

Package: heterogeneouspeereffects (via r-universe)

October 8, 2026

Title Heterogeneous Peer Effect

Version 0.1.0

Description Heterogeneous Peer Effect Package provides two-step Generalized Method of Moments (GMM) estimators for heterogeneous peer effects in group-level treatment models developed by Pasquier, Rossi and Wang (2026) <https://crest.science/wp-content/uploads/2026/09/2026-11.pdf>. The package separates the direct effect of treatment from within-group and between-group spillover effects, using a cross-fitted, semiparametric approach that leaves the propensity score unspecified and estimates it nonparametrically. Two identification settings are implemented: one in which eligibility for treatment coincides with group identity, and one in which identity is orthogonal to eligibility, allowing peer effects to differ across subgroups (e.g. by gender). Point estimates, standard errors, and test statistics are returned for the direct effect and for each within- and between-group peer effect.

License MIT + file LICENSE

Encoding UTF-8

Imports dplyr, ggplot2, purrr, flextable, caret, MASS, mvtnorm

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

Config/testthat/edition 3

Author Felix Pasquier [aut], Laurine Meier [aut, cre], Pauline Rossi [aut]

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Maintainer Laurine Meier <laurine.meier@ensae.fr>

URL <https://github.com/LaurineMir/heterogeneouspeereffects>

BugReports <https://github.com/LaurineMir/heterogeneouspeereffects/issues>

Config/pak/sysreqs
libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev

Repository <https://laurinemir.r-universe.dev>

Date/Publication 2026-09-28 10:08:37 UTC

RemoteUrl <https://github.com/laurinemir/heterogeneouspeereffects>

RemoteRef HEAD

RemoteSha 189f88c6924506639ac496a02958d8734f350740

Contents

heter_endo_gmm	2
ortho_heter_endo_gmm	3
Index	6

heter_endo_gmm	<i>Two-Step GMM Estimation of Heterogeneous Peer Effects When Identity Equals Eligibility</i>
----------------	---

Description

Two-Step GMM Estimation of Heterogeneous Peer Effects When Identity Equals Eligibility

Usage

heter_endo_gmm(YE, YN, D, s, n_param = 5, tol = 1e-06)

Arguments

- YE : average outcome for eligible individuals within a group (vector)
- YN average outcome for non-eligible individuals within a group (vector)
- D group binary treatment indicator (vector)
- s share of eligible individuals in the group (vector)
- n_param : number of parameters to estimate
- tol : convergency criteria of optimization

Value

A list with three elements:

- estimates data frame with the estimated values of the direct effect (delta), intra-group effect (theta_within), inter-group effect (theta_between) and their standard errors.
- table a flextable object formatting estimates for display.
- plot a ggplot object showing the treatment effects on eligible and non-eligible individuals across values of s.

Examples

```
# Parameters
delta = -3
thetaW = 0.7
thetaB = 0.3
beta_NE = -2
beta_E = -1
G <- 1000
# Generate the mean and sd of the random vector alpha
mu_alphaX <- c(8, 4, -3, 2)
Sigma_alphaX <- matrix(c(4, 1, 0, 0, 1, 4, 0, 0, 0, 0, 4, -4.2, 0, 0, -4.2, 9), nrow = 4)
# Create a logistic function to correlate D with s
logistic <- function(x) {
  1 / (1 + exp(-x))
}
# Generate the alpha vector
alphaX <- rmvnorm::rmvnorm(n = G, mean = mu_alphaX, sigma = Sigma_alphaX)
# Scale s to control correlation strength (optional)
s <- runif(n = G)
scaled_s <- (s - mean(s)) / sd(s) # Standardize s for better control
# Compute probabilities for D based on scaled s
prob_D <- logistic(scaled_s) # Adjust scaling for desired correlation
# Generate D based on the probabilities
D <- rbinom(n = G, size = 1, prob = prob_D)
# Generate YE and YNE
YN <- ((1 - thetaW * s) * (alphaX[,1] + beta_NE * alphaX[,3]) +
  thetaB * s * (alphaX[,2] + beta_E * alphaX[,4]) +
  delta * thetaB * s)
YE <- ((1 - thetaW * s) * (1 - s) * thetaB * (alphaX[,1] + beta_NE * alphaX[,3]) +
  (1 + thetaW * (s * (1 - s) * thetaW - 1)) * (alphaX[,2] + beta_E * alphaX[,4]) +
  delta * (1 + thetaW * (s * (1 - s) * thetaW - 1)) * D) /
  ((1 - thetaW * s) * (1 - thetaW + s * (1 - s) * (thetaW^2 - thetaB^2)))
result_simple_5param <- heter_endo_gmm(YE, YN, D, s)
# Access the estimates data frame:
result_simple_5param$estimates
# Display the formatted table:
# print(result_simple_5param$table)
# Display the treatment effect plot:
# print(result_simple_5param$plot)
```

ortho_heter_endo_gmm	<i>Two-Step GMM Estimation of Heterogeneous Peer Effects with Orthogonal Identity-Based Eligibility</i>
----------------------	---

Description

Two-Step GMM Estimation of Heterogeneous Peer Effects with Orthogonal Identity-Based Eligibility

Usage

```
ortho_heter_endo_gmm(YM, YF, D, sM, sEM, sEF, tol = 1e-06)
```

Arguments

YM	: average outcome for "male" individuals within a group (vector)
YF	average outcome for "female" individuals within a group (vector)
D	group binary treatment indicator (vector)
sM	share of "male" individuals in the group (vector)
sEM	share of eligible "male" individuals in the group (vector)
sEF	share of eligible "female" individuals in the group (vector)
tol	: convergency criteria of optimization

Value

A list with two elements:

estimates	data frame with the estimated values of the direct effect (delta), within effect among male (theta_within_M), within effect among female (theta_within_F), between effect from male to female (theta_between_F_M), between effect from female to male (theta_between_M_F) and their standard errors.
table	a flextable object formatting estimates for display.

Examples

```
delta = -3
thetaWM = 0.6
thetaWW = 0.2
thetaBWM = -0.5
thetaBMW = 0.5
beta_W = -2
beta_M = -1
G <- 500
# Generate the mean and sd of the random vector alpha
mu_alphaX <- c(8, 4, -3, 2)
Sigma_alphaX <- matrix(c(4, 1, 0, 0, 1, 4, 0, 0, 0, 0, 4, -4.2, 0, 0, -4.2, 9), nrow = 4)
# Create a logistic function to correlate D with s
logistic <- function(x) {
  1 / (1 + exp(-x))
}
# Generate the alpha vector
alphaX <- rmvnorm::rmvnorm(n = G, mean = mu_alphaX, sigma = Sigma_alphaX)
# Generate the vectors (s, T)
sM <- runif(n = G)
sWE <- runif(n = G)
sME <- runif(n = G)
D <- rbinom(n = G, size = 1, prob = 0.5)
# Generate YE and YNE
YW <- ((1 - thetaWM * sM) * (alphaX[,2] + beta_M * alphaX[,4]) +
```

```

thetaBWM * sM * (alphaX[,1] + beta_W * alphaX[,3]) +
D * delta * (sWE * (1 - thetaWM * sM) + sME * thetaBWM * sM)) /
(1 - thetaWW * (1 - sM) - thetaWM * sM +
  sM * (1 - sM) * (thetaWW * thetaWM - thetaBWM * thetaBMW))
YM <- ((1 - thetaWW * (1 - sM)) * (alphaX[,1] + beta_M * alphaX[,3]) +
thetaBMW * (1 - sM) * (alphaX[,2] + beta_W * alphaX[,4]) +
D * delta * (sWE * thetaBMW * (1 - sM) +
  sME * (1 - thetaWW * (1 - sM)))) /
(1 - thetaWW * (1 - sM) - thetaWM * sM +
  sM * (1 - sM) * (thetaWW * thetaWM - thetaBWM * thetaBMW))
# Resultats analytique
result_ortho <- ortho_heter_endo_gmm(YM, YW, D, sM, sME, sWE)
result_ortho$estimates
# print(result_ortho$table)

```

Index

heter_endo_gmm, [2](#)

ortho_heter_endo_gmm, [3](#)