

# Package ‘CaseControl’

May 23, 2016

**Type** Package

**Title** Case-Control

**Version** 0.0.1

**Date** 2016-05-18

**Author** Martijn Schuemie

**Maintainer** Martijn Schuemie <schuemie@ohdsi.org>

**Description** CaseControl is an R package for performing (nested) matched case-control 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),  
survival

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

**Suggests** testthat,  
knitr,  
rmarkdown

**License** Apache License 2.0

**LinkingTo** Rcpp

**NeedsCompilation** yes

**RoxygenNote** 5.0.1

## R topics documented:

CaseControl . . . . .	2
createCaseControlData . . . . .	2
fitCaseControlModel . . . . .	3
getDbCaseData . . . . .	3
getDbExposureData . . . . .	5

insertDbPopulation . . . . .	5
loadCaseData . . . . .	6
saveCaseData . . . . .	7
selectControls . . . . .	7
<b>Index</b>	<b>9</b>

---

CaseControl	<i>CaseControl</i>
-------------	--------------------

---

**Description**

CaseControl

---

createCaseControlData	<i>Create case-control data</i>
-----------------------	---------------------------------

---

**Description**

Create case-control data

**Usage**

```
createCaseControlData(caseControlsExposure, exposureId,  
  firstExposureOnly = FALSE, riskWindowStart = 0, riskWindowEnd = 0)
```

**Arguments**

- caseControlsExposure  
An object of type caseControlsExposure as created using the [getDbExposureData](#) function.
- exposureId  
The identifier of the exposure.
- firstExposureOnly  
Should only the first exposure per subject be included?
- riskWindowStart  
The start of the risk window (in days) relative to the index date. This number should be non-positive.
- riskWindowEnd  
The end of the risk window (in days) relative to the index date. This number should be non-positive.

**Details**

For each case and control, assesses whether exposure takes place within the risk window. The output can be directly used in a conditional logistic regression.

**Value**

A data frame with these columns:

**personId** The person ID

**indexDate** The index date

**isCase** Is the person a case or a control?

**stratumId** The ID linking cases and controls in a matched set

**exposed** Was the subject exposed during the risk window?

---

fitCaseControlModel	<i>Fit the case-control model</i>
---------------------	-----------------------------------

---

**Description**

Fit the case-control model

**Usage**

```
fitCaseControlModel(caseControlData)
```

**Arguments**

caseControlData

A data frame as generated by the [createCaseControlData](#) function.

**Details**

Fits the model using a conditional logistic regression.

**Value**

An object of type outcomeModel.

---

getDbCaseData	<i>Load case data from the database</i>
---------------	---

---

**Description**

Load all data about the cases and nesting cohort from the database.

**Usage**

```
getDbCaseData(connectionDetails, cdmDatabaseSchema,
  oracleTempSchema = cdmDatabaseSchema,
  outcomeDatabaseSchema = cdmDatabaseSchema, outcomeTable = "condition_era",
  outcomeIds = c(), useNestingCohort = FALSE,
  nestingCohortDatabaseSchema = cdmDatabaseSchema,
  nestingCohortTable = "cohort", nestingCohortId,
  useObservationEndAsNestingEndDate = TRUE, getVisits = TRUE,
  studyStartDate = "", studyEndDate = "")
```

## 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_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.
useNestingCohort	Should the study be nested in a cohort (e.g. people with a specific indication)? If not, the study will be nested in the general population.
nestingCohortDatabaseSchema	The name of the database schema that is the location where the nesting cohort is defined.
nestingCohortTable	Name of the table holding the nesting cohort. This table should have the same structure as the cohort table.
nestingCohortId	A cohort definition ID identifying the records in the nestingCohortTable to use as nesting cohort
useObservationEndAsNestingEndDate	When using a nesting cohort, should the observation period end date be used instead of the cohort end date?
getVisits	Get data on visits? This is needed when matching on visit date is requested later on.
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'.

## Value

Returns an object of type caseData, containing information on the cases, the nesting cohort, and optionally visits. Information about multiple outcomes can be captured at once for efficiency reasons. The generic summary() function has been implemented for this object.

---

getDbExposureData	<i>Get exposure data for cases and controls from a database</i>
-------------------	---

---

## Description

Get exposure data for cases and controls from a database

## Usage

```
getDbExposureData(connectionDetails, caseControls, oracleTempSchema = NULL,
  exposureDatabaseSchema, exposureTable = "drug_era", exposureIds = c())
```

## Arguments

connectionDetails

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

caseControls A data frame as generated by the [selectControls](#) function.

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\_definition\_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\_definition\_id in the cohort-like table. If no exposureIds are provided, all drugs or cohorts in the exposureTable are included as exposures.

---

insertDbPopulation	<i>Insert cases and controls into a database</i>
--------------------	--

---

## Description

Insert cases and controls into a database

## Usage

```
insertDbPopulation(caseControls, cohortIds = c(1, 0), connectionDetails,
  cohortDatabaseSchema, cohortTable = "cohort", createTable = FALSE,
  dropTableIfExists = TRUE)
```

**Arguments**

caseControls	A data frame as generated by the <a href="#">selectControls</a> function.
cohortIds	The IDs to be used for the cohorts of cases and controls, respectively.
connectionDetails	An R object of type <code>connectionDetails</code> created using the function <code>createConnectionDetails</code> in the <code>DatabaseConnector</code> package.
cohortDatabaseSchema	The name of the database schema where the data will be written. Requires write permissions to this database. On SQL Server, this should specify both the database and the schema, so for example <code>'cdm_instance.dbo'</code> .
cohortTable	The name of the table in the database schema where the data will be written.
createTable	Should a new table be created? If not, the data will be inserted into an existing table.
dropTableIfExists	If <code>createTable = TRUE</code> and the table already exists it will be overwritten.

**Details**

Inserts cases and controls into a database. The table in the database will have the same structure as the 'cohort' table in the Common Data Model.

---

loadCaseData	<i>Load the case data from a folder</i>
--------------	---

---

**Description**

loadCaseData loads an object of type `caseData` from a folder in the file system.

**Usage**

```
loadCaseData(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 `caseData`.

---

saveCaseData	<i>Save the case data to folder</i>
--------------	-------------------------------------

---

### Description

saveCaseData saves an object of type caseData to folder.

### Usage

```
saveCaseData(caseData, folder)
```

### Arguments

caseData	An object of type caseData as generated using <a href="#">getDbCaseData</a> .
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.

---

selectControls	<i>Select matched controls per case</i>
----------------	---

---

### Description

Select matched controls per case

### Usage

```
selectControls(caseData, outcomeId, firstOutcomeOnly = TRUE,
  washoutPeriod = 180, controlsPerCase = 2, matchOnAge = TRUE,
  ageCaliper = 2, matchOnGender = TRUE, matchOnProvider = FALSE,
  matchOnVisitDate = FALSE, visitDateCaliper = 30)
```

### Arguments

caseData	An object of type caseData as generated using the <a href="#">getDbCaseData</a> function.
outcomeId	The outcome ID of the cases for which we need to pick controls.
firstOutcomeOnly	Use the first outcome per person?
washoutPeriod	Minimum required numbers of days of observation for inclusion as either case or control.
matchOnAge	Match on age?
ageCaliper	Maximum difference (in years) in age when matching on age.
matchOnGender	Match on gender?
matchOnProvider	Match on provider (as specified in the person table)?

`matchOnVisitDate`

Should the index date of the control be changed to the nearest visit date?

`visitDateCaliper`

Maximum difference (in days) between the index date and the visit date when matching on visit date.

`controlsPerCase`

Maximum number of controls to select per case.

### Details

Select controls per case. Controls are matched on calendar time and the criteria defined in the arguments. Controls are randomly sampled to the required number.

### Value

A data frame with these columns:

**personId** The person ID

**indexDate** The index date

**isCase** Is the person a case or a control?

**stratumId** The ID linking cases and controls in a matched set



# Index

CaseControl, [2](#)  
CaseControl-package (CaseControl), [2](#)  
createCaseControlData, [2](#), [3](#)  
  
fitCaseControlModel, [3](#)  
  
getDbCaseData, [3](#), [7](#)  
getDbExposureData, [2](#), [5](#)  
  
insertDbPopulation, [5](#)  
  
loadCaseData, [6](#)  
  
saveCaseData, [7](#)  
selectControls, [5](#), [6](#), [7](#)