

Package ‘CohortMethod’

June 5, 2015

Type Package

Title New-user cohort method with large scale propensity and outcome models

Version 1.0.4

Date 2015-05-07

Author Martijn J. Schuemie [aut, cre],
Marc A. Suchard [aut],
Patrick B. Ryan [aut]

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

Description CohortMethod is an R package for performing new-user cohort studies in an observational database in the OMOP Common Data Model. It extracts the necessary data from a database in OMOP Common Data Model format, and uses a large set of covariates for both the propensity and outcome model, including for example all drugs, diagnoses, procedures, as well as age, comorbidity indexes, etc. Large scale regularized regression is used to fit the propensity and outcome models. Functions are included for trimming, stratifying and matching on propensity scores, as well as diagnostic functions, such as propensity score distribution plots and plots showing covariate balance before and after matching and/or trimming. Supported outcome models are (conditional) logistic regression, (conditional) Poisson regression, and (stratified) Cox regression.

License Apache License 2.0

VignetteBuilder knitr

Depends R (>= 3.1.0),
DatabaseConnector (>= 1.2.1),
Cyclops (>= 1.1.0)

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

Suggests testthat,
pROC,
gnm,
knitr,
rmarkdown,
OhdsiRTools

LinkingTo Rcpp

NeedsCompilation yes

R topics documented:

cohortDataSimulationProfile	2
CohortMethod	3
computeCovariateBalance	3
computePsAuc	4
constructEras	4
createCohortDataSimulationProfile	6
createPs	7
drawAttritionDiagram	8
fitOutcomeModel	8
getDbCohortData	9
getDbCovariates	16
getDbOutcomes	21
getOutcomeModel	22
getPsModel	23
grepCovariateNames	23
loadCohortData	24
matchOnPs	25
matchOnPsAndCovariates	26
plotCovariateBalanceOfTopVariables	27
plotCovariateBalanceScatterPlot	28
plotKaplanMeier	28
plotPs	29
saveCohortData	31
simulateCohortData	32
stratifyByPs	32
stratifyByPsAndCovariates	33
trimByPs	34
trimByPsToEquipoise	35
vignetteBalance	35
vignetteOutcomeModel1	36
vignetteOutcomeModel2	36
vignetteOutcomeModel3	36
vignettePs	36
Index	37

cohortDataSimulationProfile

A simulation profile

Description

A simulation profile

Usage

```
data(cohortDataSimulationProfile)
```

CohortMethod	<i>CohortMethod</i>
--------------	---------------------

Description

CohortMethod

computeCovariateBalance	<i>Compute covariate balance before and after matching and trimming</i>
-------------------------	---

Description

For every covariate, prevalence in treatment and comparator groups before and after matching/trimming are computed.

Usage

```
computeCovariateBalance(restrictedCohorts, cohortData,
  outcomeConceptId = NULL)
```

Arguments

restrictedCohorts
A data frame containing the people that are remaining after matching and/or trimming.

cohortData
An object of type cohortData as generated using getDbCohortData.

outcomeConceptId
The concept ID of the outcome. Persons marked for removal for the outcome will be removed when computing the balance before matching/trimming.

Details

The restrictedCohorts data frame should have at least the following columns:

rowId	(integer)	A unique identifier for each row (e.g. the person ID)
treatment	(integer)	Column indicating whether the person is in the treated (1) or comparator (0) group

Value

Returns a data frame describing the covariate balance before and after matching/trimming.

computePsAuc	<i>Compute the area under the ROC curve</i>
--------------	---

Description

computePsAuc computes the area under the ROC curve of the propensity score

Usage

```
computePsAuc(data, confidenceIntervals = FALSE)
```

Arguments

data	A data frame with at least the two columns described below
confidenceIntervals	Compute 95 percent confidence intervals (computationally expensive for large data sets)

Details

The data frame should have a least the following two columns:

treatment	(integer)	Column indicating whether the person is in the treated (1) or comparator (0) group
propensityScore	(real)	Propensity score

Value

A data frame holding the AUC and its 95 percent confidence interval

Examples

```
treatment = rep(0:1, each = 100)
propensityScore = c(rnorm(100, mean=0.4, sd=0.25), rnorm(100, mean=0.6, sd=0.25))
data <- data.frame(treatment = treatment, propensityScore = propensityScore)
data <- data[data$propensityScore > 0 & data$propensityScore < 1,]
computePsAuc(data)
```

constructEras	<i>Build eras</i>
---------------	-------------------

Description

Constructs eras (continuous periods of exposure or disease).

Usage

```
constructEras(connectionDetails, sourceDatabaseSchema,
  sourceTable = "drug_exposure",
  targetDatabaseSchema = sourceDatabaseSchema, targetTable = "drug_era",
  createTargetTable = FALSE, cdmDatabaseSchema = sourceDatabaseSchema,
  gracePeriod = 30, rollUp = TRUE, rollUpConceptClassId = "Ingredient",
  rollUpVocabularyId = "RxNorm", cdmVersion = "5")
```

Arguments

connectionDetails	An R object of type connectionDetails created using the function createConnectionDetails in the DatabaseConnector package.
sourceDatabaseSchema	The name of the database schema that contains the source table. Requires read permissions to this database. On SQL Server, this should specify both the database and the schema, so for example 'cdm_instance.dbo'.
sourceTable	The name of the source table.
targetDatabaseSchema	The name of the database schema that contains the target table. Requires write permissions to this database. On SQL Server, this should specify both the database and the schema, so for example 'cdm_instance.dbo'.
targetTable	The name of the target table.
createTargetTable	Should the target table be created? If not, the data is inserted in an existing table.
cdmDatabaseSchema	Only needed when rolling up concepts to ancestors: The name of the database schema that contains the vocabulary files. Requires read permissions to this database. On SQL Server, this should specify both the database and the schema, so for example 'cdm_instance.dbo'.
gracePeriod	The number of days allowed between periods for them to still be considered part of the same era.
rollUp	Should concepts be rolled up to their ancestors?
rollUpConceptClassId	The identifier of the concept class to which concepts should be rolled up.
rollUpVocabularyId	The identifier of the vocabulary to which concepts should be rolled up.
cdmVersion	The version of the CDM that is being used.

Details

This function creates eras from source data. For example, one could use this function to create drug eras based on drug exposures. The function allows drugs to be rolled up to ingredients, and prescriptions to the same ingredient that overlap in time are merged into a single ingredient. Note that stockpiling is not assumed to take place (ie. overlap is discarded), but a grace period can be specified allowing for a small gap between prescriptions when merging.

The user can specify the source and target table. These tables are assumed to have the same structure as the cohort table in the Common Data Model (CDM), except when the table names are "drug_exposure" or "condition_occurrence" for the source table, or "drug_era" or "condition_era"

for the target table, in which case the tables are assumed to have the structure defined for those tables in the CDM.

If both the source and target table specify a field for type_concept_id, the era construction will partition by the type_concept_id, in other words periods with different type_concept_ids will be treated independently.

Examples

```
## Not run:
#Constructing drug eras in CDM v4:
constructEras(connectionDetails,
               sourceDatabaseSchema = cdmDatabaseSchema,
               sourceTable = "drug_exposure",
               targetTable = "drug_era",
               createTargetTable = FALSE,
               gracePeriod = 30,
               rollUpVocabularyId = 8,
               rollUpConceptClassId = 'Ingredient',
               cdmVersion = "4")

#Constructing drug eras in CDM v5:
constructEras(connectionDetails,
               sourceDatabaseSchema = cdmDatabaseSchema,
               sourceTable = "drug_exposure",
               targetTable = "drug_era",
               createTargetTable = FALSE,
               gracePeriod = 30,
               rollUpVocabularyId = 'RxNorm',
               rollUpConceptClassId = 'Ingredient',
               cdmVersion = "5")

## End(Not run)
```

createCohortDataSimulationProfile

Create simulation profile

Description

createCohortDataSimulationProfile creates a profile based on the provided cohortData object, which can be used to generate simulated data that has similar characteristics.

Usage

```
createCohortDataSimulationProfile(cohortData)
```

Arguments

cohortData An object of type cohortData as generated using getDbCohortData.

Details

The output of this function is an object that can be used by the `simulateCohortData` function to generate a `cohortData` object.

Value

An object of type `cohortDataSimulationProfile`.

createPs	<i>Create propensity scores</i>
----------	---------------------------------

Description

`createPs` creates propensity scores using a regularized logistic regression.

Usage

```
createPs(cohortData, checkSorting = TRUE, outcomeConceptId = NULL,
  excludeCovariateIds = NULL, prior = createPrior("laplace", exclude = c(0),
  useCrossValidation = TRUE), control = createControl(noiseLevel = "silent",
  cvType = "auto", startingVariance = 0.1))
```

Arguments

<code>cohortData</code>	An object of type <code>cohortData</code> as generated using <code>getDbCohortData</code> .
<code>checkSorting</code>	Checks if the covariate data is sorted by <code>rowId</code> (necessary for fitting the model). Checking can be very time-consuming if the data is already sorted.
<code>outcomeConceptId</code>	The concept ID of the outcome. Persons marked for removal for the outcome will be removed prior to creating the propensity score model.
<code>excludeCovariateIds</code>	Exclude these covariates from the propensity model.
<code>prior</code>	The prior used to fit the model. See createPrior for details.
<code>control</code>	The control object used to control the cross-validation used to determine the hyperparameters of the prior (if applicable). See createControl for details.

Details

`createPs` creates propensity scores using a regularized logistic regression.

Examples

```
data(cohortDataSimulationProfile)
cohortData <- simulateCohortData(cohortDataSimulationProfile, n=1000)
ps <- createPs(cohortData)
```

drawAttritionDiagram	<i>Draw the attrition diagram</i>
----------------------	-----------------------------------

Description

drawAttritionDiagram draws the attrition diagram, showing how many people were excluded from the study population, and for what reasons.

Usage

```
drawAttritionDiagram(outcomeModel, treatmentLabel = "Treated",
  comparatorLabel = "Comparator", fileName = NULL)
```

Arguments

outcomeModel	An object of type outcomeModel as generated using the createOutcomeModel function.
treatmentLabel	A label to use for the treated cohort.
comparatorLabel	A label to use for the comparator cohort.
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.

Value

A ggplot object. Use the [ggsave](#) function to save to file in a different format.

fitOutcomeModel	<i>Create an outcome model, and compute the relative risk</i>
-----------------	---

Description

fitOutcomeModel creates an outcome model, and computes the relative risk

Usage

```
fitOutcomeModel(outcomeConceptId, cohortData, subPopulation = NULL,
  stratifiedCox = TRUE, riskWindowStart = 0, riskWindowEnd = 9999,
  addExposureDaysToEnd = FALSE, useCovariates = TRUE, fitModel = TRUE,
  modelType = "cox", prior = createPrior("laplace", useCrossValidation =
  TRUE), control = createControl(cvType = "auto", startingVariance = 0.1,
  selectorType = "byPid", noiseLevel = "quiet"))
```


Arguments

outcomeConceptId	The concept ID of the outcome. Persons marked for removal for the outcome will be removed prior to creating the outcome model.
cohortData	An object of type cohortData as generated using getDbCohortData.
subPopulation	A data frame specifying the (matched and/or trimmed) subpopulation to be used in the study, as well as their strata (for conditional models). This data frame should have at least a RowId, and a StratumId when including stratification.
stratifiedCox	Specifically for Cox regressions: specify whether to use the strata defined in subPopulation in the analysis. For Poisson regression and logistic regression, this is implied in 'clr' and 'cpr'.
riskWindowStart	The start of the risk window (in days) relative to the index data.
riskWindowEnd	The end of the risk window (in days) relative to the index data (+ days of exposure if the addExposureDaysToEnd parameter is specified).
addExposureDaysToEnd	Add the length of exposure the risk window?
useCovariates	Whether to use the covariate matrix in the cohortData in the outcome model.
fitModel	If false, the model will not be fit, and only summary statistics are available.
modelType	The type of model to be fitted. See details for options.
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

The model type can be one of these:

lr	Logistic regression
clr	Conditional logistic regression
cox	Cox regression (stratified or not, depending on whether stata is specified)
pr	Poisson regression
cpr	Conditional Poisson regression

Value

An object of class outcomeModel. Generic function summary, coef, and confint are available.

Examples

```
#todo
```

getDbCohortData

Get the cohort data from the server

Description

This function executes a large set of SQL statements against the database in OMOP CDM format to extract the data needed to perform the analysis.

Usage

```
getDbCohortData(connectionDetails, cdmDatabaseSchema,
  oracleTempSchema = cdmDatabaseSchema, targetDrugConceptId,
  comparatorDrugConceptId, indicationConceptIds = c(), washoutWindow = 183,
  indicationLookbackWindow = 183, studyStartDate = "", studyEndDate = "",
  exclusionConceptIds = c(), outcomeConceptIds,
  outcomeConditionTypeConceptIds = c(),
  exposureDatabaseSchema = cdmDatabaseSchema, exposureTable = "drug_era",
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "condition_occurrence", excludeDrugsFromCovariates = TRUE,
  useCovariateDemographics = TRUE, useCovariateDemographicsGender = TRUE,
  useCovariateDemographicsRace = TRUE,
  useCovariateDemographicsEthnicity = TRUE,
  useCovariateDemographicsAge = TRUE, useCovariateDemographicsYear = TRUE,
  useCovariateDemographicsMonth = TRUE,
  useCovariateConditionOccurrence = TRUE,
  useCovariateConditionOccurrence365d = TRUE,
  useCovariateConditionOccurrence30d = TRUE,
  useCovariateConditionOccurrenceInpt180d = TRUE,
  useCovariateConditionEra = FALSE, useCovariateConditionEraEver = FALSE,
  useCovariateConditionEraOverlap = FALSE,
  useCovariateConditionGroup = FALSE,
  useCovariateConditionGroupMeddra = FALSE,
  useCovariateConditionGroupSnomed = FALSE,
  useCovariateDrugExposure = FALSE, useCovariateDrugExposure365d = FALSE,
  useCovariateDrugExposure30d = FALSE, useCovariateDrugEra = FALSE,
  useCovariateDrugEra365d = FALSE, useCovariateDrugEra30d = FALSE,
  useCovariateDrugEraOverlap = FALSE, useCovariateDrugEraEver = FALSE,
  useCovariateDrugGroup = FALSE, useCovariateProcedureOccurrence = FALSE,
  useCovariateProcedureOccurrence365d = FALSE,
  useCovariateProcedureOccurrence30d = FALSE,
  useCovariateProcedureGroup = FALSE, useCovariateObservation = FALSE,
  useCovariateObservation365d = FALSE, useCovariateObservation30d = FALSE,
  useCovariateObservationBelow = FALSE,
  useCovariateObservationAbove = FALSE,
  useCovariateObservationCount365d = FALSE,
  useCovariateConceptCounts = FALSE, useCovariateRiskScores = FALSE,
  useCovariateRiskScoresCharlson = FALSE,
  useCovariateRiskScoresDCSI = FALSE, useCovariateRiskScoresCHADS2 = FALSE,
  useCovariateInteractionYear = FALSE, useCovariateInteractionMonth = FALSE,
  excludedCovariateConceptIds = c(), includedCovariateConceptIds = c(),
  deleteCovariatesSmallCount = 100)
```

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

For Oracle only: the name of the database schema where you want all temporary tables to be managed. Requires create/insert permissions to this database.

targetDrugConceptId

A unique identifier to define the target cohort. If `exposureTable = DRUG_ERA`, `targetDrugConceptId` is a `CONCEPT_ID` and all descendant concepts within that `CONCEPT_ID` will be used to define the cohort. If `exposureTable <> DRUG_ERA`, `targetDrugConceptId` is used to select the `COHORT_DEFINITION_ID` in the cohort-like table.

comparatorDrugConceptId

A unique identifier to define the comparator cohort. If `exposureTable = DRUG_ERA`, `comparatorDrugConceptId` is a `CONCEPT_ID` and all descendant concepts within that `CONCEPT_ID` will be used to define the cohort. If `exposureTable <> DRUG_ERA`, `comparatorDrugConceptId` is used to select the `COHORT_DEFINITION_ID` in the cohort-like table.

indicationConceptIds

A list of `CONCEPT_ID`s used to restrict the target and comparator cohorts, based on any descendant condition of this list occurring at least once within the `indicationLookbackWindow` prior to the cohort index date. If no concept IDs are specified, the cohorts are not restricted to any indication.

washoutWindow The minimum required continuous observation time prior to index date for a person to be included in the cohort.

indicationLookbackWindow

The window to look back prior to cohort index date to identify records of a indication condition. Only applicable if `indicationConceptIds != ''`.

studyStartDate A calendar date specifying the minimum date that a cohort index date can appear. Date format is `'yyyymmdd'`.

studyEndDate A calendar date specifying the maximum date that a cohort index date can appear. Date format is `'yyyymmdd'`.

exclusionConceptIds

A list of `CONCEPT_ID`s used to restrict the cohorts, based on any descendant conditions/drugs/procedures occurring at least once anytime prior to the cohort index date.

outcomeConceptIds

A list of `CONCEPT_ID`s 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.

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 by 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.
outcomeDatabaseSchema	The name of the database schema that is the location where the data used to define the outcome cohorts is available. If exposureTable = CONDITION_ERA, exposureDatabaseSchema is not used by assumed to be cdmSchema. Requires read permissions to this database.
outcomeTable	The tablename that contains the outcome cohorts. If outcomeTable <> CONDITION_OCCURRENCE, then expectation is outcomeTable has format of COHORT table: COHORT_DEFINITION_ID, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.
excludeDrugsFromCovariates	Should the target and comparator drugs (and their descendant concepts) be excluded from the covariates? Note that this will work if the drugs are actually drug concept IDs (and not cohort IDs).
useCovariateDemographics	A boolean value (TRUE/FALSE) to determine if demographic covariates (age in 5-yr increments, gender, race, ethnicity, year of index date, month of index date) will be created and included in future models.
useCovariateDemographicsGender	A boolean value (TRUE/FALSE) to determine if gender should be included in the model.
useCovariateDemographicsRace	A boolean value (TRUE/FALSE) to determine if race should be included in the model.
useCovariateDemographicsEthnicity	A boolean value (TRUE/FALSE) to determine if ethnicity should be included in the model.
useCovariateDemographicsAge	A boolean value (TRUE/FALSE) to determine if age (in 5 year increments) should be included in the model.
useCovariateDemographicsYear	A boolean value (TRUE/FALSE) to determine if calendar year should be included in the model.
useCovariateDemographicsMonth	A boolean value (TRUE/FALSE) to determine if calendar month should be included in the model.
useCovariateConditionOccurrence	A boolean value (TRUE/FALSE) to determine if covariates derived from CONDITION_OCCURRENCE table will be created and included in future models.
useCovariateConditionOccurrence365d	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition in 365d window prior to or on cohort index date. Only applicable if useCovariateConditionOccurrence = TRUE.

useCovariateConditionOccurrence30d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition in 30d window prior to or on cohort index date. Only applicable if useCovariateConditionOccurrence = TRUE.

useCovariateConditionOccurrenceInpt180d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition within inpatient type in 180d window prior to or on cohort index date. Only applicable if useCovariateConditionOccurrence = TRUE.

useCovariateConditionEra

A boolean value (TRUE/FALSE) to determine if covariates derived from CONDITION_ERA table will be created and included in future models.

useCovariateConditionEraEver

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition era anytime prior to or on cohort index date. Only applicable if useCovariateConditionEra = TRUE.

useCovariateConditionEraOverlap

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition era that overlaps the cohort index date. Only applicable if useCovariateConditionEra = TRUE.

useCovariateConditionGroup

A boolean value (TRUE/FALSE) to determine if all CONDITION_OCCURRENCE and CONDITION_ERA covariates should be aggregated or rolled-up to higher-level concepts based on vocabulary classification.

useCovariateConditionGroupMeddra

A boolean value (TRUE/FALSE) to determine if all CONDITION_OCCURRENCE and CONDITION_ERA covariates should be aggregated or rolled-up to higher-level concepts based on the MEDDRA classification.

useCovariateConditionGroupSnomed

A boolean value (TRUE/FALSE) to determine if all CONDITION_OCCURRENCE and CONDITION_ERA covariates should be aggregated or rolled-up to higher-level concepts based on the SNOMED classification.

useCovariateDrugExposure

A boolean value (TRUE/FALSE) to determine if covariates derived from DRUG_EXPOSURE table will be created and included in future models.

useCovariateDrugExposure365d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug in 365d window prior to or on cohort index date. Only applicable if useCovariateDrugExposure = TRUE.

useCovariateDrugExposure30d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug in 30d window prior to or on cohort index date. Only applicable if useCovariateDrugExposure = TRUE.

useCovariateDrugEra

A boolean value (TRUE/FALSE) to determine if covariates derived from DRUG_ERA table will be created and included in future models.

useCovariateDrugEra365d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug era in 365d window prior to or on cohort index date. Only applicable if useCovariateDrugEra = TRUE.

useCovariateDrugEra30d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug era in 30d window prior to or on cohort index date. Only applicable if useCovariateDrugEra = TRUE.

useCovariateDrugEraOverlap

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug era that overlaps the cohort index date. Only applicable if useCovariateDrugEra = TRUE.

useCovariateDrugEraEver

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug era anytime prior to or on cohort index date. Only applicable if useCovariateDrugEra = TRUE.

useCovariateDrugGroup

A boolean value (TRUE/FALSE) to determine if all DRUG_EXPOSURE and DRUG_ERA covariates should be aggregated or rolled-up to higher-level concepts of drug classes based on vocabulary classification.

useCovariateProcedureOccurrence

A boolean value (TRUE/FALSE) to determine if covariates derived from PROCEDURE_OCCURRENCE table will be created and included in future models.

useCovariateProcedureOccurrence365d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of procedure in 365d window prior to or on cohort index date. Only applicable if useCovariateProcedureOccurrence = TRUE.

useCovariateProcedureOccurrence30d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of procedure in 30d window prior to or on cohort index date. Only applicable if useCovariateProcedureOccurrence = TRUE.

useCovariateProcedureGroup

A boolean value (TRUE/FALSE) to determine if all PROCEDURE_OCCURRENCE covariates should be aggregated or rolled-up to higher-level concepts based on vocabulary classification.

useCovariateObservation

A boolean value (TRUE/FALSE) to determine if covariates derived from OBSERVATION table will be created and included in future models.

useCovariateObservation365d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of observation in 365d window prior to or on cohort index date. Only applicable if useCovariateObservation = TRUE.

useCovariateObservation30d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of observation in 30d window prior to or on cohort index date. Only applicable if useCovariateObservation = TRUE.

useCovariateObservationBelow

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of observation with a numeric value below normal range for latest value within 180d of cohort index. Only applicable if useCovariateObservation = TRUE.

useCovariateObservationAbove	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of observation with a numeric value above normal range for latest value within 180d of cohort index. Only applicable if useCovariateObservation = TRUE.
useCovariateObservationCount365d	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for the count of each observation concept in 365d window prior to or on cohort index date. Only applicable if useCovariateObservation = TRUE.
useCovariateConceptCounts	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that count the number of concepts that a person has within each domain (CONDITION, DRUG, PROCEDURE, OBSERVATION)
useCovariateRiskScores	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that calculate various Risk Scores, including Charlson, DCSI.
useCovariateRiskScoresCharlson	A boolean value (TRUE/FALSE) to determine if the Charlson comorbidity index should be included in the model.
useCovariateRiskScoresDCSI	A boolean value (TRUE/FALSE) to determine if the DCSI score should be included in the model.
useCovariateRiskScoresCHADS2	A boolean value (TRUE/FALSE) to determine if the CHADS2 score should be included in the model.
useCovariateInteractionYear	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that represent interaction terms between all other covariates and the year of the cohort index date.
useCovariateInteractionMonth	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that represent interaction terms between all other covariates and the month of the cohort index date.
excludedCovariateConceptIds	A list of concept IDs that should NOT be used to construct covariates.
includedCovariateConceptIds	A list of concept IDs that should be used to construct covariates.
deleteCovariatesSmallCount	A numeric value used to remove covariates that occur in both cohorts fewer than deleteCovariateSmallCounts time.

Details

Based on the parameters, the treatment and comparator cohorts are constructed. Baseline covariates at or before the index date are extracted, as well as outcomes occurring on or after the index date. The treatment and comparator cohorts can be identified using the drug_era table, or through user-defined cohorts in a cohort table either inside the CDM instance or in a separate schema. Similarly, outcomes are identified using the condition_occurrence or condition_era table, or through user-defined cohorts in a cohort table either inside the CDM instance or in a separate schema. Covariates are automatically extracted from the appropriate tables within the CDM. This function calls the getDbCovariates and getDbOutcomes functions.

Value

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

outcomes An `ffdf` object listing the outcomes per person, including the time to event, and the outcome concept ID. Outcomes are not yet filtered based on risk window, since this is done at a later stage.

cohorts An `ffdf` object listing the persons in each cohort, listing their exposure status as well as the time to the end of the observation period and time to the end of the cohort (usually the end of the exposure era).

covariates An `ffdf` object listing the baseline covariates per person in the two cohorts. This is done using a sparse representation: covariates with a value of 0 are omitted to save space.

exclude An `ffdf` object listing for each outcome concept ID the persons that need to be excluded from the analysis because of prior outcomes.

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

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

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

getDbCovariates

Get covariate information from the database

Description

Constructs a large set of covariates for one or more cohorts using data in the CDM schema.

Usage

```
getDbCovariates(connectionDetails = NULL, connection = NULL,
  oracleTempSchema = NULL, cdmDatabaseSchema,
  useExistingCohortPerson = TRUE, cohortDatabaseSchema = cdmDatabaseSchema,
  cohortTable = "cohort", cohortConceptIds = c(0, 1),
  useCovariateDemographics = TRUE, useCovariateDemographicsGender = TRUE,
  useCovariateDemographicsRace = TRUE,
  useCovariateDemographicsEthnicity = TRUE,
  useCovariateDemographicsAge = TRUE, useCovariateDemographicsYear = TRUE,
  useCovariateDemographicsMonth = TRUE,
  useCovariateConditionOccurrence = TRUE,
  useCovariateConditionOccurrence365d = TRUE,
  useCovariateConditionOccurrence30d = FALSE,
  useCovariateConditionOccurrenceInpt180d = FALSE,
  useCovariateConditionEra = FALSE, useCovariateConditionEraEver = FALSE,
  useCovariateConditionEraOverlap = FALSE,
  useCovariateConditionGroup = FALSE,
  useCovariateConditionGroupMeddra = FALSE,
  useCovariateConditionGroupSnomed = FALSE,
  useCovariateDrugExposure = FALSE, useCovariateDrugExposure365d = FALSE,
  useCovariateDrugExposure30d = FALSE, useCovariateDrugEra = FALSE,
```



```

useCovariateDrugEra365d = FALSE, useCovariateDrugEra30d = FALSE,
useCovariateDrugEraOverlap = FALSE, useCovariateDrugEraEver = FALSE,
useCovariateDrugGroup = FALSE, useCovariateProcedureOccurrence = FALSE,
useCovariateProcedureOccurrence365d = FALSE,
useCovariateProcedureOccurrence30d = FALSE,
useCovariateProcedureGroup = FALSE, useCovariateObservation = FALSE,
useCovariateObservation365d = FALSE, useCovariateObservation30d = FALSE,
useCovariateObservationBelow = FALSE,
useCovariateObservationAbove = FALSE,
useCovariateObservationCount365d = FALSE,
useCovariateConceptCounts = FALSE, useCovariateRiskScores = FALSE,
useCovariateRiskScoresCharlson = FALSE,
useCovariateRiskScoresDCSI = FALSE, useCovariateRiskScoresCHADS2 = FALSE,
useCovariateInteractionYear = FALSE, useCovariateInteractionMonth = FALSE,
excludedCovariateConceptIds = c(), includedCovariateConceptIds = c(),
deleteCovariatesSmallCount = 100)

```

Arguments

connectionDetails

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

connection

A connection to the server containing the schema as created using the `connect` function in the `DatabaseConnector` package.

oracleTempSchema

A schema where temp tables can be created in Oracle.

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'`.

useExistingCohortPerson

Does the temporary table `cohort_person` already exist? Can only be used when the `connection` parameter is not `NULL`.

cohortDatabaseSchema

If not using an existing `cohort_person` temp table, where is the source cohort table located? Note that on SQL Server, one should include both the database and schema, e.g. `"cdm_schema.dbo"`.

cohortTable

cohortConceptIds

If not using an existing `cohort_person` temp table, what is the name of the source cohort table?

useCovariateDemographics

A boolean value (TRUE/FALSE) to determine if demographic covariates (age in 5-yr increments, gender, race, ethnicity, year of index date, month of index date) will be created and included in future models.

useCovariateDemographicsGender

A boolean value (TRUE/FALSE) to determine if gender should be included in the model.

useCovariateDemographicsRace

A boolean value (TRUE/FALSE) to determine if race should be included in the model.

`useCovariateDemographicsEthnicity`
 A boolean value (TRUE/FALSE) to determine if ethnicity should be included in the model.

`useCovariateDemographicsAge`
 A boolean value (TRUE/FALSE) to determine if age (in 5 year increments) should be included in the model.

`useCovariateDemographicsYear`
 A boolean value (TRUE/FALSE) to determine if calendar year should be included in the model.

`useCovariateDemographicsMonth`
 A boolean value (TRUE/FALSE) to determine if calendar month should be included in the model.

`useCovariateConditionOccurrence`
 A boolean value (TRUE/FALSE) to determine if covariates derived from `CONDITION_OCCURRENCE` table will be created and included in future models.

`useCovariateConditionOccurrence365d`
 A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition in 365d window prior to or on cohort index date. Only applicable if `useCovariateConditionOccurrence = TRUE`.

`useCovariateConditionOccurrence30d`
 A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition in 30d window prior to or on cohort index date. Only applicable if `useCovariateConditionOccurrence = TRUE`.

`useCovariateConditionOccurrenceInpt180d`
 A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition within inpatient type in 180d window prior to or on cohort index date. Only applicable if `useCovariateConditionOccurrence = TRUE`.

`useCovariateConditionEra`
 A boolean value (TRUE/FALSE) to determine if covariates derived from `CONDITION_ERA` table will be created and included in future models.

`useCovariateConditionEraEver`
 A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition era anytime prior to or on cohort index date. Only applicable if `useCovariateConditionEra = TRUE`.

`useCovariateConditionEraOverlap`
 A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition era that overlaps the cohort index date. Only applicable if `useCovariateConditionEra = TRUE`.

`useCovariateConditionGroup`
 A boolean value (TRUE/FALSE) to determine if all `CONDITION_OCCURRENCE` and `CONDITION_ERA` covariates should be aggregated or rolled-up to higher-level concepts based on vocabulary classification.

`useCovariateConditionGroupMeddra`
 A boolean value (TRUE/FALSE) to determine if all `CONDITION_OCCURRENCE` and `CONDITION_ERA` covariates should be aggregated or rolled-up to higher-level concepts based on the MEDDRA classification.

`useCovariateConditionGroupSnomed`
 A boolean value (TRUE/FALSE) to determine if all `CONDITION_OCCURRENCE` and `CONDITION_ERA` covariates should be aggregated or rolled-up to higher-level concepts based on the SNOMED classification.

useCovariateDrugExposure	A boolean value (TRUE/FALSE) to determine if covariates derived from DRUG_EXPOSURE table will be created and included in future models.
useCovariateDrugExposure365d	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug in 365d window prior to or on cohort index date. Only applicable if useCovariateDrugExposure = TRUE.
useCovariateDrugExposure30d	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug in 30d window prior to or on cohort index date. Only applicable if useCovariateDrugExposure = TRUE.
useCovariateDrugEra	A boolean value (TRUE/FALSE) to determine if covariates derived from DRUG_ERA table will be created and included in future models.
useCovariateDrugEra365d	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug era in 365d window prior to or on cohort index date. Only applicable if useCovariateDrugEra = TRUE.
useCovariateDrugEra30d	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug era in 30d window prior to or on cohort index date. Only applicable if useCovariateDrugEra = TRUE.
useCovariateDrugEraOverlap	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug era that overlaps the cohort index date. Only applicable if useCovariateDrugEra = TRUE.
useCovariateDrugEraEver	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug era anytime prior to or on cohort index date. Only applicable if useCovariateDrugEra = TRUE.
useCovariateDrugGroup	A boolean value (TRUE/FALSE) to determine if all DRUG_EXPOSURE and DRUG_ERA covariates should be aggregated or rolled-up to higher-level concepts of drug classes based on vocabulary classification.
useCovariateProcedureOccurrence	A boolean value (TRUE/FALSE) to determine if covariates derived from PROCEDURE_OCCURRENCE table will be created and included in future models.
useCovariateProcedureOccurrence365d	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of procedure in 365d window prior to or on cohort index date. Only applicable if useCovariateProcedureOccurrence = TRUE.
useCovariateProcedureOccurrence30d	A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of procedure in 30d window prior to or on cohort index date. Only applicable if useCovariateProcedureOccurrence = TRUE.
useCovariateProcedureGroup	A boolean value (TRUE/FALSE) to determine if all PROCEDURE_OCCURRENCE covariates should be aggregated or rolled-up to higher-level concepts based on vocabulary classification.

useCovariateObservation

A boolean value (TRUE/FALSE) to determine if covariates derived from OBSERVATION table will be created and included in future models.

useCovariateObservation365d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of observation in 365d window prior to or on cohort index date. Only applicable if useCovariateObservation = TRUE.

useCovariateObservation30d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of observation in 30d window prior to or on cohort index date. Only applicable if useCovariateObservation = TRUE.

useCovariateObservationBelow

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of observation with a numeric value below normal range for latest value within 180d of cohort index. Only applicable if useCovariateObservation = TRUE.

useCovariateObservationAbove

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of observation with a numeric value above normal range for latest value within 180d of cohort index. Only applicable if useCovariateObservation = TRUE.

useCovariateObservationCount365d

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for the count of each observation concept in 365d window prior to or on cohort index date. Only applicable if useCovariateObservation = TRUE.

useCovariateConceptCounts

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that count the number of concepts that a person has within each domain (CONDITION, DRUG, PROCEDURE, OBSERVATION)

useCovariateRiskScores

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that calculate various Risk Scores, including Charlson, DCSI.

useCovariateRiskScoresCharlson

A boolean value (TRUE/FALSE) to determine if the Charlson comorbidity index should be included in the model.

useCovariateRiskScoresDCSI

A boolean value (TRUE/FALSE) to determine if the DCSI score should be included in the model.

useCovariateRiskScoresCHADS2

A boolean value (TRUE/FALSE) to determine if the CHADS2 score should be included in the model.

useCovariateInteractionYear

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that represent interaction terms between all other covariates and the year of the cohort index date.

useCovariateInteractionMonth

A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that represent interaction terms between all other covariates and the month of the cohort index date.

excludedCovariateConceptIds
A list of concept IDs that should NOT be used to construct covariates.

includedCovariateConceptIds
A list of concept IDs that should be used to construct covariates.

deleteCovariatesSmallCount
A numeric value used to remove covariates that occur in both cohorts fewer than deleteCovariateSmallCounts time.

Details

This function uses the data in the CDM to construct a large set of covariates for the provided cohorts. The cohorts are assumed to be in a table with the same structure as the cohort table in the OMOP CDM. The subject_id in this table must refer to person_ids in the CDM. One person can occur multiple times, but the combination of subject_id and cohort_start_date is assumed to be unique.

Value

Returns an object of type covariateData, containing information on the baseline covariates. Information about multiple outcomes can be captured at once for efficiency reasons. This object is a list with the following components:

covariates An ffdm object listing the baseline covariates per person in the two cohorts. This is done using a sparse representation: covariates with a value of 0 are omitted to save space.

covariateRef An ffdm object describing the covariates that have been extracted.

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

getDbOutcomes	<i>Get outcomes for persons in the cohorts</i>
---------------	--

Description

Gets the outcomes for the cohorts in the cohortData object.

Usage

```
getDbOutcomes(connectionDetails = NULL, connection = NULL,
  cdmDatabaseSchema, oracleTempSchema = cdmDatabaseSchema, cohortData,
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "condition_occurrence", outcomeConceptIds = "",
  outcomeConditionTypeConceptIds = "")
```

Arguments

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

connection
A connection to the server containing the schema as created using the connect function 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.
cohortData	An object of type cohortData as generated using getDbCohortData.
outcomeDatabaseSchema	The name of the database schema that is the location where the data used to define the outcome cohorts is available. If exposureTable = CONDITION_ERA, exposureDatabaseSchema is not used by assumed to be cdmSchema. Requires read permissions to this database.
outcomeTable	The tablename that contains the outcome cohorts. If outcomeTable <> CONDITION_OCCURRENCE, then expectation is outcomeTable has format of COHORT table: COHORT_DEFINITION_ID, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.
outcomeConceptIds	A list of CONCEPT_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.

Details

If the connection parameter is specified, the cohorts are already assumed to be on the server in the appropriate temp table. Else, the temp table will be created by loading the cohorts from the cohortData object to the server.

This function can be used to add additional outcomes to an existing cohortData object.

Value

The original cohortData object with the new outcome data added.

getOutcomeModel	<i>Get the outcome model</i>
-----------------	------------------------------

Description

getOutcomeModel shows the full outcome model, so showing the betas of all variables included in the outcome model, not just the treatment variable.

Usage

```
getOutcomeModel(outcomeModel, cohortData)
```

Arguments

outcomeModel	An object of type outcomeModel as generated using the createOutcomeModel function.
cohortData	An object of type cohortData as generated using getDbCohortData.

Details

Shows the coefficients and names of the covariates with non-zero coefficients.

Examples

```
#todo
```

getPsModel	<i>Get the propensity model</i>
------------	---------------------------------

Description

getPsModel shows the propensity score model

Usage

```
getPsModel(propensityScore, cohortData)
```

Arguments

propensityScore

The propensity scores as generated using the createPs function.

cohortData

An object of type cohortData as generated using getDbCohortData.

Details

Shows the coefficients and names of the covariates with non-zero coefficients.

Examples

```
#todo
```

grepCovariateNames	<i>Extract covariate names</i>
--------------------	--------------------------------

Description

Extracts covariate names using a regular-expression.

Usage

```
grepCovariateNames(pattern, object)
```

Arguments

pattern

A regular expression with which to name covariate names

object

An R object of type cohortData or covariateData.

Details

This function extracts covariate names that match a regular-expression for a cohortData or covariateData object.

Value

Returns a data.frame containing information about covariates that match a regular expression. This data.frame has the following columns:

covariateId Numerical identifier for use in model fitting using these covariates

covariateName Text identifier

analysisId Analysis identifier

conceptId OMOP common data model concept identifier, or 0

loadCohortData	<i>Load the cohort data from a folder</i>
----------------	---

Description

loadCohortData loads an object of type cohortData from a folder in the file system.

Usage

```
loadCohortData(file, readOnly = FALSE)
```

Arguments

file	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.

Examples

```
#todo
```

matchOnPs	<i>Match persons by propensity score</i>
-----------	--

Description

matchOnPs uses the provided propensity scores to match treated to comparator persons.

Usage

```
matchOnPs(data, caliper = 0.25, caliperScale = "standardized",
  maxRatio = 1, stratificationColumns = c())
```

Arguments

data	A data frame with the three columns described below.
caliper	The caliper for matching. A caliper is the distance which is acceptable for any match. Observations which are outside of the caliper are dropped. A caliper of 0 means no caliper is used.
caliperScale	The scale on which the caliper is defined. Two scales are supported: caliperScale = "propensity score" or caliperScale = "standardized". On the standardized scale, the caliper is interpreted in standard deviations of the propensity score distribution.
maxRatio	The maximum number of persons in the comparator arm to be matched to each person in the treatment arm. A maxRatio of 0 means no maximum: all comparators will be assigned to a treated person.
stratificationColumns	Names or numbers of one or more columns in the data data.frame on which subjects should be stratified prior to matching. No persons will be matched with persons outside of the strata identified by the values in these columns.

Details

The data frame should have at least the following three columns:

rowId	(integer)	A unique identifier for each row (e.g. the person ID)
treatment	(integer)	Column indicating whether the person is in the treated (1) or comparator (0) group
propensityScore	(real)	Propensity score

This function implements the greedy variable-ratio matching algorithm described in Rassen et al (2012).

Value

Returns a data frame with the same columns as the input data plus one extra column: stratumId. Any rows that could not be matched are removed

References

Rassen JA, Shelat AA, Myers J, Glynn RJ, Rothman KJ, Schneeweiss S. (2012) One-to-many propensity score matching in cohort studies, *Pharmacoepidemiology and Drug Safety*, May, 21 Suppl 2:69-80.

Examples

```
rowId = 1:5
treatment = c(1,0,1,0,1)
propensityScore = c(0,0.1,0.3,0.4,1)
age_group = c(1,1,1,1,1) #everyone in the same age group, so will not influence the matching
data <- data.frame(rowId = rowId, treatment = treatment, propensityScore = propensityScore,
                  age_group = age_group)
result <- matchOnPs(data, caliper = 0, maxRatio = 1, stratificationColumns = "age_group")
```

matchOnPsAndCovariates

Match by propensity score as well as other covariates

Description

matchOnPsAndCovariates uses the provided propensity scores and a set of covariates to match treated to comparator persons.

Usage

```
matchOnPsAndCovariates(data, caliper = 0.25, caliperScale = "standardized",
                      maxRatio = 1, cohortData, covariateIds)
```

Arguments

data	A data frame with the three columns described below.
caliper	The caliper for matching. A caliper is the distance which is acceptable for any match. Observations which are outside of the caliper are dropped. A caliper of 0 means no caliper is used.
caliperScale	The scale on which the caliper is defined. Two scales are supported: caliperScale = "propensity score" or caliperScale = "standardized". On the standardized scale, the caliper is interpreted in standard deviations of the propensity score distribution.
maxRatio	The maximum number of persons in the comparator arm to be matched to each person in the treatment arm. A maxRatio of 0 means no maximum: all comparators will be assigned to a treated person.
cohortData	An object of type cohortData as generated using getDbCohortData.
covariateIds	One or more covariate IDs in the cohortData object on which subjects should be also matched.

Details

The data frame should have at least the following three columns:

rowId	(integer)	A unique identifier for each row (e.g. the person ID)
treatment	(integer)	Column indicating whether the person is in the treated (1) or comparator (0) group
propensityScore	(real)	Propensity score

This function implements the greedy variable-ratio matching algorithm described in Rassen et al (2012).

Value

Returns a data frame with the same columns as the input data plus one extra column: stratumId. Any rows that could not be matched are removed

References

Rassen JA, Shelat AA, Myers J, Glynn RJ, Rothman KJ, Schneeweiss S. (2012) One-to-many propensity score matching in cohort studies, *Pharmacoepidemiology and Drug Safety*, May, 21 Suppl 2:69-80.

Examples

```
#todo
```

```
plotCovariateBalanceOfTopVariables
```

Plot variables with largest imbalance

Description

Create a plot showing those variables having the largest imbalance before matching, and those variables having the largest imbalance after matching. Requires running computeCovariateBalance first.

Usage

```
plotCovariateBalanceOfTopVariables(balance, n = 20, maxNameWidth = 100,
  fileName = NULL)
```

Arguments

balance	A data frame created by the computeCovariateBalance function.
n	Count of variates to plot.
maxNameWidth	Covariate names longer than this number of characters are truncated to create a nicer 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.

Value

A ggplot object. Use the [ggsave](#) function to save to file in a different format.

plotCovariateBalanceScatterPlot

Create a scatterplot of the covariate balance

Description

Create a scatterplot of the covariate balance, showing all variables with balance before and after matching on the x and y axis respectively. Requires running computeCovariateBalance first.

Usage

```
plotCovariateBalanceScatterPlot(balance, fileName = NULL)
```

Arguments

balance	A data frame created by the computeCovariateBalance function.
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.

Value

A ggplot object. Use the [ggsave](#) function to save to file in a different format.

plotKaplanMeier

Plot the Kaplan-Meier curve

Description

plotKaplanMeier creates the Kaplan-Meier survival plot

Usage

```
plotKaplanMeier(outcomeModel, censorMarks = FALSE,
  confidenceIntervals = TRUE, includeZero = TRUE, dataCutoff = 0.99,
  treatmentLabel = "Treated", comparatorLabel = "Comparator",
  fileName = NULL)
```

Arguments

outcomeModel	An object of type outcomeModel as generated using the fitOutcomeModel function.
censorMarks	Whether or not to include censor marks in the plot.
confidenceIntervals	Plot 95 percent confidence intervals?
includeZero	Should the y axis include zero, or only go down to the lowest observed survival?
dataCutoff	Fraction of the data (number censored) after which the graph will not be shown.
treatmentLabel	A label to us for the treated cohort.
comparatorLabel	A label to us for the comparator cohort.
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.

Value

A ggplot object. Use the [ggsave](#) function to save to file in a different format.

Examples

```
#todo
```

plotPs	<i>Plot the propensity score distribution</i>
--------	---

Description

plotPs shows the propensity (or preference) score distribution

Usage

```
plotPs(data, unfilteredData = NULL, scale = "preference",
       type = "density", binWidth = 0.05, fileName = NULL)
```

Arguments

data	A data frame with at least the two columns described below
unfilteredData	To be used when computing preference scores on data from which subjects have already been removed, e.g. through trimming and/or matching. This data frame should have the same structure as data.
scale	The scale of the graph. Two scales are supported: scale = "propensity" or scale = "preference". The preference score scale is defined by Walker et al (2013).
type	Type of plot. Two possible values: type = "density" or type = "histogram"
binWidth	For histograms, the width of the bins
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

The data frame should have a least the following two columns:

treatment	(integer)	Column indicating whether the person is in the treated (1) or comparator (0) group
propensityScore	(real)	Propensity score

Value

A ggplot object. Use the [ggsave](#) function to save to file in a different format.

References

Walker AM, Patrick AR, Lauer MS, Hornbrook MC, Marin MG, Platt R, Roger VL, Stang P, and Schneeweiss S. (2013) A tool for assessing the feasibility of comparative effectiveness research, *Comparative Effective Research*, 3, 11-20

Examples

```
treatment = rep(0:1, each = 100)
propensityScore = c(rnorm(100,mean=0.4, sd=0.25),rnorm(100,mean=0.6, sd=0.25))
data <- data.frame(treatment = treatment, propensityScore = propensityScore)
data <- data[data$propensityScore > 0 & data$propensityScore < 1,]
plotPs(data)
```

saveCohortData	<i>Save the cohort data to folder</i>
----------------	---------------------------------------

Description

saveCohortData saves an object of type cohortData to folder.

Usage

```
saveCohortData(cohortData, file)
```

Arguments

cohortData	An object of type cohortData as generated using getDbCohortData.
file	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 folder specified by the user.

Examples

```
#todo
```

simulateCohortData	<i>Generate simulated data</i>
--------------------	--------------------------------

Description

simulateCohortData creates a cohortData object with simulated data.

Usage

```
simulateCohortData(cohortDataSimulationProfile, n = 10000)
```

Arguments

cohortDataSimulationProfile	An object of type cohortDataSimulationProfile as generated using the createCohortDataSimulationProfile function.
n	The size of the population to be generated.

Details

This function generates simulated data that is in many ways similar to the original data on which the simulation profile is based. The contains same outcome, comparator, and outcome concept IDs, and the covariates and their 1st order statistics should be comparable.

Value

An object of type cohortData.

stratifyByPs	<i>Stratify persons by propensity score</i>
--------------	---

Description

stratifyByPs uses the provided propensity scores to stratify persons. Additional stratification variables for stratifications can also be used.

Usage

```
stratifyByPs(data, numberOfStrata = 5, stratificationColumns = c())
```

Arguments

data	A data frame with the three columns described below
numberOfStrata	How many strata? The boundaries of the strata are automatically defined to contain equal numbers of treated persons.
stratificationColumns	Names of one or more columns in the data data.frame on which subjects should also be stratified in addition to stratification on propensity score.

Details

The data frame should have the following three columns:

rowId	(integer)	A unique identifier for each row (e.g. the person ID)
treatment	(integer)	Column indicating whether the person is in the treated (1) or comparator (0) group
propensityScore	(real)	Propensity score

Value

Returns a data frame with the same columns as the input data plus one extra column: stratumId.

Examples

```
rowId = 1:200
treatment = rep(0:1, each = 100)
propensityScore = c(runif(100,min=0,max=1),runif(100,min=0,max=1))
data <- data.frame(rowId = rowId, treatment = treatment, propensityScore = propensityScore)
result <- stratifyByPs(data,5)
```

stratifyByPsAndCovariates

Stratify persons by propensity score and other covariates

Description

stratifyByPsAndCovariates uses the provided propensity scores and covariatesto stratify persons.

Usage

```
stratifyByPsAndCovariates(data, numberOfStrata = 5, cohortData, covariateIds)
```

Arguments

data	A data frame with the three columns described below
numberOfStrata	Into how many strata should the propensity score be divided? The boundaries of the strata are automatically defined to contain equal numbers of treated persons.
cohortData	An object of type cohortData as generated using getDbCohortData.
covariateIds	One or more covariate IDs in the cohortData object on which subjects should also be stratified.

Details

The data frame should have the following three columns:

rowId	(integer)	A unique identifier for each row (e.g. the person ID)
treatment	(integer)	Column indicating whether the person is in the treated (1) or comparator (0) group
propensityScore	(real)	Propensity score

Value

Returns a data frame with the same columns as the input data plus one extra column: stratumId.

Examples

```
#todo
```

trimByPs	<i>Trim persons by propensity score</i>
----------	---

Description

trimByPs uses the provided propensity scores to trim subjects with extreme scores.

Usage

```
trimByPs(data, trimFraction = 0.05)
```

Arguments

data	A data frame with the three columns described below
trimFraction	This fraction will be removed from each treatment group. In the treatment group, persons with the highest propensity scores will be removed, in the comparator group person with the lowest scores will be removed.

Details

The data frame should have the following three columns:

rowId	(integer)	A unique identifier for each row (e.g. the person ID)
treatment	(integer)	Column indicating whether the person is in the treated (1) or comparator (0) group
propensityScore	(real)	Propensity score

Value

Returns a data frame with the same three columns as the input.

Examples

```
rowId = 1:2000
treatment = rep(0:1, each = 1000)
propensityScore = c(runif(1000, min = 0, max = 1), runif(1000, min = 0, max = 1))
data <- data.frame(rowId = rowId, treatment = treatment, propensityScore = propensityScore)
result <- trimByPs(data, 0.05)
```

trimByPsToEquipoise	<i>Keep only persons in clinical equipoise</i>
---------------------	--

Description

trimByPsToEquipoise uses the preference score to trim subjects that are not in clinical equipoise

Usage

```
trimByPsToEquipoise(data, bounds = c(0.25, 0.75))
```

Arguments

data	A data frame with at least the three columns described below
bounds	The upper and lower bound on the preference score for keeping persons

Details

The data frame should have the following three columns:

rowId	(integer)	A unique identifier for each row (e.g. the person ID)
treatment	(integer)	Column indicating whether the person is in the treated (1) or comparator (0) group
propensityScore	(real)	Propensity score

Value

Returns a data frame with the same three columns as the input.

References

Walker AM, Patrick AR, Lauer MS, Hornbrook MC, Marin MG, Platt R, Roger VL, Stang P, and Schneeweiss S. (2013) A tool for assessing the feasibility of comparative effectiveness research, *Comparative Effective Research*, 3, 11-20

Examples

```
rowId = 1:2000
treatment = rep(0:1, each = 1000)
propensityScore = c(runif(1000,min=0,max=1),runif(1000,min=0,max=1))
data <- data.frame(rowId = rowId, treatment = treatment, propensityScore = propensityScore)
result <- trimByPsToEquipoise(data)
```

vignetteBalance	<i>Balance data for the vignette</i>
-----------------	--------------------------------------

Description

Balance data for the vignette

Usage

data(vignetteBalance)

vignetteOutcomeModel1 *Outcome data for the vignette*

Description

Outcome data for the vignette

Usage

data(vignetteOutcomeModel1)

vignetteOutcomeModel2 *Outcome data for the vignette*

Description

Outcome data for the vignette

Usage

data(vignetteOutcomeModel2)

vignetteOutcomeModel3 *Outcome data for the vignette*

Description

Outcome data for the vignette

Usage

data(vignetteOutcomeModel3)

vignettePs *Propensity scores for the vignette*

Description

Propensity scores for the vignette

Usage

data(vignettePs)

Index

*Topic **datasets**

- cohortDataSimulationProfile, [2](#)
- vignetteBalance, [35](#)
- vignetteOutcomeModel1, [36](#)
- vignetteOutcomeModel2, [36](#)
- vignetteOutcomeModel3, [36](#)
- vignettePs, [36](#)

cohortDataSimulationProfile, [2](#)

CohortMethod, [3](#)

CohortMethod-package (CohortMethod), [3](#)

computeCovariateBalance, [3](#)

computePsAuc, [4](#)

constructEras, [4](#)

createCohortDataSimulationProfile, [6](#)

createControl, [7](#), [9](#)

createPrior, [7](#), [9](#)

createPs, [7](#)

drawAttritionDiagram, [8](#)

fitOutcomeModel, [8](#)

getDbCohortData, [9](#)

getDbCovariates, [16](#)

getDbOutcomes, [21](#)

getOutcomeModel, [22](#)

getPsModel, [23](#)

ggsave, [8](#), [28](#), [29](#), [31](#)

grepCovariateNames, [23](#)

loadCohortData, [24](#)

matchOnPs, [25](#)

matchOnPsAndCovariates, [26](#)

plotCovariateBalanceOfTopVariables, [27](#)

plotCovariateBalanceScatterPlot, [28](#)

plotKaplanMeier, [28](#)

plotPs, [29](#)

saveCohortData, [31](#)

simulateCohortData, [32](#)

stratifyByPs, [32](#)

stratifyByPsAndCovariates, [33](#)

trimByPs, [34](#)

trimByPsToEquipoise, [35](#)

vignetteBalance, [35](#)

vignetteOutcomeModel1, [36](#)

vignetteOutcomeModel2, [36](#)

vignetteOutcomeModel3, [36](#)

vignettePs, [36](#)