Package 'TimeSeriesAnalysis'

November 28, 2022

Type Package

Title Time series analysis exploration for cohorts generated against the OMOP Common Data Model

Version 0.0.1

Maintainer Anthony G. Sena <asena5@its.jnj.com>

Description A package for time series analysis exploration for cohorts generated against the OMOP Common Data Model.

Depends DatabaseConnector (>= 5.0.2),
R (>= 2.10)

Imports checkmate,
CohortGenerator (>= 0.7.0),
dplyr,
ggplot2,
lubridate,
ocp (>= 0.1),
ParallelLogger (>= 3.0.1),
readr,
rlang,
segmented (>= 1.6.0),
SqlRender (>= 1.7.0)

Suggests DT,
Eunomia,
htmltools,
knitr,
rmarkdown,
shiny,
shinydashboard,
shinyWidgets,
testthat (>= 3.0.0)

Remotes ohdsi/CohortGenerator,
ohdsi/Eunomia

License Apache License

VignetteBuilder knitr

URL https://github.com/EHDEN/TimeSeriesAnalysis

BugReports https://github.com/EHDEN/TimeSeriesAnalysis/issues

RoxygenNote 7.2.1

Encoding UTF-8

Language en-US

LazyData true

Config/testthat/edition 3

R topics documented:

`createCohortTimeSeriesArgs` *Create a parameter object for the function getCohortTimeSeriesData*

Description

Create a parameter object for the function getCohortTimeSeriesData

Usage

```
createCohortTimeSeriesArgs(cohortIds = -1, timeInterval = "month")
```

Arguments

`cohortIds`	A vector of cohort Ids; set to -1 to use all cohorts in the cohort table
`timeInterval`	The time interval (year, month, day) to use when extracting the cohort time series.

Details

Create an object defining the parameter values for use with the

See Also

[getCohortTimeSeriesData]

createOcpArgs *Create a parameter object for the function createOcpModel*

Description

Create a parameter object for the function createOcpModel

Usage

```
createOcpArgs(
  oCPD = NULL,
  missPts = "none",
  hazardFunc = function(x, lambda) {
     ocp::const_hazard(x, lambda = 100)
 },
  probModel = list("g"),
  initParams = list(list(m = 0, k = 0.01, a = 0.01, b = 1e-04)),
  multivariate = FALSE,
  cpthreshold = 0.5,
  truncRlim = .Machine$double.xmin,
  minRlength = 1,
  maxRlength = 10^4,
  minsep = 1,
  maxsep = 10^4,
  timing = FALSE,
  getR = FALSE,
  optionalOutputs = FALSE,
  printupdates = FALSE
)
```

Arguments

oCPD	ocp object computed in a previous run of an algorithm. it can be built upon with the input data points, as long as the settings for both are the same.
missPts	This setting indicates how to deal with missing points (e.g. NA). The options are: "mean", "prev", "none", and a numeric value. If the data is multivariate. The numeric replacement value could either be a single value which would apply to all dimensions, or a vector of the same length as the number of dimensions of the data.
hazardFunc	This setting allows choosing a hazard function, and also setting the constants within that function. For example, the default hazard function is: function(x, lambda)const_hazard(x, lambda=100) and the lambda can be set as appropriate.
probModel	This parameter is a function to be used to calculate the predictive probabilities and update the parameters of the model. The default setting uses a gaussian underlying distribution: "gaussian"
initParams	The parameters used to initialize the probability model. The default settings correspond to the input default gaussian model.
multivariate	This setting indicates if the incoming data is multivariate or univariate.

cpthreshold	Probability threshold for the method of extracting a list of all changepoints that have a run length probability higher than a specified value. The default is set to 0.5.
truncRlim	The probability threshold to begin truncating the R vector. The R vector is a vector of run-length probabilities. To prevent truncation, set this to 0. The defaults setting is 10^(-4) as suggested by the paper.
minRlength	The minimum size the run length probabilities vector must be before beginning to check for the truncation threshold.
maxRlength	The maximum size the R vector is allowed to be, before enforcing truncation to happen.
minsep	This setting constrains the possible changepoint locations considered in determining the optimal set of changepoints. It prevents considered changepoints that are closer together than the value of minsep. The default is 3.
maxsep	This setting constrains the possible changepoint locations considered in determining the optimal set of changepoints. It prevents considered changepoints that are closer farther apart than the value of maxsep. The default is 100.
timing	To print out times during the algorithm running, to track its progress, set this setting to true.
getR	To output the full R matrix, set this setting to TRUE. Outputting this matrix causes a major slow down in efficiency.
optionalOutputs	Output additional values calculated during running the algorithm, including a matrix containing all the input data, the predictive probability vectors at each step of the algorithm, and the vector of means at each step of the algorithm.
printupdates	This setting prints out updates on the progress of the algorithm if set to TRUE.

Details

Create an object defining the parameter values for use with the createOcpModel function

See Also

[ocp::onlineCPD()] which this function wraps its arguments.

`createOcpModel` *Create a Bayesian Online Changepoint model*

Description

Create a Bayesian Online Changepoint model

Usage

```
createOcpModel(tsData, ocpArgs)
```

Arguments

tsData	The time series data to use for fitting the model
ocpArgs	The OCP linear model arguments. See `createSegmentedArgs`. for more details.

Details

Create a Bayesian Online Changepoint model using the time-series data and arguments from `createSegmentedArgs`.

`createSegmentedArgs` *Create a parameter object for the function createSegmentedModel*

Description

Create a parameter object for the function createSegmentedModel

Usage

```
createSegmentedArgs(
  modelType,
  seg.Z = ~eventDate,
  psi = NA,
  npsi = NA,
  fixed.psi = NULL,
  control = segmented::seg.control(),
  model = TRUE
)
```

Arguments

`modelType`	One of: linear or poisson
`seg.Z`	the segmented variables(s), i.e. the continuous covariate(s) understood to have a piecewise-linear relationship with response. It is a formula with no response variable, such as seg.Z=~x or seg.Z=~x1+x2.
`psi`	Starting values for the breakpoints to be estimated. If there is a single segmented variable specified in seg.Z, psi is a numeric vector
`npsi`	A named vector or list meaning the number (and not locations) of breakpoints to be estimated.
`fixed.psi`	An optional named list meaning the breakpoints to be kept fixed during the estimation procedure. The names should be a subset of (or even the same) variables specified in seg.Z. If there is a single variable in seg.Z, a simple numeric vector can be specified.
`control`	a list of parameters for controlling the fitting process. See the documentation for [segmented::seg.control] for details
`model`	logical value indicating if the model.frame should be returned

Details

Create an object defining the parameter values for use with the createSegmentedModel.

See Also

[segmented::segmented()] which this function wraps its arguments.

createSegmentedModel *Create a segmented linear model*

Description

Create a segmented linear model

Usage

```
createSegmentedModel(tsData, segmentedArgs)
```

Arguments

tsData	The time series data to use for fitting the model
segmentedArgs	The segmented linear model arguments. See createSegmentedArgs. for more details.

Details

Create a segmented linear model using the time-series data and arguments from createSegmentedArgs.

createTsAnalysis *Create a TimeSeries analysis specification*

Description

Create a TimeSeries analysis specification

Usage

```
createTsAnalysis(analysisId = 1, description = "", tsArgs)
```

Arguments

analysisId	An integer that will be used later to refer to this specific set of analysis choices.
description	A short description of the analysis.
tsArgs	An object representing the time series arguments to be used when calling the various time series model functions

Details

Create a set of analysis choices, to be used with the [runTsAnalyses()] function.

Providing a NULL value for any of the argument applies the corresponding step will not be executed.

drugData *Sample drug data for using the change detection time series methods*

Description

A dataset containing the use of Hydroxychloroquine and Dexamethasone in COVID patients from a large US Claims dataset.

Usage

```
data(drugData)
```

Format

A data frame with 20 rows and 6 variables:

cohortDefinitionId The cohort definition id

cohortName The drug cohort of interest

cohortStartDate The cohort start date

timeInterval The time interval of capture for the time series data

subjectCount The number of people in the cohort for the month combination

eventCount The number of events in the cohort for the month combination. For cohorts, people may have > 1 event

databaseId The database ID that contributed the data

executeTimeSeriesAnalyses *Extracts the time series data and runs the time series analyses on all (or a subset of) the cohorts. This function assumes the cohorts have already been generated using the OHDSI CohortGenerator package*

Description

Extracts the time series data and runs the time series analyses on all (or a subset of) the cohorts. This function assumes the cohorts have already been generated using the OHDSI CohortGenerator package

Usage

```
executeTimeSeriesAnalyses(
  connectionDetails = NULL,
  connection = NULL,
  cdmDatabaseSchema,
  tempEmulationSchema = NULL,
  cohortDatabaseSchema,
  cohortTable = "cohort",
  outputFolder,
  databaseId,
```

```
  tsDataFieldName = "subjectCount",
  cohortTimeSeriesArgs,
  tsAnalysisList
)
```

Arguments

connectionDetails	An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the connect function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cdmDatabaseSchema	Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.
tempEmulationSchema	Some database platforms like Oracle and Impala do not truly support temp tables. To emulate temp tables, provide a schema with write privileges where temp tables can be created.
cohortDatabaseSchema	Schema name where your cohort table resides. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortTable	The name of the cohort table.
outputFolder	The location where the cohort time series data results will be written.
databaseId	The database identifier for the time series data
tsDataFieldName	The time series data field name to use. This will be either the "subjectCount" or "eventCount" as computed based on the cohort.
cohortTimeSeriesArgs	The cohort time series arguments. @seealso[createCohortTimeSeriesArgs] for more information.
tsAnalysisList	A list of time series analyses as specified by using the @seealso [createSegmentedArgs] and/or @seealso [createOcpArgs]

Details

The time series data for each cohort is extracted and stored to the file system. Then each cohort's time series is used to build one or more models.

`getCohortTimeSeriesData` *Retrieve time series data for one or more cohorts*

Description

Retrieve time series data for one or more cohorts

Usage

```
getCohortTimeSeriesData(
  connectionDetails = NULL,
  connection = NULL,
  cdmDatabaseSchema,
  tempEmulationSchema = NULL,
  cohortDatabaseSchema,
  cohortTable = "cohort",
  outputFolder,
  databaseId,
  cohortTimeSeriesArgs
)
```

Arguments

connectionDetails	An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the connect function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cdmDatabaseSchema	Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.
tempEmulationSchema	Some database platforms like Oracle and Impala do not truly support temp tables. To emulate temp tables, provide a schema with write privileges where temp tables can be created.
cohortDatabaseSchema	Schema name where your cohort table resides. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortTable	The name of the cohort table.
outputFolder	The location where the cohort time series data results will be written.
databaseId	The database identifier for the time series data
cohortTimeSeriesArgs	The cohort time series arguments. @seealso[createCohortTimeSeriesArgs] for more information.

Details

This function will retrieve a data frame representing entry events for one or more cohorts grouped into time intervals as specified by the cohortTimeSeriesArgs

plotOcp — *Plots time series data with change points computed by the ocp package*

Description

Plots time series data with change points computed by the ocp package

Usage

```
plotOcp(
  tsData,
  model,
  plotTitle = "Bayesian Online Changepoint Detection",
  plotSubtitle = ""
)
```

Arguments

tsData	The time series data used to train the model
model	The trained ocp model
plotTitle	The title for the plot
plotSubtitle	The subtitle for the plot

Details

Create a plot to display the time series data along with any change points detected by the ocp package

plotSegmented — *Plots time series data with change points computed with a segmented generalized linear model*

Description

Plots time series data with change points computed with a segmented generalized linear model

Usage

```
plotSegmented(
  tsData,
  model,
  plotTitle = "Segmented Regression",
  plotSubtitle = ""
)
```

Arguments

tsData	The time series data used to train the model
model	The trained segmented model
plotTitle	The title for the plot
plotSubtitle	The subtitle for the plot

Details

Create a plot to display the time series data along with any change points detected by the segmented package

---

preMergeResultsFiles *Premerge Shiny results files*

Description

If there are many results files, starting the Shiny app may take a very long time. This function already does most of the preprocessing, increasing loading speed. The merged data will be stored in the same folder, and will automatically be recognized by the Shiny app.

Usage

```
preMergeResultsFiles(dataFolder)
```

Arguments

dataFolder	folder where the exported zip files are stored. Use the extractTimeSeriesData function to generate these zip files. Zip files containing results from multiple databases can be placed in the same folder.

---

runTsAnalyses *Run the time series analyses for the selected time series data set*

Description

Run the time series analyses for the selected time series data set

Usage

```
runTsAnalyses(
  tsData,
  tsDataId,
  outputFolder = "./TimeSeriesAnalysis",
  tsAnalysisList
)
```

Arguments

tsData	A vector representing the time series data
tsDataId	An integer value to uniquely identify the time series data. Most often this value will be the cohort ID that represents the cohort time series data.
outputFolder	The folder where the time series models are saved
tsAnalysisList	A list of time series analyses as specified by using the @seealso [createSegmentedArgs] and/or @seealso [createOcpArgs]

Details

This function will run the list of time series analyses for the time series data and save the output to the outputFolder.

Index