

# Package ‘eflm’

October 13, 2022

**Title** Efficient Fitting of Linear and Generalized Linear Models

**Version** 0.3.0

**Description** Efficient Fitting of Linear and Generalized Linear Models by using just base R. As an alternative to `lm()` and `glm()`, this package provides `elm()` and `eglm()`, with a significant speedup when the number of observations is larger than the number of parameters to estimate. The speed gains are obtained by reducing the  $N \times P$  model matrix to a  $P \times P$  matrix, and the best computational performance is obtained when R is linked against 'OpenBLAS', 'Intel MKL' or other optimized 'BLAS' library. This implementation aims at being compatible with 'broom' and 'sandwich' packages for summary statistics and clustering by providing S3 methods.

**URL** <https://github.com/pachadotdev/eflm/>

**BugReports** <https://github.com/pachadotdev/eflm/issues/>

**License** Apache License ( $\geq 2$ )

**Imports** stats, tibble

**Suggests** broom, sandwich, testthat, patrick, rlang, covr

**Encoding** UTF-8

**Depends** R ( $\geq 4.0$ )

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Mauricio Vargas [aut, cre] (<<https://orcid.org/0000-0003-1017-7574>>),  
Alexey Kravchenko [ths],  
Constanza Prado [ths],  
Yoto Yotov [ths],  
Neal Fulz [rev] (independent code auditing),  
Rahul Saxena [ctb] (improved broom integration),  
The United Nations [dct, cph, fnd]

**Maintainer** Mauricio Vargas <mavargas11@uc.cl>

**Repository** CRAN

**Date/Publication** 2021-05-31 21:20:02 UTC

## R topics documented:

eglm . . . . .	2
eglm.wfit . . . . .	4
elm . . . . .	6
elm.wfit . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

eglm *Efficient Fitting of Generalized Linear Models*

---

### Description

Efficient Generalized Linear Model ("eglm") is used to fit generalized linear models in an equivalent way to "glm" but in a reduced time depending on the design matrix and the family (or link).

### Usage

```
eglm(
  formula,
  family = gaussian,
  data,
  weights,
  subset,
  na.action,
  start = NULL,
  etastart,
  mustart,
  offset,
  control = list(...),
  model = TRUE,
  method = "eglm.wfit",
  x = FALSE,
  y = TRUE,
  singular.ok = TRUE,
  contrasts = NULL,
  reduce = FALSE,
  ...
)
```

### Arguments

formula	an object of class " <a href="#">formula</a> " (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
---------	--

family	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. See <a href="#">family</a> for details of family functions.
data	an optional data frame, list or environment (or object coercible by <a href="#">as.data.frame</a> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lm</code> is called.
weights	an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If non-NULL, weighted least squares is used with weights <code>weights</code> (that is, minimizing $\sum(w \cdot e^2)$ ); otherwise ordinary least squares is used.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the <code>na.action</code> setting of <a href="#">options</a> , and is <code>na.fail</code> if that is unset. The ‘factory-fresh’ default is <code>na.omit</code> . Another possible value is NULL, no action. Value <code>na.exclude</code> can be useful.
start	starting values for the parameters in the linear predictor.
etastart	starting values for the linear predictor.
mustart	starting values for the vector of means.
offset	this can be used to specify an <i>a priori</i> known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector or matrix of extents matching those of the response. One or more <code>offset</code> terms can be included in the formula instead or as well, and if more than one are specified their sum is used. See <a href="#">model.offset</a> .
control	a list of parameters for controlling the fitting process. For <code>eglm.wfit</code> this is passed to <code>glm.control</code> .
model	a logical value indicating whether <i>model frame</i> should be included as a component of the returned value.
method	the method to be used in fitting the model. The default method <code>"eglm.wfit"</code> uses iteratively reweighted least squares (IWLS); the alternative <code>"model.frame"</code> returns the model frame and does no fitting. User-supplied fitting functions can be supplied either as a function or a character string naming a function, with a function which takes the same arguments as <code>glm.fit</code> from the <b>stats</b> package. If specified as a character string it is looked up from within the <b>eflm</b> namespace.
x, y	logical values indicating whether the <i>model matrix</i> ( <code>x</code> ) and the <i>response vector</i> ( <code>y</code> ) used in the fitting process should be returned as components of the returned value.
singular.ok	logical; if FALSE a singular fit is an error.
contrasts	an optional list. See the <code>contrasts.arg</code> of <code>model.matrix.default</code> .
reduce	logical; if TRUE an alternate design matrix of $p \times p$ is used for the fitting instead of the traditional $n \times p$ design matrix.
...	For <code>eglm</code> : arguments to be used to form the default control argument if it is not supplied directly. For <code>weights</code> : further arguments passed to or from other methods.

## Details

Models for `eglm` are specified symbolically. A typical model has the form `response ~ terms` where `response` is the (numeric) response vector and `terms` is a series of terms which specifies a linear predictor for response. A terms specification of the form `first + second` indicates all the terms in `first` together with all the terms in `second` with duplicates removed. A specification of the form `first:second` indicates the set of terms obtained by taking the interactions of all terms in `first` with all terms in `second`. The specification `first*second` indicates the *cross* of `first` and `second`. This is the same as `first + second + first:second`, and exactly the same as "`glm`" from the `stats` package.

## Value

An object of class "`eglm`" that behaves the same way as the "`glm`" class, see the function "`glm`". This output also includes the logical "reduce" and, depending on it, the reduced design matrix "`xtx`" when the reduce argument is set to `TRUE`.

## Examples

```
eglm(mpg ~ wt, family = gaussian, data = mtcars)
```

---

eglm.wfit

*Fitting Generalized Linear Models*


---

## Description

Efficient Generalized Linear Model Weighted Fit ("`eglm.wfit`") is used to fit generalized linear models in an equivalent way to "`glm.fit`" but in a reduced time depending on the design matrix and the family (or link).

## Usage

```
eglm.wfit(
  x,
  y,
  weights = rep.int(1, nobs),
  start = NULL,
  etastart = NULL,
  mustart = NULL,
  offset = rep.int(0, nobs),
  family = gaussian(),
  control = list(),
  intercept = TRUE,
  singular.ok = TRUE,
  reduce = FALSE
)
```

**Arguments**

<code>x, y</code>	For <code>eglm.wfit</code> : <code>x</code> is a design matrix of dimension $n * p$ , and <code>y</code> is a vector of observations of length <code>n</code> , or a matrix with <code>n</code> rows.
<code>weights</code>	an optional vector of weights to be used in the fitting process. Should be <code>NULL</code> or a numeric vector. If non- <code>NULL</code> , weighted least squares is used with weights <code>weights</code> (that is, minimizing $\sum(w * e^2)$ ); otherwise ordinary least squares is used.
<code>start</code>	starting values for the parameters in the linear predictor.
<code>etastart</code>	starting values for the linear predictor.
<code>mustart</code>	starting values for the vector of means.
<code>offset</code>	this can be used to specify an <i>a priori</i> known component to be included in the linear predictor during fitting. This should be <code>NULL</code> or a numeric vector or matrix of extents matching those of the response. One or more <code>offset</code> terms can be included in the formula instead or as well, and if more than one are specified their sum is used. See <code>model.offset</code> .
<code>family</code>	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. See <code>family</code> for details of family functions.
<code>control</code>	a list of parameters for controlling the fitting process. For <code>eglm.wfit</code> this is passed to <code>glm.control</code> .
<code>intercept</code>	logical value indicating whether <i>intercept</i> should be included in the <i>null</i> model. Defaults to <code>TRUE</code> .
<code>singular.ok</code>	logical; if <code>FALSE</code> a singular fit is an error.
<code>reduce</code>	logical; if <code>TRUE</code> an alternate design matrix of $p * p$ is used for the fitting instead of the traditional $n * p$ design matrix.

**Details**

`eglm.wfit` is a workhorse function: it is not normally called directly but can be more efficient where the response vector, design matrix and family have already been calculated. Use `eglm` for most of the cases.

**Value**

A list that contains the same elements as the output from "`glm.fit`", with the addition of the vector "good" that indicates with logicals which observations were used in the fitting process.

**Examples**

```
x <- cbind(rep(1, nrow(mtcars)), mtcars$wt)
y <- mtcars$mpg
eglm.wfit(x, y)
```

**Description**

Efficient Linear Model ("elm") is used to fit linear models in an equivalent way to "lm" but in a reduced time depending on the design matrix.

**Usage**

```
elm(
  formula,
  data,
  subset,
  weights,
  na.action,
  method = "qr",
  model = TRUE,
  x = FALSE,
  y = FALSE,
  qr = TRUE,
  singular.ok = TRUE,
  contrasts = NULL,
  offset,
  reduce = TRUE,
  ...
)
```

**Arguments**

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by <a href="#">as.data.frame</a> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lm</code> is called.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If non-NULL, weighted least squares is used with weights <code>weights</code> (that is, minimizing $\sum(w \cdot e^2)$ ); otherwise ordinary least squares is used.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the <code>na.action</code> setting of <a href="#">options</a> , and is <code>na.fail</code> if that is unset. The 'factory-fresh' default is <code>na.omit</code> . Another possible value is NULL, no action. Value <code>na.exclude</code> can be useful.

method	the method to be used in fitting the model. The default method "eglm.wfit" uses iteratively reweighted least squares (IWLS); the alternative "model.frame" returns the model frame and does no fitting. User-supplied fitting functions can be supplied either as a function or a character string naming a function, with a function which takes the same arguments as <code>glm.fit</code> from the <b>stats</b> package. If specified as a character string it is looked up from within the <b>eflm</b> namespace.
model	a logical value indicating whether <i>model frame</i> should be included as a component of the returned value.
x	logical values indicating whether the <i>model matrix</i> (x) and the <i>response vector</i> (y) used in the fitting process should be returned as components of the returned value.
y	logical values indicating whether the <i>model matrix</i> (x) and the <i>response vector</i> (y) used in the fitting process should be returned as components of the returned value.
qr	logical. If TRUE the corresponding QR decomposition component of the fit is returned.
singular.ok	logical; if FALSE a singular fit is an error.
contrasts	an optional list. See the <code>contrasts.arg</code> of <code>model.matrix.default</code> .
offset	this can be used to specify an <i>a priori</i> known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector or matrix of extents matching those of the response. One or more <code>offset</code> terms can be included in the formula instead or as well, and if more than one are specified their sum is used. See <code>model.offset</code> .
reduce	logical; if TRUE an alternate design matrix of $p * p$ is used for the fitting instead of the traditional $n * p$ design matrix.
...	For <code>eglm</code> : arguments to be used to form the default control argument if it is not supplied directly. For <code>weights</code> : further arguments passed to or from other methods.

## Details

Models for `elm` are specified symbolically. A typical model has the form `response ~ terms` where `response` is the (numeric) response vector and `terms` is a series of terms which specifies a linear predictor for response. A terms specification of the form `first + second` indicates all the terms in `first` together with all the terms in `second` with duplicates removed. A specification of the form `first:second` indicates the set of terms obtained by taking the interactions of all terms in `first` with all terms in `second`. The specification `first*second` indicates the *cross* of `first` and `second`. This is the same as `first + second + first:second`, and exactly the same as "`lm`" from the `stats` package.

## Value

An object of class "elm" that behaves the same way as the "lm" class, see the function "`lm`". This output also includes the logical "reduce" and, depending on it, the reduced design matrix "xtx" when the `reduce` argument is set to TRUE.

## Examples

```
elm(mpg ~ wt, data = mtcars)
```

---

 elm.wfit

*Fitting Linear Models*


---

## Description

Efficient Linear Model Weighted Fit ("`elm.wfit`") is used to fit linear models in an equivalent way to "`glm.fit`" but in a reduced time depending on the design matrix and the family (or link).

## Usage

```
elm.wfit(
  x,
  y,
  weights = rep.int(1, n),
  offset = NULL,
  method = "qr",
  tol = 1e-07,
  singular.ok = TRUE,
  reduce = TRUE,
  ...
)
```

## Arguments

- |                      |  |
|----------------------|--|
| <code>x</code>       | logical values indicating whether the <i>model matrix</i> ( <code>x</code> ) and the <i>response vector</i> ( <code>y</code> ) used in the fitting process should be returned as components of the returned value.   |
| <code>y</code>       | logical values indicating whether the <i>model matrix</i> ( <code>x</code> ) and the <i>response vector</i> ( <code>y</code> ) used in the fitting process should be returned as components of the returned value.   |
| <code>weights</code> | an optional vector of weights to be used in the fitting process. Should be <code>NULL</code> or a numeric vector. If non- <code>NULL</code> , weighted least squares is used with weights <code>weights</code> (that is, minimizing $\sum(w * e^2)$ ); otherwise ordinary least squares is used.   |
| <code>offset</code>  | this can be used to specify an <i>a priori</i> known component to be included in the linear predictor during fitting. This should be <code>NULL</code> or a numeric vector or matrix of extents matching those of the response. One or more <code>offset</code> terms can be included in the formula instead or as well, and if more than one are specified their sum is used. See <code>model.offset</code> . |
| <code>method</code>  | the method to be used in fitting the model. The default method " <code>elm.wfit</code> " uses iteratively reweighted least squares (IWLS); the alternative " <code>model.frame</code> " returns the model frame and does no fitting. User-supplied fitting functions can   |



	be supplied either as a function or a character string naming a function, with a function which takes the same arguments as <code>glm.fit</code> from the <b>stats</b> package. If specified as a character string it is looked up from within the <b>eflm</b> namespace.
<code>tol</code>	tolerance for the <b>qr</b> decomposition. Default is <code>1e-7</code> .
<code>singular.ok</code>	logical; if <code>FALSE</code> a singular fit is an error.
<code>reduce</code>	logical; if <code>TRUE</code> an alternate design matrix of $p * p$ is used for the fitting instead of the traditional $n * p$ design matrix.
<code>...</code>	For <code>eglm</code> : arguments to be used to form the default control argument if it is not supplied directly. For <code>weights</code> : further arguments passed to or from other methods.

### Details

`elm.wfit` is a workhorse function: it is not normally called directly but can be more efficient where the response vector, design matrix and family have already been calculated. Use `elm` for most of the cases.

### Value

A list that contains the same elements as the output from "`lm.fit`".

### Examples

```
x <- cbind(rep(1, nrow(mtcars)), mtcars$wt)
y <- mtcars$mpg
elm.wfit(x, y)
```

# Index

`as.data.frame`, 3, 6

`eglm`, 2

`eglm.wfit`, 4

`elm`, 6

`elm.wfit`, 8

`family`, 3, 5

`formula`, 2, 6

`glm`, 2, 4

`glm.fit`, 4, 5, 8

`lm`, 6, 7

`lm.fit`, 9

`model.offset`, 3, 5, 7, 8

`na.exclude`, 3, 6

`na.fail`, 3, 6

`na.omit`, 3, 6

`offset`, 3, 5, 7, 8

`options`, 3, 6

`qr`, 9

`stats`, 4, 7