept_id = c(1125315, 1503328, 1516978, 317009, 378253, 4266367),
  domain_id = NA_character_,
  vocabulary_id = NA_character_,
  concept_class_id = NA_character_,
  concept_code = NA_character_,
  valid_start_date = as.Date("1900-01-01"),
  valid_end_date = as.Date("2099-01-01")
) |> 
  dplyr::mutate(concept_name = paste0("concept: ", .data$concept_id))

concept <- CDMConnector::insertTable(cdm, "concept", concept)

results <- cdm$cohort2 |> 
  summariseLargeScaleCharacteristics(
    episodeInWindow = c("condition_occurrence"),
    minimumFrequency = 0
  )

mockDisconnect(cdm = cdm)
```
tableCharacteristics  
*Format a summarised_characteristics object into a visual table.*

**Description**

`r lifecycle::badge("experimental")`

**Usage**

```r
tableCharacteristics(
  result,
  type = "gt",
  formatEstimateName = c(`N (%)` = "<count> (<percentage>%)", N = "<count>",
    `Median [Q25 - Q75]` = "<median> [<q25> - <q75>]", `Mean (SD)` = "<mean> (<sd>)",
    Range = "<min> to <max>"),
  header = c("group"),
  split = c("group", "strata"),
  groupColumn = NULL,
  excludeColumns = c("result_id", "estimate_type", "additional_name", "additional_level"),
  .options = list()
)
```

**Arguments**

- **result**: A summarised_characteristics object.
- **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 `<...>`.
- **header**: A vector containing which elements should go into the header in order. Allowed are: `cdm_name`, `group`, `strata`, `additional`, `variable`, `estimate`, `settings`.
- **split**: A vector containing the name-level groups to split ("group", "strata", "additional"), or an empty character vector to not split.
- **groupColumn**: Column to use as group labels.
- **excludeColumns**: Columns to drop from the output table.
- **.options**: Named list with additional formatting options. `CohortCharacteristics::optionsTableCharacteristics()` shows allowed arguments and their default values.

**Value**

A table with a formatted version of the summariseCharacteristics result.
Examples

```r
library(CohortCharacteristics)

cdm <- mockCohortCharacteristics()

cdm$cohort1 |> summariseCharacteristics() |> tableCharacteristics()

mockDisconnect(cdm = cdm)
```

---

tableCohortAttrition Create a visual table from the output of summariseCohortAttrition. `r lifecycle::badge("experimental")`

Description

Create a visual table from the output of summariseCohortAttrition. `r lifecycle::badge("experimental")`

Usage

```r
tableCohortAttrition(
    result,
    header = "cdm_name",
    groupColumn = "cohort_name",
    type = "gt"
)
```

Arguments

- `result`: A summarised_result object generated by summariseCohortAttrition().
- `header`: Columns to use as headers.
- `groupColumn`: Columns to use to group.
- `type`: Whether a 'gt', 'flextable' or a 'tibble' should be created.

Value

A visual table.
Examples

library(CohortCharacteristics)

cdm <- mockCohortCharacteristics()

cdm$cohort2 |> summariseCohortAttrition() |> tableCohortAttrition()

tableCohortCount

Format a summarised_characteristics object into a visual table.

Description

'r lifecycle::badge("experimental")'

Usage

tableCohortCount(
  result,
  type = "gt",
  formatEstimateName = c(N = "<count>")
)

Arguments

result A summarised_characteristics object.

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 <...>.


split A vector containing the name-level groups to split ("group", "strata", "additional"), or an empty character vector to not split.

groupColumn Column to use as group labels.

excludeColumns Columns to drop from the output table.

$options Named list with additional formatting options. CohortCharacteristics::optionsTableCharacteristics() shows allowed arguments and their default values.
**Value**

A table with a formatted version of the summariseCohortResult result.

**Examples**

```r
library(CohortCharacteristics)

cdm <- mockCohortCharacteristics()

cdm$cohort1 |>
  summariseCharacteristics() |>
  tableCharacteristics()

mockDisconnect(cdm = cdm)
```

---

**tableCohortOverlap**

Format a summariseOverlapCohort result into a visual table.

**Description**

`r lifecycle::badge("experimental")`

**Usage**

```r
tableCohortOverlap(
  result,
  type = "gt",
  formatEstimateName = c("N (%) = <count> (<percentage>%)",
  header = c("strata"),
  split = c("group", "strata", "additional"),
  groupColumn = NULL,
  excludeColumns = c("result_id", "estimate_type"),
  .options = list()
)
```

**Arguments**

- **result**: A summariseOverlapCohort result.
- **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 `<...>`.
- **header**: A vector containing which elements should go into the header in order. Allowed are: `cdm_name`, `group`, `strata`, `additional`, `variable`, `estimate`, `settings`. 
tableCohortTiming

Format a summariseCohortTiming result into a visual table.

Description

`r lifecycle::badge("experimental")`

Usage

tableCohortTiming(
    result,
    timeScale = "days",
    type = "gt",
    formatEstimateName = c(N = "<count>", `Median [Q25 - Q75]` =
        "<median> [<q25> - <q75>]", Range = "<min> - <max>"),
    header = c("strata"),
    split = c("group", "strata", "additional"),
    groupColumn = NULL,
    excludeColumns = c("result_id", "estimate_type", "variable_level"),
    .options = list()
)
Arguments

result  A summariseCohortTiming result

timeScale  Time scale to plot results. Can be days or years.

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 <...>.


split  A vector containing the name-level groups to split (”group”, ”strata”, ”additional”), or an empty character vector to not split.

groupColumn  Column to use as group labels.

excludeColumns  Columns to drop from the output table.

$options  named list with additional formatting options. CohortCharacteristics::optionsTableCohortTiming() shows allowed arguments and their default values.

Value

A formatted table of the summariseCohortTiming result.

tableLargeScaleCharacteristics

Format a summarised_large_scale_characteristics object into a visual table.

Description

‘r lifecycle::badge("experimental")’

Usage

tableLargeScaleCharacteristics(
result,
type = ”gt”,
formatEstimateName = c(”N (%)” = ”<count> (<percentage>%)”),
splitStrata = TRUE,
header = c(”cdm name”, ”cohort name”, ”strata”, ”window name”),
topConcepts = NULL
)
### Arguments

- **result**: A summarised_large_scale_characteristics object.
- **type**: Output type ("gt" or "flextable").
- **formatEstimateName**: Named list of estimate name’s to join, sorted by computation order. Indicate estimate_name’s between <...>.
- **splitStrata**: Whether to split strata_group and strata_level to multiple columns.
- **header**: Specify the headers of the table.
- **topConcepts**: Number of concepts to restrict the table.

### Value

A formatted table.

### Examples

```r
## Not run:
library(DBI)
library(duckdb)
library(CDMConnector)

con <- dbConnect(duckdb(), eunomia_dir())
cdm <- cdmFromCon(con = con, cdmSchema = "main", writeSchema = "main")
cdm <- generateConceptCohortSet(
  cdm = cdm,
  conceptSet = list("viral_pharyngitis" = 4112343),
  name = "my_cohort"
)
result <- summariseLargeScaleCharacteristics(
  cohort = cdm$my_cohort,
  eventInWindow = "condition_occurrence",
  episodeInWindow = "drug_exposure"
)
tableLargeScaleCharacteristics(result)

## End(Not run)
```
Index

mockCohortCharacteristics, 2
optionsTableCharacteristics, 3
optionsTableCohortOverlap, 4
optionsTableCohortTiming, 4
plotCharacteristics, 5
plotCohortAttrition, 6
plotCohortOverlap, 7
plotCohortTiming, 8
plotComparedLargeScaleCharacteristics, 9
plotLargeScaleCharacteristics, 10
summariseCharacteristics, 11
summariseCohortAttrition, 13
summariseCohortCount, 14
summariseCohortOverlap, 15
summariseCohortTiming, 15
summariseLargeScaleCharacteristics, 16
tableCharacteristics, 18
tableCohortAttrition, 19
tableCohortCount, 20
tableCohortOverlap, 21
tableCohortTiming, 22
tableLargeScaleCharacteristics, 23