Package: panglm (via r-universe)

October 22, 2026

Type Package

Title Generalized Linear Models for Panel Data

Version 1.1.4

Description Provides generalized linear models for panel data, including pooled, fixed-effects, and random-effects estimators for continuous, binary, and count outcomes. Offers a unified interface for fitting and analysing panel regression models, with efficient computation for large datasets. Estimators and tests follow standard panel-data references, including Hausman (1978) <doi:10.2307/1913827>, Chamberlain (1980) <doi:10.2307/2297110>, Allison and Waterman (2002) <doi:10.1111/1467-9531.00117>, and Croissant and Millo (2008) <doi:10.18637/jss.v027.i02>. Core numerical routines are implemented using 'Rcpp', 'RcppArmadillo', and 'RcppParallel'.

License MIT + file LICENSE

URL https://CRAN.R-project.org/package=panglm, https://github.com/ielbadisy/panglm

BugReports https://github.com/ielbadisy/panglm/issues

Encoding UTF-8

Depends R (>= 3.6.0)

Imports Rcpp, data.table, graphics, stats

LinkingTo Rcpp, RcppArmadillo, RcppParallel

Suggests testthat (>= 3.0.0), plm, pglm, fixest, sandwich, survival, MASS, AER, glmmTMB, lme4, pscl, generics, broom, knitr, rmarkdown

Config/testthat/edition 3

SystemRequirements GNU make

RoxygenNote 7.3.3

VignetteBuilder knitr

LazyData true

NeedsCompilation yes

Author Imad El Badisy [aut, cre]

Maintainer Imad El Badisy <elbadisyimad@gmail.com>

Config/pak/sysreqs make

Repository https://cran.r-universe.dev

Date/Publication 2026-09-22 07:10:02 UTC

RemoteUrl https://github.com/cran/panglm

RemoteRef HEAD

RemoteSha a6571c2365e3a1ea7d14e4a7aa8b9f248b439a52

Contents

confint.panglm 2
copd 3
fitted.panglm 4
glance.panglm 5
logLik.panglm 5
panglm 6
panglm-families 8
panglm_dispersiontest 9
panglm_hausman 9
panglm_hurdle 10
plot.panglm 12
plot.panglm_hurdle 13
predict.panglm 13
summary.panglm 14
tidy.panglm 15
vcov.panglm 16

Index **18**

`confint.panglm` *Confidence intervals for panglm coefficients*

Description

Wald-type intervals ‘estimate +/- z * se‘, using whichever ‘vcov()‘ the object currently carries (classical by default; HC1/cluster if the fit or a prior [vcov.panglm()] call set it).

Usage

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

Arguments

`object`	a ‘"panglm"‘ fit
`parm`	which parameters (names or indices); defaults to all
`level`	confidence level
`...`	unused

Value

a numeric matrix with one row per requested parameter and two columns giving the lower and upper confidence limits, named after the quantiles implied by ‘level‘ (e.g. ‘"2.5 are the parameter names.

Examples

```
data(copd)
fit <- panglm(fev1 ~ crp, data = copd, index = c("id", "visit"),
              model = "within", family = "gaussian")
confint(fit)
```

---

copd *Synthetic COPD follow-up panel*

Description

A simulated longitudinal (panel) dataset of chronic obstructive pulmonary disease (COPD) patients followed over several clinic visits, used throughout this package's examples and vignette. It is entirely synthetic (not real patient data), generated to have a realistic panel structure – time-invariant baseline covariates plus an individual-level frailty/random intercept – so it exercises pooled, fixed-effects, and random-effects fits across all four families (‘gaussian‘, ‘poisson‘, ‘binomial‘, ‘negbin‘) without depending on any Suggested package. See ‘data-raw/copd.R‘ for the generating code.

Usage

```
copd
```

Format

A data frame with 300 rows (60 patients x 5 visits) and 9 columns:

id patient identifier (1-60)

visit visit number (1-5), the panel time index

treatment treatment arm, time-invariant (0 = control, 1 = active)

age patient age in years at baseline, time-invariant

smoker current smoker, time-invariant (0/1)

bmi body mass index at baseline, time-invariant

crp C-reactive protein (mg/L) measured at each visit; the one time-varying covariate, needed for fixed-effects ("within") fits to have something left to estimate after demeaning

fev1 forced expiratory volume in 1 second (litres); a continuous lung-function outcome, one value per visit

exacerbations number of COPD exacerbations since the previous visit; a count outcome, one value per visit

hospitalized hospital admission in the interval since the previous visit; a binary outcome, one value per visit

Source

Simulated; see ‘data-raw/copd.R‘.

`fitted.panglm` *Extract fitted values from a panglm model*

Description

Values are returned in the row order of the data supplied to [panglm()], even though estimation internally sorts observations by panel index. Fixed-effects binomial values are conditional inclusion probabilities, conditional on each individual’s observed number of successes.

Usage

```
## S3 method for class 'panglm'
fitted(object, ...)
```

Arguments

`object`	a fitted ‘"panglm"‘ object
`...`	unused

Value

a numeric vector of fitted values, one per row of the original data supplied to [panglm()] (in that row order), or ‘NA‘ at rows dropped for missingness, following the model’s ‘na.action‘.

glance.panglm *One-row model summary of a panglm fit*

Description

Implements the ‘generics::glance()‘ generic.

Usage

```
## S3 method for class 'panglm'
glance(x, ...)
```

Arguments

x	a ‘"panglm"‘ fit
...	unused

Value

a one-row ‘data.frame‘ including log-likelihood, AIC, BIC, the likelihood parameter count when a likelihood is available, and (Gaussian-family fits only) R-squared and adjusted R-squared

Examples

```
data(copd)
fit <- panglm(fev1 ~ treatment + age + smoker + crp, data = copd,
              index = c("id", "visit"), model = "pooling", family = "gaussian")
if (requireNamespace("generics", quietly = TRUE)) {
  generics::glance(fit)
}
```

logLik.panglm *Extract the model log-likelihood*

Description

The ‘df‘ attribute counts every parameter associated with the reported likelihood, including fitted shape, dispersion, variance, and explicit fixed-effect parameters. For conditional Poisson and conditional logistic likelihoods, only parameters remaining in the conditional likelihood are counted. Models estimated without a likelihood return ‘NA‘.

Usage

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

Arguments

object	a fitted ‘"panglm"‘ object
...	unused

Value

an object of class ‘"logLik"‘

Examples

```
data(copd)
fit <- panglm(exacerbations ~ crp, data = copd,
              index = c("id", "visit"), model = "pooling",
              family = "poisson")
logLik(fit)
AIC(fit)
BIC(fit)
```

---

panglm *Fit a generalized linear model for panel data*

Description

Fast pooled, fixed-effects ("within") and random-effects panel GLMs, backed by an Rcpp/RcppArmadillo/RcppParallel numerical core. See the package vignette for the estimators currently implemented per family.

Usage

```
panglm(
  formula,
  data,
  index,
  model = c("pooling", "within", "random"),
  family = "gaussian",
  weights = NULL,
  offset = NULL,
  na.action = stats::na.omit,
  effect = c("individual", "twoways"),
  vcov = c("classical", "HC1", "cluster"),
  R = 21,
  maxit = 100,
  tol = 1e-10
)
```

Arguments

formula	a model formula, e.g. ‘y ~ x1 + x2‘
data	a data.frame containing the variables in ‘formula‘ and ‘index‘
index	length-1 or length-2 character vector identifying the individual (and optionally time) columns, e.g. ‘c("id", "time")‘
model	one of ‘"pooling"‘, ‘"within"‘, ‘"random"‘
family	one of ‘"gaussian"‘, ‘"poisson"‘, ‘"binomial"‘, ‘"negbin"‘, or a family spec from [gaussian_family()] / [poisson_family()] / [binomial_family()] / [negbin_family()]
weights	optional observation weights. Only ‘NULL‘ or a vector of ones is currently accepted; nontrivial weights are rejected explicitly because weighted conditional panel likelihoods require different estimators.
offset	optional model offset. Nonzero offsets, including offsets specified in ‘formula‘, are rejected explicitly rather than ignored.
na.action	missing-data action passed to [stats::model.frame()]
effect	one of ‘"individual"‘ or ‘"twoways"‘ (individual + time fixed effects). ‘"twoways"‘ is currently only implemented for ‘model = "within"‘, ‘family = "gaussian"‘ (exact alternating-projections demeaning), ‘"poisson"‘, or ‘"negbin"‘ (both via outer-IRLS / inner-weighted-FWL, the algorithm behind ‘fixest::feglm()‘); not yet for ‘"binomial"‘ (no general closed-form two-way conditional logit exists).
vcov	one of ‘"classical"‘, ‘"HC1"‘ (heteroskedasticity-robust) or ‘"cluster"‘ (cluster-robust, clustered by the panel individual by default). Only applies to ‘model = "pooling"‘/‘"within"‘; random-effects models always report model-based (information-matrix) standard errors. See [vcov.panglm()] to recompute a different type after fitting.
R	number of Gauss-Hermite quadrature nodes (random-effects binomial models only)
maxit	maximum IRLS/Newton iterations
tol	convergence tolerance

Value

an object of class ‘"panglm"‘

Examples

```
data(copd)

# pooled OLS on lung function (ignores the panel structure)
fit_pool <- panglm(fev1 ~ treatment + age + smoker + crp, data = copd,
                   index = c("id", "visit"), model = "pooling", family = "gaussian")
summary(fit_pool)

# fixed-effects ("within") estimator -- only the time-varying regressor
# (crp) survives demeaning; time-invariant covariates like treatment
# or age are not identified by a fixed-effects estimator
fit_fe <- panglm(fev1 ~ crp, data = copd,
```

```
                  index = c("id", "visit"), model = "within", family = "gaussian")
coef(fit_fe)

# random-effects (Swamy-Arora) estimator
fit_re <- panglm(fev1 ~ treatment + age + smoker + crp, data = copd,
                  index = c("id", "visit"), model = "random", family = "gaussian")
coef(fit_re)

# fixed-effects Poisson on exacerbation counts
fit_pois <- panglm(exacerbations ~ crp, data = copd,
                   index = c("id", "visit"), model = "within", family = "poisson")
coef(fit_pois)
```

`panglm-families` *Family specifications for panglm*

Description

Lightweight family objects used by [panglm()], analogous in spirit to [stats::family] objects but keyed to the integer codes the C++ backend expects (see 'src/families.h').

Usage

```
gaussian_family(link = "identity")

poisson_family(link = "log")

binomial_family(link = c("logit", "probit"))

negbin_family(link = "log")
```

Arguments

`link`	canonical link name

Value

a list with 'family', 'link', 'family_id', 'link_id'

Examples

```
gaussian_family()
poisson_family()
binomial_family("probit")
negbin_family()

data(copd)
fit <- panglm(hospitalized ~ crp, data = copd, index = c("id", "visit"),
              model = "within", family = binomial_family("logit"))
coef(fit)
```

panglm_dispersiontest *Pearson chi-squared/df overdispersion test*

Description

A quick diagnostic for count models: the Pearson goodness-of-fit statistic ‘sum((y - mu)^2 / mu)‘ divided by its residual degrees of freedom should be close to 1 under a correctly-specified Poisson model; values well above 1 indicate overdispersion (evidence for a negative binomial specification instead). The diagnostic is available for pooled and fixed-effects Poisson or negative binomial models. It is not applied to random-effects models because their marginal variance includes the fitted mixing distribution and is not equal to the observation-level Poisson variance used by this statistic.

Usage

```
panglm_dispersiontest(object)
```

Arguments

object	a ‘"panglm"‘ fit with ‘family‘ ‘"poisson"‘ or ‘"negbin"‘

Value

a list of class ‘"panglm_dispersiontest"‘ with the Pearson statistic, its degrees of freedom, the ratio (statistic/df), a one-sided p-value against ‘chisq(df)‘, and ‘n_excluded‘: the number of observations dropped from the statistic because their fitted value was exactly zero or unavailable (e.g. rows in a structurally-zero panel individual screened out of a fixed-effects negbin fit)

Examples

```
data(copd)
fit <- panglm(exacerbations ~ treatment + age + smoker + crp, data = copd,
              index = c("id", "visit"), model = "pooling", family = "poisson")
panglm_dispersiontest(fit)
```

panglm_hausman *Hausman specification test (fixed vs. random effects)*

Description

Tests the null that the random-effects estimator is consistent (i.e. individual effects are uncorrelated with the regressors) against the alternative that only the fixed-effects estimator is consistent, using the classic Hausman (1978) statistic ‘t(b_fe - b_re) computed over the coefficients common to both models (the random-effects model’s intercept is dropped, since the within/FE model has none).

Usage

```
panglm_hausman(fe, re)
```

Arguments

fe	a ‘"panglm"‘ fit with ‘model = "within"‘
re	a ‘"panglm"‘ fit with ‘model = "random"‘, same formula/data/family

Value

a list with class ‘"panglm_hausman"‘: ‘statistic‘, ‘parameter‘ (df), ‘p.value‘, and the compared coefficient vectors

Examples

```
data(copd)
fit_fe <- panglm(fev1 ~ crp, data = copd, index = c("id", "visit"),
                 model = "within", family = "gaussian")
fit_re <- panglm(fev1 ~ crp, data = copd, index = c("id", "visit"),
                 model = "random", family = "gaussian")
panglm_hausman(fit_fe, fit_re)
```

---

panglm_hurdle *Fixed-effects hurdle model for panel count data*

Description

A hurdle decomposition for panels where a large fraction of groups report *structural* zeros (never observed to occur), rather than zeros arising from ordinary Poisson/NB sampling variation at a low mean – a pattern that shows up as extreme, family-invariant overdispersion under either ‘family = "poisson"‘ or ‘family = "negbin"‘ in [panglm()].

Usage

```
panglm_hurdle(
  formula,
  data,
  index,
  count_family = c("poisson", "negbin"),
  maxit = 100,
  tol = 1e-10
)
```

Arguments

formula	a model formula, e.g. ‘y ~ x1 + x2‘
data	a data.frame containing the variables in ‘formula‘ and ‘index‘
index	length-2 character vector identifying the individual and time columns, e.g. ‘c("id", "time")‘
count_family	count distribution for positive observations, either ‘"poisson"‘ or ‘"negbin"‘ (NB2 parameterization)
maxit	maximum IRLS/Newton iterations
tol	convergence tolerance

Details

The model is fit in two independent parts, both using fixed effects, and reported together rather than folded into ‘panglm()‘’s single-family dispatch, since they operate on different sample sizes (the count part drops the zeros):

1. **Zero vs. positive.** ‘1(y > 0)‘ fit via the exact conditional logistic regression already used for ‘panglm(..., family = "binomial")‘ (Chamberlain 1980) – no new estimator, just a new outcome vector.
2. **Count given ‘y > 0‘.** A zero-truncated Poisson or NB2 fixed-effects model (Allison-Waterman-style dummy-variable intercepts, the same unconditional approach used for FE-NB2) on the subsample where ‘y > 0‘. The log-density is the ordinary count log-density minus ‘log(1 - P(Y = 0))‘; see ‘pscl::hurdle()‘ for the reference form of the truncated-count likelihood.

Groups with no remaining within-group variation after truncation (a single surviving observation) contribute no information about the shared count-part slope, the same profiling fact that applies to the other fixed-effects estimators in this package; they are not dropped explicitly since a perfectly-fit singleton doesn’t diverge, it simply carries zero weight.

Value

an object of class ‘"panglm_hurdle"‘: a list with ‘zero‘ (the ‘panglm‘ binomial fit for the zero-vs-positive part) and ‘count‘ (the zero-truncated fixed-effects fit for the count part)

Examples

```
data(copd)
fit <- panglm_hurdle(exacerbations ~ crp, data = copd, index = c("id", "visit"))
fit
```

`plot.panglm` *Plot a fitted panglm model*

Description

Provides base-graphics coefficient and diagnostic plots. The coefficient view shows Wald confidence intervals. Diagnostic views are available when the fitted object stores fitted values.

Usage

```
## S3 method for class 'panglm'
plot(
  x,
  which = c("coefficients", "residuals", "fitted"),
  level = 0.95,
  main = NULL,
  xlab = NULL,
  ...
)
```

Arguments

`x`	a fitted ‘"panglm"‘ object
`which`	one of ‘"coefficients"‘, ‘"residuals"‘, or ‘"fitted"‘
`level`	confidence level for coefficient intervals
`main`	optional plot title
`xlab`	optional horizontal-axis label
`...`	additional arguments passed to [graphics::plot.default()]

Value

‘x‘, invisibly

Examples

```
data(copd)
fit <- panglm(fev1 ~ crp, data = copd, index = c("id", "visit"),
              model = "within", family = "gaussian")
plot(fit)
```

plot.panglm_hurdle *Plot a fitted panglm hurdle model*

Description

Displays coefficient estimates and Wald confidence intervals for the binary hurdle component, the positive-count component, or both.

Usage

```
## S3 method for class 'panglm_hurdle'
plot(x, which = c("both", "zero", "count"), level = 0.95, ...)
```

Arguments

x	a fitted ‘"panglm_hurdle"‘ object
which	one of ‘"both"‘, ‘"zero"‘, or ‘"count"‘
level	confidence level for coefficient intervals
...	additional arguments passed to [graphics::plot.default()]

Value

‘x‘, invisibly

Examples

```
data(copd)
fit <- panglm_hurdle(exacerbations ~ crp, data = copd,
                     index = c("id", "visit"))
plot(fit)
```

predict.panglm *Predict from a fitted panglm model*

Description

Pooled predictions use the ordinary GLM mean. Random-effects response predictions are population averages over the fitted latent-effect distribution. Fixed-effects predictions include the estimated individual and, where applicable, time effects. New fixed-effect levels produce an error unless ‘allow.new.levels = TRUE‘, in which case their contribution is set to zero.

Usage

```
## S3 method for class 'panglm'
predict(
  object,
  newdata = NULL,
  type = c("link", "response"),
  allow.new.levels = FALSE,
  ...
)
```

Arguments

object	a fitted ‘"panglm"‘ object
newdata	optional data frame; when omitted, return in-sample values
type	either ‘"link"‘ or ‘"response"‘
allow.new.levels	if ‘TRUE‘, assign zero to unseen fixed-effect levels; otherwise unseen levels are an error
...	unused

Details

Exact conditional binomial fits provide in-sample conditional inclusion probabilities. They cannot provide predictions for arbitrary ‘newdata‘ because the individual intercept is conditioned out and the probability depends on the complete response set and its success total.

Value

a numeric vector of predictions

Examples

```
data(copd)
fit <- panglm(fev1 ~ crp, data = copd, index = c("id", "visit"),
              model = "random", family = "gaussian")
predict(fit, newdata = copd[1:4, ], type = "response")
```

---

summary.panglm *Summarize a panglm fit*

Description

Produces a classic coefficient table (Estimate, Std. Error, z value, Pr(>|z|)) in the style of [summary.glm()].

Usage

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

Arguments

object	a ‘"panglm"‘ object
...	unused

Value

an object of class ‘"summary.panglm"‘, a list with components ‘call‘ (the matched call), ‘model‘, ‘effect‘, and ‘family‘ (as stored on the fit), ‘coefficients‘ (a numeric matrix with one row per parameter and columns ‘Estimate‘, ‘Std. Error‘, ‘z value‘, ‘Pr(>|z|)‘), ‘nobs‘, ‘n_groups‘, ‘df.residual‘, ‘loglik‘, ‘dispersion‘ (‘NULL‘ when not applicable), ‘iterations‘, and ‘vcov_type‘ (which covariance estimator produced the reported standard errors). It has an associated ‘print.summary.panglm()‘ method and is normally printed rather than used programmatically.

Examples

```
data(copd)
fit <- panglm(fev1 ~ treatment + age + smoker + crp, data = copd,
              index = c("id", "visit"), model = "pooling", family = "gaussian")
summary(fit)
```

---

tidy.panglm *Tidy a panglm fit into a data frame of coefficients*

Description

Implements the ‘generics::tidy()‘ generic so ‘panglm‘ results work with ‘broom‘/‘modelsummary‘ table-building workflows.

Usage

```
## S3 method for class 'panglm'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x	a ‘"panglm"‘ fit
conf.int	if ‘TRUE‘, add ‘conf.low‘/‘conf.high‘ columns
conf.level	confidence level for the interval
...	unused

Value

a ‘data.frame‘ with one row per coefficient

Examples

```
data(copd)
fit <- panglm(fev1 ~ treatment + age + smoker + crp, data = copd,
              index = c("id", "visit"), model = "pooling", family = "gaussian")
if (requireNamespace("generics", quietly = TRUE)) {
  generics::tidy(fit, conf.int = TRUE)
}
```

`vcov.panglm` *Recompute the coefficient covariance matrix for a panglm fit*

Description

‘type = "classical"‘ returns the model-based covariance the object was fit with. ‘type = "HC1"‘ (heteroskedasticity-robust / White) and ‘type = "cluster"‘ (cluster-robust sandwich, clustered by the panel individual unless ‘cluster‘ is given) are only available for ‘model = "pooling"‘ and ‘model = "within"‘; random-effects models already integrate out the individual-level correlation via the likelihood, so they report model-based (information-matrix) standard errors, matching the convention used by ‘lme4‘/‘glmmTMB‘. For exact conditional binomial models, the independent likelihood units are panel strata rather than observations. Their HC1 covariance therefore uses stratum-level scores. A custom cluster may combine complete strata, but it may not split observations from one conditioned stratum.

Usage

```
## S3 method for class 'panglm'
vcov(object, type = c("classical", "HC1", "cluster"), cluster = NULL, ...)
```

Arguments

`object`	a ‘"panglm"‘ fit
`type`	one of ‘"classical"‘, ‘"HC1"‘, ‘"cluster"‘
`cluster`	optional vector giving the clustering variable (one value per original observation, i.e. ‘nrow(data)‘ long); defaults to the panel individual identifier used to fit the model
`...`	unused

Value

a numeric covariance matrix for the estimated coefficients, with row and column names matching ‘names(coef(object))‘. For ‘type = "classical"‘ this is the model-based covariance stored on the fit; for ‘"HC1"‘ and ‘"cluster"‘ it is a sandwich (robust or cluster-robust) covariance estimate of the same dimension.

Examples

```
data(copd)
fit <- panglm(fev1 ~ crp, data = copd, index = c("id", "visit"),
              model = "within", family = "gaussian")
vcov(fit)
vcov(fit, type = "cluster")
```

Index

∗ **datasets**
copd, 3

binomial_family *(*panglm-families*)*, 8

confint.panglm, 2
copd, 3

fitted.panglm, 4

gaussian_family *(*panglm-families*)*, 8
glance.panglm, 5

logLik.panglm, 5

negbin_family *(*panglm-families*)*, 8

panglm, 6
panglm-families, 8
panglm_dispersiontest, 9
panglm_hausman, 9
panglm_hurdle, 10
plot.panglm, 12
plot.panglm_hurdle, 13
poisson_family *(*panglm-families*)*, 8
predict.panglm, 13

summary.panglm, 14

tidy.panglm, 15

vcov.panglm, 16