

# Package ‘Cyclops’

December 5, 2014

**Type** Package

**Title** Cyclic coordinate descent for logistic, Poisson and survival analysis

**Version** 0.1

**Date** 2014-03-26

**Author** Marc A. Suchard

**Maintainer** Marc A. Suchard <msuchard@gmail.com>

**Description** This package incorporates cyclic coordinate descent and majorization-minimization approaches to fit a variety of regression models found in observational healthcare data. Implementations focus on computational optimization and fine-scale parallelization to yield efficient inference in massive datasets.

**License** Apache License 2.0

**LazyData** Yes

**Depends** Rcpp (>= 0.10.6),BH (>= 1.51.0),RcppEigen (>= 0.3.2),Matrix

**Imports** ff,ffbase

**LinkingTo** Rcpp,BH,RcppEigen

**Suggests** testthat,survival,gnm

## R topics documented:

aconfint . . . . .	2
appendSqlCyclopsData . . . . .	3
coef.cyclopsFit . . . . .	4
confint.cyclopsFit . . . . .	4
convertToCyclopsDataObject . . . . .	5
convertToCyclopsVariance . . . . .	6
convertToGlmnetLambda . . . . .	7
createControl . . . . .	7
createCyclopsDataFrame . . . . .	8
createPrior . . . . .	10
createSqlCyclopsData . . . . .	11

finalizeSqlCyclopsData . . . . .	11
fitCyclopsModel . . . . .	12
getCovariateIds . . . . .	13
getCovariateTypes . . . . .	13
getHyperParameter . . . . .	14
getNumberOfCovariates . . . . .	14
getNumberOfRows . . . . .	14
getNumberOfStrata . . . . .	15
getNumberOfTypes . . . . .	15
getSEs . . . . .	15
isInitialized . . . . .	16
isSorted . . . . .	16
isValidModelType . . . . .	17
logLik.cyclopsFit . . . . .	17
predict.cyclopsFit . . . . .	18
print.cyclopsData . . . . .	18
print.cyclopsFit . . . . .	19
printCyclopsRowIds . . . . .	19
readCyclopsData . . . . .	19
reduce . . . . .	21
summary.cyclopsData . . . . .	22
vcov.cyclopsFit . . . . .	22

<b>Index</b>	<b>23</b>
--------------	-----------

---

aconfint	<i>Asymptotic confidence intervals for a fitted Cyclops model object</i>
----------	--

---

## Description

aconfint constructs confidence intervals of arbitrary level using asymptotic standard error estimates.

## Usage

```
aconfint(object, parm, level = 0.95, control,
         overrideNoRegularization = FALSE, ...)
```

## Arguments

object	A fitted Cyclops model object
parm	A specification of which parameters require confidence intervals, either a vector of numbers of covariateId names
level	Numeric: confidence level required
control	A Cyclops <a href="#">control</a> object
overrideNoRegularization	Logical: Enable confidence interval estimation for regularized parameters
...	Additional argument(s) for methods

**Value**

A matrix with columns reporting lower and upper confidence limits for each parameter. These columns are labelled as (1-level) / 2 and 1 - (1 - level) / 2 in (by default 2.5

---

```
appendSqlCyclopsData  appendSqlCyclopsData
```

---

**Description**

appendSqlCyclopsData appends data to an OHDSI data object.

**Usage**

```
appendSqlCyclopsData(object, oStratumId, oRowId, oY, oTime, cRowId,
  cCovariateId, cCovariateValue)
```

**Arguments**

object	OHDSI Cyclops data object to append entries
oStratumId	Integer vector (optional): non-unique stratum identifier for each row in outcomes table
oRowId	Integer vector: unique row identifier for each row in outcomes table
oY	Numeric vector: model outcome variable for each row in outcomes table
oTime	Numeric vector (optional): exposure interval or censoring time for each row in outcomes table
cRowId	Integer vector: non-unique row identifier for each row in covariates table that matches a single outcomes table entry
cCovariateId	Integer vector: covariate identifier
cCovariateValue	Numeric vector: covariate value

**Details**

Append data using two tables. The outcomes table is dense and contains ... The covariates table is sparse and contains ... All entries in the outcome table must be sorted in increasing order by oStratumId, oRowId. All entries in the covariate table must be sorted in increasing order by cRowId. Each cRowId value must match exactly one oRowId value.

---

coef.cyclopsFit	<i>Extract model coefficients</i>
-----------------	-----------------------------------

---

### Description

coef.cyclopsFit extracts model coefficients from an OHDSI Cyclops model fit object

### Usage

```
## S3 method for class 'cyclopsFit'
coef(object, ...)
```

### Arguments

object	OHDSI Cyclops model fit object
...	Other arguments

### Value

Named numeric vector of model coefficients.

---

confint.cyclopsFit	<i>confint.cyclopsFit</i>
--------------------	---------------------------

---

### Description

confint.cyclopsFit profiles the data likelihood to construct confidence intervals of arbitrary level. TODO: Profile data likelihood or joint distribution of remaining parameters.

### Usage

```
## S3 method for class 'cyclopsFit'
confint(object, parm, level = 0.95, control,
  overrideNoRegularization = FALSE, includePenalty = FALSE, ...)
```

### Arguments

object	A fitted Cyclops model object
parm	A specification of which parameters require confidence intervals, either a vector of numbers of covariateId names
level	Numeric: confidence level required
control	A Cyclops <a href="#">control</a> object
overrideNoRegularization	Logical: Enable confidence interval estimation for regularized parameters
includePenalty	Logical: Include regularized covariate penalty in profile
...	Additional argument(s) for methods

### Value

A matrix with columns reporting lower and upper confidence limits for each parameter. These columns are labelled as  $(1 - \text{level}) / 2$  and  $1 - (1 - \text{level}) / 2$  in (by default 2.5

---

convertToCyclopsDataObject

*Convert data from two data frames or ffdof objects into a CyclopsData object*

---

## Description

convertToCyclopsDataObject loads data from two data frames or ffdof objects, and inserts it into a Cyclops data object.

## Usage

```
convertToCyclopsDataObject(outcomes, covariates, modelType = "lr",
  addIntercept = TRUE, offsetAlreadyOnLogScale = FALSE,
  makeCovariatesDense = NULL, checkSorting = TRUE, checkRowIds = TRUE,
  quiet = FALSE)
```

## Arguments

outcomes	A data frame or ffdof object containing the outcomes with predefined columns (see below).
covariates	A data frame or ffdof object containing the covariates with predefined columns (see below).
modelType	Cyclops model type. Current supported types are "pr", "cpr", "lr", "clr", or "cox"
addIntercept	Add an intercept to the model?
useOffsetCovariate	Use the time variable in the model as an offset?
offsetAlreadyOnLogScale	Is the time variable already on a log scale?
checkSorting	Check if the data is sorted appropriately, and if not, sort.
checkRowIds	Check if all rowIds in the covariates appear in the outcomes.
quiet	If true, (warning) messages are suppressed.

## Details

These columns are expected in the outcome object:

stratumId	(integer)	(optional) Stratum ID for conditional regression models
rowId	(integer)	Row ID is used to link multiple covariates (x) to a single outcome (y)
y	(real)	The outcome variable
time	(real)	For models that use time (e.g. Poisson or Cox regression) this contains time (e.g. number of days)

These columns are expected in the covariates object:

stratumId	(integer)	(optional) Stratum ID for conditional regression models
rowId	(integer)	Row ID is used to link multiple covariates (x) to a single outcome (y)
covariateId	(integer)	A numeric identifier of a covariate
covariateValue	(real)	The value of the specified covariate

Note: If checkSorting is turned off, the outcome table should be sorted by stratumId (if present) and then rowId except for Cox regression when the table should be sorted by stratumId (if present), -time, y, and rowId. The covariate table should be sorted by stratumId (if present), rowId and covariateId except for Cox regression when the table should be sorted by stratumId (if present), -time, y, and rowId.

## Value

An object of type cyclopsData

## Examples

```
#Convert infert dataset to Cyclops format:
covariates <- data.frame(stratumId = rep(infert$stratum,2),
                        rowId = rep(1:nrow(infert),2),
                        covariateId = rep(1:2,each=nrow(infert)),
                        covariateValue = c(infert$spontaneous,infert$induced))
outcomes <- data.frame(stratumId = infert$stratum,
                      rowId = 1:nrow(infert),
                      y = infert$case)

#Make sparse:
covariates <- covariates[covariates$covariateValue != 0,]

#Create Cyclops data object:
cyclopsData <- convertToCyclopsDataObject(outcomes,covariates,modelType = "clr",addIntercept = FALSE)

#Fit model:
fit <- fitCyclopsModel(cyclopsData,prior = createPrior("none"))
```

---

convertToCyclopsVariance

*Convert to Cyclops Prior Variance*

---

## Description

convertToCyclopsVariance converts the regularization parameter lambda from glmnet into a prior variance.

## Usage

```
convertToCyclopsVariance(lambda, nobs)
```

## Arguments

lambda	Regularization parameter from glmnet
nobs	Number of observation rows in dataset

## Value

Prior variance under a Laplace() prior

---

convertToGlmnetLambda    *Convert to glmnet regularization parameter*

---

### Description

convertToGlmnetLambda converts a prior variance from Cyclops into the regularization parameter lambda.

### Usage

```
convertToGlmnetLambda(variance, nobs)
```

### Arguments

variance	Prior variance
nobs	Number of observation rows in dataset

### Value

lambda

---

createControl    *control*

---

### Description

control builds a Cyclops control object

### Usage

```
createControl(maxIterations = 1000, tolerance = 1e-06,
  convergenceType = "gradient", cvType = "grid", fold = 10,
  lowerLimit = 0.01, upperLimit = 20, gridSteps = 10, cvRepetitions = 1,
  minCVData = 100, noiseLevel = "silent", seed = NULL)
```

### Arguments

maxIterations	Integer: maximum iterations of Cyclops to attempt before returning a failed-to-converge error
tolerance	Numeric: maximum relative change in convergence criterion from successive iterations to achieve convergence
convergenceType	String: name of convergence criterion to employ (described in more detail below)
cvType	String: name of cross validation search. Option "auto" selects an auto-search following BBR. Option "grid" selects a grid-search cross validation
fold	Numeric: Number of random folds to employ in cross validation
lowerLimit	Numeric: Lower prior variance limit for grid-search

upperLimit	Numeric: Upper prior variance limit for grid-search
gridSteps	Numeric: Number of steps in grid-search
cvRepetitions	Numeric: Number of repetitions of X-fold cross validation
minCVData	Numeric: Minimum number of data for cross validation
noiseLevel	String: level of Cyclops screen output ("silent", "quiet", "noisy")
seed	Numeric: Specify random number generator seed. A null value sets seed via <code>Sys.time</code> .

### Value

A Cyclops convergence criteria object of class inheriting from "cyclopsConvergence" for use with `fitCyclopsModel`.

### Criteria

TODO

### Examples

```
## Not run:
# Add cross-validation example

## End(Not run)
```

---

```
createCyclopsDataFrame
      createCyclopsDataFrame
```

---

### Description

`createCyclopsDataFrame` creates a Cyclops model data object from an R formula

### Usage

```
createCyclopsDataFrame(formula, sparseFormula, indicatorFormula, modelType,
  data, subset, weights, offset, time = NULL, pid = NULL, y = NULL,
  type = NULL, dx = NULL, sx = NULL, ix = NULL, model = FALSE,
  method = "cyclops.fit")
```

### Arguments

modelType	character string: Valid types are listed below.
formula	An object of class "formula" that provides a symbolic description of the numerically dense model response and terms.
sparseFormula	An object of class "formula" that provides a symbolic description of numerically sparse model terms.
indicatorFormula	An object of class "formula" that provides a symbolic description of {0,1} model terms.



data	An optional data frame, list or environment containing the variables in the model.
subset	Currently unused
weights	Currently unused
offset	Currently unused
pid	Optional vector of integer stratum identifiers. If supplied, all rows must be sorted by increasing identifiers
y	Currently undocumented
time	Currently undocumented
z	Currently unused
dx	Optional dense "Matrix" of covariates
sx	Optional sparse "Matrix" of covariates
ix	Optional {0,1} "Matrix" of covariates
model	Currently undocumented
method	Currently undocumented

## Details

This function creates a Cyclops model data object from R "formula" or directly from numeric vectors and matrices to define the model response and covariates. If specifying a model using a "formula", then the left-hand side define the model response and the right-hand side defines dense covariate terms. Objects provided with "sparseFormula" and "indicatorFormula" must be include left-hand side responses and terms are coerced into sparse and indicator representations for computational efficiency.

Items to discuss: \* Only use formula or (y,dx,...) \* stratum() in formula \* offset() in formula \* when "stratum" (renamed from pid) are necessary \* when "time" are necessary

## Value

A list that contains a Cyclops model data object pointer and an operation duration

## Models

Currently supported model types are:

"ls"	Least squares
"pr"	Poisson regression
"lr"	Logistic regression
"clr"	Conditional logistic regression
"cpr"	Conditional Poisson regression
"sccs"	Self-controlled case series
"cox"	Cox proportional hazards regression

## Examples

```
## Dobson (1990) Page 93: Randomized Controlled Trial :
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
cyclopsData <- createCyclopsDataFrame(
```

```

      counts ~ outcome + treatment,
      modelType = "pr")
cyclopsFit <- fitCyclopsModel(cyclopsData)

cyclopsData2 <- createCyclopsDataFrame(
  counts ~ outcome,
  indicatorFormula = ~ treatment,
  modelType = "pr")
summary(cyclopsData2)
cyclopsFit2 <- fitCyclopsModel(cyclopsData2)

```

---

createPrior	<i>prior</i>
-------------	--------------

---

## Description

`prior` builds a Cyclops prior object

## Usage

```

createPrior(priorType, variance = 1, exclude = c(), graph = NULL,
  useCrossValidation = FALSE, forceIntercept = FALSE)

```

## Arguments

<code>priorType</code>	Character: specifies prior distribution. See below for options
<code>variance</code>	Numeric: prior distribution variance
<code>exclude</code>	A vector of numbers or covariateId names to exclude from prior
<code>useCrossValidation</code>	Logical: Perform cross-validation to determine prior variance.
<code>forceIntercept</code>	Logical: Force intercept coefficient into prior

## Value

A Cyclops prior object of class inheriting from "cyclopsPrior" for use with `fitCyclopsModel`.

## Prior types

We specify all priors in terms of their variance parameters. Similar fitting tools for regularized regression often parameterize the Laplace distribution in terms of a rate "lambda" per observation. See "glmnet", for example.

$\text{variance} = 2 * / (\text{nobs} * \text{lambda})^2$  or  $\text{lambda} = \text{sqrt}(2 / \text{variance}) / \text{nobs}$

---

```
createSqlCyclopsData
```

*Create an OHDSI Cyclops data object from SQL input*

---

### Description

createSqlCyclopsData creates an empty OHDSI Cyclops data object into which data can be appended in chunks.

### Usage

```
createSqlCyclopsData(modelType, control)
```

### Arguments

modelType	character string: Valid types are listed below.
control	An OHDSI Cyclops fit control object (optional)

### Models

Currently supported model types are:

"ls"	Least squares
"pr"	Poisson regression
"lr"	Logistic regression
"clr"	Conditional logistic regression
"cpr"	Conditional Poisson regression
"sccs"	Self-controlled case series
"cox"	Cox proportional hazards regression

---

```
finalizeSqlCyclopsData
```

*finalizeSqlCyclopsData*

---

### Description

finalizeSqlCyclopsData finalizes a Cyclops data object

### Usage

```
finalizeSqlCyclopsData(object, addIntercept = FALSE,
  useOffsetCovariate = NULL, offsetAlreadyOnLogScale = FALSE,
  sortCovariates = FALSE, makeCovariatesDense = NULL)
```

### Arguments

object	Cyclops data object
addIntercept	Add an intercept covariate if one was not imported through SQL

useOffsetCovariate	Specify is a covariate should be used as an offset (fixed coefficient = 1). Set option to "useTime" to specify the time-to-event column, otherwise include a single numeric or character covariate name.
offsetAlreadyOnLogScale	Set to TRUE to indicate that offsets were log-transformed before importing into Cyclops data object.
sortCovariates	Sort covariates in numeric-order with intercept first if it exists.
makeCovariatesDense	List of numeric or character covariates names to densely represent in Cyclops data object. For efficiency, we suggest making atleast the intercept dense.

---

fitCyclopsModel	<i>fitCyclopsModel</i>
-----------------	------------------------

---

## Description

fitCyclopsModel fits a Cyclops model data object

## Usage

```
fitCyclopsModel(cyclopsData, prior, control, forceColdStart = FALSE,
  returnEstimates = TRUE)
```

## Arguments

cyclopsData	An OHDSI data object
prior	A prior object. More details are given below.
control	OHDSI control object, see " <a href="#">control</a> "
forceColdStart	Logical, forces fitting algorithm to restart at regression coefficients = 0
returnEstimates	Logical, return regression coefficient estimates in Cyclops model fit object

## Details

This function performs numerical optimization to fit a Cyclops model data object.

## Value

A list that contains a Cyclops model fit object pointer and an operation duration

## Prior

Currently supported prior types are:

"none"	Useful for finding MLE
"laplace"	L_1 regularization
"normal"	L_2 regularization

## References

Suchard MA, Simpson SE, Zorych I, Ryan P, Madigan D. Massive parallelization of serial inference algorithms for complex generalized linear models. *ACM Transactions on Modeling and Computer Simulation*, 23, 10, 2013.

Simpson SE, Madigan D, Zorych I, Schuemie M, Ryan PB, Suchard MA. Multiple self-controlled case series for large-scale longitudinal observational databases. *Biometrics*, 69, 893-902, 2013.

Mittal S, Madigan D, Burd RS, Suchard MA. High-dimensional, massive sample-size Cox proportional hazards regression for survival analysis. *Biostatistics*, 15, 207-221, 2014.

## Examples

```
## Dobson (1990) Page 93: Randomized Controlled Trial :
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
cyclopsData <- createCyclopsDataFrame(counts ~ outcome + treatment, modelType = "pr")
cyclopsFit <- fitCyclopsModel(cyclopsData, prior = createPrior("none"))
coef(cyclopsFit)
confint(cyclopsFit, c("outcome2", "treatment3"))
predict(cyclopsFit)
```

---

getCovariateIds	<i>Get covariate identifiers</i>
-----------------	----------------------------------

---

## Description

getCovariateIds returns a vector of integer covariate identifiers in an OHDSI Cyclops data object

## Usage

```
getCovariateIds(object)
```

## Arguments

object	An OHDSI Cyclops data object
--------	------------------------------

---

getCovariateTypes	<i>Get covariate types</i>
-------------------	----------------------------

---

## Description

getCovariateTypes returns a vector covariate types in an OHDSI Cyclops data object

## Usage

```
getCovariateTypes(object, covariateLabel)
```

## Arguments

object	An OHDSI Cyclops data object
covariateLabel	Integer vector: covariate identifiers to return

---

getHyperParameter	<i>Get hyperparameter</i>
-------------------	---------------------------

---

**Description**

getHyperParameter returns the current hyper parameter in an OHDSI Cyclops model fit object

**Usage**

```
getHyperParameter(object)
```

**Arguments**

object	An OHDSI Cyclops model fit object
--------	-----------------------------------

---

getNumberOfCovariates	<i>Get total number of covariates</i>
-----------------------	---------------------------------------

---

**Description**

getNumberOfCovariates returns the total number of covariates in an OHDSI Cyclops data object

**Usage**

```
getNumberOfCovariates(object)
```

**Arguments**

object	An OHDSI Cyclops data object
--------	------------------------------

---

getNumberOfRows	<i>Get total number of rows</i>
-----------------	---------------------------------

---

**Description**

getNumberOfRows returns the total number of outcome rows in an OHDSI Cyclops data object

**Usage**

```
getNumberOfRows(object)
```

**Arguments**

object	An OHDSI Cyclops data object
--------	------------------------------

---

getNumberOfStrata	<i>Get number of strata</i>
-------------------	-----------------------------

---

**Description**

getNumberOfStrata return the number of unique strata in an OHDSI Cyclops data object

**Usage**

```
getNumberOfStrata(object)
```

**Arguments**

object	An OHDSI Cyclops data object
--------	------------------------------

---

getNumberOfTypes	<i>Get total number of outcome types</i>
------------------	--

---

**Description**

getNumberOfTypes returns the total number of outcome types in an OHDSI Cyclops data object

**Usage**

```
getNumberOfTypes(object)
```

**Arguments**

object	An OHDSI Cyclops data object
--------	------------------------------

---

getSEs	<i>Extract standard errors</i>
--------	--------------------------------

---

**Description**

getSEs extracts asymptotic standard errors for specific covariates from an OHDSI Cyclops model fit object.

**Usage**

```
getSEs(object, covariates)
```

**Arguments**

object	An OHDSI Cyclops model fit object
covariates	Integer or string vector: list of covariates for which asymptotic standard errors are wanted

Details

This function first computes the (partial) Fisher information matrix for just the requested covariates and then returns the square root of the diagonal elements of the inverse of the Fisher information matrix. These are the asymptotic standard errors when all possible covariates are included. When the requested covariates do not equate to all coefficients in the model, then interpretation is more challenging.

Value

Vector of standard error estimates

---

isInitialized	<i>isInitialized</i>
---------------	----------------------

---

Description

isInitialized determines if an OHDSI data object is properly initialized and remains in memory. OHSDI data objects do not serialized/deserialize their back-end memory across R sessions.

Usage

isInitialized(object)

Arguments

object                      OHDSI data object to test

---

isSorted	<i>Check if data is sorted by one or more columns</i>
----------	---

---

Description

isSorted checks wether data is sorted by one or more specified columns.

Usage

isSorted(data, columnNames, ascending = rep(TRUE, length(columnNames)))

Arguments

data                      Either a data.frame of ffdof object.  
columnNames              Vector of one or more column names.  
ascending                Logical vector indicating the data should be sorted ascending or descending according the specified columns.

Details

This function currently only supports checking for sorting on numeric values.



**Value**

True or false

**Examples**

```
x <- data.frame(a = runif(1000), b = runif(1000))
x <- round(x, digits=2)
isSorted(x, c("a", "b"))

x <- x[order(x$a, x$b), ]
isSorted(x, c("a", "b"))

x <- x[order(x$a, -x$b), ]
isSorted(x, c("a", "b"), c(TRUE, FALSE))
```

---

isValidModelType	<i>isValidModelType</i>
------------------	-------------------------

---

**Description**

isValidModelType checks for a valid Cyclops model type

**Usage**

```
isValidModelType(modelType)
```

**Arguments**

modelType      character string: Valid types are listed below.

**Value**

TRUE/FALSE

**Models**

Currently supported model types are:

"ls"	Least squares
"pr"	Poisson regression
"lr"	Logistic regression
"clr"	Conditional logistic regression
"cpr"	Conditional Poisson regression
"sccs"	Self-controlled case series
"cox"	Cox proportional hazards regression

---

logLik.cyclopsFit	<i>Extract log-likelihood</i>
-------------------	-------------------------------

---

**Description**

logLik returns the current log-likelihood of the fit in an OHDSI Cyclops model fit object

**Usage**

```
## S3 method for class 'cyclopsFit'
logLik(object, ...)
```

**Arguments**

object	An OHDSI Cyclops model fit object
...	Additional arguments

---

predict.cyclopsFit	<i>Model predictions</i>
--------------------	--------------------------

---

**Description**

predict.cyclopsFit computes model response-scale predictive values for all data rows

**Usage**

```
## S3 method for class 'cyclopsFit'
predict(object, ...)
```

**Arguments**

object	An OHDSI Cyclops model fit object
...	Additional arguments

---

print.cyclopsData	<i>Print an OHDSI Cyclops data model object</i>
-------------------	---

---

**Description**

print.cyclopsData displays information about an OHDSI Cyclops data model object

**Usage**

```
## S3 method for class 'cyclopsData'
print(x, show.call = TRUE, ...)
```

**Arguments**

x	An OHDSI Cyclops data model object
show.call	Logical: display last call to construct the OHDSI Cyclops data model object
...	Additional arguments

---

print.cyclopsFit	<i>Print an OHDSI Cyclops model fit object</i>
------------------	--

---

**Description**

print.cyclopsFit displays information about an OHDSI Cyclops model fit object

**Usage**

```
## S3 method for class 'cyclopsFit'
print(x, show.call = TRUE, ...)
```

**Arguments**

x	An OHDSI Cyclops model fit object
show.call	Logical: display last call to update the OHDSI Cyclops model fit object
...	Additional arguments

---

printCyclopsRowIds	<i>Print row identifiers</i>
--------------------	------------------------------

---

**Description**

printCcdRowIds return the row identifiers in an OHDSI Cyclops data object

**Usage**

```
printCyclopsRowIds(object)
```

**Arguments**

object	An OHDSI Cyclops data object
--------	------------------------------

---

readCyclopsData	<i>readCyclopsData</i>
-----------------	------------------------

---

**Description**

readCyclopsData reads a Cyclops-formatted text file

**Usage**

```
readCyclopsData(fileName, modelType)
```

## Arguments

modelType	character string: Valid types are listed below.
fileName	Name of text file to be read. If fileName does not contain an absolute path, the name is relative to the current working directory, <a href="#">getwd</a> .

## Details

This function reads a Cyclops-formatted text file and returns a Cyclops data object. The first line of the file may start with '#', indicating that it contains header options. Valid header options are:

row_label	(assume file contains a numeric column of unique row identifiers)
stratum_label	(assume file contains a numeric column of stratum identifiers)
weight	(assume file contains a column of row-specific model weights, currently unused)
offset	(assume file contains a dense column of linear predictor offsets)
bbr_outcome	(assume logistic outcomes are encoded -1/+1 following BBR)
log_offset	(assume file contains a dense column of values $x_i$ for which $\log(x_i)$ is the offset)
add_intercept	(automatically include an intercept column of all 1s for each entry)
indicator_only	(assume all covariates 0/1-valued and only covariate name is given)
sparse	(force all BBR formatted covariates to be represented as sparse, instead of sparse-indicator, columns .. really only for debugging)
dense	(force all BBR formatted covariates to be represented as dense columns .. really only for debugging)

Successive lines of the file are white-space delimited and follow the format:

[Row ID] {Stratum ID} [Weight] <Outcome> {Censored} {Offset} <BBR covariates>

- [optional]
- <required>
- {required or optional depending on model}

Bayesian binary regression (BBR) covariates are white-space delimited and generally in a sparse '<name>:<value>' format, where 'name' must (currently) be numeric and 'value' is non-zero. If option 'indicator\_only' is specified, then format is simply '<name>'. 'Row ID' and 'Stratum ID' must be numeric, and rows must be sorted such that equal 'Stratum ID' are consecutive. 'Stratum ID' is required for 'clr' and 'scs' models. 'Censored' is required for a 'cox' model. 'Offset' is (currently) required for a 'scs' model.

## Value

Returns an [environment](#) of class inheriting from "ccdData" that contains at least the following objects:

ccdDataPtr	An Rcpp externalptr to the C++ object that holds the data. The object contents does not get written to disk.
ccdInterfacePtr	An Rcpp externalptr to the C++ object that fits the data. Again, this is not reusable across R invocations.
timeLoad	Amount of time (in seconds) taken to load the data.

The generic functions [print](#) and [summary](#) can be used to display various summaries of the data.

## Models

Currently supported model types are:

"ls"	Least squares
"pr"	Poisson regression
"lr"	Logistic regression
"clr"	Conditional logistic regression
"cpr"	Conditional Poisson regression
"sccs"	Self-controlled case series
"cox"	Cox proportional hazards regression

## Examples

```
dataPtr = readCyclopsData(system.file("extdata/infert_ccd.txt", package="Cyclops"), "clr")
```

---

reduce	<i>Apply simple data reductions</i>
--------	-------------------------------------

---

## Description

reduce reports the count of non-zero elements, sum and sum-of-squares for specified covariates in an OHDSI data object.

## Usage

```
reduce(object, covariates, groupBy, power = 1)
```

## Arguments

object	An OHDSI Cyclops data object
covariates	Integer or string vector: list of covariates to report
groupBy	Integer or string (optional): generates a segmented reduction stratified by this covariate. Setting groupBy = "stratum" segments reduction for strataID
power	Integer: 0 = non-zero count, 1 = sum, 2 = sum-of-squares

## Value

Specified reduction as number or data.frame if segmented.

---

summary.cyclopsData	<i>OHDSI Cyclops data object summary</i>
---------------------	--

---

### Description

summary.cyclopsData summarizes the data held in an OHDSI Cyclops data object.

### Usage

```
## S3 method for class 'cyclopsData'
summary(object, ...)
```

### Arguments

object	An OHDSI Cyclops data object
...	Additional arguments

### Value

Returns a data.frame that reports simply summarize statistics for each covariate in an OHDSI Cyclops data object.

---

vcov.cyclopsFit	<i>Calculate variance-covariance matrix for a fitted Cyclops model object</i>
-----------------	---

---

### Description

vcov.cyclopsFit returns the variance-covariance matrix for all covariates of a Cyclops model object

### Usage

```
## S3 method for class 'cyclopsFit'
vcov(object, control, overrideNoRegularization = FALSE,
      ...)
```

### Arguments

object	A fitted Cyclops model object
control	A Cyclops <a href="#">control</a> object
overrideNoRegularization	Logical: Enable variance-covariance estimation for regularized parameters
...	Additional argument(s) for methods

### Value

A matrix of the estimates covariances between all covariate estimates.

# Index

aconfint, [2](#)  
appendSqlCyclopsData, [3](#)  
  
coef.cyclopsFit, [4](#)  
confint.cyclopsFit, [4](#)  
control, [2](#), [4](#), [12](#), [22](#)  
convertToCyclopsDataObject, [5](#)  
convertToCyclopsVariance, [6](#)  
convertToGlmnetLambda, [7](#)  
createControl, [7](#)  
createCyclopsDataFrame, [8](#)  
createPrior, [10](#)  
createSqlCyclopsData, [11](#)  
  
environment, [20](#)  
  
finalizeSqlCyclopsData, [11](#)  
fitCyclopsModel, [12](#)  
formula, [8](#), [9](#)  
  
getCovariateIds, [13](#)  
getCovariateTypes, [13](#)  
getHyperParameter, [14](#)  
getNumberOfCovariates, [14](#)  
getNumberOfRows, [14](#)  
getNumberOfStrata, [15](#)  
getNumberOfTypes, [15](#)  
getSEs, [15](#)  
getwd, [20](#)  
  
isInitialized, [16](#)  
isSorted, [16](#)  
isValidModelType, [17](#)  
  
logLik.cyclopsFit, [17](#)  
  
Matrix, [9](#)  
  
predict.cyclopsFit, [18](#)  
print, [20](#)  
print.cyclopsData, [18](#)  
print.cyclopsFit, [19](#)  
printCyclopsRowIds, [19](#)  
  
readCyclopsData, [19](#)  
  
reduce, [21](#)  
  
summary, [20](#)  
summary.cyclopsData, [22](#)  
Sys.time, [8](#)  
  
vcov.cyclopsFit, [22](#)