

# Package ‘PhaseGMM’

April 28, 2026

**Type** Package

**Title** Phase-Function Based Estimation and Inference for Linear  
Errors-in-Variables (EIV) Models

**Version** 0.1.1

**Description** Estimation and inference for coefficients of linear EIV models with symmetric measurement errors.

The measurement errors can be homoscedastic or heteroscedastic, for the latter, replication for at least some observations needs to be available. The estimation method and asymptotic inference are

based on a generalised method of moments framework, where the estimating equations are formed from (1) minimising the distance between the empirical phase function (normalised characteristic function) of the response and that of the linear combination of all the covariates at the estimates,

and (2) minimising a corrected least-square discrepancy function. Specifically, for a linear EIV model with  $p$  error-prone and  $q$  error-free covariates, if replicates are available, the GMM approach is based on a  $2(p+q)$  estimating equations if some replicates are available and based on  $p+2q$  estimating equations if no replicate is available.

The details of the method are described in Nghiem and Potgieter (2020) <[doi:10.1093/biomet/asaa025](https://doi.org/10.1093/biomet/asaa025)> and Nghiem and Potgieter (2025) <[doi:10.5705/ss.202022.0331](https://doi.org/10.5705/ss.202022.0331)>.

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**Depends** R (>= 3.5)

**Imports** nleqslv

**Suggests** extraDistr

**NeedsCompilation** no

**Author** Chang Liu [aut, cre],  
Linh Nghiem [aut]

**Maintainer** Chang Liu <[leo12345liu@gmail.com](mailto:leo12345liu@gmail.com)>

**Repository** CRAN

**Date/Publication** 2026-04-28 11:00:02 UTC

## Contents

coef.eiv_mlr . . . . .	2
confint.eiv_mlr . . . . .	2
dietary_white_women . . . . .	3
eiv_mlr . . . . .	4
predict.eiv_mlr . . . . .	8
print.eiv_mlr . . . . .	9
residuals.eiv_mlr . . . . .	9
summary.eiv_mlr . . . . .	10
vcov.eiv_mlr . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

coef.eiv_mlr	<i>Extract Coefficients from an eiv_mlr Object</i>
--------------	--

---

### Description

Extract Coefficients from an eiv\_mlr Object

### Usage

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

### Arguments

object	An object of class "eiv_mlr".
...	Not used.

### Value

A named numeric vector of estimated regression coefficients.

---

confint.eiv_mlr	<i>Confidence Intervals for eiv_mlr Coefficients</i>
-----------------	--

---

### Description

Computes Wald-type confidence intervals for regression coefficients.

### Usage

```
## S3 method for class 'eiv_mlr'
confint(object, parm = NULL, level = 0.95, ...)
```

**Arguments**

<code>object</code>	An object of class "eiv_mlr".
<code>parm</code>	Not used.
<code>level</code>	Confidence level; defaults to 0.95.
<code>...</code>	Not used.

**Value**

A matrix with columns lower and upper.

---

`dietary_white_women`    *dietary\_white\_women*

---

**Description**

A processed Dataset containing repeated 24-hour dietary recalls.

**Usage**

```
dietary_white_women
```

**Format**

A data frame with 2 rows per individual (one per recall day).

**unit** Integer unit identifier  
**bmi** Body Mass Index (kg/m<sup>2</sup>)  
**energy** Total energy intake (kcal)  
**protein** Protein intake (g)  
**fat** Fat intake (g)  
**age\_in\_month** Age in months  
**replicate** Replicate id of each observation

**Details**

The variables energy, protein, and fat are treated as error-prone covariates with two replicate measurements per individual, and age\_in\_month is treated as error-free covariate.

**Source**

Processed from NHANES dietary recall data.

---

eiv_mlr	<i>Linear Regression with Errors-in-Variables Using Replicated Measurements</i>
---------	---

---

### Description

Fits a linear regression model in the presence of measurement error in covariates using replicated measurements and a combination of phase-function estimation and generalized method of moments (GMM).

### Usage

```
eiv_mlr(
  formula,
  data,
  weight_method = c("uniform", "minimax", "quasi-likelihood"),
  B = 100,
  t_grid_length = 1000
)
```

### Arguments

formula	<p>A symbolic description of the model to be fitted. Error-prone covariates must be wrapped in <math>W()</math> and error-free covariates must be wrapped in <math>Z()</math>. The general form is:</p> $y \sim W(W1 + W2 + \dots) + Z(Z1 + Z2 + \dots)$ <p>An intercept is included automatically unless removed explicitly.</p>
data	<p>A data frame containing the response variable and all covariates appearing in formula. Each row corresponds to one replicate measurement of a statistical unit.</p> <p>The data frame must contain:</p> <ul style="list-style-type: none"> <li>• A column named <code>unit</code> identifying statistical units.</li> <li>• One or more rows per <code>unit</code> if replicated measurements exist.</li> <li>• One column for each error-prone covariate (appearing in <math>W()</math>).</li> <li>• One column for each error-free covariate (appearing in <math>Z()</math>).</li> </ul> <p>Replicated measurements are represented by multiple rows sharing the same <code>unit</code> identifier. Error-free covariates and the response should be constant within each unit.</p>
weight_method	<p>Character string specifying the observation weighting method used in estimation. One of:</p> <p>"uniform" Uniform weights across observations.</p> <p>"minimax" Minimax optimal weights.</p>

	"quasi-likelihood" Quasi-likelihood-based weights (default recommended).
B	Integer specifying the number of bootstrap replications used to estimate the GMM weighting matrix. Defaults to 100.
t_grid_length	Integer specifying the number of frequency grid points used in phase-function integration. Larger values improve numerical accuracy at the cost of computation time.

## Details

The function provides an lm-like interface while internally handling replicated error-prone covariates, measurement error correction, and robust variance estimation.

This function implements a measurement-error-corrected linear regression estimator for models with replicated error-prone covariates. When fewer than two units contain replicated measurements, the function automatically falls back to a quadratic (identity-weight) estimator. This ensures the model remains estimable even in the absence of replication.

The estimation procedure:

1. Aggregates replicated measurements into a structured array.
2. Uses phase-function estimating equations to correct for unknown measurement error distributions.
3. Combines moment conditions via GMM when sufficient replication information is available.
4. Automatically switches to a quadratic (identity-weight) estimator when fewer than two statistical units contain replicated measurements.

Variance estimation is performed using a sandwich estimator, with the GMM weighting matrix estimated via a cluster bootstrap over statistical units.

## Value

An object of class "eiv\_mlr" containing:

**coef** Estimated regression coefficients.

**vcov** Estimated variance-covariance matrix.

**se** Standard errors of the estimates.

**zvalue** Z-statistics for hypothesis testing.

**pvalue** Two-sided p-values.

**fitted** Fitted values at the unit level.

**method** Estimation method used (GMM or quadratic fallback).

**n** Number of statistical units.

Standard methods such as `summary()`, `coef()`, `vcov()`, `confint()`, `predict()`, and `residuals()` are available for objects of this class.

**Examples**

```

## -----
## Small reproducible example (for speed reasons, we chose a too small number of bootstrap samples)
## -----

set.seed(1)

n <- 30
J <- 2

unit <- rep(1:n, each = J)

W_true <- rnorm(n)
W_obs <- rep(W_true, each = J) + rnorm(n * J, sd = 0.5)

Z1 <- rep(rnorm(n), each = J)
y <- rep(1 + 2 * W_true - 0.5 * Z1[seq(1, n * J, by = J)], each = J) +
  rnorm(n * J)

sim_data <- data.frame(
  unit = unit,
  y = y,
  W1 = W_obs,
  Z1 = Z1
)

# For speed reasons, we use a very small number of bootstrap samples
fit <- eiv_mlr(
  y ~ W(W1) + Z(Z1),
  data = sim_data,
  B = 10,
  t_grid_length = 20
)

coef(fit)
summary(fit)

## -----
## Additional examples (not run during checks)
## -----

## -----
## Example using included dataset
## -----

fit <- eiv_mlr(
  bmi ~ W(energy + protein + fat) + Z(age_in_month),
  data = dietary_white_women,
  weight_method = "minimax",
  B = 100,

```

```

    t_grid_length = 200
  )

summary(fit)
confint(fit)

## -----
## Simulated example with replication
## -----

set.seed(1)

n <- 200
J <- 2

unit <- rep(1:n, each = J)

W_true <- rnorm(n)
W_obs <- rep(W_true, each = J) + rnorm(n * J, sd = 0.5)

Z1 <- rep(rnorm(n), each = J)
y <- rep(1 + 2 * W_true - 0.5 * Z1[seq(1, n * J, by = J)], each = J) +
  rnorm(n * J)

sim_data <- data.frame(
  unit = unit,
  y = y,
  W1 = W_obs,
  Z1 = Z1
)

fit_rep <- eiv_mlr(
  y ~ W(W1) + Z(Z1),
  data = sim_data,
  B = 20
)

summary(fit_rep)

## -----
## Simulated example without replication
## -----

sim_norep <- sim_data[!duplicated(sim_data$unit), ]

fit_norep <- eiv_mlr(
  y ~ W(W1) + Z(Z1),
  data = sim_norep,
  B = 20
)

```

```
summary(fit_norep)
```

---

predict.eiv\_mlr      *Predictions from an Errors-in-Variables Linear Model*

---

### Description

Generates fitted values or confidence intervals for new observations using plug-in estimates. Measurement error is not corrected for new data.

### Usage

```
## S3 method for class 'eiv_mlr'
predict(
  object,
  newdata = NULL,
  interval = c("none", "confidence"),
  level = 0.95,
  ...
)
```

### Arguments

object	An object of class "eiv_mlr".
newdata	Optional data frame containing covariates. If omitted, predictions are returned for the training data.
interval	Type of interval to compute: "none" or "confidence".
level	Confidence level for intervals.
...	Not used.

### Value

A numeric vector of predictions, or a matrix with columns fit, lwr, and upr if intervals are requested.

---

print.eiv_mlr	<i>Print Method for eiv_mlr Objects</i>
---------------	---

---

**Description**

Displays a concise summary of an errors-in-variables linear model fit.

**Usage**

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

**Arguments**

x	An object of class "eiv_mlr".
...	Not used.

**Value**

Invisibly returns the input object x of class "eiv\_mlr". This function is called for its side effect of printing a summary of the model.

---

residuals.eiv_mlr	<i>Residuals from an Errors-in-Variables Linear Model</i>
-------------------	---

---

**Description**

Computes plug-in residuals based on observed covariates and estimated coefficients. Measurement error is not corrected in residuals.

**Usage**

```
## S3 method for class 'eiv_mlr'
residuals(object, ...)
```

**Arguments**

object	An object of class "eiv_mlr".
...	Not used.

**Value**

A numeric vector of residuals.

---

summary.eiv_mlr	<i>Summary of an Errors-in-Variables Linear Model</i>
-----------------	---

---

**Description**

Produces a coefficient table including standard errors, z-statistics, p-values, and significance stars.

**Usage**

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

**Arguments**

object	An object of class "eiv_mlr".
...	Not used.

**Value**

An object of class "summary" containing a coefficient table and model information.

---

vcov.eiv_mlr	<i>Variance-Covariance Matrix for eiv_mlr Objects</i>
--------------	---

---

**Description**

Variance-Covariance Matrix for eiv\_mlr Objects

**Usage**

```
## S3 method for class 'eiv_mlr'  
vcov(object, ...)
```

**Arguments**

object	An object of class "eiv_mlr".
...	Not used.

**Value**

A numeric variance-covariance matrix of the parameter estimates.

# Index

## \* datasets

    dietary\_white\_women, 3

coef.eiv\_mlr, 2

confint.eiv\_mlr, 2

dietary\_white\_women, 3

eiv\_mlr, 4

predict.eiv\_mlr, 8

print.eiv\_mlr, 9

residuals.eiv\_mlr, 9

summary.eiv\_mlr, 10

vcov.eiv\_mlr, 10