Package 'MeasurementDiagnostics'

February 17, 2026

Type Package

Title Diagnostics for Lists of Codes Based on Measurements

Version 0.3.0

Description Diagnostics of list of codes based on concepts from the domains measurement and observation. This package works for data mapped to the Observational Medical Outcomes Partnership Common Data Model.

Imports cli, clock, CohortCharacteristics, CohortConstructor (>= 0.4.0), DBI, dplyr, glue, omopgenerics (>= 1.2.0), PatientProfiles (>= 1.4.0), purrr, rlang, stringr, tidyr

Suggests CDMConnector (>= 2.0.0), CodelistGenerator (>= 3.5.0), visOmopResults (>= 1.4.0), duckdb, knitr, omock (>= 0.4.0), rmarkdown, testthat, ggplot2, gt, flextable, RPostgres, lubridate, odbc

Depends R (>= 4.1)

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.3

URL https://ohdsi.github.io/MeasurementDiagnostics/

BugReports https://github.com/ohdsi/MeasurementDiagnostics/issues

VignetteBuilder knitr

NeedsCompilation no

Author Edward Burn [aut] (ORCID: <https://orcid.org/0000-0002-9286-1128>), Nuria Mercade-Besora [aut, cre] (ORCID: <https://orcid.org/0009-0006-7948-3747>), Marta Alcalde-Herraiz [aut] (ORCID: <https://orcid.org/0009-0002-4405-1814>)

Maintainer Nuria Mercade-Besora <nuria.mercadebesora@ndorms.ox.ac.uk>

Repository CRAN

Date/Publication 2026-02-17 16:40:02 UTC

Contents

`mockMeasurementDiagnostics`

Function to create a mock cdm reference.

Description

Creates an example dataset that can be used to show how the package works

Usage

```
mockMeasurementDiagnostics(
  nPerson = 100,
  con = DBI::dbConnect(duckdb::duckdb()),
  writeSchema = "main",
  seed = 111
)
```

Arguments

`nPerson`	number of people in the cdm.
`con`	A DBI connection to create the cdm mock object.
`writeSchema`	Name of an schema on the same connection with writing permissions.
`seed`	seed to use when creating the mock data.

Value

cdm object

Examples

```
library(MeasurementDiagnostics)
cdm <- mockMeasurementDiagnostics()
cdm
```

plotMeasurementSummary *Plot summariseMeasurementTiming results.*

Description

Plot summariseMeasurementTiming results.

Usage

```
plotMeasurementSummary(
  result,
  x = "codelist_name",
  y = "days_between_measurements",
  plotType = "boxplot",
  facet = visOmopResults::strataColumns(result),
  colour = c("codelist_name"),
  style = NULL
)
```

Arguments

result	A summarised_result object.
x	Variable to plot on the x axis when plotType is "boxlot" or "barplot".
y	Variable to plot, it can be "days_between_measurements" or "measurements_per_subject".
plotType	Type of plot, either "boxplot", "barplot", or "densityplot".
facet	Columns to facet by. See options with ‘visOmopResults::plotColumns(result)‘. Formula input is also allowed to specify rows and columns.
colour	Columns to color by. See options with ‘visOmopResults::plotColumns(result)‘.
style	Pre-defined style to apply: "default" or "darwin" - the latter just for gt and flextable. If NULL the "default" style is used.

Value

A ggplot.

Examples

```
library(MeasurementDiagnostics)
library(dplyr)

cdm <- mockMeasurementDiagnostics()

result <- summariseMeasurementUse(
  cdm = cdm,
  codes = list("test_codelist" = c(3001467L, 45875977L))
```

```
)

result |>
  filter(variable_name == "days_between_measurements") |>
  plotMeasurementSummary()

CDMConnector::cdmDisconnect(cdm)
```

`plotMeasurementValueAsConcept` *Plot summariseMeasurementTiming results.*

Description

Plot summariseMeasurementTiming results.

Usage

```
plotMeasurementValueAsConcept(
  result,
  x = "count",
  y = "codelist_name",
  facet = c("cdm_name"),
 colour = c("concept_name", "variable_level", visOmopResults::strataColumns(result)),
  style = NULL
)
```

Arguments

`result`	A summarised_result object.
`x`	Columns to use as horizontal axes. See options with 'visOmopResults::plotColumns(result)'.
`y`	Columns to use as horizontal axes. See options with 'visOmopResults::plotColumns(result)'.
`facet`	Columns to facet by. See options with 'visOmopResults::plotColumns(result)'. Formula input is also allowed to specify rows and columns.
`colour`	Columns to color by. See options with 'visOmopResults::plotColumns(result)'.
`style`	Pre-defined style to apply: "default" or "darwin" - the latter just for gt and flextable. If NULL the "default" style is used.

Value

A ggplot.

Examples

```
library(MeasurementDiagnostics)

cdm <- mockMeasurementDiagnostics()

result <- summariseMeasurementUse(
  cdm = cdm,
  bySex = TRUE,
  codes = list("test_codelist" = c(3001467L, 45875977L))
)

plotMeasurementValueAsConcept(result)

CDMConnector::cdmDisconnect(cdm)
```

`plotMeasurementValueAsNumber` *Plot summariseMeasurementTiming results.*

Description

Plot summariseMeasurementTiming results.

Usage

```
plotMeasurementValueAsNumber(
  result,
  x = "unit_concept_name",
  plotType = "boxplot",
  facet = c("codelist_name", "concept_name"),
 colour = c("cdm_name", "unit_concept_name", visOmopResults::strataColumns(result)),
  style = NULL
)
```

Arguments

`result`	A summarised_result object.
`x`	Variable to plot on the x axis when plotType is "boxlot" or "barplot".
`plotType`	Type of plot, either "boxplot", "barplot", or "densityplot".
`facet`	Columns to facet by. See options with ‘visOmopResults::plotColumns(result)‘. Formula input is also allowed to specify rows and columns.
`colour`	Columns to color by. See options with ‘visOmopResults::plotColumns(result)‘.
`style`	Pre-defined style to apply: "default" or "darwin" - the latter just for gt and flextable. If NULL the "default" style is used.

Value

A ggplot.

Examples

```
library(MeasurementDiagnostics)

cdm <- mockMeasurementDiagnostics()

result <- summariseMeasurementUse(
  cdm = cdm,
  bySex = TRUE,
  codes = list("test_codelist" = c(3001467L, 45875977L))
)

plotMeasurementValueAsNumber(result)

CDMConnector::cdmDisconnect(cdm)
```

---

summariseCohortMeasurementUse

Diagnostics of a codelist of measurement codes within a cohort

---

Description

Diagnostics of a codelist of measurement codes within a cohort

Usage

```
summariseCohortMeasurementUse(
  cohort,
  codes = NULL,
  timing = "during",
  byConcept = TRUE,
  byYear = FALSE,
  bySex = FALSE,
  ageGroup = NULL,
  dateRange = as.Date(c(NA, NA)),
  estimates = list(measurement_summary = c("min", "q25", "median", "q75", "max",
   "density"), measurement_value_as_number = c("min", "q01", "q05", "q25", "median",
   "q75", "q95", "q99", "max", "count_missing", "percentage_missing", "density"),
    measurement_value_as_concept = c("count", "percentage")),
  histogram = NULL,
  checks = c("measurement_summary", "measurement_value_as_number",
    "measurement_value_as_concept")
)
```

Arguments

cohort	A cohort in which to perform the diagnostics of the measurement codes provided.
codes	A codelist of measurement/observation codes for which to perform diagnostics. If NULL it uses the codelist used to create each of the cohorts.
timing	Three options: 1) "any" if the interest is on measurement recorded any time, 2) "during", if interested in measurements while the subject is in the cohort, and 3) "cohort_start_date" for measurements occurring at cohort start date.
byConcept	TRUE or FALSE. If TRUE code use will be summarised by concept.
byYear	TRUE or FALSE. If TRUE code use will be summarised by year.
bySex	TRUE or FALSE. If TRUE code use will be summarised by sex.
ageGroup	If not NULL, a list of ageGroup vectors of length two.
dateRange	Two dates. The first indicating the earliest measurement date and the second indicating the latest possible measurement date.
estimates	A named list indicating, for each measurement diagnostics check, which estimates to retrieve. The names of the list should correspond to the diagnostics checks, and each list element should be a character vector specifying the estimates to compute. Allowed estimates are those supported by the ‘summariseResult()‘ function in the **PatientProfiles** package. If omitted, all available estimates for each check will be returned.
histogram	Named list where names point to checks for which to get estimates for a histogram, and elements are numeric vectors indicating the bind-width. See function examples. Histogram only available for "measurement_summary" and "measurement_value_as_number".
checks	Diagnostics to run. Options are: "measurement_summary", "measurement_value_as_number", and "measurement_value_as_concept".

Value

A summarised result

Examples

```
library(MeasurementDiagnostics)

cdm <- mockMeasurementDiagnostics()

result <- summariseCohortMeasurementUse(
  codes = list("test_codelist" = c(3001467L, 45875977L)),
  cohort = cdm$my_cohort, timing = "cohort_start_date"
)

# Histogram
result <- summariseCohortMeasurementUse(
  codes = list("test_codelist" = c(3001467L, 45875977L)),
```

```
  cohort = cdm$my_cohort, timing = "cohort_start_date",
  histogram = list(
    "days_between_measurements" = list(
      '0 to 100' = c(0, 100), '110 to 200' = c(110, 200),
      '210 to 300' = c(210, 300), '310 to Inf' = c(310, Inf)
    ),
    "measurements_per_subject" = list(
      '0 to 10' = c(0, 10), '11 to 20' = c(11, 20), '21 to 30' = c(21, 30),
      '31 to Inf' = c(31, Inf)
    ),
    "value_as_number" =  list(
      '0 to 5' = c(0, 5), '6 to 10' = c(6, 10), '11 to 15' = c(11, 15),
      '>15' = c(16, Inf)
    )
  )
)

# Different age groups
result <- summariseCohortMeasurementUse(
  codes = list("test_codelist" = c(3001467L, 45875977L)),
  cohort = cdm$my_cohort,
  ageGroup = list(
    "age_group_1" = list(c(0, 17), c(18, 64), c(65, 150)),
   "age_group_2" = list(c(0, 19), c(20, 39), c(40, 59), c(60, 79), c(80, 99), c(100, 120))
  )
)

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

---

summariseMeasurementUse *Diagnostics of a codelist of measurement codes in the database*

Description

Diagnostics of a codelist of measurement codes in the database

Usage

```
summariseMeasurementUse(
  cdm,
  codes,
  byConcept = TRUE,
  byYear = FALSE,
  bySex = FALSE,
  ageGroup = NULL,
  dateRange = as.Date(c(NA, NA)),
  personSample = 20000,
```

```
  estimates = list(measurement_summary = c("min", "q25", "median", "q75", "max",
   "density"), measurement_value_as_number = c("min", "q01", "q05", "q25", "median",
   "q75", "q95", "q99", "max", "count_missing", "percentage_missing", "density"),
    measurement_value_as_concept = c("count", "percentage")),
  histogram = NULL,
  checks = c("measurement_summary", "measurement_value_as_number",
    "measurement_value_as_concept")
)
```

Arguments

cdm	A reference to the cdm object.
codes	A codelist of measurement/observation codes for which to perform diagnostics.
byConcept	TRUE or FALSE. If TRUE code use will be summarised by concept.
byYear	TRUE or FALSE. If TRUE code use will be summarised by year.
bySex	TRUE or FALSE. If TRUE code use will be summarised by sex.
ageGroup	If not NULL, a list of ageGroup vectors of length two.
dateRange	Two dates. The first indicating the earliest measurement date and the second indicating the latest possible measurement date.
personSample	Integerish or 'NULL'. Number of persons to sample the measurement and observation tables. If 'NULL', no sampling is performed.
estimates	A named list indicating, for each measurement diagnostics check, which estimates to retrieve. The names of the list should correspond to the diagnostics checks, and each list element should be a character vector specifying the estimates to compute. Allowed estimates are those supported by the 'summariseResult()' function in the **PatientProfiles** package. If omitted, all available estimates for each check will be returned.
histogram	Named list where names point to checks for which to get estimates for a histogram, and elements are numeric vectors indicating the bind-width. See function examples. Histogram only available for "measurement_summary" and "measurement_value_as_number".
checks	Diagnostics to run. Options are: "measurement_summary", "measurement_value_as_number", and "measurement_value_as_concept".

Value

A summarised result

Examples

```
library(MeasurementDiagnostics)

cdm <- mockMeasurementDiagnostics()

result <- summariseMeasurementUse(
```

```
  cdm = cdm, codes = list("test_codelist" = c(3001467L, 45875977L))
)

resultHistogram <- summariseMeasurementUse(
  cdm = cdm,
  codes = list("test_codelist" = c(3001467L, 45875977L)),
  byConcept = TRUE,
  byYear = FALSE,
  bySex = FALSE,
  ageGroup = NULL,
  dateRange = as.Date(c(NA, NA)),
  estimates = list(
    "measurement_summary" = c("min", "q25", "median", "q75", "max", "density"),
    "measurement_value_as_number" = c(
      "min", "q01", "q05", "q25", "median", "q75", "q95", "q99", "max",
      "count_missing", "percentage_missing", "density"
    ),
    "measurement_value_as_concept" = c("count", "percentage")
  ),
  histogram = list(
    "days_between_measurements" = list(
      '0 to 100' = c(0, 100), '110 to 200' = c(110, 200),
      '210 to 300' = c(210, 300), '310 to Inf' = c(310, Inf)
    ),
    "measurements_per_subject" = list(
      '0 to 10' = c(0, 10), '11 to 20' = c(11, 20),
      '21 to 30' = c(21, 30), '31 to Inf' = c(31, Inf)
    ),
    "value_as_number" =  list(
      '0 to 5' = c(0, 5), '6 to 10' = c(6, 10),
      '11 to 15' = c(11, 15), '>15' = c(16, Inf)
    )
  ),
 checks = c("measurement_summary", "measurement_value_as_number", "measurement_value_as_concept")
)

# more than one age group:
result <- summariseMeasurementUse(
  cdm = cdm,
  codes = list("test_codelist" = c(3001467L, 45875977L)),
  ageGroup = list(
    "age_group_1" = list(c(0, 17), c(18, 64), c(65, 150)),
   "age_group_2" = list(c(0, 19), c(20, 39), c(40, 59), c(60, 79), c(80, 99), c(100, 120))
  )
)

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

`tableMeasurementSummary` *Format a measurement_summary object into a visual table*

Description

Format a measurement_summary object into a visual table

Usage

```
tableMeasurementSummary(
  result,
  header = c(visOmopResults::strataColumns(result)),
  groupColumn = character(),
  settingsColumn = character(),
  hide = c("variable_level"),
  style = NULL,
  type = NULL,
  .options = list()
)
```

Arguments

`result`	A summarised_result object.
`header`	Columns to use as header. See options with 'visOmopResults::tableColumns(result)'.
`groupColumn`	Columns to group by. See options with 'visOmopResults::tableColumns(result)'.
`settingsColumn`	Columns from settings to include in results. See options with 'visOmopResults::settingsColumns(result)'.
`hide`	Columns to hide from the visualisation. See options with 'visOmopResults::tableColumns(result)'.
`style`	Named list that specifies how to style the different parts of the table generated. It can either be a pre-defined style ("default" or "darwin" - the latter just for gt and flextable), or NULL which converts to "default" style, or custom code.
`type`	Type of table. Check supported types with 'visOmopResults::tableType()'. If NULL 'gt' type will be used.
`.options`	A named list with additional formatting options. 'visOmopResults::tableOptions()' shows allowed arguments and their default values.

Value

A formatted table

Examples

```
library(MeasurementDiagnostics)

cdm <- mockMeasurementDiagnostics()
```

```
result <- summariseMeasurementUse(
  cdm = cdm,
  codes = list("test_codelist" = c(3001467L, 45875977L))
)

tableMeasurementSummary(result)

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

`tableMeasurementValueAsConcept` *Format a measurement_summary object into a visual table*

Description

Format a measurement_summary object into a visual table

Usage

```
tableMeasurementValueAsConcept(
  result,
  header = c(visOmopResults::strataColumns(result)),
  groupColumn = c("codelist_name"),
  settingsColumn = character(),
  hide = character(),
  style = NULL,
  type = NULL,
  .options = list()
)
```

Arguments

`result`	A summarised_result object.
`header`	Columns to use as header. See options with 'visOmopResults::tableColumns(result)'.
`groupColumn`	Columns to group by. See options with 'visOmopResults::tableColumns(result)'.
`settingsColumn`	Columns from settings to include in results. See options with 'visOmopResults::settingsColumns(result)'.
`hide`	Columns to hide from the visualisation. See options with 'visOmopResults::tableColumns(result)'.
`style`	Named list that specifies how to style the different parts of the table generated. It can either be a pre-defined style ("default" or "darwin" - the latter just for gt and flextable), or NULL which converts to "default" style, or custom code.
`type`	Type of table. Check supported types with 'visOmopResults::tableType()'. If NULL 'gt' type will be used.
`.options`	A named list with additional formatting options. 'visOmopResults::tableOptions()' shows allowed arguments and their default values.

Value

A formatted table

Examples

```
library(MeasurementDiagnostics)

cdm <- mockMeasurementDiagnostics()

result <- summariseMeasurementUse(
  cdm = cdm,
  codes = list("test_codelist" = c(3001467L, 45875977L))
)

tableMeasurementValueAsConcept(result)

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

---

tableMeasurementValueAsNumber

Format a measurement_summary object into a visual table

Description

Format a measurement_summary object into a visual table

Usage

```
tableMeasurementValueAsNumber(
  result,
  header = c(visOmopResults::strataColumns(result)),
  groupColumn = c("codelist_name"),
  settingsColumn = character(),
  hide = c("variable_level"),
  style = NULL,
  type = NULL,
  .options = list()
)
```

Arguments

result	A summarised_result object.
header	Columns to use as header. See options with 'visOmopResults::tableColumns(result)'.
groupColumn	Columns to group by. See options with 'visOmopResults::tableColumns(result)'.
settingsColumn	Columns from settings to include in results. See options with 'visOmopResults::settingsColumns(result)'.

hide	Columns to hide from the visualisation. See options with 'visOmopResults::tableColumns(result)'.
style	Named list that specifies how to style the different parts of the table generated. It can either be a pre-defined style ("default" or "darwin" - the latter just for gt and flextable), or NULL which converts to "default" style, or custom code.
type	Type of table. Check supported types with 'visOmopResults::tableType()'. If NULL 'gt' type will be used.
.options	A named list with additional formatting options. 'visOmopResults::tableOptions()' shows allowed arguments and their default values.

Value

A formatted table

Examples

```
library(MeasurementDiagnostics)

cdm <- mockMeasurementDiagnostics()

result <- summariseMeasurementUse(
  cdm = cdm,
  codes = list("test_codelist" = c(3001467L, 45875977L))
)

tableMeasurementValueAsNumber(result)

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

Index