Package 'SelfControlledCaseSeries'

March 10, 2016

Type Package

Title Self-Controlled Case Series

Version 0.0.2

Date 2015-11-10

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 (>= 1.0.1),
splines,
ggplot2

Suggests testthat,
knitr,
rmarkdown

License Apache License 2.0

LinkingTo Rcpp

NeedsCompilation yes

RoxygenNote 5.0.1

R topics documented:

`createAgeSettings` *Create age settings*

Description

Create age settings

Usage

```
createAgeSettings(includeAge = FALSE, ageKnots = 5,
  allowRegularization = FALSE)
```

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?

Details

Create an object specifing 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.

---

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 sccsData 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 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

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 `simulationRiskWindow` 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

includeSeasonality	Should seasonlaity be included in the model?
seasonKnots	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 equaly 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.
allowRegularization	When fitting the model, should the covariates defined here be allowed to be regularized?

Details

Create an object specifing 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 *Create a design matrix for a cyclic spline*

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 *Fit the SCCS model*

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 *Load data for SCCS from the database*

Description

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

Usage

```
getDbSccsData(connectionDetails, cdmDatabaseSchema,
  oracleTempSchema = cdmDatabaseSchema,
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "condition_occurrence", outcomeIds,
  outcomeConditionTypeConceptIds = c(),
  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 specifiy 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_IDs, and all occurrences of all descendant concepts will be selected. If outcomeTable <> CONDITION_OCCURRENCE, the list contains records found in COHORT_DEFINITION_ID field.

`outcomeConditionTypeConceptIds`
: A list of TYPE_CONCEPT_ID values that will restrict condition occurrences. Only applicable if outcomeTable = CONDITION_OCCURRENCE.

`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 unique identifier 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?

`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.

`customCovariateIds`
: A list of cohort definition IDS identifying the records in the customCovariateTable to use for building custom covariates.

`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'.

`cdmVersion`
: 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 *Output the full model*

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.

loadSccsData *Load the SCCS data from a folder*

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 *Load the SCCS era data from a folder*

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 *Plot the age effect*

Description

Plot the age effect

Usage

```
plotAgeEffect(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 age effect.

plotSeasonality *Plot the seasonality effect*

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.

saveSccsData *Save the SCCS data to folder*

Description

sccsData saves an object of type sccsData to folder.

Usage

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

Arguments

sccsData	An object of type sccsData as generated using getDbSccsData.
folder	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 *Save the SCCS era data to folder*

Description

saveSccsEraData saves an object of type sccsEraData to folder.

Usage

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

Arguments

sccsEraData	An object of type sccsEraData as generated using createSccsEraData.
folder	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 *SelfControlledCaseSeries*

Description

SelfControlledCaseSeries

simulateSccsData *Simulate SCCS data*

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.

Index