

Package ‘SelfControlledCaseSeries’

March 27, 2017

Type Package

Title Self-Controlled Case Series

Version 1.1.0

Date 2017-03-23

Author Martijn Schuemie, Patrick Ryan, Trevor Shaddox, Marc A. Suchard

Maintainer Martijn Schuemie <schuemie@ohdsi.org>

Description SelfControlledCaseSeries is an R package for performing self-controlled case series (SCCS) analyses in an observational database in the OMOP Common Data Model.

VignetteBuilder knitr

Depends R (>= 3.2.2),
Cyclops (>= 1.2.0),
DatabaseConnector (>= 1.3.0)

Imports RJDBC,
SqlRender (>= 1.1.1),
bit,
ff,
ffbase (>= 0.12.1),
Rcpp (>= 0.11.2),
OhdsiRTools (>= 1.1.1),
splines,
ggplot2

Suggests testthat,
knitr,
rmarkdown,
EmpiricalCalibration

License Apache License 2.0

LinkingTo Rcpp

NeedsCompilation yes

RoxygenNote 6.0.1

R topics documented:

createAgeSettings	2
createCovariateSettings	3

createCreateSccsEraDataArgs	4
createExposureOutcome	5
createFitSccsModelArgs	5
createGetDbSccsDataArgs	6
createSccsAnalysis	7
createSccsEraData	7
createSccsSimulationSettings	8
createSeasonalitySettings	10
createSimulationRiskWindow	10
cyclicSplineDesign	11
fitSccsModel	12
getDbSccsData	12
getModel	15
loadExposureOutcomeList	15
loadSccsAnalysisList	16
loadSccsData	16
loadSccsEraData	17
plotAgeEffect	17
plotSeasonality	18
runSccsAnalyses	18
saveExposureOutcomeList	20
saveSccsAnalysisList	20
saveSccsData	21
saveSccsEraData	21
SelfControlledCaseSeries	22
simulateSccsData	22
summarizeSccsAnalyses	22

Index 24

createAgeSettings	<i>Create age settings</i>
-------------------	----------------------------

Description

Create age settings

Usage

```
createAgeSettings(includeAge = FALSE, ageKnots = 5,
  allowRegularization = FALSE, minAge = NULL, maxAge = NULL)
```

Arguments

includeAge	Should age be included in the model?
ageKnots	If a single number is provided this is assumed to indicate the number of knots to use for the spline, and the knots are automatically spaced according to equal percentiles of the data. If more than one number is provided these are assumed to be the exact location of the knots in age-days
allowRegularization	When fitting the model, should the covariates defined here be allowed to be regularized?

minAge	Minimum age at which patient time will be included in the analysis. Note that information prior to the min age is still used to determine exposure status after the minimum age (e.g. when a prescription was started just prior to reaching the minimum age). Also, outcomes occurring before the minimum age is reached will be considered as prior outcomes when using first outcomes only. Age should be specified in years, but non-integer values are allowed. If not specified, no age restriction will be applied.
maxAge	Maximum age at which patient time will be included in the analysis. Age should be specified in years, but non-integer values are allowed. If not specified, no age restriction will be applied.

Details

Create an object specifying whether and how age should be included in the model. Age can be included by splitting patient time into calendar months. During a month, the relative risk attributed to age is assumed to be constant, and the risk from month to month is modeled using a cubic spline.

Value

An object of type ageSettings.

```
createCovariateSettings
```

Create covariate settings

Description

Create covariate settings

Usage

```
createCovariateSettings(includeCovariateIds = NULL,
  excludeCovariateIds = NULL, label = "Covariates", stratifyById = TRUE,
  start = 0, addExposedDaysToStart = FALSE, end = 0,
  addExposedDaysToEnd = FALSE, firstOccurrenceOnly = FALSE,
  splitPoints = c(), allowRegularization = FALSE)
```

Arguments

includeCovariateIds	One or more IDs of variables in the sccsData object that should be used to construct this covariate. If no IDs are specified, all variables will be used.
excludeCovariateIds	One or more IDs of variables in the sccsData object that should not be used to construct this covariate.
label	A label used to identify the covariates created using these settings.
stratifyById	Should a single covariate be created for every ID in the sccsData object, or should a single covariate be constructed? For example, if the IDs identify exposures to different drugs, should a covariate be constructed for every drug, or a single covariate for exposure to any of these drugs. Note that overlap will be considered a single exposure.

start	The start of the risk window in days, relative to the exposure start date.
addExposedDaysToStart	Should the length of exposure be added to the start date?
end	The start of the risk window in days, relative to the exposure start date.
addExposedDaysToEnd	Should the length of exposure be added to the end date?
firstOccurrenceOnly	Should only the first occurrence of the exposure be used?
splitPoints	To split the risk window into several smaller windows, specify the end of each sub-window relative to the start of the main risk window. If addExposedDaysToStart is TRUE, the split points will be considered to be relative to the end of the main risk window instead.
allowRegularization	When fitting the model, should the covariates defined here be allowed to be regularized?

Details

Create an object specifying how to create a (set of) covariates.

Value

An object of type covariateSettings.

createCreateSccsEraDataArgs

Create a parameter object for the function createSccsEraData

Description

Create a parameter object for the function createSccsEraData

Usage

```
createCreateSccsEraDataArgs(naivePeriod = 0, firstOutcomeOnly = FALSE,
  covariateSettings, ageSettings = createAgeSettings(includeAge = FALSE),
  seasonalitySettings = createSeasonalitySettings(includeSeasonality = FALSE),
  eventDependentObservation = FALSE)
```

Arguments

naivePeriod	The number of days at the start of a patient's observation period that should not be included in the risk calculations. Note that the naive period can be used to determine current covariate status right after the naive period, and whether an outcome is the first one.
firstOutcomeOnly	Whether only the first occurrence of an outcome should be considered.
covariateSettings	Either an object of type covariateSettings as created using the createCovariateSettings function, or a list of such objects.

ageSettings An object of type ageSettings as created using the createAgeSettings function.

seasonalitySettings An object of type seasonalitySettings as created using the createSeasonalitySettings function.

eventDependentObservation Should the extension proposed by Farrington et al. be used to adjust for event-dependent observation time?

Details

Create an object defining the parameter values.

createExposureOutcome *Create a exposure-outcome combination.*

Description

Create a exposure-outcome combination.

Usage

```
createExposureOutcome(exposureId, outcomeId, ...)
```

Arguments

exposureId A concept ID indentifying the target drug in the exposure table. If multiple strategies for picking the exposure will be tested in the analysis, a named list of numbers can be provided instead. In the analysis, the name of the number to be used can be specified using the #' exposureType parameter in the [createSccsAnalysis](#) function.

outcomeId A concept ID indentifying the outcome in the outcome table.

... Custom variables, to be used in the analyses.

Details

Create a set of hypotheses of interest, to be used with the [runSccsAnalyses](#) function.

createFitSccsModelArgs *Create a parameter object for the function fitSccsModel*

Description

Create a parameter object for the function fitSccsModel

Usage

```
createFitSccsModelArgs(prior = createPrior("laplace", useCrossValidation =
  TRUE), control = createControl(cvType = "auto", selectorType = "byPid",
  startingVariance = 0.1, noiseLevel = "quiet"))
```

Arguments

prior	The prior used to fit the model. See createPrior for details.
control	The control object used to control the cross-validation used to determine the hyperparameters of the prior (if applicable). See createControl for details.

Details

Create an object defining the parameter values.

```
createGetDbSccsDataArgs
```

Create a parameter object for the function getDbSccsData

Description

Create a parameter object for the function getDbSccsData

Usage

```
createGetDbSccsDataArgs(useCustomCovariates = FALSE,
  deleteCovariatesSmallCount = 100, studyStartDate = "",
  studyEndDate = "", exposureIds = "exposureId", customCovariateIds = "")
```

Arguments

useCustomCovariates	Create covariates from a custom table?
deleteCovariatesSmallCount	The minimum count for a covariate to appear in the data to be kept.
studyStartDate	A calendar date specifying the minimum date where data is used. Date format is 'yyyymmdd'.
studyEndDate	A calendar date specifying the maximum date where data is used. Date format is 'yyyymmdd'.
exposureIds	A list of identifiers to define the exposures of interest. If exposureTable = DRUG_ERA, exposureIds should be CONCEPT_ID. If exposureTable <> DRUG_ERA, exposureIds is used to select the cohort_concept_id in the cohort-like table. If no exposureIds are provided, all drugs or cohorts in the exposureTable are included as exposures.
customCovariateIds	A list of cohort definition IDS identifying the records in the customCovariateTable to use for building customcovariates.

Details

Create an object defining the parameter values.

createSccsAnalysis	Create a SelfControlledCaseSeries analysis specification
--------------------	--

Description

Create a SelfControlledCaseSeries analysis specification

Usage

```
createSccsAnalysis(analysisId = 1, description = "", exposureType = NULL,  
  outcomeType = NULL, getDbSccsDataArgs, createSccsEraDataArgs,  
  fitSccsModelArgs)
```

Arguments

analysisId	An integer that will be used later to refer to this specific set of analysis choices.
description	A short description of the analysis.
exposureType	If more than one exposure is provided for each exposureOutcome, this field should be used to select the specific exposure to use in this analysis.
outcomeType	If more than one outcome is provided for each exposureOutcome, this field should be used to select the specific outcome to use in this analysis.
getDbSccsDataArgs	An object representing the arguments to be used when calling the getDbSccsData function.
createSccsEraDataArgs	An object representing the arguments to be used when calling the createSccsEraData function.
fitSccsModelArgs	An object representing the arguments to be used when calling the fitSccsModel function.

Details

Create a set of analysis choices, to be used with the [runSccsAnalyses](#) function.

createSccsEraData	Create SCCS era data
-------------------	----------------------

Description

Create SCCS era data

Usage

```
createSccsEraData(sccsData, outcomeId = NULL, naivePeriod = 0,  
  firstOutcomeOnly = FALSE, covariateSettings,  
  ageSettings = createAgeSettings(includeAge = FALSE),  
  seasonalitySettings = createSeasonalitySettings(includeSeasonality = FALSE),  
  eventDependentObservation = FALSE)
```

Arguments

sccsData	An object of type <code>sccsData</code> as created using the getDbSccsData function.
outcomeId	The outcome to create the era data for. If not specified it is assumed to be the one outcome for which the data was loaded from the database.
naivePeriod	The number of days at the start of a patient's observation period that should not be included in the risk calculations. Note that the naive period can be used to determine current covariate status right after the naive period, and whether an outcome is the first one.
firstOutcomeOnly	Whether only the first occurrence of an outcome should be considered.
covariateSettings	Either an object of type <code>covariateSettings</code> as created using the createCovariateSettings function, or a list of such objects.
ageSettings	An object of type <code>ageSettings</code> as created using the createAgeSettings function.
seasonalitySettings	An object of type <code>seasonalitySettings</code> as created using the createSeasonalitySettings function.
eventDependentObservation	Should the extension proposed by Farrington et al. be used to adjust for event-dependent observation time?

Details

This function creates covariates based on the data in the `sccsData` object, according to the provided settings. It chops patient time into periods during which all covariates remain constant. The output details these periods, their durations, and a sparse representation of the covariate values.

Value

An object of type `sccsEraData`.

References

Farrington, C. P., Anaya-Izquierdo, A., Whitaker, H. J., Hocine, M.N., Douglas, I., and Smeeth, L. (2011). Self-Controlled case series analysis with event-dependent observation periods. *Journal of the American Statistical Association* 106 (494), 417-426

```
createSccsSimulationSettings
```

Create SCCS simulation settings

Description

Create SCCS simulation settings

Usage

```
createSccsSimulationSettings(meanPatientTime = 4 * 365, sdPatientTime = 2 *
  365, minAge = 18 * 365, maxAge = 65 * 365, minBaselineRate = 0.001,
  maxBaselineRate = 0.01, covariateIds = c(1, 2), patientUsages = c(0.2,
  0.1), usageRate = c(0.01, 0.01), meanPrescriptionDurations = c(14, 30),
  sdPrescriptionDurations = c(7, 14),
  simulationRiskWindows = list(createSimulationRiskWindow(relativeRisks = 1),
  createSimulationRiskWindow(relativeRisks = 1.5)), includeAgeEffect = TRUE,
  ageKnots = 5, includeSeasonality = TRUE, seasonKnots = 5,
  outcomeId = 10)
```

Arguments

meanPatientTime	Mean number of observation days per patient.
sdPatientTime	Standard deviation of the observation days per patient.
minAge	The minimum age in days.
maxAge	The maximum age in days.
minBaselineRate	The minimum baseline rate (per day).
maxBaselineRate	The maximum baseline rate (per day).
covariateIds	The IDs for the covariates to be generated.
patientUsages	The fraction of patients that use the drugs.
usageRate	The rate of prescriptions per person that uses the drug.
meanPrescriptionDurations	The mean duration of a prescription, per drug.
sdPrescriptionDurations	The standard deviation of the duration of a prescription, per drug.
simulationRiskWindows	One or a list of objects of type <code>simulationRiskWindow</code> as created using the createSimulationRiskWindow function.
includeAgeEffect	Include an age effect for the outcome?
ageKnots	Number of knots in the age spline.
includeSeasonality	Include seasonality for the outcome?
seasonKnots	Number of knots in the seasonality spline.
outcomeId	The ID to be used for the outcome.

Details

Create an object of settings for an SCCS simulation.

Value

An object of type `sccsSimulationSettings`.

```
createSeasonalitySettings
```

Create seasonality settings

Description

Create seasonality settings

Usage

```
createSeasonalitySettings(includeSeasonality = FALSE, seasonKnots = 5,
  allowRegularization = FALSE)
```

Arguments

<code>includeSeasonality</code>	Should seasonlaity be included in the model?
<code>seasonKnots</code>	If a single number is provided this is assumed to indicate the number of knots to use for the spline, and the knots are automatically equally spaced across the year. If more than one number is provided these are assumed to be the exact location of the knots in days relative to the start of the year.
<code>allowRegularization</code>	When fitting the model, should the covariates defined here be allowed to be regularized?

Details

Create an object specifying whether and how seasonality should be included in the model. Seasonality can be included by splitting patient time into calendar months. During a month, the relative risk attributed to season is assumed to be constant, and the risk from month to month is modeled using a cyclic cubic spline.

Value

An object of type `seasonalitySettings`.

```
createSimulationRiskWindow
```

Create a risk window definition for simulation

Description

Create a risk window definition for simulation

Usage

```
createSimulationRiskWindow(start = 0, end = 0, addExposedDaysToEnd = TRUE,
  splitPoints = c(), relativeRisks = c(0))
```

Arguments

start	Start of the risk window relative to exposure start.
end	End of risk window relative to exposure start, or if addExposedDaysToEnd is TRUE, relative to the end date.
addExposedDaysToEnd	Should the length of exposure be added to the end date? In other words, should the exposure end date be used as reference point for the risk window end?
splitPoints	Subdivision of the risk window in to smaller sub-windows.
relativeRisks	Either a single number representing the relative risk in the risk window, or when splitPoints have been defined a vector of relative risks, one for each sub-window.

Value

An object of type `simulationRiskWindow`.

cyclicSplineDesign	<i>Create a design matrix for a cyclic spline</i>
--------------------	---

Description

Create a design matrix for a cyclic spline

Usage

```
cyclicSplineDesign(x, knots, ord = 4)
```

Arguments

x	Vector of coordinates of the points to be interpolated.
knots	Location of the knots.
ord	Order of the spline function.

Details

This function is used by other functions in this package.

fitSccsModel	<i>Fit the SCCS model</i>
--------------	---------------------------

Description

Fit the SCCS model

Usage

```
fitSccsModel(sccsEraData, prior = createPrior("laplace", useCrossValidation =
  TRUE), control = createControl(cvType = "auto", selectorType = "byPid",
  startingVariance = 0.1, noiseLevel = "quiet"))
```

Arguments

sccsEraData	An object of type sccsEraData as created using the createSccsEraData function.
prior	The prior used to fit the model. See createPrior for details.
control	The control object used to control the cross-validation used to determine the hyperparameters of the prior (if applicable). See createControl for details.

Details

Fits the SCCS model as a conditional Poisson regression. When allowed, coefficients for some or all covariates can be regularized.

Value

An object of type sccsModel. Generic functions `summary`, `coef`, and `confint` are available.

References

Suchard, M.A., Simpson, S.E., Zorych, I., Ryan, P., and Madigan, D. (2013). Massive parallelization of serial inference algorithms for complex generalized linear models. *ACM Transactions on Modeling and Computer Simulation* 23, 10

getDbSccsData	<i>Load data for SCCS from the database</i>
---------------	---

Description

Load all data needed to perform an SCCS analysis from the database.

Usage

```
getDbSccsData(connectionDetails, cdmDatabaseSchema,
  oracleTempSchema = cdmDatabaseSchema,
  outcomeDatabaseSchema = cdmDatabaseSchema, outcomeTable = "condition_era",
  outcomeIds, exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era", exposureIds = c(),
  useCustomCovariates = FALSE,
  customCovariateDatabaseSchema = cdmDatabaseSchema,
  customCovariateTable = "cohort", customCovariateIds = c(),
  deleteCovariatesSmallCount = 100, studyStartDate = "",
  studyEndDate = "", cdmVersion = "4")
```

Arguments

connectionDetails

An R object of type `ConnectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package.

cdmDatabaseSchema

The name of the database schema that contains the OMOP CDM instance. Requires read permissions to this database. On SQL Server, this should specify both the database and the schema, so for example `'cdm_instance.dbo'`.

oracleTempSchema

A schema where temp tables can be created in Oracle.

outcomeDatabaseSchema

The name of the database schema that is the location where the data used to define the outcome cohorts is available. If `outcomeTable = CONDITION_ERA`, `outcomeDatabaseSchema` is not used. Requires read permissions to this database.

outcomeTable

The tablename that contains the outcome cohorts. If `outcomeTable` is not `CONDITION_OCCURRENCE` or `CONDITION_ERA`, then expectation is `outcomeTable` has format of `COHORT` table: `COHORT_DEFINITION_ID`, `SUBJECT_ID`, `COHORT_START_DATE`, `COHORT_END_DATE`.

outcomeIds

A list of ids used to define outcomes. If `outcomeTable = CONDITION_OCCURRENCE`, the list is a set of ancestor `CONCEPT_ID`s, and all occurrences of all descendant concepts will be selected. If `outcomeTable <> CONDITION_OCCURRENCE`, the list contains records found in `COHORT_DEFINITION_ID` field.

exposureDatabaseSchema

The name of the database schema that is the location where the exposure data used to define the exposure cohorts is available. If `exposureTable = DRUG_ERA`, `exposureDatabaseSchema` is not used but assumed to be `cdmSchema`. Requires read permissions to this database.

exposureTable

The tablename that contains the exposure cohorts. If `exposureTable <> DRUG_ERA`, then expectation is `exposureTable` has format of `COHORT` table: `cohort_concept_id`, `SUBJECT_ID`, `COHORT_START_DATE`, `COHORT_END_DATE`.

exposureIds

A list of identifiers to define the exposures of interest. If `exposureTable = DRUG_ERA`, `exposureIds` should be `CONCEPT_ID`. If `exposureTable <> DRUG_ERA`, `exposureIds` is used to select the `cohort_concept_id` in the cohort-like table. If no `exposureIds` are provided, all drugs or cohorts in the `exposureTable` are included as exposures.

useCustomCovariates

Create covariates from a custom table?

<code>customCovariateDatabaseSchema</code>	The name of the database schema that is the location where the custom covariate data is available.
<code>customCovariateTable</code>	Name of the table holding the custom covariates. This table should have the same structure as the cohort table.
<code>customCovariateIds</code>	A list of cohort definition IDS identifying the records in the <code>customCovariateTable</code> to use for building custom covariates.
<code>deleteCovariatesSmallCount</code>	The minimum count for a covariate to appear in the data to be kept.
<code>studyStartDate</code>	A calendar date specifying the minimum date where data is used. Date format is 'yyyymmdd'.
<code>studyEndDate</code>	A calendar date specifying the maximum date where data is used. Date format is 'yyyymmdd'.
<code>cdmVersion</code>	Define the OMOP CDM version used: currently support "4" and "5".

Details

This function downloads several types of information:

- Information on the occurrences of the outcome(s) of interest. Note that information for multiple outcomes can be fetched in one go, and later the specific outcome can be specified for which we want to build a model.
- Information on the observation time and age for the people with the outcomes.
- Information on exposures of interest which we want to include in the model.

Four different database schemas can be specified, for four different types of information: The `cdmDatabaseSchema` is used to extract patient age and observation period. The `outcomeDatabaseSchema` is used to extract information about the outcomes, the `exposureDatabaseSchema` is used to retrieve information on exposures, and the `customCovariateDatabaseSchema` is optionally used to find additional, user-defined covariates. All four locations could point to the same database schema.

Value

Returns an object of type `sccsData`, containing information on the cases, their outcomes, exposures, and potentially other covariates. Information about multiple outcomes can be captured at once for efficiency reasons. This object is a list with the following components:

cases An `ffdf` object listing the persons that have the outcome(s), their age, and observation time.

eras An `ffdf` object listing the exposures, outcomes and other covariates.

covariateRef An `ffdf` object describing the covariates that have been extracted.

metaData A list of objects with information on how the `sccsData` object was constructed.

The generic `summary()` function has been implemented for this object.

getModel	<i>Output the full model</i>
----------	------------------------------

Description

Output the full model

Usage

```
getModel(sccsModel)
```

Arguments

sccsModel An object of type sccsModel as created using the [fitSccsModel](#) function.

Value

A data frame with the coefficients and confidence intervals (when not-regularized) for all covariates in the model.

loadExposureOutcomeList	<i>Load a list of exposureOutcome from file</i>
-------------------------	---

Description

Load a list of objects of type exposureOutcome from file. The file is in JSON format.

Usage

```
loadExposureOutcomeList(file)
```

Arguments

file The name of the file

Value

A list of objects of type exposureOutcome.

loadSccsAnalysisList	<i>Load a list of sccsAnalysis from file</i>
----------------------	--

Description

Load a list of objects of type sccsAnalysis from file. The file is in JSON format.

Usage

```
loadSccsAnalysisList(file)
```

Arguments

file	The name of the file
------	----------------------

Value

A list of objects of type sccsAnalysis.

loadSccsData	<i>Load the SCCS data from a folder</i>
--------------	---

Description

loadSccsData loads an object of type sccsData from a folder in the file system.

Usage

```
loadSccsData(folder, readOnly = TRUE)
```

Arguments

folder	The name of the folder containing the data.
readOnly	If true, the data is opened read only.

Details

The data will be written to a set of files in the folder specified by the user.

Value

An object of class cohortData.

loadSccsEraData	<i>Load the SCCS era data from a folder</i>
-----------------	---

Description

loadSccsEraData loads an object of type `sccsEraData` from a folder in the file system.

Usage

```
loadSccsEraData(folder, readOnly = FALSE)
```

Arguments

folder	The name of the folder containing the data.
readOnly	If true, the data is opened read only.

Details

The data will be written to a set of files in the folder specified by the user.

Value

An object of class `sccsEraData`

plotAgeEffect	<i>Plot the age effect</i>
---------------	----------------------------

Description

Plot the age effect

Usage

```
plotAgeEffect(sccsModel, rrLim = c(0.1, 10), fileName = NULL)
```

Arguments

sccsModel	An object of type <code>sccsModel</code> as created using the fitSccsModel function.
rrLim	The limits on the incidence rate ratio scale in the plot.
fileName	Name of the file where the plot should be saved, for example 'plot.png'. See the function <code>ggsave</code> in the <code>ggplot2</code> package for supported file formats.

Details

Plot the spline curve of the age effect.

plotSeasonality	<i>Plot the seasonality effect</i>
-----------------	------------------------------------

Description

Plot the seasonality effect

Usage

```
plotSeasonality(sccsModel, rrLim = c(0.1, 10), fileName = NULL)
```

Arguments

sccsModel	An object of type sccsModel as created using the fitSccsModel function.
rrLim	The limits on the incidence rate ratio scale in the plot.
fileName	Name of the file where the plot should be saved, for example 'plot.png'. See the function ggsave in the ggplot2 package for supported file formats.

Details

Plot the spline curve of the seasonality effect.

runSccsAnalyses	<i>Run a list of analyses</i>
-----------------	-------------------------------

Description

Run a list of analyses

Usage

```
runSccsAnalyses(connectionDetails, cdmDatabaseSchema,
  oracleTempSchema = cdmDatabaseSchema,
  exposureDatabaseSchema = cdmDatabaseSchema, exposureTable = "drug_era",
  outcomeDatabaseSchema = cdmDatabaseSchema, outcomeTable = "condition_era",
  customCovariateDatabaseSchema = cdmDatabaseSchema,
  customCovariateTable = "cohort", cdmVersion = 5,
  outputFolder = "./SccsOutput", sccsAnalysisList, exposureOutcomeList,
  combineDataFetchAcrossOutcomes = TRUE, getDbSccsDataThreads = 1,
  createSccsEraDataThreads = 1, fitSccsModelThreads = 1, cvThreads = 1)
```

Arguments

connectionDetails	An R object of type ConnectionDetails created using the function createConnectionDetails in the DatabaseConnector package.
cdmDatabaseSchema	The name of the database schema that contains the OMOP CDM instance. Requires read permissions to this database. On SQL Server, this should specify both the database and the schema, so for example 'cdm_instance.dbo'.

oracleTempSchema	A schema where temp tables can be created in Oracle.
exposureDatabaseSchema	The name of the database schema that is the location where the exposure data used to define the exposure cohorts is available. If exposureTable = DRUG_ERA, exposureDatabaseSchema is not used but assumed to be cdmSchema. Requires read permissions to this database.
exposureTable	The tablename that contains the exposure cohorts. If exposureTable <> DRUG_ERA, then expectation is exposureTable has format of COHORT table: cohort_concept_id, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.
outcomeDatabaseSchema	The name of the database schema that is the location where the data used to define the outcome cohorts is available. If outcomeTable = CONDITION_ERA, outcomeDatabaseSchema is not used. Requires read permissions to this database.
outcomeTable	The tablename that contains the outcome cohorts. If outcomeTable is not CONDITION_OCCURRENCE or CONDITION_ERA, then expectation is outcomeTable has format of COHORT table: COHORT_DEFINITION_ID, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.
customCovariateDatabaseSchema	The name of the database schema that is the location where the custom covariate data is available.
customCovariateTable	Name of the table holding the custom covariates. This table should have the same structure as the cohort table.
cdmVersion	Define the OMOP CDM version used: currently support "4" and "5".
outputFolder	Name of the folder where all the outputs will written to.
sccsAnalysisList	A list of objects of type sccsAnalysis as created using the createSccsAnalysis function.
exposureOutcomeList	A list of objects of type exposureOutcome as created using the createExposureOutcome function.
combineDataFetchAcrossOutcomes	Should fetching data from the database be done one outcome at a time, or for all outcomes in one fetch? Combining fetches will be more efficient if there is large overlap in the subjects that have the different outcomes.
getDbSccsDataThreads	The number of parallel threads to use for building the sccsData objects.
createSccsEraDataThreads	The number of parallel threads to use for building the sccsEraData objects.
fitSccsModelThreads	The number of parallel threads to use for fitting the models.
cvThreads	The number of parallel threads to use for the cross- validation when estimating the hyperparameter for the outcome model. Note that the total number of CV threads at one time could be 'fitSccsModelThreads * cvThreads'.

Details

Run a list of analyses for the drug-comparator-outcomes of interest. This function will run all specified analyses against all hypotheses of interest, meaning that the total number of outcome

models is 'length(cmAnalysisList) * length(drugComparatorOutcomesList)' (if all analyses specify an outcome model should be fitted). When you provide several analyses it will determine whether any of the analyses have anything in common, and will take advantage of this fact. For example, if we specify several analyses that only differ in the way the outcome model is fitted, then this function will extract the data and fit the propensity model only once, and re-use this in all the analysis.

Value

A data frame with the following columns:

analysisId	The unique identifier for a set of analysis choices.
exposureId	The ID of the target drug.
outcomeId	The ID of the outcome.
sccsDataFolder	The folder where the sccsData object is stored.
sccsEraDataFolder	The folder where the sccsEraData object is stored.
sccsModelFile	The file where the fitted SCCS model is stored.

saveExposureOutcomeList

Save a list of exposureOutcome to file

Description

Write a list of objects of type exposureOutcome to file. The file is in JSON format.

Usage

```
saveExposureOutcomeList(exposureOutcomeList, file)
```

Arguments

exposureOutcomeList	The exposureOutcome list to be written to file
file	The name of the file where the results will be written

saveSccsAnalysisList *Save a list of sccsAnalysis to file*

Description

Write a list of objects of type sccsAnalysis to file. The file is in JSON format.

Usage

```
saveSccsAnalysisList(sccsAnalysisList, file)
```

Arguments

sccsAnalysisList	The sccsAnalysis list to be written to file
file	The name of the file where the results will be written

saveSccsData	<i>Save the SCCS data to folder</i>
--------------	-------------------------------------

Description

saveSccsData saves an object of type `sccsData` to folder.

Usage

```
saveSccsData(sccsData, folder)
```

Arguments

<code>sccsData</code>	An object of type <code>sccsData</code> as generated using getDbSccsData .
<code>folder</code>	The name of the folder where the data will be written. The folder should not yet exist.

Details

The data will be written to a set of files in the specified folder.

Examples

```
# todo
```

saveSccsEraData	<i>Save the SCCS era data to folder</i>
-----------------	---

Description

saveSccsEraData saves an object of type `sccsEraData` to folder.

Usage

```
saveSccsEraData(sccsEraData, folder)
```

Arguments

<code>sccsEraData</code>	An object of type <code>sccsEraData</code> as generated using createSccsEraData .
<code>folder</code>	The name of the folder where the data will be written. The folder should not yet exist.

Details

The data will be written to a set of files in the specified folder.

SelfControlledCaseSeries	<i>SelfControlledCaseSeries</i>
--------------------------	---------------------------------

Description

SelfControlledCaseSeries

simulateSccsData	<i>Simulate SCCS data</i>
------------------	---------------------------

Description

Simulate SCCS data

Usage

simulateSccsData(nCases, settings)

Arguments

- nCases The number of cases to simulate.
- settings An object of type sccsSimulationSettings as created using the [createSccsSimulationSettings](#)

Value

An object of type sccsData.

summarizeSccsAnalyses	<i>Create a summary report of the analyses</i>
-----------------------	--

Description

Create a summary report of the analyses

Usage

summarizeSccsAnalyses(outcomeReference)

Arguments

- outcomeReference A data.frame as created by the [runSccsAnalyses](#) function.

Value

A data frame with the following columns:

analysisId	The unique identifier for a set of analysis choices.
targetId	The ID of the target drug.
comparatorId	The ID of the comparator group.
indicationConceptIds	The ID(s) of indications in which to nest to study.
outcomeId	The ID of the outcome.
rr	The estimated effect size.
ci95lb	The lower bound of the 95 percent confidence interval.
ci95ub	The upper bound of the 95 percent confidence interval.
treated	The number of subjects in the treated group (after any trimming and matching).
comparator	The number of subjects in the comparator group (after any trimming and matching).
eventsTreated	The number of outcomes in the treated group (after any trimming and matching).
eventsComparator	The number of outcomes in the comparator group (after any trimming and matching).
logRr	The log of the estimated relative risk.
seLogRr	The standard error of the log of the estimated relative risk.

Index

`createAgeSettings`, [2](#), [8](#)
`createControl`, [12](#)
`createCovariateSettings`, [3](#), [8](#)
`createCreateSccsEraDataArgs`, [4](#)
`createExposureOutcome`, [5](#), [19](#)
`createFitSccsModelArgs`, [5](#)
`createGetDbSccsDataArgs`, [6](#)
`createPrior`, [12](#)
`createSccsAnalysis`, [5](#), [7](#), [19](#)
`createSccsEraData`, [7](#), [7](#), [12](#), [21](#)
`createSccsSimulationSettings`, [8](#), [22](#)
`createSeasonalitySettings`, [8](#), [10](#)
`createSimulationRiskWindow`, [9](#), [10](#)
`cyclicSplineDesign`, [11](#)

`fitSccsModel`, [7](#), [12](#), [15](#), [17](#), [18](#)

`getDbSccsData`, [7](#), [8](#), [12](#), [21](#)
`getModel`, [15](#)

`loadExposureOutcomeList`, [15](#)
`loadSccsAnalysisList`, [16](#)
`loadSccsData`, [16](#)
`loadSccsEraData`, [17](#)

`plotAgeEffect`, [17](#)
`plotSeasonality`, [18](#)

`runSccsAnalyses`, [5](#), [7](#), [18](#), [22](#)

`saveExposureOutcomeList`, [20](#)
`saveSccsAnalysisList`, [20](#)
`saveSccsData`, [21](#)
`saveSccsEraData`, [21](#)
`SelfControlledCaseSeries`, [22](#)
`SelfControlledCaseSeries-package`
 (`SelfControlledCaseSeries`), [22](#)
`simulateSccsData`, [22](#)
`summarizeSccsAnalyses`, [22](#)