

# Package ‘bigReg’

October 12, 2022

**Type** Package

**Title** Generalized Linear Models (GLM) for Large Data Sets

**Version** 0.1.2

**Date** 2016-07-23

**Author** Chibisi Chima-Okereke <chibisi@active-analytics.com>

**Maintainer** Chibisi Chima-Okereke <chibisi@active-analytics.com>

**Description** Allows the user to carry out GLM on very large data sets. Data can be created using the `data_frame()` function and appended to the object with `object$append(data)`; `data_frame` and `data_matrix` objects are available that allow the user to store large data on disk. The data is stored as doubles in binary format and any character columns are transformed to factors and then stored as numeric (binary) data while a look-up table is stored in a separate `.meta_data` file in the same folder. The data is stored in blocks and GLM regression algorithm is modified and carries out a MapReduce-like algorithm to fit the model. The functions `bglm()`, and `summary()` and `bglm_predict()` are available for creating and post-processing of models. The library requires Armadillo installed on your system. It probably won't function on windows since multi-core processing is done using `mclapply()` which forks R on Unix/Linux type operating systems.

**License** GPL (>= 2)

**Depends** R (>= 3.2.0), Rcpp (>= 0.12.3), parallel, methods, stats, uuid (>= 0.1-2), MASS (>= 7.3-39)

**LinkingTo** Rcpp, RcppArmadillo (>= 0.5.200.1.0)

**OS\_type** unix

**Collate** 'RcppExports.R' 'data\_frame.r' 'data\_matrix.r' 'family.r' 'map\_reg.r'

**RoxygenNote** 5.0.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2016-07-25 19:16:58

**R topics documented:**

asInteger . . . . .	3
bglm . . . . .	3
bglm_predict . . . . .	4
binomial_ . . . . .	4
blm . . . . .	5
CreateFactor . . . . .	5
data_frame . . . . .	6
data_matrix . . . . .	7
family_ . . . . .	7
Gamma_ . . . . .	8
gaussian_ . . . . .	8
inverse.gaussian_ . . . . .	8
load_data_frame . . . . .	9
load_data_matrix . . . . .	9
myIn . . . . .	9
mySeq . . . . .	10
plasma . . . . .	10
poisson_ . . . . .	11
print.bglm . . . . .	11
print.blm . . . . .	12
print.data_frame . . . . .	12
print.data_matrix . . . . .	13
print.summary.bglm . . . . .	13
print.summary.blm . . . . .	14
process_bglm_block . . . . .	14
quasibinomial_ . . . . .	15
quasipoisson_ . . . . .	15
quasi_ . . . . .	15
readNumericVector . . . . .	16
read_df_block . . . . .	16
read_df_blocks . . . . .	17
read_matrix_block . . . . .	17
read_matrix_blocks . . . . .	18
r_bind . . . . .	18
summary.bglm . . . . .	19
summary.blm . . . . .	19
sum_bglm_block . . . . .	20
SVD . . . . .	20
writeNumericVector . . . . .	21
write_numeric_vector . . . . .	21
XWXMatrix . . . . .	21
XWXMatrixW . . . . .	22

---

asInteger	<i>converts numeric vector to integer</i>
-----------	---

---

**Description**

converts numeric vector to integer

**Usage**

```
asInteger(x)
```

**Arguments**

x	numeric vector
---	----------------

---

bglm	<i>Function to carry out generalized linear regression on a data_frame data object</i>
------	--

---

**Description**

Function to carry out generalized linear regression on a data\_frame data object

**Usage**

```
bglm(formula, family = gaussian_(), data, weights = NULL, offset = NULL,
      start = NULL, control = list(), etastart = NULL, mustart = NULL)
```

**Arguments**

formula	formula that defines your regression model
family	family object from activeReg, e.g. .gaussian(), .binomial(), .poisson(), .quasipoisson(), .quasibinomial(), .Gamma(), .inverse.gaussian(), .quasi()
data	data_frame object containing data for linear regression
weights	weights for the model
offset	offsets for the model
start	starting values for the linear predictor
control	list of parameters for .control() function
etastart	starting values for the linear predictor
mustart	starting values for vector of means

**Examples**

```
require(parallel)
data("plasma", package = "bigReg")
plasma1 <- plasma
plasma1 <- data_frame(plasma1, 10, path = "outputs", nCores = 1)
plasma_glm <- bglm(ESR ~ fibrinogen + globulin, data = plasma1, family = binomial_("logit"))
summary(plasma_glm)
```

---

bglm_predict	<i>predict function for bglm object</i>
--------------	---

---

**Description**

predict function for bglm object

**Usage**

```
bglm_predict(mf = stop("mf: model frame must be supplied"),
             object = stop("object: bglm object must be supplied"),
             type = stop("type: either \"link\", \"response\", \"terms\""))
```

**Arguments**

mf	model frame
object	a bglm object
type	one of c("link", "response", "terms")

---

binomial_	<i>binomial family function</i>
-----------	---------------------------------

---

**Description**

binomial family function

**Usage**

```
binomial_(link = "logit")
```

**Arguments**

link	function character
------	--------------------

---

blm	<i>Function to carry out linear regression on a data_frame data object</i>
-----	--

---

**Description**

Function to carry out linear regression on a data\_frame data object

**Usage**

```
blm(formula = stop("formula: not supplied"),  
    data = stop("data: data not supplied"), control = list(),  
    weights = NULL, offset = NULL)
```

**Arguments**

formula	formula that defines your regression model
data	data_frame object containing data for linear regression
control	list of parameters for control() function
weights	weights for the model
offset	offsets for the model

---

CreateFactor	<i>creates factor from numeric vector and character vector as levels</i>
--------------	--

---

**Description**

The CreateFactor function creates a factor from a numeric vector and a character vector for levels

**Usage**

```
CreateFactor(x, levels)
```

**Arguments**

x	numeric vector containing the numeric indices of the levels
levels	character vector levels

---

data_frame	<i>function to create a data_frame object</i>
------------	---

---

### Description

function to create a data\_frame object. The data\_frame object is an object that is held on disk. It is written to a folder path on disk where the data is written to in blocks or chunks. The data is written in binary format using a C++ function in purely numerical data and a mapping to the table is held in a ".meta\_data" file in the folder. The table object accomodates numeric, factor, and character (converted to factor).

### Usage

```
data_frame(data = stop("data must be supplied"),
  chunkSize = stop("chunkSize must be specified, a good number is 50000"),
  path = stop("path must be specified"), nCores = parallel::detectCores(),
  ...)
```

### Arguments

data	data.frame object to be converted into a data_frame object
chunkSize	number of rows to be used in each chunk
path	character to folder where the object will be created
nCores	the number of cores to use defaults to parallel::detectCores()
...	not currently used.

### Details

Creates a data\_frame object

### Examples

```
irisA <- data_frame(iris[1:75,], 10, "irisA", nCores = 1)
irisA$append(iris[76:150,])
irisA$head()
irisA$tail(10)
irisA$delete(); rm(irisA)
```

---

data_matrix	<i>function to create a data_frame object</i>
-------------	---

---

**Description**

function to create a data\_matrix object. The data\_matrix object is an object that is held on disk. It is written to a folder path on disk where the data is written to in blocks or chunks. The data is written in binary format using a C++ function in purely numerical data.

**Usage**

```
data_matrix(data = stop("data: matrix must be supplied"),
            chunkSize = stop("chunkSize must be specified, a good number is 50000"),
            path = stop("path must be specified"), nCores = parallel::detectCores(),
            ...)
```

**Arguments**

data	object to be converted into a data_matrix object
chunkSize	number of rows to be used in each chunk
path	character to folder where the object will be created
nCores	the number of cores to use defaults to parallel::detectCores()
...	not used at the moment

**Details**

Creates a data\_matrix object

---

family_	<i>family function</i>
---------	------------------------

---

**Description**

family function

**Usage**

```
family_(distr, link)
```

**Arguments**

distr	distr character one of "binomial", "poisson", "gaussian", "quasipoisson", "quasi-binomial", "Gamma", "inverse.gaussian", "quasi"
link	function character

---

Gamma_	<i>Gamma family function</i>
--------	------------------------------

---

**Description**

Gamma family function

**Usage**

```
Gamma_(link = "inverse")
```

**Arguments**

link	function character
------	--------------------

---

gaussian_	<i>gaussian family function</i>
-----------	---------------------------------

---

**Description**

gaussian family function

**Usage**

```
gaussian_(link = "identity")
```

**Arguments**

link	function character
------	--------------------

---

inverse.gaussian_	<i>inverse.gaussian family function</i>
-------------------	---

---

**Description**

inverse.gaussian family function

**Usage**

```
inverse.gaussian_(link = "1/mu^2")
```

**Arguments**

link	function character
------	--------------------



---

load_data_frame	<i>function to load data_frame object</i>
-----------------	---

---

**Description**

function to load data\_frame object

**Usage**

```
load_data_frame(path = stop("path: to data_frame folder must be supplied"))
```

**Arguments**

path	character to folder containing object
------	---------------------------------------

---

---

load_data_matrix	<i>function to load data_frame object</i>
------------------	---

---

**Description**

function to load data\_frame object

**Usage**

```
load_data_matrix(path = stop("path: to data_matrix folder must be supplied"))
```

**Arguments**

path	character to folder containing object
------	---------------------------------------

---

---

myIn	<i>finds whether x is in y</i>
------	--------------------------------

---

**Description**

finds whether x is in y

**Usage**

```
myIn(x, y)
```

**Arguments**

x	item to be sought
y	vector to be matched against

---

mySeq	<i>mySeq function to sequence integers</i>
-------	--

---

**Description**

a function to create a sequence of integers

**Usage**

```
mySeq(start, end)
```

**Arguments**

start	integer from where sequence should start
end	integer where sequence should end

---

plasma	<i>plasma data from the HSAUR package</i>
--------	---

---

**Description**

Dataset from the HSAUR package

**Usage**

```
data(plasma)
```

**Format**

a data.frame

**Details**

...

**Source**

[HSAUR package](#)

**References**

HSAUR R package ([HSAUR package](#))

**Examples**

```
data(plasma)
head(plasma)
```

---

poisson_	<i>poisson family function</i>
----------	--------------------------------

---

**Description**

poisson family function

**Usage**

```
poisson_(link = "log")
```

**Arguments**

link	function character
------	--------------------

---

print.bglm	<i>print function for the bglm object</i>
------------	---

---

**Description**

print function for the bglm object

**Usage**

```
## S3 method for class 'bglm'  
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

**Arguments**

x	bglm object to be displayed
digits	number of significant digits to use
...	not yet used

---

`print.blm`                    *print function for the blm object*

---

**Description**

print function for the blm object

**Usage**

```
## S3 method for class 'blm'  
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

**Arguments**

<code>x</code>	blm object to be displayed
<code>digits</code>	number of significant digits to use
<code>...</code>	not yet used

---

`print.data_frame`            *print function for a data\_frame*

---

**Description**

print function for a data\_frame

**Usage**

```
## S3 method for class 'data_frame'  
print(x, ...)
```

**Arguments**

<code>x</code>	data_frame object to print
<code>...</code>	not used

---

print.data\_matrix      *print function for a data\_matrix*

---

**Description**

print function for a data\_matrix

**Usage**

```
## S3 method for class 'data_matrix'  
print(x, ...)
```

**Arguments**

x	data_matrix object to print
...	not used

---

print.summary.bglm      *Function to print the summary object from the bglm object*

---

**Description**

Function to print the summary object from the bglm object

**Usage**

```
## S3 method for class 'summary.bglm'  
print(x, digits = max(3L, getOption("digits") - 3L),  
      signif.stars = getOption("show.signif.stars"), ...)
```

**Arguments**

x	summary blm object
digits	- the digits to be displayed
signif.stars	passed to printCoefmat
...	arguments passed to printCoefmat() function

---

print.summary.blm      *Function to print the summary object from the blm object*

---

### Description

Function to print the summary object from the blm object

### Usage

```
## S3 method for class 'summary.blm'
print(x, digits = max(3L, getOption("digits") - 3L),
      signif.stars = getOption("show.signif.stars"), ...)
```

### Arguments

x	summary blm object
digits	- the digits to be displayed
signif.stars	passed to printCoefmat
...	arguments passed to printCoefmat() function

---

process\_bglm\_block      *Function to print the summary object from the blm object*

---

### Description

Function to print the summary object from the blm object

### Usage

```
process_bglm_block(mf, formula, mmCall, family, offset, weights, start, niter,
                  etastart, mustart)
```

### Arguments

mf	the data block to be processed
formula	the formula of for the model
mmCall	the call object of the model
family	the family object for the model
offset	the model offset
weights	the model weights
start	the starting coefficient estimates
niter	the current number of iterations
etastart	the start for eta
mustart	the start for mu

---

quasibinomial\_      *quasibinomial family function*

---

**Description**

quasibinomial family function

**Usage**

quasibinomial\_(link = "logit")

**Arguments**

link                  function character

---

quasipoisson\_      *quasipoisson family function*

---

**Description**

quasipoisson family function

**Usage**

quasipoisson\_(link = "log")

**Arguments**

link                  function character

---

quasi\_                  *quasi family function*

---

**Description**

quasi family function

**Usage**

quasi\_(link = "identity", variance = "constant")

**Arguments**

link                  function character  
variance              choice character

---

readNumericVector      *reads numeric vector to file*

---

**Description**

reads numeric vector to file

**Usage**

```
readNumericVector(size, filePath)
```

**Arguments**

size	the length of the numeric vector
filePath	dependent variable

---

read\_df\_block      *read data frame block from file*

---

**Description**

read data frame block from file

**Usage**

```
read_df_block(size, filePath, df, ncol, factors, factor_indices)
```

**Arguments**

size	number of elements in the block
filePath	path to where the block is stored
df	an empty list having the same number of elements as columns in the table
ncol	number of columns in the dataframe block
factors	list containing factors
factor_indices	numeric vector containing the indices that denote the factors



---

read_df_blocks	<i>read multiple blocks of data frames from file</i>
----------------	--

---

**Description**

read multiple blocks of data frames from file

**Usage**

```
read_df_blocks(size, filePaths, df, ncols, factors, factor_indices)
```

**Arguments**

size	number of elements in each block
filePaths	path to where the blocks are stored
df	an empty list having the same number of elements as columns in the table
ncols	number of columns in the dataframe block
factors	list containing factors
factor_indices	numeric vector containing the indices that denote the factors

---

read_matrix_block	<i>read matrix block from file</i>
-------------------	------------------------------------

---

**Description**

read matrix block from file

**Usage**

```
read_matrix_block(filePath, size, ncol)
```

**Arguments**

filePath	path to file where matrix should be read from
size	total number of elements to be read
ncol	number of columns in the matrix

---

<code>read_matrix_blocks</code>	<i>read matrix blocks from file</i>
---------------------------------	-------------------------------------

---

**Description**

read matrix blocks from file

**Usage**

```
read_matrix_blocks(filePaths, size, ncols)
```

**Arguments**

<code>filePaths</code>	file paths from where the matrix blocks will be read
<code>size</code>	numeric vector containing the number of elements in each block
<code>ncols</code>	number of columns in the matrix

---

<code>r_bind</code>	<i>row binding for benchmarking ...</i>
---------------------	---

---

**Description**

row binding for benchmarking

**Usage**

```
r_bind(x, y)
```

**Arguments**

<code>x</code>	first matrix to be bound together
<code>y</code>	second matrix to be bound together

---

summary.bglm	<i>summary function for the bglm object</i>
--------------	---

---

**Description**

summary function for the bglm object

**Usage**

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

**Arguments**

object	bglm object to be summarized
...	not used

---

summary.blm	<i>summary function for the blm object</i>
-------------	--

---

**Description**

summary function for the blm object

**Usage**

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

**Arguments**

object	blm object to be summarized
...	not used

---

sum_bglm_block	<i>The reduction function for the algorithm</i>
----------------	---

---

**Description**

The reduction function for the algorithm

**Usage**

```
sum_bglm_block(x1, x2)
```

**Arguments**

x1	the first list object to be reduced
x2	the second list object to be reduced

---

SVD	<i>Singular value decomposition of the aggregated list from XWXMatrix(W) functions</i>
-----	--

---

**Description**

Singular value decomposition of the aggregated list from XWXMatrix(W) functions

**Usage**

```
SVD(out, epsilon)
```

**Arguments**

out	list containing requisite computed values
epsilon	either machine epsilon or user determined epsilon

---

writeNumericVector     *writes numeric vector to file*

---

**Description**

writes numeric vector to file

**Usage**

```
writeNumericVector(v, filePath)
```

**Arguments**

v	numeric vector
filePath	dependent variable

---

write\_numeric\_vector     *writes numeric vector to file*

---

**Description**

writes numeric vector to file

**Usage**

```
write_numeric_vector(v, filePath)
```

**Arguments**

v	numeric vector to be written to file
filePath	path to file where the numeric vector should be written

---

XWXMatrix     *Calculation of iterative regression components*

---

**Description**

Calculation of iterative regression components

**Usage**

```
XWXMatrix(X, y)
```

**Arguments**

X	design matrix
y	dependent variable

---

`XWXMatrixW`*Calculation of iterative regression components*

---

**Description**

Calculation of iterative regression components

**Usage**

```
XWXMatrixW(X, y, W)
```

**Arguments**

<code>X</code>	design matrix
<code>y</code>	dependent variable
<code>W</code>	weights

# Index

- \* **datasets**
  - plasma, [10](#)
- asInteger, [3](#)
- bglm, [3](#)
- bglm\_predict, [4](#)
- binomial\_, [4](#)
- blm, [5](#)
- CreateFactor, [5](#)
- data\_frame, [6](#)
- data\_matrix, [7](#)
- family\_, [7](#)
- Gamma\_, [8](#)
- gaussian\_, [8](#)
- inverse.gaussian\_, [8](#)
- load\_data\_frame, [9](#)
- load\_data\_matrix, [9](#)
- myIn, [9](#)
- mySeq, [10](#)
- plasma, [10](#)
- poisson\_, [11](#)
- print.bglm, [11](#)
- print.blm, [12](#)
- print.data\_frame, [12](#)
- print.data\_matrix, [13](#)
- print.summary.bglm, [13](#)
- print.summary.blm, [14](#)
- process\_bglm\_block, [14](#)
- quasi\_, [15](#)
- quasibinomial\_, [15](#)
- quasipoisson\_, [15](#)
- r\_bind, [18](#)
- read\_df\_block, [16](#)
- read\_df\_blocks, [17](#)
- read\_matrix\_block, [17](#)
- read\_matrix\_blocks, [18](#)
- readNumericVector, [16](#)
- sum\_bglm\_block, [20](#)
- summary.bglm, [19](#)
- summary.blm, [19](#)
- SVD, [20](#)
- write\_numeric\_vector, [21](#)
- writeNumericVector, [21](#)
- XWXMatrix, [21](#)
- XWXMatrixW, [22](#)