

Package ‘CohortCharacteristics’

April 16, 2024

Type Package

Title Summarise and Visualise Characteristics of Patients in the OMOP CDM

Version 0.1.0

Maintainer Marti Catala <marti.catalasabate@dorms.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 7.3.1

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

Imports CDMConnector (>= 1.3.1), dplyr, tidyr, checkmate, lubridate, rlang, cli, stringr, omopgenerics (>= 0.1.2), visOmopResults (>= 0.2.0), PatientProfiles (>= 0.8.0), ggpubr, magrittr

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>>),

Kim Lopez-Guell [aut] (<<https://orcid.org/0000-0002-8462-8668>>),
 Edward Burn [aut] (<<https://orcid.org/0000-0002-9286-1128>>),
 Nuria Mercade-Besora [ctb] (<<https://orcid.org/0009-0006-7948-3747>>),
 Marta Alcalde [aut] (<<https://orcid.org/0009-0002-4405-1814>>)

Repository CRAN

Date/Publication 2024-04-16 20:50:11 UTC

R topics documented:

mockCohortCharacteristics	2
optionsTableCharacteristics	5
optionsTableCohortOverlap	6
optionsTableCohortTiming	6
plotCharacteristics	7
plotCohortOverlap	8
plotCohortTiming	9
plotLargeScaleCharacteristics	10
summariseCharacteristics	11
summariseCohortCounts	13
summariseCohortIntersect	14
summariseCohortOverlap	15
summariseCohortTiming	15
summariseConceptIntersect	16
summariseLargeScaleCharacteristics	17
summariseTableIntersect	18
tableCharacteristics	19
tableCohortOverlap	20
tableCohortTiming	21
tableLargeScaleCharacteristics	23
Index	25

mockCohortCharacteristics

It creates a mock database for testing PatientProfiles package

Description

It creates a mock database for testing PatientProfiles package

Usage

```
mockCohortCharacteristics(
  connectionDetails = list(con = DBI::dbConnect(duckdb::duckdb(), ":memory:"),
    write_schema = "main", mock_prefix = NULL),
  drug_exposure = NULL,
```

```

drug_strength = NULL,
observation_period = NULL,
condition_occurrence = NULL,
visit_occurrence = NULL,
concept_ancestor = NULL,
person = NULL,
death = NULL,
cohort1 = NULL,
cohort2 = NULL,
drug_concept_id_size = 5,
ancestor_concept_id_size = 5,
condition_concept_id_size = 5,
visit_concept_id_size = 5,
visit_occurrence_id_size = 5,
ingredient_concept_id_size = 1,
drug_exposure_size = 10,
patient_size = 1,
min_drug_exposure_start_date = "2000-01-01",
max_drug_exposure_start_date = "2020-01-01",
earliest_date_of_birth = NULL,
latest_date_of_birth = NULL,
earliest_observation_start_date = NULL,
latest_observation_start_date = NULL,
min_days_to_observation_end = NULL,
max_days_to_observation_end = NULL,
earliest_condition_start_date = NULL,
latest_condition_start_date = NULL,
min_days_to_condition_end = NULL,
max_days_to_condition_end = NULL,
earliest_visit_start_date = NULL,
latest_visit_start_date = NULL,
min_days_to_visit_end = NULL,
max_days_to_visit_end = NULL,
seed = 1,
...
)

```

Arguments

connectionDetails	Connection an details to create the cdm mock object.
drug_exposure	default null user can define its own table.
drug_strength	default null user can define its own table.
observation_period	default null user can define its own table.
condition_occurrence	default null user can define its own table.
visit_occurrence	default null user can define its own visit_occurrence table.

concept_ancestor the concept ancestor table.
 person default null user can define its own table.
 death default null user can define its own table
 cohort1 cohort table for test to run in getindication.
 cohort2 cohort table for test to run in getindication.
 drug_concept_id_size number of unique drug concept id.
 ancestor_concept_id_size the size of concept ancestor table.
 condition_concept_id_size number of unique row in the condition concept table.
 visit_concept_id_size number of unique visit concept id.
 visit_occurrence_id_size number of unique visit occurrence id.
 ingredient_concept_id_size number of unique drug ingredient concept id.
 drug_exposure_size number of unique drug exposure.
 patient_size number of unique patient.
 min_drug_exposure_start_date user define minimum drug exposure start date.
 max_drug_exposure_start_date user define maximum drug exposure start date.
 earliest_date_of_birth the earliest date of birth of patient in person table format "dd-mm-yyyy".
 latest_date_of_birth the latest date of birth for patient in person table format "dd-mm-yyyy".
 earliest_observation_start_date the earliest observation start date for patient format "dd-mm-yyyy".
 latest_observation_start_date the latest observation start date for patient format "dd-mm-yyyy".
 min_days_to_observation_end the minimum number of days of the observational integer.
 max_days_to_observation_end the maximum number of days of the observation period integer.
 earliest_condition_start_date the earliest condition start date for patient format "dd-mm-yyyy".
 latest_condition_start_date the latest condition start date for patient format "dd-mm-yyyy".
 min_days_to_condition_end the minimum number of days of the condition integer.

max_days_to_condition_end	the maximum number of days of the condition integer.
earliest_visit_start_date	the earliest visit start date for patient format "dd-mm-yyyy".
latest_visit_start_date	the latest visit start date for patient format "dd-mm-yyyy".
min_days_to_visit_end	the minimum number of days of the visit integer.
max_days_to_visit_end	the maximum number of days of the visit integer.
seed	seed.
...	user self defined tibble table to put in cdm, it can input as many as the user want.

Value

cdm of the mock database following user's specifications.

Examples

```
library(CohortCharacteristics)
cdm <- mockCohortCharacteristics()
cdm
CDMConnector::cdmDisconnect(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()
```

Value

The default .options named list.

Examples

```
{
  optionsTableCohortTiming()
}
```

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

Description

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

Usage

```
plotCharacteristics(
  data,
  xAxis = "variable_name",
  yAxis = "estimate_value",
  plotStyle = "barplot",
  facetVarX = NULL,
  facetVarY = NULL,
  colorVars = NULL,
  vertical_x = FALSE
)
```

Arguments

<code>data</code>	output of summariseCharacteristics.
<code>xAxis</code>	what to plot on x axis, default as variable_name column. Has to be a column in data.
<code>yAxis</code>	what to plot on y axis, default as estimate_value column. Has to be a column in data. One of the xAxis or yAxis has to be estimate_value.
<code>plotStyle</code>	Now allows boxplot or barplot only.
<code>facetVarX</code>	column in data to facet by on horizontal axis
<code>facetVarY</code>	column in data to facet by on vertical axis
<code>colorVars</code>	column in data to color by.
<code>vertical_x</code>	whether to display x axis string vertically.

Value

A ggplot.

Examples

```

library(CohortCharacteristics)

cdm <- mockCohortCharacteristics()

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

plotCharacteristics(results)

CDMConnector::cdmDisconnect(cdm = cdm)

```

plotCohortOverlap *Plot the result of summariseCohortOverlap.*

Description

‘r lifecycle::badge("experimental")‘

Usage

```

plotCohortOverlap(
  result,
  facetVarX = "variable_name",
  facetVarY = "strata_level",
  colorVars = "variable_level",
  overlapLabel = "{cohort_name_reference} &&& {cohort_name_comparator}",
  uniqueCombinations = TRUE
)

```

Arguments

result	A summariseCohortOverlap result.
facetVarX	column in data to facet by on horizontal axis
facetVarY	column in data to facet by on vertical axis
colorVars	column in data to color by
overlapLabel	A glue expression to identify each plotted cohort overlap.

uniqueCombinations

If TRUE, only unique combinations of reference and comparator plots will be plotted.

Value

A ggplot.

Examples

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

plotCohortTiming *Plot summariseCohortTiming results.*

Description

‘r lifecycle::badge("experimental")‘

Usage

```
plotCohortTiming(
  result,
  plotType = "boxplot",
  facetVarX = "variable_name",
  facetVarY = "group_level",
  colorVars = "group_level",
  timingLabel = "{cohort_name_reference} &&& {cohort_name_comparator}",
  uniqueCombinations = TRUE
)
```

Arguments

result A summariseCohortTiming result.

plotType Type of desired formatted table, possibilities are "boxplot" and "density".

facetVarX column in data to facet by on horizontal axis

facetVarY column in data to facet by on vertical axis

colorVars Column names to distinct by colors. default set to group_level

timingLabel A glue expression to identify each plotted cohort overlap.

uniqueCombinations If TRUE, only unique combinations of reference and comparator plots will be plotted.

Value

A ggplot.

Examples

```
library(CohortCharacteristics)
cdm <- CohortCharacteristics::mockCohortCharacteristics()
timing <- summariseCohortTiming(cdm$cohort2)
plotCohortTiming(timing)
CDMConnector::cdmDisconnect(cdm = cdm)
```

plotLargeScaleCharacteristics

create a ggplot from the output of summariseLargeScaleCharacteristics.

Description

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

Usage

```
plotLargeScaleCharacteristics(
  data,
  position = "horizontal",
  splitStrata = FALSE,
  facet = NULL,
  colorVars = "variable_level"
)
```

Arguments

data	output of summariseLargeScaleCharacteristics().
position	if set to <i>horizontal</i> the horizontal axis will plot "variable_name" column and the vertical axis "estimate_value" column. If <i>vertical</i> , 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`.

Examples

```
library(CohortCharacteristics)

cdm <- 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$concept_id))
cdm <- CDMConnector::insertTable(cdm, "concept", concept)
results <- cdm$cohort2 |>
  summariseLargeScaleCharacteristics(
    episodeInWindow = c("condition_occurrence"),
    minimumFrequency = 0
  )

graphs <- plotLargeScaleCharacteristics(results)
CDMConnector::cdmDisconnect(cdm = cdm)
```

`summariseCharacteristics`

Summarise characteristics of individuals

Description

Summarise characteristics of individuals

Usage

```
summariseCharacteristics(
  cohort,
```

```

strata = list(),
demographics = TRUE,
ageGroup = NULL,
tableIntersect = list(),
cohortIntersect = list(),
conceptIntersect = list(),
otherVariables = character()
)

```

Arguments

cohort	A cohort in the cdm.
strata	Stratification list.
demographics	Whether to summarise demographics data.
ageGroup	A list of age groups.
tableIntersect	A list of arguments that uses addTableIntersect function to add variables to summarise.
cohortIntersect	A list of arguments that uses addCohortIntersect function to add variables to summarise.
conceptIntersect	A list of arguments that uses addConceptIntersect function to add variables to summarise.
otherVariables	Other variables contained in cohort that you want to be summarised.

Value

A summary of the characteristics of the individuals.

Examples

```

library(CohortCharacteristics)

cdm <- mockCohortCharacteristics()

summariseCharacteristics(
  cohort = cdm$cohort1,
  ageGroup = list(c(0, 19), c(20, 39), c(40, 59), c(60, 79), c(80, 150)),
  tableIntersect = list(
    "Number visits prior year" = list(
      tableName = "visit_occurrence", value = "count", window = c(-365, -1)
    )
  ),
  cohortIntersect = list(
    "Drugs prior year" = list(
      targetCohortTable = "cohort2", value = "flag", window = c(-365, -1)
    ),
    "Conditions any time prior" = list(

```

```
        targetCohortTable = "cohort2", value = "flag", window = c(-Inf, -1)
      )
    )
  )
  CDMConnector::cdmDisconnect(cdm = cdm)
```

summariseCohortCounts *Summarise counts for each different cohort. You can add a list of stratifications.*

Description

Summarise counts for each different cohort. You can add a list of stratifications.

Usage

```
summariseCohortCounts(cohort, strata = list())
```

Arguments

cohort	A cohort in the cdm.
strata	Stratification list.

Value

A summary of the number of individuals in each cohort and strata.

Examples

```
cdm <- mockCohortCharacteristics()

cdm$cohort1 |>
  PatientProfiles::addSex() |>
  summariseCohortCounts(strata = "sex")
```

`summariseCohortIntersect`*Summarise cohort intersection information*

Description

Summarise cohort intersection information

Usage

```
summariseCohortIntersect(cohort, cohortIntersect, strata = list())
```

Arguments

<code>cohort</code>	A cohort in the cdm.
<code>cohortIntersect</code>	The settings for cohort intersection settings.
<code>strata</code>	Stratification list.

Value

A summary of the cohort intersection informations.

Examples

```
library(CohortCharacteristics)

cdm <- mockCohortCharacteristics()

summariseCohortIntersect(
  cohort = cdm$cohort1,
  cohortIntersect = list(
    "Medications in the prior year" = list(
      targetCohortTable = "cohort2", value = "flag", window = c(-365, -1)
    )
  )
)
CDMConnector::cdmDisconnect(cdm = cdm)
```

`summariseCohortOverlap`*Summarise cohort overlap*

Description

Summarise cohort overlap

Usage

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

Arguments

<code>cohort</code>	A cohort table in a cdm reference.
<code>cohortId</code>	Vector of cohort definition ids to include, if NULL, all cohort definition ids will be used.
<code>strata</code>	List of the stratifications within each group to be considered. Must be column names in the cohort table provided.

Value

A summarised result.

Examples

```
library(CohortCharacteristics)
cdm <- CohortCharacteristics::mockCohortCharacteristics()
results <- summariseCohortOverlap(cdm$cohort2)
CDMConnector::cdmDisconnect(cdm = cdm)
```

`summariseCohortTiming` *Summarise cohort timing*

Description

Summarise cohort timing

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	Vector of cohort definition ids to include, if NULL, all cohort definition ids will be used.
strata	List of the stratifications within each group to be considered. Must be column names in the cohort table provided.
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 for timing.
density	Get data for density plot.

Value

A summarised result.

Examples

```
library(CohortCharacteristics)  
cdm <- CohortCharacteristics::mockCohortCharacteristics()  
results <- summariseCohortTiming(cdm$cohort2)  
CDMConnector::cdmDisconnect(cdm = cdm)
```

summariseConceptIntersect

Summarise concept intersect with a cohort_table

Description

Summarise concept intersect with a cohort_table

Usage

```
summariseConceptIntersect(cohort, conceptIntersect, strata = list())
```

Arguments

cohort	A cohort in the cdm
conceptIntersect	A list of arguments that uses addConceptIntersect function to add variables to summarise.
strata	Stratification list

Value

A summary of the concept intersect of the individuals

```
summariseLargeScaleCharacteristics
```

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

Description

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

Usage

```
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)
)
```

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.
sensorDate	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

```
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$concept_id))
cdm <- CDMConnector::insertTable(cdm, "concept", concept)
results <- cdm$cohort2 |>
  summariseLargeScaleCharacteristics(
    episodeInWindow = c("condition_occurrence"),
    minimumFrequency = 0
  )
CDMConnector::cdmDisconnect(cdm = cdm)
```

summariseTableIntersect

Summarise table intersection information

Description

Summarise table intersection information

Usage

```
summariseTableIntersect(cohort, tableIntersect = list(), strata = list())
```

Arguments

cohort	A cohort in the cdm.
tableIntersect	A list of arguments that uses addTableIntersect function to add variables to summarise.
strata	Stratification list.

Value

A summary of the table intersections.

`tableCharacteristics` *Format a summarised_characteristics object into a visual table.*

Description

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

Usage

```
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,
  minCellCount = 5,
  excludeColumns = c("result_id", "result_type", "package_name", "package_version",
    "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.
minCellCount	Counts below which results will be clouded.
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

```
library(CohortCharacteristics)

cdm <- mockCohortCharacteristics()

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

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

tableCohortOverlap *Format a summariseOverlapCohort result into a visual table.*

Description

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

Usage

```
tableCohortOverlap(
  result,
  type = "gt",
  formatEstimateName = c(`N (%)` = "<count> (<percentage>%)"),
  header = c("strata"),
  split = c("group", "strata", "additional"),
  groupColumn = NULL,
  minCellCount = 5,
```

```

  excludeColumns = c("result_id", "result_type", "package_name", "package_version",
    "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'.
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.
minCellCount	Counts below which results will be clouded.
excludeColumns	Columns to drop from the output table.
.options	Named list with additional formatting options. CohortCharacteristics::optionsTableCohortOverlap() shows allowed arguments and their default values.

Value

A formatted table of the summariseOverlapCohort result.

Examples

```

library(CohortCharacteristics)
cdm <- CohortCharacteristics::mockCohortCharacteristics()
overlap <- summariseCohortOverlap(cdm$cohort2)
tableCohortOverlap(overlap)
CDMConnector::cdmDisconnect(cdm = cdm)

```

tableCohortTiming	<i>Format a summariseCohortTiming result into a visual table.</i>
-------------------	---

Description

‘r lifecycle::badge("experimental")‘

Usage

```
tableCohortTiming(
  result,
  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,
  minCellCount = 5,
  excludeColumns = c("result_id", "result_type", "package_name", "package_version",
    "estimate_type", "variable_level"),
  .options = list()
)
```

Arguments

result	A summariseCohortTiming 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'.
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.
minCellCount	Counts below which results will be clouded.
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.

Examples

```
library(CohortCharacteristics)
cdm <- CohortCharacteristics::mockCohortCharacteristics()
timing <- summariseCohortTiming(cdm$cohort2)
tableCohortTiming(timing)
CDMConnector::cdmDisconnect(cdm = cdm)
```

```
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 = 10,
  minCellCount = 5
)
```

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.
minCellCount	Minimum number of counts to display.

Value

A formatted table.

Examples

```
## 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`, 5
`optionsTableCohortOverlap`, 6
`optionsTableCohortTiming`, 6

`plotCharacteristics`, 7
`plotCohortOverlap`, 8
`plotCohortTiming`, 9
`plotLargeScaleCharacteristics`, 10

`summariseCharacteristics`, 11
`summariseCohortCounts`, 13
`summariseCohortIntersect`, 14
`summariseCohortOverlap`, 15
`summariseCohortTiming`, 15
`summariseConceptIntersect`, 16
`summariseLargeScaleCharacteristics`, 17
`summariseTableIntersect`, 18

`tableCharacteristics`, 19
`tableCohortOverlap`, 20
`tableCohortTiming`, 21
`tableLargeScaleCharacteristics`, 23