

# Package ‘RiskStratifiedEstimation’

May 25, 2020

**Type** Package

**Title** Risk Stratified Effect Estimation

**Version** 0.9.0

**Author** Alexandros Rekkas

**Maintainer** Alexandros Rekkas <a.rekkas@erasmusmc.nl>

**Description** RiskStratifiedEstimation is an R package for implementing risk stratified analyses of treatment effect in an observational database in the OMOP Common Data Model. It combines functionality of PatientLevelPrediction and CohortMethod R packages to extract and analyze data from a database in the OMOP Common Data Model format. Analyses are performed in two distinct steps. First, a prediction step is implemented, where personalized risks are derived from a large set of covariates applying one of various possible prediction methods. In the second step, risk stratified treatment comparisons are performed. Large scale regularized regression is used to fit the propensity model. Inverse probability of treatment weights are calculated and truncated within strata of predicted risk. Only weighted Cox regression models are supported for the outcome model.

**License** Apache License 2.0

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.0

**Depends** R (>= 3.2.2),  
DatabaseConnector (>= 2.2.0),  
Cyclops,  
FeatureExtraction (>= 2.2.3)

**Imports** PatientLevelPrediction,  
DatabaseConnector,  
Cyclops,  
FeatureExtraction,  
CohortMethod,  
ParallelLogger,  
SqlRender,  
foreach,  
dplyr,  
ff,  
shiny,  
reshape,  
caret,  
ffbase,

ggplot2,  
gridExtra,  
survival,  
tidyr,  
doSNOW,  
formatR,  
ggpubr,  
survminer

**Suggests** knitr,  
rmarkdown

**VignetteBuilder** knitr

## R topics documented:

absoluteRiskReduction . . . . .	3
createAnalysisSettings . . . . .	3
createCreateIPWArgs . . . . .	4
createCreatePsArgs . . . . .	5
createDatabaseSettings . . . . .	6
createGetCmDataArgs . . . . .	7
createGetCovariateSettings . . . . .	8
createGetDataSettings . . . . .	9
createGetPlpDataArgs . . . . .	9
createIPW . . . . .	10
createMatchOnPsArgs . . . . .	11
createOverallResults . . . . .	11
createPopulationCmSettingsArgs . . . . .	12
createPopulationPlpSettingsArgs . . . . .	13
createPopulationSettings . . . . .	15
createRunCmSettingsArgs . . . . .	15
createRunPlpSettingsArgs . . . . .	16
createRunSettings . . . . .	18
createStratifyByPsArgs . . . . .	18
estimateTreatmentEffect . . . . .	19
fitOutcomeModels . . . . .	20
fitOutcomeModelsOverall . . . . .	20
fitPsModel . . . . .	21
fitPsModelOverall . . . . .	22
fitPsModelSwitch . . . . .	22
outcomeModelWeighted . . . . .	23
prepareForPlpData . . . . .	24
relativeRiskReduction . . . . .	25
runRiskStratifiedEstimation . . . . .	25
stratifiedKaplanMeier . . . . .	26

**Index**

**27**

---

absoluteRiskReduction *Absolute risk reduction*

---

### Description

Calculates absolute risk reduction based on the Kaplan-Meier estimates within risk strata

### Usage

```
absoluteRiskReduction(population, timePoint, psMethod)
```

### Arguments

population	The study population generated by <a href="#">matchOnPs</a> when using propensity score matching or by <a href="#">stratifyByPs</a> when stratifying on the propensity score. In case of inverse probability of treatment weighting approach, it is a dataframe with a weights column.
timePoint	The time at which the absolute risk difference is estimated
psMethod	Can be one of "matchOnPs", "stratifyByPs" or "inversePtWeighted".

### Value

A dataframe with the absolute risk-stratum specific absolute risk difference estimates, along with 95 percent confidence interval.

---

createAnalysisSettings  
*Create a parameter defining the performed risk stratified analysis*

---

### Description

Create a parameter defining the performed risk stratified analysis

### Usage

```
createAnalysisSettings(
  analysisId = NULL,
  databaseName,
  analysisType,
  treatmentCohortId,
  comparatorCohortId,
  outcomeIds,
  analysisMatrix = diag(length(outcomeIds)),
  mapTreatments,
  mapOutcomes,
  verbosity = NULL,
  saveDirectory = NULL
)
```

**Arguments**

analysisId	The analysis ID.
databaseName	The name of the database.
analysisType	The type of the analysis. Could be "matching", "stratifyByPs" or "inversePtWeighted".
treatmentCohortId	The cohort definition id of the treatment cohort in the cohortTable.
comparatorCohortId	The cohort definition id of the comparator cohort in the cohortTable.
outcomeIds	The cohort definition ids of the outcome cohorts in the outcomeTable.
analysisMatrix	Boolean matrix defining the outcomes to be assessed (rows) within risk strata (columns). The order in columns should match the the order of outcomeIds. Default is the diagonal matrix, which leads to the risk stratified assessment of only the outcome for which the risk strata were defined.
mapTreatments	Dataframe containing 2 columns: *idNumber* with the id numbers of the treatment and comparator cohorts and *label* the cohort names.
mapOutcomes	Dataframe containing 2 columns: *idNumber* with the cohort names of the outcomes of interest and *label* with their names.
verbosity	Sets the level of the verbosity. If the log level is at or higher in priority than the logger threshold, a message will print. The levels are: <ul style="list-style-type: none"> <li>• DEBUGHighest verbosity showing all debug statements</li> <li>• TRACEShowing information about start and end of steps</li> <li>• INFOShow informative information (Default)</li> <li>• WARNShow warning messages</li> <li>• ERRORShow error messages</li> <li>• FATALBe silent except for fatal errors</li> </ul>
saveDirectory	The directory name where the results of the analyses will be stored.

**Value**

An analysisSettings object providing the identification information of the analysis.

---

createCreateIPWArgs	<i>Create a parameter object for the function createIPW Create an object defining the parameter values.</i>
---------------------	---

---

**Description**

Create a parameter object for the function createIPW Create an object defining the parameter values.

**Usage**

```
createCreateIPWArgs(
  weightsType = "ATE",
  useStabilizedWeights = TRUE,
  truncationLevels = c(0.01, 0.99)
)
```

**Arguments**

weightsType	The type of the weights to be used. Allowed options are 'ATE' for average treatment effect and 'ATT' for average treatment effect on the treated weights
useStabilizedWeights	Should stabilized weights be used?
truncationLevels	The level of truncation expressed in percentiles of the propensity score.

---

createCreatePsArgs	<i>Create a parameter object for the function createPs</i>
--------------------	--

---

**Description**

Create a parameter object for the function createPs

**Usage**

```
createCreatePsArgs(
  excludeCovariateIds = c(),
  includeCovariateIds = c(),
  maxCohortSizeForFitting = 250000,
  errorOnHighCorrelation = TRUE,
  stopOnError = TRUE,
  prior = createPrior("laplace", exclude = c(0), useCrossValidation = TRUE),
  control = createControl(noiseLevel = "silent", cvType = "auto", tolerance = 2e-07,
    cvRepetitions = 10, startingVariance = 0.01)
)
```

**Arguments**

excludeCovariateIds	Exclude these covariates from the propensity model.
includeCovariateIds	Include only these covariates in the propensity model.
maxCohortSizeForFitting	If the target or comparator cohort are larger than this number, they will be down-sampled before fitting the propensity model. The model will be used to compute propensity scores for all subjects. The purpose of the sampling is to gain speed. Setting this number to 0 means no downsampling will be applied.
errorOnHighCorrelation	If true, the function will test each covariate for correlation with the treatment assignment. If any covariate has an unusually high correlation (either positive or negative), this will throw an error.
stopOnError	If an error occurs, should the function stop? Else, the two cohorts will be assumed to be perfectly separable.
prior	The prior used to fit the model. See createPrior for details.
control	The control object used to control the cross-validation used to determine the hyperparameters of the prior (if applicable). See createControl for details.

**Details**

Create an object defining the parameter values.

**Value**

A parameter object for creating propensity scores.

---

```
createDatabaseSettings
```

*Create parameter object for database to be reached*

---

**Description**

Create parameter object for database to be reached

**Usage**

```
createDatabaseSettings(  
  cdmVersion = "5",  
  cdmDatabaseSchema,  
  cohortDatabaseSchema = cdmDatabaseSchema,  
  outcomeDatabaseSchema = cdmDatabaseSchema,  
  resultsDatabaseSchema = cdmDatabaseSchema,  
  exposureDatabaseSchema = cdmDatabaseSchema,  
  cohortTable,  
  outcomeTable,  
  exposureTable = "drug_era",  
  mergedCohortTable,  
  targetCohortId = 1  
)
```

**Arguments**

cdmVersion      Define the OMOP CDM version

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"

cohortDatabaseSchema

The name of the database schema that is the location where the cohort data used to define the at risk cohort is available. If cohortTable = DRUG\_ERA, cohortDatabaseSchema is not used by assumed to be cdmDatabaseSchema.

outcomeDatabaseSchema

The name of the database schema that is the location where the data used to define the outcome cohorts is available. If cohortTable = CONDITION\_ERA, exposureDatabaseSchema is not used by assumed to be cdmSchema. Requires read permissions to this database.

resultsDatabaseSchema

The name of the database schema with write permissions.

exposureDatabaseSchema	Input of function <a href="#">getDbCohortMethodData</a> : The name of the database schema that is the location where the exposure data used to define the exposure cohorts is available.
cohortTable	The tablename that contains the at risk cohort. If cohortTable <> DRUG_ERA, then expectation is cohortTable has format of COHORT table: cohort_concept_id, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.
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.
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.
mergedCohortTable	The name of the table where the merged treatment and comparator cohorts will be stored.
targetCohortId	The cohort definition id of of the merged cohort in the mergedCohortTable.

---

createGetCmDataArgs	<i>Create a parameter object for the function <a href="#">getDbCohortMethodData</a></i>
---------------------	---

---

## Description

Create a parameter object for the function [getDbCohortMethodData](#)

## Usage

```
createGetCmDataArgs(
  studyStartDate = "",
  studyEndDate = "",
  excludeDrugsFromCovariates = TRUE,
  firstExposureOnly = FALSE,
  removeDuplicateSubjects = FALSE,
  restrictToCommonPeriod = FALSE,
  washoutPeriod = 0,
  maxCohortSize = 0
)
```

## Arguments

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'. Important: the study end data is also used to truncate risk windows, meaning no outcomes beyond the study end date will be considered.
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 cohortIDs).

firstExposureOnly	Should only the first exposure per subject be included? Note that this is typically done in the createStudyPopulationfunction, but can already be done here for efficiency reasons.
removeDuplicateSubjects	Remove subjects that are in both the target and comparator cohort? See details for allowed values. Note that this is typically done in the createStudyPopulation function, but can already be done here for efficiency reasons.
restrictToCommonPeriod	Restrict the analysis to the period when both treatments are observed?
washoutPeriod	The minimum required continuous observation time prior to index date for a person to be included in the cohort. Note that this is typically done in the createStudyPopulation function, but can already be done here for efficiency reasons.
maxCohortSize	If either the target or the comparator cohort is larger than this number it will be sampled to this size. maxCohortSize = 0 indicates no maximum size.

## Details

Create an object defining the parameter values.

---

createGetCovariateSettings

*Create the parameter object for extracting the covariates Contains the arguments for the extraction of both the covariates related to the prediction step and the estimation step.*

---

## Description

Create the parameter object for extracting the covariates Contains the arguments for the extraction of both the covariates related to the prediction step and the estimation step.

## Usage

```
createGetCovariateSettings(
  covariateSettingsCm = FeatureExtraction::createCovariateSettings(),
  covariateSettingsPlp = FeatureExtraction::createCovariateSettings()
)
```

## Arguments

covariateSettingsCm

The covariate settings object related to the estimation step created from [createCovariateSettings](#).

covariateSettingsPlp

The covariate settings object related to the prediction step created from [createCovariateSettings](#).



---

createGetDataSettings    *Create parameter object for extracting the data*

---

### Description

Create parameter object for extracting the data

### Usage

```
createGetDataSettings(  
  getPlpDataSettings = createGetPlpDataArgs(),  
  getCmDataSettings = createGetCmDataArgs(),  
  plpDataFolder = NULL,  
  cohortMethodDataFolder = NULL  
)
```

### Arguments

getPlpDataSettings    Parameter object for the extraction of the plpData object created from [createGetPlpDataArgs](#).

getCmDataSettings    Parameter object for the extraction of the cohortMethodData object created from [createGetPlpDataArgs](#).

plpDataFolder    The directory path where the plpData object has already been saved locally.

cohortMethodDataFolder    The directory path where the cohortMethodData object has already been saved locally.

---

createGetPlpDataArgs    *Create a parameter object for the function getPlpData*

---

### Description

Create a parameter object for the function getPlpData

### Usage

```
createGetPlpDataArgs(  
  studyStartDate = "",  
  studyEndDate = "",  
  firstExposureOnly = FALSE,  
  washoutPeriod = 0,  
  excludeDrugsFromCovariates = FALSE  
)
```

**Arguments**

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'. Important: the studyend data is also used to truncate risk windows, meaning no outcomes beyond the study end date will be considered.
firstExposureOnly	Should only the first exposure per subject be included? Note that this is typically done in the createStudyPopulation function, but can already be done here for efficiency reasons.
washoutPeriod	The minimum required continuous observation time prior to index date for a person to be included in the at risk cohort. Note that this is typically done in the createStudyPopulation function, but can already be done here for efficiency reasons.
excludeDrugsFromCovariates	A redundant option

**Details**

Create an object defining the parameter values.

---

createIPW	<i>Creates Inverse Probability Weights</i>
-----------	--

---

**Description**

Calculates inverse probability weights based on the propensity score

**Usage**

```
createIPW(
  ps,
  weightsType = "ATE",
  useStabilizedWeights = TRUE,
  truncationLevels = c(0.01, 0.99)
)
```

**Arguments**

ps	A propensity score data frame as created from <a href="#">createPs</a>
weightsType	The type of the weights to be used. Allowed options are 'ATE' for average treatment effect and 'ATT' for average treatment effect on the treated weights
useStabilizedWeights	Should stabilized weights be used?
truncationLevels	The level of truncation expressed in percentiles of the propensity score.

**Value**

The ps data frame provided as input along with a weights column

---

createMatchOnPsArgs     *Create a parameter object for the function matchOnPs*

---

## Description

Create a parameter object for the function matchOnPs

## Usage

```
createMatchOnPsArgs(
  caliper = 0.2,
  caliperScale = "standardized logit",
  maxRatio = 1,
  stratificationColumns = c()
)
```

## Arguments

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. Three scales are supported: caliperScale = 'propensity score', caliperScale = 'standardized', or caliperScale = 'standardized logit'. On the standardized scale, the caliper is interpreted in standard deviations of the propensity score distribution. 'standardized logit' is similar, except that the propensity score is transformed to the logit scale because the PS is more likely to be normally distributed on that scale (Austin, 2011).
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 target person.
stratificationColumns	Names or numbers of one or more columns in the 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

Create an object defining the parameter values.

---

createOverallResults     *Combines the overall results*

---

## Description

Combines the overall results

## Usage

```
createOverallResults(analysisSettings)
```

**Arguments**

- analysisSettings      An R object of type analysisSettings created using the function [createAnalysisSettings](#).
- runSettings          An R object of type runSettings created using the function [createRunSettings](#).

**Value**

Stores the overall results along with the required data to launch the shiny application in the ‘shiny’ directory

---

```
createPopulationCmSettingsArgs
```

*Create a parameter object for the function createStudyPopulation*

---

**Description**

Create a parameter object for the function createStudyPopulation

**Usage**

```
createPopulationCmSettingsArgs(
  firstExposureOnly = FALSE,
  restrictToCommonPeriod = FALSE,
  washoutPeriod = 0,
  removeDuplicateSubjects = FALSE,
  removeSubjectsWithPriorOutcome = TRUE,
  priorOutcomeLookback = 99999,
  minDaysAtRisk = 1,
  riskWindowStart = 0,
  addExposureDaysToStart = NULL,
  startAnchor = "cohort start",
  riskWindowEnd = 0,
  addExposureDaysToEnd = NULL,
  endAnchor = "cohort end",
  censorAtNewRiskWindow = FALSE
)
```

**Arguments**

- firstExposureOnly      Should only the first exposure per subject be included? Note that this is typically done in the createStudyPopulation function,
- restrictToCommonPeriod      Restrict the analysis to the period when both exposures are observed?
- washoutPeriod      The minimum required continuous observation time prior to index date for a person to be included in the cohort.
- removeDuplicateSubjects      Remove subjects that are in both the target and comparator cohort? See details for allowed values.

removeSubjectsWithPriorOutcome	Remove subjects that have the outcome prior to the riskwindow start?
priorOutcomeLookback	How many days should we look back when identifying prioroutcomes?
minDaysAtRisk	The minimum required number of days at risk.
riskWindowStart	The start of the risk window (in days) relative to the startAnchor.
addExposureDaysToStart	DEPRECATED: Add the length of exposure the start of the risk window?Use startAnchor instead.
startAnchor	The anchor point for the start of the risk window. Can be "cohort start"or "cohort end".
riskWindowEnd	The end of the risk window (in days) relative to the endAnchor.
addExposureDaysToEnd	DEPRECATED: Add the length of exposure the risk window?Use endAnchor instead.
endAnchor	The anchor point for the end of the risk window. Can be "cohort start"or "cohort end".
censorAtNewRiskWindow	If a subject is in multiple cohorts, should time-at-risk be censoredwhen the new time-at-risk starts to prevent overlap?

## Details

Create an object defining the parameter values.

---

```
createPopulationPlpSettingsArgs
```

*Create a parameter object for the function createStudyPopulation*

---

## Description

Create a parameter object for the function createStudyPopulation

## Usage

```
createPopulationPlpSettingsArgs(
  binary = T,
  includeAllOutcomes = T,
  firstExposureOnly = FALSE,
  washoutPeriod = 0,
  removeSubjectsWithPriorOutcome = TRUE,
  priorOutcomeLookback = 99999,
  requireTimeAtRisk = F,
  minTimeAtRisk = 365,
  riskWindowStart = 0,
  startAnchor = "cohort start",
  addExposureDaysToStart = NULL,
  riskWindowEnd = 365,
```

```

    endAnchor = "cohort start",
    addExposureDaysToEnd = NULL,
    verbosity = "INFO"
)

```

## Arguments

<code>binary</code>	Forces the outcomeCount to be 0 or 1 (use for binary prediction problems)
<code>includeAllOutcomes</code>	(binary) indicating whether to include people with outcomes who are not observed for the whole at risk period
<code>firstExposureOnly</code>	Should only the first exposure per subject be included? Note that this is typically done in the createStudyPopulation function,
<code>washoutPeriod</code>	The minimum required continuous observation time prior to indexdate for a person to be included in the cohort.
<code>removeSubjectsWithPriorOutcome</code>	Remove subjects that have the outcome prior to the risk window start?
<code>priorOutcomeLookback</code>	How many days should we look back when identifying prior outcomes?
<code>requireTimeAtRisk</code>	Should subject without time at risk be removed?
<code>minTimeAtRisk</code>	The minimum number of days at risk required to be included
<code>riskWindowStart</code>	The start of the risk window (in days) relative to the startAnchor.
<code>startAnchor</code>	The anchor point for the start of the risk window. Can be "cohort start" or "cohort end".
<code>addExposureDaysToStart</code>	DEPRECATED: Add the length of exposure the start of the risk window? Use startAnchor instead.
<code>riskWindowEnd</code>	The end of the risk window (in days) relative to the endAnchor parameter
<code>endAnchor</code>	The anchor point for the end of the risk window. Can be "cohort start" or "cohort end".
<code>addExposureDaysToEnd</code>	DEPRECATED: Add the length of exposure the risk window? Use endAnchor instead.
<code>verbosity</code>	Sets the level of the verbosity. If the log level is at or higher in priority than the logger threshold, a message will print. The levels are: DEBUG Highest verbosity showing all debug statements TRACE Showing information about start and end of steps INFO Show informative information (Default) WARN Show warning messages ERROR Show error messages FATAL Be silent except for fatal errors

## Details

Create an object defining the parameter values.

---

```
createPopulationSettings
```

*Create parameter object for defining the analysis populations Contains the settings for defining both the populationPlp and the populationCm objects.*

---

## Description

Create parameter object for defining the analysis populations Contains the settings for defining both the populationPlp and the populationCm objects.

## Usage

```
createPopulationSettings(
  populationPlpSettings = createPopulationPlpSettingsArgs(),
  populationCmSettings = createPopulationCmSettingsArgs()
)
```

## Arguments

populationPlpSettings

Parameter object for the definition of the populationPlp object created from [createPopulationPlpSettingsArgs](#).

populationCmSettings

Parameter object for the definition of the populationCm object created from [createPopulationCmSettingsArgs](#).

---

```
createRunCmSettingsArgs
```

*Create a parameter object for running the estimation step Create a parameter object for running the estimation step. This function is used to create part of the input of [createRunSettings](#).*

---

## Description

Create a parameter object for running the estimation step Create a parameter object for running the estimation step. This function is used to create part of the input of [createRunSettings](#).

## Usage

```
createRunCmSettingsArgs(
  psMethod = "inversePtWeighted",
  effectEstimationSettings = createCreateIPWArgs(),
  psSettings = createCreatePsArgs(),
  createPsThreads = 1,
  fitOutcomeModelsThreads = 1,
  createPsThreadsNegativeControls = 1,
  estimateOverallResults = FALSE,
  timePoint = 365,
  riskStrata = 4
)
```

**Arguments**

psMethod	How should the propensity scores be used? Can be one of "inversePtWeighted", "stratifyByPs" or "matchOnPs".
effectEstimationSettings	Parameter object providing further settings for the implementation of selected psMethod to the estimation process. Can be created using one of <a href="#">createCreateIPWArgs</a> , when inversePtWeighted is selected, <a href="#">createStratifyByPsArgs</a> when stratifyByPs is selected or <a href="#">createMatchOnPsArgs</a> when matchOnPs is selected.
psSettings	Parameter object for <a href="#">createPs</a>
createPsThreads	The number of parallel threads for the estimation of the propensity scores. Default is 1.
fitOutcomeModelsThreads	The number of parallel threads for the estimation of the outcome models
createPsThreadsNegativeControls	The number of parallel threads for the estimation of the negative control outcomes
estimateOverallResults	Should overall results be estimated? Default is FALSE
timePoint	The time point after cohort start that absolute differences should be estimated.
riskStrata	The number of risk strata. Default is 4.

**Value**

A parameter object for running the the estimation step.

---

```
createRunPlpSettingsArgs
```

*Create a parameter object for the function runPlp*

---

**Description**

Create a parameter object for the function runPlp

**Usage**

```
createRunPlpSettingsArgs(
  minCovariateFraction = 0.001,
  normalizeData = TRUE,
  modelSettings = PatientLevelPrediction::setLassoLogisticRegression(),
  testSplit = "person",
  testFraction = 0.25,
  trainFraction = NULL,
  splitSeed = NULL,
  nfold = 3,
  indexes = NULL,
  savePlpData = FALSE,
  savePlpPlots = TRUE,
  saveEvaluation = TRUE,
```



```

    verbosity = "INFO",
    timeStamp = FALSE,
    analysisId = NULL
)

```

## Arguments

minCovariateFraction	The minimum fraction of target population who must have a covariate for it to be included in the model training
normalizeData	Whether to normalise the covariates before training (Default: TRUE)
modelSettings	An object of class modelSettings created using one of the function: setLassoLogisticRegression() A lasso logistic regression model setGradientBoostingMachine() A gradient boosting machine setAdaBoost() An ada boost model setRandomForest() A random forest model setDecisionTree() A decision tree model setCovNN() A convolutional neural network model setCIReNN() A recurrent neural network model setMLP() A neural network model setDeepNN() A deep neural network model setKNN() A KNN model
testSplit	Either 'person' or 'time' specifying the type of evaluation used. 'time' find the date where testFraction of patients had an index after the date and assigns patients with an index prior to this date into the training set and post the date into the test set 'person' splits the data into test (1-testFraction of the data) and train (validationFraction of the data) sets. The split is stratified by the class label.
testFraction	The fraction of the data to be used as the test set in the patient split evaluation.
trainFraction	A real number between 0 and 1 indicating the train set fraction of the data. If not set trainFraction is equal to 1 - test
splitSeed	The seed used to split the test/train set when using a person type testSplit
nfold	The number of folds used in the cross validation (default 3)
indexes	A dataframe containing a rowId and index column where the index value of -1 means in the test set, and positive integer represents the cross validation fold (default is NULL)
savePlpData	Binary indicating whether to save the plpData object (default is FALSE)
savePlpPlots	Binary indicating whether to save the performance plots as pdf files (default is TRUE)
saveEvaluation	Binary indicating whether to save the oerformance as csv files (default is TRUE)
verbosity	Sets the level of the verbosity. If the log level is at or higher in priority than the logger threshold, a message will print. The levels are: <ul style="list-style-type: none"> <li>• DEBUG Highest verbosity showing all debug statements</li> <li>• TRACE Showing information about start and end of steps</li> <li>• INFO Show informative information (Default)</li> <li>• WARN Show warning messages</li> <li>• ERROR Show error messages</li> <li>• FATAL Be silent except for fatal errors</li> </ul>
timeStamp	If TRUE a timestamp will be added to each logging statement. Automatically switched on for TRACE level.
analysisId	Identifier for the analysis. It is used to create, e.g., the result folder. Default is a timestamp.

## Details

Create an object defining the parameter values.

---

<code>createRunSettings</code>	<i>Create a settings object for running the analyses Create the settings for running the analyses. The input consists of two parts: 1) the settings for running the prediction algorithms and 2) the settings for estimating treatment effects within strata of predicted risk.</i>
--------------------------------	---

---

## Description

Create a settings object for running the analyses Create the settings for running the analyses. The input consists of two parts: 1) the settings for running the prediction algorithms and 2) the settings for estimating treatment effects within strata of predicted risk.

## Usage

```
createRunSettings(
  runPlpSettings = createRunPlpSettingsArgs(modelSettings =
    PatientLevelPrediction::setLassoLogisticRegression()),
  runCmSettings = createRunCmSettingsArgs()
)
```

## Arguments

`runPlpSettings` A parameterer object of type `runPlpSettingsArgs` defined using the function [createRunPlpSettingsArgs](#).

`runCmSettings` A parameter object of type `runCmSettingsArgs` defined using the function [createRunCmSettingsArgs](#)

## Value

An R object of type `runSettings`

---

<code>createStratifyByPsArgs</code>	<i>Create a parameter object for the function stratifyByPs</i>
-------------------------------------	--

---

## Description

Create a parameter object for the function `stratifyByPs`

## Usage

```
createStratifyByPsArgs(
  numberOfStrata = 5,
  stratificationColumns = c(),
  baseSelection = "all"
)
```

**Arguments**

- `numberOfStrata` How many strata? The boundaries of the strata are automatically defined to contain equal numbers of target 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.
- `baseSelection` What is the base selection of subjects where the strata bounds are to be determined? Strata are defined as equally-sized strata inside this selection. Possible values are "all", "target", and "comparator".

**Details**

Create an object defining the parameter values.

---

`estimateTreatmentEffect`

*Estimate treatment effects within risk strata*

---

**Description**

Estimates treatment effects within risk strata based on the length of list `ps`.

**Usage**

```
estimateTreatmentEffect(ps, runSettings)
```

**Arguments**

- `ps` A list of objects created from [createPs](#) estimated within risk strata.
- `runSettings` An R object of type `runSettings` created using the function [createRunSettings](#)

**Value**

A list containing :

- `relativeRiskReductionHazard` ratios along with confidence intervals within risk strata
- `absoluteRiskReductionAbsolute` risk differences along with confidence intervals within risk strata
- `casesObserved` outcome proportions within risk strata
- `models` The models used to estimate relative risk reduction within risk strata

---

fitOutcomeModels	<i>Fit outcome models</i>
------------------	---------------------------

---

### Description

Fits outcome models within risk strata, estimating relative and absolute differences. Designed to be performed within a parallelized analysis.

### Usage

```
fitOutcomeModels(outcomeId, getDataSettings, pathToPs, runSettings)
```

### Arguments

outcomeId	The outcome of interest for which the estimation is performed. That is the outcome for which risk stratification is performed.
getDataSettings	An R object of type <code>getDataSettings</code> created using the function <a href="#">createGetDataSettings</a> .
pathToPs	The path to the RSEE analysis results.
runSettings	An R object of type <code>runSettings</code> created using the function <a href="#">createRunSettings</a> .

### Value

NULL. The results are all saved.

---

fitOutcomeModelsOverall	<i>Fit overall outcome model</i>
-------------------------	----------------------------------

---

### Description

Fits outcome models within risk strata, estimating relative and absolute differences. Designed to be performed within a parallelized analysis.

### Usage

```
fitOutcomeModelsOverall(
  outcomeId,
  analysisSettings,
  getDataSettings,
  runCmSettings
)
```

**Arguments**

outcomeId	The outcome of interest for which the estimation is performed. That is the outcome for which risk stratification is performed.
analysisSettings	An R object of type analysisSettings created using the function <a href="#">createAnalysisSettings</a> .
getDataSettings	An R object of type getDataSettings created using the function <a href="#">createGetDataSettings</a> .
runCmSettings	A parameter object of type runCmSettingsArgs defined using the function <a href="#">createRunCmSettingsArgs</a>

**Value**

NULL. The results are all saved.

---

fitPsModel	<i>Calculate propensity scores for a specific outcome</i>
------------	---

---

**Description**

Fits a large-scale regularized regression model to estimate propensity scores within predicted risk strata. Designed to be applied in a parallelized analysis.

**Usage**

```
fitPsModel(
  outcomeId,
  getDataSettings,
  populationSettings,
  runSettings,
  analysisSettings
)
```

**Arguments**

outcomeId	The outcome of interest for which the estimation is performed. That is the outcome for which risk stratification is performed.
getDataSettings	An R object of type getDataSettings created using the function <a href="#">createGetDataSettings</a> .
populationSettings	An R object of type populationSettings created using the function <a href="#">createPopulationSettings</a> .
runSettings	An R object of type runSettings created using the function <a href="#">createRunSettings</a> .
analysisSettings	An R object of type analysisSettings created using the function <a href="#">createAnalysisSettings</a> .

**Value**

NULL. The results are all saved.

---

fitPsModelOverall	<i>Calculate propensity scores for a specific outcome</i>
-------------------	---

---

### Description

Fits a large-scale regularized regression model to estimate propensity scores within predicted risk strata. Designed to be applied in a parallelized analysis.

### Usage

```
fitPsModelOverall(
  outcomeId,
  getDataSettings,
  populationSettings,
  analysisSettings,
  runCmSettings
)
```

### Arguments

outcomeId	The outcome of interest for which the risk stratification is performed.
getDataSettings	An R object of type <code>getDataSettings</code> created using the function <a href="#">createGetDataSettings</a> .
populationSettings	An R object of type <code>covariateSettings</code> created using the function <a href="#">createPopulationSettings</a> .
analysisSettings	An R object of type <code>analysisSettings</code> created using the function <a href="#">createAnalysisSettings</a> .
runCmSettings	A parameter object of type <code>runCmSettingsArgs</code> defined using the function <a href="#">createRunCmSettingsArgs</a>

### Value

NULL. The results are all saved.

---

fitPsModelSwitch	<i>Fits switched propensity score models</i>
------------------	--

---

### Description

Fits propensity score models where the population is stratified based on prediction model for a certain outcome of interest and estimation is focused on a different outcome

**Usage**

```
fitPsModelSwitch(
  predictOutcome,
  compareOutcome,
  analysisSettings,
  getDataSettings,
  populationSettings,
  runSettings
)
```

**Arguments**

`predictOutcome` The outcome of the prediction step

`compareOutcome` The outcome of interest for the estimation step

`analysisSettings` An R object of type `analysisSettings` created using the function [createAnalysisSettings](#).

`getDataSettings` An R object of type `getDataSettings` created using the function [createGetDataSettings](#).

`populationSettings` An R object of type `populationSettings` created using the function [createPopulationSettings](#).

`runSettings` An R object of type `runSettings` created using the function [createRunSettings](#).

---

`outcomeModelWeighted` *Fits a weighted cox regression model*

---

**Description**

Fits a weighted cox regression model using an inverse probability of treatment weighting approach

**Usage**

```
outcomeModelWeighted(
  ps,
  calculateWeights = TRUE,
  weightsType = "ATE",
  useStabilizedWeights = TRUE,
  truncationLevels
)
```

**Arguments**

`ps` A dataframe with propensity scores as generated from [createPs](#).

`calculateWeights` Should weights be calculated?

`weightsType` The type of weights for the balancing of covariates. Should be either 'ATE' or 'ATT'

`useStabilizedWeights` Should stabilized weights be used?

`truncationLevels` The level of truncation expressed in percentiles of the propensity score.

**Value**

A weighted cox regression model.

---

prepareForPlpData	<i>Prepares for the running the PatientLevelPrediction package</i>
-------------------	--

---

**Description**

Prepares for running the PatientLevelPrediction package by merging the treatment and comparator cohorts and defining a new covariate for treatment.

**Usage**

```
prepareForPlpData(
  treatmentCohortId,
  comparatorCohortId,
  targetCohortId,
  cohortDatabaseSchema,
  cohortTable,
  resultsDatabaseSchema,
  mergedCohortTable,
  connectionDetails
)
```

**Arguments**

treatmentCohortId	The treatment cohort id
comparatorCohortId	The comparator cohort id
targetCohortId	The id of the merged cohorts
cohortDatabaseSchema	The name of the database schema that is the location where the cohort data used to define the at risk cohort is available
cohortTable	The table that contains the treatment and comparator cohorts.
resultsDatabaseSchema	The name of the database schema to store the new tables. Need to have write access.
mergedCohortTable	The table that will contain the merged cohorts.
connectionDetails	The connection details required to connect to a database.

**Value**

Creates the tables resultsDatabaseSchema.mergedCohortTable, resultsDatabaseSchema.attributeDefinitionTable and resultsDatabaseSchema.cohortAttributeTable



---

relativeRiskReduction	<i>Relative risk reduction</i>
-----------------------	--------------------------------

---

**Description**

Calculates hazard ratios within risk strata.

**Usage**

```
relativeRiskReduction(model)
```

**Arguments**

model	The model that was used to fit a cox regression model to the data.
-------	--

**Value**

A dataframe with hazard ratios for treatment effect across risk strata along with 95 percent confidence intervals

---

runRiskStratifiedEstimation	<i>Runs a risk stratified analysis</i>
-----------------------------	--

---

**Description**

Runs a risk stratified analysis in two stages. It first runs a prediction algorithm using PatientLevelPrediction to derive baseline patient risks and then derives estimates within risk strata incorporating functionality from CohortMethod package.

**Usage**

```
runRiskStratifiedEstimation(  
  connectionDetails,  
  analysisSettings,  
  databaseSettings,  
  getDataSettings,  
  covariateSettings,  
  populationSettings,  
  runSettings,  
  tempDir = NULL  
)
```

**Arguments**

connectionDetails	An R object of type connectionDetails created using the function <a href="#">createConnectionDetails</a> .
analysisSettings	An R object of type analysisSettings created using the function <a href="#">createAnalysisSettings</a> .
databaseSettings	An R object of type databaseSettings created using the function <a href="#">createDatabaseSettings</a> .
getDataSettings	An R object of type getDataSettings created using the function <a href="#">createGetDataSettings</a> .
covariateSettings	An R object of type covariateSettings created using the function <a href="#">createCovariateSettings</a> .
populationSettings	An R object of type populationSettings created using the function <a href="#">createPopulationSettings</a> .
runSettings	An R object of type runSettings created using the function <a href="#">createRunSettings</a> .
tempDir	Directory where temporary files required for the analysis will be stored.

**Value**

The function saves all results based on analysisSettings. No results are returned.

---

stratifiedKaplanMeier    *Calculate stratified Kaplan-Meier estimates*

---

**Description**

Calculate stratified Kaplan-Meier estimates

**Usage**

```
stratifiedKaplanMeier(population, timePoint)
```

**Arguments**

population	The population of interest stratified using the <a href="#">stratifyByPs</a>
timePoint	The point in time for which the absolute risk difference is required

**Value**

A vector of the absolute risk difference along with the lowest and highest limits of the the 95 percent confidence interval

# Index

`absoluteRiskReduction`, [3](#)

`createAnalysisSettings`, [3](#), [12](#), [21–23](#), [26](#)  
`createConnectionDetails`, [26](#)  
`createCovariateSettings`, [8](#), [26](#)  
`createCreateIPWArgs`, [4](#), [16](#)  
`createCreatePsArgs`, [5](#)  
`createDatabaseSettings`, [6](#), [26](#)  
`createGetCmDataArgs`, [7](#)  
`createGetCovariateSettings`, [8](#)  
`createGetDataSettings`, [9](#), [20–23](#), [26](#)  
`createGetPlpDataArgs`, [9](#), [9](#)  
`createIPW`, [10](#)  
`createMatchOnPsArgs`, [11](#), [16](#)  
`createOverallResults`, [11](#)  
`createPopulationCmSettingsArgs`, [12](#), [15](#)  
`createPopulationPlpSettingsArgs`, [13](#), [15](#)  
`createPopulationSettings`, [15](#), [21–23](#), [26](#)  
`createPs`, [10](#), [16](#), [19](#), [23](#)  
`createRunCmSettingsArgs`, [15](#), [18](#), [21](#), [22](#)  
`createRunPlpSettingsArgs`, [16](#), [18](#)  
`createRunSettings`, [12](#), [15](#), [18](#), [19–21](#), [23](#), [26](#)  
`createStratifyByPsArgs`, [16](#), [18](#)

`estimateTreatmentEffect`, [19](#)

`fitOutcomeModels`, [20](#)  
`fitOutcomeModelsOverall`, [20](#)  
`fitPsModel`, [21](#)  
`fitPsModelOverall`, [22](#)  
`fitPsModelSwitch`, [22](#)

`getDbCohortMethodData`, [7](#)

`matchOnPs`, [3](#)

`outcomeModelWeighted`, [23](#)

`prepareForPlpData`, [24](#)

`relativeRiskReduction`, [25](#)  
`runRiskStratifiedEstimation`, [25](#)

`stratifiedKaplanMeier`, [26](#)  
`stratifyByPs`, [3](#), [26](#)