Package: CohortIncidence (via r-universe)

October 21, 2026

Type Package

Title Cohort Incidence Analysis for the OMOP Common Data Model

Version 4.2.0

Date 2026-07-27

Description Provides functionality for performing cohort incidence analyses against data stored in the OMOP Common Data Model (CDM). The package generates database-specific SQL, executes analyses across supported database platforms, and returns standardized incidence estimates using configurable time-at-risk, stratification, and cohort definitions. It is intended to support reproducible observational research within the OHDSI analytics framework.

License Apache License 2.0

VignetteBuilder knitr

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

BugReports https://github.com/OHDSI/CohortIncidence/issues

Depends R (>= 3.2.2), DatabaseConnector (>= 3.0.0)

Imports rlang, rJava(>= 0.9-10), checkmate, jsonlite, SqlRender (>= 1.6.0)

SystemRequirements Java (>= 8)

NeedsCompilation yes

Encoding UTF-8

Suggests testthat (>= 3.0.0), knitr, rmarkdown, withr, readr, Eunomia, duckdb, R6

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Author Christopher Knoll [aut, cre]

Maintainer Christopher Knoll <cknoll@ohdsi.org>

Config/pak/sysreqs default-jdk

Repository https://cran.r-universe.dev

Date/Publication 2026-08-25 14:10:02 UTC

RemoteUrl https://github.com/cran/CohortIncidence

RemoteRef HEAD

RemoteSha 8d94a1f459d80400f019194acb1cef15c1b53368

Contents

buildOptions . . . 2
buildQuery . . . 3
CohortDefinition . . . 4
CohortReference . . . 5
CohortSubgroup . . . 6
createCohortRef . . . 8
createCohortSubgroup . . . 8
createDateRange . . . 9
createIncidenceAnalysis . . . 9
createIncidenceDesign . . . 10
createOutcomeDef . . . 11
createStrataSettings . . . 11
createTimeAtRiskDef . . . 12
DateRange . . . 13
executeAnalysis . . . 14
getCleanupSql . . . 16
getResultsDdl . . . 17
IncidenceAnalysis . . . 17
IncidenceDesign . . . 19
Outcome . . . 20
StrataSettings . . . 22
TimeAtRisk . . . 23

Index **25**

`buildOptions` *Builds the BuilderOptions jObject with the specified paramaters*

Description

Builds the BuilderOptions jObject with the specified paramaters

Usage

```
buildOptions(
  cohortTable,
  outcomeCohortTable = cohortTable,
  subgroupCohortTable = cohortTable,
  sourceName,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  vocabularySchema = cdmDatabaseSchema,
  useTempTables = FALSE,
  refId
)
```

Arguments

cohortTable	The name of table with cohorts
outcomeCohortTable	The name of table with outcome cohorts, defaults to cohortTable param.
subgroupCohortTable	The name of table with subgroup cohorts, defaults to cohortTable param.
sourceName	A value to inject to the results table for the source name.
cdmDatabaseSchema	the name of schema containing data in CDM format
resultsDatabaseSchema	the name of schema where results would be placed
vocabularySchema	the name of schema with vocabulary tables, defaults to cdmDatabaseSchema param
useTempTables	use temp tables instead of a results schema.
refId	A number tagged to the results for retrieval purposes.

Value

a BuilderOptions object used in buildQuery.

---

buildQuery *Builds SQL code to run analyses according given Cohort Characterization design*

Description

Builds SQL code to run analyses according given Cohort Characterization design

Usage

```
buildQuery(incidenceDesign, buildOptions)
```

Arguments

`incidenceDesign`	A string object containing valid JSON that represents cohort incidence design
`buildOptions`	the parameters to use in building the analysis queries, created by buildOptions()

Value

SQL code in MS Sql Server dialect, if it's required to run analysis on another DBMS you have to use `translateSql` function in the SqlRender package.

`CohortDefinition` *R6 Class Representing a CohortDefinition*

Description

The CohortDefinition class, encapsulating the id, name and expression of a cohort definition.

Details

This R6 class is intended to wrap the Cohort Defintion expression used to generate the cohort, and provide the id and name attribute for this cohort.

Note, when serializing with a library such as jsonlite, first call toList() on the R6 class before calling jsonlite::toJSON().

Value

A `CohortDefinition` R6 object used to hold optional cohort metadata and a serialized cohort definition expression. This object is primarily metadata for the design and is not required for analysis generation.

Public fields

`id` The cohort ID

`name` The cohort name

`expression` The cohort expression

Methods

Public methods:

- `CohortDefinition$toList()`
- `CohortDefinition$asJSON()`
- `CohortDefinition$clone()`

`CohortDefinition$toList()`: returns the R6 class elements as a list for use in jsonlite::toJSON()

Usage:

`CohortDefinition$toList()`

Returns: A named list with `id`, `name`, and `expression`, where `expression` is itself a nested list representation of the cohort definition expression.

`CohortDefinition$asJSON()`: returns the JSON string for this R6 class

Usage:

`CohortDefinition$asJSON()`

Returns: A JSON string representing the `CohortDefinition` object.

`CohortDefinition$clone()`: The objects of this class are cloneable with this method.

Usage:

`CohortDefinition$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

`CohortReference` *R6 Class Representing a CohortReference*

Description

The CohortReference class, encapsulating the id, name and descritpion fields that refernces a cohort definition.

Details

This class is used to reference a cohort definition by ID, while providing a way to substitute a name for this specific reference.

Note, when serializing with a library such as jsonlite, first call toList() on the R6 class before calling jsonlite::toJSON().

Value

A `CohortReference` R6 object that stores the cohort id, display name, and optional description used to reference a target, outcome, or subgroup cohort.

Active bindings

`id` the cohort id being referenced

`name` the name for the cohort to be used in this reference.

`description` A description for this Cohort Reference.

Methods

Public methods:

- CohortReference$new()
- CohortReference$toList()
- CohortReference$asJSON()
- CohortReference$clone()

`CohortReference$new()`: creates a new instance, using the provided data param if provided.

Usage:

```
CohortReference$new(data = list())
```

Arguments:

`data` the data (as a json string or list) to initialize with

Returns: A new `CohortReference` object initialized from a list or JSON string.

`CohortReference$toList()`: returns the R6 class elements as a list for use in jsonlite::toJSON()

Usage:

```
CohortReference$toList()
```

Returns: A named list with `id`, `name`, and `description` fields.

`CohortReference$asJSON()`: returns the JSON string for this R6 class

Usage:

```
CohortReference$asJSON()
```

Returns: A JSON string representation of the `CohortReference` object.

`CohortReference$clone()`: The objects of this class are cloneable with this method.

Usage:

```
CohortReference$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`CohortSubgroup` *R6 Class Representing a Cohort Subgroup definition*

Description

The CohortSubgroup class, encapsulating the id, name, description, and CohortRef.

Details

This class is used to specify a cohort subgroup to be used in the analysis. A TAR will be considered part of the subgroup if the TAR starts between the subgroup's cohort start and cohort end.

Note, when serializing with a library such as jsonlite, first call toList() on the R6 class before calling jsonlite::toJSON().

Value

A CohortSubgroup R6 object containing subgroup metadata and the CohortReference used to determine subgroup membership.

Active bindings

id an integer uniquely identifying this subgroup

name The name to use for this cohort reference

description The description for this subgroup

cohort the cohort used to represent this subgroup. Must be class CohortReference

Methods

Public methods:

- CohortSubgroup$new()
- CohortSubgroup$toList()
- CohortSubgroup$asJSON()
- CohortSubgroup$clone()

CohortSubgroup$new(): creates a new instance, using the provided data param if provided. The JSON takes the form: {"id":1,"name":"some name","description":"some description","cohort":{"id":99, "na

Usage:

```
CohortSubgroup$new(data = list(CohortSubgroup = list()))
```

Arguments:

data the data (as a json string or list) to initialize with

Returns: A new CohortSubgroup object initialized from a list or JSON string.

CohortSubgroup$toList(): returns the R6 class elements as a list for use in jsonlite::toJSON()

Usage:

```
CohortSubgroup$toList()
```

Returns: A named list with a single CohortSubgroup element that contains id, name, description, and a nested cohort definition.

CohortSubgroup$asJSON(): returns the JSON string for this R6 class

Usage:

```
CohortSubgroup$asJSON()
```

Returns: A JSON string representation of the CohortSubgroup object.

CohortSubgroup$clone(): The objects of this class are cloneable with this method.

Usage:

```
CohortSubgroup$clone(deep = FALSE)
```

Arguments:

deep Whether to make a deep clone.

createCohortRef *Creates R6 object for CohortReference*

Description

Creates R6 object for CohortReference

Usage

```
createCohortRef(id, name, description)
```

Arguments

id	the cohort id that is being referenced
name	the name to use for this reference
description	an optional description to use for this reference.

Value

A CohortReference R6 object that stores the cohort identifier, display name, and optional description used to reference a cohort definition in the design.

createCohortSubgroup *Creates R6 object for CohortSubgroup*

Description

Creates R6 object for CohortSubgroup

Usage

```
createCohortSubgroup(id, name, description, cohortRef)
```

Arguments

id	the unique identifier for this subgroup
name	The subgroup name
description	The subgroup description
cohortRef	A cohort reference, as an R6 Class CohortReference

Value

A CohortSubgroup R6 object containing subgroup metadata and the CohortReference used to determine subgroup membership.

createDateRange *Creates R6 object for DateRange*

Description

Creates R6 object for DateRange

Usage

```
createDateRange(startDate, endDate)
```

Arguments

startDate	a character vector representing a date in YYYY-MM-DD format
endDate	a character vector representing a date in YYYY-MM-DD format

Value

A DateRange R6 object containing the start and end date strings in YYYY-MM-DD format.

createIncidenceAnalysis *Creates R6 object for IncidenceAnalysis*

Description

Creates R6 object for IncidenceAnalysis

Usage

```
createIncidenceAnalysis(targets, outcomes, tars)
```

Arguments

targets	A list or vector of target IDs from target definitions.
outcomes	A list or vector of outcome IDs from outcome definitions.
tars	A list or vector of TAR IDs from time-at-risk definitions.

Value

An IncidenceAnalysis R6 object containing the target, outcome, and TAR identifier vectors used to define one analysis combination.

`createIncidenceDesign` *Creates R6 object for IncidenceDesign*

Description

Creates R6 object for IncidenceDesign

Usage

```
createIncidenceDesign(
  cohortDefs,
  targetDefs,
  outcomeDefs,
  tars,
  analysisList,
  conceptSets,
  subgroups,
  strataSettings,
  studyWindow,
  firstAtRisk,
  firstPostOutcome
)
```

Arguments

`cohortDefs`	The set of cohort definitions. Optional metadata used when building a design, not required for analysis execution.
`targetDefs`	A list of target definitions, each element must be class CohortReference.
`outcomeDefs`	A list of outcome definitions, each element must be class Outcome
`tars`	A list of TAR definitions, each element must be class TimeAtRisk
`analysisList`	A list of analysis definitions, each element must be class IncidenceAnalysis
`conceptSets`	A list of concept sets, currently unused.
`subgroups`	A list of cohort subgroups, each element must be class Subgroup.
`strataSettings`	The strata settings used in the anlaysis, must be class StrataSettings.
`studyWindow`	Limits time at risk to the specified study window. Must be class DateRange.
`firstAtRisk`	Limits time at risk to the first per person. Must be boolean.
`firstPostOutcome`	Limits outcomes to first outcome within clean window of the first TAR or after. Must be boolean.

Value

An `IncidenceDesign` R6 object containing the design metadata, cohort definitions, target definitions, outcome definitions, time-at-risk definitions, analyses, subgroups, strata settings, and study window used to generate SQL. The object can be serialized to JSON for analysis execution.

createOutcomeDef *Creates R6 object for Outcome*

Description

Creates R6 object for Outcome

Usage

```
createOutcomeDef(id, name, cohortId = 0, cleanWindow = 0, excludeCohortId)
```

Arguments

id	the unique identifier for this outcome definition
name	an optional name for this outcome definition
cohortId	the cohort id reference for this outcome
cleanWindow	the number of days to extend the outcome cohort's end date. See executeAnalysis() for details on how this is applied.
excludeCohortId	a cohort ID from the outcomeCohrotTable that is used to exclude time at risk. See executeAnalysis() for details on how this is applied.

Value

An Outcome R6 object describing one outcome definition, including the outcome id, cohort id, clean window, and optional exclusion cohort id.

createStrataSettings *Creates R6 object for StrataSettings*

Description

Creates R6 object for StrataSettings

Usage

```
createStrataSettings(
  byAge = FALSE,
  byGender = FALSE,
  byYear = FALSE,
  ageBreaks,
  ageBreakList
)
```

Arguments

byAge	a boolean indicating to stratify by age, defaults to FALSE
byGender	a boolean indicating to stratify by gender, defaults to FALSE
byYear	a boolean indicating to stratify by year, defaults to FALSE
ageBreaks	a vector of integers indicating the age group bounds.
ageBreakList	a list of ageBreaks, used to specify multiple age break strata.

Value

A StrataSettings R6 object containing the three stratification flags plus optional age break definitions. The serialized form matches org.ohdsi.cohortincidence.design.StratifySettings JSON.

---

createTimeAtRiskDef *Creates R6 object for TimeAtRisk*

Description

Creates R6 object for TimeAtRisk

Usage

```
createTimeAtRiskDef(
  id,
  startWith = "start",
  startOffset = 0,
  endWith = "end",
  endOffset = 0
)
```

Arguments

id	the unique identifier for this outcome definition
startWith	Specifies the field (start or end) to offset from for the TAR start. Allowed values: 'start', 'end'
startOffset	The number of days to offset for the TAR start, defaults to 0.
endWith	Specifies the field (start or end) to offset from for the TAR end. Allowed values: 'start', 'end'
endOffset	The number of days to offset for the TAR start, defaults to 0.

Value

A TimeAtRisk R6 object describing the TAR start and end anchors plus their offsets. The serialized form contains a start and end field, each with a dateField and offset.

DateRange *R6 Class Representing a DataRange*

Description

The DateRange class, encapsulating the startDate and endDate of a date range.

Details

This class is used to specify a DateRange, with start and end dates specified as strings formatted as YYYY-MM-DD.

Value

A `DateRange` R6 object storing `startDate` and `endDate` as `YYYY-MM-DD` strings.

Active bindings

`startDate` a character with format YYYY-MM-DD to be used as the date range's start date.

`endDate` a character with format YYYY-MM-DD to be used as the date range's end date.

Methods

Public methods:

- `DateRange$new()`
- `DateRange$toList()`
- `DateRange$asJSON()`
- `DateRange$clone()`

`DateRange$new()`: creates a new instance, using the provided data param if provided.

Usage:

```
DateRange$new(data = list())
```

Arguments:

`data` the data (as a json string or list) to initialize with

Returns: A new `DateRange` object initialized from a list or JSON string.

`DateRange$toList()`: returns the R6 class elements as a list for use in jsonlite::toJSON()

Usage:

```
DateRange$toList()
```

Returns: A named list with `startDate` and `endDate` fields.

`DateRange$asJSON()`: returns the JSON string for this R6 class

Usage:

```
DateRange$asJSON()
```

Returns: A JSON string representation of the DateRange object.

`DateRange$clone()`: The objects of this class are cloneable with this method.

Usage:

```
DateRange$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`executeAnalysis` *Executes IR analysis given a design, options, and connection settings.*

Description

Executes IR analysis given a design, options, and connection settings.

Usage

```
executeAnalysis(
  connectionDetails = NULL,
  connection = NULL,
  incidenceDesign,
  buildOptions,
  sourceName = "default"
)
```

Arguments

`connectionDetails`	An R object of type `connectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package. Either the `connection` or `connectionDetails` argument should be specified.
`connection`	A connection to the server containing the schema as created using the `connect` function in the `DatabaseConnector` package. Either the `connection` or `connectionDetails` argument should be specified.
`incidenceDesign`	A string object containing valid JSON that represents cohort incidence design
`buildOptions`	the parameters to use in building the analysis queries, created by buildOptions()
`sourceName`	the source name to attach to the results

Details

This method performs the entire cohort incidence analysis in one step without the need for creating any permenant tables.

The process for calculating the cohort incidence is as follows:

- Create the time at risk episodes from the target cohorts, using the specified startWith, endWith and offsets
- Calculate the excluded time by finding the overlap of TAR with the following:
- 1. Outcome cohort episodes (adding the clean window to each outcome episode end date)
- 1. Exclusion cohort that was specified in the outcome definition excludeCohortId.
- Count the distinct persons, distinct persons with cases, total time at risk and total cases, and calcuate the proportion and rates.

The resut contains the following dataframes:

incidence_summary:

Name	Description
REF_ID	The reference id specified in buildOptions() to track results to the analysis execution.
SOURCE_NAME	The name of the source for these results
TARGET_COHORT_DEFIITION_ID	The cohort ID of the target population
TAR_ID	The TAR identifier
SUBGROUP_ID	The subgroup identifier
OUTCOME_ID	The outcome identifier
AGE_GROUP_ID	The age ID for this strata representing the age band specified in the strata settings
GENDER_ID	The gender concept ID for this gender strata
GENDER_NAME	The name of the gender
START_YEAR	The year strata, defined by using the year the TAR started
PERSONS_AT_RISK_PE	Distinct persons at risk before removing excluded time from TAR
PERSONS_AT_RISK	Distinct persons at risk after removing excluded time from TAR. A person must have
PERSON_DAYS_PE	Total TAR (in days) before excluded time was removed from TAR.
PERSON_DAYS	Total TAR (in days) after excluded time was removed from TAR.
PERSON_OUTCOMES_PE	Distinct persons with outcome before removing excluded time from TAR
PERSON_OUTCOMES	Distinct persons with outcome after removing excluded time from TAR. A person mu
OUTCOMES_PE	Number of cases before excluding TAR.
OUTCOMES	Number of cases after excluding TAR.
INCIDENCE_PROPORTION_P100P	The Incidence Proportion (per 100 people), calculated by person_outcomes / persons
INCIDENCE_RATE_P100PY	The Incidence Rate (per 100 person years), calculated by outcomes / person_days / 3

target_def:

Name	Description
REF_ID	The reference id specified in buildOptions() to track results to the analysis execution.
TARGET_COHORT_DEFIITION_ID	The cohort ID of the target population
TARGET_NAME	The name of the target cohort

outcome_def:

Name	Description
REF_ID	The reference id specified in buildOptions() to track results to the analysis exec
OUTCOME_ID	The outcome identifier
OUTCOME_COHORT_DEFINITION_ID	The cohort ID of the outcome population
OUTCOME_NAME	The outcome name

CLEAN_WINDOW	The clean window for this outcome definition
EXCLUDED_COHORT_DEFINITION_ID	The cohort used to exclude TAR

tar_def:

Name	Description
REF_ID	The reference id specified in buildOptions() to track results to the analysis execution.
TAR_ID	The TAR identifier
TAR_START_WITH	Indicates if the TAR starts with the 'start' or 'end' of the target cohort episode
TAR_START_OFFSET	The days added to the date field specified in TAR_START_WITH
TAR_END_WITH	Indicates if the TAR ends with the 'start' or 'end' of the target cohort episode
TAR_END_OFFSET	The days added to the date field specified in TAR_END_WITH

age_group_def:

Name	Description
REF_ID	The reference id specified in buildOptions() to track results to the analysis execution.
AGE_GROUP_ID	The age ID for this strata representing the age band specified in the strata settings
AGE_GROUP_NAME	The name for this age group
MIN_AGE	The minimum age for this group
MAX_AGE	the max age for this group

subgroup_def:

Name	Description
REF_ID	The reference id specified in buildOptions() to track results to the analysis execution.
SUBGROUP_NAME	The name of the subgroup

Value

A named list of `data.frame` objects containing the IR results. The list includes `incidence_summary`, `target_def`, `outcome_def`, `tar_def`, `age_group_def`, and `subgroup_def`, each representing one result table produced during execution.

`getCleanupSql` *Gets the sql that drops the results tables*

Description

Gets the sql that drops the results tables

Usage

```
getCleanupSql(useTempTables = FALSE)
```

Arguments

useTempTables	if TRUE, then temp table notation will be used.

Value

SQL code in MS Sql Server dialect. If analysis will run on another DBMS, translate the SQL with `SqlRender::translateSql()` before execution.

---

getResultsDdl *Gets the results schema DDL for Incidence Analysis*

Description

Gets the results schema DDL for Incidence Analysis

Usage

```
getResultsDdl(useTempTables = FALSE)
```

Arguments

useTempTables	if TRUE, then temp table notation will be used.

Value

SQL code in MS Sql Server dialect, if it's required to run analysis on another DBMS you have to use `translateSql` function in the SqlRender package.

IncidenceAnalysis *R6 Class Representing a IncidenceAnalysis*

Description

The IncidenceAnalysis class, encapsulating the targets, outcomes and tars.

Details

The targets, outcomes and tars fields are referencing IDs of the targetDef, outcomeDef and tarDefs R6 classes.

Note, when serializing with a library such as jsonlite, first call toList() on the R6 class before calling jsonlite::toJSON().

Value

A `IncidenceAnalysis` R6 object that stores the analysis combination as three integer ID vectors: targets, outcomes, and time-at-risk definitions.

Active bindings

`targets` A vector of target IDs from target definitions. Must be a vector.

`outcomes` A vector of outcome IDs from outcome definitions. Must be a vector.

`tars` A vector of TAR IDs from time-at-risk definitions. Must be a vector.

Methods

Public methods:

- `IncidenceAnalysis$new()`
- `IncidenceAnalysis$toList()`
- `IncidenceAnalysis$asJSON()`
- `IncidenceAnalysis$clone()`

`IncidenceAnalysis$new()`: creates a new instance, using the provided data param if provided.

Usage:

```
IncidenceAnalysis$new(data = list())
```

Arguments:

`data` the data (as a json string or list) to initialize with

Returns: A new `IncidenceAnalysis` object initialized from a list or JSON string representation of one analysis combination.

`IncidenceAnalysis$toList()`: returns the R6 class elements as a list for use in jsonlite::toJSON()

Usage:

```
IncidenceAnalysis$toList()
```

`IncidenceAnalysis$asJSON()`: returns the JSON string for this R6 class

Usage:

```
IncidenceAnalysis$asJSON()
```

`IncidenceAnalysis$clone()`: The objects of this class are cloneable with this method.

Usage:

```
IncidenceAnalysis$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`IncidenceDesign` *R6 Class Representing a IncidenceDesign*

Description

This class encapsulates the other R6 Class elements that define an IncidenceDesign

Details

The IncidenceDesign class encapsulates the following:

- Cohort Definitions
- Target Definitions
- Outcome Definitions
- Time At Risk Definitions
- A List of Analyses
- Concept Sets
- Subgruops
- Strata Settings Note, when serializing with a library such as jsonlite, first call toList() on the R6 class before calling jsonlite::toJSON(), or call toJSON directy on this class.

Value

A `IncidenceDesign` R6 object that stores the full cohort incidence analysis design, including cohort definitions, target definitions, outcomes, time-at-risk definitions, analyses, subgroups, strata settings, and study window metadata used to generate SQL.

Active bindings

`cohortDefs` A list of cohort definitions. Must be a list of CohortDefinition

`conceptSets` A list of concept set expressions. Currently unused.

`targetDefs` A list of cohort references to be used as target cohorts. Must be a list of CohortReference

`outcomeDefs` A list of outcome definitions. Must be a list of Outcome

`timeAtRiskDefs` A list of time-at-risk definitions. Must be a list of TimeAtRisk

`analysisList` A list of analyses, containing the T-O-TAR combinations to perform. Must be a list of IncidenceAnalysis

`subgroups` A list of subgroups. Must be a list of CohortSubgroup

`strataSettings` The strata settings for this design. Must be a class StrataSettings

`studyWindow` a study window for this design. Must be a list of class DateRange

`firstAtRisk` a boolean indicating to limit TAR to first per person.

`firstPostOutcome` a boolean indicating to limit TAR to first per person.

Methods

Public methods:

- IncidenceDesign$new()
- IncidenceDesign$toList()
- IncidenceDesign$asJSON()
- IncidenceDesign$clone()

`IncidenceDesign$new()`: creates a new instance, using the provided data param if provided.

Usage:

```
IncidenceDesign$new(data = list())
```

Arguments:

`data` the data (as a json string or list) to initialize with

Returns: A new `IncidenceDesign` object initialized from a list or JSON string representation of the full design.

`IncidenceDesign$toList()`: returns the R6 class elements as a list for use in jsonlite::toJSON()

Usage:

```
IncidenceDesign$toList()
```

`IncidenceDesign$asJSON()`: returns the JSON string for this R6 class

Usage:

```
IncidenceDesign$asJSON(...)
```

Arguments:

`...` paramaters that are passed forward to rjsonlite::toJSON()

`IncidenceDesign$clone()`: The objects of this class are cloneable with this method.

Usage:

```
IncidenceDesign$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`Outcome` *R6 Class Representing an Outcome definition*

Description

The Outcome class, encapsulating the id, name, outcome cohortId, exclusion cohortId, and clean window.

Details

This class is used to specify an outcome definition. The outcome id is distinct from the outcome cohort ID in that you can define multiple outcomes that use the same outcome cohort with different clean windows or exclusion cohort.

Note, when serializing with a library such as jsonlite, first call toList() on the R6 class before calling jsonlite::toJSON().

Value

An `Outcome` R6 object describing one outcome definition, including the outcome cohort id, clean window, and optional exclusion cohort id.

Active bindings

`id` an integer uniquely identifying this outcome definition

`name` the name given to this outcome definition

`cohortId` The outcome cohort ID for this outcome.

`cleanWindow` The clean window for this outcome.

`excludeCohortId` The cohort that will be used to exclude time at risk.

Methods

Public methods:

- `Outcome$new()`
- `Outcome$toList()`
- `Outcome$asJSON()`
- `Outcome$clone()`

`Outcome$new()`: creates a new instance, using the provided data param if provided.

Usage:

```
Outcome$new(data = list())
```

Arguments:

`data` the data (as a json string or list) to initialize with

Returns: A new `Outcome` object initialized from a list or JSON string.

`Outcome$toList()`: returns the R6 class elements as a list for use in jsonlite::toJSON()

Usage:

```
Outcome$toList()
```

Returns: A named list with `id`, `name`, `cohortId`, `cleanWindow`, and `excludeCohortId` fields.

`Outcome$asJSON()`: returns the JSON string for this R6 class

Usage:

```
Outcome$asJSON()
```

Returns: A JSON string representation of the `Outcome` object.

`Outcome$clone()`: The objects of this class are cloneable with this method.

Usage:

```
Outcome$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`StrataSettings` *R6 Class Representing the Stratification Settings of a IncidenceDesign*

Description

The StrataSettings class, encapsulating the age, gender and start-year + age breaks settings.

Details

This class is used to specify the stratification settings for an analysis. The settings can indicate the statistics should be grouped by the age, gender, or start year, and any combination of those selections.

Example: age = TRUE and gender = TRUE will produce statisics by age, by gender, and by age and gender.

Value

A `StrataSettings` R6 object that stores the age, gender, year, and age-break stratification options used during analysis.

Note, when serializing with a library such as jsonlite, first call toList() on the R6 class before calling jsonlite::toJSON().

Active bindings

`byAge` enables stratification by age

`byGender` enables stratification by gender

`byYear` enables stratification by start year of TAR

`ageBreaks` a list of age breaks with at least 1 member

`ageBreakList` a list of age breaks

Methods

Public methods:

- `StrataSettings$new()`
- `StrataSettings$toList()`
- `StrataSettings$asJSON()`
- `StrataSettings$clone()`

`StrataSettings$new()`: creates a new instance, using the provided data param if provided.

Usage:

```
StrataSettings$new(data = list())
```

Arguments:

`data` the data (as a json string or list) to initialize with

Returns: A new `StrataSettings` object initialized from a list or JSON string.

`StrataSettings$toList()`: returns the R6 class elements as a list for use in jsonlite::toJSON()

Usage:

```
StrataSettings$toList()
```

Returns: A named list containing the logical flags and age break lists used by the design.

`StrataSettings$asJSON()`: returns the JSON string for this R6 class

Usage:

```
StrataSettings$asJSON()
```

Returns: A JSON string representation of the `StrataSettings` object.

`StrataSettings$clone()`: The objects of this class are cloneable with this method.

Usage:

```
StrataSettings$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`TimeAtRisk` *R6 Class Representing an Time-at-Risk (TAR) definition*

Description

The TimeAtRisk class, encapsulating the id, startWith, startOffset, endWith and endOffset

Details

This class is used to specify a time-at-risk (TAR) definition. A TAR is defined by choosing the start/end date of a cohort to start with (plus an offset), and a start/end date of the cohort to end with (plus an offset).

Note, when serializing with a library such as jsonlite, first call toList() on the R6 class before calling jsonlite::toJSON().

Value

A `TimeAtRisk` R6 object describing the TAR start and end anchors plus offsets used to compute person-time.

Active bindings

`id` an integer uniquely identifying this time at risk

`startWith` the cohort date to start the time-at-risk. Can be either "start" or "end".

`startOffset` The number of days added to the date specified in startWith.

`endWith` the cohort date to start the time-at-risk. Can be either "start" or "end".

`endOffset` The number of days added to the date specified in startWith.

Methods

Public methods:

- `TimeAtRisk$new()`
- `TimeAtRisk$toList()`
- `TimeAtRisk$asJSON()`
- `TimeAtRisk$clone()`

`TimeAtRisk$new()`: creates a new instance, using the provided data param if provided. The JSON takes the form:`{"id":1,"start":{"dateField":"start","offset":1},"end":{"dateField":"start","offs`

Usage:

```
TimeAtRisk$new(data = list())
```

Arguments:

`data` the data (as a json string or list) to initialize with

Returns: A new `TimeAtRisk` object initialized from a list or JSON string.

`TimeAtRisk$toList()`: returns the R6 class elements as a list for use in jsonlite::toJSON()

Usage:

```
TimeAtRisk$toList()
```

Returns: A named list with `id`, `start`, and `end` fields, where `start` and `end` are nested lists containing `dateField` and `offset`.

`TimeAtRisk$asJSON()`: returns the JSON string for this R6 class

Usage:

```
TimeAtRisk$asJSON()
```

Returns: A JSON string representation of the `TimeAtRisk` object.

`TimeAtRisk$clone()`: The objects of this class are cloneable with this method.

Usage:

```
TimeAtRisk$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Index

buildOptions, 2
buildQuery, 3

CohortDefinition, 4, *19*
CohortReference, 5, *19*
CohortSubgroup, 6, *19*
createCohortRef, 8
createCohortSubgroup, 8
createDateRange, 9
createIncidenceAnalysis, 9
createIncidenceDesign, 10
createOutcomeDef, 11
createStrataSettings, 11
createTimeAtRiskDef, 12

DateRange, 13, *19*

executeAnalysis, 14

getCleanupSql, 16
getResultsDdl, 17

IncidenceAnalysis, 17, *19*
IncidenceDesign, 19

Outcome, *19*, 20

StrataSettings, *19*, 22

TimeAtRisk, *19*, 23
translateSql, *4*, *17*