Package 'FeatureExtraction'

March 28, 2016

Type Package

Title Generating Features for a Cohort

Version 0.0.1

Date 2016-3-25

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

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

Description An R package for generating features (covariates) for a cohort using data in the Common Data Model.

License Apache License 2.0

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

Imports bit,
ff,
ffbase (>= 0.12.1),
plyr,
Rcpp (>= 0.11.2),
RJDBC,
SqlRender (>= 1.1.3),

Suggests testthat,
knitr,
rmarkdown

LinkingTo Rcpp

NeedsCompilation yes

RoxygenNote 5.0.1

R topics documented:

byMaxFf 2
createCohortAttrCovariateSettings 2
createCovariateSettings 3
createHdpsCovariateSettings 8
createTextCovariateSettings 11

FeatureExtraction 12
getDbCohortAttrCovariatesData 12
getDbCovariateData 13
getDbDefaultCovariateData 15
getDbHdpsCovariateData 16
getDbTextCovariateData 17
loadCovariateData 18
normalizeCovariates 19
saveCovariateData 19

Index **20**

`byMaxFf` *Compute max of values binned by a second variable*

Description

Compute max of values binned by a second variable

Usage

```
byMaxFf(values, bins)
```

Arguments

`values`	An ff object containing the numeric values to take the max of.
`bins`	An ff object containing the numeric values to bin by.

Examples

```
values <- ff::as.ff(c(1, 1, 2, 2, 1))
bins <- ff::as.ff(c(1, 1, 1, 2, 2))
byMaxFf(values, bins)
```

`createCohortAttrCovariateSettings`
Create cohort attribute covariate settings

Description

Create cohort attribute covariate settings

Usage

```
createCohortAttrCovariateSettings(attrDatabaseSchema,
  attrDefinitionTable = "attribute_definition",
  cohortAttrTable = "cohort_attribute", includeAttrIds = c())
```

Arguments

`attrDatabaseSchema`
: The database schema where the attribute definition and cohort attribute table can be found.

`attrDefinitionTable`
: The name of the attribute definition table.

`cohortAttrTable`
: The name of the cohort attribute table.

`includeAttrIds`
: (optional) A list of attribute definition IDs to restrict to.

Details

Creates an object specifying where the cohort attributes can be found to construct covariates. The attributes should be defined in a table with the same structure as the attribute_definition table in the Common Data Model. It should at least have these columns:

attribute_definition_id A unique identifier of type integer.

attribute_name A short description of the attribute.

The cohort attributes themselves should be stored in a table with the same format as the cohort_attribute table in the Common Data Model. It should at least have these columns:

cohort_definition_id A key to link to the cohort table. On CDM v4, this field should be called `cohort_concept_id`.

subject_id A key to link to the cohort table.

cohort_start_date A key to link to the cohort table.

attribute_definition_id An foreign key linking to the attribute definition table.

value_as_number A real number.

Value

An object of type `covariateSettings`, to be used in other functions.

---

createCovariateSettings

Create covariate settings

Description

Create covariate settings

Usage

```
createCovariateSettings(useCovariateCohortIdIs1 = FALSE,
  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,
  useCovariateObservationCount365d = FALSE, useCovariateMeasurement = FALSE,
  useCovariateMeasurement365d = FALSE, useCovariateMeasurement30d = FALSE,
  useCovariateMeasurementCount365d = FALSE,
  useCovariateMeasurementBelow = FALSE,
  useCovariateMeasurementAbove = FALSE, useCovariateConceptCounts = FALSE,
  useCovariateRiskScores = FALSE, useCovariateRiskScoresCharlson = FALSE,
  useCovariateRiskScoresDCSI = FALSE, useCovariateRiskScoresCHADS2 = FALSE,
  useCovariateRiskScoresCHADS2VASc = FALSE,
  useCovariateInteractionYear = FALSE, useCovariateInteractionMonth = FALSE,
  excludedCovariateConceptIds = c(), includedCovariateConceptIds = c(),
  deleteCovariatesSmallCount = 100)
```

Arguments

`useCovariateCohortIdIs1`
: A boolean value (TRUE/FALSE) to determine if a covariate should be contructed for whether the cohort ID is 1 (currently primarily used in CohortMethod).

`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 vocabluary 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 vocabluary 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 vocabluary 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.

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.

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

useCovariateMeasurement365d
: A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of measurement in 365d window prior to or on cohort index date. Only applicable if useCovariateMeasurement = TRUE.

useCovariateMeasurement30d
: A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of measurement in 30d window prior to or on cohort index date. Only applicable if useCovariateMeasurement = TRUE.

useCovariateMeasurementCount365d
: A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for the count of each measurement concept in 365d window prior to or on cohort index date. Only applicable if useCovariateMeasurement = TRUE.

useCovariateMeasurementBelow
: A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of measurement with a numeric value below normal range for latest value within 180d of cohort index. Only applicable if useCovariateMeasurement = TRUE (CDM v5+) or useCovariateObservation = TRUE (CDM v4).

useCovariateMeasurementAbove
: A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of measurement with a numeric value above normal range for latest value within 180d of cohort index. Only applicable if useCovariateMeasurement = TRUE (CDM v5+) or useCovariateObservation = TRUE (CDM v4).

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.

useCovariateRiskScoresCHADS2VASc
: A boolean value (TRUE/FALSE) to determine if the CHADS2VASc 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

creates an object specifying how covariates should be contructed from data in the CDM model.

Value

An object of type `defaultCovariateSettings`, to be used in other functions.

`createHdpsCovariateSettings` *Create HDPS covariate settings*

Description

Create HDPS covariate settings

Usage

```
createHdpsCovariateSettings(useCovariateCohortIdIs1 = FALSE,
  useCovariateDemographics = TRUE, useCovariateDemographicsGender = TRUE,
  useCovariateDemographicsRace = TRUE,
  useCovariateDemographicsEthnicity = TRUE,
  useCovariateDemographicsAge = TRUE, useCovariateDemographicsYear = TRUE,
```

```
  useCovariateDemographicsMonth = TRUE,
  useCovariateConditionOccurrence = TRUE,
  useCovariate3DigitIcd9Inpatient180d = FALSE,
  useCovariate3DigitIcd9Inpatient180dMedF = FALSE,
  useCovariate3DigitIcd9Inpatient180d75F = FALSE,
  useCovariate3DigitIcd9Ambulatory180d = FALSE,
  useCovariate3DigitIcd9Ambulatory180dMedF = FALSE,
  useCovariate3DigitIcd9Ambulatory180d75F = FALSE,
  useCovariateDrugExposure = FALSE,
  useCovariateIngredientExposure180d = FALSE,
  useCovariateIngredientExposure180dMedF = FALSE,
  useCovariateIngredientExposure180d75F = FALSE,
  useCovariateProcedureOccurrence = FALSE,
  useCovariateProcedureOccurrenceInpatient180d = FALSE,
  useCovariateProcedureOccurrenceInpatient180dMedF = FALSE,
  useCovariateProcedureOccurrenceInpatient180d75F = FALSE,
  useCovariateProcedureOccurrenceAmbulatory180d = FALSE,
  useCovariateProcedureOccurrenceAmbulatory180dMedF = FALSE,
  useCovariateProcedureOccurrenceAmbulatory180d75F = FALSE,
  excludedCovariateConceptIds = c(), includedCovariateConceptIds = c(),
  deleteCovariatesSmallCount = 100)
```

Arguments

`useCovariateCohortIdIs1`
: A boolean value (TRUE/FALSE) to determine if a covariate should be contructed for whether the cohort ID is 1 (currently primarily used in CohortMethod).

`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.

useCovariate3DigitIcd9Inpatient180d
: 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 setting in 180d window prior to or on cohort index date. Conditions are aggregated at the ICD-9 3-digit level. Only applicable if useCovariateConditionOccurrence = TRUE.

useCovariate3DigitIcd9Inpatient180dMedF
: Similar to useCovariate3DigitIcd9Inpatient180d, but now only if the frequency of the ICD-9 code is higher than the median.

useCovariate3DigitIcd9Inpatient180d75F
: Similar to useCovariate3DigitIcd9Inpatient180d, but now only if the frequency of the ICD-9 code is higher than the 75th percentile.

useCovariate3DigitIcd9Ambulatory180d
: A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of condition within ambulatory setting in 180d window prior to or on cohort index date. Conditions are aggregated at the ICD-9 3-digit level. Only applicable if useCovariateConditionOccurrence = TRUE.

useCovariate3DigitIcd9Ambulatory180dMedF
: Similar to useCovariate3DigitIcd9Ambulatory180d, but now only if the frequency of the ICD-9 code is higher than the median.

useCovariate3DigitIcd9Ambulatory180d75F
: Similar to useCovariate3DigitIcd9Ambulatory180d, but now only if the frequency of the ICD-9 code is higher than the 75th percentile.

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

useCovariateIngredientExposure180d
: A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of drug ingredients within inpatient setting in 180d window prior to or on cohort index date. Only applicable if useCovariateDrugExposure = TRUE.

useCovariateIngredientExposure180dMedF
: Similar to useCovariateIngredientExposure180d, but now only if the frequency of the ingredient is higher than the median.

useCovariateIngredientExposure180d75F
: Similar to useCovariateIngredientExposure180d, but now only if the frequency of the ingredient is higher than the 75th percentile.

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

useCovariateProcedureOccurrenceInpatient180d
: A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of procedures within inpatient setting in 180d window prior to or on cohort index date. Only applicable if useCovariateProcedureOccurrence = TRUE.

useCovariateProcedureOccurrenceInpatient180dMedF
: Similar to useCovariateProcedureOccurrenceInpatient180d, but now only if the frequency of the procedure code is higher than the median.

useCovariateProcedureOccurrenceInpatient180d75F
: Similar to useCovariateProcedureOccurrenceInpatient180d, but now only if the frequency of the procedure code is higher than the 75th percentile.

`useCovariateProcedureOccurrenceAmbulatory180d`
: A boolean value (TRUE/FALSE) to determine if covariates will be created and used in models that look for presence/absence of procedures within ambulatory setting in 180d window prior to or on cohort index date. Only applicable if useCovariateProcedureOccurrence = TRUE.

`useCovariateProcedureOccurrenceAmbulatory180dMedF`
: Similar to `useCovariateProcedureOccurrenceAmbulatory180d`, but now only if the frequency of the procedure code is higher than the median.

`useCovariateProcedureOccurrenceAmbulatory180d75F`
: Similar to `useCovariateProcedureOccurrenceAmbulatory180d`, but now only if the frequency of the procedure code is higher than the 75th percentile.

`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

creates an object specifying how covariates should be contructed from data in the CDM model.

Value

An object of type `hdpsCovariateSettings`, to be used in other functions.

`createTextCovariateSettings` *Create text covariate settings*

Description

Create text covariate settings

Usage

```
createTextCovariateSettings(language = "eng", removeNegations = TRUE,
  deleteCovariatesSmallCount = 100)
```

Arguments

`language` Specify the language of the free-text.

`removeNegations`
: Remove negated text prior to constructing features.

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

Details

creates an object specifying how covariates should be constructed from text in notes table in the CDM model.

Value

An object of type `covariateSettings`, to be used in other functions.

`FeatureExtraction` *FeatureExtraction*

Description

FeatureExtraction

`getDbCohortAttrCovariatesData`
Getcovariate information from the database through the cohort_attribute table

---

Description

Constructs a large default set of covariates for one or more cohorts using data in the CDM schema. Includes covariates for all drugs, drug classes, condition, condition classes, procedures, observations, etc.

Usage

```
getDbCohortAttrCovariatesData(connection, oracleTempSchema = NULL,
  cdmDatabaseSchema, cdmVersion = "4", cohortTempTable = "cohort_person",
  rowIdField = "subject_id", covariateSettings)
```

Arguments

`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 specifiy both the database and the schema, so for example 'cdm_instance.dbo'.
`cdmVersion`	Define the OMOP CDM version used: currently support "4" and "5".
`cohortTempTable`	Name of the temp table holding the cohort for which we want to construct covaraites

`rowIdField` The name of the field in the cohort temp table that is to be used as the row_id field in the output table. This can be especially usefull if there is more than one period per person.

`covariateSettings` An object of type `covariateSettings` as created using the `createCohortAttrCovariateSettings` function.

Details

This function uses the data in the CDM to construct a large set of covariates for the provided cohort. The cohort is assumed to be in an existing temp table with these fields: 'subject_id', 'cohort_definition_id', 'cohort_start_date'. Optionally, an extra field can be added containing the unique identifier that will be used as rowID in the output. Typically, users don't call this function directly but rather use the `getDbCovariateData` function instead.

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 ffdf object listing the baseline covariates per person in the cohorts. This is done using a sparse representation: covariates with a value of 0 are omitted to save space. The covariates object will have three columns: rowId, covariateId, and covariateValue. The rowId is usually equal to the person_id, unless specified otherwise in the rowIdField argument.

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

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

`getDbCovariateData` *Get covariate information from the database*

Description

Uses one or several covariate builder functions to construct covariates.

Usage

```
getDbCovariateData(connectionDetails = NULL, connection = NULL,
  oracleTempSchema = NULL, cdmDatabaseSchema, cdmVersion = "4",
  cohortTable = "cohort", cohortDatabaseSchema = cdmDatabaseSchema,
  cohortTableIsTemp = FALSE, cohortIds = c(), rowIdField = "subject_id",
  covariateSettings, normalize = TRUE)
```

Arguments

`connectionDetails` An R object of type `connectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package. Either the `connection` or `connectionDetails` argument should be specified.

`connection` A connection to the server containing the schema as created using the `connect` function in the `DatabaseConnector` package. Either the `connection` or `connectionDetails` argument should be specified.

`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 specifiy both the database and the schema, so for example 'cdm_instance.dbo'.

`cdmVersion`
: Define the OMOP CDM version used: currently support "4" and "5".

`cohortTable`
: Name of the (temp) table holding the cohort for which we want to construct covariates

`cohortDatabaseSchema`
: If the cohort table is not a temp table, specify the database schema where the cohort table can be found. On SQL Server, this should specifiy both the database and the schema, so for example 'cdm_instance.dbo'.

`cohortTableIsTemp`
: Is the cohort table a temp table?

`cohortIds`
: For which cohort IDs should covariates be constructed? If left empty, covariates will be constructed for all cohorts in the specified cohort table.

`rowIdField`
: The name of the field in the cohort table that is to be used as the row_id field in the output table. This can be especially usefull if there is more than one period per person.

`covariateSettings`
: Either an object of type `covariateSettings` as created using one of the createCovariate functions, or a list of such objects.

`normalize`
: Should covariate values be normalized? If true, values will be divided by the max value per covariate.

Details

This function uses the data in the CDM to construct a large set of covariates for the provided cohort. The cohort is assumed to be in an existing table with these fields: 'subject_id', 'cohort_definition_id', 'cohort_start_date'. Optionally, an extra field can be added containing the unique identifier that will be used as rowID in the output.

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 ffdf object listing the baseline covariates per person in the cohorts. This is done using a sparse representation: covariates with a value of 0 are omitted to save space. The covariates object will have three columns: rowId, covariateId, and covariateValue. The rowId is usually equal to the person_id, unless specified otherwise in the rowIdField argument.

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

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

getDbDefaultCovariateData *Get default covariate information from the database*

Description

Constructs a large default set of covariates for one or more cohorts using data in the CDM schema. Includes covariates for all drugs, drug classes, condition, condition classes, procedures, observations, etc.

Usage

```
getDbDefaultCovariateData(connection, oracleTempSchema = NULL,
  cdmDatabaseSchema, cdmVersion = "4", cohortTempTable = "cohort_person",
  rowIdField = "subject_id", covariateSettings)
```

Arguments

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 specifiy both the database and the schema, so for example 'cdm_instance.dbo'.
cdmVersion	Define the OMOP CDM version used: currently support "4" and "5".
cohortTempTable	Name of the temp table holding the cohort for which we want to construct covaraites
rowIdField	The name of the field in the cohort temp table that is to be used as the row_id field in the output table. This can be especially usefull if there is more than one period per person.
covariateSettings	An object of type defaultCovariateSettings as created using the createCovariateSettings function.

Details

This function uses the data in the CDM to construct a large set of covariates for the provided cohort. The cohort is assumed to be in an existing temp table with these fields: 'subject_id', 'cohort_definition_id', 'cohort_start_date'. Optionally, an extra field can be added containing the unique identifier that will be used as rowID in the output. Typically, users don't call this function directly but rather use the getDbCovariateData function instead.

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 ffdf object listing the baseline covariates per person in the cohorts. This is done using a sparse representation: covariates with a value of 0 are omitted to save space. The covariates object will have three columns: rowId, covariateId, and covariateValue. The rowId is usually equal to the person_id, unless specified otherwise in the rowIdField argument.

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

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

`getDbHdpsCovariateData` *Get HDPS covariate information from the database*

Description

Constructs the set of covariates for one or more cohorts using data in the CDM schema. This implements the covariates typically used in the HDPS algorithm.

Usage

```
getDbHdpsCovariateData(connection, oracleTempSchema = NULL, cdmDatabaseSchema,
  cdmVersion = "4", cohortTempTable = "cohort_person",
  rowIdField = "subject_id", covariateSettings)
```

Arguments

`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 specifiy both the database and the schema, so for example 'cdm_instance.dbo'.
`cdmVersion`	Define the OMOP CDM version used: currently support "4" and "5".
`cohortTempTable`	Name of the temp table holding the cohort for which we want to construct covaraites
`rowIdField`	The name of the field in the cohort temp table that is to be used as the row_id field in the output table. This can be especially usefull if there is more than one period per person.
`covariateSettings`	An object of type `covariateSettings` as created using the `createHdpsCovariateSettings` function.

Details

This function uses the data in the CDM to construct a large set of covariates for the provided cohort. The cohort is assumed to be in an existing temp table with these fields: 'subject_id', 'cohort_definition_id', 'cohort_start_date'. Optionally, an extra field can be added containing the unique identifier that will be used as rowID in the output. Typically, users don't call this function directly but rather use the `getDbCovariateData` function instead.

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 ffdf object listing the baseline covariates per person in the cohorts. This is done using a sparse representation: covariates with a value of 0 are omitted to save space. The covariates object will have three columns: rowId, covariateId, and covariateValue. The rowId is usually equal to the person_id, unless specified otherwise in the rowIdField argument.

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

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

`getDbTextCovariateData` *Get text covariate information from the database*

Description

Uses a bag-of-words approach to construct covariates based on free-text.

Usage

```
getDbTextCovariateData(connection, oracleTempSchema = NULL, cdmDatabaseSchema,
  cdmVersion = "4", cohortTempTable = "cohort_person",
  rowIdField = "subject_id", covariateSettings)
```

Arguments

`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 specifiy both the database and the schema, so for example 'cdm_instance.dbo'.
`cdmVersion`	Define the OMOP CDM version used: currently support "4" and "5".
`cohortTempTable`	Name of the temp table holding the cohort for which we want to construct covaraites
`rowIdField`	The name of the field in the cohort temp table that is to be used as the row_id field in the output table. This can be especially usefull if there is more than one period per person.
`covariateSettings`	An object of type `covariateSettings` as created using the `createTextCovariateSettings` function.

Details

This function uses the data in the CDM to construct a large set of covariates for the provided cohort. The cohort is assumed to be in an existing temp table with these fields: 'subject_id', 'cohort_definition_id', 'cohort_start_date'. Optionally, an extra field can be added containing the unique identifier that will be used as rowID in the output. Typically, users don't call this function directly but rather use the `getDbCovariateData` function instead.

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 ffdf object listing the baseline covariates per person in the cohorts. This is done using a sparse representation: covariates with a value of 0 are omitted to save space. The covariates object will have three columns: rowId, covariateId, and covariateValue. The rowId is usually equal to the person_id, unless specified otherwise in the rowIdField argument.

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

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

`loadCovariateData` *Load the covariate data from a folder*

Description

`loadCovariateData` loads an object of type covariateData from a folder in the file system.

Usage

```
loadCovariateData(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 covariateData

Examples

```
# todo
```

normalizeCovariates *Normalize covariate values*

Description

Normalize covariate values

Usage

```
normalizeCovariates(covariates)
```

Arguments

covariates	An ffdf object as generated using the getDbCovariateData function.#'

Details

Normalize covariate values by dividing by the max. This is to avoid numeric problems when fitting models.

saveCovariateData *Save the covariate data to folder*

Description

saveCovariateData saves an object of type covariateData to folder.

Usage

```
saveCovariateData(covariateData, file)
```

Arguments

covariateData	An object of type covariateData as generated using getDbCovariateData.
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
```

Index

byMaxFf, 2

createCohortAttrCovariateSettings, 2, *13*
createCovariateSettings, 3, *15*
createHdpsCovariateSettings, 8, *16*
createTextCovariateSettings, 11, *17*

FeatureExtraction, 12
FeatureExtraction-package *(*FeatureExtraction*)*, 12

getDbCohortAttrCovariatesData, 12
getDbCovariateData, *13*, 13, *15, 16*, *18, 19*
getDbDefaultCovariateData, 15
getDbHdpsCovariateData, 16
getDbTextCovariateData, 17

loadCovariateData, 18

normalizeCovariates, 19

saveCovariateData, 19